INFLUENCE OF MIXING AND REACTION KINETICS ON THE PERFORMANCE OF A BIOLOGICAL REACTOR

INFLUENCE OF MIXING AND REACTION KINETICS ON THE PERFORMANCE OF A BIOLOGICAL REACTOR

By

PAUL MALCOLM CRAWFORD, B.Sc.

A Major Study Report

Submitted to the Faculty of Graduate Studies

in Partial Fulfillment of the Requirements

for the Degree

Master of Engineering

McMaster University

April 1973

MASTER OF ENGINEERING (Chemical Engineering)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: Influence of Mixing and Reaction Kinetics on the Performance of a Biological Reactor

AUTHOR: Paul Malcolm Crawford, B.Sc. (Queen's University)

SUPERVISOR: Dr. J. D. Norman

NUMBER OF PAGES: ix, 167

SCOPE AND CONTENTS:

The pulse response of a full scale aeration tank is mathematically modelled with an arbitrary network of idealized perfectly mixed and plug flow component vessels. The model is fitted in the frequency domain, then inverse transformed to the time domain. The soluble carbon concentration curve of batch biokinetic run is modelled by a modified logistics equation and a piecewise linear expression. The mixing and kinetic models are combined to predict the degree of conversion assuming the degree of segregation, J, to be one. The pulse responses of a lab scale tank for varying water flow rates are also modelled by the same methods. An attempt is made to correlate the mathematical model parameters to the water flow rate.

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to:

1. Dr. J. D. Norman for his guidance and encouragement through the course of this study.
2. Dr. K. L. Murphy for his help with the tracer response data.
3. Mr. T. E. Pollock, a fellow graduate student, for the use of his batch biokinetic data.
4. Mr. P. B. Melnyk, another fellow graduate, for his assistance and suggestions.
5. Miss Anna Di Trani for the typing of the report.
6. Mr. G. B. Crawford for his suggestion that I attend McMaster.
7. The consulting firm of Gore & Storrie Limited for the use of their printing facilities.
8. My wife, Judy, for her patience and encouragement throughout the year.

Financial support provided by the Department of Chemical Engineering, McMaster University is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 - INTRODUCTION

1.1 Problem Statement

A widely used process in the water pollution control field is biological treatment. Its purpose is to remove the biodegradable organic compounds present in wastewater. This is accomplished by mixing the wastewater with a liquor containing a high concentration of micro-organisms capable of degrading organic compounds. With the present trend toward stricter effluent regulations, it is necessary to be able to predict the degree of conversion of pollutants to harmless compounds.

In general, the flow rate and organic concentration of the influent stream are known. Therefore, to determine the level of removal of organics from the wastewater, only mathematical models for both the kinetics of the reaction and the mixing of the reactor are required. It is worthwhile knowing how complex these mathematical models must be in order to yield accurate conversion predictions.

There has been much emphasis on the development of complex kinetic models, but with the present state of measurement techniques it is questionable whether so much research is justified. In contrast, very little work has been done on the mixing phenomena of the biological reactor.

However, research has been done in other branches of science, but much of the information gathered is

inapplicable to the field of Environmental Engineering because of the existence of several unique conditions. These include: long holding times of from six hours to five days; diurnal hydraulic and organic loading variations; the presence of mixed and undefined wastes; and the presence of mixed and undefined microbial populations.

The purpose of this investigation is to determine whether the conversion predictions are more sensitive to the mathematical models chosen to describe the mixing or the kinetics.

In addition, because the process of biological treatment is operated under unsteady-state conditions, the effect of the fluid flow rate on the mathematical models chosen to represent the mixing characteristics of a given reactor will also be investigated.

1.2 Report Outline

Chapter 2 presents a review of the literature pertinent to this study. Various mixing models used in both the Chemical Engineering field and the Environmental Engineering field are introduced. Several biokinetic models, applied to both the fields of Microbiology and Environmental Engineering, are also described. Finally a review of previous work done in combining mixing models with kinetic models is given.

The theoretical background required to use the modelling technique employed to describe mixing is presented in Chapter 3. The concepts of the Laplace and Fourier transforms, the transfer function, and frequency

response are introduced. A method of obtaining frequency response information from a system's impulse response is described. Finally, a statistical technique of parameter estimation in the frequency domain is given.

In Chapter 4 a brief review of the experimental methods used by the two workers whose data was used is presented.

The data obtained from the experiment described in Chapter 4 is analyzed in Chapter 5. This involves fitting a PFTR/CSTR component model to data obtained from a prototype reactor. Then a biokinetic model is fitted to the data from a batch kinetic run. Finally these two models are combined to yield the degree of conversion of substrate. Also an attempt is made to correlate the parameters of the mixing models to the water flow rate.

Chapter 6 presents conclusions based on this study.

CHAPTER 2 - LITERATURE REVIEW

2.1 Mixing Models

In general, mathematical models to describe mixing are meant to characterize the residence time distribution of a given reactor. The curve, E(t), (or $E(\theta)$ in reduced time units) is a measure of the distribution of residence times of the fluid within the vessel, (L-1). Experimentally this curve is obtained by injecting an impulse or Dirac delta-function of tracer mass M into the input stream and continuously monitoring the tracer output concentration, C(t). Then,

$$E(t) = \frac{V\ C(t)}{M\ \bar{t}} \qquad (2.1)$$

$$\text{and } E(\theta) = \bar{t}\ E(t) \qquad (2.2)$$

where E(t) is the residence time distribution

t is time

V is the volume of the reactor

M is the mass of the tracer

C(t) is the tracer output concentration

$E(\theta)$ is the residence time distribution in reduced time units

θ is reduced time $(=t/\bar{t})$

$\bar{t}$ is the mean residence time $(= \frac{V}{Q})$

Q is the flow rate through the system

The variation in residence time experienced by the molecules during their passage through the system, i.e. the residence time distribution (RTD) will be termed "external mixing" in this report. The reason for this is that it may be found experimentally without reference to what occurs within the reactor.

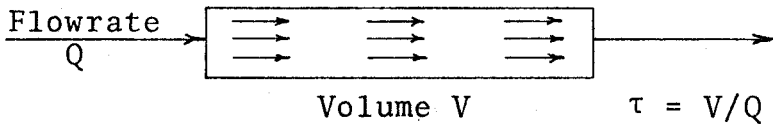
The two ideal extremes are plug flow and complete mixing. Levenspiel reviews the development of these models (L-1). The models are presented in Table 2.1.

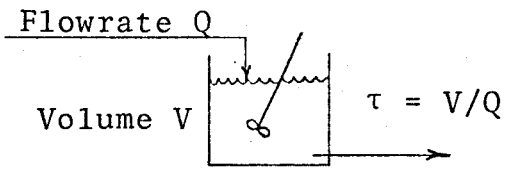
Since the plug flow tubular reactor (PFTR) and continuous stirred tank reactor (CSTR) are ideal models which cannot be achieved in practice the RTD of any non-ideal reactor must lie between that of a PFTR and a CSTR. The PFTR represents the limit of highest conversion for any reactor while the CSTR embodies the limit of best damping for shock loading.

In many cases in Environmental Engineering the mixing found is not sufficiently close to the ideal PFTR and CSTR. Examples of these vessels are aeration tanks, aerated lagoons, chlorine contact chambers, flocculation basins and streams and rivers. Therefore mathematical models other than the PFTR and CSTR must be used to describe external mixing.

One model used by several workers (T-1), (B-3) is the one dimensional axial dispersion model. Basically, it is a plug flow model which allows for mixing along the

TABLE (2.1) IDEAL PLUG FLOW AND COMPLETELY MIXED MODELS

Plug Flow Model	
Residence Time Distribution	$E(t) = \delta(t - \tau)$ (2.3)
Transfer Function	$P(s) = \exp(-\tau s)$ (2.4)

Completely Mixed Model	
Residence Time Distribution	$E(t) = \frac{1}{\tau} \exp(-t/\tau)$ (2.5)
Transfer Function	$P(s) = (1 + \tau s)^{-1}$ (2.6)

(ADAPTED FROM WILSON (W-1))

axis of the reactor. The dispersion number is used to characterize the amount of mixing caused by turbulence in the reactor. The dispersion number is the inverse of the Peclet number.

$$\frac{D}{uL} = \frac{1}{Pe} \tag{2.7}$$

where D = axial dispersion coefficient

u = mean velocity of fluid

L = length of reactor

and Pe = Peclet number

When $\frac{D}{uL} \rightarrow \infty$, the dispersion model approaches the behaviour of an ideally mixed vessel. For $\frac{D}{uL} = 0$, the model becomes the plug flow reactor.

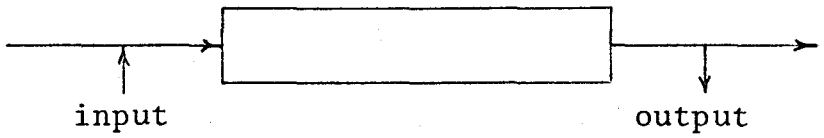
Since the model is defined by the second order differential equation (T-1)

$$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial x^2} - u \frac{\partial C}{\partial x} \tag{2.8}$$

boundary conditions must be established in order to solve the equation. One assumption that can be made is that dispersion does not occur at the inlet or the outlet. Then the boundary conditions are termed closed. Table 2.2 summarizes the model with closed boundary conditions.

Wilson (W-1) has developed a modelling technique utilizing networks of PFTR's and CSTR's in series and parallel. Although these models bear no resemblance to the actual conditions within the tank they are probably the most flexible models available to characterize mixing. As Wilson concludes,

TABLE (2.2) THE AXIAL DISPERSION MODEL FOR CLOSED BOUNDARY CONDITIONS

The Model	
Residence Time Distribution (W-1)	$$E(t)=\frac{Pe}{4}\exp\{\frac{Pe}{4}(2-t)\}\sum_{k=1}^{\infty}\{(\frac{\lambda_k \sin(2\lambda_k)}{\lambda_k^2+(\frac{Pe}{4})^2+\frac{Pe}{4}})\exp(\frac{-4\lambda_k^2 t}{Pe})\}$$ $$+\exp\{\frac{Pe}{4}(2-t)\}\sum_{k=1}^{\infty}\{\frac{4\lambda_k^2}{Pe}(\frac{\lambda_k \sin(2\lambda_k)}{\lambda_k^2+(\frac{Pe}{4})^2+\frac{Pe}{4}})\exp(\frac{-4\lambda_k^2 t}{Pe})\}$$ (2.9) where λ_k are the positive non-trivial roots of the transcendental equation: $$\tan(2\lambda) = (\frac{\lambda\ Pe}{2(\lambda^2-\frac{Pe}{4})})$$
Frequency Domain Solution	See Clements (C-3) or alternatively the appropriate subroutine listed in Appendix D. (2.10)

(ADAPTED FROM WILSON (W-1))

> "The approach developed does not restrict itself to using preconceived models but rather it enables the best model to be selected from a large number of possible alternatives."

Wilson also gives an excellent review of many other mixing models.

Unfortunately the RTD does not completely describe the mixing which occurs in a reactor. As Danckwerts (D-1) points out, it gives only the length of time which portions of the fluid remain the reactor. For a first order reaction this is sufficient to calculate the extent of conversion of reactants. But for more complex kinetics more information is required. The variation of environment experienced by molecules during their passage through the system will be called "internal mixing" because it cannot be measured from the tracer output concentration but must be found by examination of the interior of the reactor.

Danckwerts defined the degree of segregation, J, as the ratio of the variance of ages between the points in a reactor to the variance of the ages of all molecules in the reactor. This is the measure of internal mixing. Zwietering (Z-1) showed that, for a CSTR, J must be between 0 and 1 while for a PFTR J can only be 1. He also proved that for any intermediate non-ideal RTD the minimum value of J must be greater than 0 while the maximum value remains

at 1. Although Danckwerts (D-2) has derived methods for calculating J, in practice it is almost impossible to measure.

Rippin (R-1) discusses the conclusions reached by Danckwerts and Zweitering. Through the use of a two-environmental model he shows that even the specification of both external and internal mixing is insufficient to predict reactant conversion exactly. However, he does note that at the two extremes of segregation there can be no variation in mixing.

Finally Bischoff (B-1) presents a good review of mixing principles. He notes the need for methods of equipment modifications required to optimize the process.

2.2 Kinetic Models

While many mathematical models have been proposed to describe biological kinetics, only a few have been used extensively. Probably the most quoted is the Monod model. It was applied to pure bacterial cultures by Monod (M-5) to describe the batch growth rate. It has the form:

$$\mu = \mu_{max} \frac{C_C}{K_m + C_C} \tag{2.11}$$

where μ is the specific growth rate $(= \frac{1}{C_B} \frac{dC_B}{dt})$

μ_{max} is the maximum value of specific growth rate

K_m is the Michaelis-Menton constant

$(= C_C$ for $\mu = 0.5\ \mu_{max})$

C_C is the concentration of substrate

C_B is the concentration of biomass

t is time

Another widely used model is the one proposed by Garrett and Sawyer (G-1). This model was proposed in a study which attempted to fit the Monod model to batch data using activated sludge. They found instead that a discontinuous function consisting of two straight line portions fit the data better. In this model the specific growth rate is assumed to be proportional to the substrate concentration at low values, while for high values the growth rate is independent of the concentration of substrate.

There appears to be a great controversy over the Monod model in the literature. Some workers have verified it while others have proved it inadequate. There are also many who have modified it to improve upon its shortcomings.

Gaudy et al (G-2) compared the Monod model with that of Moser

$$\mu = \mu_{max} \frac{1}{1 + K^{x} C_C^{-x}} \qquad (2.12)$$

where K and x are empirical constants

and Tessier

$$\mu = \mu_{max} \left(1 - e^{-C_C/K} \right) \qquad (2.13)$$

where K is an empirical constant

They used a laboratory scale CSTR for their studies. Although they found that the Monod model fit the data best, they concede that it does not prove that the Monod

model is correct. In fact they felt that the two phase model of Garrett and Sawyer could be used if warranted by the practical application.

Naito et al (N-1) introduced the concept of activity used in catalytic reactions. They reasoned that they could explain the initial absorption phenomenon, so often found in activated sludge, by the use of the active and inactive site idea. The basic model which they utilized was still the Monod model, but a correction was made for the death of cells.

Verhoff et al (V-1) introduced another modification to the basic Monod model. They used a mechanistic approach to the problem of kinetics. For this model of cell growth it was postulated that there are two models: assimilation and ingestion. A first order death relation with respect to biomass concentration was included in the model. Their conclusion was that the dependency of the rate of growth of the cells on the substrate concentration also depended greatly on the mode of growth.

On the negative side Young et al (Y-2) showed that the Monod model was unable to accurately predict transient behaviour. They employed the systems approach to develop block diagram models for the kinetics. Subsequently, by using pulse testing methods they obtained frequency response data which enabled them to relate the output limiting substrate concentration and the specific growth rate. They stated that the Monod model is analogous

to a thermodynamic state equation which is incapable of predicting how a system will move from one state to another.

McCabe and Eckenfelder (M-1) favoured the Garrett and Sawyer model. They extended it to show a decrease in sludge mass in the so-called "auto-oxidation phase".

Edwards and Wilke (E-1) were not concerned with a mechanistic model of batch kinetics. Their concern was an equation which was capable of fitting the observed data. Thus they presented the desirable properties of the sought-for equation:

1. Capable of representing all phases of batch culture growth and metabolism.
2. Sufficiently flexible to fit many types of data without introducing distortion.
3. At least some of the fittable parameters have direct physical meaning.
4. Continuously differentiable explicit time derivative.
5. Derivative zero initially and at very large times.
6. Parameters easily evaluated.
7. Easy to use model once parameters determined.

On the basis of the above criteria they rejected the polynomial-in-time equation because of its failure to satisfy numbers 1, 3 and 5. The integrated form of the Monod model

was also rejected since it did not meet the properties in numbers 1 and 7. The third equation which they considered was the so-called logistic equation.

$$C_C = \frac{K}{1 + \exp\ (a_o + a_1 t)} \qquad (2.14)$$

where K, a_o and a_1, are constants.

While this equation was the most satisfactory of those looked into it was felt to be slightly inflexible. To improve upon this shortcoming they introduced a new second term in the denominator. This function was:

$$\exp\ (F(t))$$

where F(t) is a polynomial in t ($=a_o + a_1 t + a_2 t^2 + \ldots + a_n t^n$)

Upon applying this equation to numerous sets of data they found it very useful and accurate for the interpretation of batch kinetic data.

Pollock (P-1) gives a good review of bio-kinetics. He concludes that the "kinetic" models reported in the literature have little reaction kinetic significance and represent, at best, curve-fitting techniques.

2.3 Combined Kinetic and Mixing Models

Some previous work has been carried out in which the basic kinetic and mixing models were combined in order to predict the degree of substrate conversion and its sensitivity to various model parameters. Unfortunately much of the research has been of a theoretical nature using parameter values from the literature or using broad

assumptions regarding the parameters and the models. Here the usual procedure is to use a CSTR-PFTR-in-series combination or an n-CSTR's in series configuration with guessed parameter values. Only rarely have any tracer studies been carried out to substantiate the proposed parameter values.

Tsai et al (T-2) employed the Monod model for bacterial kinetics to predict the degree of conversion for three different mixing models, namely a CSTR with maximum mixedness, a completely segregated CSTR and a PFTR. When discussing the internal mixing they point out that Danckwerts (K-2) has stated that in a continuous fermentation, where no biomass is fed in, washout will occur if the internal mixing is in a state of complete segregation. However, with the activated sludge process biomass is recycled and thus could exist in a state of complete segregation without violating basic scientific principles. In conclusion they state that if the Monod model is used the effects of internal mixing are very important.

Sinclair and Brown (S-1) considered a laboratory scale chemostat. They reasoned that the mixing could be approximated by a two region mixing model with interchange. On this basis they employed the Monod model to predict the effect of bypassing, agitator position and the rate of interchange between regions. However their use of pure cultures and substrates rather limits the use of their findings.

Fan et al (F-1) discussed the effects of internal mixing extremes on washout and the attainment of multiple steady states. Again the Monod model was employed to describe the kinetics while the mixing models used were: n - CSTR's-in-series, CSTR-PFTR-in-series, dispersion model and PFTR.

Chen (C-2) is one of the very few researchers to consider the effects of unsteady state mixing. Examples are given which show the performance of a CSTR for sinusoidally varying flow rates. Unfortunately the effects of varying substrate concentration in the feed stream were not considered. He also employed the Monod model to illustrate the difference between complete segregation and maximum mixedness in a CSTR subjected to a sinusoidally varying flow rate. His conclusion is that complete segregation is highly unfavourable for micro-organism reactions. Thus he suggests that, when the micro-organism flocs are recirculated from the secondary clarifiers, good mixing should be supplied at the feed stream for maximum substrate utilization.

Goda and Nakanishi (G-3) introduced two mixing models. The first was a PFTR followed by a CSTR. Then they derived a much more complex model which was essentially a modified dispersion model. The Monod model was applied to calculate biomass and substrate concentrations. They concluded that a PFTR would always be better than a CSTR for zero or first order kinetics.

Grieves et al (G-4) decided that the Cholette-Cloutier mixing model would be better than either the PFTR assumption used by McCabe and Eckenfelder (M-1) or the CSTR assumption made by McKinney (M-2). This model uses a PFTR-in-series with a CSTR, bypassing, and dead volume components. Again the Monod model was used for the bio-kinetics. In another paper Grieves et al (G-5) added a recycle stream to the basic Cholette-Cloutier model. In addition to the one phase Monod model they used the two phase model of Garrett and Sawyer. In conclusion, they found little difference between the one and two phase kinetic models.

Milbury et al (M-4) continued the above work. They carried out tracer studies on laboratory scale aeration tanks. Two different traces were employed in the research, namely NaCl and skim milk powder. Later in the work they used the skim milk powder as a substrate for a mixed culture. However, they noted that the skim milk had slightly different characteristics as a tracer. Thus they concluded that tracer studies must use substances that behave in the same manner as the critical components of the feed stream. Batch studies were used to evaluate the parameter of the Monod model. Then the experimental results from a laboratory scale aeration tank were compared with the theoretical calculations and mixing model. They concluded that the effect of short circuiting on a lab-oratory scale system is much less than that predicted

by the theorectical models. They also felt that absorption was the main cause of the reduction in effect of short circuiting especially for collodial nutrients.

Bischoff (B-2) attempted to find the optimal reactor for continuous culture. Using Monod kinetics he found that a CSTR followed by a PFTR would be best. He also states:

> "Until more is known about microbial growth, however, there is probably a low limit to the degree of complexity justified for design calculations."

Fan et al (F-2) added a term for endogenous respiration to the Monod model. Then using the n-CSTR's in series and PFTR-CSTR-in-series models they discussed the various possible internal mixing states and their effect on substrate conversion and micro-organism growth.

McKinney (M-2) introduced the concept of complete mixing activated sludge and developed the mathematics to show how it could be applied to design procedures for sewage treatment plants.

The basic assumptions he made were that the mixing model should be a CSTR and that the kinetics should be first order as in the first phase of the Garrett and Sawyer model. McKinney and Ooten (M-3) give a much more detailed analysis of the procedure as well as some experimental results to substantiate the assumptions.

CHAPTER 3 - THEORY

3.1 The Laplace Transform and Transfer Functions

One technique used in solving engineering problems is the transformation relating time functions to frequency dependent functions of a complex variable, called the Laplace transform (D-3). Consider f(t), a real function of a real variable, t, which is defined for t>0. Its Laplace transform pair is defined as:

$$L\{f(t)\} \equiv F(s) = \int_0^\infty f(t)e^{-st}dt \qquad (3.1a)$$

$$L^{-1}\{F(s)\} \equiv f(t) = \frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty} F(s)e^{st}ds \qquad (3.1b)$$

where F(s) is the Laplace transform function of f(t)

s is a complex variable defined by

$$s \equiv \sigma + j\omega$$

σ and ω are real variables

$c > \sigma_0$ (σ_0 is defined below)

$j = \sqrt{-1}$

t is a real variable always denoting time

The existence of the Laplace transform is dependent upon the convergence of the transform integral, that is,

$$\int_0^\infty |f(t)|e^{-\sigma_0 t}dt < +\infty \qquad (3.2)$$

where σ_0 is a real number

If equation (3.2) holds then f(t) is Laplace transformable for any value of s for which the real part of s, σ, is

greater than σ_0.

The Laplace transform and its inverse have several properties which are useful in the solution of linear, constant coefficient differential equations. Only the ones pertinent to this study will be introduced here.

1. Both the Laplace transform and the inverse Laplace transform are linear transformations between functions defined in the time domain and functions defined in the s - domain, that is

$$L\{a_1f_1(t) + a_2f_2(t)\} = a_1F_1(s) + a_2F_2(s) \quad (3.3)$$

$$L^{-1}\{a_1F_1(s) + a_2F_2(s)\} = a_1f_1(t) + a_2f_2(t) \quad (3.4)$$

2. The Laplace transform of the function f(t-T) (Time Delay) where T>0 and f(t-T) = 0 for $t \leq T$ is

$$L\{f(t-T)\} = e^{-sT}F(s) \quad (3.5)$$

where $F(s) = L\{f(t)\}$

Also,

$$L^{-1}\{e^{-sT}F(s)\} = f(t-T) \quad (3.6)$$

3. The Laplace transform of the derivative df/dt of a function f(t) whose Laplace transform is F(s) is

$$L\{df/dt\} = s\,F(s) - f(0^+) \quad (3.7)$$

where $f(0^+)$ is the initial value of f(t) evaluated as the one sided limit of f(t) as t approaches zero from positive values.

The transfer function of a time invariant linear system is defined as the ratio of the Laplace transform of the output variable to the Laplace transform of the input variable with all initial conditions set to zero, that is, (D-3),

$$P(s) = \frac{Y(s)}{X(s)} \tag{3.8}$$

Thus the transfer function describes the system in the s-domain in terms of an algebraic relationship rather than a linear differential equation in the time domain. This fact facilitiates of the use of "block diagram algebra" which greatly simplifies the handling of large complex systems. The transfer function has several properties to be found useful in this study.

1. The transfer function of a system is the Laplace transform of its impulse response. That is, if the input to a system with transfer function P(s) is an impulse and all initial values are zero, the transform of the output is P(s).

   This can easily be seen by considering that,

   $$L\{\delta(t)\} = 1.0 \tag{3.9}$$

   where $\delta(t)$ is the Dirac delta function defined by

   $$\delta(t) = \infty \text{ for } t = 0$$

   $$= 0 \text{ for } t > 0$$

   $$\int_0^\infty \delta(t)\,dt = 1.0$$

Therefore,

for $X(s) = \delta(s)$

$$P(s) = Y(s)/1.0 = Y(s)$$

Thus, the Laplace transform of a mixing system's residence time distribution (output concentration response to a unit impulse tracer input) will yield the system transfer function directly (W-1).

In order to obtain the system's time response the inverse Laplace transform can be used. The corollary of the preceding paragraph is that inverse Laplace transformation of a mixing vessel's transfer function will yield the residence time distribution directly.

The inverse transform may be performed analytically as follows (D-3):

Consider the general form of the Laplace transform, the rational function

$$F(s) = \frac{\sum_{i=0}^{m} b_i s_i}{\sum_{i=0}^{n} a_i s_i} \qquad (3.10)$$

where $a_n = 1$ and $n \geq m$

By the fundamental theorem of algebra the denominator polynomial equation

$$\sum_{i=0}^{n} a_i s_i = 0 \qquad (3.11)$$

has n roots some of which may be repeated. Suppose that the denominator polynomial equation above has n_1 roots equal to p_1, n_2 roots equal

to $-p_2$, ..., n_r roots equal to $-p_r$, where $\sum_{i=1}^{r} n_i = n$.

Then,

$$\sum_{i=0}^{n} a_i s_i = \prod_{i=1}^{r} (s + p_i)^{n_i} \tag{3.12}$$

and the rational function F(s) can be represented as a partial fraction expansion.

$$F(s) = \sum_{i=1}^{r} \sum_{k=1}^{n_i} \frac{c_{ik}}{(s + p_i)^k} \tag{3.13}$$

Where the c_{ik} coefficients are called the "residues" of F(s) at $-p_i$, i = 1, 2,..., r and are given by:

$$c_{ik} = \frac{1}{(n_i - k)!} \frac{d^{n_i - k}}{ds^{n_i - k}} \{(s + p_i)^{n_i} F(s)\} \Big|_{s = p_i} \tag{3.14}$$

Thus any transfer function having the form of equation (3.10) can be inverse Laplace transformed by using the relationship:

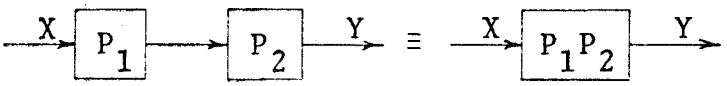
$$f(t) = L^{-1}\{F(s)\} = \sum_{i=1}^{r} \sum_{k=1}^{n_i} \frac{c_{ik}}{(k - 1)!} t^{k-1} e^{-p_i t} \tag{3.15}$$

2. The transfer functions of subsystems can be combined with each other to define an overall system transfer function by means of "block diagram algebra". The two basic rules of "block diagram algebra" to be utilized here will be:

a) Combining blocks in cascade

$$Y = P_2 Z \text{ and } Z = P_1 X \equiv Y = (P_1 P_2)\, X \qquad (3.16)$$

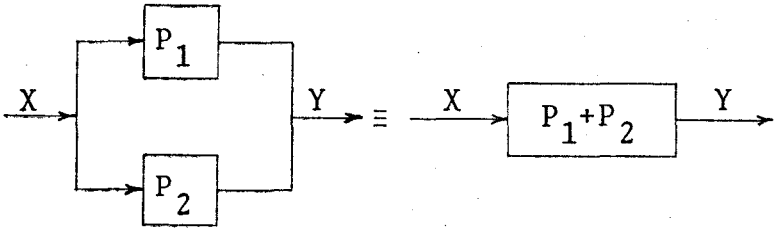
that is,



b) Combining blocks in parallel

$$Y = P_1 X + P_2 X \equiv Y = (P_1 + P_2)\, X \qquad (3.17a)$$

that is,

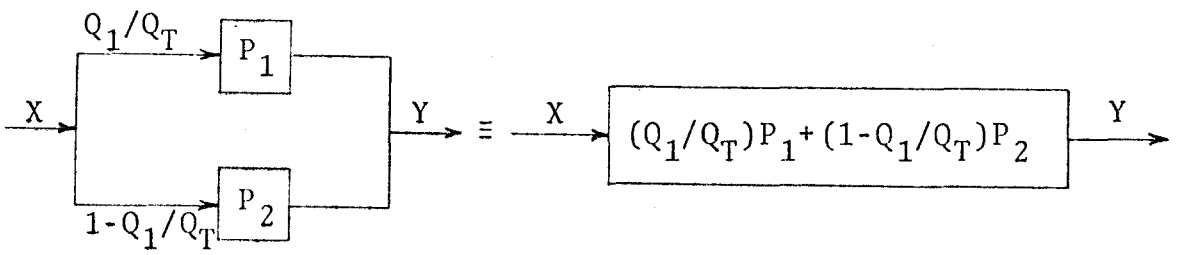


However, in this field of study, since the input and output singals are concentration measurements, a mass balance must be satisfied at all junction points. Therefore, whenever two or more parallel paths (streams) join, the resulting transfer function will be the sum of the individual transfer functions of each path weighted according to the respective fuild flow rate in each path. Thus the preceding rule must be amended slightly as follows:

$$Y = (Q_1/Q_T)\, P_1 X + (1-Q_1/Q_T)\, P_2 X$$

$$\equiv Y = \{(Q_1/Q_T) P_1 + (1-Q_1/Q_T) P_2\}\, X \qquad (3.17b)$$

that is,



3.2 The Fourier Transform and Frequency Response

The Fourier transform pair is defined as (S-2):

$$F(j\omega) \equiv F\{f(t)\} = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t}dt \qquad (3.18a)$$

$$f(t) \equiv F^{-1}\{F(j\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(j\omega)e^{j\omega t}d\omega \qquad (3.18b)$$

The existence of the Fourier transform of f(t) depends on the satisfaction of the following inequality:

$$\int_{-\infty}^{+\infty} |f(t)|dt < \infty \qquad (3.19)$$

Thus the Fourier transform and the Laplace transform are equivalent for f(t) if f(t) = 0 for t < 0 (in equations (3.18a),(3.18b) and (3.19)) and the real part of s is equal to zero in equations (3.1a), (3.1b) and (3.2).

The steady state response of a stable system to an input x = A sin ωt given by (D-3):

$$y_{ss} = A\,|P(j\omega)|\,\sin(\omega t + \phi) \qquad (3.20)$$

where $|P(j\omega)|$ = magnitude of $P(j\omega)$

ϕ = arg $P(j\omega)$

and $P(j\omega)$ is found from P(s) by replacing s with $j\omega$ or from Fourier transforming the original system equations in the time domain.

Therefore the system frequency response is defined by the output:input amplitude ratio, $|P(j\omega)|$, and the output: input phase angle, arg $P(j\omega)$, for all ω where,

$$|P(j\omega)| = \sqrt{\text{Real } \{P(j\omega)\}^2 + \text{Imag. } \{P(j\omega)\}^2} \qquad (3.21)$$

$$\arg P(j\omega) = \tan^{-1} \frac{\text{Imag. } \{P(j\omega)\}}{\text{Real } \{P(j\omega)\}} \qquad (3.22)$$

and P(jw) is a complex number.

As an example consider the case of the ideal mixer whose transfer function is (W-1):

$$P(s) = (1 + \tau s)^{-1} \qquad (3.23)$$

Substitute $s = j\omega$:

$$P(j\omega) = \frac{1}{1 + j\omega\tau} \qquad (3.24)$$

Multiply top and bottom by $(1 - j\omega\tau)$

$$P(j\omega) = \frac{1 - j\omega\tau}{1 + \omega^2\tau^2} = \frac{1}{1 + \omega^2\tau^2} - j \frac{\omega\tau}{1 + \omega^2\tau^2} \qquad (3.25)$$

$$\text{Note that Real } \{P(j\omega)\} = \frac{1}{1 + \omega^2\tau^2} \qquad (3.26)$$

$$\text{and Imaginary } \{P(j\omega)\} = \frac{\omega\tau}{1 + \omega^2\tau^2} \qquad (3.27)$$

Then using the definitions of equations (3.21) and (3.22):

$$|P(j\omega)| = \frac{1}{1 + \omega^2\tau^2} \qquad (3.28)$$

$$\text{and } \arg P(j\omega) = -\tan^{-1} (\omega\tau) \qquad (3.29)$$

Frequently, the magnitude ratio is expressed as a "gain" in terms of decibels:

$$\{\text{Gain}\} = 20 \log_{10} |P(j\omega)| \qquad (3.30)$$

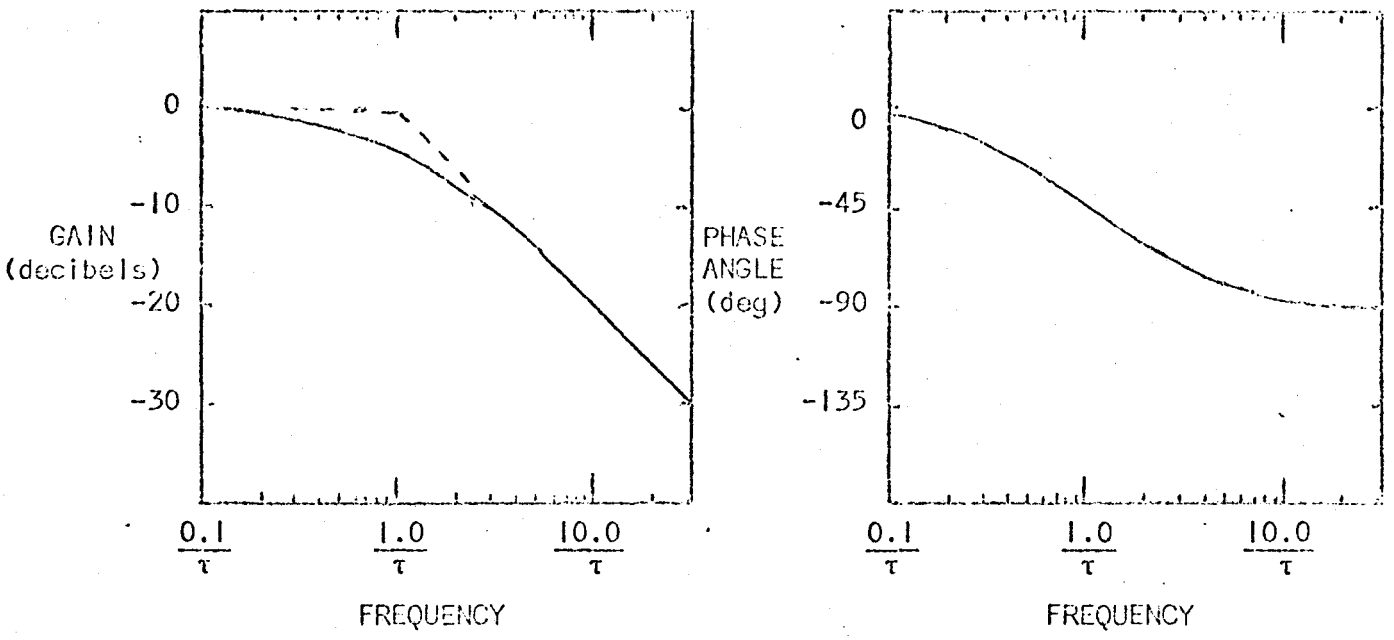
In order to visualize frequency response information two different graphical techniques may be employed, the Bodé plot or the polar plot. The Bodé plot consists of two diagrams, one being a plot of the magnitude ratio or gain versus frequency and the other being a plot of the phase angle versus frequency. The polar plot combines this information into a single diagram and the magnitude ratio and phase angle (or the corresponding real and imaginary responses) are plotted for various frequencies. An example of the Bodé and polar plots for the ideally mixed vessel is shown in Figure (3.1). Note that the low and high frequency asymptotes of the amplitude portion of the Bode plot intersect at a so-called "break" or "corner frequency" of $\omega = 1/\tau$ and that the high frequency asyomptote slope is -20db/decade.

3.3 Analysis of Experimental Pulse Data

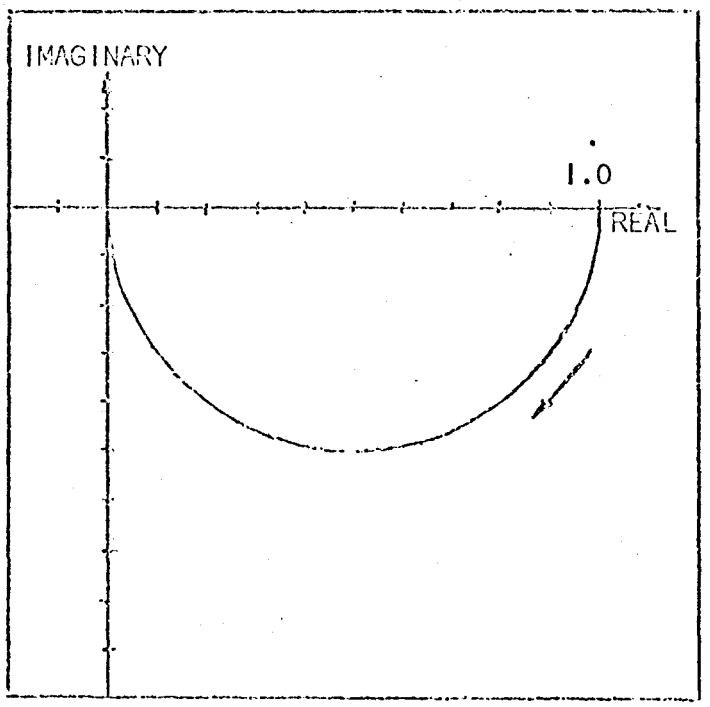
While the previous section dealt with the derivation of the frequency response of a mathematical model, the present section will present a method of obtaining the frequency response of a prototype system.

While disturbing the system with an input consisting of sinusoidal fluctuations of a given frequency and measuring the output waveform is probably the most direct way of obtaining frequency response information, it also has many drawbacks. These include the long time required to obtain data over a broad range of frequencies and the expense and difficulty of constructing accurate

FIGURE (3.1) BODÉ AND POLAR PLOTS FOR AN IDEAL MIXER



Bodé Plot for an Ideal Mixer



Polar Plot for an Ideal Mixer

(ADAPTED FROM WILSON (W-1))

sinusoidal generators for some process operating variables (W-1).

One alternative to direct sinusoidal testing is impulse testing. In this method an impulse (Dirac-delta function) disturbance is used as the input to the system.

Clements and Schnelle (C-4), who have investigated numerical methods for the computation of a system's frequency response from experimental pulse information, have defined the normalized frequency content of a pulse, $S(\omega)_n$ at any frequency, ω, to be the ratio of the Fourier transform of the pulse at frequency = ω to the Fourier transform of the pulse at zero frequency.

$$S(\omega)_n = \frac{\int_0^\infty f(t)e^{-j\omega t}dt}{\int_0^\infty f(t)dt} = \frac{F\{f(t)\}}{(\text{Area under pulse})} \qquad (3.31)$$

Note that if the input pulse $x(t)$ is equal to $\delta(t)$ then $S_x(\omega)_n = 1.0$ for all ω. In experimental pulse testing, the pulse usually has non-zero values only for some finite time, T. Thus the upper limit of integration in equation (3.31) may be set at T. Now, using $y(t)$ to represent the output pulse, the magnitude ratio is given by:

$$MR(\omega) = \left| \int_0^{T_y} y(t)e^{-j\omega t}dt \right| \qquad (3.32)$$

and the phase angle is given by

$$\phi(\omega) = \arg \{S_y(\omega)_n\} \qquad (3.33)$$

On substituting the Euler identity, $\exp(-j\omega t) = \cos(\omega t) - j \sin(\omega t)$, into equation (3.32), algerabic manipulation yields the real and imaginary parts of the system's

experimental complex frequency response:

$$RL(\omega) = \int_0^{T_y} y(t) \cos(\omega t) dt \tag{3.34}$$

$$IM(\omega) = \int_0^{T_y} y(t) \sin(\omega t) dt \tag{3.35}$$

Then the magnitude ratio and phase angle are given by:

$$MR(\omega) = \sqrt{RL(\omega)^2 + IM(\omega)^2} \tag{3.36}$$

$$\phi(\omega) = \tan^{-1} \frac{\{IM(\omega)\}}{\{RL(\omega)\}} \tag{3.37}$$

Both $RL(\omega)$ and $IM(\omega)$ can be evaluated numerically by a suitable quadrature method (W-1). (Note that Wilson develops this topic in terms of pulse but not necessarily impulse input disturbances. The development given here is therefore simpler and should not be used if the input pulse is suspected of not being a good approximation to the Dirac-delta function.)

Clements and Schnelle (C-4) have found that Filon's Quadrature was the best numerical method available to evaluate equations (3.34) and (3.35). A computer program, based on Wilson's program, was used in this study for the calculation of the Fourier transform of the output data from the impulse testing done by Timpany, (T-1). A program listing appears in Appendix B.

3.4 Least Squares Parameter Estimation

Previously, the methods for obtaining the frequency response of a mathematical model and of a prototype system have been presented. Now a method of estimating the model parameters, in order that the model

response matches the prototype response, will be developed. Figure (3.2) illustrates the steps involved in the method.

A popular statistical criterion used to fit a model to a set of data is based on the minimization of the sum of squared differences between the observed prototype response and the predicted model response under similar conditions. This is the least residual sum of squares criterion. The model parameters are established such that the residual sum of squares between the prototype and model responses is at a minimum.

The residual sum of squares for n data points is defined by:

$$RSS = \sum_{i=1}^{n} \{ y_{i_{obs}} - y_{i_{pred}} \}^2 \qquad (3.38)$$

where $y_{i_{obs}}$ and $y_{i_{pred}}$ are the observed and predicted responses of the experimental system and the mathematical model respectively for the i'th data point.

If the responses are continuous functions of time, then equation (3.38) becomes (W-1):

$$RSS = \int_0^{\infty} \{ y_{obs}(t) - y_{pred}(t) \}^2 dt \qquad (3.39)$$

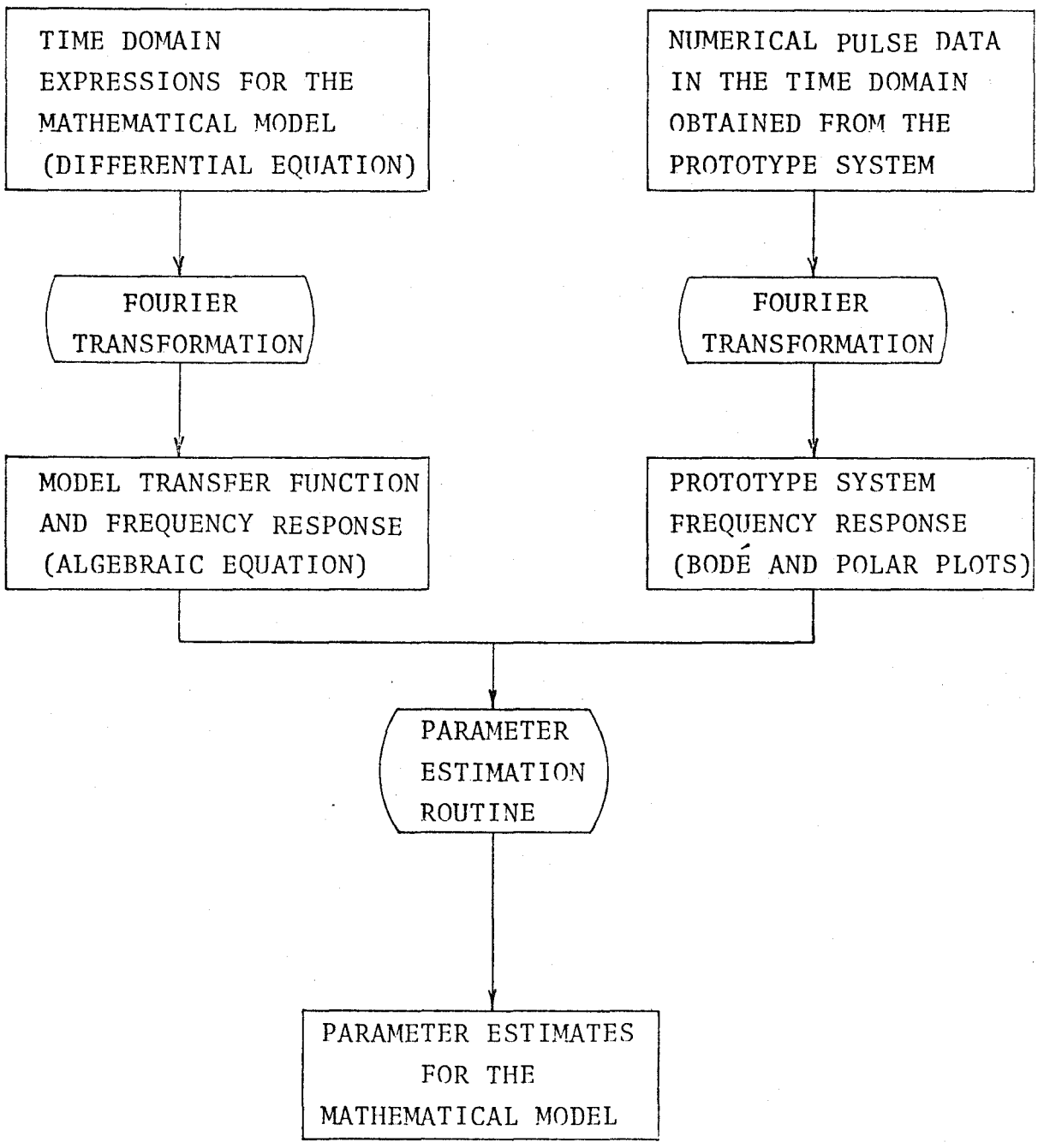
If the residual error at time, t, is defined as:

$$e(t) = y_{obs}(t) - y_{pred}(t) \qquad (3.40)$$

then

$$RSS = \int_0^{\infty} \{e(t)\}^2 dt \qquad (3.41)$$

FIGURE (3.2) MATCHING THE MODEL TO THE PROTOTYPE



(ADAPTED FROM WILSON (W-1))

and using Parseval's Theorem (S-2):

$$RSS = \int_0^\infty \{e(t)\}^2 dt = \frac{1}{\pi} \int_0^\infty |E(j\omega)|^2 d\omega \tag{3.42}$$

where $E(j\omega) = F\{e(t)\} = Y_{obs}(j\omega) - Y_{pred}(j\omega)$ (3.43)

Now, the experimentally observed and predicted model frequency responses may be expressed as the sum of their real and imaginary parts as follows:

$$E(j\omega) = \{RL_{obs}(\omega) + j\ IM_{obs}(\omega)\} - \{RL_{pred}(\omega) + j\ IM_{pred}(\omega)\} \tag{3.44}$$

and therefore

$$|E(j\omega)|^2 = \{RL_{obs}(\omega) - RL_{pred}(\omega)\}^2 + \{IM_{obs}(\omega) - IM_{pred}(\omega)\}^2 \tag{3.45}$$

and combining equations (3.42) and (3.45), there results:

$$RSS = \int_0^\infty \{e(t)\}^2 dt = \frac{1}{\pi} \int_0^\infty (\{RL_{obs}(\omega) - RL_{pred}(\omega)\}^2 + \{IM_{obs}(\omega) - IM_{pred}(\omega)\}^2) d\omega \tag{3.46}$$

Therefore minimizing the squared vectorial residuals between the observed and predicted responses in the frequency domain is equivalent to minimizing the residual sum of squares in the time domain. Theoretically then, the same "best" parameter values should be obtained by minimizing the residual sum of squares in either the time or the frequency domain according to the expressions defined by equation (3.46),(W-1).

If a mathematical model is linear in the parameters, then the partial derivative of the model response with respect to the parameters vanishes. The models used in this

report are found to be non-linear by this definition and hence a non-linear least squares parameter estimation routine must be used. Because of the iterative nature of these procedures, rapid convergence is the major selection criterion.

CHAPTER 4 - EXPERIMENTAL METHODS

Although no experimentation to obtain data was actually carried out in this work, a brief summary of the experimental procedures utilized by the original researchers is presented here. For a more complete description the original works cited, (T-1) and (P-1) should be consulted.

4.1 Tracer Study

4.1.1 Field Prototype Tests

The prototype used was an activated sludge aeration tank 66 feet long, 15 feet deep and 30 feet wide. City water from a nearby fire hydrant was employed to minimize the flow fluctuations. An inlet baffle was installed in the tank to lessen the chances of hydraulic short circuiting. The temperature of the incoming sewage was 19.5^{o}F. Air diffusers in the tank were of the "Sparger" type and were adjusted by eye to obtain uniform distribution of air. A rectangular weir was used to measure the sewage flow.

The impulse input of the tracer solution was accomplished by adding a 10 litre jar of the stock solution -1% Rhodamine Lissamine B-200 - to the influent. This addition marked the beginning of the test. Re-circulation of the tracer was eliminated by: using only tap water for the spray foam controls; eliminating the wasting of the secondary clarifier solids to the primary settler; and stopping the return of solids to the aeration tank. Samples of the aeration tank effluent, taken at

specified times, were centrifuged for about five minutes to remove solids. Then the fluorescence was determined using the discrete sampling door on the fluorometer.

4.1.2 Laboratory Scale Model

The laboratory scale runs were made on a rectangular tank five feet long, three feet wide and having a maximum depth of two and one half feet. Uniform diffused air was supplied through carborundum tubes recessed in the tank bottom and measured by a positive displacement type air meter. The water flow was measured by a six inch diameter Reynolds column using brass orifice plates of varying diameters to gauge a range of flows. Constant flow was maintained by a constant head tank with continuous overflow. The flow entered and exited by means of short weirs. The water temperature was controlled to 20^{o}F.

At the beginning of each run, stock tracer solution was added at the entrance weir at about 0.03 U.S. g.p.m. for five to ten seconds. A continuous sampling tube syphoned the effluent to the fluorometer whose output was recorded by a strip chart recorder.

4.2 Biokinetic Study

The batch bio-reactors were fabricated from 350 ml. fritted glass disc equipped Buchner funnels, which were glass welded to four litre pyrex percolators. Compressed air at 20 p.s.i. supplied through a glass wool filter to the funnel stem escaped from the fritted glass

discs as fine dispersed bubbles. This action provided a completely mixed hydraulic regime as well as saturation levels (six mg/1) of dissolved oxygen. A water cooled condensor was mounted on top of each reactor to prevent evaporation losses. Organic carbon, as dextrose, was used as a limiting nutrient to observe the growth phenomenon. The mixed microbial cultures employed were obtained from a continuous bench scale reactor operated for several months in a chemical environment similar to that of the batch runs.

At regular intervals after the run began, a ten to twenty ml. grab sample was pipetted from the centroid of the reactor. The sample was filtered through a 0.45 μ pore diameter membrane filter for a maximum of fifteen minutes. The purpose of the time limit was to reduce the possibility of non-representative measurement of substrate levels as a function of time.

The soluble organic carbon in the filtrate of the sample was measured on an infared carbonaceous analyzer. First, the inorganic carbon was converted to CO_2 by titration with two drops of concentrated HCl and liberated by means of a five minute helium purge. Then twenty microlitre aliquots were syringe collected and injected until three successive determinations produced output signals differing by less than one percent of full scale. The organic carbon concentration was determined by linear interpolation from

five point analyzer calibration curves made using solutions of a stable organic compound (sodium oxalate in distilled water) of known carbon concentration. The run was stopped when the carbon concentration remained constant for at least 0.5 hours (i.e. steady state).

CHAPTER 5 - DATA ANALYSIS

5.1 Mixing and Kinetic Models for Full Scale Prototype

5.1.1 Summary of Procedures

Applying a slightly modified version of Wilson's (W-1) technique, mathematical models were fitted to the prototype tracer study data of Timpany (T-1). Then using the Edwards model (E-1) the batch biokinetic data of Pollock (P-1) was fitted. A piecewise linearized approximation was also made on Pollock's data. It should be noted here that data concerning the changes in biological solids were completely ignored and only the degradation of soluble organic carbon was considered.

In order to predict the degree of conversion the information from both the mixing and kinetic models must be combined. However, for the mixing model both the external and the internal mixing must be known. External mixing is characterized by the RTD and is thus known in this case. Internal mixing is measured by J, the degree of segregation. This parameter is very difficult to measure so an assumption will be made as to its value. As Busch points out (B-4),

> " ... leads to the recognition of the reaction as heterogenous or two-phase, in which the dissolved organic molecules have an affinity for attachment to the solid biomass clumps. Segregation can now be used as a conceptual

analogy to visualize that each clump and its attached molecules move through the reactor as an entity and indeed, behave exactly the same as in a batch reactor, even to showing identical average reaction rates for the same initial condition."

The above is, of course, the definition of complete segregation or J=1 and this, in fact, will be the assumption made here for the purposes of conversion prediction. Levenspiel (L-1) gives the equation for this conversion.

$$X = 1 - \int_0^\infty \frac{(C)}{(C_o)}_{batch} E(t)dt \qquad (5.1)$$

where X is the degree of conversion

$\frac{(C)}{(C_o)}_{batch}$ is concentration of reactant as function of time for a batch reaction divided by the initial concentration of the reactant.

E(t) is the residence time distribution of the reactor.

t is time.

Using the above equation conversion predictions were made as a function of total reactor residence time. Then the percent differences between the conversion based on the most accurate mixing and kinetic models and the other predictions were calculated.

5.1.2 Parameter Estimation for Mixing Model

5.1.2a Least Squares Tail Fit

As Timpany (T-1) has shown in order to obtain an accurate mean residence time for an impulse test tracer curve, the curve must be extended to approximately 8 residence times. To carry out an experiment for this period of time is impractical. However, as noted by Yamazaki and Ichikowa (Y-1) the tail of the curve can be satisfactorily fitted by an equation of the form

$$C = a\, e^{bt} \qquad (5.2)$$

where C is the concentration of tracer

t is time

a and b are empirical constants.

Thus a computer program was written extend the data to approximately 8 residence times. The program appears in Appendix A.

First of all the raw fluorescence data for Timpany's run #61 was transferred to data cards, (see Appendix J). The data for this run is shown in Table 5.1.

TABLE (5.1)

EXPERIMENTAL DATA FOR RUN

Run No.	Air Flow (cfm)	Water Flow (USgpm)	Temp ^{o}C	Tank Length (ft.)	Tank Width (ft.)	Tank Depth (ft.)	Theoretical Det. Time
61	660	850	19.5	66	30	15	4.36

The time between the beginning of the run and the first appearance of the tracer in the effluent was subtracted off and modelled as a PFTR-in-series with the rest of the model. Two step sizes were used. The first size was used until slightly past the peak of the curve. This was a small step because of the rapid changes in fluorescence readings. It was chosen arbitrarily so that the trapezodial rule used for the numerical integration would be reasonably accurate. Past the peak on the tracer curve, the changes are comparatively slow and hence a larger step size may be employed. The second step size was chosen in proportion to the theorectical residence time of the reactor. The ratio of step size to residence time used was approximately 1.25 minutes per hour.

The program converts the raw fluorescence readings to the concentration of tracer using the calibration curves derived by Timpany (T-1). Then the program fits an exponential curve (i.e. $C = a\, e^{bt}$), using the least squares method (C-1), to the tail portion of the curve to the required 8 residence times.

The output of this program is punched cards suitable for inputing to the Fourier transform program.

5.1.2b Fourier Transformation

In order to fit models to the data in the frequency domain the raw data must be Fourier transformed. A program was written by Wilson (W-1) to perform a

numerical Fourier transform. This program was suitably modified to accept the data from the previous program. A listing of the program appears in Appendix B.

The program calculates the first moment (i.e. the mean residence time) of the C vs t curve and the Fourier transform of the curve. The trapezoidal rule is used for the first moment calculations while Filon's quadrature is used for the Fourier transformation.

The range of frequencies used in the Fourier transformation is critical. Approximately two decades of frequency are required for modelling purposes. However, the high frequency data must be limited as it can be unreliable. The criterion adopted by Wilson (W-1) was that of Hays which states that the frequency content of the output pulse must be greater than or equal to the percentage error in the magnitude ratio. Knowledge of this percentage error, however, implies the availability of replicate data. In the study done by Timpany (T-1) no duplicate data was available, and so a different approach was adopted. The high frequency cutoff used was:

$$\omega \simeq \frac{10}{\tau} \qquad (5.3)$$

where ω is the frequency in radian/hr.

τ is the theorectical residence time in hours.

By comparison with Wilson's results the cutoff was slightly low and hence the data was considered to be reliable.

The output of this program was again in the form of punched cards for use in subsequent programs.

5.1.2c Category Selection

Wilson's (W-1) models wre divided into categories according to their "complexity". The "complexity" is defined as the excess of poles over zeros in the transfer function. Certain characteristics of the frequency domain data can be used to indicate in which category the best model will be found. While Wilson uses primarily the Bodé plot to select the proper category (see Table 5.2), the use of his procedure in this work was found to yield the incorrect category on several occasions. Thus a procedure using the polar plot exclusively was adopted. Table 5.3 illustrates the method.

For example, if the high frequency portion of the polar trajectory was asymptotic to the negative real axis, then category 2 models having 2 more poles than zeros would be selected.

The Bodé magnitude diagram for Timpany's prototype run, #61, is shown in Figure 5.1. The slope of the high frequency asymptote is -20 decibels per decade and therefore using Wilson' category selection criteria the category to be selected would be category 1. However, looking at Figure 5.2, the polar plot of the same data, the high frequency trajectory is seen to be asymptotic to the negative real axis. Therefore the category to be selected should be category 2. Later it will be shown

TABLE (5.2) CRITERIA FOR SELECTION OF THE APPROPRIATE MODEL CATEGORY

High Frequency Asymptote Slope in Amplitude Ratio Portion of Bodé Plot	Polar Plot Axis Appearing Asymptotic to High Frequency Trajectory of Transformed Data	Appropriate Models to select	
		Model Category	$\frac{\#Zeros}{\#poles}$
-20	Negative Imaginary Axis	1	$[\frac{N}{N+1}]$ *
-20	Any Axis	1	$[\frac{N}{N+1}]$ **
-40	Negative Real Axis	2	$[\frac{N}{N+2}]$ *
-40	Any Axis	2	$[\frac{N}{N+2}]$ **
-60	Positive Imaginary Axis	3	$[\frac{N}{N+3}]$ *
-60	Any Axis	3	$[\frac{N}{N+3}]$ **
-80	Positive Real Axis	4	$[\frac{N}{N+4}]$ *
-80	Any Axis	4	$[\frac{N}{N+4}]$ **

*N is an integer, usually small (e.g. 0,1,2,3.....)

**$N \longrightarrow \infty$

(ADAPTED FROM WILSON (W-1))

TABLE (5.3)

REVISED CRITERIA FOR SELECTION

OF THE

APPROPRIATE MODEL CATEGORY

Polar Plot Axis Appearing Asymptotic to High Frequency Trajectory of Transformed Data	Appropriate Models to Select	
	Model Category	# Zeros / # Poles
Negative Imaginary Axis	1	$\frac{N}{N+1}$ *
Negative Real Axis	2	$\frac{N}{N+2}$ *
Postive Imaginary Axis	3	$\frac{N}{N+3}$ *
Postive Real Axis	4	$\frac{N}{N+4}$ *

* N is an integer.

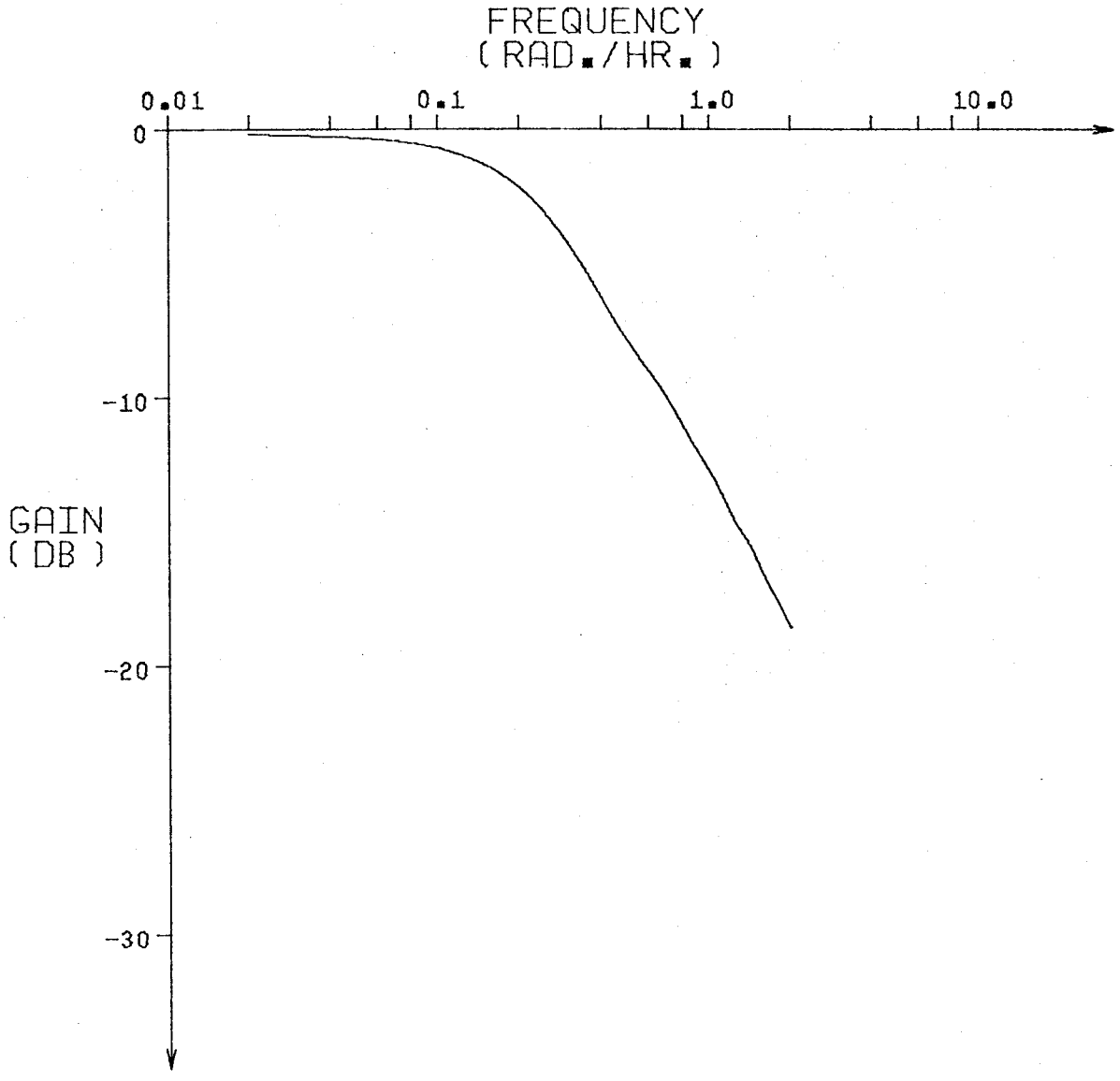


FIGURE (5.1) BODÉ GAIN PLOT FOR RUN #61

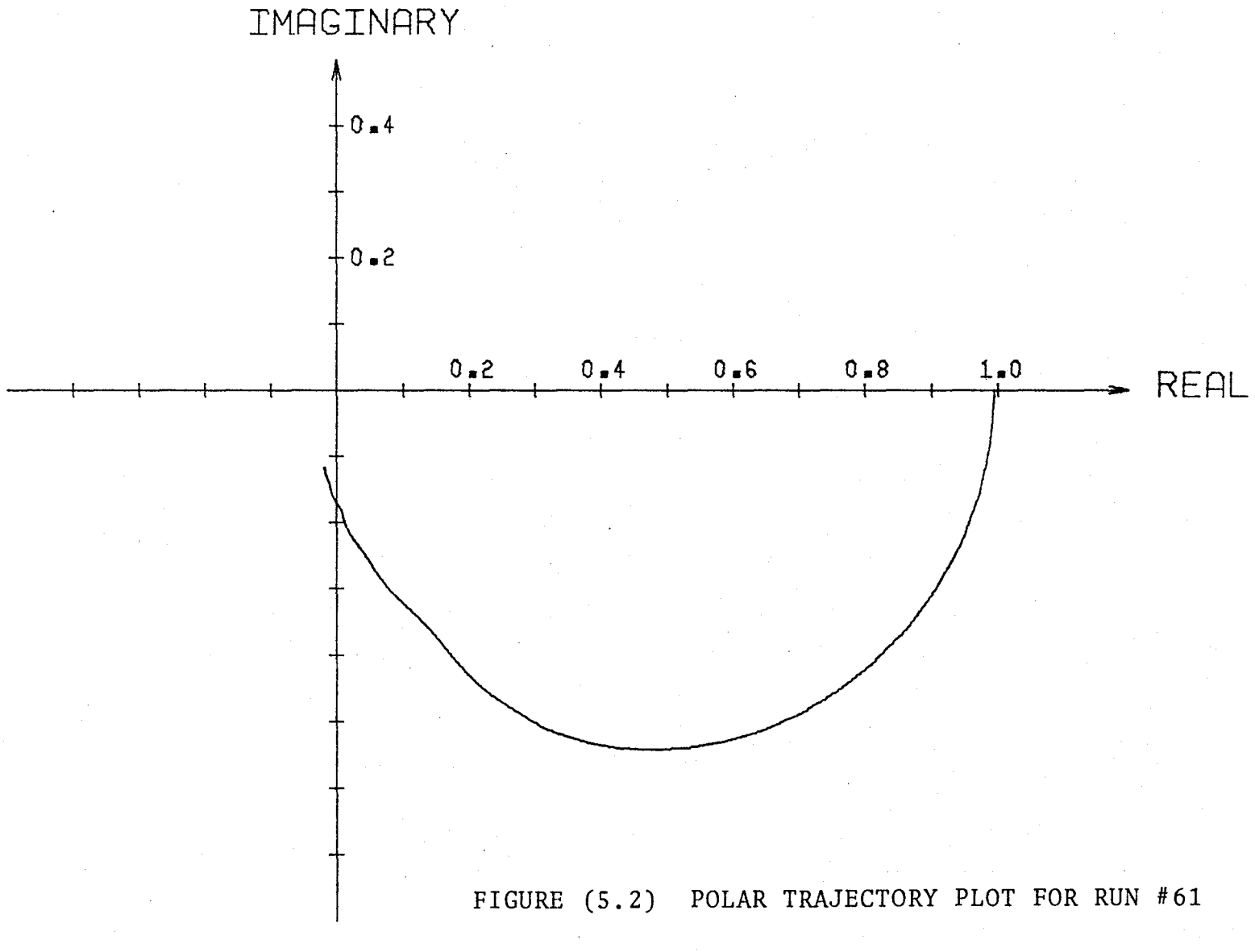


FIGURE (5.2) POLAR TRAJECTORY PLOT FOR RUN #61

that the category 2 models were indeed the correct choice.

The models for categories 1 and 2 are illustrated in Appendix C. The other two categories, 3 and 4 can be found in Wilson's thesis.

5.1.2d Parameter Estimation

Since the mathematical models used are non-linear in the parameters, a non-linear least squares technique must be used for parameter estimation. An unconstrained Simplex search of the type proposed by Nelder and Mead (N-2) was employed. This type of optimization routine was recommended by Wilson since it consumes less computer time than the Rosenbrock search which he used.

One four parameter model was fitted, but only a slight change in the residual sum of squares (RSS) was noted over that of the three parameter model. Therefore, since they require much more computer time, the four parameter models were not used further.

The values for the initial parameter estimates fed to the computer were chosen randomly. After the program had run, any models, whose final estimates of the parameters showed that any component residence times were negative or that the flow split was not between 0 and 1, were rejected as being inherently poor fits to the data because of their mathematical structure. Then the model(s) with the lowest RSS were run again with three different sets of initial estimates. This was done to discover if the search routine would return to the same minimum.

In most cases it did return to the same minimum.

The Fourier transform program produces 101 data points, but only 26 of these, evenly spaced, were used to estimate the parameters. This again follows the recommendation of Wilson who noted that 15-30 points are sufficient for parameter estimation. All 101 data points were used, however, in the calculation of the RSS.

For comparison purposes, the dispersion model, designated as model 9990, with the parameters estimated by Timpany, using the peak time correlation method, are shown, in Figure 5.3, along with the models resulting from category 1 and 2 selections. As can be seen the PFTR-CSTR component models are a better fit than the dispersion model in the frequency domain. Also the model from category 1, 3240, is almost identical to the model from category 2, 1460. (See Appendix C pp. 107 & 113)

Finally, the total residence time of each model was calculated using the component residence times and the flow splits. All of these values were found to be between the theoretical residence time of the prototype and the mean residence time calculated from the tracer output curve. A summary of the parameter estimates can be found in Table 5.4, while listings of the computer programs used can be found in Appendix D.

5.1.2c Time Response

In order to compare the time response of the chosen models with the original time domain data, the

TABLE (5.4)

BEST PARAMETER ESTIMATES FOR

RUN #61

Run No.	Mean Res. Time	Model No.	Category	RSS X10^4	τ_T	τ_1	τ_2	Q_1/Q_T	$D/_{uL}$
61	4.08	1460	N/N+2	31.2	4.19	2.29	0.204	0.634	--
61	4.08	3240	N/N+1	52.3	4.20	2.31	0.182	0.618	--
61	4.08	9990	DISP	344	4.16	--	--	--	4.99

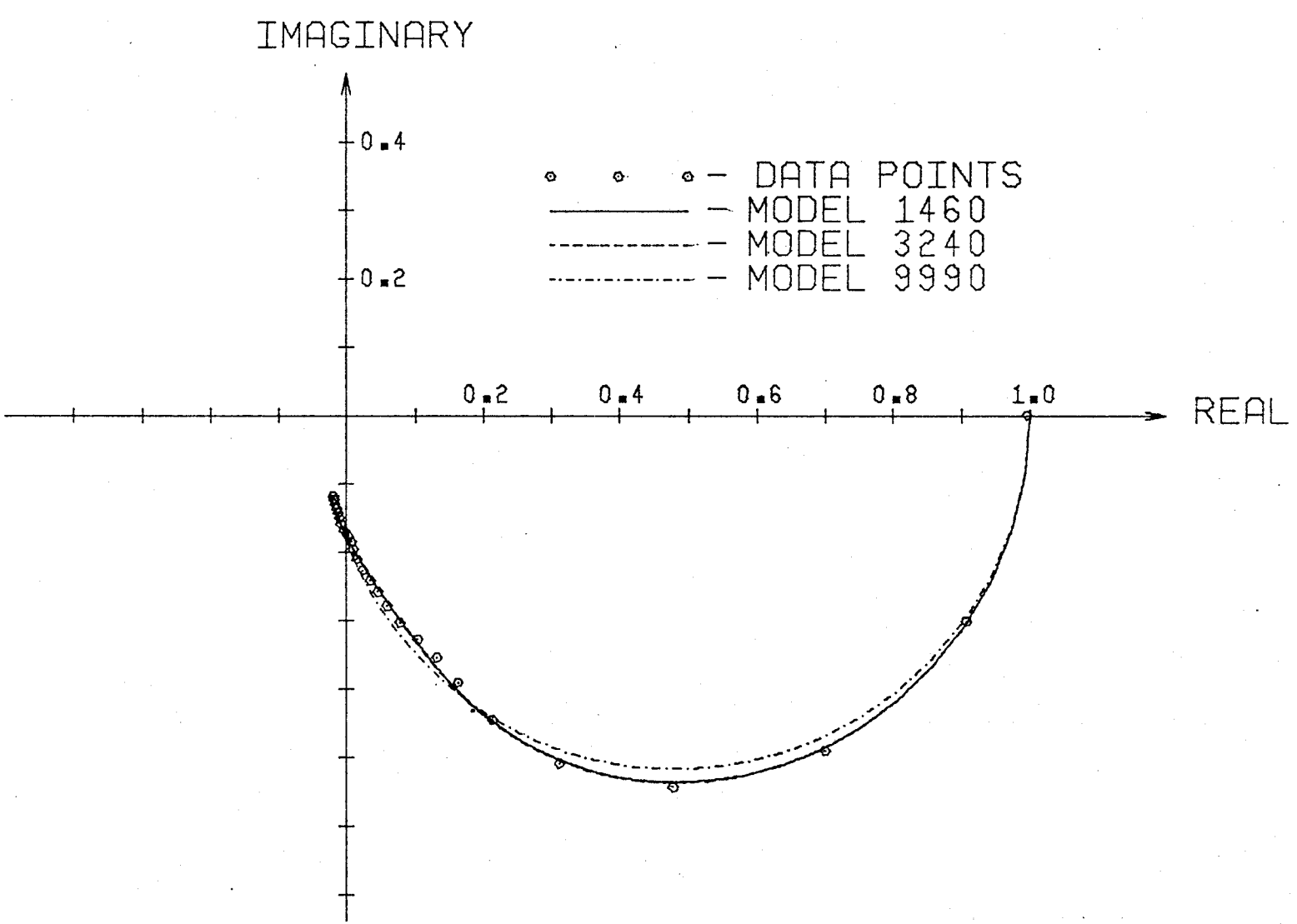


FIGURE (5.3) MIXING MODEL FITS IN THE FREQUENCY DOMAIN FOR RUN #61

models were inverse Laplace transformed. Then the final parameter estimates were used to produce the model time response. Figure 5.4 shows the plotted results. As can be seen model 3240 is a very poor fit to the data. This model is from category 1, the category which was selected by the use of Wilson's criteria. Model 1460 from category 2 is a much better fit despite the fact that it and model 3240 were indistinguishable in the frequency domain. Model 9990 Timpany's dispersion model seems to be a better fit before the peak time but is not very good after that.

The listing for the program which performed the time response calculations is listed in Appendix E.

5.1.3 Parameter Estimation for Kinetic Model

The data used for modelling purposes was one run of batch biokinetic data from Pollock (P-1). However, only the concentration of soluble carbon versus time was modelled; the change in the mass of microbial solids was neglected entirely. Pollock used polynomials and piecewise linearizations to characterize his data. While the polynomials fit well in the time range of the experiment, they do not tend to be very satisfactory outside of that range. That is, they tend to either plus or minus infinity as time goes to infinity, rather than remaining at a steady value as observed in a batch reaction after a very long time. As shown in Chapter 2 the Edwards model gives a better fit to the data.

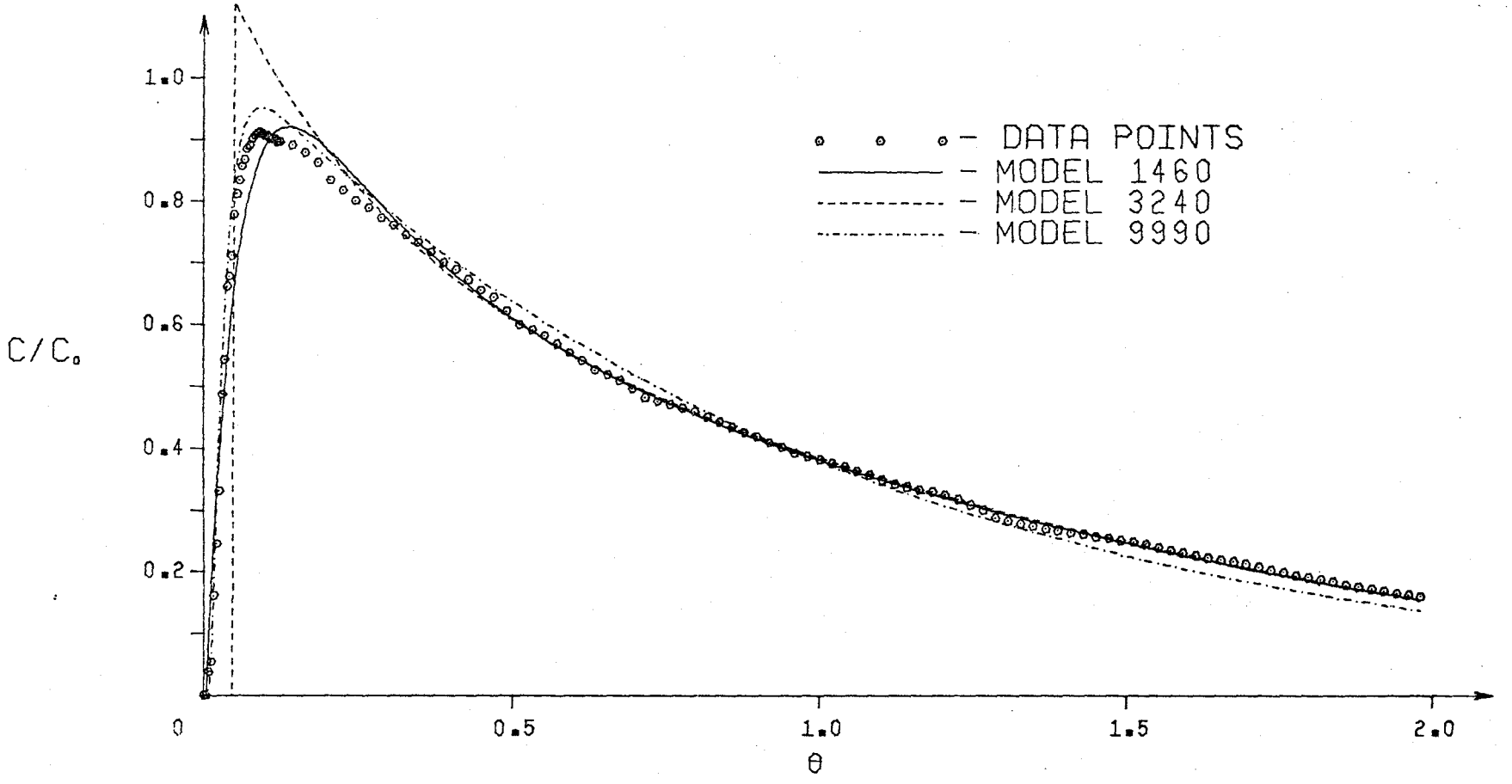


FIGURE (5.4) TIME RESPONSE OF MIXING MODELS FOR RUN #61

The Edwards model,

$$C_C = \frac{K}{1 + \exp\,(a_o + a_1 + a_2 t^2 + \ldots + a_n t^n)}$$

where C_C is the concentration of soluble carbon

t is time

K, a_o, a_1, ..., a_n are empirical parameters

n is an integer

is non-linear in the parameters and therefore was fitted using the Simplex search technique. The data was fitted for n = 1, 2, 3, 4 (i.e. 3, 4, 5, 6 parameters).

After some preliminary investigation it was found that the best fit of the model was obtained when the steady state value of soluble carbon concentration was removed from all data points by subtraction. Since the steady-state value represents an unremovable residual, it was also left out in the conversion predictions. Thus the conversions represent the percent removed based on the removable carbon. Not surprisingly, the six parameter model was the best fit to the data. By eye it was a much better than the three, four or five parameter models. Thus it was chosen to represent the most accurate kinetic model.

A piecewise linear discontinuous model was also employed. Here one straight line was fitted by a linear least squares technique to the down-sloping segment of the data. Another horizontal straight line was used for the steady-state portion of the curve. The residual sum of squares for the straight line curves were much higher than those for any of the Edwards models.

Figure 5.5 shows the fit of the six parameter model as well as the linear model. Table 5.5 shows the final values of the fitted parameters for both models. The computer program used for the fitting is listed in Appendix F.

TABLE (5.5)

BEST PARAMETERS FOR BATCH KINETIC RUN

Model	Best Fitted Equation
Edwards Six Parameter	$C_C = \frac{285}{1+\exp(0.44+1.55t-0.032t^2-0.31t^3+0.45t^4)}$
Piecewise Linear	$C_C = -82.1t + 132$ for $0 \leq t \leq 1.61$ $C_C = 18$ for $t > 1.61$

5.1.4 Conversion Predictions

5.1.4a Conversion for different model combinations.

The equation used for conversion prediction assumes that the degree of segregation, J, is equal to one. This allows the use of equation (5.1). As shown in the previous sections the mathematical models for $\frac{(C)}{(C_o)}_{batch}$ and E (t) have been fitted to the data and are thus known. However, an analytic integration in equation (5.1) is not possible and hence a numerical technique was employed. The trapezodial rule was considered to be sufficiently precise for the calculation.

As mentioned previously, four mixing models and two kinetic models were available. The PFTR model will henceforth be designated 100 and the CSTR as 101 for identification purposes only. By means of varying the total residence times of each mixing model, curves of percent conversion vs the ratio of the actual residence

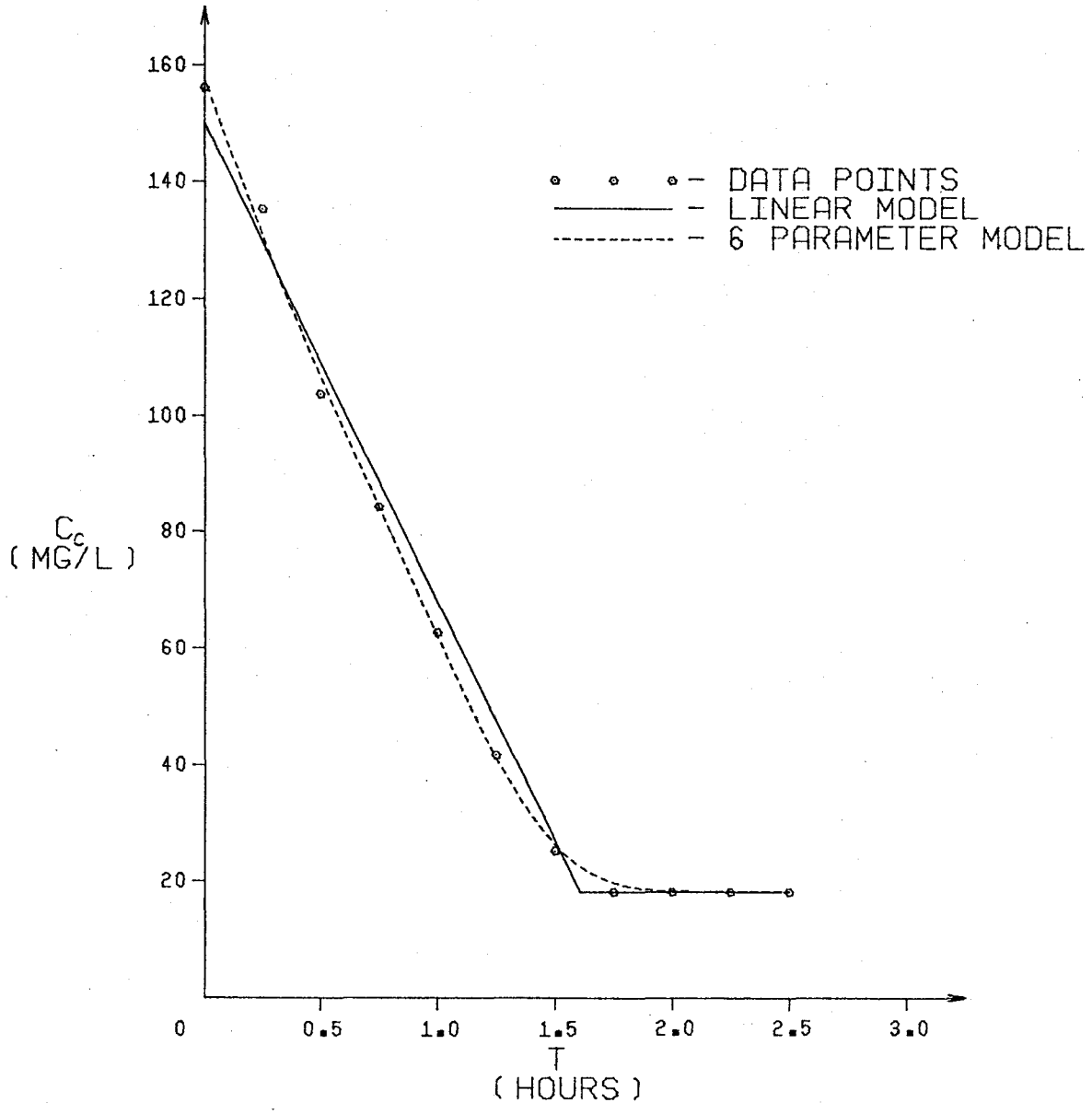


FIGURE (5.5) KINETIC MODEL FITS FOR THE BATCH KINETIC RUN

time, τ, to the nominal residence time, τ_N, were obtained. Each mixing model was in turn combined with each kinetic model. The results are shown in Figure 5.6 and 5.7. As can be seen the percent conversions for models 100 and 101 form an "envelope" within which the conversions for all other mixing models must lie. The computer program used for the calculations is listed in Appendix G.

5.1.4b Percentage Differences in Conversion

In order to assess which factor was more important, the mixing model or the kinetic model, percentage differences between the best fit models and all of the other models were calculated. The models which fit the data best were the 1460 mixing model and the six parameter kinetic model. The plotted results are shown in Figures 5.8 and 5.9. From the graphs it can be seen that there is negligible difference between models 3240 and 1460 in the range $0.5 \leq \tau/\tau_N \leq 2.0$. If this ratio could be considered to be proportional to the flow rate, then for practical applications it would not matter which mixing model of the two was chosen. This results from the fact that the sewage flow to most biological treatment units varies from about one half to twice the average flow rate. Also in this range it is to be noted that there is little difference between the predictions of the six parameter model and the linear model for the biokinetics. However, the differences are definitely more pronounced if it had been assumed that the reactor was a CSTR or a PFTR, as is often done in the Environmental Engineering literature. Thus

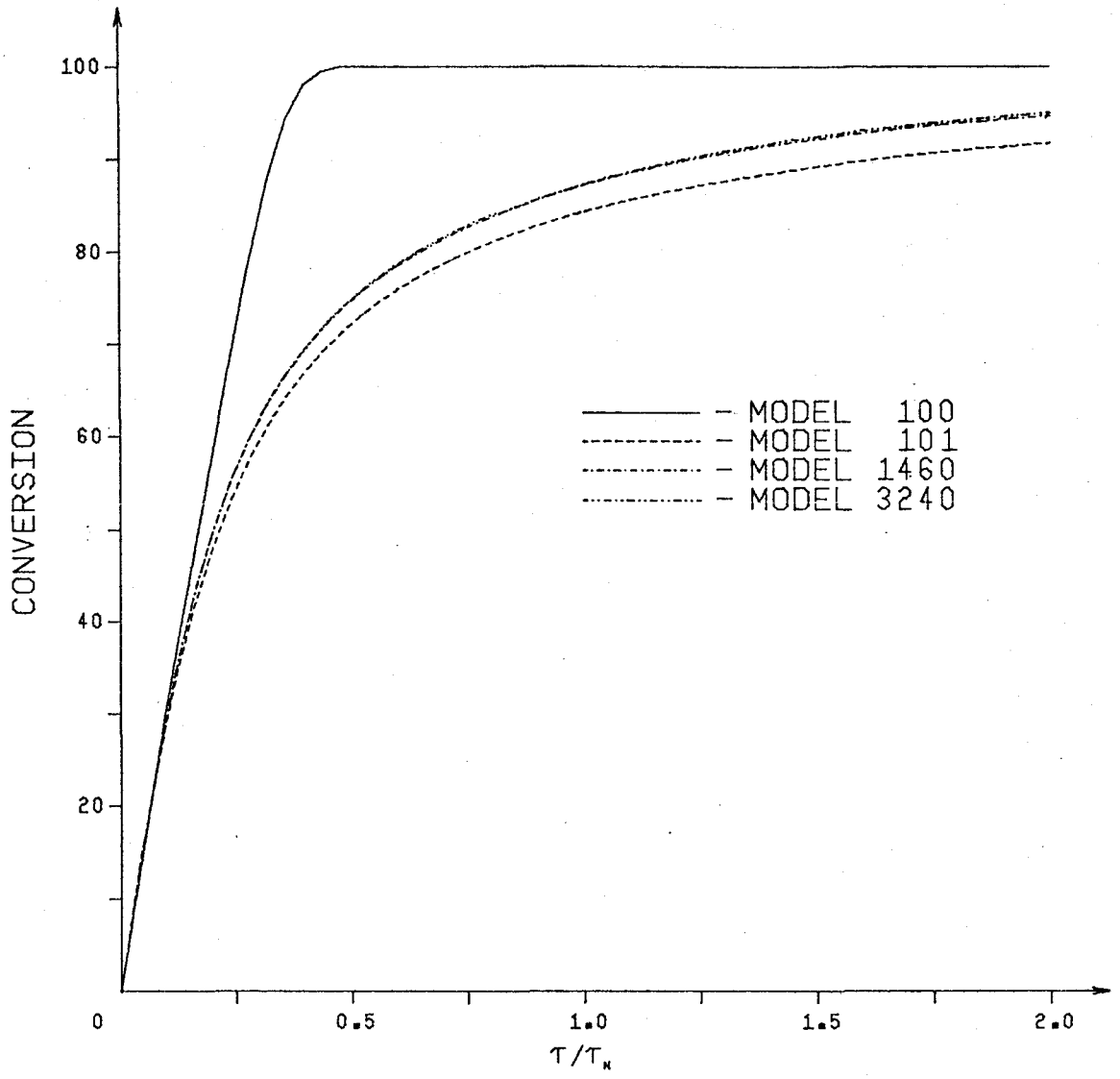


FIGURE (5.6) CONVERSION PREDICTIONS USING THE SIX PARAMETER EDWARDS KINETIC MODEL

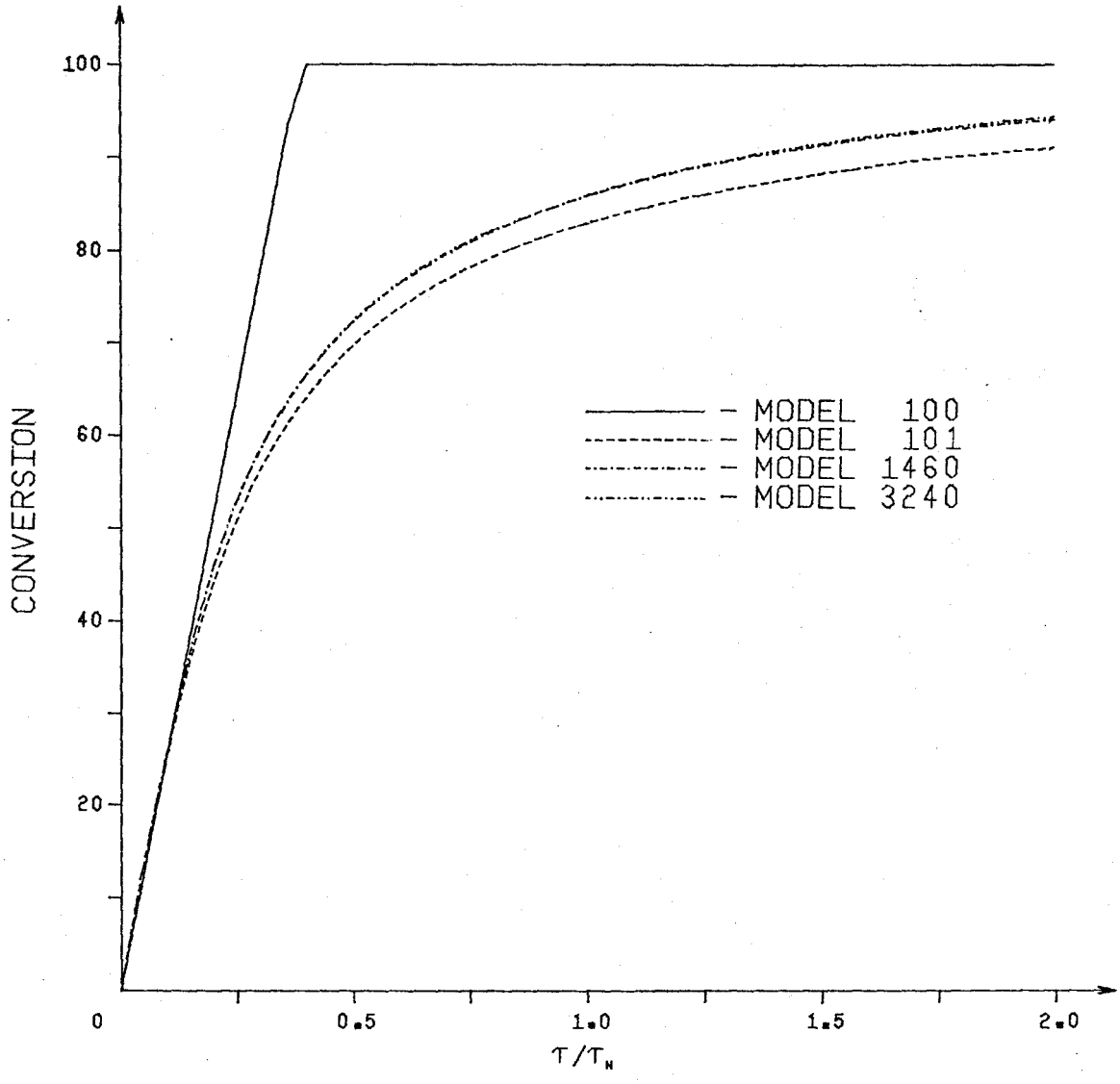


FIGURE (5.7) CONVERSION PREDICTIONS USING THE PIECEWISE LINEAR KINETIC MODEL

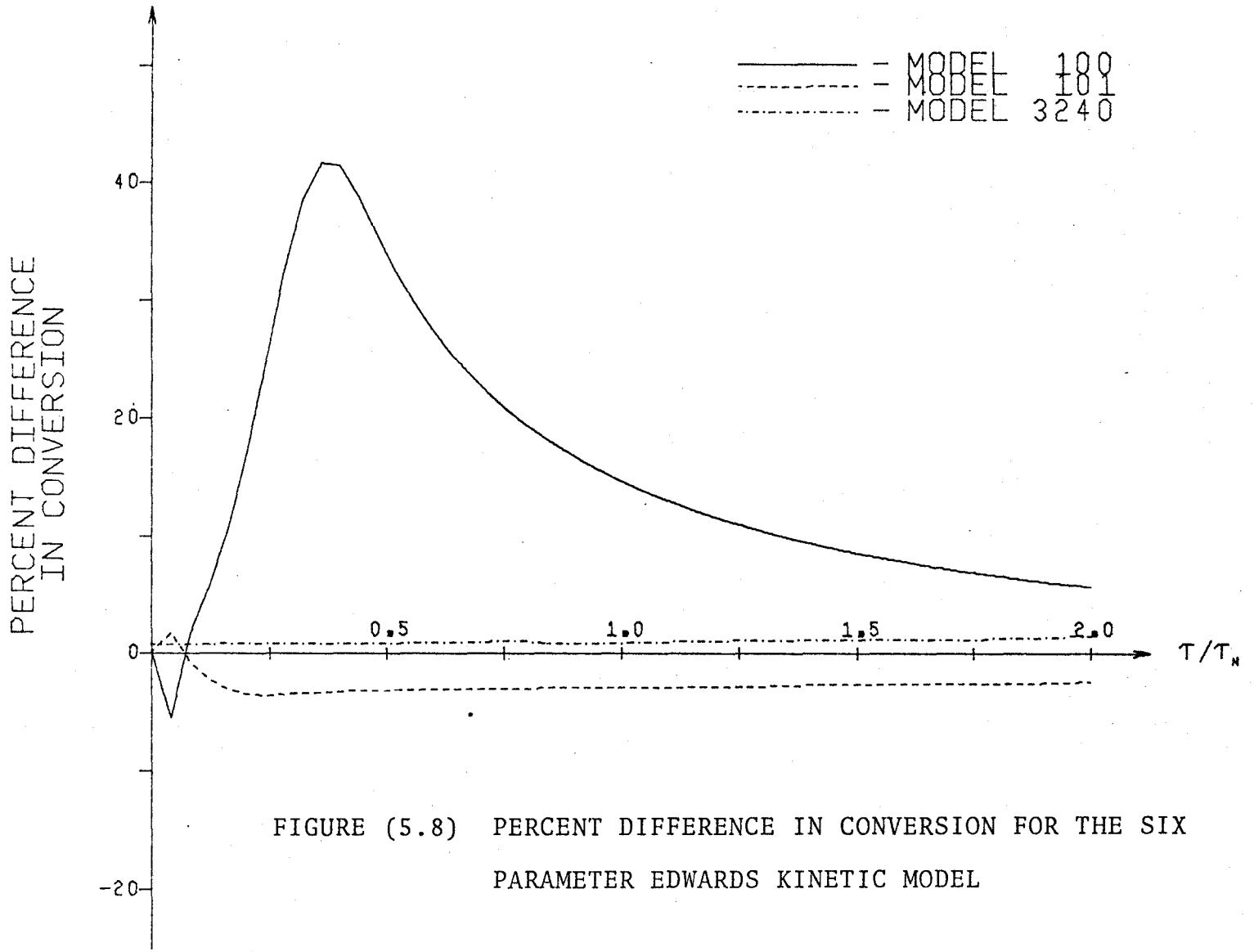


FIGURE (5.8) PERCENT DIFFERENCE IN CONVERSION FOR THE SIX PARAMETER EDWARDS KINETIC MODEL

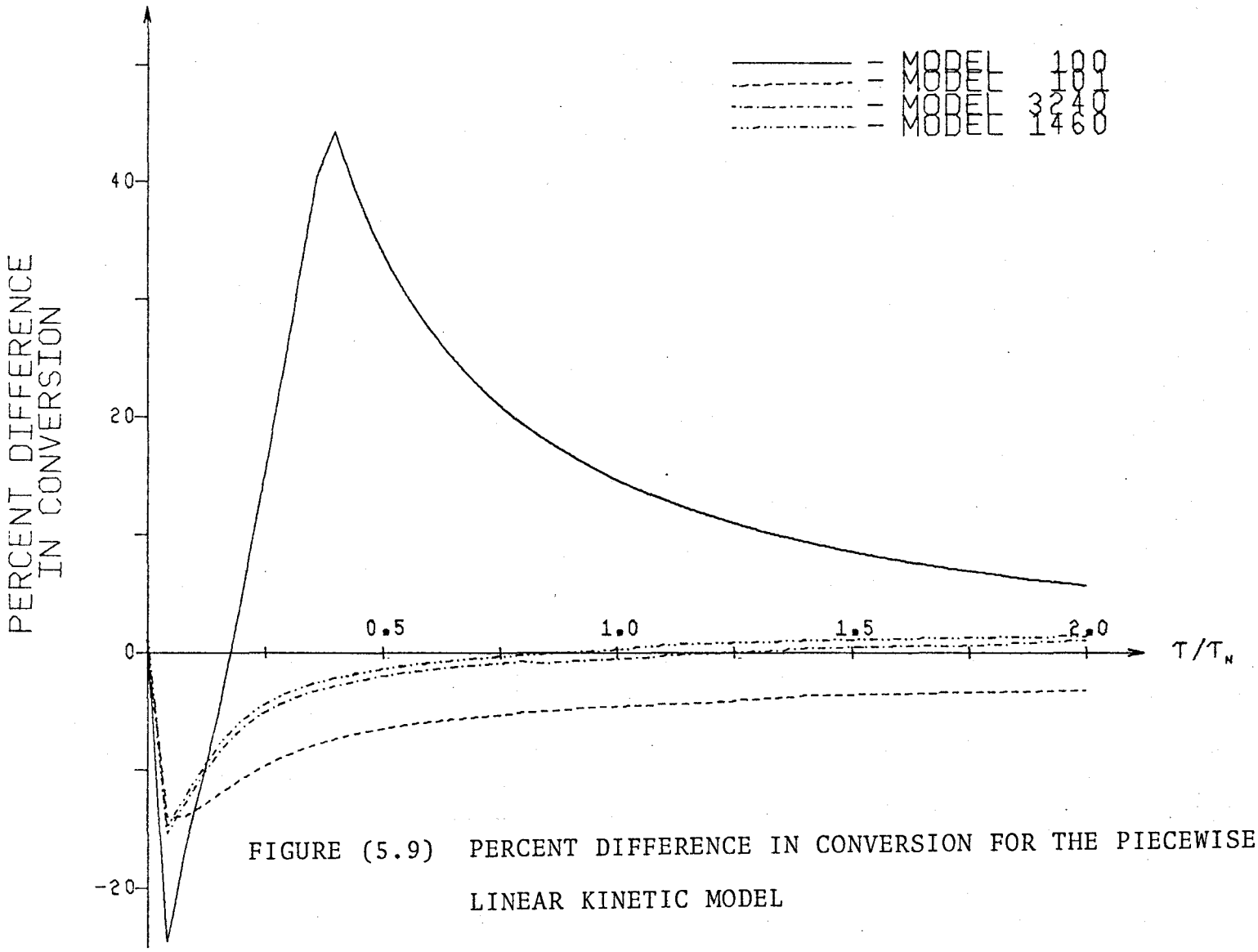


FIGURE (5.9) PERCENT DIFFERENCE IN CONVERSION FOR THE PIECEWISE LINEAR KINETIC MODEL

for the data employed in this study, it appears that the modelling of the mixing phenomena is more critical than the modelling of the biokinetic phenomena with respect to conversion prediction. This is not to say that this would hold true for all cases of biological reactors and reactions. The program listing appears in Appendix H.

5.2 Correlation of Model Parameters to Water Flow Rate

5.2.1 Summary of Procedures

As stated in Chapter 1, biological reactors in the water pollution control field are subjected to unsteady-state flows. In order to predict how the conversion potential of the reactor will be affected by this flow variation, the response of the mixing models, for a given reactor geometry and air flow, to the change in flow must be known.

The same techniques for mixing model fitting as described in previous sections were used. This time, however, they were applied to the four data runs on the laboratory scale reactor made by Timpany. Each run used the same geometry and air flow rate, but different water flow rates. The data on these runs is shown in Table 5.6. After the parameters were evaluated an attempt was made to correlate the parameters to the water flow rate.

5.2.2a Parameter Estimation

As with the prototype data, two of the four runs proved to have the wrong category selected using the

TABLE (5.6)

EXPERIMENTAL DATA FOR

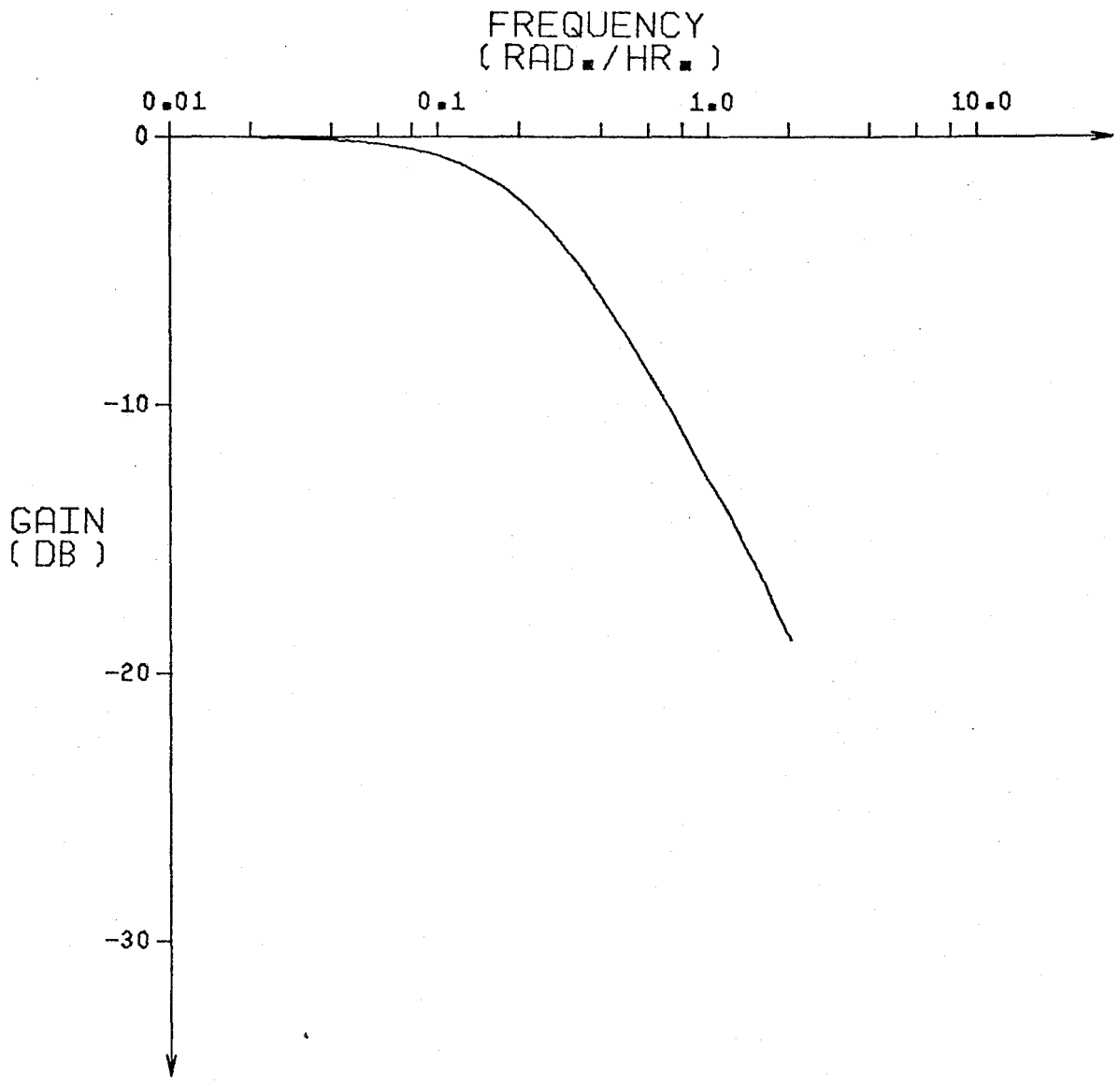
RUNS #16, #18, #19, #20

Run No.	Air Flow (cfm)	Water Flow (USgpm)	Temp. ^{o}C	Tank Length (ft.)	Tank Width (ft.)	Tank Depth (ft.)	Theoretical Det. Time (hr.)
16	0.865	1.06	20.0	5.0	3.0	2.51	4.44
18	0.865	0.525	20.0	5.0	3.0	2.52	8.98
19	0.865	5.04	20.0	5.0	3.0	2.52	0.935
20	0.865	2.27	20.0	5.0	3.0	2.52	2.08

criteria set forth by Wilson. Figures 5.10, 5.11, 5.12 and 5.13 show the Bodé magnitude plots for the four runs. They all have an asymptotic high frequency slope - 20 db/ decade. However, from the polar plots Figures 5.14, 5.15, 5.16 and 5.17, runs #16 and #18 have their high frequency trajectory asymptotic to the negative real axis while runs #19 and #20 have their high frequency trajectory asymptotic to the negative real axis. Thus while the Wilson criteria would dictate category 1 models for all runs, using the polar plots yields category 1 for runs #16 and #18, and category 2 for runs #19 and #20. As with the prototype run, #61, the category 2 models fit much better than the category 1 models for runs #19 and #20.

Table 5.7 shows the final parameter values and models chosen to represent the data while Figures 5.18, 5.19, 5.20 and 5.21 illustrate graphically the fits obtained in the frequency domain. Again it can be seen that there is no detectable difference by eye between the fit of the category 1 model and the category 2 model.

As with the prototype, the mathematical models in this section were inverse Laplace transformed to yield the residence time distributions. These are plotted in Figures 5.22, 5.23, 5.24 and 5.25. The models are seen to be quite good fits to the data except in the case of the category 1 models fitted to the data from runs #19 and #20.



FIGURE(5.10) BODÉ GAIN PLOT FOR RUN #16

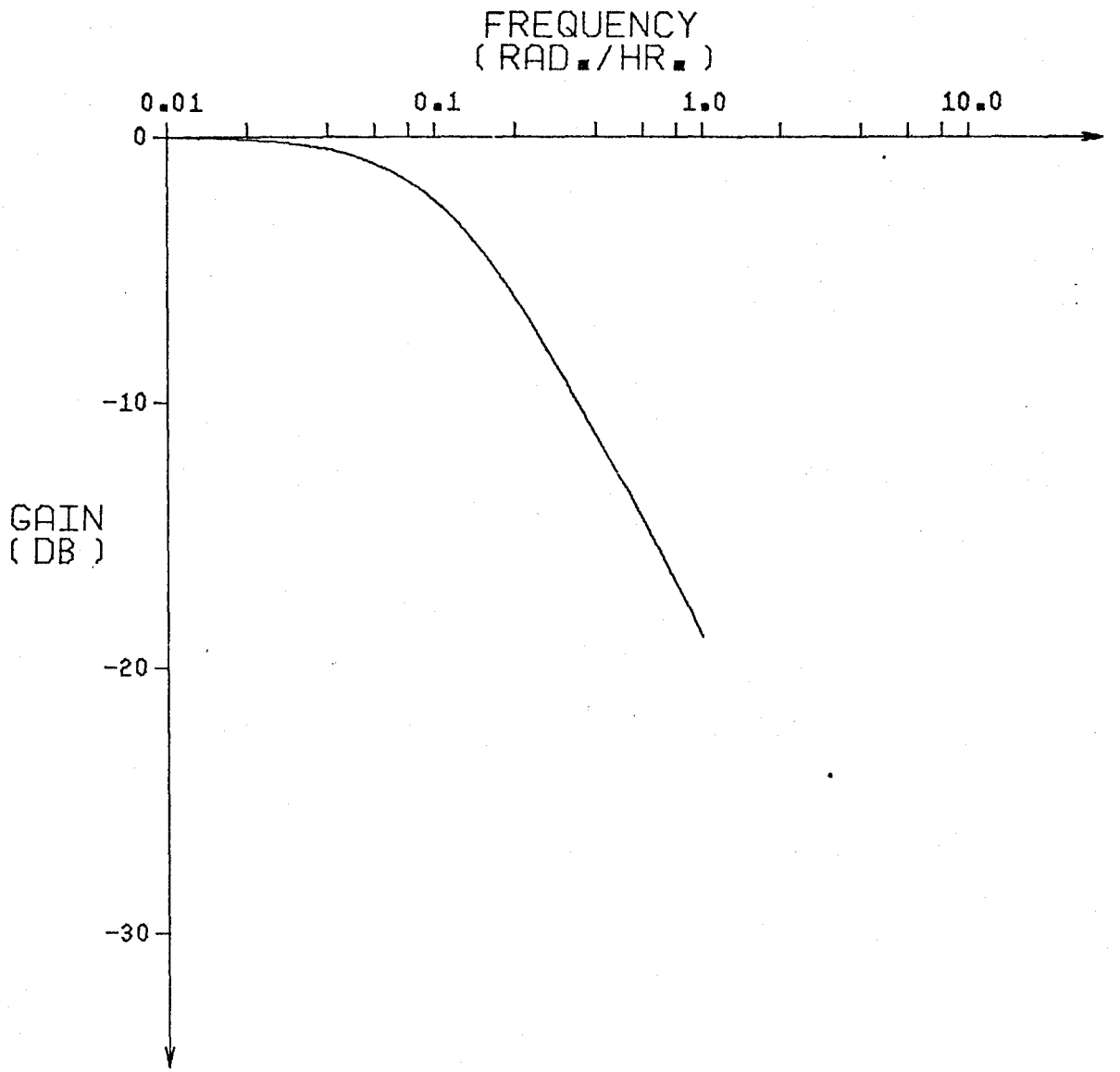


FIGURE (5.11) BODÉ GAIN PLOT FOR RUN #18

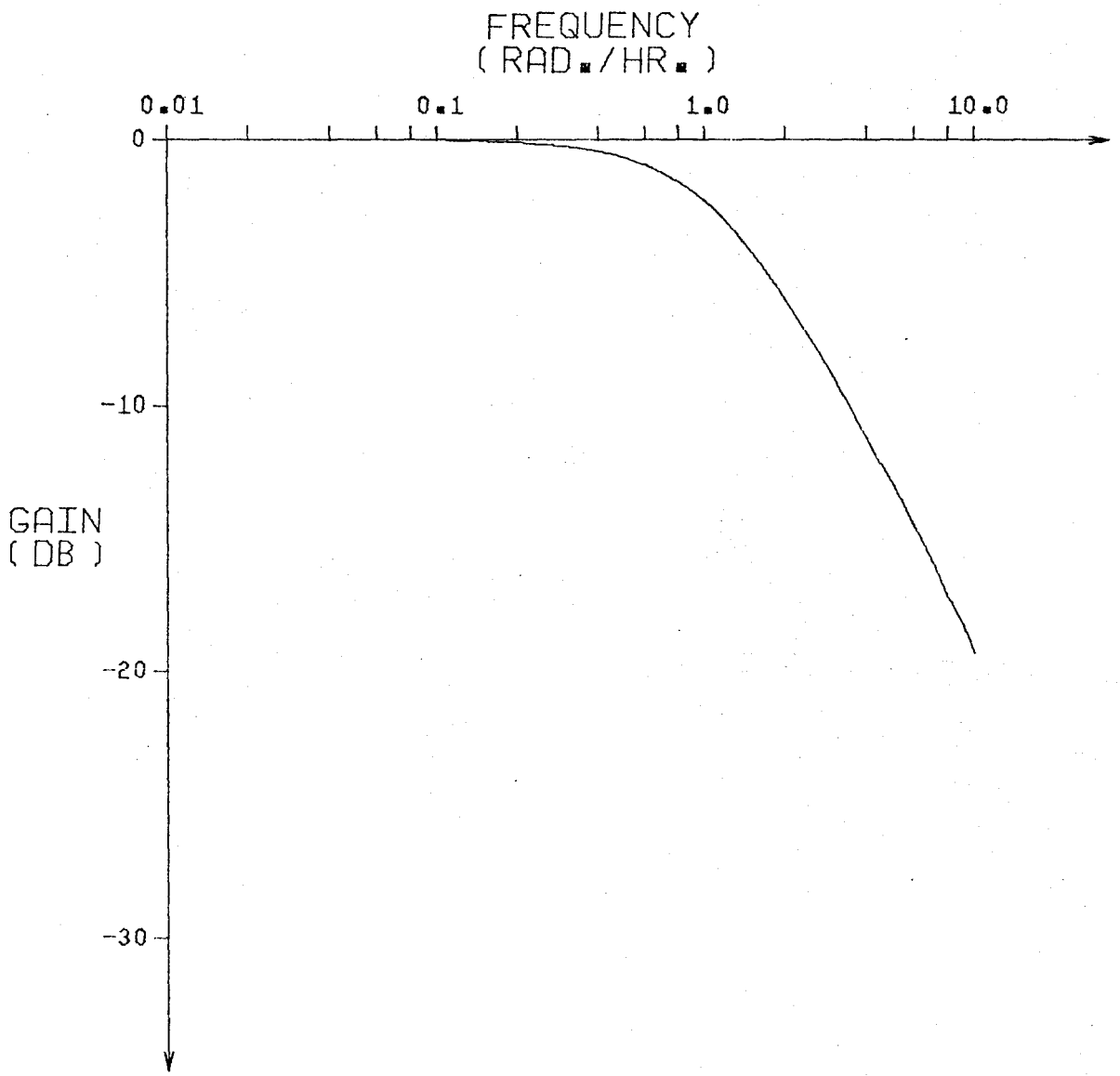


FIGURE (5.12) BODÉ GAIN PLOT FOR RUN #19

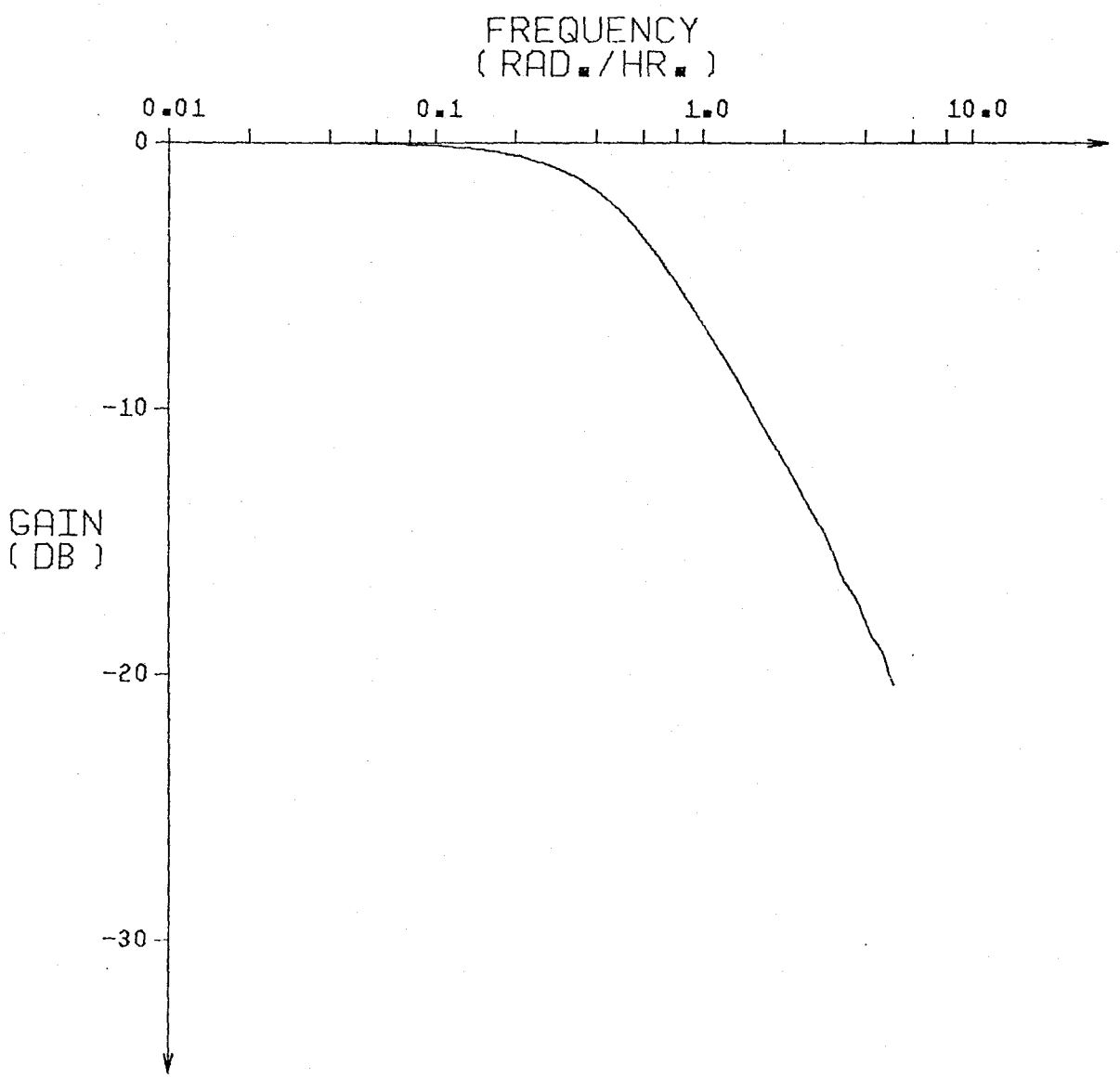


FIGURE (5.13) BODÉ GAIN PLOT FOR RUN #20

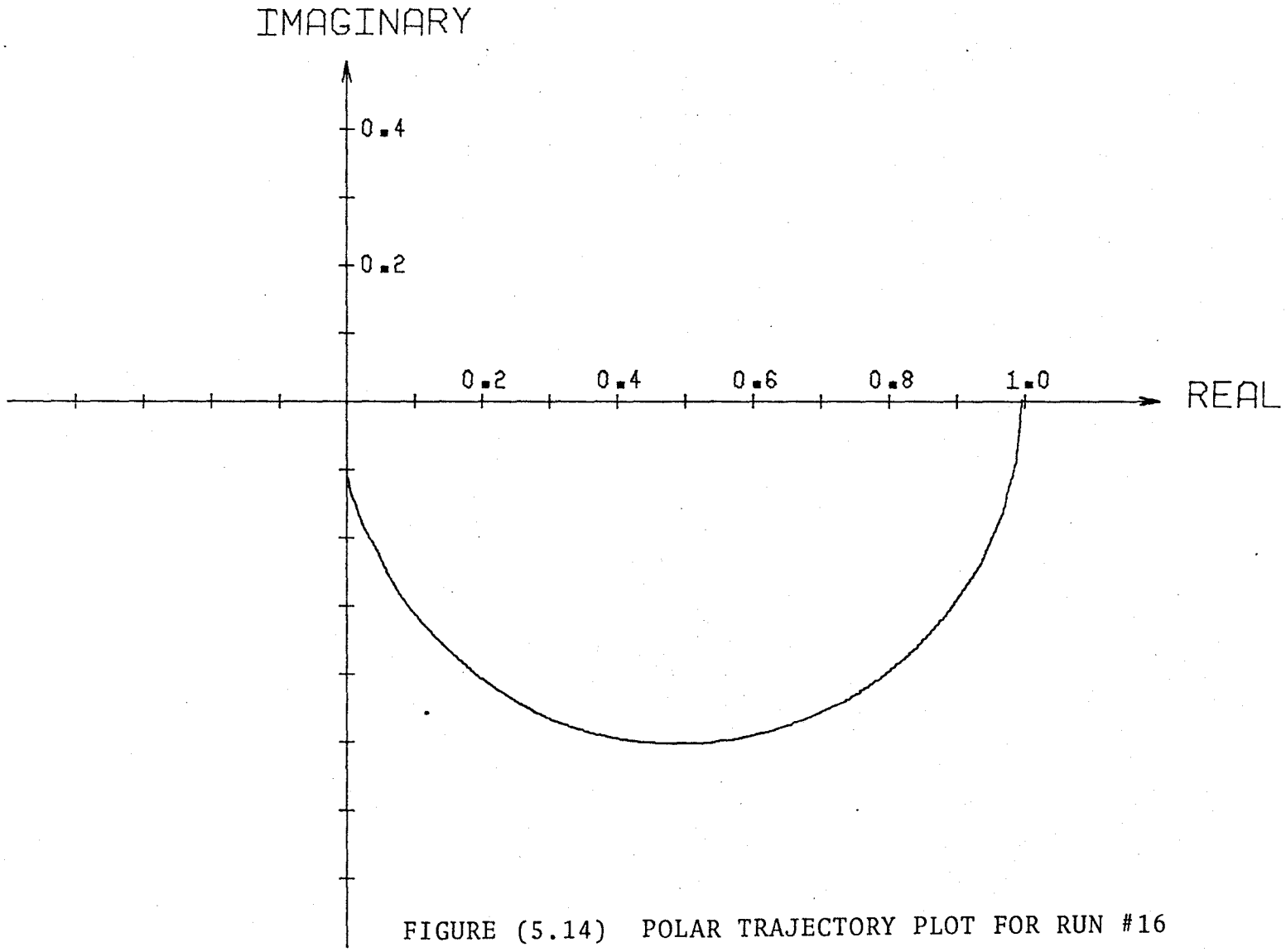


FIGURE (5.14) POLAR TRAJECTORY PLOT FOR RUN #16

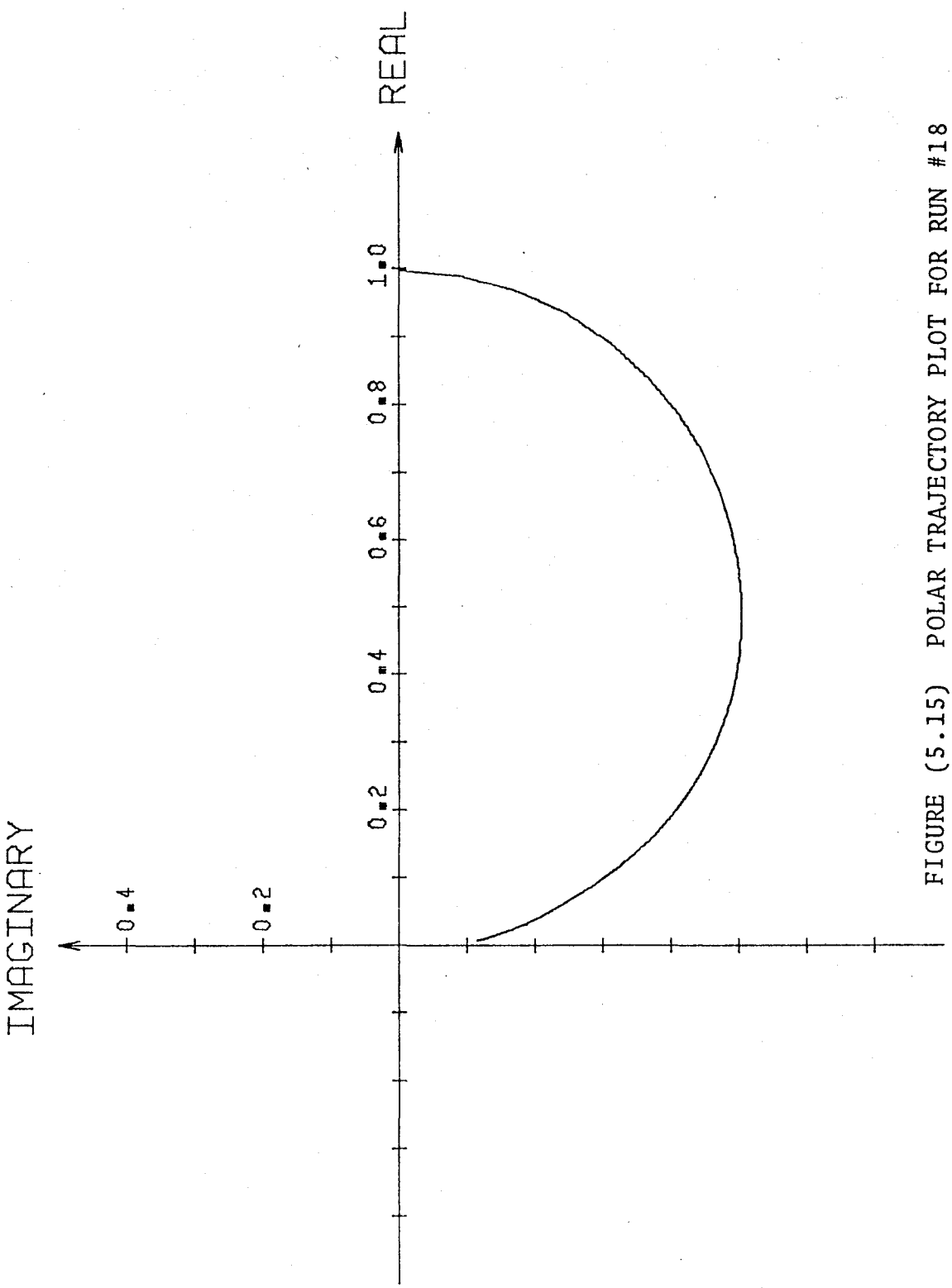


FIGURE (5.15) POLAR TRAJECTORY PLOT FOR RUN #18

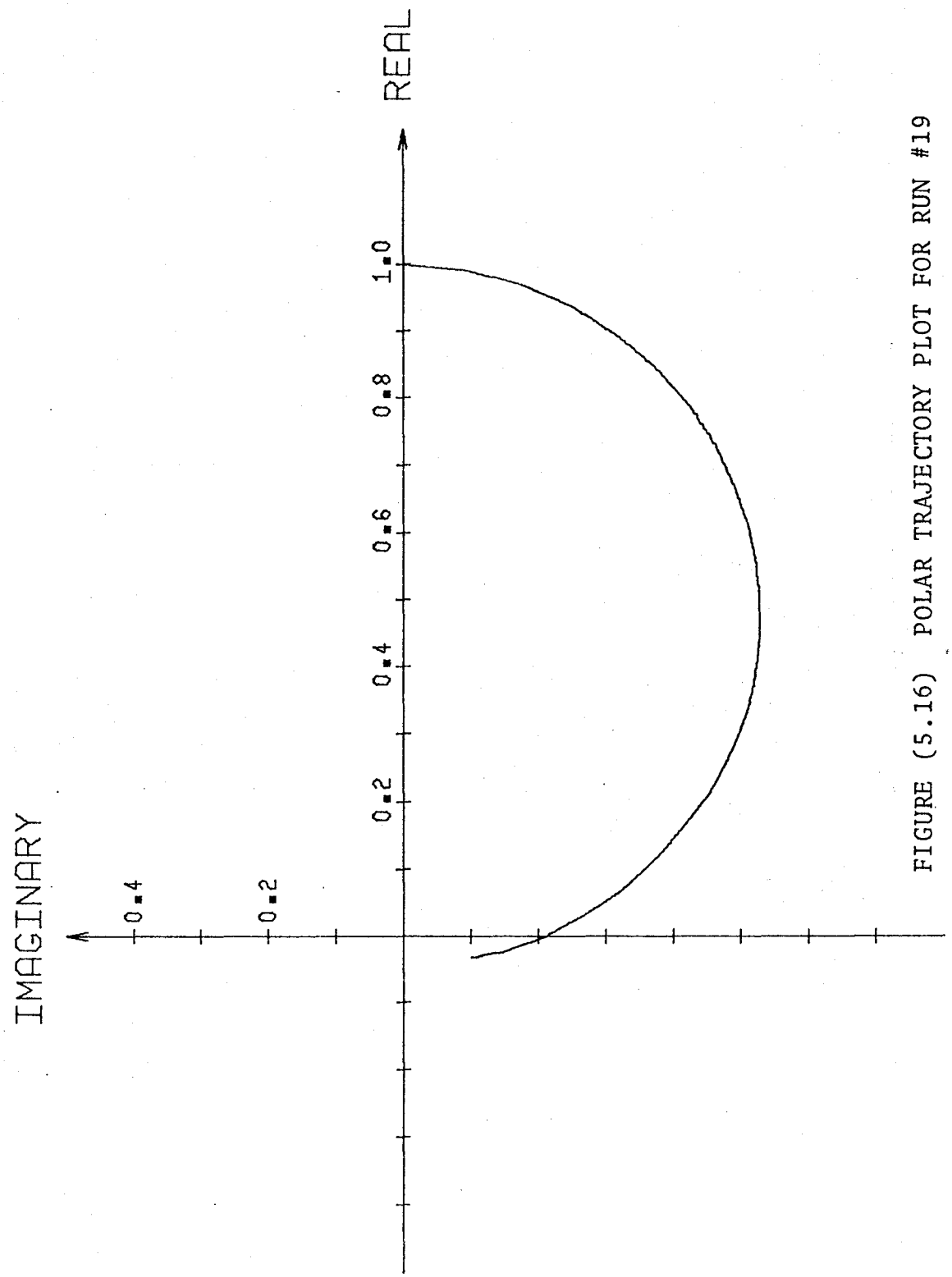


FIGURE (5.16) POLAR TRAJECTORY PLOT FOR RUN #19

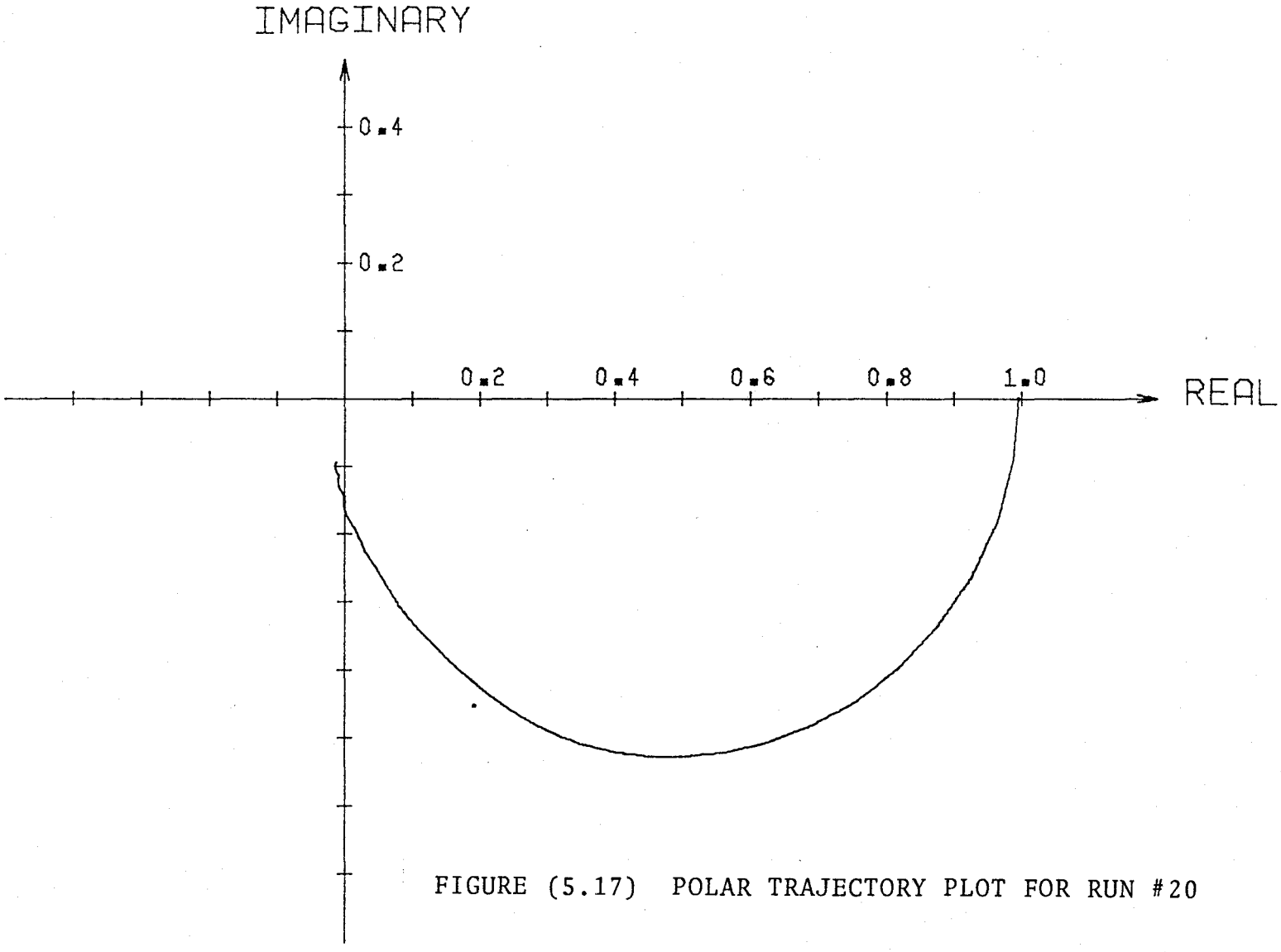


FIGURE (5.17) POLAR TRAJECTORY PLOT FOR RUN #20

TABLE (5.7)

BEST PARAMETER ESTIMATES

FOR

RUNS #16, #18, #19, #20

Run No.	Mean Res. Time	Model No.	Category	RSS X10^4	τ_T	τ_1	τ_2	Q_1/Q_T
16	4.31	3240	N/N+1	5.74	4.39	4.28	0.046	0.995
18	8.70	2730	N/N+1	7.99	8.80	8.17	0.039	0.930
19	0.906	1190	N/N+2	8.62	0.918	0.863	0.022	0.123
19	0.906	2730	N/N+1	8.87	0.912	0.751	0.040	0.847
20	1.91	1490	N/N+2	32.2	1.98	1.61	0.045	0.815
20	1.91	2730	N/N+1	30.5	1.98	1.60	0.044	0.794

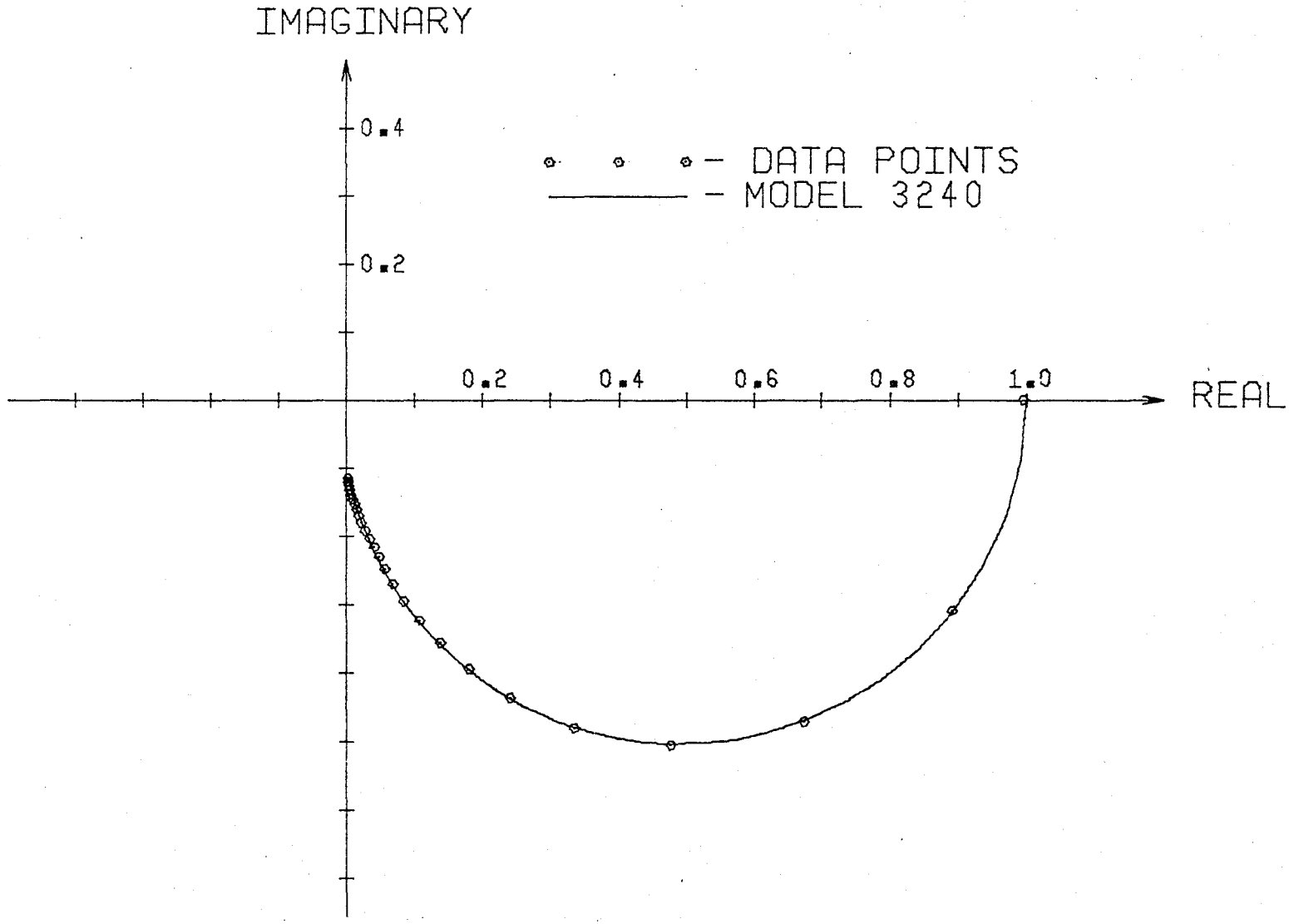


FIGURE (5.18) MIXING MODEL FIT IN THE FREQUENCY DOMAIN FOR RUN #16

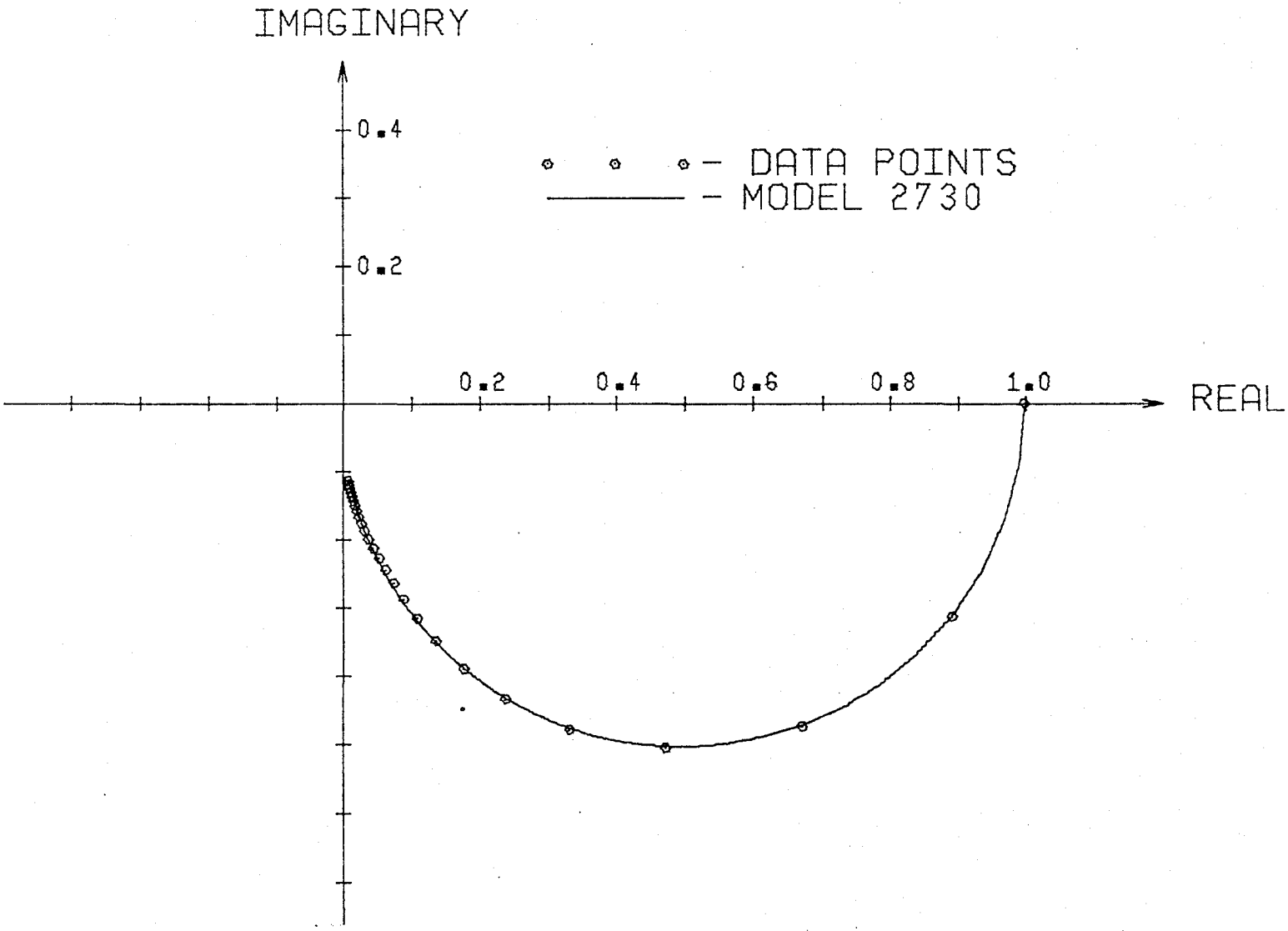


FIGURE (5.19) MIXING MODEL FIT IN THE FREQUENCY DOMAIN FOR RUN #18

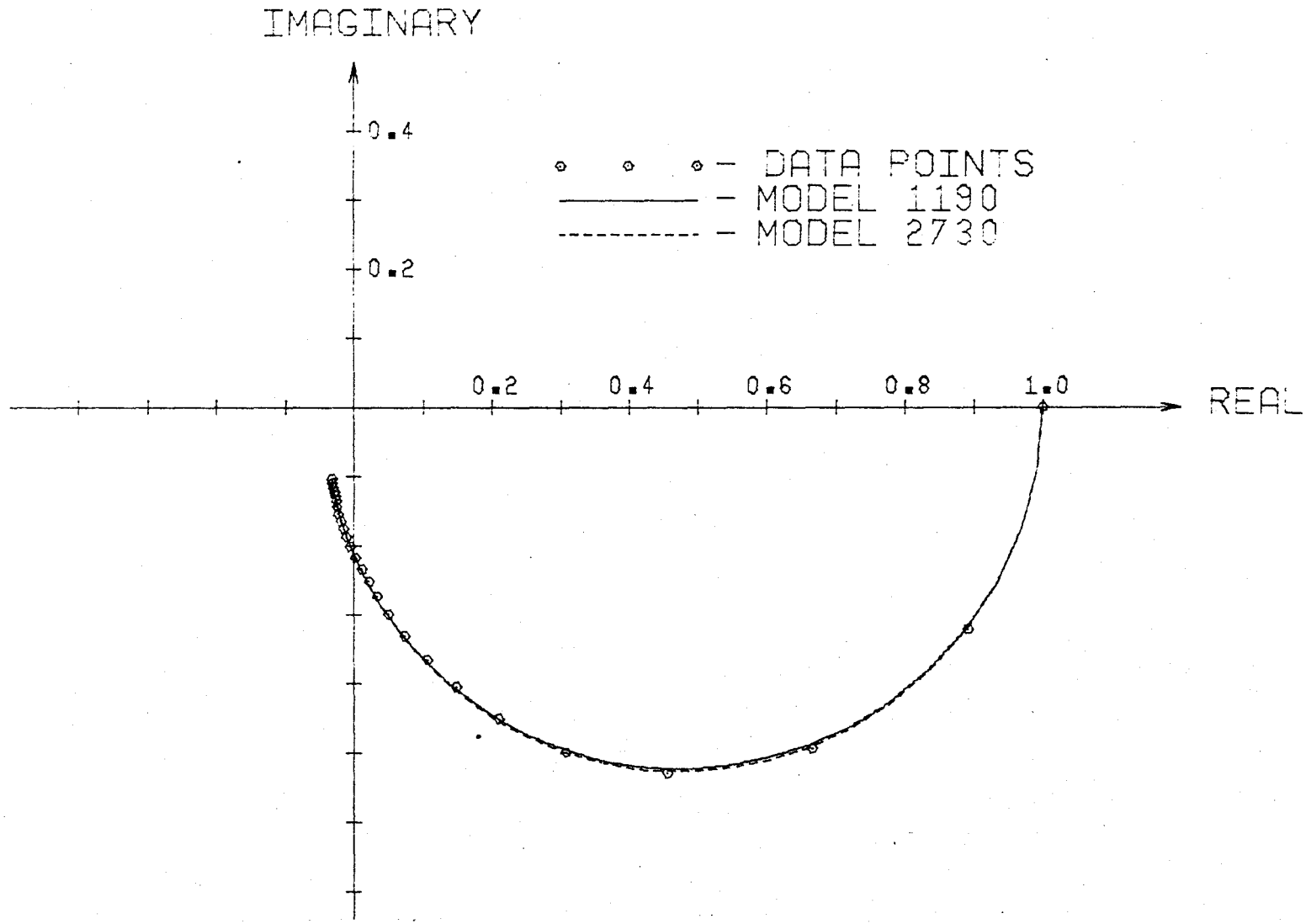


FIGURE (5.20) MIXING MODEL FITS IN THE FREQUENCY DOMAIN FOR RUN #19

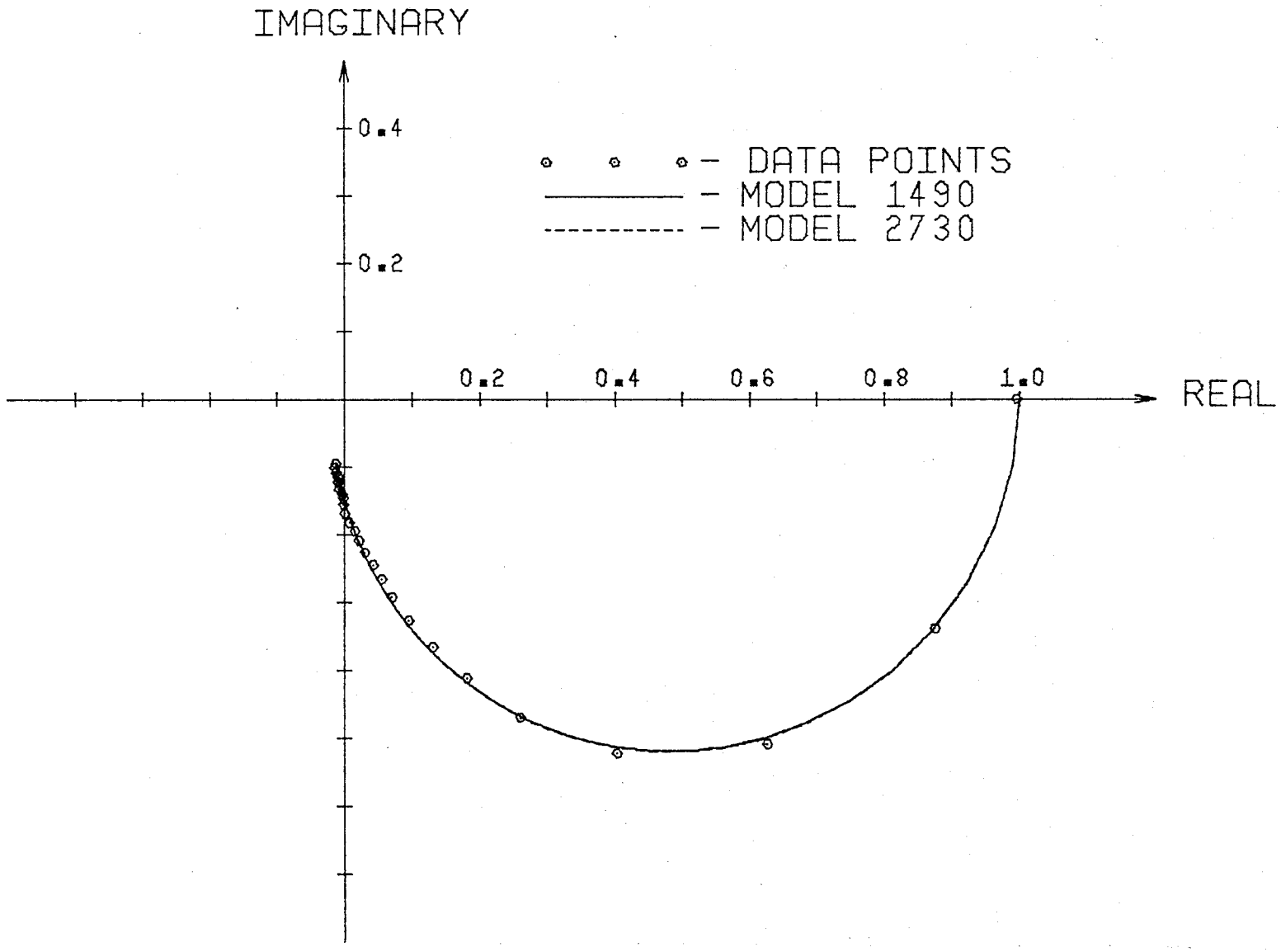


FIGURE (5.21) MIXING MODEL FITS IN THE FREQUENCY DOMAIN FOR RUN #20

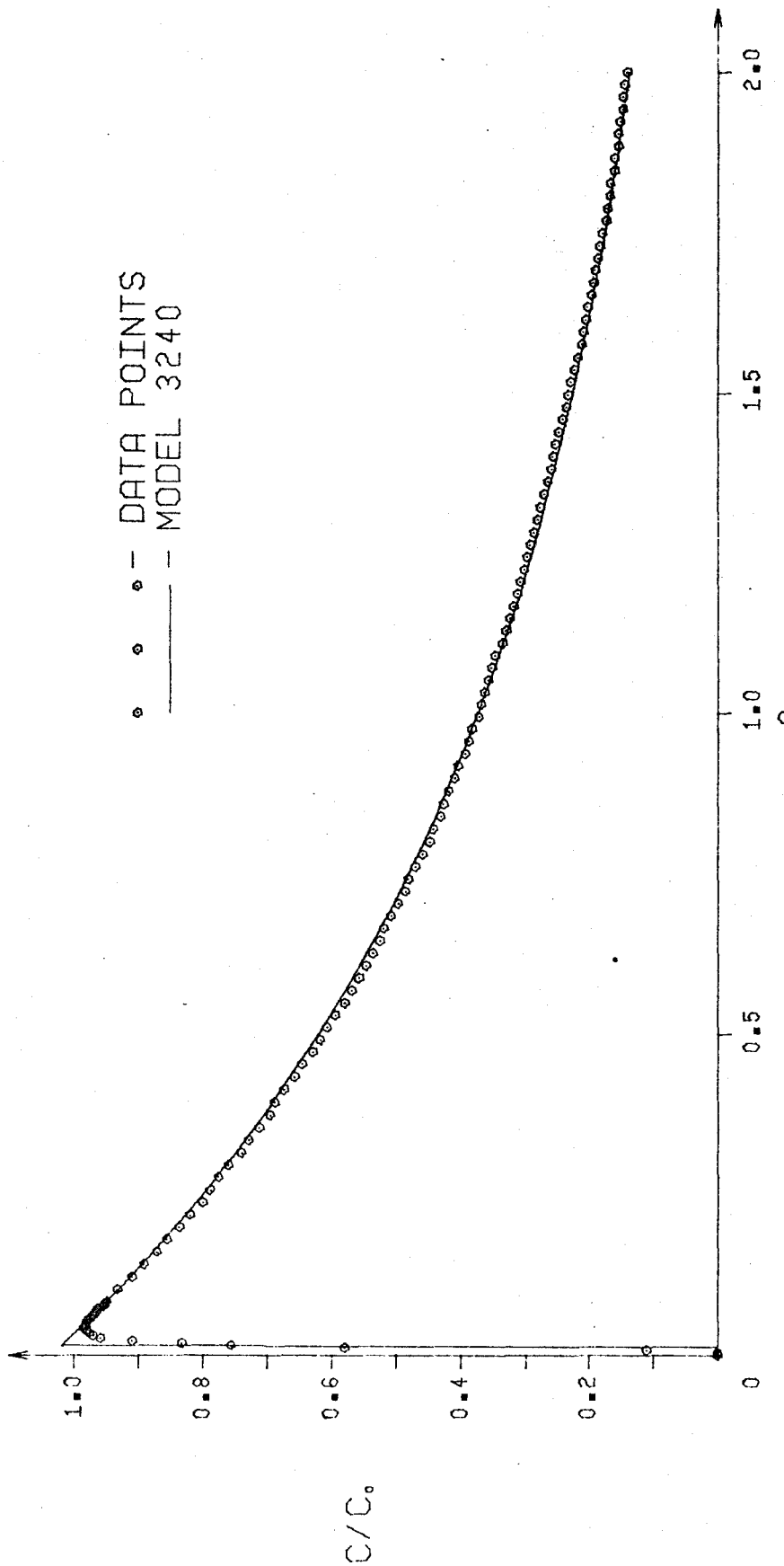


FIGURE (5.22) TIME RESPONSE OF MIXING MODEL FOR RUN #16

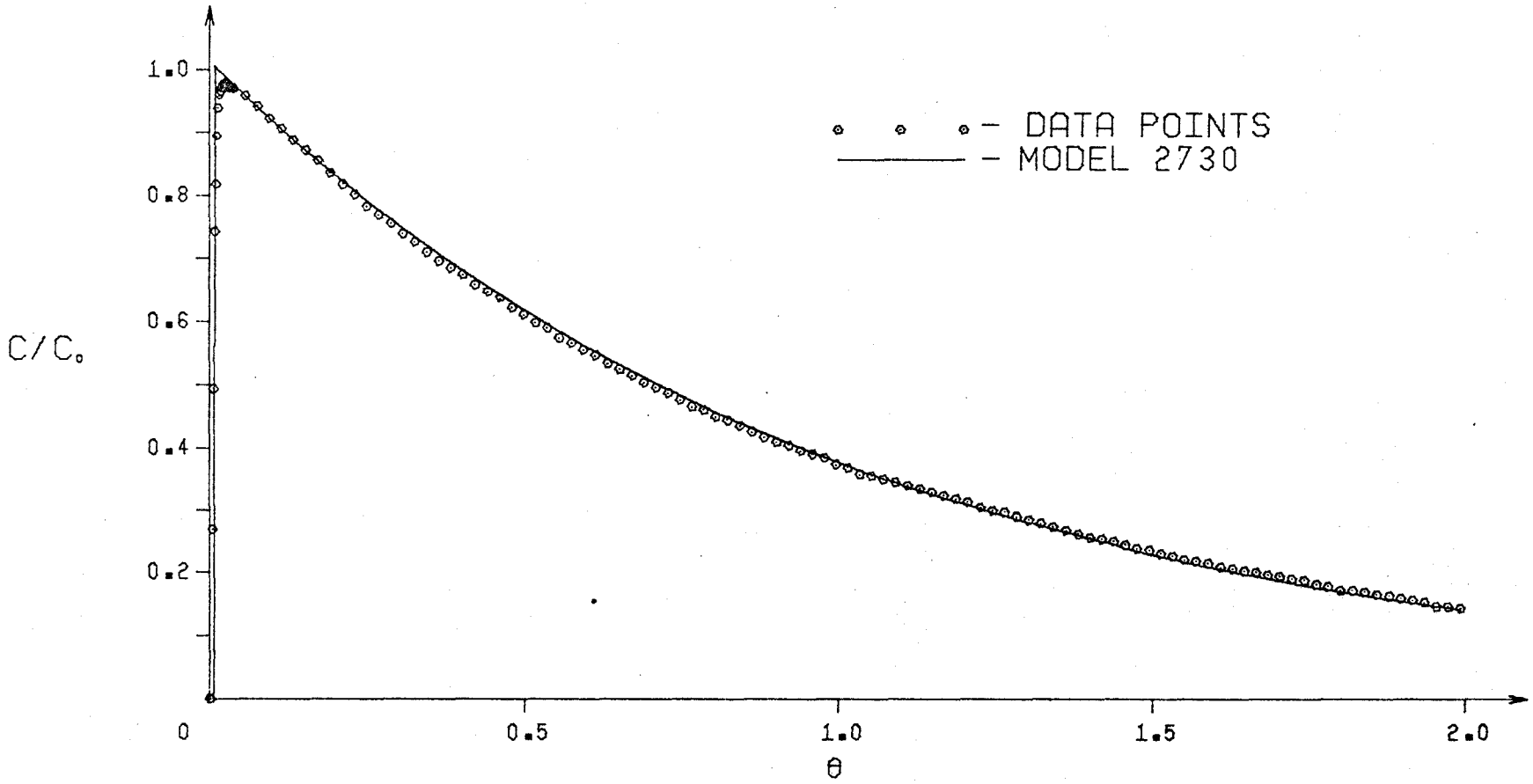


FIGURE (5.23) TIME RESPONSE OF MIXING MODEL FOR RUN #18

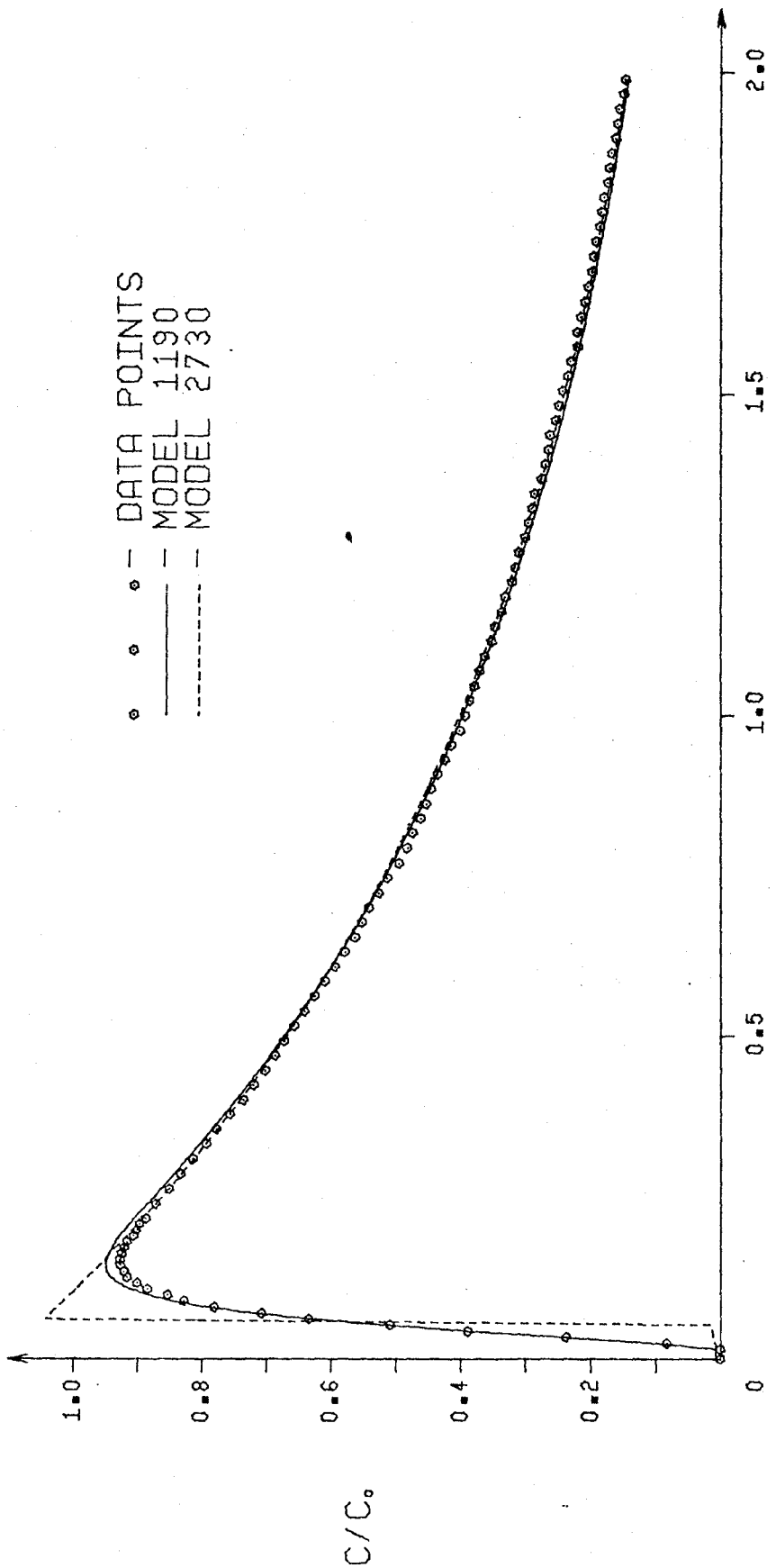


FIGURE (5.24) TIME RESPONSE OF MIXING MODELS FOR RUN #19

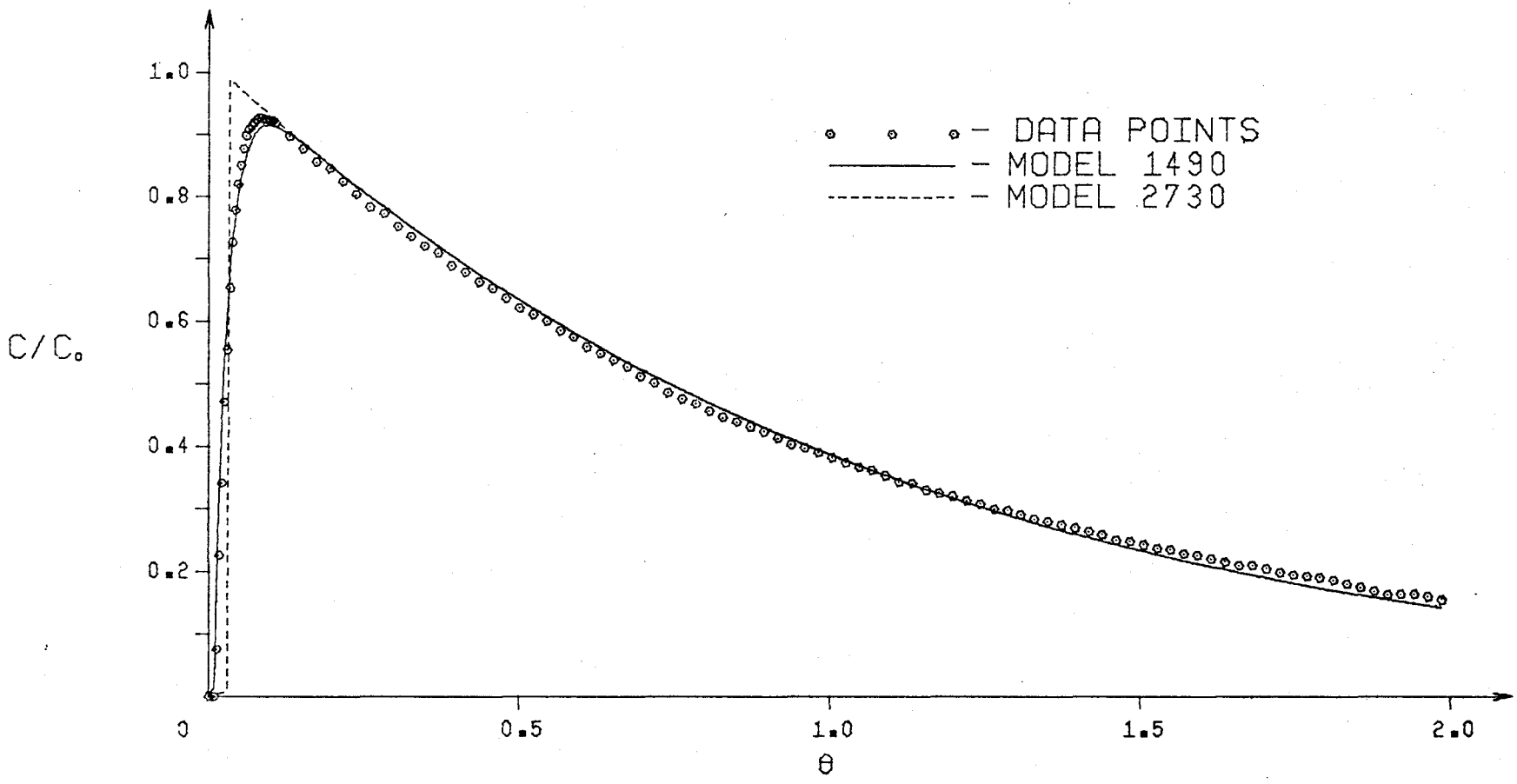


FIGURE (5.25) TIME RESPONSE OF MIXING MODELS FOR RUN #20

5.2.2b Correlation of Parameters to Water Flow Rate

Hudspith (H-1) found that the parameters in the models he used to describe natural stream flow could be correlated to flow. However, in his work he found the Adler-Hovorka model to give the best fit to the data and thus all correlations were made using only this model.

In this study the form of the model was unrestricted. That is the model chosen was the best of a whole range of different PFTR-CSTR component models of a given degree of complexity. At the outset it was hoped that the geometry and air flow rate would be predominate in defining the mixing pattern in the reactor. If this were true, then the model chosen for each of runs #16, #18, #19 and #20 would be the same. Also the residence times of the component PFTR's and CSTR's would be related to the water flow rate. Unfortunately, for the data available for analysis, the form of the model was strongly dependent on the water flow rate. In fact the dependence was so strong that for the range of flows used the "complexity" of the transformed data was altered by the water flow rate.

Ideally the mixing pattern should be defined only by the geometry of the reactor and the air flow rate since these are the only quantities which can be controlled to any extent. For a tank of that type the same model would be chosen for different flow rates and then the component residence times and the flow splits could be correlated with the water flow rate. For the present

work this is not possible and the only conclusion to be drawn is that the laboratory scale model was of poor design for the flow rates used.

CHAPTER 6 - CONCLUSIONS

The following conclusions were made based on the analysis of data in Chapter 5.

1. Since the difference in the prediction of the degree of conversion between that given by the two different kinetic models is negligible with respect to the difference between choosing a CSTR or a PFTR mixing model rather than one of the component models, the choosing of a mixing model is more critical than the choosing of a kinetic model for the particular data analyzed.

2. For the data and mixing models used, the piecewise linearization approximation to the batch kinetic data yields conversion predictions about as accurate as that given by the six parameter Edwards model in the realistic range of total residence times.

3. Because of the strong influence of hydraulic flow rate the mixing phenomena occurring in the laboratory scale model, that resulted in a different mixing model being selected for each flow rate, it was not possible to correlate the model parameters to that flow rate.

4. In Wilson's modelling technique, the category selection should be based solely on the polar trajectory of the transformed data and not the asymptotic slope of the Bodé magnitude plot, since the latter procedure can result in the selection of an incorrect category.

5. Although previously it had only been applied to laboratory scale experiments, the modelling technique developed by Wilson was found to be applicable to full scale reactors.

REFERENCES

A-1 ABRAMOWITZ, M. and STEGUN, I.A., Handbook of Mathematical Functions, Dover Publications, New York, 1965.

B-1 BISCHOFF, K.B., "Mixing and Contacting in Chemical Reactors", Industrial and Engineering Chemistry, 58, 11, 18 (1966).

B-2 BISCHOFF, K.B., "Optimal Continuous Fermentation Reactor Design", Canadian Journal of Chem. Eng., 44, 281 (1966).

B-3 BOYKO, B.I., "Mixing Studies on A Full Scale Aeration Tank", M.Eng. Thesis, McMaster University (1968).

B-4 BUSCH, A.W., Aerobic Biological Treatment of Waste Waters, Oligodynamics Press, Houston, Texas, (1971).

C-1 CARNAHAN, B., LUTHER, H.A., and WILKES, J.O., Applied Numerical Methods, John Wiley & Sons, New York, 1969.

C-2 CHEN, M.S.K., "The Theory of Micromixing for Unsteady State Flow Reactors", Chem. Eng. Sci., 26, 17 (1971).

C-3 CLEMENTS, W.C., "A Note on Determination of the Parameters of the Longitudinal Dispersion Model from Experimental Data", Chem. Eng. Sci., 24, 957 (1969).

C-4 CLEMENTS, W.C., and SCHNELLE, K.B., "Pulse Testing for Dynamic Analysis", Ind. and Eng. Chem. Proc. Design and Development, 2, 94 (1963).

D-1 DANCKWERTS, P.V., "Continuous Flow Systems, Distribution of Residence Times", Chem. Eng. Sci., 2, 1 (1953).

D-2 DANCKWERTS, P.V., "The Effects of Incomplete Mixing on Homogeneous Reactions", Chem. Eng. Sci., 8, 93 (1958).

D-3 DiSTEFANO, J.J., III, STUBBERUD, A.R., and WILLIAMS, I.J., Feedback and Control Systems, McGraw-Hill Inc., New York, 1967.

E-1 EDWARDS, V.H., and WILKE, C.R., "Mathematical Representation of Batch Culture Data", Biotech. and Bioeng., 10, 205 (1968).

F-1 FAN, L.T., ERICKSON, L.E., SHAH, P.S., and TSAI, B.I., "Effect of Mixing on the Washout and Steady State Performance of Continuous Cultures", Biotech. and Bioeng., 12, 1019 (1970).

F-2 FAN, L.T., TSAI, B.I., and ERICKSON, L.E., "Simultaneous Effect of Macromixing and Micromixing on Growth Processes", A.I. Ch. E. Journal, 17, 689 (1971).

G-1 GARRETT, M.T., and SAWYER, C.N., "Kinetics of Removal of Soluble B.O.D. Dry Activated Sludge", Proc. 7th Ind. Waste Conf., Purdue Univ., Ext. Ser. 79, 51 (1952).

G-2 GAUDY, A.F., JR., RAMANATHAN, M., and RAO, B.S., "Kinetic Behavious of Heterogeneous Populations in Completely Mixing Reactors", Biotech and Bioeng, 9, 387 (1967).

G-3 GODA, G., and NAKANISHI, H., "Mixing and Microbial Kinetics of Activated Sludge", Memoirs of the Faculty of Engineering, Kyoto University, 27, 31 (1965).

G-4 GRIEVES, R.B., MILBURY, W.F., and PIPES, W.O., "A Mixing Model for Activated Sludge", Journal Water Pollution Control Federation, 36, 619 (1964).

G-5 GRIEVES, R.B., MILBURY, W.F., and PIPES, W.O., "The Effect of Short Circuiting Upon Sludge Process", Int. Journal of Air and Water Pollution, 8, 199 (1964).

H-1 HUDSPITH, R.C.,"The Mathematical Modelling of Mixing in Natural Streams", M.Eng. Thesis, McMaster University (1971).

K-1 KOWALIK, J., and OSBORNE, M.R., Methods for Unconstrained Optimization Problems, Elsevier Press, New York, 1968.

L-1 LEVENSPIEL, O., Chemical Reaction Engineering, John Wiley and Sons Inc., New York,1962.

M-1 McCABE, BROTHER J., and ECKENFELDER, W.W., JR., "B.O.D. Removal and Sludge Growth in the Activated Sludge Process", Journal Water Pollution Control Federation, 33, 258 (1961).

M-2 McKINNEY, R.E., "Mathematics of Complete Mixing Activated Sludge", Jour. San. Engr. Div., ASCE, 88, SA3, 87 (1962).

M-3 McKINNEY, R.E., and OOTEN, R.J. "Concepts of Complete Mixing Activated Sludge", Transactions of the Nineteenth Annual Conference on Sanitary Engineering, University of Kansa, The Bulletin of Engineering and Architecture, No. 60 (1969).

M-4 MILBURY, W.F., PIPES, W.O., and GRIEVES, R.B., "A Laboratory Study of the Effect of Short Circuiting Upon the Completely Mixed Activated Sludge Process", Int. Journ. of Air and Water Pollution, 9, 41 (1965).

M-5 MONOD, J., "The Growth of Bacterial Cultures", Annual Review of Microbiology, 3, 371 (1949).

N-1 NAITO, M., TAKAMATSU, T., FAN, L.T., and LEE, E.S., "Model Indentification of the Biochemical Oxidation Process", Biotech. and Bioeng., 11, 731 (1969).

N-2 NELDER, J.A.,and MEAD, R., "A Simplex Method of Function Minimization", Computer J, 7, 308 (1965).

P-1 POLLOCK, T.E., "A Batch Biokinetic Study of the Preferential Separation of a Mixed Culture of Micro-organisms Using Small Hydrocyclones", M.Eng. Thesis, McMaster University (1969).

R-1 RIPPIN, D.W.T., "Segregation in a Two Environment Model of a Partially Mixed Chemical Reactor", Chem. Eng. Sci., 22, 247 (1967).

S-1 SINCLAIR, C.G., and BROWN, D.E., "Effect of Incomplete Mixing on the Analysis of the Static Behaviour of Continuous Cultures", Biotech. and Bioeng., 12, 1001 (1970).

S-2 SPIEGEL, M.R., Laplace Transforms, McGraw-Hill Inc., New York, 1965.

T-1 TIMPANY, P.L., "Variations of Axial Mixing in an Aeration Tank", M.Eng. Thesis, McMaster University (1967).

T-2 TSAI, B.I., ERIKSON, L.E., and FAN, L.T. "The Effect of Micromixing on Growth Process", Biotech. and Bioeng., 11, 181 (1969).

V-1 VERHOFF, T.H., SUNDARESAN, K.R., and TENNEY, M.W., "A Mechanism of Microbial Cell Growth", Biotech. and Bioeng., 13, 411 (1972).

W-1 WILSON, A.W.,"Mathematical Modelling of Partially Mixed Reactors", Ph.D. Thesis, McMaster University (1971).

Y-1 YAMAZAKI, H., and ICHIKAWA, A., "Effect of Segregation on Dynamic Response of a Continuous Flow Chemical Reactor", Journal of Chem. Eng. of Japan, 2, 100 (1969).

Y-2 YOUNG, T.B., BRULEY, D.F., and BUNGAY, H.R., III, "A Dynamic Mathematical Model of the Chemostat", Biotech. and Bioeng., 12, 747 (1970).

Z-1 ZWEITERING, Th.N., "The Degree of Mixing in Continuous Flow Systems", Chem. Eng. Sci., 11, 1 (1959).

NOMENCLATURE

		UNITS
	English Alphabetic Characters	
A	= magnitude of sinusoidal input function	
a	= empirical constant, also a coefficient	
b	= empirical constant, also a coefficient	
C	= concentration	$(M)(L)^{-3}$
c	= real number	
c_{ik}	= residue of transfer function for i'th repetition of k'th pole	
D	= dispersion coefficient	$(L)^2(T)^{-1}$
E(jω)	= error function in frequency domain	
E(t)	= residence time distribution function	$(T)^{-1}$
e(t)	= error function in time domain	
F(jω)	= Fourier transform function of f(t)	
F(s)	= Laplace transform function of f(t)	
f(t)	= some function of time	
IM	= imaginary response	
J	= degree of segregation	
j	= $\sqrt{-1}$	
K	= empirical constant	
L	= length	(L)
M	= mass	(M)
m	= index limit	
N	= integer number	
n	= index limit	

		UNITS
Pe	= Peclet number	
$P(j\omega)$	= transfer function in frequency domain	
$P(s)$	= transfer function in Laplace domain	
p_i	= i'th pole of tansfer function	$(T)^{-1}$
Q	= flow rate	$(L)^3(T)^{-1}$
RL	= real response	
RSS	= residual sum of squares	
r	= index limit	
$S(\omega)_n$	= normalized frequency content of pulse waveform	
s	= Laplace variable	$(T)^{-1}$
t	= time	(T)
$\bar{t}$	= mean residence time	(T)
u	= velocity	$(L)(T)^{-1}$
V	= volume	$(L)^3$
X	= conversion	
$X(s)$	= input function in Laplace domain	
x	= distance	(L)
$x(t)$	= input function in time domain	
$Y(s)$	= output function in Laplace domain	
$y(t)$	= output function in time domain	

Greek Alphabetic Characters

δ	= Dirac-delta or impulse function
θ	= reduced time (= $t/\bar{t}$ or t/τ)
λ	= angle (for residence time distribution of dispersion model

		UNITS
μ	= specific growth rate ($= \frac{1}{C_B}\frac{dC_B}{dt}$)	$(T)^{-1}$
Π	indicates a product operation	
π	= 3.14159	
Σ	indicates a summation operation	
σ	= real part of Laplace variable, s	$(T)^{-1}$
τ	= residence time	(T)
ϕ	= phase angle	
$\times$	= empirical constant	
ω	= frequency, also imaginary part of Laplace variable, s	$(T)^{-1}$

Subscripts

B	refers to biomass
batch	refers to batch kinetic experiment
C	refers to soluble carbon
i	index
k	index
max	refers to maximum of upper limit
N	refers to nominal
o	refers to initial (as for initial concentration)
obs	refers to observed response
pred	refers to predicted response
ss	refers to steady-state
T	refers to total

A P P E N D I C I E S

APPENDIX A

```
C         THIS PROGRAM CONVERTS RAW FLUORESCENCE DATA TO CONCENTRATION
C         DATA AND PERFORMS A LEAST SQUARES FIT OF THE FORM C=A*EXP(B*T)
C         ON THE TAIL PORTION OF THE CURVE
C
C         NJOBS - NO. OF EXPERIMENTAL RUNS TO BE DONE
C         NRUN - EXPERIMENTAL RUN NUMBER
C         N1 - NUMBER OF POINTS IN FIRST STEP SIZE INTERVAL
C         NO - NUMBER OF FIRST POINT IN SECOND STEP SIZE INTERVAL WHICH
C              USES LOGARITHMIC CALIBRATION CURVE
C         NN - NUMBER OF POINTS IN SECOND STEP SIZE INTERVAL
C         NCC - CALIBRATION CURVE NUMBER
C         DT - SECOND STEP SIZE IN TIME
C         CO1 - FLUORESCENCE DATA FOR FIRST STEP SIZE
C         CO2 - FLUORESCENCE DATA FOR SECOND STEP SIZE
C         A - PARAMETER IN FIT EQUATION
C         B - PARAMETER IN FIT EQUATION
C
      DIMENSION CO1(300),CO2(800),CP(800),T(800)
      REAL N
      READ(5,120) NJOBS
      DO 399 IJK=1,NJOBS
      READ(5,130) NRUN,N1,NO,NN,NCC
      READ(5,100) DT
C
C         BUILD TIME ARRAY
C
      T(1)=DT
      DO 500 I=2,800
      T(I)=T(I-1)+DT
  500 CONTINUE
      CO1(1)=0.
      READ(5,100) (CO1(I),I=2,N1)
      CO2(1)=CO1(N1)
      READ(5,100) (CO2(I),I=2,NN)
      N=FLOAT(NN-NO+1)
C
C         USE CALIBRATION EQUATIONS TO CONVERT RAW FLUORESCENCE DATA TO
C         CONCENTRATION
C
      IF(NCC.NE.1) GO TO 400
      DO 600 I=1,N1
      IF(CO1(I).LE.21.) GO TO 610
      CO1(I)=8.5+1.48993*CO1(I)
      GO TO 600
  610 CONTINUE
      CO1(I)=3.6*CO1(I)**.790706
  600 CONTINUE
      GO TO 420
  400 CONTINUE
      DO 620 I=1,N1
      IF(CO1(I).LE.21.) GO TO 630
      CO1(I)=8.+1.43077*CO1(I)
      GO TO 620
  630 CONTINUE
      CO1(I)=2.65*CO1(I)**.871047
```

```
  620 CONTINUE
  420 CONTINUE
      IF(NCC.NE.1) GO TO 410
      DO 700 I=1,NN
      IF(CO2(I).LE.21.) GO TO 710
      CO2(I)=8.5+1.48993*CO2(I)
      GO TO 700
  710 CONTINUE
      CO2(I)=3.6*CO2(I)**.790766
  700 CONTINUE
      GO TO 430
  410 CONTINUE
      DO 720 I=1,NN
      IF(CO2(I).LE.21.) GO TO 730
      CO2(I)=8.+1.43077*CO2(I)
      GO TO 720
  730 CONTINUE
      CO2(I)=2.65*CO2(I)**.871047
  720 CONTINUE
  430 CONTINUE
C
C        PERFORM LINEAR LEAST SQUARES FIT
C
      SX=0.
      SY=0.
      SXY=0.
      SX2=0.
      DO 510 I=NO,NN
      Y=ALOG(CO2(I))
      X=T(I)
      SX=SX+X
      SY=SY+Y
      SXY=SXY+X*Y
      SX2=SX2+X*X
  510 CONTINUE
      B=(N*SXY-SX*SY)/(N*SX2-SX*SX)
      A=EXP(SY/N-B*SX/N)
      WRITE(6,230) NRUN
      WRITE(6,200)
C
C        CALCULATE PREDICTED VALUES OF C FOR TAIL
C
      DO 520 I=NO,NN
      CP(I)=A*EXP(B*T(I))
      WRITE(6,210) T(I),CO2(I),CP(I)
  520 CONTINUE
      NN=NN+1
      DO 530 I=NN,400
      CO2(I)=A*EXP(B*T(I))
  530 CONTINUE
      WRITE(7,240)
      WRITE(7,220) (CO1(I),I=1,N1)
      WRITE(7,220) (CO2(I),I=1,400)
  399 CONTINUE
      STOP
  100 FORMAT(16F5.2)
  110 FORMAT(2I5)
```

```
120 FORMAT(16I5)
130 FORMAT(A10,14I5)
200 FORMAT(//18X,*T*,11X,*CO*,11X,*CP*//)
210 FORMAT(10X,3E15.3)
220 FORMAT(8E10.3)
230 FORMAT(1H1,19X,*LEAST SQUARES TAIL FIT*//1H0,20X,*RUN NUMBER *,A10
   1)
240 FORMAT(*1111111111111111111111111111111111111111111111111111111111*)
    END
```

```
C       FREQUENCY RESPONSE CALCULATIONS FROM IMPULSE DATA USING FILON'
C       QUADRATURE FORMULAE
C
C       (ADAPTED FROM WILSON (W-1))
C
C       PROVISION IS MADE FOR TWO STEP SIZES IF DESIRED
C       THE LAST AND FIRST POINTS OF THE FIRST AND LAST STEP SIZE
C       SEGMENTS RESPECTIVELY MUST BE THE SAME POINT
C
C       NJOBS - NO. OF EXPERIMENTAL RUNS TO BE DONE
C       NFREQS - NO. OF FREQUENCIES TO BE USED
C       NRUN - EXPERIMENTAL RUN IDENTIFICATION
C       NPTSY1 - NO. OF POINTS FOR FIRST STEP SIZE
C       NPTSY2 - NO. OF POINTS FOR SECOND STEP SIZE
C       DELTY1 - FIRST TIME STEP SIZE
C       DELTY2 - SECOND TIME STEP SIZE
C       DTIME - DELAY TIME BETWEEN IMPULSE INPUT AND FIRST OUTPUT
C               RESPONSE
C       DW - SIZE OF FREQUENCY INPUTS
C       TAUTOT - TOTAL RESIDENCE TIME
C
      REAL  MR
      DIMENSION W(200),Y1(300),Y2(850),
     1TRFFR(200), TRFFI(200), MR(200), GAIN(200), PHASE(200),
     2SWNIN(200), SWNOUT(200)
      READ (5,100)  NJOBS
      READ (5,100)  NFREQS
      DO 399  JOBS = 1, NJOBS
      READ(5,103) NRUN,NPTSY1,NPTSY2
      READ (5,101)  DELTY1, DELTY2
      READ (5,101)  DTIME
      READ(5,102) DW
      W(1)=0.
      DO 900 J=2,NFREQS
      W(J)=W(J-1)+DW/100.
  900 CONTINUE
      DELTY1=DELTY1/60.
      DELTY2=DELTY2/60.
      DTIME=DTIME/60.
      READ(5,101) (Y1(I),I=1,NPTSY1)
C
C       CHECK TO SEE IF TWO STEP SIZES ARE BEING USED
C
      IF (NPTSY2.EQ.0)  GO TO 440
      READ(5,101) (Y2(I),I=1,NPTSY2)
  440 CONTINUE
      WRITE (6,200)  NRUN
      WRITE (6,201)  NPTSY1, NPTSY2
      WRITE (6,202)  DELTY1, DELTY2
      NY1M1 = NPTSY1 - 1
      NY2M1 = NPTSY2 - 1
      ENY1M1 = FLOAT(NY1M1)
      ENY2M1 = FLOAT(NY2M1)
      NY1M2 = NPTSY1 - 2
      NY2M2 = NPTSY2 - 2
```

```
C
C         CALCULATIONS OF AREAS AND FIRST MOMENTS OF CONCENTRATION-TIME
C         DATA
C
      TOPY1 = Y1(1) * DELTY1 / 2.0 * DELTY1 / 4.0 + Y1(NPTSY1) * DELTY1
     1 / 2.0 * (ENY1M1 * DELTY1 - DELTY1 / 4.0)
      BOTY1 = (Y1(1) + Y1(NPTSY1)) * DELTY1 / 2.0
      DO 306  I = 2, NY1M1
      FLTIM1 = I - 1
      TOPY1 = TOPY1 + Y1(I) * DELTY1 * FLTIM1 * DELTY1
  306 BOTY1 = BOTY1 + DELTY1 * Y1(I)
      IF (NPTSY2.EQ.0)  GO TO 476
      TIMEY1 = ENY1M1 * DELTY1
      TOPY2 = Y2(1) * DELTY2 / 2.0 * (TIMEY1 + DELTY2 / 4.0) +
     1Y2(NPTSY2) * DELTY2 / 2.0 * (TIMEY1 + ENY2M1 * DELTY2 -
     2DELTY2 / 4.0)
      BOTY2 = (Y2(1) + Y2(NPTSY2)) * DELTY2 / 2.0
      DO 312  I = 2, NY2M1
      FLTIM1 = I - 1
      TOPY2 = TOPY2 + Y2(I) * DELTY2 * (TIMEY1 + FLTIM1 * DELTY2)
  312 BOTY2 = BOTY2 + Y2(I) * DELTY2
      GO TO 477
  476 TOPY2 = 0.0
      BOTY2 = 0.0
  477 CONTINUE
      TOPY = TOPY1 + TOPY2
      BOTY = BOTY1 + BOTY2
      CENTY = TOPY / BOTY
      TAUTOT = CENTY + DTIME
      WRITE (6,213)
      DO 370  I = 1, NPTSY1
  370 Y1(I) = Y1(I) / BOTY
      WRITE (6,210)  (Y1(I), I = 1, NPTSY1)
      IF (NPTSY2.EQ.0)  GO TO 470
      DO 371  I = 1, NPTSY2
  371 Y2(I) = Y2(I) / BOTY
      WRITE (6,211)  (Y2(I), I = 1, NPTSY2)
  470 CONTINUE
      DO 398  J = 1, NFREQS
      THETA = W(J) * DELTY1
      CALL ABG(ALPHA, BETA, GAMMA, THETA)
      S2N = Y1(1) * COS(W(J) * 0.0) / 2.0 + Y1(NPTSY1) * COS(W(J) *
     1ENY1M1 * DELTY1) / 2.0
      DO 324  I = 3, NY1M2, 2
      FLTIM1 = I - 1
  324 S2N = S2N + Y1(I) * COS(W(J) * FLTIM1 * DELTY1)
      S2NM1 = 0.0
      DO 325  I = 2, NY1M1, 2
      FLTIM1 = I - 1
  325 S2NM1 = S2NM1 + Y1(I) * COS(W(J) * FLTIM1 * DELTY1)
      A1 = DELTY1 * (ALPHA * (Y1(1) * COS(W(J) * 0.0 + 1.5707963271)
     1 - Y1(NPTSY1) * COS(W(J) * ENY1M1 * DELTY1 + 1.5707963271)) +
     2BETA * S2N + GAMMA * S2NM1)
      S2N = Y1(1) * SIN(W(J) * 0.0) / 2.0 + Y1(NPTSY1) * SIN(W(J) *
     1ENY1M1 * DELTY1) / 2.0
      DO 304  I = 3, NY1M2, 2
      FLTIM1 = I - 1
```

```
304 S2N = S2N + Y1(I) * SIN(W(J) * FLTIM1 * DELTY1)
    S2NM1 = 0.0
    DO 305  I = 2, NY1M1, 2
    FLTIM1 = I - 1
305 S2NM1 = S2NM1 + Y1(I) * SIN(W(J) * FLTIM1 * DELTY1)
    B1 = DELTY1 * (ALPHA *(Y1(1) * COS(W(J) * 0.0) - Y1(NPTSY1) *
   1COS(W(J) * ENY1M1 * DELTY1)) + BETA * S2N + GAMMA * S2NM1)
    IF (NPTSY2.EQ.0)  GO TO 406
    THETA = W(J) * DELTY2
    CALL ABG(ALPHA, BETA, GAMMA, THETA)
    S2N = Y2(1) * COS(W(J) * TIMEY1) / 2.0 + Y2(NPTSY2) * COS(W(J) *
   1(TIMEY1 + ENY2M1 * DELTY2)) / 2.0
    DO 330  I = 3, NY2M2, 2
    FLTIM1 = I - 1
330 S2N = S2N + Y2(I) * COS(W(J) *(TIMEY1 + FLTIM1 * DELTY2))
    S2NM1 = 0.0
    DO 331  I = 2, NY2M1, 2
    FLTIM1 = I - 1
331 S2NM1 = S2NM1 + Y2(I) * COS(W(J) * (TIMEY1 + FLTIM1 * DELTY2))
    A2 = DELTY2 * (ALPHA * (Y2(1) * COS(W(J) * TIMEY1 + 1.5707963267
   1-Y2(NPTSY2) * COS(W(J) * (TIMEY1 + ENY2M1 * DELTY2) + 1.5707963267
   2) + BETA * S2N + GAMMA * S2NM1)
    S2N = Y2(1) * SIN(W(J) * TIMEY1) / 2.0 + Y2(NPTSY2) * SIN(W(J) *
   1(TIMEY1 + ENY2M1 * DELTY2)) / 2.0
    DO 310  I = 3, NY2M2, 2
    FLTIM1 = I - 1
310 S2N = S2N + Y2(I) * SIN(W(J) * (TIMEY1 + FLTIM1 * DELTY2))
    S2NM1 = 0.0
    DO 311  I = 2, NY2M1, 2
    FLTIM1 = I - 1
311 S2NM1 = S2NM1 + Y2(I) * SIN(W(J) * (TIMEY1 + FLTIM1 * DELTY2))
    B2 = DELTY2 * (ALPHA *(Y2(1) * COS(W(J) * TIMEY1) - Y2(NPTSY2) *
   1COS(W(J) *(TIMEY1 + ENY2M1 * DELTY2))) + BETA * S2N + GAMMA
   2 * S2NM1)
    GO TO 407
406 A2 = 0.0
    B2 = 0.0
407 CONTINUE
    A = A1 + A2
    B = B1 + B2
    C = 1.0
    D = 0.0
    TRFFR(J) = (A * C + B * D) / (C * C + D * D)
    TRFFI(J) = (A * D - B * C) / (C * C + D * D)
    MR(J) = SQRT(TRFFR(J) * * 2 + TRFFI(J) * * 2)
    IF (J.GT.1)  GO TO 452
    SSMAG = MR(1)
    SSGAIN = 20.0 * ALOG10(SSMAG)
452 CONTINUE
    MR(J) = MR(J) / SSMAG
    GAIN(J) = 20.0 * ALOG10(MR(J))
    PHASE(J) = 57.29578 * ATAN(TRFFI(J) / TRFFR(J))
    SWNIN(J) = SQRT(C * * 2 + D * * 2)
    SWNOUT(J) = SQRT(A * * 2 + B * * 2)
    IF (TRFFR(J))  421, 424, 427
421 IF (TRFFI(J)) 422, 423, 422
422 TAN = PHASE(J) - 180.0
```

```
      GO TO 430
  423 TAN = -180.0
      GO TO 430
  424 IF (TRFFI(J)) 425, 431, 426
  425 TAN = -90.0
      GO TO 430
  426 TAN = -270.0
      GO TO 430
  427 IF (TRFFI(J)) 431, 431, 428
  428 TAN = PHASE(J) - 360.0
  430 PHASE(J) = TAN
  431 CONTINUE
C
C         CORRECT PHASE ANGLE FOR PFTR IN SERIES TIME
C
      PHASE(J) = PHASE(J) - DTIME * W(J) * 57.29578
  398 CONTINUE
      WRITE (6,205)  BOTY
      WRITE (6,206)  SSGAIN
      WRITE (6,208)  CENTY, TAUTOT, DTIME
      WRITE (6,203)
      WRITE (6,204)  ((W(J), MR(J), GAIN(J), PHASE(J), SWNIN(J),
     1SWNOUT(J), TRFFR(J), TRFFI(J)), J = 1, NFREQS)
      WR+TE(7,230)
      WRITE (7,299)  ((W(J), MR(J), GAIN(J), PHASE(J), SWNIN(J),
     1SWNOUT(J), TRFFR(J), TRFFI(J)), J = 1, NFREQS)
  399 CONTINUE
  100 FORMAT  (8I10)
  101 FORMAT  (8E10.3)
  102 FORMAT(16F5.0)
  103 FORMAT  (A10, 7I10)
  200 FORMAT  (1H1, 49HFREQUENCY RESPONSE CALCULATIONS FROM IMPULSE DATA
     1/10X, 23HUSING FILONS QUADRATURE/// 1H0, 10HRUN NUMBER, 1X, A10)
  201 FORMAT  (1H0, 8HNPTSY1 =, I5 / 1H0, 8HNPTSY2 =, I5)
  202 FORMAT  (1H0, 8HDELTY1 =, E10.3 / 1H0, 8HDELTY2 =, E10.3)
  203 FORMAT  (1H1, 9HFREQUENCY, 6X, 9HMAGNITUDE, 8X, 4HGAIN, 8X,
     111HPHASE ANGLE, 7X, 5HINPUT, 9X, 6HOUTPUT, 8X, 9HREAL PART, 6X,
     29HIMAG PART/ 1X, 9H(RAD/HR) , 8X, 5HRATIO, 7X, 10H(DECIBELS), 6X,
     39H(DEGREES), 6X, 9HFREQUENCY, 6X, 9HFREQUENCY, 5X, 11HOF TRANSFER
     44X, 11HOF TRANSFER/ 62X, 7HCONTENT, 8X, 7HCONTENT, 7X,
     58HFUNCTION, 7X, 8HFUNCTION)
  204 FO-MAT  (1H0, E10.3, 4X, E10.3, 5X, E10.3, 5X, E10.3, 5X, E10.3,
     15X, E10.3, 5X, E10.3, 5X, E10.3)
  205 FORMAT  (1H0, 25HAREA UNDER OUTPUT PULSE =, E10.3)
  206 FORMAT  (1H0, 8HSSGAIN = E10.3, 9H DECIBELS)
  207 FORMAT  (8E10.3)
  208 FORMAT  (
     11H0, 26HCENTROID OF OUTPUT CURVE =, E10.3, 1X, 7HHOURS   / 1H0,
     227HMEAN TOTAL RESIDENCE TIME =, E10.3, 1X,
     334HHOURS   (INCLUDING PFTR-IN-SERIES) / 1H0,
     431HPFTR-IN-SERIES RESIDENCE TIME =, E10.3, 1X, 7HHOURS  )
  209 FORMAT  (1H-, 43HTHE RAW DATA CONVERTED TO CONCENTRATION IS- ///)
  210 FORMAT  (1H0, 16HFIRST STEP SIZE- / (1H ,5E10.3/))
  211 FORMAT  (1H0, 17HSECOND STEP SIZE- / (1H ,5E10.3/ ))
  212 FORMAT  (1H0, 8HYLIMLO =, E10.3 / 1H0, 7HYSPAN =, E10.3)
  213 FORMAT  (1H-, 35HTHE RESIDENCE TIME DISTRIBUTION IS- //)
  230 FORMAT(*1111111111111111111111111111111111111111111111111111111*)
```

```
  299 FORMAT  (8E10.3)
      STOP
      END
      SUBROUTINE ABG(ALPHA, BETA, GAMMA, THETA)
C     CHECK TO SEE IF THETA IS SMALL
      IF (THETA.LT.0.35)  GO TO 400
      ALPHA = 1.0 / THETA + SIN(2.0 * THETA) / (2.0 * THETA * * 2)
     1 - 2.0 * SIN(THETA) * * 2 / THETA * * 3
      BETA = 2.0 * ((COS(THETA) * * 2 + 1.0) / THETA * * 2 -
     1SIN(2.0 * THETA) / THETA * * 3)
      GAMMA = 4.0 * (SIN(THETA) / THETA * * 3 - COS(THETA) /
     1THETA * * 2)
      RETURN
  400 ALPHA = 2.0 * THETA * * 3 / 45.0 - 2.0 * THETA * * 5 / 315.0 +
     12.0 * THETA * * 7 / 4725.0
      BETA = 2.0 / 3.0 + 2.0 * THETA * * 2 / 15.0 - 4.0 * THETA * * 4 /
     1105.0 + 2.0 * THETA * * 6 / 567.0 - 4.0 * THETA * * 8 / 22275.0
      GAMMA = 4.0 / 3.0 - 2.0 * THETA * * 2 / 15.0 + THETA * * 4 / 210.0
     1- THETA * * 6 / 11340.0 + THETA * * 8 / 997920.0
      RETURN
      END



C         THIS PROGRAM PLOTS BODE GAIN GRAPHS WITH DATA
C         SUPPLIED BY THE FOURIER TRANSFORM PROGRAM
C
C         NJOBS - NO. OF PLOTS TO BE DONE
C         RUN - DUMMY VARIABLE
C         X - ARRAY CONTAINING VALUES OF FREQUENCY AND GAIN
C         W - LOG 10 (FREQUENCY)
C         G - GAIN
C         Y - PARAMETER FOR 'Y' GRADUATIONS
C         T - VALUES FOR PLOTTING LOGARITHMIC 'X' GRADUATIONS
C         Z - PARAMETER FOR 'X' GRADUATIONS
C         XX - PARAMETER FOR 'X' GRADUATIONS
C
      DIMENSION X(101,2),W(100),G(100),T(5)
      CALL DATE(THEDATE)
C
C         PLOT NAME AND DATE
C
      CALL LETTER(8,.24,90.,.5,1.,8HCRAWFORD)
      CALL LETTER(10,.24,90.,1.,1.,THEDATE)
      CALL PLOT(1.5,0.,-3)
      READ(5,110) XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX
C
C         SET UP DATA SCALING
C
      CALL PLTIN(XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX)
      READ(5,100) NJOBS
      DO 399 IJK=1,NJOBS
      READ(5,110) RUN
      READ(5,130) ((X(I,J),J=1,2),I=1,101)
      DO 600 I=1,100
      W(I)=ALOG10(X(I+1,1))
      G(I)=X(I+1,2)
```

```
  600 CONTINUE
C
C         PLOT AXES
C
      CALL ARROW(1.9,8.,9.,8.,3)
      CALL ARROW(2.,8.1,2.,1.,3)
C
C         PLOT ,Y, GRADUATIONS
C
      Y=2.
      DO 700 I=1,3
      CALL PLOT(1.9,Y,3)
      CALL PLOT(2.,Y,2)
      Y=Y+2.
  700 CONTINUE
      T(1)=2.*ALOG10(2.)
      T(2)=2.*ALOG10(4.)
      T(3)=2.*ALOG10(6.)
      T(4)=2.*ALOG10(8.)
      T(5)=2.
C
C         PLOT ,X, GRADUATIONS
C
      Z=2.
      DO 710 J=1,3
      DO 720 I=1,5
      XX=Z
      XX=XX+T(I)
      CALL PLOT(XX,8.1,3)
      CALL PLOT(XX,8.,2)
  720 CONTINUE
      Z=Z+2.
  710 CONTINUE
C
C         NUMBER AXES
C
      CALL LETTER(3,.1,0.,1.5,1.95,3H-30)
      CALL LETTER(3,.1,0.,1.5,3.95,3H-20)
      CALL LETTER(3,.1,0.,1.5,5.95,3H-10)
      CALL LETTER(3,.1,0.,1.5,7.95,3H  0)
      CALL LETTER(4,.1,0.,1.8,8.2,4H0.01)
      CALL LETTER(3,.1,0.,3.85,8.2,3H0.1)
      CALL LETTER(3,.1,0.,5.85,8.2,3H1.0)
      CALL LETTER(4,.1,0.,7.8,8.2,4H10.0)
C
C         LABEL AXES
C
      CALL LETTER(4,.15,0.,.8,5.05,4HGAIN)
      CALL LETTER(4,.15,0.,.8,4.8,4H(DB))
      CALL LETTER(9,.15,0.,4.365,8.75,9HFREQUENCY)
      CALL LETTER(10,.15,0.,4.3,8.5,10H(RAD./HR.))
C
C         PLOT SOLID LINE
C
      CALL PLTMPL(W,G,100)
      CALL PLOT(10.,0.,-3)
  399 CONTINUE
```

```
      CALL PLOT(12.,0.,999)
      STOP
  100 FORMAT(1615)
  110 FORMAT(8F10.0)
  130 FORMAT(E10.3,10X,E10.3)
      END



C         THIS PROGRAM PLOTS POLAR GRAPHS WITH DATA SUPPLIED BY THE
C         FOURIER TRANSFORM PROGRAM
C
C         NJOBS - NO. OF PLOTS TO BE DONE
C         Z - PARAMETER FOR 'X' GRADUATIONS
C         Y - PARAMETER FOR 'Y' GRADUATIONS
C         RUN - DUMMY VARIABLE
C         X - ARRAY CONTAINING REAL AND IMAG. PARTS OF DATA
C         RL - REAL PART IN DATA UNITS
C         IM - IMAG. IN DATA UNITS
C
      DIMENSION X(101,4),RL(101)
      REAL IM(101)
      READ(5,110) XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX
C
C         SET UP DATA SCALING
C
      CALL PLTIN(XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX)
      CALL DATE(THEDATE)
C
C         PLOT NAME AND DATE
C
      CALL LETTER(8,.24,90.,.5,1.,8HCRAWFORD)
      CALL LETTER(10,.24,90.,1.,1.,THEDATE)
      CALL PLOT(1.5,0.,-3)
      READ(5,120) NJOBS
      DO 399 IJK=1,NJOBS
C
C         PLOT AXES
C
      CALL ARROW(2.,6.,10.5,6.,3)
      CALL ARROW(4.5,8.5,4.5,2.,2)
C
C         PLOT ,X, GRADUATIONS
C
      Z=2.5
      DO 600 I=1,4
      CALL PLOT(Z,6.05,3)
      CALL PLOT(Z,5.95,2)
      Z=Z+.5
  600 CONTINUE
      Z=5.
      DO 610 I=1,10
      CALL PLOT(Z,6.05,3)
      CALL PLOT(Z,5.95,2)
      Z=Z+.5
  610 CONTINUE
C
```

```
C         PLOT ,Y, GRADUATIONS
C
      Y=8.
      DO 620 I=1,4
      CALL PLOT(4.45,Y,3)
      CALL PLOT(4.55,Y,2)
      Y=Y-.5
  620 CONTINUE
      Y=5.5
      DO 630 I=1,7
      CALL PLOT(4.45,Y,3)
      CALL PLOT(4.55,Y,2)
      Y=Y-.5
  630 CONTINUE
C
C         NUMBER AXES
C
      CALL LETTER(3,.1,0.,4.6,7.95,3H0.4)
      CALL LETTER(3,.1,0.,4.6,6.95,3H0.2)
      CALL LETTER(3,.1,0.,5.35,6.1,3H0.2)
      CALL LETTER(3,.1,0.,6.35,6.1,3H0.4)
      CALL LETTER(3,.1,0.,7.35,6.1,3H0.6)
      CALL LETTER(3,.1,0.,8.35,6.1,3H0.8)
      CALL LETTER(3,.1,0.,9.35,6.1,3H1.0)
C
C         LABEL AXES
C
      CALL LETTER(9,.15,0.,3.825,8.7,9HIMAGINARY)
      CALL LETTER(4,.15,0.,10.7,5.95,4HREAL)
      READ(5,110) RUN
      READ(5,100) ((X(I,J),J=1,2),I=1,101)
      DO 640 I=1,101
      RL(I)=X(I,1)
      IM(I)=X(I,2)
  640 CONTINUE
C
C         PLOT SOLID LINE
C
      CALL PLTMPL(RL,IM,101)
      CALL PLOT(12.,0.,-3)
  399 CONTINUE
      CALL PLOT(12.,0.,999)
      STOP
  100 FORMAT(60X,2E10.3)
  110 FORMAT(8F10.0)
  120 FORMAT(5I5)
      END
```

APPENDIX C

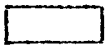
CATALOGUE OF PFTR/CSTR COMPONENT MODELS

LEGEND - Q_T = total flowrate.

Q_1 = flowrate through upper branch of a parallel network.

(Q_T-Q_1) = flowrate through lower branch of a parallel network.

 = indicates a CSTR

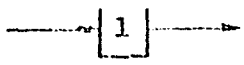
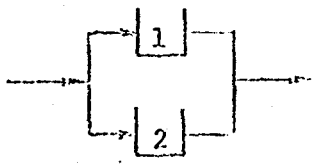
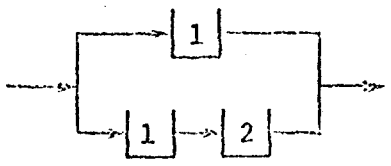
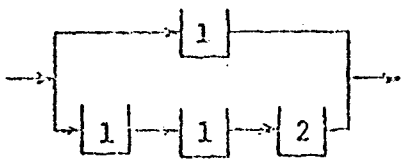
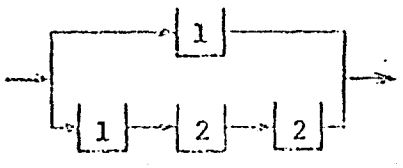
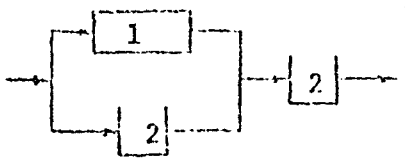
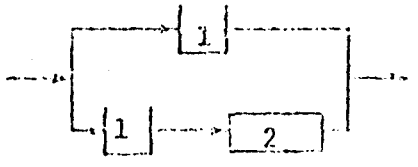
 = indicates a PFTR

NOTE - Repeated numbers identifying the model components mean that the residence time of those components are equal.

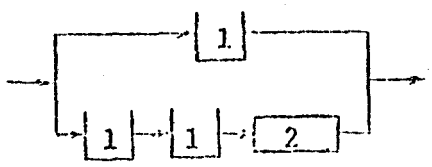
- The code number associated with each model is for use in the parameter estimation routine listed in Appendix D.

(ADAPTED FROM WILSON (W-1))

CATEGORY 1 MODELS
3 PARAMETERS OR LESS

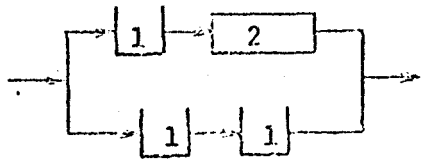
	Model Configuration	Transfer Function
0100 1		$\frac{1}{1 + T_1 s}$
0600 2		$\frac{Q_1}{Q_T}\left[\frac{1}{1 + T_1 s}\right] + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1 + T_2 s}\right]$
1010 3		$\frac{Q_1}{Q_T}\left[\frac{1}{1 + T_1 s}\right] + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1 + T_1 s}\right]\left[\frac{1}{1 + T_2 s}\right]$
1320 4		$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right] + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2\left[\frac{1}{1+T_2 s}\right]$
1330 5		$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right] + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_2 s}\right]^2$
1910 6		$\left\{\frac{Q_1}{Q_T}\exp(-T_1 s) + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]\right\}\left[\frac{1}{1+T_2 s}\right]$
2210 7		$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right] + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\exp(-T_2 s)$

2530
8



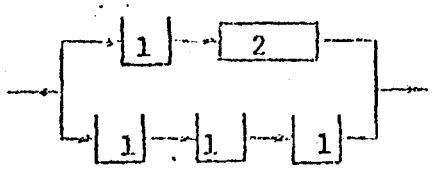
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right] + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2 \exp(-T_2 s)$$

2730
9



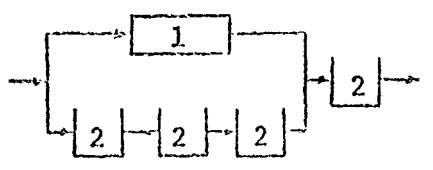
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]\exp(-T_2 s) + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2$$

3240
10



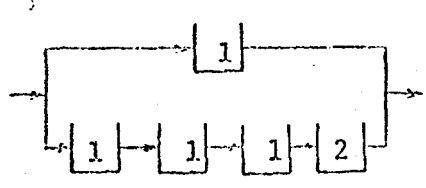
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]\exp(-T_2 s) + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^3$$

3440
11



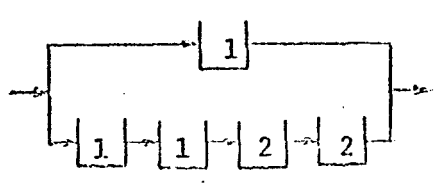
$$\left\{\frac{Q_1}{Q_T}\exp(-T_1 s) + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^3\right\}\left[\frac{1}{1+T_2 s}\right]$$

3520
12



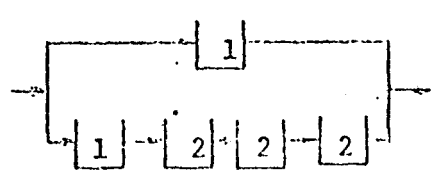
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right] + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^3\left[\frac{1}{1+T_2 s}\right]$$

3530
13



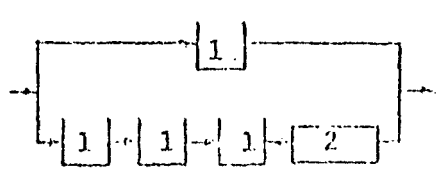
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right] + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2\left[\frac{1}{1+T_2 s}\right]^2$$

3540
14



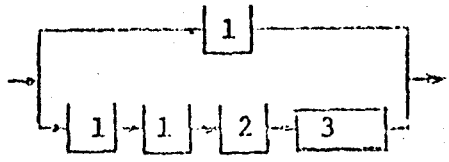
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right] + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_2 s}\right]^3$$

3640
15



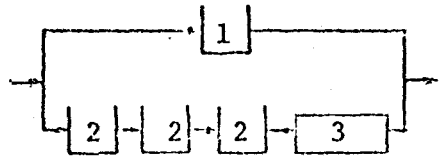
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right] + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^3\exp(-T_2 s)$$

3610
16



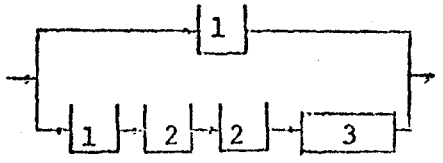
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2\left[\frac{1}{1+T_2 s}\right]\exp(-T_3 s)$$

3620
17



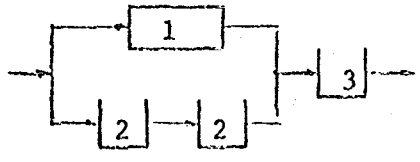
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^3\exp(-T_3 s)$$

3630
18



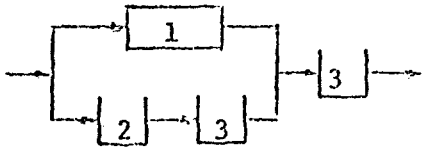
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_2 s}\right]^2\exp(-T_3 s)$$

3910
19



$$\left\{\frac{Q_1}{Q_T}\exp(-T_1 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]\right\}\left[\frac{1}{1+T_3 s}\right]$$

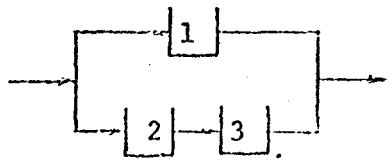
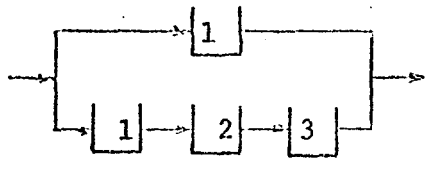
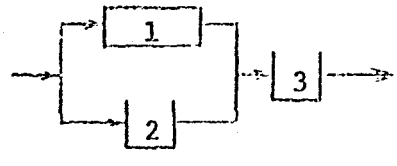
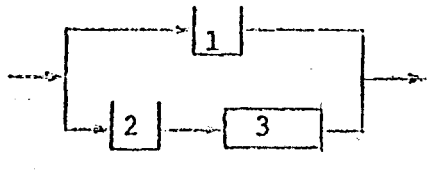
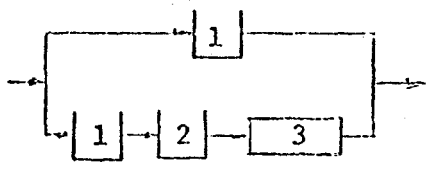
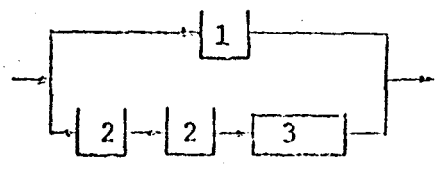
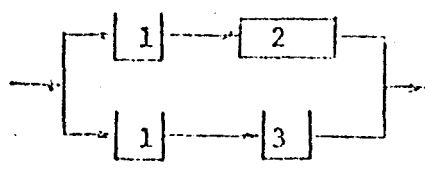
3920
20



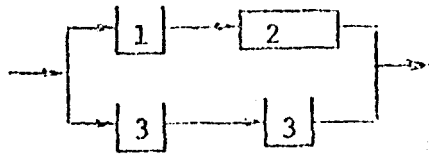
$$\left\{\frac{Q_1}{Q_T}\exp(-T_1 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]\left[\frac{1}{1+T_3 s}\right]\right\}\left[\frac{1}{1+T_3 s}\right]$$

CATEGORY 1 MODELS

4 PARAMETERS

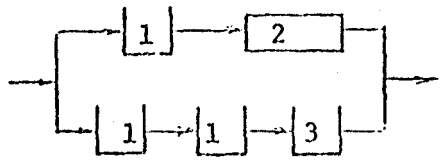
Model # & Code	Model Configuration	Transfer Function
1000 1		$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]\left[\frac{1}{1+T_3 s}\right]$
1310 2		$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_2 s}\right]\left[\frac{1}{1+T_3 s}\right]$
1900 3		$\left\{\frac{Q_1}{Q_T}\exp(-T_1 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]\right\}\left[\frac{1}{1+T_3 s}\right]$
2200 4		$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]\exp(-T_3 s)$
2510 5		$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_2 s}\right]\exp(-T_3 s)$
2520 6		$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^2\exp(-T_3 s)$
2710 7		$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]\exp(-T_2 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_3 s}\right]$

2720
8



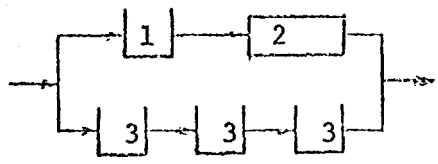
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]\exp(-T_2 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_3 s}\right]^2$$

3210
9



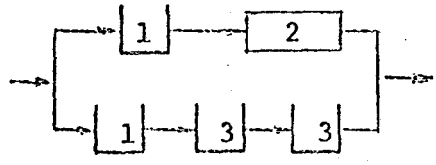
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]\exp(-T_2 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2\left[\frac{1}{1+T_3 s}\right]$$

3220
10



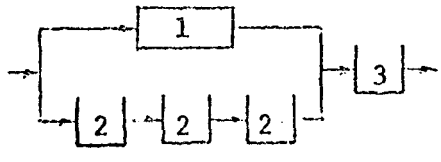
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]\exp(-T_2 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_3 s}\right]^3$$

3230
11



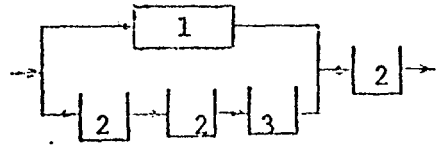
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]\exp(-T_2 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_3 s}\right]^2$$

3410
12



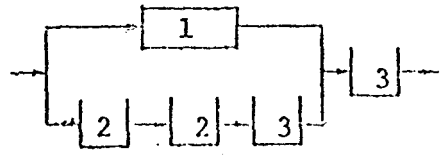
$$\left\{\frac{Q_1}{Q_t}\exp(-T_1 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^3\right\}\left[\frac{1}{1+T_3 s}\right]$$

3420
13



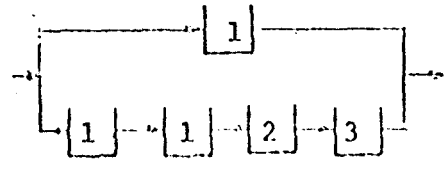
$$\left\{\frac{Q_1}{Q_T}\exp(-T_1 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^2\left[\frac{1}{1+T_3 s}\right]\right\}\left[\frac{1}{1+T_2 s}\right]$$

3430
14



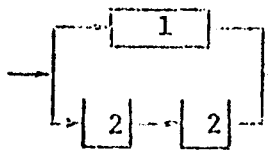
$$\left\{\frac{Q_1}{Q_T}\exp(-T_1 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^2\left[\frac{1}{1+T_3 s}\right]\right\}\left[\frac{1}{1+T_3 s}\right]$$

3510
15



$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2\left[\frac{1}{1+T_2 s}\right]\left[\frac{1}{1+T_3 s}\right]$$

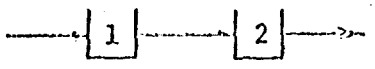
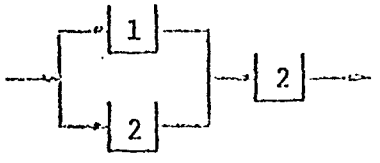
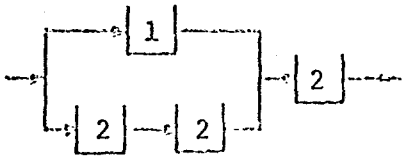
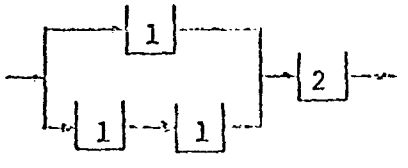
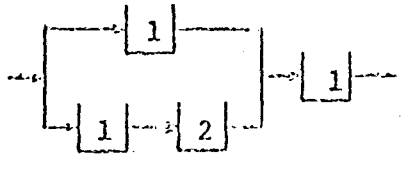
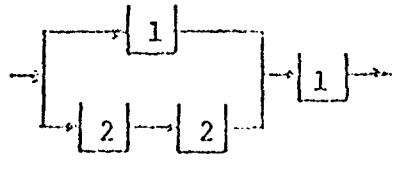
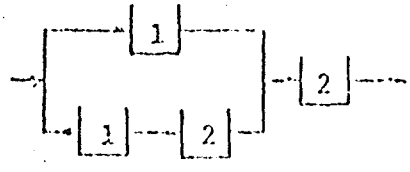
3930
16



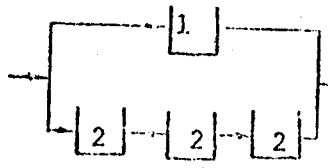
$$\left\{\frac{Q_1}{Q_T}\exp(-T_1 s) + \frac{(Q_T - Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^2\right\}\left[\frac{1}{1+T_2 s}\right]$$

CATEGORY 2 MODELS

3 PARAMETERS OR LESS

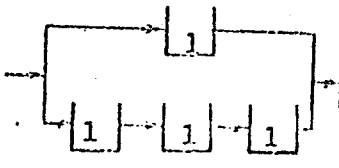
Model # & Code	Model Configuration	Transfer Function
0200 1		$\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_2 s}\right]$
0710 2		$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]\right\}\left[\frac{1}{1+T_2 s}\right]$
1150 3		$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^2\right\}\left[\frac{1}{1+T_2 s}\right]$
1160 4		$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2\right\}\left[\frac{1}{1+T_2 s}\right]$
1170 5		$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_2 s}\right]\right\}\left[\frac{1}{1+T_1 s}\right]$
1180 6		$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^2\right\}\left[\frac{1}{1+T_1 s}\right]$
1190 7		$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_2 s}\right]\right\}\left[\frac{1}{1+T_2 s}\right]$

1450
8



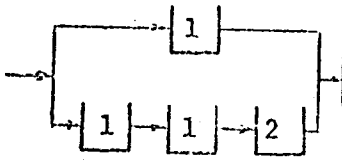
$$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^3\right\}\left[\frac{1}{1+T_2 s}\right]$$

1460
9



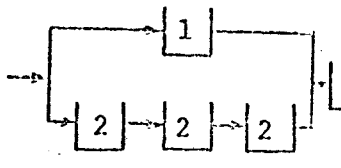
$$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^3\right\}\left[\frac{1}{1+T_2 s}\right]$$

1470
10



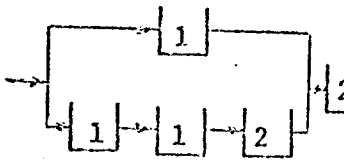
$$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2\left[\frac{1}{1+T_2 s}\right]\right\}\left[\frac{1}{1+T_1 s}\right]$$

1480
11



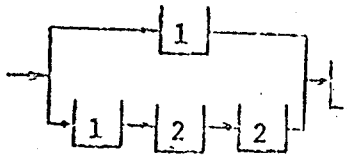
$$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^2\right\}\left[\frac{1}{1+T_1 s}\right]$$

1490
12



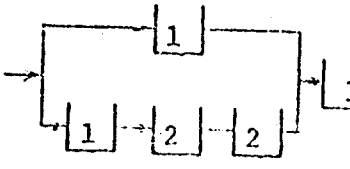
$$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2\left[\frac{1}{1+T_2 s}\right]\right\}\left[\frac{1}{1+T_2 s}\right]$$

1495
13



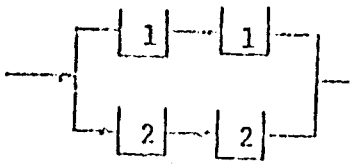
$$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_2 s}\right]^2\right\}\left[\frac{1}{1+T_2 s}\right]$$

1497
14



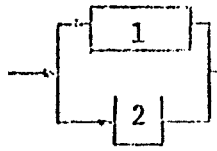
$$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\left[\frac{1}{1+T_2 s}\right]^2\right\}\left[\frac{1}{1+T_1 s}\right]$$

1510
15



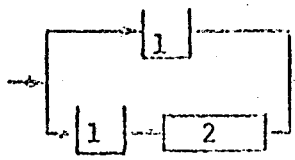
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^2$$

2020
16



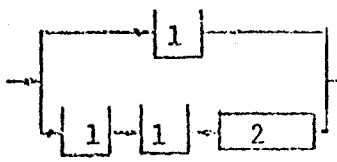
$$\left\{\frac{Q_1}{Q_T}\exp(-T_1 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]\right\}\left[\frac{1}{1+T_2 s}\right]^2$$

2330
17



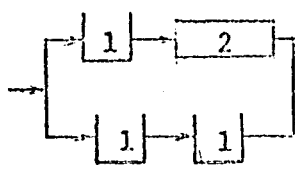
$$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]\exp(-T_2 s)\right\}\left[\frac{1}{1+T_1 s}\right]$$

2640
18



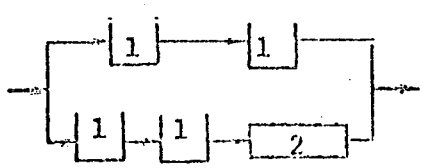
$$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2\exp(-T_2 s)\right\}\left[\frac{1}{1+T_1 s}\right]$$

2840
19



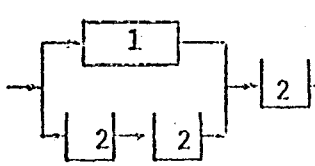
$$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]\exp(-T_2 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2\right\}\left[\frac{1}{1+T_1 s}\right]$$

2930
20



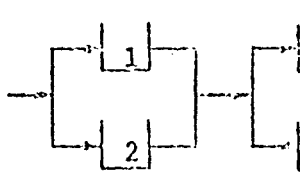
$$\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_1 s}\right]^2\exp(-T_2 s)$$

4030
21



$$\left\{\frac{Q_1}{Q_T}\exp(-T_1 s)+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]^2\right\}\left[\frac{1}{1+T_2 s}\right]^2$$

4110
22



$$\left\{\frac{Q_1}{Q_T}\left[\frac{1}{1+T_1 s}\right]+\frac{(Q_T-Q_1)}{Q_T}\left[\frac{1}{1+T_2 s}\right]\right\}^2$$

APPENDIX D

```
C       THIS PROGRAM USES THE SIMPLEX SEARCH TECHNIQUE TO FIND THE
C       PARAMETERS FOR THE BEST MIXING MODEL USING THE DATA FROM THE
C       FOURIER TRANSFORM PROGRAM
C
C       NTHETA - NO. OF DATA POINTS SUPPLIED
C       QTOT - NORMALIZED TOTAL FLOW (I.E. SET TO 1.0)
C       RUN - EXPERIMENTAL RUN IDENTIFICATION
C       ARRAY - ARRAY CONTAINING FREQUENCY AND REAL AND IMAG. PARTS OF
C       FOURIER TRANSFORM OF DATA
C       NJOBS - NO. OF DIFFERENT INITIAL GUESSES OF PARAMETER VALUES
C       NPUNCH - PUNCH CONTROL - 1 - PUNCHES DATA
C                              - 0 - DOES NOT PUNCH
C       N - NO. OF PARAMETERS
C       NPOLES - NO. OF POLES IN EXCESS OF ZEROS FOR MODEL CATEGORY
C       FMT - FORMAT ARRAY
C       IRES - RESULT ARRAY CONTAINING MIXING MODEL CONFIGURATION
C              NUMBERS
C       NPROBS - NO. OF MIXING MODELS TO BE FITTED
C       X - ARRAY CONTAINING VALUES OF PARAMETERS AT EACH VERTEX OF
C           SIMPLEX
C       Y - TEMPORARY STORAGE ARRAY FOR X
C       XX - ARRAY CONTAINING OPTIMUM PARAMETER VALUES
C       NJAM - CONTROL PARAMETER TO CHANGE NUMBER OF PARAMETERS
C       RLPRED - REAL PART OF PREDICTED RESPONSE
C       IMPRED - IMAG. PART OF PREDICTED RESPONSE
C       NSKIP - CONTROL PARAMETER - 4 - USES ONLY EVERY FOURTH DATA
C                                       POINT
C                                 - 1 - USES ALL DATA POINTS
C       RMS - RESIDUAL SUM OF SQUARES BETWEEN DATA POINTS AND MIXING
C             MODEL
C       RES - ARRAY CONTAINING RMS VALUES AND OPTIMUM PARAMETER VALUES
C
      COMMON ARRAY(160,3),NTHETA,NMODEL,S,PRED,QTOT,NCODE,NSKIP
      COMPLEX S,PRED
      REAL IMPRED
      DIMENSION X(10,11),Y(10,11),XX(10),FMT(16),RES(25,7),IRES(25)
      READ(5,120) NTHETA
      READ(5,130) QTOT
      READ(5,130) RUN
      READ(5,100) ((ARRAY(I,J),J=1,3),I=1,NTHETA)
      READ(5,120) NJOBS
      READ(5,120) NPUNCH
      READ(5,120) N
      READ(5,120) NPOLES
      N1=N+1
      READ(5,140) (FMT(I),I=1,16)
      DO 960 IJKL=1,NJOBS
      DO 800 I=1,25
      IRES(I)=0
  800 CONTINUE
      READ(5,120) NPROBS
      READ(5,130) ((X(J,I),J=1,N),I=1,N1)
      DO 910 I=1,N
      DO 910 J=1,N1
      Y(I,J)=X(I,J)
```

```
  910 CONTINUE
      DO 999 IJK=1,NPROBS
      DO 920 I=1,N
      DO 920 J=1,N1
      X(I,J)=Y(I,J)
  920 CONTINUE
      DO 930 I=1,5
      XX(I)=0.
  930 CONTINUE
      READ(5,120) NMODEL,NCODE,NJAM
      IF(NJAM.EQ.0) GO TO 950
      READ(5,120) N
      N1=N+1
      READ(5,140) (FMT(I),I=1,16)
      READ(5,130) ((X(J,I),J=1,N),I=1,N1)
      NFM=1
  950 CONTINUE
      WRITE(6,201) NMODEL,RUN
      NSKIP=4
C
C        ESTIMATE PARAMETERS BY SIMPLEX
C
      CALL SIMPLEX(N,X,1.E-08,700,1.,.5,2.,1)
      DO 900 I=1,N
      XX(I)=X(I,1)
  900 CONTINUE
      WRITE(6,201) NMODEL,RUN
      WRITE(6,200)
      IF(NPUNCH.NE.1) GO TO 540
      WRITE(7,110) (X(I,1),I=1,N)
  540 CONTINUE
      DO 530 I=1,NTHETA
      CALL CALC(RLPRED,IMPRED,I,XX)
      WRITE(6,210) (ARRAY(I,J),J=1,2),RLPRED,ARRAY(I,3),IMPRED
      IF(NPUNCH.NE.1) GO TO 530
      WRITE(7,270) ARRAY(I,2),RLPRED,ARRAY(I,3),IMPRED
  530 CONTINUE
      WRITE(6,FMT) (X(I,1),I=1,N)
      NSKIP=1
C
C        CALCULATE RESIDUAL SUM OF SQUARES
C
      CALL OBJECT(XX,RMS)
      WRITE(6,230) RMS
C
C        FILL ARRAY OF RESULTS
C
      IRES(NCODE)=NMODEL
      RES(NCODE,1)=RMS
      RES(NCODE,2)=XX(1)
      RES(NCODE,3)=XX(2)
      RES(NCODE,4)=XX(3)
      RES(NCODE,5)=XX(4)
      RES(NCODE,6)=XX(5)
      IF(NPROBS.EQ.1) GO TO 960
  999 CONTINUE
      WRITE(6,280) NPOLES,N,RUN
```

```
      WRITE(6,250)
      DO 810 I=1,25
      IF(IRES(I).EQ.0) GO TO 810
      WRITE(6,260) IRES(I),(RES(I,J),J=1,6)
  810 CONTINUE
      WRITE(6,240)
  960 CONTINUE
      STOP
  100 FORMAT(E10.3,50X,2E10.3)
  110 FORMAT(4E20.13)
  120 FORMAT(16I5)
  130 FORMAT(8F10.0)
  140 FORMAT(8A10)
  200 FORMAT(//15X,*W*/11X,*(RAD/HR)*,6X,*RL(OB)*,7X,*RL(PR)*,7X,*IM(OB
     1*,7X,*IM(PR)*/)
  201 FORMAT(1H1,10X,*PARAMETER ESTIMATION FOR MODEL *,I5//20X,*ON TIMPA
     1NY *,F4.1)
  210 FORMAT(7X,5E13.3)
  220 FORMAT(//7X,5E13.3)
  230 FORMAT(//11X,*RESIDUAL SUM OF SQUARES = *,E10.3)
  240 FORMAT(1H1)
  250 FORMAT(//10X,*MODEL*,7X,*RSS*,9X,*PAR1*,9X,*PAR2*,9X,*PAR3*,9X,*PA
     1R4*,9X,*PAR5*//)
  260 FORMAT(/10X,I5,6E13.3)
  270 FORMAT(4E10.3)
  280 FORMAT(1H1,10X,*PARAMETER ESTIMATION FOR N/N+*,I1,*, *,I1,* PARAME
     1TER MODELS*//25X,*ON TIMPANY *,F4.1)
      END


      SUBROUTINE OBJECT(Y,RMS)
C
C        THIS SUBROUTINE CALCULATES THE RESIDUAL SUM OF SQUARES BETWEEN
C        THE OBSERVED AND PREDICTED RESPONSE
C
C        S - COMPLEX LAPLACE TRANSFORM
C        RMS - RESIDUAL SUM OF SQUARES
C
      COMMON ARRAY(160,3),NTHETA,NMODEL,S,PRED,WTOT,NCODE,NSKIP
      COMPLEX S,PRED
      REAL IMPRED
      DIMENSION Y(10)
      RMS=0.
C
C        CALCULATE RESIDUAL SUM OF SQUARES
C
      DO 300 I=1,NTHETA,NSKIP
      S=CMPLX(0.,ARRAY(I,1))
      CALL MODEL(Y)
      RLPRED=REAL(PRED)
      IMPRED=AIMAG(PRED)
      RMS=RMS+(ARRAY(I,2)-RLPRED)**2+(ARRAY(I,3)-IMPRED)**2
  300 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CALC(RLPRED,IMPRED,I,XX)
C
C        THIS SUBROUTINE CALCULATES THE PREDICTED RESPONSE USING THE
C        OPTIMUM PARAMETER VALUES
C
C        S - COMPLEX LAPLACE TRANSFORM VARIABLE
C        RLPRED - REAL PART OF PREDICTED RESPONSE
C        IMPRED - IMAG. PART OF PREDICTED RESPONSE
C
      COMMON ARRAY(160,3),NTHETA,NMODEL,S,PRED,QTOT,NCODE,NSKIP
      COMPLEX S,PRED
      REAL IMPRED
      S=CMPLX(0.,ARRAY(I,1))
      CALL MODEL(XX)
      RLPRED=REAL(PRED)
      IMPRED=AIMAG(PRED)
      RETURN
      END


      SUBROUTINE MODEL (AKE)
C
C        THIS SUBROUTINE CALCULATES THE FREQUENCY RESPONSE FOR THE
C        DISPERSION MODEL WITH CLOSED BOUNDARY CONDITIONS
C
C        (ADAPTED FROM WILSON (W-1))
C
C        AKE - MODEL PARAMETERS
C        W - FREQUENCY
C        T - TAU
C        P - UL/D
C
      COMMON ARRAY(160,3),NTHETA,NMODEL,S,PRED,QTOT,NCODE,NSKIP
      COMPLEX S,PRED
      REAL IM
      DIMENSION AKE(5)
      W=AIMAG(S)
      T=AKE(1)
      P=1./AKE(2)
      AA = 4.0 * W * T / P
      BB = 0.5 * ATAN(AA)
      B = (1.0 + AA * * 2) * * 0.25 * COS(BB)
      D = (1.0 + AA * * 2) * * 0.25 * SIN(BB)
      ARG1 = P * B / 2.0
      ARG2 = P * D / 2.0
      CALL  HYPERS(ARG1, SINH)
      CALL  HYPERC(ARG1, COSH)
      TERM1 = (1.0 + B * * 2 - D * * 2) * SINH * COS(ARG2)
      TERM2 = -2.0 * B * D * COSH * SIN(ARG2)
      TERM3 = 2.0 * B * COSH * COS(ARG2)
      TERM4 = -2.0 * D * SINH * SIN(ARG2)
      ALPHA1 = TERM1 + TERM2 + TERM3 + TERM4
      TERM5 = 2.0 * B * D * SINH * COS(ARG2)
      TERM6 = (1.0 + B * * 2 - D * * 2) * COSH * SIN(ARG2)
      TERM7 = 2.0 * D * COSH * COS(ARG2)
      TERM8 = 2.0 * B * SINH * SIN(ARG2)
      ALPHA2 = TERM5 + TERM6 + TERM7 + TERM8
```

```
      TOP1 = 2.0 * EXP(P / 2.0) * (B * ALPHA1 + D * ALPHA2)
      TOP2 = 2.0 * EXP(P / 2.0) * (D * ALPHA1 - B * ALPHA2)
      BOT = ALPHA1 * * 2 + ALPHA2 * * 2
      RL = TOP1 / BOT
      IM = TOP2 / BOT
      PRED=CMPLX(RL,IM)
      RETURN
      END


      SUBROUTINE  HYPERS(Z, SINH)
C
C         (ADAPTED FROM WILSON (w-1))
C
C     REFERENCE - ABRAMOWITZ AND STEGUN, PAGE 85
      PROD = 1.0
      DO 300  K = 1, 100
      FLTK = FLOAT(K)
      BEFORE = PROD
      AFTER = PROD * (1.0 + Z * * 2 / (FLTK * * 2 * 3.14159 * * 2))
      PROD = AFTER
      DIFF = BEFORE - AFTER
      ADIFF = ABS(DIFF)
      IF (ADIFF.LE.0.001)  GO TO 400
  300 CONTINUE
  400 SINH = PROD * Z
      RETURN
      END


      SUBROUTINE  HYPERC(Z, COSH)
C
C         (ADAPTED FROM WILSON (w-1))
C
C     REFERENCE - ABRAMOWITZ AND STEGUN, PAGE 85
      PROD = 1.0
      DO 300  K = 1, 100
      FLTK = FLOAT(K)
      BEFORE = PROD
      AFTER = PROD * (1.0 + 4.0 * Z * * 2 / ((2.0 * FLTK - 1.0)* * 2
     1 * 3.14159 * * 2))
      PROD = AFTER
      DIFF = BEFORE - AFTER
      ADIFF = ABS(DIFF)
      IF (ADIFF.LE.0.001)  GO TO 400
  300 CONTINUE
  400 COSH = PROD
      RETURN
      END


      SUBROUTINE MODEL(AKE)
C
C         (ADAPTED FROM WILSON (w-1))
C
C         THIS SUBROUTINE CALCULATES THE FREQUENCY RESPONSE FOR N/N+1, 3
C         PARAMETER MODELS
```

```
C
C         AKE - MODEL PARAMETERS
C
      COMMON ARRAY(160,3),NTHETA,NMODEL,S,PRED,QTOT,NCODE,NSKIP
      COMPLEX S,W,PRED
      COMPLEX G0100,G0600,G1010,G1320,G1330,G1910,G2210,G2530,
     1G2730,G3240,G3440,G3520,G3530,G3540,G3640,G3930,G5900
      DIMENSION AKE(5)
C
C         DEFINE THE MODEL TRANSFER FUNCTIONS
C
      G0100(T1, W) = 1.0 / (1.0 + T1 * W)
      G0600(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T2 * W)
      G1010(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T1 * W) / (1.0 + T2 * W)
      G1320(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T1 * W) * * 2 / (1.0 + T2 * W)
      G1330(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T1 * W) / (1.0 + T2 * W) * * 2
      G1910(T1, T2, Q1, QT, W) = (Q1 / QT * CEXP(-T1 * W) +
     1(QT - Q1) / QT / (1.0 + T2 * W)) / (1.0 + T2 * W)
      G2210(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) + (QT - Q1) /
     1QT / (1.0 + T1 * W) * CEXP(-T2 * W)
      G2530(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) + (QT - Q1) /
     1QT / (1.0 + T1 * W) * * 2 * CEXP(-T2 * W)
      G2730(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) *
     1CEXP(-T2 * W) + (QT - Q1) / QT / (1.0 + T1 * W) * * 2
      G3240(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) *
     1CEXP(-T2 * W) + (QT - Q1) / QT / (1.0 + T1 * W) * * 3
      G3440(T1, T2, Q1, QT, W) = (Q1 / QT * CEXP(-T1 * W) + (QT - Q1) /
     1QT / (1.0 + T2 * W) * * 3) / (1.0 + T2 * W)
      G3520(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) + (QT - Q1) /
     1QT / (1.0 + T1 * W) * * 3 / (1.0 + T2 * W)
      G3530(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) + (QT - Q1) /
     1QT / (1.0 + T1 * W) * * 2 / (1.0 + T2 * W) * * 2
      G3540(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) + (QT - Q1) /
     1QT / (1.0 + T1 * W) / (1.0 + T2 * W) * * 3
      G3640(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) +
     1(QT -Q1) / QT / (1.0 + T1 * W) * * 3 * CEXP(-T2 * W)
      G3930(T1, T2, Q1, QT, W) = (Q1 / QT * CEXP(-T1 * W) +
     1(QT - Q1) / QT / (1.0 + T2 * W) * * 2) / (1.0 + T2 * W)
C
C         GO TO REQUIRED MODEL
C
      GO TO (0100,0600,1010,1320,1330,1910,2210,2530,2730,
     13240,3440,3520,3530,3540,3640,3930) NCODE
 0100 PRED=G0100(AKE(1),S)
      RETURN
 0600 PRED=G0600(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1010 PRED=G1010(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1320 PRED=G1320(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1330 PRED=G1330(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
```

```
1910 PRED=G1910(AKE(1),AKE(2),AKE(3),WTOT,S)
     RETURN
2210 PRED=G2210(AKE(1),AKE(2),AKE(3),WTOT,S)
     RETURN
2530 PRED=G2530(AKE(1),AKE(2),AKE(3),WTOT,S)
     RETURN
2730 PRED=G2730(AKE(1),AKE(2),AKE(3),WTOT,S)
     RETURN
3240 PRED=G3240(AKE(1),AKE(2),AKE(3),WTOT,S)
     RETURN
3440 PRED=G3440(AKE(1),AKE(2),AKE(3),WTOT,S)
     RETURN
3520 PRED=G3520(AKE(1),AKE(2),AKE(3),WTOT,S)
     RETURN
3530 PRED=G3530(AKE(1),AKE(2),AKE(3),WTOT,S)
     RETURN
3540 PRED=G3540(AKE(1),AKE(2),AKE(3),WTOT,S)
     RETURN
3640 PRED=G3640(AKE(1),AKE(2),AKE(3),WTOT,S)
     RETURN
3930 PRED=G3930(AKE(1),AKE(2),AKE(3),WTOT,S)
     RETURN
     END


     SUBROUTINE MODEL(AKE)
C
C        (ADAPTED FROM WILSON (W-1))
C
C        THIS SUBROUTINE CALCULATES THE FREQUENCY RESPONSE FOR N/N+1,
C        4 PARAMETER MODELS
C
C        AKE - MODEL PARAMETERS
C
     COMMON ARRAY(160,3),NTHETA,NMODEL,S,PRED,WTOT,NCODE,NSKIP
     COMPLEX S,W,PRED
C
C        DEFINE MODEL TRANSFER FUNCTIONS
C
     COMPLEX G1000,G1310,G1900,G2200,G2510,G2520,G2710,G2720,G3210,
    1G3220,G3230,G3410,G3420,G3430,G3510,G3610,G3620,G3630,G3910,G3920
     DIMENSION AKE(5)
     G1000(T1,T2,T3,W1,WT,W)=W1/WT/(1.+T1*W)+(WT-W1)/WT/(1.+T2*W)/(1.+T
    13*W)
     G1310(T1,T2,T3,W1,WT,W)=W1/WT/(1.+T1*W)+(WT-W1)/WT/(1.+T1*W)/(1.+T
    12*W)/(1.+T3*W)
     G1900(T1,T2,T3,W1,WT,W)=(W1/WT*CEXP(-T1*W)+(WT-W1)/WT/(1.+T2*W))/(
    11.+T3*W)
     G2200(T1,T2,T3,W1,WT,W)=W1/WT/(1.+T1*W)+(WT-W1)/WT/(1.+T2*W)*CEXP(
    1-T3*W)
     G2510(T1,T2,T3,W1,WT,W)=W1/WT/(1.+T1*W)+(WT-W1)/WT/(1.+T1*W)/(1.+T
    12*W)*CEXP(-T3*W)
     G2520(T1,T2,T3,W1,WT,W)=W1/WT/(1.+T1*W)+(WT-W1)/WT/(1.+T2*W)**2*CE
    1XP(-T3*W)
     G2710(T1,T2,T3,W1,WT,W)=W1/WT/(1.+T1*W)*CEXP(-T2*W)+(WT-W1)/WT/(1.
    1+T1*W)/(1.+T3*W)
     G2720(T1,T2,T3,W1,WT,W)=W1/WT/(1.+T1*W)*CEXP(-T2*W)+(WT-W1)/WT/(1.
```

```
     1+T3*W)**2
      G3210(T1,T2,T3,Q1,QT,W)=Q1/QT/(1.+T1*W)*CEXP(-T2*W)+(QT-Q1)/QT/(1.
     1+T1*W)**2/(1.+T3*W)
      G3220(T1,T2,T3,Q1,QT,W)=Q1/QT/(1.+T1*W)*CEXP(-T2*W)+(QT-Q1)/QT/(1.
     1+T3*W)**3
      G3230(T1,T2,T3,Q1,QT,W)=Q1/QT/(1.+T1*W)*CEXP(-T2*W)+(QT-Q1)/QT/(1.
     1+T1*W)/(1.+T3*W)**2
      G3410(T1,T2,T3,Q1,QT,W)=(Q1/QT*CEXP(-T1*W)+(QT-Q1)/QT/(1.+T2*W)**3
     1)/(1.+T3*W)
      G3420(T1,T2,T3,Q1,QT,W)=(Q1/Q1*CEXP(-T1*W)+(QT-Q1)/QT/(1.+T2*W)**2
     1/(1.+T3*W))/(1.+T2*W)
      G3430(T1,T2,T3,Q1,QT,W)=(Q1/QT*CEXP(-T1*W)+(QT-Q1)/QT/(1.+T2*W)**2
     1/(1.+T3*W))/(1.+T3*W)
      G3510(T1,T2,T3,Q1,QT,W)=Q1/QT/(1.+T1*W)+(QT-Q1)/QT/(1.+T1*W)**2/(1
     1.+T2*W)/(1=+T3*W)
      G3610(T1,T2,T3,Q1,QT,W)=Q1/Q1/(1.+T1*W)+(QT-Q1)/Q1/(1.+T1*W)**2/(1
     1.+T2*W)*CEXP(-T3*W)
      G3620(T1,T2,T3,Q1,QT,W)=Q1/Q1/(1.+T1*W)+(QT-Q1)/Q1/(1.+T2*W)**3*CE
     1XP(-T3*W)
      G3630(T1,T2,T3,Q1,QT,W)=Q1/QT/(1.+T1*W)+(QT-Q1)/QT/(1.+T1*W)/(1.+T
     12*W)**2*CEXP(-T3*W)
      G3910(T1,T2,T3,Q1,QT,W)=(Q1/QT*CEXP(-T1*W)+(QT-Q1)/QT/(1.+T2*W)**2
     1)/(1.+T3*W)
      G3920(T1,T2,T3,Q1,QT,W)=(Q1/QT*CEXP(-T1*W)+(QT-Q1)/QT/(1.+T2*W)/(1
     1.+T3*W))/(1.+T3*W)
C
C         GO TO REQUIRED MODEL
C
      GO TO (1000,1310,1900,2200,2510,2520,2710,2720,3210,3220,3230,3410
     1,3420,3430,3510,3610,3620,3630,3910,3920) NCODE
 1000 PRED=G1000(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 1310 PRED=G1310(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 1900 PRED=G1900(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 2200 PRED=G2200(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 2510 PRED=G2510(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 2520 PRED=G2520(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 2710 PRED=G2710(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 2720 PRED=G2720(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 3210 PRED=G3210(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 3220 PRED=G3220(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 3230 PRED=G3230(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 3410 PRED=G3410(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 3420 PRED=G3420(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 3430 PRED=G3430(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
```

```
      RETURN
 3510 PRED=G3510(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 3610 PRED=G3610(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 3620 PRED=G3620(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 3630 PRED=G3630(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 3910 PRED=G3910(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
 3920 PRED=G3920(AKE(1),AKE(2),AKE(3),AKE(4),QTOT,S)
      RETURN
      END


      SUBROUTINE MODEL(AKE)
C
C        (ADAPTED FROM WILSON (W-1))
C
C        THIS SUBROUTINE CALCULATES THE FREQUENCY RESPONSES FOR N/N+2,
C        3 PARAMETER MODELS
C
C        AKE - MODEL PARAMETERS
C
      COMMON ARRAY(160,3),NTHETA,NMODEL,S,PRED,QTOT,NCODE,NSKIP
      COMPLEX S,W,PRED
      COMPLEX G0200,G0710,G1150,G1160,G1170,G1180,G1190,G1450,G1460,G147
     10,G1480,G1490,G1495,G1497,G1510,G2020,G2330,G2640,G2840,G2930,G403
     10,G4110
      DIMENSION AKE(5)
C
C        DEFINE THE MODEL TRANSFER FUNCTIONS
C
      G0200(T1, T2, W) = 1.0 / (1.0 + T1 * W) / (1.0 + T2 * W)
      G0710(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T2 * W)) / (1.0 + T2 * W)
      G1150(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T2 * W) * * 2) / (1.0 + T2 * W)
      G1160(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T1 * W) * * 2) / (1.0 + T2 * W)
      G1170(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T1 * W) / (1.0 + T2 * W)) / (1.0 + T1 * W)
      G1180(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T2 * W) * * 2) / (1.0 + T1 * W)
      G1190(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T1 * W) / (1.0 + T2 * W)) / (1.0 + T2 * W)
      G1450(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T2 * W) * * 3) / (1.0 + T2 * W)
      G1460(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T1 * W) * * 3) / (1.0 + T2 * W)
      G1470(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T1 * W) * * 2 / (1.0 + T2 * W)) /
     2(1.0 + T1 * W)
      G1480(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T2 * W) * * 3) / (1.0 + T1 * W)
      G1490(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
```

```
     1(QT - Q1) / QT / (1.0 + T1 * W) * * 2 / (1.0 + T2 * W)) /
     2(1.0 + T2 * W)
      G1495(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T1 * W) / (1.0 + T2 * W) * * 2) /
     2(1.0 + T2 * W)
      G1497(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) + (QT - Q1) /
     1QT / (1.0 + T1 * W) / (1.0 + T2 * W) * * 2) / (1.0 + T1 * W)
      G1510(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) * * 2 +
     1(QT - Q1) / QT / (1.0 + T2 * W) * * 2
      G2020(T1, T2, Q1, QT, W) = (Q1 / QT * CEXP(-T1 * W) + (QT - Q1) /
     1QT / (1.0 + T2 * W)) / (1.0 + T2 * W) * * 2
      G2330(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T1 * W) * CEXP(-T2 * W)) / (1.0 + T1 * W)
      G2640(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T1 * W) * * 2 * CEXP(-T2 * W)) /
     2(1.0 + T1 * W)
      G2840(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) *
     1CEXP(-T2 * W) + (QT - Q1) / QT / (1.0 + T1 * W) * * 2) /
     2(1.0 + T1 * W)
      G2930(T1, T2, Q1, QT, W) = Q1 / QT / (1.0 + T1 * W) * * 2 +
     1(QT - Q1) / QT / (1.0 + T1 * W) * * 2 * CEXP(-T2 * W)
      G4030(T1, T2, Q1, QT, W) = (Q1 / QT * CEXP(-T1 * W) +
     1(QT - Q1) / QT / (1.0 + T2 * W) * * 2)/ (1.0 + T2 * W) * * 2
      G4110(T1, T2, Q1, QT, W) = (Q1 / QT / (1.0 + T1 * W) +
     1(QT - Q1) / QT / (1.0 + T2 * W)) * * 2
C
C         GO TO REQUIRED MODEL
C
      GO TO (0200,0710,1150,1160,1170,1180,1190,1450,1460,1470,1480,149
     1,1495,1497,1510,2020,2330,2640,2840,2930,4030,4110) NCODE
 0200 PRED=G0200(AKE(1),AKE(2),S)
      RETURN
 0710 PRED=G0710(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1150 PRED=G1150(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1160 PRED=G1160(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1170 PRED=G1170(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1180 PRED=G1180(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1190 PRED=G1190(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1450 PRED=G1450(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1460 PRED=G1460(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1470 PRED=G1470(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1480 PRED=G1480(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1490 PRED=G1490(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1495 PRED=G1495(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 1497 PRED=G1497(AKE(1),AKE(2),AKE(3),QTOT,S)
```

```
      RETURN
 1510 PRED=G1510(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 2020 PRED=G2020(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 2330 PRED=G2330(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 2640 PRED=G2640(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 2840 PRED=G2840(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 2930 PRED=G2930(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 4030 PRED=G4030(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
 4110 PRED=G4110(AKE(1),AKE(2),AKE(3),QTOT,S)
      RETURN
      END


      SUBROUTINE SIMPLEX(N,X,EPS,NMAX,ALPHA,BETA,GAMA,NWRIT)
      DIMENSION X(10,11),F(11),Y(10),XR(10),XH(10),XO(10),XS(10),XL(10)
     K,XE(10),XC(10)
C
C*****************************************************************************
C
C  OPTIMIZATION BY THE SIMPLEX METHOD
C
C*****************************************************************************
C  NOTATION AND ALGORITHM IS THE SAME AS THAT FOUND IN THE BOOK--
C
C     METHODS FOR UNCONSTRAINED OPTIMIZATION PROBLEMS BY J.KOWALIK AND
C     M.R.OSBORNE ELSEVIER,NY(1968)
C  EXCEPT FOR SLIGHT CHANGES IN NOTATION IT IS THE SAME AS THAT FOUND
C     IN THE PAPER BY NELDER AND MEAD (COMPUTER J.,VOL7,308(1965)
C
C
C
C  NWRIT -  PRINT CONTROL
C  XH -  THE HIGHEST VERTEX (FUNCTION VALUE =FXH)
C  XS -  THE SECOND HIGHEST VERTEX (FUNCTION VALUE = FXS)
C  XL -  THE LOWEST VERTEX (FUNCTION VALUE = FXL)
C  XO IS THE CENTROID OF ALL X EXCLUDING XH
C  VALUES OF ALPHA, BETA AND GAMA ARE SET IN PARAMETER LIST
C  FOR REFLECTION, CONTRACTION AND EXPANSION RESPECTIVELY
      AL1=ALPHA+1.
      BET1=1.-BETA
      GAM1=1.-GAMA
C  MAXIMUM NUMBER OF FUNCTION EVALUATIONS IS SET
C  NUMBER OF POINTS IN SIMPLEX IS N1=1+N
C
C     EVALUATE INITIAL SIMPLEX
C
C  X(J,I)=X(CO-ORDINATE,POINT)
C  EVALUATE FUNCTION AT EACH X(J,I)
      WRITE(6,901)
  901 FORMAT (////* OPTIMIZATION USING THE SIMPLEX METHOD *////)
```

```
      FUN1=1.E06
      FUN2=1.E6
      NTRY=0
      N1=N+1
      RN=N
21    DO 2 I = 1,N1
      DO 22 J = 1,N
      Y(J)=X(J,I)
 2    CONTINUE
      NTRY=NTRY+1
      CALL OBJECT(Y,F(I))
      CONTINUE
      IF (NWRIT .LE. 0) GO TO 1001
      WRITE(6,605)
605   FORMAT(1X,*X VALUES*)
      DO 51 IL=1,N1
      WRITE(6,603)(X(IM,IL),IM=1,N)
51    CONTINUE
603   FORMAT(1X,10E12.5)
      WRITE(6,606)
606   FORMAT(1X,*F VALUES*)
      WRITE(6,603)(F(KK),KK=1,N1)
 1001 CONTINUE
      DO 36 I = 1,N
      XR(I)=0.
      XO(I)=0.
      XE(I)=0.
      XS(I)=0.
      XH(I)=0.
      XC(I)=0.
      XL(I)=0.
36    CONTINUE
      GO TO 24
   11 IF(NTRY.GE.NMAX) GO TO 23
   24 FXH = F(1)
      IXH=1
      DO 3 I = 2,N1
      IF(F(I).LT.FXH)GO TO 3
      FXH=F(I)
      IXH=I
3     CONTINUE
      DO 101 I = 1,N1
      IF (I .EQ. IXH) GO TO 101
      FXS = F(I)
      GO TO 102
  101 CONTINUE
  102 CONTINUE
      DO 4 I = 1,N1
      IF(F(I).LT.FXS)GO TO 4
      IF(I.EQ.IXH)GO TO 4
      FXS=F(I)
      IXS=I
4     CONTINUE
      FXL=F(1)
      IXL=1
      DO 5 I = 2,N1
      IF(F(I).GT.FXL)GO TO 5
```

```
      FXL=F(I)
      IXL=I
5     CONTINUE
      FSTP=ABS(FUN2-FXL)
      FUN2=FXL
C
C     PRINT ONLY IF THERE IS AN IMPROVEMENT IN MINIMUM VALUE
C
      IF(FSTP.LT.1.E-6) GO TO 777
      IF(NWRIT.LE.1) GO TO 1002
      WRITE(6,610) FSTOP,NTRY
610   FORMAT(1H0,*ERROR  *,E12.5, *   FUN EVAL NO  *,I5)
      WRITE(6,605)
      DO 6010  IL=1,N1
      WRITE(6,603) (X(IM,IL),IM=1,N)
6010  CONTINUE
      WRITE(6,606)
      WRITE(6,603) (F(KK),KK=1,N1)
 1002 CONTINUE
  777 DO 8 I=1,N
      XH(I)=X(I,IXH)
      XS(I)=X(I,IXS)
      XL(I)=X(I,IXL)
8     CONTINUE
C  CALCULATE XO
C  CALCULATE XH,XS,AND XL
      DO 7 J = 1,N
      SUM=0.
      DO 6 I = 1,N1
      IF(I.EQ.IXH)GO TO 6
      SUM=SUM+X(J,I)
6     CONTINUE
      XO(J)=SUM/RN
7     CONTINUE
      NTRY=NTRY+1
      CALL OBJECT(XO,FXO)
      FSTOP=0.
      DO 810 I=1,N1
      FSTOP=FSTOP+(F(I)-FXO)**2
  810 CONTINUE
      FSTOP=SQRT(FSTOP/RN)
      IF(NTRY.GE.NMAX)GO TO 23
      IF(FSTOP.LT.EPS) GO TO 23
C REFLECTION
      DO 31 I = 1,N
      XR(I)=AL1*XO(I)-ALPHA*XH(I)
31    CONTINUE
      NTRY=NTRY+1
      CALL OBJECT(XR,FXR)
      IF(NTRY.GE.NMAX)GO TO 23
      IF((FXS.GE.FXR).AND.(FXR.GE.FXL))GO TO 9
      IF(FXR.LT.FXL)GO TO 12
      IF((FXH.GT.FXR).AND.(FXR.GT.FXS))GO TO 15
C  CONTRACTION
35    DO 33 I = 1,N
      XC(I)=BETA*XH(I)+BET1*XO(I)
33    CONTINUE
```

```
      NTRY=NTRY+1
      CALL OBJECT(XC,FXC)
      IF(NTRY.GE.NMAX)GO TO 23
      IF(FXH.GT.FXC)GO TO 17
C  SHRINKING OF SIMPLEX
      WRITE(6,650)(XL(I),I=1,N)
650   FORMAT(1H0,*SHRINK SIMPLEX,XL =    *,3E12.5)
      DO 20 I = 1,N1
      DO 20 J = 1,N
      IF(I.EQ.IXL) GO TO 800
      X(J,I)=.5*(X(J,I)+XL(J))
  800 CONTINUE
20    CONTINUE
      GO TO 21
C REPLACE XH BY XR AND RESTART PROCESS
9     DO 10 J = 1,N
      X(J,IXH)=XR(J)
      XH(J)=XR(J)
10    CONTINUE
      F(IXH)=FXR
      GO TO 11
C  EXPANSION
12    DO 32 I=1,N
      XE(I)=GAMA*XR(I)+GAM1*XO(I)
32    CONTINUE
      NTRY=NTRY+1
      CALL OBJECT(XE,FXE)
      IF(FXL.GT.FXE)GO TO 13
C  EXPANSION UNSUCCESSFUL SO GO TO 9 TO REPLACE XH BY XR
      GO TO 9
C  EXPANSION SUCCESSFUL, REPLACE XH BY XE
13    DO 14 J = 1,N
      X(J,IXH)=XE(J)
      XH(J)=XE(J)
14    CONTINUE
      F(IXH)=FXE
      GO TO 11
C  REPLACE XH BY XR AND CONTRACT
15    DO 16 J=1,N
      XH(J)=XR(J)
      X(J,IXH)=XR(J)
16    CONTINUE
      F(IXH)=FXR
      FXH=FXR
C  CONTRACTION IS AT STATEMENT NO 35
      GO TO 35
C  CONTRACTION WAS SUCESSFUL
17    DO 18 J = 1,N
      X(J,IXH)=XC(J)
      XH(J)=XC(J)
18    CONTINUE
      F(IXH)=FXC
      GO TO 11
23    WRITE(6,600)
600   FORMAT(1H0,*CONCLUSION OF SEARCH*)
      IF (NWRIT .LE. 0) GO TO 1000
      WRITE(6,605)
```

```
      DO 54 IL=1,N1
      WRITE(6,603)(X(IM,IL),IM=1,N )
 54   CONTINUE
      WRITE(6,606)
      WRITE(6,603)(F(KK),KK=1,N1)
      WRITE(6,700)
602   FORMAT(1H ,7E12.4)
700   FORMAT(1H0,*XH,XS,XL,XR,XE,XC,XO*//)
      WRITE(6,602)(XH(K),XS(K),XL(K),XR(K),XE(K),XC(K),XO(K),K=1,N)
 1000 CONTINUE
      WRITE(6,601)NTRY,NMAX,FSTOP,EPS
601   FORMAT(1H0,*NO. OF FUNCTION EVALUATIONS =  *,I5//,
     1  *  MAXIMUM PERMITTED  =  *,I5//,
     1  *  ERROR  =  *,E20.5//,
     1  *  MAXIMUM ERROR REQUIRED  =  *,E20.5)
      DO 820 I=1,N
      X(I,1)=XL(I)
  820 CONTINUE
      RETURN
      END


C         THIS PROGRAM CALCULATES THE TOTAL RESIDENCE TIME FOR THE MIXIN
C         MODELS USING THE PARAMETER VALUES FROM THE SIMPLEX MIXING MODEL
C         FIT
C.
C         NJOBS - NO. OF MODELS TO BE DONE
C         NMODEL - MIXING MODEL CONFIGURATION NUMBER
C         NCODE - MIXING MODEL CODE NUMBER
C         T1 - TAU1
C         T2 - TAU2
C         T3 - TAU3
C         Q - Q1/QT
C         TT - TOTAL RESIDENCE TIME
C
      READ(5,120) NJOBS
      WRITE(6,220)
      DO 399 I=1,NJOBS
      READ(5,110) NMODEL,NCODE,RUN
C
C         GO TO REQUIRED MODEL
C
      GO TO (2730,3240,3220,1460,1490,1190,1150) NCODE
 3220 CONTINUE
      READ(5,100) T1,T2,T3,Q
C
C         CALCULATE TAUTOT
C
      TT=Q*(T1+T2)+(1.-Q)*3.*T3
C
C         NORMALIZE TAUS
C
      T1=T1/TT
      T2=T2/TT
      T3=T3/TT
      WRITE(6,200) TT,NMODEL,RUN
      WRITE(6,230) T1,T2,T3,Q
```

```
      WRITE(7,210) TT,NMODEL,RUN
      GO TO 399
 3240 CONTINUE
      READ(5,100) T1,T2,Q
C
C        CALCULATE TAUTOT
C
      TT=Q*(T1+T2)+(1.-Q)*3.*T1
C
C        NORMALIZE TAUS
C
      T1=T1/TT
      T2=T2/TT
      WRITE(6,200) TT,NMODEL,RUN
      WRITE(6,240) T1,T2,Q
      WRITE(7,210) TT,NMODEL,RUN
      GO TO 399
 2730 CONTINUE
      READ(5,100) T1,T2,Q
C
C        CALCULATE TAUTOT
C
      TT=Q*(T1+T2)+(1.-Q)*2.*T1
C
C        NORMALIZE TAUS
C
      T1=T1/TT
      T2=T2/TT
      WRITE(6,200) TT,NMODEL,RUN
      WRITE(6,240) T1,T2,Q
      WRITE(7,210) TT,NMODEL,RUN
      GO TO 399
 1460 CONTINUE
      READ(5,100) T1,T2,Q
C
C        CALCULATE TAUTOT
C
      TT=Q*T1+(1.-Q)*3.*T1+T2
C
C        NORMALIZE TAUS
C
      T1=T1/TT
      T2=T2/TT
      WRITE(6,200) TT,NMODEL,RUN
      WRITE(6,240) T1,T2,Q
      WRITE(7,210) TT,NMODEL,RUN
      GO TO 399
 1490 CONTINUE
      READ(5,100) T1,T2,Q
C
C        CALCULATE TAUTOT
C
      TT=Q*T1+(1.-Q)*(2.*T1+T2)+T2
C
C        NORMALIZE TAUS
C
      T1=T1/TT
```

```
      T2=T2/TT
      WRITE(6,200) TT,NMODEL,RUN
      WRITE(6,240) T1,T2,Q
      WRITE(7,210) TT,NMODEL,RUN
      GO TO 399
 1190 CONTINUE
      READ(5,100) T1,T2,Q
C
C        CALCULATE TAUTOT
C
      TT=Q*T1+(1.-Q)*(T1+T2)+T2
C
C        NORMALIZE TAUS
C
      T1=T1/TT
      T2=T2/TT
      WRITE(6,200) TT,NMODEL,RUN
      WRITE(6,240) T1,T2,Q
      WRITE(7,210) TT,NMODEL,RUN
      GO TO 399
 1150 CONTINUE
      READ(5,100) T1,T2,Q
C
C        CALCULATE TAUTOT
C
      TT=Q*T1+(1.-Q)*2.*T2+T2
C
C        NORMALIZE TAUS
C
      T1=T1/TT
      T2=T2/TT
      WRITE(6,200) TT,NMODEL,RUN
      WRITE(6,240) T1,T2,Q
      WRITE(7,210) TT,NMODEL,RUN
      IF(I.EQ.5) WRITE(6,220)
  399 CONTINUE
      STOP
  100 FORMAT(4E20.13)
  110 FORMAT(2I5,A10)
  120 FORMAT(16I5)
  200 FORMAT(//10X,*TAUTOT = *,E10.3,*  FOR MODEL*,I5,*  ON  *,A10//)
  210 FORMAT(E20.13,I5,A10)
  220 FORMAT(1H1)
  230 FORMAT(/15X,*NORMALIZED TAUS - TAU1 = *,E10.3,3X,*TAU2 = *,E10.3,3
     1X,*TAU3 = *,E10.3//15X,*FLOW SPLIT - Q1/QT = *,E10.3/)
  240 FORMAT(/15X,*NORMALIZED TAUS - TAU1 = *,E10.3,3X,*TAU2 = *,E10.3//
     115X,*FLOW SPLIT - Q1/QT = *,E10.3/)
      END


C        THIS PROGRAM PLOTS POLAR GRAPHS USING DATA FROM THE SIMPLEX
C        MIXING MODEL FIT.  THE RAW DATA IS PLOTTED AS POINTS AND THE
C        FITTED MODELS ARE PLOTTED AS LINES.  THE FIRST IS A SOLID LINE
C        WHILE THE SUCCEEDING LINES ARE DASHED LINES.
C
C        NJOBS - NO. OF PLOTS TO BE DONE
C        Z - PARAMETER FOR 'X' GRADUATIONS
```

```
C         Y - PARAMETER FOR 'Y' GRADUATIONS
C         NFR - FREQUENCY OF DOTS IN DASHED LINE
C         NPROBS - NO. OF MODELS TO BE FITTED TO ONE SET OF POINTS
C         XX - PARAMETER FOR LEGEND
C         YY - PARAMETER FOR LEGEND
C         NMODEL - MODEL CONFIGURATION NUMBER
C         X - ARRAY CONTAINING REAL AND IMAG. PARTS OF BOTH DATA POINTS
C             AND MODELS
C         RL - REAL PART IN DATA UNITS
C         IM - IMAG. PART IN DATA UNITS
C         PRL - REAL PART IN PLOTTER UNITS
C         PIM - IMAG. PART IN PLOTTER UNITS
C         YL - PARAMETER FOR LEGEND
C
      DIMENSION X(101,4),RL(101),PRL(101),PIM(101)
      DIMENSION B(101),S(101)
      DIMENSION XX(2),YY(2)
      REAL IM(101)
      REAL NMODEL
      READ(5,110) XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX
C
C         SET UP DATA SCALING
C
      CALL PLTIN(XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX)
      CALL DATE(THEDATE)
C
C         PLOT NAME AND DATE
C
      CALL LETTER(8,.24,90.,.5,1.,8HCRAWFORD)
      CALL LETTER(10,.24,90.,1.,1.,THEDATE)
      CALL PLOT(1.5,0.,-3)
      READ(5,120) NJOBS
      DO 399 IJK=1,NJOBS
C
C         PLOT AXES
C
      CALL ARROW(2.,6.,10.5,6.,3)
      CALL ARROW(4.5,8.5,4.5,2.,2)
C
C         PLOT ,X, GRADUATIONS
C
      Z=2.5
      DO 600 I=1,4
      CALL PLOT(Z,6.05,3)
      CALL PLOT(Z,5.95,2)
      Z=Z+.5
  600 CONTINUE
      Z=5.
      DO 610 I=1,10
      CALL PLOT(Z,6.05,3)
      CALL PLOT(Z,5.95,2)
      Z=Z+.5
  610 CONTINUE
C
C         PLOT ,Y, GRADUATIONS
C
      Y=8.
```

```
      DO 620 I=1,4
      CALL PLOT(4.45,Y,3)
      CALL PLOT(4.55,Y,2)
      Y=Y-.5
  620 CONTINUE
      Y=5.5
      DO 630 I=1,7
      CALL PLOT(4.45,Y,3)
      CALL PLOT(4.55,Y,2)
      Y=Y-.5
  630 CONTINUE
C
C        NUMBER AXES
C
      CALL LETTER(3,.1,0.,4.6,7.95,3H0.4)
      CALL LETTER(3,.1,0.,4.6,6.95,3H0.2)
      CALL LETTER(3,.1,0.,5.35,6.1,3H0.2)
      CALL LETTER(3,.1,0.,6.35,6.1,3H0.4)
      CALL LETTER(3,.1,0.,7.35,6.1,3H0.6)
      CALL LETTER(3,.1,0.,8.35,6.1,3H0.8)
      CALL LETTER(3,.1,0.,9.35,6.1,3H1.0)
C
C        LABEL AXES
C
      CALL LETTER(9,.15,0.,3.825,8.7,9HIMAGINARY)
      CALL LETTER(4,.15,0.,10.7,5.95,4HREAL)
      NFR=0
      READ(5,120) NPROBS
      XX(1)=6.
      XX(2)=7.
      YY(1)=YY(2)=7.75
      DO 398 K=1,NPROBS
      READ(5,130) NMODEL
      READ(5,100) ((X(I,J),J=1,4),I=1,101)
      IF(K.NE.1) GO TO 800
C
C        PLOT DATA POINTS
C
      DO 510 I=1,101,4
      RL(I)=X(I,1)
      IM(I)=X(I,3)
      CALL UNITTO(RL(I),IM(I),PRL(I),PIM(I))
      CALL GRAF(PRL(I),PIM(I),.07,3HCIR)
  510 CONTINUE
C
C        PLOT DATA POINTS IN LEGEND
C
      DO 810 I=1,3
      CALL GRAF(XX(1),YY(1),.07,3HCIR)
      XX(1)=XX(1)+.5
  810 CONTINUE
      XX(1)=6.
      YY(1)=YY(2)=7.5
C
C        LABEL DATA POINTS IN LEGEND
C
      CALL LETTER(13,.15,0.,7.12,7.675,13H- DATA POINTS)
```

```
  800 CONTINUE
C
C        CONVERT FROM DATA UNITS TO PLOTTER UNITS
C
      DO 500 I=1,101
      RL(I)=X(I,2)
      IM(I)=X(I,4)
      CALL UNITTO(RL(I),IM(I),PRL(I),PIM(I))
  500 CONTINUE
      IF(K.NE.1) GO TO 700
C
C        PLOT SOLID LINE
C
      CALL PLTMPL(RL,IM,101)
C
C        PLOT SOLID LINE IN LEGEND
C
      CALL PLOT(XX(1),YY(1),3)
      CALL PLOT(XX(2),YY(2),2)
C
C        LABEL SOLID LINE IN LEGEND
C
      CALL LETTER(2,.15,0.,7.12,7.425,2H- )
      CALL LETTER(10,.15,0.,7.42,7.425,NMODEL)
      YY(1)=YY(2)=7.25
      GO TO 398
  700 CONTINUE
C
C        PLOT DASHED LINE
C
      CALL DDASHM(PRL,PIM,101,.5,0.,NFR,.5,.1,IE,B,S)
C
C        PLOT DASHED LINE IN LEGEND
C
      CALL DDASHM(XX,YY,2,.5,0.,NFR,.5,.1,IE,B,S)
      YL=7.675-FLOAT(K)*.25
C
C        LABEL DASHED LINE IN LEGEND
C
      CALL LETTER(2,.15,0.,7.12,YL,2H- )
      CALL LETTER(10,.15,0.,7.42,YL,NMODEL)
      YY(1)=YY(2)=7.75-FLOAT(K+1)*.25
      NFR=K-1
  398 CONTINUE
      CALL PLOT(12.,0.,-3)
  399 CONTINUE
      CALL PLOT(12.,0.,999)
      STOP
  100 FORMAT(4E10.3)
  110 FORMAT(8F10.0)
  120 FORMAT(5I5)
  130 FORMAT(8A10)
      END
```

```
C       THIS PROGRAM GENERATES C/CO VS. THETA DATA FOR THE PLOTTER
C       USING DATA FROM THE LEAST SQUARES TAIL FIT PROGRAM AND
C       PARAMETERS FROM THE SIMPLEX MIXING MODEL FIT
C
C       NJOBS - NO. OF DIFFERENT RUNS TO BE DONE
C       NRUN - USERS EXPERIMENTAL RUN IDENTIFICATION
C       NPTSY1 - NO. OF POINTS FOR FIRST STEP SIZE
C       NPTSY2 - NO. OF POINTS FOR SECOND STEP SIZE
C       DELTY1 - FIRST STEP SIZE (MINUTES)
C       DELTY2 - SECOND STEP SIZE (MINUTES)
C       DTIME - DELAY TIME BETWEEN IMPULSE INPUT AND FIRST OUTPUT
C               RESPONSE
C       DW - DUMMY VARIABLE
C       Y1 - CONCENTRATION DATA FOR FIRST STEP SIZE
C       Y2 - CONCENTRATION DATA FOR SECOND STEP SIZE
C       NPROBS - NO. OF MIXING MODELS FITTED TO ONE SET OF DATA
C       NMODEL - MIXING MODEL CONFIGURATION NUMBER
C       NCODE - MIXING MODEL CODE NUMBER
C       N - NO. OF PARAMETERS IN MIXING MODEL
C       TAUTOTM - TOTAL RESIDENCE TIME OF MIXING MODEL
C       X - ARRAY CONTAINING PARAMETER VALUES
C       FMT - FORMAT ARRAY
C       TAUTOT - TOTAL RESIDENCE TIME OF DATA
C       TIME - ARRAY OF REAL TIME VALUES
C       P - PREDICTED VALUE OF E(T)
C       THETA - ARRAY OF REDUCED TIME VALUES
C       TAU - RESIDENCE TIME FOR DISPERSION MODEL
C       PE - D/UL FOR DISPERSION MODEL
C       R - ARRAY OF VALUES OF LAMDA USED FOR DISPERSION MODEL WITH
C           CLOSED BOUNDARY CONDITIONS
C       SC - VALUE OF C/CO FOR DISPERSION MODEL
C
      DIMENSION YO(500),YP(500),TIME(500),Y1(300),Y2(450),FMT(16),R(50),
     1THETA(500)
      COMMON T,P,X(5),NMODEL,NCODE,DTIME
      READ (5,100)  NJOBS
      DO 399  JOBS = 1, NJOBS
      READ(5,120) NRUN,NPTSY1,NPTSY2
      READ (5,101)  DELTY1, DELTY2
      READ (5,101)  DTIME
      READ(5,102) DW
      DELTY1=DELTY1/60.
      DELTY2=DELTY2/60.
      DTIME=DTIME/60.
      READ(5,101) (Y1(I),I=1,NPTSY1)
      IF (NPTSY2.EQ.0)  GO TO 440
      READ(5,101) (Y2(I),I=1,NPTSY2)
  440 CONTINUE
      READ(5,100) NPROBS
      DO 399 IJK=1,NPROBS
      READ(5,100) NMODEL,NCODE,N
      READ(5,140) TAUTOTM
      READ(5,110) (X(I),I=1,N)
      READ(5,130) (FMT(I),I=1,16)
      TAUTOTM=TAUTOTM+DTIME
```

```
C
C         CALCULATE FIRST MOMENT OF RESIDENCE TIME DISTRIBUTIONS
C
      NY1M1 = NPTSY1 - 1
      NY2M1 = NPTSY2 - 1
      ENY1M1 = FLOAT(NY1M1)
      ENY2M1 = FLOAT(NY2M1)
      NY1M2 = NPTSY1 - 2
      NY2M2 = NPTSY2 - 2
      TOPY1 = Y1(1) * DELTY1 / 2.0 * DELTY1 / 4.0 + Y1(NPTSY1) * DELTY1
     1 / 2.0 * (ENY1M1 * DELTY1 - DELTY1 / 4.0)
      BOTY1 = (Y1(1) + Y1(NPTSY1)) * DELTY1 / 2.0
      DO 306  I = 2, NY1M1
      FLTIM1 = I - 1
      TOPY1 = TOPY1 + Y1(I) * DELTY1 * FLTIM1 * DELTY1
  306 BOTY1 = BOTY1 + DELTY1 * Y1(I)
      IF (NPTSY2.EQ.0)  GO TO 476
      TIMEY1 = ENY1M1 * DELTY1
      TOPY2 = Y2(1) * DELTY2 / 2.0 * (TIMEY1 + DELTY2 / 4.0) +
     1Y2(NPTSY2) * DELTY2 / 2.0 * (TIMEY1 + ENY2M1 * DELTY2 -
     2DELTY2 / 4=0)
      BOTY2 = (Y2(1) + Y2(NPTSY2)) * DELTY2 / 2.0
      DO 312  I = 2, NY2M1
      FLTIM1 = I - 1
      TOPY2 = TOPY2 + Y2(I) * DELTY2 * (TIMEY1 + FLTIM1 * DELTY2)
  312 BOTY2 = BOTY2 + Y2(I) * DELTY2
      GO TO 477
  476 TOPY2 = 0.0
      BOTY2 = 0.0
  477 CONTINUE
      TOPY = TOPY1 + TOPY2
      BOTY = BOTY1 + BOTY2
      CENTY = TOPY / BOTY
      TAUTOT = CENTY + DTIME
C
C         CALCULATE C/CO VALUES FROM C VALUES
C
      DO 370  I = 1, NPTSY1
      Y1(I)=Y1(I)*TAUTOT/BOTY
  370 CONTINUE
      IF (NPTSY2.EQ.0)  GO TO 470
      DO 371  I = 1, NPTSY2
      Y2(I)=Y2(I)*TAUTOT/BOTY
  371 CONTINUE
  470 CONTINUE
C
C         BUILD TIME ARRAY OVER BOTH TIME INTERVALS
C
      TIME(1)=0.
      TIME(2)=DTIME
      NPT1=NPTSY1+1
      DO 600 I=3,NPT1
      TIME(I)=TIME(I-1)+DELTY1
  600 CONTINUE
      NPT2=NPTSY1+2
      NPT3=NPTSY2+NPTSY1
      DO 610 I=NPT2,NPT3
```

```
      TIME(I)=TIME(I-1)+DELTY2
  610 CONTINUE
C
C        BUILD OBSERVED C/CO ARRAY OVER BOTH INTERVALS
C
      YO(1)=0.
      YO(2)=0.
      DO 630 I=3,NPT1
      YO(I)=Y1(I-1)
  630 CONTINUE
      DO 640 I=NPT2,NPT3
      YO(I)=Y2(I-NPTSY1)
  640 CONTINUE
C
C        GO TO DISPERSION MODEL FOR 9990 OR 9999
C
      IF((NMODEL.EQ.9999).OR.(NMODEL.EQ.9990)) GO TO 800
C
C        CALCULATE PREDICTED C/CO FROM INVERSE LAPLACE TRANSFORM
C
      DO 650 I=1,NPT3
      T=TIME(I)*TAUTOTM/TAUTOT
      CALL TMODEL
      YP(I)=P*TAUTOTM
      THETA(I)=TIME(I)/TAUTOT
  650 CONTINUE
      GO TO 820
  800 CONTINUE
      TAU=X(1)
      PE=X(2)
C
C        CALCULATE R VALUES FOR DISPERSION MODEL
C
      CALL RCALC (PE,R)
      YP(1)=0.
      THETA(1)=0.
C
C        CALCULATE PREDICTED C/CO FROM DISPERSION MODEL
C
      DO 810 I=2,NPT3
      T=TIME(I)*TAU/TAUTOT
      CALL DISP (T,TAU,PE,R,SC)
      YP(I)=SC
      THETA(I)=TIME(I)/TAUTOT
  810 CONTINUE
  820 CONTINUE
      WRITE(6,200) NMODEL,NRUN
      WRITE(6,FMT) (X(I),I=1,N)
      WRITE(6,210)
      WRITE(6,220) (THETA(I),YO(I),YP(I),I=1,NPT3,5)
      DO 700 I=1,NPT3
      IF(THETA(I).GE.4.) GO TO 710
  700 CONTINUE
  710 CONTINUE
      NPT3=I
      WRITE(7,250) I
      WRITE(7,230) (THETA(I),YO(I),YP(I),I=1,NPT3)
```

```
  399 CONTINUE
      STOP
  100 FORMAT(16I5)
  101 FORMAT(8E10.3)
  102 FORMAT(16F5.0)
  110 FORMAT(4E20.13)
  120 FORMAT(A10,7I10)
  130 FORMAT(8A10)
  140 FORMAT(E20.13)
  200 FORMAT(1H1,13X,*TIME RESPONSE FOR MODEL *,I5,//23X,*ON *,A10//)
  220 FORMAT(10X,3E13.3)
  230 FORMAT(8E10.3)
  250 FORMAT(16I5)
  210 FORMAT(//16X,*THETA*,7X,*C/C0(0)*,6X,*C/C0(P)*//)
      END


      SUBROUTINE TMODEL
C
C         THIS SUBROUTINE CALCULATES VALUES OF E(T) FOR GIVEN T, MODEL,
C         DTIME AND PARAMETERS USING INVERSE LAPLACE TRANSFORMS
C
      COMMON T,P,X(5),NMODEL,NCODE,DTIME
C
C         DEFINE INVERSE LAPLACE TRANSFORMS
C
      F2730A(Q,T1,T,DT)=(1.-Q)/T1**2*(T-DT)*EXP((-T+DT)/T1)
      F2730B(Q,T1,T2,T,DT)=Q/T1*EXP((-T+T2+DT)/T1)
      F3240A(Q,T1,T,DT)=(1.-Q)/T1**3*(T-DT)**2/2.*EXP((-T+DT)/T1)
      F3240B(Q,T1,T2,T,DT)=Q/T1*EXP((-T+T2+DT)/T1)
      F3220A(Q,T3,T,DT)=(1.-Q)/T3**3*(T-DT)**2/2.*EXP((-T+DT)/T3)
      F3220B(Q,T1,T2,T,DT)=Q/T1*EXP((-T+T2+DT)/T1)
      F1460(T1,T2,Q,T,DT)=Q/T1/T2*((EXP((-T+DT)/T1)-EXP((-T+DT)/T2))/(1.
     1/T2-1./T1))+(1.-Q)/T1**3/T2*((1./(1./T2-1./T1)**3-(T-DT)/(1./T2-1.
     2/T1)**2+(T-DT)**2/2./(1./T2-1./T1))*EXP((-T+DT)/T1)-EXP((-T+DT)/T2
     3)/(1./T2-1./T1)**3)
      F1490(T1,T2,Q,T,DT)=Q/T1/T2*((EXP((-T+DT)/T2)-EXP((-T+DT)/T1))/(1.
     1/T1-1./T2))+(1.-Q)/T1**2/T2**2*((-2./(1./T2-1./T1)**3+(T-DT)/(1./T
     22-1./T1)**2)*EXP((-T+DT)/T1)+(2./(1./T2-1./T1)**3+(T-DT)/(1./T2-1.
     3/T1)**2)*EXP((-T+DT)/T2))
      F1190(T1,T2,Q,T,DT)=Q/T1/T2*((EXP((-T+DT)/T1)-EXP((-T+DT)/T2))/(1.
     1/T2-1./T1))+(1.-Q)/T1/T2**2*((EXP((-T+DT)/T1)-(1.+(1./T2-1./T1)*(T
     2-DT))*EXP((-T+DT)/T2))/(1./T2-1./T1)**2)
      F1150(T1,T2,Q,T,DT)=Q/T1/T2*((EXP((-T+DT)/T1)-EXP((-T+DT)/T2))/(1.
     1/T2-1./T1))+(1.-Q)/T2**3*(T-DT)**2/2.*EXP((-T+DT)/T2)
C
C         GO TO REQUIRED MODEL
C
      GO TO (2730,3240,3220,1460,1490,1190,1150) NCODE
 2730 CONTINUE
      IF(T.LE.DTIME) GO TO 300
      IF(T.LE.DTIME+X(2)) GO TO 310
      P=F2730A(X(3),X(1),T,DTIME)+F2730B(X(3),X(1),X(2),T,DTIME)
      RETURN
  300 CONTINUE
      P=0.
      RETURN
```

```
 310 CONTINUE
     P=F2730A(X(3),X(1),T,DTIME)
     RETURN
3240 CONTINUE
     IF(T.LE.DTIME) GO TO 320
     IF(T.LE.DTIME+X(2)) GO TO 330
     P=F3240A(X(3),X(1),T,DTIME)+F3240B(X(3),X(1),X(2),T,DTIME)
     RETURN
 320 CONTINUE
     P=0.
     RETURN
 330 CONTINUE
     P=F3240A(X(3),X(1),T,DTIME)
     RETURN
3220 CONTINUE
     IF(T.LE.DTIME) GO TO 340
     IF(T.LE.DTIME+X(2)) GO TO 350
     P=F3220A(X(4),X(3),T,DTIME)+F3220B(X(4),X(1),X(2),T,DTIME)
     RETURN
 340 CONTINUE
     P=0.
     RETURN
 350 CONTINUE
     P=F3220A(X(4),X(3),T,DTIME)
     RETURN
1460 CONTINUE
     IF(T.LE.DTIME) GO TO 360
     P=F1460(X(1),X(2),X(3),T,DTIME)
     RETURN
 360 CONTINUE
     P=0.
     RETURN
1490 CONTINUE
     IF(T.LE.DTIME) GO TO 370
     P=F1490(X(1),X(2),X(3),T,DTIME)
     RETURN
 370 CONTINUE
     P=0.
     RETURN
1190 CONTINUE
     IF(T.LE.DTIME) GO TO 380
     P=F1190(X(1),X(2),X(3),T,DTIME)
     RETURN
 380 CONTINUE
     P=0.
     RETURN
1150 CONTINUE
     IF(T.LE.DTIME) GO TO 390
     P=F1150(X(1),X(2),X(3),T,DTIME)
     RETURN
 390 CONTINUE
     P=0.
     RETURN
     END


     SUBROUTINE RCALC (P,R)
```

```
C
C         THIS SUBROUTINE SOLVES THE TRANSCENDENTAL EQUATION FOR POSITIVE
C         NON-TRIVIAL VALUES OF LAMDA
C
      DIMENSION R(50)
C
C         FIND 50 VALUES FOR R
C
      FR(R,P)=COS(R)/SIN(R)-R*P+.25/(R*P)
      R(1)=1.4
      DO 500 I=1,50
  510 CONTINUE
      R(I)=R(I)-1.E-3
      F=FR(R(I),P)
      IF(F.LE.0.) GO TO 510
  520 CONTINUE
      R(I)=R(I)+1.E-5
      F=FR(R(I),P)
      IF(F.GE.0.) GO TO 520
  530 CONTINUE
      R(I)=R(I)-1.E-7
      F=FR(R(I),P)
      IF(F.LT.0.) GO TO 530
      R(I+1)=R(I)+3.1417
  500 CONTINUE
      RETURN
      END


      SUBROUTINE DISP (T,TAU,P,R,SC)
C
C         THIS SUBROUTINE CALCULATES VALUES OF C/CO FOR THE DISPERSION
C         MODEL
C
      DIMENSION R(50),C(50)
C
C         CALCULATE PREDICTED ,C, VALUES FOR DISPERSION MODEL
C
      SC=0.
      U=.5/P
      DO 500 I=1,50
      C(I)=2.*R(I)*(U*SIN(R(I))+R(I)*COS(R(I)))*EXP(U-((U**2+R(I)**2)/(2
     1.*U))*T/TAU)/(U**2+2.*U+R(I)**2)
      SC=SC+C(I)
      IF(I.EQ.1) GO TO 500
      IF(ABS(C(I)+C(I-1)).LE.1.E-6) GO TO 510
  500 CONTINUE
  510 CONTINUE
      RETURN
      END


C         THIS PROGRAM PLOTS C/CO VS. THETA USING DATA FROM THE TIME
C         RESPONSE PROGRAM. THE RAW DATA IS PLOTTED AS POINTS AND THE
C         FITTED MODELS ARE PLOTTED AS LINES. THE FIRST MODEL IS A
C         SOLID LINE WHILE THE SUCCEEDING ONES ARE DASHED LINES
C
```

```
C          NJOBS - NO. OF PLOTS TO BE DONE
C          NFR - FREQUENCY OF DOTS IN DASHED LINES
C          Y - PARAMETER FOR 'Y' GRADUATIONS
C          X - PARAMETER FOR 'X' GRADUATIONS
C          NPROBS - NO. OF MODELS TO BE FITTED TO ONE SET OF POINTS
C          XX - PARAMETER FOR LEGEND
C          YY - PARAMETER FOR LEGEND
C          NMODEL - MODEL CONFIGURATION NUMBER
C          N - NUMBER OF DATA POINTS USED
C          T - VALUE OF THETA IN DATA UNITS
C          YO - VALUE OF OBSERVED C/CO IN DATA UNITS
C          YP - VALUE OF PREDICTED C/CO IN DATA UNITS
C          TP - VALUE OF THETA IN PLOTTER UNITS
C          YOP - VILUE OF OBSERVED C/CO IN PLOTTER UNITS
C          YPP - VALUE OF PREDICTED C/CO IN PLOTTER UNITS
C          YL - PARAMETER FOR LEGEND
C
      DIMENSION T(400),YO(400),YP(400),YOP(400),TP(400),YPP(400)
      DIMENSION B(400),S(400)
      DIMENSION XX(2),YY(2)
      REAL NMODEL
      READ(5,110) XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX
C
C          SET UP DATA SCALING
C
      CALL PLTIN(XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX)
      CALL DATE(THEDATE)
C
C          PLOT NAME AND DATE
C
      CALL LETTER(8,.24,90.,.5,1.,8HCRAWFORD)
      CALL LETTER(10,.24,90.,1.,1.,THEDATE)
      CALL PLOT(1.5,0.,-3)
      READ(5,120) NJOBS
      DO 399 IJK=1,NJOBS
      NFR=0
C
C          PLOT AXES
C
      CALL ARROW(2.,2.,12.5,2.,3)
      CALL ARROW(2.,7.5,2.,2.,2)
C
C          PLOT ,Y, GRADUATIONS
C
      Y=2.5
      DO 300 I=1,10
      CALL PLOT(1.9,Y,3)
      CALL PLOT(2.,Y,2)
      Y=Y+.5
  300 CONTINUE
C
C          PLOT ,X, GRADUATIONS
C
      X=4.5
      DO 310 I=1,4
      CALL PLOT(X,1.9,3)
      CALL PLOT(X,2.,2)
```

```
      X=X+2.5
  310 CONTINUE
C
C        NUMBER AXES
C
      CALL LETTER(3,.1,0.,1.5,1.68,3H  0)
      CALL LETTER(3,.1,0.,1.5,2.95,3H0.2)
      CALL LETTER(3,.1,0.,1.5,3.95,3H0.4)
      CALL LETTER(3,.1,0.,1.5,4.95,3H0.6)
      CALL LETTER(3,.1,0.,1.5,5.95,3H0.8)
      CALL LETTER(3,.1,0.,1.5,6.95,3H1.0)
      CALL LETTER(3,.1,0., 4.35,1.68,3H0.5)
      CALL LETTER(3,.1,0., 6.85,1.68,3H1.0)
      CALL LETTER(3,.1,0., 9.35,1.68,3H1.5)
      CALL LETTER(3,.1,0.,11.85,1.68,3H2.0)
C
C        LABEL AXES
C
      CALL LETTER(3,.15,0.,.38,4.7,3HC/C)
      CALL LETTER(1,.075,0.,.93,4.675,1HO)
      CALL GREEK(6.925,1.35,.15,0.,5HTHETA)
      READ(5,120) NPROBS
      XX(1)=7.
      XX(2)=8.
      YY(1)=YY(2)=6.5
      DO 398 K=1,NPROBS
      READ(5,130) NMODEL
      READ(5,120) N
      READ(5,100) (T(I),YO(I),YP(I),I=1,N)
      DO 900 I=1,N
      IF(T(I).GT.2.) GO TO 910
      JJ=I
  900 CONTINUE
  910 CONTINUE
      N=JJ
      IF(K.NE.1) GO TO 800
C
C        PLOT DATA POINTS
C
      DO 520 I=1,N
      CALL UNITTO(T(I),YO(I),TP(I),YOP(I))
      CALL GRAF(TP(I),YOP(I),.07,3HCIR)
  520 CONTINUE
C
C        PLOT DATA POINTS IN LEGEND
C
      DO 810 I=1,3
      CALL GRAF(XX(I),YY(I),.07,3HCIR)
      XX(I)=XX(I)+.5
  810 CONTINUE
      XX(1)=7.
      YY(1)=YY(2)=6.25
C
C        LABEL DATA POINTS IN LEGEND
C
      CALL LETTER(13,.15,0.,8.12,6.45,13H= DATA POINTS)
C
```

```
C          PLOT SOLID LINE
C
      CALL PLTMPL(T,YP,N)
C
C          PLOT SOLID LINE IN LEGEND
C
      CALL PLOT(XX(1),YY(1),3)
      CALL PLOT(XX(2),YY(2),2)
C
C          LABEL SOLID LINE IN LEGEND
C
      CALL LETTER(2,.15,0.,8.12,6.175,2H- )
      CALL LETTER(10,.15,0.,8.42,6.175,NMODEL)
      YY(1)=YY(2)=6.
      GO TO 398
  800 CONTINUE
C
C          CONVERT FROM DATA UNITS TO PLOTTER UNITS
C
      DO 720 I=1,N
      CALL UNITTO(T(I),YP(I),TP(I),YPP(I))
  720 CONTINUE
C
C          PLOT DASHED LINE
C
      CALL DDASHM(TP,YPP,N,.5,0.,NFR,.5,.1,IE,B,S)
C
C          PLOT DASHED LINE IN LEGEND
C
      CALL DDASHM(XX,YY,2,.5,0.,NFR,.5,.1,IE,B,S)
      YL=6.425-FLOAT(K)*.25
C
C          LABEL DASHED LINE IN LEGEND
C
      CALL LETTER(2,.15,0.,8.12,YL,2H- )
      CALL LETTER(10,.15,0.,8.42,YL,NMODEL)
      YY(1)=YY(2)=6.5-FLOAT(K+1)*.25
      NFR=K-1
  398 CONTINUE
      CALL PLOT(15.,0.,-3)
  399 CONTINUE
      CALL PLOT(15.,0.,999)
      STOP
  100 FORMAT(8E10.3)
  110 FORMAT(8F10.0)
  120 FORMAT(5I5)
  130 FORMAT(8A10)
      END
```

APPENDIX F

```
C       THIS PROGRAM FITS THE EDWARDS BATCH KINETIC MODEL TO BATCH
C       KINETIC DATA. IT WILL FIT THE MODEL WITH FROM 3 TO 6 PARAMETERS
C
C       NN - NO. OF INDEPENDENT AND DEPENDENT VARIABLES
C       NTHETA - NO. OF DATA POINTS
C       ARRAY - ARRAY CONTAINING OBSERVED INDEPENDENT AND DEPENDENT
C               VARIABLES
C       NPUNCH - PUNCH CONTROL - 1 - PUNCHES DATA
C                                0 - DOES NOT PUNCH
C       NJOBS - NO. OF DIFFERENT KINETIC MODELS TO BE USED
C       N - NUMBER OF PARAMETERS
C       NPROBS - NO. OF DIFFERENT INITIAL GUESSES OF PARAMETERS TO BE
C                USED
C       X - ARRAY CONTAINING VALUES OF PARAMETERS AT EACH VERTEX OF
C           SIMPLEX
C       CRDL - RESIDUAL CARBON CONCENTRATION
C       XX - ARRAY OF FINAL PARAMETER VALUES
C       RMS - RESIDUAL SUM OF SQUARES
C       CP - PREDICTED VALUES OF CARBON CONCENTRATION
C       T - VALUES OF TIME
C
      DIMENSION X(10,11),XX(10),T(55),CP(55)
      COMMON ARRAY(20,2),NTHETA,JJ,CRDL,N
      NN=2
      NTHETA=8
      READ(5,100) ((ARRAY(I,J),J=1,NN),I=1,NTHETA)
      READ(5,120) NPUNCH
      READ(5,120) NJOBS
      DO 399 IJKL=1,NJOBS
      READ(5,120) N
      N1=N+1
      READ(5,120) NPROBS
      DO 399 IJK=1,NPROBS
      DO 700 I=1,6
      X(I,1)=0.
  700 CONTINUE
      READ(5,100) ((X(J,I),J=1,N),I=1,N1)
      WRITE(6,230)
      CRDL=0.
      NTHETA=8
C
C        CALL SIMPLEX TO ESTIMATE PARAMETERS
C
      CALL SIMPLEX(N,X,1.E-06,1500,1.,.5,2.,1)
      NTHETA=20
      CRDL=18.
C
C        EXTEND RANGE OF TIME AND CARBON
C
      DO 901 I=9,NTHETA
      ARRAY(I,1)=ARRAY(I-1,1)+2.5/10
      ARRAY(I,2)=ARRAY(I-1,2)
  901 CONTINUE
C
C        ADD RESIDUAL TO RAW DATA
```

```
C
      DO 900 I=1,NTHETA
      ARRAY(I,2)=ARRAY(I,2)+CRDL
  900 CONTINUE
      DO 902 I=1,6
      XX(I)=X(I,1)
  902 CONTINUE
C
C         CALCULATE RESIDUAL SUM OF SQUARES
C
      CALL OBJECT (XX,RMS)
      WRITE(6,230)
      WRITE(6,200)
C
C         CALCULATE PREDICTED VALUES OF CARBON
C
      DO 530 I=1,NTHETA
      C=X(1,1)/(1.+EXP(X(2,1)+X(3,1)*ARRAY(I,1)+X(4,1)*ARRAY(I,1)**2+
     1X(5,1)*ARRAY(I,1)**3+X(6,1)*ARRAY(I,1)**4))+CRDL
      WRITE(6,210) ARRAY(I,1),ARRAY(I,2),C
  530 CONTINUE
      WRITE(6,220) (X(I,1),I=1,6)
      WRITE(6,240) RMS
      IF(NPUNCH.NE.1) GO TO 810
      WRITE(7,110) (X(I,1),I=1,N)
      WRITE(7,250) ((ARRAY(I,J),J=1,NN),I=1,11)
      T(1)=0.
C
C         CALCULATE PREDICTED VALUE OF CARBON FOR PLOTTER
C
      DO 800 I=1,51
      CP(I)=XX(1)/(1.+EXP(XX(2)+XX(3)*T(I)+XX(4)*T(I)**2+XX(5)*T(I)**3+X
     1X(6)*T(I)**4))+CRDL
      T(I+1)=T(I)+5./100.
  800 CONTINUE
      WRITE(7,250) (T(I),CP(I),I=1,51)
  810 CONTINUE
      DO 910 I=1,NTHETA
      ARRAY(I,2)=ARRAY(I,2)-CRDL
  910 CONTINUE
  399 CONTINUE
      STOP
  100 FORMAT(8F10.0)
  110 FORMAT(4E20.13)
  120 FORMAT(16I5)
  200 FORMAT(14X,* T *,8X,*TOC(O)*,7X,* TOC(C)*/)
  210 FORMAT(7X,3E13.3)
  220 FORMAT(//10X,*K = *,E10.3,5X,*A0 = *,E10.3,5X,*A1 = *,E10.3//9X,*A
     12 = *,E10.3,5X,*A3 = *,E10.3,5X,*A4 = *,E10.3)
  230 FORMAT(1H1)
  240 FORMAT(1H0,7X,*RMS = *,E10.3)
  250 FORMAT(8E10.3)
      END

      SUBROUTINE OBJECT(Y,RMS)
C
```

```
C         THIS SUBROUTINE CALCULATES THE RESIDUAL SUM OF SQUARES BETWEEN
C         THE OBSERVED AND PREDICTED VALUES OF CARBON CONCENTRATION
C
      COMMON ARRAY(20,2),NTHETA,JJ,CRDL,N
      DIMENSION Y(6)
      IF(N.EQ.6) GO TO 510
C
C         SET HIGHER ORDER PARAMETERS TO ZERO
C
      K=N+1
      DO 520 I=K,6
      Y(I)=0.
  520 CONTINUE
  510 CONTINUE
      RMS=0.
C
C         CALCULATE RESIDUAL SUM OF SQUARES
C
      DO 500 I=1,NTHETA
      C=Y(1)/(1.+EXP(Y(2)+Y(3)*ARRAY(I,1)+Y(4)*ARRAY(I,1)**2+Y(5)*ARRAY
     1(I,1)**3+Y(6)*ARRAY(I,1)**4))+CRDL
      RMS=RMS+(C-ARRAY(I,2))**2
  500 CONTINUE
      RETURN
      END


         FOR SIMPLEX SUBROUTINE SEE MIXING MODEL PARAMETER ESTIMATION
         PROGRAM


C         THIS PROGRAM PERFORMS A LINEAR LEAST SQUARES FIT ON CARBON
C         CONCENTRATION VS. TIME DATA
C
C         NJOBS - NO. OF SETS OF DATA TO BE FITTED
C         RUN - EXPERIMENTAL RUN NUMBER
C         DT - TIME STEP SIZE
C         NO - FIRST POINT NUMBER
C         NN - LAST POINT NUMBER
C         CO - OBSERVED CARBON CONCENTRATION VALUES
C         A - CARBON CONCENTRATION INTERCEPT VALUE
C         B - SLOPE OF FITTED LINE
C         CP - PREDICTED CARBON CONCENTRATION VALUES
C         RSS - RESIDUAL SUM OF SQUARES BETWEEN OBSERVED AND PREDICTED
C               VALUES OF CARBON CONCENTRATION
C         D - TIME INTERCEPT VALUE
C         TT - TIME VALUES FOR PLOTTER
C
      DIMENSION CO(10),CP(10),T(10),TT(3),C(3)
      REAL N
      READ(5,120) NJOBS
      DO 399 IJK=1,NJOBS
      READ(5,130) RUN,DT
C
C         BUILD TIME ARRAY
C
      T(1)=0.
```

```
      DO 500 I=2,8
      T(I)=T(I-1)+DT
  500 CONTINUE
      READ(5,110) NO,NN
C
C        READ OBSERVED CARBON VALUES
C
      READ(5,100) (CO(I),I=1,NN)
C
C        SUBTRACT RESIDUAL CARBON
C
      DO 530 I=1,NN
      CO(I)=CO(I)-18.
  530 CONTINUE
      N=FLOAT(NN-NO+1)
C
C        PERFORM LINEAR LEAST SQUARES FIT
C
      SX=0.
      SY=0.
      SXY=0.
      SX2=0.
      DO 510 I=NO,NN
      Y=CO(I)
      X=T(I)
      SX=SX+X
      SY=SY+Y
      SXY=SXY+X*Y
      SX2=SX2+X*X
  510 CONTINUE
      B=(N*SXY-SX*SY)/(N*SX2-SX*SX)
      A=SY/N-B*SX/N
      WRITE(6,230) RUN
      RSS=0.
C
C        CALCULATE PREDICTED CARBON AND RESIDUAL SUM OF SQUARES
C
      DO 520 I=NO,NN
      CP(I)=A+B*T(I)
      IF(I.GT.7) GO TO 700
      RSS=RSS+(CP(I)-CO(I))**2
  700 CONTINUE
      WRITE(6,200) T(I),CO(I),CP(I)
  520 CONTINUE
      D=-A/B
      WRITE(6,250) A,B,D
      WRITE(6,260) RSS
      WRITE(7,240) A,B,D
C
C        PUNCH DATA FOR PLOTTER
C
      C(1)=A+18.
      C(2)=C(3)=18.
      TT(1)=0.
      TT(2)=D
      TT(3)=2.5
      WRITE(7,220) (TT(I),C(I),I=1,3)
```

```
  399 CONTINUE
      STOP
  100 FORMAT(16F5.2)
  110 FORMAT(2I5)
  120 FORMAT(16I5)
  130 FORMAT(A10,F5.0)
  200 FORMAT(10X,3E13.3)
  210 FORMAT(5X,10E13.3)
  220 FORMAT(8E10.3)
  230 FORMAT(1H1,20X,*RUN NUMBER *,A10//19X,*LINEAR LEAST SQUARES FIT*//
     1/17X,*T*,11X,*C(OB)*,8X,*C(PR)*/)
  240 FORMAT(4E20.13)
  250 FORMAT(//10X,*A = *,E10.3,3X,*B = *,E10.3,3X,*D = *,E10.3)
  260 FORMAT(//10X,*RESIDUAL SUM OF SQUARES = *,E10.3)
      END


C         THIS PROGRAM PLOTS CARBON CONCENTRATION VS. TIME USING DATA
C         FROM THE LINEAR LEAST SQUARES FIT AND THE SIMPLEX EDWARDS MODEL
C         FIT. THE RAW DATA IS PLOTTED AS POINTS AND THE FITTED MODELS
C         ARE PLOTTED AS LINES. THE LINEAR MODEL IS A SOLID LINE WHILE
C         THE EDWARDS MODELS ARE DASHED LINES.
C
C         Y - PARAMETER FOR 'Y' GRADUATIONS
C         Z - PARAMETER FOR 'X' GRADUATIONS
C         XX - PARAMETER FOR LEGEND
C         YY - PARAMETER FOR LEGEND
C         X1 - TIME VALUES IN DATA UNITS
C         X2 - CARBON CONCENTRATION VALUES IN DATA UNITS
C         PX1 - TIME VALUES IN PLOTTED UNITS
C         PX2 - CARBON CONCENTRATION VALUES IN PLOTTER UNITS
C         NJOBS - NO. OF EDWARDS MODELS USED
C         NFR - FREQUENCY OF DOTS IN DASHED LINE
C
      DIMENSION X1(55),X2(55),PX1(55),PX2(55)
      DIMENSION XX(2),YY(2)
      DIMENSION B(52),S(52)
      CALL DATE(THEDATE)
C
C         PLOT NAME AND DATE
C
      CALL LETTER(8,.24,90.,.5,1.,8HCRAWFORD)
      CALL LETTER(10,.24,90.,1.,1.,THEDATE)
      CALL PLOT(1.5,0.,-3)
C
C         PLOT AXES
C
      CALL ARROW(2.,2.,8.5,2.,3)
      CALL ARROW(2.,2.,2.,10.5,3)
C
C         PLOT ,Y, GRADUATIONS
C
      Y=10.
      DO 600 I=1,8
      CALL PLOT(1.9,Y,3)
      CALL PLOT(2.,Y,2)
      Y=Y-1.
```

```
  600 CONTINUE
C
C         PLOT ,X, GRADUATIONS
C
      Z=3.
      DO 610 I=1,6
      CALL PLOT(Z,1.9,3)
      CALL PLOT(Z,2.,2)
      Z=Z+1.
  610 CONTINUE
C
C         NUMBER AXES
C
      CALL LETTER(3,.1,0.,1.5,9.95,3H160)
      CALL LETTER(3,.1,0.,1.5,8.95,3H140)
      CALL LETTER(3,.1,0.,1.5,7.95,3H120)
      CALL LETTER(3,.1,0.,1.5,6.95,3H100)
      CALL LETTER(3,.1,0.,1.5,5.95,3H 80)
      CALL LETTER(3,.1,0.,1.5,4.95,3H 60)
      CALL LETTER(3,.1,0.,1.5,3.95,3H 40)
      CALL LETTER(3,.1,0.,1.5,2.95,3H 20)
      CALL LETTER(3,.1,0.,1.5,1.68,3H  0)
      CALL LETTER(3,.1,0.,2.85,1.68,3H0.5)
      CALL LETTER(3,.1,0.,3.85,1.68,3H1.0)
      CALL LETTER(3,.1,0.,4.85,1.68,3H1.5)
      CALL LETTER(3,.1,0.,5.85,1.68,3H2.0)
      CALL LETTER(3,.1,0.,6.85,1.68,3H2.5)
      CALL LETTER(3,.1,0.,7.85,1.68,3H3.0)
C
C         LABEL AXES
C
      CALL LETTER(1,.15,0.,4.925,1.38,1HT)
      CALL LETTER(7,.15,0.,4.475,1.13,7H(HOURS))
      CALL LETTER(1,.15,0.,.7,5.925,1HC)
      CALL LETTER(6,.15,0.,.325,5.675,6H(MG/L))
      READ(5,110) XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX
C
C         SET UP DATA SCALING
C
      CALL PLTIN(XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX)
      READ(5,100) (X1(I),X2(I),I=1,11)
      XX(1)=5.
      XX(2)=6.
      YY(1)=YY(2)=9.
C
C         PLOT DATA POINTS
C
      DO 700 I=1,11
      CALL UNITTO(X1(I),X2(I),PX1(I),PX2(I))
      CALL GRAF(PX1(I),PX2(I),.07,3HC1R)
  700 CONTINUE
C
C         PLOT DATA POINTS IN LEGEND
C
      DO 810 I=1,3
      CALL GRAF(XX(I),YY(I),.07,3HC1R)
      XX(I)=XX(I)+.5
```

```
  810 CONTINUE
      XX(1)=5.
      YY(1)=YY(2)=8.75
C
C        LABEL DATA POINTS IN LEGEND
C
      CALL LETTER(13,.15,0.,6.12,8.925,13H- DATA POINTS)
      READ(5,100) (X1(I),X2(I),I=1,3)
C
C        PLOT SOLID LINE
C
      CALL PLTMPL(X1,X2,3)
C
C        PLOT SOLID LINE IN LEGEND
C
      CALL PLOT(XX(1),YY(1),3)
      CALL PLOT(XX(2),YY(2),2)
      YY(1)=YY(2)=8.5
C
C        LABEL SOLID LINE IN LEGEND
C
      CALL LETTER(14,.15,0.,6.12,8.675,14H- LINEAR MODEL)
      READ(5,120) NJOBS
      NFR=0
      DO 399 K=1,NJOBS
      READ(5,100) (X1(I),X2(I),I=1,51)
C
C        CONVERT FROM DATA UNITS TO PLOTTER UNITS
C
      DO 800 I=1,51
      CALL UNITTO(X1(I),X2(I),PX1(I),PX2(I))
  800 CONTINUE
C
C        PLOT DASHED LINE
C
      CALL DDASHM(PX1,PX2,51,.5,0.,NFR,.5,.1,IE,6,S)
C
C        PLOT DASHED LINE IN LEGEND
C
      CALL DDASHM(XX,YY,2,.5,0.,NFR,.5,.1,IE,6,S)
C
C        LABEL DASHED LINE IN LEGEND
C
      CALL LETTER(19,.15,0.,6.12,8.425,19H- 6 PARAMETER MODEL)
      NFR=K
  399 CONTINUE
      CALL PLOT(12.,0.,-3)
      CALL PLOT(15.,0.,999)
      STOP
  100 FORMAT(8E10.3)
  110 FORMAT(8F10.0)
  120 FORMAT(16I5)
      END
```

APPENDIX G

```
C       THIS PROGRAM CALCULATES CONVERSIONS USING THE BATCH KINETIC
C       MODELS AND MIXING MODELS FOR THE CASE OF COMPLETE SEGREGATION
C       (J=1)
C       NJOBS - NO. OF MIXING MODELS USED
C       RUN - EXPERIMENTAL MIXING RUN NUMBER
C       DT - REDUCED TIME (THETA) INTERVAL
C       DTM - NOMINAL DTIME VALUE
C       NTAU - NO. OF DIFFERENT TOTAL RESIDENCE TIMES TO BE USED
C       DTMULT - TOTAL RESIDENCE TIME MULTIPLIER INCREMENT
C       TMULT - TOTAL RESIDENCE TIME MULTIPLIER
C       NMODEL - MIXING MODEL CONFIGURATION NUMBER
C       NCODE - MIXING MODEL CODE
C       N - NO. OF PARAMETERS IN MIXING MODEL
C       TAUTOTM - TOTAL RESIDENCE TIME
C       XX - NOMINAL PARAMETERS
C       TM1 - TOTAL RESIDENCE TIME (LESS PFTR IN SERIES) FOR BEST MIXING
C             MODEL
C       TM2 - TOTAL RESIDENCE TIME (LESS PFTR IN SERIES) FOR MIXING
C             MODEL BEING CONSIDERED
C       NPROBS - NO. OF KINETIC MODELS TO BE USED
C       KNMODEL - KINRTIC MODEL NUMBER
C       NK - NO. OF KINETIC MODEL PARAMETERS
C       CAIN - INLET CONCENTRATION
C       AA - KINETIC MODEL PARAMETERS
C       A - KINETIC MODEL PARAMETERS
C       DTIME - RESIDENCE TIME OF PFTR IN SERIES
C       MULTIPLY NOMINAL TAU VALUES BY TMULT AND NORMALIZE TO BEST
C       MODEL
      DIMENSION T(3000),XX(5),AA(2,6),F(50,5,2),TMULT(50),TAUTOT(50),FT1
     1(16),FT2(16),NK(2)
      COMMON X(5),A(6),NMODEL,NCODE,DTIME,KNMODEL,P,C
      READ(5,120) (FT1(I),I=1,16),(FT2(I),I=1,16)
      READ(5,100) NJOBS
      DO 399 IJK=1,NJOBS
      READ(5,150) RUN
      READ(5,110) DT,DTM
      DTM=DTM/60.
      READ(5,100) NTAU
      READ(5,150) DTMULT
      TMULT(1)=DTMULT
      DO 700 I=2,NTAU
      TMULT(I)=TMULT(I-1)+DTMULT
  700 CONTINUE
      READ(5,100) NMODEL,NCODE,N
      READ(5,140) TAUTOTM
      READ(5,130) (XX(I),I=1,N)
      READ(5,130) TM1,TM2
      READ(5,100) NPROBS
      DO 399 IJ=1,NPROBS
      READ(5,100) KNMODEL,NK(IJ)
      READ(5,110) CAIN
      K=NK(IJ)
      READ(5,130) (AA(IJ,I),I=1,K)
      DO 730 I=1,K
      A(I)=AA(IJ,I)
```

```
  730 CONTINUE
      DO 399 NT=1,NTAU
      N1=N-1
      IF(N.EQ.1) N1=N
      DO 710 I=1,N1
      X(I)=XX(I)*TMULT(NT)*TM1/TM2
  710 CONTINUE
      IF(N.EQ.1) GO TO 720
      X(N)=XX(N)
  720 CONTINUE
C
C        MULTIPLY NOMINAL DTIME VALUE BY TMULT
C
      DTIME=DTM*TMULT(NT)
C
C        CALCULATE TAUTOT VALUE
C
      TAUTOT(NT)=TAUTOTM*TMULT(NT)*TM1/TM2+DTIME
C
C        GO TO PFTR SECTION FOR NMODEL = 100
C
      IF(NMODEL.EQ.100) GO TO 800
C
C        CALCULATE STEP SIZE FOR NUMERICAL INTEGRATION
C
      DTK=.5/60.
      DTT=DT*TAUTOT(NT)
      IF(DTT.GT.DTK) DTT=DTK
      DTA=DTT/TAUTOT(NT)
      TMAX=2.5/TAUTOT(NT)
      NPTS=TMAX/DTA
      IF(NPTS.GT.5000) NPTS=5000
      T(1)=0.
C
C        CALL SUBROUTINES TMODEL AND KMODEL FOR VALUES OF RESIDENCE TIME
C        DISTRIBUTION AND BATCH KINETIC CURVE
C
      CALL TMODEL(T(1))
      CALL KMODEL(T(1))
C
C        INTEGRATE BY TRAPEZOIDAL RULE
C
      F1=P*C/2.
      NPTS1=NPTS-1
      DO 620 I=2,NPTS1
      T(I)=T(I-1)+DTT
      J=I
      CALL TMODEL(T(I))
      CALL KMODEL(T(I))
      F1=F1+P*C
      IF(F1.EQ.0.) GO TO 620
      E=P*C/F1
      IF(E.LE..0001) GO TO 610
  620 CONTINUE
  610 CONTINUE
      NPTS=J+1
      NPTS1=J
```

```
      T(NPTS)=T(NPTS1)+DTI
      CALL TMODEL(T(NPTS))
      CALL KMODEL(T(NPTS))
      F1=F1+P*C/2.
      F1=F1*T(NPTS)/FLOAT(NPTS1)
C
C        CALCULATE CONVERSION
C
      F(NT,IJK,IJ)=1.-F1/CAIN
      GO TO 399
  800 CONTINUE
C
C        CALCULATE CONVERSION FOR PFTR
C
      CALL KMODEL(TAUTOT(NT))
      F(NT,IJK,IJ)=1.-C/CAIN
  399 CONTINUE
      K=NK(1)
      WRITE(6,220) RUN,NK(1),(AA(1,I),I=1,K)
      WRITE(6,FT1)
      WRITE(6,FT2) (TAUTOT(I),(F(I,J,1),J=1,NJOBS),I=1,NTAU,3)
      WRITE(6,200) RUN,AA(2,1),AA(2,2)
      WRITE(6,FT1)
      WRITE(6,FT2) (TAUTOT(I),(F(I,J,2),J=1,NJOBS),I=1,NTAU,3)
      WRITE(7,230) (TMULT (I),(F(I,J,1),J=1,NJOBS),I=1,NTAU)
      WRITE(7,230) (TMULT (I),(F(I,J,2),J=1,NJOBS),I=1,NTAU)
      STOP
  100 FORMAT(16I5)
  110 FORMAT(8E10.3)
  120 FORMAT(8A10)
  130 FORMAT(4E20.13)
  140 FORMAT(E20.13)
  150 FORMAT(16F5.0)
  200 FORMAT(1H1,40X,*CONVERSION PREDICTIONS FOR TIMPANY *,F4.1,//47X,*B
     1ATCH KINETIC MODEL - LINEAR*//45X,*B = *,E10.3,3X,*M = *,E10.3///)
  220 FORMAT(1H1,40X,*CONVERSION PREDICTIONS FOR TIMPANY *,F4.1,//40X,*B
     1ATCH KINETIC MODEL - EDWARDS *,I1,* PARAMETER*//9X,*K = *,E10.3,3X
     2,*A0 = *,E10.3,3X,*A1 = *,E10.3,3X,*A2 = *,E10.3,3X,*A3 = *,E10.3,
     33X,*A4 = *,E10.3//)
  230 FORMAT(8E10.3)
      END


      SUBROUTINE KMODEL(T)
      COMMON X(5),A(6),NMODEL,NCODE,DTIME,KNMODEL,P,C
      IF(KNMODEL.EQ.2) GO TO 600
C
C        BATCH KINETIC CURVE FOR EDWARDS 6 PARAMETER MODEL
C
      B=A(2)+A(3)*T+A(4)*T**2+A(5)*T**3+A(6)*T**4
      IF(B.GT.676.) B=676.
      C=A(1)/(1.+EXP(B))
      RETURN
  600 CONTINUE
C
C        BATCH KINETIC CURVE FOR PIECEWISE LINEAR MODEL
C
```

```
      IF(T.GE.A(3)) GO TO 610
      C=A(1)+A(2)*T
      RETURN
  610 CONTINUE
      C=0.
      RETURN
      END


      SUBROUTINE TMODEL(T)
      COMMON X(5),A(6),NMODEL,NCODE,DTIME,KNMODEL,P,C
C
C        DEFINE RESIDENCE TIME DISTRIBUTIONS FOR VARIOUS MODELS
C
      F2730A(Q,T1,T,DT)=(1.-Q)/T1**2*(T-DT)*EXP((-T+DT)/T1)
      F2730B(Q,T1,T2,T,DT)=Q/T1*EXP((-T+T2+DT)/T1)
      F3240A(Q,T1,T,DT)=(1.-Q)/T1**3*(T-DT)**2/2.*EXP((-T+DT)/T1)
      F3240B(Q,T1,T2,T,DT)=Q/T1*EXP((-T+T2+DT)/T1)
      F3220A(Q,T3,T,DT)=(1.-Q)/T3**3*(T-DT)**2/2.*EXP((-T+DT)/T3)
      F3220B(Q,T1,T2,T,DT)=Q/T1*EXP((-T+T2+DT)/T1)
      F1460(T1,T2,Q,T,DT)=Q/T1/T2*((EXP((-T+DT)/T1)-EXP((-T+DT)/T2))/(1.
     1/T2-1./T1))+(1.-Q)/T1**3/T2*((1./(1./T2-1./T1)**3-(T-DT)/(1./T2-1.
     2/T1)**2+(T-DT)**2/2./(1./T2-1./T1))*EXP((-T+DT)/T1)-EXP((-T+DT)/T2
     3)/(1./T2-1./T1)**3)
      F1490(T1,T2,Q,T,DT)=Q/T1/T2*((EXP((-T+DT)/T2)-EXP((-T+DT)/T1))/(1.
     1/T1-1./T2))+(1.-Q)/T1**2/T2**2*((-2./(1./T2-1./T1)**3+(T-DT)/(1./T
     22-1./T1)**2)*EXP((-T+DT)/T1)+(2./(1./T2-1./T1)**3+(T-DT)/(1./T2-1.
     3/T1)**2)*EXP((-T+DT)/T2))
      F1190(T1,T2,Q,T,DT)=Q/T1/T2*((EXP((-T+DT)/T1)-EXP((-T+DT)/T2))/(1.
     1/T2-1./T1))+(1.-Q)/T1/T2**2*((EXP((-T+DT)/T1)-(1.+(1./T2-1./T1)*(T
     2-DT))*EXP((-T+DT)/T2))/(1./T2-1./T1)**2)
      F1150(T1,T2,Q,T,DT)=Q/T1/T2*((EXP((-T+DT)/T1)-EXP((-T+DT)/T2))/(1.
     1/T2-1./T1))+(1.-Q)/T2**3*(T-DT)**2/2.*EXP((-T+DT)/T2)
      F101(T1,T)=1./T1*EXP(-T/T1)
C
C        GO TO REQUIRED MODEL
C
      GO TO (2730,3240,3220,1460,1490,1190,1150,101) NCODE
 2730 CONTINUE
      IF(T.LE.DTIME) GO TO 300
      IF(T.LE.DTIME+X(2)) GO TO 310
      P=F2730A(X(3),X(1),T,DTIME)+F2730B(X(3),X(1),X(2),T,DTIME)
      RETURN
  300 CONTINUE
      P=0.
      RETURN
  310 CONTINUE
      P=F2730A(X(3),X(1),T,DTIME)
      RETURN
 3240 CONTINUE
      IF(T.LE.DTIME) GO TO 320
      IF(T.LE.DTIME+X(2)) GO TO 330
      P=F3240A(X(3),X(1),T,DTIME)+F3240B(X(3),X(1),X(2),T,DTIME)
      RETURN
  320 CONTINUE
      P=0.
      RETURN
```

```
  330 CONTINUE
      P=F3240A(X(3),X(1),T,DTIME)
      RETURN
 3220 CONTINUE
      IF(T.LE.DTIME) GO TO 340
      IF(T.LE.DTIME+X(2)) GO TO 350
      P=F3220A(X(4),X(3),T,DTIME)+F3220B(X(4),X(1),X(2),T,DTIME)
      RETURN
  340 CONTINUE
      P=0.
      RETURN
  350 CONTINUE
      P=F3220A(X(4),X(3),T,DTIME)
      RETURN
 1460 CONTINUE
      IF(T.LE.DTIME) GO TO 360
      P=F1460(X(1),X(2),X(3),T,DTIME)
      RETURN
  360 CONTINUE
      P=0.
      RETURN
 1490 CONTINUE
      IF(T.LE.DTIME) GO TO 370
      P=F1490(X(1),X(2),X(3),T,DTIME)
      RETURN
  370 CONTINUE
      P=0.
      RETURN
 1190 CONTINUE
      IF(T.LE.DTIME) GO TO 380
      P=F1190(X(1),X(2),X(3),T,DTIME)
      RETURN
  380 CONTINUE
      P=0.
      RETURN
 1150 CONTINUE
      IF(T.LE.DTIME) GO TO 390
      P=F1150(X(1),X(2),X(3),T,DTIME)
      RETURN
  390 CONTINUE
      P=0.
      RETURN
  101 CONTINUE
      P=F101(X(1),T)
      RETURN
      END


C         THIS PROGRAM PLOTS CONVERSION VS. TAU/TAUN USING DATA FROM THE
C         CONVERSION PROGRAM
C
C         NMODEL - ARRAY CONTAINING MIXING MODEL CONFIGURATION NUMBERS
C         NJOBS - NO. OF PLOTS TO BE DONE
C         Y - PARAMETER FOR 'Y' GRADUATIONS
C         X - PARAMETER FOR 'X' GRADUATIONS
C         NPROBS - NO. OF DIFFERENT MIXING MODELS TO BE USED
C         RES - ARRAY CONTAINING VALUES OF TAU/TAUN AND CONVERSION
```

```
C         NFR - FREQUENCY OF DOTS IN DASHED LINES
C         XX - PARAMETER FOR LEGEND
C         YY - PARAMETER FOR LEGEND
C         RES1 - VALUES OF TAU/TAUN IN DATA UNITS
C         RES2 - VALUES OF CONVERSION IN DATA UNITS
C         XP - VALUES OF TAU/TAUN IN PLOTTER UNITS
C         YP - VALUES OF CONVERSION IN PLOTTER UNITS
C         YL - PARAMETER FOR LEGEND
C
      DIMENSION RES(51,7),RES1(51),RES2(51),XP(51),YP(51)
      DIMENSION B(52),S(52)
      DIMENSION XX(2),YY(2),NMODEL(4),RS(51,7)
      CALL DATE(THEDATE)
C
C         PLOT NAME AND DATE
C
      CALL LETTER(8,.24,90.,.5,1.,8HCRAWFORD)
      CALL LETTER(10,.24,90.,1.,1.,THEDATE)
      CALL PLOT(1.5,0.,-3)
      READ(5,110) XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX
C
C         SET UP DATA SCALING
C
      CALL PLTIN(XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX)
      READ(5,130) (NMODEL(I),I=1,4)
      READ(5,120) NJOBS
      DO 399 IJK=1,NJOBS
C
C         PLOT AXES
C
      CALL ARROW(2.,2.,10.5,2.,3)
      CALL ARROW(2.,2.,2.,10.5,3)
C
C         PLOT ,Y, GRADUATIONS
C
      Y=10.
      DO 600 I=1,10
      CALL PLOT(1.9,Y,3)
      CALL PLOT(2.,Y,2)
      Y=Y-.8
  600 CONTINUE
C
C         PLOT ,X, GRADUATIONS
C
      X=3.
      DO 610 I=1,8
      CALL PLOT(X,1.9,3)
      CALL PLOT(X,2.,2)
      X=X+1.
  610 CONTINUE
C
C         NUMBER AXES
C
      CALL LETTER(3,.1,0.,1.5,9.95,3H100)
      CALL LETTER(3,.1,0.,1.5,8.35,3H 80)
      CALL LETTER(3,.1,0.,1.5,6.75,3H 60)
      CALL LETTER(3,.1,0.,1.5,5.15,3H 40)
```

```
      CALL LETTER(3,.1,0.,1.5,3.55,3H 20)
      CALL LETTER(3,.1,0.,1.5,1.68,3H  0)
      CALL LETTER(3,.1,0.,3.85,1.68,3H0.5)
      CALL LETTER(3,.1,0.,5.85,1.68,3H1.0)
      CALL LETTER(3,.1,0.,7.85,1.68,3H1.5)
      CALL LETTER(3,.1,0.,9.85,1.68,3H2.0)
C
C        LABEL AXES
C
      CALL LETTER(10,.15,90.,1.25,5.25,10HCONVERSION)
      CALL GREEK(5.7,1.35,.15,0.,3HTAU)
      CALL LETTER(1,.15,0.,5.85,1.35,1H/)
      CALL GREEK(6.,1.35,.15,0.,3HTAU)
      CALL LETTER(1,.075,0.,6.15,1.324,1HN)
      DO 500 J=1,7
      RES(1,J)=0.
  500 CONTINUE
      READ(5,120) NPROBS
      NP=NPROBS+1
      READ(5,100) ((RES(I,J),J=1,NP),I=2,51)
      NFR=0
      DO 800 I=1,51
      RES1(I)=RES(I,1)
  800 CONTINUE
      XX(1)=6.
      XX(2)=7.
      YY(1)=YY(2)=7.
      DO 398 K=1,NPROBS
      L=K+1
C
C        CONVERT FROM DATA UNITS TO PLOTTER UNITS
C
      DO 810 I=1,51
      RES2(I)=RES(I,L)
      CALL UNITTO(RES1(I),RES2(I),XP(I),YP(I))
  810 CONTINUE
      IF(L.NE.2) GO TO 820
C
C        PLOT SOLID LINE
C
      CALL PLTMPL(RES1,RES2,51)
C
C        PLOT SOLID LINE IN LEGEND
C
      CALL PLOT(XX(1),YY(1),3)
      CALL PLOT(XX(2),YY(2),2)
C
C        LABEL SOLID LINE IN LEGEND
C
      CALL LETTER(2,.15,0.,7.12,6.925,2H- )
      CALL LETTER(10,.15,0.,7.42,6.925,NMODEL(K))
      YY(1)=YY(2)=6.75
      GO TO 398
  820 CONTINUE
C
C        PLOT DASHED LINE
C
```

```
      CALL DDASHM(XP,YP,51,.5,0.,NFR,.5,.1,IE,B,S)
C
C        PLOT DASHED LINE IN LEGEND
C
      CALL DDASHM(XX,YY,2,.5,0.,NFR,.5,.1,IE,B,S)
      YL=6.925-FLOAT(K-1)*.25
C
C        LABEL DASHED LINE IN LEGEND
C
      CALL LETTER(2,.15,0.,7.12,YL,2H- )
      CALL LETTER(10,.15,0.,7.42,YL,NMODEL(K))
      YY(1)=YY(2)=7.-FLOAT(K)*.25
      NFR=K-1
  398 CONTINUE
      CALL PLOT(15.,0.,-3)
  399 CONTINUE
      CALL PLOT(15.,0.,999)
      STOP
  100 FORMAT(8E10.3)
  110 FORMAT(8F10.0)
  120 FORMAT(16I5)
  130 FORMAT(8A10)
      END
```

```
C         THIS PROGRAM CALCULATES PERCENT DIFFERENCES FROM THE NOMINAL
C         FOR THE DIFFERENT CONVERSIONS USING DATA FROM THE CONVERSION
C         PROGRAM
C
C         NJOBS - NUMBER OF KINETIC MODELS USED
C         RUN - EXPERIMENTAL RUN NUMBER
C         RES - ARRAY CONTAINING TAU/TAUN AND CONVERSIONS
C         PC - NOMINAL CONVERSION
C         DPC - PERCENT DIFFERENCE CONVERSION
C
      DIMENSION RS(51,7),RES(51,7),PC(51),DPC(51,7)
      READ(5,110) NJOBS
      DO 399 IJK=1,NJOBS
      READ(5,120) RUN
      DO 500 I=1,7
      RES(1,I)=0.
      DPC(1,I)=0.
  500 CONTINUE
      READ(5,100) ((RES(I,J),J=1,5),I=2,51)
      PC(1)=0.
      DO 510 I=2,51
      IF(IJK.EQ.2) GO TO 520
C
C         SET ONE CONVERSION AS NOMINAL
C
      PC(I)=RES(I,4)
  520 CONTINUE
C
C         CALCULATE PCT DIFFERENCES
C
      DPC(I,1)=(RES(I,2)/PC(I)-1.)*100.
      DPC(I,2)=(RES(I,3)/PC(I)-1.)*100.
      DPC(I,3)=(RES(I,5)/PC(I)-1.)*100.
      IF(IJK.EQ.1) GO TO 510
      DPC(I,4)=(RES(I,4)/PC(I)-1.)*100.
  510 CONTINUE
      IF(IJK.EQ.1) GO TO 700
      WRITE(6,210) RUN
      WRITE(6,240) (RES(I,1),(DPC(I,J),J=1,4),I=1,51,2)
      WRITE(7,230) (RES(I,1),(DPC(I,J),J=1,4),I=1,51)
      GO TO 399
  700 CONTINUE
      WRITE(6,220) RUN
      WRITE(6,200) (RES(I,1),(DPC(I,J),J=1,3),I=1,51,2)
      WRITE(7,230) (RES(I,1),(DPC(I,J),J=1,3),I=1,51)
  399 CONTINUE
      STOP
  100 FORMAT(8E10.3)
  110 FORMAT(16I5)
  120 FORMAT(8F10.0)
  200 FORMAT(36X,4E13.3)
  210 FORMAT(1H1,40X,*PERCENT DIFFERENCE IN CONVERSION FOR LINEAR KINETI
     1CS*//55X,*ON TIMPANY *,F4.1///55X,*PCT. DIFF. IN CONV.*//,30X,* TA
     2U/TAUN*,6X,*PFTR*,9X,*CSTR*,9X,*3240*,9X,*1460*/)
  220 FORMAT(1H1,40X,*PERCENT DIFFERENCE IN CONVERSION FOR 6 PAR. KINETI
```

```
     1CS*//55X,*ON TIMPANY *,F4.1///55X,*PCT. DIFF. IN CONV.*//,40X,* TA
     2U/TAUN*,6X,*PFTR*,9X,*CSTR*,9X,*3240*//
  230 FORMAT(8E10.3)
  240 FORMAT(26X,5E13.3)
      END


C          THIS PROGRAM PLOTS PERCENT DIFFERENCE IN CONVERSION VS.
C          TAU/TAUN USING DATA SUPPLIED BY THE PERCENT DIFFERENCE IN
C          CONVERSION PROGRAM
C
C          NJOBS - NO. OF PLOTS TO BE DONE
C          NMODEL - ARRAY CONTAINING MIXING MODEL CONFIGURATION NUMBERS
C          Z - PARAMETER FOR 'X' GRADUATIONS
C          Y - PARAMETER FOR 'Y' GRADUATIONS
C          NFR - FREQUENCY OF DOTS IN DASHED LINES
C          XL - PARAMETER FOR LEGEND
C          YL - PARAMETER FOR LEGEND
C          NPROBS - NO. OF DIFFERENT MIXING MODELS TO BE USED
C          X - ARRAY CONTAINING TAU/TAUN AND PCT. DIFF. IN CONVERSION
C          XX - VALUES OF TAU/TAUN IN DATA UNITS
C          YY - VALUES OF PCT. DIFF. IN DATA UNITS
C          XXP - VALUES OF TAU/TAUN IN PLOTTER UNITS
C          YYP - VALUES OF PCT. DIFF. IN PLOTTER UNITS
C          YYL - PARAMETER FOR LEGEND
C
      DIMENSION X(51,7),XX(51),YY(51),XXP(51),YYP(51),B(52),S(52)
      DIMENSION XL(2),YL(2)
      REAL NMODEL(6)
      READ(5,110) XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX
C
C          SET UP DATA SCALING
C
      CALL PLTIN(XSCALE,YSCALE,V,D,XMIN,XMAX,YMIN,YMAX)
      CALL DATE(THEDATE)
C
C          PLOT NAME AND DATE
C
      CALL LETTER(8,.24,90.,.5,1.,8HCRAWFORD)
      CALL LETTER(10,.24,90.,1.,1.,THEDATE)
      CALL PLOT(1.5,0.,-3)
      READ(5,120) NJOBS
      READ(5,130) (NMODEL(I),I=1,6)
      DO 399 IJK=1,NJOBS
C
C          PLOT AXES
C
      CALL ARROW(1.9,4.,10.5,4.,3)
      CALL ARROW(2.,9.5,2.,1.5,2)
C
C          PLOT ,X, GRADUATIONS
C
      Z=3.
      DO 600 I=1,8
      CALL PLOT(Z,4.05,3)
      CALL PLOT(Z,3.95,2)
      Z=Z+1.
```

```
  600 CONTINUE
C
C         PLOT ,Y, GRADUATIONS
C
      Y=9.
      DO 610 I=1,5
      CALL PLOT(1.9,Y,3)
      CALL PLOT(2.,Y,2)
      Y=Y-1.
  610 CONTINUE
      Y=3.
      DO 620 I=1,2
      CALL PLOT(1.9,Y,3)
      CALL PLOT(2.,Y,2)
      Y=Y-1.
  620 CONTINUE
C
C         NUMBER AXES
C
      CALL LETTER(3,.1,0.,1.55,1.95,3H-20)
      CALL LETTER(3,.1,0.,3.55,3.95,3H  0)
      CALL LETTER(3,.1,0.,5.55,5.95,3H 20)
      CALL LETTER(3,.1,0.,7.55,7.95,3H 40)
      CALL LETTER(3,.1,0.,3.85,4.15,3H0.5)
      CALL LETTER(3,.1,0.,5.85,4.15,3H1.0)
      CALL LETTER(3,.1,0.,7.85,4.15,3H1.5)
      CALL LETTER(3,.1,0.,9.85,4.15,3H2.0)
C
C         LABEL AXES
C
      CALL LETTER(18,.15,90.,1.,4.15,18HPERCENT DIFFERENCE)
      CALL LETTER(13,.15,90.,1.25,4.6,13HIN CONVERSION)
      CALL GREEK(10.75,3.925,.15,0.,3HTAU)
      CALL LETTER(1,.15,0.,10.9,3.925,1H/)
      CALL GREEK(11.05,3.925,.15,0.,3HTAU)
      CALL LETTER(1,.075,0.,11.2,3.9,1HN)
      NPR=0
      XL(1)=7.
      XL(2)=8.
      YL(1)=YL(2)=9.
      READ(5,120) NPROBS
      NPR=NPROBS+1
      READ(5,100) ((X(I,J),J=1,NPR),I=1,51)
      DO 398 K=1,NPROBS
      J=K+1
      IF(K.NE.1) GO TO 510
      DO 500 I=1,51
      XX(I)=X(I,1)
  500 CONTINUE
  510 CONTINUE
C
C         CONVERT FROM DATA UNITS TO PLOTTER UNITS
C
      DO 520 I=1,51
      YY(I)=X(I,J)
      CALL UNITTO(XX(I),YY(I),XXP(I),YYP(I))
  520 CONTINUE
```

```
      IF(K.NE.1) GO TO 530
C
C        PLOT SOLID LINE
C
      CALL PLTMPL(XX,YY,51)
C
C        PLOT SOLID LINE IN LEGEND
C
      CALL PLOT(XL(1),YL(1),3)
      CALL PLOT(XL(2),YL(2),2)
C
C        LABEL SOLID LINE IN LEGEND
C
      CALL LETTER(2,.15,0.,8.12,8.925,2H- )
      CALL LETTER(10,.15,0.,8.42,8.925,NMODEL(K))
      YL(1)=YL(2)=8.75
      GO TO 398
  530 CONTINUE
C
C        PLOT DASHED LINE
C
      CALL DDASHM(XXP,YYP,51,.5,0.,NFR,.5,.1,IE,B,S)
C
C        PLOT DASHED LINE IN LEGEND
C
      CALL DDASHM(XL,YL,2,.5,0.,NFR,.5,.1,IE,B,S)
      YYL=8.925-FLOAT(K-1)*.25
C
C        LABEL DASHED LINE IN LEGEND
C
      CALL LETTER(2,.15,0.,8.12,YYL,2H- )
      CALL LETTER(10,.15,0.,8.42,YYL,NMODEL(K))
      YL(1)=YL(2)=9.-FLOAT(K)*.25
      NFR=K-1
  398 CONTINUE
      CALL PLOT(12.,0.,-3)
  399 CONTINUE
      CALL PLOT(12.,0.,999)
      STOP
  100 FORMAT(8E10.3)
  110 FORMAT(8F10.0)
  120 FORMAT(16I5)
  130 FORMAT(8A10)
      END
```

APPENDIX I

RAW DATA

FOR

MIXING AND KINETIC MODELS

RAW DATA FOR EXPERIMENTAL RESPONSE

TIMPANY 61 30 58 204 2

5.

2. 3. 10.5 17. 24. 38. 43. 53.5 55. 58. 64. 67. 69. 71. 72. 73.5
74. 75. 75.5 75.7576. 75.7575.5 75.5 75. 75. 75. 74.5 74.5
74. 73. 71.5 69. 67.5 66. 65. 63.5 62.5 61. 60. 58.5 57. 56. 54.5 53.
52. 50. 48. 47.3 46.5 45.2 44. 42.8 41.5 40.8 40. 38.8 37.5 37. 36.5 36.
35.5 34.7 34. 33.2 32.5 31.8 31. 30.3 29.5 29. 28.5 28. 27.5 26.9 26.3 25.8
25. 24.6 24.2 23.9 23.5 22.8 22. 21.3 20.5 20.1 19.7 19.4 19. 18.8 18.5 18.3
18. 17.8 17.5 17.3 17. 16.6 16.2 15.9 15.5 15.3 15. 14.8 14.5 14.1 13.7 13.4
13. 12.8 12.5 12.3 12. 11.8 11.5 11.3 11. 10.8 10.6 10.4 10.2 10.1 9.9 9.7
9.5 9.3 9.1 8.9 8.7 8.6 8.4 8.2 8. 7.8 7.6 7.4 7.2 7.1 6.9 6.7
6.5 6.4 6.3 6.2 6. 5.9 5.8 5.7 5.5 5.3 5.1 4.9 4.7 4.6 4.4 4.2
4.0 3.9 3.8 3.8 3.7 3.6 3.5 3.4 3.4 3.3 3.2 3.1 3. 2.9 2.8 2.8
2.7 2.6 2.5 2.4 2.4 2.3 2.2 2.1 2. 1.9 1.9 1.8 1.7 1.7 1.6 1.6
1.5 1.5 1.4 1.4 1.3 1.3 1.2 1.2 1.1 1.1 1. 1. .9 .9 .8 .8
.8 .7 .7 .7 .6 .6 .6 .5 .5 .5 .4 .4 .4 .3 .3 .3
.2 .2 .2 .2 .2 .1 .1 .1 .1 .1 .1

TIMPANY 16 22 61 179 1

5.

6. 47. 63. 70. 77. 81.5 82.5 83. 83.5 83.8 83.5 83.5 83.3 83. 82.5 82.3
82. 81.8 81. 80.8 80.5
79. 77. 75.3 73.5 72. 70.3 68.8 67. 66. 64.8 63.3 61.5 60.5 59. 57.5 56.8
55.5 54. 53. 51.5 50.5 49.5 48.3 47. 46. 45. 44. 43. 42. 41.5 40.5 39.5
38.5 38. 37. 36. 35. 34.5 33.5 33. 32.3 31.5 31. 30. 29.5 29. 28. 27.5
27. 26.5 26. 25.5 24.5 24. 23.5 23. 22.5 22. 21.5 21. 20.5 20. 19.5 19.
18.5 18. 17.5 17.3 17. 16.5 16. 15.5 15.3 15. 14.5 14. 13.5 13.3 13. 12.8
12.3 12. 11.8 11.5 11.3 11. 10.5 10.3 10. 10. 9.5 9.5 9. 9. 8.8 8.5
8.5 8.3 8. 8. 7.5 7.5 7.3 7. 7. 6.8 6.5 6.3 6. 6. 6. 6.
5.5 5.5 5.3 5. 5. 5. 4.8 4.8 4.8 4.5 4.5 4.3 4. 4. 4. 4.
4. 3.8 3.8 3.5 3.5 3.5 3.5 3.5 3.3 3.3 3.3 3. 3. 2.8 2.8 2.5
2.5 2.5 2.5 2.3 2.3 2.3 2.3 2. 2. 2. 2. 2. 2. 2. 2. 1.8
1.8 1.8 1.5 1.5 1.5 1.5 1.5 1.5 1.3 1.3 1.3 1.3 1. 1. 1. 1.
1. 1.

RAW DATA FOR EXPERIMENTAL RESPONSE (CONT'D)

TIMPANY 18 20 66 121 1
10.
19. 40. 63. 70. 77. 81. 83. 83.5 84. 84.5 84.5 84.8 84.5 84.3 84.3 84.
84. 84. 83.8
82.8 81.3 79.5 78. 76.3 74.8 73.3 71.5 69.8 68.3 66.5 65.3 64. 62.5 61.3 59.8
58.5 57.5 56.5 55. 54. 53. 51.5 50.5 49.3 48.5 47. 46.3 45.3 44.5 43.3 42.5
41.5 40.5 39.8 39. 38. 37. 36.5 35.5 35. 34.2 33.3 32.5 31.8 31.3 30.5 30.
29.5 28.5 28. 27. 26.8 26.3 25.8 25.3 24.8 24.3 23.8 23.3 22.8 22. 21.5 21.3
20.5 20. 19.5 19. 18.5 18. 17.5 17.3 17. 16.5 16. 15.8 15.3 15. 14.5 14.3
14. 13.5 13.3 13. 12.8 12.5 12.3 12. 11.8 11.3 11. 10.5 10.5 10.3 10. 9.8
9.5 9.3 9. 8.5 8.5 8.3 8. 8. 7.8 7.5 7.5 7.3 7. 7. 6.8 6.5
6.3 6. 6. 5.8 5.5 5.5 5.3 5.

TIMPANY 19 23 50 161 1
1.25
4.5 17. 31.5 43. 55. 62. 69. 73.5 76. 79. 80.5 82. 82.5 83. 83. 82.8
82.5 82. 81. 80.5 80. 79.
77.5 75.5 73.8 72. 70. 68.5 66.5 64.5 63. 61.3 59.8 58.5 57. 55.5 54. 52.5
51. 49.5 48. 47. 46. 44.5 43.3 41.5 40.3 39.5 38.3 37.5 36.8 35.8 34.8 33.8
32.5 31.8 31. 30. 29.3 28.5 27.5 27. 26. 25.5 24.5 24. 23.5 22.5 22. 21.5
21. 20. 19.5 19. 18.8 18. 17.5 17. 16.3 15.8 15. 15. 14.5 14. 13.5 13.
12.8 12.5 12. 11.8 11.5 11. 10.8 10.5 10. 9.8 9.5 9. 8.8 8.5 8.3 8.
7.8 7.5 7.3 7. 7. 6.8 6.8 6.5 6. 6. 5.5 5.3 5.3 5. 5. 4.8
4.8 4.5 4.5 4.5 4.5 4.5 4.3 4.3 4. 4. 3.5 3.3 3. 3. 3. 2.8
2.8 2.5 2.5 2.5 2.5 2.5 2.3 2.3 2.3 2.3 2.3 2. 2. 2. 2. 2.
1.8 1.8 1.8 1.8 1.5 1.5 1.5 1.3 1.3 1.3 1. 1. 1. 1. 1. 1.
1. 1. 1. .8 .8 .8 .8 .8 .8 .8 .5 .5 .5 .5 .5 .5

RAW DATA FOR EXPERIMENTAL RESPONSE (CONT'D)

TIMPANY 20 24 58 203 1

.5

4. 16. 27. 39.5 47.5 57. 64. 69. 73. 76. 78.5 80.5 81.5 82. 82.5 83.
83.3 83.3 83. 83. 82.8 82.8 82.5
80.5 78.5 76.5 75.5 73.5 71.5 69.5 68.5 66.5 65. 63.5 62.5 60.5 59.5 58. 57.
55.5 54. 53. 52. 50.5 49.5 48. 47. 46. 45. 43.5 42.5 41. 40. 39.3 38.2
37.2 36.5 35.8 35. 34. 33. 32.5 31.8 31. 30.2 29.5 29. 28.2 27.2 27. 26.
25.5 25. 24.3 23.8 23. 22.8 22.2 21.5 21. 20.5 20. 19.5 19. 18.2 18. 17.5
17. 16.8 16.2 16. 15.5 15. 14.5 14.5 14. 13.5 13.2 13. 12.8 12.5 12. 11.5
11. 10.5 10.5 10.5 10.2 9.8 9.8 9.2 8.5 8.2 8. 8. 7.8 7.8 7.2 7.1
7.1 7. 7. 6.8 6.5 6.2 5.8 5.5 5.3 5.3 5.3 5.2 5.1 5. 5. 4.7
4.7 4.5 4.2 4. 4. 4. 3.8 3.8 3.7 3.5 3.2 3.1 3.1 3. 3. 3.
2.9 2.8 2.8 2.6 2.5 2.2 2.2 2.2 2.2 2.1 2.1 2. 2. 1.9 1.8 1.8
1.5 1.5 1.5 1.5 1.2 1.2 1.2 1.1 1.1 1.1 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. .8 .8 .8 .8 .8 .8 .8 .6 .6 .5
.5 .5 .5 .4 .3 .3 .3 .2 .2 .2 .2 .2 .2 .2 .2 .2
.2 .2 .2 .2 .2 .2 .2 .1 .1 .1

RAW DATA FOR EDWARDS BATCH KINETIC MODEL PROGRAM

```
0.      138.    .25     117.    .5      85.5    .75     66.
1.      44.5    1.25    23.5    1.5     7.      1.75    0.
```

RAW DATA FOR LINEAR LEAST SQUARES KINETIC MODEL PROGRAM

```
POLLOCK 1 .25
    1    8
156. 135. 103.584.  62.5 41.5 25.  18.
```