# Design and Construction of an Experimental Paper Calender

by
Thomas Browne

A thesis submitted to the Faculty of Graduate Stuaies and Research
in partial fulfillment of the requirements for the
degree of Master of Engineering

Department of Chemical Engineering

McGill University

Montréal, Québec
Canada

March 1991

## ABSTRACT

An experimental paper calender was designed and built to measure paper strain in the nip of the calender as a function of machine speed, load and calender roll radius. The effect of roll deflection due to bending was eliminated by making the calender as narrow as possible while still reproducing industrial conditions in all other aspects. The width of the new machine is 75 mm.

Permanent paper strains measured with the new equipment reproduced previously published results. In-nip paper strains were obtained under a variety of calendering conditions. Paper strains under load have previously been obtained in a platen press; the new data also reproduce these results.

The equipment can now be used for a full experimental program to determine an empirical relationship between paper stress (in terms of the calendering variables) and strain (in terms of paper thickness in the nip).

## RESUME

On a conçu et construit une calandre expérimentale ayant pour but de mesurer la déformation du papier dans la pince en fonction de la vitesse, de la charge et du rayon du rouleau. On a éliminé l'effet de fléchissement du rouleau en réduisant au minimum la largeur de la calandre, tout en reproduisant les conditions industrielles à tout autre égard. La largeur de la nouvelle machine est de 75 millimètres.

La déformation permanente du papier, mesurée au moyen du nouvel équipment, correspond aux résultats déjà publiés. La déformation du papier dans la pince fût obtenu pour plusieurs conditions de calandrage. Ces données reproduisent les résultats obtenus pour la compression du papier dans une presse.

L'équipment peut maintenant servir pour un programme expérimental complet visant à déterminer une relation empirique entre l'effort de compression (en fonction des variables du calandrage) et déformation qui en résulte (en fonction de l'épaisseur du papier dans la pince).

## ACKNOWLEDGEMENTS

First and foremost, I would like to acknowledge the continuous support, both technical and financial, of my advisors, Dr. R.H. Crotogino and Dr. W.J.M. Douglas.

I would also like to acknowledge the assistance of the design office staff at the Pulp and Paper Research Institute of Canada, in particular R. Stone, whose experience in design and operation of experimental paper machines was invaluable in guiding the early stages of this work, and J.C. Béchet and J.C. Adrien, without whose experience in the machine shop the equipment described here could not have been built.

The staffs of the mechanical and electronics shops of the departments of Chemical and Mechanical Engineering at McGill University also deserve mention for their unfailing helpfulness in the construction phase of this project.

Finally, I wish to thank my wife, Mary, and my children, Tara and Alexander, for their patience and understanding during the time spent on this thesis.

## CONTENTS

FIGURES

## TABLES

## Nomenclature

| | |
|---|---|
| $A$ | Calendering equation intercept |
| $a_1$ | Calendering equation intercept |
| $a_i$ | Fourier amplitude |
| $a_L$ | Calendering equation load constant |
| $a_M$ | Calendering equation moisture constant |
| $a_R$ | Calendering equation radius constant |
| $a_S$ | Calendering equation speed constant |
| $a_T$ | Calendering equation temperature constant |
| $a_V$ | Experimental speed constant |
| $B_f$ | Final paper specific volume or bulk, $cm^3/g$ |
| $B_i$ | Initial paper specific volume or bulk, $cm^3/g$ |
| $B_n$ | In-nip paper specific volume or bulk, $cm^3/g$ |
| $e_v$ | Eccentricity of lower roll with paper, $\mu m$ |
| $e_{vo}$ | Eccentricity of lower roll without paper, $\mu m$ |
| $k_i$ | Fourier index |
| $L$ | Load, kN/m |
| $M$ | Moisture content, % dry basis |
| $N$ | Number of terms in sum |
| $P_{max}$ | Maximum in-nip pressure, Pa |
| $R$ | Radius, m |
| $S$ | Sheet speed, m/min |
| $T$ | Temperature, °C |
| $t$ | Time, s |
| $t$ | In-nip or final paper thickness, $\mu m$ |
| $t_i$ | Initial paper thickness, $\mu m$ |
| $t_n$ | In-nip paper thickness, $\mu m$ |
| $t_f$ | Final paper thickness, $\mu m$ |
| $t_u$ | In-nip paper thickness due to upper roll, $\mu m$ |
| $t_v$ | In-nip paper thickness due to lower roll, $\mu m$ |
| $u_r$ | Upper rim displacement with paper, $\mu m$ |
| $u_{ro}$ | Upper rim displacement without paper, $\mu m$ |
| $u_s$ | Upper shoulder displacement with paper, $\mu m$ |
| $u_{so}$ | Upper shoulder displacement without paper, $\mu m$ |
| $v_r$ | Lower rim displacement with paper, $\mu m$ |

| | |
|---|---|
| $v_{ro}$ | Lower rim displacement without paper, μm |
| $v_s$ | Lower shoulder displacement with paper, μm |
| $v_{so}$ | Lower shoulder displacement without paper, μm |
| $\delta u$ | Upper shoulder to rim distance with paper, μm |
| $\delta u_o$ | Upper shoulder to rim distance without paper, μm |
| $\delta v$ | Lower shoulder to rim distance with paper, μm |
| $\varepsilon_p$ | Permanent strain, $(B_i-B_f)/B_i$ |
| $\varepsilon_n$ | In-nip strain, $(B_i-B_n)/B_i$ |
| $\mu$ | Nip intensity factor, defined by Equation 2.2.2 |
| $\psi$ | Phase shift between readings, rad |
| $\phi_i$ | Fourier phase shift, rad |
| $\omega$ | Frequency, rad/s |

# 1. INTRODUCTION

## 1.1 BACKGROUND

The final step in the manufacture of many grades of paper is calendering. This process, illustrated schematically in Figure 1.1.1, reduces the thickness and roughness of a paper sheet by pressing it between two or more heavy cast iron rolls.

The relationships between thickness reduction and paper surface properties such as roughness and gloss have been well documented by de Montmorency [1] and Crotogino [2]. In general, calendering reduces both thickness and roughness of paper and improves its printing properties. Thus any change in thickness reduction can be related to changes in paper properties by referring to published work. The focus of this research is thickness reduction; the corresponding changes in paper properties will therefore not be discussed.

Paper thickness during and after calendering depends on basis weight, defined as the mass per unit area, and on the calendering parameters. Since this research is primarily concerned with the effect of the calendering parameters, the effect of basis weight will be eliminated by using the specific volume of paper. Also known as bulk, specific volume is defined as thickness divided by basis weight.

The bulk of a sheet after calendering depends on several process variables. The load applied to the rolls, the machine speed and the roll radius determine the magnitude and duration of the pressure pulse applied to the paper in the nip. These machine-dependent variables, along with the number of successive compressions (or nips), determine the work done

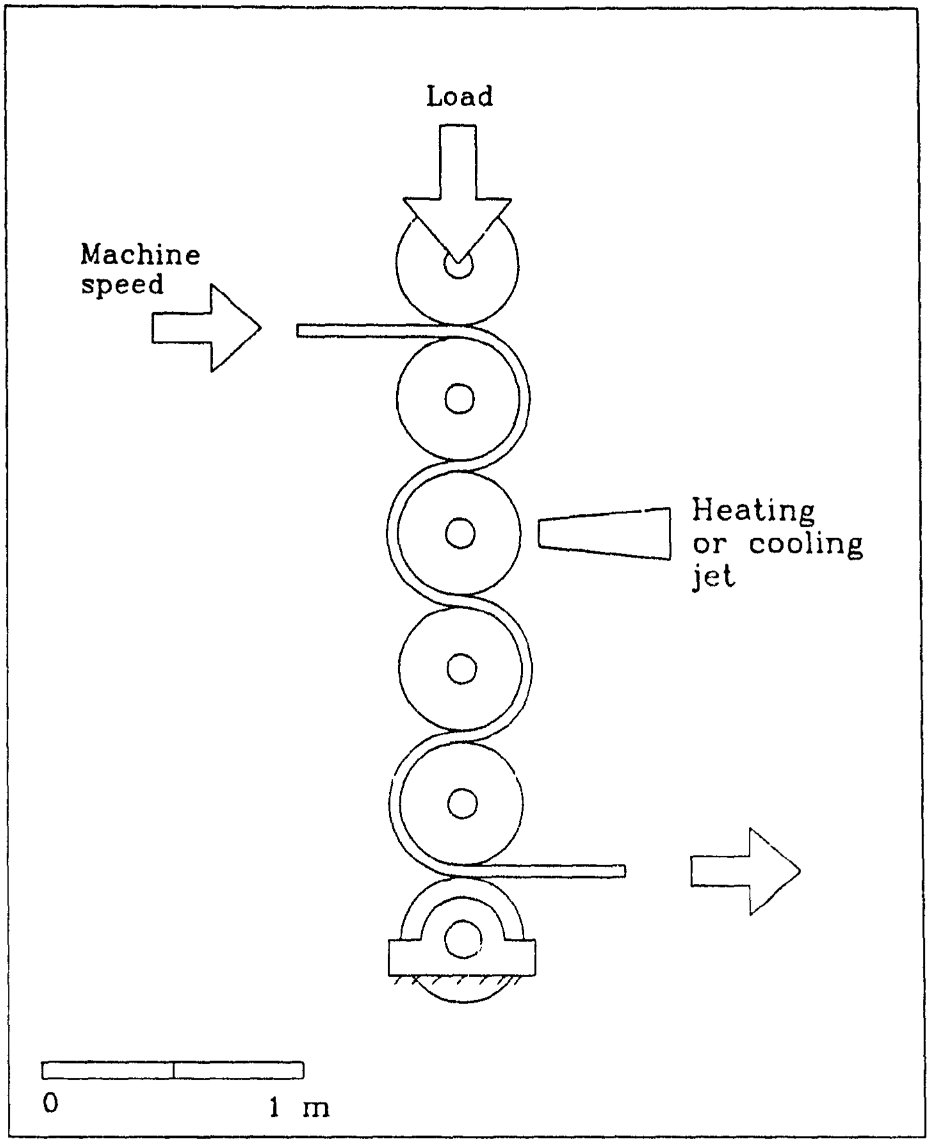



Figure 1.1.1: A typical calender stack

on the paper.

The quality of the final product is also affected by initial paper properties such as bulk, temperature (which may be altered by contact with a heated roll) and moisture content. These paper-dependent variables determine the response of the paper to the applied pressure pulse. Crotogino [14] has proposed the calendering equation, which describes the relationship between the average bulk reduction in a calender and the various machine and paper variables.

Initial paper properties vary in both the machine direction and cross direction due to variations introduced in previous steps of the papermaking process. Only cross-direction variations are of interest in this thesis; machine-direction variations will not be discussed since these are dependent on average calendering parameters.

Sheet temperature and moisture content can vary across the width of a paper machine, causing variations in roll temperature which in turn cause local changes in roll radius due to thermal expansion. Along with variations in initial bulk, these thermal deformations result in variations in the load distribution. At steady state, in other words, variations in the initial paper properties can cause variations in all calendering variables except machine speed.

These variations may be substantial in industrial-scale paper machine calenders, which can have rolls as wide as 10 meters. Cross-direction variations cannot be taken into account using the calendering equation since local radius and nip load values cannot be measured. In order to determine the local bulk reduction or the final bulk profile using the calendering equation, either the load distribution or the radius distribution and its effect on the load must be known.

It should be noted that while a typical commercial roll has a radius of the order of 500 mm, paper thickness in the nip is of the order of tens of micrometers. A change in roll radius of 5 μm, which can easily be caused by a roll surface temperature change of only 5 °C [5], is significant compared to the paper thickness and thus has a large effect on the nip load and a correspondingly large effect on final bulk.

It is desirable to control the local roll radius, and thus the local shape of the pressure pulse, in order to control the final bulk. The method most commonly used today is to heat the rolls locally where greater bulk reduction is required and to cool them for less bulk reduction. The radius profile resulting from a given temperature profile can be calculated, thanks to previous work by Pelletier et al. [3, 4] and Journeaux [5]. Their work describes local heat transfer rates to a roll from an impinging air jet, the resulting heat distribution in the roll, and the local roll deformation due to this internal temperature distribution.

In other words, it is possible to calculate the roll temperature profile required, and the local control actuator settings required to arrive at that profile, in order to create a desired radius or nip gap profile. However, it is not currently possible to determine the radius profile required to create a desired bulk profile since the rheological properties of paper in a rolling nip are not known. A relationship is required between stress (in the form of a pressure distribution) and strain (in the form of a nip gap distribution) for paper in a calender nip. These relationships are the final elements required for a complete explicit description of cross-direction calender control.

## 1.2 Objectives

The objectives of this project are therefore

– to build an experimental calender enabling accurate measurement of the local in-nip paper thickness during calendering while reproducing as closely as possible the conditions in a commercial paper machine, and

– to demonstrate the correct functioning of the equipment by performing an initial set of measurements designed to reproduce the cross-machine results obtained by Crotogino et al. [14, 15].

# 2. REVIEW OF PREVIOUS WORK

## 2.1 CALENDERING LITERATURE REVIEW

Paper is subjected to a pressure pulse in a calender nip. Since paper behaves like a viscoelastic solid, it is compressed by the pulse. When the pulse ends, the paper exhibits time-dependent partial recovery of the initial thickness. There is usually some permanent deformation, which is the desired outcome of the calendering process.

Chapman and Peel [6] investigated the effect of a pressure pulse on compressed and recovered paper thicknesses using a platen press. The pressures applied were fairly constant for the duration of the tests, and so their work did not duplicate the approximately parabolic nature of the pulse in a rolling mill. Figure 2.1.1 shows typical pressure and thickness versus time curves for their experiments. From this data they were able to derive relationships for compressed and recovered paper thicknesses, $t_n$ and $t_r$, as a function of maximum pressure $P_{max}$ and pulse duration.

The duration of a typical pressure pulse was long (the shortest pulse lasted 6 ms) compared to the duration of the pulse in a commercial calender (in the hundreds of microseconds). The relationships derived in their work are therefore related to the creep characteristics of paper; extrapolation to very short pulse durations was not performed. Furthermore, the effect of successive compressions was not analyzed. The work was later extended by Colley and Peel [7] to include web temperature and moisture effects.

Kerekes [8, 10] modified these empirical relationships to predict directly the thickness reduction in a rolling nip in

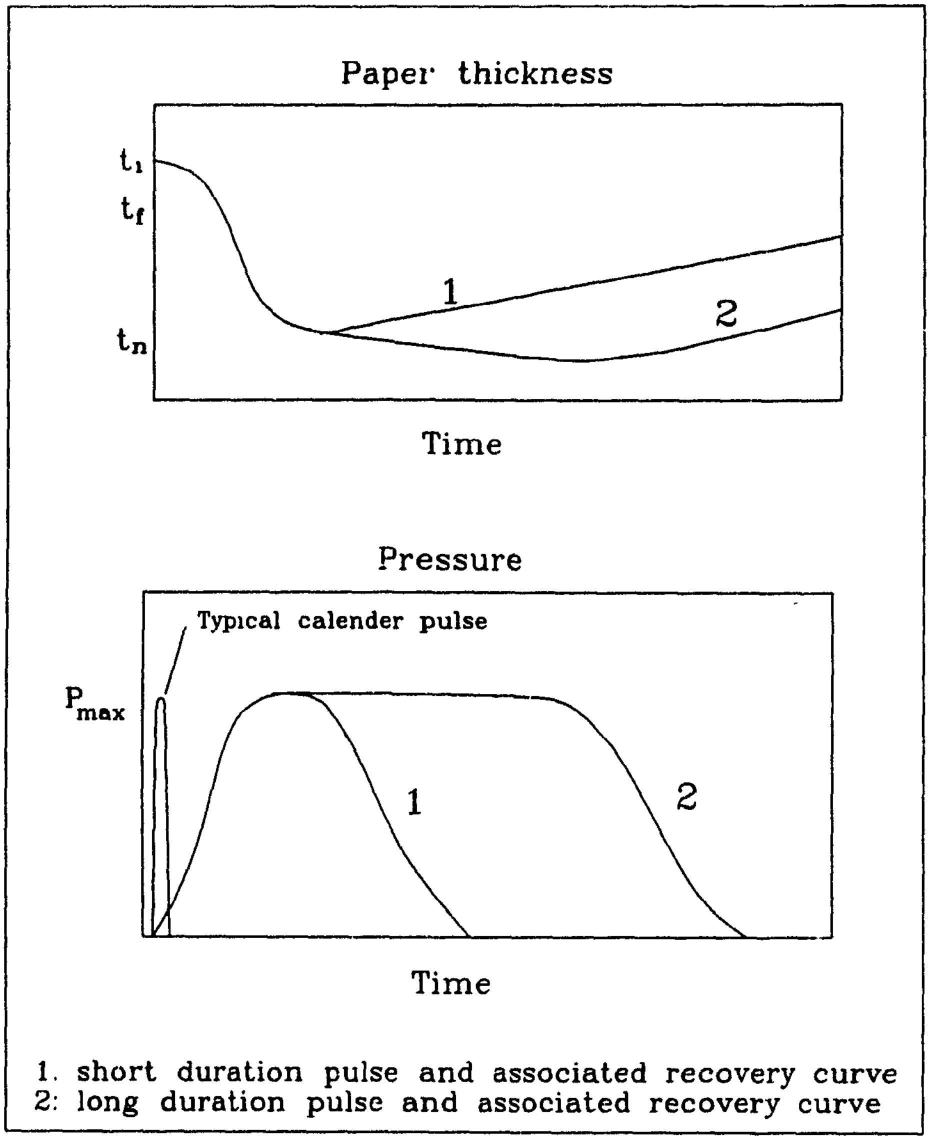



Figure 2.1.1: Typical data from Chapman and Peel [6].

terms of the machine variables nip load, roll radius and machine speed, whereas Peel et al. [6, 7] defined it in terms of maximum pressure and pulse duration. Kerekes [8] also verified these predictions using a laboratory-sized model of an actual paper machine calender run at speeds approaching those typical of commercial equipment. As a result, the equipment approximated the shape and duration of the pressure pulse in an industrial calender more closely than did the platen press used by Peel et al.

The shape of the pressure pulse in a rolling nip was predicted by Kerekes [10] to be approximately parabolic with a certain amount of skew due to the time-dependent response of paper. This work was based on earlier work by May et al. [11], Hunter [12] and Alblas and Kuipers [13]. All of these analyses are based on modeling paper using the standard linear solid, which is made up of a spring in parallel with a Maxwell model. Inherent in this choice of model is the assumption that the permanent thickness reduction, or strain, imposed on the paper is negligible compared to the initial thickness, and can therefore be neglected. In fact, the permanent strain is often 20% or more [7, 9], and thus the standard linear solid only approximates the true behaviour of paper in a rolling nip.

Haglund and Robertson [9] used optical methods to measure paper thickness in the nip of a small laboratory calender run at low speeds. In spite of systematic errors in the experimental procedure, they found (in agreement with Colley and Peel [7]) that initial thickness had no effect on the percent reduction in bulk in a calender, but that paper with a low initial bulk showed less reduction than a bulkier sheet.

Crotogino et al. [14, 15] extended Colley and Peel's work [7] to include the effects of initial bulk; their calendering equation is an empirical relationship giving the average recovered paper thickness as a function of nip load, sheet speed, effective roll radius and initial sheet properties. Since the effect of initial bulk is taken into account, the calendering equation may be used repeatedly for successive nips, with the computed final bulk from one nip used as the initial bulk for the next.

Hamel et al. [16] used the calendering equation to calculate the nip load distribution from the recovered thickness profiles of calendered paper. This reverse use of the equation only yields information about local calendering conditions after the paper has been calendered. However the paper thickness in the nip is still not known since the paper response to the nip load distribution is not known.

## 2.2 THE CALENDERING EQUATION

Much of the work of preceding authors, summarized in the calendering equation proposed by Crotogino et al. [14, 15], provides the basis for the present work. Therefore, a detailed description of the calendering equation is in order.

Permanent relative bulk reduction $\varepsilon_p$ is a measure of strain and is defined by

$$\varepsilon_p = \frac{B_i - B_f}{B_i} \qquad [2.2.1]$$

where $B_i$ and $B_f$ are the initial and final bulk, respectively. The calendering equation is an empirical relationship between $\varepsilon_p$ and the nip load L, machine speed S, effective roll radius R, paper temperature T and paper moisture content M:

$$\varepsilon_p = A + \mu B_i$$

where

$$\begin{aligned} \mu = a_1 &+ a_L \log L \\ &+ a_S \log S \\ &+ a_R \log R \\ &+ a_T\, T \\ &+ a_M\, M \end{aligned} \qquad [2.2.2]$$

The constants A, $a_1$, $a_L$, $a_S$, $a_R$, $a_T$ and $a_M$ are all dependent on pulp type and paper structure and must be determined experimentally for each type of paper. The equation, illustrated in Figure 2.2.1, is valid over a range of $B_i$ defined by

$$\frac{-A}{\mu} \le B_i \le \frac{1 - A}{2\mu} \qquad [2.2.3]$$

The lower limit is the point beyond which no further bulk reduction is likely without increasing the nip intensity $\mu$:

$$B_f = B_i \qquad \text{when } B_i \le \frac{-A}{\mu} \qquad [2.2.4]$$

Below this limit, the calendering equation predicts $B_r > B_i$, which is not generally the case. An accurate model of the calendering process should approach the line $B_r = B_i$ asymptotically. The calendering equation does not do so because it is an empirical approximation of physical reality.

The upper limit is the point beyond which an increase in initial bulk has no effect on bulk reduction. This occurs when a very bulky paper is subjected to a very high nip intensity factor $\mu$:

$$B_r = \frac{[1 - A]^2}{4\mu} \qquad \text{when} \quad B_i \geq \frac{1 - A}{2\mu} \qquad [2.2.5]$$

This is the point where $B_r$ reaches a maximum:

$$\left. \begin{aligned} B_i &= \frac{1 - A}{2\mu} \\ B_r &= \frac{[1 - A]^2}{4\mu} \end{aligned} \right\} \quad \text{when} \quad \frac{dB_r}{dB_i} = 0 \qquad [2.2.6]$$

Mathematically, the upper limit on states that there cannot be two values of $B_i$ resulting in the same $B_r$; the relationship between $B_r$ and $B_i$ is approximately parabolic only as long as the derivative is non-negative.

Thus the calendering equation provides an empirical relationship between machine and paper variables on the one hand and the recovered paper bulk on the other. The curve-fitting constants are material properties which must be determined experimentally for each pulp type. The predicted final bulk is the average of the cross-machine bulk profile, since both the nip gap and pressure distributions are difficult to measure.

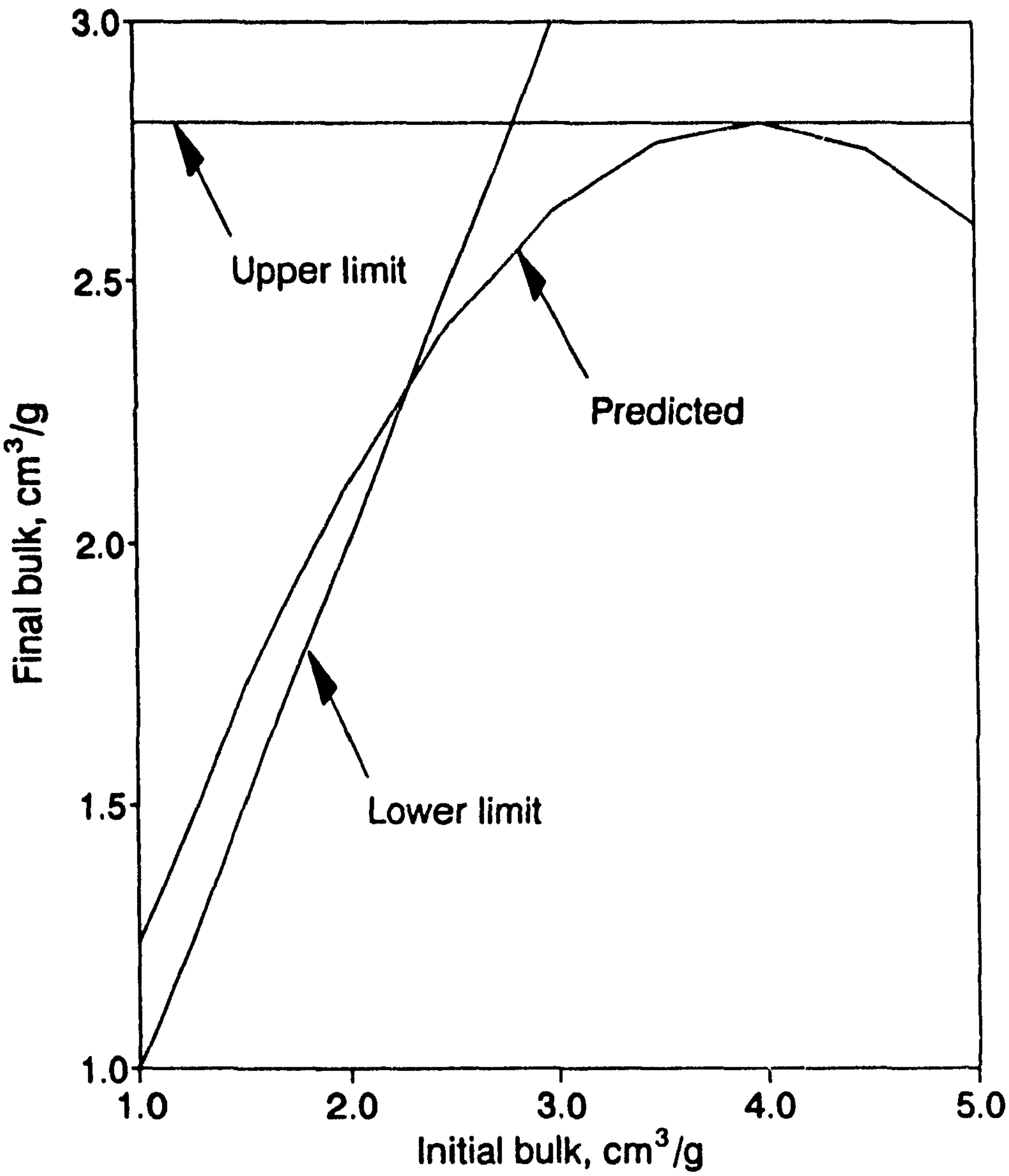



Figure 2.2.1: Effect of initial bulk on final bulk, as predicted by the calendering equation.

## 2.3 EFFECT OF CALENDERING VARIABLES ON PAPER PROPERTIES

The physical response of paper to changes in the various calendering parameters is illustrated next using the calendering equation with a typical set of parameters, listed in Table 2.3.1.

The effect of nip load, shown in Figure 2.3.1 for a single-nip calender under typical industrial conditions, is to reduce recovered paper bulk. A larger load lengthens the nip slightly by pressing the roll further into the paper; thus both the magnitude and the duration of the pressure pulse are increased, the paper is deformed to a greater extent and the bulk reduction, or permanent strain, is greater.

Varying the machine speed (Figure 2.3.1) has the opposite effect. At higher speeds the dwell time is decreased while the magnitude of the pressure pulse is constant. The result is less bulk reduction.

Changes in roll radius (Figure 2.3.2) influence recovered bulk in two ways. If the radius is changed by installing a larger roll, the nip load will also increase unless the new roll weighs the same as the old one. In addition, a larger radius due to either a new roll or thermal expansion will cause the load to be distributed over a longer nip. If the radius is increased independently of load (by using a hollow roll to keep roll weight constant), the maximum pressure will decrease while the dwell time increases. The effect of reduced pressure is greater than that of increased dwell time, and so the net result of increasing roll radius while maintaining a constant roll weight is slightly less bulk reduction.

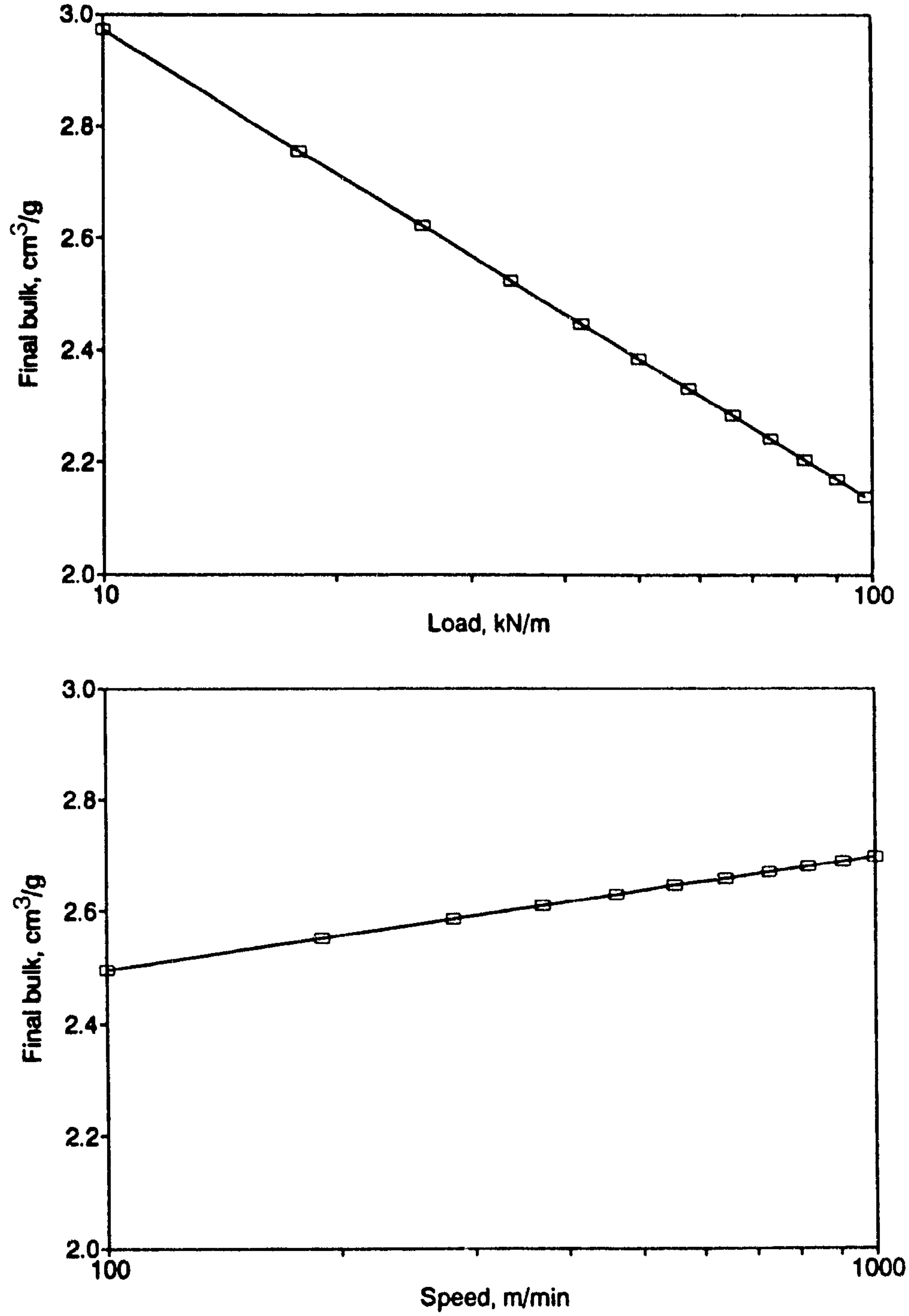



Figure 2.3.1: Effect of load and speed on final bulk, as predicted by the calendering equation.

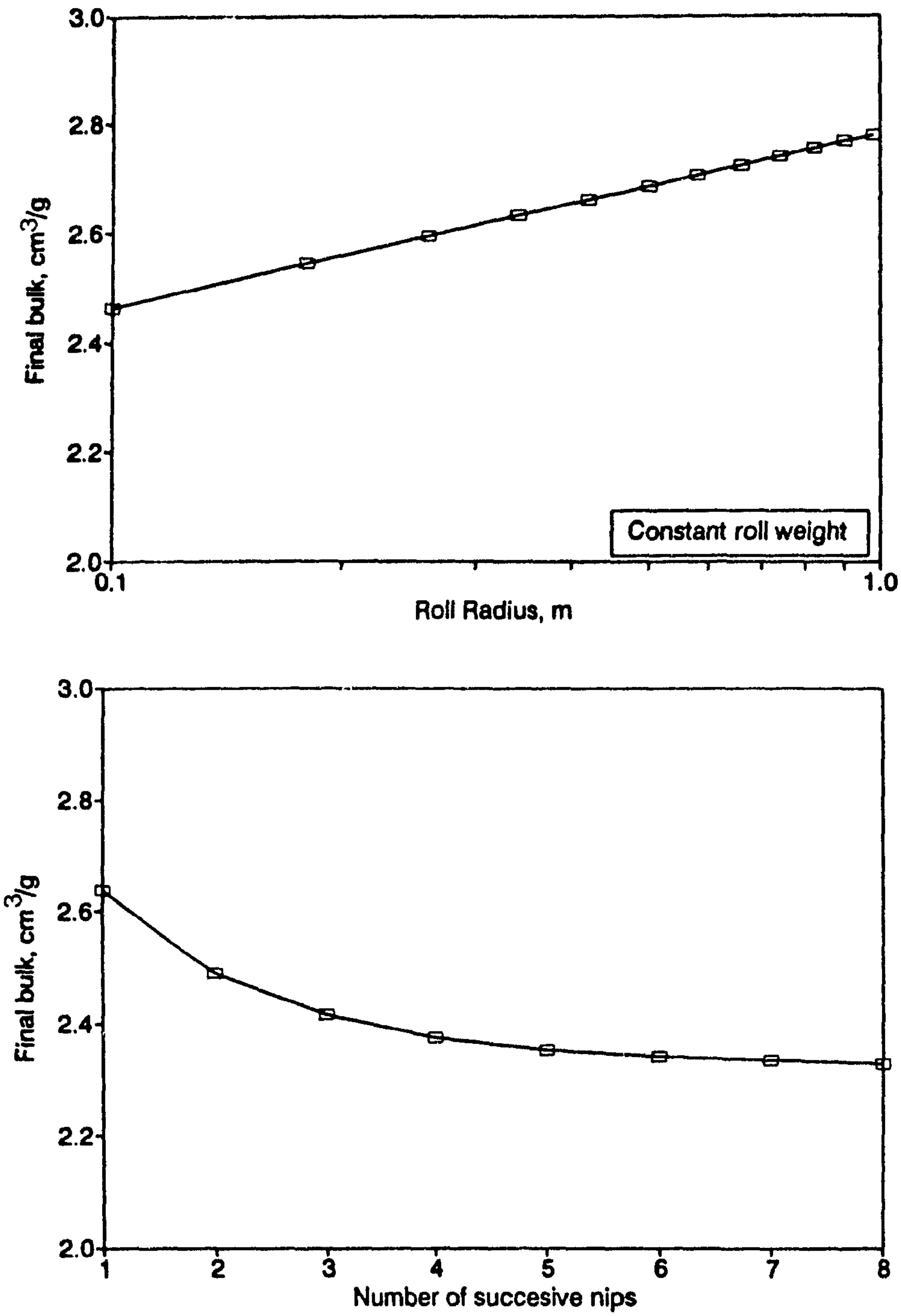



Figure 2.3.2: Effect of roll radius and multiple nips on final bulk, as predicted by the calendering equation.

Bulk reduction as a function of the number of nips is shown in Figure 2.3.2. If all other conditions are exactly the same in each successive nip, bulk reduction will be less in later nips than in earlier ones, due to lower initial bulk.

The effect of initial paper variables on recovered bulk is shown in Figure 2.3.3. If the paper is heated before entering the nip, the fibres in the sheet become more pliable, resulting in greater plastic deformation of the sheet and correspondingly lower final bulk.

Temperature and nip load are the two variables with the greatest effect on final paper properties. They are also the easiest to control while the paper machine is running. Thus a desired change in bulk reduction can be achieved by changing either the load, the temperature, or a combination of the two. Since paper strength is reduced in calendering at high loads, a higher temperature and lower load can result in a stronger paper for a desired bulk reduction.

Increased initial moisture content also makes the fibres more pliable, leading to greater permanent deformation. However, as shown in Figure 2.3.3, this effect is reversed at high moisture content, where the amount of water in the sheet begins to hinder further compression.

Table 2.3.1: Calendering variables illustrating Equation 2.2.2 [15]

| | | | |
|---|---|---|---|
| A | -0.415 | | |
| $a_o$, $g/cm^3$ | 0.0212 | | |
| $a_L$, $g/cm^3$ | 0.0935 | Load, kN/m | 25 |
| $a_s$, $g/cm^3$ | -0.0226 | Speed, m/min | 500 |
| $a_R$, $g/cm^3$ | -0.0355 | Roll radius, m | 0.35 |
| $a_T$, $g/cm^3C$ | 0.00078 | Temperature, C | 35 |
| $a_M$, $g/cm^3$ | 0.0068 | Moisture content | 6.5% |

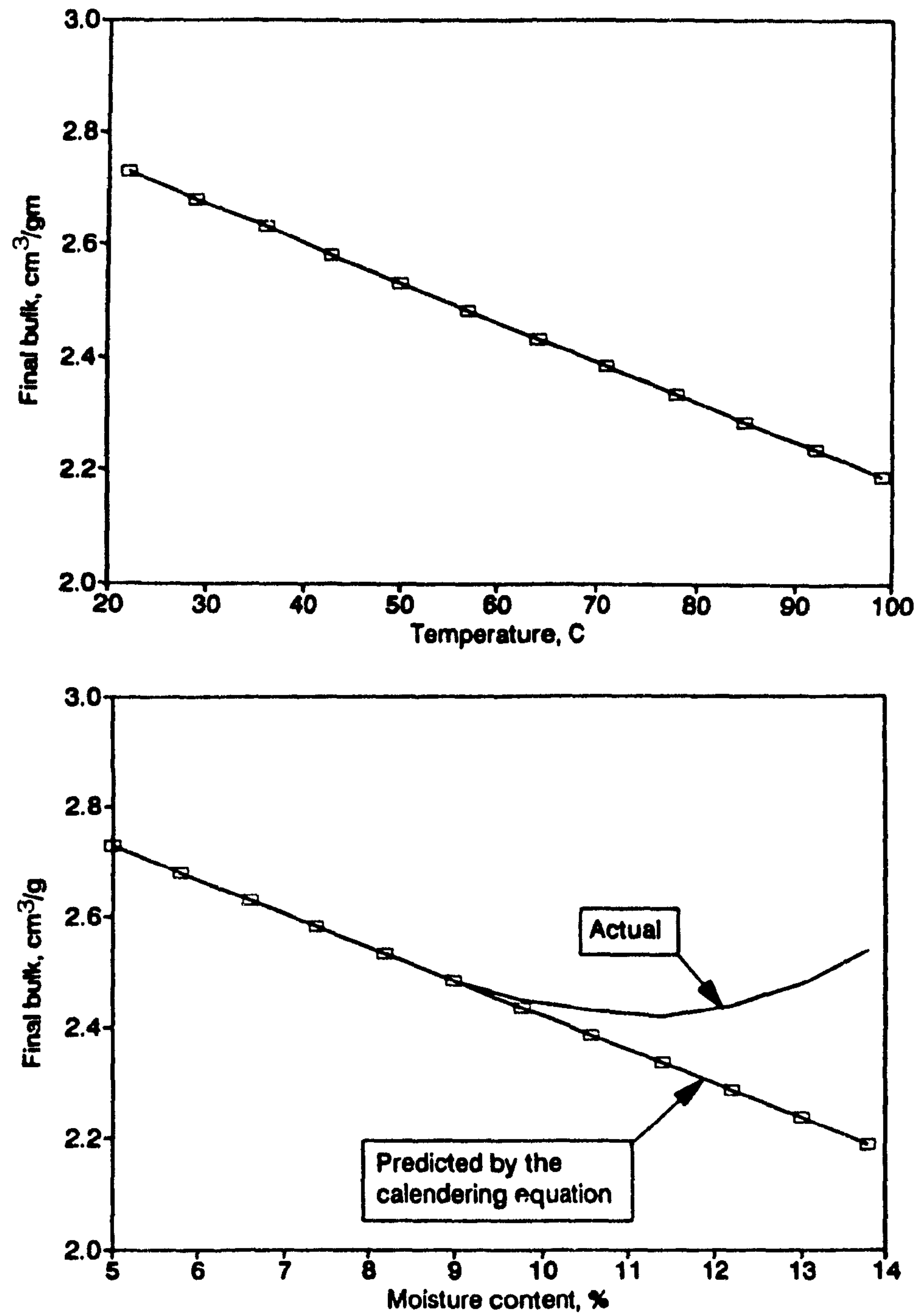



Figure 2.3.3: Effect of paper temperature and moisture on final bulk, as predicted by the calendering equation.

## 2.4 Conclusion

Paper is pressed in a calender, where it is deformed to an extent which depends on the calendering parameters. The calendering equation predicts only the average permanent deformation in the cross-machine direction.

Local control of paper deformation is achieved by heating or cooling local areas of the roll, which alters the local radius profile and thus the local pressure distribution. The amount of heating or cooling required to generate a desired local radius profile is known [3, 4, 5], but the local pressure distribution and thus the bulk reduction profile resulting from that radius profile are not known since the material properties of paper in a rolling nip are not known.

Similarly the local pressure distribution can be determined from the recovered paper thickness profile using the calendering equation [16], but the relation between load distribution and roll radius profile, which depends on the material properties of the paper, is not known. The amount of heating or cooling necessary for a given recovered thickness profile is therefore not known.

The goal of the present work is to take the next step towards the design of a closed-loop control system for local bulk reduction in a calender nip. The unknown quantity is the relationship between local pressure and local in-nip bulk, which depends on the viscoelastic properties of paper. The required next step, described in the next chapters, is the design and construction of equipment enabling the simultaneous measurement of local pressure and local in-nip paper thickness in a calender nip.

# 3. Equipment design

## 3.1 Introduction

In order to develop an in-nip version of the calendering equation, it was necessary to design and build a new laboratory calender specifically for accurate and repeatable measurement of in-nip paper thickness as a function of the key parameters, machine speed, nip load and roll radius.

The design and construction of the equipment is the major portion of the work reported in this thesis and is therefore described in detail in the following sections.

## 3.2 DESIGN CRITERIA

To meet the goal set forth in the last section, a list of minimum design criteria was made. The list included the following minimum objectives:

1. accurate pressure profile measurement;
2. accurate in-nip paper thickness measurement;
3. accurate initial and final caliper measurements;
4. control and measurement of nip loads from 20 to 200 kN/m;
5. control and measurement of machine speeds from 10 to 1000 m/min;
6. provision for interchangeable rolls, with diameters from 400 to 800 mm;
7. constant initial paper temperature and moisture content;
8. control of web tension and alignment to minimize breakage;
9. overall ease of operation.

Each of these criteria is described in greater detail below.

Cross-direction pressure profile variations in a commercial calender, which may be as wide as 10 m, are caused in part by roll bending. To minimize this bending, the width of the new calender was minimized, thus ensuring a constant cross-direction pressure profile. This also ensures an invariant machine-direction pressure profile across the width of the machine.

The following considerations influenced the choice of machine width: calender width had to be small enough to minimize roll bending while allowing sufficient width to minimize edge effects and sheet breakage, and the calendered sheet had to be wide enough for use in standard testing equipment, such as caliper, tear and burst gauges.

To determine a lower bound on paper width, several narrow reels of newsprint were cut and calendered in a small laboratory calender at the Pulp And Paper Research Institute of Canada. It was found that web breakage was a problem with paper narrower than 50 mm. It was also difficult to cut and wind a reel narrower than this. Furthermore, most standard tests can be performed with a long sample 25 mm wide. To provide a margin of safety, a calender width of 75 mm was chosen, with a paper width of 70 mm. This satisfied the requirement of narrowest possible nip width while reducing the likelihood of repeated sheet breaks.

The narrow calender width and constant pressure profile also ensure that any measurement of the gap between the rolls at one edge of the sheet is representative of the gap across the entire width. The range of paper thicknesses to be calendered is 50 to 200 μm; thus the second criterion is met with on-line displacement gauges having this range, mounted outside the nip and measuring the displacement of both rolls. Target resolution was ±1 μm.

Initial and final paper thickness gauges were also subject to the same range and resolution limitations as the in-nip thickness gauges.

The purpose of this work is measurement of in-nip paper thickness as a function of the process variables nip load, machine speed and roll diameter. These variables control the shape and duration of the pressure pulse applied to the paper in the nip and thus control the magnitude and rate of application of stress. Typical industrial values were selected to ensure the greatest possible relevance of the results: nip loads from 20 to 175 kN per metre of nip width, speeds up to 1000 m/min, and roll diameters of 400, 500 and 700 mm.

Sheet temperature and moisture content affect the rheological properties of the sheet. They will not be considered in this study and so will be held constant. Paper temperature will be the ambient temperature in the laboratory, 19 to 23 °C, and moisture content will correspond to the equilibrium content when the roll was wound, 8 to 10%.

The paper must be fed through the nip with a minimum of alignment and tension problems. Misalignment of feeds can lead to the the paper tending to run out of the nip, and excessive sheet tension on a narrow web can result in frequent sheet breaks.

To simplify the general running of experiments, careful attention was given to ease of operation, including calender roll and paper reel replacement, initial feeding of a new paper reel through the equipment and adjustment and zeroing of all sensors (particularly the gap sensors).

The equipment designed to meet these criteria is described next.

## 3.3 MECHANICAL SYSTEM

Overall views of the equipment that meets the criteria outlined above are shown in Figures 3.3.1 and 3.3.2. A full set of assembly drawings may be found in Appendix 2.

The calender stack consists of two rolls (Figure 3.3.1, #7 and #16) supported in a frame (#10) designed to withstand the high nip loads desired in this study. The upper roll is supported by an arm (#8), and the nip load is applied using a hydraulic cylinder (#9).

The sheet is taken from an unwind reel (#12) and stand (#13); sheet tension on this side of the nip is provided by an external drum brake adjusted with a handwheel. This mechanism is shown in Figure 3.3.7.

The calender is belt-driven by a 7.5 HP DC motor (#15) through an idler shaft which also serves as a surface drive roll (#2) for the windup reel (#1). Sheet tension on the windup side of the nip is controlled by a small DC motor (#3) driving a differential gearbox built into the idler shaft; this gearbox and motor allow the web speed at the windup reel to be varied slightly compared to the speed at the calender roll, resulting in fine control of the sheet tension.

Sheet tension and caliper are measured before and after the nip (#6). In-nip caliper is measured using a pair of displacement gauges, described in the following section. Nip load is measured using a load cell mounted between the cylinder (#9) and stand (#10).

Overall, the equipment is 4.0 m long and 2.0 m high. Individual components are described in detail below.

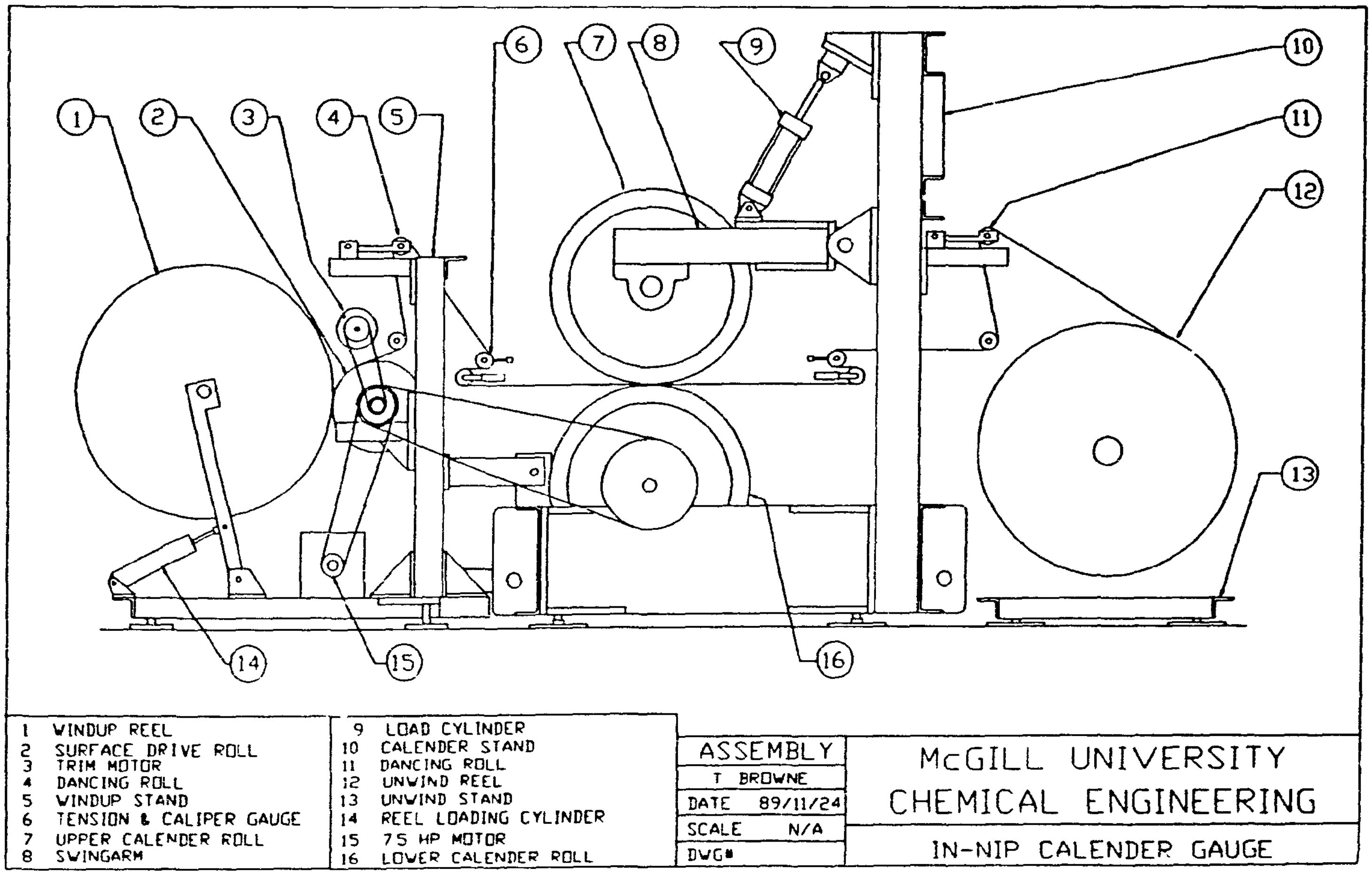



Figure 3.3.1: Overall view of calender assembly

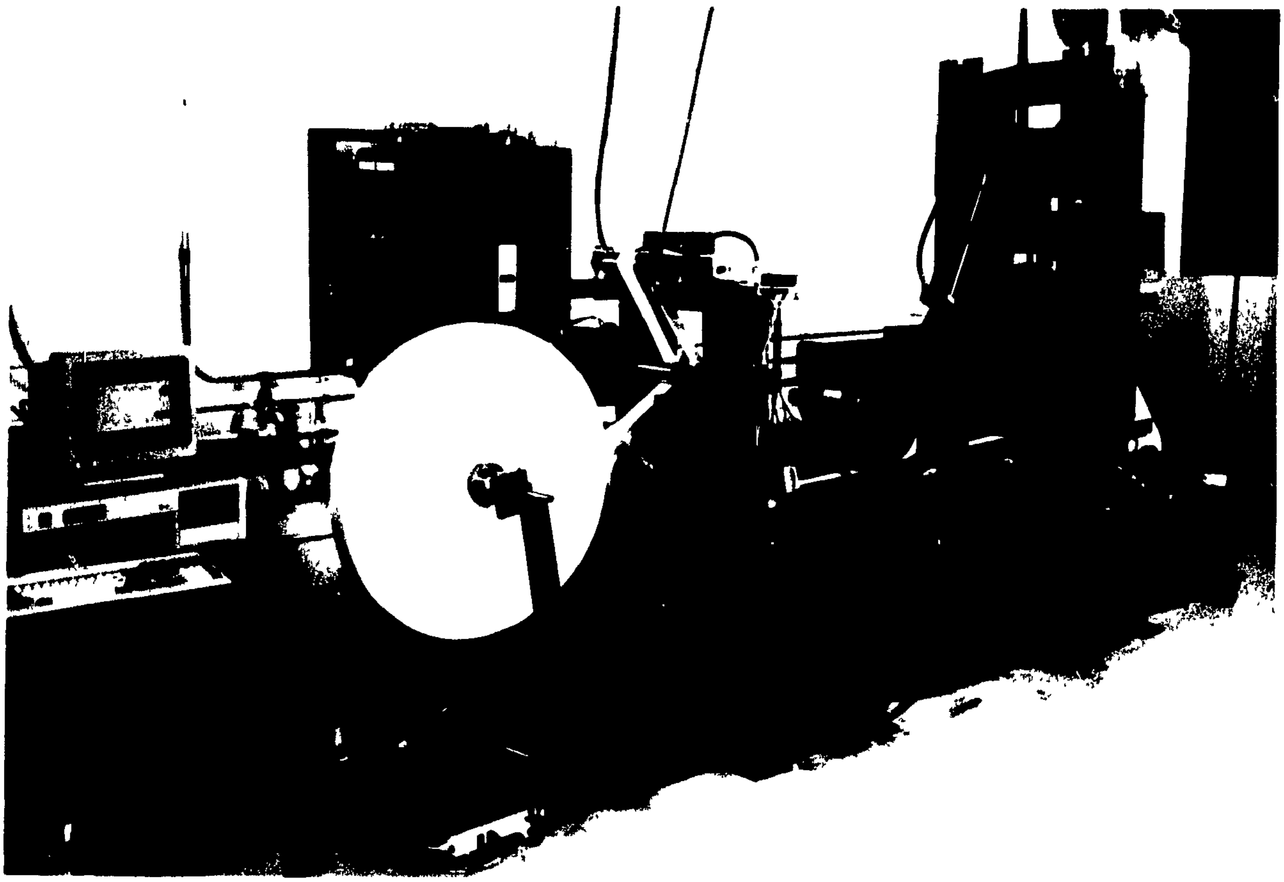


Figure 3.3.2: Overall view of calender assembly (photo)

The smallest roll is illustrated in Figures 3.3.3 and 3.3.4. Similar rolls with diameters of 20 and 28 in. were also made.

The narrow width of the rolls precluded fabrication as a welded structure; the rolls were to be solid. For reasons of cost, casting was eliminated; this left machining from a solid block of metal as the best fabrication method.

Analysis of the Hertzian contact stresses between two 75 mm wide discs pressed together in the absence of paper and with a load of 200 kN/m showed that mild steel would probably not be strong enough for long-term use as a calender roll, especially given the magnitude of the loads envisaged. High carbon steel, on the other hand, is difficult to machine, so the rolls were machined from mild steel and heat-treated to a greater hardness before the final grinding step.

Three pairs of rolls were made; their diameters are 16, 20 and 28 in. (404, 506 and 711 mm). The calender shafts are large, with a diameter of 75 mm, and ride in high-quality roller bearings for minimal radial and axial play.

The arm supporting the upper roll is shown in Figure 3.3.5. It is made of square section tubing welded to a pair of plates to provide high bending and torsional stiffness. The pivot points are widely separated for torsional reasons, and the large diameter pins ride in carefully machined bronze bushings. Both rolls are supported in a calender stand made of large section C-channels (see Figure 3.3.6).

The unwind stand (Figure 3.3.7) supports paper reels up to 1 m in diameter. The reel is mounted on a shaft provided with a brake for adjustment of sheet tension on the wind-up side. The brake is applied by one of two means: manually, using a

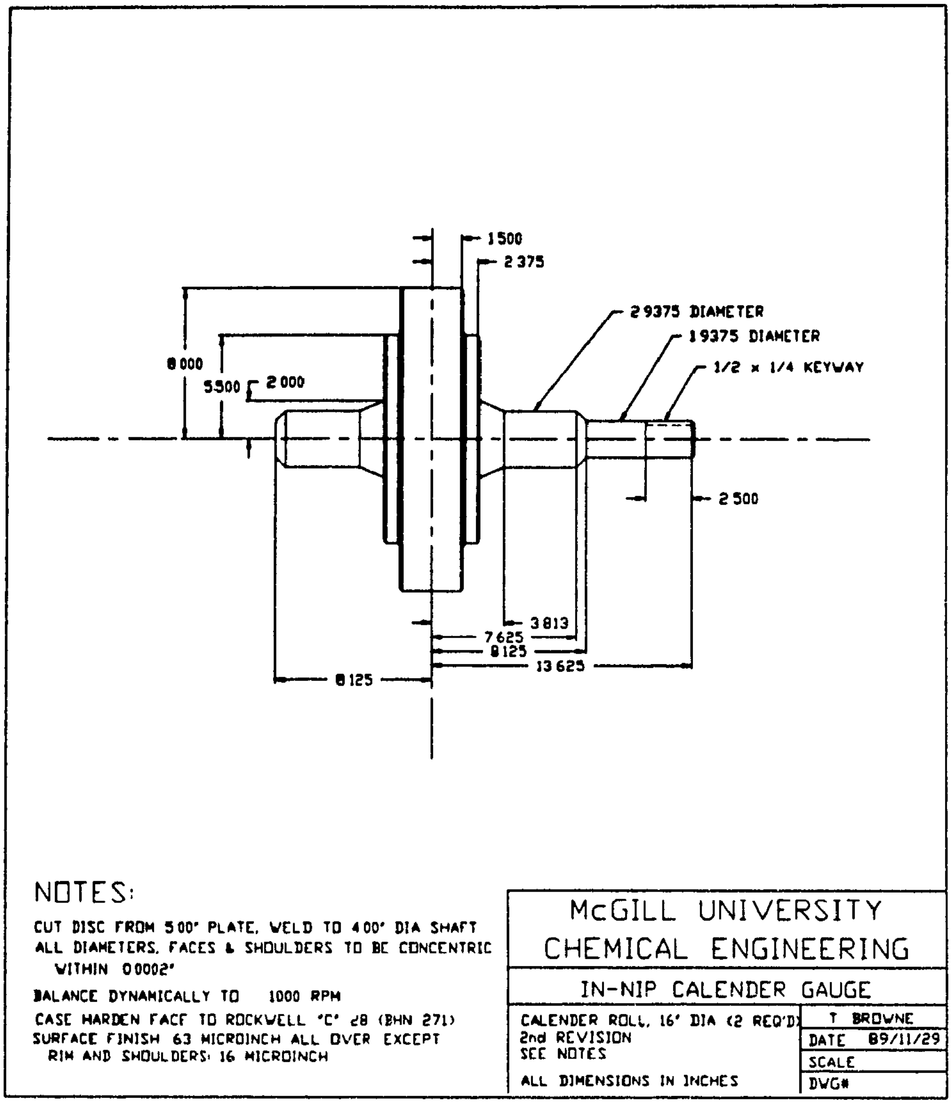



Figure 3.3.3: 16" Calender roll

Figure 3.3.4: 16" Calender roll (photo)

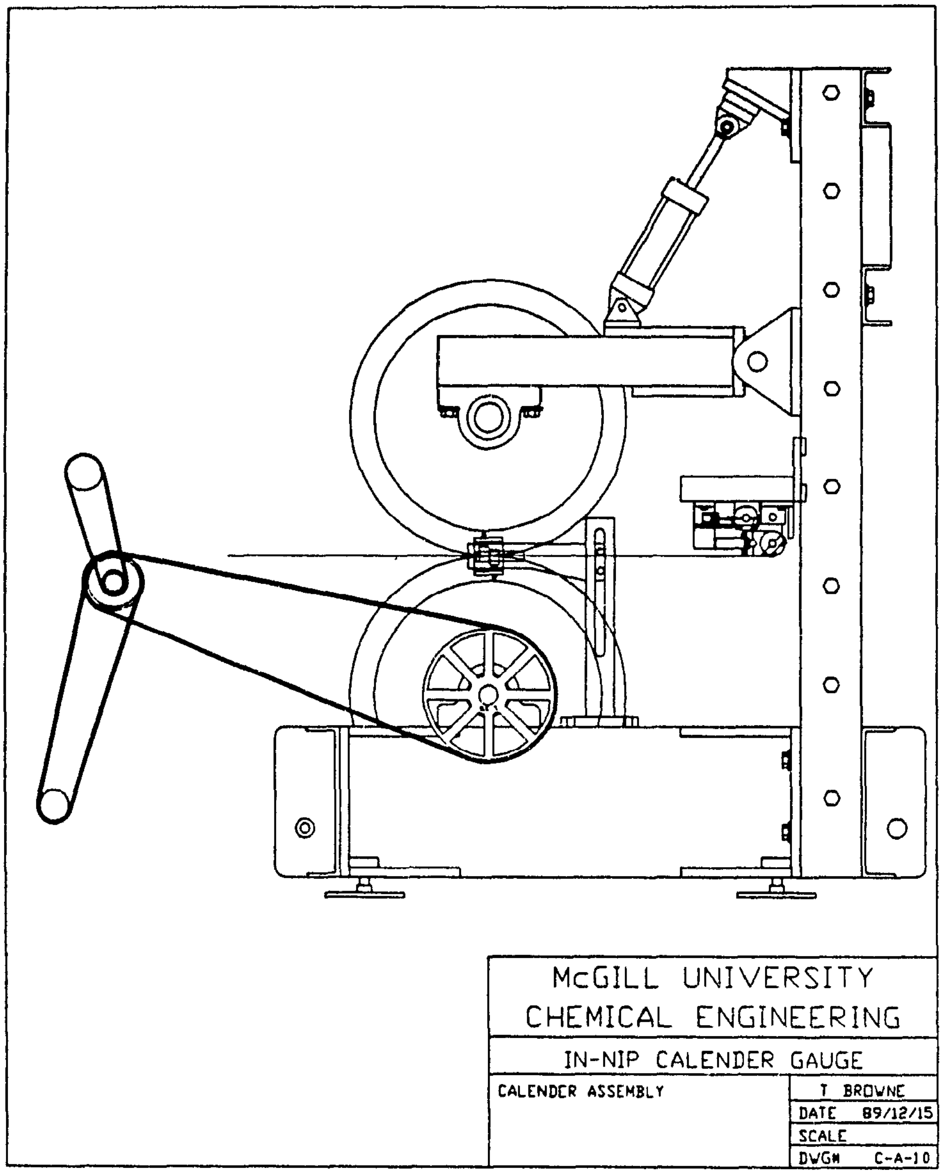



Figure 3.3.6: Calender stand assembly

handscrew, or automatically in the case of a sheet break, using a small pneumatic cylinder.

The wind-up stand (Figure 3.3.8) carries a 7.5 HP DC motor, which drives the lower calender roll and the wind-up reel surface drive by means of toothed belts.

To avoid having to provide an infinitely variable, high-torque drive to the wind-up reel as the reel builds, a surface drive is used with a small control motor to provide fine adjustment to the windup pulley ratio. As illustrated in Figures 3.3.9 and 3.3.10, the belt ratios along with the surface drive and calender roll diameters were chosen to provide a sheet speed 6.5% faster at the windup reel than at the calender nip. The control motor then slows the surface drive roll through a differential gearbox by an amount calculated to maintain sheet tension within bounds. In effect the surface drive roll is driven by two independent motors through the differential gearbox.

The main input to the differential gearbox (Figure 3.3.9, #5) is from the main motor through an idler shaft (#7). The control input (#12 and #13) is from a small DC motor. The output shaft (#4) of the differential gearbox (#8 through #11) drives the surface drive roll (#3) at a speed approximately equal to the main input, less 1 RPM for every 67 RPM of the control motor. Thus, at the maximum control motor speed of 1750 RPM, the surface drive roll has been slowed by about 26 m/min. Under these conditions, the minimum speed difference between the calender nip and the surface drive roll is 3.8% at a sheet speed of 1047 m/min.

This is also illustrated in the flow chart, Figure 3.3.10, where gear ratios for the three calender roll sizes are

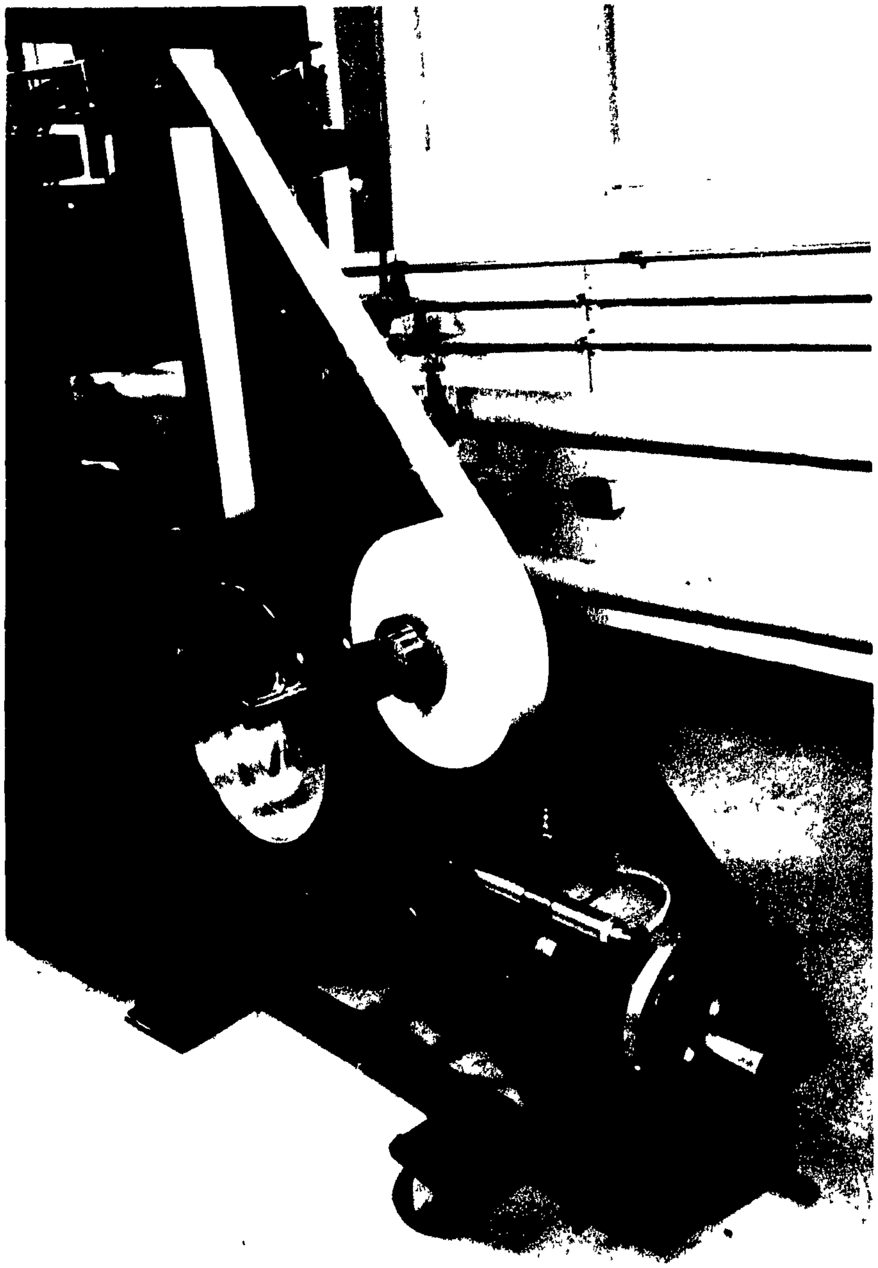


Figure 3.3.7: Unwind stand

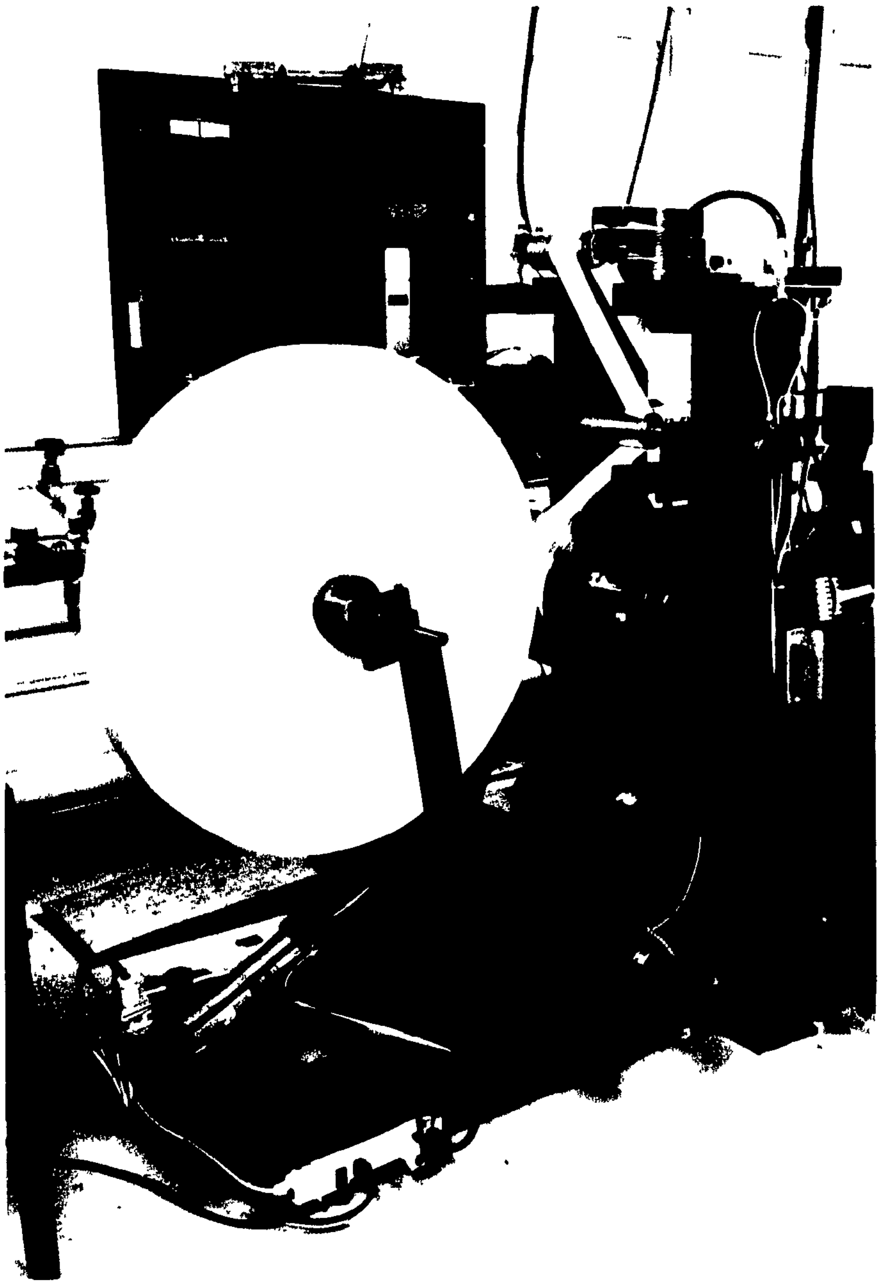


Figure 3.3.8: Windup stand

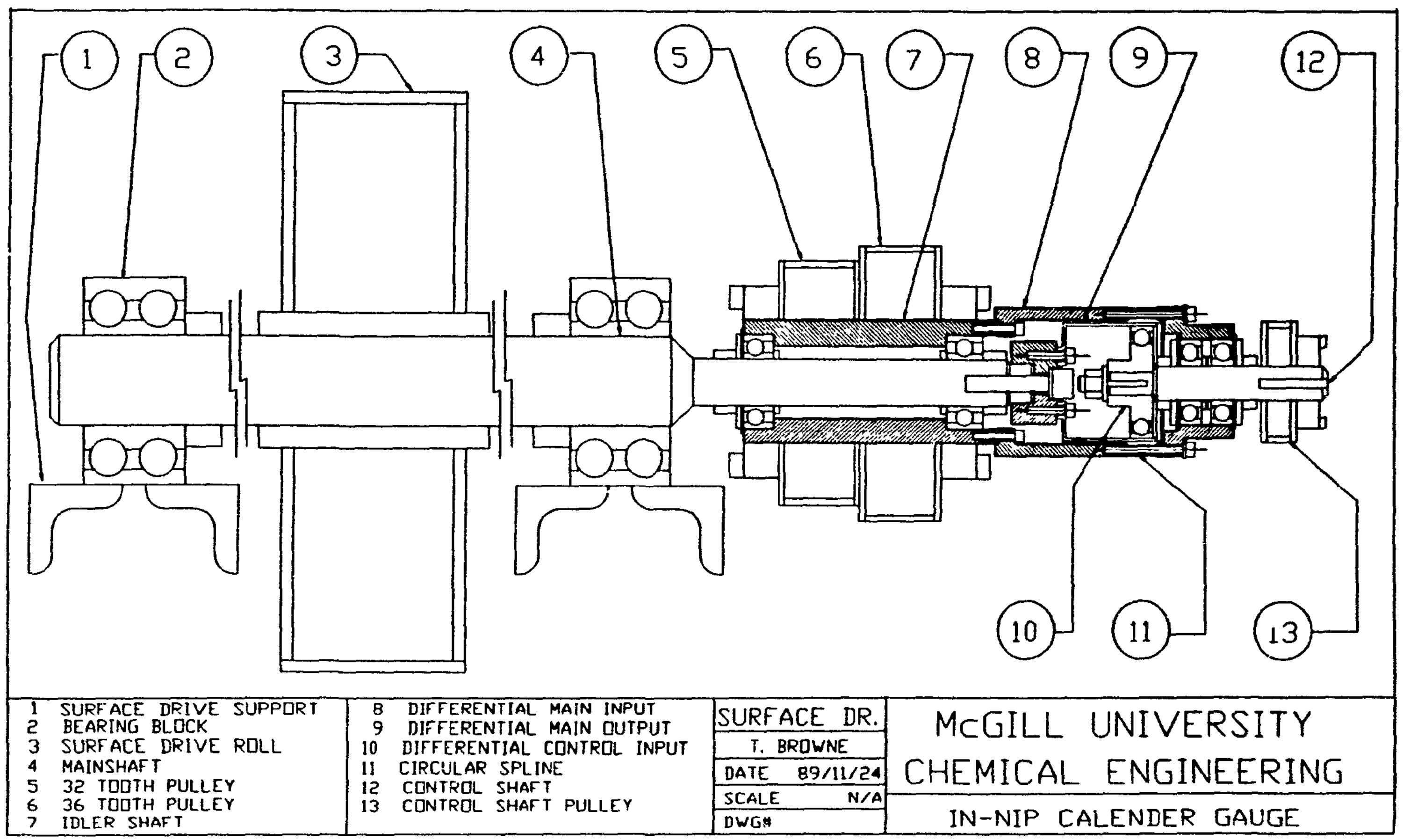



Figure 3.3.9: Variable speed drive section

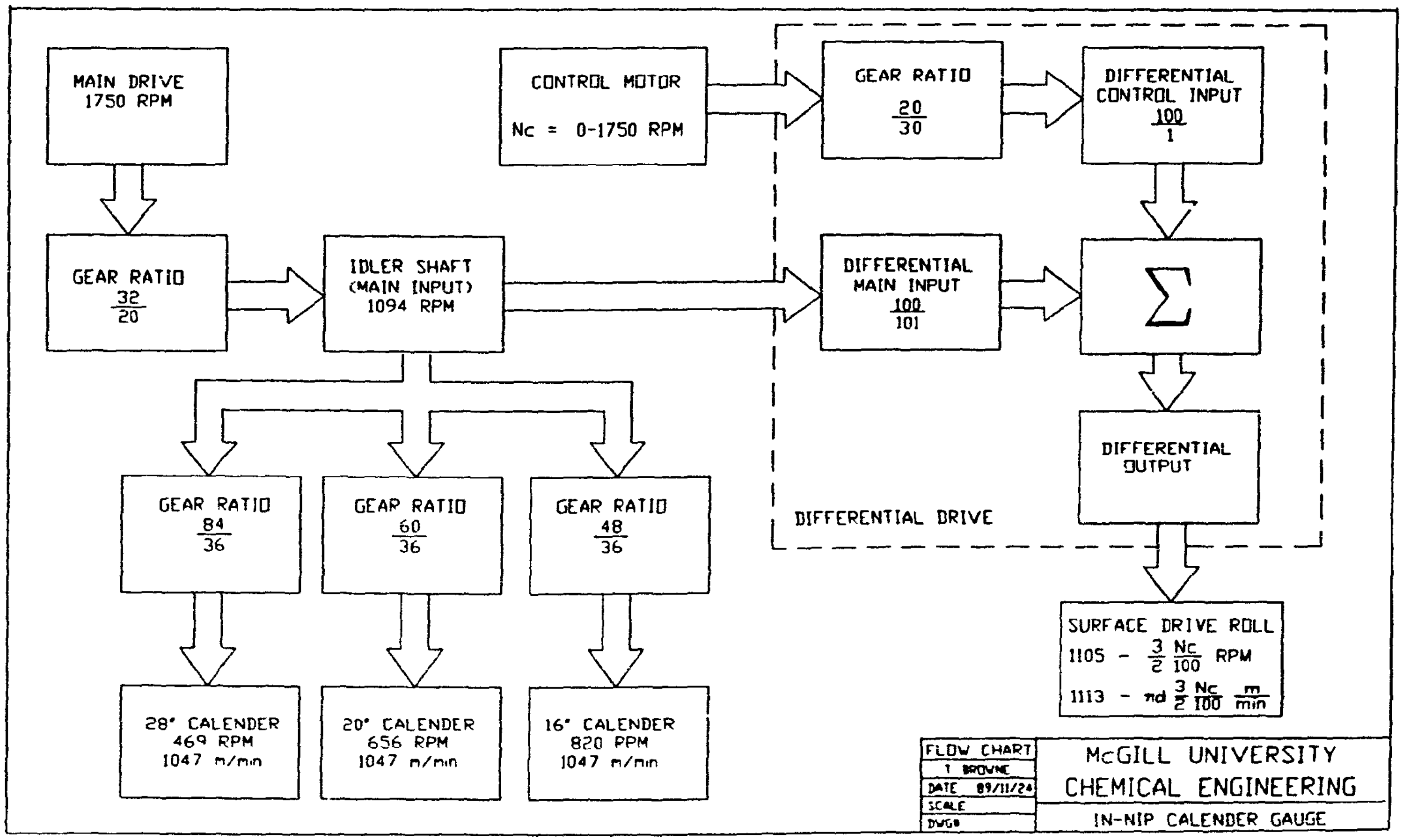



Figure 3.3.10: Variable speed drive flow chart

given, along with the corresponding sheet speeds at a main motor speed of 1750 RPM. The surface drive speed is also given for the same main motor speed, as a function of control input speed $N_c$ and roll circumference $\pi d$. Given the roll diameter of 321 mm, the surface speed at the winder can vary from 1087 to 1113 m/min when the surface speed at the calender nip is 1047 m/min, a difference of 3.8% to 6.3%.

## 3.4 Sensors, data acquisition and control system

### 3.4.1 Sensors

Mechanically, the most important design criterion was the narrow width of the nip, and the machining tolerances required to provide a uniform nip. Electronically, the most important aspect was the equipment for measuring in-nip paper thickness.

Several types of sensor were considered.

Lasers and other optical methods were rejected because of poor cost to performance ratios, and because other systems promised the desired resolution with less expense.

A variety of non-contacting systems were considered next. All involved a capacitive or inductive sensor to measure the position of a shoulder machined into the side of the rolls. Capacitive sensors were rejected because they are sensitive to other factors such as dirt on the surface and humidity in the air. Inductive sensors were rejected because they require a well-defined magnetic target; in the presence of large steel rolls they could give false readings.

Among the contacting devices, linear variable differential transformers (LVDT) were considered. The sensor consists of a plunger and core assembly in contact with the roll. The plunger is displaced through the hollow centre of a transformer. The altered output of the transformer as the core moves through the magnetic field is proportional to the displacement of the core over a small range. Tests using an LVDT on laboratory calender at PAPRICAN showed that the resolution was as good as that of the more troublesome non-contacting devices. For these reasons this type of transducer was selected for measuring both sheet thickness and roll displacement.

To measure roll displacement, the plunger is mounted in a linear bearing and carries a carbide tip which rests on a shoulder machined in the side of the roll, as shown in Figures 3.4.1 and 3.4.2. The plunger is spring-loaded to keep it in contact with the shoulder in spite of slight eccentricities in the shoulders, but the loading is not so heavy that undue wear of either the tip or the shoulder occurs.

One of the caliper gauges is shown in Figure 3.4.3. The plunger is actuated by a paddle with a plastic face in contact with the moving sheet. The pressure applied to the paddle is about 3.5 kPa to prevent marking of the sheet. The pressure is lower than that used in a standard caliper gauge, and the readings from the on-line sensors are corrected for this difference by comparing readings with a standard caliper gauge.

Nip loads are measured with a load cell placed under the clevis for the hydraulic cylinder. The reading is then corrected for the geometry of the swingarm, and for the basic load due to the weight of the upper roll and arm.

The sheet tension sensors consist of load cells mounted under an idler roll. There is one such sensor either side of the nip. Figures 3.4.4 and 3.4.5 show the mechanism for eliminating most of the weight of the idler roll from the reading obtained from the load cell.

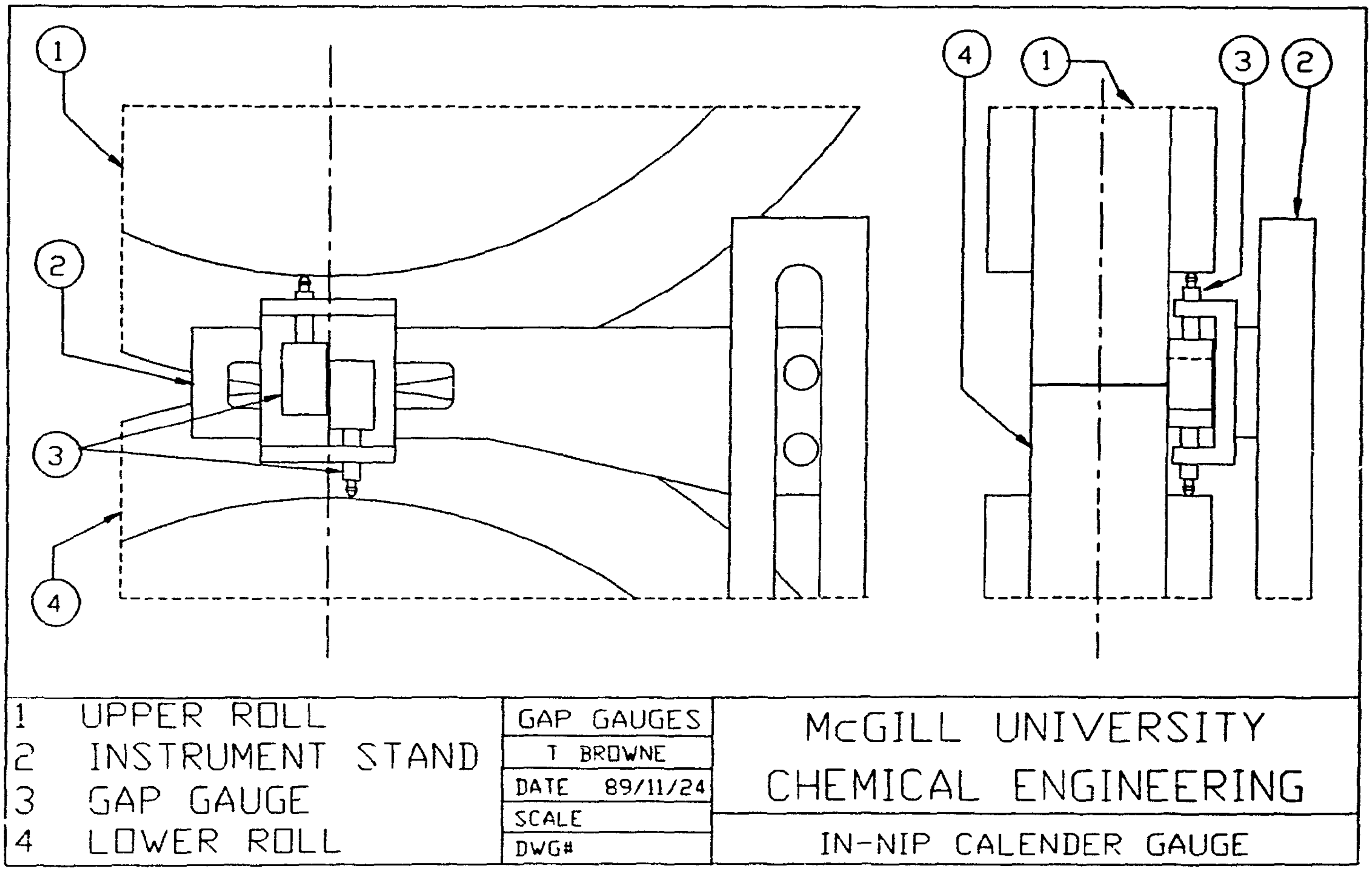



Figure 3.4.1: LVDT mounting, nip gap

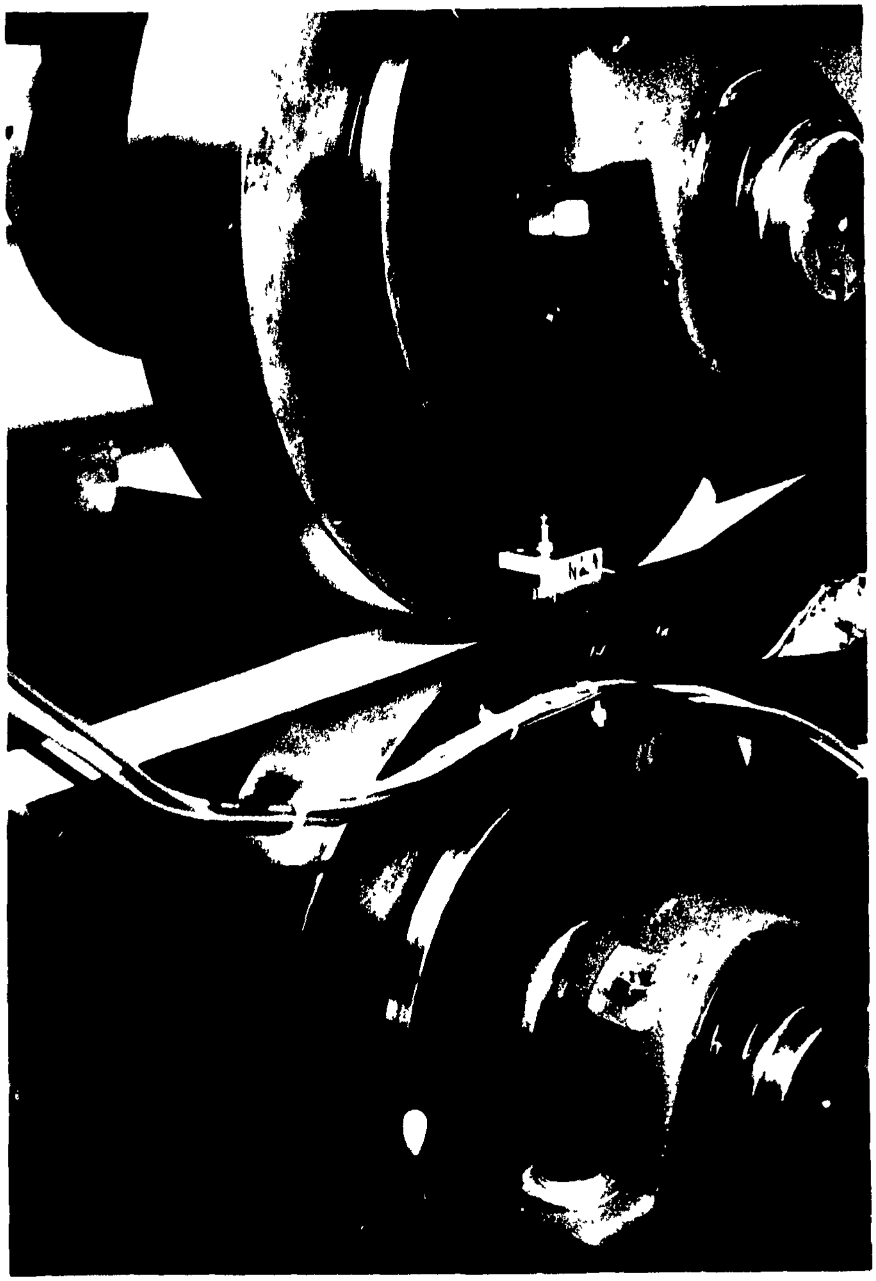


Figure 3.4.2: LVDT mounting, nip gap (photo)

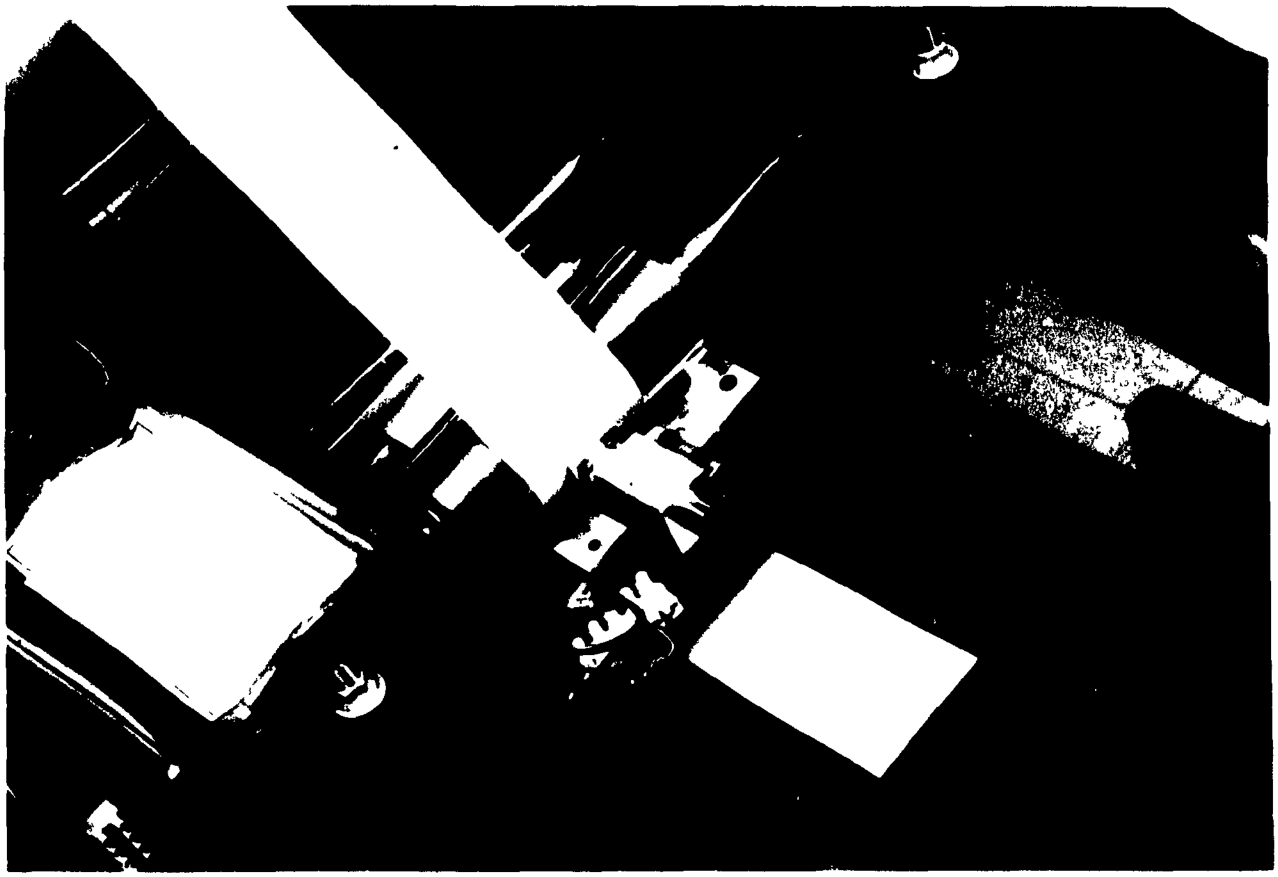


Figure 3.4.3: LVDT mounting, caliper (photo)

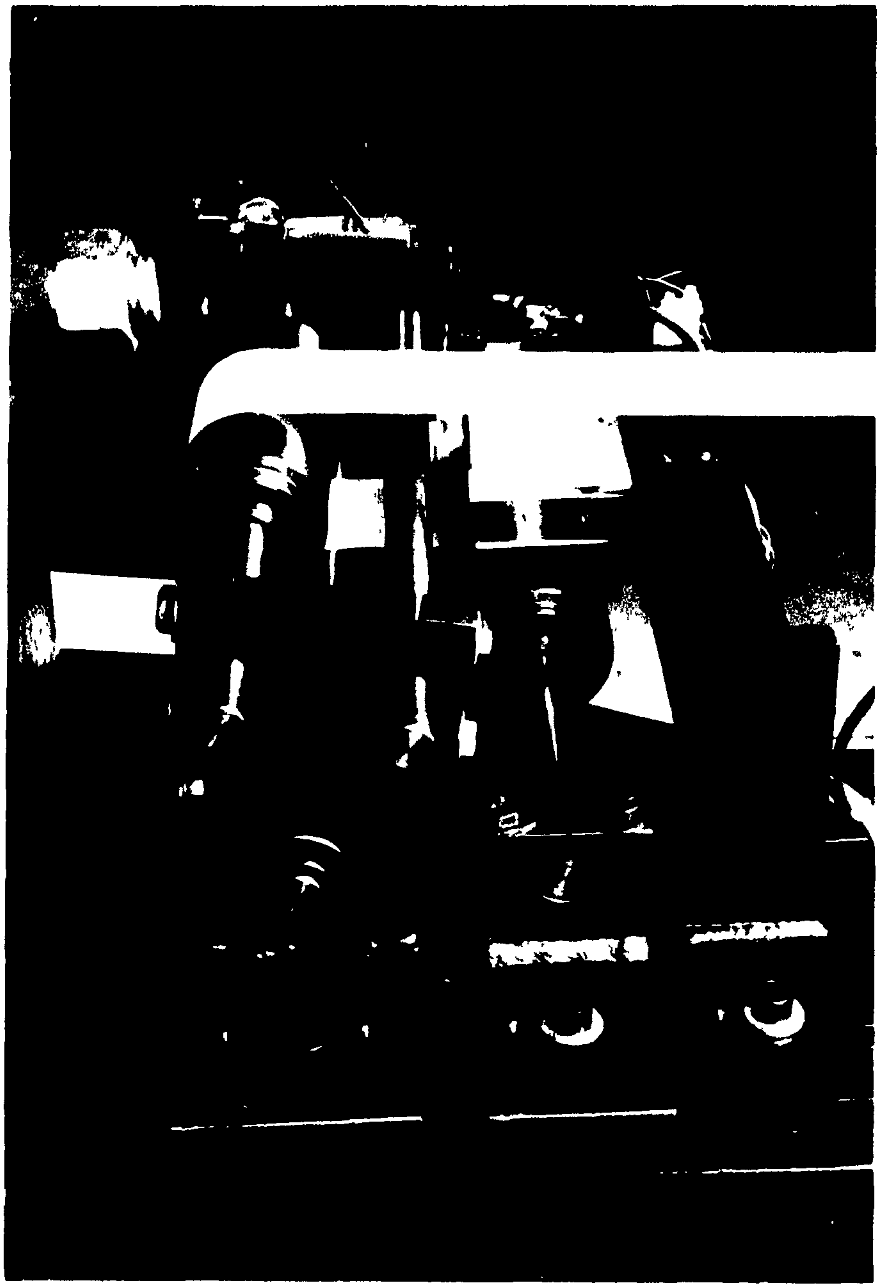


Figure 3.4.4: Sheet tension device (photo)

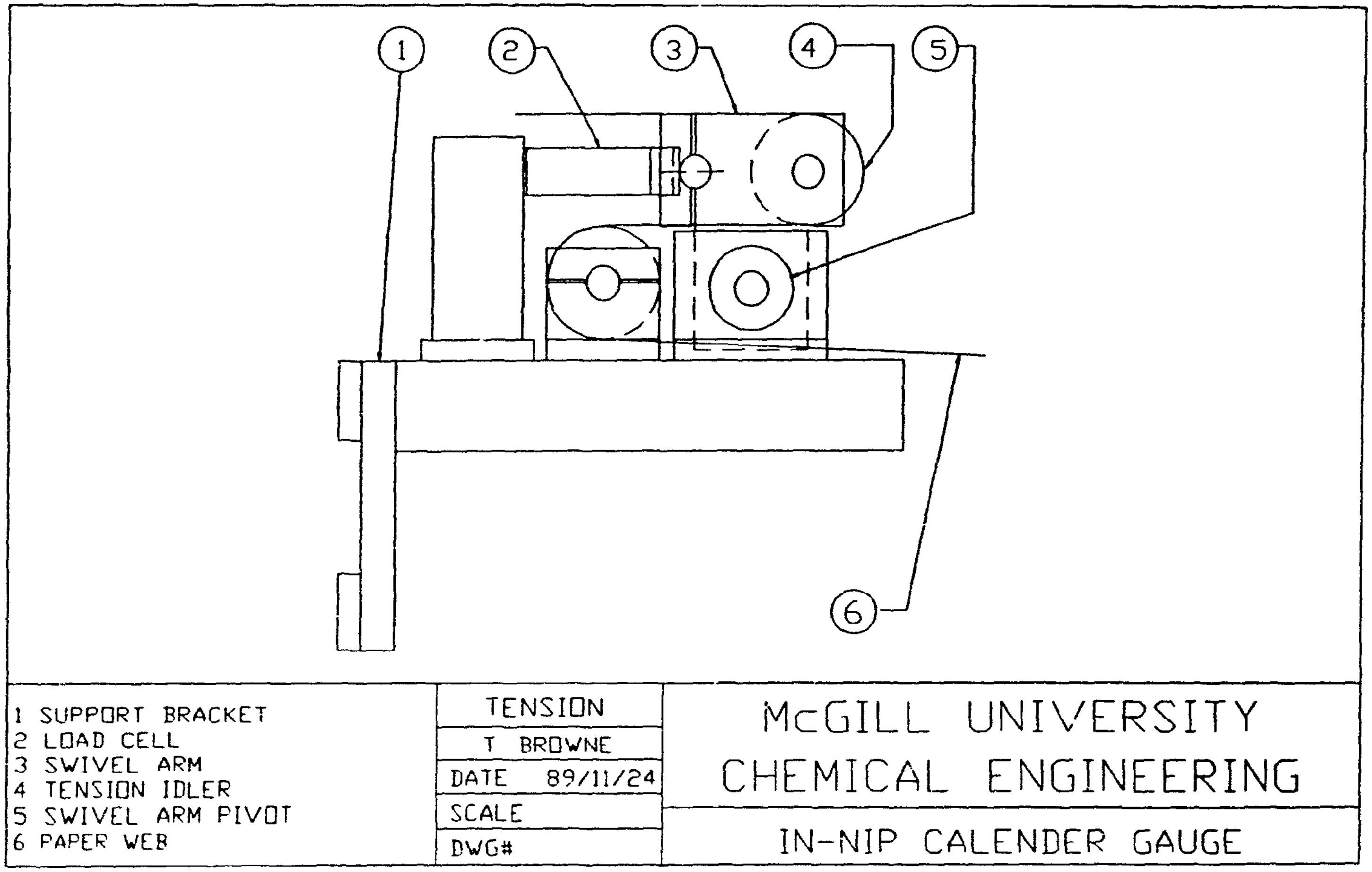



Figure 3.4.5: Sheet tension device assembly

### 3.4.2 Data acquisition and control system

All the sensors described in Section 3.4.1 are connected to a 12-bit analog to digital conversion board installed in an IBM AT compatible computer. Data is acquired and displayed and sheet speed and tension are controlled by means of a program written in Microsoft's QuickC programming language. A listing of the program appears in Appendix 1.

Sheet speed is controlled by sending a 0-10 V signal to a modified controller supplied by the manufacturer of the main DC motor. Speed control is within 1%.

Sheet tension after the nip is controlled using a PID loop in software to send a 0-10 V signal to a similarly modified controller supplied by the manufacturer of the control motor. Sheet tension before the nip is controlled by the operator in response to the displayed value; this has proven to be sufficient due to the greater strength of the sheet before calendering.

Nip load is also controlled by the operator by means of a pressure regulator in the hydraulic circuit supplying the load cylinder.

The output from the sheet tension, nip load, caliper and nip gap gauges over a certain time period are saved to disk for further processing. The most important processing is the calculation of in-nip paper thickness from nip-gauge data; this is described next.

### 3.4.3 CALIBRATION OF NIP GAP GAUGES

Determination of the in-nip calendering equation depends on accurate and repeatable measurements of in-nip paper thickness. The calibration of the nip gap gauges, assuming their outputs are linear, determine the slope and intercept of the displacement vs. digital output curves. The curves are indeed linear, and the slopes have been determined. Figure 3.4.6 gives the digital value read from the A/D board by the acquisition program as a function of displacement of the gauge plunger.

Ideally, the intercepts can also be determined by calibration. The in-nip paper thickness $t_n$ then depends only on the extension of the upper gauge, $u_s$, compared with its extension with no paper in the nip, $u_{so}$:

$$t_n = u_s - u_{so} \qquad [3.4.1]$$

There are two problems with this approach. First, the lower roll may show some small deflection (in spite of the strong mechanical design), especially at high loads. Thus the deflection of the lower gauge must also be taken into account to determine the true total in-nip paper thickness. Second, both rolls show a certain amount of out-of-round due to the difficulty in creating a perfectly circular roll; as a result all signals from the gauges are superimposed on a periodic signal. The intercepts, or offsets, which would normally be obtained by calibration with a standard thickness gauge are therefore periodic functions of distance around the circumference of any given roll. As well, the shoulder eccentricity (which is what is measured) is different from the rim eccentricity (which is the quantity desired).

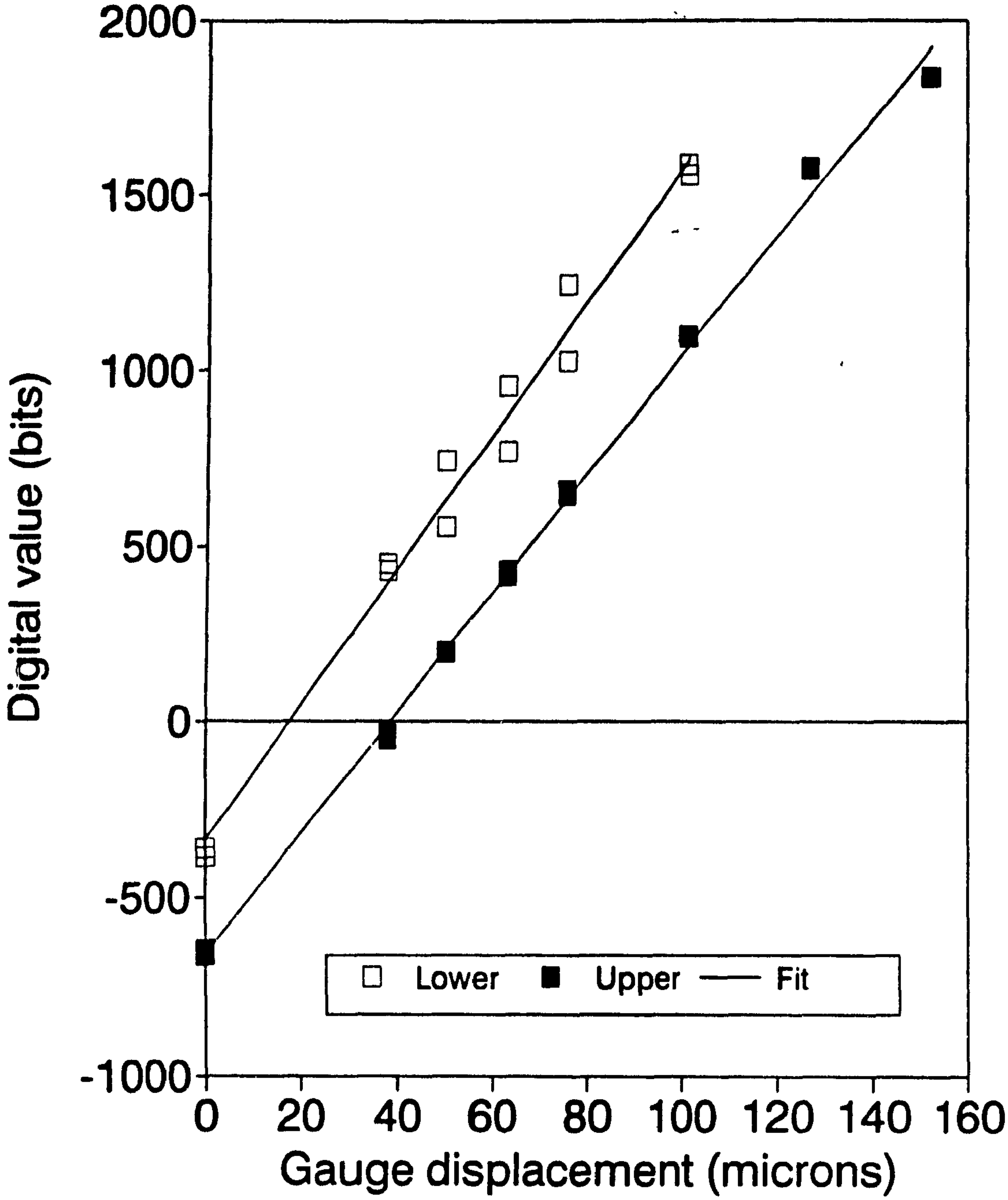



Figure 3.4.6: Calibration curve for the in-nip displacement gauges

Since the superimposed signal is perfectly periodic with a constant amplitude, averaging the signal over many rotations of the roll should give an accurate estimate of the average machine-direction paper thickness. However, this averaging process eliminates random variations in paper thickness which might be of interest.

An alternative solution involves taking the "signatures" of the rolls and approximating the eccentricities with a small number of Fourier terms. This data can then be used to subtract the effect of roll eccentricity from the raw data. This system would also have to take into account the fact that the two rolls are not always in the same rotational position relative to one another and that the upper roll rides on the lower roll. The periodic variation in the reading from the upper gauge therefore depends on the rotational position of both rolls at any given time.

Let us first define the following variables, which are illustrated in Figure 3.4.7:

- $u$, position of the upper roll, positive upwards, a periodic function of radial position of the roll;
- $v$, position of the lower roll, positive downwards, a periodic function of radial position of the roll;
- $t_u$, $t_v$, paper thickness due to deflection of upper or lower roll;
- s, r, subscripts indicating shoulder and rim, respectively;
- o, subscript indicating the absence of paper in the nip;
- $e$, eccentricity of lower roll rim, a periodic function of radial position of the roll, describing the deviation from circularity.

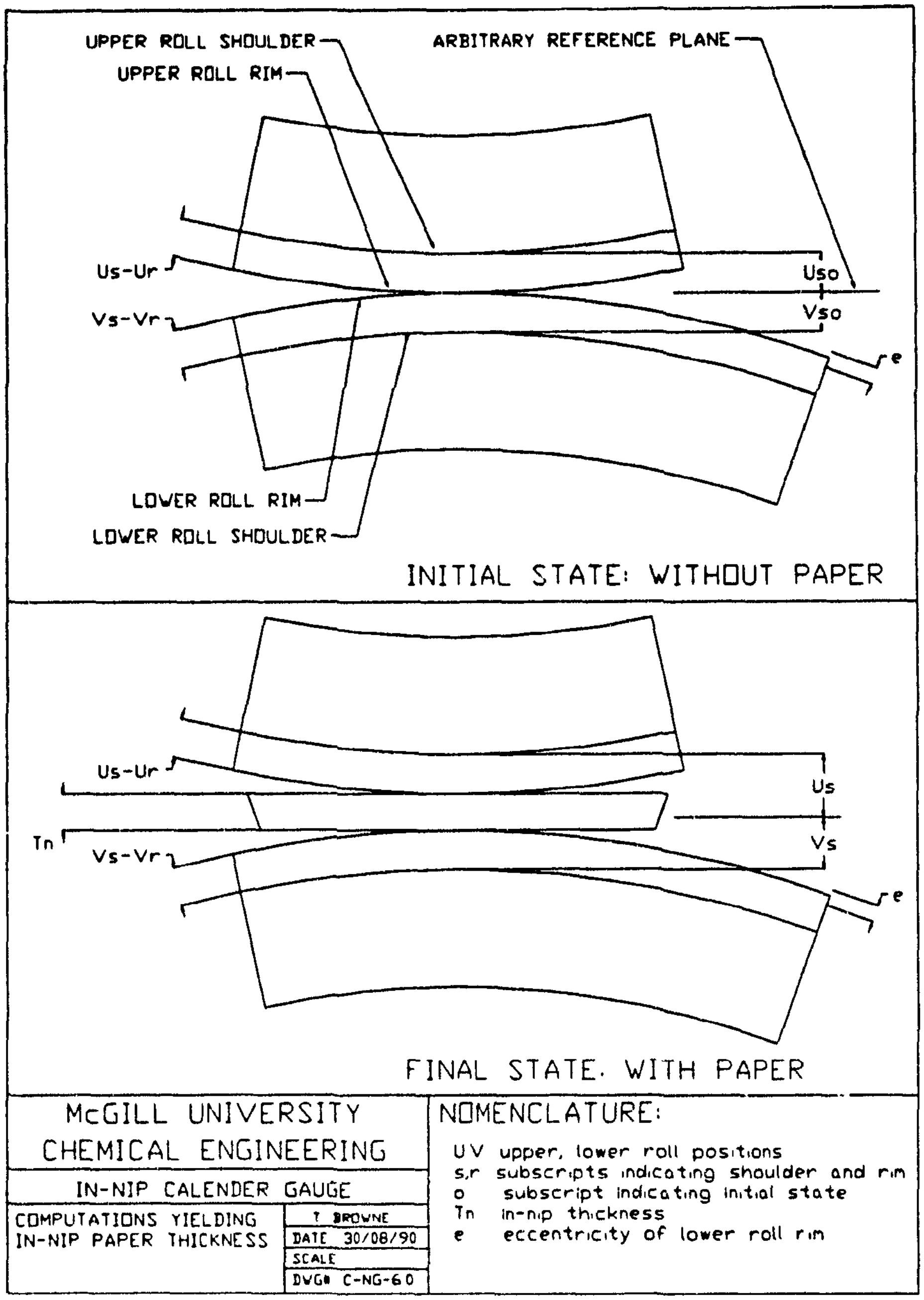



**Figure 3.4.7: Definition of variables describing roll eccentricities**

The location of the lower roll rim at any time as measured from the arbitrary reference plane shown in Figure 3.4.7 and in the absence of paper is then

$$\begin{aligned} v_{ro} &= v_{so} - \left(v_s - v_r\right) \\ &= v_{so} - \delta v \end{aligned} \qquad [3.4.2]$$

where $\delta v$ is the known difference between the shoulder and rim out-of-rounds. Similarly, when paper is inserted in the nip and a load applied, the lower roll may be displaced a small distance due to deflections in the bearings; the new location of the lower roll rim is

$$v_r = v_s - \delta v \qquad [3.4.3]$$

The difference between these two readings represents the part of the in-nip paper thickness due to lower roll deflection:

$$\begin{aligned} t_v &= v_r - v_{ro} \\ &= v_s - v_{so} \end{aligned} \qquad [3.4.4]$$

A similar argument can be used for the upper roll, with the added complication that the displacement of the upper roll depends on the eccentricity $\varepsilon_{vo}$ of the lower roll since the upper roll floats on the lower roll. When there is no paper in the nip, the displacement of the upper roll is

$$u_{ro} = u_{so} - \delta u_o + e_{vo} \qquad [3.4.5]$$

Here, the relation between upper-roll offset $\delta u_o$ and lower-roll eccentricity $e_{vo}$ depends on the relative position of the two rolls; there is now the possibility of a pnase shift once paper is inserted in the nip since the two rolls are not exactly the same diameter. With paper in the nip the new location of the upper roll rim is

$$u_r = u_s - \delta u + e_v \qquad [3.4.6]$$

Paper thickness in the nip, $t_n$, is now the sum of $t_u$ and $t_v$:

$$t_u = \left(u_s - \delta u + e_v\right) - \left(u_{so} - \delta u_o + e_{vo}\right)$$
$$t_n = \left(u_s - \delta u + e_v\right) - \left(u_{so} - \delta u_o + e_{vo}\right) + \left(v_s - v_{so}\right) \qquad [3.4.7]$$

The eccentricities cancel if and only if the relative position of the two rolls does not change between taking the paperless reading and inserting paper; then $e_v$ and $e_{vo}$ (which already have the same amplitude and frequency) also have the same phase. Otherwise, there is a phase difference between the two rolls.

Similarly, upper-roll offsets cancel if shoulder and rim eccentricities are both zero (in which case $\delta u$ and $\delta u_o$ are both constants) or if there is no phase difference between the two readings. Once these phase differences have been removed, any remaining variation in $t_n$ should be due to random paper thickness variations. As well, the displacement $v_s - v_{so}$ is expected to be small; if the eccentricities are also small, these approximations yield the expected expression for in-nip paper thickness due only to the displacement of a perfectly circular upper roll:

$$t_n \cong u_s - u_{so} \qquad [3.4.8]$$

The eccentricity of the lower roll was measured and fitted to a sum of periodic terms using the Fourier transform; components with amplitudes less than 0.5 μm were eliminated, resulting in an expression for $e_v$ as a function of radial position of the lower roll:

$$e_v = \sum_{i=1}^{N} a_i \sin\left(k_i \omega t + \phi_i\right) \qquad [3.4.9]$$

The values of $a_i$ and $\phi_i$ were found using Fourier analysis; $\omega$ is the base frequency, equal to the reciprocal of the total data acquisition time, and $k_i$ is an integer. When terms with $a_i < 0.5\ \mu m$ were eliminated, N became equal to 12. The value of the correction $e_v - e_{vo}$ is then

$$e_v - e_{vo} = \sum_{i=1}^{N} a_i \left[\sin\left(k_i \omega t + \phi_i\right) - \sin\left(k_i \omega t + \phi_i + \psi\right)\right]$$

[3.4.10]

where $\psi$ is the phase shift between readings taken with no paper and those taken later with paper. Radial position sensors were fabricated to give an indication of the position of the two rolls relative to each other and to the thickness data; this was used to determine $\psi$.

The amplitude of the Fourier term corresponding to one cycle per rotation of the lower roll was 13 $\mu m$. This signal was thus quite evident, given in-nip paper thicknesses of 30 to 80 $\mu m$.

# 4. Commissioning of the equipment

## 4.1 Introduction

Two initial series of commissioning experiments were performed to verify the proper functioning of the equipment. In the first experiment, nip load was held constant while sheet speed was varied; in the other load was varied at constant speed. Both runs were performed with a single pair of rolls pressing one type of paper. The results were encouraging but showed that minor alterations were necessary for routine high-speed use of the equipment.

Test procedures and results are described in this chapter.

## 4.2 Test procedures and results

Several experiments were performed. Startup of the calender for a typical run proceeded as follows.

Power supply to the sensors was switched on. The sensors were then left to warm up for at least 90 minutes, since early tests showed that sensors and power supply were fairly sensitive to temperature changes.

Next, paper was threaded through the calender. The computer program was used to accelerate the main motor slowly up to speed. Once the sheet speed was constant and the windup-side sheet tension was stable, acquisition of a set of data was initiated. The complete set of sensors was scanned a total of 2048 times, at a rate of 500 Hz. Thus each data set represents an elapsed time of 4.096 s.

The data was saved to disk, and the motor was accelerated to the next speed. The process was repeated until interrupted by a sheet break or until the experiment was terminated.

The procedure was then repeated at a single speed with various loads. Both experiments were repeated several times.

The initial data processing did not involve use of the Fourier method described in Chapter 3; instead the data points were averaged over several complete revolutions of the lower calender roll to produce an approximation to the true shape of the paper. The large standard deviation in in-nip paper thickness due to the sinusoidal signal was thus reduced.

Next strips of calendered and uncalendered paper were conditioned at 23 ℃ and 50% R.H. for 24 hours before thicknesses were measured using a standard micrometer. These values of thickness were used to compute in-nip and permanent strains, or bulk reduction.

The paper sample used in this study was a high-grade newsprint made at the Boise-Cascade mill in Fort Frances, Ontario. The furnish was a mixture of kraft and groundwood pulps, as listed in Table 4.2.1.

The calendering variables for the two experiments are given in Table 4.2.1, and a summary of the results in terms of relative compression or strain are shown in Figure 4.2.1 and Table 4.2.2. Typical raw data is given in Appendix 3.

Table 4.2.1: Calendering variables for commissioning tests

At constant load

| | |
|---|---|
| Load | 39 kN/m |
| Roll radius | 0.2 m |
| Sheet speeds | 50, 90, 100, 150, 160, 280 m/min |
| Paper type | Newsprint, made from<br>71.5% Semi-bleached groundwood pulp<br>28.5% Bleached kraft pulp |
| Initial thickness | 102 μm |
| Basis weight | 52 $g/m^2$ |
| Initial bulk | 1.97 $cm^3/g$ |

At constant speed: as above, with the following changes:

| | |
|---|---|
| Loads | 39, 80, 140, 207, 250 kN/m |
| Sheet speed | 300 m/min |

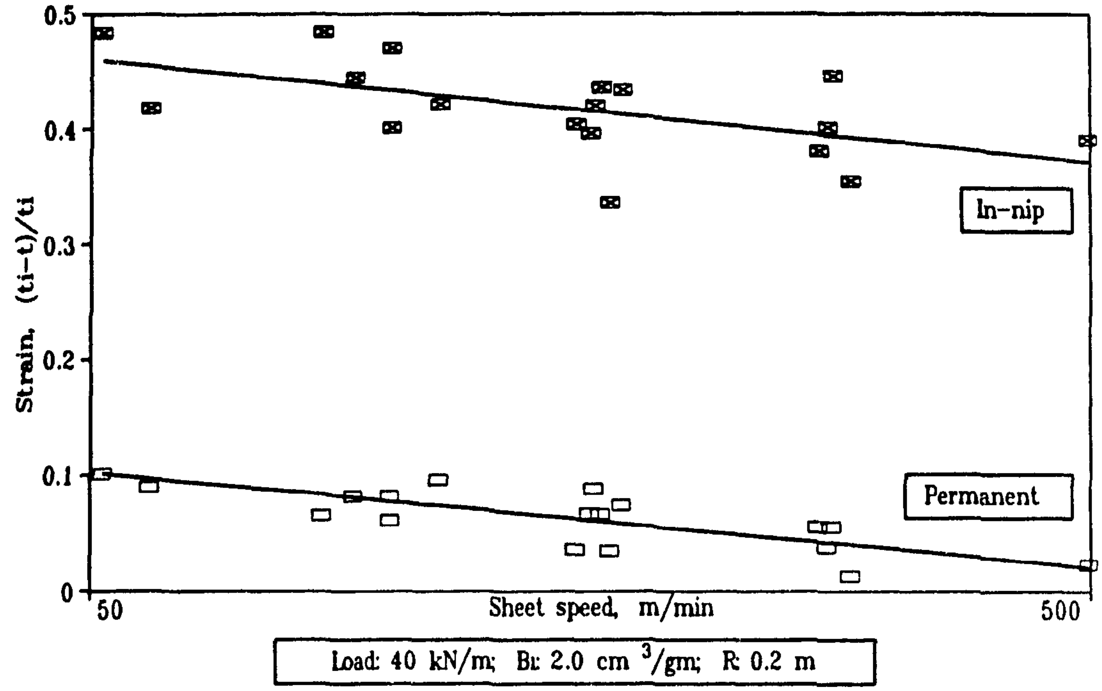
Table 4.2.2: Summary of commissioning tests

At constant load

| Expt. # | Sheet speed, m/min | Load, kN/m | Strains Permanent | Strains In-nip |
|---|---|---|---|---|
| 21 a | 52 | 39 | 0.1006 | 0.484 |
| 18 a | 58 | 39 | 0.0893 | 0.418 |
| 43 a | 87 | 40 | 0.0654 | 0.484 |
| 27 a | 94 | 39 | 0.0818 | 0.444 |
| 37 b | 103 | 39 | 0.0608 | 0.401 |
| 20 b | 103 | 36 | 0.0813 | 0.470 |
| 18 b | 115 | 39 | 0.0359 | 0.421 |
| 59 a | 160 | 37 | 0.0351 | 0.404 |
| 37 c | 165 | 39 | 0.0671 | 0.396 |
| 18 c | 167 | 39 | 0.0875 | 0.420 |
| 27 b | 169 | 40 | 0.0660 | 0.436 |
| 65 a | 173 | 38 | 0.0344 | 0.335 |
| 20 c | 178 | 36 | 0.0736 | 0.433 |
| 37 d | 284 | 37 | 0.0552 | 0.380 |
| 58 a | 290 | 38 | 0.0368 | 0.401 |
| 28 a | 293 | 40 | 0.0546 | 0.445 |
| 67 a | 306 | 40 | 0.0129 | 0.354 |
| 54 a | 535 | 38 | 0.0222 | 0.391 |

At constant speed

| | | | | |
|---|---|---|---|---|
| 58 a | 290 | 38 | 0.0368 | 0.401 |
| 28 a | 293 | 40 | 0.0546 | 0.445 |
| 67 a | 306 | 40 | 0.0129 | 0.354 |
| 67 b | 304 | 80 | 0.0691 | 0.577 |
| 67 c | 301 | 137 | 0.1159 | 0.674 |
| 58 b | 284 | 146 | 0.1299 | 0.774 |
| 67 d | 300 | 207 | 0.1615 | 0.753 |
| 67 e | 297 | 250 | 0.1977 | 0.791 |



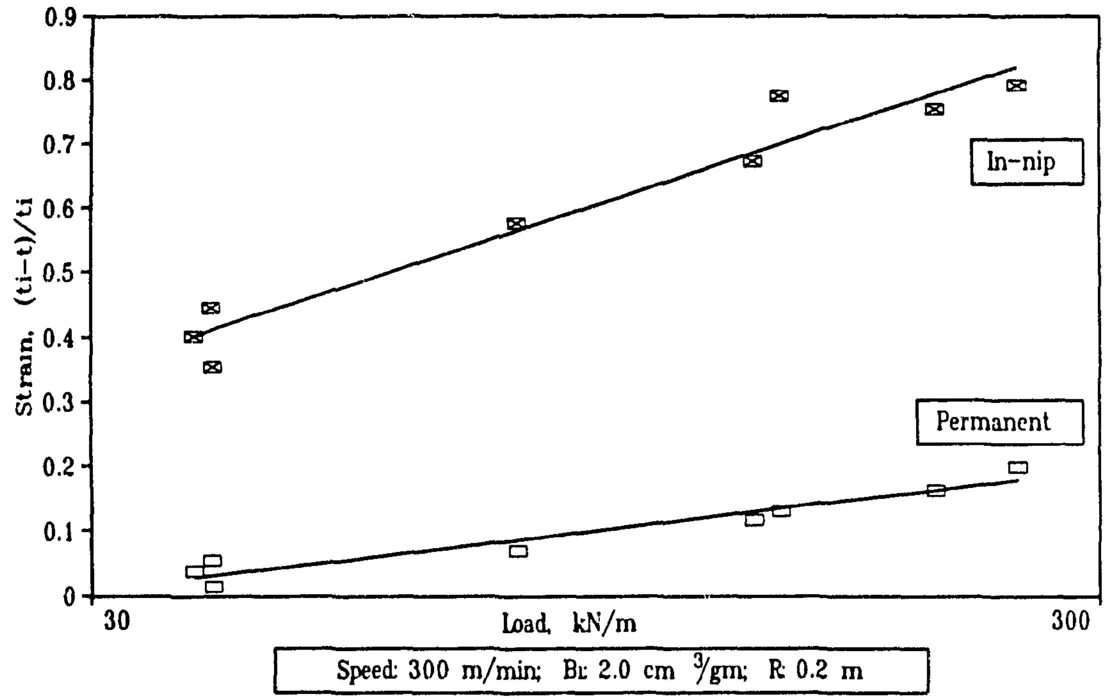




Figure 4.2.1: Paper strain in a calender, initial results

Each point in Figure 4.2.1 was the result of a single experiment. Several experiments were performed at lower speeds over several days with the same reel of paper. Fewer runs were made at higher speeds due to difficulties controlling sheet tension as speeds increase.

The upper curve in Figure 4.2.1 shows permanent and in-nip strain as a function of sheet speed. When the permanent strain data are fitted to an equation of the form

$$\varepsilon_p = A + \frac{a_V}{B_i} \log_{10} V \qquad [4.2.1]$$

where $B_i$ is the initial bulk and V is the sheet speed in m/min, the slope $a_V$ is -0.0408 g/cm$^3$. This is in reasonable agreement with Crotogino et al. [14, 15], who found a value of -0.0208 g/cm$^3$ for a different newsprint furnish. Intercepts are not comparable since the variable A above includes the effect of all the other terms in the calendering equation which were kept constant in the present study. When the in-nip data are fitted to Equation 4.2.1, the slope is -0.0444 g/cm$^3$.

Similarly, when the constant speed data are fitted to an equation of the form

$$\varepsilon_p = A + \frac{a_L}{B_i} \log_{10} L \qquad [4.2.2]$$

where L is the load in kN/m, the slope $a_L$ is 0.0941 g/cm$^3$, which is in excellent agreement with Crotogino et al [14, 15], who reported values ranging from 0.0912 to 0.0988 g/cm$^3$. When the in-nip data are fitted to Equation 4.2.2, the slope is 0.261 g/cm$^3$. The difference between the two slopes will be a topic for future research.

The in-nip strain at constant load is about 0.45 for the range of speeds studied, and the permanent strain is about 0.10. These figures are lower than those predicted by Haglund and Robertson [9] (0.52 and 0.32, respectively) for a similar initial bulk; for a larger initial bulk calendered at 20 kN/m they report $\varepsilon_n$ = 0.63 and $\varepsilon_p$ = 0.40.

With the nip length estimated at 7 mm [8, 10], the average pressure across the width and length of the nip at a load of 23 kN/m is roughly 3.0 MPa, or 450 psi; Chapman and Peel [6] obtained data at 1000 psi giving $\varepsilon_n$ = 0.48 and $\varepsilon_p$ = 0.22 using a platen press and a much longer residence time.

## 4.3 EXPERIMENTAL ERRORS

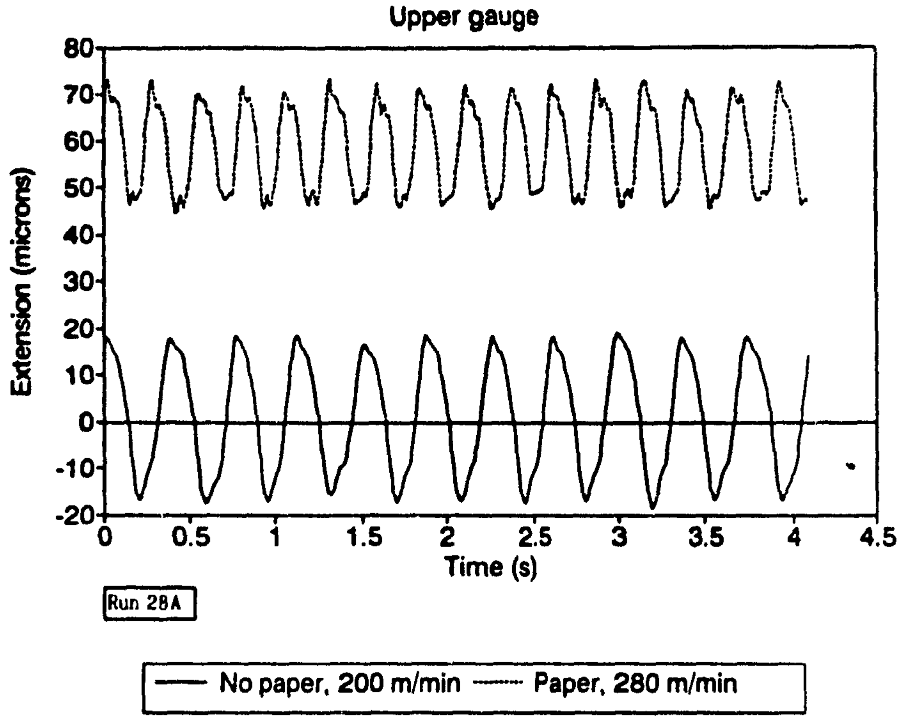
Several sources of error were identified in the computation of the various paper strains.

Errors in the measured in-nip gap were due to several problems. First, the amplitude of random noise due to power supply variations, electro-magnetic interference, dirt, etc., was ±1 μm. Second, the sinusoidal nature of the signal was eliminated by averaging over several revolutions of the lower roll. Raw signals from a typical run are plotted in Figure 4.2.2.

Before averaging of the sinusoidal signal, the standard deviation of the raw data was about 5 μm. For the purpose of error analysis, the standard deviation of the averaged nip gauge signal was estimated at 1 μm. Since the sinusoidal signal itself is unchanging over many revolutions and indeed over several days, the averaging method is a reasonable approximation.

Errors in the measurement of initial paper thickness are due to the resolution of the standard digital micrometer, and the standard deviations are therefore of the order of 0.5 μm.

Errors in the measurement of final paper thickness were caused by two problems. The first is the resolution of the standard micrometer. The second was have a small but measurable cross-machine direction thickness variation in the calendered paper (of the order of 2 μm thicker at one edge than at the other) due to slight radius variations (about 20 μm) in both rolls. These variations were caused by the grinding process.



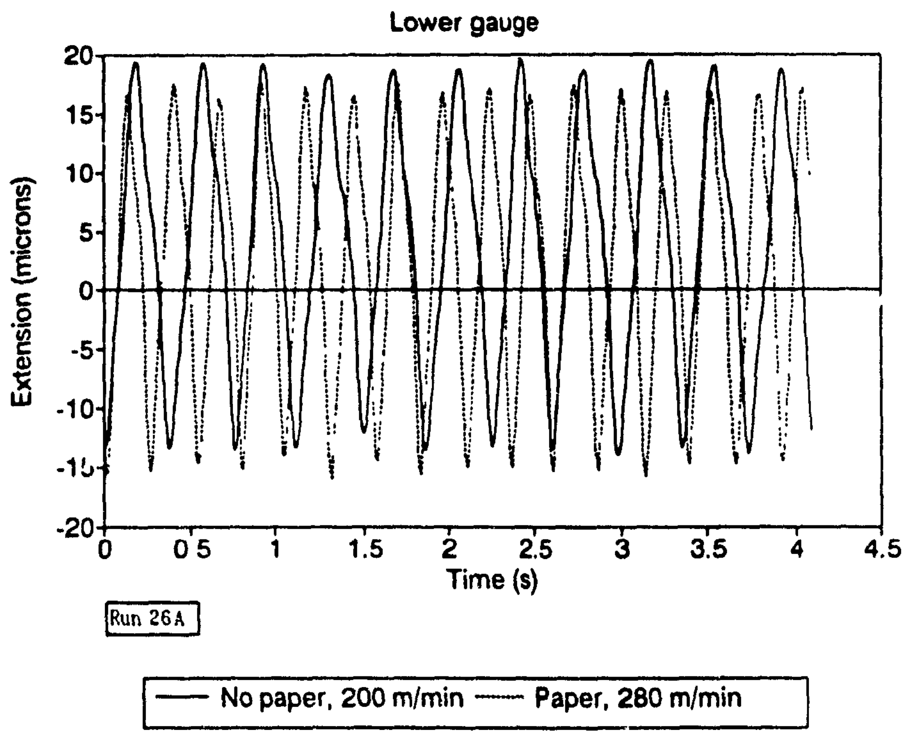




Figure 4.3.1: Extension of nip gauges in a typical run

The standard deviation in the final paper thickness was about 1.5 μm when 30 readings were taken, ten in the middle of a strip one roll circumference in length, and ten within 10 mm of the edges. To correct this the rolls were lapped together as a pair, eliminating the systematic cross-machine variation. The new standard deviation was then about 1.2 μm.

Finally the magnitude of the load slightly varied from run to run. As a result, data acquired at constant load shows a greater standard deviation than that acquired at constant speed due to the load variation shown in Table 4.2.2.

Given these problems, the standard deviations were estimated at 0.028 to 0.032 for in-nip, and 0.015 to 0.020 for permanent strain. The higher values come from the constant load experiments. Colley and Peel [7] estimate the error in their work at 0.013 for both strains, measured in a platen press rather than dynamically as in this study. Crotogino et al. [15] report a standard deviation of 0.1 $cm^3/g$ in recovered bulk; this corresponds to a strain variation of about 0.015.

The permanent strain data reproduces the values reported by other researchers [7, 9, 14, 15] and therefore meets the second objective set forth in Section 1.2, that is, to demonstrate the correct functioning of the equipment.

The in-nip data shows larger errors, partly due to difficulty with load control, and partly due to dynamic difficulties in measuring small separations between large rotating masses. Nonetheless the equipment meets the first objective, that is, accurate measurement of in-nip paper thickness under industrial conditions.

# 5. SUMMARY AND CONCLUSIONS

Calendering is the final stage in the manufacture of many grades of paper. This process reduces the surface roughness of the finished product, thus making it more suitable for printing, by pressing it between two or more large iron rolls.

Calendering also reduces the thickness of the sheet. The amount of permanent thickness reduction depends, among other things, on the initial paper properties and on the shape of the pressure pulse applied to the sheet in a calender nip. In a typical calender, both of these may vary across the width of the machine, leading to cross-machine variations in the permanent thickness reduction imposed on the paper. These variations are undesirable, since they lead to difficulties in winding the finished sheet for shipping. Thus a control system promoting sheet uniformity is necessary.

Some of the elements necessary for a closed-loop system for control of local paper thickness exist today. In particular, local permanent paper deformation is controlled by heating or cooling local areas of the calender roll, thus altering the shape of radius and pressure profiles. The relationship between local heat flux profile and the resulting roll radius profile is known [3, 4, 5], but the relationship between radius and pressure profiles is not. This relationship, which depends on the viscoelastic properties of paper, is the missing link in a complete closed-loop control system for cross-direction paper thickness in a calender.

This thesis describes the next step in filling this void, namely the design and construction of equipment suitable for the measurement of in-nip pressure and paper thickness in a calender nip. These data can then be used to determine the viscoelastic properties of paper in a calender nip.

The equipment consists of an experimental calender reproducing all of the conditions found in an industrial calender, except width. With a width of 75 mm, pressure profile variations due to roll bending or bearing deflection are effectively eliminated and precise measurement of the local nip gap is possible.

In commissioning the equipment several tests were performed; two of the calendering coefficients defined by Equation 2.2.2 were determined. The permanent paper strains measured with the new apparatus agree well with values published in the literature [14, 15]. The initial measurements of in-nip strain are consistent and repeatable, with standard deviations of the order of 5% of the measured value. Further improvements in the data processing procedure and the load control system, as detailed in Sections 3.4.3 and 4.3, are expected to improve this figure. The objectives set forth in Section 1.2 have thus been met.

The equipment may now be used to determine the viscoelastic properties of paper in a calender nip, either as an empirical expression relating local in-nip strain to local calendering parameters or as a constitutive equation based on a theoretical description of paper structure. Given these viscoelastic properties, the change in local permanent deformation due to a local control input can be determined, thus closing the control loop for cross-direction permanent strain in a paper calender.

## REFERENCES


1. de Montmorency, W.H., The calendering of newsprint: a laboratory study of temperature and pressure effects, Pulp & Paper Mag. Can. 68(7): T326-T345, 1967.

2. Crotogino, R.H., Supercalendered and conventionally calendered newsprint - A comparison of surface and printing properties, TAPPI 63(11): 101-105, 1980.

3. Pelletier, L., Impingement heat transfer on a rotating cylinder - an experimental study of calender cooling, Master's thesis, McGill University, 1984.

4. Pelletier, L., Douglas, W.J.M., and Crotogino, R.H., CD calender control with air jets - an experimental study of impingement heat transfer, J. Pulp Paper Science 13(2): 49-55, JPPS 1987.

5. Journeaux, I.A., Impinging jet heat transfer and thermal deformation for rotating calender rolls, Ph.D. Thesis, McGill University, 1990.

6. Chapman, D.L.T. and Peel, J.D., Calendering processes and the compressibility of paper (part 1), Paper Technol. 10(2): T116-T124, 1969.

7. Colley, J. and Peel, J.D., Calendering processes and the compressibility of paper (part 2), Paper Technol. 13(5): T166-T173, 1972.

8. Kerekes, R.J., Speed and loading effects in a calender nip, Trans. Tech. Sect. CPPA 2(3): TR88-TR91, 1976.

9. Haglund, L. and Robertson, G., Local Thickness Reduction in a Calender Nip - An Experimental Study, Svensk Papperstidning 77(14): 521-530, 1974.

10. Kerekes, R.J., Pressure distribution on a thin viscoelastic strip rolling between two rigid cylinders, Trans. CSME 4(1): 27-32, 1976.

11. May, W.D., Morris, E.L. and Atack, D., Rolling friction of a hard cylinder over a viscoelastic material, J. Appl. Phys. 30(11): 1713-1724, 1959.

12. Hunter, S.C., The rolling contact of a rigid cylinder with a viscoelastic half-space, J. Appl. Mechanics Trans ASME, 611-617, December 1961.

13. Alblas, J.B. and Kuipers, M., The contact problem of a rigid cylinder rolling on a thin viscoelastic layer, Int. J. Engng. Sci. 8(5): 363-380, 1970.

14. Crotogino, R.H., Towards a comprehensive calendering equation, Trans. Tech. Sect. CPPA 6(4): TR89-TR94, 1980.

15. Crotogino, R.H., Hussain, S.M. and McDonald, J.D., Mill application of the calendering equation, J. Pulp & Paper Science 9(5): TR128-TR133, 1983.

16. Hamel, J., Crotogino, R.H. and Gratton, M.F., Measurement of nip load distribution in calenders using uncalendered paper, presented at the Annual Conference of the Technical Section of the CPPA, Montreal, January 1991.

# APPENDIX 1: DATA ACQUISITION AND CONTROL PROGRAMS

```
/*                DATA ACQUISITION AND CONTROL PROGRAM                    */

/*                Written by Miles Sherman, August 1990                   */
/*                Last modification: December 11, 1990                    */
/*                        by Thomas Browne                                */

#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include <malloc.h>
#include <graph.h>
#include <ctype.h>
#include "func.lib"

/* Declare global variables  */

int huge *exp_buff;                     /* data buffer */
int      motor_bits,                    /* main motor output, bits */
         trim_bits,                     /* trim motor output, bits */
         apply_load_mask = 1,
         raise_load_mask = 2,
         brake_reel_mask = 4,
         mask,
         sensvar,
         mintrim_bits = 500,
         maxtrim_bits = 4000,
         firstrun = 1,                  /* number of runs saved in a file */
         num_chans = 8,                 /* number of channels to read */
         num_scans = 2048,              /* number of channel scans */
         TMid = 500,                    /* target windup tension, bits */
         tenschan = 5,                  /* sheet tension channel */
         tenserr[3],                    /* error history */
         readval[8];
char     file_name[50],                 /* data file name */
         par_file[10] = "info.txt";     /* parameter file name */
float    cons[3][7],                    /* calibration constants */
         Tp,                            /* proportional constant */
         Td,                            /* derivative constant  */
         Ti,                            /* integral constant */
         deltaT,
         adj[3];                        /* control parameters */
struct   dosdate_t ddate;
struct   dostime_t dtime;
struct   {
           float dry_bulb,              /* laboratory ambient temp */
                 dew_pt,                /* laboratory dew point */
                 baro,                  /* laboratory barometric pressure */
                 rel_hum,               /* laboratory relative humidity */
                 bas_wt,                /* basis weight */
                 volt_seq[7],           /* sequence of supply voltages */
```

```
               calroll_d1,
               calroll_d2,              /* calender reel diameters in m */
               i_vel;                   /* velocity in meters/min */
         int   file_no,                 /* file identification number */
               calroll_n1,
               calroll_n2,              /* calender roll id numbers */
               chan_seq[8],             /* sequence channels read */
               gain_seq[8],             /* sequence of gains */
               sens_seq[7];             /* sequence of sensor connections */
         char  pap_type[30];            /* type of paper */
         double s_rate,                 /* rate between channels, Hz */
                scan_rate;              /* rate between scans, Hz */
        } data;

/*************************************************************************/

main()
{
  FILE    *outfbin,                /* data file */
          *inftxt;                 /* parameter file */
  int     correct_data=0,
          correct_const=0,
          ans_data,                /* stores if data is correct */
          ans_cons,                /* stores if constants are correct */
          cont,
          correct,                 /* stores if motor data is correct */
          num_samples,
          endfile = 11111,         /* flag at the end of a file */
          endblock = 10000,        /* flag at the end of a block */
          temp,
          ascii,                   /* ascii value */
          i,j,
          press,                   /* value read from nip load cell */
          errNum,
          pdest,
          length;
  char    cr,                      /* carriage return */
          quit,                    /* first character entered for velocity */
          vel_char,                /* following chars entered for velocity */
          vel_temp[100],           /* temporary velocity as char string */
          first[50] = "c:\\meng\\calcs\\data\\",
          last[5]=".dta";


  _clearscreen(_GCLEARSCREEN);
  num_samples = num_scans*num_chans;
  exp_buff = NULL;                 /* allocate storage buffer */
  if  (!(exp_buff = (int huge*) halloc (((long) num_samples), sizeof(int))))
  {
    clear(17,1,2);
    perror ("error allocating memory");
```

```
  clear(5,1,2);
  printf("ending program....");
  exit(1);
}
if ((inftxt=fopen("control.txt","rt")) != NULL)
{
  fscanf(inftxt,"%f",&Tp);
  fscanf(inftxt,"%f",&Ti);
  fscanf(inftxt,"%f",&Td);
  fscanf(inftxt,"%f",&deltaT);
  fclose(inftxt);
}
else
{
  perror("error opening CONTROL.TXT");
  exit(1);
}
adj[0] = Tp + 0.5*Ti/deltaT + Td*deltaT;
adj[1] = -Tp + 0.5*Ti/deltaT - 2*Td*deltaT;
adj[2] = Td*deltaT;

if ((inftxt=fopen(par_file,"rt")) != NULL)
{
  fscanf(inftxt,"%f",&data.dry_bulb);      /* Open and read INFO.TXT */
  fscanf(inftxt,"%f",&data.dew_pt);
  fscanf(inftxt,"%f",&data baro);
  fscanf(inftxt,"%f",&data.rel_hum);
  fscanf(inftxt,"%f",&data calroll_d1);
  fscanf(inftxt,"%d",&data calroll_n1);
  fscanf(inftxt,"%f",&data.calroll_d2);
  fscanf(inftxt,"%d",&data.calroll_n2);
  fscanf(inftxt,"%f",&data.bas_wt);
  fscanf(inftxt,"%s",data.pap_type);
  fscanf(inftxt,"%lf",&data.s_rate);
  fscanf(inftxt,"%lf",&data scan_rate);
  for (i=0; i<num_chans; i++)
    fscanf(inftxt,"%d",&data chan_seq[i]);
  for (i=0; i<num_chans; i++)
    fscanf(inftxt,"%d",&data.gain_seq[i]);
  for (i=0; i<7 ; i++)
    fscanf(inftxt,"%d",&data sens_seq[i]);
  for (i=0; i<7 ; i++)
    fscanf(inftxt,"%f",&data.volt_seq[i]);
  fscanf(inftxt,"\n%f",&data.i_vel);
  fscanf(inftxt,"\n%d",&data.file_no);
  for (i=1;i<=7;i++)
    fscanf(inftxt,"%d",temp),
  for (i=0;i<=2;i++)
  {
    for (j=0;j<=6;j++)
      fscanf(inftxt,"%f",&cons[i][j]);
```

```
    }
    fclose(inftxt);
  }
  else
  {
    perror("error opening INFO.TXT");
    exit(1);
  }
  _dos_getdate(&ddate);
  _dos_gettime(&dtime);
  mask = 0;

  AO_Config(1,0,1,10.0,0);
  AO_Config(1,1,1,10.0,0);
  DIG_Prt_Config(1,0,0,0);
  DIG_Prt_Config(1,1,0,1);
  DAQ_Config(1,1,0);          /* second parameter = 0 for software trigger */
                              /*                  = 1 for hardware trigger */

  do                          /* main loop begins here */
  {
    for (i=0; i<3; i++)           /* reset trim control */
      tenserr[i] = 0;
    trim_bits = maxtrim_bits;
    motor_bits = 0;
    AO_Write(1,0,motor_bits);     /* Main motor off */
    AO_Write(1,1,trim_bits);      /* Trim motor off */
    mask = mask & 11;             /* leave load alone */
    DIG_Out_Port(1,1,mask);       /* Release brake */
    data_screen();                  /* displays current parameters */
    _settextposition(25,1);
    printf("P, I, D parameters: %f,    %f,    %f   ",Tp,Ti,Td);
    _settextposition(22,1);
    printf("          (enter choice)`.n\t>");
    clear(23,10,2);
    cr = getche();
    while ((cr != 'q') && (cr != 'Q'))
    {
      change(cr);                 /* change selected parameter */
      clear(23,10,2);
      cr = getche();
    }
    _settextposition(24,1);
    printf ("Lowering roll . . .");
    mask = apply_load_mask;                   /* lower roll */
    DIG_Out_Port(1,1,mask);
    do
      AI_Read (1,6,data.gain_seq[6],&press);
    while (press < -50);
    mask = 0;
    DIG_Out_Port(1,1,mask);
```

```
do                    /* experiment loop begins here */
{
  correct = 0;
  cont = 1;
  _clearscreen(_GCLEARSCREEN);
  _settextposition(20,1);
  printf("Control elements: %f,  %f,   %f ",adj[0],adj[1],adj[2]);
  _settextposition(1,1);
  printf("\t\t\t        START UP");
  _settextposition(3,42);
  printf("velocity: %8.2f m/min",data.i_vel);
  _settextposition(5,10);
  printf("ENTER \"Q\" IF NO VELOCITY CHANGE IS REQUIRED");
  _settextposition(6,10);
  while (!correct)          /* get sheet speed */
  {
    correct = 2;
    _dos_getdate(&ddate);
    _dos_gettime(&dtime);   /*gets current date and time*/
    _settextposition(17,1);
    printf("enter velocity(in meters/min): ");
    clearl(17,38,7);
    quit = getche();
    ascii = toascii(quit);
    if (ascii==13)           /* if enter is pressed as first char */
      _settextposition(17,38);
                            /* if velocity change required */
    if ((quit!='Q') && (quit!='q'))
    {
      strcpy(vel_temp,&quit);
      i=0;
      do
      {
        i++;
        vel_char = getche();
        ascii = toascii(vel_char);
        strncpy(vel_temp+i,&vel_char,100);
      } while ((ascii != 13) && (i != 5));
      data.i_vel = atof(vel_temp);    /* converts vel_temp to float */
      clearl(3,52,8);
      printf("%8.2f",data.i_vel);
      _settextposition(22,1);
      printf("Accept changes? (Y/N) ");
      do                      /* Get keyboard input */
      {
        if (kbhit())
        {
          correct = getche();
          correct = toupper(correct);
          if (correct==78) correct=0;
```

```
            else if (correct==89) correct=1;
          }
          if (firstrun > 1)
          {
           AI_Read(1,tenschan,data.gain_seq[tenschan],&readval[tenschan]);
           tension(readval[tenschan]);
          }
        } while (correct>1);  /* End get keyboard input */
      }                       /* End if velocity change required */
    }                         /* End get sheet speed */
    _clearscreen(_GCLEARSCREEN);
    _settextposition(1,1);
    printf("\"ESC\" - EMERGENCY EXIT");
    _settextposition(7,1);
    printf("\n\n\n0\t1\t2\t3\t4\t5\t6\t7");
    clear(15,1,1);
    _settextposition(6,1);
    printf("starting motor....");
    errNum = 0;
    if (cont = motor())          /* if motor() returns without ESC */
    {
      mask = apply_load_mask;
      DIG_Out_Port(1,1,mask);    /* apply load */
      if (cont = wait())         /* if wait() returns without ESC */
      {
        if ((cont = exprmnt(num_samples))==27)
          errNum = eexit(1);     /* if exprmnt() returns with ESC */
      }
      else errNum = eexit(2);    /* if wait() returns with ESC */
    }
    else errNum = eexit(3);      /* if motor() returns with ESC */

    if (errNum < 2)              /* save whatever data was acquired */
    {
      clear(1,1,3);           /* prevent overwriting old data file */
      clear(5,1,3);
      itoa(data.file_no, file_name,10);
      pdest =strcpy( first + 19, file_name );
      length=strlen(first);
      pdest =strcpy( first + length, last );
      pdest=strcpy( file_name , first);
      if (firstrun==1)
      {
        while ((outfbin= fopen(file_name, "rb")) != NULL)
        {
          fclose(outfbin);
          _settextposition(17,1);
          printf("Data file %d exists.",data.file_no);
          printf("File number to store the data? \n");
          scanf("%s",file_name);
          data.file_no = atoi(file_name);
```

```
          pdest =strcpy( first + 19, file_name );
          length=strlen(first);
          pdest =strcpy( first + length, last );
          pdest=strcpy( file_name , first);
        }
      }
      _settextposition(5,1);
      printf("writing file: %s\nsets of data: %d",file_name,firstrun);
      if ((outfbin = fopen(file_name, "ab")) != NULL )
      {
        for (i=0; i<8; i++)
          fwrite (&endblock, sizeof(endblock), 1, outfbin);
        fwrite (&ddate, sizeof(ddate), 1, outfbin);
        fwrite (&dtime, sizeof(dtime), 1, outfbin);
        fwrite (&data, sizeof(data), 1, outfbin);
        fwrite (cons, sizeof(cons), 1, outfbin);
        fwrite (&endblock, sizeof(endblock), 1, outfbin);
        for (i=0; i<8; i++)
          fwrite (&endblock, sizeof(endblock), 1, outfbin);
        fwrite (&exp_buff[0], sizeof(exp_buff[0]), num_samples, outfbin);
        fclose(outfbin);
      }
      firstrun++;
    }
    if (!errNum)
    {
      _clearscreen(_GCLEARSCREEN);
      printf("\"E\" - END\n\"P\" - ENTER NEW PARAMETERS"):
      printf("\n\"V\" - ENTER NEW VELOCITY");
      while ((cont != 69) && (cont != 80) && (cont != 86))
      {                           /* if E, V or P have not been pressed */
        if (kbhit())
        {
          cont = getch();
          cont = toupper(cont);
        }
        AI_Read(1,tenschan,data.gain_seq[tenschan],&readval[tenschan]);
        tension(readval[tenschan]);
      }
    }
    else cont = 69;    /* end if emergency exit */
    clear(1,1,3);
    _settextposition(7,1);
  } while (cont == 86);
                               /* End experiment loop */
} while (cont == 80);          /* End main loop */
if (errNum < 2)
{
  data.file_no++;
  write_data();                  /* Update INFO.TXT if data acquired */
  if (firstrun>1)                /* if this is not the first run */
```

```
    {
      outfbin = fopen(file_name,"ab");
      for (i=0; i<8; i++)
        fwrite(&endfile,sizeof(endfile),1,outfbin);
      fclose(outfbin);
    }
  }
  if (errNum)                    /* Don't release brake until keypressed */
  {
    _settextposition (24,60);
    printf("Press a key to exit:");
    do {} while (!kbhit());
    cont = getch();
  }
  _clearscreen(_GCLEARSCREEN);
  DIG_Out_Port(1,1,0);
  _settextposition (24,1);
  hfree(exp_buff);               /* frees allocated memory */
  AO_Write(1,0,0);               /* Main motor off */
  AO_Write(1,1,maxtrim_bits);    /* Trim motor off */
  printf ("Motors off \n");

}                                /* END MAIN PROGRAM */

/**********************************************************************/

/*                    SUBROUTINES BEGIN HERE                          */

/**********************************************************************/

int tension(tensval)

{
    tenserr[0] = TMid-tensval;
    trim_bits += adj[0]*tenserr[0] + adj[1]*tenserr[1] + adj[2]*tenserr[2];
    if (trim_bits>maxtrim_bits) trim_bits = maxtrim_bits;
    else if (trim_bits<mintrim_bits) trim_bits = mintrim_bits;
    tenserr[2]=tenserr[1];
    tenserr[1]=tenserr[0];
    AO_Write(1,1,trim_bits);      /* send specified bits to trim */
    _settextposition(23,23);
    printf("m %d    t %d   ",motor_bits,trim_bits);
}

/**********************************************************************/

/*       THIS SUBROUTINE ACCELERATES THE MOTOR TO THE DESIRED SPEED  */
/*       AND RETURNS 0 - ESCAPE WAS PRESSED                           */
/*                   1 - ESCAPE HAS NOT BEEN PRESSED                  */

int motor()
```

```
{
  int d_v = 2,                    /* change in velocity in bits */
      d_t = 10,                   /* change in time in msec */
      dd_v,                       /* change in time in msec */
      k,                          /* returned value of keybrd */
      time,                       /* time spent in loop */
      tens,                       /* sheet tension */
      over,
      count1,                     /* time when loop starts */
      count2,                     /* time when loop finishes */
      v,                          /* desired velocity in bits */
      again=1;
  char cr;
  double bittovel = 0.017686963;  /* see Design book Vol 4, p13 */
                    /* 2.87711779054;  20/32, 16" roll */
                    /* 2.697297930;     40/60, 16" roll */
                    /* 5.993999540;     18/60, 16" roll */
  /*Old value = 3.08244413969; older value = 3.3847405885; */

  bittovel = bittovel*48*(60/30)/0.40375;
  v = (data.i_vel*bittovel) * 1;          /* requested speed, bits */
  if (v>4095) v=4095;
  if (motor_bits>v) dd_v = -d_v;
  else dd_v = d_v;
  while (!((motor_bits >= (v-abs(d_v)/2))
      && (motor_bits <= (v+abs(d_v)/2))) && (again))
  {
    time=0;
    CTR_EvCount(1,5,4,1);        /* starts timer */
    CTR_EvRead(1,5,&over,&count1);
    while ((time<d_t) && (again))
    {
      k=keybrd();                /* reads key if pressed */
      if (k==27) again=0;
      sens();                    /* reads and displays */
      CTR_EvRead(1,5,&over,&count2);
      time=count2-count1;
    }
    motor_bits=motor_bits+dd_v;
    if (motor_bits<0) motor_bits=0;
    AO_Write(1,0,motor_bits);
  }
  data.i_vel = motor_bits/bittovel;
  if (data.i_vel < 1) data.i_vel=0;
  CTR_Stop(1,5);                  /* stop the timer */
  return again;
}

/**********************************************************************/

int exprmnt(int num)
```

```
{
  int samp_tb,
      scan_tb,
      status,
      status1,
      monitor_buff[8],          /* array which stores last scanned data */
      newpntind,
      k = 1,
      errnum,
      i;
  unsigned int samp_int,      /* intervals between acqs in timebase mode */
               scan_int,      /* interval between scan of num channels */
               retrieved;     /* number of points of data retrieved */

  clear(6,1,2);
  printf("commencing experiment....");
  clear(15,1,4);
    k=0;
    DAQ_Rate (data.s_rate, 0, &samp_tb, &samp_int);
    DAQ_Rate (data.scan_rate, 0, &scan_tb, &scan_int);
    SCAN_Setup(1,num_chans,data.chan_seq,data.gain_seq);
    if ((errnum = SCAN_Start(1, exp_buff, num, samp_tb, samp_int,scan_tb,
                              scan_int)) != 0)
    {
      _settextposition(6,1);
      printf ("Acquisition error in SCAN_Scart %d\n", errnum);
      k=4;
      return k;
    }
    _settextposition(23,1);
    printf("\nNumber of samples acquired: ");
    do
    {
      if ((errnum = DAQ_Check( 1, &status, &retrieved )) != 0)
      {
        printf ("Acquisition error in DAQ_Check %d\n", errnum);
        k=4;
        return k;
      }
      if ((retrieved > num_chans) && (retrieved < (num_scans-num_chans))
         && (!status))
      {
        if ((errnum =
          DAQ_Monitor(1,-1,0,num_chans,monitor_buff,&newpntind,&status1))
          != 0)
          printf ("Acquisition error in DAQ_Monitor %d\n", errnum);
        _settextposition(13,1);
        for (i=0; i<num_chans; i++)
          printf("%d\t",monitor_buff[i]);
        tension(monitor_buff[5]);
```

```
      }
      _settextposition(24,29);
      printf("%5u\b\b\b\b\b",retrieved);
      k=keybrd();
    }while ((!status) && (k!=27) && (k!=80) && (k!=86) &&
                        (k!=112) && (k!=118));
  DAQ_Clear(1);
  return k;
}

/**********************************************************************/

sens()
{
  _settextposition(13,1);
  for (sensvar=0;sensvar<num_chans;sensvar++)
  {
    AI_Read(1,sensvar,data.gain_seq[sensvar],&readval[sensvar]);
    printf("%d\t",readval[sensvar]);
  }
  tension (readval[tenschan]);
}

/**********************************************************************/

write_data()                    /* update parameter file */
{
  FILE *x;
  int i,
      j,
      endblock = 10000;

  if ((x=fopen(par_file,"wt")) != NULL)
  {
    fprintf(x,"%5.3f\n",data.dry_bulb);
    fprintf(x,"%5.3f\n",data.dew_pt);
    fprintf(x,"%5.3f\n",data.baro);
    fprintf(x,"%5.3f\n",data.rel_hum);
    fprintf(x,"%5.3f\n",data.calroll_d1);
    fprintf(x,"%d\n",data.calroll_n1);
    fprintf(x,"%5.3f\n",data.calroll_d2);
    fprintf(x,"%d\n",data.calroll_n2);
    fprintf(x,"%5.3f\n",data.bas_wt);
    fprintf(x,"%s\n",data.pap_type);
    fprintf(x,"%8.2lf\n",data.s_rate);
    fprintf(x,"%8.2lf\n",data.scan_rate);
    for (i=0; i<num_chans; i++)
      fprintf(x,"%1d\t",data.chan_seq[i]);
    fprintf(x,"\n");
    for (i=0; i<num_chans; i++)
      fprintf(x,"%3d\t",data.gain_seq[i]);
```

```
    fprintf(x,"\n");
    for (i=0; i<7 ; i++)
      fprintf(x,"%1d\t",data.sens_seq[i]);
    fprintf(x,"\n");
    for (i=0; i<7 ; i++)
      fprintf(x,"%5.2f\t",data.volt_seq[i]);
    fprintf(x,"\n%6.2f",data.i_vel);
    fprintf(x,"\n%.4d\n",data.file_no);
    for (i=1;i<=7;i++)
      fprintf(x,"%d",endblock);
    fprintf(x,"\n");
    for (i=0;i<=2;i++)
    {
      for (j=0;j<=6;j++)
      {
        fprintf(x,"%3.9f\t",cons[i][j]);
      }
      fprintf(x,"\n");
    }
  }
  else
  {
    perror("write error: can't open INFO.TXT for writing");
    exit(1);
  }

  fclose(x);
}

/*****************************************************************/

data_screen()

{
  int i;

  _clearscreen(_GCLEARSCREEN);
  printf("                DATE:%u/%u/%u\tTIME:%u:%u:%u\n",ddate.day,
                        ddate.month,ddate.year,dtime.hour,
                        dtime.minute,dtime.second);
  printf("Q. THESE VALUES ARE CORRECT\n");
  printf("1. Dry bulb temperature                  %5.3f  \n",data.dry_bulb);
  printf("2. Dew point temperature                 %5.3f \n",data.dew_pt);
  printf("3. Barometric pressure                   %5.3f \n",data.baro);
  printf("4. Relative humidity                     %5.3f \n",data rel_hum);
  printf("5. top calender reel - diameter (cm):   %5.3f \n",data.calroll_d1);
  printf("6.                       - id #:          %d \n",data.calroll_n1);
  printf("7. bottom calender reel -diameter (cm) %5.3f \n",data.calroll_d2);
  printf("8.                        -id #:          %d \n",data.calroll_n2);
  printf("9. basis weight (Kg):                    %5.3f \n",data.bas_wt);
  printf("10. paper type:                           %s\n",data.pap_type);
```

```
  printf("11. scan rate-one channel (scan/sec): %8.2lf\n",data.s_rate);
  printf("12.          -all channels (scan/sec):%8.2lf\n",data.scan_rate);
  printf("13.channel v:");
  printf("  %1d        ",data.chan_seq[0]);
  for (i=1; i<num_chans; i++)
    printf("%1d        ",data.chan_seq[i]);
  printf("\n14.gain v: ");
  printf("  %3d      " ,data.gain_seq[0]);
  for (i=1; i<num_chans; i++)
    printf("%3d      ",data.gain_seq[i]);
  printf("\n15.sensor v: ");
  printf("  %1d        ",data.sens_seq[0]);
  for (i=1; i<7 ; i++)
    printf("%1d        ",data.sens_seq[i]);
  printf("\n16.volts v: ");
  printf("  %5.2f  ",data.volt_seq[0]);
  for (i=1; i<7 ; i++)
    printf(" %5.2f  ",data.volt_seq[i]);
  printf("\n17. initial velocity (m/min):           %6.2f",data.i_vel);
  printf("\n18. file number:                          %.4d",data.file_no);
}

/*********************************************************************/

int change(ch1)

char ch1;                           /* first character of inputted choice */

  {
    int   ascii,                    /* ascii value */
          x,
          i,
          endfile = 11111;              /* buffer signifying end of file */
    char  temp[100],
          cr,
          ch2,                      /* second character for choice */
          y[3];                     /* final choice to input data */
    FILE  *outfbin;                /* file for storing data acquisition */

    strcpy(y, &ch1);
    ch2 = getche();
    strcpy(y + 1, &ch2);
    ascii=toascii(ch2);
    while (ascii != 13)
    {
      ch2 = getch();
      ascii=toascii(ch2);
    }
    x=atoi(y);
    if ((x>0) && (x<=18))
    {
```

```
_settextposition(24,6);
printf("%d. ",x);
if ((x!=11) && (x!=12))
  scanf("%s", temp );
if (x==1)
  data.dry_bulb=atof( temp );
else if (x==2)
  data.dew_pt = atof( temp );
else if (x==3)
  data.baro = atof( temp );
else if (x==4)
  data.rel_hum = atof( temp );
else if (x==5)
  data.calroll_d1 = atof( temp );
else if (x==6)
  data.calroll_n1 = atoi( temp );
else if (x==7)
  data.calroll_d2 = atof( temp );
else if (x==8)
  data.calroll_n2 = atoi( temp );
else if (x==9)
  data.bas_wt = atoi( temp );
else if (x==10)
  strcpy(data.pap_type, temp);
else if (x==11)
{
  scanf("%lf",&data.s_rate);
  clearl(13,39,8);
  printf("%8.2lf",data.s_rate);
}
else if (x==12)
{
  scanf("%lf",&data.scan_rate);
  clearl(14,39,8);
  printf("%8.2lf",data.scan_rate);
}
else if (x==13)
{
  data.chan_seq[0] = atof( temp );
  clearl(15,16,8);
  printf("%1d",data.chan_seq[0]);
  for (i=1;i<num_chans;i++)
  {
    _settextposition(24,10);
    scanf("%s", temp );
    data.chan_seq[i] = atoi( temp );
    clearl(15,(16+8*i),8);
    printf("%1d",data.chan_seq[i]);
  }
}
else if (x==14)
```

```
{
  data.gain_seq[0] = atof( temp );
  clearl(16,13,8);
  _settextposition(16,16);
  printf("%ld",data.gain_seq[0]);
  for (i=1;i<num_chans;i++)
  {
    _settextposition(24,10);
    scanf("%s", temp );
    data.gain_seq[i] = atoi( temp );
    clearl(16,(13+8*i),8);
    _settextposition(16,(16+8*i));
    printf("%ld",data.gain_seq[i]);
  }
}
else if (x==15)
{
  data.sens_seq[0] = atoi( temp );
  clearl(17,16,8);
  printf("%ld",data.sens_seq[0]);
  for (i=1;i<7;i++)
  {
    _settextposition(24,10);
    scanf("%s", temp );
    data.sens_seq[i] = atoi( temp );
    clearl(17,(16+8*i),8);
    printf("%ld",data.sens_seq[i]);
  }
}
else if (x==16)
{
  data.volt_seq[0] = atof( temp );
  clearl(18,15,8);
  printf("%5.2f",data.volt_seq[0]);
  for (i=1;i<7;i++)
  {
    _settextposition(24,10);
    scanf("%s", temp );
    data.volt_seq[i] = atof( temp );
    clearl(18,(15+8*i),8);
    printf("%5.2f",data.volt_seq[i]);
  }
}
else if (x==17)
  data.i_vel = atof( temp );
else if (x==18)
{
  data.file_no = atoi( temp );
  if (firstrun>1)
  {
    outfbin = fopen(file_name,"ab");
```

```
          for (i=1;i<=8;i++)
            fwrite(&endfile,sizeof(endfile),1,outfbin);
          fclose(outfbin);
        }
        firstrun = 1;
      }
      if ((x>=1) && (x<=10) && (x!=0) || (x==17) || (x==18))
      {
        clearl((x+2),40,12);
        _settextposition((x+2),43);
        printf("%s",temp);
      }
    }
}

/*********************************************************************/

int eexit(int x)
{
  DIG_Out_Port(1,1,raise_load_mask);
  DIG_Out_Port(1,1,brake_reel_mask);
  AO_Write(1,0,0);
  AO_Write(1,1,4000);
  motor_bits= 0;
  trim_bits = 4000;
  _settextposition(21,60);
  printf ("Emergency stop");
  _settextposition(22,60);
  printf ("error # %d",x);
  _settextposition(23,60);
  if (x==1)
    printf("subroutine EXPRMNT");
  else if (x==2)
    printf("subroutine WAIT");
  else if (x==3)
    printf("subroutine MOTOR");
  else if (x==4)
    printf("PANIC bar pressed");
  return x;
}

/*********************************************************************/

/*      THIS SUBROUTINE WAITS WHILE SENSORS ARE BEING READ            */
/*      UNTIL "S" OR ESCAPE HAS BEEN PRESSED                          */

int wait()
{
  int k=0;
  clear(14,1,5);
  clear(2,1,5);
```

```
  printf("\"S\" - continue ");
  _settextposition(17,1);
  printf("requested speed reached of %6.2f m/min \n",data.i_vel);
  printf("adjust load and type \"s\" when you are ready to begin");
  while ((k!=27) && (k!=115) && (k!=83))
  {
    sens();
    k=keybrd();           /* checks if calender need to be shut off  */
  }
  if (k==27) k=0;
  else k=1;
  return k;
}

/************************************************************************/

/*      CHECKS IF ESCAPE HAS BEEN PRESSED , EMERGENCY BRAKE         */
/*      HAS BEEN INITIATED OR IF SHEET BREAK HAS OCCURED            */
/*      (RETURNS 27 IF ANY OF THE ABOVE HAVE OCCURED )              */
/*      OTHERWISE RETURNS ASCII VALUE OF PRESSED KEY                */

int keybrd()
{
  int state0=0,                /* value of digital input  # 26 */
      kbdval;                  /* ascii value of key */

  DIG_In_Line(1,0,0,&state0);  /* reads panic bar pressed */
  if (kbhit()) kbdval = getch();
  if (state0==0)
    kbdval = 27;
  return kbdval;
}


/*          THIS VERSION OF TEXT.C READS FILES CREATED BY            */
/*                  ACQUIRE.C AFTER SEPT. 12,1990                    */

/* In particular, it expects 2048 data points from each of 8 channels,
   rather than 3000 data points from each of 7 channels.            */

/* Written by Miles Sherman, August 1990                             */
/* Last modification: Tom Browne, October 9, 1990                    */
/* Averaging routine and extraction added October 9th, 1990          */
/* Modified again, November 6, 1990                                  */
/* Modified again, January 8 1991                                    */

#include <stdio.h>
#include <DOS.h>
#include <conio.h>
#include "txlib.c"
```

```
main()
{
  FILE *dtabin;
  int end[8],
      scan,
      chan,
      trial = 0,
      file_no,
      length,
      pdest;
  char filename[50],
       file_[10],
       ans = 'y';

  printf("Enter file number\n");
  scanf("%d",&file_no);
  itoa(file_no,file_,10);
  strcpy(filename,file_);
  pdest = strcpy( first + 19, filename );
  length = strlen(first);
  pdest = strcpy( first + length, dta );
  pdest = strcpy( filename , first);
  printf("reading file: %s\n",filename);

  if ((dtabin = fopen(filename, "rb")) == NULL)
  {
    printf("error opening file");
    exit(1);
  }

  fread (end, sizeof(end), 1, dtabin);
  do
  {
    trial++;
    printf("reading trial.....%d\n",trial);
    fread (&ddate, sizeof(ddate), 1, dtabin);
    fread (&dtime, sizeof(dtime), 1, dtabin);
    fread (&data, sizeof(data), 1, dtabin);
    fread (cons, sizeof(cons), 1, dtabin);
    fread (&amount, sizeof(amount), 1, dtabin);
    fread (end, sizeof(end), 1, dtabin);
    fread (inbuff, sizeof(inbuff), 1, dtabin);
    fread (end, sizeof(end), 1, dtabin);

    printf("\nSave the data as a PRN file?\n");
    ans=getche();
    if (ans=='y' || ans=='Y')
    {
      true_speed();
      average();
      svetxt(file_);
```

```
      }

    } while (end[7] != 11111);
    fclose(dtabin);
  }
```

```
/*              THIS VERSION OF TEXT.C READS FILES CREATED BY              */
/*                    ACQUIRE.C AFTER SEPT. 12,1990                         */

/* In particular, it expects 2048 data points from each of 2 channels,
   rather than 3000 data points from each of 7 channels.                  */

/* Written by Miles Sherman, August 1990                                   */
/* Last modification: Tom Browne, October 9, 1990                          */
/* FFT subroutine inserted December 1990                                   */

#include <stdio.h>
#include <dos.h>
#include <math.h>
#include "nrutil.c"
#include "four1.c"
#include "txlib.c"

main()
{
  FILE *dtabin;
  int end[8],
      trial = 0,
      file_no,
      length,
      pdest;
  char filename[50],
       file_[10],
       ans = 'y';

  fftbuff = vector(1,num_fft);

  printf("Enter file number:  ");
  scanf("%d",&file_no);
  itoa(file_no,file_,10);

  printf("Enter channel:      ");
  scanf("%d",&nip);

  strcpy(filename,file_);
  pdest = strcpy( first + 19, filename );
  length = strlen(first);
  pdest = strcpy( first + length, dta );
  pdest = strcpy( filename , first);
  printf("\nreading file: %s",filename);

  if ((dtabin = fopen(filename, "rb")) == NULL)
  {
    printf("error opening file");
    exit(1);
  }
```

```
    fread (end, sizeof(end), 1, dtabin);
    do
    {
      trial++;
      printf("\nreading trial.....%d",trial);
      fread (&ddate, sizeof(ddate), 1, dtabin);
      fread (&dtime, sizeof(dtime), 1, dtabin);
      fread (&data, sizeof(data), 1, dtabin);
      fread (cons, sizeof(cons), 1, dtabin);
      fread (&amourt, sizeof(amount), 1, dtabin);
      fread (end, sizeof(end), 1, dtabin);
      fread (inbuff, sizeof(inbuff), 1, dtabin);
      fread (end, sizeof(end), 1, dtabin);

      printf("\nSave the data as an FFT file?  ");
      ans=getche();
      if (ans=='y' || ans=='Y')
      {
        true_speed();
        svefft(file_);
      }
    } while (end[7] != 11111);

    free_vector (fftbuff,1,num_fft);
    fclose(dtabin);
}

/******************************************************************/

svefft(file1)
char file1[50];
{
  FILE *dtatxt;
  int length,
      pdest,
      divisor,
      j, k;
  char ext[5],
       file[50],
       new_ext;

  do
  {
    new_ext='h';
    strcpy (first,"c:\\meng\\calcs\\data\\");
    strcpy (fft,".fft");
    strcpy(file,file1);

    printf("\nFile extension (a, b, etc)                          ");
    scanf("%s",ext);
```

```
      pdest =strcpy( first + 19, file );
      length=strlen( first);
      pdest =strcpy( first + length, ext );
      length=strlen( first);
      pdest =strcpy( first + length, fft );
      pdest=strcpy( file , first);

      if ((dtatxt = fopen(file, "rt")) != NULL)
      {
        printf("\n\nfile already exists - try a new extension? (y/n)  ");
        while ((new_ext!='y') && (new_ext!='n') && (new_ext!='Y') &&
(new_ext!='N'))
          new_ext=getche();
        fclose(dtatxt);
      }
    }while ((new_ext=='Y') || (new_ext=='y'));

    if (new_ext=='h')
    {
      dtatxt = fopen(file, "wt");
      save_data(file,ext,dtatxt);
      for (scan=0;scan<num_scans;scan++)
      {
        fftbuff[2*scan+1]=
          (float)(cons[0][nip]+cons[1][nip]*inbuff[num_chans*scan+nip]);
        fftbuff[2*scan+2]=0.0;
      }
      printf("\nBegining FFT . . . \n");
      four1(fftbuff,num_scans,isign);
      for (scan=1;scan<(num_scans/2)+1;scan++)
      {
        magn = 2.0*sqrt(pow(fftbuff[2*scan],2.0)+
          pow(fftbuff[2*scan+1],2.0))/num_scans;
        theta = atan2(fftbuff[2*scan],fftbuff[2*scan+1]);
        fprintf(dtatxt,"%d    %10.7f    %10.4f\n",scan,magn,theta);
      }
      fclose(dtatxt);
    }
}
```

```
/*          SUBROUTINES FOR ALL VFRSIONS OF TX???.C                    */

#define num_chans 8
#define num_chans_saved 2
#define num_scans 2048
#define num_scans_saved 2048
#define num_avg 4
#define num_saved 512
#define num_fft (2*num_scans)
#define num_samples (num_chans*num_scans)
#define isign 1

int inbuff[num_samples],
    outbuff[num_saved*num_chans],
    nip,
    trigger,
    scan,
    chan;
float cons[3][7];
double correct_vel,
       magn,
       minmagn = 0.5,
       theta;
float *fftbuff;
char  first[50] = "c:\\meng\\calcs\\data\\",
       prn[5] = ".prn",
       sig[5] = ".sig",
       fft[5] = ".fft",
       dta[5] = ".dta";
unsigned int amount;

struct dosdate_t ddate;
struct dostime_t dtime;
struct {
    float Tdry,       /*initial paper reel weight in kg*/
          Twet,       /*final paper reel weight in kg*/
          BarP,          /*initial moisture content*/
          RelH,          /*final moisture content*/
          bas_wt,          /*basis weight*/
          volt_seq[7],     /*sequence of supply voltages*/
          calroll_d1,
          calroll_d2,      /*calender reel diameters*/
          i_vel;           /* desired velocity in meters/min */
    int file_no,           /*file identification number*/
        calroll_n1,

        calroll_n2,         /*calender roll id numbers*/
        chan_seq[8],       /*sequence channels read*/
        gain_seq[8],       /*sequence of gains*/
        sens_seq[7];       /*sequence of sensor connections*/
    char pap_type[30];     /*type of paper*/
```

```
    double s_rate,        /* rate samples are scanned (scan/sec) */
           scan_rate;     /* rate to repeat each scan (scan/sec) */
  }data;

/*****************************************************************/

true_speed()
{
  float delta_t;
  int trig,
      t1,
      init;

  init = (inbuff[7]>1000)? 1 : 0;
  scan = 0;
  delta_t = 1/data.scan_rate;
  do
  {
    scan++;
    trig = (inbuff[num_chans*scan+7]>1000)? 1 : 0;
  } while (trig==init);                          /* find first edge */
  t1=scan;                                       /* start timer */
  do
  {
    scan++;
    trig = (inbuff[num_chans*scan+7]>1000)? 1 : 0;
  } while (trig!=init);                          /* find second edge */
  do
  {
    scan++;
    trig = (inbuff[num_chans*scan+7]>1000)? 1 : 0;
  } while (trig==init);                          /* find third edge */
  correct_vel = 60*3.141592654*data.calroll_d2/((scan-t1)*delta_t);
}

/*****************************************************************/

average()
{
  int index,
      ch,
      sc,
      i;

  for (ch=0;ch<num_chans-1;ch++)
  {
    outbuff[ch]=0;
    for (i=0;i<num_avg+1;i++)
      outbuff[ch] += inbuff[i*num_chans+ch];
  }
  outbuff[num_chans-1] = inbuff[num_chans-1];
```

```
  for (sc=1;sc<(num_saved-1);sc++)
  {
    for (ch=0;ch<num_chans-1;ch++)
    {
      index = sc*num_chans+ch;
      outbuff[index]=0;
      for (i=-num_avg;i<num_avg+1;i++)
        outbuff[index] += inbuff[(num_avg*sc+i)*num_chans+ch];
    }
    outbuff[index+1] = inbuff[(num_avg*sc+1)*num_chans-1];
  }
  for (ch=0;ch<num_chans-1;ch++)
  {
    index = (num_saved-1)*num_chans+ch;
    outbuff[index]=0;
    for (i=-num_avg;i<num_avg;i++)
      outbuff[index] += inbuff[(num_avg*(num_saved-1)+i)*num_chans+ch];
  }
  outbuff[index+1] = inbuff[(num_avg*sc+1)*num_chans-1];
}

/***************************************************************/

svetxt(file1)
char file1[50];
{
  FILE *dtatxt;
  int length,
      pdest,
      divisor,
      ch,
      sc,
      j, k;
  double writebuff;
  char ext[5],
       file[50],
       new_ext;

  do
  {
    new_ext='h';
    strcpy (first,"c:\\meng\\calcs\\data\\");
    strcpy (prn,".prn");
    strcpy(file,file1);

    printf("\n\nFile extension (a, b, etc)\n");
    scanf("%s",ext);

    pdest =strcpy( first + 19, file );
    length=strlen(first);
    pdest =strcpy( first + length, ext );
```

```
    length=strlen(first);
    pdest =strcpy( first + length, prn );
    pdest=strcpy( file , first);

    if ((dtatxt = fopen(file, "rt")) != NULL)
    {
      printf("\n\nfile already exists - try a new extension?");
      printf("\n\t(y/n)\n");
      while ((new_ext!='y') && (new_ext!='n') && (new_ext!='Y') &&
(new_ext!='N'))
         new_ext=getche();
      fclose(dtatxt);
    }
  }while ((new_ext=='Y') || (new_ext=='y'));

  if (new_ext=='h')
  {
    dtatxt = fopen(file, "wt");
    save_data(file,ext,dtatxt);
    divisor = num_avg+1;
    for (ch=0;ch<num_chans-1;ch++)
    {
      writebuff = cons[0][ch]+cons[1][ch]*outbuff[ch]/divisor;
      fprintf(dtatxt," %4.8f,",writebuff);
    }
    trigger = (outbuff[num_chans-1]>1000)? 1 : 0;
    fprintf(dtatxt," %d  \n",trigger);

    divisor = 2*num_avg+1;
    for (sc=1;sc<(num_saved-1);sc++)
    {
      for (ch=0;ch<num_chans-1;ch++)
      {
        writebuff =
            cons[0][ch]+cons[1][ch]*outbuff[num_chans*sc+ch]/divisor;
        fprintf(dtatxt," %4.8f,",writebuff);
      }
      trigger = (outbuff[num_chans*sc-1]>1000)? 1 : 0;
      fprintf(dtatxt," %d  \n",trigger);
    }
    divisor = 2*num_avg;
    for (ch=0;ch<num_chans-1;ch++)
    {
      writebuff =
        cons[0][ch]+cons[1][ch]*outbuff[num_chans*(num_saved-1)+ch]/divisor;
      fprintf(dtatxt," %4.8f,",writebuff);
    }
    trigger = (outbuff[num_chans*(num_saved-1)-1]>1000)? 1 : 0;
    fprintf(dtatxt," %d  \n",trigger);

    fclose(dtatxt);
```

```
  }
}

/*****************************************************************/
/* TXSIGOLD */

svesig(file1)
char file1[50];
{
  FILE *dtatxt;
  int length,
      pdest,
      divisor,
      j, k;
  double writebuff;
  char ext[5],
       file[50],
       new_ext;

  do
  {
    new_ext='h';
    strcpy (first,"c:\\meng\\calcs\\data\\");
    strcpy (sig,".sig");
    strcpy(file,file1);

    printf("\nFile extension (a, b, etc)                        ");
    scanf("%s",ext);

    pdest =strcpy( first + 19, file );
    length=strlen(first);
    pdest =strcpy( first + length, ext );
    length=strlen(first);
    pdest =strcpy( first + length, sig );
    pdest=strcpy( file , first);

    if ((dtatxt = fopen(file, "rt")) != NULL)
    {
      printf("\n\nfile already exists - try a new extension? (y/n)  ");
      while ((new_ext!='y') && (new_ext!='n') && (new_ext!='Y') &&
(new_ext!='N'))
        new_ext=getche();
      fclose(dtatxt);
    }
  }while ((new_ext=='Y') || (new_ext=='y'));

  if (new_ext=='h')
  {
    dtatxt = fopen(file, "wt");
    save_data(file,ext,dtatxt);
```

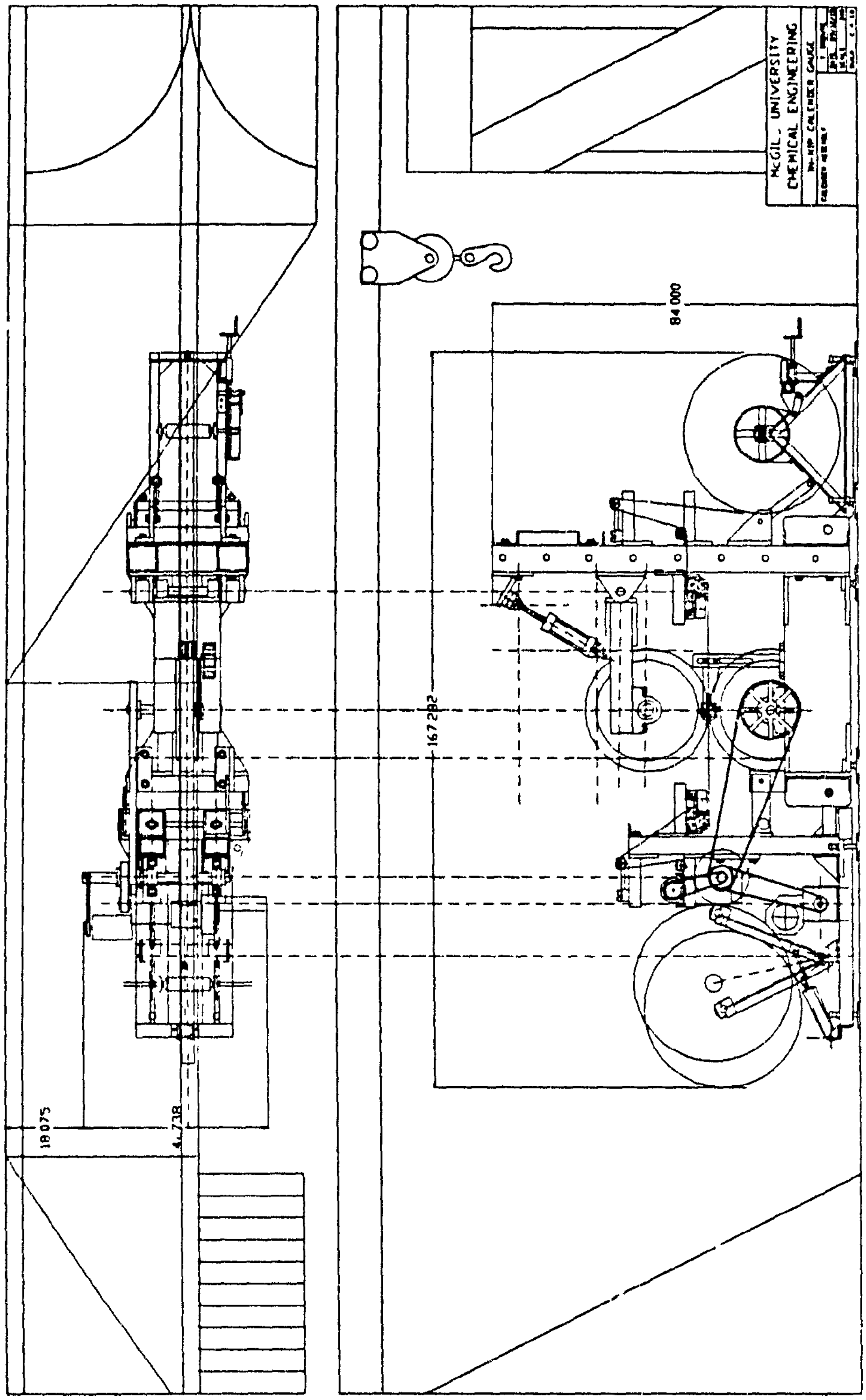
```
    for (scan=0;scan<num_scans_saved;scan++)
    {
      for (chan=0;chan<num_chans_saved;chan++)
      {
        writebuff=cons[0][chan]+(cons[2][chan]*inbuff[num_chans*scan+chan]+
          cons[1][chan])*inbuff[num_chans*scan+chan];
        fprintf(dtatxt," %4.8f,",writebuff);
      }
      trigger = (inbuff[num_chans*(scan+1)-1]>1000)? 1 : 0;
      fprintf(dtatxt," %d  \n",trigger);
    }

    fclose(dtatxt);
  }
}

/***********************************************************************/

save_data(file,ext,dtatxt)
char file[50],
     ext[5];
FILE *dtatxt;
{
int j, k;
    printf("writing file: %s\n",file);

    fprintf(dtatxt,"\"DATE: \"\"%u/%u/%u\"\t\"TIME:
\"\"%u/%u/%u\"\n",ddate.day,
             ddate.month,ddate.year,dtime.hour, dtime minute,dtime.second),
    fprintf(dtatxt,"\"Dry bulb  \"      %5.3f   \n",data.Tdry),
    fprintf(dtatxt,"\"Dew point \"      %5.3f \n",data.Twet);
    fprintf(dtatxt,"\"Barometer \"      %5.3f \n",data BarP);
    fprintf(dtatxt,"\"R.H.      \"      %5.3f \n",data RelH);
    fprintf(dtatxt,"\"TrollD    \"   %5.3f \n",data.calroll_d1),
    fprintf(dtatxt,"\"id        \"    %d \n",data.calroll_n1);
    fprintf(dtatxt,"\"BrollD    \" %5.3f \n",data calroll_d2),
    fprintf(dtatxt,"\"id        \"   %d \n",data.calroll_n2),
    fprintf(dtatxt,"\"BW        \"  %5.3f \n",data.bas_wt);
    fprintf(dtatxt,"\"paper type\"\" %s\"\n",data.pap_type);
    fprintf(dtatxt,"\"CRate     \" %8.2lf\n",data s_rate);
    fprintf(dtatxt,"\"SRate     \"%8.2lf,
%8 2lf\n",data.scan_rate,data.scan_rate/num_avg);
    fprintf(dtatxt,"\"channel v:\"");
    for (j=0; j<8 ; j++)
      fprintf(dtatxt,"%1d       ",data.chan_seq[j]);
    fprintf(dtatxt,"\n\"gain v: \"");
    for (j=0; j<8 ; j++)
      fprintf(dtatxt,"%3d      ",data.gain_seq[j]);
    fprintf(dtatxt,"\n\"sensor v: \"");
    for (j=0; j<7 ; j++)
      fprintf(dtatxt,"%1d       ",data.sens_seq[j]);
```

```
    fprintf(dtatxt,"\n\"volts v:\" ");
    for (j=0; j<7 ; j++)
      fprintf(dtatxt," %5.2f  ",data.volt_seq[j]);
    fprintf(dtatxt,"\n\"Vel\"         %6.2f  %6.2f",data.i_vel,correct_vel);
    fprintf(dtatxt,"\n\"file:\" %.4d  \" %s \"\n",data.file_no, ext);
    fprintf(dtatxt,"\n\"calibration \" \n");
    for (j=0;j<3;j++)
    {
      for (k=0;k<7;k++)
        fprintf(dtatxt,"  %.14f, ",cons[j][k]);
      fprintf(dtatxt,"\n");
    }
    fprintf(dtatxt,"\n\n");
}
```

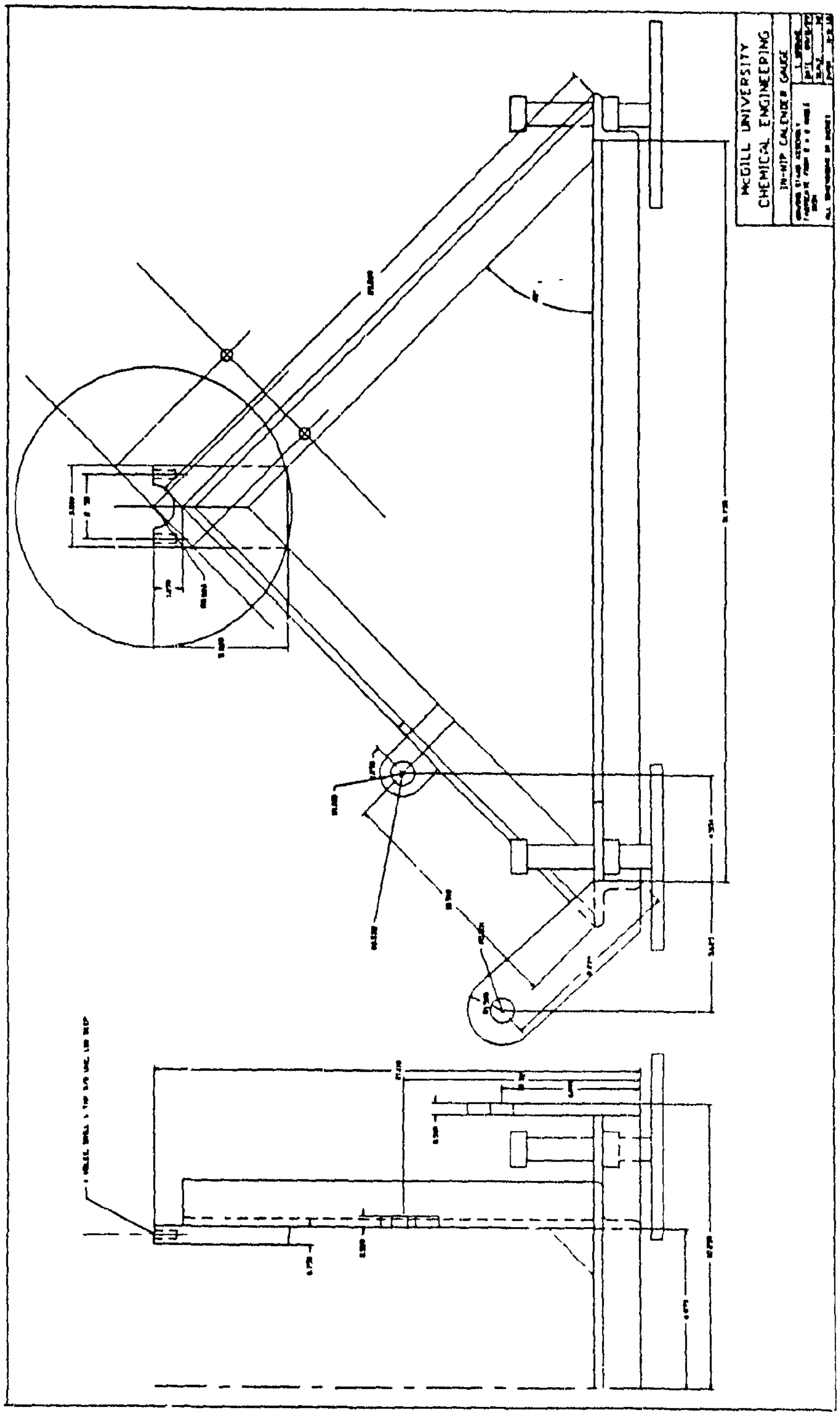
```
/* Fast Fourier Transform subroutine */
/* from Numerical Recipes, by William H. Press, Brian P. Flannery,
                                 Saul A. Teukolsky and William T. Vetterling,
        Cambridge University Press */

#define SWAP(a,b) tempr=(a);(a)=(b);(b)=tempr

void four1(data,nn,isign)
float data[];
int nn,isign;
{
  int n,mmax,m,j,istep,i;
  double wtemp,wr,wpr,wpi,wi,theta;
  float tempr,tempi;
  n=nn << 1;
  j=1;
  for (i=1;i<n;i+=2)
  {
    if (j > i)
    {
      SWAP(data[j],data[i]);
      SWAP(data[j+1],data[i+1]);
    }
    m=n >> 1;
    while (m >= 2 && j > m)
    {
      j -= m;
      m >>= 1;
    }
    j += m;
  }
  mmax=2;
  while (n > mmax)
  {
    istep=2*mmax;
    theta=6.28318530717959/(isign*mmax);
    wtemp=sin(0.5*theta);
    wpr = -2.0*wtemp*wtemp;
    wpi=sin(theta);
    wr=1.0;
    wi=0.0;
    for (m=1;m<mmax;m+=2)
    {
      for (i=m;i<=n;i+=istep)
      {
        j=i+mmax;
        tempr=wr*data[j]-wi*data[j+1];
        tempi=wr*data[j+1]+wi*data[j];
        data[j]=data[i]-tempr;
        data[j+1]=data[i+1]-tempi;
        data[i] += tempr;
```

```
          data[i+1] += tempi;
        }
        wr=(wtemp=wr)*wpr-wi*wpi+wr;
        wi=wi*wpr+wtemp*wpi+wi;
      }
      mmax=istep;
    }
  }

#undef SWAP
```

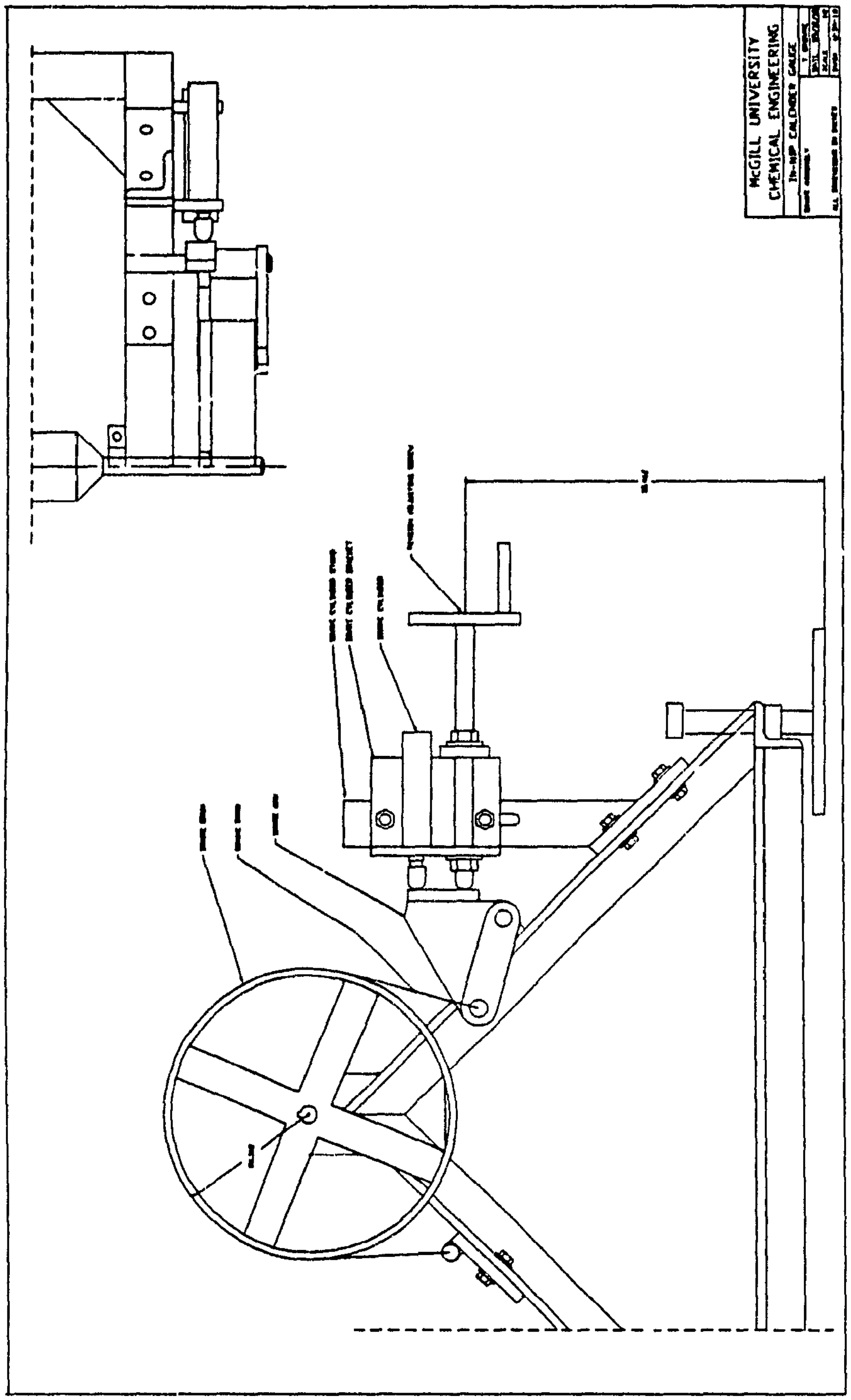
# APPENDIX 2: ASSEMBLY DRAWINGS



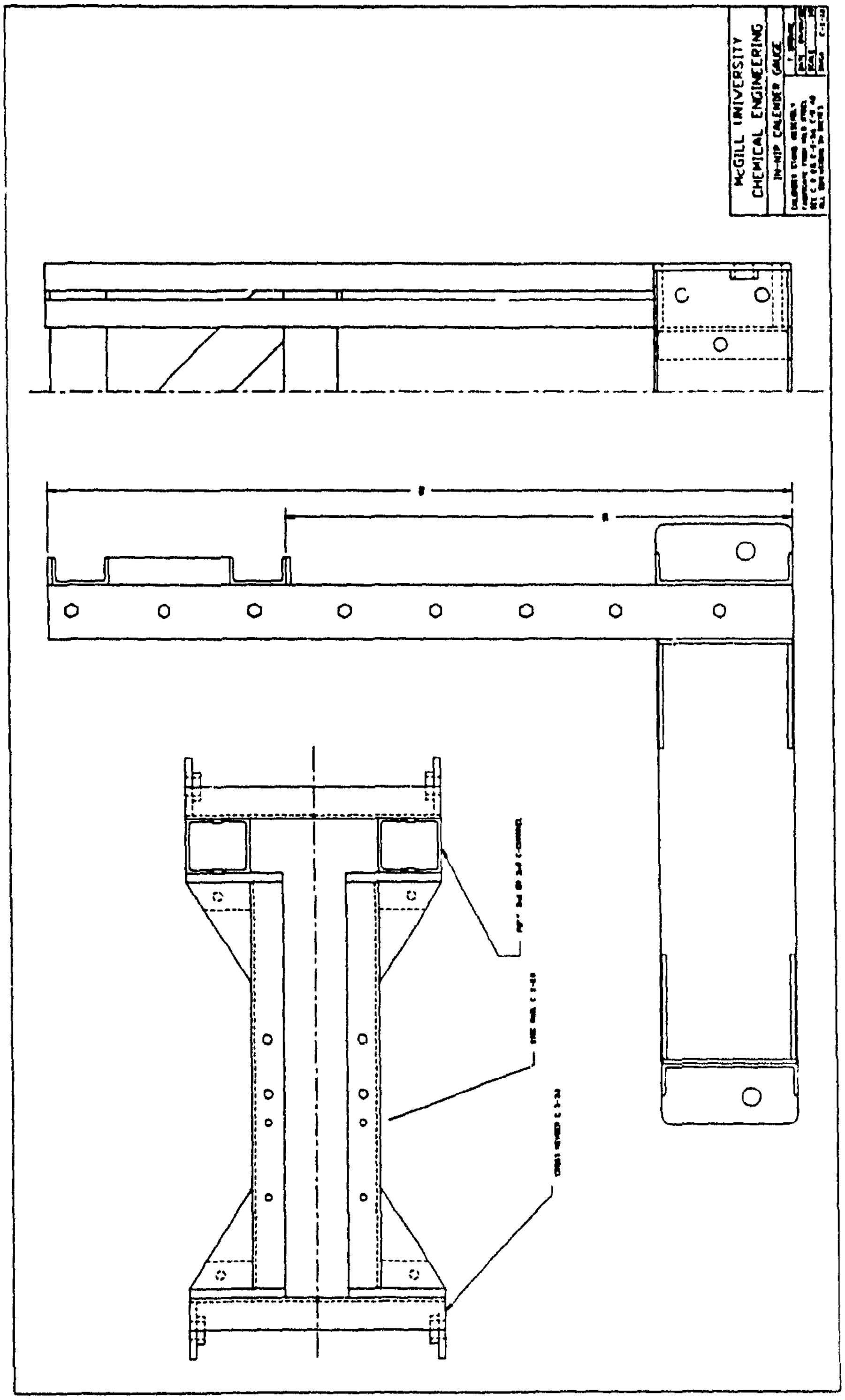

Overall view. Dimensions in inches; not to scale.



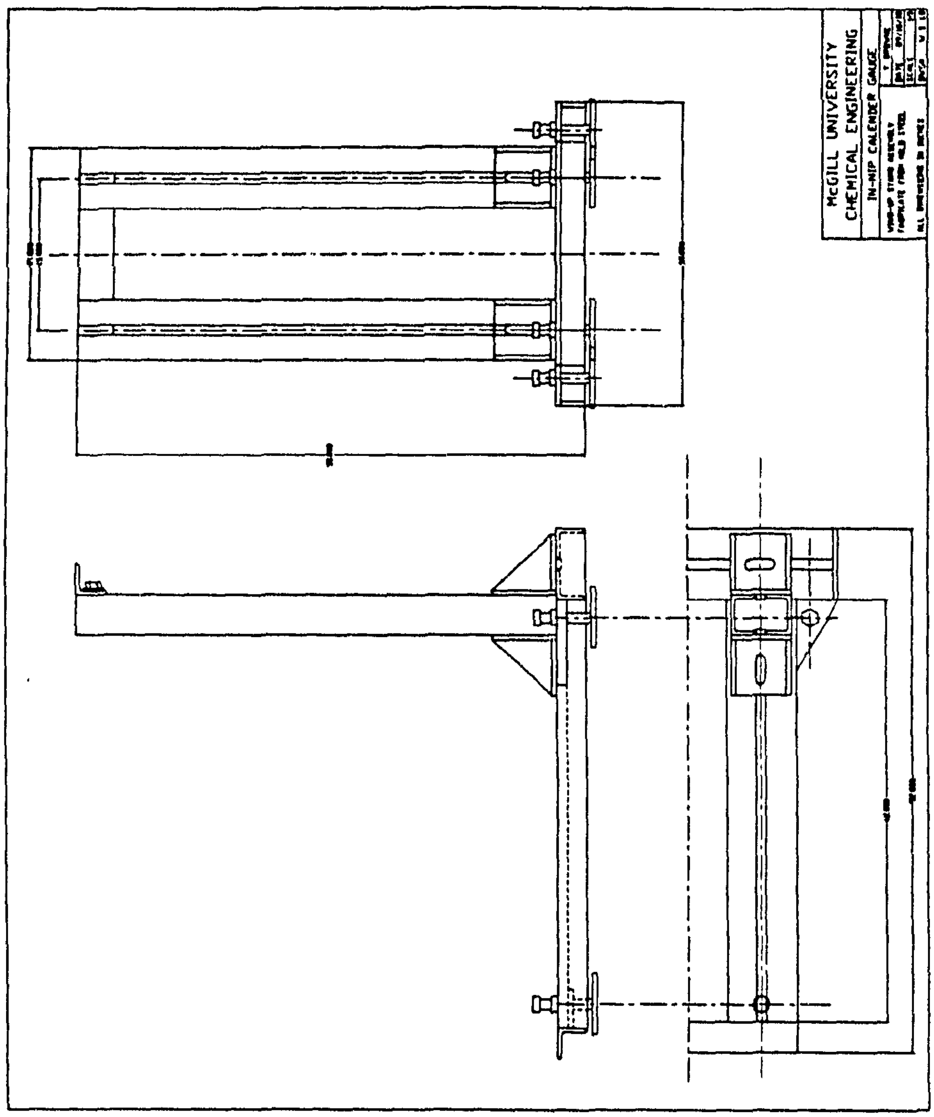

Unwind stand details. Dimensions in inches; not to scale.



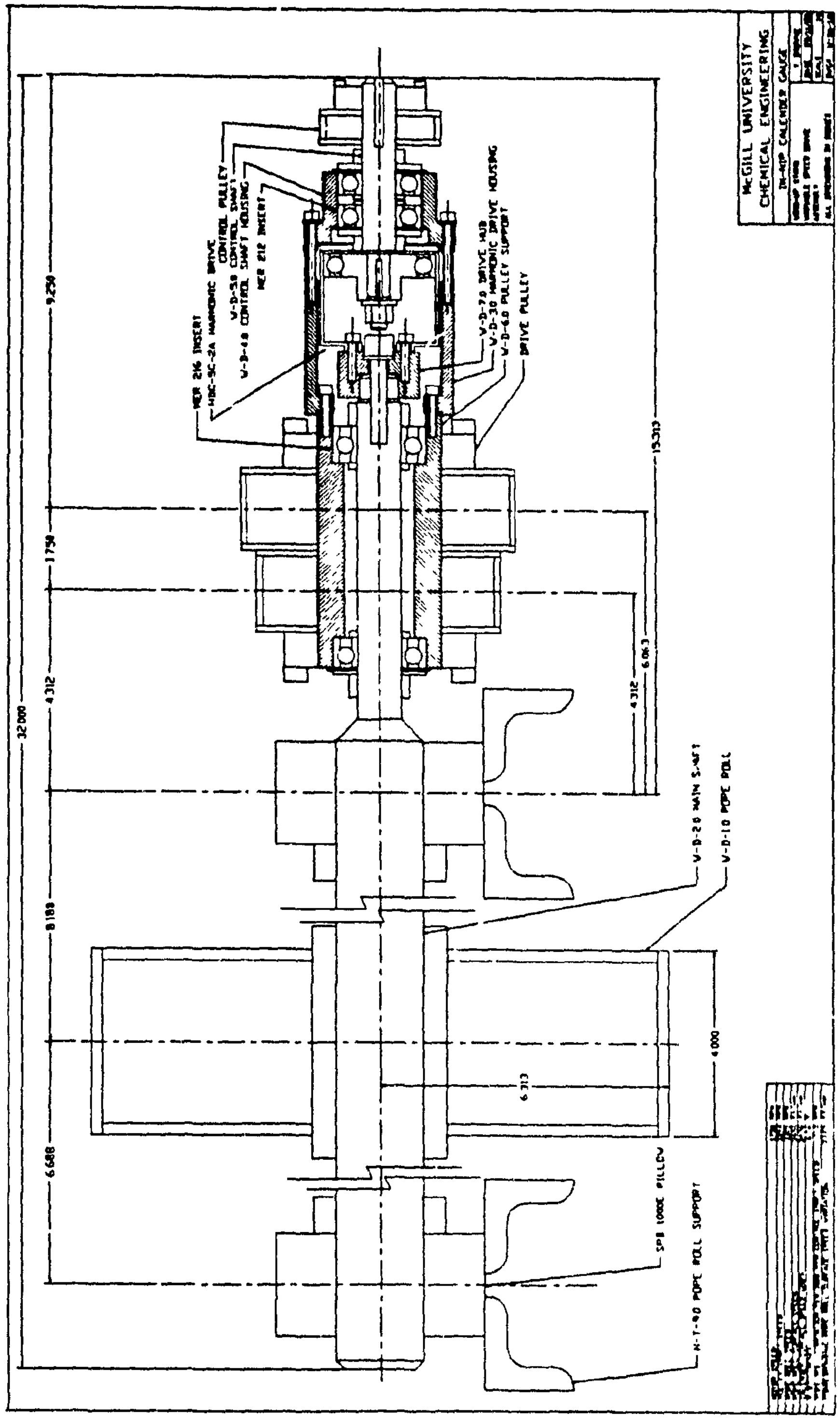

Unwind brake assembly. Dimensions in inches; not to scale.



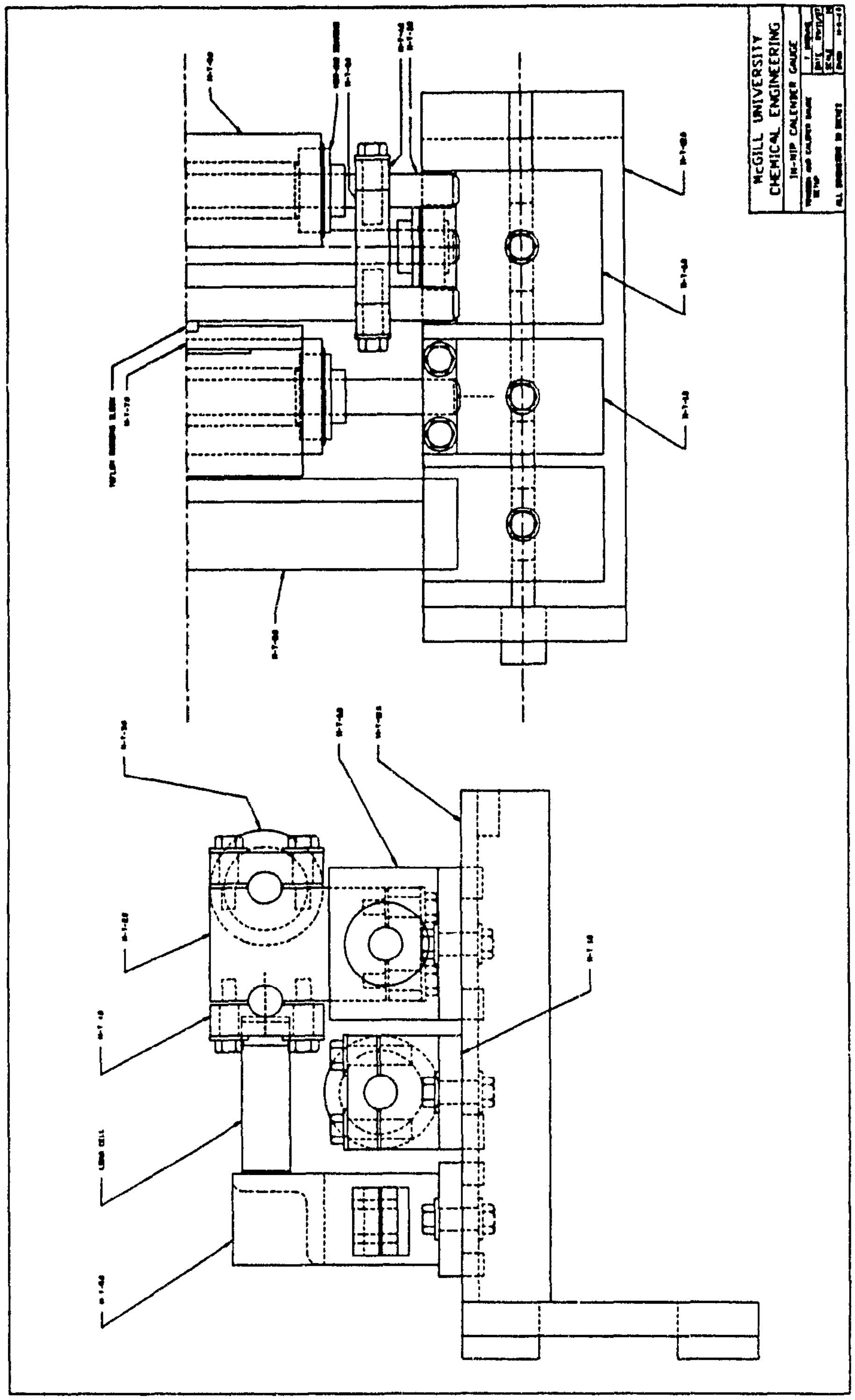

Main stand details. Dimensions in inches; not to scale.



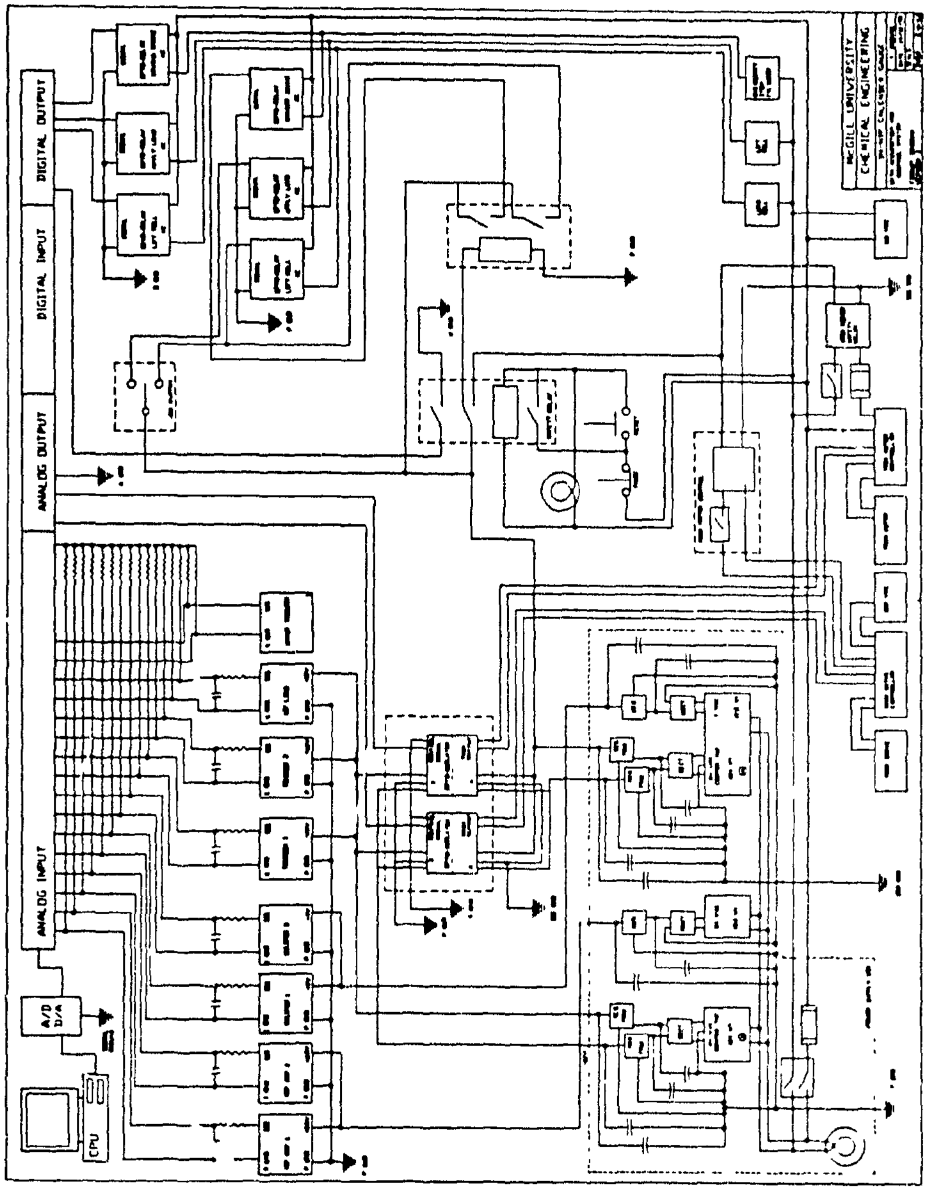

Windup stand details. Dimensions in inches; not to scale.




Variable speed assembly. Dimensions in inches; not to scale.




Tension and caliper gauge assemblies. Dimensions in inches; not to scale.




Circuit diagram, data acquisition and control system.

# APPENDIX 3: EXPERIMENTAL RESULTS

| Expt # | Vel m/min | Load kN/m | Thicknesses Initial μm | In-nip μm | Final μm | $\varepsilon_p$ | $\varepsilon_n$ |
|---|---|---|---|---|---|---|---|
| 14 a | 50.74 | 35.9 | 103.60 | 58.28 | 91.87 | 0.1132 | 0.437 |
| 14 b | 98.84 | 35.9 | 103.60 | 58.62 | 92.50 | 0.1071 | 0.434 |
| 16 a | 53.41 | 38.3 | 103.00 | 55.61 | 91.11 | 0.1154 | 0.460 |
| 18 a | 57.80 | 38.7 | 100.46 | 58.43 | 91.49 | 0.0893 | 0.418 |
| 18 b | 115.31 | 38.7 | 100.46 | 58.13 | 90.82 | 0.0959 | 0.421 |
| 18 c | 166.80 | 38.9 | 100.46 | 58.25 | 91.67 | 0.0875 | 0.420 |
| 20 b | 102.84 | 36.1 | 102.66 | 54.40 | 94.31 | 0.0813 | 0.470 |
| 20 c | 177.91 | 36.3 | 102.66 | 58.21 | 95.10 | 0.0736 | 0.433 |
| 21 a | 51.60 | 38.7 | 102.66 | 53.01 | 92.33 | 0.1006 | 0.484 |
| 24 a | 93.67 | 40.9 | 101.96 | 55.70 | 91.87 | 0.0990 | 0.454 |
| 24 b | 164.55 | 40.9 | 101.96 | 55.95 | 93.83 | 0.0797 | 0.451 |
| 27 a | 93.96 | 39.4 | 100.45 | 55.80 | 92.23 | 0.0818 | 0.444 |
| 27 b | 169.12 | 39.5 | 100.45 | 56.70 | 93.82 | 0.0660 | 0.436 |
| 28 a | 292.71 | 39.9 | 100.45 | 55.70 | 94.96 | 0.0546 | 0.445 |
| 29 a | 47.2 | 34.0 | 100.45 | 61.25 | 94.85 | 0.0557 | 0.390 |
| 29 b | 31.28 | 34.8 | 100.45 | 60.05 | 93.30 | 0.0712 | 0.402 |
| 29 c | 22.38 | 34.9 | 100.45 | 60.28 | 93.60 | 0.0682 | 0.400 |
| 29 d | 10.57 | 35.0 | 100.45 | 58.85 | 92.76 | 0.0765 | 0.414 |
| 29 e | 6.50 | 36.3 | 100.45 | 61.79 | 89.96 | 0.1044 | 0.385 |
| 37 a | 45.44 | 38.[illegible] | 101.15 | 60.19 | 93.43 | 0.0763 | 0.405 |
| 37 b | 102.84 | 38.7 | 101.15 | 60.56 | 95.00 | 0.0608 | 0.401 |
| 37 c | 164.55 | 38.9 | 101.15 | 61.13 | 94.36 | 0.0671 | 0.396 |
| 37 d | 284.13 | 39.0 | 101.15 | 62.73 | 95.57 | 0.0552 | 0.380 |
| 38 a | 52.49 | 42.2 | 101.15 | 57.12 | 90.74 | 0.1029 | 0.435 |
| 39 a | 53.41 | 42.5 | 101.15 | 57.34 | 91.63 | 0.0941 | 0.433 |
| 39 b | 27.18 | 42.7 | 101.15 | 58.75 | 91.09 | 0.0995 | 0.419 |
| 40 a | 50.74 | 42.5 | 101.15 | 56.98 | 92.06 | 0.0899 | 0.437 |
| 41 a | 89.67 | 42.2 | 99.61 | 51.69 | 94.03 | 0.0560 | 0.481 |
| 41 b | 275.31 | 42.5 | 99.61 | 51.65 | 95.82 | 0.0380 | 0.481 |
| 43 a | 87.31 | 40.4 | 99.61 | 51.41 | 93.10 | 0.0654 | 0.484 |
| 43 b | 158.26 | 40.8 | 99.61 | 51.61 | 94.81 | 0.0482 | 0.482 |
| 43 c | 277.24 | 41.0 | 99.61 | 51.55 | 95.67 | 0.0396 | 0.482 |
| 47 a | 52.39 | 35.6 | 100.74 | 60.69 | 96.32 | 0.0438 | 0.398 |
| 48 a | 55.29 | 147.8 | 100.74 | 22.57 | 85.40 | 0.1522 | 0.776 |
| 50 a | 55.21 | 155.1 | 100.74 | 21.83 | 84.45 | 0.1616 | 0.783 |

| Expt # | Vel m/min | Load kN/m | Thicknesses Initial µm | In-nip µm | Final µm | $\varepsilon_p$ | $\varepsilon_n$ |
|---|---|---|---|---|---|---|---|
| 54 a | 534.96 | 37.6 | 101.00 | 61.54 | 98.76 | 0.0222 | 0.391 |
| 56 a | 101.02 | 146.5 | 101.00 | 22.12 | 86.24 | 0.1462 | 0.781 |
| 58 a | 289.94 | 38.2 | 100.27 | 60.04 | 96.58 | 0.0368 | 0.401 |
| 58 b | 283.45 | 146.5 | 100.27 | 22.68 | 87.25 | 0.1299 | 0.774 |
| 59 a | 159.79 | 37.5 | 100.27 | 59.76 | 96.75 | 0.0351 | 0.404 |
| 60 a | 162.32 | 140.9 | 100.27 | 23.44 | 86.13 | 0.1411 | 0.766 |
| 65 a | 172.64 | 37.8 | 100.50 | 66.79 | 97.04 | 0.0344 | 0.335 |
| 65 b | 167.32 | 161.3 | 100.50 | 27.30 | 89.13 | 0.1132 | 0.728 |
| 65 c | 167.32 | 107.5 | 100.50 | 36.55 | 85.00 | 0.1542 | 0.636 |
| 67 a | 306.31 | 40.0 | 98.83 | 63.84 | 97.56 | 0.0129 | 0.354 |
| 67 b | 303.85 | 80.2 | 98.83 | 41.80 | 92.00 | 0.0691 | 0.577 |
| 67 c | 301.44 | 137.3 | 98.83 | 32.24 | 87.38 | 0.1159 | 0.674 |
| 67 d | 299.07 | 207.3 | 98.83 | 24.41 | 82.88 | 0.1615 | 0.753 |
| 67 e | 296.73 | 250.3 | 98.83 | 20.66 | 79.29 | 0.1977 | 0.791 |
| 68 a | 41.97 | 38.8 | 98.83 | 64.75 | 95.50 | 0.0337 | 0.345 |