A PERSONAL COMPUTER BASED INSTRUMENTATION SYSTEM FOR DETERMINING REAL-TIME DYNAMIC TORQUE IN ROTATING MACHINERY

by

Ratnakar M. Kanth

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

APPROVED:

Dr. Reginald G. Mitchiner, Chairman

Dr. Robert G. Leonard

Dr. Charles E. Knight

June 10, 1987

Blacksburg, Virginia

A PERSONAL COMPUTER BASED INSTRUMENTATION SYSTEM FOR DETERMINING REAL-TIME DYNAMIC TORQUE IN ROTATING MACHINERY

by

Ratnakar M. Kanth

Dr. Reginald G. Mitchiner, Chairman

Mechanical Engineering

(ABSTRACT)

Measurement of dynamic/transient torques is important in the dynamic analysis of rotating machinery as it provides insight into the internal state of the machine. Existing methods are difficult to implement, results are not obtained in real-time and are not very accurate.

This thesis introduces a new method of determining real-time dynamic torque. An optical encoder is used to sense motion at a convenient point in the rotating system containing the rigid shaft of interest. The encoder's output is processed digitally to yield angular velocity, acceleration and dynamic torque. Two different experiments were conducted to demonstrate the advantages of this new method of determining dynamic torque over conventional methods.

In one experiment, an extension spring was mounted on a crank arrangement coupled to a fractional horsepower motor to apply a periodic load to the system. A mathematical model of this dynamic system was developed to compare the results of this model with that of the instrumentation system. In another experiment, the instrumentation system was used on an existing motor-compressor system. The dynamic torque thus determined was again compared with the results of a simulation program.

In both the above experiments the evaluated dynamic torque and computed dynamic torque were within 5% of each other, demonstrating accuracy and reliability of this personal computer based dynamic torque determining system.

Acknowledgements

I would like to express my deep gratitude to the following people for their help and encouragement during my graduate studies:

Dr. R.G. Mitchiner, who served as the chairman of my advisory committee. I thank him for his patience, guidance, and support. I have never left his office without gaining at least one new piece of knowledge.

Dr. R.G. Leonard and Dr. C.E. Knight for serving on my advisory and examining committee. Their guidance and support have been invaluable. , mechanical engineering department, for his excellent help in setting-up the experimental apparatus.

My fellow graduate students, for their technical guidance and comradery. In particular, , , , , and .

Dr. and Mrs. Sumathisena of New Jersey, for making me feel at home in the United States and for their affectionate caretaking.

Finally, I am indebted to my parents for their unfailing love and support. I value their faith in my ability to achieve anything I can imagine.

Table of Contents

List of Illustrations

Chapter 1

Introduction

The day the wheel was invented came the need to estimate and measure torque. As historians have recorded, the wheel was invented between 3500 B.C. and 3000 B.C. But it was only in the 15th century that Leonardo Da Vinci (1452-1519) demonstrated his clear understanding of the concept by using it effectively in several of his famous inventions [12]. The concept of torque, or moment, as it is used in mechanics is the tendency of a force to rotate the body to which it is applied about a point or axis.

In machinery where a main rotor operates at moderate to high rotational speeds, and where there may be reciprocating masses attached to this rotor, there will exist torques acting about the axis of the rotor or shaft. These torques may be a result of the internal configuration of the machine or may be imposed upon the shaft by a connecting machine.

The total torque acting upon the rotor, at any instant, provides for the rotary motion of the shaft. Thus an experimental determination of that torque or the rotary dynamic state of the shaft, can provide insight into internal conditions of the machine under consideration.

The measurement of torque in rotating shafts is difficult. Over the years several methods have been formulated and used. One of the more common methods has been the measurement of torque in shafts using strain gages. But strain gages are generally not very sensitive. Often, to obtain a satisfactory signal, the shaft has to be slit to allow insertion of a torsion tube upon which the gages are mounted. This method has the undesirable effect of modifying the dynamic behavior of the rotating system. Further difficulties are encountered in extracting strain gage signals from rotating shafts without the introduction of undue distortion due, in the case of slip rings for example, to brush bounce and varying contact resistance [2].

Because of the complex nature of the measurement techniques and the apparatus required, such as in the case above, torque measurement applications have been restricted to universities and research laboratories. Nevertheless, industry has always had the need to measure torque accurately and without major/permanent modification to the rotating machinery. It is the aim of this investigation to present a new method of dynamic torque determination which is reliable and can be implemented with little or no modifications to an existing piece of rotating machinery.

The last decade has seen the advent of high resolution optical shaft encoders capable of sensing angular displacement of the order of 10^{-1} degrees. Fig. 1 shows the schematic of an optical shaft encoder. As the disc rotates, the transparent segments on the circumference of the disc come in-line with the light source. The light detector

which is on the other side of disc, detects the light coming through the transparent segment in the disc, producing a square wave electrical pulse. By processing these digital pulses and converting them to analog signals, the angular displacement of the encoder shaft per unit time and hence the angular velocity can be determined. By rigidly coupling the encoder shaft with the shaft of interest and analyzing the measured angular velocity, angular acceleration and dynamic torque of the rotating system can be evaluated.

To demonstrate the usefulness of this instrumentation scheme for real-time dynamic torque evaluation of rotating machinery, two experiments have been conducted.

A schematic diagram of the spring load experimental set-up is shown in Fig. 2. The electric motor is coupled with the shaft encoder. An extension spring has one of its ends fixed to the ground and the other end mounted on a flange on the encoder shaft. As the motor shaft rotates, so does the flange. The moving end of the spring is displaced as a function of angular displacement of the motor. Thus, the spring applies a periodic load on the motor.

In this investigation, an Encoder Products Co. optical shaft encoder is used. It provides two square wave electrical signals of constant pulse widths (with respect to angle of rotation) as the encoder shaft is rotated. One is a 1 PPR (pulses per revolution) signal which corresponds to the average speed of the rotating shaft and the other is a 600 PPR signal which corresponds to the angular velocity of the shaft (Fig. 1). These signals have to be refined before the personal computer can process them to obtain angular acceleration and dynamic torque. For this reason, the encoder output signals are input to a pre-processor which consists of a phase locked loop circuit and a F/V (Frequency to Voltage) conversion and active filter circuit. The re-

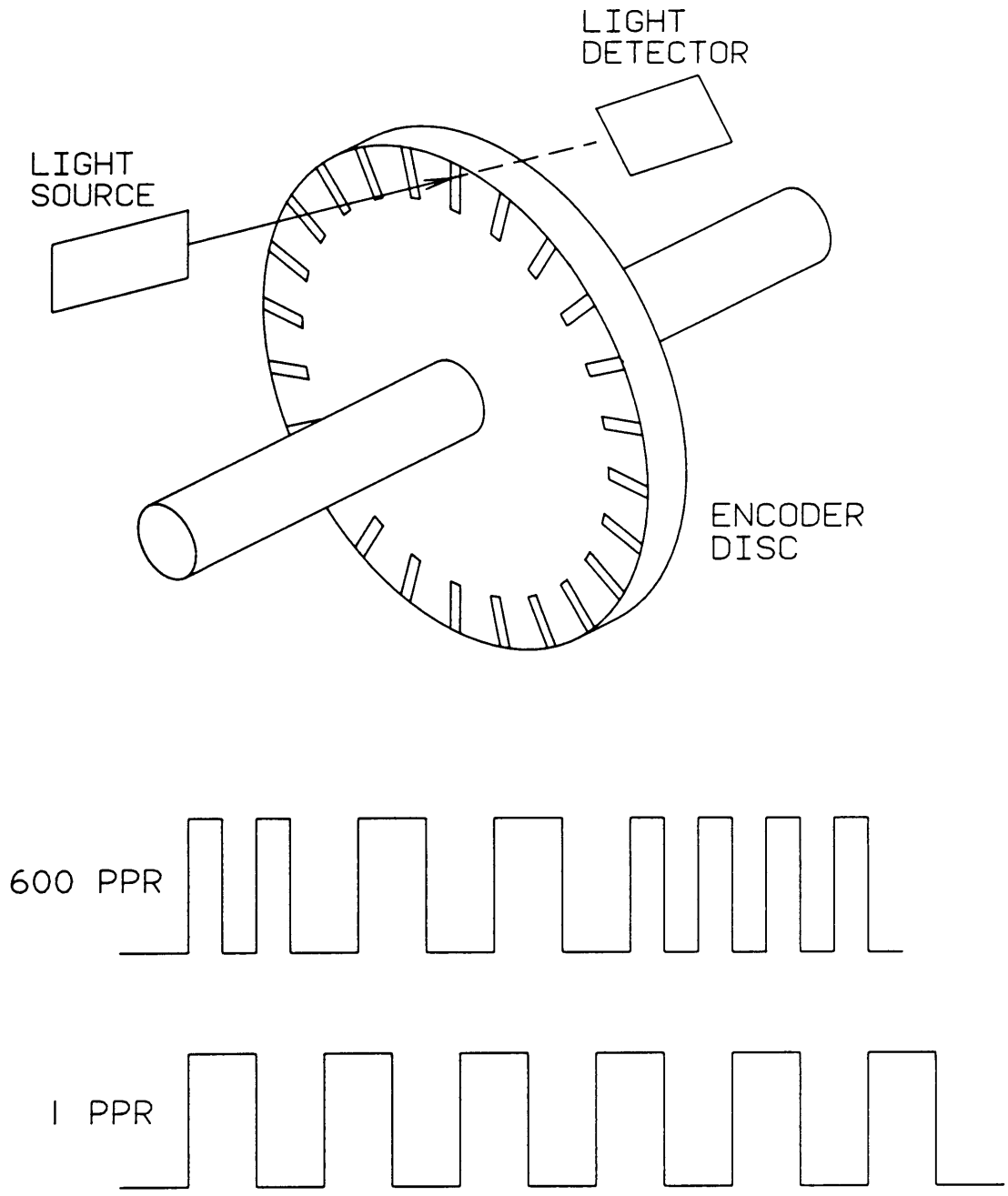


Figure 1. Schematic of an Optical Shaft Encoder

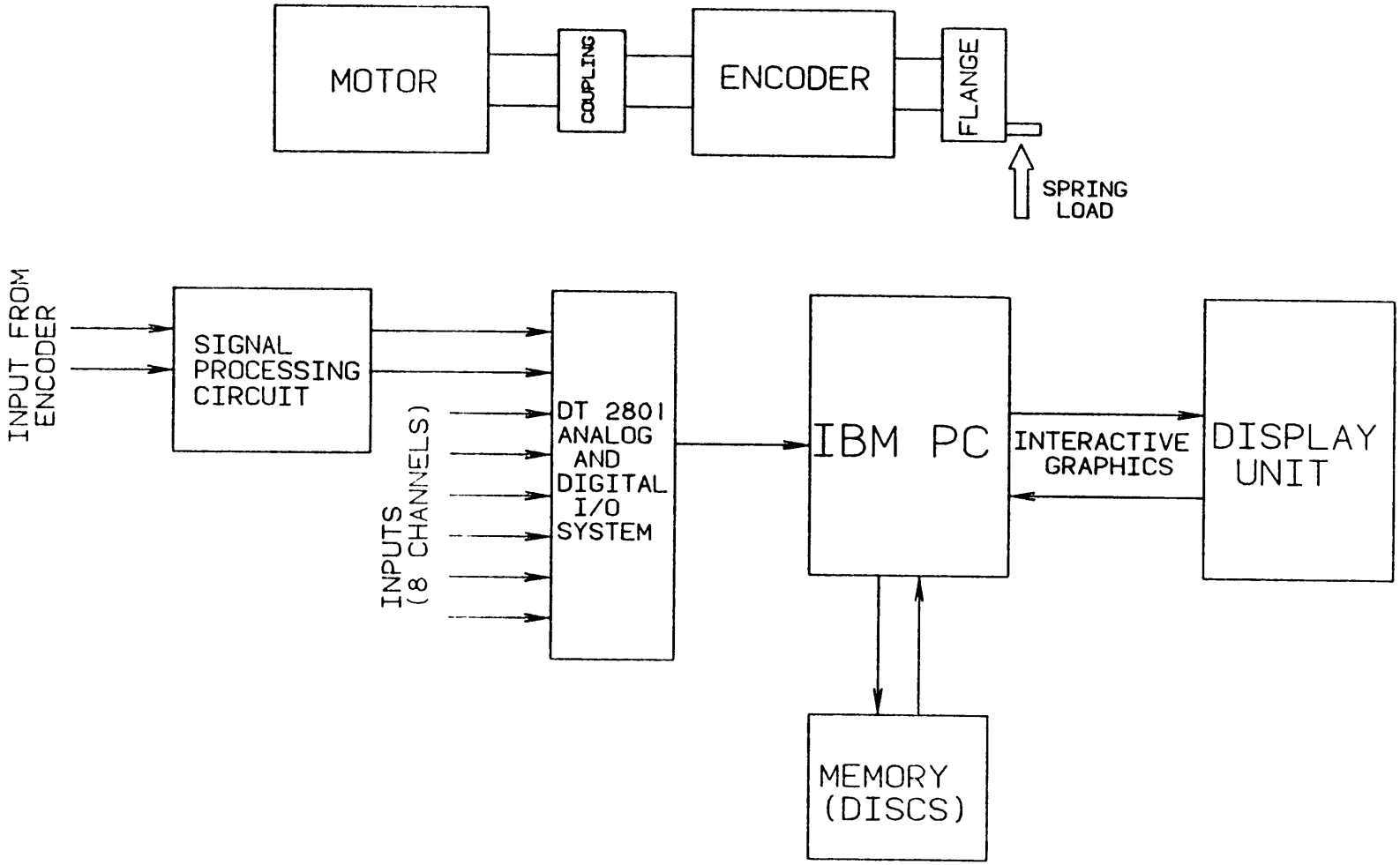


Figure 2. Schematic of the Spring Load Experiment

fined signals are then input to the Data Translation ™ DT2801-A board for A/D (Analog to Digital) conversion.

A FORTRAN 77 program was written to operate the data acquisition system and differentiate the measured velocity signal to obtain acceleration in the frequency domain using an FFT (Fast Fourier Transform) algorithm. Then the acceleration was multiplied by I, the mass moment of inertia of the rotating system, to obtain dynamic torque in the rotating system. The torque waveform thus obtained was compared with the torque waveform of a simulation program (theoretical model of the spring load experiment).

To demonstrate the use of shaft encoder in a practical application, experiments were conducted on an Ingersoll Rand ™ model IR242, a two-stage reciprocating air compressor. A schematic of the experimental set-up is shown in Fig. 3. The instrumentation system was implemented on this set-up by including the shaft encoder into the rotating system with the aid of a belt drive transmission system. A Baldor ™ 3 phase induction motor drives the compressor. The encoder shaft rotates at the same speed as the motor and compressor shafts. The data acquisition system and procedure used in this experiment were the same as in the spring load experiment. The mass moment of inertia of the compressor and motor were experimentally determined. The dynamic torque determined by this method was compared with the mathematical model developed by Mitchiner at VPI [21].

Chapter two reviews the literature in two distinct areas. First torque measurement in general and then development of shaft encoder technology. This is done in chronological order for both topics.

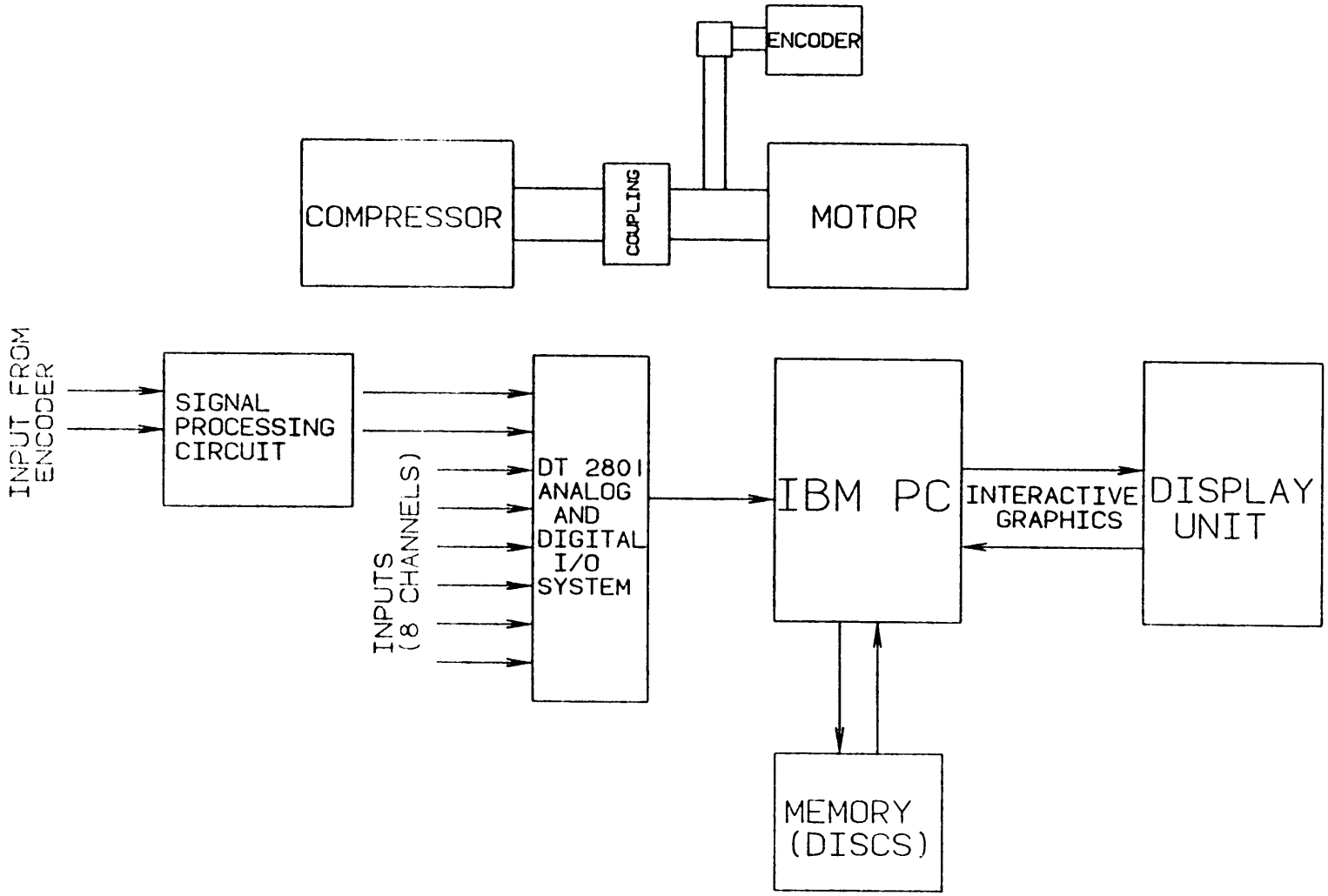


Figure 3. Schematic of the Compressor Experiment

Chapter three develops the theoretical basis for the experiments performed in this investigation. The spring load experiment and the compressor experiment are dealt with separately in different sections and the equations developed for the theoretical modelling of the these experiments are presented. This is followed by the presentation of computations required in the experiments such as determination of the mass moments of inertia of the rotating masses. The computed and measured experimental data are presented with the aid of plots.

Chapter four delineates the hardware used in the two experiments, i.e. the spring load experiment and the compressor experiment. Chapter five presents an overview of the commercial software used and a brief description of the torque measurement program and the simulation program for the spring load experiment. Chapter six discusses the results obtained, comparison with the theoretical model results and recommendations for future work.

Chapter 2

Literature Review

Because the purpose of this thesis is to demonstrate the use of a shaft encoder for accurate torque measurement in rotating machinery, this chapter is divided into two sections. The first section briefly describes the different methods of torque measurement that are in use today. The second section outlines the development in the technology of shaft encoders over the years and their use in the evaluation of dynamic torque in rotating machinery. The chapter concludes with a brief note on how the author's investigation compares with the literature.

2.1 Torque measurement methods in general

The strain gage torquemeter [22] method is one of the most popular types of torque transducers. It uses four metal-foil gages that are bonded in a bridge arrangement

on the surface of a torsion member. The bridge arrangement is made such that one pair of gages increases in resistance due to tensile strain while the other pair decreases in resistance due to compressive strain. This is one of the most commonly used methods of torque measurement by the automotive industry. Fleming [10] in his review of several torque measurement methods, states that the popularity of this method is due to the short shaft length required.

The dc cradled dynamometer is one of the most widely used devices for power and torque measurements on internal combustion engines, pumps, small steam turbines, and other mechanical equipment. In this method, a dc motor-generator stator is mounted on bearings with a moment arm extending from the stator body of the motor to a force-measurement device, often a pendulum scale, a spring scale, or a load cell. When the device is connected to a power-producing machine, it acts as a dc generator whose power output is dissipated in resistance racks. The torque impressed on the dynamometer is the product of the moment arm and the load [14].

An indication of the torque transmitted by a shaft can be obtained by measurement of twist angle, i.e., the torsional strain in the shaft. An optical shutter arrangement can provide for a digital measurement of the angular displacement due to torsional twist of a long solid shaft. A torquemeter based on this principle was used for dynamometer testing of transmission gearbox efficiencies by Craddock & Brittain [5]. Pratt [24] has proposed that this type of sensor would be suitable for use in engine control systems. Ward *et al* [29] have demonstrated that a highly accurate torquemeter of this type can be built provided proper attention to the details of construction is made. Another type of twist-angle torquemeter was developed for use with gas turbine engines by Ohigashi *et al* [23]. In this method the sensor utilizes

magnetic pickups to detect passage of gear teeth at both ends of a torsion bar. Rogers [25] has presented the construction and operation of a transducer which operates on the same principle. Further, the associated pickup and circuitry is also provided. The electronic circuitry has improved sensitivity and takes advantage of more recently developed detectors.

Utility of a magnetostrictive torque sensor has been demonstrated by its use on gas turbine aircraft engines. Scoppe [26] presents a product line of interchangeable sensors developed by Avco-Lycoming for onboard measurement of aircraft engine torque.

2.2 *Shaft encoders and torque measurement*

A shaft encoder is an electromechanical device which translates the angular rotational movement of a rotating shaft into a series of digital pulses, usually of a square waveform, which can be used to measure, monitor, or control a system. Shaft encoders are relatively new to the engineering scene. One of the first commercial shaft encoders appeared in 1969, marketed by Astrosystems, Inc. Wheatcroft [30] discusses the principle of operation and accuracy of measurement of this encoder. The disadvantage of this encoder system was that the shaft-encoder, made of solid-state devices, depended on an electromagnetic resolver which acted as the rotary transducer - coupled to the rotating shaft. The encoder would receive the signal from the rotary transducer and convert the rotary displacement into decimal or binary form

for readout or for use with computers, printers or other off-line equipment. Other disadvantages of this system were: it was comprised of two separate units viz. the rotary transducer and the encoder; and the encoder required elaborate calibration. However, it provided as high a resolution as 0.01°.

In 1971, Baasch [3], while the editor of *Electronics Products Magazine*, discussed some of the popular shaft encoders available at that time. There is a discussion of both absolute and incremental type encoders. Most encoders at that time were used mainly for precise linear or angular displacements such as the traverse of a milling machine table, motor control, radar tracking antenna, and so on.

Clarke and Parrot [4] were among the first authors in 1972 to write about optical shaft-angle encoders. According to them "the key to the successful performance of photo-electric encoders is in the manufacture of the coded glass discs." They discuss a machine that was developed to repeatedly produce accurately coded master discs which, in themselves, define the absolute maximum accuracy obtainable from an encoder. This machine, according to these authors, made possible the development of special code patterns to meet any specific requirements. The biggest drawback of these optical encoders was that, to increase the resolution, it was necessary to employ a larger diameter disc. This led to much development time being spent trying to reach a compromise between the two factors, size and resolution. In the same paper Clarke and Parrot propose a way of solving this problem: to use an optical resolver encoder which would give a resolution of 1 part in 20 (18°) and would have all amplification and code-conversion logic built in.

Midson [20] in his 1976 paper on shaft encoders has discussed the principle of operation of shaft encoders in detail. This paper is a good attempt to clarify the confusion

between the names at that time: shaft encoders, digitizers and rotary pulse generators. In addition to giving the salient features of both incremental and absolute encoder types, this paper also clearly identifies the applications in which the use of shaft encoders would be highly desirable, such as controlling table motion in NC machines, antenna position, motor speed control, etc.

Hu and Raudkivi [15] in their 1980 paper have given different classifications of shaft encoders based on type of sensors, function and coding schemes. This paper also proposes a new cyclic encoding scheme for an absolute digital encoder in units of degrees. In comparison with the then prevalent methods of encoding the angular displacement data, this method simplified the required external decoding circuitry while preserving the same coding efficiency. When higher resolution is needed, additional digits, representing fractions of a degree in this case, can also be added.

At the turn of the decade, industry had realized the usefulness of encoders, thus they found their way into several applications. But many of the applications required accurate bidirectional sensing of angular displacement. Kuchela [16] in 1981 came up with a solution for this problem. He presented a circuit based on a PROM (Programmable Read Only Memory) for separating up and down pulses from the outputs of an optical encoder. The logic of this circuit provides up and down counting pulses unambiguously and improves the resolution of the shaft encoder by a factor of four.

Shaft enocoders often find application in NC machines and other machine tools where stepper motors are used for accurate displacements. The shaft of a stepper motor often undergoes damped oscillations as the motor executes an angle step. Naval [21], in his 1985 paper, discusses how these vibrations can be bothersome when they cause the optical shaft encoder to generate multiple output pulses. To prevent erro-

neous readings, processing of the encoder outputs is necessary so that only one output pulse per step is issued, even in the presence of single-step oscillations. The author presents stalled rotor detection capability and an automatic sign change at zero crossover feature for displaying information in the sign-magnitude form.

The most relevant work to this investigation has been done in 1984 by Auckland *et al* [2] at the University of Manchester, UK. Their paper discusses a contactless method of measuring transient changes in the torque transmitted by a rotating shaft. These authors used an optical encoder as a sensing device of torsion in the shaft of interest, then used a Digital Equipment PDP 11E computer to record and manipulate the data. Specifically, they applied it to the measurement of transient torques in the drive shaft of a small motor-alternator set used to model practical turbogenerators. They compared the waveform from the encoder system with that of a conventional strain gauge instrument and have claimed that the results from the encoder system to be very consistent with the strain gauge measurement. However, the order of accuracy could not be calculated as the encoder system they developed was not calibrated to measure torque in force units, but only provided measurements in volts.

The instrumentation scheme introduced in the current investigation is more versatile and useful than that of Auckland *et al* [2] as dynamic torque can be accurately quantized in torque units - as demonstrated by the spring load experiment and the compressor experiment; the response is real-time; and since the system is personal computer based, the whole set-up is portable and economical.

Chapter 3

Theoretical Basis

This chapter presents the theoretical basis for the experimental procedures used in this investigation. First, the introduction outlines the concept of measuring torque in a dynamic system. This is followed by two sections in which different methods of applying known torques are presented.

3.1 Introduction

The study of classical rigid body dynamics is based on the laws of motion published in 1687 by Sir Isaac Newton. In particular, Newton's second law states:

> *The time rate of change of linear momentum of a body is proportional to the force acting upon it and occurs in the direction in which the force acts.* [13]

Mathematically, the law is:

$$\Sigma F = \frac{d}{dt}(mv) \tag{3.1}$$

For a constant mass system

$$\begin{aligned} \Sigma F &= m\frac{dv}{dt} \\ &= ma \end{aligned} \tag{3.2}$$

where ΣF is the sum of the forces acting on the body, $\boldsymbol{m}$ is the mass of the body, $\boldsymbol{v}$ is the velocity of the body, and $\boldsymbol{a}$ is the acceleration of the body in the direction in which the force acts.

For rotation, the equation is

$$\Sigma M = J\alpha \tag{3.3}$$

where ΣM is the sum of the moments acting on the body, $\boldsymbol{J}$ is the mass moment of inertia of the body about the axis of rotation, and α is the angular acceleration about that axis.

This equation is often written as

$$\begin{aligned} \Sigma M &= J\ddot{\theta} \quad \text{and,} \\ \Sigma T &= J\ddot{\theta} \end{aligned} \tag{3.4}$$

where ΣT is the sum of torques acting on the body and $\ddot{\theta}$ represents the second time derivative of angular displacement θ, the angular acceleration.

Now, consider a general dynamic system shown in Fig. 4, where

J_M = mass moment of inertia of the motor
J_C = mass moment of inertia of the coupling shaft
J_L = mass moment of inertia of the load
$\ddot{\theta}$ = angular acceleration of the system
T_M = Instantaneous torque applied by the motor
T_L = Instantaneous torque applied by the load

Let

$$J_S = J_M + J_C + J_L \tag{3.5}$$

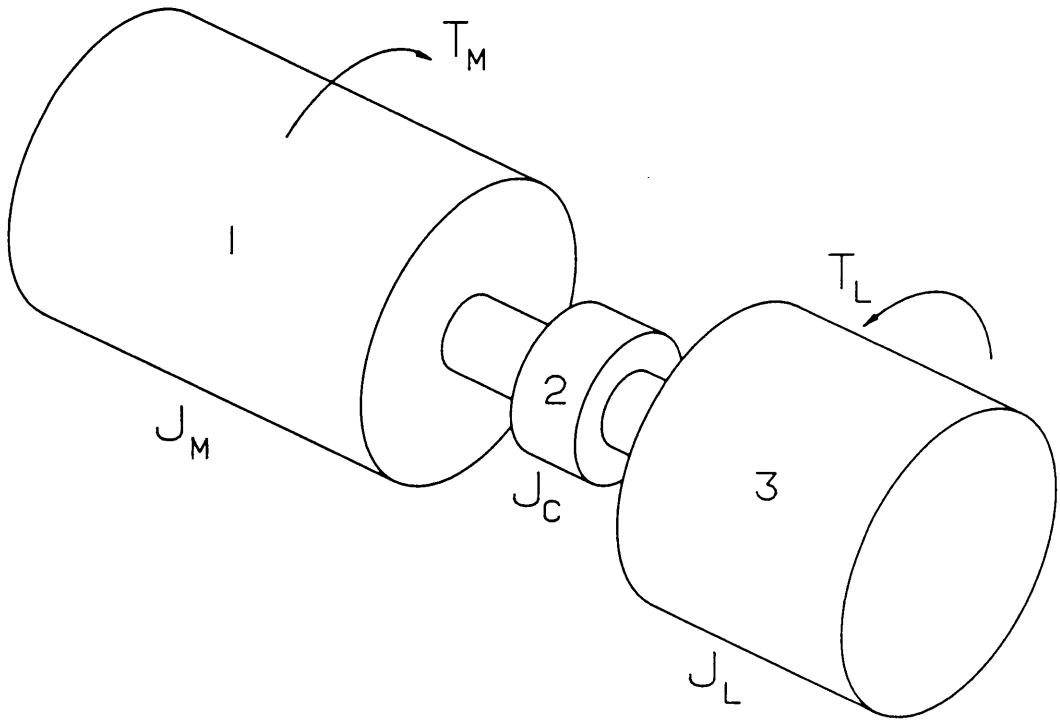
where J_s is the mass moment of inertia of the entire system.

The instantaneous torque of the motor (T_M) is a function of the rotor position and the rotor velocity. For induction motors, the position dependence is manifested in the stator and rotor polor spacings and the velocity dependence is a function of the rate at which the magnetic lines of flux are cut.

For the load device, the applied torque (T_L) is dependent on instantaneous position, velocity and acceleration. In air compressors, the position of the crank determines the pressure load developed, as manifested by the location of the piston with respect to the cylinder head. The velocity dependence is due to friction and lubricaton effects. And the acceleration is a function of the inertia forces developed by the moving parts of the compressor.

Mathematically,

$$T_M = T_M(\theta, \dot{\theta})$$
$$T_L = T_L(\theta, \dot{\theta}, \ddot{\theta})$$



LEGEND: 1. MOTOR
2. COUPLING
3. LOAD

Figure 4. A General Dynamic System

Applying Newton's second law:

$$\begin{aligned} J_S \ddot{\theta} &= T_M - T_L \\ &= T_N \end{aligned} \tag{3.6}$$

where T_N is the net dynamic torque acting on the system.

J_S can be accurately determined by computing J_M, J_C and J_L from the principles of mechanics. Then by measuring angular acceleration ($\ddot{\theta}$), the net torque on the system can be determined. Conversely, if the applied load is known, the torque generated by the motor can be determined.

In this investigation, two different experimental set-ups were used for the purpose of determining dynamic torques in rotating machinery. In both cases, an optical encoder based instrumentation system was used to accurately determine angular velocity from the measured rotational data and evaluate the dynamic torque on the system. Theoretical models were developed for both cases, to predict the dynamic system torque, and are presented in the following sections.

3.2 *The Spring Load Experiment*

Applying a known dynamic load, in this case torque, is a difficult task. Nevertheless, it was equally important to apply a known load, to be able to analyze the results and estimate the accuracy of the torque measurement using shaft

encoders. For this reason, an extension spring was used to apply a known periodic load on a motor. This is schematically shown in Fig. 5.

The experimental setup used for the spring load experiment is shown in Fig. 6. The motor shaft is coupled to one end of the encoder shaft. The other end of the encoder shaft has a flange mounted on it. A linear spring has one of its ends fixed to the base, and the other end attached to the flange.

As the motor turns, so does the encoder shaft and the flange. The force exerted by the spring on the rotating shaft is a function of θ, the angular displacement of the motor shaft and the whipping effects of the spring. This produces a periodic torque load on the system.

3.2.1 The Theoretical Model

An equation of motion can be derived to determine the force applied by the spring as a function of θ, the angular displacement of the motor shaft. Then, by multiplying the force with the torque arm, the magnitude of torque can be determined.

The Linear Spring

The linear spring represents a general stiffness element. When stretched, the spring exerts a force which is proportional to the displacement and opposite in direction.

Mathematically,

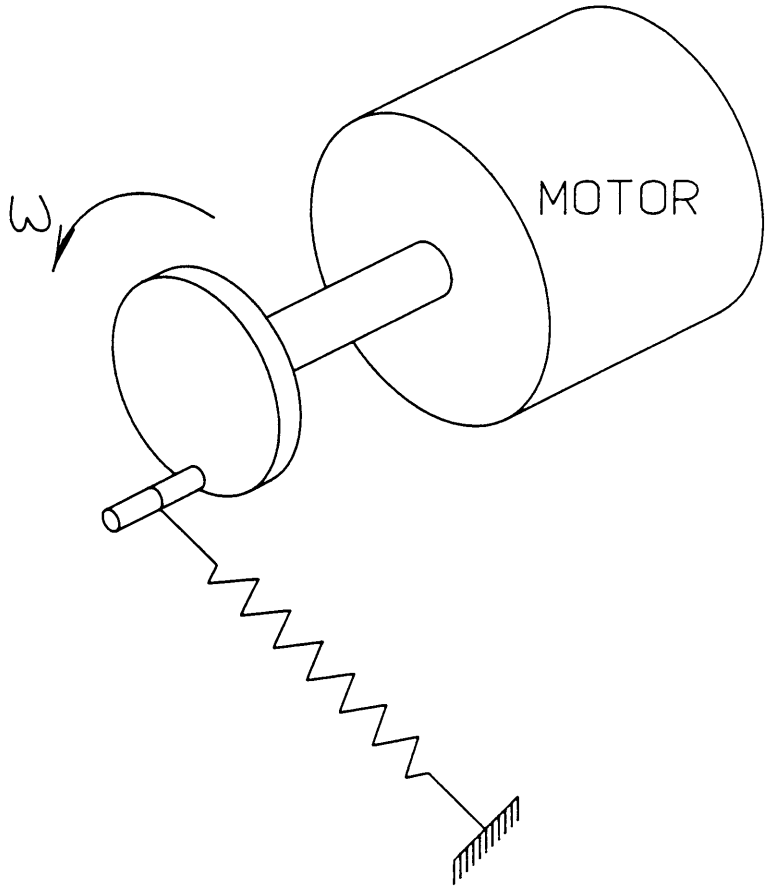


Figure 5. A Dynamic Spring Load System

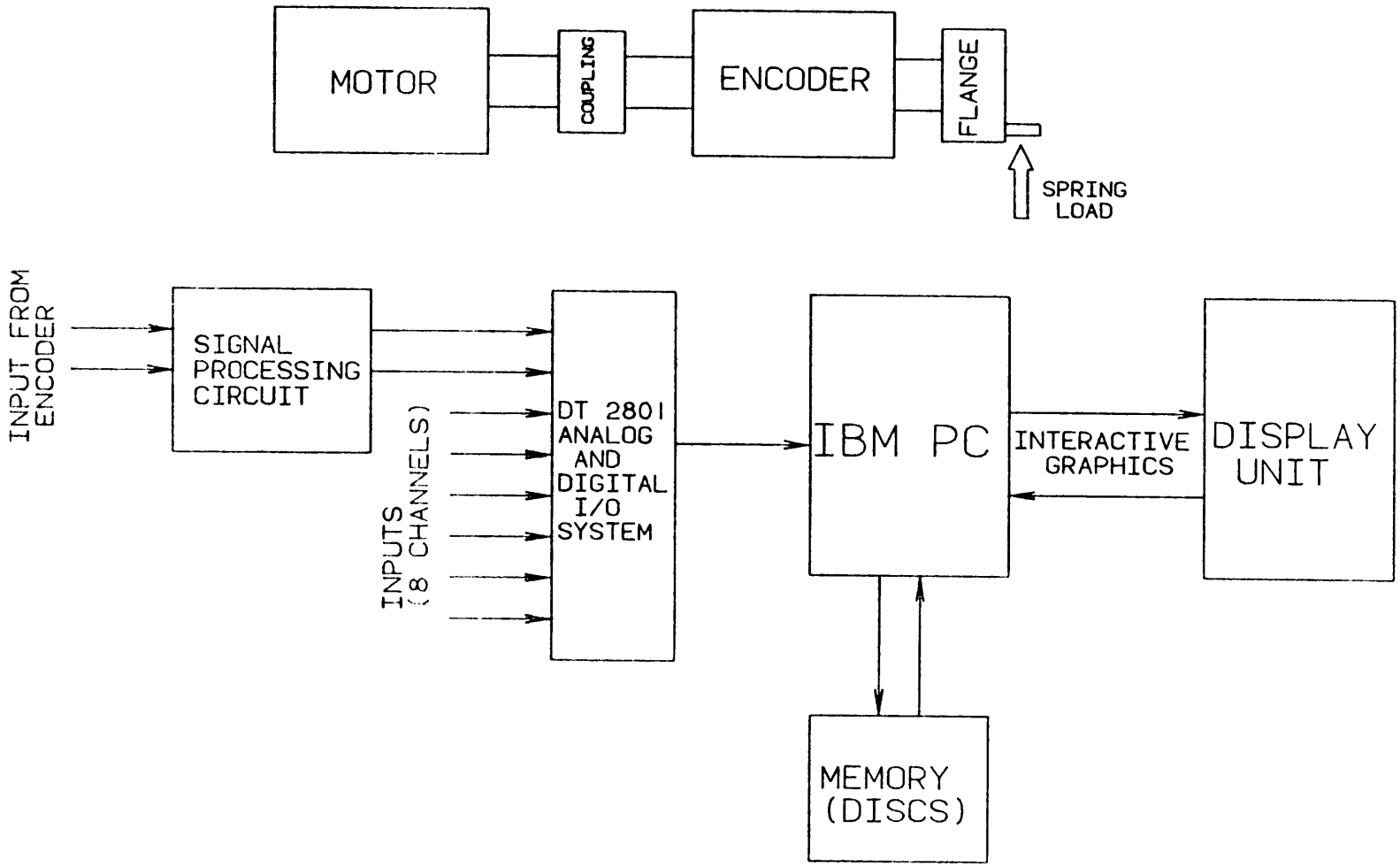


Figure 6. Schematic of Spring Load Experiment

$$F = kx \tag{3.7}$$

where k is the spring constant for the given spring.

By convention, a spring that exerts a positive force is in tension; conversely, a negative force implies compression. The spring used in this experiment is an extension spring and hence operates only in tension.

The Equation of Motion for the Spring Load System

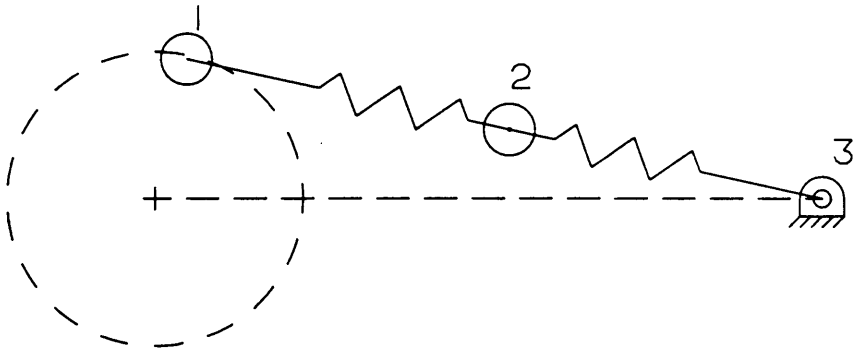
When the linear spring was modeled as having one-half of the mass at each end of the spring, the lateral deflection at the middle of the spring in the actual experimental set-up was found to be significantly higher than that in the case of the theoretical model. To account for this "whipping effect" of the spring, the same was modeled as two equivalent springs with an intermediate lumped mass, as shown in Fig. 7. With this arrangement, the deflection of the spring due to the whipping effect in the theoretical model was found to equal the actual deflection measured in the experimental set-up. Also, the deflection pattern of the spring in operation was in the first mode, confirming the validity of the single intermediate lumped mass.

Consider the free body diagram shown in Fig. 8.

Applying Newton's second law for mass m_2 in the x-direction only,

$$\begin{aligned} m_2\ddot{x} &= \Sigma F_x \\ &= -F_{2x} + F_{1x} \\ &= -kx + k(r\cos\theta - x) \end{aligned}$$

$$m_2\ddot{x} + 2kx - kr\cos\theta = 0 \tag{3.8}$$



SPRING 1-2:
MASS = M/2 STIFFNESS = 2K

SPRING 2-3:
MASS = M/2 STIFFNESS = 2K

COMPONENT OF MASS AT "1" = 1/2(M/2) = M/4

COMPONENT OF MASS AT "2" = 1/2(M/2) + 1/2(M/2)
= M/2

COMPONENT OF MASS AT "3" = 1/2(M/2) = M/4

Figure 7. Dynamic model of spring

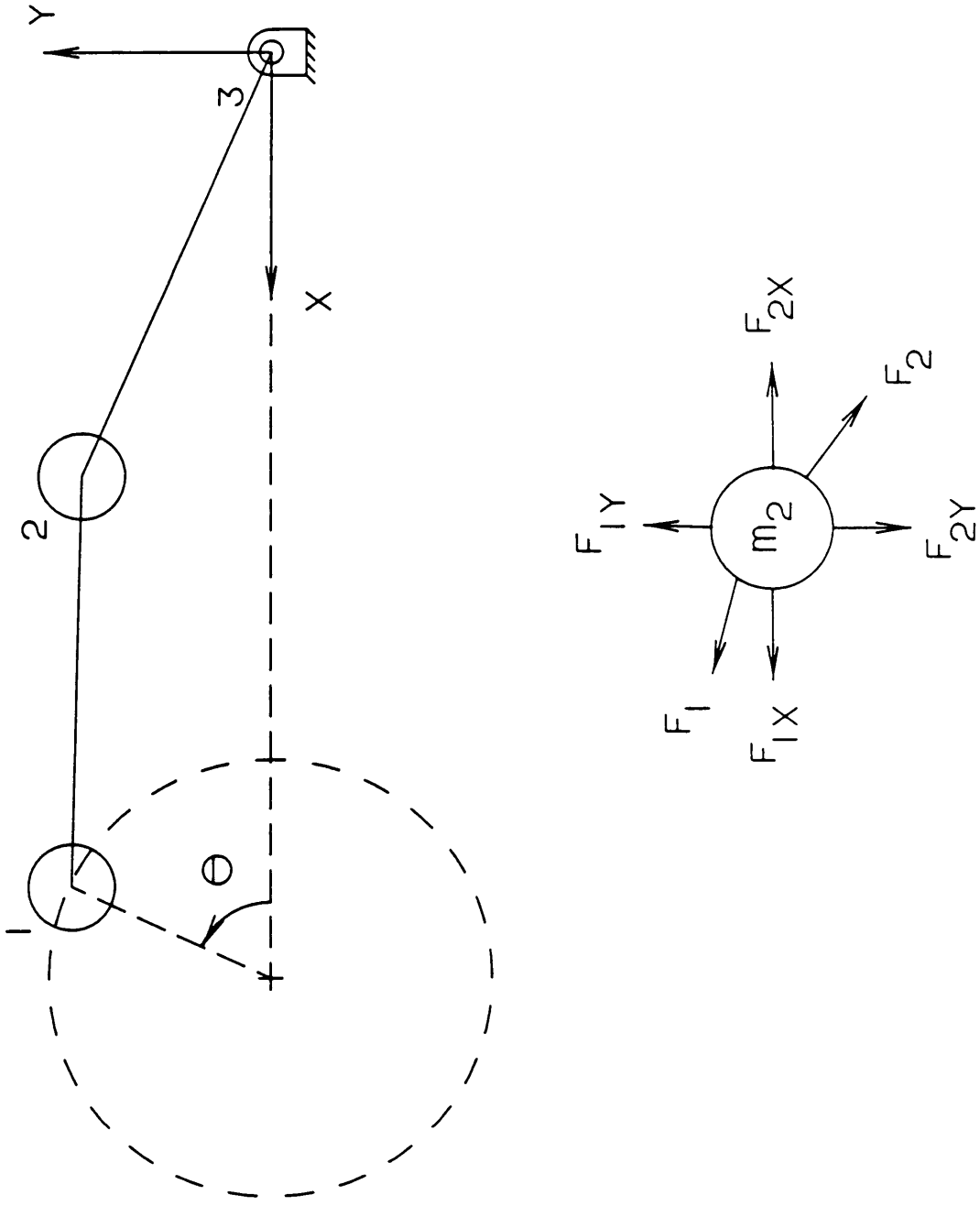


Figure 8. Freebody Diagram for Spring Load Experiment

Similarly, in the y-direction

$$m_2\ddot{y} + 2ky - kr\sin\theta = 0 \tag{3.9}$$

The stiffness K of the spring used in this experiment was theoretically computed [28] from

$$K = \frac{d^4G}{8D^3N} \tag{3.10}$$

where

d = diameter of wire, 0.040 in

G = Shear modulus, 11.85×10^6 psi for music wire

D = mean spring diameter, 0.370 in

N = number of coils, 25

Substituting these values

$$K = 2.995 \text{ lbf/in}$$

The stiffness of the spring was also determined experimentally and the result was within 2% of the theoretical value. This was done by applying known weights and measuring the deflection of the spring.

The total weight of the spring as measured is

$$W = 0.0128 \text{ lb}$$

Therefore, the total mass is

$$M = 3.32 \times 10^{-5} \text{ lbf-sec}^2/\text{in}$$

From Fig. 7 it is evident that

$$m_2 = \frac{M}{2}$$
$$= 1.66 \times 10^{-5} \text{ lbf-sec}^2/\text{in}$$

Also,

$$k = 2 \times K$$
$$= 5.990 \text{ lbf/in}$$

The torque arm

$$r = 0.380 \text{ in}$$

Now, substituting for m_2 and k in equations 3.8 and 3.9

$$1.66 \times 10^{-5}\ddot{x} + 11.980x - 2.156 \cos\theta = 0 \quad (3.11)$$

$$1.66 \times 10^{-5}\ddot{y} + 11.980y - 2.156 \sin\theta = 0 \quad (3.12)$$

The physical system used in this experiment is essentially an undamped system. However, for computational stability, minimal damping is introduced into the above equations to quickly arrive at the steady state solution. Once the steady state solution is reached, the damping is removed from the solution with the aid of a weighting function. This is achieved as follows:

Re-writing the above equations,

$$1.66 \times 10^{-5}\ddot{x} + c\dot{x} + 11.980x - 2.156 \cos\theta = 0 \quad (3.13)$$

$$1.66 \times 10^{-5}\ddot{y} + c\dot{y} + 11.980y - 2.156 \sin\theta = 0 \quad (3.14)$$

Where **c** is the coefficient of viscous damping. Now to determine **c** in the above equations:

It is known that for a single degree of freedom system [18]

$$\zeta = \frac{c}{2m\omega_n} \tag{3.15}$$

where ζ = viscous damping ratio

ω_N = undamped natural frequency of oscillation

$$= \sqrt{\frac{k}{m}}$$

$$= \sqrt{\frac{11.980}{1.66 \times 10^{-5}}}$$

$$= 849.520 \text{ rad/sec}$$

Now Eq. (3.15) can be re-written as:

$$c = 2\zeta \times 1.66 \times 10^{-5} \times 849.520$$

$$= 0.028\,\zeta \text{ lbf-sec/in}$$

Here, a weighting function e^{-At}, where A is any suitable positive constant and t is the time, is introduced in the above equation to gradually remove the effect of the damping factor in the solution.

$$c = 0.028\,\zeta e^{-At} \text{ lbf-sec/in} \tag{3.16}$$

Therefore, equations 3.13 and 3.14 become

$$\begin{bmatrix} 1.66 \times 10^{-5} & 0 \\ 0 & 1.66 \times 10^{-5} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0.028 & 0 \\ 0 & 0.028 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} e^{-At}$$

$$+ \begin{bmatrix} 11.057 & 0 \\ 0 & 11.057 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 1.9903 \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{3.17}$$

The above equations can be simultaneously solved for x and y in the time domain by integration.

The initial conditions used to solve the above equation are:

$$\dot{x} = 0.000$$

$$\dot{y} = \frac{r\omega}{2}$$

$$= 32.044 \text{ in/sec}$$

where

r = radius of rotation, 0.3825 in

ω = angular velocity of motor, 167.552 rad/sec (1600 rpm)

With the position of m_2 as defined by x and y (which are functions of θ) as a known quantity, the force exerted by the spring on the rotating flange can be determined.

Now, consider mass m_1 in Fig. 8

$$F_{1x} = -k\,(r\cos\theta - x)$$

$$F_{1y} = -k\,(r\sin\theta - y)$$

Therefore

$$F_1 = \sqrt{F_{1x}^2 + F_{1y}^2} \qquad (3.18)$$

Thus, for any value of θ in the range 0 to 360°, a unique set of x and y values and hence F_1 can be computed.

The product of F_1 and the torque arm gives the applied torque.

Mathematically

$$T_{load} = F_1 \times \text{Torque arm} \tag{3.19}$$

A FORTRAN program was written to compute x, y and T_{load} for one rotation and display the results graphically. The code for the same is listed in Appendix A.

3.2.2 Computations for Spring Load Experiment

As discussed in Sec. 3.1, if the system is essentially undamped, given the mass moment of inertia of the whole rotating system, the net dynamic torque on the system can be determined in real-time, by measuring the angular acceleration ($\ddot{\theta}$) of the rotating system. The following sub-sections present the determination of mass moments of inertia of the rotating masses in the spring load experimental set-up.

A schematic of the system of rotating masses in the spring load experimental set-up is shown in Fig. 9.

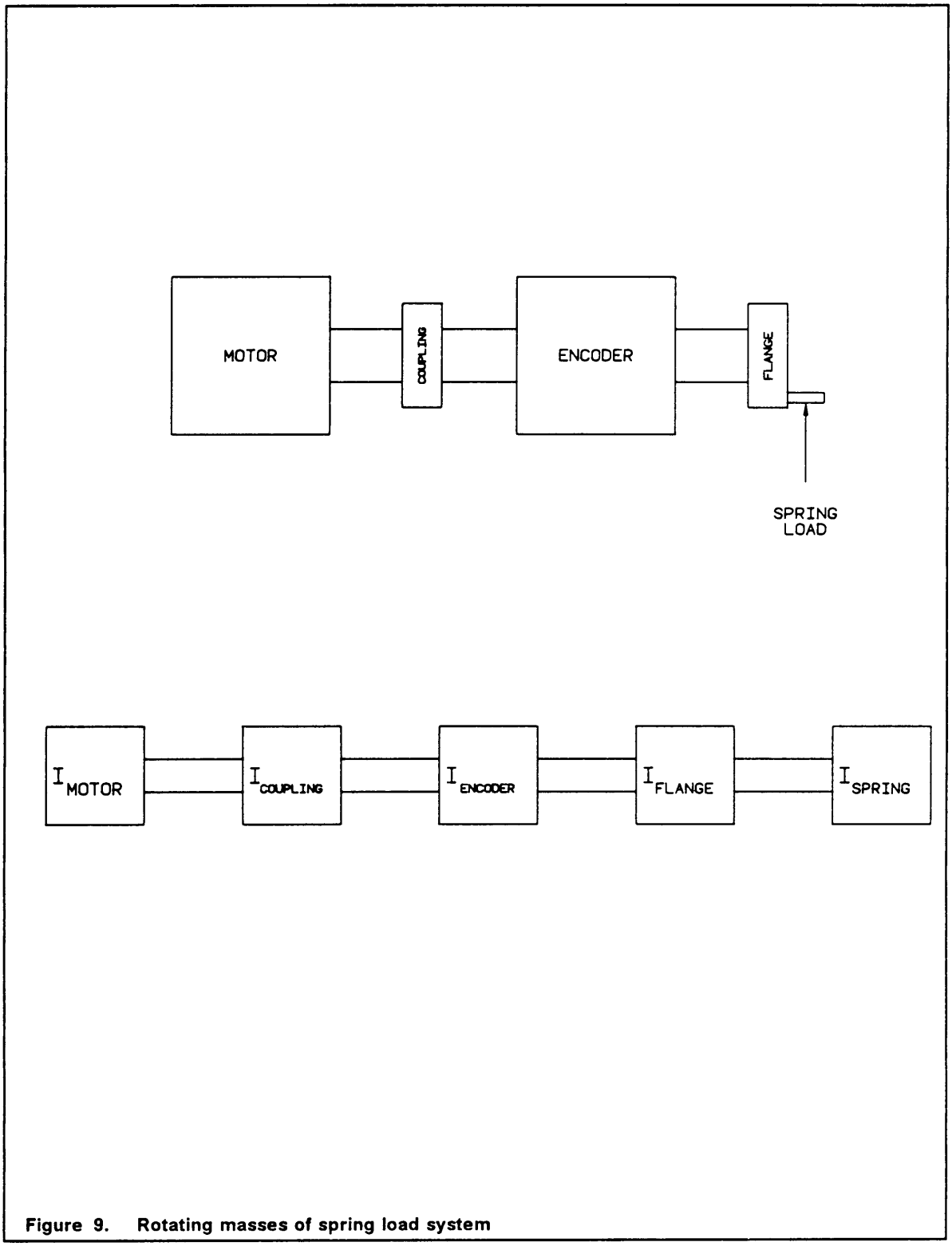


Figure 9. Rotating masses of spring load system

Inertia of the Motor

As the mass moment of inertia of the motor was not available from the specifications of the motor, it had to be determined. For this purpose, the rotor and shaft assembly, which was the rotating mass in the motor, was disassembled.

The following experimental method was adopted to accurately determine the mass moment of inertia of the rotor and shaft assembly.

A piece of sheet metal was cut-out into an equilateral triangular shape as shown in Fig. 10. The mass moment of inertia of this can be determined from the geometry as follows:

$$I = \frac{W/g(3b^2 + 4h^2)}{72} \tag{3.20}$$

where

W = weight of the triangular sheet, 0.234 lbf

b = width of the triangular sheet, 13.375 in (Fig. 10)

h = height of the triangular sheet, 11.580 in (Fig. 10)

Substituting,

$$I = 9.035 \times 10^{-3} \text{ lbf-in-sec}^2$$

The effects of the circular hole at the center and the three holes near the vertices of the triangular sheet were neglected as their combined mass amounted to less than 1% of the triangular sheet's weight.

Then, the triangular sheet was hung with cords as shown in Fig. 10. The cords were assumed to be stiff and massless.

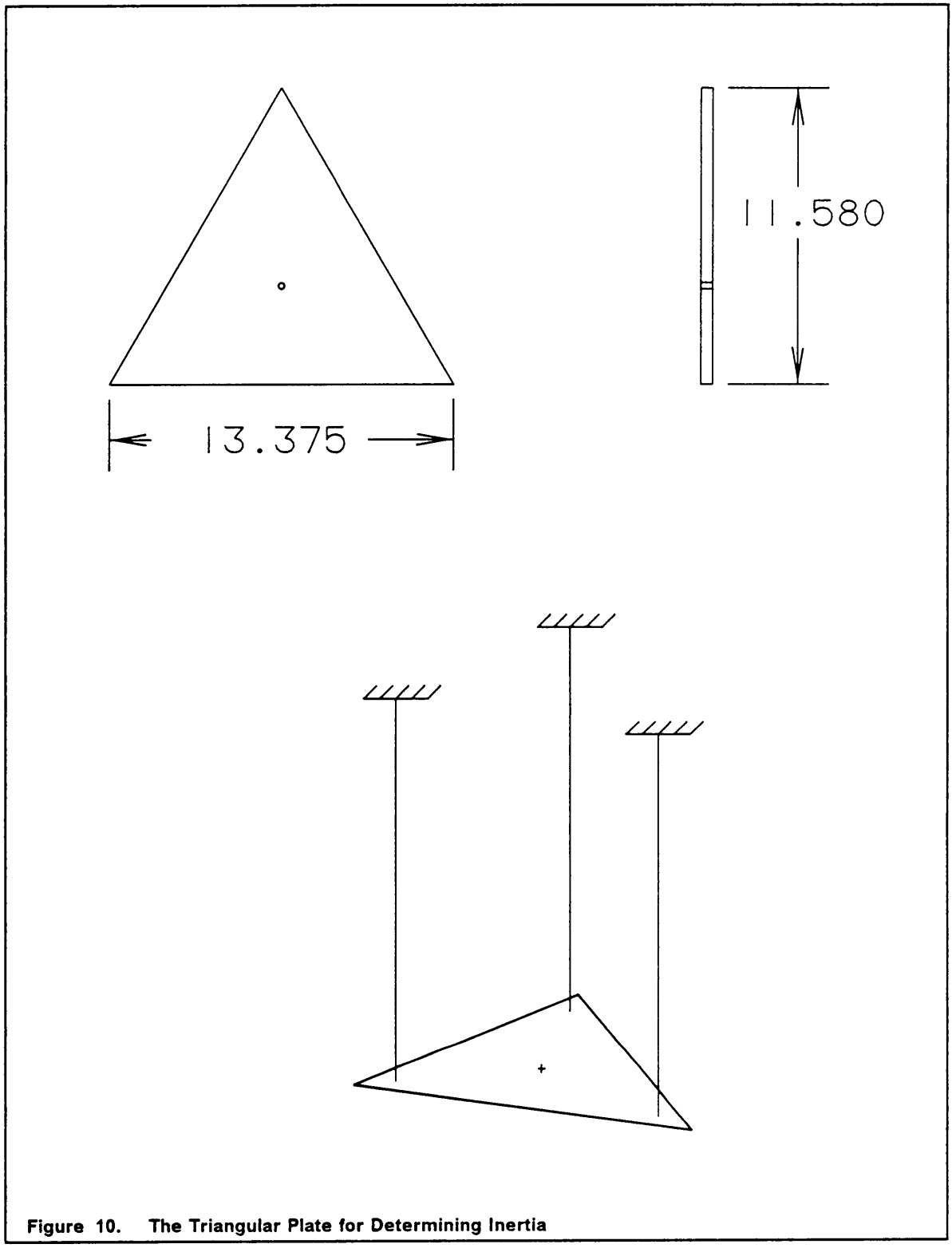


Figure 10. The Triangular Plate for Determining Inertia

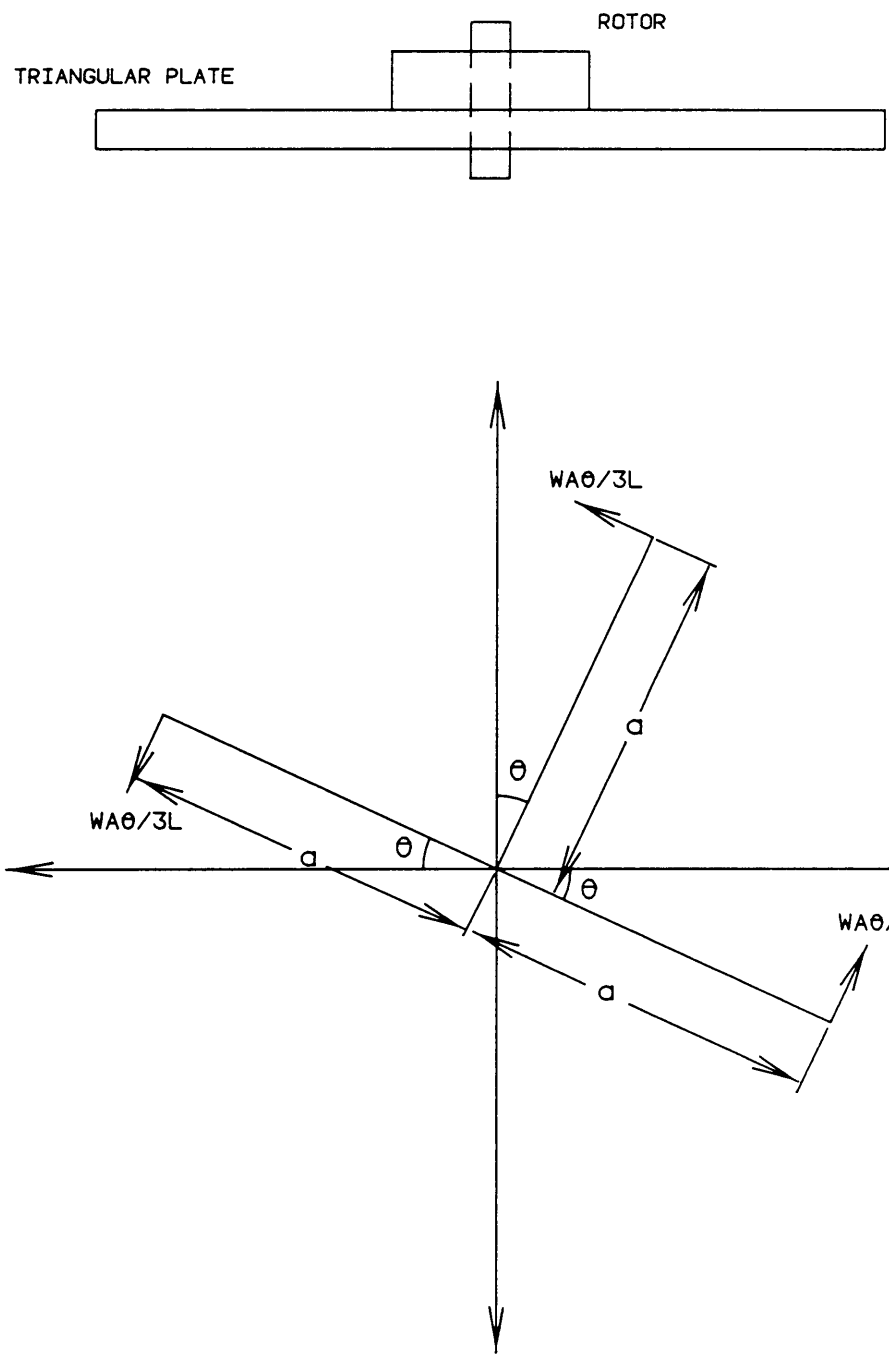


Figure 11. Free body diagram for triangular plate system

The equation of motion for the angular oscillation of the triangular plate about a normal through the plate centroid axis is derived as follows:

Applying Newton's second law for the free-body diagram (Fig. 11)

$$\begin{aligned} I\ddot{\theta} &= \Sigma M \\ &= -\frac{W}{3}\frac{a\theta}{L}(a) - \frac{W}{3}\frac{a\theta}{L}(a) - \frac{W}{3}\frac{a\theta}{L}(a) \\ &= -\frac{Wa^2\theta}{L} \end{aligned}$$

$$I\ddot{\theta} + \frac{Wa^2}{L}\theta = 0 \tag{3.21}$$

From the above equation

$$\omega_n = \frac{1}{2}\pi\sqrt{\frac{Wa^2}{LI}}\ \text{Hz} \tag{3.22}$$

Re-writing the above equation

$$I = \frac{Wa^2}{4\pi^2 L\omega_N^2}\ \text{lbf-in-sec}^2 \tag{3.23}$$

Substituting the known values:

$$I = \frac{0.234(6)^2}{4\pi^2(15.125)\omega_N^2}\ \text{lbf-in-sec}^2 \tag{3.24}$$

From the above equation, the mass moment of inertia of the triangular plate can be determined by substituting for ω_N, the natural frequency of oscillation of the triangular plate.

The plate was given a small angle twist from its equilibrium position and the time period for 25 oscillations of the triangular plate about the vertical axis was measured with the help of a chronometer.

$$\omega_N = \frac{25}{\text{Average time for 25 oscillations}}$$
$$= \frac{25}{19.83}$$
$$= 1.26 \text{ Hz}$$

Substituting this in Eq. 3.24

$$I_{Plate} = 8.886 \times 10^{-3} \text{ lbf-in-sec}^2$$

The difference in values in the moments of inertia obtained by the experimental method and theoretical method was found to be as small as 1.6%.

Then, the rotor and shaft assembly was placed at the center of the triangular plate as shown in Fig. 10. The rotor and shaft assembly was firm on the triangular plate as the shaft of the rotor and shaft assembly fitted exactly into the circular hole at the center of the triangular plate. The equation of motion for this system can be obtained by modifying Eq. 3.21 as follows:

$$(I_{Plate} + I_{Rotor})\,\ddot{\theta} + \frac{(W_{Plate} + W_{Rotor})\,a^2}{L}\theta = 0 \qquad (3.25)$$

where

W_{Rotor} = weight of the rotor and shaft assembly
= 0.388 lbf

From the above equation

$$\omega_n = \frac{1}{2\pi}\sqrt{\frac{(W_{Plate} + W_{Rotor})\, a^2}{L(I_{Plate} + I_{Rotor})}} \text{ Hz} \tag{3.26}$$

Re-writing the above equation

$$I = \frac{(W_{Plate} + W_{Rotor})\, a^2}{4\pi^2 L \omega_N^2} - I_{Plate} \tag{3.27}$$

Substituting the known values:

$$\begin{aligned} I &= \frac{(0.234 + 0.388)\, 6^2}{4\pi^2 13.6875 \omega_N^2} - 9.035 \times 10^{-3} \\ &= 0.0414\, \tau^2 - 9.035 \times 10^{-3} \text{ lbf-in-sec}^2 \end{aligned} \tag{3.28}$$

The plate was given a small angle twist from its equilibrium position and the time period (τ) for 50 oscillations of the triangular plate about the vertical axis was measured with the help of a chronometer.

$$\begin{aligned} \tau &= \frac{\text{Average time for 50 oscillations}}{50} \\ &= \frac{23.41}{50} \\ &= 0.47 \text{ sec} \end{aligned}$$

Substituting this in Eq. 3.28

$$I_{Rotor} = 1.10 \times 10^{-4} \text{ lbf-in-sec}^2$$

Inertia of the Coupling

The flexible coupling which connects the motor shaft with the encoder shaft is shown in Fig. 12.

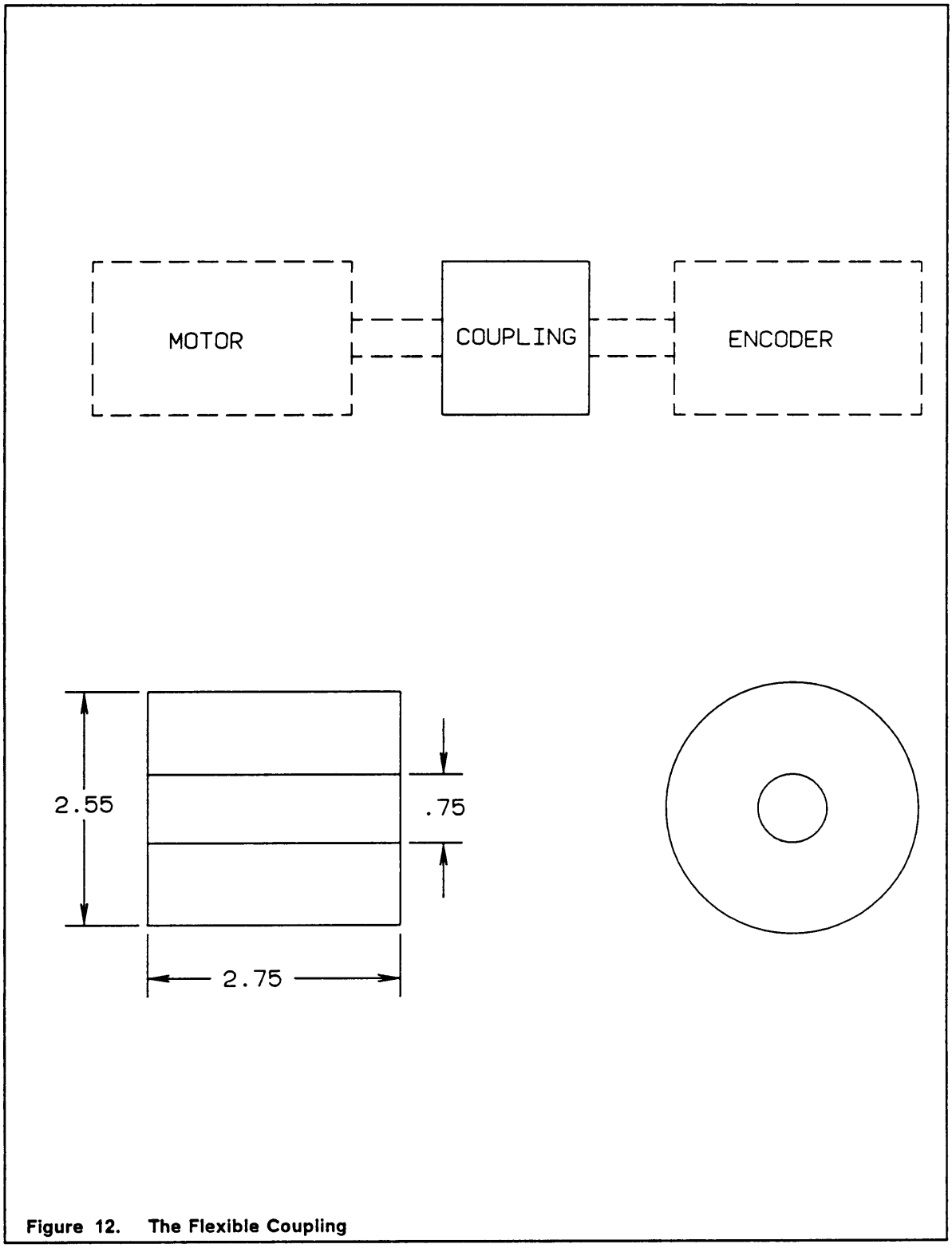


Figure 12. The Flexible Coupling

The mass moment inertia of the coupling is determined as follows:

$$\text{Weight of the coupling} = 0.0463 \text{ lbf}$$
$$m = \text{Mass of the coupling} = 1.199 \times 10^{-4} \text{ lbf-sec}^2/\text{in}$$

From Fig. 12.

$$r_o = \text{Outer radius} = 2.55/2 \text{ in}$$
$$r_i = \text{Inner radius} = 0.75/2 \text{ in}$$

Substituting,

$$I_{Coupling} = \frac{1}{2} \times m\,(r_o^2 - r_i^2)$$
$$= 9.9 \times 10^{-6} \text{ lbf-in-sec}^2$$

Inertia of the Encoder

The mass moment of inertia of the shaft encoder as given by the manufacturer [8] for the Model 714 ACCU-CODER is:

$$I_{Encoder} = 0.0025 \text{ ounce-in-sec}^2$$
$$= 1.563 \times 10^{-4} \text{ lbf-in-sec}^2$$

Inertia of the Flange

The flange is mounted on the encoder shaft. The flange acts as a fixture for the "moving end" of the spring as shown in Fig. 13.

The mass moment of inertia of the Flange with the crank pin is

$$I_{Flange} = 1.17 \times 10^{-5} \text{ lbf-in-sec}^2 \qquad (3.29)$$

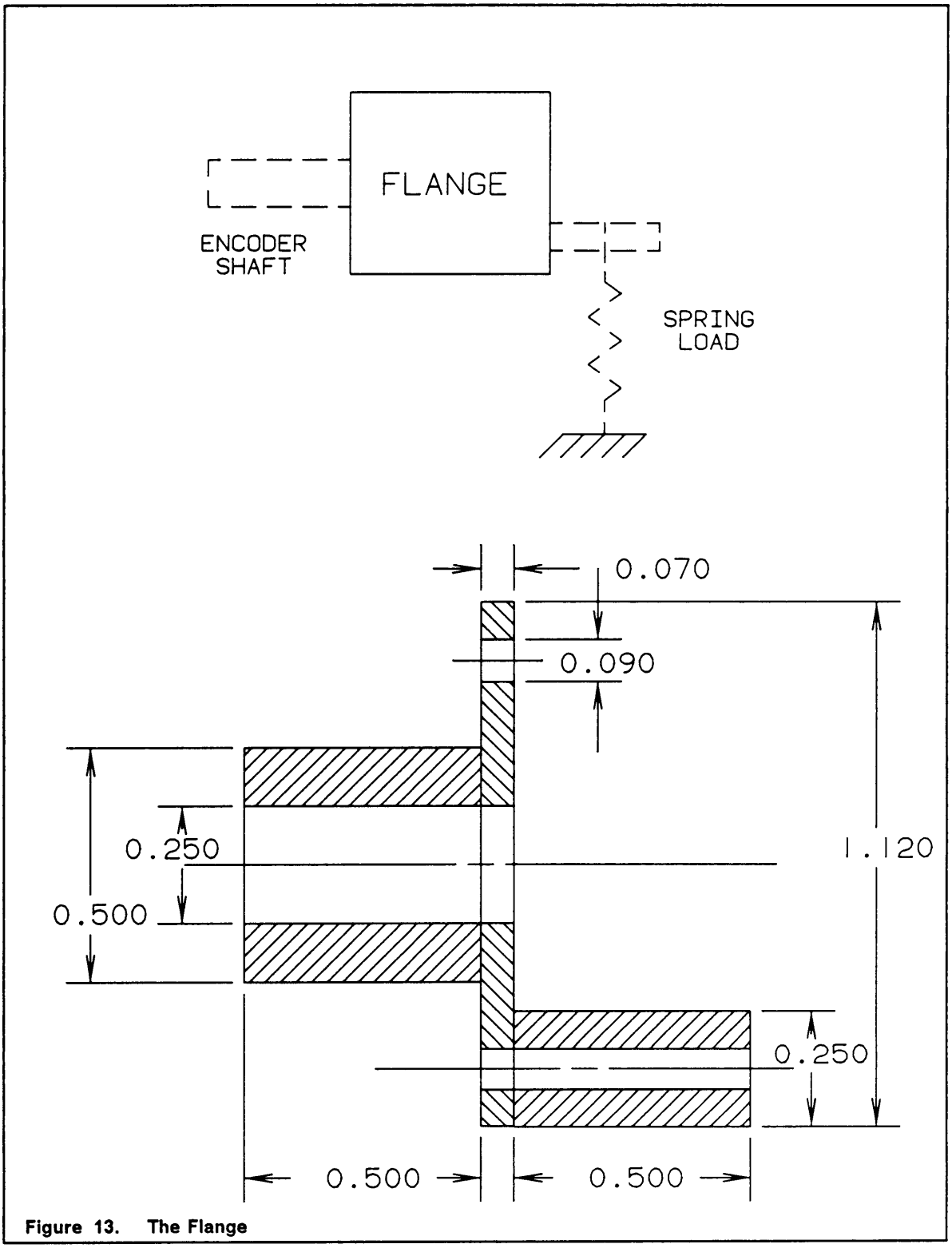


Figure 13. The Flange

Inertia of the Spring

Fig. 7 shows the distribution of spring mass in the dynamic spring model.

$$\begin{aligned} \text{Weight of the spring} &= 0.01278 \text{ lbf} \\ m = \text{Mass of the spring} &= \frac{0.01278}{386} \\ &= 3.31 \times 10^{-5} \text{ lbf-sec}^2/\text{in} \\ r = \text{radius of rotation} &= 0.3825/2 \end{aligned}$$

Therefore the mass moment of inertia of the spring is

$$\begin{aligned} I_{Spring} &= \frac{1}{2}mr^2 \\ &= \frac{1}{2}\left(\frac{3.31 \times 10^{-5}}{2}\right)\left(\frac{0.3825}{2}\right)^2 \\ &= 3 \times 10^{-7} \text{ lbf-in-sec}^2 \end{aligned}$$

Inertia of the System

Now, the total inertia of the system is

$$\begin{aligned} I &= I_{Rotor} + I_{Coupling} + I_{Encoder} + I_{Flange} + I_{Spring} \\ &= 1.10 \times 10^{-4} + 9.9 \times 10^{-6} + 1.563 \times 10^{-4} + 1.17 \times 10^{-5} + 3 \times 10^{-7}(3.30) \\ &= 2.882 \times 10^{-4} \text{ lbf-in-sec}^2 \end{aligned}$$

3.2.3 Results of the Spring Load Experiment

Equations 3.17, 3.18 and 3.19 for the theoretical model were coded in FORTRAN and a Runge-Kutta algorithm was used to integrate the differential equations. A listing of the program is included in Appendix A. The program was run on a Digital Equipment VAX 11/780.

In the experimental determination of the torque, the encoder output (600 Pulses Per Revolution) signal which is an FM (Frequency Modulated) velocity signal, is converted to an AM (Amplitude Modulated) velocity signal in the signal processing circuitry. Then the FORTRAN program uses an FFT algorithm and differentiates the measured angular velocity signal ($\dot{\theta}$) in the frequency domain to obtain angular acceleration ($\ddot{\theta}$). This is multiplied by I, the mass moment of inertia of the rotating system and the torque signal is displayed for one revolution of the motor shaft.

Fig. 14 shows the measured velocity signal. The spectrum of the first 12 harmonics of the signal is shown in Fig. 15. Because of the significant presence of the 2nd and higher order harmonics, the angular velocity signal is not truly sinusoidal. This characteristic is typical of single phase induction motors and is called *double-frequency torque pulsation* [9]. The double-frequency current in the rotor reacting upon the slip-frequency forward magnetic field produces a torque pulsation on the motor shaft, even at no load. Figures 16 and 17 show the velocity and acceleration spectra when the motor was running under no load condition. The motor used in this particular experiment runs at approximately 26.7 Hz (1600 rpm) and double-frequency (line) is 120 Hz, which is the reason for the 4th harmonic to be prominent, as seen in Figures 16 and 17. Fig. 18 shows the acceleration spectrum in the spring load experiment. Now, the 2nd and higher order harmonics are filtered (removed) using an FFT algorithm (Fig. 19). Fig. 20 shows the dynamic torque waveform after the acceleration signal has been multiplied by the mass moment of inertia as well as the corresponding dynamic torque waveform of the theoretical model. Fig. 21 shows the spectra of these torque waveforms. Fig. 22 shows a typical dynamic torque signal when the acceleration signal is not filtered. The FORTRAN program used for the

above computations and graphics is listed in Appendix B. A discussion of the results obtained is included in Chapter 5.

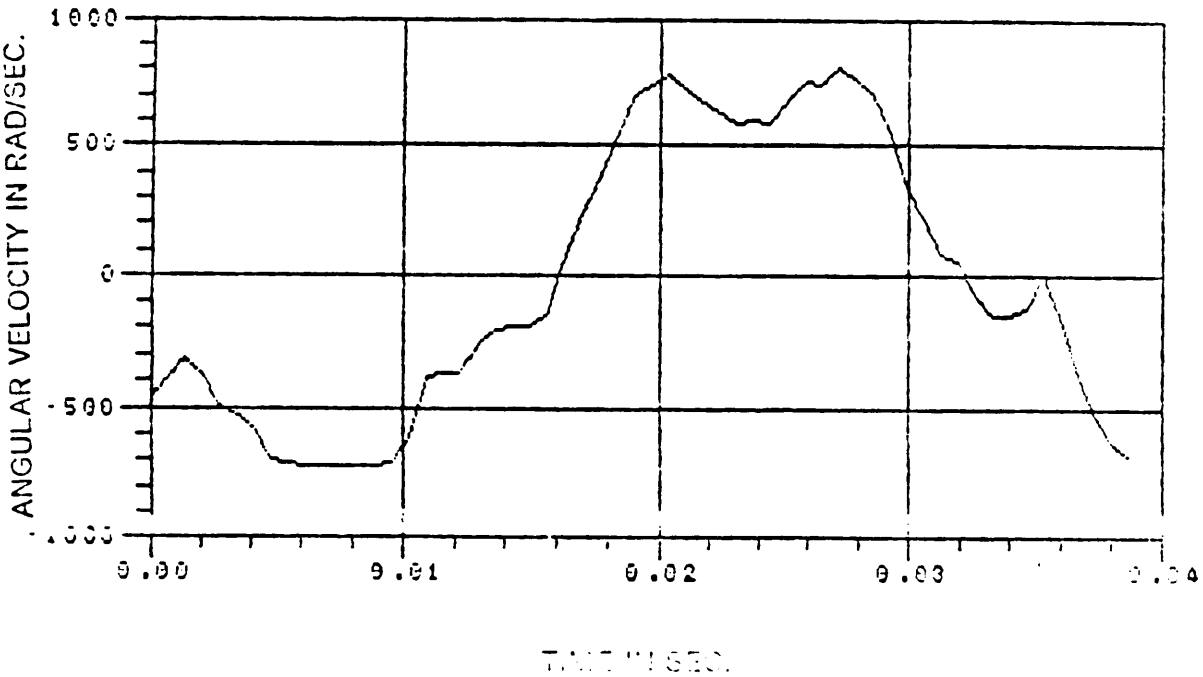


Figure 14. The measured velocity signal

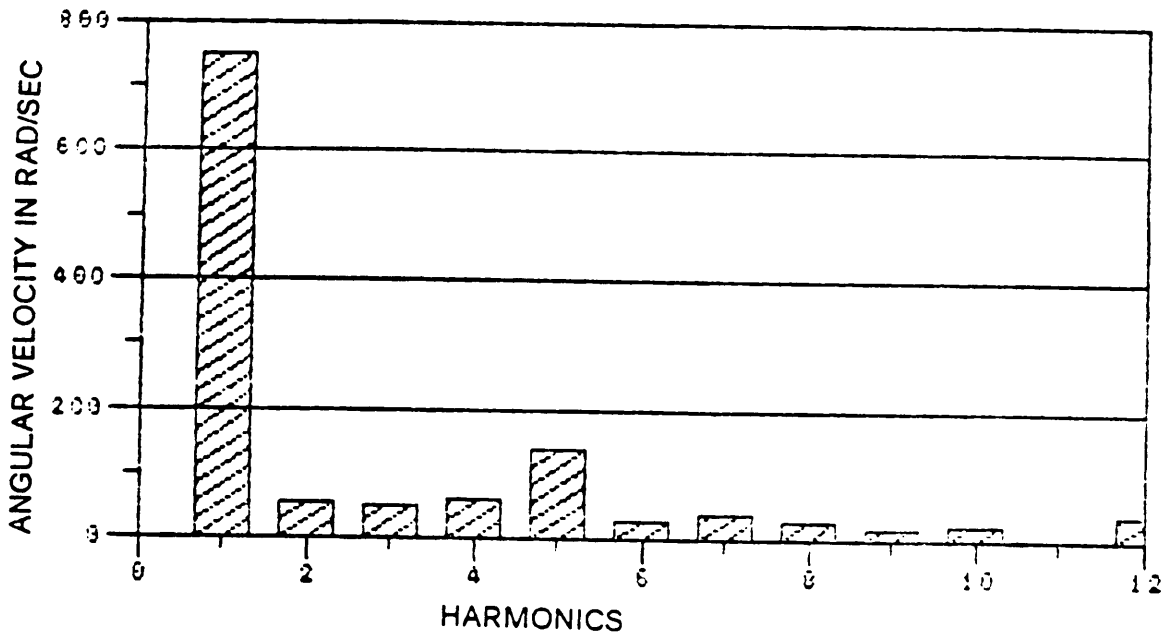


Figure 15. The measured velocity spectrum

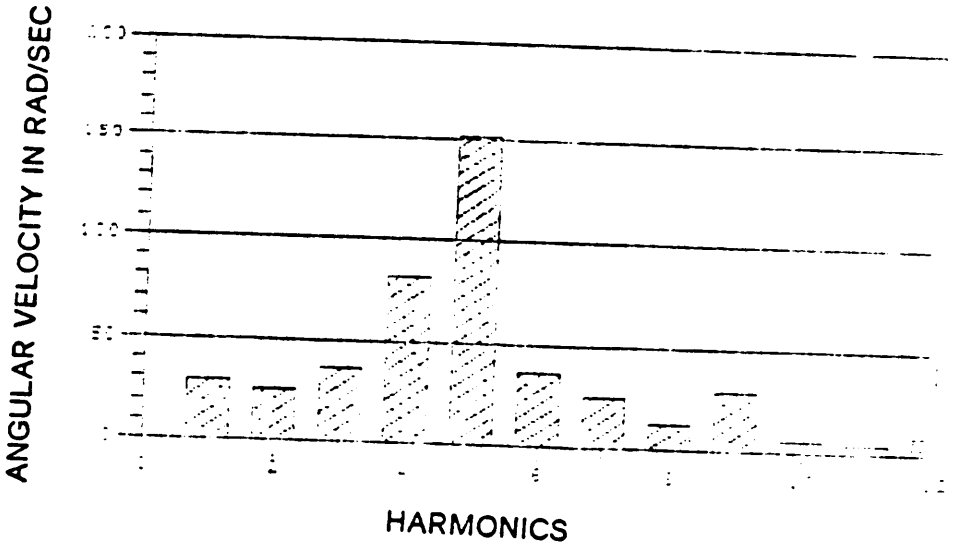


Figure 16. The no load measured velocity spectrum

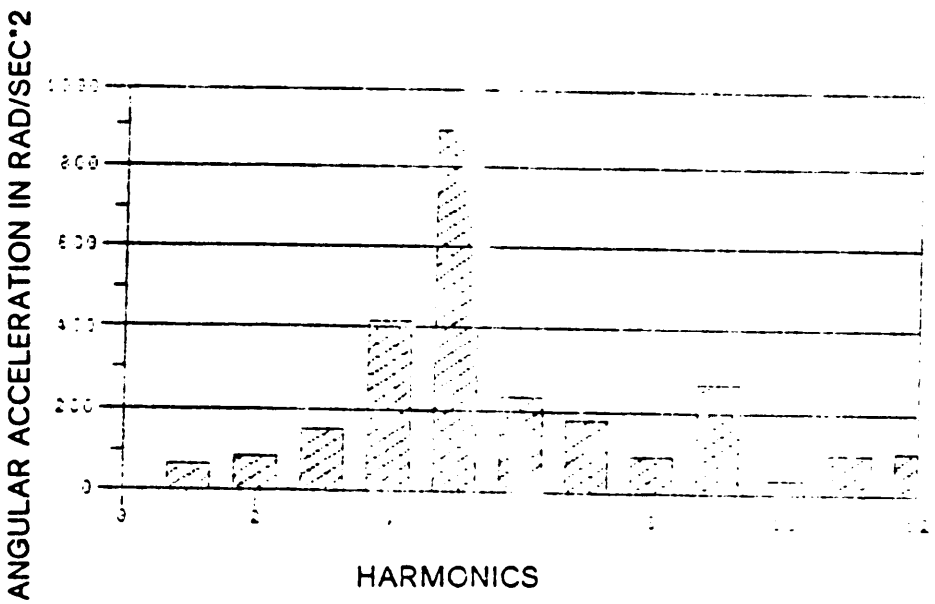


Figure 17. The no load measured acceleration spectrum

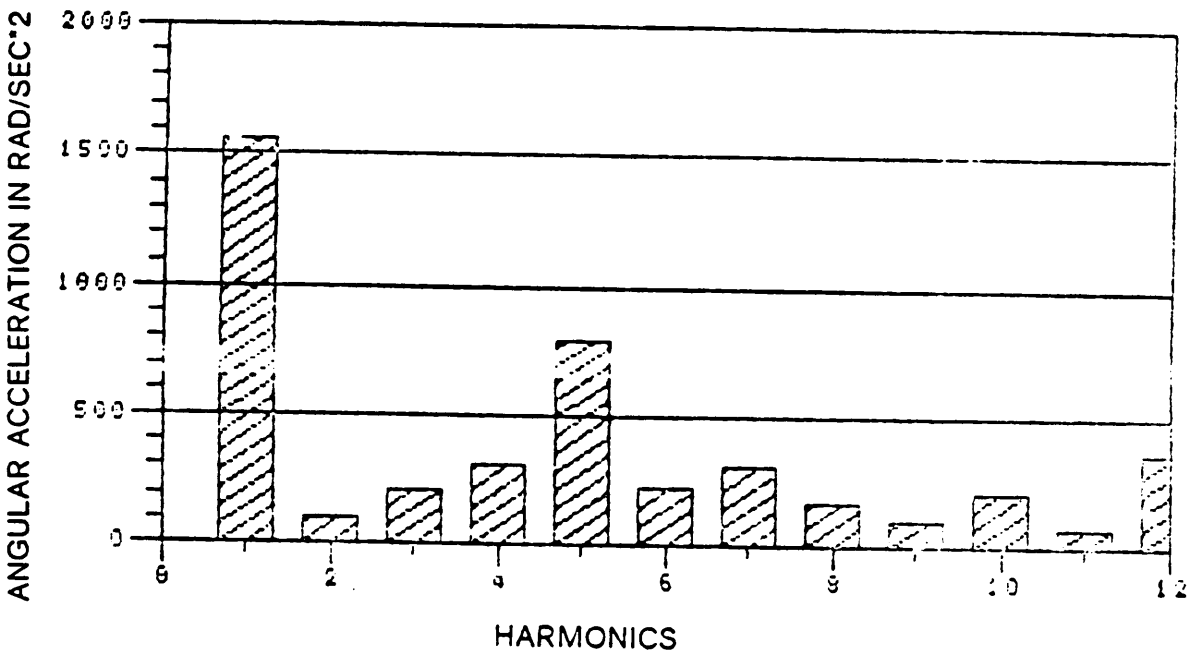


Figure 18. The measured acceleration spectrum (unfiltered)

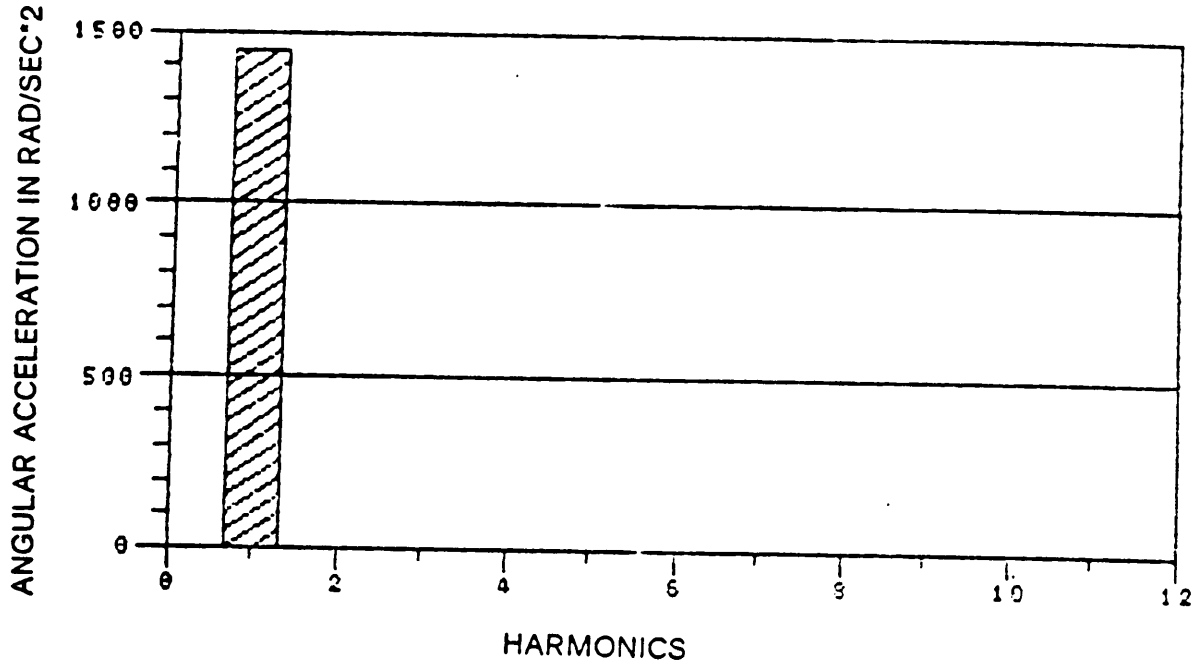


Figure 19. The measured acceleration spectrum (filtered)

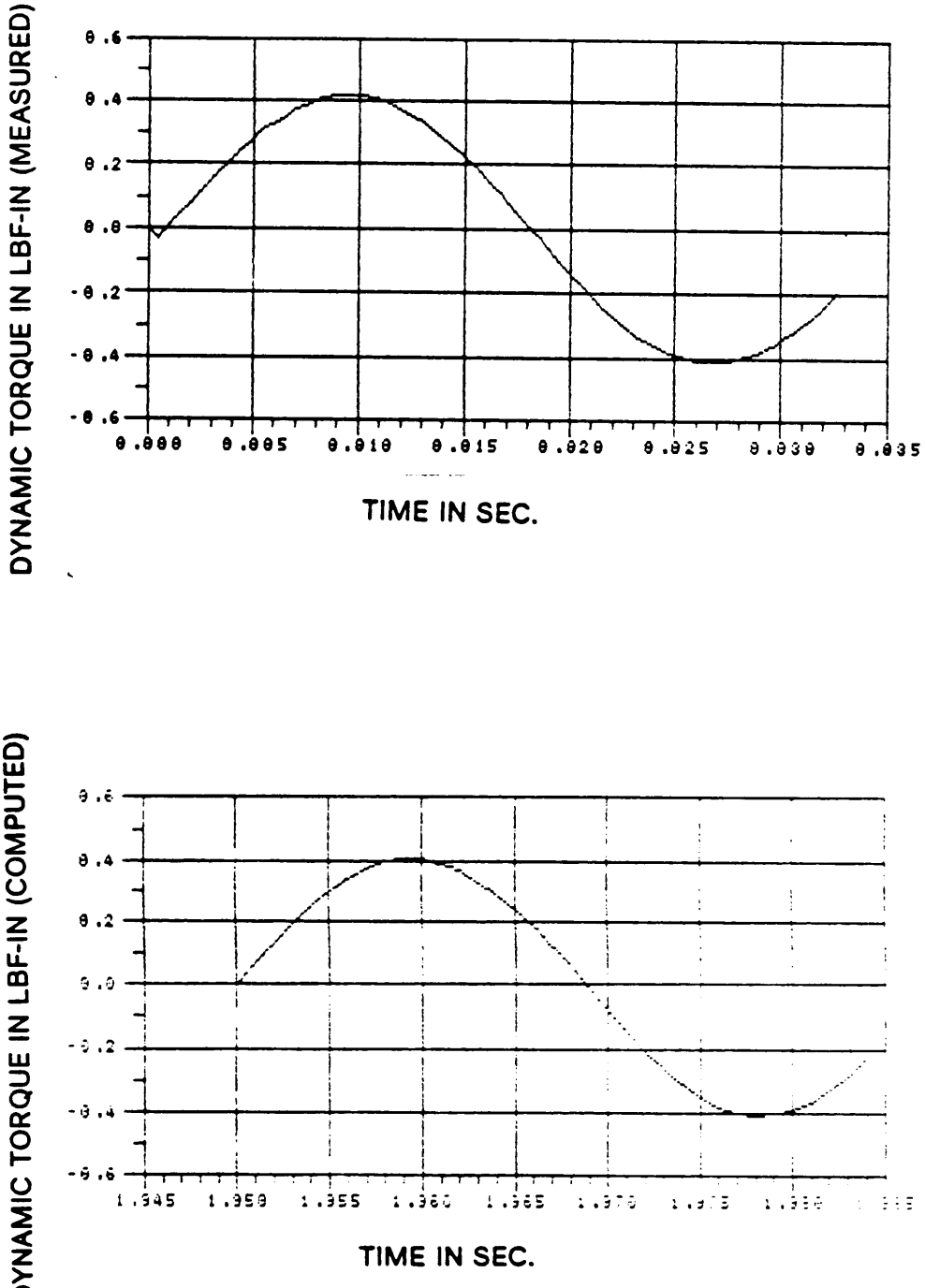
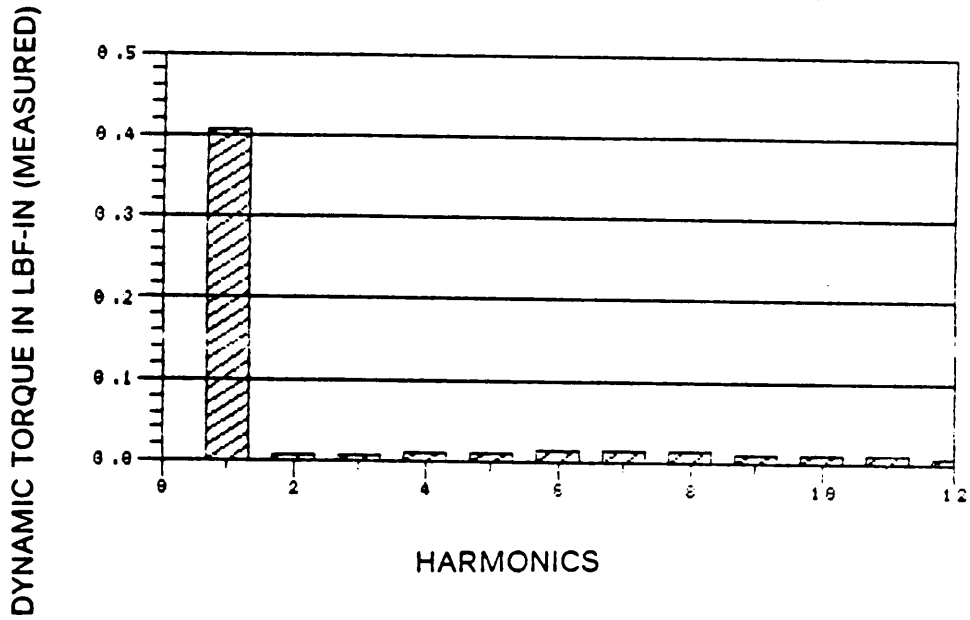


Figure 20. The measured and computed torque waveforms



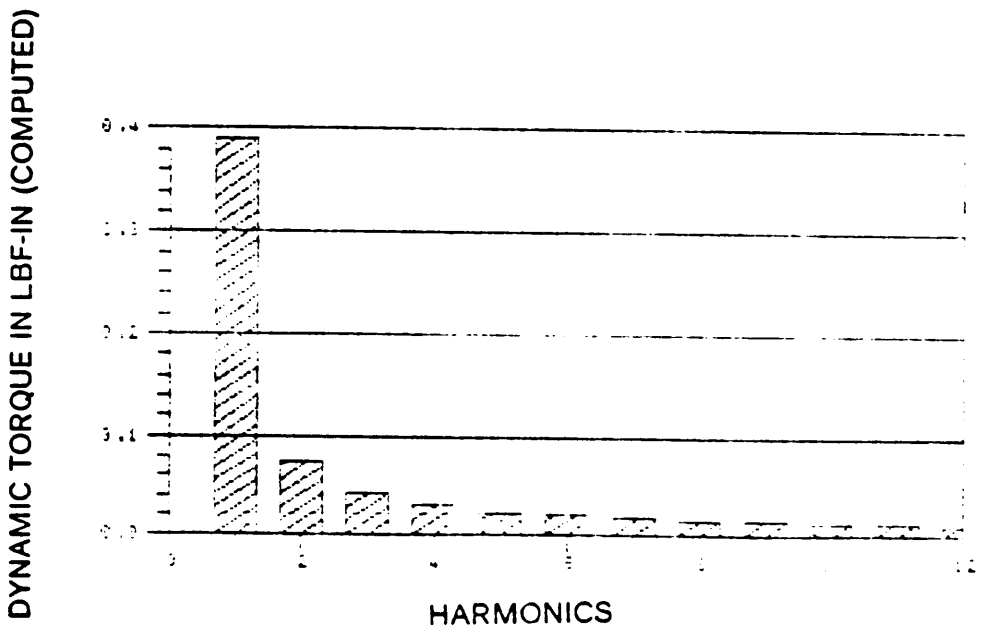


Figure 21. The measured and computed torque spectra

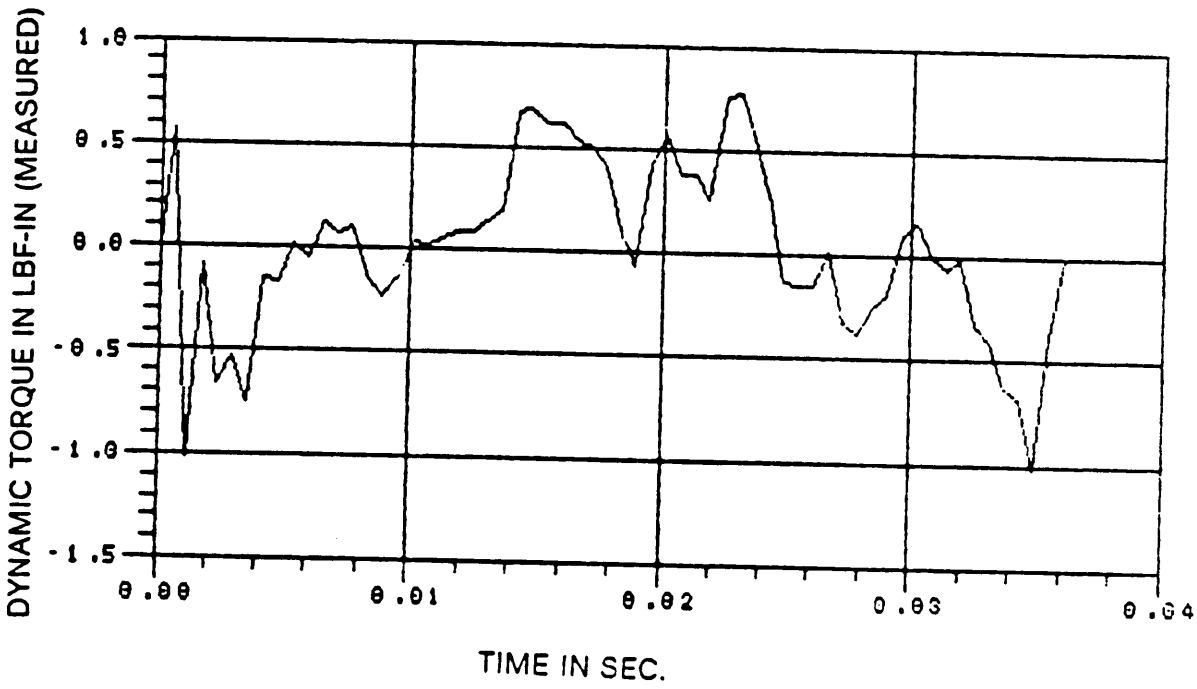


Figure 22. The measured torque curve from unfiltered acceleration signal

3.3 The Compressor Experiment

Dynamic torque measurement using the shaft encoder method was also applied to an existing motor-compressor system (Fig. 23). A Baldor ™ 3 phase induction motor drives the 2 stage Ingersoll-Rand ™ model IR242 compressor. The compressor normally runs at 790 rpm, developing an output pressure of 100 psig. The shaft encoder is included in the system by a belt drive arrangement as shown schematically in Fig. 23. The belt pulleys on the encoder and the compressor/motor shaft are identical in size and are connected by an inextensible belt.

The dynamic model of this motor-compressor system has been formulated into a computer simulation program by Mitchiner [20]. The actual pressures developed in the two stages of the compressor are measured for one complete cycle using pressure transducers. This data is input to the simulation program to compute the net dynamic torque on the motor-compressor system.

3.3.1 Computations for Compressor Experiment

To determine the dynamic torque on the system, the net mass moment of inertia of all the rotating/reciprocating masses in the system has to be determined. Fig. 24 shows the moving parts of the compressor. Fig. 25 shows the rotating/reciprocating masses in the system, schematically.

Inertia of the motor

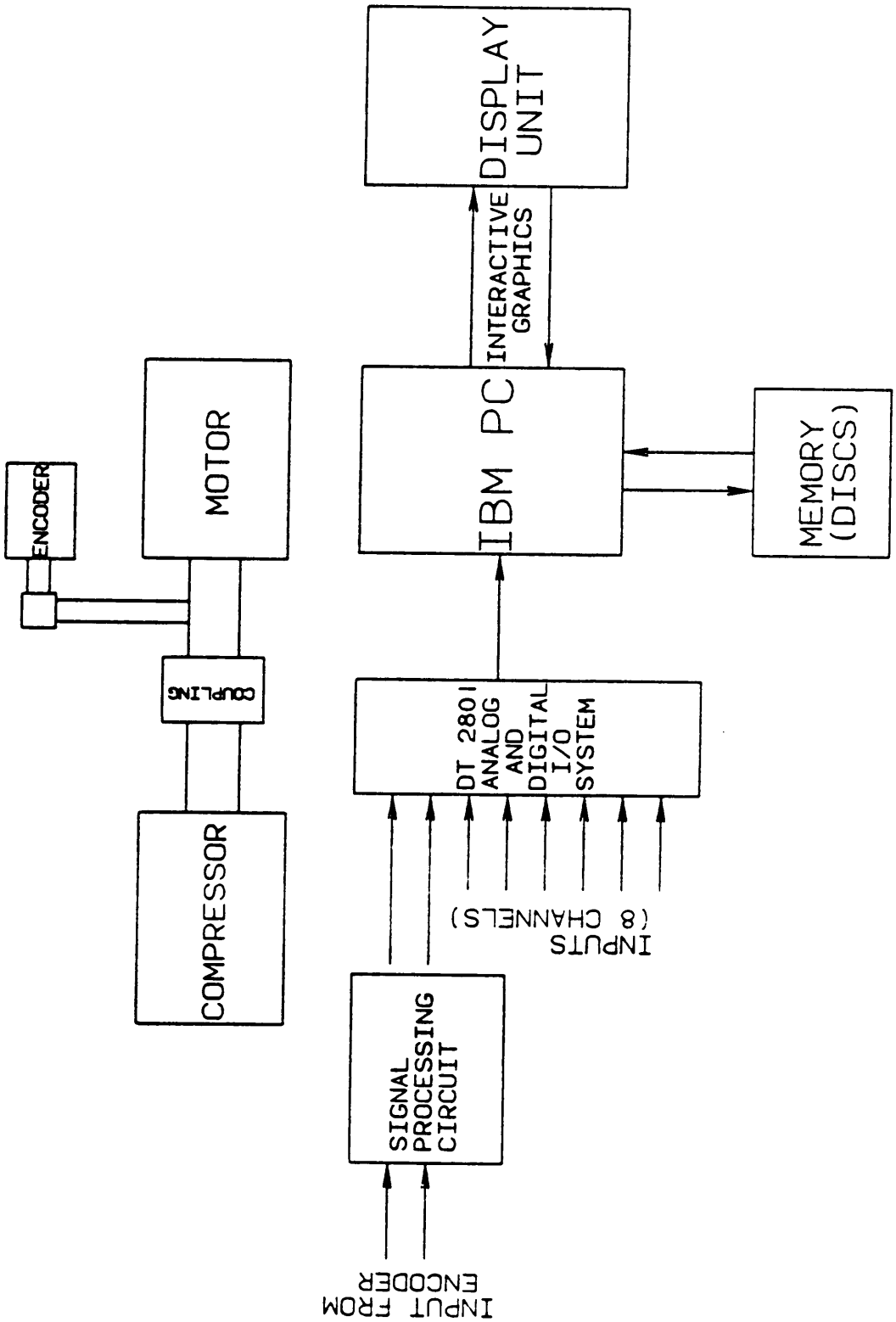


Figure 23. Schematic of Compressor experiment

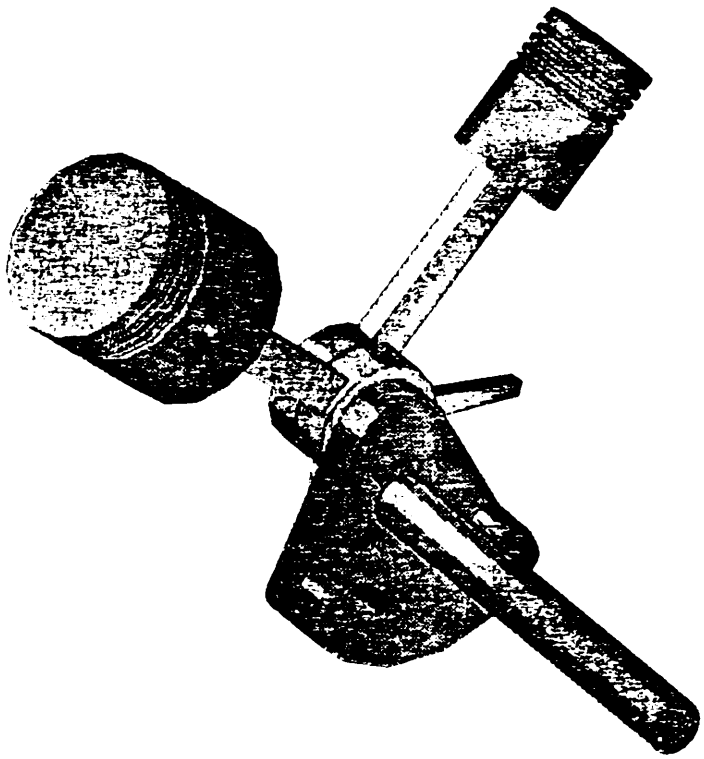


Figure 24. The Ingersoll Rand model IR242 compressor

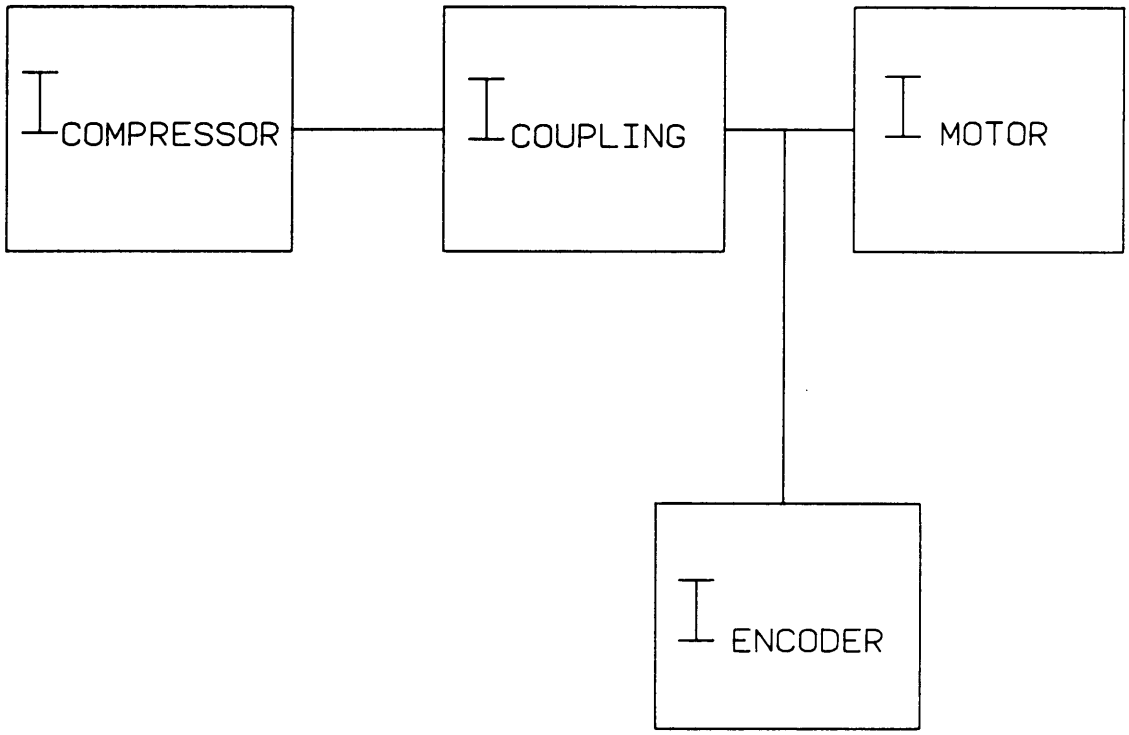


Figure 25. Rotating masses of the compressor experiment

The following experimental method was employed to determine the mass moments of inertia of the motor and the coupling. Fig. 26 shows the experimental set-up used and the free body diagram of this system is shown in Fig. 27.

The equation of motion for the experimental set-up is:

$$J_T \ddot{\theta} = -(k_1 L\theta)L - (k_2 L\theta)L$$

Substituting $k_1 = k_2 = k$, and re-writing

$$J_T \ddot{\theta} + 2kL^2\theta = 0 \tag{3.31}$$

Where

J_T = mass moment of inertia of the total system
k = stiffness of the extension spring
L = moment arm

Therefore,

$$\omega_N = \frac{1}{2\pi}\sqrt{\frac{2kL^2}{J_T}} \text{ Hz} \tag{3.32}$$

The stiffness k of the spring used in this experiment was theoretically computed [28] from

$$k = \frac{d^4 G}{8D^3 N}$$

where

d = diameter of wire, 0.055 in

G = Shear modulus, 11.85×10^6 psi for music wire

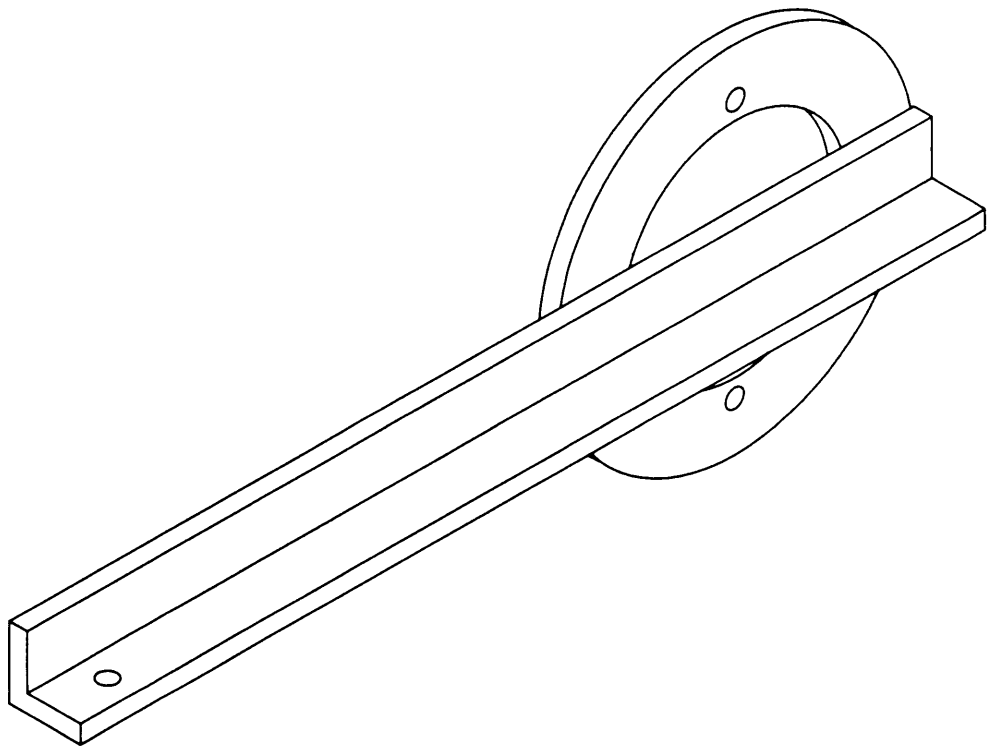
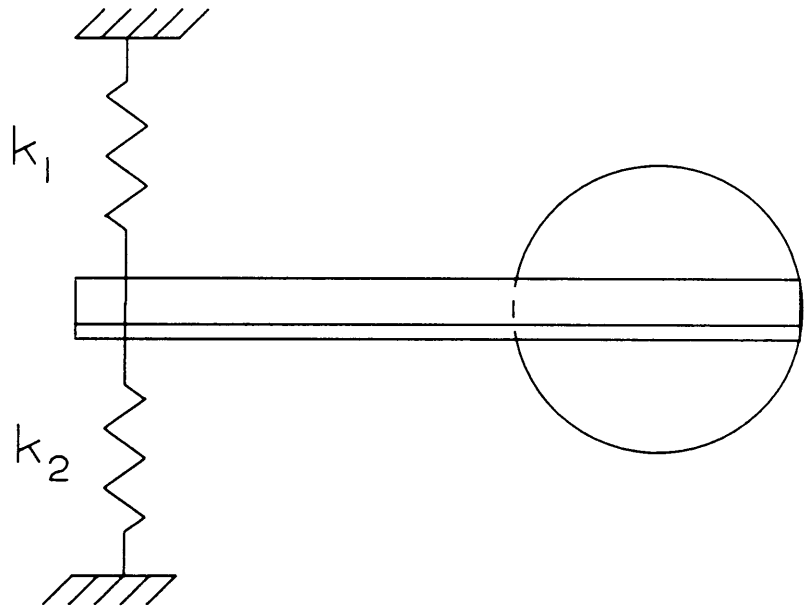


Figure 26. The fixture for finding inertia of motor



FREE BODY DIAGRAM

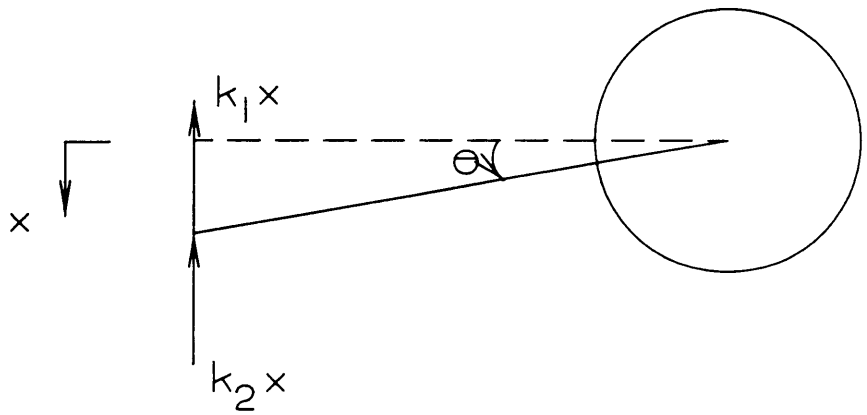


Figure 27. Free body diagram for the fixture

D = mean spring diameter, 0.445 in

N = number of coils, 8

Substituting these values

$$k = 19.23 \text{ lbf/in}$$

Also, L = 13.50 in.

Substituting the known values in Eq. 3.32 and re-arranging:

$$J_T = \frac{2kL^2}{4\pi^2\omega^2} \text{ lbf-in-sec}^2 \tag{3.33}$$

An accelerometer was mounted at the tip of the L-arm (Fig. 26) plate and the arm was displaced by a gentle tap with a hammer, in the vertical plane. The voltage signal from the accelerometer was amplified and input to the A/D (Analog to Digital) card on the personal computer. The time period for one cycle was determined on the personal computer using the data acquisition system and recording the acceleration. Thus the natural frequency of oscillation was determined. Substituting ω_N = 28.0 Hz in Eq. 3.32.

$$J_T = 0.227 \text{ lbf-in-sec}^2$$

Now

$$J_{M,C} = J_T - J_{LF} - 2J_{ES} \tag{3.34}$$

where

$J_{M,C}$ = mass moment of inertia of motor and coupling
J_{LF} = mass moment of inertia of L-arm & flange
J_{ES} = mass moment of inertia of extension spring

From Fig. 26

$$J_{LF} = J_{\text{L-arm}} + J_{flange}$$
$$= \frac{1}{3}mL^2 + \frac{1}{2}m(r_o^2 - r_i^2)$$
$$= \frac{1}{3} \times 2.936 \times 10^{-3}(13.5^2) + \frac{1}{2}3.412 \times 10^{-3}(2.5000^2 + 1.5625^2)$$
$$= 0.1829 \text{ lbf-in-sec}^2$$

$$J_{ES} = \frac{1}{3}mL^2$$
$$= \frac{1}{3}2.2 \times 10^{-5}(13.5^2)$$
$$= 1.318 \times 10^{-3} \text{ lbf-in-sec}^2$$

Substituting these in Eq. 3.34

$$J_{M,C} = 0.227 - 0.1829 - (2 \times 1.318 \times 10^{-3})$$
$$= 0.042 \text{ lbf-in-sec}^2$$

Now

$$J_S = J_{M,C} + J_E + J_C$$

where

J_S = mass moment of inertia of the entire system
$J_{M,C}$ = mass moment of inertia of motor and coupling
J_E = mass moment of inertia of the shaft encoder
J_C = mass moment of inertia of the compressor

The mass moment of inertia of the Ingersoll-Rand ™ compressor is available from a previous experiment [28].

Therefore

$$J_S = 0.042 + 1.563 \times 10^{-4} + 0.2202$$
$$= 0.262 \text{ lbf-in-sec}^2$$

3.3.2 Results of the Compressor Experiment

A simulation program written by Mitchiner [20], which simulates the Ingersoll-Rand two stage air compressor was run on a Digital Equipment VAX 11/780 to determine the computed dynamic torque.

In the experimental part, the encoder output was input to the instrumentation system as in the case of the spring load experiment. Fig. 28 shows the measured angular velocity signal. The dynamic torque waveforms of the theoretical model and the experiment are shown in Fig. 29. The spectra of the same signals are shown in Fig. 30.

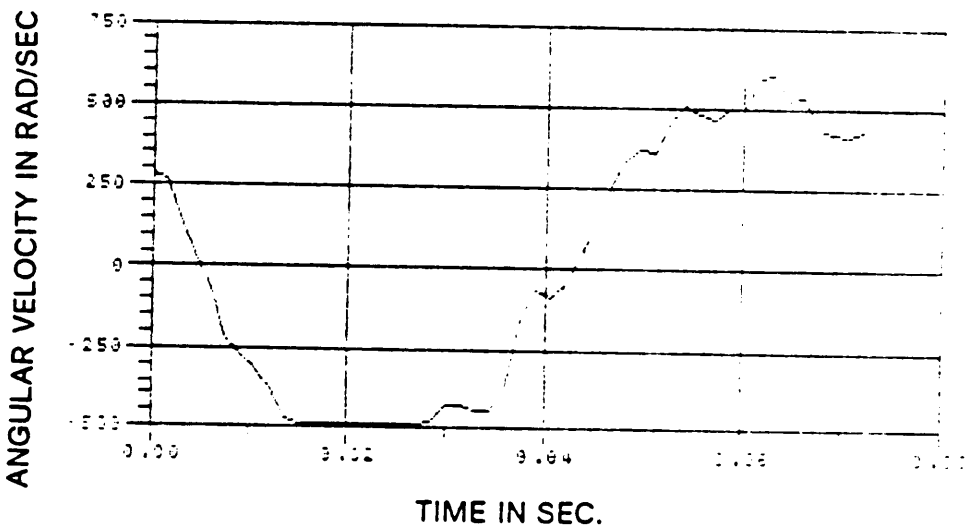


Figure 28. The measured velocity signal

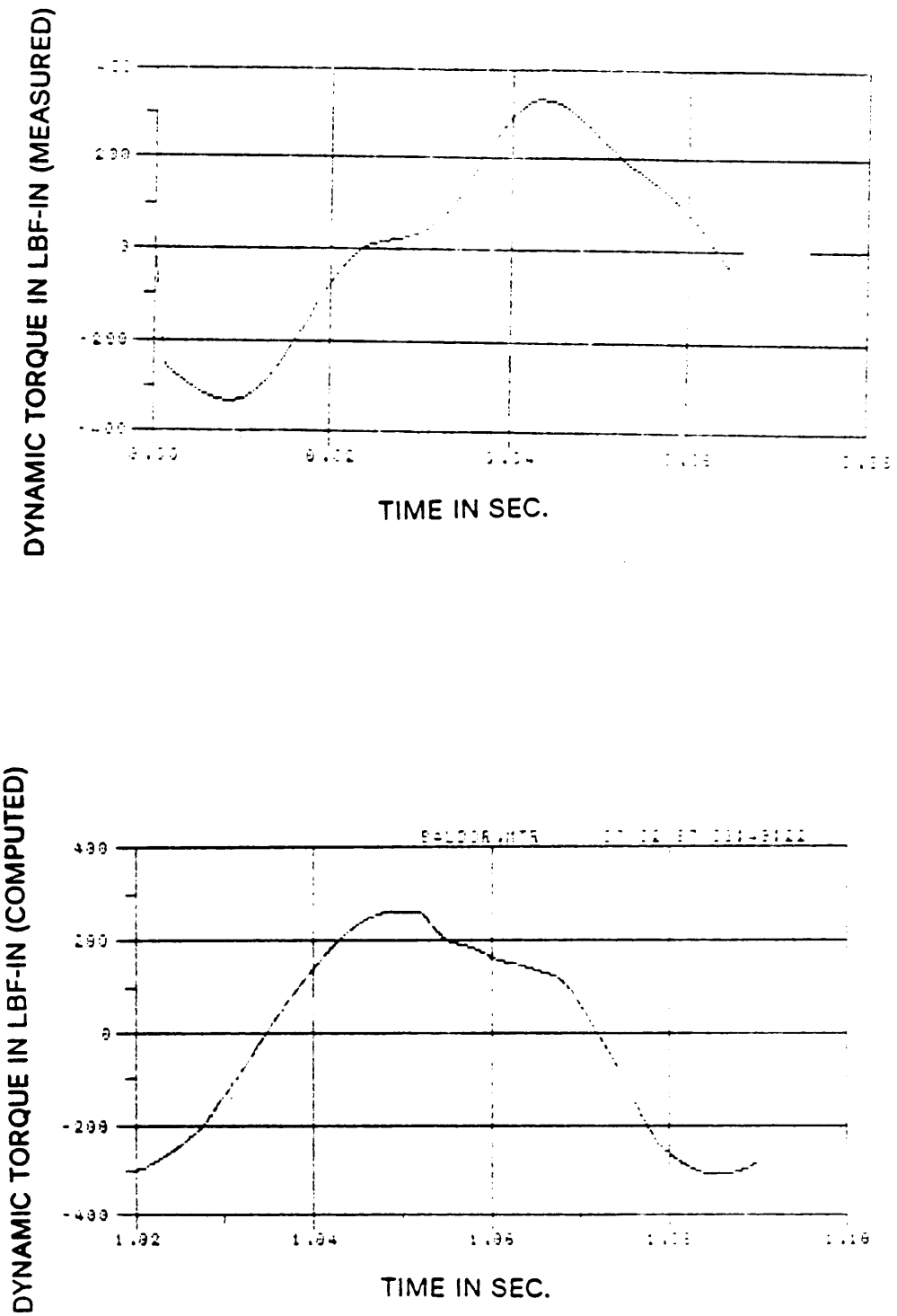


Figure 29. The measured and computed torque waveforms

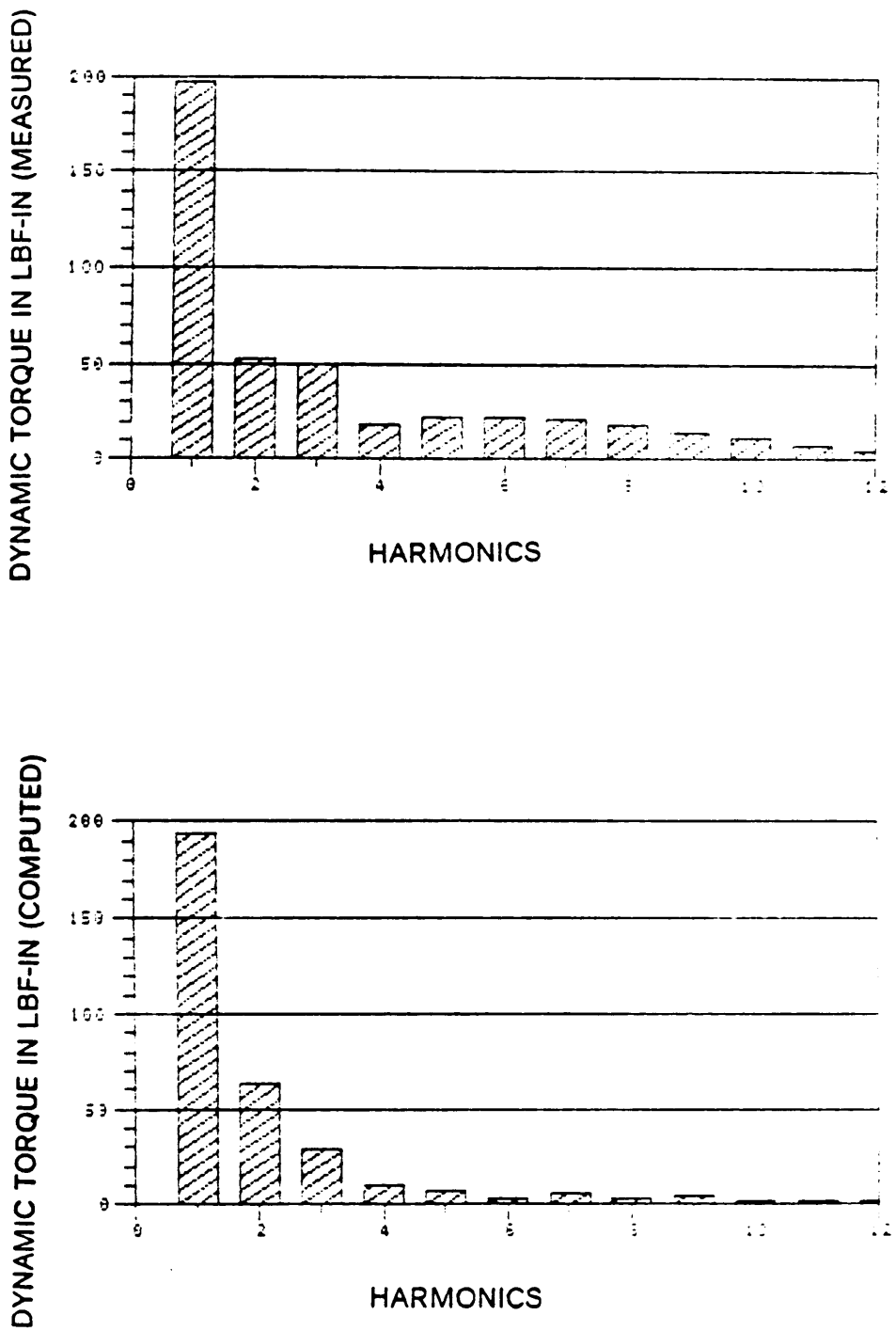


Figure 30. The measured and computed torque spectrums

Chapter 4

Tools of the Experiment

This chapter delineates the experimental set-up used in this investigation. An introduction to the overall set-up is followed by explanation of each component of the experimental set-up in independent sections.

4.1 Introduction

A computerized data acquisition and processing system was developed and interfaced with a shaft encoder. The "brain" of the data acquisition system was an IBM personal computer. A signal pre-processor and a Data Translation ™ Analog and Digital Input/Output system for data acquisition were the hardware used. PCLAB machine language routines and FORGRF graphics package were used with FORTRAN

for data processing and displaying the results on the IBM personal computer monitor. Fig. 31 is a schematic diagram of the data acquisition system.

4.2 The Shaft Encoder

The fundamental principle of encoders is to change angular rotational movement into a series of digital pulses, usually of a square waveform, which can be used to measure, monitor, or control a system.

There are many ways to classify encoders. Based on sensor types, there are brush and optical encoders. Based on functions, there are linear and angular encoders. Based on coding schemes, there are incremental and absolute encoders.

Brush and Optical Encoders

In the brush type shaft encoder, a reed switch acts as the sensor. Alternate North and South poles placed around the encoder shaft make and break the reed switch depending on the polarity of the magnetic field passing at any given instant. But, as the number of pulses per turn of the encoder shaft is limited by physical constraints, the resolution which can be achieved is low. This type of encoder has become more or less obsolete.

For high resolution, optical scanning is the most popular system. Optical techniques rely on the photosensor-detected lighted beam reflected from, or passing through, a radially-stripped disk mirror or plate. Another method makes use of the light beam

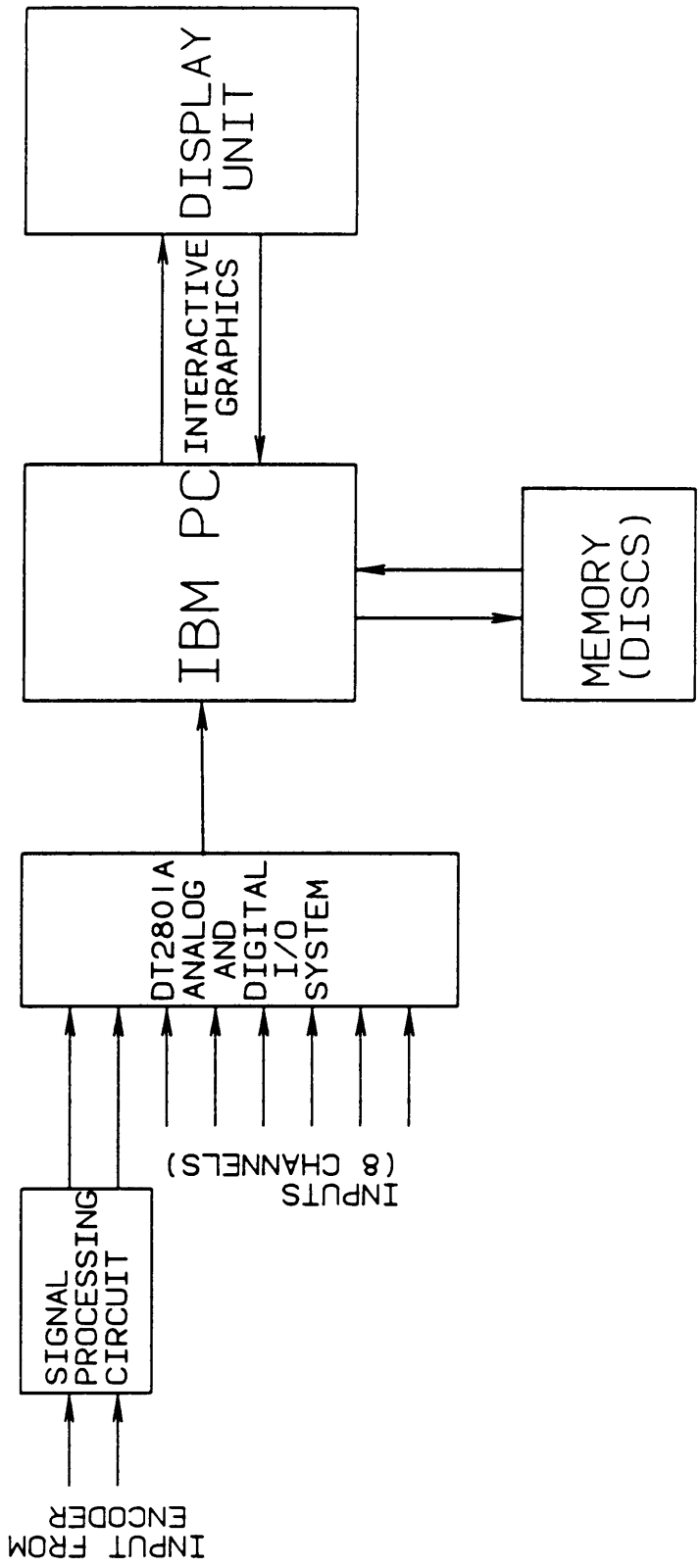


Figure 31. Schematic of Data Acquisition System

modulated by the moving fringes of Moire optical gratings [21]. Even though optical system has the highest resolution, it is not as reliable as the magnetic system under extreme working conditions [19].

Linear and Angular Encoders

While linear (encoded straight) encoders find frequent applications, their limited length is surpassed by angular encoders in whose case a circular to linear transposition of the disc, revolution after revolution, can rack out miles of measured goods [3]. The angular encoder counts equally spaced radii representing quantized angular increments around the periphery of a disc.

Incremental and Absolute Encoders

The incremental encoder is characterized by short code words whose range spans only a fraction of a revolution. The same code pattern is repeated many times in the encoding disc. If absolute position must be derived from the output of an incremental encoder, external counting facilities as well as marker (referencing) pulse for resetting the counter must be provided. A disadvantage of this method is that electronic counters lose count when power goes off, hence requiring either a battery backup or resetting at reference location after each power outage. On the other hand, absolute encoder assigns a distinct code word for each increment. These distinct code words may be decoded for position indication using external combinational logic circuits. While it can easily recover its reading after a power outage, it requires longer code words which translate to more complicated encoding disk and sensing circuits.

In the present investigation, a *Model 714 ACCU-CODER* encoder manufactured by the Encoder Products Co. is used. It is an incremental shaft encoder (optical type) that provides two square wave outputs with constant pulse widths as the encoder shaft is

rotated. One is a 600 pulses per revolution (PPR) signal and the other, a 1 PPR signal. The 1 PPR is used as the reference signal and gives a measure of the average rotational speed of the shaft. The 600 PPR signal is used to determine the variations in the rotational speed, which is $\dot{\theta}$, the angular velocity of the shaft.

4.3 Signal Pre-processing

The 1 PPR and 600 PPR signals from the shaft encoder have to be processed before they can be input to the A/D card. For this reason a phase locked loop with a range switch, an F/V (Frequency to Voltage) converter, an active filter, and a flip flop are used (Fig. 32). These are described in the following sub-sections.

4.3.1 Phase Locked Loop

The 600 PPR signal from the shaft encoder, which is an FM (Frequency Modulated) signal, has to be filtered of additive noise and converted into a proportional voltage such that the voltage represents the angular velocity ($\dot{\theta}$) of the encoder shaft (Fig. 33). This is achieved by employing a phase locked loop circuit.

A phase locked loop consists of three basic components (Fig. 34): A phase detector, a loop filter and a voltage controlled oscillator (VCO).

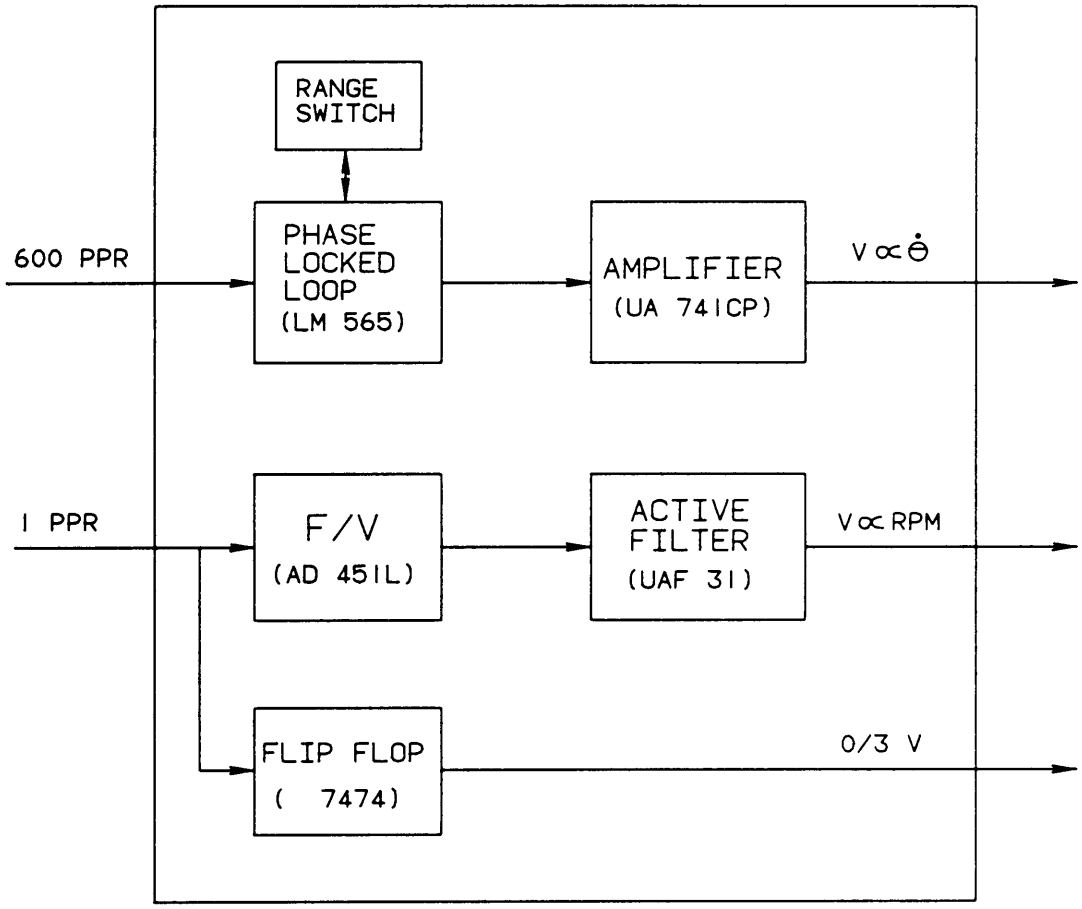


Figure 32. Schematic of signal pre-processing circuitry

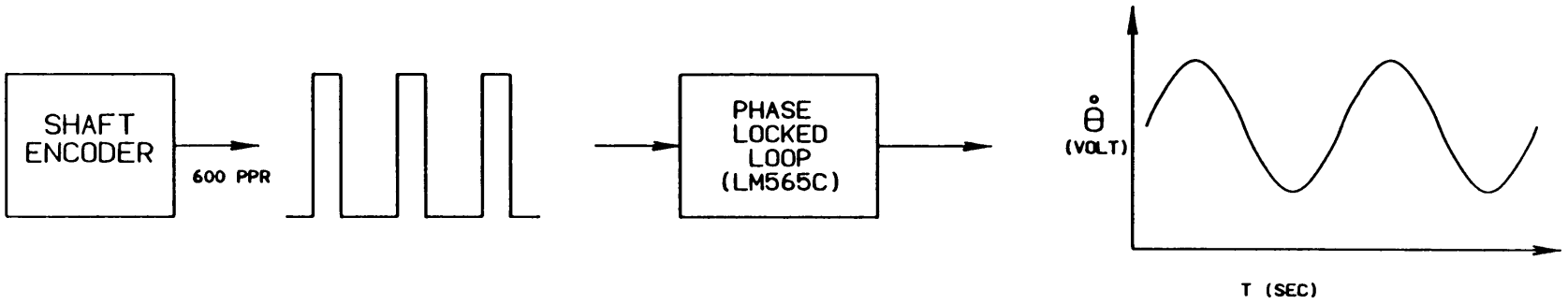


Figure 33. Function of phase locked loop

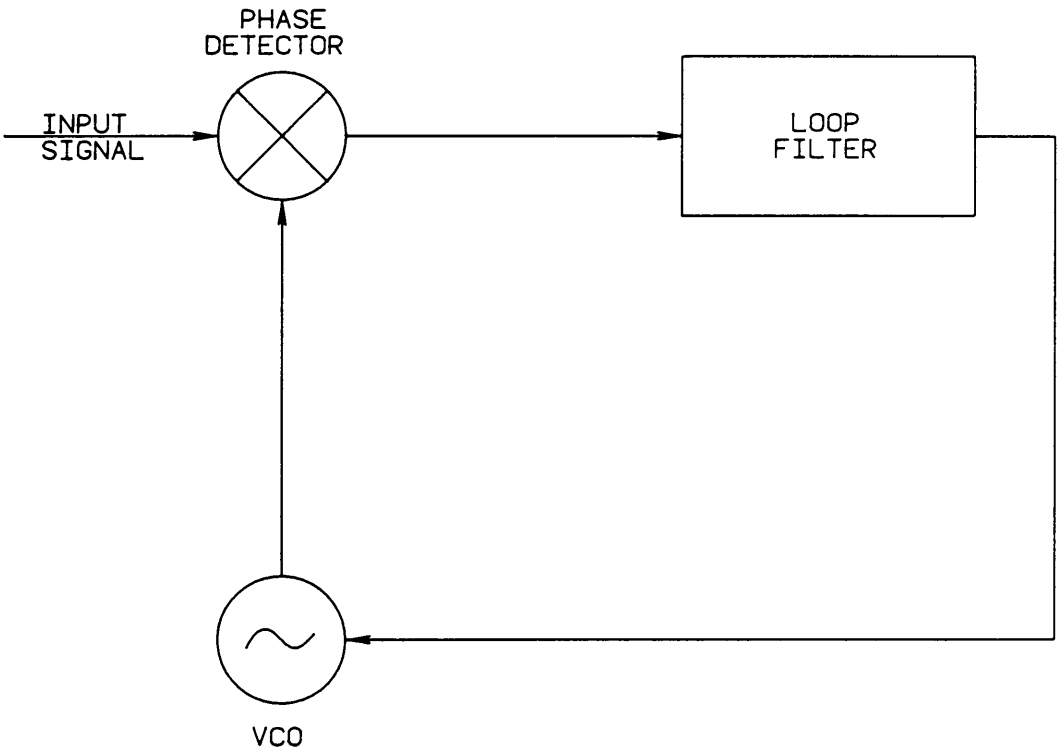


Figure 34. Schematic of phase locked loop

The phase detector compares the phase of a periodic input signal against the phase of the VCO; output of the phase detector is a measure of the phase difference between its two inputs. The difference voltage is then filtered by the loop filter and applied to the VCO. Control voltage on the VCO changes the frequency in a direction that reduces the phase difference between the input signal and the local oscillator [11].

When the loop is *locked*, the control voltage is such that the frequency of the VCO is exactly equal to the average frequency of the input signal. Thus the phase locked loop reproduces the original FM signal in voltage form while removing as much of the noise as possible.

In this investigation, National Semiconductor Corporation's model LM565CN Phase Locked Loop is used. As against the conventional circuit just described, the output from the phase locked loop is taken right from the loop filter, making it open loop. This is because it is only the correction signal, which represents angular velocity, that is required. Also a range selector switch is used to couple the circuit with different capacitors, depending on the frequency range of the input signal. The output voltage signal which is proportional to the angular velocity ($\dot{\theta}$) of the shaft, is input to channel 0 of the A/D (Analog to Digital) card.

Calibration of output signal from phase locked loop

The output from the phase locked loop is a voltage signal proportional to the angular velocity ($\dot{\theta}$) of the encoder shaft. For analysis purposes, this voltage has to be interpreted in rad/sec. Hence the need for calibration.

A schematic of the calibration scheme is shown in Fig. 35. Using this arrangement, known frequencies which simulate the 600 PPR FM signal from the shaft encoder are input to the phase locked loop. The response from the A/D card, as displayed on the personal computer monitor is measured to find the calibration factor i.e., $V/\dot{\theta}$ relation.

First the VCG (Voltage Controlled Generator, a Wavetek ™ model 110) is made to produce a square pulse of known frequency. Then a known dc voltage is input to the VCG. The output frequency from the VCG is a resultant of the dial setting and the input voltage. Thus, the increase in frequency generated per unit voltage is determined. Then the dc voltage input to the VCG is replaced by a sine function from a function generator (a B & K Precision ™ model 3020). The ratio of the frequencies of function generator to VCG is set to 1:600. The output from the VCG is connected to an oscilloscope (to keep track of the magnitude and frequency of the signals) and the phase locked loop. In this arrangement, the output from the VCG represents the 600 PPR of the shaft encoder and the sine function from the function generator represents the reference signal with respect to which the 600 PPR signal is generated.

For each setting of frequencies on the function generator and VCG, the voltage signal on the personal computer was recorded. Three different capacitors were used in the phase locked loop circuitry with the aid of a switch. Depending on the position of the switch, only one capacitor was included in the circuit, providing a linear output over a range of input frequencies, namely, 600 to 1200 rpm, 1000 to 2400 rpm, and 2000 to 4500 rpm.

Thus the 600 PPR output from the shaft encoder, which is an FM signal, is converted to a proportional AM (amplitude modulated) signal which is a dc voltage. Multipliying this with the calibration factor gives angular velocity of the rotating shaft in rad/sec.

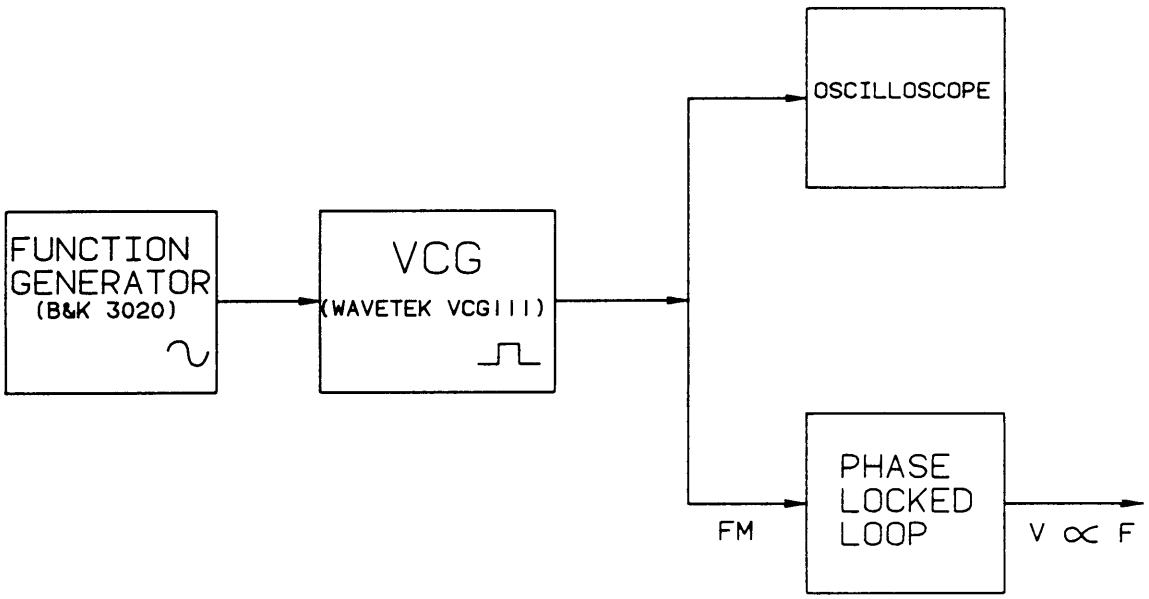


Figure 35. Schematic of the calibration scheme

This velocity waveform which is in the time domain, is analyzed into its constituent harmonics in the frequency domain by using an FFT algorithm and differentiated to obtain angular acceleration.

4.3.2 F/V Converter

As mentioned earlier, the 1 PPR signal from the shaft encoder is the reference signal which represents the average speed of the encoder shaft. Analog Device's ™ model 451L F/V (Frequency/Voltage) converter is used to obtain output voltage (dc) which is an AM signal, proportional to the 1 PPR FM signal (Fig. 36).

This signal is filtered using a Burr Brown ™ model UAF31 active low pass filter to attenuate the high frequency noise. The output voltage signal is input to channel 2 of the A/D card.

A 74LS74 flip flop is used (Fig. 32) to sample for exactly one revolution of the encoder shaft. This flip flop reads in the 1 PPR signal and for each revolution of the encoder shaft, it keeps switching between states, 0 V and 3 V. This signal is input to channel 1 of the A/D card. As exactly one cycle of encoder shaft rotation is recorded with 50 data points in equal intervals of time, the phase lag of the evaluated acceleration signal with respect to the measured velocity signal is of the order of one tenth of one percent and hence negligible.

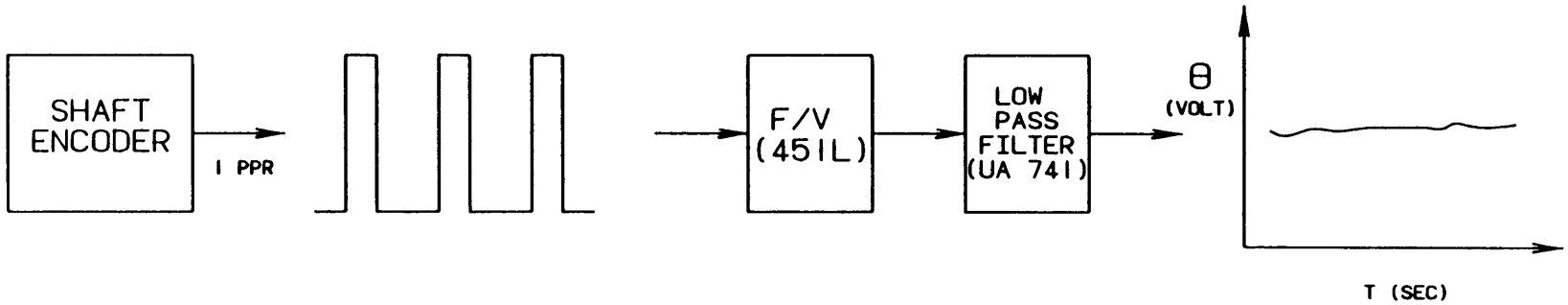


Figure 36. Function of F/V converter

4.4 Analog and Digital I/O System

The DT2801-A [6], manufactured by Data Translation Inc., is a low cost, high performance analog and digital I/O system interface, compatible with the IBM personal computer. Full I/O functionality is provided on a single printed circuit board (Fig. 37). This board can be plugged into one of the system expansion slots in the IBM personal computer backpanel.

The DT2801-A board includes a 12-bit analog/digital (A/D) converter system for 8 differential or 16 single ended input channels. It also includes two 12-bit digital to analog (D/A) converters. These D/A converters can provide either sequential outputs (their analog outputs change at different times) or simultaneous outputs (the outputs of both D/A converters change at the same time). In addition, the DT2801-A board features two 8-line digital I/O ports (which can be used separately to read or write 8-bit transfers, or simultaneous for a 16-bit transfer). A programmable gain amplifier provides selectable gains of 1, 2, 4, and 8 which allow the user to accommodate a 20mV to 10V full scale range of input signal levels.

The board also contains an on-board programmable clock, which can be used to provide clock pulses to control the operations of the board's A/D and D/A subsystems. This internal clock can be programmed to operate in the clock period ranging from 37.5 microseconds to 5 seconds in 1.25 microsecond increments, and can be used for internal timing of both A/D and D/A conversions. The DT2801-A also provides for an external clock, which permits the synchronization of block mode A/D and D/A conversions with an external frequency source. An external trigger input allows

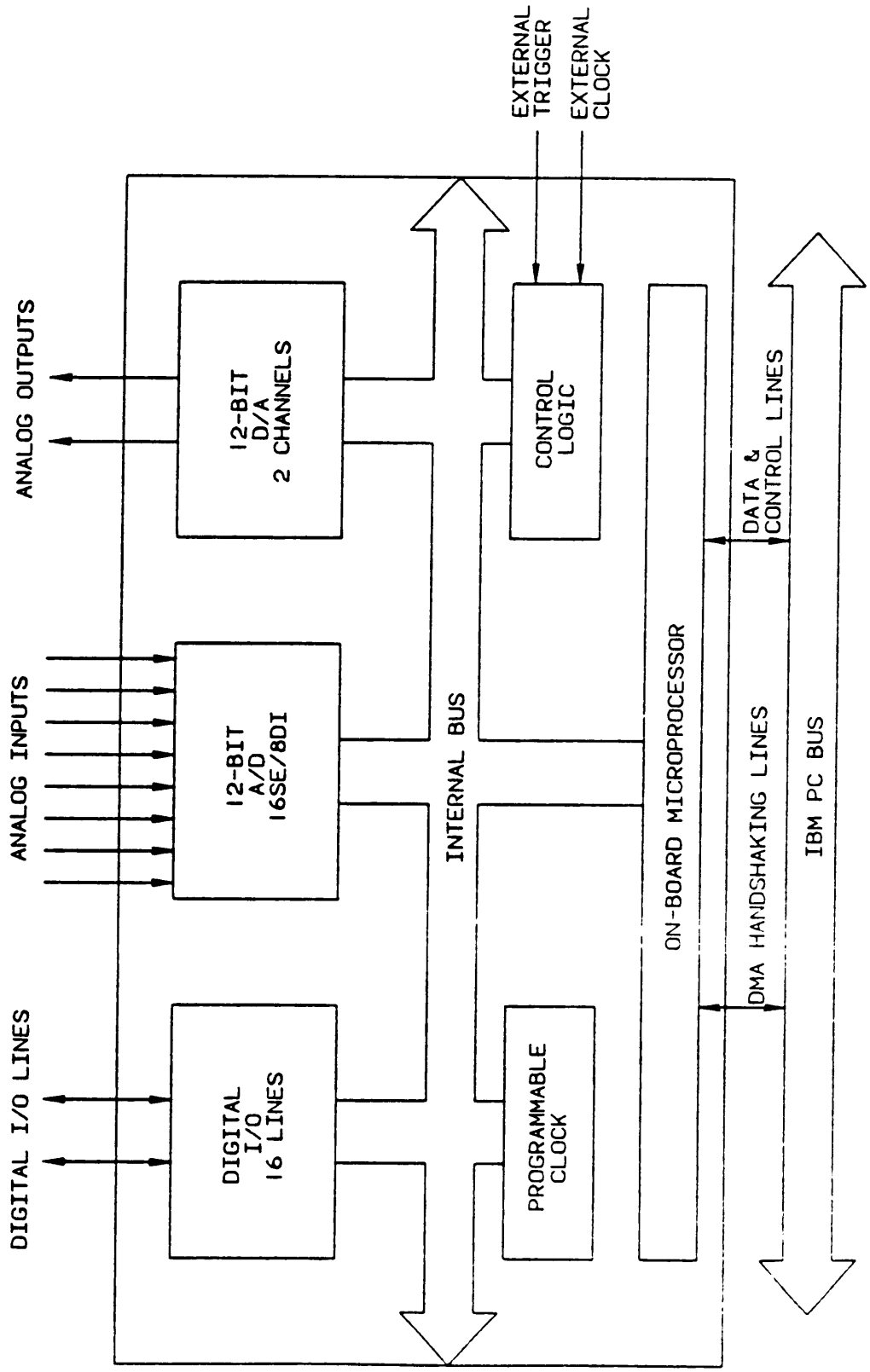


Figure 37. Schematic of DT2801-A Board

the user to synchronize an I/O event with an external event by deferring command execution until an external trigger pulse is received. An on-board DC to DC power converter generates all the necessary analog supply voltages from the 5V power supply available on the IBM personal computer backpanel, thus providing high noise isolation from the computer system's power supply.

This I/O card is designed around an on-board microprocessor which acts as the interface between the DT2801-A and the IBM personal computer. This microprocessor controls all on-board operations and simplifies program control by the IBM personal computer. Routine tasks, such as A/D converter sequencing and error checking are carried out by this processor. The DT2801-A can be easily programmed in any language that accesses the IBM personal computer's I/O registers. Microcoding within the on-board controller greatly simplifies user access to the board's functions for use with DT2801-A series boards.

This is done by PCLAB, a real-time software package from Data Translation Inc., designed for use with DT2801-A series boards [6]. The PCLAB libraries support analog and digital transfers both into and out of the IBM personal computer. The software package allows the user to manipulate the interface board through operation-specific routine calls. PCLAB consists of libraries of subroutines that can be called from a number of Microsoft languages supported by MSDOS [7]. These languages are Microsoft C, FORTRAN, PASCAL, BASIC and assembly language. In this investigation FORTRAN was used and the code is listed in Appendix B. With the hardware capabilities of DT2801-A card and with the aid of PCLAB routines, the operating range of the different parameters in the instrumentation are as follows:

Number of sampling points = 25 - 250

Range of sampling time = 0.005 - 1.000 sec

Number of channels used for sampling = 3

Range of voltage sensed = +/- 10 V

Resolution (word size) = 12 bits

Chapter 5

Program Development

This chapter first presents the commercial software used in this investigation followed by a brief description of the programs written.

5.1 Overview

Recent computer hardware and software (for modelling, visual display, processing, etc.) have enabled engineers to make decisions based on the interactive graphic techniques. The computational analysis of a design problem or the construction of a drawing can be carried out using a dialogue, with the instructions being carried out immediately and the result being displayed on a video display unit.

The personal computer is a powerful tool in many Computer Aided Engineering applications. It combines easy accessibility/control with economy. In this investigation, the programming has been done in ANSI FORTRAN 77. For monitoring data acquisition, machine language routines have been used. FORGRF is the graphics package used for displaying plots.

PCLAB is a real-time software package from Data Translation Inc., designed for use with DT2801-A series boards [6]. The PCLAB libraries support analog and digital transfers both into and out of the IBM personal computer. The software package allows the user to manipulate the interface board through operation-specific routine calls. PCLAB consists of libraries of subroutines that can be called from a number of Microsoft languages supported by MSDOS [7]. These languages are Microsoft C, FORTRAN, PASCAL, BASIC and assembly language. In this investigation FORTRAN has been used and the code is listed in Appendix B. AG II (Advanced Graphing II) is a software package designed for use with the Tektronix ™ 4010A01 PLOT 10 Terminal Control System. This package is the advanced version of the original Advanced Graphing package consisting of FORTRAN subroutines. The personal computer version of this package is called FORGRF. FORGRF has been used in this investigation for displaying and manipulating (such as zooming) plots.

5.2 *Program SPRING*

SPRING is the simulation program written in FORTRAN 77 for the spring load experiment. This program simulates the motion of the central mass on the spring for the

spring load experiment. With the position of the central mass defined in time, the time varying torque (dynamic torque) on the motor shaft may be computed.

A flow diagram of the program is shown in Fig. 38. The program consists of a single loop in which a numeric integration routine is called to solve the differential equation of motion for the spring load system. The integrator used is a fourth order Runge-Kutta algorithm with a fixed time step. This algorithm requires a routine that computes the derivative of the desired function. This routine determines all the force applied by the spring. At each time step, the spring moves a finite amount based upon the equation of motion and the value of the time step. As the simulation progresses, the integration routine computes the orientations of the moving lumped mass centers of the spring in terms of x and y coordinates. The integration process continues until a user-specified time interval is reached.

The user inputs the time interval over which integration is desired. The program gives graphical output of the computed dynamic torque waveform. This torque waveform is further analyzed to display the spectrum of the first 12 harmonics using a program that employs an FFT algorithm.

5.3 *Program TORQUE*

The 600 PPR (Pulses Per Revolution) signal from the shaft encoder is pre-processed and is then input to the computer from the A/D card. The input signal which is a voltage signal is multiplied by the calibration factor to obtain angular velocity in

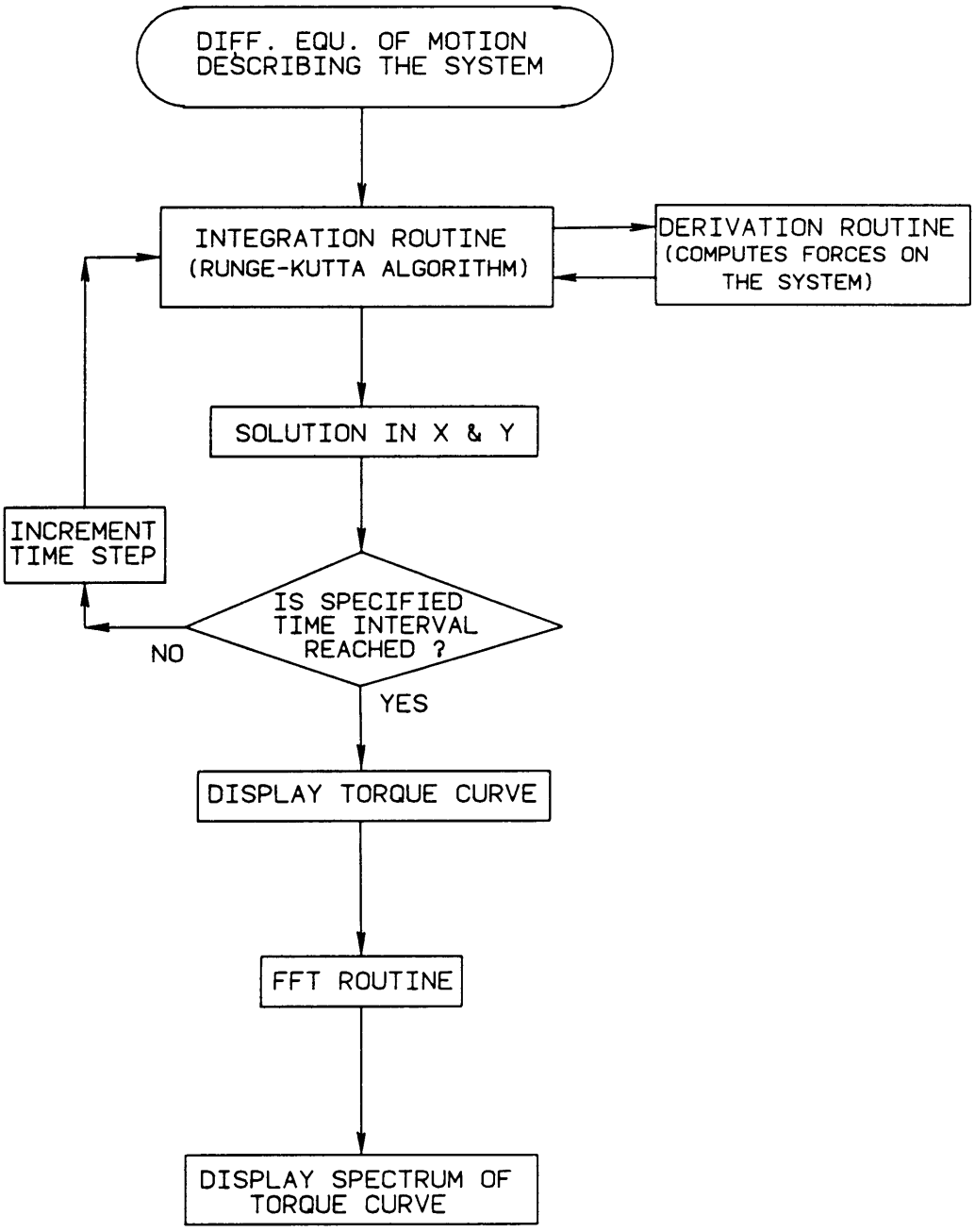


Figure 38. Flow Diagram for Program SPRING

rad/sec. This velocity signal which is in the time domain is analyzed into its first 12 harmonics in the frequency domain using an FFT algorithm and the user is prompted to filter (remove) the harmonics, if any. The velocity signal is then differentiated in the frequency domain to obtain the angular acceleration. The user again has the option of filtering any desired range of harmonics. The filtered acceleration is then multiplied by I, the mass moment of inertia of the rotating system, to obtain the dynamic torque.

Chapter 6

Conclusions and Recommendations

This study has demonstrated a Computer Aided Engineering technique of dynamic torque evaluation. An optical shaft encoder was used as the sensing element and its output was digitally processed to yield records of angular velocity, acceleration and dynamic torque for one revolution of the rotating shaft. The method has been applied to the determination of dynamic torque in two experiments.

First, the instrumentation system was used on a spring load system. The records obtained were accurate, as these results were within 5% of the results obtained from the theoretical model.

The instrumentation system was also used on an existing two-stage air compressor. The dynamic torque determined by the instrumentation system was within 5% of the dynamic torque computed in the simulation program.

The instrumentation system proved to be very accurate and reliable. Also, since the system is personal computer based, the whole set-up is portable. A simple transmission system such as a belt drive arrangement will suffice to couple the shaft encoder to the shaft of interest.

The accuracy of this method of dynamic torque determination is constrained by the accuracy with which the mass moment of inertia of the rotating machinery can be determined. However, it can be assumed in most cases that the mass moment of inertia is known, as it is an important parameter in the dynamics of machinery.

Recommendations for future research

Since the shaft encoder can be placed at any convenient point in the rotating system to sense motion in the rigid shaft of interest, this method can be considered as a "contactless method" of determining dynamic torque. This distinct advantage makes it a powerful tool for industrial applications as well as in the study of dynamics of rotating machinery.

This study used a motor-compressor set-up to evaluate the dynamic torque in the system. A thorough mathematical model can be developed to represent the dynamic process which includes components to account for the pressure developed in the compressor, leakage in the valves and piston rings, friction, vibrational effects, etc. in terms of angular position, velocity and acceleration. Then the angular position and velocity measured by the encoder and the angular acceleration determined there from can be used to drive this mathematical model, in real-time, to detect faults, if any, in the machinery. Mathematically, a generalized system equation would be of the form:

$$A\ddot{\theta} + B\dot{\theta} + C\theta + D = 0$$

where the constants A,B,C and D represent the mechanics of operation of the compressor as discussed above and θ ,$\dot{\theta}$ and $\ddot{\theta}$ which represent angular position, velocity and acceleration respectively, can be measured using the encoder based instrumentation system. By formulating a set of simultaneous equations of the above form, they can be solved for the constants A,B,C and D, thus developing a comprehensive, real-time diagnostics system for a compressor.

In this investigation, the encoder technique was developed and tested on two experimental set-ups, both having a single drive shaft. The dynamic torques were determined based on lumped mass models. The lumped mass approach can also be used to model more complex arrangements involving a multiplicity of drive shafts for example, in the multiturbine generator units used in large power stations.

This method of torque determination can also be used in in-process detection of tool breakages in machine tools. Matsushima and Sata [17] have demonstrated that there is a definite pattern in the variation in cutting torque when the cutting-tool breaks. Specifically, these authors have found that the breakage of the cutting tool tip causes momentary decrease and successive increase of the cutting torque. Thus by continuously measuring the in-process torque and comparing it with the tool breakage pattern, tool breakage can be determined in real-time and appropriate action can be taken.

Bibliography

1. Analog Devices, Inc., *Data Acquisition Databook 1984*, Vol. 2, Norwood, Massachusetts, 1984.

2. Auckland, D.W., Sundram, S., Shuttleworth, R. and Posner, D.I., "The Measurement of Shaft Torque using an Optical Encoder", *Journal of Physics E-Scientific Instruments*, Vol. 17, The Institue of Physics Publication, Dec. 1984, pp.1193-1198.

3. Baasch, T.L., "Shaft Angle Encoders", *Electronic Products Magazine*, Vol. 13, Sept. 1971, pp.118-123.

4. Clarke, M. and Parrot, E.J., "Shaft Encoders offer Reliability with high Resolution", *Electronic Engineering*, Vol. 44, Sept. 1972, pp.71.

5. Craddox, M. and Brittain, W., "A Digital Device for Measuring Torque", *Industrial Electronics*, Vol. 3, June 1965, pp.258-261.

6. Data Translation, Inc., *User Manual for DT2801 SERIES*, Third ed., Marlborough, Massachusetts, 1984.

7. Data Translation, Inc., *PCLAB Release Notes*, Version 2.0, Marlborough, Massachusetts, 1984.

8. Encoder Products Co., *ACCU-CODER Shaft Encoder Selection Guide*, Sandpoint, Idaho.

9. Fink, D.G. and Carroll, J.M., Editors, *Standard Handbook for Electrical Engineers*, McGraw-Hill Book Co., Tenth ed., 1969.

10. Fleming, W.J., "Automotive Torque Measurement: A summary of Seven Different Methods", *IEEE Transactions on Vehicular Technology*, Vol. 31, Aug. 1982, pp.117-124.

11. Gardner, F.M., *Phaselock Techniques*, John Wiley & Sons, Second ed., 1979.

12. Gibbs-Smith, C. and Rees, G., *The Inventions of Leonardo Da Vinci*, Phaidon Press Ltd., Oxford, G.B., 1978.

13. Greenwood, D.T., *Principles of Dynamics*, Prentice-Hall, N.J., 1965.

14. Holman, J.P. and Gajda, W.J., Jr., *Experimental methods for engineers*, McGraw-Hill Book Co., Third ed., 1978.

15. Hu, S.C. and Raudkivi, U., "A New Encoding Scheme for an Absolute Digital Encoder in Units of Degrees", *IEEE Transactions on Industrial Electronics and Control Instruments*, Vol. 27, Jan. 1980, pp.5-9.

16. Kuchela, K.S., "Electronics for Shaft Encoder", *Nuclear Instruments and Methods in Physics Research*, Vol. 188, Jan. 1981, pp.247-248.

17. Matsushima, K. and Sata, T., "In-Process Detection of Tool Breakages", *Journal of Faculty in Engineering, Univ. of Tokyo*, Vol. A-17, 1979, pp.20-21.

18. Meirovitch, L., *Elements of Vibration Analysis*, McGraw-Hill Book Co., Second ed., 1986.

19. Midson, R., "Shaft Encoder, Digitizer, Rotary Pulse Generator - Which and When ?", *Control and Instrumentation*, Vol. 8, June 1976, pp.39-40.

20. Mitchiner, R.G. and Leonard, R.G., "The Design of Centrifugal Pendulum Absorbers", *11th Biennial Conference on Mechanical Vibrations and Noise of the ASME*, Boston, Massachusetts, Sept. 1987.

21. Naval, P.C., "Optical Shaft Encoder Processor with Stall Detection Capability", *IEEE Transactions on Industrial Electronics*, Vol. 32, Nov. 1985, pp.449-450.

22. Norton, H., "Force and Torque Sensors", *Handbook of Transducers for Electronic Measuring Systems*, Prentice-Hall, 1969, ch. 5, pp.99-163.

23. Ohigashi, S., Ono, G. and Machida, S., "A New Electronic Device for Measuring Torque", *Society of Automotive Engineers Transactions*, Vol. 74, 1966, pp.226-235.

24. Pratt, G., Jr., "An Opto-elctronic Torquemeter for Engine Control", *Automotive Engineering Congress*, Feb. 1976, SAE Paper 760070.

25. Rogers, K.J., "A Transducer to Measure the Torque Imposed on the Engine of a Passenger Car by Accessory and Ancillary Devices", *Journal of Physics E-Scientific Instruments*, Vol. 16, The Institue of Physics Publication, Nov. 1983, pp.1086-1090.

26. Scoppe, F., "Magnetostrictive Torque Transducer", *Instrumentation and Technology*, Vol. 16, Oct. 1969, pp.95-99.

27. Sherman, G., *A Method for the Spatial Dynamic Simulation of Compressors using the Digital Computer*, Masters Thesis, VPI&SU, 1987.

28. Shigley, J.E. and Mitchell, L.D., *Mechanical Engineering Design*, McGraw-Hill Book Co., Fourth ed., 1983, pp.447.

29. Ward, J., Silvey, J. and Roberge, J., "High Accuracy Torquemeter", *U.S. Patent 3 940 979*, Mar. 1976.

30. Wheatcroft, N., "Shaft Angle Encoder", *Instruments & Control Systems*, Vol. 42, May 1969, pp.131.

Appendix A
Program SPRING

```
C=====================================================================
C
C    PROGRAM        SPRING
C
C    AUTHOR         Ratnakar M. Kanth
C
C    DATE           Spring 1987
C
C    DESCRIPTION    Dynamic simulation of a compression spring, one end
C                   of which is fixed to the ground and the other end
C                   rotating on the circumference of a disc coupled to
C                   a fractional horse power motor
C
C    HARDWARE       DEC VAX 11/780
C                   IBM PC/XT/AT
C
C    SYSTEM         VAX-VMS; MS-DOS
C
C    LANGUAGE       ANSI FORTRAN-77 W/ PLOT 10 AGII
C
C=====================================================================
C
C
      INTEGER        ANS
      REAL           Y(4),DY(4),Q(4),XPLOT(15000)
      REAL           XDIS(15000),YDIS(15000),TORQ(15000)
      REAL           YPLOTX(15000),YPLOTY(15000),YPLOTT(15000)
      DATA           R/.380/,W/167.552/,DISP0/1.820/,SRATE/2.995/
      DATA           PI/3.1415926/
      COMMON         A,Z
      CHARACTER*9    FNAME
C
C
 10   CALL NEWPAG
      WRITE (*,*) ' ENTER START AND END TIME'
      READ (*,*) TSTART,TEND
C      WRITE (*,*)
C      WRITE (*,*) ' ENTER DAMPING FACTOR'
C      READ (*,*) Z
      Z = 5E-05
C      WRITE (*,*) ' ENTER VALUE OF "A" FOR DAMPING'
C      READ (*,*) A
      A = 3.
C
C  Initialize the arrays
C
      DO 20 I=1,4
```

```
          Y(I)  = 0.0
          DY(I) = 0.0
          Q(I)  = 0.0
   20  CONTINUE
C
C  Set the initial conditions
C
       Y(3) = 32.044
       TIME = 0.
       DT = 0.001
       KOUNT = 1
C
C    The above choice for DT gives approx. 37 data points
C    per revolution of the rotor
C
  30   CALL RKG ( 4, DT, TIME, Y, DY, Q )
       IF (TIME .LE. TEND) THEN
         KOUNT = KOUNT + 1
         XPLOT(KOUNT) = TIME
         XDIS(KOUNT) = Y(2)
         YDIS(KOUNT) = Y(4)
         THETA = W*TIME
         FX = -2.*SRATE*(R*COS(THETA)-Y(2))
         FY = -2.*SRATE*(R*SIN(THETA)-Y(4))
         F = SQRT (FX**2 + FY**2)
         DISP = DISP0 + SQRT (XDIS(KOUNT)**2 + YDIS(KOUNT)**2)
         SI = ASIN ((SIN(THETA)*DISP0)/DISP)
         PHI = PI - SI
         TARM = R*SIN(PHI)
         TORQ(KOUNT) = F*TARM
         GO TO 30
       ENDIF
C
C  Initialize plot arrays for x & y displacements and torque curve
C
       KPTS = INT ((TEND - TSTART)/DT)
       KBEG = INT (TSTART/DT) + 1
       DO 40 I=2,KPTS+1
         XPLOT(I) = TSTART
         YPLOTX(I) = XDIS(KBEG)
         YPLOTY(I) = YDIS(KBEG)
         YPLOTT(I) = TORQ(KBEG)
         TSTART = TSTART + DT
         KBEG = KBEG + 1
  40   CONTINUE
C
       XPLOT(1) = REAL (KPTS)
       YPLOTX(1) = REAL (KPTS)
       YPLOTY(1) = REAL (KPTS)
       YPLOTT(1) = REAL (KPTS)
C
C    Display the X & Y displacements in separate graphs
C    in the top half of the screen
C
       CALL INITT (960)
C      CALL BINITT
C      CALL SLIMX (150,480)
C      CALL SLIMY (445,700)
C      CALL CHECK (XPLOT,YPLOTX)
C      CALL DSPLAY (XPLOT,YPLOTX)
C      CALL BINITT
C      CALL SLIMX (570,900)
C      CALL SLIMY (445,700)
C      CALL CHECK (XPLOT,YPLOTY)
C      CALL DSPLAY (XPLOT,YPLOTY)
C
```

```
C   Display the torque curve in the bottom half of the screen
C
      CALL BINITT
C     CALL SLIMX (150,900)
C     CALL SLIMY (125,380)
      CALL CHECK (XPLOT,YPLOTT)
      CALL DSPLAY (XPLOT,YPLOTT)
      CALL TINPUT (II)
      CALL ANMODE
C
C   Prompt the user "What Next ?"
C
 50   CALL ERASE
      CALL ANMODE
      WRITE (*,55)
 55   FORMAT (/,' ENTER THE APPROPRIATE NUMBER:',/,
     1           ' 1  -  DO AGAIN',/,
     2           ' 2  -  SAVE ARRAYS IN A DATA FILE',/,
     3           ' 3  -  END  >> (3) ',$)
      READ (*,60,ERR=50) ANS
 60   FORMAT (I)
      IF (ANS .EQ. 0) ANS=3
      IF (ANS.LT.1 .OR. ANS.GT.3) GOTO 50
      IF (ANS .EQ. 1) GOTO 65
      IF (ANS .EQ. 2) GOTO 80
      IF (ANS .EQ. 3) GOTO 120
C
 65   CALL ERASE
      DO 70 I=1,KOUNT+1
        XPLOT(I) = 0.
        YPLOTX(I) = 0.
        YPLOTY(I) = 0.
        YPLOTT(I) = 0.
 70   CONTINUE
      GO TO 10
C
 80   CALL ERASE
      WRITE (*,*) 'GIVE THE NAME OF THE FILE (<9 CHARACTER)'
      READ (*,90) FNAME
 90   FORMAT (A9)
      OPEN (UNIT=12,FILE=FNAME,STATUS='NEW',FORM='FORMATTED')
      N = INT((KPTS-1)/2)
        J = 2
      DO 110 I=2,N
        WRITE (12,100) XPLOT(J),YPLOTT(J),XPLOT(J+1),YPLOTT(J+1)
100     FORMAT (1X,4F13.8)
        J = J + 2
110   CONTINUE
      GO TO 50

120   CALL FINITT(0,0)

      STOP
      END
C
C
C=====================================================================
C
      SUBROUTINE RKG (NEQ,H,X,Y,DY,Q)
C
C   This routine uses the Runge-Kutta algorithm for integration
C   of 4th order differential equations
C
C=====================================================================
C
      DIMENSION   A(2), Y(NEQ), Q(NEQ), DY(NEQ)
      DATA        A(1) /0.292893218813452475/
```

```
      DATA        A(2) /1.70710678118654752 /
C
      H2 = 0.5 * H
      CALL DERIV ( Y, DY, X )
      DO 10 I = 1, NEQ
        B = H2*DY(I) - Q(I)
        Y(I) = Y(I) + B
        Q(I) = Q(I) + 3.0*B - H2*DY(I)
 10   CONTINUE
      X = X + H2
      DO 30 J = 1, 2
        CALL DERIV ( Y, DY, X )
        DO 20 I = 1, NEQ
          B = A(J)*( H*DY(I) - Q(I) )
          Y(I) = Y(I) + B
          Q(I) = Q(I) + 3.0*B - A(J)*H*DY(I)
 20     CONTINUE
 30   CONTINUE
      X = X + H2
      CALL DERIV ( Y, DY, X )
      DO 40 I = 1, NEQ
        B = ( H*DY(I) - 2.0*Q(I) ) / 6.0
        Y(I) = Y(I) + B
        Q(I) = Q(I) + 3.0*B - H2*DY(I)
 40   CONTINUE
C
      RETURN
      END
C
C=====================================================================
C
      SUBROUTINE DERIV ( Y, DY, T )
C
C   This routine, given the formation for the derivative, will
C   compute the derivative vector, DY, on the basis of the Y vector
C
C=====================================================================
C
      DIMENSION   Y(4),DY(4)
      DATA        OMEGA/167.552/
      DATA        WN/849.520/
      COMMON      A,Z
C
      THETA=OMEGA*T
C
      DY(1) = (-11.98*Y(2) + 2.156*COS(THETA) - 2.*Z*1.66E-5*EXP(-A*T)
     1        *WN*Y(1))*1.66E05
C
      DY(2)=Y(1)
C
      DY(3) = (-11.98*Y(4) + 2.156*SIN(THETA) - 2.*Z*1.66E-5*EXP(-A*T)
     1        *WN*Y(3))*1.66E05
C
      DY(4)=Y(3)
C
      RETURN
      END
```

Appendix B
Program TORQUE

```
C=====================================================================
C
C    PROGRAM        TORQUE
C
C    AUTHOR         RATNAKAR M. KANTH
C
C    DATE           SPRING 1987
C
C    DESCRIPTION    A personal computer based real-time
C                   angular velocity, acceleration and torque
C                   measuring system for rotating machinery
C
C    HARDWARE       A PC with color graphics,
C                   A sytem  disk with PCLAB drivers installed,
C                   A signals pre-processing card consisting of
C                   a phase locked loop circuit and
C                   an F/V converter with a low pass filter,
C                   A DATA TRANSLATION DT2801-A card configured with
C                   single ended, bipolar inputs
C
C    EXTERNAL       An optical shaft encoder with a 1 PPR & 600 PPR
C                   signals, mounted on the shaft of interest
C
C=====================================================================
C
$LARGE
$DEBUG
C
C    PCLAB subroutines definition
C
      INTEGER*2     STATUS,STARTC,ENDCHA,GAIN,TIMING,SAMPLE,ANS,STYPE
      INTEGER*2     XINIT,XTERM,XST,XSCD,ANALOG(1501),DATPTS,SPEED
      INTEGER*2     CHANLS,RANGE
      INTEGER*2     XSA,XBAD,XWAD,XSAD
      INTEGER*2     XRD,XESC,XDSC,PLOTS,PLTPNT,IERR
      EXTERNAL     XINIT,XTERM,XSAD,XESC,XDSC
C
      INTEGER*4     CLOCKD,KL,KH,ITORQ,IWK2(6000),IWK3(6000),IWK4(6000)
      CHARACTER*9   FLNAME
      REAL*4     X(501),X1(500),XV(501),XA(501),Y(501),Y1(500)
      REAL*4     YCH0(500),YCH1(500),YCH2(500),CLKPER
      REAL*4     YY(500),YV(501),YYY(500),YA(501),YT(501),WK2(6000)
      REAL*4     STV(500),CTV(500),STA(500),CTA(500),STVF(1),CTVF(1)
      REAL*4     YTSC(500),WK3(6000),WK4(6000),STAF(1),CTAF(1)
      COMPLEX    CWK2(1000),CWK3(1000),CWK4(1000)
C
      DATA     IFIRST/0/,IER/0/,IERR/0/
```

```
      DATA     XMN,XMX,YMN,YMX/4*0./
      DATA     CLKPER/1.25E-06/,PI/3.1415926/
      DATA     FLIP,FLOP/3.0,1.0/
C
C
 10   STATUS = 0
 20   WRITE (*,25)
 25   FORMAT (1X,'INPUT THE NUMBER OF DATA POINTS (25-250)')
      READ (*,*) KPTS
      IF (KPTS.LT.25 .OR. KPTS.GT.250) THEN
        WRITE (*,*)
        WRITE (*,*) '---------------------------------------'
       'WRITE (*,*) '   IMPROPER CHOICE OF DATA POINTS      '
        WRITE (*,*) '---------------------------------------'
        GO TO 20
      ENDIF
C
      KPTS = 2*KPTS
C
      WRITE (*,*) 'ENTER'
      WRITE (*,*) '     "1" FOR SPRING LOAD APPLICATION'
      WRITE (*,*) '     "2" FOR COMPRESSOR APPLICATION'
      READ (*,*) ITORQ
C
      WRITE (*,*) 'ENTER'
      WRITE (*,*) '     "1" FOR SAMPLING SPEED SPECIFICATION'
      WRITE (*,*) '     "2" FOR SAMPLING TIME SPECIFICATION'
      READ (*,*) STYPE
      IF (STYPE .EQ. 1) GOTO 30
      IF (STYPE .EQ. 2) GOTO 50
C
 30   WRITE (*,35)
 35   FORMAT(1X,'INPUT THE APPROX. SPEED (500-5000 RPM)')
      READ(*,*) SPEED
      IF (SPEED.LT.500. OR .SPEED.GT.5000) THEN
        WRITE (*,*)
        WRITE (*,*) '---------------------------------------'
       'WRITE (*,*) '   IMPROPER CHOICE OF SAMPLING SPEED   '
        WRITE (*,*) '---------------------------------------'
        GO TO 30
      ENDIF
      IF (SPEED .LE. 999) RANGE=1
      IF (SPEED.GE.1000 .AND. SPEED.LE.2399) RANGE=2
      IF (SPEED .GE. 2400) RANGE=3
      WRITE (*,*)
      WRITE (*,*) 'ENSURE THAT THE RANGE SELECTOR SWITCH IS AT'
      WRITE (*,45) RANGE
 45   FORMAT (1X,'POSITION ',I2)
      WRITE (*,*)
      TIME = 2.*60./SPEED
      GO TO 80
 50   WRITE (*,55)
 55   FORMAT(1X,'INPUT THE SAMPLING TIME (0.005-1.000 SEC)')
      READ(*,*) TIME
      IF (TIME.LT.0.005 .OR. TIME.GT.1.00) THEN
        WRITE (*,*)
        WRITE (*,*) '---------------------------------------'
       'WRITE (*,*) '   IMPROPER CHOICE OF SAMPLING TIME    '
        WRITE (*,*) '---------------------------------------'
        GO TO 50
      ENDIF
C
      TIME = 2.*TIME
C
      WRITE (*,*) 'ENTER THE RANGE OF OPERATION'
      WRITE (*,*) '      1 - FOR 500-1000 RPM'
      WRITE (*,*) '      2 - FOR 1000-2400 RPM'
```

```
      WRITE (*,*) '      3 - FOR 2000-4500 RPM'
      READ (*,*) RANGE
      WRITE (*,*)
      WRITE (*,*) 'ENSURE THAT THE RANGE SELECTOR SWITCH IS AT'
      WRITE (*,65) RANGE
 65   FORMAT ('POSITION ',I2)
C
C   Prompt the user for the type of triggering desired
C
C      WRITE (*,75)
C 75   FORMAT(1X,'EXT. TRIGGER/CLOCK(0,1,2,3)')
C   0 is software trigger, internal clock
C   1 is software trigger, external clock
C   2 is external trigger, internal clock
C   3 is external trigger, external clock
C      READ (*,*) TIMING
      TIMING = 0
C
C   Prompt the user for the channels on which sampling is desired
C
 80   WRITE (*,85)
 85   FORMAT (1X,'START CHANNEL ? (0-5)')
      READ (*,*) STARTC
      WRITE (*,95)
 95   FORMAT (1X,'END CHANNEL ? (0-5)')
      READ (*,*) ENDCHA
C
C   Determine user's choice of gain for plotting
C
      WRITE (*,105)
 105  FORMAT (1X,'GAIN ? (1,2,4,8) ')
      READ (*,*) GAIN
C
      CHANLS = (ENDCHA+1) - STARTC
      SAMPLE = CHANLS*KPTS
      ANALOG(1) = SAMPLE
C
C   Each clock tick = .00000125 sec., minimum clock ticks = 30
C   Find the number of clock ticks required for the user specified sampling
C
      CLOCKD = INT (TIME/(CLKPER*KPTS*CHANLS))
C
C   The following 'STATUS ='statements are functions that effect specified
C   operations on the interface card and if 'STATUS' is not zero an error
C   has occurred and the number delivered specifies the particular error
C   that can be used in error handling routines.
C
C   Disable PC's time of day clock so it won't interrupt
C
      STATUS = XDSC
C
C   Reset 2801-A card
C
      STATUS = XRD
C
C   Initialize card
C
      STATUS = XINIT
C
C   Set clock divider with number of ticks between samples
C
      STATUS = XSCD (CLOCKD)
C
C   Set respective parameters on card
C
      STATUS = XSA (TIMING, STARTC, ENDCHA, GAIN)
C
```

```
C   Disable PC's time of day clock so it won't interrupt
C
 110  STATUS = XDSC
C
C   Take specified no. of samples and fill array beginning at
C   specified element
C
      STATUS = XBAD (SAMPLE, ANALOG(2))
C
C   Wait until specified element is filled before continuing
C
      STATUS = XWAD (ANALOG(SAMPLE+1))
C
C   Terminate DMA operation with 2801-A card
C
      STATUS = XTERM
C
C   Enable PC's time of day clock
C
      STATUS = XESC()
C
C   Initialize counts for flip-flop
C
      MID = INT (KPTS/2)
      J1 = 3
      K = 0
      KFLAG = 1
C
      DO 130 I=1,KPTS
C
C   Scale analog from binary offset to volts and put in y array
C
        YCH1(I) = (FLOAT (ANALOG(J1)-2048)/204.8)/FLOAT(GAIN)
C
C   Retain data for only 1 cycle by checking with flip-flop
C
        IF (KFLAG .EQ. 1) THEN
          IF (YCH1(I) .LT. FLOP) THEN
            K = K + 1
           ELSE
            IF (K .GE. MID) GOTO 140
            K = 1
            KFLAG = 2
            GO TO 120
          ENDIF
        ENDIF
        IF (KFLAG .EQ. 2) THEN
          IF (YCH1(I) .GT. FLIP) THEN
            K = K + 1
          ENDIF
        ENDIF
 120    J1 = J1 + 3
 130    CONTINUE
C
 140  KBEG = I - K + 1
      J0 = KBEG - 1
      J2 = KBEG + 1
      DATPTS = K
C
      DO 150 I=1,DATPTS
        YCH0(I) = (FLOAT (ANALOG(J0)-2048)/204.8)/FLOAT (GAIN)
        YCH2(I) = (FLOAT (ANALOG(J2)-2048)/204.8)/FLOAT (GAIN)
        J0 = J0 + 3
        J2 = J2 + 3
 150  CONTINUE
C
```

```
C   Scale the ordinate to velocity - with the calibration factor
C
      DO 160 I = 1,DATPTS
        IF (RANGE .EQ. 1) Y(I+1) = 2.*PI*26.3*YCH0(I)
        IF (RANGE .EQ. 2) Y(I+1) = 2.*PI*39.*YCH0(I)
        IF (RANGE .EQ. 3) Y(I+1) = 2.*PI*100.*YCH0(I)
 160  CONTINUE
C
      Y(1) = REAL (DATPTS)
C
C   Initialize time sum to zero
C
      TIMSUM = 0
C
C   Find the time increment between sampling - for x axis
C
      DELTAT = FLOAT (CLOCKD)*.00000125*FLOAT(CHANLS)
C
      DO 170 I=1,DATPTS
C
C   Put accumulated time sum in x-array
C
        X(I+1) = TIMSUM
C
C   Increment time element
C
        TIMSUM = TIMSUM + DELTAT
 170  CONTINUE
C
      X(1) = REAL (DATPTS)
C
C   Initialize routines for plotting raw velocity vs. time
C   where velocity is in rad/sec and time is in sec
C
      CALL INITT (1)
      CALL BINITT
      CALL STROUT (250,700,'ANGULAR VELOCITY-RAD/SEC VS. TIME-SEC',37)
      CALL CHECK (X,Y)
      CALL DSPLAY (X,Y)
      CALL TCSBSV ('VEL.PIC ',IERR)
      IF (IERR.NE.0) WRITE (*,*) 'ERROR IN VEL.PIC'
C
C   Re-initialize arrays for use in the FFT routine
C
      DO 180 J=1,DATPTS
        X1(J) = X(J+1)
        Y1(J) = Y(J+1)
 180  CONTINUE
C
C   Call the routine FFT to analyse the raw velocity signal into
C   its Sine and Cosine components
C
      CALL  FFT (DATPTS,X1,Y1,IN,DELTAX,STV,CTV)
C
C   Call the routine which gives a spectrum of the first 20 harmonics
C   in the raw velocity signal
C
      CALL  SPCTRA (IN,STV,CTV)
      CALL TCSBSV ('VSPEC.PIC ',IERR)
      IF (IERR.NE.0) WRITE(*,*) 'ERROR IN VSPEC.PIC'

C
C   Prompt the user for the range of harmonic nos. to filtered
C
      WRITE (*,*) ' ENTER THE LOWEST HARMONIC TO BE FILTERED'
      READ (*,*) KL
      WRITE (*,*) ' ENTER THE HIGHEST HARMONIC TO BE FILTERED'
```

```
      READ (*,*) KH
C
C   Call the routine FFTINV to filter the noisy velocity signal
C   and rebuild the "filtered" signal
C

      CALL  FFTINV (IN,KL,KH,STV,CTV,YY,IER)
      IF (IER .NE. 0) WRITE(185,*)
 185  FORMAT (' ERROR IN SUBR. FFTINV FOR VEL.')
C
C   Call the IMSL routine to analyse the filtered velocity signal into
C   its Sine and Cosine components
C
      CALL  FFTSC (YY,IN,STVF,CTVF,IWK2,WK2,CWK2)
C
C   Call the routine which gives a spectrum of the first 20 harmonics
C   in the filtered velocity signal
C
      CALL  SPCTRA (IN,STVF,CTVF)
C
      XV(1) = FLOAT (IN)
      YV(1) = FLOAT (IN)
      XV(2) = 0.
      DO 190 J=1,IN
        IF(J.NE.IN) XV(J+2) = XV(J+1) + DELTAX
        YV(J+1) = YY(J)
 190  CONTINUE
C
C   Initialize routines for plotting velocity vs. time
C
      CALL NEWPAG
      CALL BINITT
      CALL STROUT(190,700,'ANGULAR VELOCITY(FILTERED)-RAD/SEC VS. TIME-S
     &EC',47)
      CALL CHECK (XV,YV)
      IF (IFIRST .EQ. 0) THEN
        CALL SEEDW (XMN,XMX,YMN,YMX)
        IFIRST = 1
      ENDIF
      CALL DSPLAY (XV,YV)
      CALL TINPUT (II)
C
C   If response is "Z" then Zoom
C
      IF (II.EQ.90.OR.II.EQ.122) THEN
        CALL  ZOOM (II,XV,YV)
      ENDIF
C
C   Differentiate the components of the velocity signal in the
C   frequency domain
C
      DO 200 I=1,IN
        STA(I) = -I*CTVF(I)
        CTA(I) = I*STVF(I)
 200  CONTINUE
C
C   Call the routine which gives a spectrum of the first 20 harmonics
C   in the acceleration signal
C
      CALL  SPCTRA (IN,STA,CTA)
      CALL TCSBSV ('UASPEC.PIC ',IERR)
      IF (IERR.NE.0) WRITE(*,*) 'ERROR IN UASPEC.PIC'
C
C   Build the acceleration signal from the sine and cosine components
C
C   Prompt the user for the range of harmonic nos. to filtered
C
```

```
      WRITE (*,*) ' ENTER THE LOWEST HARMONIC TO BE FILTERED'
      READ (*,*) KL
      WRITE (*,*) ' ENTER THE HIGHEST HARMONIC TO BE FILTERED'
      READ (*,*) KH

      CALL  FFTINV (IN,KL,KH,STA,CTA,YYY,IER)
      IF (IER .NE. 0) WRITE(205,*)
 205  FORMAT (' ERROR IN SUBR. FFTINV FOR ACCEL.')
C
C   Call the IMSL routine to analyse the acceleration signal into
C   its Sine and Cosine components
C
      CALL  FFTSC (YYY,IN,STAF,CTAF,IWK3,WK3,CWK3)
C
C   Call the routine which gives a spectrum of the first 20 harmonics
C   in the filtered acceleration signal
C
      CALL  SPCTRA (IN,STAF,CTAF)
      CALL TCSBSV ('FASPEC.PIC ',IERR)
      WRITE (*,*)
C
C

      XA(1) = FLOAT (IN)
      YA(1) = FLOAT (IN)
      DO 210 J=1,IN
        XA(J+1) = XV(J+1)
        YA(J+1) = YYY(J)
 210  CONTINUE
C
C   Initialize routines for plotting acceleration vs. time
C
      CALL NEWPAG
      CALL BINITT
      CALL STROUT(150,700,'ANGULAR ACCELERATION(FILTERED)-RAD/SEC*2 VS.
     &TIME-SEC',53)
      CALL CHECK (XA,YA)
      IF (IFIRST .EQ. 0) THEN
        CALL SEEDW (XMN,XMX,YMN,YMX)
        IFIRST = 1
      ENDIF
      CALL DSPLAY (XA,YA)
      CALL TINPUT (II)
C
C   If response is "Z" then Zoom
C
      IF (II.EQ.90.OR.II.EQ.122) THEN
        CALL  ZOOM (II,XA,YA)
      ENDIF
C
C   Scale the ordinate from accel. to torque - with the "J" factor
C
      DO 220 I=2,IN-2
        IF (ITORQ .EQ. 1) YT(I) = 2.882E-04*YA(I)
        IF (ITORQ .EQ. 2) YT(I) = 0.262*YA(I)
 220  CONTINUE
C
      DO 230 J=1,IN
        YTSC(J) = YT(J+1)
 230  CONTINUE
C
C   Call the IMSL routine to analyse the torque signal into
C   its Sine and Cosine components
C
      CALL  FFTSC (YTSC,IN,STT,CTT,IWK4,WK4,CWK4)
C
C   Call the routine which gives a spectrum of the first 20 harmonics
```

```
C   in the torque signal
C
      CALL  SPCTRA (IN,STT,CTT)
      CALL TCSBSV ('TSPEC.PIC ',IERR)
      IF (IERR.NE.0) WRITE (*,*) 'ERROR IN TSPEC.PIC'
C
C   Set number of plot array elements
C
      YT(1) = FLOAT (IN)
C
C   Initialize routines for plotting torque vs. time
C
      CALL NEWPAG
      CALL BINITT
      CALL STROUT (280,700,'DYNAMIC TORQUE-LBF.IN VS. TIME-SEC',34)
      CALL CHECK (XA,YT)
      IF (IFIRST .EQ. 0) THEN
        CALL SEEDW (XMN,XMX,YMN,YMX)
        IFIRST = 1
      ENDIF
      CALL DSPLAY (XA,YT)
      CALL TINPUT (II)
C
C   If response is "Z" then Zoom
C
      IF (II.EQ.90 .OR. II.EQ.122) THEN
        CALL  ZOOM (II,XA,YT)
      ENDIF

      CALL TCSBSV ('TORQUE.PIC ',IERR)
      IF (IERR .NE. 0) WRITE (*,*) 'ERROR IN TORQUE.PIC'
C
C   Prompt the user 'what next ?'
C
 240  CALL ANMODE
      WRITE (*,245)
 245  FORMAT (//,' ENTER THE APPROPRIATE NUMBER:',/,
     1    ' 1  -  DO AGAIN',/,
     2    ' 2  -  CHANGE PARAMETERS',/,
     3    ' 3  -  SAVE TORQUE ARRAY',/,
     4    ' 4  -  END  >> (4) ',$)
      READ (*,255,ERR=240) ANS
 255  FORMAT (I1)
      IF (ANS .EQ. 0) ANS=4
      IF (ANS.LT.1 .OR. ANS.GT.4) GOTO 240
C
C   Transfer program control
C
      IF (ANS.EQ.1) GOTO 110
      IF (ANS.EQ.2) GOTO 10
      IF (ANS.EQ.3) GOTO 260
      IF (ANS.EQ.4) GOTO 280
C
 260  CALL NEWPAG
      WRITE (*,*)
      WRITE (*,*) 'GIVE THE NAME OF THE FILE (<9 CHARACTERS)'
      READ (*,'(A)') FNAME
      OPEN (9,FILE=FLNAME,STATUS='NEW',FORM='FORMATTED')
      WRITE (9,*) IN
      DO 270 I = 1,IN
        WRITE (9,*) XV(I)
        WRITE (9,*) YV(I)
 270  CONTINUE
      CLOSE (9)
      GO TO 240
C
```

```
 280  CALL FINITT (0,0)
C
C
      STOP
      END
C
C
C=====================================================================

      SUBROUTINE  FFT (N2,XS,YS,IN,DELX,ST,CT)

C   This is a routine which uses Fast Fourier Transform algorithm
C   to analyse the i/p waveform which is in the time domain into
C   its Sine & Cosine components in the frequency domain

C=====================================================================

      INTEGER*2  N2
      INTEGER    IWK(6000)
      REAL       A1(1001),XS(1),YS(1),WK(6000),ST(1),CT(1)
      REAL       INTERP
      COMPLEX    CWK(1000)

      NPTS = N2
C
C   Prompt the user for # points that are to result from interpolation
C
      WRITE(*,*)'SPECIFY # POINTS THAT ARE TO RESULT FROM INTERPOLATION'
      READ (*,*) NDES
      NNN = (NDES/2)*2
      IF (NNN .EQ. NDES) NDES=NDES+1
      DELX = (XS(NPTS)-XS(1))/FLOAT(NDES-1)
      DO 10 I=1,NDES
        IF (I .EQ. 1) TMP=XS(1)
        IF (I .NE. 1) TMP=TMP+DELX
        A1(I) = INTERP (TMP,XS,YS,NDES,IER)
        IF (IER .NE. 0) WRITE(*,5) I,TMP
 5      FORMAT (' ERROR IN INTERPOLATING ',I5,'TH POINT'/,
     &          ' CHECK FOR OUT-OF-RANGE VALUE, 'F15.8)
 10   CONTINUE
      IN = NDES - 1
C
C   Call the IMSL routine for Sine & Cosine Transforms
C
      CALL  FFTSC (A1,IN,ST,CT,IWK,WK,CWK)


      RETURN
      END


C=====================================================================

      REAL FUNCTION INTERP (Z,X,Y,N,IER)

C   This is a function which interpolates "N" no. of pts. "Z",
C   given the "X" and "Y" arrays

C=====================================================================

      DIMENSION   X(1),Y(1)

      DO 10 I=1,N
        IF (Z.GE.X(I) .AND. Z.LT.X(I+1)) GOTO 20
 10   CONTINUE
C
```

```
C   Set an error code for out-of-range
C
      IER = 9
      INTERP = Y(N)
      RETURN

 20   IER = 0
      INTERP = Y(I) + (Y(I+1)-Y(I))/(X(I+1)-X(I))*(Z-X(I))

      RETURN
      END


C=====================================================================

      SUBROUTINE  SPCTRA (IN,ST,CT)

C   This is a routine which gives the spectrum of
C   the first 20 harmonics of an input waveform

C=====================================================================

      REAL    ST(1),CT(1),FREQ(21),YMAG(21)


      FREQ(1) = 20.
      YMAG(1) = 20.
      DO 10 I=2,21
        YMAG(I) = (SQRT (ST(I)**2+CT(I)**2))/FLOAT(IN)
        FREQ(I) = I - 1
 10   CONTINUE

      CALL  NEWPAG
      CALL  BINITT
      CALL  XFRM (2)
      CALL  VBARST (4,0,0)
      CALL  DLIMX (0.,10.1)
      CALL  ANMODE
      CALL  CHECK (FREQ,YMAG)
      CALL  DSPLAY (FREQ,YMAG)
      CALL  FRAME
      CALL  MOVABS (0,0)
      CALL  ANMODE
      CALL  TINPUT (IOP)
      CALL  ANMODE

      RETURN
      END


C=====================================================================

      SUBROUTINE  FFTINV (N,KL,KH,ST,CT,B,IER)

C   This routine uses the inverse FFT algorithm to
C   build a waveform from Sine and Cosine components

C=====================================================================

      DIMENSION   B(1),ST(1),CT(1)
      DATA   PI/3.141593/

      IER = 0
      NN = N/2 + 1
      NEND = N/2 - 1

      DO 100 JJ=1,N
```

```
        J = JJ - 1
        ISIGN = 1
C
C   IF J  IS EVEN THEN ISIGN = 1
C   IF J  IS ODD  THEN ISIGN = -1
C
        JTST1 = (J+1)/2
        JTST2 = J/2
        IF (JTST1 .EQ. JTST2) ISIGN=-1
        SUM = CT(1)/(2.*N) + CT(NN)/(2.*N)*ISIGN
        DO 10 K=1,NEND
          IF (K.GE.KL .AND. K.LE.KH) GOTO 10
C
C   Filter harmonics
C
          SUM = SUM + CT(K+1)/N*COS((2.*PI*J*K)/N)
     &              + ST(K+1)/N*SIN((2.*PI*J*K)/N)
 10       CONTINUE
          B(JJ+1) = SUM
 100  CONTINUE

      RETURN
      END


C=====================================================================

       SUBROUTINE  ZOOM (II,X,Y)

C   This is a routine which magnifies the given array of x and y
C   and displays it as also saving and re-displaying the original array

C=====================================================================


      REAL    X(1),Y(1)
      DATA    XMN,XMX,YMN,YMX/4*0./

      GO TO 15

 10   CALL CHECK (X,Y)
      IF (IFIRST.EQ.0) THEN
        CALL SEEDW (XMN,XMX,YMN,YMX)
        IFIRST = 1
      ENDIF
      CALL DSPLAY (X,Y)
      CALL TINPUT (II)
C
C   If response is "Z" then Zoom
C
 15   IF (II.EQ.90 .OR. II.EQ.122) THEN
C
C   Put up prompts for corners, determine and set new windows
C
        CALL  STROUT (0,30,'SELECT NEW CORNER, PRESS SPACE BAR ',42)
        CALL  VCURSR (ICHAR,X1,Y1)
        CALL  MOVEA (X1,Y1)
        CALL  MOVREL (-10,0)
        CALL  DRWREL (20,0)
        CALL  MOVREL (-10,-10)
        CALL  DRWREL (0,20)
        CALL  STROUT (0,0,'SELECT OPPOSITE CORNER, PRESS SPACE BAR',40)
        CALL  VCURSR (ICHAR,X2,Y2)
        XMIN = X1
        IF (X2.LT.X1) XMIN = X2
        YMIN = Y1
        IF (Y2.LT.Y1) YMIN = Y2
```

```
          DELX = ABS (X2-X1)
          DELY = ABS (Y2-Y1)
          XAVG = (X1+X2)/2.
          YAVG = (Y1+Y2)/2.
          IF (DELX.EQ.0) DELX = 0.001
          IF (DELY.EQ.0) DELY = 0.001
          XMIN = XAVG - DELX/2.
          YMIN = YAVG - DELY/2.
          XMAX = XAVG + DELX/2.
          YMAX = YAVG + DELY/2.
          CALL  BINITT
          CALL  COMSET (IBASEX(11),XMIN)
          CALL  COMSET (IBASEY(11),YMIN)
          CALL  COMSET (IBASEX(12),XMAX)
          CALL  COMSET (IBASEY(12),YMAX)
          CALL  ERASE
          GO TO 10
        ENDIF
C
C   Return to original window by resetting and redrawing
C   with "0" for Over
C
        IF (II.EQ.79.OR.II.EQ.111) THEN
          CALL  DLIMX (XMN,XMX)
          CALL  DLIMY (YMN,YMX)
          CALL  DWINDO (XMN,XMX,YMN,YMX)
          CALL  BINITT
          CALL  ERASE
          GO TO 10
        ENDIF
C
        RETURN
        END
C
C
C=====================================================================

        SUBROUTINE  STROUT (IX,IY,KSTR,NCHR)

C   This is a routine which displays specified character string
C   at specified location on the console

C=====================================================================


        DIMENSION  KSTR(1),KAS(80)

        CALL  KAM2AS (NCHR,KSTR,KAS)
        CALL  ANSTR (NCHR,KAS)

        RETURN
        END
C
C
C=====================================================================
C
        SUBROUTINE  FFTSC (A,N,ST,CT,IWK,WK,CWK)
C
C   Computes the sine and cosine transforms of a
C   real valued sequence
C   Required IMSL routines: FFTSC, FFTRC, FFT2C
C
C   This is an International Mathematical and Statistical Library
C   routine copyrighted by IMSL, Inc., Houston, Texas.
C
C=====================================================================
C
```

```
C
C=====================================================================
C
      SUBROUTINE  FFTCC (A,N,IWK,WK)
C
C   Computes the fast fourier transform of a
C   complex valued sequence
C   Required IMSL routines: None
C
C   This is an International Mathematical and Statistical Library
C   routine copyrighted by IMSL, Inc., Houston, Texas.
C
C=====================================================================
C
C
C=====================================================================
C
      SUBROUTINE  FFTRC (A,N,X,IWK,WK)
C
C   Computes the fast fourier transform of a
C   real valued sequence
C   Required IMSL routines: None
C
C   This is an International Mathematical and Statistical Library
C   routine copyrighted by IMSL, Inc., Houston, Texas.
C
C=====================================================================
C
C
C=====================================================================
C
      SUBROUTINE  FFT2C (A,N,IWK)
C
C   Computes the fast fourier transform of a
C   complex valued sequence of length equal to a power two
C   Required IMSL routines: None
C
C   This is an International Mathematical and Statistical Library
C   routine copyrighted by IMSL, Inc., Houston, Texas.
C
C=====================================================================
```

The vita has been removed from the scanned document