THE UNIVERSITY OF CALGARY

MODELLING AND COMPUTER CONTROL OF CONTINUOUS IMMOBILIZED CELL PENICILLIN FERMENTATIONS

by

THEODOROS LINARDOS

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN ENGINEERING

DEPARTMENT OF CHEMICAL AND PETROLEUM ENGINEERING

CALGARY, ALBERTA, CANADA

April 1987

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,

MODELLING AND COMPUTER CONTROL OF CONTINUOUS IMMOBILIZED CELL PENICILLIN FERMENTATIONS

submitted by Theodoros I. Linardos in partial fulfillment of the requirements for the degree of Master of Science in Engineering

NKalogerakis

Dr. N.E. Kalogerakis, Supervisor/Committe Chairman
Department of Chemical and Petroleum Engineering

Franco Berruti

Dr. F. Berruti
Department of Chemical and Petroleum Engineering

Leo Behie

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

T.A. Thorpe

Dr. T.A. Thorpe,
Department of Biology

April 28, 1987

date

ABSTRACT

Immobilized Cell Technology provides a viable alternative to fed-batch free cell penicillin fermentations since confinement of cells inside a porous matrix alleviates the problems associated with chemostats, e.g. high broth viscosity, poor oxygen mass transfer and washout danger if the growth rate for any reason becomes smaller than the dilution rate being used.

In the present work a simple mathematical model of continuous immobilized cell fermentations was developed and it was shown that a Quasi-Steady-State (QSS) can be achieved, whereby biomass grows exponentially at constant growth rate while the penicillin, precursor and glucose concentrations in the fermentor remain constant.

A computer control algorithm was designed, based on the developed model, that can be easily implemented and can ensure fast transition to and regulation of the system at optimally selected QSS operating conditions throughout the run. In the controller design an Extended Kalman Filter was used for the estimation of the non-directly measurable variables (biomass concentration and average cell age), based on measurements of penicillin and precursor concentrations in the broth and, the CO_2 concentration in the exit gas.

Experimental results showed overall good agreement with the model predictions and confirmed the existence of a QSS for growth and glucose concentration. The value of the maintenance energy was found to be as low as 0.008 h^{-1} and the value of the Monod saturation constant approximately 0.3 g/L.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Nicolas Kalogerakis for many stimulating and helpful discussions during the course of this project. His enthusiasm for the project was a constant source of encouragement to me.

I would also like to acknowledge the many helpful suggestions by Dr. Leo Behie regarding the serious reactor design problems encountered throughout the experimental work.

I wish to thank Dr. Maurice Gaucher from the Division of Biochemistry for providing the laboratory facilities and analytical equipment as well as valuable insight on penicillin fermentations.

Special thanks go to Dr. John W. Wong and Mr. Yinliang Chen for their invaluable assistance in carrying out the experiments. Also I would like to thank Mr. Phil Bakes for experimental assistance during the early part of this research.

Financial assistance for this project was provided by NSERC and Chembiomed Ltd. for which I am very grateful.

TO MY PARENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$\underline{A}$	process matrix in linearized model
$\underline{B}$	control matrix in linearized model
$\underline{C}$	measurement matrix in linearized model
c	CO_2 concentration in exit gas, g/L
D	dilution rate, h^{-1}
$\underline{f}$	vector of state equations
F	total flow rate into or out of the reactor, L/h
$\underline{h}$	vector of measuring equations
$\underline{K}$	Kalman filter gain matrix
K_{cs}	feedback gain in growth controller
K_{cz}	feedback gain in PAA controller
k_s	Monod equation saturation constant, g/L
k_z	parameter in Equation (11), g/L
k_4, k_5, k_6	parameters in Equation (15)
$\underline{L}_o$	observability matrix
m	maintenance energy , h^{-1}
$\dot{N}_c$	CO_2 flux into gas phase, $g/L\text{-}h$
$\underline{P}$	matrix of estimate covariance
PAA	phenylacetic acid
p	penicillin concentration, g/L
q_p	specific penicillin production rate, $g/(gbiomass-h)$
$\underline{Q}^{-1}$	covariance of measurement errors
QSS	quasi-steady-state

r	volumetric production or consumption rate, g/Lh
$\underline{R}^{-1}$	covariance of process noise
s	glucose concentration, g/L
t	time, h
$\underline{u}$	vector of manipulated variables
$\underline{y}$	vector of measured variables
Y_G	cell growth yield (g–*biomass per* g–*glucose*)
Y_p	penicillin to glucose yield
x	biomass concentration, g/L
$\underline{x}$	vector of state variables
ν	dimensionless quantity in Eq. (16)
V	working volume, L
z	precursor (PAA) concentration, g/L

GREEK SYMBOLS

α	parameter in Equation (11), h^{-1}
β	stoichiometric constant (PAA to penicillin)
$\underline{\eta}$	vector of random measurement errors
$\theta_1, \theta_2, \theta_3$	parameters in Eq. (47)
λ	average cell age, h
μ	specific growth rate, h^{-1}
$\underline{\xi}$	vector of process noise
σ	specific uptake rate of glucose, h^{-1}
$\hat{\sigma}$	estimated σ at QSS ,h^{-1}
τ	time constant, min

SUBSCRIPTS

c	variable refers to CO_2
d	desired value of the variable
f	value of the variable in the feed stream
g	variable refers to the gas phase
k	value of the variable at sampling point k
min	minimum value of the variable
max	maximum value of the variable
o	initial value of the variable
opt	optimum value of the variable
s	variable refers to glucose
x	variable refers to biomass
z	variable refers to PAA

1.0 INTRODUCTION

A 1983 world production exceeding 20,000 metric tonnes at an annual growth rate of 2.5% clearly demonstrates the importance of penicillin among the modern chemotherapeutic agents. The current high levels of penicillin production have been made possible through an extensive research activity and numerous innovations introduced into the penicillin fermentation over the last two decades (Pirt, 1985).

Between the years 1950 and 1980 the value of the specific production rate of penicillin, q_p, increased 10-fold due to genetic improvement of the used strains of *P. chrysogenum*. The output rate however increased 40-fold with the contribution of improvements in the fermentation techniques and a better understanding of the physiology of the producing organism (Pirt, 1985). This indicates that there is a great potential for improvement through optimization of the culture conditions and the physiological environment of the organism.

Operation of a bioreactor in a continuous fashion allows easiest control of environmental and physiological factors (e.g. concentrations of glucose, precursor, NH_3, etc.), while minimizing down-time for cleaning and maintenance. However, serious problems associated with the chemostat mode of operation, have prevented the use of continuous bioreactors for industrial penicillin production. As a result fed-batch is the usual mode of operation of industrial reactors today.

In a chemostat the free cells are the cause of many practical problems like clogging of output lines, filters etc. Furthermore, the free cells are washed out if the dilution rate is higher than the specific growth rate. This is due to the fact that more cells are being removed than being produced. As a result the biomass concentration in the fermentor drops fast and so does the reactor productivity. Immobilization of the cells on a support matrix offers a promising solution to the above problems. The cells are permanently anchored on the beads and they remain in the reactor at all times even when the growth rate is much smaller than the dilution rate.

In a continuous immobilized cell reactor high cell concentrations can be achieved, giving rise to high volumetric productivities. The problem of oxygenation at high cell densities is expected to be relieved by the altered morphology of the immobilized mycelia. Confinement of the cells on the beads produces sphere-like particles that move freely and change favorably the rheological characteristics of the broth. (Gbewonyo and Wang, 1983a; 1983b).

It is a widespread misconception in the bioengineering field, that a continuous culture should operate under steady state conditions. In fact, an immobilized cell reactor cannot reach a true steady state due to the biomass accumulation. However, a Quasi-Steady-State (QSS) can be achieved, whereby the biomass is increasing exponentially at an optimally selected growth rate while the concentrations of substrate, precursor and end product are kept at a constant value.

Proper control of the Quasi-Steady-State requires the use of a computer control algorithm, since a constant dilution rate will not result in a constant growth rate, as would be the case for a chemostat. The lack of computer control algorithms for immobilized cell cultures has been one of the main reasons for reluctance in the industrial implementation of continuous immobilized cell reactors.

In the present work a simple mathematical model for immobilized cell continuous penicillin fermentations has been formulated and it constituted the basis for the design of a computer control scheme that can ensure optimum QSS conditions throughout a production run.

2.0 LITERATURE REVIEW

2.1 Immobilized Cells in Penicillin Fermentation

The filamentous morphology of *P. chrysogenum* cells is associated with non-Newtonian rheological behavior and high viscosity of the fermentation broth. This results in severe oxygen transfer limitations at moderate to high free cell concentrations. For example, Deindorfer and Gaden (1955) have reported an 85% decline in the k_La value (from an initial value of 0.08 *moles/L h atm*) at cell concentrations 'of 13 *g/L*.

In industrial practice oxygen transfer limitations are partially overcome by strong agitation of the broth. This practice requires excessive power input that results in high power and cooling costs, which constitute up to 20% of the overall fermentation cost (Chibata et al., 1983). In addition the cells are damaged by the high shear rates that develop and a high percentage (30 - 40 %) of the glucose fed is used for cell maintenance requirements.

The use of the pellet form of the fungi is an alternative to free cell fermentations. The reduced viscosity of the broth permits use of tower loop reactors with resulting savings up to 90% of the costs associated with the culture aeration. At the same time the volumetric mass transfer coefficient, k_La is 4-fold or more higher than in filamentous cultures. Furthermore, the cell maintenance requirements are drastically reduced and the glucose to product yield improves accordingly (Konig et al., 1982).

Pelletization, however, depends on a number of biological and environmental factors such as the particular strain, inoculum concentration, medium composition, pH and hydrodynamic shear forces (Gbewonyo and Wang, 1983a). Therefore, it is not always possible to control the fermentation conditions in order to obtain reproducible pellet formation.

Immobilization of live cells on spherical beads provides a more convenient means of changing the mould morphology. The bead size can be selected so that mass transfer problems can be alleviated. In addition, the pore size distribution and the total pore volume can be modified so that high biomass concentrations can be supported.

Early experiments by Morikawa et al. (1979) with *P. chrysogenum* cells immobilized in polyacrylamide gel, collagen or calcium alginate beads did not prove very successful. Due to the harsh immobilization techniques used and the toxicity of the materials, the immobilized cell productivity was low. The oxygen consumption for immobilized cells was only 30% of the value for the free cells. This can be explained either by reduced cell viability or mass transfer problems or both. An interesting outcome of this study, however, was that the half life of the immobilized biocatalyst was six times higher of that for free cells.

In 1983 Gbewonyo and Wang immobilized spores of *P. chrysogenum* in small (450 μm average diameter) beads of Celite, an inorganic substance made of diatomatious earth and composed mainly of SiO_2. Subsequently, the spores were transferred into shake flasks to germinate and the produced cells were naturally entrapped in the porous beads. With a 10% concentration of Celite beads a maximum cell concentration of 60 g/L was obtained as compared to 30 g/L in the control (free cell) cultures.

Gbewonyo and Wang (1983b) also used the new immobilization technique for production of penicillin-G in an experimental bubble column. In the confined cell cultures the final cell densities were 24 g/L and the Penicillin titer 5.5 g/L. Only 17 g/L of final biomass and 2 g/L of penicillin were obtained in the free cell cultures. In addition, the k_La for oxygen transfer increased 3-fold and the power requirements were reduced by 50% in the immobilized cell cultures.

Deo and Gaucher (1984) experimented with *P. chrysogenum* cells immobilized in κ-carrageenan, a natural polymer. The half-life of the immobilized cells in a replacement culture was ninefold higher than that exhibited by free cells. A continuous fluidized bed bioreactor was operated with immobilized cells for sixteen days. At the same time the effects on the penicillin productivity of glucose feed concentration and dilution rate were studied.

Jones et al. (1986) compared κ-carrageenan to Celite beads with respect to the penicillin production by immobilized cells. They found that the small diameter (410 μm) of the Celite beads as compared to the κ-carrageenan beads (3 mm), resulted in a five times higher volume productivity with cells immobilized on Celite. The specific productivity for both support matrices was comparable to the values obtained for free cells.

The most recent work was reported by Kim et al. (1986). They used a three phase fluidized bed fermentor in the semicontinuous and repeated fed-batch modes of operation. A significant increase in cell growth and penicillin yields was observed compared to filamentous cultures. Moreover the specific productivities were roughly the same for immobilized or free cell cultures. This result supports the assumption that no nutrient mass transfer limitations exist within the Celite beads.

The above researchers also identified the problem of poor mixing at high cell concentrations due to increased concentration of free cells and the development of fluffy loose bioparticles. They successfully controlled the bioparticle size by phosphate-limiting the culture. Under these conditions, they claimed that penicillin production was maintained at 80% of its maximum for at least one month.

2.2 Computer Control of Penicillin Fermentations

In the fermentation industry the usual variables being controlled are pH, dissolved oxygen (DO) concentration and temperature. Conventional analog PID controllers are adequate for this purpose. Application of more sophisticated computer control schemes has been delayed for a number of reasons.

The most important reason is the marginal productivity gains achieved by proper control, when these are compared to the enormous gains attainable by strain development. Secondly, mathematical modelling the penicillin fermentation has been a difficult task and very few models have been reported which describe adequately the dynamic characteristics of the system and at the same time are simple enough to be used for control purposes. The third reason is the lack of biosensors suitable for measuring intracellular activities.

Nevertheless, in the past decade some very important aspects of the penicillin fermentations have been elucidated, a fact that permitted the development of some successful models. A notable contribution in this direction was the establishment of the Theory of Fed-Batch Culture by S.J.Pirt (1974). According to this theory, a Quasi-Steady-State could be reached where the growth rate is equal to the dilution rate. This provides a means of maintaining a culture at environmental conditions optimally selected, for maximum productivity.

Kalogerakis and Boyle (1981) successfully tested a control system that forced fed-batch baker's yeast fermentations to rapidly reach and maintain any desired QSS conditions. In the meantime, Aiba et al. (1976) had proposed a computer control scheme for the optimization of the overall yield in fed-batch baker's yeast fermentations. They manipulated the feed rate in order to control the respiratory quotient (RQ) within a desired range.

Wang et al. (1977) used the approach of material balancing to monitor growth in a yeast fermentation. The same method was later used by Heijnen et al. (1979) in the modelling of penicillin fermentations.. A few years later Meyer and Beyeler (1984) based on the material balancing method designed a control scheme for continuous baker's yeast fermentations. Callegos and Callegos (1984) presented an estimation and control technique tested on a simulated continuous fermentation for biomass production. The control law was based on knowledge of the biomass and glucose concentration values at discrete sampling points. The authors assumed that the biomass concentration could be measured on-line and they designed an "observer" to estimate the value of the substrate concentration from the biomass measurements.

The complicated nature of penicillin fermentations has resulted in a lag in the development of computer control schemes. Most important contributions in this field were those of Mou and Cooney (1983a; 1983b) and Mou (1983) who applied the material balancing method in order to control the growth in a fed batch penicillin fermentation by manipulating the glucose feeding rate. Recently, Montague et al. (1986) and Frueh et al. (1986), have applied Kalman Filtering techniques for monitoring and controlling fed-batch penicillin fermentations.

All the proposed control schemes for penicillin fermentations are of the "single input-single output" (SISO) type. That means only one process variable is being controlled, by manipulation of one of the input variables (e.g. growth rate is controlled by manipulation of the glucose feeding rate). It is clear that optimization of a continuous penicillin fermentation cannot be achieved by controlling the growth rate alone. The penicillin concentration should be also controlled at a constant high value, so that downstream processing costs are minimized. Furthermore, the precursor concentration should be kept within certain levels to avoid toxicity but ensure availability as well. It has also been argued that there are some optimum values for the $NH^{+}{}_{4}$ ion concentration as well as for several other ions (Pirt, 1985).

One step in the direction of multiple control in fermentations is the work presented here. A simple non-interactive control scheme is described for simultaneous control of growth rate, penicillin concentration and precursor concentration. Design of the controller is based on a simple mathematical model describing the phenomena of growth and penicillin production.

3.0 MODEL DEVELOPMENT

The complicated nature of a fermentation process makes rigorous modelling a very difficult task. A detailed mechanistic model would require a large number of equations and variables to describe the dynamic behavior of the system. Most importantly a tremendous experimental effort would be required for the determination of all the kinetic parameters present in the model.

However, modelling for control purposes requires only description of the main dynamic characteristics of the process. The equations involved should be simple in structure so that can be easily used for the design of a control scheme. Moreover, macroscopic variables should be used that are well defined and relatively easy to measure.

The development of a control-oriented model for the penicillin fermentation will be presented in this chapter. The time evolution of the important phenomena of growth, product formation and uptake of limiting substrate and precursor will be modelled through the use of simple differential equations describing appropriate mass balances.

3.1 Selection of State Variables

The status of the system at any point in time can be described by a set of variables, called "state variables". In a fermentation model these variables are selected so that they adequately represent the main biological phenomena the designer wishes to include in the process model.

The two important phenomena included in the present model are cell growth and penicillin formation and they are represented by the biomass and penicillin concentrations respectively. Glucose concentration is included in order to account for the effect of the limiting substrate on the growth rate. The effect of precursor availability on penicillin formation is modelled through the phenylacetic acid (PAA) concentration. Finally the average age of the cell population is introduced as the fifth variable in order to account for the effects of the cell physiology on product formation.

Knowledge of the system status requires knowledge of the values of state variables at any point in time. However, from the above set of state variables only the concentrations can be measured directly. Furthermore, only concentrations of penicillin and PAA can be measured on-line relatively easily and reliably. It is clear that if we want to design a computer control algorithm based on this model we have to estimate biomass, glucose and average cell age by using suitable mathematical techniques and make use of other auxiliary measurements.

Estimation of the unknown variables is only possible, if the measured or "output" variables are dependent on - therefore contain information about - the estimated ones. In the present model penicillin production is related to the average cell age and the biomass concentration. No direct information is available about the glucose concentration. The PAA equation does not add any new information since PAA consumption is merely proportional to the penicillin production.

The problem identified above can be solved by introducing an additional state variable that will contain information about the glucose concentration. Such a variable is the carbon dioxide concentration in the exit gas. This variable also meets the additional requirement of easy and reliable on-line measurement (by using a infrared analyzer or a mass spectrometer, if available). The dependence of CO_2 concentration on the glucose concentration will become clear in the following paragraphs, where the state equations are presented.

3.2 Governing State Equations

In deriving the governing differential equations the following assumptions have been made

3.2.1 Assumptions in the Proposed Model

(1) The working volume remains constant at all times. This is a good assumption provided the inlet and outlet flowrates are closely monitored.

(2) The reactor is considered to be perfectly mixed. Therefore, no spatial variations exist on a macroscopic level.

(3) Reaction rates are considered slow compared to the nutrient mass transfer rates and hence, the reactor can be considered to be under kinetic control (Kalogerakis et al., 1986b).

(4) The cells are immobilized on Celite beads and the number of free cells in the reactor is negligible.

(5) Only glucose is the limiting substrate (carbon source). Media formulation can ensure that this assumption remains always valid.

(6) No oxygen limitation appears as long as the dissolved oxygen is kept above a critical value (approximately 25% of the saturation concentration). Oxygen becomes limiting only at high cell densities.

(7) The overhead volume of the reactor is small compared to the gas flow rate. Therefore, the composition of the gas exiting the reactor vessel can be assumed to be the same as of the gas coming out of the liquid phase in the fermentor.

(8) The amounts of N_2, O_2 and CO_2 that dissolve into the liquid phase are negligible, compared to the high air flow rate used for fluidization. In addition, the volume of O_2 consumed is small and approximately equal to the volume of produced CO_2. Therefore, there is no significant change in the molar flow in and out of the reactor.

3.2.2 Unsteady State Mass balances

The equations describing the changes in the state variables during a fermentation evolve directly from corresponding unsteady-state partial mass balances. In the following section each component is treated separately.

(a) Biomass

The mass balance for the biomass present in the liquid phase is

$$V \frac{dx}{dt} = r_x V \tag{1}$$

or

$$\frac{dx}{dt} = r_x \tag{1.a}$$

where x is the biomass concentration and V the working volume. There is no inlet term in the right hand side of the equation because the feed is sterile. The outlet term is also zero according to assumption #4.

The rate of biomass production per unit of working volume is related to the biomass concentration in the fermentor as follows

$$r_x = \mu x \tag{2}$$

Therefore, the final form of the biomass equation is

$$\frac{dx}{dt} = \mu x \tag{3}$$

The specific growth rate is predominantly a function of s and here it is assumed that it follows the usual Monod kinetics, namely

$$\mu = \mu_{\max} \frac{s}{(k_s+s)} \tag{4}$$

where $\mu_{\max}$ is the maximum growth rate, k_s the saturation constant and

s the concentration of the limiting substrate (glucose).

(b) Glucose

The mass balance for glucose is given by

$$\frac{ds}{dt} = -r_s + D(s_f - s) \qquad (5)$$

where s is the glucose concentration in the reactor, r_s is the rate of glucose consumption per unit of working volume and s_f is the glucose concentration in the feed. The variable D is the dilution rate defined as

$$D = \frac{F}{V} \qquad (6)$$

where F is the volumetric feed flowrate. The rate of glucose consumption r_s is related to the biomass concentration through an equation of the form

$$r_s = \sigma x \qquad (7)$$

where σ is the specific uptake rate of glucose, usually expressed in *g-glucose* consumed per *g-biomass* produced. The specific uptake rate is related to growth and penicillin production through the equation

$$\sigma = \frac{\mu}{Y_G} + m + \frac{q_p}{Y_p} \qquad (8)$$

where Y_G is the cell growth yield, m is the maintenance energy, q_p is the penicillin production rate and Y_p is the penicillin from substrate yield (Pirt, 1974).

(c) Penicillin

The production of penicillin has been found to be proportional to the biomass concentration. The proportionality constant, q_p, is the specific penicillin production rate and is usually expressed in *g–penicillin* produced per *g–biomass* per hour. In terms of q_p the penicillin mass balance is

$$\frac{dp}{dt} = q_p\, x - D\, p \tag{9}$$

If the specific production rate was constant, then calculation of product formation would be straightforward. However, q_p varies widely during a single fermentation. It is usually smaller in the beginning and at the end of a batch production run. The changing ability of cells to produce penicillin has been modelled in this work by using the average age of the cells, λ.

The introduction of the average cell age as a new state variable requires an extra differential equation to define it. The equation which relates λ to the specific growth rate is (Fishman and Biryukov, 1974; Holmberg and Randa, 1982).

$$\frac{d\lambda}{dt} = 1 - \lambda\mu \tag{10}$$

The relation between λ and q_p is given by the equation

$$q_p = q_p^{\max}\, \alpha\, \lambda\, exp(1 - \alpha\lambda)\, \frac{z}{(k_z + z)} \tag{11}$$

Parameter α is set equal to $1/\lambda_{opt}$, where λ_{opt} is the average cell age when q_p takes its maximum value. The term $\frac{z}{(k_z + z)}$ accounts for limitation in penicillin production by reduced precursor availability. The parameter k_z takes a very small value (0.001 g/L). If the concentration $z >> k_z$ then Eq. (11) reduces to

$$q_p = q_p^{\max} \alpha \lambda \, exp(1 - \alpha\lambda) \tag{12}$$

(d) Phenylacetic acid (PAA)

The precursor equation is similar to the one for penicillin. The only difference is the inclusion of the stoichiometric constant β, which relates penicillin produced to the amount of precursor consumed. Therefore, the mass balance yields

$$\frac{dz}{dt} = -\beta q_p x + D(z_f - z) \tag{13}$$

where z is the PAA concentration in the reactor and z_f is the PAA concentration in the feed stream.

(e) Carbon dioxide in gas phase

The mass balance for CO_2 in the gas phase is

$$V_g \frac{dc}{dt} = \dot{N}_c + F_g(c_f - c) \tag{14}$$

where V_g is the gas phase volume, $\dot{N}_c$ is the CO_2 flux into the gas phase, F_g is the volumetric flow of gas into or out of the reactor and c_f is the CO_2 concentration in the gas entering the reactor. Volumes and volumetric flows of gasses are taken at $25^o\,C$ and at the reaction pressure (approximately 0.25 *baru*).

According to assumption #8 the amount of CO_2 entering the gas phase is taken equal to the amount produced, namely

$$\dot{N}_c = r_c V \tag{15}$$

where r_c is the volumetric production rate of CO_2 in the broth. The production rate of CO_2 is a function of specific growth rate, specific production rate of penicillin and the biomass level in the reactor(Calam and Ismail, 1981). The equation describing this relationship is

$$r_c = \left(\frac{\mu}{k_4} + k_5 + k_6 q_p \right) x \tag{16}$$

where k_4, k_5 and k_6 are experimentally estimated parameters. The dependence of c on the glucose concentration becomes clear if we recall that μ is directly related to s through the growth kinetics (Monod equation).

If we define $v = \frac{V}{V_g}$ and $D_g = \frac{F_g}{V_g}$, Eq.(14) becomes

$$\frac{dc}{dt} = v\, r_c + D_g (c_f - c) \tag{17}$$

The dynamics of the CO_2 concentration with respect to changes in the air flow rate or input CO_2 concentration are very fast. This can be easily seen when the above equation is rearranged to

$$\frac{dc}{dt} + D_g\, c = v r_c + D_g c_f \tag{17.a}$$

Assuming an air flow rate of 15 L/min and a gas volume of 5 L the value of the constant D_g is 180 h^{-1}. The time constant for the response of c is given by

$$\tau_c = \frac{1}{D_g} \tag{18}$$

and its value would be approximately 0.3 *min*. For a slow process like penicillin fermentation we can assume that the CO_2 concentration in the exit gas is practically always at steady state. Therefore, we can simply use the algebraic equation that results from Eq. (17), by setting $\frac{dc}{dt}$ equal to zero instead of the differential equation, namely

$$c = c_f + \frac{V}{F_g} r_c \tag{19}$$

It should be noted that the QSS value of c is not a constant, but rather it increases exponentially at the same rate as the biomass in the fermentor.

In conclusion, the set of five differential equations which describe the dynamic characteristics of the system (state equations) are

$$\frac{dx}{dt} = \mu x \tag{20}$$

$$\frac{ds}{dt} = -\sigma x + D(s_f - s) \tag{21}$$

$$\frac{dp}{dt} = q_p x - Dp \tag{22}$$

$$\frac{d\lambda}{dt} = 1 - \lambda\mu \tag{23}$$

$$\frac{dz}{dt} = -\beta q_p x + D(z_f - z) \tag{24}$$

$$\frac{dc}{dt} = D_g (c_f - c) + v(\frac{\mu}{k_4} + k_5 + q_p k_6)x \tag{25}$$

If the values of the state variables are known at any point in time, then we can fully describe the status of the system.

It is noted, however, that the experimentally measured variables, which constitute the so called "output vector", are only the penicillin and the PAA concentrations (p and z) and the CO_2 concentration in the effluent gas (c). As it will be shown later, from these three measurements one can obtain a good estimate of all state variables through the use of an Extended Kalman Filter.

3.3 Dynamic Behavior of the Model

3.3.1 Degrees of freedom

By examining the model equations it can be seen that some variables, e.g s_f, D, z_f etc., have to be specified, in order to define the process completely. These variables are the degrees of freedom of the system and usually they can be set by the experimenter. When these variables are used to control the process they are called "manipulated" variables.

The number of degrees of freedom is given by the equation

$$[\textit{degrees of freedom}] = [\textit{number of independent variables}] - [\textit{number of independent equations}] \qquad (26)$$

In the proposed model there are six state equations and ten independent variables (x, s, D, s_f, p, λ, z, z_f, D_g and c). That means that up to four variables must be externally specified. Control of the process can be facilitated by manipulating all or a subset of these variables. Potential manipulated variables are the feed concentrations of glucose and PAA (i.e. s_f and z_f), the dilution rate, D, and the dilution rate for the gas phase, D_g. Usually the value of D_g is dictated by the requirement for acceptable levels of dissolved oxygen, therefore, D_g will not be treated as a manipulated variable from now on.

3.3.2 The Quasi-Steady-State (QSS)

A steady state is usually defined by constant values of all the state variables with respect to time. In our process a typical steady state cannot be achieved. Biomass concentration cannot be constant because the cells are always immobilized on the beads (assumption #4) and hence, they accumulate in the reactor. However, the growth rate can be maintained constant, causing an exponential rise in x. The rest of the state variables can assume constant values. These conditions define the Quasi-Steady-State (QSS).

Growing the microorganisms under QSS conditions is of particular importance, since it implies that the fermentation is carried out under constant environmental conditions. These conditions can be set by the experimenter and in principle should be chosen in some optimal fashion. Proper manipulation of the input variables D, s_f, and z_f provides the means to reach in a reasonable amount of time and maintain a desired QSS.

The values of the manipulated variables at a certain QSS are determined by the solution of the corresponding QSS equations. These equations result from the system equations after setting the desired, constant, specific growth rate in the biomass equation and the time differentials equal to zero in the equations of glucose, penicillin, PAA, average cell age and CO_2, i.e.,

$$x(t) = x_o \exp(\mu t) \; ; \; x_o = x(0) \tag{27}$$

$$D(s_f - s) = \sigma \, x(t) \tag{28}$$

$$D(z_f - z) = \beta q_p x(t) \tag{29}$$

$$\lambda = \frac{1}{\mu} \tag{30}$$

$$Dp = q_p x(t) \tag{31}$$

$$c = c_f + \frac{V}{F_g}(\frac{\mu}{k_4} + k_5 + q_p k_6) x(t) \tag{32}$$

The above equations are the necessary conditions at a certain QSS. Furthermore, it can be proven that the same conditions are sufficient to drive the system to the desired steady state. Starting from an initial biomass concentration of x_o, the profiles of the manipulated variables can be obtained by solving Eq. (28), (29) and (31) for the respective variables

$$D = \frac{q_p}{p} x_0 \exp(\mu t) \tag{33}$$

$$s_f = s + \sigma \frac{p}{q_p} \tag{34}$$

$$z_f = z + \beta p \tag{35}$$

It can be observed that feed concentrations take a constant value dependent only on the desired values of state variables at QSS. The dilution rate, however, is not constant, but rather it follows the biomass in its exponential rise.

3.3.3 Open Loop Simulation

A simulation of the modelled process can provide valuable information and aid us in designing a suitable control system. In an "open loop" run the externally specified variables are given their steady state values. The dynamic behavior of the system, from any initial conditions to the QSS corresponding to the prespecified external variables, is then simulated by integration of the state equations. It was assumed that the same parameter values as for a free cell system in a stirred tank apply, since, to the author's knowledge, there were no such data available for the fluidized bed immobilized cell penicillin fermentations. The values of the kinetic parameters used in the following simulations runs are given in Table 1.

Such a simulation run has been performed starting from an initial biomass concentration of 3 g/L. The external variables are selected such that the penicillin and PAA concentrations are 3.0 and 0.5 g/L respectively at the desired QSS. The steady state growth rate has been chosen to be 0.02 h^{-1}. The time profiles of μ, p and z for this run are shown in Fig. 1.

In this figure the transient response of the growth rate appears to be very fast. Within approximately 10 hours it has reached its desired value. In contrast to the growth rate, the penicillin transient is very slow and 150 hours do not seem enough for p to reach its steady state value. The PAA response is quite slow too and in addition it slightly overshoots the steady state value. Overshoot is

TABLE 1: Nominal Parameter Values used in the Simulations

Parameter	*Value*	
μ_{max}	0.123	h^{-1}
k_s	1.0	g/L
Y_G	0.026	g–*biomass*/g–*glucose*
m	0.026	h^{-1}
Y_P	1.2	g–*penicillin*/g–*glucose*
q_P^{max}	0.008	h^{-1}
β	0.407	

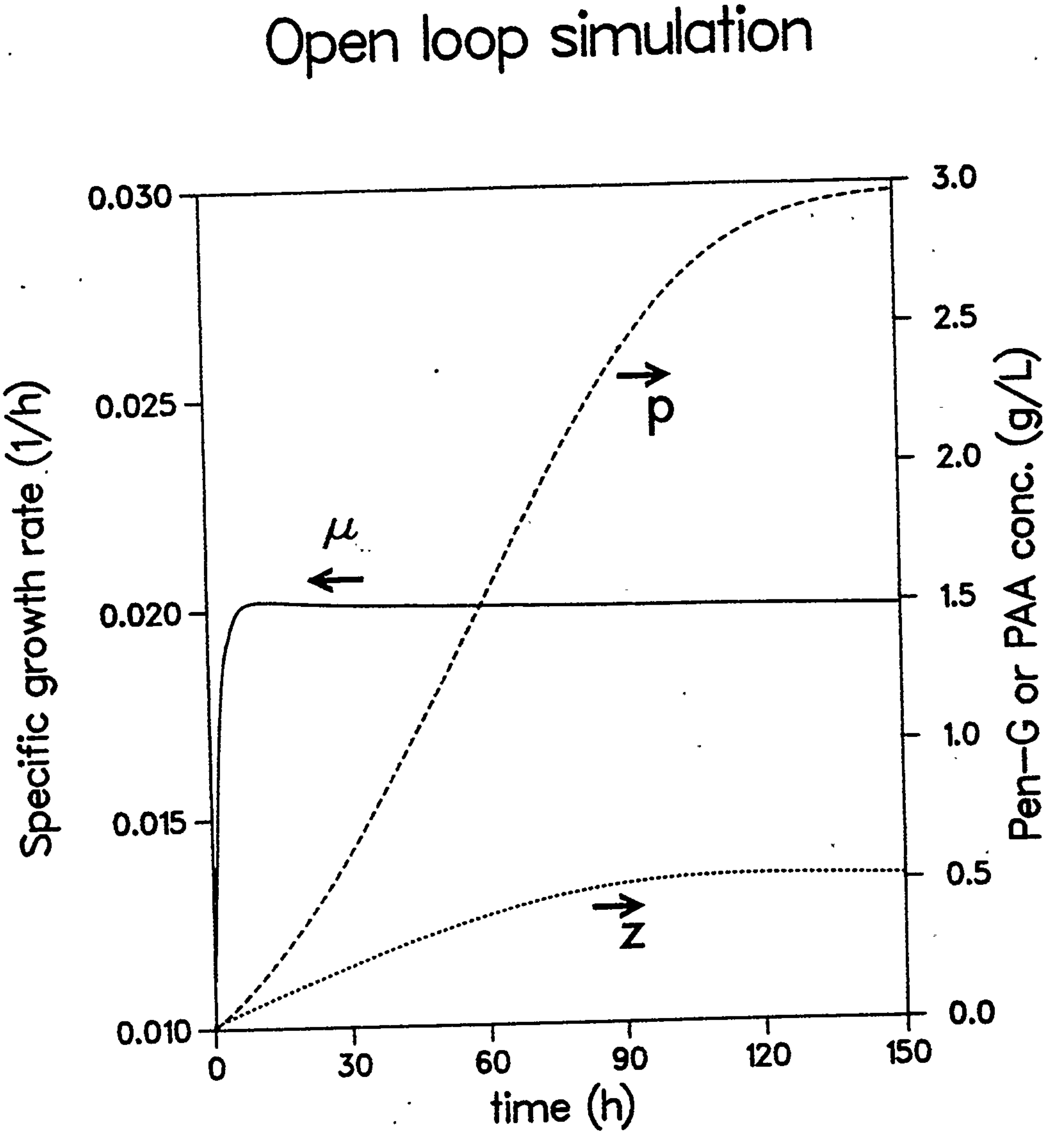

FIGURE 1 Simulated response of μ, *p*, *z* to inputs driving the system to a QSS.

a potential problem, if excessive, because of the toxicity of the precursor at higher concentrations.

So far useful information about the inherent dynamics of the process has been obtained. Slow penicillin response and potentially harmful precursor overshoots are the major drawbacks. These are the first two problems to be addressed by the control system to be designed.

From a control point of view another problem that might be inherent in the process is strong coupling between the responses. This is an undesirable characteristic since it does not allow independent control of the state variables. A second simulation run has been performed where we consider small perturbations around a QSS, to help us investigate such a possibility.

In the second run the system is initially (time = 0) at a QSS with a biomass concentration of 10 g/L, growth rate of 0.02 h^{-1} and penicillin and PAA concentrations of 2.0 and 0.5 g/L, respectively. At time t = 10 h, the manipulated variables are changed so that the system will move to a new QSS with a growth rate of 0.01 h^{-1}.

In Fig. 2, growth rate, penicillin and PAA transients have been plotted. It can be seen again that the response of the growth rate is fast. Furthermore, the growth rate transient, although very sharp, has not produced significant penicillin or PAA transients.

It is interesting to see the response of the system when the manipulated variables are changed so that the system will reach a new

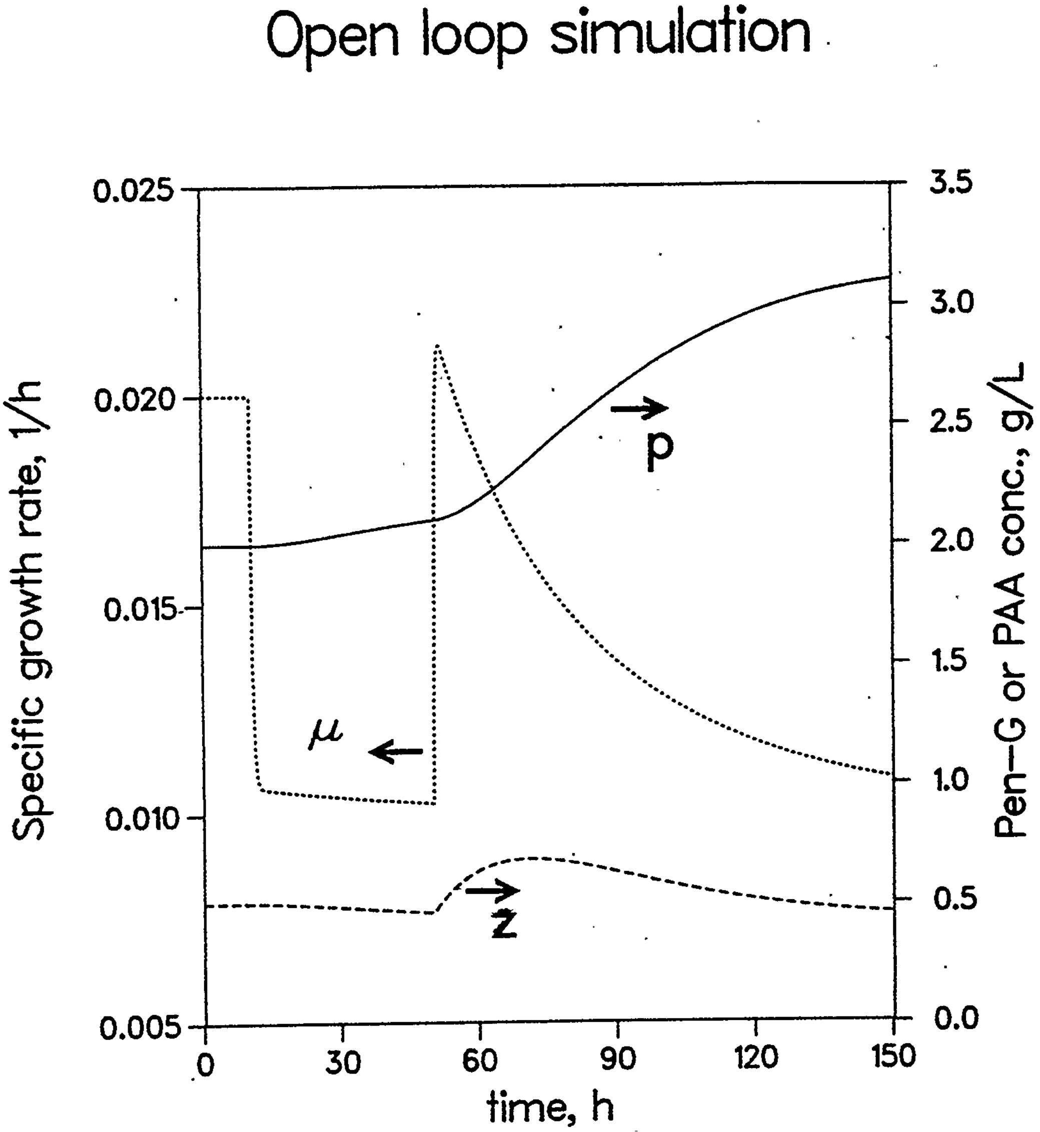

FIGURE 2 Simulated response of μ, *p* and *z* to input changes that drive the system to a new QSS.

steady state with different penicillin concentration. At time 50 h, a new QSS is selected. Here, the steady state penicillin concentration has only been changed from 2.0 to 3.0 g/L. It is seen, however, that a sharp growth rate transient is induced by the penicillin transient. The growth rate of the system has been seriously disturbed and requires a long time until it settles to its steady state value. Besides the growth rate, PAA steady state has also been disturbed and an overshoot can be observed.

The "open loop" dynamic behavior of the process has led to the following conclusions about the behavior of the process

(a) Growth rate dynamics are fast compared to the penicillin dynamics.

(b) Penicillin dynamics are very slow and speeding them up should be a primary objective of the control system design.

(c) There is practically no effect of growth rate transients on penicillin concentration. However, penicillin transients induce strong growth rate disturbances (strong interaction). Proper controller design should eliminate this problem (decoupling).

(d) Potentially dangerous overshoots of PAA can develop without proper control of the PAA concentration in the feed stream.

4.0 CONTROL SYSTEM

4.1 Control objectives

In a production run of continuous penicillin fermentation, penicillin is collected from the effluent stream at all times. Hence, it is important to increase the penicillin concentration to a preset high value and maintain that level throughout the run. It is also desirable to keep the growth rate constant at the value that ensures maximum penicillin productivity. Finally the level of the precursor must be high enough to ensure availability but also low enough to prevent toxicity effects on the cells.

The above requirements, together with the information retrieved from the open loop simulation, suggest the following objectives for the control system

(I) It must ensure fast penicillin transients without any deterioration to the already fast response of the growth rate.

(II) The control system should minimize the interaction among the transients of the controlled variables.

These two objectives can be satisfied by proper design of the controller. The design includes two main steps: (a) the selection of manipulated variables and (b) the development of the control algorithm.

4.2 Controller Design

4.2.1 Selection of Manipulated Variables

The interactions between controlled variables can often be significantly reduced by the proper choice of the manipulated variables. Therefore, an important step in the designing procedure is to determine the pairs of manipulated and controlled variables so that the interaction between control loops is minimum.

By examining Eq. (22) it can be seen that only D can affect the penicillin concentration directly. Therefore, the dilution rate is used to control the penicillin concentration. Eq. (21) suggests that both D and s_f could be used to control the glucose concentration (growth rate). Since D has already been used, s_f will be selected. The same reasoning applies for Eq. (24), so z_f is chosen to control the PAA concentration.

The interaction between penicillin and growth rate (or substrate concentration) transients can be overcome because the product $D(s_f - s)$ appearing in Eq. (21) can take, in principle, any value, independently of D. This is similar to having a new manipulated variable that is independent of D and acts directly on the glucose concentration only. Since the value of the product $D(s_f - s)$ is fixed as the output of the glucose controller, the degree of non-interaction depends on the freedom of s_f to take a very high value when a very low D is required by the penicillin controller and vice versa. The same lines of thought apply to the PAA control scheme.

An alternative and more rigorous way of selecting the manipulated variables is through the examination of the Bristol array (Ray, 1981). Due to the existence of interactions, a unit step change in one of of the input variables, D, s_f, z_f, will result in steady state changes to all three of the controlled variables p, s and z. The steady state response of the system to unit step changes in each of the manipulated variables is presented in the form of a 3x3 matrix, called "Bristol" or "Relative Gain" Array.

When there are no interactions, each of the controlled variables is affected by only one of the manipulated variables. This results in a Bristol array with all the diagonal elements equal to one and all off-diagonal elements equal to zero. The pairing of the manipulated to the controlled variables is chosen so that the diagonal elements of the Bristol Array are closest to unity. Furthermore, if some of the diagonal elements take a negative value, then the closed loop process is not stable and an alternative pairing must be sought.

For the process studied in the present work, examination of the Bristol array at various fermentation times showed that the combination selected above is the most preferred.

4.2.2 Control Algorithm

Fast response of the controlled system, as well as overall stability can be achieved by using the proper control algorithm.

To speed up the penicillin transients the dilution rate must be kept low until the desired concentration has been reached. From that point on, the dilution rate must be continuously adjusted, in order to keep the concentration constant.

Since measurement of the penicillin concentration is relatively easy, a simple feedback controller will be adequate for this task. A simple Proportional plus Integral (PI) controller will be used.

Growth rate transients are fast and a steady state controller should be sufficient to provide the desired response. Constant growth rate implies constant or very slowly varying glucose concentration. Therefore, the controller equation is based on the steady state relationship resulting from Eq. (28)

$$s_f = s_d + \frac{\hat{\sigma} x}{D} \tag{36}$$

where s_d and $\hat{\sigma}$ are the estimated values of the glucose concentration and the specific uptake rate at the desired QSS operating conditions.

However, uncertainty in the model parameters could result in a slightly erroneous value for the specific uptake rate. This would drive the system to a different than the desired growth rate. A simple feedback term is introduced in the specific rate value, to account for small modelling errors. The corrected value is given by

$$\hat{\sigma}_c = \hat{\sigma} + K_{cs} (\mu_d - \mu) \tag{37}$$

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

Therefore, the final controller equation is

$$s_f = s_d + \frac{\hat{\sigma}_c\, x}{D} \tag{38}$$

The feedback correction term is expected to compensate for any uncertainties in estimated parameter values.

In the case that Contois kinetics apply, rather than Monod, the only difference is that the saturation constant k_s is proportional to the biomass concentration. Therefore, the glucose concentration must increase slowly for the growth rate to remain constant. The feedback correction term will modify the concentration of glucose being fed in order to maintain the desired growth rate.

The approach for the PAA controller is similar. A steady state controller is used, given by the equations

$$z_f = z_d + \frac{\beta\, \hat{q}_c\, x}{D} \tag{39}$$

$$\hat{q}_c = \hat{q}_p + K_{cz}\,(z_d - z) \tag{40}$$

where z_d is the desired PAA concentration, $\hat{q}_p$ is the estimated specific production rate of penicillin at the desired QSS, $\hat{q}_c$ is the corrected specific production rate and K_{cz} is the gain of the feedback term.

4.2.3 Constraints in the manipulated variables

The manipulated variables cannot take any value dictated by the control algorithm. Physical parameters like pump characteristics, size of tubing, etc., set an upper and a lower limit for the dilution rate. Hence the following physical constraint must be satisfied at all times

$$D_{\min} \leq D \leq D_{\max} \tag{41}$$

The solubility of glucose and other necessary ingredients of the medium set an upper limit to the glucose feed concentration

$$0 \leq s_f \leq s_{fmax} \tag{42}$$

A similar constraint holds for the PAA concentration

$$0 \leq z_f \leq z_{fmax} \tag{43}$$

The above constraints do not normally change during the course of a fermentation run. However, another, time varying, constraint is imposed on the dilution rate by the necessity to satisfy the growth rate controller (Eq. (38)).

The dilution rate is calculated first by the PI control equation and subsequently it is used for the calculation of glucose feed concentration. However, the value of D should be such that the calculated value of s_f lies within its physical constraints. Therefore, the upper limit for s_f implies a lower limit for the dilution rate, if Eq. (38) is to be satisfied at all times, namely

$$D_{minc} = \frac{\hat{\sigma}_c\, x}{s_f - s_d} \tag{44}$$

Since x changes throughout the fermentation, it follows that D_{minc} varies with time as well.

Therefore, the final form of the dilution rate constraint becomes

$$\max\{D_{\min}, D_{minc}\} \leq D \leq D_{\max} \tag{45}$$

This constraint must be computed at every sampling interval as part of the control calculations.

An additional constraint would be imposed by the PAA control equation. However, there is no demand for exact PAA control. Therefore, we prefer to set the constraint on the maximum PAA concentration, instead of the dilution rate. The result is a slightly slower response of the PAA concentration. However, any unnecessary extra dilution is avoided, so that the speed of the penicillin response is maximized.

5.0 STATE ESTIMATION USING A KALMAN FILTER

When the state variables for the model were selected, it was mentioned that CO_2 concentration in the exit gas was included so that the system could become observable. Measurement of c complements the measurement of penicillin and PAA in providing information about the entire set of state variables.

Under the condition that all measurements can be obtained on-line, without significant delay and with an acceptable accuracy, estimates of all the state variables can be calculated, provided that the system is "observable". The mathematical technique that can help us with this task is the Extended Kalman Filter. This is a state estimation method that uses on-line measurements to provide estimates of the state variables. These estimates are a weighted average of the model predictions and the information retrieved from the measured data.

In the Kalman Filtering Theory, the process under study is considered to be corrupted by unidentified disturbances or modelling errors with statistics that can be determined. As a result, instead of the exact value of a given state variable, a most probable value and a corresponding probability distribution are defined at every point in time. The probability distribution widens with time if actual data are not available, reflecting the decreasing confidence in the model predictions due to the accumulating effect of the noise processes. When data are being collected the distribution narrows with time at a rate dependent on the statistics of the measurement error and the process noise. At any given time, the most probable value of any state variable and the corresponding distribution are a function of all the data points collected up to that time, from all the measured variables, directly or indirectly related to the given state variable. This is the main advantage of the Kalman Filter over techniques which use only current data and only directly related measurements for the estimation of the state vector.

It should be noted here that most of the results in estimation theory have been developed for linear systems. Nonlinearities make mathematical manipulations very difficult. The usual approach is to linearize the model around a desired trajectory. Subsequently, the linear system estimation theory can be applied on the linearized model. In the next paragraphs we shall develop the linearized model of our process. We will also introduce the matrix notation, which will be very useful in the subsequent mathematical manipulations.

5.1 The Linearized Model

5.1.1 Linearization

The set of nonlinear differential equations that describe the dynamic characteristics of our system can be written in the form

$$\dot{\underline{x}} = \underline{f}\,(\underline{x}(t), \underline{u}(t)) \;\; ; \;\; \underline{x}(0) = \underline{x}_0 \qquad (46)$$

$$\underline{y} = \underline{h}\,(\underline{x}) \qquad (47)$$

where $\underline{x} = [x, s, p, \lambda, z, c\,]^T$ is the 6-dimensional vector of the state variables, $\underline{u} = [D, s_f, z_f\,]^T$ is the 3-dimensional vector of the manipulated variables and $\underline{y} = [p, z, c\,]^T$ is the 3-dimensional vector of the output variables (measurements). The set of differential equations is represented by the 6-dimensional vector $\underline{f}$ and the measuring equations by the 3-dimensional vector $\underline{h}$.

A QSS is mathematically described by a nominal trajectory $\underline{x}_{QSS}(t)$ of the state variables. The linearization is performed around the desired trajectory by employing Taylor series expansion of the nonlinear terms. The produced second and higher order terms are neglected under the assumption that operating conditions are close enough to the QSS. The resulting linearized model has the form

$$\delta\dot{\underline{x}} = \underline{A}\ \delta\underline{x}(t) + \underline{B}\ \delta\underline{u}(t) \ \ ; \ \ \delta\underline{x}(0) = \delta\underline{x}_o \qquad (48)$$

$$\delta\underline{y} = \underline{C}\ \delta\underline{x} \qquad (49)$$

where

$$\delta\underline{x}(t) = \underline{x}(t) - \underline{x}_{QSS}(t) \ \ , \ \ \delta\underline{u}(t) = \underline{u}(t) - \underline{u}_{QSS}(t) \text{ and } \delta\underline{y}(t) = \underline{y} - \underline{y}_{QSS}(t) \qquad (50)$$

The vectors $\underline{x}_{QSS}$, $\underline{u}_{QSS}$ and $\underline{y}_{QSS}$ are the QSS values of the corresponding variables. It is reminded at this point that the values of these variables vary with time, since the biomass concentration increases at all times. The subscript QSS and the prefix δ will be omitted in the subsequent derivations, for the sake of simplicity.

The matrices $\underline{A}_{(6\times 6)}$, $\underline{B}_{(6\times 3)}$ and $\underline{C}_{(3\times 6)}$ are given by Jacobians

$$\underline{A}(t) = \left[\frac{\partial \underline{f}}{\partial \underline{x}}\right]_{QSS}, \quad \underline{B}(t) = \left[\frac{\partial \underline{f}}{\partial \underline{u}}\right]_{QSS}, \quad \underline{C}(t) = \left[\frac{\partial \underline{h}}{\partial \underline{x}}\right]_{QSS} \tag{51}$$

The "process matrix" $\underline{A}$ contains all the interactions among the state variables and has the form

$$\underline{A} = \begin{bmatrix} \mu & x\theta_1 & 0 & 0 & 0 & 0 \\ -\sigma & -\frac{\theta_1}{Y_g}x - D & 0 & -\frac{\theta_2}{Y_p}x & -\frac{\theta_3}{Y_p}x & 0 \\ q_p & 0 & -D & x\theta_2 & x\theta_3 & 0 \\ 0 & -\lambda\theta_1 & 0 & -\mu & 0 & 0 \\ -\beta q_p & 0 & 0 & -\beta x\theta_2 & -\beta x\theta_3 - D & 0 \\ vq_c & \frac{v\theta_1}{k_4}x & 0 & vk_6x\theta_2 & vk_6x\theta_3 & -D_g \end{bmatrix} \tag{52}$$

where $\theta_1 = \frac{\partial \mu}{\partial s}$, $\theta_2 = \frac{\partial q_p}{\partial \lambda}$ and $\theta_3 = \frac{\partial q_p}{\partial z}$. All the elements of matrix $\underline{A}$ are evaluated at QSS conditions.

The "control matrix" $\underline{B}$ contains the effects of the manipulated variables on the state variables and is given by

$$\underline{B} = \begin{bmatrix} 0 & 0 & 0 \\ s_f - s & D & 0 \\ -p & 0 & 0 \\ 0 & 0 & 0 \\ z_f - z & 0 & D \\ 0 & 0 & 0 \end{bmatrix} \tag{53}$$

Finally, the relationship between the measured variables and the state variables is represented by the "observation matrix" $\underline{C}$ which is given below

$$\underline{C} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{54}$$

The linearized form of the proposed model will be used in the next section in order to determine if the three available measurements (concentrations of PAA, penicillin and CO_2 in the effluent gas) are sufficient for the estimation of the entire set of the state variables.

5.1.2 Observability

The observability of our system can be examined by applying the condition for linear systems on the linearized model. However, since the matrices $\underline{A}$ and $\underline{B}$ are not constant, the observability condition should be examined over a grid of points in time. If the linearized system is observable over the entire grid it can be expected that the nonlinear system is observable as well.

It should be made clear at this point, that observability is not a property of the model only, but depends on the number and the choice of the measured variables. This is reflected on the dependence of the observability condition for the linearized model on the observation matrix $\underline{C}$.

For the linearized system to be observable the (6 x 18) observability matrix $\underline{L_o}$ should have rank equal to six at all times. The matrix $\underline{L_o}$ is defined as

$$\underline{L_o} = [\, \underline{C}^T \,|\, \underline{A}^T \underline{C}^T \,|\, (\underline{A}^T)^2 \underline{C}^T \,|\ldots|\, (\underline{A}^T)^5 \underline{C}^T \,] \tag{55}$$

The observability condition was examined for the system under study. It was assumed that three measurements were available on-line, namely penicillin concentration, PAA concentration and CO_2 concentration in the effluent gas. The model was linearized along a Quasi-Steady-State trajectory and the observability matrix was checked for a grid of points in time along the trajectory. The rank of the matrix $\underline{L_o}$ was found to be equal to 6 at all times. Therefore, the nonlinear system is expected to be observable in the neighborhood of the QSS.

5.2 The Stochastic model of the process

State estimation relies on the knowledge of the inputs to the process (manipulated variables, disturbances) and the measured outputs. However, the values of the manipulated variables can never be precisely what they were set to, because of interfering "noise" processes that cause small fluctuations around the desired value. For example, the dilution rate or equivalently the feed flowrate, can never be manipulated with a precision greater than the precision of the Digital to Analog (D/A) converter, which translates the computer signal to a voltage signal driving the pump. Furthermore, the state variables themselves deviate from their expected values due to small modelling errors or disturbances unaccounted for by the model. Finally, the measurement of the output variables is always subject to experimental error that is introduced during the sample analysis.

A model that incorporates these "noise" processes is called "stochastic". The values of the state variables are interpreted as the most probable values, exactly in the same manner as the measurements are considered to be the most probable estimates of the the process outputs. Moreover, the probability distribution of the state variables depends on the statistics of the experimental errors and the statistics of the process noise. In a stochastic model, the equations are written in the form

$$\dot{\underline{x}} = \underline{f}(\underline{x},\underline{u}) + \underline{\xi}(t) \qquad (56)$$

$$\underline{y}(t_k) = \underline{h}(\underline{x}(t_k),t_k) + \underline{\eta}_k \qquad (57)$$

$$\underline{x}(0) = \underline{x}_o + \underline{\xi}_o \qquad (58)$$

where $\underline{\xi}(t)$ is a 6-dimensional vector of process noise, $\underline{\eta}(t)$ is a 3-dimensional vector of random measurement noise, $\underline{x}_o$ is an estimate of the initial state and $\underline{\xi}_o$ is its random error. Since we are interested in computer control, sampling is discrete and Eq. (57) is written in its discrete form, k being the sampling interval counter.

The stochastic model is used in the derivation of the equations for the Extended Kalman Filter, presented in the next section.

5.3 Extended Kalman Filter equations

The prediction equations of the First Order Extended Kalman Filter for a Non-Linear system with discrete sampling are (Ray, 1981)

$$\dot{\hat{\underline{x}}}(t\,|\,t_{k-1}) = \underline{f}(\hat{\underline{x}},t) \qquad (59)$$

$$\dot{\underline{P}}(t\,|\,t_{k-1}) = \hat{\underline{A}}\,\underline{P} + \underline{P}\,\hat{\underline{A}}^T + \underline{R}^{-1} \qquad (60)$$

The first equation provides the model estimates of the state variables, $\hat{\underline{x}}$, at the time t, based on the data collected up to the point t_{k-1}, hence the notation $t\,|\,t_{k-1}$.

The covariance matrix, $\underline{P}$, of the state estimates is calculated by the second equation from its value at the previous sampling point and the covariance $\underline{R}^{-1}$ of the process noise. The initial value $\underline{P}_o$ of the covariance matrix $\underline{P}$ is supplied by the user and reflects his confidence in the initial state estimates, which have to be supplied as well.

At every sampling interval, a set of measurements is taken and the model predictions and the corresponding matrix of covariances are updated by using the following equations

$$\hat{\underline{x}}(t_k|t_k) = \hat{\underline{x}}(t_k|t_{k-1}) + \underline{K}\,[\underline{y}(t_k) - \underline{h}(\hat{\underline{x}}(t_k|t_{k-1}, t_k)] \tag{61}$$

$$\underline{P}(t_k|t_k) = \underline{P}(t_k|t_{k-1}) - \underline{K}(t_k)\,\underline{C}\,\underline{P}(t_k|t_{k-1}) \tag{62}$$

where

$$\underline{K}(t_k) = \underline{P}(t_k|t_{k-1})\underline{C}^T\,[\underline{C}\,\underline{P}(t_k|t_{k-1})\underline{C}^T + \underline{Q}^{-1}]^{-1} \tag{63}$$

$\underline{Q}^{-1}$ is the covariance matrix of the experimental measurements.

As it can be seen from Eq. (61), the updated (optimal) estimate of the state variables is made out of two parts, one representing the model predictions, $\hat{\underline{x}}(t_k|t_{k-1})$ and the second representing the contribution of the experimental measurements. The Kalman Filter gain matrix $\underline{K}$ provides the optimal weighting between these two contributions and its value is determined by the user specified uncertainty in the measurements and the model through the matrices $\underline{Q}$, $\underline{R}$ and $\underline{P}_o$.

To implement the filter, the model parameters must be estimated first and then the statistics of the process noise under given operational conditions as well as the statistics of the measurement errors must be determined. However, before the implementation in an actual fermentation, the filter can be evaluated by simulation runs and tested along with the designed controller. The evaluation of the entire control system on the basis of simulation runs performed on a digital computer will be the subject of the next Chapter.

6.0 EVALUATION OF CONTROL SYSTEM

Computer simulation is an excellent tool in assessing the effectiveness of the design of a control system prior to its implementation on the physical system. In this chapter the designed control system will be evaluated, based on simulation runs performed on a digital computer (CDC Cyber 175). First the controller performance will be tested.

6.1 Performance of Controller

"Isolation" of the controller performance can be achieved by assuming that "perfect" measurements can be made available to it. This translates into assuming negligible experimental errors and no delay in the measurements. Furthermore, since a computer control scheme is being tested, sampling is not considered continuous but rather discrete, with a sampling interval of 0.5 *h*. Similarly the control action is discrete and is implemented only at every sampling point.

6.1.1 Response to set point changes

Let's assume that the system is at a QSS where penicillin concentration is 2.0 *g/L*, PAA concentration is 0.5 *g/L* and the growth rate is 0.01 h^{-1}. The biomass at time t = 0 is taken equal to 5 *g/L*. At time t = 20 *h* we change the growth rate set point to 0.015 *g/L*, while maintaining the other set points constant. The response of the controlled process is very fast and the new QSS has been reached within

2 to 3 *h* (Fig. 3). Practically no interaction can be observed after this set point change.

Subsequently, at time t = 40 *h* a step increase in the penicillin set point is made. The response of penicillin concentration is much faster compared to the open loop case. The desired new level has been reached within approximately 30 hours, while in the open loop run (Fig. 2), more than 100 hours were required for a set point change of half the size. Furthermore, no interaction between penicillin and growth rate can be observed. In the open loop run the penicillin transient had induced a very strong growth rate transient.

At time t = 100 *h* the set point of PAA concentration is changed to 0.8 *g/L* from 0.5 *g/L*. The rise time is very short (approximately 7 hours) and no interaction problems appear. In this and the two previous runs the control objectives have been met. Next we examine the response of the system to negative step changes in the set points, although such a change in the penicillin concentration is not of particular interest.

In Fig. 4, we can see that for negative set point changes no interaction problem appears. A difference with the previous run though exists in the response of penicillin concentration being much faster. The reasons of the difference can be better understood when the behavior of the manipulated variables during the set point changes is examined.

For positive set point changes the behavior of the glucose feed

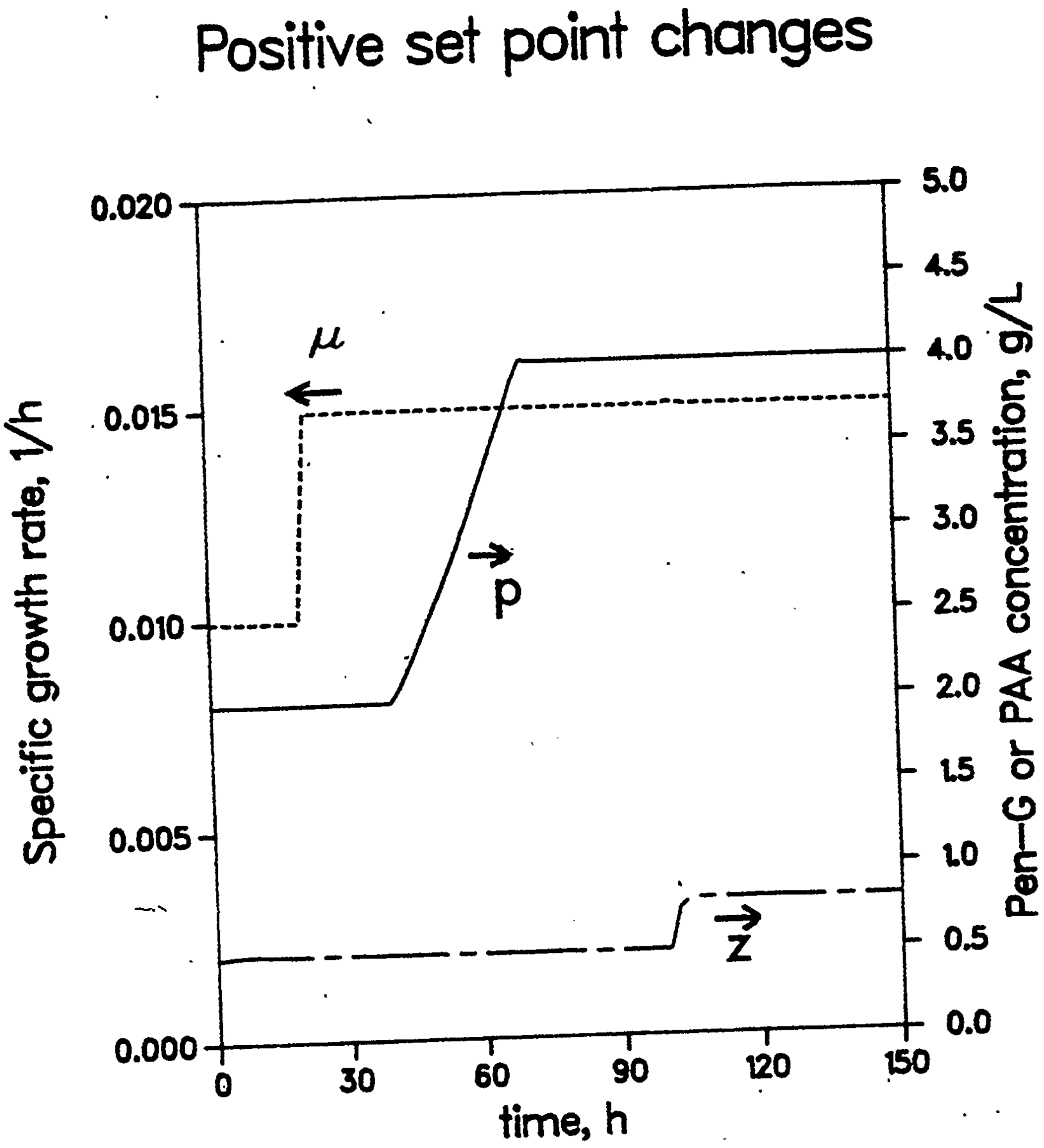

FIGURE 3 Simulated response of μ, p and z to positive set point changes.

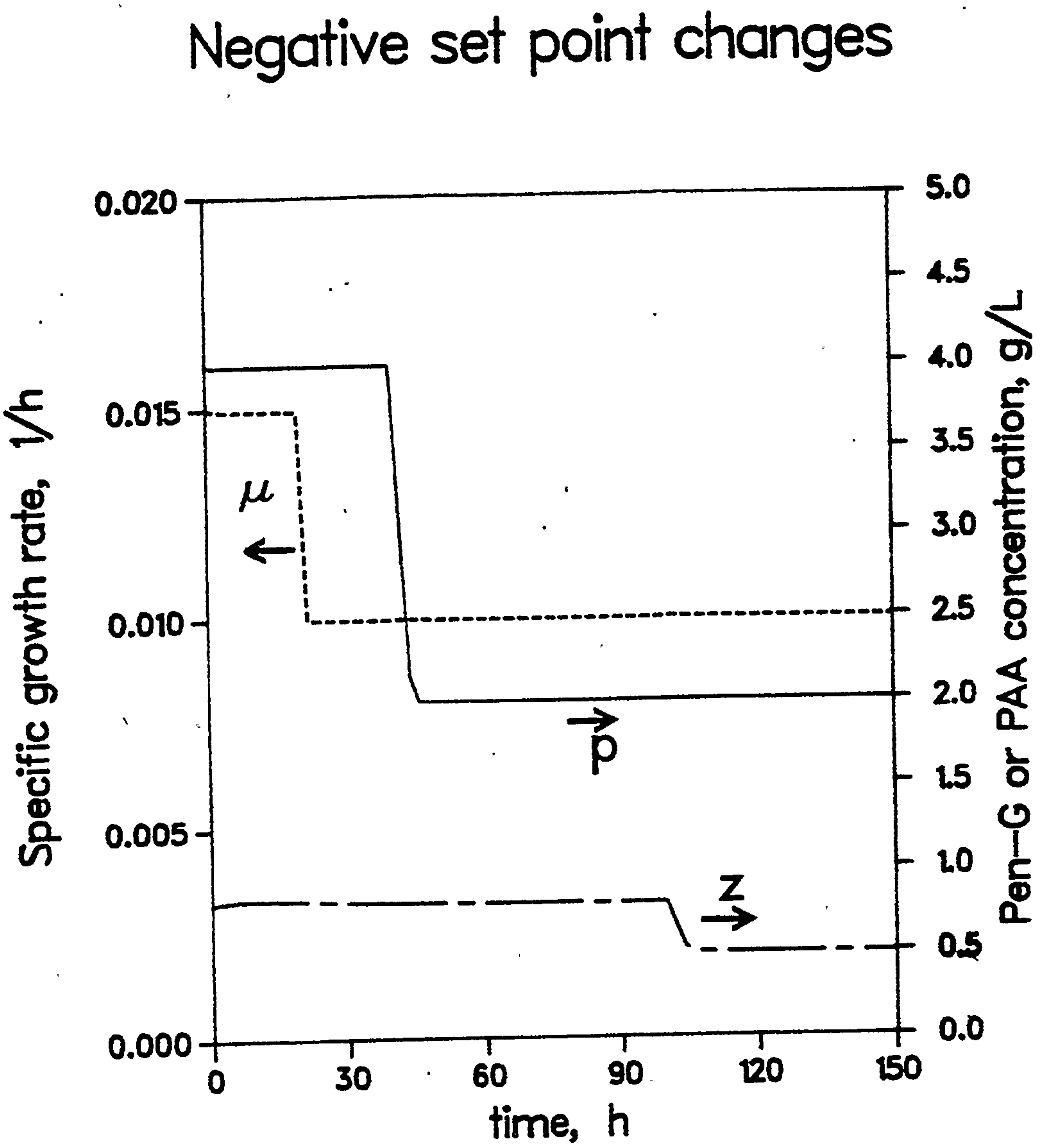

FIGURE 4 Simulated response of μ, *p* and *z* to negative set point changes.

concentration and the dilution rate is shown in Fig. 5. When at time t = 20 h a higher growth rate is required, glucose feed concentration is increased by a small step to produce a slightly higher glucose concentration in the reactor. At the same time the dilution rate rises exponentially following the exponential rise of biomass. Moreover, from time t = 20 h on, D starts rising faster, responding to a higher growth rate.

At time t = 40 h, a higher penicillin concentration is demanded. Immediately the dilution rate drops to its minimum value allowing a fast accumulation of penicillin in the reactor. At the same time the glucose feed concentration takes its maximum value. For the next few hours the dilution rate follows the computed minimum constraint and can be seen to rise slightly. When the penicillin concentration is close to the desired value, dilution rate and glucose feed concentration move fast to their QSS values. From that point on, s_f remains constant, while D continues to rise exponentially.

When a negative change in penicillin concentration is desired the dilution rate takes its maximum value (0.2 h^{-1}), in order to dilute the reactor content as fast as possible. We can see this response in Fig. 6. At the same time glucose feed concentration takes a very low value. After the transient is over, both control variables take the values that correspond to the new QSS.

It is obvious now that during a transient penicillin accumulation depends mostly on the production term $q_p x$. The product $-Dp$ is very

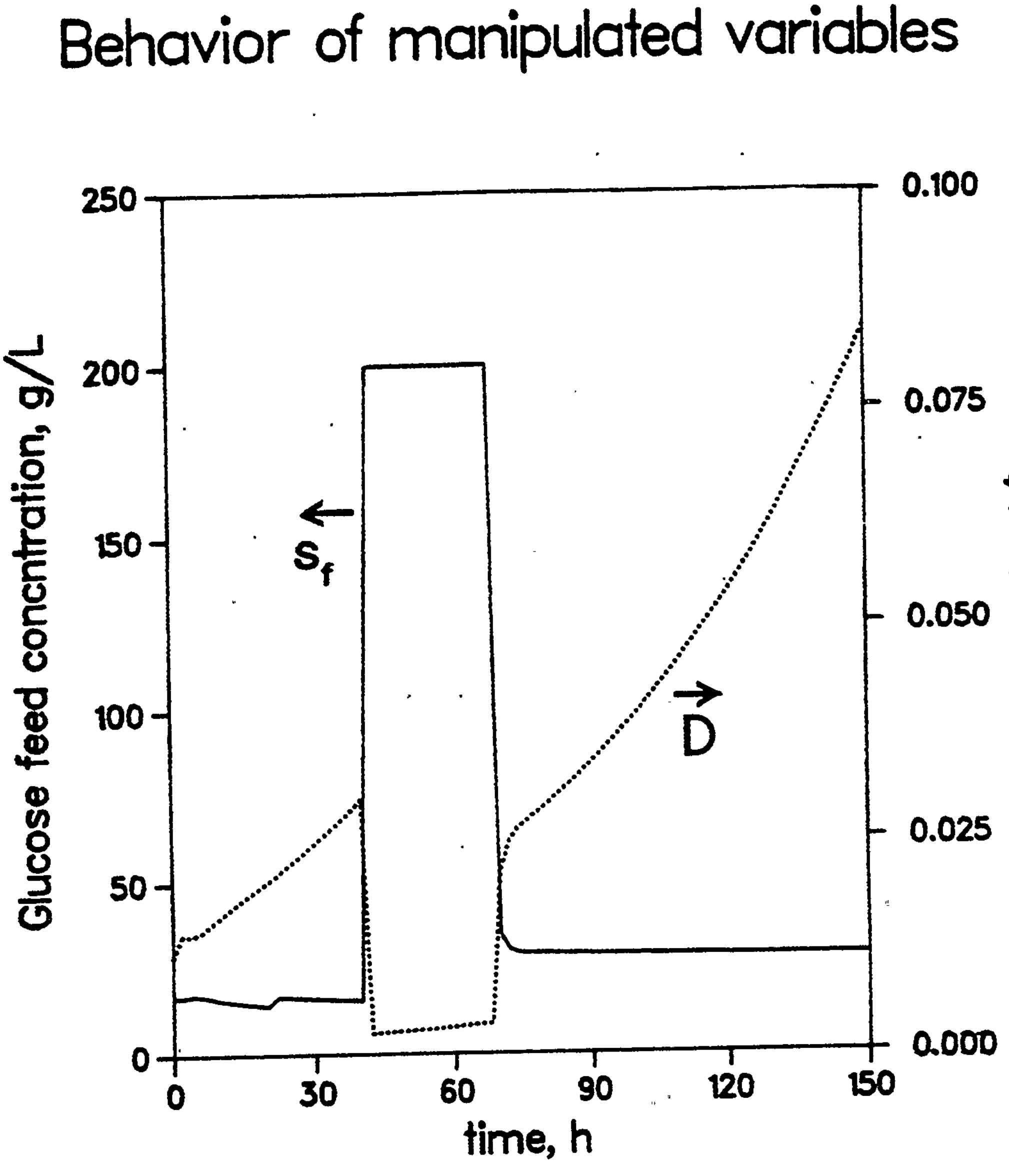

FIGURE 5 Simulated behavior of the manipulated variables s_f and D after positive set point changes in μ, p and z.

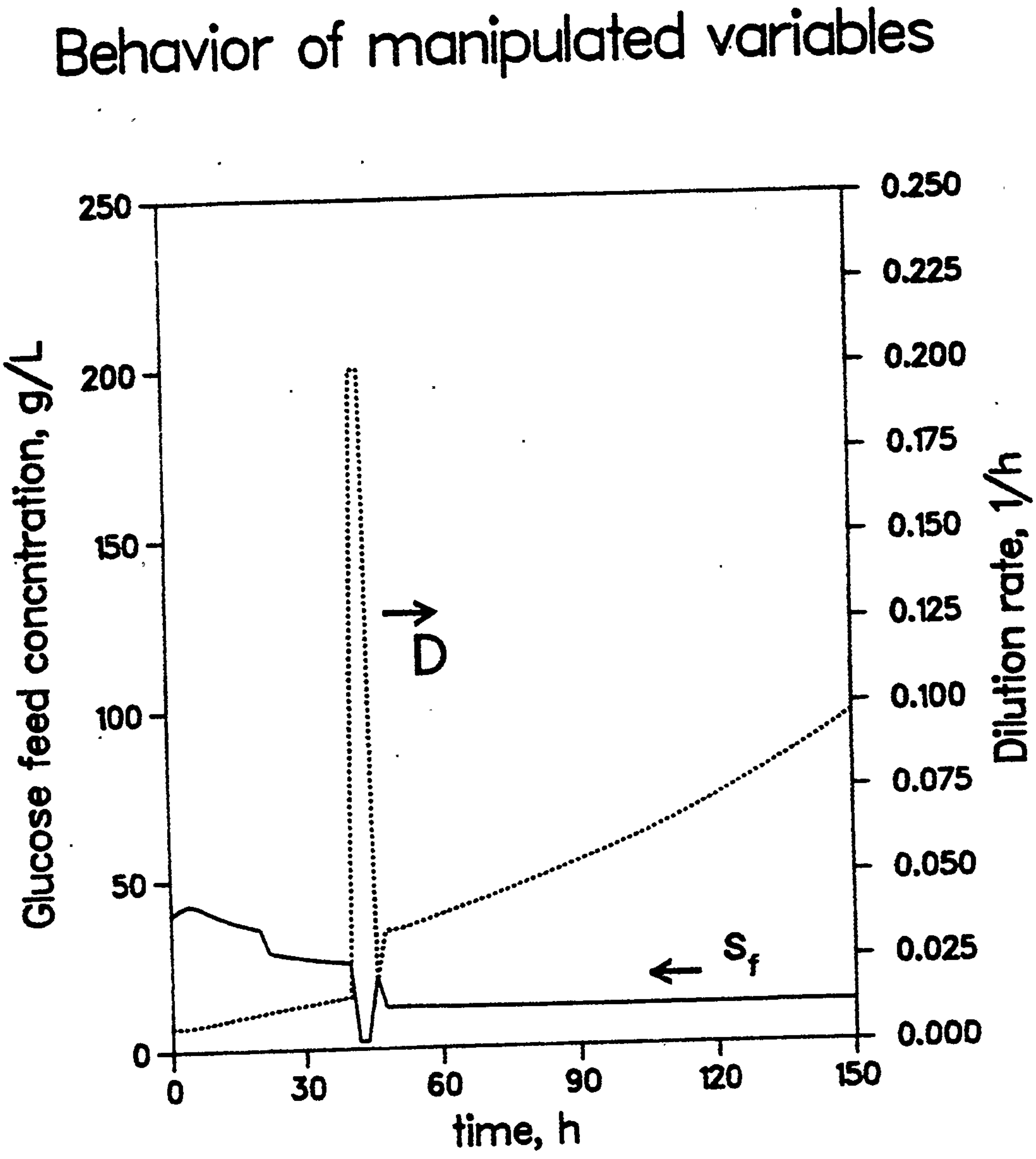

FIGURE 6 Simulated behavior of the manipulated variables s_f and D after negative set point changes in μ, p and z.

small, because of the low dilution rate. Therefore, the positive penicillin transients are necessarily slow, due to the small value of q_p. The situation is worst when the biomass concentration is low or the growth rate is such, that the productivity is low.

Negative penicillin transients are expected to be very fast, because dilution rate can take a high value. The upper limit of D used in the simulation was 0.2 h^{-1}. With a biomass concentration of 10 g/L and penicillin concentration of 4 g/L the magnitude of the term Dp is 0.4 g/Lh. The product $q_p x$ gives a small 0.08 g/Lh in the best case (when q_p takes its maximum value).

After having made the above observations, a final test is done, to examine the response of the controller to simultaneous set point changes. The results shown in Fig. 7 indicate very good behavior of the controlled variables even in this demanding situation.

6.1.2 Controlled Production Run

During a production run the penicillin set point is set to a desired high concentration and the growth rate set point is set at a value where the productivity is expected to be optimum.

The control system should drive the growth rate rapidly to its optimum value, in order to achieve the shortest possible rise time for the penicillin concentration. For the same reason, dilution rate must be kept at its minimum value until the desired penicillin concentration has been reached. Finally, during the entire run, PAA

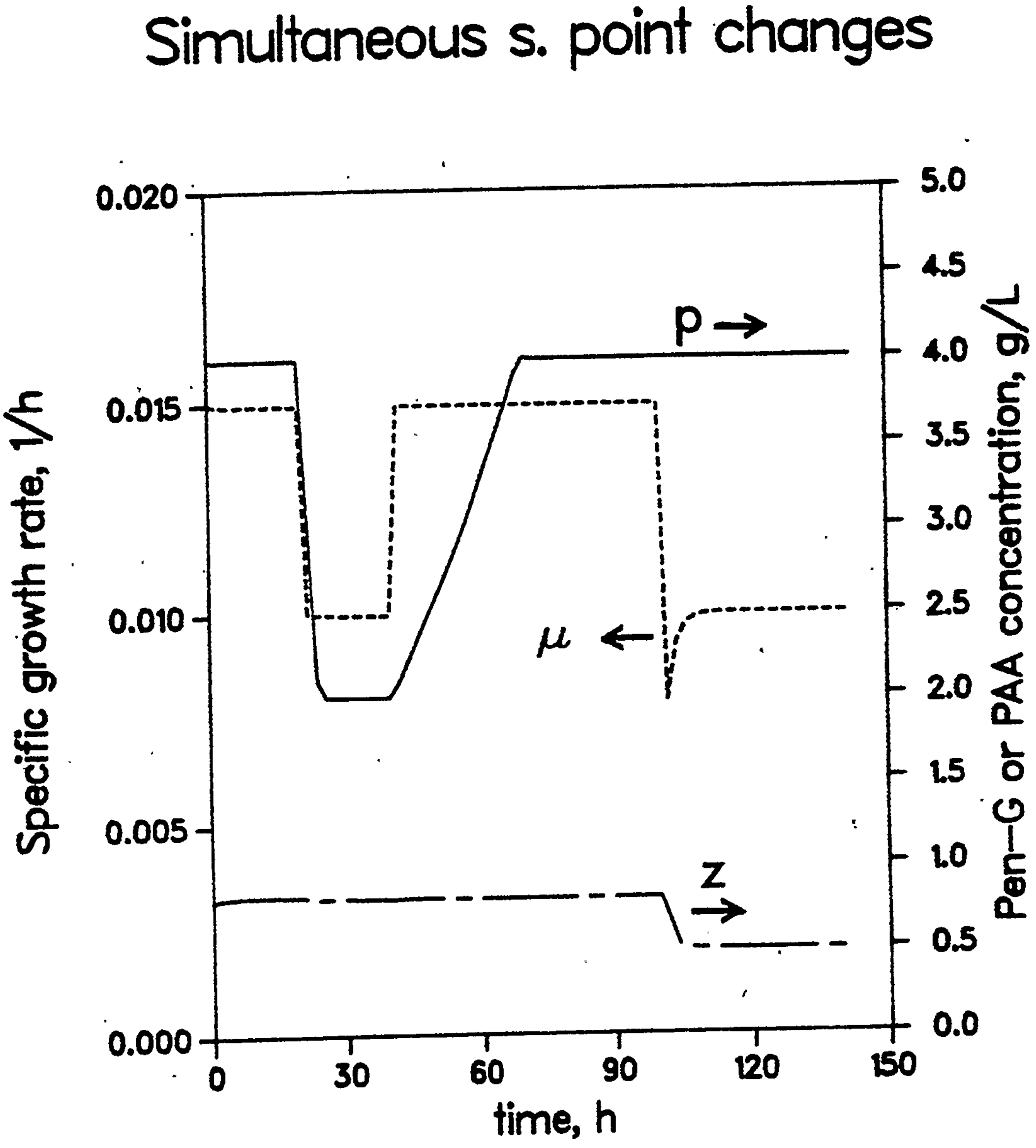

FIGURE 7 Simulated response of μ, *p* and *z* to simultaneous set point changes.

concentration must be regulated at an acceptable level.

The profiles of controlled variables during a simulated production run are shown in Fig. 8. The starting biomass concentration has been taken as 3 g/L. The low biomass value can explain the relatively long rise time of the penicillin concentration. The response of growth rate is very fast, as expected and the PAA response is satisfying. Faster PAA response is inhibited by the low solubility of PAA in the feeding medium. An additional PAA feed could have been used but this would contribute to a higher dilution rate and even slower response of the penicillin concentration.

The profiles of glucose feed concentration and dilution rate are shown in Fig 9. We can see that they exhibit the expected behavior. It is interesting to see the behavior of the PAA feed concentration during the same production run (Fig 10). In the beginning PAA is fed at the maximum concentration because its concentration in the reactor is low and, in addition, the dilution rate is at its minimum value. When the desired PAA level in the reactor has been reached, the feed concentration drops to a lower value. At approximately 90 h into the run the dilution rate takes a higher value that corresponds to the desired QSS. In response to this change, PAA feed concentration drops to its QSS value as well.

6.2 Performance of extended Kalman Filter

The role of the Kalman filter is to provide us with estimates of the

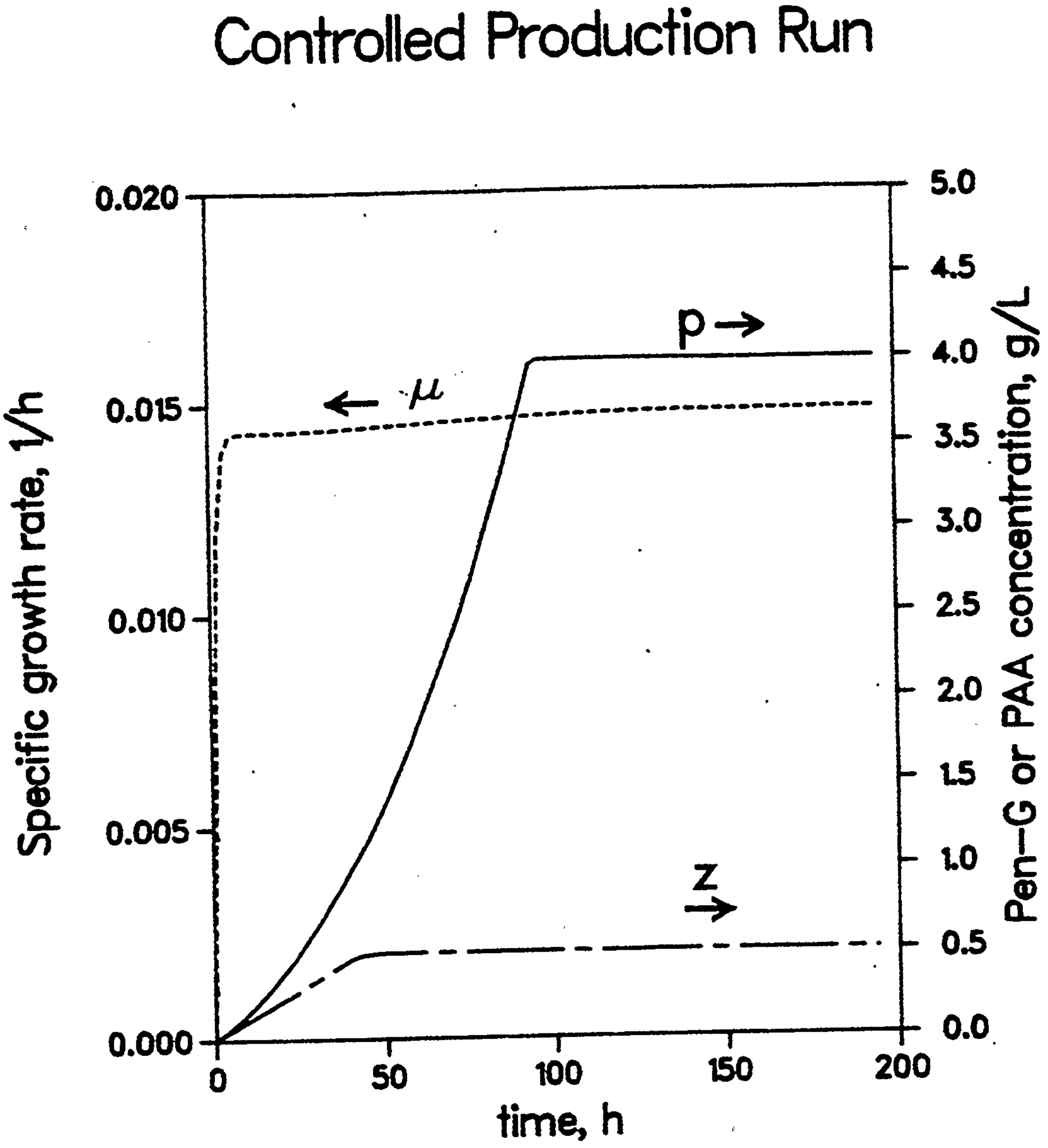

FIGURE 8 Simulated profiles of μ, p and z during a controlled production run.

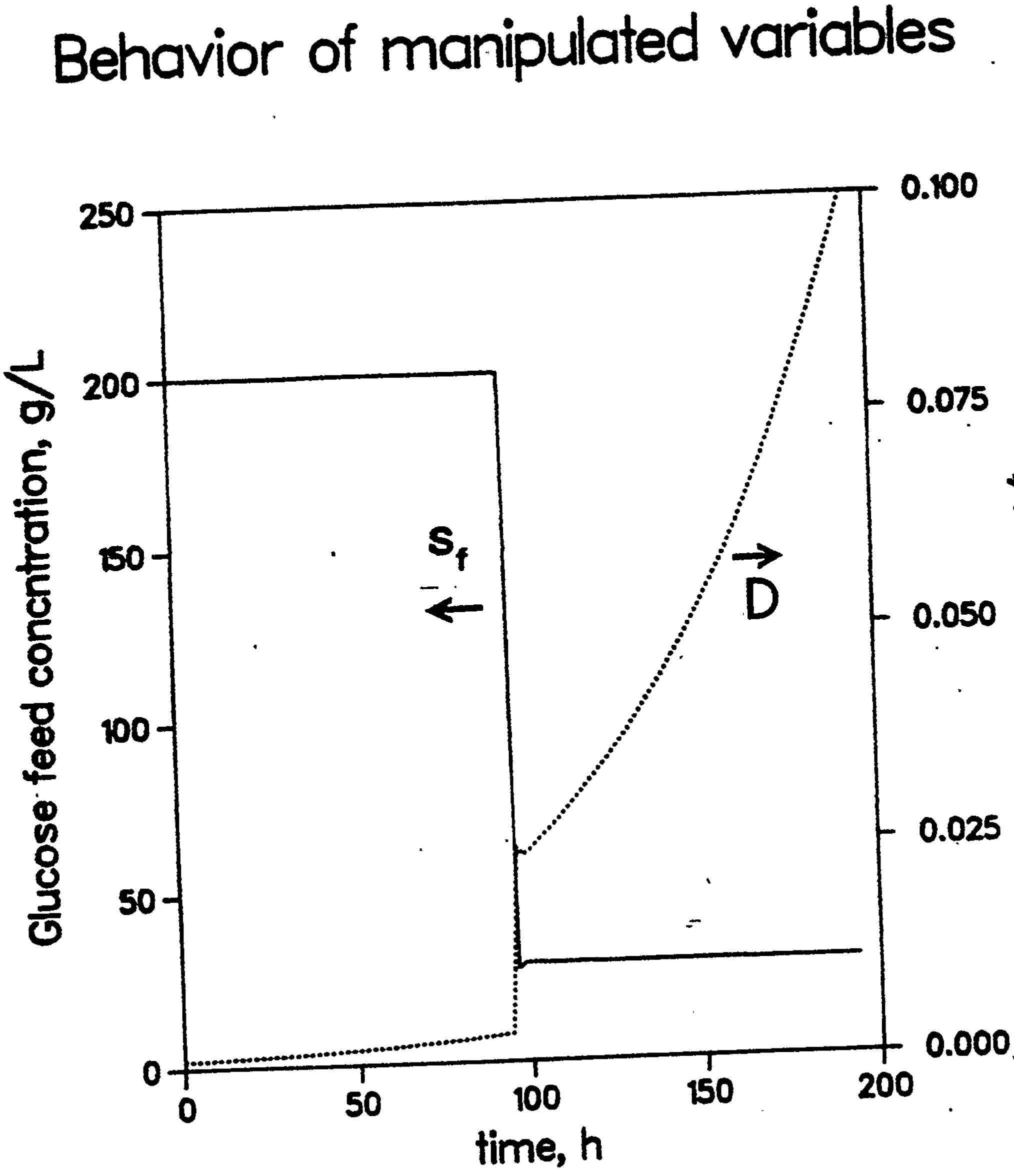

FIGURE 9 Simulated profiles of the manipulated variables s_f and D during a controlled production run.

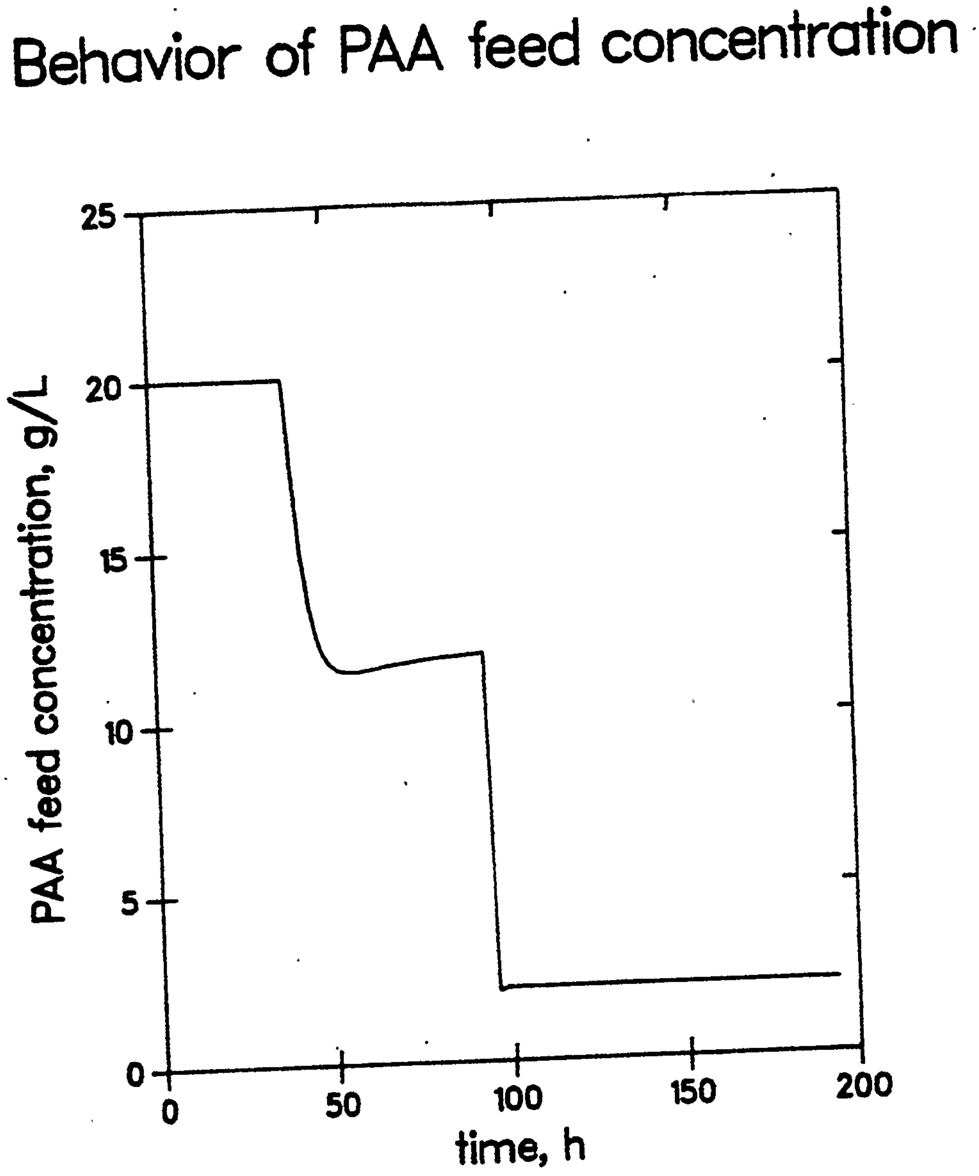

FIGURE 10 Simulated profile of the manipulated variable z_f during a controlled production run.

state variables, which are as as accurate as possible. We can expect that the accuracy increases with time as more and more data are being collected. In order to test the performance of the filter we must implement it in a simulation run and monitor both the "real" and estimated values of the state variables.

6.2.1 Implementation of the Filter

In order to approach a real situation the measurements are considered to be corrupted by noise, due to experimental errors. The experimental covariances have been given realistic values. The covariance of CO_2 measurement is taken equal to 0.0001 g/L corresponding to 1% standard error of measurement and for penicillin and PAA (measured with the same method) equal to 0.005 g/L corresponding to 1% standard measurement error.

The initial state estimate has been taken to be approximately 20% off the real value, in order to test the ability of the filter to converge to the real states. Accordingly, the covariance of the error in the initial estimates has been given a big value, to denote that our confidence in the initial estimate is rather small.

The model equations have been numerically integrated with the help of the IMSL subroutine DGEAR . The filter covariance equations - Eq. (60)- have been solved approximately. The reason is that this algorithm is destined to be implemented on a microcomputer. Therefore, the volume of computations should be as small as possible, for optimum speed.

To solve approximately the filter covariance matrix differential equations it is assumed that the matrices $\underline{P}$, $\underline{A}$ and $\underline{R}$ take a constant value within a sampling interval. In this case, if the value of the matrices at time t_{k-1} is known, the value of $\underline{P}$ at time t_k can be obtained from the matrix equation

$$\underline{P}(t_k | t_{k-1}) = e^{\underline{A}\Delta t}\underline{P}\,(e^{\underline{A}\Delta t})^T + \underline{R}^{-1}\Delta t \qquad (64)$$

where Δt is the sampling interval and $e^{\underline{A}\Delta t}$ is the (6 x 6) transition matrix approximated by the power series

$$e^{\underline{A}\Delta t} = \sum_{k=0}^{\infty}\frac{(\underline{A}\Delta t)^k}{k!} = \underline{I} + \frac{\underline{A}\Delta t}{1!} + \frac{\underline{A}^2(\Delta t)^2}{2!} + \frac{\underline{A}^3(\Delta t)^3}{3!} + \cdots \qquad (65)$$

6.2.2 Simulation Results

The results of a typical simulation run are shown in figures 11 to 16. In Fig. 11 the profiles of the real (intermittent line) and the estimated biomass concentration during a 250 *h* run are shown. The estimate converges very fast to the real value and agrees very well until the end of the run.

In Fig. 12, the profiles of glucose concentration are given. The coincidence is not so close as in the case of the biomass, however, the response of the controller is still very fast. The growth rate is proportional to the glucose concentration at these low glucose levels. Therefore, the growth rate profile is expected to be similar

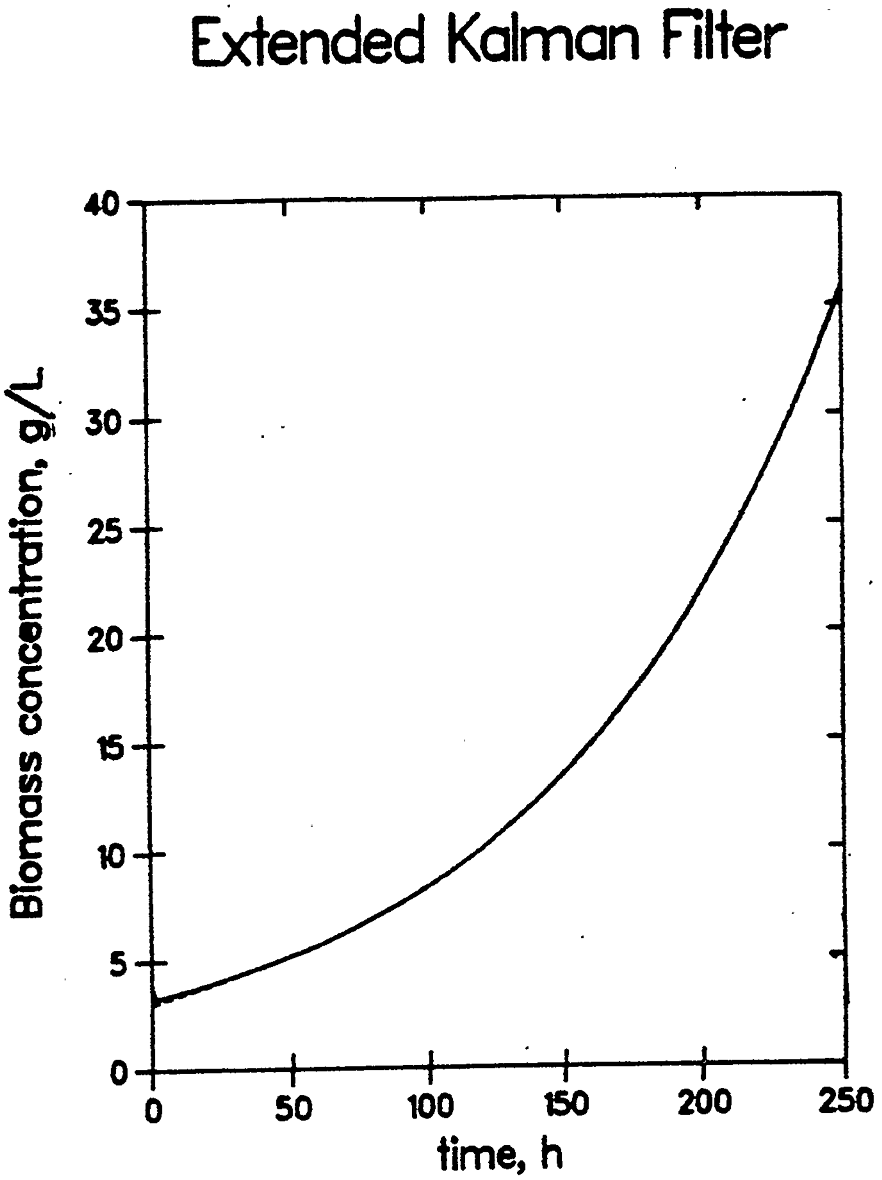

FIGURE 11 Simulated profiles of actual biomass concentration (---) and its Kalman Filter Estimate during a production run.

to the one of the glucose. Indeed, this is the case, as it can be seen in Fig. 15.

The penicillin profiles in Fig. 13 coincide during the entire run. This is expected, since a very accurate measurement is available to the filter. Because of the close agreement between estimate and real value, the response of the penicillin concentration is as fast as in the ideal case. PAA concentration profiles in Fig. 14 agree very well. However small fluctuations can be observed.

The estimated and the "real" value of the average cell age is shown in Fig. 16. The convergence is not as fast as for the previous variables. There is a simple explanation for this. Average cell age is only indirectly coupled to the measurements, through the growth rate which behaves like a filter estimate itself.

In conclusion, the Kalman filter was capable to accurately estimate the entire set of state variables, during a production run, on the basis of the penicillin, PAA and CO_2 measurements.

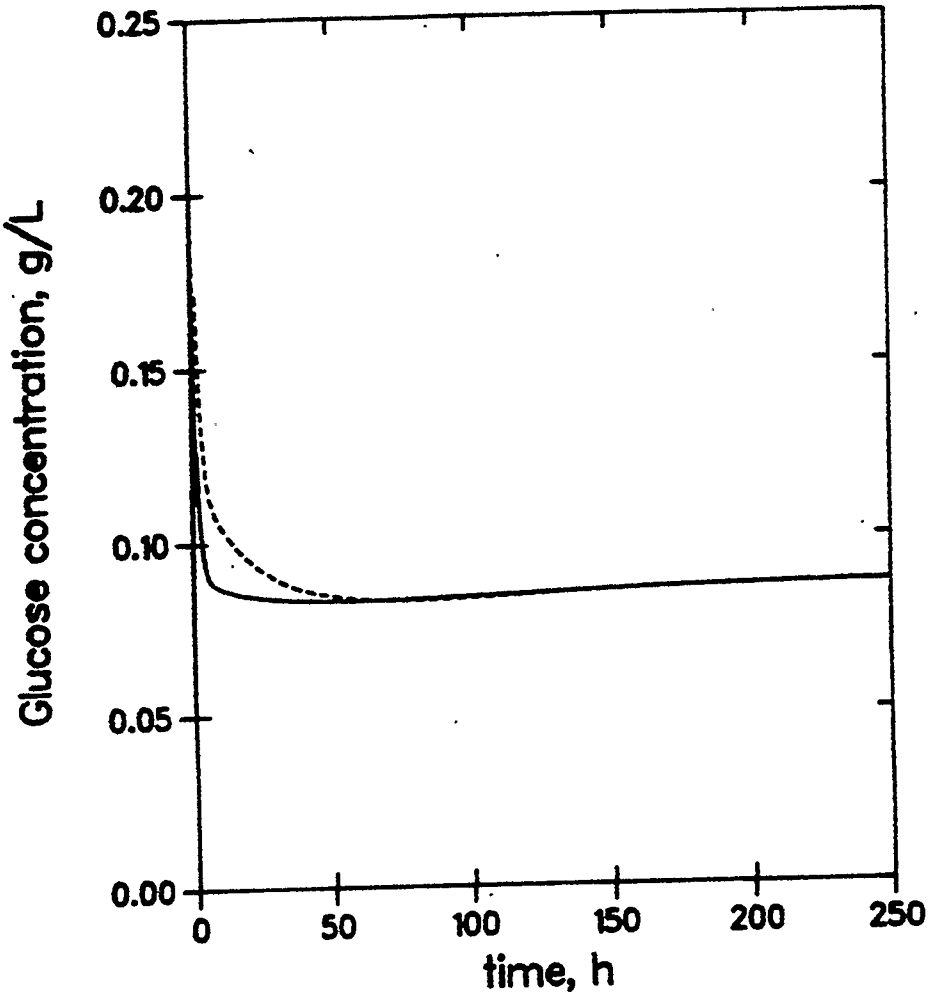

FIGURE 12 Simulated profiles of actual glucose concentration (---) and its Kalman Filter Estimate during a controlled production run.

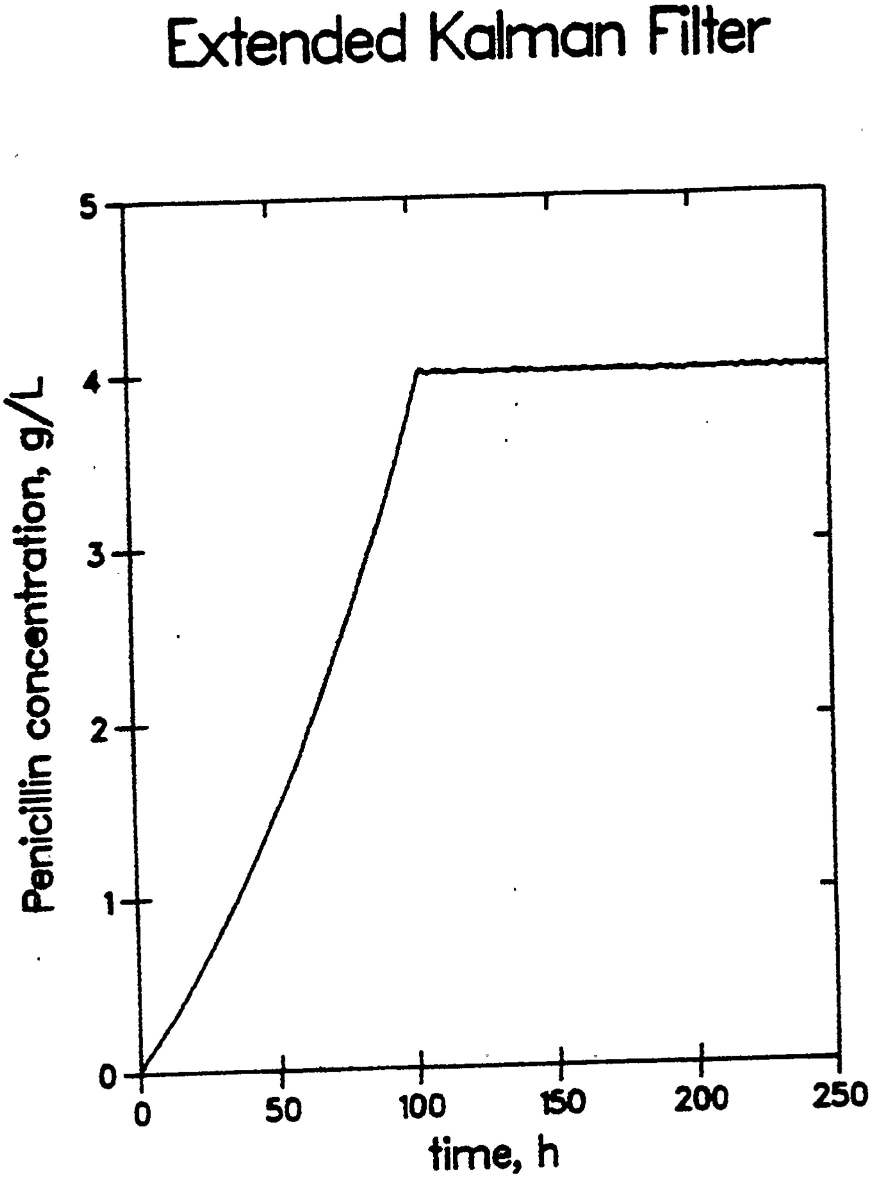

FIGURE 13 Simulated profiles of actual penicillin concentration (---) and its Kalman Filter Estimate during a controlled production run.

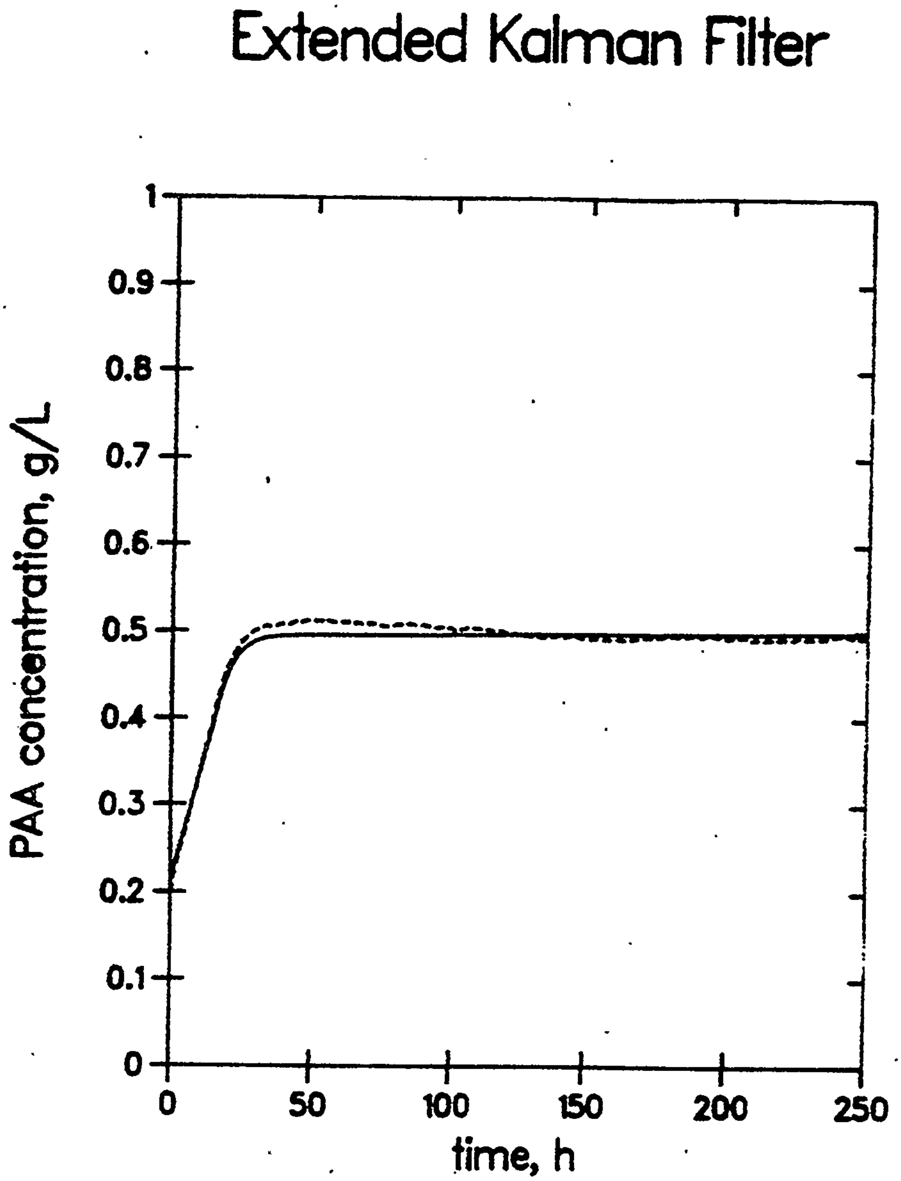

FIGURE 14 Simulated profiles of actual PAA concentration (---) and its Kalman Filter Estimate during a controlled production run.

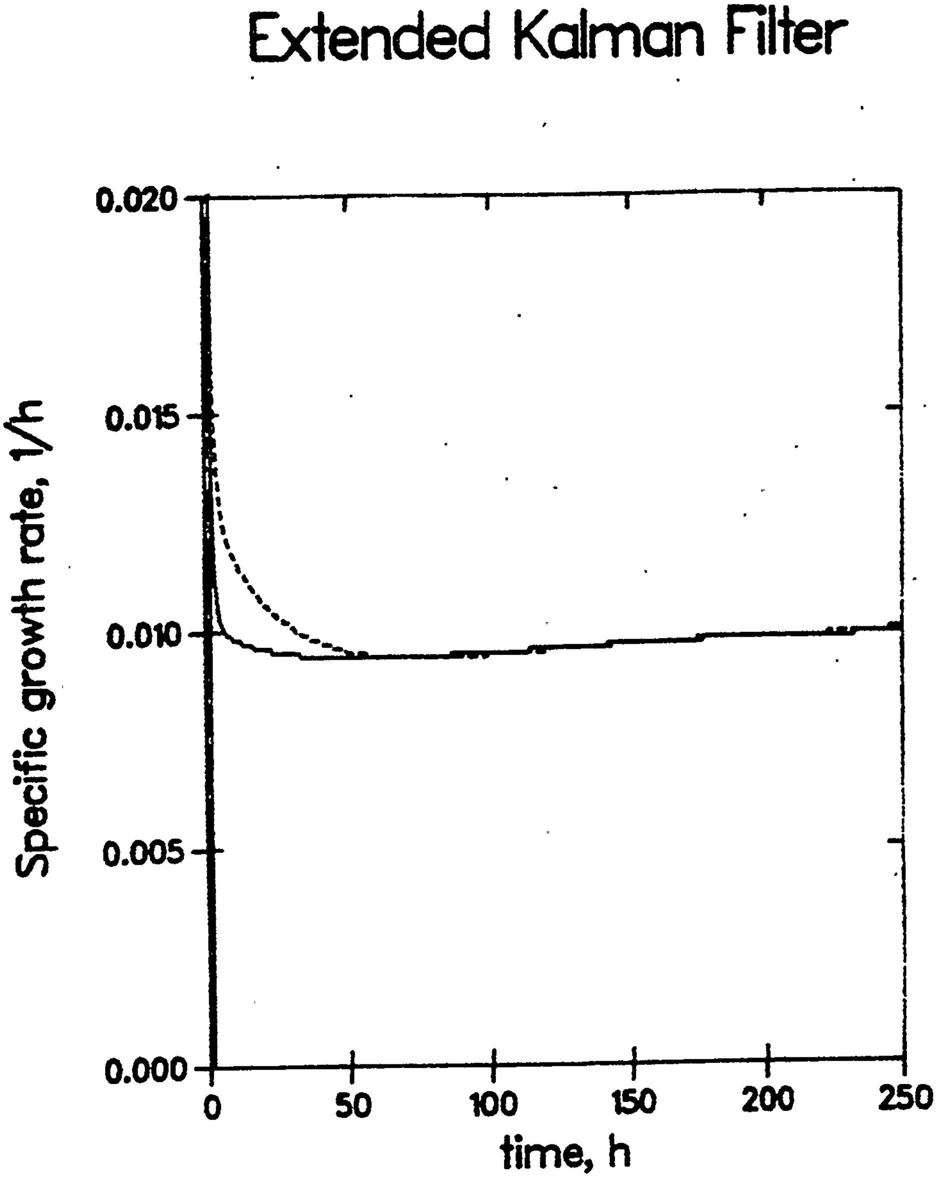

FIGURE 15 Simulated profiles of actual specific growth rate (---) and its Kalman Filter estimate during a controlled production run.

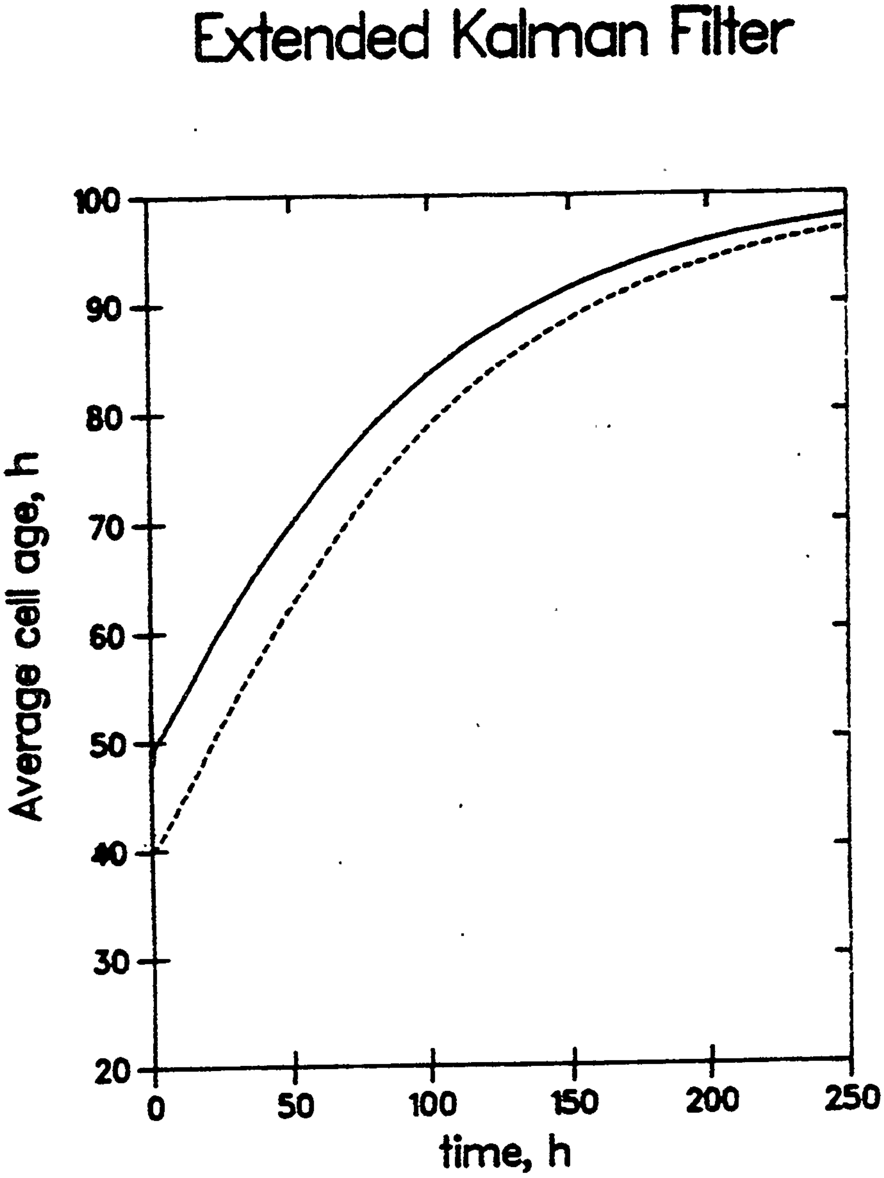

FIGURE 16 Simulated profiles of actual average cell age (---) and its Kalman Filter Estimate during a controlled production run.

7.0 EXPERIMENTAL APPARATUS

7.1 FERMENTATION HARDWARE

A schematic diagram of the whole experimental apparatus is shown in Fig. 17. In the following sections each major component of the apparatus is described in detail.

7.1.1 The fermentor

A 19 *L* BIOENGINEERING AG fermentor was used throughout this experimental study. The cylindrical 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. Its volume capacity is approximately 15 *L*.

The bottom of the reactor consists of a double-walled stainless steel bowl that contributes an additional 4 *L* to the reactor volume. Temperature control of the fermentation was facilitated by the circulation of warm or cold water through the walls of the bottom part of the reactor.

The impeller shaft enters the reactor through a center opening in the bottom jacket. Aseptic operation is ensured by a mechanical seal. Rotation speed up to 2000 *rpm* can be achieved with the help of an electric motor located underneath the reactor vessel. Various types of impellers can be easily fitted on the shaft to meet diverse mixing requirements.

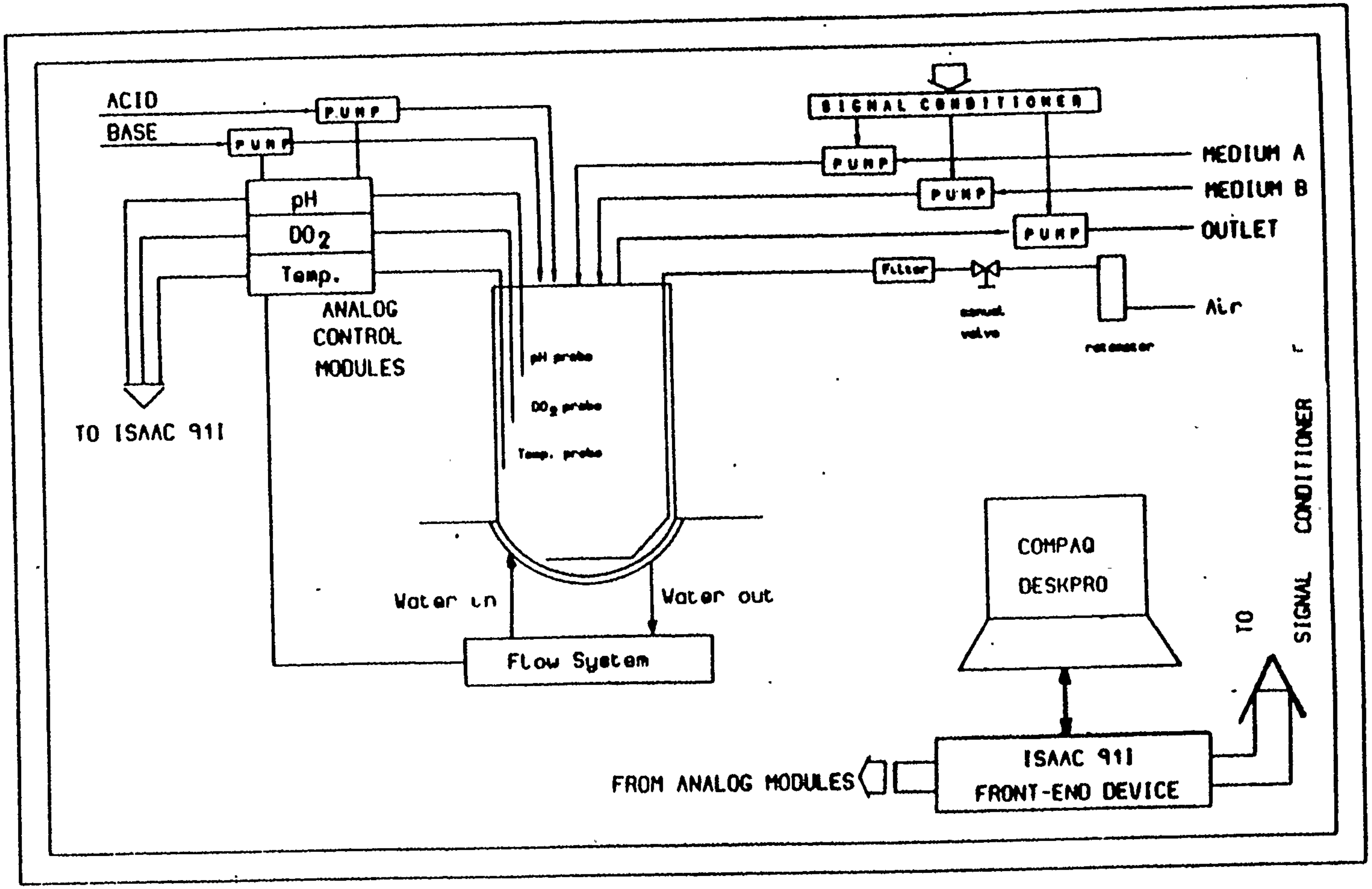

FIGURE 17 Set-up of the Experimental Apparatus.

The reactor top is closed by a 2 *cm* thick stainless steel head plate. Various probes, medium and air lines can be fitted on the head plate through 12 ports especially designed for air tight and aseptic operation. A safety valve set at 1.2 *bar* is fitted on the central port and protects the glass vessel from overpressurization during autoclaving. The port assignment during a typical open loop run is shown in Fig. 18.

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 ring shaped sparger at the reactor bottom. The air enters the fermentation in the form of bubbles, through minute holes in the sparger body. Spiral shaped spargers were used in some experiments for improved aeration efficiency.

The experimental set-up of the reactor is shown in Fig. 17. The analog and digital control hardware with the corresponding connections are shown in the same picture and will be explained in the next sections.

7.1.2 Hardware for Analog Measurement and Control

The fermentor was equipped with a number of analog modules that performed independent measurement and control of temperature, pH and DO concentration. A 4-20 *mA* recorder output is provided by each module. This output was used for monitoring and controlling the pro-

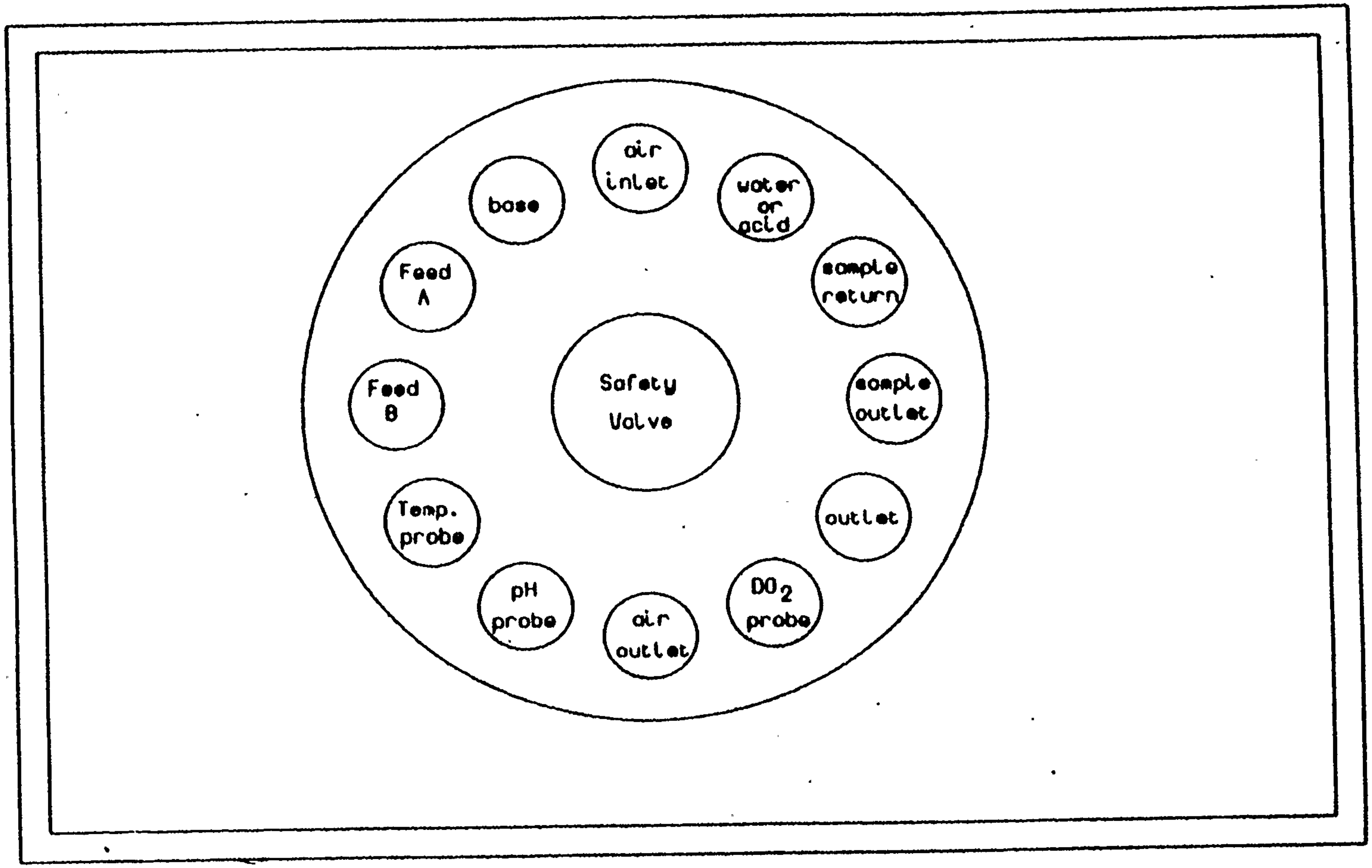

FIGURE 18 Head Plate Port assignment during a typical open loop run.

cess with a digital computer. Capability of the modules to accept a remote set point in the form of 4-20 *mA* signal makes them appropriate for use as local controllers in a Supervisory Computer Control Scheme.

7.1.2.a Temperature control

Temperature is measured with an 100 *Ohm* RTD. An electronic PID controller is used for controlling the temperature within ± $0.1^o C$. Cooling is performed by circulating cold tap water through the bottom jacket. When heating is required, the tap water is shut off and water recirculates through the jacket after passing through an on-off electric heater. The regulated temperature range is 5 degrees above tap water temperature up to $125^o C$ (sterilization conditions).

7.1.2:b PH control

PH was measured using an INGOLD, steam sterilizable combination probe. The output of the pH controller was used to drive two pumps for addition of acid (2 N HCl) or base (4 N $NaOH$) depending on the set point and the actual pH value. Stable pH control, within ± 0.02 units, was facilitated by the addition of a KH_2PO_4 buffer in the medium.

7.1.2.c Dissolved oxygen measurement

DO tension was measured with an INGOLD IL501 steam sterilizable polarographic oxygen electrode. The DO level in the reactor was controlled manually, by adjusting the air flow rate.

7.2 COMPUTER CONTROL HARDWARE

7.2.1 Computer and peripherals

A COMPAQ-Deskpro microcomputer was used, equipped with two 360 *Kbyte* floppy-disk drives and a 10 *Mbyte* hard disk. A RAM memory of 640 *Kbytes* was available for running the control software.

The peripherals included a COMPAQ-Deskpro keypad and a monochrome monitor. A printer was connected to the computer through a parallel interface card and a serial interface card was used for communication with other computers.

7.2.2 Computer - Process Interface

An ISAAC 91I front-end device was used to interface the process to the supervising computer. The unit included the following

- An A/D and a D/A converter
- 16 Analog Inputs
- 8 Analog Outputs
- 16 Digital Inputs
- 16 Digital Outputs
- Frequency Counter
- 4 Smith Triggers

Signal ranges from 0-10 *mV* up to 0-10 *V* could be accommodated by the analog inputs. The 4 - 20 *mA* outputs from the control modules were converted to 0-10 *V* signals by using one 500 *Ohm* resistor in parallel to the ISAAC analog input.

The analog outputs could provide a 0-10 *V* signal. A 4-20 *mA* output signal was taken through conversion of the voltage signal by a signal conditioner (Cyborg Corp.). The current signal was used to drive the feed and outlet peristaltic pumps.

The ISAAC 91I was supported by software written in machine language. The machine language subroutines can be called within a BASIC or FORTRAN program and are designed to run at the maximum speed allowable by the hardware.

8.0 CONTROL AND DATA ACQUISITION SOFTWARE

The software used for data acquisition and control was written in the programming language BASIC. Two control programs have been developed, one for open loop and one for closed loop runs. Complete listings of the programs are given in Appendix C.

The structure of the two control programs is very similar with the exception of the inclusion of the Extended Kalman Filter in the closed loop one. It is reminded that the filter calculations require integration of the model equations and solution of the probability distribution equations in real time.

Software design has accommodated the discrete form of the computer control algorithm by using the feature of "Time Interrupt" built in the BASIC language. "Key interrupts" are used to program special functions that can be "called up" by the operator. The use of interrupts implies an hierarchy in the program execution that is presented in the following paragraphs.

8.1 Program Structure

At any given time, the program is running on any of the three following levels

1. The Main (display) level
2. The Key Interrupt (Special Function) level and
3. The Time Interrupt(control) level.

Between sampling points the program is normally running on the Display level. The values of set points, manipulated variables and environmental parameters are displayed on the monitor, while the computer is idle waiting for an interrupt.

During this time the operator can use one of ten "Function Keys" to cause a "Key Interrupt" and initiate the execution of the corresponding special function. These functions facilitate the interactive communication between the computer and the operator. Set points and other important parameters can be changed on-line, measurements can be entered etc. Upon completion of the requested function, execution of the display subroutine resumes.

A Time Interrupt originates from the computer real time clock at every sampling point and has the highest execution priority. In the event of a Time interrupt, the currently running function is halted and the control algorithm is executed. When the control sequence is completed, execution of the halted function can resume.

The functions controlled by the computer on each of the execution levels are specified by the subroutines described in the next section.

8.2 Program Subroutines

8.2.1 Display subroutine

The main task of this subroutine is to display process information important to the operator. Temperature, dissolved oxygen tension and pH are being read on-line and displayed along with the values of variables that have been measured off-line and entered manually (for example glucose concentration, biomass concentration etc.). In addition, the values of the manipulated variables are displayed as well as the real time and time elapsed from the beginning of the experimental run.

In the closed loop program the integration of the model equations is taking place within this subroutine. In this case, the Function Keys are deactivated until the integration has been completed.

8.2.2 Special Function Subroutines

Each of these subroutines is executed by depressing the corresponding Function Key on the keypad. There are ten function keys labeled F1 to F10. In the initialization part of the program, one function and the corresponding subroutine is assigned to each of the keys as follows

F1	----->	display help screen
F2	----->	clear screen
F3	----->	change sampling interval
F4	----->	reserved key
F5	----->	plot the history of a process variable
F6	----->	rearrange the pumps
F7	----->	change various run parameters

F8 -----> linear regression

F9 -----> enter off-line measurements

F10 -----> change set points

It should be noted here that subroutine F7 is very important since it allows on-line change of parameters such as controller gains, feed concentrations, pump calibration constants etc.

8.2.3 Control Computations Subroutine

The most important program functions are performed within the control subroutine. Immediately after the Time Interrupt this subroutine instructs the front-end device (ISAAC 91) to read the analog inputs. The raw 12-bit number (integer ranging from 0 to 4095) is converted into a value in the appropriate engineering units. These values are checked for alarm-conditions and the corresponding alarm flags are set, if necessary. Finally the values are stored into the corresponding program variables.

Next task is to read the on-line measurements if the closed loop algorithm is used. In the first stages of the implementation of the control system the measurements are made off-line. In this case, an audible signal warns the operator that a limited time (typically 5 *min*) is available for the input of the off-line measurements. At that time the new data can be entered to be used for the calculation of the updated filter estimates. Otherwise, if time expires without any measurements entered, the program assumes that the Kalman Filter predictions are correct and continues. In the open loop program the above steps are omitted.

The new values of the manipulated variables are calculated, based on the control algorithm, and the physical constraints on the manipulated variables are checked. The resulting values are translated into flowrates and converted into 12-bit numbers. Through the interface a 4-20 *mA* current signal is downloaded to the pumps so that the calculated flowrates are delivered.

At this point the control part of the subroutine has been completed. However, several secondary but important functions have been included in this part of the program, taking advantage of the fact that this is the only part which is executed regularly and with the highest priority.

First the values of important program parameters are saved into a file to be used for program recovery after a possible power failure. This set of parameters includes the sampling interval, set points, feed concentrations, reactor volume and pump arrangement constants. Furthermore, the open loop program saves the expected biomass concentration while the closed loop one saves the most recent filter estimates. After a power failure these parameters are automatically read by a special program and execution of the control program resumes with the values stored last.

Next the values of manipulated variables as well as temperature, dissolved oxygen concentration and the latest off-line measurements are printed out and also written into a file to be used for plotting the history of these variables.

After the last task has been performed execution continues at the point where it was halted by the Time Interrupt.

9.0 EXPERIMENTAL PROCEDURES

9.1 Strain Preservation

The culture of *Penicillium chrysogenum* E15 (ATCC 26818) was obtained as freeze-dried ampoules from Eli Lilly Co. Ltd. A silica-gel stock was prepared from the original culture and was used for the entire experimental work. Portions of the silica gel were streaked onto 15 cm^2 slants of sporulation agar with composition given in Appendix A. The slants were incubated at 26 ^oC for 7-10 days until complete sporulation. Spores were removed from the slants by washing with water containing 50 mg/L Aerosol OT and were immediately used to incubate Fernbach flasks coated internally with sporulation medium. The total culture area in this stage was approximately 9000 cm^2 distributed in 9-10 flasks. The spores produced were directly used for the inoculation of the 19 L fermentor.

A spore suspension was obtained by washing the spores off the Fernbach flask cultures. Typical spore concentrations were of the order of $10^9\,spores/mL$. The suspension was typically sonicated for 30 *min* for breaking up the spore clamps.

9.2 Pretreatment of Celite Beads

Celite R630 (30 x 50 *mesh*) from Johns Manville Co. Ltd., was used as the immobilization matrix. The average diameter of the beads was 450 μm. The pretreatment of the beads included a series of washings with distilled water to remove fines and water soluble contaminants. Drying in an 100° *C* oven followed and then the beads were transferred into a muffle furnace. There the temperature was gradually risen to 500° *C* within 2 hours and maintained at this level for, typically, 8 hours to burn off any organic contaminants. After cooling to room temperature Celite was washed again and dried. The required quantity was weighed and placed in a Pyrex glass carbuoy for sterilization at 121° *C* for 1 hour.

9.3 Reactor Sterilization

Before the reactor sterilization the pH probe was calibrated in buffer solutions and the oxygen probe zeroed in helium gas. Oxygen, pH and temperature probes were inserted into the reactor through the head plate and secured in place. The reactor was filled with approximately 10 *L* of distilled water and sterilized at 121° *C* for 40 *min*. The feed lines, outlet, base and acid lines were then attached aseptically and the reactor was resterilized.

After the second sterilization the air flow was started in order to avoid negative pressures during cooling. The temperature set point was set at 25° *C*. When the reading on the oxygen meter stabilized, the 100% saturation point was set. Growth medium (with composition given in Appendix A) was added up to the concentration desired for the spore germination. The presence of buffer in the medium allowed a much tighter pH control within ± 0.02 units when the pH controller was used with a set point of 6.8.

As a precaution against contamination, the flow was initiated in all lines for a short time (typically 5 *min*) and then the system monitored for signs of contamination during the next 48-72 hours.

9.4 Inoculation

This step includes the immobilization of spores in the Celite beads and the subsequent transfer of the beads into the reactor. The large volume (typically 4 *L*) and the weight (4 to 5 *kg*) of the wet, spore-loaded beads necessitated the establishment of a strict protocol designed to minimize the risk of contamination.

The removal of the spores from the surface culture was always performed in a laminar flow hood. Inside the hood the spore suspension was transferred into a 5 *L* conical flask where it was diluted with sterilized water to a final volume determined by the quantity of Celite used. A sample of the final suspension was removed for spore counting and the flask was closed with a cotton plug.

An outlet port fitted at the bottom of the flask side was used for the transfer of the solution to the Celite container that was located outside the hood. Silicone tubing was used as the connecting line. During the addition of the suspension the Celite container was shaken manually for even distribution of the spores. Immediately after the end of the addition, the container with the attached flask were moved to a rotary shaker where the spores were contacted with the Celite for 1 hour at 100 *rpm*.

Subsequently, the Celite beads in the container are washed with three to four volumes of distilled water. The wash water is collected, the volume is measured and a sample is taken to determine the spore uptake by the beads (by difference). Finally the washed Celite beads are transferred to the fermentor under aseptic conditions.

10.0 EXPERIMENTAL RESULTS AND DISCUSSION

In this work the experiments have been conducted in the "open loop" mode only. The experiments have been designed around the primary objective of maximizing the penicillin concentration in the effluent stream, based on the mathematical model and the principles used in the controller design. Furthermore, the objective of the open loop runs was to estimate various kinetic parameters which are needed before proceeding to the "closed loop" runs.

A parallel objective was the development of proper experimental techniques for the trouble-free operation of the 19 L fermentor under computer control. At the same time the software had to be tested in its real operating environment and adapted to the particular fermentation needs.

10.1 Design of Experiments

10.1.1 Strategy for Maximization of Penicillin Concentration

If we refer back to the steady state equation for penicillin production, we can see that

$$p = \frac{q_p x}{D} \tag{66}$$

This relationship implies that for maximum penicillin concentration, q_p and x should take maximum values. At the same time the dilution rate, D, should be as small as possible.

The maximum value of q_p corresponds to an optimum cell age of approximately 70 h or, in more practical terms, to an optimum growth rate of 0.0145 h^{-1}. However, maintaining an optimum specific growth rate is only a necessary condition. It is equally important that no oxygen or other nutrient limitations appear that would adversely effect the value of q_p.

Minimization of the dilution rate, D, is not quite as straightforward, because the nutritional requirements of the culture have to be satisfied at all times. The demand on glucose, at any given time, is determined mainly by the growth rate and the biomass concentration and is equal to the product σx. For a constant growth rate the steady state equation (28) dictates:

$$D\,(s_f - s) = \sigma x \qquad (67)$$

Therefore, a low dilution rate implies the use of a very high feed concentration of glucose. It is obvious that the minimum for the dilution rate is imposed by the feed concentration of the glucose, according to the previous equation.

The requirement for maximum biomass concentration, x, is the most challenging to meet. The high oxygen demand posed by a high biomass concentration has to be satisfied by increasing the oxygen transfer from the air bubbles to the liquid broth. The confinement of cells on beads significantly reduces the mass transfer resistance but there is always a percentage of free cells in the broth that moderate this effect. The number of free cells increases significantly when the shear rate in the reactor causes the immobilized cells to break and enter the liquid phase. Therefore, maximization of biomass depends heavily on the configuration and the design of the fermentation vessel, as well as the design of the air distributor.

As a final note on this subject, it is mentioned that penicillin production can be strongly inhibited by toxicity effects. Most important is the contribution of the precursor PAA. This occurs when its concentration in the reactor exceeds a certain limit (typically 1 g/L). This factor has to be considered along with the others mentioned above, if a high final penicillin titer is desired.

10.1.2 Implementation

Designed runs simulated an open loop experiment with a high desired penicillin concentration. From Eq. (34) we can see that the penicillin concentration that can be achieved depends on the glucose feed concentration according to the relationship

$$p = (s_f - s_d)\ \frac{q_p}{\sigma} \tag{68}$$

The maximum glucose concentration was 200 g/L, limited by the solubilities of the salts included in the medium. According to the above equation the steady state penicillin concentration would be expected to be 4.73 g/L for a q_p value of 0.001 h^{-1} and a σ value of 0.038 h^{-1} at the optimum growth rate of 0.0145 h^{-1}. However, with such a small value for qp the response of the penicillin concentration is expected to be very low and , in fact, the results of a simulation run show that at a biomass concentration of 50.0 g/L the penicillin concentration has only reached 2.7 g/L, or 56% of the QSS concentration after approximately 150 hours.

The dilution rate was calculated by using Eq. (28) expressed as

$$D = \frac{\sigma_c x}{s_f - s_d} \tag{69}$$

The glucose concentration in the feed was kept constant at its maximum value, therefore, D takes its minimum value for the given growth rate.

The steady state controller for the PAA was also used, as given by Eq. (39). To facilitate manipulation of the PAA feed concentration two flasks of medium had to be used, flask A without any PAA and flask B with a PAA concentration higher than the maximum expected. The two media were simultaneously added with a total flow F dictated by the value of the dilution rate and the working volume

$$F = D V \tag{70}$$

The proportion of each of the flows was determined by the output of the PAA controller. Flow of medium B was given by

$$F_B = F \frac{z_f}{z_{\max}} \tag{71}$$

where $z_{\max}$ was the PAA concentration in medium B.

10.2 Description of Performed Experiments

Two preliminary experimental runs helped in resolving numerous technical difficulties associated with the procedures of spore immobilization and aseptic inoculation of the reactor. The experience from these runs was used in refining the experimental procedures that are presented in Chapter 9 and the analytical methods given in Appendix B. The gradual improvements brought us to a point where the risk of contamination had been minimized and a first successful run was made possible.

10.2.1 Experimental Run 3.

For this run it was decided that the biomass concentration in the beginning of the production phase should have a rather high value, in the vicinity of 10 g/L. At this biomass level the culture was expected to show an increased resistance against contaminating microorganisms like bacteria.

The measured immobilized biomass at the end of the growth phase reached 14 g/L and indeed the culture showed no signs of contamination throughout the run. However, a series of technical difficulties caused large flowrate and volume fluctuations and prevented the culture from reaching a Quasi-Steady-State.

Nevertheless, the biomass concentration measurements were expected to be reliable, since they were estimated from the total biomass on the basis of a constant volume. The experimental results in Fig. 19, however, showed no definite growth trend and they were highly scattered. At that time, the phenomenon was attributed to unrefined measurement techniques, since this was one of the first runs that such measurements were made.

With respect to the penicillin production, it would be useful to mention here that the concentration of penicillin (expressed as Na^+ salt) in this run reached 0.6 g/L under adverse fermentation conditions which included great variations of glucose concentration and a rather high peak PAA concentration (above 1.0 g/L).

10.2.2 Experimental Run 4.

In this run, the initial set point for the growth rate was set at 0.005 h^{-1}. It was planned to approach the optimum growth rate of 0.0145 h^{-1} in a second step, for a smoother transition. The starting biomass concentration in the production phase was set at 4.2 g/L based on a measurement at the end of the growth phase.

The response of the growth rate in the first 10 hours of the production phase was reflected in the dissolved oxygen profile during this time. As shown in Fig. 20, the DO concentration dropped very fast to a value of 52% within the 6 first hours. A period of slowly decreasing DO values followed, indicating that the growth rate in the culture was being controlled at a low level. At time t= 10 h the set point of the growth rate was changed to 0.01 h^{-1} and the air flow rate was increased from 20 to 24 l/min. After these changes were made, the rate of drop of the dissolved oxygen concentration

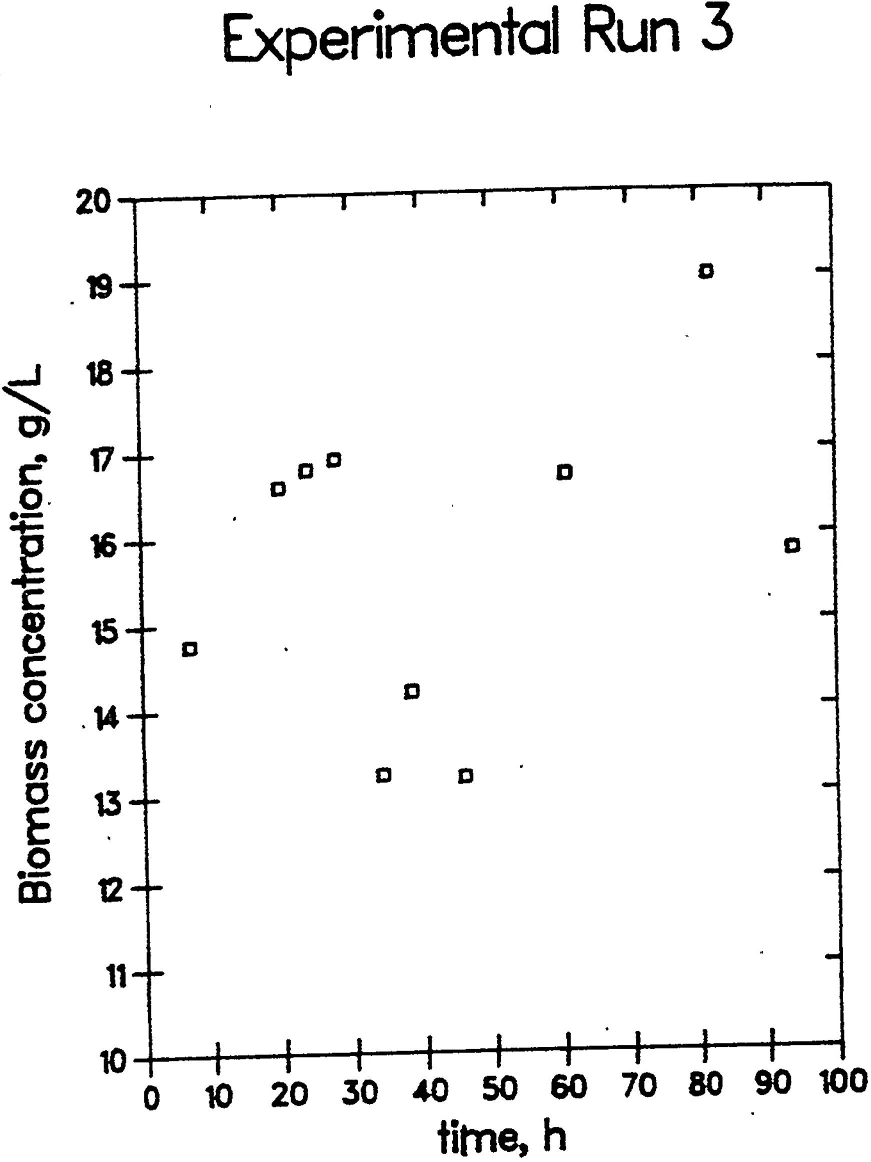

FIGURE 19 Biomass concentration versus Time in Run #3.

increased, an indication of faster growth.

At time t = 45 *h* the most recent biomass measurement was entered, in order to correct the value of the dilution rate. This change resulted in a very sharp increase of glucose concentration, shown in Fig. 21, and a sharp decrease of the DO concentration, from 35% to 15% within 3 hours. From the 55 hour point until the end of the run the dissolved oxygen concentration in the reactor was below 10% despite the fact that the maximum possible air flowrate was used.

It can be argued that the sharp increase and the subsequent low values of DO can be a result of a sudden increase in the number of free cells in the medium. Indeed, microscopic examination of the broth has showed that the number of free cells increases very fast at immobilized biomass concentrations over 16-18 *g/L* (with a 10% *v/v* Celite concentration in the fermentor). However, Wittler et al. (1983) have supported that the problem might be inherent in the *P. chrysogenum* pellet suspensions. Their experimental studies showed that the viscosity increases exponentially with biomass concentration even in the absence of free cells. As a result the oxygen transfer is expected to deteriorate very fast too.

It was attempted to overcome the problem of reduced oxygen transfer in the later stages of the fermentation by improving the design of the air distributor. The objective was to keep the size of the bubbles as small as possible and to achieve a more even dispersion of the bubbles. Although there was a definite improvement from this

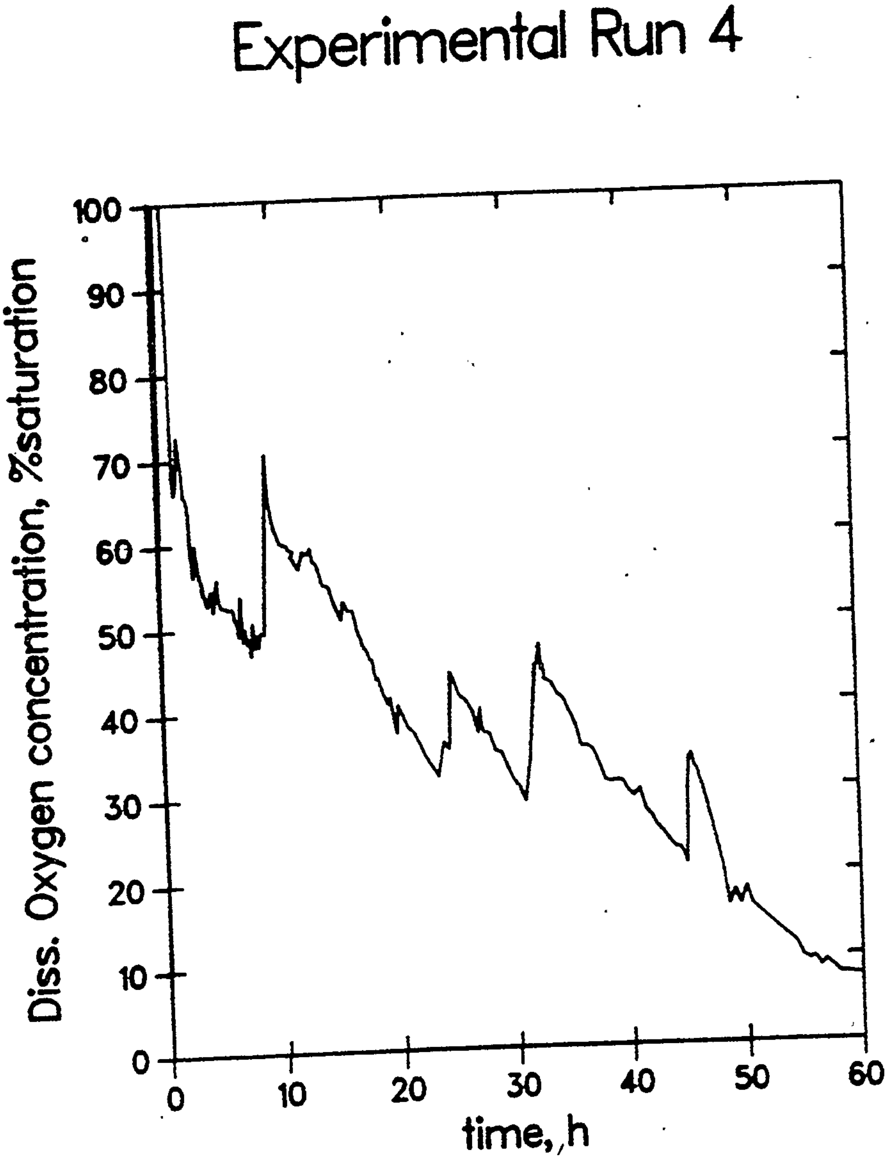

FIGURE 20 DO concentration versus Time in Run #4.

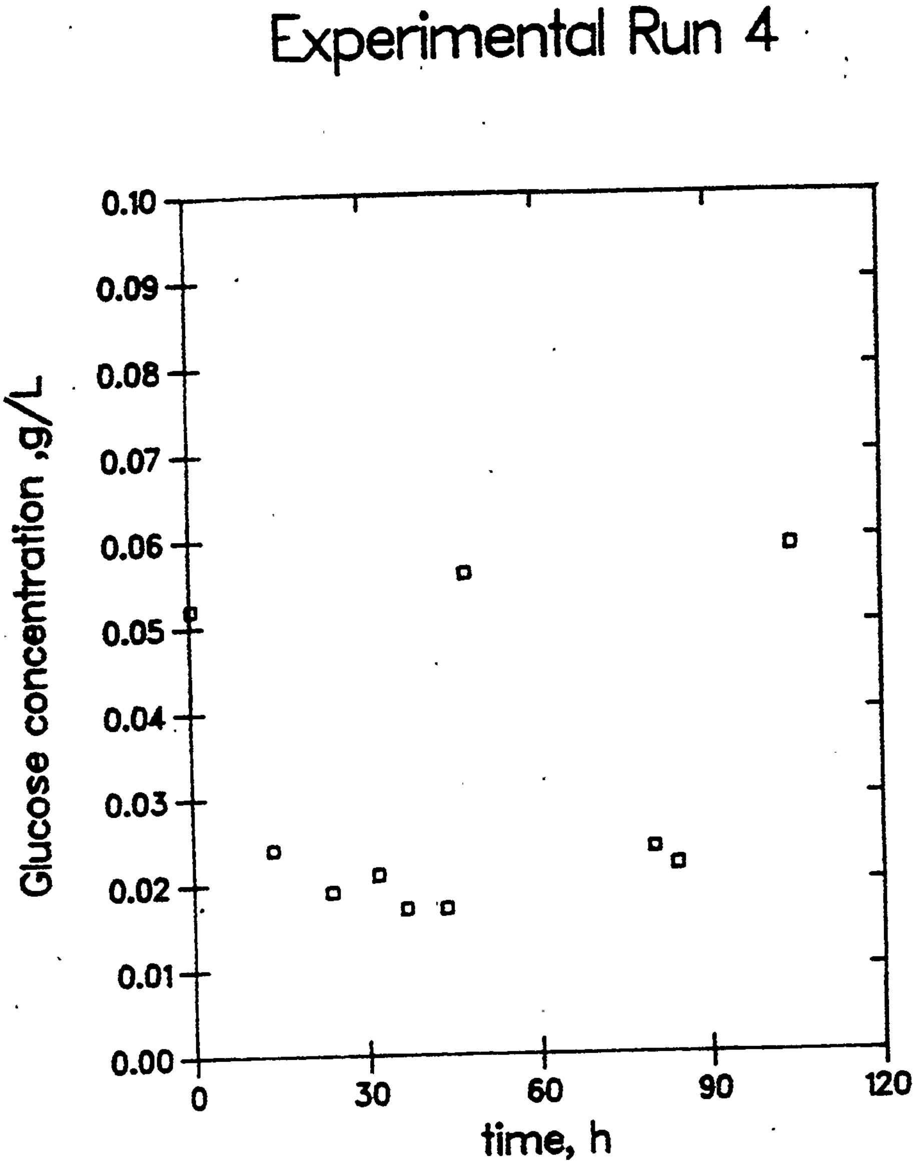

FIGURE 21 Glucose concentration versus Time in Run #4.

point of view, the effects on the oxygen transfer were minimized by the persistent viscosity problem.

From 20 to 45 hours the points on the logarithmic biomass plot fell close to a straight line, indicating a constant specific growth rate. Later in the run, the measurements showed a high scatter and the immobilized biomass concentration seemed to level off at the value of 18 *g/l*. After the 100 hour point the measurements showed a notable decrease in the immobilized biomass concentration.

The constant growth rate in the 20 to 45 hour interval is also indicated by the constant value of the glucose concentration in Fig. 21. The penicillin concentration jumped from 0.018 *g/L* to 5 *g/L* when the new biomass value was used in the program.

The disturbance introduced at 45 hours strongly affected the penicillin concentration profile as well, as shown in Fig. 24. Starting at 10 hours, penicillin increased at an almost exponential rate until the 47 hour point, when it reached a concentration of 0.65 *g/L*. At this point the productivity dropped very fast and penicillin concentration sank to the 0.4 *g/L* level, without recovering for the remainder of the 110 hour run. The PAA concentration was controlled reasonably well at 0.7 *g/L* for the most part of the fermentation (Fig. 25).

10.2.3 Experimental Run 5.

As our confidence in the used aseptic techniques increased, an ini-

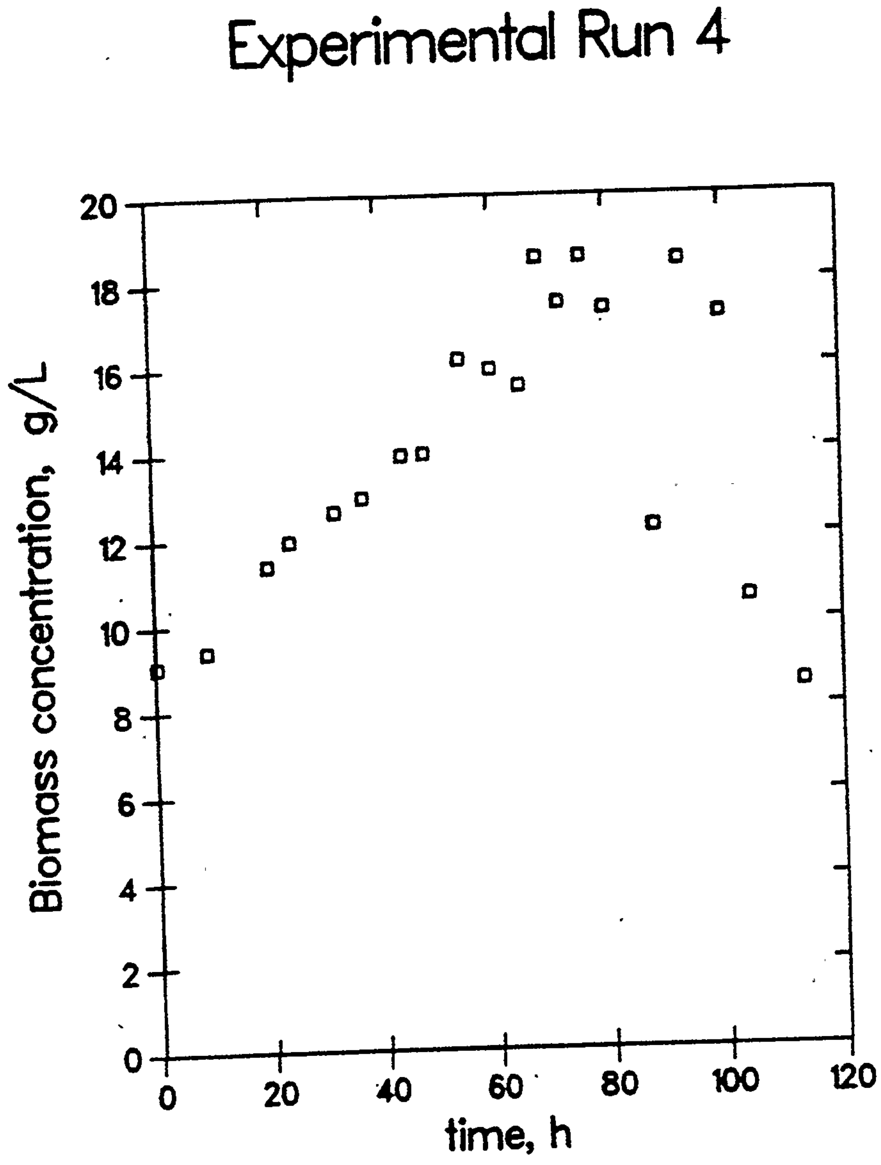

FIGURE 22 Biomass concentration versus Time in Run #4.

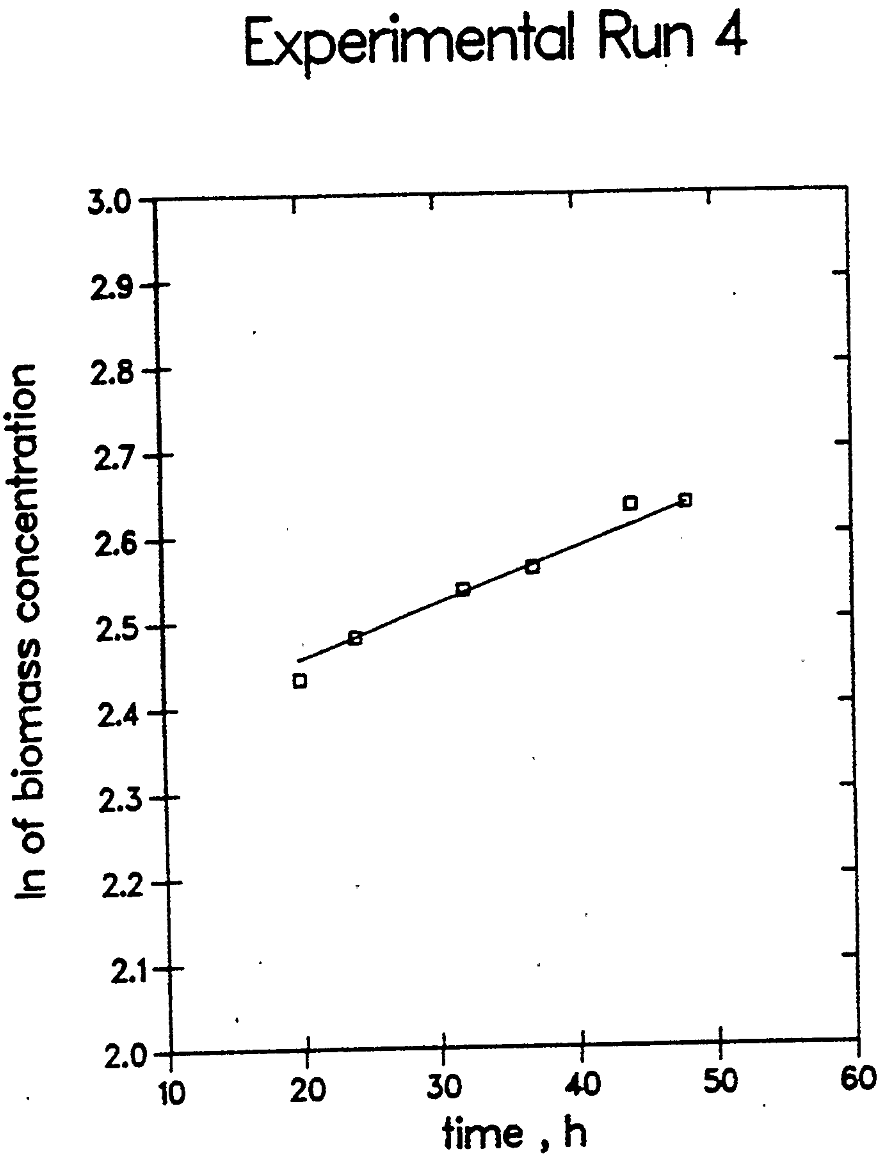

FIGURE 23 ln of Biomass Concentration versus Time in Run #4.

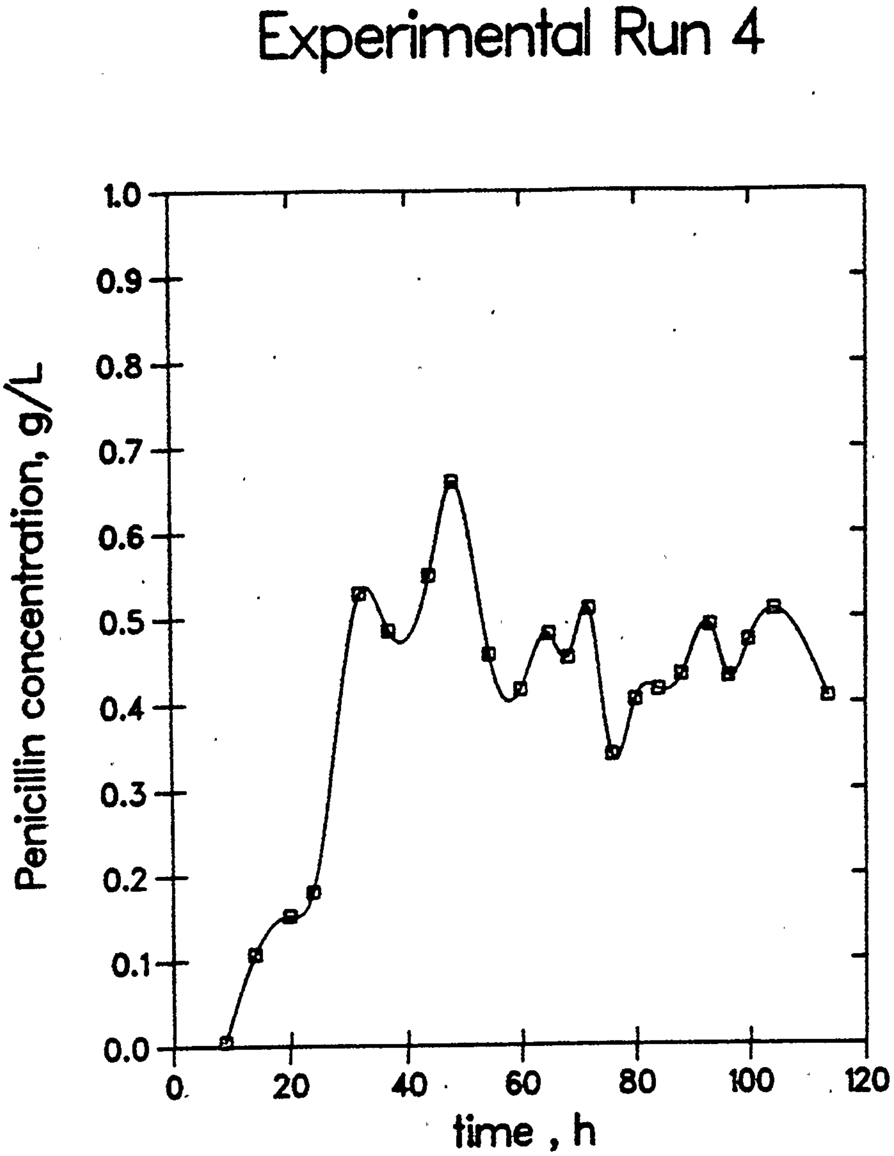

FIGURE 24 Penicillin concentration versus Time in Run #4.

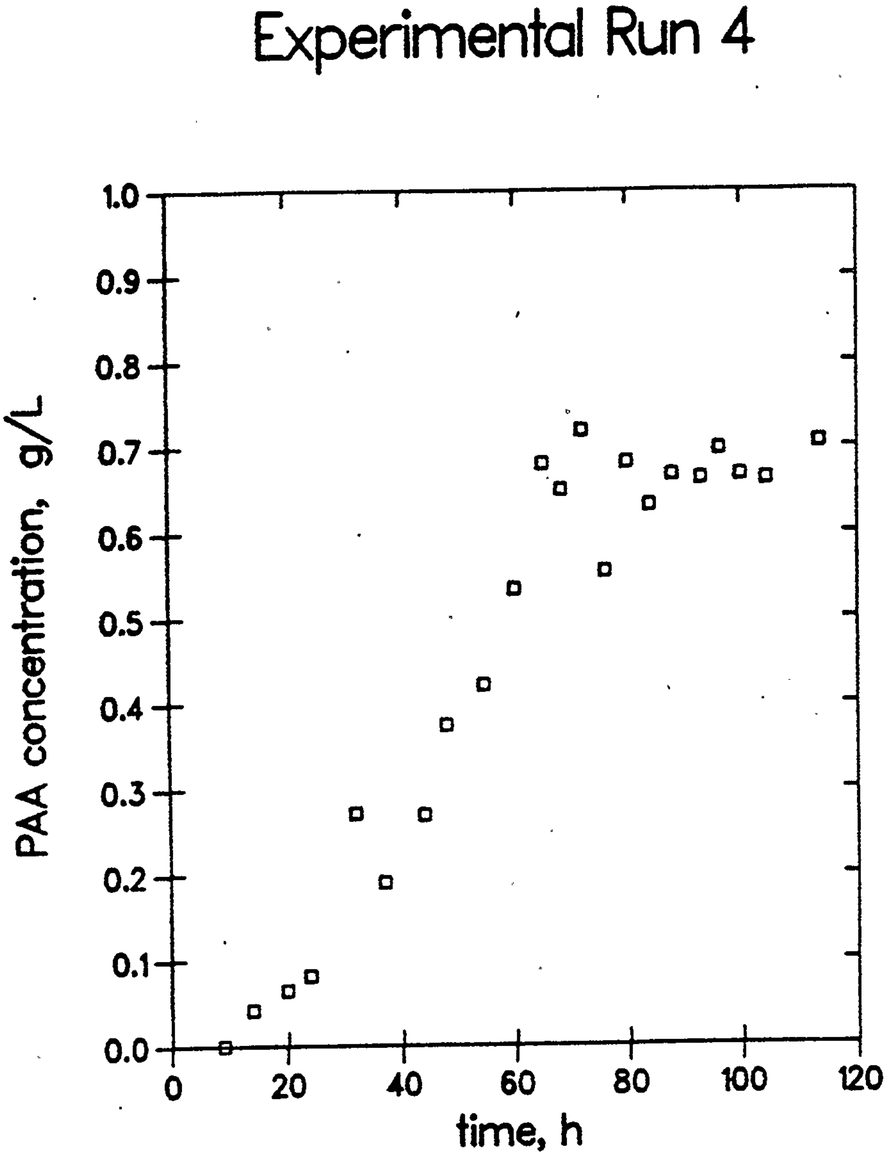

FIGURE 25 PAA concentration versus Time in Run #4.

tial biomass concentration of only 3.3 g/L was used for the production phase of Run 5. The set point for the growth rate was 0.005 h^{-1}.

The biomass profile during this run is presented in Figures 26 and 27. It can be seen that a steady state growth rate was achieved within the 10 first hours and it was maintained throughout the run. The profile of the glucose concentration is shown in Fig. 28 where a steady state concentration of 0.008 g/L can be observed. Small fluctuations around the steady state value during the first part of the fermentation can very well be due to experimental error, since the measured concentration was close to the detection limit of the employed method.

Later, at time t = 40 hours the growth rate set point was changed to 0.008 h^{-1}, however, no immediate response of the glucose concentration could be observed. This may be due to the fact that the dilution rate calculated by the program corresponded to a biomass of only 4.9 g/L while the actual biomass was close to 7 g/L. This discrepancy is mostly a result of the deviation of the actual growth rate from the desired set point, which resulted from the use of biased values for the kinetic parameters in the model.

A peak in the glucose concentration can be observed at 70 hours from the beginning of the production phase. The origin of this disturbance may be a delayed reaction to the set point change at 40 hours. At 72 hours the set point was brought back to 0.005 h^{-1} and in this

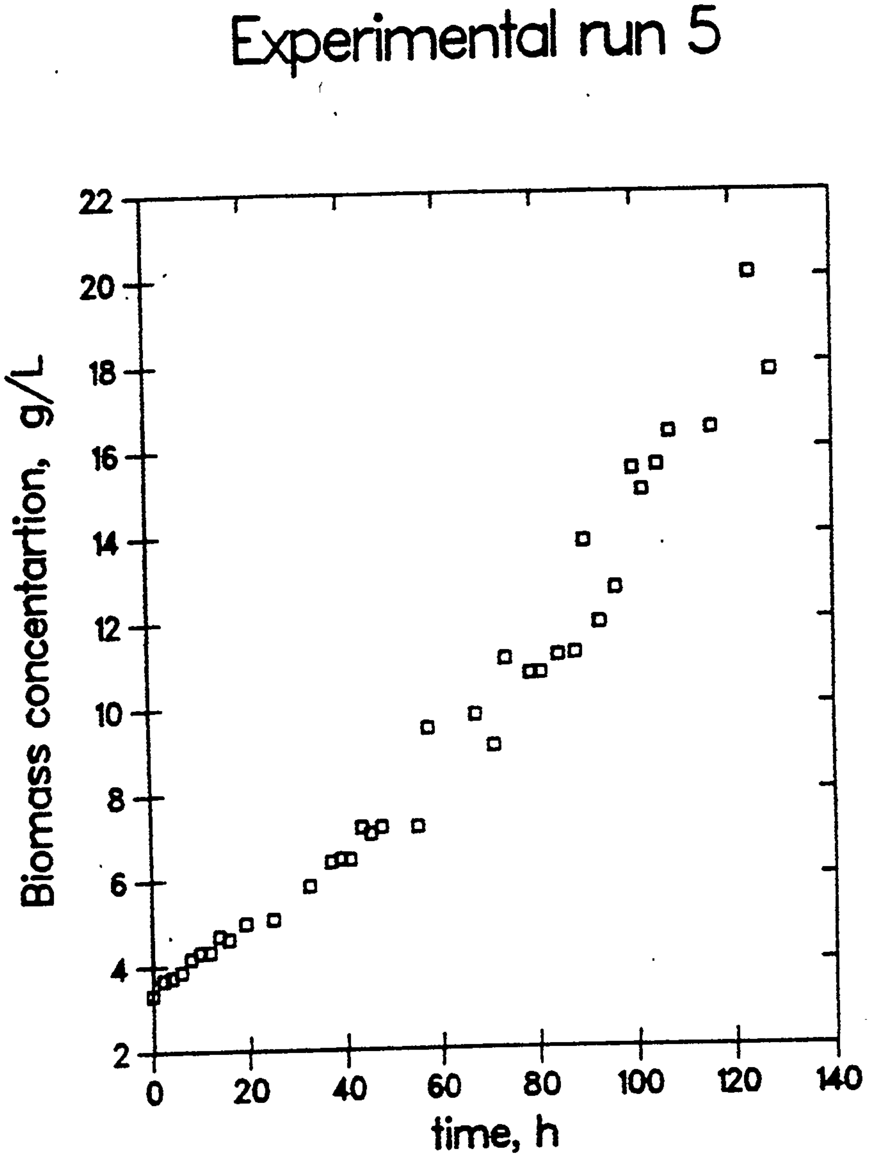

FIGURE 26 Biomass concentration versus Time in Run #5.

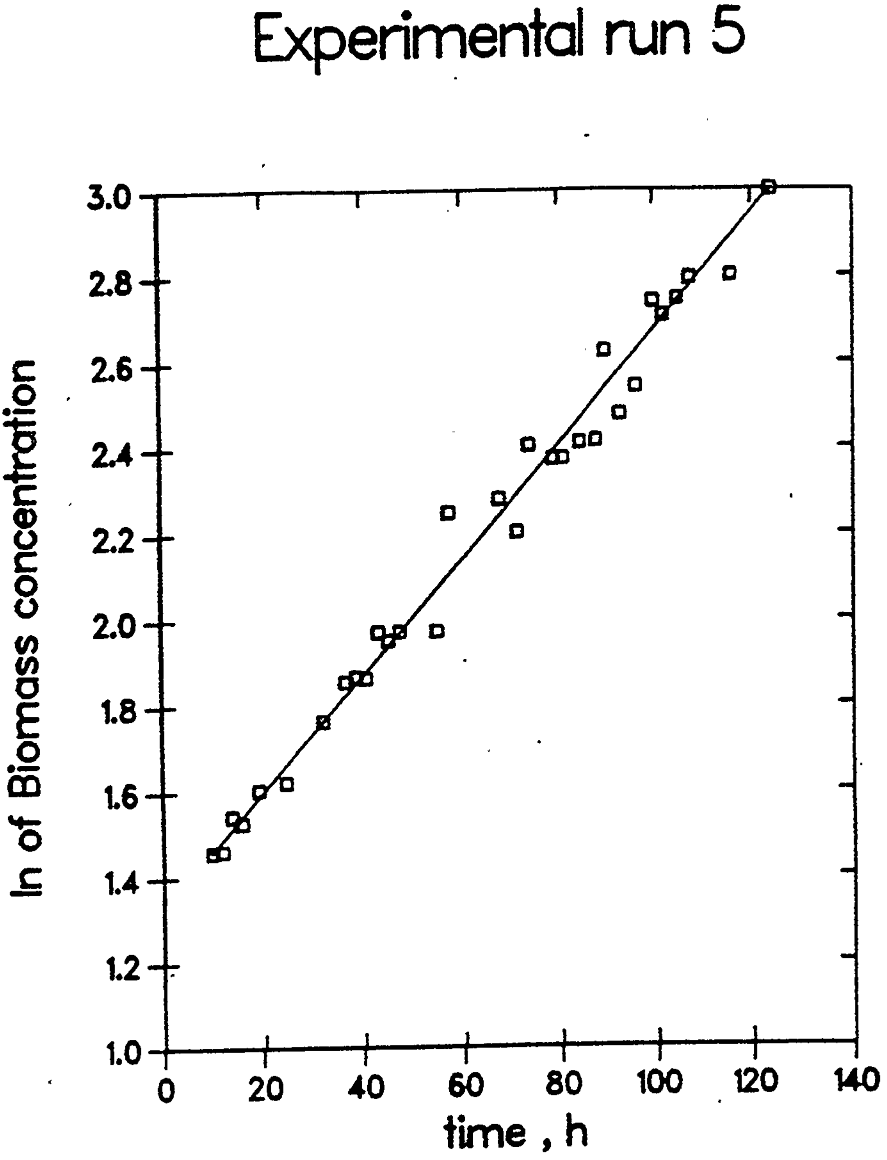

FIGURE 27 ln of biomass Concentration versus Time in Run #5.

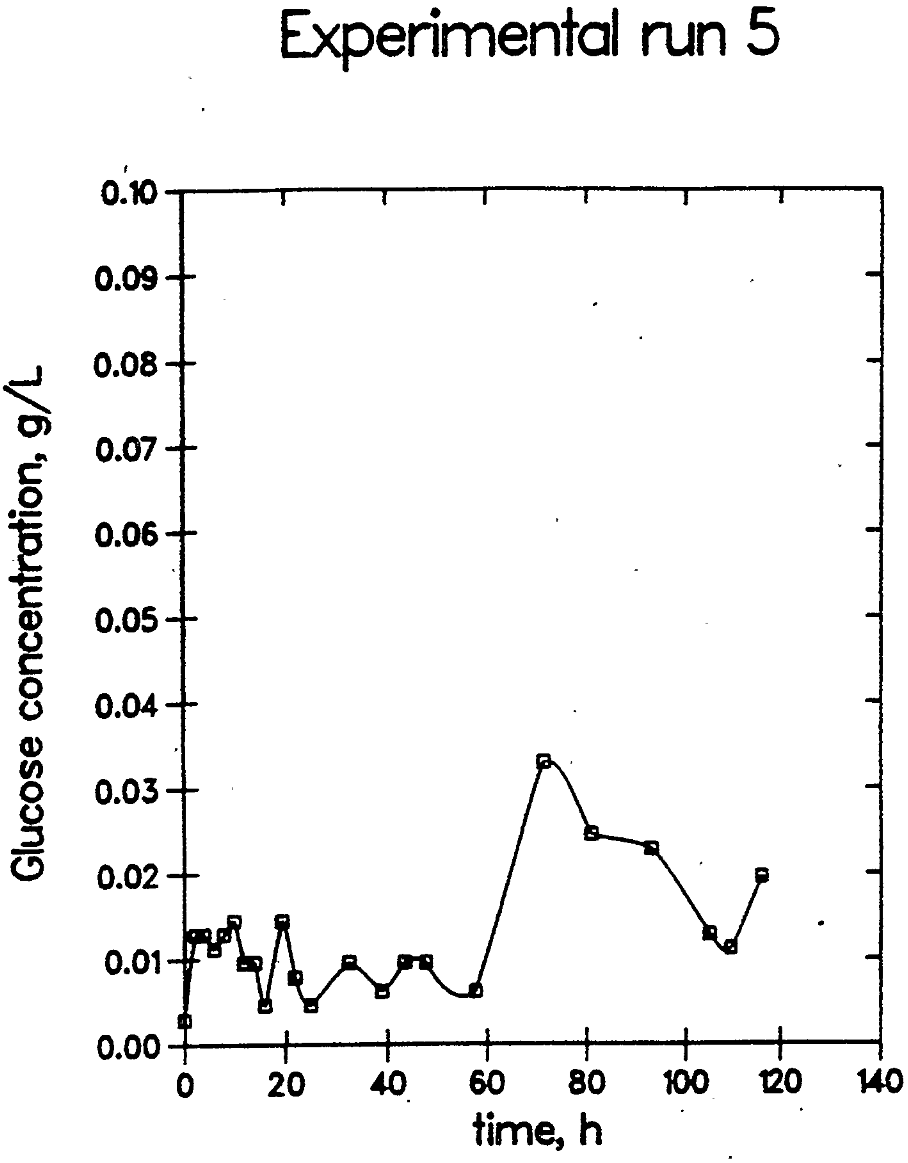

FIGURE 28   Glucose concentration versus Time in Run#5.

case we can see a faster response of the residual glucose. Since a higher biomass existed than accounted for, a drop in glucose being fed had a greater impact than an increase.

In Fig. 29 the penicillin concentration is seen to increase in an exponential rate and reach a final concentration of 1.95 g/L. The PAA concentration increased up to 0.4 g/L within the first 75 hours, overshooting the set point of 0.3 g/L (Fig. 30). The overshoot can be explained by the high value (0.1) used for the feedback gain, in order to achieve a short rise time. Nevertheless, the PAA concentration was kept below 0.6 g/L throughout the run.

10.2.4 Experimental Run 12.

In the final run included in this work, the initial biomass for the production phase was 5 g/L and the set point for the growth rate was 0.005 h^{-1}. After a short transition period a steady state was quickly achieved as shown in Fig. 31.

It should be noted that in this experiment the total biomass concentration was measured, including immobilized as well as free cells. The scattering of the data was considerably reduced compared to previous runs (Fig. 31). Furthermore, the duplicate biomass determinations agreed very closely, a fact implying that the remaining scattering is mostly due to sampling error rather than measuring error.

The glucose concentration had an approximately constant value of

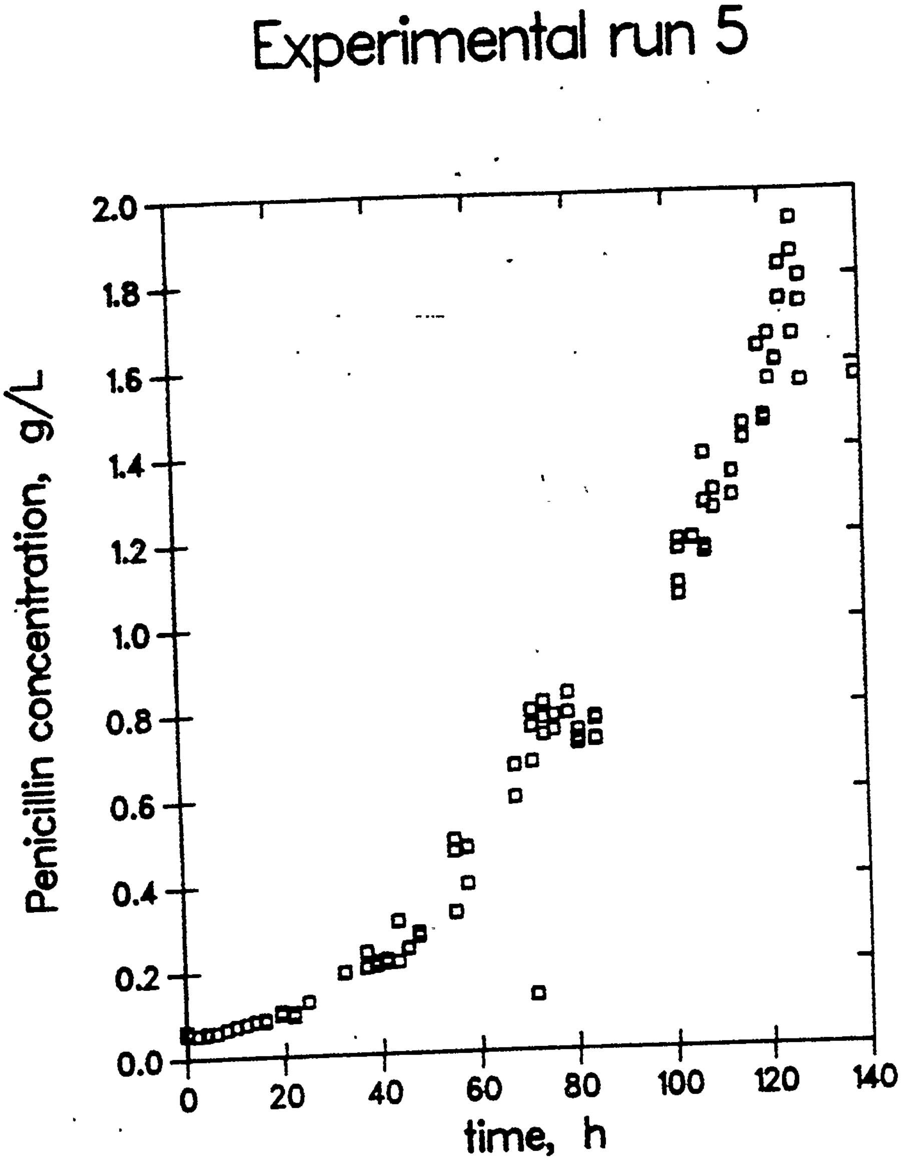

FIGURE 29 Penicillin concentration versus Time Run #5.

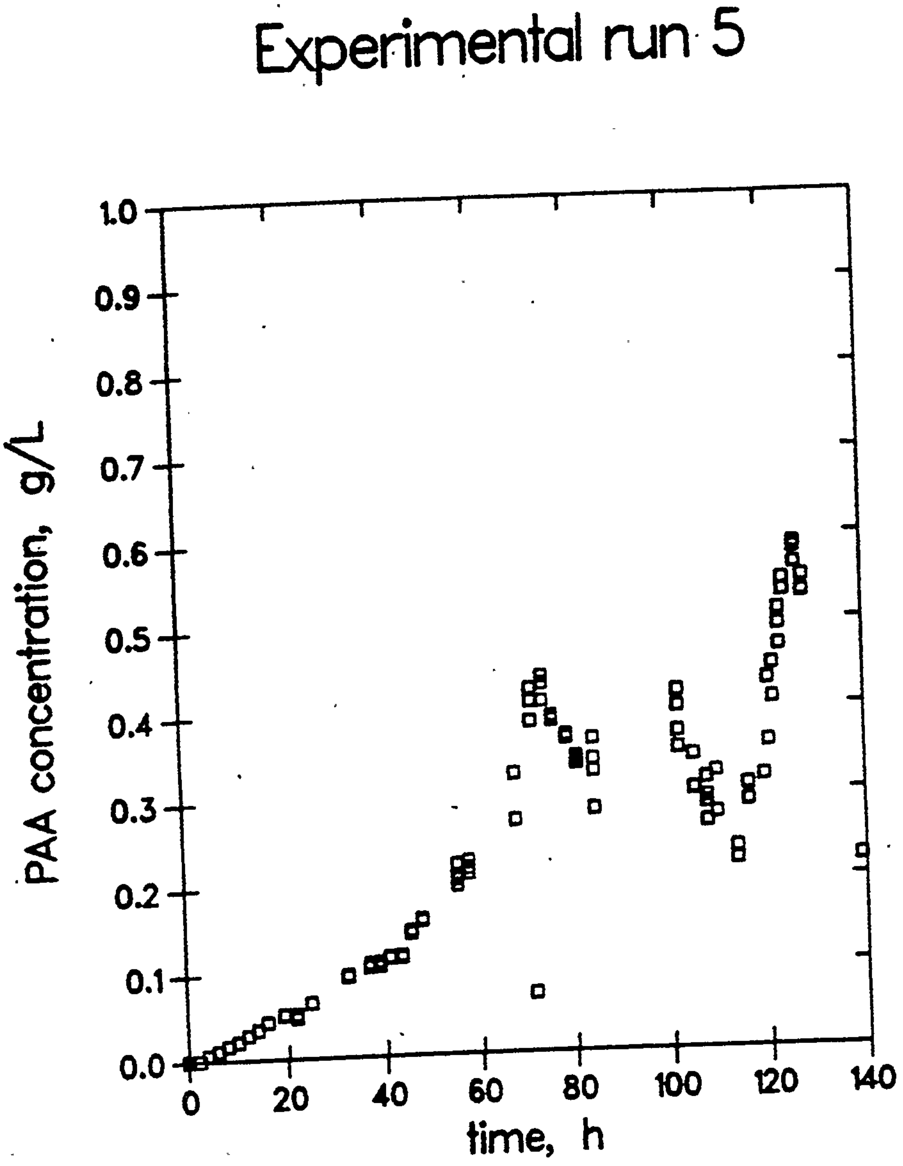

FIGURE 30 PAA concentration versus Time in Run #5.

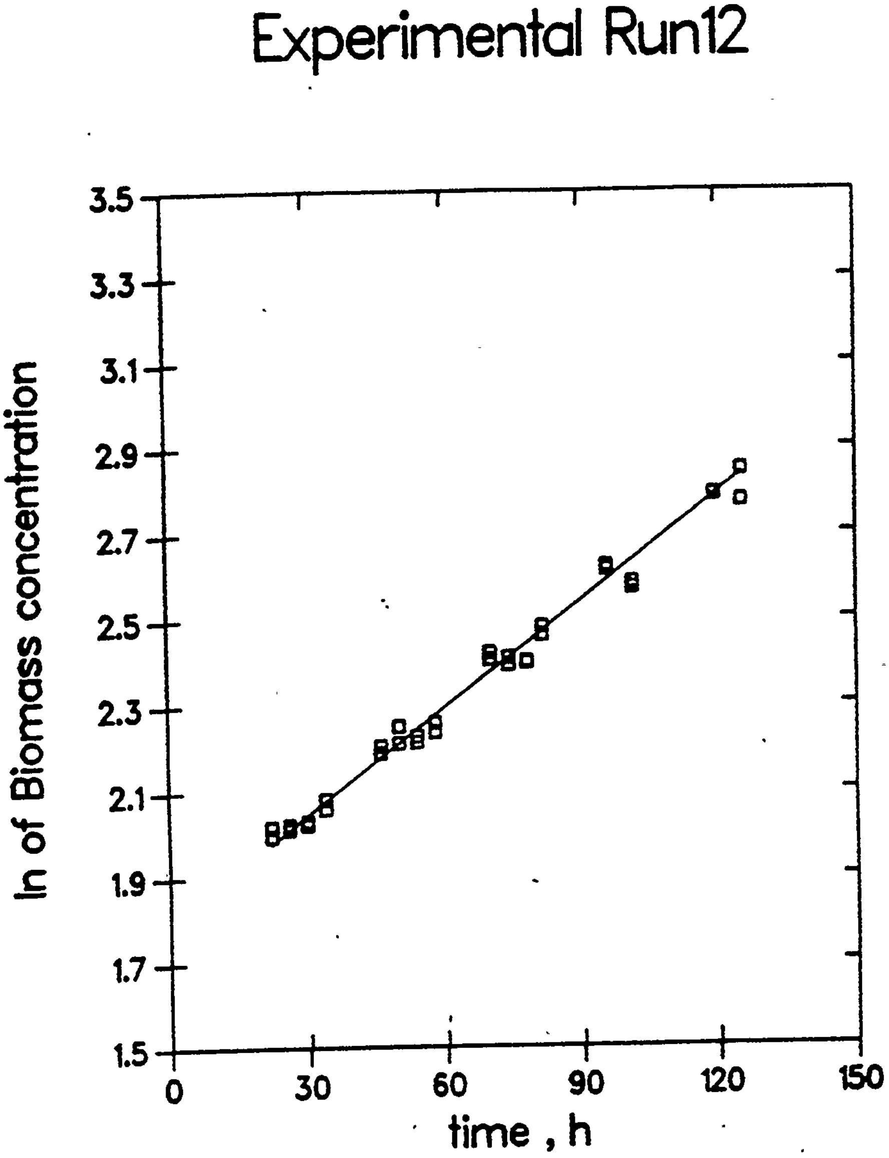

FIGURE 31   ln of Biomass concentration versus Time in Run #12.

0.022 g/L during the QSS (Fig. 32). The penicillin concentration was increasing steadily during the experiment to reach a final value of 0.9 g/L, as shown in Fig. 33. The PAA concentration did not exceed 0.2 g/L throughout the run neither it became limiting (Fig. 34).

10.3 Estimation of Important Model Parameters

The fact that a constant growth rate could be achieved with an exponentially increasing dilution rate confirms the validity of the glucose equation used in the model. Least-squares linear regression was used to estimate the steady state growth rate for each of the runs 4, 5 and 12.

The actual growth rates were in general much higher than expected and this fact has given us the incentive to calculate the maintenance energy for the immobilized cells. A reduced maintenance energy would explain the high growth rates and would come in agreement with earlier findings in tower loop reactors where the maintenance energy for lactose was half the value as in a stirred tank reactor (Konig et al., 1982).

The calculation of maintenance was based on a mass balance for glucose and biomass at a steady state, where the glucose consumed was equal to the glucose fed. The outlet term was neglected as insignificant, compared to the glucose fed - because s_f >> s. From the mass balance the value of σ was calculated and then the expression

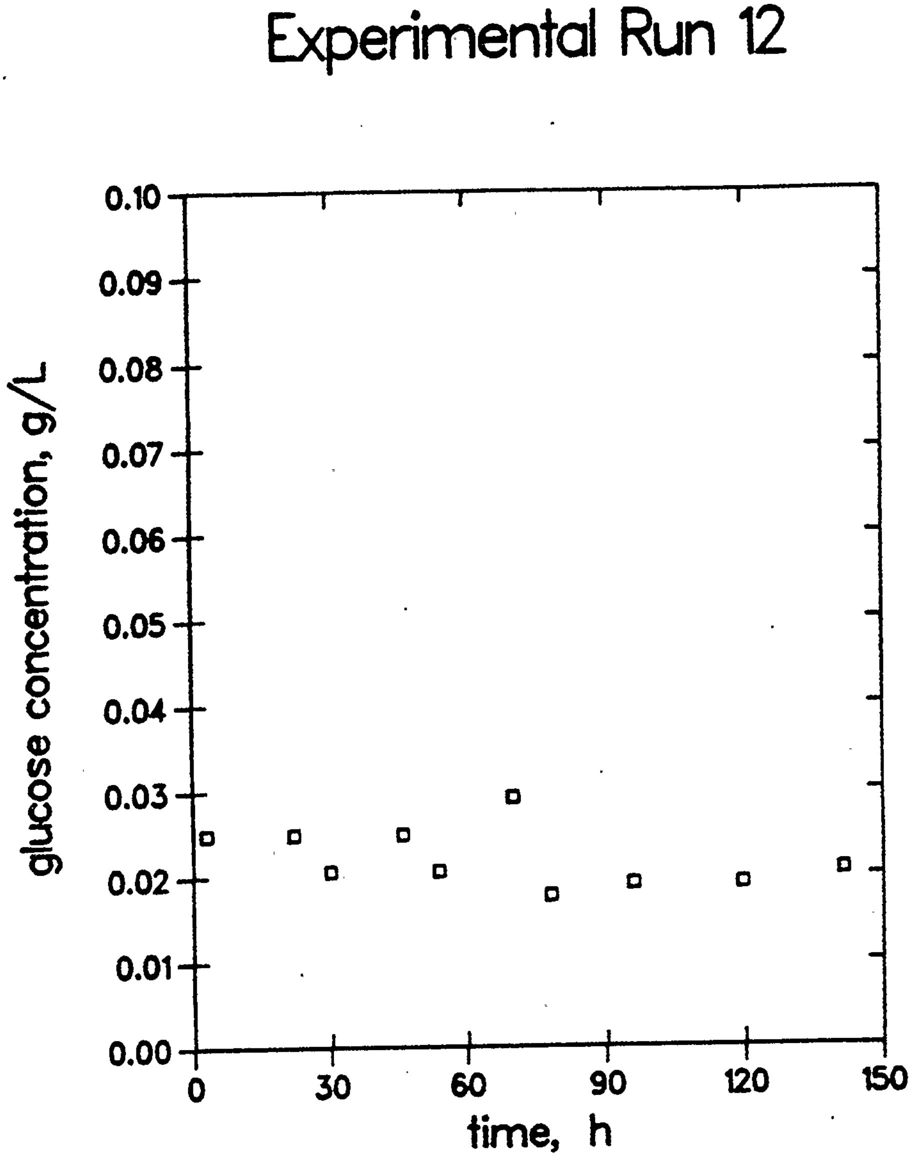

FIGURE 32 Glucose concentration versus Time in Run #12.

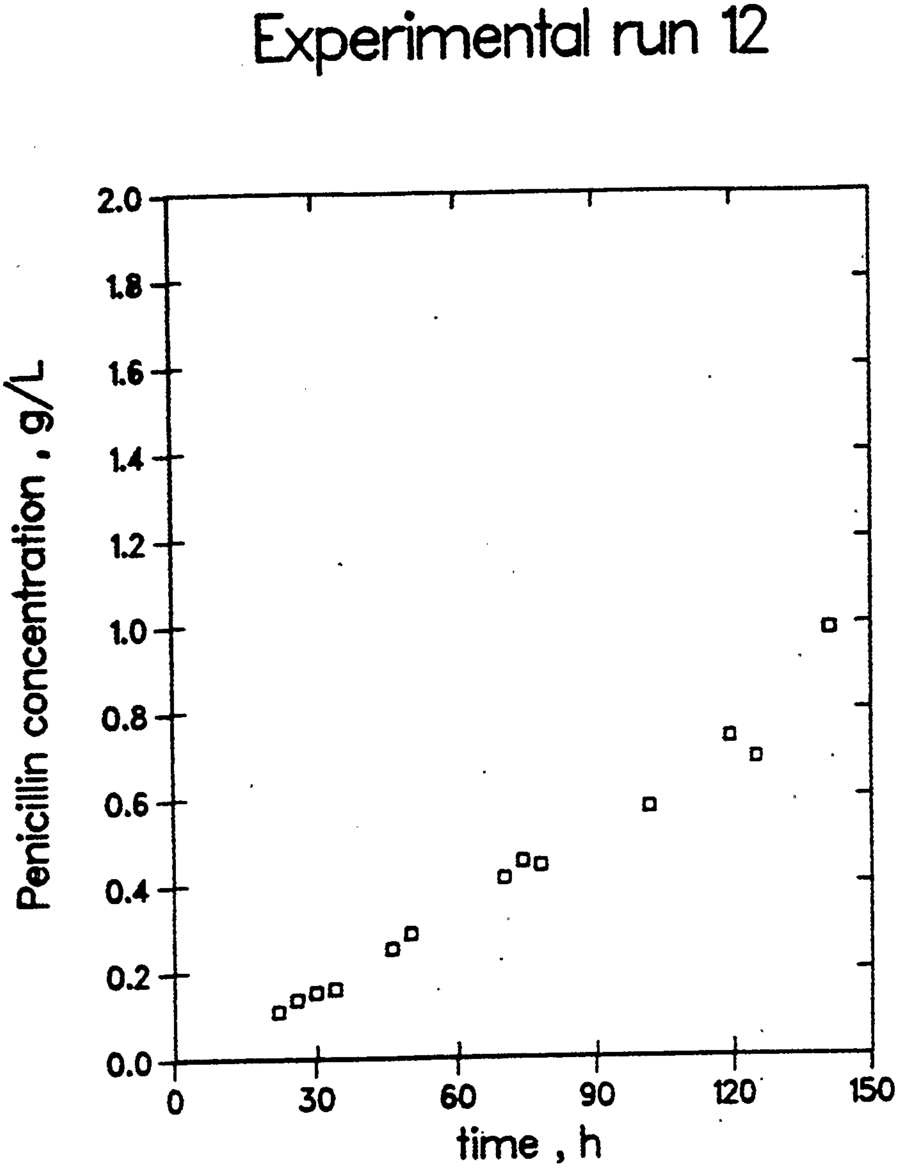

FIGURE 33 Penicillin concentration versus Time in Run #12.

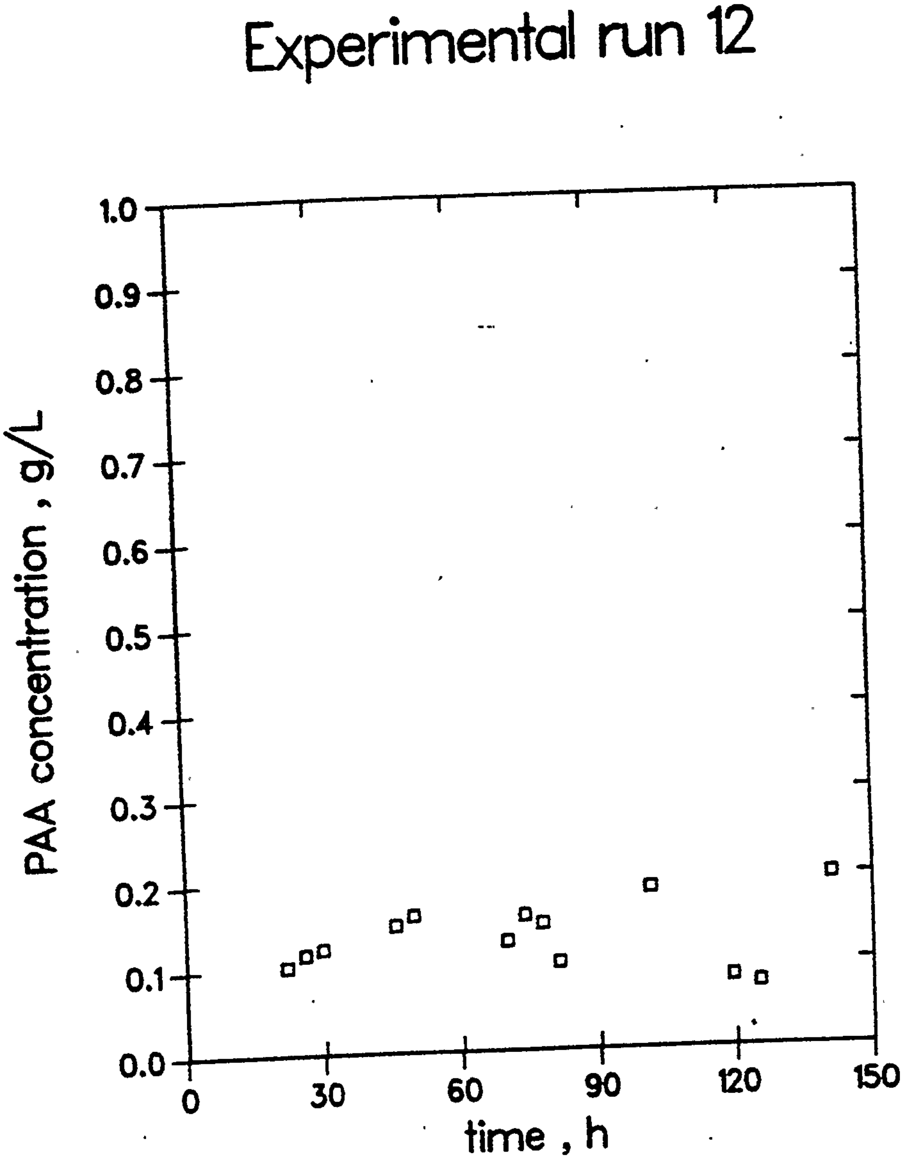

FIGURE 34 PAA concetration versus Time in Run #12.

$$\sigma = \frac{\mu}{Y_G} + m + \frac{q_P}{Y_P} \tag{72}$$

was solved for m, to give

$$m = \sigma - \frac{\mu}{Y_G} - \frac{q_P}{Y_P} \tag{73}$$

In the calculation it was assumed that the value of glucose to biomass yield, Y_G, remained unchanged, since it expresses the stoichiometric conversion of glucose to biomass. This assumption is in agreement to the observations of Konig et al. (1982). The value of q_p had to be calculated from an mass balance for penicillin at the steady state and Y_P was taken as constant with a value of 1.2 (Pirt, 1985).

According to these calculations for Run #4 q_p was equal to 0.00036 h^{-1} and m was approximately 0.006 h^{-1}. For Run #5 q_p was 0.0013 h^{-1} and m equal to 0.009 h^{-1}. Finally for Run #12 q_p was 0.0083 h^{-1} and a negative value close to zero was found for the maintenance energy. For Run #12 there are some reservations about the accuracy of the feed flow rate of medium A. Such a situation would have a strong effect on value of m due to the high feed concentration of glucose but would not affect the value of q_p significantly due to the small value of the Penicillin concentration in the reactor.

Another important observation had to do with unexpectedly low glucose concentrations at the steady state. The maximum growth rate was found in a growth experiment to have a value of 0.12 h^{-1} in good agreement to the values found in in the literature. Therefore, the saturation constant k_s should take a small value. In fact, for Runs

#4, #5 and #12 the value of k_s was found to be 0.33, 0.08 and 0.28 g/L respectively. In comparison, the value commonly found in literature is 1.0 g/L.

A simulation run has been performed using ACSL on Cyber 175 to investigate the adequacy of the model. The values of the parameters in the model are those estimated from the performed experiments.

The results of this simulation run are shown in the next two figures. The logarithm of the biomass concentration is shown in Fig. 35. As seen, the model predictions are within the scatter of the experimental data. The growth rate drops very slowly as indicated by the changing slope of $ln(x)$ versus time. This drop is due to slightly erroneous kinetic parameters used in the open loop program when the experiment was conducted. When at time t = 40 h the set point of the growth rate was changed from 0.005 to 0.008 h^{-1}, the actual growth rate increased slightly, however, it started dropping again as expected. Finally, the agreement seems to be satisfactory for the penicillin concentration, as shown in Fig. 36.

It should be noted at this point that the above simulation run is a good indication of the model performance. However, it does not constitute a scholastic proof of the model validity. The values of the kinetic parameters under the employed fermentation conditions and for the particular strain used, should be accurately established first and then the model should be tested under a diverse set of operating conditions.

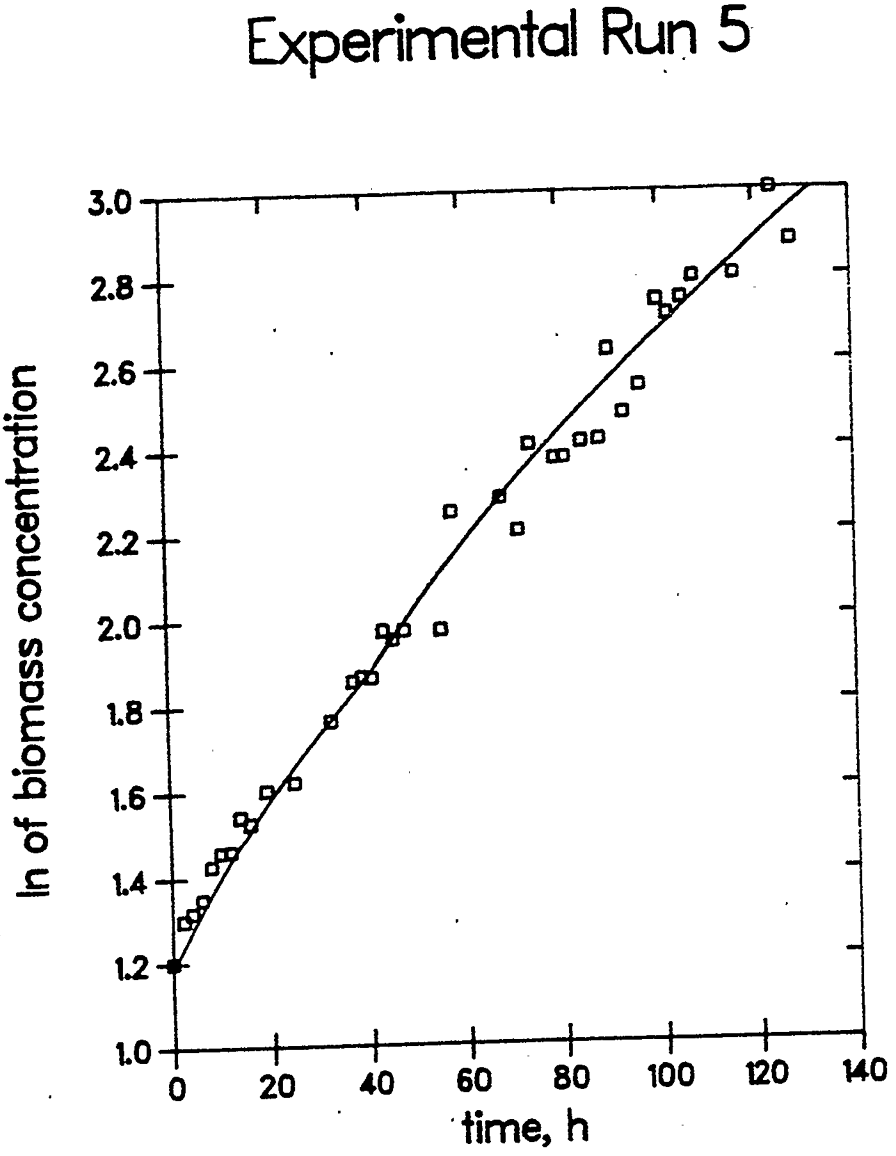

FIGURE 35 ln of Biomass concentration versus Time in Run #5 compared to the model predictions.

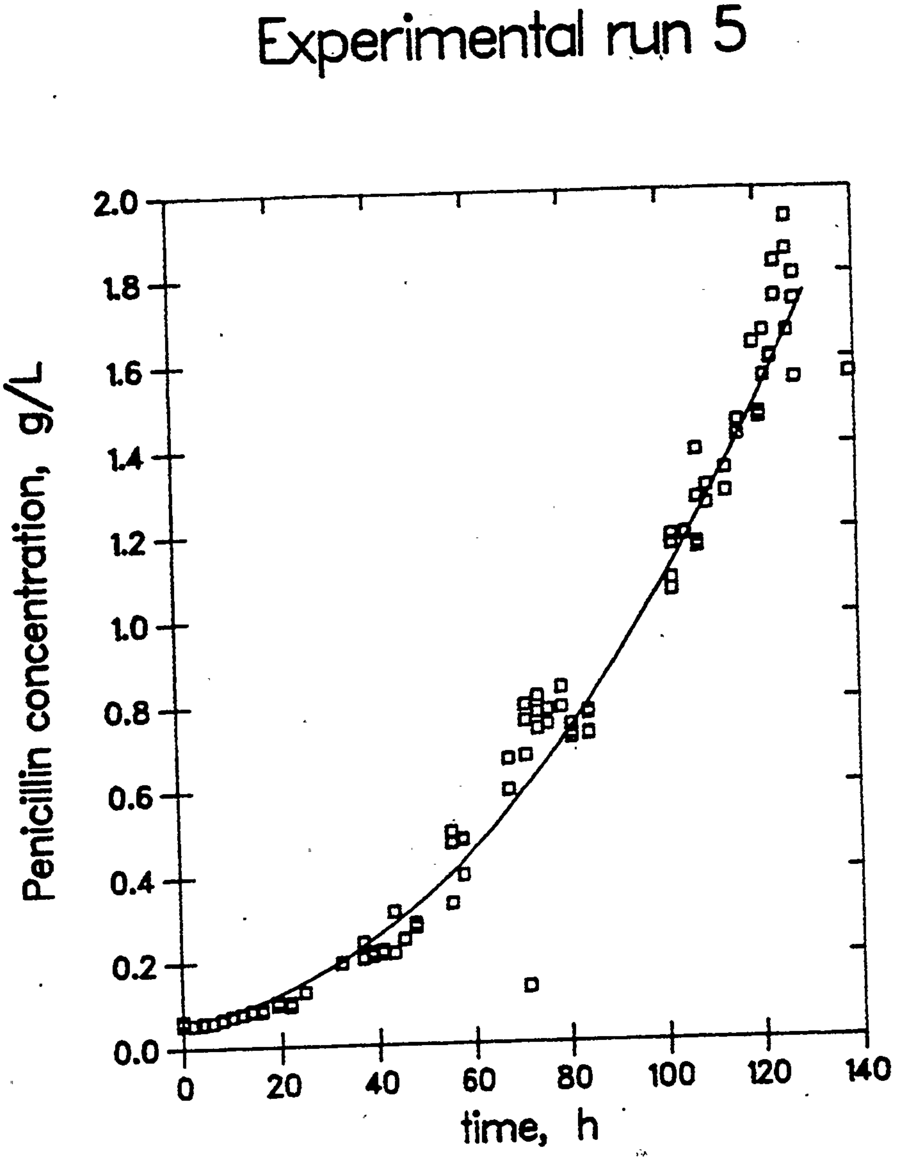

FIGURE 36 Penicillin concentration versus Time in Run #5, compared to the model predictions.

10.4 Further implications of the experimental work

10.4.1 Economic advantages from a reduced maintenance energy.

In the free cell fermentations the value of maintenance energy is approximately 0.026 h^{-1}. In the present study the estimate for the value of maintenance energy for immobilized cell cultures was approximately 0.008 h^{-1}. Assuming a q_p value of 0.01 h^{-1} for an industrial strain of *P. chrysogenum* and an optimum growth rate of 0.015 h^{-1} during the production phase, the value of σ is equal to 0.064 h^{-1} for a free cell system and only 0.046 h^{-1} for the immobilized cell system. The difference corresponds to glucose savings of up to 30%.

Swartz (1979) reports that the cost of glucose in an industrial fermentation accounts for approximately 12% of the 20.72/*kgPen−G* total fermentation cost. Therefore in 1979 approximately 2.48/*kgbulkPen−G* were spent on glucose. With an annual production of approximately 20,000 tonnes the total amount spent on glucose was of the order of 50 million dollars. It is obvious that 30% savings on this amount is a strong incentive for the continuation of research on immobilized cell production of penicillin.

10.4.2 Areas of future process development

The immobilized cell fermentations performed in this work were not truly continuous because after a certain biomass concentration value (approximately 20 g/L) the beads could not support any more cells.

For that reason the free cell population increased very fast after this point and so did the viscosity, while the dissolved oxygen dropped to very low values.

A continuous process would offer very high economic advantages due to minimization of the "turnaround time" for harvest, clean up and sterilization. When the biomass concentration has reached a desired high value and before mass transfer problems appear, heavily populated beads can start being removed continuously and fresh beads inoculated with spores added. This way a constant biomass concentration can be achieved with the growth rate and penicillin concentration also constant at optimum values. Under these conditions the reactor can be run continuously for as long as other factors (e.g. contamination) do not become limiting.

In addition to the establishment of the techniques for continuous operation, some important design parameters must be optimized. For example, the use of smaller size beads can result in higher biomass loadings per unit volume of Celite and an optimum distributor design can increase the biomass concentration where oxygen limitation becomes critical. Furthermore, optimum medium design may be a significant factor in controlling the morphology of the pellets and consequently the mass transfer characteristics of the broth (Kim et al., 1986). The net result of the above improvements will be a higher steady-state Pen-G titer and favourable economics so that continuous antibiotic fermentations will become attractive.

11.0 CONCLUSIONS AND RECOMMENDATIONS

11.1 Conclusions

(1) A simple mathematical model has been developed that describes the most important dynamic characteristics of continuous immobilized cell penicillin fermentations.

(2) An easily implementable non-interactive control system was designed to control the growth rate, the outlet penicillin concentration and the precursor concentration by manipulating the dilution rate and the concentrations of glucose and precursor in the feed stream.

(3) Simulation runs showed that an Extended Kalman Filter can be used for the estimation of biomass concentration from measurements of the penicillin concentration, the PAA concentration and the CO_2 concentration in the effluent gas.

(4) Experimental runs confirmed the existence of a Quasi-Steady-State predicted by the proposed model where the glucose concentration and the growth rate remain constant while the biomass increases exponentially.

(5) The value of the maintenance energy under the employed fermentation conditions appeared drastically reduced to 0.008 h^{-1}.

(6) The experimental data indicate that the value of the saturation constant in the Monod equation has a very small value (approxi-

mately 0.3 *g/L*) for the fermentation conditions and the strain used. This resulted in very low residual glucose concentration in the reactor.

11.2 Recommendations for further work

(1) A number of experimental runs should be conducted at different growth rates, for the accurate determination of all important model parameters, i.e. maintenance energy, m, glucose to biomass yield, Y_G, and maximum specific penicillin productivity q_p^{max}.

(2) For the implementation of the Kalman Filter in an experimental run, the parameters k_4, k_5 and k_6 must also be evaluated from the results of the above proposed runs.

(3) Fine tuning of the model must follow with the determination of parameters that have smaller effect on its performance, namely maximum growth rate μ_{max}, Monod saturation constant k_s and glucose to penicillin yield Y_P.

(4) A technique for the on-line measurement of Penicillin-G and PAA concentration should be developed, possibly based on a modification of the already existing HPLC system.

(5) The problem of free cells must be addressed by improving the effectiveness of aeration so that lower air flow rates can be used. In addition, experiments should be performed with smaller bead size, in order to achieve a higher biomass loading per unit volume of beads.

12.0 REFERENCES

1. Aiba S., S. Nagai and Y. Nishizawa, "Fed Batch Culture of *Saccharomyces cerevisiae* : A Perspective of Computer Control to Enhance the Productivity in Baker's Yeast Cultivation", Biotech. Bioeng., 18, 1001-1016 (1976).

2. Calam C.T. and B.A.K. Ismail, "Investigation of Factors in the Optimization of Penicillin Production", J. Chem. Tech. Biotech., 30, 249-262 (1980).

3. Callegos, J.A and J.A Callegos, "Estimation and Control Techniques for Continuous Culture Fermentation Processes", Biotech. Bioeng., 26, 442-451 (1984).

4. Chibata I., T. Tosa and M. Fujimura, "Immobilized Living Microbial Cells", Ann. Rep. on Ferm. Proc., 6, 1-23 (1983).

5. Deindoerfer F.H. and E.L.Gaden, "Effects of liquid physical properties on oxygen transfer in penicillin fermentation", Appl. Microbiol., 3, 253 (1955).

6. 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).

7. Fishman V.M. and V.V. Biryukov, "Kinetic Model of Secondary Metabolite Production and its Use in Computation of Optimal Conditions", Biotech. Bioeng. Symp., 4, 647-662 (1974).

8. Frueh, K., T.H.Lorenz, J.Niehoff, J.Diekmann, R.Hiddessen and K.Schugerl, "On-Line Measurement and Control of Penicillin V Production", 1st IFAC Symp. on Modelling and Control of Biotechn. Processes, 75 (1985).

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. Gbewonyo K. and D.I.C Wang, "Enhancing Gas-Liquid Mass Transfer Rate in Non-Newtonian Fermentations by Confining Mycelial Growth to Microbeads in a Bubble Column", Biotech. Bioeng., 25, 2873-2887 (1983).

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

12. Holmberg A. and J. Ranta, "Procedures for Parameter and State Estimation of Microbial Growth Process Models", Automatica, 18, 181- 193 (1982).

13. 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).

14. Kalogerakis N. and T.J.Boyle, "Experimental Evaluation of a Quasi-Steady-State Controller for Yeast Fermentation", Biotech. Bioeng., 23, 921-38 (1981).

15. Kalogerakis, N., L.A. Behie and G.M. Gaucher, "Simulation and Optimization of a Continuous Fluidized Bed Bioreactor Producing the Antibiotic Penicillin-G", Proc. *V*th International Conference on Fluidization, Elsinore, Denmark, May(1986).

16. 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).

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

18. Meyer C. and W. Beyeler, "Control Strategies for Continuous Bioprocesses Based on Biological Activities", Biotech. Bioeng., 26, 916-925 (1984).

19. 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).

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

21. Mou, D.G. "Biochemical Engineering and β-Lactam Antibiotic Production", Handbook of Experimental Pharmacology, A.L.Demain and N.A.Solomon, Eds., Springer-Verlag (1983), p.255.

22. Mou D.G and C.L.Cooney, "Growth monitoring and Control through Computer-Aided On-Line Mass Balancing in a Fed-Batch Penicillin Fermentation", Biotech. Bioeng., 25, 225-255 (1983).

23. Mou D.G and C.L.Cooney, "Growth monitoring and Control in Complex Medium: A Case Study Employing Fed-Batch Penicillin Fermentation and Computer-Aided On-Line Mass Balancing", Biotech. Bioeng. , 25, 257-269 (1983).

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

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

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

27. Swartz R.W., "The Use of Economic Analysis of Penicillin G Manufacturing Costs in Establishing Priorities for Fermentation Process Improvement", Ann. Rep. Ferment. Proc., 3, 75-110 (1979).

28. Wang H.Y, C.L. Cooney and D.I.C Wang, "Computer-Aided Baker's Yeast Fermentations", Biotech. Bioeng., 19, 69-86 (1977).

29. Wittler R., R. Matthes and K.Schugerl, "Rheology of *Penicillium chrysogenum* Pellet Suspensions", Eur. J. Appl. Microbiol. Biotechnol., 18, 17-23 (1983).

APPENDIX A

MEDIA COMPOSITION

A.1. Sporulation Medium

The sporulation (surface growth) medium contained per liter: peptone 6 *g*, casaminoacids 4 *g*, yeast extract 3 *g*, beef extract 1.5 *g*, malt extract 20 *g*, bactoagar 40 *g*, glucose 1 *g*, KH_2PO_4 20 *g*, $CaCl_2$ solution 2 *ml* and trace metal solution 10 *ml*.

The trace metal solution contained per liter: $MgSO_4.7H_2O$ 25 *g*, $FeSO_4.7H_2O$ 10 *g*, $ZnSO_4.7H_2O$ 10 *g*, $MnSO_4.H_2O$ 2 *g*, and $CuSO_4.5H_2O$ 0.5 *g*. The concentration of the $CaCl_2$ solution was 25 *g/L*.

A.2. Growth medium

The composition of the growth medium per liter was: glucose 100 *g*, NH_4Cl 10.8 *g*, ammonium acetate 12.0 *g*, sodium lactate 21.6 *g*, Na_2SO_4 2.0 *g*, trace metal solution 40 *ml*, $CaCl_2$ solution 8.0 *ml*, KH_2PO_4 12.0 *g*. The trace metal and $CaCl_2$ solutions used were the same as in sporulation medium.

A.3. Production Medium

The production medium contained per liter: glucose 200 *g*, NH_4Cl 40 *g*, K_2SO_4 71 *g*, $MgSO_4.7H_2O$ 5 *g*, KH_2PO_4 27 *g*. Production medium A contained no PAA, while medium B contained 7 *g/L* of PAA.

APPENDIX B

ANALYTICAL METHODS

B.1. Biomass Measurement

Biomass accumulation was estimated by determining dry cell weight. Total biomass, which included free and immobilized cells, was determined by the following procedure.

A 50-75 *ml* sample of the fermentation broth containing cells and Celite beads was collected on Whatman No. 1. filter paper by vacuum filtration. The mat of cells and Celite was rinsed with deionized water and dried to constant weight at 90 °*C* for 12-14 hours. The dried cells and Celite were weighed and then burned in muffle furnace at 500 °*C* for a minimum of 1 *h*. Biomass concentration was calculated by taking the weight difference between the oven dried sample and the burned sample and converting to *g–dry cell weight/L* by dividing by the weight of Celite in the sample and multiplying by the concentration of Celite in the reactor. The cell ash produced during the burning was accounted for in the calculations. The ash content of the cells was taken as 5% of the dry mass.

To determine the immobilized biomass concentration, a 25-50 *ml* sample of fermentation broth was washed free of free cells by repeatedly adding deionized water and decanting away the liquid containing the free cells. Usually 6-8 washes were sufficient to remove the cells, as determined by microscopic inspection of the decanted liquid. The washed beads were then dried and burned as described for the total biomass determination.

B.2 Glucose measurement

Glucose concentration in the fermentation broth was determined by the Worthington Statzyme (500 *nm*) glucose assay (Cooper Biomedical Inc., Malvern PA). A 20 *μl* sample of fermentation broth filtrate was assayed following the procedure provided with the assay kit. The detection limit was approximately 1 *mg/L*.

B.3 Penicillin-G and PAA measurements

Penicillin-G and PAA were quantitated by High Performance Liquid Chromatography (HPLC) on a Hewlett-Packard 1084B liquid chromatograph equipped with an automatic sample injector and a variable wavelength UV detector. Samples of fermentation broth were filtered and then centrifuged to remove particulate matter prior to injection onto the HPLC. All samples were analyzed on a 25 *cm* x 0.46 *cm* RP-8 column (Brownlee Lab. Inc., Santa Clara, CA) using the following conditions

solvent A : 0.075 M NaH_2PO_4, pH 4.7
solvent B : acetonitrile (CH_3CN)

Elution program :	Time	0 *min*	%B = 10
	"	15 *min*	" 35
	"	18 *min*	" 35
	"	21 *min*	" 10
	"	25 *min*	stop run

Solvent Flow Rate = 1.0 *ml/min*

Detector Wavelength 220 *nm*

Quantitation of Penicillin-G and PAA were carried out by the external standard method.

APPENDIX C

COMPUTER CONTROL AND DATA ACQUISITION SOFTWARE

DATA ACQUISITION AND CONTROL SOFTWARE

FOR OPEN LOOP RUNS

```
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,&h0c
10 DATA &h00,&h8c,&hc8,&h8e,&hd8,&h8e,&hd0,&hc6,&h87,&h04,&h00,&h00,&h8d,&ha7
11 DATA &h0d,&h01,&hb4,&h3d,&hb0,&h00,&h8d,&h97,&h05,&h00,&h53,&hcd,&h21,&h72
12 DATA &h17,&h5b,&h53,&h8d,&h97,&H00,&h00,&h50,&h8b,&hd8,&hb4,&h3f,&hb9,&h04
13 DATA &h00,&hcd,&h21,&h5b,&h72,&H04,&hb4,&h3e,&hcd,&h21,&h5b,&h73,&h09,&h89
14 DATA &h87,&h00,&h00,&hc6,&h87,&h04,&h00,&hff,&h8b,&ha7,&h0c,&h00,&h8e,&h97
15 DATA &h0e,&h00,&h07,&h1f,&hcb
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 No
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(O+48)+256*PEEK(O+49)
47 :
48 STINM=PEEK(O+50)+256*PEEK(O+51)
49 STINS=PEEK(O+52)+256*PEEK(O+53)
50 STINTS=PEEK(O+54)+256*PEEK(O+55)
51 CINM=PEEK(O+56)+256*PEEK(O+57)
52 CINS=PEEK(O+58)+256*PEEK(O+59)
53 CINTS=PEEK(O+60)+256*PEEK(O+61)
54 CSET=PEEK(O+62)+256*PEEK(O+63)
55 BEEPFUN=PEEK(O+64)+256*PEEK(O+65)
56 CONTIN=PEEK(O+66)+256*PEEK(O+67)
57 DELAY=PEEK(O+68)+256*PEEK(O+69)
58 DINS=PEEK(O+70)+256*PEEK(O+71)
59 DOUS=PEEK(O+72)+256*PEEK(O+73)
60 NORMAL=PEEK(O+74)+256*PEEK(O+75)
61 STATS=PEEK(O+76)+256*PEEK(O+77)
62 STOPFUN=PEEK(O+78)+256*PEEK(O+79)
63 VERSION=PEEK(O+80)+256*PEEK(O+81)
64 TINFM=PEEK(O+82)+256*PEEK(O+83)
65 TLIN=PEEK(O+84)+256*PEEK(O+85)
66 TINM=PEEK(O+86)+256*PEEK(O+87)
67 TINS=PEEK(O+88)+256*PEEK(O+89)
68 TINSC=PEEK(O+90)+256*PEEK(O+91)
69 TINTS=PEEK(O+92)+256*PEEK(O+93)
70 :
71 :
100 '        OPEN LOOP RUN PROGRAM   - NO PREDICTION
110 '                   Rev. 1.0  -MARCH 1986 -TED L.
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),RAW3%(500)
215 DIM OFFP(3),IMIN(3),SLP(3),VCOM(4)
230 '------------------Kinetic data----------------
240 READ K1 ,K2,KA0,KA1,YD,MGY,BETA,K3,K4,K5,K6,YP
250 DATA 0.123,1.0,0.001,0.0145,0.5,0.008,0.407,0.001,3.0,0.018,2.625,1.2
260 '-----------------Controller Parameters -----------
270 READ KCP,NF,SA,SP
280 DATA 0.001,0.,200.,12
290 '-----------------Pump Calibration data ---------------
293  IMIN(1) = 5.2    :IMIN(2) = 5.2   :IMIN (3) =5.2
297  OFFP(1) =4.8003 :OFFP(2) =4.9242 :OFFP(3) =4.7868
299  SLP (1) = 2.6101 :SLP (2) =2.4571 :SLP(3) = 3.0048
300  PA%=1 :PB%=3:PO%=2
301  IAMIN = IMIN(1) :OFFA =OFFP(1) : SLA = SLP(1) :' Defaults for gr. phase
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: FOMAX= (20!-OFFO)/SLO
```

```
308  FAMIN=(IAMIN-OFFA)/SLA :FBMIN=(IBMIN-OFFB)/SLB:FOMIN=(IOMIN-OFFO)/SLO
309 LPRINT " ### PUMP ARRANGEMENT : A , B , O  : "; PA%;"   ";PB%;"   ";PO%
310 '-----------------Input Calibration data ---------------
320  READ PO0   , POSL
322  DATA  -.9305928   , 108.50313
325  READ TM0    , 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$(70):ON KEY (15) GOSUB  5300:KEY (15) ON
450 ON KEY ( 1) GOSUB  4000:KEY ( 1) ON
460 ON KEY ( 2) GOSUB  4680:KEY ( 2) ON
470 ON KEY ( 3) GOSUB  4800:KEY ( 3) ON
480 ON KEY ( 4) GOSUB  5560:KEY ( 4) ON
490 ON KEY ( 5) GOSUB 58000:KEY ( 5) ON
495 ON KEY ( 6) GOSUB 14000:KEY ( 6) ON
500 ON KEY ( 7) GOSUB 56000:KEY ( 7) ON
505 ON KEY ( 8) GOSUB 55000:KEY ( 8) ON
510 ON KEY ( 9) GOSUB  7000:KEY ( 9) ON
520 ON KEY (10) GOSUB  3420:KEY (10) ON
530 '------------- Initialize ISAAC  -  OUTPUTS   ------
550 OADIN =4095  :OADFF =0  :OOFF = 4   :OSPAN  =16
552 AOCH1%=0    :AOCH2%=2  :AOCH3%=1
560 AOPT$="device 1"    :OSLOPE = OSPAN/OADIN
562 '-------------- Initialize ISAAC -   INPUTS   ------
565 IADIN =4095  :IADFF =0  : IOFF =0   :ISPAN  =1!
570 AICH1%=0    :AICH2%=15 :AICH3%=1   :AICH4%=3
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 ";
```

```
690 LOCATE 8,1: PRINT"------------------------------------------------------------
700 LOCATE 11,21 :PRINT "Revision 1.0 --- MARCH , 1986"
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(3)=.3124.: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
800 LOCATE 14,15: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 OFLES <> "" 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$(3) ="   *** Alarm Limits *****  "
1260 '
1270 GOSUB 56000
1550 BMASS0=8!:SGR0=2.1:P0=0:AGE0=20:PAA0=.01: MT%=4 :CLS
1570 FGC%=0 :ISML%=1 : TMO=0 :DAY=86400!
1580 '
1590 GOSUB  3440 :'------- Set Points - Initializations --------------
1640 '
1670 XEXP#=BMASS0
1740 '
1760 LOCATE 4,1: PRINT "-------------------------------------------------------
1780 LOCATE 8,20: PRINT "FOR HELP ON ACCEPTABLE KEYBOARD INTERRUPTS";
1800 LOCATE 11,35: PRINT "press <F1>";
1820 LOCATE 15,1: PRINT "------------------------------------------------------
1840 LOCATE 19,20 : PRINT "When you Enter <CR> , The Timer STARTS ...";
1860 LOCATE 25,71: PRINT TIME$
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 LPRINT USING "###.##";TMEL;
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) =";
2180 LOCATE 2,72 : PRINT USING "###.###";TMEL;
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$;
3480 LOCATE 3,1:PRINT "-----------------------------------------------------------
3500 LOCATE 5,25: PRINT "UPDATING  SETPOINTS";
3520 LOCATE 7,1: PRINT "-----------------------------------------------------------
3540 LOCATE 12,15: INPUT "Enter desired growth rate (1/h) ";RXD1
3560 IF RXD1=0 THEN 3620 ELSE RXD=RXD1
3580 RSD = RXD/YD + MGY +KA0/YP :  SD = K2* RXD / (K1-RXD)
3620 '
3640 '
3700 LOCATE 22,15: INPUT "Enter desired PAA conc. (g/L) ";PAAD1
3720 IF PAAD1=0 AND PHASE% >1 THEN 3780
3740 IF (PAAD1<.001 OR PAAD1>5!) AND PHASE% > 1  THEN CLS:LOCATE 20,15:
3760 PAAD=PAAD1
3820 '
3840 LPRINT " ### SET POINTS RXD,PAAD : ";RXD;"   ";PAAD
3940 CLS: FGK$="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 ";
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
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 AOCH1%,VALA%,AOPT$,AOCH2%,VALB%,AOCH3%,VALO%
6815 GOSUB 15000
6816  CALL AOUS (AOCH1%,VALA%,AOPT$)       :' Call ISAAC'S output routine
6818  CALL AOUS (AOCH2%,VALB%,AOPT$)
6820  CALL AOUS (AOCH3%,VALO%,AOPT$)
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(AICH2%,COUNT,HZ,RAW2%(0),OPT$)
6933  CALL AINM (AICH3%, COUNT,HZ,RAW3%(0),OPT$)
6934 SUM1=0:SUM2=0 :SUM3=0 :FOR IR%=1 TO COUNT
6935 SUM2 =SUM2 +RAW2%(IR%-1) :SUM3=SUM3 +RAW3%(IR%-1)
6936 NEXT IR%
6937 VPO%=SUM2/COUNT:VTM%=SUM3/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 GOSUB 8000 : 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;"
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$="N" THEN 7200
7180 LOCATE 16,10:PRINT "Then do you want to correct D ,based on it(y/n)?"
7185 A$=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$="off":RETURN 2080
7380 '
8000 '------------------- COMMUNICATION   SUBROUTINE ----------------------
8010 ACOM$="":NCOM1=0:NCOM2=0:NCOM3=0
8020 IF PAUSE = TRUE THEN PAUSE = FALSE : PRINT #1,XON$
8030 FOR I%=1 TO 1000:NEXT I%
8040 IF EOF(1) AND PAUSE = FALSE THEN GOTO 8200
```

```
8050 IF PAUSE = TRUE THEN PAUSE =FALSE:PRINT #1,XON$:GOTO 8030
8060 IF LOC(1) >128 THEN PAUSE = TRUE :PRINT #1,XOFF$
8070 A$=INPUT$(LOC(1),#1)
8080 ACOM$=ACOM$+A$
8090 IF LEN(ACOM$) > 210 THEN  ACOM$=RIGHT$(ACOM$,80)
8100 GOTO 8040
8200 NCOM1 = INSTR(NCOM1+1,ACOM$,"!"):NCOM2=INSTR(NCOM2+1,ACOM$,CHR$(13))
8202 NCOM10=NCOM1:NCOM20=NCOM2
8210 IF NCOM1 >0 AND NCOM2 >0 AND NCOM2>NCOM1 THEN GOTO 8400
8220 RETURN
8400 '
8420 NCOM3=INSTR(NCOM1+1,ACOM$,"A"):INL=VAL(MID$(ACOM$,NCOM3+1,1))
8430 VCOMSUM=0
8433 FOR I%=1 TO 4
8435 NCOM3 = INSTR(NCOM3+1,ACOM$,";")
8438 VCOM(I%)=VAL(MID$(ACOM$,NCOM3+1,5)):VCOMSUM=VCOMSUM+VCOM(I%)
8440 NEXT I%
8450 FOR I% = 1 TO 4:VCOM(I%) = VCOM(I%)*100/VCOMSUM:NEXT I%
8460 INL=INL+1
8470 Y(INL)=VCOM(4):Y(INL+2)=VCOM(1):T(INL)=TMEL:T(INL+2)=TMEL
8475 IF NCOM2<LEN(ACOM$) THEN GOTO 8200
8480 RETURN
8500 '
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 TQ 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$="N" THEN 56100
56440 CLS:LOCATE 10,1 :PRINT"PO2 input parameters  "PO0;POSL:VAL0=PO0:VSL=POSL
56442 LOCATE 12,1 :INPUT "New values for offset,slope";PO0,POSL
56444 IF ABS(PO0)<.000001 AND POSL<.000001 THEN PO0=VAL0:POSL=VSL
56445 LOCATE 15,1 :PRINT"Temp. input parameters   ";TM0,TMSL:VAL0=TM0:VSL=TMSL
56447 LOCATE 17,1 :INPUT  "New values for zero,offset";TM0,TMSL
56448 IF ABS(TM0)<.000001 AND TMSL<.001 THEN TM0=VAL0:TMSL=VSL
56449 LPRINT " ### PARAMETERS PO0 ,PSL ,  TM0 , TMSL :";PO0;"  ";PSL;"  ";TM0;"
56450 LOCATE 19,1:PRINT "CHANGES IN PUMP PARAMETERS ? (y/n)"
56451 A$=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;"   "
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
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)
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) -
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)
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%)) -
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
```

DATA ACQUISITION AND CONTROL SOFTWARE

FOR CLOSED LOOP RUNS

```
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,&h0c
10 DATA &h00,&h8c,&hc8,&h8e,&hd8,&h8e,&hd0,&hc6,&h87,&h04,&h00,&h00,&h8d,&ha7
11 DATA &h0d,&h01,&hb4,&h3d,&hb0,&h00,&h8d,&h97,&h05,&h00,&h53,&hcd,&h21,&h72
12 DATA &h17,&h5b,&h53,&h8d,&h97,&H00,&h00,&h50,&h8b,&hd8,&hb4,&h3f,&hb9,&h04
13 DATA &h00,&hcd,&h21,&h5b,&h72,&H04,&hb4,&h3e,&hcd,&h21,&h5b,&h73,&h09,&h89
14 DATA &h87,&h00,&h00,&hc6,&h87,&h04,&h00,&hff,&h8b,&ha7,&h0c,&h00,&h8e,&h97
15 DATA &h0e,&h00,&h07,&h1f,&hcb
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 No
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(O+48)+256*PEEK(O+49)
47 :
48 STINM=PEEK(O+50)+256*PEEK(O+51)
49 STINS=PEEK(O+52)+256*PEEK(O+53)
50 STINTS=PEEK(O+54)+256*PEEK(O+55)
51 CINM=PEEK(O+56)+256*PEEK(O+57)
52 CINS=PEEK(O+58)+256*PEEK(O+59)
53 CINTS=PEEK(O+60)+256*PEEK(O+61)
54 CSET=PEEK(O+62)+256*PEEK(O+63)
55 BEEPFUN=PEEK(O+64)+256*PEEK(O+65)
56 CONTIN=PEEK(O+66)+256*PEEK(O+67)
57 DELAY=PEEK(O+68)+256*PEEK(O+69)
58 DINS=PEEK(O+70)+256*PEEK(O+71)
59 DOUS=PEEK(O+72)+256*PEEK(O+73)
60 NORMAL=PEEK(O+74)+256*PEEK(O+75)
61 STATS=PEEK(O+76)+256*PEEK(O+77)
62 STOPFUN=PEEK(O+78)+256*PEEK(O+79)
63 VERSION=PEEK(O+80)+256*PEEK(O+81)
64 TINFM=PEEK(O+82)+256*PEEK(O+83)
65 TLIN=PEEK(O+84)+256*PEEK(O+85)
66 TINM=PEEK(O+86)+256*PEEK(O+87)
67 TINS=PEEK(O+88)+256*PEEK(O+89)
68 TINSC=PEEK(O+90)+256*PEEK(O+91)
69 TINTS=PEEK(O+92)+256*PEEK(O+93)
70 :
71 :
100 '                           DATA AQUSITION AND CONTROL SYSTEM
110 '                 Rev. 1.13  -NOVEMBER 1985 -NICK K.,TED L.
120 '-------------------------------------------------------------------------
130 ON ERROR GOTO 11000
140 '------------------dimensions-----------------
150 DIM WT(4,20),W(4),K(4,4),XV(4),CO(4),CO1(4),DIL(10)
160 DIM MA(4,4),MB(4,4),MC(4,4),MD(4,4),MATI(4,4),Y(2),FORMAT%(5)
170 DIM H(4),HXMT(4,4),HXTMT(4,4),X1(4),DMIN(3)
180 DIM FXMT(4,4),FXTMT(4,4)
190 DIM QMT(4,4),PMT(4,4),RMT(4,4)
200 DIM CNV(2,2),JNV(23)
220 DIM LAST.GRF(3),FORMAT.GRF%(4),PLT.GRF(291),BC$(20),PLV(900)
230 '------------------Kinetic data---------------
240 READ K1 ,K2,KA0,KA1,YD,MGY,BETA,K3,K4,K5,K6
250 DATA 0.123,1.0,0.008,0.0145,0.5,0.026,0.407,0.001,3.0,0.018,2.625
260 '-----------------Controller Parameters -----------
```

```
270 READ KCD,KCS,KCP,TI,FAMIN,FAMAX,FBMIN,FBMAX,NF,SA,SP,FM
280 DATA -1.0,0.001, 0.001,0.5,0.0100,0.55,0.12,5.5,0.,400.,15,0.0050933
290 '------------------Functions----------------
300 DEF FNRX (   S) =K1*S/(K2+S)
310 DEF FNRP ( L,Z) =KA0*KA1*L*EXP(1!-KA1*L)*Z/(K3+Z)
320 DEF FNDIL( TMO) =DILR0*EXP(RXD*(TM#+TM0))
330 DEF FNDL (  TD) =DILR0*EXP(RXD*TD)
340 DEF FNBA (   S) =K1*K2/(K2+S)/(K2+S)
350 DEF FNBD (RP,L) =RP*(1-KA1*L)/L
360 DEF FNBE (RP,Z) =RP/(K3+Z)/Z*K3
370 DEF FNFLR( D,V) =D*V*1000/60
380 '------------------Flag initialization--------
390 FGBR$="off":FGTS$="off":FGPR$="off":FGSP$="off":FGINT$="on":FGMS$="on"
400 '------------------Key definitions----------
410 KEY 16,CHR$(44)+CHR$(70):ON KEY (16) GOSUB  5300:KEY (16) ON
420 KEY 17,CHR$(24)+CHR$(70):ON KEY (17) GOSUB  5300:KEY (17) ON
430 KEY 18,CHR$(64)+CHR$(70):ON KEY (18) GOSUB  5300:KEY (18) ON
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
500 ON KEY ( 7) GOSUB 57000:KEY ( 7) ON
510 ON KEY ( 9) GOSUB  7480:KEY ( 9) ON
520 ON KEY (10) GOSUB  3420:KEY (10) ON
530 '-------------- Isaac Initialization ---------
540 AOCHAN%=0 :AICHAN%=0 :STAT%=0 :VALIN%=0
550 AOPT$=""              :INOPT$=""
560 CALL SETSTAT(STAT%)
570 '----------------------------------------------
580  CO(1)=0:  CO(2)=.5:  CO(3)=.5:  CO(4)= 1
590 CO1(1)=0: CO1(2)=.5: CO1(3)= 0: CO1(4)=.5
600 FOR I%=0 TO 4:FOR J%=0 TO 4
610 FXMT(I%,J%)=0:HXMT(I%,J%)=0:MATI(I%,J%)=0:PMT(I%,J%)=0:QMT(I%,J%)=0
620 IF I%=J% THEN MATI(I%,J%)=1:PMT(I%,J%)=(I%+1)*.1#
630 NEXT J%:NEXT I%
640 QMT(0,0)=.0002:QMT(1,1)=.0002:QMT(2,2)=.0002
650 CLS:SCREEN 0,0,0:WIDTH 80:KEY OFF
660 LOCATE 2,1: PRINT"-----------------------------------------------------------
670 LOCATE 4, 25: PRINT "DATA AQUISITION AND CONTROL SYSTEM";
680 LOCATE 6, 21: PRINT "For Penicillin_G Continuous Fermentations"
690 LOCATE 8,1: PRINT"-----------------------------------------------------------
700 LOCATE 11,27 :PRINT "Revision 1.13 --- November,1985"
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
770 '---------------Reactor Parameter-------------------------------------
773 'LOCATE 9,15: INPUT " Volume of Liquid (L) ";VOL
774 'LOCATE 11,15:INPUT "What is the air flow (L/min) ";GF
775 VOL=15:GF=15
780 '
790 '------------------Sampling interval---------------------------------
800 LOCATE 14,15:INPUT"Please Enter Sampling Interval (min) ";TSAMPLE%
810 N2%=TSAMPLE%/10:TSH#=TSAMPLE%/60#:TSAMPLE=TSAMPLE%*60
820 '------------------Integration parameters---------------------------
830 LOCATE 16,15:INPUT "How many steps per sampling interval";NINRK%
```

```
840 '------------------Display parameters-------------------------------
845 LOCATE 18,15
850 INPUT "Rate of updating the display in a sampl.int.(min)";UR%
860 LOCATE 20,15:PRINT "Should be a dividor of Sampl. Interval"
870 N1%=TSAMPLE%/UR%:IF N1%*UR%<>TSAMPLE% THEN CLS:BEEP:GOTO 850
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"
960  OPEN OFLE$+".dat" FOR APPEND AS 1 :CLS
970 '
980 KILL "plot1.dat":OPEN "plot1.dat" FOR APPEND AS 3
1000 '------------------------------------------------------------------
1010 LOVAL=0:HIVAL=4095:TYPE%=4:PLETE%=1:BACKGROUND%=0
1020 FORMAT%(0)=3:FORMAT%(1)=-1:FORMAT%(2)=0:FORMAT%(3)=0:FORMAT%(4)=0
1030 XLABEL$="":YLABEL$="":COMMENT$="test"
1040 '------------------------------------------------------------------
1050 BC$(1)=" BIOMASS CONCENTRATION    (G/L)"
1060 BC$(2)=" glucose concentration    (g/l)"
1070 BC$(3)=" Pen_G   concentration    (g/l)"
1080 BC$(4)=" PAA     concentration    (g/l)"
1090 BC$(5)=" average cell age         (h)   "
1100 BC$(6)=" dilution rate            (1/h)"
1110 BC$(7)=" glycose feed conc.       (g/l)"
1120 BC$(8)=" measured growth rate     (1/h)"
1130 BC$(9)=" biomass covariance "
1140 BC$(10)=" glycose  covariance"
1160 BC$(11)=" Pen_G    covariance"
1170 BC$(12)=" PAA      covariance"
1180 BC$(13)=" Av. cell age covariance"
1190 '--------------------------------------------------------------------
1550 BMASS0=3!:SGR0=.194:P0=0:AGE0=20:PAA0=.1: MT%=4 :CLS
1560 XV(0)=BMASS0:XV(1)=SGR0:XV(2)=P0:XV(3)=AGE0:XV(4)=PAA0:WT0=XV(0)
1580 GOSUB  3440 :'------- Set Points - Initializations --------------
1620 FGC%=0 :ISML%=1 : TMO=0 :DAY=86400!
1640 '
1660 EP1=PDES:EP=PDES:RSH=RSD:BRP=BETA*RPD:D=0!:RXM=FNRX(XV(1))
1740 '
1760 LOCATE 4,1: PRINT "---------------------------------------------------------
1780 LOCATE 8,20: PRINT "FOR HELP ON ACCEPTABLE KEYBOARD INTERRUPTS";
1800 LOCATE 11,35: PRINT "press <F1>";
1820 LOCATE 15,1: PRINT "--------------------------------------------------------
1840 LOCATE 19,20 : PRINT "When you Enter <CR> , The Timer STARTS ...";
1860 LOCATE 25,71: PRINT TIME$
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:CLS
1990 '
2000 GOSUB 6280
2120 '
2140 IF TIMER<TIME1 THEN TIME1=TIME1-DAY:IF TIMER<TIME0 THEN TIME0=TIME0-DAY
2142 'IF FGC%=10 AND (TIMER-TIME1)>560 THEN LPRINT TIMER-TIME1,TSAMPLE-12,FGMS$
2145 IF (TIMER-TIME1)>(TSAMPLE-12) AND FGMS$="on" THEN GOSUB 7300
2150 IF FGINT$="on" THEN 2152 ELSE 2160
```

```
2152 GOSUB 7580
2154 GOSUB 8240
2156 FGINT$="off"
2160 J1%=INT((TIMER-TIME1)/(TSAMPLE/N1%)):TMEL=(TIMER-TIME0)/3600!
2170 LOCATE 1,52 : PRINT "ELAPSED TIME (h) =";
2180 LOCATE 1,72 : PRINT USING "###.###";TMEL:LOCATE 2,8
2250 PRINT "************ J1%= ";J1% :LOCATE 3,8
2300 PRINT"___________N1%=  ";N1%:LOCATE 4,8
2320 PRINT "Dilution Factor (1/h)      =" TAB(35) ;
2340 PRINT USING "###.#######";D :LOCATE 5,8
2360  PRINT "Feed flowrate (mL/min)     =" TAB(35);
2380 PRINT USING "###.###";FNFLR(D,VOL) :LOCATE 6,8
2385 PRINT "Current Liquid Volume (L) =" TAB(35) ;
2390 PRINT USING "###.###";VOL  :LOCATE 7,8
2400 PRINT "Glucose Feed conc. (g/L)  ="  TAB(35);
2420 PRINT USING "###.##";SF    :LOCATE 8,8
2422 PRINT "Exp. mean cell AGE  (h)    =" TAB(35) ;
2424 PRINT USING "###.###";WT (3,J1%)  :LOCATE 9,8
2425 PRINT "Exp. Glucose conc.(g/L)   =" TAB(35) ;
2430 PRINT USING "###.####";WT (1,J1%) :LOCATE 10,8
2435 PRINT "Expected PAA concen. (g/L)=" TAB(35) ;
2436 PRINT USING "###.####";WT (4,J1%) ;
2437 PRINT TAB(47);"..S.P=" ;:PRINT USING "###.#####";PAAD :LOCATE 11,8
2440 PRINT "Expected Cell conc. (g/L) =" TAB(35) ;
2460 PRINT USING "###.######";WT(0,J1%) :LOCATE 12,8
2480 PRINT "Expected GROWTH RATE(1/h)=" TAB(35) ;
2490 PRINT USING "###.######";FNRX(WT(1,J1%)) ;
2500 PRINT TAB(47);"..S.P=" ;:PRINT USING "###.#####";RXD :LOCATE 13,8
2540 PRINT "Expected Pen_G conc.(g/L) =" TAB(35) ;
2560 PRINT USING "###.######";WT (2,J1%);
2580 PRINT TAB(47);"..S.P=" ;:PRINT USING "###.#####";PDES :LOCATE 15,8
2640 PRINT FAN%,FBN%,FPN%,FGC% :LOCATE 16,8
2660 PRINT VALA%,VALB%,VALP%   :LOCATE 17,1
2670 PRINT "-------------------------------------------------------------------
2680 LOCATE 19,7:PRINT "ENTER DESIRED COMMAND..." TAB(56) "PRESS <F1> FOR HELP"
2700 LOCATE 20,8:PRINT ISML%,J1%;:PRINT USING "######.";TIMER-TIME1;
2710 PRINT TSAMPLE/N1%
2720 LOCATE 21,1: PRINT "--------------------------------------------------------
2740 LOCATE 23,35:PRINT XV(0);XV(4)
2880 TMEL = (TIMER - TIME0)/3600!
2940 A$=INKEY$
2960 IF A$="" THEN 3000
2980 LOCATE 22,5:  PRINT "Can't understand --> ";A$;
3000 LOCATE 25,1: PRINT "System waiting...          " TAB(71) TIME$;
3020 IF FGBR$="on" GOTO 3140
3060 IF FGSP$="on" THEN GOSUB 3440
3080 IF FGPR$="on" THEN GOSUB 6080
3120 GOTO 2140
3140 '
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 "UPDATE  CONTROLLER  SETPOINTS";
3520 LOCATE 7,1: PRINT "-----------------------------------------------------
3540 LOCATE 12,15: INPUT "Enter desired growth rate (1/h) ";RXD1
3560 IF RXD1=0 THEN 3620
3580 IF RXD1<.001 OR RXD1>.12 THEN CLS:LOCATE 10,15:
3600 RXD=RXD1
3620 LOCATE 17,15: INPUT "Enter desired Pen_G conc. (g/L) ";PDES1
3640 IF PDES1=0 THEN 3700
3660 IF PDES1<.001 OR PDES1>30! THEN CLS:LOCATE 10,15:
3680 PDES=PDES1
3700 LOCATE 22,15: INPUT "Enter desired PAA conc. (g/L) ";PAAD1
3720 IF PAAD1=0 THEN 3780
3740 IF PAAD1<.001 OR PAAD1>5! THEN CLS:LOCATE 20,15:
3760 PAAD=PAAD1
3780 SD=K2*RXD/(K1-RXD)
3800 RPD=FNRP(1!/RXD,PAAD)
3820 RSD=RXD/YD + MGY
3840 SFD=SD+RSD*PDES/RPD
3860 DILRO=BMASSO *RPD/PDES
3880 SFAD=PAAD +BETA*PDES
3900 ALP=SFAD/SFD :BLP=1+ALP*SA/SP
3910 G1=SA/SD-1-SA/SP*PAAD/SD :G2=1/SD-PAAD/SD/SP
3920 '
3940 CLS: FGK$="off" : RETURN
3960 '
3980 '-----------------------------------------------------------------------
4000 '                                  trap help key ... <F1>
4020 '-----------------------------------------------------------------------
4040 '
4060 CLS
4080 LOCATE 25,1: PRINT "Servicing <F1> Key...      " TAB(71) TIME$;
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>................not yet";
4300 LOCATE 12,15: PRINT "<F7>................change parameters ";
4320 LOCATE 13,15: PRINT "<F8>................not yet ";
4340 LOCATE 14,15: PRINT "<F 9>...............not yet ";
4360 LOCATE 15,15: PRINT "<F10>...............change setpoint";
4380 LOCATE 16,15: PRINT "^BREAK..............finish this run";
4400 TIME0=TIMER
4420 LOCATE 19,15: PRINT "----------------------------------------------
4440 LOCATE 21,15: PRINT "Hit any Key to Continue";
4460 C$=INKEY$
4480 IF C$="" AND TIMER - TIME0 < 10 THEN 4460
4500 CLS
4520 IF C$="" THEN'BEEP
```

```
4540 CLS:LOCATE 20,15:PRINT "SORRY TIMEOUT"
4560 RETURN
4580 '
4600 '-----------------------------------------------------------------------
4620 '                                                  trap <F2>  key,  clear screen
4640 '-----------------------------------------------------------------------
4660 '
4680 CLS:RETURN
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: GOTO 5180
4940 IF TSAMPL1%< 1 THEN CLS :LOCATE 10,10:PRINT "A sampling Interval of ";
4960 CLS:LOCATE 8,15:PRINT "NEW Sampling Interval = ";TSAMPL1%;"  (min)";
4980 FGTS$="on":HHINT=TSAMPL1%/NINRK%/60!
5000 LOCATE 11,15:PRINT "new int. interv =";HHINT;"hours"
5020 LOCATE 13,15:INPUT "Would you rather use the OLD ONE ? (y/n)";OK$
5040 IF OK$="N" OR OK$="n" THEN CLS:GOTO 5180
5060 NINRK1%=CINT(NINRK%*TSAMPL1%/TSAMPL0%):CLS
5080 HHINT=TSAMPL1%/NINRK1%/60!:LOCATE 10,15
5100 PRINT "Sampling interval = ";TSAMPL1%;" (min)":LOCATE 12,15
5120 PRINT "Number of integr. per interval=";NINRK1%;"INT.interval ";HHINT
5140 PRINT : INPUT "is that O.K  (y/n) ";OK$:CLS
5160 IF OK$="y" OR OK$="Y" THEN  FGII$="on" :CLS:FGK$="off":RETURN ELSE CLS
5180 LOCATE 8,15 :INPUT "How many integrations per interval";NINRK1%
5200 GOTO 5080
5220 '
5240 '-----------------------------------------------------------------------
5260 '                                                  trap ^BREAK key,  finish run
5280 '-----------------------------------------------------------------------
5300 '
5320 CLS
5340 LOCATE 25,1: PRINT "Servicing  ^BREAK....   " TAB(71) TIME$;
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:
5420 CLS:FGBR$="on"
5440 RETURN
5460 '
5480 '-----------------------------------------------------------------------
5500 '                                                  trap <F4> key,  request logging
5520 '-----------------------------------------------------------------------
5540 '
5560 FGK$="on"
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 TIME0=TIMER
5680 C$=INKEY$
5700 IF C$="" AND TIMER - TIME0 < 15 THEN 5680
```

```
5703 PRINT  "                2            ";GC$
5706 PRINT  "                3            ";PC$
5709 PRINT  "                4            ";PAC$
5712 PRINT  "                5            ";AVA$
5715 PRINT  "                6            ";DR$
5718 PRINT  "                7            ";GFC$
5720 CLS
5721 PRINT  "                8            ";MGR$
5724 PRINT  "-------- Covariances --------------------------------"
5727 PRINT
5730 PRINT  "press          11        for";BV$
5733 PRINT  "               12            ";SV$
5736 PRINT  "               13            ";PV$
5739 PRINT  "               14            ";PAV$
5740 IF C$="" THEN CLS:LOCATE 20,15:PRINT"SORRY TIMEOUT...       ";
5742 PRINT  "               15            ";AGV$
5745 PRINT  "                0             TO EXIT"
5760 IF C$="" THEN RETURN
5780 LOCATE 25,1:'PRINT "Servicing <F4> PRINTER...    " TAB(71) TIME$;
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 FGK$="off" :RETURN
6080 '
6100 '---------------------------------------------------------------------
6120 '                                        write output on Disk
6140 '---------------------------------------------------------------------
6160 '
6180 FGPR$="off"
6200 LOCATE 25,1: PRINT "Writing Output on Disk...      " TAB(71) TIME$;
6220 TMEL= (TIMER - TIME0)/3600!
6240 PRINT #1, USING "   ###.###";TMEL;VOL;XV(0);
6260 RETURN
6280 '*********************************************************************
6300 '               Servicing     Isaac
6320 '-------------------------------------------------------------
6340 KEY(1) STOP: KEY(3) STOP:KEY(4) STOP:KEY(7) STOP:KEY(10) STOP
6342 FGC%=FGC%+1 :CLS
6345 '
6348 IF ISML%=1 AND FGC%=1 THEN GOTO 6850
6350 IF FGC%=11 THEN 6357 ELSE 7080
6355 '
6357 TIME1=TIMER:EP1=EP:FGMS$="on":FGINT$="on"
6360 FGC%=1 :TMO=TMO+TSH# :ISML%=ISML%+1
6380 '
6420 IF FGTS$<>"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:N1%=TSAMPLE%/UR%
6530 '
```

```
6540 IF FGII$<>"on" THEN 6620
6560 NINRK%=NINRK1% : FGII$="off"
6620 '
6640 VALO%=4095/.025*H(0)
6645 IF VALO%>4095 THEN BEEP:VALO%=4095
6660 CALL AOUS(AOCHAN%,VALO%,AOPT$)
6680 PRINT STAT%
6700 CALL AINS(AICHAN%,VALIN%,INOPT$)                  :'------Read Input---
6720 Y(0)=VALIN%*.025/4095
6740 PRINT VALO%,VALIN%
6760 PRINT 1000000!*H(0),1000000!*Y(0),STAT%
6770 '
6775 'LPRINT Y(0),Y(1),Y(2)
6776 'LPRINT H(0),H(1),H(2)
6780 FOR I%=0 TO 4:X1(I%)=0:FOR K%=0 TO 2 :
6800 X1(I%)=X1(I%)+MC(I%,K%)*(Y(K%)-H(K%)):NEXT K%
6820 XV(I%)=XV(I%)+X1(I%):NEXT I%                      :'------Estimates----
6830 '
6835 EP=PDES-XV(2)
6840 RXM=LOG(XV(0)/WT0)/TSH#
6841 'LPRINT XV(0),WT0,RXM
6842 RSH=RSD +KCS*(RXD-RXM)
6845 BRP=BETA*RPD +KCP*(PAAD-XV(4))
6846 WT0=XV(0)
6847 '
6850 D=D+KCD/PDES*((1+TSH#/TI)*EP-EP1) :' -----Dilution Rate
6855 SF=SD+RSH*XV(0)/D                  :' -----Glucose concentration
6860 'SFA=PAAD+BRP*XV(0)/D               :' -----PAA       "
6862 PRINT D,KCD,TSH#,BRP,SFA
6865 '
6870 GOSUB 9720                         :' ----Check constraints
6880 ''
6885 FAN%=CINT(FA/FAMIN*10)
6890 FBN%=CINT(FB/FBMIN*10)
6900 FPN%=CINT(FP/FAMIN*10)
6902 '
6905 IF FAN%>10 THEN  FAF=FA ELSE FAF=FAMIN
6910 IF FBN%>10 THEN  FBF=FB ELSE FBF=FBMIN
6915 IF FPN%>10 THEN  FPF=FP ELSE FPF=FAMIN
6920 '
6925 VALA%=4095*(FAF-FAMIN)/(FAMAX-FAMIN) +1
6930 VALB%=4095*(FBF-FBMIN)/(FBMAX-FBMIN) +1
6935 VALP%=4095*(FPF-FAMIN)/(FAMAX-FAMIN) +1
6940 '
6947 CLOSE #3
6948 OPEN "plot1.dat" FOR APPEND AS 3
6950 PRINT #3,XV(0);XV(1);XV(2);XV(4);XV(3);D;SF;RXM;PMT(0,0);
6960 PRINT #3,PMT(1,1);PMT(2,2);PMT(4,4);PMT(3,3)
6962 CLOSE #3
6990 '
7000 OPEN "RECOVER.DAT" FOR OUTPUT AS 2
7010 PRINT #2,UR%,TSAMPLE%,IPRINT%,OFLE$
7020 PRINT #2,XV(0),XV(1),XV(2),XV(3),XV(4),4
7030 PRINT #2,VOL,GF,NINRK%
7040 PRINT #2,RXD,PDES,PAAD
7050 PRINT #2,D,SF
7060 PRINT #2,PMT(0,0),PMT(1,1),PMT(2,2),PMT(3,3),PMT(4,4)
7070 CLOSE #2
7075 '
7080 IF FGC%>FAN% THEN VALA%=0
```

```
7090 IF FGC%>FBN% THEN VALB%=0
7100 IF FGC%>FPN% THEN VALP%=0
7110 '--------------Add output statements***************************here
7140 '
7260 KEY(1) ON:KEY(3) ON:KEY(4) ON:KEY(7) ON:KEY(10) ON
7280 CLS:IF FGINT$="on" OR FGK$="on" THEN RETURN ELSE RETURN 2140
7300 CLS:LOCATE 7,10:PRINT"You have FIVE minutes to enter measurements"
7310 FGMS$="off"
7320 LOCATE 9,10:PRINT "Press any key to continue"
7340 TIMINIT =TIMER
7360 TIMELEFT=TIMINIT + 10 -TIMER:LOCATE 11,10 :PRINT "only ***",CINT(TIMELEFT),
7380 A$=INKEY$:IF A$<>"" THEN 7460
7400 IF TIMELEFT<=0 THEN 7420 ELSE 7360
7420 Y(1)=XV(2):Y(2)=XV(4):CLS :IW%=0:WHILE IW%<100 :LOCATE 10,20
7440 PRINT "Sorry ...TIMEOUT":IW%=IW%+1 :WEND :RETURN
7460 Y1=Y(1):Y2=Y(2)
7480 CLS:LOCATE 10,10:INPUT "Penicillin concentration (g/l) ";Y(1)
7500 LOCATE 12,10:INPUT     "       PAA concentration (g/l) ";Y(2)
7520 IF Y(1)=0 AND Y1>.0000001 THEN Y(1)=Y1
7540 IF Y(2)=0 THEN Y(2)=Y2
7560 CLS:RETURN
7580 '
7600 FOR I%=0 TO 4:WT(I%,0)=XV(I%):PRINT WT(I%,0),I% : NEXT I%
7620 CN1%=0:HINT#=TSH#/NINRK%
7640 FOR J%=1 TO NINRK%
7660 TM#=(J%-1)*HINT#
7680 IF ABS(TM#*60-CN1%*UR%)<HINT#*60 THEN 7700 ELSE 7740
7700 FOR I%=0 TO 4:WT(I%,CN1%)=XV(I%):NEXT I%
7720 CN1%=CN1%+1
7740 FOR IJ%=1 TO 4
7760 FOR I%=0 TO 4
7780 K(0,I%)=0!
7800 W(I%)=XV(I%)+CO(IJ%)*K(IJ%-1,I%)
7820 NEXT I%
7840 TM#=TM#+CO1(IJ%)*HINT#
7860 RX=FNRX(W(1)):RS=RX/YD +MGY:RP=FNRP(W(3),W(4))
7880 FOR I%=0 TO 4
7900 ON I%+1 GOTO 8120,8140,8160,8180,8200
7920 K(IJ%,I%)=HINT#*FF
7940 NEXT I%
7960 NEXT IJ%
7980 FOR I%=0 TO 4
8020 XV(I%)=XV(I%)+1!/6!*(K(1,I%) +2*K(2,I%)+2*K(3,I%)+K(4,I%))
8040 NEXT I%
8050 PRINT J%
8060 NEXT J%
8080 FOR I%=0 TO 4:WT(I%,N1%)=XV(I%):PRINT WT(I%,N1%);:NEXT I%
8083 PRINT RX:RS=RX/YD +MGY
8086 H(0)=VOL/GF/60*(RX/K4+K5+K6*FNRP(XV(3),XV(4)))*XV(0) +.0003
8087 PRINT RP,FNRP(XV(3),XV(4))
8090 H(1)=XV(2):H(2)=XV(4)
8100 RETURN
8120 FF=RX*W(0):GOTO 7920
8140 FF=-RS*W(0)+D*(SF-W(1)):GOTO 7920
8160 FF=RP*W(0)-D*W(2):GOTO 7920
8180 FF=1!-W(3)*RX :GOTO 7920
8200 FF=-BETA*RP*W(0)+D*(SFA-W(4)):GOTO 7920
8220 '
8240 'Matrix calculations---------------------------------------------
8260 '-------------------------------------------------------------------
```

```
8280 RX=FNRX(XV(1)):RS=RX/YD+MGY:BA=FNBA(XV(1))
8300 RP=FNRP(XV(3),XV(4)):BE=FNBE(RP,XV(4)):BD=FNBD(RP,XV(3))
8320 '------------ Give values to the elements of  matr. A -----------
8340 FXMT(0,0)=RX:FXMT(0,1)=BA*XV(0)
8360 FXMT(1,0)=-RS:FXMT(1,1)=-BA*XV(0)/YD -D:FXMT(2,0)=RP
8380 FXMT(2,2)=-D:FXMT(2,3)=XV(0)*BD:FXMT(2,4)=XV(0)*BE
8400 FXMT(3,1)=-XV(3)*BA:FXMT(3,3)=-RX:FXMT(4,0)=-BETA*RP
8420 FXMT(4,3)=-BETA*XV(0)*BD:FXMT(4,4)=-D-BE*BETA*XV(0)
8440 '
8460 FG=VOL/GF/60
8480 HXMT(0,0)=(1!/K4*RX+K5+K6*RP)*FG:HXMT(0,1)=BA*WT(0,0)/K4*FG
8500 HXMT(0,3)=K6*BD*WT(0,0)*FG:HXMT(0,4)=K6*BE*WT(0,0)*FG
8520 HXMT(1,2)=1:HXMT(2,4)=1
8540 '
8560 'FOR I%=0 TO 4:FOR J%=0 TO 4: PRINT FXMT(I%,J%);
8580 'NEXT J%:PRINT:NEXT I%:PRINT:PRINT
8600 '
8620 FOR I%=0 TO 4:FOR J% =0 TO 4:MC(I%,J%)=0
8640 FOR K%=0 TO 4:MC(I%,J%)=MC(I%,J%)+FXMT(I%,K%)*FXMT(K%,J%)
8660 NEXT K%:MC(I%,J%)=TSH#*(TSH#*MC(I%,J%)/2!+FXMT(I%,J%))+MATI(I%,J%)
8680 'PRINT MC(I%,J%);
8700 FXTMT(J%,I%)=MC(I%,J%):NEXT J%:PRINT:NEXT I% :PRINT:PRINT
8720 '
8740 'FOR I%=0 TO 4:FOR J%=0 TO 4:PRINT FXTMT(I%,J%);
8760 'NEXT J%:PRINT:NEXT I%:PRINT:PRINT
8780 '
8800 FOR I%=0 TO 4:FOR J%=0 TO 4:MA(I%,J%)=0
8820 FOR K%=0 TO 4:MA(I%,J%)=MA(I%,J%)+MC(I%,K%)*PMT(K%,J%)
8840 NEXT K%:NEXT J%:NEXT I%
8860 '
8880 FOR I%=0 TO 4:FOR J%=0 TO 4:MC(I%,J%)=0
8900 FOR K%=0 TO 4:MC(I%,J%)=MC(I%,J%)+MA(I%,K%)*FXTMT(K%,J%)
8920 NEXT K%:PMT(I%,J%)=MC(I%,J%)+RMT(I%,J%)*TSH#
8940 PRINT PMT(I%,J%);:NEXT J%:PRINT:NEXT I%
8960 '
8980 'PRINT :PRINT:FOR I%=0 TO 4:FOR J%=0 TO 4:PRINT HXTMT(I%,J%);
9000 'NEXT J%:PRINT:NEXT I%:PRINT:PRINT
9020 '
9040 FOR I%=0 TO 2:FOR J%=0 TO 4:HXTMT(J%,I%)=HXMT(I%,J%)
9060 NEXT J%:NEXT I%
9080 '
9100 FOR I%= 0 TO 4: FOR J% = 0 TO 2:MA(I%,J%)=0
9120 FOR K% = 0 TO 4:MA (I%,J%)=MA(I%,J%)+PMT(I%,K%)*HXTMT(K%,J%)
9140 NEXT K%:NEXT J%:NEXT I%
9160 '
9180 FOR I%=0 TO 2:FOR J%=0 TO 4:MB(I%,J%)=0
9200 FOR K%=0 TO 4:MB(I%,J%)=MB(I%,J%)+HXMT(I%,K%)*PMT(K%,J%)
9220 NEXT K%:NEXT J%:NEXT I%
9240 '
9260 FOR I%=0 TO 2:FOR J%= 0 TO 2:MC(I%,J%)=0
9280 FOR K%= 0 TO 4:MC(I%,J%)=MC(I%,J%)+HXMT(I%,K%)*MA(K%,J%)
9300 NEXT K%:CNV(I%,J%)=MC(I%,J%)+QMT(I%,J%)
9320 MC(I%,J%)=CNV(I%,J%):PRINT CNV(I%,J%);:NEXT J%:PRINT:NEXT I%
9340 '
9360 PRINT:PRINT:GOSUB 10140
9380 '
9400 FOR I%=0 TO 2:FOR J%=0 TO 2:MD(I%,J%)=0
9420 FOR K%=0 TO 2:MD(I%,J%)=MD(I%,J%)+CNV(I%,K%)*MC(K%,J%)
9440 NEXT K%:PRINT MD(I%,J%);:NEXT J%:PRINT:NEXT I%
9460 '
```

```
9480 FOR I%=0 TO 4:FOR J%=0 TO 2:MC(I%,J%)=0
9500 FOR K%= 0 TO 2:MC(I%,J%)=MC(I%,J%)+ MA(I%,K%)*CNV(K%,J%)
9520 NEXT K%:NEXT J%:NEXT I%
9540 '
9560 PRINT:PRINT
9580 FOR I%= 0 TO 4:FOR J%= 0 TO 4:MA(I%,J%)=0
9600 FOR K% = 0 TO 2:MA(I%,J%)=MA(I%,J%)+MC(I%,K%)*MB(K%,J%)
9620 NEXT K%:PMT(I%,J%)=PMT(I%,J%)-MA(I%,J%)
9640 PRINT PMT(I%,J%);:NEXT J%:FOR II=1 TO 100:NEXT II
9660 PRINT:NEXT I%:FOR II=1 TO 2000:NEXT II
9680 CLS:RETURN
9700 '
9720 '----------------------CONSTRAINTS ON D -----------------------
9760 FA = SF/SA*D*VOL
9764 FPMIN=FAMIN/10:FPMAX=FAMAX
9766 FAMINP=FPMIN*SD*SP/PAAD/SA+ (RSH/SA-BRP*SD/PAAD/SA)*XV(0)*VOL
9768 FAMAXP=FPMAX*SD*SP/PAAD/SA+ (RSH/SA-BRP*SD/PAAD/SA)*XV(0)*VOL
9770 FWMIN=FBMIN/10+NF :FWMAX=FBMAX+NF
9780 FAMINW =(FWMIN+(RSH*G2+BRP/SP)*XV(0)*VOL)/G1
9800 FAMAXW =(FWMAX+(RSH*G2+BRP/SP)*XV(0)*VOL)/G1
9820 FAMMIN =FAMIN/10:FAMMAX=FAMAX
9835 '
9840 IF FAMINP>FAMMIN THEN FAMMIN=FAMINP
9845 IF FAMINW>FAMMIN THEN FAMMIN=FAMINW
9850 '
9860 IF FAMAXW<FAMMAX THEN FAMMAX=FAMAXW
9870 IF FAMAXP<FAMMAX THEN FAMMAX=FAMAXP
9875 '
9880 IF FA<FAMMIN THEN FA=FAMMIN
9900 IF FA>FAMMAX THEN FA=FAMMAX
9910 FP=FA*SA/SP*PAAD/SD -(RSH*PAAD/SD-BRP)*XV(0)*VOL/SP
9920 FW=FA*G1 -(RSH*G2+BRP/SP)*XV(0)*VOL
9925 'LPRINT FA,FAMINP,FAMINW,FP,FW
9940 FT=FA+FW+FP+NF: D=FT/VOL:SF=FA/FT*SA:SFA=FP/FT*SP:FB=FW-NF
10120 RETURN
10140 'subr.  for matrix inversion using G-J elim. with column pivoting
10160 'input :matrix cnv(n%,n%)
10180 'output : inverse cnv, magn. of determinant detm, and dtnrm
10200 PDNV=1!
10220 FOR I%=0 TO 2 : DDNV=0!
10240 FOR K%=0 TO 2 : DDNV=DDNV+CNV(I%,K%)*CNV(I%,K%)
10260 NEXT K%:DDNV=SQR(DDNV):PDNV=PDNV*DDNV
10280 NEXT I%:DETM=1!
10300 FOR I%=0 TO 2:JNV(I%+20)=I%:NEXT I%
10320 '
10340 FOR I%=0 TO 2 : DDNV=0!: M%=I%
10360 FOR K%=I% TO 2 :
10380 IF ABS(DDNV)-ABS(CNV(I%,K%)) >=0! THEN 10420
10400 M%=K% :DDNV= CNV(I%,K%)
10420 NEXT K%
10440 IF I%=M% THEN 10500 ELSE SWAP JNV(M%+20),JNV(I%+20)
10460 FOR K%=0 TO 2
10480 SWAP CNV(K%,M%) ,CNV(K%,I%) :NEXT K%
10500 CNV(I%,I%)=1! :DETM =DETM*DDNV
10520 FOR M%=0 TO 2:CNV(I%,M%)=CNV(I%,M%)/DDNV:NEXT M%
10540 FOR M%=0 TO 2
10560 IF I%= M% THEN 10640 ELSE DDNV=CNV(M%,I%)
10580 IF DDNV=0!      THEN 10640 ELSE CNV(M%,I%)=0!
10600 FOR K%=0 TO 2
10620 CNV(M%,K%)=  CNV(M%,K%)- DDNV* CNV(I%,K%)   :NEXT K%
```

```
10640 NEXT M%
10660 NEXT I%
10680 FOR I%=0 TO 2
10700 IF JNV(I%+20)=I% THEN 10860
10720 M%=I%
10740 M%=M%+1
10760 IF JNV(M%+20)=I% THEN 10800
10780 IF 2 > M% THEN 10740
10800 JNV(M%+20)=JNV(I%+20)
10820 FOR K%=0 TO 2: SWAP CNV(I%,K%),CNV(M%,K%):NEXT K%
10840 JNV(I%+20)=I%
10860 NEXT I%
10880 DETM=ABS(DETM):DTNRM=DETM/PDNV
10900 RETURN
10920 '
10980 '
11000 IF (ERR=24)OR (ERR=25 ) THEN 11020 ELSE 11080
11020 PRINT "****** PRINTER IS NOT  O.K ********"
11040 PRINT "Hit any key to return to main program"
11060 A$=INKEY$:IF A$="" THEN 11060 ELSE CLS:RESUME 6060
11080 IF ERR=6 OR ERR=11 THEN PRINT ERR,ERL:END
11085 IF ERL>=58000 THEN PRINT ERR,ERL:STOP
11090 RESUME NEXT
57000 FGK$="on":VOL0=VOL:GF0=GF:SA0=SA:CLS
57005 INPUT "New  culture volume (lit)";VOL
57010 INPUT "NEW gas flow rate (lit/min)";GF
57015 INPUT "NEW maximum glucose concentration (g/lit)";SA
57020 IF VOL=0 THEN VOL=VOL0
57025 IF  GF=0 THEN  GF= GF0
57030 IF  SA=0 THEN  SA= SA0
57035 PRINT "change in controllers' parameters (y/n)"
57040 C$=INKEY$:IF C$="" THEN 57040
57045 IF C$="y" OR C$="Y" THEN 57065
57050 IF C$="n" OR C$="N" THEN CLS :RETURN
57055 GOTO 57040
57065 KCD0=KCD:KCS0=KCS:TI0=TI
57070 INPUT "NEW D-controller Gain ";KCD
57075 INPUT "NEW D-controller Int. time ";TI
57080 INPUT "new feedb. corr. gain";KCS
57085 IF  TI=0 THEN  TI= TI0
57090 IF KCD=0 THEN KCD=KCD0
57095 IF KCS=0 THEN KCS=KCS0
57100 CLS:FGK$="off" :RETURN
57184 RETURN
58000 FGK$="on" :CLS
58003 PRINT "press            1       for   ";BC$(1)
58006 PRINT "                 2             ";BC$(2)
58009 PRINT "                 3             ";BC$(3)
58012 PRINT "                 4             ";BC$(4)
58015 PRINT "                 5             ";BC$(5)
58018 PRINT "                 6             ";BC$(6)
58021 PRINT "                 7             ";BC$(7)
58024 PRINT "                 8             ";BC$(8)
58027 PRINT
58030 PRINT "                 9             ";BC$(9)
58033 PRINT "                10             ";BC$(10)
58036 PRINT "                11             ";BC$(11)
58039 PRINT "                12             ";BC$(12)
58042 PRINT "                13             ";BC$(13)
58045 PRINT "                 0      TO EXIT"
```

```
58048 PRINT :INPUT "Well....";PLOTIN%:IPL%=0:IPL1%=0
58051 IF PLOTIN%>0 THEN  58057
58054 FGK$="off":CLS:RETURN
58057 OPEN "plot1.dat" FOR INPUT AS 3
58060 INPUT "Start from interval No :";IPST%:INPUT "Stop at No";IPEN%
58061 IF IPEN%<IPST% THEN BEEP: GOTO 58060
58063 IF IPST%=0 THEN IPST%=1
58064 IF IPEN%=0 THEN IPEN%=ISML%
58066 IPL%=IPL%+1:IF EOF(3) THEN 58093
58069 IF IPL%<IPST% OR IPL%>IPEN% THEN IPL1%=0 ELSE IPL1%=IPL1% +1
58072 FOR IP%=1 TO 12
58075 IF IP%=PLOTIN% THEN INPUT #3,PLV(IPL%):GOTO 58081
58078 INPUT #3,NL
58081 NEXT IP%
58084 IF PLOTIN%=13 THEN INPUT #3,PLV(IPL%) ELSE INPUT #3,NL
58087 'PRINT : print plV(1),PLV(2),PLV(3)
58090 GOTO 58066
58093 CLOSE #3: TITLE$=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%=1 TO ISML%
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$="A" THEN 58111
58151 CLS: SCREEN 0,0,0 :WIDTH 80 :GOTO 58003
58153 ' name        INITGRAPH
58156 BASE.GRF%             = 0
58159 I.GRF%                = 0
58162 INPTR.GRF%            = 0
58165 J.GRF%                = 0
58168 NWAVES.GRF%           = 1
58171 XVALUE.GRF            = 0
58174 YVALUE.GRF            = 0
58177 SCREEN 1 : VIEW : WINDOW
58180 KEY OFF
58183 COLOR BACKGROUND%,PLETE%
58186 CLS
58189 FOR I.GRF% = 0 TO 3
58192    FORMAT.GRF%(I.GRF%) = FORMAT%(I.GRF%)
58195 NEXT I.GRF%
58198 WHILE (NWAVES.GRF% < 4) AND FORMAT.GRF%(NWAVES.GRF%) <> -1
58201    NWAVES.GRF% = NWAVES.GRF% + 1
58204 WEND
58207 ON (TYPE% + 1) GOSUB 58267,58267,58267,58312,58312,58312
58210 IF LEN(TITLE$) > 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)
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) -
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)
```

```
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%)) -
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
```

National Library of Canada
Ottawa, Canada
K1A 0N4

Bibliothèque nationale
du Canada

95.

TC –

ISBN

CANADIAN THESES ON MICROFICHE SERVICE – SERVICE DES THÈSES CANADIENNES SUR MICROFICHE

PERMISION TO MICROFILM – AUTORISATION DE MICROFILMER

• Please print or type – Écrire en lettres moulées ou dactylographier

AUTHOR – AUTEUR

Full Name of Author – Nom complet de l'auteur

THEODOROS LINARDOS

Date of Birth – Date de naissance 1960	Canadian Citizen – Citoyen canadien □ Yes / Oui ☑ No / Non
Country of Birth – Lieu de naissance GREECE	Permanent Address – Résidence fixe 541 Rundle Hall, U. of Calgary CALGARY, Alberta T2N-1N4

THESIS – THÈSE

Title of Thesis – Titre de la thèse

"MODELLING AND COMPUTER CONTROL OF CONTINUOUS IMMOBILIZED CELL PENICILLIN FERMENTATIONS"

~~M. Sc.~~

Degree for which thesis was presented Grade pour lequel cette thèse fut présentée M. Sc.	Year this degree conferred Année d'obtention de ce grade 1987
University – Université University of Calgary	Name of Supervisor – Nom du directeur de thèse Nicolas Kalogerakis

AUTHORIZATION – AUTORISATION

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to **microfilm** this thesis and to **lend** or **sell copies of the film.**

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de **microfilmer** cette thèse et de **prêter** ou de **vendre des exemplaires du film.**

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

ATTACH FORM TO THESIS – VEUILLEZ JOINDRE CE FORMULAIRE À LA THÈSE

Signature Linardos	Date April 28, 1987

Canada

Rejected
P.xiv : & 149 the letter
are running into together
&
P.69 the letters are faint &
broken.

GC 26-11-87