HARDNESS DISCRIMINATION ABILITY OF THE KSU INDIVIDUAL WHEAT HARDNESS TESTER AS AFFECTED BY BLADE PENETRATION AND PLATE VELOCITY

by

PAUL JAMES BARRY

B.A., Benedictine College, 1984

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

College of Engineering
Department of Agricultural Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1989

Approved by:

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Increased cross-breeding between hard and soft wheat varieties over the past few years has challenged the integrity of the current wheat classification method. Cross-breeding has produced hard wheats that look like soft wheats on the exterior, yet they mill and bake like hard wheats. A similiar situation exists for soft wheats.

A Single Kernel Wheat Hardness Tester (SKWHT) has been developed as described by Eckhoff et al. (10), which measures the maximum force required to slice each individual kernel. This maximum force value is then used as a means to specify if the kernel is hard or soft. The SKWHT is rapid (approximately 200 kernels per minute) and is easy to operate. Initial test results showed that less than 1.5% of field samples tested were misclassified using this instrument.

The purpose of this study was to evaluate how the adjustable operational settings on the instrument of blade clearance, blade type (sharp, curved, or blunt), and plate velocity affect the ability of the instrument to delineate hard from soft wheat kernels.

OBJECTIVE

To study the effect of angular velocity, blade clearance, and blade type (sharp, curve, or blunt) on the ability of the Single Kernel Wheat Hardness Tester (SKWHT) to discriminate between 5H and Crayon Pencil lead, and to apply these results to hard and soft wheat.

LITERATURE REVIEW

One of the earlier studies on the hardness of wheat was done by Biffen (7) in 1908. In this study, there were three methods of determining the hardness of wheat. The first method was by visual means, which categorized strong wheat kernels as semi-translucent, whereas weak wheat kernels were more opaque and had a starchy appearance. The second method was to crush the kernels under an iron plate, with hard wheat kernels crushing into angular fragments or a gritty powder, and weak wheat kernels turned to fine powder. The final method was to actually chew approximately 20 to 30 grains until the starch and the grain coats have disappeared. This last method classified strong wheat kernels by possessing more gluten which could be stretched into long threads and would retain it's shape when pressed flat. Weak kernels leave small quantities of soft and slightly viscid gluten in the mouth, whereas strong kernels leave considerably more gluten in the mouth, and when it is rolled into a ball and pressed flat, it will return to its original shape.

Biffen studied cross-breeding of different varieties of grain. The grain was harvested and then samples were pressed with an flat plate iron to identify the sample. Usually the samples were identified via visual inspection. However, some of the samples were very hard to distinguish and these samples were subjected to the chewing method to further classify the sample and the parents of the cross-bred kernel. Two Crosses were performed, Rough Chaff (Weak) with Red Fife (Strong), and Red Lammas (Strong) and Red Fife (Strong). In the first of the two crosses, the property of strength is dominant over the lack of strength. It is also noted that the heterozygotes are indistinguishable from the dominant homozygotes. In the second cross, dominance is not clear, and the heterozygous individuals are easily distinguishable with a given degree of accuracy. The cross-bred kernels were also tested in a baking lab to determine the relative hardness of each group. The analysis from the grinding and baking concluded the same as the chewing and visual methods, which was that the F.2 generation's relative strength and lack of strength were self-evident. It was also noted that the primitive methods worked out more reliably than had been anticipated by the investigator. The baking methods led to the conclusion that in all of the crosses, the strength of the cross had been inherited in its entirety. It was also determined in the study that high-yielding capacity and strength could be obtained in combination in the same variety, but a natural separation between high- and low-yielding capacity at an F.2 generation is questionable. Further investigation is required to determine this question. The total nitrogen content of kernels was studied as a relative index to hardness, but no conclusive evidence was presented to support this hypothesis.

Roberts (30), in 1910, studied the hardness of wheat with a device known as the "grain crusher." The kernels to be tested were first dried for a period of seven days, and then the kernels were each laid down on the table, and constant addition of weights were added to the level mechanism until the strain on the kernel was too great, and the crushing-point of the kernel was reached. Each of the kernels tested were laid with the crease down on the crushing table. The "soft" wheats crushed at about 6000 grams or less (13 pounds), "semi-hard" wheats at about 9000 grams (20 pounds), and "hard" wheats at 12,000 grams and over (26 pounds and over). Roberts determined the number of kernels to crush in order to arrive at an approximately correct average estimate of the hardness was 350 kernels. The 350 kernels were found by modeling the crushing forces, and then taking the first and second derivatives of the model force, and determining the point at which the slopes of the derivatives do not change as rapidly, and this point was the number chosen.

Newton et al. (25) used six different varieties grown at six different locations and studied the effects of kernel texture, protein content, and hardness by measuring the strain required to crack the kernels transversely. A machine which was designed by the Field Husbandry Department of the Ontario Agricultural College was used. The machine contained an ordinary pair of pincers mounted vertically with jaws at the top. One arm was rigidly fixed to a standard, while the free arm was attached by a cord to a spring balance lying in a horizontal plane. Another cord was connected on the opposite edge of a hand-operated windlass, the turning of which was transmitted through the pincers with the tension being indicated on the balance. This hardness machine was modified in the following ways: 1) a vernier was added to the spring balance to prevent the spring balance to not be displaced once the kernel was cracked; 2) a free jaw of the pincers had a long pointer which indicated the diameter of the kernels on a scale graduated to fifths of millimeters.

In order to determine the number of samples which would yield a representative sample, an equation was derived for the percentage error for each variety which was:

$$E=bn^a$$

where

E is the percent error,
b is a constant,
n is the number of kernels cracked, and
a is the constant exponent which gives the change in shape.

After evaluating this expression with data collected every 100 kernels, up to 700 kernels, it was determined that the slope decrease beyond 200 kernels was very small and that 250 kernels should be used as the sample size.

One of the relationships found in this study was that cracking strain increased with the size of the kernel, and a compensating factor needed to be introduced if different samples wanted to be compared. The strain was correlated with kernel diameter to see if dividing the strain by the diameter was a valid way to compensate strain for different diameter kernels. A hardness value "a" was determined in this fashion, and there seemed to be no relationship between it and kernel size, and thus this could be used to compare different samples.

A relationship between cracking strain and a function of the diameter approximating the cross section of the kernel was attempted, but no significant relationship was found. The strain was also compared to protein content and there was not a significant relationship (r=-0.23). The negative relationship pointed out that the yield and kernel size would increase at the expense of protein content. The two variables "a" and "b" were compared to protein content but no significant relationship was found (r=-0.12 and r=0.16, respectively). Vitreous kernels were found to be higher in protein than the starch kernels, except in one case out of 100 kernels per class. It was concluded by the study that the relationship between hardness and protein content existed only within each sample, and the same was true for vitreousness and protein content. The protein content in relationship to hardness was said to be "too complicated to give promise of much practical utility in wheat grading."

Two varieties of wheat, which were very different in protein content, Marquis and Standup, were tested for moisture effect on hardness with 18.2 and 10.2 percent protein, respectively. These samples were subjected to six different moisture contents from two to 14 percent. It was shown that the moisture content from two to 14 percent had little affect on hardness. The hardness factor "a" was found to decrease when the moisture content was above 11 percent. It was advised that in the future, hardness should be determined in the 2-11 percent moisture range.

Taylor (37) employed a barley pearler to distinguish relative hardness of wheat in 1939. In the study, the percentage of wheat pearled off was correlated to other hardness measurements such as the Particle Size Index (PSI), and the dough ball time. The investigation used the following procedure to test the pearling method: 1) approximately 100 grams of the variety to be tested was placed on a No. 6 Tyler screen over a No. 8 Tyler screen; 2) the sample was shaken a predetermined number of times and the grain above the No. 8 Tyler screen was subdivided into three 20 gram samples; 3) each sample was subjected to the pearler for three minutes; and 4) then the sample was screened on a No. 20 screen and the grain remaining on the screen was calculated as a percentage of the 20 gram sample. All of the wheat samples tested were allowed to equilibrate in a seed storage room for two months at 10-11% moisture. High correlation coefficients were noted between the percentage of the kernels pearled off with the particle size index test. The correlation coefficients were of smaller magnitudes than the particle size percentage of wheat pearled off. There was little correlation between: the percentage pearled off, PSI, doughball time, and protein content of the grain studied. Twenty-seven varieties were studied at the five different stations. The five different stations were located in Lincoln, NE; Urbana, IL; Ithaca, NY; Kearneysville, W. Va; and Arlington, Va. The correlation coefficients for pearling against other locations were almost all above 0.9 except for Ithaca vs. Lincoln which had r=0.863, which indicated that the relative hardness of a particular variety as measured by the pearling test was much the same, no matter where the variety was grown. Comparing the varieties of typical spring and hard winter varieties (Marquis and Kharkof) to the very soft common white and club wheats (Irwin Dicklocs and Albit), it was noted that only a slight relationship, if any, existed between the percentage pearled off and the protein content, or between the particle size index and the protein content, for either winter or spring varieties. This investigation also showed high correlation coefficients (r=.857,r=.835) between the percentage of the kernels pearled off and the particle size index. Also, slightly lower correlation coefficients (r=-.769, and r=-.755) were obtained between the percentage of kernels pearled off and the doughball time. It should be noted that certain varieties responded differently under the two tests (doughball time and PSI).

In 1943, McCluggage (20) studied the hardness of a sample of wheat by modifying the Strong-Scott barley pearler. The barley pearler was modified by adding a variable pitch pulley to provide various speeds at the grinding stone. Also, the barley pearler was equipped with a No. 30 grit stone. The procedure involved the following steps: 1) each "charge", 20g of wheat, was weighed from a cleaned wheat sample which had been thoroughly mixed, but not sized; 2) the charge was released after the stone had been running for 60 seconds; 3) the charge remained in the pearler for ten seconds, and then the motor was turned off; 4) the pearled wheat was then sifted over a 20-wire screen; and 5) finally the amount of material above the 20-wire screen was weighed and recorded. The following effects on the results of the pearling were noted: 1) the effect of temperature on the wheat and pearler; 2) the effects of stone velocity and the timer period of the pearling; 3) the effect of the weight of the sample; 4) the effect of the screen; 5) the effect of moisture; and 6) the different test sites. The study derived the following conclusions: 1) the pearling test was not sensitive to a wide range of temperatures, and therefore temperate had little effect on the pearling results; 2) the effect of sifting the grain prior to pearling it slightly increased the mean for one variety and reduced the mean for the other variety. Experience had led to the observation that sifting yielded a greater accuracy in the weighing; 3) the accuracy of the experiment had the same range of error for the three different operating velocities, and three time periods. Thus, the standard velocity of 1725 rpm and a pearling time of one minute was recommended; 4) the amount of material difference between varieties for different charges remained fairly constant and therefore a given charge could be chosen throughout the experiment; 5) the effect of the screen was studied and it was found that when a metal plate replaced the screen, the variability of the pearled wheat varied significantly and that when holes were drilled in the plate, the results were more closely correlated; 6) the screen was determined to provide the grinding action of the pearler. It was concluded that the screen did most of the grinding and it should be replaced often (10-mesh screen with 0.041 inch in diameter); and 7) the effect of moisture was analyzed and the correlation coefficient between the percent moisture of the wheat and the percentage of pearled off was +0.029. The data revealed variations in the percent moisture of the Hard Red Winter Wheat within the limits of the study had little or no influence on the percentage pearled off.

The study of the pearling index on crosses between hard and soft wheats was performed by Beard and Poehlman (6) in 1954. In addition to the pearling index study, the validity of visual inspection for hardness was evaluated for successive generations of crosses. Two hard wheat parent varieties (Kawvale and Pawnee) were crossed with five soft parent varieties (Trumboll-Wabash-Hope-Hossar, Fultz-P.I. 94587-Fultz-Hungarian, Trubull-W38-Fultz-Hungarian, Mediterranean Selection (w 5638) and Mediterranean Selection (w 5652). The head selections were taken from random and planted in one foot rows in 1950. The rows with normal stands were harvested and were allowed to reach equilibrium before conducting the pearling test. The pearling tests were conducted with a Strong-Scott barley pearler with a ten by ten mesh bronze wire tyler screen of 0.041-inch diameter wires. ten grams of wheat was pearled for two minutes with a grinding wheel speed of 1435 r.p.m.

Selections of bulk hybrids Kawvale and Pawnee crosses were classified by visual inspection into hard, medium or soft texture classes. Heads where all the kernels appeared light in color, plump, opaque and starchy were classified as soft; heads in which all the kernels were dark, hard and vitreous were classified as hard; and finally samples with a mixture of hard and soft kernels, or with mottled kernels, or ones which could not be classified in appearance were classified as medium.

After evaluating the two samples by visual methods, they were tested by the Strong-Scott barley pearler, proceeding harvest in 1951. The kernels classified as soft in 1950, only 35% had a pearling index of 35% or above in 1951, whereas 27.39% of those classified as medium and 16.7 classified as hard pearled above 35% in 1951. A random sample of these two varieties would have 24.1% of the sample pearled above 35%, and therefore the visual selection did not classify the strains in the same manner as the pearling test.

Beard and Poehlman (6) then proceeded to evaluate the pearling test on seven crosses and samples from four of the parent varieties grown in 1950 and the progenies grown in 1951. It was found that the correlation coefficients were highly significant of the pearling-indexes. The average R-value for the seven families was 0.841 and $R^2 = 0.707$ indicating 70% of the variation in hardness of the second year crop was associated with variation in hardness of the first year crop, and 30% was independent of the hardness of the previous year's crop.

Results from this study showed that segregates from hard X soft crosses might be expected to vary widely in kernel hardness as measured by the pearling test. Distribution of the segregates indicated that it was probably a multigeneic character for hardness.

In 1959, Katz et al. (13) studied the hardness of grain by adapting the Barcol Impressor which was a commercial soft metal tester. The adapted tester used small sections of wheat kernels which were then mounted to glass microscope slides with Duco cement. Sections of wheat kernels were sliced transverse to the crease in the kernel in order to ease testing procedures. The actual tester consisted of a spring-loaded stylus, a case, and a dial micrometer. The hardness measurements were made by moving the framework of the Impressor down by hand until the stylus came in contact with the specimen. The dial reading on the micrometer would achieve a maximum reading as long as pressure was applied to the stylus. The following numbers are unitless relative hardness values, based on the distance of penetration in a given sample. Measurements taken from Ponca wheat were 39.7 ±2.5 in the central region of the kernel, 34.9 ±0.4 near the crease, and 41.6 ±1.7 near the bran. Similar results were found for Mindum wheat. In testing durum wheat an average reading of 38.9 ±2.2 in the center, 36.8 ±2.1 near the crease and 40.0 ±2.0 near the bran. In this investigation, variations of ten hardness numbers across one Ponca section were not uncommon, while the hardness of a Mindum section seldom varied more than four or five hardness numbers. Results of the experiment revealed that the periphery of the kernel appeared harder than the region around the crease. The effects of the experimental technique of measuring hardness needed more investigation, such as 1) the technique of the specimen preparation (the influence of freezing, thawing, and cementing on hardness measurements); 2) the influence of ambient humidity; and 3) the varietal and agronomic conditions on wheat hardness.

Another one of the earlier studies on wheat hardness was by Katz et al. (14), where the major objective was to study the effect of moisture content on the relative hardness of the wheat kernel. The hardness was measured by a special device created by Katz et al. (13), and it was developed from a commercial hardness tester called the Barcol Impressor. Transverse sections of wheat kernels, approximately 1mm thick, were taken from the central portion of the kernel to be tested, and then they were cemented to glass microscope slides with Duco cement. The samples were then viewed under the microscope in order to detect either fractures or mold growth. Once the samples had equilibrated to a given moisture content, then the glass slide was placed on the micrometer stage of the hardness tester and the framework was pressed down until the flat part of the tester spindle was in contact with the specimen. At this time, the dial reading would achieve a stabilized constant maximum value. Katz concluded that "hardness of hard wheat varieties (hard red winter and durum) diminished with increasing moisture content" and that soft white wheat showed no significant effect until the moisture content was above 13%, and then the hardness decreased rapidly.

Symes (36) in 1965 studied hardness utilizing the particle size index method and related this to the inheritance of grain hardness. For this study a 10g sample of wheat was ground in a LabConco mill set to grind as finely as possible. A gravity feed was added to the mill which consisted of a funnel five in. long by 7/8 in. diameter pipe to guarantee a uniform rate of grinding. The meal was then sieved through a 200 mesh brass cloth (with an opening of 74 microns), in half height, eight in. diameter Tyler sieves, each with its own cover and bottom pan. Six units were placed on a Ro-Tap sieve shaker for ten min., with whole wheat kernels being placed on the sieve to prevent clogging of the sieve. The material which passed through the sieve was weighed to the nearest 0.01g and expressed as the percentage of the total meal which was known as the particle size index (PSI).

The gene or genes which determine PSI hardness were investigated by a method known as backcrossing. The method of backcrossing involved cross-breeding one parent cultivar with crosses between two cultivars. The new cross-breed was then crossed with the parent donor and thus losing the genes of the other cultivar which was not cross-bred. The two major wheats studied were Heron (a soft wheat), and Falcon (a hard wheat). The investigation's main goal was to determine whether or not a single gene was responsible for hardness of a particular wheat. In seven other crosses between hard and soft wheats, there also appeared to be one gene responsible for hardness. The transference of a singe gene which would convert a soft wheat to a hard wheat, and vice versa. Although the test did not yield an exact estimate of how a parental type was recovered, the range however, obtained from the parents was consistently narrower than that obtained from the corresponding homozygous class. The crosses studied, and in particular, the PSI values for the hard groups of the F_2 (second generation) in the crosses Spica X Heron and Spica X Bordon were as follows: the mean value for Spica was 17.4%, and for the hard groups of the two crosses 17.3% and 16.7%, respectively, and the absence of any values below 14.7 but above this led to the thought. If the hardness of Spica was controlled by the same major genes as Gabo and Falcon modified by a lot of other genes, then a value should be in the range obtained for Gabo and Falcon crosses. Since the data did not show this, a single gene separated the hardness of the soft and hard wheat. The data has shown a viable method of determining hardness by the PSI test and by using backcrossing, it would be possible to convert a hard wheat to a soft wheat and vice versa.

Anderson et al. (1), studied 34 different varieties in six different classes which were durum, HRS (hard red spring), HRW (hard red winter), SRW (soft red winter), SWW (soft white winter), and WC (white club). The samples were tested on a Brabender Hardness Tester and a Pin Mill to achieve hardness on friability indices. A sample size of 100g was sent through the tester with the preset grinding index. The resultant ground product was then sifted over 18, 30, 50 and 100 USS screens. The particle size distribution of the flour fraction (that remained above the 100 USS screen) was determined with a Micromerograph air sedimentation apparatus. Wheat tested on the Pin Mill was first tempered to the correct moisture content overnight and 125g was introduced into a Model 160Z Alpine Pin Mill operating at a rotor speed of 9,000 r.p.m. The product collected in a small bag was screened and the particle-size distribution was determined.

In determining the viability of the Brabender hardness tester, the optimum operating conditions were sought by comparing results between a HRW wheat (Rio variety) and a soft WC wheat (Omar variety). The tester was operated at three different index settings and two moisture contents. The analysis showed the index setting of 1.0 and the moisture content of 15.0% yielded the best separation between the two wheats. The flour fraction surface given per unit work was the most discriminating parameter with a four fold difference between Omar WC and Rio HRW at 15% m.c.

In addition to the optimum settings for the Brabender hardness tester, the setting on the Pin Mill also required evaluation to obtain maximum separation between hard and soft wheat. Preliminary tests revealed that a rotor speed of 9,000 r.p.m. would maximize the difference between hard and soft wheat. At a rotor speed of 9,000 r.p.m.; 1) the flour produced was approximately the same particle size as the Brabender; 2) it was the slowest standard stock speed for a stock machine; 3) flour yields differed substantially between hard and soft wheat. The most discriminating variable was the surface area of the flour fraction with approximately a four fold variation between Rio HRW and Omar WC.

The test proceeded with all 34 varieties tested at 15% m.c. and with the settings mentioned earlier for the Brabender Hardness Tester and Pin Mill. The data showed that the flour yield or flour fraction surface area per unit work by the Brabender tester or Pin Mill could be used to rate wheats according to kernel hardness and friability, even though the flour yields from the Pin Mill were three times that of the Brabender. The total surface area in grinding was not used due to lack of sensitivity in the test. The work expended in grinding also proved to be too insensitive for a kernel hardness index. This test indicated that wheat could be classified by hardness with a Brabender or Pin Mill with the measurement of the flour yield or of the flour fraction surface area. The sensitivity of the test is increased by utilizing the flour fraction surface area and yields the most sensitivity when the flour fraction surface area per unit of work is used.

Williams (38) measured the particle size index (PSI) as a means of measuring kernel hardness and relating the values to chemical methods. The kernel texture was determined by grinding in a LabConco mill set at its finest setting, and the product from the mill was sieved for 10 minutes on a 200-mesh wire sieve. The percentage of throughs was recorded as the PSI. The PSI provided a consistent measure of the relative kernel hardness for a wide range of wheat varieties grown in Australia. It should be noted that all of the samples tested were in the moisture range of 9.3 to 10.3%. A multiple regression was carried out relating PSI to damaged-starch content, and incorporating protein content as a second variable. The r-squared value raised slightly and the conclusion was that protein content had relatively little influence on the relationship between PSI and damaged-starch content. Another implication from their results was the fact that not only hard wheats yielded flour which contained a higher proportion of damaged starch, but that the starch itself was more susceptible to diastatic attack even in the undamaged state.

The Brabender Hardness Tester (BHT) was again used by Greenway (11) in 1966. The procedure involved determining the protein content on 63 hard red winter (HRW), 16 hard red spring (HRS) and 22 soft red winter (SRW) and soft winter (SW) wheat samples with all possessing approximately the same moisture. All of the samples were then ground on the BHT and graphs were recorded for each sample with the peak value tagged as the "wheat hardness peak." The meal from the sample was sieved on a U.S. No. 100 woven-wire cloth for 15 min. via Ro-tap shaker, and then the percentage of flour was determined.

In addition to the flour, the flour particle diameter and total flour surface area were determined for each sample. The wheat hardness index for each sample was determined as well. The wheat hardness index was found by dividing the wheat-hardness peak on the BHT by the percent flour yield. The bran was excluded from the wheat hardness index due to the complexity involved in calculating total surface area. As part of the wheat hardness index test, 5 portions each of 3 HRW and 5 portions of a SW wheat were tempered to 5 different moisture contents (8, 10, 12, 14, and 16% m.c.). The following correlations were noticed: 1) the WHI was inversely related to moisture content and conversely it was directly related to dry sample weight; 2) in most cases, flour yields increased with moisture content whereas particle diameter decreased; 3) the WHI was directly proportional to protein content; 4) the WHI was more sensitive to wheat hardness than the hardness peak.

It was concluded that flour yield, total flour surface area, protein and moisture contents were important in the determination of hardness. Some of the more difficult quantifying contributors to hardness were: 1) complex physical interactions between protein, starch, minerals, and moisture within the endosperm matrix during maturation; and 2) the bran itself, which contributed to the hardness. Again, the hardness peak is a measure of the work required to grind 100g of wheat. When this value was divided by the percent flour, the quotient was a factor named the wheat hardness index. The wheat hardness index correlated highly with protein content per m^2 of flour.

In 1969, Symes (35) used the particle size index method to determine the hardness of near-isogeneic lines of Falcon (a hard wheat), and Heron (a soft wheat) and attempted to track a gene which influenced the hardness of a particular wheat kernel. although the degree of hardness was slightly influenced by at least two minor genes, it was possible to convert Falcon to Heron or vice versa through cross-breeding. Over a seven year period, cross-breeding was performed on Falcon and Heron to produce six different groups: Heron, Falcon/7*Heron soft, Falcon/7*Heron hard, Falcon, Heron/7*Falcon hard and Heron/7*Falcon soft. The "7*" referred to backcrossing the cultivar following the * with the cultivar (preceding) the "/" symbol. The crossing of cultivars had led to the hypothesis of a single gene causing hardness. The difference between hard and soft types was clearly visible and the variability was extremely low. The lack of minor modifying genes is relevant between Heron/7*Falcon soft with 26.9 PSI compared to Heron 28.7 PSI, and Falcon/7*Heron soft 28.6 PSI. The trend was present in Falcon/7*Heron hard (13.4 PSI), Falcon (12.7 PSI), and Heron/7*Falcon hard (12.9 PSI) but it was not significant.

The protein was lower in Heron than Falcon and any backcross of Heron as the recurrent parent. It was determined that there was a lack of correlation between protein content and hardness. The baking tests on a near-isogeneic lines, with groups only differing by a single gene which determined hardness as measured by the PSI method, showed that this gene had a great influence on the baking quality of flour milled with these wheats. Milling extraction from one year's testing also showed that it was strongly influenced by this gene. There was no evidence that the gene which influenced hardness was associated with protein content in the kernel.

The backcrossing of kernels lost the genes from the non-recurrent parent if linkage was ignored. Thirteen backcrosses were required before the gene contribution of the non-recurrent parent drops below 0.01%. However, only six backcrosses were required to drop this percentage to 0.78%. Material derived from six backcrosses were said to not contain isogeneic lines, but they did possess "near-isogeneic" lines. If the hardness lines of the same PSI as the recurrent parent did not differ significantly from the parent, then the isogeneic state had been approached sufficiently and any differences between these two and the backcross material with the PSI of the donor parent could be considered due to the action of a gene which determined PSI.

Chesterfield (9) measured the hardness of Australian wheats by means of a modified barley pearler in 1971. The barley pearler used was a Strong-Scott barley pearler Model 38 driven at 1440 rpm by a 1/4 H.P. electric motor. Modifications to the pearler were as follows: 1) the slotted silicon carbide wheel supplied with the machine was replaced by one with a smooth edge to prevent retention of grains in slots between determinations; 2) the wheel was made 1.5 in. wide by 6.25 in. in diameter with a tolerance of -0.00 in., to +0.030 in.; 3) the wheel was specially made from very hard grit and had a very strong bonding (coding 37 C 24 VVK) to reduce wear and prevent a subsequent increase in the gap; 4) fibre washers and epoxy resin putty to build up the casting were used to prevent grains lodging in the space formed by the shaft with the casting; 5) the hopper was built up so that the sample could be inserted with one hand, leaving the other free to start the stop watch; and 6) the solid-bottomed drawer was replaced by one with a No. 20 wire gauze bottom to allow the sample to be sieved without double handling. Also, the edges of the 10 mesh wire screen were brazed and ground to give a smooth movement of the slide opening.

Various sample sizes and grinding times were tried with a final sample size of 10g and a pearling time of one min. for the pearling tests. To test the reproducible results of the method, a sample of 20g was placed in the pearler and the standard deviation of the pearling resistance was determined with 0.315-0.093g for a soft wheat and from 0.094-0.051g for a hard wheat, which was sufficient for reproducible results. The effect of moisture was tested on samples of a very hard (Festiguary) and very soft wheat (Pinnacle) which were in the moisture ranges of 7-17%. It was found that the pearling resistances had a linear correlation with moisture with a coefficient of 0.272, and 0.119 pearling resistance units for each percent of moisture for very soft and very hard wheats, respectively. In the second stage of moisture effects, sub-samples of a wide variety of wheats were conditioned to two higher levels of moisture. The regression coefficient was highly correlated to initial pearling resistance. It was also determined that no regression coefficients were large, with ranges of -0.1541 (for hard wheat) to +0.2068 (for soft wheat), to make an major impact on pearling resistance.

Chesterfield also compared pearling resistance figures to particle size index values. Data by Symes (35) in 1963-64 was used to achieve a regression equation of Pearling Resistance = 7.39-0.13*PSI. Results on samples of wheat from the 1968-9 harvest yielded the following equation:

$$PR = 8.19 - 0.14*PSI$$

where
PR = Pearling Resistance, and
PSI = Particle Size Index

Although the samples taken by Symes and Chesterfield differ not only in year but location, the intercepts and slopes were not extremely different. The high correlation ($r=0.94$) between Pearling resistance and the PSI, indicated that the modified pearler provided as good a method of determining grain hardness as the PSI. Modifications of the barley pearler have yielded an accurate determination of the pearling resistance which could be accomplished in one step and was faster than the PSI test. If the moisture range was not very large, the moisture effect can be negligible in the pearling resistance value.

Stenvert (33) utilized: 1) the particle size index (PSI) test; 2) the pearling test; and 3) starch damage to check the hardness of wheat in 1972. Forty-six samples of flour were checked in all. Varieties ranging in hardness for all of the methods ranked the flours similarly. Frequency plots revealed bimodal distributions in all of the hardness tests except for Gamenya. Soft wheats were: typically stratified as possessing a particle size index greater than 22; a pearling resistance below 4.9; starch damage below 14% when milled (each sample was tempered to 16% moisture for 24hr. before milling at a feed rate of 100g per minute in a Buhler experimental mill); and a diastatic activity below 1.8% as the environment was stabilized. Hard wheats had: particle sizes below 20; pearling resistance above 5.1; starch damage over 16%;

and diastatic activity above 2.0%. There was a high correlation between all of the hardness measurement techniques. The correlation coefficients of the logarithmic relationship between the particle size index test and starch damage and diastatic activity were -.95 and -0.91, respectively. The coefficients between pearling resistance and starch damage and diastatic activity were 0.96 and 0.94 with all significant to the 0.1% level of probability. The tests also indicated that hard wheats produce higher yields of flour than the soft wheats.

Barlow and Buttrose et al. (3) in 1973, studied the nature of the starch-protein interface in wheat endosperm. There where several different tests conducted on the starch-protein interface, and for the hardness testing, purified starch and storage proteins were used. The samples were dispersed in a polyester-type resin (Astic), and polished according to Zeilder and Taylor (44).The specimens were then subjected to a micropenetrometer (Leitz Miniload hardness tester). Hardness values were measured in Vickers units and calculated from tables supplied with the testing instrument, or by the formula:

$$HV = \frac{1854*P}{d^2}$$

where

HV = Vickers hardness in kg per mm squared,
P = measuring force in pounds, and
d = length of the indentation diagonal in microns.

Values obtained from the micropenetrometer for the starch and the protein were very similar over a range of wheat varieties differing widely in particle size index values. Results from the hardness tests suggested that the individual storage components did not differ in hardness between varieties, but the adhesion between starch and the protein did differ.

Hoseney and Seib (12) used the Scanning Electron Microscope to view the native structures of wheat and its fractions to determine the difference between hard and soft wheat. The study pointed out the three possible theories to support the difference in breaking strength of hard and soft wheat. The theories were: 1) hardness is due to the variation in the ratio of protein to starch components; 2) the starch and protein components are intrinsically harder in hard wheats; and 3) the binding forces between the starch and the proteins differed between hard and soft wheat. The first theory was dismissed as a possible explanation due to the fact that a soft wheat variety grown under conditions designed to produce higher than normal protein content would still be relatively soft, and on the other hand, a hard wheat with a relatively low protein would still remain hard. As for the second theory, the inherent differences is also an inadequate description of the relative hardness of a kernel. Micropenetrometer tests conducted by Barlow and Simmonds (4) revealed no difference between granular wheat starch and protein matrix of isogeneic lines of hard and soft wheats in terms of hardness. Including these two authors, Wrigley (42) credited the difference in hardness to the variations in the adhesion between the starch and the protein components. Utilizing a fluorescent antibody technique, it was shown that hard wheats contained a layer of water-soluble protein around the starch granules whereas the soft wheats did not have this same layer. The procedure involved the slicing of wheat kernels with a razor blade, coating the kernel with a 150 Angstrom coating of gold-palladium alloy, and then viewing these fractions under the scanning electron microscope. Results of the research concluded that the hardness of wheat was determined by the strength of the protein-starch bond. Evidence to support this conclusion were the conditions of the starch granules after fracturing the kernels. In soft wheat varieties, the starch granules were more intact, than were the hard wheats. On the other hand, the hard wheat starch granules were fractured and not whole as found in soft wheat.

Simmonds and Barlow et al. (32) studied the biochemical basis of grain hardness in wheat. Although other tests were performed on the wheat, one of the tests consisted of measuring the grain hardness by performing a particle size index (PSI) test utilizing a LabConco mill. The fractions from the mill were sieved for two minutes on a No. 15 nylon screen in a Simon laboratory sifter. It was suggested that the adhesion between starch and storage protein is more important in determining grain hardness than is the composition of the protein matrix. Examination of pyrophosphate-soluble material surrounding the starch granules from endosperm of a range of wheats did not implicate any specific compounds as adhesives at the starch-protein interface. The observation of water-soluble material of uniform composition associated with starch granules of hard wheats might equate greater adhesion in hard over soft wheats. In another part of the study, it was postulated that protein matrices which held starch granules together were related to hardness. In order to obtain a starch and storage protein separation, the wheat was first placed in an Alpine Kolloplex Mill. Samples were run repeatedly until the flour aggregates were disrupted (this was checked microscopically by ensuring that no more than 5% of the particles by weight had a diameter of greater than 40 microns). Next, 300g of flour was suspended in 800 ml. of chloroform-benzene having a specific gravity of 1.45. The protein-rich material which rose to the top was allowed to stand for two days, and then it was purified by resuspension in the same solution with specific gravities of 1.34 and 1.32. In the study, they found that different protein compositions did not result in different levels of hardness. The strength of the bond between the starch and protein was a possible explanation for hardness. The findings of Simmonds and Barlow pointed out that starch granules of hard wheats possess a larger amount of water-soluble material of uniform composition which in itself provided an explanation for greater adhesion than soft wheats.

Simmonds (31) in 1974 investigated the chemical background of the hardness of wheat. The most effective methods for determining hardness had been techniques of grinding or abrasion (the main drawback of these techniques was the kernel size). The pearling index or pearling resistance was an example of one of the abrasion techniques used to measure hardness. The particle size index test was another test of hardness which was more dependent upon the hardness of the endosperm and the correlation between these two tests were high ($-0.92 < R < 0.96$, with a significant probability level at $P < 0.001$). Another test used to measure hardness involves crushing or indentation on single kernels. This type of hardness evaluation was affected by softening of the grain with increased moisture present in the kernel.

These tests were good for evaluating milling characteristics, but their main focus did not reveal the basic mechanisms of hardness or vitreous for wheat kernels, and this type of examination required a study of the endosperm, the interface between starch granules and the storage protein of the kernels. The aid of a scanning electron microscope had revealed that hard wheats fractured around endosperm cell walls, directly through starch granules, and through storage protein. Soft wheat kernels, on the other hand, tended to fracture through cell contents and around individual starch granules. A significant discovery in the distinction between hardness and vitreousness was that whether the hard wheat kernel was vitreous or opaque, the kernels fractured in the same way as hard wheats regardless of vitreousness. The scanning electron microscope also revealed that low protein wheats tend to be filled with large numbers of starch granules in the outer endosperm region, and that high protein wheats have the greatest proportion of the protein in the outer region, where the large starch granules have diminished.

The hardness found in wheat (a structurally heterogeneous material) could be due to possibly two conditions: 1) either storage protein or starch may be harder in hard wheats than in soft wheats. The storage protein is more likely to contribute to hardness because the storage protein formed a continuous phase in the endosperm cell contents, and the starch was found scattered in a discrete form in the outer endosperm region; and 2) the bonds between the starch and storage protein might be stronger in harder wheats, thus creating a coherent mass rather than discrete components.

Equipment has been designed to test these two alternatives and was accomplished by: 1) the particles to be tested were suspended in a synthetic resin of suitable physical strength; 2) after polymerization, the resin surface was ground and polished until the suspended particles were revealed in cross-sections; 3) a micro-hardness tester having a diamond stylus was then used to determine the hardness of individual kernels. Although it was not fully explained, the second hypothesis of the bond between starch granules and the protein matrix was believed to be the main source of hardness.

In 1975, Chung et al. (8), modified the Strong-Scott barley pearler as a technique in measuring wheat hardness. A Strong-Scott barley pearler from the manufacturer was modified by: 1) replacing the drive motor with a double-shaft motor operating at the same shaft velocity; 2) The torque on the motor was measured utilizing the trunnion dynamometer principle; and 3) a cantilever beam, with strain gages attached, extended from the frame of the drive motor to restrain the reaction to the torque of the motor. The signal from the strain gages was amplified by a Datronic Strain gage amplifier and recorded by a Beckman Strip Chart Recorder. The pearlograph curves are plots of the torque on the motor shaft versus time. The experiment was carried out by using Reed, Wells, Wanser, and Moro wheat varieties. Each wheat variety was divided into two different size categories, the first was the size between Tyler sieve No. 6 and 7, and the second between No. 7 and 8. The pearler was turned on, and after 12 seconds, the sample was introduced into the pearler, the peak height of the torque was noted, and the material was then sieved over a No. 10 sieve. This procedure was repeated at pearling times of 18, 30, 40 sec, etc., until the chart height did not indicate a load. The results from the study concluded the following: 1) the best measure of hardness was the area under the curve, and the peak height at any given time was the amount of material in the pearler; 2) in optimization of the pearling time, the pearlograph characteristics minimized the effects of kernel size and distribution; 3) the pearlograph chart area was affected by moisture of the grain in the range of 7-13% (an increase in moisture led to a decrease in area for hard wheats, a slight increase for soft wheats, and essentially no change for intermediate wheats); and 4) the optimum pearling time for a hardness index was 80 seconds (based on the maximum ratio of average effect of variety to that of grain moisture).

Baker (2) used grinding time to evaluate kernel hardness in 1977. A technique known as "inbred-backcross" which detected the effects of individual genes on quantitative characters was based on the realization that a set of inbred-backcross lines should consist largely of lines identically genotypical to the recurrent parent. The wheats studied were: Pitic 62 (a soft wheat); Neepawa (a hard wheat); and Glenlea (a very hard wheat). The grinding time was determined by pouring 6g of the sample into a Brabender SMI grinder with a setting of 17.6 and the time required to obtain 4.8g of meal. It was determined that soft wheats required more grinding time than hard wheats, and that two major genes controlled the difference in kernel hardness between Pitic 62 and Neepawa. The average grinding time for Pitic 62 was 1.221 min. and 0.536 min. for Neepawa. There was a significant variation in grinding times between these two crosses which indicated a minor gene or genes which modified the hardness of this cross. The average grinding time was 0.465 min. for Glenlea, and 0.564 min. for Neepawa grown in the same experiments as the Glenlea-Neepawa inbred-backcross lines. The analysis from grinding times revealed one major gene controlling the hardness between Glenlea and Neepawa. In retrospect, four classes of kernel hardness were recognized. The hardest kernels were equal to Glenlea with an average grinding time of 0.46 minutes. The next highest hardness index was represented by Neepawa with grinding time of 0.54-0.56 min. The intermediate class with a grinding time 0.74 min. was identified in the cross between Pitic 62 and Neepawa. Pitic 62 was the last class with a grinding time of 1.22 min. These results indicated the four classes of kernel hardness represent to a large extent the expression of 3 major genes. These findings support kernel hardness as being influenced primarily by a few major genes of the parent donor, and to a lesser degree minor genes, which were transferred.

Stenvert and Kingswood (34) studied the physical structure of the protein matrix and the influence on wheat hardness in 1977. The wheat hardness was determined by the grinding resistance method (a measure of the time taken to fill a specific volume when 20g of wheat was ground under standard conditions in a Culatti (Type 14-580) hammer mill). The physical structures were viewed under a scanning electron microscope (SEM), with samples being sliced transversely and coated with a thin layer of gold (400 Angstroms) before examination. The wheat studied was grown in the UK in the 1972-73 as well as 1973-74 growing seasons and the winter wheat (Triticum aestivum) cultivars were used. There were basically three different endosperm structure classifications which affected hardness. The first type was the mealy wheat grains which contained a very open structure with the protein matrix composed of very fragmented and were interspersed with air. This first type was disordered and easily yielded to stress. The second type was one which appeared intermediate in density, and the structure appeared more orderly with the protein matrix encapsulating the starch granules. This type was more likely to fracture or dislodge starch granules from out of this protein matrix encapsulation. The last type was vitreous and very hard grains which possessed a very ordered endosperm cell structure. The tight physical entrapment of starch granules in a continuous protein phase resulted in the contents of the endosperm cells attaining their maximum strength.

The location was the first influence on hardness investigated by Stenvert and Kingswood (34). Six samples of Pride (hard red winter) wheat of similar protein were grown at various locations and used in this study. Although variations in grain hardness for two separate genotypes grown at the different locations with the same protein content were complex, the grain internal structure and its relation to hardness were informative. In the cases studied for Pride, the wheat hardness was dependent on the degree of order they existed in the endosperm structure, and this was determined by the shear number of starch granules surrounded by the protein matrix. At each starch granule site, there seemed to be a threshold quantity of protein required to complete the formulation of a continuous matrix. At one of the sites (Wales), vitreous grains formed at a protein content of 11.4%, but an equivalent hard vitreous grain at East Anglia formed at 10.7% which suggested conditions at the latter site were more conducive to the formation of an ordered endosperm structure. The influence of protein content was studied on three different cultivars with increasing the fertilizer in order to achieve increased protein content at one location. The results demonstrated that within a single cultivar, the grain hardness was related to the protein content. There existed a minimum quantity of protein to complete a continuous matrix, and the quantity of protein and the hardness seemed to depend on the genetics of the particular cultivar. Each genotype had a unique endosperm ultrastructure. The differences in packing and the strength of the entrapment could help explain the softening of wheats during conditioning and milling. Moisture caused the endosperm components to become less dense, and the expansion of the endosperm structure would tend to weaken the starch-protein bonds and thus weaken the grain structure. The softening of wheat resulting from rain could be explained by a disruption of the endosperm structure during the wetting/drying cycles.

In 1978, Kosmolak (16) studied grinding time on a Brabender SM1 grinder as a method to determine hardness among wheat cultivars. There were approximately 10,000 samples of wheat which came from breeding programs in Ontario and Western Canada. The wheat was tested between 1974 to 1977, with moisture ranging from 10 to 12%. The grinding time was determined by pouring 6.0g ±0.5g into a Brabender SM1 grinder, at a setting of 17.6 and by recording the time required to trip a balance set to measure 4.8g of ground meal passed through the grinder. The grinder was a burr mill with a vertically revolving cone and stationary mantle. The surfaces of the cone and mantle were equipped with teeth. The grinding surface tapered off towards the bottom, and the clearance between the cone and mantle was adjustable by turning the threaded mounting of the mantle.

All of the wheats were classified correctly except for Kharkov 22 MC, which was considered a hard wheat, but it was characterized as a medium or soft wheat with a grinding time of 62-65 sec. The durum wheats tested were in a narrow time range of 24-26 sec. Glenlea, had a grinding time between hard wheats and durum wheats of 27-32 sec. The hard red spring wheats which had been registered as hard as Marquis fell in the 35-45 sec. range. The two winter wheats, Winalta and Sundance also fell in the hard red spring range. The soft wheats were categorized as 64-200 seconds. For convenience sake, the 64 sec. limit on grinding was established with the intention of classifying the wheat as soft. To characterize the medium hard and medium soft grinding times, mixtures of hard and soft wheat were ground. A linear relationship between grinding time and the proportion of hard and soft wheat kernels present in the sample was obtained. The local maxima for hard wheat was estimated at 45 sec., and for medium hard wheat 65 seconds. These grinding times were 20 and 35%, respectively, which would indicate that medium hard wheat was closer in grinding time to hard wheat rather than midway between hard and soft wheat grinding time.

The kernel size was also studied, and it was performed by subdividing five cultivars into three groups depending on kernel size. The "as is" sample was placed in a Carter Dockage Tester fitted with a 00 riddle and a 6/64" slotted screen, and the seeds which passed over the riddle were classified as large, kernels which passed through the riddle but not through the 6/64" screen were medium, and ones passing through the riddle, 6/64" screen were small. The kernel size did not effect the grinding time drastically. The moisture content was also studied with the distinction that the higher the moisture content of wheat, the softer the wheat. The effect of moisture was more prominent with soft wheats up to 14% moisture, where the wheat was too soft to grind. A factor not affecting the grinding time was the protein content. Samples of the same cultivars grown at various locations for several years varied in protein content from 11-19%, but each cultivar had grinding times within a 10-sec. range. The differences could be attributed to moisture contents. Samples of cultivars with grinding times less than expected tended to possess lower flour yields, higher ash contents, and inferior dough mixing properties. A closer examination revealed that the samples might have been exposed to frost before harvest. This was not verified, but if it were true, then a method for determining frost damage could be employed.

In 1978 Moss (24) conducted a study on how to optimize wheat hardness as measured by three different hardness measurements: 1) the Particle Size Index (PSI); 2) Pearling Resistance (PR); and 3) Wheatmeal volume (WV). The Wheatmeal volume, or packing density was determined by pouring 20g of wheatmeal through a funnel into a measuring cylinder at the rate of 1.5g/sec. It is expressed as milliliters per gram, and the funnel was 20cm above the base and the cylinder had an internal diameter of 22mm.

When the wheat from locations which received more than 5mm of rain during the month of harvest was excluded from the within-cultivar correlation matrices, the relationship between pearling resistance and hectoliter weight became strongly positive. The pearling resistance, when correlated to protein, was more strongly correlated in this group. A negative relationship between hectoliter weight and wheatmeal volume was less pronounced, and an often negligible relationship between pearling resistance and particle size index became significantly negative. It was apparent that even a small amount of rain affected grain hardness. The variations in kernel density affected pearling resistance but not wheatmeal volume or the particle size index. An increase in moisture reflected a softer grain in the particle size index and wheatmeal volume, but ranked harder in pearling resistance. It was also pointed out that the various tests responded differently with effects of grain size, protein, and moisture content. The fibre content did not appear to be significantly related to grain hardness. The grain characteristics differed between cultivar to cultivar, from location to location, and from year to year, and the relationship between these characteristics varied. Discrimination between cultivars was practical with each of the small-scale tests, and the relationships derived in this study should allow wheat breeders to select new cultivars with grain hardness appropriate for the given protein level desired for cultivation.

Bulk density was negatively correlated to PSI and WV, and positively correlated to PR. Protein was positively correlated to PSI and WV, negatively related to PR. Where correlation coefficients were obtainable, grains became harder with increasing bulk density, with increasing kernel density, and became softer with increasing protein content. The pearling indicated a harder grain with increasing moisture, while PSI and WV indicated greater softness.

In 1979, Kuhlman et al. (17) studied six different wheat varieties for hardness utilizing a modified barley pearler. The barley pearler was modified in the following ways: 1) a double shaft motor operating at the same speed replaced the original motor; 2) ball bearings were mounted on the motor shaft; 3) the base supporting the apparatus was extended; 4) a couple arm connected to the motor was extended in a cantilever beam which was used to counteract the torque of the motor; 5) strain gages were mounted on the coupled arm; 6) a rectangular gate was constructed to keep the sample from the sample-release gate; and 7) baffles were introduced at opposing angles inside the chute to retain possible particles from escaping. In addition to the modifications, a

Syntron vibra-flow feeder was used to introduce a way of standardizing sample introduction into the modified barley pearler. A wattmeter and an ammeter were connected to the motor to record power measurements. The output from the strain gages was amplified and recorded on a strip-chart recorder. The six wheat varieties analyzed were: hard red winter; soft white winter, hard white winter, hard red spring, soft red winter, and Durum. These varieties were tested at approximately 9, 12, and 15 percent moisture content. The procedure involved adjusting all recorders to zero, following which a 40g sample of wheat was introduced into the running barley pearler for a length of 80 seconds. At this time the sample-release gate was opened, and the sides were tapped to dislodge any material remaining in the pearler. This sample was then placed in a Tyler No. #10 sieve and was shaken to remove the dust. The sample was weighed, and this value (as a percentage of the original) was the pearling index. The strip-chart was coded to the sample and this procedure was repeated five times for each sample. Results from the investigation were as follows: 1) the pearling index was most likely affected by moisture in terms of the hardness level; 2) the pearler peak torque and pearlograph area (area under the strip-chart) was revealed as the most likely to yield hardness values; and 3) the peak wattmeter reading and wattmeter area responded to moisture effects more readily and less within each class of grain. It was concluded that the best range of moisture content for wheat-hardness testing on the pearlograph-area method was approximately 9-12%. In conclusion of the modified barley pearler, it was stated that the pearlograph area method of determining hardness would be fast, economical, and efficient. However, further testing is needed to be conducted to support this conclusion.

Williams (39) utilized Near Infrared Reflectance Spectroscopy (NIRS) in addition to mean particle size (MPS) to screen wheat into protein and hardness classes in 1979. Two separate groups were run in order to test two different types of grinders. The first series was tested by grinding samples on a Udy cyclone sample mill (1.00-mm screen), which was normally utilized by the Canadian Grain Commission in conjunction with NIRS testing for protein. The second set was ground on a Hobart Model 2040 grinder using pulverizing burrs. Hardness was measured by a modified particle size index test. A well mixed sample of 25g was split into two sub-portions of 10g each and were sieved for ten min. on a Rotap sieve shaker, using 200 mesh stainless-steel screens with an aperture of 74 microns. The PSI and MPS were evaluated with a sample size of 25g instead of 10g due to the effect of volume of grain and hardness on the MPS. The Hobart Model 2040 coffee grinder was much faster than the LabConco and it was also designed for self-cleaning. A check sample was run through the Hobart after every tenth sample to check the uniformity of the MPS. The check sample was standardized against a LabConco burr mill and then the PSI was calculated for the check sample. The MPS of the ground samples were assessed by an arbitrary system which involved sieving for 15-min. through a nest of five stainless-steel sieves (35, 45, 70, 100, and 200 Mesh) and the weights of the throughs and the overs on the top sieve were multiplied by the apertures of the sieves to yield a standard of comparison of the MPS among all of the wheat samples.

The PSI figures from Williams study (40) obtained by the Hobart were closely related to the standard LabConco PSI figures but not the Cyclotec figures. The 1.0mm screen on the Cyclotec grinder reduced the variance in the PSI figured, which was caused by the texture of the grain. The PSI figures from the Hobart and LabConco established the following order of hardness: durum, Australian varieties of hard white spring (HWhS), hard red spring (HRS), hard red winter (HRW), hard white winter (HWhW), soft red winter (SRW), soft white winter (SWhW), and soft white spring (SWhS). The MPS as determined by the sieving method provided satisfactory results to compare wheats and determined the reproducible results of a grinding technique. The similarity between the Hobart and LabConco PSI figures led to the usage of the Hobart to determine the MPS. Since the PSI could determine hardness, a NIR spectrometer could be calibrated to the PSI values for Hobart-ground samples. The Hobart-ground samples were discernable among varieties on the basis of the PSI hardness. The PSI testing by Hobart-ground samples were introduced into a Neotec Model 31 Grain Quality Analyzer (GQA), and more discrimination was achieved over the PSI hardness. The inherent accuracy of the GQA outperformed the weighing involved in a typical PSI test. Optimum accuracy could be achieved using the GQA if both hard and soft wheats had their own calibrations. Also, a person grinding with a Hobart, and then testing the hardness on a GQA could possibly analyze 200 samples in one day. Further testing was needed to determine the feasibility of such a method as well as the validity of the NIRS method for hardness.

Obuchowski and Bushuk (26) modified a two-stage Brabender Hardness Tester (BHT) to study the effects of protein and moisture on the hardness of a hard red spring wheat (11-604) grown at one location. The samples were first tempered to the five levels of moisture (9.5, 11.0, 12.5, 14.0, and 15.5%) and then they were pearled in a Strong-Scott Barley Pearler to achieve a yield of 65% pearled product. The two-step Brabender Hardness Tester was modified in the following ways: 1) the position of the indicator levers was set as for the 50g Farinograph mixing bowl; 2) the damper was set to achieve a recovery from 1,000 to 100 Bu in 4sec.; and 3) the speed of the chart paper was slowed down from 7.2 $\frac{cm}{min}$ to 1.0 $\frac{cm}{min}$. The torque measured by the two-step BHT was increased by 23-69%, and all other indices of hardness decreased (energy input by 1-14%, grinding time by 30-49%, average particle size by 7-12%, and particle size index of flour by 4%). The torque on the one-step BHT was decreased 8-28% which was contrary to results on the two-step BHT).

Debranning the grain was significantly correlated to the classification obtained by whole grain, except for the results of the one-step BHT torque. Debranned wheat showed an optimum differentiation among wheat cultivars in most indices (energy input, grinding time, PSI, and torque) at a moisture content of 12.5-14.0% moisture. The optimum moisture content for the two-step BHT was 12.5%, the Quadrumat Junior mill was 14.0, and the one-step BHT was 15.5%. Through analysis of variance, results of underbranned samples of different protein contents showed significant differences evaluated by all but two methods. There was no difference in hardness found in the nine samples for grinding time or the particle size index. There was however, a highly significant negative correlation between protein content, WHI, and average particle size. A positive correlation existed between protein content and flour yield from the two-step BHT. Since there was no correlation between the measurement of hardness and protein content for debranned wheat, this supported the hypothesis that bran had an influence on grain hardness evaluation.

Miller et al. (22), used an accessory burr mill for the Brabender Farinograph to determine the hardness of wheat in 1981. The modifications to the Brabender Farinograph burr mill attachment were : 1) The mechanical recording system was disconnected and an LVDT (Linear Variable Differential Transformer) was connected under the right-hand Farinograph lever arm, 0.20 meters from the center line of the drive shaft; and 2) an aluminum encoding disc with 360 machined slots on its surface was mounted on the mill shaft. The data obtained from the LVDT in addition to data from the Strobe tachometer from the encoding disk were digitally recorded after passing through respective filters. Four different cultivars were used in this study. Chiefkan-Tenmarq (hard red), Buckskin (winter), and Nugaines (soft white winter) were used to study the effects of temperature, moisture, and reproducibility. Centurk (HRWW) was used for studying kernel size effects. There were also protein, growth location, and variety effects on work required to grind the wheat. The results from the experiment determined that sub-samples of Nugaines (a soft white wheat) and Chiefkan-Tenmarq (a very hard, hard red winter wheat) were reproducible in narrow limits. Kernel size was found to directly affect the work required to grind, but only in a small way. The protein effect had no significant impact on the work. The location (with the similar protein content) had no apparent effect on the work. Miller concluded that the burr mill will quickly and precisely measure the work required to grind 25-55g. The growth location, protein content, temperature during grinding, and kernel size had little affect on the work required to grind wheat at 12.8% moisture with the Brabender Hardness Tester. Also, the work to grind a 50g sample increased as the moisture content increased from 7% to 13%.

Kilborn et al. (15) measured the energy consumption during flour milling in order to determine the relative hardness of wheat kernels. Energy requirements were measured using a strain gauge directly connected to the roll stand, and a watt transducer. Both instrument setups were sensitive enough to detect relative changes within a series of hard red spring wheats. The flour starch damage and break release flour were two widely used measurements for wheat hardness, and these two were highly correlated with the energy requirements. The break release flour was determined by running the wheat through the rolls and sifting the flour over a set of wires, and the percentage of the sample which passed through the 20 wire was considered the break release of the flour. Thus, if flour starch damage, and break release flour were added to the energy requirements, a relative index for hardness could be developed.

Four different methods of measuring hardness were employed by

Miller et al. (21) in 1982. Hardness measurements were obtained by: 1) the work required to grind 50g of wheat measured by a Brabender hardness tester; 2) the time required to grind 4g of wheat at 15°C; 3) the NIR (Near Infrared Reflectance) data (at 1680nm) of wheat ground on a Brabender automatic micro hardness tester with measurements from a Technicon InfraAnalyzer; and 4) grinding a 2g sample on a Brabender micro hardness tester, sifting and pulsing on a U.S. No. 140 (106 micrometer opening) stainless steel sieve using a Model L3 sonic sifter, utilizing the sifting time to the percentage of flour passed through the sieve. The effect of protein content on hardness was first studied on Lancota (hard red winter wheat) ranging from 10.5%-15.9% with no significant effect on any of the four methods. Scout, a hard red winter wheat, also was not affected significantly by protein. The hardness of a commercial hard red winter wheat sample was not affected by protein.

The effect of location on hardness was investigated, and since only single samples from each location were used on the time to grind 50g, and the NIR hardness, no comparisons could be made. However, there were three subsamples used to measure the time to grind a 4g sample, and the percentage of throughs from a No. 140 sieve. The magnitudes were very slight, but the data suggested that some unknown factor(s) in the environment might affect the hardness of wheat. The effect of irrigation on the three samples revealed consistently higher protein contents than on non-irrigated land. A reduced time to grind 4g of wheat, and a slightly increased NIR hardness value was observed in the irrigated wheats over the non-irrigated wheats. These results also carried over to rain-fed and irrigated samples of Eagle and Sage (hard red winter wheat cultivars). The protein content increased 1.8, 1.2 and 2.0%, and the time to grind 4g decreased 2.9, 3.5, and 2.4sec., and NIR hardness values increased 14, 21, and 23 for three sets of dry and irrigated Centurk, Eagle, and Sage wheats, respectively. Since there was a lack of observations, it was not possible to conclude any significant effect of irrigation on any of the samples for Centurk hard red winter wheat.

The wheat provided by the Federal Crain Inspection Service (FGIS) shows that wheat could be classified by the hardness measurements, however, the hard red winter and hard red spring could not be separated. In the study, 20 samples of durum, 27 samples of hard red spring, 17 of hard red winter, 27 of soft white and 15 samples of soft red winter wheat were used. Results from this study indicated that Durum could be distinguished from all other varieties by three of the four methods. The hard red winter and hard red spring could be separated by two methods, but by very small margins. White and soft red winter wheats were distinguishable by the Brabender automatic micro hardness tester that measured the time to grind a 4g wheat sample. The NIR hardness value measured at 1680nm distinguished hard, soft, and durum from all other classes and thus should be used as a means for determining hardness.

Yamazaki and Donelson (43) studied the effects of moisture content on the Particle Size Index (PSI) test. The PSI test for this experiment consisted of placing a 20g sample of grain into a LabConco Heavy Duty mill equipped with special burrs, and then the material passing through the mill was collected. A 15g sample was weighed on a round 20cm metal screen (425 micrometer opening) over a pan and then it was sifted for 30 seconds on a rotary sifter (190 rpm, 10 cm throw). The amount of material through the screen was weighed and the PSI was calculated as the percentage of material passing through the screen. In the tests, at a moisture content of 11%, the PSI range for hard and durum wheats was 20-30%. The durum wheats were in the low 20's, and hard red spring and winter wheats fell in the range of about 24-30%. Soft wheats ranged from 30 to almost 60%, with most in the 35-45% range. Eastern soft whites were mostly about 33-38%, and part of the current southern soft red cultivars were as high as 55-60%. The conclusion made from this experiment was that, within a cultivar, the correlation coefficient between moisture content and PSI was highly significant and denoted that within a cultivar, increasing moisture content increased the PSI value.

Miller et al. (23) in 1984, studied the effects of hard red winter (HRW) wheat grown in a soft red winter (SRW) wheat area, and SRW wheat grown in a HRW wheat region. Many samples from different cultivars across several countries were grown in Lafayette, IN and harvested in the years of 1979 and 1980. Samples were also obtained from Atchison, KS in 1980, where 30-35% of the crop was soft wheat. Also, samples of the cultivar Newton from the 1980 crop were obtained from 13 locations across the state of Kansas. Results from the study indicated higher test weights for hard wheats than soft wheats. Another finding was the large differences in hardness between hard and soft wheats at all locations, however, there was no overlap between wheats from the two classes at any single location or among all locations. Investigation revealed both hard and soft wheats grown in the soft wheat area (Indiana) were "softest", and were found to be "hardest" when grown in the hard wheat area (Montana), with average values from eastern Kansas. Also, the average differences in grinding times between hard and soft wheats were greatest in Indiana (146.9 and 132.3 sec.), and the smallest in Montana (79.9 sec.), and intermediate in eastern Kansas (126.5 sec.). The results could probably be explained by the greater effect of wheat softening, and the wider range of experimental values of the time to grind, for soft wheat. A fairly good sequence of increasing time to grind moving from typically hard to soft wheat areas was discovered.

Pomeranz et al. (27) studied the effects of kernel size and sprouting on the three methods of kernel hardness: time to grind; particle size of ground wheat; and near-infrared reflectance of ground wheat. Three samples of plump and three samples of shrunken hard red winter wheat cultivars (Centurk, Scout 66, and Newton) grown in Manhattan Kansas in 1980 were used. These samples were cleaned on a dockage tester and then by hand to eliminate broken kernels and foreign material. The samples were then sieved on Tyler sieves No. 7, 8, 10 and 12 (with openings of 2.794, 2.380, 1.651 and 1.397 mm, respectively) for 2 min. Five samples of soft white wheat from Saginaw, Michigan and six samples of western white wheat from Pullman, Washington which varied in the percentage of sprouted kernels from 0.2 to 52.3% and 3.4 to 36.2%, respectively, were also used. Plump kernels were obtained by placing these samples on a dockage tester. All the samples were stored at 28-29°C and

50-60% relative humidity to produce a moisture content of 12.8% ±0.4%. Also, four wheat samples: hard red spring (cv. Weather Master 99), hard red winter (cv. Newton), soft white (Nugaines), and soft red winter (Hart) were germinated. The time to grind the wheat was performed by the Brabender Automatic Micro Hardness Tester. The particle size index was determined by grinding a 2g sample on a micro-Brabender hardness tester, then it was sifted and pulsed on a Model L3 Sonic Sifter (ATM Corp. Milwaukee, Wisconsin) at an amplitude setting of 6. The NIR reading was taken on a Technicon InfraAnalyzer at 1680nm by wheat ground on a Udy mill. The results from the six original wheats showed that as the shrunken kernel size decreased, the grinding time increased, the NIR reflectance values at 1680nm decreased, and the particle size index decreased. The increase in grinding times was possibly related to a packing effect of slender kernels that were low in starchy endosperm contents near the pericarp. The correlated increase in particle size indices were reflected in resistance to grinding by kernels with a high content of fibrous material. Results of debranning the kernel were consistent with Obuchowski et al. (26), which showed a decrease in: 1) energy requirement for grinding; 2) time to grind; and 3) average particle size of the flour. As far as sprouting, hard winter wheats seemed to mellow (grinding times and particle size indices increased and NIR values decreased), whereas the two soft wheats were correlated to the grinding time and particle size index and negatively correlated with NIR values.

In 1985, Lookhart et al. (19) used instrumental crushing characteristics to determine hardness of soft red winter, two hard red spring, and six hard red winter wheat cultivars. The crushing apparatus first aligned the kernels one at a time by virtue of a vibrating feeder, at which point the kernel was picked up by a vacuum head. The kernel was transported to a testing cup on a rotating disc. The kernel was subsequently crushed and the crushing energy signal was recorded on a floppy disk. The hard kernels could be separated from the soft by a distinct drop in the crushing force after the first peak. The soft wheat had a gradual drop and it was relatively small. Each curve was characterized by measuring the ratio of the first valley over the first peak. The predicted hardness value (PHV) was determined from a combination of the ratio of the first peak to the first valley and the magnitudes of each. If this ratio was less than 0.25 or greater than 0.45, then the PHV was hard or soft, respectively. They noted that the soft wheats normally had a first peak to first valley ratio greater than 0.4 and hard wheats less than 0.3. The relative protein content seemed to have an effect on the energy required to crush a single seed (first peak height) and the intensity of the electrophoretic bands. Also, high protein content seeds, as indicated by intense electrophoregram bands, affected intermediate hardness values. The results from the study were not necessarily related to hardness. It was also noted that gliadin patterns were not categorically related to hardness, hardness measurements were not necessarily related to phenotype, many cultivars were significantly heterogeneous; and that standards based upon morphological characteristics for a grading system by itself might relate to the genetic background, but were of limited value for most varieties in this study.

In 1985, Pomeranz et al. (28) used four different methods in order to determine the hardness of wheat. The four methods used were: the time to grind 4g of wheat using the Brabender automatic microhardness tester (BMHT); the particle-size index (PSI); the near infrared reflectance (NIR) method at 1,680 nm; and the Stenvert Hardness Tester (SHT). The PSI and NIR tests were performed on samples that were ground on the BMHT. The ranges and coefficients of variation for the hardness measurements for soft wheat were much higher than for hard wheat. They also found that NIR measurements were the most powerful method for determing the composition of mixtures prepared from two samples of known hardness. There was little overlap between analytical parameters from the four methods, however, these methods were inaccurate when a mixture of hard and soft wheat were mixed.

Pomeranz et al. (29) studied 15 varieties or selections from the 15th International Winter Wheat Performance Nursery. The wheat hardness was measured in the following ways: 1) the time to grind 4g of wheat using a Brabender automated microhardness tester; 2) particle-size index (PSI); 3) near-infrared reflectance (NIR) at 1,680 nm; and 4) resistance to grinding by the Stenvert mill. The hardness measurements represented means for duplicate sub-samples. The objective of the study was to determine the effects of the environment, kernel weight, and protein content on the varietal hardness characteristics. In the study, it was found that the correlation coefficient between 1,000-kernel weight and protein content was not significant. The 1000-kernel weight was related to resistance to grinding and PSI (probably through the effect of kernel shape and ratio of starchy endosperm to outer kernel layers). The final results of the study were summarized in the following ways: 1) the variation in hardness of winter wheat grown under widely different environmental conditions was found to be affected mainly by genotype and to a small extent by growth conditions; 2) wheat hardness was considered to denote a characteristic of wheat class and variety and might be modified by environmental factors; 3) the kernel size might modify hardness characteristics to a limited extent; and 4) protein content affected hardness within a variety, rather than across all varieties.

In 1986, Williams and Sobering (40) sent 12 samples to 9 collaborators to test the hardness of wheat using the grinding/sieving (Particle Size Index, PSI) test for hardness. The test conducted variations on the pearling index and variations on the PSI utilizing the Udy cyclone grinder and then applying the meal to a NIR spectrometer. The twelve samples consisted of: two soft red winter (SRW) samples; two soft white winter (SWW); two soft white spring (SWS); a hard red spring (HRS); a hard red winter (HRW); and a durum. All collaborators were able to distinguish between all samples with significant differences. The PSI test based on grinding in an approved burr mill could be used to clearly differentiate among wheat varieties. This had merit in terms of a sensitive method for classifying wheat on the basis of hardness due to the fact that even differentiation between SWS and SWW was possible with the grinding test. The nine collaborators agreed upon the order of classification but differed in the absolute results of the grinding. The moisture content of the sample adversely affected the grinding action of burr mills and any grain with a moisture content over 14% should not be used for PSI testing. The LabConco grinder gave satisfactory PSI results, but they were more variable than the falling number KT series burr mills.

Williams and Sobering (40) found that the NIR technique gave the best results in terms of differentiation and the highest correlation to the PSI values. The NIR was most suited for rapid determination and it also provided moisture and protein content in addition to yielding hardness values. In order to obtain a standard among labs, a check sample should be run on all laboratories included in the hardness measurement.

In 1986 Williams and Sobering (41) attempted to standardize a Near-Infrared Reflectance technique to determine hardness in wheat. A sample size of 100g was used for all of the tests, and samples of: hard red spring (HRS); hard red winter (HRW); soft red spring (SRS); soft red winter (SRW); soft white spring (SWS); soft white winter (SWW); white club (WC); and durum were used for calibration and verification series. The grain was mildly tempered or dried and then allowed to equilibrate for two weeks in plastic bags. The moisture was determined by the AACC two-stage oven method. The samples were then ground in Udy cyclone grinders, which were fitted with a 1-mm screen. There were 21 collaborators which participated in the study with all except one in North America and one from Australia. Each of the collaborators received 10 calibration wheats and then were asked to classify 20 test samples on the basis of the ten calibration samples. Once the samples had been classified, they were returned to determine the PSI of the samples.

All of the collaborators reported close to the same results for hardness. Variability in log 1/R (R, reflectance) values at all wavelengths, the highest value for soft wheat was greater than the lowest value for the hard wheat. The differences were highly significant and it would be impossible to establish guidelines for hardness based on raw data (i.e. the log 1/R values at the 1,680 nm wavelength.). The variance in optical data was supported by a combination of grinder and instrument variation. Also, a large variation in grinders made it impractical to base the hardness on the basis of sieved PSI. Calibrations among NIR units between locations were established with all collaborators reporting the same hardness ranking, but with varying hardness values. The large variance which occurred in raw NIR data from all types of instruments in addition to the particle size variance on different cyclone grinders showed the difficulties in establishing an NIR method of measuring hardness based on raw optical data without the calibration and verification phases of the experiment.

All of the above techniques could be used to determine hardness in their own relative way, however, each method lacked separation in mixtures of hard and soft wheat. In order to determine the hardness of mixtures, a device which measured the relative hardness of each kernel individually was required. One such individual crusher was developed by the U.S. Grain Marketing Research Laboratory in Manhattan, Kansas, the tester built was called CASK-HaT, continuous automated single-kernel hardness tester. The CASK-Hat measures compression forces as a function of time (Lai et al. (18)). The maximum rate of CASK-Hat is 15 kernels per minute which was faster than an Instron, however it was not quite practical for traders to use for sales of wheat on the market.

MODES OF GRAIN KERNEL FAILURE

There are two different modes of failure in analyzing the breakage of the kernel. The first setup is the case of the sharp blade and a wheat kernel (Figure 1). The top of Figure 1 depicts the sharp blade impinging on the wheat kernel. After a short time, δt, the blade initiates a cutting action which begins the separation of the kernel into two distinct fragments (Center of Figure 1). Finally, the bottom of the picture in Figure 1 exhibits the final fracture of the segmented wheat kernel. In retrospect, the sharp blade initially slices into the wheat kernel until the wheat kernel fractures into two fragments.

The second mode of failure for wheat kernels in the Single Kernel Wheat Hardness Tester (SKWHT) is fracture. This mode of failure occurs when the blunt or curve blades are utilized. The breakage event is initiated by the contact of the blunt blade to the kernel (Top of Figure 2). After a small amount of time , δt, the blunt blade exerts pressure on the wheat kernel, and instead of penetrating into the kernel, as is the case of the sharp blade, the blunt blade deforms the shape of the kernel (Middle of Figure 2). This deformation of the kernel then progresses into a fracture initiating on the opposite side of the blade contacting with the wheat kernel (Bottom of Figure 2).

To reiterate, the sharp blade initially cuts the kernels until a fracture occurs, whereas the blunt blade fractures the kernel and the fracture initially starts on the opposite side of the blunt blade. The sharp blade is thus said to "slice" the kernel, and the blunt blade

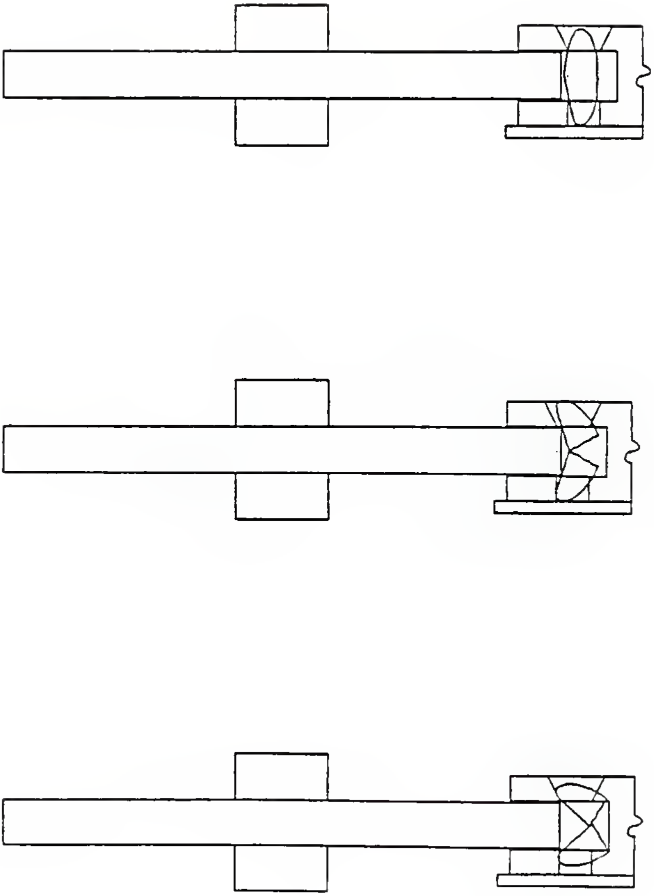


Figure 1. Blunt Blade Failure Mode

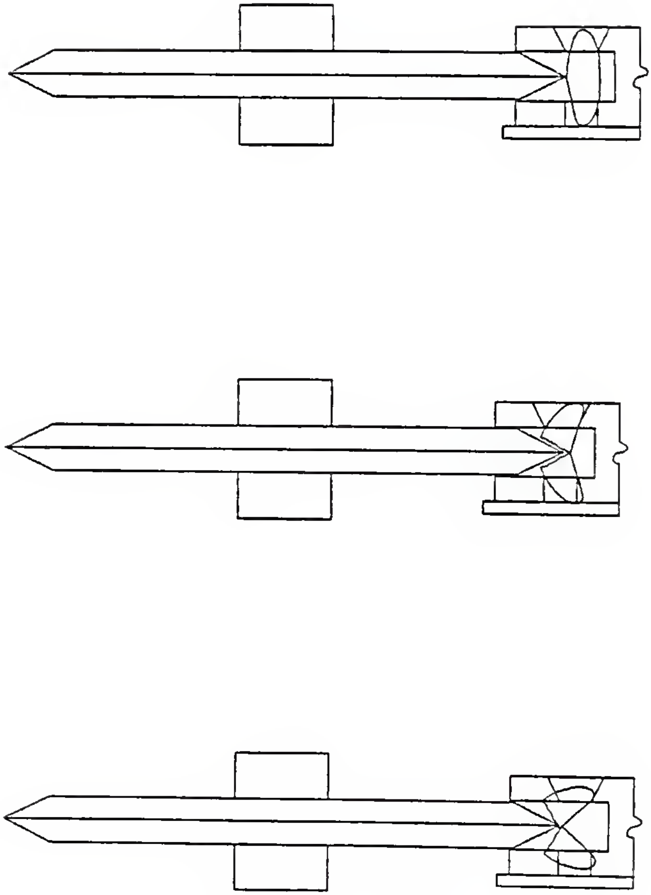


Figure 2. Sharp Blade Failure Mode

"fractures" the kernel. The differences in the maximum force readings for the blunt blade well exceeded the values for the sharp or the curve blade due to the fracture mechanism involved in the blunt blade. Since the blunt blade's fracture starts on the opposite side of the material being crushed, a higher force is required to fracture the material and thus the values yielded from the blunt blade are much higher in magnitude.

METHODS AND PROCEDURES

Component Description

The Single Kernel Wheat Hardness Tester (SKWHT) has been described by Eckhoff et al. (10). The SKWHT was designed to discriminate based upon the difference in the fracture mechanics of soft and hard wheat. Soft wheat exhibit a more ductile fracture than hard wheat, and in concept, these differences can be measured as a means of discriminating hard from soft wheat. The basic operation of the SKWHT is to separate each kernel individually from a batch sample and then slice each kernel using a free spinning rotating knife. The knife is connected to a load cell which measures the force for each individual kernel. The breakage event begins at the point from which the wheat kernel initiates contact with the knife to the point at which it is completely sliced. Previous testing has shown that the most discriminating value which can be extracted from the breakage event is the peak force and was thus chosen as the basis for determining kernel hardness. Before the experiment is started, a few minor adjustments are made to the SKWHT. The rotational velocity of the rotating plate is set via an adjustment by a rheostat from 0.628 to 1.466 rad/s. In addition to the rotational velocity, the knife's initial blade clearance into the rotating plate must also be set. The knife's blade clearance is set by utilizing a set of feeler gages placed between the outer radius of the rotating plate and the upper ridge of the knife. The knife's blade clearance is set as shown in Figure 3 for the blunt blade at the maximum clearance value of 1.175 mm (C). The distance from the outer edge of the rotating plate to the back of the hole was 7.019 mm, which is labeled as (E) in Figure 3. In the example of the blunt blade, for a clearance of 1.175 mm, the gap distance labeled as (B) was set to 1.802 mm. Each blade has a different outer radius (A) on Figure 3, and thus it is necessary to adjust the gap setting (B) to achieve the correct clearance (C). The hole diameter (D), 3.969 mm, is constant throughout the experiment since the same hole is used for all subsequent data collection. Once these settings are adjusted, then the testing phase can progress. Kernels are initially placed in the Syntron Magnetic Feeder (Figure 4); then the kernels are separated individually, and ascend the spiral ramp to a point where they fall into the drop tube. As the kernel exits the drop tube it falls into one of the 48 holes (0.3969 mm [5/32"] diameter holes) in the rotating plate with a radius of 7.785 cm (Figure 5), where it is positioned to be sliced. If necessary, the kernel is aligned by the seed alignment air tube (Figure 5) using a stream of air. As the kernel is cut by the knife, which is connected to the load cell (Figure 6), the load cell measures the cutting force. The analog signal from the load cell is converted into digital values by a built in A/D converter in the Tecmar Data Acquisition Board (Lab Master). The A/D converter was set up to sample at a rate of 5 kHz which is well below the rated sampling frequency of 50 kHz. The A/D converter is interfaced to an IBM PC compatible computer, where the digital values are stored by the C program listed in Appendix B onto 360K floppy diskettes. The data stored on the

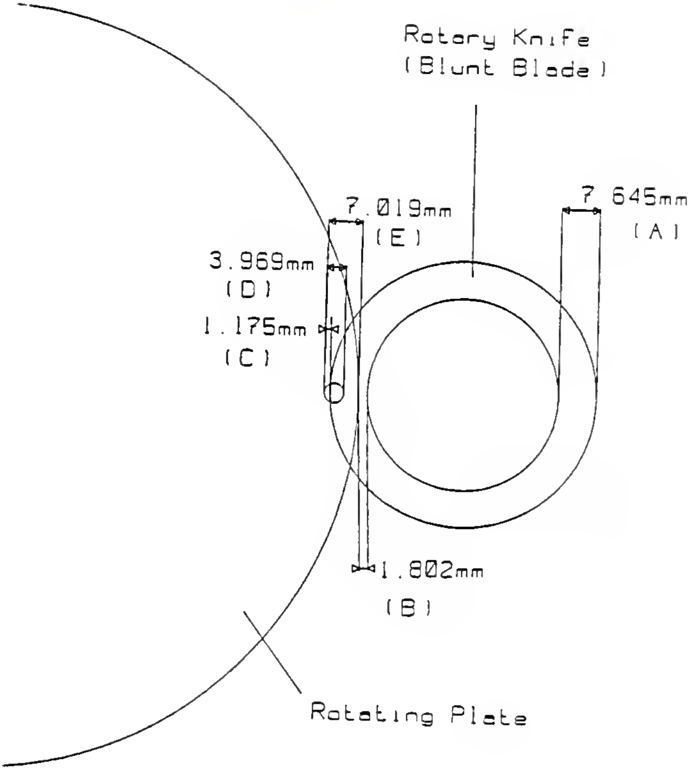


Figure 3. Clearance Setting

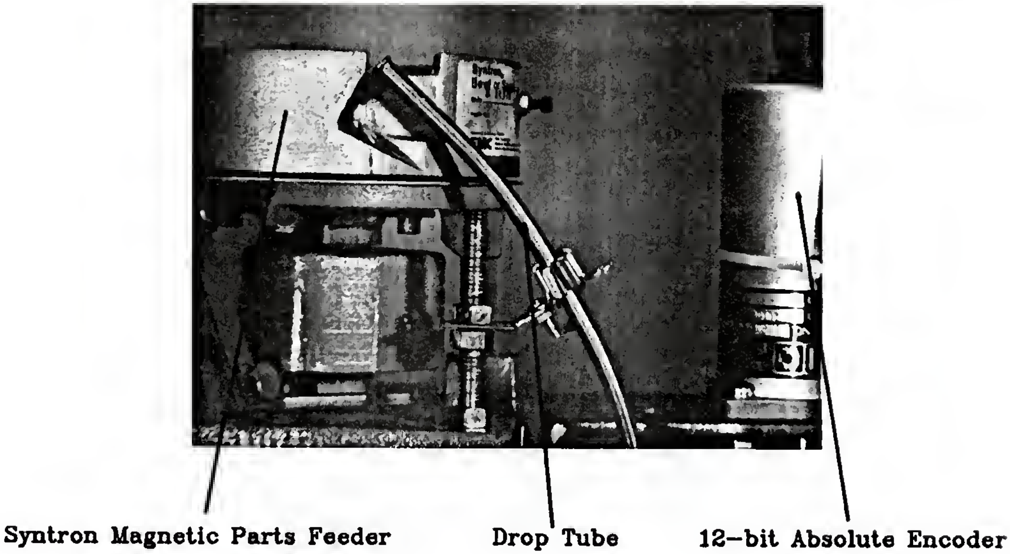


Figure 4. Initial Kernel Path

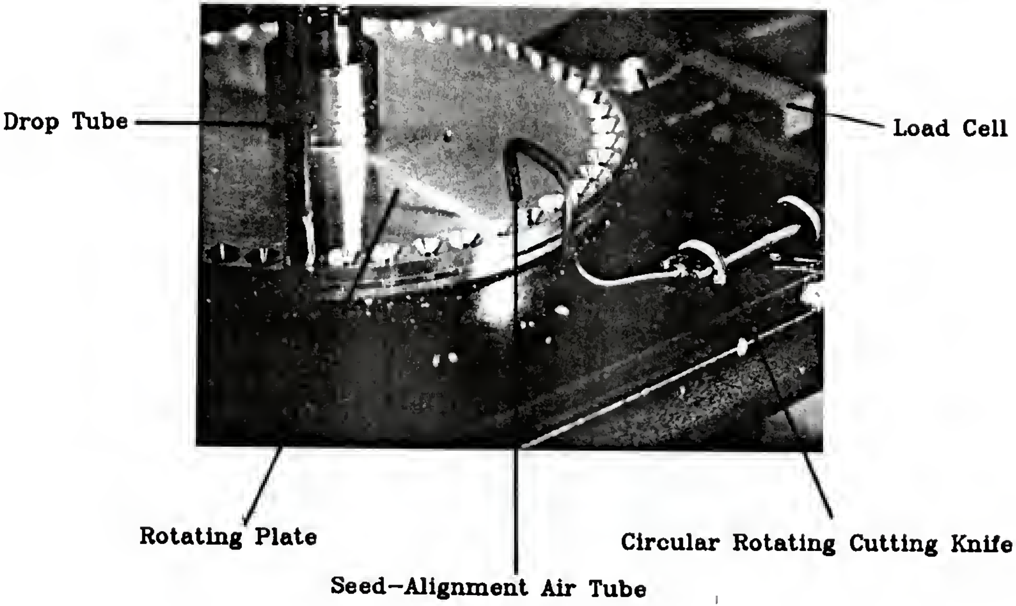


Figure 5. Pictorial View of the Seed Alignment Air Tube

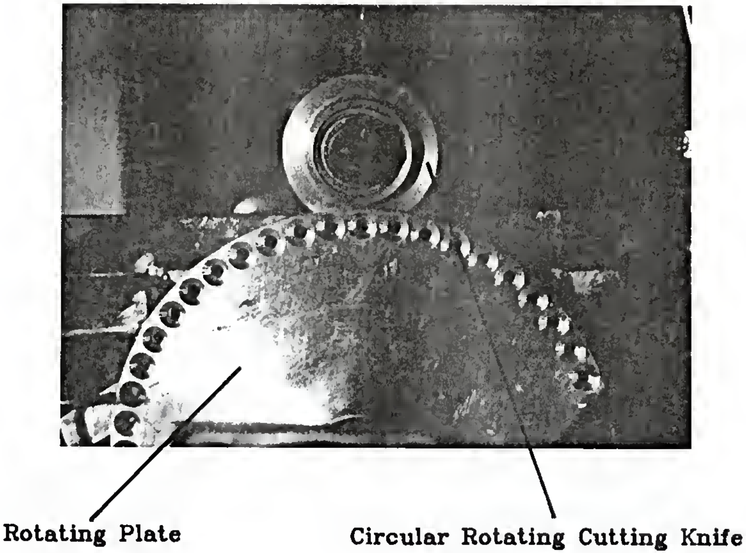


Figure 6. Top View of the Rotary Knife and Rotating Plate Assembly

360K floppy diskettes are then transferred to the Agricultural Engineering's MicroVax II computer for further data processing.

The force exerted by the wheat kernel on the circular rotating cutting knife is recorded at a regular time intervals (at a rate of 5 kHz). The computer commences recording data from the load cell when a threshold force on the load cell is reached (approximately 0.4536 kg). Usually 300 digital samples from the load cell capture the whole breakage event, however in this investigation 500 digital samples were taken.

After the breakage event is completed, the kernel fragments are cleaned out. An air stream from the first clean-out air tube (Figure 7) is used to remove particles which can easily be blown out of the blade groove and plate holes. Particles lodged in the groove are loosened through the use of a clean-out scraper (Figure 7), and a second air stream cleans these dislodged fragments out.

The main thrust of this investigation is to delineate hard from soft wheat with an unknown sample introduced into the SKWHT. Before proceeding to this stage, it is necessary to grapple with the differences depicted by the SKWHT. A representation of a typical hard wheat kernel (Mustang at 9% m.c.) is shown in Figure 8. The breakage event shown in Figure 8 contains a rapid increase in force until the kernel finally fractures. This characteristic is more typical of a brittle material. A ductile material, on the other hand, will not possess the rapid rise in force up to the fracture point; it will rise up to the fracture point with a less steep ascent to the fracture point. The ductile material yields to the force more than a brittle material, and thus

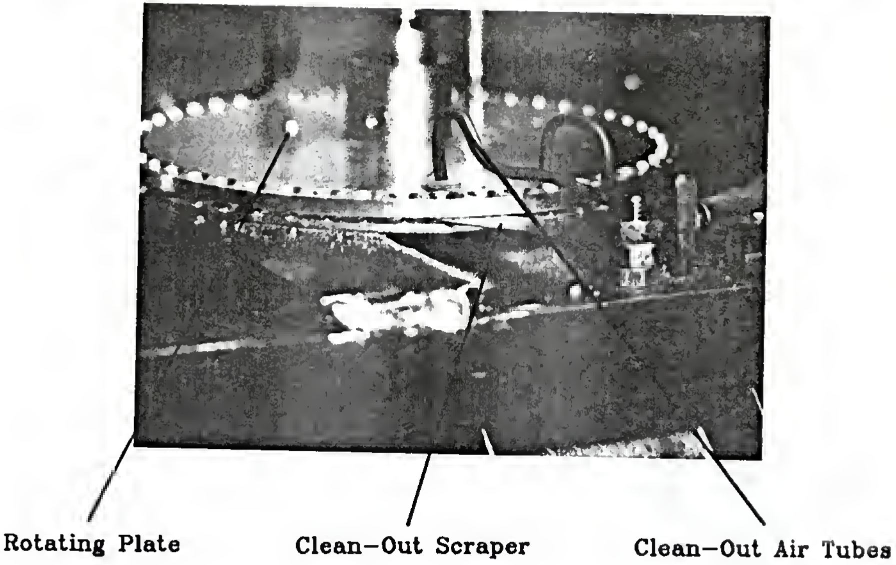


Figure 7. Pictorial View of the Kernel Clean-Out Assembly

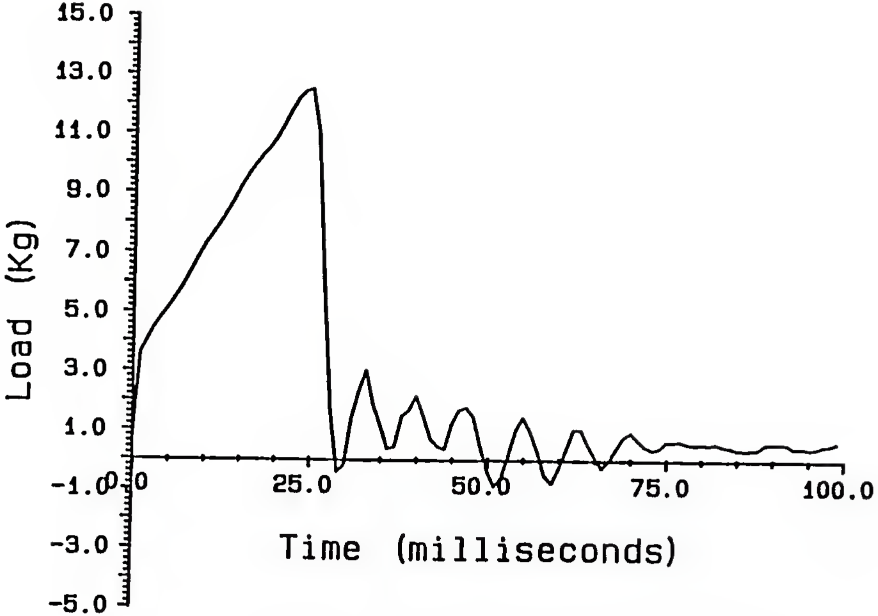


Figure 8. Representative Hard Wheat Breakage Event

the force does not rapidly rise to the fracture point. A typical breakage event to exemplify a ductile breakage event is shown in Figure 9 (Daws at 9% m.c.).

Calibration Procedure

Before testing wheat on the SKWHT, a method for calibrating the tester was sought. Selecting materials to calibrate the SKWHT were chosen on the following guidelines: 1) the material should be non-biological to reduce errors caused by interaction with the environmental conditions (mainly relative humidity); 2) the material should emulate a breakage event in terms of the characteristic shape of the breakage event for hard and soft wheats; and 3) the material should be easy to clean out of the SKWHT to ensure fragments of the calibration material would not interfere with the actual testing.

Hard wheat breakage events tend to emulate a brittle fracture as was shown in Figure 8. A soft wheat breakage event simulates a ductile fracture (Figure 9). Berol Turquoise T2375 5H (a mean radius of 0.2032 cm) drawing lead was chosen to emulate the hard wheats' breakage event, and Scripto Red Crayon Marking lead (a mean radius of 0.2946 cm) was used to emulate soft wheats' breakage events.

A typical breakage event for 5H pencil lead sliced using the sharp blade is shown in Figure 10. Notice that the force increases until the ultimate strength of the 5H pencil lead is reached, and then at this

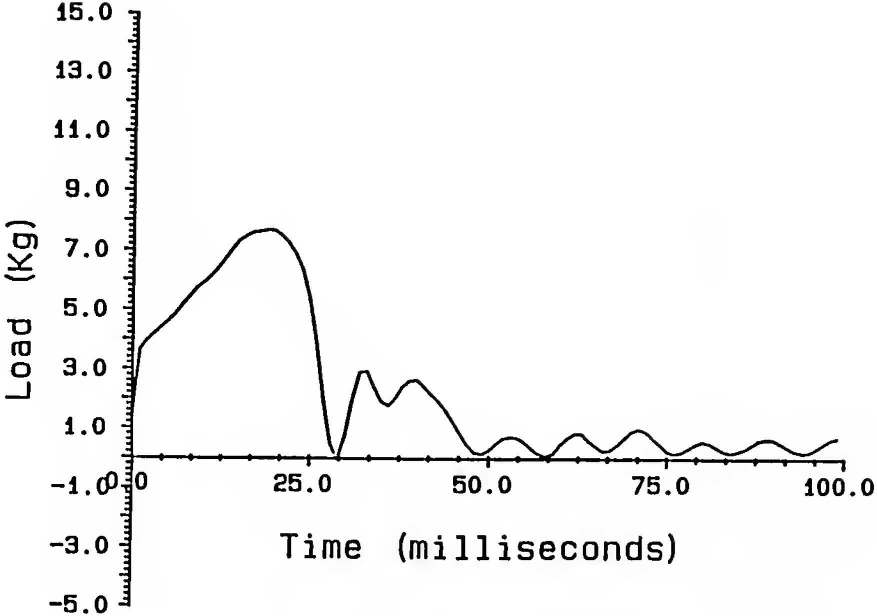


Figure 9. Representative Soft Wheat Breakage Event

point, the kernel is sliced and the load cell rebounds until it reaches its initial resting position. The 5H pencil lead differs from the Crayon pencil lead (shown in Figure 11) in that the Crayon's force increases rapidly at the beginning of the event and then slowly increases to the ultimate strength and then the force rapidly decays. The 5H and Red Crayon pencil leads possessed the same approximate "crushing diameter" as the wheat kernels. Although the breakage events from the Pencil and Crayon leads did not exactly duplicate the hard and soft wheats breakage events, they showed similar characteristics.

To verify the force reading obtained from the SKWHT using the calibration material, an Instron, Model A 1026G was used. A special bracket was designed to support each of the same three blades (sharp, curve, and blunt) which were also used in the SKWHT. The fastest crosshead setting ($25 \frac{cm}{min}$) on the Instron was used in all of the testing. A 2kg load cell was used for the calibration of the Crayon pencil lead, and a 50kg load cell was used for the calibration of the 5H pencil lead.

In comparing the results from the SKWHT and the Instron, the blades are the same in each case, however, the geometries at which the blades approach the kernels vary widely. In the scenario of the SKWHT, the blade is stationary, and the kernel is driven into the blade. In the case of the Instron, the kernel is at rest and the blade is in motion. The two methods also differ in the geometry of the actual cut. The SKWHT revolves and the blade also revolves, thus providing complicated arcs of the kernel impinging against the blade and the back of the hole.

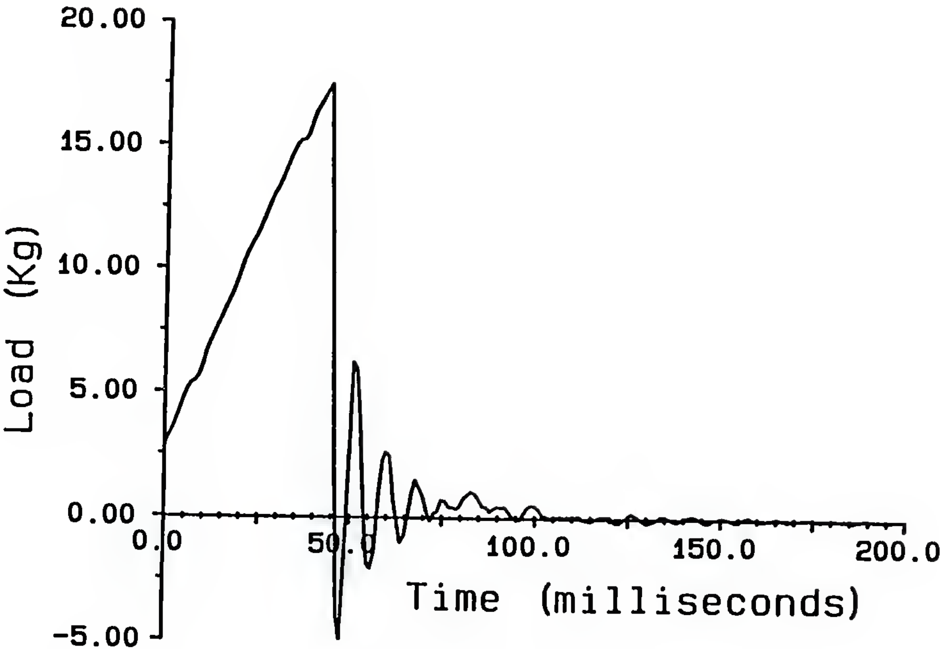


Figure 10. Typical 5H Breakage Event

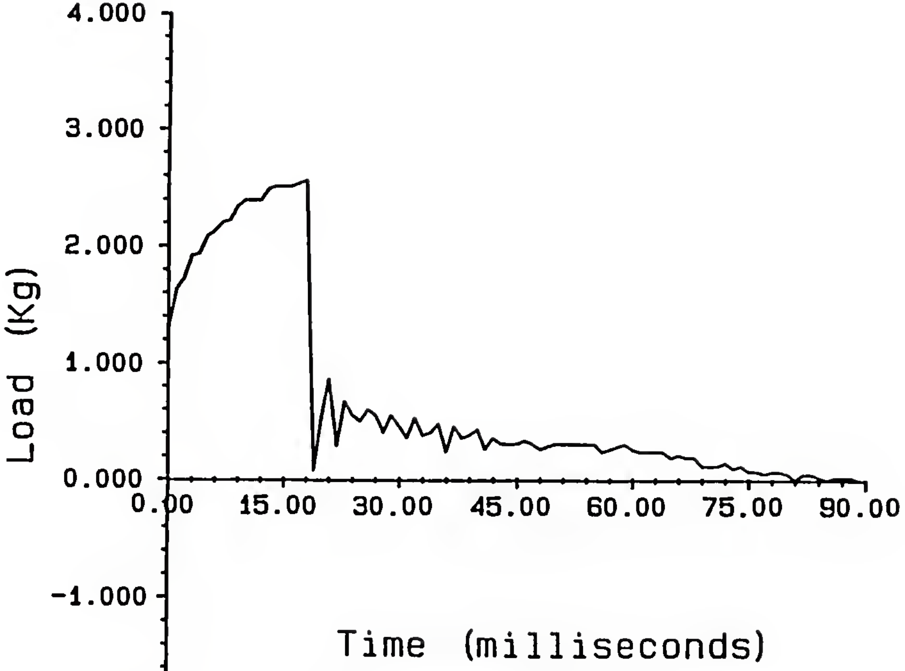


Figure 11. Typical Crayon Breakage Event

In the Instron, all the action transpires in a linear plane of the blade. Also, since the crosshead speed of the Instron was only 25 $\frac{\text{cm}}{\text{min}}$, the data from the SKWHT is taken at the lowest speed setting of 0.628 rad/s to compare the results.

Quantitative Error Source Analysis

One of the sources of error in the SKWHT is the actual rad/s of the rotating plate. The rad/s of the SKWHT was set by a rheostat on the variable speed direct drive motor. The actual rad/s values were determined by timing the plate and the values were scribed on the rheostat. The rad/s value could vary by about 0.0802 rad/s. The next source of error is the blade clearance which is set by feeler gages. The blade clearance was measured by placing the center line of the hole with the cutting blade and utilizing the feeler gage to set the gap. The feeler gages are accurate to ±0.00254cm (0.001"). Another unquantifiable inaccuracy in this experiment is any material that may be accumulated in the recesses of the hole. The Red Crayon Pencil lead tended to leave a waxy substance in the groove behind the blade. Although the scraper removes most of the substance in this groove, there may be instances where the material would be left in the groove. If there is any visually detectable material, it is removed, but there may be residual microscopic material which may affect the results. Another source of error is the sharpness of the blade. Since there is no easy technique to measure sharpness, it was assumed that the blade did not degrade during the course of this investigation. Also, the sampling was taken at 5 kHz, and the actual maximum force may have been missed by the data acquisition equipment. The diameters of the materials being sliced are also another source of error. The height of the blade in the groove is also another source of error. The height of each blade was set to clear the groove in the plate, and the placement may have differed from test to test. These are the main errors contributing to the overall error of the experiment.

Experimental Design

A response surface experimental design with 13 different observation points was used containing five levels of plate speed 0.628, 0.754, 1.047, 1.340, and 1.466 rad/s, and five blade clearances 1.175, 1.124, 0.921, 0.743, and 0.667 mm. Three observations were taken at the center operating condition 1.047 rad/s and 0.921mm. Two observations were taken at (0.754rad/s, 1.124mm), (0.754 rad/s, 0.743 mm), (1.340 rad/s, 1.124 mm), and (1.340 rad/s, 1.124 mm), and one observation at all other operating conditions. At each observation point, ten different breakage events were recorded and the maximum force averaged together. The observation points were randomly selected for each new blade or material type.

Since the peak (maximum) force was discovered to yield the best delineation between hard and soft wheat, this was used for the experimental design. The peak force was used in both the pencil lead analysis, and the wheat analysis. The peak force for each individual pencil lead was determined by the computer analysis, and then the maximum force reading for the 10 breakage events. In the case of multiple observations at a particular rad/s and blade clearance, an average of all the data points collected was used.

The wheat was analyzed in a similar fashion as the pencil lead. The operating conditions remained the same, and the sharp blade was used. The other two blades (curve, and blunt) were operated only at the optimum condition as prescribed by the pencil lead analysis. The wheat genotypes chosen for this experiment were Mustang and Daws. The two varieties were chosen on the basis of availability; the hard wheat chosen was Mustang; and the soft wheat chosen was Daws. The effect of moisture content on wheat hardness measurement was evaluated on the two varieties by tempering to three different moisture levels (9, 10, and 14% m.c.). All references made to moisture content (m.c.) are made on a wet basis measurement. The wheat was tempered by placing small amounts of the wheat kernels in porous bags inside relative humidity chambers. Salt solutions were used to create the relative humidities for each different moisture level. All of the samples were allowed to equilibrate with the proper moisture content over a period of two weeks. At the end of the two weeks, the samples were tested for moisture by the oven method. Once the samples were identified for the proper moisture content, they were then immediately tested on the SKWHT.

Procedure for Analysis

The data was collected on the IBM PC Compatible computer and then transferred to either the Departmental MicroVaxII or PDP 11/34 for further analysis. The data was analyzed for maximum force using the program in Appendix C. Once all of the maximum force values had been calculated, then dm (Data Manipulator, Public Domain Program on the MicroVaxII) was used to extract the columnar information and to also subtract the maximum force of the soft wheat or soft pencil lead from the appropriate hard wheat kernel or hard pencil lead. These difference values were then transferred onto the campus main frame computer (IBM 370) to run SAS (Statistical Analysis System, SAS Institute, Inc.). A response surface regression for each of the blades and each pair of materials was performed. The response surface regression determined the best equation for modeling the difference in force as a function of angular velocity and blade clearance. The data generated by the equations were then plotted to view the results in a more meaningful manner. The response surface regression also predicted the optimum operating condition for each of the different blades and material types, which was used from the pencil leads and applied to the wheat analysis.

RESULTS AND DISCUSSION

Comparison of Instron Results to Hardness Tester Results

Table 1 in the Appendix were the results of testing 5H and Crayon Pencil lead in the Instron. The mean and standard deviation were also shown in the table. Tables 2 through 5 in the Appendix were the results of averaging ten pencil leads' maximum force with all three blades and at the 13 different operating conditions. Tables 2 through 5 also showed the standard deviation as well as coefficient of variation for each of the ten pencil leads. The table shown below is a quick reference in comparing the Instron peak force to the SKWHT. The identification (ID) of the sample being processed is given in the first column by blade and lead types, followed by the average results for both the Instron ($\bar{x}_1$) and SKWHT ($\bar{x}_2$).

ID	$\bar{x}_1$	$\bar{x}_2$
S,Cr	1.88	2.00
S,5H	3.11	19.16
C,Cr	2.23	2.87
C,5H	2.77	18.56
B,Cr	2.42	2.69
B,5H	3.46	19.57

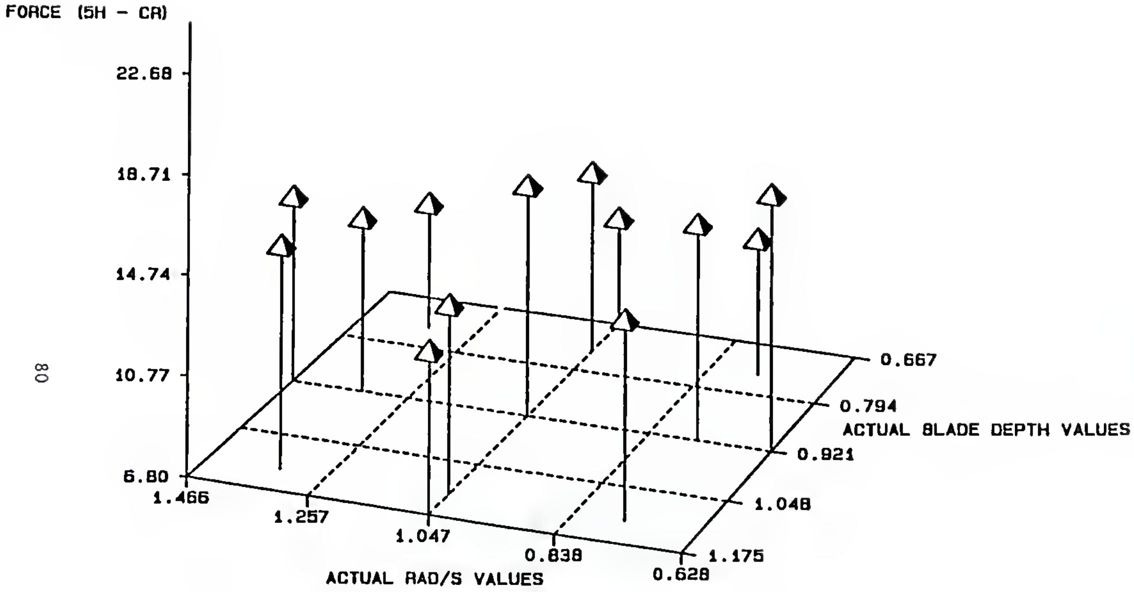
where
S = Sharp blade
C = Curve blade
B = Blunt blade
Cr = Crayon pencil lead
5H = 5H pencil lead
$\bar{x}_1$ = The mean of the particular sample from the Instron (kg).
$\bar{x}_2$ = The mean of a particular sample from the SKWHT (kg).

The results by the SKWHT are similar to those by the Instron. The Crayon pencil lead results were on the same order of magnitude for both the Instron and the SKWHT. The 5H lead showed several orders of magnitude difference between the two units. Clearly, the cutting geometries as well as the angular velocity influenced the results. A main source of error in the breakage events monitored by the Instron was the range of the load cell mounted on the Instron. The only two available load cells for the Instron during this test were the 2kg load cell, and the 50kg load cell. The peak force required to slice the 5H pencil lead was 3kg, and thus the 50kg load cell needed to be used. The limited range of 3kg out of the 50kg did not yield a very large range. The limited range of only 3kg on the 50kg load cell could cause some error in the measurement of the 5H leads. Although the Crayon Pencil leads did not exactly lie in the range of the device, the forces were on the same order of magnitude.

Discrimination of 5H pencil lead from Crayon Leads

The standard deviation as well as the coefficient of variation of the numbers were recorded. The device was set up with the sharp blade in place throughout the data collection for the first graph (Figure 12). Figure 12 is a graph of the maximum breaking force of 5H pencil lead minus the maximum breaking force of Crayon pencil lead at each of the operating conditions. Operating conditions where multiple samples were taken, the average of the maximum force readings was used and graphed. Each one of the pyramids on the graph depicts the difference of the maximum force between the 5H pencil lead and the Crayon lead. Figure 13 is a graph of modeled force difference between 5H pencil lead and Crayon lead as generated by Response Surface Regression. Figure 14 is a graph of the maximum force difference between 5H pencil lead and Crayon lead using the curve blade. This graph is similar to the first except that the curve blade is used in the data collection instead of the sharp blade. Figure 15 is similar to the second graph except the curve blade was used. Figure 16 shows the maximum breaking force between 5H and Crayon using the blunt blade. The last of the figures for pencil lead is the graph of the modeled maximum breaking force for 5H and Crayon lead using the blunt blade (Figure 17).

The R-values from the response surface regression were 0.4516, 0.3097, and 0.2731 for the sharp, blunt and curve blade, respectively. The R-values were not very high due to the variability of the peak force of the pencil leads. Results from the SAS response surface analysis



FORCE (5H - CR): REFERS TO DIFFERENCE IN FORCE

Figure 12. Maximum Breaking Force Difference (Sharp, Lead)

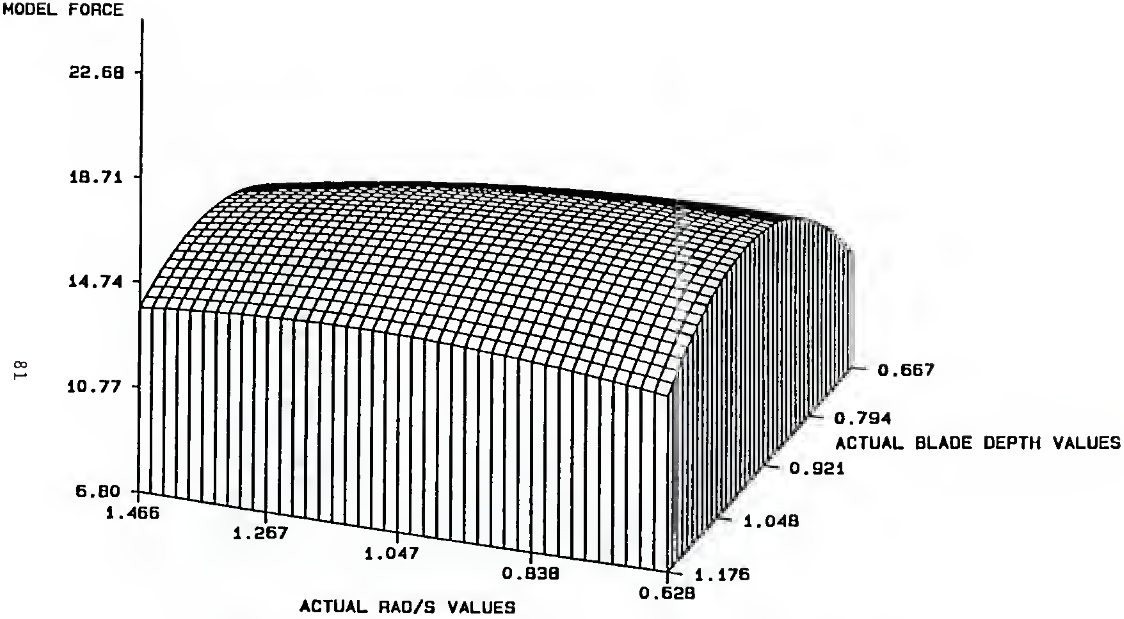
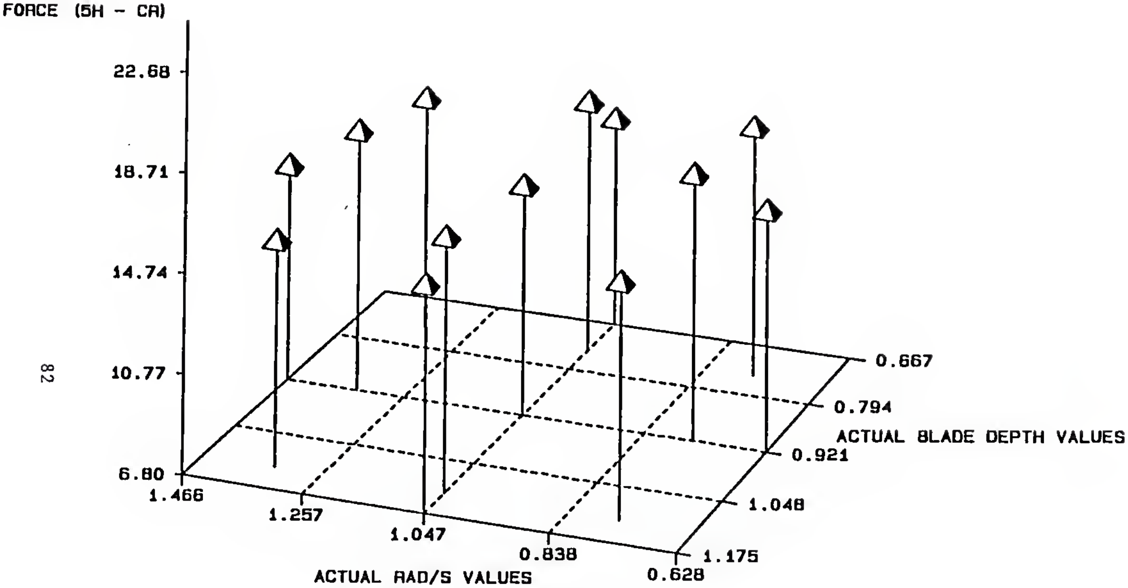


Figure 13. Model Maximum Breaking Force Difference (Sharp, Lead)



FORCE (5H-CR) REFERS TO DIFFERENCE IN FORCE.

Figure 14. Maximum Breaking Force Difference (Curve, Lead)

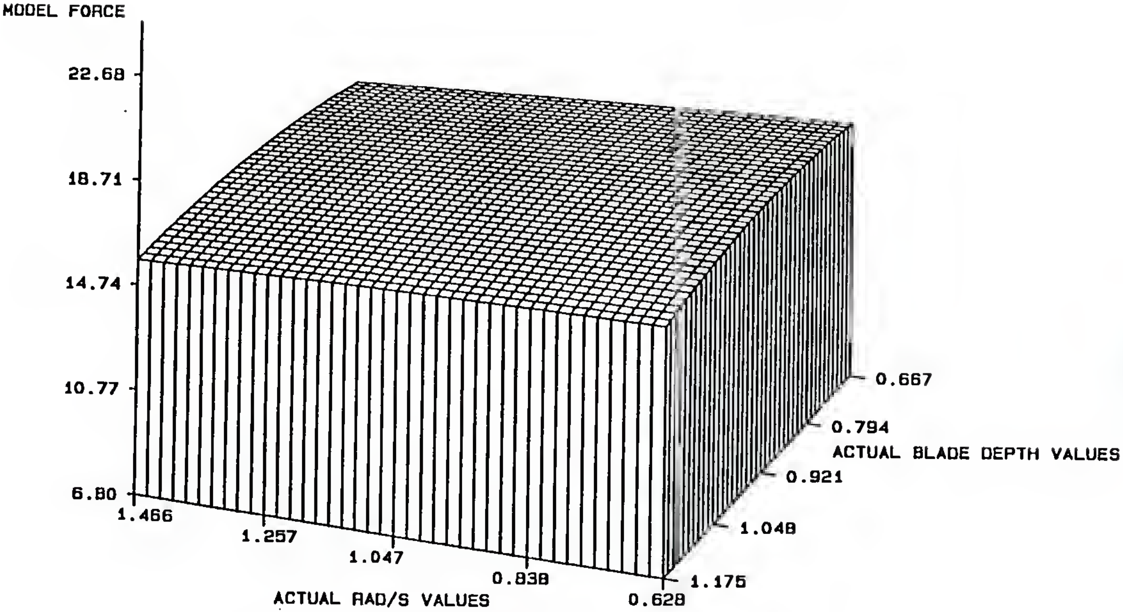
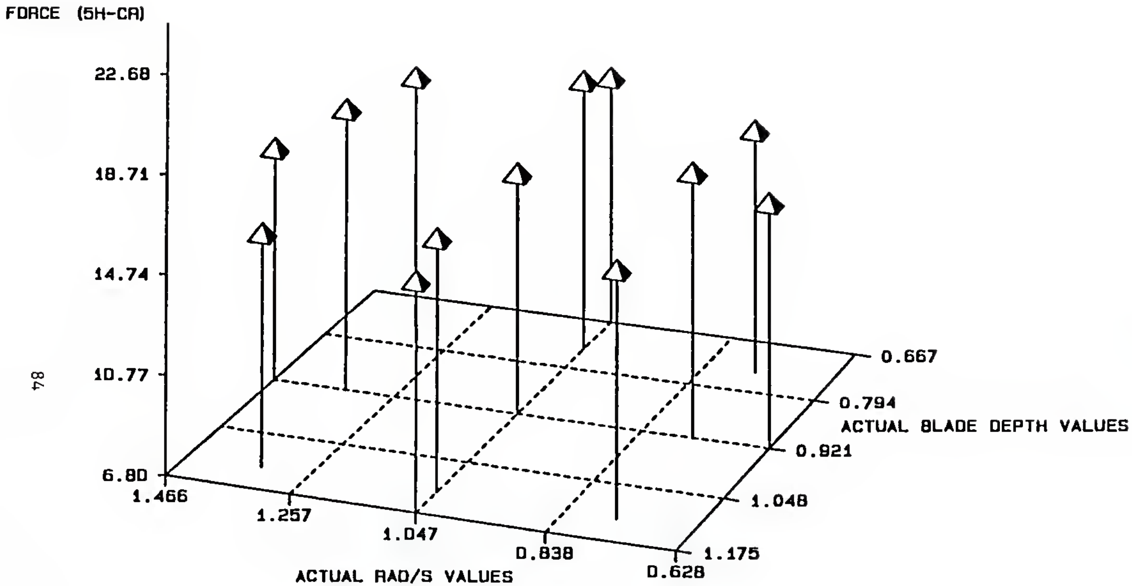


Figure 15. Model Maximum Breaking Force Difference (Curve, Lead)



FORCE (5H-CR): REFERS TO DIFFERENCE IN FORCE.

Figure 16. Maximum Breaking Force Difference (Blunt, Lead)

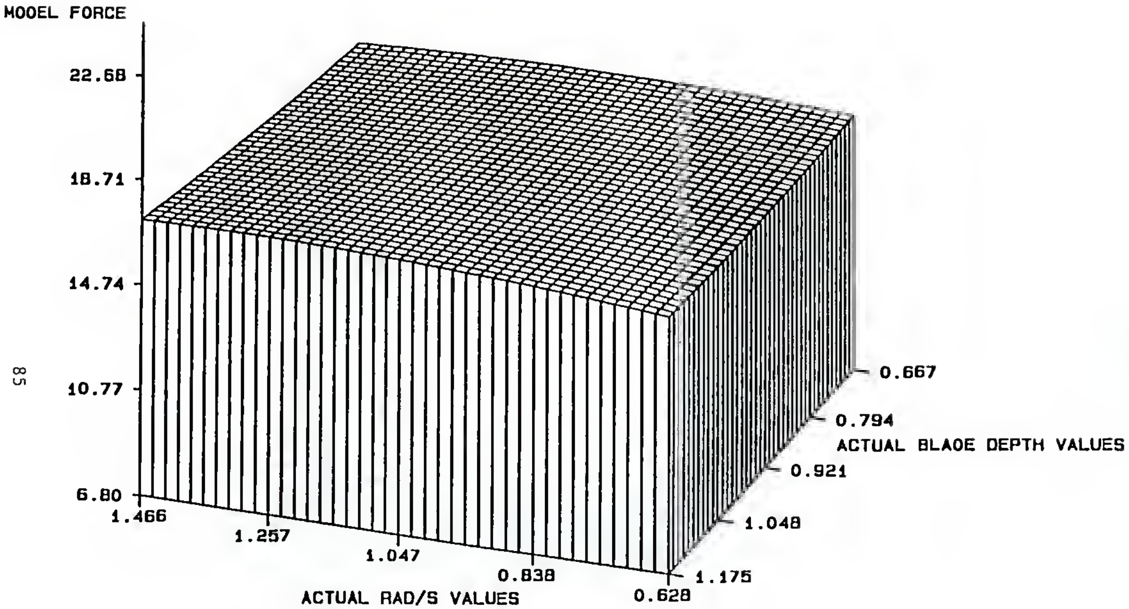


Figure 17. Model Maximum Breaking Force Difference (Blunt, Lead)

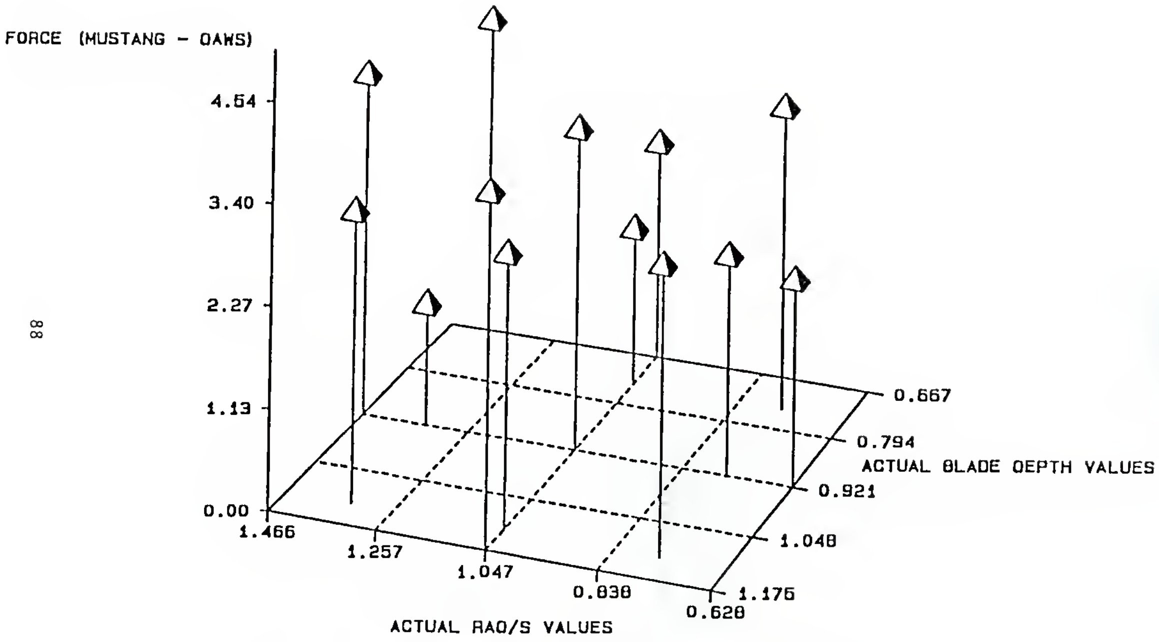
showed that the blunt blade resolved the difference between the modeled peak force for 5H and Crayon pencil leads. Also, the slower angular velocities tended to yield greater differences in the modeled peak force. The SAS response surface analysis also showed that the blade clearance did not have an appreciable effect on the peak breaking force difference on pencil leads.

The optimum operating condition for the blunt blade as predicted by SAS was 1.354 rad/s, and a blade clearance of 0.921 mm which yielded a force of 17.150 kg. The sharp blade's optimum operating condition as calculated by SAS was 0.954 rad/s, with the blade clearance set at 0.989 mm which would produce a reading of 11.445 kg. The optimum operating condition for the curve blade did not result in a practical rad/s, which was set at -0.170 rad/s, with a blade clearance of 0.743 mm which would produce the hypothetical force of 17.290 kg, however the negative rad/s was just not a practical rad/s and therefore was disregarded. Although the blunt blade had a larger difference than the sharp blade, the sharp blade was chosen as the blade to conduct the remainder of the wheat tests using the 13 different operating conditions with multiple observations at five of the operating conditions. The sharp blade was chosen due to preliminary tests conducted with this blade which yielded better crushing results.

Discrimination of Hard from Soft Wheat

Tables 6 through 13 in the Appendix showed the results of averaging ten hard or soft wheat kernels by the SKWHT, utilizing the sharp blade at three different moisture contents. The sharp blade was chosen for the investigation of the peak force difference between hard and soft wheat. The same experimental design for the pencil leads was applied to the wheat as well. Again, the SAS Response Surface analysis was also used to analyze the wheat in the same fashion as the pencil leads. The varieties of Mustang and Daws were used in this investigation. The effect of moisture on the wheat was also studied to view its action on the difference in force. Three different moisture contents were used in this investigation: 9%, 10%, and 14% measured on a wet basis.

Each of the various operating conditions for the wheat was graphed as the difference in maximum breaking force as a function of both rad/s and blade clearance. The graphs alternate between the actual difference between Mustang and Daws maximum breaking force, and the Modeled force between Mustang and Daws. Figure 18 shows the actual difference of the maximum breaking force of the Mustang wheat at 9% m.c. minus the maximum breaking force of Daws at 9% m.c. at each of the operating conditions. The SAS modeled maximum force difference between Mustang and Daws at 9% m.c. is shown in Figure 19. Figure 20 is a graph of the actual maximum breaking force between Mustang and Daws at 10% m.c. Figure 21 is a graph of the modeled maximum breaking force between Mustang and Daws at 10% m.c. Figure 22 and Figure 23 represent the actual maximum and the modeled Maximum breaking force for Mustang and Daws at 14% m.c.,



FORCE (MUST - DAWS): REFERS TO DIFFERENCE IN FORCE

Figure 18. Maximum Breaking Force Difference (9% m.c., Wheat)

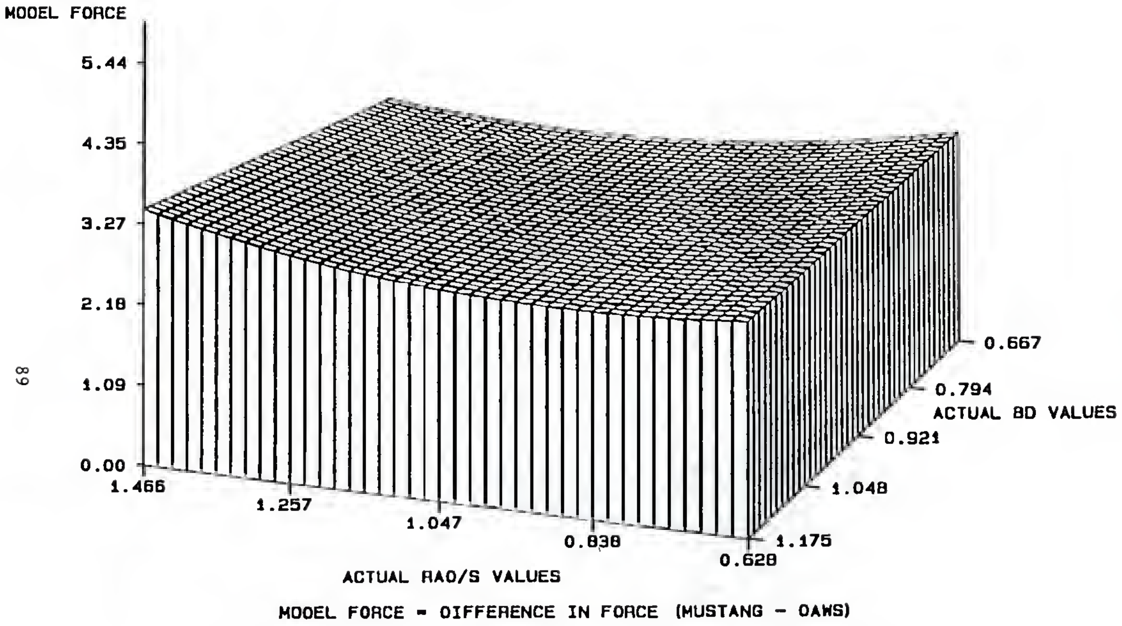
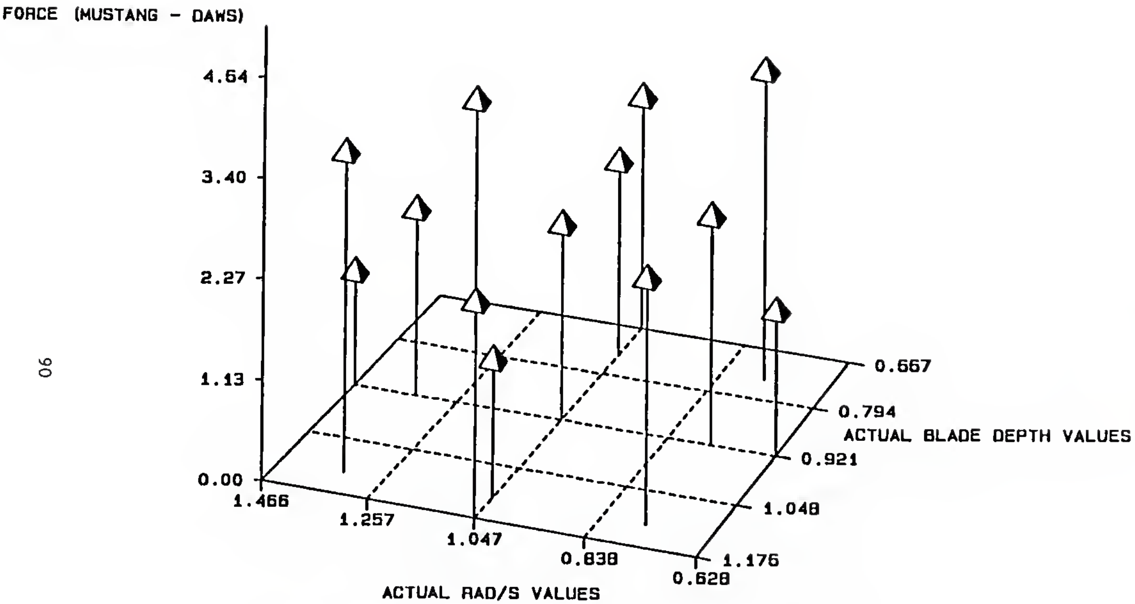


Figure 19. Model Maximum Breaking Force Difference (9% m.c., Wheat)



FORCE (MUST - OAWS): REFERS TO DIFFERENCE IN FORCE

Figure 20. Maximum Breaking Force Difference (10% m.c., Wheat)

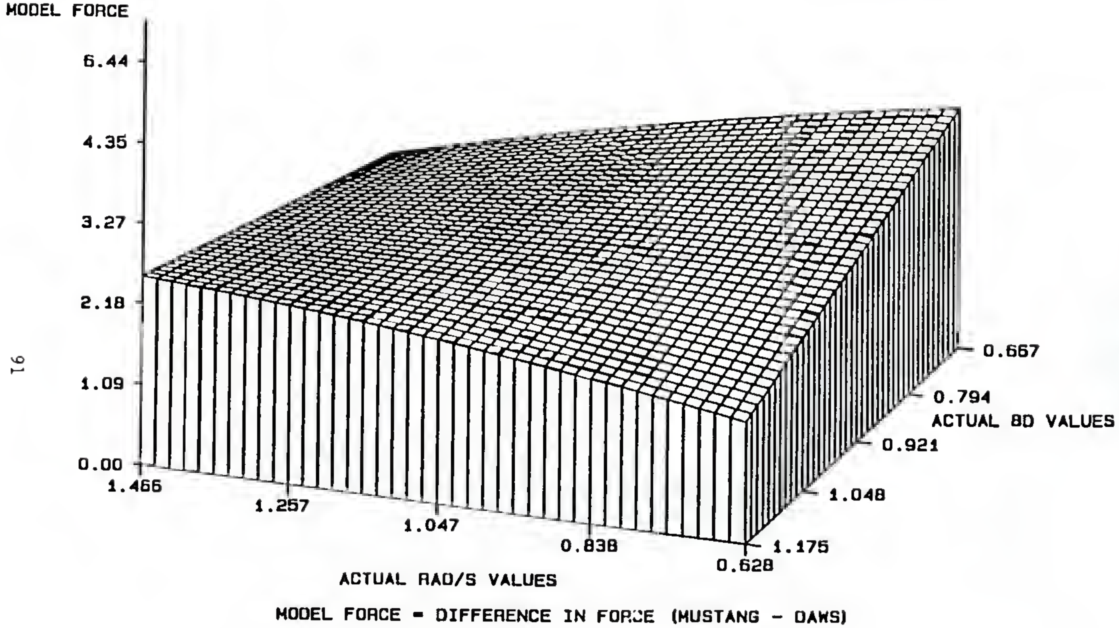
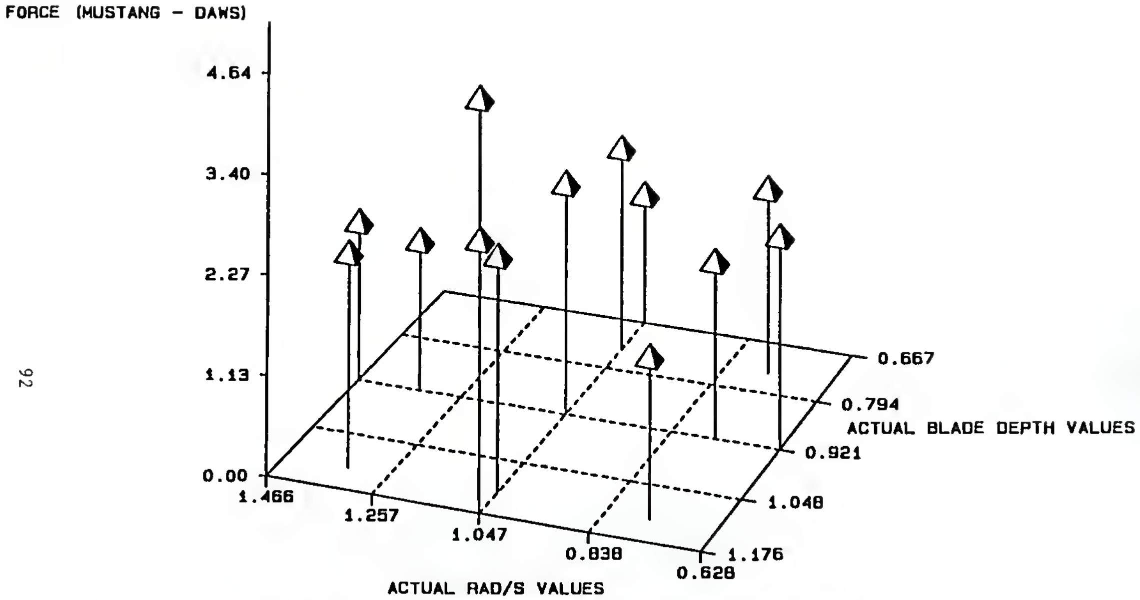


Figure 21. Model Breaking Force Difference (10% m.c., Wheat)



FORCE (MUST - DAWS): REFERS TO DIFFERENCE IN FORCE

Figure 22. Maximum Breaking Force Difference (14% m.c., Wheat)

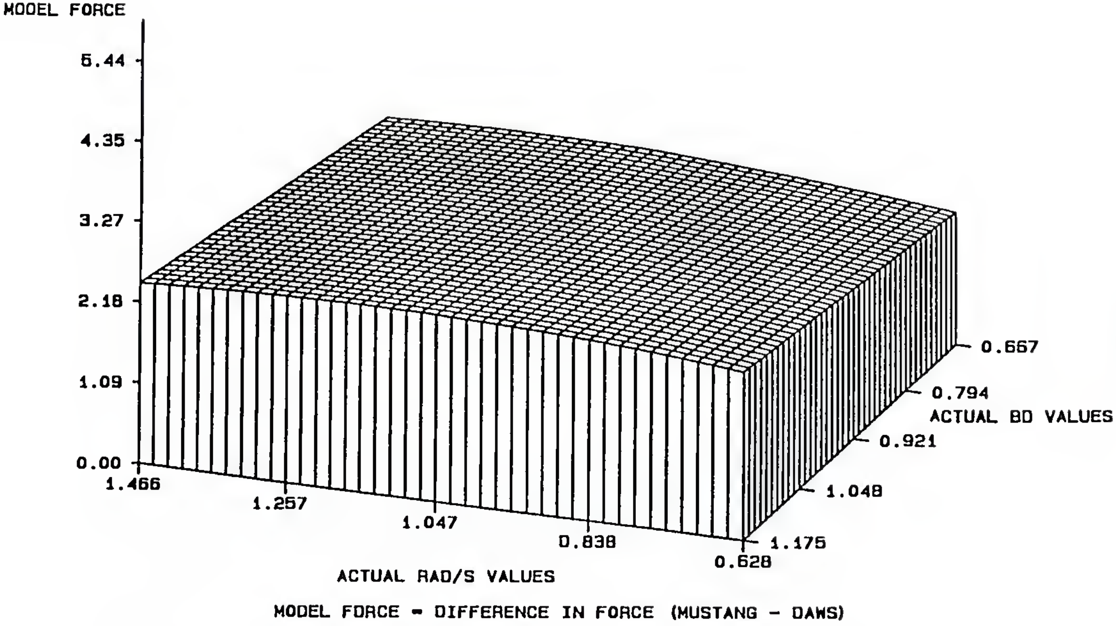


Figure 23. Model Maximum Breaking Force Difference (14% m.c., Wheat)

respectively.

The SAS Response Surface Analysis for the 9% m.c. modeled maximum breaking force showed a difference of 2.770 kg at an angular velocity of 0.740 rad/s, and a blade clearance value of 1.378 mm. For the 10% m.c. modeled maximum breaking force for the wheat, a difference of 2.372 kg at 1.241 rad/s, and 1.023 mm blade clearance was observed. The optimum difference at 14% m.c. was 2.308 kg at 1.635 rad/s, and a blade clearance setting of -0.134 mm. It should be noted that as the moisture of the wheat increased, the angular velocity of the device needed to be increased in order to achieve the optimum difference. The blade clearance also increased as the moisture content increased and thus the blade needed to be adjusted closer to the back of the vertical hole in order to obtain a larger difference. The optimum difference also decreased as the moisture content increased. The optimum force decreased by 0.462 kg by an increase in moisture content of 5%. The SAS analysis revealed that an increase in moisture content, the rad/s must be increased and the blade clearance decreased, in order to achieve an optimum difference between hard and soft wheat. The difference between Mustang and Daws was accentuated by increasing the angular velocity and decreasing the blade clearance, as the moisture content of the kernels increased.

CONCLUSIONS

The results of analysis from SAS for the pencil leads were as follows: 1) the optimum operating condition for the sharp blade was 0.954 rad/s at a blade clearance of 0.974 mm which yielded a difference in forces between 5H and Crayon pencil lead of 16.184 kg; 2) the optimum operating condition for the blunt blade was 1.354 rad/s at a blade clearance of 1.075 mm which gave a difference of 17.150 kg; and 3) the optimum operating condition for the curve blade was -0.170 rad/s at a blade clearance of 0.760 mm which was supposed to yield a difference of 17.290 kg. The blunt blade had the highest difference out of all the blades excluding the curve blade which depicted a negative angular velocity for the optimum operating condition. The response surface analysis was an attempt to determine the optimum angular velocity and blade clearance settings to maximize the differences in force. The surfaces, when plotted, did not have any extreme peaks and the surface was fairly level. The conclusion from the pencil lead was that the blunt blade was the best blade, and the SKWHT should be operated at an angular velocity of 0.954 rad/s with a blade clearance setting of 0.974 mm.

The optimum settings for the wheat analysis were as follows: 1) for the 9% m.c. wheat , 0.740 rad/s with a blade clearance of 1.365 mm in order to obtain a difference in force of 2.768 kg; 2) for the 10% m.c. wheat, 1.241 rad/s with a blade clearance of 1.022 mm to yield a difference of 2.417 kg; and 3) for the 14% m.c. wheat, 1.635 rad/s with a blade clearance of -0.138m in order to yield a difference of 2.308 kg.

Again, the variability of the wheat was such that the R-values from the SAS models were very low 0.0424, 0.1996, and 0.0939 for 9% m.c., 10% m.c., and 14% m.c., respectively. These R-values were low due to the flatness of the response surface which indicated no preferred angular velocity or blade clearance which yielded a sharp peak in the maximum force readings. The analysis showed that in order to obtain a maximum difference between Daws and Mustang, the angular velocity needed to be increased as well as the blade clearance. The analysis of the surfaces showed that as the moisture content of the wheat increased, the delineation of the two decreased.

Overall, for the best discrimination of the pencil leads, the settings on the SKWHT should be 1.354 rad/s, with a blade clearance of 1.073 mm using the blunt blade. The optimum settings for the wheat, depending upon moisture content, were listed above. The peak force difference decreased as the moisture content increased, and in order to obtain the maximum difference, the angular velocity as well as the blade clearance should be increased.

RECOMMENDATIONS FOR FUTURE RESEARCH

The results from the Pencil lead data indicated that the blade which produced the maximum difference between the 5H and Crayon pencil lead was the blunt blade. In preliminary studies, it was concluded that the sharp blade yielded better results than the other two blades. The sharp blade was used in the data collection of the wheat due to the preliminary study concluding that this was the best blade. In future studies, the blunt blade should be used for maximizing the difference in hard versus soft materials. Another aspect of the data collection phase which needs improvement is the acquisition of a load cell for the Instron in the 10kg range. Only the lower portion of the 50kg load cell was used in the testing of the 5H pencil lead and a 10kg load cell would yield a better range. To prove that the blunt blade is best blade for the maximum difference for wheat, all three different blades should be analyzed using the response surface analysis for all 13 different operating conditions. The wheat kernels should have been verified on the Instron for the three different blades. Also, the time interval for all of the data collected was on an irregular basis, and for further timing analysis, a regular sampling rate should be taken.

So, the investigation should be carried out to perform a surface analysis response on the blunt blade for the three different moisture contents of wheat using the 13 different operating conditions.

REFERENCES

1. Anderson, R. A., Pfeifer, V. F., and Peplinski, A. J. Measuring Wheat Kernel Hardness By Standardized Grinding Procedures. Cereal Sci. Today. 2:204-209, 1966.

2. Baker, R. J. Inheritance of Kernel Hardness in Spring Wheat. Crop Sci. 17(6):960-962, 1977.

3. Barlow, K. K., Buttrose, M. S., Simmonds, D. H., and Vesk, M. The Nature of the Starch-Protein Interface in Wheat Endosperm. Cereal Chem. 50:443-454, 1973.

4. Barlow, K. K., and Simmonds, D. H. Proc. R. Aust. Chem. Inst. Cereal Chem. Division. 22nd Annual Conference, 1972.

5. Barry, P. J., Eckhoff, S. R., and Davis, A. B. Analysis of an Individual Breakage Tester. ASAE Paper No. MCR 87-118, American Society of Agricultural Engineers, St. Joseph, MO., 1987.

6. Beard, B. H., and Poehlman, J. M. A Study of Quality, as Measured by the Pearling Test, in Crosses between Hard and Soft Wheats. Agron. J. 46(1):220-223, 1954.

7. Biffen, R. H. On The Inheritance Of Strength In Wheat. J. Agric. Sci. 3:86-101, 1908.

8. Chung, C. J., Clark, S. J., Lindholm, J. C., McGinty, R. J., and Watson, C. A. The Pearlograph Technique for Measuring Wheat

Hardness. Trans. ASAE. 18(1):185-193, 1975.

9. Chesterfield, R. S. A Modified Barley Pearler for Measuring Hardness of Australian Wheats. J. Aust. Inst. Agric. Sci. 37(1):148-151, 1971.

10. Eckhoff, S. R., Supak W. A., and Davis, A. B. A Rapid Single-Kernel Wheat Hardness Tester. Accepted for Publication in Cereal Chem. 1988.

11. Creenway, W. T. A Wheat Hardnesss Index. Cereal Sci. Today. 14(2):4-7, 1969.

12. Hoseney, R. C., and Seib, P. A., Structural differences in Hard and Soft Wheat. Baker's Dig. 47(6):26-28, 1973.

13. Katz, R., Cardwell, A. B., Collins, N. D., and Hostetter, A. E. A New Grain Hardness Tester. Cereal Chem. 36(5):393-401, 1959.

14. Katz, R., Collins, N. D., and Cardwell, A. B. Hardness and Moisture Content of Wheat Kernels. Cereal Chem. 38:364-368, 1961.

15. Kilborn, R. H., Black, H. C., Dexter, J. E., and Martin, D. G. Energy Consumption During Flour Milling: Description of Two Measuring Systems and the Influence of Wheat Hardness on Energy Requirements. Cereal Chem. 59(4):284-288, 1982.

16. Kosmolak, F. C. Crinding Time - A Screening Test For Kernel Hardness in Wheat. Can. J. Plant Sci. Rev Can Phytotechnie 58:415-

420, 1978.

17. Kuhlman, D. K., Chung, D. S., McGinty, R., and Watson, C. A. Modification of the Pearler for Wheat-Hardness Tests. Trans. ASAE. 22(4):881-885, 1979.

18. Lai, F. S., Rousser, R., Brabec, D., and Pomeranz, Y. Determination of Hardness in Wheat Mixtures. II. Apparatus for Automated Measurement of Hardness of Single Kernels. Cereal Chem. 62(3):178-184, 1985.

19. Lookhart, C., L, Lai, F. S., Pomeranz, Y. Variability in Gliadin Electrophoregrams and Hardness of Individual Wheat Kernels Selected from Foundation Seed on the Basis of Grain Morphology. Cereal Chem. 62(3):185-190, 1985.

20. McCluggage, M. E. Factors Influencing The Pearling Test For Kernel Hardness in Wheat. Cereal Chem. 20:686-700, 1943.

21. Miller, B. S., Afework, S., Pomeranz, Y., Bruinsma, B. L., and Booth G. D. Measuring the Hardness of Wheat. Cereal Foods World. 27(2):61-64, 1982.

22. Miller, B. S., Hughes, J. W., Afework, S., and Pomeranz, Y. A Method to Determine Hardness and Work of Grinding Wheat. J. Food Sci. 46:1851-1855, 1981.

23. Miller, B. S., Pomeranz, Y., and Afework, S. Hardness (Texture) of Hard Red Winter Wheat Crown in a Soft Wheat Area and of Soft

Red Winter Wheat Grown in a Hard Wheat Area. Cereal Chem. 61(2):201-203, 1984.

24. Moss, H. J. Factors Determining the Optimum Hardness of Wheat. Aust. J. Agric. Res. 29:1117-1126, 1978.

25. Newton, R., Cook, W. H., and Malloch, J. G. The Hardness of the Wheat Kernel in Relation to Protein Content. Sci. Agric. 8:205-219, 1927.

26. Obuchowski, W., and Bushuk, W. Wheat Hardness: Effects of Debranning and Protein Content. Cereal Chem. 57(6):426-428, 1980.

27. Pomeranz, Y., Afework, S. The Effects of Kernel Size in Plump and Shrunken Kernels, and of Sprouting, on Kernel Hardness in Wheat. J. Cereal Sci. 2(2):119-126, 1984.

28. Pomeranz, Y., Afework, S., and Lai, F. S. Determination of Hardness in Mixtures of Hard Red Winter and Soft Red Winter Wheats. I. Bulk Samples. Cereal Chem. 62(1):41-46, 1985.

29. Pomeranz, Y., Peterson, C. J., and Mattern, P. J. Hardness of Winter Wheats Grown Under Widely Different Climatic Conditions. Cereal Chem. 62(6):463-467, 1985.

30. Roberts, H. F., A Quantitative Method for the Determination of Hardness in Wheat. Kns. Agric. Exp. St. Bull. 167. 371-390, 1910.

31. Simmonds, D. H. Chemical Basis of Hardness and Vitreosity in the Wheat Kernel. Baker's Dig. 48(5):16-29, 1974.

32. Simmonds, D. H., Barlow, K. K., and Wrigley, C. W. The Biochemical Basis of Grain Hardness in Wheat. Cereal Chem. 50:553-562, 1973.

33. Stenvert, N. L. The Measurement of Wheat Hardness and its effect on milling characteristics. Aust. J. Exp. Agric. Anim. Husb. 12(54):159-164, 1972.

34. Stenvert, N. L., Kinsgwood, K. The Influence of the Physical Structure of the Protein Matrix on Wheat Hardness. J. Sci. Food Agric. 28:11-19, 1977.

35. Symes, K. J. Influence of a Gene Causing Hardness on the Milling and Baking Quality of Two Wheats. Aust. J. Agric. Res. 20:971-979, 1969.

36. Symes, K. J. The Inheritance of Grain Hardness in Wheat as Measured by the Particle Size Index. Aust. J. Agric. Res. 16:113-123, 1965.

37. Taylor, J. W., Bayles, B. B., and Fifield, C. G. A Simple Measure of Kernel Hardness in Wheat. J. Am. Soc. Agron. 30:775-784, 1939.

38. Williams, P. C. Relation of Starch Damage and Related Characteristics to Kernel Hardness in Australian Wheat Varieties.

Cereal Chem. 44:383-391, 1967.

39. Williams, P. C. Screening Wheat for Protein and Hardness by Near Infrared Reflectance Spectroscopy. Cereal Chem. 56(3):169-172, 1979.

40. Williams, P. C., and Sobering, D. C. Attempts at Standardization of Hardness Testing of Wheat. I. The Grinding/Sieving (Particle Size Index) Method. Cereal Foods World 31(5):359-364, 1986.

41. Williams, P. C., and Sobering, D. C. Attempts at Standardization of Hardness Testing of Wheat. II. The Near-Infrared Reflectance Method. Cereal Foods World. 31(6):417-420, 1986.

42. Wrigley, C. W. Cereal Sci. Today. 17:370, 1972.

43. Yamazaki, W. T., and Donelson, J. R. Kernel Hardness of Some U.S. Wheats. Ceral Chem. 60(5);344-350, 1983.

44. Zeidler, W., and Taylor, G. H. Embedding coals and cokes in plastic for microscopic examination. J. Roy. Microscop. Soc. 80:287 1962.

APPENDIX A: ACTUAL SKWHT AND INSTRON DATA VALUES

The first table generated by the analysis program was the Instron Results.

TABLE 1. Instron Pencil Lead Data
Model A1026G Instron. (Crosshead Speed: 25 cm/min, Chart Speed: 50 cm/min)

Sharp Blade 2kg Load Cell			
Lead	Range	Load (g)	Load (Lb)
Crayon	R5	853.33	1.88
"	"	826.67	1.82
"	"	753.33	1.66
"	"	820.00	1.81
"	"	1046.67	2.31
"	"	940.00	2.07
"	"	953.33	2.10
"	"	1000.00	2.20
"	"	820.00	1.81
"	"	766.67	1.69
"	"	780.00	1.72
"	"	840.00	1.85
"	"	793.33	1.75
"	"	773.33	1.70
"	"	846.67	1.87
Mean:		854.22	1.88
Std. Dev.		89.62	0.20

Sharp Blade 50kg Load Cell			
Lead	Range	Load (g)	Load (Lb)
5H	R5	1000.00	2.20
"	"	950.00	2.09
"	"	1316.67	2.90

Sharp Blade
50kg Load Cell

Lead	Range	Load (g)	Load (Lb)
"	"	1316.67	2.90
"	"	2116.67	4.67
"	"	1833.33	4.04
"	"	1050.00	2.31
"	"	1016.67	2.24
"	"	1300.00	2.86
"	"	2200.00	4.85
Mean:		1410.00	3.11
Std. Dev.		470.96	1.04

Curved Radius
2kg Load Cell

Lead	Range	Load (g)	Load (Lb)
Crayon	R20	886.67	1.95
"	"	826.67	1.82
"	"	1046.67	2.31
"	"	993.33	2.19
"	"	1066.67	2.35
"	"	1053.33	2.32
"	"	1066.67	2.35
"	"	880.00	1.94
"	"	1166.67	2.57
"	"	1106.67	2.44
Mean:		1009.34	2.23
Std. Dev.		110.32	0.24

Curved Radius
50kg Load Cell

Lead	Range	Load (g)	Load (Lb)
5H	R5	1200.00	2.65
"	"	1500.00	3.37
"	"	1383.33	3.05
"	"	1150.00	2.53
"	"	1066.67	2.35
"	"	1366.67	3.01
"	"	1366.67	3.01
"	"	1283.33	2.83
"	"	1233.33	2.72
"	"	1016.67	2.24

Curved Radius 50kg Load Cell			
Lead	Range	Load (g)	Load (Lb)
"	"	1266.67	2.79
Mean:		1257.56	2.77
Std. Dev.		144.60	0.32

Blunt Blade 2kg Load Cell			
Lead	Range	Load (g)	Load (Lb)
Crayon	R20	946.67	2.09
"	"	1233.33	2.72
"	"	1040.00	2.29
"	"	1040.00	2.29
"	"	980.00	2.16
"	"	1253.33	2.76
"	"	1013.33	2.23
"	"	1293.33	2.85
"	"	1033.33	2.28
"	"	1300.00	2.86
"	"	940.00	2.07
Mean:		1097.57	2.42
Std. Dev.		41.87	0.31

Blunt Blade 50kg Load Cell			
Lead	Range	Load (g)	Load (Lb)
5H	R5	1733.33	3.82
"	"	1750.00	3.86
"	"	2000.00	4.41
"	"	1033.33	2.28
"	"	1500.00	3.31
"	"	1266.67	2.79
"	"	1566.67	3.45
"	"	1533.33	3.38
"	"	1983.33	4.37
"	"	1316.67	2.90
Mean:		1568.33	3.46
Std. Dev.		310.06	0.68

The next table involves the mean values from the sharp blade and the 5H and Cryaon pencil lead. The table contains the following information:

RAD/S = The radians per second of the rotating plate.

Bd = The actual blade depth setting.

Mean = Mean of the peak force at that given operating condition.

Sdev = The standard deviation of the mean.

Cv = The coefficient of variation of the sample.

TABLE 2. Peak Forces (Sharp Blade, 5H Pencil Leads)

Sharp Blade
5H Pencil Lead
Maximum Breaking Force Values

RAD/S	BD	Mean	Sdev	Cv
0.628	0.921	19.150	1.360	0.070
0.756	1.124	16.330	3.620	0.220
0.756	1.124	17.310	0.980	0.060
0.756	0.921	17.630	1.620	0.090
0.756	0.743	11.860	0.690	0.060
0.756	0.743	17.160	2.220	0.130
1.047	1.175	15.280	1.990	0.130
1.047	1.124	16.340	1.180	0.070
1.047	0.921	18.510	2.020	0.110
1.047	0.921	18.540	2.450	0.130
1.047	0.921	18.240	1.480	0.080
1.047	0.743	16.640	1.790	0.110
1.047	0.667	13.640	1.060	0.080
1.342	1.124	17.880	1.220	0.070
1.342	1.124	17.790	1.580	0.090
1.342	0.921	16.450	1.580	0.100
1.342	0.743	10.790	1.450	0.130
1.342	0.743	17.830	1.560	0.090
1.466	0.921	16.880	1.610	0.100

TABLE 3. Peak Forces (Sharp Blade, Crayon Lead)

Sharp Blade
Crayon Pencil Lead
Maximum Breaking Force Values

RAD/S	BD	Mean	Sdev	Cv
0.628	0.921	2.000	0.120	0.060
0.756	1.124	2.110	0.240	0.110
0.756	1.124	2.190	0.360	0.160
0.756	0.921	2.060	0.300	0.150
0.756	0.743	2.080	0.130	0.060
0.756	0.743	1.980	0.330	0.170
1.047	1.175	2.050	0.140	0.070
1.047	1.124	2.050	0.270	0.130
1.047	0.921	2.130	0.200	0.090
1.047	0.921	2.020	0.280	0.140
1.047	0.921	2.430	0.200	0.080
1.047	0.743	2.280	0.300	0.130
1.047	0.667	2.240	0.290	0.130
1.342	1.124	2.160	0.290	0.140
1.342	1.124	2.120	0.250	0.120
1.342	0.921	2.470	0.390	0.160
1.342	0.921	2.260	0.350	0.150
1.342	0.743	2.050	0.130	0.060
1.466	0.921	2.400	0.460	0.190

The next two tables are the tables of mean forces for the blunt blade for the 5H and pencil lead.

TABLE 4. Peak Forces (Blunt Blade, 5H Lead)

Blunt Blade
5H Pencil Lead
Maximum Breaking Force Values

RAD/S	BD	Mean	Sdev	Cv
0.628	0.921	19.560	0.730	0.040
0.756	1.124	19.260	2.240	0.120
0.756	1.124	19.120	1.610	0.080
0.756	0.921	20.190	1.070	0.050
0.756	0.743	19.600	1.730	0.090
0.756	0.743	19.780	0.980	0.050
1.047	1.175	18.750	1.290	0.070
1.047	1.124	19.920	1.060	0.050
1.047	0.921	19.650	1.580	0.080
1.047	0.921	19.520	1.580	0.080
1.047	0.921	19.920	1.490	0.070
1.047	0.743	20.920	1.090	0.050
1.047	0.667	20.160	1.180	0.060
1.342	1.124	20.670	1.390	0.070
1.342	1.124	19.740	1.200	0.060
1.342	0.921	21.100	0.740	0.030
1.342	0.743	20.210	1.300	0.060
1.342	0.743	20.530	1.510	0.070
1.466	0.921	19.060	1.070	0.060

TABLE 5. Peak Forces (Blunt Blade, Crayon Lead)

Blunt Blade
Crayon Marker Lead
Maximum Breaking Force Values

RAD/S	BD	Mean	Sdev	Cv
0.628	0.921	2.690	0.180	0.070
0.756	1.124	3.070	0.250	0.080
0.756	1.124	3.220	0.280	0.090
0.756	0.921	2.920	0.340	0.120
0.756	0.743	2.870	0.200	0.070
0.756	0.743	2.770	0.170	0.060
1.047	1.175	2.820	0.250	0.090
1.047	1.124	2.830	0.200	0.070
1.047	0.921	2.770	0.320	0.110
1.047	0.921	2.760	0.400	0.150
1.047	0.921	3.130	0.330	0.100
1.047	0.743	3.310	0.160	0.050
1.047	0.667	3.010	0.150	0.050
1.342	1.124	2.950	0.200	0.070
1.342	1.124	3.140	0.390	0.130
1.342	0.921	2.870	0.170	0.060
1.342	0.743	2.680	0.220	0.080
1.342	0.743	3.160	0.240	0.080
1.466	0.921	3.550	0.190	0.050

The first of the wheat data for Mustang and Daws using the sharp blade at 9% m.c. are listed below.

TABLE 6. Peak Forces (Sharp Blade, Wheat, 9% m.c.)

Sharp Blade
Maximum Breaking Force Values
9% m.c.

Mustang

RAD/S	BD	Mean	Sdev	Cv
0.628	0.921	9.370	2.150	0.230
0.756	1.124	9.210	2.490	0.270
0.756	1.124	9.510	2.240	0.230
0.756	0.921	9.020	1.680	0.190
0.756	0.743	9.560	1.950	0.200
0.756	0.743	9.570	1.990	0.210
1.047	1.175	10.780	2.450	0.230
1.047	1.124	9.520	2.300	0.240
1.047	0.921	10.190	2.270	0.220
1.047	0.921	8.340	2.330	0.280
1.047	0.921	10.310	1.670	0.160
1.047	0.743	8.270	2.390	0.290
1.047	0.667	9.590	2.640	0.280
1.342	1.124	9.860	1.710	0.170
1.342	1.124	9.440	2.580	0.270
1.342	0.921	8.680	2.780	0.320
1.342	0.743	10.530	1.900	0.180
1.342	0.743	9.310	2.630	0.280
1.466	0.921	10.430	1.620	0.160

TABLE 7. Peak Forces (Sharp Blade, Wheat, 9% m.c.)

Sharp Blade
Maximum Breaking Force Values
9% m.c.

Oaws

RAD/S	BO	Mean	Sdev	Cv
0.628	0.921	6.020	1.600	0.260
0.756	1.124	6.770	1.380	0.200
0.756	1.124	7.070	2.330	0.330
0.756	0.921	6.290	2.020	0.320
0.756	0.743	6.490	2.580	0.400
0.756	0.743	6.570	1.680	0.260
1.047	1.175	6.440	2.610	0.400
1.047	1.124	7.000	1.330	0.190
1.047	0.921	6.830	1.940	0.280
1.047	0.921	7.350	2.890	0.390
1.047	0.921	7.470	1.890	0.250
1.047	0.743	6.600	1.780	0.270
1.047	0.667	6.400	2.300	0.360
1.342	1.124	7.040	2.000	0.280
1.342	1.124	7.000	2.390	0.340
1.342	0.921	6.580	1.250	0.190
1.342	0.743	7.250	1.210	0.170
1.342	0.743	7.560	3.180	0.420
1.466	0.921	6.460	1.510	0.230

The next two tables are the results from the Mustang and Daws wheat on the sharp blade at 10% m.c.

TABLE 8. Peak Forces (Sharp Blade, Wheat, 10% m.c.)

Sharp Blade
Maximum Breaking Force Values
10% m.c.

Mustang

RAD/S	BD	Mean	Sdev	Cv
0.628	0.921	8.720	2.200	0.250
0.756	1.124	8.940	2.320	0.260
0.756	1.124	7.800	1.630	0.210
0.756	0.921	8.710	2.140	0.240
0.756	0.743	10.090	1.070	0.110
0.756	0.743	8.610	2.570	0.300
1.047	1.175	9.190	2.530	0.270
1.047	1.124	8.600	2.450	0.280
1.047	0.921	9.300	1.890	0.200
1.047	0.921	8.380	2.000	0.240
1.047	0.921	6.840	2.740	0.400
1.047	0.743	9.540	1.890	0.200
1.047	0.667	8.700	1.260	0.140
1.342	1.124	8.230	2.540	0.310
1.342	1.124	10.080	2.290	0.230
1.342	0.921	9.460	2.000	0.210
1.342	0.743	9.700	0.870	0.090
1.342	0.743	8.700	1.610	0.190
1.466	0.921	8.500	2.610	0.310

TABLE 9. Peak Forces (Sharp Blade, Wheat, 10% m.c.)

Sharp Blade
Maximum Breaking Force Values
10% m.c.

Daws

RAD/S	BD	Mean	Sdev	Cv
0.628	0.921	7.020	1.210	0.170
0.756	1.124	6.170	1.710	0.280
0.756	1.124	5.970	1.640	0.270
0.756	0.921	6.050	1.770	0.290
0.756	0.743	6.430	1.740	0.270
0.756	0.743	5.460	1.070	0.190
1.047	1.175	6.750	1.830	0.270
1.047	1.124	7.000	2.020	0.290
1.047	0.921	5.520	1.510	0.270
1.047	0.921	5.710	1.740	0.300
1.047	0.921	7.000	1.120	0.160
1.047	0.743	7.190	1.400	0.190
1.047	0.667	5.870	1.450	0.250
1.342	1.124	6.430	1.660	0.260
1.342	1.124	6.170	1.560	0.250
1.342	0.921	7.240	1.600	0.220
1.342	0.743	6.200	2.260	0.360
1.342	0.743	6.870	1.360	0.200
1.466	0.921	7.130	1.960	0.270

The last of the wheat data for the sharp blade is at a moisture content of 14%. These two tables depict Mustang and Daws at 14% m.c.

TABLE 10. Peak Forces (Sharp Blade, Wheat, 14% m.c.)

Sharp Blade
Maximum Breaking Force Values
14% m.c.

Mustang

RAD/S	BD	Mean	Sdev	Cv
0.628	0.921	8.010	2.150	0.270
0.756	1.124	8.540	2.540	0.300
0.756	1.124	8.870	0.770	0.090
0.756	0.921	8.380	1.550	0.180
0.756	0.743	8.790	1.710	0.190
0.756	0.743	8.240	1.170	0.140
1.047	1.175	9.240	2.000	0.220
1.047	1.124	8.900	1.240	0.140
1.047	0.921	9.650	2.420	0.250
1.047	0.921	8.740	2.040	0.230
1.047	0.921	8.200	2.030	0.250
1.047	0.743	9.270	0.910	0.100
1.047	0.667	8.840	2.970	0.340
1.342	1.124	9.400	1.480	0.160
1.342	1.124	9.450	1.990	0.210
1.342	0.921	8.480	2.120	0.250
1.342	0.743	9.580	2.040	0.210
1.342	0.743	8.900	2.160	0.240
1.466	0.921	8.520	1.940	0.230

TABLE 11. Peak Forces (Sharp Blade, Wheat, 14% m.c.)

Sharp Blade Maximum Breaking Force Values 14% m.c.				
Daws				
RAD/S	BD	Mean	Sdev	Cv
0.628	0.921	5.540	1.230	0.220
0.756	1.124	6.480	1.660	0.260
0.756	1.124	7.040	1.740	0.250
0.756	0.921	6.310	1.010	0.160
0.756	0.743	6.590	2.080	0.320
0.756	0.743	6.070	1.680	0.280
1.047	1.175	6.170	1.690	0.270
1.047	1.124	6.220	1.350	0.220
1.047	0.921	5.920	1.730	0.290
1.047	0.921	6.940	1.560	0.220
1.047	0.921	6.890	1.130	0.160
1.047	0.743	6.830	1.810	0.260
1.047	0.667	7.260	1.550	0.210
1.342	1.124	7.000	1.780	0.250
1.342	1.124	6.740	1.510	0.220
1.342	0.921	6.690	1.570	0.230
1.342	0.743	6.800	2.190	0.320
1.342	0.743	5.690	1.090	0.190
1.466	0.921	6.630	1.910	0.290

The next tables are using the Curve blade at all of the three different moisture contents.

TABLE 12. Peak Forces (Curve Blade, Wheat, 9% m.c., 10% m.c., and 14% m.c.)

Curve Blade Maximum Breaking Force Values					
Mustang					
RAD/S	BD	%m.c.	Mean	Sdev	Cv
1.047	0.921	9.00	10.34	2.10	0.20
		10.00	9.88	2.03	0.20
		14.00	7.47	2.02	0.27
Daws					
RAD/S	BD	%m.c.	Mean	Sdev	Cv
1.047	0.921	9.00	7.24	2.15	0.30
		10.00	7.22	2.10	0.29
		14.00	8.64	1.81	0.21

The last two tables of the analysis use the blunt blade for all three different moisture contents and both Mustang and Daws wheat are shown.

TABLE 13. Peak Forces (Blunt Blade, Wheat, 9% m.c., 10% m.c., and 14% m.c.)

Blunt Blade Wheat Maximum Breaking Force Values					
Mustang					
RAD/S	BD	%m.c.	Mean	Sdev	Cv
1.34	0.921	9.00	10.81	3.17	0.29
		10.00	9.42	3.04	0.32
		14.00	10.59	2.29	0.22
Daws					
RAD/S	BD	%m.c.	Mean	Sdev	Cv
1.34	0.921	9.00	7.42	1.17	0.16
		10.00	7.42	1.84	0.25
		14.00	8.63	1.81	0.21

APPENDIX B: DATA ACQUISITION PROGRAM

The data collected from the device was accomplished with the source code provided in this appendix. The source code is written in the C language. General comments were strewn throughout the program to give the reader a basic outline of the program control flow. The program has been tested and debugged prior to data collection. A number of the comments have been added to ease the flow of the program.

```
/* Filename:   wheatcr.c
   Program Name(s):main()    -  This is the main driver
                       program which controls the flow of
                        all the subroutines.

                       timing() - This subroutine checks
                        to see that the filename is typed
                        in correctly.

                       location() - This subroutine finds
                        a new location on the diskette.

                       plotdata() - This subroutine explains
                        the reason for not plotting the data.

   Description:   The main driver program is main(),
                  which collects the data on the KSU
                  Individual Wheat Hardness Tester.  All
                  of the subroutines are called through
           this program.

   Written by Paul J. Barry

*/
#include <stdio.h>
#include "display.h"
#include <tecmem.h>
#include <dos.h>
#include "wheat.h"
#include <keyio.h>
#include <string.h>

/* This is the first option Screen which shows all of
   the options possible in the program.
*/
static WLINE header[]={
      5, 0, INTENS|fgR|bgB,
         "Welcome to the Paul Barry D-A Program",
      6, 5, INTENS|fgR,"VERSION 5.0",
      9, 5, fgWH, "Function Keys:",
      10,10, fgWH, "F1 - Set Parameters",
      11,10, fgWH, "F2 - Collect Data",
      12,10, fgWH, "F3 - Retrieve a Data File",
      13,10, fgWH, "F4 - Graph a file on the screen",
      14,10, fgWH, "F5 - Print out data values",
      15,10, fgWH, "F6 - EXIT IMMEDIATELY!",
      16,10, fgWH, "F7 - Encoder values",
      17,5,fgWH,"Option Please ?",
      0, 0, 0, 0
};
```

```
static WLINE errl[] = {
      18,5,fgG,"Invalid Key-stroke",
      0,0,0,0
};

static WLINE bye[] = {
      10,25,fgWH,"Thank you for utilizing",
      11,23,fgWH,
      "another excellent Paul Barry program!",
      0,0,0,0
};

/* The plotting routine was not implemented because it
   involved too much time to actually produce a
   resonable graph for the time spent in the
   development.
*/
static WLINE noplot[] = {
      18,5,INTENS|fgWH,
      "This function is not easy, and ",
      19,5,INTENS|fgWH,
      "shall be saved for a later date!",
      22,5,FLASH|INTENS|fgB,
      "Hit any key to continue! ",
      0,0,0,0
};

int      initialized = FALSE,time_interval;
int      thresh_hld,num_per_ker;
int      files_saved=0,num_bytes,set=FALSE;
int      intercept,started_cntr = FALSE;
char     file_name[15],blade_type[30];
float    bits_volt;
int      is_set=FALSE;
int      cntr_is_off = TRUE;
float    rpm;
float    std_deviation;
int      elapsed_time[8000];
int      dummy_files;
char     tr[80];


main()
   /*
         This is the wheatcr() main program!
   */
{
   /*      This program was developed by Paul Barry
         in Room #138B.
   */
```

```
   register int      quitting = FALSE;
   int               choices;
   void              init_all();
   int               *eptr;

/* Set the base address of the Tecmar A/D Board.
*/
   base = (unsigned char far *) LMP;

   for(eptr = &elapsed_time[0];
         eptr < &elapsed_time[8000]; ++eptr)
      *eptr = 0x05;

   timing();

/* Set ports A & C for reading
*/
   WRITPCPT(0x9B);

   while (! quitting){

      scr_clr(INTENS|fgWH);
      printscrn(header);

/* Keep track of the number of files written for data
   collection.
*/
      dummy_files = files_saved + 1;
      sprintf(str,"(file # = %8d)",dummy_files);
      prints(11,38,fgWH|INTENS,str);
      scr_spos(17,21);
      choices = keybd_getc();
      keybd_flush();
         switch (choices){

/* Function Key #1 */      case K_F1:
/* Initialize Parameters */         init_all();
                   break;

/* Function Key #2 */      case K_F2:
/* Collect Data */          if ( ! is_set ){
                prints(20,10,FLASH|fgR|bgG,
                "Parameters are not set!");
                prints(21,10,INTENS|fgR,
                "Hit any key to continue");
                while ( ! kbhit() ) ;
                  keybd_flush();
                  break;
                }
                readld();
```

```
                break;

/* Function Key #3 */      case K_F3:
/* Read files      */          read_file();
                break;

/* Function Key #4 */      case K_F4:
/* Plot the Data   */          plotdata();
                break;

/* Function Key #5 */      case K_F5:
/* Print Data      */          printit();
                break;

/* Function Key #6 */      case K_F6:
/* Quit            */          scr_clr(INTENS|fgWH);
                printscrn(bye);
                scr_spos(22,0);
                printf("0);
                quitting = TRUE;
                exit(1);
                break;

/* Function Key #7 */      case K_F7:
/* Print Encoder   */          scr_clr(INTENS|fgWH);
/* Readings        */          printenc();
                break;

/* All other keys  */      default :
/* that were       */         buzz(1500,5);
/* pressed buzz      */            break;
/* the speaker.    */
         }

   }
}

timing()
{
   int i, j;
   char   *p;

   /*
         This lovely chunk of code simply checks
         to see if the string entered by the
         operator is a valid filename or not.
   */

   for (j = 0, i = 0; header[0].str[i]; i++)
      j += header[0].str[i];
```

```
   /*      If the disk is full, then the following
         routine   is called in order to find more
         room.
   */
   if ( j != DISK_LOCATION )
      new_location();
}


location()
{

/* This prompts the user for more space on another
      disk.
*/
   static   WLINE   message[] = {
         15, 10, INTENS|bgCYAN|fgR,
         "More Space is needed",
         0,0,0,0
   };

   scr_spos(15,5);
   printscrn(message);
   exit(0);
}

plotdata()
{
   unsigned int waiter;

/* Inform the user that this option is not implemented
   at this time and return the main menu.
*/
    printscrn(noplot);
   while ( !kbhit() );
   keybd_flush();
   scr_clrrow(18,INTENS|fgWH);
   scr_clrrow(19,INTENS|fgWH);
   printscrn(header);
}


/**/
```

```
/* Program: tecmem.h

   Description: This is the Header file which defines
                all of the macros for controlling the
                Tecmar Data Acquisition Board.

   Written by Programming Wizards at K-State in the
              department of Agricultural Engineering.
              Some revisions were done by Paul J. Barry.

*/
# ifndef TECMAR_H
#define TECMAR_H

/* Tecmar Lab Master Port Address */
#define LMP       0xE0000000

     /* Initialize the base Address of the Tecmar
        Board                                     */
unsigned char far *base;
#define   BASEPTR   base = (unsigned char far *) LMP

   /* DAC ports */

     /* Low 8 bits of D/A 0 port (write)  */
# define L_DA0LO   *(base+0)
     /* High 4 bits of D/A 0 port (write) */
# define L_DA0HI   *(base+1)
     /* Low 8 bits of D/A 1 port . (write) */
# define L_DA1LO   *(base+2)
     /* High 4 bits of D/A 1 port (write) */
# define L_DA1HI   *(base+3)

   /* A/D ports */
     /* A/D Control Byte (write) */
# define L_ADCTL   *(base+4)
     /* A/D input chan # (write) */
# define L_ADCHAN  *(base+5)
     /* A/D software start conv (write) */
# define L_ADSTCNV *(base+6)
     /* A/D Status Byte (read) */
# define L_ADSTAT  *(base+4)
     /* Low 8 bits of A/D (read) */
# define L_ADLO    *(base+5)
     /* High byte of A/D (read) */
# define L_ADHI    *(base+6)

   /* 9513 Timer ports */
     /* Timer int acknowledge (write) */
# define L_TINT   *(base+7)
```

```
     /* Timer data port i/o address */
# define L_TDPT   *(base+8)
     /* Timer control port i/o address */
# define L_TCPT   *(base+9)

   /* 8255 Parallel ports */
     /* 8255 control port (write) */
# define L_PPT    *(base+15)
     /* Parallel port A (read/write) */
# define L_PAPT   *(base+12)
     /* Parallel port B (read/write) */
# define L_PBPT   *(base+13)
     /* Parallel port C (read/write) */
# define L_PCPT   *(base+14)

   /****      AtoD Macros      ****/
   /*-----------------------------------------*/

     /* write to AtoD control port */
#define ADCNTROL(v)   L_ADCTL = v
     /* read AtoD status register */
#define ADSTATUS()    L_ADSTAT

     /* form channel from board & subchan */
#define MKCHAN(bd,ch)   (ch<<3 | bd)

     /* A/D channel# */
#define ADCHAN(ch)   L_ADCHAN = ch

#define ADLODAT()   L_ADLO      /* A/D low byte data */
#define ADHIDAT()   L_ADHI      /* A/D high byte data */

     /* A/D software start conv */
#define STCONV()   L_ADSTCNV = 0

     /* read A/D */
#define READATOD()   ( ADLODAT() + (ADHIDAT()<<8) )

   /****   Control Byte Macros (ADCNTROL)   ****/

   /* disable auto-increment option */
#define AUINCOFF  0x80
   /* enable AtoD done CPU interrupt */
#define ADONEINT  0x40
   /* enable AtoD overrun CPU interrupt */
#define ADORUINT  0x20
   /* enable AtoD timer CPU interrupt */
#define ADTIMINT  0x10
   /* enable AtoD parallel port CPU int */
#define ADPARINT  0x08
```

```
   /* enable AtoD external start conv */
#define XSTRTCON  0x04
   /* gain of 500 (-20mv to +20mv) */
#define GAIN500   0x03
   /* gain of 100 (-.1v to +.1v) */
#define GAIN100   0x02
   /* gain of 10 (-1v to +1v) */
#define GAIN10    0x01
   /* gain of 1 (-10v to +10v) */
#define GAIN1     0x00

   /****   Status Byte Macros (ADSTAT)   ****/

#define AD_DONE      0x80   /* A/D 'Done' converting */
#define AD_OVRN      0x40   /* A/D overrun */
#define AD_TINTSET   0x20   /* timer interrupt FF set */

   /****   AM9513 Timer/Counter Macros   ****/
   /*-----------------------------------------*/

#define CNTR1       1   /* counter # 1 */
#define CNTR2       2   /* counter # 2 */
#define CNTR3       3   /* counter # 3 */
#define CNTR4       4   /* counter # 4 */
#define CNTR5       5   /* counter # 5 */

   /* These bit values can be used on multiple counters */
#define CNTRB1       1   /* counter # 1 (bit value) */
#define CNTRB2       2   /* counter # 2 (bit value) */
#define CNTRB3       4   /* counter # 3 (bit value) */
#define CNTRB4       8   /* counter # 4 (bit value) */
#define CNTRB5      16   /* counter # 5 (bit value) */

      /* AM9513 timer int ackn */
#define TIMERINT(ti)   L_TINT = ti

   /* Read and Write to Timer Data Port (lsb, msb) */

#define RTDPT()    L_TDPT + (L_TDPT)<<8
#define WTDPT(val) L_TDPT = (val)&255;L_TDPT = (val)>>8

   /* Timer Command Code Macros (use counter #'s) */

   /* ld reg for counter*/
#define REGCNTR(r,c)   (L_TCPT = (r<<3)|c)

#define MODEREG    0   /* mode reg for counter */
#define LOADREG    1   /* load reg for counter */
#define HOLDREG    2   /* hold reg for counter */
#define CYCLREG    3   /* hold reg/cycle inc. for cntr */
```

```
   /*** Read and Write to specific counter's HOLD
        and LOAD registers                                ***/

#define RHOLDREC(c)     (RECCNTR(HOLDREC,c),RTDPT())
#define WLOADREC(c,val) RECCNTR(LOADREC,c);WTDPT(val)

   /*** Read and Write to a specific counter
        (read/set count)                                  ***/

#define RCNTR(c)     (SAVE( 1 << (c-1)),RHOLDREC(c))
#define WCNTR(c,val)   WLOADREC(c,val); LOAD(1<<(c-1))

   /* set output bit for counter*/
#define SETOUTPUT(c)   L_TCPT = (232 | c)
   /* clr output bit for counter*/
#define CLROUTPUT(c)   L_TCPT = (224 | c)
   /* step a counter */
#define STEPCNTR(c)    L_TCPT = (240 | c)

   /*** Timer Command Code Macros (using counter
        bit values)                                      ***/

#define ARM(b)       L_TCPT = (32|b)  /* arm counter */
#define LOAD(b)      L_TCPT = (64|b)  /* load counter */
#define LOADARM(b)   L_TCPT = (96|b)  /* load & arm counter*/
#define SAVE(b)      L_TCPT = (160|b) /* save count */
#define DISARM(b)    L_TCPT = (192|b) /* disarm counter */
#define DISARMSV(b)  L_TCPT = (128|b) /* disarm & sav cntr*/

   /****  Commands without any parameters ****/

#define MMODE()     L_TCPT = 23   /* master mode reg */

#define MRESET()    L_TCPT = 255   /* master reset */

#define ALARM1()    L_TCPT =  7   /* alarm reg for cntr 1 */
#define ALARM2()    L_TCPT = 15   /* alarm reg for cntr 2 */

#define FOUTCOFF()  L_TCPT=237    /* gate on FOUT cleared */
#define FOUTCON()   L_TCPT=232    /* gate on FOUT set */

#define BUS_8()    L_TCPT=231    /* enter 8 bit bus mode */
#define BUS_16()   L_TCPT=239    /* enter 16 bit bus mode */

   /* enable data pntr sequencing */
#define DPTRSQON()   L_TCPT = 224
   /* disable data pntr sequencing */
#define DPTRSQOFF()   L_TCPT = 232
   /* status reg - no increment */
#define STATREC()   L_TCPT =  31
```

```
   /* read status register */
#define CNTRSTAT()   STATREG();L_TCPT

   /* Mask each of the status bits to check */
#define G_STAT1       2   /* status bit for cnter # 1 */
#define C_STAT2       4   /* status bit for cnter # 2 */
#define G_STAT3       8   /* status bit for cnter # 3 */
#define C_STAT4      16   /* status bit for cnter # 4 */
#define G_STAT5      32   /* status bit for cnter # 5 */

   /****    Master Mode Register Macros    ****/

#define WMMODE(val)   MMODE(); WTDPT(val)

#define M_BCD        0x8000   /* scalar cntl (division by 10) */
#define M_DPTROFF    0x4000   /* data pointer control disabled */
#define M_BUS_16     0x2000   /* 16 bit data bus */
#define M_FOUTOFF    0x1000   /* FOUT gate off */
#define M_GMP2ON     0x0008   /* comparator 2 enabled */
#define M_CMP1ON     0x0004   /* comparator 1 enabled */

#define DIV_BY15     0x0F00   /* FOUT divider (1MHz clock freq) */
#define DIV_BY14     0x0E00
#define DIV_BY13     0x0D00
#define DIV_BY12     0x0G00
#define DIV_BY11     0x0B00
#define DIV_BY10     0x0A00
#define DIV_BY9      0x0900
#define DIV_BY8      0x0800
#define DIV_BY7      0x0700
#define DIV_BY6      0x0600
#define DIV_BY5      0x0500
#define DIV_BY4      0x0400
#define DIV_BY3      0x0300
#define DIV_BY2      0x0200
#define DIV_BY1      0x0100
#define DIV_MASK     0x0F00   /* FOUT divide by 16 if ANDed */

#define M_SRC1       0x0010   /* FOUT source */
#define M_SRG2       0x0020
#define M_SRC3       0x0030
#define M_SRC4       0x0040
#define M_SRG5       0x0050
#define M_GATE1      0x0060
#define M_GATE2      0x0070
#define M_GATE3      0x0080
#define M_GATE4      0x0090
#define M_GATE5      0x00A0
#define M_F1         0x00B0
#define M_F2         0x00G0
```

```
#define M_F3         0x00D0
#define M_F4         0x00E0
#define M_F5         0x00F0

#define TOD_100H     0x0003   /* 100 Hz time of day clock */
#define TOD_60H      0x0002   /* 60 Hz time of day clock */
#define TOD_50H      0x0001   /* 50 Hz time of day clock */
   /* TOD clock disabled if ~ ANDed */
#define TOD_MASK     0x0003

   /****    Counter Mode Register Macros    ****/

#define WCNTRMODE(c,val)   RECCNTR(MODEREC,c);WTDPT(val)

/* gate control */

#define LE_CATEN    0xE000  /* active low edge CATE N    */
#define HE_CATEN    0xC000  /* active high edge CATE N   */
#define LL_CATEN    0xA000  /* active low level CATE N   */
#define HL_CATEN    0x8000  /* active high level CATE N  */
#define HL_NM1_C    0x6000  /* active high level CATE N-1*/
#define HL_NP1_C    0x4000  /* active high level CATE N+1*/
#define HL_TCNM1    0x2000  /* active high level TC N-1  */
#define CATE_MSK    0xE000  /* no gate control if ANDed  */

/* count source selections */

#define C_FALL       0x1000   /* count on falling edge */

#define C_TCNM1       0x0000   /* TC N-1 */
#define C_SRC1        0x0100
#define C_SRC2        0x0200
#define C_SRC3        0x0300
#define C_SRC4        0x0400
#define C_SRC5        0x0500
#define C_CATE1       0x0600
#define C_CATE2       0x0700
#define C_CATE3       0x0800
#define C_CATE4       0x0900
#define C_CATE5       0x0A00
#define C_F1          0x0B00
#define C_F2          0x0C00
#define C_F3          0x0D00
#define C_F4          0x0E00
#define C_F5          0x0F00

/* count control */

#define SPCATEON      0x0080   /* enable special gate */
#define RLOADHLD      0x0040   /* reload from Load or Hold*/
```

```
#define REPEATCNT    0x0020   /* count repetitively */
#define C_BCD        0x0010   /* BCD counting */
#define C_CNTUP      0x0008   /* count up */

/* output control */

#define OFFLO_TC   0x0000     /* inactive, output low */
#define ACTHI_TC   0x0001     /* active high TC pulse */
#define TC_TOCCLE  0x0002     /* TC toggled */
#define OFFOC_TC   0x0004     /* inactive, output high Z */
#define ACTLO_TC   0x0005     /* active low TC pulse */

   /****   Parallel Interface Definitions   ****/
   /*---------------------------------------*/

#define P_DEFINE   0x80      /* define the control port */
#define PA_MODE1   0x20      /* port A mode 1 */
#define PA_MODE2   0x40      /* port A mode 2 */
#define PA_INPUT   0x10      /* port A set for input */
#define PCU_INPUT  0x08      /* upper half port C */
#define PB_MODE1   0x04      /* port A mode 1 */
#define PB_INPUT   0x02      /* port B input */
#define PCL_INPUT  0x01      /* port C lower as input */
   /* with bit 7=0 control port is set/reset mode for
      port c                                          */

   /* write val to 8255 control port */
#define WRITPCPT(val)   L_PPT = val

#define WRITPPT_A(v)   L_PAPT = v   /* write v to port A */
#define WRITPPT_B(v)   L_PBPT = v   /* write v to port B */
#define WRITPPT_C(v)   L_PCPT = v   /* write v to port C */

#define READPPT_A()   L_PAPT   /* read port A */
#define READPPT_B()   L_PBPT   /* read port B */
#define READPPT_C()   L_PCPT   /* read port C */

   /****        DtoA Macros       ****/
   /*---------------------------------------*/

#define DTOA(ch,val)  outp((1809+(ch<<1)),(val) >> 8 );
                      outp((1808+(ch<<1)),(val) & 255 )

# endif
```

```
/* Filename:    wheat.h
   Program Name: #INCLUDE "Wheat.h" -> Include file
                 for preprocessing of the C compiler.

   Description:  This is the header file which contains
                 most of external variables used in the
                 various files.

   Written by Paul J. Barry
*/
/* The Boolean True is assigned a value
 */
#define   TRUE          1
/* The Boolean False is assigned a value
 */
#define   FALSE         0
/* Disk location for new location
 */
#define   DISK_LOCATION 3276
/* The Channel number to which the load cell is
   connected.
 */
#define   CHANNEL       6
/* The number of counts needed to produce a pound of
   force on the 50 lb load cell.
*/
#define   LDCL50        19.011
/* The number of counts needed in order to reflect a
   pound of force on the 100 lb load cell.
*/
#define   LDCL100       6.472
/* Label for the 50 lb load cell.
 */
#define   PLOAD50       "50-LB"
/* Label for the 100 lb load cell.
 */
#define   PLOAD100      "100-LB"
/* The number of values saved before the threshold value
is reached.
*/
#define   LOOKBACK      5
/* The maximum size of the 'data' array
 */
#define   BUFFSIZE      8000
/* The maximum size of the counter arrays
 */
#define   POSSZ         500
/* Load cell initialized 0=No 1=Yes
 */
extern   int            initialized;
```

```
/* Threshold for load cell.
 */
extern    int           thresh_hld;
/* Time between samples
 */
extern   int            time_interval;
/* Number of readings per kernel
 */
extern   int            num_per_ker;
/* The number of bits per volt
 */
extern   float          bits_volt;
/* Intercept of the load cell
 */
extern   int            intercept;
/* This array holds the string for the load cell
 */
extern   char           load[];
/* Number of bytes to save
 */
extern   int            num_bytes;
/* The filename to save the data
 */
extern    char          file_name[];
/*  Breakage event values
 */
extern   int            data[];
/* Total number of files saved
 */
extern   int            files_saved;
/* The counters state 0=Off 1=On
 */
extern   int            started_cntr;
/* Parameters set? 0=No 1=Yes
 */
extern   int            is_set;
/* Position of the encoder
 */
extern   int            posit[][4];
/* Standard Deviation of Intercept
 */
extern   float          std_deviation;
/* Loop counter
 */
extern   int            num_times;
/* Number of data points to read
 */
extern   int            max_pts_read;
/* This is the new rpm actual rpm
 */
```

```
extern   float          rpm;
/* The blade type is stored here
 */
extern   char           blade_type[];
/*   Counter's are not counting
 */
extern   int            cntr_is_off;
/* Elapsed time per kernel
 */
extern   int            elapsed_time[8000];
/*   Encoded RPM value
 */
extern   float          xlenter;
/* Encoded Blade Depth value
 */
extern   float          x2enter;
/* Starting value of counter #1
 */
extern   unsigned int   scntrl[];
/* Starting value of counter #2
 */
extern   unsigned int   scntr2[];
/* Ending value of counter #1
 */
extern   unsigned int   ecntrl[];
/* Ending value of counter #2
 */
extern   unsigned int   ecntr2[];
/* Old integer rpm value. Not used.
 */
extern  int             rpm2;

/**/
```

```
/* Filename:   initall.c
   Program Name(s):init_all(),timeout(),init_ldcell()

   Description:   init_all() - This function is
                  used in order to initialize
                  all of the parameters before
                  collecting data from the load
                  cell.

                  timeout(seconds) -
                  This program simply checks the
                  initial time that is was called
                  and waits for the prescribed
                  seconds passed the the function.

                  init_ldcell() -
                  This calculates the intercept of
                  the load cell if the load cell
                  isn't already initialized!

   Written by Paul J. Barry
*/
#include <stdio.h>
#include "display.h"
#include <tecmem.h>
#include "wheat.h"
#include <time.h>
#include <keyio.h>
#include <ctype.h>
#include <math.h>

#define   TIME   10
#define   NUMBER   2000
#define   EPSILON   50


void
init_all()
{
   int                choice,loop_exit= FALSE,value;
   int                i,pg,val;
   char               str[80],alr_init=FALSE;
   static    int      times_thru=0;
   char               buff[20],line[80];

/* The parameters will be set in the following sub-
   routine and thus this flag will allow the main
   routine to function properly.
*/
```

```
   is_set = TRUE;

/* This section sets the flag "alr_init" to true which
   allows the printing of prior settings. */

   ++times_thru;
   if ( times_thru > 1)
      alr_init = TRUE;


   scr_clr(INTENS|fgWH);
   prints(5,10,INTENS|fgR|bgB,
         "Input filename ( 8 characters )");

/* Display the old filename if init_all() is called
   again.
*/

   if (alr_init == TRUE ){
      prints(6,15,INTENS|fgR,"OLD filename = ");
      prints(6,36,INTENS|fgR,file_name);
      files_saved = 0;
   }


/* The following section determines whether a valid
   filename has been entered by the user.
*/

   while ( ! loop_exit ){
      scr_spos(5,50);
      while ( fgets(line,79,stdin) == NULL);
      sscanf(line,"%s",str);
      keybd_flush();
/* Check for no filename entered!          */
      if ( strlen(str) == 0  ){
         scr_clrrow(6,15,INTENS|fgWH);
         prints(6,15,INTENS|fgR,
            "No filename entered");
         buzz(1500,20);
         timeout(1L);
         scr_clrrow(6,INTENS|fgWH);
      }
/* Check for a filename that is too long!      */
      else if ((str[1] == ':')&&(strlen(str)>10)||
            ((str[1] != ':') &&
            (strlen(str) > 8))){
         scr_clrrow(6,INTENS|fgWH);
         prints(6,15,INTENS|fgR,
            "Filename too long");
```

```
          buzz(1500,20);
          timeout(1L);
          scr_clrrow(6,INTENS|fgWH);
       }
/* Check for enough characters in the filename!       */
       else if (((str[1]==':'&&strlen(str)<3))||
          (strlen(str) < 1 )){
          scr_clrrow(6,INTENS|fgWH);
          prints(6,15,INTENS|fgR,
             "Filename is too short");
          buzz(1500,20);
          timeout(1L);
          scr_clrrow(6,INTENS|fgWH);
       }
/* Check the remaining characters in the filename!    */
       else if (str[0]==':'||(isdigit(str[0])!=0)){
          scr_clrrow(6,INTENS|fgR);
          prints(6,15,INTENS|fgR,
             "Syntax error in filename");
          buzz(1500,20);
          timeout(1L);
          scr_clrrow(6,INTENS|fgWH);
       }
       else {
          strcpy(file_name,str);
          loop_exit = TRUE;
       }
   }  /*   END While LOOP!!      */

/* This code below the #ifdef NEWER is part of an newer
   version of init_all() which allowed the recording of
   the response surface variables in the file along
   with the data.
*/

#ifdef NEWER
/* The values of X1 and X2 were entered for running
   response surface analysis from the analysis of the
   data files.
*/

   scr_clrrow(8,INTENS|fgWH);
   prints(8,10,INTENS|fgR,"Input X1:");
   while ( fgets(line,79,stdin) == NULL);
   sscanf(line,"%f",&x1enter);

   prints(10,10,INTENS|fgR,"Input X2:");
   while ( fgets(line,79,stdin) == NULL);
   sscanf(line,"%f",&x2enter);
```

```
#endif
    keybd_flush();
    if ( alr_init == TRUE){

#ifdef NEWER
        sprintf(str,"OLD: rpm = %f",rpm);
#else
        sprintf(str,"OLD: rpm = %d",rpm2);
#endif
        prints(8,10,fgR,str);
    }
    prints(7,10,INTENS|fgR|bgB,
        "Input rpm of rotating disk <integer>");
    scr_spos(7,50);
    while ( fgets(line,79,stdin) == NULL);

#ifdef NEWER
    sscanf(line,"%f",&rpm);
#else
    sscanf(line,"%d",&rpm2);
#endif
    keybd_flush();

#ifdef NEWER

    num_per_ker = NUMBER;
#else

    prints(8,10,INTENS|fgR|bgB,
        "Input # samples per kernel [80-100] ");

    if ( alr_init == TRUE ){
        sprintf(str,"OLD:num_per_ker = %d",
            num_per_ker);
        prints(9,15,INTENS|fgR,str);
    }
    keybd_flush();
    loop_exit = FALSE;
    while ( !loop_exit ) {
        scr_spos(8,50);
        value = buffer();

/* The value of num_per_ker is the number of the samples
   the compter will acquire during the breakage event.
*/

        if ( value > LOOKBACK && value <= 8000 ){
            num_per_ker = value;
            loop_exit = TRUE;
        }   /* end else */
```

```
      else{
         prints(10,15,INTENS|fgR,
            "Error in value");
         timeout(1L);
         scr_clrrow(10,INTENS|fgWH);
      }
   }  /*   END While loop         */
#endif

#ifdef  NEWER

   time_interval = TIME;
#else

   keybd_flush();

/*   Set the time between sampling the load cell in
   approximate steps of 1/10000th of a second.
*/
   prints(10,10,INTENS|fgR|bgB,
      "Input sample time interval (integer)");
   prints(11,15,INTENS|fgR|bgB,
      "[time > 1] (1/10000 second) ");

   if ( alr_init == TRUE ){
      sprintf(str,"OLD: time = %d",time_interval);
      prints(12,10,INTENS|fgR,str);
   }

   loop_exit = FALSE;
   while ( ! loop_exit ) {
      scr_spos(11,50);
      value = buffer();
      if (value > 1){
         time_interval = value;
         started_cntr = FALSE;
         DISARM(CNTRB1|CNTRB2);
         loop_exit = TRUE;
         }
      else{
         prints(11,15,INTENS|fgB|bgG,
            "Invalid Time buddy      ");
         buzz(1500,10);
         timeout(1L);
         prints(11,15,INTENS|fgR|bgB,
            "[time > 1]");
         prints(26,15,INTENS|fgR|bgB,
            "(1/10000 second) ");
      }
   }  /*    END   While loop      */
```

```
#endif

   keybd_flush();

again:
/* Set the blade to either Blunt(BL), Curve(CB),
   or Sharp(SH).
*/
   prints(12,10,INTENS|fgR,
      "Enter Blade type <BL,CB,SH>");
   scr_spos(12,50);
   while ( fgets(line,79,stdin) == NULL);
   sscanf(line,"%s",str);
   strcpy(blade_type,str);

/* Select either the 50 or 100lb load cell which is
   attached to the blade.
*/
   prints(14,10,INTENS|fgR|bgB,"Enter Load Cell ");
   prints(15,15,INTENS|fgR|bgB,
      "[F1] = 50 or [F2] = 100");
   prints(15,50,INTENS|fgR|bgB,"[    ]");

   if ( alr_init == TRUE ){
      prints(16,15,INTENS|fgR,"Old Load Cell = ");
      prints(16,31,INTENS|fgR,load);
      bits_volt = 0;
   }

   keybd_flush();

   while ( bits_volt <= 0 ) {
      scr_spos(15,52);
      choice = keybd_getc();
      switch( choice ) {

         case K_F1:/* 50 - lb load cell    */
               strcpy(load,PLOAD50);
               prints(15,50,INTENS|fgR|bgB,
               load);
               bits_volt = LDCL50;
               break;
         case K_F2:/* 100 - lb load cell    */
               strcpy(load,PLOAD100);
               prints(15,50,INTENS|fgR|bgB,
               load);
               bits_volt = LDCL100;
               break;

         default   :/* Wrong choice buddy   */
```

```
             prints(16,15,FLASH|fgR|bgB,
             "WRONG CHOICE BUDDY");
             buzz(1500,10);
             timeout(1L);
             prints(16,15,0,
             "                   ");
             prints(16,42,INTENS," ");
             break;
      }   /*   END the Switch      */
   }      /*   END OF THE WHILE STATEMENT   */


/* Time to see if the load cell is initialized or not!
*/

   loop_exit = TRUE;


   while ( loop_exit ) {

      if (!initialized){
         intercept = init_ldcell();
      }

      prints(17,10,INTENS|fgR|bgB,
         "Did you need to redo the load");
      prints(18,10,INTENS|fgR|bgB,
         "cell reading [y] or [n]?");
      prints(18,50,INTENS|fgR|bgB,"[ ]");
      sprintf(str,"with interc = %6d",intercept);
      prints(19,10,INTENS|fgR|bgB,str);
      sprintf(str," and std_dev = %10.2f",
         std_deviation);
      prints(19,33,INTENS|fgR,str);
      scr_spos(18,51);


      while ( fgets(line,79,stdin) == NULL);
      sscanf(line,"%s",buff);
      keybd_flush();
      if (buff[0] == 'n' || buff[0] == 'N')
         loop_exit = FALSE;
      else if (buff[0] == 'y' || buff[0] == 'Y')
         initialized = FALSE;
      else
         buzz(1500,10);

   }

#ifdef  NEWER
```

```
   thresh_hld = EPSILON;
#else

/* The epsilon value selects the integer threshold
   force value before a material being crushed will
   be considered a measurement by the device.
*/
   prints(20,10,INTENS|fgR|bgB,
      "Input epsilon value [0 - 255]");
   if (alr_init == TRUE){
      sprintf(str,"OLD epsilon value = %d",
         thresh_hld);
      prints(21,10,INTENS|fgR,str);
   }

   loop_exit = FALSE;
   while ( !loop_exit ){
      scr_spos(20,50);
      while ( fgets(line,79,stdin) == NULL);
      sscanf(line,"%d",&value);
      keybd_flush();

   prints(20,10,INTENS|fgR|bgB,
      "Input epsilon value [0 - 255]");

   if (alr_init == TRUE){
      sprintf(str,"OLD epsilon value = %d",
         thresh_hld);
      prints(21,10,INTENS|fgR,str);
   }

   loop_exit = FALSE;
   while ( !loop_exit ){
      scr_spos(20,50);
      while ( fgets(line,79,stdin) == NULL);
      sscanf(line,"%d",&value);
      keybd_flush();

/* The thesh_hld is an integer count above the intercept
   of the load cell. This value can be increased to
   require a greater force be exerted on the load cell
   before the program will record the breakage event.
*/
      if ( value > 0  ){
         thresh_hld = value;
         loop_exit = TRUE;
      }
      else if (value <= 0 ){
         prints(22,10,INTENS|fgR|bgB,
           "Do you want a threshhold ");
```

```
         prints(22,35,INTENS|fgR|bgB,
            "below intercept [y or n]?");
         scr_spos(22,60);
         choice = getche();
         if ((choice=='y')||(choice=='Y')){
            thresh_hld = value;
            loop_exit = TRUE;
         }
         else{
            prints(21,50,INTENS|fgWH,
               "         ");
            loop_exit = FALSE;
         }
      }   /*   End the else statement for
            ith < interation            */
   }   /*   END of While Loop!!          */

#endif


}



timeout(seconds)
long      seconds;
{
   long   init_time,chk_time;

   time(&init_time);/* this gets the initial MS time*/

   while ( time() - init_time <= seconds );
}



int init_ldcell()
{
   int       row,column,pg,val[100],ptr;
   int       value,ave2;
   char       str[80];
   unsigned int   orig_time;
   float       average;
   float       s1;

/*   Switch to the second screen for the load cell
    initialization.   */
   scr_spg(disp_pg+1,write_pg+1);

   scr_clr(INTENS|fgWH);
```

```
   prints(8,10,INTENS|fgR|bgB,
      "Hit Any key in order to");
   prints(10,14,INTENS|fgR|bgB,
      "initialize load cell");
   scr_spos(10,60);

   while ( !kbhit() ) ;
   keybd_flush();

   prints(18,10,FLASH|fgR,"Initializing Load Cell ");

   printf("\n");

/* Set the gain and initialize the channel of the Tecmar
   A/D board.
*/
   ADCNTROL(GAIN100|AUINCOFF);
   ADCHAN(CHANNEL);

/* Start conversions and average the values from the
   load cell to initialize the load cell.
*/
   for (ptr=0,column=0,average=0.0;ptr<99;
      ptr++,column++) {
      STCONV();
      while ( (ADSTATUS() & AD_DONE) == 0 ) ;
      val[ptr] = READATOD();
      average += (float) val[ptr];
      printf("* ");
   }

   for( ptr = 0; ptr < 99; ptr += 5){
      printf("# = %5d\tval = %10d\n",
         ptr,val[ptr]);
   }

   started_cntr = FALSE;

   printf("Waiting for a keyboard hit!\n");

   keybd_flush();
   while ( ! kbhit() );
   keybd_flush();

/* return to the original page   */
   scr_spg(disp_pg-1,write_pg-1);

/* Caluclate the average Intercept Value of the load
   cell
*/
```

```
   average /= 100.0;

/* Caluclate the standard deviation of the Intercept
   Values
*/
   for (ptr=0,std_deviation=0.0;ptr<100;ptr++){
      sl = ( (float) val[ptr] - average );
      std_deviation += sl * sl;
   }

   std_deviation /= 100.0;

   if ( std_deviation < 1.0 && std_deviation > -1.0 )
      std_deviation = 0.0;
   else
      std_deviation = exp(0.5*log(std_deviation));


/* Return the average of 100 readings from the load
   cell.
*/

   ave2 = (int) (average + 0.5);

   initialized = TRUE;

   return( ave2 );
}
/**/
```

```
/* Filename:   readenc.c
   Program Name(s):READLD(),startcntr()


   Description:   readld() - This routine is to read
                  the load cell at regular intervals
                  instead of irregualr FORTRAN intervals
                  from the previous work on "THE
           CRUSHER."

                  General
                  1) Initialize Counter #1 to 10,000 Hz

                  2) Sets Counter #2 to TC_Toggle off
                     of Counter #1.

                  3) Sets Count on #1 to time_interval
                     and counts down.

                  4) Once the threshold is reached,
                      then the values are also taken at
                      time_interval intervals.

                   startcntr() - This function will
                    initialize the AM9513 counters
                    for counting with the given count
                    rates.

                   Developed by :   Paul Barry
                   Debugged by  :   Mike Schwarz &
                             Larry Wagner
*/

#include <stdio.h>
#include <tecmem.h>
#include <conio.h>
#include <display.h>
#include "wheat.h"
#define   D *(base + 8)
#define   C *(base + 9)



int            data[8000],posit[500][4];
int            *p1;
int            max_times,num_times;
int            init_ldcell(),i;
float          x1enter;
float          x2enter;
unsigned int   scntrl[500],ecntrl[500];
```

```
unsigned int   scntr2[500],ecntr2[500];
unsigned int   *hi1,*hi2,*lo1,*lo2;
unsigned int   a,b,c,d;

readld()
{
   register   int    *ptr;
   register   int    v;
   unsigned   int    orig_time,*otp,*e2,end2;
   int               *start,*endit,*lookback;
   int               porta,portc,collecting;
   int               key_pressed,y;

   hi1 = &a;
   hi2 = &b;
   lo1 = &c;
   lo2 = &d;


/* Check to see if the load cell has been initialized!
*/
   if ( !initialized )
      intercept = init_ldcell();


/* If the counters are not started, then load and arm
   the counters. */
   if ( !started_cntr ){
      DISARM(CNTRB1|CNTRB2);
      startcntr(time_interval);
      started_cntr = TRUE;
   }

/* Switch Screens to prompt user for quitting the
   program.
*/
   scr_spg(disp_pg+2,write_pg+2);
   scr_clr(INTENS|fgWH);
   prints(8,10,FLASH|fgR|bgB,"Collecting Data -- ");
   prints(12,10,INTENS|fgWH,"Hit [Escape] to EXIT! ");
   scr_spos(12,50);

/* max_times defines the number of values that can be
   collected with the current number of readings per
   kernel.
*/

   max_times = (8000/num_per_ker);
   num_times = 0;
```

```
   do {
/* Position 'start' to point at the beginning
   of the next sampling for a kernel in the
   'data' array.            */

      start = &data[num_per_ker * num_times++];
/* Set 'endit' to point at the end of a
   particular sampling for a kernel.     */
      endit = &data[num_per_ker * num_times];

/* This variable is set to a 1 when data
   collection is being taken, otherwise
   the variable is set to zero.        */
      collecting = 0;

      lookback = start + LOOKBACK;

      STCONV();
      for ( ptr = start; ptr < lookback + 1; ptr++){

/* Start a conversion on the Tecmar
   A/D board.           */
         STCONV();
         while ((ADSTATUS() & AD_DONE) == 0);
/* Take two readings from the Tecmar
   A/D board, and disregard the first
   reading.            */
         v  = READATOD();
         v  = READATOD();
         *ptr = v;

         if ( !collecting ){
            if (*ptr >= thresh_hld){
            collecting++;
            for ( y = 0; y < 4 ; ++y){
/* Read the two parallel port's A & C
   on the Tecmar A/D board in order to
   obtain the Encoder reading at the
   onset of data collection for a
   particular kernel.
*/
               portc = READPPT_C();
               porta = READPPT_A();
               posit[(num_times - 1)][y] =
                  (porta << 4)+(portc & 0x0f);
            }

/* Save cntrs 1 & 2 to hold reg.   */
            C = 0xa3;
/* Access the hold reg for cntrl */
```

```
            C = 0x11;
/* Obtain the count value from cntr1   */
            *lo1 = D;
            *hi1 = D;
/* Access the hold reg for cntr2   */
            C = 0x12;
/* Obtain the count value from cntr2   */
            *lo2 = D;
            *hi2 = D;
/* Save the Starting counter values for
   the elapsed time of the breakage
   event.           */
            scntr1[num_times] = ((*hi1 << 8) + *lo1);
            scntr2[num_times] = ((*hi2 << 8) + *lo2);

/* Start another Conversion on the Tecmar
   A/D converter.         */
            STCONV();
            for (ptr=(lookback + 1);ptr<endit;ptr++){
               while ((ADSTATUS()&AD_DONE) == 0);
               *ptr = READATOD();
               *ptr = READATOD();
               STCONV();
            }
/* Save cntrs 1 & 2 to the hold reg.   */
            C = 0xa3;
/* Access the hold reg for cntr1 */
            C = 0x11;
/* Obtain the count value from cntr1   */
            *lo1 = D;
            *hi1 = D;
/* Access the hold reg for cntr2   */
            C = 0x12;
/* Obtain the count value from cntr2   */
            *lo2 = D;
            *hi2 = D;
            ecntr1[num_times] = ((*hi1 << 8) + *lo1);
            ecntr2[num_times] = ((*hi2 << 8) + *lo2);

            elapsed_time[(num_times-1)] =
               (ecntr2[num_times] - scntr2[num_times])
               * 10 + ((ecntr1[num_times]
               - scntr1[num_times])*10)/10;

            printf("\nelapsed time 0  = %d",
               elapsed_time[(num_times-1)]);
            printf("\nscntr1 = %5d\tecntr1 = %5d",
               scntr1[num_times],ecntr1[num_times]);
            printf("\tscntr2 = %5d\tecntr2 = %5d",
               scntr2[num_times],ecntr2[num_times]);
```

```
            collecting = 0;
            goto next_loop;
         }

         else if ( ptr >= lookback){
            for (ptr = start; ptr < lookback; ptr++){
/* Copy the lookback region along with
   the data.
*/
               *ptr = ptr[1];
            }
            ptr--;
            if (kbhit() != 0){
               key_pressed = keybd_getkey();

/* 0x1b is escape,Stop Sampling
*/
                 if ( key_pressed == 0x01b){
                    --num_times;
                     goto done;
                 }
            }
         }
      }

      }
next_loop:
/* Start another conversion to keep the Tecmar A/D
   board active.
*/
      STCONV();

   }   while ( num_times < max_times); /* END DO Loop */

done:
   num_bytes = num_times * num_per_ker;


/* Subtract the intercept value of the load cell from
   all of the readings from the load cell.
*/
   for(ptr=&data[0];ptr<&data[(num_times*num_per_ker)];
     ++ptr)
      *ptr -= intercept;


   printf("\nnum of times = %5d\n",num_times);

   for ( i=0; i<num_times;++i){
      printf("0 - st1 = %5u\tst2 = %5u\ten1 = %5u",
```

```
            scntrl[i],scntr2[i],ecntrl[i]);
            printf("\ten2 = %5u\n",ecntr2[i]);

      }

      prints(18,10,FLASH|fgR,"Writing to the file");
      scr_spos(18,40);

   /*   Time to write the results to the file          */
      if ( num_times != 0)
         write_file();

   /*   Switch Back to the original display page       */
      scr_spg(disp_pg-2,write_pg-2);
   }



   startcntr(count)
   unsigned char   count;
   {

   /* This sets the gain of the Tecmar A/D converter to
      again of 100, and turns off the auto increment
      mode of accessing the channels on the Tecmar A/D
      board.
   */
      ADCNTROL(GAIN100|AUINCOFF);

   /* This selects the proper channel to monitor the load
      cell.
   */
      ADCHAN(CHANNEL);

   /*   Master Mode reset   */
      C = 0xff;
   /* Get into the master mode register   */
      C = 23;

      D = 0xc0;

      D = 0xc1;
   /* Select mode register for counter #1   */
      C = 0x01;

      D = 0x22;

      D = (C_F2 >> 8);
   /* Select mode register for counter #2   */
      C = 0x02;
```

```
   D = 0x28;

   D = 0x00;

/* Select the load register for counter #1   */
   C = 0x09;
/* Load counter #1 with the count of 10    */
   D = (count & 0xff);

   D = 0x00;
/* Select the load register for counter #2   */
   C = 0x0a;
/* Load conter #2 with a count of 0    */
   D = 0x00;

   D = 0x00;

/* Disarm counters #1 & #2      */
   C = 0xC7;
/* Load counters from Either the load or hold registers
   as specified in the setting of each one's mode.
*/
   C = 0x47;
/*   Arm both counters          */
   C = 0x27;


}

/**/
```

```
/* Filename:   WRITREAD.C
   Program Name(s):write_file(),read_file(),printit(),
                   disk_full(),printenc()

   Description :   write_file() - Writes the values
                   stored in the array 'data' to the
                    file.

                   read_file()  - Reads a file from
                    the diskette into the array 'data'
                    allowing the user to view the
                    breakage event.

                   printit()    - This prints the
                    values of the 'data' file to the
                    screen.

                   disk_full()  - Display a message
                    to the user that the disk is full.

                   printenc()   - This prints encoder
                   readings to the screen.

   Written by Paul J. Barry
*/
#include <stdio.h>
#include "wheat.h"
#include "display.h"
#include "string.h"
#include <stdlib.h>
#include <io.h>


/* Set up the structure which contains the pertinent
   information for a particular data collection phase.
*/
typedef struct{
      int    magic_no,time_interval,
         intercept,num_per_ker,
         numerator,denominator,bytes,thresh;

#ifdef NEWER
      float   rpm;
#else
      int   rpm;
#endif
      char      blade_types[30];
#ifdef NEWER
      float   x1;
      float   x2;
```

```
#endif

}       header;

header    hin;

int       rpm2;
write_file()
{
   FILE   *fopen(),*fp;

   int       i,staying;
   char       write_filename[15];
   char       buffer[17];

   hin.magic_no = 1;
   hin.num_per_ker = num_per_ker;
   hin.time_interval = time_interval;
   hin.intercept = intercept;
#ifdef NEWER
   hin.rpm = rpm;
#else
   hin.rpm = rpm2;
#endif
   hin.bytes = num_times;
   strcpy(hin.blade_types,blade_type);
   hin.thresh = thresh_hld;

#ifdef NEWER
   hin.x1 = x1enter;
   hin.x2 = x2enter;
#endif
   ++files_saved;

   staying = TRUE;

   while ( staying ){

/* This section of code appends the number of files
   saved under a configuration.
*/
      strcpy(write_filename,file_name);

      if ( files_saved < 10 ){
         strcat(write_filename,".00");
         itoa(files_saved,buffer,10);
         strncat(write_filename,buffer,1);
      }
      else if ( files_saved < 99 ) {
         strcat(write_filename,".0");
```

```
         itoa(files_saved,buffer,10);
         strncat(write_filename,buffer,2);
      }
      else{
         itoa(files_saved,buffer,10);
         strncat(write_filename,buffer,3);
      }

      if ( access(write_filename,0) == 0 )
         ++files_saved;
      else
         staying = FALSE;
   }

/* Check for errors in writing out the information to
   the files
*/
   if ( (i = fcloseall()) == EOF){
      printf("Error in closing all streams\n");
   }

/* Make sure that a new file can be opened for data
   collection.
*/
   if ((fp = fopen(write_filename,"wb")) == NULL )
      printf("Error in opening %s",write_filename);


/* Split the floating point number, bits_volt, into a
   numerator and denominator and store these values
   into integer variables for storage in the file.
*/
   hin.numerator = (int) bits_volt;
   hin.denominator = (int)((bits_volt-hin.numerator)
                            *100);

/* Store all of the header information in the file.
*/
   if (fwrite((char *)&hin,sizeof(header),1,fp) != 1){
      printf("\nError in writing out the header!\n");
   }

/* Write out the position of the 12-bit Absolute Encoder
   readings to the file.
*/
   if (fwrite((char *)posit,sizeof(posit[0]),num_times,
       fp)!= num_times){
      printf("\nError in writing out positions!\n");
   }
```

```
/* Store the starting values of counter #1 for each
   kernel in the file.
*/
   if (fwrite((char *)scntr1,sizeof(scntr1[0]),
       num_times,fp)!= num_times){
      printf("\nError in writing out scntr1!\n");
   }

/* Store the starting values of counter #2 for each
   kernel in the file.
*/
   if (fwrite((char *)scntr2,sizeof(scntr2[0]),
       num_times,fp)!= num_times){
      printf("\nError in writing out scntr2!\n");
   }

/* Store the ending values of counter #1 for each kernel
   in the file.
*/
   if (fwrite((char *)ecntr1,sizeof(ecntr1[0]),
       num_times,fp)!= num_times){
      printf("\nError in writing out ecntr1!\n");
   }

/* Store the ending values of counter #2 for each kernel
   in the file.
*/
   if (fwrite((char *)ecntr2,sizeof(ecntr2[0]),
         num_times,fp) != num_times){
      printf("\nError in writing out ecntr2!\n");
   }

/* Record the elapsed time to crush each kernel in the
   file.
*/
   if (fwrite((char *)elapsed_time,sizeof(
       elapsed_time[0]),num_times,fp)!= num_times){
      printf("\nError in writing out elapsed time!\n");
   }

/* Finally store all of the data collected for each
   kernel into the file.
*/
   if (fwrite((char *)data,sizeof(data[0]),num_bytes,fp)
                  !=num_bytes){
      printf("\nError in writing data");
   }

   keybd_flush();
```

```
/* Close the file and make sure that all of the data in
   the disk file buffer is flushed to the diskette.
*/
   fclose(fp);

   keybd_flush();


}

int   max_pts_read,posits_read,red1,red2,red3,red4;

read_file()
{
   int      keep_track,maxsize,i,value,starting;
   int      ending,posit_ptr;
   int      p_read;
   FILE      *fopen(),*fp;

   scr_clr(INTENS|fgWH);

/* Enter the filename to be retrieved from the diskette.
*/
   prints(8,8,INTENS|fgWH,
            "Input filename to read:    ");
   scanf("%s",file_name);

/* Make sure that the file does actually exist on the
   specified diskette.
*/
   if ( (fp=fopen(file_name,"r+b")) == NULL ){
      printf("\nCannot open a %s for ",file_name);
      printf("reading\n");
      timeout(1L);
      return;
   }
   prints(10,8,FLASH|fgR,"Reading data file ");
   printf(" %s",file_name);


/* Read in the header information.
*/
   if ( fread((char *)&hin,sizeof(header),1,fp) !=1){
      printf("\nError in reading header");
      printf(" information!\n");
   }

/* Read in the 12-bit Absolute Encoder Positions from
   the file.
*/
```

```
   posits_read = fread((char *)posit,
            sizeof(posit[0]),hin.bytes,fp);

/* Calculate the bits per volt from the two previously
   stored integers.
*/
   bits_volt = (float) (hin.numerator +
               hin.denominator/100.0);

/* Read in the starting counter values for counter #1
*/
   red1 = fread((char *)scntr1,
          sizeof(scntr1[0]),hin.bytes,fp);

/* Read in the starting counter values for counter #2
*/
   red2 = fread((char *)scntr2,
          sizeof(scntr2[0]),hin.bytes,fp) ;

/* Read in the ending counter values for counter #1
*/
   red3 = fread((char *)ecntr1,
          sizeof(ecntr1[0]),hin.bytes,fp) ;

/* Read in the ending counter values for counter #2
*/
   red4 = fread((char *)ecntr2,
          sizeof(ecntr2[0]),hin.bytes,fp);

/* Read in the elapsed time to crush the kernels
*/
   p_read = fread((char *)elapsed_time,
          sizeof(elapsed_time[0]),hin.bytes,fp);

/* Read in the actual data values stored for each
   kernel.
*/
   max_pts_read = fread((char *)data,sizeof(data[0]),
            BUFFSIZE,fp);

/* Close the data file
*/
   fclose(fp);

   prints(10,8,INTENS|fgR,"Reading data file ");
   printf("\nPoints Read = %d\n",max_pts_read);

/* Prompt the user for a range of data values to print
   on the screen.
*/
```

```
   printf("\n\nEnter Starting value : ");
   scanf("%d",&starting);
   printf("\nEnter Ending value :  ");
   scanf("%d",&ending);

   scr_clr(INTENS|fgWH);
   scr_spos(10,0);

/* Set all of the header values from those read in from
   the file.
*/
   num_per_ker = hin.num_per_ker;
   time_interval = hin.time_interval;
   intercept = hin.intercept;
#ifdef NEWER
   rpm = hin.rpm;
#else
   rpm2 = hin.rpm;
#endif
   num_bytes = hin.bytes;
   strcpy(blade_type,hin.blade_types);
   thresh_hld = hin.thresh;


/* Print out the header values on the screen.
*/
   printf("\nFor time_interval = %10d",
      time_interval);
   printf("\t num_per_ker = %10d\n",num_per_ker);
   printf("starting = %5d\tending = %5d",
      starting,ending);
   printf("\tmax_pts_read = %5d\n",max_pts_read);
   printf("rpm = %8.2f\n",rpm);
   printf("blade type = %s\n",blade_type);
   printf("thresh - %d\n",thresh_hld);
#ifdef NEWER
   printf("x1 = %f\n",hin.x1);
   printf("x2 = %f\n",hin.x2);
#endif


/* Print out the position along with the encoder, and
   force reading for a particular kernel chosen above.
*/
   for ( i = starting,posit_ptr=1,keep_track=-1;
       i <= ending && i <= max_pts_read; ++i){
      ++keep_track;
      if (keep_track == hin.num_per_ker ){
         ++posit_ptr;
         keep_track = -1;
```

```
     }

     printf("%4d - \tencod = %6x\tbit = %4d",
        i,posit[posit_ptr],data[i]);
     printf("\tForce(lbs) = %10.2f\n",
        (data[i]/bits_volt));

   }

/* Pause for a brief moment,~1 second, before prompting
   the user to continue.
*/
   timeout(1L);
   scroll(SCROLL_UP,22,0,24,79,INTENS|fgB,2);
   prints(23,10,INTENS|fgR,"Hit any key to continue ");
   scr_spos(23,60);

   while ( !kbhit() );
   keybd_flush();
}


void
disk_full()
{
/* Switch to a different output screen before informing
   the user that the diskette is full.
*/
   scr_spg(disp_pg+2,write_pg+2);
   scr_clr(INTENS|fgWH);
   prints(10,10,INTENS|fgWH,
      "Disk is full please insert another");
   prints(11,10,INTENS|fgWH,"one and hit return");
   scr_spos(11,50);


   while ( !kbhit() ) ;
   keybd_flush();

/* Return to the original screen.
*/
   scr_spg(disp_pg-2,write_pg-2);

}

printit()
{
   int keep_track,i,maxsize,posit_ptr,startit,endpoint;

/* Pop into a new screen for the print routine.
```

```
*/
   scr_spg(disp_pg+3,write_pg+3);
   scr_clr(INTENS|fgR);

/* Enter the starting point to view along with the
   ending value from the 'data' array which contains
   the crushed kernels breaking force values.
*/
   prints(14,10,INTENS|fgB,"Enter Starting Point: ");
   scanf("%d",&startit);

   prints(16,10,INTENS|fgR,"Enter Ending Point  : ");
   scanf("%d",&endpoint);


/* Prevent any subscripts out of range errors from
   occuring.
*/
   if ( startit < 0)
      startit = 0;

/* Determine the maximum number of bytes to print on the
   screen.
*/
   maxsize = (num_bytes == 0) ? max_pts_read:num_bytes;

   printf("max_pts = %5d\tnum_bytes = %5d\n",
      max_pts_read, num_bytes);
   printf("maxsize = %5\n",maxsize);
   printf("endpoint = %5d\tstarting point = %5d\n",
      endpoint,startit);

/* Prevent access to the array outside the boudaries
   once again.
*/
   if ( (endpoint > max_pts_read)&&(max_pts_read!=0))
      endpoint = max_pts_read;

   printf("num_per_ker = %d\n",num_per_ker);

/* Print out the values from the 'data' array to the
   screen.
*/
   for(i=startit,keep_track=-1,posit_ptr=1;
      i<endpoint&&i<maxsize; ++i){
      ++keep_track;
      if ( keep_track == num_per_ker ) {
         ++posit_ptr;
         keep_track = -1;
      }
```

```
      printf("%4d - \tenc = %5d\tbit = %4d",
         i,posit[posit_ptr],data[i]);
      printf(\tForce(lbs) = %10.2f\n",
         (data[i]/bits_volt));
   }

/* Pause briefly to allow the user to view the last
   element printed on the screen before prompting
   the user to continue.
*/
   timeout(1L);
   scroll(SCROLL_UP,22,0,24,79,INTENS|fgG,2);
   prints(23,10,INTENS|fgR,"Hit any key to continue");
   scr_spos(23,40);


   while ( !kbhit() );
   keybd_flush();
   scr_spg(disp_pg-3,write_pg-3);

   }


printenc()
{
      int    i,posit_ptr=1,y;

      printf("\n");
/* Print out the 12-bit Absolute Encoder values to
   the screen.
*/
      for(i=0; i < (num_bytes/ num_per_ker); ++i){
         printf("# %5d-----\n\n",i);
         for ( y=0 ; y < 4; ++y){
            printf("pos = %5d",i);
            printf("\tencoder = %10x",
               posit[i][y]);
            printf(" ( hex )\t%5d\n",
               posit[i][y]);
         }
      }

/* Print out the starting and ending counts for each of
   the two counters on the screen.
*/
      for(i=0; i < (num_bytes/num_per_ker); ++i){
         printf("Elapsed Time ker #%5d = 0 %5d\n",i,
         elapsed_time[i]);
         printf("scl = %5d\tecl = %5d",
            scntrl[i],ecntrl[i]);
```

```
            printf("\tsc2 = %5d\tec2 = %5d\n",
               scntr2[i],ecntr2[i]);
         }

         keybd_flush();
         printf("Waiting for a keyboard hit!\n");

         while (!kbhit() );
         keybd_flush();

}

/**/
```

```
#include <stdio.h>
#include <dos.h>

/* Filename:   timing.c
   Program Name:   new_location()

   Description:   If a disk becomes full, then this
                  routine will find a track and
                  sector which isn't being used at this
                  time.

   Written by Paul J. Barry
*/
new_location() {

   int          i,track=5,side=0,drive,sector;
   unsigned int buffer[512];
   union REGS   rin,rout;
   struct SREGS sreg;

   for ( i = 1; i < 510; i++) {
      buffer[i] = 5;
   }

   /* Get the actual segment registers   */
   segread(&sreg);

   sector = 1;

   for( drive = 0; drive <= 1; drive++) {
      for (track = 1; track <=39; track++) {
              rin.x.bx = buffer;
         rin.x.dx = ( side << 8) | drive;
         rin.x.cx = ( track << 8) | sector;
         rin.x.ax = ( 3 << 8) | 9;
         int86x(19,&rin,&rout,&sreg);
      }
   }
}
```

```
/* Filename:   buff.c
   Function Name(s):buffer(),key_getc()

   Description:   buffer() -
                  This function buffers the user's
                  input for a numerical value.  This
                  routine is a little more strict in
                  checking that a proper number was
                  entered.
                   (Original Source Code:
                    Mike Lasch.)

                  key_getc() -
                  This routine gets a key from
                  the keyboard buffer and flushes
                  the keyboard buffer once it is
                  done!

                  Written by Paul J. Barry

*/
#include <stdio.h>
#include "display.h"
#include <tecmar.h>
#include "wheat.h"
#include <ctype.h>
#include <stdlib.h>
#include <dos.h>


int
buffer()
{

   char buffer[16];
   int index, key, flag, frac, num,i;
   int   row,col,irow,icol;

/* Initialize the buffer with end of string NULL
   markers.
*/
   for (index = 0; index <=15; index++)
      buffer[index] ='\0';

   index = 0;

   scr_gpos(&row,&col);
   do {

      key = key_getc();
```

```
/* Mask off any unwanted sign bits.
*/
      key &= 0xff;

/* when escape is pressed, the buffer is filled with
   spaces and the index is reset to the beginning of
   of the buffer.
*/
      if (key == '\033') {
         index = 0;

/* fill the buffer with spaces.
*/
         for (i = 0; i <=14; i++)
            buffer[i] =0x20;

         buffer[15] = '\0';
         scr_spos(row,col);
      }
/* act on a backspace being pressed
*/
      else if (key == '\b') {
         if (index > 0 )
            buffer[index-1] = ' ';
         buffer[index] = '\0';
         --index;
         if ( index < 0)
            index = 0;
         scr_spos(row,col + index);
      }
/* if the buffer is filled, then the index will remain at
   the N - 1 element to keep the subscripting within the
   bounds of 'buffer'.
*/
      else if (index >= 15) {
         index = 14;
         buffer[15] = '\0';

      }
/* check for a valid digit
*/
      else if ( isdigit(key) != 0) {
         buffer[index] = key;
         index++;
      }
      if ( key != '\r')
         prints(row,col,INTENS|fgWH,buffer);
      scr_spos(row,col+index);
/* Wait for a carriage return to end the valid number.
*/
```

```
   } while ( key != '\r' );

/* Place a null character at the end of the buffer to
   denote the end of the string.
*/
   buffer[index]='\0';


/* Convert the ascii value to an integer value.
*/
   num = atoi(buffer);

   return(num);
}




int
key_getc()
{
   union REGS   rin,rout;
   unsigned int combo;
   int      value;

   rin.h.ah = 0;
/* Rom Bios Call 0x16 which clears the keyboard buffer
   and records a key press in al.
*/
   int86(0x16,&rin,&rout);


/* The character is placed in the low byte of the ax
   register.
 */
   value = rout.h.al;


   return(value);

}

/**/
```

```
 /* Filename:    buzzcool.c
    Program name: buzz()

    Function: This routine simply beeps the speaker for
              a duration of 10 ticks from the onboard
              clock.  The divisor for the count was
              chosen to be 1400, which can be altered
              in order to obtain another frequency from
              the speaker.

              In changing the code, make sure that the
              old value from the address 0x61 is saved
              and &'ed with you new 3 or 0 value, or else
              the keyboard will mysteriously lock up!

              The addresses can be checked in Peter
              Norton's,"Guide to the IBM PC's."

              This program is courtesy of Paul Barry
              and was kind of debugged by Mike Schwarz!

   Written by Paul J. Barry
*/
#include <stdio.h>
#include <dos.h>



buzz(freq,duration)
register unsigned int   freq, duration;
{
/* input and output registers */
   union RECS   rin,rout;
   unsigned int new_time,portno,count,old_time,chk_time;
   unsigned char value,old_port;
   unsigned long magic_no=1193280;

/* freq is the frequency divisor            */
   count = magic_no / freq;

/* Load the counter with the value of count in low byte,
   high byte form.
*/
   outp(0x43,0xb6);
   outp(0x42,count);
   outp(0x42,(count >> 8));

/* Read in the old value from the port */
   old_port=inp(0x61);
```

```
/* Turn the speaker on */
   outp(0x61,(old_port|0x0003));

   rin.x.ax = ( 0 << 8 );
   int86(0x1a,&rin,&rout);
/* old_time stores the original time   */
   old_time = rout.x.dx;
/* The chk_time will be the starting time plus the
    duration of the beep.
*/
   chk_time = old_time + duration;



/* Loop until the time specified by duration has
   elapsed.
*/
   while ( (new_time=rout.x.dx) < chk_time )
      int86(0x1a,&rin,&rout);

/* Restore the old port value and turn off the speaker.
*/
   outp(0x61,(old_port));

}
```

APPENDIX C: DATA ANALYSIS PROGRAM

The program, analyzzz.c, is the analysis program used to generate the maximum force readings for each of the files analyzed. The listing of the program follows.

```
/* Filename:   analyzzz.c
   Program Name(s):main(),getoptions(),printit()

   Description:   main() - This program is the main
                  driver program for the analysis
                  program.

                  getoptions() - Parses the command
                  line to set options in the printout
                  phase of the main() program.

                  printit() - This prints out the
                  values which are set to TRUE by the
                  the command line options.

   Written by Paul J. Barry
*/
#include <stdio.h>
#include <display.h>
#include "string.h"
#include <stdlib.h>
#include <math.h>


#define DIVISOR 19.011
#define BUFFERSIZE   8000

#define TRUE   1
#define FALSE    0

/* This structure defines the header information stored
   in each of the data files before the actual breakage
   events.
*/
typedef struct{
   int   magic_no,time_interval,intercept,
   int   num_per_ker,numerator,denominator,
   int   bytes,thresh; .

#ifdef NEWER
      float   rpm;
#else
      int   rpm;
#endif
      char   blade_types[30];
#ifdef NEWER
      float   x1;
      float   x2;
#endif
#ifdef OLDER
```

```
      int   x1;
      int   x2;
#endif
}      header;

header   hin,*h;

int   posit[500][4];/* Position of the encoder  */
int   data[8000];   /* Breakage event storage   */
int   elapse[1000]; /* Elapse time of the crush */
int   scntr1[200];  /* Start of counter #1      */
int   scntr2[200];  /* Start of counter #2      */
int   ecntr1[200];  /* End of counter #1        */
int   ecntr2[200];  /* End of counter #2        */
char   fileout[15]; /* Output filename          */

float   time_interval;/* Time interval of the crush */

long    sum;   /* Sum of total area under the curve */
long   sumc;   /* Sum of the area to the cross-over */
long    sumd;  /* Sum of the area to the intercept  */
char   *ptr;   /* Temporary pointer variable        */

int   max_pts_read,posits,neg_inflect,pos_inflect;
int   ptsread;
long   starting;
int   numpts,num_times,max,maxpos,endpt;
int   cross;
int   numb_pts = BUFFERSIZE;


int   areatocross; /* area to cross over 0 force     */
int   areaunder;   /* area under the curve           */
int   bactsl;      /* back slope with bactnum slopes */
int   bactnum;     /* # of slopes to average         */
int   bavesl;      /* flag for an averaged back slope*/
int   bavenum;     /* # of pts between slope         */
int   firstder;    /* flag for first derivatives     */
int   firstpts;    /* # of pts on either side of peak*/
int   second_der;  /* flag for second derivatives    */
int   secondpts;   /* # of pts of either side of peak*/
int   time_to_thre;/* flag for time to threshhold    */
int   time_to_peak;/* flag for time to peak          */
int   factsl;      /* flag for front slope ave.      */
int   factnum;     /*  number of pts averaged over   */
int   favesl;      /* flag for front slope ave.      */
int   favenum;     /*  number of pts between values  */
int   localmin;    /* flag for local minimum's       */
int   minpts;      /*  number of pts to consider     */
int   localmax;    /* flag for local maximum's       */
```

```
int   maxpts;      /*  number of pts to consider     */
int   forces;      /* flag to print out forces       */
int   logdec;      /* flag for logarithmic decrement */
int   maxforce;    /* flag to print maximum force    */
int   ratio_fb_sl; /* flag for ratio of fr/bk slope  */
int   fb_sl_pts;   /*  number of pts to calculate    */
int   ratio_time;  /* flag for time ratio            */
int   debug;       /* flag for debugging the program */
int   printenc;    /* flag to print encoder values   */
int   printela;    /* flag to print elapsed time     */
int   pheader;     /* print header information       */
int   flag;        /* General Boolean flag           */
int   *st = &data[0];/* Pointer to 'data'            */
int   *d;          /* Index ptr into the array 'data'*/
int   maxf;        /* Integer maximum force value    */
int   maxpos;      /* Position of maximum force value*/
int   localminima; /* number of local minima         */
int   localmaxima; /* number of local maxima         */
int   readinb;     /* number of elapsed times read   */
float   slope;     /* temporary slope calculation    */

main(argc,argv)
int   argc;
char   *argv[];
{
   register int   i,j,k;
   FILE      *fopen(),*fp,*fout;

   h = &hin;

/* Get the options from the command line and set the
   appropriate flags.
*/
   getoptions(argc,argv);

/* Adjust the command line arguments so that the next
   argument is the data filename to be opened for
   reading.
*/
   for(i=0;i<(argc - 1);++i)
      *argv++;

/* If the 'debug' flag is set to TRUE, then the debugg-
   ing information will be displayed on the screen.
*/
   if ( debug)
      fprintf(stderr,"*argv = %s\n",*argv);

/* Open the data file for reading in the binary mode.
*/
```

```
   if ( (fp = fopen(*argv,"rb")) == NULL){
      fprintf(stderr,
         "\nCannot open %s for reading!\n",
         *argv);
      exit(0);
   }

/* Read in the header to obtain pertinent information
   about the file.
*/
   if ( fread((char *)&hin,sizeof(header),1,fp)!=1){
      printf("\nError in reading header");
      printf(" information!\n");
      exit(0);
   }

/* Read in the 12-bit Absolute Encoder readings from the
   file.
*/
   if ((ptsread = fread((char *)posit,sizeof(posit[0]),
      hin.bytes,fp))!= hin.bytes){
      fprintf(stderr,
       "\nError in reading encoder postions!\n");
      exit(0);
   }
#ifdef NEWER

/* Load the starting of counter #1 values into the
   array.
*/
   if (fread((char *)scntrl,sizeof(scntrl[0]),
      hin.bytes,fp) != hin.bytes){
      printf("\nError in reading out scntrl!\n");
   }

/* Read in the starting of counter #2 values into the
   'scntr2' array.
*/
   if (fread((char *)scntr2,sizeof(scntr2[0]),
      hin.bytes,fp)!= hin.bytes){
      printf("\nError in reading out scntr2!\n");
   }

/* Read the values of the ending counter #1 into
   'ecntrl' array.
*/
   if (fread((char *)ecntrl,sizeof(ecntrl[0]),
      hin.bytes,fp) != hin.bytes){
      printf("\nError in reading out ecntrl!\n");
   }
```

```
/* Read in the ending values of counter #2 in the array
   'ecntr2'.
*/
   if (fread((char *)ecntr2,sizeof(ecntr2[0]),hin.bytes,
      fp)!= hin.bytes){
      printf("\nError in reading out ecntr2!\n");
   }

#endif


/* Read in the elapsed times of each of the kernels into
   the 'elapse' array.
*/
   if ((readinb=fread((char *)elapse,sizeof(elapse[0]),
      hin.bytes,fp))!= hin.bytes){
      fprintf(stderr,
         "\nError in reading elapsed times!\n");
      exit(0);
   }

/* Read in the actual breakage event values for all of
   the individual breakage events within the file.
*/
   numpts=fread((char *)data,sizeof(data[0]),BUFFERSIZE,
      fp);

   if ( debug ){
      starting = ftell(fp);
      fprintf(stderr,"\nposition in file = %ld\n",
         starting);
      fprintf(stderr,"\nptsread posit = %d\n",ptsread);
      fprintf(stderr,"readinb elapse = %d\n",readinb);
      fprintf(stderr,"After reading data file ");
      fprintf(stderr,"\nPoints Read = %d\n",numpts);

   }


/* If the magic number is not set to 1, then the
   intercept of the load cell has not been subtracted
   from the readings and thus the readings need to be
   altered.
*/
   if ( hin.magic_no != 1){
      for( i=0,d = data; i < numpts; ++i)
         *d++ -= hin.intercept;
   }
```

```
/* Determine the maximum force reading and the position
   relative to the first sample taken.
*/
   for(d = &data[0],maxf = data[0],maxpos=d-st,sum=0L;
         d <= &data[numpts]; *d++){
      sum += *d + *(d + 1);
      if ( *d > maxf){
         maxf = *d;
         maxpos = d - st;
      }
   }

/* This next portion of code calculates the number of
   readings before the load cell returns to it's initial
   rest position.
*/
   for(d=&data[maxpos],sumd=0L;d<=&data[numpts];*d++){
      sumd += *d + *(d + 1);
      if ( *d < 0 ){
         cross = d - st;
         sumc = sumd;
         break;
      }
   }

   time_interval=(float)(elapse[0])/
      (hin.num_per_ker*10.0);

   printit();
   fprintf(stdout,"\n");

}



getoptions(argc,argv)
int   argc;
char   **argv;
{

   char      options[BUFSIZ];
   char      *op = options;
   char      c;
   int      i;
   int       atoi();

   *op = NULL;

   *argv++;
```

```
   --argc;

/* Copy all of the arguments into the 'options' array.
*/
   for(i=0; i < (argc - 1); ++i)
      strcat(op,argv[i]);


/* No options were given and therefore no work is
   necessary.
*/
   if ( *op == NULL){
      fprintf(stderr,
         "consult manual entry for analysis!\n");
      exit(0);
   }
/* There are valid options on the command line and now
   it is time to process them.
*/
   else
      while ( *op ){

/* Each one of the options is set to a Boolean TRUE if
   it occurs on the command line.
*/
         switch (*op++){

            case   'a':   areatocross = TRUE;
               break;

            case 'A':   areaunder = TRUE;
               break;

            case 'b':    bactsl = TRUE;
               bactnum = atoi(op);
               break;

            case 'B':    bavesl = TRUE;
               bavenum = atoi(op);
               break;

            case 'd':   firstder = TRUE;
               firstpts = atoi(op);
               break;

            case 'D':   second_der = TRUE;
               secondpts = atoi(op);
               break;

            case 'e':   printela = TRUE;
```

```
   break;

case 'E':   printenc = TRUE;
   break;

case 'f':   factsl = TRUE;
   factnum = atoi(op);
   break;

case 'F':    favesl = TRUE;
   favenum = atoi(op);
   break;

case 'h':   pheader = TRUE;
   break;

case 'i':   localmin = TRUE;
   minpts = atoi(op);
   break;

case 'I':   localmax = TRUE;
   maxpts = atoi(op);
   break;

case 'l':   forces = TRUE;
   break;

case 'L':   logdec = TRUE;
   break;

case 'M':   maxforce = TRUE;
   break;

case 'N':   numb_pts = atoi(op);
   break;

case 'r':   ratio_time = TRUE;
   break;

case 'Q':   debug = TRUE;
   break;

case 's':   ratio_fb_sl = TRUE;
   fb_sl_pts = atoi(op);
   break;

case 't':   time_to_thre = TRUE;
   break;

case 'T':   time_to_peak = TRUE;
```

```
               break;

/* Check for invalid */      default:   c = *(op - 1);
/* options on the    */         if (c != '.'&& c!='-'&&
/* command line      */            c <'0' || c > '9'){
                   fprintf(stderr,
                   "analysis:");
                   fprintf(stderr,
                   " illegal option");
                   fprintf(stderr,
                   " '%c'\n",c);
                   exit(0);
               }
          }/*   End of the switch statement  */

      }   /*    End of the while statement   */

/* If we are dubugging at this time, then print the
   status of all of the various option flags to check
   on proper settings from the command line.
*/
   if (debug){
     fprintf(stderr,"areatocross = %d\n",areatocross);
     fprintf(stderr,"areaunder = %d\n",areaunder);
     fprintf(stderr,"bactsl = %d\n",bactsl);
     fprintf(stderr,"bactnum = %dc\n",bactnum);
     fprintf(stderr,"bavesl = %d\n",bavesl);
     fprintf(stderr,"bavenum = %d\n",bavenum);
     fprintf(stderr,"firstder = %d\n",firstder);
     fprintf(stderr,"firstpts = %d\n",firstpts);
     fprintf(stderr,"second_der = %d\n",second_der);
     fprintf(stderr,"secondpts = %d\n",secondpts);
     fprintf(stderr,"time_to_thre = %d\n",time_to_thre);
     fprintf(stderr,"time_to_peak = %d\n",time_to_peak);
     fprintf(stderr,"factsl = %d\n",factsl);
     fprintf(stderr,"factnum = %d\n",factnum);
     fprintf(stderr,"favesl = %d\n",   favesl);
     fprintf(stderr,"favenum  = %d\n",favenum);
     fprintf(stderr,"localmin = %d\n",localmin);
     fprintf(stderr,"minpts = %d\n",   minpts);
     fprintf(stderr,"localmax  = %d\n",localmax);
     fprintf(stderr,"maxpts  = %d\n",maxpts);
     fprintf(stderr,"forces = %d\n",forces);
     fprintf(stderr,"logdec = %d\n",logdec);
     fprintf(stderr,"maxforce = %d\n",maxforce);
     fprintf(stderr,"ratio_fb_sl = %d\n",ratio_fb_sl);
     fprintf(stderr,"fb_sl_pts = %d\n",fb_sl_pts);
     fprintf(stderr,"ratio_time  = %d\n",ratio_time);
     fprintf(stderr,"debug = %d\n",debug);
```

```
   }

}      /* End of the section getoptions      */



printit()
{
   register int   i,j,k;

#ifdef OLDER
   fprintf(stdout," %8d  %8d",hin.xl,hin.x2);
#endif
#ifdef NEWER
   fprintf(stdout," %8.2f  %8.2f",hin.xl,hin.x2);
#endif

/* In all of the cases enclosed in the paranthesis
   proceeding the 'if' statement are executed if the
   condition (or flag) is set to the Boolean TRUE
   value.
*/

/* This prints out the area under the curve to point at
   which the load cell first collects the breakage event
   until the time it reaches the initial rest position.
*/
   if (areatocross)
      fprintf(stdout," %6.2f",(float)sumc/2.0*DIVISOR);

/* Again, debug information is printed.
*/
   if ( debug ){
      fprintf(stderr,"maxf = %10d\tmaxpos = %d\n",
         maxf,maxpos);
      fprintf(stderr,"cross = %10d\tsumc = %101d\n",
         cross,sumc);
   }


/* The force readings from the load cell are printed in
   the order of collection from the breakage event.
   (The breakage events are actually stored in the file
     as integer values from the load cell.)
*/
   if ( forces) {
      for ( d = &data[0]; d <= &data[numpts]; *d++)
         fprintf(stdout,"\n%.2f",
            (float) *d/DIVISOR);
```

```
    }


/* This prints out the maximum force reading from the
   breakage event.
*/
   if ( maxforce)
      fprintf(stdout," %6.2f", (float) maxf/DIVISOR);

/* The area under the curve is printed.
*/
   if ( areaunder)
      fprintf(stdout," %10.2f",
         (float) sum/2.0*DIVISOR);

/* This prints the actual front slope of the breakage
   event.  (The front slope refers to the part of the
   breakage event from the onset of collection of data
   until the peak force is reached.
*/
   if ( factsl ){
      slope = (float) (data[maxpos] -
            data[maxpos - factnum]);
      fprintf(stdout," %6.2f",(slope /
              (float) factnum));
   }

/* This is the front slope from an average number of
   front  slopes.
*/
   if ( favesl ){
      for (i=(maxpos-(favenum + 1)),slope=0.0;
         i<=maxpos;++i)
         slope += (float) (data[i] - data[i-1]);
      slope /= (float) favenum;
      fprintf(stdout," %6.2f",slope);
   }

/* This is the back slope pertaining to the slope of the
   breakage event from the maximum breaking force to the
   intercept of the load cell.
*/
   if ( bactsl ){
      slope = (float) (data[maxpos + bactnum] -
            data[maxpos]);
      fprintf(stdout," %6.2f",(slope/(float)bactnum));
      if ( debug)
         fprintf(stderr," %5d\t%5d\t%5d\n",
         data[maxpos + bactnum],
         data[maxpos],bactnum);
```

```
   }

/* This is an average of back slopes from the breakage
   event.
*/
   if ( bavesl ){
      for(i=maxpos,slope=0.0;i<=(maxpos+bactnum);++i)
         slope += (float) (data[i+1] - data[i]);
      slope /= (float) bavenum;
      fprintf(stdout," %6.2f",slope);
   }

/* This prints the number of local minima found in the
   breakage event curve.
*/
   if ( localmin ){
      for(i=(minpts +1),localminima =0;
          i<=((numpts-minpts)),i<=numb_pts;++i){
         for(j=1,flag = TRUE;j <= minpts ; ++j){
            if ( (data[i] >= data[i -j]) ||
               (data[i] >= data[i+j])){
               flag = FALSE;
            }
            if ( debug ){
               fprintf(stderr,
               "data (i) = %5d\t",
               data[i]);
               fprintf(stderr,
               " i-j = %5d\t",
               data[i-j]);
               fprintf(stderr,"i+j = ");
               fprintf(stderr,
               "%5d\tf = %d\n",data[i+j],
               flag);

            }
         }
         if ( flag )
            ++localminima;
         if ( debug)
            fprintf(stderr,"\tminflag = %2d",
               flag);
            fprintf(stderr,"\tc = %d\n",
               localminima);

      }

      fprintf(stdout," %8d ",localminima);

   }
```

```
/* This prints the number of local maxima occuring in
   the breakage event curve.
*/
   if ( localmax ){
      for(i=(maxpts +1),localmaxima =0;
         i<=((numpts-maxpts)),i<=numb_pts;++i){
         for(j=1,flag = TRUE;j <= maxpts ; ++j){
            if ( (data[i] <= data[i -j]) ||
               (data[i] <= data[i+j])){
               flag = FALSE;
            }
            if ( debug ){
              fprintf(stderr,"data (i) = %5d",
                 data[i]);
              fprintf(stderr,"\t i-j = %5d\t",
               data[i-j]);
              fprintf(stderr,"i+j = %5d\tf =",
                 data[i+j]);
              fprintf(stderr," %d\n",
               flag);

            }
         }
         if ( flag )
            ++localmaxima;
         if ( debug)
            fprintf(stderr,"\tminflag = %2d",
               flag);
            fprintf(stderr,"\tc = %d\n",
               localmaxima);

      }

     fprintf(stdout," %8d ",localmaxima);

   }

/* This prints the approximate time to the maximum
   breaking force.
*/
   if ( time_to_peak)
      fprintf(stdout," %6.2f",
         (float) maxpos * time_interval);

/* This prints the time required to achieve the load
   cell intercept from the maximum breaking force.
*/
   if ( time_to_thre)
      fprintf(stdout," %6.2f",
         (float) (cross-maxpos)*time_interval);
```

```
/* This prints the 12-bit Absolute encoder readings on
   the screen.
*/
   if ( printenc)
      for(i=0; i <= (ptsread - 1); ++i)
         for( j=0; j <=3; ++j)
            fprintf(stdout,"%d\n",posit[i][j]);


/* This prints out the elapsed time of the breakage
   event.
*/
   if (printela)
      for(i=0; i <= (readinb - 1); ++i)
         fprintf(stdout,"%d\n",elapse[i]);

/* This prints out the header information which is
   contained in each data file.
*/
   if (pheader){
      fprintf(stdout,"mag = %5d\ttim = %5d\tint =",
         h->magic_no,h->time_interval);
      fprintf(stdout," %5d\tnum =%5d\n",
         h->intercept,h->num_per_ker);
      fprintf(stdout,"numer = %5d\tden = %5d\t",
         h->numerator,h->denominator);
      fprintf(stdout,"byt = %5d\tthre = %5d\n",
         h->bytes,h->thresh);

      fprintf(stdout,"rpm = %5.2f\n",   h->rpm);
      fprintf(stdout,"blade type = %s\n",
         h->blade_types);
#ifndef OLD
      fprintf(stdout,"x1 = %5f\tx2 = %5f  ",
         hin.x1,hin.x2);
#endif
   }

}
```

ACKNOWLEDGEMENTS

I would like to dedicate this thesis to my family for all of their helpful support throughout the years, and especially to my father who gave me the courage to complete this degree.

I would also like to extend my thanks to my advisor, Dr. Steven Eckhoff, and to all the members on my committee: Dr. Do Sup Chung, Dr. Mark Schrock, Dr. Art Davis, and Dr. Jon Faubion. I thank each and everyone of them for helping in revisions of the text. I would also like to thank all of the faculty and staff in the Department of Agricultural Engineering especially Mike Schwarz, and Sheri Shanks.

Thanks also goes out to all of my fellow graduate students who have given me the best years of my life, and especially to Arnold Eilert for the extra effort spent in proofreading the thesis. A grateful thanks also goes out to Becky Rages for the countless number of hours spent driving me back and forth as well as all of the moral support.

HARDNESS DISCRIMINATION ABILITY OF THE KSU INDIVIDUAL WHEAT HARDNESS TESTER AS AFFECTED BY BLADE PENETRATION AND PLATE VELOCITY

by

PAUL JAMES BARRY

B.A., Benedictine College, 1984

AN ABSTRACT OF A THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

. College of Engineering
Department of Agricultural Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1989

In order to aid with the classification of wheat kernels as hard or soft, an instrument at Kansas State University was constructed jointly by the Agricultural Engineering and the Grain Science & Industry Departments. This instrument is known as the Single Kernel Wheat Hardness Tester (SKWHT). The sole purpose of this tester is to differentiate between hard and soft wheat kernels.

The SKWHT is an instrument which singularly slices each kernel in a batch sample and records the force versus time curve on an IBM PC Compatible computer. The data is analyzed shortly after collecting all the information on the wheat kernels and each kernel is then classified by the peak (maximum) force required to slice each kernel. The SKWHT is capable of slicing around 200 kernels per minute. There are also three different types of blades which can be interchanged in order to find the best cutting angle on the wheat kernels.

Two types of pencil leads were used to emulate the wheat kernels in the SKWHT before evaluating wheat. The two types of lead used were the 5H drafting pencil lead, and the Scripto Crayon Marking leads. These two types of leads were chosen due to the brittle and ductile breakage events. The 5H pencil lead emulated hard wheat and the Crayon lead emulated the soft wheat. Analysis of the results by the three different types of blades and different angular velocitites, and blade clearance values showed that the blunt blade along with greater blade penetration resulted in better delineation.

Surface response analysis of 5H versus Crayon Pencil lead indicated that the blunt blade was the best blade for discrimination between pencil leads. The optimum settings for Pencil lead discrimination with the blunt blade were an angular velocity of 1.354 rad/sec, and a blade clearance of 1.073 mm.

The two types of wheat used in this investigation were Mustang and Daws. The two varieties were tested on the sharp blade at three different moisture levels 9%, 10%, and 14% m.c. The optimum settings for the sharp blade were an angular velocity of 1.466 rad/s, and blade clearance of 0.667 mm.