THE UNIVERSITY OF CALGARY

EVALUATION OF IMMOBILIZED CELL CONTINUOUS PENICILLIN FERMENTATIONS

by

SEAN PATRICK FORESTELL

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

DEPARTMENT OF CHEMICAL AND PETROLEUM ENGINEERING

CALGARY, ALBERTA, CANADA

JUNE, 1988

ISBN 0-315-46595-6

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled,

"EVALUATION OF IMMOBILIZED CELL CONTINUOUS PENICILLIN FERMENTATIONS"

submitted by

Sean Patrick Forestell in partial fulfillment of the requirements for the degree of Master of Science.

Dr. N.E. Kalogerakis, Committee Chairman
Department of Chemical & Petroleum Engineering

Dr. F. Berruti
Department of Chemical & Petroleum Engineering

Dr. L.A. Behie
Department of Chemical & Petroleum Engineering

Dr. G.M. Gaucher
Department of Biological Sciences

June 22, 1988

Date

ABSTRACT

Modern production of the antibiotic penicillin is still performed in the fed-batch mode despite recent advances in fermentation technology. An alternative to this mode of production is to constrain the *Penicillium* onto an immobilization matrix, and run the fermentation in a continuous fashion. This would alleviate many of the problems associated with industrial penicillin fermentations such as high broth viscosity, poor oxygen transfer and cell washout.

The present study is an evaluation of immobilized cell continuous penicillin fermentations and considers use of a control system on a 19 *L* bioreactor. This evaluation was initially performed using two separately derived kinetic models coupled to the control algorithm and simulated on a computer. The proposed control algorithm was successful in controlling key process variables such as growth rate, penicillin and precursor concentrations. Experimentation to determine kinetic parameters indicated that several kinetic parameters for this system are significantly different from those reported in the literature for submerged culture. An interesting observation was that immobilization resulted in a 50% reduction in maintenance requirements for the cells. Furthermore, penicillin productivity was found to be 85% lower than expected for the strain used, and is 50% lower in immobilized systems than that found in free cell cultures. A properly tuned controller was able to maintain a desired growth rate and limit the amount of precursor present in the fermentation broth.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to the following people who were all an important part of this project. My supervisor Dr. N. Kalogerakis who initially got me interested in this project, and provided his guidance throughout. I am also grateful to both Dr. L.A. Behie and Dr. G.M. Gaucher who were closely tied to this project and provided their encouragement and support.

I wish to thank Dr. John Wong for his invaluable guidance and help in performing all of the experimental work. Also involved in the experimental work were Ms. Cornelia Feiss and Ms. Pamela Tetarenko who each took their turn assisting in various stages of the experiments. The patience of the other graduate students sharing Dr. Gaucher's lab is also gratefully acknowledged. I would also like to thank Mr. T. Linardos whose previous experience with this project and advice helped in several difficult moments.

Finally, I would like to thank my family and friends who have always supported me in all I do, and to whom I turn to when help is most needed. These are the people who are always there, never complain about all that they do for you, and get the least amount of recognition though they do more than they ever could know.

Dedicated to my parents and family

God bless them all

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A0	structured cell type, growing hyphae tip, g
A1	structured cell type, penicillin producing cells, g
A2	structured cell type, degenerated non-producing cells, g
a	parameter in equation (11) relating q_p to cell age
a0	A0-type cell concentration, g/L
a1	A1-type cell concentration, g/L
a2	a2-type cell concentration, g/L
b	stoichiometric constant (PoAA to penicillin)
c	constant in equation (72)
c^*	integration constant, equation (58)
c_1	ratio of a_1 to a_0
D	dilution rate, h^{-1}
K_1, K_2	Kalman filter gains
K_{cs}	feedback gain of growth rate controller
K_{cz}	feedback gain in precursor concentration controller
K_{cp}	feedback gain in penicillin concentration controller
k_d	first order death rate of A_1 cells, h^{-1}
k_h	first order rate of hydrolysis. h^{-1}
$K_L a$	overall mass transfer coefficient, h^{-1}
k_{pen}	second order rate of penicillin production in structured model, $L\ g^{-1}h^{-1}$.
k_{pi}	first order rate of formation of postulated intermediate, structured model, h^{-1}.
k_s	Monod equation saturation constant, g/L
k_T	first order rate of change from A_0 cells to A_1 cells, h^{-1}.

$P_{11,12}$	covariance matrix fators
m	maintenance requirement, h^{-1}
$\hat{m}$	maintenance requirement, structured model, h^{-1}
PoAA	phenoxyacetic acid
p	penicillin concentration, *g/L*
p_i	postulated intermediate, structured model, *g/L*
Q_s	substrate consumption, structured model, $g\ L^{-1}h^{-1}$
QSS	quasi-steady-state
$\hat{q}_c$	corrected specific production rate, h^{-1}
q_p	specific penicillin production rate, h^{-1}
$\hat{q}_p$	estimated specific production rate, h^{-1}
q_z	postulated first order dissappearance of precursor, h^{-1}
R	Denominator in Kalman filter
s	glucose concentration, *g/L*
t	time, *h*
V	working volume, *L*
X	total biomass, *g*
x	biomass concentration, *g/L*
Y_G	cell growth yield, *g biomass per g glucose*
$\hat{Y}_G$	cell growth yield, structured model, *g* A_0 *cells/g glucose*
Y_P	penicillin to glucose yield
$\hat{Y}_P$	penicillin to glucose yield, structured model, *g penicillin per g glucose*
y	natural logarithm of the measured biomass
$\hat{y}$	estimated value of y
z	precursor (PoAA) concentration, *g/L*

GREEK LETTERS

λ	average cell age, h
λ_c	forgetting factor, Kalman filter
μ	specific growth rate, h^{-1}
$\hat{\mu}$	first order growth of A_0-type cells, h^{-1}
$\bar{\mu}$	$\hat{\mu}$ achieved at the QSS, structured model, h^{-1}
σ	specific uptake rate of glucose, h^{-1}
$\hat{\sigma}$	estimated σ at QSS, h^{-1}

SUBSCRIPTS

des	desired value of the variable
exp	refers to experimental value of the variable
f	value of the variable in the feed stream
i	value of the variable at time interval i
max	maximum value of the variable
min	minimum value of the variable
sp	refers to the set-point value of the variable
o	initial value of the variable

CHAPTER I

INTRODUCTION

Antibiotics are products of secondary metabolism which inhibit the growth processes of other organisms. This inhibition was first observed by Sir Alexander Fleming in 1929 when *Staphylococcal* growth was inhibited by a contaminating *Penicillium notatum* culture. Since this initial discovery, penicillin has become a leading modern chemotherapeutic agent with a 1983 world production exceeding 20,000 metric tonnes (Pirt, 1985).

Given the importance of penicillin in the present antibiotic market, much effort has been expended in understanding pencillin fermentations from both a physiological and technical point of view. The role of the microbial physiologist is to study the effects of the environment on the organism and the maximization of its genetic potential. It has been estimated (Pirt, 1987) that only about 40% of the maximum penicillin productivity is being achieved by modern producers. This would indicate a deficiency in the understanding of the influence of environmental factors on the organism's synthetic capability and a need for more basic research in this area.

The improvement of antibiotic production also depends on the contributions of biochemical engineers. Fleming's original strain of *Penicillium notatum* produced only a few milligrams of penicillin per liter, but today's production strains can achieve potencies of

penicillin-G close to 30 *g/L* (Swartz, 1985). This improvement, according to Demain (1973), was initially due to advances in the art of submerged fermentation, and later to genetic advances in *Penicillium* strains. Between 1950 and 1980 the value of the specific production rate of penicillin, q_p, increased 10 fold due to genetic improvement of strains of *P.chrysogenum*. The output rate, however, increased 40-fold with improvements in fermentation techniques and a better understanding of the physiology of the producing organism (Pirt, 1985). Since 52% of total production costs arise from the penicillin fermentation (Lorenz et al, 1987), further improvements in reactor design and process control could reduce the two major expenses of energy and substrate supply.

Continuous operation of bioreactors allows for easier control of environmental and physiological factors in antibiotic fermentations. Another advantage of this mode of operation is minimized down-time for cleaning and maintenance. This can result in significant savings in operating costs and improved environmental conditions for the organism. Industrial processes presently using continuous fermentation include waste-water treatment, the production of beer, glucose isomerase, and ethanol. Disadvantages of continuous operation of antibiotic fermentations such as reverse mutagenesis in high yielding strains, inability to maintain sterile conditions over extended periods of time, and variable substrate composition have prevented the use of continuous bioreactors in industrial penicillin fermentations.

The most widely used configuration of a continuous bioreactor is a chemostat. The disadvantage of using a chemostat with mycelial fermentation broths is that output lines and filters can become plugged, and at high dilution rates the free cells can be washed out faster than they are produced. These problems may be overcome by immobilizing the cells onto a support matrix and hence,allow the cells to accumulate in the reactor. This mode of operation which is unique to growing immobilized cells in a continuous fermentation is known as the quasi-steady-state (QSS) whereby biomass increases at a prescribed growth rate while the substrate, precursor and product reach steady state concentrations.

Due to the dynamic characteristics of the process, the rate of approach to the QSS is very slow (Kalogerakis and Boyle, 1981). QSS-fed-batch cultures therefore cannot be achieved without a control system to force the process towards the QSS more rapidly. Boyle (1978) proposed a simple control scheme whereby the dilution rate and substrate feed concentration are manipulated to rapidly achieve the QSS. This control scheme was successfully tested by Kalogerakis (1981) on a baker's yeast fermentation.

The primary objective of the present study is the evaluation of a simple computer control scheme for continuous immobilized cell penicillin fermentations designed to control the specific growth rate, penicillin and precursor concentrations. In effect, the controller rapidly forces the system to the QSS. The evaluation is performed first

through computer simulations, and then experimentally. In the simulations an independently developed structured model was employed rather than the simple unstructured model used to develop the control algorithm. This allowed the control algorithm to be tested for robustness and the effect of inherent modelling errors. The experimental evaluation was performed using a fully instrumented 19 *L* bench top fermentor connected to a supervisory microcomputer through a front-end device for data acquisition. Before these closed loop experiments could be performed, a series of open loop experiments designed to determine the kinetic parameters were conducted.

CHAPTER II

LITERATURE REVIEW

2.1 Environmental Factors in Penicillin Fermentations

Microbial physiology deals with the interaction of the organism with its environment and the optimum expression of its genetic potential. The importance of physiology in the penicillin fermentation is illustrated by the 50-fold increase in overall yield from 1950 to 1975 (Pirt 1983). Over the same period, genetic changes account for a 16-fold increase in the maximum productivity of the organism. This means that roughly a 3-fold (i.e. 50/16) increase may be attributed to physiological and engineering factors. The physiological problem is to maintain the specific penicillin production rate (q_p) at the maximum value for as long as possible.

The present approach to physiological control centers on the influence of maintenance energy and growth rate on the metabolic activity of the producing strain as originally studied by Pirt and Righelato (1967). At a critical growth rate (μ) between 0.009 and 0.014 h^{-1}, they were able to maintain a high value for q_p. Below this critical growth rate, q_p was found to decay with time. Ryu and Hospodka (1980) determined the optimum specific growth rate that maintained the maximum penicillin productivity of their strain to be 0.015 h^{-1}. These results indicate that proper control of the growth rate could enhance penicillin productivity.

Recently, interest has been shown in methods of altering the morphology of penicillin mycelial cultures. By changing the morphology of the filamentous growth of cultures it is possible to improve energy dissipation within a reactor and facilitate the mass and oxygen transfer due to changes in the non-Newtonian rheological behavior of the fermentation broth. One possibility is to grow penicillin in a pelleted form. Metz and Kossen (1977) have classified three pellet structures:

a) fluffy loose pellets with a compact center and looser outer zone,

b) compact smooth pellets, and

c) hollow smooth pellets with the center being hollow due to autolysis.

Each of these structures would have a characteristic rheological behavior different from free filamentous growth. Konig et al. (1982) have found that the volumetric mass transfer coefficient, $k_L a$, for pelleted growth is 4-fold or higher than that of filamentous growth. Furthermore, cell maintenance requirements are drastically reduced and the glucose-to-product yield increases accordingly. Lorenz et al. (1987) have successfully produced penicillin-V in a tower loop reactor using a production strain of *P.Chrysogenum* in pellet form.

Pelleted growth, however, is influenced by several environmental and biological factors. Metz and Kossen (1977) list the following factors as having an effect on pellet formation and structure: agitation, growth medium, pH, oxygen tension, polymer

additives, surface-active agents, growth rate and inoculum. As a result, reproducibility of pellet formation is uncertain and considered more of an art.

2.2 Immobilized Cells in Penicillin Fermentations

As an alternative to pelleted growth, various cell immobilization techniques have been developed. These structures can be regarded as mycelial pellets since growth is primarily observed on the external surface of the beads. Thus, immobilized fungal cells can serve as a directed form of pelletization without the unpredictability of normal pellet formation (Jones et al., 1986).

Early experiments by Morikawa et al. (1979) with *P.Chrysogenum* cells immobilized in polyacrylamide gel did not prove successful due to harsh immobilization techniques and the toxicity of the materials used. The oxygen consumption for the immobilized cells was 70% lower than that for free cells, but this can be explained by either reduced cell viability, mass transfer problems or both. It was also found that the half-life of the immobilized biocatalyst was six times higher than that for free cells.

Deo et al. (1984) experimented with the immobilization of *P.Chrysogenum* cells in κ-carrageenan, a non-toxic natural polymer. They report a better than 9-fold increase in the half-life of penicillin production, and were able to maintain relatively constant specific penicillin productivity for periods greater than 15 days. Gbewonyo and

Wang (1983) developed a technique of confining mycelial growth to porous celite beads. They report final cell densities obtained with the confined cell cultures up to 60 *g/L*, nearly twice that attainable with free cell cultures. The confined growth cultures also showed consistently lower oxygen uptake rates than free cell cultures, especially during the later stages of the fermentation. Jones et al. (1986) compared surface immobilization onto celite particles, and entrapment in κ-carrageenan with respect to penicillin production by immobilized cells. They found the volumetric productivity to be five times higher for celite immobilized cells with the specific productivity for both support matrices being comparable to values found for free cells. More recently, Kim et al. (1986) used a three-phase fluidized-bed fermentor in the semi-continuous and fed-batch modes using celite anchored *P.Chrysogenum*. Cell growth and penicillin production were observed to increase significantly compared to conventional cultures. By phosphate limiting the culture to control bioparticle size, they were able to maintain a high level of productivity (about 80% of maximum) for at least a month. Their final concentrations of penicillin, however, were quite low, a common problem of such reports in literature (Swartz, 1985).

2.3 Mathematical Modelling of Penicillin Fermentations

In fermentation processes, engineering is only an aid in the development and regulation of biological processes; the microorganism being the center of attention. To properly manage these processes, a clear understanding of microbial growth kinetics is necessary. The

complicated nature of fermentation processes coupled with a poor understanding of the metabolic activities of antibiotic production makes rigorous modelling a difficult task. As a result, microbial growth models are of a simplistic nature concentrating on a small number of variables. A small number of variables can be justified using concepts from system dynamics. All intracellular events, including diffusion, protein synthesis and increases in cell number have characteristic relaxation times. Typically, only two or three system relaxation times have time scales similar to those of environmental changes, and so the dynamics of a large system can be approximated using a low order model with only two or three system variables.

Initial attempts to model microbial growth in batch fermentations were introduced by J.Monod in 1942. This model only shows how the system behaves under specific experimental conditions, not why. The main reason for this is that the model contains no mechanisms, being purely empirical in nature. Since this first attempt some important aspects of penicillin fermentations have been clarified allowing for the development of more successful models.

Several models for the penicillin fermentation have been suggested by various authors. These models can be divided into structured and unstructured categories. Unstructured models do not take into account the internal structure of the cell or changes in this structure. Unstructured models are often called lumped parameter models and are the most common found in the literature to date. Most of these

models have their basis in Monod or Contois kinetics coupled with modifications to account for maintenance energy, inhibition and other growth traits. Calam and Russell (1973) proposed a production model which is partly growth associated, but introduced a time delay in the expression of penicillin production in newly formed biomass. Constantinides et al. (1970) treated the process as a case of purely non-growth associated product formation and introduced a term for first order product decay. Pirt (1974) presented a model for fed-batch penicillin fermentations and introduced the concept of the "quasi-steady-state" in which the growth rate, μ, is equal to the dilution rate. Bajpai and Reuß (1980) developed a mechanistic model using Contois kinetics coupled with a substrate inhibition model to predict data for fed-batch penicillin fermentations, and using this model evaluated feeding strategies (Bajpai and Reuß, 1981). Cooney et al. (1977) and Heijen and Roels (1979) have applied component balancing methods in modelling whereby balances of atomic species are coupled with kinetic equations of a Michaelis-Menten-type relationship.

The above models are all unstructured in nature. In situations where the cell population composition changes significantly during the course of the fermentation, and in which these compositional changes affect growth kinetics, structured models should be used. Nestaas and Wang (1981) have shown that the ability of *P.chrysogenum* to synthesize and excrete penicillin changes during the course of a fed-batch fermentation. In order to account for the dynamic behavior of penicillin production and maintain simplicity, average cell age

concepts have been used (Kalogerakis et al., 1986). Using the cell types postulated by Megee et al. (1971), Nestaas and Wang (1983) developed a structured model for use with a computer control scheme to control fed-batch penicillin fermentations. This was possible through the development of a filtration probe to measure physiological differences taking place in the mycelial culture.

2.4 Computer Control of Penicillin Fermentations

Computer applications in biotechnology are not yet as widespread as in the chemical industry for several reasons: the lack of reliable on-line sensors to measure intracellular activities; the biosynthesis and regulation of metabolite formation is not fully understood; and a lack of good models for the biological systems involved. As a result computers are used primarily for data acquisition and analysis, and the control of simple operating conditions such as temperature, pH and DO concentration at desired levels. Recent discoveries involving penicillin fermentations have spurred interest in the area of computer control for this process. The importance of computer control in penicillin fermentations is illustrated by Lorenz et al. (1987) who were able to decrease the aeration of a tower loop reactor (the only energy input) by an average of 75% through the use of process control.

Several studies have attempted the optimization of the penicillin fermentation through open-loop feedforward control to achieve an optimal biomass concentration trajectory. These studies

attempt to reproduce a biomass growth history which would result in a maximization of the penicillin produced. These methods are hampered by two inherent problems: the reliability of models used to predict the optimal trajectory; and the well known changes in microbial behavior under seemingly identical environmental conditions. As a result, some form of feedback control is necessary which in turn depends on the ability to measure or estimate key process variables.

Early work by Svrcek et al. (1974) applied the use of extended Kalman filtering techniques to control a model of a continuous fermentation. Stephanopoulos and San (1984) demonstrated the importance of estimator robustness (such as the Kalman filter) in the face of measurement uncertainty and model inaccuracies. They show that even with accurate initial estimates of inoculum concentration, large and diverging variances and state estimates occur. Montague et al. (1986) also use an extended Kalman filter in the control of penicillin fermentations to follow a predetermined biomass trajectory using a self-tuning control algorithm. The controlled variable, biomass, was estimated from measurements of the carbon dioxide in the exit gas and calculation of the CO_2 production rate. Nestaas and Wang (1983) achieved closed-loop feedback control of the penicillin fermentation by estimating biomass concentration with a filtration probe that had been previously developed (Nestaas and Wang, 1983).

All the control schemes developed for penicillin fermentations have been of the "single input single output" (SISO)

type. This means that only one variable is being controlled by manipulation of one of the input variables (e.g. growth rate is controlled by manipulating the glucose feeding rate). Penicillin fermentations, however, are very complex and it is obvious that optimization is not possible by controlling growth rate alone. Penicillin concentration should be kept at a constant high value to decrease downstream processing costs, and precursor concentration should be maintained at a predetermined level to ensure availability but avoid toxicity. It has also been suggested that there are optimum levels for NH_4^+ concentration as well as for other ions (Pirt, 1985).

Linardos (1987) proposed a simple non-interactive control algorithm for simultaneous control of growth rate, penicillin concentration and precursor concentration and tested this algorithm using computer simulation studies. The proposed algorithm attempts to rapidly force precursor, penicillin and substrate concentrations to their quasi- steady state concentrations. The algorithm allows for the possibilty of choosing a desired steady state concentration which once achieved can be maintained.

CHAPTER III

UNSTRUCTURED MODEL AND CONTROLLER DESIGN

Modelling is the first stage in the application of modern control theory to biological systems. Penicillin fermentations are very complex and rigorous modelling is impossible at this point due to an incomplete understanding of the mechanisms involved. For control purposes however, a simple model which preserves the main dynamic characteristics of the system is all that is necessary given that the operating conditions are well established. The model developed here is based on the work of Kalogerakis et al. (1986). The governing variables of cell growth, product formation and precursor uptake are modelled using simple differential equations based on the appropriate mass balances.

3.1 Unstructured Model

The dynamic behavior of the fermentation is modelled with differential equations assuming a constant reactor volume. The unsteady state mass balances for biomass, glucose, penicillin and the precursor phenoxyacetic acid are as follows:

$$\frac{dx}{dt} = \mu x \tag{1}$$

$$\frac{ds}{dt} = -\sigma x + D(s_f - s) \tag{2}$$

$$\frac{dp}{dt} = q_p x - Dp \tag{3}$$

$$\frac{dz}{dt} = -bq_p x + D(z_f - z) \tag{4}$$

where x, s, p, and z are the biomass concentration (dry weight), limiting substrate (glucose), penicillin-V and precursor (PoAA)

concentration in the fermentor respectively (*g/L*). The subscript "f" denotes feed stream concentration and the dilution factor, D (volumetric feed rate/liquid volume, h^{-1}) does not directly affect biomass concentration as the cells are immobilized and accumulate in the reactor.

The specific growth rate, μ (h^{-1}) follows typical Monod kinetics:

$$\mu = \frac{\mu_{max} s}{(k_s + s)} \tag{5}$$

where μ_{max} is the maximum specific growth rate and k_s is the substrate saturation constant (*g/L*). The specific uptake rate, σ (h^{-1}) is related to the growth rate through the following expression:

$$\sigma = \mu/Y_G + m + q_p/Y_p \tag{6}$$

where Y_G is the cell growth yield (*g-cells/g-glucose*), m is the maintenance energy requirement (*g-glucose/g-cells,*h^{-1}) and Y_p is the penicillin from substrate yield (*g-penicillin/g-glucose*).

To account for the dynamic behavior of the penicillin production rate average cell age concepts have been used. The average cell age, λ, is obtained from the following differential equation (Fishman and Birykov, 1974):

$$d\lambda/dt = 1 - \mu\lambda \tag{7}$$

The specific penicillin production rate, q_p, is then related to average cell age by:

$$q_p = q_p^{max} a\lambda \exp(1 - a\lambda) \tag{8}$$

The optimum value of λ where q_p attains its maximum value, q_p^{max}, is determined by the adjustable parameter "a".

The assumptions made in the formulation of the proposed model are as follows:

a) The total volume in the reactor stays constant at all times.

b) No spatial variations exist within the reactor.

c) Reaction rates are slow compared to nutrient and mass transfer rates.

d) A negligible number of free cells are present.

e) Glucose is the only limiting substrate.

f) Dissolved oxygen remains above critical levels.

3.1.1 The Quasi-Steady-State

As the growing biomass remains in the reactor at all times it is impossible to reach a true steady state. However, a Quasi-Steady-State (QSS) can be achieved where s, p and z remain constant with time while the biomass slowly accumulates at the set growth rate, μ. At the QSS equations (1) to (4) and (7) can be simplified to these expressions:

$$x(t) = x(0)\ \exp(\mu t) \tag{9}$$

$$D(s_f - s) = \sigma x(t) \tag{10}$$

$$q_p x(t) = Dp \tag{11}$$

$$D(z_f - z) = bq_p x(t) \tag{12}$$

$$\lambda = 1/\mu \tag{13}$$

In order to operate the system at a desired QSS, the feed concentrations of glucose and precursor (s_f and z_f) should remain constant while the feed rate increases exponentially and these values are calculated according to these relationships:

$$s_f = s + \sigma p/q_p \tag{14}$$

$$z_f = z + bp \tag{15}$$

$$D(t) = D(0)\ \exp(\mu t) \tag{16}$$

$$D(0) = q_p x(0)/p \tag{17}$$

In summary, once a desired QSS is chosen, input values for feed concentrations and feed rate are easily calculated.

3.2 Controller Design

Due to the nature of the system, the QSS is reached after a long period of time and once achieved, only remains at this state for a short period of time as the system reaches its oxygen transfer limitations. It is therefore desirable to speed up the transient response so that the QSS can be achieved sooner so as to ensure maximum penicillin productivity. The basic control algorithm shown here was originally proposed by Kalogerakis *et al.* (1986) and described in detail by Linardos (1987).

3.2.1 Control Objectives

The control system is designed to control penicillin and precursor concentrations, and the specific growth rate. It is desirable to force the penicillin concentration to a high preset value and maintain that value while ensuring that the penicillin productivity

remains high through control of the specific growth rate. Furthermore, the level of precursor should be kept high enough to ensure availability but also low enough to prevent possible inhibition of penicillin productivity. These requirements suggest two objectives:

a) Fast penicillin transients should be ensured without sacrificing the rapid response of the growth rate.

b) The interaction between controlled variables on the transients should be minimized.

3.2.2 Selection of Manipulated Variables

To avoid the problem of interaction between controlled variables, proper selection of the manipulated variables is necessary. This can be accomplished by looking at the pertinent differential equations and choosing the appropriate variable. If we look at equation (3) we notice that only the dilution factor, D, has an affect on penicillin concentration and so is the obvious choice of manipulated variable. Equation (2) suggests that both dilution rate and glucose feed concentration can be used to manipulate the glucose concentration (specific growth rate). As the dilution factor has been chosen to control penicillin concentration, glucose feed concentration, s_f is chosen. Finally, equation (4) suggests that precursor concentration can also be controlled by both feed concentration, z_f, and dilution rate. By the same reasoning applied for the control of growth rate, the precursor feed concentration is chosen as the manipulated variable.

It is obvious from the above choices for the manipulated

variables that there is a degree of interaction between D and s_f, and D and z_f. This interaction can be reduced by giving s_f and z_f as wide as possible a range in values. More rigorously, the selection of D as the manipulated variable to control p and hence, s_f and z_f to control μ and z is suggested by the relative gain array when computed after linearization of the governing equations around a typical QSS.

3.2.3 Control Algorithm

To speed up the penicillin transients and force the penicillin concentration toward the desired QSS the dilution rate needs to be kept to a minimum. Once the desired concentration has been achieved the dilution rate can be continuously adjusted to maintain this level. Since direct measurement of the penicillin concentration is possible through HPLC, a simple feedback controller is sufficient. A simple Proportional plus Integral (PI) controller is used.

Simulations have indicated that the growth rate transients are fast and there is no need to speed them up. As a result a steady state controller can be employed. Since μ is primarily a function of substrate concentration in the reactor as seen in equation (5), constant "s" implies constant μ. Because substrate concentration cannot be measured directly, an approximate control law for s_f is based on the steady state relationship of equation (2):

$$s_f = s_d + \hat{\sigma} x/D \tag{18}$$

where s_d is the desired glucose concentration in the fermentor (estimated) and $\hat{\sigma}$ is the estimated uptake rate at the desired QSS. To

account for modelling errors which are present, a simple feedback term is introduced to adjust $\hat{\sigma}$ slowly to maintain the desired growth rate:

$$\hat{\sigma}_c = \hat{\sigma} + K_{cs}(\mu_d - \mu) \tag{19}$$

where K_{cs} is the controller gain and μ_d is the desired growth rate.

The final controller equation is therefore:

$$s_f = s_d + \hat{\sigma}_c x/D \tag{20}$$

With the given feedback correction term, the controller can maintain μ at its desired value despite possible errors in parameter estimation or drift in the parameters should Contois kinetics apply rather than the assumed Monod kinetics.

Similar to the design of the growth rate controller, the precursor controller is based on the steady state form of equation (4):

$$z_f = z_d + b\hat{q}_c x/D \tag{21}$$

where z_d is the desired precursor concentration in the fermentor and $\hat{q}_c$ is the corrected specific production rate as calculated using a simple feedback term:

$$\hat{q}_c = \hat{q}_p + K_{cz}(z_d - z) \tag{22}$$

K_{cz} is the controller gain and $\hat{q}_p$ is the estimated specific production rate at the desired QSS.

3.2.4 Constraints on the Manipulated Variables

The manipulated variables D, s_f and z_f must at all times remain within their physical limits as defined by pump characteristics

and media solubilities. As a result the following constraints are imposed:

$$D_{min} \leq D \leq D_{max} \tag{23}$$

$$0 \leq s_f \leq s_{fmax} \tag{24}$$

$$0 \leq z_f \leq z_{fmax} \tag{25}$$

A further constraint is imposed upon the dilution rate by the selection of a maximum glucose feed concentration if the steady state growth rate controller is to be satisfied at all times. Equation (20) shows that an upper limit on s_f implies a lower limit on D so that

$$D_{minc} = \hat{\sigma}_c x/(s_{fmax} - s_d) \tag{26}$$

The final expression for the constraint of dilution rate is

$$\max\{D_{min}, D_{minc}\} \leq D \leq D_{max} \tag{27}$$

This lower bound varies with biomass and so is computed as part of the control calculations.

An additional constraint could be imposed upon the dilution rate by the precursor feed concentration in much the same manner. Exact control of precursor concentration is not demanded however, so the constraint is placed on the maximum z concentration rather than on the dilution rate. By doing so, any unnecessary extra dilution is avoided at the cost of slightly sluggish response of the precursor controller.

3.2.5 Growth Rate Estimation

Unlike penicillin and precursor concentration, specific growth rate cannot be measured directly and so must be estimated from measurements of biomass. These measurements in turn are subject to

sampling errors and so their use in growth rate estimation requires that each value be weighted to filter out inconsistencies. Furthermore, rapid response is desired but a reponse to an erroneous measurement could be harmful. To circumvent this problem, an Exponentially Discounted Recursive Least Squares algorithm (EDRLS) has been employed. This algorithm allows for the compensation of poor data measurements by interfacing a process mathematical model with statistical weighting of the known input measurements. By using a "forgetting" factor, λ_c, more attention is paid to new data than old data and by varying λ_c, a compromise between sluggishness and the ability to filter measurement errors can be achieved.

The EDRLS algorithm used for estimating specific growth rate from dry biomass measurements is accomplished in four steps as follow.

1) Prediction of the natural logarithm of the expected biomass using previously obtained values.
2) Estimation of the Kalman filter gains.
3) Parameter update.
4) Covariance matrix update.

Mathematically this is expressed as follows:

$$\hat{y}_i = \mu_{i-1} t_i + c_{i-1} \tag{28}$$

$$R = \lambda_c + P_{11} t_{i-1}^2 + 2P_{12} t_{i-1} + P_{22} \tag{29}$$

$$K_1 = (P_{11} t_{i-1} + P_{12})/R \tag{30}$$

$$K_2 = (P_{12} t_{i-1} + P_{22})/R \tag{31}$$

$$\mu_i = \mu_{i-1} + K_1(y_i - \hat{y}) \tag{32}$$

$$c_i = c_{i-1} + K_2(y_i - \hat{y}) \tag{33}$$

$$P_{11} = (P_{11} - K_1K_1R)/\lambda_c \tag{34}$$

$$P_{12} = (P_{12} - K_1K_2R)/\lambda_c \tag{35}$$

$$P_{22} = (P_{22} - K_2K_2R)/\lambda_c \tag{36}$$

where subscript "i-1" denotes the previous sample time, K_1 and K_2 are the Kalman filter gains, c is the natural logarithm of the biomass, $\hat{y}$ is the predicted value and matrix P is the covariance matrix. The entire control loop is initiallized with set-point and initial values and the initial values for the covariance matrix are set in accord with the confidence in the initial values.

CHAPTER IV

Controller Evaluation Through Simulation

A quick and easy method of evaluating the effectiveness of the proposed control algorithm is through computer simulation. Rather than performing these simulations using the mathematical model upon which the control algorithm is based, a separately derived set of differential equations describing the system more qualitatively are used. These equations make up a "structured" kinetic model of the immobilized penicillin fermentation.

4.1 Structured Model

As an alternative to using the average age concepts to account for the dynamic behavior of penicillin production, a simplistic form of structure accounting for different cell types has been postulated by Nestaas and Wang (1983). The cell types are:

A_0 - hyphae tips describing the growth geometry of mycelial cultures,

A_1 - healthy cells completely filled with cytoplasm capable of penicillin production,

A_2 - degenerated non-producing cells having lost their cytoplasmic material.

Figure 1 schematically illustrates the three cell types. Total biomass concentration, x, is the sum of a_0, a_1, and a_2 which are the concentrations (g/L) of cell types A_0, A_1 and A_2 respectively.

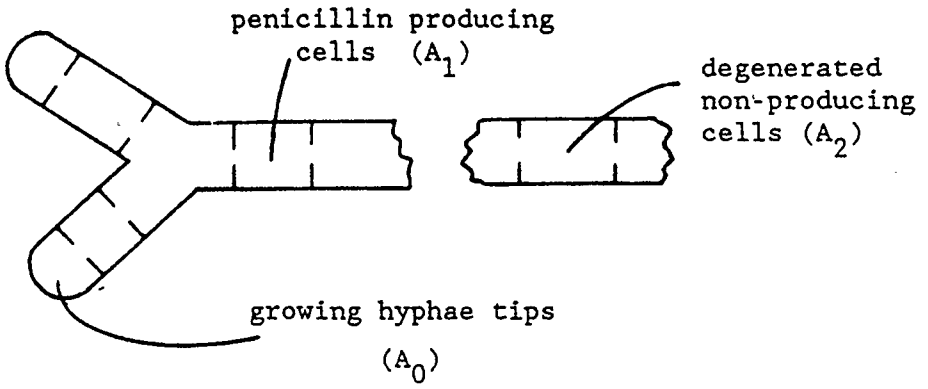

Figure 1 Postulated cell types in the structured model.

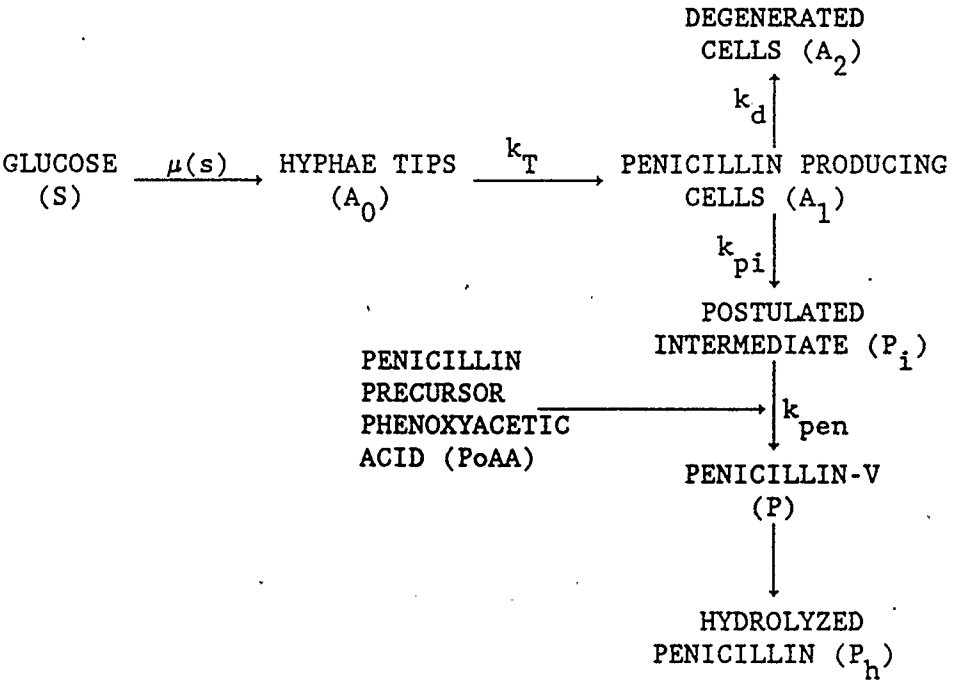

Figure 2 Postulated reaction scheme for the structured model.

The dynamics of cell growth and penicillin production is illustrated in figure 2. All new cell growth is a result of hyphal branching to produce new A_0-type cells at a first order rate, $\hat{\mu}$ (h^{-1}). These hyphae tips elongate at a rate k_T (h^{-1}) to evolve into penicillin producing A_1-type cells. At the same time, active cells are continually degenerating to become A_2-type cells at a rate k_d (h^{-1}). Mass balances on the various cell types for a continuous reactor (assuming cells are immobilized and the cells remain in the fermentor at all times) are as follows:

$$da_0/dt = \hat{\mu}a_0 - k_T a_0 \tag{37}$$

$$da_1/dt = k_T a_0 - k_d a_1 \tag{38}$$

$$da_2/dt = k_d a_1 \tag{39}$$

where it is assumed that the branching rate of the hyphae tips follows the usual Monod kinetics:

$$\hat{\mu} = \frac{\hat{\mu}_{max} s}{(k_s + s)} \tag{40}$$

The only cells capable of producing penicillin are the A_1-type cells. The first stage in production is the formation of a postulated intermediate, p_i, which accumulates in the cells before penicillin can be produced. This intermediate is a lumped parameter of all the reactions occurring inside the cell and, as such, gives some physiological meaning to the known delay in penicillin production by introducing a metabolic bottleneck. Since penicillin is unstable in aqueous solution, a term for product hydrolysis has been included in the model. Mass balances on the precursor, penicillin, intermediate and

hydrolized product are represented by the following rate equations:

$$dp_i/dt = k_{p_i} a_1 - k_{pen} p_i z \tag{41}$$

$$dp/dt = k_{pen} p_i z - k_h p - Dp \tag{42}$$

$$dp_h/dt = k_h p - Dp_h \tag{43}$$

$$dz/dt = D(z_f - z) - k_{pen} p_i z \tag{44}$$

where k_{pi} and k_h are the first order rate constants for maximum specific synthesis and penicillin hydrolysis, and k_{pen} is the first order rate for precursor conversion into penicillin. Substrate consumption is similar to the expression found in the unstructured model except for the distinction of energy requirements for the different cell types as shown below:

$$ds/dt = D(s_f - s) - Q_s \tag{45}$$

$$Q_s = \hat{\mu} a_0/\hat{Y}_G + \hat{m}(a_0 + a_1) + k_{p_i} a_1/\hat{Y}_p \tag{46}$$

where $\hat{Y}_G$, $\hat{m}$, and $\hat{Y}_p$ are the cell yield, maintenance energy and product yield respectively for the structured model. As with the unstructured model, a QSS can be achieved. However, the development of the QSS relationships is more complicated and is described in the following section.

4.1.1 The Quasi-Steady-State

At the QSS, $\hat{\mu}$ reaches a constant overall growth rate $\bar{\mu}$ due to the constant level of substrate present and so equation (37) can be written as:

$$da_0/dt = (\bar{\mu} - k_T)a_0 \tag{47}$$

$$\int_{a_0(0)}^{a_0(t)} da_0 = \int_0^t (\bar{\mu} - k_T)dt \tag{48}$$

$$a_0(t) = a_0(0)\ \exp[(\bar{\mu} - k_T)t] \tag{49}$$

Substituting equation (49) into equation (38) yields:

$$da_1/dt + k_d a_1 = a_0(0)k_T\ \exp[(\bar{\mu} - k_T)t] \tag{50}$$

which can be solved by finding the homogeneous and particular solutions and combining them as follows:

-homogeneous solution

$$da_1^{hom}/dt = -k_d a_1^{hom} \tag{51}$$

$$a_1^{hom} = c^* \exp(-k_d t) \tag{52}$$

-particular solution

$$a_1^{part} = \omega_0\ \exp\{(\bar{\mu} - k_T)t\} \tag{53}$$

The constant ω_0 is obtained by substitution of equation (53) into equation (50) yielding

$$d[\omega_0 \exp\{(\bar{\mu}-k_T)t\}]/dt+k_d\omega_0\exp\{(\bar{\mu}-k_T)t\} = k_T a_0(0)\exp\{(\bar{\mu}-k_T)t\} \tag{54}$$

which results in

$$\omega_0(\mu-k_T)\exp\{(\bar{\mu}-k_T)t\}+k_d\omega_0\exp\{(\bar{\mu}-k_T)t\} = k_T a_0(0)\exp\{(\bar{\mu}-k_T)t\} \tag{55}$$

and hence,

$$\omega_0 = \frac{k_T a_0(0)}{\mu - k_T + k_d} \tag{56}$$

Combining the homogeneous and particular solutions we obtain

$$a_1(t) = a_1^{hom} + a_1^{part} \tag{57}$$

which, upon substitution of equations (52) and (53) yields

$$a_1(t) = c^* \exp(-k_d t) + \frac{k_T a_0(0)}{\mu - k_T + k_d} \exp\{(\bar{\mu} - k_T)t\} \tag{58}$$

The integration constant is obtained from the initial conditions at $t = 0$ yielding

$$c^* = a_1(0) - \frac{k_T a_0(0)}{\mu - k_T + k_d} \tag{59}$$

At later times ($t > 4/k_d$), the first exponential tends to zero and equation (58) may be simplified to

$$a_1(t) \approx \frac{k_T a_0(0)}{\mu - k_T + k_d} \exp\{(\bar{\mu} - k_T)t\} \tag{60}$$

which when divided by equation (49) yields the following correlation:

$$\frac{a_1(t)}{a_0(t)} \approx \frac{k_T}{\mu - k_T + k_d} \tag{61}$$

Using the simplified form of equation (60), equation (39) may be written as

$$\frac{da_2}{dt} = k_d a_1 \approx \frac{a_0(0) k_d k_T}{\mu - k_T + k_d} \exp\{(\bar{\mu} - k_T)t\} \tag{62}$$

which yields

$$a_2(t) = \left[\frac{a_0(0) k_d k_T}{\mu - k_T + k_d}\right]\left[\frac{1}{\mu - k_T}\right] \exp\{(\bar{\mu} - k_T)t\} \tag{63}$$

The measured biomass is the sum of all three cell types,

$$x(t) = a_0(t) + a_1(t) + a_2(t) \tag{64}$$

Substitution of the previously derived expressions produces

$$x(t)=\left[a_0(0)+\left(\frac{a_0(0)k_T}{\mu-k_T+k_d}\right)+\left(\frac{a_0(0)k_Tk_d}{\mu-k_T+k_d}\right)\left(\frac{1}{\mu-k_T}\right)\right]e^{(\bar{\mu}-k_T)t} \tag{65}$$

Equation (45) can be written as

$$\frac{ds}{dt} = D(s_f-s) - \left[\frac{\bar{\mu}}{Y_G} + m\left(1 + \frac{a_1}{a_2}\right) + \frac{k_{pi}a_1}{Y_p a_0}\right]a_0 \tag{66}$$

From equation (59), the constant ratio of A_1 to A_2 type cells is

$$c_1 = \frac{a_1}{a_0} \approx \frac{k_T}{\mu - k_T + k_d} \tag{67}$$

At the QSS, ds/dt = 0 and hence ,from equation (66) we obtain

$$s_f = s(t) + \frac{a_0(t)}{D(t)}\left[\frac{\bar{\mu}}{Y_G} + m(1 + c_1) + \frac{k_{pi}}{Y_p}c_1\right] \tag{68}$$

where,

$$D(t) = D(0)\exp[(\bar{\mu} - k_T)t] \tag{69}$$

If we evaluate the above equations at t = 0, we obtain the following:

$$s_f = s + \frac{a_0(0)}{D(0)}\left[\frac{\bar{\mu}}{Y_G} + m(1 + c_1) + \frac{k_{pi}}{Y_p}\, c_1\right] \tag{70}$$

In addition, at the QSS $dp_i/dt = dp/dt = dp_h/dt = dz/dt = 0$. Consequently, by selecting desired values for "p" and "z", we can find z using equation (44) and the above relationships.

$$z_f = z(t) + k_{pen}p_i(t)z(t)/D(t) \tag{71}$$

4.2 Open Loop Simulations

Both the unstructured and structured models have been simulated using Advanced Continuous Simulation Language (ACSL, Mitchell and Gauthier, Assoc., Inc.) on a CDC-175 computer. The ACSL programs can be found in Appendix C.

Figure 3 illustrates the QSS achieved by the unstructured model. Penicillin and precursor concentration in this plot are just barely at the desired QSS when the fermentation ends at the high biomass concentrations showing the long time required for the system to reach the steady state. The desired steady-state concentrations of penicillin, precursor and glucose were achieved by fixing the feed concentrations and initial dilution rate (s_f = 50 *g/L*, z_f = 1.0 *g/L*, D(0) = 0.004593 h^{-1}). The nominal process parameter values used in all simulations involving the unstructured model are: μ^{max} = 0.123 h^{-1}, k_s = 1.0 *g/L*, Y_G = 0.5, m = 0.026 h^{-1}, Y_p = 1.20, b = 0.3, a = 0.0145 h^{-1}, and q_p^{max} = 0.0017 hr^{-1}. These parameter values are similar to those found in the literature except for the value used for q_p^{max} which was selected to match values determined experimentally.

Figure 4 illustrates the growth of the three cell types for the structured model with time. An open loop simulation of the structured model was performed to illustrate differences between the structured and unstructured models. Figure 5 shows the QSS achieved by the structured model. Slightly different steady-states are achieved by the structured model due to modelling and parametric differences. The

behavior of the precursor concentration appears to reach a lower steady state before rising to its final value which is lower than the steady state achieved by the unstructured model. Substrate concentration reaches a higher steady state than in the other model and shows a longer period of overshoot. Finally, the penicillin curve shows an initial lag in production and then goes on to achieve a slightly higher steady state concentration than the other model. The nominal process parameter values used in simulations involving the structured model come partially from the experimental values determined by Nestaas and Wang (1983) and partially from estimated values due to differences from their model. They are: $Y_G = 0.45$, $Y_P = 0.1$, $m = 0.022\ h^{-1}$, $K_T = 0.006\ h^{-1}$, $K_{Pi} = 0.008\ h^{-1}$, $K_{pen} = 1.3\ h^{-1}$, $K_h = 0.003\ h^{-1}$, $K_s = 1.0\ g/L$, $\mu^{max} = 0.123\ h^{-1}$, and $K_d = 0.0008\ h^{-1}$.

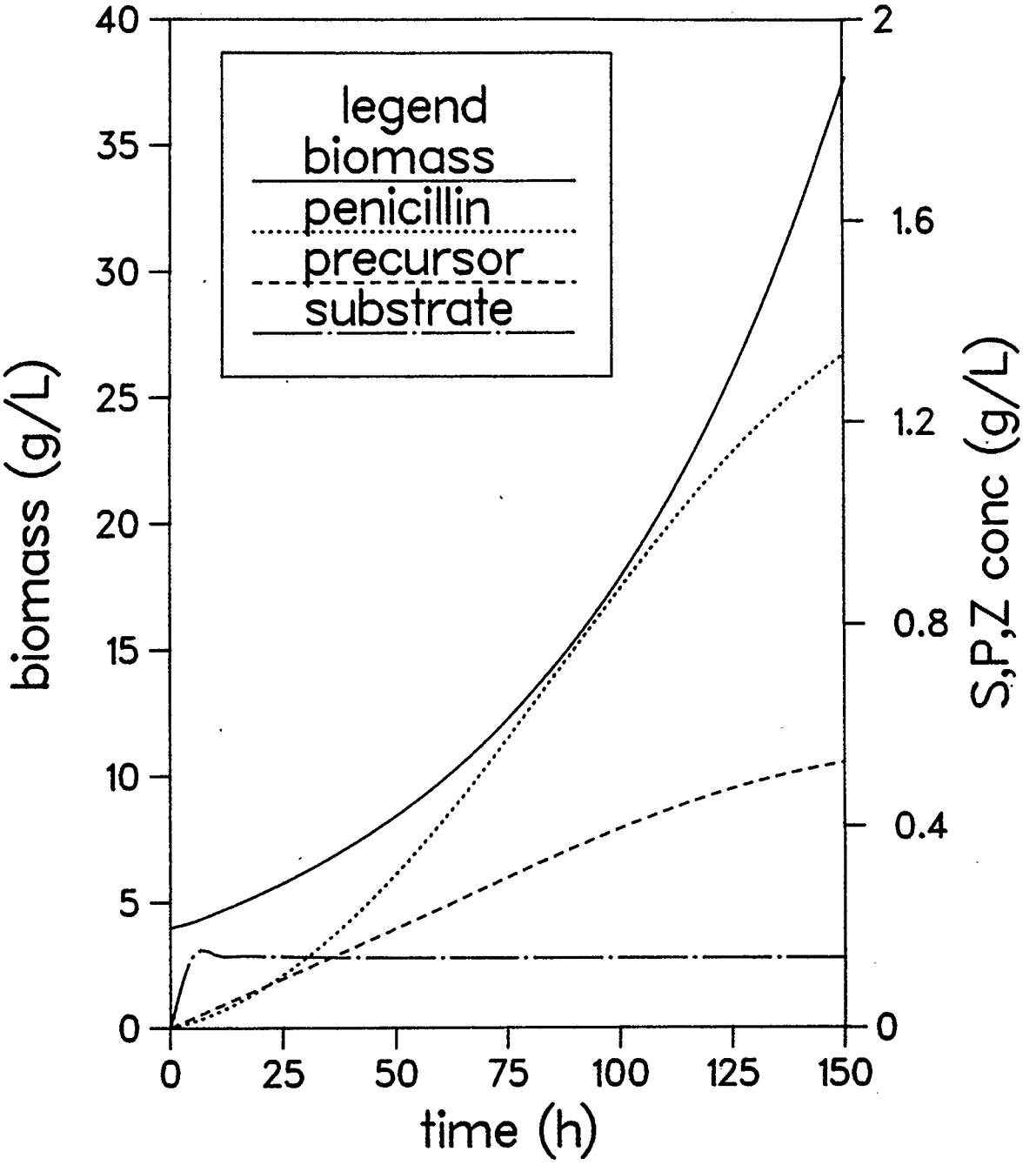

Figure 3 Simulated results of a penicillin fermentation using an unstructured model and illustrating the long period of time required to obtain QSS conditions.

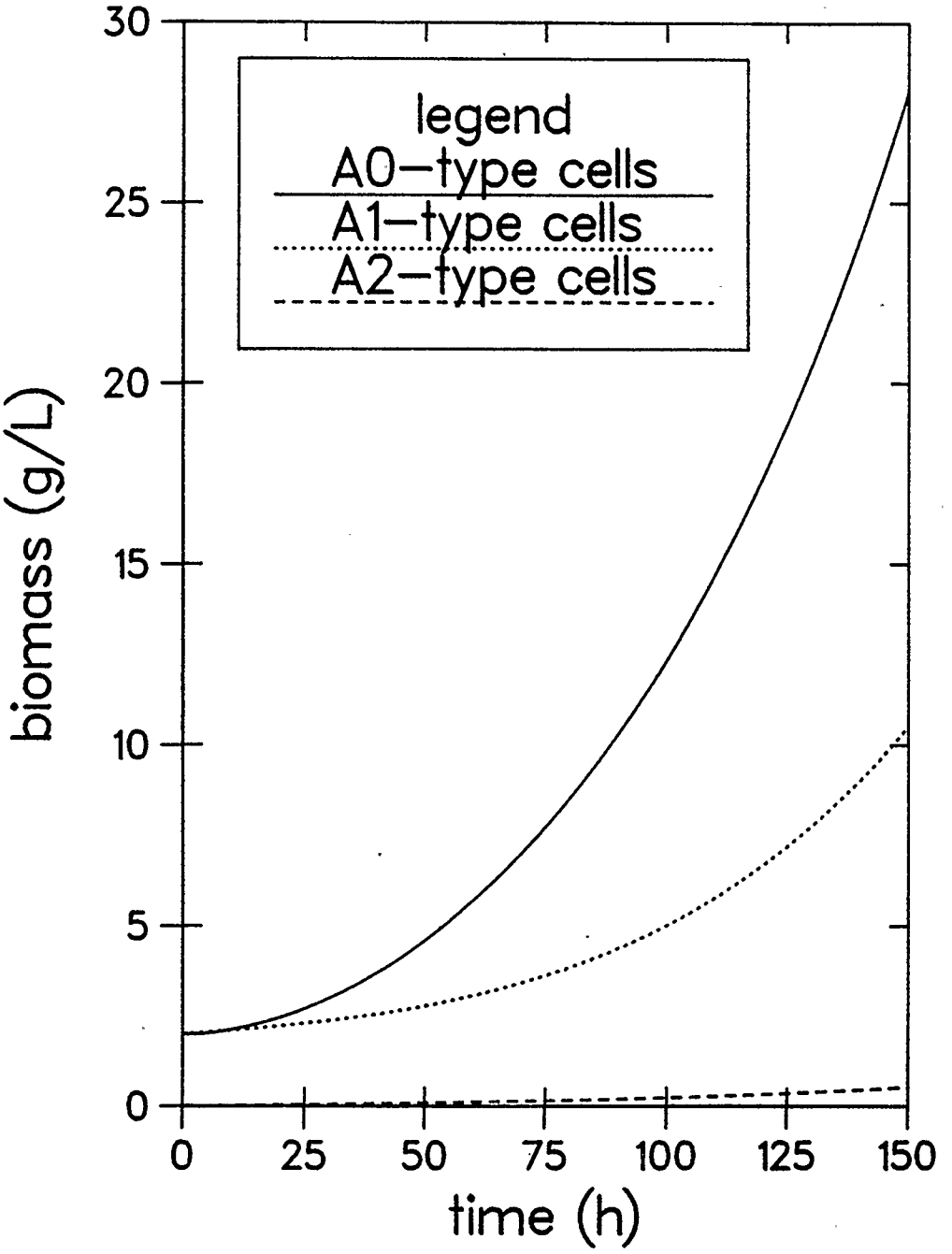

Figure 4 Plot of biomass accumulation for a penicillin fermentation simulated using a structured kinetic model and illustrating different cell types.

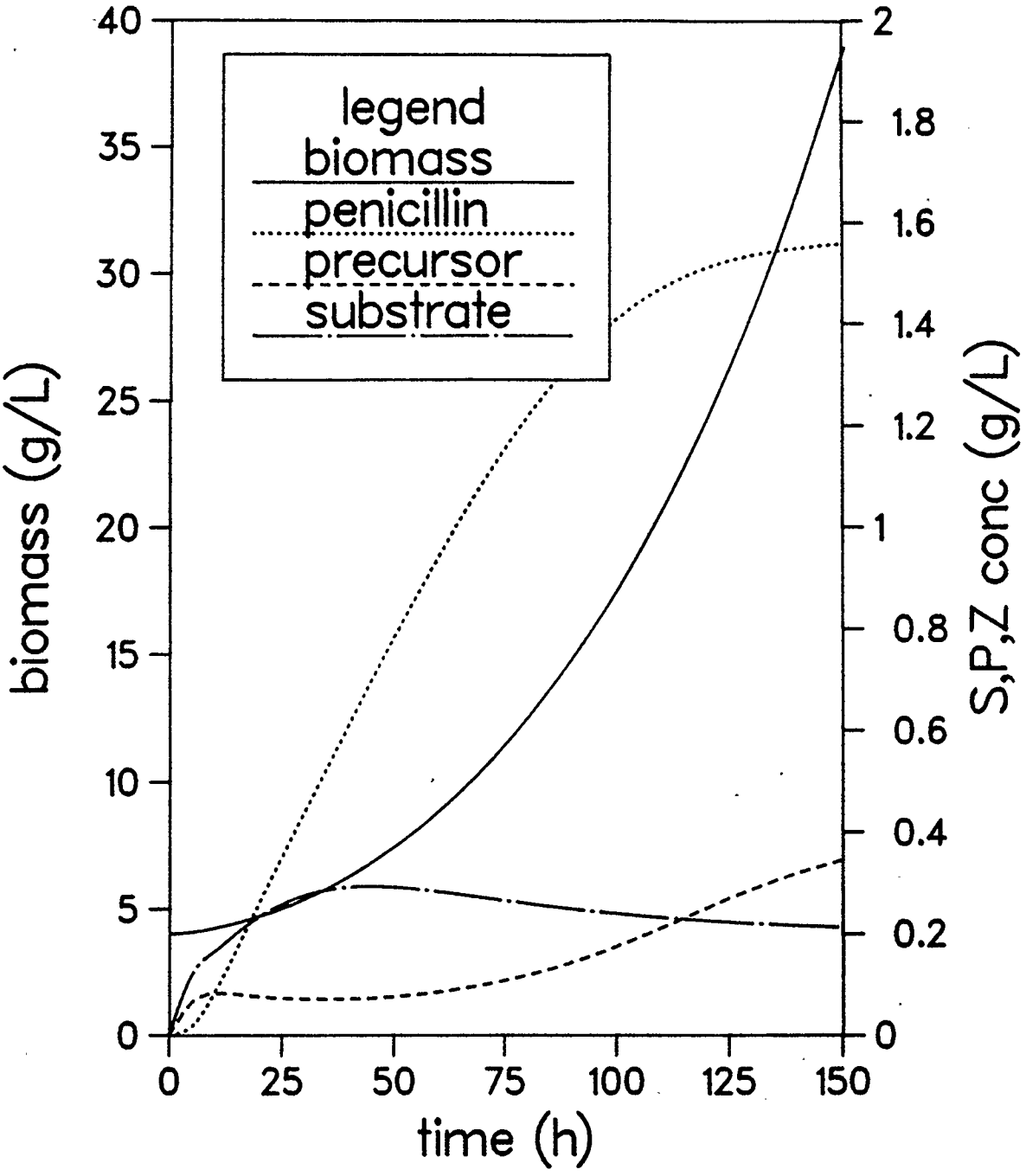

Figure 5 Simulated results of a penicillin fermentation from a structured kinetic model and illustrating the QSS.

4.3 Closed Loop Simulations

For the purposes of tuning and evaluation, the control algorithm has been coupled with the proposed models and simulated using ACSL. These simulation programs can be found in Appendix C. So that the simulated results match experimental conditions, a 4 hour sampling interval with a 90 minute deadtime between sampling and data input has been introduced, and a random number generator with a 10% spread about the expected biomass value has been added to simulate sampling error and other forms of "noise".

The control algorithm was tuned using response to step inputs to the system and determining the integral of the absolute error (IAE tuning) for the unstructured model upon which the control algorithm is based. Fig. 6 illustrates how well a constant growth rate can be achieved and controlled under these conditions. This plot shows that initially, however, there is an adjustment period before the growth rate achieves the set point due to control action to erroneous input. This period only lasts for 25 hours after which the covariance matrix in the growth rate estimator is sufficiently small to filter out noise. The value for growth rate measured by the Kalman filter and the "actual" growth rate as determined by the substrate concentration in the fermentor seem to diverge with time as was predicted by Stephanopoulos and San (1984). Figure 7 shows how the precusor concentration is forced to its set point and also how rapidly the penicillin concentration climbs in an attempt to reach the desired set point (4 *g/L*). A high set point was chosen simply so that the maximum

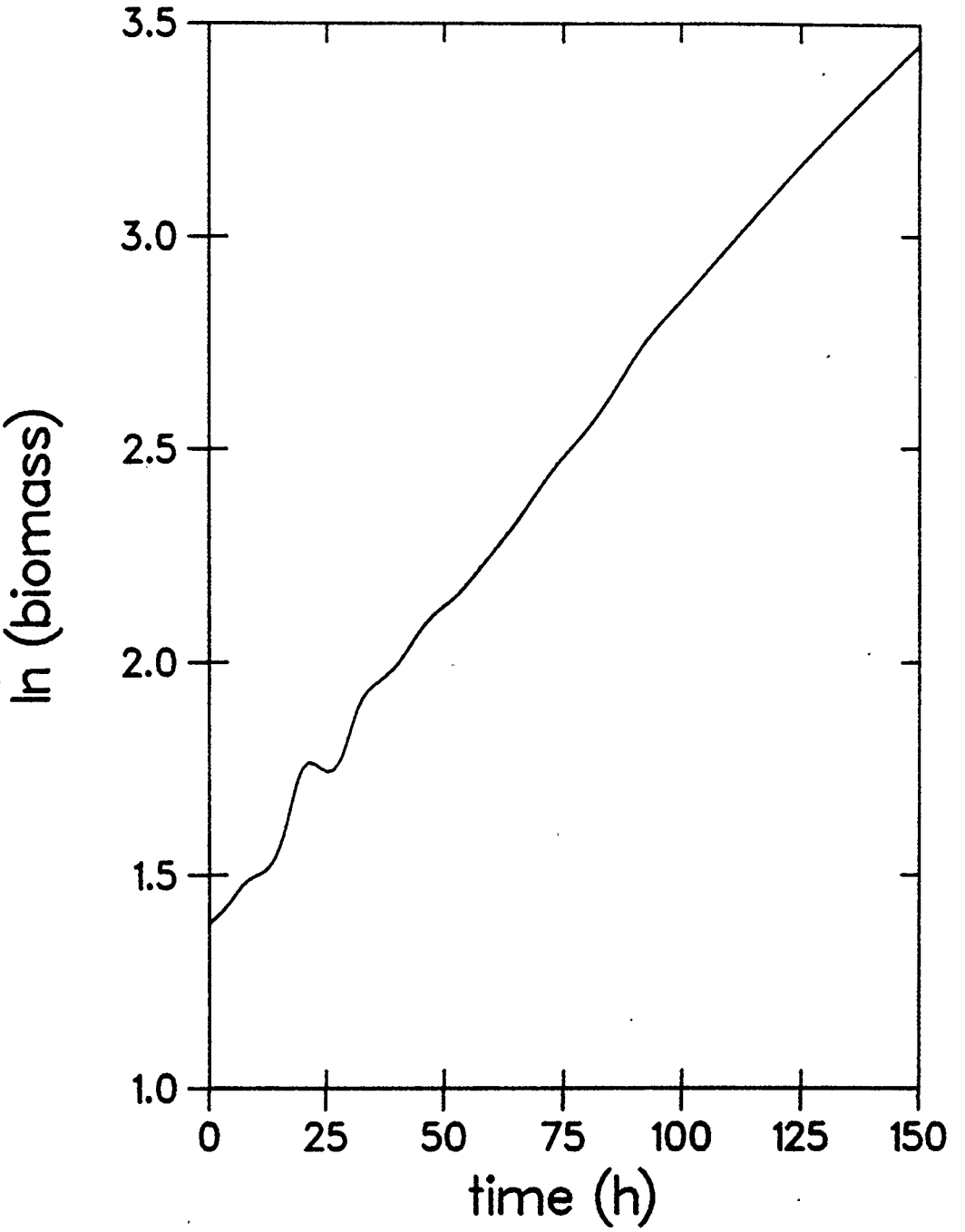

Figure 6 Plot of ln(biomass) vs time for the unstructured model generated by simulation under closed loop control conditions illustrating the constant growth rate achieved under control.

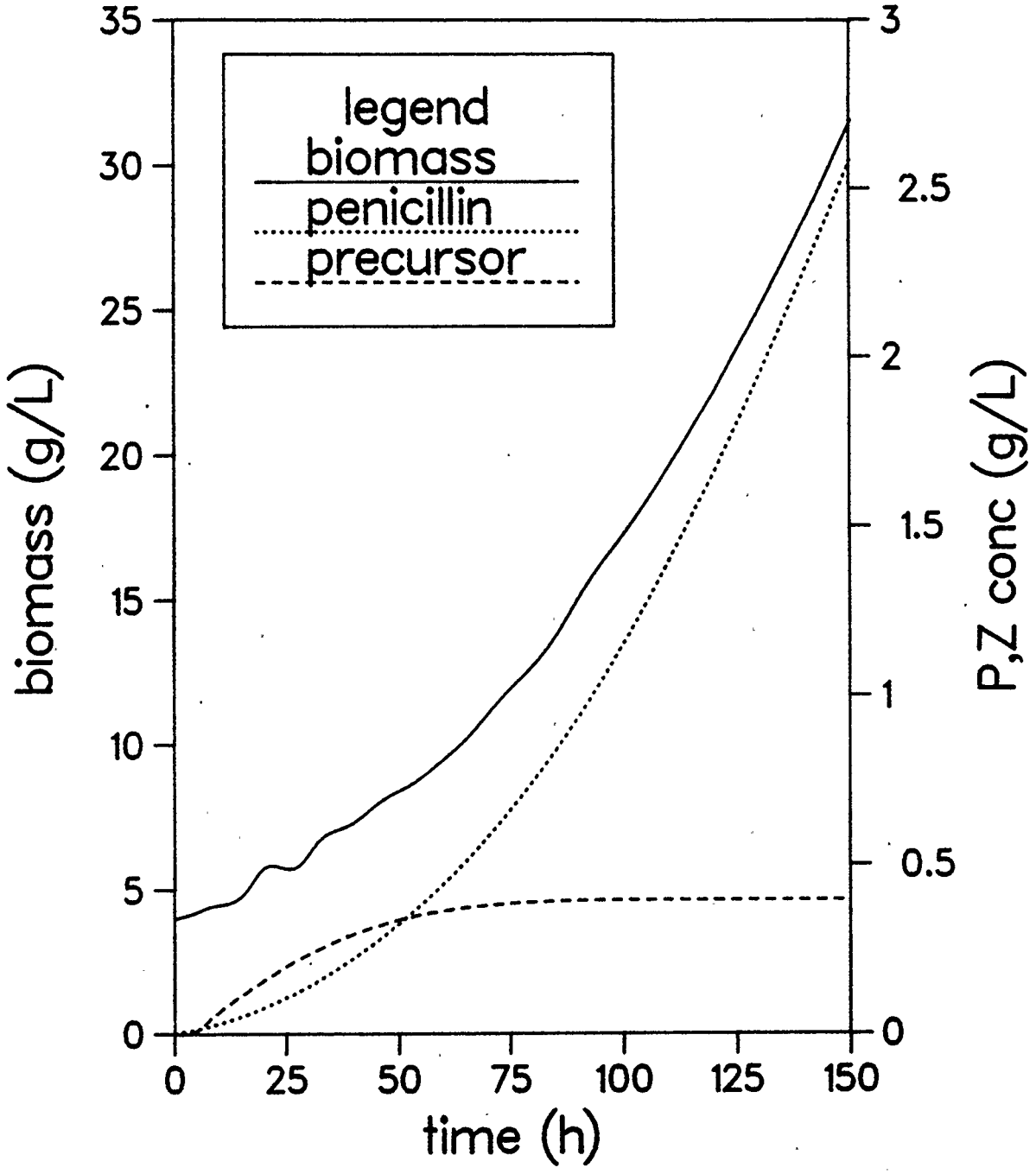

Figure 7 Plot of the biomass, precursor and penicillin concentrations achieved by the unstructured model under closed loop control as simulated using ACSL.

penicillin titer could be achieved under the fixed operating conditions. Under controlled conditions the penicillin concentration reached a final titer of over 2.5 *g/L*, which is almost twice that achieved in the previous open loop simulation. It should also be noted that the glucose concentration varies initially as the Kalman filter is more sensitive to early errors in measurement until the covariance matrix gets smaller to indicate some confidence in the measured values.

Next, the control scheme was tested for robustness by testing it with the structured growth model. Fig. 8 once again shows that control of growth rate about a linear operating set point is possible with the same initial adjustment period. It seems that the algorithm is most sensitive to errors in biomass measurements early in the run where a period of transient over-feeding supplies a large amount of substrate to the biomass resulting in a rapid increase in growth. The final biomass achieved is much lower than that expected given the amount of substrate fed. This illustrates a major difference between the two models. Despite this, control of the growth rate was successful, though there is an offset from the operating set point, possibly due to the lack of integral control. Figure 9 illustrates yet another difference in the two models. The precursor concentration initially rises rapidly towards the set point and then declines as penicillin production increases, and the maximum value for precursor feed concentration is not great enough to match it. Despite this, the final penicillin titer was approximately 50% higher than in the open loop simulations and so the main objective of the proposed control scheme was achieved. The

reason for the increased titers is the decreased dilution of the penicillin from the reactor.

In summary, it can be seen that the control scheme proposed by Kalogerakis *et al.* (1986) is implementable and robust enough to be able to handle modelling errors. Simulations on the proposed algorithm indicate some problems in controlling growth rate early in the run due to sampling errors and the large dead time encountered. However, the main objective of the control scheme; that is increasing penicillin, production is achieved.

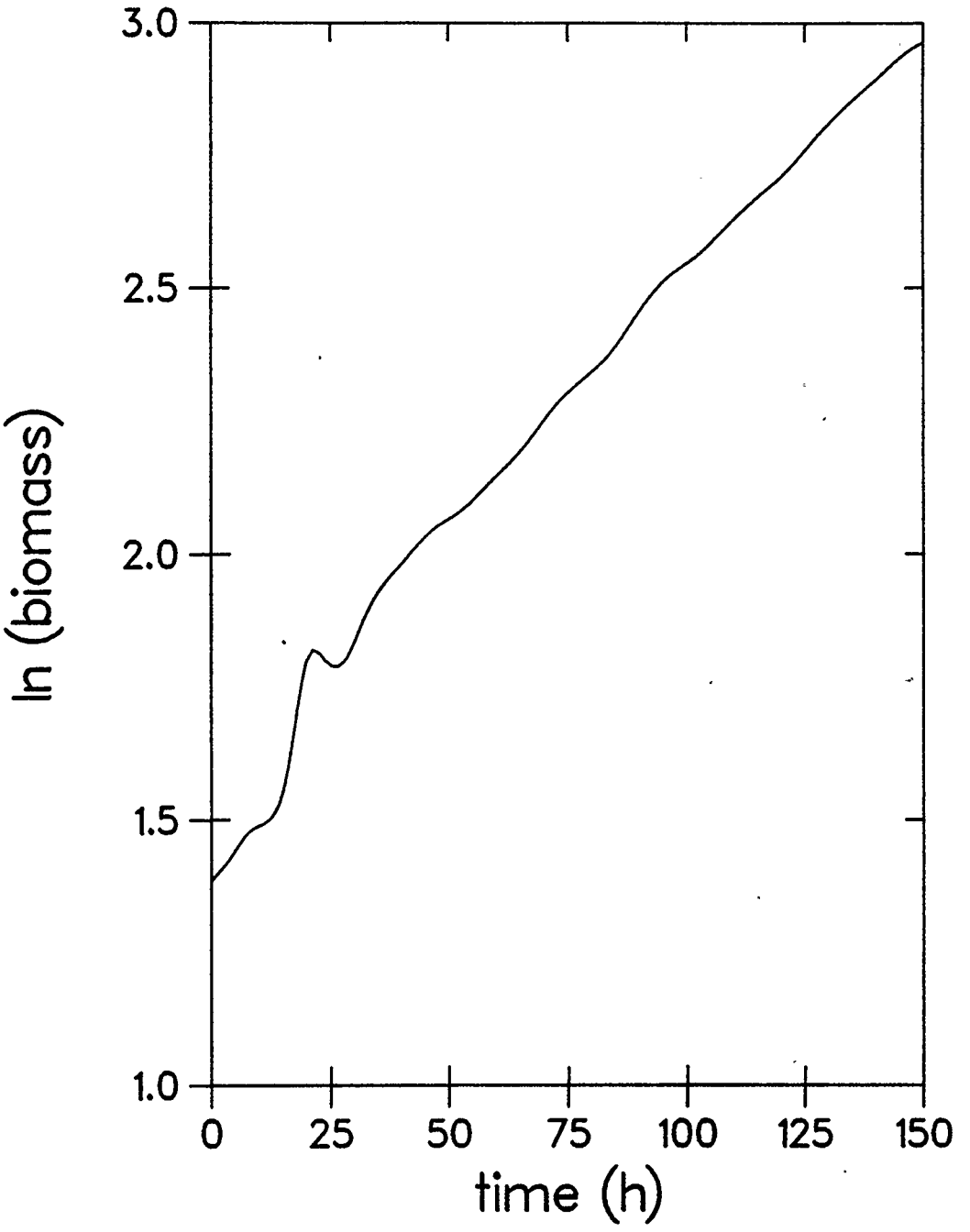

Figure 8 Plot of ln(biomass) vs time for the structured model simulated by ACSL under closed loop control illustrating the constant growth rate achieved under control.

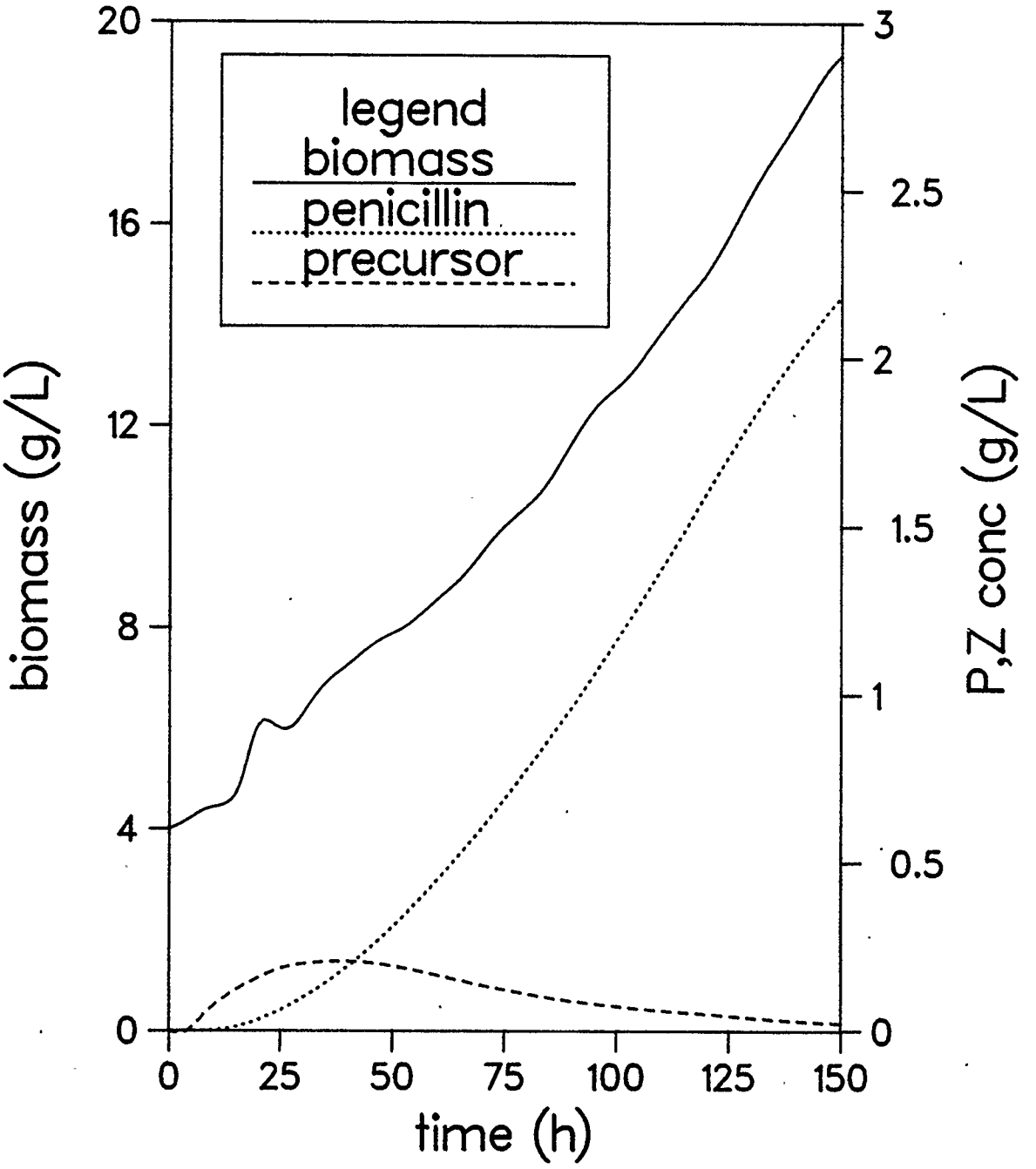

Figure 9 Plot of the biomass, penicillin and precursor concentrations for the structured model simulated using ACSL.

CHAPTER V

EXPERIMENTAL APPARATUS

5.1 FERMENTATION HARDWARE

In the following sections each of the major components of the apparatus is fully described. A schematic diagram of the entire apparatus is shown in Figure 10 and photographs of the actual apparatus are shown in Figures 11a and 11b.

5.1.1 The Fermentor

A 19 *liter* BIOENGINEERING AG fermentor was used. The main reactor body is made of thick PYREX glass to enable *in-situ* sterilization. The internal diameter of the glass cylinder is 22 *cm* and it stands 40 *cm* high, making the volumetric capacity approximately 15 *liters*.

The lower portion of the reactor consists of a double-walled stainless steel bowl that adds an additional 4 *L* to the reactor volume. Temperature control of the fermentor is facilitated by the circulation of warm or cold water through the walls of this section of the vessel.

The impeller shaft enters the reactor through a center opening in the bottom jacket. Aseptic operation is ensured by a mechanical seal. Rotation speeds up to 2000 *RPM* can be achieved with an electric motor located beneath the reactor vessel. Various types of impellers can easily be fitted on the shaft for various operating

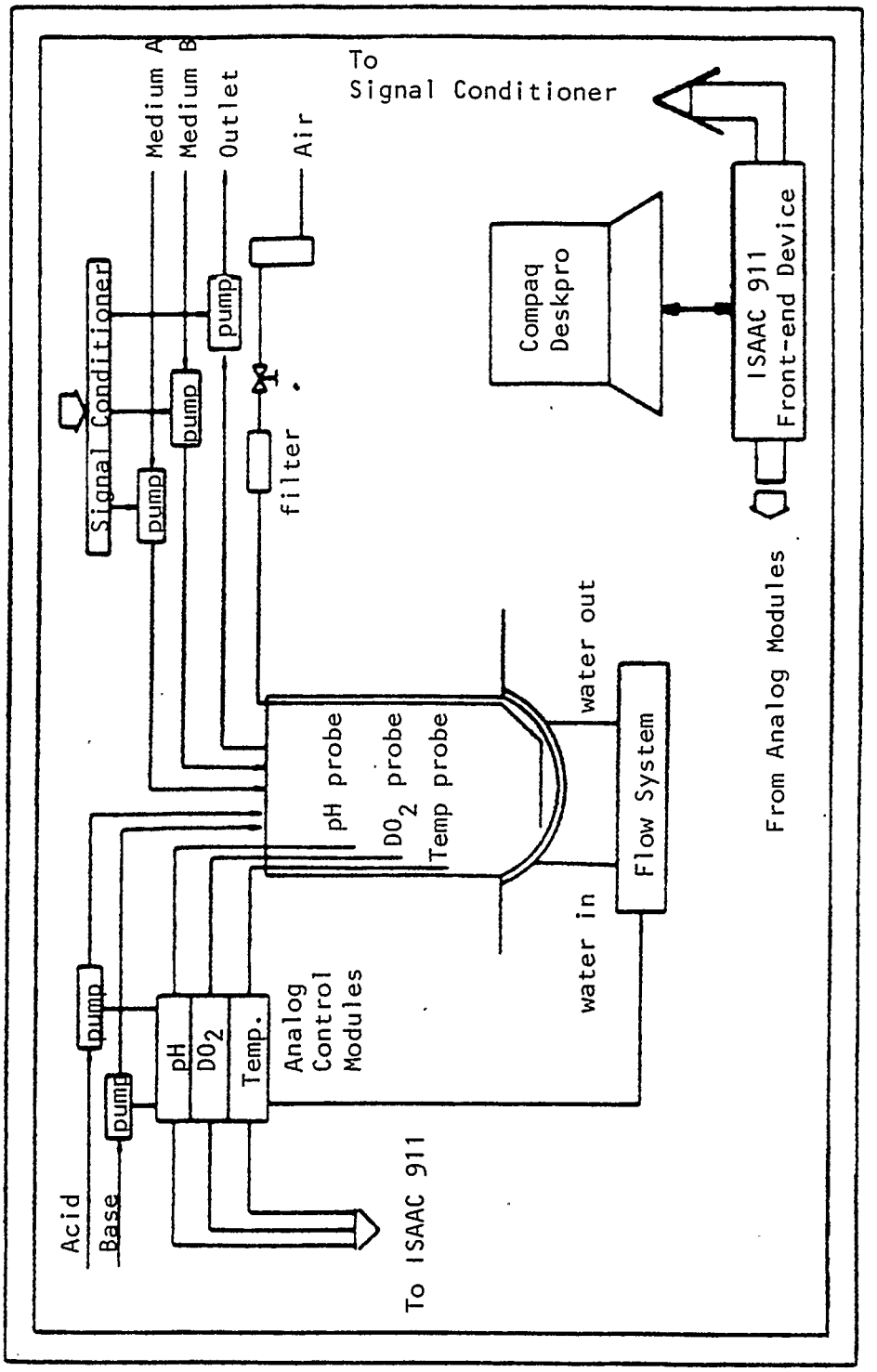

Figure 10 Schematic drawing of the experimental apparatus. (From Linardos, 1987, MSc Thesis, University of Calgary, page 69)

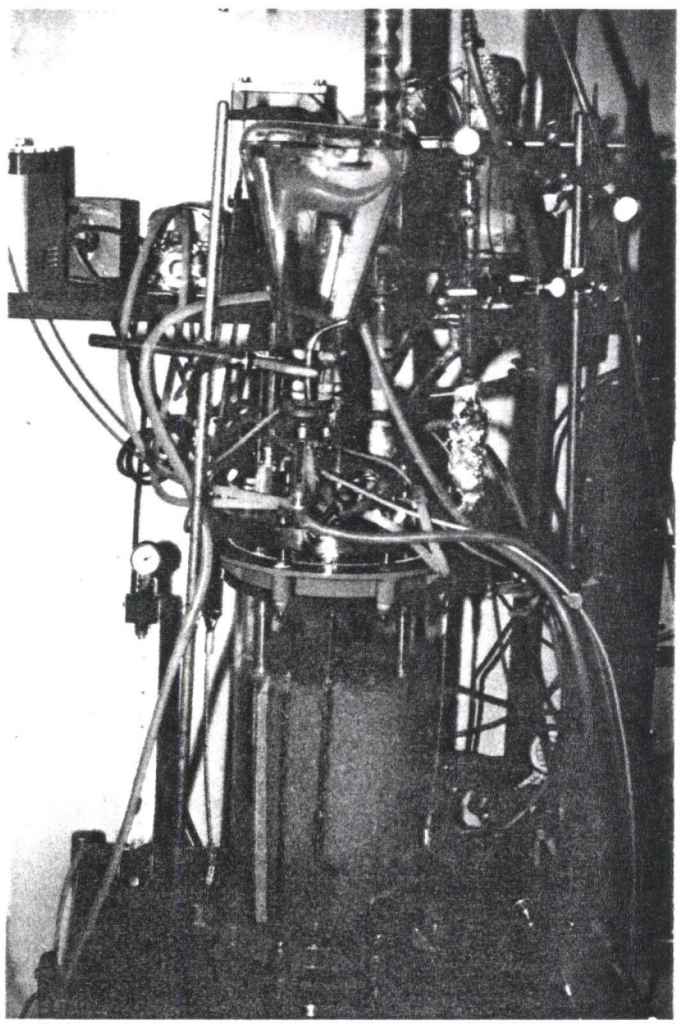

Figure 11a Photograph of the experimental apparatus showing the operating fermentor.

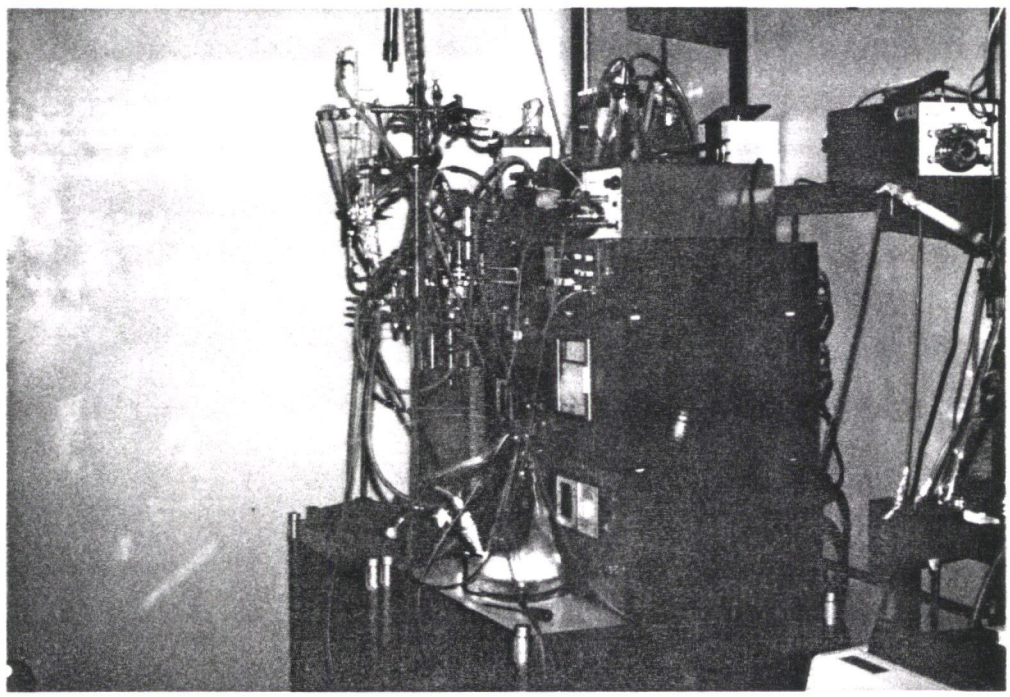

Figure 11b Photograph of the experimental apparatus showing the local controllers.

conditions.

The reactor head plate is a 2 *cm* thick stainless steel with twelve specially designed ports for air-tight aseptic operation through which probes, medium and air lines can be passed. A safety valve set at 1.2 *bar* is fitted on the central port to protect the glass from overpressurization during sterilization. The port assignment for typical open loop and closed loop runs is shown in Figure 12.

Compressed air at a regulated pressure of 1.5 *bar* passes through a rotameter and then is sterilized through a ceramic filter. A stainless steel tube carries the sterile air to a spiral-shaped sparger at the reactor bottom which is wrapped in nylon to disperse fine bubbles evenly throughout the fermentation broth. The nylon also acts to disperse some of the force of the pressurized air passing through the holes in the sparger, and protect the cells from the force of the air jets.

5.1.2 Analog Measurements and Control

The fermentor was outfitted with independent BIOENGINEERING AG measurement and control units for temperature, pH and dissolved oxygen. Recorded output was achieved via a 4-20 *mA* signal which was tied into the microcomputer for data acquisition. These independent units were also able to receive remote set points making them suitable for use as local controllers in a supervisory computer control scheme.

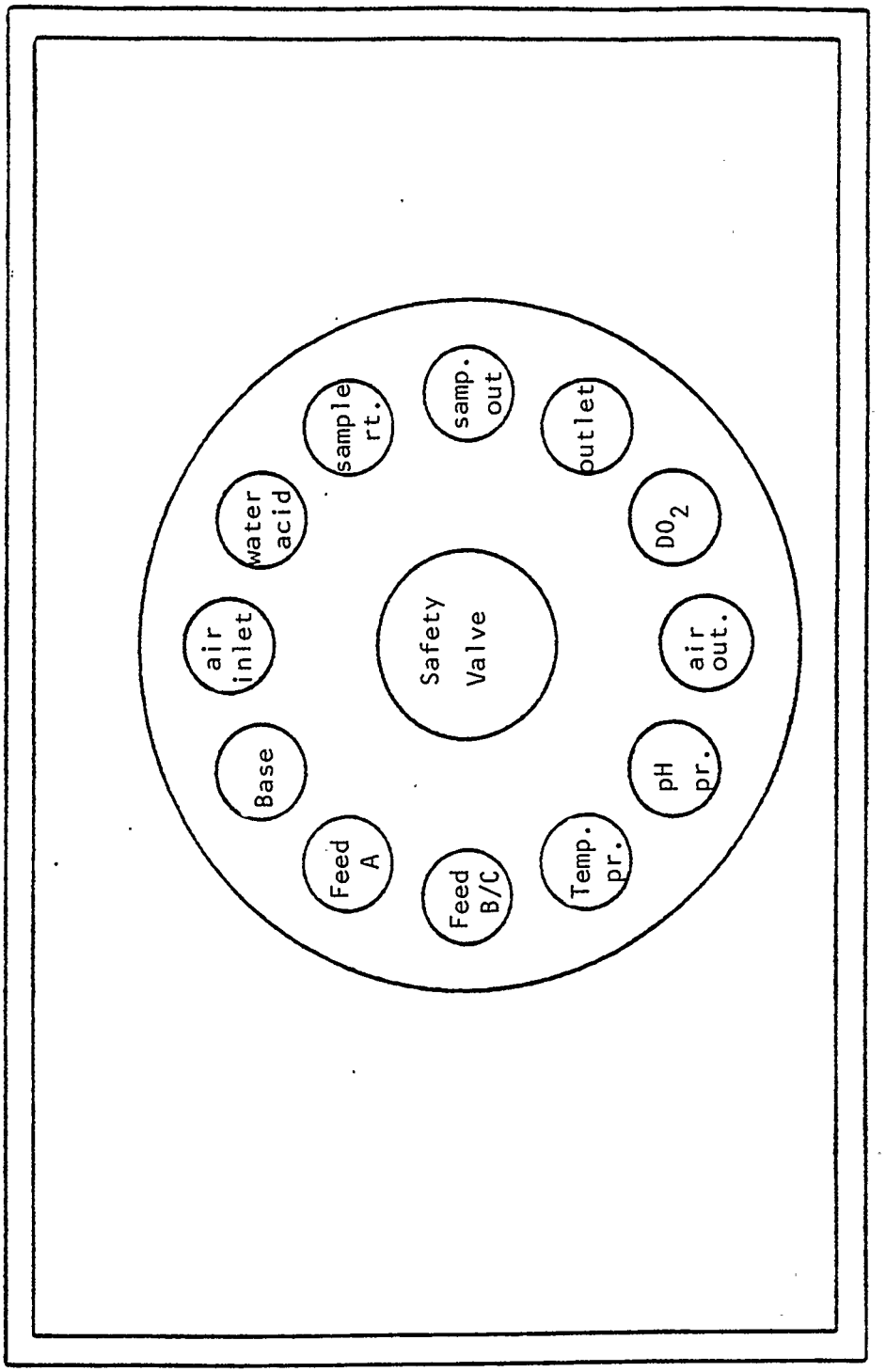

Figure 12 Fermentor head-plate port assignment during a typical open loop run. (From Linardos, 1987, MSc. Thesis, University of Calgary, page 67)

Temperature control was achieved via a PHILLIPS electronic PID control unit which was accurate to within ± 0.1°C of the set point. Cooling was achieved by circulating cold tap water through the jacketed bowl. This water tap was closed when heating was required and the recirculated water was passed through an on/off heater. The regulated temperature range was from 5 degrees above tap water temperature to a sterilization temperature of 125°C.

The pH was measured using an INGOLD steam sterilizable combination probe and controlled from a PI controller which operated an acid (2 N HCl) or base (4 N NaOH) pump depending on need. The pH was controlled to within ± 0.02 units of the set point for the KH_2PO_4 buffered medium.

The DO concentration was measured via the surface tension with an INGOLD IL501 steam sterilizable polarographic oxygen electrode connected to the monitor. The oxygen concentration in the reactor was then controlled manually by increasing the air flow rate and by supplementing the feed with pure oxygen late in the run when demand was high.

5.2 COMPUTER CONTROL HARDWARE

5.2.1 Computer and Peripherals

The supervisory control microcomputer was a COMPAQ Deskpro with two 360 *Kbyte* floppy disk drives and a 10 *Mbyte* hard disk. A RAM

memory space of 640 *Kbytes* was available for running the control software. A STAR SR-10 dot matrix printer was connected via a parallel interface card and a serial interface card was used for communication with the other units. The microcomputer was also furnished with a COMPAQ Deskpro keypad and monochrome monitor.

5.2.2 Computer-Process Interface

To facilitate communication between the supervisory computer and the process an ISAAC 91I front-end device was employed and included the following:

- A/D and D/A convertor
- 16 analog inputs, 8 outputs
- 16 digital inputs, 16 outputs
- frequency counter
- 4 Smith triggers

Analog input signal ranges from 0-10 *mV* to 0-10 *V* were possible. The 4-20 *mA* signals coming from the independent control units were converted to 0-10 *V* signals by using one 500 Ω resistor in parallel to the ISAAC analog input. The analog outputs were 0-10 *V* signals, converted to 4-20 *mA* signals by a signal conditioner (Cyborg Corp.), and used to drive the peristaltic pumps. The ISAAC 91I was supported by software written in machine language with subroutines accessed from BASIC and run at the maximum speed allowed by the hardware.

5.3 CONTROL SOFTWARE

All software written to be used with the COMPAQ personal computer used the BASIC programming language. The software performs the two functions of data acquisition and control with two programs developed for open and closed loop control of the fermentation. These programs were initially developed by Linardos (1987) and subsequently modified to fit the requirements of the present study. The complete program listings can be found in Appendix C.

5.3.1 Program Hierarchy

Once the program has been initiated it has been designed to operate in one of three levels, with each level assigned a certain priority. At most times, the program remains in the display or top level which graphically displays important information while the computer remains idle.

While the computer is in the display mode, the operator may elect to interrupt the top level by using one of the ten function keys to perform a special task. This ability to interrupt the display mode allows for communication between the computer and the operator. Upon completion of a given function, the terminal automatically returns to the top level.

The highest execution priority is given to the time interrupt or control mode which supercedes any operating level at a given time interval as calculated by the internal real time clock. When the control algorithm has been executed, the level at which the system had

been operating is resumed.

5.3.2 Subroutines

The main subroutine is the screen display subroutine previously described as the lowest operating level. The main function of this algorithm is to provide the operator with an easy-to-read graphic display of important operating parameters.

The time interrupt level has its control calculations organized into the "servicing ISAAC" subroutine. This subroutine performs the calculations for the controlled variables in both the open and closed loop programs with some inherent differences. As well, input and output values to and from the ISAAC front-end device are calculated to determine temperature and DO measurements and the signal to be sent to a given pump. Calculated data is then sent to the printer to record operating conditions.

Under the key interrupt mode there are ten special subroutines which differ between the open and closed loop algorithms. Beside the function keys labeled <F1> to <F10>, the "^BREAK" key exits and finishes the experimental run and if the "NUM LOCK" key is on the "^BREAK" key simply interrupts the program and allows the operator complete freedom without leaving the program itself. The ten function keys are as follows:

F1 ——→ help display

F2 ——→ clear screen

F3 ——→ change sampling interval

F4 ——→ status request

F5 ——→ plot process variable information

F6 ——→ rearrange pumps

F7 ——→ change operating parameters

F8 ——→ linear regression routine

$F8_c$ ——→ pump flow control routine

F9 ——→ enter measurements

F10 ——→ change set points

It should be noted that in the closed loop program, a pump flow calibration correction has been assigned to <F8> rather than the linear regression routine. Also in the case of the closed loop program, under key interrupt <F9> the measured variables are entered, and the feedback control calculations are performed along with the calculations for estimating specific growth rate.

5.3.3 Intermittent Pump Flow

In an attempt to lower the physical pumping rate of the peristaltic pumps and, consequently, the dilution rate, an intermittent pump flow routine was added. Each 20 minute update interval was subdivided into ten 2 minute pumping intervals during which the pump was either on or off. The number of intervals for which the pump was on was determined by dividing the desired flowrate by the minimum flow rate and multiplying by ten. This value, rounded to the nearest

integer, represented the number of pumping intervals. To account for rounding errors, these differences were tracked and accumulated so that they could be accounted for in the next update interval. Although this method does not ensure a "constant" growth rate at this level, the overall growth rate as observed should remain constant. In so doing, a low dilution rate is ensured to maximize penicillin titers.

CHAPTER VI

EXPERIMENTAL PROCEDURES

6.1 Bench Scale Experiments

The following is a description of the experimental procedures followed for the operation of the 19 *L* BIOENGINEERING AG fermentor and all of the supplemental procedures involved in the operation of the three-phase fluidized-bed immobilized cell penicillin fermentation.

6.1.1 Strain Preservation and Propagation

The original culture of *Penicillium Chrysogenum* E15 (ATCC 26818) was acquired from Eli Lilly Co. Ltd. as lyophillized spores stored on dried milk powder and vacuum sealed. Every six months fresh spores were lyophillized and stored on silica gel to ensure strain preservation. To obtain a large enough inoculum for the fermentation, 15 cm^2 slants of sporulation medium (appendix A) were streaked and incubated at 26°C for 10-14 days to ensure complete sporulation. Spores were removed by washing the slants with a 50 *mg/L* aerosol OT solution and then transferred to a larger Fernbach flask (9000 cm^2) coated with sporulation medium. This was incubated a further 10-14 days at 26°C at which point 3 to 4 of the Fernbach flasks were sufficient to inoculate the 19 *L* fermentor. These spores were also removed with a 50 *mg/L* aerosol solution and yielded a typical spore count of 10^9 *spores/mL*. This suspension was sonicated for 30 *min* to break apart any spore clumps.

6.1.2 Pretreatment of Celite

Celite R630 (30 x 50 *mesh*, average diameter 450 μm) from Johns Manville Co. Ltd. was used as the immobilization matrix. Initially the celite was placed in a muffle furnace at 500°C for greater than eight hours to burn off any organic contaminants. The celite was then washed several times with distilled water to remove any fines, and dried in an oven at 110°C. The desired amount of celite for an experiment was then weighed into a Pyrex glass carbuoy and autoclaved for one hour.

6.1.3 Spore Immobilization and Inoculation

After the spore suspension collected from the Fernbach flasks was sonicated, it was transferred to a 3 *L* Erlenmeyer flask which was connected to the carbuoy containing sterile celite. After being diluted to an appropriate volume for the given amount of celite, the spore suspension was drained into the carbuoy. The carbuoy with the attached Erlenmeyer was then placed on a rotary shaker for one hour at 100 *RPM* to provide good even contact of the spores with the celite. When this was complete the celite was rinsed with three to four volumes of distilled water and the rinse water collected. The difference in spore counts between the initial spore suspension and that in the wash was taken as the total spore uptake by the beads.

6.1.4 Reactor Sterilization and Start-up

The reactor was initially sterilized at 121°C for 39 *min* with the inoculation ports sealed with either blank plugs or with silicon

rubber septums in place. A second sterilization with all the probes and input lines in place was performed (all medium lines and other port connections were separately autoclaved and aseptically attached to the fermentor), again under the same conditions. For the sterilization, the vessel was filled with approximately 10 *L* of distilled water. Prior to the sterilization, the oxygen probe was zeroed in helium and the pH probe was calibrated in buffer solutions. After the second sterilization air flow was started to avoid the formation of a vacuum as the reactor cools. The operating temperature and pH setpoints (25°C, pH 6.8) were attained before innoculation. The initial growth medium was then pumped in and the celite immobilized spores aseptically added next. Water was then pumped into the fermentor to make a total operating volume of 14 *L*.

6.1.5 Operation and Sampling Procedure

Once the fermentor was inoculated, the reactor was left for approximately 48 hours for the spores to germinate (roughly 20 *h*) and proceed through an initial rapid growth phase. At the point where the glucose in the fermentor has been consumed, as dually determined by a decrease in cell growth and an *in-vitro* urine glucose test stick (Miles Laboratories Inc.), the computer control program was started.

Prior to removing a sample from the fermentor, air was forced back through the sampling port to clean out any old cells. Sampling was achieved by positive pressure forcing a uniform sample through the port, with sample size averaging 50 *mL*. In the case of the free cell

experiment the sample volume was recorded. The sample was then filtered as quickly as possible to avoid further substrate consumption and the filtrate and cells collected. The filtrate was placed in a freezer for later glucose analysis. A small sample was centrifuged and used for analysis by a HEWLET PACKARD HP1084B HPLC to test for penicillin-V and precursor levels. The filtered cells were completely dried in a microwave for 10 minutes, weighed, and in the case of the immobilized cells burned in a muffle furnace for 30 *min* at 400°C and weighed again so that a dry weight biomass measurement could be obtained based on the initial amount of celite in the reactor. In the closed loop control experiments the measured values for biomass and penicillin and precursor concentration were entered into the computer as the fed-back data. For a complete listing of media composition and analytical methods used refer to Appendices A and B.

6.2 Shake Flask Experiments

Shake flask experiments were performed using 500 *mL* Erlenmeyer flasks with sterile cotton plugs, and placed on a rotary shaker. 50 *mL* of media (appendix A) was placed in each flask and the flask was inoculated and observed for the duration of the growth phase. During the production phase, 5 ml aliquots of production media were added every 12 to 24 hours using sterile pipettes in a sterile transfer hood. Media for the shake flask experiments was slightly different than that used for the bench scale experiments. The main difference was that lactose was used as the energy source in lieu of glucose as it is a slowly utilized substrate and therefore compensates for the feeding

scheme used in the bench scale experiments. To determine biomass, penicillin and precursor concentrations, 1 to 2 flasks were harvested, and were assumed to be representative of all the flasks. This procedure was continued until penicillin productivity dropped.

CHAPTER VII

EXPERIMENTAL RESULTS

7.1 Open Loop Experiments

The purpose of the open loop experiments was to determine optimum operating conditions, and to estimate kinetic parameters in the model. A total of fifteen open loop experiments were performed. The single most obvious finding of these experiments was the variability that was observed between experimental runs despite apparent similarity in operating conditions. Furthermore, overall penicillin productivity in all of the experiments was quite low in comparison with industrial fermentations. This is a common problem noted in the literature (Swartz, 1985), and can be explained by an incomplete knowledge of the operating conditions under which the industrial strains are grown. As well, the strain used in this study is known to produce significantly lower titers than present production strains. However, despite the above mentioned problems, a good comparative study can be accomplished with the present experimental findings.

Experimental runs #12 to #16 were not performed by the author, however they were included simply for the calculation of the average parameter values. Runs #17 to #24 were performed using defined medium and runs #25 to #34 used a semi-defined medium containing less than 5% corn steep liquor. Table 1 shows the operational changes from run to run and Table 2 shows the calculated kinetic parameters for the open loop experiments.

RUN #	$\mu_{des}(h^{-1})$	μ_{gr} (h^{-1})	μ_{pr} (h^{-1})	x_{gr} (g/L)	$x_{pr}(g/L)$
12	0.005	0.0260	0.0080	5.566	20.120
13	0.003	0.1579	0.0214	3.345	9.889
14	0.005	0.1174	0.0158	11.327	25.555
15	0.010	0.1312	0.0082	11.838	29.573
16	0.010	0.1181	0.0099	4.804	26.265
17	0.015	0.0337	0.0010	1.280	30.508
18	*****	******	******	******	******
19	0.012	0.0166	0.0326	1.496	23.493
20	*****	******	******	******	******
21	*****	******	******	******	******
22	0.016	------	0.0108	3.151	25.113
23	0.015	0.0267	0.0162	1.430	34.150
24	0.015	0.0356	0.0207	5.625	25.908
25†	0.012	0.0643	0.0180	10.438	27.928
26†	0.012	------	0.0167	3.930	24.232
34†‡	0.015	0.0316	0.0107	3.565	13.290

N.B. *** indicates a contaminated run.

† indicates semi-defined medium

‡ indicates non-immobilized free cell run

des = desired value during production phase

gr = growth phase

pr = production phase

Table 1 Operating conditions of the open loop experiments indicating operating growth rates and final biomass.

RUN #	$Y_G(g/g)$	$\sigma(h^{-1})$	$m(h^{-1})$	$q_p^{max}(h^{-1})$	$a(h^{-1})$	$K_s(g/L)$
12	0.4134	0.0333	0.0134	0.00098 f	0.0159	0.0000
13	0.5493	0.0279	0.0000	0.00046 p	0.0735	0.0000
14	0.5656	0.0375	0.0104	0.00017 f	0.0500	0.2887
15	0.5535	0.0271	0.0121	0.00025 p	0.0308	0.2950
16	0.5250	0.0281	0.0089	0.00037 f	0.0159	0.1910
17	0.4167	0.0265	0.0154	0.00113 p	0.0037	0.0000
18	******	******	******	*******	******	******
19	0.7247	0.0399	0.0168	0.00572 p	0.0338	------
20	******	******	******	*******	******	******
21	******	******	******	*******	******	******
22	0.5600	******	******	*******	******	******
23	0.7899	0.0391	0.0184	0.00022 p	0.0233	------
24	0.6306	0.0368	0.0007	0.00108 p	0.0250	------
25†	0.6413	0.0404	0.0120	0.00072 p	0.0250	0.1623
26†	0.5991	0.0544	0.0256	0.00122 p	0.0227	0.1142
34†‡	0.3782	0.0387	0.0098	0.00251 f	0.0143	0.2331

N.B. *** indicates a contaminated or incomplete run.

† indicates use of a semi-defined medium.
‡ indicates a non-immobilized free cell run
p indicates pelleted cell morphology
f indicates free cell morphology

Table 2 Calculated kinetic parameters of the open loop experiments.

Despite the variability, average values were calculated for cell growth yield ,Y_G , maintenance energy m, and the constant "a" relating cell age to penicillin productivity. The mean value for glucose yield was determined to be 0.566 *g-biomass/g-glucose,* which is similar to values found in the literature. This value was calculated by dividing the net biomass (*g*) produced by the net amount of glucose supplied (*g*) while accounting for dilution.

The mean maintenance ration was 0.011 h^{-1} which is about 50% lower than the values reported by other authors working with submerged, free cell cultures. This value was calculated by solving for σ for the steady-state substrate balance (equation 10). Assuming a value for Y_p from literature, and using the previously calculated value for Y_G, m is simply calculated from equation (6) using calculated values for the other variables. The low value calculated for the maintenance requirement could be explained by the choice of an air lift reactor and immobilization of the cells, as the cells are exposed to less severe agitation. As a result of the less severe environmental conditions experienced by the cells, less energy should be required for cell repair. This could be an important finding as it could potentially result in a significant savings due to the decreased consumption of substrate.

The average value calculated for the constant "a" which relates penicillin productivity, q_p, to average cell age, λ, is 0.027 h^{-1}, a value nearly twice that found in the literature. However, the

calculated values show a lot of variability, presumably due to various known and unknown environmental factors affecting penicillin productivity. This same spread is evident in the irregular values for q_p^{max} which are consistently lower than reported values in the literature.

The saturation constant for the Monod equation, "k_s", was also calculated to be much smaller than expected. This is believed to be a result of the inaccuracy of the enzymatic glucose assay used (at times negative values for glucose were determined), and possibly due to sampling errors. Once a sample has been taken, if the cells are not removed immediately, the glucose present will continue to be consumed by the biomass until it is gone, dependent upon the growth rate and the amount of biomass present. Figure 13 is a plot of glucose concentration vs time at a growth rate of 0.015 h^{-1}. It can be seen that at high biomass concentrations, residual glucose is rapidly consumed. At a biomass of 30 *g/L* and a growth rate of 0.015 h^{-1}, 63.6 % of the glucose present is metabolized within 3 minutes.

In an attempt to increase penicillin productivity, a semi-defined medium containing 5% *w/v* corn steep liquor (CSL) was tested in Experiment #25. Apart from acting as an energy source, the addition of CSL also provides complex nutrients required for penicillin production. This resulted in increased growth with no significant increase in penicillin production. Subsequently, the CSL concentration was decreased to 3% *w/v*. Again, a significant increase in penicillin

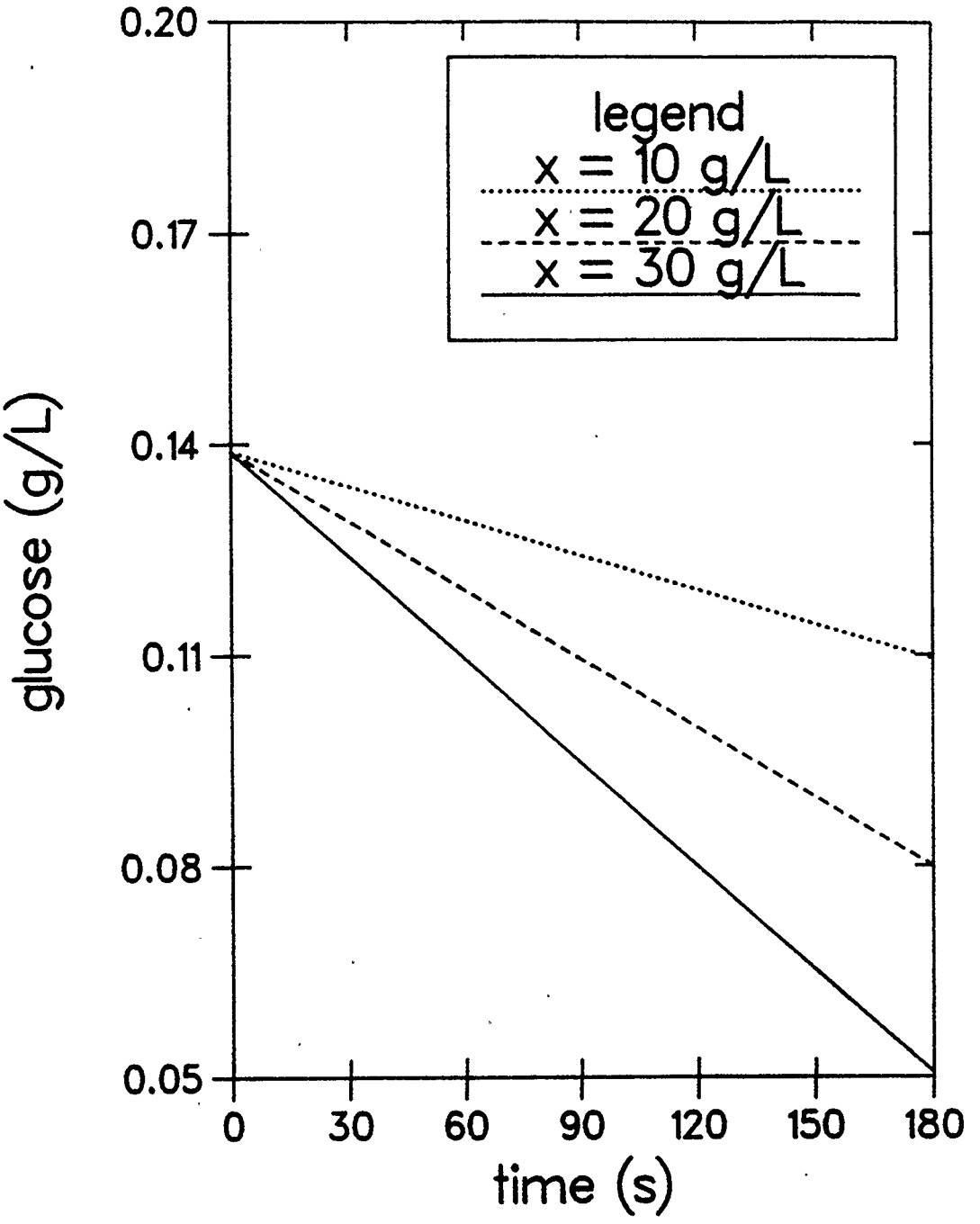

Figure 13 The effect of sampling time on measured glucose concentration at a specific growth rate of 0.015 h^{-1}.

productivity was not observed. The small amount of CSL should not affect calculations for projected growth rate based on glucose concentration and, as it is a typical ingredient in most industrial media, it was left in the media at low concentrations.

During the experiments, two separate growth morphologies were observed: a relatively clear fermentation broth where the majority of the biomass (>85%) was anchored to the celite; and a condition where a large portion of the biomass was in the form of free cells coexisting with the immobilized biomass. In the case where a large portion of the biomass was free cells, a loose fluffy-type of growth was dominant and toward the end of a run, the free cells could comprise up to 80% of the total biomass. In the situation where the fermentation broth was essentially clear, cell growth was predominantly as a smooth compact matt around the celite. This form of pelletization was fairly reproducible with the addition of CSL to the fermentation medium, and greatly facilitated oxygen transfer as well as allowing for higher biomass concentrations due to the compact nature of the beads. During some of the experiments, the compact beads would form a hollow center around the celite particle with the cells no longer anchored to the celite itself. This condition occured with large beads up to 2.5 *mm* in diameter and was probably due to autolysis caused by poor mass and oxygen transfer to the interior of the pellet. A photograph of the pellet structures showing one of the hollow beads split open is shown in Figure 14.

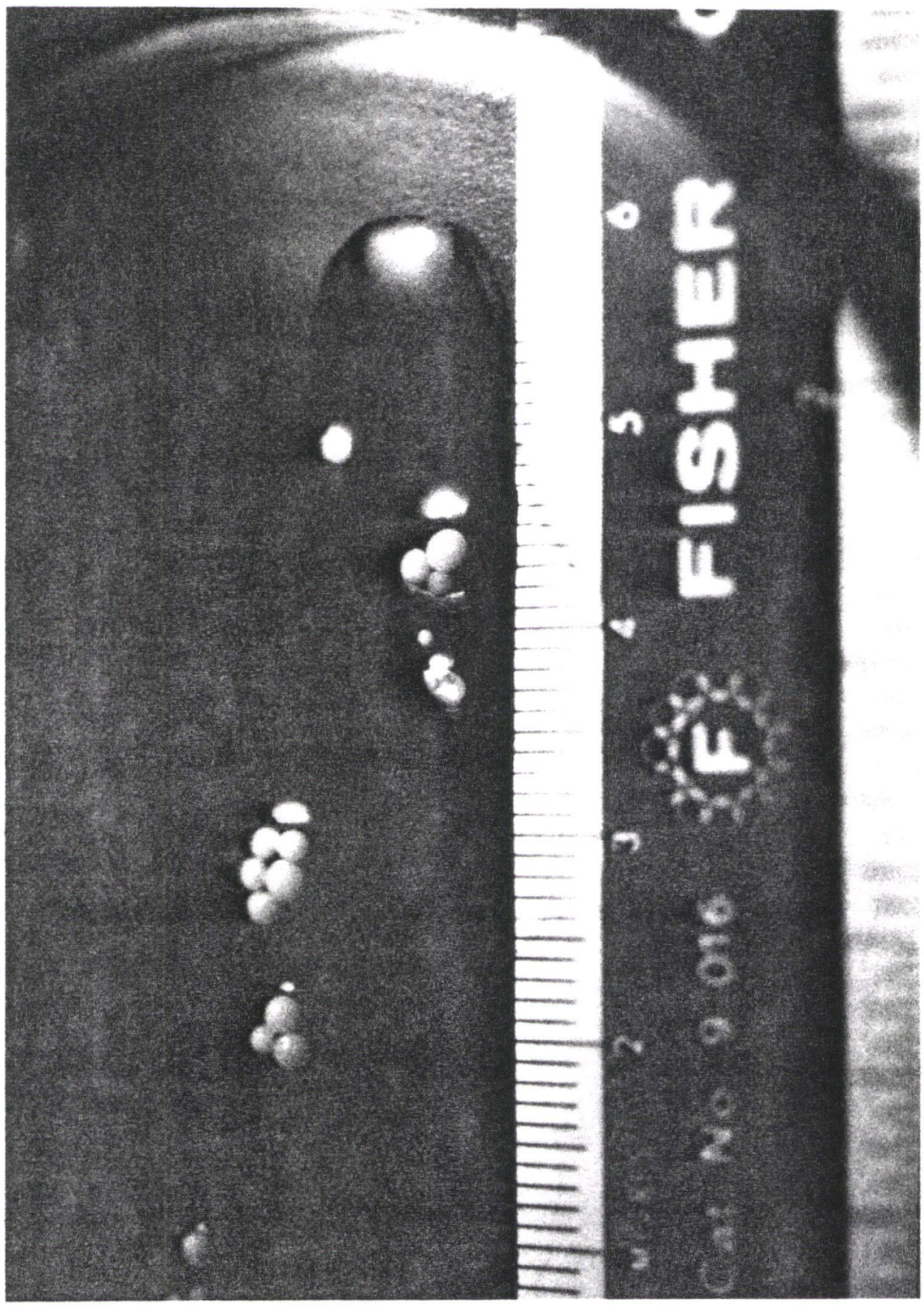

Figure 14 Typical pellet structure obtained after 80 hours of growth showing the hollow center due to autolysis.

No specific trend relating pelleted or free-cell morphology to higher or lower q_p was observed when comparing open loop experiments 12 through 26. It is well known that morphology can alter an organism's ability to produce secondary metabolites. This lack of differentiation may be due to several experimental variations which could mask this important experimental condition. As a result a truly free-cell experiment was run without celite immobilization and partial anchorage as a variable.

Experiment #34 was performed using free non-immobilized cells under the same conditions used in the other experiments. This was done simply as a control experiment to determine if the immobilization was adversely affecting penicillin production. Because only one experiment was performed, a conclusive answer cannot be given concerning observed differences. However, three interesting features were noted. First, the overall biomass from glucose yield was significantly lower than it was in previous experiments. Second, it is also interesting that due to the rheology of the free-cell broth, it was necessary to supplement the air stream with pure oxygen early in the run to ensure a high dissolved oxygen concentration in the broth. Oxygen supplementation was begun at a much lower biomass than in the immobilized runs (7 *g/L* compared to 16 *g/L*).

The third point to be noted in the free cell experiment is that the value for q_p^{max} is higher than in the immobilized experiments. As will be discussed later, this was also found to be the case in shake

flask experiments performed with free cells, though the value is still much lower than those found in the literature.

Figures 15, 16 and 17 show the experimental data from run #26 and the values predicted by the model using the calculated parameters. It can be seen that the correlation is quite good in this case. The model predictions follow both the biomass and penicillin curves very well, though the model does fail to predict the large plateau found in the phenoxyacetic acid curve. The shape of the curve, however, is approximated by the model. Further comments regarding matching experimental values to the proposed model can be found in Chapter VIII.

Figures 18, 19 and 20 show the results from the free-cell experiment performed for a comparison of penicillin productivity and overall growth achieved under similar environmental conditions. Once again, the plots show the values expected by simulation under the experimental conditions using the unstructured model, and the correlation is very good. The predicted values for precursor concentration are once again poor. The predicted values for precursor concentration employ a different equation than was originally proposed and this is discussed in Chapter VIII.

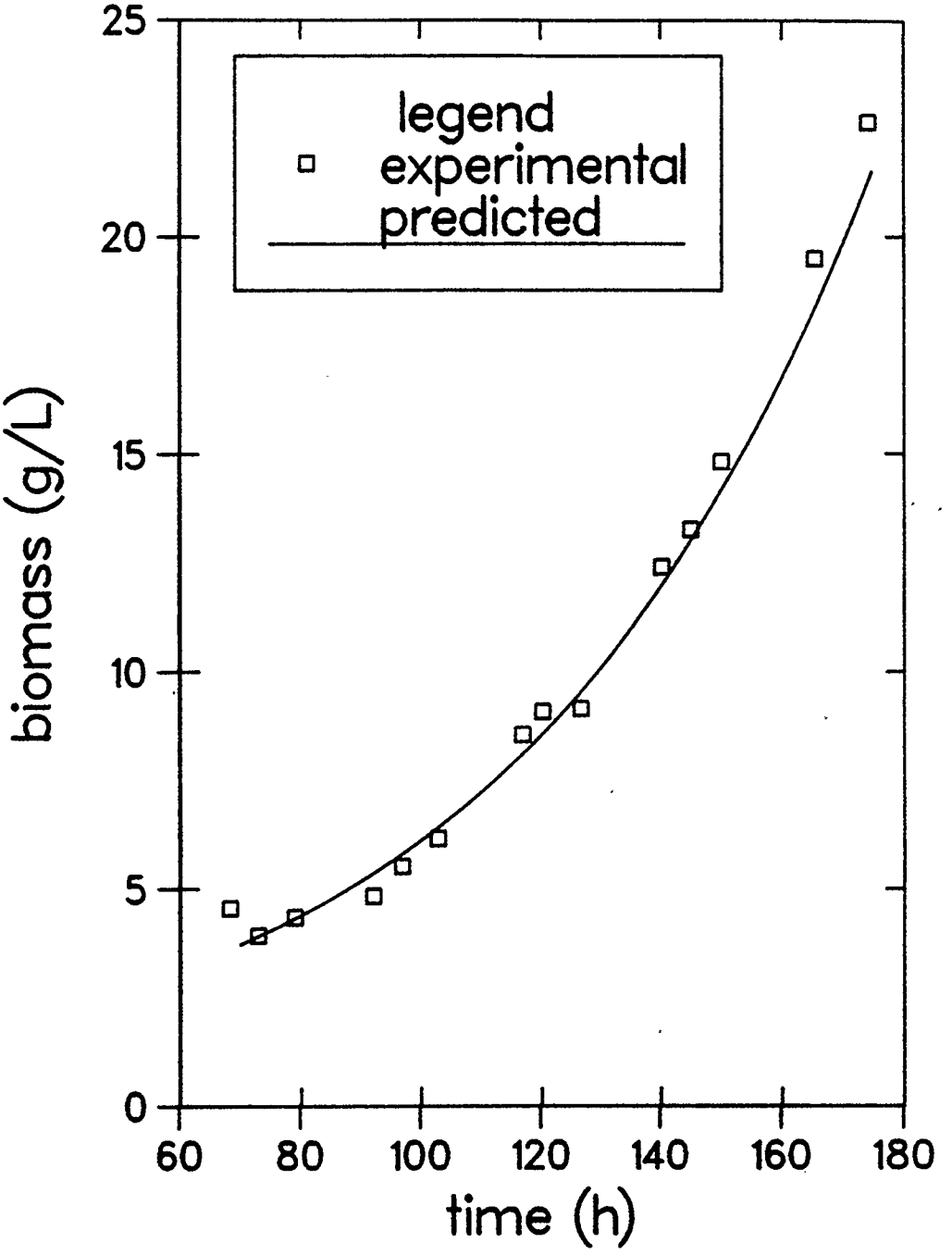

Figure 15 Biomass values from Experiment #26 and the predicted biomass values from the unstructured model.

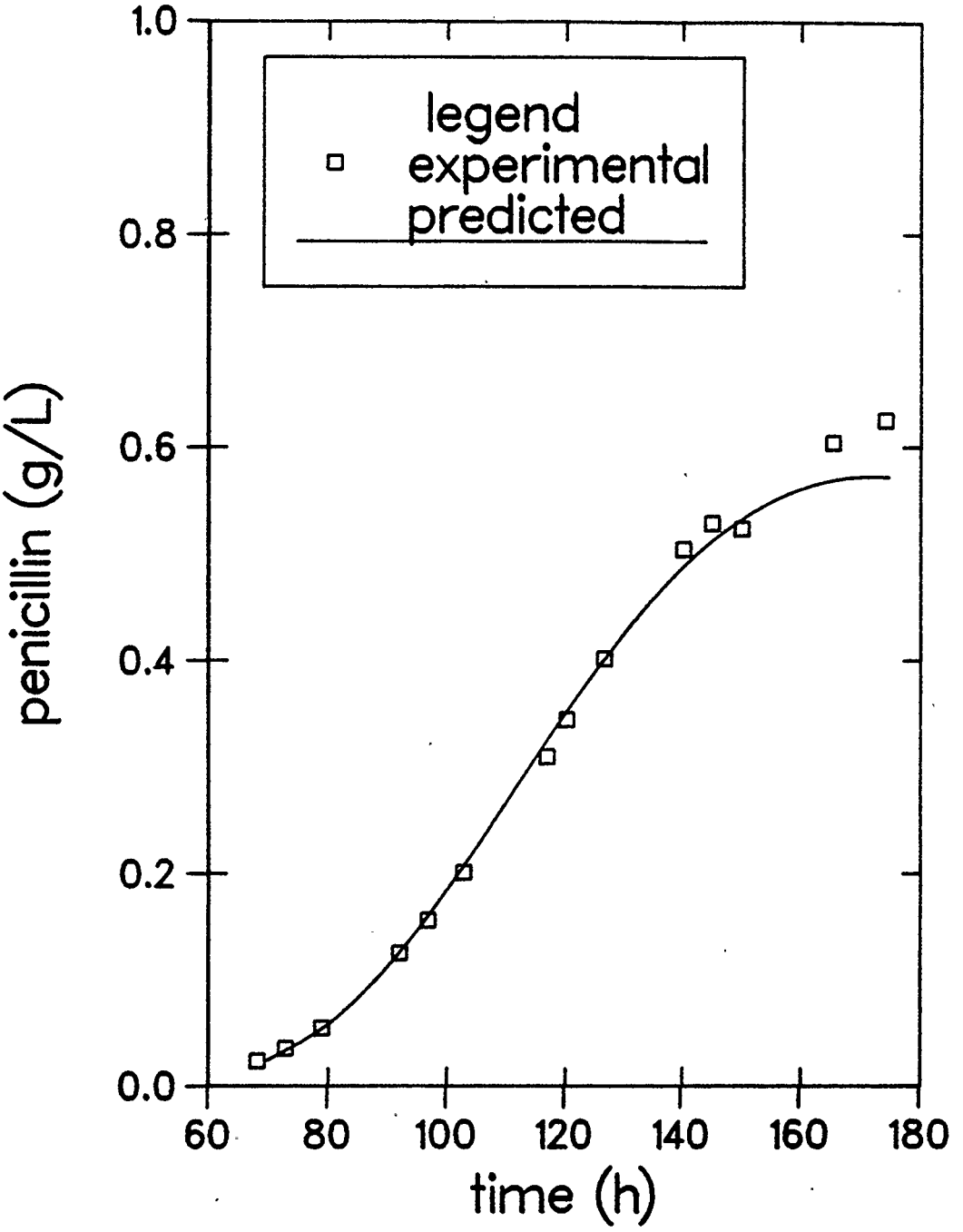

Figure 16 Penicillin concentration values from Experiment #26 and the predicted values from the unstructured model.

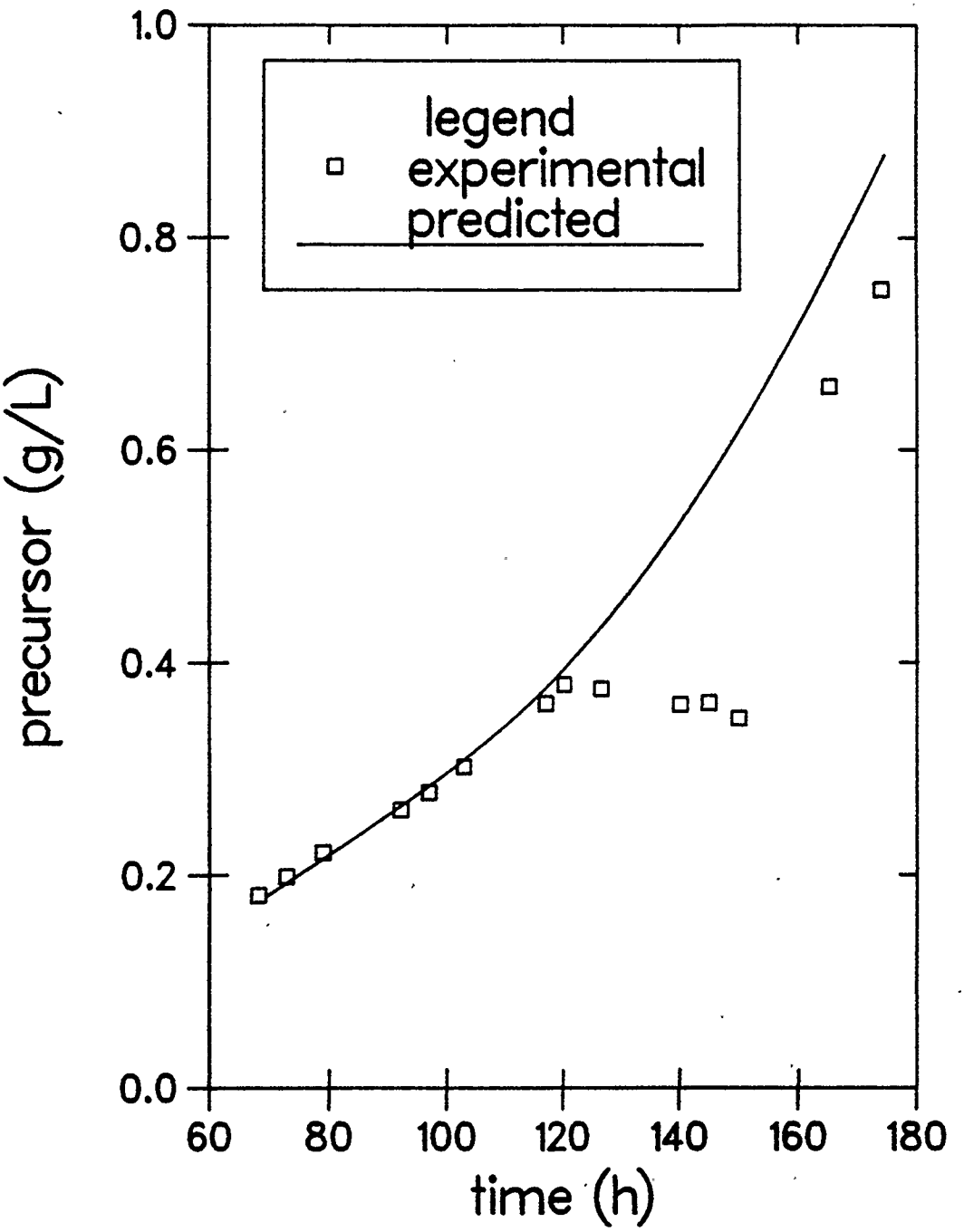

Figure 17 Precursor concentration values from Experiment #26 and the predicted values from the unstructured model.

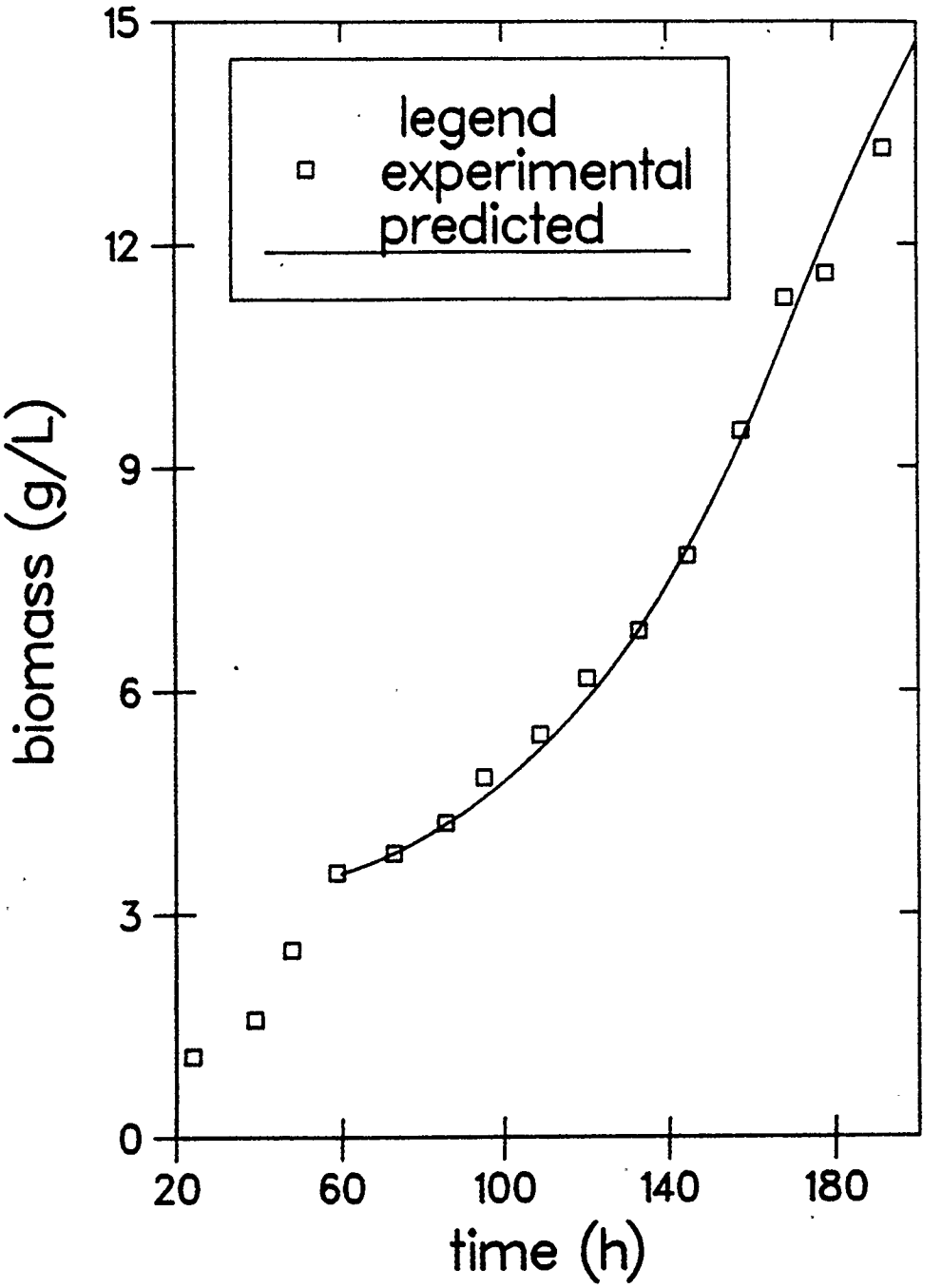

Figure 18 Experimental and predicted values of biomass for Experiment #34 performed with free non-immobilized mycelia.

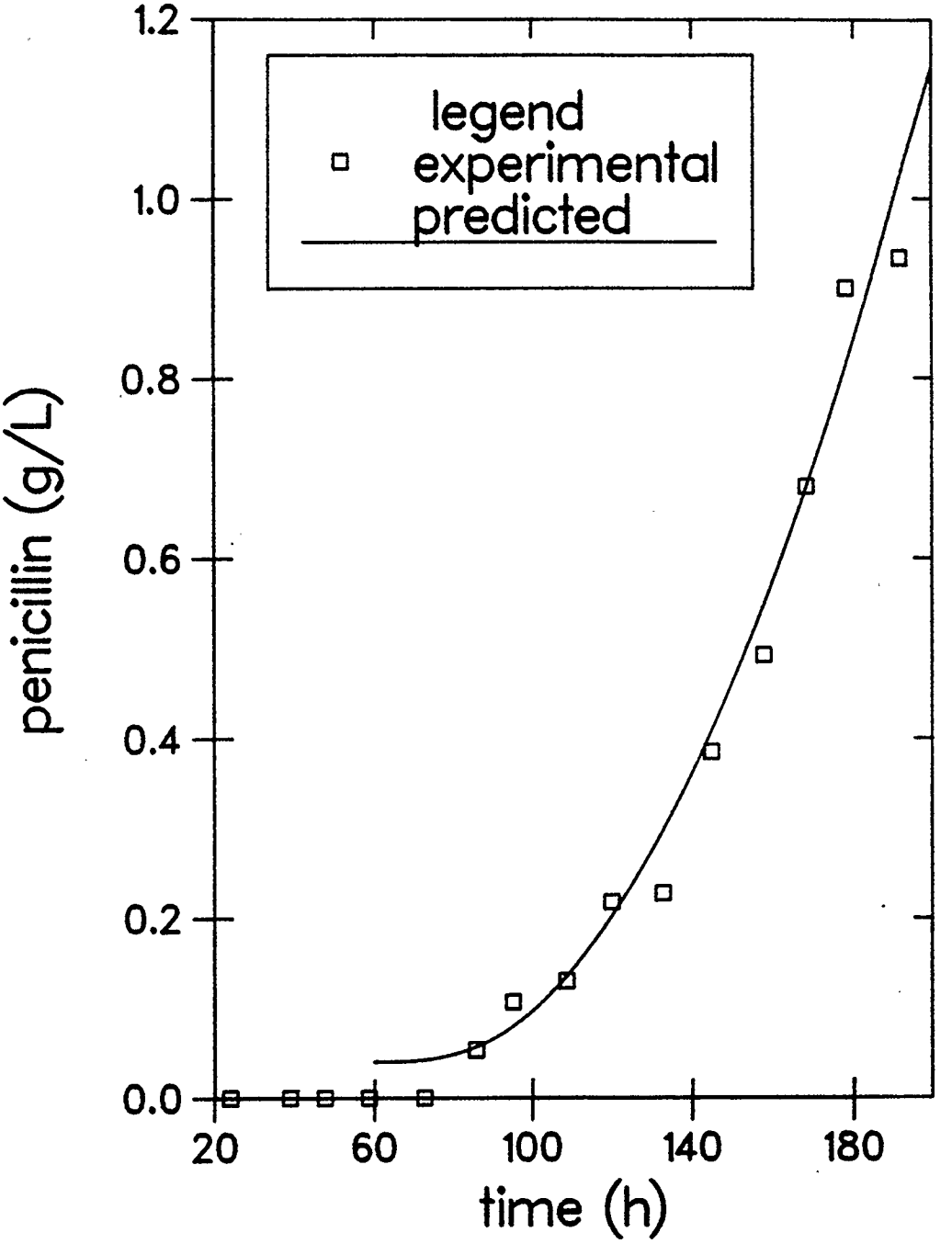

Figure 19 Experimental and predicted values for penicillin concentration for Expeiment #34.

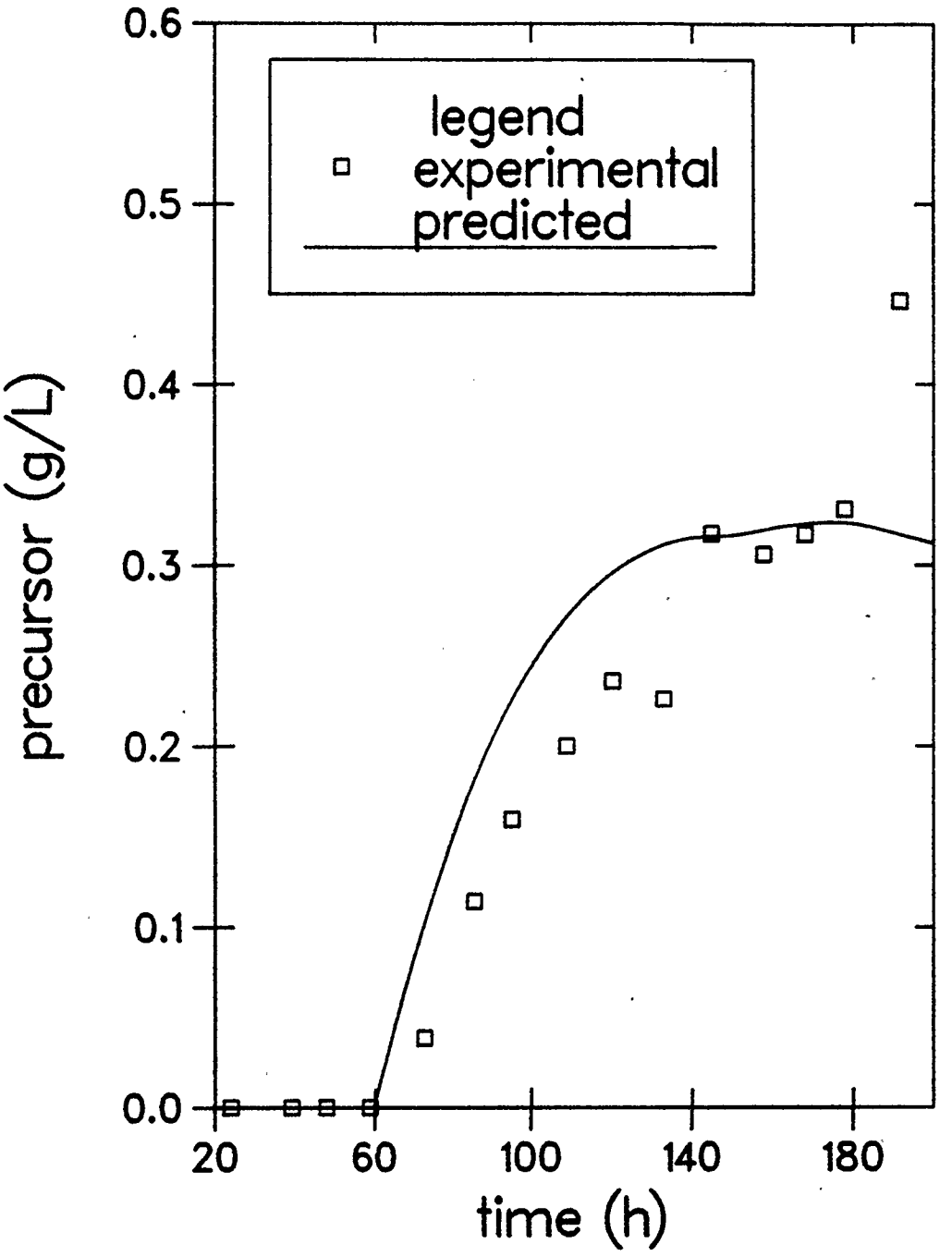

Figure 20 Experimental and predicted results for Experiment #34 for precursor concentration.

7.2 Shake Flask Experiments

In an effort to determine the reason for the low penicillin productivity, a series of shake flask experiments were performed. In these experiments strain-type, media, inoculum size and precursor type were varied. The results from these experiments are shown in Table 3. On the average, the shake flask cultures appear to be more productive than the bench scale runs although the physical differences between the two systems are great enough to negate a direct comparison. However, comparisons can be made concerning variables such as strain, initial inoculum size and media formulation. The shake flask experiments reinforce the finding that CSL does not improve penicillin production, and indicate that the size of the initial inoculum has no effect on productivity for the range tested. Furthermore, the experiments also show similar productivities between the P2 and E15 strains tested, indicating that the low productivities encountered were not a result of the chosen strain.

While the shake flask experiments illustrated that slightly higher penicillin productivities could be achieved, these values were still significantly lower than titers expected for the E15 strain. The reasons for the improved productivities could not be explained by the variables tested for in the experiments, and cannot be explained by other variables due to differences in experimental set-up. It can be concluded, however, that the current strain and medium formulation are not the reason for the improved titers observed in the shake flask experiments though this may be a contributing reason that values are

RUN #	$x^{max}(g/L)$	$q_p^{max}(h^{-1})$	comments
1	11.02	0.0021	E-15 in a defined media producing pen-V in E-15 media up to 1.26 *g/L*
2	7.98	0.0011	new E-15 strain in defined medium produced 0.945 *g/L* pen-V
3	5.58	0.0019	P-2 strain in defined E-15 medium only producing 0.312 *g/L*, poor growth with some sporulation
4	14.46	0.0012	excellent growth of E-15 in semi-defined medium but only 0.617 *g/L* of pen-V as final titer
5	15.77	0.0023	E-15 in defined medium with a high initial inoculum; 0.9532 *g/L* final
6	13.94	0.0024	E-15 in defined medium with a low initial inoculum; 1.276 *g/L* final
7	18.46	0.0014	E-15 in semi-defined medium with a final titer of 1.140 g/L pen-V
8	6.72	0.0021	P2 in defined medium shows low growth and final titer 0.728 *g/L*
9	18.38	0.0015	P2 in semi-defined medium shows good growth and 1.262 *g/L* final
10	20.40	0.0021	E15 in semi-defined medium with a final titer of 1.669 *g/L* pen-V

Table 3 Summary of shake flask experiments comparing media formulations and strain type in regard to penicillin production.

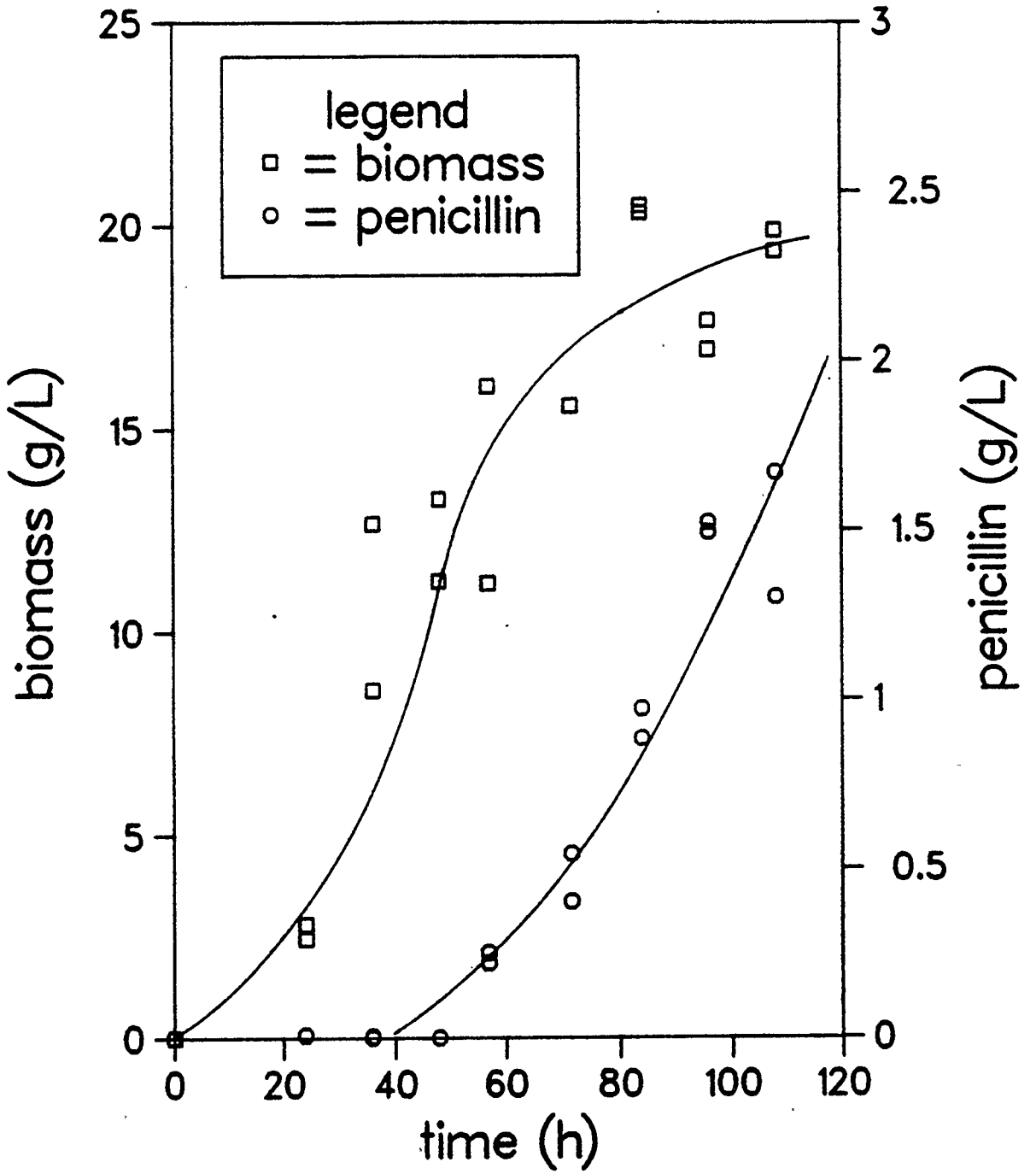

Figure 21 Penicillin and biomass concentrations for shake flask Experiment #3.

lower than those reported else-where. Fig. 21 shows the results of shake flask Experiment #3 where the higher penicillin productivities observed in the shake flask experiments are evident.

7.3 Closed Loop Experiments

A total of seven experiments were conducted in the closed loop mode to evaluate the proposed control scheme. A brief summary of the experimental set points and the operating conditions can be found in Table 4. Table 5 shows the controller constants used for each of the closed-loop experiments. A more detailed description of the experiments, and discussion of the results follows.

7.3.1 Experiment #27

Run #27 was the first experiment under closed loop conditions and it served to acquaint the author with the new set-up used and aided in resolving initial technical difficulties. An error in the computer program negated any useful results, and this was the only run where nutrients were included in media B. The CSL present in media B tended to unaccountably increase the growth rate and so all subsequent runs were performed with media B simply containing precursor and water.

7.3.2 Experiment #28

With many of the technical difficulties solved, a successfully controlled run was performed. Initial control of the growth rate was poor as the controller placed a lot of importance on the initial biomass samples. The first entry was lower than the initial

RUN #	$\mu_{sp}(h^{-1})$	$\mu_{exp}(h^{-1})$	$P_{sp}(g/L)$	$P_{max}(g/L)$	$z_{sp}(g/L)$	$z_{max}(g/L)$
27	0.014	0.0372	0.60	0.058	0.40	0.251
28	0.013	0.0174	0.60	0.388	0.40	0.431
29	0.013	0.0031	1.00	0.268	0.40	0.540
30	0.100	0.0772	2.00	0.000	0.40	0.167
31	0.015	0.0070	5.00	0.152	0.40	0.402
32	0.015	0.0163	5.00	0.255	0.40	0.477
33	0.015	0.0148	5.00	0.510	0.40	0.531

Table 4 Set points and experimental values obtained in the closed loop experiments.

RUN #	K_{cs}	K_{cp}	K_{cz}	T_{ip}	T_{is}	λ_c	P_{**}
27	13	-0.44	0.019	2.8	8.0	0.95	100
28	13	-0.44	0.019	2.8	8.0	0.95	100
29	13	-0.44	0.019	2.8	8.0	0.90	10
30	13	-0.44	0.019	2.8	8.0	0.90	10
31	13	-0.44	0.019	2.8	8.0	0.75	1.0
32*	8	-0.44	0.019	2.8	8.0	0.75	1.0
33*	10	-0.44	0.019	2.8	8.0	0.75	1.0

* In these runs the value of $\hat{\sigma}_c$ was constrained in value

Table 5 Table of the controller constants for the closed loop experiments.

value which forced the glucose feed rate slightly higher as expected. The subsequent biomass samples, however, were significantly higher which forced the glucose feed rate to zero where it stayed for 17 hours while the estimator caught up to the actual value. This under-feeding of substrate could adversely affect penicillin production. Once the estimator had stabilized, the control of the growth rate was very good with a final measured value of $\mu = 0.0151\ h^{-1}$ and an overall value of $0.0174\ h^{-1}$. The offset in operating condition is as predicted in simulations, and is due to a lack of integral control.

For the penicillin controller, the penicillin concentration never achieved the set point, so the dilution rate was kept at a minimum. It should be noted that the dilution rate began increasing before the glucose feed concentration reached its maximum value. This was due to an error in assigning constraining conditions on the dilution rate based on substrate concentration. As was expected, the precursor controller showed sluggish response due to the interaction with dilution rate, but the set point was achieved, and control about the set point was adequate for the purpose intended. Fig. 22 shows the controlled variables x, p and z for Experiment #28. Fig. 23 is a plot of ln(biomass) vs time and shows how the growth rate controller achieves linearity about the operating point. Fig. 24 is a plot of the controlled variable μ with time, and Fig.25 shows the response of the manipulated variables.

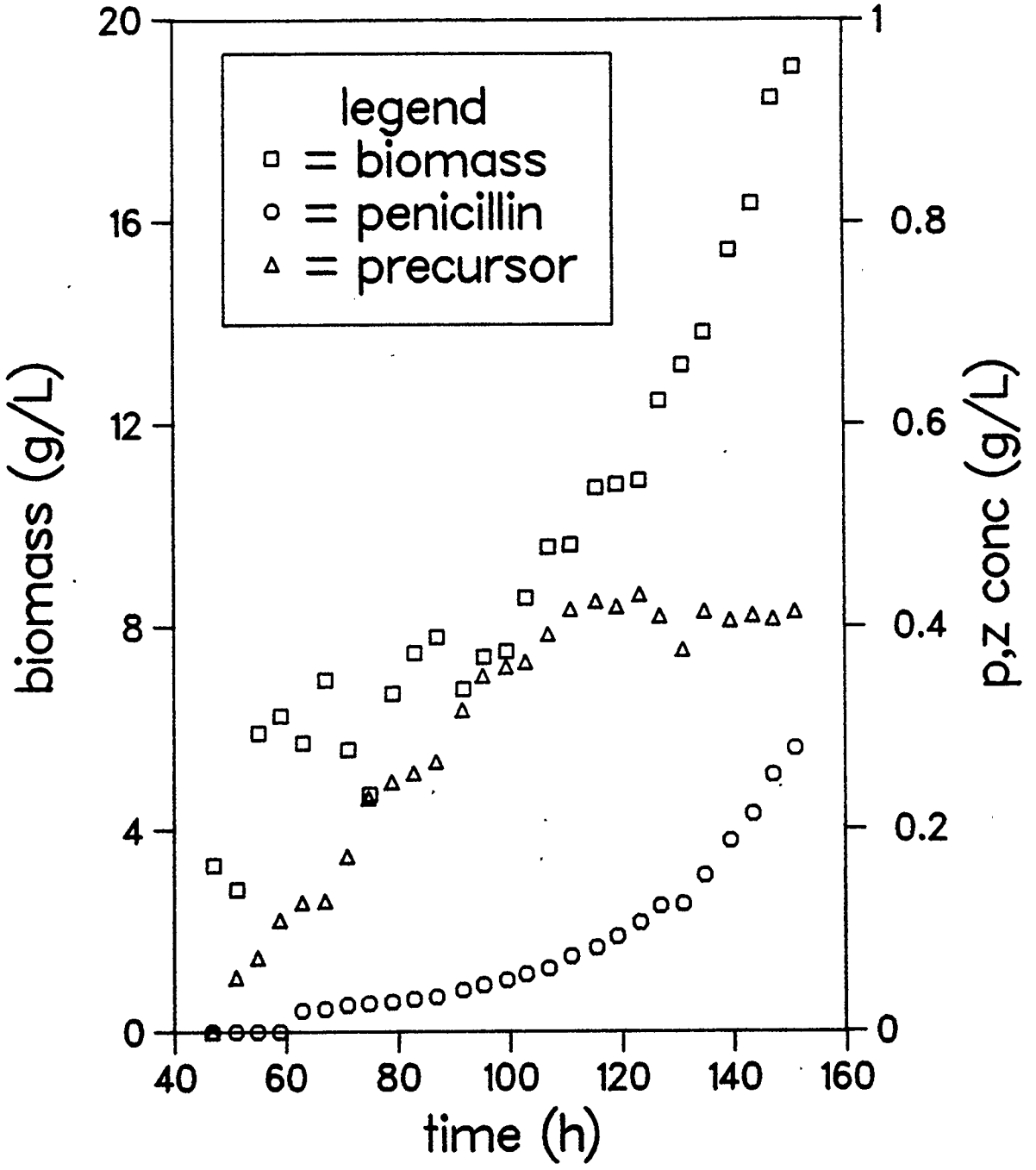

Figure 22 Biomass, penicillin and precursor transients during Experiment #28.

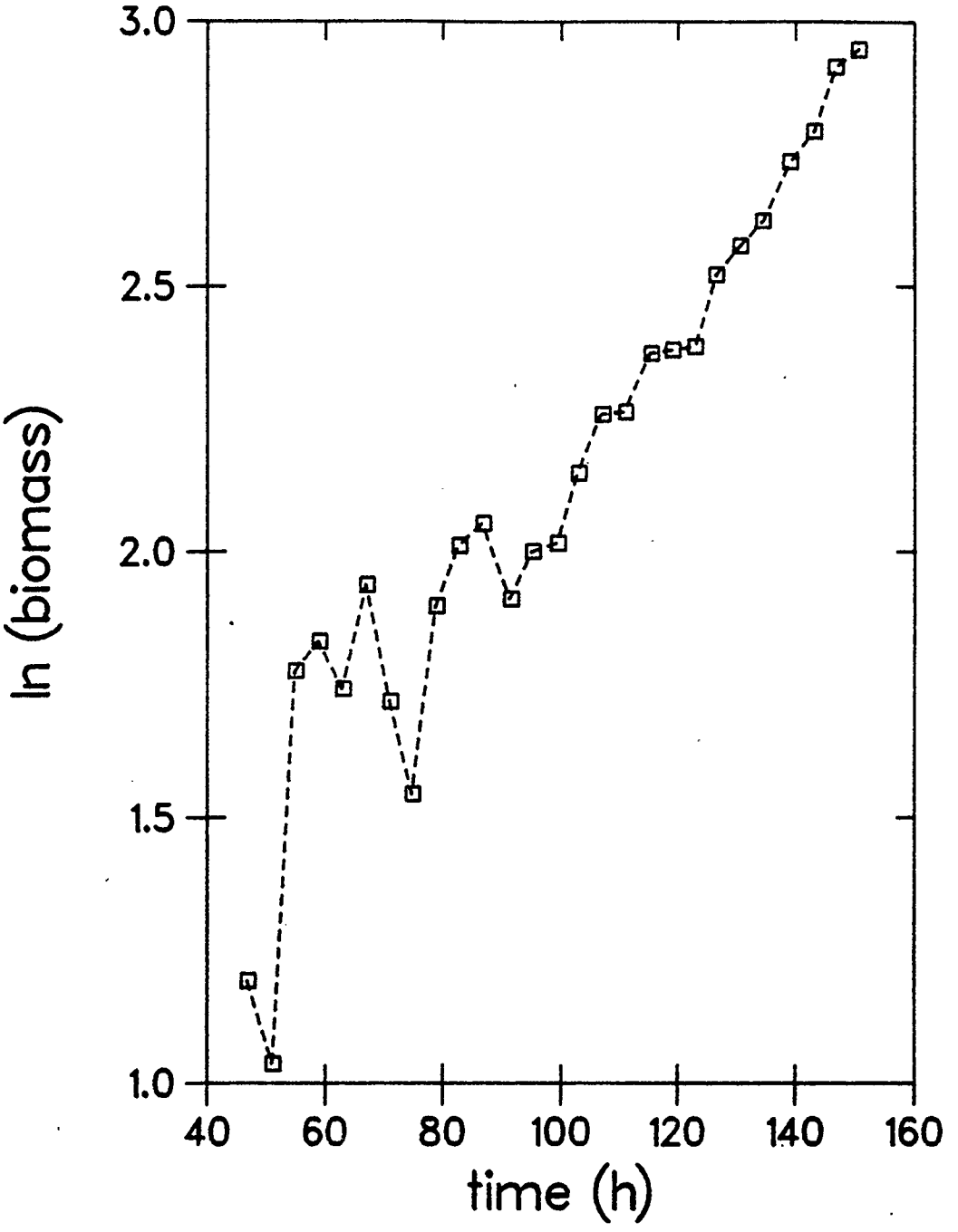

Figure 23 Plot of ln (biomass) vs time for Experiment #28 illustrating the constant growth rate achieved.

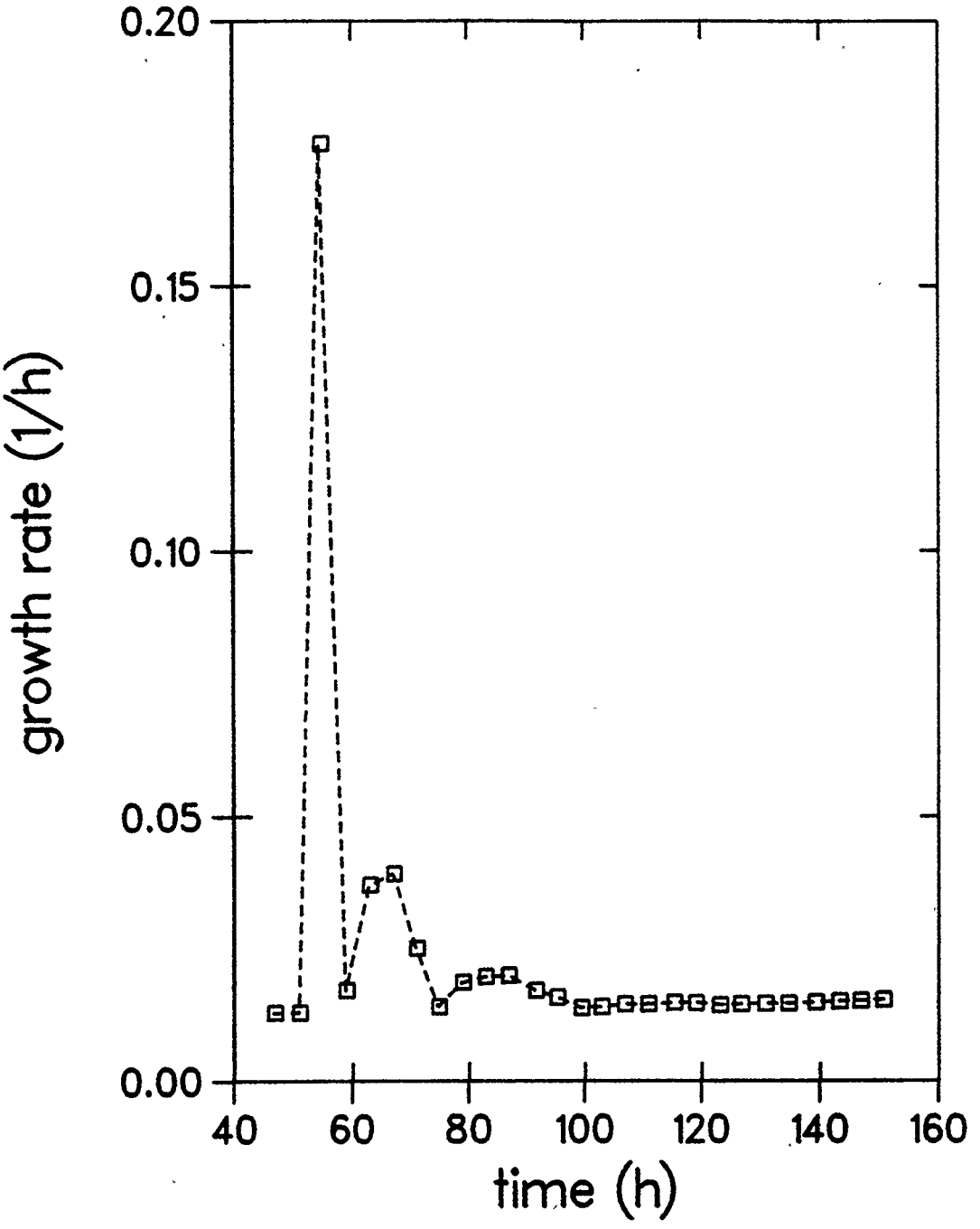

Figure 24 Plot of estimated growth rate measured through the Kalman filter for Experiment #28.

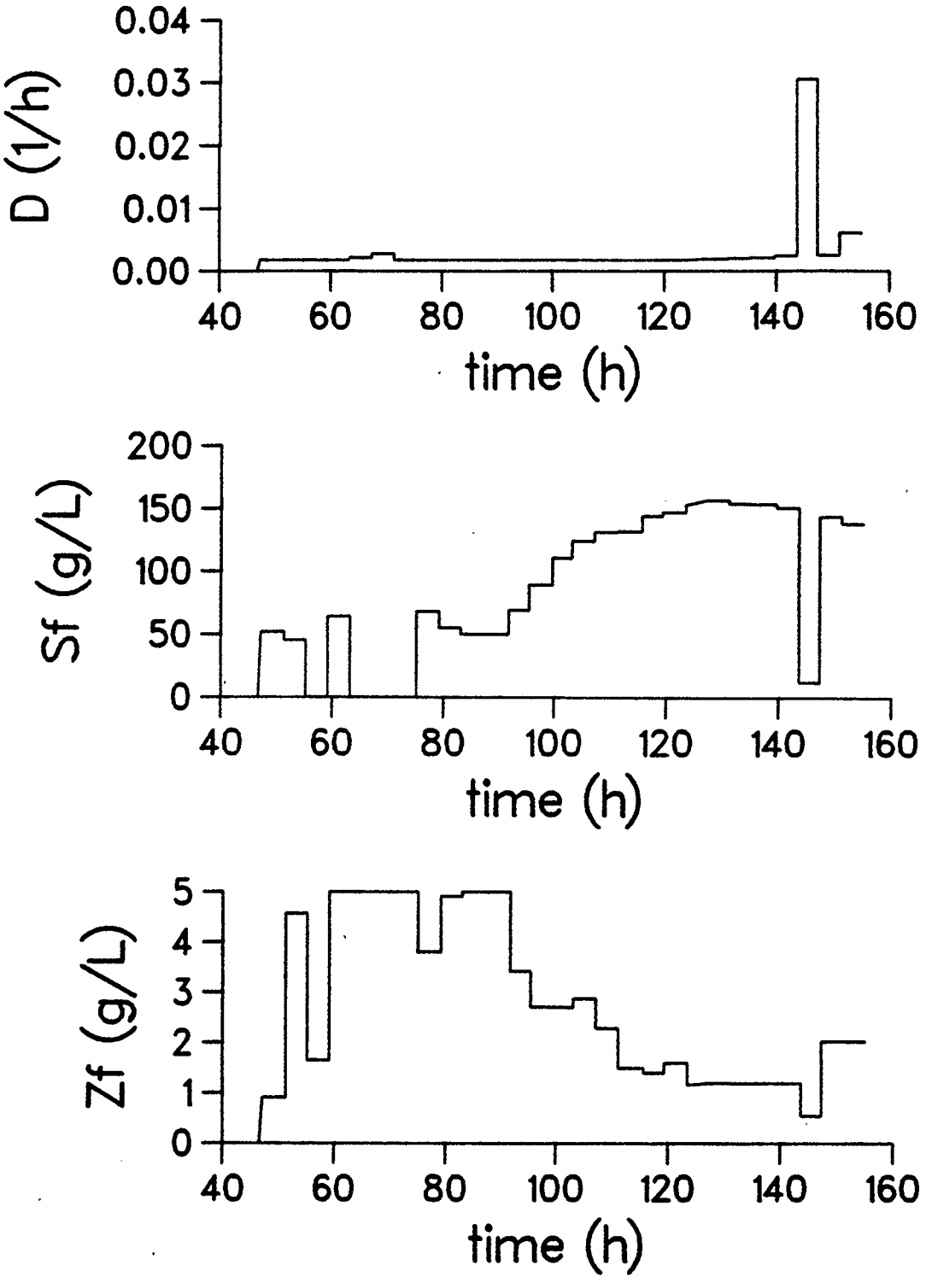

Figure 25 Manipulated variable behavior during Experiment #28.

7.3.3 Experiment #29

Run #29 was plagued with the problem of reliability and reproducibility of the biomass measurements. This was due to poor initial immobilization of the cells onto the celite particles. As a result, control of all the variables was very poor leading to transient over and under-feeding of substrate. Variations in substrate feed had a noticeable effect on penicillin productivity. Penicillin concentration remained at a relatively low concentration, increasing slowly until biomass began to increase at a controlled rate, after which penicillin also began to increase at a greater rate.

Changes introduced for this run to correct problems noticed in Experiment #28 were two-fold. First, the initial values in the covariance matrix of the Kalman filter were reduced so that less importance was placed on the initial biomass readings. Second, the forgetting factor, λ_c, in the filter was decreased to 0.75 from 0.95 so that more importance was placed on new input, and long periods of inactivity as noticed in Experiment #28 were reduced.

7.3.4 Experiment #30

Run #30 was a deviation from the other runs in that it was an attempt to duplicate the growth history of industrial fermentations (*i.e.* rapid initial growth to a high biomass, after which the fermentation is extended, usually until oxygen diffusion becomes limiting). It also was an opportunity to test the growth rate controller at much higher growth rates and evaluate its performance in

transition between a much higher growth phase and the low growth rate associated with the production phase. The reason for this change in strategy was to determine if growth history had any effect on penicillin productivity.

The most noticeable effect of this run was that no penicillin was produced. This was not necessarily a result of the altered growth history, but as a result it was decided to continue with the established protocol for subsequent experiments. The growth rate was measured at 0.1029 h^{-1} but this began to drop as the dilution rate increased. The inability to maintain this high growth rate can be explained by the set point being too near the maximum reported value, and the ability of the reactor to maintain a sufficiently high glucose concentration with the increased dilution rates. The transition between the growth and production set-points was very poor resulting in a shutting down of the glucose feed and subsequent loss of biomass. As the Kalman filter still maintained information from the growth period, the measured growth rate was still much higher than the set-point. If the proposed algorithm were to be used to control both the rapid growth and production phases, some reset of the covariance matrix would be required for a smooth transition phase. Due to the dynamic nature of the fermentation, kinetic parameters differ between the growth and production phases and would also need to be reset. All subsequent experiments were performed using the control scheme strictly in the production phase with the established protocol to assure some reproducibility and to properly establish the effectiveness of the

control scheme.

7.3.5 Experiment #31

Run #31 demonstrated another flaw in controller design early in the run. Transient over-feeding of substrate resulted from the input of a biomass value which indicated a negative growth rate even though this value was within sampling error. Not only did glucose feed concentration go to the maximum, the dilution rate was also increased to meet the apparent glucose demand resulting in a large addition of glucose being erroneously added to the fermentor. It took the system over twenty hours to recover from this over-feeding, and penicillin productivity was likewise adversely affected. Subsequent control was poor due to a large scatter in the data. Control of precursor concentration was quite good, though sluggish response was still observed. Again, penicillin concentration never reached the set-point and so the dilution rate was kept at the minimum value calculated. Fig. 26 shows the response of the controlled variables. Fig 27 clearly shows where the shift in growth rates occurs due to the over-reaction of the growth rate controller and Fig. 28 shows how the controlled variable, μ, slowly comes down to the set-point after the large initial disturbance. Fig. 29 the response of the manipulated variables for this run and shows the effect of the over-feeding.

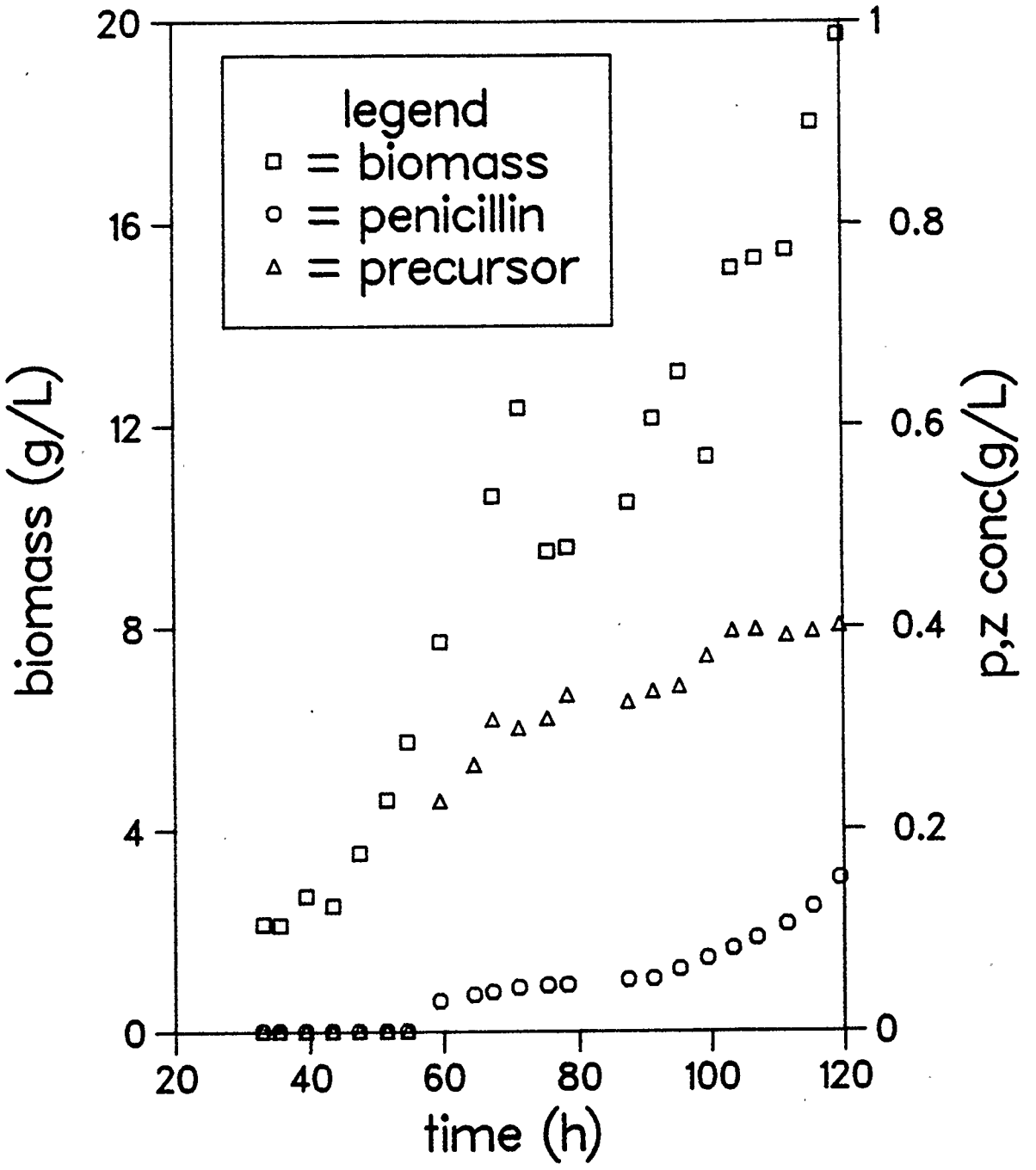

Figure 26 Biomass, penicillin and precursor transients during Experiment #31.

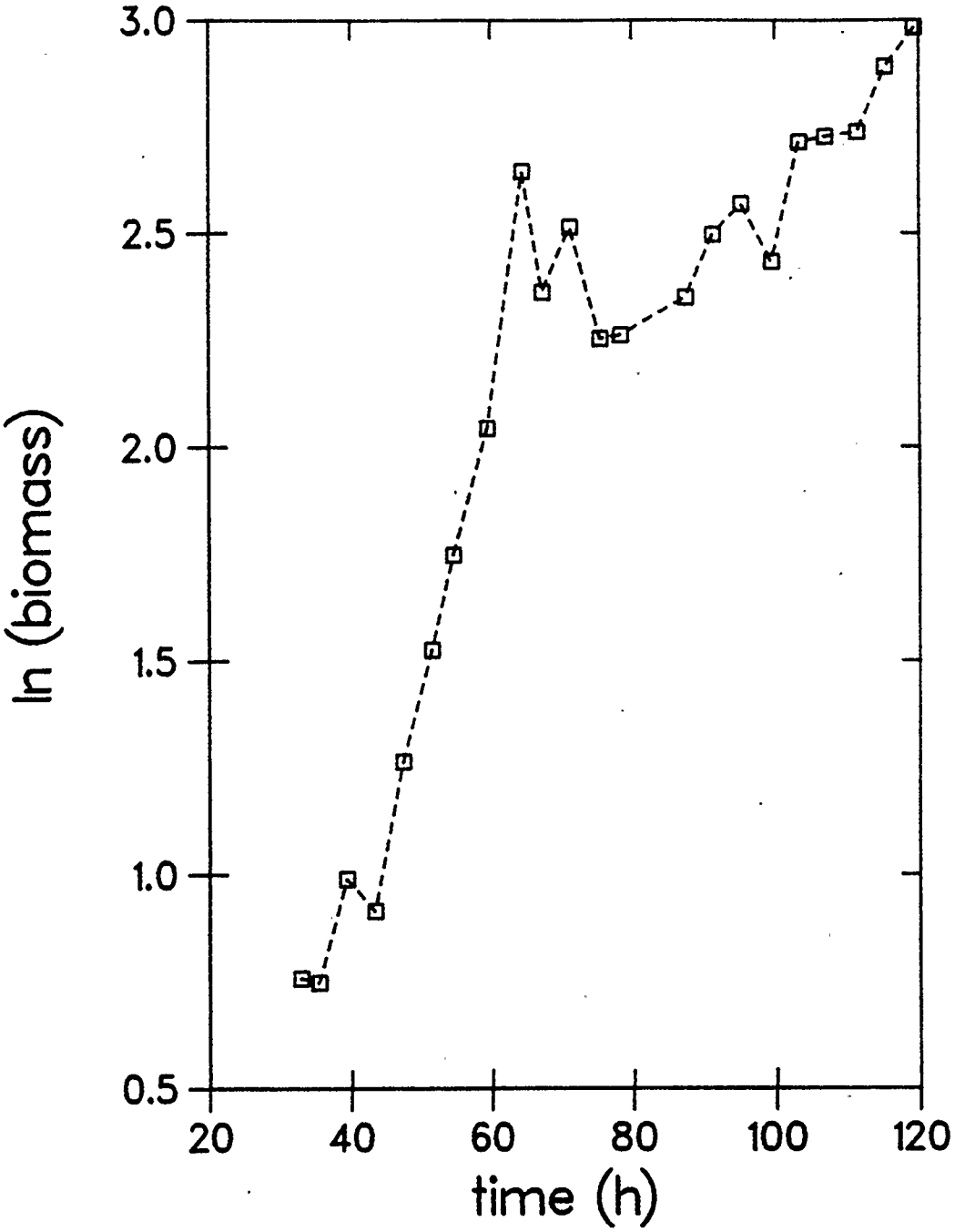

Figure 27 Plot of ln (biomass) vs time for Experiment #31 illustrating the sudden change in growth rates.

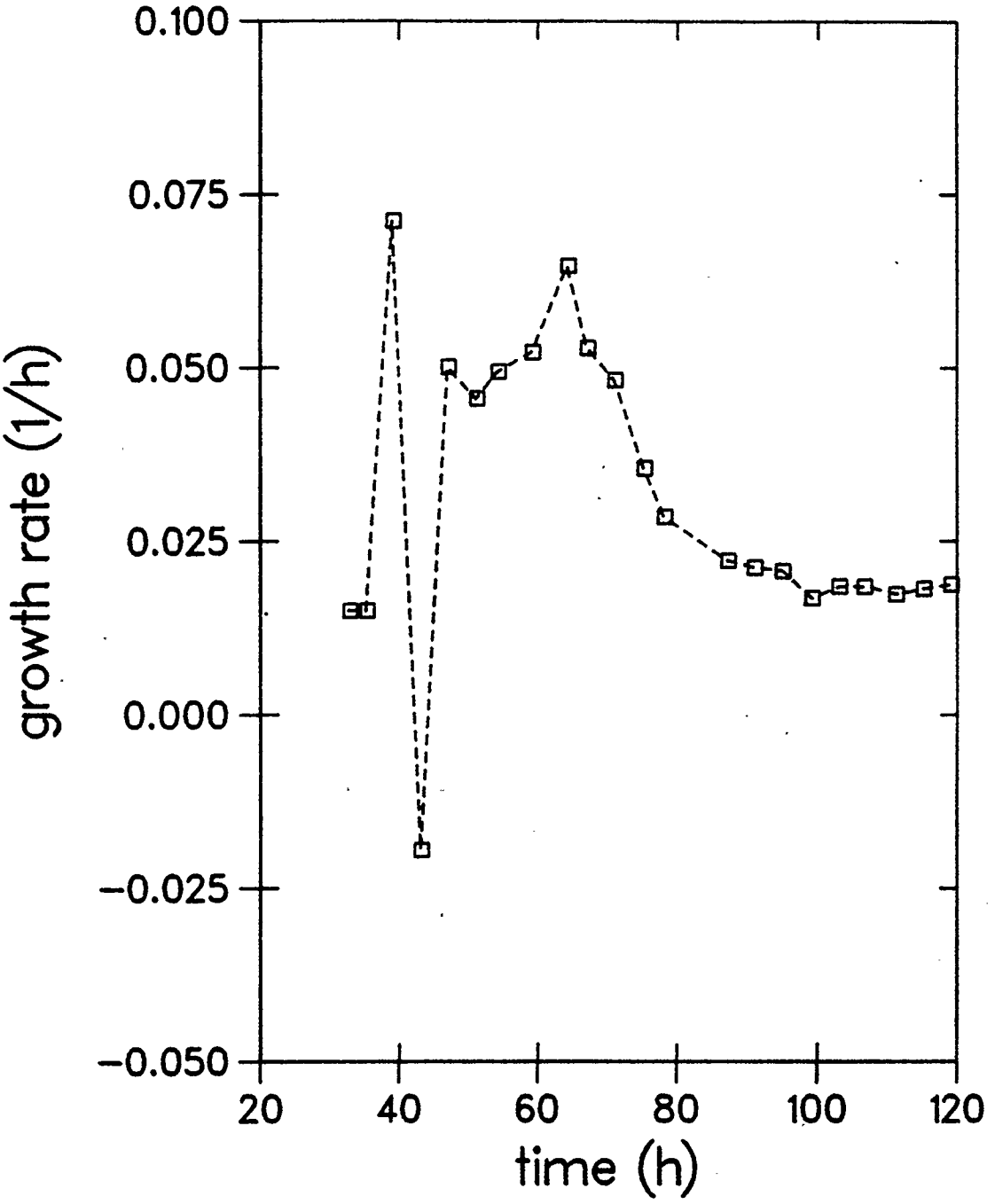

Figure 28 Plot of estimated growth rate measured through the Kalman filter for Experiment #31

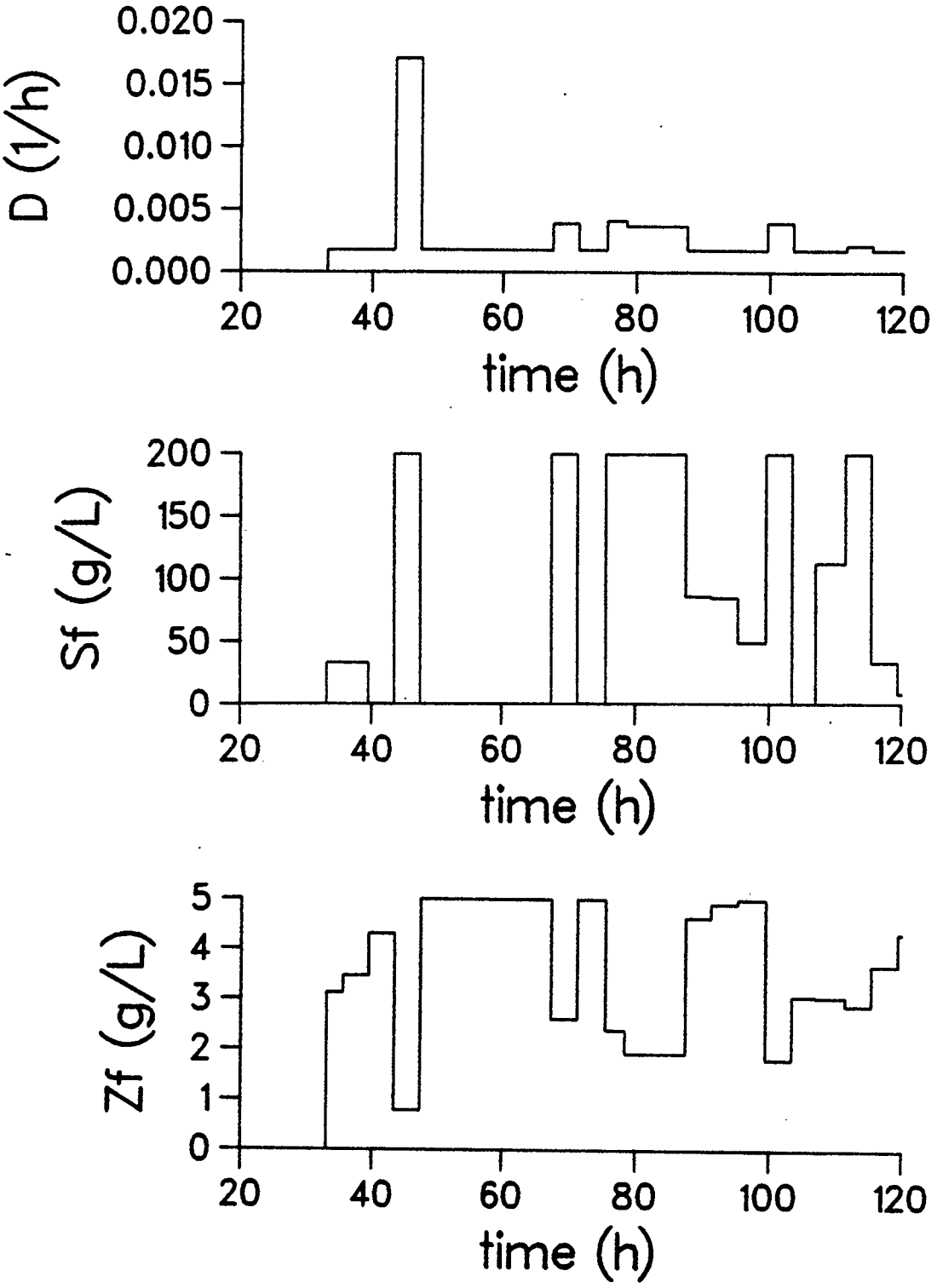

Figure 29 Manipulated variable behavior during Experiment #31.

7.3.6 Experiment #32

In an attempt to reduce the sensitivity of the control algorithm to sampling errors early in the run, the controller gain (K_{cs}) was reduced from 13 to 8 and the controlled parameter ($\hat{\sigma}_c$) was constrained. By doing this, transient over and under-feeding should be minimized with a small sacrifice in control action. This should not hamper control as parameters are already well established from the open loop experiments. The result of these changes are evident in the improved control of growth rate despite a large amount of scatter in biomass measurements. Furthermore, penicillin production, though still low, is improved over previous controlled runs. Precursor concentration overshot the set point by over 0.1 g/L before coming back down to the set point illustrating the sluggish response expected, but this higher concentration is of no concern as it is still well below the level at which precursor concentration becomes inhibitory. Figures 30 and 33 show the controlled and manipulated variables respectively for Experiment #32. Figure #31 is a plot of ln(biomass) vs time, and clearly illustrates the transition between the growth and production phases.Figure #31 also shows the relatively linear growth rate during production, despite the obvious scatter in the data. Fig. 32 shows the controlled variable, μ, with time and indicates the difficulty the estimator had following the growth rate with this scatter.

7.3.7 Experiment #33

In this run, the growth controller gain, K_{cs}, was changed to 10 and once again, $\hat{\sigma}_c$ was constrained. Again, despite some spread in

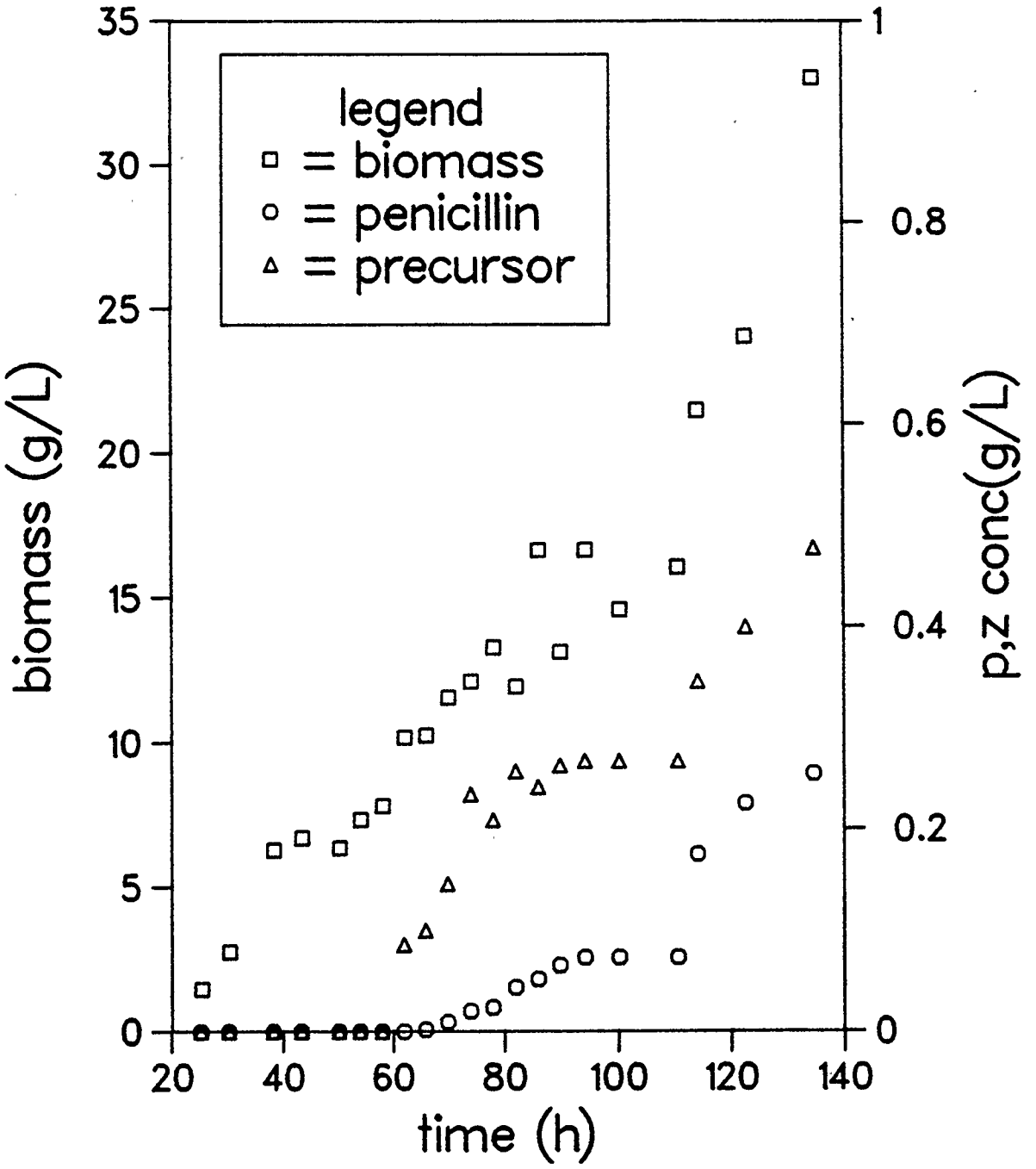

Figure 30 Biomass, penicillin and precursor transients during Experiment #32.

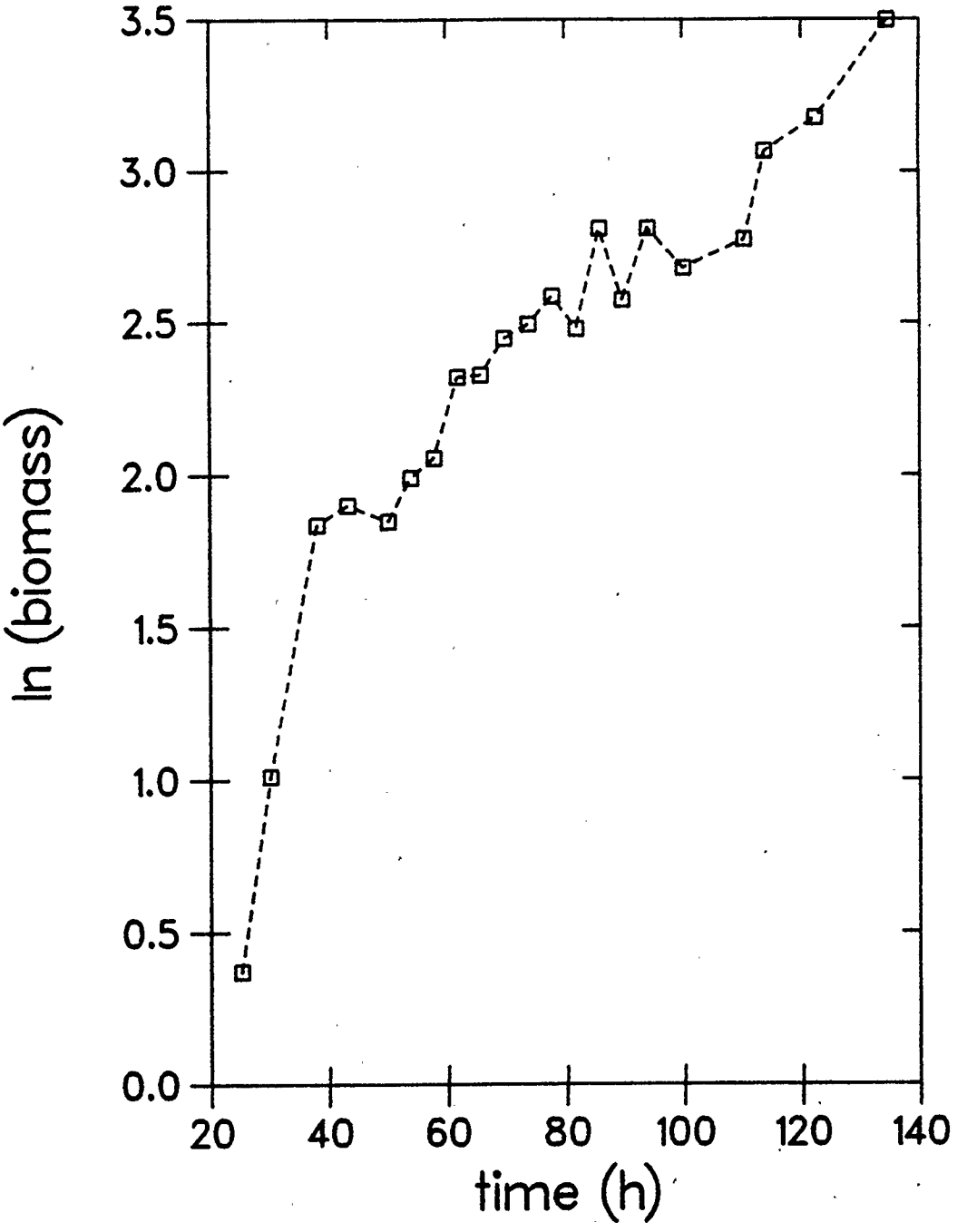

Figure 31 Plot of ln(biomass) vs time for Experiment #32 illustrating growth rate transients.

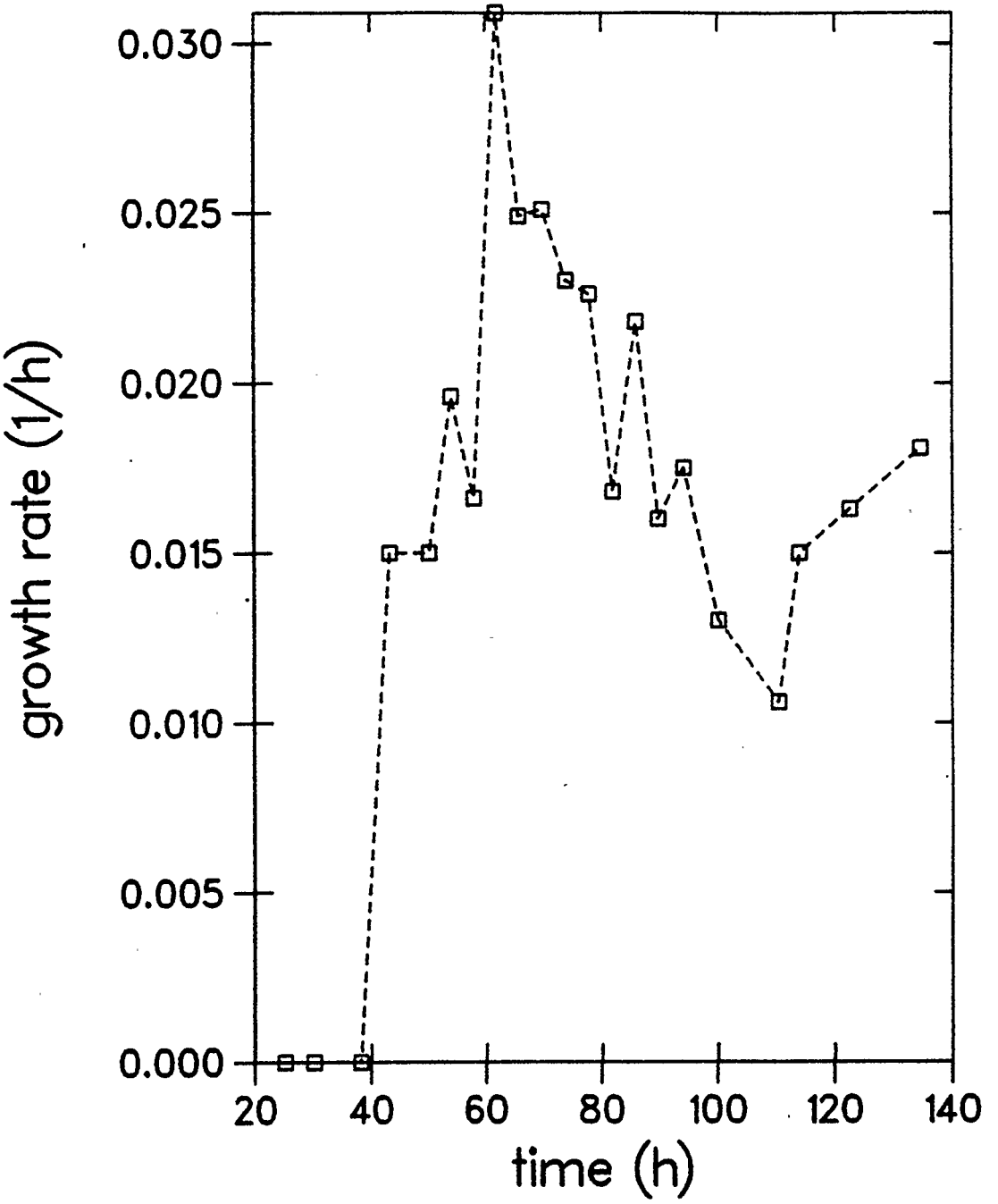

Figure 32 Plot of estimated growth rate measured through the Kalman filter for Experiment #32.

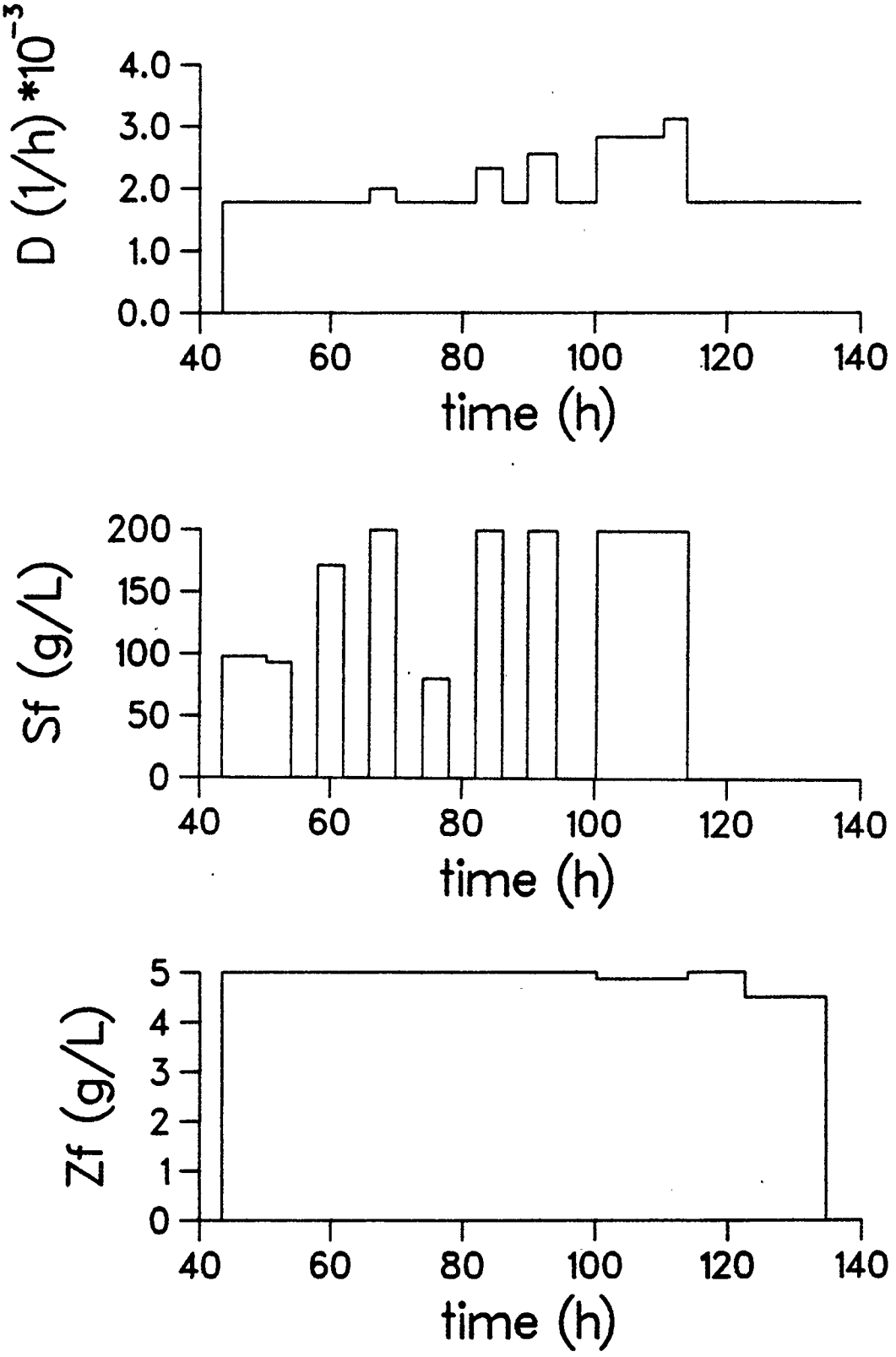

Figure 33 Manipulated variable response during Experiment #32

the biomass measurements, control of the growth was improved compared to the earlier runs. Precursor control once again shows significant overshoot but this is expected as the predicted consumption of PoAA is higher than the actual amount consumed through penicillin production. Penicillin production is better than it was in the earlier controlled runs, probably due to less variability in the reactor conditions early in the run. Fig. 34 shows the response of the controlled variables in the run and Fig. 37 shows the response of the manipulated variables. Fig. 35 is a plot of ln(biomass) vs time and illustrates the improved control of the growth rate about the set point. Fig. 36 demonstrates the sluggish response of the controller due to the constraints imposed, as the measured growth rate slowly returns to the set point after a large initial disturbance.

7.4 Penicillin Hydrolysis

In a further attempt to understand the poor penicillin productivities, the penicillin hydrolysis rate which was assumed to be negligible, was determined experimentally. This was simply accomplished by following product degradation using the HPLC and an actual sample from the fermentation broth. No effort was made to determine what the degradation products were. The results from this experiment can be seen in Figure 38. Assuming the degradation to be first order with time, the degradation rate determined through linear regression is 0.00216 h^{-1}. This value is similar to the value given by Nestaas and Wang (1986) of 0.003 h^{-1} and corresponds to a half life of approximately 464 hours.

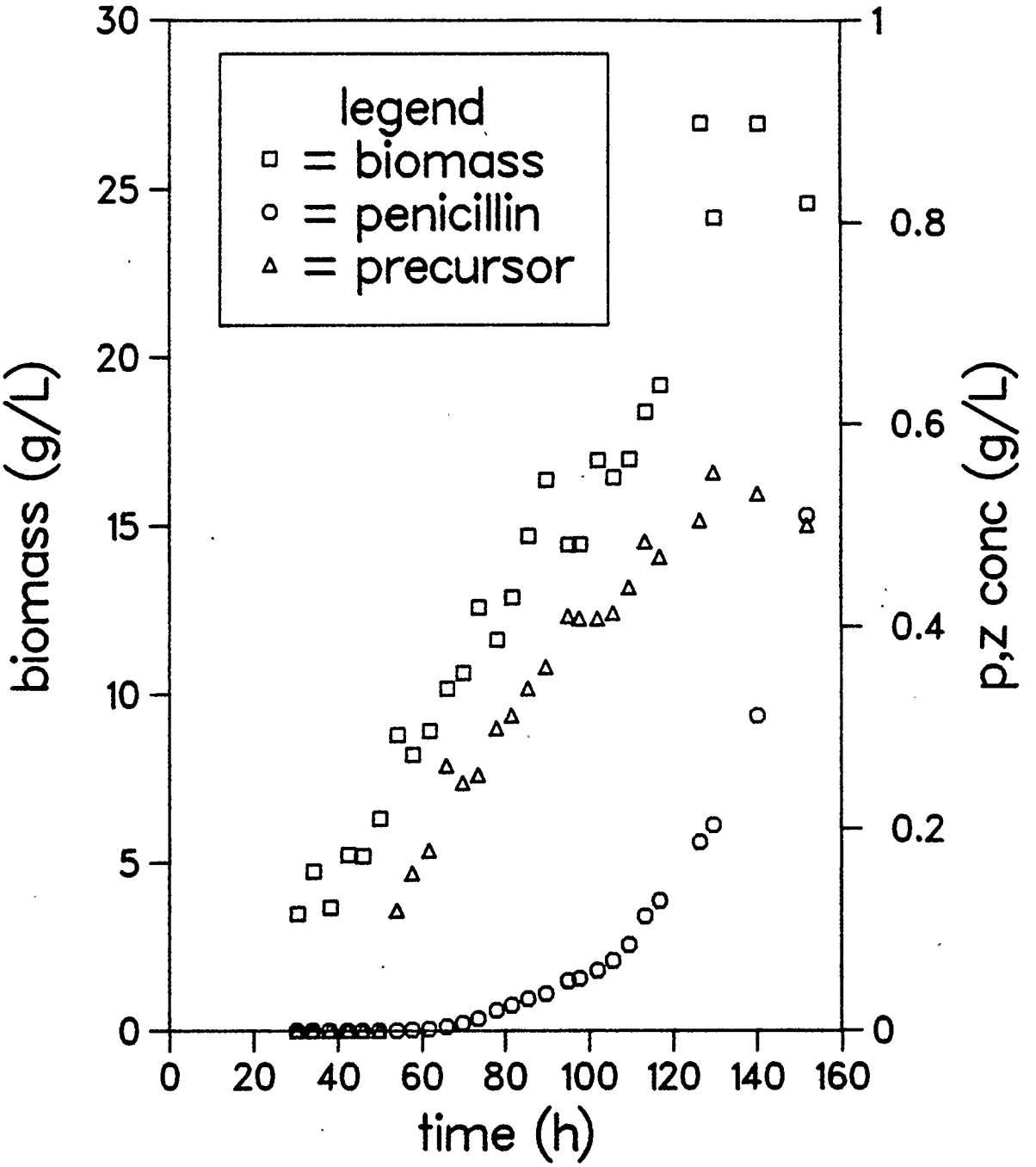

Figure 34 Biomass, penicillin and precursor transients during Experiment #33

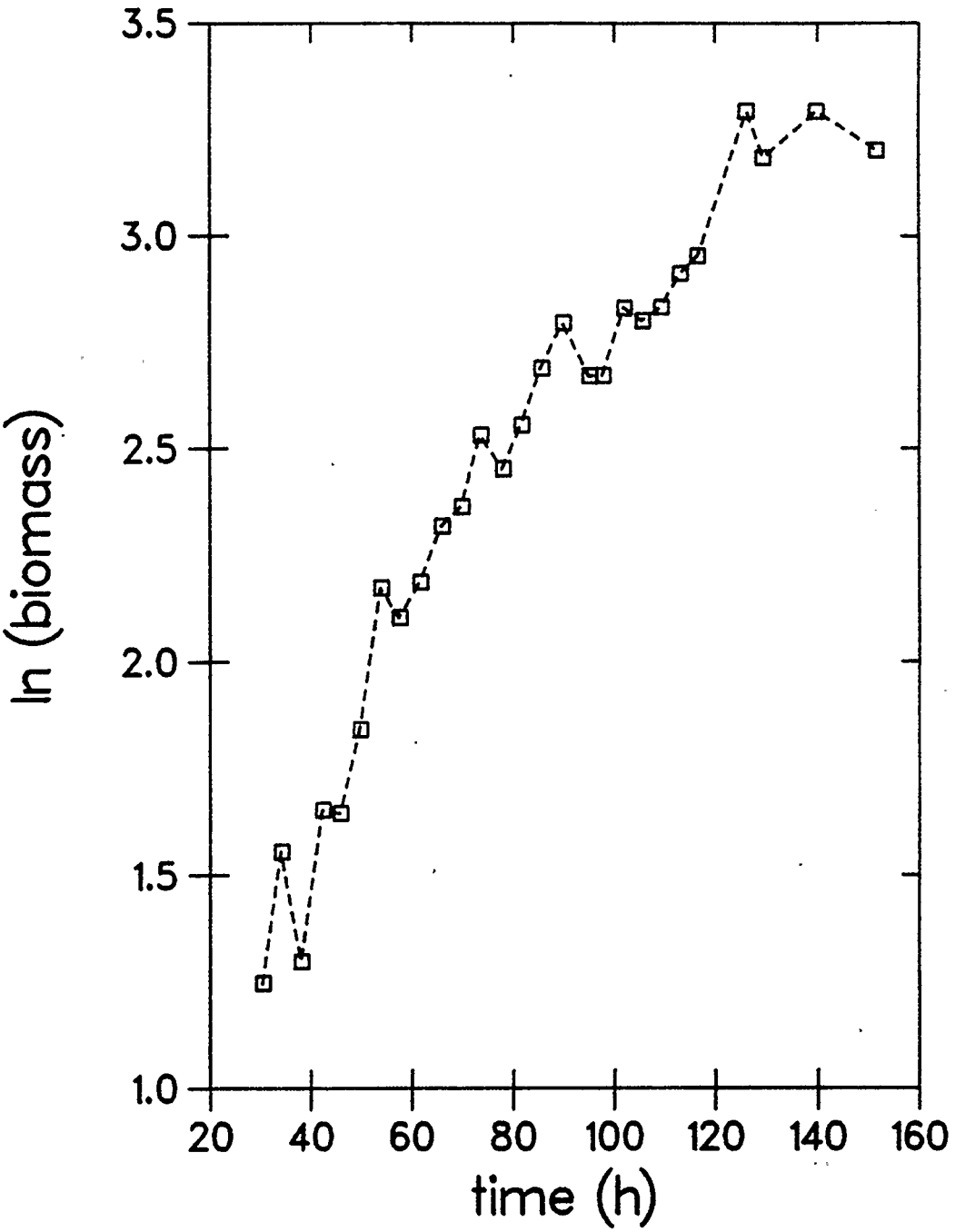

Figure 35 Plot of ln(biomass) vs time for Experiment #33 illustrating growth rate transients.

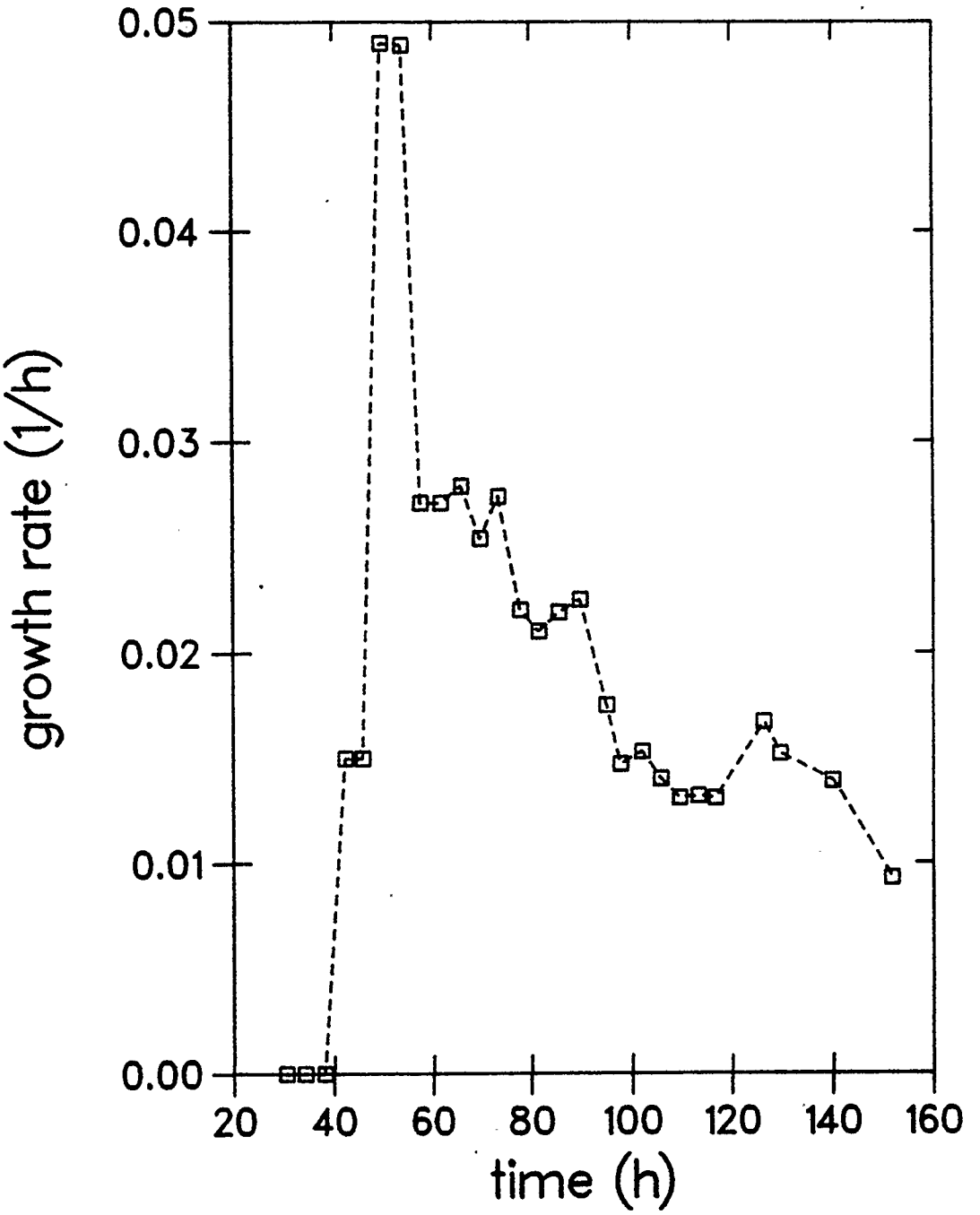

Figure 36 Plot of estimated growth rate measured through the Kalman filter for Experiment #33

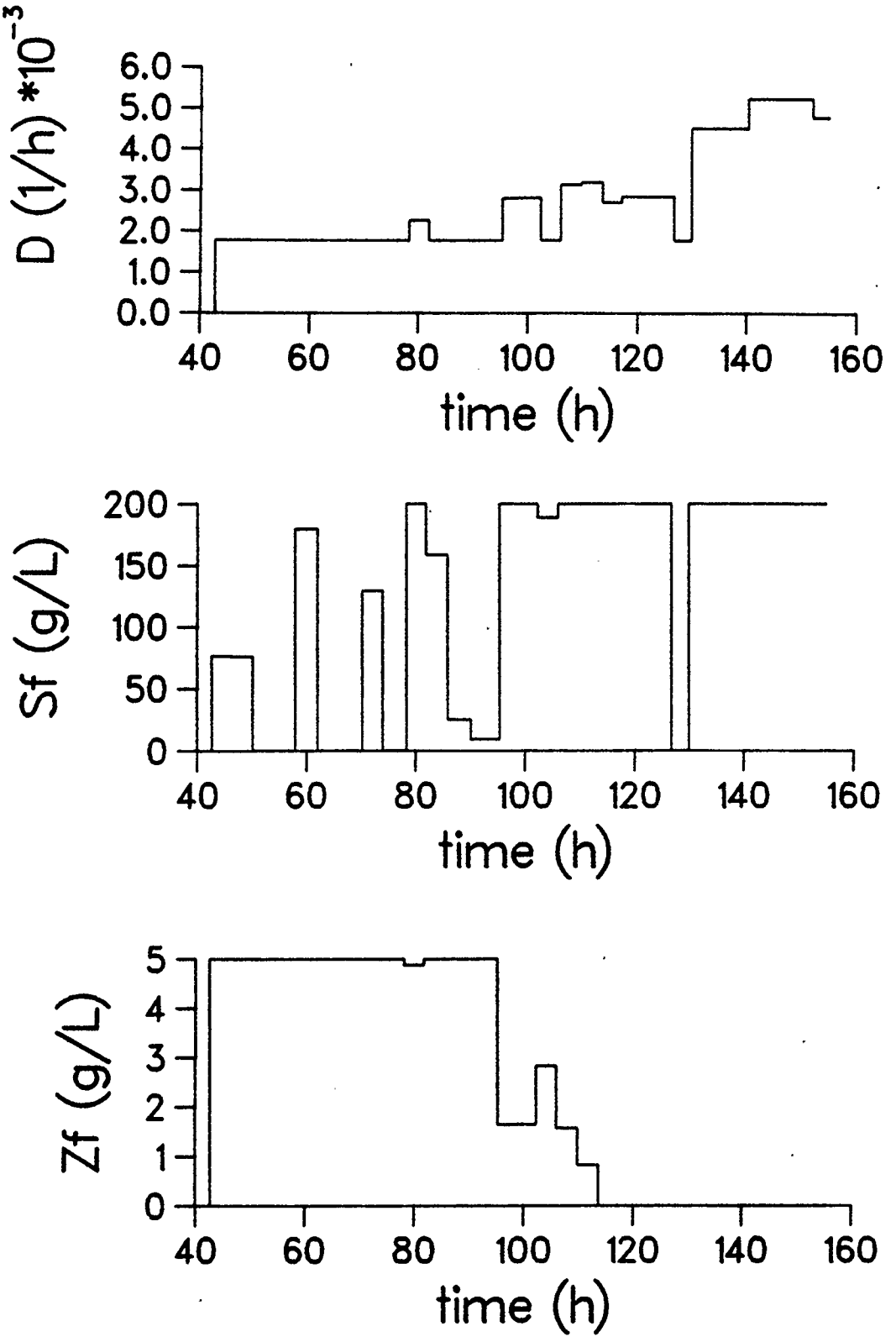

Figure 37 Manipulated variable behavior during Experiment #33

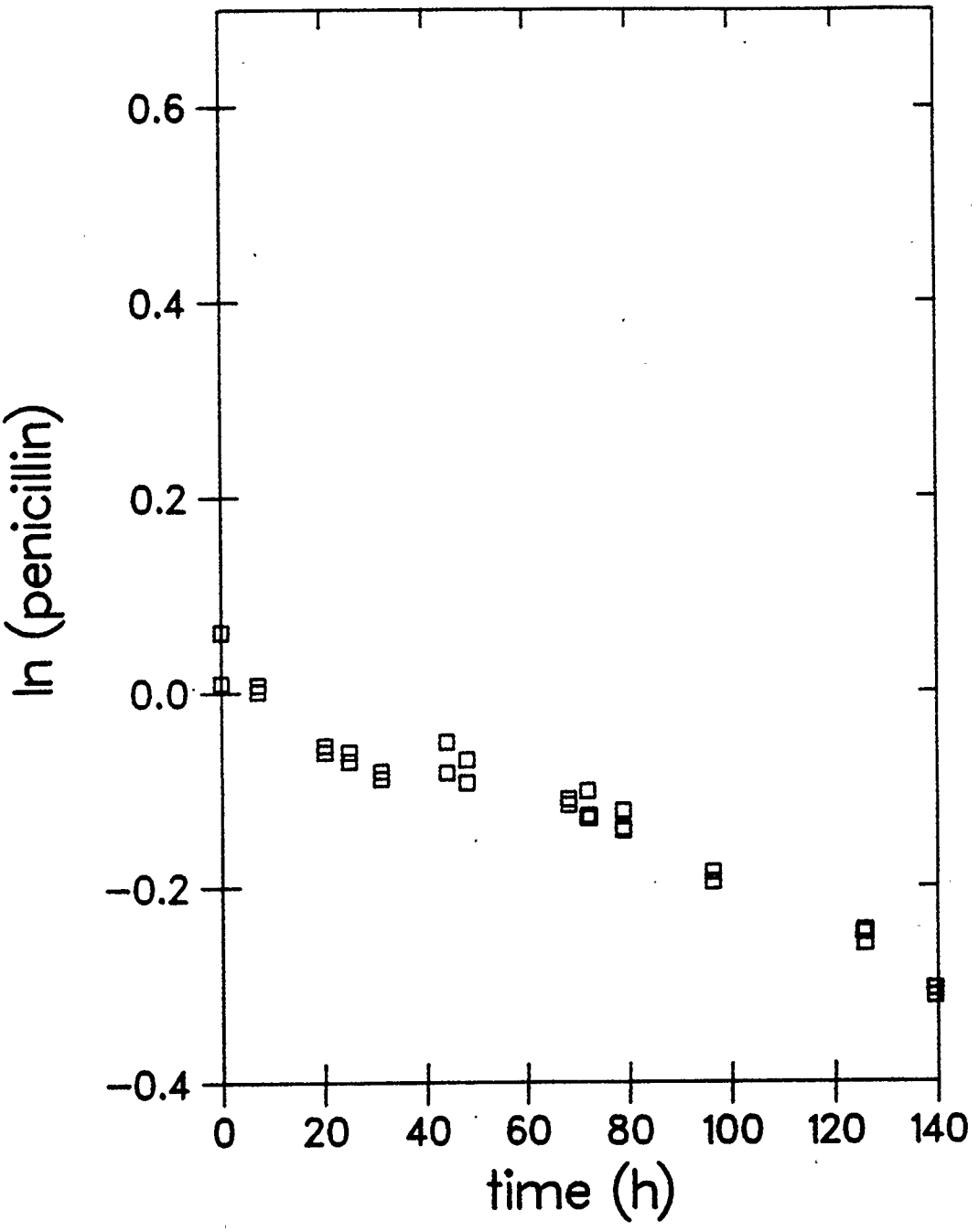

Figure 38 Plot illustrating the first order hydrolysis of penicillin with time.

CHAPTER VIII

DISCUSSION

The initial goal of this project was computer control of the penicillin fermentation at the quasi-steady-state. During the course of the experiments a second goal, to determine the reason for low penicillin productivities encountered, was pursued. Each of these goals will be discussed separately in the following sections.

8.1 Penicillin Productivity

Despite the generally low penicillin productivities, some marked differences were observed from run to run. Trends were noted, but the variability in data makes any findings inconclusive and only further experimentation could verify these findings. Nevertheless, these trends do point to some interesting factors which may affect the penicillin fermentation, and could help elucidate part of the physiological puzzle surrounding penicillin production.

The use of a semi-defined media to supply the mycelia with essential complex nutrients was the first variable tested. Corn steep liquor (CSL) is a commonly used energy source in industrial fermentations as it is cheap, readily available and supplies most of the nutrients required for penicillin fermentations. A small amount of CSL (3% w/v) had little effect on the growth rate, despite the additional energy supplied, and so accurate control based on the feed concentration of glucose was still feasible. Both the bench scale and

shake-flask experiments failed to show any increase in the penicillin productivity in the presence of 3% *w/v* CSL. However, the addition of CSL seemed to improve the reproducibilty of the pelleted growth of mycelia around the celite, and so it was decided to leave the CSL in the media formulations.

Because of the numerous differences between the shake-flask and bench scale experiments, it is impossible to pinpoint a single reason for the increased penicillin titers noted in the shake flask runs. Some of the major differences are: immobilized vs free-cell morphology; controlled growth rate vs uncontrolled repeated batch growth history; media differences (glucose vs lactose); reactor design and configuration; and oxygen transfer. For this reason, a free-cell experiment was conducted using the BIOENGINEERING AG fermentor to eliminate most of the experimental differences and concentrate on cell morphology. The results of this run showed an increase in penicillin production over previous experiments conducted by the author, but the results are similar to those for immobilized cells found by Linardos (1987). The experiments by Linardos, however, were plagued with large amounts of free cells up to 80% of the total biomass, and therefore could be considered essentially free-cell runs.

It is interesting to note that the biomass from glucose yield for the free cell experiment was much smaller than in the immobilized runs. Furthermore, oxygen transfer to the cells was significantly reduced and greater agitation was necessary to ensure a sufficiently

high DO-concentration. This confirms two of the major advantages of cell immobilization; reduced energy and substrate consumption resulting in reduced overall operating costs, and improved oxygen transfer within the reactor which is one of the major problems facing modern industrial penicillin fermentations.

Finally, the rate of penicillin hydrolysis was determined in order to quantify the effect of hydroysis on penicillin titers. As mentioned in Chapter VII, the first order rate constant for penicillin hydrolysis was determined to be 0.00216 h^{-1}. Simulations based on the results of Experiment #26 indicate that if penicillin hydrolysis is taken into consideration, the value for q_p^{max} would have to increase by 7.2% to account for penicillin losses due to degradation. Considering the spread in penicillin productivities obtained in the performed experiments, the hydrolyzed amount of penicillin is essentially negligible as originally suggested.

Penicillin productivities and titers predicted by the present model were consistently higher than the experimental values due to modelling errors and an inability to consistently maintain optimum environmental conditions. To correct the potential modelling error, the form of the productivity equation (equation 8) was studied. The present form makes physiological sense as it describes that early in the run while the cells are still young, productivity is low. As the cells age, the productivity peaks and then decreases as the cells grow old and lose their capability to synthesize penicillin. For this reason, the

basic form of the equation was altered as follows:

$$q_p = q_p^{max}(a\lambda)^c \, e^{[1-(a\lambda)^c]} \qquad (72)$$

where c is a constant which affects the shape of the productivity profile as it varies with average cell age. The above expression reduces to equation (8) when c = 1.

It was also found that equation (4), which accounts for the precursor concentration in the fermentor, consistently predicted higher concentrations in the reactor than were determined experimentally. The present mass balance failed to predict the plateaus in precursor concentration encountered in the open-loop experiments, followed by sharp increases in concentration due to falling penicillin productivity late in the runs. The failure to predict the sharp increase is due to the inability of the model to account for factors such as low oxygen concentration which affect the value of q_p late in the run. The reason for the inability to account for the plateaus is unknown, though it may be due to another reaction which consumes precursor (*i.e.* oxidation), or the formation of an intermediate complex as postulated in the structured model. To test this equation (4) was modified to include a first order consumption of precursor:

$$dz/dt = D(z_f - z) - q_p bx - q_z z \qquad (73)$$

By including this term it was possible to improve model predictions of experimental data though the model still gives poor predictions of precursor concentration.

A simulation based on Experiment #26 (see Figures 15, 16 and 17, Chapter VII) was performed, and the constant, c, was determined to obtain a best fit of the data. Figure 39 illustrates the productivity profile as a function of age for Experiment #26, and the fit is very good, supporting the choice of equation (72). Experimental values for q_p were determined, with hydrolysis accounted for, as are the predicted values. The nominal kinetic parameters used in Experiment #26 are: K_s = 0.2141 *g/L*, Y_G = 0.5991, a = 0.0215 h^{-1}, q_p^{max} = 0.015 h^{-1}, m = 0.011 h^{-1}, q_h = 0.00216 h^{-1}, q_z = 0.005 h^{-1}, c = 6.7, and λ_o = 36.5 *h*.

Experiment #34, using free non-immobilized mycelia, was also used to test the values predicted by the model with the proposed modifications. Figures 18,19 and 20 in Chapter VII show the results, and it can be said that the model does a good job predicting the concentration profiles with the exception of the precursor, though it does predict the temporary steady state. The nominal kinetic parameters used in this simulation are: K_s = 0.2141 *g/L*, Y_G = 0.5657, m = 0.015 h^{-1}, c = 4.0, q_p^{max} = 0.0015 h^{-1}, q_h = 0.00216 h^{-1}, q_z = 0.015 h^{-1}, a = 0.0145, and λ_o = 21.526 *h*.

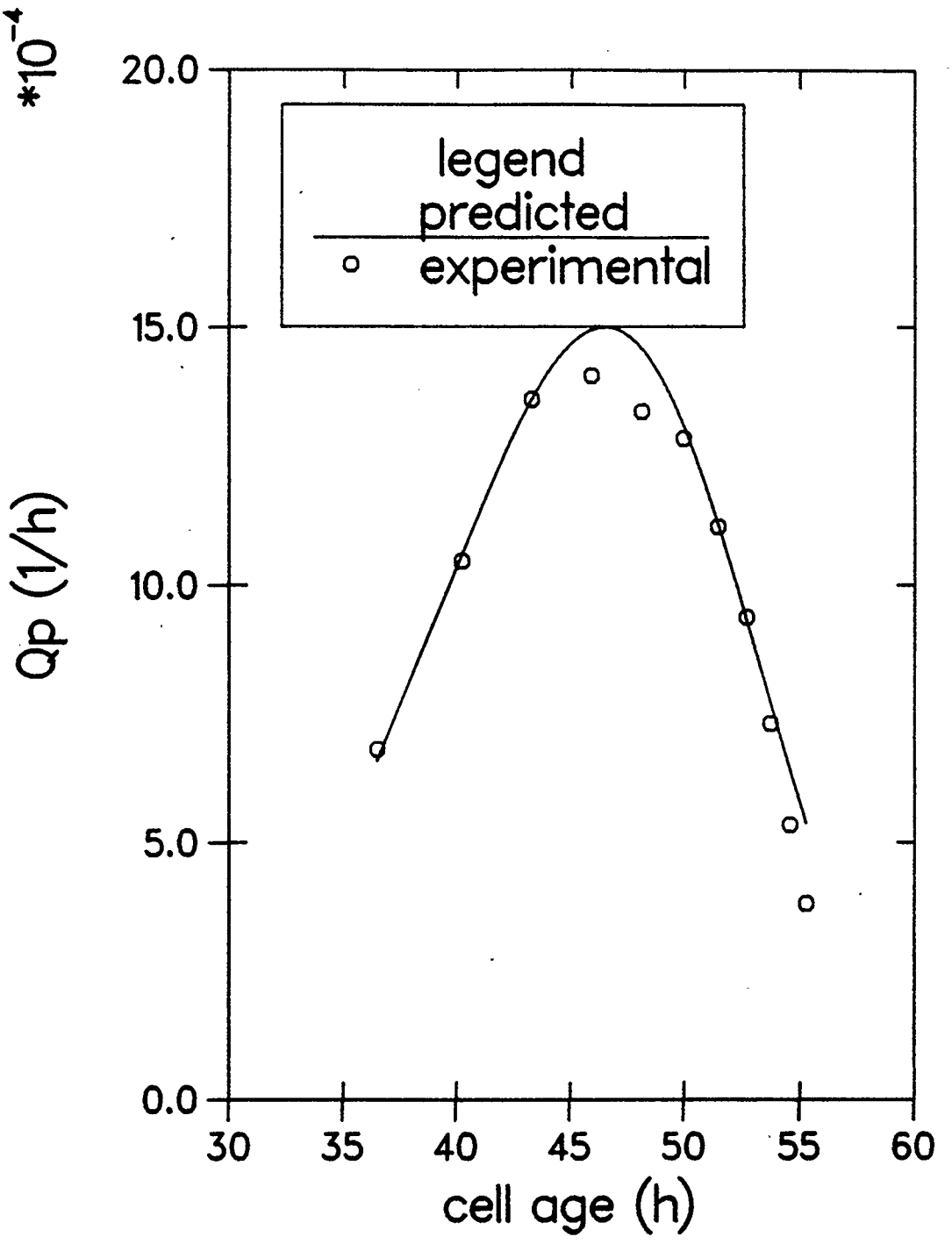

Figure 39 Plot of penicillin productivity, q_p, as a function of average cell age, λ, for Experiment #26

8.2 Computer Control

The control algorithm initially proposed by Kalogerakis *et al.* (1986) was designed under the assumption that the controlled variables of penicillin, precursor and, indirectly, growth rate could be measured on-line. Initial simulations by Linardos (1987) did not account for any dead-time in measurements and as a result, simulated results showed excellent control. On-line measurement of growth rate was assumed possible through off-gas analysis using a mass spectrometer. In this study, however, a 90 *min* deadtime was introduced through manual off-line analysis techniques, and this was found to have a large effect on the controller's performance.

The major effect of the mentioned deadtime was slow response to an erroneous control action caused by sampling errors. Considering the sampling interval imposed due to manpower restrictions, an erroneous control action could result in up to four hours of either over or under-feeding of substrate before it was detected. This condition occurred in early control experiments and seemed to have an adverse effect on penicillin productivity as was particularly noted in Experiment #31. Correction of over-feeding by turning off substrate feed probably caused a rapid decline in mould viability and irreversible damage to the organism's ability to synthesize penicillin by a momentary reduction in the specific growth rate. This reduction in productivity below a critical growth rate was noted by Pirt and Righelato (1967). Likewise, correction to undershoot by large additions of substrate, subjects the mould to catabolite repression of penicillin

production. To counter this problem, the value for the estimated specific uptake rate for glucose was constrained so as to ensure that the minimum maintenance requirement for glucose was always available for the cells, and to avoid the shock of sudden increases in glucose concentration. With this constraint in place, proper control was possible, though the response was more sluggish.

Another problem experienced was the divergence of the growth rate estimator from the actual growth rate, even with accurate initial values for biomass. This was the most evident in Experiment #28 where the measured and actual growth rate differed by 15%. This divergence was improved in later experiments by changing the value of the forgetting factor in the Kalman filter, λ_c, to 0.75 from 0.95. Estimator divergence is a result of data saturation. As the filter gain approaches zero, any new system information is ignored, though this is a problem later in a run when operating conditions should be well established. The most important time for accurate control of the fermentation is early in the run so that differences in inoculum and media can be accounted for. If the proposed model were exact, then this would not be a problem, but as this obviously is not the case, estimate divergence does occur. Controlling the covariance matrix so that it doesn't approach zero could prevent this problem.

The goal of the control scheme was to optimize penicillin productivity by extending the fermentation period and maintaining the growth rate above a certain critical value. The control scheme was able

to maintain this growth rate after an initial period of adjustment, but it is believed that this initial period is critical to successful penicillin fermentations, and fluctuations in this region were detrimental to this goal. This problem was curtailed by tuning the controller and constraining control action to make response sluggish. The use of manual off-line sampling was a constraint placed on the present apparatus, but still enabled reasonably good performance. If the problem of on-line measurement and the encountered dead-time could be solved, increased speed of response would smooth out the fluctuations found early in the fermentation. Another method of avoiding variations encountered early in the runs would be to start control earlier in the fermentation during the initial growth phase. Due to the dynamic nature of the fermentation, however, fixed parameter control would be inadequate.

This study has shown that reasonable control of an immobilized cell penicillin fermentation is possible using manual off-line measurements with a 90 minute dead time in controller feedback, and a long sampling interval of 4 hours. The proposed model upon which the control algorithm was based gave very good predictions of the experimental data when the proposed modifications were implemented. Perhaps control could be further improved by matching initial environmental conditions to the parameters "a", "c" and q_p^{max} to account for variations from run to run.

8.3 Supplement

Calam (1986) has stated that the field of antibiotic process development is a mixture of chance, biochemical and physiological understanding, and the adaptation of a biological process to within limits defined by the fermentation equipment. Furthermore, the subject largely remains an observational science. In the present study, it was observed that the maintenance requirements of immobilized cells is 50% lower than that for free cells. This savings in substrate consumption could be very significant, but is offset by a rate of penicillin production, q_p, which was found to be 50% lower for immobilized cells than for free mycelia. It is unfortunate that these findings are for a low yielding strain of *P.Chrysogenum*. As Calam (1986) has argued, the higher the industrial firms are able to lift their veil of secrecy and reveal the behavior of high-producing strains, the better it will be for the field of biotechnology.

CHAPTER IX

CONCLUSIONS AND RECOMMENDATIONS

9.1 Conclusions

(1) Two simple and separately derived models have been developed which show the major dynamic characteristics of the penicillin fermentation.

(2) A modified form of the unstructured kinetic model was successful in predicting biomass and penicillin concentrations as well as penicillin productivity, q_p, as a function of average cell age.

(3) Kinetic constants for the unstructured model have been determined through open loop experiments.

(4) The rate of penicillin hydrolysis under experimental conditions was determined to be 0.00216 h^{-1}.

(5) A control scheme has been developed to control the penicillin fermentation and has been successfully tested using computer simulations on both structured and unstructured models.

(6) A sluggishly tuned controller was successful in controlling growth rate and precursor concentration during experiments on the penicillin fermentation in a 19L bench scale fermentor.

(7) It was confirmed that immobilized-cell fluid bed bioreactors consume less substrate and have improved oxygen transfer characteristics compared to free-cell CSTR operations.

(8) Experiments show that the values for penicillin productivity, q_p, and maintenance requirement, m, are lower for immobilized cells than those for free cells under similar operating conditions.

9.2 Recommendations

(1) A method for on-line measurement of biomass, penicillin and precursor should be implemented to eliminate the deadtime present in the present controller.

(2) The possibility of an adaptive control scheme which would begin control during the the initial growth phase should be investigated.

(3) The present control algorithm should be fine tuned to prevent transient over and under-feeding of substrate.

(4) The kinetic model should be modified to account for catabolite repression and other factors which affect penicillin productivity.

(5) More experiments should be conducted to determine the kinetic parameters to a higher degree of confidence and under different environmental conditions.

(6) More experiments should be conducted to quantitatively determine the effect of immobilization on penicillin productivity.

(7) The kinetic model should be altered to more accurately describe precursor consumption.

(8) Further experimentation should be conducted to determine the reasons for the low penicillin productivities encountered.

CHAPTER X

REFERENCES

1. Bajpai, R.K. and M. Reuß, "Evaluation of Feeding Strategies in Carbon Regulated Secondary Metabolite Production Through Mathematical Modelling", Biotech. Bioeng., 23, 717-738 (1981)

2. Bajpai, R.K. and M. Reuß, "A Mechanistic Model for Penicillin Production", J. Chem. Tech. Biotechnol., 30, 332-344 (1980)

3. T.J. Boyle, Proceedings of the Joint Automatic Control Conference, ISA, Pittsburgh, PA, 1978

4. Calam, C.T. and D.W. Russell, "Microbial Aspects of Fermentation Process Development", J. Appl. Chem. Biotechnol.,23, 225 (1973)

5. Constantinides, A, J.L. Spencer and E.L. Gaden Jr., "Optimization of Batch Fermentation Processes: I. Development of Mathematical Models for Batch Penicillin Fermentations", Biotech. Bioeng., 12, 803-830 (1970)

6. Cooney, C.L, H.Y. Wang and D.I.C. Wang, "Computer Aided Material Balancing for Prediction of Fermentation Parameters", Biotech. Bioeng., 19, 55-67 (1977)

7. Demain, A.L., "Biochemistry of Penicillin and Cephalosporin Fermentation", Lloydia, 37, 147-167 (1974)

8. Deo, Y.M. and G.M. Gaucher, "Semicontinuous and Continuous Production of Penicillin-G by *Penicillium Chrysogenum* Cells Immobilized in κ-Carrageenan Beads" Biotech. Bioeng, 26, 285-295 (1984)

9. Gbewonyo, K. and D.I.C. Wang, "Confining Mycelial Growth to Porous Microbeads: A Novel Technique to Alter the Morphology of Non-Newtonian Mycelial Cultures", Biotech. Bioeng., 25, 967-983 (1983)

10. Heijen, J.J. and J.A. Roels, "Application of Balancing Methods in Modelling the Penicillin Fermentation", Biotech. Bioeng., 21, 2175-2201 (1979)

11. Jones, A., D.N. Wood, T. Razniewska, G.M. Gaucher and L.A. Behie, "Continuous Production of Penicillin-G by *Penicillium Chrysogenum* Cells Immobilized on Celite Biocatalyst Support Particles", Can. J. Chem. Eng., 64, 547-552 (1986)

12. Kalogerakis, N.E., T. Linardos, L.A. Behie, W.Y. Svrcek and G.M. Gaucher, "Computer Control of Continuous Immobilized Cell Penicillin Fermentations: Simulation Studies", Can. J. Chem. Eng., 64, 581-587 (1986)

13. Kalogerakis, N.E. and T.J. Boyle, "Experimental Evaluation of a Quasi-Steady-State Controller for Yeast Fermentation", Biotech. Bioeng., 23, 921-938 (1981)

14. Kim, J.H., D.K. Oh, S.K. Park, Y.H. Park and D.A. Wallis, "Production of Penicillin in a Fluidized Bed Bioreactor Using a Carrier Supported Mycelial Growth", Biotech. Bioeng., 28, 1838-1844 (1986)

15. Konig, B., K. Schuegerl, and C Seewald, "Strategies for Penicillin Fermentation in Tower-Loop Reactors", Biotech. Bioeng., 24, 259-280 (1982)

16. Linardos, T., "Modelling and Computer Control of Continuous Immobilized Cell Penicillin Fermentations" MSc. Thesis, University of Calgary (1987)

17. Lorenz, T., J. Diekmann, K. Frueh, R. Hiddessen, J. Moeller, J. Niehoff and K Schügerl, "On-line Measurement and Control of Penicillin-V Production in a Tower Loop Reactor", J. Chem. Tech. Biotechnol., 38, 41-53 (1987)

18. Megee, R.D., S. Kinoshita, A.G. Fredrickson and H.M. Tsuchiya, Biotech. Bioeng., 12, 771 (1970)

19. Metz, B., and N.W.F. Kossen, "BIOTECHNOLOGY REVIEW: The Growth of Molds in the Form of Pellets — A Literature Review", Biotech. Bioeng, 19, 781-799 (1977)

20. Montague, G.A., A.J. Morris, A.R. Wright, M. Aynsley and A. Ward, "Modelling and Adaptive Control of Fed-Batch Penicillin Fermentation", Can. J. Chem. Eng., 64, 567-580 (1986)

21. Morikawa, Y., I. Karube and S. Suzuki, "Penicillin-G Production by Immobilized Whole Cells of *Penicillium Chrysogenum*", Biotech. Bioeng., 21, 261-270 (1979)

22. Nestaas, E. and D.I.C. Wang, "Computer Control of the Penicillin Fermentation Using the Filtration Probe in Conjunction with a Structured Process Model", Biotech. Bioeng., 25, 781-796 (1983)

23. Nestaas, E. and D.I.C. Wang, "A New Sensor, The Filtration Probe for Quantitative Characterization of the Penicillin Fermentation: I Mycelial Morphology and Culture Activity", Biotech. Bioeng., 23, 2803-2813 (1981)

24. Pirt, S.J., "Microbial Physiology in the Penicillin Fermentation", TIBTECH, 5, 69-72 (1987)

25. Pirt, S.J., "The Penicillin Fermentation: A Model for the Development of Antibiotic Fermentations", Kem. Ind., 34, 13-19 (1985)

26. Pirt, S.J., "The Role of Microbial Physiology in Biotechnology", J. Chem. Tech. Biotechnol., 33B, 137-138 (1983)

27. Pirt, S.J., "The Theory of Fed-Batch Culture with Reference to the Penicillin Fermentation", J. Appl. Chem. Biotechnol., 24, 415-424 (1974)

28. Pirt, S.J. and R.C. Righelato, "Effect of Growth Rate on the Synthesis of Penicillin by *Penicillium Chrysogenum* in Batch and Chemostat Cultures", Appl. Microbiol., 15, 1284-1290 (1967)

29. Ryu, D.D.Y. and J. Hospodka, "Quantitative Physiology of *Penicillium Chrysogenum* in Penicillin Fermentations", Biotech. Bioeng., 26, 289-298 (1980)

30. Stephanopoulos, G. and K.Y. San, " Studies on On-Line Bioreactor Identification. I. Theory", Biotech. Bioeng., 26, 1176-1188 (1984)

31. Svrcek, W.Y., R.F. Elliot and J.E. Zajic, "The Extended Kalman Filter Applied to a Continuous Culture Model", Biotech. Bioeng., 16, 827-846 (1974)

32. Swartz, R.W., "Chapter 2: Penicillins", Comprehensive Biotechnology, The Principles, Applications and Regulations of Biotechnology in Industry, Agriculture and Medicine. Volume 3. Pergamon Press, Toronto ON. (1985)

APPENDIX A

MEDIA COMPOSITION

A.1 Sporulation Medium

The solid surface growth sporulation medium contained per *liter*: 6*g* peptone; 4*g* casaminoacids; 3*g* yeast extract; 1.5*g* beef extract; 20*g* malt extract; 40*g* bactoagar; 1*g* glucose; 20*g* KH_2PO_4; 2 *mL* $CaCl_2$ solution (25 *g/L*) and 10 *mL* of trace metal solution. The trace metal solution contained per liter: 25*g* $MgSO_4 7H_2O$; 10*g* $FeSO_4 7H_2O$; 10*g* $ZnSO_4 7H_2O$; 2*g* $MnSO_4 H_2O$; and 0.5*g* $CuSO_4 5H_2O$.

A.2 Growth Medium

The growth medium contained per liter: 25 *g* glucose (varied from run to run); may contain corn steep liquor 30 *mL* (varied); 8.1*g* NH_4Cl; 3*g* Na_2SO_4; 3*g* KH_2PO_4; 300 *mg* $FeSO_4 7H_2O$; 6*g* MES Hydrate; 30*mL* trace metals and 6*mL* $CaCl_2$ solution. The growth medium was adjusted to pH 6.80.

A.3 Production Media

For the open loop experiments, production medium A contained per liter: 50*g* glucose; 30*mL* CSL; 10*g* NH_4Cl; 19.8*g* K_2SO_4; 1.25*g* $MgSO_4 7H_2O$; 7.5*g* KH_2PO_4; 25*mg* $FeSO_4 7H_2O$; and 30*mL* of trace metals. Production medium B was identical to medium A with the addition of 2*g* of precursor phenoxyacetic acid per liter. Again, the glucose and CSL concentrations may vary from run to run and CSL may not always be present.

For the closed loop experiments, production medium A contained per liter: 200*g* glucose; 16.7*mL* CSL; 40*g* NH_4Cl; 79*g* K_2SO_4; 5*g* $MgSO_4 7H_2O$; 30*g* KH_2PO_4; 50*mg* $FeSO_4 7H_2O$; 2*g* PoAA; and 20*mL* of trace metals. Production medium B contained only 5 *g/L* of precursor in water and medium C contained only water.

A.4 Shake Flask Experimental Media

The growth medium for the shake flask experiments contained per liter: 10 *g* glucose; 17.1*g* lactose; 30*mL* CSL; 2.7*g* NH_4Cl; 1*g* Na_2SO_4; 1*g* KH_2PO_4; 5.85*g* MES Hydrate; 2 *mL* $CaCl_2$ solution; and 10 *mL* of trace metals solution. The production medium contained 100*g* lactose; 30 CSL; 68*g* KH_2PO_4; 39.5*g* K_2SO_4; 20*g* NH_4CL; 2.5*g* $MgSO_4 7H_2O$; 3*g* PoAA; and 10*mL* of trace metal solution.

The above media are all formulated for the E15 strain of *Penicillium Chrysogeneum*, but in the shake flask experiments the Pan Labs P2 strain was also tested. For this strain the growth medium contained per liter: 30*g* glucose; 10*g* lactose; 30*mL* CSL; 2*g* $(NH_4)_2SO_4$; 5*g* $CaCO_3$; 0.5*g* KH_2PO_4; 10*g* Pharmamedia; and 10*g* yeast extract. The production medium contained per liter: 120*g* lactose; 27.5*g* Pharmamedia; 10*g* $(NH_4)_2SO_4$; 10 *g* $CaCO_3$; 0.5*g* KH_2PO_4; 5*g* K_2SO_4; 10*g* vegetable oil (substitute for lard oil); and 10*g* PoAA.

APPENDIX B

ANALYTICAL METHODS

B.1 Biomass Measurement

Biomass concentration within the fermentor was estimated by taking the dry cell biomass. Approximately 50 *mL* of broth was collected and the cells separated by filtering the broth through Whatman No. 1 filter paper by vacuum filtration. The collected cells were then dried to constant weight in a microwave oven (approximately 10 *min*). In the case of the shake flask and free-cell experiments, the volume of broth removed from the reactor with cells was measured in a graduated cylinder. The measured dry weight divided by the collected volume then represented the cell concentration. For the immobilized cells, the amount of celite needed to be known so the dried sample was placed in a muffle furnace at 500°C for 30 *min* and then weighed. Biomass concentration was calculated by taking the weight difference between the dried and burned sample (actual biomass present in the sample with 5% ash accounted for in the burned sample), divided by the weight of celite in the sample and then multiplying by the celite concentration in the reactor. The actual immobilized cell concentration could be determined by washing the broth sample several times and decanting away the free cells before the above procedure is conducted.

B.2 Glucose Measurement

The glucose concentration was measured using two different enzymatic tests to determine UV adsorbance differences on a

spectrophotometer. The two assay kits used were the Worthington Statzyme glucose assay (Cooper Biomedical Inc., Malvern PA) and the D-Glucose assay from Boehringer Mannheim. The detection limit was approximately 1 *mg/L* and the procedure followed is as described with the assay.

B.3 Penicillin-V and PoAA Determination

Penicillin-V and PoAA concentration were determined using High Performance Liquid Chromatography (HPLC) on a Hewlett-Packard 1084B liquid chromatograph equipped with an auto-injector and variable UV wavelength detector. Filtered broth samples were centrifuged to remove particulate matter and injected through a 25 *cm* x 0.46 *cm* RP-8 (reverse phase) column (Brownlee Lab. Inc., Santa Clara, CA) using the following conditions:

solvent A : 0.075 N NaH_2PO_4, pH 4.7

solvent B : acetonitrile (CH_3CN)

solvent flow rate = 1.0 *mL/min*

solvent temp. = 35.0°C

Elution program :

Time	0 *min*	%B = 10
Time	4 *min*	%B = 10
Time	20 *min*	%B = 35
Time	28 *min*	%B = 10
Time	30 *min*	STOP RUN

The detector signal to response wavelengths were 220 : 430 and under these conditions, the PoAA peak appeared at 8.4 *min* and the penicillin-V peak appeared at 22.1 *min*.

Penicillin hydrolysis was followed using a new Hewlett-Packard HP1090 HPLC with an HP hypersil ODS 5μm reverse phase column (100 x 2.1 mm). The solvents and other conditions used with the HP1084B were used except a smaller solvent flowrate (0.1 ml/min) was used with the following elution program:

time 0 *min* %B = 20

time 2 *min* %B = 30

time 4 *min* %B = 30

time 5 *min* %B = 20 STOP RUN

For this system, the PoAA peak came out at 1.4 *min* and the penicillin peak came out at 3.4 *min*. Using both instruments, the PoAA and penicillin were quantified through the external standard method. A sample chromatograph from the HP1090 can be seen in figure B-1.

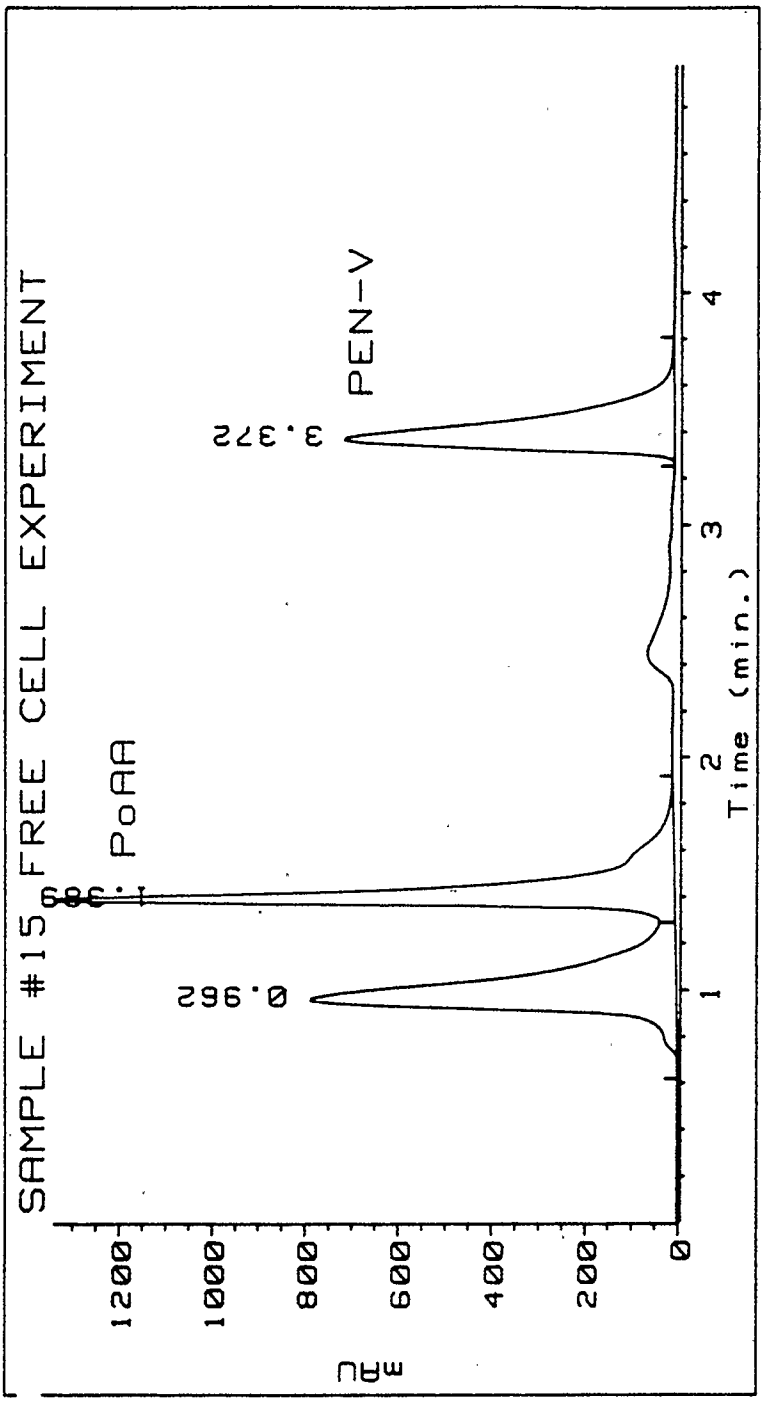

Figure B-1 Sample chromatograph from the HP1090 HPLC system clearly showing both the penicillin-V and phenoxyacetic acid peaks.

APPENDIX C

COMPUTER PROGRAMS

APPENDIX C.1

Open Loop Control Program

```
1 DEF SEG
2 A=0:I=0:J=0:ADR=0:LABSOFT.SEG=0:O=0
3 DIM ZPRGM%(150):A=VARPTR(ZPRGM%(0)) ' Get a pointer to the array
4 IF A<0 THEN A=A+65536!
5 FOR I=5 TO 11:READ J:POKE A+I,J:NEXT ' Poke program into array
6 FOR I=20 TO 108:READ J:POKE A+I,J:NEXT
7 POKE A+21,A-INT(A/256)*256:POKE A+22,INT(A/256) 'Poke in the address
8 DATA &H42,&h41,&h53,&h4c,&h49,&h42,&h00
9 DATA &hbb,&h00,&h00,&h1e,&h06,&h2e,&h8c,&h97,&h0e,&h00,&h2e,&h89,&ha7
10 DATA&h00,&h8c,&hc8,&h8e,&hd8,&h8e,&hd0,&hc6,&h87,&h04,&h00,&h00,&h8d
11 DATA&h0d,&h01,&hb4,&h3d,&hb0,&h00,&h8d,&h97,&h05,&h00,&h53,&hcd,&h21
12 DATA&h17,&h5b,&h53,&h8d,&h97,&H00,&h00,&h50,&h8b,&hd8,&hb4,&h3f,&hb9
13 DATA&h00,&hcd,&h21,&h5b,&h72,&H04,&hb4,&h3e,&hcd,&h21,&h5b,&h73,&h09
14 DATA&h87,&h00,&h00,&hc6,&h87,&h04,&h00,&hff,&h8b,&ha7,&h0c,&h00,&h8e
15 DATA&h0e,&h00,&h07,&h1f,&hcb,&h0c,&ha7,&h72,&h04,&h89,&h97
16 ADR= +20:CALL ADR ' Get address of the device driver
17 IF PEEK(A+4)=255 THEN BEEP:PRINT"*** ERROR - LABBASIC.COM Device
   Driver Is Not Installed":END
18 LABSOFT.SEG=PEEK(A)+256*PEEK(A+1)
19 IF LABSOFT.SEG<0 THEN LABSOFT.SEG=LABSOFT.SEG+65536!
20 O=PEEK(A+2)+256*PEEK(A+3)+197
21 DEF SEG = LABSOFT.SEG
22 :
23 COMPAT=PEEK(O+0)+256*PEEK(O+1)
24 SETSTAT=PEEK(O+2)+256*PEEK(O+3)
25 AINFM=PEEK(O+6)+256*PEEK(O+7)
26 AINM=PEEK(O+8)+256*PEEK(O+9)
27 AINS=PEEK(O+10)+256*PEEK(O+11)
28 AINSC=PEEK(O+12)+256*PEEK(O+13)
29 AINTS=PEEK(O+14)+256*PEEK(O+15)
30 AOUFM=PEEK(O+16)+256*PEEK(O+17)
31 AOUM=PEEK(O+18)+256*PEEK(O+19)
32 AOUS=PEEK(O+20)+256*PEEK(O+21)
33 AOUSC=PEEK(O+22)+256*PEEK(O+23)
34 BCDINM=PEEK(O+24)+256*PEEK(O+25)
35 BCDINS=PEEK(O+26)+256*PEEK(O+27)
36 BCDINTS=PEEK(O+28)+256*PEEK(O+29)
37 BCDOUM=PEEK(O+30)+256*PEEK(O+31)
38 BCDOUS=PEEK(O+32)+256*PEEK(O+33)
39 BINM=PEEK(O+34)+256*PEEK(O+35)
40 BINS=PEEK(O+36)+256*PEEK(O+37)
41 BINTS=PEEK(O+38)+256*PEEK(O+39)
42 BITINS=PEEK(O+40)+256*PEEK(O+41)
43 BITINTS=PEEK(O+42)+256*PEEK(O+43)
44 BITOUS=PEEK(O+44)+256*PEEK(O+45)
```

```
45 BOUM=PEEK(0+46)+256*PEEK(0+47)
46 BOUS=PEEK(0+48)+256*PEEK(0+49)
47 :
48 STINM=PEEK(0+50)+256*PEEK(0+51)
49 STINS=PEEK(0+52)+256*PEEK(0+53)
50 STINTS=PEEK(0+54)+256*PEEK(0+55)
51 CINM=PEEK(0+56)+256*PEEK(0+57)
52 CINS=PEEK(0+58)+256*PEEK(0+59)
53 CINTS=PEEK(0+60)+256*PEEK(0+61)
54 CSET=PEEK(0+62)+256*PEEK(0+63)
55 BEEPFUN=PEEK(0+64)+256*PEEK(0+65)
56 CONTIN=PEEK(0+66)+256*PEEK(0+67)
57 DELAY=PEEK(0+68)+256*PEEK(0+69)
58 DINS=PEEK(0+70)+256*PEEK(0+71)
59 DOUS=PEEK(0+72)+256*PEEK(0+73)
60 NORMAL=PEEK(0+74)+256*PEEK(0+75)
61 STATS=PEEK(0+76)+256*PEEK(0+77)
62 STOPFUN=PEEK(0+78)+256*PEEK(0+79)
63 VERSION=PEEK(0+80)+256*PEEK(0+81)
64 TINFM=PEEK(0+82)+256*PEEK(0+83)
65 TLIN=PEEK(0+84)+256*PEEK(0+85)
66 TINM=PEEK(0+86)+256*PEEK(0+87)
67 TINS=PEEK(0+88)+256*PEEK(0+89)
68 TINSC=PEEK(0+90)+256*PEEK(0+91)
69 TINTS=PEEK(0+92)+256*PEEK(0+93)
70 :
71 :
100 '            OPEN LOOP RUN PROGRAM   - NO PREDICTION
110 '       Rev. 12.0  -SEPTEMBER 17, 1987 Sean P. Forestell
120
'-------------------------------------------------------------------
130 ON ERROR GOTO 11000
140 '------------------dimensions-----------------
160 DIM FORMAT%(5),BC$(20),PLV(900),Y(10),BM$(10),T(10)
200 DIM LAST.GRF(3),FORMAT.GRF%(4),PLT.GRF(291)
210 DIM RAW2%(500),FGAL%(15),MP$(10),RAW1%(500)
215 DIM OFFP(3),IMIN(3),SLP(3)
230 '------------------Kinetic data--------------
240 READ K1 ,K2,KA0,KA1,YD,MGY,BETA,K3,K4,K5,K6,YP
250 DATA 0.123,1.0,0.001,0.0145,0.5,.007,.407,.001,3.0,.018,2.625,1.2
260 '-----------------Controller Parameters -----------
270 READ KCP,SA,SP
280 DATA 0.01,200.,12
290 '-----------------Pump Calibration data --------------
293  IMIN(1) = 5.2    :IMIN(2) = 5.2   :IMIN (3) =5.2
297  OFFP(1) =4.3367 :OFFP(2) =4.6619 :OFFP(3) =4.2549
299  SLP (1) = 2.9446 :SLP (2) =2.6506 :SLP(3) = 2.8385
300  PA%=2 :PB%=3 :PO%=1
301  IAMIN = IMIN(1) :OFFA =OFFP(1) : SLA = SLP(1) :' Defaults
303  IBMIN = IMIN(3) :OFFB =OFFP(3) : SLB = SLP(3)
305  IOMIN = IMIN(2) :OFFO =OFFP(2) : SLO = SLP(2)
```

```
306  FAMAX = (20-OFFA)/SLA: FBMAX = (20!-OFFB)/SLB
307  FOMAX = (20!-OFFO)/SLO: FOMIN=(IOMIN-OFFO)/SLO
308  FAMIN=(IAMIN-OFFA)/SLA :FBMIN=(IBMIN-OFFB)/SLB
309 LPRINT " ### PUMP ARRANGEMENT : A , B , O  : "
    ; PA%;"   ";PB%;"   ";PO%
310 '-----------------Input Calibration data --------------
320  READ POO  , POSL
322  DATA  -.9305928  , 108.50313
325  READ TMO   , TMSL
327  DATA  -.728981  , 150.4392
330 '
332 FALSE = 0 :TRUE = NOT FALSE
335 XON$=CHR$(17) : XOFF$=CHR$(19)
337 COMFIL$="com1:300,e,7"
340 OPEN COMFIL$ AS #1:PAUSE =FALSE
380 '------------------Flag initialization--------
390 FGBR$="off":FGTS$="off":FGPR$="off":FGSP$="off" :FGD$="on"
400 '------------------Key definitions----------
440 KEY 15,CHR$( 4)+CHR$(70):ON KEY (15) GOSUB  5300:KEY (15) ON
450 ON KEY ( 1) GOSUB  4000:KEY ( 1) ON
460 ON KEY ( 2) GOSUB  4680:KEY ( 2) ON
470 ON KEY ( 3) GOSUB  4800:KEY ( 3) ON
480 ON KEY ( 4) GOSUB  5560:KEY ( 4) ON
490 ON KEY ( 5) GOSUB 58000:KEY ( 5) ON
495 ON KEY ( 6) GOSUB 14000:KEY ( 6) ON
500 ON KEY ( 7) GOSUB 56000:KEY ( 7) ON
505 ON KEY ( 8) GOSUB 55000:KEY ( 8) ON
510 ON KEY ( 9) GOSUB  7000:KEY ( 9) ON
520 ON KEY (10) GOSUB  3420:KEY (10) ON
530 '------------- Initialize ISAAC  -  OUTPUTS   ------
550 OADIN =4095  :OADFF =0  :OOFF = 4   :OSPAN  =16
552 AOCHA%=1    :AOCHB%=2  :AOCHO%=0
560 AOPT$=""    :OSLOPE = OSPAN/OADIN
562 '-------------- Initialize ISAAC -   INPUTS   ------
565 IADIN =4095  :IADFF =0  : IOFF =0   :ISPAN  =1!
570 AICH1% = 1 : AICH2% = 2
572 INOPT$=""   :ISLOPE = ISPAN/IADIN
575 '
585 STAT%=0     :CALL SETSTAT(STAT%)
590 COUNT=225:HZ=75:OPT$=""
600 MP$(1)= "o4132t220cdefgabo5116c":MP$(3)="o5132co4bagfed116c"
601 MP$(2)= "mfmst150116o3eo118g"   :MP$(4)="mfo5t80fcfcfcfc"
610 'PLAY "mbo214ao314ao414ao514a"
650 CLS:SCREEN 0,0,0:WIDTH 80:KEY OFF
660 LOCATE 2,1:
PRINT"------------------------------------------------------------
-----------------";
670 LOCATE 4, 21: PRINT "OPEN LOOP RUN PROGRAM -  NO  PREDICTION ";
680 LOCATE 6, 21: PRINT "For Penicillin_G Continuous Fermentations"
690 LOCATE 8,1:
PRINT"------------------------------------------------------------
----------------";
```

```
700 LOCATE 11,21 :PRINT "Revision 12.0 --- SEPTEMBER 17, 1987 "
710 LOCATE 22,1:PRINT
"------------------------------------------------------------------
-------------";
720 LOCATE 25,1:INPUT "Press <CR> to continue ...",C$:CLS
730 LOCATE 2,1:
PRINT"------------------------------------------------------------
-------------------";
740 LOCATE 5,30: PRINT "SYSTEM INITIALIZATION"
750 LOCATE 8,1:
PRINT"------------------------------------------------------------
--------------------";
760 LOCATE 12,15
765 '
775 VOL=14:GF=15:GP=1.2:GT=19.5:COFFO=1!:PHASE$="   GROWTH PHASE"
776 Y(1) = 0!: Y(2)=.1 :Y(4)=.3287 :Y(5)=.002
777 Y(6)=21.4567:Y(7)=10.23
778 FOR I%=0 TO 7:T(I%)=12.31:NEXT I%
780 '
790 '------------------Sampling interval
795 LOCATE 14,15:INPUT" What's the Initial Biomass (g/L) "; BMASS0
797 XEXP# = BMASS0 : Y(3) =BMASS0
800 LOCATE 17,15:INPUT" Enter the offtime (h) ";OFFTIME
807 LOCATE 20,15:INPUT"Please Enter Sampling Interval (min) ";TSAMPLE%
810 N2%=TSAMPLE%/10:TSH#=TSAMPLE%/60#:TSAMPLE=TSAMPLE%*60
812 LPRINT " ### SAMPLING INTERVAL   "; TSAMPLE%;"  min"
815 PO1=0:PO2=110:TM1=15:TM2=30
820 '
880 CLS:LOCATE 25,71:PRINT TIME$ :LOCATE 9,10
890 '----------------------Printing Parameters
900 'INPUT "Enter PRINTING Interval as SAMPLING Interval Multitude"
;IPRINT%
910 '----------------------Disk output parameters
920 LOCATE 15,10:INPUT"Enter filename for Disk Output(up to 7 letters)"
;OFLE$
930 IF OFLE$ <> "" THEN 960  ELSE LOCATE 18,10
940 PRINT "The default name OUTPUT.DAT will be used";
950  INPUT "Enter <CR> to continue...",A$:OFLE$="OUTPUT.DAT"
960 CLS
970 '
980 OPEN "plot1.dat" FOR APPEND AS 3
1000 '
1010 LOVAL=0:HIVAL=4095:TYPE%=4:PLETE%=1:BACKGROUND%=0
1020 FORMAT%(0)=3:FORMAT%(1)=-1:FORMAT%(2)=0:FORMAT%(3)=0:FORMAT%(4)=0
1030 XLABEL$="":YLABEL$="":COMMENT$="test"
1040 '
1050 BC$(1)="  DIL. RATE          (h-1)"
1060 BC$(2)=" PAA Feed Conc.     (g/L)"
1063 BC$(3)=" CO2   in          (%vol)"
1065 BC$(4)=" CO2  out          (%vol)"
1068 BC$(5)="  O2   in          (%vol)"
```

```
1070 BC$(6)="  O2  out          (%vol)"
1080 BC$(7)=" CO2  produced     (%vol)"
1090 BC$(8)="  O2  consumed     (%vol)"
1110 BC$(9)=" PO2             (% Sat.)"
1120 BC$(10)="Temperature     (Cent. )"
1130 '
1140 BM$(1)="PAA          ( g/L)"
1150 BM$(2)="Glucose      ( g/L)"
1160 BM$(3)="Biomass    (g.dw/L)"
1170 BM$(4)="CO2 in       (%vol)"
1180 BM$(5)="CO2 out      (%vol)"
1190 BM$(6)="O2  in       (%vol)"
1195 BM$(7)="O2  out      (%vol)"
1200 BM$(8)="Pen_G        ( g/L)"
1210 '
1220 CHNG$(0) ="   Max. Feed Concentrations"
1230 CHNG$(1) ="   Controller gain          "
1240 CHNG$(2) ="   ISAAC PARAMET.  (input or output)"
1250 CHNG$(3) ="   *** Alarm Limits *****  "
1260 '
1270 GOSUB 56000
1550 SGR0=2.1:P0=0:AGE0=20:PAA0=.01: MT%=4 :CLS
1570 FGC%=0 :ISML%=1 : TMO=0 :DAY=86400!
1580 '
1590 GOSUB  3440 :'------- Set Points - Initializations -------------
1640 '
1740 '
1760 LOCATE 4,1: PRINT
"------------------------------------------------------------------------
-----------"
1780 LOCATE 8,20: PRINT "FOR HELP ON ACCEPTABLE KEYBOARD INTERRUPTS";
1800 LOCATE 11,35: PRINT "press <F1>";
1820 LOCATE 15,1: PRINT
"------------------------------------------------------------------------
-----------"
1840 LOCATE 19,20 : PRINT "When you Enter <CR> , The Timer STARTS ...";
1860 LOCATE 25,71: PRINT TIME$
1880 LOCATE 25,1: INPUT "Enter <CR> to continue.....",C$
1900
'------------------------------------------------------------------------
1920 '                                          SET TIMER AND TRAP KEYS
"on"
1940
'------------------------------------------------------------------------
1960 ON TIMER (N2%*60) GOSUB 6280 :TIMER ON :TIME1=TIMER:TIME0=TIMER
1970 TIMEC0=TIME0
1990 '
1995 FGK$="on"
2000 GOSUB 6280
2005 IPR%=19 :FGK$="off":FGPR$="on"
2080 M=4:N=6:GOSUB 12000
```

```
2083 IF FGPR$="off" THEN 2092 ELSE IPR%=IPR%+1
2084 IF IPR% <> 20 THEN 2087 ELSE IPR%=0
2085 LPRINT "    Date    Time    Tel   PO2   Temp.  B.exp.
FA   FB   FO "
2087 LPRINT "* "; LEFT$(DATE$,5);" ";LEFT$(TIME$,5);" * ";
2088 RTMEL =OFFTIME + TMEL: LPRINT USING "###.##";RTMEL;
2089 LPRINT USING " ###.## ";RPO;RTM;
2090 LPRINT USING "#.####";D#;:LPRINT USING
"##.###";XEXP#;FA;FB;FO:FGPR$="off"
2092 IF SUMAL < .99 THEN 2135
2095 LOCATE 25,31:FOR I%=1 TO 4:PRINT FGAL%(I%);:NEXT I%:
PRINT "         ALARM";
2097 PLAY MP$(4)
2100 A$=INKEY$:IF A$="" THEN 2097 ELSE SUMAL=0
2105 IF FGER$="on" THEN PLAY MP$(4) :LOCATE 24,7:PRINT ERR ,ERL;
2135 LOCATE 2,4:PRINT ISML%,FGC%
2140 IF TIMER<TIME0 AND FGD$="on" THEN TIMEC0=TIMEC0-DAY :FGD$="off"
2145 IF TIMER>TIME0 THEN FGD$="on"
2152 KEY (1) ON :KEY(3) ON:KEY(4) ON:KEY(5) ON
2156 KEY (7) ON :KEY(8) ON:KEY(9) ON:KEY(10) ON:KEY(15) ON
2160 'J1%=INT((TIMER-TIME1)/(TSAMPLE/N1%))
2162 TMEL=(TIMER-TIMEC0)/3600!
2165 '
2170 LOCATE 2,52 : PRINT "ELAPSED TIME (h) =";: RTMEL = OFFTIME +TMEL
2180 LOCATE 2,72 : PRINT USING "###.###";RTMEL;
2185 LOCATE MT+4,32: PRINT USING  "##.#";RTM;
2195 LOCATE ,38:PRINT USING "###.#";RPO;
2200 '
2375 '
2400 A$=INKEY$
2410 IF A$=""OR A$=CHR$(13) THEN 2430
2415 IF A$="1" THEN 2500           :'Next screen
2420 LOCATE 24,1:  PRINT "Can't understand  ---> ";A$;
2430 LOCATE 25,1: PRINT " -1- --> Change screen ";:LOCATE 25,70
2440 PRINT TIME$;
2450 IF FGBR$="on" GOTO 3160
2460 IF FGSP$="on" THEN GOSUB 3440
2470 IF FGPR$="on" THEN GOSUB 6080
2480 GOTO 2140
2490 '
2500 ' SCREEN 2 ------------ MEASUREMENTS
2510 GOTO 2140
3160
'-------------------------------------------------------------
3180 '                                  Close Output Files and Logout
3200
'-------------------------------------------------------------
3220 '
3240 CLS
3250 CLOSE #1
3260 LOCATE 14,15: PRINT "CLEANING UP..and..SAVING OUTPUT FILES...."
```

```
3280 END
3300 '
3320 '
3340
'-------------------------------------------------------------------
3360 '                              trap <F10> key,   service setpoints
3380
'-------------------------------------------------------------------
3400 '
3420 CLS:FGSP$="on":RETURN
3440 FGK$="on":CLS:LOCATE 25,1
3460 FGSP$="off":PRINT "Servicing <F10> Key...      " TAB(71) TIME$;
3480 LOCATE 3,1:PRINT
"-------------------------------------------------------------------
--------------";
3500 LOCATE 5,25: PRINT "UPDATING  SETPOINTS";
3520 LOCATE 7,1: PRINT
"-------------------------------------------------------------------
---------------";
3540 LOCATE 12,15: INPUT "Enter desired growth rate (1/h) ";RXD1
3560 IF RXD1=0 THEN 3620 ELSE RXD=RXD1
3580 RSD = RXD/YD + MGY +KA0/YP :  SD = K2* RXD / (K1-RXD)
3620 '
3640 '
3700 LOCATE 22,15: INPUT "Enter desired PAA conc. (g/L) ";PAAD1
3720 IF PAAD1=0 AND PHASE% >1 THEN 3780
3740 IF (PAAD1<.001 OR PAAD1>5!) AND PHASE% > 1  THEN CLS:
LOCATE 20,15:                  PRINT"That can't be right...PAA conc.
(g/L) = ";PAAD1:CLS: GOTO 3700
3760 PAAD=PAAD1
3820 '
3840 LPRINT " ### SET POINTS RXD,PAAD : ";RXD;"   ";PAAD
3940 CLS: FGK$="off" :IF FGC%=0 THEN RETURN ELSE RETURN 2080
3960 '
3980
'-------------------------------------------------------------------
4000 '                                    trap help key ... <F1>
4020
'-------------------------------------------------------------------
4040 '
4060 CLS
4080 LOCATE 25,1: PRINT "Servicing <F1> Key...      " TAB(71) TIME$;
4100 'SOUND 880,2!
4120 'SOUND 440,2!
4140 LOCATE 2,1: PRINT
"-------------------------------------------------------------------
--------------"
4160 LOCATE 4,8 :
PRINT ">>>>>  ACCEPATABLE KEYBOARD INTERRUPTS ARE  <<<<<";
4180 LOCATE 6, 15: PRINT "<F1>.....print this message"
4200 LOCATE 7, 15: PRINT "<F2>.....clear the screen"
```

```
4220 LOCATE 8,15: PRINT "<F3>.....change sampling interval";
4240 LOCATE 9,15: PRINT "<F4>.....REQUEST LOGGING ";
4260 LOCATE 10,15: PRINT "<F5>.....GRAPHS  ";
4280 LOCATE 11,15: PRINT "<F6>.....REARRANGE PUMPS ";
4300 LOCATE 12,15: PRINT "<F7>.....change parameters ";
4320 LOCATE 13,15: PRINT "<F8>.....linear regression routine ";
4340 LOCATE 14,15: PRINT "<F9>.....ENTER MEASUREMENTS";
4360 LOCATE 15,15: PRINT "<F10>.....change setpoint";
4380 LOCATE 16,15: PRINT "^BREAK.....finish this run";
4400 TIME2=TIMER
4420 LOCATE 19,15: PRINT
"------------------------------------------------------------------
-------------"
4440 LOCATE 21,15: PRINT "Hit any Key to Continue";
4460 C$=INKEY$
4480 IF C$="" AND TIMER - TIME2 < 10 THEN 4460
4500 CLS
4520 IF C$="" THEN'BEEP
4540 CLS:LOCATE 20,15:PRINT "SORRY TIMEOUT"
4560 RETURN 2080
4580 '
4600
'-----------------------------------------------------------------
4620 '                                  trap <F2>  key,  clear screen
4640
'-----------------------------------------------------------------
4660 '
4680 CLS:RETURN 2080
4700 '
4720
'-----------------------------------------------------------------
4740 '                                  trap <F3>  key, TSAMPLE change
4760
'-----------------------------------------------------------------
4780 '
4800 FGK$="on"
4810 CLS:LOCATE 4,1:PRINT
"------------------------------------------------------------------
---";
4820 LOCATE 7,26: PRINT "CHANGE SAMPLING INTERVAL"
4840 LOCATE 10,1: PRINT
"-----------------------------------------------------------------"
4860 LOCATE 25,1: PRINT "Servicing <F3> key...      " TAB(71) TIME$;
4900 LOCATE 15,15: INPUT "Please Enter NEW Sampling Interval (min) "
;TSAMPL1%
4920 IF TSAMPL1%=0 THEN CLS: RETURN 2080
4940 IF TSAMPL1%< 10 THEN CLS :LOCATE 10,10:
PRINT "A sampling Interval of ";          TSAMPL1%;" is too
small...Try again...":GOTO 4810
4950 LPRINT " ### Sampling Interval  : ";TSAMPLE%
4980 FGTS$="on":FGK$="off":RETURN 2080
```

```
5220 '
5240
'-------------------------------------------------------------------
5260 '                                        trap ^BREAK key,  finish run
5280
'-------------------------------------------------------------------
5300 '
5320 CLS
5340 LOCATE 25,1: PRINT "Servicing  ^BREAK....    " TAB(71) TIME$;
5360 LOCATE 10,15: PRINT "Do you really want to FINISH this run ??"
5380 LOCATE 12,15: INPUT "Please answer (y/n)";C$
5400 IF C$<>"y" AND C$<>"Y" THEN CLS:LOCATE 20,10:
 PRINT "STOP PLAYING with the BREAK KEY....":RETURN 2080
5420 CLS:FGBR$="on"
5440 RETURN
5460 '
5480
'-------------------------------------------------------------------
5500 '                                    trap <F4> key,  request logging
5520
'-------------------------------------------------------------------
5560 '
5570 CLS:LOCATE 4,1: PRINT
"-------------------------------------------------------------------
---";
5580 LOCATE 7,10:PRINT "Make sure the PRINTER is ON LINE and
the PAUSE key OFF";
5600 LOCATE 10,10:PRINT "Else the Program will ABORT and all
DATA will be lost";
5620 LOCATE 13,1: PRINT
"-------------------------------------------------------------------
------------";
5640 LOCATE 21,15: PRINT "Hit any Key AFTER you have checked...";
5660 TIME2=TIMER
5680 C$=INKEY$
5700 IF C$="" AND TIMER - TIME2 < 15 THEN 5680
5720 CLS
5740 IF C$="" THEN CLS:LOCATE 20,15:PRINT"SORRY TIMEOUT...        ";
5780 LOCATE 25,1: PRINT "Servicing <F4> PRINTER...     " TAB(71) TIME$;
5800 'LPRINT " ":LPRINT " ":LPRINT " "
5820 'LPRINT
"-------------------------------------------------------------------
----------"
5840 'LPRINT " "
5860 'LPRINT "STATUS OF THE SYSTEM at TIME = ";TIME$
5880 'LPRINT " "
5900 'LPRINT
"-------------------------------------------------------------------
-----------"
5920 'LPRINT " "
5940 'LPRINT "Elapsed Time = ";(TIMER-TIME0)/3600;" (h)"
```

```
5960 'LPRINT "Sampling Interval = ";TSAMPLE%;" (MIN)"
5980 'LPRINT "Current Liquid Volume = ";VOL;" (L)"
6000 'LPRINT "Current Biomass = ";WT(0,CN1%) ;" (g/L)"
6020 'LPRINT "***"
6040 'LPRINT " ":'LPRINT " ":'LPRINT " "
6060  RETURN 2080
6080 '
6280
'*********************************************************************
6300 '              Servicing     Isaac
6320 '-------------------------------------------------------------
6340 KEY(1) STOP: KEY(3) STOP:KEY(4) STOP:KEY(5) STOP
6341 KEY(7) STOP: KEY(8) STOP:KEY(9) STOP:KEY(10) STOP
6342 FGC%=FGC%+1
6345 '
6348 IF ISML%=1 AND FGC%=1 THEN PLAY MP$(2):GOTO 6600
6350 IF FGC%=11 THEN 6357 ELSE PLAY MP$(1):GOTO 6806
6355 '
6357 TIME1=TIMER:PLAY MP$(2)
6360 FGC%=1 :TMO=TMO+TSH# :ISML%=ISML%+1
6380 '
6420 IF FGTS$<>"on" THEN 6530
6440 TSAMPLE%=TSAMPL1%:N2%=TSAMPLE%/10
6460 ON TIMER (N2%*60) GOSUB 6280:TIMER ON :TIME1=TIMER
6480 FGTS$="off":TSH#=TSAMPLE%/60# :TSAMPLE=TSAMPLE%*60
6530 '
6600 FGPR$="on"
6602 D#=RSD*XEXP#/SA
6610 XEXP#=XEXP#*EXP(RXD*TSH#)
6620 IF PAAD<.0005 THEN SFA=0 :GOTO 6650
6622 BRPC= BETA*(KA0+KCP*(PAAD-Y(1)))*XEXP#/D#
6630 SFA=PAAD +BRPC                    :' -----PAA CONCENTRATION
6640 '
6650 '
6660 '
6670 GOSUB 9720                        :' ----Check constraints
6680 '
6690 FAN%=CINT(FA/FAMIN*10)
6700 FBN%=CINT(FB/FBMIN*10)
6710 FON%=CINT(FO/FOMIN*10)
6720 '
6730 IF FAN%>10 THEN  FAF=FA ELSE FAF=FAMIN
6735 IF FBN%>10 THEN  FBF=FB ELSE FBF=FBMIN
6740 IF FON%>10 THEN  FOF=FO ELSE FOF=FOMIN
6750 '
6760 IA= IAMIN + SLA*(FAF-FAMIN)
6770 IB= IBMIN + SLB*(FBF-FBMIN)
6780 IO= IOMIN + SLO*(FOF-FOMIN)
6790 '
6800  VALA%=(IA-OOFF) / OSLOPE  +OADFF    :
' Calculate binary output values
```

```
6802  VALB%=(IB-OOFF) / OSLOPE  +OADFF
6804  VALO%=(IO-OOFF) / OSLOPE  +OADFF
6806  IF FGC%>FAN% THEN VALA%=0        :' Apply Intermittent Flow
6808  IF FGC%>FBN% THEN VALB%=0
6810  IF FGC%>FON% THEN VALO%=0
6811 CLS:TRON
6812 ' ------------- Output statements
----------------------------------------
6814 PRINT AOCHA%,VALA%,AOPT$,AOCHB%,VALB%,AOCHO%,VALO%
6815 GOSUB 15000
6816  CALL AOUS (AOCHA%,VALA%,AOPT$) :' Call ISAAC'S output routine
6818  CALL AOUS (AOCHB%,VALB%,AOPT$)
6820  CALL AOUS (AOCHO%,VALO%,AOPT$)
6822 '
6830 IF FGC% > 1 THEN 6927
6840 '
6860 DR = FI/VOL
6900 OPEN "RECOVER.DAT" FOR OUTPUT AS 2    :' Write in recovery file
6905 PRINT #2,TSAMPLE%,OFLE$
6910 PRINT #2,VOL,SA,SP
6915 PRINT #2,RXD,PAAD,KCP
6920 PRINT #2,XEXP#,Y(1)
6922 PRINT #2 ,PA%,PB%,PO%
6925 CLOSE #2
6926 '
6927  CALL AINM(AICH1%,COUNT,HZ,RAW1%(0),OPT$)
6933  CALL AINM (AICH2%, COUNT,HZ,RAW2%(0),OPT$)
6934 SUM1=0:SUM2=0  :FOR IR%=1 TO COUNT
6935 SUM1 =SUM1 +RAW1%(IR%-1) :SUM2=SUM2 +RAW2%(IR%-1)
6936 NEXT IR%
6937 VTM%=SUM1/COUNT : VPO%=SUM2/COUNT
6938 '
6942  RPO =  PO0 + ( VPO% - IADFF ) *ISLOPE *POSL
6944  RTM =  TM0 + ( VTM% - IADFF ) *ISLOPE *TMSL :RTM=RTM - 1.5
6945 '
6947 FGAL%(1)=0:FGAL%(2)=0:FGAL%(3)=0:FGAL%(4)=0
6949 IF RPO<PO1 THEN FGAL%(1)=1 ELSE IF RPO>PO2 THEN FGAL%(2)=1
6951 IF RTM<TM1 THEN FGAL%(3)=1 ELSE IF RTM>TM2 THEN FGAL%(4)=1
6952 SUMAL%=0:FOR IR%=1 TO 4:SUMAL=SUMAL+FGAL%(IR%):NEXT IR%
6954 IF FGC% >1 THEN 6975
6956 '
6960 '
6962 '
6968 CLOSE #3                              :' Write in plot - file
6970 OPEN "plot1.dat" FOR APPEND AS 3
6973 PRINT #3,D#;SFA;Y(4);Y(5);Y(6);Y(7);Y(5)-Y(4);Y(7)-Y(6);RPO;RTM
6974 LPRINT " ON LINE : PO2 ,TEMP  :";RPO;"   ";RTM
6975 '
6980 CLOSE #3 : OPEN OFLE$ FOR APPEND AS 3 :'sAVE DATA
6982 PRINT #3,DR;" ";FA;" ";FB;" ";FO;" ";FAN%;" ";FBN%;" ";FON%;"
";SA;" ";SP;" ";SFA;" ";RPO;" ";RTM
```

```
6984 CLOSE #3
6990 TROFF:PLAY MP$(3):CLS
6991 IF FGINT$="on" OR FGK$="on" THEN RETURN ELSE RETURN 2080
6992 '
7000 ' -------------- enter measuremants ----------------------------
7010 FGK$="on":CLS:KEY OFF
7017 LOCATE 3,20:PRINT "to ENTER a MEASUREMENT  OF ......"
7020 FOR I%=1 TO 8
7030 LOCATE 4+I%,20:PRINT BM$(I%);"     PRESS    ";I%;" <CR>"
7040 NEXT I%
7050 LOCATE 15,10: INPUT "Well....";IM%
7060 IF IM%<1 OR IM%>8 THEN 7312
7070 LOCATE 18,10 : PRINT "Old Value : ";BM$(IM%);" =  ";Y(IM%)
7090 LOCATE 19,12 :PRINT "for the sample taken at ";T(IM%)
7092 LOCATE 23,10 :PRINT "CHECK YOUR UNITS - Press any key when ready"
7094 A$=INKEY$:IF A$="" THEN 7094
7100 LOCATE 23,10 :PRINT "                                                "
7120 LOCATE 20,30:INPUT "**********  NEW value   =";Y(IM%)
7140 LOCATE 21,30:INPUT "SAmple was taken at ";T(IM%)
7145 LPRINT "***  ";LEFT$(DATE$,5);" * ";LEFT$(TIME$,5);" ** ";
7146 LPRINT USING "###.##";TMEL;
7148 LPRINT " * ";BM$(IM%);"  ";Y(IM%);"  AT : ";T(IM%);" hrs"
7160 IF IM%<>3 THEN 7220 ELSE CLS:LOCATE 10,10
7163 PRINT "RE : Biomass measurement":LOCATE 12,10
7165 PRINT "You can use this value to CORRECT the DILUTION RATE (D)"
7167 LOCATE 14,5 :PRINT "DO YOU TRUST this measurement (y/n)?"
7170 A$=INKEY$:IF A$="" THEN 7170 ELSE IF A$="n" OR A$="N" THEN 7200
7180 LOCATE 16,10:PRINT "do you want to correct D ,based on it(y/n)?"
7185 A$=INKEY$ :IF A$="" THEN 7185 ELSE IF A$="n" OR A$="N" THEN 7195
7190 LOCATE 18,1:PRINT "O.K. MASTER!!":XEXP#=Y(3)*EXP((TMEL-T(3))*RXD)
:GOTO 7205
7195 LOCATE 18,1:PRINT "MAY BE ANOTHER TIME THEN!!!":GOTO 7205
7200 LOCATE 18,1:PRINT "I HOPE YOU'LL come up with better
measurements SOON !!!"
7205 LOCATE 23,10:PRINT "Press <CR> to continue"
7210 A$=INKEY$:IF A$="" THEN 7210 ELSE CLS :GOTO 7017
7220 LOCATE 23,10:PRINT "Another  Entrance  (y/n) ?"
7225 C$=INKEY$:IF C$="" THEN 7225
7230 IF C$="n" OR C$="N" THEN 7312
7235 LOCATE 18,10:PRINT "                                                "
7240 LOCATE 19,12:PRINT "                                                "
7245 LOCATE 20,30:PRINT "                                                "
7250 LOCATE 21,30:PRINT "                                                "
7300 LOCATE 22,1 :PRINT "                                                "
7305 LOCATE 15,17:PRINT "    "
7310 GOTO 7050
7312 LOCATE 23,5 :PRINT "Do you want to update air flow
measurements (y/n)?"
7315 A$=INKEY$:IF A$="" THEN 7315 ELSE IF A$="n" OR A$="N" THEN 7375
7320 CLS:LOCATE 5,1 :INPUT "AIR FLOW RATE (L/min) ";GF
7325 LOCATE 7,1:INPUT "Inlet air pressure (baru) ";GP
```

```
7330 LOCATE 10,1 :INPUT "Inlet air temperature ( Cent.) ";GT
7340 LPRINT "Inlet air flow ,press. , temp.";GF;"  ";GP;"  ";GT
7350 LOCATE 15,10:INPUT "Print <CR> to continue";A$
7375 CLS:FGK$="off":RETURN 2080
7380 '
9720 '----------- CONSTRAINTS ON MANIPULATED VARIABLES --------------
9760 IF SFA > SP THEN SFA = SP
9764 IF SFA < 0! THEN SFA = 0!
9770 '
9780 FAMMIN=FAMIN/10! :FBMMIN=FBMIN/10! :FOMMIN=FOMIN/10!
9790 FI = D#*VOL*(1000!/60!):FB= FI*SFA/SP :FA= FI*(1!-SFA/SP)
9795 IF FAMMIN<FOMMIN  THEN FAMMIN=FOMMIN
9797 IF FBMMIN<FOMMIN THEN FBMMIN=FOMMIN
9800 IF FA<FAMMIN THEN FA=FAMMIN
9820 IF FA>FAMAX  THEN FA=FAMAX
9835 '
9840 IF FB<FBMMIN THEN FB=FBMMIN
9842 IF PAAD<.0001 THEN FB=0
9845 IF FB>FBMAX  THEN FB=FBMAX
9850 '
9855 ' IF (FA+FB)<FOMMIN  THEN FO=0!  :goto 9900
9857 '
9860 FI = FA +FB :SFA = FB/FI*SP  :FO = FI*COFFO
9870 IF FO < FOMMIN THEN FO = FOMMIN
9880 IF FO > FOMAX  THEN FO = FOMAX
9900 RETURN
9950 '
10990 '---------- Error handling subroutine
-----------------------------------
11000 '
11010 IF (ERR=24)OR (ERR=25 ) OR (ERR=26) THEN FGER$="on":RESUME NEXT
11015 IF 5460<=ERL AND 6080>ERL THEN FGER$="on":RESUME 2080
11020 IF 7000<=ERL AND 8000>ERL THEN FGER$="on":RESUME 2080
11030 IF 55000!<= ERL AND 56000!>ERL THEN FGER$="on":RESUME 2080
11040 IF 56000!<=ERL AND 57200!>ERL THEN FGER$="on" :RESUME 2080
11050 IF ERL>57200 THEN FGER$="on" :RESUME 2080
11060 PRINT ERR,ERL:RESUME NEXT
12000 '
12005 ' -------------dISPLAY SUBROUTINE -----------------------
12010 KEY OFF:CLS
12030 LOCATE 1,1:FOR I=1 TO 80:PRINT CHR$(196);:NEXT I
12050 FOR I=2 TO 17 :LOCATE I,1:PRINT CHR$(179)
12070 LOCATE I,80:PRINT CHR$(179):NEXT I
12090 LOCATE 3,1:FOR I=1 TO 80 :PRINT CHR$(196);:NEXT I
12110 LOCATE 2,19:PRINT "         "PHASE$
12130 LOCATE M,N-2:FOR I=1 TO 28:PRINT CHR$(205);
:NEXT I:PRINT CHR$(187)
12150 FOR I=1 TO 1:LOCATE M+I,N+26: PRINT CHR$(186):NEXT I
12190 LOCATE M,35+N:PRINT CHR$(201);:FOR I=1 TO 36
:PRINT CHR$(205);:NEXT I
12210 FOR I=1 TO 1:LOCATE M+I,N+35:PRINT CHR$(186):NEXT I
```

```
12230 LOCATE M+1+1,N+35:PRINT CHR$(25)
12250 MT=M+1+1 :FOR I=1 TO 8 :LOCATE  MT+(I-1),27:PRINT CHR$(221)
12270 LOCATE  MT+(I-1),27+20-1:PRINT CHR$(222):NEXT I
12290 FOR I=1 TO 18 :LOCATE MT+1,27+I:PRINT CHR$(247):NEXT I
12310 LOCATE MT+8-1,27+20:FOR I=1 TO 24:PRINT CHR$(205);:NEXT I
12330 PRINT CHR$(187):LOCATE MT+8+1,27+20+24:PRINT CHR$(25)
12350 FOR I=1 TO 1:LOCATE MT+8-1+I,27+20+24:PRINT CHR$(186):NEXT I
12370 LOCATE MT+8,27:FOR I=1 TO 20:PRINT CHR$(223);:NEXT I
12390 LOCATE MT+8+1,2:FOR I=1 TO 34:PRINT CHR$(205);:NEXT I
12410 PRINT CHR$(188):LOCATE MT+8-1,36:PRINT CHR$(186)
12430 LOCATE MT+8-2,36:PRINT CHR$(24)
12450 LOCATE M,N:PRINT  "Sol. A : GL (       g/L)"
12470 LOCATE M,44+N:PRINT "Sol. B: GL + PAA (       g/L)"
12490 LOCATE M+3,N :PRINT "Flow =        ml/min "
12510 LOCATE M+4,N :PRINT "% of total =      "
12530 LOCATE M+1,45+N:PRINT "          ml/min ,       %"
12550 LOCATE MT+6,47:PRINT "          ml/min ,      %"
12570 LOCATE MT+8-1,2:PRINT "  Air flow:       L/min"
12590 LOCATE MT+8,2:PRINT "        bar ,     ";CHR$(248);"C"
12670 LOCATE MT+2,31: PRINT "   T     PO  "
12770 LOCATE 17,1:FOR I=1 TO 80:PRINT CHR$(176);:NEXT I
12775 LOCATE 17,3:PRINT "PAA(G/L)";:LOCATE ,19:PRINT "PEN_G";
12780 LOCATE ,35:PRINT "GLUCOSE";:LOCATE ,50:PRINT "Biomass";
12785 LOCATE ,66:PRINT "CO2% out"
12810 LOCATE 18,1:FOR I=1 TO 80:PRINT CHR$(196);:NEXT I
12850 LOCATE 20,1:FOR I=1 TO 80:PRINT CHR$(220);:NEXT I
12860 LOCATE 18,26:PRINT " LATEST..       .. DATA "
12865 LOCATE 20,26:PRINT " SAMPLE..       .. TIME "
12890 LOCATE 25,31 : PRINT "Sampling Interval ";" 20"; " min";
13000 '
13010 LOCATE M,N+14:PRINT USING "###.#" ;SA
13020 LOCATE M,63+N:PRINT USING "##.##";SP
13030 LOCATE M+3,N+6:PRINT USING "##.####"; FA
13050 LOCATE M+1,46+N:PRINT USING "##.####";FB
13055 IF (FA+FB)<1E-08 THEN FTOT =1 ELSE FTOT =FA+FB
13057 LOCATE M+4,N+13:PRINT USING "###.#"; FA/FTOT*100!
13060 LOCATE M+1,62+N:PRINT USING "###.#";FB/FTOT*100!
13070 LOCATE MT+6,48:PRINT USING "##.####";FO
13080 LOCATE MT+6,66:PRINT USING "###.#";COFFO*100!
13090 LOCATE MT+7,14:PRINT USING "##.#";GF
13100 LOCATE MT+8,6:PRINT USING "#.#";GP
13110 LOCATE MT+8,16:PRINT USING "##.#";GT
13150 LOCATE 19,1:PRINT USING " ##.####       " ;
Y(1),Y(8),Y(2),Y(3),Y(5)
13160 LOCATE 21,1:PRINT USING "  ###.#        " ;
T(1),T(8),T(2),T(3),T(5)
13170 RETURN
13200 '
14000 ' --------   REARRANGE THE PUMPS   -------------------------
14010 CLS :FGK$="on":LOCATE 10,10 :PRINT "PRESENT ARRANGEMENT :"
14020 LOCATE 12,10      :PRINT "Glucose only          Pump ";PA%
```

```
14040 LOCATE 14,10       :PRINT "Glucose + PAA          Pump ";PB%
14050 LOCATE 16,10       :PRINT "OUTLET                  ''  ";PO%
14060 LOCATE 18,10       :INPUT "NEW ARRANGEMENT (pa,pb,po) ";
PA%,PB%,PO%
14070 IAMIN = IMIN(PA%) :OFFA =OFFP(PA%) :SLA =SLP(PA%)
14080 IBMIN = IMIN(PB%) :OFFB =OFFP(PB%) :SLB =SLP(PB%)
14090 IOMIN = IMIN(PO%) :OFFO =OFFP(PO%) :SLO =SLP(PO%)
14095 FAMAX = (20-OFFA)/SLA: FBMAX = (20!-OFFB)/SLB:
FOMAX= (20!-OFFO)/SLO
14100 FAMIN=(IAMIN-OFFA)/SLA :FBMIN=(IBMIN-OFFB)/SLB:
FOMIN=(IOMIN-OFFO)/SLO
14110 LOCATE 20,10 :INPUT "O.K.! Now Press <CR> to continue",A$
14115 LPRINT " ### PUMP ARRANGEMENT : A , B , O : ";PA%;"   ";
PB%;"   ";PO%
14120 CLS:FGK$="off":RETURN 2080
14130 '
15000 IF VALA%>OADIN THEN VALA%=OADIN
15005 IF VALA%<OADFF THEN VALA%=OADFF
15010 IF VALB%<OADFF THEN VALB%=OADFF
15015 IF VALB%>OADIN THEN VALB%=OADIN
15020 IF VALO%<OADFF THEN VALO%=OADFF
15025 IF VALO%>OADIN THEN VALO%=OADIN
15030 '
15035 RETURN
55000 ' ---------- LINEAR REGRESSION ROUTINE ---------------------
55005 ' INPUTS :  NUMBER OF POINTS , X VALUES , Y VALUES
55010 ' OUTPUT : Parameters a ,b SUCH THAT sum(Y-a*X -b)**2 =MINIMUM
55020 '
55025 CLS :FGK$="on": LOCATE 3,1
55027 PRINT "****** FIRST MAKE SURE THE POINTS FALL CLOSE TO A
STRAIGHT LINE ****"
55030 LOCATE  5,1 :INPUT " How many points do you have ( >= 3 )";NREG%
55035 IF NREG%<2 THEN 55130 ELSE SUMX=0:SUMY=0:SUMX2=0:SUMXY=0:SUMY2=0
55036 LOCATE 7,5:PRINT "  X's                Y's"
55040 FOR IRR% = 1  TO NREG% :LOCATE IRR%+7,5 :INPUT XR
55045 LOCATE IRR%+7,23 :INPUT YR
55050 SUMX=SUMX + XR : SUMY = SUMY + YR : SUMX2 =SUMX2 + XR*XR
55060 SUMXY =SUMXY +XR*YR :SUMY2 =SUMY2 + YR*YR : NEXT IRR%
55070 DETA=SUMXY * NREG% -SUMY * SUMX
55080 DETB=SUMX2 * SUMY  -SUMX * SUMXY
55090 DET =SUMX2 * NREG% -SUMX * SUMX
55095 AREG = DETA/DET : BREG = DETB/DET :AVERY= SUMY/NREG%:CLS
55096 SUMSQ = SUMY2 + AREG*AREG*SUMX2 + NREG%*BREG*BREG
55097 SUMSQ=SUMSQ  -2 *AREG *SUMXY -2 *BREG *SUMY +2 *AREG *BREG *SUMX
55098 PERIN = SQR(SUMSQ)/AVERY
55100 LOCATE 10,10 :PRINT "the equation of the closest straight line
is ..."
55110 LOCATE 13,10 : PRINT " Y =  " ;AREG;" * X  +  ";BREG
55115 LOCATE 15,5  :PRINT  "Performance ind = sqr(sqsum)/avery = ";
ABS(PERIN)
55120 LOCATE 18,18 :INPUT "Value for X ";XR
```

```
55155 YR = AREG*XR +BREG :LOCATE 19,1:PRINT "Y value = ";YR
55156 LOCATE 20,10:PRINT "Do you have another value of X (y/n) ?"
55157 A$=INKEY$:IF A$="" THEN 55157 ELSE IF A$="n" OR A$="N" THEN 55165
55158 LOCATE 19,11:PRINT "                ":LOCATE 18,30:PRINT"     "
55160 LOCATE 20,10:PRINT "                                          ":
GOTO 55120
55165 CLS :FGK$="off": RETURN 2080
55170 '
56000 FGK$="on":' ---------- CHANGE PARAMETERS --------------------
56100 CLS:LOCATE 5,1
56110 PRINT "       TO   UPDATE                                      ENTER"
56115 PRINT:PRINT:PRINT CHNG$(0);"                                    1"
56120 PRINT:PRINT CHNG$(1);"                                  2"
56125 PRINT:PRINT CHNG$(2);"                          3"
56130 PRINT:PRINT CHNG$(3);"                                  4"
56135 LOCATE 17,1:PRINT "ENTER 0 to return to display...."
56140 LOCATE 18,45:INPUT "Well .........";CP%
56145 IF CP%<0 OR CP%>4 THEN 56100 ELSE IF CP%=0 THEN 56560
56146 CLS :LOCATE 25,10 :PRINT "CHANGING ......";CHNG$(CP%-1);
56150 ON CP% GOTO 56200,56300,56400,56500
56200 '
56205 LOCATE 5,1 :PRINT "OLD SUGAR MAX. FEED CONCENTR.  ";SA:SA0=SA
56210 LOCATE 7,1 :PRINT "OLD  PAA   '    '       '      ";SP:SP0=SP
56215 LOCATE 5,40 :INPUT "new value";SA
56220 LOCATE 7,41 :INPUT "NEW VALUE";SP
56225 IF SA<.0001 THEN SA=SA0
56230 IF SP<1E-09 THEN SP=SP0
56240 LPRINT " ### PARAMETERS SA ,SP  : ";SA;"  ";SP: GOTO 56100
56300 LOCATE 5,1 :PRINT "old PAA controller gain = ";KCP:KCP0=KCP
56305 LOCATE 5,40:INPUT "new value ";KCP
56310 IF KCP<.000001 THEN KCP=KCP0
56320 LPRINT " ### PARAMETERS KCP :      ";KCP:GOTO 56100
56400 LOCATE 10,10:INPUT"input parameters (1) or output (2)";CP%
56402 IF CP%<1 OR CP%>2 THEN 56400 ELSE ON CP% GOTO 56409,56452
56409 CLS:LOCATE 5,1 :PRINT "count = ";COUNT;" hz = ";HZ:
VAL0=COUNT:VSL=HZ
56410 LOCATE 7,1 :INPUT "new values ";COUNT,HZ
56412 IF COUNT<.1 AND HZ<.1 THEN COUNT=VAL0:HZ=VSL
56413 LPRINT " ### PARAMETERS COUNT , HZ :  ";COUNT;"    ";HZ
56415 LOCATE 8,1 :PRINT "OTHER CHANGES (y/n) "
56420 A$=INKEY$:IF A$="" THEN 56420 ELSE IF A$="n" OR A$="N" THEN 56100
56440 CLS:LOCATE 10,1 :PRINT"PO2 input parameters  "
PO0;POSL:VAL0=PO0:VSL=POSL
56442 LOCATE 12,1 :INPUT "New values for offset,slope";PO0,POSL
56444 IF ABS(PO0)<.000001 AND POSL<.000001 THEN PO0=VAL0:POSL=VSL
56445 LOCATE 15,1 :PRINT"Temp. input parameters   ";TM0,TMSL:
VAL0=TM0:VSL=TMSL
56447 LOCATE 17,1 :INPUT  "New values for zero,offset";TM0,TMSL
56448 IF ABS(TM0)<.000001 AND TMSL<.001 THEN TM0=VAL0:TMSL=VSL
56449 LPRINT " ### PARAMETERS PO0 ,PSL ,  TM0 , TMSL :";PO0;"  ";PSL;"
";TM0;"  ";TMSL
```

```
56450 LOCATE 19,1:PRINT "CHANGES IN PUMP PARAMETERS ? (y/n)"
56451 A$=INKEY$:IF A$="" THEN 56451 ELSE IF A$="n" OR A$="N" THEN 56100
56452 CLS :LOCATE 5,1:PRINT "PUMP A :";"offa = ";OFFA;" mL/min ,
 Slope = ";SLA
56455 VAL0=OFFA:VSL=SLA:LOCATE 7,1 :INPUT"NEW values";OFFA,SLA
56457 IF OFFA< 4 THEN OFFA=VAL0:SLA=VSL
56460 LOCATE 10,1 :PRINT    "PUMP B :";"OFFB = ";OFFB;" ml/min,
 Slope = ";SLB
56462 VAL0=OFFB:VSL=SLB:LOCATE 12,1:INPUT "NEW values";OFFB,SLB
56464 IF OFFB< 4 THEN OFFB=VAL0:SLB=VSL
56466 LOCATE 15,1 :PRINT    "PUMP C :";"OFFO = ";OFFO;" ml/min,
Slope = ";SLO
56468 VAL0=OFFO:VSL=SLO:LOCATE 17,1:INPUT "NEW values";OFFO,SLO
56470 IF OFFO < 4 THEN OFFO=VAL0:SLO=VSL
56480 LPRINT " ### PARAMETERS ,PUMPS : ";OFFA;"  ";SLA;"   ";OFFB;
"  ";SLB;"   ";OFFO ;"  ";SLO:GOTO 56100
56500 '
56510 LOCATE 5,1 :PRINT "OFF LINE MEASUREMENT ALARMS (y/n)"
56513 A$=INKEY$:IF A$="" THEN 56510 ELSE IF A$="n" OR A$="N" THEN 56530
56516 LOCATE 7,1 :INPUT "PAA  conc.  ALARMS ...Low , High";AL1,AL2
56518 IF AL2>.1 THEN PAA1=AL1:PAA2=AL2
56520 LOCATE 9,1 :INPUT "Glucose conc.                    ";AL1,AL2
56522 IF AL1 >.1 OR AL2>0 THEN GL1=AL1:GL2=AL2
56524 LPRINT " ### PARAMETERS OFF-ALARMS : ";PAA1;" ";PAA2;"  ";GL1;
" ";GL2
56526 LOCATE 12,1 :PRINT "CONTINUE TO ON_LINE ALARMS (y/n)"
56527 A$=INKEY$:IF A$="" THEN 56527 ELSE IF A$="n" OR A$="N" THEN 56560
56530 CLS:LOCATE 3,15 :PRINT "ON LINE MEASUREMENTS ALARMS"
56536 LOCATE 7,1 :INPUT "PO ALARMS              ";AL1,AL2
56539 IF AL1>15 AND AL2>15 THEN PO1=AL1:PO2 = AL2
56542 LOCATE 9,1 :INPUT "Temp. ALARMS           ";AL1,AL2
56545 IF AL1>15 AND AL2 >10 THEN TM1= AL1 :TM2=AL2
56550 LPRINT " ### PARAMETERS ON -ALARMS :  ";PO1;" ";PO2;"   ";TM1;"
";TM2:GOTO 56100
56560 CLS:LOCATE 5,1:PRINT "Change phase (y/n) ?"
56570 A$=INKEY$:IF A$="" THEN 56570 ELSE IF A$="n" OR A$="N" THEN 56800
56575 LOCATE 7,1 :PRINT "Growth (1) or production (2) ?"
56580 A$=INKEY$ :IF A$="" THEN 56580 ELSE IF A$<>"1" AND A$<>"2"
THEN 56560
56590 IF A$="1" THEN PHASE$=" GROWTH PHASE ":PHASE%=1:GOTO 56610
56600 PHASE$=" PRODUCTION PHASE ":PHASE%=2
56610 LOCATE 12,1 :INPUT "O.K !  Now  Press <CR> to return ",A$
56800 CLS:FGK$="off":IF ISML%<1  THEN RETURN ELSE RETURN 2080
57200 '       -------------- G R A P H I C S  --------------
57500 '
58000 FGK$="on" :CLS :LOCATE 5,1
58003 PRINT "press            1       for   ";BC$(1)
58006 FOR IP%=2 TO 11:
58009 PRINT "                ";  IP% ;"             ";BC$(IP%)
58012 NEXT IP%
58045 PRINT "                 0      TO EXIT"
```

```
58048 PRINT :INPUT "Well....";PLOTIN%:IPL%=0:IPL1%=0
58051 IF PLOTIN%>0 THEN  58057
58054 FGK$="off":CLS:RETURN 2080
58057 CLOSE #3: OPEN "plot1.dat" FOR INPUT AS 3
58060 INPUT "Start from interval No :";IPST%:INPUT "Stop at No";IPEN%
58061 IF IPEN%<IPST% THEN BEEP: GOTO 58060
58063 IF IPST%=0 THEN IPST%=1
58064 IF IPEN%=0 THEN IPEN%=ISML%
58066 IPL%=IPL%+1:IF EOF(3) THEN 58093
58069 IF IPL%<IPST% OR IPL%>IPEN% THEN IPL1%=0 ELSE IPL1%=IPL1% +1
58072 FOR IP%=1 TO 10
58075 IF IP%=PLOTIN% THEN INPUT #3,PLV(IPL%):GOTO 58081
58078 INPUT #3,NL
58081 NEXT IP%
58084 'IF PLOTIN%=11 THEN INPUT #3,PLV(IPL%) ELSE INPUT #3,NL
58090 GOTO 58066
58093 CLOSE #3: TITLE$=BC$(PLOTIN%)
58096 MINPL=PLV(1):MAXPL=PLV(1)
58099 FOR IP%=1 TO 1+(IPEN%-IPST%)
58102 IF PLV(IP%) <MINPL THEN MINPL=PLV(IP%)
58105 IF PLV(IP%) >MAXPL THEN MAXPL=PLV(IP%)
58108 NEXT IP%
58111 PRINT "RANGE of";BC$(PLOTIN%),MINPL;"  to  " ;MAXPL
58114 INPUT "highest value";HH:IF HH<0 THEN GOTO 58054
58117 INPUT "lowest value  ";LL
58120 IF HH=0 THEN HH=MAXPL
58123 IF LL=0 THEN LL=MINPL
58126 IF ABS((HH-LL))<=.001*LL THEN CLS:GOTO 58114
58129 GOSUB 58153
58132 FOR II%= IPST% TO IPEN%
58135 IF PLV(II%)<LL OR PLV(II%) >HH  THEN 58138 ELSE 58141
58138 LOCATE 23,1:PRINT "data out of range":GOTO 58147
58141 VALUE=(PLV(II%)-LL)/(HH-LL)*4095
58144 GOSUB 58471
58147 NEXT II%
58150 IF INKEY$="" THEN 58150 ELSE IF INKEY$="a" OR INKEY$="A"
THEN 58111
58151 CLS: SCREEN 0,0,0 :WIDTH 80 :LOCATE 5,1:GOTO 58003
58153 ' name       INITGRAPH
58156 BASE.GRF%           = 0
58159 I.GRF%              = 0
58162 INPTR.GRF%          = 0
58165 J.GRF%              = 0
58168 NWAVES.GRF%         = 1
58171 XVALUE.GRF          = 0
58174 YVALUE.GRF          = 0
58177 SCREEN 1 : VIEW : WINDOW
58180 KEY OFF
58183 COLOR BACKGROUND%,PLETE%
58186 CLS
58189 FOR I.GRF% = 0 TO 3
```

```
58192    FORMAT.GRF%(I.GRF%) = FORMAT%(I.GRF%)
58195 NEXT I.GRF%
58198 WHILE (NWAVES.GRF% < 4) AND FORMAT.GRF%(NWAVES.GRF%) <> -1
58201    NWAVES.GRF% = NWAVES.GRF% + 1
58204 WEND
58207 ON (TYPE% + 1) GOSUB 58267,58267,58267,58312,58312,58312
58210 IF LEN(TITLE$) > 30 THEN TITLE$ = MID$(TITLE$,1,30)
58213 LOCATE 1,(20 - LEN(TITLE$) \ 2)
58216 PRINT TITLE$;
58219 IF LEN(XLABEL$) > 37 THEN XLABEL$ = MID$(XLABEL$,1,37)
58222 LOCATE 21,(21-LEN(XLABEL$) \ 2)
58225 PRINT XLABEL$;
58228 IF LEN(YLABEL$) > 18 THEN YLABEL$ = MID$(YLABEL$,1,18)
58231 FOR I.GRF% = 0 TO LEN(YLABEL$)
58234    LOCATE (11 - (LEN(YLABEL$) \ 2) + I.GRF%),1
58237    PRINT MID$(YLABEL$,(I.GRF% + 1),1)
58240 NEXT I.GRF%
58243 IF LEN(COMMENT$) > 78 THEN COMMENT$ = MID$(COMMENT$,1,78)
58246 FOR I.GRF% = 1 TO LEN(COMMENT$)
58249    IF I.GRF% <= 39 THEN LOCATE 23,I.GRF% ELSE LOCATE 24,
I.GRF% - 39
58252    PRINT MID$(COMMENT$,I.GRF%,1);
58255 NEXT I.GRF%
58258 WINDOW (0,LOVAL) - (291,HIVAL)
58261 IF TYPE% > 2 THEN VIEW (17,16) - (308,79)
ELSE VIEW (17,16) - (308,143)
58264 RETURN
58267 ' *** SUBROUTINE TO DRAW AXES AND TIC MARKS FOR SCR. GRAPH
58270 LINE (16,15)-(310,144),,B
58273 FOR I.GRF% = 26 TO 310 STEP 10
58276    LINE (I.GRF%,12) - (I.GRF%,15)
58279    LINE (I.GRF%,147) - (I.GRF%,144)
58282 NEXT I.GRF%
58285 FOR I.GRF% = 116 TO 310 STEP 100
58288    LINE (I.GRF%,10) - (I.GRF%,15)
58291    LINE (I.GRF%,149) - (I.GRF%,144)
58294 NEXT I.GRF%
58297 FOR I.GRF% = 25 TO 144 STEP 10
58300    LINE (13,I.GRF%) - (16,I.GRF%)
58303    LINE (312,I.GRF%) - (310,I.GRF%)
58306 NEXT I.GRF%
58309 RETURN
58312 ' *** SUBROUTINE to draw axes and tic marks for alt. graph
58315 LINE (16,15) - (309,80),,B
58318 LINE (16,90) - (309,155),,B
58321 FOR I.GRF% = 26 TO 309 STEP 10
58324    LINE (I.GRF%,12) - (I.GRF%,15)
58327    LINE (I.GRF%,158) - (I.GRF%,155)
58330 NEXT I.GRF%
58333 FOR I.GRF% = 116 TO 309 STEP 100
58336    LINE (I.GRF%,10) - (I.GRF%,15)
```

```
58339    LINE (I.GRF%,160) - (I.GRF%,155)
58342 NEXT I.GRF%
58345 FOR I.GRF% = 70 TO 16 STEP -10
58348    LINE (13,I.GRF%) - (16,I.GRF%)
58351    LINE (312,I.GRF%) - (309,I.GRF%)
58354 NEXT I.GRF%
58357 FOR I.GRF% = 145 TO 90 STEP -10
58360    LINE (13,I.GRF%) - (16,I.GRF%)
58363    LINE (312,I.GRF%) - (309,I.GRF%)
58366 NEXT I.GRF%
58369 RETURN
58372 REM PAGE
58375 ' NAME       INITGRAPH_default
58378 LOVAL = 0
58381 HIVAL = 4095
58384 TYPE% = 1
58387 PLETE% = 1
58390 BACKGROUND% = 0
58393 FORMAT%(0) = 3
58396 FORMAT%(1) = -1
58399 FORMAT%(2) = 0
58402 FORMAT%(3) = 0
58405 FORMAT%(4) = 0
58408 XLABEL$ = ""
58411 YLABEL$ = ""
58414 COMMENT$ = ""
58417 TITLE$ = ""
58420 GOSUB 58153
58423 RETURN
58426 REM PAGE
58429 ' NAME       CLEARGRAPH
58432 ON (TYPE% + 1) GOSUB 58438,58438,58438,58447,58447,58447
58435 RETURN
58438 ' *** SUBROUTINE to erase  scrolling graph
58441 CLS
58444 RETURN
58447 ' *** SUBROUTINE TO ERASE  ALTERNAT. GRAPH
58450 VIEW (17,16) - (308,79)
58453 CLS
58456 VIEW (17,91) - (308,154)
58459 CLS
58462 IF BASE.GRF% = 0 THEN VIEW (17,16) - (308,79)
        ELSE VIEW (17,91) - (308,154)
58465 RETURN
58468 REM PAGE
58471 ' NAME       NEXTPOINT
58474 ON TYPE%+1 GOSUB 58480,58480,58480,58555,58555,58555
58477 RETURN
58480 ' *** SUBROUTINE for scrolling type plot
58483 IF INPTR.GRF% < 292 THEN GOTO 58528
58486   FOR I.GRF% = 0 TO NWAVES.GRF% - 2
```

```
58489           SWAP FORMAT.GRF%(I.GRF%),FORMAT.GRF%(I.GRF% + 1)
58492   NEXT I.GRF%
58495   FOR I.GRF% = 0 TO 290
58498           J.GRF% = I.GRF% MOD NWAVES.GRF%
58501           XVALUE.GRF = I.GRF%
58504           IF (I.GRF% MOD NWAVES.GRF%) = 0 THEN LINE
(XVALUE.GRF,HIVAL) -    ((XVALUE.GRF + NWAVES.GRF%),LOVAL),0,BF
58507           PLT.GRF(I.GRF%) = PLT.GRF(I.GRF% + 1)
58510           YVALUE.GRF = PLT.GRF(I.GRF%)
58513           PLOTCOLOR.GRF% = FORMAT.GRF%(J.GRF%)
58516           ON TYPE% + 1 GOSUB 58594,58603,58624
58519           INPTR.GRF% = 291
58522   NEXT I.GRF%
58525   J.GRF% = (J.GRF% + 1) MOD NWAVES.GRF%
58528 '
58531 PLT.GRF(INPTR.GRF%) = VALUE
58534 XVALUE.GRF = INPTR.GRF%
58537 YVALUE.GRF = PLT.GRF(INPTR.GRF%)
58540 PLOTCOLOR.GRF% = FORMAT.GRF%(J.GRF%)
58543 ON TYPE% + 1 GOSUB 58594,58603,58624
58546 J.GRF% = (J.GRF% + 1) MOD NWAVES.GRF%
58549 INPTR.GRF% = INPTR.GRF% + 1
58552 RETURN
58555 ' *** SUBROUTINE for alternating type plot
58558 YVALUE.GRF = VALUE
58561 PLOTCOLOR.GRF% = FORMAT.GRF%(J.GRF%)
58564 ON TYPE% - 2 GOSUB 58594,58603,58624
58567 XVALUE.GRF = XVALUE.GRF + 1
58570 IF XVALUE.GRF <= 291 THEN GOTO 58588
58573   XVALUE.GRF = 0
58576   IF BASE.GRF% = 0 THEN BASE.GRF% = 1 : VIEW (17,91) - (308,154)
          : CLS : GOTO 58588
58579   BASE.GRF% = 0
58582   VIEW (17,16) - (308,79)
58585   CLS
58588 J.GRF% = (J.GRF% + 1) MOD NWAVES.GRF%
58591 RETURN
58594 ' *** SUBROUTINE for dot plot
58597 PSET (XVALUE.GRF,YVALUE.GRF),PLOTCOLOR.GRF%
58600 RETURN
58603 ' *** SUBROUTINE for line plot
58606 IF XVALUE.GRF >= NWAVES.GRF% THEN GOTO 58615
58609   PSET (XVALUE.GRF,YVALUE.GRF),PLOTCOLOR.GRF%
58612   GOTO 58618
58615  LINE ((XVALUE.GRF - NWAVES.GRF%),LAST.GRF(J.GRF%)) -
         (XVALUE.GRF,YVALUE.GRF ),PLOTCOLOR.GRF%
58618 LAST.GRF(J.GRF%) = YVALUE.GRF
58621  RETURN
58624 '  ******* suBROUTINE FOR HISTOGRAM PLOT (TYPE%=2 OR 5).
58627 LINE (XVALUE.GRF,LOVAL)-(XVALUE.GRF,YVALUE.GRF) ,PLOTCOLOR.GRF%
58630 RETURN
```

```
58633 REM PAGE
58636 '******************** L A S T   P O I N T ************************
58639 PLOTCOLOR% = PLOTCOLOR.GRF%
58642 XVALUE =XVALUE.GRF
58645 YVALUE =YVALUE.GRF
58648 RETURN
58651 REM PAGE
```

APPENDIX C.2

Closed Loop Control Program

```
1 DEF SEG
2 A=0:I=0:J=0:ADR=0:LABSOFT.SEG=0:O=0
3 DIM ZPRGM%(150):A=VARPTR(ZPRGM%(0)) ' Get a pointer to the array
4 IF A<0 THEN A=A+65536!
5 FOR I=5 TO 11:READ J:POKE A+I,J:NEXT ' Poke program into array
6 FOR I=20 TO 108:READ J:POKE A+I,J:NEXT
7 POKE A+21,A-INT(A/256)*256:POKE A+22,INT(A/256) ' Poke in the address
8 DATA &H42,&h41,&h53,&h4c,&h49,&h42,&h00
9 DATA &hbb,&h00,&h00,&h1e,&h06,&h2e,&h8c,&h97,&h0e,&h00,&h2e,&h89,&ha7
10 DATA&h00,&h8c,&hc8,&h8e,&hd8,&h8e,&hd0,&hc6,&h87,&h04,&h00,&h00,&h8d
11 DATA&h0d,&h01,&hb4,&h3d,&hb0,&h00,&h8d,&h97,&h05,&h00,&h53,&hcd,&h21
12 DATA&h17,&h5b,&h53,&h8d,&h97,&H00,&h00,&h50,&h8b,&hd8,&hb4,&h3f,&hb9
13 DATA&h00,&hcd,&h21,&h5b,&h72,&H04,&hb4,&h3e,&hcd,&h21,&h5b,&h73,&h09
14 DATA&h87,&h00,&h00,&hc6,&h87,&h04,&h00,&hff,&h8b,&ha7,&h0c,&h00,&h8e
15 DATA&h0e,&h00,&h07,&h1f,&hcb,&h0c,&ha7,&h72,&h04,&h89,&h97
16 ADR=A+20:CALL ADR ' Get address of the device driver
17 IF PEEK(A+4)=255 THEN BEEP:PRINT"*** ERROR - LABBASIC.COM Device
Driver Is Not Installed":END
18 LABSOFT.SEG=PEEK(A)+256*PEEK(A+1)
19 IF LABSOFT.SEG<0 THEN LABSOFT.SEG=LABSOFT.SEG+65536!
20 O=PEEK(A+2)+256*PEEK(A+3)+197
21 DEF SEG = LABSOFT.SEG
22 :
23 COMPAT=PEEK(O+0)+256*PEEK(O+1)
24 SETSTAT=PEEK(O+2)+256*PEEK(O+3)
25 AINFM=PEEK(O+6)+256*PEEK(O+7)
26 AINM=PEEK(O+8)+256*PEEK(O+9)
27 AINS=PEEK(O+10)+256*PEEK(O+11)
28 AINSC=PEEK(O+12)+256*PEEK(O+13)
29 AINTS=PEEK(O+14)+256*PEEK(O+15)
30 AOUFM=PEEK(O+16)+256*PEEK(O+17)
31 AOUM=PEEK(O+18)+256*PEEK(O+19)
32 AOUS=PEEK(O+20)+256*PEEK(O+21)
33 AOUSC=PEEK(O+22)+256*PEEK(O+23)
34 BCDINM=PEEK(O+24)+256*PEEK(O+25)
35 BCDINS=PEEK(O+26)+256*PEEK(O+27)
36 BCDINTS=PEEK(O+28)+256*PEEK(O+29)
37 BCDOUM=PEEK(O+30)+256*PEEK(O+31)
38 BCDOUS=PEEK(O+32)+256*PEEK(O+33)
39 BINM=PEEK(O+34)+256*PEEK(O+35)
40 BINS=PEEK(O+36)+256*PEEK(O+37)
41 BINTS=PEEK(O+38)+256*PEEK(O+39)
42 BITINS=PEEK(O+40)+256*PEEK(O+41)
43 BITINTS=PEEK(O+42)+256*PEEK(O+43)
44 BITOUS=PEEK(O+44)+256*PEEK(O+45)
45 BOUM=PEEK(O+46)+256*PEEK(O+47)
```

```
46 BOUS=PEEK(0+48)+256*PEEK(0+49)
47 :
48 STINM=PEEK(0+50)+256*PEEK(0+51)
49 STINS=PEEK(0+52)+256*PEEK(0+53)
50 STINTS=PEEK(0+54)+256*PEEK(0+55)
51 CINM=PEEK(0+56)+256*PEEK(0+57)
52 CINS=PEEK(0+58)+256*PEEK(0+59)
53 CINTS=PEEK(0+60)+256*PEEK(0+61)
54 CSET=PEEK(0+62)+256*PEEK(0+63)
55 BEEPFUN=PEEK(0+64)+256*PEEK(0+65)
56 CONTIN=PEEK(0+66)+256*PEEK(0+67)
57 DELAY=PEEK(0+68)+256*PEEK(0+69)
58 DINS=PEEK(0+70)+256*PEEK(0+71)
59 DOUS=PEEK(0+72)+256*PEEK(0+73)
60 NORMAL=PEEK(0+74)+256*PEEK(0+75)
61 STATS=PEEK(0+76)+256*PEEK(0+77)
62 STOPFUN=PEEK(0+78)+256*PEEK(0+79)
63 VERSION=PEEK(0+80)+256*PEEK(0+81)
64 TINFM=PEEK(0+82)+256*PEEK(0+83)
65 TLIN=PEEK(0+84)+256*PEEK(0+85)
66 TINM=PEEK(0+86)+256*PEEK(0+87)
67 TINS=PEEK(0+88)+256*PEEK(0+89)
68 TINSC=PEEK(0+90)+256*PEEK(0+91)
69 TINTS=PEEK(0+92)+256*PEEK(0+93)
70 :
71 :
99 ' ------------------------------------------------------------------
100 '                    Closed Loop Run Program
110 '                 Rev. 4.0  - JANUARY 27, 1988
115 '                       Sean P. Forestell
120 '------------------------------------------------------------------
125 '
130 ON ERROR GOTO 11000
131 '
132 ' ------------------------------------------------------------------
133 '                        System Initialization
134 ' ------------------------------------------------------------------
135 '
160 DIM FORMAT%(5),BC$(20),PLV(900),Y(10),BM$(10),T(10)
200 DIM LAST.GRF(3),FORMAT.GRF%(4),PLT.GRF(291)
210 DIM RAW2%(500),FGAL%(15),MP$(10),RAW1%(500)
215 DIM OFFP(4),IMIN(4),SLP(4)
220 '
230 '------------------------ Kinetic Constants ----------------------
235 '
240 READ V,MUMAX,KS,MAINT,B,QPMAX,A
250 DATA 14.0,0.123,1.0,0.002,0.30,0.001,0.030
255 '
260 '---------------------- Controller Parameters --------------------
265 '
270 READ KCP,KCS,KCZ,TIP,TIS,LAM,P11,P12,P22,KCF
280 DATA -0.44,10.0,0.019,2.80,8.0,0.75,1.0,0.0,1.0,0.1
```

```
285 '
290 '---------------------- Pump Calibration data --------------------
291 '
293  IMIN(1) = 6!: IMIN(2) = 6!: IMIN(3) = 6!: IMIN(4) = 6!
297  OFFP(1) = 4.96987: OFFP(2) = 5.1405: OFFP(3) = 4.5928:
OFFP(4) = 4.6119
299  SLP(1) = 2.2413: SLP(2) = 2.0696: SLP(3) = 3.2978: SLP(4) = 3.3196
300  PA%=4 :PB%=3 :PO%=2 :PC%=1
301  IAMIN = IMIN(4): OFFA = OFFP(4): SLA = SLP(4): ' Defaults
302  IBMIN = IMIN(3): OFFB = OFFP(3): SLB = SLP(3)
303  ICMIN = IMIN(1): OFFC = OFFP(1): SLC = SLP(1)
304  IOMIN = IMIN(2): OFFO = OFFP(2): SLO = SLP(2)
305  FAMAX = (20!-OFFA)/SLA: FBMAX = (20!-OFFB)/SLB
306  FCMAX = (20!-OFFC)/SLC: FOMAX = (20!-OFFO)/SLO
307  FAMIN=(IAMIN-OFFA)/SLA: FBMIN=(IBMIN-OFFB)/SLB
308  FCMIN=(ICMIN-OFFC)/SLC: FOMIN=(IOMIN-OFFO)/SLO
309  LPRINT " ### PUMP ARRANGEMENT : A , B , C , O  : "; PA%;
"   ";PB%;"   ";PC%;"   ";PO%
314 '
315 '----------------- PO & TEMP Probe Calibration -------------------
316 '
320  READ POO   , POSL
322  DATA  -.9305928   , 108.50313
325  READ TMO    , TMSL
327  DATA  -.728981   , 150.4392
330 '
332 FALSE = 0 :TRUE = NOT FALSE
335 XON$=CHR$(17) : XOFF$=CHR$(19)
337 COMFIL$="com1:300,e,7"
340 OPEN COMFIL$ AS #1:PAUSE =FALSE
375 '
380 '---------------------- Flag initialization ----------------------
385 '
390 FGBR$="off":FGTS$="off":FGPR$="off":FGSP$="off" :FGD$="on"
395 '
400 '----------------------- Key definitions -------------------------
410 '
440 KEY 15,CHR$( 4)+CHR$(70):ON KEY (15) GOSUB  5300:KEY (15) ON
450 ON KEY ( 1) GOSUB  4000:KEY ( 1) ON
460 ON KEY ( 2) GOSUB  4680:KEY ( 2) ON
470 ON KEY ( 3) GOSUB  4800:KEY ( 3) ON
480 ON KEY ( 4) GOSUB  5560:KEY ( 4) ON
490 ON KEY ( 5) GOSUB 58000:KEY ( 5) ON
495 ON KEY ( 6) GOSUB 14000:KEY ( 6) ON
500 ON KEY ( 7) GOSUB 56000:KEY ( 7) ON
505 ON KEY ( 8) GOSUB 55000:KEY ( 8) ON
510 ON KEY ( 9) GOSUB  7000:KEY ( 9) ON
520 ON KEY (10) GOSUB  3420:KEY (10) ON
521 KEY OFF
525 '
530 '------------------- Initialize ISAAC  -  OUTPUTS ----------------
535 '
```

```
550 OADIN = 4095: OADFF = 0: OOFF = 4: OSPAN  = 16
552 AOCHA%=3: AOCHB%=2: AOCHO%=1: AOCHC%=0
560 AOPT$=""     : OSLOPE = OSPAN/OADIN
561 '
562 '------------------- Initialize ISAAC  -  INPUTS -----------------
563 '
565 IADIN =4095  :IADFF =0  : IOFF =0   :ISPAN  =1!
570 AICH1% = 1 : AICH2% = 2
572 INOPT$=""   :ISLOPE = ISPAN/IADIN
575 '
585 STAT%=0     :CALL SETSTAT(STAT%)
590 COUNT=225:HZ=75:OPT$=""
600 MP$(1)= "o4132t220cdefgabo5116c":MP$(3)="o5132co4bagfedl16c"
601 MP$(2)= "mfmst150116o3eo118g"   :MP$(4)="mfo5t80fcfcfcfc"
610 'PLAY "mbo214ao314ao414ao514a"
650 CLS:SCREEN 0,0,0:WIDTH 80:KEY OFF
651 '
652 ' ------------------------------------------------------------------
653 '              Program Start-up & Initiallization
654 ' ------------------------------------------------------------------
655 '
660 LOCATE 2,1:
PRINT"------------------------------------------------------------------
--------";
670 LOCATE 4, 21: PRINT " CLOSED LOOP RUN PROGRAM  -  PREDICTION ";
680 LOCATE 6, 21: PRINT "For Penicillin_G Continuous Fermentations"
690 LOCATE 8,1:
PRINT"------------------------------------------------------------------
----------------";
700 LOCATE 11,21 :PRINT "   Revision 4.0 --- JANUARY 27, 1988 "
710 LOCATE 22,1:PRINT
"------------------------------------------------------------------
-----------";
720 LOCATE 25,1:INPUT "Press <CR> to continue ...",C$:CLS
730 LOCATE 2,1:
PRINT"------------------------------------------------------------------
-----------------";
740 LOCATE 5,30: PRINT "SYSTEM INITIALIZATION"
750 LOCATE 8,1:
PRINT"------------------------------------------------------------------
-----------------";
760 LOCATE 12,15
765 '
775 GF=15:GP=1.2:GT=19.5:COFFO=1!:PHASE$="   GROWTH PHASE"
776 Y(1) = 0!: Y(2)=.1 :Y(4)=.3287 :Y(5)=.002
777 Y(6)=21.4567:Y(7)=10.23
778 FOR I%=0 TO 7:T(I%)=12.31:NEXT I%
780 '
790 '---------------------- Initial Values --------------------------
791 '
795 LOCATE 14,15:INPUT" What's the Initial Biomass (g/L) ";BMASSO
800 LOCATE 17,15:INPUT" Enter the offtime (h) ";OFFTIME
```

```
807 LOCATE 20,15:INPUT "Please Enter Update Interval (min) ";TSAMPLE%
810 N2%=TSAMPLE%/10:TSH#=TSAMPLE%/60#:TSAMPLE=TSAMPLE%*60:
TSAMPL1%=TSAMPLE%
812 LPRINT " ###  UPDATE  INTERVAL   "; TSAMPLE%;"  min"
815 PO1=0:PO2=110:TM1=15:TM2=30
820 '
880 CLS:LOCATE 25,71:PRINT TIME$ :LOCATE 9,10
905 '
910 '---------------------- Disk output parameters ---------------------
915 '
920 LOCATE 15,10:INPUT"Enter filename for Disk Output(up to 7 letters)"
;OFLE$
930 IF OFLE$ <> "" THEN 960  ELSE LOCATE 18,10
940 PRINT "The default name OUTPUT.DAT will be used";
950  INPUT "Enter <CR> to continue...",A$:OFLE$="OUTPUT.DAT"
960 CLS
970 '
980 OPEN "plot1.dat" FOR APPEND AS 3
1000 '
1010 LOVAL=0:HIVAL=4095:TYPE%=4:PLETE%=1:BACKGROUND%=0
1020 FORMAT%(0)=3:FORMAT%(1)=-1:FORMAT%(2)=0:FORMAT%(3)=0:FORMAT%(4)=0
1030 XLABEL$="":YLABEL$="":COMMENT$="test"
1040 '
1050 BC$(1)="  DIL. RATE          (h-1)"
1060 BC$(2)=" PAA Feed Conc.     (g/L)"
1063 BC$(3)=" CO2   in          (%vol)"
1065 BC$(4)=" CO2  out          (%vol)"
1068 BC$(5)="  O2   in          (%vol)"
1070 BC$(6)="  O2  out          (%vol)"
1080 BC$(7)=" CO2  produced      (%vol)"
1090 BC$(8)="  O2  consumed      (%vol)"
1110 BC$(9)=" PO2              (% Sat.)"
1120 BC$(10)="Temperature      (Cent. )"
1130 '
1140 BM$(1)="PAA          ( g/L)"
1150 BM$(2)="Glucose      ( g/L)"
1160 BM$(3)="Biomass    (g.dw/L)"
1170 BM$(4)="CO2 in       (%vol)"
1180 BM$(5)="CO2 out      (%vol)"
1190 BM$(6)="O2  in       (%vol)"
1195 BM$(7)="O2  out      (%vol)"
1200 BM$(8)="Pen_G        ( g/L)"
1210 '
1220 CHNG$(0) ="   Feed Concentrations"
1230 CHNG$(1) ="   Controller Constants    "
1240 CHNG$(2) ="   ISAAC Parameters (input or output)"
1250 CHNG$(3) ="   *** Alarm Limits *****  "
1260 '
1270 GOSUB 56000
1550 SGR0=2.1:P0=0:AGE0=20:PAA0=.01: MT%=4 :CLS
1560 PAAM=0!: PENM=0!: VAD=0!: VBD=0!: VCD=0!
1565 VADES=0!: VBDES=0!: VCDES=0!
```

```
1570 FGC%=0 :ISML%=1 : TMO=0 :DAY=86400!
1571 GOSUB  3440
1572 SFMAX=SA: SFMIN=SC: ZFMAX=ZB: ZFMIN=ZC
1575 YG=(1.5*CSLA + .5*SFMAX)/SFMAX
1577 SIGC=MUDES/YG+MAINT:TSAMP=0!:SIGQ=SIGC
1579 SIGMIN=0!: SIGMAX=1.5*SIGQ
1580 QC=.00757
1590 '
1600 DMIN=FOMIN/V*60/1000: DMAX=FOMAX/V*60/1000: TTTO=0!: ERMUO=0!
1610 MU=MUDES:ERPO=PDES:ERZO=ZDES:TSAMPO=OFFTIME:DILR=FOMIN/V*60!/1000!
1620 CO=LOG(BMASSO):MUO=MUDES:ERORFA=0:ERORFB=0:ERORFC=0:TSAMP=OFFTIME
1640 '
1740 '
1760 LOCATE 4,1: PRINT
"------------------------------------------------------------------
------------"
1780 LOCATE 8,20: PRINT "FOR HELP ON ACCEPTABLE KEYBOARD INTERRUPTS";
1800 LOCATE 11,35: PRINT "press <F1>";
1820 LOCATE 15,1: PRINT
"------------------------------------------------------------------
-----------"
1840 LOCATE 19,20 : PRINT "When you Enter <CR> , The Timer STARTS ...";
1860 LOCATE 25,71: PRINT TIME$
1880 LOCATE 25,1: INPUT "Enter <CR> to continue.....",C$
1900
'------------------------------------------------------------------
1920 '                                        SET TIMER AND TRAP KEYS
"on"
1940
'------------------------------------------------------------------
1960 ON TIMER (N2%*60) GOSUB 6280 :TIMER ON :TIME1=TIMER:TIME0=TIMER
1970 TIMEC0=TIME0
1990 '
1995 FGK$="on"
2000 GOSUB 6280
2001 '
2002 ' ----------------------------------------------------------------
2003 '                              Main Program
2004 ' ----------------------------------------------------------------
2005 '
2006 IPR%=19 :FGK$="off":FGPR$="on"
2080 M=4:N=6:GOSUB 12000
2083 IF FGPR$="off" THEN 2092 ELSE IPR%=IPR%+1
2084 IF IPR% <> 20 THEN 2087 ELSE IPR%=0
2085 LPRINT "    Date    Time    Tel   PO2   Temp.  B.exp.
FA   FB   FC  FO "
2087 LPRINT "* "; LEFT$(DATE$,5);" ";LEFT$(TIME$,5);" * ";
2088 RTMEL =OFFTIME + TMEL: LPRINT USING "###.##";RTMEL;
2089 LPRINT USING " ###.## ";RPO;RTM;
2090 LPRINT USING "##.###";XEXP#;FA;FB;FC;FO:FGPR$="off"
2091 LPRINT " "
2092 IF SUMAL < .99 THEN 2135
```

```
2095 LOCATE 25,31:FOR I%=1 TO 4:PRINT FGAL%(I%);:NEXT I%:PRINT "
ALARM";
2097 PLAY MP$(4)
2100 A$=INKEY$:IF A$="" THEN 2097 ELSE SUMAL=0
2105 IF FGER$="on" THEN PLAY MP$(4) :LOCATE 24,7:PRINT ERR ,ERL;
2135 LOCATE 2,4:PRINT ISML%,FGC%
2140 IF TIMER<TIME0 AND FGD$="on" THEN TIMEC0=TIMEC0-DAY :FGD$="off"
2145 IF TIMER>TIME0 THEN FGD$="on"
2152 KEY (1) ON :KEY(3) ON:KEY(4) ON:KEY(5) ON
2156 KEY (7) ON :KEY(8) ON:KEY(9) ON:KEY(10) ON:KEY(15) ON
2160 'J1%=INT((TIMER-TIME1)/(TSAMPLE/N1%))
2162 TMEL=(TIMER-TIMEC0)/3600!
2165 '
2170 LOCATE 2,52 : PRINT "ELAPSED TIME (h) =";: RTMEL = OFFTIME +TMEL
2180 LOCATE 2,72 : PRINT USING "###.###";RTMEL;
2185 LOCATE MT+4,32: PRINT USING  "##.#";RTM;
2195 LOCATE ,38:PRINT USING "###.#";RPO;
2200 '
2375 '
2400 A$=INKEY$
2410 IF A$=""OR A$=CHR$(13) THEN 2430
2415 IF A$="1" THEN 2500           :'Next screen
2420 LOCATE 24,1:  PRINT "Can't understand  ---> ";A$;
2430 LOCATE 25,1: PRINT " -1- --> Change screen ";:LOCATE 25,70
2440 PRINT TIME$;
2450 IF FGBR$="on" GOTO 3160
2460 IF FGSP$="on" THEN GOSUB 3440
2470 IF FGPR$="on" THEN GOSUB 6080
2480 GOTO 2140
2490 '
2500 ' SCREEN 2 ------------- MEASUREMENTS
2510 GOTO 2140
3150 '
3160
'-------------------------------------------------------------------
3180 '                       Close Output Files and Logout
3200
'-------------------------------------------------------------------
3220 '
3240 CLS
3250 CLOSE #1
3260 LOCATE 14,15: PRINT "CLEANING UP....and....SAVING OUTPUT FILES..."
3280 END
3300 '
3320 '
3340
'-------------------------------------------------------------------
3360 '                     trap <F10> key,   service setpoints
3380
'-------------------------------------------------------------------
3400 '
3420 CLS:FGSP$="on":RETURN
```

```
3440 FGK$="on":CLS:LOCATE 25,1
3460 FGSP$="off":PRINT "Servicing <F10> Key...      " TAB(71) TIME$;
3480 LOCATE 3,1:PRINT
"------------------------------------------------------------------------
------------";
3500 LOCATE 5,25: PRINT "UPDATING  SETPOINTS";
3520 LOCATE 7,1: PRINT
"------------------------------------------------------------------------
------------";
3540 LOCATE 12,15: INPUT "Enter desired growth rate (1/h) ";MUDES
3560 IF MUDES<=0! OR MUDES>=MUMAX THEN 3565 ELSE 3580
3565 CLS:LOCATE 20,15: PRINT "That can't be right, try again."
3570 GOTO 3540
3580 SDES = MUDES*KS/(MUMAX - MUDES)
3590 '
3600 CLS: LOCATE 12,15: INPUT "Enter desired PAA conc. (g/L) ";ZDES
3620 IF ZDES>=0! AND ZDES<=.5 THEN 3750 ELSE 3730
3630 CLS: LOCATE 20,15: PRINT "That can't be right, try again."
3640 GOTO 3700
3650 '
3700 CLS: LOCATE 7,1: INPUT "Enter desired penicillin conc (g/L)";PDES
3710 LOCATE 9,1: INPUT "Is the inputted value correct? (y/n)";CC$
3720 IF CC$="y" OR CC$="Y" THEN 3730 ELSE 3700
3730 '
3820 '
3840 LPRINT " ### SET POINTS MUDES, ZDES, PDES: ";MUDES;"    ";
ZDES;"    ";PDES
3940 CLS: FGK$="off" :IF FGC%=0 THEN RETURN ELSE RETURN 2080
3960 '
3980
'------------------------------------------------------------------------
4000 '                        trap help key ... <F1>
4020
'------------------------------------------------------------------------
4040 '
4060 CLS
4080 LOCATE 25,1: PRINT "Servicing <F1> Key...      " TAB(71) TIME$;
4100 'SOUND 880,2!
4120 'SOUND 440,2!
4140 LOCATE 2,1: PRINT
"------------------------------------------------------------------------
------------"
4160 LOCATE 4,8 :PRINT ">>>>>  ACCEPATABLE KEYBOARD
INTERRUPTS ARE  <<<<<";
4180 LOCATE 6, 15: PRINT "<F1>.....print this message"
4200 LOCATE 7, 15: PRINT "<F2>.....clear the screen"
4220 LOCATE 8,15: PRINT "<F3>.....change sampling interval"
4240 LOCATE 9,15: PRINT "<F4>.....REQUEST LOGGING ";
4260 LOCATE 10,15: PRINT "<F5>.....GRAPHS  ";
4280 LOCATE 11,15: PRINT "<F6>.....REARRANGE PUMPS ";
4300 LOCATE 12,15: PRINT "<F7>.....change parameters ";
4320 LOCATE 13,15: PRINT "<F8>.....Pump Flow Control Routine ";
```

```
4340 LOCATE 14,15: PRINT "<F9>.....ENTER MEASUREMENTS";
4360 LOCATE 15,15: PRINT "<F10>.....change setpoint";
4380 LOCATE 16,15: PRINT "^BREAK.....finish this run";
4400 TIME2=TIMER
4420 LOCATE 19,15: PRINT
"-------------------------------------------------------------------
-----------"
4440 LOCATE 21,15: PRINT "Hit any Key to Continue";
4460 C$=INKEY$
4480 IF C$="" AND TIMER - TIME2 < 10 THEN 4460
4500 CLS
4520 IF C$="" THEN'BEEP
4540 CLS:LOCATE 20,15:PRINT "SORRY TIMEOUT"
4560 RETURN 2080
4580 '
4600
'-------------------------------------------------------------------
4620 '                        trap <F2>  key,  clear screen
4640
'-------------------------------------------------------------------
4660 '
4680 CLS:RETURN 2080
4700 '
4720
'-------------------------------------------------------------------
4740 '                        trap <F3>  key, TSAMPLE change
4760
'-------------------------------------------------------------------
4780 '
4800 FGK$="on"
4810 CLS:LOCATE 4,1:PRINT
"-------------------------------------------------------------------
------------";
4820 LOCATE 7,26: PRINT "CHANGE SAMPLING INTERVAL"
4840 LOCATE 10,1: PRINT
"-------------------------------------------------------------------
-------------";
4860 LOCATE 25,1: PRINT "Servicing <F3> key...      " TAB(71) TIME$;
4900 LOCATE 15,15: INPUT "Please Enter NEW Sampling Interval (min) ";
TSAMPL1%
4920 IF TSAMPL1%=0 THEN CLS: RETURN 2080
4940 IF TSAMPL1%< 10 THEN CLS :LOCATE 10,10:PRINT "A sampling
Interval of ";    TSAMPL1%;" is too small...Try again...":GOTO 4810
4950 LPRINT " ### Sampling Interval  : ";TSAMPLE%
4980 FGTS$="on":FGK$="off":RETURN 2080
5220 '
5240
'-------------------------------------------------------------------
5260 '                      trap ^BREAK key,  finish run
5280
'-------------------------------------------------------------------
5300 '
```

```
5320 CLS
5340 LOCATE 25,1: PRINT "Servicing  ^BREAK....    " TAB(71) TIME$;
5360 LOCATE 10,15: PRINT "Do you really want to FINISH this run ??"
5380 LOCATE 12,15: INPUT "Please answer (y/n)";EE$
5400 IF EE$="y" OR EE$="Y" THEN 3200 ELSE 5410
5410 CLS: LOCATE 20,10: PRINT "THEN STOP PLAYING WITH THE BREAK KEY"
5420 CLS: LOCATE 20,10: PRINT "THEN STOP PLAYING WITH THE BREAK KEY"
5440 RETURN
5460 '
5480
'-----------------------------------------------------------------------
5500 '                       trap <F4> key,  request logging
5520
'-----------------------------------------------------------------------
5560 '
5570 CLS:LOCATE 4,1: PRINT
"-----------------------------------------------------------------------
---";
5580 LOCATE 7,10:PRINT "Make sure the PRINTER is ON LINE and the
PAUSE key OFF";
5600 LOCATE 10,10:PRINT "Else the Program will ABORT and all DATA
will be lost";
5620 LOCATE 13,1: PRINT
"-----------------------------------------------------------------------
------------";
5640 LOCATE 21,15: PRINT "Hit any Key AFTER you have checked...";
5660 TIME2=TIMER
5680 C$=INKEY$
5700 IF C$="" AND TIMER - TIME2 < 15 THEN 5680
5720 CLS
5740 IF C$="" THEN CLS:LOCATE 20,15:PRINT"SORRY TIMEOUT...        ";
5780 LOCATE 25,1: PRINT "Servicing <F4> PRINTER...     " TAB(71) TIME$;
5800 'LPRINT " ":LPRINT " ":LPRINT " "
5820 'LPRINT
"-----------------------------------------------------------------------
----------"
5840 'LPRINT " "
5860 'LPRINT "STATUS OF THE SYSTEM at TIME = ";TIME$
5880 'LPRINT " "
5900 'LPRINT
"-----------------------------------------------------------------------
----------"
5920 'LPRINT " "
5940 'LPRINT "Elapsed Time = ";(TIMER-TIME0)/3600;" (h)"
5960 'LPRINT "Sampling Interval = ";TSAMPLE%;" (MIN)"
5980 'LPRINT "Current Liquid Volume = ";VOL;" (L)"
6000 'LPRINT "Current Biomass = ";WT(0,CN1%) ;" (g/L)"
6020 'LPRINT "***"
6040 'LPRINT " ":'LPRINT " ":'LPRINT " "
6060  RETURN 2080
6080 '
```

```
6280 ' ------------------------------------------------------------
6300 '                          Servicing     Isaac
6320 ' ------------------------------------------------------------
6330 '
6340 KEY(1) STOP: KEY(3) STOP:KEY(4) STOP:KEY(5) STOP
6341 KEY(7) STOP: KEY(8) STOP:KEY(9) STOP:KEY(10) STOP
6342 FGC%=FGC%+1
6345 '
6348 IF ISML%=1 AND FGC%=1 THEN PLAY MP$(2):GOTO 6600
6350 IF FGC%=11 THEN 6357 ELSE PLAY MP$(1):GOTO 6806
6355 '
6357 TIME1=TIMER:PLAY MP$(2)
6360 FGC%=1 :TMO=TMO+TSH# :ISML%=ISML%+1
6380 '
6420 IF FGTS$<>"on" THEN 6530
6440 TSAMPLE%=TSAMPL1%:N2%=TSAMPLE%/10
6460 ON TIMER (N2%*60) GOSUB 6280:TIMER ON :TIME1=TIMER
6480 FGTS$="off":TSH#=TSAMPLE%/60# :TSAMPLE=TSAMPLE%*60
6530 '
6600 FGPR$="on"
6602 XEXP#=BMASSO*EXP(MUDES*(TMEL-(TSAMP-OFFTIME)))
6610 SF=SIGC*XEXP#/DILR + SDES
6611 ZF=ZDES+QC*B*XEXP#/DILR
6612 IF SF>SFMAX THEN SF=SFMAX
6614 IF SF<SFMIN THEN SF=SFMIN
6615 IF ZF > ZFMAX THEN ZF=ZFMAX
6616 IF ZF < ZFMIN THEN ZF=ZFMIN
6620 F = V*1000!*DILR/60!
6622 FA=SF*F/SA: FB=(ZF*F-ZA*FA)/ZB: FC=F-FA-FB: FO=FA+FB+FC
6623 LPRINT "***XEXP, SF, ZF ";XEXP#,SF,ZF
6624 IF FC<0! THEN FC=0!
6625 IF FC=0 THEN FB=F-FA
6630 IF FB < 0 THEN FB=0!
6640 IF FB=0 THEN FC=F-FA
6680 '
6690 FAN%=CINT(FA/FAMIN*10+ERORFA): ERORFA=10*FA+ERORFA-FAN%*FAMIN
6700 FBN%=CINT(FB/FBMIN*10+ERORFB): ERORFB=10*FB+ERORFB-FBN%*FBMIN
6705 FCN%=CINT(FC/FCMIN*10+ERORFC): ERORFC=10*FC+ERORFC-FCN%*FCMIN
6706 '
6707 ' ------------- Keep track of volume fed & desired ---------------
6708 '
6709 IF FAN%>10 THEN VAD=VAD+FA*20 ELSE VAD=VAD+FAMIN*FAN%*2
6710 IF FBN%>10 THEN VBD=VBD+FB*20 ELSE VBD=VBD+FBMIN*FBN%*2
6711 IF FCN%>10 THEN VCN=VCN+FC*20 ELSE VCD=VCD+FCMIN*FCN%*2
6712 VADES=VADES+FA*20: VBDES=VBDES+FB*20: VCDES=VCDES+FC*20
6715 LPRINT "*** VAD, VBD, VCD ***";VAD,VBD,VCD
6717 LPRINT "*** VADES, VBDES, VCDES ***";VADES,VBDES,VCDES
6720 '
6730 IF FAN%>10 THEN  FAF=FA ELSE FAF=FAMIN
6735 IF FBN%>10 THEN  FBF=FB ELSE FBF=FBMIN
6737 IF FCN%>10 THEN  FCF=FC ELSE FCF=FCMIN
6750 '
```

```
6760 IA= IAMIN + SLA*(FAF-FAMIN)
6770 IB= IBMIN + SLB*(FBF-FBMIN)
6775 IC= ICMIN + SLC*(FCF-FCMIN)
6780 IO= IOMIN + SLO*(FO-FOMIN)
6790 '
6800  VALA%=(IA-OOFF) / OSLOPE  +OADFF :'Calculate binary output values
6802  VALB%=(IB-OOFF) / OSLOPE  +OADFF
6803  VALC%=(IC-OOFF) / OSLOPE + OADFF
6804  VALO%=(IO-OOFF) / OSLOPE  +OADFF
6805 ' ----------- Intermittent Flow -----------
6806  IF FGC%>FAN% THEN VALA%=0           :' Apply Intermittent Flow
6807  IF FGC%>FBN% THEN VALB%=0
6808  IF FGC%>FCN% THEN VALC%=0
6809 '
6810 ' --------------------- Output statements ----------------------
6811 '
6814 PRINT AOCHA%,VALA%,AOPT$,AOCHB%,VALB%,AOCHO%,VALO%
6815 GOSUB 15000
6816  CALL AOUS (AOCHA%,VALA%,AOPT$)   :' Call ISAAC'S output routine
6818  CALL AOUS (AOCHB%,VALB%,AOPT$)
6819  CALL AOUS (AOCHC%,VALC%,AOPT$)
6820  CALL AOUS (AOCHO%,VALO%,AOPT$)
6822 '
6830 IF FGC% > 1 THEN 6927
6840 '
6900 OPEN "RECOVER.DAT" FOR OUTPUT AS 2    :' Write in recovery - file
6905 PRINT #2,TSAMPLE%,OFLE$
6910 PRINT #2,SA,SB,SC,ZA,ZB,ZC
6915 PRINT #2,MU,SF,ZF
6920 PRINT #2,XEXP#,DILR
6922 PRINT #2,FA,FB,FC,FO
6925 CLOSE #2
6926 '
6927  CALL AINM(AICH1%,COUNT,HZ,RAW1%(0),OPT$)
6933  CALL AINM (AICH2%, COUNT,HZ,RAW2%(0),OPT$)
6934 SUM1=0:SUM2=0  :FOR IR%=1 TO COUNT
6935 SUM1 =SUM1 +RAW1%(IR%-1) :SUM2=SUM2 +RAW2%(IR%-1)
6936 NEXT IR%
6937 VTM%=SUM1/COUNT : VPO%=SUM2/COUNT
6938 '
6942  RPO =  POO + ( VPO% - IADFF ) *ISLOPE *POSL
6944  RTM =  TMO + ( VTM% - IADFF ) *ISLOPE *TMSL :RTM=RTM - 1.5
6945 '
6947 FGAL%(1)=0:FGAL%(2)=0:FGAL%(3)=0:FGAL%(4)=0
6949 IF RPO<PO1 THEN FGAL%(1)=1 ELSE IF RPO>PO2 THEN FGAL%(2)=1
6951 IF RTM<TM1 THEN FGAL%(3)=1 ELSE IF RTM>TM2 THEN FGAL%(4)=1
6952 SUMAL%=0:FOR IR%=1 TO 4:SUMAL=SUMAL+FGAL%(IR%):NEXT IR%
6954 IF FGC% >1 THEN 6975
6956 '
6960 '
6962 '
6968 CLOSE #3                          :' Write in plot - file
```

```
6970 OPEN "plot1.dat" FOR APPEND AS 3
6973 PRINT #3,D#;SFA;Y(4);Y(5);Y(6);Y(7);Y(5)-Y(4);Y(7)-Y(6);RPO;RTM
6974 LPRINT " ON LINE : PO2 ,TEMP  :";RPO;"   ";RTM
6975 '
6980 CLOSE #3 : OPEN OFLE$ FOR APPEND AS 3 :'sAVE DATA
6982 PRINT #3,DR;" ";FA;" ";FB;" ";FO;" ";FAN%;" ";FBN%;" ";FON%;" ";
SA;" ";SP;" ";SFA;" ";RPO;" ";RTM
6984 CLOSE #3
6990 TROFF:PLAY MP$(3):CLS
6991 IF FGINT$="on" OR FGK$="on" THEN RETURN ELSE RETURN 2080
6992 '
7000 ' ------------------------------------------------------------
7010 '        <F9> Enter Measurements & Perform Control
7020 ' ------------------------------------------------------------
7030 '
7040 FGK$="on" : CLS
7050 LOCATE 7,1:INPUT "Enter the time of sample (hrs)";TSAMP
7060 LOCATE 9,1:PRINT "The inputted sample time is ...";TSAMP
7070 LOCATE 11,1: INPUT "Is this correct? (y/n)";FF$
7080 IF FF$="y" OR FF$="Y" THEN 7090 ELSE 7050
7090 CLS: LOCATE 7,1:INPUT "Enter current biomass (g/L)";BMASSO
7100 LOCATE 9,1:PRINT "The inputted biomass is ...";BMASSO
7110 LOCATE 11,1:INPUT "Is this correct? (y/n)";GG$
7120 IF GG$="y" OR GG$="Y" THEN 7130 ELSE 7090
7130 YM=LOG(BMASSO)
7140 CLS: LOCATE 7,1: INPUT "Enter penicillin concentration (g/L)";PENM
7150 LOCATE 9,1: PRINT "The entered concentration is ...";PENM
7160 LOCATE 11,1: INPUT "Is this correct? (y/n)";HH$
7170 IF HH$="y" OR HH$="Y" THEN 7180 ELSE 7140
7180 CLS: LOCATE 7,1: INPUT "Enter precursor concentration (g/L)";PAAM
7190 LOCATE 9,1: PRINT "The entered concentration is ...";PAAM
7200 LOCATE 11,1: INPUT "Is this correct? (y/n)";II$
7210 IF II$="y" OR II$="Y" THEN 7220 ELSE 7180
7220 '
7230 ' ------------------------------------------------------------
7240 '        Calculate growth rate using a Kalman filter
7250 ' ------------------------------------------------------------
7260 '
7265 TTT=TSAMP-OFFTIME
7270 YB=MUO*TTT+CO
7280 D=LAM + P11*TTTO^2 + 2*P12*TTTO + P22
7290 K1=(P11*TTTO + P12)/D
7300 K2=(P12*TTTO + P22)/D
7310 MU=MUO + K1*(YM - YB)
7320 C=CO + K2*(YM - YB)
7330 P11=1/LAM*(P11 - K1*K1*D)
7340 P12=1/LAM*(P12 - K1*K2*D)
7350 P22=1/LAM*(P22 - K2*K2*D)
7360 CO=C: MUO=MU: TTTO=TTT
7370 '
```

```
7380 ' ------------------------------------------------------
7390 '     Calculate errors and take control action
7400 ' ------------------------------------------------------
7410 '
7420 ERP=PDES-PENM: ERMU=MUDES-MU: ERZ=ZDES-PAAM
7430 DILR=DILR + KCP*(1 +(TSAMP-TSAMPO)/TIP)*ERP - KCP*ERPO
7450 ERPO=ERP
7460 SIGH=MU/YG+MAINT
7470 SIGC=SIGH + KCS*(1 +(TSAMP-TSAMPO)/TIS)*ERMU - KCS*ERMUO
7472 TSAMPO=TSAMP
7475 ERMUO=ERMU
7477 IF SIGC < SIGMIN THEN SIGC=SIGMIN
7478 IF SIGC > SIGMAX THEN SIGC=SIGMAX
7480 DMINS=SIGC*BMASSO/(SFMAX-SDES)
7485 IF DILR < DMIN THEN DILR=DMIN
7490 IF DMINS < DMIN THEN DMINS=DMIN
7505 IF DILR > DMAX THEN DILR=DMAX
7510 SF = SIGC*BMASSO/DILR + SDES
7520 IF SF < SFMAX THEN 7530
7525 IF DILR < DMINS THEN DILR=DMINS
7527 SF = SIGC*BMASSO/DILR + SDES
7530 IF SF < SFMIN THEN SF=SFMIN
7540 QC=QPMAX + KCZ*ERZ
7550 ZF=ZDES + QC*B*BMASSO/DILR
7560 IF ZF > ZFMAX THEN ZF=ZFMAX
7570 IF ZF < ZFMIN THEN ZF=ZFMIN
7580 LPRINT "SAMPLE TIME, BIOMASS, PEN-V, PRECURSOR"
7590 LPRINT USING "###.####";TSAMP;BMASSO;PENM;PAAM
7600 LPRINT "MEASURED MU, SIGH, SIGC, QC, DILR, SF, ZF"
7610 LPRINT USING "###.#####";MU;SIGH;SIGC;QC;DILR;SF;ZF
7620 LPRINT " "
7800 CLS: FGK$="off"
7810 RETURN 2080
7820 '
10990 '---------------- Error handling subroutine --------------------
11000 '
11010 IF (ERR=24)OR (ERR=25 ) OR (ERR=26) THEN FGER$="on":RESUME NEXT
11015 IF 5460<=ERL AND 6080>ERL THEN FGER$="on":RESUME 2080
11020 IF 7000<=ERL AND 8000>ERL THEN FGER$="on":RESUME 2080
11030 IF 55000!<= ERL AND 56000!>ERL THEN FGER$="on":RESUME 2080
11040 IF 56000!<=ERL AND 57200!>ERL THEN FGER$="on" :RESUME 2080
11050 IF ERL>57200 THEN FGER$="on" :RESUME 2080
11060 PRINT ERR,ERL:RESUME NEXT
12000 '
12005 ' ------------------- DISPLAY SUBROUTINE ---------------------
12007 '
12010 CLS
12030 LOCATE 1,1:FOR I=1 TO 80:PRINT CHR$(196);:NEXT I
12050 FOR I=2 TO 17 :LOCATE I,1:PRINT CHR$(179)
12070 LOCATE I,80:PRINT CHR$(179):NEXT I
12090 LOCATE 3,1:FOR I=1 TO 80 :PRINT CHR$(196);:NEXT I
12110 LOCATE 2,19:PRINT "        "PHASE$
```

```
12130 LOCATE M,N-2:FOR I=1 TO 28:PRINT CHR$(205);
:NEXT I:PRINT CHR$(187)
12150 FOR I=1 TO 1:LOCATE M+I,N+26: PRINT CHR$(186):NEXT I
12190 LOCATE M,35+N:PRINT CHR$(201);:FOR I=1 TO 36:
PRINT CHR$(205);:NEXT I
12210 FOR I=1 TO 1:LOCATE M+I,N+35:PRINT CHR$(186):NEXT I
12230 LOCATE M+1+1,N+35:PRINT CHR$(25)
12250 MT=M+1+1 :FOR I=1 TO 8 :LOCATE  MT+(I-1),27:PRINT CHR$(221)
12270 LOCATE  MT+(I-1),27+20-1:PRINT CHR$(222):NEXT I
12290 FOR I=1 TO 18 :LOCATE MT+1,27+I:PRINT CHR$(247):NEXT I
12310 LOCATE MT+8-1,27+20:FOR I=1 TO 24:PRINT CHR$(205);:NEXT I
12330 PRINT CHR$(187):LOCATE MT+8+1,27+20+24:PRINT CHR$(25)
12350 FOR I=1 TO 1:LOCATE MT+8-1+I,27+20+24:PRINT CHR$(186):NEXT I
12370 LOCATE MT+8,27:FOR I=1 TO 20:PRINT CHR$(223);:NEXT I
12390 LOCATE MT+8+1,2:FOR I=1 TO 34:PRINT CHR$(205);:NEXT I
12410 PRINT CHR$(188):LOCATE MT+8-1,36:PRINT CHR$(186)
12430 LOCATE MT+8-2,36:PRINT CHR$(24)
12450 LOCATE M,N:PRINT  "Sol. A : GL (       g/L)"
12460 LOCATE M+1,N:PRINT  "         PAA (       g/L)"
12470 LOCATE M,44+N:PRINT "Sol. B: PAA (       g/L)"
12490 LOCATE M+3,N :PRINT "Flow =        ml/min "
12510 LOCATE M+4,N :PRINT "% of total =      "
12530 LOCATE M+1,45+N:PRINT "          ml/min ,       %"
12550 LOCATE MT+6,47:PRINT "          ml/min ,      %"
12570 LOCATE MT+6,2:PRINT "Sol. C: water          "
12590 LOCATE MT+7,2:PRINT "Flow =         ml/min "
12600 LOCATE MT+8,2:PRINT "% of total "
12670 LOCATE MT+2,31: PRINT "   T     PO  "
12770 LOCATE 17,1:FOR I=1 TO 80:PRINT CHR$(176);:NEXT I
12775 LOCATE 17,3:PRINT "PAA(G/L)":LOCATE 17,19:PRINT "PEN_G"
12780 LOCATE 17,35:PRINT "GLUCOSE":LOCATE 17,50:PRINT "Biomass"
12785 LOCATE 17,66:PRINT "GR. RATE"
12810 LOCATE 18,1:FOR I=1 TO 80:PRINT CHR$(196);:NEXT I
12850 LOCATE 20,1:FOR I=1 TO 80:PRINT CHR$(220);:NEXT I
12860 LOCATE 18,26:PRINT " LATEST..       .. DATA "
12865 LOCATE 20,26:PRINT " SAMPLE..       .. TIME "
12890 LOCATE 25,31 : PRINT "Sampling Interval ";" 20"; " min";
13000 '
13010 LOCATE M,N+14:PRINT USING "###.#" ;SA
13015 LOCATE M+1,N+14:PRINT USING "##.##" ;ZA
13020 LOCATE M,58+N:PRINT USING "##.##";ZB
13030 LOCATE M+3,N+6:PRINT USING "##.####"; FA
13050 LOCATE M+1,46+N:PRINT USING "##.####";FB
13055 IF (FA+FB)<1E-08 THEN FTOT =1 ELSE FTOT =FA+FB
13057 LOCATE M+4,N+13:PRINT USING "###.#"; FA/FO*100!
13060 LOCATE M+1,62+N:PRINT USING "###.#";FB/FO*100!
13070 LOCATE MT+6,48:PRINT USING "##.####";FO
13080 LOCATE MT+6,66:PRINT USING "###.#";FO/FO*100!
13090 LOCATE MT+7,9:PRINT USING "##.####";FC
13100 LOCATE MT+8,12:PRINT USING "###.#";FC/FO*100!
13150 LOCATE 19,1:PRINT USING " ##.####       "
;PAAM,PENM,SDES,BMASSO,MU
```

```
13160 LOCATE 21,1:PRINT USING "  ###.#         "
;TSAMP,TSAMP,TSAMP,TSAMP,TSAMP
13170 RETURN
13180 '
13200 ' ------------------------------------------------------------
14000 '               <F6>   REARRANGE THE PUMPS
14001 ' ------------------------------------------------------------
14002 '
14010 CLS :FGK$="on":LOCATE 10,10 :PRINT "PRESENT ARRANGEMENT :"
14020 LOCATE 12,10  :PRINT "Glucose + PAA          Pump ";PA%
14040 LOCATE 14,10  :PRINT "Maximum PAA only       Pump ";PB%
14050 LOCATE 16,10  :PRINT "Water only             Pump ";PC%
14055 LOCATE 18,10  :PRINT "Outlet                 Pump ";PO%
14060 LOCATE 20,10  :INPUT "New Arrangement (pa,pb,pc,po) "
;PA%,PB%,PC%,PO%
14070 IAMIN = IMIN(PA%) :OFFA =OFFP(PA%) :SLA =SLP(PA%)
14080 IBMIN = IMIN(PB%) :OFFB =OFFP(PB%) :SLB =SLP(PB%)
14085 ICMIN = IMIN(PC%) :OFFC =OFFP(PC%) ;SLC =SLP(PC%)
14090 IOMIN = IMIN(PO%) :OFFO =OFFP(PO%) :SLO =SLP(PO%)
14095 FAMAX = (20-OFFA)/SLA: FBMAX = (20-OFFB)/SLB
14097 FCMAX = (20-OFFC)/SLC: FOMAX = (20-OFFO)/SLO
14100 FAMIN = (IAMIN-OFFA)/SLA: FBMIN = (IBMIN-OFFB)/SLB
14105 FCMIN = (ICMIN-OFFC)/SLC: FOMIN = (IOMIN-OFFO)/SLO
14110 LOCATE 20,10 :INPUT "O.K.! Now Press <CR> to continue",A$
14115 LPRINT " ### PUMP ARRANGEMENT : A , B , C , O : "
;PA%;"   ";PB%;"   ";PC%;"   ";PO%
14120 CLS:FGK$="off":RETURN 2080
14130 '
14140 ' ---------------- Check output voltage to pumps -------------
14150 '
15000 IF VALA%>OADIN THEN VALA%=OADIN
15005 IF VALA%<OADFF THEN VALA%=OADFF
15010 IF VALB%<OADFF THEN VALB%=OADFF
15015 IF VALB%>OADIN THEN VALB%=OADIN
15017 IF VALC%<OADFF THEN VALC%=OADFF
15018 IF VALC%>OADIN THEN VALC%=OADIN
15020 IF VALO%<OADFF THEN VALO%=OADFF
15025 IF VALO%>OADIN THEN VALO%=OADIN
15030 RETURN
15032 '
15035 ' ------------------------------------------------------------
55000 '              <F8>  PUMP FLOW CONTROLLER ROUTINE
55001 ' ------------------------------------------------------------
55002 '
55005 FGK$="on"
55010 CLS: LOCATE 7,1: PRINT "Servicing <F8> Pump Flow Controller"
55015 LOCATE 9,1: PRINT "    TO UPDATE                  ENTER"
55020 LOCATE 10,1:PRINT "        FA                       1"
55022 LOCATE 12,1:PRINT "        FB                       2"
55024 LOCATE 14,1:PRINT "        FC                       3"
55026 LOCATE 16,1:PRINT "        FO                       4"
55035 LOCATE 20,1: INPUT "Well .....";ZZ%
```

```
55040 IF ZZ%<0 OR ZZ%>4 THEN 55010
55045 ON ZZ% GOTO 55100,55200,55300,55400
55055 '
55100 CLS: LOCATE 7,1: INPUT "Enter the measured flow rate (FA) ";FAM
55110 LOCATE 9,1: INPUT "Is the entered value correct? (y/n)";XA$
55120 IF XA$="y" OR XA$="Y" THEN 55130 ELSE 55100
55130 ERFA = FA - FAM
55140 SLA = SLA + KCF*ERF*SLA
55150 GOTO 55500
55155 '
55160 '
55200 CLS: LOCATE 7,1: INPUT "Enter the measured flow rate (FB) ";FBM
55210 LOCATE 9,1: INPUT "Is the entered value correct? (y/n)";XB$
55220 IF XB$="y" OR XB$="Y" THEN 55230 ELSE 55200
55230 ERFB = FB - FBM
55240 SLB = SLB + KCF*ERF*SLB
55250 GOTO 55500
55260 '
55300 CLS: LOCATE 7,1: INPUT "Enter the measured flow rate (FC) ";FCM
55310 LOCATE 9,1: INPUT "Is the entered value correct? (y/n)";ZD$
55320 IF ZD$="y" OR ZD$="Y" THEN 55330 ELSE 55300
55330 ERFC = FC - FCM
55340 SLC = SLC + KCF*ERF*SLC
55350 GOTO 55500
55360 '
55400 CLS: LOCATE 7,1: INPUT "Enter the measured flow rate (FO) ";FOM
55410 LOCATE 9,1: INPUT "Is the entered value correct? (y/n)";ZE$
55420 IF ZE$="y" OR ZE$="Y" THEN 55430 ELSE 55400
55430 ERFO = FO - FOM
55440 SLO = SLO + KCF*ERFO*SLO
55450 GOTO 55500
55460 '
55500 LOCATE 11,1: INPUT "Do you wish to enter another value?(y/n)";ZF$
55510 IF ZF$="y" OR ZF$="Y" THEN 55010 ELSE 55520
55520 FGK$="off": RETURN 2080
55530 '
55580 ' ------------------------------------------------------------
56000 '                    <F7>   CHANGE PARAMETERS
56001 ' ------------------------------------------------------------
56005 '
56010 FGK$="on"
56100 CLS:LOCATE 5,1
56110 PRINT "       TO   UPDATE                                    ENTER"
56115 PRINT:PRINT:PRINT CHNG$(0);"                                    1"
56120 PRINT:PRINT CHNG$(1);"                                  2"
56125 PRINT:PRINT CHNG$(2);"                        3"
56130 PRINT:PRINT CHNG$(3);"                                  4"
56135 LOCATE 17,1:PRINT "ENTER 0 to return to display...."
56140 LOCATE 18,45:INPUT "Well .........";CP%
56145 IF CP%<0 OR CP%>4 THEN 56100 ELSE IF CP%=0 THEN 56560
56146 CLS :LOCATE 25,10 :PRINT "CHANGING ......";CHNG$(CP%-1);
56150 ON CP% GOTO 56200,56300,56400,56500
```

```
56200 '
56205 LOCATE 5,1 :PRINT "Sugar Concentrations (SA,SB,SC) ";SA,SB,SC
56210 LOCATE 7,1 :INPUT " New Values ";SA,SB,SC
56215 LOCATE 9,1 :PRINT "PAA Concentrations (ZA,ZB,ZC) ";ZA,ZB,ZC
56220 LOCATE 11,1 :INPUT " New Values ";ZA,ZB,ZC
56223 LOCATE 13,1 :INPUT "CSL Concentration (CSLA, ml/L) ";CSLA
56225 LOCATE 15,1: INPUT "Are these  values correct? (y/n)";AA$
56230 IF AA$="Y" OR AA$="y" THEN 56240 ELSE 56205
56240 LPRINT " ### PARAMETERS SA, SB, SC : ";SA;"  ";SB;"  ";SC
56245 LPRINT " ### PARAMETERS ZA, ZB, ZC : ";ZA;"  ";ZB;"  ";ZC
56247 LPRINT " ### PARAMETER CSLA : ";CSLA: GOTO 56100
56250 '
56300 LOCATE 5,1 :PRINT "old PAA controller gain = ";KCZ:KCZ0=KCZ
56305 LOCATE 5,40:INPUT "new value ";KCZ
56310 IF KCZ<.000001 THEN KCZ=KCZ0
56320 LOCATE 7,1: PRINT "old Pen-V contrtoller gain = ";KCP:KCPO=KCP
56330 LOCATE 7,40: INPUT "new value ";KCP
56340 IF KCP<0! THEN KCP = KCPO
56342 LOCATE 9,1: PRINT "Integral time constant = ";TIP:TIPO=TIP
56344 LOCATE 9,40: INPUT "new value ";TIP
56346 IF TIP<0! OR TIP=0! THEN TIP=TIPO
56350 LOCATE 11,1: PRINT "growth rate controller gain = ";KCS:KCSO=KCS
56360 LOCATE 11,40: INPUT "new value ";KCS
56370 IF KCS < .1  THEN KCS = KCSO
56380 LOCATE 13,1: INPUT "Are these control constants correct? (y/n)"
;BB$
56382 IF BB$="Y" OR BB$="y" THEN 56390 ELSE 56300
56390 LPRINT " ### PARAMETERS KCP, KCS, KCZ, TIP :";KCP;"  ";KCS;"  ";
KCZ;"  ";    TIP: GOTO 56100
56395 '
56400 LOCATE 10,10:INPUT"input parameters (1) or output (2)";CP%
56402 IF CP%<1 OR CP%>2 THEN 56400 ELSE ON CP% GOTO 56409,56452
56409 CLS:LOCATE 5,1 :PRINT "count = ";COUNT;" hz = ";HZ:VAL0=COUNT:
VSL=HZ
56410 LOCATE 7,1 :INPUT "new values ";COUNT,HZ
56412 IF COUNT<.1 AND HZ<.1 THEN COUNT=VAL0:HZ=VSL
56413 LPRINT " ### PARAMETERS COUNT , HZ :  ";COUNT;"    ";HZ
56415 LOCATE 8,1 :PRINT "OTHER CHANGES (y/n) "
56420 A$=INKEY$:IF A$="" THEN 56420 ELSE IF A$="n" OR A$="N" THEN 56100
56440 CLS:LOCATE 10,1 :PRINT"PO2 input parameters  "PO0;POSL:VAL0=PO0
:VSL=POSL
56442 LOCATE 12,1 :INPUT "New values for offset,slope";PO0,POSL
56444 IF ABS(PO0)<.000001 AND POSL<.000001 THEN PO0=VAL0:POSL=VSL
56445 LOCATE 15,1 :PRINT"Temp. input parameters   ";TM0,TMSL:
VAL0=TM0:VSL=TMSL
56447 LOCATE 17,1 :INPUT  "New values for zero,offset";TM0,TMSL
56448 IF ABS(TM0)<.000001 AND TMSL<.001 THEN TM0=VAL0:TMSL=VSL
56449 LPRINT " ### PARAMETERS PO0 ,PSL ,  TM0 , TMSL :";PO0;"  ";
PSL;"  ";TM0;"  ";TMSL
56450 LOCATE 19,1:PRINT "CHANGES IN PUMP PARAMETERS ? (y/n)"
56451 A$=INKEY$:IF A$="" THEN 56451 ELSE IF A$="n" OR A$="N" THEN 56100
56452 CLS :LOCATE 5,1:PRINT "PUMP A :";"offa = ";OFFA;" mL/min ,
```

```
Slope = ";SLA
56455 VAL0=OFFA:VSL=SLA:LOCATE 7,1 :INPUT"NEW values";OFFA,SLA
56457 IF OFFA< 4 THEN OFFA=VAL0:SLA=VSL
56460 LOCATE 10,1 :PRINT    "PUMP B :";"OFFB = ";OFFB;" ml/min,
 Slope = ";SLB
56462 VAL0=OFFB:VSL=SLB:LOCATE 12,1:INPUT "NEW values";OFFB,SLB
56464 IF OFFB< 4 THEN OFFB=VAL0:SLB=VSL
56466 LOCATE 15,1 :PRINT    "PUMP C :";"OFFC = ";OFFC;" ml/min,
Slope = ";SLC
56468 VAL0=OFFC:VSL=SLC:LOCATE 17,1:INPUT "NEW values";OFFC,SLC
56470 IF OFFC < 4 THEN OFFC=VAL0:SLC=VSL
56472 LOCATE 20,1: PRINT "PUMP O :";"OFFO = ";OFFO;" ml/min,
 Slope = ";SLO
56474 VAL0=OFFO:VSL=SLO:LOCATE 22,1:INPUT "NEW values";OFFO,SLO
56476 IF OFFO < 4 THEN OFFO=VAL0:SLO=VSL
56480 LPRINT " ### PARAMETERS ,PUMPS : ";OFFA;"  ";SLA;"    ";
OFFB;"  ";SLB;"    ";OFFO ;"  ";SLO:GOTO 56100
56500 '
56510 LOCATE 5,1 :INPUT "OFF LINE MEASUREMENT ALARMS (y/n)";CON$
56513 IF CON$="" THEN 56516 ELSE IF CON$="n" OR CON$="N" THEN 56530
56516 LOCATE 7,1 :INPUT "PAA  CONC.  ALARMS ... Low , High";AL1,AL2
56518 IF AL2>.1 THEN PAA1=AL1:PAA2=AL2
56520 LOCATE 9,1 :INPUT "Glucose conc.                    ";AL1,AL2
56522 IF AL1 >.1 OR AL2>0 THEN GL1=AL1:GL2=AL2
56524 LPRINT " ### PARAMETERS OFF-ALARMS : ";PAA1;" ";PAA2;"  ";
GL1;" ";GL2
56526 LOCATE 12,1 :PRINT "CONTINUE TO ON-LINE ALARMS (y/n)"
56527 A$=INKEY$:IF A$="" THEN 56527 ELSE IF A$="n" OR A$="N" THEN 56560
56530 CLS:LOCATE 3,15 :PRINT "ON LINE MEASUREMENT ALARMS"
56536 LOCATE 7,1 :INPUT "PO ALARMS (low,high)    ";AL1,AL2
56539 IF AL1>15 AND AL2>15 THEN PO1=AL1:PO2 = AL2
56542 LOCATE 9,1 :INPUT "Temp. ALARMS (low,high)    ";AL1,AL2
56545 IF AL1>15 AND AL2 >10 THEN TM1= AL1 :TM2=AL2
56550 LPRINT "### PARAMETERS ON ALARMS: ";PO1;" ";PO2;" ";TM1;" ";TM2:
GOTO 56100
56555 '
56560 CLS:LOCATE 5,1:INPUT "Change phase (y/n) ?";JOHN$
56570 IF JOHN$="" THEN 56570 ELSE IF JOHN$="n" OR JOHN$="N" THEN 56800
56575 LOCATE 7,1 :INPUT "Growth (1) or production (2) ?";MM$
56580 IF MM$="" THEN 56580 ELSE IF MM$<>"1" AND MM$<>"2" THEN 56560
56590 IF A$="1" THEN PHASE$=" GROWTH PHASE ":PHASE%=1:GOTO 56610
56600 PHASE$=" PRODUCTION PHASE ":PHASE%=2
56610 LOCATE 12,1 :INPUT "O.K !  Now  Press <CR> to return ",A$
56800 CLS:FGK$="off":IF ISML%<1  THEN RETURN ELSE RETURN 2080
57100 '
57200 ' ------------------------------------------------------------
57300 '                        <F5>  GRAPHS
57400 ' ------------------------------------------------------------
57500 '
58000 FGK$="on" :CLS :LOCATE 5,1
58003 PRINT "press            1       for   ";BC$(1)
58006 FOR IP%=2 TO 11:
```

```
58009 PRINT "                 "; IP% ;"             ";BC$(IP%)
58012 NEXT IP%
58045 PRINT "                  0      TO EXIT"
58048 PRINT :INPUT "Well....";PLOTIN%:IPL%=0:IPL1%=0
58051 IF PLOTIN%>0 THEN  58057
58054 FGK$="off":CLS:RETURN 2080
58057 CLOSE #3: OPEN "plot1.dat" FOR INPUT AS 3
58060 INPUT "Start from interval No :";IPST%:INPUT "Stop at No";IPEN%
58061 IF IPEN%<IPST% THEN BEEP: GOTO 58060
58063 IF IPST%=0 THEN IPST%=1
58064 IF IPEN%=0 THEN IPEN%=ISML%
58066 IPL%=IPL%+1:IF EOF(3) THEN 58093
58069 IF IPL%<IPST% OR IPL%>IPEN% THEN IPL1%=0 ELSE IPL1%=IPL1% +1
58072 FOR IP%=1 TO 10
58075 IF IP%=PLOTIN% THEN INPUT #3,PLV(IPL%):GOTO 58081
58078 INPUT #3,NL
58081 NEXT IP%
58084 'IF PLOTIN%=11 THEN INPUT #3,PLV(IPL%) ELSE INPUT #3,NL
58090 GOTO 58066
58093 CLOSE #3: TITLE$=BC$(PLOTIN%)
58096 MINPL=PLV(1):MAXPL=PLV(1)
58099 FOR IP%=1 TO 1+(IPEN%-IPST%)
58102 IF PLV(IP%) <MINPL THEN MINPL=PLV(IP%)
58105 IF PLV(IP%) >MAXPL THEN MAXPL=PLV(IP%)
58108 NEXT IP%
58111 PRINT "RANGE of";BC$(PLOTIN%),MINPL;"  to  " ;MAXPL
58114 INPUT "highest value";HH:IF HH<0 THEN GOTO 58054
58117 INPUT "lowest value  ";LL
58120 IF HH=0 THEN HH=MAXPL
58123 IF LL=0 THEN LL=MINPL
58126 IF ABS((HH-LL))<=.001*LL THEN CLS:GOTO 58114
58129 GOSUB 58153
58132 FOR II%= IPST% TO IPEN%
58135 IF PLV(II%)<LL OR PLV(II%) >HH  THEN 58138 ELSE 58141
58138 LOCATE 23,1:PRINT "data out of range":GOTO 58147
58141 VALUE=(PLV(II%)-LL)/(HH-LL)*4095
58144 GOSUB 58471
58147 NEXT II%
58150 IF INKEY$="" THEN 58150 ELSE IF INKEY$="a" OR INKEY$="A"
THEN 58111
58151 CLS: SCREEN 0,0,0 :WIDTH 80 :LOCATE 5,1:GOTO 58003
58153 ' name        INITGRAPH
58156 BASE.GRF%           = 0
58159 I.GRF%              = 0
58162 INPTR.GRF%          = 0
58165 J.GRF%              = 0
58168 NWAVES.GRF%         = 1
58171 XVALUE.GRF          = 0
58174 YVALUE.GRF          = 0
58177 SCREEN 1 : VIEW : WINDOW
58180 KEY OFF
58183 COLOR BACKGROUND%,PLETE%
```

```
58186 CLS
58189 FOR I.GRF% = 0 TO 3
58192    FORMAT.GRF%(I.GRF%) = FORMAT%(I.GRF%)
58195 NEXT I.GRF%
58198 WHILE (NWAVES.GRF% < 4) AND FORMAT.GRF%(NWAVES.GRF%) <> -1
58201    NWAVES.GRF% = NWAVES.GRF% + 1
58204 WEND
58207 ON (TYPE% + 1) GOSUB 58267,58267,58267,58312,58312,58312
58210 IF LEN(TITLE$) > 30 THEN TITLE$ = MID$(TITLE$,1,30)
58213 LOCATE 1,(20 - LEN(TITLE$) \ 2)
58216 PRINT TITLE$;
58219 IF LEN(XLABEL$) > 37 THEN XLABEL$ = MID$(XLABEL$,1,37)
58222 LOCATE 21,(21-LEN(XLABEL$) \ 2)
58225 PRINT XLABEL$;
58228 IF LEN(YLABEL$) > 18 THEN YLABEL$ = MID$(YLABEL$,1,18)
58231 FOR I.GRF% = 0 TO LEN(YLABEL$)
58234    LOCATE (11 - (LEN(YLABEL$) \ 2) + I.GRF%),1
58237    PRINT MID$(YLABEL$,(I.GRF% + 1),1)
58240 NEXT I.GRF%
58243 IF LEN(COMMENT$) > 78 THEN COMMENT$ = MID$(COMMENT$,1,78)
58246 FOR I.GRF% = 1 TO LEN(COMMENT$)
58249    IF I.GRF% <= 39 THEN LOCATE 23,I.GRF% ELSE LOCATE 24,I.GRF%-39
58252    PRINT MID$(COMMENT$,I.GRF%,1);
58255 NEXT I.GRF%
58258 WINDOW (0,LOVAL) - (291,HIVAL)
58261 IF TYPE%>2 THEN VIEW (17,16)-(308,79) ELSE VIEW (17,16)-(308,143)
58264 RETURN
58267 ' *** SUBROUTINE TO DRAW AXES AND TIC MARKS FOR SCR. GRAPH
58270 LINE (16,15)-(310,144),,B
58273 FOR I.GRF% = 26 TO 310 STEP 10
58276    LINE (I.GRF%,12) - (I.GRF%,15)
58279    LINE (I.GRF%,147) - (I.GRF%,144)
58282 NEXT I.GRF%
58285 FOR I.GRF% = 116 TO 310 STEP 100
58288    LINE (I.GRF%,10) - (I.GRF%,15)
58291    LINE (I.GRF%,149) - (I.GRF%,144)
58294 NEXT I.GRF%
58297 FOR I.GRF% = 25 TO 144 STEP 10
58300    LINE (13,I.GRF%) - (16,I.GRF%)
58303    LINE (312,I.GRF%) - (310,I.GRF%)
58306 NEXT I.GRF%
58309 RETURN
58312 ' *** SUBROUTINE to draw axes and tic marks for alt. graph
58315 LINE (16,15) - (309,80),,B
58318 LINE (16,90) - (309,155),,B
58321 FOR I.GRF% = 26 TO 309 STEP 10
58324    LINE (I.GRF%,12) - (I.GRF%,15)
58327    LINE (I.GRF%,158) - (I.GRF%,155)
58330 NEXT I.GRF%
58333 FOR I.GRF% = 116 TO 309 STEP 100
58336    LINE (I.GRF%,10) - (I.GRF%,15)
58339    LINE (I.GRF%,160) - (I.GRF%,155)
```

```
58342 NEXT I.GRF%
58345 FOR I.GRF% = 70 TO 16 STEP -10
58348    LINE (13,I.GRF%) - (16,I.GRF%)
58351    LINE (312,I.GRF%) - (309,I.GRF%)
58354 NEXT I.GRF%
58357 FOR I.GRF% = 145 TO 90 STEP -10
58360    LINE (13,I.GRF%) - (16,I.GRF%)
58363    LINE (312,I.GRF%) - (309,I.GRF%)
58366 NEXT I.GRF%
58369 RETURN
58372 REM PAGE
58375 ' NAME      INITGRAPH_default
58378 LOVAL = 0
58381 HIVAL = 4095
58384 TYPE% = 1
58387 PLETE% = 1
58390 BACKGROUND% = 0
58393 FORMAT%(0) = 3
58396 FORMAT%(1) = -1
58399 FORMAT%(2) = 0
58402 FORMAT%(3) = 0
58405 FORMAT%(4) = 0
58408 XLABEL$ = ""
58411 YLABEL$ = ""
58414 COMMENT$ = ""
58417 TITLE$ = ""
58420 GOSUB 58153
58423 RETURN
58426 REM PAGE
58429 ' NAME      CLEARGRAPH
58432 ON (TYPE% + 1) GOSUB 58438,58438,58438,58447,58447,58447
58435 RETURN
58438 ' *** SUBROUTINE to erase  scrolling graph
58441 CLS
58444 RETURN
58447 ' *** SUBROUTINE TO ERASE  ALTERNAT. GRAPH
58450 VIEW (17,16) - (308,79)
58453 CLS
58456 VIEW (17,91) - (308,154)
58459 CLS
58462 IF BASE.GRF% = 0 THEN VIEW (17,16) - (308,79)
        ELSE VIEW (17,91) - (308,154)
58465 RETURN
58468 REM PAGE
58471 ' NAME      NEXTPOINT
58474 ON TYPE%+1 GOSUB 58480,58480,58480,58555,58555,58555
58477 RETURN
58480 ' *** SUBROUTINE for scrolling type plot
58483 IF INPTR.GRF% < 292 THEN GOTO 58528
58486   FOR I.GRF% = 0 TO NWAVES.GRF% - 2
58489           SWAP FORMAT.GRF%(I.GRF%),FORMAT.GRF%(I.GRF% + 1)
58492   NEXT I.GRF%
```

```
58495   FOR I.GRF% = 0 TO 290
58498           J.GRF% = I.GRF% MOD NWAVES.GRF%
58501           XVALUE.GRF = I.GRF%
58504           IF (I.GRF% MOD NWAVES.GRF%) = 0 THEN
LINE(XVALUE.GRF,HIVAL) -   ((XVALUE.GRF + NWAVES.GRF%),LOVAL),0,BF
58507           PLT.GRF(I.GRF%) = PLT.GRF(I.GRF% + 1)
58510           YVALUE.GRF = PLT.GRF(I.GRF%)
58513           PLOTCOLOR.GRF% = FORMAT.GRF%(J.GRF%)
58516           ON TYPE% + 1 GOSUB 58594,58603,58624
58519           INPTR.GRF% = 291
58522   NEXT I.GRF%
58525   J.GRF% = (J.GRF% + 1) MOD NWAVES.GRF%
58528 '
58531 PLT.GRF(INPTR.GRF%) = VALUE
58534 XVALUE.GRF = INPTR.GRF%
58537 YVALUE.GRF = PLT.GRF(INPTR.GRF%)
58540 PLOTCOLOR.GRF% = FORMAT.GRF%(J.GRF%)
58543 ON TYPE% + 1 GOSUB 58594,58603,58624
58546 J.GRF% = (J.GRF% + 1) MOD NWAVES.GRF%
58549 INPTR.GRF% = INPTR.GRF% + 1
58552 RETURN
58555 ' *** SUBROUTINE for alternating type plot
58558 YVALUE.GRF = VALUE
58561 PLOTCOLOR.GRF% = FORMAT.GRF%(J.GRF%)
58564 ON TYPE% - 2 GOSUB 58594,58603,58624
58567 XVALUE.GRF = XVALUE.GRF + 1
58570 IF XVALUE.GRF <= 291 THEN GOTO 58588
58573   XVALUE.GRF = 0
58576   IF BASE.GRF% = 0 THEN BASE.GRF% = 1 : VIEW (17,91) - (308,154)
          : CLS : GOTO 58588
58579   BASE.GRF% = 0
58582   VIEW (17,16) - (308,79)
58585   CLS
58588 J.GRF% = (J.GRF% + 1) MOD NWAVES.GRF%
58591 RETURN
58594 ' *** SUBROUTINE for dot plot
58597 PSET (XVALUE.GRF,YVALUE.GRF),PLOTCOLOR.GRF%
58600 RETURN
58603 ' *** SUBROUTINE for line plot
58606 IF XVALUE.GRF >= NWAVES.GRF% THEN GOTO 58615
58609   PSET (XVALUE.GRF,YVALUE.GRF),PLOTCOLOR.GRF%
58612   GOTO 58618
58615  LINE ((XVALUE.GRF - NWAVES.GRF%),LAST.GRF(J.GRF%)) -
         (XVALUE.GRF,YVALUE.GRF ),PLOTCOLOR.GRF%
58618 LAST.GRF(J.GRF%) = YVALUE.GRF
58621  RETURN
58624 '  ******* suBROUTINE FOR HISTOGRAM PLOT (TYPE%=2 OR 5).
58627 LINE (XVALUE.GRF,LOVAL)-(XVALUE.GRF,YVALUE.GRF) ,PLOTCOLOR.GRF%
58630 RETURN
58633 REM PAGE
```

```
58636 '******************** L A S T   P O I N T **********************
58639 PLOTCOLOR% = PLOTCOLOR.GRF%
58642 XVALUE =XVALUE.GRF
58645 YVALUE =YVALUE.GRF
58648 RETURN
58651 REM PAGE
```

APPENDIX C.3
ACSL: Unstructured Model, Open Loop

```
PROGRAM BIOEQ

  INITIAL
    CONSTANT MUMAX=0.123
    CONSTANT KS=1.0000
    CONSTANT YG=0.5000
    CONSTANT YP=1.20
    CONSTANT M=0.026
    CONSTANT B=0.30
    CONSTANT QPMAX=0.0017
    CONSTANT QH=0.00
    CONSTANT QPA=0.00
    CONSTANT A=0.0145
    CONSTANT SF=50
    CONSTANT ZF=1.0
    CONSTANT MUDES=0.015
    CONSTANT XO=4.000
    CONSTANT SO=0.000
    CONSTANT PO=0.000
    CONSTANT ZO=0.0000
    CONSTANT CC=1.0
    CONSTANT AGEO=8.1301
    QPO=QPMAX*A*AGEO*EXP(1-A*AGEO)
    SIGO=MUDES/YG+M+QPO/YP
    DILO=0.0045933
    CONSTANT TEND=150
    ALGORITHM IALG=2
      VARIABLE T=0.0
  END $ "OF INITIAL"

  DYNAMIC
    CINTERVAL CI=5.00
      DERIVATIVE BIOEQ
         DILR=DILO*EXP(MUDES*T)
         MU=MUMAX*S/(KS+S)
         DXDT=MU*X
         X=INTEG(DXDT,XO)
         SIG=MU/YG+M+QP/YP
         DSDT=-SIG*X+(SF-S)*DILR
         S=INTEG(DSDT,SO)
         DPDT=QP*X-P*DILR-QH*P
         P=INTEG(DPDT,PO)
         DZDT=-B*QP*X+DILR*(ZF-Z)-QPA*Z
         Z=INTEG(DZDT,ZO)
         DAGEDT=1.0-AGE*MU
         AGE=INTEG(DAGEDT,AGEO)
         QP=QPMAX*((A*AGE)**CC)*EXP(1.-(A*AGE)**CC)
```

```
      END $ "OF DERIVATIVE"

    TERMT (T .GE. TEND)
  END $ "OF DYNAMIC"

  TERMINAL
  END $ "OF TERMINAL"

END $ "OF PROGRAM"
```

APPENDIX C.4
ACSL: Unstructured Model, Closed Loop

```
PROGRAM BIOEQ

  INITIAL
    CONSTANT V=14.0
    CONSTANT MUMAX=0.123
    CONSTANT KS=1.0
    CONSTANT YG=0.500
    CONSTANT YP=1.20
    CONSTANT M=0.026
    CONSTANT B=0.30
    CONSTANT QPMAX=0.0017
    CONSTANT A=0.0145
    CONSTANT SA=200.0, SB=000.0, SC=0.0
    CONSTANT ZA=2.0, ZB=5.0, ZC=0.0
    CONSTANT SFMAX=200.0, SFMIN=0.0
    CONSTANT ZFMAX=5.0, ZFMIN=0.0
    CONSTANT MUDES=0.015
    CONSTANT XO=4.0000
    CONSTANT XMO=4.0
    CONSTANT PO=0.0
    CONSTANT ZO=0.0
    CONSTANT DMAX=0.02143, DMIN=0.0045933
    CONSTANT KCS=13.0, KCP=-0.44, KCZ=0.019, TIP=2.80, TIS=8.0
    CONSTANT PDES=4.0, ZDES=0.4, SDES=0.139
    CONSTANT LAM=0.75
    CONSTANT SO=0.0
    CO=ALOG(XO)
    TO=0.0
    P11=1.0
    P22=1.0
    P12=0.0
    ERPO=PDES-PO
    ERZO=ZDES-ZO
    MUO=MUDES
    ERMUO=MUDES-MUO
    DILR=DMIN
    DILC=DMIN
    DILO=DMIN
    SIGC=MUDES/YG+M
    SIGD=SIGC
    SIGA=MUDES/YG+M+QP/YP
    CTT=0.0
    AGEO=8.1301
    CONSTANT TEND=150.0
    ALGORITHM IALG=2
      VARIABLE T=0.0
  END $ "OF INITIAL"
```

```
DYNAMIC
  CINTERVAL CI=0.20

    DERIVATIVE MODEL
      PROCEDURAL(F,FA,FB,FC=SA,SB,SC,ZA,ZB,ZC,SF,ZF)
        F=V*1000*DILC/60
        FA=SF*F/SA
        FB=(ZF*F-ZA*FA)/ZB
        FC=F-FA-FB
      END $ "OF PROCEDURAL"
      PROCEDURAL(SIGC=T,CTT,SIGD)
        IF(T.GE.CTT) SIGC=SIGD
      END $ "OF PROCEDURAL"
      MU=MUMAX*S/(KS+S)
      DXDT=MU*X
      X=INTEG(DXDT,XO)
      XM=XMO*EXP(MUDES*(T-CTT))
      PROCEDURAL(DILC,SF=DMAX,SIGC,XM,DILO,SDES,SFMAX,SFMIN,T,CTT)
        IF(T.GE.CTT) DILC=DILR
        IF(T.GE.CTT) DILO=DILR
        SF=SIGC*XM/DILC+SDES
        IF (SF .GE. SFMAX) SF=SFMAX
        IF (SF .EQ. SFMAX) GOTO M3
        IF (SF .LE. SFMIN) SF=SFMIN
        GOTO M4
      M3..DILC=DILO*EXP(MUDES*(T-CTT))
        IF (DILC .GE. DMAX) DILC=DMAX
      M4..CONTINUE
      END $ "OF PROCEDURAL"
      SIGA=MU/YG+M+QP/YP
      DSDT=-SIGA*X+(SF-S)*DILC
      S=INTEG(DSDT,SO)
      DPDT=QP*X-P*DILC
      P=INTEG(DPDT,PO)
      DZDT=-B*QP*X+DILC*(ZF-Z)
      Z=INTEG(DZDT,ZO)
      DAGEDT=1.0-AGE*MU
      AGE=INTEG(DAGEDT,AGEO)
      QP=QPMAX*A*AGE*EXP(1.-A*AGE)
      Y=ALOG(X)
      PROCEDURAL(S,Z=S,Z)
        IF (S .LE. 0.0) S=0.0
        IF (Z .LE. 0.0) Z=0.0
      END $"OF PROCEDURAL"
      IAE=INTEG(ABS(ERMU),ERMUO)
    END $ "OF DERIVATIVE"

    DISCRETE CONTR
      PROCEDURAL
        INTERVAL SAMPLE=4.0
        IF(T .LT. 4.0) GO TO M1
```

```
            CTT=T+1.5
            XM=OU(10.0,X,0.3000)
            XMO=XM
            YM=ALOG(XM)
            ERP=PDES-P
            ERZ=ZDES-Z
            YB=MUO*T+CO
            D=LAM+P11*TO**2+2*P12*TO+P22
            K1=(P11*TO+P12)/D
            K2=(P12*TO+P22)/D
            MU=MUO+K1*(YM-YB)
            C=CO+K2*(YM-YB)
            P11=1/LAM*(P11-K1*K1*D)
            P12=1/LAM*(P12-K1*K2*D)
            P22=1/LAM*(P22-K2*K2*D)
            CO=C
            MUO=MU
            TO=T
            DILR=DILR+KCP*(1+SAMPLE/TIP)*ERP-KCP*ERPO
            ERPO=ERP
            ERMU=MUDES-MU
            SIGH=MU/YG+M
            SIGD=SIGH+KCS*(1+SAMPLE/TIS)*ERMU-KCS*ERMUO
            ERMUO=ERMU
            DMINS=SIGH*X/(SFMAX-SDES)
            IF (DILR.GE.DMAX) DILR=DMAX
            IF (DMINS.LE.DMIN) DMINS=DMIN
            IF (DILR.LE.DMINS) DILR=DMINS
            SF=SIGD*X/DILR+SDES
            IF (SF .LE. SFMIN) SF=SFMIN
            IF (SF .GE. SFMAX) SF=SFMAX
            QC=QP+KCZ*ERZ
            ZF=ZDES+QC*B*X/DILR
            IF (ZF.GE.ZFMAX) ZF=ZFMAX
            IF (ZF.LE.ZFMIN) ZF=ZFMIN
        M1..CONTINUE
          END $"OF PROCEDURAL"
        END $"OF DISCRETE"

      TERMT (T .GE. TEND)
    END $ "OF DYNAMIC"

    TERMINAL
    END $ "OF TERMINAL"

END $ "OF PROGRAM"
```

APPENDIX C.5
ACSL: Structured Model, Open Loop

```
PROGRAM PROJ

  INITIAL
    CONSTANT TEND=150
    CONSTANT MUMAX=0.123
    CONSTANT KT=0.006
    CONSTANT KD=0.0008
    CONSTANT KS=1.0
    CONSTANT KPP=0.012
    CONSTANT KH=0.003
    CONSTANT KPEN=1.3
    CONSTANT MUDES=0.015
    CONSTANT YP=0.100
    CONSTANT YG=0.400
    CONSTANT M=0.026
    CONSTANT XO=4.000
    CONSTANT A00=2.000
    CONSTANT A10=2.00
    CONSTANT A20=0.0
    CONSTANT PAAF=2.00
    CONSTANT SF=50.0
    CONSTANT PAAO=0.0
    CONSTANT PPO=0.0
    CONSTANT PO=0.0
    CONSTANT PHO=0.0
    CONSTANT DILO=0.008
    CONSTANT SO=0.000
      ALGORITHM IALG=2
      VARIABLE T=0.0
  END $ "OF INITIAL"

  DYNAMIC
    CINTERVAL CI=5.00
      DERIVATIVE PROJ
        X=A0+A1+A2
        DILR=DILO*EXP(MUDES*T)
        MU=MUMAX*S/(KS+S)
        DA0DT=MU*A0-KT*A0
        A0=INTEG(DA0DT,A00)
        DA1DT=KT*A0-KD*A1
        A1=INTEG(DA1DT,A10)
        DA2DT=KD*A1
        A2=INTEG(DA2DT,A20)
        DPPDT=KPP*A1-KPEN*PAA*PP
        PP=INTEG(DPPDT,PP0)
        DPDT=KPEN*PAA*PP-KH*P-DILR*P
        P=INTEG(DPDT,PO)
```

```
        DPHDT=KH*P-PH*DILR
        PH=INTEG(DPHDT,PHO)
        DPAADT=DILR*(PAAF-PAA)-KPEN*PP*PAA
        PAA=INTEG(DPAADT,PAAO)
        DSDT=DILR*(SF-S)-(MU*A0/YG+M*(A0+A1)+A1*KPP/YP)
        S=INTEG(DSDT,SO)
        PROCEDURAL (PAA,S,PP,PH,P=PAA,S,PP,PH,P)
          IF (PAA .LE. 0.0) PAA=0.0
          IF (S .LE. 0.0) S=0.0
          IF (PP .LE. 0.0) PP=0.0
          IF (PH .LE. 0.0) PH=0.0
          IF (P .LE. 0.0) P=0.0
        END $ "OF PROCEDURAL"
      END $ "OF DERIVATIVE"

    TERMT (T .GE. TEND)
  END $ "OF DYNAMIC"

  TERMINAL
  END $ "OF TERMINAL"

END $ "OF PROGRAM"
```

APPENDIX C.6
ACSL: Structured Model, Closed Loop

```
PROGRAM BIOEQ

  INITIAL
    CONSTANT V=14.0
    CONSTANT MUMAX=0.123
    CONSTANT KS=1.0
    CONSTANT YG=0.500
    CONSTANT YP=1.20
    CONSTANT M=0.026
    CONSTANT B=0.30
    CONSTANT QPMAX=0.0017
    CONSTANT A=0.0145
    CONSTANT KT=0.006, KD=0.0008, KPP=0.012, KH=0.003
    CONSTANT KPEN=1.3, YPA=0.1, YGA=0.4
    CONSTANT XO=4.0, A00=4.0, A10=0.0, A20=0.0
    CONSTANT SA=200.0, SB=000.0, SC=0.0
    CONSTANT ZA=2.0, ZB=5.0, ZC=0.0
    CONSTANT SFMAX=200.0, SFMIN=0.0
    CONSTANT ZFMAX=5.0, ZFMIN=0.0
    CONSTANT MUDES=0.015
    CONSTANT XO=4.0000
    CONSTANT XMO=4.0
    CONSTANT PO=0.0, PPO=0.0
    CONSTANT ZO=0.0
    CONSTANT DMAX=0.02143, DMIN=0.0080000
    CONSTANT KCS=13.0, KCP=-0.44, KCZ=0.019, TIP=2.80, TIS=8.0
    CONSTANT PDES=4.0, ZDES=0.4, SDES=0.139
    CONSTANT LAM=0.75
    CONSTANT SO=0.0
    CO=ALOG(XO)
    TO=0.0
    P11=1.0
    P22=1.0
    P12=0.0
    ERPO=PDES-PO
    ERZO=ZDES-ZO
    MUO=MUDES
    ERMUO=MUDES-MUO
    DILR=DMIN
    DILC=DMIN
    DILO=DMIN
    SIGC=MUDES/YG+M
    SIGD=SIGC
    CTT=0.0
    AGEO=8.1301
    CONSTANT TEND=150.0
    ALGORITHM IALG=2
```

```
    VARIABLE T=0.0
END $ "OF INITIAL"

DYNAMIC
  CINTERVAL CI=0.20
    DERIVATIVE MODEL
      PROCEDURAL(F,FA,FB,FC=SA,SB,SC,ZA,ZB,ZC,SF,ZF)
        F=V*1000*DILC/60
        FA=SF*F/SA
        FB=(ZF*F-ZA*FA)/ZB
        FC=F-FA-FB
      END $ "OF PROCEDURAL"
      PROCEDURAL(SIGC=T,CTT,SIGD)
        IF(T.GE.CTT) SIGC=SIGD
      END $ "OF PROCEDURAL"
      MU=MUMAX*S/(KS+S)
      X=A0+A1+A2
      DA0DT=MU*A0-KT*A0
      A0=INTEG(DA0DT,A00)
      DA1DT=KT*A0-KD*A1
      A1=INTEG(DA1DT,A10)
      DA2DT=KD*A1
      A2=INTEG(DA2DT,A20)
      XM=XM0*EXP(MUDES*(T-CTT))
      PROCEDURAL(DILC,SF=DMAX,SIGC,XM,DILO,SDES,SFMAX,SFMIN,T,CTT)
        IF(T.GE.CTT) DILC=DILR
        IF(T.GE.CTT) DILO=DILR
        SF=SIGC*XM/DILC+SDES
        IF (SF .GE. SFMAX) SF=SFMAX
        IF (SF .EQ. SFMAX) GOTO M3
        IF (SF .LE. SFMIN) SF=SFMIN
        GOTO M4
      M3..DILC=DILO*EXP(MUDES*(T-CTT))
        IF (DILC .GE. DMAX) DILC=DMAX
      M4..CONTINUE
      END $ "OF PROCEDURAL"
      SIGA=MU/YG*A0+M*(A0+A1)+KPP/YPA*A1
      DSDT=-SIGA+(SF-S)*DILC
      S=INTEG(DSDT,S0)
      DPPDT=KPP*A1-KPEN*PP*Z
      PP=INTEG(DPPDT,PP0)
      DPDT=KPEN*PP*Z-KH*P-DILC*P
      P=INTEG(DPDT,P0)
      DZDT=-KPEN*Z*PP+DILC*(ZF-Z)
      Z=INTEG(DZDT,Z0)
      DAGEDT=1.0-AGE*MU
      AGE=INTEG(DAGEDT,AGE0)
      QP=QPMAX*A*AGE*EXP(1.-A*AGE)
      Y=ALOG(X)
      PROCEDURAL(S,Z=S,Z)
        IF (S .LE. 0.0) S=0.0
```

```
        IF (Z .LE. 0.0) Z=0.0
      END $"OF PROCEDURAL"
      IAE=INTEG(ABS(ERMU),ERMUO)
    END $ "OF DERIVATIVE"

    DISCRETE CONTR
      PROCEDURAL
        INTERVAL SAMPLE=4.0
        IF(T .LT. 4.0) GO TO M1
        CTT=T+1.5
        XM=OU(10.0,X,0.3000)
        XMO=XM
        YM=ALOG(XM)
        ERP=PDES-P
        ERZ=ZDES-Z
        YB=MUO*T+CO
        D=LAM+P11*TO**2+2*P12*TO+P22
        K1=(P11*TO+P12)/D
        K2=(P12*TO+P22)/D
        MU=MUO+K1*(YM-YB)
        C=CO+K2*(YM-YB)
        P11=1/LAM*(P11-K1*K1*D)
        P12=1/LAM*(P12-K1*K2*D)
        P22=1/LAM*(P22-K2*K2*D)
        CO=C
        MUO=MU
        TO=T
        DILR=DILR+KCP*(1+SAMPLE/TIP)*ERP-KCP*ERPO
        ERPO=ERP
        ERMU=MUDES-MU
        SIGH=MU/YG+M
        SIGD=SIGH+KCS*(1+SAMPLE/TIS)*ERMU-KCS*ERMUO
        ERMUO=ERMU
        DMINS=SIGH*X/(SFMAX-SDES)
        IF (DILR.GE.DMAX) DILR=DMAX
        IF (DMINS.LE.DMIN) DMINS=DMIN
        IF (DILR.LE.DMINS) DILR=DMINS
        SF=SIGD*X/DILR+SDES
        IF (SF .LE. SFMIN) SF=SFMIN
        IF (SF .GE. SFMAX) SF=SFMAX
        QC=QP+KCZ*ERZ
        ZF=ZDES+QC*B*X/DILR
        IF (ZF.GE.ZFMAX) ZF=ZFMAX
        IF (ZF.LE.ZFMIN) ZF=ZFMIN
    M1..CONTINUE
      END $"OF PROCEDURAL"
    END $"OF DISCRETE"
  TERMT (T .GE. TEND)
END $ "OF DYNAMIC"
TERMINAL
END $ "OF TERMINAL"
END $ "OF PROGRAM"
```