MODELLING AND CONTROL OF DIELESS WIRE DRAWING

by

Yuandan Xie BSc

A thesis submitted to the Council for National Academic Awards in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Sponsoring Establishment : Department of Electrical and Electronic Engineering
Sheffield City Polytechnic

Collaborating Establishment : Davy McKee (Sheffield) Ltd.

December 1987

CONTENTS PAGE

LIST OF FIGURES PAGE

LIST OF PLATES PAGE

LISTS OF TABLES PAGE

ACKNOWLEDGEMENT

The author gratefully acknowledges the invaluable suggestions and continual assistance given by Dr. J. R. Travis and Dr. G .R. Symmons under whose supervision this work was carried out. Thanks are expressed to Dr. K. Dutton and Professor M. S. J. Hashmi, whose suggestions were received with gratitude.

The technical assistance offered by Mr. F. H. Brindley, Mr. R. Teasdale and their staff was much appreciated and particular thanks go to Mr. V. R. G. Siddons and Mr. A. K. Acaster for their cooperation during the duration of this work.

A final acknowledgement is paid to the Davy McKee (Sheffield) Ltd. for their collaborations.

DECLARATION

The author declares that no part of this work has been submitted in support of another degree or qualification to this or any other establishment. The author further declares that he has not been a registered candidate or enrolled student for another award of the CNAA or other academic or professional institution during the course of the research programme.

Yuandan Xie

ABSTRACT

MODELLING AND CONTROL OF DIELESS WIRE DRAWING

Yuandan Xie

A novel technique of drawing wire, i.e., dieless wire drawing, is described in which no conventional reduction dies are used. The wire is passed through a unit having stepped parallel bores filled with polymer melts, the smallest bore diameter being greater than the initial nominal wire diameter. The technique eliminates the need for a leader wire, prevents breakage during start-up, and eliminates the use of conventional reduction dies and, hence, the problem of die wear.

An analysis has been developed for predicting the produced wire sizes for given drawing speeds, the pressure distribution within the unit and the drawing load during the drawing process. The performance of the drawing process is effected by means of the plasto-hydrodynamic action of the polymer melt within the unit. Heat is generated by mechanical dissipation during the drawing process. The effect of this upon the performance of the process is considered by coupling the energy equation into the analysis and allowing the polymer viscosity to be dependent on temperature. Agreement between the theory and experiment is found to be close. To further investigate the performance of the drawing process, an extensive experimental study has been undertaken, in which parameters such as drawing speeds, the temperature of the polymer and type of polymers are varied.

Based on the theoretical and experimental results, a microcomputer-based control system for the dieless wire drawing process has been designed and built. A method with which the percentage reduction in area of the wire can be measured on line is proposed. Wires of desired qualities in terms of consistency in diameter over long lengths, or uniform change in area of the wire, are capable of being produced from the new system.

A self-tuning controller for determinisic systems is proposed, which has the same structure as a conventional PID controller. The new self-tuning PID controller is based on a generalised self-tuning controller with pole assignment for deterministic systems. Simulation and experimental studies for several examples indicate that the new controller performs well and can be further improved to provide a fundamental method of tuning a PID controller.

CHAPTER 1 : Introduction

1.1 - The Wire Drawing Process

1.2 - The Dieless Wire Drawing Process

1.3 - Background Development Leading to the Design of the Microcomputer-Based Control System and the Scope of the Present Work.

1.1 - The Wire Drawing Process

In the conventional wire drawing process, the reduction is achieved by pulling the wire through a tapered die. The die in this case acts primarily to reduce the wire diameter to a specific size, with an acceptable surface finish. Normally a die has a trumpet shaped bore, with a conical portion which serves to deform the wire. The minimum bore size of the die is always smaller than the inlet wire diameter. The possible reduction in area at each die may vary from 10 to 45 percent.

In wire drawing practice, lubrication is used to reduce the drawing load and die wear and hence improve the machine life and surface finish of the product. Generally two types of lubrication methods are used in wire drawing depending upon the material and the size of the wire. These are:

i) "Wet Drawing", in which the entire apparatus is submerged in a bath of lubricant. This type of lubrication is usually employed for wire diameter of less than 0.5 mm.

ii) "Dry Drawing", in which the wire is passed through a box of soap powder before entering the die. To facilitate the pick up of soap powder, the wire is passed through lime, borax or other alkaline substances. Dry drawing is usually used for wire diameters of more than 0.5 mm.

In conventional wire drawing, friction between the wire and the die is of the boundary type where metal to metal contact takes place in spite of the presence of a lubricant, resulting in die wear. Boundary lubrication methods have been used for the wire drawing process since the inception of the process itself. However, the tasks of producing satisfactory physical properties of the drawn wire to meet the demand for increased production and quality of product are proving to be beyond the scope of these traditional lubrication methods. Therefore, attempts

have been made to introduce other means of lubrication and to develop a more theoretical background to the process enabling greater understanding of the mechanisms involved, leading to improved and more efficient systems.

The hydrodynamic lubrication system was first introduced by Christopherson and Naylor [1]. In this system, a long close-fitting tube was placed before a conventional die. Oil was used for lubrication, and as the wire was pulled through the die, it pressurized the lubricant by viscous action. Experimental results showed that hydrodynamic lubrication was achieved under the designed conditions. It was necessary, however, to provide a leader to the full size wire in order to induce effective lubrication at the starting condition. The pressure nozzle had to be placed vertically and was of such a length that the wire industry found it too inconvenient to put it into practice. Also some die wear was still present. Modification to the above hydrodynamic system and other designs of lubricating system have been considered; for example, the combined hydrostatic and hydrodynamic lubrication and the double-die system [2], [3], [4]. However, the problem of breakage during start-up and the need for a leader to the full wire size was not solved by these designs.

1.2 - The Dieless Wire Drawing Process

A novel technique of wire drawing, i.e., dieless wire drawing, has been invented by Dr. G. R. Symmons and Dr. M. S. J. Hashmi, in which no conventional reduction dies are used and polymer melts are introduced as the lubricant in the drawing process [5].

The main feature of this technique is that the conventional dies are replaced by a dieless reduction unit (DRU) to reduce the wire diameter. Two types of DRU, one of tapered bore and the other of stepped

bore, have been designed and manufactured. Fig. 1.1 shows the details of both these units, in each case the smallest bore size of the DRU is always greater than the diameter of the undeformed wire.

The pulling action of the wire through the bore filled with viscous fluid gives rise to drag forces and generates hydrodynamic pressure; thus an effective die is generated. The combined effect of this hydrodynamic pressure and the drag force initiates plastic yielding and permanent deformation to the wire. The magnitude of the deformation depends on the performance of the viscous fluid, the geometrical configuration of the orifice and the drawing speed. It has been shown that reductions in cross-sectional area in excess of 20 percent may be obtained in a single pass when a wire (copper, mild steel or stainless) is pulled through the DRU filled with polymer melt. With a multi-unit drawing system higher deformations can be obtained. The polymer melt, in addition to acting as a lubricant, is also found to form a coating on the drawn wire. This coating is thought to be useful in protecting the wire against corrosion and also as a lubricant during any subsequent forming operation, e.g., bending or cold heading.

This technique has been useful in a number of ways in solving the problems associated with the conventional wire drawing process; e.g., die wear, initial wire breakage and the need for a leader wire. Since the diameter at the exit end of the DRU is greater than that of the wire, there is no metal to metal contact, and hence wear is no longer a problem. As no conventional reduction die is used, the need for a leader wire and breakage during start-up are also eliminated.

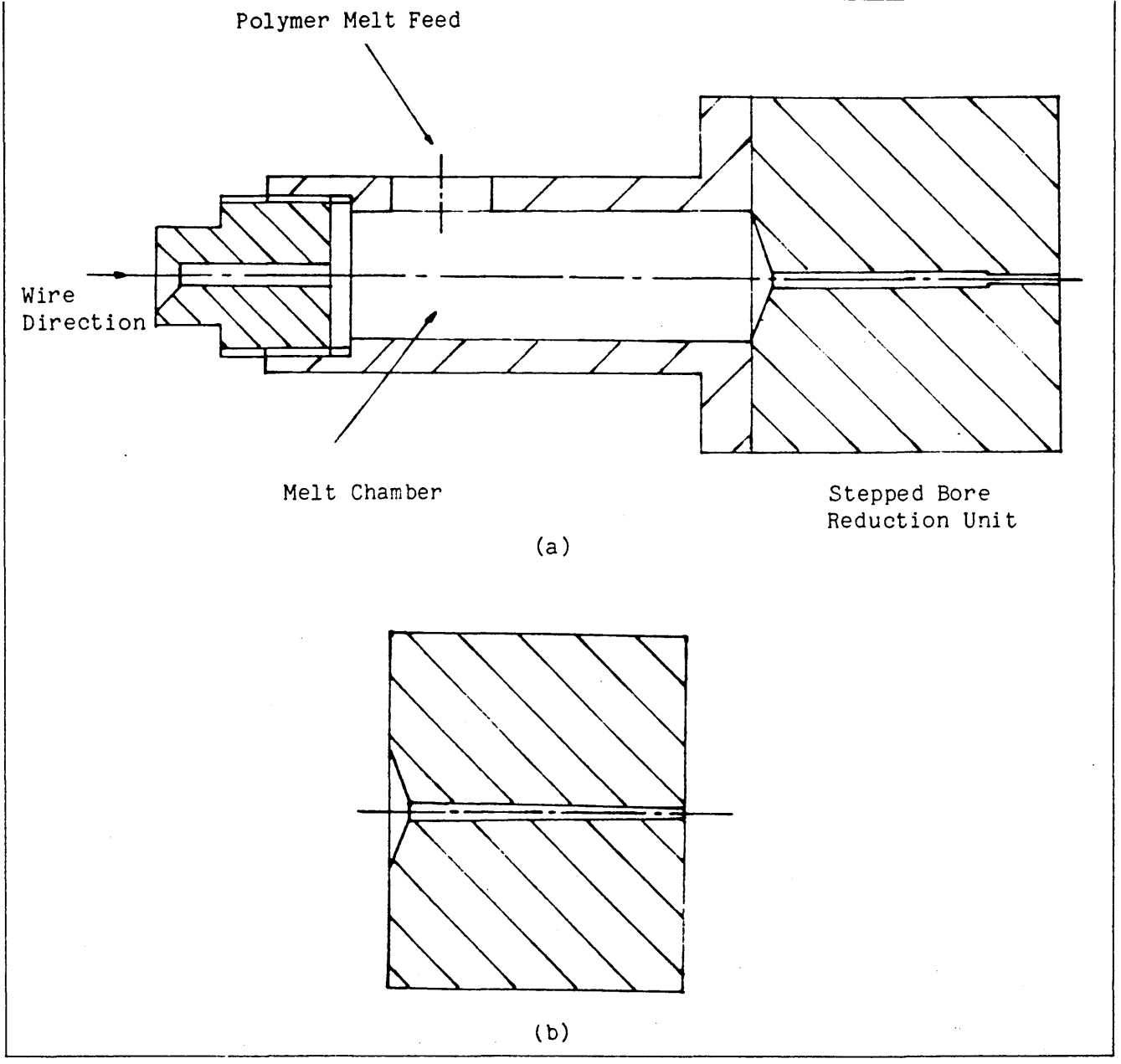


Fig.1.1 a) Stepped Bore Reduction Unit Assembly
b) Tapered Bore Reduction Unit

1.3 - Background Development Leading to the Design of the Microcomputer-Based Control System and the Scope of the Present Work.

Previous results from both theoretical analyses and practical tests in references [5], [6] and [7] show that, in the dieless wire drawing process, the reduction in area of the wire is affected by several factors. These are mainly the drawing speed, the performance of the

polymer used as pressure medium, the temperature of the polymer melt and the configuration of the DRU. The reduction of the wire is very sensitive to the drawing speed and the temperature of the polymer melt within the DRU. The results suggest that not only can normal wire of a uniform diameter be produced, but also that tapered wire with a uniform change in cross-sectional area of the wire can be produced by this drawing process. An application of this is the production of fine steel tapered wire for the manufacture of long wire ropes with a uniform change in cross-sectional area as used in deep sea mining.

Since the quality of the product is of utmost importance, methods of controlling the process must be developed in order to produce wires of desired qualities in terms of consistancy in diameter over long lengths (for normal wire), or uniform change in area of the wire with a given rate (for tapered wire).

A theoretical investigation of the effect of the drawing speed and the temperature of the polymer melt within the DRU on the performance of the dieless wire drawing process must be undertaken in order to provide a greater understanding of the process. To further investigate the process, it is necessary to carry out an extensive range of experiments, in which parameters such as drawing speed, the temperature of polymer melt and polymers are varied. The results from the analysis and the experiments will be essential for designing a control strategy for the process.

A microcomputer-based control system is needed to ensure the required product qualities. Such a system can be made very flexible, and programmed to control the drawing process automatically. The operator inputs the specification via a keyboard, following a series of prompts on the computer screen. The control scheme is designed to produce the desired products with very high accuracy. Since the controlled variable

is the reduction in area of the wire, a transducer which can measure the reduction in area of the wire during the drawing process is required for a closed-loop control system.

A digital PID controller could be used to implement the control system with parameters tuned according to existing methods. However, because the majority of controllers used in industry are PID and because some disadvantages are associated with conventional PID tuning techniques, providing a procedure to automatically generate PID tuning parameters instead of the usual trial-and-error procedure is a problem whose importance in engineering practice may not be overestimated. An adaptive algorithm is required, and must be directed towards providing the field engineers with the capability to meet the needs of increasingly stringent control objectives. This is evidenced by the recent release of some PID self-tuning controllers in the market. Shortcomings in these designs have prompted the study of alternative methods of designing a self-tuning PID controller. In this work, a new PID self-tuning controller for a deterministic system, based on the generalised self-tuning controller with pole assignment, is proposed.

The objectives of the present work are therefore as follows:

1. To provide an improved theoretical analysis of the dieless wire drawing process and to investigate its performance by extensive practical tests. Upon the analysis and experimental results, a control strategy can be established.
2. To build a microcomputer-based control system for the dieless wire drawing process and to examine the product qualities from the new drawing process.
3. To provide a new adaptive controller which may also be used for tuning digital PID controllers.

CHAPTER 2 : Analysis of the Wire Drawing Process Using a Dieless Reduction Unit

2.1 - Introduction

2.2 - Review of the Previous Analyses

2.3 - Theoretical Analysis

2.3.1 - Determination of the Pressure in the Unit and the Axial Stress in the Wire Prior to Deformation of the Wire

2.3.2 - Prediction of the Position of Yield of the Wire Inside the Unit

2.3.3 - Energy Equation Applied to the Process

2.3.4 - Prediction of Deformation Profile in the Deformation Zone

2.4 - Results and Discussion

Notation:

L_1 Length of the first section of the DRU

L_2 Length of the second section of the DRU

h_1 Gap between the undeformed wire and the bore in the first section of the DRU

h_2 Gap between the undeformed wire and the bore in the second section of the DRU

h Gap between the deformed wire and the bore in the first section of the DRU

P Pressure at any point within the DRU

X Distance along the length of the DRU

μ Viscosity of the polymer melt

μ_0 Initial Viscosity of the polymer melt

μ^* Viscosity in the reduced Reynolds equation

a Viscosity temperature dependency constant

T Temperature variation of the polymer melt from the reference temperature

Y Distance from the wire surface into the gap

R_1 Initial radius of the wire

b Factor determining theoretical profile

D_1 Initial diameter of the wire

D Diameter of the wire

τ Shear stress in the melt

u Polymer melt velocity

U_0 Drawing speed

n Strain hardening index

K Thermal conductivity of polymer melt

τ_x Shear stress on wire surface before deformation

τ_c Shear stress on deformed wire surface

R_1 Length ratio L_1 / L_2

R_h Gap ratio h_1 / h_2

Q_x Flow of polymer melt

P_m Pressure at the step

X_0 Yielding position of wire

A Strain hardening constant

$\overline{Y}$ Yield Stress in wire

Y_0 Initial yield stress in the wire

σ_x Axial stress in the wire

J_0, J_1, J_2, J'_0, J'_1, J'_2, A_1, B, A_2, I_0, I_1, I_2 Computation variables

2.1 - Introduction

This chapter presents a theoretical analysis of the dieless wire drawing process, i.e., a static model of the process rather than a dynamic mathematical model of the process in control terminology. In this model, based upon mechanical principles, investigation of the effect of drawing speed and the temperature of the polymer melt within the DRU on the performance of the process are undertaken. The results of the analysis are essential for designing a control strategy for the process. The development of the dynamic mathematical models and, further, the Laplace transfer functions of the process, will be presented in chapter 3.

2.2 - Review of the Previous Analyses

Since this novel technique of wire drawing was invented by Dr. G. R. Symmons and Dr. M. S. J. Hashmi, in which no conventional reduction dies were used and polymer melts were introduced as the pressure medium in drawing process, various theories concerning the dieless wire drawing process have been developed.

Dr. Symmons and Dr. Hashmi first developed a Newtonian plasto-hydrodynamic analysis for a stepped bore reducton unit. In this the pressure medium of the polymer melt was assumed to have a Newtonian fluid behaviour and the deformation profile of the wire within the unit was assumed to be a function of distance squared [6]. Theoretical results enabled them to predict the percentage reduction in area, the pressure of the polymer melt within the unit and the axial stress in the wire. Although these were in close agreement with those observed experimentally at lower drawing speeds, the theory was found to be inadequate at drawing speeds in excess of about 1.2 ms^{-1}. For example,

experimental results showed that the percentage reduction in area increased with an increase in drawing speed, and reached a maximum magnitude of about 25 percent corresponding to a drawing speed of 1.2 ms^{-1}. Thereafter, the percentage reduction in area decreased when drawing speed was increased further. However, unlike the experimentally observed results, the theoretically calculated reduction in area always kept on increasing with an increase in drawing speed.

Later, Dr. Hashmi and Dr. Symmons presented a mathematical model for wire drawing through a conical orifice reduction unit filled with polymer melt, in which the polymer melt was assumed to be a Newtonian fluid and the deformation of the wire was assumed to take place in a linear profile [8]. A further development [9] presented a more precise numerical solution for wire drawing through a conical orifice unit. In their studies, the viscous fluid was assumed to be Newtonian and the continuum was assumed to be rigid non-linearly strain hardening. Finite difference numerical techniques were applied to solve the equations for the plasto- hydrodynamic pressure and the resulting axial stress which, in turn, enabled prediction of the non-linear deformation profile of the continuum and the reduction in area for a given drawing speed. Unlike the previous analyses, the need for an assumed deformation profile was eliminated. This analysis produced good correlation with experimental results at lower drawing speeds. However, the agreement between the experimental and theoretical results was still poor at higher drawing speeds. Several factors might contribute to the discrepancy between the theoretical and experimental results. Mainly, the polymer melt did not behave like a Newtonian fluid.

Dr. H. Parvinmehr presented a Non-Newtonian plasto-hydrodynamic analysis of the wire drawing process for a stepped bore reduction unit [7]. The effects of shear rate and pressure on the viscosity of the

polymer, together with the limiting shear stress, were included in the analysis. Strain hardening and strain rate sensitivity of the wire material were incorporated in the solution. Finite-difference numerical techniques were used to solve the relevant equations. The Non-Newtonian results showed a generally close agreement compared with the experimental results. However, the curve indicating the changes of percentage reduction in area versus drawing speed appeared to have a point at which the direction of the curve changed abruptly; i.e., a cusp appeared in the curve, which did not appear in the experimental results.

In the present study, a Non-Newtonian plasto-hydrodynamic analysis of the wire drawing process is presented for a stepped bore reduction unit. The effect of temperature on the viscosity of the polymer melt and the energy equation predicting the increasing in temperature during the wire drawing process are introduced into the analysis.

The present study predicts the percentage reduction in area, drawing load in the wire and the pressure of the polymer melt within the unit against drawing speed. The theoretical results agree well with those observed experimentally throughout the range of drawing speeds.

2.3 - Theoretical Analysis

When wire is pulled through the viscous polymer melt within the DRU, shearing takes place at the interface between the melt and the wire. This shearing action gives rise to a drag force on the wire and generates hydrodynamic pressure, the magnitude of which depends upon the speed with which the wire is pulled, the viscosity of the polymer melt and the geometrical configuration of the orifice which contains the melt. The combined effect of this hydrodynamic pressure and the drag force may initiate plastic yielding and cause permanent deformation to the wire itself.

During the wire drawing process, energy is dissipated mainly in two ways: through the action of viscous "friction" and through deformation. These sources will generate heat. Because the heat conductivity of the polymer melt used in wire drawing process is substantially low, the small amount of heat can not be transferred outside and will cause the temperature within the polymer to increase significantly. The change of the temperature, in turn, changes the viscocity of the polymer melt which, in turn, changes the dissipation as well as the shear stress, the pressure and thus the reduction in area. So we must indeed consider the thermal energy coupled with the plasto-hydrodynamic process.

In this section an attempt to analyse theoretically the deformation process taking place within the DRU is described. The effect of the thermal energy generated by viscous "friction" in the drawing process will also be taken into consideration. The results of the analysis tend to be very similar to the experimental results both at slower speeds and at higher speeds, and show better agreement than previous results.

The following theoretical analysis of deformation process which takes place within the DRU is based on the geometrical configuration of the stepped bore reduction unit as shown in Fig. 2.1a. To formulate the analysis the following assumptions were made:

i) Flow of polymer melt is axial and laminar

ii) Thickness of the polymer layer is small compared with the dimensions of the stepped bore

iii) The pressure in the fluid is uniform in the thickness direction at any point along the length of the unit, that is $P = P(X)$ only.

iv) The viscosity-temperature dependence is given by the equation (Appendix 1),

$$\mu = \mu_0 e^{-aT} \qquad (2.1)$$

where, μ is the viscosity of the polymer melt

μ_0 is the melt viscosity for a given polymer at reference temperature

"a" is a constant indicating the temperature dependency of the melt viscosity for a given polymer

T is the temperature variation of the polymer melt from the reference temperature

v) The deformation profile in the deformation zone within the unit may be described by

$Y = R_1 - bX^2$ (see Fig. 2.1a)

Where $R_1 = D_1/2$ the initial radius of the wire.

vi) The temperature in the fluid is constant along the drawing direction, but varies across the section of the polymer, that is $T = T(Y)$ only.

The analysis is presented in four parts:

2.3.1 - Determination of the Pressure in the Unit and the Axial Stress in the Wire Prior to Deformation of the Wire.

2.3.2 - Prediction of the Position of Yield of the Wire Inside the Unit.

2.3.3 - Energy Equation Applied to the Process.

2.3.4 - Prediction of Deformation Profile in the Deformation zone.

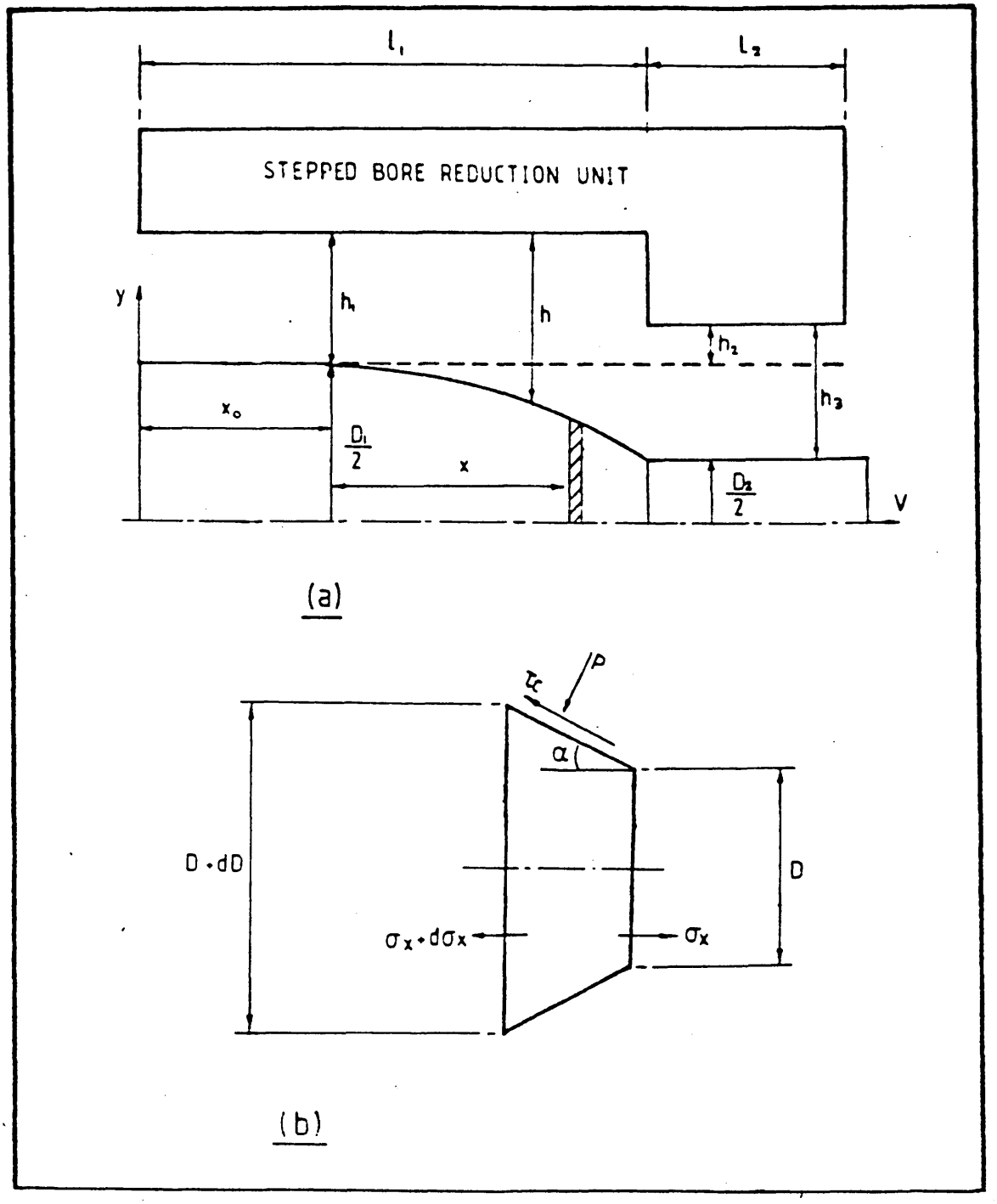


Fig. 2.1 a) Theoretically Assumed Deformation Mode Within the DRU
b) Stresses Action on an Element of the Wire

2.3.1 - Determination of the Pressure in the Unit and the Axial Stress in the Wire Prior to Deformation of the Wire

The pressure in the unit prior to deformation of the wire

Starting with the fundamental equations

$$\frac{\partial P}{\partial X} = \frac{\partial \tau}{\partial Y} \tag{2.2}$$

and

$$\tau = \mu \frac{\partial u}{\partial Y} \tag{2.3}$$

Where $\frac{\partial P}{\partial X}$ is the pressure gradient within the gap

τ is the shear stress

u is the velocity of the polymer melt at a distance Y from the surface of the wire within the gap

μ is the viscosity of the polymer melt

Substituting for τ from equation (2.3) in equation (2.2) gives

$$\frac{\partial P}{\partial X} = \frac{\partial}{\partial Y} \left(\mu \frac{\partial u}{\partial Y}\right) \tag{2.4}$$

Integrating equation (2.4) with respect to Y, we have

$$\frac{\partial u}{\partial Y} = \frac{1}{\mu} \frac{\partial P}{\partial X} Y + \frac{C_1}{\mu} \tag{2.5}$$

Noting that P and $\frac{\partial P}{\partial X}$ are assumed to remain constant across the gap, and since $T = T(Y)$, from equation (2.1) it can be seen that μ is the function of Y. Integrating equation (2.5) gives

$$u = \frac{\partial P}{\partial X} \int_o^Y \frac{Y'}{\mu(Y')} dY' + C_1 \int_o^Y \frac{dY'}{\mu(Y')} dY + C_2 \tag{2.6}$$

Applying the boundary condition $u = U_0$, at $Y = 0$ gives

$$C_2 = U_0$$

Where U_0 is the velocity with which the wire is being pulled.

The flow of polymer melt in the stepped bore reduction unit is divided into two sections, the first section being the entry part of the

unit before the step and the second section being the tube after the step. Applying the boundary condition for the first part that u = 0 at $Y = h_1$, and substituting for C_2 in equation (2.6), we have

$$\left(\frac{\partial X}{\partial P}\right)_1 \int_o^{h_1} \frac{Y'}{\mu(Y')}\, dY' + C_1 \int_o^{h_1} \frac{dY'}{\mu(Y')} + U_0 = 0 \qquad (2.7)$$

Define

$$J_0 = \int_o^{h_1} \frac{dY'}{\mu(Y')} \qquad (2.8)$$

and

$$J_1 = \int_o^{h_1} \frac{Y'}{\mu(Y')}\, dY' \qquad (2.9)$$

Then from equation (2.7), we have

$$C_1 = - \frac{U_0 + \left(\frac{\partial P}{\partial X}\right)_1 J_1}{J_0} \qquad (2.10)$$

which we assume to be known.

The flow of the polymer melt in the direction of pull in the first section of the unit is given by

$$Q_{X_1} = \int_o^{h_1} u\, dY \qquad (2.11)$$

Substituting for u from equation (2.6) in equation (2.11) gives

$$Q_{X_1} = \left(\frac{\partial P}{\partial X}\right)_1 \int_o^{h_1} \int_o^{Y} \frac{Y'}{\mu(Y')}\, dY'dY + C_1 \int_o^{h_1} \int_o^{Y} \frac{dY'dY}{\mu(Y')} + \int_o^{h_1} U_0 dY \qquad (2.12)$$

Referring to Appendix 2, the first term of equation (2.12)

$$\left(\frac{\partial P}{\partial X}\right)_1 \int_o^{h_1} \int_o^{Y} \frac{Y'}{\mu(Y')}\, dY'dY = \left(\frac{\partial P}{\partial X}\right)_1 \int_o^{h_1} \int_{Y'}^{h_1} \frac{Y'}{\mu(Y')}\, dYdY'$$

$$= \left(\frac{\partial P}{\partial X}\right)_1 \int_o^{h_1} (h_1 - Y') \frac{Y'}{\mu(Y')}\, dY'$$

$$= \left(\frac{\partial P}{\partial X}\right)_1 (h_1 J_1 - J_2) \qquad (2.13)$$

Where $$J_2 = \int_o^{h_1} \frac{Y^2}{\mu(Y)}\, dY \qquad (2.14)$$

The second term

$$C_1 \int_o^{h_1} \int_o^{Y} \frac{dY'}{\mu(Y')}\, dY = C_1 \int_o^{h_1} \int_{Y'}^{h_1} \frac{dY'}{\mu(Y')}\, dY'$$

$$= C_1 \int_o^{h_1} \frac{(h_1 - Y')}{\mu(Y')} dY$$

$$= C_1 (h_1 J_0 - J_1) \qquad (2.15)$$

The third term

$$\int_o^{h_1} U_0 dY = U_0 h_1 \qquad (2.16)$$

Equation (2.12) in combination with equations (2.13) to (2.16) gives

$$Q_{X1} = \left(\frac{\partial P}{\partial X}\right)_1 (h_1 J_1 - J_2) + C_1 (h_1 J_0 - J_1) + U_0 h_1 \qquad (2.17)$$

The continuity equation for this case gives

$$\frac{\partial}{\partial X} Q_{X1} = 0 \qquad (2.18)$$

Substituting for Q_{X1} in equation (2.17), we have

$$\frac{\partial}{\partial X}\left[\left(\frac{\partial P}{\partial X}\right)_1 (h_1 J_1 - J_2) + C_1 (h_1 J_0 - J_1) + U_0 h_1\right] = 0 \qquad (2.19)$$

Substituting for C_1 from equation (2.10) in equation (2.19) and rearranging gives

$$\left(\frac{\partial^2 P}{\partial X^2}\right)_1 \left(\frac{J_1^2}{J_0} - J_2\right) = 0 \tag{2.20}$$

So we have

$$\left(\frac{\partial^2 P}{\partial X^2}\right)_1 = 0 \tag{2.21}$$

that is

$$\left(\frac{\partial P}{\partial X}\right)_1 = \text{Constant} \tag{2.22}$$

Assuming that at the point where $X = L_1$, pressure reaches its maximum value P_m, we have

$$\left(\frac{\partial P}{\partial X}\right)_1 = \frac{P_m}{L_1} \tag{2.23}$$

Following the same procedure mentioned above, we have the following equations for the second section of the unit

$$Q_{X2} = \left(\frac{\partial P}{\partial X}\right)_2 (h_2 J_1' - J_2') + C_1'(h_2 J_0' - J_1') + U_0 h_2 \tag{2.24}$$

$$J_0' = \int_0^{h_2} \frac{1}{\mu(Y')}\, dY' \tag{2.25}$$

$$J_1' = \int_0^{h_2} \frac{Y'}{\mu(Y')}\, dY' \tag{2.26}$$

$$J_2' = \int_0^{h_2} \frac{Y'^2}{\mu(Y')}\, dY' \tag{2.27}$$

$$\left(\frac{\partial P}{\partial X}\right)_2 = -\frac{P_m}{L_2} \tag{2.28}$$

The continuity of flow gives

$$Q_{X1} = Q_{X2} \tag{2.29}$$

Substituting for Q_{X1}, Q_{X2}, $\left(\frac{\partial P}{\partial X}\right)_1$ and $\left(\frac{\partial P}{\partial X}\right)_2$ from equations (2.17), (2.24), (2.23) and (2.28) respectively in equation (2.29) and rearranging gives

$$P_m = \frac{U_0 L_1 L_2 (J_1 J_0' - J_1' J_0)}{-L_1 J_0 (J_1'^2 - J_2' J_0') - L_2 J_0' (J_1^2 - J_2 J_0)} \tag{2.30}$$

Axial stress in the wire

Substituting for $\frac{\partial u}{\partial Y}$ from equation (2.5) in equation (2.3) gives

$$\tau = \frac{\partial P}{\partial X} Y + C_1 \tag{2.31}$$

On the surface of the wire in the first section of the wire, where Y = 0, the shear stress is given by

$$\tau_X = C_1 = - \frac{U_0 + \frac{P_m}{L_1} J_1}{J_0} \tag{2.32}$$

This shear stress gives rise to drag forces in the wire and at a point X from the entry point within the unit in the first section, the axial stress developed in the wire is given by

$$\sigma(X) = \frac{4}{D_1} \int_0^X \tau_X dX \tag{2.33}$$

Integrating equation (2.33) gives

$$\sigma(X) = \frac{4X\tau_X}{D_1} \tag{2.34}$$

2.3.2 - Prediction of the Position of Yield of the Wire Inside the Unit

The axial distance, X_0, from the entry point of the unit where yield occurs in the wire may be determined by the combined effects of the axial stress and the hydrodynamic pressure.

The pressure at point X is given by

$$P(X) = \frac{P_m}{L_1} X \tag{2.35}$$

The flow stress of the wire is assumed to be expressed as

$$\overline{Y} = Y_0 + A\varepsilon^n$$

Where A and n are the strain hardening factor of the wire and the material constant respectively, $\varepsilon = 2 \ln(\frac{D_1}{D})$. Then the plastic yield will commence as soon as the yield criterion

$$P(X_0) + \sigma(X_0) = Y_0 \tag{2.36}$$

is satisfied

Substituting for $P(X_0)$ and $\sigma(X_0)$ from equations (2.34) and (2.35) respectively in equation (2.36) and rearranging gives

$$X_0 = \frac{Y_0}{\frac{P_m}{L_1} - \frac{4\tau_X}{D_1}} \tag{2.37}$$

Now J_0, J_1, J_2, J'_0, J'_1 and J'_2 are still unknown. They can be solved by application of the energy equation to the process of wire drawing.

Once plastic yield commences at point $X = X_0$, further permanent deformation will continue to take place in the wire as long as the yield criterion

$$P(X) + \sigma(X) > = \overline{Y} \tag{2.38}$$

is satisfied at any point within the first section, distance X from the yield starting point X_0.

2.3.3 - Energy Equation Applied to the Process

Determination of J_0, J_1 and J_2 in the first section of the unit

The Energy equation for the plasto-hydrodynamic process of the dieless wire drawing is given (Appendix 3) by

$$\frac{\partial^2 T}{\partial Y^2} = -\frac{\mu}{K}\left(\frac{\partial u}{\partial Y}\right)^2 \tag{2.39}$$

Where T is the temperature variation of the polymer melt from the reference temperature

K is the conductivity of the polymer melt

Here only the effect of the thermal energy generated by viscous "friction" in the drawing process is taken into consideration.

Rewriting equation (2.5)

$$\frac{\partial u}{\partial Y} = \frac{1}{\mu}\frac{\partial P}{\partial X} Y + \frac{C_1}{\mu} \tag{2.40}$$

where $$C_1 = -\frac{U_0 + \frac{\partial P}{\partial X} J_1}{J_0}$$

To simplify mathematical processing, here we assume that $\frac{\partial P}{\partial X} = 0$

hence $$\frac{\partial u}{\partial Y} = -\frac{U_0}{\mu J_0} \tag{2.41}$$

Substituting for $\frac{\partial u}{\partial Y}$ from equation (2.41) in equation (2.39) gives

$$\frac{\partial^2 T}{\partial Y^2} = -\frac{U_0^2}{K\mu J_0^2} \tag{2.42}$$

Substituting $\mu = \mu_0 e^{-aT}$ into equation (2.42), we have

$$\frac{\partial^2 T}{\partial Y^2} = -\frac{U_0^2}{K\mu_0 J_0^2} e^{aT} \tag{2.43}$$

Multiplying both sides by $\frac{\partial T}{\partial Y}$ and rewriting, we have

$$\frac{\partial}{\partial Y}\left[\frac{1}{2}\left(\frac{\partial T}{\partial Y}\right)^2\right] = -\frac{U_0^2}{K\mu_0 J_0^2 a}\frac{\partial}{\partial Y} e^{aT}$$

Integrating both sides with respect to Y gives

$$\left(\frac{\partial T}{\partial Y}\right)^2 = -\frac{2U_0^2 e^{aT}}{K\mu_0 J_0^2 a} + C_3 \tag{2.44}$$

Now integrating equation (2.42) with respect to Y between 0 and h_1 gives

$$\left.\frac{\partial T}{\partial Y}\right|_{Y=h_1} - \left.\frac{\partial T}{\partial Y}\right|_{Y=0} = -\frac{U_0^2}{KJ_0^2}\int_0^{h_1}\frac{dY}{\mu(Y)}$$

$$= -\frac{U_0^2}{KJ_0} \qquad (2.45)$$

By symmetry

$$\left.\frac{\partial T}{\partial Y}\right|_{Y=0} = -\left.\frac{\partial T}{\partial Y}\right|_{Y=h_1} \qquad (2.46)$$

Hence from equations (2.45) and (2.46), we have

$$\left.\frac{\partial T}{\partial Y}\right|_{Y=h_1} = -\frac{U_0^2}{2KJ_0} \qquad (2.47)$$

Thus

$$\left(\frac{\partial T}{\partial Y}\right)^2_{Y=h_1} = \frac{U_0^4}{4K^2J_0^2} \qquad (2.48)$$

Applying the boundary condition that $T = 0$ at $Y = h_1$ to equation (2.44), we have

$$\left(\frac{\partial T}{\partial Y}\right)^2_{Y=h_1} = -\frac{2U_0^2}{K\mu_0 J_0^2 a} + C_3 \qquad (2.49)$$

Equating equations (2.48) and (2.49) gives

$$C_3 = \frac{U_0^2}{KJ_0^2}\left[\frac{U_0^2}{4K} + \frac{2}{a\mu_0}\right]$$

Substituting for C_3 in equation (2.44) gives

$$\left(\frac{\partial T}{\partial Y}\right)^2 = \frac{2U_0^2}{aK\mu_0 J_0^2}\left[\left(1 + \frac{a\mu_0 U_0^2}{8K}\right) - e^{aT}\right]$$

(2.50)

Let $$A_1 = \frac{2U_0^2}{aK\mu_0 J_0^2} \tag{2.51}$$

$$B = 1 + \frac{a\mu_0 U_0^2}{8K} \tag{2.52}$$

Equation (2.50) becomes

$$\left(\frac{\partial T}{\partial Y}\right)^2 = A_1(B - e^{aT}) \tag{2.53}$$

Because T = T(Y) only, rearranging equation (2.53) gives

$$\frac{dT}{dY} = \sqrt{A_1(B - e^{aT})} \qquad 0 < Y < h_1/2 \tag{2.54}$$

and $$\frac{dT}{dY} = -\sqrt{A_1(B - e^{aT})} \qquad h_1/2 < Y < h_1 \tag{2.55}$$

By symmetry, at point where $Y = h_1/2$ and $\frac{dT}{dY} = 0$, T reaches its maximum value T_m. From equation (2.53), we have

$$B - e^{aT_m} = 0$$

That is $$T_m = \frac{1}{a} \ln B$$

Rearranging equation (2.54) gives

$$\frac{dT}{\sqrt{B - e^{aT}}} = \sqrt{A_1}\, dY \tag{2.56}$$

Now integrating equation (2.56) both sides gives

$$\frac{1}{a\sqrt{B}} \ln \left| \frac{\sqrt{B - e^{aT}} - \sqrt{B}}{\sqrt{B - e^{aT}} + \sqrt{B}} \right| = \sqrt{A_1}\, Y + C_4 \tag{2.57}$$

Applying the boundary condition that $T = T_m$ at $Y = h_1/2$ to equation (2.57) gives

$$C_4 = -\frac{h_1}{2}\sqrt{A_1} \tag{2.58}$$

Applying the boundary condition that T = 0 at Y = 0 to equation (2.57) gives

$$C_4 = \frac{1}{A\sqrt{B}} \ln\left|\frac{\sqrt{B-1} - \sqrt{B}}{\sqrt{B-1} + \sqrt{B}}\right| \tag{2.59}$$

Equating equation (2.58) and equation (2.59), we have

$$\frac{h_1}{2} = -\frac{1}{a\sqrt{A_1 B}} \ln\left|\frac{\sqrt{B-1} - \sqrt{B}}{\sqrt{B-1} + \sqrt{B}}\right| \tag{2.60}$$

Substituting for A_1 from equation (2.51) in equation (2.60) and rearranging gives

$$J_0 = \frac{\sqrt{\frac{BU_0^2 h_1^2 a}{2K\mu_0}}}{\ln\left|\frac{\sqrt{B-1} + \sqrt{B}}{\sqrt{B-1} - \sqrt{B}}\right|} \tag{2.61}$$

Substituting for C_4 from equation (2.59) in equation (2.57) and rearranging gives

$$Y = \frac{1}{a\sqrt{A_1 B}} \ln\left|\frac{\sqrt{B - e^{aT}} - \sqrt{B}}{\sqrt{B - e^{aT}} + \sqrt{B}}\right| - \frac{1}{a\sqrt{A_1 B}} \ln\left|\frac{\sqrt{B-1} - \sqrt{B}}{\sqrt{B-1} + \sqrt{B}}\right| \tag{2.62}$$

for $0 < Y < h_1/2$

Rearranging equation (2.55) gives

$$\frac{dT}{\sqrt{B - e^{aT}}} = -\sqrt{A_1}\, dY \tag{2.63}$$

Integrating both sides gives

$$\frac{1}{a\sqrt{B}} \ln\left|\frac{\sqrt{B - e^{aT}} - \sqrt{B}}{\sqrt{B - e^{aT}} + \sqrt{B}}\right| = -\sqrt{A_1}\, Y + C_5 \tag{2.64}$$

Applying the boundary condition that T = 0 at Y = h_1 to equation (2.64), we have

$$C_5 = \sqrt{A_1}h_1 + \frac{1}{a\sqrt{B}} \ln\left|\frac{\sqrt{B-1} - \sqrt{B}}{\sqrt{B-1} + \sqrt{B}}\right| \tag{2.65}$$

Substituting for C_5 in equation (2.64) gives

$$Y = \frac{-1}{a\sqrt{A_1 B}} \ln\left|\frac{\sqrt{B - e^{aT}} - \sqrt{B}}{\sqrt{B - e^{aT}} + \sqrt{B}}\right| + \frac{1}{a\sqrt{A_1 B}} \ln\left|\frac{\sqrt{B-1} - \sqrt{B}}{\sqrt{B-1} + \sqrt{B}}\right| + h_1 \tag{2.66}$$

Substituting for h_1 from equation (2.60) in equation (2.66) gives

$$Y = -\frac{1}{a\sqrt{A_1 B}} \ln\left|\frac{\sqrt{B - e^{aT}} - \sqrt{B}}{\sqrt{B - e^{aT}} + \sqrt{B}}\right| - \frac{1}{a\sqrt{A_1 B}} \ln\left|\frac{\sqrt{B-1} - \sqrt{B}}{\sqrt{B-1} + \sqrt{B}}\right| \tag{2.67}$$

for $h_1/2 < Y < h_1$

Rewriting equation (2.9) gives

$$J_1 = \int_0^{h_1} \frac{Y}{\mu}\, dY = \int_0^{\frac{h_1}{2}} \frac{Y}{\mu}\, dY + \int_{\frac{h_1}{2}}^{h_1} \frac{Y}{\mu}\, dY \tag{2.68}$$

Substituting for $\mu = \mu_0 e^{-aT}$, for dY from equation (2.56) and Y from equation (2.62) in the first integration term of equation (2.68); and for dY from equation (2.63) and Y from equation (2.67) in the second integration term of equation (2.68), we have

$$J_1 = \int_0^{T_m} \frac{e^{aT}}{\sqrt{A_1(B - e^{aT})}\,\mu_0} \left[\frac{1}{a\sqrt{A_1 B}} \ln\left|\frac{\sqrt{B - e^{aT}} - \sqrt{B}}{\sqrt{B - e^{aT}} + \sqrt{B}}\right| - \frac{1}{a\sqrt{A_1 B}} \ln\left|\frac{\sqrt{B-1} - \sqrt{B}}{\sqrt{B-1} + \sqrt{B}}\right|\right] dT$$

$$+ \int_{T_m}^{0} \frac{-e^{aT}}{\sqrt{A_1(B - e^{aT})}\,\mu_0} \left[\frac{-1}{a\sqrt{A_1 B}} \ln\left|\frac{\sqrt{B - e^{aT}} - \sqrt{B}}{\sqrt{B - e^{aT}} + \sqrt{B}}\right| - \frac{1}{a\sqrt{A_1 B}} \ln\left|\frac{\sqrt{B-1} - \sqrt{B}}{\sqrt{B-1} + \sqrt{B}}\right|\right] dT$$

$$= \frac{4\sqrt{B-1}}{a^2 \mu_0 A_1 \sqrt{B}} \ln\left|\frac{\sqrt{B-1} + \sqrt{B}}{\sqrt{B-1} - \sqrt{B}}\right| \tag{2.69}$$

Rewriting equation (2.14) gives

$$J_2 = \int_0^{h_1} \frac{Y^2}{\mu}\, dY = \int_0^{\frac{h_1}{2}} \frac{Y^2}{\mu}\, dY + \int_{\frac{h_1}{2}}^{h_1} \frac{Y^2}{\mu}\, dY \tag{2.70}$$

Following the same procedure as above, we have

$$J_2 = 2\int_0^{T_m} \frac{e^{aT}}{\sqrt{A_1(B-e^{aT})}\ \mu_0 a^2 A_1 B} \left[\ln\left|\frac{\sqrt{B-e^{aT}} - \sqrt{B}}{\sqrt{B-e^{aT}} + \sqrt{B}}\right|\right]^2 dT$$

$$+ \frac{4\sqrt{B-1}}{a^3 A_1 B\sqrt{A_1}\ \mu_0} \left[\ln\left|\frac{\sqrt{B-1} - \sqrt{B}}{\sqrt{B-1} + \sqrt{B}}\right|\right]^2 \tag{2.71}$$

The first term on the right side of equation (2.71) could be calculated by using a numerical integration method.

Determination of J'_0, J'_1 and J'_2 in the second section of the unit

Now similarly, for the second section of the unit, we have

$$J'_0 = \frac{\sqrt{\frac{BU_0^2 h_2^2 a}{2K\mu_0}}}{\ln\left|\frac{\sqrt{B-1} + \sqrt{B}}{\sqrt{B-1} - \sqrt{B}}\right|} \tag{2.72}$$

$$A_2 = \frac{2U_0^2}{aK\mu_0 J'^2_0} \tag{2.73}$$

$$J'_1 = \frac{4\sqrt{B-1}}{a^2\mu_0 A_2\sqrt{B}} \ln\left|\frac{\sqrt{B-1} + \sqrt{B}}{\sqrt{B-1} - \sqrt{B}}\right| \tag{2.74}$$

$$J_2' = 2\int_0^{T_m} \frac{e^{aT}}{\sqrt{A_2(B-e^{aT})\ \mu_0 a^2 A_2 B}} \left[\ln\left|\frac{\sqrt{B-e^{aT}} - \sqrt{B}}{\sqrt{B-e^{aT}} + \sqrt{B}}\right|\right]^2 dT$$

$$+ \frac{4\sqrt{B-1}}{a^3 A_2 B\sqrt{A_2}\ \mu_0} \left[\ln\left|\frac{\sqrt{B-1} - \sqrt{B}}{\sqrt{B-1} + \sqrt{B}}\right|\right]^2 \qquad (2.75)$$

2.3.4 - Prediction of Deformation Profile in the Deformation Zone

The reduced Reynolds equation

It is assumed that deformation takes place from the point of yield where $X = X_0$ and ceases at the step. In the deformation zone, rewriting equation (2.19) with gap h, we have

$$\frac{\partial}{\partial X}\left[\frac{\partial P}{\partial X}\ (h\ J_1 - J_2) + C_1(h\ J_0 - J_1) + U_0 h\ \right] = 0 \qquad (2.76)$$

where

$$J_0 = \int_o^h \frac{dY}{\mu(Y)}$$

$$J_1 = \int_o^h \frac{Y}{\mu(Y)}\ dY$$

$$J_2 = \int_o^h \frac{Y^2}{\mu(Y)}\ dY$$

Substituting for C_1 from equation (2.10) in equation (2.76) and rearranging gives

$$\frac{\partial}{\partial X}\left[\left(\frac{J_1^2 - J_2 J_0}{J_0}\right)\frac{\partial P}{\partial X}\right] = - U_0 \frac{\partial}{\partial X}\left(\frac{J_1}{J_0}\right) \qquad (2.77)$$

Let $Z = Y/h$, then

$$J_0 = \int_0^h \frac{dY}{\mu(Y)} = \int_0^1 \frac{h}{\mu(Z)}\ dZ = h\ I_0 \qquad (2.78)$$

where

$$I_0 = \int_0^1 \frac{dZ}{\mu(Z)} \tag{2.79}$$

Following the same procedure, we have

$$J_1 = \int_o^h \frac{Y}{\mu(Y)}\, dY = h^2 I_1 \tag{2.80}$$

$$J_2 = \int_0^h \frac{Y^2}{\mu(Y)}\, dY = h^3 I_2 \tag{2.81}$$

where

$$I_1 = \int_0^1 \frac{Z}{\mu(Z)}\, dZ \tag{2.82}$$

$$I_2 = \int_0^1 \frac{Z^2}{\mu(Z)}\, dZ \tag{2.83}$$

Substituting for J_0, J_1 and J_2 in equation (2.77) gives

$$\frac{\partial}{\partial X}\left[6h^3 \frac{(I_1^2 - I_2 I_0)}{-I_1} \frac{\partial P}{\partial X}\right] = 6U_0 \frac{\partial h}{\partial X} \tag{2.84}$$

Then define

$$\mu^* = \frac{6(I_1^2 - I_2 I_0)}{-I_1} \tag{2.85}$$

Then, equation (2.84) becomes

$$\frac{\partial}{\partial X}\left[\mu^* h^3 \frac{\partial P}{\partial X}\right] = 6U_0 \frac{\partial h}{\partial X} \tag{2.86}$$

which is the reduced Reynolds equation with variable viscosity μ^*.

After deformation takes place, the gap between the surface of the wire and the orifice within the unit will no longer hold constant. Since $I_n = I_n(h)$, (n=0,1,2) and $h = h(X)$, we have $\mu^* = \mu^*(h) = \mu^*(X)$. Now it is assumed that in deformation zone, the values of J_n and I_n are the

same as those before deformation occurs. With this assumption it follows that μ^* will have a constant value at a given drawing velocity before and after deformation occurs and thus its value can be easily calculated from equation (2.85). When deformation is within a limited range, this assumption can be accepted.

Determination of pressure in the deformation zone

With the assumption that μ^* is constant along the length of the unit at a given drawing velocity, rewriting equation (2.86), we have

$$\frac{\partial}{\partial X}\left(h^3 \frac{\partial P}{\partial X}\right) = 6U_0\mu^* \frac{\partial h}{\partial X} \tag{2.87}$$

where X is the distance from the point of yield, h is the gap at any point X in the deformation zone

$$h = h_1 + bX^2 \tag{2.88}$$

It is assumed that deformation takes place from the point of yield and ceases at the step, where $X = L_1 - X_0$

From equation (2.88)

$$\frac{\partial h}{\partial X} = 2bX \tag{2.89}$$

Substituting for $\frac{\partial h}{\partial X}$ in equation (2.87) and integrating, we have

$$h^3\frac{\partial P}{\partial X} = 6\mu^*U_0bX^2 + C_6 \tag{2.90}$$

Applying the boundary condition that $\left(\frac{\partial P}{\partial X}\right)_1 = \frac{P_m}{L_1}$, at X=0, where $h = h_1$, the constant C_6 is given by

$$C_6 = \frac{h_1^3 P_m}{L_1} \tag{2.91}$$

Substituting for h and C_6 in equation (2.90), we have

$$\frac{\partial P}{\partial X} = 6\mu^*U_0 \frac{bX^2}{(h_1 + bX^2)^3} + \frac{P_m}{L_1}\frac{h_1^3}{(h_1 + bX^2)^3} \tag{2.92}$$

Let $m^2 = \frac{h_1}{b}$

Integrating equation (2.92) gives

$$P = 6\mu^* U_0 \left[\frac{\tan^{-1}\frac{X}{m}}{8m^3b^2} + \frac{X^3 - m^2X}{8m^2b^2(m^2+X^2)^2}\right]$$

$$+ \frac{P_m h_1^3}{L_1 b^3}\left[\frac{3\tan^{-1}\frac{X}{m}}{8m^5} + \frac{5m^2X + 3X^3}{8m^4(m^2 + X^2)^2}\right] + C_7 \qquad (2.93)$$

Applying the boundary condition that at X = 0, where plastic deformation starts, $P = \frac{P_m X_0}{L_1}$, we have

$$C_7 = \frac{P_m X_0}{L_1} \qquad (2.94)$$

So equation (2.93) becomes

$$P = 6\mu^* U_0 \left[\frac{\tan^{-1}\frac{X}{m}}{8m^3b^2} + \frac{X^3 - m^2X}{8m^2b^2(m^2 + X^2)^2}\right]$$

$$+ \frac{P_m}{L_1}\left[X_0 + \frac{3m\tan^{-1}\frac{X}{m}}{8} + \frac{5m^4X + 3m^2X^3}{8(m^2 + X^2)^2}\right] \qquad (2.95)$$

which gives the pressure distribution along the length ($L_1 - X_0$) within the DRU and up to the step.

Determination of axial stress in the deformation zone

In the deformation zone, the expression of $\frac{\partial u}{\partial Y}$ may be given by equation (2.5) by simply replacing μ^* for μ, thus

$$\frac{\partial u}{\partial Y} = \frac{1}{\mu^*}\frac{\partial P}{\partial X}Y + \frac{C_8}{\mu^*} \qquad (2.96)$$

Integrating both sides of equation (2.96) with respect to Y gives

$$u = \frac{1}{\mu^*}\frac{\partial P}{\partial X}\frac{Y^2}{2} + \frac{C_8}{\mu^*}Y + C_9 \qquad (2.97)$$

Applying the boundary condition that $u = U_0$, at $Y = 0$ and $u = 0$ at $Y = h$, we have

$$C_8 = -\frac{\partial P}{\partial X}\frac{h}{2} - \frac{U_0\mu^*}{h}$$

$$C_9 = U_0$$

Substituting for C_8 into equation (2.96) gives

$$\frac{\partial u}{\partial Y} = \frac{1}{\mu^*}\frac{\partial P}{\partial X}\left(Y - \frac{h}{2}\right) - \frac{U_0}{h} \tag{2.98}$$

The shear stress on the deformed wire surface may be expressed by

$$\tau_c = \mu^* \left.\frac{\partial u}{\partial Y}\right|_{Y=0} \tag{2.99}$$

The combination of equation (2.98) and equation (2.99) gives

$$\tau_c = -\frac{h}{2}\left(\frac{\partial P}{\partial X}\right) - \frac{\mu^* U_0}{h} \tag{2.100}$$

Now substituting for $\frac{\partial P}{\partial X}$ from equation (2.92), we have

$$\tau_c = -\frac{\mu^* U_0 (m^2 + 4X^2)}{b(m^2 + X^2)^2} - \frac{P_m h_1 m^4}{2L_1(m^2 + X^2)^2} \tag{2.101}$$

The axial force equilibrium condition for a small element of the wire shown in Fig. 2.1b gives

$$d\sigma_X = -2\frac{dD}{D}(Y + \tau_c \cot\alpha) \tag{2.102}$$

But $D = D_1 - 2bX^2$, hence $dD = -4bX\,dX$, $\tan\alpha = \frac{dD}{2dX} = -2bX$ and $\cot\alpha = -\frac{1}{2bX}$. Substituting for dD, cot α and τ_c in equation (2.102), we obtain

$$d\sigma = \frac{8bXY\,dX}{(D_1 - 2bX)} + \frac{4\mu^* U_0 (m^2 + 4X^2)\,dX}{b(D_1 - 2bX^2)(m^2 + X^2)^2} + \frac{2P_m h_1 m^4 dX}{L_1(m^2 + X^2)^2(D_1 - 2bX^2)} \tag{2.103}$$

As mentioned before the stress-strain relationship of the wire material has been shown to take the form

$$\bar{Y} = Y_0 + A\varepsilon^n$$

where

$$\varepsilon = 2 \ln\left(\frac{D_1}{D_1 - 2bX^2}\right)$$

Therefore

$$\bar{Y} = Y_0 + A\left[2 \ln \frac{D_1}{D_1 - 2bX^2}\right]^n \tag{2.104}$$

Substituting for $\bar{Y}$ into equation (2.103) gives

$$d\sigma_X = \frac{8bXY_0 dX}{D_1 - 2bX^2} + \frac{8bXA\left[2 \ln\left(\frac{D_1}{D_1 - 2bX^2}\right)\right]^2 dX}{D_1 - 2bX^2}$$

$$+ \frac{4\mu^* U_0 (m^2 + 4X^2) dX}{b(D_1 - 2bX^2)(m^2 + X^2)^2} + \frac{2P_m h_1 m^4 dX}{L_1 (m^2 + X^2)^2 (D_1 - 2bX^2)} \tag{2.105}$$

Integrating equation (2.105), the expression for the axial stress is obtained as

$$\sigma_X = -2Y_0 \ln(D_1 - 2bX^2) - \frac{2A}{n+1}\left[2 \ln(D_1 - 2bX^2)\right]^{n+1}$$

$$- \frac{4\mu^* U_0}{b}\left[\frac{4b(2D_1 + m^2 b)}{\sqrt{2bD_1}(D_1 + 2bm^2)^2} \ln\left(\frac{\sqrt{D_1} + X\sqrt{2b}}{\sqrt{D_1 + 2bX^2}}\right)\right.$$

$$\left. + \frac{5D_1 - 2bm^2}{2m(D_1 + 2bm^2)^2} \tan^{-1}\frac{X}{m} - \frac{3X}{2(D_1 + 2bm^2)(m^2 + X^2)}\right]$$

$$+ \frac{2P_m h_1}{L_1}\left[\frac{m^2 X}{2(D_1 + 2bm^2)(m^2 + X^2)} + \frac{m(D_1 + 6bm^2)}{2(D_1 + 2bm^2)^2} \tan^{-1}\frac{X}{m}\right.$$

$$\left. + \frac{4m^2 b^2}{\sqrt{2bD_1}(D_1 + 2bm^2)^2} \ln\left(\frac{\sqrt{D_1} + X\sqrt{2b}}{\sqrt{D_1} - X\sqrt{2b}}\right)\right] + C_{10} \tag{2.106}$$

Applying the boundary condition, $\sigma_X = \frac{4X_0\tau_x}{D_1}$ at $X = 0$ into equation (2.106), the constant C_{10} is obtained as

$$C_{10} = \frac{4X_0\tau_x}{D_1} + 2Y_0 \ln D_1 + \frac{2A}{n+1}[2 \ln D_1]^{n+1}$$

which upon substituting in equation (2.106) yields

$$\sigma_X = 2Y_0 \ln \frac{D_1}{D_1 - 2bX^2} + \frac{4\mu^* U_0}{b}\left[\frac{2D_1 + bm^2)\sqrt{2b}}{\sqrt{D_1}(D_1 + 2bm^2)^2} \ln \frac{\sqrt{D_1} + X\sqrt{2b}}{\sqrt{D_1} - X\sqrt{2b}}\right.$$
$$\left. + \frac{5D_1 - 2bm^2}{2m(D_1+2bm^2)^2}\tan^{-1}\frac{X}{m} - \frac{3X}{2(D_1 + 2bm^2)(m^2 + X^2)}\right]$$
$$+ \frac{2P_m h_1}{L_1}\left[\frac{m^2 X}{2(D_1 + 2bm^2)(m^2 + X^2)} + \frac{m(D_1+6bm^2)}{2(D_1+2bm^2)^2}\tan^{-1}\frac{X}{m}\right.$$
$$\left. + \frac{4m^4b^2}{\sqrt{2bD_1}(D_1 + 2bm^2)^2} \ln \frac{\sqrt{D_1} + X\sqrt{2b}}{\sqrt{D_1} - X\sqrt{2b}}\right]$$
$$+ \frac{A}{n+1}[2 \ln \frac{D_1}{D_1 - 2bX^2}]^{n+1} + \frac{4X_0}{D_1}\tau_x \qquad (2.107)$$

Deformation Profile

Applying the yield criterion again

$$P + \sigma_X = Y_0 + A[2 \ln \frac{D_1}{D_1 - 2bX^2}]^n \qquad (2.108)$$

Equation (2.108), when substitutions are made for P and σ_X from equations (2.95) and (2.107) respectively, may be iterated to establish the value of b for a given drawing velocity and DRU geometry. Once the value of b is established, the deformation profile is obtained and the reduction in area at any point for $X_0 < X < L_1$ is given by

$$PRA = [1 - (\frac{D_1 - 2bX^2}{D_1})^2] * 100$$

where the PRA is the percentage reduction in area. It is assumed that no further deformation takes place after the step within the DRU and hence the maximum reduction in area is given by

$$PRA = \{1 - [\frac{D_1 - 2b(L_1 - X_0)^2}{D_1}]^2\} * 100$$

A computer program was written to solve the relevant equations, the flowchart and listing of which are given in Appendix 4.

2.4 - Results and Discussion

Theoretical results were calculated on the basis of the equations derived in the previous analysis for copper wire. The standard values of the parameters were assumed to be as follows [7]:

Viscosity $\mu_0 = 110$ Ns/m^2

Length of the first section $L_1 = 50$ mm

Length of the second section $L_1 = 20$ mm

Initial gap $h_1 = 0.15$ mm or $h_1 = 0.2$ mm

Final gap $h_2 = 0.1$ mm or $h_2 = 0.05$ mm

Original diameter of the wire $D_1 = 0.46$ mm or $D_1 = 1.6$ mm

Initial yield stress of the wire $Y_0 = 50$ MN/m^2

Strain hardening factor of the wire $A = 440$ MN/m^2

Material constant $n = 0.52$

Viscosity temperature dependency constant $a = 0.175$ K^{-1}

Thermal conductivity $K = 0.13$ W/mK

Theoretical percentage reduction in area

Figs. 2.2 to 2.4 show the variations of the percentage reduction in area versus drawing speed predicted theoretically. For comparison purposes, some experimental results [7] are also shown in Figs. 2.2 to 2.4.

In Fig. 2.3 experimental results were obtained at the temperature of 130°C, so the equivalent theoretical results were produced by changing the initial viscosity of the polymer melt to 85 Ns/m^2 due to the effect of the reference temperature on the initial viscosity. In Figs. 2.2 and 2.3, the results predicted theoretically show very close agreement with those obtained experimentally.

In Fig. 2.4, both theoretical and experimental results were produced with the parameters of the gap ratio R_h = 4, final gap h_2 =0.05 mm and the initial diameter of the wire D_1 = 1.6 mm. The discrepancies between the theoretical and experimental results become apparent. Theoretical results appear to estimate lower percentage reduction in area at lower speeds and higher percentage reduction in area at higher speeds compared to those obtained experimentally.

Theoretical effect of changing the parameters representing the polymer melt rheology are shown in Fig. 2.5 and Fig. 2.6. The theoretical effects of changing thermal conductivity K is shown in Fig. 2.5, where the percentage reduction in area is shown to increase as K is increased. Fig. 2.6 shows the effect of the viscosity temperature dependency constant "a" on the predicted percentage reduction in area where the percentage reduction in area is shown to increase as "a" is decreased.

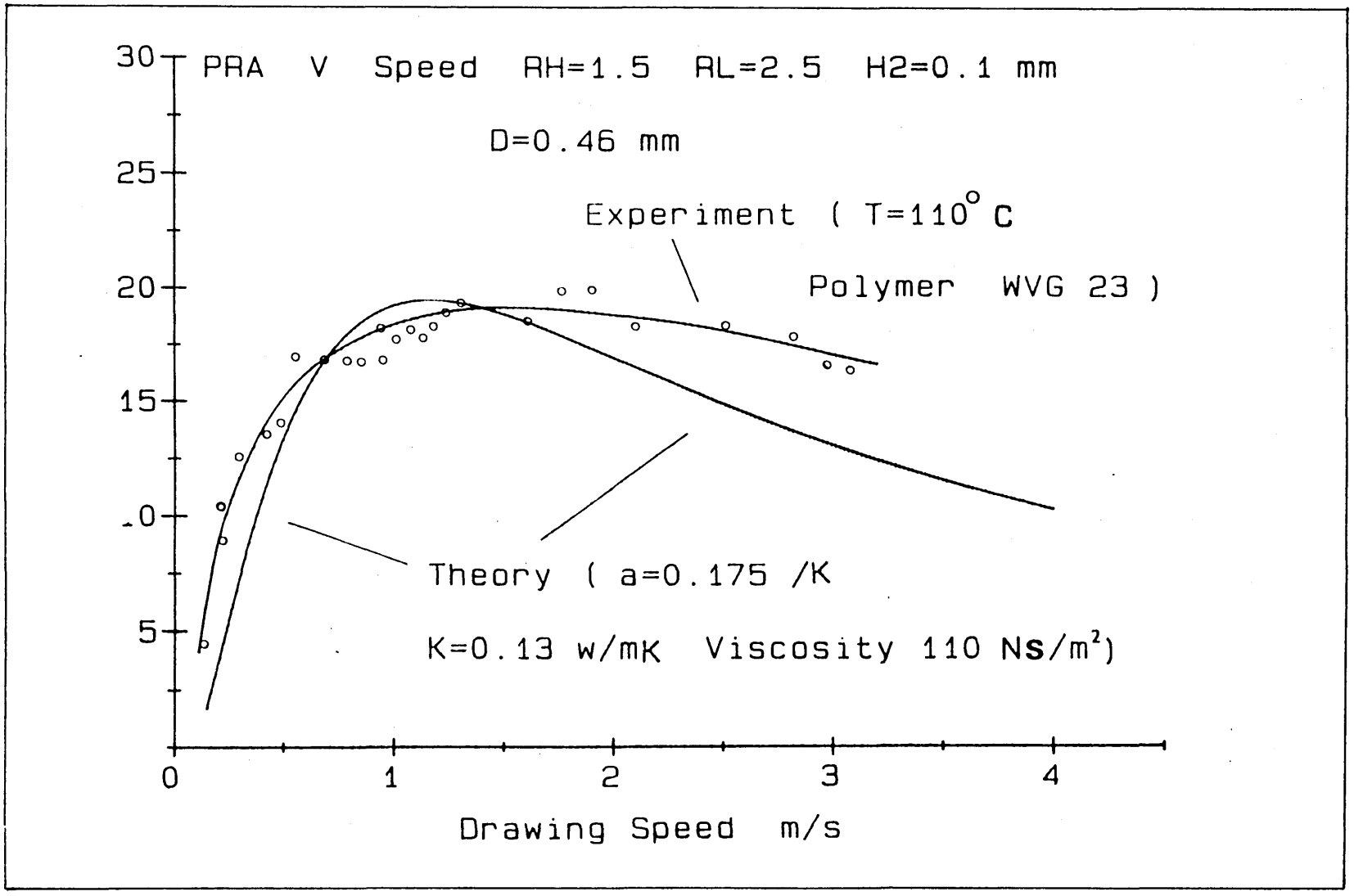


Fig. 2.2 Percentage Reduction in Area Against Drawing Speed

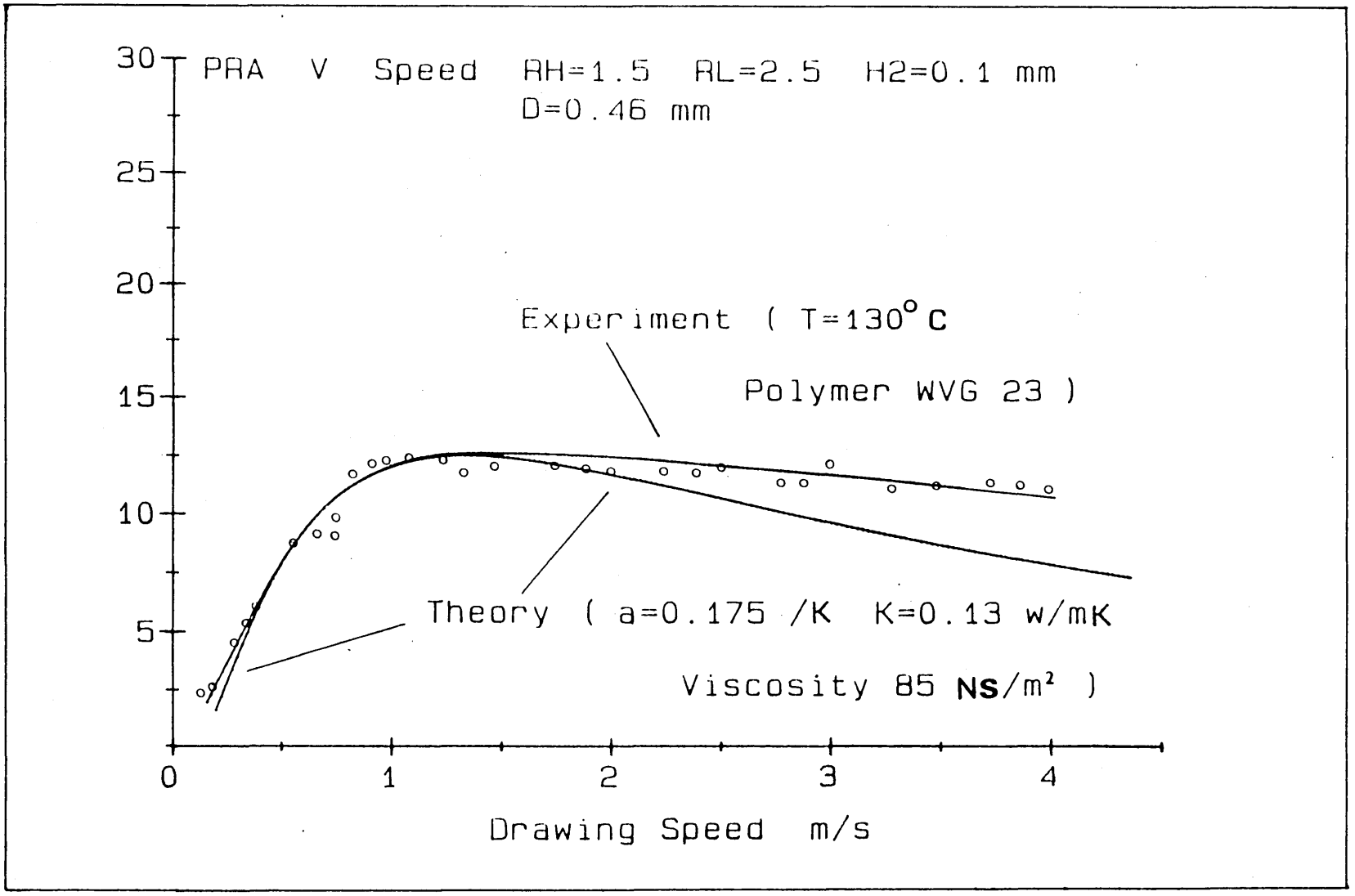


Fig. 2.3 Percentage Reduction in Area Against Drawing Speed

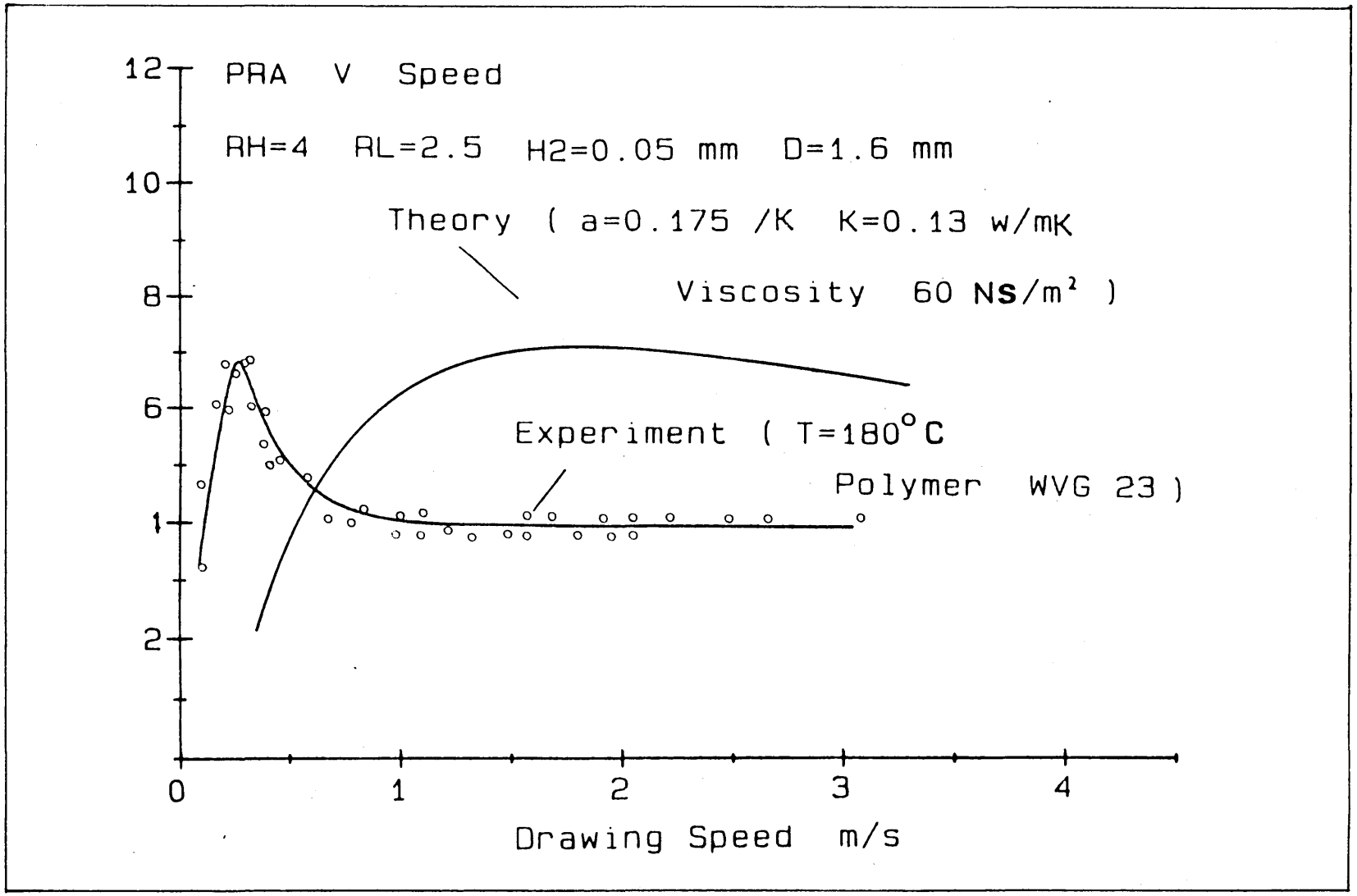


Fig. 2.4 Percentage Reduction in Area Against Drawing Speed

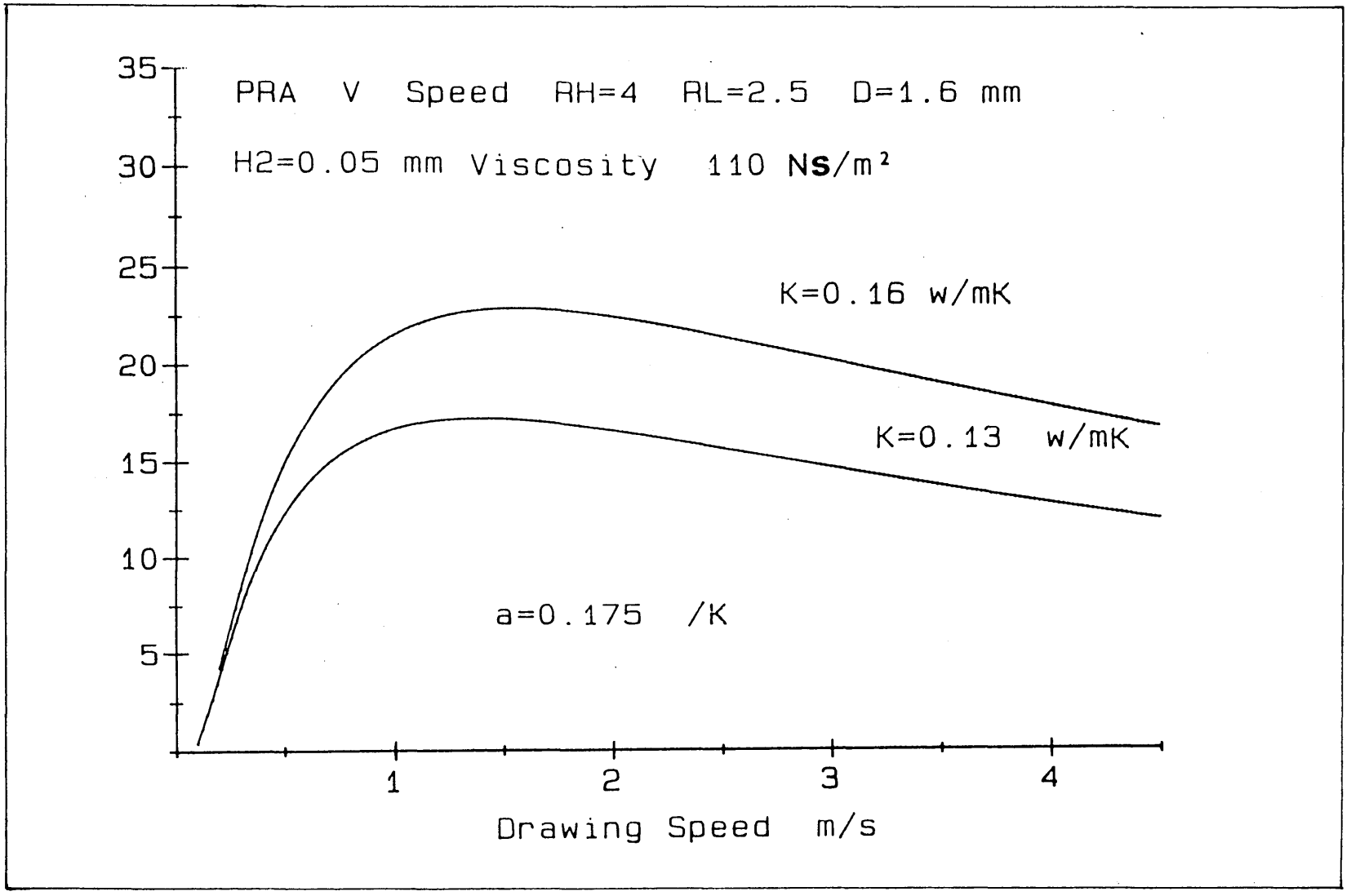


Fig. 2.5 Theoretical Effect of the Thermal Conductivity on the Predicted PRA

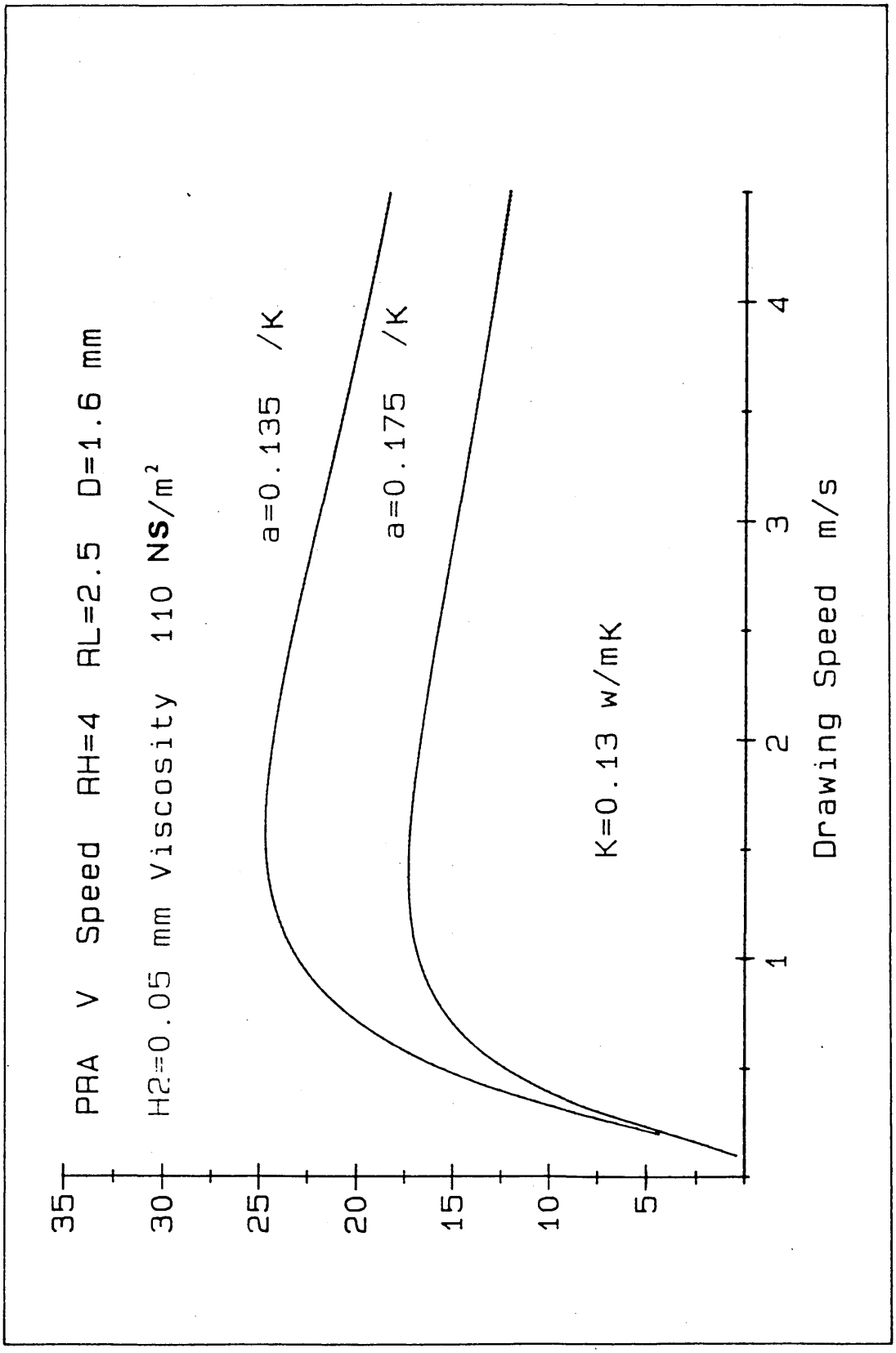


Fig. 2.6 Theoretical Effect of the Viscosity-temperature Dependancy Constant on the Predicted PRA

The pressure distributions within the unit were predicted theoretically for drawing speeds of 0.5 ms^{-1} and 1 ms^{-1} and those are shown in Figs. 2.7 and 2.8 respectively. Figs. 2.7 and 2.8 also show the pressure distributions within the unit obtained experimentally [7]. In the experiment, the maximum pressure of about 87.5 MN/m^2 occurred at drawing speed of 1 ms^{-1}, and about 77 MN/m^2 at drawing speed of 0.5 ms^{-1}, whilst the maximum pressure of about 109 MN/m^2 was predicted to occur at a speed of 1 ms^{-1} and 93 MN/m^2 at a speed of 0.5 ms^{-1}. It was clearly shown that the magnitudes of the maximum pressure predicted theoretically were close to those obtained experimentally.

The pressure values predicted theoretically versus drawing speed at the point where the distance from the entry point was 45 mm within the unit, and those obtained experimentally [7] at the same position, both for the maximum reduction in area of about 7 percent are shown in Fig. 2.9. In the experiment, the pressure of about 62.5 MN/m^2 occurred at a speed of 0.3 ms^{-1} for a reduction in area of about 7 percent, whilst the pressure of about 113 MN/m^2 was predicted to occur at a speed of 1.5 ms^{-1} for the same percentage reduction in area. Theoretical results tended to under-estimate the experimental results at slower drawing speeds and over-estimate the measured results at higher speeds.

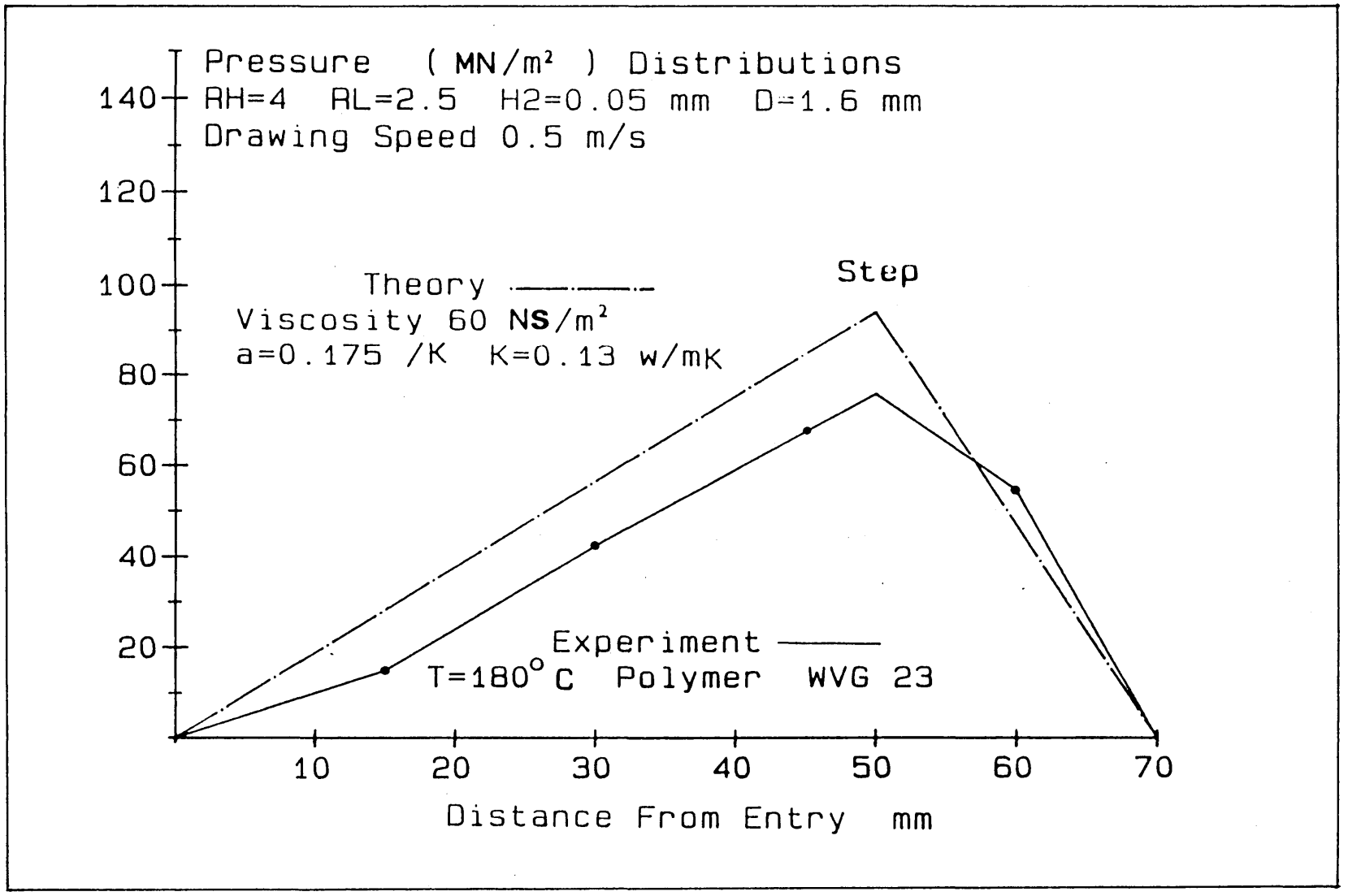


Fig. 2.7 Pressure Distributions Within the DRU

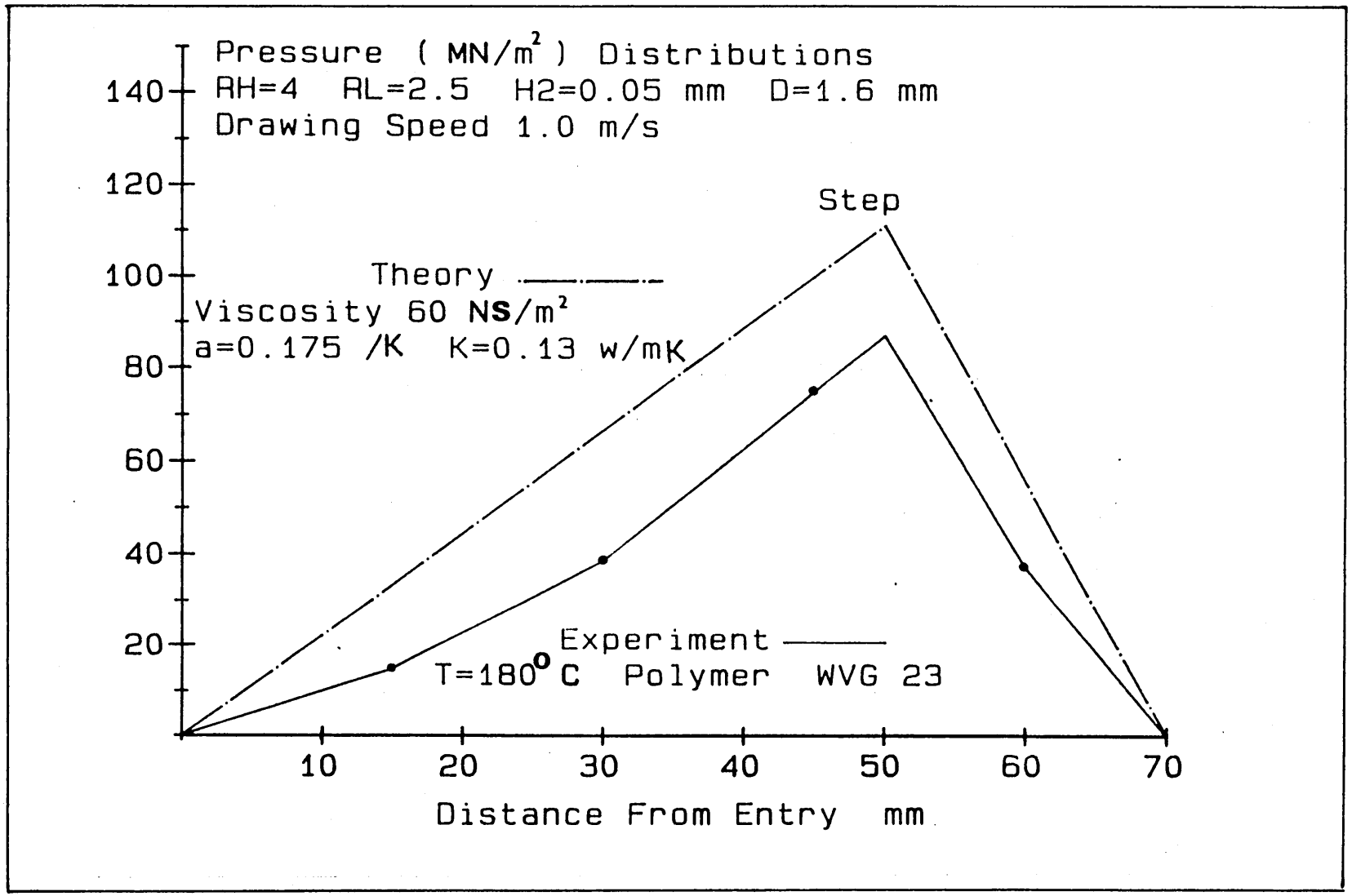


Fig. 2.8 Pressure Distributions Within the DRU

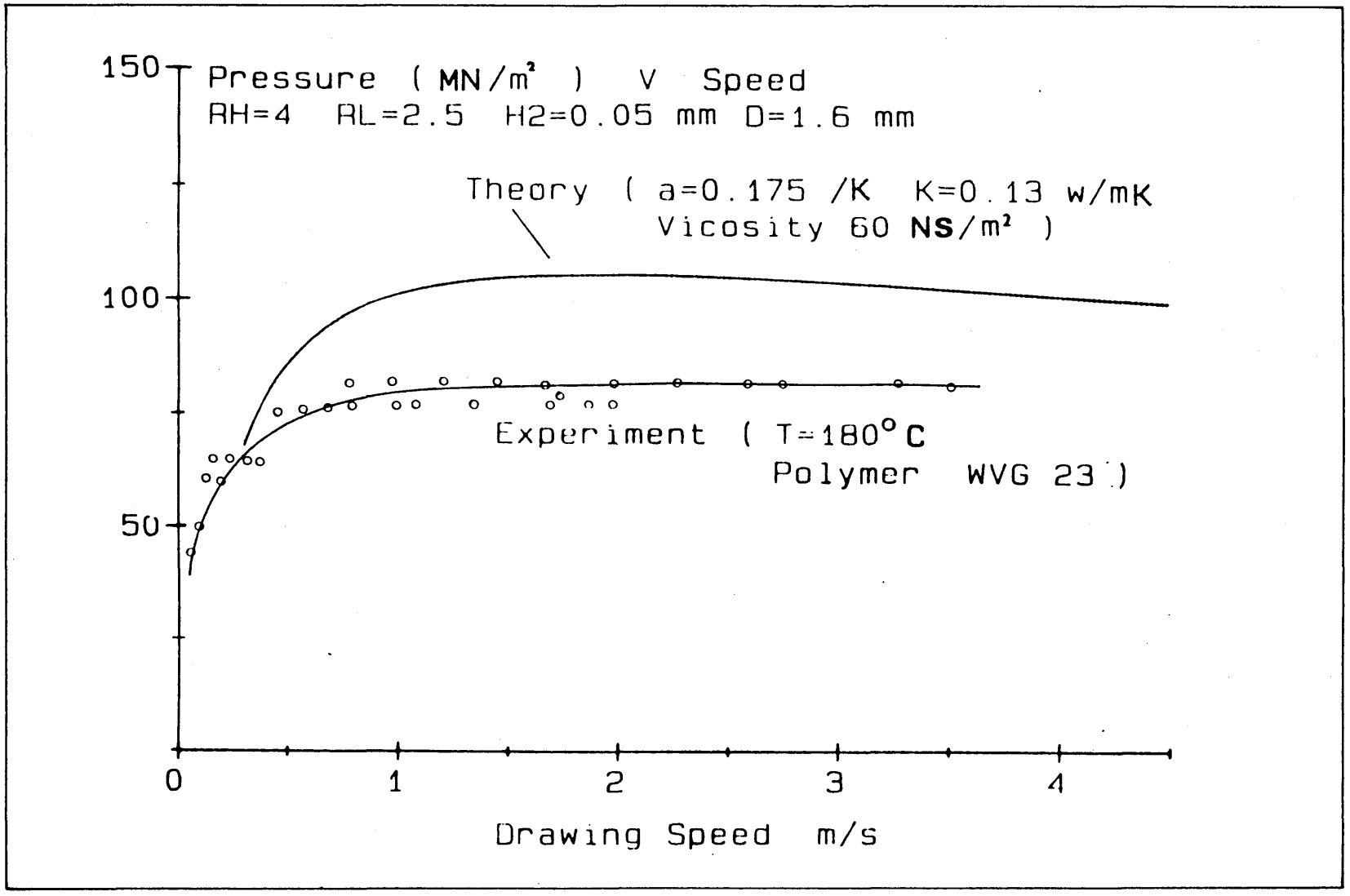


Fig. 2.9 Pressure Variations Versus Drawing Speed

Theoretical Drawing Loads

Drawing loads predicted theoretically and those obtained experimentally, both for the maximum reduction in area of about 7 percent, are shown in Fig. 2.10. In the experiment, the maximum drawing load of about 120 N occurred at the speed of 0.3 ms^{-1} for the reduction in area of about 7 percent, whilst the maximum drawing load of about 100 N was predicted to occur at the speed of 1.5 ms^{-1} for the same reduction in area. Theoretical results appeared to have lower values through all the range of the drawing speeds compared with the experimental results.

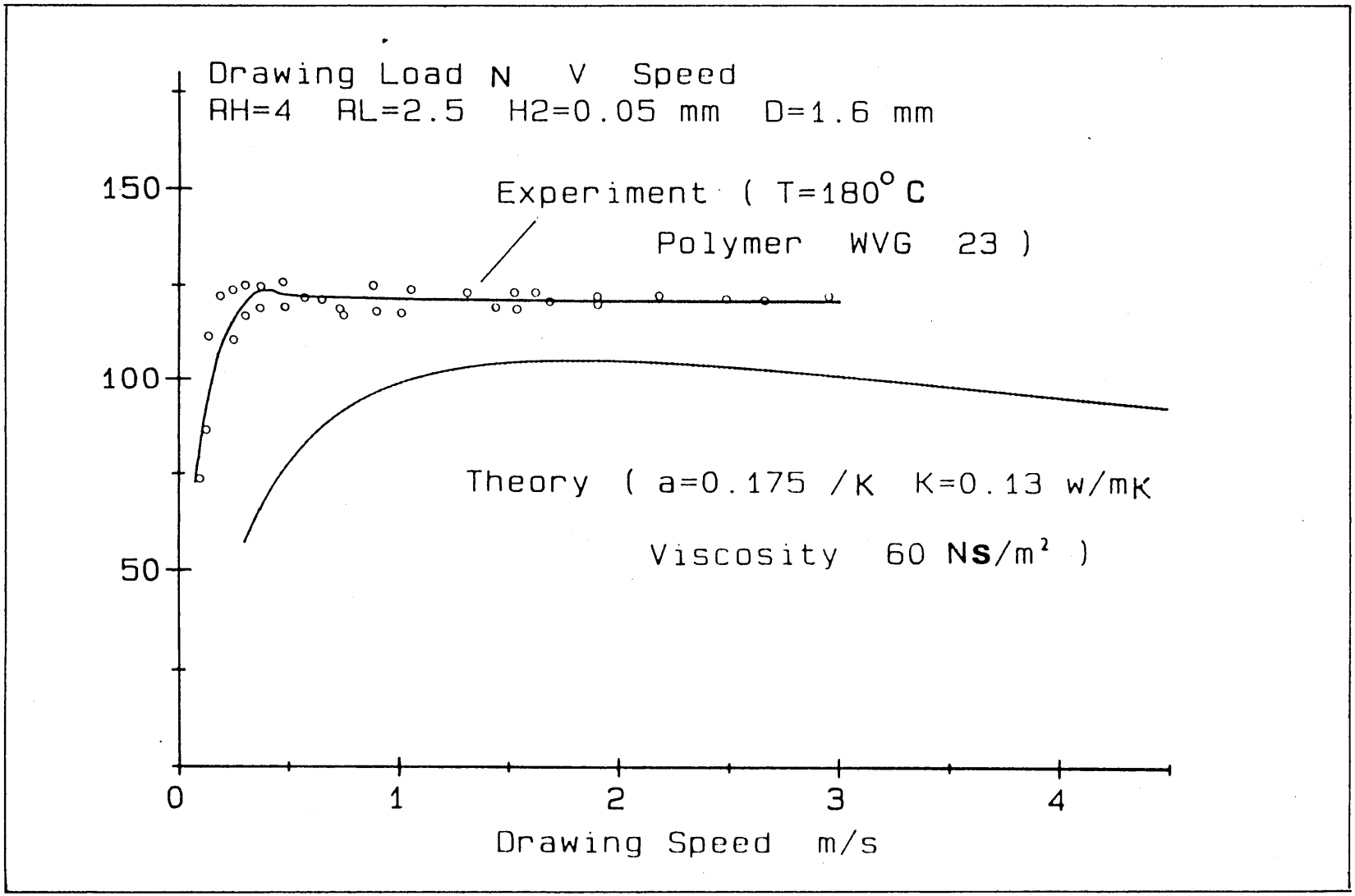


Fig. 2.10 Drawing Load Variations Versus Drawing Speed

The present analysis enabled the prediction of the percentage reduction in area, the pressure of the polymer melt within the unit and the axial stress in the wire. The theoretical investigation of the effect of several factors (for example, drawing speed and the temperature of the polymer melt) on the percentage reduction in area, enabled a basic control strategy to be established. This will be discussed in Chapter 3.

The Energy equation used in analysis takes the form:

$$\frac{\partial^2 T}{\partial Y^2} = -\frac{\mu}{K}\left(\frac{\partial u}{\partial Y}\right)^2$$

This simplified equation was used to predict the increase in the temperature of the polymer melt within the unit. During the wire drawing process energy is dissipated in two ways: through the action of viscous "friction" and through the plastic deformation of the wire. Both of these sources liberate heat which changes the melt temperature. However, the amount of heat generated from the deformation of the wire was not taken into consideration in the analysis. The viscosity of the polymer melt is known to be sensitive to temperature, therefore this may have introduced some discrepancy between the theoretical and practical results.

The apparent viscosity as a function of the melt temperature in the analysis is given by $\mu = \mu_0 e^{-aT}$. It is derived from Fig. 2.11. Note that the data indicating the viscosity dependance of temperature in Fig. 2.11 are at constant shear stress and pressure. It is known that usually for shear thinning polymer melt the increase in shear stress causes the decrease in apparent viscosity. Fig. 2.12 shows the behaviour of the shear thinning polymer. Fig. 2.13 presents the apparent viscosity of the polymer melt as a function of pressure. It must be remembered

that the shear stress and the pressure applied to the polymer melt within the unit during the wire drawing process were very high. However these effects were not incorporated in analysis and hence this might have contributed to the discrepancy in the results.

The value of the thermal conductivity K, used in the analysis, was 0.13 W/mK. This value was selected as being within the range normally encountered [13].

From Fig. 2.11 the magnitude of the viscosity temperature dependency constant "a" was estimated to be between 0.02 and 0.1 K^{-1} However, in the analysis "a" was taken to 0.175 K^{-1} in order to have reasonable results, hence the value of "a" chosen for analysis was only an approximation.

The predicted results of the percentage reduction in area showed two zones:

1) At slower speeds, the percentage reduction in area increased as the drawing speed was increased and reached its maximum value.
2) Then, the percentage reduction in area reduced slowly as the drawing speed was increased further.

The drawing speed was found to have the most dramatic effect on the percentage reduction in area. Fig. 2.14 shows the predicted values of the temperature change, percentage reduction in area, viscosity, maximum pressure and axial stress versus drawing speeds. As might have been expected, an increase in drawing speed increased the temperature of the polymer melt within the unit which caused a decrease in viscosity of the polymer melt, hence a percentage reduction in area, and decreases in the maximum pressure and axial stress.

The effects of other variables were also examined theoretically in order to investigate their influences on the predicted results. An increase in initial viscosity increased the percentage reduction in

area. Increasing the thermal conductivity also had the same effect. A reduction in viscosity temperature dependency constant "a" increased the percentage reduction in area. An increase in the initial temperature of the polymer melt was also expected to decrease the percentage reduction in area.

A very good agreement appeared to exist between the theoretical and experimental results as shown in Figs. 2.2 and 2.3. However differences were observed between the theoretical and experimental results in Figs. 2.4, 2.9 and 2.10, where the geometry of the dieless reduction unit and the initial diameter of the wire differed from those in Figs. 2.2 and 2.3. This suggested that the analysis did not fully represent the actual drawing process and certain additional factors, as mentioned before, should be incorporated in the analysis in order to improve the theoretical results.

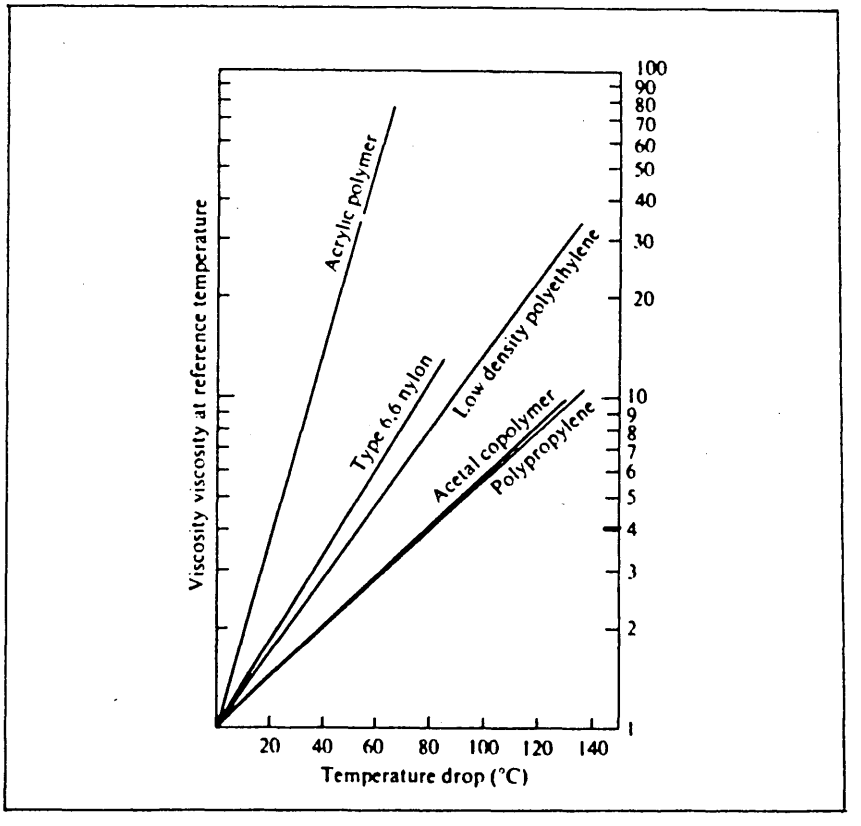


Fig. 2.11 Melt Viscosity at Constant Stress and Pressure as a Function of Temperature (Reference 11)

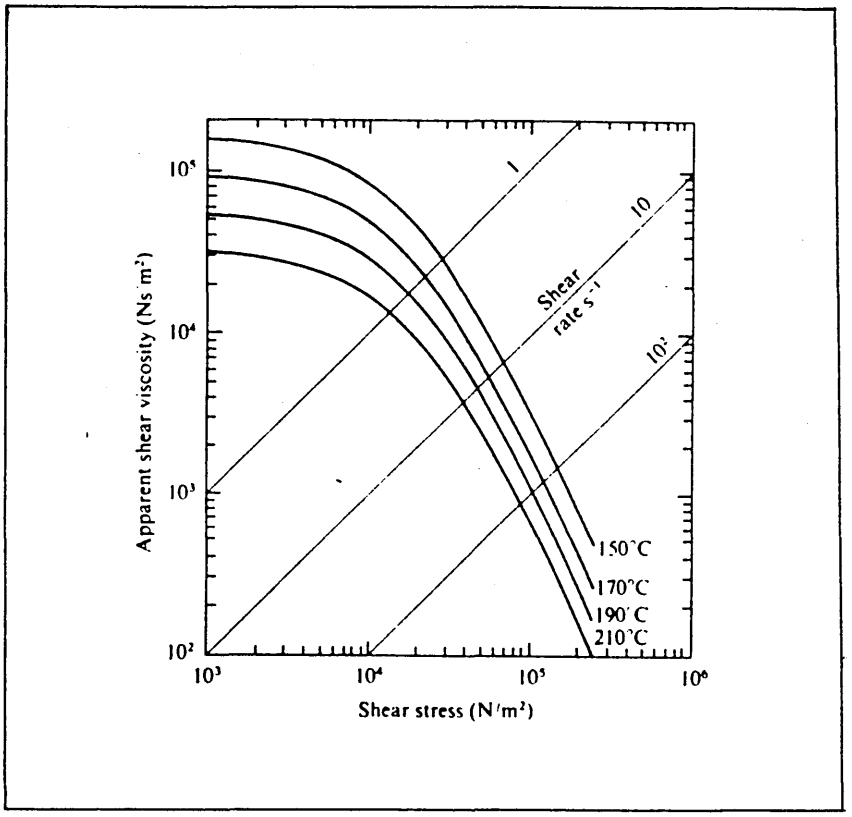


Fig. 2.12 Apparent Viscosity of a Low-density Polyethylene (Alkathene XDG 23) as a Function of Shear Stress (Reference 11)

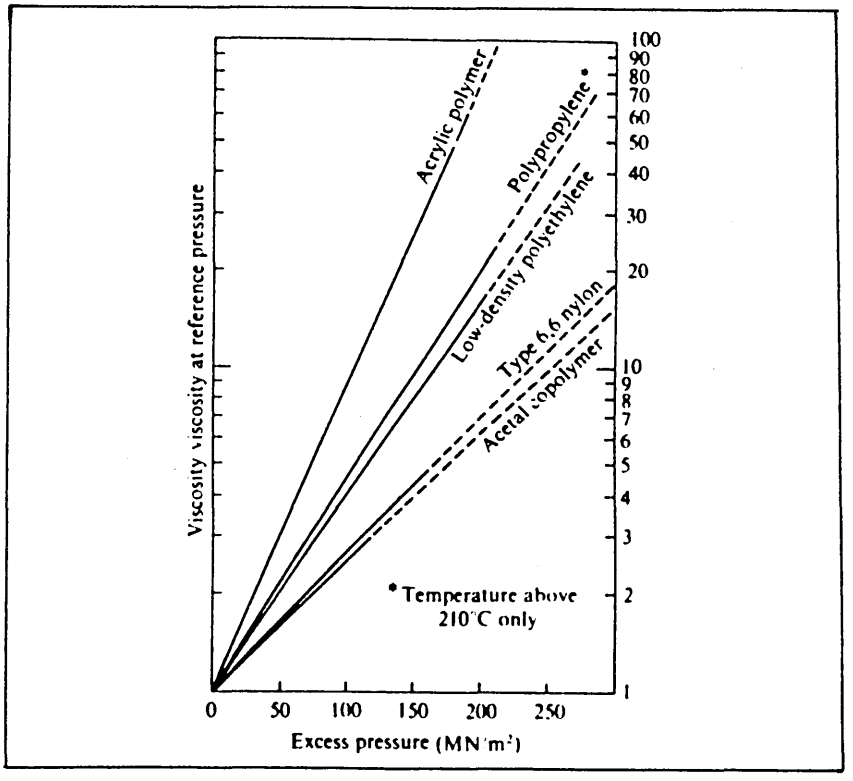


Fig. 2.13 Melt Viscosity at Constant Stress and Temperature as a Function of Pressure (Reference 11)

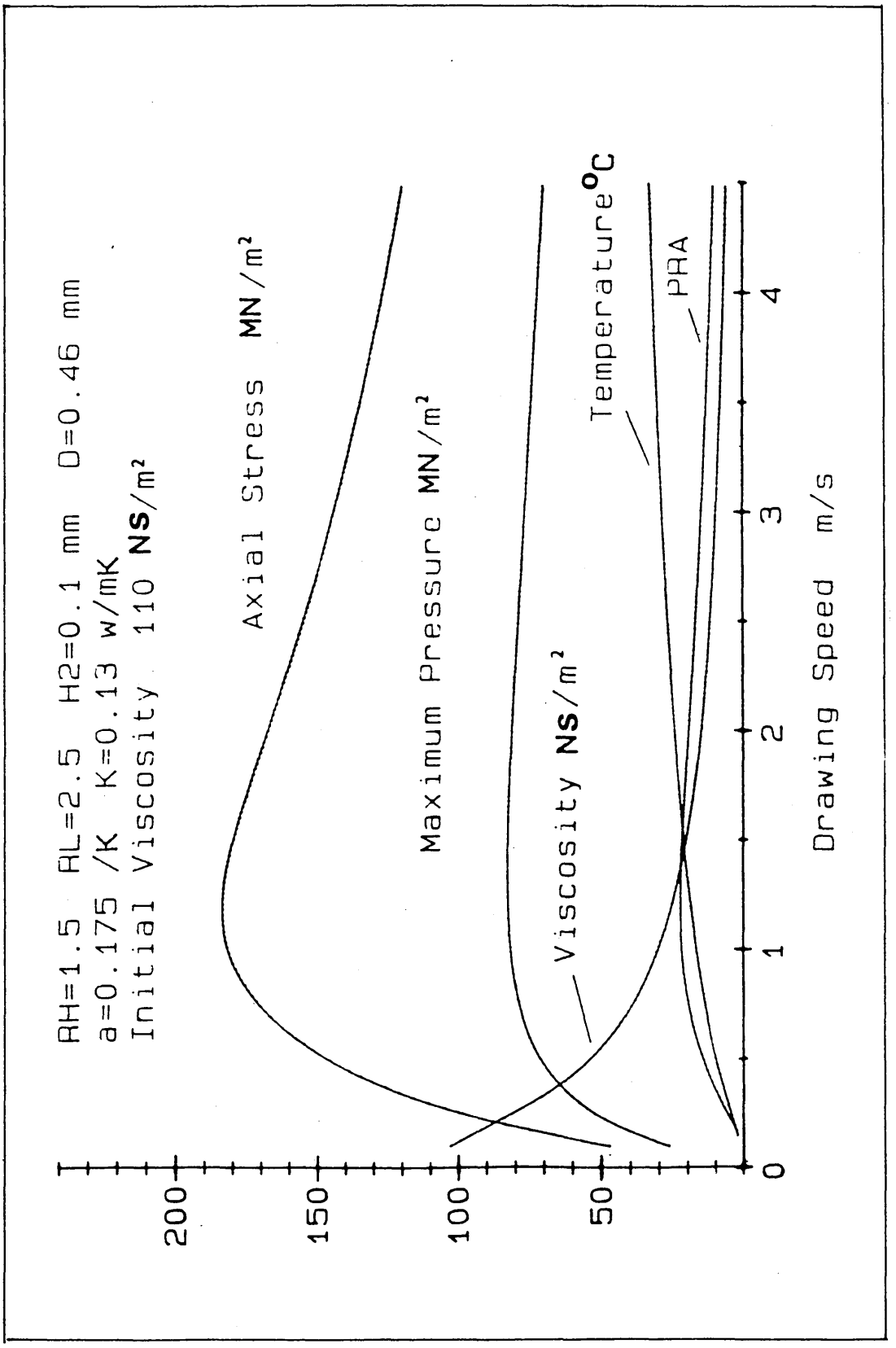


Fig. 2.14 Theoretical Temperature, Viscosity , PRA, Pressure and Axial Stress Variations versus Drawing Speeds

Chapter 3 : Microcomputer-Based Dieless Wire Drawing Control System

3.1 - Introduction

From the previous theoretical analysis, it can be seen that during the wire drawing process, the reduction in area of the wire is affected by several factors; the significant ones are the drawing speed, the performance of the polymer used as pressure medium, the temperature of the polymer melt and the configuration of the DRU. The control of the process is primarily required for product quality, so methods of controlling the wire drawing process should be looked at in order to produce wires of desired qualities in terms of consistency in the diameter over a long length, or uniform change in area with a desired rate for a given length.

To further investigate the drawing process, an extensive experimental program was conducted, in which parameters such as drawing speed, the temperature of the polymer melt and polymers were varied, and a considerable amount of data was obtained. Some of the results are shown in Fig. 3.1 to Fig. 3.7.

From both theoretical and experimental results, the drawing speed is found to have the most dramatic effect on the reduction in area of the wire. The percentage reduction in area (PRA) versus drawing speed shows two zones: at slower drawing speed, the PRA increases as the drawing speed is increased, and then it reaches its maximum value. After that, the PRA reduces as the drawing speed is increased further. Compared with the second zone, the PRA curve in the first zone has better linearity and the full range of the PRA is spanned within a limited speed extent. So the design of the control system was based on the first zone.

The effect of the temperature on the vicosity of the polymer melt was included in the analysis by varying the initial viscosity. It was assumed that varying the initial viscosity had the same effect as the

change in temperature (see Fig. 2.2 to Fig. 2.3). Both theoretical and experimental results showed that the overall percentage reduction in area was affected greatly by the temperature of the polymer melt. Experiments also showed that it took about 70-80 minutes for the temperature to reach a uniform distribution within the melt chamber after a change in set temperature. Uniform temperature distribution was of great importance for a steady wire drawing process. The temperature could be a manipulated variable to control the PRA, but in consideration of the time needed for temperature to reach a uniform distribution, the control strategy was designed to keep the temperature constant during the whole drawing period. Instead, the drawing speed was regulated to produce the desired products. For each polymer, optimum drawing speed and temperature could be found at which the deformation of the wire reached its maximum value. It is often the case that the maximum PRA is desired for one pass during the drawing process.

Three polymers having different densities and viscosities were selected as pressure media. They were polymer WVG 23, Nylon 6 and ELVAX 650. With ELVAX 650 at 150°C the PRA curve was almost linear from a reduction in area of 4 percent at speed of 0.03 ms^{-1} to a reduction in area of 21 percent at speed of 1.04 ms^{-1}, as shown in Fig. 3.6. Polymer WVG 23 at 110°C and polymer Nylon 6 at 260°C also showed good performance (see Fig. 3.1 and Fig. 3.3) in terms of wide range of PRA in the first zone. These results were employed in producing samples.

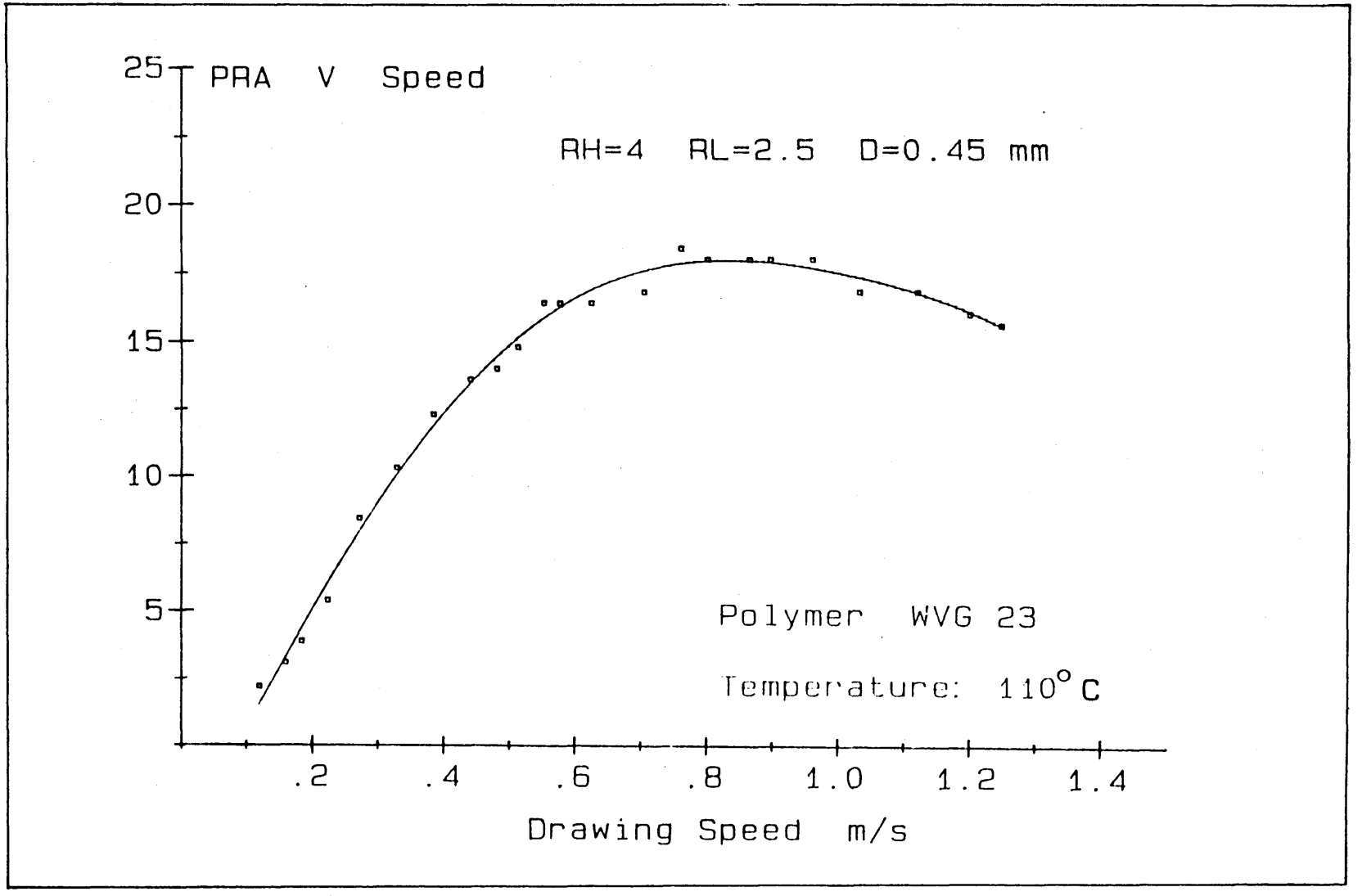


Fig. 3.1 Percentage Reduction in Area Versus Drawing Speed

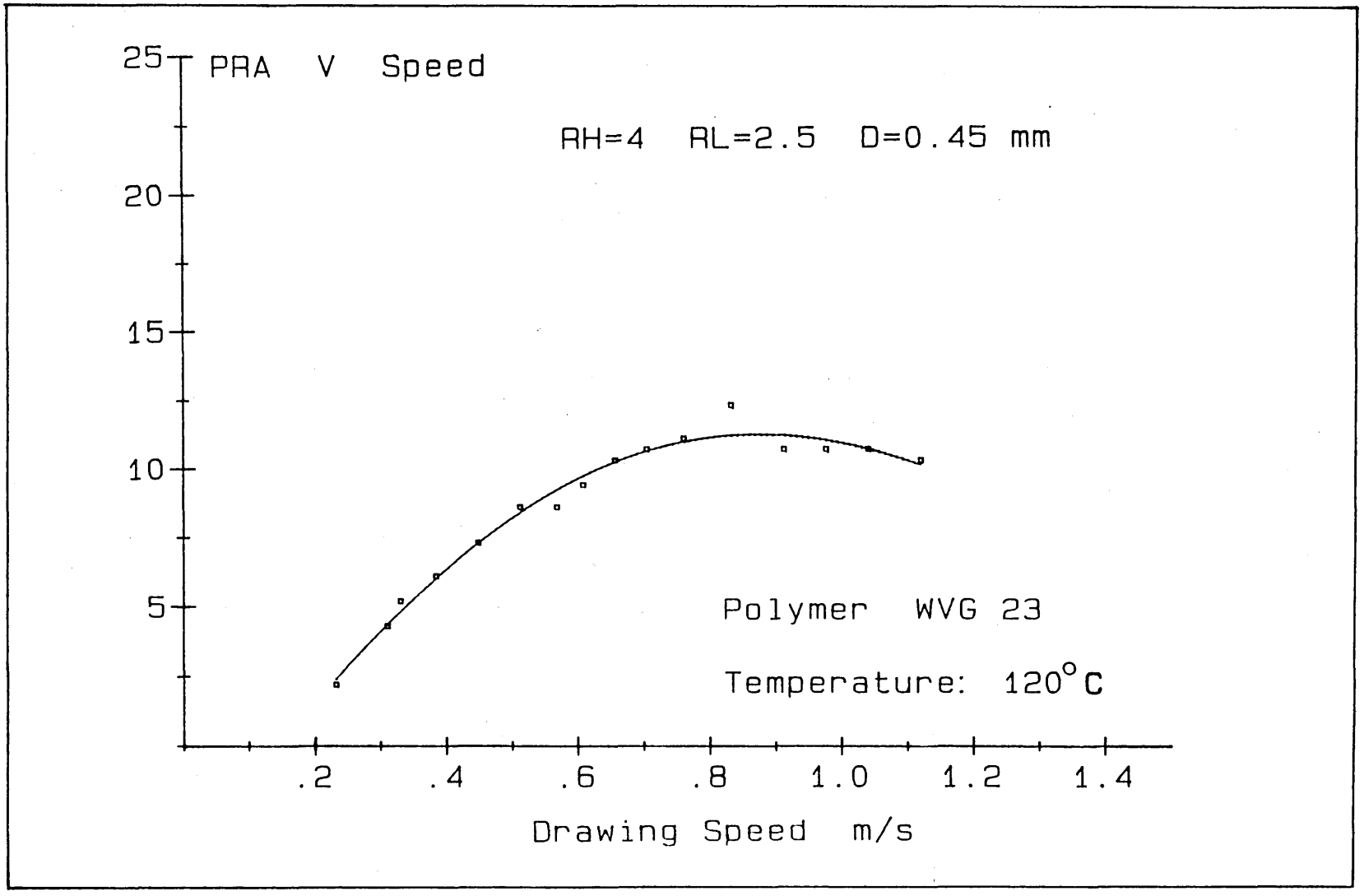


Fig. 3.2 Percentage Reduction in Area Versus Drawing Speed

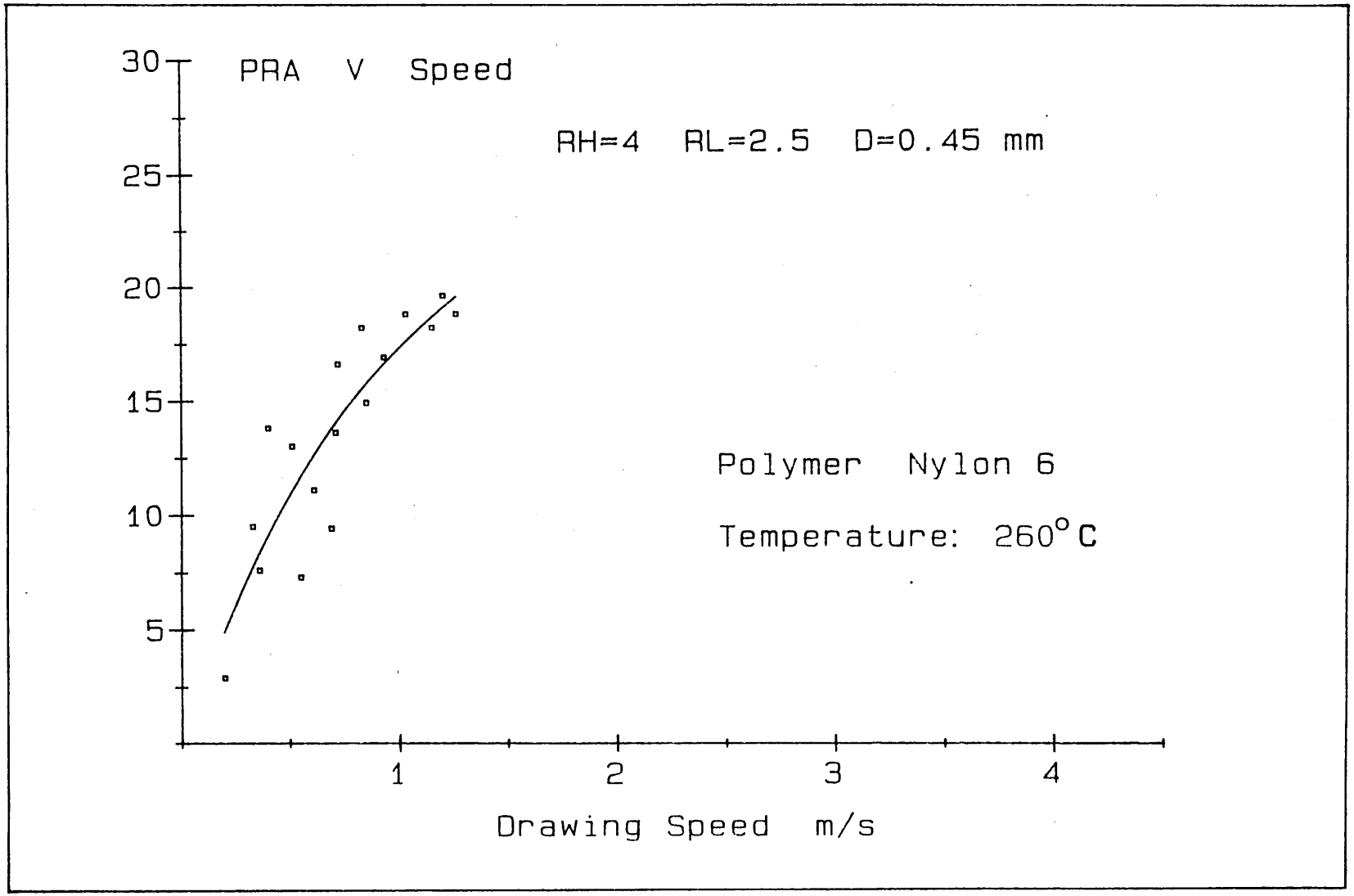


Fig. 3.3 Percentage Reduction in Area Versus Drawing Speed

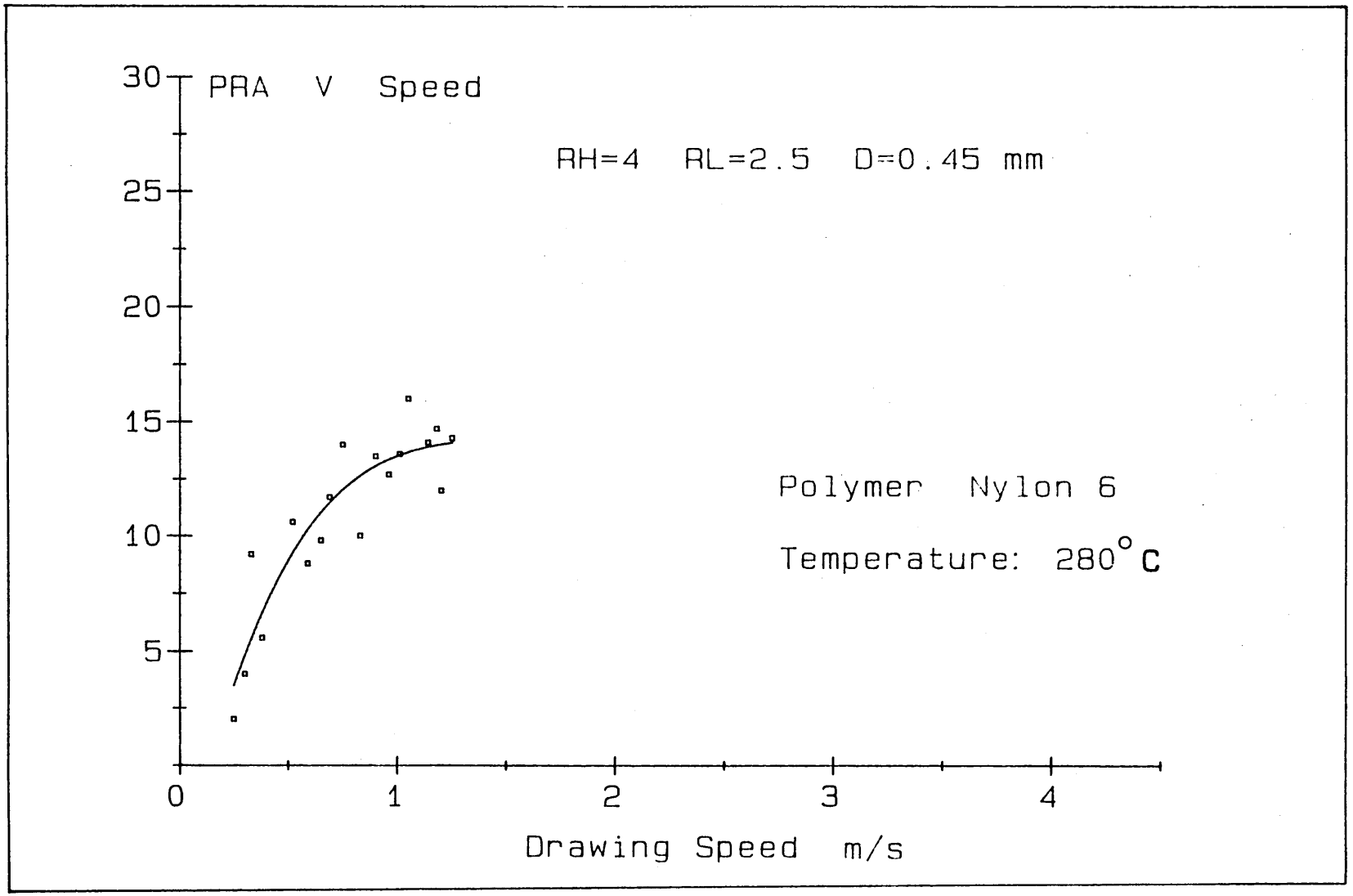


Fig. 3.4 Percentage Reduction in Area Versus Drawing Speed

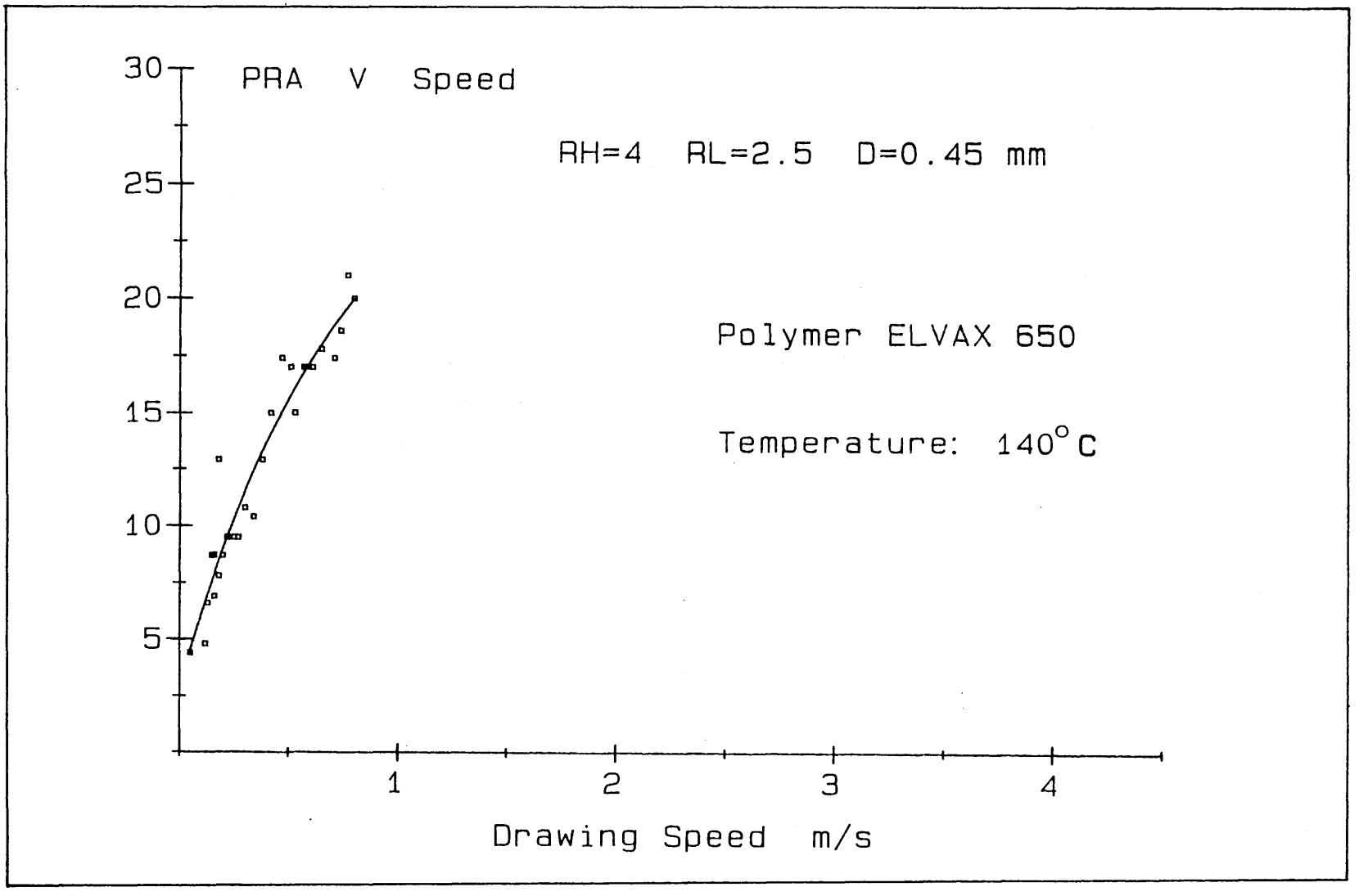


Fig. 3.5 Percentage Reduction in Area Versus Drawing Speed

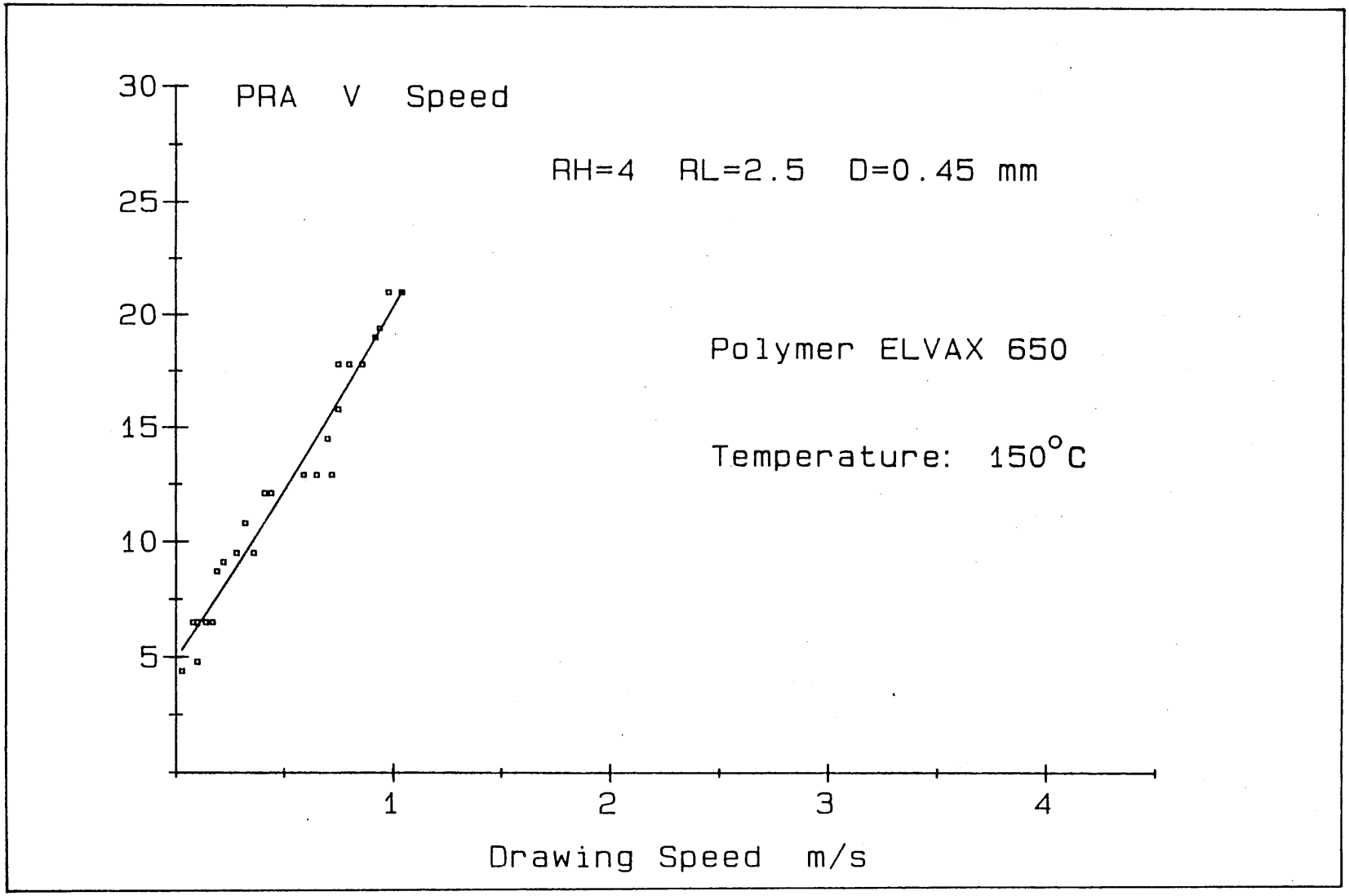


Fig. 3.6 Percentage Reduction in Area Versus Drawing Speed

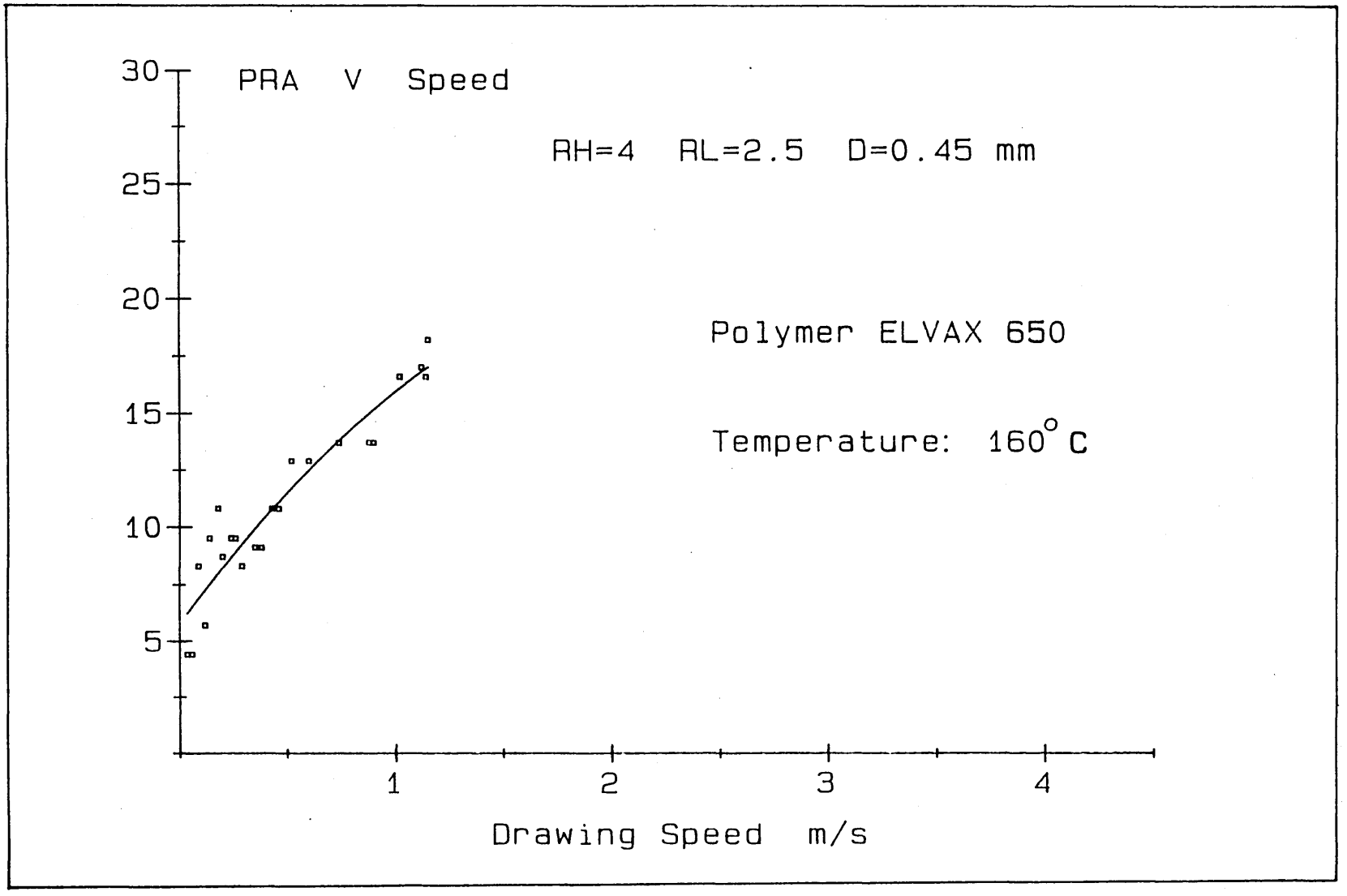


Fig. 3.7 Percentage Reduction in Area Versus Drawing Speed

In practice, a single loop control system, as shown in Fig. 3.8 can be used to control the PRA of the wire, i.e., the diameter of the wire.

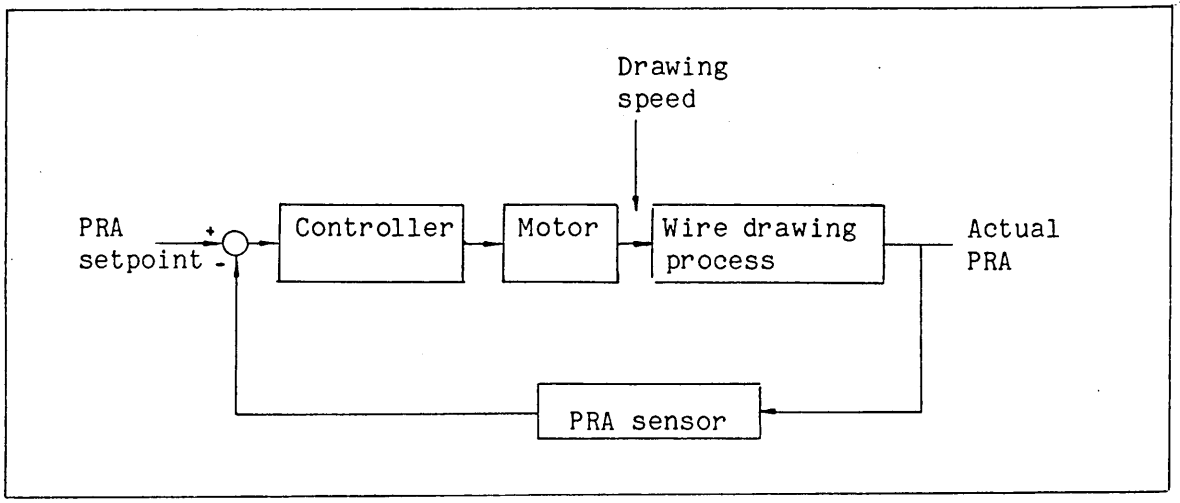


Fig. 3.8 Closed-loop Control system

In Fig. 3.8, the drawing speed is a manipulated variable, the PRA is a controlled variable, and the temperature is kept at its preselected value during the whole drawing process. The heart of the control system is the sensor. After the wire leaves the DRU, there is a polymer coating on the surface of the wire which remains as a protective cover of the product. Thus, there is not a simple method to directly measure the PRA of the wire due to this coating on its surface. A method which can measure the PRA of the wire during the drawing process has been designed and constructed, and this is described in section 3.7.

Further, a simpler configuration was taken into consideration. The theoretical and experimental results showed that the reduction in area of the wire remained constant provided that both the temperature of the polymer melt and the drawing speed were kept constant throughout the test. A unique value of the PRA thus existed for a particular

temperature and drawing speed. This suggested that an open-loop system for the reduction control in combination with two closed-loop systems for drawing speed and temperature control might be used to maintain the quality of the products. A system was therefore designed in which open-loop control was implemented for the PRA using experimental results in the form of look-up tables in the microcomputer memory locations. From these, both temperature and drawing speed for a given polymer corresponding to a required PRA could be found. After the required specification was input into the microcomputer via a keyboard, the microcomputer would automatically set the temperature, and then start the drawing process with a selected speed from the look-up tables to obtain the required product. The temperature and drawing speed were each controlled by a closed-loop control system, but the setpoints of the two systems were given by the microcomputer, so that each individual loop was operated in such a way as to produce the required wire. The sensor to detect the PRA was not included in the whole system at this stage. Experimental results showed that the quality of the product achieved the desired level with this control configuration. Fig. 3.9 shows the schematic diagram of the control system.

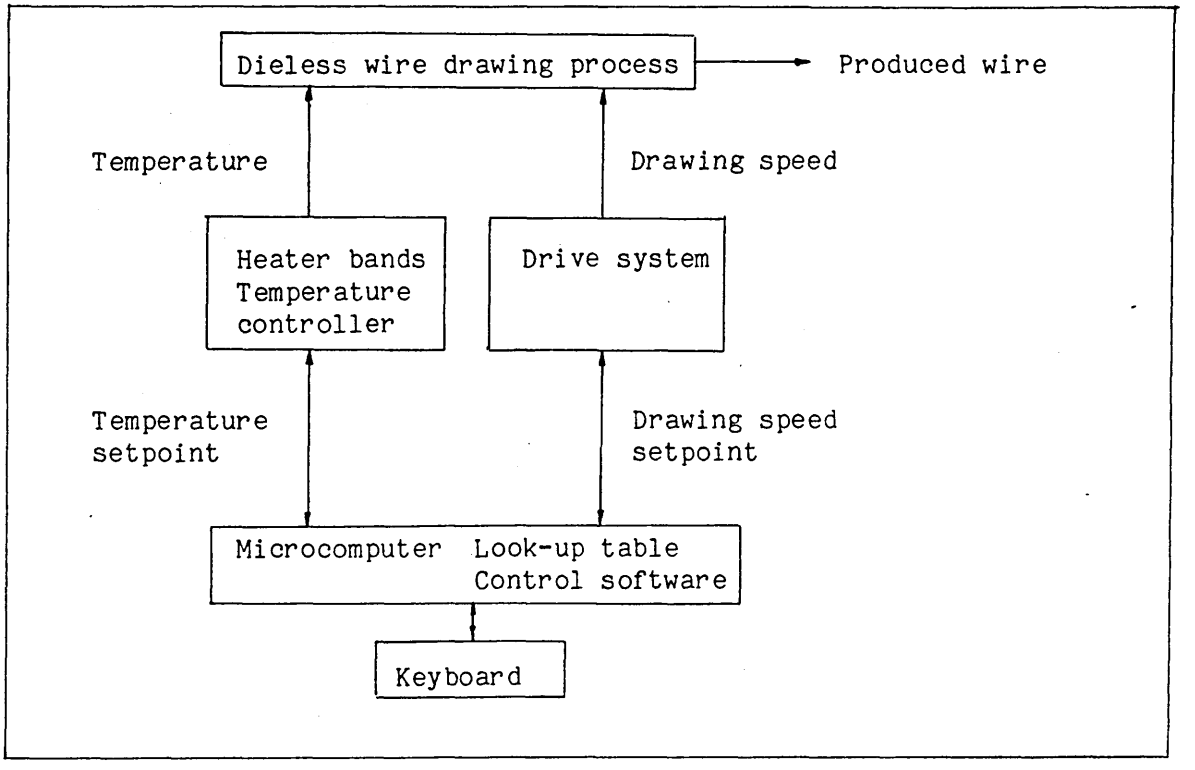


Fig. 3.9 Schematic Diagram of the Control System

3.2 - Description of the Experimental Equipment

The existing dieless wire drawing system as shown in Fig. 3.10 was designed by Dr. H. Parvinmehr. Wire from the coil placed on the stand was passed through the guides and over the pulley, and then pulled through the polymer feeder-DRU assembly and then wound onto the bull block. The detail of the polymer feeder-DRU assembly is shown in the schematic diagram of Fig. 3.11.

The undeformed wire passed through the polymer melt chamber before undergoing deformation within the DRU. The melt chamber was kept filled with polymer which was melted by means of an electric heater band. Because the polymer melt chamber could only store a limted amount of polymer, a hopper was mounted on the melt chamber having sufficient capacity to supply the chamber for a complete series of tests. The

hopper was also heated by an electric heater band and the temperature of the polymer melt within the chamber and the hopper was controlled thermostatically within ±3°C of the set temperature. This was achieved by use of thermocouples connected to controlled power supplies which fed electrical power to the heater bands. A digital thermometer was connected to the thermocouples to give a temperature display.

The hopper was also connected to a pressurized air supply of 100 psi via a 2-way servo valve. Hence the air pressure was activated onto the polymer surface, helping the polymer to flow with the wire into the DRU.

The original drawing bench consisted of interchangeable bull blocks of 50, 100 and 300 mm diameter, driven by a Schrage 3-phase electric motor capable of running at speeds between 550 and 2200 revolutions per minute. Since this was considered virtually impossible to control automatically with a microcomputer, the drive system was completely redesigned to incorporate a bull block of 300 mm diameter driven by a hydraulic motor (type OMP 50) capable of running at speeds infinitely variable between 0 and 800 revolutions per minute. The power was transmitted from the motor to the bull block via a 29:3 reduction worm gear box (CROFT type 41/551/05) and a coupling clutch (BROADBENT type DP25) which enabled the bull block to be engaged when the hydraulic motor was running at the required speed, thus providing some flexibility during the tests. The above arrangements facilitated drawing speeds infinitely variable between 0 and 1.3 ms^{-1}.

The above equipment was mounted on a suitable bench of welded steel construction and guarding was placed around all moving parts to provide adequate safety.

The rotational speed of the hydraulic motor was measured by coupling a tachogenerator (type AM 363) to the gear box rotating shaft. The feedback signal from the tachogenerator was transmitted into the

microcomputer through an electronic ampli-filter circuit and I/O interface. The microcomputer calculated the value of the manipulated variable (valve position) directly from the values of the speed setpoint and the feedback signal according to a predefined algorithm. The result of the calculation was applied to the electro-hydraulic servo valve through an output interface and an electronic amplifier. This arrangement produced a hydraulic flow from the pump to the hydraulic motor which was proportional to the microcomputer output. The details concerning the control system will be discussed in the following sections. Fig. 3.12 shows the schematic diagram of the dieless wire drawing system.

Plates 1 and 2 show the final arrangements of the dieless wire drawing equipment.

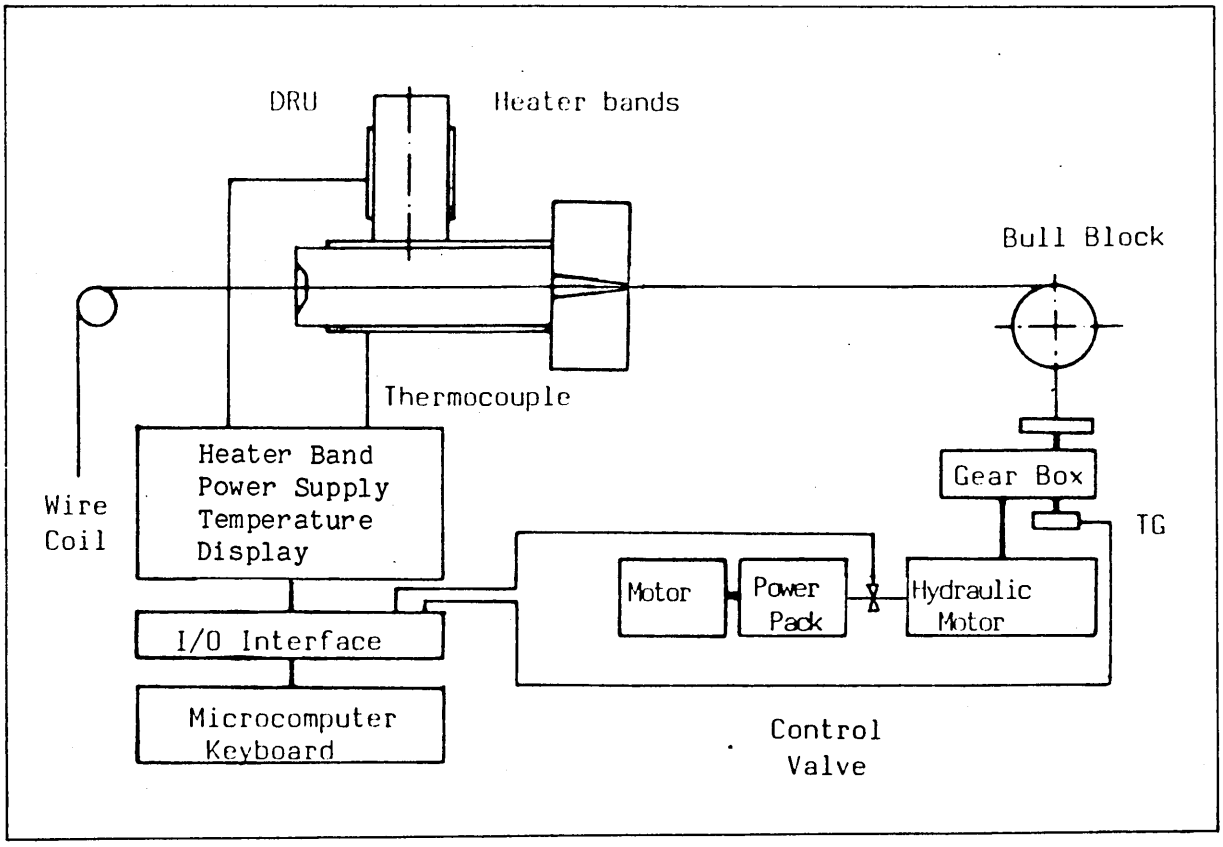


Fig. 3.12 Schematic Diagram of the Dieless Wire Drawing System

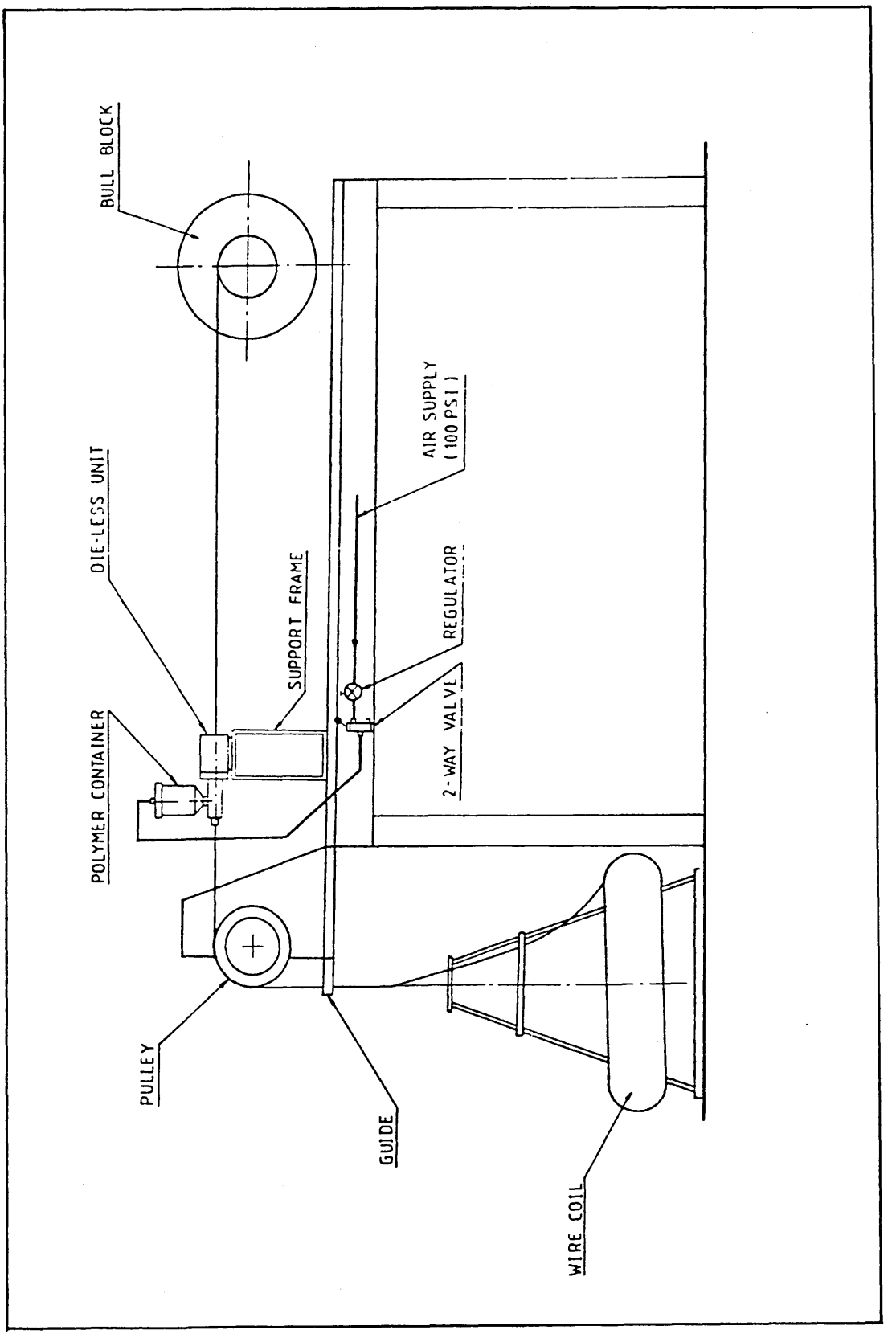


Fig. 3.10 Wire Feed Assembly

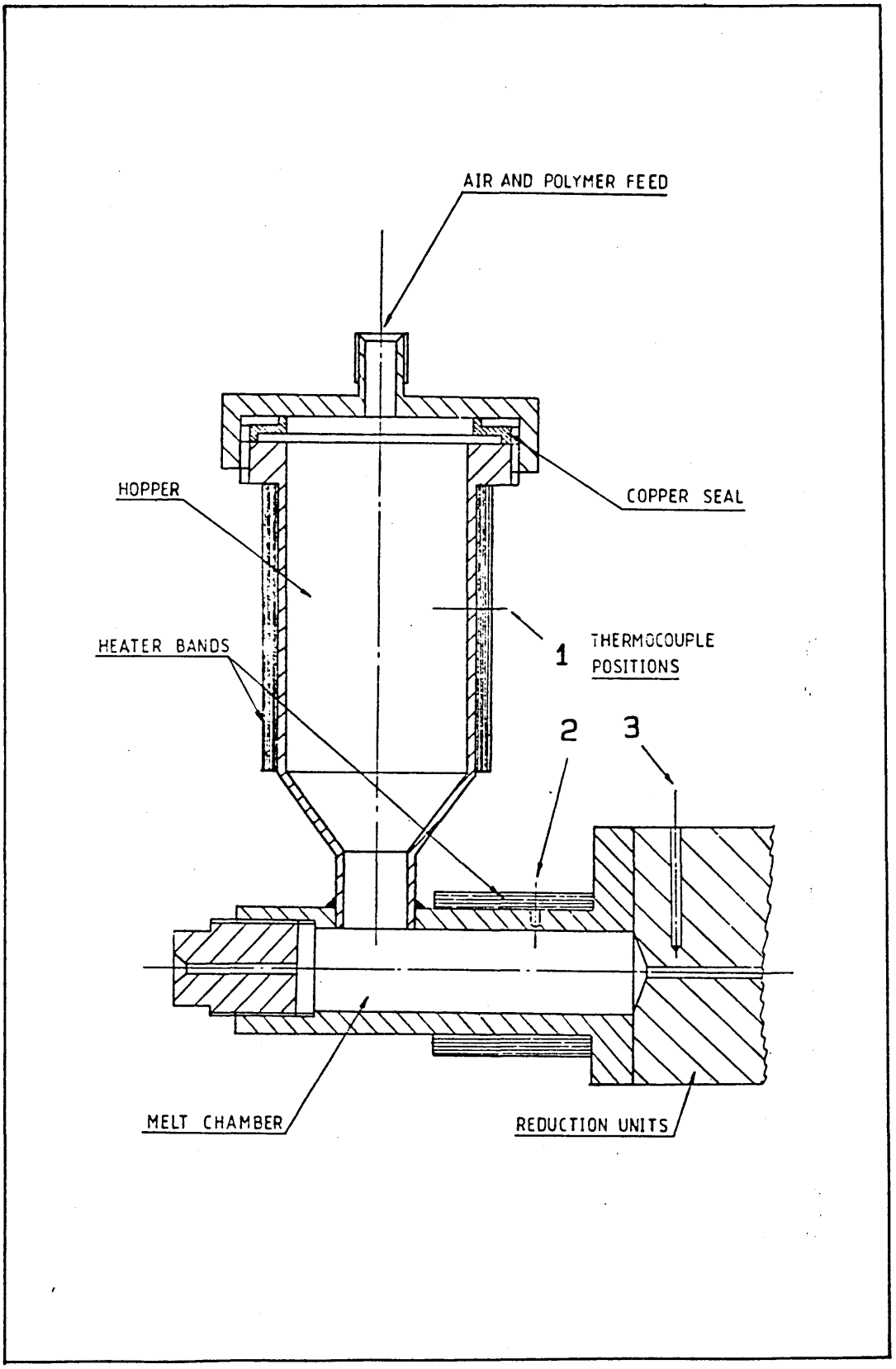


Fig. 3.11 Polymer Feeder-DRU Assembly

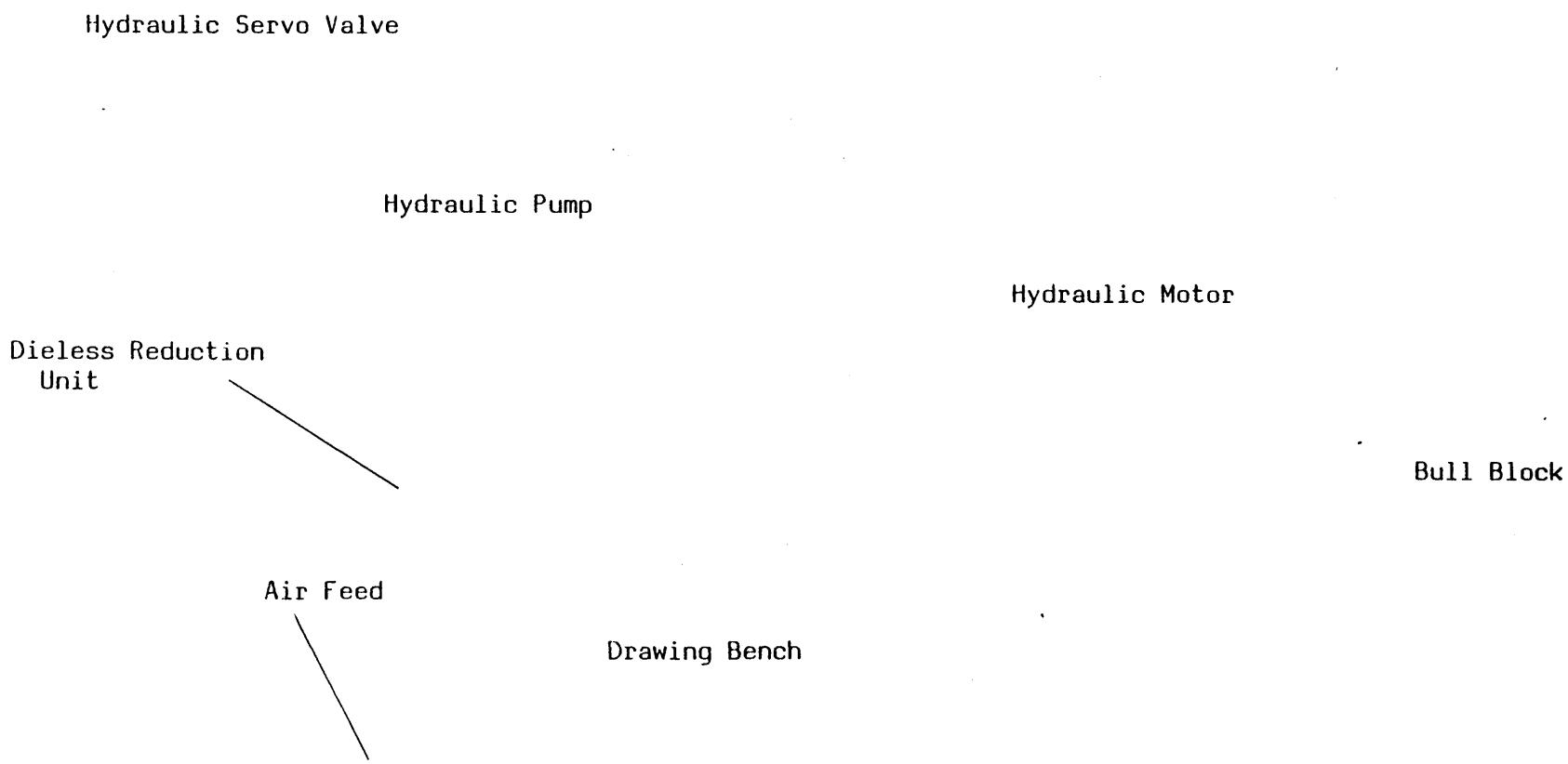

Hydraulic Servo Valve
Hydraulic Pump
Hydraulic Motor
Dieless Reduction Unit
Bull Block
Air Feed
Drawing Bench

Plate 1

Hydraulic Servo Valve

Hydraulic Pump

Hydraulic Motor

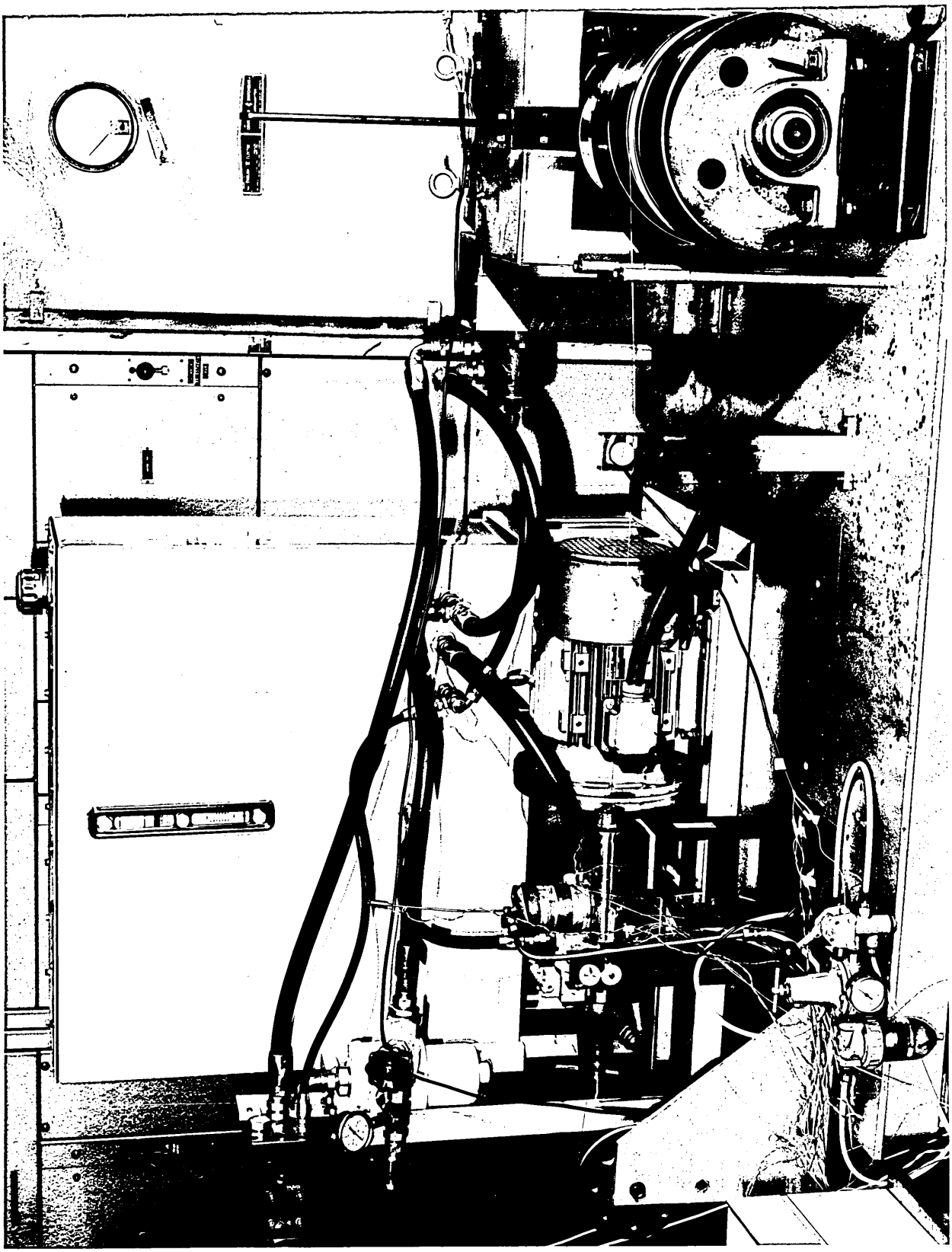
Dieless Reduction
Unit

Bull Block

Air Feed

Drawing Bench

Plate I



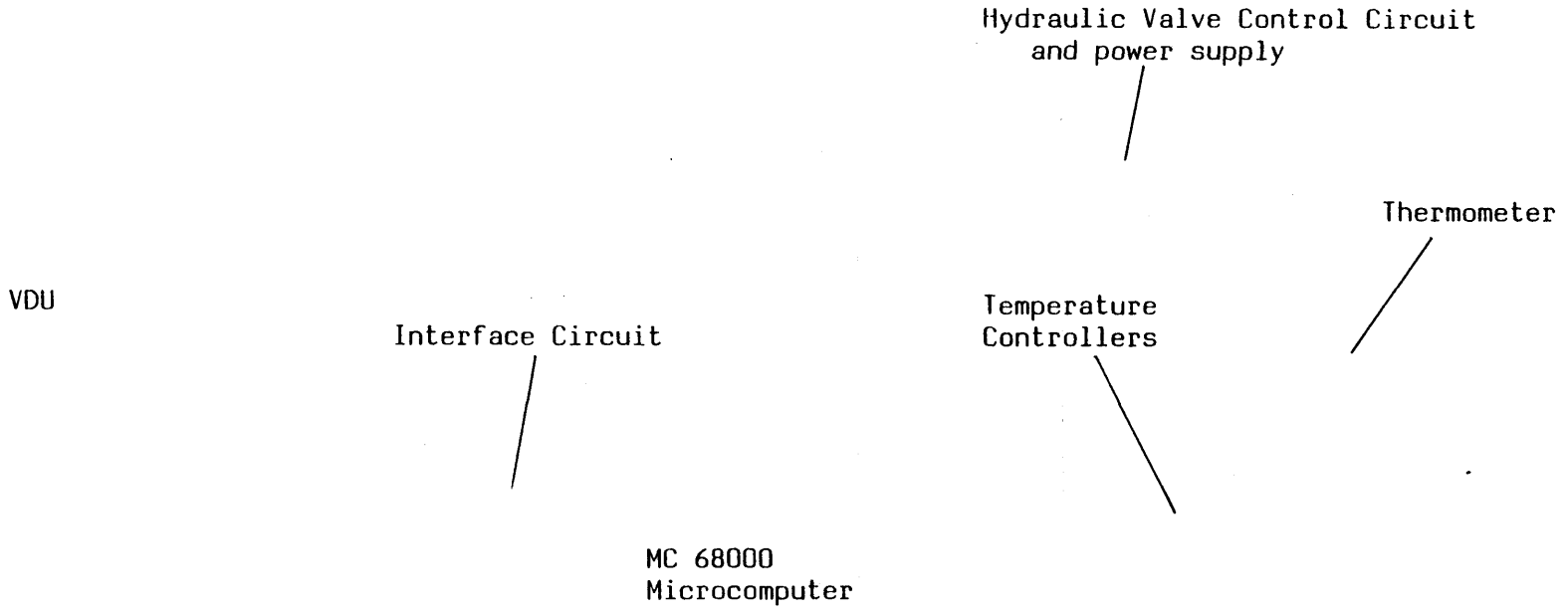

Hydraulic Valve Control Circuit
and power supply
Thermometer
VDU
Temperature
Controllers
Interface Circuit
MC 68000
Microcomputer

Plate 2

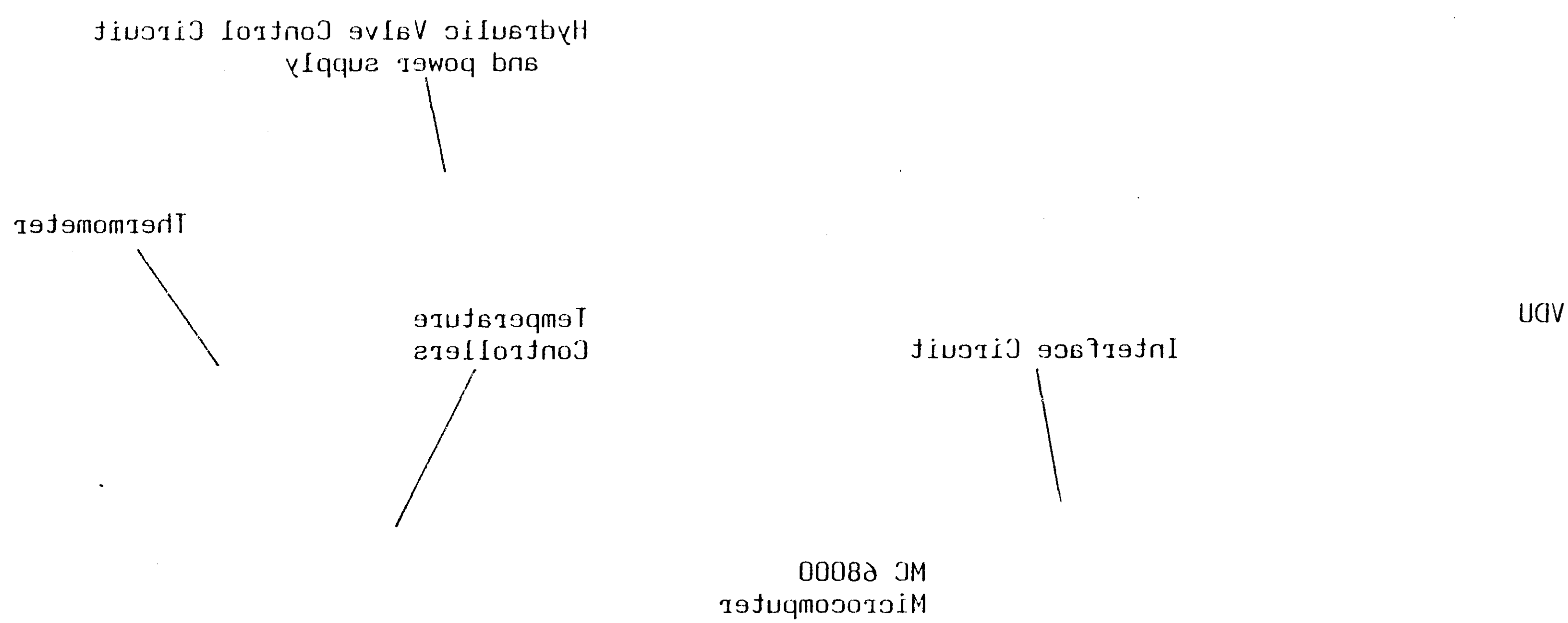
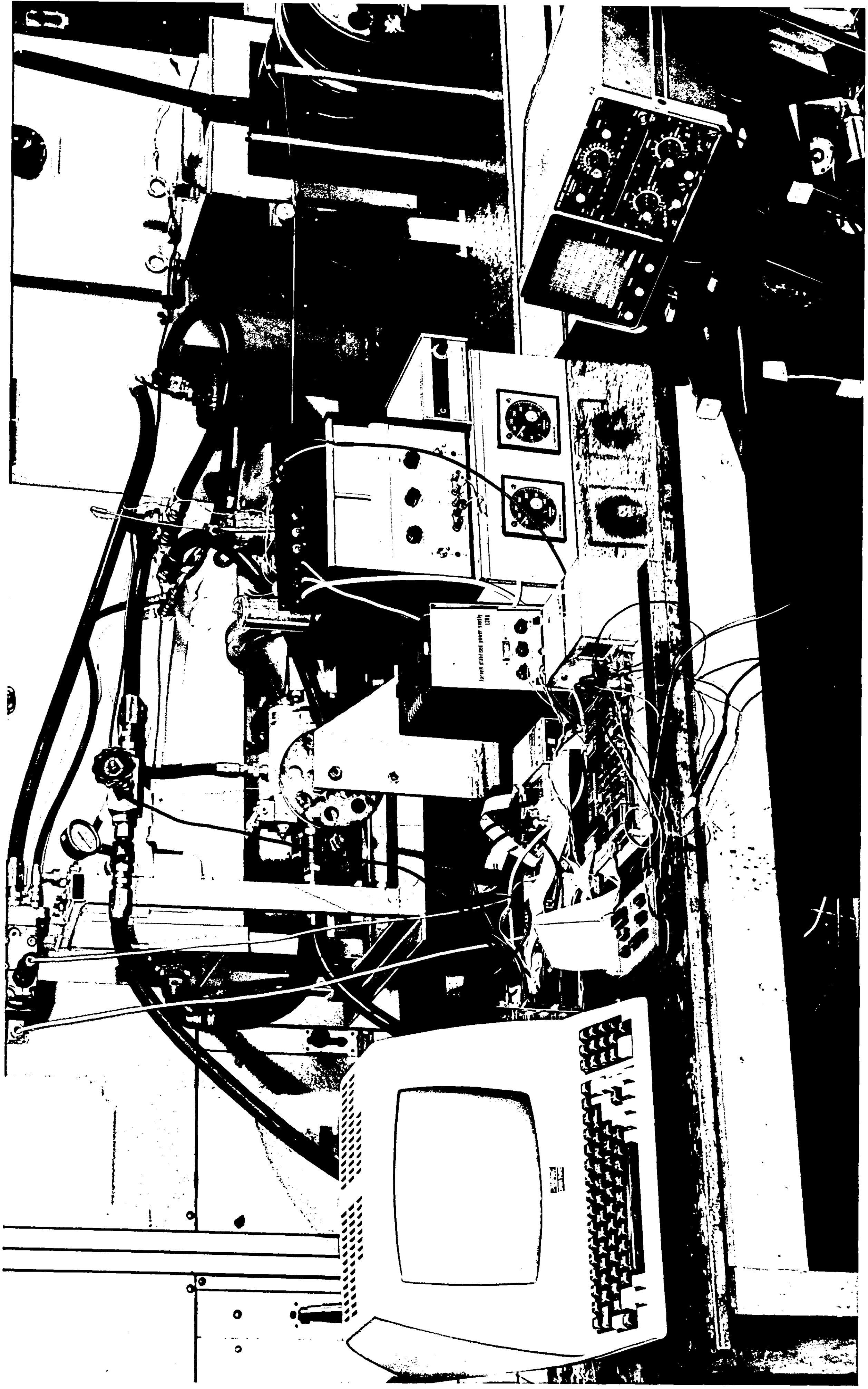

Hydraulic Valve Control Circuit
and power supply
Thermometer
Temperature
Controllers
Interface Circuit
VDU
MC 68000
Microcomputer

Plate 2



3.3 - Microcomputer and I/O Interface Circuit

The microcomputer used to implement the automatic control system was based on an MC68000 microprocessor. Fig. 3.13 shows the functional block diagram of the microcomputer.

The microcomputer includes:

a) 4-megahertz MC68000 16-bit MPU

b) 32k bytes of dynamic RAM and 16k byte firmware ROM / EPROM monitor.

c) Two serial communication ports for a terminal and a host. Both are RS-232 compatible and have selectable baud rates.

d) A parallel port which can be used for I/O or for a printer interface.

e) Audio tape serial I/O port.

f) Self-contained operating firmware that provides monitor, debug and disassembly / assembly functions.

g) 24-bit programmable timer.

h) Wire-wrap area provided for custom circuitry.

i) Reset and Abort function switches.

The MC68000 based microcomputer was chosen for a number of reasons. The 24-bit timer can be programmed to generate interrupts to drive the execution of user's program. Multiply and divide operations are available for signed and unsigned operands using byte or word multiply to produce a word or a long product and a long word dividend with word divisor to produce a word quotient with a word remainder. This makes it possible to optimise 8-bit fixed-point or 24-bit floating-point arithmetic, resulting in a short execution time of the user's program, which can often meet the requirement of the sampling period for a real-time process control. The wire-wrap area on the microcomputer board facilitates convenient interface between the microcomputer and the process. The self-contained operating firmware of monitor, debug and disassembly / assembly functions provides the fundamentals of

man-machine conversation. Also there is a substantial support available both in hardware and software. All of these indicate that the MC68000 microcomputer is appropriate to be used to implement the control system of the dieless wire drawing process.

The main parts of the interface circuit will be described in the following paragraphs.

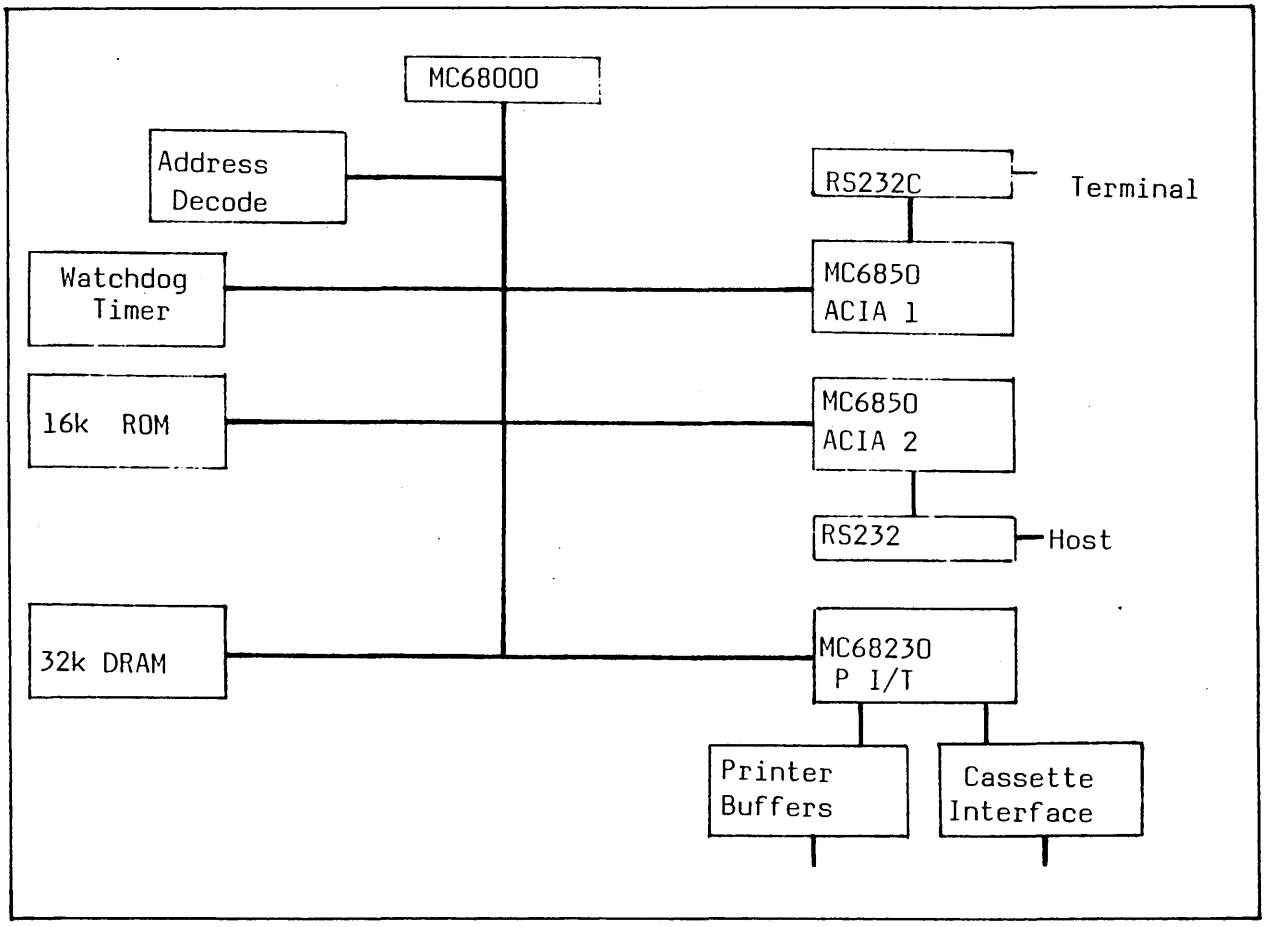


Fig. 3.13 Functional Block Diagram of MC68000 Microcomputer Board

3.3.1 - Interface to Speed and Temperature Control Systems

MC6821 Peripheral Interface Adapter (PIA) [15]

The block diagram of the MC6821 peripheral interface adaptor is shown in Fig. 3.14. It provides universal means of interfacing

peripheral equipment to the MC6800 or MC68000 microprocessor unit with 8-bit bidirectional data bus (D0-D7) for communication with the MPU, two bidirectional 8-bit buses (PA0-PA7, PB0-PB7) for interface to peripherals, three chip select lines (CS0, CS1 and $\overline{CS2}$), two register select lines (RS0 and RS1), two interrupt request lines ($\overline{IRQA}$ and $\overline{IRQB}$), Read / Write line (R/$\overline{W}$), enable line, reset line, interrupt input lines (CA1 and CB1) and peripheral control lines (CA2 and CB2). These signals, in conjuction with the MC68000 $\overline{VMA}$, $\overline{VPA}$ and E three signals, permit the MPU to have complete control over the PIA and interface with peripheral equipment.

PIA Address Decode

A 64k-byte segment of the system map is reserved for an MC6800 type interface in the MC68000 microcomputer board. The MC6800 page enable E6 is activated when memory page $030000-$03FFFF is selected and both $\overline{VMA}$ and $\overline{LDS}$ are asserted. Fig. 3.15 shows the logic generating signal E6. The memory page enable is first activated, which, in turn, activates $\overline{VPA}$. After the MC68000 receives $\overline{VPA}$, the processor synchronizes itself to the E clock and continues the bus cycle by asserting $\overline{VMA}$. Signal E6 recognizes that the MC6800 page has been selected, $\overline{VMA}$ has been asserted for a synchronous cycle, and the $\overline{LDS}$ is asserted indicating a bus transfer on the lower eight data bus bits. Thus, the PIA must be interfaced into the lower eight bits of the data bus when using signal E6. The interfacing PIA chips to the MPU is shown in Fig. 3.17, where PIA1 is interfaced to the speed control system and PIA2 is interfaced to the temperature control system.

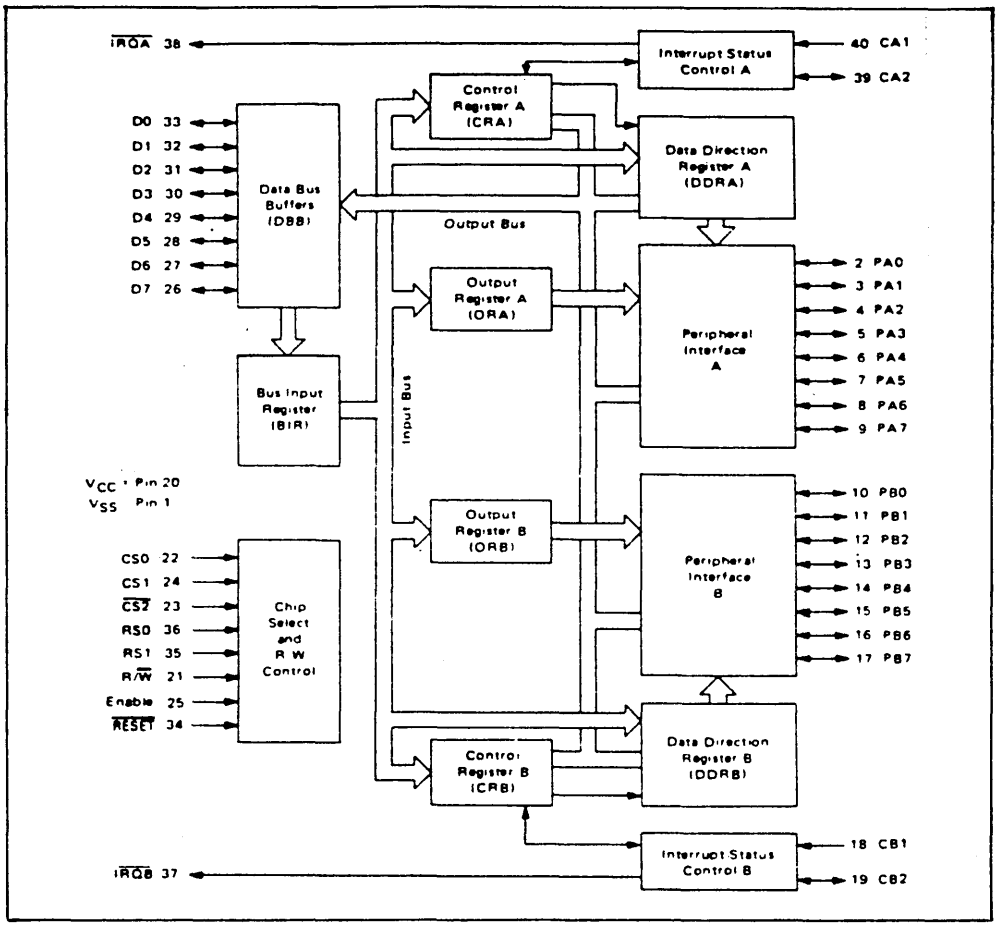


Fig. 3.14 Block Diagram of MC6821

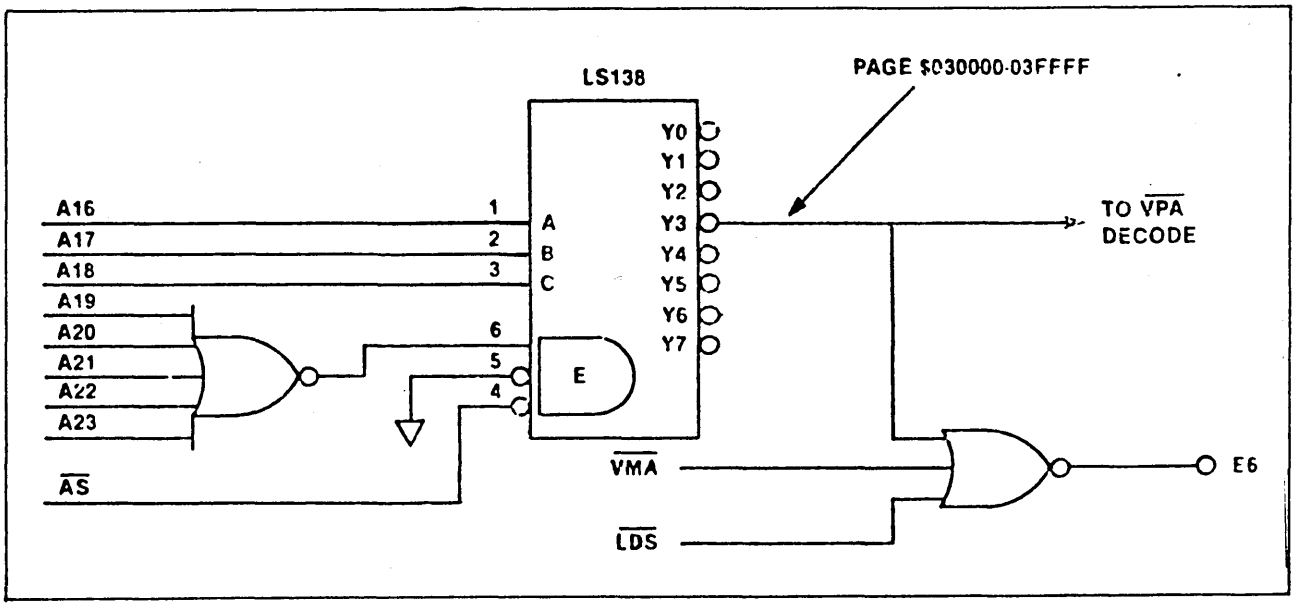


Fig. 3.15 MC6800 Page Address Signal Generation

There are six locations within the PIA accessible to the MPU data bus: two peripheral registers (PRA and PRB), two data direction registers (DDRA and DDRB), and two control registers (CRA and CRB). Selection of these locations is controlled by RS0 and RS1 (see Fig. 3.14) inputs together with bit2 in the control register, as shown in Table 3.1.

Table 3.1 PIA Internal Addressing

RS1	RS0	Control Register Bit CRA2	CRB2	Location Selected
0	0	1	*	Peripheral Register A
0	0	0	*	Data Direction Register A
0	1	*	*	Control Register A
1	0	*	1	Peripheral Register B
1	0	*	0	Data Direction Register B
1	1	*	*	Control Register B

Table 3.2 gives the PIA1 and PIA2 address map in detail.

Table 3.2 PIA Chips Address Decoding

	Register	Address bus Bit A7	A6	A5	A4	A3	A2	A1	A0	Address Decode
PRA1	DDRA	1	1	1	1	*	0	0	*	$03FFF8
	CRA	1	1	1	1	*	0	1	*	$03FFFA
	DDRB	1	1	1	1	*	1	0	*	$03FFFC
	CRB	1	1	1	1	*	1	1	*	$03FFFE
PRA2	DDRA	1	1	1	0	*	0	0	*	$03FFE8
	CRA	1	1	1	0	*	0	1	*	$03FFEA
	DDRB	1	1	1	0	*	1	0	*	$03FFEC
	CRB	1	1	1	0	*	1	1	*	$03FFEE

Note: * undefined

Each of the peripheral data lines can be programmed to act as an input or output. This is accomplished by setting a "1" in the corresponding data direction register bit for those lines which are to be outputs. A "0" in a bit of the data direction register causes the corresponding peripheral data line to act as an input. The following program is an example of configuring a PIA and data access.

```
* MC68000
* Initialization of port A as input
* Initialization of port B as output

    MOVE.B  #0,   CRA
    MOVE.B  #0,   DDRA
    MOVE.B  #4,   CRA       ; Port A = input
    MOVE.B  #0,   CRB
    MOVE.B  #$FF,DDRB
    MOVE.B  #4,   CRB       ; Port B = output


* Input data to register D1 via port A

IN: MOVE.B   PRA, D1

* Output data from D1 via port B

OU: MOVE.B  D1,  PRB
```

Analogue-Digital Conversion

An 8-bit ADC ZN449 is used to convert analogue signal to digital value [16] (see Fig. 3.17). It is a successive approximation converter with 9 μs conversion time. Fig. 3.16 shows analogue-digital converter timing. Upon receipt of a negative-going pulse at the WR input, the BUSY output goes low and the conversion starts. On the eighth negative edge of the clock pulse BUSY goes high indicating that the conversion is complete. Data can be read by taking RD low to enable the 3-state outputs.

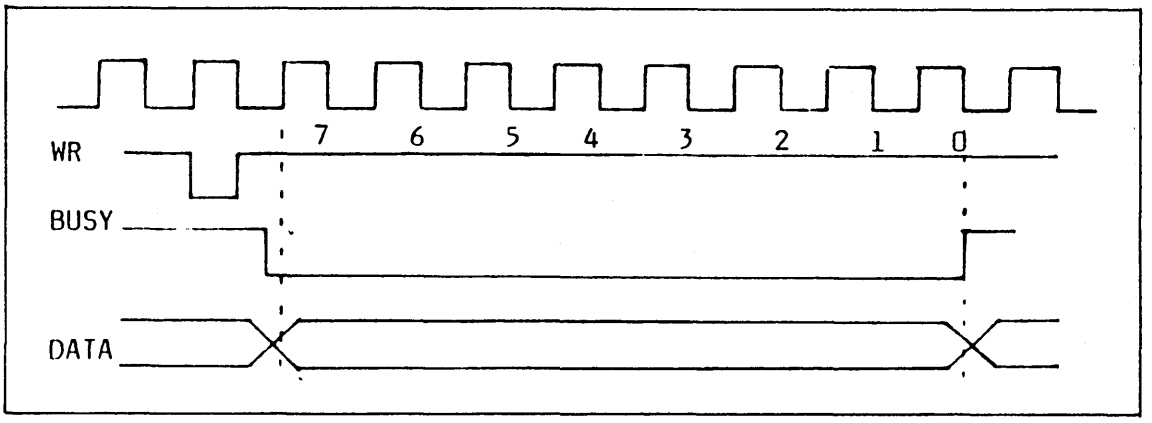


Fig. 3.16 ZN449 Analogue-Digital Converter Timing

Handshaking Between the PIA and the ADC

The control register CRA within the PIA allows the MPU to control the operation of the two peripheral control lines CA1 and CA2. In addition it allows the MPU to enable interrupt line ($\overline{IRQA}$) and monitor the status of the interrupt flags. Bits 0 to 5 of the CRA may be written or read by the MPU when the proper chip select and register select signals are applied. Bit6 and Bit7 of the CRA are read-only and are modified by exernal interrupts occurring on control lines CA1 and CA2. The format of the control words is shown in Table 3.3.

Table 3.3 Control Word Format of CRA

7	6	5	4	3	2	1	0
IRQA1	IRQA2	CA2 Control			DDRA Access	CA1 Control	

Bits 3, 4 and 5 of the CRA are used to control the CA2 peripheral control line. These bits determine if the CA2 will be an interrupt input or an output control signal. Table 3.4 gives the format of the three bits when CA2 works in the output mode to control peripheral data transfer.

Table 3.4 Control of CA2 as an Output

CRA5	CRA4	CRA3	CA2	
			Cleared	Set
1	1	0	Low when CRA3 goes low as a result of an MPU write to control register A	Always low as long as CRA3 is low. Will go high on an MPU write to control register A that changes CRA3 to 1
1	1	1	Always high as long as CRA3 is high. Will be clear on an MPU write to control register A that clears CRA3 to 0	High when CRA3 goes high as a result of an MPU write to control register A

In Fig. 3.17, CA2 is used to generate a negative-going pulse at the WR input of the ADC as a conversion starting signal. Bit0 and bit1 of the CRA are used to control the interrupt input line CA1. Bit CRA0 is used to enable the MPU interrupt signal $\overline{\text{IRQA}}$. Bit CRA1 determines the active transition of the interrupt input signal CA1. The format of the two bits is given in Table 3.5.

Table 3.5 Control of Interrupt Input CA1

CRA1	CRA0	Interrupt Input CA1	Interrupt Flag CRA7	MPU Interrupt Request IRQA
1	0	↑ Active	Set high on ↑ of CA1	Disable IRQ remains high
1	1	↑ Active	Set high on ↑ of CA1	Goes low when the interrupt flag bit CRA7 goes high

Note: ↑ indicates positive transition

In Fig. 3.17, CA1 is connected to the BUSY line of the ADC. When conversion finishes, BUSY goes high, thus enabling CRA7 to go high. During the conversion period, the program will keep testing the value of bit CRA7. If CRA7 goes high, a data input instruction will be executed. The program which initiates the analogue-digital conversion and the data input is as follows.

```
* MC68000
* Initializing an analogue-digital conversion

CON:   MOVE.B #$36,  CRA    ; Disable interrupt, CA2 goes low
                            ; initializing A-D conversion
       NOP
       MOVE.B #$3E,  CRA    ; CA2 goes high
CON1:  BTST   #7,    CRA    ; If the conversion finishes
       BEQ    CON1          ;
       MOVE.B PRA,   D0     ; If bit CRA7 goes high, input
                            ; data to register D0
```

Digital-Analogue Conversion

Two 8-bit digital analogue converters (ZN425) [16] Were used to implement the digital-analogue conversions for speed and temperature sets respectively (see Fig. 3.17). The settling time of the DAC is 1 μs, and the maximum output is 5 V corresponding to digital input $FF.

Amplifier

Amplifiers ahead of the ADC and after the DAC were needed to match the magnitudes of the signals. As shown in Fig. 3.17, the amplifiers were simply built using 741 operational amplifiers.

Fig. 3.17 shows the arrangement of the interface circuit and Plate 3 shows the interface circuit board.

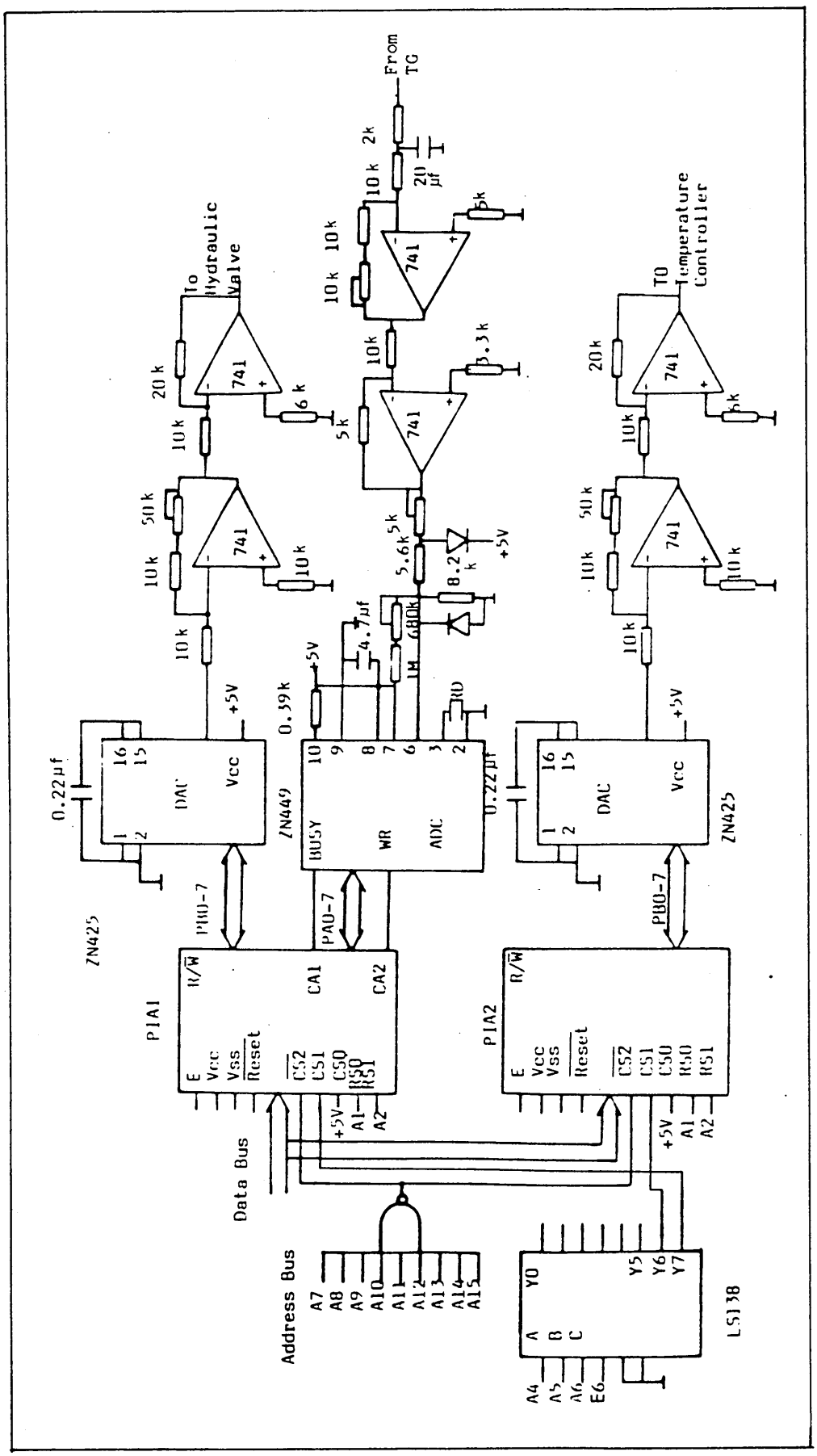


Fig. 3.17 Interface Circuit

Plate 3

3.3.2 - Generation of Interrupts

The program which executed once in each successive sampling period was interrupt driven. In the MC68000 microcomputer board, an MC68230 parallel interface/timer (PI/T) (see Fig. 3.13) was available and it was programmed to generate a periodic vectored interrupt corresponding to the required sampling period.

Fig. 3.18 shows the MC68230 block diagram [17]. The on-board timer contains a 24-bit synchronous down counter that can generate periodic interrupts, a square wave, or a single interrupt after a programmed time period. The TOUT pin is connected to the MPU interrupt request circuitry and the TIACK pin is used as an interrupt acknowledge input to the timer. The TIN pin is connected to 4 MHz system clock. The PI/T timer is

loaded from three 8-bit counter preload registers (CPRH, CPRM and CPRL). The 24-bit counter (CRH, CRM and CRL) is clocked from the output of a 5-bit (divide-by-32) prescaler. The clock source is the system clock. The counter signals the occurrence of an event primarily through zero detection (a zero is when the 24-bit counter is equal to zero). This sets the zero detect status (ZDS) bit in the timer status register and generates a timer interrupt. The ZDS bit must be reset by writing a "1" to the timer status register in that bit position to enable next interrupt. Thus, when the 24-bit counter is loaded with $FFFFFF, the interrupt interval reaches its maximum value T = 134.22 s. The minimum interrupt interval is 8 μs with "1" loaded in counter preload registers. The timer is fully configured and controlled by programming an 8-bit timer control register (TCR) in the PI/T. The format of the TCR for generating periodic vectored interrupts is shown in Table 3.6.

Table 3.6 The TCR Format

Bit	7	6	5	4	3	2	1	0
	1	0	1	0	*	0	0	1

Bit 7 6 5
Value 1 0 1 The pin TOUT is used as a timer interrupt request output. The pin $\overline{\text{TIACK}}$ is used as a timer interrupt acknowledge input. This combination supports vectored timer interrupts

Bit 4
Value 0 The counter is loaded from the counter preload register on the first clock to the 24-bit counter after zero detect, and resumes counting.

Bit 2 1
Value 0 0 Prescaler is clocked by the TIN pin input. The 24-bit counter is decremented or loaded from the CPR when the prescaler rolls over from $0 to $1F.

Bit 0
Value 0 Disable timer
1 Enable timer

To use the interrupts, an 8-bit vector number must be loaded into the timer interrupt vector register (TIVR) at the initialization stage. When an interrupt occurs, the MC68000 translates this 8-bit vector number into an address by multiplying the 8-bit number by four. At this address location, the starting address (32 bits) of the interrupt handler routine is stored. Table 3.7 gives the address map of the PI/T registers concerning interrupt generating.

Table 3.7 Address Map of the PI/T

Address	Register
$010021	Timer control register (TCR)
$010023	Timer interrupt vector register (TIVR)
$010027	Counter preload register high (CPRH)
$010029	Counter preload register middle (CPRM)
$01002B	Counter preload register low (CPRL)
$010035	Timer status register (TSR)

The program which enables the interrupt and the interrupt handler routine execution is described as follows.

```
* MC68000
* Initialization for interrupt generating

     MOVE.B   #0,  CPRH
     MOVE.B   #9,  CPRM
     MOVE.B   #$C4,CPRL  ; Set 20 ms interrupt interval
     MOVE.B   #$1B,TIVR  ; Load 8-bit vector number 1BH into the TIVR
     MOVE.L   SB,  $006C ; The starting address ( 32 bit ) of the
                         ; interrupt handler routine is stored in
                         ; the address location of $006C
     MOVE.B   #$A1,TCR   ; Initialize the TCR and enable the timer


* Interrupt handler routine

     MOVE.B   #1,  TSR  ; Enable next interrupt
```

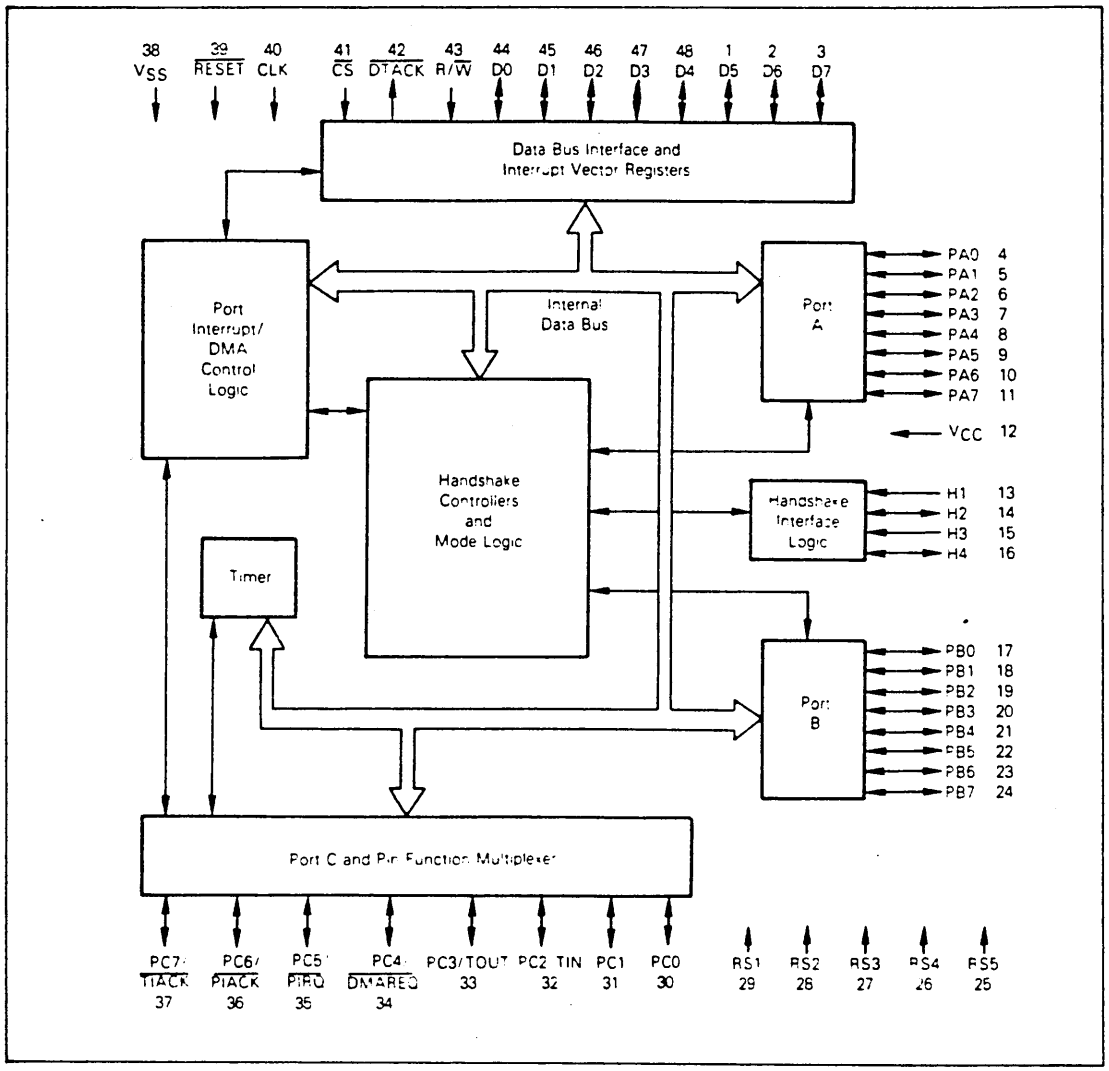


Fig. 3.18 Block Diagram of MC68230

3.4 - Temperature Control of the Polymer Melt

As mentioned before, both polymer hopper and polymer chamber were heated by electrical heater bands. A simplified method is taken here in evaluating the transfer function for the polymer hopper and the polymer chamber in order to design the control system. Fig. 3.19 shows the schematic diagram of the polymer chamber heating process.

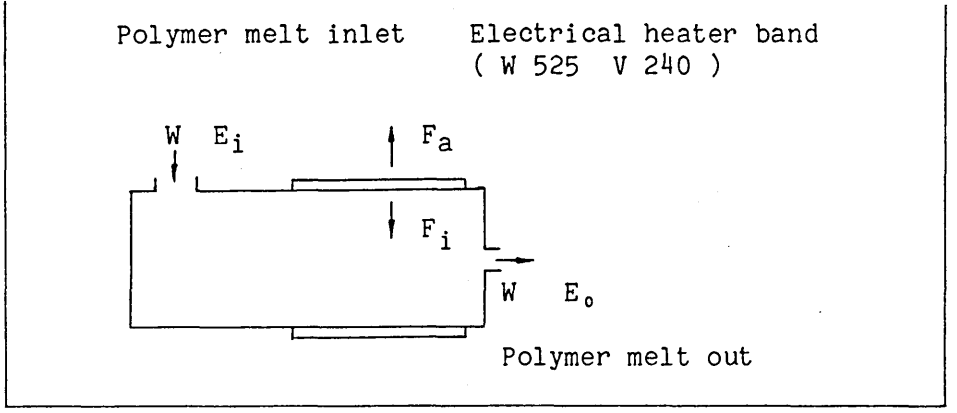


Fig. 3.19 Polymer Melt Chamber Heating Process

Where E_o the temperature of the polymer melt within the chamber.

E_i the temperature of entering polymer melt.

W the rate (both entering and leaving) of the polymer melt mass.

F_i the rate of heat supply from the electrical heater.

F_a the rate of heat loss to surroundings.

Note that the thermal parameters have been considered to be lumped [18], which means that the polymer melt temperature has been assumed to be uniform within the polymer chamber. If the temperature were to be considered as functions of both time and position, it would be necessary to describe the system by partial differential equations. However,in this case, the result derived from the lumped parameter model proved accurate enough for the design of the temperature control system. Under the assumption that a small change f_i occurs in F_i, and then it causes small changes in F_a and E_o, denoted by f_a and e_o respectively, the energy balance for the system gives

$$C\frac{de_o}{dt} = - f_a + f_i - Cp\,W\,e_o \qquad (3.1)$$

Where C_p is the specific heat capacity of the polymer melt.

C is the capacitance of the polymer melt within the chamber,

i.e., $C = G\,C_p$

G is the mass of the polymer melt within the chamber

The capacity of the wall of the polymer chamber is negligible.

Taking Laplace transforms for equation (3.1) and assuming zero initial conditions yields equation (3.2).

$$E_0(s) = \frac{RF_i(s)}{1 + RCs} - \frac{RF_a(s)}{1 + RCs} \tag{3.2}$$

Where $R = 1 / C_p W$

Consider the heat loss as illustrated in Fig. 3.20

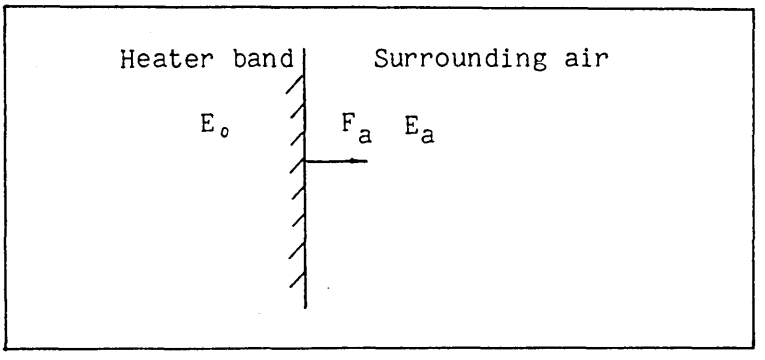


Fig. 3.20 Heat Loss to Surrounding

Taking the change of the ambient temperature E_a as zero and defining the constant $R_1 = 1 / A\,U$, where A is an effective heat transfer area and U is the overall convection heat transfer coefficient, we get

$$e_o = R_1\, f_a \tag{3.3}$$

Taking Laplace transforms for equation (3.3) gives

$$E_o(s) = R_1\, F_a(s) \tag{3.4}$$

Combining equation (3.2) and equation (3.4) gives

$$E_0(s) = \frac{RF_i(s)}{1 + RCs} - \frac{RE_0(s)}{R_1(1 + RCs)} \tag{3.5}$$

Rearranging equation (3.5) gives

$$E_0(s) = \frac{R R_1 F_i(s)}{(R_1 + R) + R_1 RCs} \tag{3.6}$$

Now, since $F_i(s)$ can approximately expressed by

$$F_i(s) = K\, U(s)$$

Where U(s) is the Laplace transform of input voltage of the electrical heater band. Equation (3.6) becomes

$$E_0(s) = \frac{R R_1 K U(s)}{(R_1 + R) + R_1 RCs} \tag{3.7}$$

Considering the transport delay for heat transfer, we introduce dead time into equation (3.7).

$$E_0(s) = \frac{R R_1 K U(s)}{(R_1 + R) + R_1 RCs}\, e^{-\tau s} \tag{3.8}$$

Where τ is the dead time.

Thus, the thermal dynamic systems both for polymer hopper and polymer chamber can be approximately described by a first order process with dead time. Fig. 3.21 shows the temperature dynamic response of the polymer chamber heating process.

The control of a first order process can be simply implemented by a single-loop feedback control system with a controller such as PI, PID, or two position controller and a controllable power supply. Here two controllable power supplies with built in two position controllers were used to control the temperature of the polymer hopper and the polymer chamber respectively. Fig. 3.22 shows the block diagram of the control system.

Fig. 3.23 shows the closed loop dynamic response of the temperature measured from the thermocouple at position 2 (see Fig. 3.11). The

polymer in the melt hopper and melt chamber melted very quickly, but it took 80 minutes for the polymer to reach a uniform distribution and steady temperature level before the drawing process could be actually commenced. At steady state, the temperature was controlled within ±3°C of the set temperature. The setpoint of the temperature control system was given by the microcomputer.

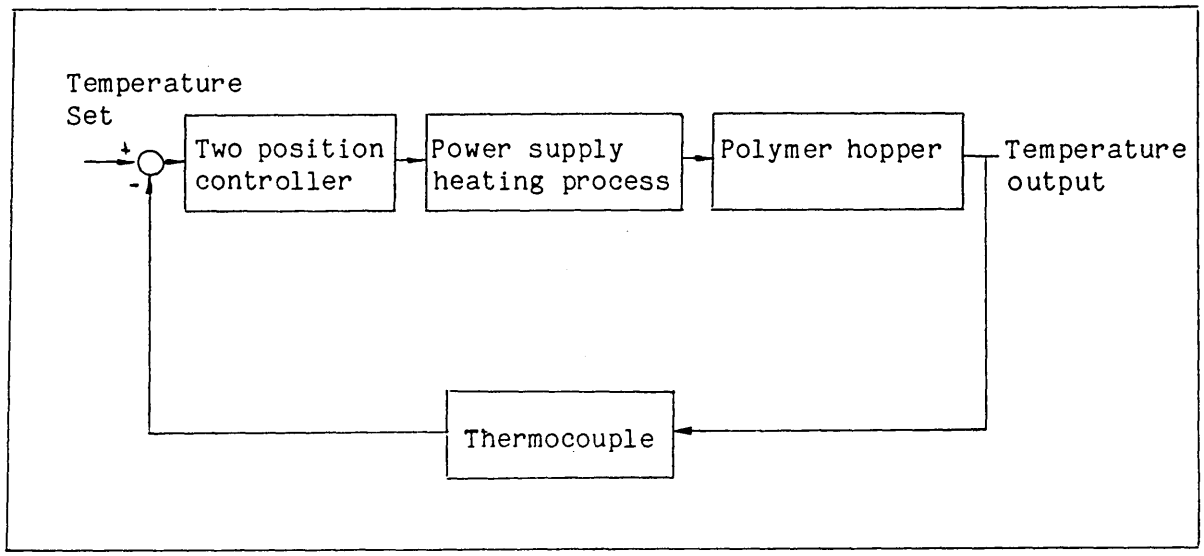


Fig. 3.22 Polymer Chamber Heating Control System

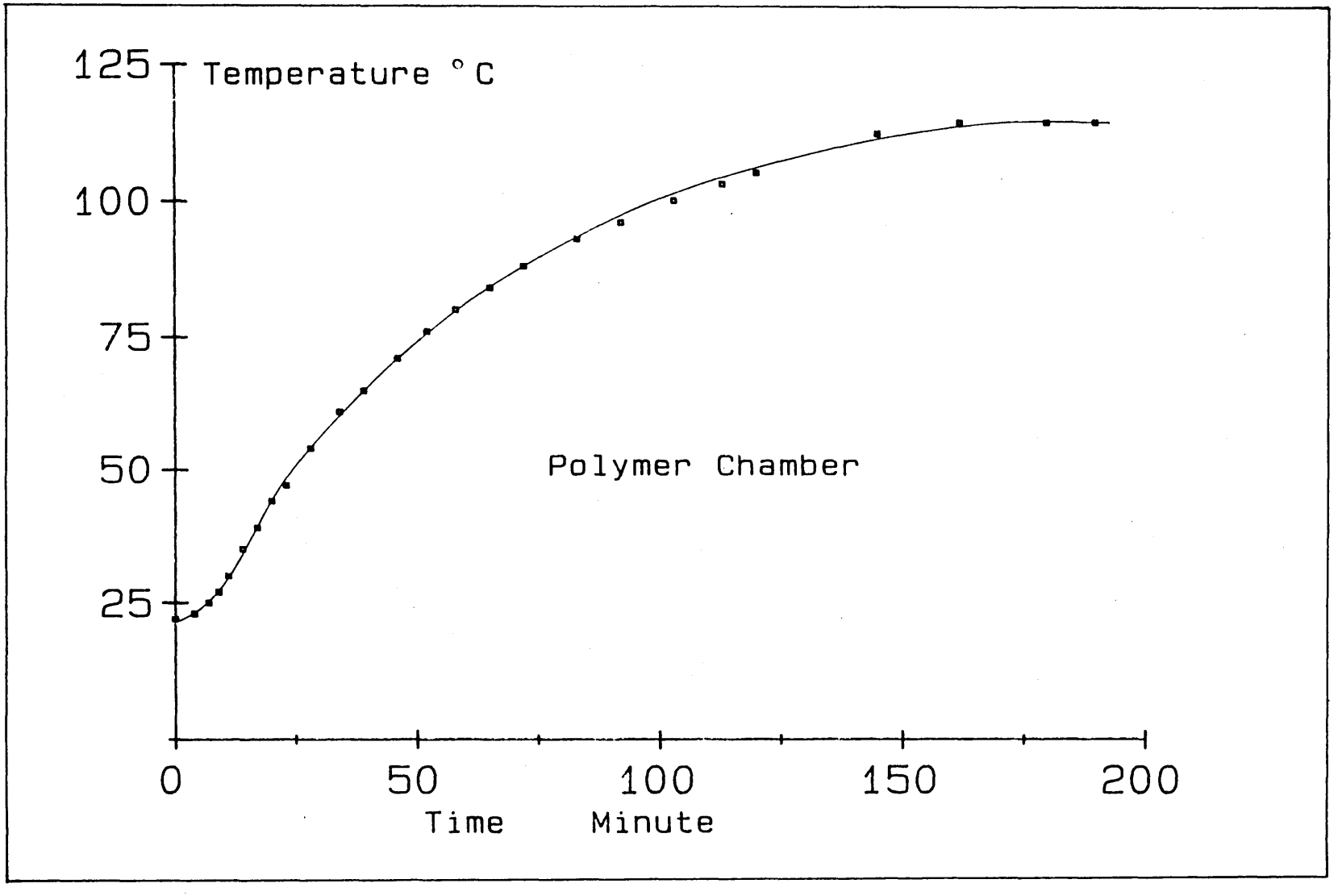


Fig. 3.21 Step Response of the Polymer Melt Chamber
Heater Band Step Input 100 V

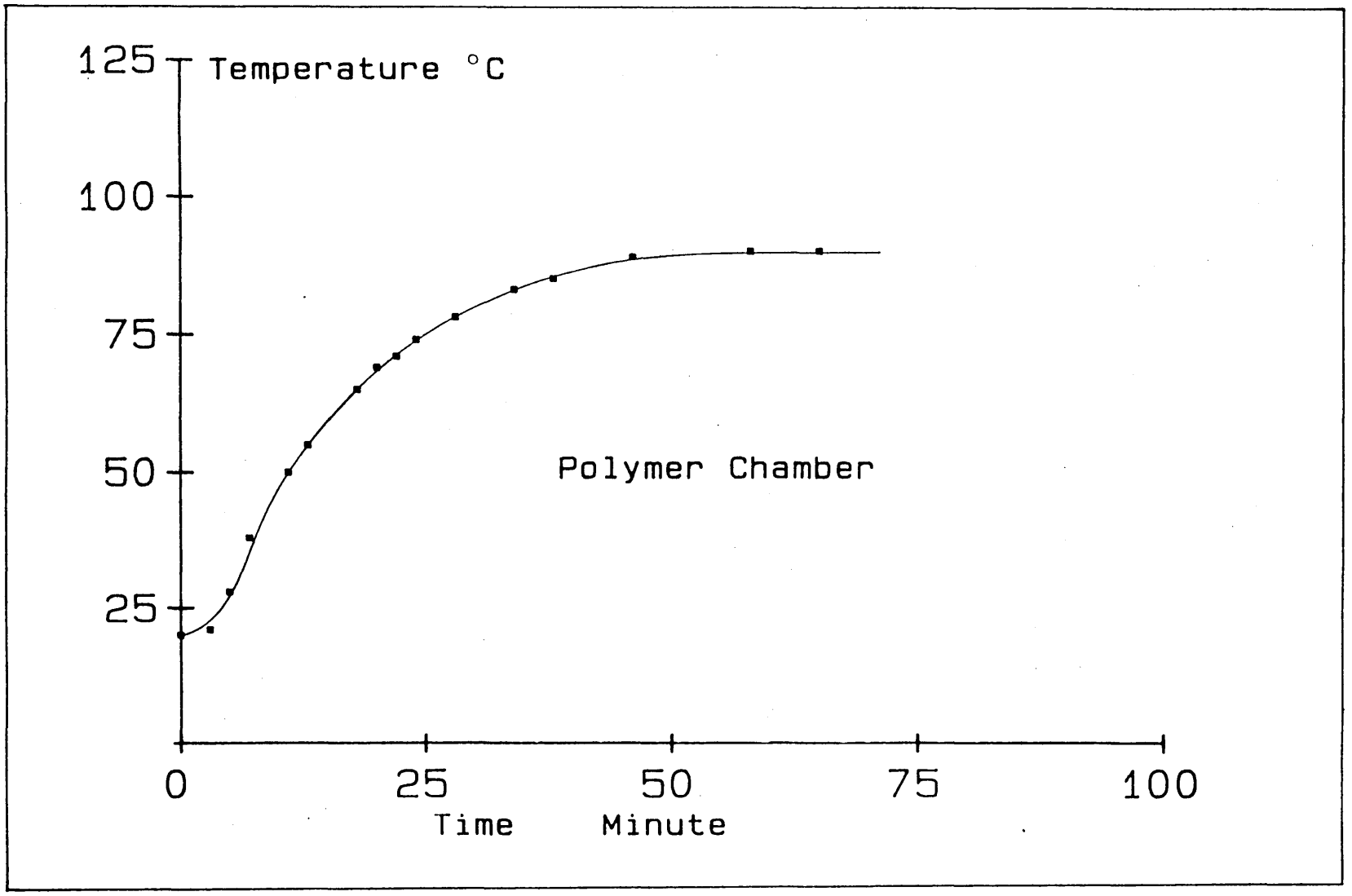


Fig. 3.23 Closed-loop Step Response of the Polymer Chamber Heating Process (Set Temperature 90°C)

As mentioned before, in the wire drawing process, it was required to adjust drawing speed over a given range with a desired accuracy and repeatability. A MC68000 microcomputer was used to implement the sampled data feedback control system. The schematic diagram of the microcomputer based system for controlling the speed of the hydraulic motor is shown in Fig. 3.24.

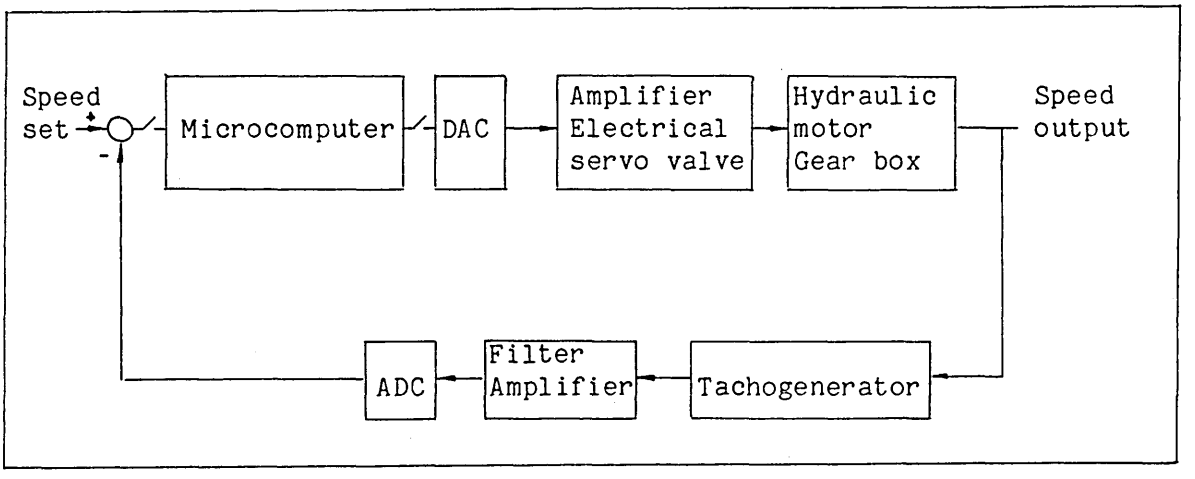


Fig. 3.24 Electro-Hydraulic Motor Control System

The system was not continuous, but measured and corrected the motor speed at discrete intervals. The order of the events in a complete control cycle was as follows.

a) The actual speed measured from the tachogenerator, after being filtered and amplified, was converted into a digital number via the ADC.

b) The digital feedback signal was compared with the setpoint to produce an error.

c) The digital controller programmed in the computer generated a compensation signal.

d) The compensation signal then was transmitted to the

electro-hydraulic servo valve via the DAC and the amplifier to vary the speed of the electro-hydraulic motor.

3.5.1 - Major Components of the Speed Control System

Electro-hydraulic Servo Valve

The servo valve is the main part of a hydraulic servo and its performance significantly affects the overall performance of the system. The servo valve in an electro-hydraulic system is essentially a component which produces hydraulic flow proportional to the electrical current applied. It is often the case that the transfer function of the servo valve can be described by a first order system [19], that is,

$$G_{sv}(s) = \frac{K_{sv}}{1 + T_{sv}s}$$

Hydraulic Motor

The function of the hydraulic motor in a hydraulic control system is to convert the hydraulic energy supplied by the pump into mechanical energy at its output shaft. The transfer function of a hydraulic motor is given by

$$G_M(s) = \frac{n(s)}{Y(s)} = \frac{K_m \omega_m^2}{s^2 + 2\zeta\omega_m s + \omega_m^2}$$

Where n(s) is the velocity output

Y(s) the flowrate of the valve

K_m the steady state gain

ω_m the natural frequency and ζ the damping ratio

Fig. 3.25 shows the step response of the hydraulic drive system (servo valve and motor), from which the system could be described by a second order system with ω_m = 36.6 rad s^{-1} and ζ = 0.68.

The rest of the components of the system include a tachogenerator with transfer function $K_g/(1 + T_g s)$, gear box with transfer function 3/29, two amplifiers before the ADC and after the DAC respectively with transfer functions of K_1 and K_2.

The gain of the process including the DAC, the amplifier, the hydraulic drive system, the tachogenerator and the ADC was set to 1. Fig. 3.26 shows the sampled data / output data of the microcomputer, from which it can be seen that the system has a good linearity.

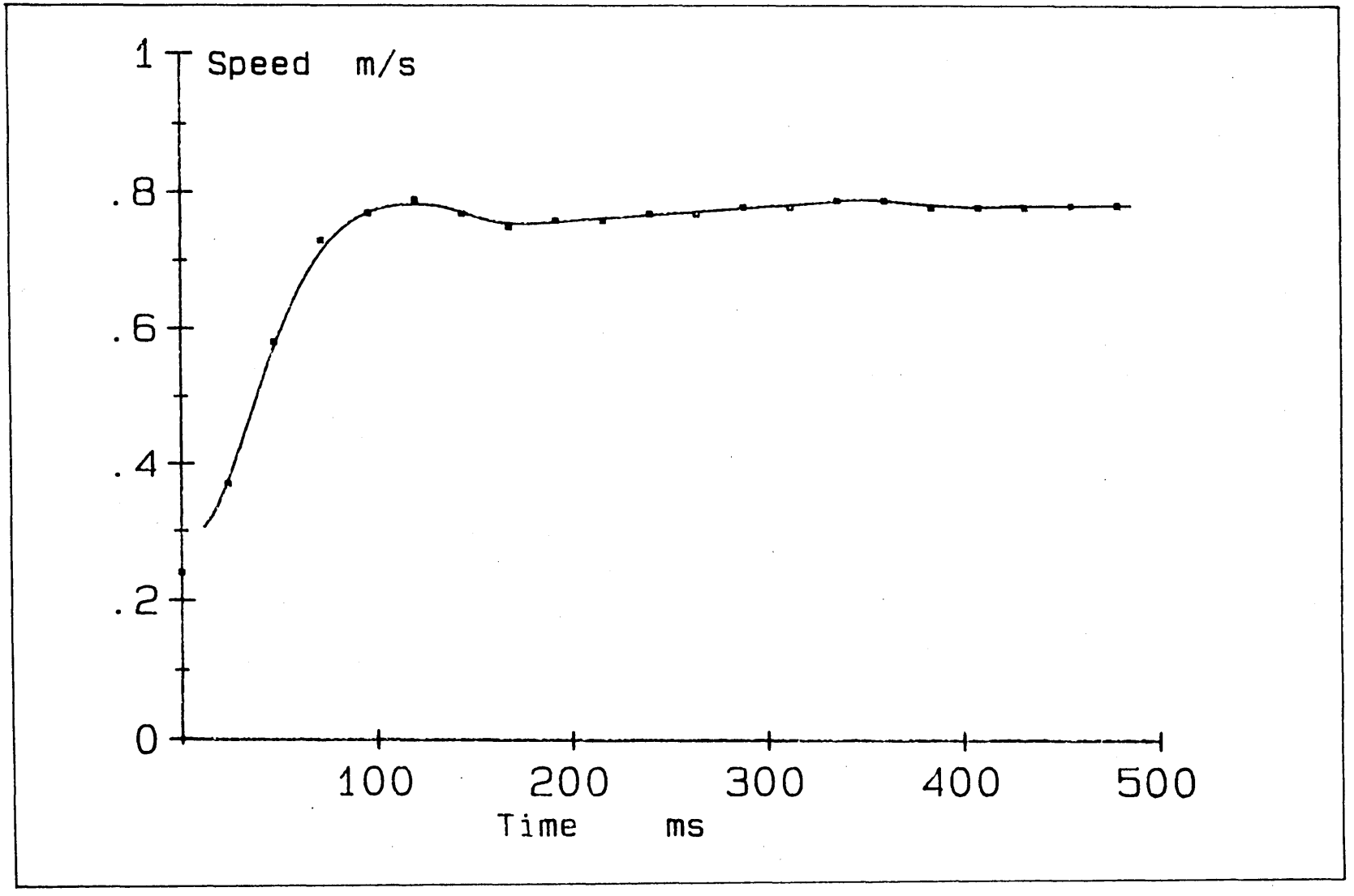


Fig. 3.25 Step Response of the Hydraulic Drive System

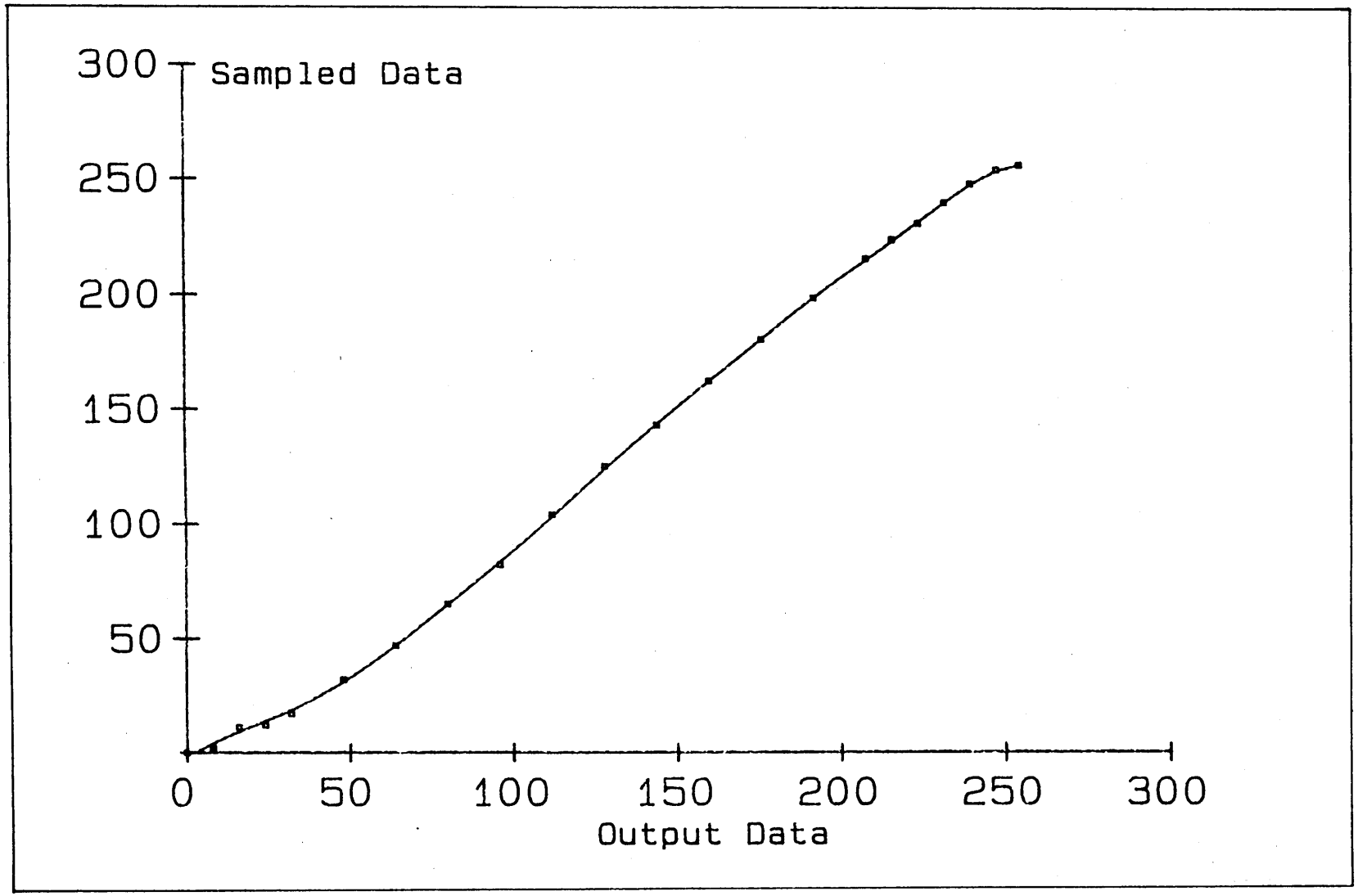


Fig. 3.26 Microcomputer Sampled Data / Output Data

Sampling Period

The sampling period is an important design parameter for all sampled data control systems. There are several rules of thumb which can be used to determine the sampling period [20]. One is to relate the sampling period T to the desired rise time t_r, and t_r/T should be in the range of 2-4 in order to get a good servo performance of the closed-loop system. Alternatively assuming a sampling period of one tenth the sum of loop dead time plus first order time constant is often a good compromise. Some typical values of sampling periods obtained by experience can be easily found in published works [21]. For example, a typical speed control loop generally requires 0.01-0.02 second sampling period. Here, the sampling period for hydraulic motor speed control system was chosen in a range of 8 to 15 ms

Integral Windup (Reset Windup) and Its Correction

A digital integrator was incorporated in the forward path of the system to eliminate steady state error. With such a system, it was possible that the integrator could assume very large values if the control signal saturated when there was an error. This, referred to as the integral windup effect, could cause excessive overshooting and stability problems. Thus the controller had to be modified to give some degree of windup protection. This was accomplished by disabling the integrator whenever its output signal caused saturation in the actuator. In a digitally implemented controller such modification is easily available through software changes. In this case, the integrator was

disabled by the program when its output value reached $F0 corresponding to the maximum open position of the electro-hydraulic valve, as shown in Fig. 3.27. Here, $F0 is the saturation value, and whenever the controller output U reaches saturation value $F0, the integrator input is set to zero.

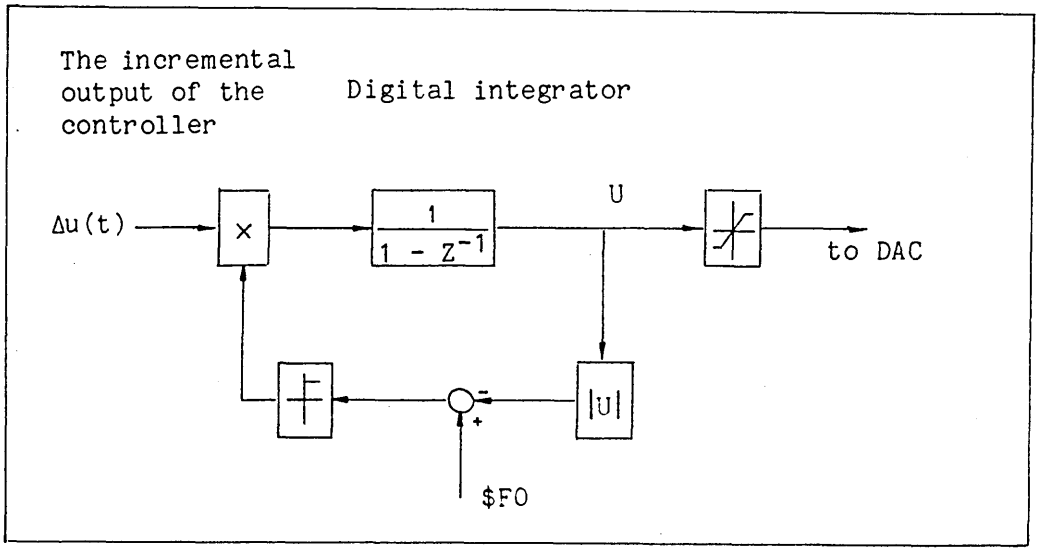


Fig. 3.27 Anti-Reset-Windup for Integrator

The program executing this function goes as follows

```
* Anti-integral windup, the initial value of the integrator is in the
* memory location of $U1, incremental output in D4, $F0 the saturation
* level.

Star:   ADD  $U1,   D4     ; Integrating
        CMP  #$F0,  D4     ; If D1>$F0
        BGT  STAR1
        CMP  #$-F0, D4     ; If D1<$-F0
        BLT  STAR2
        BRA  STAR3
Star1:  MOVE #$F0,  D4     ; D1=$F0
        BRA  STAR3
Star2:  MOVE #$-F0, D4     ; D1=$-F0
Star3:
```

Filtering

The analog feedback signal coming from the tachogenerator contained wide bandwidth noise of relatively large magnitude due to serious commutation problems. A simple analogue RC low-pass filter, as shown in Fig. 3.17, $1/(1+T_f s)$, with T_f chosen to be 0.04 second, had to be used to attenuate sufficiently the noise component. However, its dynamics somewhat degraded the response of the control system. An additional stage of digital low-pass filter (after the sampler) was also programmed into the microcomputer, in order to improve the performance of the system. The digital filter given by equation (3.9) is a version of a simple first order lag, where T is the sampling period, τ is the time constant of the first order lag, and τ can be simply chosen to be about T/2 [21].

$$Y(n) = \frac{1}{Q} X(n) + (1 - \frac{1}{Q}) Y(n - 1) \qquad (3.9)$$

where $$Q = \frac{1}{1 - e^{-T/\tau}}$$

Accuracy and Word Size

The component introducing the largest error into the control system was the tachogenerator with accuracy of about ±1 percent. Thus the whole accuracy of the control system could not be better than ±1 percent. An 8 bit ADC giving a resolution of 0.4 percent was used, and 16 bit or 32 bit word length was employed to represent a number and to handle the mathmatical operation. However, the final words were truncated to 8 bits, the DAC size. A hand micrometer was used to measure the wire

diameter and ±0.005 mm error was allowed in the reading, therefore ±2 percent error was estimated in the reduction in area of the wire. The maximum error of the whole speed control system was designed within ±2 percent, which only caused less than ±1 percent error in the reduction in area of the wire, so the accuracy of the control system could meet the demand of the allowed error in the reduction in area of within ±2 percent.

3.5.3 - Analysis of the Control System

Fig. 3.28 shows the open loop step response of the process including the amplifier, the servo valve, the hydraulic motor, the gear box and the tachogenerator, from which the transfer function of the process can be approximately given by

$$G(s) = \frac{e^{-0.012s}}{(0.06s + 1)(0.01s + 1)}$$

and

$$G(Z^{-1}) = \frac{Z^{-2}(0.088 + 0.043Z^{-1})}{1 - 1.11Z^{-1} + 0.243Z^{-2}}$$

with sampling period 0.012 second.

The Z transfer function of the digital PID controller employed in the closed loop is given by

$$G(Z^{-1}) = \frac{g_0 + g_1Z^{-1} + g_2Z^{-2}}{(h_0 + h_1Z^{-1})(1 - Z^{-1})}$$

Where $h_0 = 0.3$, $h_1 = 0.19$, $g_0 = 1.33$, $g_1 = -1.37$, and $g_2 = 0.125$. The design of the digital PID controller will be described in chapter 4. Fig. 3.29 shows the block diagram of the speed control system.

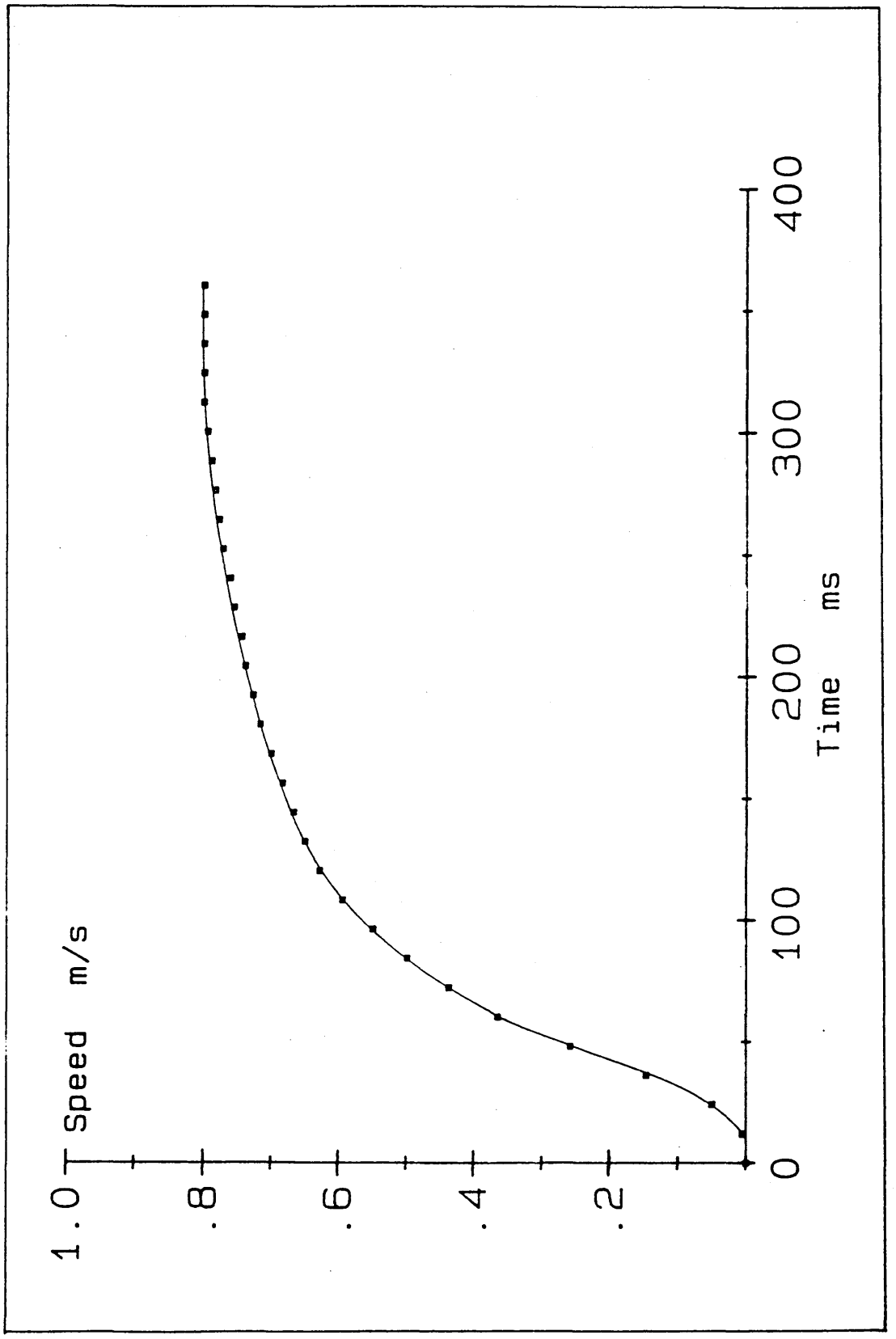


Fig. 3.28 Step Response of the Drive System

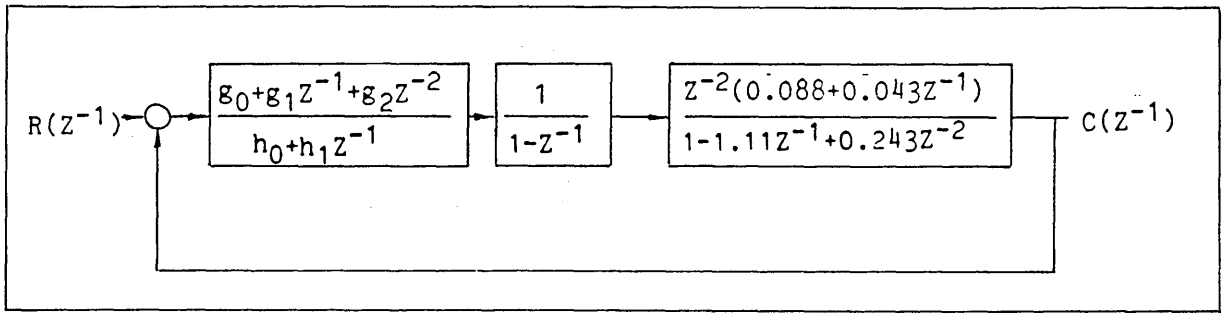


Fig. 3.29 Block Diagram of the Speed Control System

The stability of the system can be determined from the locations of the roots of the characteristic equation given by

$$30X^5 - 44.3X^4 + 12.2X^3 + 12.02X^2 - 9.4X + 0.54 = 0$$

from which the roots are found to be 0.94, -0.596, 0.0624 and $0.53 \pm 0.95j$. Thus any root of the characteristic equation has a magnitude less than unity, and the system is stable. Fig. 3.30 shows the the closed loop step response of the speed control system.

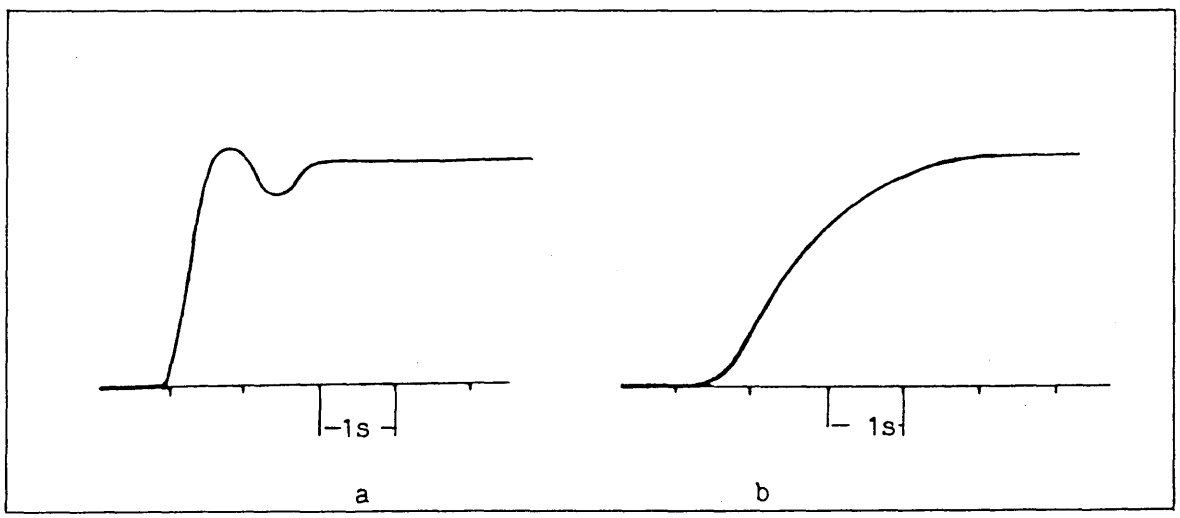


Fig. 3.30 a) Closed-Loop Step Response of the Speed Control System
b) Closed-Loop Response of the Speed Control System with Same Step Input Applied, via a Digital Filter

3.6 - Software Configuration

Based on the experimental results, polymer WVG 23 at 110°C set temperature, polymer ELVAX 650 at 150°C set temperature and polymer Nylon 6 at 260°C set temperature were chosen as pressure media for the dieless wire drawing.

3.6.1 - Producing Wire of a Uniform Diameter

Before the drawing process is started, the specification is input by following a prompt on the microcomputer screen: " Input the polymer number (1-WVG, 2-ELVAX, 3-NYLON) in memory location of $2000, the desired value of PRA in memory location $2002, then run the program starting address $2100 ".

Depending upon the input data, the microcomputer determined the temperature and the drawing speed setpoints via built-in look-up tables. The look-up tables were based on the experimental results, for example, as shown in Fig. 3.31, in which the incremental zone of the curve was portioned into 5 segments and each nonlinear segment was replaced by a linear approximation. Thus, only the values of the six break points needed to be stored in the microcomputer, and the speed setpoint corresponding to a given PRA could be calculated from a linear equation. A simplified flowchart as shown in Fig. 3.32 can easily explain the nature of the program.

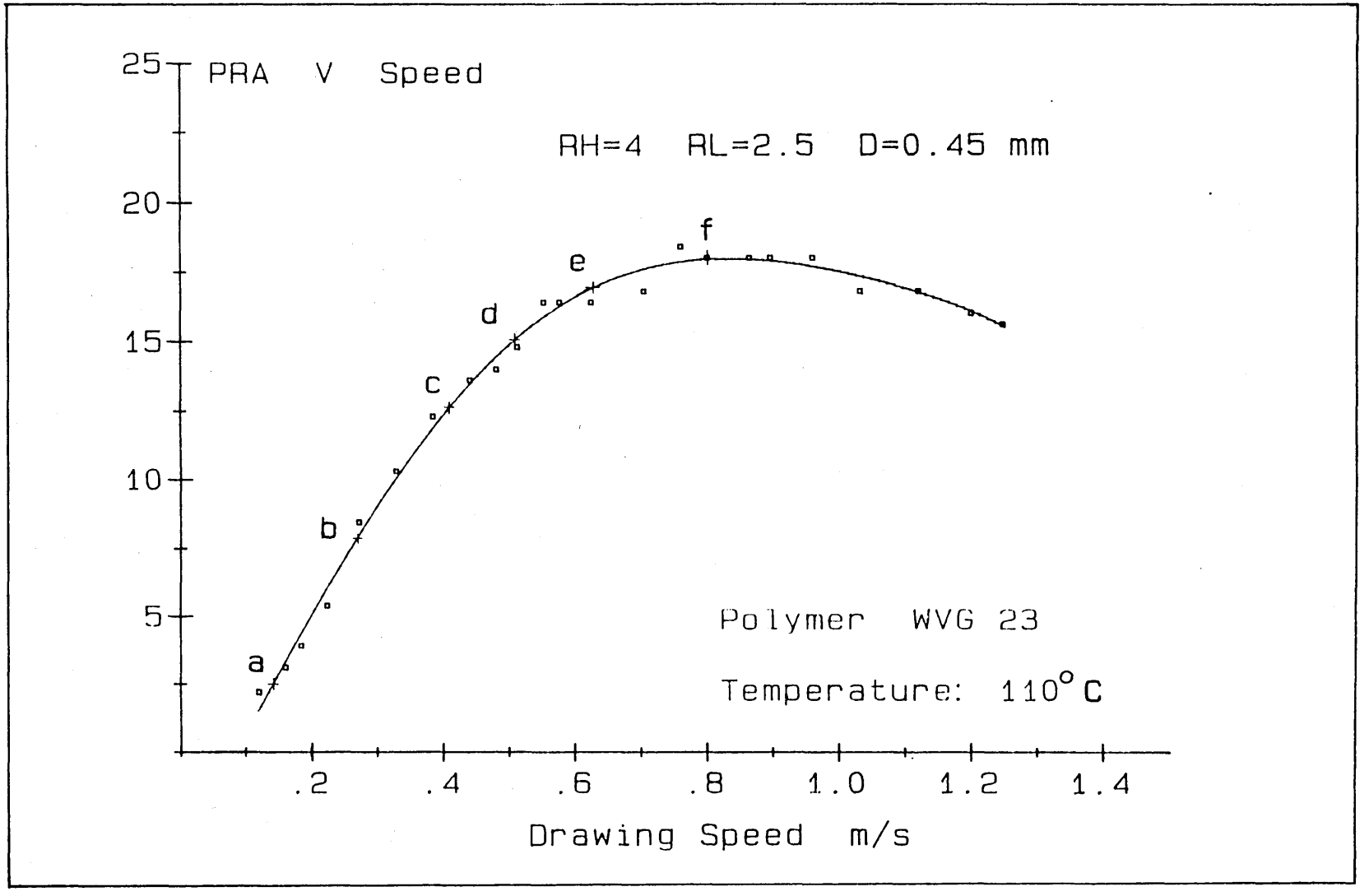
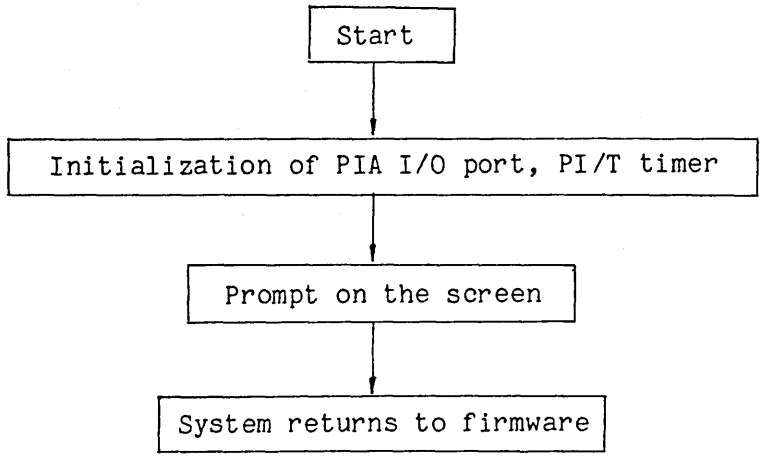
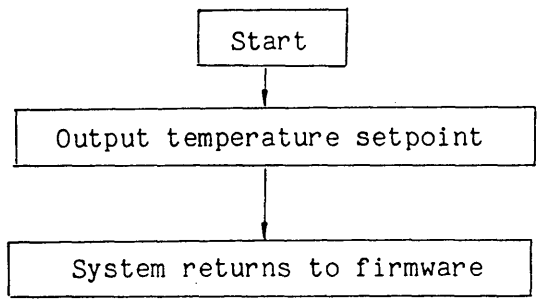


Fig. 3.31 The Percentage Reduction in Area



Input the value of the desired PRA and the polymer number via the keyboard.



Wait for the increase in temperature and the uniform distribution of the temperature within the DRU.

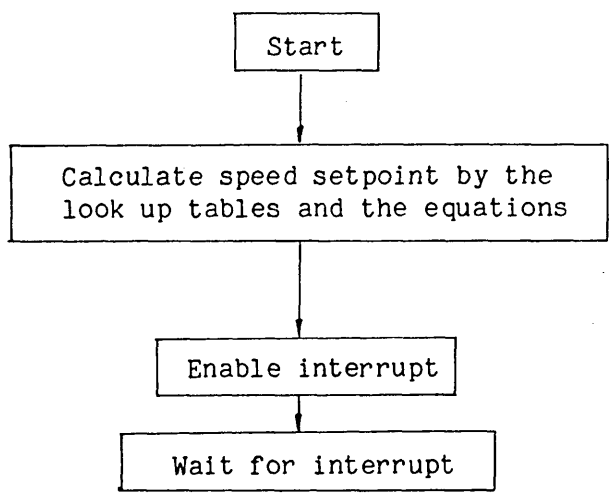


Fig. 3.32 Flowchart of Drawing Wire of a Uniform Diameter

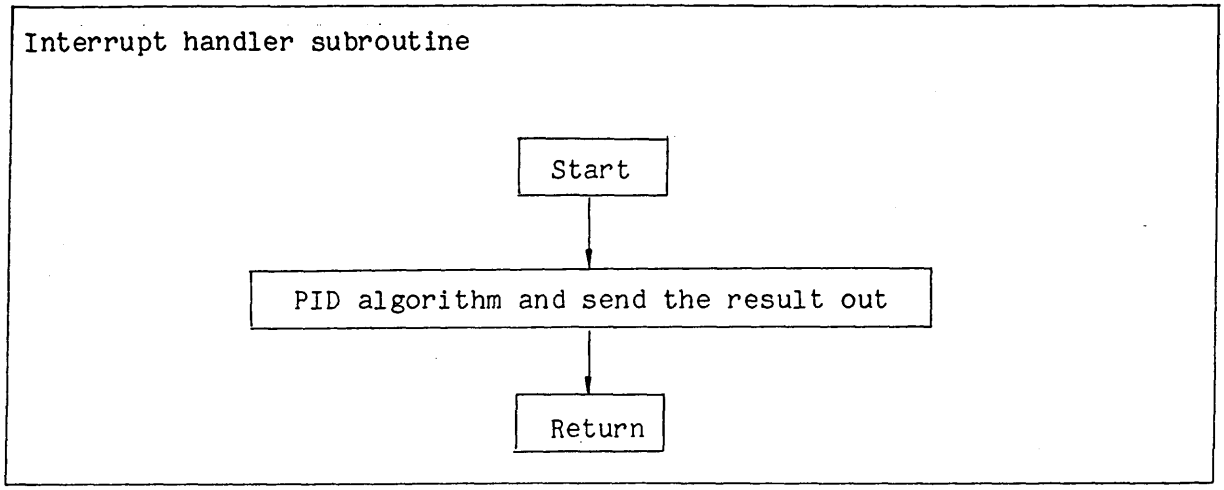


Fig. 3.32 Flowchart of Drawing Wire of a Uniform Diameter

After the microcomputer had displayed the prompt on the screen, instructions in the user program returned control of the system to the firmware. Data could then be input via the keyboard, and the user program was restarted by the operator. After the temperature reached a steady state (usually 80 minutes), the wire drawing process was initiated by the program. The program listing is given in Appendix 5.

3.6.2 - Producing Tapered Wire

The lack of a PRA transducer in the control system caused some complexity in drawing tapered wires. Here only the linear portion of the PRA curve was used to produce tapered wire. Because of its good linearity properties (as shown in Fig. 3.6), polymer ELVAX at 150°C was chosen as the best pressure medium to produce tapered wires. The drawing speed was controlled to increase with a properly selected rate to produce the wire with a uniform change in area along the length of the wire. The specifications of the required tapered wire were PRA1 and PRA2, corresponding to the values of the PRA at the large and the small end of the wire respectively, and L, the desired length of the tapered

wire. From Fig. 3.6, the speeds V_1 and V_2 corresponding to the given PRA1 and PRA2 respectively could be calculated by equation (3.10).

$$V = \frac{(PRA - 0.045)}{0.1584} \qquad (3.10)$$

As mentioned before, the speed control system facilitated a drawing speed variable between 0 and 1.3 ms^{-1}. For 8-bit length of word, the step size of the speed was given by

$$1_{1sb} = \frac{1.3}{255} = 0.0051 \ ms^{-1}$$

Thus the numbers N_1 and N_2 corresponding to V_1 and V_2 respectively were

$$N_1 = \frac{V_1}{1_{1sb}}$$

and

$$N_2 = \frac{V_2}{1_{1sb}}$$

The drawn length l in every speed step should be equal and was given by

$$l = \frac{L}{(N_2 - N_1)}$$

Interrupt with interval T was used to count the time passed and so at any speed step N, the drawing time n in terms of number of the interrupt interval was given by

$$n = \frac{l}{1_{1sb} N T}$$

$$= \frac{L}{(N_2 - N_1) 1_{1sb} T N}$$

The drawing process started at the lowest speed N_1 with drawing time $n_1 = c/N_1$, where $c = L / ((N_2 - N_1) 1_{1sb} T N)$. After that, the microcomputer increased the speed step by 1, i.e., N_1+1, and calculated the drawing time $n_2 = c/(N_1+1)$, this sequence continued step by step until the final speed step N_2 was completed and the process stopped.

During the above procedure, the microcomputer not only calculated

the drawing speed and time, but executed the PID action simultaneously to ensure that the actual drawing speed would follow the speed setpoint.

Fig. 3.33 shows the flowchart of the program. The first step to start a drawing process was to input the specifications of the wire. Then the microcomputer calculated the starting speed step N_1, the final speed N_2 and the constant c, initialized the I/O ports and the timer, and started the drawing process. The interrupt handler routine decided whether the PID action was to be executed and the speed step was to be increased. Having finished the drawing process, the microcomputer turned to execute an alternative interrupt handler routine which would enable hydraulic motor to stop with a somewhat slower rate. The program listing for the tapered wire drawing is given in Appendix 6.

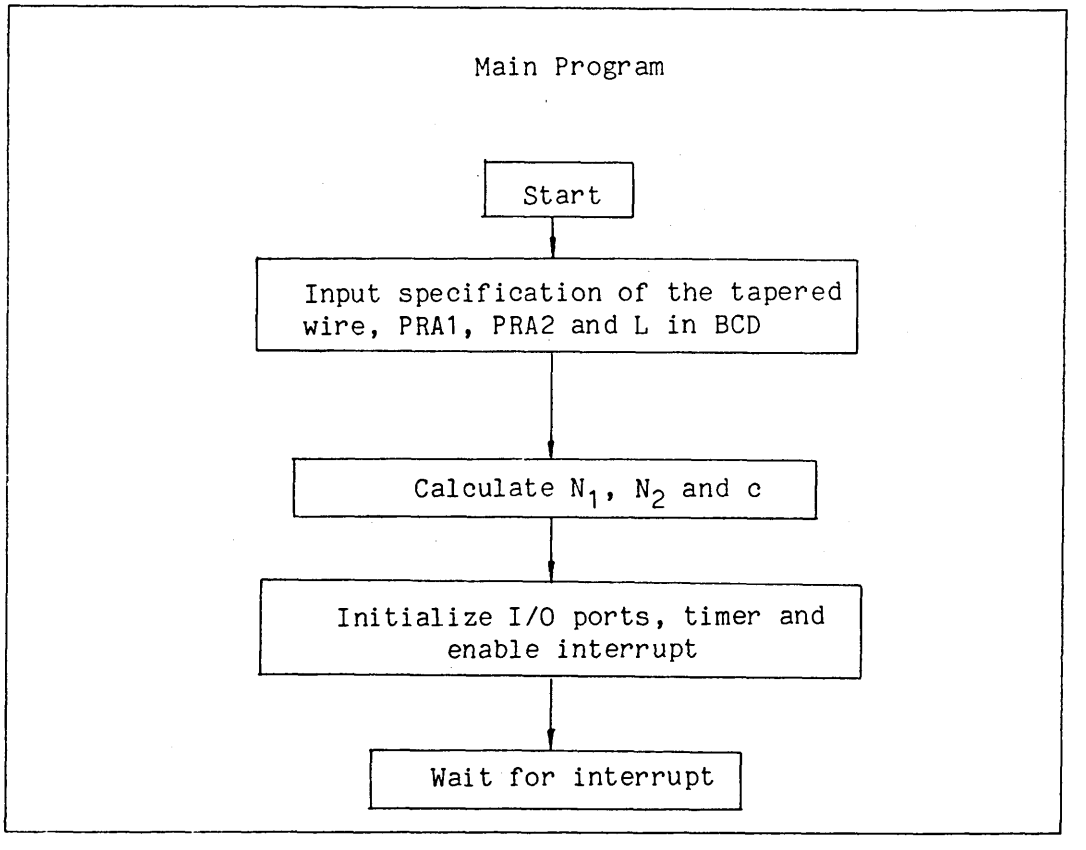


Fig. 3.33 Flowchart of the Tapered Wire Drawing Program

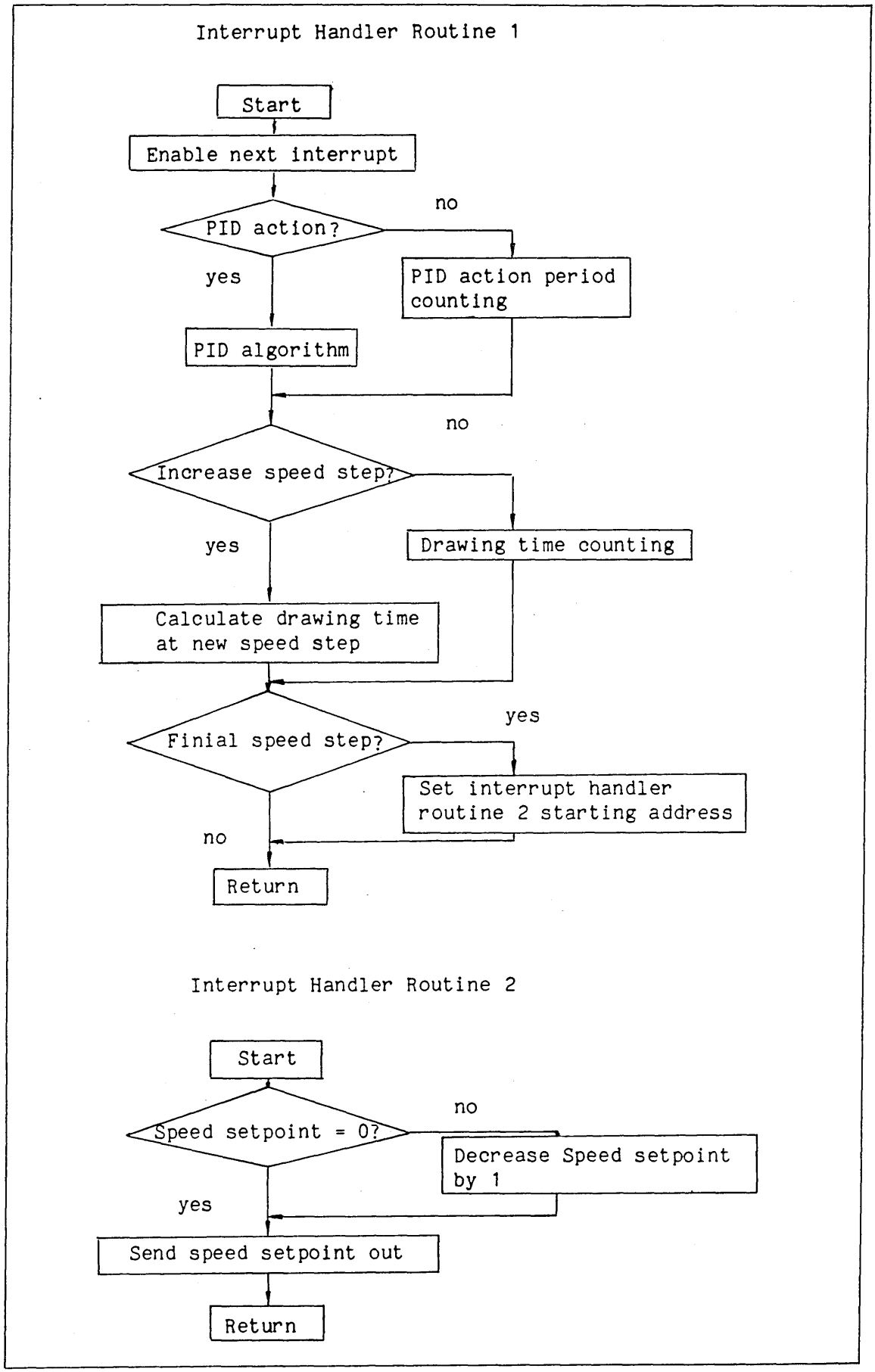


Fig. 3.33 Flowchart of the Tapered Wire Drawing Program

3.7 - The PRA Transducer

To implement a closed-loop control system in which the controlled variable is the PRA of the wire and the manipulated variable is the drawing speed, a transducer is needed to make the information available from a measurement of the PRA. In closed-loop control, since the accuracy of the system output, i.e., the quality of the product will be decided by the accuracy with which the measurement is performed, emphasis must be given to the development of a new measuring method with desired accuracy.

As mentioned before, when the wire is pulled through the DRU, there is a polymer coating on the surface of the wire. Therefore the diameter of the wire can not be measured with a conventional measuring method. A method was developed with which the reduction in area of the wire could be measured. The new design utilises two rotary shaft encoders, one on each side of the DRU, as shown in Fig. 3.34a. The encoders are driven by the wire being drawn via friction wheels mounted on the encoder shafts. The encoders generate two trains of pulses whose frequencies are proportional to the speeds of the wire at the inlet and exit of the DRU respectively. The flow continuity law gives

$$V_1 S_1 = V_2 S_2$$

Where V_1 - the speed of the wire at the inlet of the DRU

V_2 - the speed of the wire at the exit of the DRU

S_1 - the cross section area of the wire before the DRU

S_2 - the cross section area of the wire after the DRU

So the reduction in area of the wire is given by

$$PRA = (1 - S_2 / S_1) * 100$$

$$= (1 - V_1 / V_2) * 100$$

By measuring the speeds of V_1 and V_2, the microcomputer can easily calculate the PRA on line.

3.7.1 - Hardware

Fig. 3.34a shows the schematic diagram of the transducer and Plate 4 shows its arrangment on the bench. The major components of the transducer are described as follows.

The rotary shaft encoder (Type EA6-CW100) used was a subminiature optical incremental encoder with resolution of 100 pulses per revolution and bi-directional output. It provided a train of pulses with a frequency proportional to the angular velocity

A Schmitt trigger (HEF 4093B) used to shape the pulse train received from the encoder to produce a clean square wave at the output of the pulse shaper. The square wave was fed to a ÷ 4 frequency divider implemented on a 7-bit binary counter (HEF 4024B) as shown in Fig. 3.35. This provided averaging of the indicated speed over two encoder pulses.

To reduce the influence of the stray noise on the signal from the encoder, a 12 V power supply was required for the encoders. The outputs of the dividers, however, could not be directly connected to the MC6840, because the MC6840 required a TTL voltage. Hence optoisolators (Quadtransistor 307-064) were interposed between the pulse generating circuit and the MC6840. In addition to the function of voltage matching, optoisolators also eliminated the influence of the noise.

The MC6840 was used to count the pulses from the encoders. The MC6840 was a programmable subsystem component of the MC6800 family designed to provide variable system time intervals.

The MC6840 has three 16-bit binary counters (timer 1-3), three corresponding control registers (CR 1-3), and a status register (SR). These counters are under software control and may be used to cause system interrupts or generate output signals. The MC6840 may be utilized for such tasks as frequency measurements, event counting, interval

measuring, and similar tasks. The device may be used for square wave generation, gated delay signals, single pulse of controlled duration, and pulse width modulation as well as system interrupts. Here it is used to measure the pulse width.

The interface of the MC6840 chip to MC68000 bus is shown in Fig. 3.35 and Table 3.8 gives the register addressing concerned.

Table 3.8

Register seleted address					Operation	
Address	A3 RS2	A2 RS1	A1 RS0	A0	R/W = 0	R/W = 1
$30050	0	0	0	0	CR20=0 Write CR3 CR20=1 Write CR1	No operation
$30052	0	0	1	0	Write CR2	Read SR
$30054	0	1	0	0	Write MSB buffer register	Read counter 1
$30056	0	1	1	0	Write timer 1 latches	Read LSB buffer counter
$30058	1	0	0	0	Write MSB buffer register	Read counter 1
$3005A	1	0	1	0	Write timer 2 latches	Read LSB buffer counter

Note: Because the MC6840 interfaces into the lower eight bits of the data bus, odd addresses must be used when the microprocessor accesses to the MC6840.

By properly programming the control register CR1 and CR2, timer 1 and timer 2 were defined into pulse width comparison mode. Table 3.9 explains the definition of the CR1 and CR2.

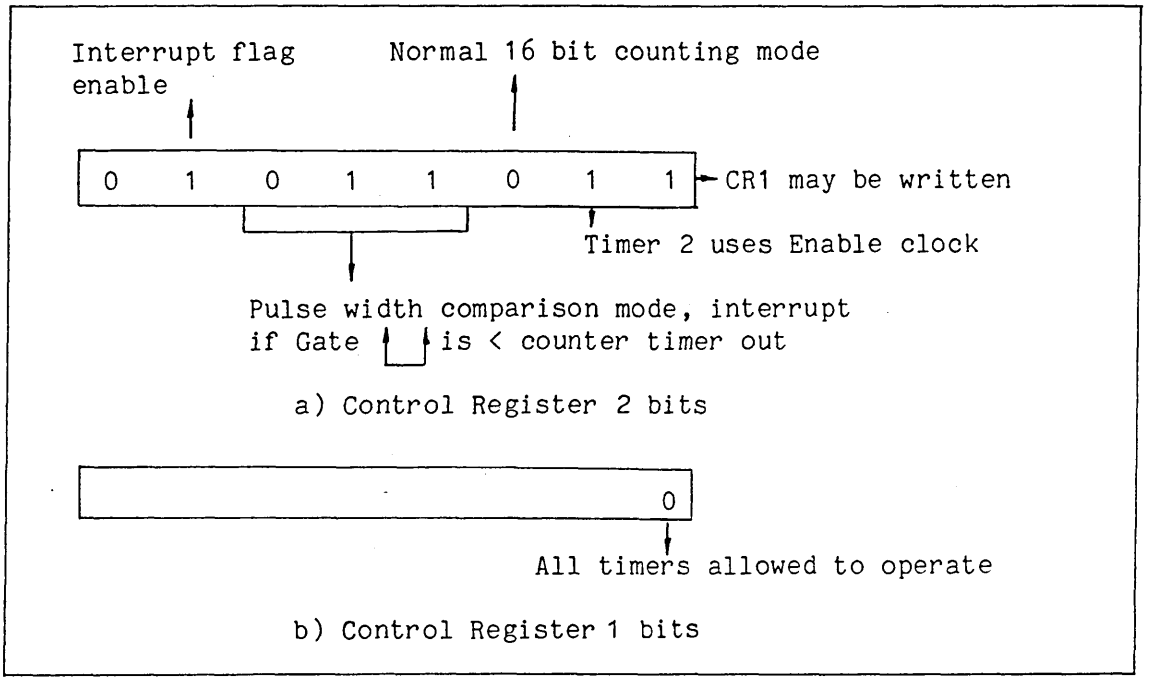


Table 3.9

In this case, timer 1 and timer 2 were programmed to measure the width of the pulses at the Gate inputs. After a write timer latch command was issued, a negative transition of the Gate input enabled the counter to decrement on each clock signal recognized (Enable clock $4*10^5$ Hz), and positive transition of Gate input would terminate the count. Thus, the pulse width could be calculated from the reading of the counter and the frequency of the Enable clock. The following example will reveal the programming method for the MC6840.

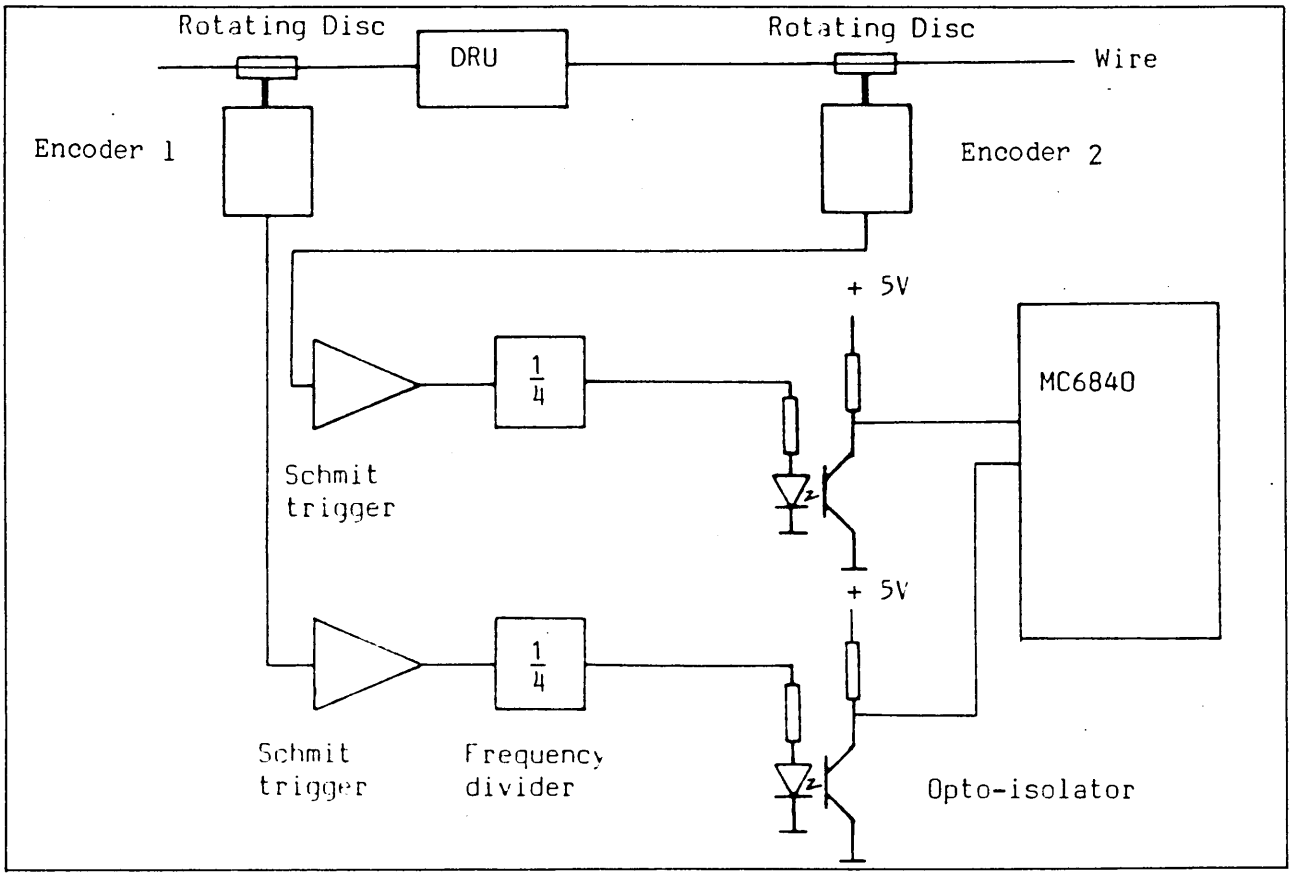
```
INI:  MOVE.B #$5B,   $30053   ;
      MOVE.B #$5A,   $30051   ; set CR1 and CR2 into pulse
                              ; width comparison mode
      MOVE.B #$FF,   $30055
      MOVE.B #$FF,   $30057   ; Counter 1 = $FFFF
      MOVE.B #$FF,   $30059   ;
      MOVE.B #$FF,   $3005B   ; Counter 2 = $FFFF
```

When negative transitions appeared at Gate 1 and Gate 2 inputs, counter 1 and counter 2 started to decrement on each Enable clock recognized, and positive transitions of Gate 1 and Gate 2 inputs would terminate the count, Fig. 3.34b shows the timing diagram. If the reading

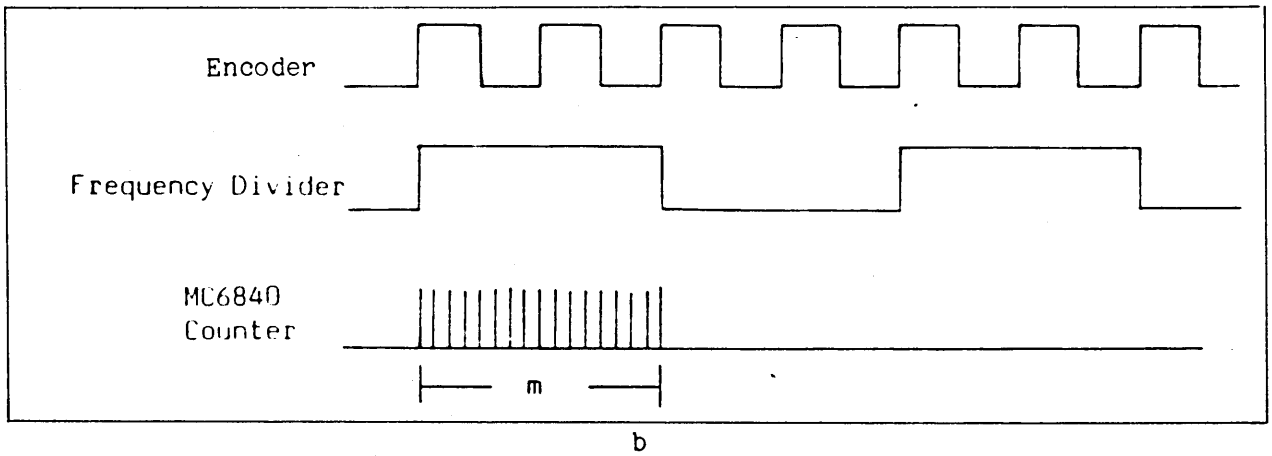
from counter 1 and counter 2 are $FBDF and $FC42 respectively, the pulse width will be

$$T1 = \$420 / 400000 = 1056 / 400000 = 2.64 \text{ ms}$$

and $$T2 = \$3BD / 400000 = 957 / 400000 = 2.39 \text{ ms}$$



a



b

Fig. 3.34 a) The Scheme for the Reduction in Area Transducer
b) The Timing Diagram

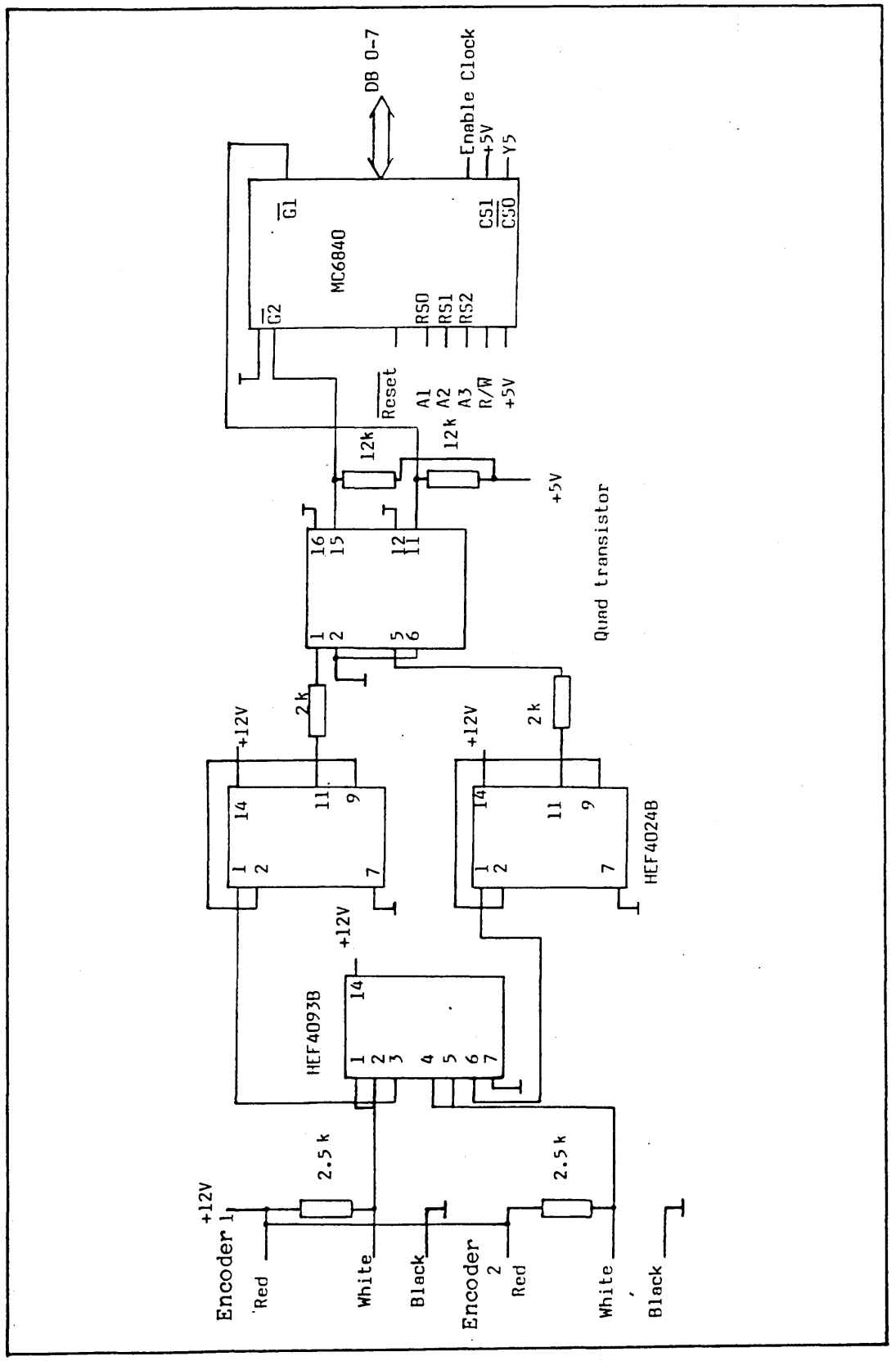


Fig. 3.35 The Diagram of the Interface Circuit

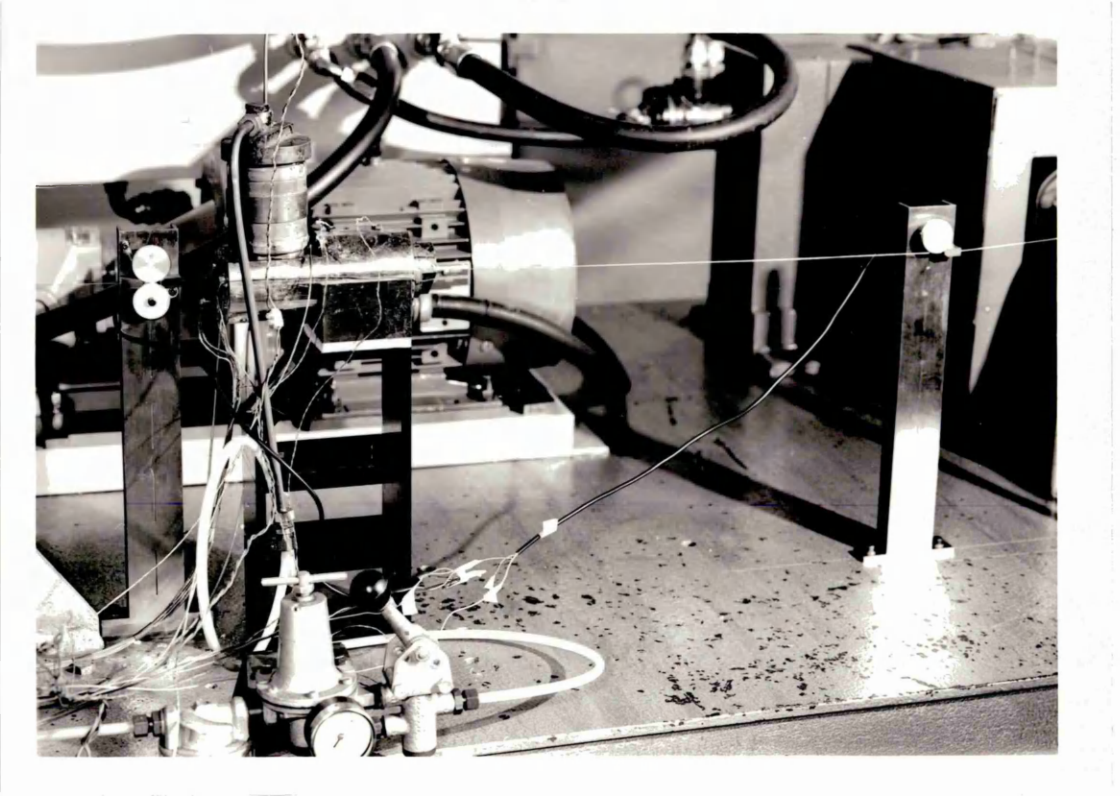


Plate 4

The T method [22] was used for pulse train processing, i.e., the detecting time was the width of the pulses received from the encoders. However, with this method the measuring accuracy decreases as the speed increases. At the highest drawing speed of 1.3 ms^{-1}, the frequency of the pulses reaches its maximum value

$$F_{max} = V_{max}*100 / L = 1300 \text{ Hz}$$

Where L is the circumference of the rotating wheel and L=100 mm. Noting that the frequency is divided by 4 and the width of the pulses is

measured, we have

$$CB = 2 F_c / F_{max} = 615$$

Where CB is the counting number during the duration of one pulse width, and F_c the frequency of the counting clock, F_c = 400 kHz. Then the maximum possible error in measurement is given by

$$\text{Error} = 1 / 615 = 0.16\%$$

Similarly at the lowest drawing speed of 0.1 ms^{-1}, the minimum frequency of the pulse is

$$F_{min} = V_{min} * 100 / L = 100 \text{ Hz}$$

Thus, the maximum detecting time is 20 ms.

3.7.2 - On Line Percentage Reduction in Area Indicator

In this case, the transducer interfaced with the MC68000 facilitated an on line indication of the percentage reduction in areas during the wire drawing process. The microcomputer calculated the actual PRA from the measurement of the transducer and showed the value of the PRA on the screen for every given period. A very simple program loop was used to work from sampling data to sending the PRA to the VDU. Fig. 3.36 shows the flowchart of the program and the listing of the program is given in Appendix 7.

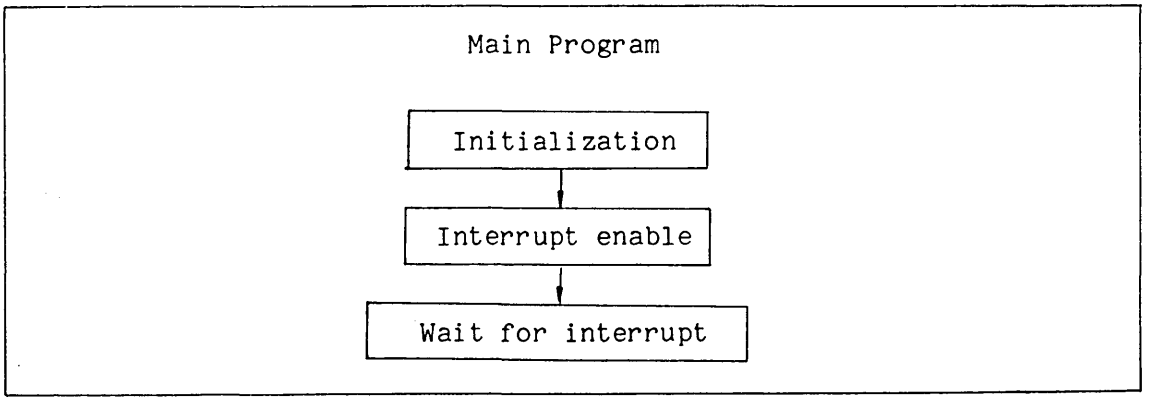


Fig. 3.36 Flowchart of the Program for the PRA Indicator

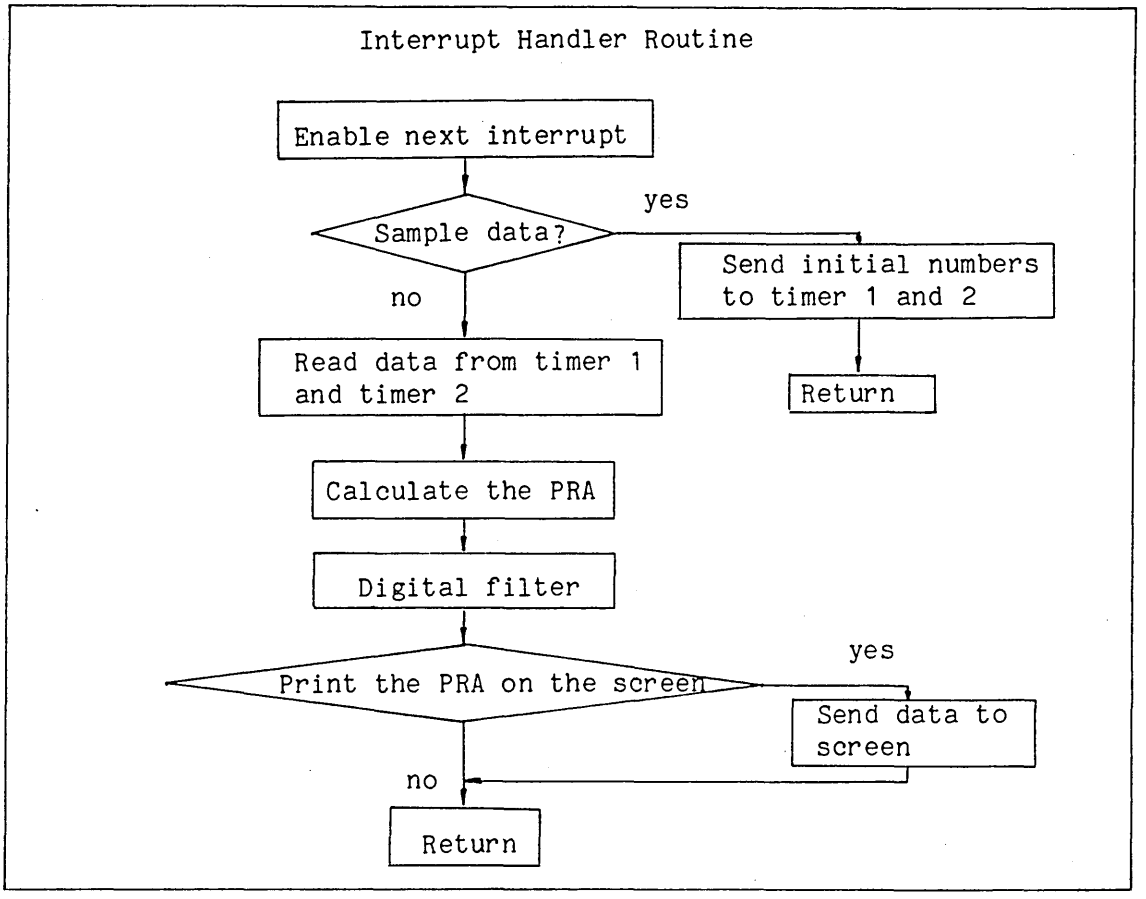


Fig. 3.36 Flowchart of the Program for the PRA Indicator

In determining the lowest drawing speed, a 40 ms sampling period was employed. An initial value of $FFFF was sent to each timer and thus the timer was enabled to start counting. If the counting values of the timer 1 and timer 2 were N_1 and N_2 respectively, then

$$PRA = (1 - N_2 / N_1)*100$$

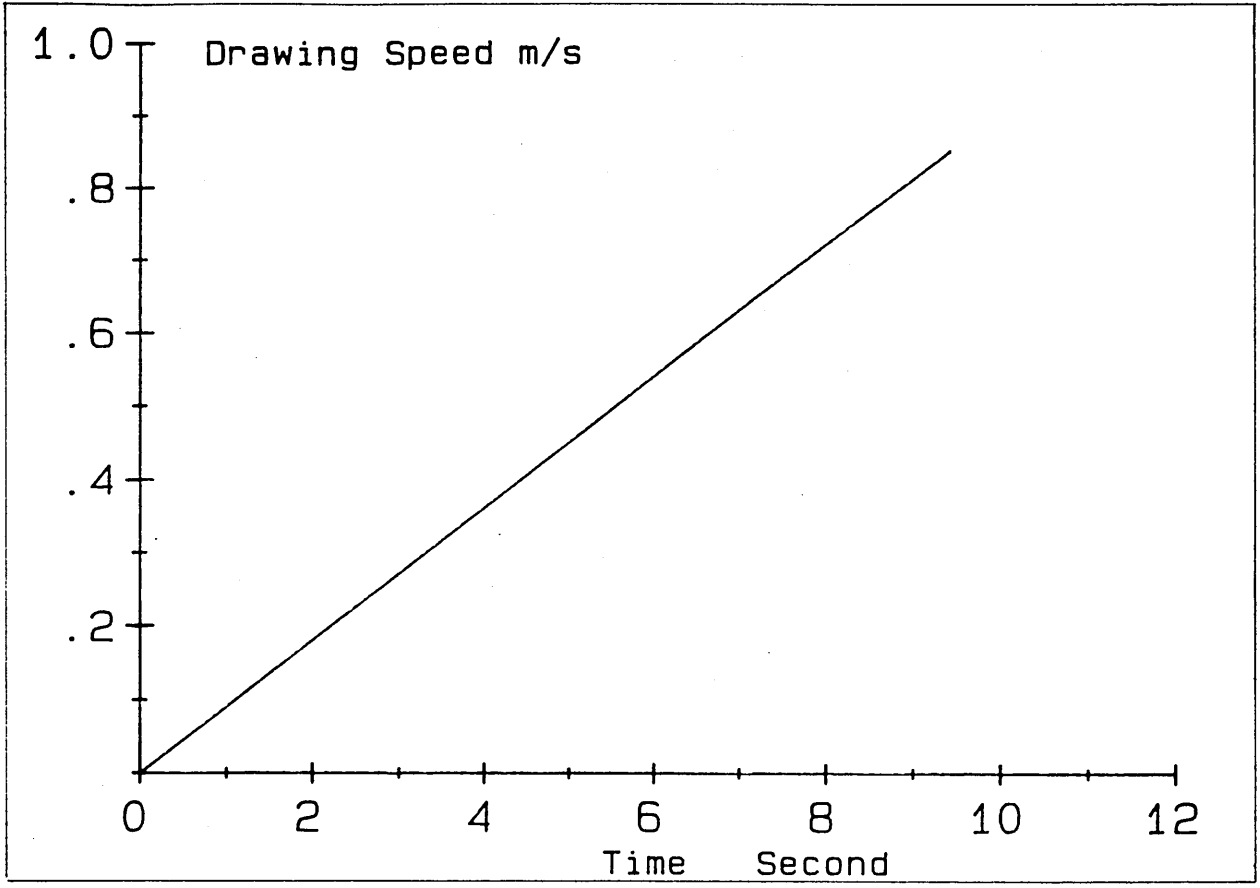
The arithmetic average of the last 10 samples was used as the algorithm of a digital filter. To show the value of the PRA in decimal notation on the screen, two TRAP 14 handler routines provided by the MC68000 resident firmware package were used. One was HEX2DEC which converted 8 bit hex number found in register D0 to the equivalent decimal number, the other, OUTICR, output a string to the VDU. The number of the PRA in decimal was of 3 characters, tens, units, and tenths, with the decimal point implied.

3.7.3 - Closed-Loop Control System

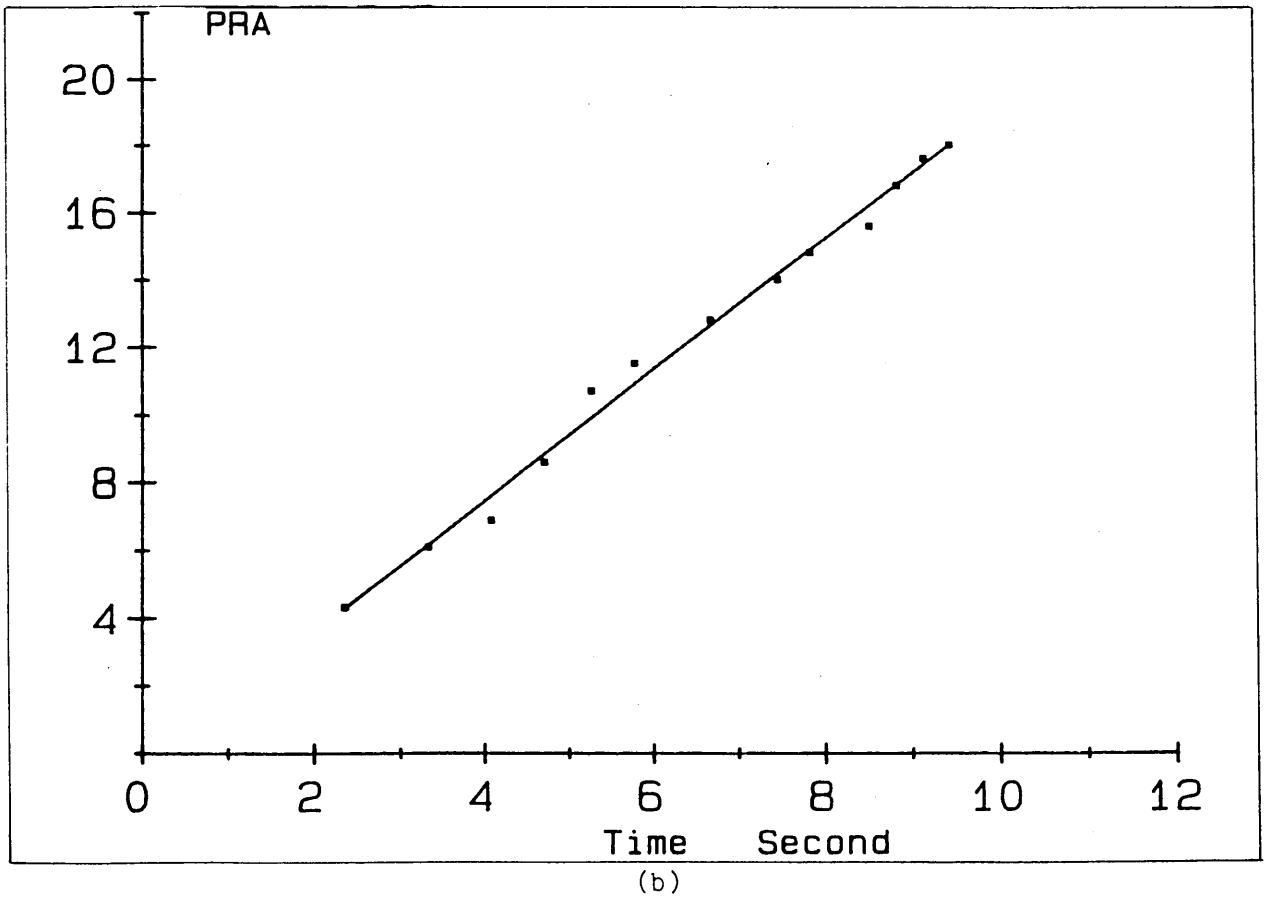
In this configuration, the microcomputer calculated the PRA from the measurement of the transducer, compared the calculated value with the PRA setpoint to find the error, and determined the manipulated signal to control the drawing speed to bring the PRA to its setpoint value.

The theoretical analysis in chapter 2 only provided a steady state model of the dieless wire drawing process. Figs. 3.37a and b show some experimental results of the dynamic characteristics of the dieless wire drawing process, in which the drawing speed increases linearly and the PRA also follows to increase linearly. Compared with the time constants of the drive elements, the dieless wire drawing process has a relatively small time constant, and thus the dieless wire drawing process may be assumed to be a proportional element. The transducer also is a proportional element and the hydraulic drive system is a second order system. Thus, the block diagram of the control system is as shown in Fig. 3.38.

In Fig. 3.38, the gain K of the process including the hydraulic drive system, the dieless wire drawing process, the transducer, and the interface circuit was set to 1. A proportional controller with gain $K_p=1$ was simply employed to implement the closed-loop control system. The steady error could be eliminated by intentionally off-setting the value of the setpoint. In this case, the setpoint was 2*PRA, where the PRA was the desired value. The algorithm for digital filtering was to calculate the arithmetic average of 5 last sampled data. Fig. 3.39 shows the flowchart of the program for drawing the wire of a uniform diameter and the program listing is given in Appendix 8.



(a)



(b)

Fig. 3.37 a) The Speed Input of the Dieless Wire Drawing Process
b) The PRA output of the Dieless Wire Drawing Process

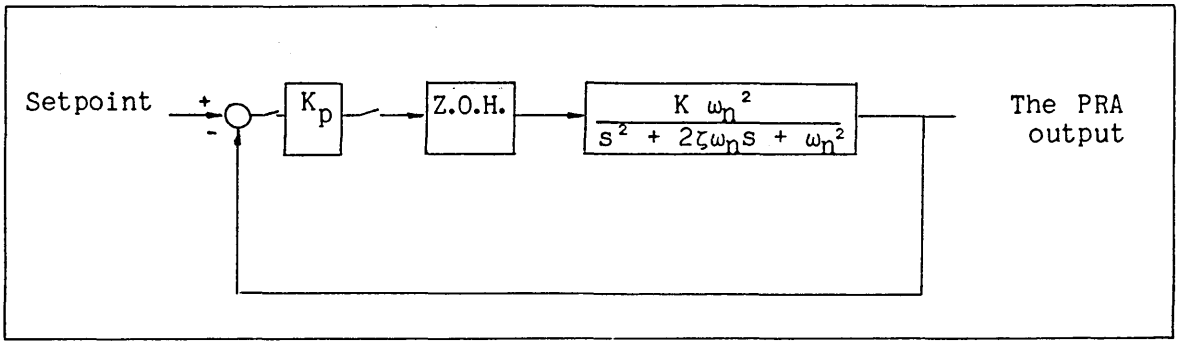


Fig. 3.38 Block Diagram of the Closed-Loop Speed Control System

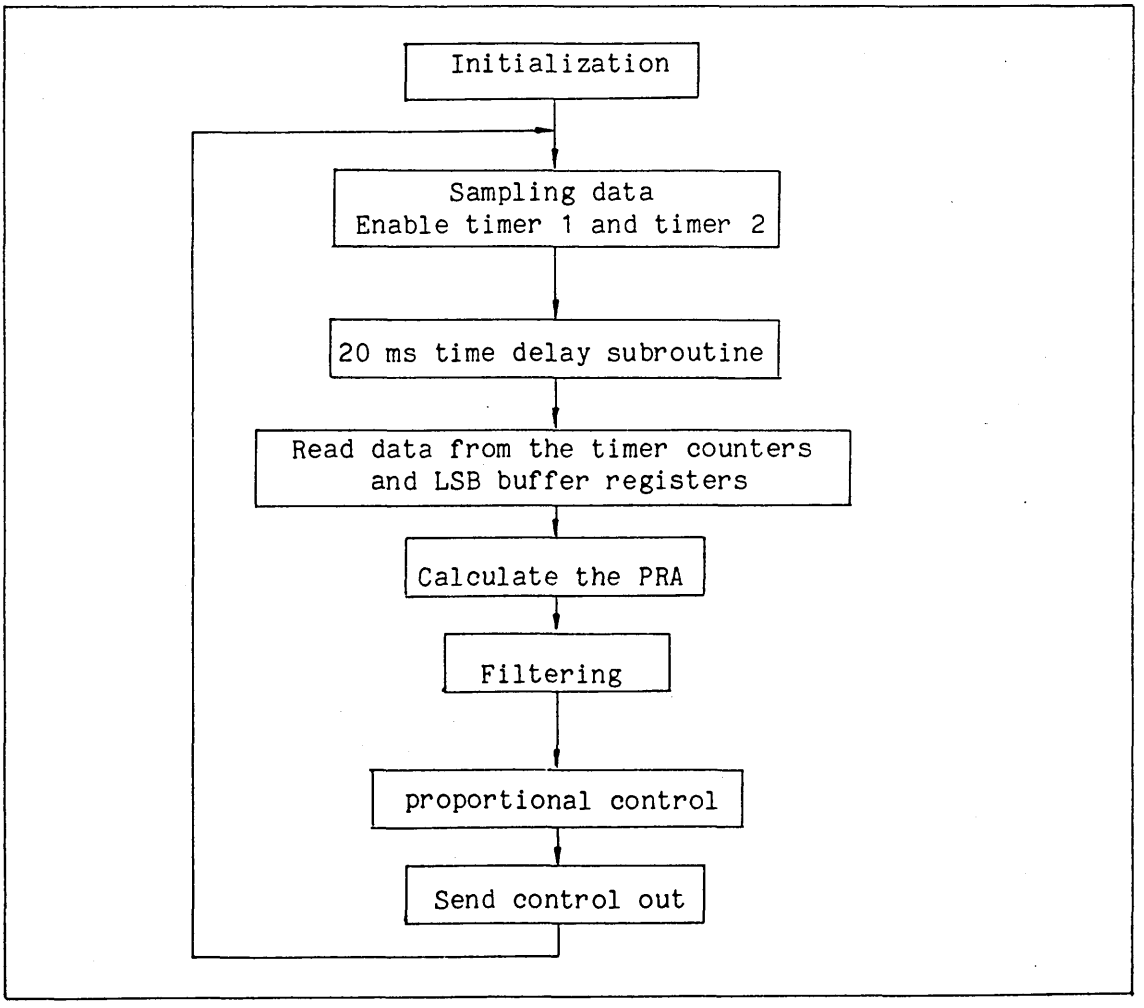


Fig. 3.39 The Flowchart of the Closed-loop Control System

3.8 - Results and Discussion

Test Procedure

The following procedure was followed to carry out a test. The power supplies for heater bands and other instrumentation were first switched on. The user program was loaded into the microcomputer from a tape and the program was executed from its starting address by an execution command. Following the prompt on the screen, the operator input the specification via the keyboard, and then the program execution was restarted. The temperature of the polymer melt in the polymer hopper and chamber were controlled by the microcomputer at the preset values. After 80 minutes for temperature to reach a uniform distribution state, wire from the coil was passed through the guides and over the pulley before being inserted and pulled through the PRA transducer and the DRU, then wound onto the bull block. The air pressure was put on, the hydraulic pump was started and the pressure valve was turned down to build hydraulic pressure up to 100 psi. Again the control program execution was restarted and the wire drawing process started.

Results

The hydraulic motor controlled by the microcomputer operated smoothly at its speed setpoint and the performance of the speed control system was shown by the speed step response in Fig. 3.30 (p. 103). The 0.4 percent quantization error could lead to a hunting problem at slow speed in which the system output varied between two discrete positions about the particular setpoint. This hunting problem was not of much practical importance since the motor usually operated at high speed for maximum PRA . Even in the tapered wire drawing process in which the drawing speed varied in a rather wide range and the hunting problem

occurred, the error caused by the hunting was still within the allowed range of ±2 percent. Fig. 3.40 shows the speed / time response of the hydraulic motor during a tapered wire drawing process, in which the speed range is set from 0.02 ms^{-1} to 1.14 ms^{-1} and the total drawing time is about 5 seconds.

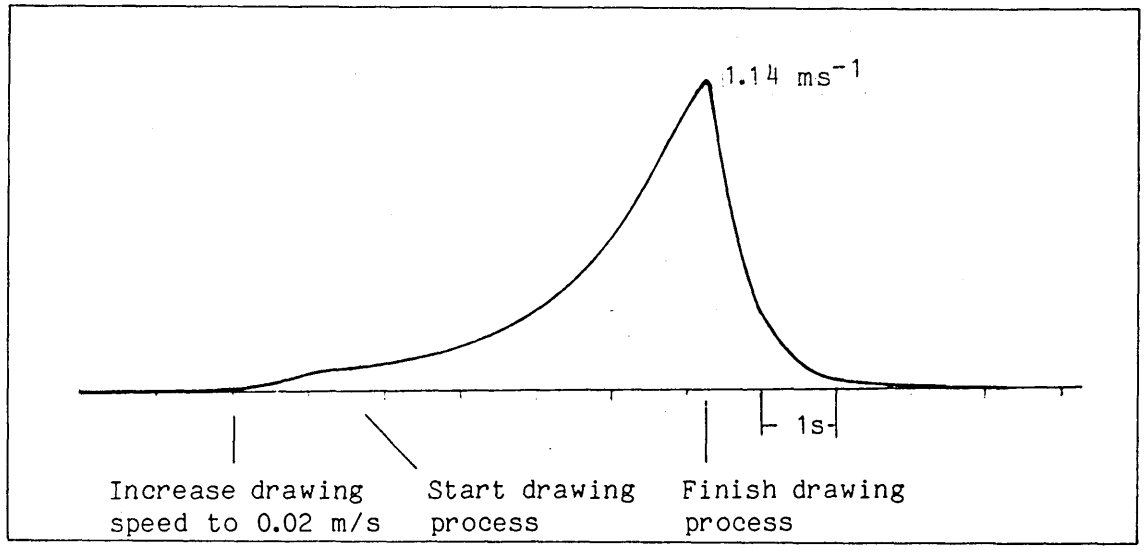


Fig. 3.40 Speed / Time Response During a Tapered Wire Drawing Process

The temperature of the polymer both in the polymer hopper and melt chamber was controlled thermostatically at its preset level to within ±3°C variations and the experimental results showed that with this accuracy, the requirement of producing products of the correct quality was met.

During the drawing process with a constant speed, the PRA indicator worked well over a wide measurement range from less than 1 percent to the maximum percentage reduction in area. In particular, at high drawing speed, the fluctuation of the reading of the indicator was very small. Plate 5 shows two pulse trains from encoders 1 and 2 respectively at a drawing speed of about 1 ms^{-1}, and a reduction in area of about 10

percent. The difference between the periods of the two pulse trains is clearly evident. However, when a rapid change occurred in drawing speed, the reading of the indicator fluctuated seriously and only after a relatively long period, did it become stable again. The most serious case of the measurement fluctuation is illustrated in Fig. 3.41. Here after a rapid change in the drawing speed as shown in Fig. 3.25 (p. 96), the readings of the indicator fluctuated over a wide range. In Fig. 3.41, all the readings above 11.6 percent are faulty and the actual measured values of the PRA during this process never showed larger values than 11.6 percent. The main problem causing this faulty reading stemmed from the mechanical driving parts. The encoder was driven by wire through a "friction" contact. During the process of a rapid change in drawing speed, because of the inertia of the encoder and the driving disc, slip between the wire and the disc might occur. During the process of a slow change in drawing speed, the indicator maintained the correct reading.

The fluctuation in readings during the process of a rapid change in drawing speed gave rise to difficulties in using this transducer to implement a closed loop control system. Some caution had to be exercised in order to reduce the oscillation during the transient part of the process. A digital filtering method was employed in sampled data processing. Even so, the final closed loop control system appeared not to work as well as expected, and its performance did not meet the requirement of drawing the tapered wire.

A qualitative test was conducted to examine the products using dieless wire drawing method controlled by microcomputer. The results of the test showed that the PRA kept close to the desired value with ±2 percent allowed errors over long length. Also, the fluctuation of the measured diameter of the products was very little, and close to that

measured in the original wire diameter, so the products had good uniformity in diameter. Figs. 3.42 to 3.45 show some results of the tapered wires produced, from the variations of the PRA versus the length of the produced wires, the uniformity of the change in the area of the tapered wires is demonstrated.

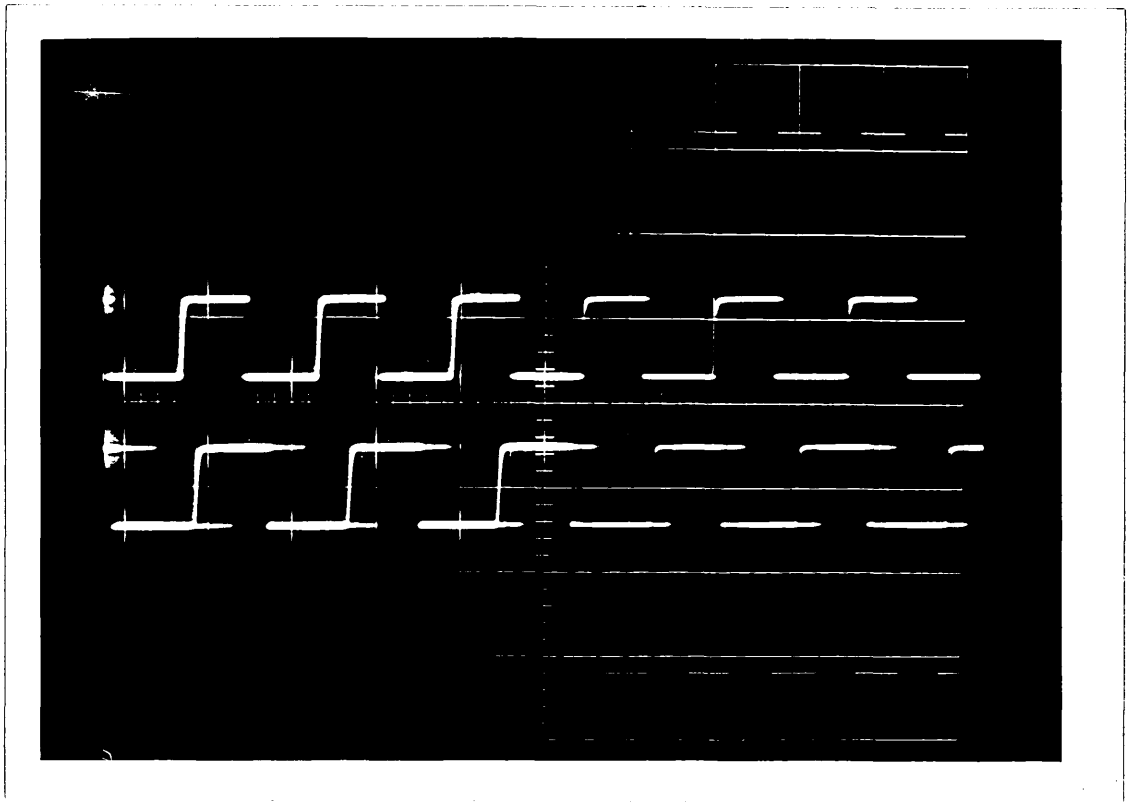


Plate 5 Pulse trains from Encoders

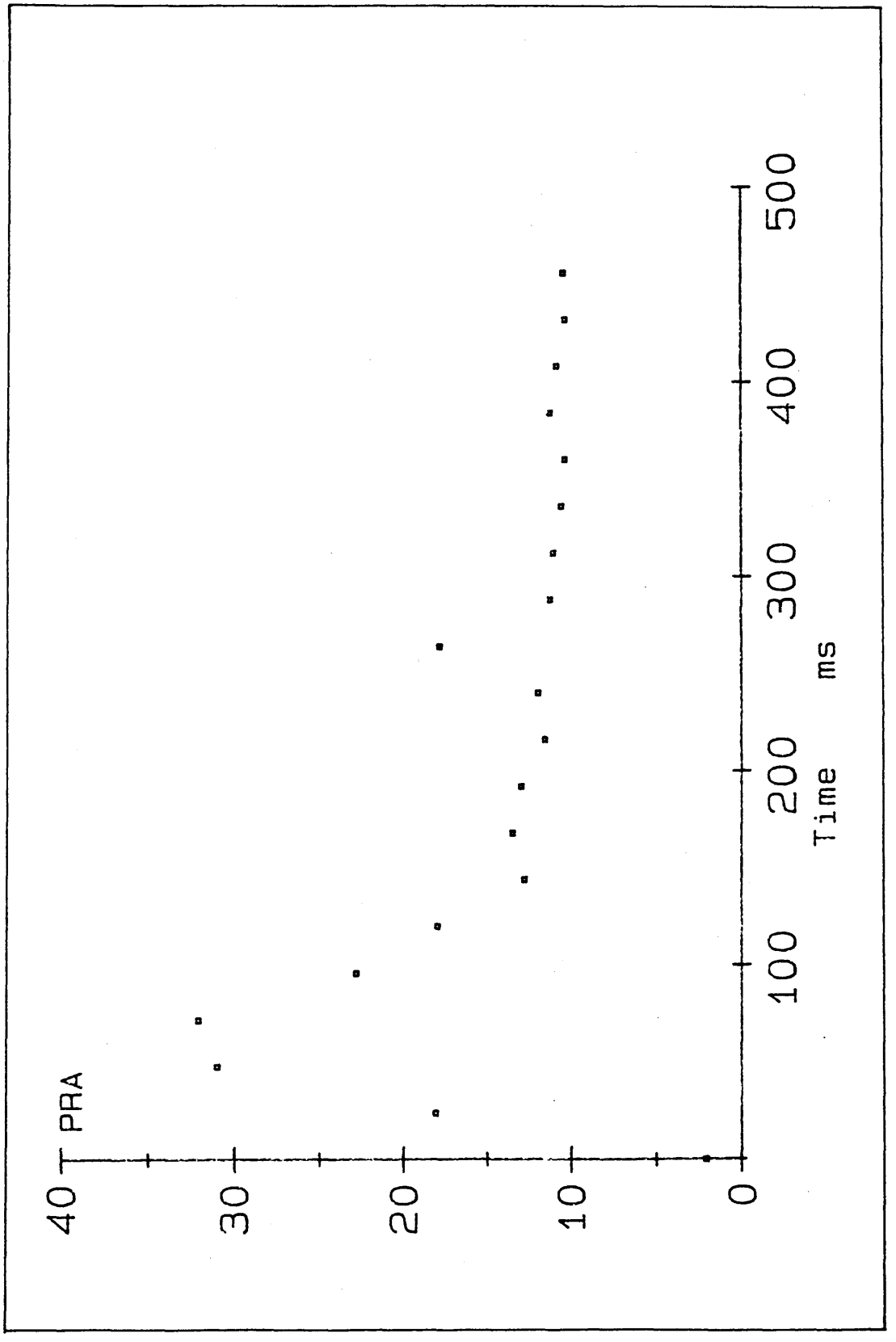


Fig. 3.41 Fluctuation in the PRA Readings After a Rapid Change in Drawing speed

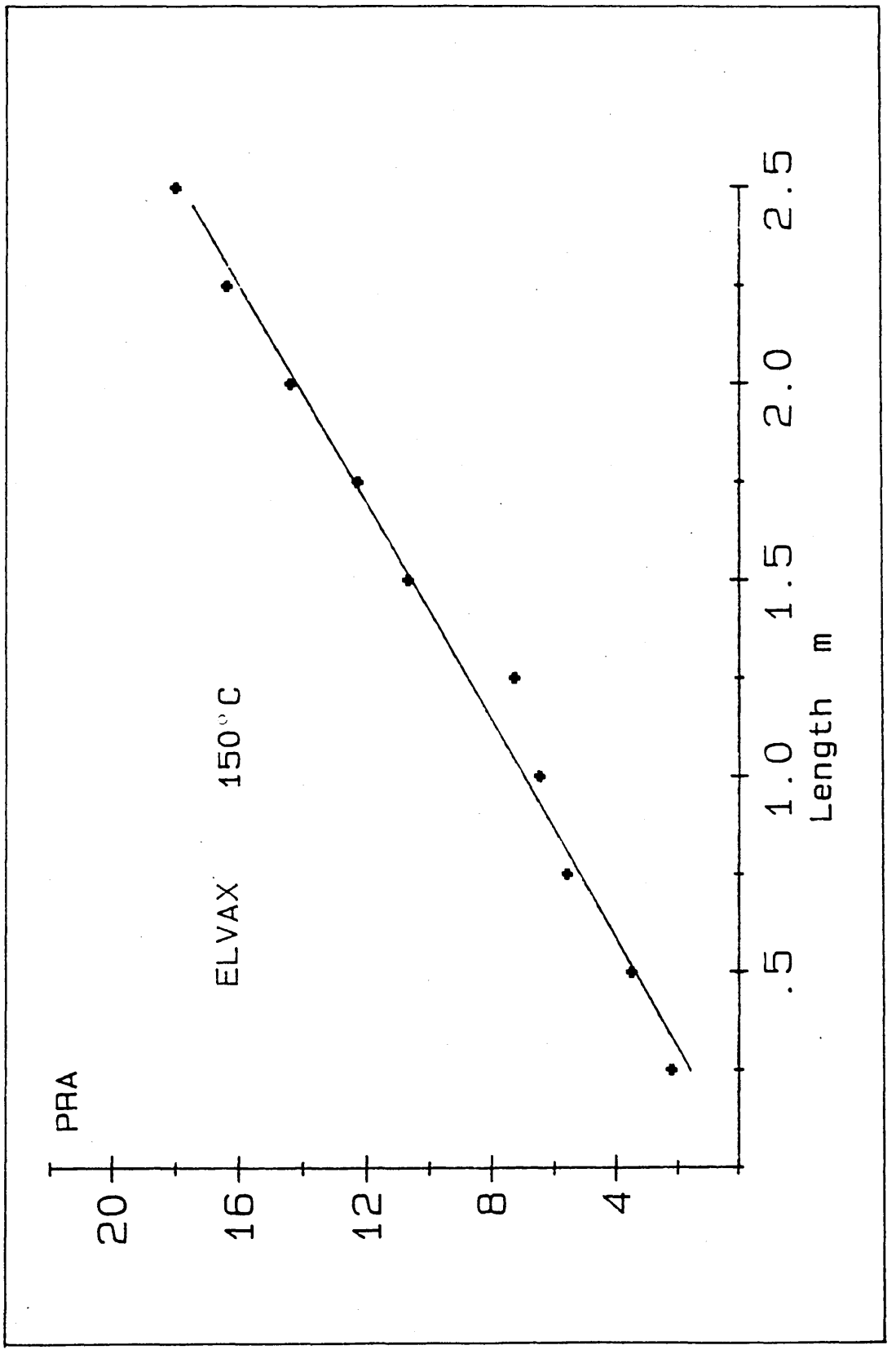


Fig. 3.42 PRA / Length of the Produced Tapered Wire

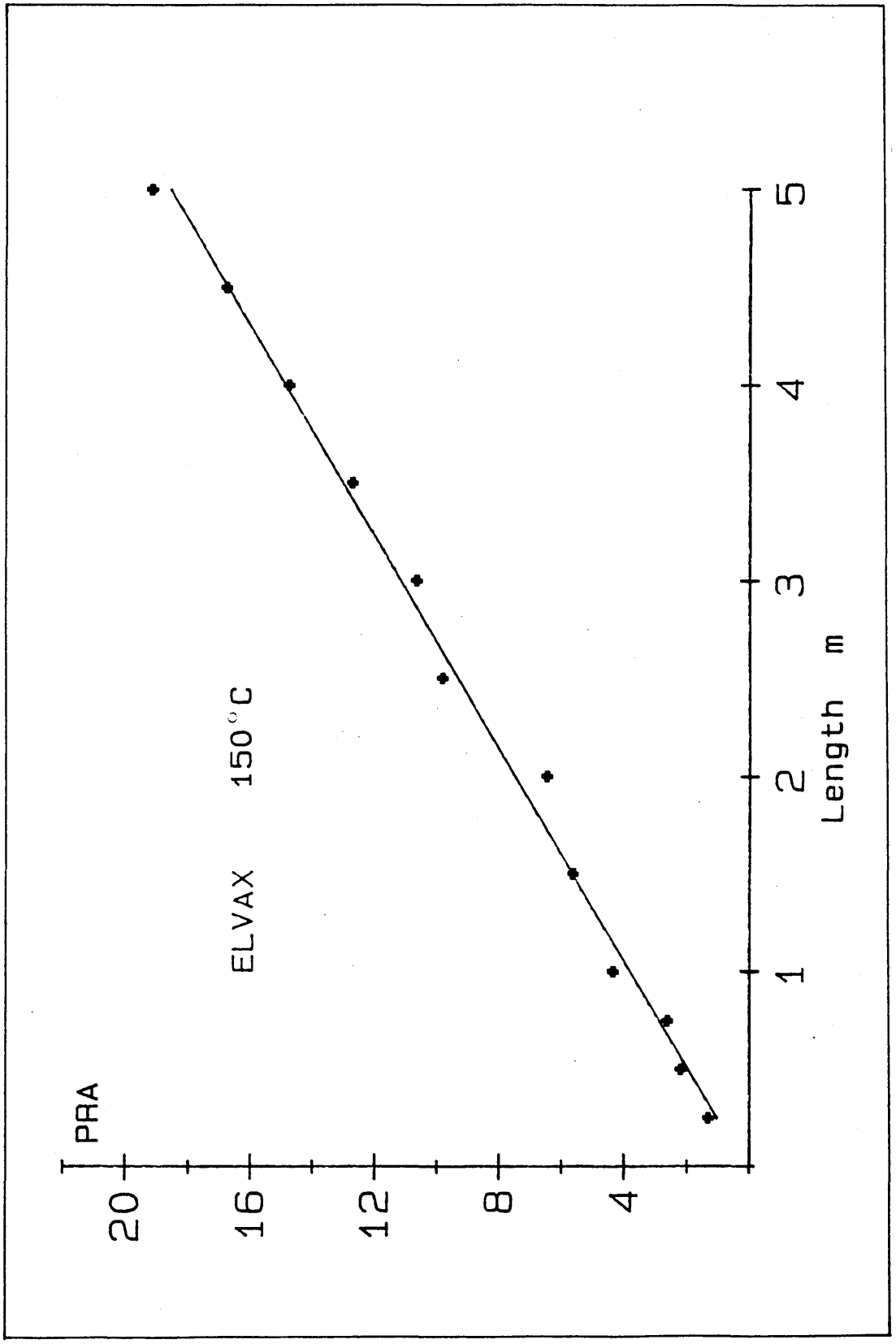


Fig. 3.43 PRA / Length of the Produced Tapered Wire

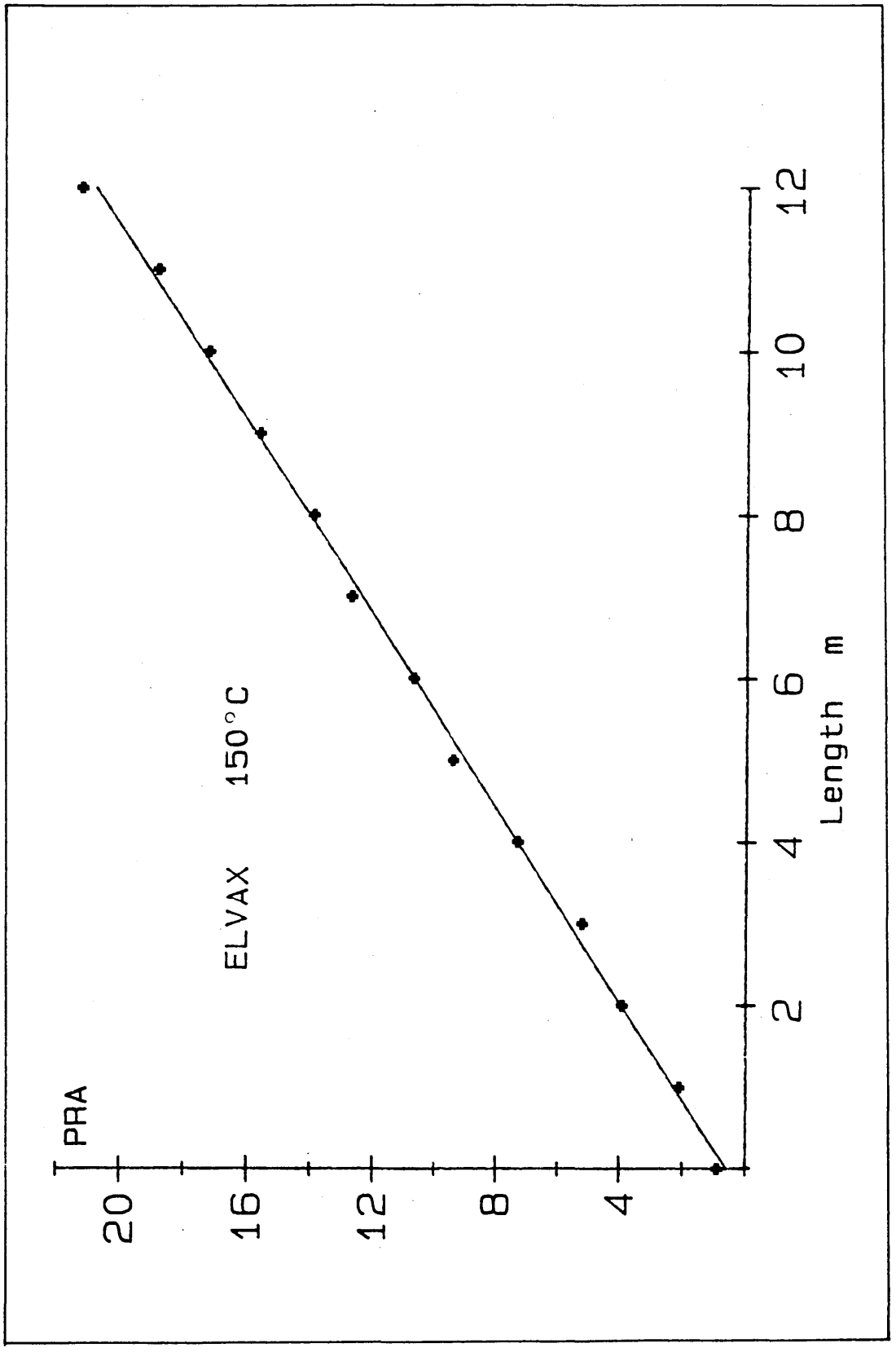


Fig. 3.44 PRA / Length of the Produced Tapered Wire

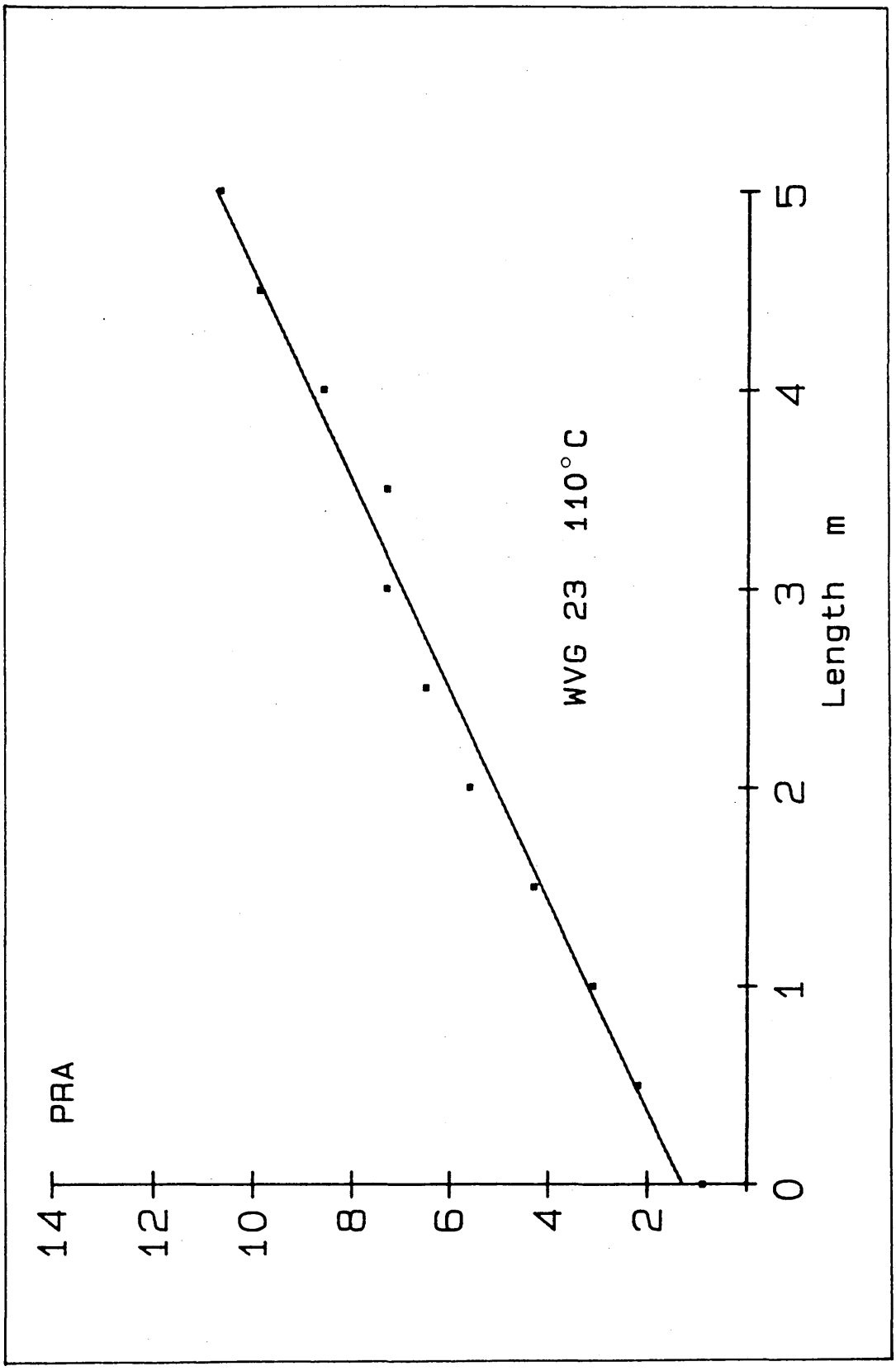


Fig. 3.45 PRA / Length of the Produced Tapered Wire

Discussion

The design of the speed and the temperature control systems was primarily based on the requirement of ±2 percent allowed error in the area of the produced wire. The tachogenerator introduced the main error to the control system and the 8-bit DAC and ADC with 0.4 percent resolution also added some error into the system. For high quality of products, the performance of the system can be easily improved by using digital speed measurement approach or 10 bit ADC and DAC. However, the complexity of the system will be increased and only for specially high quality of products would this improvement need to be considered.

Tapered wires with different size could be easily produced by this dieless wire drawing method. To date no other method using conventional wire drawing for the production of tapered wires has been reported.

The PRA indicator worked well under steady-state conditions, but fluctuation in readings occurred under transient conditions. To enable a closed-loop control system with good performance to be built using the PRA as a feedback signal, the mechanical components of the transducer require some redesign.

CHAPTER 4 : Design of the Digital PID Controller

K_P Proportional gain

T_I Integral time constant

T_D Derivative time constant

g_i Coefficient of Z^{-1} in $G(Z^{-1})$

T Sampling period

h_i Coefficient of Z^{-1} in $H(Z^{-1})$

a_i Coefficient of Z^{-1} in $A(Z^{-1})$

b_i Coefficient of Z^{-1} in $B(Z^{-1})$

c_i Coefficient of Z^{-1} in $C(Z^{-1})$

t_i Coefficient of Z^{-1} in $T(Z^{-1})$

P_i Coefficient of Z^{-1} in $P(Z^{-1})$

f_i Coefficient of Z^{-1} in $F(Z^{-1})$

u(t) The control increment

U(t) Sequence of controller output variables

e(t) Error

$A(Z^{-1})$, $B(Z^{-1})$, $C(Z^{-1})$ System polynomial in Z^{-1}

Y(t) Sequence of system output variables

$\zeta(t)$ White noise sequence

$\phi(t)$ Pseudo system output

$P(Z^{-1})$, $Q(Z^{-1})$, $G(Z^{-1})$, $P_n(Z^{-1})$, $P_d(Z^{-1})$, $H(Z^{-1})$, $F(Z^{-1})$ polynomials in Z^{-1}

W(t) Reference value of system output

$\phi_y(t)$ Filtered system output

$T(Z^{-1})$ Desired closed loop characteritic polynomial

$\varepsilon(t)$ Estimation error

Z(t) Vector of delayed input / output variable

$\theta(t)$ Vector of controller polynomial coefficients

$P(t)$	Covariance matrix
$\overline{P(t)}$	Finite memory version of P(t)
ρ	Forgetting factor
I	Unit matrix
ω_n	Natural frequency of second order system
ζ	Damping factor of second order system
K	System time delay (integer)
V	Control weighting
ρ_o	Coefficients down the diagonal of the initial matrix P(0)
T	Sampling period

One of the most widely used controllers in the design of continuous-data control systems is the PID controller, where PID stands for proportional + integral + derivative control. Fig. 4.1 shows the block diagram of a typical continuous-data PID controller acting on an error signal e(t). The transfer function of the controller is given by equation (4.1)

$$U(s) = (K_p + \frac{K_I}{s} + K_D s)\, E(s) \qquad (4.1)$$

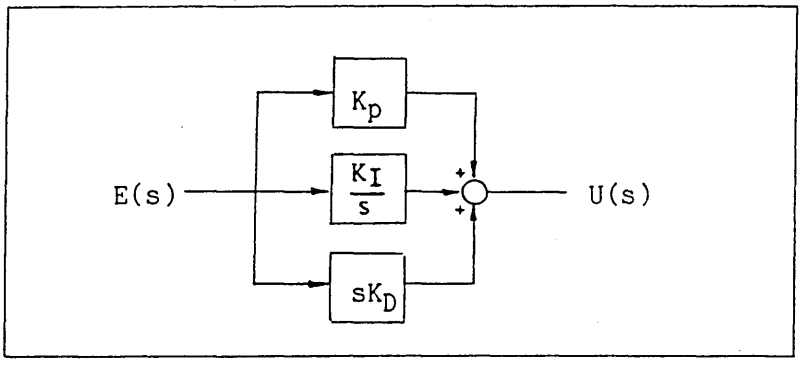


Fig. 4.1 A Continuous-Data PID Controller

There are several PID controllers with different structures up to the point where the only remaining common feature is the integral action over the error. We will consider here the basic structure[23], given by equation (4.2)

$$U(s) = K_p(1 + \frac{1}{T_I s} + \frac{T_D s}{1 + \frac{T_D s}{N}})\, E(s) \qquad (4.2)$$

where K_p , T_I , T_D are proportional gain, integral time, and derivative time respectively, N is a constant >>1.

The same principle of the PID control can be applied to digital control and there is a number of ways to implement a PID digital

controller. For example, taking Z-transforms on both sides of equation (4.2), we have the discrete-data transfer function of the PID digital controller, i.e.

$$U(t) = \frac{g_0 + g_1 Z^{-1} + g_2 Z^{-2}}{(1 - Z^{-1})(1 + h_1 Z^{-1})} E(t) \qquad (4.3)$$

where $g_0 = (N + 1)K_p$

$$g_1 = -Kp(1 + e^{\frac{-TN}{T_D}} + 2N - \frac{T}{T_I})$$

$$g_2 = Kp(N + e^{\frac{-TN}{T_D}} - \frac{T}{T_I} e^{\frac{-TN}{T_D}})$$

$$h_1 = - e^{\frac{-TN}{T_D}}$$

and T is the sampling period

After performing cross multiplication, the velocity form of PID controller is given by

$$u(t)(1 + h_1 Z^{-1}) = (g_0 + g_1 Z^{-1} + g_2 Z^{-2}) E(t) \qquad (4.4)$$

or $$u(t) = \frac{G(Z^{-1})}{H(Z^{-1})} E(t)$$

where $u(t) = (1 - Z^{-1})U(t)$

$$G(Z^{-1}) = g_0 + g_1 Z^{-1} + g_2 Z^{-2}$$

and $H(Z^{-1}) = 1 + h_1 Z^{-1}$

The PID controller can be easily implemented by a program in a digital computer and a number of alternatives to the PID controller described by equation (4.4) are available, [24] [25].

4.2 - Self-Tuning Controllers

In recent years, a great deal of attention has been focussed on adaptive control. Self-tuning controllers represent an important class of adaptive controllers; they are easy to implement and applicable to processes with a wide variety of characteristics: unknown parameters,

the presence of time delay, time-varying process dynamics and stochastic disturbances. The aim of the self-tuning algorithm is to carry out the synthesis, implementation and validation of a digital controller in an on-line manner. In its essential form, a self-tuning controller combines the sequence of identification, controller synthesis and implementation as shown in Fig. 4.2. The sequence is carried out at each sampling interval and proceeds in an iterative manner until the controller coefficients achieve steady values, at which point the identification and synthesis stages are stopped.

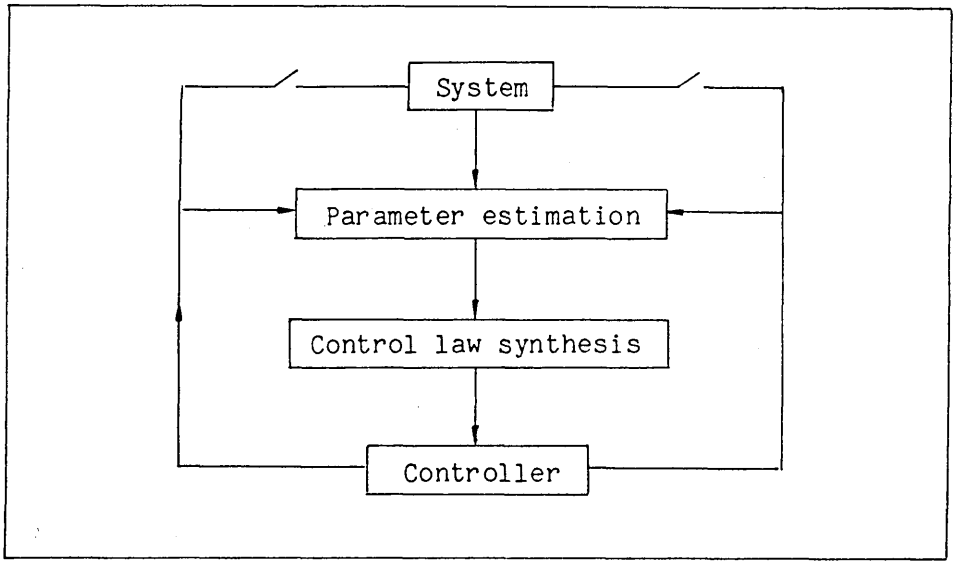


Fig. 4.2 The Self-Tuning Sequence

The first major development within this class of controllers was the self-tuning regulator (STR) designed to maintain a constant setpoint in the face of unmeasured disturbances [26] [27]. The technique is very sensitive to the presence of non-minimum phase characteristics. Also the control signals generated may be excessively large, such that the system actuators will saturate. A later development in the form of a self-tuning controller (STC), utilizes a more general control objective than the STR, by minimizing the variance of a generalized output (i.e. a combination of the output, input and setpoint). This approach allows

the STC to be applied to a wider variety of processes; for example, it can easily handle non-minimum phase systems, and setpoint tracking is incorporated directly in the algorithm. However, the technique requires a known time delay, so it is not applicable to a system which has an unknown time delay or one in which the time delay varies, unless such variation can be specified analytically, [28] [29].

The approach taken by Wellstead [30] [31], was to modify the STR, using pole assignment as the design criterion. The method is not sensitive to the presence of non-minimum phase characteristics; it can be detuned to avoid the excessive control action characteristic of minimum-variance controllers, and unknown or varying time delays can easily be handled. However, the number of polynomials which need to be chosen is more than for the minimum variance algorithm.

The weighted minimum-variance self-tuning controller [32] is based upon a k-step-ahead control law and can be used for unstable and non-minimum-phase plants. The controller includes control weighting and this results in relatively smooth control signals. The controller is also stable for ranges of control weighting where the STC is unstable.

The generalized STC with pole assignment [33] combines the suboptimal approach of the STC and the classical approach of pole assignment to yield robustness and ease of setpoint tracking.

All these schemes were developed for stochastic systems. The pole/zero assignment technique was also used to design a self-tuner for a class of deterministic systems [34]. Recently, it has been shown that the basic STC and the generalized STC with pole assignment can be used correctly for both deterministic and stochastic systems [35]. The algorithm of the generalized STC with pole assignment for deterministic systems avoids the polynomial factorization or a non-linear identification procedure as in [34], merely requiring the solution of a

polynomial identity.

Because of the desirable practical features of self-tuning control, its application to engineering almost came together with its theory. Cegrell and Hedquist, Borisson and Wittenmark have applied it to a problem within the paper industry as a regulator. The work was reported to be successful [36] [37]. The application by Borisson and Syding concerned the self-tuning control of an ore-crusher [38]. The target was to regulate the power absorbed by the crusher via control of ore feed-rate. Other such chemical process steady-state regulation problems [39], domestic heating system control [40] et al were also reported. The practical results have demonstrated the effectiveness of the self-tuners.

4.3 - Self-Tuning PID Controllers

It is clear that a self-tuning controller, in general, need not necessarily have a PID structure, and with the availability of cheap and fast computing capabilities, there is no justification to limit designs to simple PID controllers. Furthermore, powerful identification techniques are readily available to provide more accurate high order models than the second order approximation. Several papers have discussed these points [23] [41] [42]. The conclusions reached are in summary as follows:

(1) In the process industries, conventional PID control still predominates because it is robust and remarkably effective for a wide range of processes. It is also well-understood by many plant engineers.

(2) When faced with linear time-invariant plants, the PID controller design technique allows one to simply conduct the tracking capability-noise rejection compromise, given that low frequency

modelling uncertainty is overcome by the integrator's high gain. This allows the system to follow a step command with no steady-state error, even in the presence of a disturbance of unknown magnitude, as long as the closed-loop system remains stable. In a practical servo mechanism problem where the operating point moves as the output tries to follow the reference trajectory, the validity of the linearization of the process nonlinear differential equation is reduced. The integral term effectively cancels the most offending part of such an approximation error, the bias term.

(3) It is often difficult to establish a priority of economic benefits for proposed applications of modern control. Furthermore, the control system hardware cost is usually only a small portion of the total project cost.

However, the designer is faced with the requirement to detune the regulator parameters, thus reducing the performance, in order to cope with variations of gain and time constants at different operating points. In some industrial processes, fixed parameter controllers become inadequate because of more stringent performance requirements or the existence of large modelling uncertainties, or time varying plant.

Some controllers need regular retuning to account for process changes and ageing. Also there are some disadvantages associated with conventional PID parameter tuning techniques [42]. In PID process control, two tuning methods are commonly used: The first is the Ziegler-Nichols method and the second is the method by Chien. The former needs the ultimate gain and the period of ultimate oscillation at the stability limit. However it is often difficult to determine their exact values experimentally in real processes because oscillatory operation may have to be avoided. The latter method requires an exact form of the

process expressed by a transfer function. However many real processes fail to reveal their transfer functions.

Providing the PID controller with the capability of self-tuning (that is, adjusting its parameters from on-line measurements and the deviation from the desired index of performance - the tracking error) may be the possible solution to overcome these disadvantages . It is for the above reasons that the self-tuning PID controller described in this work was developed.

A self-tuning PID controller can be used in two different ways [24] [43]:

(1) To provide a procedure to automatically generate PID tuning constants instead of the usual trial-and-error procedure; after the PID settings have converged, the self-tuner can be turned off.

(2) To provide an adaptive control system in which the PID controller parameters are modified to adapt to a changing situation.

Several reported self-tuning PID controllers are summarized in the following sections.

4.3.1 - Self-Tuning PID Controllers Based on STC [24]

This is a modified version of the STC with PID structure. The process model is given by

$$A(Z^{-1})Y(t) = Z^{-K}B(Z^{-1})U(t) + C(Z^{-1})\zeta(t) \qquad (4.5)$$

where Y(t) is the measured output

U(t) the manipulated input

$\zeta(t)$ an uncorrelated sequence of random variables

and t denotes the sampling instant, t = 0,1,2,.. The polynomials $A(Z^{-1})$, $B(Z^{-1})$ and $C(Z^{-1})$ are expressed in terms of the backward shift operator Z^{-1}

$$A(Z^{-1}) = 1 + a_1Z^{-1} + ,,,, + a_nZ^{-n}$$

$$B(Z^{-1}) = b_0 + b_1Z^{-1} + ,,,, + b_mZ^{-m} \qquad b_0 \neq 0$$

$$C(Z^{-1}) = 1 + C_1Z^{-1} + ,,,, + C_nZ^{-n}$$

The self-tuning controller is designed to minimize the variance of an auxiliary output $\phi(t)$

$$\phi(t) = P(Z^{-1})Y(t) + Q(Z^{-1})U(t - K) - R(Z^{-1})W(t - K) \tag{4.6}$$

where W(t) is the set point and P, Q and R are transfer functions of the forms

$$P(Z^{-1}) = \frac{P_n(Z^{-1})}{P_d(Z^{-1})}, \text{ etc} \tag{4.7}$$

If the process model in equation (4.5) is combined with the identity

$$CP_n = AP_dE + Z^{-K}F \tag{4.8}$$

where the degree of the polynomial F is given by

$$\deg F = n + \deg P_d - 1 \tag{4.9}$$

and $$\deg E = K - 1 \tag{4.10}$$

The predictive model is obtained

$$\phi(t) = \frac{F}{P_dC}Y(t - K) + (\frac{EB}{C} + Q)U(t - K) - RW(t - K) + E\zeta(t) \tag{4.11}$$

A control law that minimizes the variance of $\phi(t)$ is given by

$$U(t) = \frac{CRW(t) - FY(t)/P_d}{EB + CQ} \tag{4.12}$$

An alternative predictive model is given by

$$\phi_y(t) = PY(t) \tag{4.13}$$

From equation (4.5) and equation (4.7), it follows that

$$\phi_y(t) = \frac{F}{P_dC}Y(t - K) + \frac{EB}{C}U(t - K) + E\zeta(t) \tag{4.14}$$

We set

$$G=EB \tag{4.15}$$

and define $$Y_f(t) = \frac{Y(t)}{P_d} \tag{4.16}$$

If we assume that C = 1, then linear least squares estimation can be used and the estimation model is given by

$$\phi_y(t) = FY_f(t - K) + GU(t - K) + \varepsilon(t) \tag{4.17}$$

Estimates of the elements of F and G can be obtained from the standard least squares equations. Thus the control law is given by

$$U(t) = \frac{RW(t) - F Y_f(t)}{G + Q} \tag{4.18}$$

In order that the controller (4.18) may have a PID structure, the F polynomial is specified to have degree 2, i.e.

$$F = f_o + f_1 Z^{-1} + f_2 Z^{-2} \tag{4.19}$$

and the output filter is deg $P_d = 1$, i.e., $P_d = 1 + P_{d1} Z^{-1}$. Then from equation (4.9), we have n = 2, i.e. a second order process model.

As a final design step, R is set by

$$R = H_o = \frac{f_0 + f_1 + f_2}{1 + P_{d_1}} \tag{4.20}$$

and since Q may be freely chosen, integral action is introduced by setting

$$G + Q = \frac{1 - Z^{-1}}{V} \tag{4.21}$$

where V is a design parameter. Substituting equations (4.19), (4.20) and (4.21) into equation (4.18) yields controller

$$u(t) = V \{ H_o W(t) - (f_o + f_1 Z^{-1} + f_2 Z^{-2}) Y_f(t) \}$$

where $u(t) = (1-Z^{-1})U(t) = U(t) - U(t-1)$

It is an ideal digital PID controller in velocity form [44].

4.3.2 - Self-Tuning PID Controllers Based on Zero/Pole Assignment [23]

The process model to be controlled is given by

$$Y(t) = Z^{-1}\frac{B(Z^{-1})}{A(Z^{-1})}U(t) + \frac{C(Z^{-1})}{A(Z^{-1})}\zeta(t) \qquad (4.22)$$

where $A(Z^{-1}) = 1 + a_1Z^{-1} + a_2Z^{-2}$

$B(Z^{-1}) = b_0 + b_1Z^{-1}$; $b_0 \neq 0$

Time delay is not considered here.

Place the system in closed loop with an incremental controller given by equation (4.23), as shown in Fig. 4.3

$$u(t) = \frac{G(Z^{-1})}{H(Z^{-1})}e(t) \qquad (4.23)$$

where $G(Z^{-1}) = g_0 + g_1Z^{-1} + g_2Z^{-2}$

and $H(Z^{-1}) = 1 + h_1Z^{-1}$

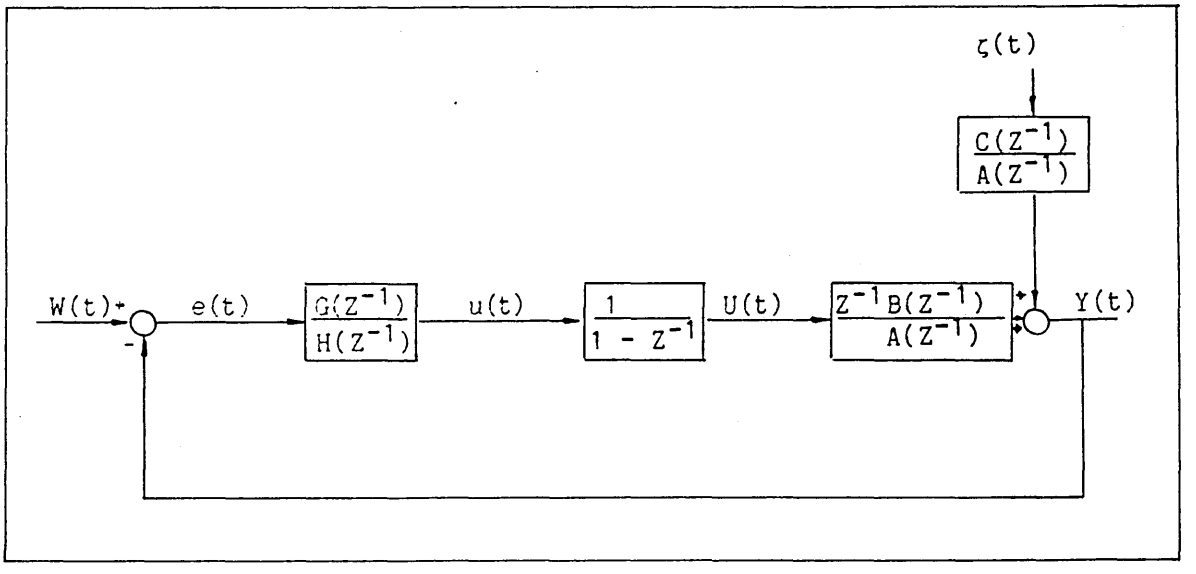


Fig. 4.3 Composite Servo/Regulator Configuration

Notice that a digital integrator is incorporated to ensure steady state correspondence between Y(t) and W(t) (stochastic regulator), or Y(t) tracking of changes in W(t) (servo-control). The closed-loop equation is given by

$$\{(1-Z^{-1})A(Z^{-1})H(Z^{-1})+Z^{-1}\,B(Z^{-1})G(Z^{-1})\}Y(t)$$

$$= Z^{-1}\,B(Z^{-1})G(Z^{-1})W(t)+C(Z^{-1})H(Z^{-1})\zeta(t) \qquad (4.24)$$

If the stochastic disturbance is negligible, then the servo tracking control law is found by solving the identity [45]

$$(1 - Z^{-1})A(Z^{-1})H(Z^{-1}) + Z^{-1}B(Z^{-1})G(Z^{-1}) = T(Z^{-1}) \qquad (4.25)$$

for the unknown controller coefficients, where $T(Z^{-1})$ is a polynomial which defines the desired pole set. The regulator coefficients (stochastic regulator) are obtained by solving the identity

$$(1 - Z^{-1})A(Z^{-1})H(Z^{-1}) + Z^{-1}B(Z^{-1})G(Z^{-1}) = C(Z^{-1})T(Z^{-1}) \qquad (4.26)$$

The self-tuning algorithm at each sample interval can be summarised as follows

1) Estimate the coefficients of the system model given by

$$Y(t) = -(a_1Z^{-1} + a_2Z^{-2})Y(t) + Z^{-1}(b_0 + b_1Z^{-1})U(t) + \varepsilon(t) \qquad (4.27)$$

2) Use the estimated data to synthesise the controller coefficients via the identities of equations (4.25) or (4.26).

3) Determine incremental u(t) and apply the control U(t) which is obtained by integrating u(t) via a digital integrator.

Other different procedures of autotuning PID controllers also have been reported [23] [46] [47] [48]. In [46], a continuous-time self-tuning PID controller was proposed by Gawthrop. The integral action in this method arises naturally from a suitable model of disturbance rather than forcing integral action into the controller as in the previous two methods and a hybrid approach with a continuous-time model but discrete-time control law and estimation is used in the development of this method. Some commercial products with built-in PID self-tuners are available in the market [49] [50]. There is of no doubt that the provision of a procedure to automatically generate PID tuning constants instead of the usual trial-and-error method is very important in engineering practice.

4.4 - Design of Self-Tuning PID Controllers With Pole Assignment for Deterministic Systems

As mentioned before, self-tuning algorithms have been developed separately for the stochastic and deterministic cases. It has been shown that the basic self-tuning algorithm of the STC can be used correctly for deterministic systems; and pole assignment also can be incorporated as in [33] to provide pole assignment controllers focusing entirely on the servo or reference following problem.

The algorithm presented here shows that by inserting an integrator in the forward path, the self-tuning controller proposed in [35] has the same structure as a conventional three term PID controller (for second order systems) or PI controller (for first order systems). The major use of the resulting self-tuning PID controller for deterministic systems would be as an initial tuning device to determine the PID controller parameters needed to provide the specified control performance. Because deterministic system identification is only feasible under disturbance conditions, the controller set point will be suitably disturbed until convergence is achieved and the controller parameters then fixed until retuning is considered necessary. The tuning disturbance must be kept within the bounds acceptable to the plant.

In 4.4.1, the algorithm for deterministic systems is summarized, and in 4.4.2, it is shown how this leads to a self-tuning PID controller. The results of digital simulation and experiments are given in 4.4.3 and 4.4.4 respectively.

4.4.1 - Self-Tuning controllers for Deterministic Systems [35]

The system to be controlled is assumed to be described by

$$A(Z^{-1})Y(t) = Z^{-K}B(Z^{-1})U(t) \qquad (4.28)$$

A controller of the following form is required

$$HU(t)+GY(t)-RW(t) = 0 \tag{4.29}$$

where H, G and R are polynomials in Z^{-1}.

Introducing polynomials P and Q, and defining

$$PB+AQ = TR \qquad \text{for } Q \neq 0 \tag{4.30}$$

or

$$P = TR \qquad \text{for } Q = 0 \tag{4.31}$$

Where T is the polynomial representing the desired closed-loop system poles, T and R are prespecified. Also, let polynomials F, G and H be defined from

$$P = AF + Z^{-K}G \tag{4.32}$$

$$H = BF+Q \tag{4.33}$$

Combining equation (4.28) with equation (4.29) gives the closed-loop description

$$Y(t) = Z^{-K} \frac{BR}{AH + Z^{-K} BG} W(t) \tag{4.34}$$

Making use of equations (4.32) and (4.33), we have

$$AH + Z^{-K}BG = BP + AQ \tag{4.35}$$

Then, using equations (4.30) and (4.31), equation (4.34) may be rewritten as

$$Y(t) = \frac{Z^{-K} BR}{PB + AQ} W(t)$$

$$= \frac{Z^{-K}B}{T} W(t) \qquad \text{for } Q \neq 0$$

or

$$Y(t) = \frac{Z^{-K}}{T} W(t) \qquad \text{for } Q = 0 \tag{4.36}$$

It gives a closed-loop system with the desired closed-loop transfer function.

Multiplying equation (4.35) by Y(t) and substituting for AY(t) from equation (4.28), equation (4.35) becomes

$$PY(t) + Z^{-K}QU(t) = H Z^{-K}U(t) + Z^{-K}GY(t) \tag{4.37}$$

Equation (4.37) can be written as

$$\phi(t) = HZ^{-K}U(t) + Z^{-K}GY(t) \tag{4.38}$$

and

$$\phi(t) = PY(t) + Z^{-K}QU(t) \tag{4.39}$$

Equation (4.38) and equation (4.39) can be used to estimate the parameters of the polynomials H and G, and thus from equation (4.29) control U(t) can be determined.

4.4.2 - Self-Tuning PID Controllers With Pole Assignment for Deterministic Systems

The system is considered to be described by the equation (4.40)

$$Y(t) = Z^{-K}\frac{B}{A}U(t) + d \tag{4.40}$$

where U(t) and Y(t) are the system input and output respectively. A and B are polynomials in Z^{-1}, with $a_0 = 1$, d is a constant bias or off-set, and K represents the system time delay in sample instants.

If now an integrator is cascaded with the system, as in Fig. 4.4, the newly formed system can be expressed as

$$Y(t) = Z^{-K}\frac{B}{A'}u(t) + d \tag{4.41}$$

where

$$u(t) = (1-Z^{-1})U(t) \tag{4.42}$$

and

$$A' = (1-Z^{-1})A$$

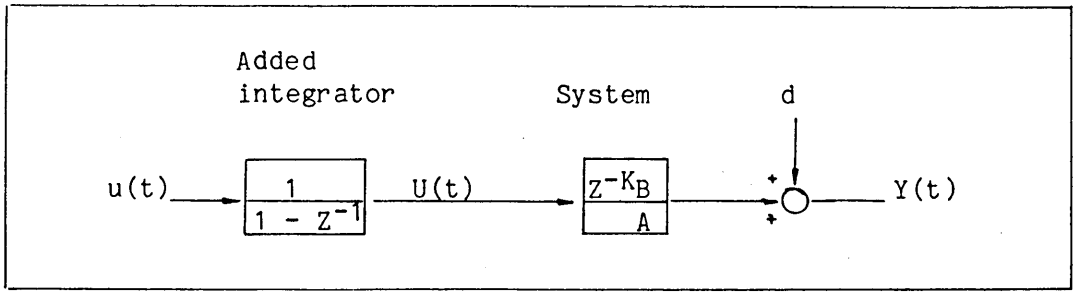


Fig. 4.4 Integrator Cascaded With System

In terms of the generalized minimum variance self-tuning strategy, a performance cost function $\phi(t)$ for the system can be defined as

$$\phi(t) = PY(t) + Qu(t-k) - RW(t-k) \tag{4.43}$$

Define polynomials F, G, E and H

$$P = FA' + Z^{-K}G \tag{4.44}$$

$$E = -R \tag{4.45}$$

$$H = BF + Q \tag{4.46}$$

Equations (4.41) to (4.46) may be combined to give

$$\phi(t) = Hu(t-k) + GY(t-k) + EW(t-k) + \delta \tag{4.47}$$

where $$\delta = (A'F)\Big|_{Z=1}^{d} \tag{4.48}$$

Thus the control law is given by

$$Hu(t) + GY(t) + EW(t) + \delta = 0 \tag{4.49}$$

Note that, since A' contains a factor $(1-Z^{-1})$, $\delta = 0$ from equation (4.48), and hence δ can be deleted from the estimation algorithm. Rewriting equations (4.43), (4.47) and (4.49) gives

$$\phi(t) = PY(t) + Qu(t-k) - RW(t-k) \tag{4.50}$$

$$\phi(t) = Hu(t-k) + GY(t-k) + EW(t-k) \tag{4.51}$$

$$Hu(t)+GY(t)+EW(t)=0 \tag{4.52}$$

Equations (4.50) and (4.51) may be written as

$$\phi(t) = PY(t) + Qu(t-k) \tag{4.53}$$

$$\phi(t) = Hu(t-k) + GY(t-k) \tag{4.54}$$

Equations (4.50), (4.51), or equations (4.53), (4.54) and equation (4.52) define a self-tuning control algorithm with an integrator operating on the system of equation (4.40).

A closed-loop description of the system may be obtained by substituting for u(t) from equation (4.52) into equation (4.41)

$$Y(t) = \frac{Z^{-K}\,BR}{A'H + Z^{-K}\,BG}\,W(t) + \frac{A'H}{A'H + Z^{-K}\,BG}\,d \tag{4.55}$$

Since d is a constant bias and $A' = (1-Z^{-1})A$, the second term in equation (4.55) vanishes and can therefore be eliminated. The added integrator therefore ensures that the bias does not influence the controller output.

Let P and Q be chosen such that:

$$BP + A'Q = T \tag{4.56}$$

where T is a prespecified polynomial in Z^{-1} whose roots are the desired values of the poles of the closed-loop transfer function. Substituting for P and H from equations (4.44) and (4.46) into equation (4.56) gives

$$BP + A'Q = A'H + Z^{-K}BG = T \tag{4.57}$$

Thus, the closed-loop transfer function is given by

$$Y(t) = \frac{Z^{-K}\,BR}{T}\,W(t) \tag{4.58}$$

For equations (4.44) and (4.57) to have a solution, the degrees of the polynomials P, Q, T, F, G and thus H (equation (4.46)) must be

$$\begin{aligned} &\deg P = \deg A' - 1 \\ &\deg Q = \deg B - 1 \\ &\deg T \leq \deg A' + \deg B - 1 \\ &\deg F = k-1 \\ &\deg G = \deg A' - 1 \\ &\deg H = \deg B + K - 1 \end{aligned} \tag{4.59}$$

When self-tuning, the system parameters A and B are unknown and hence equation (4.57) can not be solved directly to obtain the parameters of polynomials P and Q. However, multiplying equation (4.56) by F gives

$$FPB + QA'F = TF \tag{4.60}$$

Substituting for A'F from equation (4.44) results in

$$FPB + QP - Z^{-K}\,GQ = TF \tag{4.61}$$

Further use of equation (4.46) gives

$$PH - Z^{-K} GQ = TF \tag{4.62}$$

In self-tuning control, the parameters H and G are identified recursively from equations (4.50) and (4.51) or equations (4.53) and (4.54). Then equation (4.62) may be solved to get the values for P and Q that will be used in the next step. Obviously for the first step, starting values for P and Q need to be assigned.

From equation (4.44), $p_0 = a_0' f_0$, but $a_0' = 1$, hence $p_0 = f_0$. p_0 may be set equal to unity, hence, $p_0 = f_0 = 1$. However, from equation (4.62) it can be seen that because h_0 is estimated freely from equation (4.51), the relationship $h_0 = t_0$ will not be satisfied. This problem can be avoided by modifying equation (4.62) as

$$PH - Z^{-K} GQ = h_0 FT \tag{4.63}$$

Similarly equation (4.56) becomes

$$BP + A'Q = h_0 T \tag{4.64}$$

Now for the second order system, $A(Z^{-1}) = 1 + a_1 Z^{-1} + a_2 Z^{-2}$, and $B(Z^{-1}) = b_0 + b_1 Z^{-1}$, from equation (4.59), deg G = 2, deg H = 1, i.e.

$$G(Z^{-1}) = g_0 + g_1 Z^{-1} + g_2 Z^{-2} \tag{4.65}$$

and

$$H(Z^{-1}) = h_0 + h_1 Z^{-1} \tag{4.66}$$

From equation (4.52), the controller u(t) is given by

$$u(t) = \frac{RW(t) - (g_0 + g_1 Z^{-1} + g_2 Z^{-2}) Y(t)}{h_0 + h_1 Z^{-1}} \tag{4.67}$$

where $R = g_0 + g_1 + g_2$

Equation (4.67) gives the velocity form of the PID controller.

For the first order system, $A = 1 + a_1 Z^{-1}$, $B = b_0$, thus

$$G(Z^{-1}) = g_0 + g_1 Z^{-1} \tag{4.68}$$

$$H = h_0 \tag{4.69}$$

The controller is given by

$$u(t) = \frac{RW(t) - (g_0 + g_1 Z^{-1}) Y(t)}{h_0} \tag{4.70}$$

It is the velocity form of PI controller.

In the foregoing discussion of synthesis procedures, a recursive estimation is needed to estimate the parameters of the controller, i.e. H and G. For a second order system with K = 1, equation (4.54) can be written in vector matrix format as

$$\phi(t) = Z^T(t)\theta + \varepsilon(t) \tag{4.71}$$

where $$Z^T(t) = [\, u(t-1), u(t-2), y(t-1), y(t-2), y(t-3) \,] \tag{4.72}$$

$$\theta^T(t) = [\, h_0, h_1, g_0, g_1, g_2 \,] \tag{4.73}$$

and $\varepsilon(t)$ is the estimation error.

The recursive least square estimation of the vector of the controller coefficient θ at time interval t is given by [45]

$$\theta(t) = \theta(t-1) - P(t)Z(t)[Z^T(t)\theta(t-1) - \phi(t)] \tag{4.74}$$

$$P(t) = \overline{P(t-1)} - \overline{P(t-1)}Z(t)Z^T(t)\overline{P(t-1)}[1 + Z^T(t)\overline{P(t-1)}Z(t)] \tag{4.75}$$

where $$\overline{P(t-1)} = P(t-1)/\rho \tag{4.76}$$

ρ is an exponential forgetting factor and is used to make the controller follow time varying parameters. Its value is not less than 0.98 and not greater than 1.

At the time step t = 1, the initial matrices P(0) and $\theta(0)$ must be specified. $\theta(0)$ can be set with guesses of coefficients, P(0) can be set that $P(0) = \rho_0 I$, where ρ_0 is a large number, I is the unit matrix.

The algorithm is summarised as follows

1) Form $\phi(t) = PY(t) + Qu(t-k)$

   Starting values for P and Q need to be assigned

2) Estimate H and G from $\phi(t) = Hu(t-k) + GY(t-k)$

3) Apply control $Hu(t) + GY(t) - RW(t) = 0$

4) Solve for P and Q from $PH - Z^{-K}GQ = h_0FT$

5) Repeat from 1) for the incremented value of t

The desired closed-loop pole-set is best specified in terms of either a first order or a second order polynomial in Z^{-1} [45]. For a first order response

$$T(Z^{-1}) = 1 + Z^{-1}t \tag{4.77}$$

where $t = -\exp(-T/\tau)$

T is the sampling period

τ is the desired closed-loop time constant, and typically $\tau = 3T$

For a second order system

$$T(Z^{-1}) = 1 + Z^{-1}t_1 + Z^{-2}t_2$$

$$t_1 = -2\exp(-\zeta\omega_n T)\,\mathrm{Cos}(\omega_n\sqrt{1-\zeta^2}\,T)$$

$$t_2 = \exp(-2\zeta\omega_n T) \tag{4.78}$$

where ω_n is the undamped natural frequency

ζ is the damping ratio of the corresponding continuous time second order characteristic polynomial, and typically $\omega_n\sqrt{1-\zeta^2}\,T = 1/3$.

4.4.3 - Simulation and Results

A demonstration of how the self-tuning PID controller for deterministic systems works will be given in terms of simulated examples. The flowchart of the digital simulation program and the computer listing are given in Appendix 9.

Example 1

A second order system which is stable and minimum phase is considered having the system description

$$Y(t) = Z^{-1}\,\frac{1.57 + 1.36Z^{-1}}{1 - 1.628Z^{-1} + 0.657Z^{-2}}\,U(t) + d$$

Corresponding to the continuous-time system

$$G(s) = \frac{100}{(10s+1)(25s+1)}$$

with sampling period T = 3 second.

The aim is to have the closed-loop poles defined by $T(Z^{-1}) = 1 - 1.47Z^{-1} + 0.604Z^{-2}$, corresponding to the continuous time second order characteristic polynomial with $\zeta = 0.6$, and $\omega_n = 0.14$.

Since polynomials A, B and T are known equations (4.44), (4.46) and (4.64) can be solved directly to find the required values of P, Q, H and G to which the self-tuning algorithm should converge, thus we have

$$h_0 = \frac{b_0 a_2 + \frac{b_1^2}{b_0} - a_1 b_1 - \frac{a_3 b_0^2}{b_1}}{\frac{b_1 t_1}{b_0} + a_2 - t_2 - \frac{a_1 b_1}{b_0} - \frac{a_3 b_0}{b_1}}$$

$$q_0 = h_0 - b_0$$

$$p_2 = -q_0 \, a_3/b_1$$

$$p_1 = (h_0 t_1 + a_1 b_0 - a_1 h_0 - b_1)/b_0$$

$$h_1 = b_1$$

$$g_0 = p_1 - a_1$$

$$g_1 = p_2 - a_2$$

$$g_2 = -a_3 \qquad (4.79)$$

Substituting for the parameters of the system into equation (4.79), we have

$$P = 1 - 1.58Z^{-1} + 0.58Z^{-2}$$

$$Q = 1.2$$

$$G = 1.18 - 1.71Z^{-1} + 0.657Z^{-2}$$

$$H = 2.77 + 1.36Z^{-1}$$

Define the parameter set $\theta = (h_0, h_1, g_0, g_1, g_2)$, then the parameter set θ_{true} which satisfies the required control constraint is given by $\theta_{true} = (2.77, 1.36, 1.18, -1.71, 0.657)$. In this case, offset d = 1, the algorithm is started with initial values of P = 1, Q = 0 and

$\theta_{initial}$ = (0.8, 0.4, 0.4, -0.4, 0.2). Fig. 4.5 shows the system reference input, output, system control U(t), the parameters of $Q = q_0$, $H(Z^{-1}) = h_0 + h_1 Z^{-1}$ and $G(Z^{-1}) = g_0 + g_1 Z^{-1} + g_2 Z^{-2}$. As can be seen, all the parameters converge to the expected values. It should be noted that, when $\theta_{initial}$=(0.5, 0.36, 0.4, -0.4, 0.1), the estimation tended to diverge.

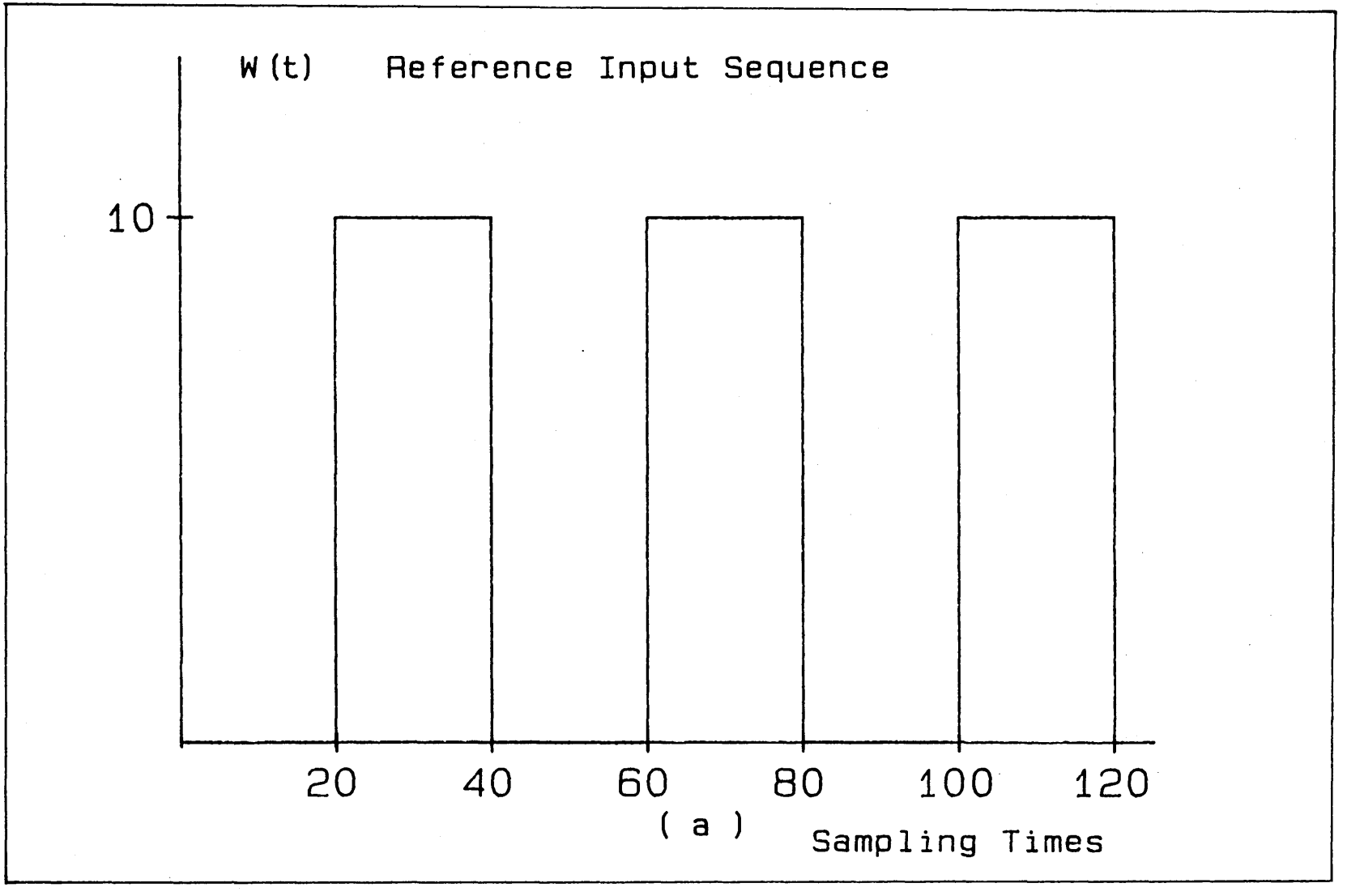


Fig. 4.5 System Response and Parameter Convergence for Example 1

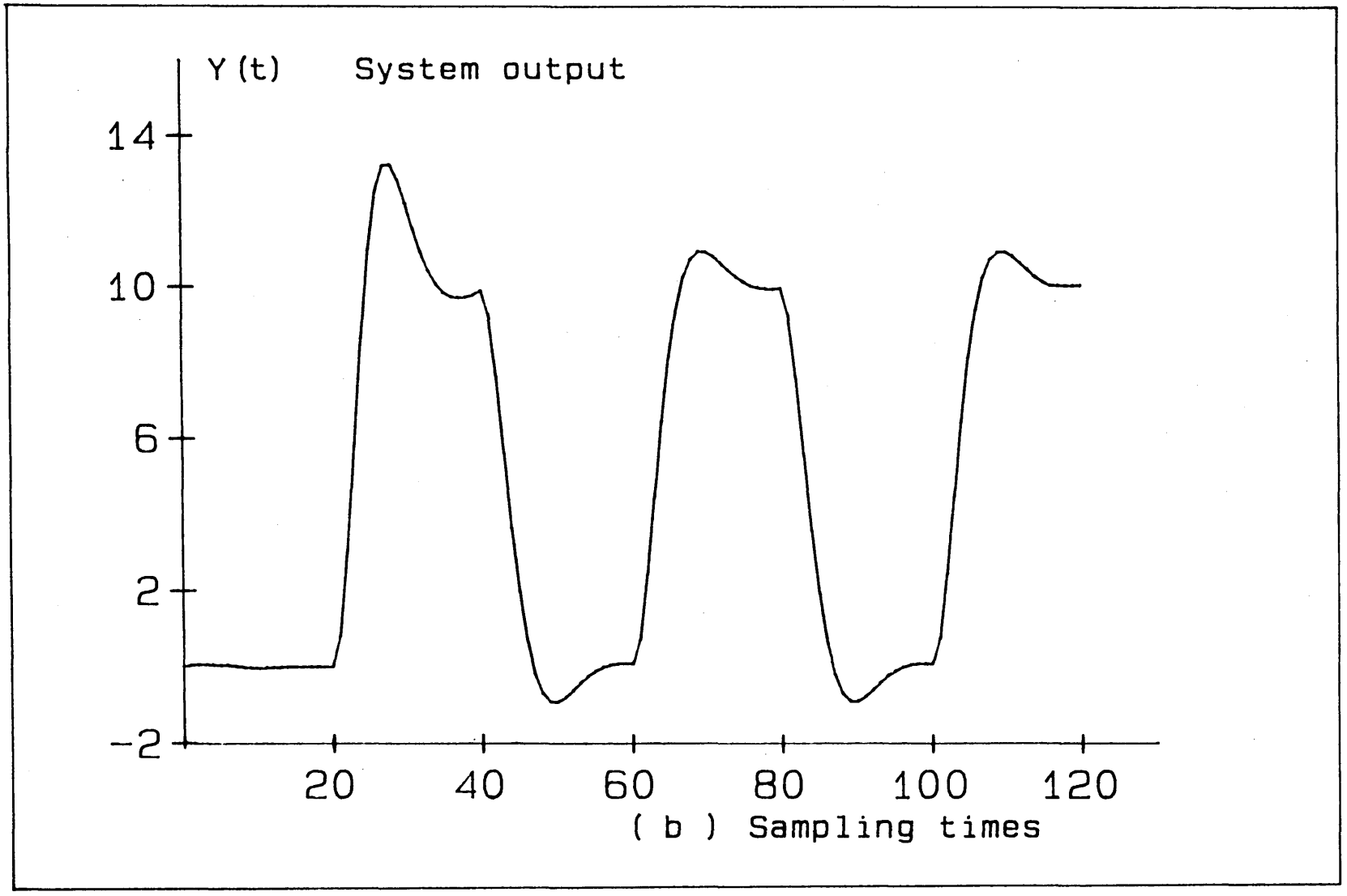


Fig. 4.5 System Response and Parameter Convergence for Example 1

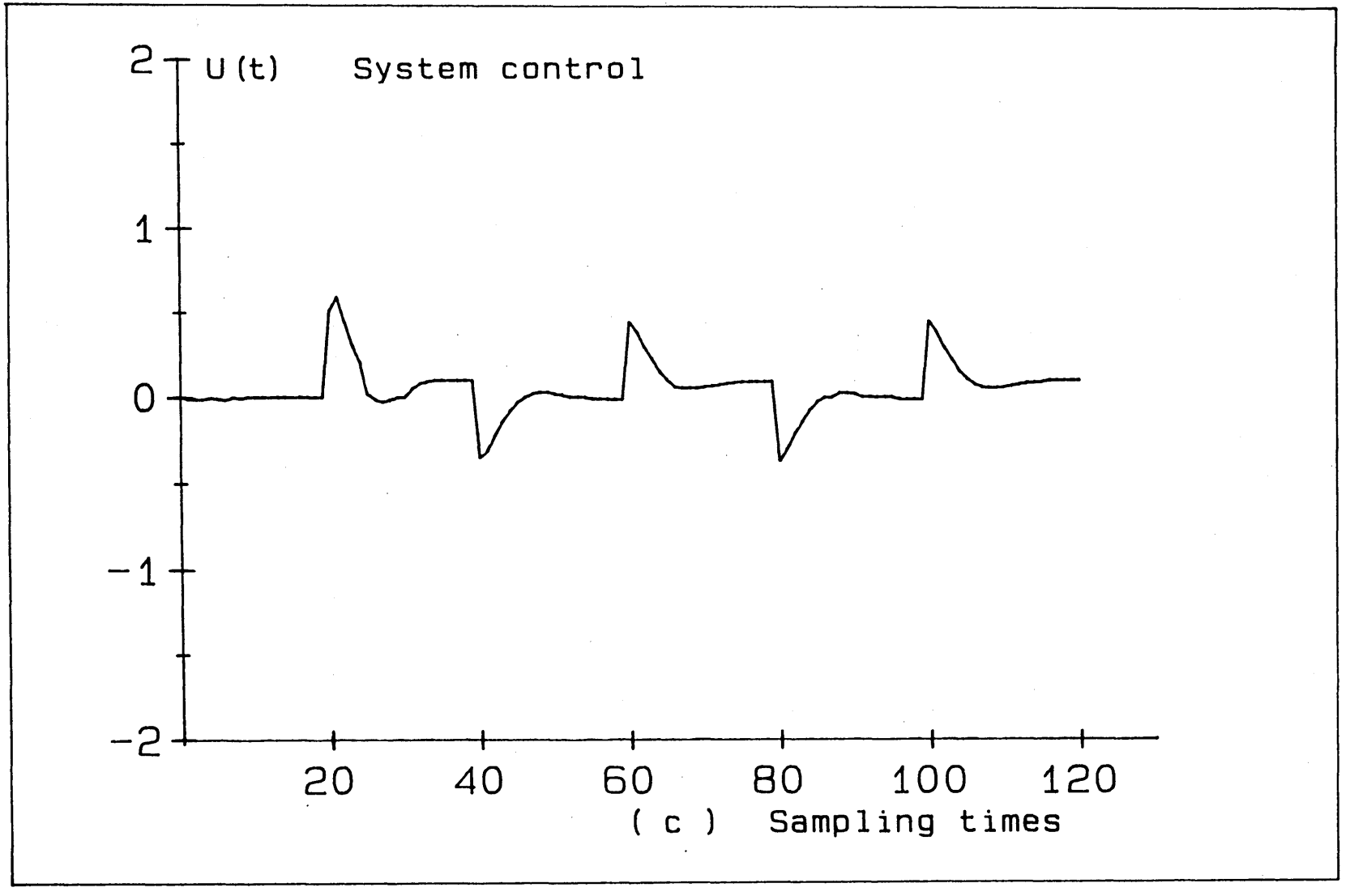


Fig. 4.5 System Response and Parameter Convergence for Example 1

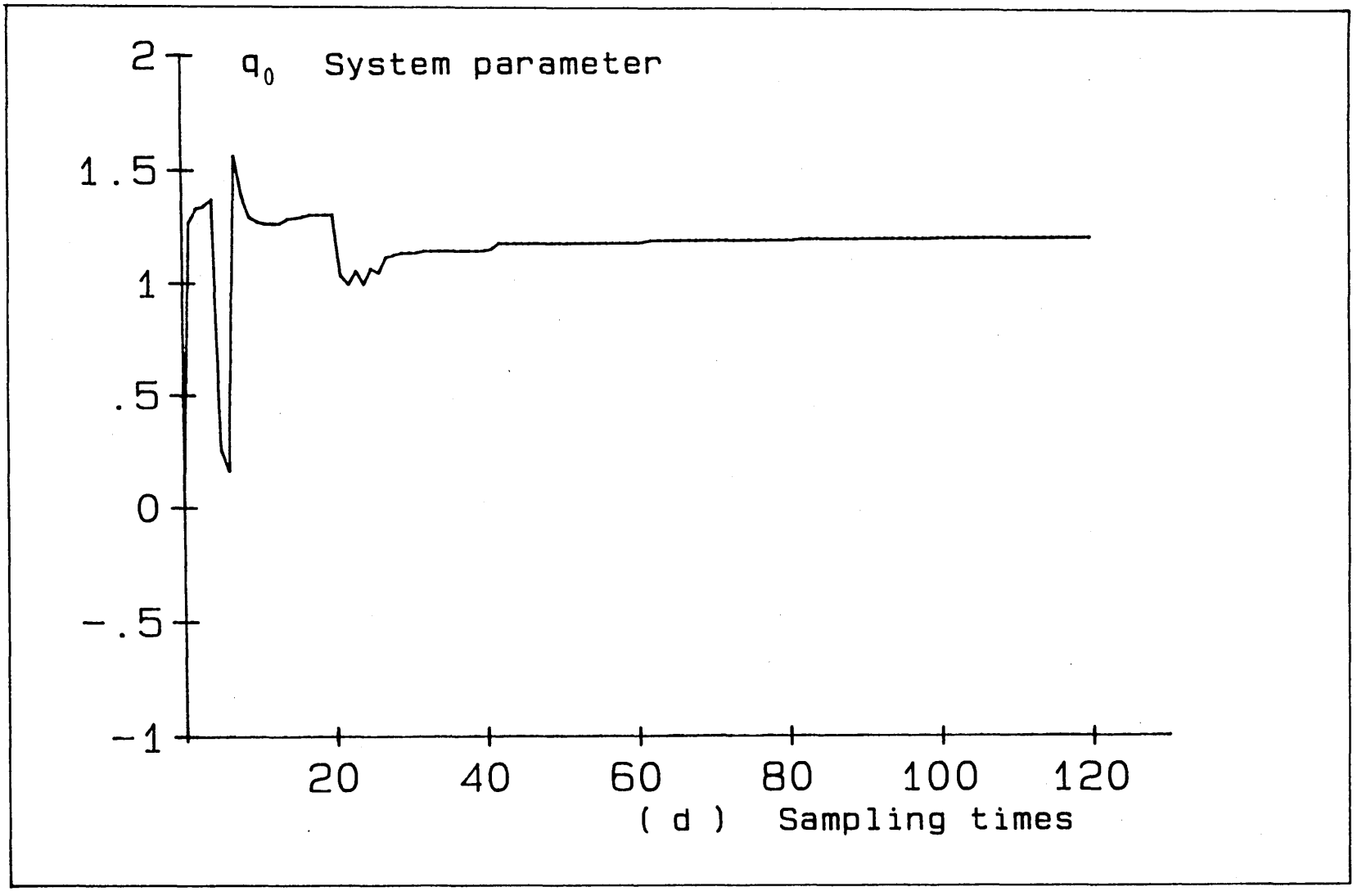


Fig. 4.5 System Response and Parameter Convergence for Example 1

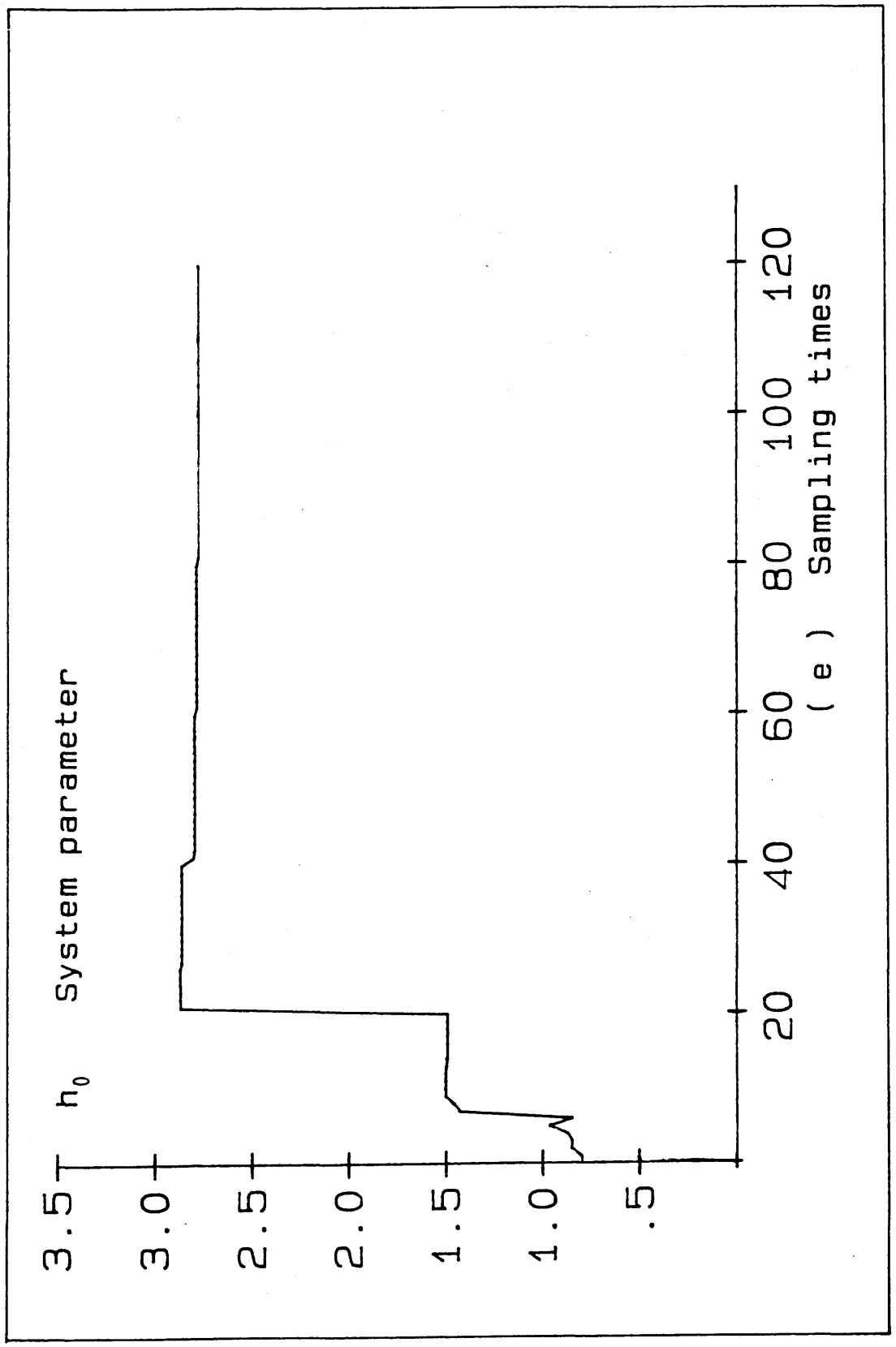


Fig. 4.5 System Response and Parameter Convergence for Example 1

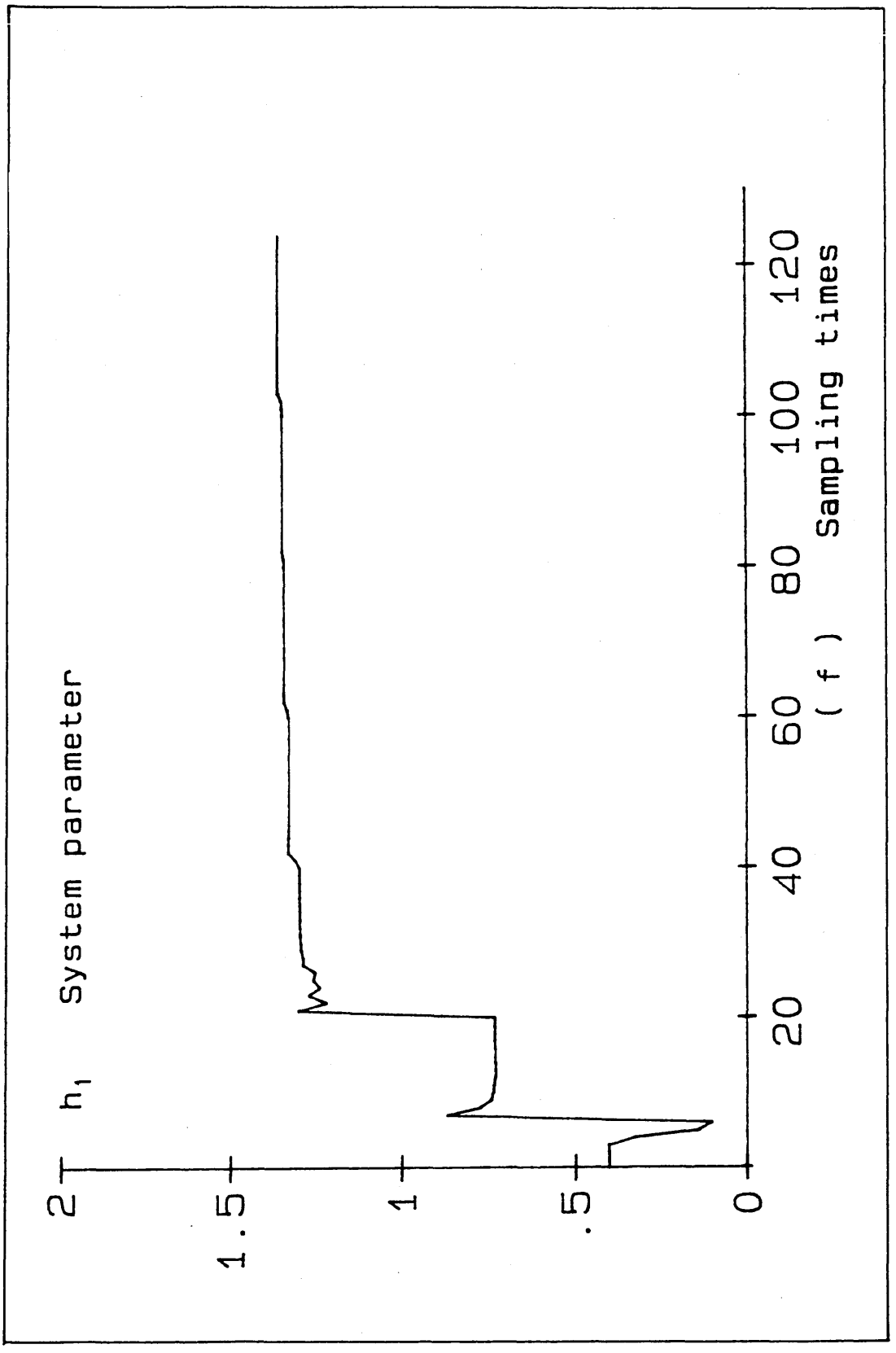


Fig. 4.5 System Response and Parameter Convergence for Example 1

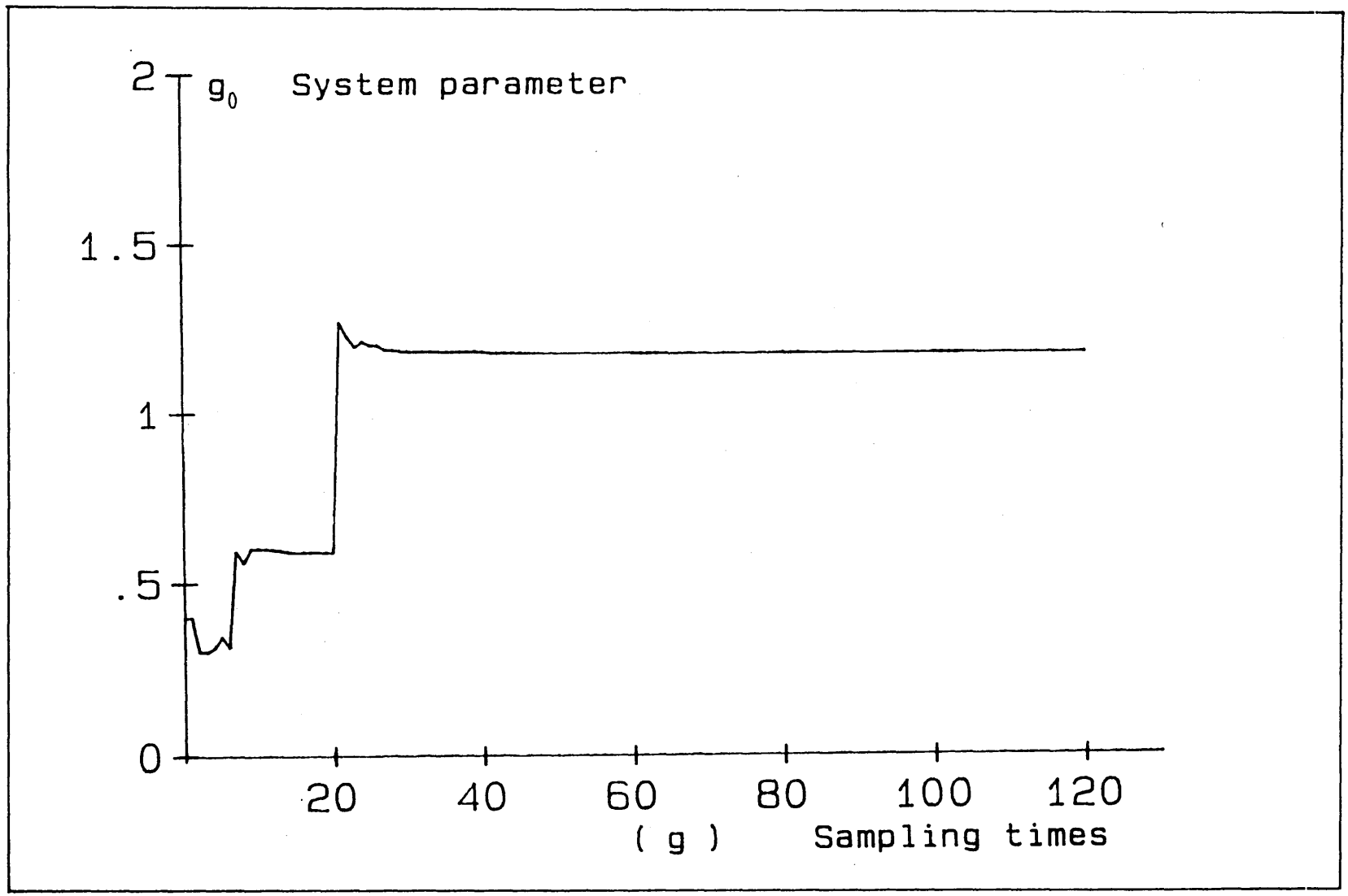


Fig. 4.5 System Response and Parameter Convergence for Example 1

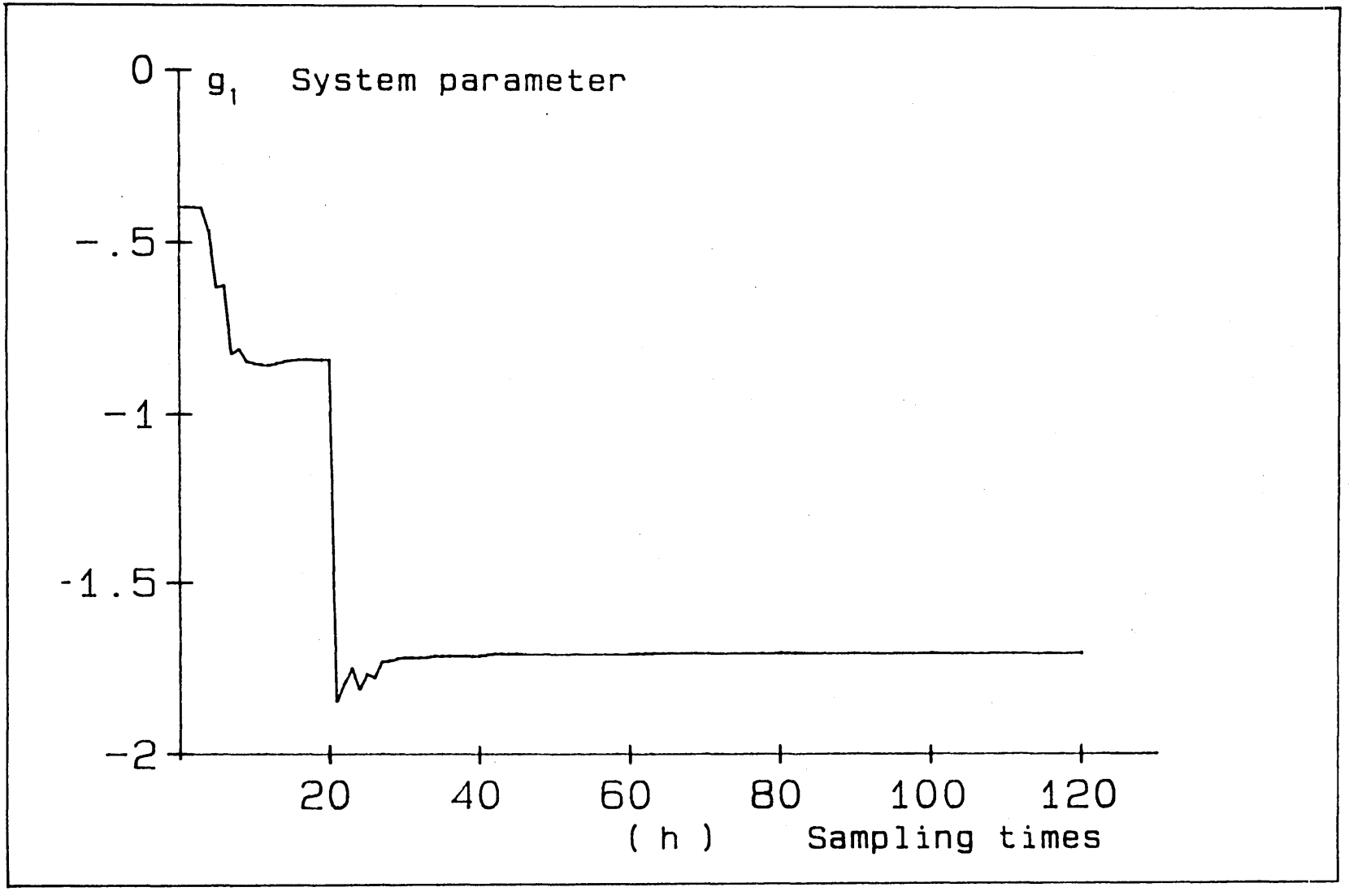


Fig. 4.5 System Response and Parameter Convergence for Example 1

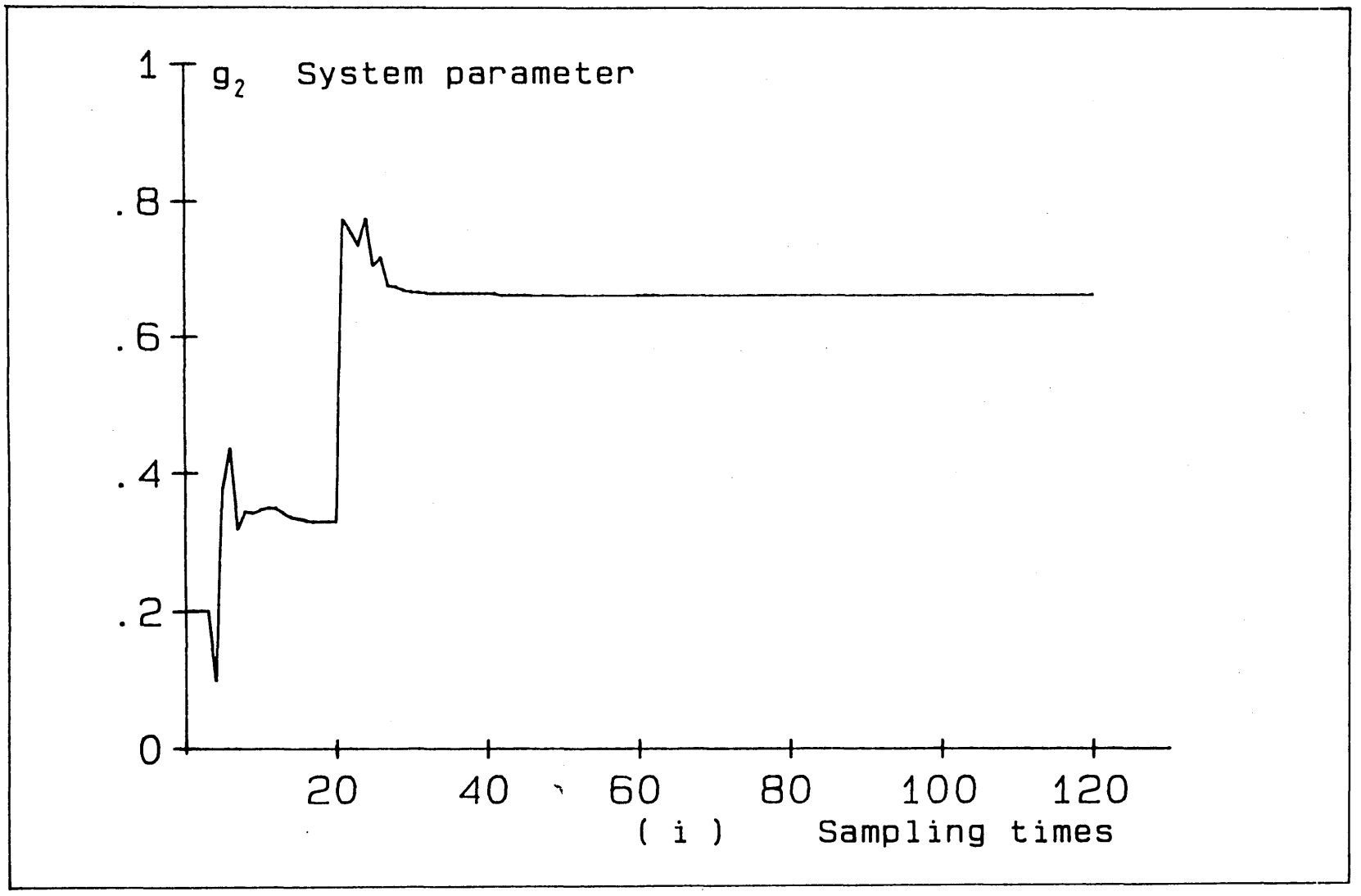


Fig. 4.5 System Response and Parameter Convergence for Example 1

Example 2

In this case, the situation is the same as in example 1, but the desired closed-loop poles are defined by $T(Z^{-1}) = 1 - 1.2Z^{-1} + 0.41Z^{-2}$, corresponding to the continuous time second order characteristic polynomial with $\zeta = 0.8$, $\omega_n = 0.19$. Fig. 4.6 shows the system output. Fig. 4.7 and Fig. 4.8 show the system outputs, if the desired closed-loop poles are defined by $T(Z^{-1}) = 1 - 1.2Z^{-1} + 0.604Z^{-2}$ and $T(Z^{-1}) = 1 - 0.82Z^{-1} + 0.41Z^{-2}$ respectively.

Example 3

The system to be considered is given by

$$Y(t) = \frac{0.48Z^{-1}}{1 - 0.91Z^{-1}} U(t) + d$$

corresponding to the continuous-time system

$$G(s) = \frac{5}{1 + 0.1s}$$

with sampling period $T = 0.01$ second. The desired closed-loop pole is defined by $T(Z^{-1}) = 1 - 0.517Z^{-1}$. The algorithm is started with initial values of $P = 1$ and $\theta_{initial} = (0.1, 0.2, -0.2)$. Fig. 4.9 shows the system output and the parameter of h_0, it can be seen that, h_0 converges to its expected value $h_0 = b_0 = 0.48$.

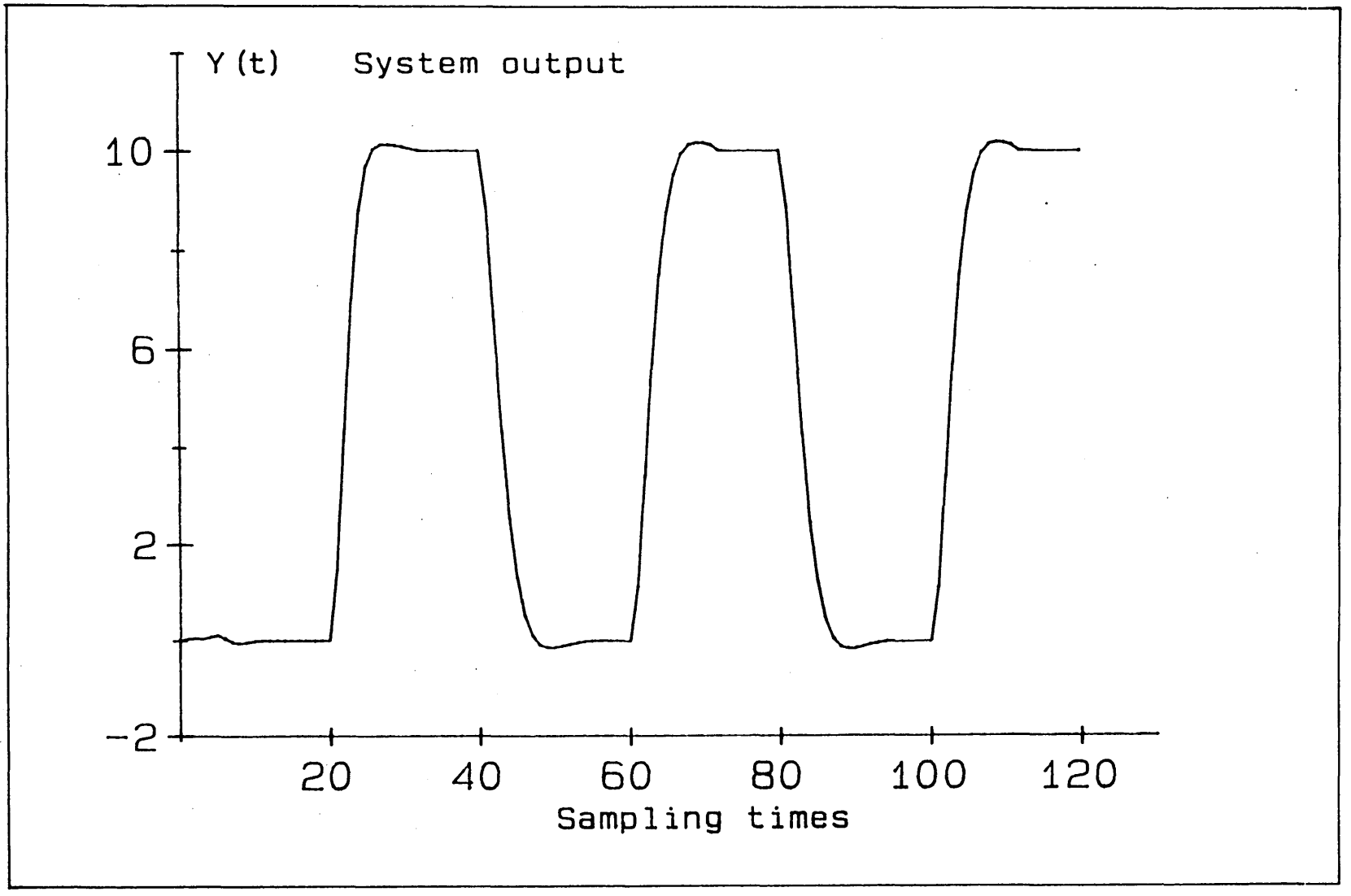


Fig. 4.6 System Response for Example 2
$T(Z^{-1}) = 1 - 1.2Z^{-1} + 0.41Z^{-2}$

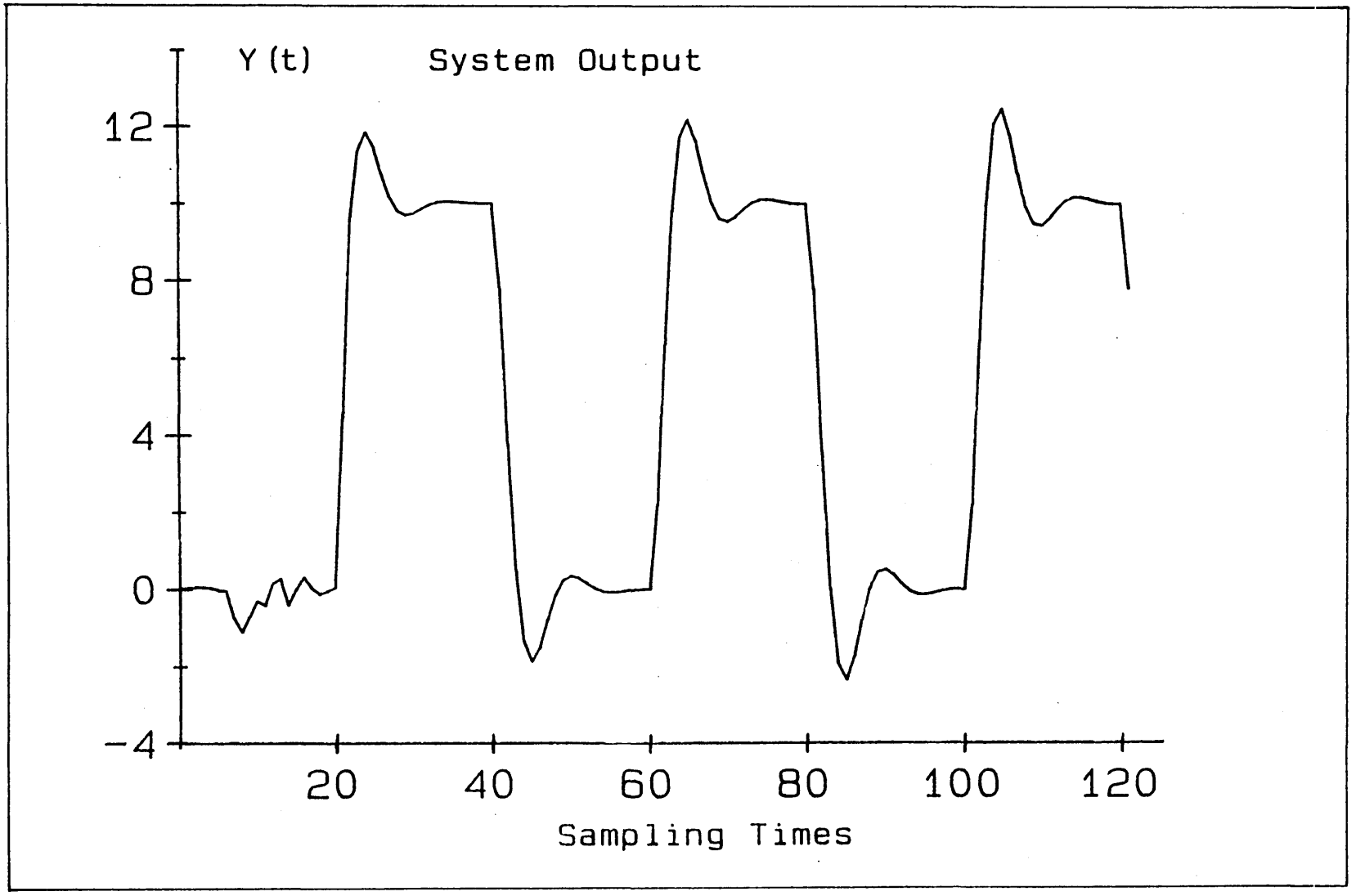


Fig. 4.7 System Response for Example 2
$T(Z^{-1}) = 1 - 1.2Z^{-1} + 0.604Z^{-2}$

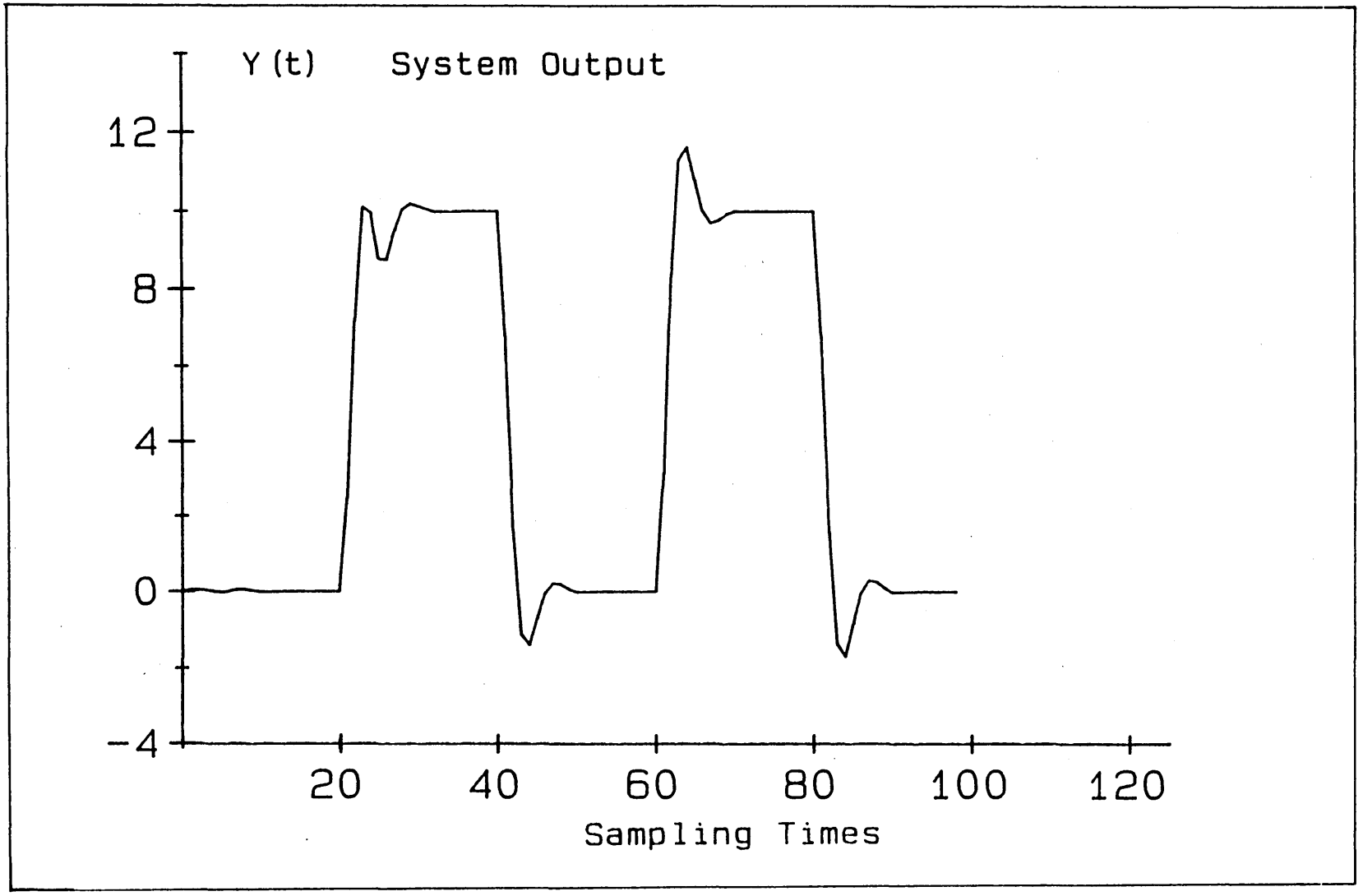


Fig. 4.8 System Response for Example 2
$T(Z^{-1}) = 1 - 0.82Z^{-1} + 0.41Z^{-2}$

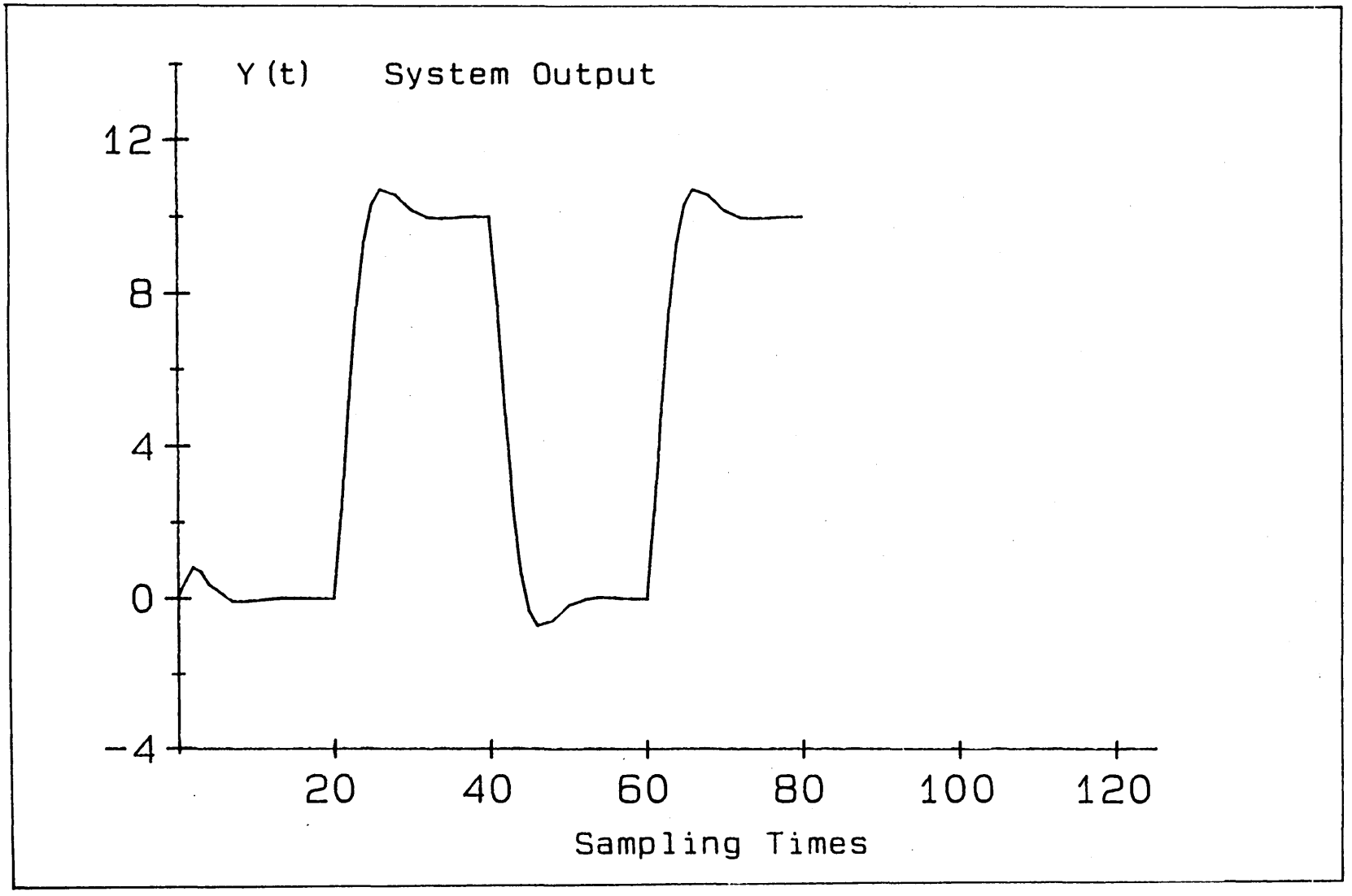


Fig. 4.9a System Response and Parameter Convergence for Example 3

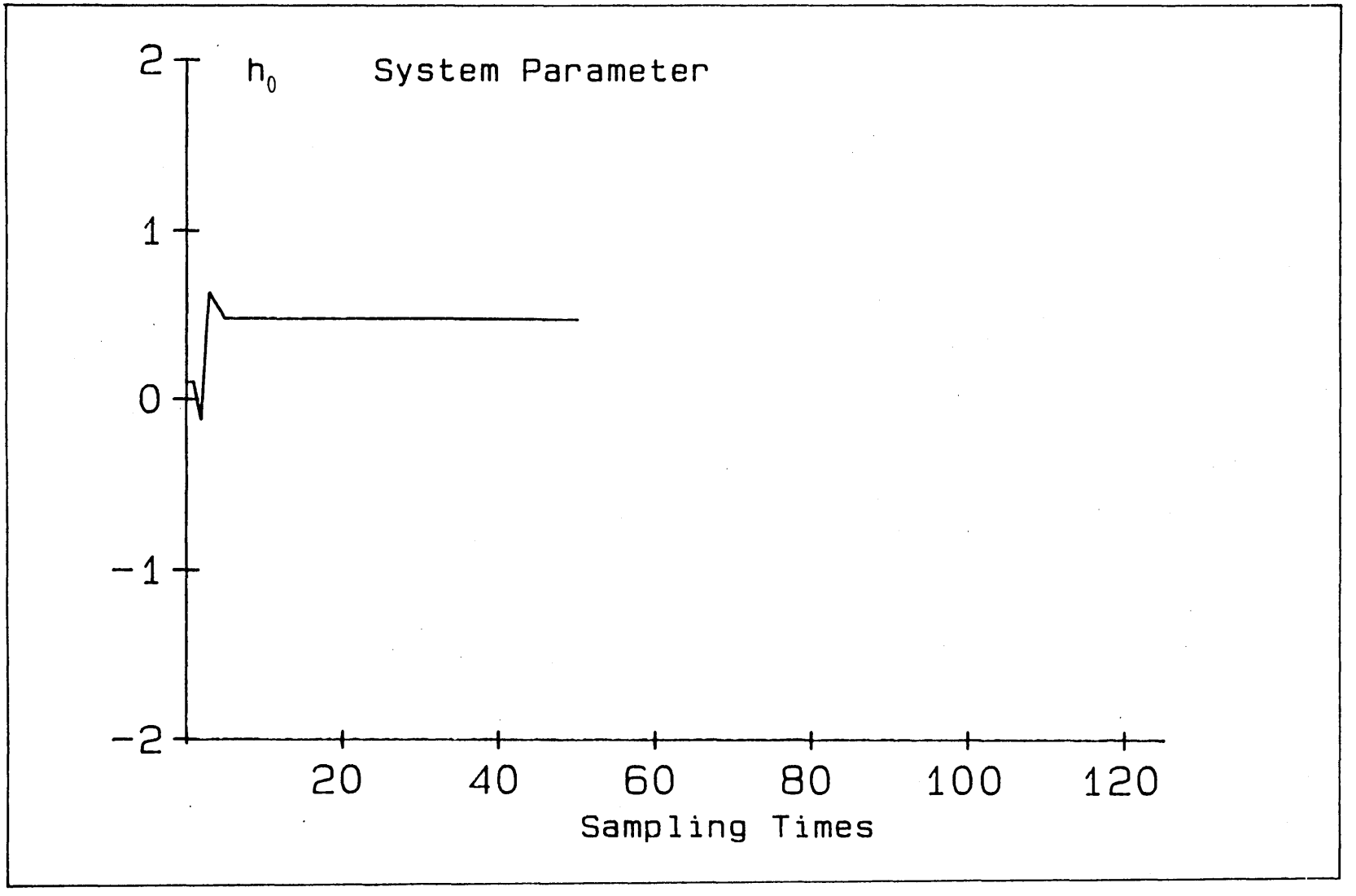


Fig. 4.9b System Response and Parameter Convergence for Example 3

Example 4

In this case, a system which is a stable and nonminimum phase second order system is to be considered [52]

$$Y(t) = \frac{Z^{-1}(-0.0231 + 0.0745Z^{-1})}{1 - 1.535Z^{-1} + 0.587Z^{-2}} U(t) + d$$

corresponding to the continuous-time system

$$G(s) = \frac{-s + 1}{(3s + 1)(5s + 1)}$$

with samples every second.

The algorithm is started with initial values of P = 1, Q = 0 and $\theta_{initial}$ = (0.014, 0.03, 0.5, -0.8, 0.2). The desired closed-loop poles are defined by $T(Z^{-1}) = 1 - 1.2Z^{-1} + 0.41Z^{-2}$, corresponding to the continuous time second order characteristic polynomial with $\zeta = 0.8$ $\omega_n = 0.56$. Fig. 4.10 shows the system output and the parameter of q_0. Fig. 4.11 shows the system output and the parameter of q_0, if $T(Z^{-1}) = 1 - 1.04Z^{-1} + 0.52Z^{-2}$.

When $\theta_{initial}$ = (0.005, 0.01, 0.2, -0.5, 0.1), the estimation tended to diverge.

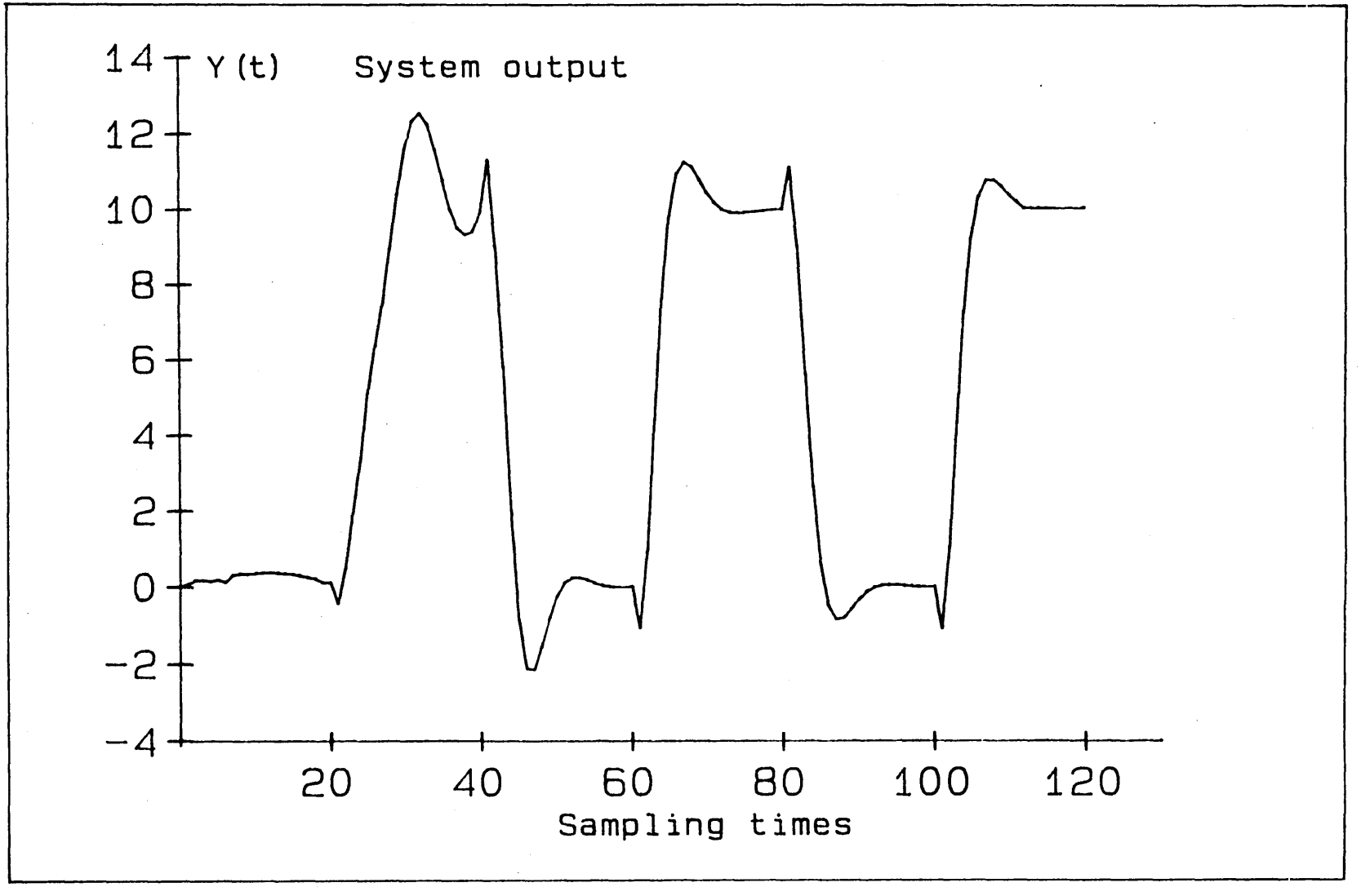


Fig. 4.10a System Response and Parameter Convergence for Example 4
$T(Z^{-1}) = 1 - 1.2Z^{-1} + 0.41Z^{-2}$

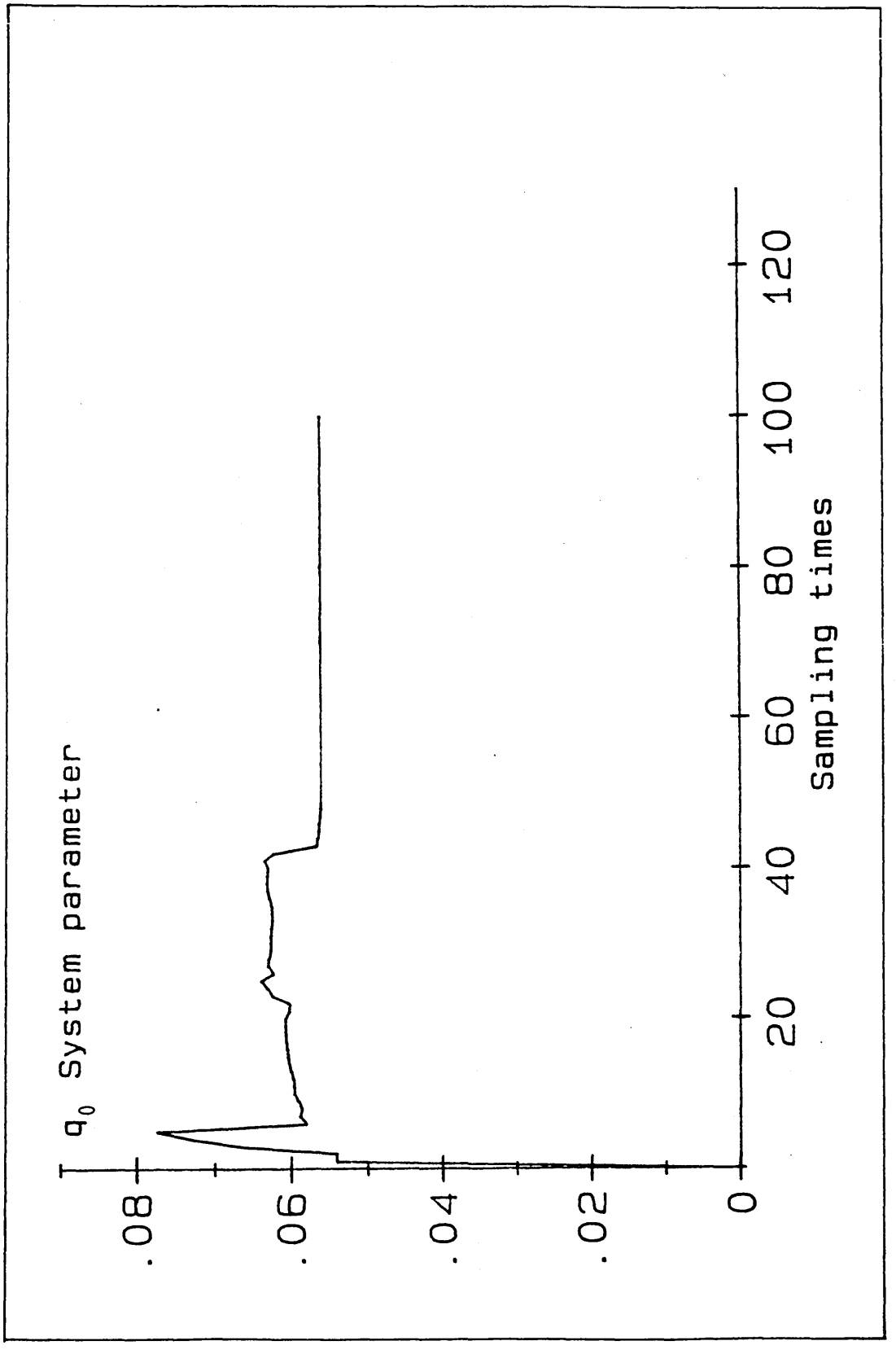


Fig. 4.10b System Response and Parameter Convergence for Example 4
$T(Z^{-1}) = 1 - 1.22Z^{-1} + 0.41Z^{-2}$

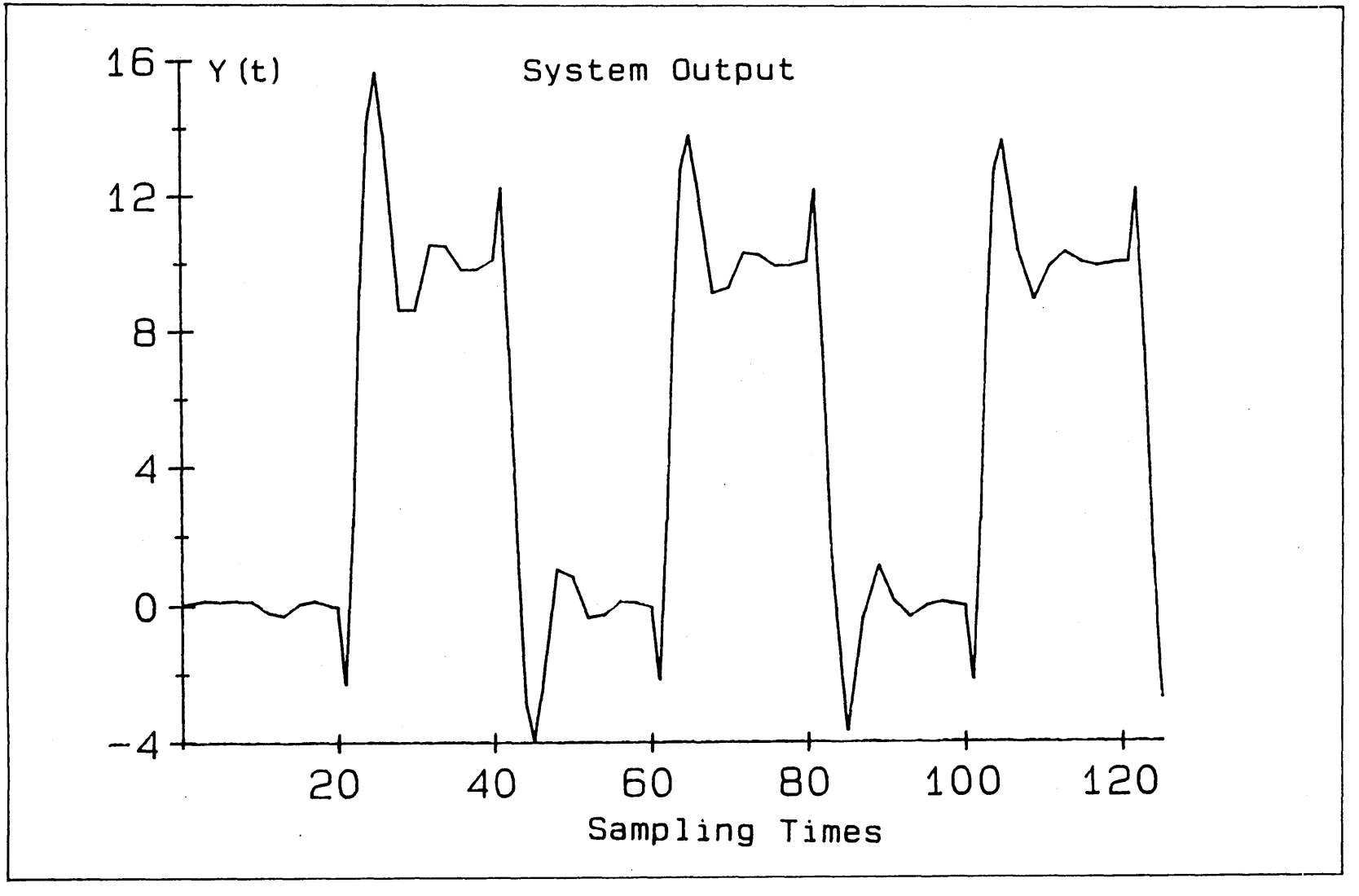


Fig. 4.11a System Response and Parameter Convergence for Example 4
$T(z^{-1}) = 1 - 1.04Z^{-1} + 0.52Z^{-2}$

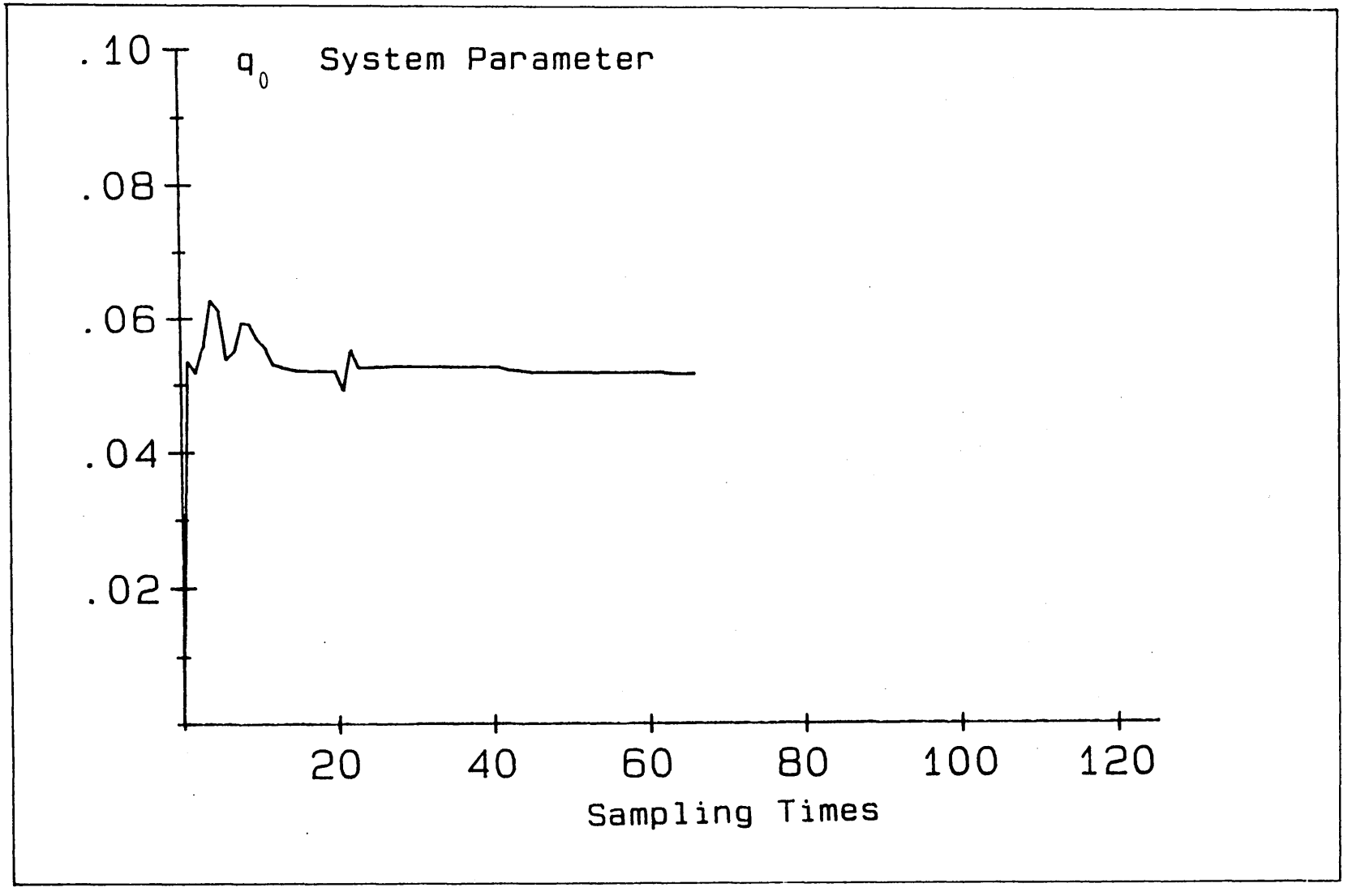


Fig. 4.11b System Response and Parameter Convergence for Example 4
$T(Z^{-1}) = 1 - 1.04Z^{-1} + 0.52Z^{-2}$

4.4.4 - Experiments and Results

The behaviour of the self-tuning PID controller has been tested in several experimental applications. The control algorithm was implemented using a 16-bit MC 68000 single board computer which was interfaced to the controlled system. The control algorithm was written in assembly language. For 16-bit fixed-point arithmetic, the program required less than 1.2k-bytes of RAM for the control program itself and 0.25k-bytes of RAM to store intermediate values. The total computation time for this program was less than 12 ms. Except for some very fast processes, this time was acceptable. However, the 16-bit configuration limted the accuracy of computation.

To improve the accuracy of computation, software was rewritten in 24-bit floating-point arithmetic. This version required 2k-bytes of RAM, of which about 0.5k-bytes were required for storing intermediate values. The total computation time was about 100 ms.

The programs for both fixed-point and floating-point arithmetic and their brief specifications are given in Appendix 10.

Experiment 1

The process in this case is an analogue simulator and the configuration of the control system is shown in Fig. 4.12.

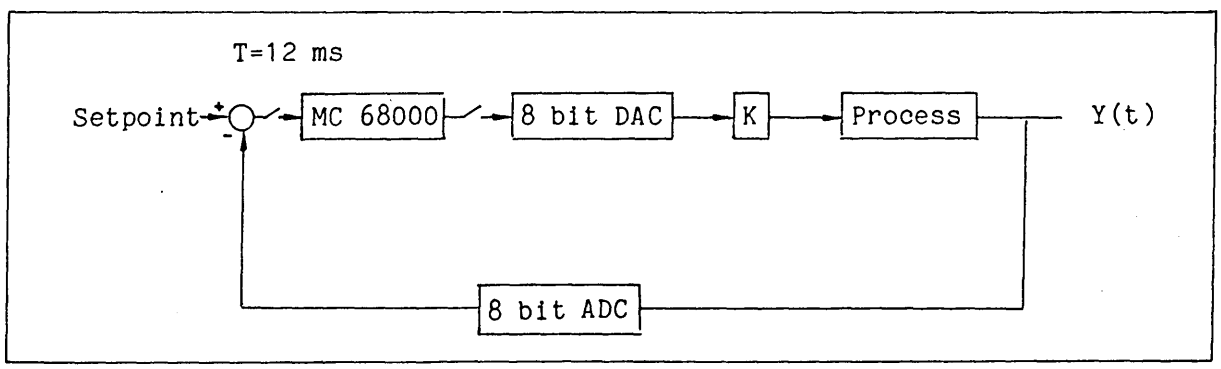


Fig. 4.12 Block Diagram of the Process Control System

The open-loop step response of the process is shown in Fig. 4.13, and from which the transfer function of the process is approximately given by

$$G(s) = \frac{K}{(1+0.064s)(1+0.015s)}$$

and

$$G(Z^{-1}) = \frac{Z^{-1}(0.052 + 0.042\,Z^{-1})}{1 - 1.28Z^{-1} + 0.374Z^{-2}}$$

with sampling period 12 ms.

Since polynomials A and B are known, equation (4.79) can be solved directly to find the values of the controller parameters which satisfy the required control constraint. Thus we have

$$\theta_{true} = (0.103,\ 0.042,\ 1.0,\ -1.2,\ 0.374)$$

with $T(Z^{-1}) = 1 - 1.363Z^{-1} + 0.52Z^{-2}$

The flowchart for the control algorithm implemented on the MC 68000 is shown in Fig. 4.14. In this case, the program ran continuously in a loop and the computation time was about 11.5 ms. The initial value of ρ_0 was taken equal to 128 (7FFF Hex with 7 bit integer and 8 bit fraction). The setpiont was periodically changed by the program from 0.08 to 0.47 (maximum value is equal to 1) for every 20 sampling periods, i.e. 0.23 s, as a system disturbance input for the estimation procedure. In actual process control, as soon as convergence was achieved, the estimation and the disturbance as well should be terminated and then the controller parameters would be fixed at the estimated values. Here, after 150 sampling periods, the estimation procedure was interrupted by the program itself and then the estimated data were read out.

Fig. 4.15 shows the system output during the estimation procedure. Table 1 lists 6 sets of estimated values of θ and the values of matrix P(t) when the estimation procedure finishes. The estimation started with

initial values of $P(Z^{-1}) = 1$, $Q(Z^{-1}) = 0$ and

$\theta_{initial}$=(0.063, 0.0468, 1.344, -1.61, 0.375) and the control constraint was given by $T(Z^{-1}) = 1 - 1.363Z^{-1} + 0.52Z^{-2}$.

Fig. 4.16 shows the closed loop step responses of the system with the controller parameters of

$\theta_{initial}$= (0.063, .0468, 1.344, -1.61, 0.375) and

θ = (0.078, 0.027, 1.18, -1.35, 0.43). Experiments showed that with $\theta_{initial}$ = (0, 0, 0, 0, 0), the convergence of the estimation process became very poor.

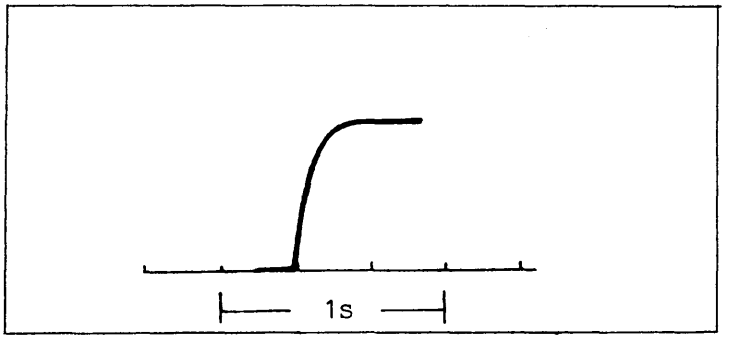


Fig. 4.13 Step Response of the Process

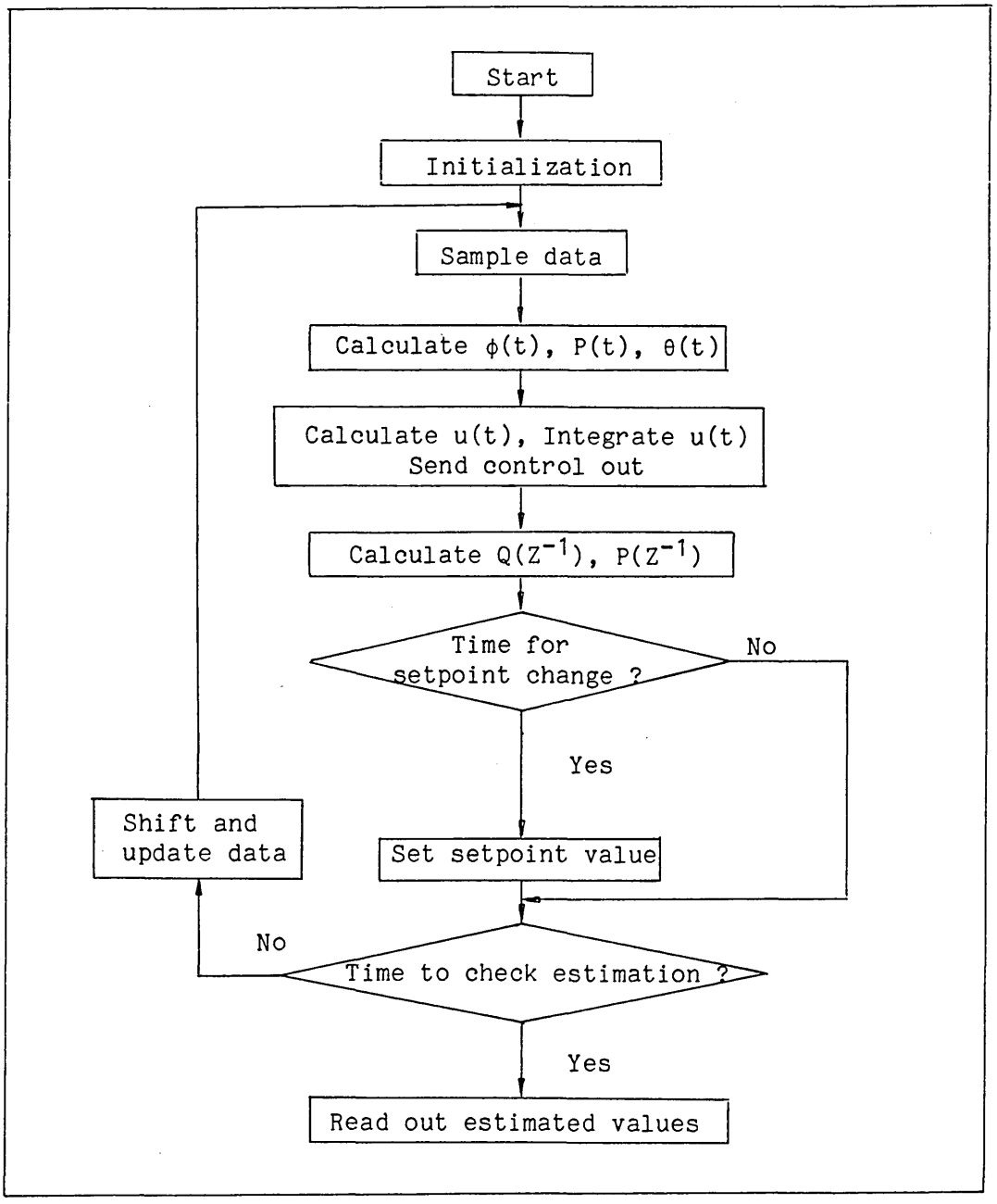


Fig. 4.14 Flowchart for the Estimation Process

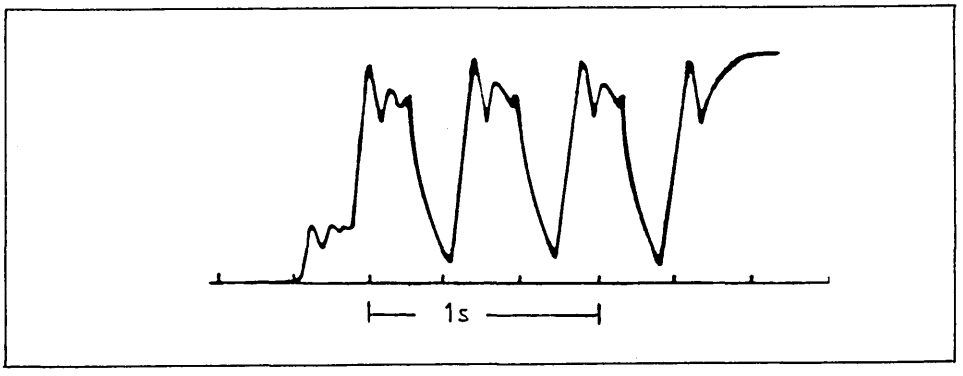


Fig. 4.15 System Output During the Tuning Process

h_0	h_1	g_0	g_1	g_2
0.082	0.023	1.18	-1.29	0.45
0.086	0.035	1.17	-1.36	0.46
0.07	0.031	1.20	-1.31	0.43
0.09	0.031	1.15	-1.27	0.51
0.078	0.027	1.18	-1.35	0.43
0.082	0.031	1.27	-1.31	0.41

$$P(t) = \begin{bmatrix} 0.035 & 0.008 & 0.051 & -0.094 & 0.16 \\ 0.012 & 0.035 & 0.071 & -0.156 & 0.20 \\ 0.023 & -0.02 & 11.9 & -20.83 & 9.91 \\ -0.02 & 0.035 & -21.13 & 42.02 & -21.27 \\ 0.102 & 0.082 & 9.53 & -21.55 & 12.41 \end{bmatrix}$$

Table 4.1 Estimated Values and Matrix P(t)

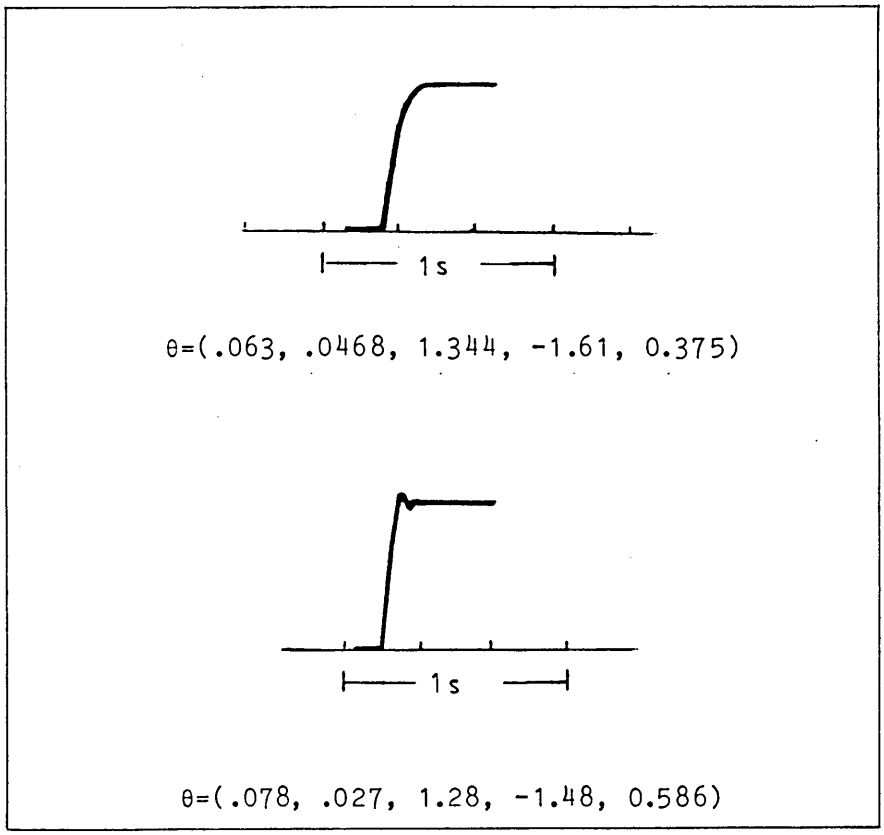


Fig. 4.16 Step Response of the Closed-loop System

Experiment 2

The configuration of the control system is shown in Fig. 4.12, but the process is a hydraulic motor system and the sampling period is 12 ms.

The continuous time transfer function of the process is approximately given by

$$G(s)= \frac{e^{-0.012s}}{(0.06s+1)(0.01s+1)}$$

The flowchart of the estimation algorithm is shown in Fig. 4.17 and the program execution was interrupt driven. In this case, at the commisioning stage a fixed parameter PID controller was used to control the process directly and the setpoint was suitably disturbed. The outputs of the process and the controller were used to estimate the parameters for the self-tuning PID controller. The self-tuning PID controller itself did not generate a control signal. When convergence was achieved, the estimated values were transmitted to the PID controller. Then the parameters were fixed until retuning was considered necessary. Fig. 4.18 shows the system output at the commissioning stage. However, in this experiment at point "a" (Fig. 4.18), the computation was interrupted by the program and then the estimated values were read out. Table 4.2 lists some estimated values with initial values of θ = (0.2, 0.01, 1.6, -1.7, 0.4), and control constraint $T(Z^{-1}) = 1 - 1.2Z^{-1} + 0.41Z^{-2}$

It was found that in order to have better estimation results, at the commissioning stage, the range of the disturbance at the setpoint and the estimation times should reach certain values. Experiments also showed that the controller fixed with the estimated values listed in Table 4.2 gave rise to a poor response (seriously overdamped). In order to obtain a good performance, the estimated values in Table 4.2 had to be

modified.

Fig. 3.30 (p.103) shows the closed loop response of the system with the controller whose parameters are those estimated with some modification, i.e., θ = (0.3, 0.19, 1.33, -1.37, 0.125). It is judged that, because of dead time in the process, the discrepancy between the model needed for this algorithm and the actual process model causes the behaviour of the estimated parameters to be poor.

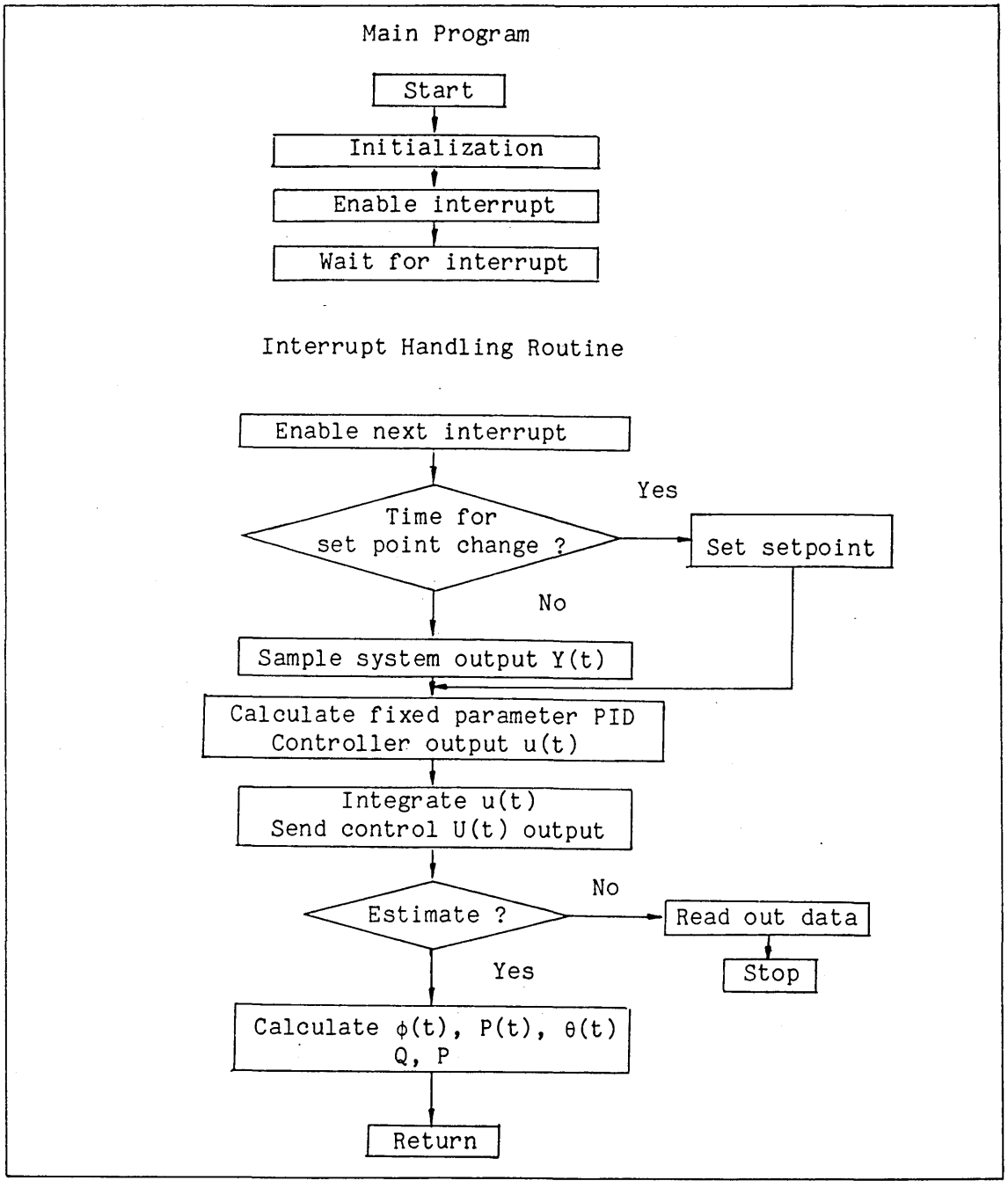


Fig. 4.17 Flowchart of the Estimation Algorithm

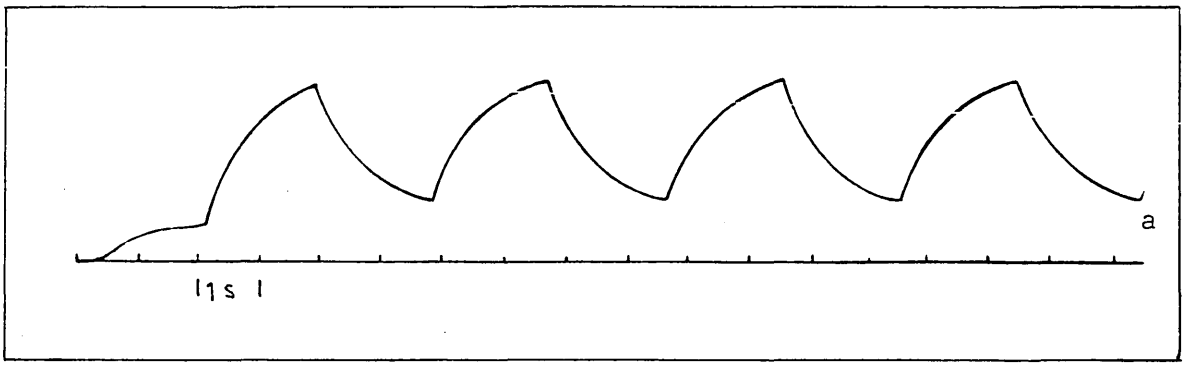


Fig. 4.18 System Output at Commissioning Stage

h_0	h_1	g_0	g_1	g_2
0.406	-0.047	1.473	-1.387	0.289
0.359	-0.027	1.363	-1.340	0.234
0.469	-0.034	1.437	-1.332	0.214
0.469	-0.039	1.453	-1.20	0.238
0.457	-0.047	1.410	-1.293	0.352
0.398	-0.039	1.484	-1.324	0.313

Table 4.2 Some Estimated Values

Experiment 3

For this experiment, a temperature platform was used, being able to provide a temperature range from room temperature up to 80°C. Temperature control was achieved by varying the average output voltage of the associated power supply. A temperature sensor was used to detect the platform temperature. The configuration of the control system is shown in Fig. 4.12.

From the open loop step response of the process as shown in Fig. 4.19, the transfer function of the process is given by

$$G(s) = \frac{1}{1 + \tau s}$$

where $\tau = 840$ second

and $$G(Z^{-1}) = \frac{b_0}{1 - a_0 z^{-1}}$$

where $b_0 = 0.0465$, $a_0 = 0.954$, with sampling period 40 second. From equation (4.79), the parameters of the PI controller are given by Table 4.3.

$T = 1-0.6Z^{-1}$	$\theta_{true} = (0.0465, 1.354, -0.954)$
$T = 1-0.7Z^{-1}$	$\theta_{true} = (0.0465, 1.254, -0.954)$
$T = 1-0.8Z^{-1}$	$\theta_{true} = (0.0465, 1.154, -0.954)$

Table 4.3

The flowchart of the estimation algorithm is the same as in experiment 2. Fig. 4.20 shows the system reference input sequence and output under the periodic setpoint change as system disturbance input at commissioning stage. Table 4.5 gives some estimation results with different initial values and the values of the matrix P(t) when the estimation procedure finishes. The commisioning stage is about 100 sampling periods, i.e., 4000 second. Experiments showed that parameters converged to their expected values with very high repeatability.

Closed loop pole	Estimated Values
$T = 1-0.6Z^{-1}$	$\theta = (0.037, 1.34, -1.04)$
$T = 1-0.7Z^{-1}$	$\theta = (0.036, 1.29, -1.09)$
$T = 1-0.8Z^{-1}$	$\theta = (0.04, 1.24, -1.13)$
Initial Values	$\theta = (0.1, 0.4, -2)$

Table 4.4

Closed Loop Pole	Estimated values
$T = 1-0.6Z^{-1}$	$\theta = (0.036, 1.34, -0.92)$
$T = 1-0.7Z^{-1}$	$\theta = (0.04, 1.30, -0.97)$
$T = 1-0.8Z^{-1}$	$\theta = (0.037, 1.26, -1.02)$

$$P(t) = \begin{bmatrix} 67.47 & 12.37 & -12.09 \\ 12.37 & 510.56 & -487.06 \\ -12.11 & -487.06 & 515.63 \end{bmatrix}$$

Initial Values $\theta = (0.01, 2.0, -0.4)$

Table 4.4

With $\theta_{initial} = (0, 0, 0)$, the estimation is divergent, Table 4.5 gives the estimated results and the values of the matrix P(t) when the estimation procedure finishes. Fig. 4.21 gives the closed loop step response of the system with the controller parameters $\theta = (0.04, 1.30, -0.97)$. When controller parameters were set with $\theta_{initial} = (0.01, 2.0, -0.4)$, the closed loop system did not work at all.

Closed loop pole	Estimated values
$T = 1-0.8Z^{-1}$	$\theta = (0, 0.102, -0.091)$

$$P(t) = \begin{bmatrix} 0.003 & 0.029 & 0.04 \\ 0.028 & 495.84 & -476.3 \\ 0.04 & -476.3 & 498.5 \end{bmatrix}$$

Initial values $\theta = (0, 0, 0)$

Table 4.5

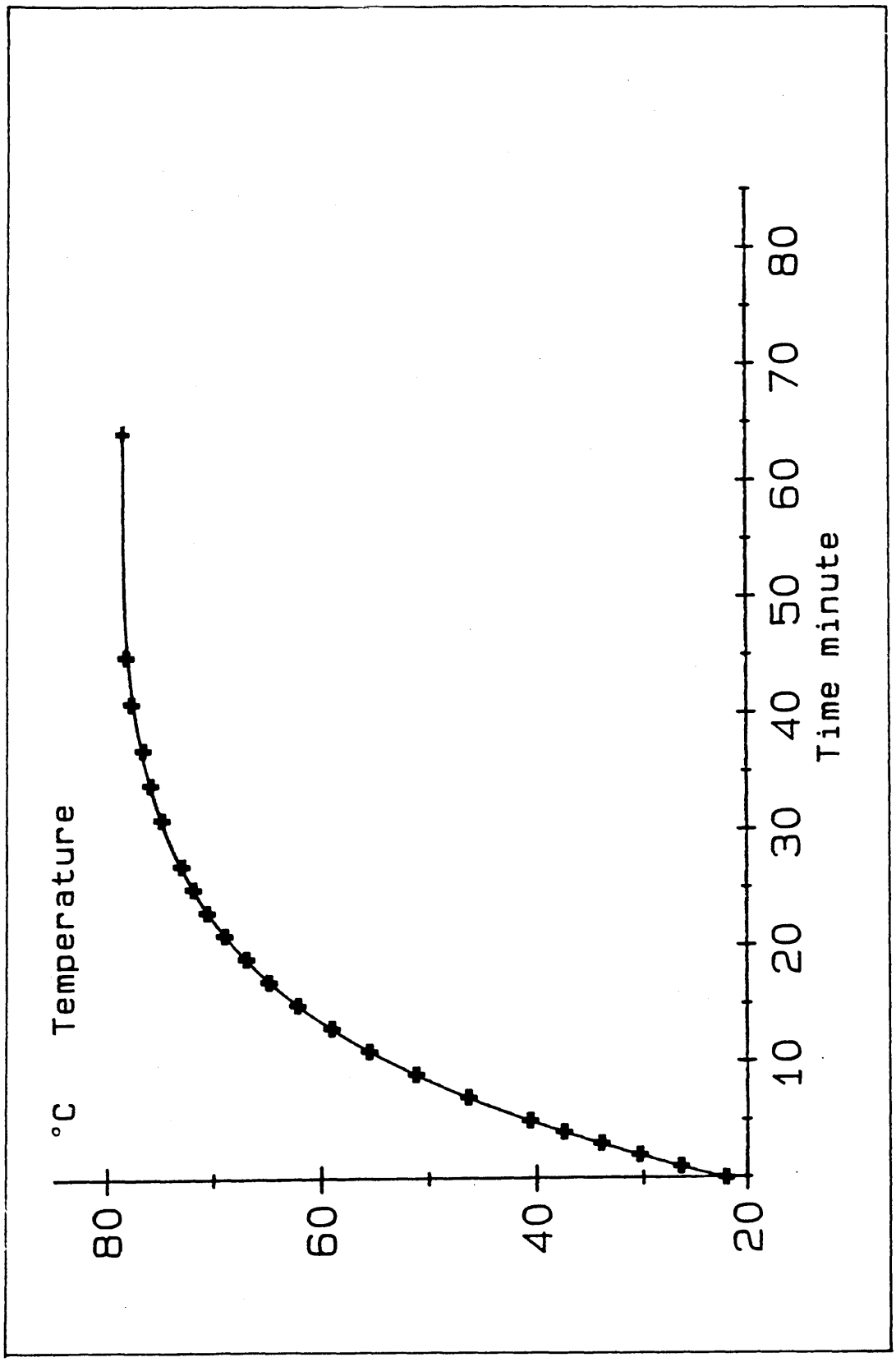


Fig. 4.19 Step Response of the Process

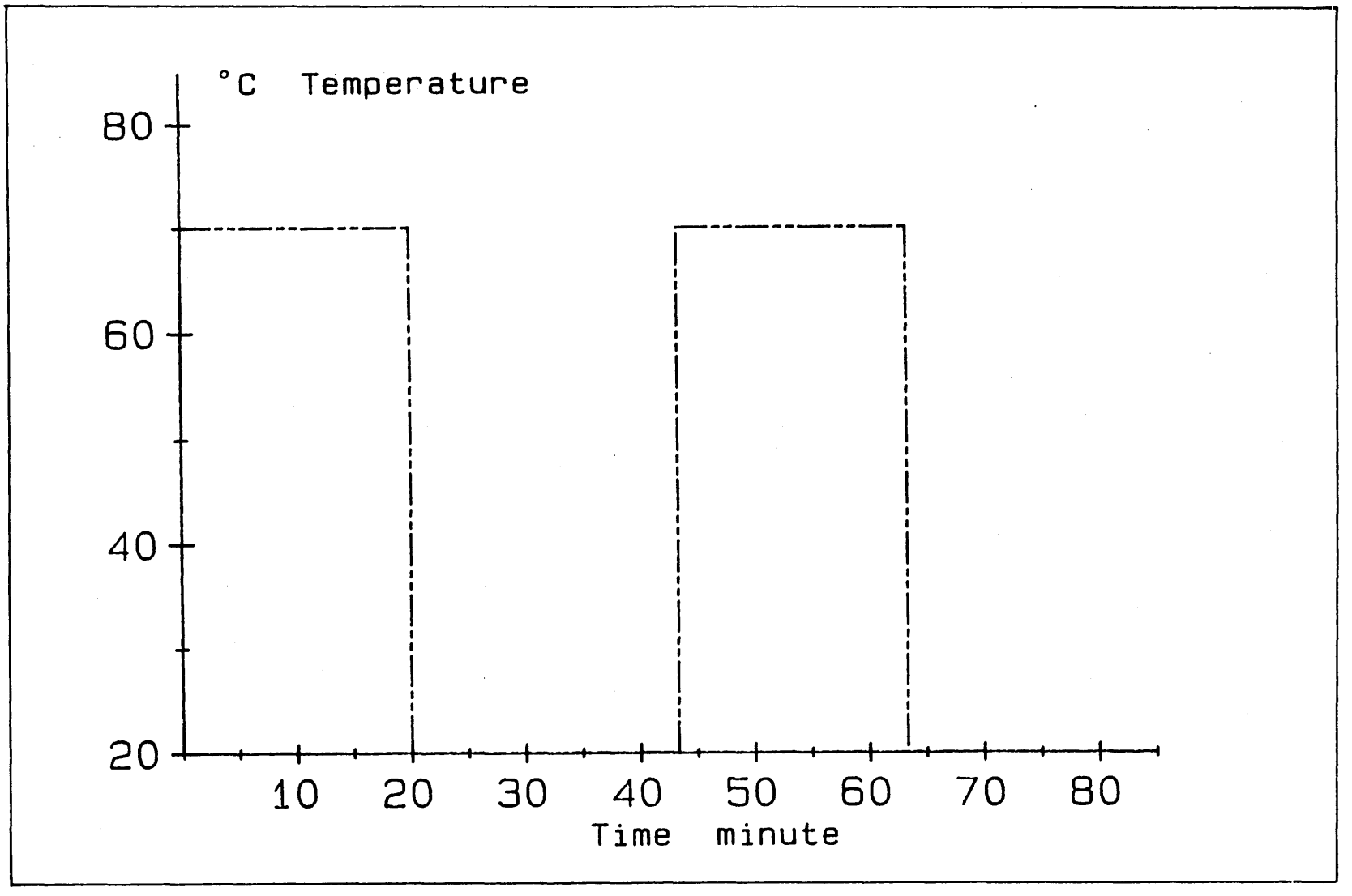


Fig. 4.20a System Reference Input Sequence at Commissioning Stage

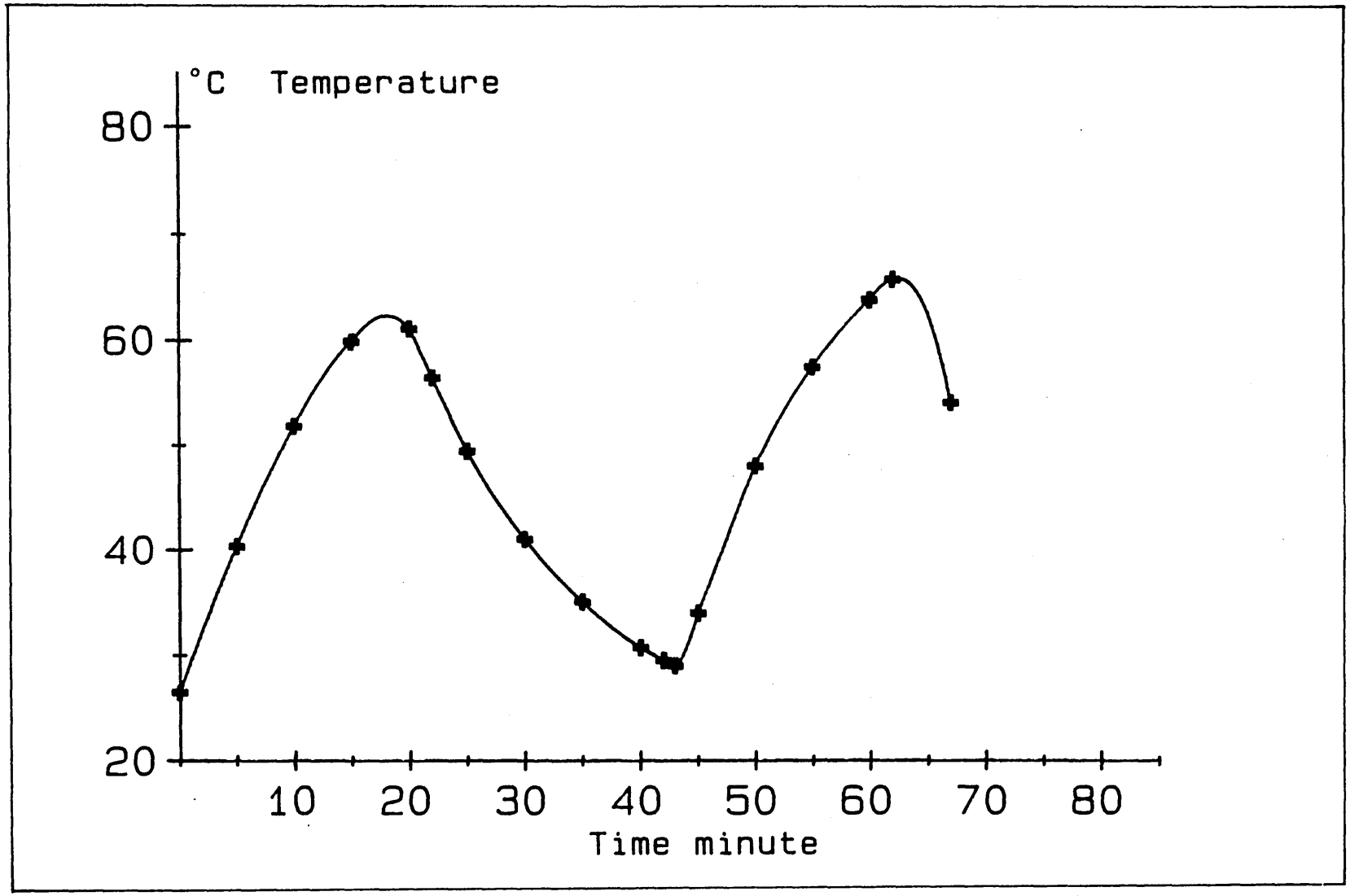


Fig. 4.20b System Output at Commissioning Stage

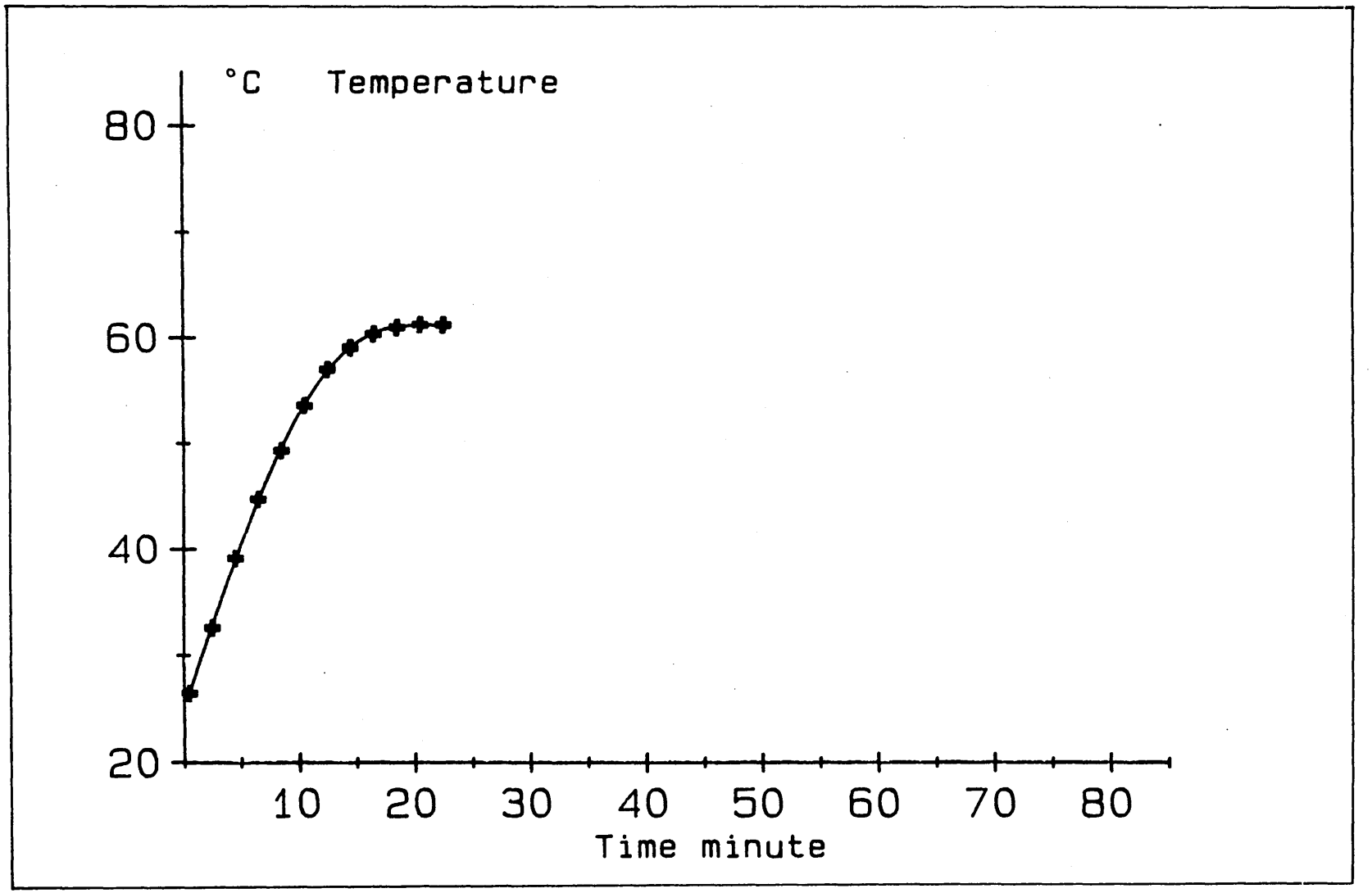


Fig. 4.21 Closed Loop Step Response of the System

Discussion

The experimental applications of the self-tuning PID controller have indicated that it is not entirely straightforward to use this kind of controller. There are many precautions that must be taken and also it is important to have as much a prior knowledge about the process as possible. The initial values of the parameters to be estimated must be prespecified with care otherwise the estimation will diverge. So if the process has been controlled before with a conventional controller, the initial values should be such that they correspond to the controller used before. The system set point also should be suitably disturbed in order to have a proper conditioning of signals which will give rise to a good convergence.

Although the 12 ms operation time of fixed-point arithmetic is acceptable for many fast processes, including large electrical and hydraulic drive systems, the accuracy of the computation limts its application in cases where unusually high accuracy is desired. The 100 ms operation time of floating-point arithmetic is acceptable for some fairly slow process, such as temperature, flow, level, etc, with very high operational accuracy, thereby resulting in better performance with the controller. Therefore this floating-point algorithm can be recommended to tune a PID controller either as an initial tuning device, or when required, provided that the model of the process is suitable.

When a process exhibits large apparent dead-time characteristics, the self-tuning PID algorithm is no longer suitable for this model, and like a conventional PID controller, it does not work very well. However, other self-tuning controllers can improve this situation.

Conclusion

A self-tuning PID controller with pole assignment for a deterministic process has been presented and investigated in this chapter. The algorithm proposed here is the same as the self-tuning controller for deterministic systems [35], but an integrator is incorporated into the control loop, which results in a self-tuning controller with PID structure. As we are dealing with a deterministic system, an implicit algorithm can be used. But control system identification is only feasible under disturbed conditions.

The algorithm has been implemented on the MC 68000 with both 16-bit fixed-point and 24-bit floating-point arithmetics. It has been shown that the algorithm can be easily implemented on a microcomputer system or on single board computers. Experimental applications have indicated that the floating-point algorithm performs well and can be used in industrial applications.

The major use of the controller proposed here would be as an initial tuning device to determine the controller parameters, or to allow the the controller to track the operating conditions of the slowly time-varying or non-linear process under contol.

CHAPTER 5 : Conclusion and Recommendations for Future Work

5.1 - Conclusion

An analysis of the dieless wire drawing process, based on Non-Newtonian fluid behaviour of the polymer melt, has been developed. The effect of the thermal energy generated by viscous "friction" in the wire drawing process has been taken into consideration by coupling the energy equation into the analysis and assuming the polymer viscosity to be dependent on temperature. The theoretical results in terms of predicting the deformation profile, the pressure distribution within the DRU and the drawing load during the drawing process, agreed very well with those observed experimentally. An extensive experiment has been carried out, in which parameters such as drawing speeds, the temperature of the polymer melt and type of polymers were varied.

Based on the theoretical and experimental results, a microcomputer-based control system for the dieless wire drawing process has been designed and constructed in order to produce wires of desired qualities. A method detecting the PRA of the wire during the drawing process was proposed. The PRA indicator worked well under steady-state conditions, but fluctuation in readings occurred under transient conditions.

The qualities of the wire of a uniform diameter drawn by the new drawing system were found to be comparable to those drawn by using the conventional drawing process. Tapered wire of a uniform change in area with a desired rate for a given length was produced. The error in the PRA of the produced wires was within the range of ±2 percent.

A self-tuning PID controller with pole assignment for deterministic systems has been proposed and investigated. An implicit algorithm was used and the control system identification was only feasible under

disturbed conditions. The algorithm has been implemented on the MC68000 microcomputer with both 16-bit fixed-point and 24-bit floating-point arithmetics. Experimental applications indicated that the floating-point algorithm performed well and could be used in practical applications. The major use of the controller is as an initial tuning device to determine the controller parameters, or to provide an adaptive control system in which the PID controller parameters are modified to adapt to a changing situation.

5.2 - Recommendations for Future Work

In the theoretical analysis of the dieless wire drawing process, the amount of heat generated from the deformation of the wire needs to be taken into consideration to further improve the theoretical results. Since the apparent viscosity of the polymer melt is a function of the temperature of the polymer melt and the pressure within the DRU, the viscosity of the polymer melt can be described by

$$\mu = \mu_0 \, e^{-aT + bP}$$

Where b is the viscosity pressure dependency constant. However, with this equation introduced into the analysis, the analysis will become much more complicated than the present one.

10-bit ADC and DAC converters can be used to increase the accuracy of the dieless wire drawing speed control system, and as a result of this, the quality of the produced wire will be improved. To enable a closed-loop control system with good performance to be built using the PRA as a feedback signal, the mechanical components of the PRA transducer require some improvement.

The commissioning stage of the self-tuning PID controller for deterministic systems is only feasible under disturbance condition, and the controller setpoint will be suitably disturbed until the

identification convergence is achieved. The controller parameters will then be fixed until retuning is considered necessary. Here several questions arise as to when this controller can be actually applied. What is the magnitude of the "suitable disturbance" which needs to be added into the input of the controller at the commissioning stage? How much time must elapse (how many time steps of the on line identification algorithm) for the identification to reach convergence? How can we judge a proper set of perameters of the controller has been estimated? To have the solutions of these questions, there is still some work to be done both in theory and practice.

[1] Promotion of Fluid Lubrication in Wire Drawing, Proc. Inst. Mech. Eng., Christopherson, D. G., Naylor, P. B., 1955, p. 169

[2] Lubrication in Wire Drawing, Wear, March, Wistreich, J. G., 1957, pp. 505-511

[3] Hydrostatic Lubrication for Drawing Steel Wire, Tribology in Iron and Steel Works, ISI, Publication, p. 125

[4] Integrated Development and Introduction of New High Speed Mills and Hydrodynamic Lubrication System for Drawing Wires, Steel in the USSR, Vol.10, pp.953-956, 1974, Orlov, S. I., Kolmogror, V. L., Uralskll, V. I., Stukalov, V. T.

[5] A Novel Technique for Wire Drawing, Hashmi, M. S .J., Symmons, G. R. and Parvinmehr, H., J. of Mech. Eng. Sci. Vol. 24, p. 1, 1982

[6] Plasto-hydrodynamic Dieless Wire Drawing: Theoretical Treatment and Experimental Results, Symmons, G. R., Hashmi, M. S. J. and Parvinmehr, H., Proc. Int. Conf. on Developments in Drawing of Metals, Metals Society, London, p. 54, May 1983

[7] Optimisation of Plasto-hydrodynamic System of Wire Drawing Using Polymer Melts , Parvinmehr, H., Ph.D Thesis, Machanical and Production Engineering Department, Sheffield City Polytechnic

[8] A Mathematical Model for the Drawing of a Solid Continuum Through Newtonian Fluid Filled Tubular Orifice, Hashmi, M. S. J. and Symmons, G. R., Proc. 4th Int. Conf. On Mathematical Modelling, Zurich, August 1983.

[9] A Numerical Solution for the Plasto-hydrodynamic Drawing of A Rigid Non-linearly Strain Hardening Continuum Through A Conical Orifice, Hashmi, M. S. J. and Symmons, G. R., 2nd Int. Conf. on Numerical Methods for Non-linear Problems, Spain, April 1984.

[10] Encyclopedia of Polymer Science and Technology, Vol. 8, pp. 587-606 , Interscience Publishers.

[11] Thermoplastics Properties and Design, Ogorkiewicz, R.M. pp. 183-187 , Wiley-Interscience.

[12] Modern Developments in Lubrication Mechanics, J. A. Walowit and J. N. Anno, Applied Science Publishers, pp. 49-59

[13] International Plastics Handbook, Saechtling, Hanser Publishers, p. 353

[14] MC68000 Educational Computer Board Users Manual, Motorola

[15] Motorola Semiconductors Microprocessor Data Manual 1982, pp. 4-335

[16] Data Converters and Reference ICS, Ferranti Semiconductors Technical Handbook, Vol. 10, Farnell Electronic Components Ltd. Issue 2, pp. 2-79

[17] Motorola Semiconductor Products Data

[18] Control Systems, John W. Brewer

[19] Hydraulic and Electro-Hydraulic Servo Systems, R. Walters, London ILIFFE Book Ltd.

[20] Control System Principles and Design, Ernest O., Doebelin, p. 526 , Wiley.

[21] Elements of Computer Process Control With Advanced Control Applications, p. 162, Pradeep B. Deshpande, Raymond H. Ash

[22] A Microprocessor-Controlled High-Accuracy Wide-Range Speed Regulator for Motor Drives, Tsutomu, Ohmae, IEEE Transactions on Industrial Electronics, VOL. IE-29, NO. 3, August 1982

[23] PID Self-tuners: Some Theoretical and Practical Aspects, R. Ortega, R. Kelly, IEEE Transactions on Industrial Electronics Vol. IE-31, No. 4, Nov. 1984.

[24] A Self-tuning Controller With PID Structure, F. Cameron and D. E. Seborg, Int. J. Control, 1983, Vol. 38, No. 2, pp. 401-417.

[25] Report TFRT-7179, Lund Inst. Sweden, Wittenmark, B., 1979.

[26] On Self-tuning Regulators, Astrom, K. J. and Wittenmark, B., Automatica, Vol.9, pp. 185-199, 1973.

[27] Theory and Applications of Self-tuning Regulators, Astrom, K. J., Borisson, U., Ljung, L., and Wittenmark, B., Automatica, Vol. 13, pp. 457-476, 1977.

[28] Self-tuning controller, Clark, D. W., Gawthrop, P., Proc. IEE , Vol. 122, No. 9, pp. 929-934, 1975.

[29] Self-tuning Control, Clark, D. W., and Gawthrop, P., Proc. IEE Vol. 126, No.6, pp. 633-640, 1979.

[30] Pole Assignment Self-tuning Regulator, Wellstead, P. E., Prager, D. L. and Zanker, P.M., Proc. IEE Vol. 126, pp. 781-787, 1979.

[31] Self-tuning Pole-Zero Assignment Regulators, Wellstead, P. E., Edmunds, J. M., et al, Int. J. Control, Vol. 30, No. 1, pp. 1-26, 1979.

[32] Weighted Minimum-variance Self-tuning Control, M. J. Grimble, Int. J. Control, Vol. 36, No.4, pp.597-609, 1982.

[33] Generalised Self-tuning Controller With Pole Assignment, A. Y. Allidina and F. M. Hughes, IEE Proc., Vol. 127, 1980.

[34] Self-tuning Controllers Based On Pole-Zero Placement, K. J. Astrom and B. Wittenmark, IEE Proc., Vol. 127, No.3, 1980.

[35] Self-tuning Controllers for Deterministic Systems, A. Y. Allidina and F. M. Hughes, Int. J. Control, Vol. 37, No.4, pp. 831-841, 1983.

[36] Successful Adaptive Control of Paper Machines, Torsten Cegrell and Torbjorn Hedquist, Automatica, Vol. 11, pp. 53-59, 1975.

[37] Moisture Control of Paper Machine: An Application of A Self-tuning Regulator, Borisson, U. and Wittenmark, B., Lund Report 7337, 1973.

[38] Self-tuning Control of an Ore Crusher, Ulf Borisson and Rolf Syding, Automatica, Vol. 12, pp. 1-7, 1976.

[39] Application of Self-tuning Regulators to the Control Chemical Processes; Digital Computer applications to Process Control, IFAC, 1977, Morris, A. J., Fenton, T. P. and Nazer, Y..

[40] Self-tuning Control Algorithm for Single-chip Microcomputer Implementation, A. L. Dexter, IEE Proc., Vol. 130, No.5, 1983.

[41] Implementation and Application of Microprocessor-Based Self-tuners, Clark, D. W. and Gawthrop, P. J., Automatica, Vol. 17, pp. 233-244, 1981.

[42] A Method for Auto-tuning of PID Control Parameters, Yoshikazu Nishikawa et al, Automatica, Vol. 20, No.3, pp. 321-332, 1984.

[43] Using the Self-tuning Controller to Tune PID Regulators, P. J. Gawthrop, Report No. CE/T/2, University of Sussex, School of Engineering and Applied Sciences.

[44] Digital Computer process Control, Smith, C. L. 1972. Intext Educational.

[45] Computer Control of Industrial Processes, p. 71, S. Bennett and D. A. Linkens, IEE Control Engineering Series 21.

[46] Self-tuning PI and PID Controllers, P. J. Gawthrop, IEEE Conference on Applications of Adaptive and Multivariable Control, Hull, pp. 19-21, July, 1982.

[47] Hybrid Self-tuning Control, P. J. Gawthrop, IEE Proc. Vol. 127, No. 5, Sept. 1980.

[48] Self-tuning PID Controllers: Algorithms and Implementation P. J. Gawthrop, IEEE Transactions on Automatic Control, Vol. AC-31, No. 3, March 1986.

[49] Tuning into the Latest Process Control Electronics, pp. 44-43, Process Engineering, June 1986.

[50] Programmable Controllers Push in Process Control, pp. 35-36, Control and Instrumentation, January 1987.

[51] Practical Issues in the Implementation of Self-tuning Control, Bjorn Wittenmark and K. J. Astrom, Automatica, Vol. 20, NO.5, pp. 595-605, 1984.

[52] An Auto-Pole-Placement Self-tuning Controller,
P. E. Mcdermott, and D. A. Mellichamp, Int. J. Control, Vol. 40,
NO.6, pp. 1131-1147, 1984.

[53] Introduction to Computers and Computer Programming, Samuel
Bergman, Steven Bruckner, 1972 , Addision-Wesley.

Appendix 1 : Viscosity-Temperature Dependence of the Polymer Melt [10]

For many polymers at temperatures considerably above their glass-transition temperature, the melt viscosity has been experimentally determined to be an exponential function of the absolute temperature as given by the Arrhenius-type equation

$$\mu = A \exp (\Delta E / RT) \qquad (A1.1)$$

where μ is the melt viscosity of the polymer

R a constant

A a frequency term

ΔE the energy of activation

A and ΔE must be evaluated empirically for each polymer. In addition to the variation of ΔE and A from polymer to polymer, A and ΔE are also the function of shear rate or shear stress.

The data [11] indicating the temperature dependence of melt viscosity at constant stress and pressure are presented in Fig. 2.11. It can be seen from Fig. 2.11 that, for low density polyethylene, increasing the temperature by 40°C decreases the viscosity at constant stress and constant pressure by a factor of about 3. The melt viscosity-temperature dependence resulting from Fig. 2.11 could be expressed by

$$\mu = \mu_0 e^{-aT} \qquad (A1.2)$$

where "a" is a constant for a given polymer and its value is within the range from 0.02 to 0.1 . μ_0 is the initial viscosity at reference temperature.

However, for many polymers, ΔE (equation A1.1), or "a" (equation A1.2) is a function of temperature. For such polymers, Williams, Landel and Ferry [10] showed that the reduced viscosity data

for a wide variety of polymers could be expressed by the universal WLF equation

$$\mu = \mu_s \exp \left[- \frac{8.86(T - T_s)}{101.6 + T - T_s} \right]$$

where T_s is the reference temperature

μ_s is the viscosity at the reference temperature

The WLF equation is applicable over the temperature range of T_s+50°.

An alternative equation of viscosity-temperature dependence is given by

$$\mu = \mu_{Tg} \exp \left[\frac{-17.44(T - T_g)}{51.6 + T - T_g} \right]$$

where T_g is the glass-transition temperature of the polymer

μ_{Tg} is the melt viscosity at temperature T_g

In general, the WLF equation begins to fail as a valid representation of the viscosity-temperature dependence in the temperature region of T_g+100°, and in some high region the behavior becomes Arrhenius in form.

In theoretical analysis equation (A1.2) is particularly amenable to mathematical treatment.

Appendix 2 : A Integration Formula

The integral region R of the double integral

$$\int_0^{h_1}\int_0^{Y} f(Y')\, dY'dY$$

can be described by inequalities of the form $0 <= Y <= h_1$ and $0 <= Y' <= Y$ (Fig. A2.1). R also can be described by inequalities of the form $0 <= y' <= h_1$ and $Y' <= y <= h_1$. Then we obtain

$$\int_0^{h_1}\int_0^{Y} f(Y')\, dY'dY = \int_0^{h_1}\int_{Y'}^{h_1} f(Y')\, dYdY'$$

$$= \int_0^{h_1} (h_1 - Y')f(Y')\, dY'$$

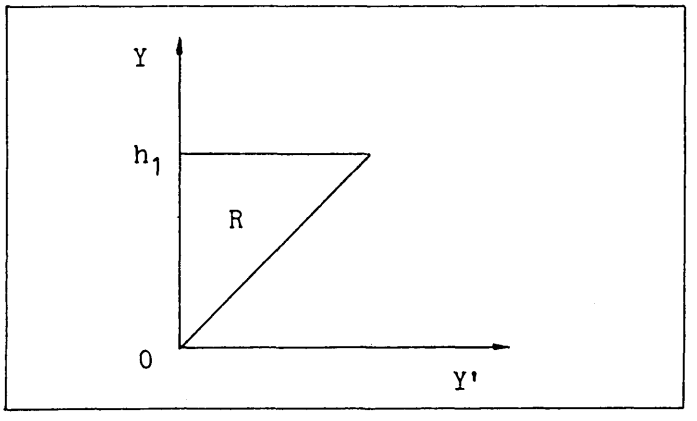


Fig. A2.1

Appendix 3 : Energy Equation Applied to the Process of Dieless Wire Drawing

For incompressible flow, the energy equation is given by [12]

$$\rho c \bar{u} \nabla T = k \nabla^2 T + R \qquad \text{(A3.1)}$$

where ρ is the melt density

c the specific heat

$\bar{u}$ the velocity vector

R the volumetric rate of heat generation

k the thermal conductivity

T the temperature

∇T the temperature gradient

$\nabla^2 T$ the temperature divergence.

Now consider applying the energy equation to process of the dieless wire drawing as sketched in Fig. 2.1a (p. 16). For viscous heating with a Newtonian fluid

$$R = \mu \left(\frac{\partial u}{\partial Y}\right)^2 = \tau \dot{\epsilon} \qquad \text{(A3.2)}$$

where $\dot{\epsilon}$ is the strain rate. In two dimensions we have

$$\bar{u} = u\, i \qquad \text{(A3.3)}$$

(assuming the flow of the polymer melt is axial)

and $$\nabla T = \frac{\partial T}{\partial X} i + \frac{\partial T}{\partial Y} j \qquad \text{(A3.4)}$$

$$\nabla^2 T = \frac{\partial^2 T}{\partial X^2} + \frac{\partial^2 T}{\partial Y^2} \qquad \text{(A3.5)}$$

Substituting for R, u, ∇T and $\nabla^2 T$ from equations (A3.2) to (A3.5) respectively in equation (A3.1) gives

$$\rho c u \frac{\partial T}{\partial X} = k \left[\frac{\partial^2 T}{\partial X^2} + \frac{\partial^2 T}{\partial Y^2} \right] + \mu \left(\frac{\partial u}{\partial Y}\right)^2 \qquad \text{(A3.6)}$$

Let

$$X' = X / L_1$$

$$Y' = Y / h$$

$$u' = u / u_0$$

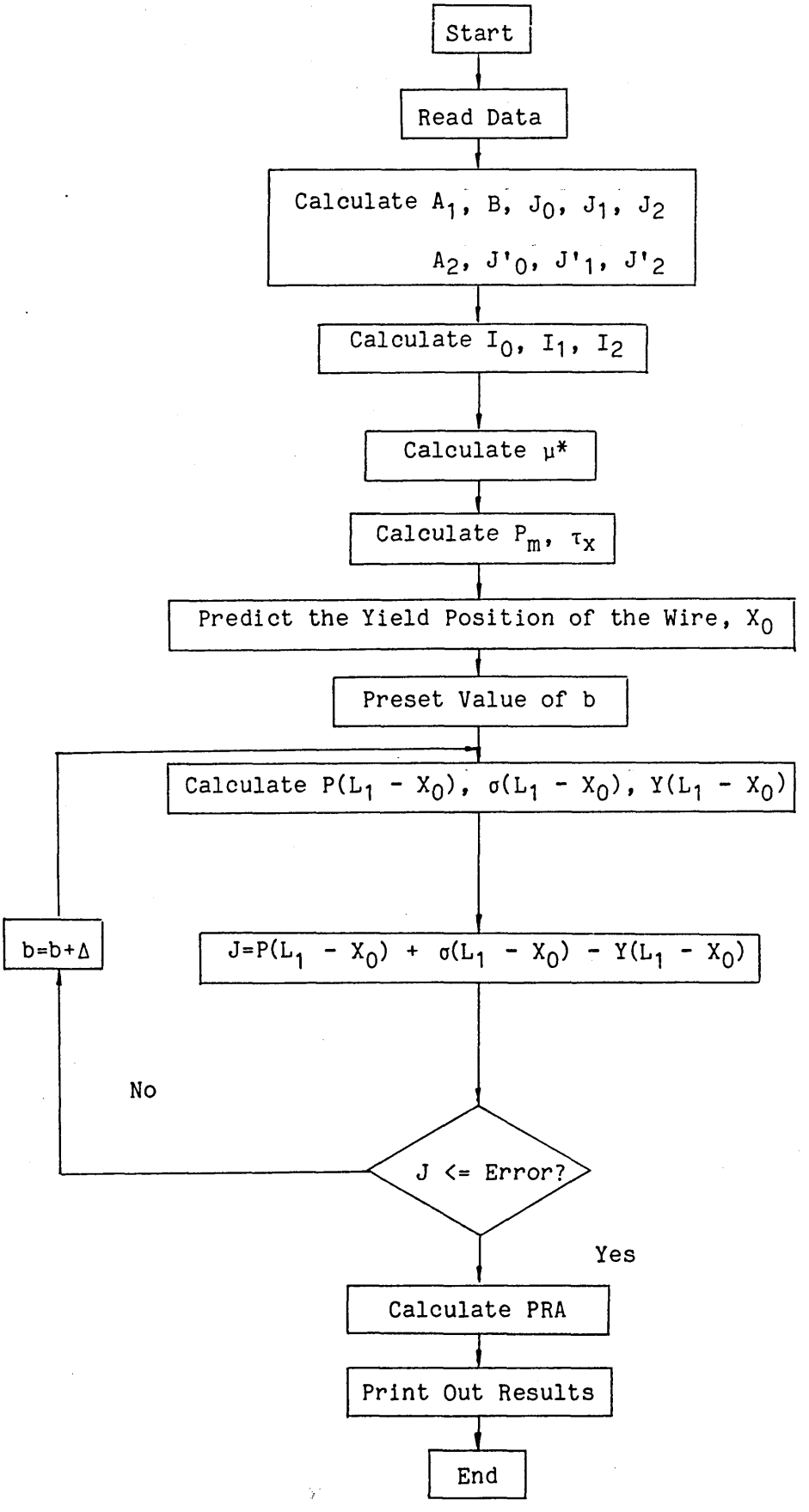
Then equation (A3.6) becomes

$$\frac{\rho c u_0 L_1}{k} \left(\frac{h}{L_1}\right)^2 u' \frac{\partial T}{\partial X'} = \left[\frac{\partial^2 T}{\partial Y'^2} + \left(\frac{h}{L_1}\right)^2 \frac{\partial^2 T}{\partial X'^2}\right] + \frac{\mu u_0^2}{k} \left(\frac{\partial u'}{\partial Y'}\right)^2 \qquad \text{(A3.7)}$$

With $h / L_1 << 1$, the left-hand side of equation and the second term in the brackets on the right-hand side of equation (A3.7) can be neglected. Thus the energy equation applied to the process of dieless wire drawing is given by

$$\frac{\partial^2 T}{\partial Y^2} = -\frac{\mu}{K} \left(\frac{\partial u}{\partial Y}\right)^2 \qquad \text{(A3.8)}$$

Appendix 4 : Flowchart and Listing of the Computer Program for the Theoretical Analysis Using the Stepped Bore Reduction Unit



Listing of the Computer Program for the Theoretical Analysis Using the

Stepped Bore Reduction Unit

```
10  REM program for calculating the reduction in area of the wire
20  REM coupling with the Energy equation
30  REM title W5
40  REM L12 the length of the unit
50  REM XI variable viscosity in the Reynolds equation
60  REM PM maximum pressure at the step
70  REM L1,L2 the lengths of the first and second parts of the unit
80  REM TAU1 shear stress before deformation
90  REM P pressure in deformation zone
100  REM SIGX axial stress in deformation zone
110  REM KK strain hardenig factor of the wire
120  REM RL the ratio of length
130  REM H1 and H2 the gap of unit
140  REM RH the ratio of gap
150  REM D1 initial diameter
160  REM XI0 initial viscosity
170 REM Y0 initial yield stress of the wire
180 REM V  drawing velocity
190 REM N1 the length of the step in simpson numerical integration
200 REM N material constant
210 REM K1 thermal conductivity W/MK
220 REM T1 the maximun temperature within the gap
230 REM K2 viscosity temperture constant XI=XI0*EXP(-K2*T)
240 REM T temperature
250 DIM TT(100)
260 N1=50
270 READ H2,RH,RL,L12,XI0,Y0,D1,N,KK
280 DATA 0.1E-3,1.6,2.5,0.07,110,5.0E+07,1.6E-3,0.52,4.4E+08
290 READ K1,K2,V
300 DATA 0.16,0.08,0.30
310 READ M,M1
320 DATA 1.0E-10,1.56E-10
330 LPRINT "K1";K1;"K2";K2
340 REM H1,L1,L2 computation
350 H1=RH*H2
360 L2=L12/(1+RL)
370 L1=RL*L2
380 LPRINT "H1=";H1;"L1=";L1;"L2=";L2;"H2=";H2;"V=";V;"XI0";XI0
390 REM A,B,J0,J1,J2,JJ0,JJ1,JJ2 computation
400 B1=1+K2*XI0*V*V/(8*K1)
410 E1=K2*SQR(B1*V*V*H1*H1/(2*K2*K1*XI0))
420 E2=LOG(ABS((SQR(B1-1)+SQR(B1))/(SQR(B1-1)-SQR(B1))))
430 J0=E1/E2
440 A1=2*V*V/(K1*K2*XI0*J0*J0)
450 LPRINT "B1";B1;"J0";J0;"A1";A1
460 AA=A1
470 C30=SQR(B1-1)
480 C31=-E2/SQR(A1*B1)/K2
490 J1=-C31*4*C30/K2/XI0/SQR(A1)
500 GOSUB 1140
510 C32=(C31^2)*4*C30/K2/XI0/SQR(A1)
520 J2=C6+C32
```

Listing of the Computer Program for the Theoretical Analysis Using the

Stepped Bore Reduction Unit

```
530 E3=K2*SQR(B1*V*V*H2*H2/(2*K2*K1*XIØ))
540 JJØ=E3/E2
550 A2=2*V*V/(K1*K2*XIØ*JJØ*JJØ)
560 AA=A2
570 C33=-E2/SQR(A2*B1)/K2
580 GOSUB 1140
590 JJ1=-C33*4*C3Ø/K2/XIØ/SQR(A2)
600 C34=(C33^2)*4*C3Ø/K2/XIØ/SQR(A2)
610 JJ2=C6+C34
620 REM I1, I2, I3 computation
630 IØ=JØ/H1
640 I1=J1/H1^2
650 I2=J2/H1^3
660 REM XI computation
670 XI=-I1/6/(I1^2-I2*IØ)
680 EFF=XI*V
690 PRINT "XI",XI,"T1",T1
700 LPRINT "JØ";JØ;"J1";J1;"J2";J2
710 LPRINT "JJØ";JJØ;"JJ1";JJ1;"JJ2";JJ2
720 LPRINT "IØ";IØ;"I1";I1;"I2";I2;"XI";XI;"T1";T1;"EFF";EFF
730 REM PM,TAUX computation
740 PM1=V*L1*L2*(J1*JJØ-JJ1*JØ)
741 PM=PM1/(-L1*JØ*(JJ1^2-JJ2*JJØ)-L2*JJØ*(J1^2-J2*JØ))
750 LPRINT "PM";PM
760 TAU1=-(V+PM*J1/L1)/JØ
770 LPRINT "TAU1";TAU1
780 REM deformation starts point computation
790 X1=YØ/(PM/L1-4*TAU1/D1)
800 LPRINT "X1";X1
810 REM PRA computation
820 X=L1-X1
830 B=4.Ø74542E-Ø2
840 A=SQR(H1/B)
850 ZZ1=6*XI*V*((ATN(X/A))
851 Z1=ZZ1/(8*A^3*B^2)+(X^3-A^2*X)/(8*A^2*B^2*(A^2+X^2)^2))
860 ZZ2=PM/L1*(X1+(3*A*ATN(X/A))
861 Z2=ZZ2/8+(5*(A^4)*X+3*(A^2)*X^3)/(8*(A^2+X^2)^2))
870 P=Z1+Z2
880 Z3=2*YØ*LOG(D1/(D1-2*B*X^2))
890 Z4=LOG((SQR(D1)+X*SQR(2*B))/(SQR(D1)-SQR(2*B)*X))
900 Z5=(5*D1-2*B*A^2)/(2*A*(D1+2*B*A^2)^2)*ATN(X/A)
910 Z6=3*X/(2*(D1+2*B*A^2)*(A^2+X^2))
920 ZZ7=(4*XI*V)/B*((2*D1+B*A^2)*2*B
921 Z7=ZZ7/(SQR(2*B*D1)*((D1+2*B*A^2)^2))*Z4+Z5-Z6)
930 Z8=A*(D1+6*B*A^2)/(2*(D1+2*B*A^2)^2)*(ATN(X/A))
940 Z9=SQR(D1)-SQR(2*B)*X
950 ZZ1Ø=LOG((SQR(D1)+X*SQR(2*B))/Z9)
951 Z1Ø=(4*A^4*B^2)/(SQR(2*B*D1)*(D1+2*B*A^2)^2)*ZZ1Ø
960 Z11=(2*PM*H1)/L1*((A^2)*X/(2*(D1+2*B*A^2)*(A^2+X^2))+Z8+Z1Ø)
970 ZZ12=KK/(N+1)
971 Z12=ZZ12*((2*LOG(D1/(D1-2*B*X^2)))^(N+1))+4*X1*ABS(TAU1)/D1
980 SIGX=Z3+Z7+Z11+Z12
990 C= YØ+KK*(2*LOG(D1/(D1-2*B*X^2)))^N
```

Listing of the Computer Program for the Theoretical Analysis Using the

Stepped Bore Reduction Unit

```
1000 J=P+ABS(SIGX)-C
1010 PRINT "J";J;
1020 IF ABS(J)<=2000 GOTO 1060
1030 B=B+M*J
1040 PRINT "B";B
1050 GOTO 840
1060 IF ABS(J)<100 GOTO 1100
1070 B=B+M1*J
1080 PRINT "B";B
1090 GOTO 840
1100 PRA=((D1-2*B*X^2)^2)/(D1^2)
1110 L=1-PRA
1120 LPRINT "PRA";L;"B";B;"J";J;"SIGX";SIGX;"P";P
1130 END
1140 REM J2,JJ2 calculate subroutine
1150 T1=LOG(B1)/K2
1160 DT=T1/N1
1170 FOR I=0 TO N1
1180 IF I=N1 GOTO 1210
1190 T=DT*I
1200 GOTO 1220
1210 T=DT*I-.01
1220 C9=SQR(B1-EXP(K2*T))
1230 C1=((LOG(ABS((C9-SQR(B1))/(C9+SQR(B1)))))/SQR(AA*B1)/K2)^2
1240 C2=XI0*SQR(AA)*C9
1250 TT(I)=EXP(K2*T)*C1/C2
1260 NEXT I
1270 C4=0
1280 C5=0
1290 FOR I=1 TO N1/2
1300 C4=C4+TT(2*I-1)
1310 NEXT I
1320 FOR I=1 TO N1/2-1
1330 C5=C5+TT(2*I)
1340 NEXT I
1350 C6=2*DT/3*(TT(0)+TT(N1)+4*C4+2*C5)
1360 RETURN
```

Appendix 5 : Program for Drawing the Wire of a Uniform Diameter

Linearisation consists of approximating a curve by a finite number of straight line segments. Values at the six break points (see Fig. 3.31), tangent values and the equation for each linear segment are given in Table A5.1.

Table A5.1

Point	PRA	V (ms^{-1})
a	2.5	0.137
b	7.5	0.259
c	12.5	0.407
d	15.0	0.506
e	17.0	0.632
f	18.0	0.8

Segment	Tangent Value	Equation V=A+(PRA-B)C
ab	1/0.0244	0.137+(PRA-2.5)*0.0244
bc	1/0.0296	0.259+(PRA-7.5)*0.0296
cd	1/0.0396	0.407+(PRA-12.5)*0.0396
de	1/0.063	0.506+(PRA-15)*0.063
ef	1/0.168	0.632+(PRA-17)*0.168

```
* MC6821 I/O interface addressing on MC68000 microcomputer board
* PRA peripheral register A
* PRB peripheral register B
* DDRA data direction register A
* DDRB data direction register B
* CRA control register A
* CRB control register B
* MC68230 PI/T addressing
* CPRH counter preload register high
* CPRM counter preload register middle
* CPRL counter preload register low
* TCR timer control register
* TIVR timer interrupt vector register
* TSR timer status register
* SAIHR at this location in RAM, the starting address of the interrupt
* handler routine is stored

u1      DS.W      1    ; control output u(t-1)
Y0-2    DS.W      3    ; process output y(t) . . y(t-2)
```

```
U1      DS.W      1    ; integrator output U(t-1)
h0-1    DS.W      2    ; controller parameters
g0-2    DS.W      3    ; controller parameters
DDRA1   EQU       $3FFF9
CRA1    EQU       $3FFFB
DDRB1   EQU       $3FFFD
CRB1    EQU       $3FFFF
PRA1    EQU       $3FFF9
PRB1    EQU       $3FFFD
SAIHR   EQU       $6C
TCR     EQU       $10021
TIVR    EQU       $10023
DDRB2   EQU       $3FFED
CRB2    EQU       $3FFEF
PRB2    EQU       $3FFED
CPRH    EQU       $10027
CPRM    EQU       $10029
CPRL    EQU       $1002B
TSR     EQU       $10035
1LSB    DS.W      1      ; the value of the 1LSB
PRA     DS.W      1      ; the value of the PRA
POLY    DS.W      1      ; polymer number
STRI    DW               ; prompt string in ASCII characters


* Look-up table

BASE0-15 DW 2.5, 7.5, 12.5, 15.0, 17.0, 18.0, 0.137, 0.259, 0.407,
            0.506, 0.632, 0.0244, 0.0296, 0.0396, 0.0630,0.168



* Program for prompt on the screen. The string in memory for prompt is
* in ASCII characters. 227 is the string output routine number. A5 is
* used to point the begining address $STRI of the string. A6 points the
* address $FINA immediately  following the last byte of the string.


PRO:  MOVE.W  #$STRI,  A5
      MOVE.W  #$FINA,  A6
      MOVE.B  #227,    D7
      TRAP    #14
      TRAP    #1          ; Output string on screen and return control
                          ; of system firmware


* Polymer number 1, 2 and 3 corresponding to WVG 23, ELVAX 650 and
* NYLON 6 respectively, the value of the required PRA is of three
* characters in BCD, with 1 character fraction and. Input PRA value and
* polymer number in locations $PRA and $POLY respectively via keyboard,
* then input command: " GO $TEM " to send temperature setpoint out.
```

```
        ANDI.W  #$F00,   D1
        MULU    #$A,     D1
        ADD.W   D1,      D3

* Set drawing speed
* Exit-D3, speed setpoint, 8-bit integer


        MOVE.L  $BASE1, A1
CM:     CMP.W   (A1)+,  D3
        BLE     CM1
        BRA     CM
CM1:    SUBA    #4,     A1
        MOVE.W  (A1),   D0     ; D0=B
        ADDA    #$C,    A1
        MOVE.W  (A1),   D1     ; D1=A
        ADDA    #$A,    A1
        MOVE.W  (A1),   D2     ; D2=C
        SUB.W   D0,     D3     ; PRA-B
        MULM.W  D2,     D3     ; (PRA-B)C
        LSL.L   #8,     D1
        LSL.L   #8,     D1
        ADD.L   D1,     D3     ; A+(PRA-B)C
        DIVU    $1LSB,  D3     ; V=(A+(PRA-B)C)/1LSB
        LSR.W   #8,     D3

* Enable interrupt

        MOVE.B #0,      CPRH
        MOVE.B #4,      CPRM
        MOVE.B #$E2,    CPRL   ; set 10 ms interrupt interval
        MOVE.B #$1B,    TIVR   ; interrupt vector
        MOVE.B #$TA,    SAIHR  ; interrupt handler routine starting address
        MOVE.B #$A1,    TCR    ; enable interrupt
COM2:   NOP
        BRA     COM2


* Interrupt handler routine
* PID control


TA:     MOVE.B  #$36,    CRA1
        NOP
        NOP
        MOVE.B  #$3E,    CRA1
TA1:    BTST    #7,      CRA1
        BEQ     TA1
        MOVE.B  $PRA1,   $Y0   ; sample data
        MOVE.B  #1,      TSR   ; enable  next interrupt
        BSR     PID            ; call PID algorithm
        RTE
```

```
* PID algorithm subroutine
* incremental output u(t) = [ RW(t)-GY(t) ]/H
* Entry: D3 - W(t), setpoint. $Y0 -  process output
* Exit: result u(t) in D1


PID:  MOVE.L   #0,    D1
      LEA      $g0,   A1
      MOVE.L   #2,    D2
PID1: MOVE.W   (A1)+, D0
      MULS.W   D3,    D0
      ADD.L    D0,    D1
      DBF.L    D2,    PID1 ; (g0+g1+g2)W(t)
      LEA      $h1,   A1
      MOVE.W   (A1)+, D0
      MULS.W   $u1,   D0
      SUB.L    D0,    D1    ; RW(t)-h1u(t-1)
      MOVE.W   (A1)+, D0
      MULS.W   $Y0,   D0    ; g0Y(t)
      SUB.L    D0,    D1
      MOVE.W   (A1)+, D0
      MULS.W   $Y1,   D0    ; g1Y(t-1)
      SUB.L    D0,    D1
      MOVE.W   (A1)+, D0
      MULS.W   $Y2,   0     ; g2Y(t-2)
      SUB.L    D0,    D1    ; RW(t)-GY(t)-h1u(t-1)
      DIVS     $h0,   D1    ; u(t) in D1

* Move data for next step computation

      MOVE.W  D1,    $u1
      MOVE.W  $Y1,   $Y2
      MOVE.W  $Y0,   $Y1

* Integrator, n4 is the upper limit of the integrator, n5 is the lower
* limit of the integrator.

      ADD.W   $U1,   D1     ; integrate
      MOVE.W  D1,    D2
      CMP.W   #$n4,  D1
      BGE     PID2
      CMP.W   #$n5,  D1
      BLE     PID3
      BRA     PID4
PID2: MOVE.W  #$n4,  D1     ; integrator limitation
      BRA     PID4
PID3: MOVE.W  #$n5,  D1     ; integrator limitation
PID4: MOVE.W  D1,    $U1
      CMPI.W  #$FF,  D2
      BLE     PID5
      MOVE.W  #$FF,  D2
      BRA     PID6
PID5: CMPI.W  #0,    D2
      BLE     PID7
PID6: MOVE.B  D2,    PRB1   ; output data
      RTS
PID7: MOVE.B  #0,    PRB1   ; output data
      RTS
```

Appendix 6 : Program for Drawing Tapered Wire

```
PR1     DS.W    1  ; value of the PIA1*1000 with 3 bit integer in BCD
PR2     DS.W    1  ; value of the PIA2*1000 with 3 bit integer in BCD
L       DS.W    1  ; the length of the wire ( cm )
N2      DS.W    1  ; the final speed step
C       DS.W    1  ; constant c = L / ((N2-N1) 1LSBT)
NUM1    DS.W    1  ; PID computation period


* Subroutine
* BCD - binary conversion
* Entry: D1 bit0-15 -  number in BCD
* Exit : D3 bit0-31 number in binary

BBR:  CLR.L   D3
      CLR.L   D4
      MOVEQ   #1,    D2
BBR1: CMPI.W  #10000,D2
      BGE.S   $EXIT
      MOVE.W  D1,    D4
      ANDI.W  #$F,   D4
      MULU    D2,    D4
      ADD.W   D4,    D3
      LSR.W   #4,    D1
      MULU    #10,   D2
      BRA.S   BBR1
EXIT: RTS


* Calculate starting speed step N1, final speed step N2, and c
* Entry: $PR1  - the value of PRA1
         $PR2  - the value of PRA2
         $L    - the length of the wire
* Exit:  D6  - N1
         $N2 - N2
         $C  - c

TC:   MOVE.W  $PR1,    D1
      BSR     $BBR
      SUBI.W  #45,     D3    ; PRA1-0.045*1000
      MULU    #10000,  D3
      DIVU    #8078,   D3    ; 0.1584*0.0051=0.0008078
      MOVE.W  D3,      D6    ; D6 = N1
      MOVE.W  PR2,     D1
      BSR     BBR
      SUBI.W  #45,     D3    ; PRA2-0.045*1000
      MULU    #10000,  D3
      DIVU    #8078,   D3
      MOVE.W  D3,      $N2   ; ( $N2 ) = N2
      MOVE.W  $L,      D1
      BSR     BBR
      MULU    #10000,  D3
      LSL.L   #4,      D3
      MOVE.W  $N2,     D5    ; D3=L*16000
      SUB.W   D6,      D5    ; D5=N2 - N1
      MULU    #$1052,  D5    ; 1LSB*T*100000*16 = 16.32 = $1052
```

```
      LSR.L   #8,       D5     ; T=2 ms
      DIVU    D5,       D3     ; 1LSB = 0.0051
      MOVE.W  D3,       $C

* Initialization of portA, portB, timer, USS and USP

CON:  MOVE.L  #$7000,   A7     ; set USS
      MOVE.L  #$A00,    A1
      MOVE.L  A1,       USP    ; set USP
      MOVE.W  #$2000,   SR
      MOVE.W  #$50,     $h0
      MOVE.W  #$30,     $h1
      MOVE.W  #$155,    $g0
      MOVE.W  #$FEA0,   $g1
      MOVE.W  #$20,     $g2    ; set PID controller parameters
      MOVE.B  #0,       CRB1
      MOVE.B  #$FF,     DDRB1
      MOVE.B  #4,       CRB1   ; PIA1 port B output
      MOVE.B  #0,       CRB2
      MOVE.B  #$FF,     DDRB2
      MOVE.B  #4,       CRB2   ; PIA2 port B output
      MOVE.B  #0,       CRA1
      MOVE.B  #0,       DDRA1
      MOVE.B  #4,       CRA1   ; PIA1 port A input
      CLR.W   $Y1
      CLR.W   $Y2
      CLR.W   $U1
      MOVE.W  #5,       D4     ; PID computation period, 5 times of interrupt
                               ; interval,interrupt interval is 2 ms
      CLR.L   D5
      MOVE.W  #5,       $NUM1
      MOVE.W  $C,       D5
      DIVU    D6,       D5     ; n1 = c/N1
      MOVE.B  #0,       CPRH
      MOVE.B  #0,       CPRM
      MOVE.B  #$FA,     CPRL   ; set 2 ms interrupt interval
      MOVE.B  #$1B,     TIVR   ; interrupt vector
      MOVE.B  #$TA,     SAIHR  ; interrupt handler routine starting address
      MOVE.B  #$A1,     TCR    ; enable interrupt
COM2: NOP
      BRA     COM2

* Interrupt handler routine

TA:   MOVE.B  #1,       TSR
      CMP.W   $NUM1,    D4     ; PID computation
      BEQ     TA1
      SUBI    #1,       D4     ; no, D4=D4-1
      BRA.L   TA3
```

```
TA1:  MOVE.W  D6,      D3     ; PID computation
      MOVE,B  #$36,    CRA1
      NOP
      NOP
      MOVE.B  #$3E,    CRA1
TA2:  BTST    #7,      CRA1
      BEQ     TA2
      MOVE.B  PRA1,    $Y0    ; sample data
      BSR     PID              ; PID action
TA3:  CMP.W   #0,      D4
      BEQ     TA4
      BRA     TA5
TA4:  MOVE.W  NUM1,    D4     ; set counter D4, PID conputation period
TA5:  SUBI    #1,      D5     ; D5-1=D5
      CMP.W   #0,      D5     ; increase drawing speed step
      BEQ     TA7
      BRA     TA8
TA7:  ADDI    #1,      D6     ; yes, increase speed step
      MOVE.L  #0,      D5
      MOVE.W  #$C,     D5
      DIVU    D6,      D5     ; calculate n at speed step N
      CMP.W   $N2,     D6
      BGT     TA9              ; final speed step
TA8:  RTE                      ; no, return to continue
TA9:  MOVE.B  #$1B,    TIVR
      MOVE.L  #$ST,    $6C
      RTE                      ; set new interrupt handler routine starting
                               ; address for slow stop


* Interrupt handler routine
* Slow stop

ST:   MOVE.B  #1,      TSR
      CMP.W   #0,      D6
      BEQ     ST1
      SUBI.W  #1,      D6
ST1:  MOVE.B  D6,      PRB1
      RTE
```

Appendix 7 : Program Listing for the PRA Indicator

```
* MC68000 program for the PRA Indicator
* Initialization
N1    DS.W    1    ; Buffer for the PRA in Hex
N2    DS.W    1    ; Buffer for the PRA in Decimal


INI:    MOVE.L  #$7000,  A7
        MOVE.L  #$A00,   A1
        MOVE.L  A1,      USP
        MOVE.W  #$2000,  SR
        MOVE.B  #0,      CPRH
        MOVE.B  #9,      CPRM
        MOVE.B  #$C4,    CPRL ; Set 20 ms interrupt interval
        MOVE.B  #$1B,    TIVR
        MOVE.L  #$IND,   $6C
        MOVE.W  #0,      $N1
        MOVE.W  #1,      D5    ; Set counter 1 and 2 initial
        MOVE.W  #0,      D6    ; values
        MOVE.B  #$A1,    TCR   ; Interrupt enable
INI1:   NOP
        JMP     INI1


* Interrupt Handler Routine

IND:    MOVE.B   #1,      TSR  ; Enable next interrupt
        CMPI.W   #0,      D5
        BEQ      IND1
        MOVE.W   #0,      D5
        MOVE.B   #$5B,    $30003
        MOVE,B   #$5A,    $30001; Set timer 1 and 2 mode
        MOVE.B   #$FF,    $30005
        MOVE.B   #$FF,    $30007
        MOVE.B   #$FF,    $30009
        MOVE.B   #$FF,    $3000B; Enable counter 1 and 2
        RTE


IND1:   MOVE.W    #1,      D5
        MOVE.L    #0,      D3
        MOVE.L    #0,      D4
        MOVE.B    $30005,  D4
        ASL.W     #8,      D4
        MOVE.B    $30007,  D4   ; Read counting value from timer 1
        MOVE.B    $30009,  D3
        ASL.W     #8,      D3
        MOVE.B    $3000B,  D3   ; Read counting value from timer 2
        NEG.W     D4
        NEG.W     D3
```

```
        SUB.W     D4,      D3
        NEG.W     D3
        MULU.W    #1000,   D3    ; ( V1-V2 )*1000/V1
        DIVU      D4,      D3
        MOVE.W    $N1      D4
        ADD.W     D4,      D3
        ASR.W     #1,      D3    ; Calculate the PRA from the reading
        MOVE.W    D3,      $N1   ; The PRA in D3
        ADD.W     #1,      D6
        CMPI.W    #10,     D6
        BEQ       IND2
        RTE
IND2:   MOVE.W    #0,      D6
        MOVE.W    #$N1,    D0
        MOVE.W    #$N2,    A6
        MOVE.B    #236,    D7
        TRAP      #14            ; Hex-Decimal conversion
        MOVE.W    #$N2,    A5
        MOVE.W    #$N2+#8,A6
        MOVE.B    #227,    D7
        TRAP      #14            ; Send sampled data to VDU
        RTE
```

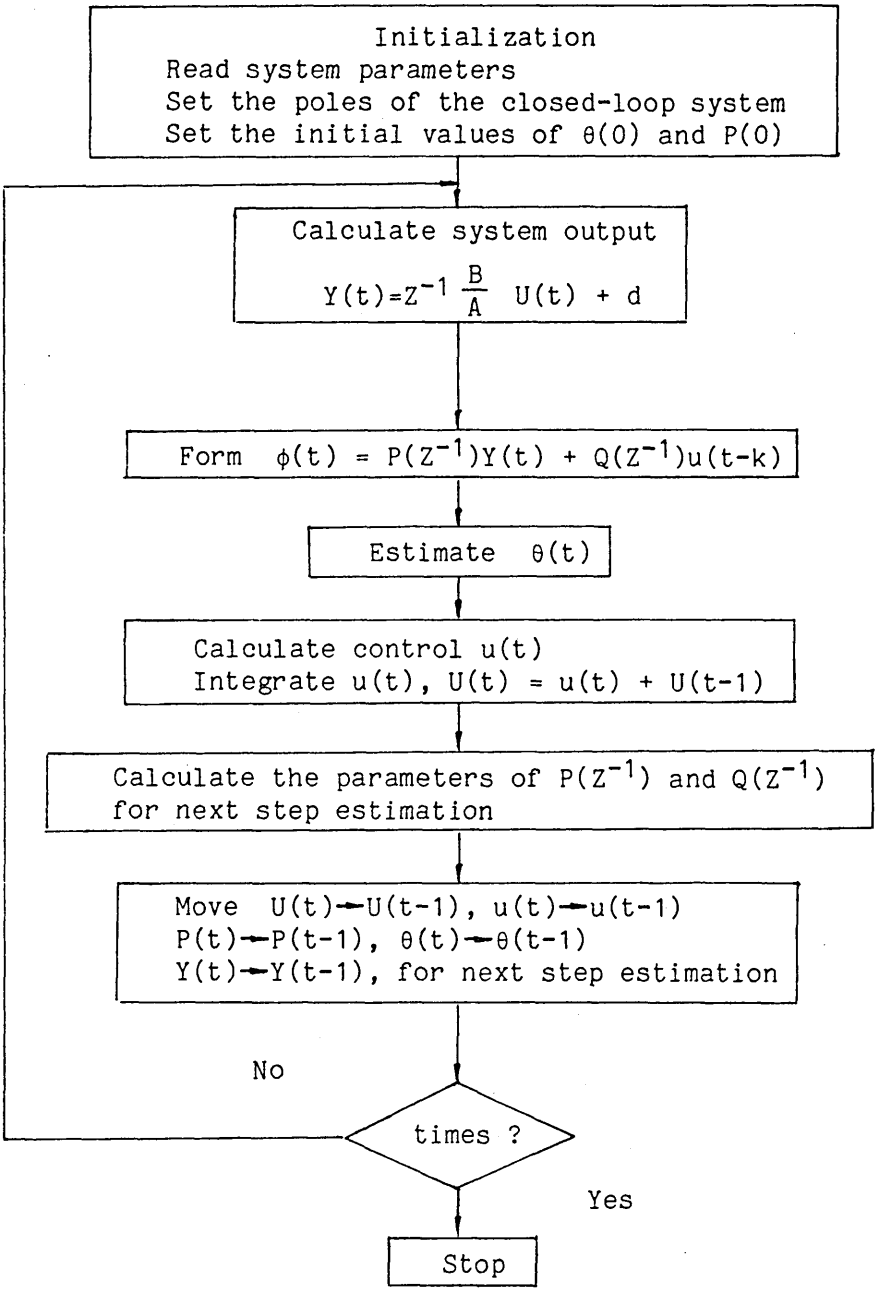
Appendix 8 : Program for the Closed-loop Control With the PRA Transducer

```
* MC68000  Closed-loop control program

N1-N1+4 DS.W  5 ; Data buffer
N2      DW        Setpoint with 3 bit integer

* Initialization

INI:  MOVE.L   #$7000,  A7
      MOVE.L   #$A00,   A1
      MOVE.L   A1,      USP
      MOVE.W   #$2000,  SR
      MOVE.B   #0,      CRB1
      MOVE.B   #$FF,    DDRB1
      MOVE.B   #4,      CRB1 ; set port B output
      MOVE.W   #4,      D1
      LEA      #$N1,    A1
INI1: MOVE.W   #0,      (A1)+
      DBF.W    D1,      INI1
INI2: MOVE.B   #$5B,    $30003
      MOVE,B   #$5A,    $30001; Set timer#1 and #2 mode
      MOVE.B   #$FF,    $30005
      MOVE.B   #$FF,    $30007
      MOVE.B   #$FF,    $30009
      MOVE.B   #$FF,    $3000B; Enable counter#1 and #2
      MOVE.W   #$950,   D1
INI3: SUB.W    #1,      D1
      CMPI.W   #0,      D1
      BNE      INI3           ; 20 ms time delay
      MOVE.L   #0,      D3
      MOVE.L   #0,      D4
      MOVE.B   $30005,  D4
      ASL.W    #8,      D4
      MOVE.B   $30007,  D4    ; Read counting value from
      MOVE.B   $30009,  D3    ; timer#1
      ASL.W    #8,      D3
      MOVE.B   $3000B,  D3    ; Read counting value from
      NEG.W    D4             ;  timer#2
      NEG.W    D3
      SUB.W    D4,      D3
      NEG.W    D3
      MULU.W   #1000,   D3
      DIVU     D4,      D3    ; the measured PRA in D3
      MOVE.W   $N1+1,   $N1
      MOVE.W   $N1+2,   $N1+1
      MOVE.W   $N1+3,   $N1+2
      MOVE.W   $N1+4,   $N1+3
      MOVE.W   D3,      $N1+4 ; move data for next time
      LEA      #N1,     A1    ; computation
      MOVE.W   #3,      D1
```

```
INI4: ADD.W     (A1)+, D3
      ASR.W     #1,    D3
      DBF.W     D1,    $INI4 ; average the last 5 sampled data
      MOVE.W    $N2,   D1    ; D1=setpoint
      SUB.W     D3,    D1
      MULS.W    #P,    D1    ; propotional control
      CMP.W     #$FF,  D1
      BLE       INI5
      MOVE.W    #$FF,  D1
      BRA       INI6
INI5: CMP.W     #0,    D1
      BLE       INI7
INI6: MOVE.B    D1,    $3FFFD; output
      BRA       INI2
INI7: MOVE.W    #0,    $3FFFD; output
      BRA       INI2
```

Appendix 9 : Flowchart of Digital Simulation Program and Computer Listing for the Self-tuning PID Controller

Flowchart of the Digital Simulation Program



```
1Ø REM PID-STC with pole assignment for deterministic system
2Ø REM key words PID-STC
3Ø REM W(Ø) setpoint
4Ø REM T(1),T(2) time constants for pole assignment
5Ø REM NSS the number of parameters to be estimated
6Ø REM K-time delay
7Ø REM NAA,NB degree of system polynomials A,B
8Ø REM NA=NAA+1 introducing integrator
9Ø REM K1-factor to discount past data
1ØØ REM NA=deg A+1, NG=deg G, NH=deg H, NQ=deg Q
11Ø REM Q(I)= (t), Q1(I)= (t-1), P(I,J)=P(t), P1(I,J)=P(t-1)
12Ø REM DU(I)=u(t), U(I)=U(t), PP(I)=P(z ), QQ(I)=Q(z )
13Ø REM Y(t) system output
14Ø REM N,N1 counter,NN,NN1 preset value of counter
15Ø REM D constant offset
16Ø REM initialization
17Ø READ K,NA,NB,K1,T(1),T(2),NAA
18Ø READ PP(Ø),D
19Ø READ N,NN,N1,NN1
2ØØ NP=NA-1
21Ø NG=NA-1
22Ø NH=NB+K-1
23Ø NQ=NB-1
24Ø NSS=NH+NG+2
25Ø REM Initialize matrix P1(I.J),Q1(I)
26Ø FOR I=1 TO NSS
27Ø LET P1(I,I)=32767
28Ø NEXT I
29Ø FOR I=1 TO NSS
3ØØ READ Q1(I)
31Ø NEXT I
32Ø INPUT "W=";W(Ø)
33Ø REM parameters of system to be simulated
34Ø FOR I=1 TO NAA
35Ø READ AØ(I)
36Ø NEXT I
37Ø FOR I=Ø TO NB
38Ø READ BØ(I)
39Ø NEXT I
4ØØ DATA 1,3,1,1,-1.21,Ø.6Ø4,2
41Ø DATA 1,1
42Ø DATA 1,21,1,261
43Ø DATA Ø.5,Ø.36,Ø.2,-Ø.4,Ø.1
44Ø DATA -1.628,Ø.657,1.57,1.36
45Ø REM y1(t) system output calculation
46Ø Y1(Ø)=Ø
47Ø FOR I=1 TO NAA
48Ø Y1(Ø)=Y1(Ø)-AØ(I)*Y1(I)
49Ø NEXT I
5ØØ FOR I=K TO NB+K
51Ø Y1(Ø)=Y1(Ø)+BØ(I-K)*U(I)
52Ø NEXT I
```

```
530 Y1(Ø)=Y1(Ø)+(1+AØ(1)+AØ(2))*D
540 REM y(t) calculation
550 Y(Ø)=Y1(Ø)
560 REM move y1(t) for next step calculation
570 FOR I=Ø TO NAA-1
580 Y1(NAA-I)=Y1(NAA-I-1)
590 NEXT I
600 REM FA(t)=PP*Y(t)+QQ*DU(t-k) calculation
610 FA=Ø
620 FOR I=Ø TO NP
630 FA=FA+PP(I)*Y(I)
640 NEXT I
650 FOR I=Ø TO NQ
660 FA=FA+QQ(I)*DU(I+K)
670 NEXT I
680 LPRINT "N1";N1;"N";N;"FA";FA;
690 REM recursive estimation FA=HH*DU(t-k)+GG*Y(t k)
700  REM FORM (DU(t-k),,DU(t-k-nh),Y(t-K),,,Y(t-k-ng))
710  FOR I=1 TO NH+1
720  Z1(1,I)=DU(I+K-1)
730  NEXT I
740  FOR I=NH+2 TO NH+NG+2
750  Z1(1,I)=Y(I-NH-2+K)
760  NEXT I
770  FOR I=1 TO NSS
780  Z(I,1)=Z1(1,I)
790  NEXT I
800  REM P(t) calculation
810  FOR I=1 TO NSS
820  C(1,I)=Ø
830  NEXT I
840  FOR I=1 TO NSS
850  FOR J=1 TO NSS
860  C(1,I)=C(1,I)+Z1(1,J)*P1(J,I)
870  NEXT J
880  NEXT I
890  X=Ø
900  FOR I=1 TO NSS
910  X=X+C(1,I)*Z(I,1)
920  NEXT I
930  X=X+1
940  FOR I=1 TO NSS
950  C(I,1)=Ø
960  NEXT I
970  FOR I=1 TO NSS
980  FOR J=1 TO NSS
990  C(I,1)=C(I,1)+P1(I,J)*Z(J,1)
1000  NEXT J
1010  NEXT I
1020  FOR I=1 TO NSS
1030  FOR J=1 TO NSS
1040  H(I,J)=Ø
1050  NEXT J
```

```
1060  NEXT I
1070  FOR I=1 TO NSS
1080  FOR J=1 TO NSS
1090  H(I,J)=H(I,J)+C(I,1)*Z1(1,J)
1100  NEXT J
1110  NEXT I
1120  FOR I=1 TO NSS
1130  FOR J=1 TO NSS
1140  V(I,J)=0
1150  NEXT J
1160  NEXT I
1170  FOR I=1 TO NSS
1180  FOR J=1 TO NSS
1190  FOR L=1 TO NSS
1200  V(I,J)=V(I,J)+H(I,L)*P1(L,J)
1210  NEXT L
1220  NEXT J
1230  NEXT I
1240  FOR I=1 TO NSS
1250  FOR J=1 TO NSS
1260  H(I,J)=P1(I,J)-V(I,J)/X
1270  NEXT J
1280  NEXT I
1290  FOR I=1 TO NSS
1300  FOR J=1 TO NSS
1310  P(I,J)=H(I,J)/K1
1320  NEXT J
1330  NEXT I
1340  REM Q(t)=(HH(0),,HH(nh),GG(0),,GG(ng))
1350  REM Q(I)=(Q(1),,,Q(nh+1),,,,,,,,Q(nh+ng+2),,,Q(nh+ng+ne+3))
1360  FOR I=1 TO NSS
1370  M(I)=0
1380  NEXT I
1390  FOR I=1 TO NSS
1400  FOR J=1 TO NSS
1410  M(I)=M(I)+P(I,J)*Z(J,1)
1420  NEXT J
1430  NEXT I
1440  X=0
1450  FOR I=1 TO NSS
1460 X=X+Z1(1,I)*Q1(I)
1470 NEXT I
1480 X=X-FA
1490 FOR I=1 TO NSS
1500 M(I)=M(I)*X
1510 NEXT I
1520 FOR I=1 TO NSS
1530 Q(I)=Q1(I)-M(I)
1540 NEXT I
1550 LPRINT "Q" ;
1560 FOR I=1 TO NSS
1570 LPRINT Q(I);
1580 NEXT I
```

```
1590 LPRINT
1600 REM move P(t) for next step calculation
1610 FOR I=1 TO NSS
1620 FOR J=1 TO NSS
1630 P1(I,J)=P(I,J)
1640 NEXT J
1650 NEXT I
1660 REM input W(Ø)
1670 N=N+1
1680 IF N=NN GOTO 1700
1690 GOTO 1720
1700 N=Ø
1710 INPUT "W=";W(Ø)
1720 REM apply control H*DU(t)+G*Y(t)+E*W(t)=Ø
1730 U1(Ø)=Ø
1740 FOR I=1 TO NH
1750 U1(Ø)=U1(Ø)+Q(I+1)*DU(I)
1760 NEXT I
1770 FOR I=NH+1 TO NH+NG+1
1780 U1(Ø)=U1(Ø)+Q(I+1)*Y(I-NH-1)
1790 NEXT I
1800 U1(Ø)=(Q(3)+Q(4)+Q(5))*W(Ø)-U1(Ø)
1810 DU(Ø)=U1(Ø)/Q(1)
1820 U(Ø)=DU(Ø)+U(1)
1830 LPRINT "U";U(Ø);"DU";DU(Ø);
1840 REM solve for P,Q from PH-q-kGQ-hFT
1850 F1=T(2)-Q(2)*T(1)/Q(1)+Q(2)*Q(2)/Q(1)/Q(1)
1860 QQ(Ø)=F1/(Q(5)/Q(2)-Q(4)/Q(1)+Q(2)*Q(3)/Q(1)/Q(1))
1870 PP(2)=QQ(Ø)*Q(5)/Q(2)
1880 PP(1)=T(1)+QQ(Ø)*Q(3)/Q(1)-Q(2)/Q(1)
1890 LPRINT "QQ";QQ(Ø);
1900 REM move Q(I),DU(I),Y(I) for next step calculation
1910 FOR I=1 TO NSS
1920 Q1(I)=Q(I)
1930 NEXT I
1940 FOR I=Ø TO NG+K-1
1950 Y(NG+K-I)=Y(NG+K-I-1)
1960 NEXT I
1970 FOR I=Ø TO NH+K-1
1980 DU(NH+K-I)=DU(NH+K-I-1)
1990 NEXT I
2000 FOR I=Ø TO NB+K-1
2010 U(NB+K-I)=U(NB+K-I-1)
2020 NEXT I
2030 N1=N1+1
2040 IF N1=NN1 GOTO 2070
2050 LPRINT "Y";Y(Ø);"W";W(Ø)
2060 GOTO 450
2070 END
```

Appendix 10 : Controller Algorithm Implemented on the MC68000 Microcomputer

16-bit Fixed-Point Arithmetic

The algorithm is implemented in 16-bit fixed-point arithmetic on the MC68000 microcomputer. The representation of a word is shown in Fig. A10.1.

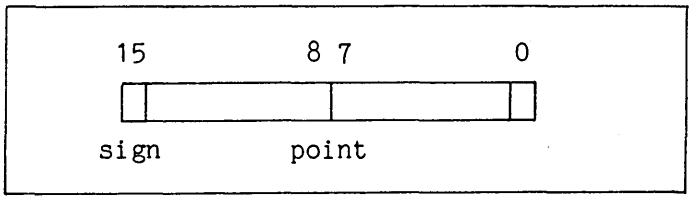


Fig. A10.1 One Fixed-Point Word

In this format, a 16-bit number is composed of 1 bit sign, 7 bit integer and 8 bit fraction, and it lies in the range of $-128 < X < 127.996$. Selection of this range, however, causes overflows in computation. Thus, after every step of computation where overflow may occur, the program will test if the overflow has occurred and, if it has, an overflow treatment subroutine will be called. The input data is a fraction of 8-bit length and the output is also 8-bit length. The input and output of the process should be scaled such that they are the same magnitude. This will improve the numerical conditions in the estimation. The program implemented in MC 68000 assembly language is given as follows. Depending upon the control flowchart in use, the sections of the program are linked.

```
* Motorola 68000 Assembly Language
* n is the number of parameters to be estimated
* For the first order system n=3
* Z^T(t) = [ u(t-1), y(t-1), y(t-2) ]
* θ(t) = [ h0, g0, g1 ]
```

```
*          [p11 p12 p13]
* P(t) =    [p21 p22 p23]
*          [p31 p32 p33]
* For the second order system n = 5

* Z^T(t) = [ u(t-1), u(t-2), y(t-1), y(t-2), y(t-3) ]
* θ(t) = [ h0, h1, g0, g1, g2 ]
*          [p11 . . . p15]
*          [              ]
* P(t) =   [              ]
*          [              ]
*          [p51. .  . p55]
*
*
* Z^T(t) = [ z1j]
* θ̇(t) = [ θ1j]
* P(t) = [ pij]
* For first order system i=3, j=3
* For second order system i=5, j=5
*
* MC 6821 I/O interface addressing on MC68000 education computer board
* PRA  Peripheral Register A
* PRB  Peripheral Register B
* MC68230 PI/T addressing on the MC68000 education microcomputer board
* DDRA Data Direction Register A
* DDRB Data Direction Register B
* CRA  Control Register A
* CRB  Control Register B
* CPRH Counter Preload Register High
* CPRM Counter Preload Register Middle
* CPRL Counter Preload Register Lower
* TCR  TImer Control Register
* TIVR Timer Interrupt Vector Register
* TSR  Timer Status Register
* SAIHR at this location in RAM, the starting address of the intrrupt
* handler routine is stored


u1-u2   DS.W    2   ; controller output u(t-1),u(t-2)
Y0-Y3   DS.W    4   ; process output y(t)...y(t-3)
U1      DS.W    1   ; integrator output U(t-1)
T1-T2   DS.W    2   ; time constants t1 and t2 for pole assignment
θ1-θ5   DS.W    5   ; h0,h1,g0,g1,g2
P1-P25  DS.W    25  ; elements of matrix P(t), P1=p11
                    ; P2=p12... P6=p21, etc.
W       DS.W    1   ; setpoint
L1-L5   DS.W    5   ; intermediate values
S1-S5   DS.W    5   ; intermediate values
H1-H25  DS.W    25  ; intermediate values
K1-K25  DS.W    25  ; intermediate values
R1-R5   DS.W    5   ; intermediate values
DDRA    EQU     $3FFF9
CRA     EQU     $3FFFB
DDRB    EQU     $3FFFD
CRB     EQU     $3FFFF
```

```
PRA     EQU       $3FFF9
PRB     EQU       $3FFFD
SAIHR   EQU       $6C
TCR     EQU       $10021
TIVR    EQU       $10023
CPRH    EQU       $10027
CPRM    EQU       $10029
CPRL    EQU       $1002B
TSR     EQU       $10035

* Subroutine
* Matrix Multiplication  [aij][bi1]=[ci1], i=1...n, j=1...n
* Entry: D1, D2 - n
*        A1     - $a11
*        A2     - $b11
*        A3     - $c11
* Exit:  Results are in the memory location from $c11
MX:   MOVE.L   A2,A4
MX1:  MOVE.W   (A1)+,D0
      MULS.W   (A4)+,D0
      ADD.L    D0,D4
      DBF.L    D1,MX1
      BSR      LMS
      MOVE.W   D4,(A3)+
      MOVE.L   #(n-1),D1
      MOVE.L   A2,A4
      MOVE.L   #0,D4
      DBF.L    D2,MX1
      RTS


* Overflow Treatment
* Check a 32-bit number with 1 bit sign, 15 bit integer and 16 bit
* fraction in D4, if it is between -128 to 127.996, then return, if it
* is biger than 127.996, then place 127.996 in D4, if it is lower than
* -127.996, then place -127.996 in D4.
* Entry: D4      - Number
* Exit:  D4      - The number after overflow treatment

LMS:    ASR.L    #8,D4
        CMP.L    #$7FFF,D4
        BGT      LMS2
        CMP.L    #$FFFF8000,D4
        BLT      LMS3
LMS1:   RTS
LMS2:   MOVE.W   #$7FFF,D4
        BRA      LMS1
LMS3:   MOVE.W   #$8000,D4
        BRA      LMS1
* Initialization
* Set USS with n1
* Set USP with n2

INI:    MOVE.L   #$n1,A7        ;set USS
        MOVE.L   #$n2,A1
        MOVE.L   A1,USP         ;set USP
```

```
        MOVE.W  #$2000,SR       ;interrupt enable
        MOVE.W  #$7FFF,$p11     ;
        MOVE.W  #$7FFF,$p22     ;
        MOVE.W  #$7FFF,$p33     ;
        MOVE.W  #$7FFF,$p44     ;
        MOVE.W  #$7FFF,$p55     ;set initial values of pij, i=j
        MOVE.W  #$     ,$θ1     ;
        MOVE.W  #$     ,$θ2     ;
        MOVE.W  #$     ,$θ3     ;
        MOVE.W  #$     ,$θ4     ;
        MOVE.W  #$     ,$θ5     ;set initial values of θ1j
        MOVE.W  #$     ,$T1     ;
        MOVE.W  #$     ,$T2     ;set T1, T2
        MOVE.L  #$     ,D1
        LEA     $u1    ,A1
INI1:   MOVE.W  #0,(A1)+
        DBF.L   D1,INI1         ;clear memory locations
        MOVE.B  #0,CRB          ;
        MOVE.B  #$FF,DDRB       ;
        MOVE.B  #4,CRB          ;I/O port B output
        MOVE.B  #0,CRA          ;
        MOVE.B  #0,DDRA         ;
        MOVE.B  #4,CRA          ;I/O port A input
        MOVE.B  #$0,CPRH        ;
        MOVE.B  #$5,CPRM        ;
        MOVE.B  #$DC,CPRL       ;set interrupt interval 12 ms
        MOVE.B  #$1B ,TIVR      ;interrupt vector
        MOVE.B  #$MA,SAIHR      ;interrupt handler routine starting address
        MOVE.B  #$A1,TCR        ;enable interrupt
INI2:   NOP
        BRA     INI2            ;wait for interrupt


* Interrupt Handler Routine
* Sampling data
* Exit: sampled data in $Y0

MJ:     MOVE.B  #$36,CRA
        NOP
        NOP
        MOVE.B  #$3E,CRA
MJ1:    BTST    #7,CRA
        BEQ     MJ1
        MOVE.B  PRA,$Y0


* y(t)+p1y(t-1)+p2y(t-2)+q0u(t-1) computation
* Entry: D5     - p1
*        D6     - p2
*        D7     - q0
* Exit:  D7     - Result

MA:     MOVE.B  #1,TSR          ;enable next interrupt
        MULS.W  $Y1,D5
        MULS.W  $Y2,D6
        MULS.W  $u1,D7
        ADD.L   D5,D6
```

```
        ADD.L   D6,D7
        MOVE.W  $Y0,D5
        MULS.W  #$100,D5
        ADD.L   D5,D7

* Calculate [1+Z^T(t)P(t-1)Z(t)]
* Exit: Result in D6

MB:     LEA     $P1,A2
        LEA     $P1,A3
        LEA     $L1,A4
        MOVE.L  #4,D3
MB1:    MOVE.L  #4,D2
        MOVE.L  #0,D4
        LEA     $u1,A1
MB2:    MOVE.W  (A1)+,D0
        MULS.W  (A2),D0
        ADD.L   D0,D4
        ADDA    #10,A2
        DBF.L   D2,MB2
        BSR     LMS
        MOVE.W  D4,(A4)+
        ADDA    #2,A3
        MOVE.L  A3,A2
        DBF.L   D3,MB1          ;Lij=z11p1j+z12p2j+...+z1npnj ;for j=1 to n
        MOVE.L  #0,D4
        MOVE.L  #4,D1
        LEA     $L1,A1
        LEA     $u1,A2
MB3:    MOVE.W  (A1)+,D0
        MULS.W  (A2)+,D0
        ADD.L   D0,D4
        DBF.L   D1,MB3          ;L11z11+L12z21+.....+L1nzn1
        ADD.L   #1,D4           ;k=L11z11+L12z21+.....+L1nzn1+1
        BSR     LMS
        MOVE.W  D4,D6           ;result in D6

* Calculate
*P(t)=P(t-1)-P(t-1)Z(t)Z^T(t)P(t-1)[1+Z^T(t)P(t-1)Z(t)]^-1
* Exit: P(t) in memory locations from address $P1

MC:     MOVE.L  #4,D1
        MOVE.L  D1,D2
        LEA     $P1,A1
        LEA     $u1,A2
        LEA     $S1,A3
        MOVE.L  #0,D4
        BSR     MX              ;[pij][zj1]=[si1], i=1..n  ;P(t-1)Z(t)
        LEA     $S1,A1
        LEA     $H1,A3
        MOVE.L  #4,D3
MC1:    LEA     $u1,A2
        MOVE.L  #4,D2
        MOVE.W  (A1)+,D1
MC2:    MOVE.W  D1,D4
        MULS.W  (A2)+,D4
```

```
        DIVS    D6,D4
        MOVE.W  D4,(A3)+
        DBF.L   D2,MC2
        DBF.L   D3,MC1      ;hij=si1z1j, i=1..n, j=1..n ; hij=hij/k
        MOVE.L  #4,D5       ;P(t-1)Z(t)Z^T(t)/k
        MOVE.L  #0,D4
        LEA     $H1,A1
        LEA     $K1,A4
        LEA     $P1,A3
MC3:    MOVE.L  #4,D2
MC4:    MOVE.L  A3,A2
        MOVE.L  #4,D1
MC5:    MOVE.W  (A1)+,D0
        MULS.W  (A2),D0
        ADD.L   D0,D4
        BVS     OVER
MC6:    ADDA    #10,A2
        DBF.L   D1,MC5
        BSR     LMS
        MOVE.W  D4,(A4)+
        SUBA    #10,A1
        ADDA    #2,A3
        MOVE.L  #0,D4
        DBF.L   D2,MC4
        ADDA    #10,A1
        LEA     $P1,A3
        DBF.L   D5,MC3      ;kij=hi1p1j+hi2p2j+..+hinpnj
                            ;i=1...n, j=1...n
        BRA     MC8         ;
OVER:   BTST    #31,D0        ;P(t-1)Z(t)Z^T(t)P(t-1)[1+Z^T(t)P(t-1)Z(t)]
        BEQ     MC7
        MOVE.L  #$8000000,D4
        BRA     MC6
MC7:    MOVE.L  #$7FFFFFFF,D4
        BRA     MC6
MC8:    MOVE.L  #25,D1
        LEA     $P1,A1
        LEA     $K1,A2
        LEA     $P1,A3
MC9:    MOVE.W  (A1)+,D0
        MOVE.W  (A2)+,D2
        SUB.W   D2,D0
        BVS     OVER1
        MOVE.W  D0,(A3)+
MC10:   DBF.L   D1,MC9      ;pij=pij-kij,        P(t-1)
        BRA     MD
OVER1:  BTST    #15,D2
        BEQ     MC11
        MOVE.W  #$7FFF,(A3)+
        BRA     MC10
MC11:   MOVE.W  #$8000,(A3)+
        BRA     MC10
```

```
* Calculate θ(t)=θ(t-1)-P(t)Z(t)[Z^T(t)θ(t-1)-φ(t)]
* Exit: Results of θ(t) in the memory locations starting address $θ1
MD:     MOVE.L  #(n-1),D1
        LEA     $u1,A1
        LEA     $θ1,A2
        MOVE.L  #0,D4
MD1:    MOVE.W  (A1)+,D0
        MULS.W  (A2)+,D0
        ADD.L   D0,D4
        DBF.L   D1,MD1          ;Z^T(t)θ(t-1)
        SUB.L   D7,D4
        BSR     LMS
        MOVE.W  D4,D7           ;Z^T(t)θ(t-1)-φ(t)
        MOVE.L  #(n-1),D1
        MOVE.L  D1,D2
        LEA     $P1,A1
        LEA     $u1,A2
        LEA     $R1,A3
        MOVE.L  #0,D4
        BSR     MX              ;P(t)Z(t)
        MOVE.L  #(n-1),D1
        LEA     $R1,A1
        LEA     $R1,A2
MD2:    MOVE.W  (A1)+,D4
        MULS.W  D7,D4
        BSR     LMS
        MOVE.W  D4,(A2)+
        DBF.L   D1,MD2          ;P(t)Z(t)[Z^T(t)θ(t-1)-φ(t)]
        LEA     $θ1,A3
        LEA     $θ1,A1
        LEA     $R1,A2
        MOVE.L  #(n-1),D0
MD3:    MOVE.W  (A1)+,D4
        MOVE.W  (A2)+,D1
        SUB.W   D1,D4
        BVS     MD6
        CMPI.W  #0,D4
        BNE     MD4
        ADD.W   #1,D4
MD4:    MOVE.W  D4,(A3)+
MD5:    DBF.L   D0,MD3    ;θ(t)=θ(t-1)-P(t)Z(t)[Z^T(t)θ(t-1)-φ(t)]
        BRA     ME
MD6:    BTST    #15,D1
        BEQ     MD7
        MOVE.W  #$7FFF,(A3)+
        BRA     MD5
MD7:    MOVE.W  #$8000,(A3)+
        BRA     MD5


* Calculate controller output u(t)=[RW(t)-GY(t)]/H
* PID algorithm
* Entry: D3 - W(t), setpoint
* Exit:  Result u(t) in D1
```

```
ME:     MOVE.L  #0,D1
        LEA     $03,A1          ;g0
        MOVE.L  #(n3-1),D2      ;counter, if n=5, n3=3; if n=3, n3=2
ME1:    MOVE.W  (A1)+,D0
        MULS.W  D3,D0           ;giW(t)  i=0  to  n3-1
        ADD.L   D0,D1
        DBF.L   D2,ME1         ;(g0+g1+g2)W(t)
        LEA     $02,A1
        MOVE.W  (A1)+,D0
        MULS.W  $u1,D0          ;h1u(t-1)
        SUB.L   D0,D1           ;RW(t)-h1u(t-1)
        MOVE.W  (A1)+,D0
        MULS.W  $Y0,D0          ;g0y(t)
        SUB.L   D0,D1
        MOVE.W  (A1)+,D0
        MULS.W  $Y1,D0          ;g1y(t-1)
        SUB.L   D0,D1
        MOVE.W  (A1)+,D0
        MULS.W  $Y2,D0          ;RW(t)-GY(t)-h1u(t-1)
        SUB.L   D0,D1
        DIVS    $01,D1          ;u(t) in D1


* Move data for next step computation

MF:     MOVE.W  $u1,$u2
        MOVE.W  D1,$u1
        MOVE.W  $Y2,$Y3
        MOVE.W  $Y1,$Y2
        MOVE.W  $Y0,$Y1


* Integrator
* n4 upper limit of the integrator
* n5 lower limit of the integrator
* Integrator output limit

MG:     ADD.W   $U1,D1          ;integrate
        MOVE.W  D1,D2
        CMP.W   #n4,D1
        BGE     MG1
        CMP.W   #n5,D1
        BLE     MG2
        BRA     MG3
MG1:    MOVE.W  #n4,D1          ;integrator limitation
        BRA     MG3
MG2:    MOVE.W  #n5,D1          ;integrator limitation
MG3:    MOVE.W  D1,$U1
        CMPI.W  #$FF,D2
        BLE     MG4
        MOVE.W  #$FF,D2
        BRA     MG5
MG4:    CMPI.W  #0,D2
        BLE     MG6
MG5:    MOVE.B  D2,PRB          ;output data
        BRA     MI
MG6:    MOVE.B  #0,PRB          ;output data
MG7:    BRA     MI
```

```
* Calculate q0, p1, p2
* Exit: D7      - q0
        D6      - p2
        D5      - p1

MI:     MOVE.W  $T1,D0
        MULS.W  $02,D0
        DIVS    $01,D0          ;h1t1/h0
        MULS.W  #$100,D0
        MOVE.W  $02,D1
        MULS.W  D1,D1
        DIVS    $01,D1
        MULS.W  #$100,D1
        DIVS    $01,D1          ;h1h1/(h0h0)
        MULS.W  #$100,D1
        MOVE.W  $T2,D7
        MULS.W  #$100,D7
        SUB.L   D0,D7
        ADD.L   D1,D7           ;t2-h1t1/h0+h1h1/(h0h0)
        MOVE.W  $05,D4
        MULS.W  #$100,D4
        DIVS    $02,D4          ;g2/h1
        MULS.W  #$100,D4
        MOVE.W  $04,D1
        MULS.W  #$100,D1
        DIVS    $01,D1          ;g1/h0
        MULS.W  #$100,D1
        MOVE.W  $02,D3
        MULS.W  $03,D3
        DIVS    $01,D3
        MULS.W  #$100,D3
        DIVS    $01,D3          ;h1g0/h0h0
        MULS.W  #$100,D3
        SUB.L   D1,D4
        ADD.L   D3,D4           ;g2/h1-g1/h0+h1g0/(h0h0)
        BSR     LMS
        CMP.W   #0,D4
        BNE     MI1
        ADD.W   #1,D4
MI1:    DIVS    D4,D7           ;q0
        MOVE.W  D7,D6
        MULS.W  $05,D6
        DIVS    $02,D6          ;p2
        MOVE.L  D7,D0
        MULS.W  $03,D0
        DIVS    $01,D0
        MULS.W  #$100,D0        ;q0g0/h0
        MOVE.W  $02,D1
        MULS.W  #$100,D1
        DIVS    $01,D1
        MULS.W  #$100,D1        ;h1/h0
        MOVE.W  $T1,D4
        MULS.W  #$100,D4
        SUB.L   D1,D4
        ADD.L   D0,D4           ;t1-h1/h0+q0g0/h0
        BSR     LMS
        MOVE.W  D4,D5           ;p1
```

Floating-point arithmetic

It is generally impractical to employ floating-point representation in computers with less than 24 bit per word, since the precision (number of significant digits in the fraction) is not sufficient for most applications [53]. However, simulation experience and practical application showed that a fractional-part of 16-bits would be adequate for on line self-tuning algorithm [41].

To improve the accuracy of computation, software was rewritten in 24-bit floating-point arithmetic on 16-bit MC 68000 microcomputer. The representation of a floating-point word is shown in Fig. A10.2.

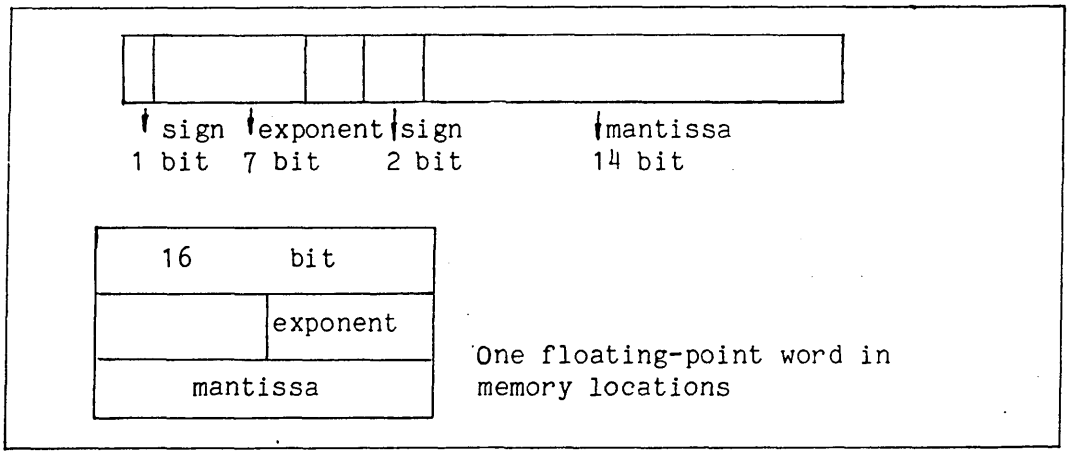


Fig. A10.2 One Floating-Point Word

In this format, the floating-point numbers lie in the range of $1.47*10^{-39} < X < 1.70*10^{+38}$. Numbers represented with 3 bytes make it possible to optimise floating-point calculations using 16-bit MC 68000; the operation time is only about 100 ms and this is not likely to cause any problem for most of processes.

The algorithm implemented in MC 68000 assembly language is given as follows. In this case, both input and output data are integers of 8-bit length.

```
* Motorola 68000
* 24-bit floating-point arithmetic
* Self-tuning PID algorithm

u1-u2   DS.L    2       ;controller output u(t-1),u(t-2)
Y0-Y3   DS.L    4       ;process output y(t),y(t-1),y(t-2),y(t-3)
U1      DS.L    1       ;interator output U(t-1)
T1-T1   DS.L    2       ;time constants t1 and t2 for pole assignment
p1-p2   DS.L    2       ;P(z^-1)=1+p1z^-1+p2z^-2
Q       DS.L    1       ;Q=q0
φ       DS.L    1       ;φ=P(z^-1)Y(t)+Q(z^-1)u(t-1)
θ1-θ5   DS.L    5       ;h0, h1, g0, g1, g2
W       DS.L    1       ;setpoint
NUM6-9  DS.L    4       ;32-bit intermediate results
S1-5    DS.L    5       ;intermediate results
P1-P25  DS.L    25      ;intermediate results
L1-5    DS.L    5       ;intermediate results
H1-25   DS.L    25      ;intermediate results
K1-25   DS.L    25      ;intermediate results
R1-5    DS.L    5       ;intermediate results


* Subroutine
* Normalization to the right
* Normalize a floating-point number
* Entry: Exponent in D0 bit0-7
*        Fraction in D1 bit0-15
* Exit:  Exponent in D0 bit0-7
*        Fraction in D1 bit0-15

NORDW:  MOVE.W   D1,D2
        ASL.W    #1,D2
        ROXL.W   #1,D2
        MOVE.B   #0,D3
        ADDX     D3,D2
        BTST     #0,D2
        BNE      NOR1
        RTS
NOR1:   ADDB     #1,D0
        ASR.W    #1,D1
        RTS

* Subroutine
* Normalization to the left
* Normalize a floating-point number
* Entry: Exponent in D0 bit0-7
*        Fraction in D1 bit0-15
* Exit:  Exponent in D0 bit0-7
*        Fraction in D1 bit0-15

NORUP:  MOVE.W   D1,D2
        AND.W    D2,D2
        BEQ      NULL
        AND.W    #$E000,D2
        CMPI.W   #$2000,D2
        BEQ      CPL
        CMPI.W   #$C000,D2
```

```
        BEQ     CPL
        ASL.W   #1,D1
        SUB.B   #1,D0
        BRA     NORUP
NULL:   MOVE.B  #0,D0
        CPL:    RTS


* Number disassembly subroutine 1
* Entry: D5 the number
* Exit:  D0 bit0-7 the exponent of the number
*        D1 bit0-15 the fraction of the number

SPER1: MOVE.W  D5,D1
       ASR.L   #8,D5
       ASR.L   #8,D5
       MOVE.B  D5,D0
       RTS


* Number disassembly subroutine 2
* Entry: D5 the number
* Exit:  D2 bit0-7 the exponent of the number
*        D3 bit0-15 the fraction of the number

SPER2: MOVE.W  D5,D3
       ASR.L   #8,D5
       ASR.L   #8,D5
       MOVE.B  D5,D2
       RTS


* Number assembly subroutine
* Entry: D0 bit0-7 the exponent of the number
*        D1 bit0-15 the fraction of the number
* Exit:  D0 the number

STOR:  ASL.L   #8,D0
       ASL.L   #8,D0
       AND.L   #$FFFF,D1
       ADD.L   D1,D0
       RTS


* Subroutine
* The sum of two numbers
* Entry: D0 bit0-7 the exponent of number 1
*        D1 bit0-15 the fraction of number 1
*        D2 bit0-7 the exponent of number 2
*        D3 bit0-15 the fraction of number 2
* Exit:  D0 bit0-7 the exponent of the sum
*        D1 bit0-15 the fraction of the sum

AD:    MOVE.L  #0,D5
       MOVE.B  D2,D5
       SUB.B   D0,D5
       BEQ     NOS
       BGT     BIG
```

```
        NEG.B   D5
        ASR.W   D5,D3
        BRA     NOS
BIG:    ASR.W   D5,D1
        MOVE.B  D2,D0
NOS:    ADD.W   D3,D1
        BSR     NORDW
        BSR     NORUP
        RTS


* Subroutine
* The subtraction of two numbers
* Entry: D0 bit0-7 the exponent of number 1
*        D1 bit0-15 the fraction of number 1
*        D2 bit0-7 the exponent of number 2
*        D3 bit0-15 the fraction of number 2
* Exit:  D0 bit0-7 the exponent of the subtraction
*        D1 bit0-15 the fraction of the subtraction

SUBT:  NEG.W   D3
       BRA     AD


* Subroutine
* The multiplication of two numbers
* Entry: D0 bit0-7 the exponent of number 1
*        D1 bit0-15 the fraction of number 1
*        D2 bit0-7 the exponent of number 2
*        D3 bit0-15 the fraction of number 2
* Exit:  D0 bit0-7 the exponent of the multiplication
*        D1 bit0-15 the fraction of the multiplication

MULTP: ADD.B   D2,D0
       ASL.W   #1,D1
       ASL.W   #1,D3
       MULS.W  D3,D1
       ASR.L   #8,D1
       ASR.L   #8,D1
       BSR     NORDW
       BSR     NORUP
       RTS

* Subroutine
* The division of two numbers
* Entry: D0 bit0-7 the exponent of number 1
*        D1 bit0-15 the fraction of number 1
*        D2 bit0-7 the exponent of number 2
*        D3 bit0-15 the fraction of number 2
* Exit:  D0 bit0-7 the exponent of the result
*        D1 bit0-15 the fraction of the result


DIVID: AND.W   D3,D3
       BNE     DIVID1
       ADD.W   #1,D3
```

```
DIVID1:SUB.B    D2,D0
       ASL.L    #1,D1
       MULS.W   #$4000,D1
       ASL.L    #1,D3
       DIVS.W   D3,D1
       BSR      NORDW
       BSR      NORUP
       RTS

* Subroutine
* The sum of a series of numbers
* Entry: the starting address of the n numbers is $NUM1
* Exit:  the sum in D0

TEMAD: MOVE.L   #(n-2),D7
       LEA      $NUM1,A4
       MOVE.L   (A4)+,D5
       BSR      SPER1
TEMAD1:MOVE.L   (A4)+,D5
       BSR      SPER2
       BST      AD
       DBF.L    D7,TEMAD1
       BSR      STOR
       RTS


* Subroutine
* Matrix multiplication
* [aij][bj1]=[ci1]  i=1..n  j=1..n
* Entry: A1  $a11
*        A2  $b11
*        A3  $c11
* Exit: results in the memory locations starting address $c11

MX:    LEA      $NUM1,A4
       MOVE.L   #(n-1),D4
       MOVE.L   #(n-1),D6
       MOVE.L   A2,A5
MX1:   MOVE.L   (A1)+,D5
       BSR      SPER1
       MOVE.L   (A2)+,D5
       BSR      SPER2
       BSR      MULTP
       BSR      STOR
       MOVE.L   D0,(A4)+
       DBF.L    D4,MX1
       BSR      TEMAD
       MOVE.L   D0,(A3)+
       MOVE.L   #(n-1),D4
       MOVE.L   A5,A2
       LEA      $NUM1,A4
       DBF.L    D6,MX1
       RTS
```

```
* φ=Y(t)+p1Y(t-1)+p2Y(t-2)+q0u(t-1)
* Exit: result in $φ


MA:     LEA     $p1,A1
        LEA     $Y1,A2
        LEA     $NUM1,A3
        MOVE.L  #2,D7
MA1:    MOVE.L  (A1)+,D5
        BSR     SPER1
        MOVE.L  (A2)+,D5
        BSR     SPER2
        BSR     MULTP
        BSR     STOR
        MOVE.L  D0,(A3)+
        DBF.L   D7,MA1          ;p1Y(t-1), p2Y(t-2), q0u(t-1)
        LEA     $NUM1,A3
        MOVE.L  #2,D7
        MOVE.L  $Y0,D5
        BSR     SPER1
MA2:    MOVE.L  (A3)+,D5
        BSR     SPER2
        BSR     AD
        DBF.L   D7,MA2          ;Y(t)+p1Y(t-1)+p2Y(t-2)+q0u(t-1)
        BSR     STOR
        MOVE.L  D0,$φ           ;store φ


* Z^T(t)P(t-1)
* L1j=z11p1j+z12p2j+....z1npnj
* Exit: result in memory locations starting address $L1


MB:     LEA     $P1,A2
        MOVE.L  A2,A3
        LEA     $L1,A5
        MOVE.L  #(n-1),D6       ;counter
MB1:    LEA     $NUM1,A4
        LEA     $u1,A1
        MOVE.L  #(n-1),D4
MB2;    MOVE.L  (A1)+,D5
        BSR     SPER1
        MOVE.L  (A2),D5
        BSR     SPER2
        BSR     MULTP
        BSR     STOR
        MOVE.L  D0,(A4)+
        ADDA    #20,A2
        DBF.L   D4,MB2
        BSR     TEMAD           ; L1j=z11p1j+z12p2j+....z1npnj
        MOVE.L  D0,(A5)+        ; j=1..n
        ADDA    #4,A3
        MOVE.L  A3,A2
        DBF.L   D6,$MB1
```

```
* 1/[Z^T(t)P(t-1)Z(t)+1]
* Exit: result in location $NUM6

MC:     LEA     $L1,A1
        LEA     $u1,A2
        LEA     $NUM1,A4
        MOVE.L #(n-1),D4
MC1:    MOVE.L (A1)+,D5
        BSR     SPER1
        MOVE.L (A2)+,D5
        BSR     SPER2
        BSR     MULTP
        BSR     STOR
        MOVE.L D0,(A4)+
        DBF.L   D4,MC1      ;L11u1, L12u2, L13Y1, L14Y2, L15Y3
        MOVE.L #3,D7
        LEA     $NUM1,A4
        MOVE.L (A4)+,D5
        BSR     SPER1
MC2:    MOVE.L (A4)+,D5
        BSR     SPER2
        BSR     AD
        DBF.L   D7,MC2      ;L11u1+L12u2+L13Y1+L14Y2+L15Y3
        MOVE.B #1,D2
        MOVE.W #$2000,D3    ;1+Z^T(t)P(t-1)Z(t)
        BSR     AD
        MOVE.B D0,D2
        MOVE.W D1,D3
        MOVE.B #1,D0
        MOVE.W #$2000,D1
        BSR     DIVID       ;1/[ 1+Z^T(t)P(t-1)Z(t) ]
        BSR     STOR
        MOVE.L D0,NUM6



* P(t-1)-P(t-1)Z(t)Z^T(t)P(t-1)/[Z^T(t)P(t-1)Z(t)+1]
* Exit: results in memory locations starting address $P1

MD:     LEA     $P1,A1
        LEA     $u1,A2
        LEA     $S1,A3
        BSR     MX           ;si1=pi1z11+pi2z21+...+pinzn1
                             ;i=1  to n, P(t-1)Z(t)
        LEA     $S1,A1       ;
        LEA     $H1,A3
        MOVE.L #(n-1),D4
MD1:    LEA     $u1,A2
        MOVE.L #(n-1),D6
        MOVE.L (A1)+,D7
MD2:    MOVE.L D7,D5
        BSR     SPER1
        MOVE.L (A2)+,D5
        BSR     SPER2
        BSR     MULTP
        BSR     STOR
```

```
        MOVE.L  D0,(A3)+
        DBF.L   D6,MD2
        DBF.L   D4,MD1        ;hij=si1z1j, i=1 to n, j=1 to n
        MOVE.L  #(n-1),D4
        LEA     $H1,A1
        LEA     $P1,A2
        LEA     $K1,A3
MD3:    MOVE.L  #(n-1),D6
MD4:    LEA     $NUM1,A4
        MOVE.L  A2,A5
        MOVE.L  #(n-1),D7
MD5:    MOVE.L  (A1)+,D5
        BSR     SPER1
        MOVE.L  (A5),D5
        BSR     SPER2
        BSR     MULTP
        BSR     STOR
        MOVE.L  D0,(A4)+
        ADDA    #20,A5
        DBF.L   D7,MD5
        BSR     TEMAD
        MOVE.L  D0,(A3)+
        SUBA    #20,A1
        ADDA    #(n-1),A2
        DBF.L   D6,MD4
        ADDA    #20,A1
        LEA     $P1,A2
        DBF.L   D4,MD3        ;Kij=hi1p1j+hi2p2j+...+ hinpnj
                              ;i=1 to n, j=1 to n
MD6:    MOVE.L  #15,D4          ;P(t-1)Z(t)Z^T(t)P(t-1)
        LEA     $K1,A1
        LEA     $NUM6,A2
        LEA     $P1,A3
        MOVE.L  A3,A4
MD7:    MOVE.L  (A1)+,D5
        BSR     SPER1
        MOVE.L  (A2),D5
        BSR     SPER2
        BSR     MULTP
        MOVE.B  D0,D2
        MOVE.W  D1,D3
        MOVE.L  (A3)+,D5
        BSR     SPER1
        BSR     SUBT
        BSR     STOR
        MOVE.L  D0,(A4)+
        DBF.L   D4,MD7        ;P(t)
```

* $\theta(t-1)-P(t)Z(t)[Z^T(t)\theta(t-1)-\phi\ ]$

```
MF:     LEA     $P1,A1
        LEA     $u1,A2
        LEA     $R1,A3
        BSR     MX              ;P(t)Z(t)
        LEA     $u1,A1
        LEA     $θ1,A2
```

```
        LEA     $NUM1,A4
        MOVE.L  #(n-1),D4
MF1:    MOVE.L  (A1)+,D5
        BSR     SPER1
        MOVE.L  (A2)+,D5
        BSR     SPER2
        BSR     MULTP
        BSR     STOR
        MOVE.L  D0,(A4)+
        DBF.L   D4,MF1
        MOVE.L  #(n-2),D7
        LEA     $NUM1,A4
        MOVE.L  (A4)+,D5
        BSR     SPER1
MF2:    MOVE.L  (A4)+,D5
        BSR     SPER2
        BSR     AD
        DBF.L   D7,MF2          ;Z^T(t)θ(t-1)
        MOVE.L  $φ,D5
        BSR     SPER2
        BSR     SUBT
        BSR     STOR            ;Z^T(t)θ(t-1)-φ
        MOVE.L  #(n-1),D4
        MOVE.L  D0,D7
        LEA     $R1,A1
        LEA     $θ1,A2
        MOVE.L  A2,A3
MF3:    MOVE.L  (A1)+,D5
        BSR     SPER1
        MOVE.L  D7,D5
        BSR     SPER2
        BSR     MULTP
        MOVE.B  D0,D2
        MOVE.W  D1,D3
        MOVE.L  (A2)+,D5
        BSR     SPER1
        BSR     SUBT
        BSR     STOR
        MOVE.L  D0,(A3)+
        DBF.L   D4,MF3    ;θ(t)=θ(t-1)-P(t)Z(t)[Z^T(t)θ(t-1)-φ]


* u(t)=[RW(t)-GY(t)]/H
* Exit: D0 the exponent of u (t)
*       D1 the fraction of u (t)

MG:     LEA     $NUM1,A3
        LEA     $NUM7,A2
        MOVE.L  #1,D4
        LEA     $θ3,A1
        MOVE.L  (A1)+,D5
        BSR     SPER1
MG1:    MOVE.L  (A1)+,D5
        BSR     SPER2
        BSR     AD
        DBF.L   D4,MG1          ;g0+g1+g2
        MOVE.L  $W,D5
        BSR     SPER2
```

```
        BSR     MULTP
        BSR     STOR
        MOVE.L  D0,(A2)+        ;(g0+g1+g2)W(t)
        LEA     $03,A1
        LEA     $Y0,A4
        MOVE.L  #2,D4
MG2:    MOVE.L  (A1)+,D5
        BSR     SPER1
        MOVE.L  (A4)+,D5
        BSR     SPER2
        BSR     MULTP
        BSR     STOR
        MOVE.L  D0,(A3)+
        DBF.L   D4,MG2          ;g0Y(t), g1Y(t-1), g2Y(t-2)
        LEA     NUM1,A3
        MOVE.L  #1,D4
        MOVE.L  (A3)+,D5
        BSR     SPER1
MG3:    MOVE.L  (A3)+,D5
        BSR     SPER2
        BSR     AD
        DBF.L   D4,MG3          ;g0Y(t)+g1Y(t-1)+g2Y(t-2)
        NEG.W   D1
        BSR     STOR
        MOVE.L  D0,(A2)+        ;[g0+g1+g2]W(t)-[g0Y(t)+g1Y(t-1)+g2Y(t-2)]
        MOVE.L  $02,D5
        BSR     SPER1
        MOVE.L  $u1,D5
        BSR     SPER2
        BSR     MULTP
        NEG.W   D1              ;-h1u(t-1)
        LEA     $NUM1,A2
        MOVE.L  #1,D4
MG4:    MOVE.L  (A2)+,D5
        BSR     SPER2
        BSR     AD
        DBF.L   D4,MG4
        MOVE.L  $01,D5
        BSR     SPER2
        BSR     DIVID           ;u(t)


* q0=[t2-h1t1/h0+h1h1/h0h0]/[g2/h1-g1/h0+h1g0/h0h0]
* Exit: q0 in the memory location addressing $Q

MH:     LEA     $NUM1,A1
        MOVE.L  $02,D5
        BSR     SPER1
        MOVE.L  $T1,D5
        BSR     SPER2
        BSR     MULTP
        MOVE.L  $01,D5
        BSR     SPER2
        BSR     DIVID
        NEG.W   D1
        BSR     STOR
```

```
MOVE.L  D0,(A1)+        ;-h1t1/h0
MOVE.L  $02,D5
BSR     SPER1
MOVE.B  D0,D2
MOVE.W  D1,D3
BSR     MULTP
MOVE.B  D0,D4
MOVE.W  D1,D6
MOVE.L  $01,D5
BSR     SPER1
MOVE.B  D0,D2
MOVE.W  D1,D3
BSR     MULTP
MOVE.B  D0,D2
MOVE.W  D1,D3
MOVE.B  D4,D0
MOVE.W  D6,D1
BSR     DIVID           ;h1h1/h0h0
MOVE.L  -(A1),D5
BSR     SPER2
BSR     AD              ;h1h1/(h0h0)-h1t1/h0
MOVE.L  $T2,D5
BSR     SPER2
BSR     AD
BSR     STOR
MOVE.L  D0,(A1)+        ;t1+h1h1/(h0h0)-h1t1/h0
LEA     $03,A2
MOVE.L  (A2)+,D5
BSR     SPER1
MOVE.L  $02,D5
BSR     SPER2
BSR     MULTP
MOVE.B  D0,D4
MOVE.W  D1,D6
MOVE.L  $01,D5
BSR     SPER1
MOVE.B  D0,D2
MOVE.W  D1,D3
BSR     MULTP
MOVE.B  D0,D2
MOVE.W  D1,D3
MOVE.B  D4,D0
MOVE.W  D6,D1
BSR     DIVID
BSR     STOR
MOVE.L  D0,(A1)+        ;g0h1/(h0h0)
MOVE.L  (A2)+,D5
BSR     SPER1
MOVE.L  $01,D5
BSR     SPER2
BSR     DIVID
NEG.W   D1
BSR     STOR
MOVE.L  D0,(A1)+        ;-g0/h0
MOVE.L  (A2)+,D5
BSR     SPER1
MOVE.L  $02,D5
BSR     SPER2
```

```
        BSR     DIVID
        MOVE.L  -(A1),D5
        BSR     SPER2
        BSR     AD
        MOVE.L  -(A1),D5
        BSR     SPER2
        BSR     AD              ;g2/h1-g1/h0+h1g0/(h0h0)
        MOVE.B  D0,D2
        MOVE.W  D1,D3
        MOVE.L  -(A1),D5
        BSR     SPER1
        BSR     DIVID           ;q0
        MOVE.B  D0,D4
        MOVE.W  D1,D6
        MOVE.B  D0,D2
        MOVE.W  D1,D3
        BSR     STOR
        MOVE.L  D0,$Q


* p2=q0g2/h1
* Exit: p2 in the memory location addressing $p2


MI:     MOVE.L  $05,D5
        BSR     SPER1
        BSR     MULTP
        MOVE.L  $02,D5
        BSR     SPER2
        BSR     DIVID
        BSR     STOR
        MOVE.L  D0,$p2          ;p2


* p1=t1+q0g0/h0-h1/h0
* Exit: p1 in the memory location addressing $p1

MJ:   MOVE.B  D4,D0
      MOVE.W  D6,D1
      MOVE.L  $03,D5
      BSR     SPER2
      BSR     MULTP
      MOVE.L  $01,D5
      BSR     SPER2
      BSR     DIVID
      MOVE.B  D0,D4
      MOVE.W  D1,D6         ;q0g0/h0
      MOVE.L  $02,D5
      BSR     SPER1
      MOVE.L  $01,D5
      BSR     SPER2
      BSR     DIVID
      NEG.W   D1
      MOVE.B  D4,D2
      MOVE.W  D6,D3
      BSR     AD            ;q0g0/h0-h1/h0
      MOVE.L  $T1,D5
      BSR     SPER2
```

Appendix 11 : Experimental Results in Tabular Form

This Appendix contains some experimental results which are classified and tabulated as follows:

Table 1-8 are experimental results obtained using the stepped bore reduction unit.

Table 1 : Copper wire - WVG 23, 110°C

Table 2 : Copper wire - WVG 23, 120°C

Table 3 : Copper wire - Nylon 6, 240°C

Table 4 : Copper wire - Nylon 6, 260°C

Table 5 : Copper wire - Nylon 6, 280°C

Table 6 : Copper wire - ELVAX 650, 140°C

Table 7 : Copper wire - ELVAX 650, 150°C

Table 8 : Copper wire - ELVAX 650, 160°C

Table 9 : Temperature step response of the polymer melt chamber

Table 10 : Closed-loop step response of the polymer melt chamber

Table 11 : Speed step response of the hydraulic drive system

Table 12 : Sampled data / output data of the microcomputer

Table 13 : Open-loop step response of the speed control system

Table 14 : Fluctuation of the PRA readings after a rapid change in drawing speed

Table 15 : PRA / length of the produced wire, ELVAX 650, 150°C

Table 16 : PRA / length of the produced wire, ELVAX 650, 150°C

Table 17 : PRA / length of the produced wire, ELVAX 650, 150°C

Table 18 : PRA / length of the produced wire, WVG 23, 110°C

Table 19 : Open-loop step response of the temperature platform

Table 20 : system output of the temperature platform at commissioning stage

Table 21 : Closed-loop step response of the temperature platform

Table 1 : Copper wire - WVG 23, 110°C

(RH=4, RL=2.5, D=0.45 mm)

Speed ms^{-1}	0.088	0.12	0.16	0.184	0.224	0.272	0.328	0.384	0.44
PRA	0	2.2	3.1	3.9	5.4	8.45	10.3	12.3	13.6
	0.48	0.512	0.552	0.576	0.624	0.704	0.76	0.8	0.864
	14.0	14.8	16.4	16.4	16.4	16.8	18.4	18.0	18.0
	0.896	0.96	1.032	1.12	1.2	1.248			
	18.0	18.0	16.8	16.8	16.0	15.6			

Table 2 : Copper wire - WVG 23, 120°C

(RH=4, RL=2.5, D=0.45 mm)

Speed ms^{-1}	0.232	0.31	0.33	0.384	0.448	0.512	0.568	0.608	0.656
PRA	2.2	4.3	5.2	6.1	7.3	8.6	8.6	9.4	10.3
	0.704	0.76	0.832	0.912	0.976	1.04	1.12		
	10.7	11.1	12.3	10.7	10.7	10.7	10.3		

Table 3 : Copper wire - Nylon 6, 240°C

(RH=4, RL=2.5, D=0.45 mm)

Speed ms^{-1}	0.85	0.98	1.05	1.13	1.2	1.3	1.38	1.5	1.6
PRA	13.4	16.6	18.6	18.5	20.6	20.6	21.3	20.6	21.0
	1.65	1.75	1.9	1.93	2.0				
	21.0	23.3	21.3	20.1	19.6				

Table 4 : Copper wire - Nylon 6, 260°C

(RH=4, RL=2.5, D=0.45 mm)

Speed ms^{-1}	0.2	0.33	0.36	0.4	0.51	0.55	0.61	0.69	0.71
PRA	2.9	9.5	7.6	13.8	13.0	7.3	11.1	9.4	13.6
	0.72	0.83	0.85	0.93	1.03	1.15	1.2	1.26	
	16.6	18.2	14.9	16.9	18.8	18.2	19.6	18.8	

Table 5 : Copper wire - Nylon 6, 280°C

(RH=4, RL=2.5, D=0.45 mm)

Speed ms^{-1}	0.25	0.3	0.33	0.38	0.52	0.59	0.65	0.69	0.75
PRA	2.0	4.0	9.2	5.6	10.6	8.8	9.81	11.7	14.0
	0.83	0.9	0.96	1.01	1.05	1.14	1.18	1.2	1.25
	10.0	13.5	12.7	13.6	16.0	14.1	14.7	12.0	14.3

Table 6 : Copper wire - ELVAX 650, 140°C

(RH=4, RL=2.5, D=0.45 mm)

Speed ms^{-1}	0.05	0.12	0.13	0.15	0.16	0.16	0.18	0.18	0.2
PRA	4.4	4.8	6.6	8.7	8.7	6.9	12.9	7.8	8.69
	0.22	0.25	0.27	0.3	0.34	0.38	0.42	0.47	0.51
	9.5	9.5	9.5	10.8	10.4	12.9	15.0	17.4	17.0
	0.53	0.57	0.58	0.61	0.65	0.71	0.74	0.77	0.8
	15.0	17.0	17.0	17.0	17.8	17.4	18.6	21.0	20.0

Table 7 : Copper wire - ELVAX 650, 150°C

(RH=4, RL=2.5, D=0.45 mm)

Speed ms^{-1}	0.03	0.08	0.1	0.1	0.14	0.17	0.19	0.22	0.28
PRA	4.4	6.5	4.8	6.5	6.5	6.5	8.7	9.1	9.5
	0.32	0.36	0.41	0.44	0.59	0.65	0.7	0.72	0.75
	10.8	9.5	12.1	12.1	12.9	12.9	14.5	12.9	15.8
	0.75	0.8	0.86	0.92	0.94	0.98	1.04		
	17.8	17.8	17.8	19.0	19.4	21.0	21.0		

Table 8 : Copper wire - ELVAX 650, 160°C

(RH=4, RL=2.5, D=0.45 mm)

Speed ms^{-1}	0.038	0.06	0.09	0.12	0.14	0.18	0.2	0.24	0.26
PRA	4.4	4.4	8.3	5.7	9.5	10.8	8.7	9.5	9.5
	0.29	0.35	0.38	0.43	0.46	0.52	0.6	0.74	0.88
	8.3	9.1	9.1	10.8	10.8	12.9	12.9	13.7	13.7
	0.9	1.02	1.12	1.14	1,15				
	13.7	16.6	17.0	16.6	18.2				

Table 9 : Temperature step response of the polymer melt chamber

(step input 100 V)

Time minute	0	4	7	9	11	14	17	20	23	28	34
Temperature °C	22	23	25	27	30	35	39	44	47	54	61
	39	46	52	58	65	72	83	92	103	113	120
	65	71	76	80	84	88	93	96	100	103	105
	145	162	180	190							
	112	114	114	114							

Table 10: Closed-loop step response of the polymer melt chamber

Time minute	0	3	5	7	11	13	18	20	22	24
Temperature °C	21	22	29	39	51	56	66	69	72	75
	28	34	38	46	60					
	79	83	86	90	90					

Table 11: Speed step response of the hydraulic drive system

Time second	0	1	2	3	4	5	6	7	8	9
Speed ms^{-1}	0.24	0.37	0.58	0.73	0.77	0.79	0.77	0.75	0.76	0.76
	10	11	12	13	14	15	16	17	18	19
	0.77	0.77	0.78	0.78	0.79	0.79	0.78	0.78	0.78	0.78
	20	21	22	23						
	0.79	0.79	0.78	0.78						

Table 12: Sampled data / output data of the microcomputer

Sampled data	0	8	16	24	32	48	64	80	96	112
Output data	0	2	11	12	17	32	47	65	82	104
	128	144	160	176	192	208	216	224	232	240
	125	143	162	180	198	215	223	230	239	247
	248	255								
	253	255								

Table 13: Open-loop step response of the speed control sysetm

Time ms	0	12	24	36	48	60	72	84	96	108
Speed ms^{-1}	0	0	0.05	0.14	0.26	0.36	0.43	0.49	0.54	0.59
	120	132	144	156	168	180	192	204	216	228
	0.62	0.64	0.66	0.68	0.69	0.71	0.72	0.73	0.74	0.75
	240	252	264	276	288	300	312	324	336	348
	0.75	0.76	0.77	0.78	0.78	0.79	0.79	0.79	0.79	0.80
	360									
	0.80									

Table 14: Fluctuation of the PRA readings after a rapid change in drawing speed

Time ms	0	24	48	72	96	120	144	168	192	216
PRA	2.1	18.1	3	32.1	22.8	18	12.8	13.5	13	11.6
	240	264	288	312	336	360	384	408	432	456
	12	17.9	11.3	11.1	10.6	10.4	11.3	10.9	10.4	10.4

Table 15: PRA / length of the produced wire
ELVAX 650, 150 °C

Length m	0.25	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.25	2.5
PRA	2.2	3.5	5.6	6.5	7.3	10.7	12.3	14.4	16.4	18.0

Table 16: PRA / length of the produced wire
ELVAX 650, 150 °C

Length m	0.25	0.5	0.75	1.0	1.5	2.0	2.5	3.0	3.5	4.0
PRA	1.31	2.19	2.62	4.35	5.63	6.49	9.85	10.69	12.75	14.79
	4.5	5.0								
	16.81	19.2								

Table 17 : PRA / length of the produced wire
ELVAX 650, 150 °C

Length m	0	1	2	3	4	5	6	7	8	9
PRA	0.9	2.1	3.92	5.2	7.3	9.4	10.7	12.7	13.9	15.6
	10	11	12							
	17.2	18.8	21.2							

Table 18 : PRA / length of the produced wire

WVG 23, 110°C

Length m	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
PRA	0.9	2.2	3.1	4.3	5.6	6.5	7.3	7.3	8.6	9.9	10.7

Table 19 : Open-loop step response of the temperature platform

Time minute	0	1	2	3	4	5	7	9	11	13	15
Temperature°C	22.0	26.4	30.3	33.9	37.4	40.6	46.3	51.2	55.5	60.0	62.1
	17	19	21	23	25	27	31	34	37	41	45
	64.8	66.9	68.9	70.5	71.8	72.8	74.7	75.7	76.4	77.5	78.0
	50										
	78.0										

Table 20 : system output of the temperature platform

at commissioning stage

Time minute	0	5	10	15	20	22	25	30	35	40	42
Temperature°C	26.4	40.3	51.8	59.8	61.0	56.4	49.4	41.0	35.0	30.7	29.5
	43	45	50	55	60	62	67				
	29.0	34.0	48.0	57.4	63.8	65.7	54.0				

Table 21 : Closed-loop step response of the temperature platform

Time minute	0	0.5	2.5	4.5	6.5	8.5	10.5	12.5	14.5	16.5	18.5
Temperature °C	25.8	26.4	32.5	39.1	44.7	49.3	53.6	57.0	59.1	60.4	61.0
	20.5	22.5									
	61.2	61.2									

THE OPTIMISATION OF A PLASTO-HYDRODYNAMIC WIRE-DRAWING PROCESS

G. R. Symmons†, M. S. J. Hashmi* and Y. D. Xie†

† Sheffield City Polytechnic, England
* N.I.H.E. Dublin, Ireland.

ABSTRACT

Non linear equations have been formulated for the pressure and axial stress increment in a conical orifice filled with a viscous fluid through which a circular cross section solid continuum is being pulled. The viscous fluid was assumed to be non Newtonian, where the viscosity is dependent upon pressure and the continuum was assumed to be rigid nonlinearly strain hardening.

A finite-differences numerical technique has been applied to solve these equations to enable predictions to be made of the non linear deformation profile of the continuum. Computer results on the effects of changes in geometry of the conical orifice on the magnitude and extent of deformation of the wire are presented in graphical form.

KEYWORDS

Plasto-hydrodynamics, wire-drawing, die-less, non Newtonian fluids, non linear strain hardening.

INTRODUCTION

In conventional drawing of wire, the diameter is reduced by pulling it through a reduction die, resulting in wear and time dependent size of product. A developed technique (1,2) enables the wire diameter to be reduced without using any conventional reduction die. In this process the wire is pulled through a tubular orifice of tapered bore which is filled with a viscous fluid. An important feature of the process is that the smallest bore size of the orifice is always greater than the diameter of the undeformed wire and hence no metal to metal contact takes place.

The pulling action of the wire through the viscous fluid gives rise to a drag force and generates hydrodynamic pressure on the wire. The magnitude of the combined drag force and pressure generated depend upon the type of fluid and geometric configuration of the orifice. By design and optimisation the combined drag force and pressure initiates plastic yielding and permanent deformation to the wire.

In order to optimise the process to give its full potential it is necessary to analyse the mechanics of the deformation process inside the tubular process. In references (3,4) closed form analytical solutions were presented based upon predetermined deformation profiles. An improved method of determining deformation values using a finite-difference technique was presented in (5), assuming a Newtonian fluid with a rigid nonlinearly strain hardening wire.

High pressures are generated in an optimised condition to cause deformation of the wire such that the effect of pressure on the viscosity of the fluid becomes significant. The present analysis uses a finite-difference technique assuming a non Newtonian fluid dependent upon pressure with a rigid non-linearly strain hardening wire. The results enable the geometry of the orifice to be optimised for maximum reduction in area of the wire.

ANALYSIS

The analysis is based on the geometrical configuration shown in Fig. 1a. where a round continuum, diameter D_1 is pulled through a tapered orifice filled with a fluid, the gap at any point is given by

$$h = h_1 - kx$$

where

$$k = (h_1 - h_2)/B$$

In order to establish a mathematical formulation of the process the following assumptions are made:

(i) the fluid has a viscosity which is independent upon shear stress but dependent upon pressure

$$\mu = \mu_o e^{ap} \quad \text{and} \quad \tau = \frac{\partial u}{\partial y}$$

(ii) flow is axial and laminar

(iii) the thickness of the fluid layer is small compared to the bore of the orifice, and

(iv) the pressure in the fluid is uniform in the thickness direction at any point along the length of the orifice, $\mu = \mu(x)$.

Considering the hydrodynamics of the fluid

$$\frac{dp}{dx} = \frac{\partial \tau}{\partial y} \quad \text{for equilibrium}$$

hence the velocity distribution in the gap is given by

$$u = \frac{1}{2\mu}\left(\frac{dp}{dx}\right)(y'-hy) + V(1-y/h)$$

for the boundary conditions u = o at y = h and u = V and y = o.

The flow of the fluid in the direction of pull is

$$Q_x = \int_o^h u\,dy = -\frac{h'}{12\mu}\left(\frac{dp}{dx}\right) + \frac{Vh}{2}$$

For steady state conditions

$$\frac{h'}{6\mu}\left(\frac{dp}{dx}\right) = Vh + \text{Constant}$$

boundary condition when $x = \bar{x}$, p = max,

hence constant $= -V\bar{h}$

where $\bar{h} = h_1 - K\bar{x}$

$$\therefore \frac{dp}{dx} = 6\mu V \left(\frac{1}{h^2} - \frac{\bar{h}}{h^3}\right)$$

substituting for $\mu = \mu_o e^{ap}$

$$\frac{dp}{dx} = 6\mu_o e^{ap} V \frac{1}{(h_1-Kx)^2} - \frac{\bar{h}}{(h_1-Kx)^3} \quad (1)$$

integrating and applying the boundary condition $p = o$ at $x = o$

$$p = -\frac{1}{a}\ln\left[1 - \frac{6\mu_o Va}{K}\left(\frac{1}{h_1-kx} - \frac{\bar{h}}{2(h_1-kx)^2}\right) - \left(\frac{1}{h_1} - \frac{\bar{h}}{2h_1^2}\right)\right] \quad (2)$$

Position of maximum pressure

Applying the boundary condition $p = o$ at $x = B$ where $h_1 - kx = h_2$, the right hand side of equation must be unity i.e.

$$1 - \frac{6\mu_o Va}{K}\left[\frac{1}{h_2} - \frac{\bar{h}}{2h_2^2} - \frac{1}{h_1} + \frac{\bar{h}}{2h_1^2}\right] = 1$$

$$\therefore \bar{h} = \frac{2h_1h_2}{h_1+h_2} \quad (3)$$

and

$$\bar{x} = \frac{h_1B}{h_1+h_2} \quad (4)$$

Axial stress in the wire prior to deformation

An axial stress in the wire will be produced by the shear stresses in the fluid acting on the surface of the wire.

Substituting the velocity gradient of the fluid into

$$\tau = \mu\frac{\partial u}{\partial y} \text{ gives } \tau = \frac{1}{2}\left(\frac{dp}{dx}\right)(2y-h) - \frac{\mu V}{h}$$

$$\text{at } y = o \quad \tau_x = -\frac{h}{2}\left(\frac{dp}{dx}\right) - \mu\frac{V}{h} \quad (5)$$

Substituting for $\frac{dp}{dx}$ from equation (1) into (5)

$$\tau_x = \mu_o V e^{ap}\left\{\frac{3\bar{h}}{(h_1-kx)^2} - \frac{4}{h_1-kx}\right\} \quad (6)$$

From equation (2)

$$e^{ap} = \frac{1}{1 - \frac{6\mu_o Va}{K}\left\{\left(\frac{1}{h_1-Kx} - \frac{\bar{h}}{2(h_1-Kx)^2}\right) - \left(\frac{1}{h_1} - \frac{\bar{h}}{2h_1^2}\right)\right\}} \quad (7)$$

The drag force $F = \int_o^x \pi D_1 \tau_x dx$

$$\text{Axial stress } \delta_x = \frac{4F}{\pi D_1^2} = \frac{4}{D_1}\int_o^x \tau_x dx \quad (8)$$

Substituting τ_x and e^{ap} from equations (6) and (7) respectively and integrating gives the solution δ_x for two conditions

(i) when $12\mu_o\bar{h}Va\left(\frac{1}{h_1} - \frac{\bar{h}}{2h_1^2}\right) + 2\bar{h}K - 6\mu_o Va > 0$

$$\delta_x = A_3 + A_4\left(\tan^{-1}\frac{2KA_2x + \frac{6\mu_o Va(\bar{h}-h_1)}{h_1} - 2h_1K}{\sqrt{6\mu_o VaA_1}} - \tan^{-1}\frac{\frac{6\mu_o Va(\bar{h}-h_1)}{h_1} - 2h_1K}{\sqrt{6\mu_o VaA_1}}\right) \quad (9)$$

where $A_1 = \left|12\mu_o\bar{h}Va\left(\frac{1}{h_1} - \frac{\bar{h}}{2h_1^2}\right) + 2\bar{h}K - 6\mu_o Va\right|$

$$A_2 = 6\mu_o Va\left(\frac{1}{h_1} - \frac{\bar{h}}{2h_1^2}\right) + K$$

$$A_3 = \frac{8V\mu_o}{D_1A_2}\ln\left|\left(1 - \frac{Kx}{h_1}\right)^2 - \frac{6\mu_o Va}{Kh_1^2}\left(h_1 - Kx - \frac{\bar{h}}{2} - \left(\frac{1}{h_1} - \frac{\bar{h}}{2h_1^2}\right)(h_1-Kx)^2\right)\right|$$

$$A_4 = \left(3\bar{h} - \frac{2}{\frac{1}{\bar{h}} - \frac{\bar{h}}{2h_1^2} + \frac{K}{6\mu_o Va}}\right)\frac{2}{\sqrt{6\mu_o VaA_1}}$$

and

(ii) when $12\mu_o\bar{h}Va\left(\frac{1}{h_1} - \frac{\bar{h}}{2h_1^2}\right) + 2\bar{h}K - 6\mu_o Va < 0$

$$\delta x = A_3 + A_4 - 1n\left|\frac{(2KA_2x + B_2 - B_1B_2 + B_1)}{(2KA_2x + B_2 + B_1B_2 - B_1)}\right| \quad (10)$$

where $B_1 = 2h_1K + \sqrt{6\mu_o VaA_1}$

$$B_2 = \frac{6\mu_o Va(\bar{h}-h_1)}{h_1}$$

$$B_3 = -2h_1K + \sqrt{6\mu_o VaA_1}$$

Plastic yielding

The combined effects of the axial stress and the hydrodynamic pressure can cause plastic yielding of the continuum if the yield criteria is satisfied. If the material of the continuum is assumed to be rigid non linearly strain hardening then the flow stress can be expressed as

$$Y = Y_o + A\varepsilon^n \quad (11)$$

then according to the Tresca yield criterion, plastic yielding will commence at a point x_o provided that

$$p + \delta x = Y_o$$

Once plastic yielding has commenced, strain hardening of the material will take place and yielding will continue as long as

$$p + \delta_x = Y > Y_o + A\varepsilon^n \quad (12)$$

The deformation zone

Consider the geometrical configuration shown in Fig 1b and an element of continuum being deformed in Fig 1c. A finite difference technique is adopted to express the pressure and axial stress equations between two points Δx distance apart on the deforming continuum assuming the deformation takes place linearly. It is assumed that deformation ceases at $x = \bar{x}$ where

$$\frac{dp}{dx} = 0$$

Equation (1) when expressed in finite difference form gives the pressure values at different points, distance Δx apart.

$$p_i = p_{i-1} + 6\mu_i V_i \Delta x \left(\frac{1}{h_i^2} - \frac{\bar{h}}{h_i^3}\right) \qquad (13)$$

where

$$V_i = V_{i-1} \ (D_{i-1}/D_i)^2$$

$$D_i = D_{i-1} - 2b_i\Delta x$$

$$h_i = h_{i-1} - (K-b_i)\Delta x$$

$$\mu_i = \mu_o e^{ap_{i-1}}$$

The equalibrium condition for a small element being deformed shown in Fig 1c gives

$$d\delta x = -2\frac{dD}{D}(Y + \tau_x \cot\alpha)$$

where $D = D_1 - 2bx \ \therefore \ dD = -2bdx$

and $\cot\alpha = -1/2b$

hence the axial stress in the continuum can be expressed in finite difference form

$$\delta_{xi} = \delta_{xi-1} \frac{4b_i\Delta x Y_i}{D_i} - \frac{4\Delta x\mu_i}{D_i h_i}\left\{4V_i - \frac{3V_i\bar{h}}{h_i}\right\} \qquad (14)$$

where

$$Y_i = Y_o + A\{\ln(D_1/D_i)\}^n \qquad (15)$$

Deformation of the continuum continues as the plasto-hydrodynamic compatability equation is satisfied

$$p_i + \delta_{xi} = Y_i$$

The procedure for predicting the theoretical results involved the determination of x_o the distance from the entry of the orifice where plastic deformation commences. This position satisfies the equation $p + \delta_x = Y_o$ and is found by iterative computation of equations (2), (8) and Y_o. From this point onward, the extent of the plastic deformation is known. Equations (13) and (14) are combined for small increments Δx to equate to equation (15) and the slopes b are found by iterative computation. The deformation profile and percentage reduction in area are predicted for changes in drawing speed and orifice geometry.

Notations

- B = Length of the orifice
- D = Diameter of wire
- F = Drag force
- V = Pulling velocity
- p = Hydrodynamic pressure
- τ_x = Shear stress at fluid/wire interface
- δ = Axial stress in wire
- Y = Yield stress of wire material
- h = Gap between wire and orifice
- μ = Dynamic viscosity of fluid
- x = Distance along orifice from leading edge
- y = Distance from wire surface into gap
- a = Viscosity pressure coefficient
- b = Deformation profile constant
- ε = Plastic strain in the wire
- n = Strain hardening coefficient
- A = Material constant

RESULTS AND DISCUSSION

Theoretical results were calculated on standard values of parameter obtained from experimental studies. The equations governing the deformation zone were solved at steps of

$$-\Delta x = \frac{x_o - \bar{x}}{15}$$

from the position of the onset of yield x_o.

The data below were the parameters used: length of orifice B = 45mm, inlet gap h_1 = 0.5mm, outlet gap h_2 = 0.075mm, initial diameter of wire D_1 = 1.625mm, viscosity of fluid at normal pressure μ_o = 120Ns/m², viscosity constant relating to pressure a = 2.0×10^{-8}, initial yield stress of continuum Y_o = 100MN/m², material constant A = 341MN/m² and strain hardening index n = 0.25.

Fig 2 shows the hydrodynamic pressure profile generated in the orifice when pulling the wire at 0.31m/s. The effect of non-Newtonian behaviour can be observed with the steep increase in pressure approaching the maximum at 38mm along the orifice. A maximum pressure of 160 MN/m² can be generated in the fluid by hydrodynamic means, with a shape factor K = 0.0094'.

The onset of yield in the wire begins when the sum of the hydrodynamic pressure and the axial stress equals the yield stress of the wire material. Fig 3 shows the computer results of the position of the onset of yield in the orifice. This position is dependent upon drawing speed and shape factor K and the graph shows the optimum value of K for three speeds. For the standard values of parameters stated above, a low K value in the region of 0.001 provides early deformation in the orifice.

The completion of yielding is at the maximum pressure position because there is a change of sign of the pressure gradient. The Poiseuille type flow is then in opposition to the Couette flow thus significantly reducing the shear stress together with the reduction in pressure. Fig 4 shows this position for changes in the outlet and inlet fluid gaps. A high value of K extends the maximum pressure position towards the outlet end of the orifice. There is therefore a conflict in the optimised requirements for x_o and $\bar{x}$ in terms of K. Optimisation of K to produce a long length of deformation profile is therefore a more practical way of gaining maximum reduction in area. Fig 5 shows the computer results for the length of deformation for three drawing speeds and a K value in the region of 0.002 give maximum lengths.

Deformation profiles for three drawing speeds are given in Fig 6 and a comparison of theoretical profile with an experimental profile from ref (5) is given in Fig 7. Comparison of the experimental and theoretical deformation profiles are for the same level of percentage reduction in area of 23%. Although there is a mismatch in the drawing speeds to attain a common percentage reduction in area, the difference is significantly less than the Newtonian Solution in (5).

Finally, Fig 8 shows the changes of maximum drag force F_{max}, length of deformation zone L and percentage reduction in area PRA against K. The effect of change in K is significant and optimises to a maximum of 0.0023.

CONCLUSIONS

A numerical solution has been developed to predict the deformation of wire drawn through a conical orifice filled with a non-Newtonian fluid.

For the same extent of deformation the theoretically predicted profiles show a closer agreement to observed experimental results than a previous Newtonian solution.

At drawing speeds in excess of 1m/s, the agreement between actual and predicted percentage reduction in area is poor. However at low drawing speeds, the optimisation technique described in this paper is valid for the design of the orifice in this new process.

REFERENCES

1. Hashmi, M.S.J., Symmons, G.R. and Parvinmehr, H. 'A novel technique for wire drawing' J. Mech. Eng. Sci. Vol. 24 p.1, 1982.

2. Hashmi, M.S.J. and Symmons, G.R., U.K. Patent No. 8311301, 1983.

3. Symmons, G.R., Hashmi, M.S.J. and Parvinmehr, H. 'Plasto-hydrodynamic lubrication, die-less wire drawing.' Proc. Int. Conf. on Developments in Drawings of Metals, Metals Society, London p. 54, May 1983

4. Hahsmi, M.S.J. and Symmons, G.R. 'A mathematical model for the drawing of a solid continuum through Newtonian fluid filled tubular orifice, Proc. 4th Int. Conf. on Mathematical Modelling, Zurich, August, 1983.

5. Hashmi, M.S.J. and Symmons, G.R. ' A numerical solution for the plasto-hydrodynamic drawing of a rigid non linearly strain hardening continuum through a conical orifice.' Proc. 2nd Int. Conf. on numerical methods for non linear problems, Barcelona, April 1984.

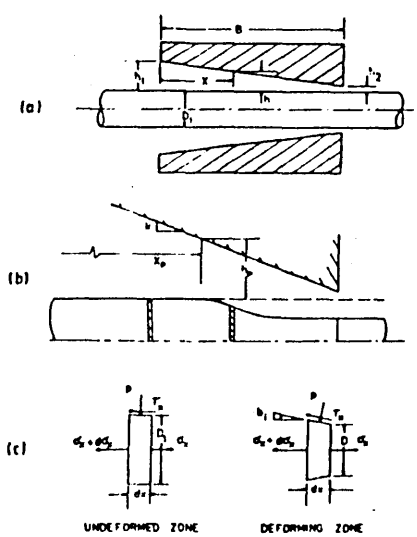


Fig.1 Geometry of the dieless reduction unit and the deformation mode

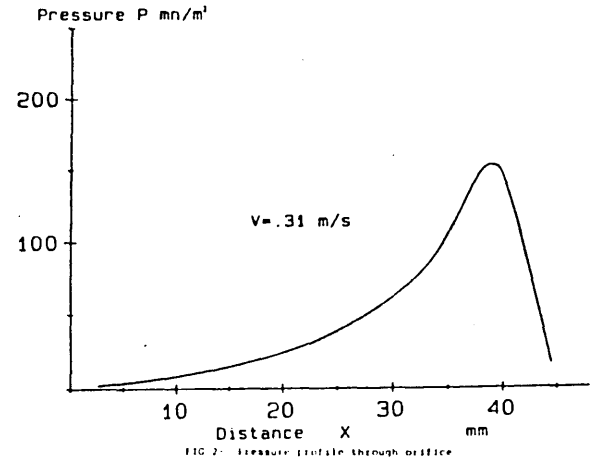


FIG 2 Pressure profile through orifice

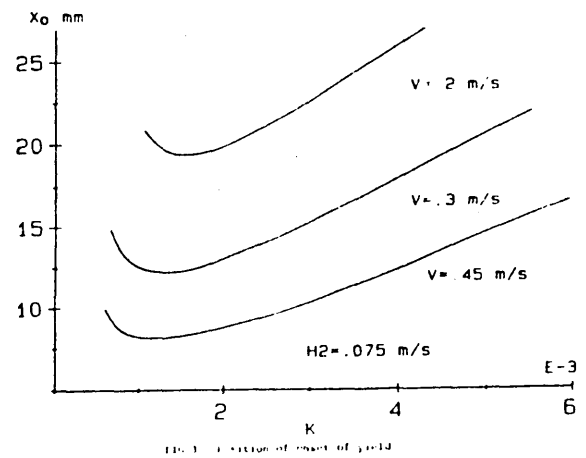


FIG 3 Position of onset of yield

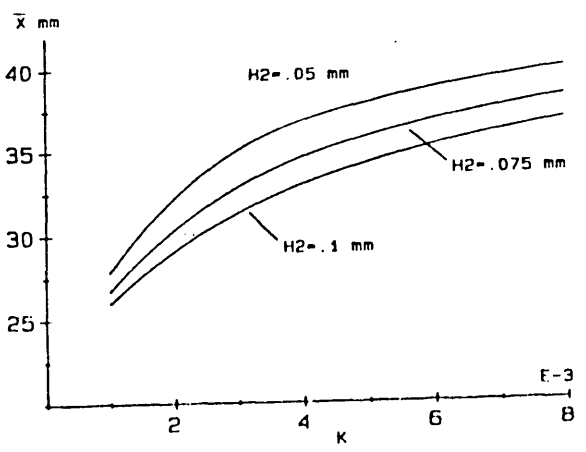
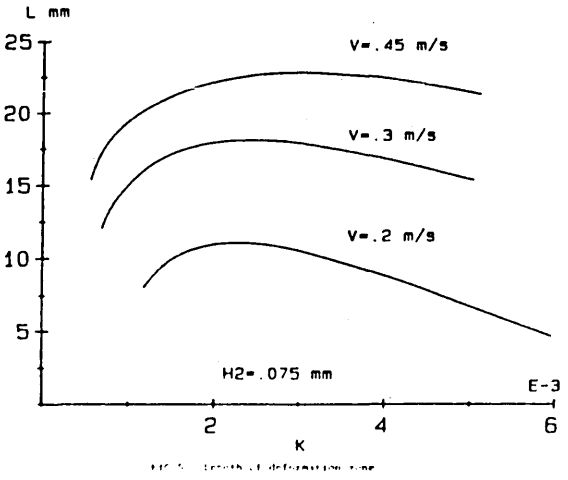


FIG 4 Position of maximum pressure



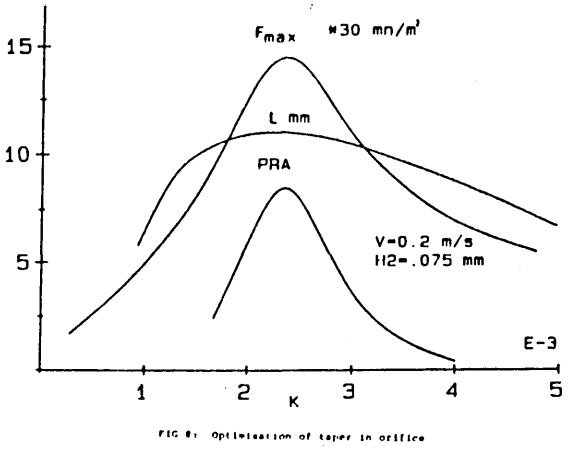


FIG 8: Optimisation of taper in orifice