INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

Production of Lipase by *Candida bombicola* in a Self-Cycling Fermenter (SCF)

By

Michael May

Department of Chemical Engineering
McGill University, Montreal
February 1997

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfilment of the requirements of the degree of Master of Engineering.

ABSTRACT

The Self-Cycling Fermentation (SCF) technique and an agar diffusion assay were used to follow the production of lipase by *Candida bombicola*. Repeatable lipase production profiles were determined for two different media. Each medium gave a considerably different profile.

When the broth contained oil, 60 to 80% of the lipase activity remained after freezing. All activity was lost when the broth did not contain oil.

The lipase was found to be associated with the cell. Up to a certain concentration of lipase, the presence of agar caused the lipase to disassociate from the cell and diffuse into the agar. Beyond this concentration of lipase the agar was found to saturate.

A new technique of level control in the cyclone reactor was used. A pressure transducer was found to be an accurate and inexpensive device for level control.

RÉSUMÉ

La technique de fermentation auto-cyclique (FAC) et une essai de diffusion en agar ont été utilisé pour suivre la production de la lipase par *Candida bombicola.* Des profils répété ont été obtenu pour deux media différent. Le profil pour chaque medium était considerablement différent.

Quand il'y avait de l'huile végétal dans le bouillon, 60% à 80% du lipase a été maintenue. Tout l'activité a été perdue quand le bouillon n'avait pas d'huile.

La lipase a été associé avec la cellule. Jusqu'à une certain concentration, la présence de l'agar a causé une désassociation, et la lipase s'est diffusé dans l'agar. Au delà de cette concentration, l'agar a été saturer.

Une nouvelle technique utilisant une jauge de pression a été utiliser pour contrôller le niveau du bouillon dans le reactor de type cyclone. Cette technique était précis et pas cher.

ACKNOWLEDGEMENTS

This work was made possible by the financial support of the Natural Sciences and Engineering Research Council (NSERC) by means of a scholarship and a research grant.

My deepest gratitude is extended to Prof.D.G.Cooper, without whose constant attention and support I doubtlessly could not have completed this work.

Many thanks are owed to Wayne Brown who did most of the work developing the pressure transducer idea.

To the entire Falcon Research team, Huan, Kevin, and my Parents, who all supported me when I was feeling low, and generally made the degree so darned fun: thank-you.

Finally, thanks to David Thomson, whose philanthropic decision kept us all on the best of terms.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

1.1 Lipase Production

A lipase is an enzyme that is capable of hydrolysing the ester bond in a triglyceride between a fatty acid and glycerol. What differentiates lipases from esterases is their ability to hydrolyse water insoluble substrates (19). Lipases are defined by their activity and not their form since, like most enzymes, they vary in structure considerably from one organism to the next. Indeed, a single organism may produce several different lipases (17). For this reason lipases are identified by the organism which produces them, *e.g.* "the lipase produced by *Candida bombicola*".

Lipase is used mostly in the food industry, where its uses are diverse. It has been used in food production including cheese, yogurt, soybean milk, and apple wine. Lipase can accelerate production, as well as modify flavours. It can also be used to produce imitation cheese and creamy products by partial hydrolysis of cream. As for the industrial hydrolysis of fats and oils, it produces a superior product compared with the conventional saponification process. The specificity of some lipases can been used advantageously: new fats with specific properties can be produced by esterification or interesterification (29).

Lipase has also long been a problem particularly in the dairy industry. Pasteurization, and now ultra-high temperature (UHT) treatment removes most microbial contamination. Lipase produced before treatment, however, can often remain afterward, causing hydrolysis of the milk fats (7,28). The fatty acids produced from this hydrolysis result in a product with a bad flavour.

1.2 Synchronous Growth

1.2.1 Cell Synchrony

Most fermentation techniques used today, such as batch, fed-batch, and chemostat, consist of a population of cells with a wide distribution of cell ages. That is, some cells would have recently divided and be accumulating nutrients in preparation for future division, whereas others would be in the process of dividing. The organism would be expected to behave differently during these distinct growth stages. These differences will be averaged out when dealing with asynchronous fermentations (13). This averaging is reflected in the cell number which would increases exponentially (under ideal conditions), as in Figure 1.1.

A synchronous culture is one in which all of the cells are at very similar stages in their life cycles. For instance, all of the cells in the culture will divide around the same time (see Figure 1.2). The behaviour of the population in a synchronous system, such as with self-cycling fermentation (SCF), therefore, reflect the behaviour of the individual cells much more closely than with asynchronous systems such as batch and chemostat (13).

Perfect synchrony is never observed; all cultures maintain some variability. Because of this, a method of determining the degree of synchronization of a given culture was developed by Blumenthal and Zahler (1):

$$F = \frac{N_f}{N_o} - 2^{\frac{t}{g}}$$

Where F is the synchrony index and can range from 0, for an asynchronous culture such as batch growth, to 1 for perfect synchrony.

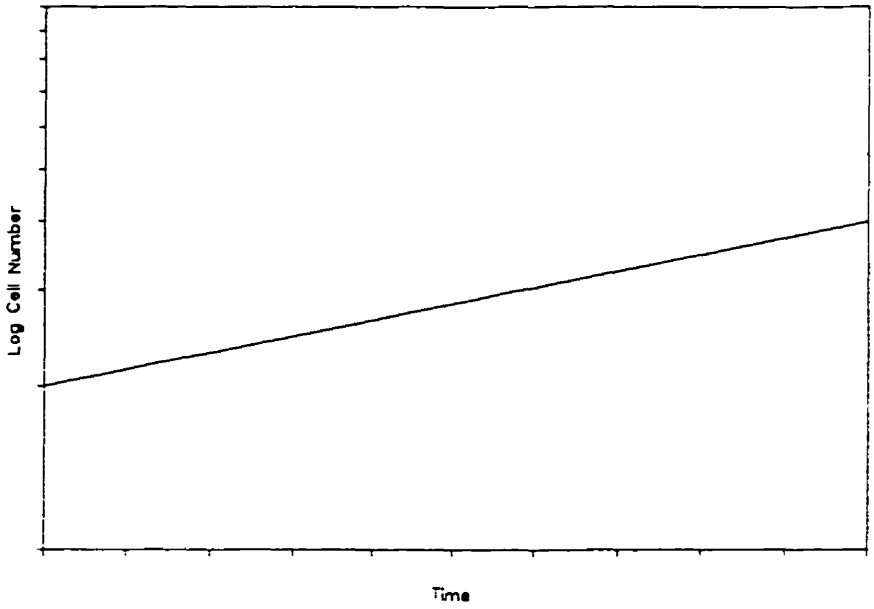


Figure 1.1 Asynchronous growth.

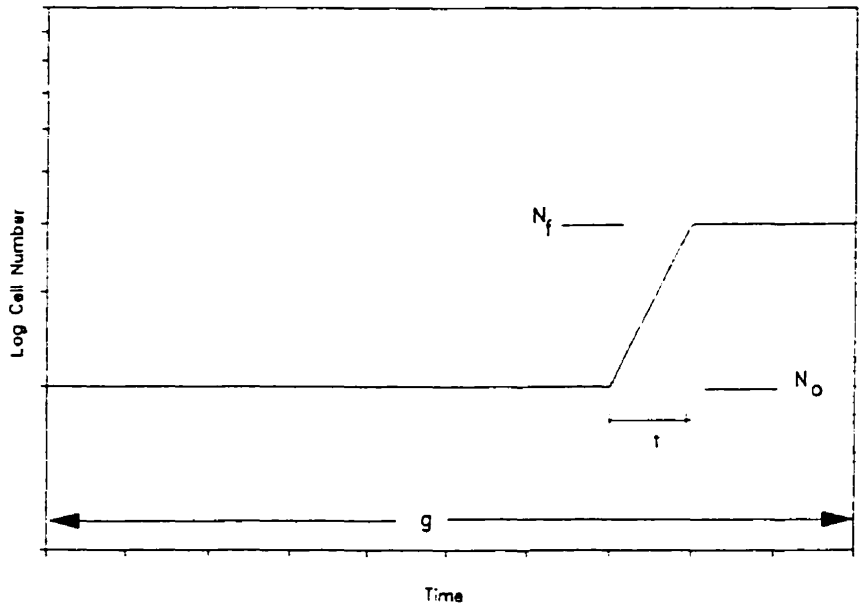


Figure 1.2 Synchronous Growth.
Nf = Final cell number.
No = Initial cell number.
t = Period of time during which cells are dividing.
g = Doubling time.

1.2.2 Self-Cycling Fermentation

Self-cycling fermentation (SCF) is similar to a sequencing batch reactor (SBR) in that the nutrients are periodically "phased", resulting in a sequence of batch reactions. Phasing is the act of removing broth and replacing it with fresh medium. There are, however, important differences between how nutrients are phased in the two systems. In a SBR biomass is allowed to settle and the supernatant and biomass are removed separately. The volumes removed, and the volume of fresh nutrients added are not necessarily equal, or the same from cycle to cycle. In SCF, biomass is not allowed to settle and the broth is kept homogeneous. The amount of fresh medium added is the same for each cycle, and equal to the amount of harvested broth. The major difference between SCF and SBR, however, is that the point of cycling in the SCF is dependent on the metabolic state of the organism. The organisms thus determine the time at which cycling occurs, hence the name self-cycling fermentation. Whereas a SBR, as the name suggests, is simply a sequence of batch reactions, the conditions of SCF result in a significantly different fermentation.

SCF is extremely stable, and can handle fluctuations in medium conditions or metabolic changes (see Figure 1.3) (4,5,21,23,30). With traditional phasing systems the frequency of phasing must be set by the operator (12,13,14). Setting the frequency of phasing can be a problem. If the period between phasing is too short, cell washout will occur, as in each cycle less biomass is present. A period between phasing that is too long will lead to extended nutrient starvation, and possibly cell death, which can affect the future growth and product formation. This method is also

poor at handling changes in the growth conditions. Even if the frequency of phasing were set perfectly, a sudden drop in feed substrate concentration could be detrimental to the stability of the system.

Past work indicates that growth, production, and nutrient consumption rates tend to be much higher in the SCF than in other fermentation methods (4,21,23,31,33). As mentioned earlier, the SCF technique leads to cultures with a high degree of synchrony with a synchrony index around 0.8. SCF has also been found to give very reproducible cycles (see Figure 1.3) (4,21,23,31). These qualities make the SCF ideal for studying growth and metabolite production by almost any biological system.

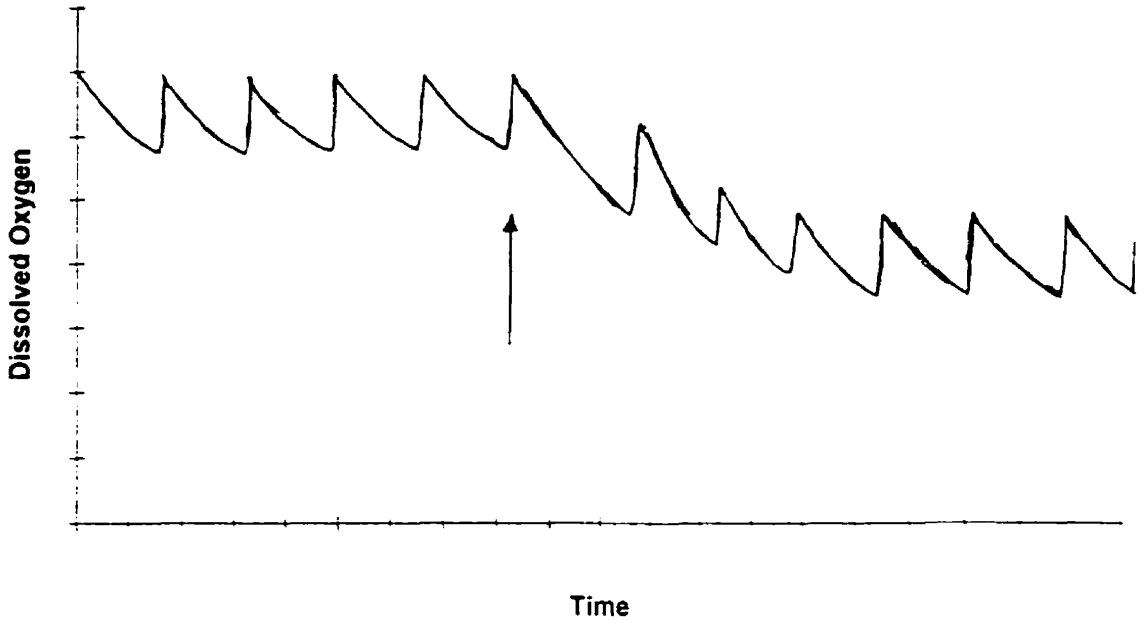


Figure 1.3 Hypothetical dissolved oxygen trace. The hypothetical response of the dissolved oxygen trace to a step increase in substrate concentration is shown. The point at which substrate concentration increases is indicated by the arrow: (↑).

1.3 Substrate Utilization by Micro-Organisms

Biomass production and nutrient utilization in the SCF have been found to be well described by Monod's models (32):

$$\frac{dX}{dt} = \frac{\mu_{max} S}{K_s + S} X$$

$$\frac{dS}{dt} = \frac{1}{Y_{\frac{X}{S}}} \frac{dX}{dt}$$

Where X is biomass and S is substrate. K_s is a constant. μ_{max} is the maximum specific growth rate.

It has been found that biomass grows at the maximum specific growth rate in SCF, so the simplification of $K_s << S$ can be used (32). The solution for biomass and substrate levels is then given by,

$$X = X_0 e^{\mu_{max} t}$$

$$\ln\left(S_0 - S - \frac{X_0}{Y_{\frac{X}{S}}}\right) = \ln\left(-\frac{X_0}{Y_{\frac{X}{S}}}\right) + \mu_{max} t$$

In the SCF, biomass has been shown to double every cycle. The time of doubling is equal to the time to the minimum in dissolved oxygen (26). The maximum specific growth rate is thus inversely proportional to the time to the minimum in dissolved oxygen.

$$\mu_{max} = \frac{\ln(2)}{t_m}$$

Where t_m is the time to the minimum in dissolved oxygen.

2.0 OBJECTIVES

The objectives of this work were to explore the suitability of Self-Cycling Fermentation for the production of microbial metabolites. It is necessary to determine how synchrony affects production by micro-organisms, and to further determine if aspects of growing synchronously can be used to improve the production rates.

McCaffrey had shown that *Candida bombicola* grows well in the Self-Cycling Fermenter, obtaining stable cycles and cell synchrony (21). Because of the stability and relatively short cycle times, it was decided that this system would be ideal to study product formation. This yeast could be grown on vegetable oil as the only carbon source so it must produce a lipase. Lipase is a relatively easy enzyme to monitor, so this was considered suitable.

3.0 MATERIALS AND METHODS

3.1 Fermentation System

3.1.1 Reactor Setup

The reactor setup was similar to that described by McCaffrey (20) but with a new and much simpler method for level control. The electronic balance and the harvesting and dosing vessels were removed and replaced with a differential pressure transducer attached directly to the fermentation vessel. The new fermenter setup can be seen in Figure 3.1.

As in the system described by McCaffrey, Vegetable oil or hexadecane were added to the system using a syringe pump (Sage model 355). A rotameter (Brooks Sho-Rate Model 1355XBIBIAAA) was used to adjust air flowrate to the system. In-line filters were used on the vents and air supply to maintain sterility in the system. These consisted either of glass filter tubes stuffed with glass wool, or Reeve Angel 934AH filter paper in Nalgene in-line filters (Sartorious SM165 98). All valves were Skinner V52LB2100 solenoid valves. The broth was circulated by a March MDX pump. The temperature of the broth was controlled using a side-arm heat-exchanger and a Haake FE2 constant temperature bath. A Friedrich's condenser was used with cold water to prevent excess evaporation. Control and data acquisition were performed with an IBM compatible 8088 with a data acquisition board (Data Translation Model DT-2801). Wiring was routed through a relay board (Opto 22 model PB16A) and a screw terminal board (Data Translation DT707).

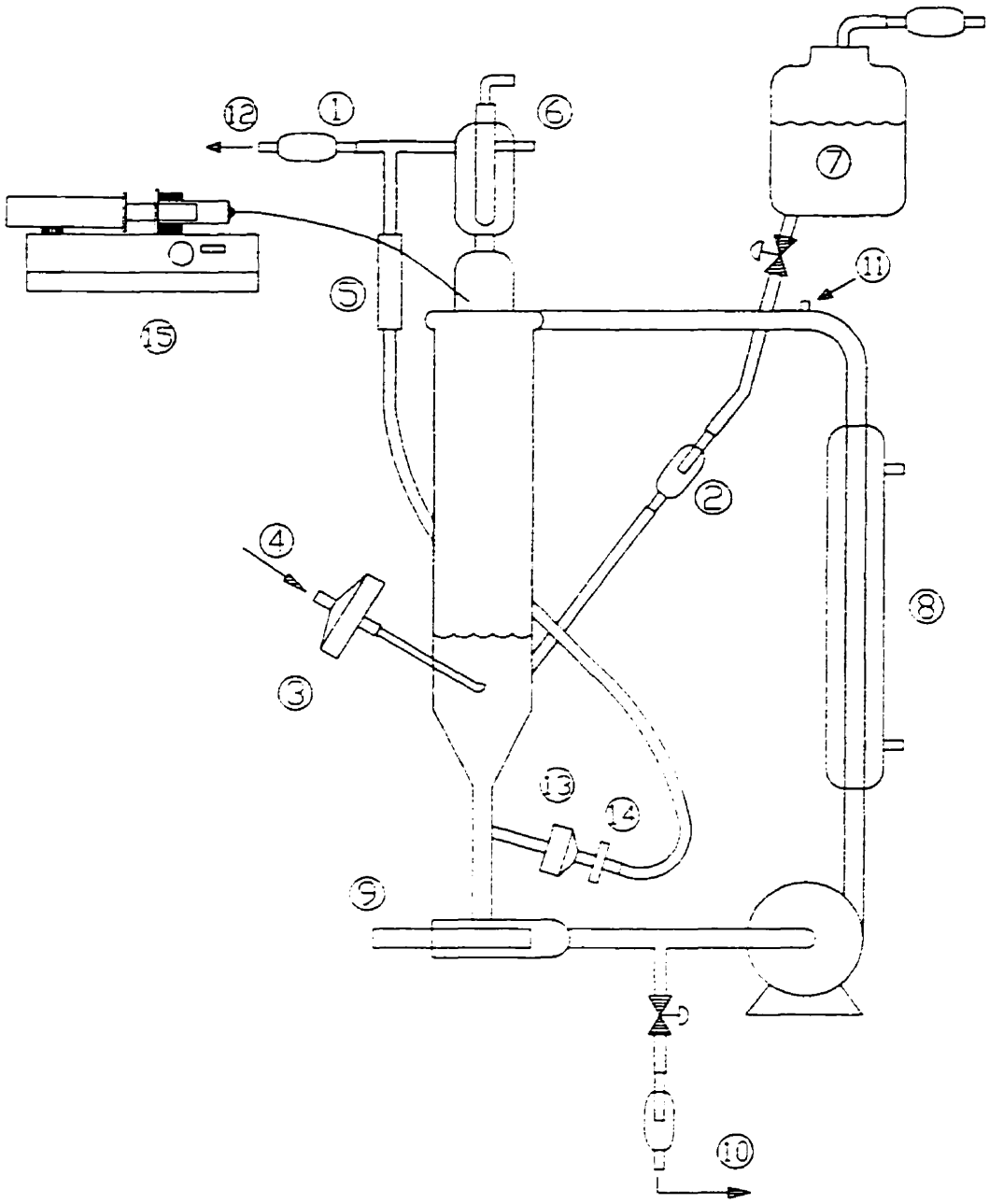


Figure 3.1 Reactor Setup

1,3 - Air Filters
2 - Isolators
4 - Air Supply (from Rotameter)
5 - Desiccant
6 - Condensor
7 - Fresh Medium
8 - Heat Exchanger
9 - Dissolved Oxygen Probe
10 - To Harvesting System
11 - Septum for Sampling
12 - Offgas to Fume Hood
13 - Membrane
14 - Pressure Transducer
15 - Syringe Pump

3.1.2 Pressure Transducer Setup

With the pressure transducer attached to the fermenter, pressure fluctuations caused by the pump gave erratic transducer readings. As a result of this problem the pump was turned off during cycling, and broth was removed by gravity. The harvest line attachment point was moved from near the top of the heat exchanger, to a point at the base of the system near the pump.

An Omega PX170 differential pressure transducer was used. The air side was connected to the off-gas line of the reactor. A tube containing DryRite was used to isolate moisture from the air side of the transducer, which was susceptible to corrosion. The wet side was isolated from the reactor to prevent biomass accumulation within the transducer. Isolation was performed using a Nalgene filter holder (Millipore Swinnex-25) with a 0.45μ Millipore filter (HA). Another advantage of isolating the transducer was that it did not need to be autoclaved. This process would have required careful protection of the air side and electrical leads. The transducer used a bridge circuit to measure pressure. Power was supplied to the transducer by a +12V DC power supply (circuit diagram in appendix). The voltage signal from the transducer was returned through an unshielded cable. Although an amplification circuit was made (circuit diagram in appendix) it was not used since the unamplified signal was found to be satisfactory.

3.1.3 Dissolved Oxygen Concentration

The dissolved oxygen concentration was measured with an Ingold polarographic dissolved oxygen probe (model IL531). The signal from the probe was amplified using a Pegasus amplifier before being sent to the data acquisition board and the strip chart recorder. All values presented are in % saturation of air in water at 29°C. To determine the 100% saturation a measurement was made with the medium after autoclaving and before inoculation. The 0% level was determined after the reactor had been purged with nitrogen.

3.1.4 Control Software

The control software was modified from that used by McCaffrey (20). The code was compiled in Turbo Pascal 7.0. Major modifications include changing the code to accept a signal from the pressure transducer instead of the balance. The timing algorithm was modified to avoid situations of ambiguity as to length of cycles. The new system determines the number of seconds elapsed since the beginning of the run, and takes into account change of day, month (including the extra day in February during leap year) and year. The graphical display was updated to show more pertinent information, such as the exact time at which time delays will end. The system is more adjustable now, including a choice of three new methods for performing level control during cycling. A listing of the important parts of the updated program can be found in the Appendix.

3.2 Micro-Organism

3.2.1 Strain

The organism used was *Candida bombicola* ATCC 22214. The strain was obtained from the American Type Culture Collection, and was periodically checked for contamination by plating on YM or Nutrient agar plates.

3.2.2 Culture Maintenance

To maintain the culture, YM agar slants were inoculated and then incubated at 24 to 30°C until large colonies were visible, which typically took two days. The slants were then transferred to the refrigerator to be stored at 4°C. Fresh slants were inoculated bimonthly. To transfer the culture from one slant to the next a transfer medium was used (22):

Transfer Medium

Component	(g/l)
Glucose	40
Yeast Extract	1
Ammonium Sulfate	3.3

3.2.3 Growth Media

The media used were based on those used by McCaffrey (21). The media consisted of a basic mineral salts medium to which various carbon and nitrogen sources were added.

Mineral Salts Medium

Component	(g/l)
KH_2PO_4	1.0
$MgSO_4 \cdot 7H_2O$	0.5
$CaCl_2$	0.1
NaCl	0.1
Yeast Extract	1.5

The following carbon sources were used either alone or in combination.

Carbon Source	(g/l)
Glucose	10
Mazola Corn Oil	5
Hexadecane	5

Nitrogen was supplied in the form of ammonium salts, at one of the following levels:

Nitrogen Source	(g/l)
Ammonium Nitrate	8
Ammonium Sulfate	4
	2
	1.1

The yeast extract in the base medium also supplied some nitrogen and carbon (22). When glucose was used it was sterilized separately to avoid caramelization. The vegetable oil and hexadecane were added without sterilization.

3.3 Biomass Determination

3.3.1 Optical Density

Optical density measurements could be used as a measure of biomass concentration for samples that did not contain vegetable oil or hexadecane. These measurements were done using a Varian Model DMS200 dual beam scanning UV-visible spectrophotometer at 488nm. Since readings became non-linear above an optical density of 2, samples were diluted with water to one nineteenth concentration to obtain satisfactory readings. About 3ml of diluted sample was placed in the sample cell, and an equivalent amount of distilled water was placed in the reference cell. The optical density reading was taken immediately.

McCaffrey had found that the optical density method could not be used to determine biomass for samples containing hydrocarbons (70). Small droplets of hydrocarbon scattered the transmitted light and caused erratic readings. Vegetable oil in the sample was found to affect the results in the same way as hydrocarbons. Furthermore, the distribution of biomass itself was not very homogenous in the broth. This heterogeneity caused fluctuations in the optical density reading even when oil and hydrocarbons were not present. As a result of these problems most biomass measurements were made using the dry weight method.

3.3.2 Dry Weight

A 25ml sample was centrifuged at 9,000 rpm and ~0°C for 15 minutes in a DuPont Instruments Sorval Model RC-5 refrigerated centrifuge using an SS-34 rotor head. The supernatant was decanted and the cell pellet was washed with 30ml of distilled water and recentrifuged. After decanting, a white oily residue clung to the side of the centrifuge tube. This residue was wiped out using a KimWipe. A second washing and recentrifugation was performed. The cell pellet was then resuspended in a small amount of distilled water and transferred to a tared aluminum weighing dish. The dishes were dried in a Fisher Isotemp oven (100 series model 126G) at 90°C until constant weight. A Mettler Model AE 160 analytical balance was used to measure weight to within 0.1mg. Before weighing, the pans were allowed to cool to room temperature in a desiccator.

3.4 Analysis of Media Components

3.4.1 Glucose Concentration

The glucose concentration was measured using a total carbohydrate analysis technique (10). This involved mixing 1ml of sample with 1ml of 5% phenol solution in a test-tube. 5ml of concentrated Sulfuric Acid was added, mixed, and allowed to sit at room temperature for 10 minutes. The test-tube was transferred to a 25°C constant temperature bath for 15 minutes and the optical density was measured. The samples were diluted 100-fold in order to obtain readings within range of the spectrophotometer. The carbohydrate level was then obtained using the calibration curve shown in Figure 3.2. This curve gives the carbohydrate level in terms of the equivalent concentration of glucose required to obtain the same absorbance.

This test is not specific to glucose. The results would include a measurement of any simple sugars or polysaccharides produced by the organism.

3.4.2 Ammonium Concentration

The ammonium concentration was measured by Conway Microdiffusion Analysis (8). This test used dishes consisting of two concentric chambers over which an air-tight lid would be placed, see Figure 3.3. The sample and a basic mixture in the outer chamber react, causing ammonium ions to convert to ammonia gas.

$$NH_4^+ + OH^- \rightarrow NH_3\uparrow + H_2O$$

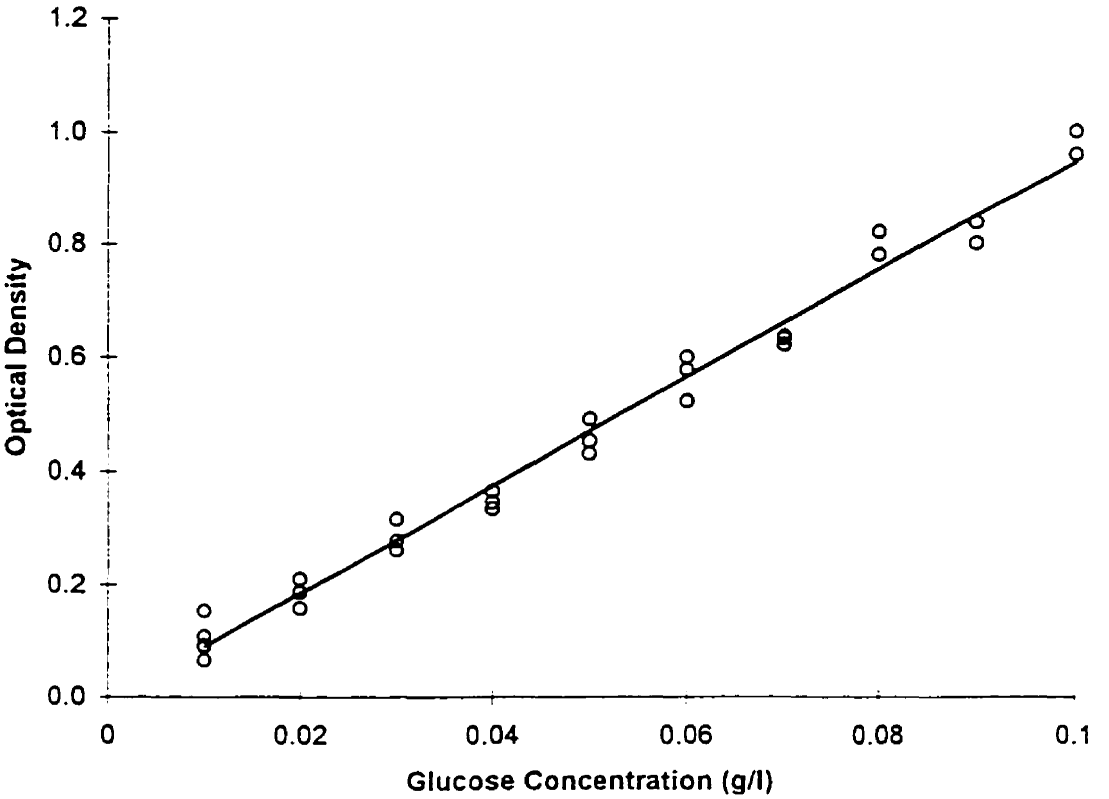


Figure 3.2 Total carbohydrate calibration curve. Glucose is used as a standard solution. Total carbohydrate readings are given as "glucose equivalents", that is the concentration of glucose required to obtain the corresponding optical density of the sample.

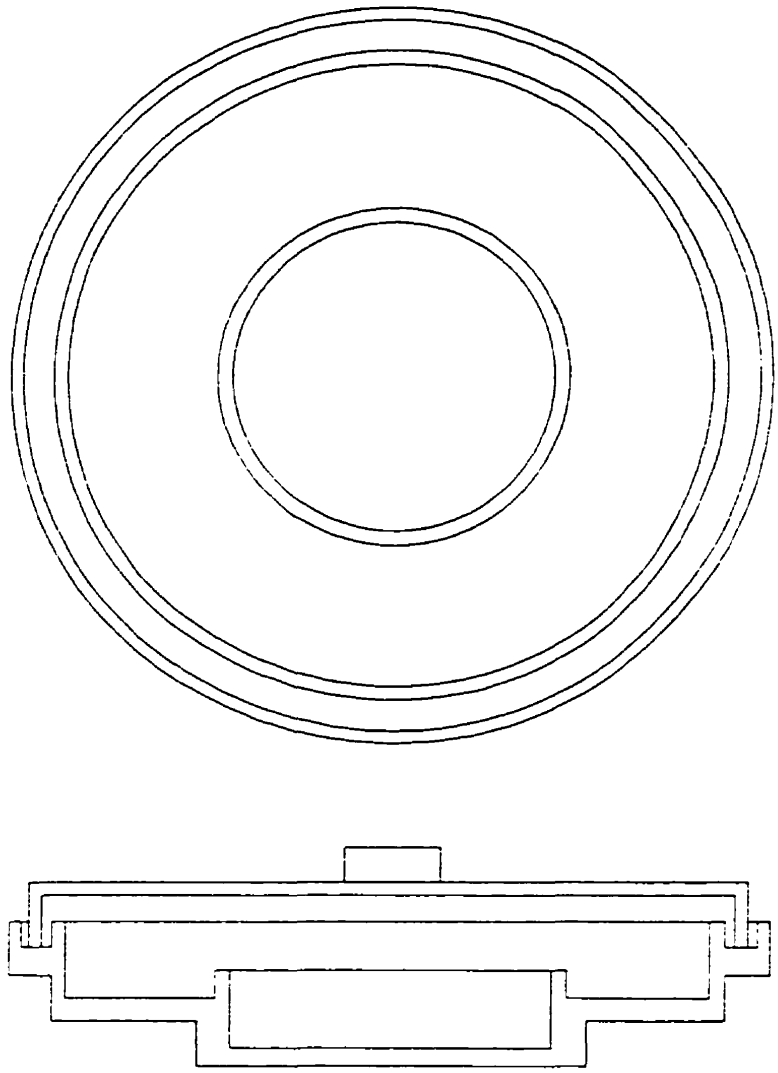


Figure 3.3 Conway Microdiffusion apparatus.

With the lid on, the gas diffuses into the inner chamber where it is neutralized by acid. The amount of acid required to titrate the inner chamber sample back to its original acidity is directly proportional to the amount of ammonium ion in the sample. The following chemicals were used:

Chemical	Contains
Acid Reagent	1% boric acid with 1% mixed indicator.
Mixed Indicator	0.033% bromcresol green and 0.066% methyl red in alcohol.
Titrant	0.02N HCl.
Alkali	saturated Na_2CO_3.

One mL of acid reagent was placed in the central chamber, and 0.5 mL of alkali was placed in the outer annular chamber. The volumes of acid and alkali need not be exact. 0.05 mL of the sample to be tested was placed in the outer chamber using a Gilson micropipette. The sample was placed in such a way as to not be in contact with the alkali. The lid was smeared with silicone vacuum grease and placed on the dish to make an air-tight seal. The dish was rocked gently to mix the sample and the alkali, making sure to avoid spilling over the dividing wall of the inner and outer chambers. The dishes were left overnight to allow the reaction go to completion. The solution in the central chamber was titrated from green back to its original pink using a Gilmont micro-syringe.

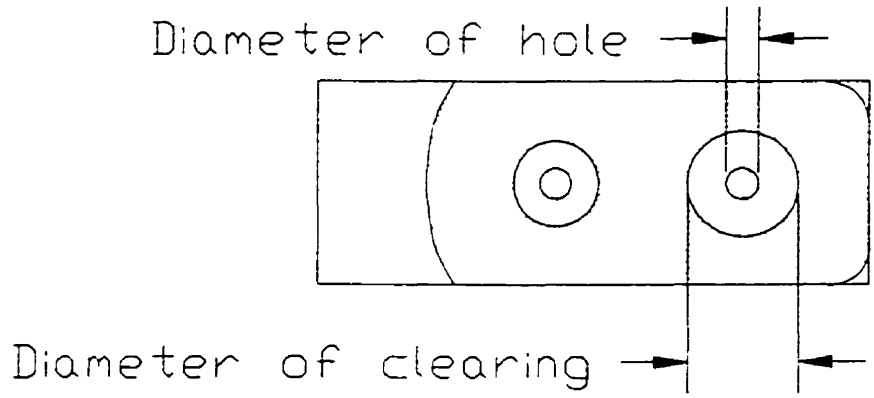
3.5 Lipase Measurement

Lipase was measured by an agar-diffusion method (18). 1ml of 50°C 1.2% Agar solution containing 0.1% tributyrin was pipetted onto a microscope slide. After allowing the agar to cool for 5 minutes, two holes were bored into the agar using a steel cork hole borer. The agar in the holes was removed using a Pasteur pipette attached to the opposite end of the cork hole borer to provide suction. The holes formed were approximately 0.18 inches in diameter. 12μL of sample was added to each hole and the slide was placed in a petri dish in a zip-lock bag, and kept in an incubator at 29°C.

The tributyrin forms an emulsion in water. The emulsion globules scatter incident light, making the agar initially cloudy. As the lipase diffused through the agar, it hydrolysed the tributyrin into fatty acids, mono and diglycerides, and glycerol. The hydrolysis products are all soluble in water, so the agar turns from translucent to transparent, see Figure 3.4. At a given time, the size of the zone of clearing is proportional to the initial lipase concentration. Three measurements of the diameter of the zone of clearing were made at approximately twelve hour intervals using calipers. The measurements were converted to an area of clearing by the following equation:

$$A = d_c^2 - d_h^2$$

Where A is Area of clearing, d_c is the diameter of clearing, and d_h is the diameter of the hole.



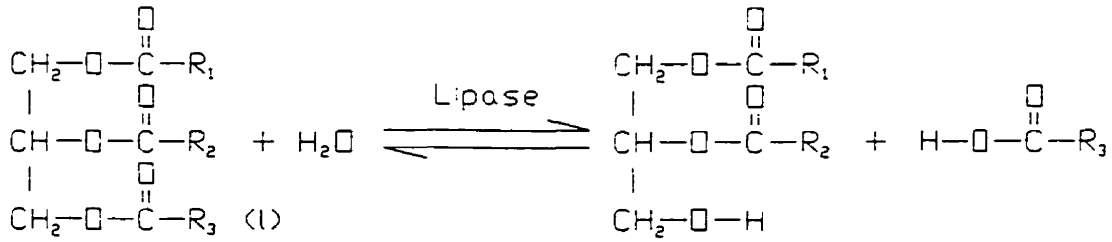


Figure 3.4 Agar diffusion assay for lipase.

These areas were plotted versus time. The area at 24 hours was interpolated from this line. This value was then converted back to a diameter by the following equation:

$$D = \sqrt{A_{24\,hrs} + d_h^2} - d_h$$

This diameter, D, was used as a measure of the lipase concentration, and from now on will be referred to as the agar clearing at 24hrs. The agar clearing at 24hrs should be directly proportional to the logarithm of the concentration of lipase. A form of calibration was made by performing lipase tests on a series of dilutions of an experimental sample from the reactor. This curve would allow the relative concentration of samples to be compared. The concentration was determined by the degree to which the sample used to make the curve would have to be diluted to give the equivalent result of a weaker sample.

3.6 Cell Disruption

Samples for cell disruption were prepared by centrifuging up to 500ml of broth. The pellets were collected and resuspended in 150ml of distilled water. This suspension was placed in a Bead Beater (BioSpec Products) with 80oz of glass beads. The cooling jacket was used with crushed ice and salt to prevent overheating the sample. The beater was operated for about one and a half minutes, and then allowed to sit for two to six minutes. This procedure was repeated seven to nine times. After the first beating, the jar was opened to allow air bubbles to escape.

4.0 RESULTS

4.1 Glucose Limitation

Most Cycles were operated under conditions where nitrogen was not limiting. The carbohydrate data for two consecutive cycles under these conditions is presented in Figure 4.1. Although residual carbohydrate was still present at the end of each cycle, the evidence suggests that glucose was limiting. Cycle 63 was extended, that is the reactor contents were withheld beyond the usual point of cycling. No further decrease in the carbohydrate level was found during this extended period. Since nitrogen was in large excess, as seen from Figure 4.2, it was suggested that the minimum in the carbohydrate trace corresponded to glucose exhaustion. Additional evidence was obtained when oil was used as the sole carbon source. In Figure 4.3 it can be seen that the carbohydrate level is similar to the minimum level shown in Figure 4.1.

It should be noted that the point of glucose exhaustion is not the point of carbon limitation when the medium contains oil. Although the oil level was never measured quantitatively, it was observed that oil remained in the medium long after glucose exhaustion.

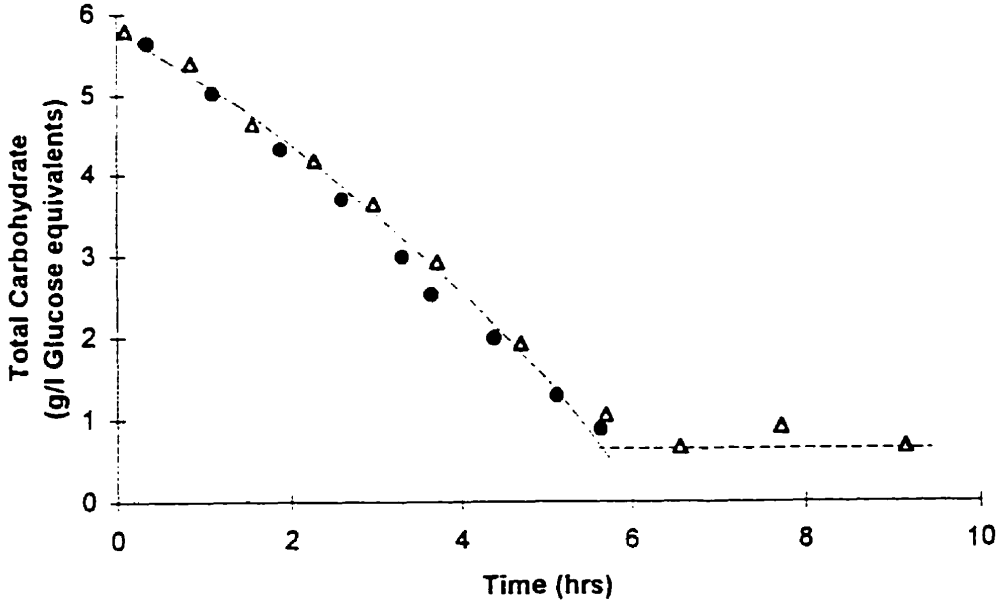


Figure 4.1 Glucose curve for cycles 62 and 63 of run 4. Glucose and oil with 2g/l ammonium sulfate. Cycle 62 (●), cycle 63 (△).

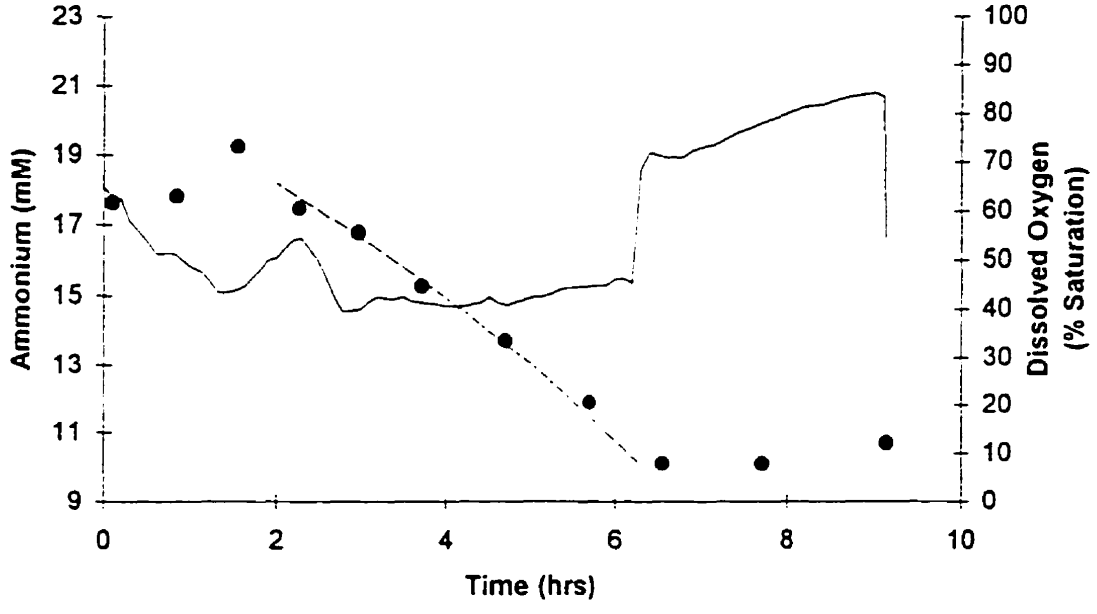


Figure 4.2 Ammonium and dissolved oxygen levels for cycle 63 of run 4. Glucose and oil, 2g/l ammonium sulfate. Ammonium (●), dissolved oxygen (-).

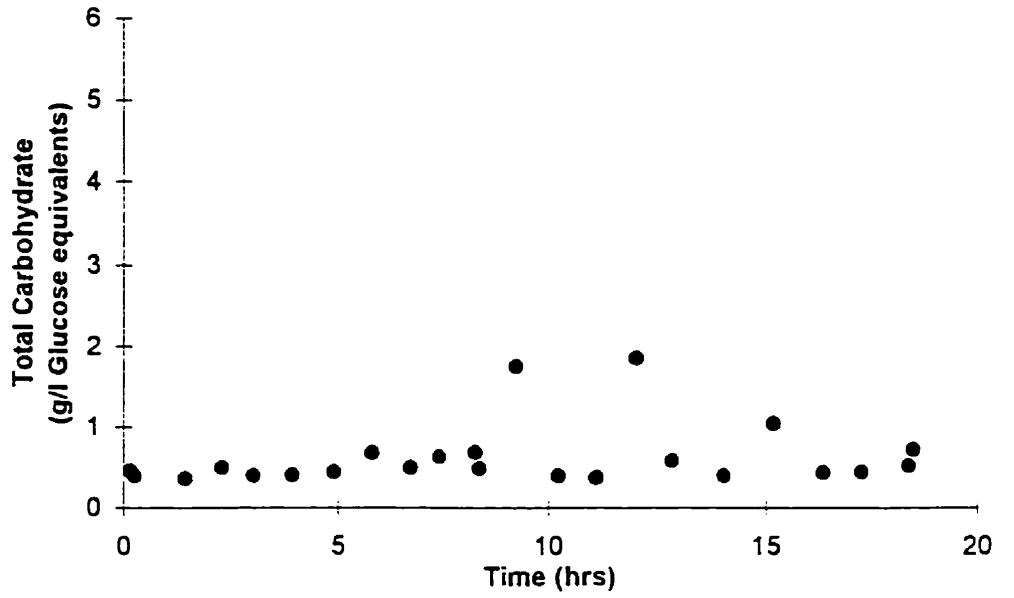


Figure 4.3 Glucose profile for cycles 33 and 34 of run 3. Oil only, 8g/l ammonium nitrate.

4.1.1 Glucose Uptake Rate

The glucose consumption rate was found to be well described by first order Monod kinetics (see Figure 4.1).

The exhaustion of glucose was found to result in the cessation of ammonium uptake, and a sharp, dramatic rise in the dissolved oxygen level, as can be seen in Figure 4.2.

4.1.2 Ammonium Uptake Rate

For the first hour or so the ammonium data was very erratic (Figures 4.2,4.4 and 4.5). After this initial period the ammonium readings became regular, demonstrating first order Monod consumption. On most of the traces, a minimum in dissolved oxygen was found to correspond with the transition between the erratic readings and the regular first order profile.

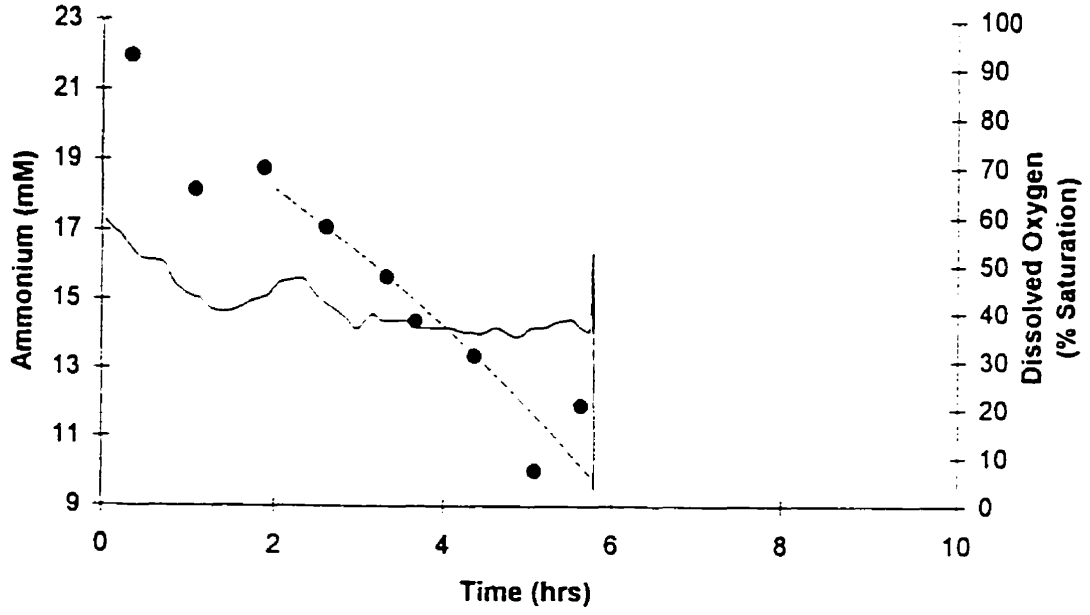


Figure 4.4 Ammonium and dissolved oxygen levels for cycle 62 of run 4. Glucose and oil, 2g/l ammonium sulfate. Ammonium (●), dissolved oxygen (—).

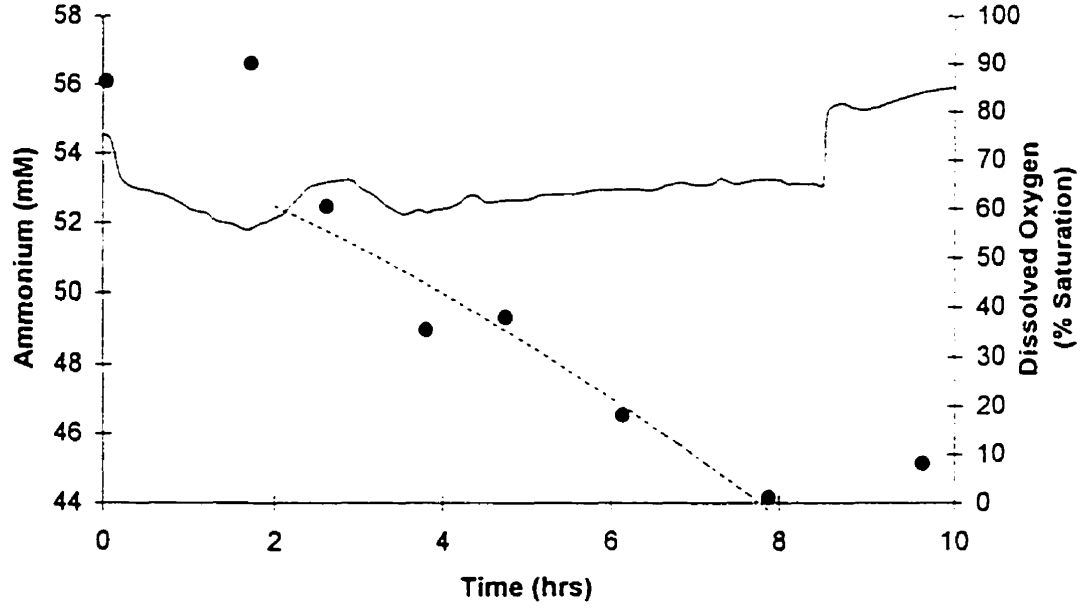


Figure 4.5 Ammonium and dissolved oxygen levels for cycle 74 of run 4. Glucose and oil, 4g/l ammonium sulfate. Ammonium (●), dissolved oxygen (—).

4.2 Ammonium Limitation

Ammonium limitation was obtained when using 1.1g/l ammonium sulfate as the nitrogen source, with glucose and oil as carbon sources.

4.2.1 Glucose Uptake Rate

Under conditions of ammonium limitation the glucose uptake profile takes on a new facet. While ammonium is present glucose consumption is first order, as in the case of glucose limitation. Upon ammonium exhaustion, the glucose level continues to drop, but at a lower rate. This rate is maintained until glucose exhaustion (Figure 4.6). The carbohydrate level is higher at the point of glucose exhaustion under these conditions than under glucose limitation.

4.2.2 Ammonium Uptake Rate

As with the glucose limiting case, the ammonium data was erratic for the first hour or so before it settled into first order consumption (Figure 4.7). Unlike the glucose limiting case, a minimum was not observed in the dissolved oxygen trace to mark the transition between the erratic and more regular readings

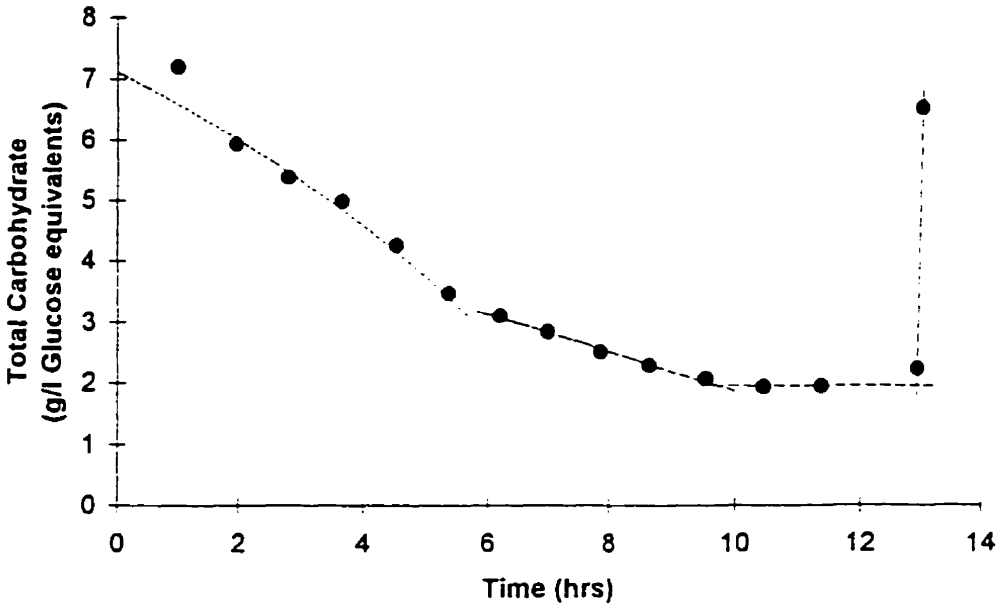


Figure 4.6 Glucose curve for cycle 43 of run 4. Glucose and oil with 1.1g/l ammonium sulfate.

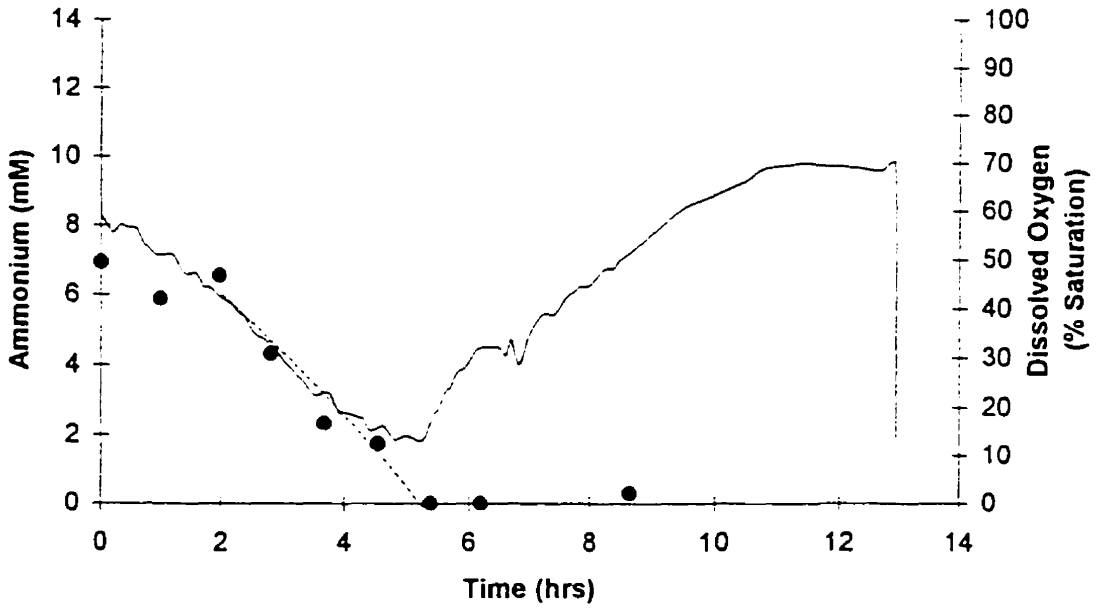


Figure 4.7 Ammonium and dissolved oxygen levels for cycle 43 of run 4. Glucose and oil, 4g/l ammonium sulfate. Ammonium (●), dissolved oxygen (—).

4.3 Dissolved Oxygen Levels

There is a noticeable difference between the dissolved oxygen traces obtained under carbon limitation (Figures 4.2,4.4 and 4.5), and that of ammonium limitation (Figure 4.7). In the former case glucose exhaustion led to a sharp rise in dissolved oxygen levels, whereas ammonium exhaustion in the latter case led to a gradual rise.

Another point of interest in the dissolved oxygen traces was that dissolved oxygen levels in all cases with glucose and oil returned to 100% saturation very slowly. Cycles with glucose and oil were also found to give more noise in the dissolved oxygen trace (Figures 4.8-4.10). Extending the cycle resulted in decreased noise, even for subsequent cycles (Figure 4.10).

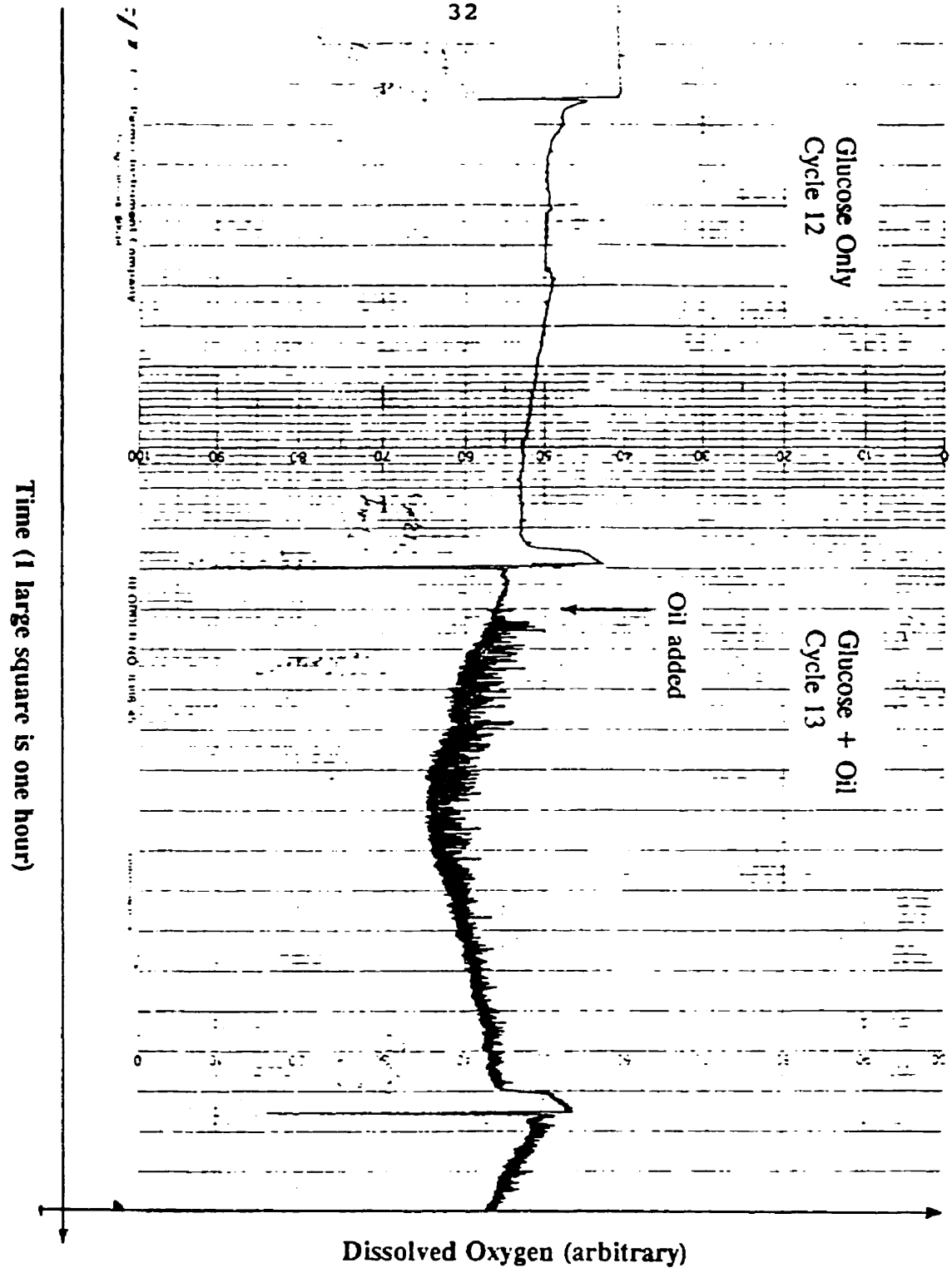


Figure 4.8 Effect on the dissolved oxygen trace of adding oil to the broth. 8 g/l Ammonium Nitrate. Cycles 12 and 13 of run 4.

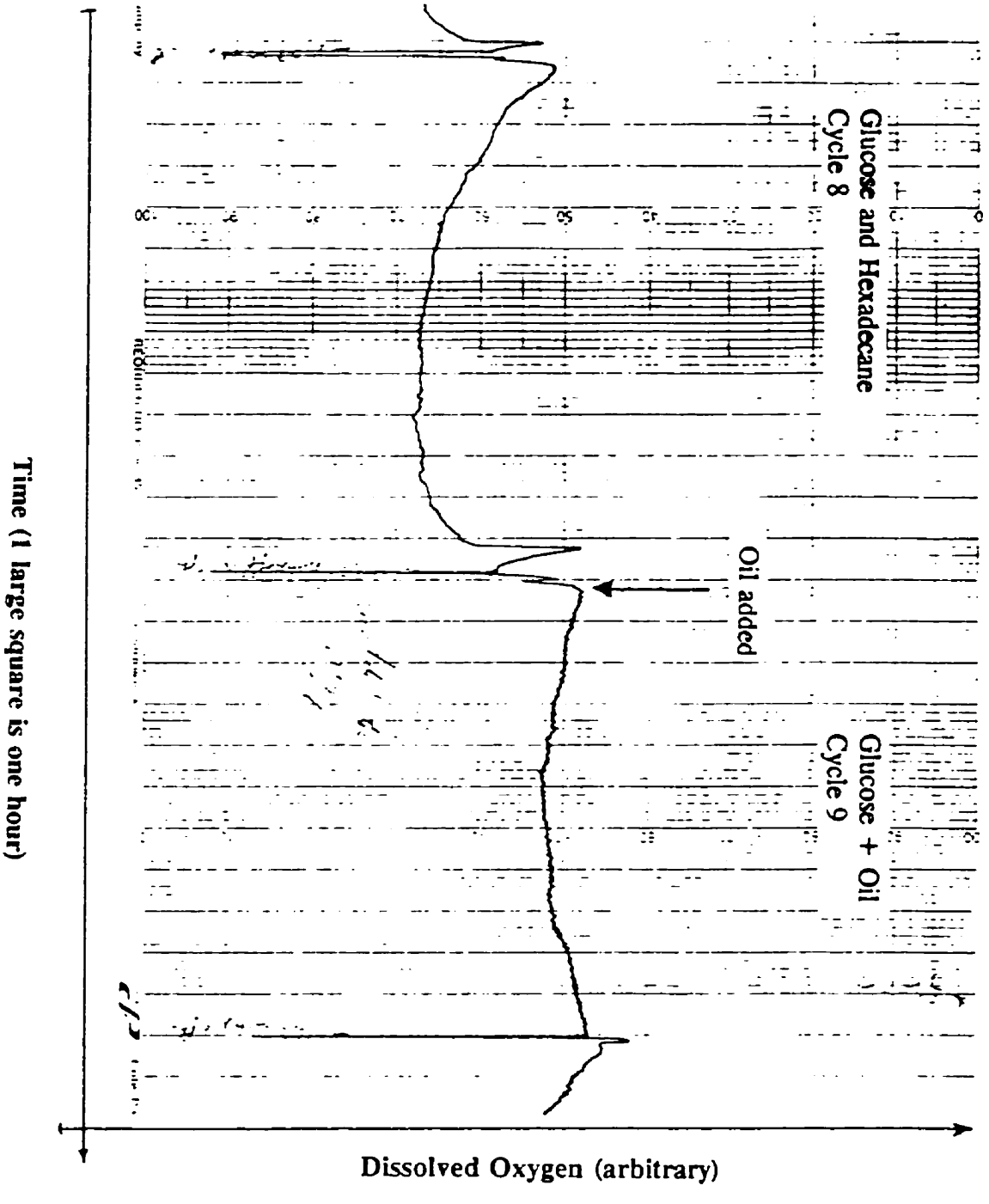


Figure 4.9 Effect on the dissolved oxygen trace of adding oil to the broth. 8 g/l Ammonium Nitrate. Cycles 8 and 9 of run 3.

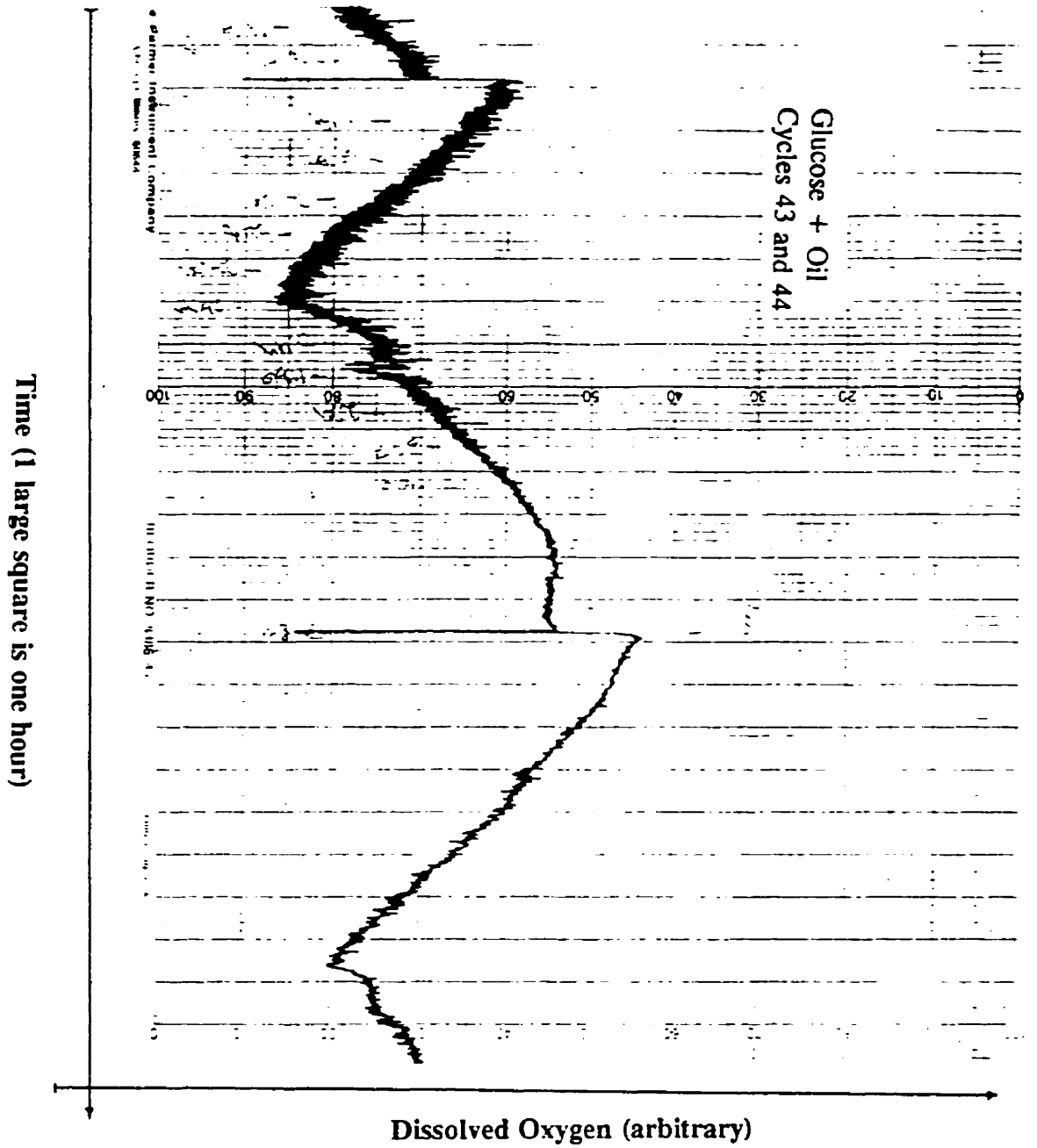


Figure 4.10 Effect on the dissolved oxygen trace of an extended cycle when oil is present in the broth. 1.1 g/l Ammonium Sulfate. Cycles 43 and 44 of run 4.

4.4 Effect of Ammonium Ion Concentration

The effect of the ammonium source on the biomass and the time to the minimum in dissolved oxygen is shown in Figure 4.11. The time to the minimum was much less when ammonium was supplied by ammonium sulfate than when supplied by ammonium nitrate. For the three levels of ammonium sulfate investigated, it seems that under nitrogen limiting conditions the time to the minimum is lowered.

The biomass level was independent of the concentration and form of ammonium supplied, even when nitrogen was limiting.

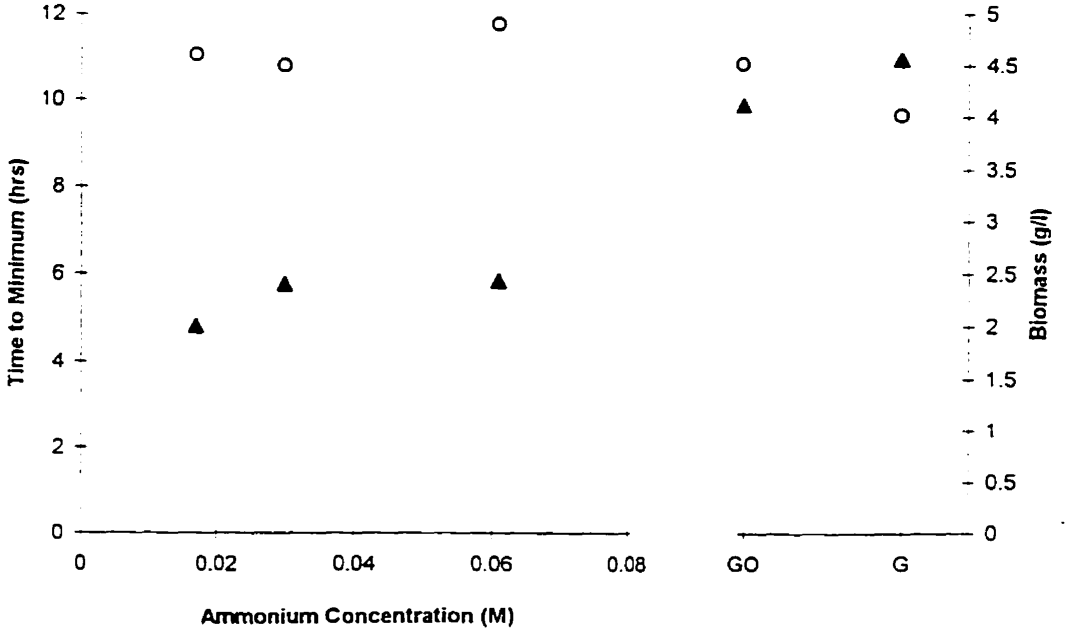


Figure 4.11 Time to minimum and biomass as a function of ammonium concentration. Time to minimum (▲), biomass (○). Note: ammonium was limiting at 0.02M. For GO and G points, ammonium nitrate was used at 0.1M. The GO data was obtained using glucose and oil, whereas the G data was obtained using glucose alone. All other points used ammonium sulfate.

4.5 Lipase Measurement

4.5.1 Sample Preparation

Samples for the lipase assay were initially prepared by centrifugation. The supernatant was retained and the pellet was resuspended in a fixed volume of distilled water. Both samples were then used in the lipase assay. Results from one shake flask experiment are shown in Figure 4.12.

Samples that had been prepared by cell disruption and subsequent centrifugation showed no clearing from samples of the supernatant, and no significant difference was found between the clearings from samples of the beaten and unbeaten pellets.

Tween 80 was added to samples in attempt to increase supernatant levels of lipase. It was found, however, that the Tween caused a certain amount of clearing itself at high concentrations. In addition, the results were not significantly improved.

Since no clearing was seen with the supernatant samples, and resuspended pellet volumes were irregular, all future samples were used without centrifugation.

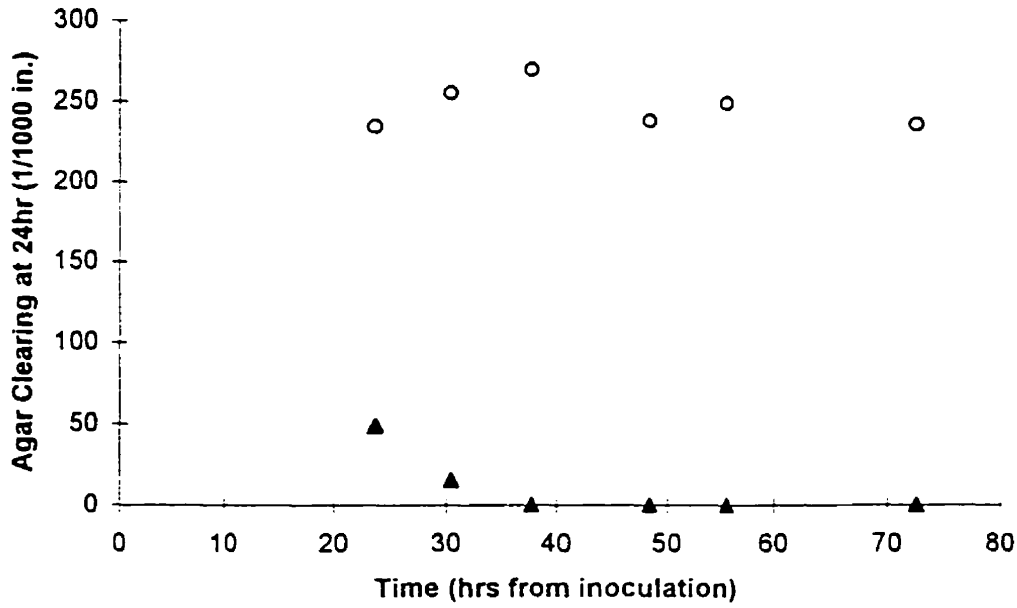


Figure 4.12 Shake flask study with lipase results from pellet and supernatant. Pellet samples (○), supernatant samples (▲).

4.5.2 Effect of Agar Substrate

Results from the lipase test are dependant on the quality of the agar substrate which can vary over time and from batch to batch. The diffusivity of agar typically varies by about 10% between one preparation and another (16). Table 4.1 shows the results for three samples on an old batch of agar and on a fresh batch of agar substrate.

Time variance of the agar was not so important over a few days, as seen in Table 4.2 which shows four samples (which were stored frozen) being applied to the same agar over three days. However, it is likely that as the agar ages beyond five to seven days (being stored at 50°C), it does begin to degrade. Qualitatively, as the agar aged it was noticed that it did not gell as well; when solidified, the agar was less firm and less smooth than fresh agar. This time variance was considered acceptable for this study since the production profiles were of greater interest than the absolute lipase levels attained.

Table 4.1 Effect of agar batch on lipase test results

	Agar Clearing at 24hr (1/1000 in.)	
Sample	Agar #1 (old)	Agar #2 (fresh)
26	243 250	196 203
27	267	209
28	264 262	211
29	228	184 186

Table 4.2 Effect of agar age on lipase test results

	Agar Clearing at 24hr (1/1000 in.)		
Sample	1 Day†	2 Day	3 Day
GO16	151 156	152 155	159 159
GO16/3‡	118	111	116
84	111 117	123 122	123 122
84/3‡	73	74	68

†Age of agar (stored at 50°C)
‡These are 1/3 concentrations of the respective samples

4.5.3 Lipase Results

The results from the first experiments, Figures 4.13-14, were surprising in that they showed a constant level of lipase, even after cycling when the concentration should immediately be halved. When the calibration curve was obtained (Figure 4.15), the explanation was found. The 3/4, 2/3, and 1/2 dilutions gave roughly the same amount of clearing as the undiluted sample. Only once the sample had been diluted to less than one half was the expected relationship obtained. Re-examining samples stored in the freezer at one third of their initial concentration gave the results shown in Figures 4.16-17. In these figures a relative concentration of lipase was obtained by applying the agar clearings to a calibration curve done at the same time as the samples. For these calculations it was assumed that there was no plateau effect. This assumption was justified since all samples were found to be below the plateau level.

Note that relative lipase concentrations are not necessarily quantifiably relatable between different figures.

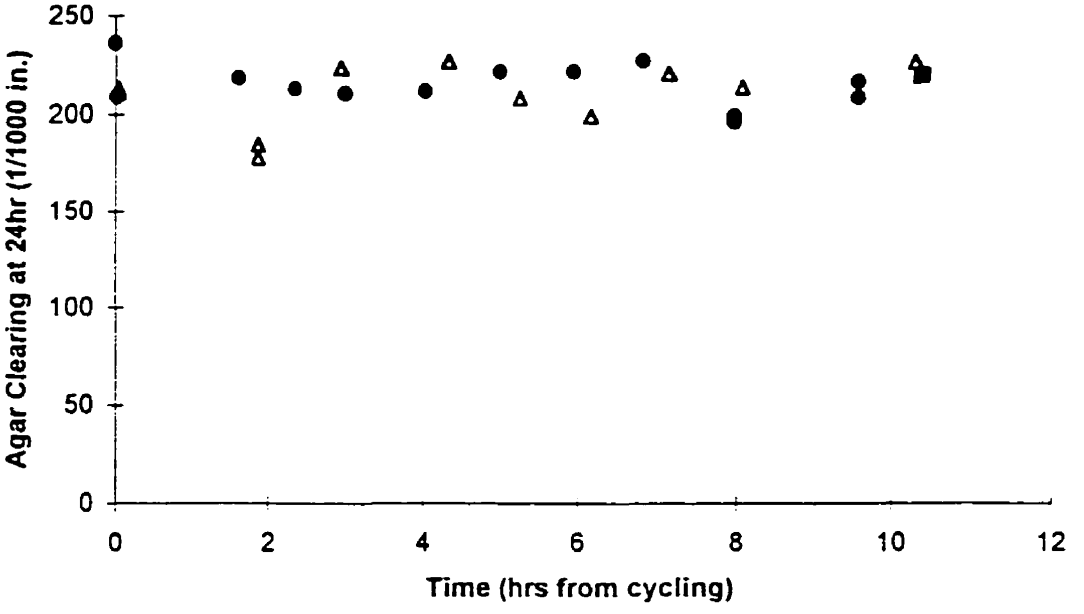


Figure 4.13 Fresh lipase results from cycles 20-22 of run 3. Glucose and oil, 8g/l ammonium nitrate. Cycle 20 (●), cycle 21 (△), cycle 22 (■).

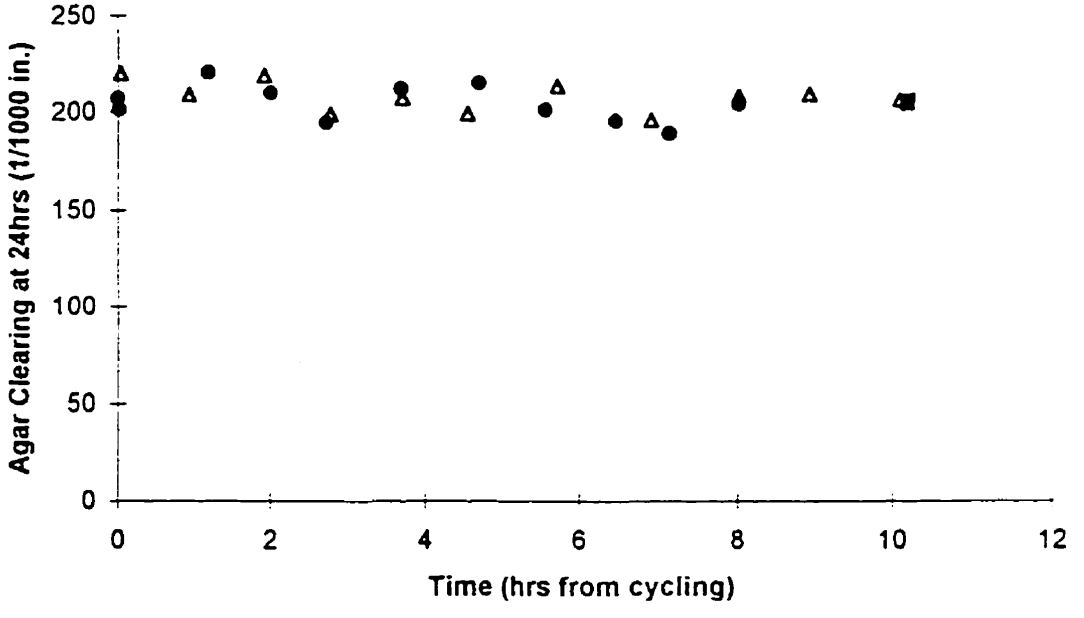


Figure 4.14 Fresh lipase results for cycles 33-35 of run 3. Oil only, 8g/l ammonium nitrate. Cycle 33 (●), cycle 34 (△), cycle 35 (■).

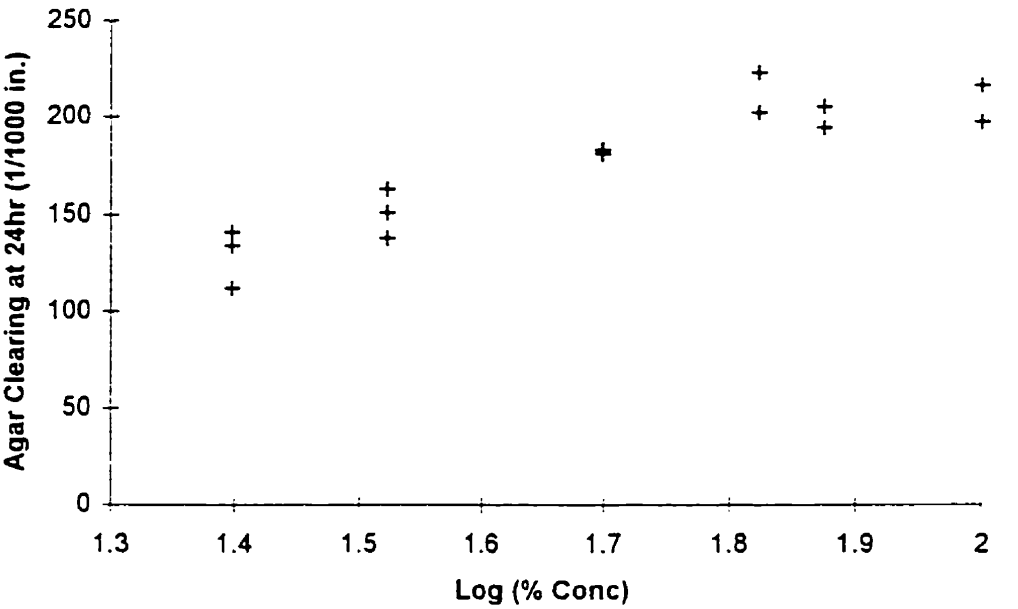


Figure 4.15 Lipase 'calibration' curve.

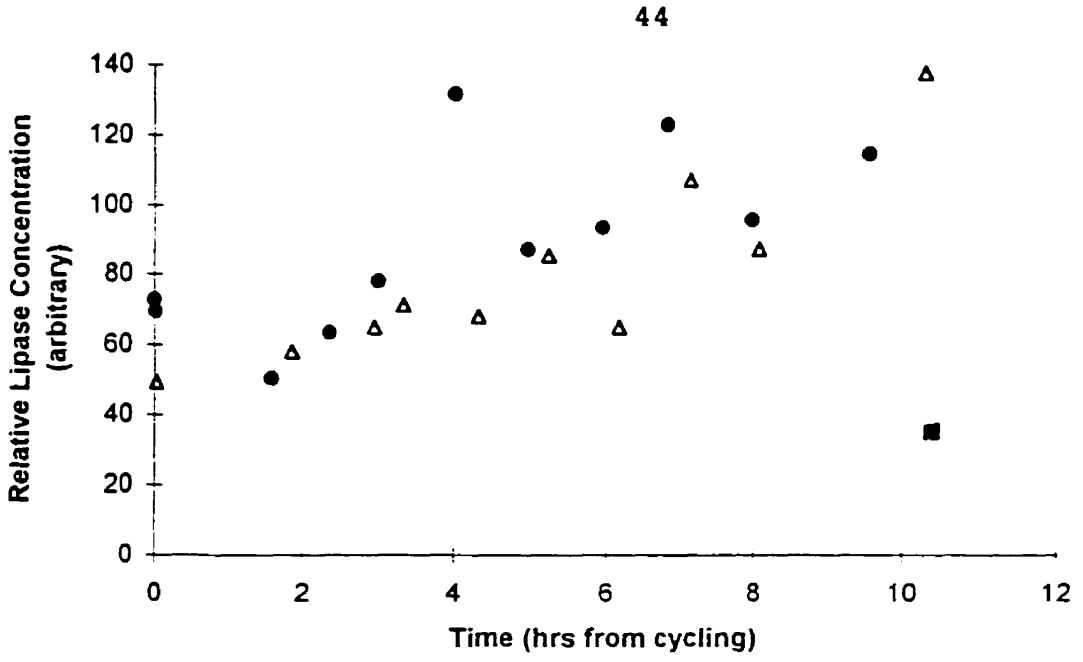


Figure 4.16 Frozen 1/3 concentration lipase results for cycles 20-22 of run 3. Glucose and oil, 8g/l ammonium nitrate. Cycle 20 (●), cycle 21 (△), cycle 22 (■).

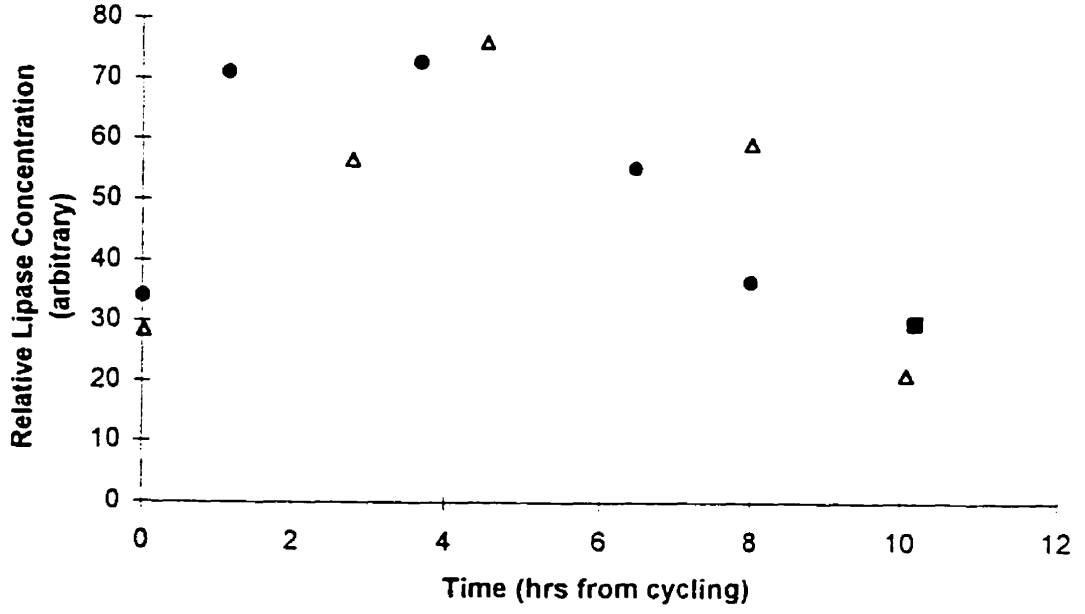


Figure 4.17 Frozen 1/3 concentration lipase results for cycles 33-35 of run 3. Oil only, 8g/l ammonium nitrate. Cycle 33 (●), cycle 34 (△), cycle 35 (■).

4.5.4 Effect of Freezing Samples

The effect of freezing samples on the results of the lipase test was examined by comparing the results from fresh samples to those of the same sample a few days later. Figure 4.18 shows that, typically, a one third dilution of a frozen sample gave only 60 to 80% of the agar clearing of an equal dilution of the fresh sample. This data was necessarily obtained on different sets of agar, but the difference in results shown here is far greater than that expected for variance among agar batches.

Where glucose was the only carbon source, that is no oil was used, all lipase activity was lost upon freezing.

The act of freezing the samples also seemed to improve the spread and quality of the data, as can be seen in Figures 4.19 and 4.20.

4.5.5 Effect of Ammonium Source

The lipase levels produced under four different conditions were compared. The test was performed on frozen samples so as to be able to use a single batch of agar. These results are presented in Figure 4.21. Despite the large scatter in the data, it appears that lipase levels were higher when ammonium nitrate was used.

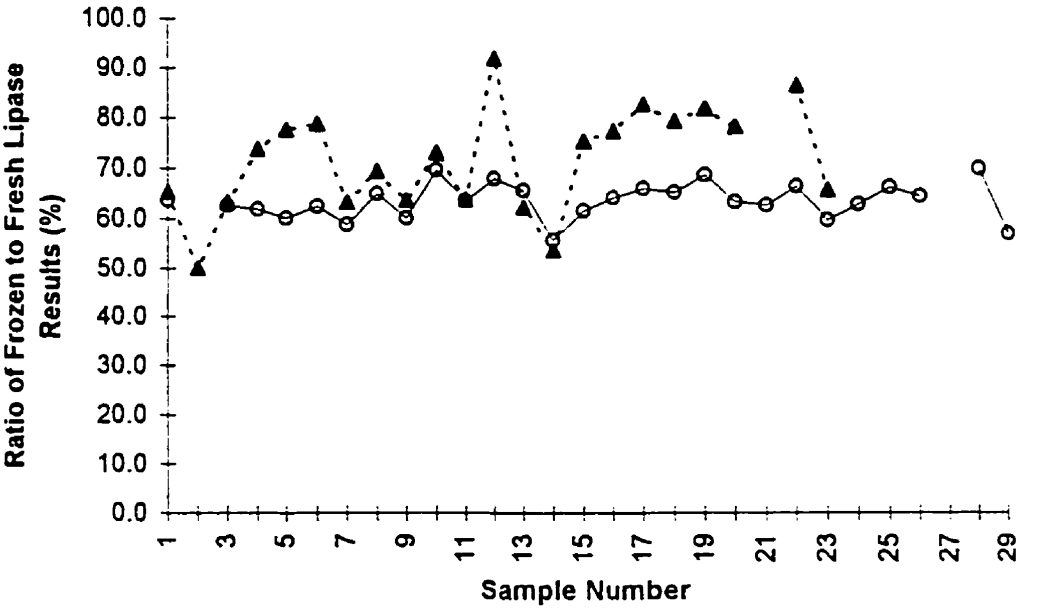


Figure 4.18 Effect of freezing on lipase samples. Data from glucose and oil cycles with 8g/l ammonium nitrate (▲), 1.1g/l ammonium sulfate (○). Run 4.

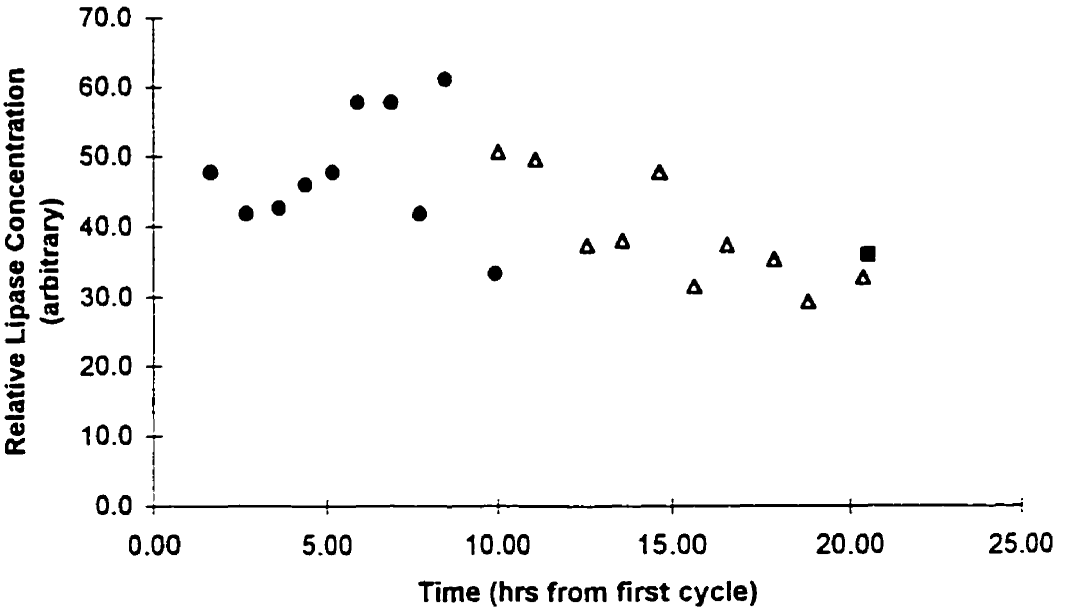


Figure 4.19 Fresh 1/3 concentration lipase results from cycles 23-25 of run 4. Glucose and oil, 8g/l ammonium nitrate. Cycle 23 (●), cycle 24 (△), cycle 25 (■).

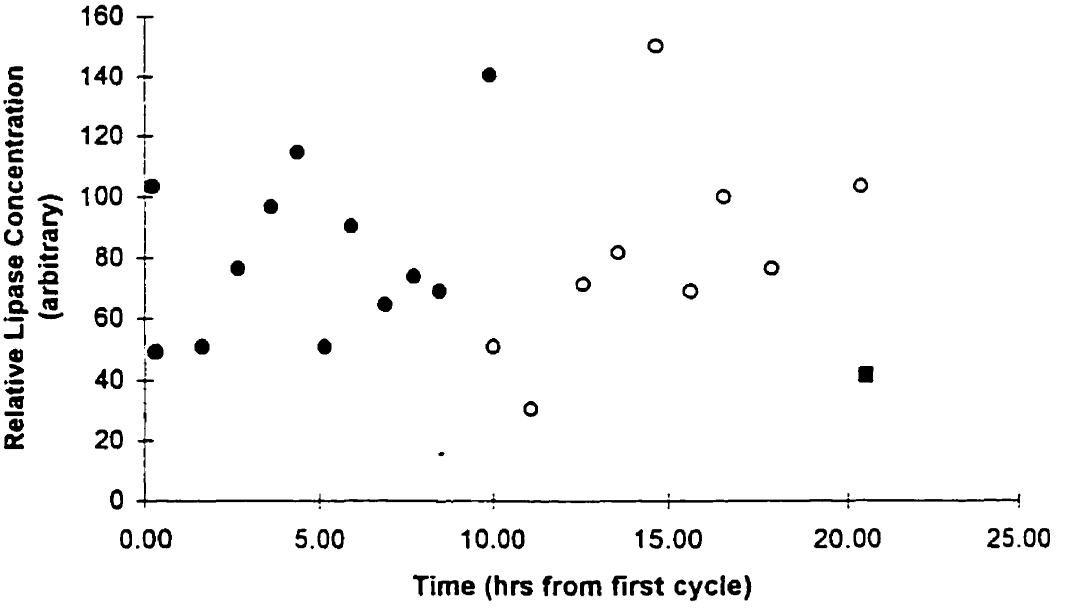


Figure 4.20 Frozen 1/3 concentration lipase results from cycles 23-25 of run 4. Glucose and oil, 8g/l ammonium nitrate. Cycle 23 (●), cycle 24 (△), cycle 25 (■).

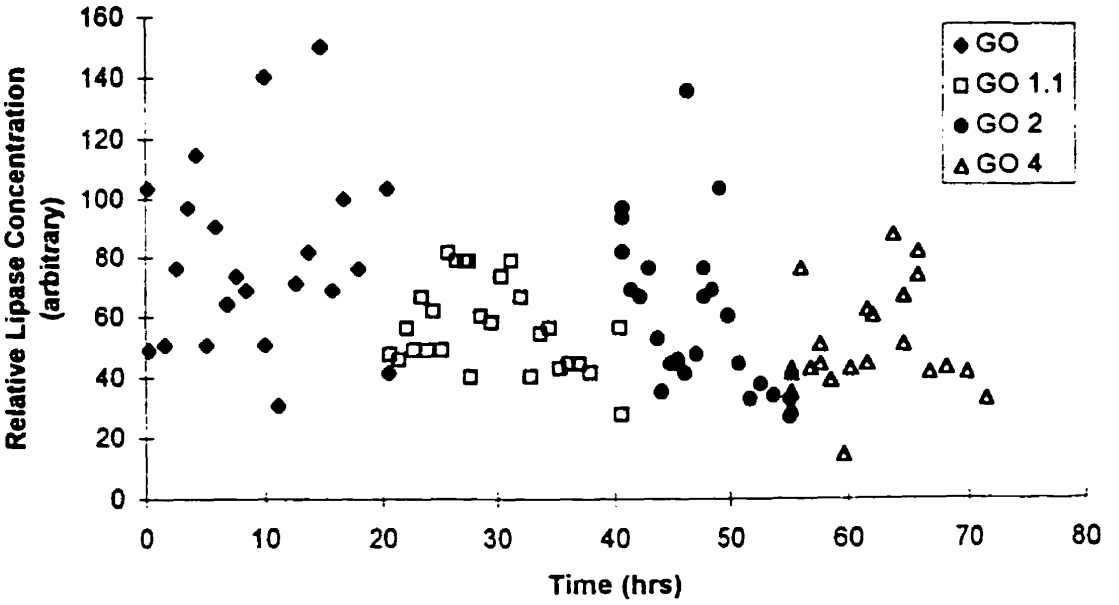


Figure 4.21 Comparison of lipase production under different conditions. All conditions have glucose and oil as carbon sources, and are from run 4.

GO - 8 g/l ammonium nitrate; cycles 23 and 24.
GO 1.1 - 1.1 g/l ammonium sulfate; cycles 42 and 43.
GO 2 - 2 g/l ammonium sulfate; cycles 62 and 63.
GO 4 - 4 g/l ammonium sulfate; cycles 73 and 74.

5.0 DISCUSSION

5.1 Use of a Pressure Transducer for Level Control

The pressure transducer was found to have certain advantages over the electronic balance as a device for level control. The level of broth in the reactor could not be measured by weight since coupling the reactor with the balance would have been excessively difficult. Fresh medium and broth, therefore, had to be measured in separate dosing and harvesting vessels. Installing the pressure transducer on the reactor greatly simplified the apparatus since the dosing and harvesting vessels were no longer required. This reduction in complexity allowed the system to be set up quicker and easier. One of the routes by which contamination can occur is when the reactor is being set up after sterilization. Being able to set up quicker, with less chance for accidents, reduces the chance of initial contamination through accidental exposure.

The pressure transducer setup would also be easier to use for larger scale setups. Use of a balance would be particularly unsuitable for larger scale situations since the large dosing and harvesting vessels would add to the expense, as well as floor space requirements. Installing a very large balance would also be difficult and expensive.

Even at the bench scale, the simplification to the setup, and the lower cost of the pressure transducer may make the self-cycling fermenter a more attractive option for other researchers.

Although it was necessary to turn the pump off during cycling, this did not usually create problems. The settling of biomass and the separation of insoluble substrates was a concern during cycling. The broth, however, was found to maintain a high degree of homogeneity. It is important to harvest homogeneous samples of broth during cycling in order to be able to obtain repeatable data, which is one of the major benefits of SCF.

The only significant problem encountered due to shutting the pump off, was foaming. Certain medium conditions could lead to a large amount of foaming. Without the cyclonic falling film breaking the foam, the air inlet caused an accelerated build up. This increased foam not only made visual confirmation of the level control impossible, the inlet air was found to blow the foam up into the exhaust port. Although only a small portion of broth was lost in this manner ($\sim 0.5\%$), it could act as a bridge for contamination across the air filter, resulting in back contamination. It should be noted, however, that no contamination was recorded by this route. Despite these problems, the foaming did not interfere with the cycling operation itself. It was still possible to remove and add accurate volumes of medium.

5.1.1 Adjustments to the Physical Setup of the Pressure Transducer

To measure the liquid level accurately, the pressure transducer had to be quite sensitive. Small shocks to the transducer, such as being bumped, caused spikes in the signal. Moreover, such shocks led to a shift in the base line signal as the transducer

physically settled into a different position. This problem was solved by using a clip to affix the transducer to the wall that the reactor was mounted on.

Initially an impermeable membrane was used to separate the reactor contents from the pressure transducer. With this arrangement, significant drift of the base line transducer signal was observed, even shortly after installation. It was postulated that the problem could be one of pressure equalization. Since the membrane had only limited mobility, leaks on the pressure transducer side resulted in a decreased ability of the membrane to equalize the pressures. Tightening the seals on the transducer side improved results, but drift still occurred. Changing to a 0.45μ Millipore filter was found to essentially eliminate the drift. This reduction was attributed to two effects. Most likely this permeable membrane allowed a transfer of mass to equilibrate pressure, instead of relying solely on membrane flex. Another possibility is that the Millipore filter was simply more flexible than the rubber membrane previously used.

It was also found to be important to install the pressure tap at a downward angle from the reactor (see Figure 3.1). With the pressure tap aimed upwards, entrained air bubbles in the broth collected in the tap. Under these conditions level control could not be obtained. A possible explanation is that a small leak that was noticed in the membrane housing, while insignificant when broth was present, became more significant when air was present. Another possible explanation is that surface tension between the air bubbles and the pressure tap wall was decreasing the pressure transmitted through the tap.

5.1.2 Pressure Transducer Signal Enhancement

Initially, the signal from the pressure transducer was too noisy to measure adequate dosing and harvesting quantities. To improve the signal, the number of analog to digital conversions was increased by an order of magnitude. The number of conversions is the number of times that the data acquisition board reads the analog transducer signal before averaging the readings and giving a representative digital signal. The extra time required to perform the conversions led to a new problem. During the period of measurement, the signal was undergoing significant change. Overshoot was inevitable due to the averaging of the changing signal. In addition, the amount of overshoot varied depending on how much time had passed between when the desired level had been reached and the signal for that level reached the computer (see Figure 5.1). Although the dosing and harvesting rates could have been lowered, resulting in less overshoot, it was desirable to keep the time required to cycle at a minimum. While the reactor is cycling, the organisms are typically in a period of nutrient starvation. This period of nutrient starvation can affect the future growth and production of the organism. One of the benefits of SCF is being able to keep the organisms growing at the maximum specific growth rate, and to be able to avoid nutrient starvation.

A new approach was required to handle the noisy signal without causing overshoot. To start with, the number of conversions was returned to the original lower value. Although the rate of level change during harvesting changed from one cycle to the next, it was found that after the first few seconds during harvesting, the

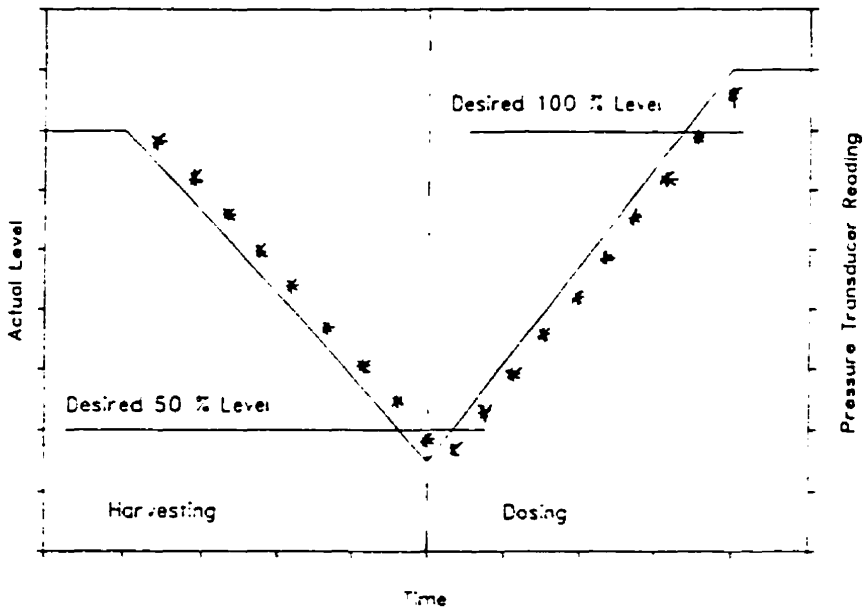


Figure 5.1 Cycling using pressure transducer.
Pressure transducer readings (*).

transducer signal decreased at a constant rate until the desired level was attained. The same effect was found for dosing. A simple solution to the noisy signal was to perform a linear regression on the data. In this way, the predicted value at the current point in the harvesting or dosing period was an improved measurement of liquid level. This is similar to performing many conversions on the analog signal, except in this modified approach the change in the signal with respect to time is taken into account.

In the previous setup a "fudge factor" was required to take into account the extra volume that was removed or added between when the computer determined that the desired level had been obtained and when the control valve was closed (70). With the best line method, sampling was stopped once the level got within a certain range (chosen by the operator) of the target value. The control valve was then closed after the predicted time to reach the target level had elapsed. This method left no delay between detection of the level and valve closure, thus no "fudge factor" was required. This method also prevents premature completion of harvesting or dosing caused by a stray signal. This simple cycling scheme could also be applied to level control with the balance.

After numerous cycles, the readings would begin to drift due to biomass accumulation in the pressure tap. At this point, the only solution was found to be the replacement of the clogged pressure tap with a new sterilized one. Both times this operation was performed, no contamination occurred. If lower biomass levels were used, it is expected that this problem would have occurred less frequently. On an

industrial scale a membrane could be mounted flush with the reactor, eliminating the problem of biomass accumulation.

5.2 Dissolved Oxygen Signal Noise

As observed with the data reported by McCaffrey (20), certain conditions gave considerable noise in the dissolved oxygen (DO) probe signal. This noise occasionally led to premature cycling, and the low sensitivity required to prevent premature cycling resulted in cycles lasting a significant amount of time beyond the minimum in the DO trace. The noise was not present when glucose was the sole carbon source. As soon as vegetable oil was added to the system, noise was evident in the oxygen trace (Figure 4.8 and 4.9). When cycles were extended, the noise level died down, and remained low even in the following cycle when fresh oil was added (Figure 4.10). McCaffrey suggested that a microbial product is accumulating on the DO probe membrane. The results shown here seem to indicate that the oil in the broth interacts with the cells, or products, causing adhesion to the probe membrane. A product formed later in the cycle could have reversed this condition, causing the build-up on the probe to slough off, and remain off even when the fresh oil was added.

When glucose and oil were used, the dissolved oxygen trace was found to return to 100% saturation levels very slowly after the minimum. Biosurfactants have been found to affect the oxygen transfer rate in the cyclone reactor (27), but this cannot account for what was observed here. If the oxygen transfer rate was lowered,

the initial increase in dissolved oxygen upon nutrient exhaustion would also be lower. The period between the initial quick increase, and the slower rise may be related to the production of some compound that does not require the limiting nutrient.

5.3 Total Carbohydrate Analysis

The total carbohydrate analysis consistently demonstrated a residual level in the broth (Figures 4.1, 4.3 and 4.6). Since in at least one case, no carbohydrates were present in the medium, this residual level must be due to a metabolite produced by the microorganism. *C.bombicola* is known to produce a sophorolipid (9,11,21,34). The sophorose portion may react in the total carbohydrate analysis, giving the observed residual levels. The levels observed reflect the expected trend, being much higher under conditions of nitrogen limitation (Figure 4.6). Indeed, the change in the glucose consumption profile upon ammonium exhaustion, shown in Figure 4.6, may indicate the switch from biomass production to sophorolipid production, but this was not confirmed.

5.4 Nutrient Utilisation

It was observed that the ammonium measurements were erratic for the first hour and half or so of the cycles. This effect is most likely due to the presence of yeast extract in the medium. The yeast extract supplies organic nitrogen (22) that is probably used by the organism before the ammonium. Note after the initial period of scatter, the ammonium readings demonstrated first order consumption consistent with

the Monod model. The point of transition was usually found to correspond with a local minimum in the dissolved oxygen trace (Figures 4.2, 4.4 and 4.5). McCaffrey postulated that a similar early minimum corresponded to a diauxic effect (20). That means that the organism was changing over from one nutrient source to another, resulting in a decrease in respiration while enzymes for metabolizing the second source were made. The results shown here suggest that the organism was switching from some nitrogen source in the yeast extract to ammonium. Curiously, however, the decrease in respiration is not reflected in a change in glucose consumption rate. A satisfactory explanation for this behaviour has not been found.

Operating with ammonium nitrate gave much slower metabolism than when using ammonium sulfate (Figure 4.11). It has been reported that nitrate can have a detrimental affect on ammonium uptake, thus slowing metabolism (25). However, the robustness of the self-cycling fermentation was illustrated once again by its ability to handle this change in substrate. If a chemostat was operated based on ammonium sulfate metabolic rates, a switch to ammonium nitrate could result in cell washout because of the reduction in growth rate. The self-cycling fermenter would automatically handle the situation by allowing a longer cycle time, as was seen in this study (Figure 4.11).

5.5 Comparison of SCF and Batch

The advantages of SCF are best demonstrated by comparing with traditional methods. Data from a fed-batch and two batch experiments by other authors is presented in Figure 5.2 and 5.3. Although the media used are somewhat different for these fermentations compared to the media presented here (Table 5.1), some important points can still be made.

Perhaps the most striking point is that these researchers report very few results from the exponential growth phase. It is difficult to determine growth and growth associated characteristics with so little data. This is a common problem when examining the results of batch fermentations.

Another point that can be seen from Figures 5.2 and 5.3 is that the inocula are considerably different from one experiment to another. Sometimes an inoculum is in its exponential phase and can begin growing quickly. Other times it can be well beyond the exponential growth phase, and will only start growing again after a significant lag period. This partly explains why the researchers did not know when the exponential period had started, since they did not know the quality of their inoculum.

The large inocula in SCF results in a large productivity gain. For instance, for a given set of conditions, using a 50% inoculum gives almost twice as much harvestable biomass (total weight per unit time) as when using a 12% inoculum.

Table 5.1 Summary of other authors media

Authors & Type of Fermentation	Medium Component	Concentration (g/l)
	Glucose	100
Zhou and Kosaric (34) †	Vegetable oil	105
	Yeast Extract	4
Batch, 30°C	Urea	1
	$FeCl_3$	0.1
Cooper and Paddock (9) ‡	Glucose	100
	Vegetable Oil	95
Batch, 30°C	Yeast Extract	5
	Initial Conditions:	
	Glucose	100
Davila *et al.* (11)	Rapeseed ethyl esters	100
	Ammonium sulfate	4
Fed-Batch, 25°C	Cornsteep liquor	5
	KH_2PO_4	1
	$MgSO_4 \cdot 7H_2O$	0.5
	Feed:	Feed Rate
	Glucose	100 g additions
	Rapeseed ethyl esters	3.5 ml/hr

†Otherwise same basal medium as reported here, except without $CaCl_2$.
‡Otherwise same basal medium as reported here.

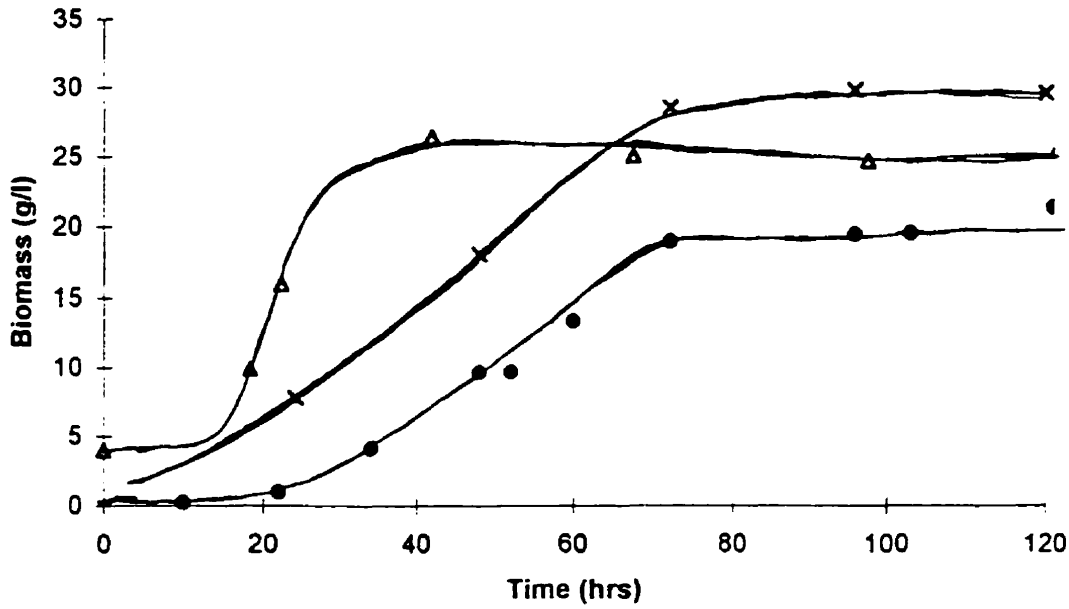


Figure 5.2 *C.bombicola* growth curves. Davila *et al.* (11) (△), Zhou and Kosaric (34) (x), Cooper and Paddock (9) (●).

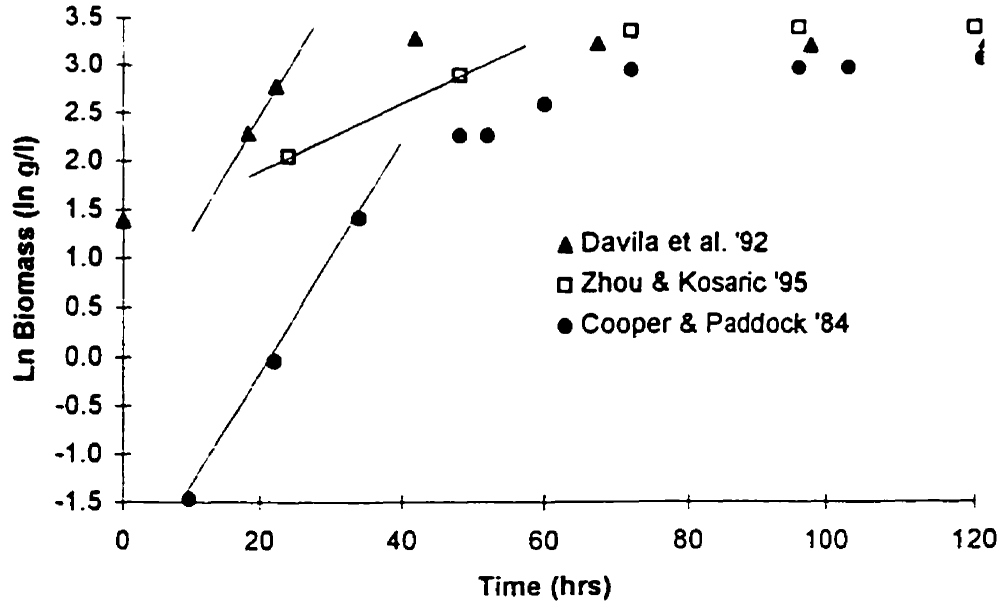


Figure 5.3 *C.bombicola* growth curves. Davila *et al.* (11) (▲), Zhou and Kosaric (34) (□), Cooper and Paddock (9) (●).

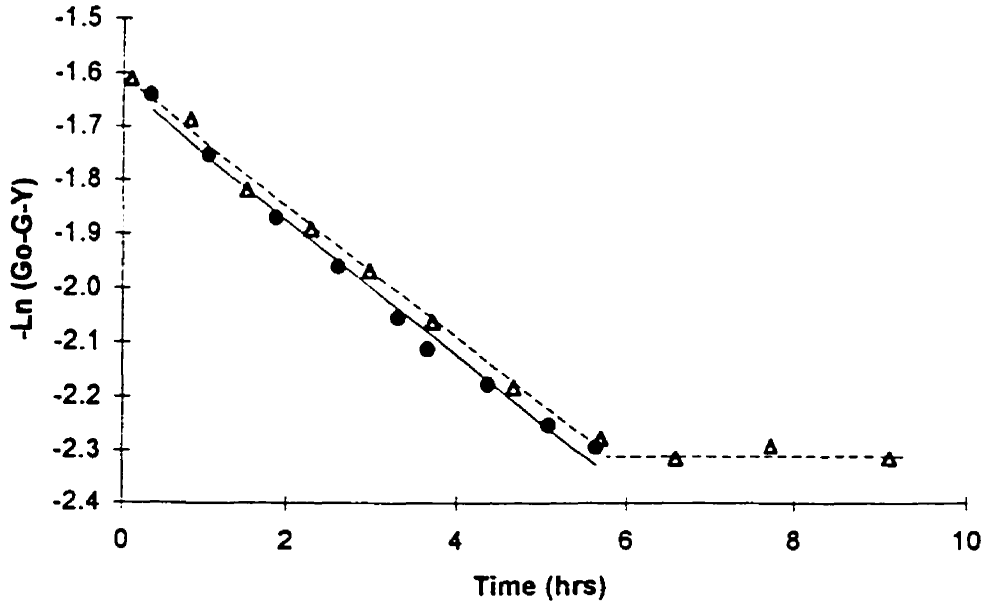


Figure 5.4 Consumption of glucose by *C.bombicola* in a SCF. Cycles 62 and 63 of run 4. Cycle 62 (●), cycle 63 (△).

The intracycle biomass levels were not measured in these particular experiments. Optical density could not be applied to samples with oil, and dry weight would have used too large a sample for intracycle measurements without significantly altering the growth conditions. The consumption of glucose, however, should reflect the production of biomass. A first order Monod plot of the glucose data from Figure 4.1 is given in Figure 5.4. Using this for comparison, it can be seen how much easier it is to work with SCF. Unlike the data in Figure 5.3, all the data points in Figure 5.4 are concentrated in the exponential growth phase. Since exponential growth is maintained throughout the cycle, the maximum specific growth rate can be obtained simply from the time between the initiation of a cycle and the minimum in dissolved oxygen, as shown in the introduction. The extended cycle pattern of cycle 63 demonstrates that studies of the stationary phase can also be made with this apparatus. These attributes of SCF allow the researcher to obtain more relevant growth related data, easier and faster.

Another important difference between the SCF and the batch and fed-batch fermentation data is that the lag and slow-down phases have been eliminated with SCF. In fact, careful comparison of the time scales indicated that the SCF had completed a full cycle before the lag phases visible in Figure 5.3 were completed. A summary of the specific growth rates obtained from Figures 5.3 and 5.4 is given in Figure 5.5. This data indicates that the organism's maximum specific growth rates are comparable for all methods of fermentation. Where SCF gains in productivity is by allowing the organism to grow at its maximum specific growth rate all the time.

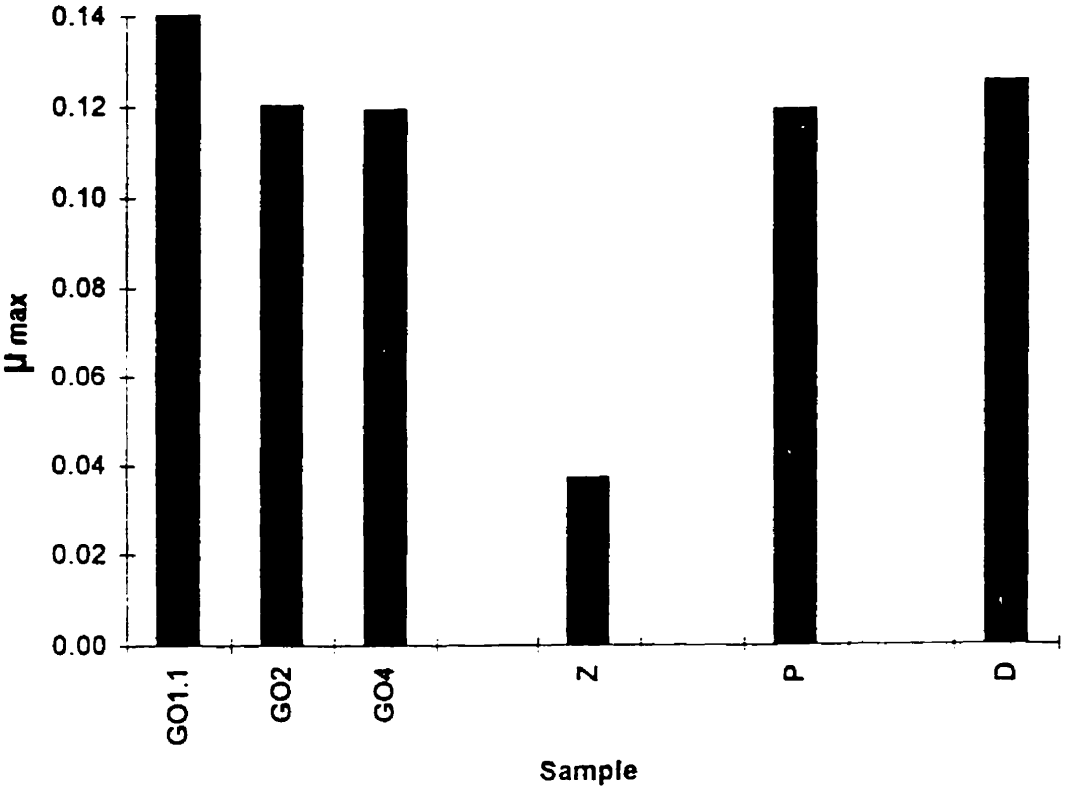


Figure 5.5 Comparison of maximum specific growth rates.
GO1.1: glucose and oil with 1.1g/l ammonium sulfate.
GO2: glucose and oil with 2g/l ammonium sulfate.
GO4: glucose and oil with 4g/l ammonium sulfate.
Z: Zhou and Kosaric (34).
P: Cooper and Paddock (9).
D: Davila *et al.* (11).

5.6 Lipase Assay

The agar diffusion assay developed by Lawrence *et al.* (18) was found to be applicable as a method to quantify the lipase produced by *C.bombicola*. The properties of the lipase from *C.bombicola* have not been reported previously. This lipase was found to be associated with the cell. Supernatants of centrifuged samples gave little or no clearing, even after cell disruption. This association did not involve covalent bonds, however. Assays of samples containing whole cells did result in clearing around the wells in the agar. Since the whole cells cannot diffuse through the agar, the lipase must disassociate from the cell under the conditions of the test. The lipase is preferentially bound to the cell wall in the initial samples. However, when the broth samples contact the agar surface, the lipase preferentially diffuses into the agar.

In other systems, some authors have found that by adding surfactants to samples of broth, cell bound lipases could be released into the supernatant (3,6,24). It is possible that a surfactant could cause dissociation of the lipase from *C.bombicola* even in aqueous suspensions. One surfactant, Tween, was investigated. Unfortunately, this surfactant caused a clearing of the agar test in a control experiment, and thus could not be used for this study.

The partitioning of the lipase into the agar reached a saturation level at higher concentrations. This led to the initially confusing results (Figure 4.13-4.14) in which no change in lipase concentration was observed throughout two consecutive cycles. A simple solution to this problem was found by diluting samples down to an appropriate

range. A dilution to one third of the initial concentration was found to be satisfactory. Because the agar had variation from batch to batch and time to time, it was necessary to do frequent calibration curves.

The final, and perhaps most important effect noticed with the lipase, was that samples which had been frozen and thawed (even multiple times) could be used in the lipase test. This allowed faster sampling rates since the test did not need to be performed immediately upon sampling. Even more importantly, many samples could be assayed within a short period of time, therefore the agar was as consistent as possible for the various samples. It should be noted that freezing the samples caused a significant but predictable drop in the lipase activity, as seen in Figure 4.18. This loss was only observed once, and subsequent freeze-thaw cycles did not cause further loss of activity (see Table 4.2). This drop in activity was not always proportional to the initial activity, but actually seemed to improve the quality of the data, as seen in Figures 4.19-4.20. Perhaps a compound in the broth was interfering with the fresh samples, and was rendered inactive on freezing. Another possibility is that there are two different types of lipase being produced, one of which does not increase exponentially and which did not survive the freezing without losing its activity.

It should be noted that samples with glucose as the only carbon source lost all activity on freezing. It is likely, therefore, that either the oil, or a metabolite produced when oil is in the medium protects the lipase during freezing. Not surprisingly, lipases have been shown to associate with triglycerides, especially on freezing (7). It is possible that the association of lipase with triglyceride protected the

enzyme activity in these experiments. The samples from the experiments with only glucose present did not have this protection and the activity was lost.

5.7 Lipase Production

Two very different profiles of lipase production were obtained. The profile for the case where glucose and oil were used as carbon sources showed clearly that the lipase was produced in a growth associated fashion (Figure 4.16). This type of production is very favourable for SCF, which has been found to have excellent biomass production rates (4,5,21,23,30,71).

When oil was used as the sole carbon source, a strikingly different production profile was obtained (Figure 4.17). The lipase production was no longer growth associated; indeed, there was a pronounced maximum half way through the cycle. A possible explanation is that the organism re-metabolized the enzyme after a certain amount of oil had been hydrolysed. Lipase activity has been found to drop in several cases, once the organism has stopped growing (2,15,19). This was not the case for the data presented here since nutrient levels were not exhausted; residual oil was always present in the broth, and for this case, nitrogen was supplied in large excess. Perhaps a certain level of fatty acids in the broth caused the microorganism to re-metabolise the lipase enzyme. Another possibility was that sufficient quantities of the triglycerides had hydrolysed to leave a portion of the enzyme unprotected from freezing. In this way, the actual lipase level may have remained constant, or

continued to increase throughout the cycle, even though measurements of the frozen samples showed a decrease.

For the different nitrogen levels investigated, no significant difference in lipase levels were noticeable (Figure 4.21). Using ammonium nitrate did result in higher lipase activities compared with when using ammonium sulfate, but this was negated by the longer cycle times. This indicates that the presence of nitrate, while inhibitory to growth, promotes a larger specific production of lipase.

6.0 CONCLUSIONS

It was found that *Candida bombicola* produces a cell-associated lipase. It was also found that the agar diffusion lipase assay of Lawrence *et al.* could be applied to such a lipase, provided that the sample was diluted into an operational range. It was also found that frozen samples could be used for the test, and that freezing decreased the activity of the samples. However, the results obtained after freezing were reproducible and generated reasonable trends.

Production profiles of the lipase were determined with two different substrates. Growing on glucose and oil *C.bombicola* produced lipase in a growth-associated fashion. On oil alone, the lipase had a prominent maximum in concentration during the middle of the cycle.

It was found that a pressure transducer could be used to measure dosing and harvesting volumes in the Self-Cycling Fermenter, provided the broth did not foam excessively. This technique lowered the cost and complexity of the fermenter.

7.0 REFERENCES

1. Blumenthal, K., Zahler, S. (1962) Index for Measurement of Synchronization of cell populations. Science 135:724

2. Breuil, C., Kushner, D.J. (1975) Lipase and Esterase Formation by Psychrophilic and Mesophilic *Acinetobacter* Species. Canadian Journal of Microbiology 21:423-433

3. Breuil, C., *et al.* (1978) Stimulation of Lipase Production During Bacterial Growth on Alkanes. Journal of Bacteriology 133:601-606

4. Brown, W.A., Cooper, D.G. (1991) Self-Cycling Fermentation Applied to *Acinetobacter calcoaceticus* RAG-1. Applied and Environmental Microbiology 57:2901-2906

5. Brown, W.A., Cooper, D.G. (1992) Hydrocarbon Degradation by *Acinetobacter calcoaceticus* RAG-1 Using the Self-Cycling Fermentation Technique. Biotechnology and Bioengineering 44:1331-1336

6. Chen, J., *et al.* (1994) Lipase Production from Hydrocarbons by *Trichosporon fermentans* WU-C12 in the Presence of Surfactants. Bioscience, Biotechnology and Biochemistry 58:773-775

7. Choi, I.W., Jeon, I.J. (1993) Patterns of Fatty Acids Released from Milk Fat by Residual Lipase During Storage of Ultra-High Temperature Processed Milk. Journal of Dairy Science 76:78-85

8. Conway, E.J. (1957) Microdiffusion Analysis and Volumetric Error. 4th edition, Crosby Lockwood, London

9. Cooper, D.G., Paddock, D.A. (1984) Production of a Biosurfactant from *Torulopsis bombicola*. Applied and Environmental Microbiology 47:173-176

10. Daniels, L., *et al.* (1994) Chemical Analysis. In Gerhardt, P. *et al.* (ed.) Methods for General and Molecular Bacteriology. ASM, Washington, D.C.

11. Davila, A., *et al.* (1992) Kinetics and balance of a fermentation free from product inhibition: sophorose lipid production by *Candida bombicola*. Applied Microbiology and Biotechnology 38:6-11

12. Dawson, P.S.S. (1971) A Pilot Plant Apparatus for Continuous Phased Culture - Some Observations on Oxygen Usage by *Candida utilis* During the Cell Cycle. Biotechnology and Bioengineering 13:877-892

13. Dawson, P.S.S. (1972) Continuously Synchronized Growth. Journal of Applied Chemistry and Biotechnology 22:79-103

14. Dawson, P.S.S., *et al.* (1971) The Cyclone Column Culture Vessel for Batch and Continuous, Synchronous or Asynchronous, Culture of Microorganisms. Biotechnology and Bioengineering 13:865-876

15. Fox, P., Stepaniak, L. (1983) Isolation and Some Properties of Extracellular Heat-Stable Lipases From *Pseudomonas fluorescens* strain AFT 36. Journal of Dairy Research 50:77-89

16. Geankoplis, C.J. (1983) Transport Processes and Unit Operations. 2nd ed. Prentice Hall, New Jersey

17. Iwai, M., *et al.* (1975) The Comparison of the Properties of two Lipases from *Penicillium cyclopium* Westring. Agricultural and Biological Chemistry 39:1063-1070

18. Lawrence, R.C., *et al.* (1967) Rapid Method for the Quantitative Estimation of Microbial Lipases. Nature 213:1264-1265.

19. Lawrence, R.C., *et al.* (1967) The Production and Characterization of Lipases from a Micrococcus and a Pseudomonad. Journal of General Microbiology 48:401-418

20. McCaffrey, W.C. (1992) Secondary Metabolite Production Using Self-Cycling Fermentation (SCF). A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of Master of Engineering. Department of Chemical Engineering, McGill University, Montreal, Canada.

21. McCaffrey, W.C., Cooper, D.G. (1995) Sophorolipids Production by *Candida bombicola* Using Self-Cycling Fermentation. Journal of Fermentation and Bioengineering 79:146-151

22. Paddock, D.A. (1982) Surface activity of *Torulopsis* Species. A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of Master of Engineering. Department of Chemical Engineering, McGill University, Montreal, Canada.

23. Sarkis, B.E., Cooper, D.G. (1993) Biodegradation of Aromatic Compounds in a Self-Cycling Fermenter (SCF). The Canadian Journal of Chemical Engineering 72:874-880

24. Shabtai, Y., Daya-Mishne, N. (1992) Production, Purification, and Properties of a Lipase from a Bacterium (*Pseudomonas aeruginosa* YS-7) Capable of Growing in Water-Restricted Environments. Applied and Environmental Microbiology 58:174-180

25. Shapiro, S., Vining, L.C. (1984) Suppression of Nitrate Utilization by Ammonium and its Relationship to Chloramphenicol Production in *Streptomyces venezuelae*. The Canadian Journal of Microbiology 30:798-804

26. Sheppard, J.D., Cooper, D.G. (1990) Development of Computerized Feedback Control for the Continuous Phasing of *Bacillus subtilis*. Biotechnology and Bioengineering 36:539-545

27. Sheppard, J.D., Cooper, D.G. (1990) The Effects of a Biosurfactant on Oxygen Transfer in a Cyclone Column Reactor. Journal of Chemical Technology and Biotechnology 48:325-336

28. Stead, D. (1984) Evaluation of a Fluorimetric Assay on the Lipases from Strains of Milk Psychrotrophic Bacteria. Journal of Dairy Research 51:123-130

29. Stead, D. (1986) Microbial Lipases: Their Characteristics, Role in Food Spoilage and Industrial Uses. Journal of Dairy Research 53:481-505

30. Van Walsum, G.P., Cooper, D.G. (1993) Self-Cycling Fermentation in a Stirred Tank Reactor. Biotechnology and Bioengineering 42:1175-1180

31. Wentworth, S.D., Cooper, D.G. (1996) ?. Journal of Fermentation and Bioengineering 81:400-405

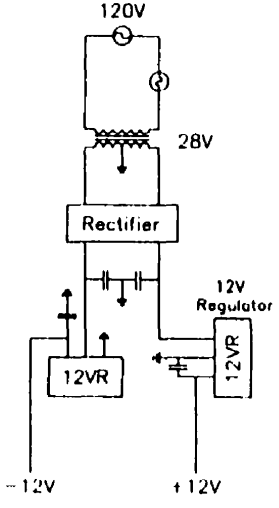
32. Wincure, B.M., et al. (1995) Mathematical Model of Self-Cycling Fermentation. Biotechnology and Bioengineering 46:180-183

33. Zenaitis, M.G., Cooper, D.G. (1994) Antibiotic Production by *Steptomyces aureofaciens* Using Self-Cycling Fermentation. Biotechnology and Bioengineering 44:1331-1336

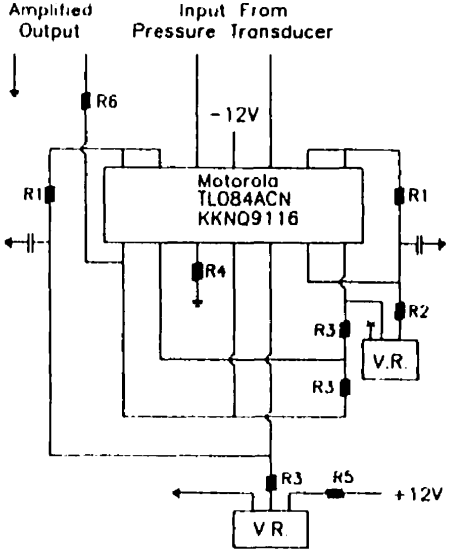
34. Zhou, Q., Kosaric, N. (1995) Utilization of Canola Oil and Lactose to Produce Biosurfactant with *Candida bombicola*. Journal of the American Oil Chemists Society 72:67-72

APPENDIX

Power Supply
(Pressure Transducer)



Amplification Circuit
(Unused)



Resistor Codes
R1=yorg
R2=yoyg
R3=brblog
R4=rblrg
R5=gbrog
R6=rblog

V.R =Variable Reistance
(W503 314C)

N.B. This circuit is located on the same board as the power supply.

8.2 Control Program

Excerpts from the control program are given on the following pages.

```
{$N+}
{
   ***********************************************
   *** MMAY14.PAS          Last Update: 96/07/17 ***
   *** Program for operation of Self-Cycling Reactor ***
   *** Modified from THIRD.PAS                    ***
   ***    Michael May                            ***
   ***********************************************
}
Program body;

{ Include files must appear in this order }

{$I MMGEN3.INC}
{$I MMAYT2.INC}
{$I MMAYF5.INC}
{$I MMAYG2.INC}
{$I MMAYI2.INC}
{$I MMAYP13.INC}

PROCEDURE Initialize_System;
VAR Reply : Char;

BEGIN {Initialize_System}
  ClrScr;
  Write('Do you want the pump on now ?  Y/N : ');
  Reply:=Get_YN;
  IF Reply='Y' THEN
    BEGIN
    Reactor_Pump_On;
    Pump_On:=True;
    END;
  Check_Time;
  Init_Variables;
  Setup_Files;
  Set_Global_Variables;
END;  {Initialize_System}

PROCEDURE Auto_Mode;
VAR Choice : Byte;

PROCEDURE Resume_Auto_Setup(var Choice,Minima_Detected : byte);
BEGIN
  Update_Data(Cycle);
  ClrScr;
  WriteIn('Choose point of entry into program');
  WriteIn;
  WriteIn('1 - Time delay before monitoring for DO minimum');
  WriteIn('2 - End of Cycle Time Delay');
  WriteIn('3 - Cycle');
  WriteIn('4 - Operate syringe pump');
  WriteIn('5 - Fill collection jar');
  WriteIn;
  Write('Make a selection -->');
  ReadIn(Choice);
  IF Choice = 1 THEN
    BEGIN
    Write(#10,'Out of ',Number_of_Minima,' time delay(s), which do you
want?  : ');
    ReadIn(Minima_Detected);
    Minima_Detected:=Minima_Detected-1;
    END;
END; {Resume_Auto_Setup}

PROCEDURE Start_New_Auto;
BEGIN
  Start_Time:=0;
  Create_Data_File;
  Minima_Detected:=0;
END;  {Start_New_Auto}

PROCEDURE Automatic_Cycle(var Choice : Byte);
VAR Dummy : Boolean;
BEGIN
  WHILE Choice <= 5 DO
    BEGIN
    Case Choice of
      1: BEGIN
```

```
      REPEAT
        Auto_Reset:=False;
        TempString[1]:='Time Delay '+Int2Str(Minima_Detected+1);
        TempString[2]:='Ends at
'+Count_to_Time(Time_Delay_Variable[Minima_Detected+2]);
        Display_Message(TempString);
        Time_Delay(Time_Delay_Variable[Minima_Detected+2],Dummy);
        IF NOT Auto_Reset THEN
          BEGIN
          TempString[1]:='Monitoring for Minimum
'+Int2Str(Minima_Detected+1);
          TempString[2]:='Cycle '+Int2Str(Cycle);
          Display_Message(TempString);
          Sample_Dissolved_Oxygen(Minimum_DO);
          END;
        IF NOT Auto_Reset THEN
          BEGIN
          Minima_Detected:=Minima_Detected+1;

Time_To_Minimum[Minima_Detected]:=Get_Time_Count-Start_Time;
          END;
      UNTIL Minima_Detected = Number_of_Minima;
      Minima_Detected:=0;
      END;
    2: BEGIN
      IF Time_Delay_Variable[1] > 0 THEN
        BEGIN
        TempString[1]:='End Of Cycle Time Delay';
        TempString[2]:='Ends at '+Count_to_Time(Time_Delay_Variable[1]);
        Display_Message(TempString);
        Time_Delay(Time_Delay_Variable[1],Dummy);
        END;
      END;
    3: BEGIN
      TempString[1]:='Cycling';
      TempString[2]:='Cycle '+Int2Str(Cycle);
      Display_Message(TempString);
      Cycle_Now(Volt_difference,PTFudge,PTFudge_Full,True);
      END;
    4: BEGIN
      TempString[1]:='Syringe Pump';
      TempString[2]:='Addition Time : '+Real2Str(Volume,6,1);
      Display_Message(TempString);
      Add_Hydrocarbon_Substrate(Volume);
      END;
    5: BEGIN
      IF Take_Samples THEN
        BEGIN
        TempString[1]:='Filling Sample Jar #'+Int2Str(Jar);
        TempString[2]:='Fill Time : '+Int2Str(Sample_Time);
        Display_Message(TempString);
        Fill_Collection_Jar(Jar,Sample_Time);
        END
      ELSE
        BEGIN
        TempString[1]:='No Sample Taken';
        TempString[2]:=' ';
        Display_Message(TempString);
        END;
      END;
    END;  {CASE statement}
    Choice:=Choice + 1;
  END;  {While choice <= 5}
  REPEAT
    Jar:=Jar+1;
    IF Jar > Max_Jars THEN Jar:= 1;
  UNTIL Usable_Jar(Jar);
  Save_Data;
  IF RecordChart THEN Save_Chart;
  Record_Data;
  Report_Data;
  Choice:=1;
END;  {automatic_cycle}

BEGIN  {Auto_Mode}
  Reactor_Pump_On;
  IF Cycle>1 THEN
    Resume_Auto_Setup(Choice,Minima_Detected)
```

```
  ELSE
    BEGIN
    Start_New_Auto;
    Choice:=1;
    Minima_Detected:=0;
    END;
  IF Sensitivity < 1e-5 THEN Get_Essential_Variables;
  Initialize_Report_Data;
  IF Cycle > 1 THEN Restore_Report_Data;
  While Cycle <= End_cycle DO Automatic_Cycle(Choice);
  Close_Report_Data;
END; {auto mode}

PROCEDURE Manual_Mode;
VAR Choice, Selection, Reply: Char;

PROCEDURE File_Stuff;
VAR Choice,Reply : Char;
BEGIN
  REPEAT
    ClrScr;
    WriteIn('File Handling',#10);
    WriteIn('1 - Start a new file and set cycle time (overwrites file)');
    WriteIn('2 - Increase cycle, reset cycle time, and save');
    WriteIn('3 - Minimum detected',#10);
    WriteIn('E - Exit',#10);
    Write('Enter your selection ',#16,' ');
    ReadIn(Choice);
    Case Choice of
      '1': BEGIN
        WriteIn(#10,'Are you sure, this will overwrite the existing phasing
file!');
        Write('(Y/N)   Y overwrites ',DataFile_Name,'.DAT  : ');
        Reply:=Get_YN;
        IF Reply = 'Y' THEN
          BEGIN
          Start_Time:=0;
          Create_Data_File;
          Cycle:=1;
          END;
        END;
      '2': BEGIN
        Cycle_time:=Get_Time_Count-Start_Time;
        Start_Time:=Get_Time_Count;
        Cycle:=Cycle+1;
        Minima_Detected:=0;
        Save_Data;
        Record_Data;
        REPEAT
          Jar:=Jar+1;
          IF Jar > Max_Jars THEN Jar:= 1;
        UNTIL Usable_Jar(Jar);
        Auto_Reset:=True;
        END;
      '3': BEGIN
        IF Minima_Detected = Number_of_Minima THEN
          BEGIN
          WriteIn(#10,'All minima have been detected, cycle now');
          ReadIn;
          END
        ELSE
          BEGIN
          Minima_Detected:=Minima_Detected+1;

Time_To_Minimum[Minima_Detected]:=Get_Time_Count-Start_Time;
          Auto_Reset:=True;
          END;
        END;
      END; {case}
  UNTIL UpCase(Choice) = 'E';
END; {File_Stuff}

PROCEDURE Direct_Control;
VAR Choice: Char;
    Solenoid_0, Solenoid_1: Integer;
BEGIN
  REPEAT
    ClrScr;
```

```
  Writeln('Direct Operation of Valve Mode',#10,#10);
  Writeln(' 1-8 samplig vials');
  Writeln(' (H)arvesting');
  Writeln(' (D)osing');
  Writeln(' (P)ump is on : ',Pump_On);
  Writeln(' (S)end command to solenoid bank',#10);
  Writeln(' (E)xit',#10);
  Write('Enter Selection ',#16,' ');
  Readln(Choice);
  Choice:=Upcase(Choice);
  Case Choice of
    '1'..'8': Open_Valve(ord(Choice)-48);
    'H' : Open_Valve(Harvest_Valve);
    'D' : Open_Valve(Dose_Valve);
    'P' : Pump_On:=NOT(Pump_On);
    'S' : BEGIN
      Write('Enter solenoid 0 value :');
      Readln(Solenoid_0);
      Write('Enter solenoid 1 value :');
      Readln(Solenoid_1);
      Write('Are these correct?  Y/N : ');
      Choice:=Get_YN;
      IF Choice = 'Y' THEN Actuate_Valve(Solenoid_0, Solenoid_1);
      END;
    END; {case}
   IF (Choice <> 'E') and (Choice <> 'P') THEN
    BEGIN
    Write(#10,'Hit Enter When Done');
    Readln;
    END;
   Close_Valve;
  UNTIL Choice='E';
END; {Direct_Control}

PROCEDURE Timed_Operations;
VAR Choice : Char;
   Fill_Time : Byte;
BEGIN
  REPEAT
  ClrScr;
  Writeln('Timed Operations',#10);
  Writeln('1 - Add/Change Hydrocarbon to Reactor');
  Writeln('2 - Fill Collection Bottle');
  Writeln('3 - Timed Dosing Vessel Fill');
  Writeln('4 - Timed Harvesting Vessel Fill',#10);
  Writeln('E - Exit',#10);
  Write('Enter Selection ',#16,' ');
  Readln(Choice);
  Case Choice of
   '1': BEGIN
    Writeln(#10,'HYDROCARBON ADDITION',#10);
    Writeln('The current time for the syringe pump is ',Volume:6:1);
    Write('Do you want to Change or Add hydrocarbon?  (C/A)');
    Readln(Reply);
    IF (Reply='C') OR (Reply='c') THEN
     BEGIN
     Write('What is the new hydrocarbon addition time? -> ');
     Readln(Volume);
     END
    ELSE IF  (Reply='A') OR (Reply='a')  THEN
     BEGIN
     Write('How many seconds of Hydrocarbon?  -> ');
     Readln(Volume);
     Add_Hydrocarbon_Substrate(Volume);
     END;
    END;
   '2': BEGIN
    Writeln(#10,'SAMPLE COLLECTION',#10);
    Write('Which sample jar? (1-8, Jar ',Jar,' is current)  -> ');
    Readln(Sample_Jar);
    Fill_Collection_Jar(Sample_jar,Sample_Time);
    END;
   '3': BEGIN
    Writeln(#10,'TIMED DOSING VESSEL FILL',#10);
    Write('How many seconds of Dosing?  -> ');
    Readln(Fill_Time);
    Timed_Dosing(Fill_Time);
    END;
```

```
    '4': BEGIN
      Writeln(#10,'TIMED HARVEST VESSEL FILL',#10);
      Write('How many seconds of Harvesting?  -> ');
      Readln(Fill_Time);
      Timed_Harvest(Fill_Time);
      END;
    END; {case}
  UNTIL UpCase(Choice) = 'E';
  ClrScr;
END; {Timed_Operations}

PROCEDURE Monitor_Signals;
VAR Signal :Real;
    Choice :Char;
    Record_Signal :Boolean;
    i :Byte;
    Test :Char;
BEGIN
  Record_Signal:=False;
  REPEAT
    ClrScr;
    Writeln('Press the following keys during monitoring for desired effect:');
    Writeln('      H - Harvest Valve     D - Dosing Valve');
    Writeln('      R - Reactor Pump      Q - Quit',#10);
    Writeln('Monitor Signal :',#10);
    Writeln('1 - Dissolved Oxygen');
    Writeln('2 - Pressure Transducer');
    Writeln('3 - Record Signal : ',Record_Signal);
    Writeln('E - Exit',#10);
    Write('Enter Selection ',#16,' ');
    Readln(Choice);
    Test:='A';
    Case Choice of
      '1': Init_DO_Probe;
      '2': Init_Pressure_Transducer;
      '3': Record_Signal:=NOT(Record_Signal);
      END; {Case}
    IF (Choice = '1') OR (Choice = '2') THEN
      BEGIN
      i:=1;
      IF MemAvail < SizeOf(ChartRecord) THEN Record_Signal:=False;
      IF Record_Signal THEN Get_Temp_Filename;
      ClearBuffer;
      REPEAT
        Signal:=Read_Data;
        IF Record_Signal THEN
          BEGIN
          Buffer[i].Signal:=Signal;
          Buffer[i].Time:=Get_Time_Count;
          i:=i+1;
          IF i = buffersize+1 THEN Update_Chart(i);
          END;
        Case Choice of
          '1': Writeln('Oxygen level : ',Signal);
          '2': Writeln('Pressure signal : ',Signal);
          END; {Case}
        IF Keypressed THEN
          BEGIN
          Test:=Readkey;
          CASE Test OF
            'h': Open_Valve(Harvest_Valve);
            'd': Open_Valve(Dose_Valve);
            'r': BEGIN
              Pump_On:=NOT(Pump_On);
              Close_Valve;
              END;
            ' ': Close_Valve;
            END; {Case}
          END;
        UNTIL (Test='Q') OR (Test='q');
        IF Record_Signal THEN
          BEGIN
          FinalPoints:=i-1;
          Save_Temp_Chart;
          END;
        END;
      ClearBuffer;
    UNTIL UpCase(Choice) = 'E';
```

```
END;  { Monitor Signals }

BEGIN  {Manual_Mode}
  ClrScr;
  Repeat
    ClrScr;
    Writeln('Choose what you want to do',#10);
    Writeln('1 - Timed Operations');
    Writeln('C - Cycle using Pressure Transducer signal');
    Writeln('P - Reactor pump is on  : ',Pump_On);
    Writeln('M - Monitor Signals');
    Writeln('F - File Stuff');
    Writeln('G - Graphical display options');
    Writeln('R - Manual reset is on (skips calling time delays) :
',Manual_Reset);
    Writeln('V - Change Variables');
    Writeln('D - Direct control of valves',#10);
    Writeln('E - Exit Manual Mode',#10,#10);
    Write('Make a selection -->');
    Readln(Choice);
    Choice:=Upcase(Choice);
    Case Choice of
      '1': Timed_Operations;
      'P': BEGIN
        Pump_On:=NOT(Pump_On);
        IF Pump_On THEN Reactor_Pump_On
        ELSE Reactor_Pump_Off;
        END;
      'M': Monitor_Signals;
      'C':BEGIN
        ClrScr;
        Writeln(#10,'CYCLE USING PRESSURE TRANSDUCER SIGNAL');
        Writeln('Type Capital Q to quit',#10);
        Writeln('Full  / Half Full / Current signal');
        Cycle_Now(Volt_difference,PTFudge,PTFudge_Full,False);
        END;
      'V': Set_Global_Variables;
      'D': Direct_Control;
      'R': Manual_Reset:=NOT(Manual_Reset);
      'F': File_Stuff;
      'G': Get_Graphics_Options;
      END; {Case}
  UNTIL Choice = 'E';
  ClrScr;
  Writeln('Manual mode complete');
END; {Manual_Mode}

PROCEDURE Main_Menu;
VAR Reply : Char;
BEGIN
  REPEAT
    ClrScr;
    Writeln;
    Writeln('Main Menu',#10,#10);
    Writeln('1 : Manual Mode');
    Writeln('2 : Automatic Mode',#10);
    Writeln('E : Exit program',#10,#10);
    Write('Enter Selection ',#16,' ');
    Readln(Reply);
    Reply:=Upcase(Reply);
    Case Reply of
      '1': Manual_Mode;
      '2': Auto_Mode;
      END;  {case}
  UNTIL Reply='E';
END; {Main_Menu}

BEGIN {program}
  Initialize_System;
  Main_Menu;
END.
```

```
{
  ************************************************************
  *** MMAYP13.INC                Last Update: 96/07/17  ***
  *** This file has all the nitty gritty stuff in it      ***
  *** including interface with the D.A. board.              ***
  *** Modified from PHASING.INC                             ***
  ***    Michael May                                     ***
  ************************************************************
}
CONST
  Base_address = $2EC;
  Command_register = Base_address + 1;
  Status_register  = Base_address + 1;
  Data_register    = Base_address;

  Command_wait = $4;      { all of these global variables are for }
  Write_wait   = $2;      { the data acquisition board            }
  Read_Wait    = $5;

  CReset      = $0;
  CError      = $2;
  CStop       = $F;
  CClock      = $3;
  DIOPort     = $2;
  CDOut       = $5;
  CSAD        = $D;
  CDIOut      = $7;
  CRAD        = $E;

  Max_Conversions = 100;

  COFF        = 0;
{Solenoid bank 0  8-15}
  CPUMP         = 1;
  CSAMPLE_JAR_5 = 2;
  CSAMPLE_JAR_6 = 4;
  CSAMPLE_JAR_7 = 8;
  CSAMPLE_JAR_8 = 16;
{Solenoid bank 1  0-7}
  CDOSING_VALVE  = 1;
  CHARVEST_VALVE = 2;

  CSAMPLE_JAR_1  = 8;
  CSAMPLE_JAR_2  = 16;
  CSAMPLE_JAR_3  = 32;
  CSAMPLE_JAR_4  = 64;
  CSYRINGE       = 128;

  Dose_Valve = Max_Jars+1;
  Harvest_Valve = Max_Jars+2;

TYPE Loop_Range = 1 .. Max_Conversions;

VAR
  number : integer;
  Solenoid_0, Solenoid_1 : Byte; { variables to actuate valves }
  Sample_jar : Byte;
  Conversions : Byte;

PROCEDURE Time_Delay(Delay : Real; var Man_Cycle : Boolean);
VAR key,reply: Char;
   Initial: LongInt;
   TempTempString: MessageString;
   TempTime: Byte;
BEGIN
 Initial:=Get_Time_Count;
 Manual_Reset:=False;
 Man_Cycle:=False;
 TempTime:=0;
 REPEAT
  IF Keypressed THEN
   IF ReadKey = Chr(0) THEN
    BEGIN
    Key:=ReadKey;
    Case Key of
     'h':   { alt F1 }
      BEGIN
```

```
        ClearBuffer;
        Graphics_to_Text;
        Manual_Mode;          { are passed to Manual_Mode     }
        Restore_Graphics;
        END;
      'q':  {  alt F10 }
        BEGIN
        ClearBuffer;
        TempTempString[1]:='Cycle Now?  Y/N';
        TempTempString[2]:=' ';
        Display_Message(TempTempString);
        Reply:=Get_YN;
        IF Reply='Y' THEN Man_Cycle:=True;
        Display_Message(TempString);
        END;
      END; {Case}
    END;
  IF (TempTime <> GetMin) THEN
    BEGIN
    Update_Time;
    TempTime:=GetMin;
    END;
 UNTIL ((Get_Time_Count-Initial) >= Delay) OR Manual_Reset OR
Man_Cycle OR Auto_Reset;
END;

PROCEDURE Reactor_Pump_On;
BEGIN
 Solenoid_0:=1;
 Solenoid_1:=0;
 Actuate_Valve(Solenoid_0, Solenoid_1);
 Pump_On:=True;
END;

PROCEDURE Reactor_Pump_Off;
BEGIN
 Solenoid_0:=0;
 Solenoid_1:=0;
 Actuate_Valve(Solenoid_0, Solenoid_1);
 Pump_On:=False
END;

PROCEDURE Open_Valve(Valve : Integer);
BEGIN
 Solenoid_0:=COFF;
 Solenoid_1:=COFF;
 Case Valve of
   1: Solenoid_1:=CSAMPLE_JAR_1;
   2: Solenoid_1:=CSAMPLE_JAR_2;
   3: Solenoid_1:=CSAMPLE_JAR_3;
   4: Solenoid_1:=CSAMPLE_JAR_4;
   5: Solenoid_0:=CSAMPLE_JAR_5;
   6: Solenoid_0:=CSAMPLE_JAR_6;
   7: Solenoid_0:=CSAMPLE_JAR_7;
   8: Solenoid_0:=CSAMPLE_JAR_8;
   Harvest_Valve: Solenoid_1:=CHARVEST_VALVE;  {10}
   Dose_Valve: Solenoid_1:=CDOSING_VALVE;  {9}
   END;
 IF Pump_On THEN Solenoid_0:=Solenoid_0+CPUMP;
 Actuate_Valve(Solenoid_0, Solenoid_1)
END;  {Open_Valve}

PROCEDURE Close_Valve;
BEGIN
 Solenoid_0:=COFF;
 IF Pump_On THEN Solenoid_0:=CPUMP;
 Solenoid_1:=COFF;
 Actuate_Valve(Solenoid_0, Solenoid_1);
END; {Close_Valve}

PROCEDURE Timed_Dosing(Seconds: Integer);
BEGIN
 Open_Valve(Dose_Valve);
 Delay(Seconds*1000);
 Close_Valve;
END;

PROCEDURE Timed_Harvest(Seconds: Integer);
```

```
BEGIN
  Open_Valve(Harvest_Valve);
  Delay(Seconds*1000);
  Close_Valve;
END;

PROCEDURE Sample_Dissolved_ Oxygen(VAR Low_DO: Real);
VAR New_DO: Real;
    Man_Cycle: Boolean;
BEGIN
  Init_DO_Probe;
  New_DO:=Read_Data;
  New_DO:=(Int(New_DO*1000+0.5))/1000;
  Low_Do:=New_DO;
  REPEAT
    New_DO:=Read_Data;
    New_DO:=(Int(New_DO*1000+0.5))/1000;
    IF (New_DO <= Low_DO) THEN
      BEGIN
      Low_DO:=New_DO;
      Time_Delay(120,Man_Cycle);
      END
    ELSE IF New_DO-Low_DO <= Sensitivity THEN
      BEGIN
      Time_Delay(120,Man_Cycle);
      END;
  Display_Current_DO(New_DO,Low_DO);
  UNTIL (New_DO-Low_DO > Sensitivity) OR Man_Cycle OR Auto_Reset;
END;

PROCEDURE Add_Hydrocarbon_Substrate(Volume:Real);
VAR Delay_Time : LongInt;
    Temp: Word;
BEGIN
  Solenoid_0:=CPUMP;
  Solenoid_1:=CSYRINGE;
  Actuate_Valve(Solenoid_0, Solenoid_1);
  Delay_Time:=Trunc(Volume* 1000);
  IF Delay_Time > 65000 THEN
    BEGIN
    FOR Temp:=1 TO (Delay_Time div 65000) DO Delay(65000);
    Delay_Time:=Delay_Time mod 65000;
    END;
  Temp:=Delay_Time;
  Delay(Delay_Time);
  Solenoid_0:=CPUMP;
  Solenoid_1:=COFF;
  Actuate_Valve(Solenoid_0, Solenoid_1);
END;

PROCEDURE Fill_Collection_Jar(jar,Sample_Time:Integer);
BEGIN
  Open_Valve(jar);
  Delay(Sample_Time*1000);
  Close_Valve;
END; {procedure}

PROCEDURE Stuck_Problem(Auto : Boolean);
VAR TempTempString: MessageString;
BEGIN
  IF Auto THEN
    BEGIN
    TempTempString[1]:='Valve Stuck or P.T. Bad';
    TempTempString[2]:='Hit Any Key';
    Display_Message(TempTempString);
    END
  ELSE
    BEGIN
    Writeln('Valve Stuck! (or pressure transducer bad)',#10);
    Writeln('Hit any key to stop alarm');
    END;
  ClearBuffer;
  REPEAT
    Sound(1000);
    Delay(500);
    NoSound;
    Delay(3000);
  UNTIL Keypressed;
```

```
 IF Auto THEN
   BEGIN
   TempTempString[1]:='Fix problem';
   TempTempString[2]:='Then Hit Enter';
   Display_Mess age(TempTempString);
   ClearBuffer;
   REPEAT UNTIL Keypressed;
   END
 ELSE
   BEGIN
   Writeln('Fix problem then hit enter');
   Readln;
   END;
END; {Stuck_Problem}

PROCEDURE Cycle_Now(Volt_difference,PTFudge,PTFudge_Full:Real;
Auto:Boolean);
CONST Loop_No=20;
VAR Half_full,Full,PT_signal: Real;
   m2,b2,m,Sxy,Sx,Sy,Sx2,b,x: Real;
   n,n2: Byte;
   Intermediate_Volume: Real;
   Start_channel,End_channel: Byte;
   Start_valve_time,t,t_final:Word;
   Stuck,Quit:Boolean;
   User_Input:Char;
   Temp_Loop:Integer;
   i,j:Byte;
BEGIN {Cycle_Now}
 Stuck:=False;
 Sxy:=0;
 Sx:=0;
 Sy:=0;
 Sx2:=0;
 n:=0;
 n2:=0;
 IF Pump_Off_Cycle THEN
   BEGIN
   Reactor_Pump_Off;
   Delay(Settle_Time*1000); {Wait 15 seconds to allow reactor volume to
settle}
   END;
 {Determine voltage signal from pressure transducer when reactor is full}
 IF MemAvail < SizeOf(ChartRecord) THEN RecordChart:=False;
 Init_Pressure_Transducer;
 Full:=0;
 i:=1;
 ChartSize:=0;
 Start_Cycling_Clock;
 IF (Cycle_Scheme = 3) OR (Cycle_Scheme = 4) THEN
   BEGIN
     FOR Temp_Loop:=1 TO Loop_No DO
       BEGIN
       Buffer[Temp_Loop].signal:=Read_Data;
       Buffer[Temp_Loop].time:=Get_Cycling_Time;
       Full:=Full+Buffer[Temp_Loop].signal;
       i:=i+1;
       IF i = buffersize+1 THEN Update_Chart(i);
       END;
   Full:=Full/Loop_No;
   END
 ELSE Full:=FFull;
 Half_full:=Full-Volt_difference-PTFudge;
 Buffer[i].signal:=Half_full;
 Buffer[i].time:=Get_Cycling_Time;
 i:=i+1;
 IF i = buffersize+1 THEN Update_Chart(i);
 Intermediate_Volume:=Approach_Fraction;
 Intermediate_Volume:=Full-Intermediate_Volume*(Full-Half_Full)/100;
 REPEAT
   Stuck:=False;
   Quit:=False;
   Start_valve_time:=Get_Cycling_Time;
   Open_Valve(Harvest_Valve);
   REPEAT
     IF NOT((n>2) AND ((t_final-t) < 100) AND ((Cycle_Scheme=1) OR
(Cycle_Scheme=3))) THEN
       BEGIN
```

```
      PT_Signal:=Read_Data;
      t:=Get_Cycling_Time;
      Buffer[i].Signal:=PT_Signal;
      Buffer[i].Time:=t;
      i:=i+1;
      IF i = buffersize+1 THEN Update_Chart(i);
      IF (PT_Signal <= Intermediate_Volume) AND ((Cycle_Scheme=1) OR
(Cycle_Scheme=3)) THEN
        BEGIN
        x:=t;
        Sxy:=Sxy+x*PT_Signal;
        Sx:=Sx+x;
        Sy:=Sy+PT_Signal;
        Sx2:=Sx2+Sqr(x);
        n:=n+1;
        IF n>1 THEN
          BEGIN
          m:=(n*Sxy-Sx*Sy)/(n*Sx2-Sqr(Sx));
          b:=(Sy*Sx2-Sxy*Sx)/(n*Sx2-Sqr(Sx));
          PT_Signal:=m*x+b;
          IF m<> 0 THEN x:=(Half_Full-b)/m ELSE x:=0;
          IF x > 0 THEN t_final:=Round(x) ELSE t_final:=t;
          END;
        END;
      IF (Start_valve_time+18000) < Get_Cycling_Time THEN
Stuck:=TRUE;
      IF PT_Signal < Minimum_Transducer_Signal THEN Stuck:=TRUE;
      IF NOT(Auto) THEN Writeln(Full,' ',Half_full,' ',PT_Signal);
      IF KeyPressed THEN User_Input:=Readkey;
      IF User_Input = 'Q' THEN Quit:=True;
      END
    ELSE
      BEGIN
      WHILE t < t_final DO t:=Get_Cycling_Time;
      PT_Signal:=Half_Full;
      END;
  UNTIL (PT_Signal <= Half_Full) OR Stuck OR Quit;
  Close_Valve;
  IF Stuck THEN Stuck_Problem(Auto);
UNTIL NOT(Stuck) OR Quit;
IF NOT Quit THEN
  BEGIN
  Sx:=0;
  Sxy:=0;
  Sy:=0;
  Sx2:=0;
  n2:=0;
  Full:=Full+PTFudge_Full;
  Buffer[i].signal:=Full;
  Buffer[i].time:=Get_Cycling_Time;
  i:=i+1;
  IF i = bufferSize+1 THEN Update_Chart(i);
  Start_Cycling_Clock;
  Intermediate_Volume:=Approach_Fraction;

Intermediate_Volume:=Half_Full+Intermediate_Volume*(Full-Half_Full)/100;
  REPEAT
    Stuck:=False;
    Start_valve_time:=Get_Cycling_Time;
    Open_Valve(Dose_Valve);
    REPEAT
      IF NOT((n2>2) AND ((t_final-t) < 100) AND ((Cycle_Scheme=1) OR
(Cycle_Scheme=3))) THEN
        BEGIN
        PT_Signal:=Read_Data;
        t:=Get_Cycling_Time;
        Buffer[i].Signal:=PT_Signal;
        Buffer[i].Time:=t;
        i:=i+1;
        IF i = buffersize+1 THEN Update_Chart(i);
        IF (PT_Signal >= Intermediate_Volume) AND ((Cycle_Scheme=1)
OR (Cycle_Scheme=3)) THEN
          BEGIN
          x:=t;
          Sxy:=Sxy+x*PT_Signal;
          Sx:=Sx+x;
          Sy:=Sy+PT_Signal;
          Sx2:=Sx2+Sqr(x);
```

```
        n2:=n2+1;
        IF n2>1 THEN
          BEGIN
          m2:=(n2*Sxy-Sx*Sy)/(n2*Sx2-Sqr(Sx));
          b2:=(Sy*Sx2-Sxy*Sx)/(n2*Sx2-Sqr(Sx));
          PT_Signal:=m2*x+b2;
          IF m2 <> 0 THEN x:=(Full-b2)/m2 ELSE x:=0;
          IF x > 0 THEN t_final:=Round(x) ELSE t_final:=t;
          END;
        END;
      IF (Start_valve_time+18000) < Get_Cycling_Time THEN
Stuck:=TRUE;
      IF PT_Signal < Minimum_Transducer_Signal THEN Stuck:=TRUE;
      IF NOT(Auto) THEN Writeln(Full,' ',Half_Full,' ',PT_Signal);
      IF KeyPressed THEN User_Input:=Readkey;
      IF User_Input = 'Q' THEN Quit:=True;
      END
    ELSE
      BEGIN
      WHILE t < t_final DO t:=Get_Cycling_Time;
      PT_Signal:=Full;
      END;
  UNTIL (PT_Signal >= Full) OR Stuck OR Quit;
  Close_Valve;
  IF Stuck THEN Stuck_Problem(Auto);
 UNTIL NOT(Stuck) OR Quit;
 END;
FinalPoints:=i-1;
Reactor_Pump_On;
Cycle_Time:=Get_Time_Count-Start_Time;
Start_Time:=Get_Time_Count;
Cycle:=Cycle+1;
Save_Stats(m,b,m2,b2,n,n2);
IF NOT(Auto) THEN
 BEGIN
 Save_Data;
 Record_Data;
 Minima_Detected:=0;
 IF RecordChart THEN Save_Chart;
  Auto_Reset:=True;
  END;
 ClearBuffer;
END; {Cycle_Now}
```

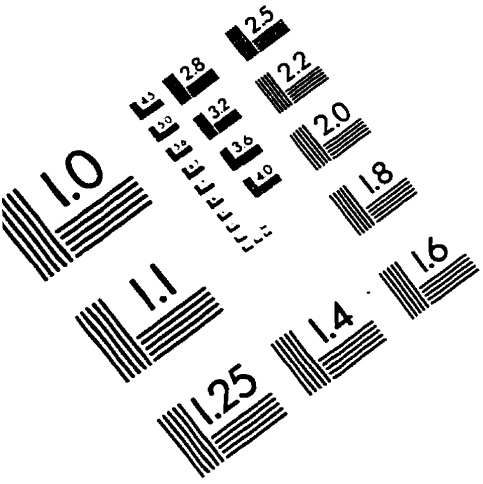
```
{
  ***************************************************************
  *** MMAYT2.INC                Last Update: 96/04/15    ***
  *** This include file covers time procedures                ***
  ***      Michael May                                       ***
  ***************************************************************
}
FUNCTION Year_Days(End_year,Start_year : Word) : Word;
VAR i: Byte;
   days: Word;
BEGIN
 IF End_year = Start_year THEN days:=0
 ELSE
   FOR i:=1 TO (End_year-Start_year) DO
     BEGIN
     days:=days+365;
     IF ((Start_year+i-1)/4-(Start_year+i-1) div 4) < 0.1 THEN
       days:=days+1;
     END;
 Year_Days:=days;
END; {Year_Days}

FUNCTION Days_In_Month(Month,Year : Word) : Byte;
BEGIN
  Case Month of
    1,3,5,7,8,10,12: Days_in_Month:=31;
    4,6,9,11: Days_in_Month:=30;
    2: IF (Year/4-Year div 4) < 0.1 THEN Days_In_Month:=29
       ELSE Days_in_Month:=28;
    END; {Case}
END; {Days_in_Month}

FUNCTION Month_Days(End_month,Start_month,End_year,Start_year :
Word) : Word;
VAR i,j : Byte;
   Days_to_Month, Month, Year : Array[1..2] of Word;
BEGIN
  Month[1]:=End_month;
  Month[2]:=Start_month;
  Year[1]:=End_year;
  Year[2]:=Start_year;
  FOR i:=1 TO 2 DO
    BEGIN
    Days_to_Month[i]:=0;
    IF Month[i] > 1 THEN
      FOR j:=1 TO Month[i]-1 DO
        Days_to_Month[i]:=Days_to_Month[i]+Days_in_Month(j,Year[i]);
    END;
  Month_Days:=Days_to_Month[1]-Days_to_Month[2];
END; {Month_Days}

FUNCTION Get_Time_Count
{(Year_Init,Month_Init,Day_Init,Hr_Init,Min_Init,Sec_Init : Word)} : LongInt;
VAR Year,Month,Day,Hr,Min,Sec,Dummy : Word;
   T1,T2 : LongInt;
   Days : LongInt;
   T3,T4,T5,T6 : Integer;
BEGIN
  GetDate(Year,Month,Day,Dummy);
  GetTime(Hr,Min,Sec,Dummy);

Days:=Year_Days(Year,Year_Init)+Month_Days(Month,Month_Init,Year,Yea
r_Init)+Day-Day_Init;
  T1:=Hr;
  T2:=Hr_Init;
  T3:=Min;
  T4:=Min_Init;
  T5:=Sec;
  T6:=Sec_Init;
  Get_Time_Count:=Days*24*3600+(T1-T2)*3600+(T3-T4)*60+T5-T6;
END; {Get_Time_Count}

FUNCTION Count_to_Time(Delta_Time :LongInt) :String;
VAR Temp : LongInt;
   Hrs,Mins,Hr,Min,Sec,Dummy : Word;
BEGIN
  Delta_Time:=Delta_Time div 60;
  Hrs:=Delta_Time div 60;
```

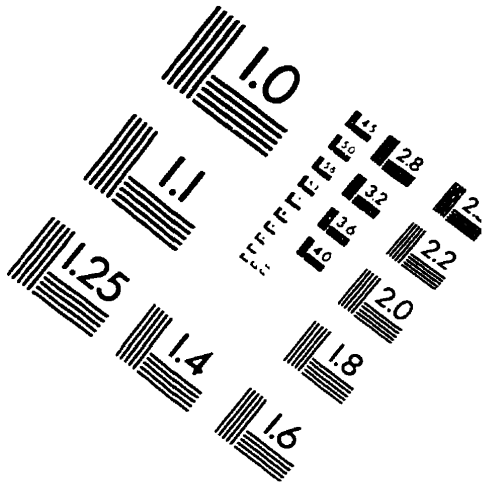
```
  Mins:=Delta_Time mod 60;
  GetTime(Hr,Min,Sec,Dummy);
  Hrs:=Hrs+(Min+Mins) div 60;
  Min:=(Min+Mins) mod 60;
  Hr:=(Hr+Hrs) mod 24;
  IF Min < 10 THEN
    Count_to_Time:=Int2Str(Hr)+':0'+Int2Str(Min)
  ELSE
    Count_to_Time:=Int2Str(Hr)+':'+Int2Str(Min);
END; {Count_to_Time}

PROCEDURE Get_Time_Now(var Year,Month,Day,Hr,Min,Sec : Word);
VAR Dummy : Word;
BEGIN
  GetDate(Year,Month,Day,Dummy);
  GetTime(Hr,Min,Sec,Dummy);
END; {Time_Now}

FUNCTION GetMin : Byte;
VAR Dum1,Dum2,Dum3,Min : Word;
BEGIN
  GetTime(Dum1,Min,Dum2,Dum3);
  GetMin:=Min;
END;

FUNCTION TimeOut :String;
VAR Year,Month,Day,Hr,Min,Sec : Word;
BEGIN
  Get_Time_Now(Year,Month,Day,Hr,Min,Sec);
  IF Min < 10 THEN
    TimeOut:=Int2Str(Hr)+':0'+Int2Str(Min)
  ELSE
    TimeOut:=Int2Str(Hr)+':'+Int2Str(Min);
END; {TimeOut}

FUNCTION DateOut :String;
VAR Year,Month,Day,Hr,Min,Sec : Word;
BEGIN
  Get_Time_Now(Year,Month,Day,Hr,Min,Sec);
  IF Day < 10 THEN
    DateOut:=Int2Str(Month)+'/0'+Int2Str(Day)
  ELSE
    DateOut:=Int2Str(Month)+'/'+Int2Str(Day);
END; {DateOut}

PROCEDURE Check_Time;
VAR Year,Month,Day,Dummy : Word;
    Hr,Min,Sec : Word;
    R1,R2,R3 : Word;
BEGIN
  GetDate(Year,Month,Day,Dummy);
  IF Year < 1995 THEN
    BEGIN
    ClrScr;
    Writeln('System Time is Incorrect, please enter correct time:',#10);
    REPEAT
      Write('Enter current year (yyyy) ',#16);
      Readln(R1);
    UNTIL (R1 >= 1995) AND (R1 < 2500);
    REPEAT
      Write('Enter current month (mm) ',#16);
      Readln(R2);
    UNTIL (R2 >= 1) AND (R2 <= 12);
    REPEAT
      Write('Enter current date (dd) ',#16);
      Readln(R3);
    UNTIL (R3 > 0) AND (R3 <= Days_In_Month(R2,R1));
    SetDate(R1,R2,R3);
    REPEAT
      Write('Enter time: (hh mm) ',#16);
      Readln(R1,R2);
    UNTIL (R1 >= 0) AND (R2 >= 0) AND (R1 <= 23) AND (R2 <= 59);
    SetTime(R1,R2,0,0);
    END;
END; {Check_Time}

PROCEDURE Start_Cycling_Clock;
VAR Dummy: Word;
```

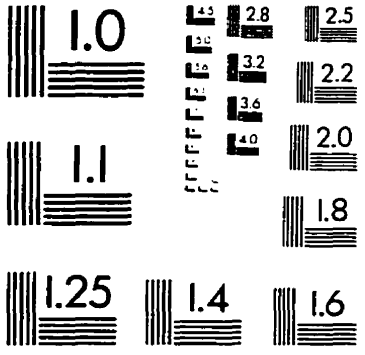
```
BEGIN
  GetTime(Dummy,Min1,Sec1,SSec1);
END;

FUNCTION Get_Cycling_Time : Word;
VAR Dummy,Min,Sec,SSec : Word;
BEGIN
  GetTime(Dummy,Min,Sec,SSec);
  IF Min < Min1 THEN Min:=Min+60;
  Get_Cycling_Time:=(Min-Min1)*6000+(Sec-Sec1)*100+SSec-SSec1;
END;
```

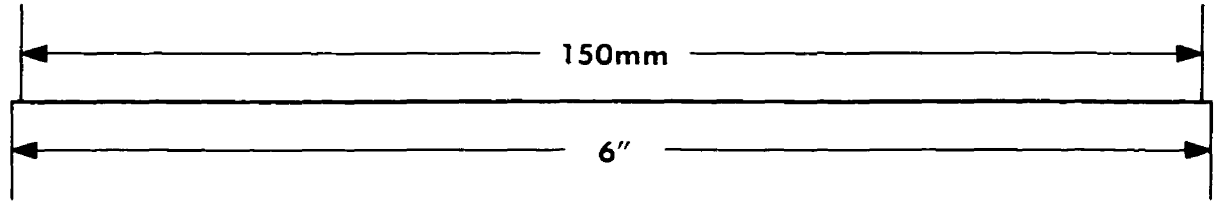
```
{
  ************************************************************
  *** MMGEN3.INC              Last Update: 96/07/17    ***
  *** This include file covers general procedures          ***
  ***      Michael May                                     ***
  ************************************************************
}

FUNCTION Boolean2Byte(Is_True: Boolean) : Byte;
BEGIN
  IF Is_True THEN Boolean2Byte:=1
  ELSE Boolean2Byte:=0;
END;  {Boolean2Byte}

FUNCTION Byte2Boolean( Number: Byte) : Boolean;
BEGIN
  IF Number = 1 THEN Byte2Boolean:=True
  ELSE Byte2Boolean:=False;
END; {Byte2Boolean}

FUNCTION Int2Str(L: LongInt) : String;
VAR S: String;
BEGIN
  Str(L,S);
  Int2Str:=S;
END; { Int2Str }

FUNCTION Real2Str(I : Real; A,B : Byte) : String;
VAR S: String;
BEGIN
   Str(I:A:B,S);
   Real2Str:=S;
END;

PROCEDURE Update_Chart(var i :Byte);

PROCEDURE Pan_Chart;
VAR j,k :Byte;
BEGIN
  j:=(ChartLength div buffersize) div 2;
  FOR k:=1 TO j DO
    BEGIN
    ChartRecord[k]^:=ChartRecord[k+j]^;
    Dispose(ChartRecord[k+j]);
    END;
  ChartSize:=j;
END;

BEGIN {Update_Chart}
  i:=1;
  IF RecordChart THEN
    BEGIN
    ChartSize:=ChartSize+1;
    New(ChartRecord[ChartSize]);
    ChartRecord[ChartSize]^:=Buffer;
    IF ChartSize >= ChartLength div buffersize THEN Pan_Chart;
    END;
END; {Update_Chart}

FUNCTION Usable_Jar(Jar_Number : Byte) : Boolean;
VAR Temp : Byte;
BEGIN
  Temp:=Jar_Flags shl (Jar_Number-1);
  Temp:=Temp shr 7;
  IF Temp = 1 THEN Usable_Jar:=True
  ELSE Usable_Jar:=False;
END;

PROCEDURE Toggle_Jar(Jar_Number : Byte);
VAR Temp : Byte;
BEGIN
  Temp:=1;
  Temp:=Temp shl (8-Jar_Number);
  Jar_Flags:=Jar_Flags xor Temp;
END;
```

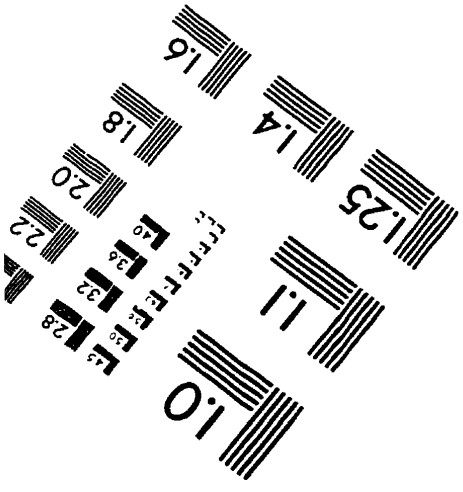
IMAGE EVALUATION
TEST TARGET (QA-3)

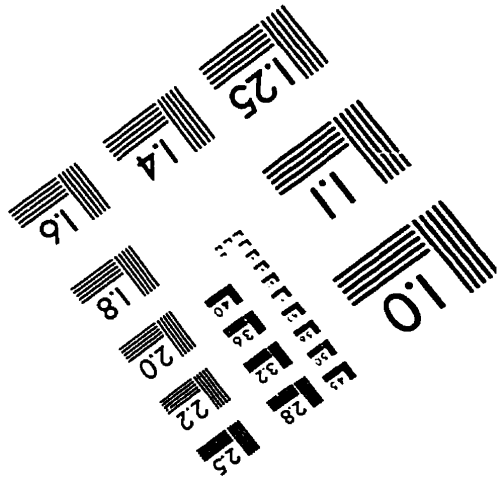












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989