

To the Graduate Council:

I am submitting herewith a thesis written by Miodrag Glumac entitled "Automatic Diagnosis of Motor-Operated Valves using Motor Power Signature Analysis." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

Belle R Upadhyaya

Belle R. Upadhyaya, Major Professor

We have read this thesis
and recommend its acceptance:

Laurence F Miller

Robert E Whiting

Accepted for the Council:

Cowminkel

Associate Vice Chancellor
and Dean of The Graduate School

**AUTOMATED DIAGNOSIS OF MOTOR-OPERATED VALVES
USING MOTOR POWER SIGNATURE ANALYSIS**

A Thesis

Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Miodrag Glumac

May 1996

ACKNOWLEDGMENTS

I would like express my sincere gratitude to my academic advisor, Dr. B. R. Upadhyaya. His guidance, technical advice, and encouragement throughout the entire period of this research have been invaluable. I would also like to thank the other committee members, Dr. L. F. Miller and Dr. R. E. Uhrig for their assistance. I greatly appreciate the help and advice provided by Dr. T. W. Kerlin, Head of the Nuclear Engineering Department. This research was made possible by a research grant from Duke Power Company, Charlotte, NC. I appreciate the assistance provided by B. W. Roberts and M. L. Cline of Duke Power Company. Thanks also go to my colleagues in the Department of Nuclear Engineering: Wu Yan, Ali Erbay, Kadir Kavagioglu for many insightful discussions. I would like to thank a friend of mine, Anthony J. Caldanaro, for reading several manuscripts. The help of several of the support staff, Richard Bailey, Gary Graves, Lynnetta Holbrook and Kathy Phillips, is also very much appreciated. I would especially like to thank my parents, Nikola and Aurelia Glumac, and my sister Bosiljka, whose constant support was essential in completing this thesis.

ABSTRACT

The development of nonintrusive diagnostics methods makes it possible to identify malfunctions in plant components during normal plant operation and to avoid catastrophic failures and associated costs. If a plant component is driven by an electric motor, then the measurement of electrical variables, generally provides enough information to establish the operability of the component. Electrical measurements have been used by Duke Power Company and Oak Ridge National Laboratories, among others.

The purpose of this research project is to develop an automated diagnostics system to identify and isolate operational malfunctions in motor-operated valves (MOVs). The traditional approach uses motor current signature analysis for valve monitoring. In this research project the motor power signature analysis (MPSA) was developed as an alternative and a more sensitive technique for MOV diagnosis. A PC-based expert system was developed for integrating MOV motor power pattern analysis, a rule-based expert system for anomaly detection and classification, a fuzzy logic decision module, a data interface between the expert system and an existing data acquisition system, and a user interface with on-line help.

The diagnostic system was then tested using data from a power utility and laboratory valve test data acquired at The University of Tennessee. The system was demonstrated for on-line MOV anomaly diagnosis. The present system has proved to be very effective in detecting MOV malfunctions with minimum user input.

LIST OF ACRONYMS

- ASME: American Society of Mechanical Engineers
- DDE: Dynamic Data Exchange
- MCSA: Motor Current Signature Analysis
- MFC: Microsoft Foundation Classes library
- MOV: Motor-Operated Valve
- MPSA: Motor Power Signature Analysis
- NRC: Nuclear Regulatory Commission
- RMS: Root Mean Square
- SQL: Structural Query Language

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 Background and Scope of Research	1
1.2 Summary of Approach and General Features	4
1.3 Significant Contributions of the Thesis	5
1.4 Organization of the Thesis	6
2. OPERATION OF MOTOR OPERATED VALVES.....	7
2.1 Applications of Motor Operated valves (MOVs)	7
2.2 Valve Types	7
2.3 MOV Operation	11
2.4 Anomalies in MOVs	16
3. CURRENT METHODS FOR MOV TESTING AND MONITORING	18
3.1 Review of Prior Work.....	18
3.2 Requirements for Testing of MOVs in Nuclear Power Plants	19
4. MOTOR POWER SIGNATURE.....	20
4.1 Introduction.....	20
4.2 Electrical Power	20
5. DEVELOPMENT OF AN AUTOMATED DIAGNOSTICS SYSTEM FOR MOTOR-OPERATED VALVES	27
5.1 Fuzzy Linguistic Description	27
5.2 Fuzzy Logic Inference.....	28
5.3 System Structure	32
5.3.1 Data Preprocessor Module.....	32
5.3.1.1 Decomposition Algorithm	34
5.3.1.2 Peak Detector	37

5.3.1.3 Waveforms Peaks	38
5.3.1.4 Peak detection algorithm	38
5.3.1.5 Detection of Steps and Ramps	39
5.3.2 Feature Extraction	42
5.3.3 Diagnostics Module - Expert System	44
5.3.3.1 Knowledge Base	44
5.3.3.2 Inference Engine	46
5.3.4 System Implementation	58
5.3.4.1 C++ Classes Created for Expert System	59
6. APPLICATIONS TO MOTOR-OPERATED VALVE DIAGNOSIS	65
6.1 Measurements	65
6.2 Valve Test Data	66
6.3 Example of Program Execution	111
7. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH 134	
7.1 Conclusions	134
7.2 Recommendations for Future Research	135
7.2.1 Program Improvement	135
7.2.2 Database Support	136
REFERENCES	138
VITA	141

TABLE OF FIGURES

Figure 1.1 Motor power signatures during strokes of an MOV.....	3
Figure 2.1 Functional diagram of an MOV.	12
Figure 2.2 Limitorque SMB000 valve.....	15
Figure 4.1. Per-phase equivalent circuit referred to the stator.	23
Figure 4.2. Vector diagram of motor currents.	24
Figure 4.3. Vector diagram of motor power.	26
Figure 5.1 Membership functions.	30
Figure 5.2 An Example of fuzzification.	30
Figure 5.3 Max-product fuzzy composition.	31
Figure 5.4 MOV diagnostics system structure.....	33
Figure 5.5 Description of signal primitives in the program.....	37
Figure 5.6 Schematic diagram of the pattern recognition algorithm.	40
Figure 5.7 Characteristic events in the open-to-close valve stroke.	43
Figure 5.8 Characteristic events in close-to-open valve stroke.	43
Figure 5.9 Schematic diagram of the expert system software (modified from J. Martin and S. Oxman, 1988)	45
Figure 5.10 The data structure of fuzzy variables.....	47
Figure 5.11 Data structure for input variable.....	49
Figure 5.12 Fuzzification.	49
Figure 5.13 Rules 1 and 2.	52
Figure 5.14 Rule 3.	52
Figure 5.15 Output membership function.....	55
Figure 5.16 Defuzzification.	56
Figure 5.17 Rule base data structure.....	57
Figure 6.1 Schematic diagram of the measurement system.....	67
Figure 6.2 Limitorque 1-inch globe valve: baseline - open-to-close stroke.	70
Figure 6.3 Limitorque 1-inch globe valve: baseline - close-to-open stroke.	71

Figure 6.31	Limitorque 2-inch gate valve: normal stroke - CO.....	99
Figure 6.32	Limitorque 2-inch gate valve: bad pinion gear.....	100
Figure 6.33	Limitorque 2-inch gate valve: misadjusted limit switch - CO.	101
Figure 6.34	Limitorque 2-inch gate valve: : misadjusted limit switch - OC.	102
Figure 6.35	Limitorque 2-inch gate valve: broken rotor bar - CO.....	103
Figure 6.36	Limitorque 2-inch gate valve: broken rotor bar - OC.....	104
Figure 6.37	Kerotest 2-inch valve: pressure in the system = 0 psig - CO and OC.	105
Figure 6.38	Kerotest 2-inch valve: pressure in the system = 500 psig - CO and OC. ...	106
Figure 6.39	Kerotest 2-inch valve: pressure in the system = 1000 psig - CO and OC. .	107
Figure 6.40	Kerotest 2-inch valve: pressure in the system = 1500 psig - CO and OC. .	108
Figure 6.41	Kerotest 2-inch valve: pressure in the system = 2000 psig - CO and OC. .	109
Figure 6.42	Kerotest 2-inch valve: pressure in the system = 2500 psig - CO and OC. .	110
Figure 6.43	The desktop with program and help icons.....	112
Figure 6.44	The main program window immediately after program is started.	113
Figure 6.45	The preprocessor dialog box.	114
Figure 6.46	The open file dialog box.....	115
Figure 6.47	The motor power signature.....	116
Figure 6.48	The selected area of the signature.....	118
Figure 6.49	The save dialog box.....	119
Figure 6.50	Save file dialog box.....	120
Figure 6.51	PMV file dialog box.....	121
Figure 6.52	The form windows with command buttons is created after file was open. .	122
Figure 6.53	The primitives are found and displayed in the edit box.	123
Figure 6.54	Events describing the stroke are displayed in the dialog box	124
Figure 6.55	The features used in diagnosis.....	125
Figure 6.56	Diagnosis window.....	126
Figure 6.57	Rule editor.....	128
Figure 6.58	Output variable.....	132
Figure 6.59	Report.....	133

LIST OF TABLES

Table 2.1	Five valve types	9
Table 5.1	Membership functions	51
Table 5.2	Rule evaluation	54
Table 5.3	CPrimitives member functions	61
Table 5.4	CData member function.....	62
Table 5.5	CEvList member function.....	62
Table 5.6	CDFList member functions	63
Table 5.7	CinferEngine member function	64
Table 6.1	Data obtained at The University of Tennessee	68
Table 6.2	Data from a power company.....	69
Table 6.3	Rules used for MOV diagnosis.....	129

CHAPTER 1

INTRODUCTION

1.1 Background and Scope of Research

The issue of aging nuclear power plant components is being studied extensively as the present generation of power plants approaches its design life. Monitoring, maintenance and reliability of plant equipment are important operational issues and will also contribute to the study of plant life extension. As the need for improved testing and diagnostics increases, we are frequently faced with the problem of managing and processing large and complex databases, and automating diagnostics procedures.

The purpose of the current research is to develop a PC-based system for the automated diagnosis of Motor-Operated Valves (MOVs) using motor power signatures analysis. The technique uses the motor power waveforms generated during close-to-open and open-to-close strokes of standard MOVs (such as the Limitorque). The methodology integrates syntactic pattern recognition for waveform characterization, an expert system for relating power signatures to actuator anomalies, a fuzzy logic decision module and a graphical user interface with hypertext features.

The automated diagnosis system was developed for implementation on a PC platform. An appropriate interface was also developed to acquire motor power data from a power utility data acquisition system. The user interface, graphical display and other

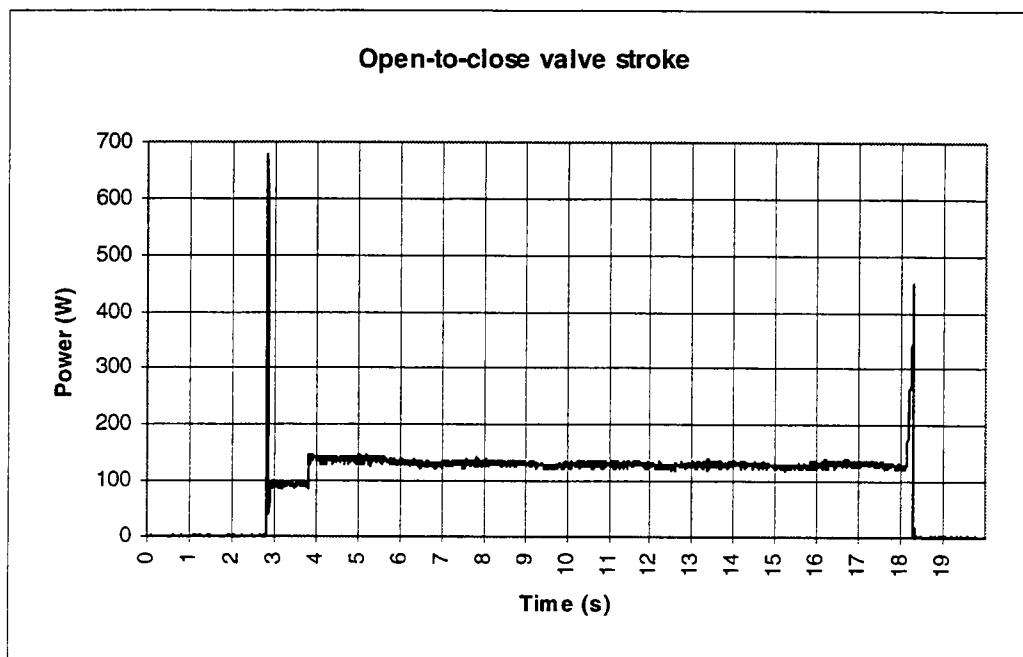
user-friendly features were developed in consultation with the sponsor. The PC-based system was interfaced with the existing data acquisition system.

The following demonstrations were presented as a part of the research tasks.

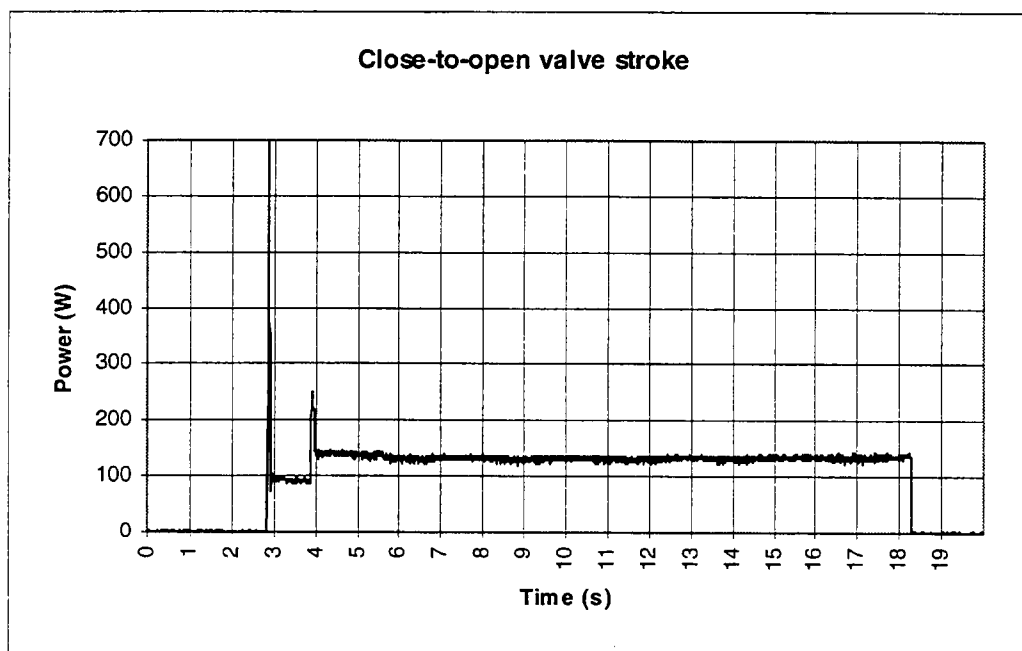
1. General functions of the software system with information display.
2. Demonstration of the system for the detection of known anomalies in the motor operator.
3. Demonstration of on-line application with data interface.

The diagnosis of anomalies during MOV operation requires the monitoring of about twenty-five features of normal valve operation. At least twenty types of faults need to be identified and characterized for proper diagnosis during operation. This imposes a heavy burden on operators and relies on human experts to perform fault detection. The reliability and speed of fault detection could be enhanced by automating the diagnostics procedure. Furthermore, the cause of detected anomalies and suggestions for troubleshooting could be provided to plant personnel. New valve databases may be installed and updated without the need to develop new software. Laboratory testing of MOVs with known anomalies was performed at The University of Tennessee, and additional test data were acquired from a power utility.

The primary focus of this research is to use the motor power signature from MOVs to characterize valve stroke operations. Figure 1.1 shows typical motor power signatures during open-to-close and close-to-open strokes. Due to the use of phase angle between motor current and voltage during power measurement, the motor power is



a. Open-to-close stroke.



b. Close-to-open stroke.

Figure 1.1 Motor power signatures during strokes of an MOV.

more closely related to the actual mechanical load experienced by the valve actuator than the motor current.

1.2 Summary of Approach and General Features

The overall purpose of the research project was to develop an automated diagnostics system to identify and isolate operational malfunctions in MOVs and to examine certain electrical parameters that change with degradation. About twenty different valve anomalies were investigated. Emphasis was given to the solution of specific problems of interest to power utilities. The following are the important features of the PC-based diagnosis system.

- Data preprocessor. The data include motor power, three-phase currents and voltages, and power factor.
- Feature extraction from the motor power signature.
- A rule-based system for anomaly detection.
- A fuzzy logic decision module for anomaly categorization.
- User interface and help file.
- Report generation.

The performance of the new system was evaluated in detail using data from a power utility and from laboratory testing at The University of Tennessee. The following tasks were identified and completed to accomplish the objectives of this research.

- Task 1. Review motor current signature analysis (MCSA) and motor power signature analysis (MPSA). Review of three-phase induction motors as applied to MOVs.
- Task 2. Establish the sensitivity of motor power signature for MOV diagnosis as compared with motor current signature.
- Task 3. Develop a PC-based fault diagnostics system using the features of motor power signature. Develop a fuzzy logic decision making module for anomaly identification.
- Task 4. Develop an interface between the analysis software and the data acquisition system currently used by a power company. Add features for on-line implementation.
- Task 5. Develop an MOV test facility at The University of Tennessee, using a nuclear-grade motor operator (Limitorque SMB000).
- Task 6. Establish the performance of the new system using both laboratory and plant data.
- Task 7. Demonstrate the capability of the analysis software for on-line diagnosis.

1.3 Significant Contributions of the Thesis

The MOV diagnostics software system was developed for use with The Microsoft Windows operating system using The Microsoft Foundation Classes library. The following is a summary of the significant accomplishment of this thesis research.

- Development of a module for the detection of characteristic events during open-to-close and close-to-open valve strokes of an MOV using syntactic pattern recognition of motor power signature.
- Development of a Dynamic Data Exchange (DDE) link for data transfer between the data acquisition system and the expert system.
- Development of a module for MOV fault detection using fuzzy logic.
- Development of a user interface that utilizes the Windows operating system.
- Performance testing of the PC-based system using measurements on a Limatorque SMB000 valve actuator at a power company laboratory. Different conditions were simulated using a 1-inch globe valve and a 3-inch gate valve.
- Demonstration of the diagnostics system for on-line MOV anomaly detection.

1.4 Organization of the Thesis

Some general information about valves and in particular MOVs, are given in Chapter 2. Chapter 3 contains a review of the various methods of MOV testing and monitoring. The application of motor power signature to MOV diagnosis is discussed in Chapter 4. Chapter 5 presents the description of the automated diagnosis system, developed in this research. Chapter 6 describes the application of the software system to MOV diagnosis. Both laboratory and plant data are presented. Conclusions and recommendations for future work are presented in Chapter 7.

CHAPTER 2

OPERATION OF MOTOR-OPERATED VALVES

2.1 Applications of Motor-Operated Valves (MOVs)

The purpose of a valve is to control the fluid flow in a system consisting of pipes, vessels, apparatus, and machines, by opening, closing, diverting, mixing, or partially obstructing the passage through itself. Valves are found in very large numbers in power plants where they are used to control the flow of water or steam. The operational readiness of such plants therefore strongly depend on the operational readiness of valves. Motor-operated valves (MOVs) are used extensively in nuclear power plant safety and control systems. MOVs are often employed when factors such as large line size, high pressure, temperature, flow rate, or inaccessible location dictate that other types of valve operators are inadequate. Applications of MOVs include: suction isolation valves for emergency core cooling system pumps, main feedwater system flow control valves, block valves for system train, pump, heat exchanger, or other equipment where isolation is desirable, block valves for pressurizer power operated relief valves, and others.

2.2 Valve Types

Valves are categorized by the design of their closure device and the direction of flow in the seating area [1]:

- Gate valve: A valve in which the closure device moves in a straight line, and in the seating area, across the direction of flow.
- Globe valve: A valve in which the closure device moves in a straight line, and in the seating area, parallel to the direction of flow.
- Plug and ball valves: Valves in which the closure device rotates, and in the open position, the flow passes around it.
- Diaphragm (pinch) valve: A valve in which the closure device is provided by the deformation of a flexible diaphragm or tube.

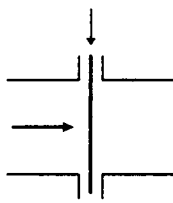
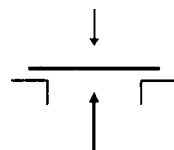
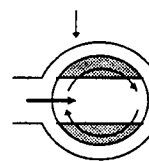
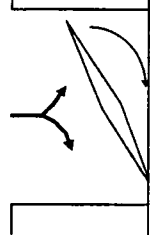
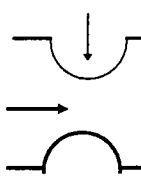
These five valve types are also shown in Table 2.1.

The valves may also be classified according to the power source needed to operate them. Therefore, the valves may be classified as follows:

- manually-operated valves,
- motor-operated valves (MOV),
- solenoid-operated valves,
- air-operated valves, and
- hydraulically-operated valves.

The expert system for this project was developed for use with MOVs. The use of an electric actuator with an MOV, allows for the collection of data representing valve operation, from electrical measurement. The use of electrical measurements are well established, accurate and are relatively cheap. The use of electric actuators offers several distinct advantages over other forms of actuation [1].

Table 2.1 Five valve types [1]

Operating motion of the closure device (obturator)	Straight line	Rotating about and axis at right angle to the direction of flow		Deformation of flexible diaphragm or tube	
Direction of flow in the seating area	At right angles to the operating motion of the obturator	Longitudinal to the operating motion of the obturator	Through the obturator	Around the obturator	Depends on design
Schematic					
Basic types	Gate valve	Globe valve	Plug valve (including ball valves)	Butterfly valve	Diaphragm valve

1. Motors have high rotational speed providing high kinetic energy to ensure unseating of valves.
2. Valve position stability is inherent due to the reduction gearing and/or nut/screw arrangement normally employed for transmitting the drive to the valve.
3. The ability to interface directly with modern computerized and telemetric remote control systems.
4. Electric circuits do not transmit dirt or moisture, and operate instantly over great distances.
5. Handwheel operation is simple to provide, together with a hand/power clutch mechanism having a power preference.
6. Interface between motor and remote control is a contact, which has fewer reliability problems than solenoid pilot valves.
7. The complete actuator and control system, apart from the final drive to the valve, can be tested without moving the valve position, by holding the handwheel engagement clutch lever in the "hand" position.
8. Where plant operation demands a particular sequence of valve operation, the necessary interlocks can be provided by the interconnection of limit switches on the appropriate apparatus.
9. Segregation of emergency and normal control circuits is relatively simple and enables fire protection considerations to be concentrated on the particular circuit that are essential for reliable emergency operation.

10. Usually only two cables, one for power and one for control connections, need to be brought into the actuator location. irrespective of the degree of sophistication of the control and instrumentation system.
11. Either complete electric operators or subassemblies are usually readily interchangeable, though they may have initially been supplied for completely different applications.

Despite the numerous advantages several disadvantages to the use of electrical actuators exist. These are [1]

1. Stay-put failure mode in the event of power failure.
2. Vulnerability of electrical equipment to high temperatures.
3. Electrical energy is difficult to store and the need for gearing between the motor and the valve usually precludes mechanical means of energy storage in the form of a spring for emergency one-shot operation.
4. Vulnerability of electrical equipment to moisture requires high integrity sealed enclosures.

2.3 MOV Operation

Valves analyzed in this project are normally in fully-open or fully-closed position. Therefore, two valve strokes were observed: open-to-close and close-to-open. During the strokes the power consumption of the actuator changes and the shape of the power signature may be used to infer MOV conditions. The functional diagram of an MOV is shown in Figure 2.1.

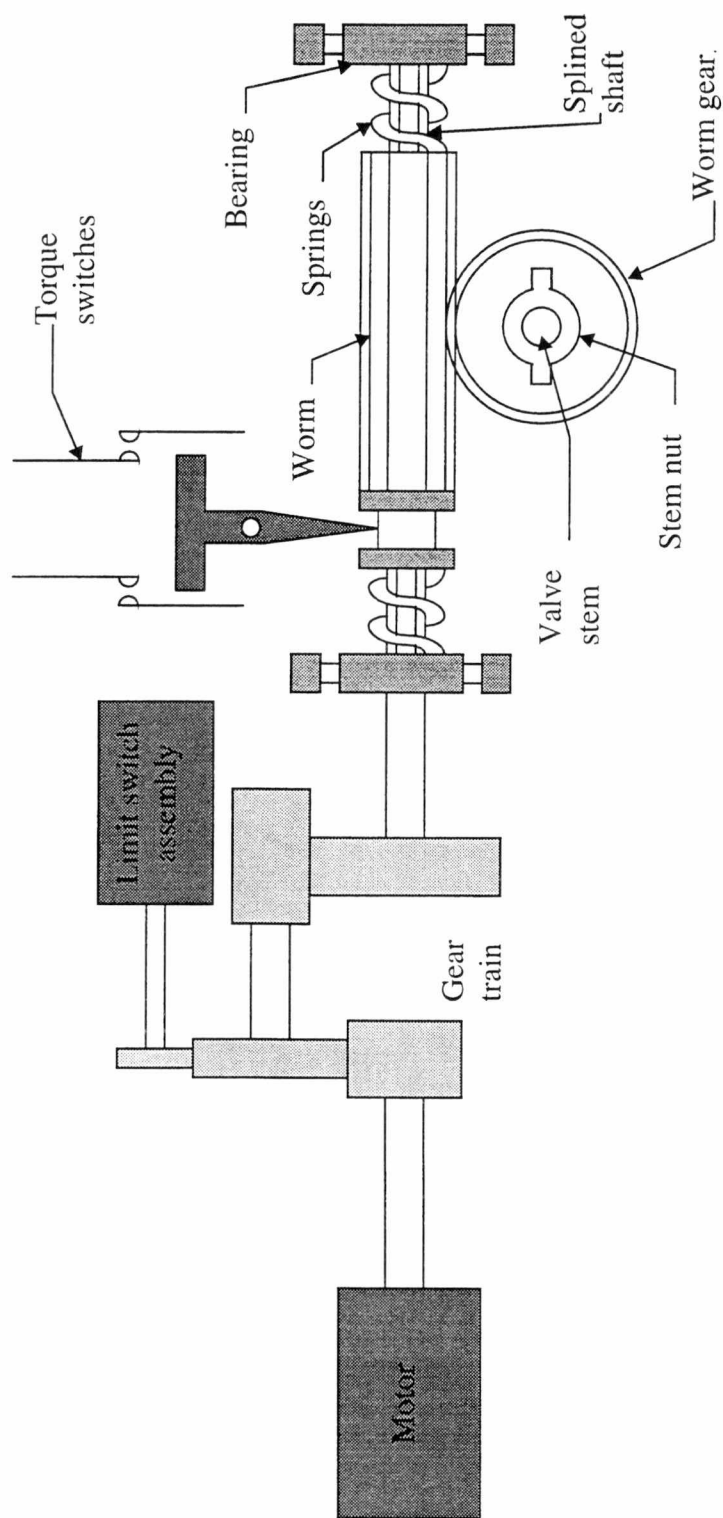


Figure 2.1 Functional diagram of an MOV.

The most common case of operation is that the valve closes by the seating of the obturator on the seat. This completely disconnects the fluid flow. The torque switch is used to sense the force between the stem and the seat, and stop the motor. The purpose of the torque switch is to allow enough force to close the valve, and at the same time, protect the valve from excessive mechanical forces that can damage the valve.

After the MOV is closed, the stem is locked in the closed position. When the MOV opens, the actuator has to unlock the stem. This causes a short increase in the mechanical load. Additionally, a force is needed to unseat the peak. The electric motor supplies enough power to ensure unseating.

The limit switches are used to indicate the valve stem position. These switches use the number of turns of a worm nut to indicate the stem position. These switches can also be used as part of the valve control logic. Usually, the limit switch, which indicates the open position, is also used to turn the electric motor off. This prevents the valve from backseating, and causing further damage.

The details of MOV operation depends on the design of the MOV. The following is a description of the operation of a Limitorque SMB valve taken from Ref [2]:

The electric motor has a helical pinion mounted on its shaft extension. This pinion, pc. #40, drives the worm shaft clutch gear, pc. #41, which is engaged with pc. #50, the worm shaft clutch. This piece is splined to the worm shaft, pc. #43. Piece #53, the worm shaft, is splined to the worm shaft, pc. #43, and when it is rotated it turns pc. #16; the worm gear. The worm gear has two lugs cast onto the top portion which

engages the two lugs on the drive sleeve, pc #11. These lugs are spaced so that when the worm gear begins to turn during motor operation there is a certain amount of lost motion before the lugs engage and cause the hammer blow effect within the operator.

As soon as the worm gear lugs engage, the drive sleeve being splined internally with pc. #20, the stem nut, causes the stem nut to rotate and open or close the threaded stem of the valve. The stem nut is threaded to fit the thread of any rising stem valve.

In the case of non-rising stem valves or where the electric operator is mounted in tandem with an additional gear drive, the stem nut, pc. #20, is merely bored and keyed to fit the shaft.

The thrust developed by a Limitorque valve control is absorbed by the heavy duty thrust bearings on the top and bottom of the main drive sleeve. As the Limitorque valve control develops a greater torque, when seating the valve, the worm slides axially along the splines of the worm shaft and compresses the Belleville springs, pc. #56, which are the torque springs. These are calibrated springs and every increment of compression for a given size unit a certain amount of torque is developed. The torque switch is mechanically actuated by the worm. When the worm moves back a preset distance and develops the required amount of output torque, the motor control circuit interrupts the circuit and stops the motor at this point.

The schematic diagram of an SMB000 valve is shown in Figure 2.2 [2]. Generally, the torque switch is wired into the motor control circuit to stop the operator in a full-closed

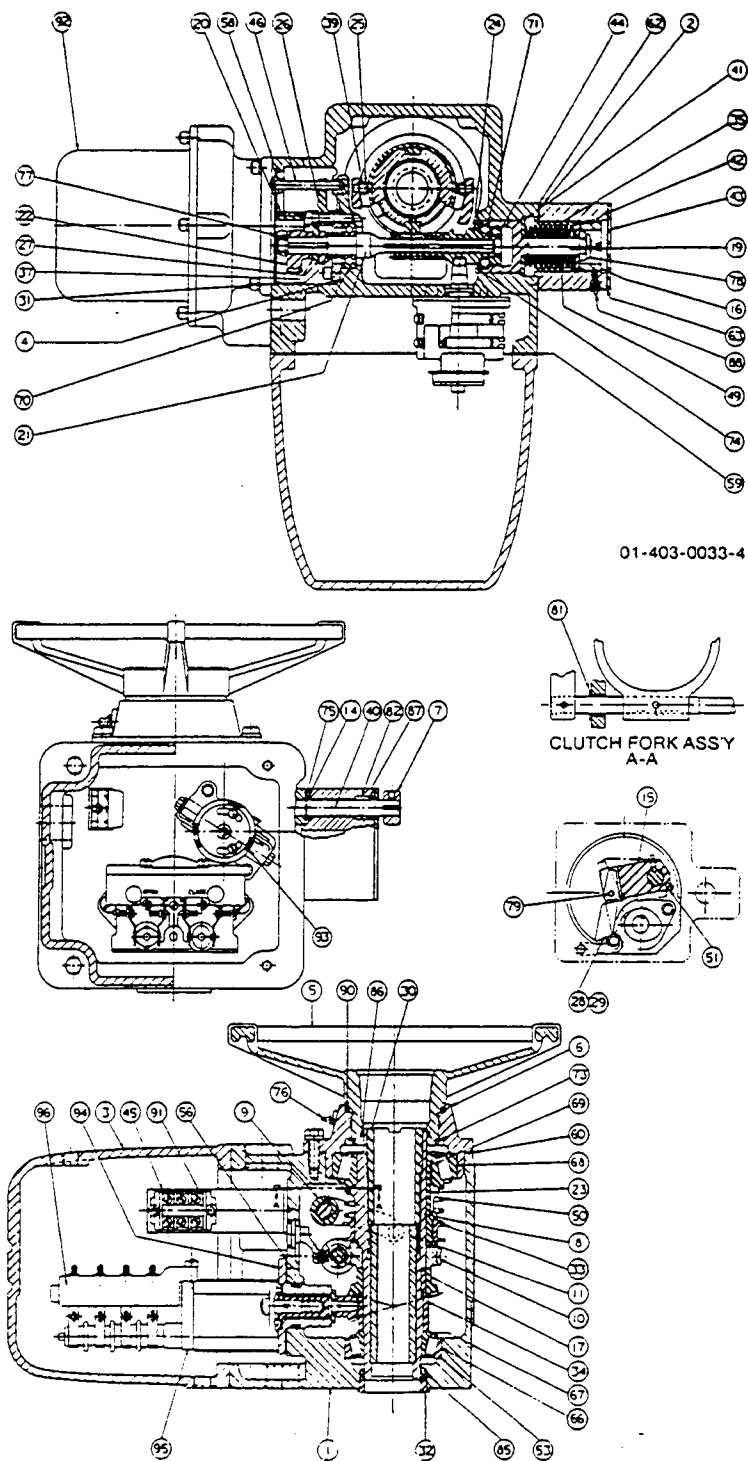


Figure 2.2 Limitorque SMB000 valve [2].

position of any rising stem type of operation and the geared limit switch is wired into the motor control circuit to stop the operation at the full-open position.

2.4 Anomalies in MOVs

The malfunctions of an MOV are closely related to its design. For example, misadjustments in limit and torque switches cause the most number of failures. These conditions result in characteristic behavior of MOV that can be detected by monitoring power signature. The following is a short list of some of the malfunctions.

- Misadjusted limit switch: Usually close-to-open stroke ends with the activation of the limit switch. Sometimes this switch ends the stroke too early. This can be seen as the shortening of time needed for the stroke. Other misadjustment include a switch that does not stop the stroke before the stem hits the back seat. In this case stroke time is increased and a peak occurs at the end of the stroke that is not normally there.
- Misadjusted torque switch: The power used at seating is indicated by the size and the time duration of the seating peak at the end of the open-to-close stroke power signature. If this peak is too small the power used to close the valve may not be enough to seal the flow. If the seating peak is too large, the power used for seating may cause damage to the valve.
- Tight packing: The main source of friction during stem movement from one position to another is the friction between stem and packing. The packing is used to isolate flow medium from environment. This friction may be large enough to cause a high mechanical load which may damage the valve.

- Loose movement: This malfunction depends on the design of the valve. In SMB000 design, for example, the movement caused by a loose stem nut may be harmful for the valve.
- Power supply malfunctions: The degradation of power supply cause a decrease in the actuator power. In this case, the actuator may not have enough power to operate the valve.
- Gear problems: If there are damages in the gear train these may be seen as periodic peaks in the power signature.
- Lubrication problems: These are seen in increased friction and power level.

CHAPTER 3

CURRENT METHODS FOR MOV TESTING AND MONITORING

3.1 Review of Prior Work

There are several parameters that may be used in MOV diagnostics. The MOVEX system, which was developed at The University of Tennessee, used the motor current signature [3,4]. Other parameters, that may be used for the diagnosis of an MOV include the following (Oak Ridge National Laboratory [4]).

1. Valve stem position,
2. valve stem velocity,
3. valve stem strain,
4. torque- and limit-switch actuations (times of occurrence),
5. internal and external motor temperatures,
6. vibration (several locations),
7. torque-switch angular position, and
8. motor current.

Research at ORNL established that motor current is a very important parameter for MOV diagnosis. It should be pointed out that the motor power signature was not used in any of this research. It was found that the motor power signature has certain

advantages over motor current [5]. A system that uses the motor power signature is the Motor Power Signal Analysis (MPSA) system, developed by the Duke Power Company [5]. The MPSA system was used to collect the motor power data for this project.

3.2 Requirements for Testing of MOVs in Nuclear Power Plants

The regulatory code 10 CFR 50.55 requires nuclear power plant operators to test pumps and valves in accordance with the requirements of Section XI of the American Society of Mechanical Engineers (ASME) Code. These codes and the Nuclear Regulatory Commission (NRC) regulations form the basis for the testing of pumps and valves in nuclear power plants.

There is an opinion that some testing performed according to the ASME Code is of low quality [6]. For example, the stroke time is one of the tests specified for MOVs by the ASME Code. This test is considered to be of little value when determining the operating condition of most MOVs. Two primary reasons are [6]:

- a stroke time test is usually conducted under light load conditions, and
- a valve would have to be in a very bad condition before the measured stroke time could be affected.

These two issues have motivated the search for new diagnostics methods.

CHAPTER 4

MOTOR POWER SIGNATURE

4.1 Introduction

In the past, motor current signatures were used to develop most MOV diagnostic techniques. It can be shown that motor power correlates better with the valve mechanical load than the motor current does [7,8,9,10]. This is very desirable if mechanical malfunctions are to be found as is the case in this work. One of the unique features of this research project is the use of motor power signature for monitoring MOVs.

4.2 Electrical Power

True power in a one phase system is computed using Equation (4.1).

$$P = UI \cos \phi \quad (4.1)$$

where:

P = True power.

U = RMS value of voltage.

I = RMS value of current.

ϕ = phase angle between the current and the voltage.

The true power is defined as the power that is used to perform mechanical work. The apparent power includes true power and the power used to maintain the magnetic field inside an electrical motor. The true power is never greater than the apparent power. If

the motor is a three-phase system, we need to take the power into account in all the three phases.

$$P = U_1 I_1 \cos \varphi_1 + U_2 I_2 \cos \varphi_2 + U_3 I_3 \cos \varphi_3 \quad (4.2)$$

where U and I refer to phase voltage and phase current, respectively. In practice the line voltage and phase current or the phase voltage and the line current are measured. Also, in many cases, it can be assumed that the phases are balanced. This means that the RMS values of the voltage and the current, as well as the phase angle between the current and the voltage are the same, for all the three phases. For a balanced system Equation (4.2) becomes

$$P = \sqrt{3}UI \cos \varphi \quad (4.3)$$

The power calculated by the above equations uses the RMS values of currents and voltages. The process of obtaining RMS values requires time averaging of the signals. This method is not suitable for MOV diagnosis, where a rapid change in the power level needs to be observed. What is needed is the instantaneous power, so that the power is known at each sampling point. In order to determine the power from voltage and current measurements, the knowledge of the instantaneous values of both current and voltage is necessary. The instantaneous power is then obtained from the following equation:

$$p = u_1 i_1 + u_2 i_2 + u_3 i_3 \quad (4.4)$$

where u_i and i_i are the instantaneous voltage and current in phase i . By convention the small letters denote the instantaneous values.

Figure 4.1 shows the equivalent circuit of an electrical motor [7]. The subscripts s and r refer to the stator and the rotor of the motor, respectively. The notation ' denotes that rotor values are normalized with respect to the stator. The circuit consists of resistances and reactances. The only value that is changed with the load is the slip s and with it the resistance R'_r/s . The slip defines the difference between the rotational speed of the magnetic field and the actual rotating speed of the rotor. The magnetizing current I_{ma} is independent of the load, and it is used to produce the magnetic field inside the motor.

The total current depends on both the magnetizing and load currents. The magnetizing current is the imaginary part of the current and the load current is the real part. The measured current is obtained from the following equation:

$$I = I_l + jI_{ma}, \quad \text{where } j = \sqrt{-1} \quad (4.5)$$

The same relationship is shown using a vector representation (Figure 4.2). The phase angle between the load and the magnetizing current is 90 degrees. The load current is in phase with the voltage and the phase angle between the voltage and the magnetizing current is 90 degrees. The magnetizing current is constant and the load current increases with the load. Because of this, the phase angle between the current and voltage is between 0° and 90° and decreases as the load increases.

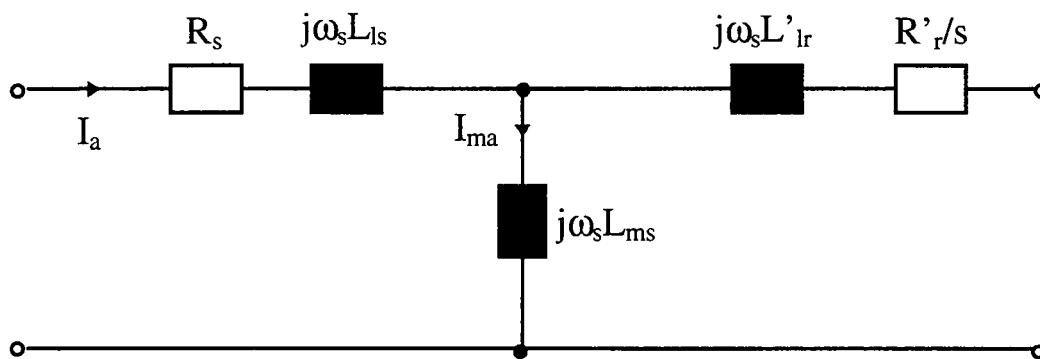


Figure 4.1. Per-phase equivalent circuit referred to the stator.

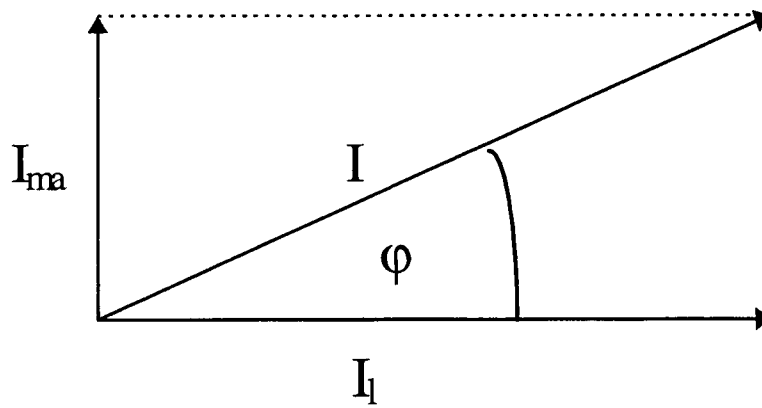


Figure 4.2. Vector diagram of motor currents.

The relationship that exists between the magnetizing and the load currents also exists between the magnetizing and the load power (see Figure 4.3). The power is calculated by measuring the instantaneous current and voltage. It enables the detection of the phase angle between the current and the voltage. This phase angle can be used to distinguish between the magnetizing and the load powers. The cosine of this phase angle ϕ , is called the power factor. The load power, obtained by using the power factor, can be directly correlated to the actual load experienced by the operator.

The power factor is expressed by the following equation:

$$\text{power factor} = \cos\left(\frac{P_l}{P}\right) \quad (4.6)$$

There is a linear dependence between motor power and the actual valve load. The magnetizing current makes the dependence between the current and the valve load to be nonlinear. This is especially true at low loads, where the magnetizing current is a larger part of the total current than at higher loads.

Both current and voltage measurements are needed to calculate the power. In earlier approaches to MOV diagnosis, motor current was the primary signature used. This is probably due to the fact that it is much simpler to measure only the current. If power measurement equipment is available, the power analysis method has a clear advantage due to its greater sensitivity to load changes.

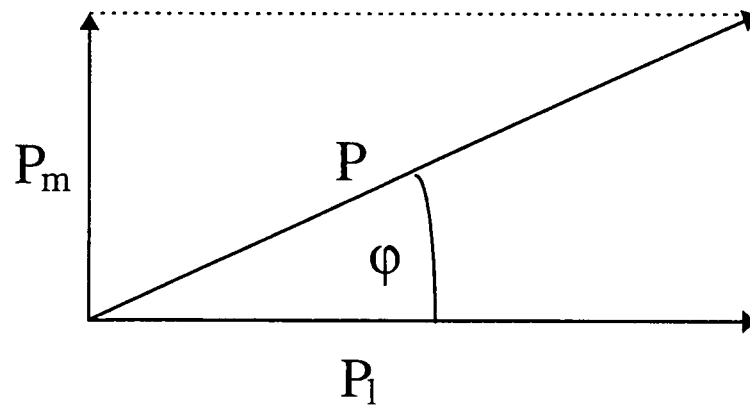


Figure 4.3. Vector diagram of motor power.

CHAPTER 5

DEVELOPMENT OF AN AUTOMATED DIAGNOSTICS SYSTEM FOR MOTOR-OPERATED VALVES

In this section the structure of the MOV diagnostics system and the general approach to the solution are presented. Syntactic pattern recognition was used to detect various events and timings using the motor power signature. These events are used by the expert system for detecting and isolating valve anomalies. The PC-based diagnostics system includes the following modules: signature pre-processing, feature extraction, decision making system using fuzzy logic, report generation, and database management.

5.1 Fuzzy Linguistic Description

A fuzzy logic decision module was developed as part of the overall diagnostics system. This provides a systematic approach for decision making, especially in situations where the anomaly type is not crisp. Fuzzy linguistic descriptions (or simply linguistic descriptions) are formal representations of systems made through fuzzy if/then rules [11]. Informal linguistic descriptions used by humans in daily life are usually the starting point for the development of fuzzy linguistic descriptions. The main difference between classical logic and fuzzy logic is that if/then rules are supposed to be true or false, and in fuzzy logic they can be true to a certain degree.

Data input and output of the expert system are linguistic variables. A linguistic variable is a variable whose arguments are referred to as fuzzy values. Examples of

linguistic variables as inputs to the system include stroke time, inrush peak power, running power, etc. Fuzzy values of the linguistic variable **stroke time** are very small, small, normal, big and very big. The output of the expert system are fuzzy variables that express the valve condition. A specific evaluation of a fuzzy variable, for instance, “stroke time is big,” is called a fuzzy proposition. Individual fuzzy propositions on either left-hand (LHS) or right-hand-side (RHS) of a rule may be joined by connectives such as AND and OR. For example, “stroke time is big AND running power is big.” Individual if-then rules are connected with the connective ELSE to form a fuzzy algorithm. The process of evaluating a fuzzy linguistic description is called fuzzy inference. Fuzzy inference will take place in the inference engine of the expert system.

5.2 Fuzzy Logic Inference

For the first time in this research on MOV diagnostics, fuzzy logic inference was used to improve decision making regarding the type of MOV anomaly. In practice, input to the system must first be fuzzified. The process of diagnosis is then performed on the fuzzy variables. After the inference procedure, the diagnosis is also expressed in fuzzy terms. Because of this, the output should be defuzzified each time a diagnosis is made. The fuzzification and defuzzification are important steps in this expert system. This was the reason why data structures for storing input and output fuzzy variables were chosen in this project in a manner that made fuzzification and defuzzification easy to perform [12].

During fuzzification, the degree of belonging of each variable to different categories is found. There are many possible choices of categories. The categories may

be defined as: negative large, negative medium, negative small, zero, positive small, positive medium, and positive large. Functions that map input crisp variables to the degree of belonging to each category are called membership functions. For the purpose of this project five categories, and therefore seven membership function were chosen. One of the possible choices for the membership functions is shown in Figure 5.1. During the fuzzification step seven numbers were found for each input variable. The fuzzification for one membership function is shown in Figure 5.2.

Once the variables are fuzzified the strength of each rule is computed using the min operation. Then the output of each rule is multiplied with each strength value and the final result is obtained by performing a max operation between outputs of all rules. This method is called max-product composition and it is shown in Figure 5.3. In this figure, rules with two fuzzy propositions are used and output from the system is the shaded area. Defuzzification is performed by the center of gravity method. The idea is to find the center of gravity of the shaded area that represents the output. The projection of the center of gravity on the x-axis is the value sought. This value is denoted by z_0 in Figure 5.3 and it represents the final result of the inference. This result is then used to generate the diagnostics information for the MOV being used.

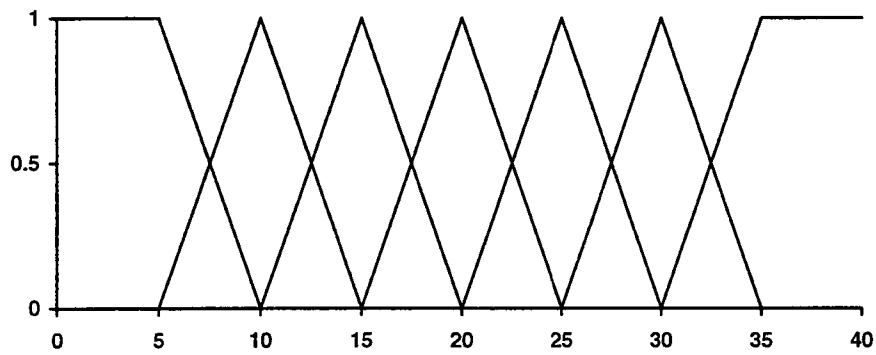


Figure 5.1 Membership functions.

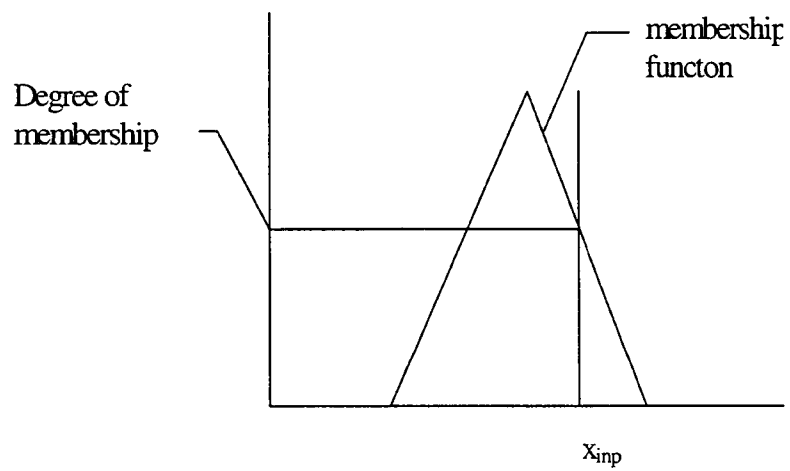


Figure 5.2 An Example of fuzzification.

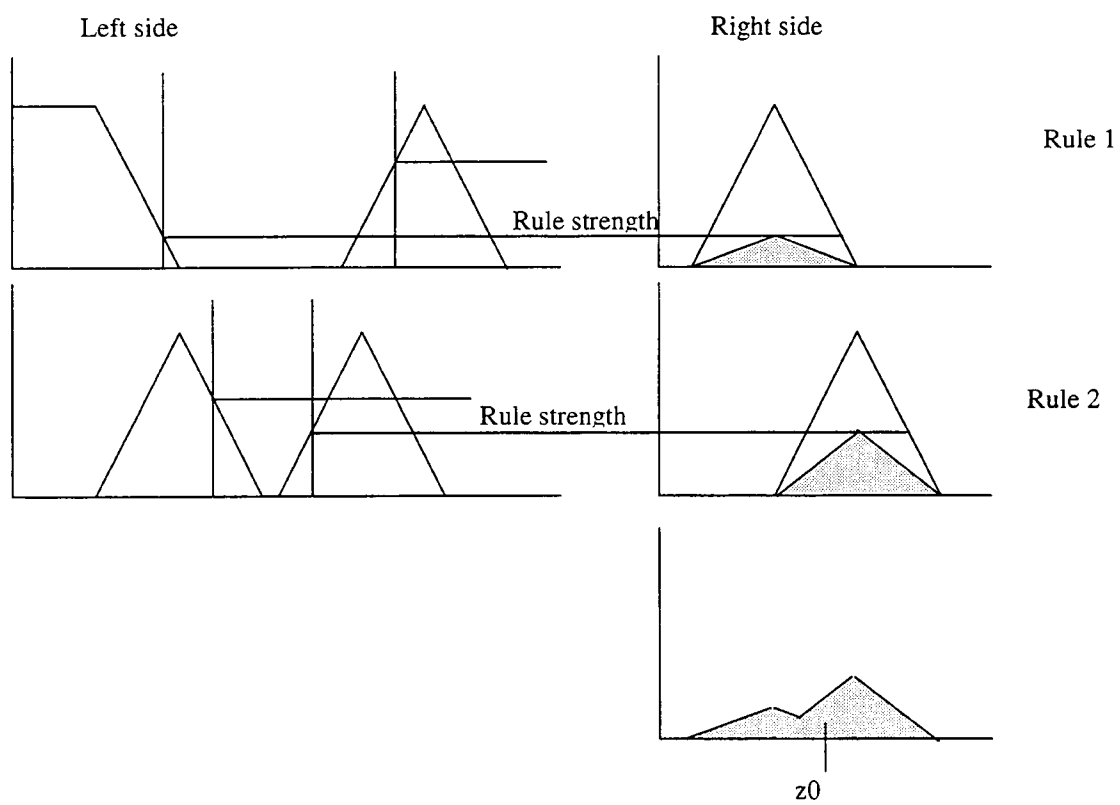


Figure 5.3 Max-product fuzzy composition.

5.3 System Structure

The system for automated diagnosis of MOV operational condition consists of many modules. The measured and digitized data need to be represented in a way that enables the creation of a knowledge base and application of the rule-base expert system to it. The elements of the system are shown in Figure 5.4. The important modules of the diagnosis system are

- Data preprocessing,
- Feature extraction,
- Decision making (Expert system) using fuzzy logic,
- Report generation, and
- Database management.

5.3.1 Data Preprocessor Module

The purpose of the preprocessor module is to retrieve data from the MPSA system and prepare them for further processing. There are two ways of retrieving the data. The data may be retrieved directly from the MPSA system using the DDE (Dynamic Data Exchange) approach. This approach allows the user to use the system for data acquisition and call the expert system from it. The second approach is to get the data from a data file that is created by the MPSA system, thus making the diagnostics system independent of the MPSA system. Another function of the preprocessor module is to remove events that

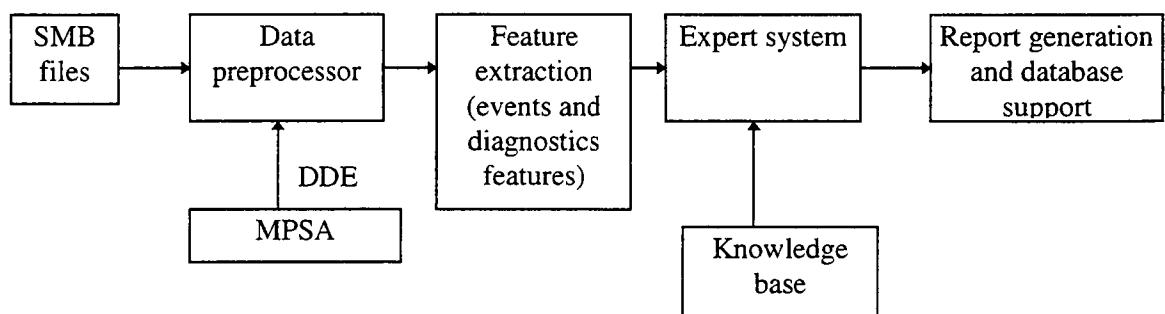


Figure 5.4 MOV diagnostics system structure.

do not represent the actual valve conditions in the power signature. An example of these events are negative power and artificial peaks.

The number of points in a motor power signature may be very large. The sampling time is 1/1000 second and the signature may be up to one minute long. This represents a number of points that is much larger than suitable for syntactic pattern recognition. The shape of a signature can be preserved with a smaller number of points, and the signature shape is what is used in syntactic pattern recognition. The preprocessor module selects points from the signature and creates a new signature with fewer points. The original signature is divided into intervals, and the interval is replaced by a point with a maximum value in that interval. Our experience shows that this method does not distort the shape of the power signature.

The power signature is represented by the magnitude of power at a certain time. This representation is not very suitable for processing in the inference engine. The signature may be several thousand points long. It is more convenient if the signature consists of a smaller number of descriptive elements. One way to achieve this is to represent the signature as a set of simple events which can be found using syntactic pattern analysis. The events used in this work are peaks, steps and ramps and the associated time information. These analyses take place in the decomposer module.

5.3.1.1 Decomposition Algorithm

The original algorithm was developed by Eklund and Upadhyaya [13]. The algorithm was written in Fortran for use under the DOS operating system. Although the

slightly modified Fortran program was linked to the rest of the expert system software, it was decided that the best way of using the algorithm was to translate it to C++ code. The main reason for this is that C++ has a variety of tools for memory management and these tools can increase the memory that is utilized by the algorithm. Specifically, the dynamic memory allocation was used in the C++ version of the program. This allowed the program to allocate memory when it was needed and not at initialization of the program.

The main purpose of the syntactic pattern recognition is to identify simple geometric shapes in the motor power signature. It is assumed that the shape of the signature is preserved if the signature is presented as the sum of simple geometric shapes or primitives. Therefore, instead of the time series, the signature may be written in the following form:

$$y = \sum_{i=1}^{N_{\delta}} a_i \delta(t - \tau_{1i}) + \sum_{i=1}^{N_u} b_i u(t - \tau_{2i}) + \sum_{i=1}^{N_r} c_i r(t - \tau_{3i}) + \eta(t) \quad (5.1)$$

where

$\delta(t), \tau_{1i}$ - impulse components and its time instant

$u(t), \tau_{2i}$ - step components and its time instant

$r(t), \tau_{3i}$ - ramp components and its time instant

N_{δ}, N_u, N_r - number of peaks, steps and ramps

$\tau_{1i}, \tau_{2i}, \tau_{3i}$ - time instants of impulse, steps and ramps

a_i, b_i, c_i - amplitudes of impulse, steps and ramps

$\eta(t)$ - random noise

The primitives are described by a series of numbers:

I. Peaks :

- A. time of occurrence
- B. absolute height
- C. relative height
- D. start time
- E. end time
- F. start amplitude
- G. end amplitude

II. Steps

- A. time of occurrence
- B. height

III. Ramps

- A. start time
- B. end time
- C. slope

These are shown in Figure 5.5.

5.3.1.2 Peak Detector

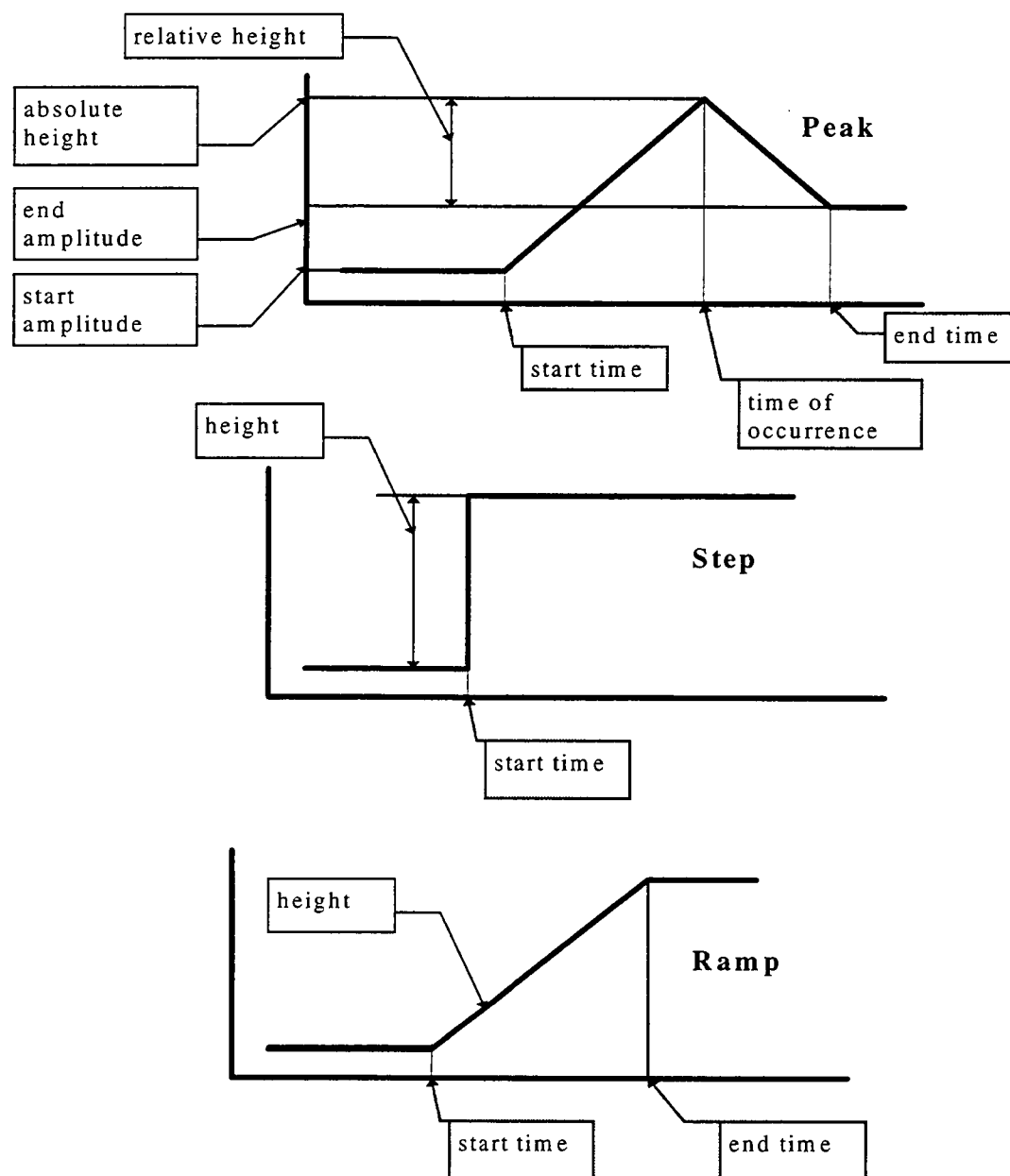


Figure 5.5 Description of signal primitives in the program.

An algorithm developed by Marshal [14] was used. The algorithm is capable of finding the following peaks features: time of occurrence, absolute height, relative height, and start and end points. The main advantage of this algorithm is its insensitivity to signal noise and baseline drift. This allows the algorithm to be applied to the signal without any previous filtering or smoothing that may distort the signal.

5.3.1.3 Waveforms Peaks

A set of z-peaks, $P(z) = \{p_1, \dots, p_n\}$, exist on interval $Y(t) = [a, b]$ if there exists an associate set of valleys $\{v_1, \dots, v_{n+1}\}$ such that

$$a \leq v_1 \leq p_1 \leq v_2 \leq \dots \leq v_n \leq p_{n+1} \leq b$$

and

$$Y(p_i) = \max Y(t), \quad v_i < t < v_{i+1}$$

$$Y(v_i) = \min Y(t), \quad p_{i-1} < t < p_i$$

$$l_i = Y(p_i) - Y(v_i) > z \text{ and}$$

$$r_i = Y(p_i) - Y(v_{i+1})$$

where

$$p_0 = a \text{ and } p_{n+1} = b.$$

5.3.1.4 Peak detection algorithm

The algorithm may be written in the following form:

Find the set of local minima and local maxima, $P(0)$

peak heights $h_i := \min(l_i, r_i)$, $i=1, \dots, n$,

$P(\text{current}) = P(0)$

```

while  $h_j = \min(h_i) < z$ ,  $i=0, \dots, n+1$ :
    delete the peak from  $P(\text{current})$ 
    reevaluate the valley between  $p_j$  and  $p_{j+1}$  as the smaller of  $v_j$  and  $v_{j+1}$ 
 $P(z) = P(\text{current})$ 

```

The peak duration is found by using another set of peaks $P(w)$ where $w \leq z$ and $P(z)$ is the original peak set. Then the duration of peak p_i is the time between adjacent valleys of p_i in $P(w)$.

5.3.1.5 *Detection of Steps and Ramps*

The schematic diagram of the detection logic is given in Figure 5.6. The following is the description of the detection algorithm modules.

A. Median Filter

The median filter is used to smooth the signal and then remove most of the noise. The filter replaces every sample of the signal by the median of the sample values in its immediate neighborhood. It can be written as follows:

$$y(i) = \text{median}\{y(j); j = i - \mu, i - \mu + 1, \dots, i - 1, i, i + 1, \dots, i + \mu - 1, i + \mu\}$$

where μ is the length of the filter.

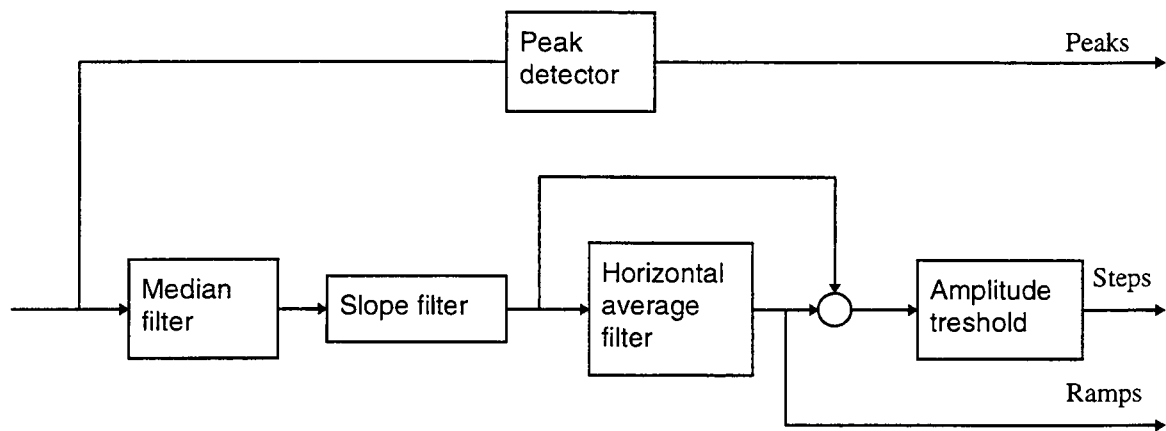


Figure 5.6 Schematic diagram of the pattern recognition algorithm.

B. Slope Filter

The output of the median filter is the input to the slope filter. The slope filter determines the slopes of low frequency components in each sample of an input. The least-squares fit to the samples in the immediate neighborhood of each sample is used to find the slope at each sample. Because the derivative is not performed locally, but rather on the neighborhood of the samples, the method is not sensitive to noise. The disadvantage of this method is that it is computationally more intensive than simple methods that use first differences of the signal.

C. Horizontal Averaging Filter

If there is a ramp in the signal, the slope filter will recognize it as a continuously positive or continuously negative slope. The purpose of the horizontal averaging filter is to detect such regions in the output of the slope filter. The horizontal averaging filter replaces samples with the average of all samples in a segment if the length of the segment is larger than a horizontal threshold and that every sample in the segment is larger than the vertical threshold.

D. Detection Logic

Beginning and ending of the steps will be seen as the peaks in the output of the slope detector. Step detection logic integrates the impulses that are greater than the step detection threshold. The time of occurrence is the time when the impulse is recorded and

the height of the step is the value of impulse integration. Ramps on the other hand are detected from a horizontal averaging filter. The detection logic determines start and end times of the ramp and its slope.

5.3.2 Feature Extraction

The next step in the process is to determine the events from the signature primitives. The most characteristic events in the open-to-close stroke are:

1. Inrush peak,
2. Start of stroke step,
3. Start of stem motion step,
4. Seating peak, and
5. End of stroke step.

For close-to-open stroke the important events are:

1. Inrush peak,
2. Hammerblow peak,
3. Unseating peak, and
4. End of stroke step.

These events are shown in Figure 5.7 and Figure 5.8.

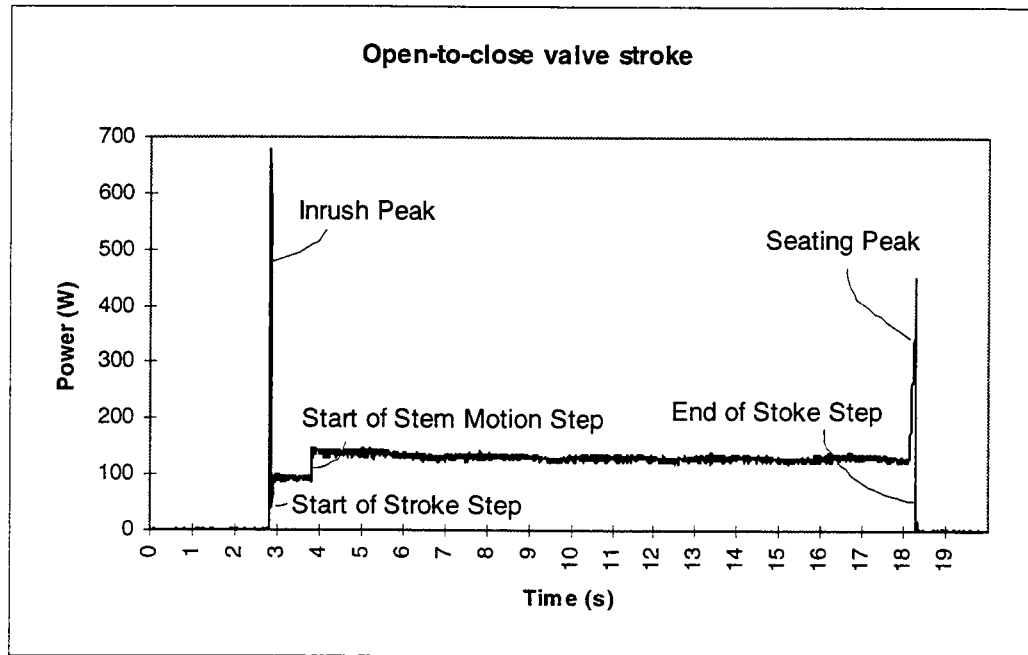


Figure 5.7 Characteristic events in the open-to-close valve stroke.

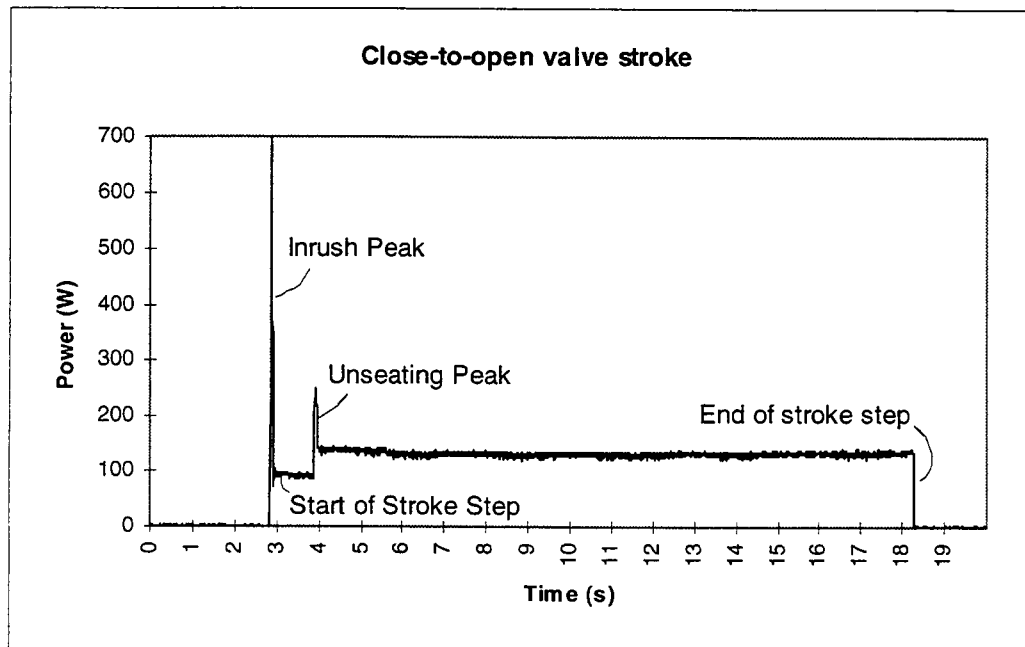


Figure 5.8 Characteristic events in close-to-open valve stroke.

The power signature is used to discern features that aid in the determination of valve conditions. These features are detected using power signature primitives rather than the power signature itself. Some of the features may be more easily extracted from voltage and current signals. The measured voltage is used to detect the supply voltage degradation. The current analysis is suitable for detecting motor phase unbalance.

5.3.3 Diagnostics Module - Expert System

Once the power signature is obtained, its characteristic events can be determined. The comparison of these events with the reference events provides information about valve conditions. The differences between measured and reference values are passed to the inference engine. Also, the rules describing the diagnosis process are needed in order to obtain the MOV condition. The schematic diagram of the expert system software is shown in Figure 5.9.

5.3.3.1 Knowledge Base

A knowledge base contains procedural and declarative knowledge. Procedural knowledge is represented by the rules and declarative by the facts. For example “**IF** the power level is too big **and** stroke time is too big **THEN** the packing is too tight.” An example of the declarative knowledge or fact is “The normal load power is 200 Watts” or “The stroke time is 15.5 s.” The knowledge should be represented in the form that allows for easy implementation in the computer.

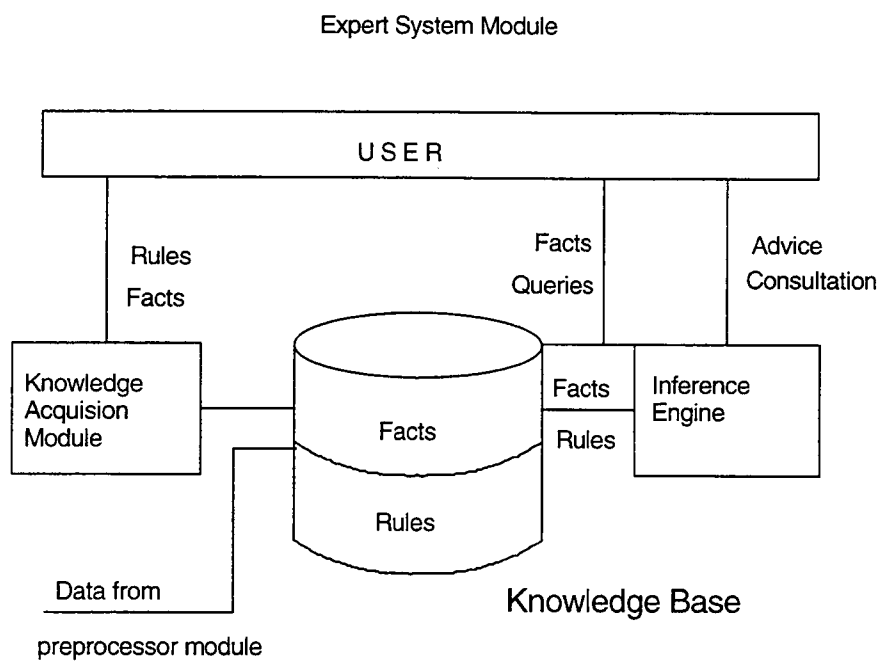


Figure 5.9 Schematic diagram of the expert system software (modified from J. Martin and S. Oxman, 1988[15])

5.3.3.2 Inference Engine

A comparison of measured and reference values are made in the inference engine, and the rules created for different valve types are used during this process. The inference engine is a rule based algorithm and uses fuzzy logic. The result of the analysis performed by the inference engine will be the diagnosis information.

The fuzzy logic inference engine was written in the C++ programming language. The structure of the inference engine reflects the object oriented programming techniques built in C++. The structure of the fuzzy variables are shown in Figure 5.10. All fuzzy variables in the input and output part are stored in list data structure. Therefore, input to the inference engine is a list of all input fuzzy variables. Also, the inference engine output is the list of all output variables. Each fuzzy variable contains name, value and a list of its membership functions. The data structure for membership function has a very similar organization. It also contains name and value, and the data describing the shape of the data structure. The shape of the membership function used by this program are trapezoids. This limits the variations of shapes used in the system, but it seems that the results of the inference are highly reliable.

This general data structure is applied to input and output variables. There is a difference, however, in how the shapes of the membership functions are defined within the program between input and output variables. The difference exists in order to make the implementation of fuzzification and defuzzification straightforward.

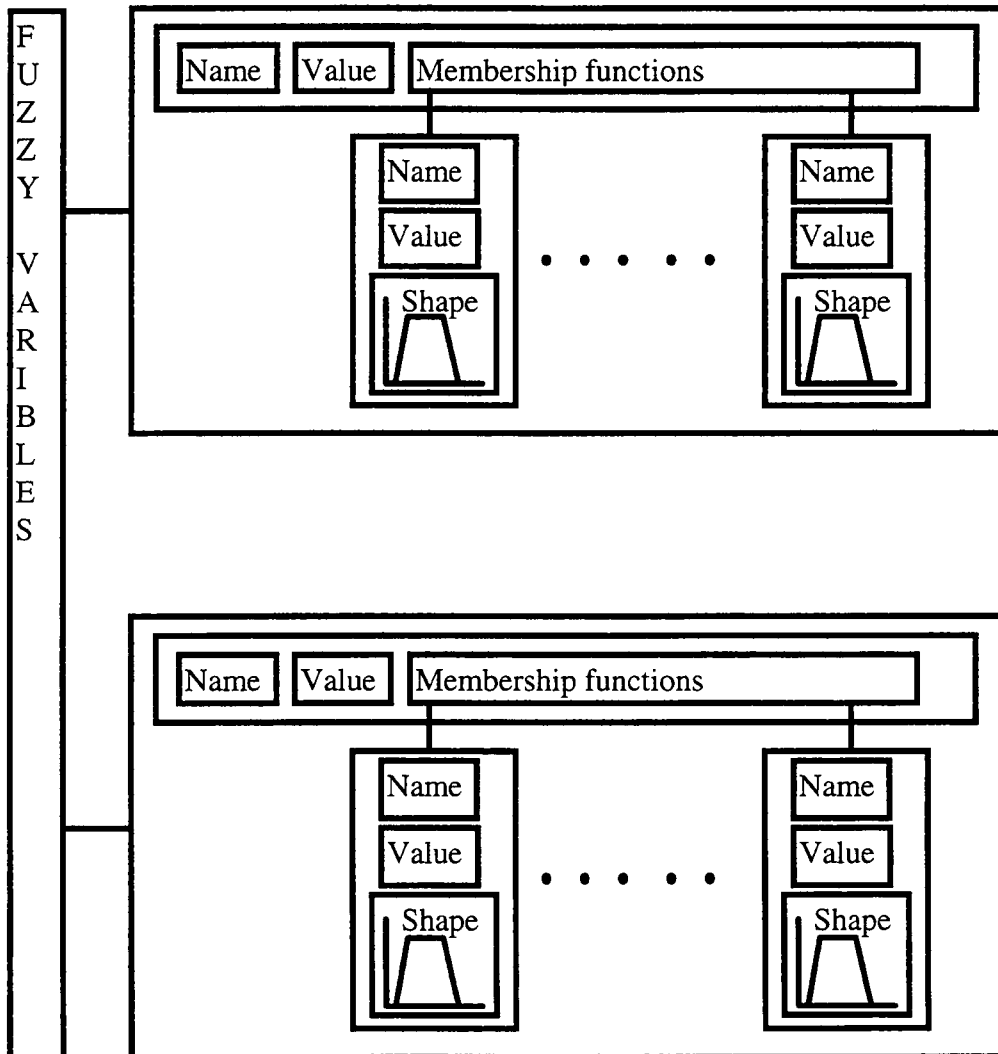


Figure 5.10 The data structure of fuzzy variables.

A. Fuzzification

The shape of the membership function for input variables consists of four values. These values describe the following quantities: first point, last point, beginning slope and ending slope. This is shown in Figure 5.11.

During fuzzification the point where a vertical line, representing the actual value, crosses the membership function is found (Figure 5.12). The projection of this point on the y axis is the value of the membership function. This can be done easily if the lines that are not parallel to the x-axis are well defined, and this is the case if one point and the slope of the line is known.

The membership functions are defined for each input variable. Due to seven different linguistic categories, seven membership function were defined. The initial values of the membership function are given in the following C statement:

```
int value_list[7][4]= { {0,0,31,63},
                        {31,63,63,95},
                        {63,95,95,127},
                        {110,127,127,159},
                        {127,159,159,191},
                        {159,191,191,223},
                        {191,223,255,255}};
```

The span of the input range is normalized to be in the range [0,255]. There are seven elements, with four values each. These four values are then used to determine the parameters of the data representing membership functions. The parameters are: first point, beginning slope, ending slope and last point.

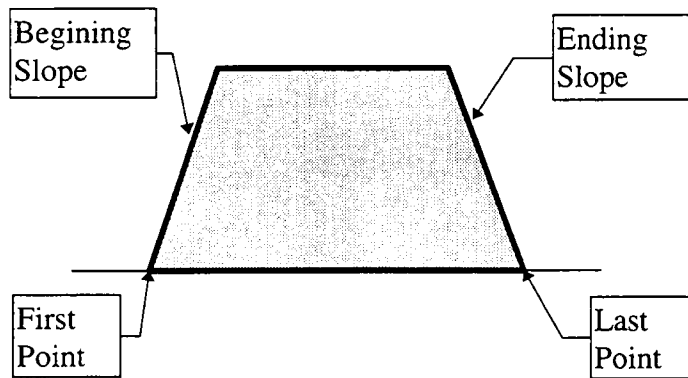


Figure 5.11 Data structure for input variable.

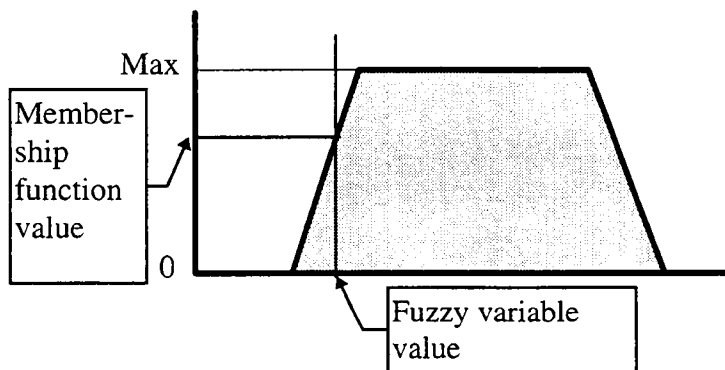


Figure 5.12 Fuziffication.

B. Rule Evaluation and Defuzzification

Each rule is tested in order to obtain its truth or strength. A fuzzy logic rule provides a degree of association of a certain anomaly. This is described by an example of an actual application to an MOV. The following are the three rules used in the expert system:

- 1) IF Running_Power is PS THEN Packing is PS.
- 2) IF Running_Power is PM THEN Packing is PM.
- 3) IF Stroke_Time is PS THEN Packing is PM.

The left-hand sides in the above rules (left of THEN) represent the input (e.g. Running Power) to the inference engine. The right hand side of a rule contains one or more output variables. In the example above there is one output variable, namely, the valve Packing. The value of the fuzzy variable is defined by one of the membership functions. If the Running_Power is ZE, this indicates normal operation. If the Running_Power is significantly larger than the normal value, then the Running_Power will be PL (Positive Large). The membership functions used in the Expert System are shown in Table 5.1.

The values for the input variables were obtained using measured and reference values. For the current example, let the reference value for the Stroke_Time be 20 seconds and the measured value be 28 seconds. The reference and measured values for the Running_Power are 100 and 140 watts, respectively. From Figures 5.13 and 5.14 we

Table 5.1 Membership functions

Membership function	Abbreviation
Negative large	NL
Negative medium	NM
Negative small	NS
Zero	ZE
Positive small	PS
Positive medium	PM
Positive large	PL

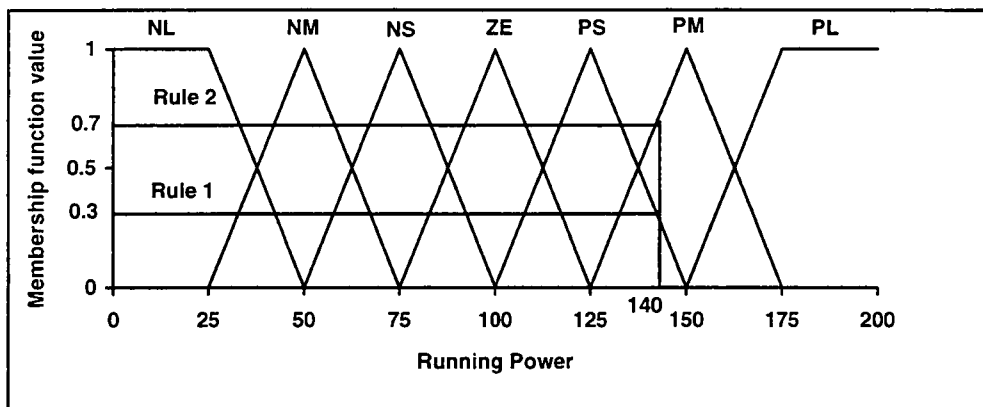


Figure 5.13 Rules 1 and 2.

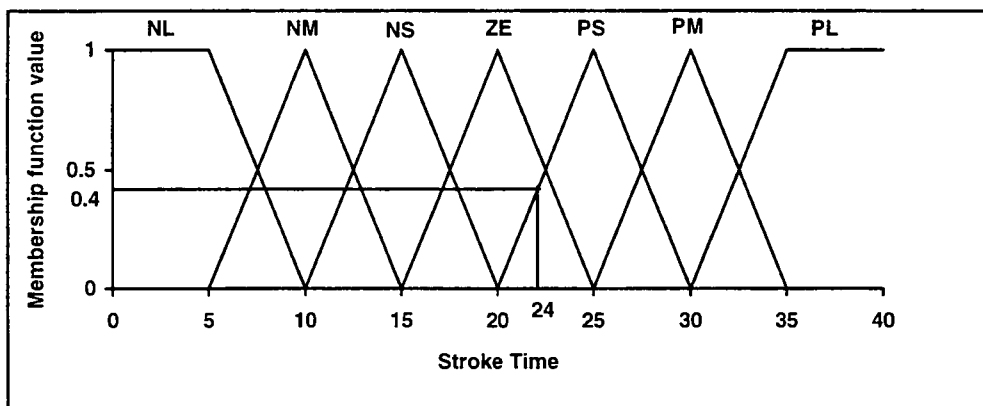


Figure 5.14 Rule 3.

observe how the rules are evaluated. After the evaluation, each rule is assigned a value. Values for rules 1, 2 and 3 are shown in Table 5.2.

The right-hand sides of the rules can be combined to give a single value of the output. The center of gravity is known for each membership (Figure 5.15). The combined output is shown in Figure 5.16. The centers of the output membership functions along the x-axis are:

$$PS = \frac{5}{8} = 0.625$$

$$PM = \frac{6}{8} = 0.75$$

(It is assumed that the x-axis starts at 0 and the maximum x value is 1). To obtain the final result the strength of the rules and their centers of gravity are used:

$$\frac{(0.3)\frac{5}{8} + (0.7)\frac{6}{8} + (0.4)\frac{6}{8}}{0.3 + 0.7 + 0.4} = 0.72$$

In this case the Packing is close to PM (positive medium).

C. Rule base

The list data structure is used to store the rules and is illustrated in Figure 5.17. Each Table element of the rule list is a particular rule. A rule itself contains two lists: input variables (if side) and output variables (then side). The variables inside the rules are connected with the logical AND operation. Therefore, the strength of a rule is the lowest value of the input variables. Rules are connected with the logical OR operation.

Table 5.2 Rule evaluation

	Output variable	Membership function	Value
Rule 1	Packing	PS	0.3
Rule 2	Packing	PM	0.7
Rule 3	Packing	PM	0.4

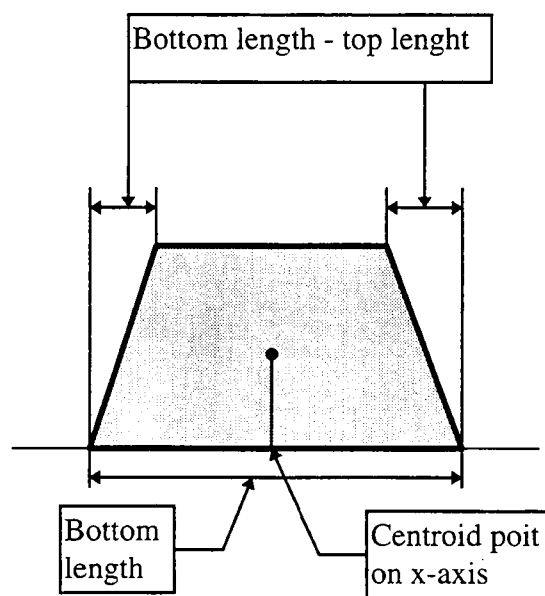


Figure 5.15 Output membership function.

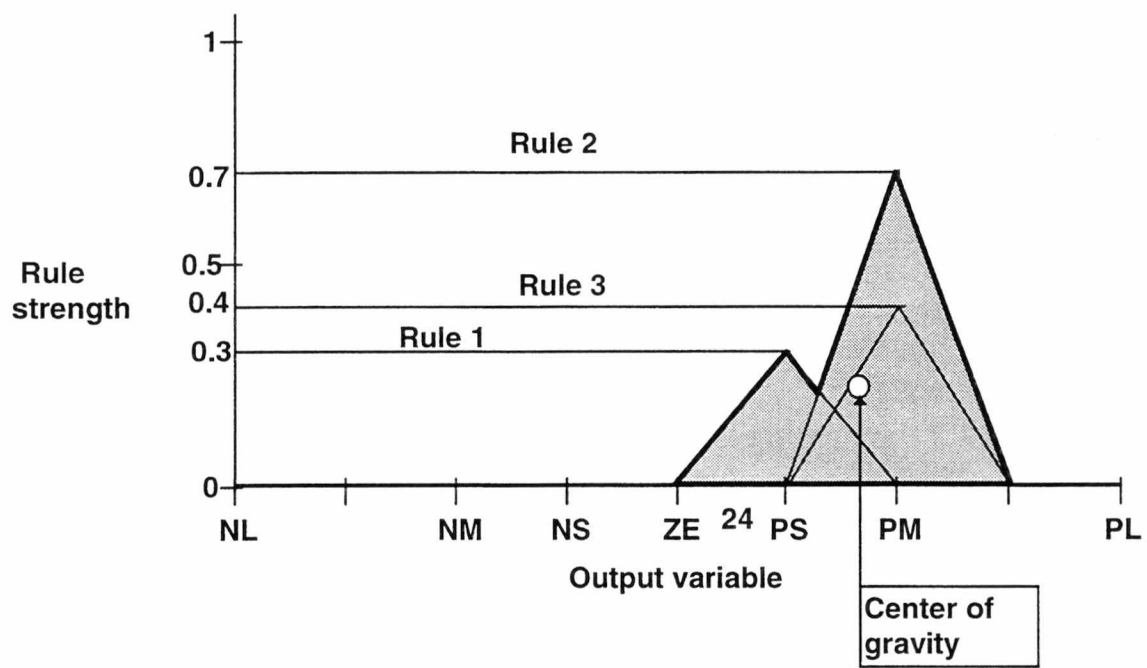


Figure 5.16 Defuzzification.

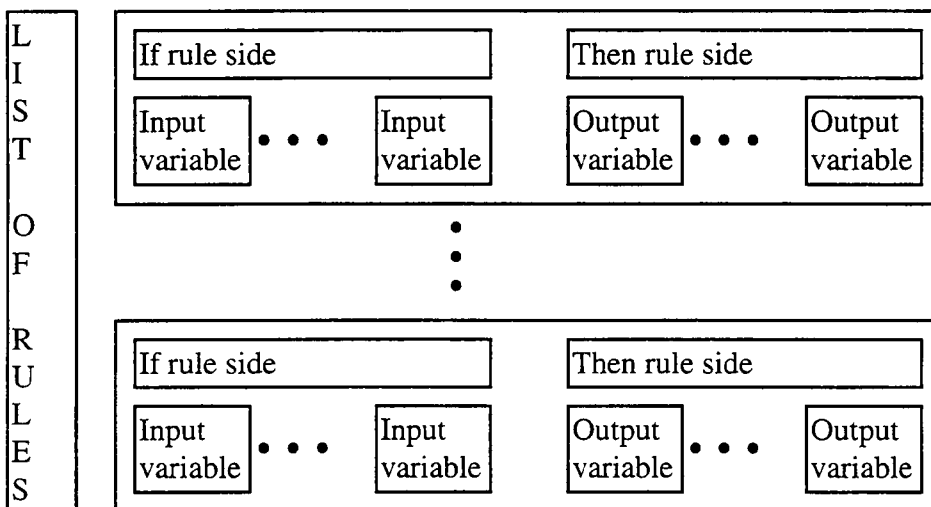


Figure 5.17 Rule base data structure.

That means that if several rules have the same output, the corresponding output variable will get a value of the strongest rule (largest output value).

5.3.4 System Implementation

The software is executed under the Microsoft Windows operating system. This makes it necessary to use software developing tools that enable development of the program as a windows application. Another requirement is that the development system allows for the creation of an expert system, including the manipulation of different data structures, such as a list. The Microsoft Visual C++ fulfills both the requirements [16, 17]. The use of an object-oriented approach also makes it easier to develop an expert system.

One of the main ideas behind the object oriented programming is to allow for easier development of large programs. This is accomplished by encapsulating the pieces of codes within blocks, or objects of certain classes, that comprise a program. The functions and data inside an object can be accessed by member functions of an object that are designated for that purpose. Therefore, the rest of the program can communicate with the object through these access functions. If the code inside the class is changed, the way of communication with the rest of the program is not. This feature makes it easy to change the code without affecting the rest of the program.

The Microsoft Foundation Class (MFC) Library contains C++ classes that may be used in creating a user interface within the Windows environment. Several classes from the MFC are used in the system as the basis for creating data windows and dialog boxes.

The system uses a rule-based inference engine to obtain the diagnosis of the MOV condition. The processing of the rules and facts requires a method of working with the lists. Again, an MFC class is used to implement the capability of list manipulation.

5.3.4.1 C++ Classes Created for Expert System

One of the characteristic features of the C++ programming language is object-oriented programming support. Each object belongs to some of the previously defined classes. Many classes from the MFC library are used directly. Some of the MFC classes are used to derive custom made classes.

A number of new classes, specific to this research project, were defined. Most classes are derived from the CObject and CObList classes found in the MFC library. This allows the derived classes to inherit properties of Serialization and List manipulation.

The following classes were designed specifically for this project:

- CData** - contains the power signature and additional information about valves.
- CPrimitives** - contains the power signature decomposed into primitives - peaks, steps and ramps. Primitives are represented by lists whose elements are objects of CPeak, CStep and CRamp.
- CEvent** - a class that is used to store characteristic events during the stroke. This is also used to describe diagnostic features.
- CEvList** - a list of characteristic events during the stroke.
- CDFList** - a list of features used in diagnosis.
- CInferEngine** - a class that contains functions of the fuzzy logic inference engine.

The program contains objects of these classes. The class function for the most important classes are listed in Tables 5.3-5.7. This approach to software development made the implementation of the diagnostic system easy compared to the traditional programming languages. At the same time, it also allowed the easy implementation of the Windows operating system features.

Table 5.3 CPrimitives member functions

CPrimitives	
CObList* GetPeaks()	Returns a pointer to a list of peaks found.
CObList* GetSteps()	Returns a pointer to a list of steps found.
CObList* GetRamps()	Returns a pointer to a list of ramps found ramps.
Cstring GetString()	Returns a string with explanation about the primitives found.
SetPeaks(const CObList&)	Finds peaks.
SetSteps(const CObList&)	Finds steps.
SetRamps(const CObList&)	Finds ramps.
EmptyPeaks()	Empties list that contains peaks.
EmptySteps()	Empties list that contains steps.
EmptyRamps()	Empties list that contains ramps.

Table 5.4 CData member function

CData : public CObject	
GetPrimitives(CPrimitives&)	Returns the pointer to the primitives.
int GetNSamples()	Returns the number of sampling points.
float GetTSample()	Returns the sampling interval.
CString GetPlant()	Returns the name of the plant.
CString GetValve()	Returns the type of valve.
CString GetStroke()	Returns the type of stroke.
CString GetDate()	Returns date.
CString GetReason()	Returns reason.
CString GetTester()	Returns the name of test engineer.
float * GetData()	Returns the array of sampling points.
SetNSamples(in)	Sets the number of sampling points.
SetPlant(CString)	Sets the name of the plant.
SetTSample(float)	Sets the sampling time.
SetValve(CString)	Sets the type of valve.
SetStroke(CString)	Sets the type of stroke.
SetDate(CString)	Sets date.
SetReason(CString)	Sets the reason for testing.
SetData(float*, int)	Sets sampling points.
PlotGraph()	Plots graph of the power signature.

Table 5.5 CEvList member function

CEvList	
void Empty()	Empties list.
CEvent * FindObject(int)	Finds event of certain type.
BOOL SetEvents(CPrimitives&,CString)	Finds events.

Table 5.6 CDFList member functions

CDFList	
CEvent * FindObject(CString)	Finds object given object name.
void Empty()	Empties list.
BOOL SetFeatures(CEvList&,CString)	Finds diagnostics features.
virtual void Serialize(CArchive& ar)	Serializes list.

Table 5.7 CinferEngine member function

CinferEngine	
CObList *GetRuleBase()	Returns pointer to the list of rules.
CObList *GetSystemInputs()	Returns pointer to the list of rules.
CObList *GetSystemOutputs()	Returns pointer to the list of rules.
void read_rules(CString)	Creates a list of rules from the ASCII file.
void fuzzification()	Performs fuzzification of the input variables.
void rule_evaluation()	Determines a fuzzy output using rules and inputs.
void deffuzzification()	Finds an output value from the fuzzy output variables.
void EmptyInputs()	Empties the list of input variables.
void EmptyOutputs()	Empties the list of output variables.
void EmptyRules()	Empties the list of rules.
BOOL SetInput(CDFList&,CDFList&)	Finds the input variables from the list of diagnostic features.
BOOL SetOutput(CString)	Creates a list of output variables.

CHAPTER 6

APPLICATIONS TO MOTOR-OPERATED VALVE DIAGNOSIS

The Limitorque SMB000 valve was setup at The University of Tennessee for simulating MOV anomalies and for data acquisition. The motor power signature analysis equipment was used to acquire motor power and other electrical parameters from test MOVs. The equipment includes transducers, A/D converter and PC-software for data acquisition.

6.1 Measurements

Measurements were performed using transducers for current and voltage. These two are combined to give power and power factor. For each measurement, eight parameters were determined:

- three currents (one for each phase)
- three voltages (one for each phase)
- power
- power factor.

Transducers are used to convert values of measured parameters into corresponding voltage levels. Therefore, there are eight voltage values at a time. These voltages form eight channels of input to the computer interface. In the computer interface analog voltage values are transformed into numbers using A/D converter. Analog values

are represented as time series for a specified time interval. The sampling time (time interval between adjacent samples) was chosen to be 0.001 second.

The final result of the measurement session is a computer file containing digitized record of the electrical signatures. This file can be loaded into the diagnosis software system. The file is recorded in binary format. A code was developed that enabled reading MPSA files and conversion to a data format that was suitable for further analysis. The schematic diagram of MPSA measurement system is shown in Figure 6.1.

6.2 Valve Test Data

The system was tested with the data that are collected during the experiments. Two SMB000 actuators were used with different valves: the valve at The University of Tennessee was one-inch globe valve and the valve tested at a power company was a three-inch gate valve. The various experiments are listed in Tables 6.1 and 6.2. The plots of the power signatures obtained during these tests are shown in Figure 6.2-Figure 6.42. These show the data from the various imposed anomalies on the valve system.

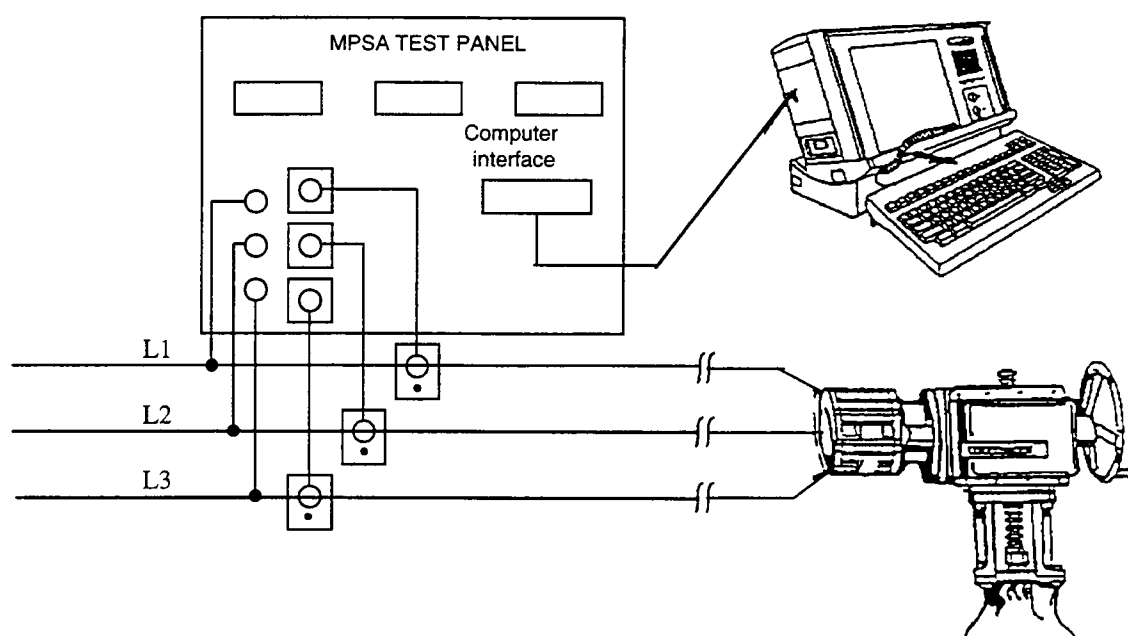


Figure 6.1 Schematic diagram of the measurement system.

Table 6.1 Data obtained at The University of Tennessee

LIMITORQUE SMB000 VALVE DATA			
Num.	File name	Stroke	Description
1.	UTBASLCO	CO	Baseline
2.	UTBASLOC	OC	Baseline.
3.	UTCLOLM	OC	Close on limit switch.
4.	UTOPTQ	CO	Open on torque switch.
5.	UTTQST1	OC	Torque switch set on 1.
6.	UTTQST2	OC	Torque switch set on 2.
7.	UTTQST25	OC	Torque switch set on 2.5.
8.	UTLMADJ	CO	Misadjust limit switch - deenergize earlier.
9.	UTLMADJ2	OC	Misadjust limit switch - hard seat.
10.	UTIRP1	CO	Resistor in one phase.
11.	UTIRP1-1	OC	Resistor in one coil.
12.	UTIRP1-2	CO	Resistor in one coil.
13.	UTSGPHPO	CO	Single phase after starting.
14.	UTSGPHBS	CO	Single phase prior to start.
15.	UTRVRC	CO	Reduced voltage control logic.
16.	TESTV		Voltage test.
17.	UTTGTPK	CO	Tight packing.
18.	UTTGTPK1	OC	Tight packing.
19.	UTMPNOFF	CO	Motor pinion disconnected.
20.	UTOBSTVS	OC	Obstruction in valve seat.
21.	UTBASLN2	OC	Baseline.
22.	UTBASLN3	CO	Baseline.
23.	UTBELVIL	OC	Washer removed from the spring pack.
24.	UTBELVI2	CO	Washer removed from the spring pack.
25.	UTRV10%	CO	Voltage reduced by 10%.
26.	UTRV10%2	OC	Voltage reduced by 10%.
27.	UTRV15%	CO	Voltage reduced by 15%.
28.	UTRV15%2	OC	Voltage reduced by 15%.
29.	UT100%	CO	Normal voltage.
30.	UT100%2	OC	Normal voltage.
31.	UT20%	CO	Voltage reduced by 20%.
32.	UT20%2	OC	Voltage reduced by 20%.
33.	UTVOUBAL	OC	Unbalanced phases - grounded.
34.	UTVOUBA2	CO	Unbalanced phases.
35.	UTDCM	OC	Declutch mechanism malfunction.

Table 6.2 Data from a power company

LIMITORQUE SMB000 VALVE DATA			
Num.	File name	Stroke	Description
1.	BDRTRUT	CO-OC	Broken rotor bar.
2.	GDRTRUT	CO-OC	Good rotor.
3.	BKPINGR	CO-OC	Bad pinion gear.
4.	DESPLOK	CO-OC	Loose locknut.
5.	SPKCOMP	CO-OC	Tight spring pack.
6.	BDWMGR	CO-OC	Bad worm gear.
7.	LSNLTN	CO-OC	Loose stem nut locknut.
8.	WLSUT	CO-OC	Disconnected limit switch.
9.	LSDBOTH	CO-OC	Disconnected both limit switches.
10.	MADJLMSW	CO-OC	Misadjusted limit switch.
11.	MADJTQSW	CO-OC	Misadjusted torque switch.

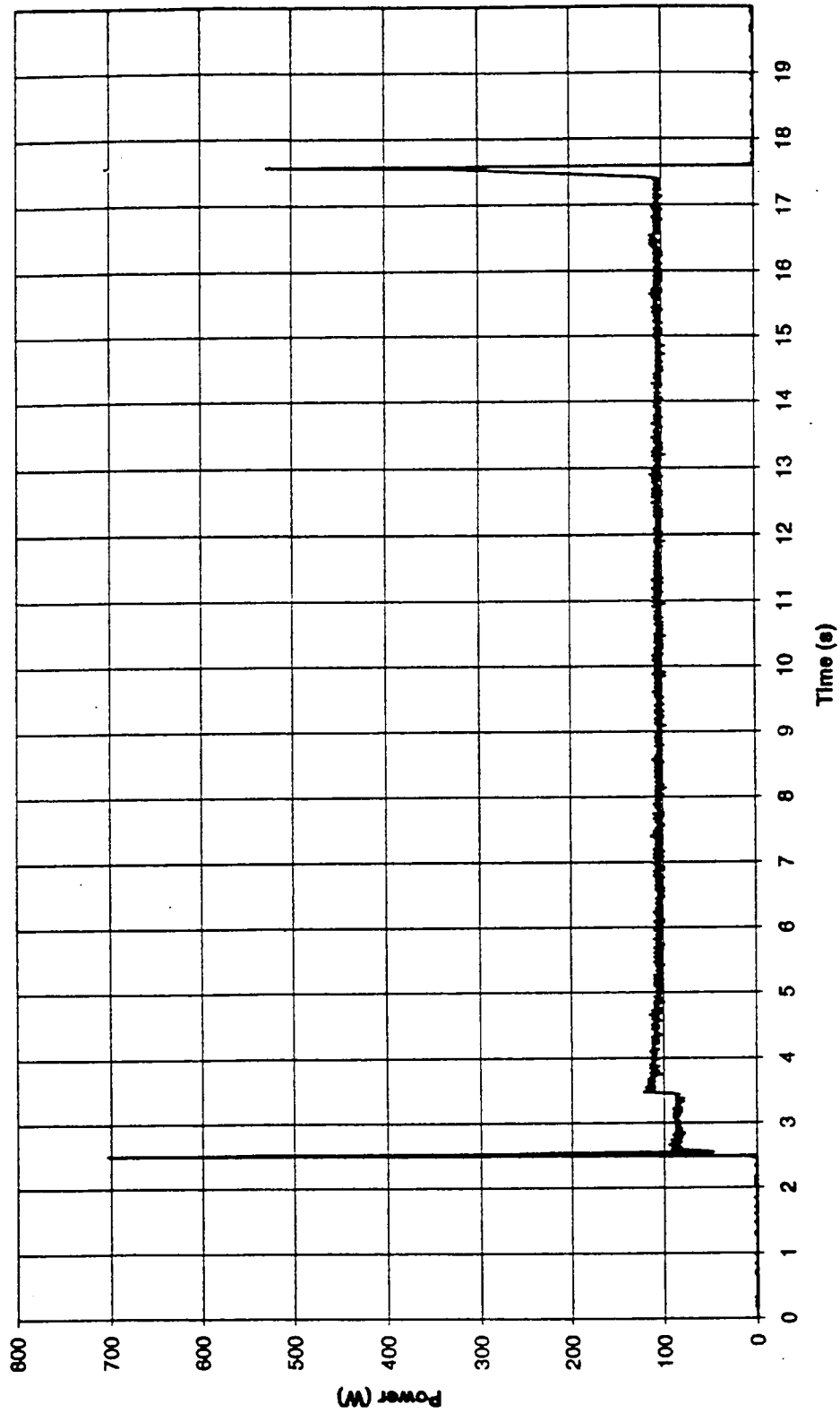


Figure 6.2 Limitorque 1-inch globe valve: baseline - open-to-close stroke.

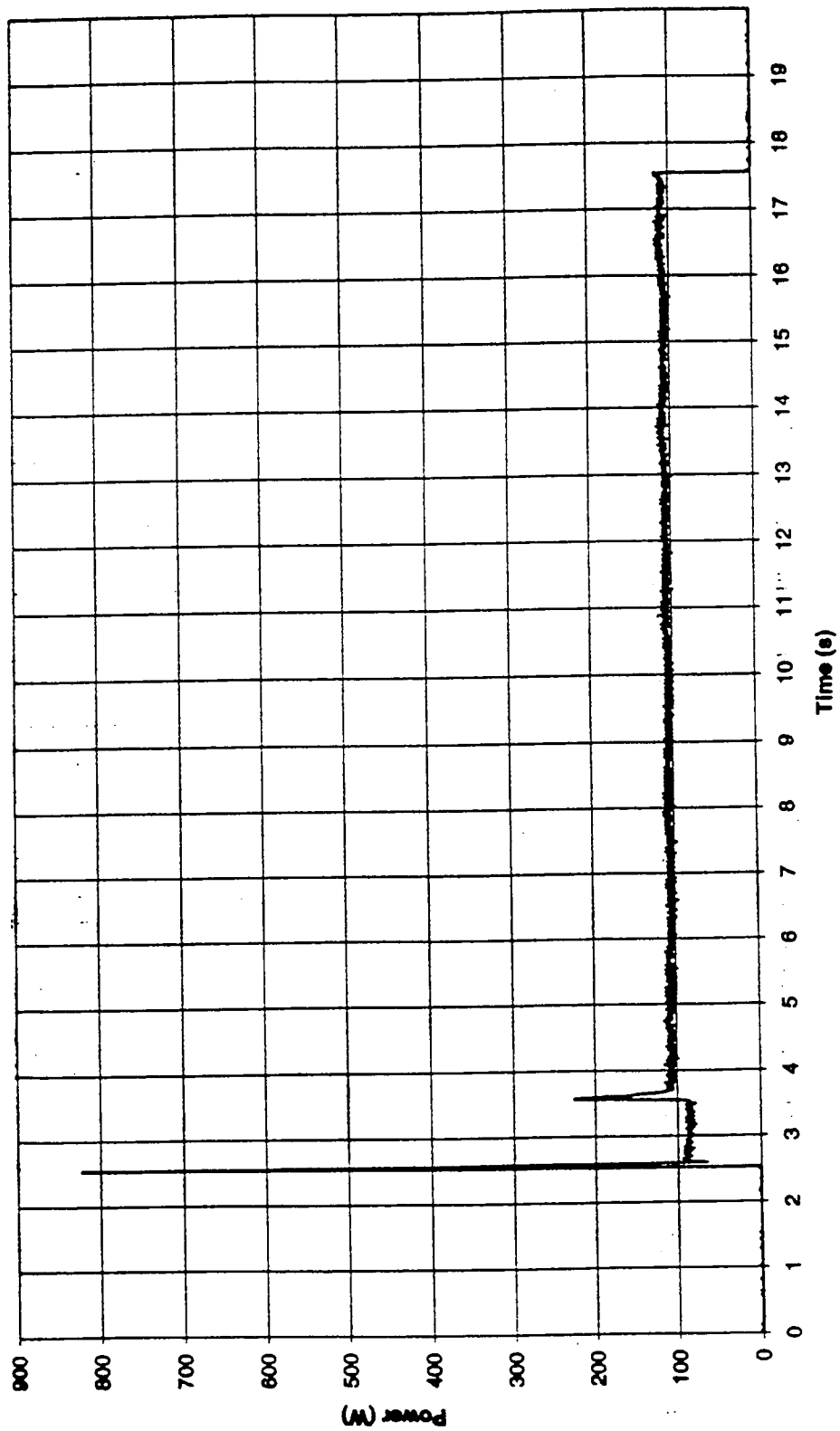


Figure 6.3 Limitorque 1-inch globe valve: baseline - close-to-open stroke.

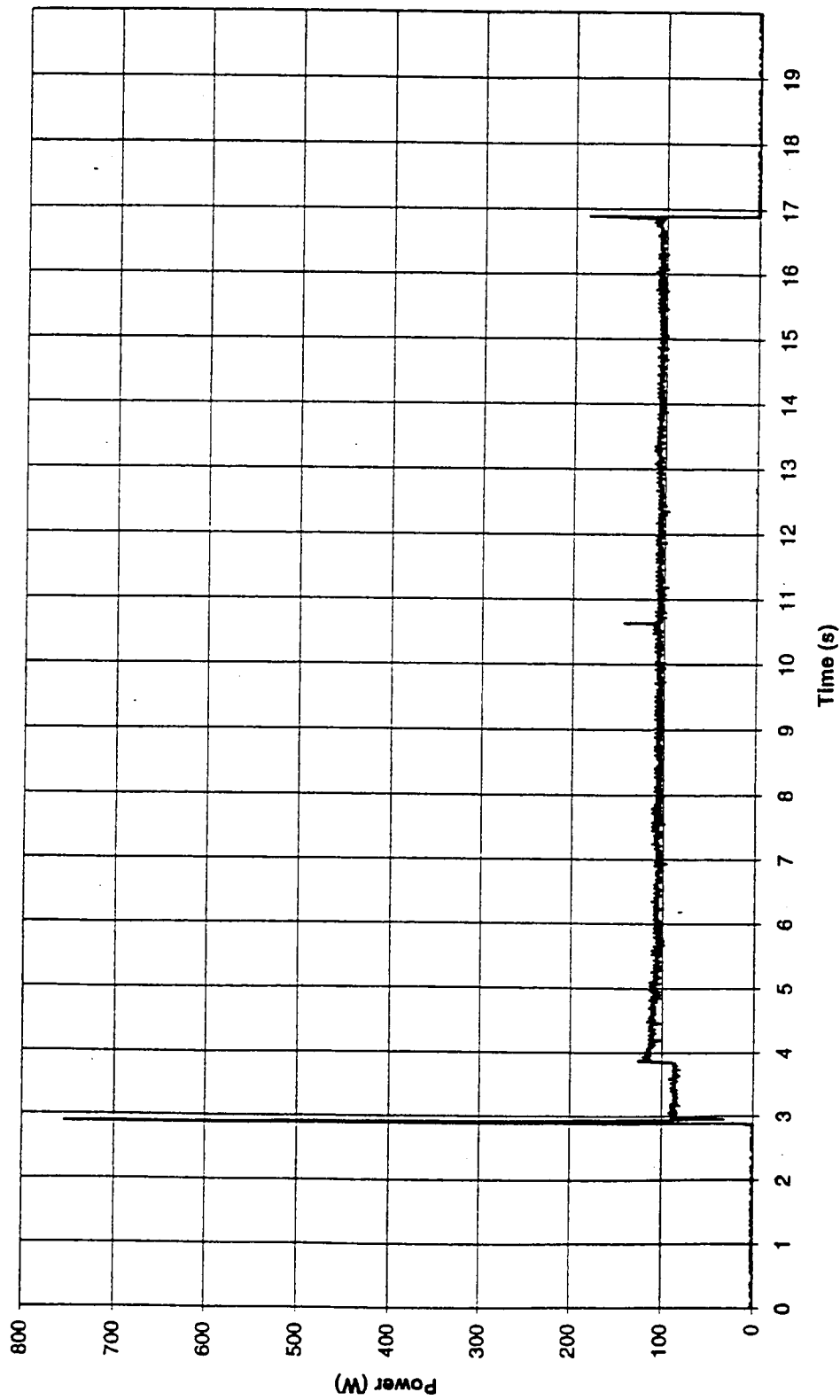


Figure 6.4 Limitorque 1-inch globe valve: close on limit switch - OC.

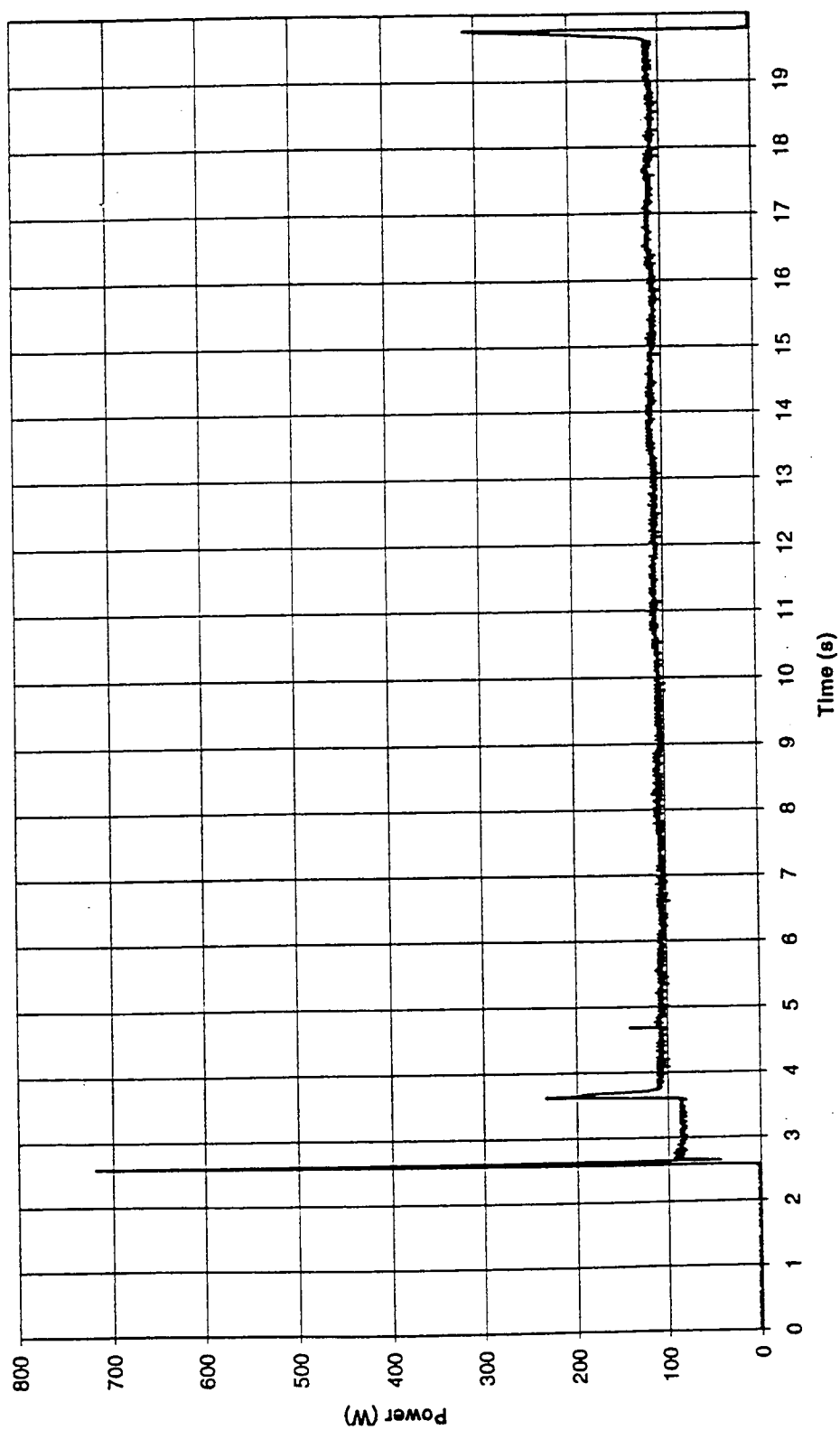


Figure 6.5 Limitorque 1-inch globe valve: open on torque switch - CO.

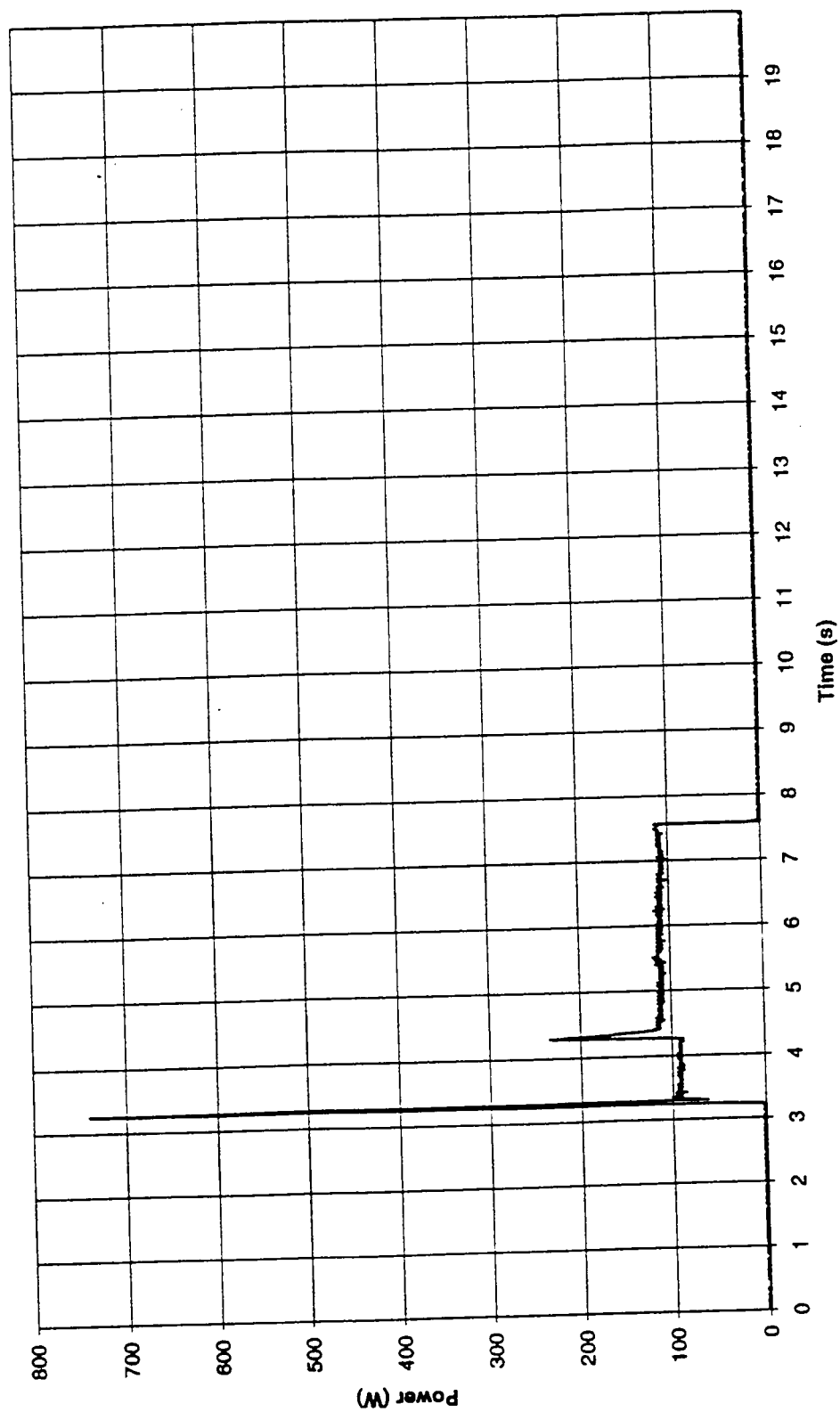


Figure 6.6 Limitorque 1-inch globe valve: misadjusted limit switch - CO.

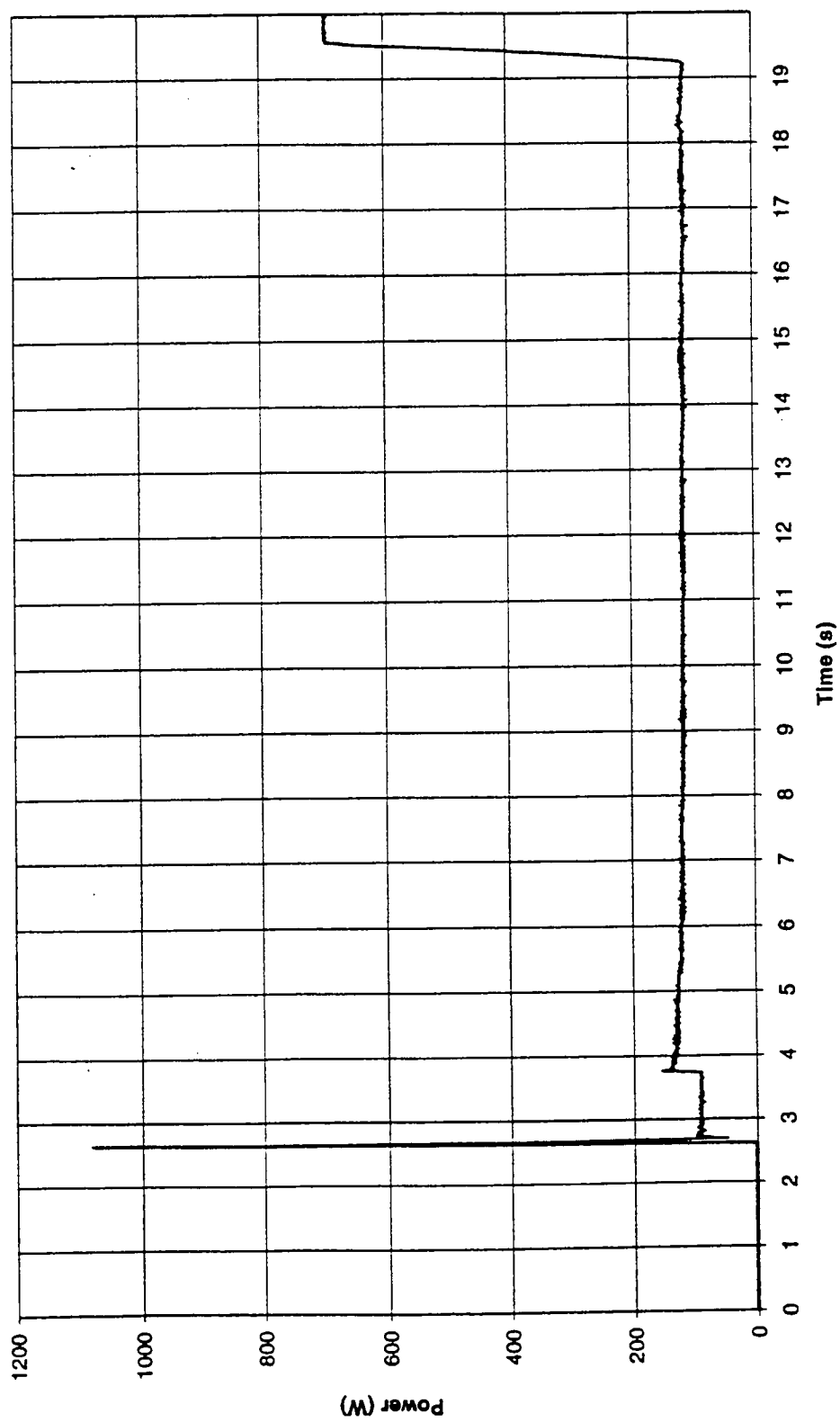


Figure 6.7 Limitorque 1-inch globe valve: misadjusted limit switch - OC.

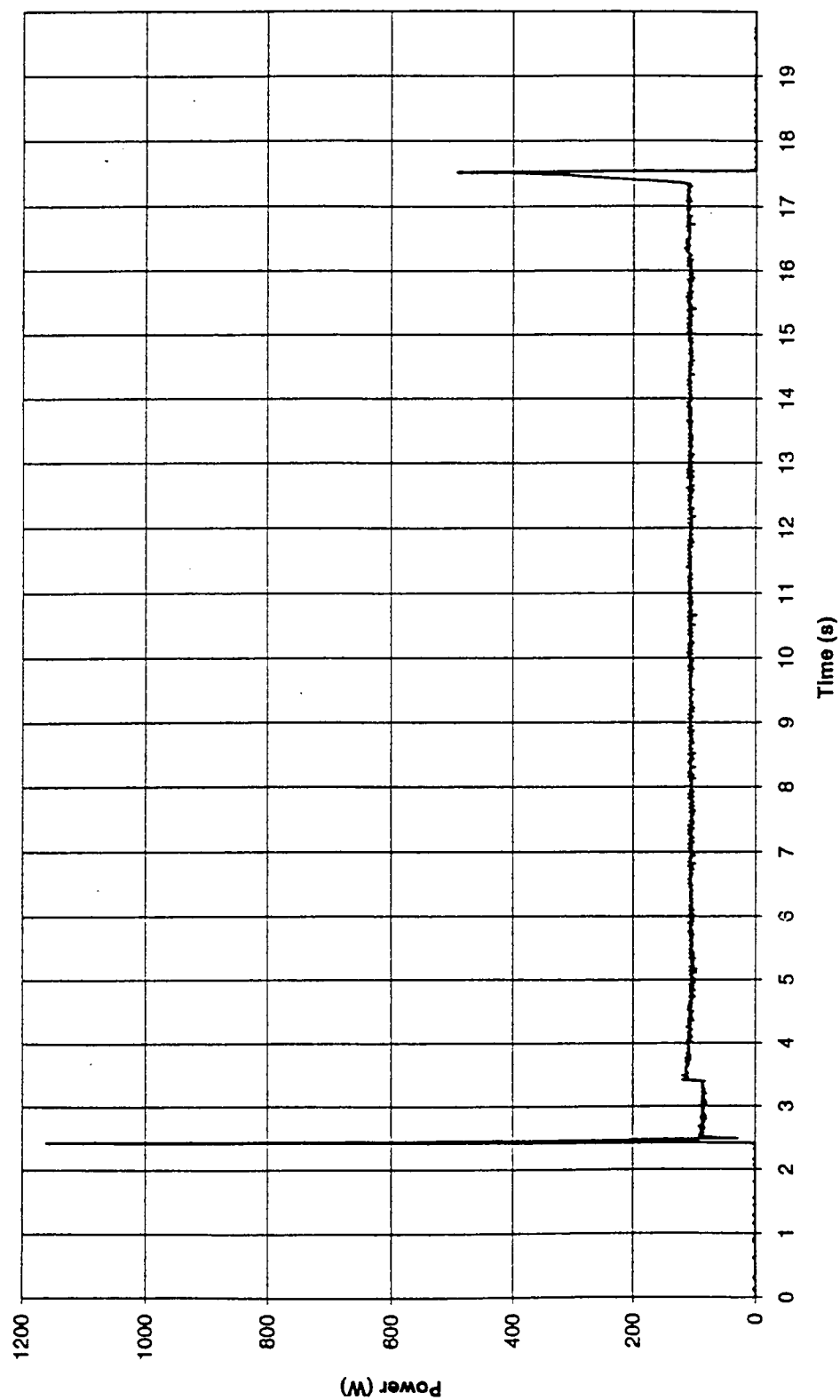


Figure 6.8 Limitorque 1-inch globe valve: torque switch set on 1 - OC.

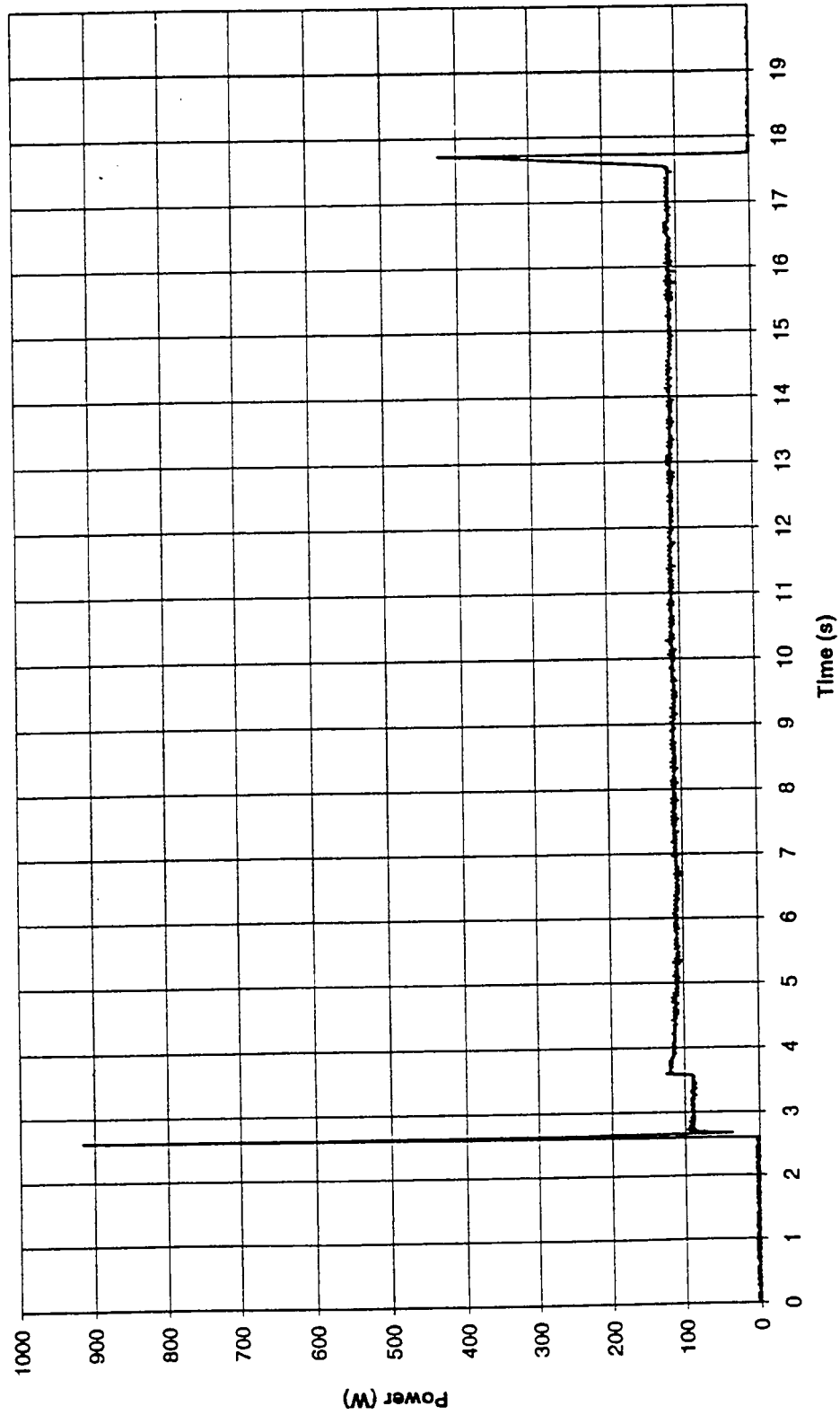


Figure 6.9 Limitorque 1-inch globe valve: torque switch set on 2 - OC.

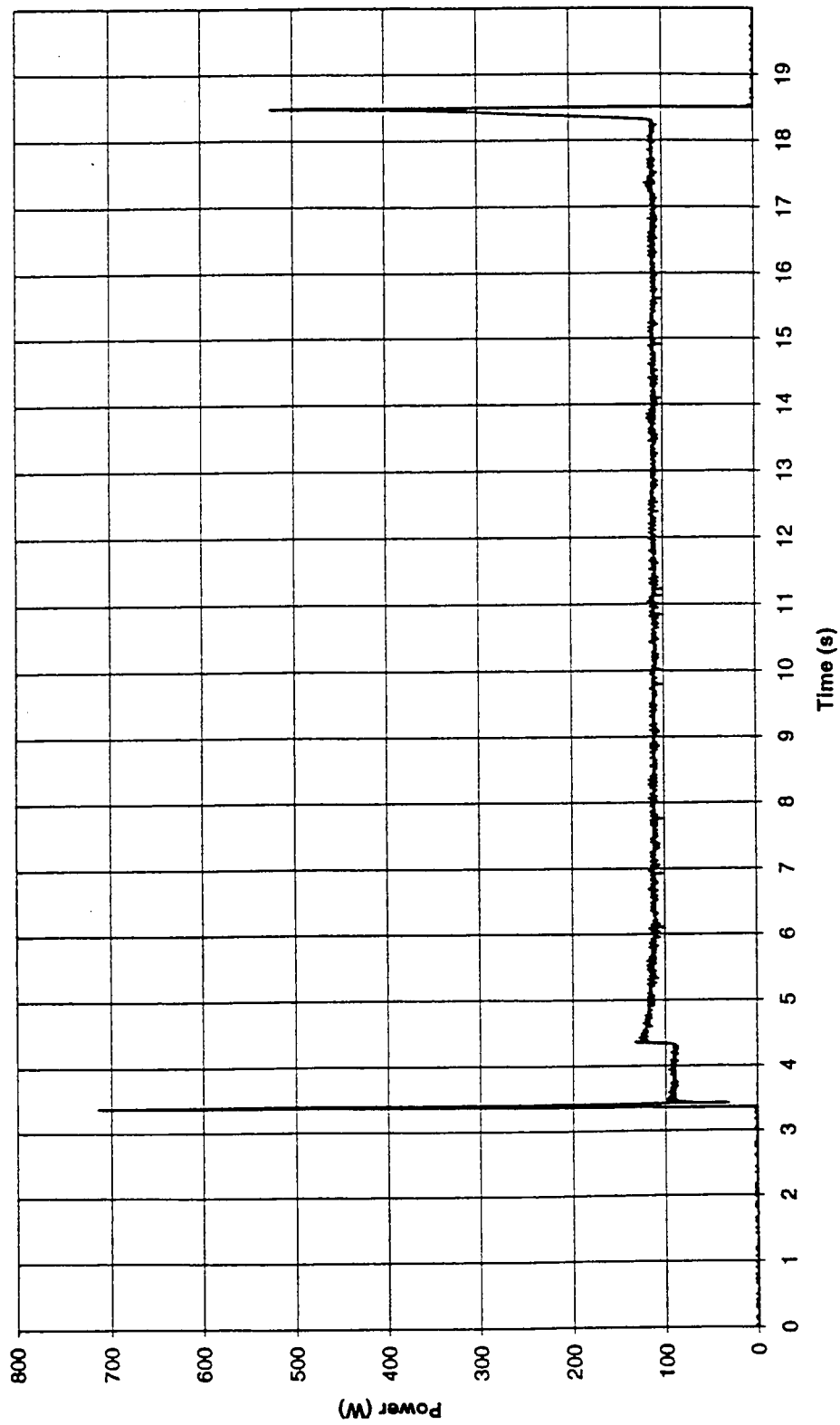


Figure 6.10 Limitorque 1-inch globe valve: torque switch set on 2.5 - OC.

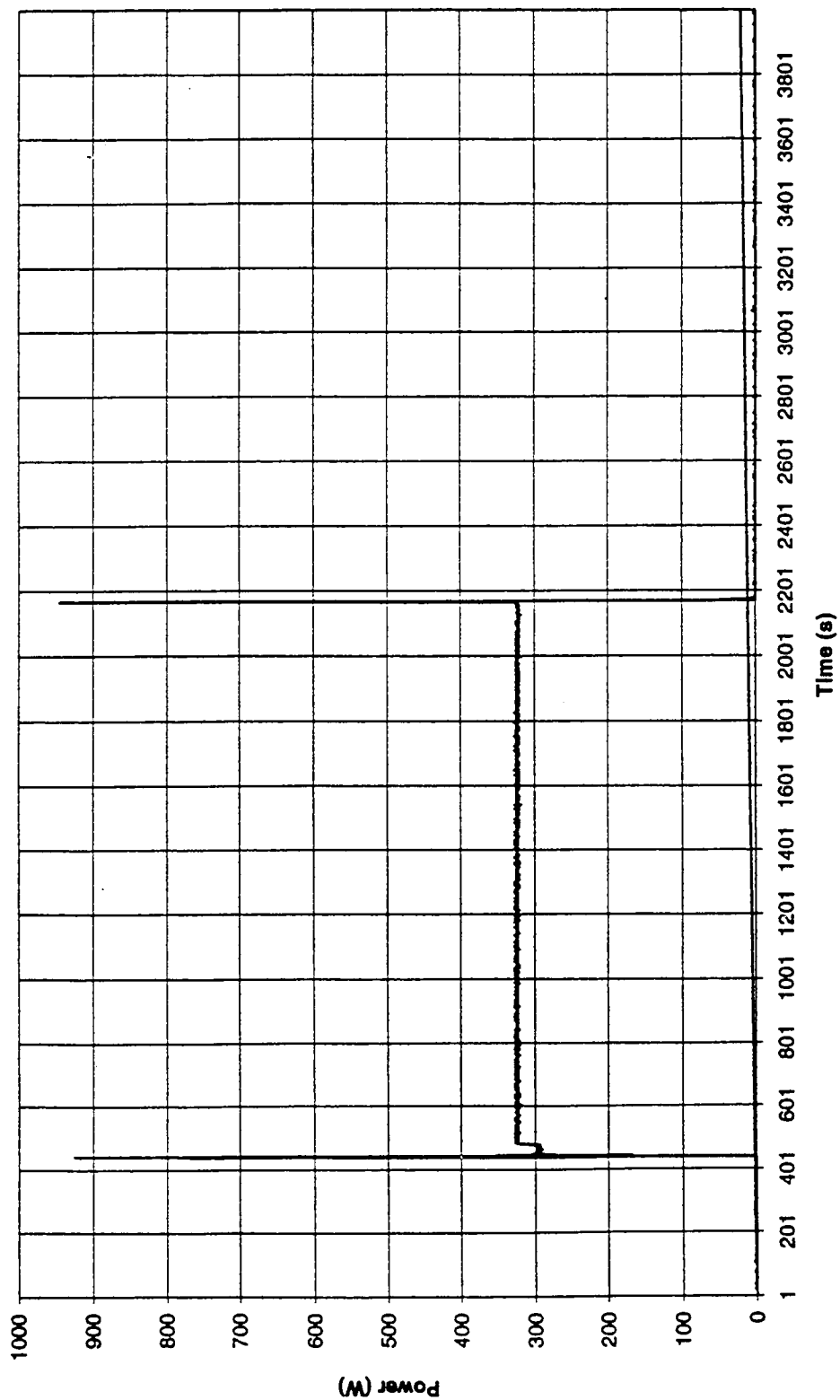


Figure 6.11 Limitorque 1-inch globe valve: single phase prior to start - CO.

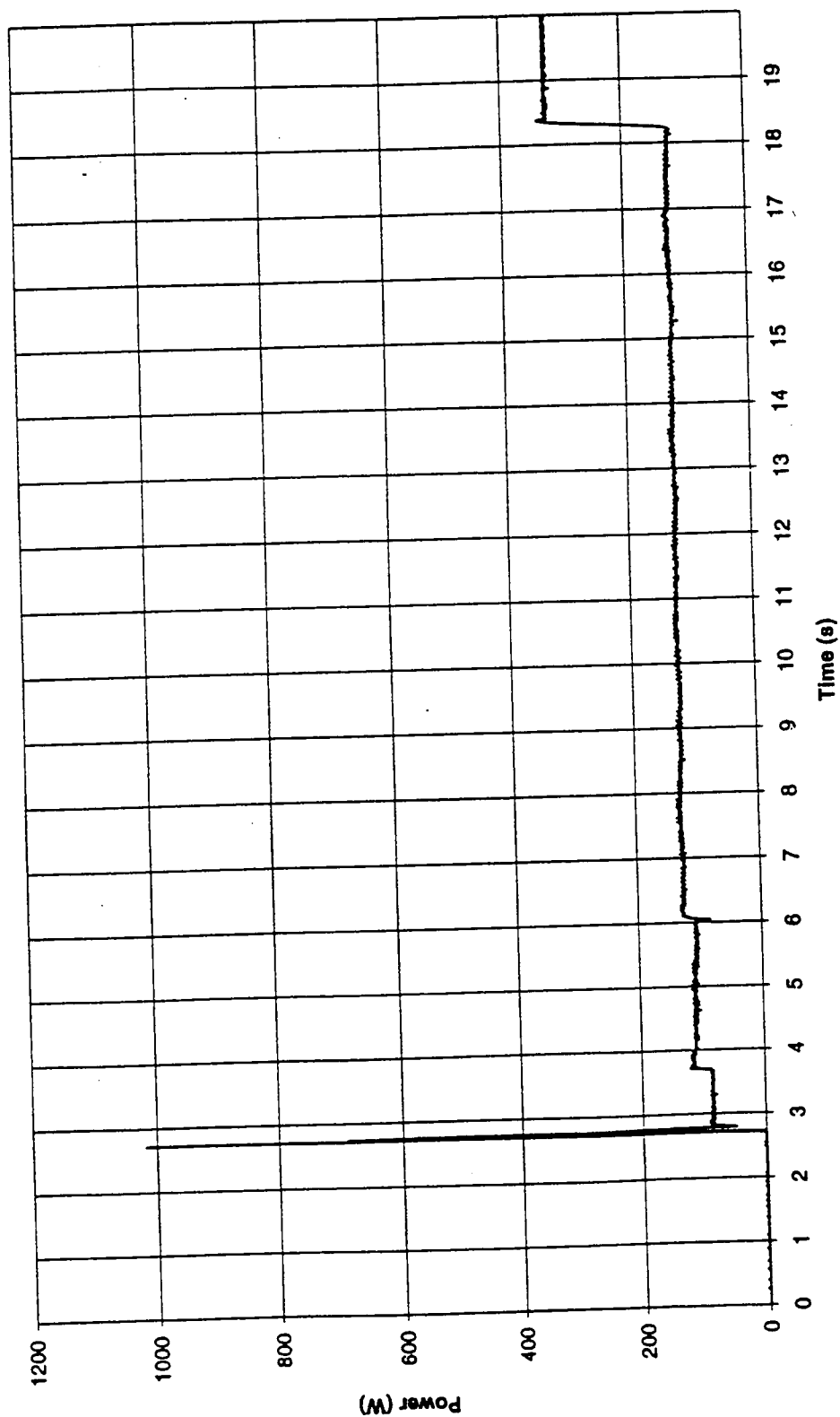


Figure 6.12 Limitorque 1-inch globe valve: single phase after start - CO.

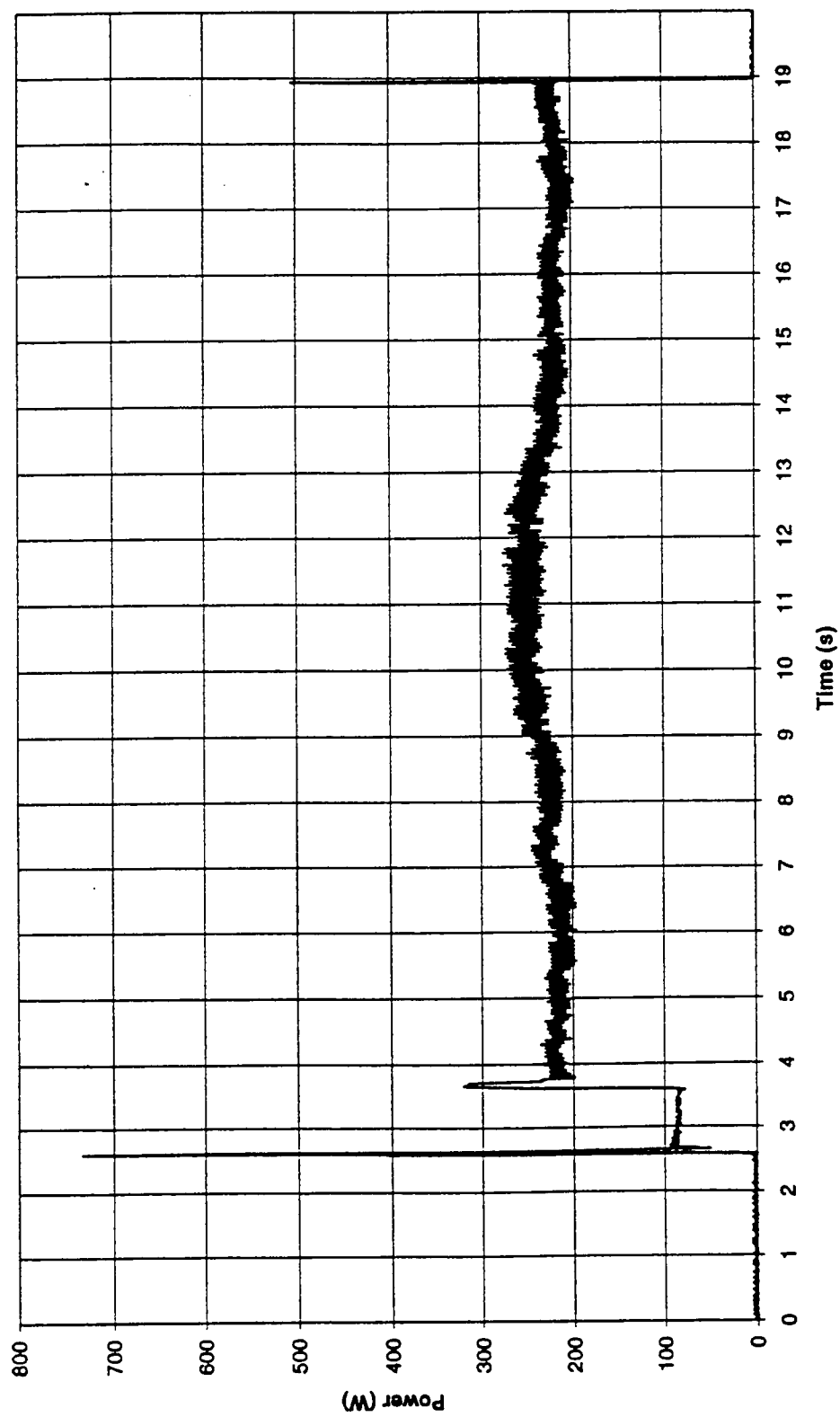


Figure 6.13 Limitorque 1-inch globe valve: tight packing - CO.

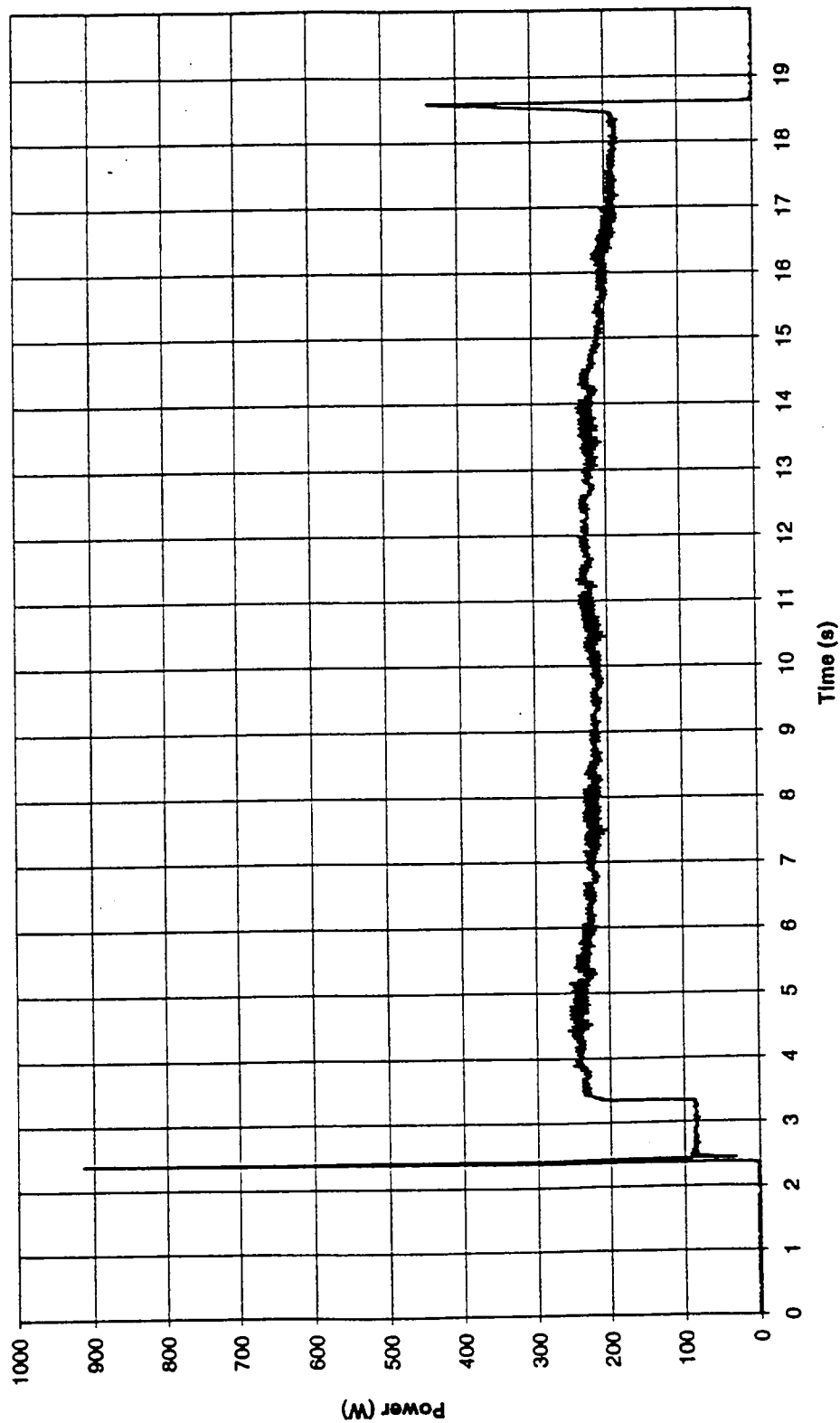


Figure 6.14 1-inch globe valve: tight packing - OC.

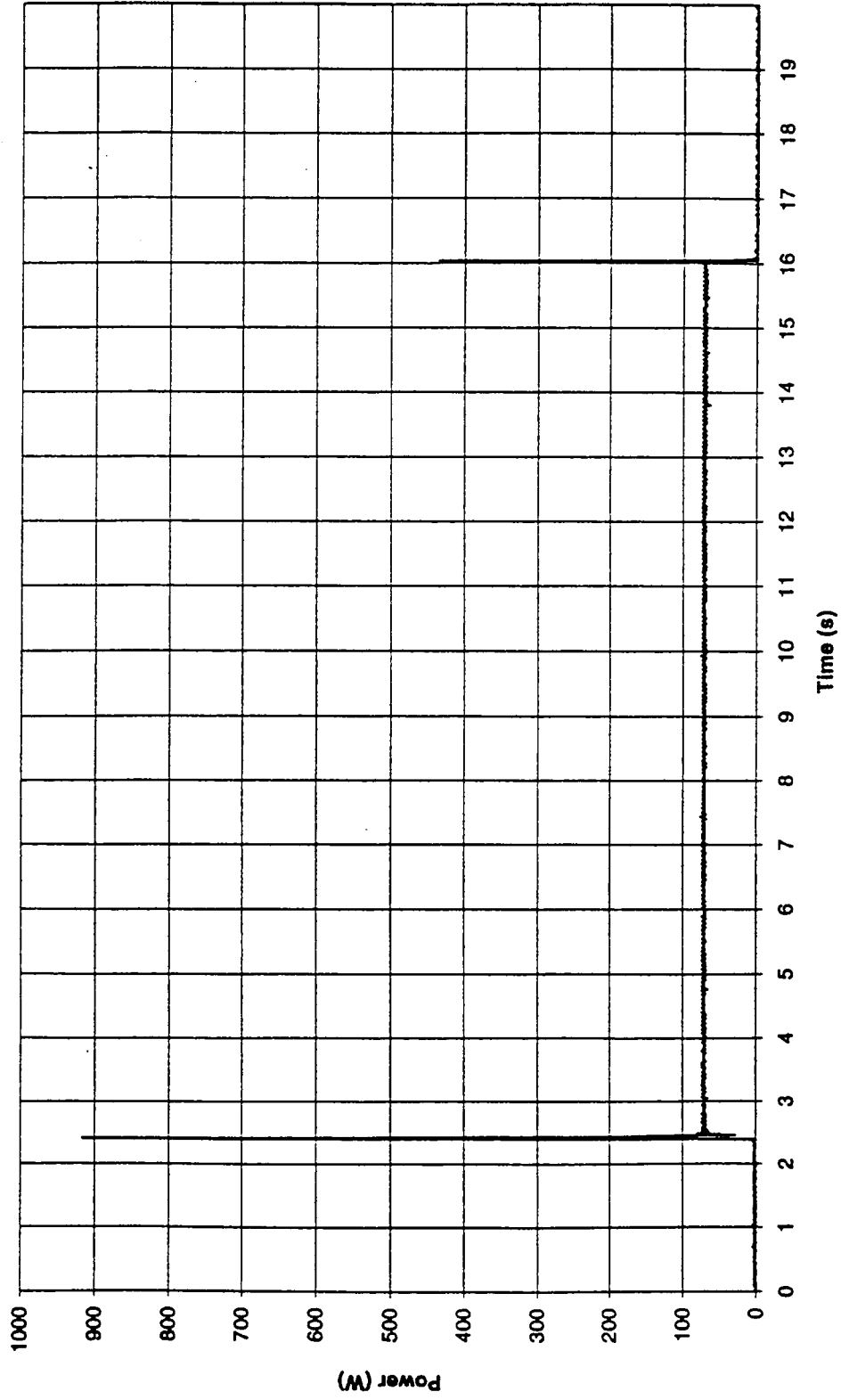


Figure 6.15 Limitorque 1-inch globe valve: motor pinion disconnected.

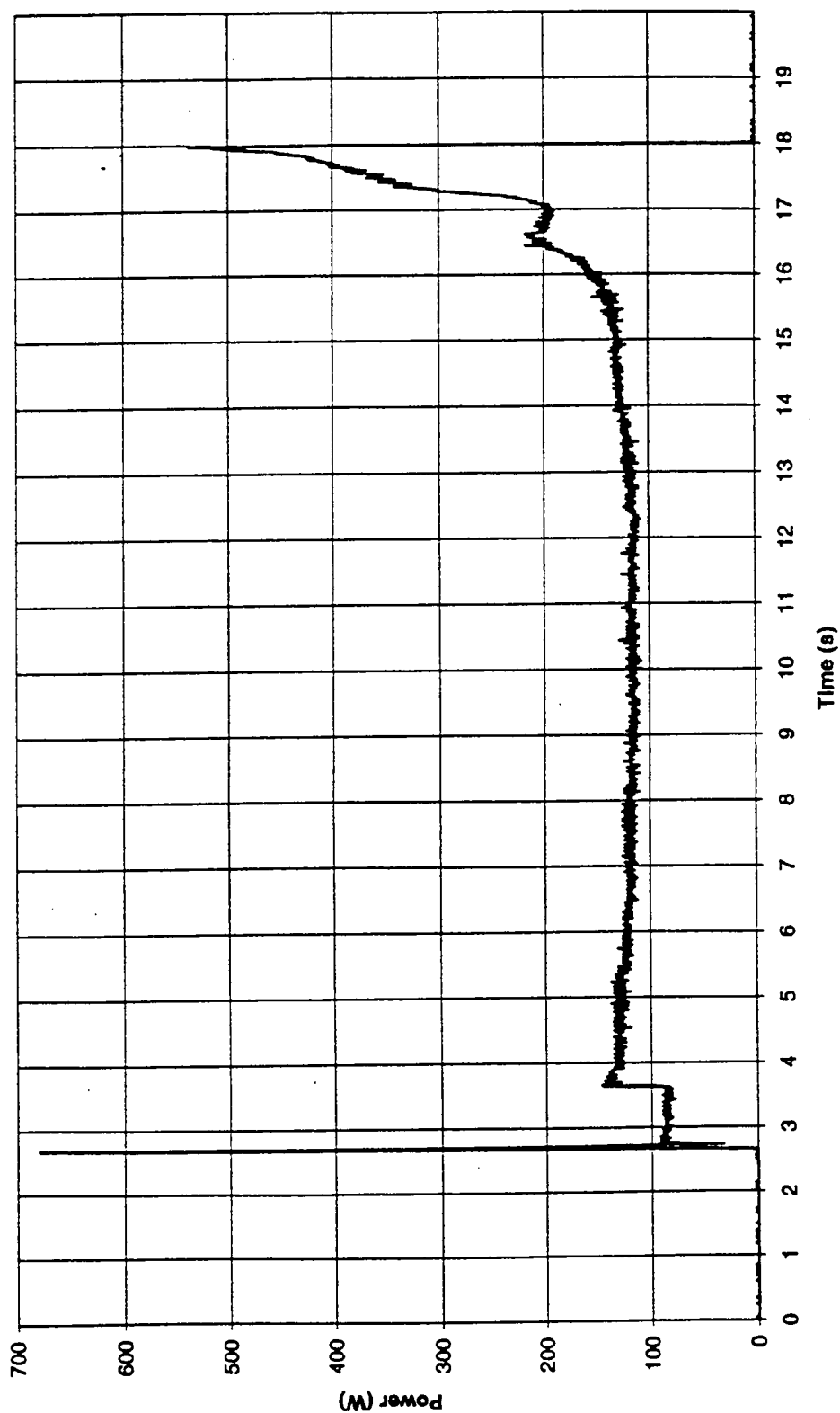


Figure 6.16 Limitorque 1-inch globe valve: obstruction in valve seat .

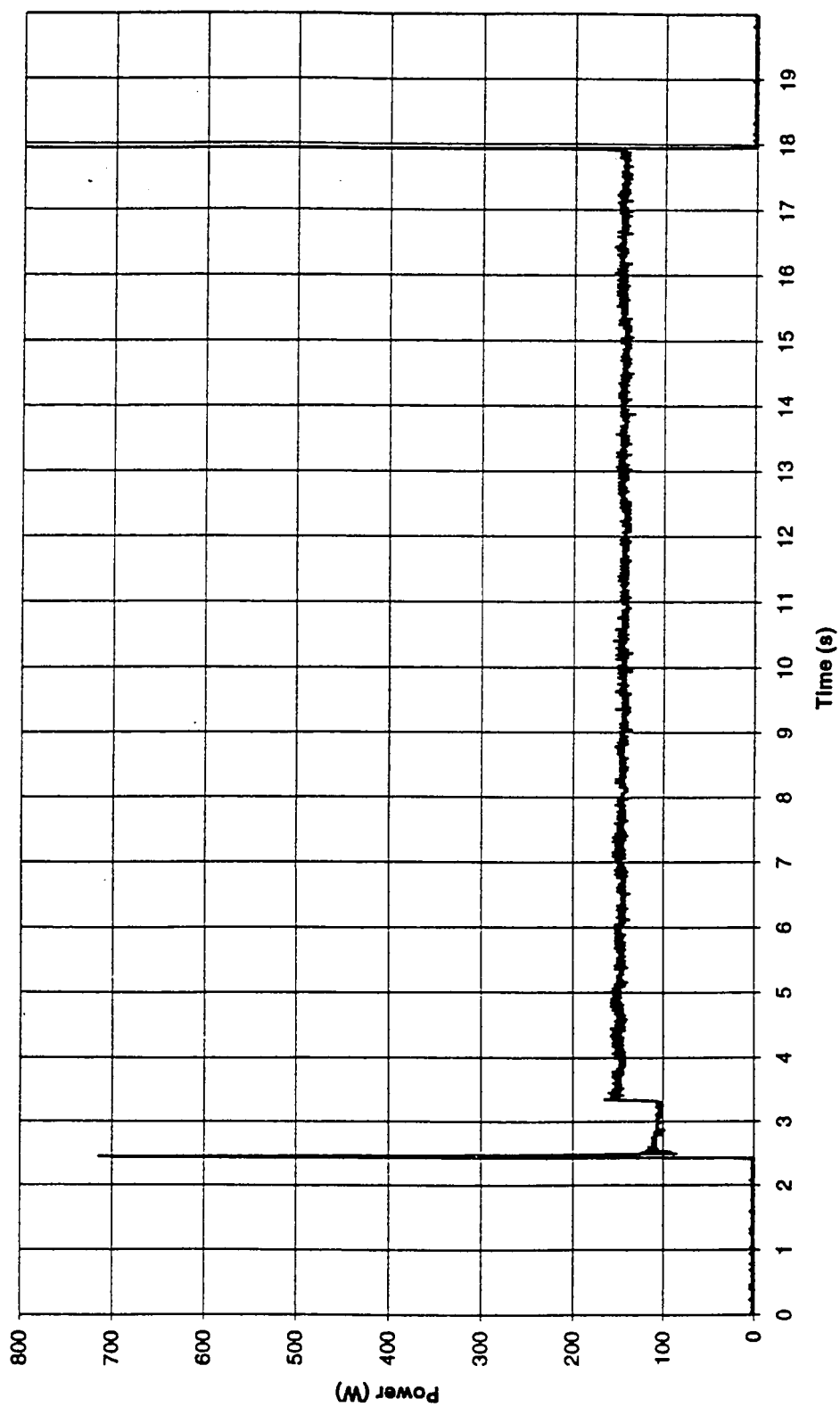


Figure 6.17 Limitorque 1-inch globe valve: washer removed from the spring pack - OC.

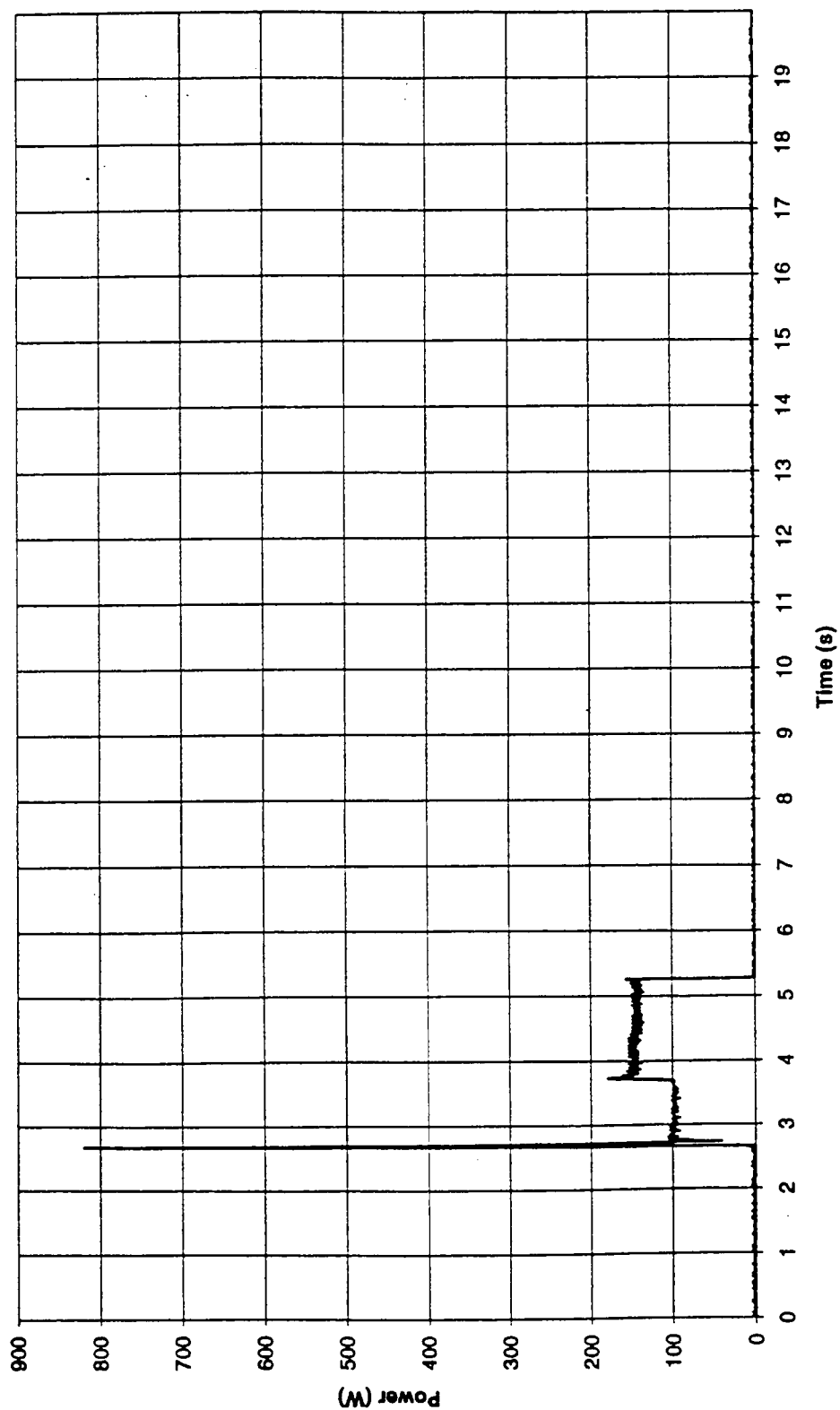


Figure 6.18 Limitorque 1-inch globe valve: washer removed from the spring pack - CO.

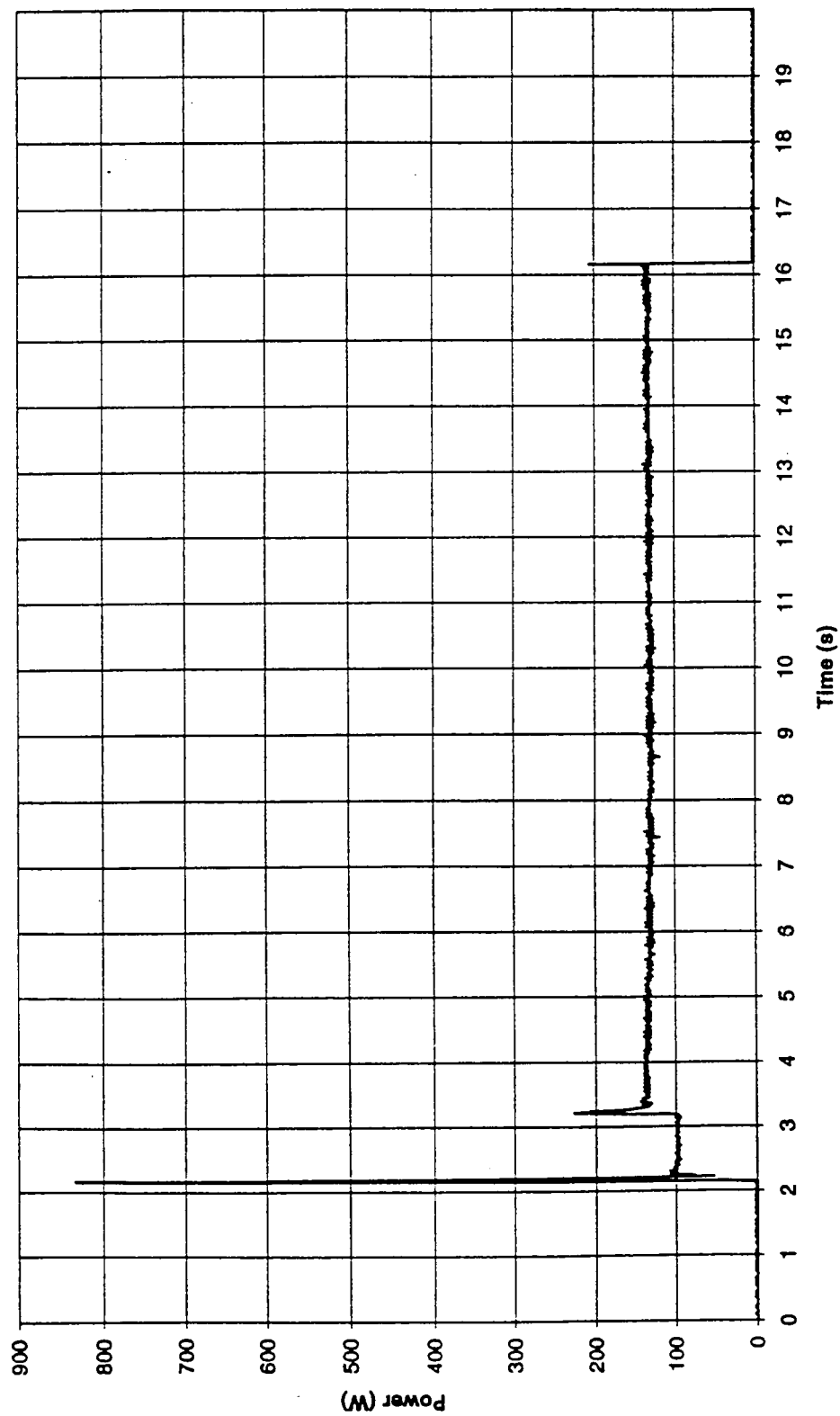


Figure 6.19 Limitorque 1-inch globe valve: normal voltage - CO.

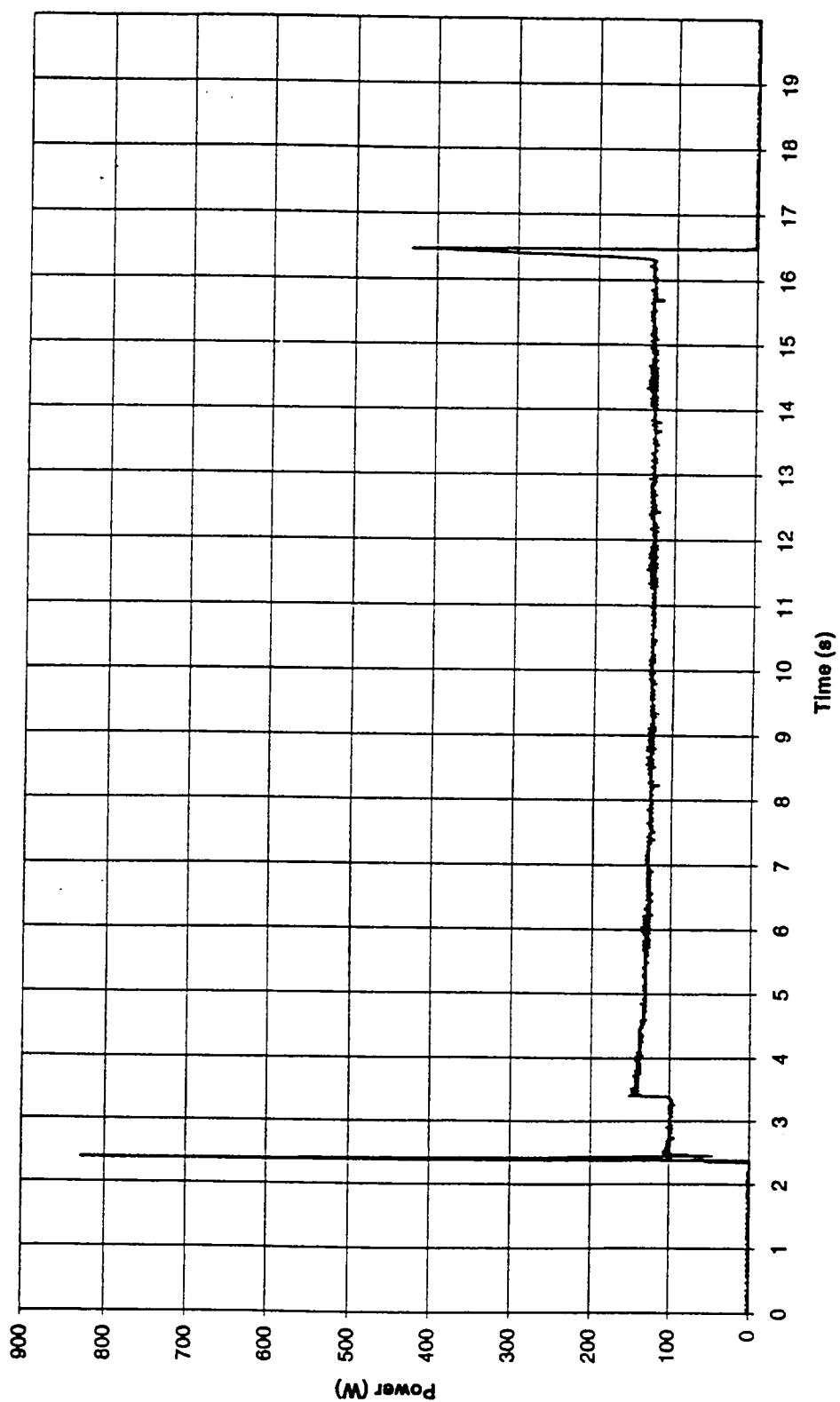


Figure 6.20 Limitorque 1-inch globe valve: normal voltage - OC.

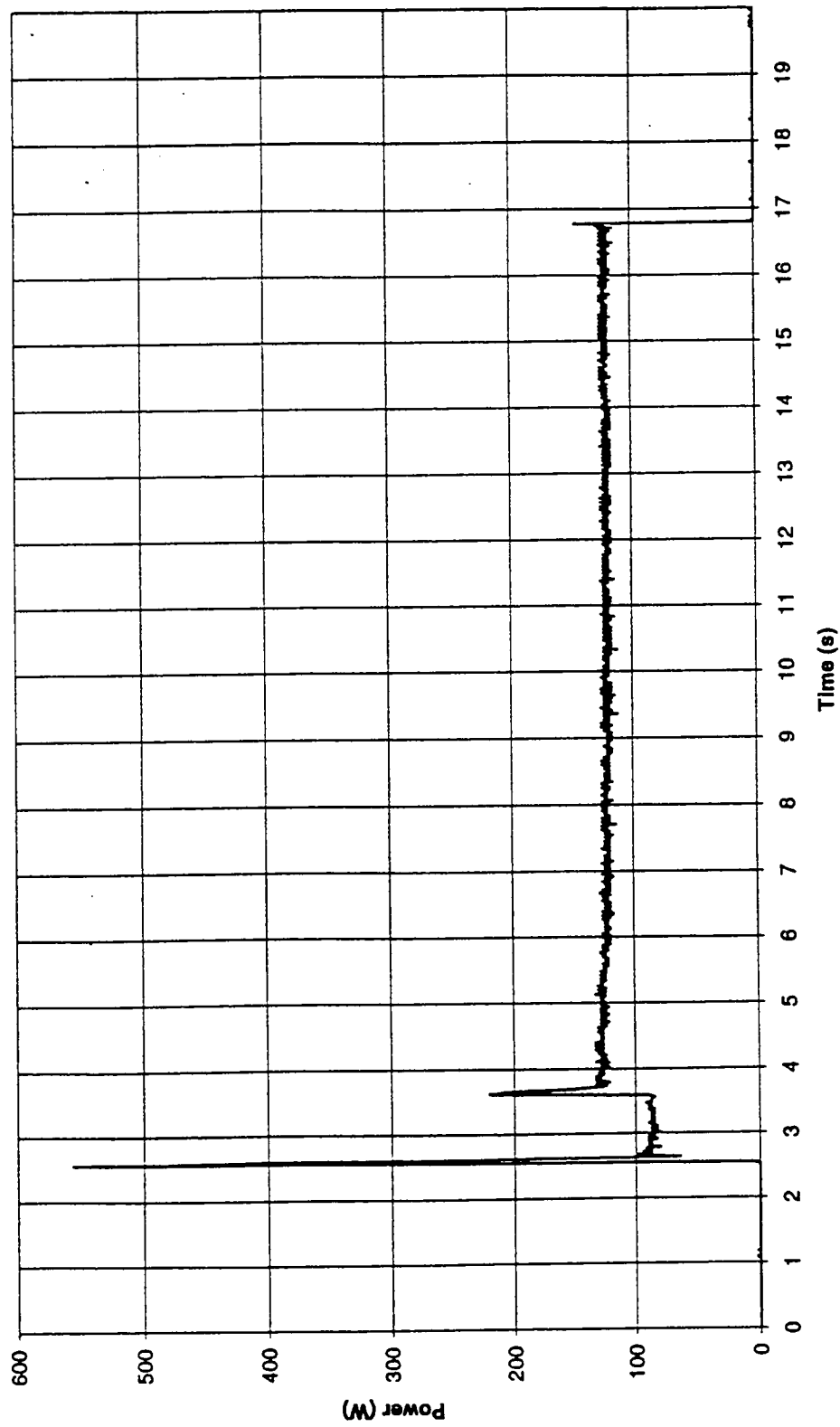


Figure 6.21 Limitorque 1-inch globe valve: Voltage reduced by 10% - CO.

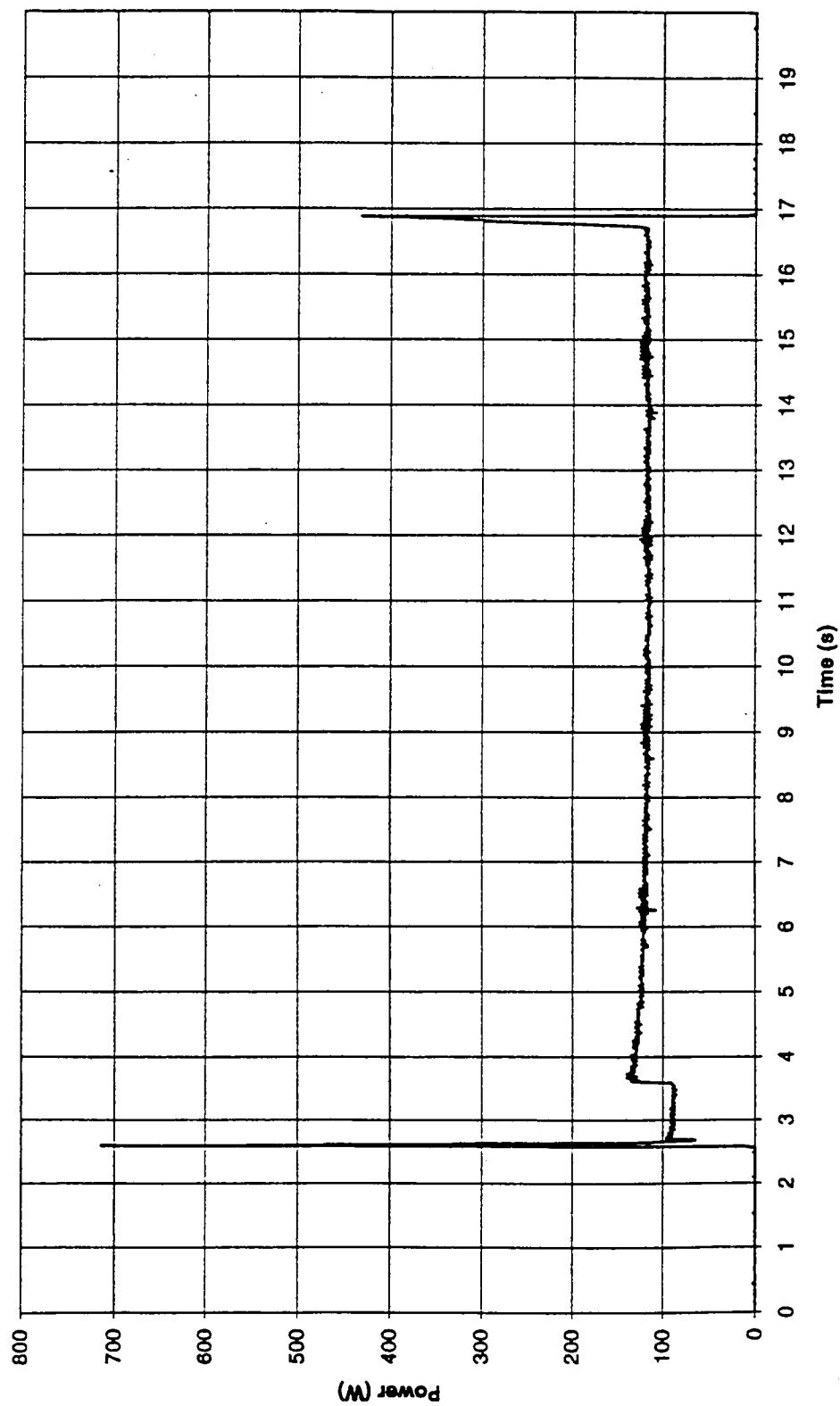


Figure 6.22 Limitorque 1-inch globe valve: Voltage reduced by 10% - OC.

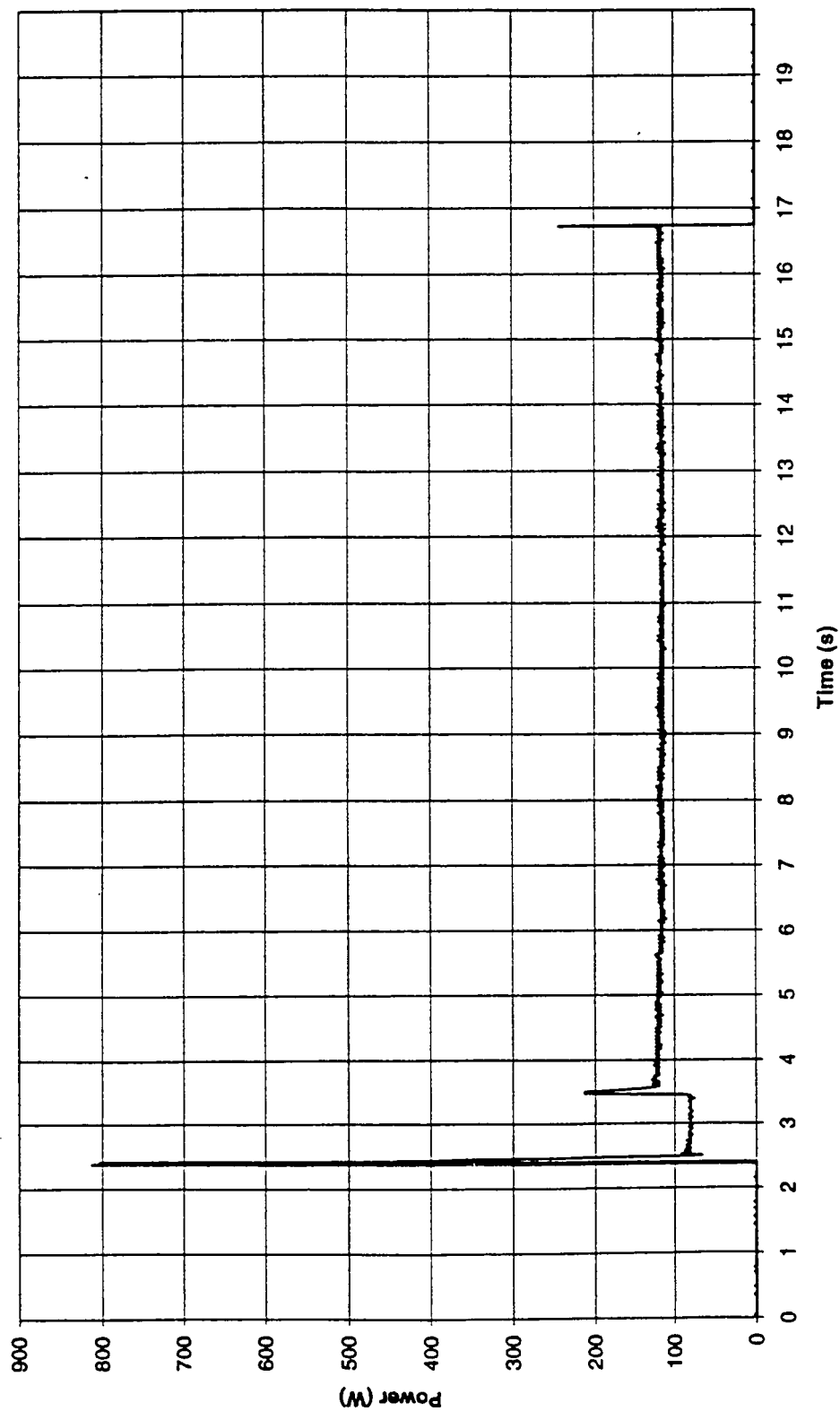


Figure 6.23 Limitorque 1-inch globe valve: voltage reduced by 15% - CO.

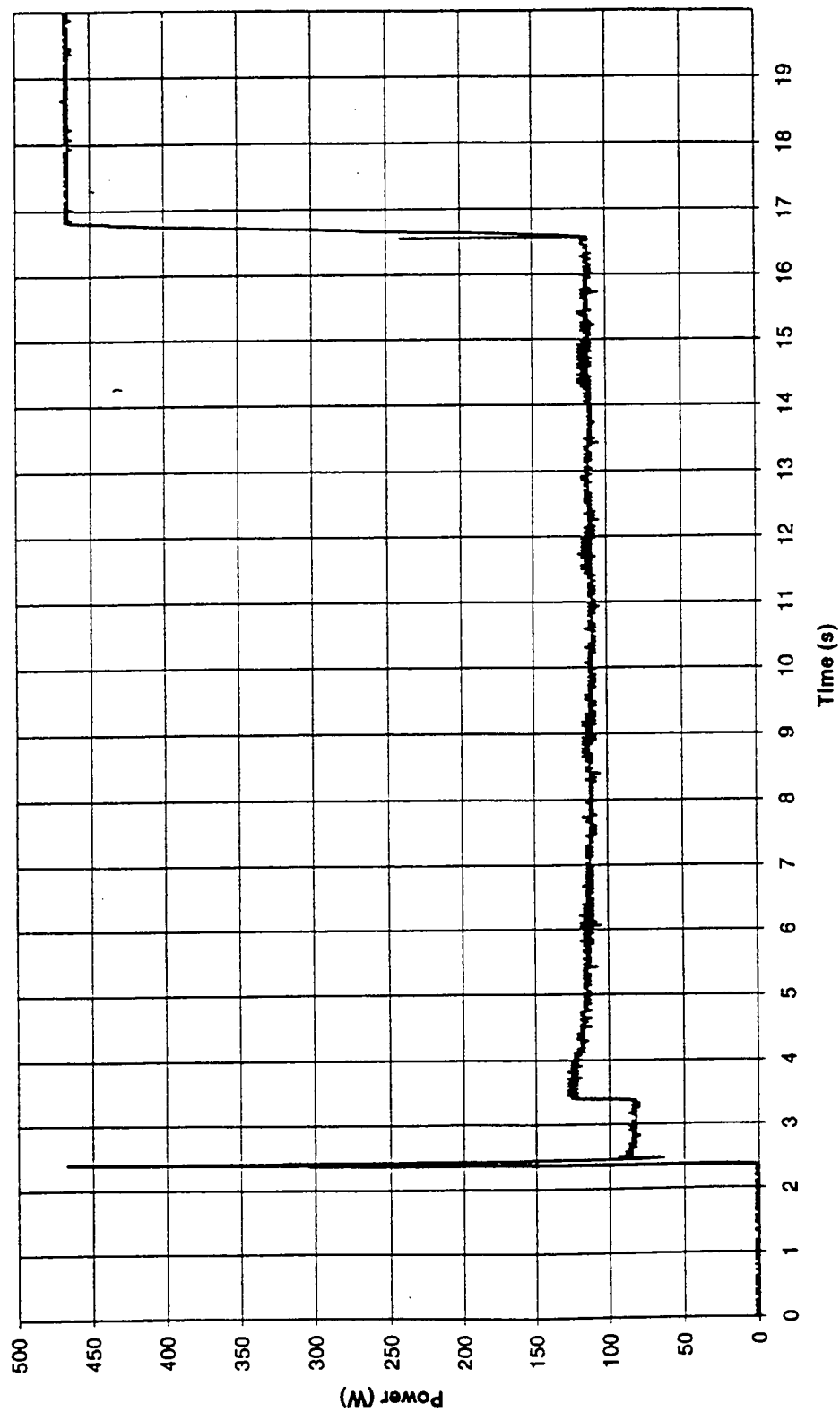


Figure 6.24 Limitorque 1-inch globe valve: voltage reduced by 15% - OC.

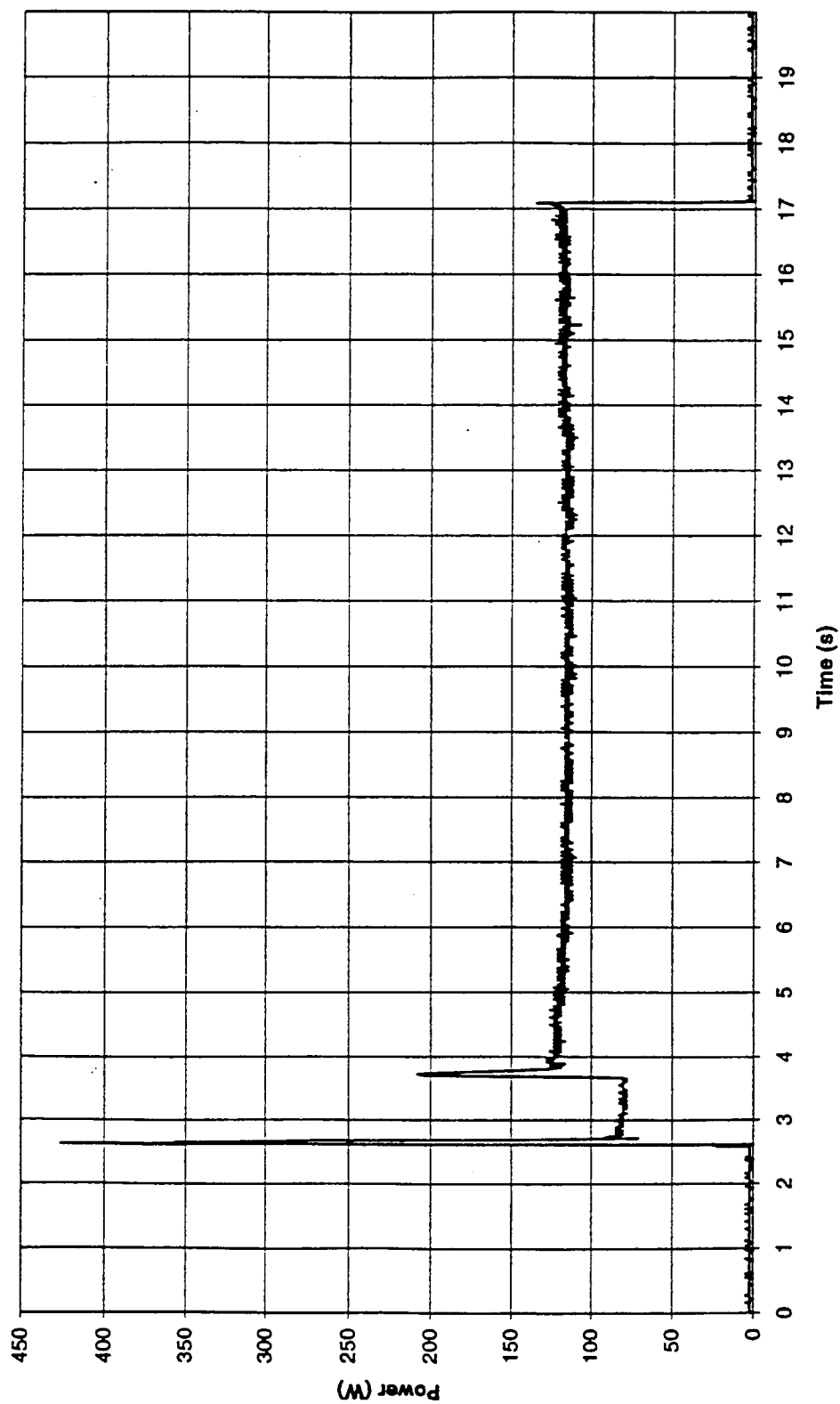


Figure 6.25 Limitorque 1-inch globe valve: voltage reduced by 20% - CO.

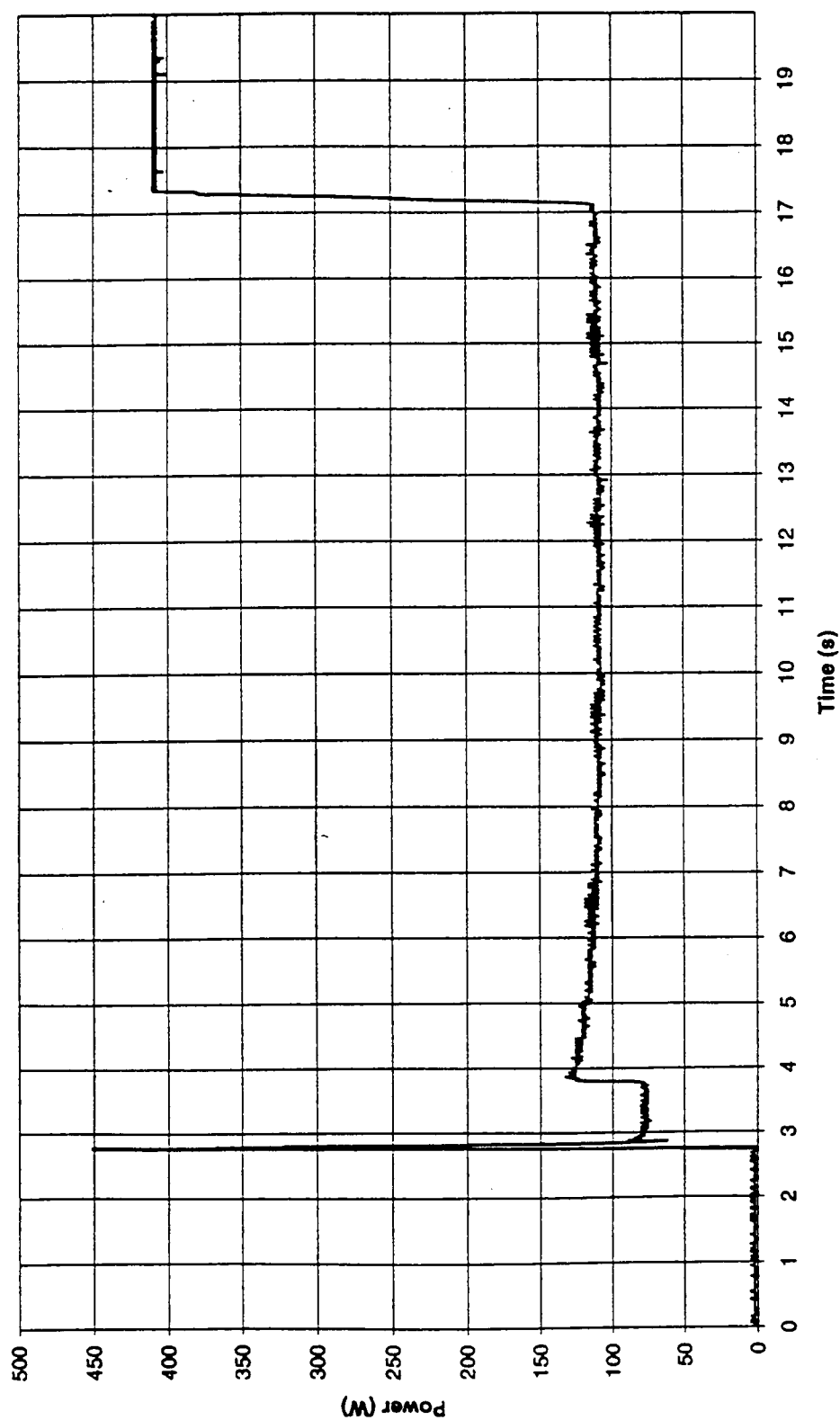


Figure 6.26 Limitorque 1-inch globe valve: voltage reduced by 20% - CO.

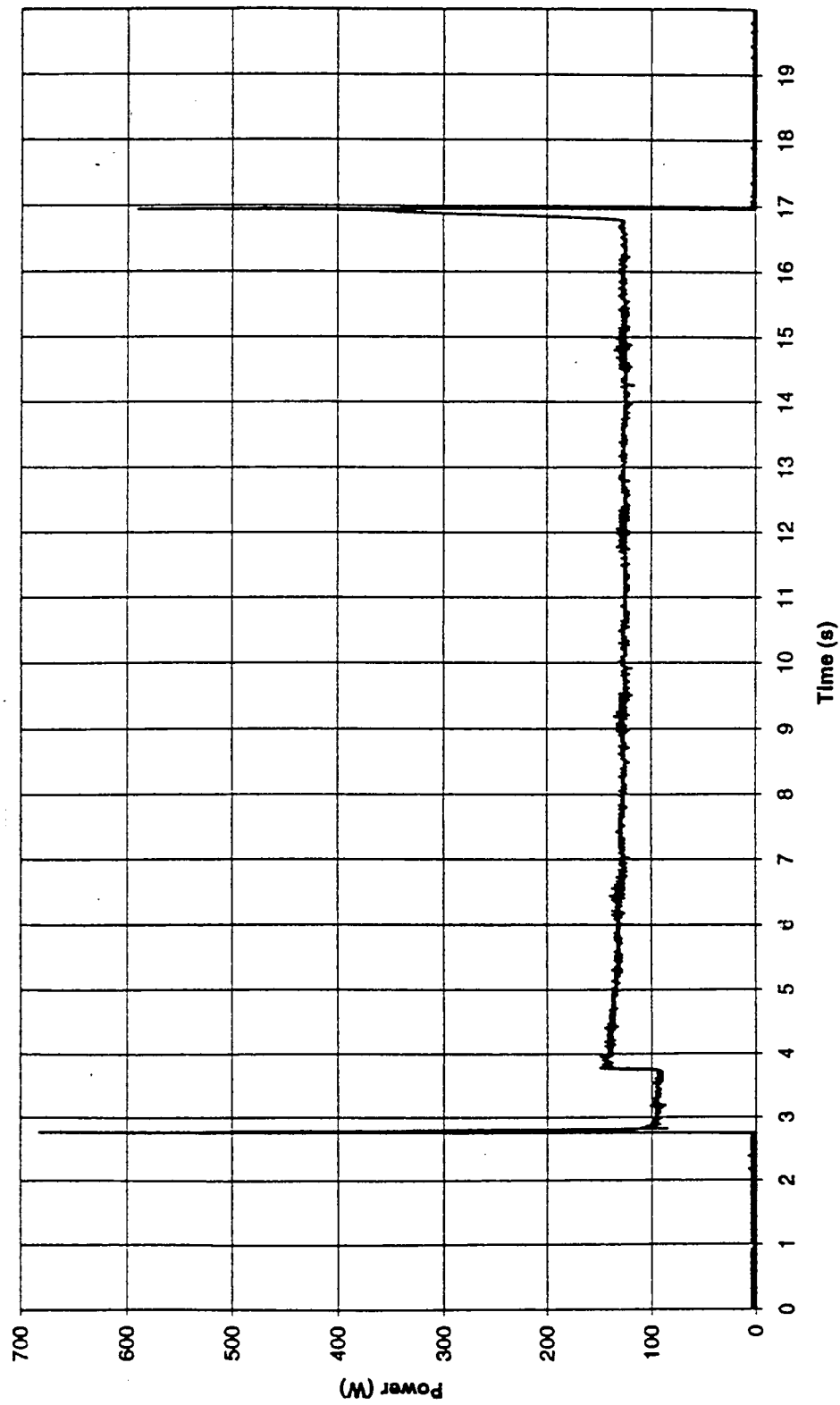


Figure 6.27 Limitorque 1-inch globe valve: unbalanced phases - OC.

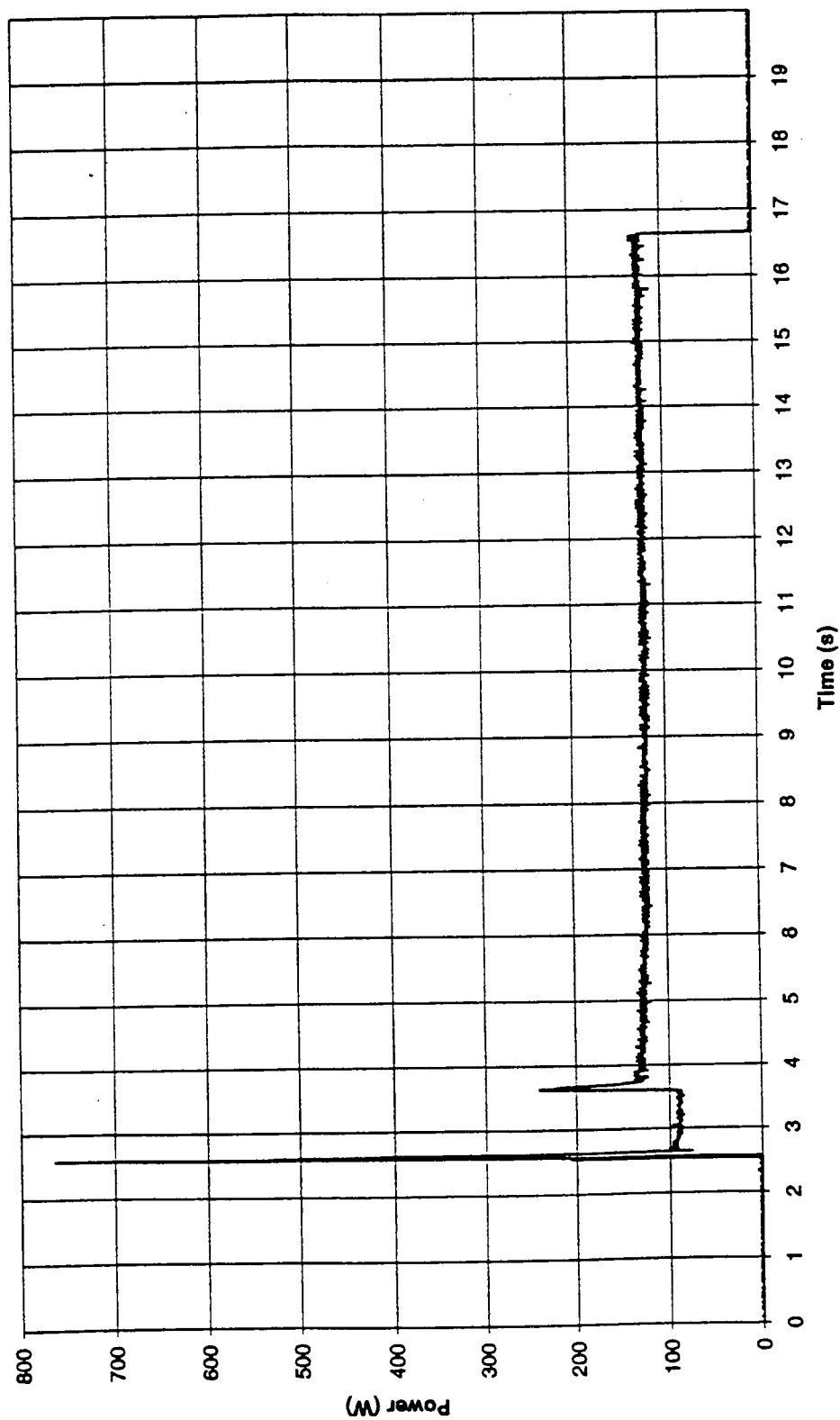


Figure 6.28 Limitorque 1-inch globe valve: unbalanced phases - CO.

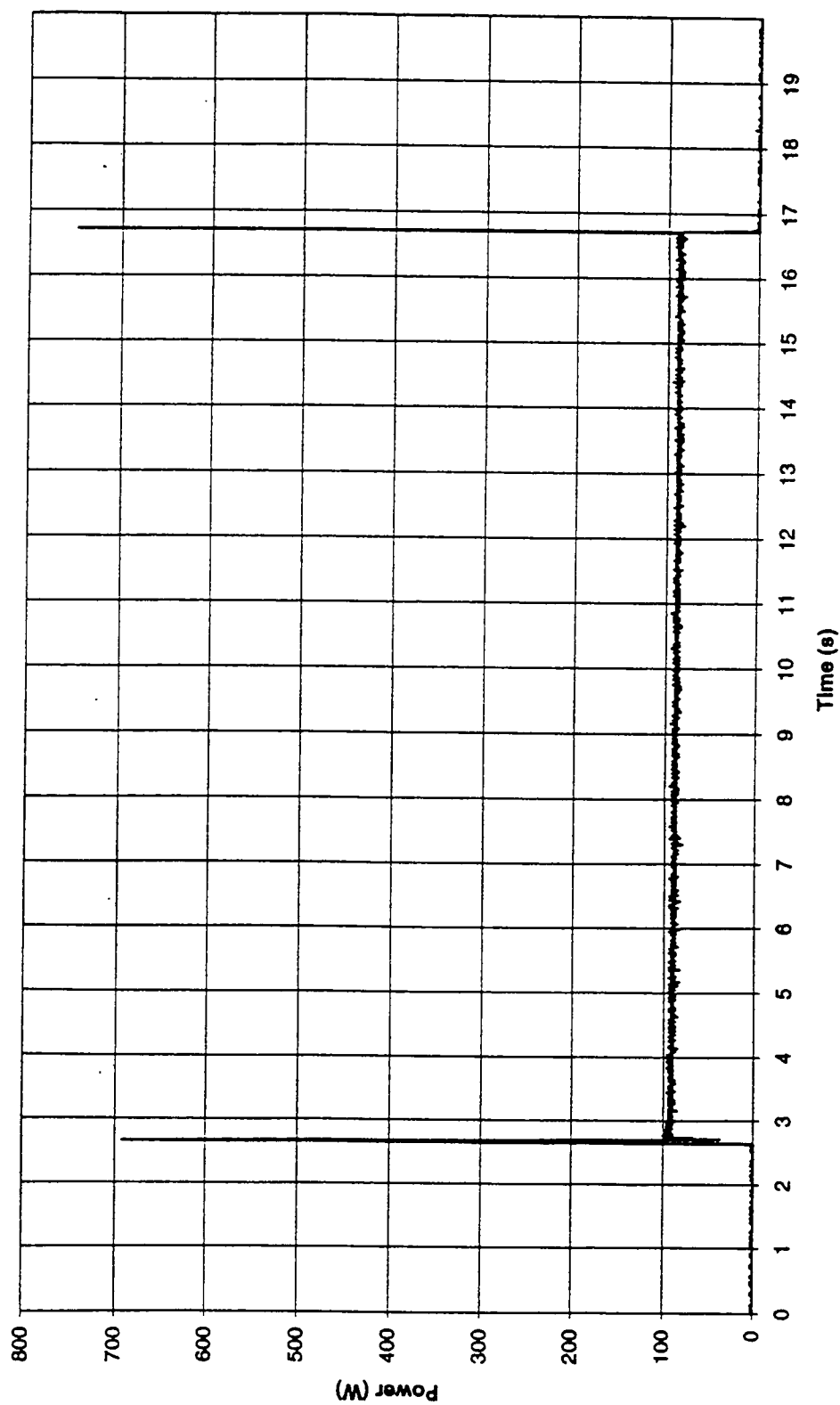


Figure 6.29 Limitorque 1-inch globe valve: declutch mechanism malfunction.

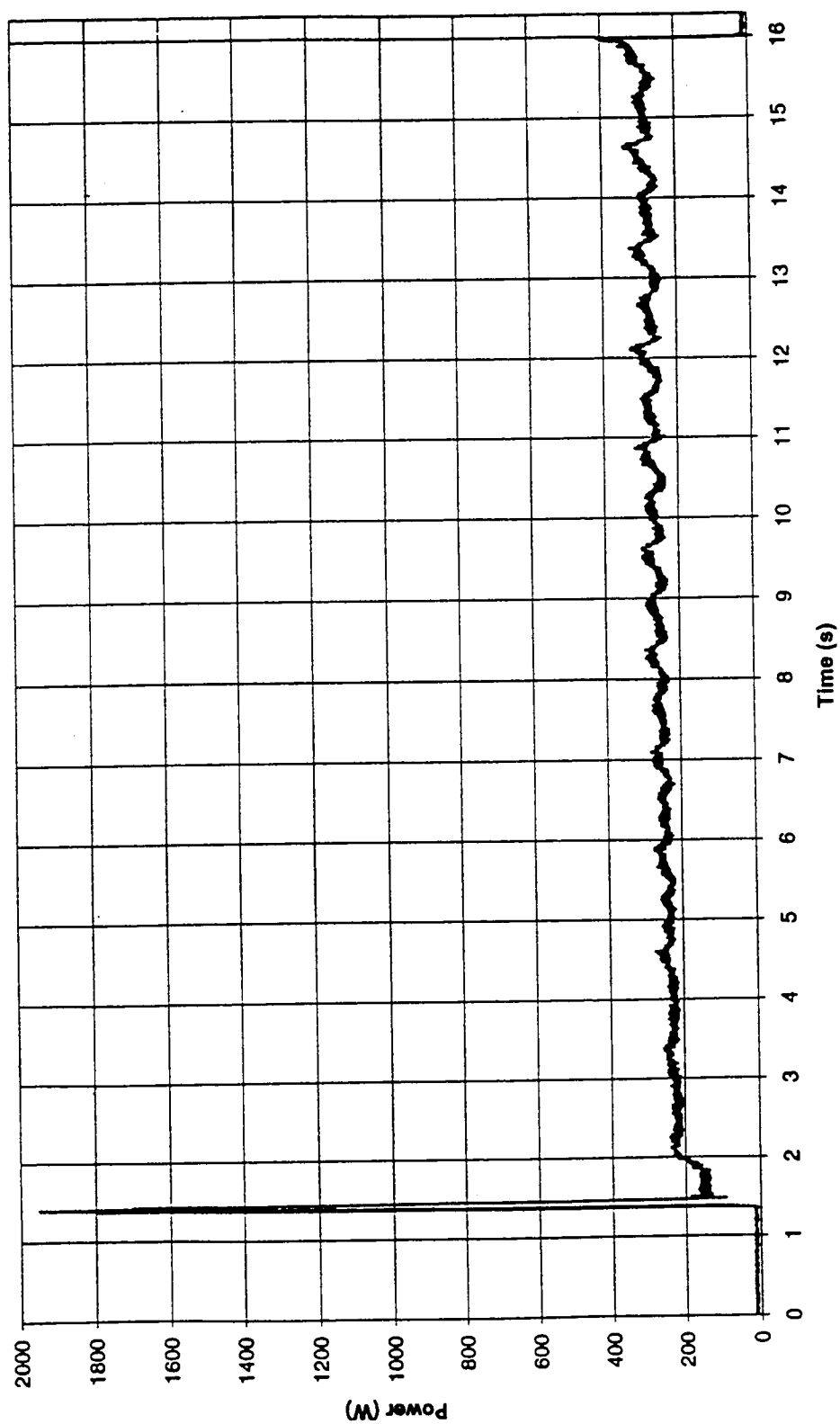


Figure 6.30 Limitorque 2-inch gate valve: normal stroke - OC.

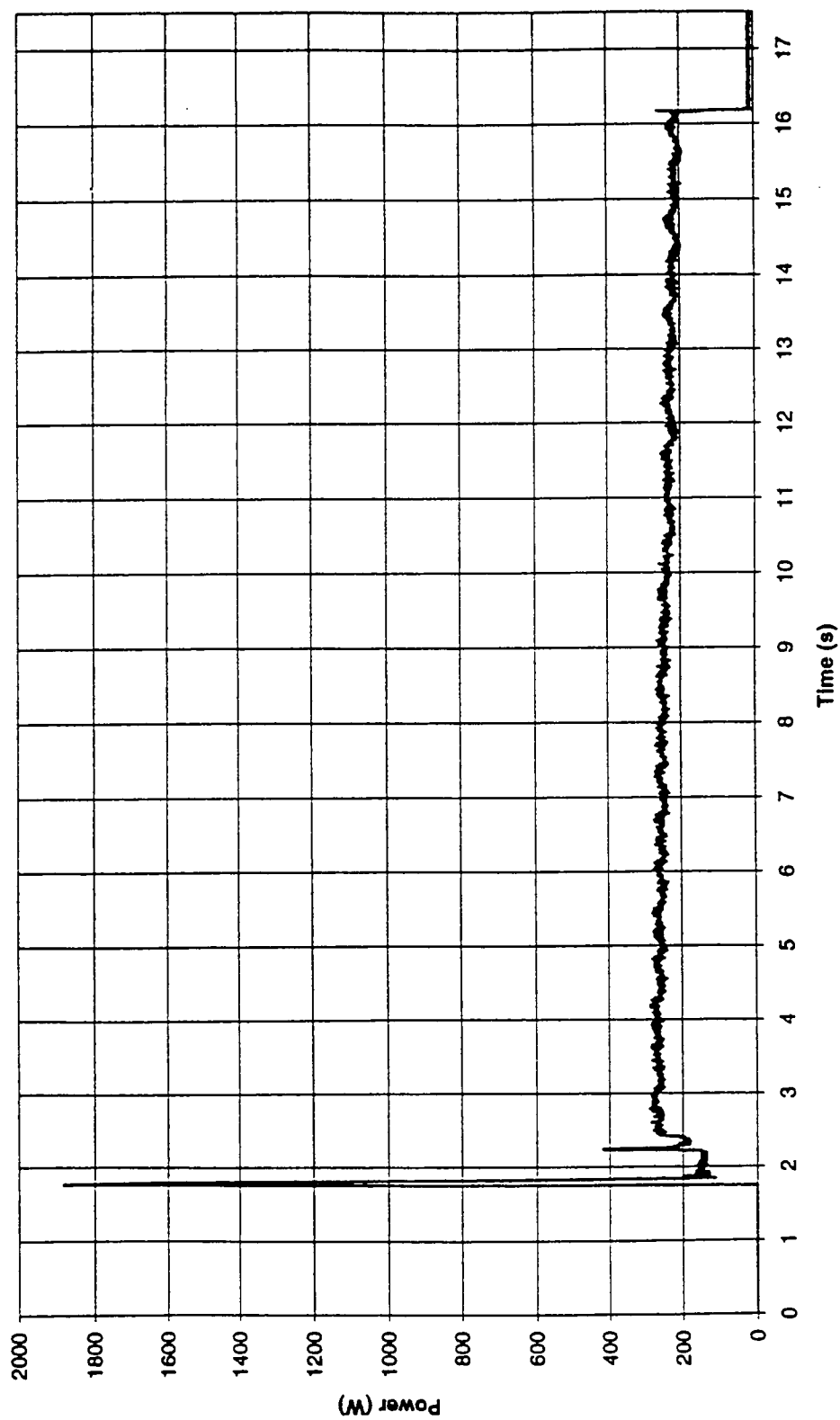


Figure 6.31 Limitorque 2-inch gate valve: normal stroke - CO.

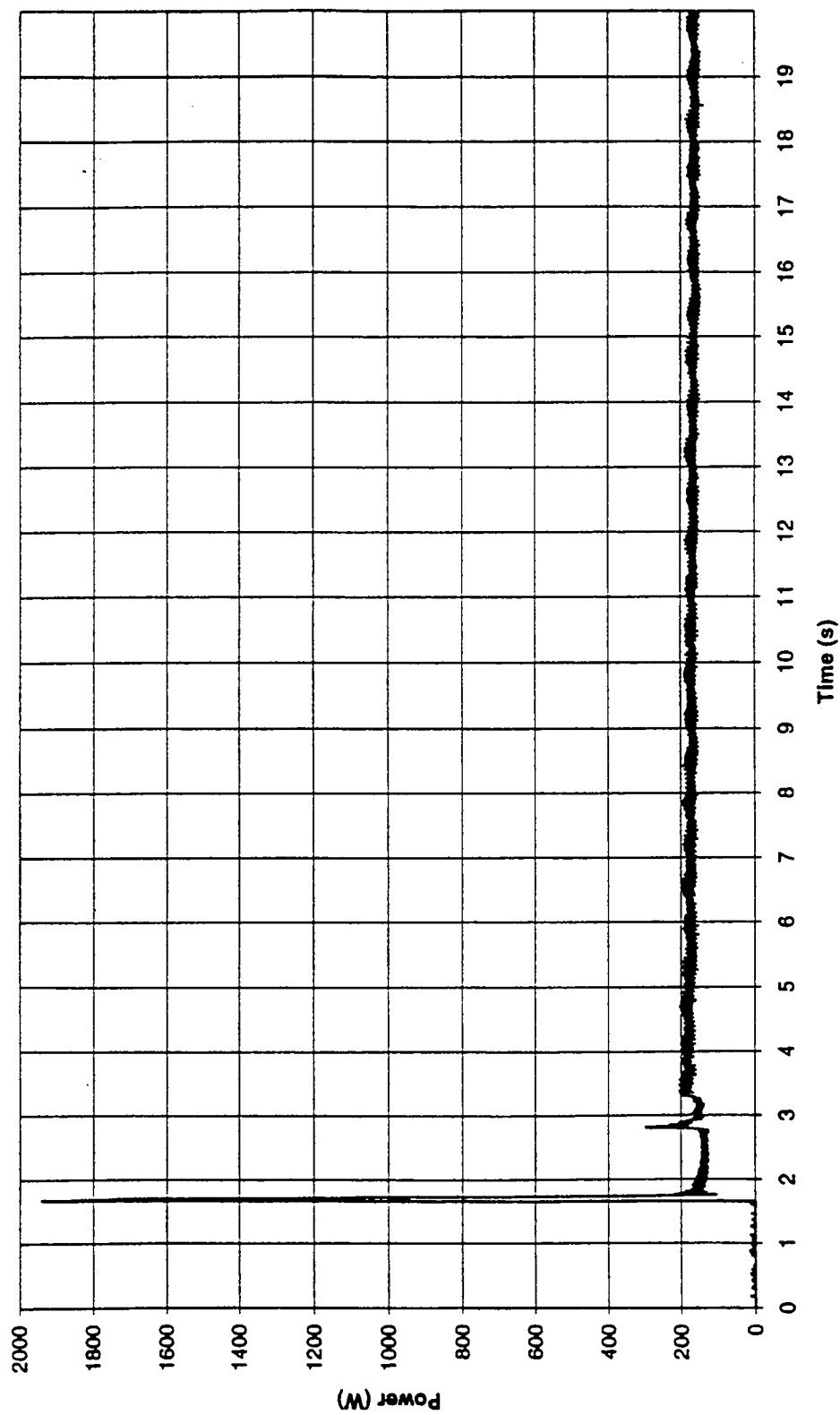


Figure 6.32 Limitorque 2-inch gate valve: bad pinion gear.

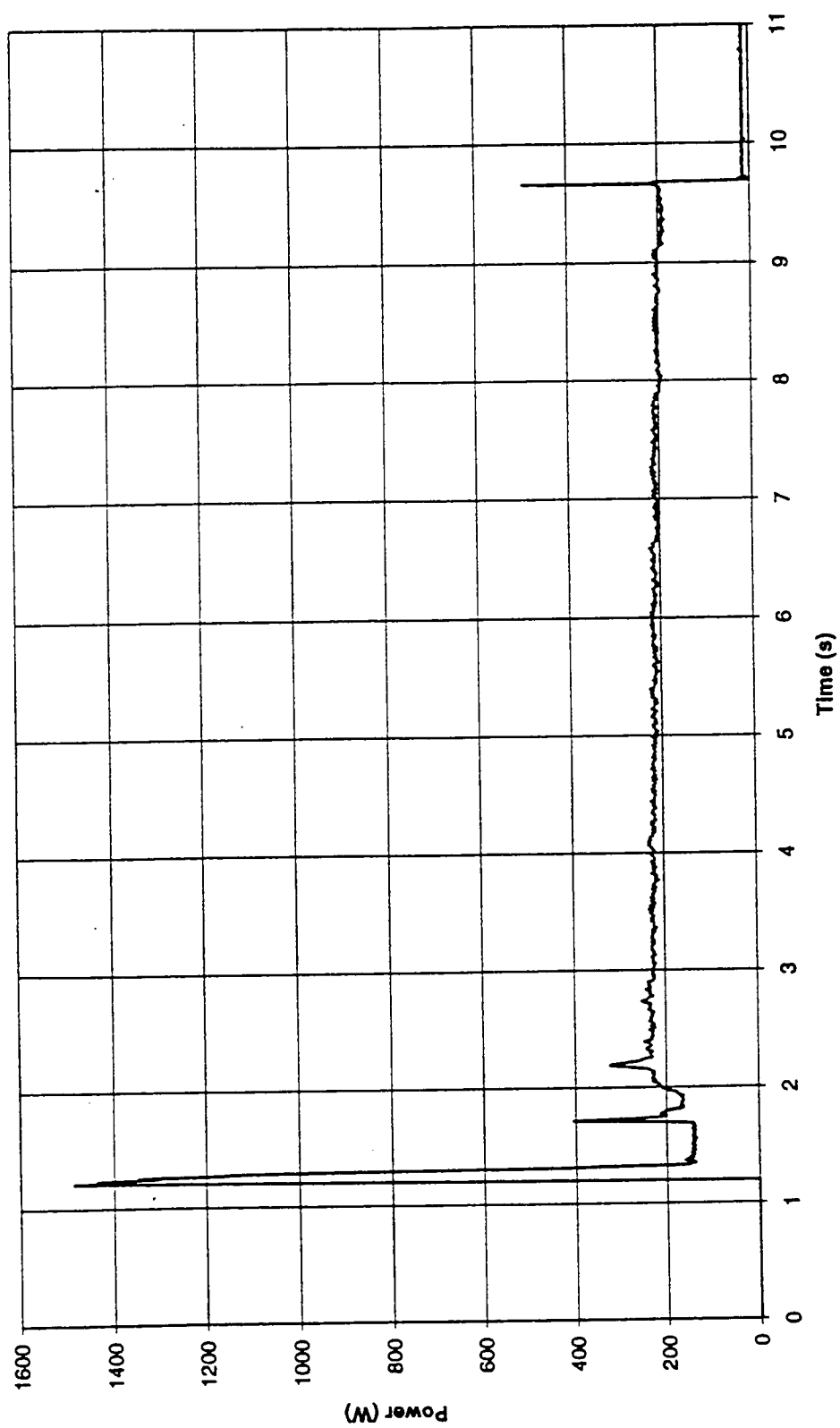


Figure 6.33 Limitorque 2-inch gate valve: misadjusted limit switch - CO.

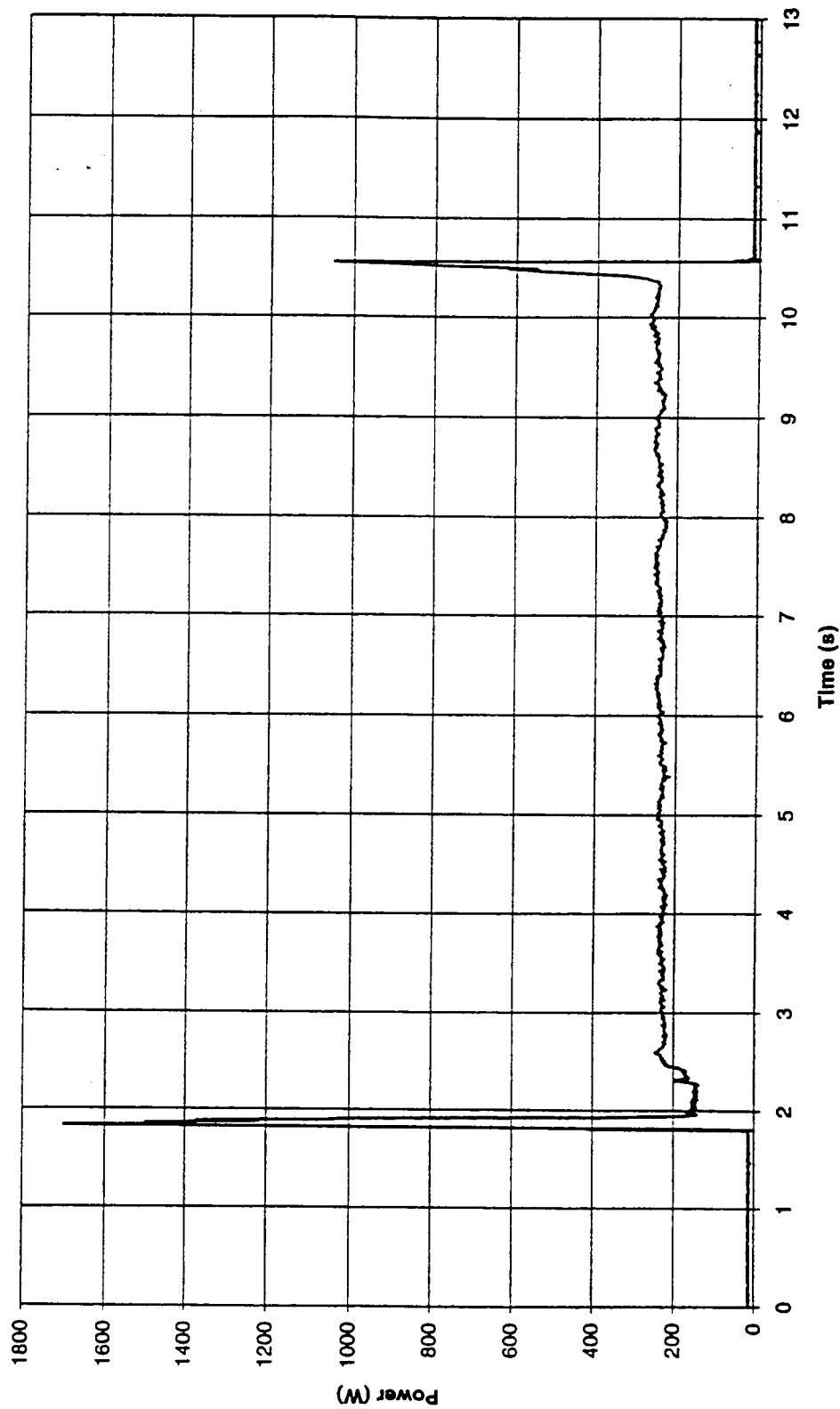


Figure 6.34 Limitorque 2-inch gas valve: : misadjusted limit switch - OC.

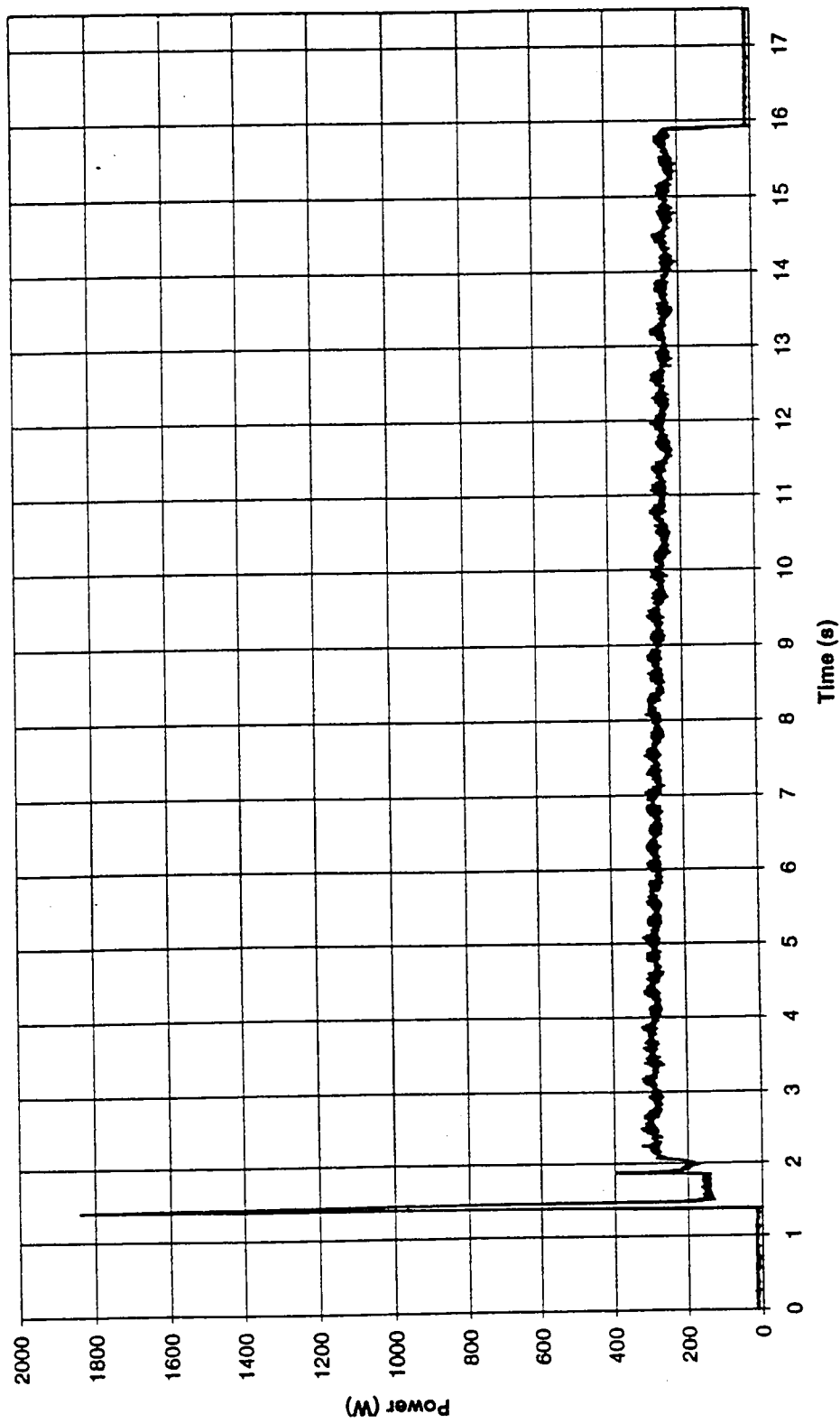


Figure 6.35 Limitorque 2-inch gate valve: broken rotor bar - CO.

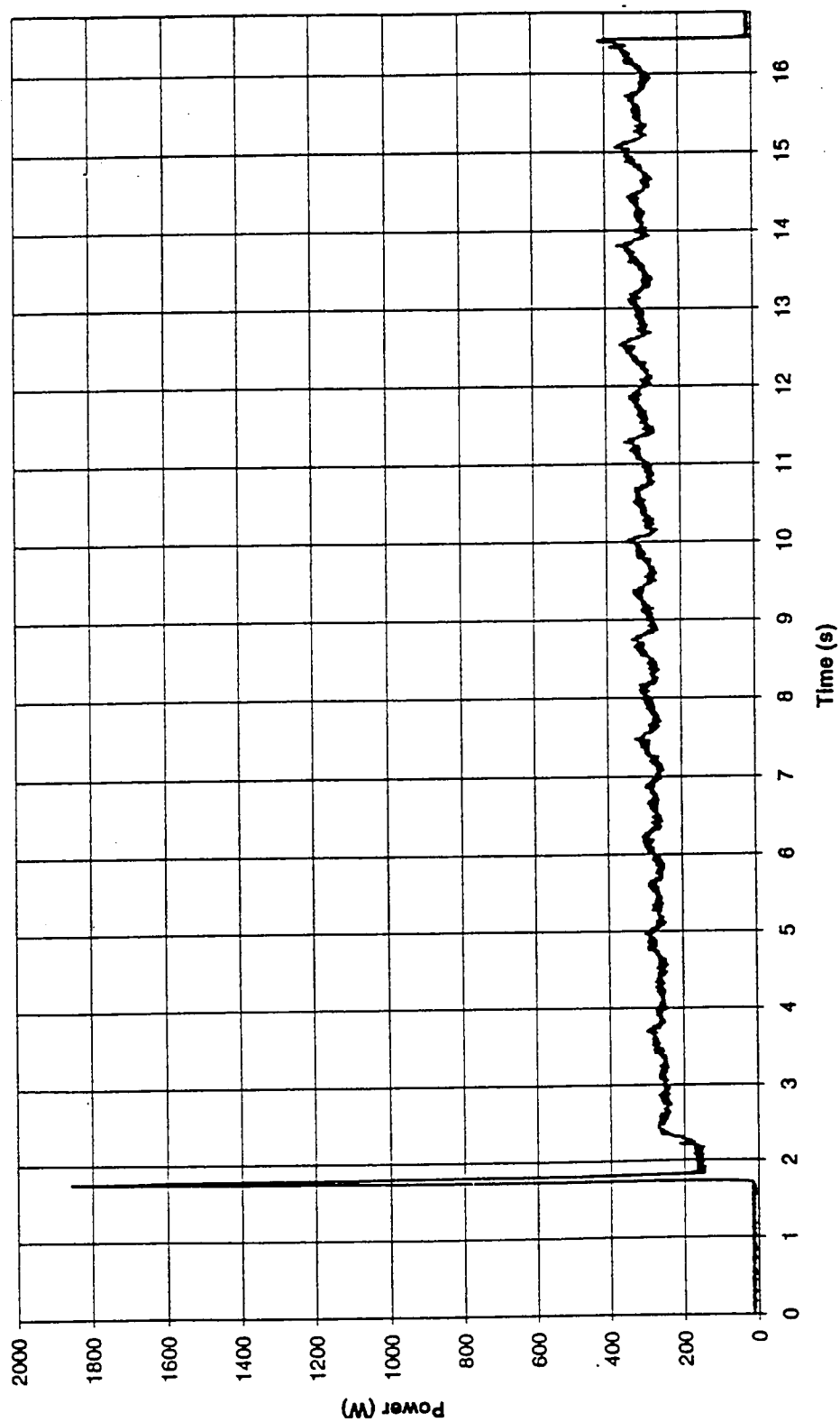


Figure 6.36 Limitorque 2-inch gate valve: broken rotor bar - OC.

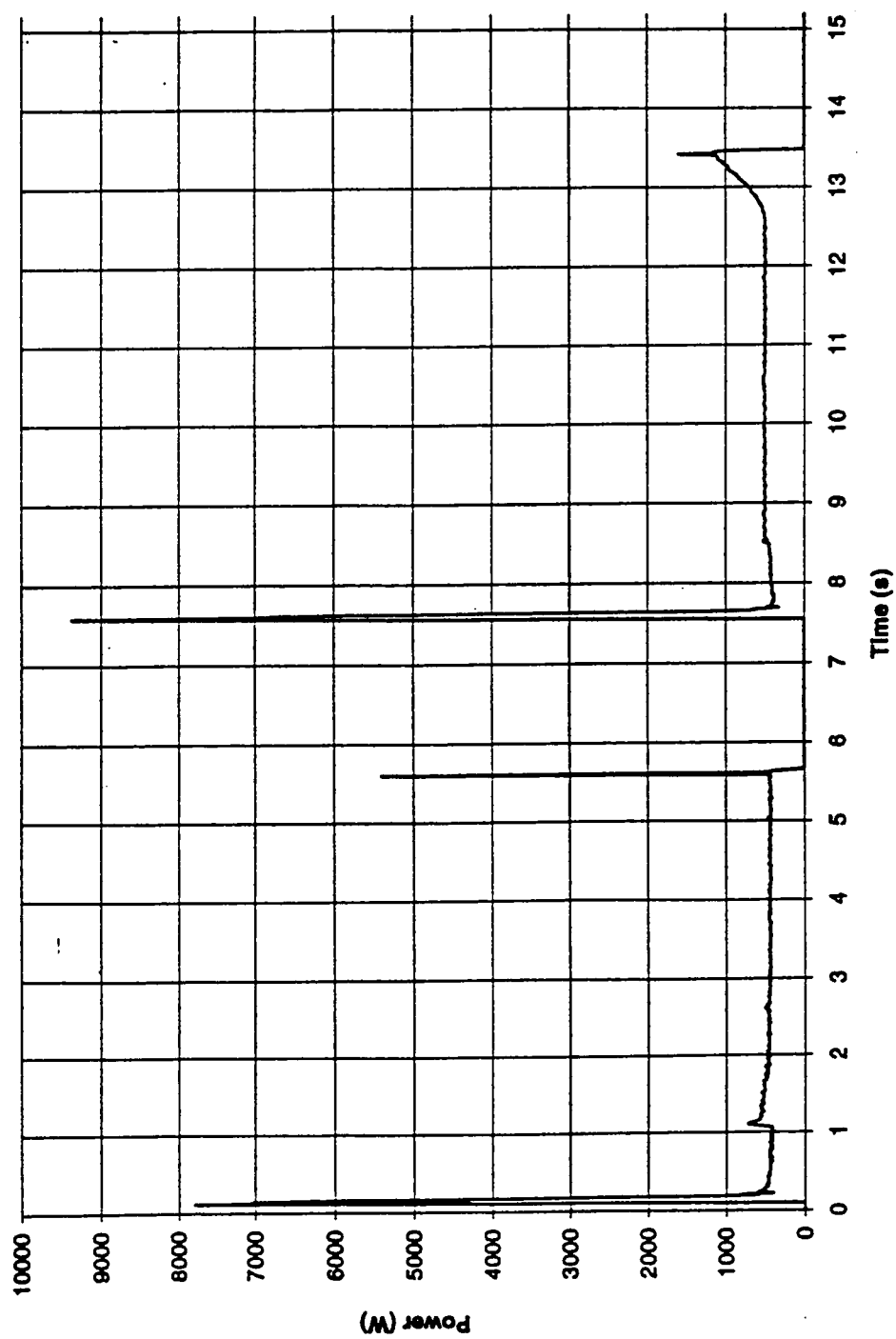


Figure 6.37 Kerotest 2-inch valve: pressure in the system = 0 psig - CO and OC.

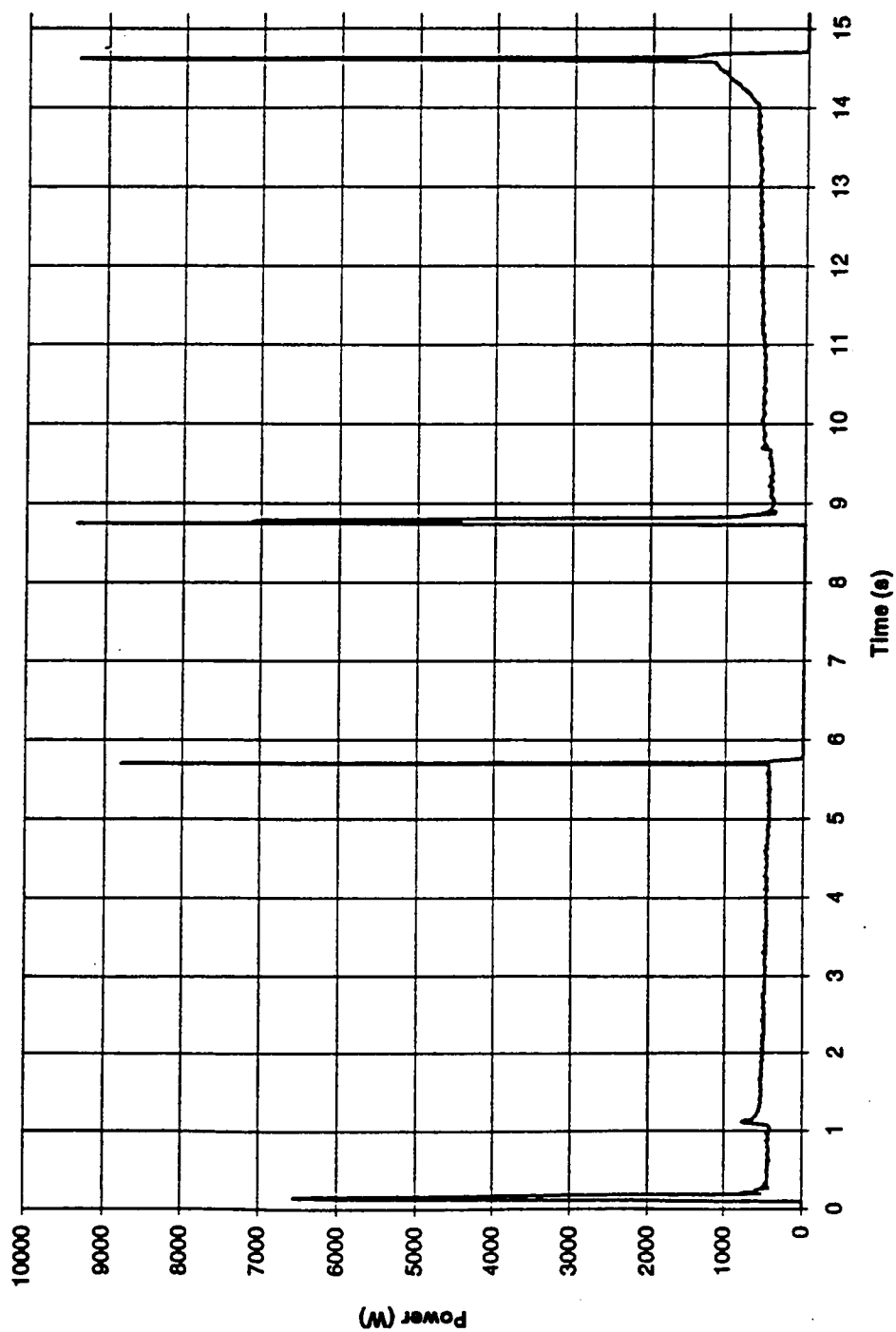


Figure 6.38 Kerotest 2-inch valve: pressure in the system = 500 psig - CO and OC.

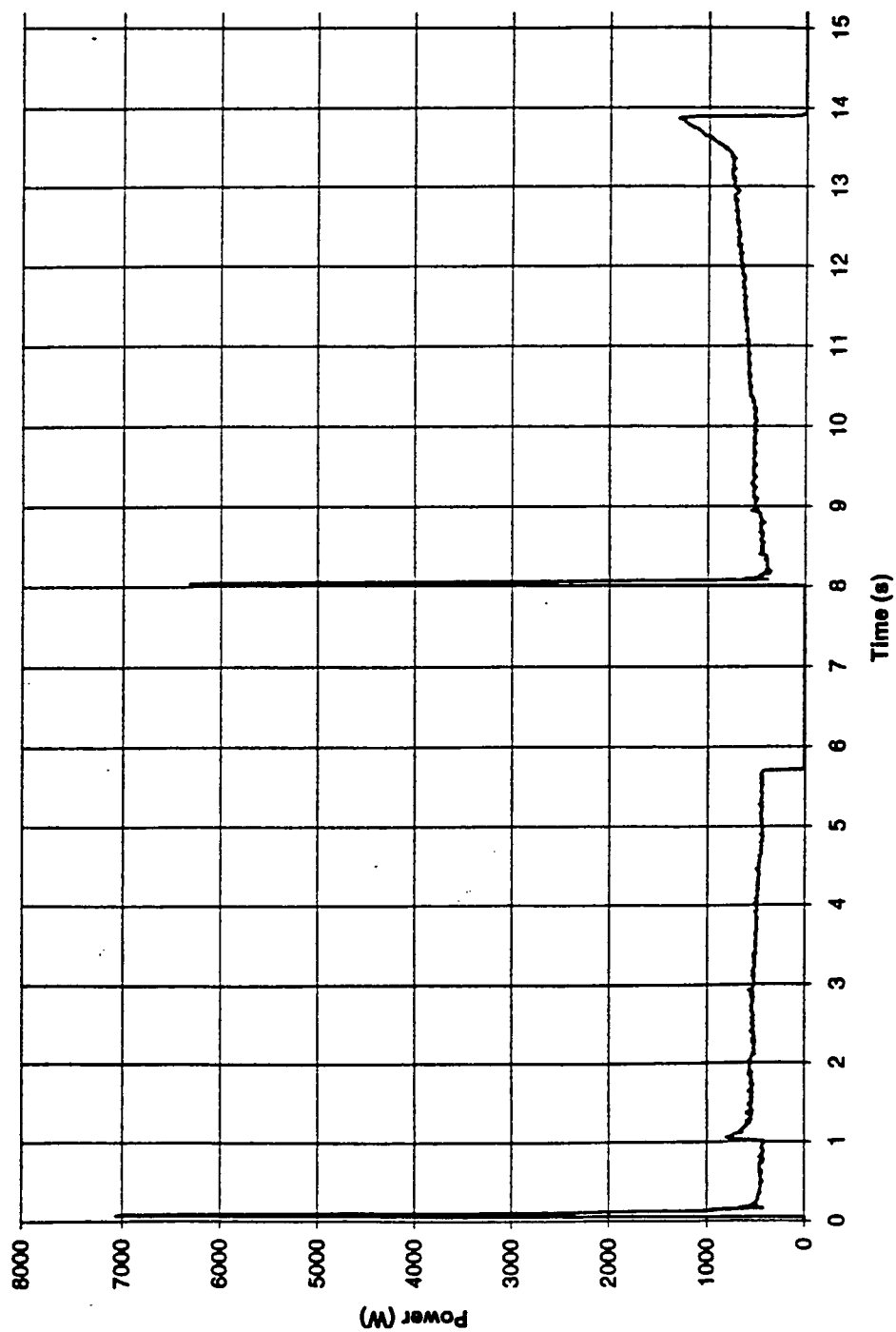


Figure 6.39 Kerotest 2-inch valve: pressure in the system = 1000 psig - CO and OC.

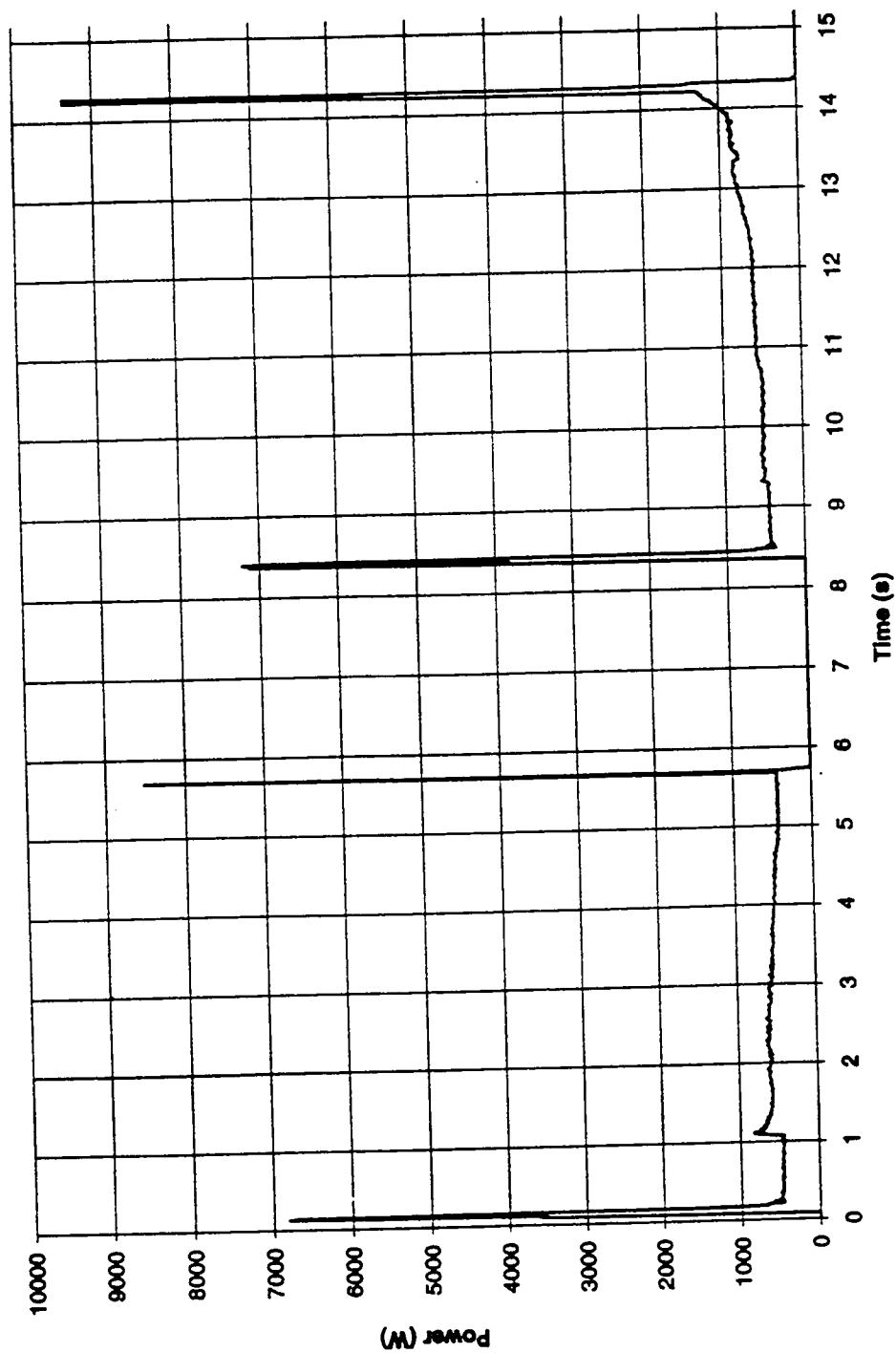


Figure 6.40 Kerotest 2-inch valve: pressure in the system = 1500 psig - CO and OC.

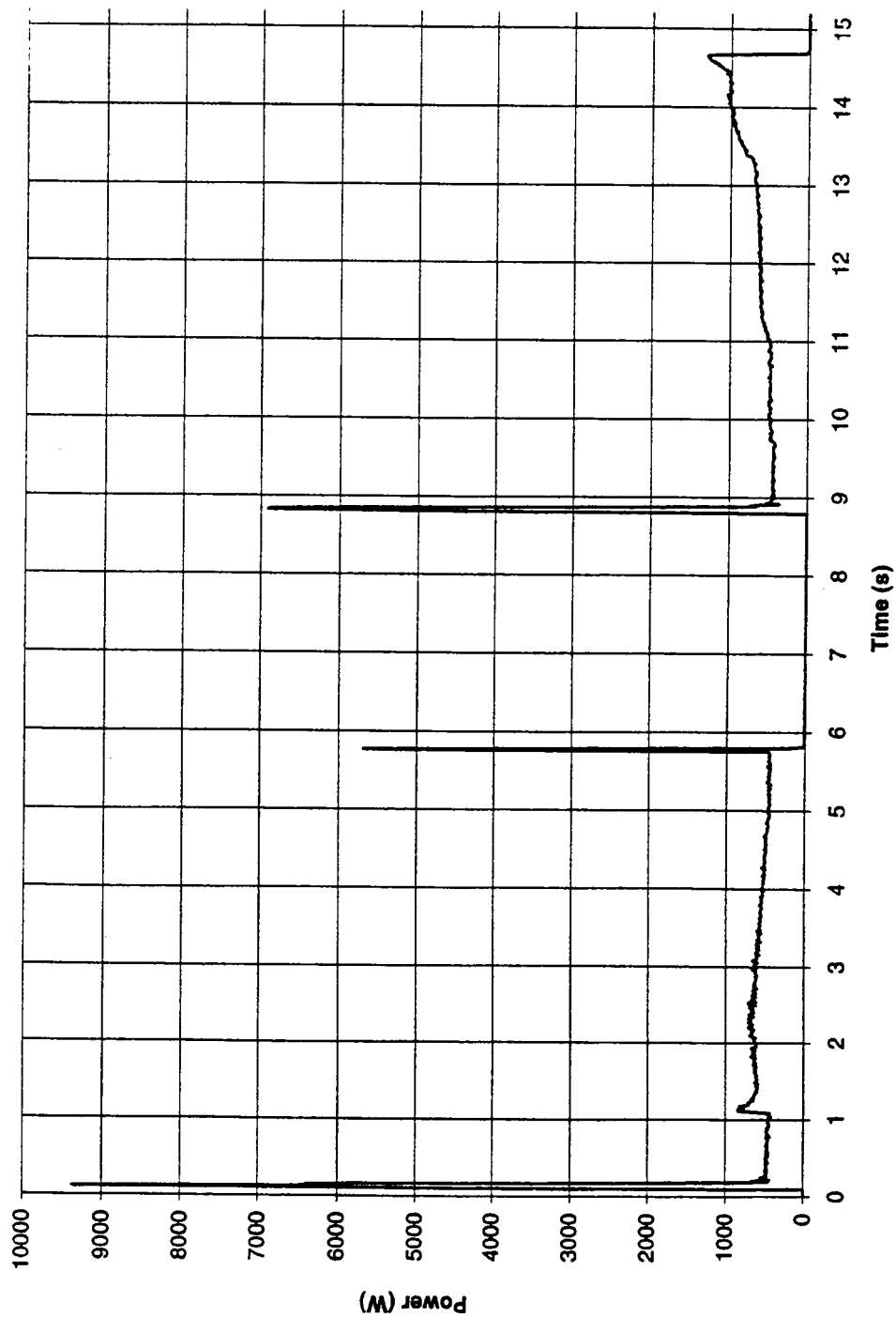


Figure 6.41 Kerotest 2-inch valve: pressure in the system = 2000 psig - CO and OC.

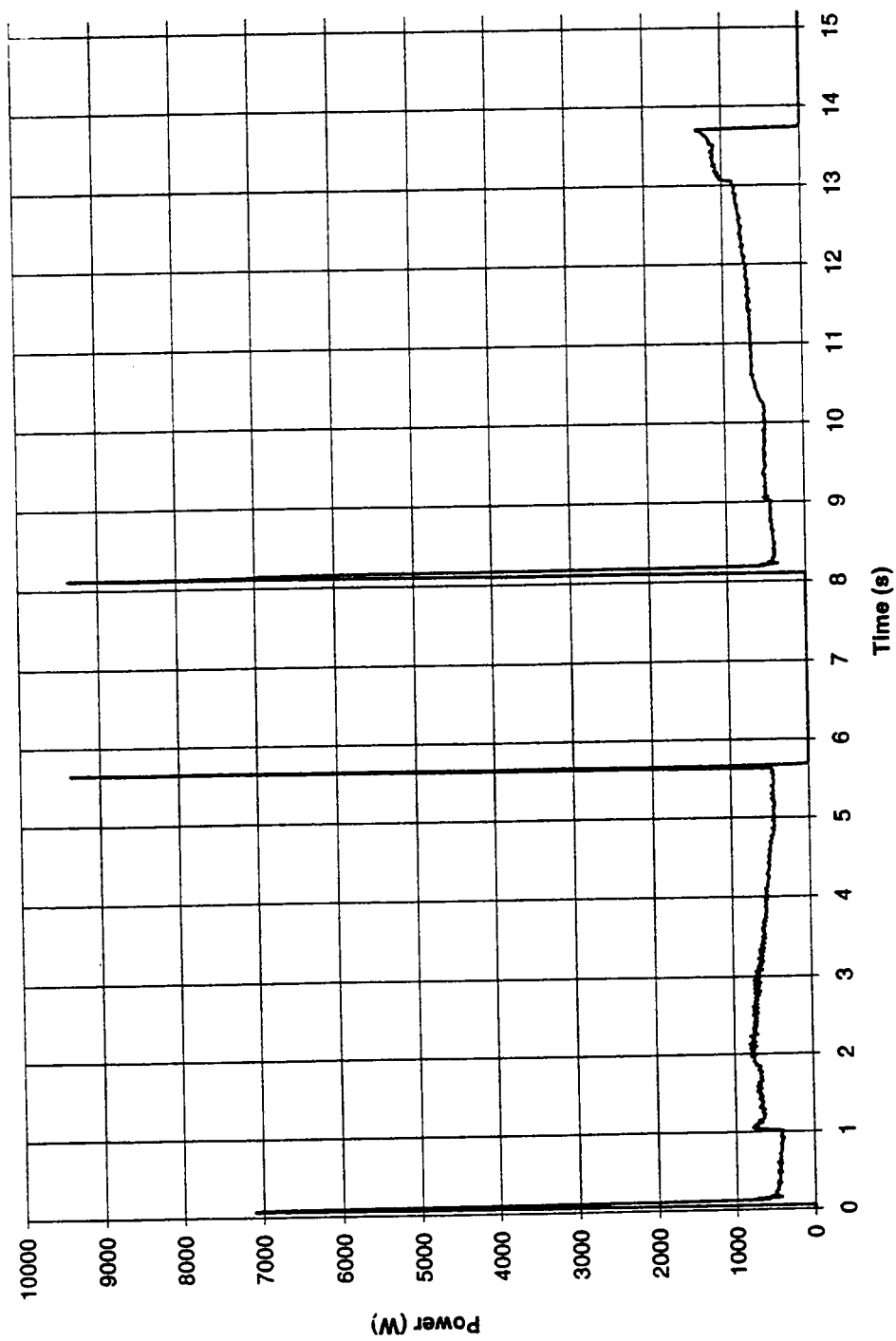


Figure 6.42 Kerotest 2-inch valve: pressure in the system = 2500 psig - CO and OC.

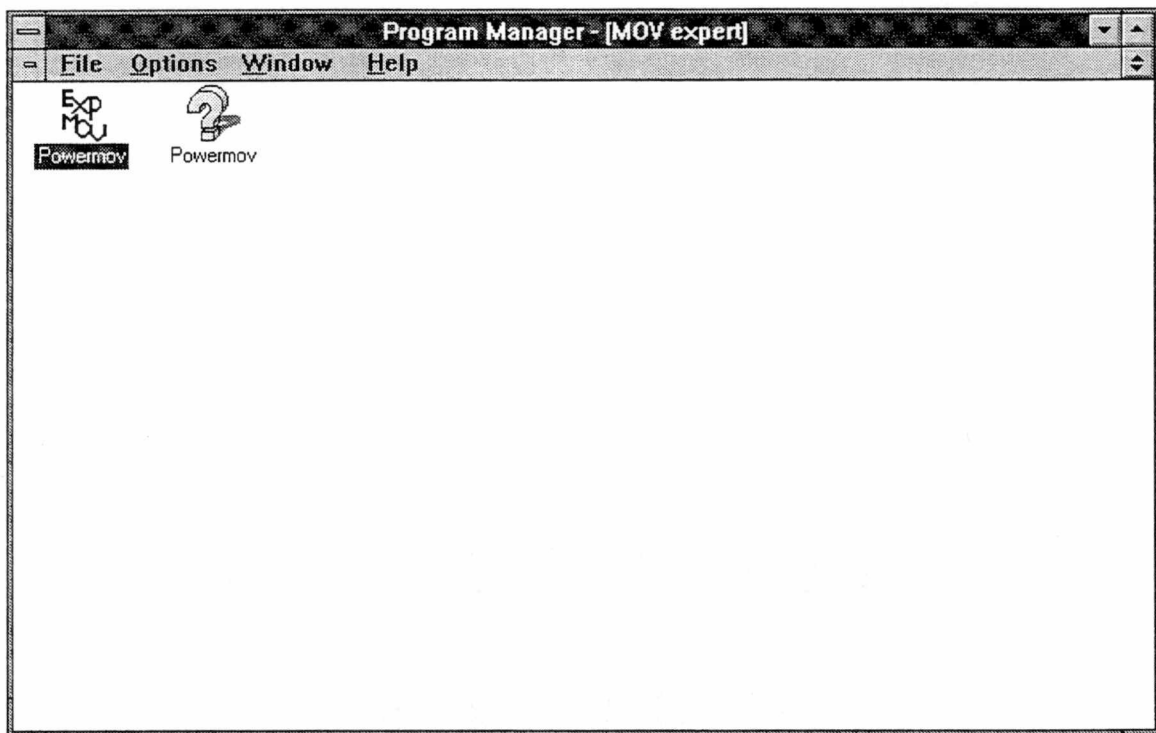


Figure 6.43 The desktop with program and help icons.

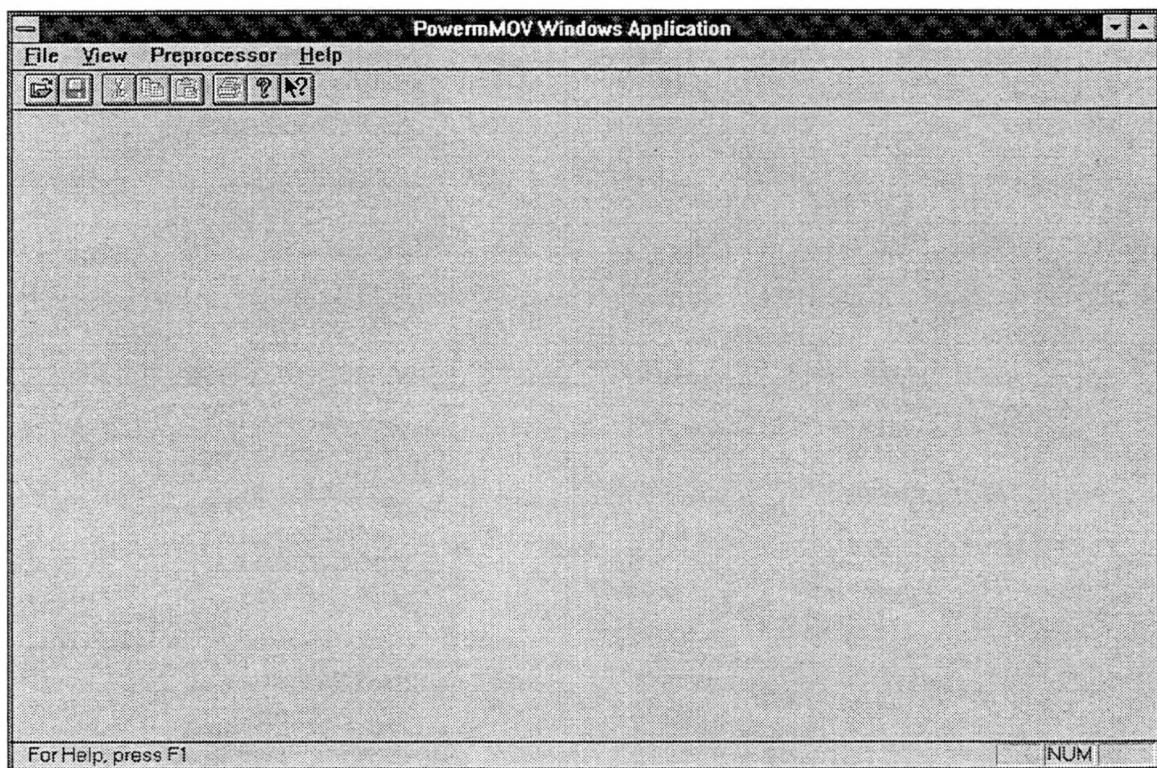


Figure 6.44 The main program window immediately after program is started.

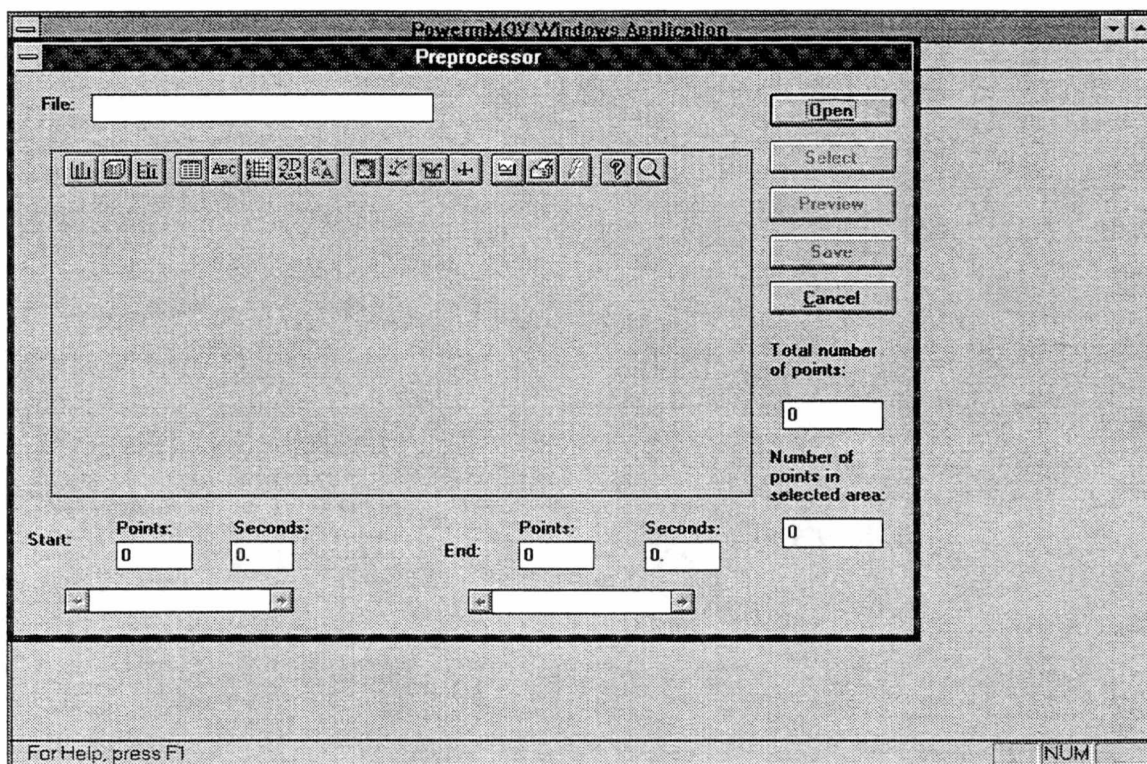


Figure 6.45 The preprocessor dialog box.

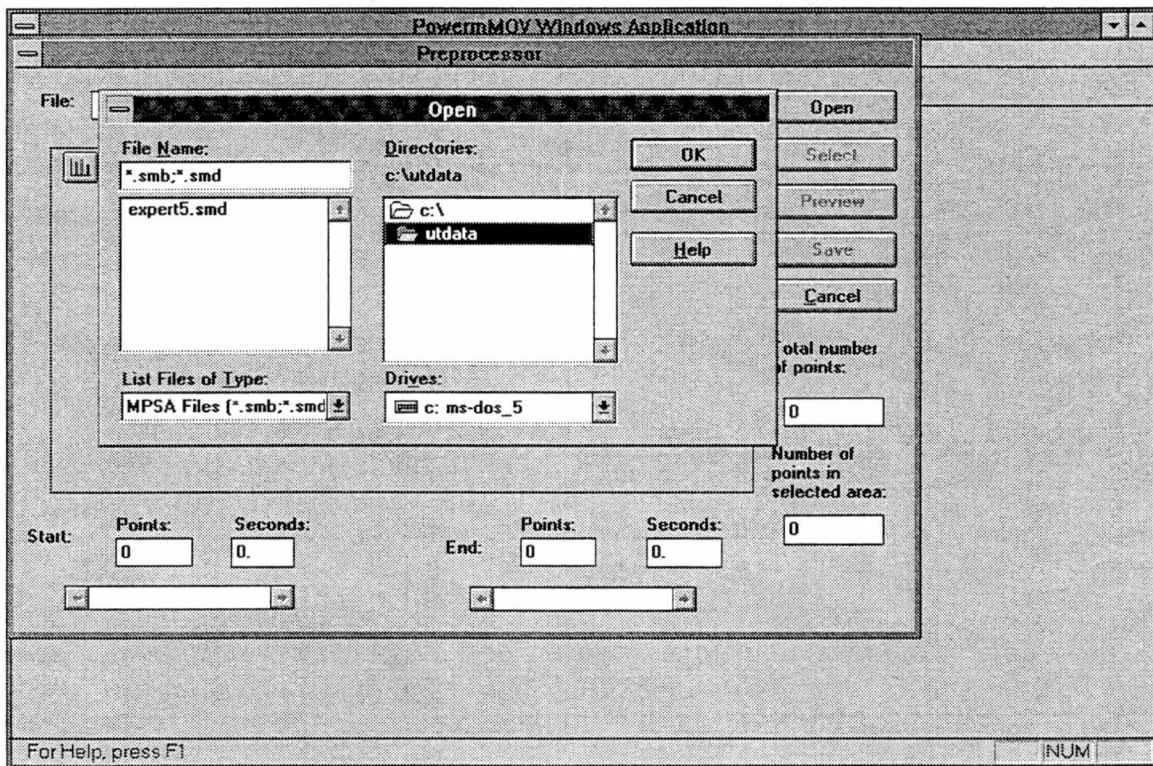


Figure 6.46 The open file dialog box.

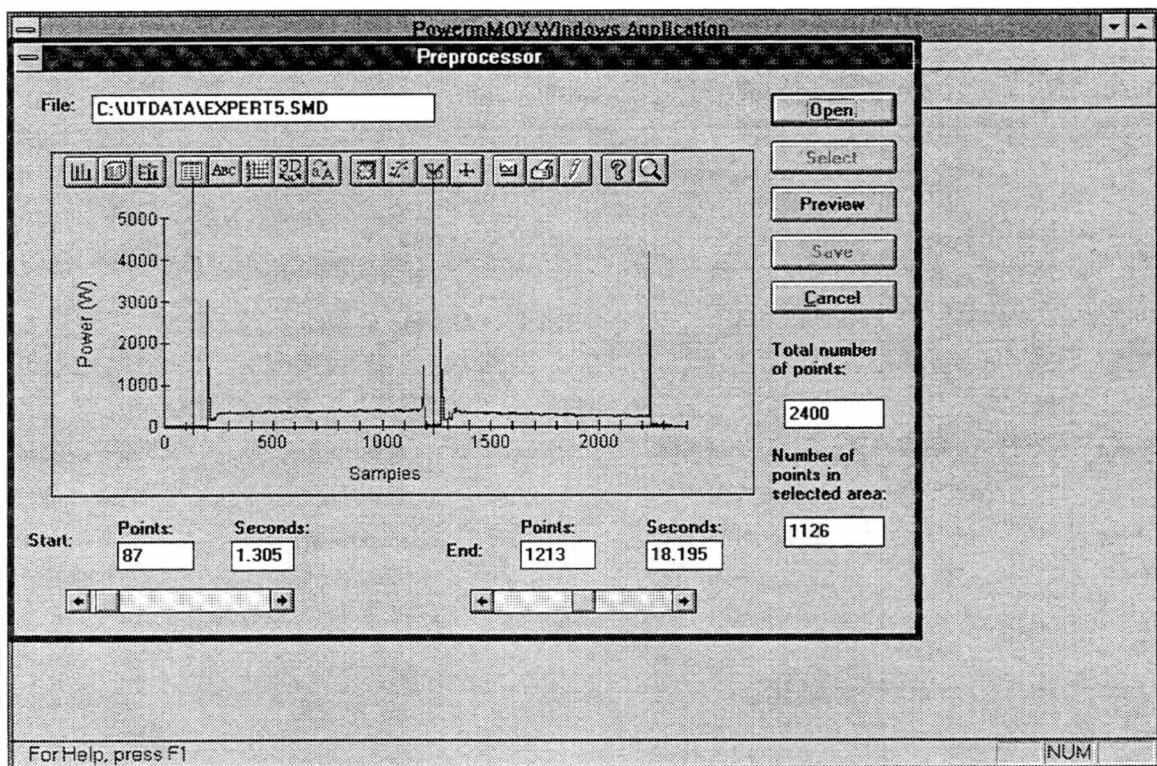


Figure 6.47 The motor power signature.

in the file. A single stroke may be selected using two scroll bars at the bottom of the dialog box. After the stroke is selected, it may be displayed by using the preview button. If the preview option shows single complete stroke (Figure 6.48) it may be saved using the save button. Before executing this step, the user needs to fill the dialog box with the data related to the valve. This dialog box is shown in Figure 6.49. The RMS values of the voltage and the average value of the current during the stroke are automatically computed and the corresponding boxes are filled. The file may be then saved using Save File dialog box, as it is shown in Figure 6.50. The preprocessed files have extension PMV.

The preprocessed files may then be further used in the program. The preprocessed files may be loaded using the open command from the file menu (Figure 6.51). The name of the file is then displayed in the form associated with each open file. The form is shown in Figure 6.52. The first step is to find signal primitives, and this can be done by clicking on the primitives button (Figure 6.53). The primitives are then used to find the events in the stroke (Figure 6.54). Within the Events dialog box there is a button that is used for the next step of the analysis: determining the features that are used in diagnosis process. Together with the measured features, the reference values are displayed, as shown in Figure 6.55. The difference between the measured and reference values is then fed into inference engine. The inference engine may be started by clicking on Diagnosis button. Then the new dialog box appears showing the results of the diagnosis (Figure 6.56).

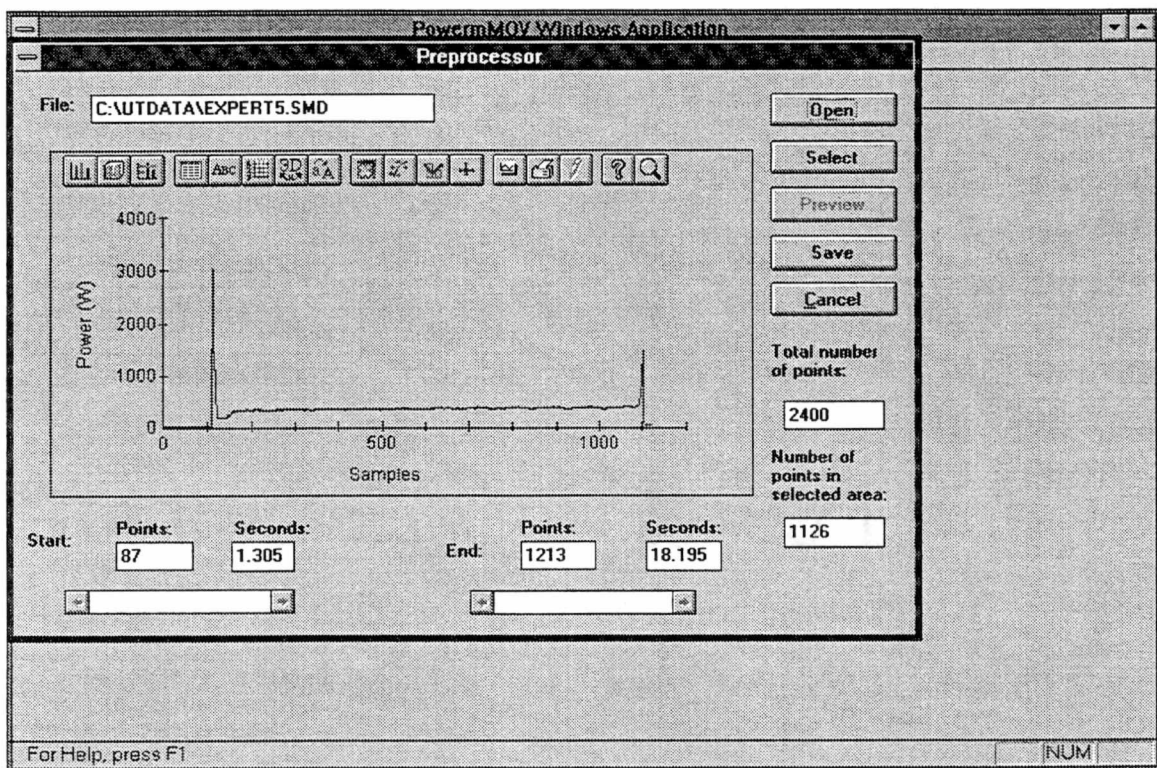


Figure 6.48 The selected area of the signature.

PowermMOV Windows Application

Valve Data

Plant:		Voltage 1	114.376	Volts
Unit:		Voltage 2	114.258	Volts
Valve type:		Voltage 3	112.162	Volts
Valve size:		Current 1	1.49958	Amps
Valve ID:		Current 2	1.33024	Amps
Stroke:		Current 3	1.16165	Amps
Date:		Save		
Reason:		Cancel		
Test Engineer:				

Total number of points:

 Number of points in selected area:

07	1.303	1213	10.133

For Help, press F1

NUM

Figure 6.49 The save dialog box.

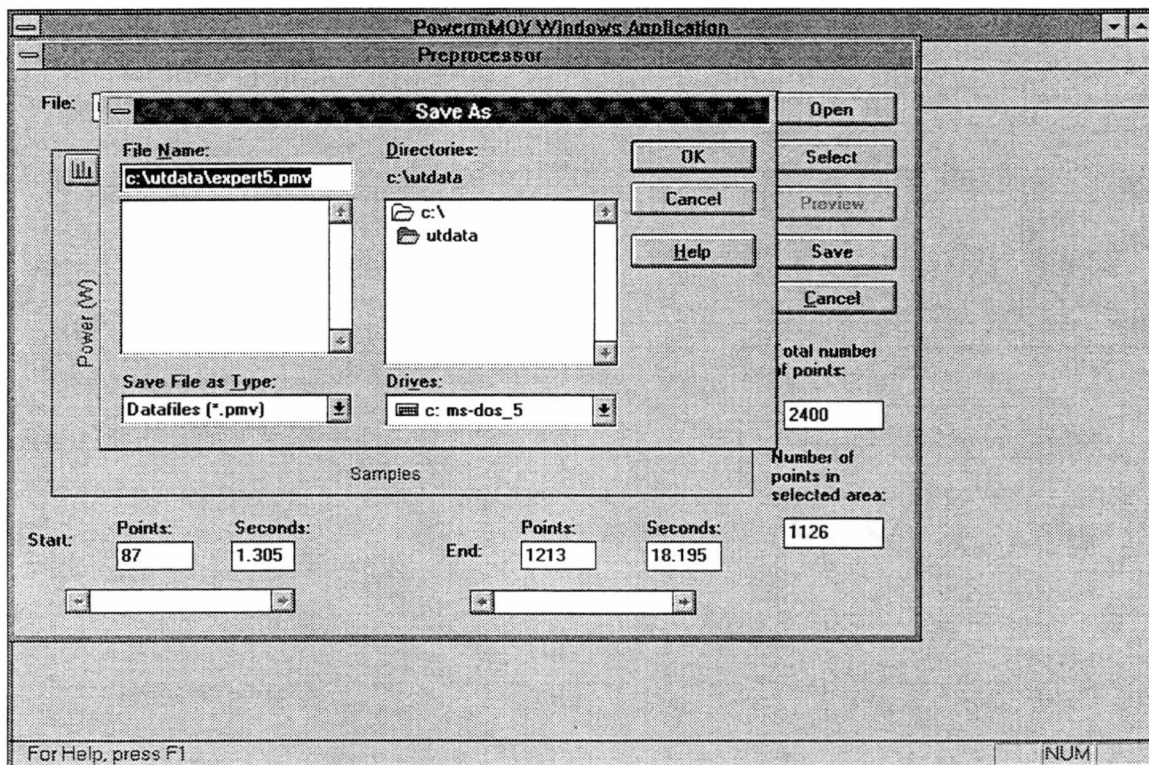


Figure 6.50 Save file dialog box.

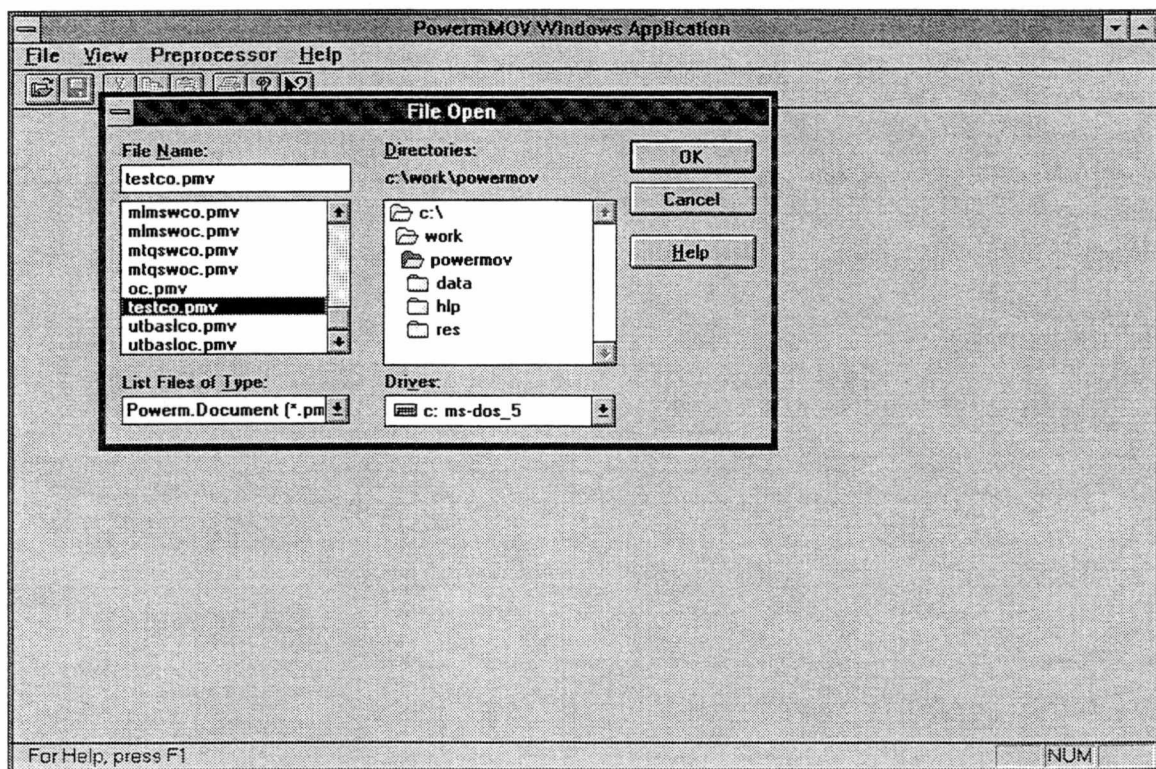


Figure 6.51 PMV file dialog box.

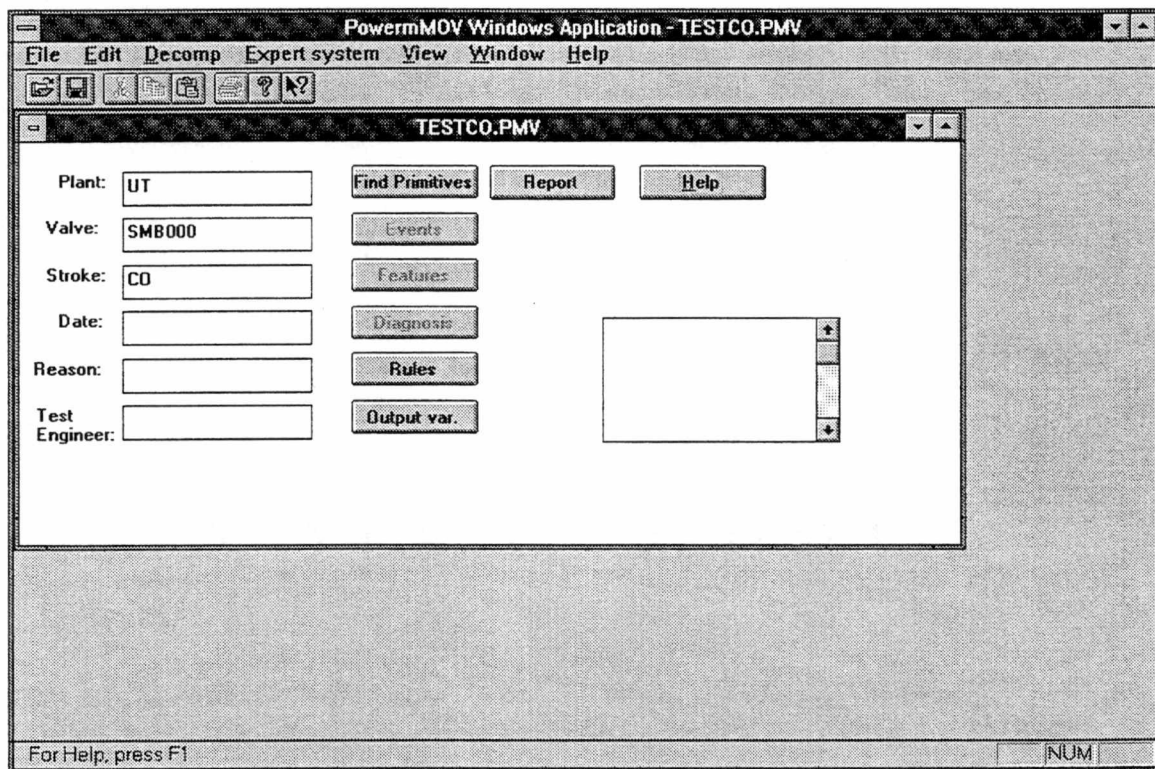


Figure 6.52 The form windows with command buttons is created after file was open.

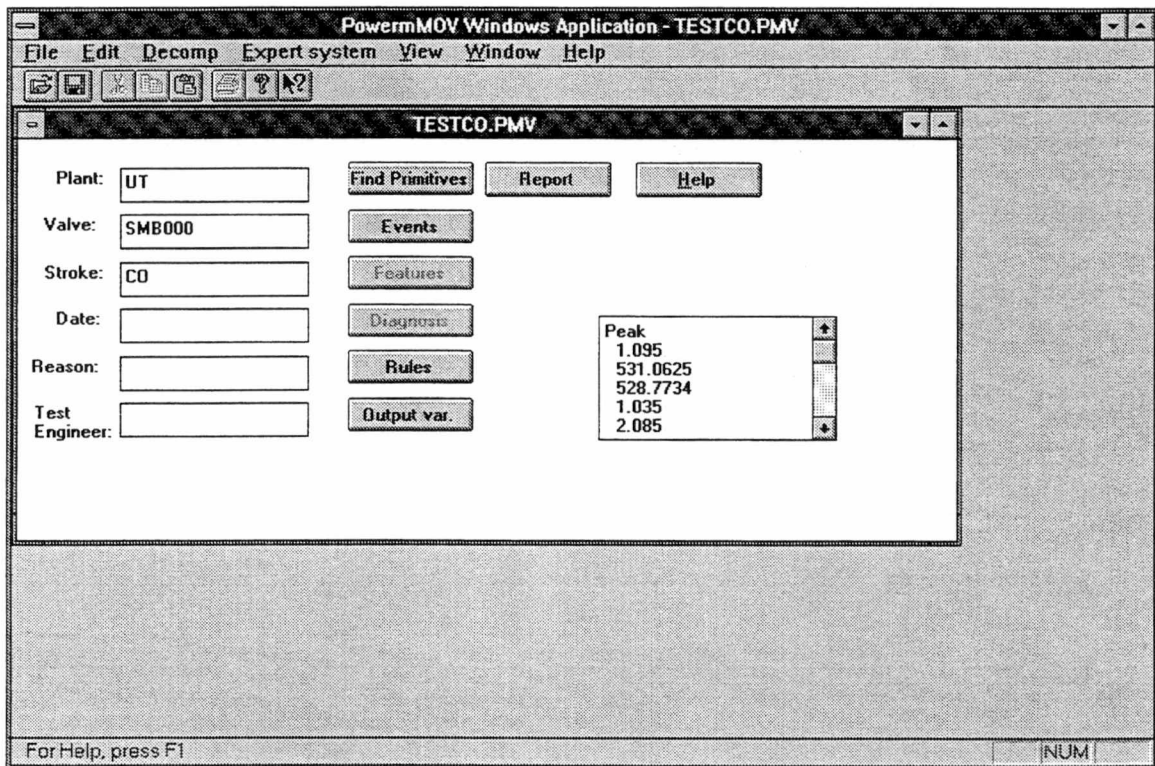


Figure 6.53 The primitives are found and displayed in the edit box.

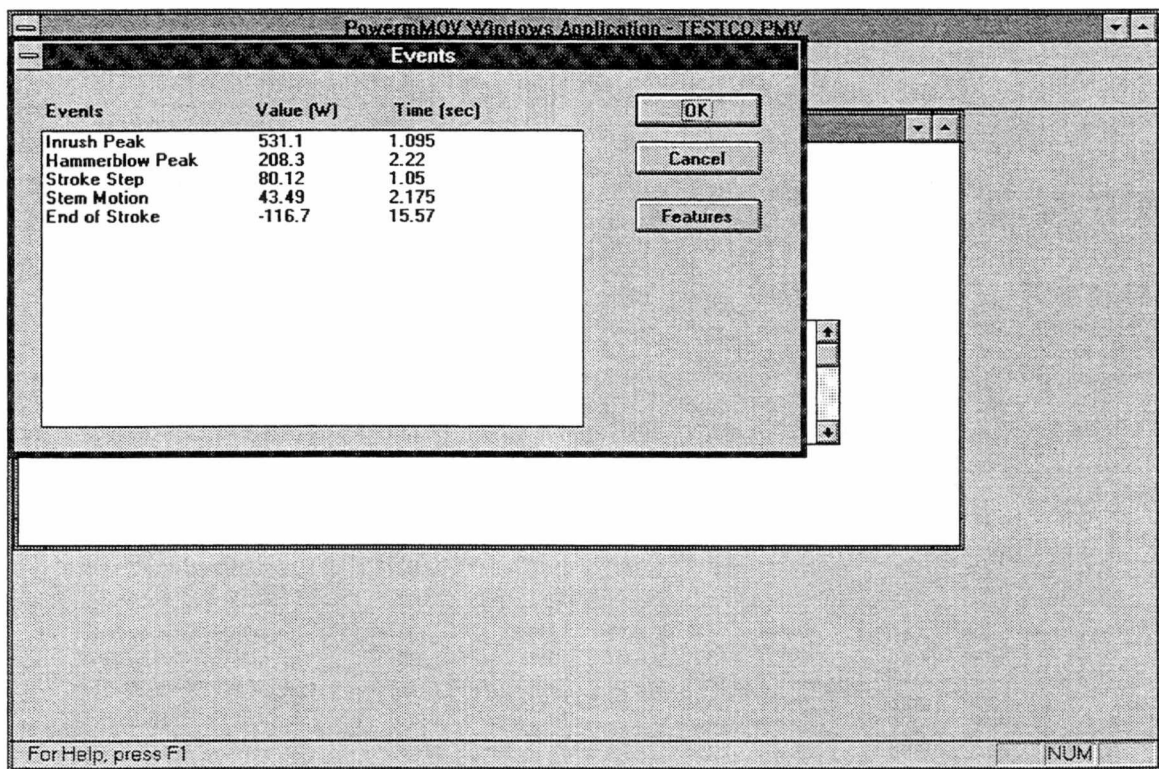


Figure 6.54 Events describing the stroke are displayed in the dialog box .

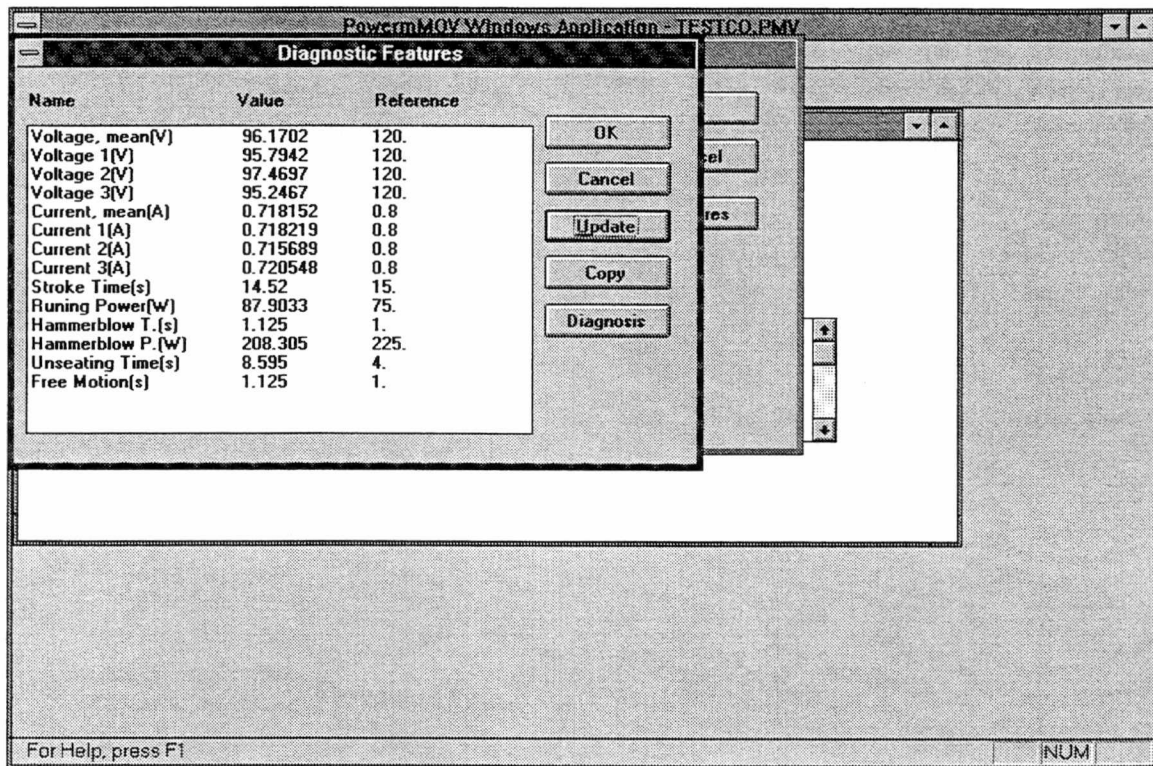


Figure 6.55 The features used in diagnosis.

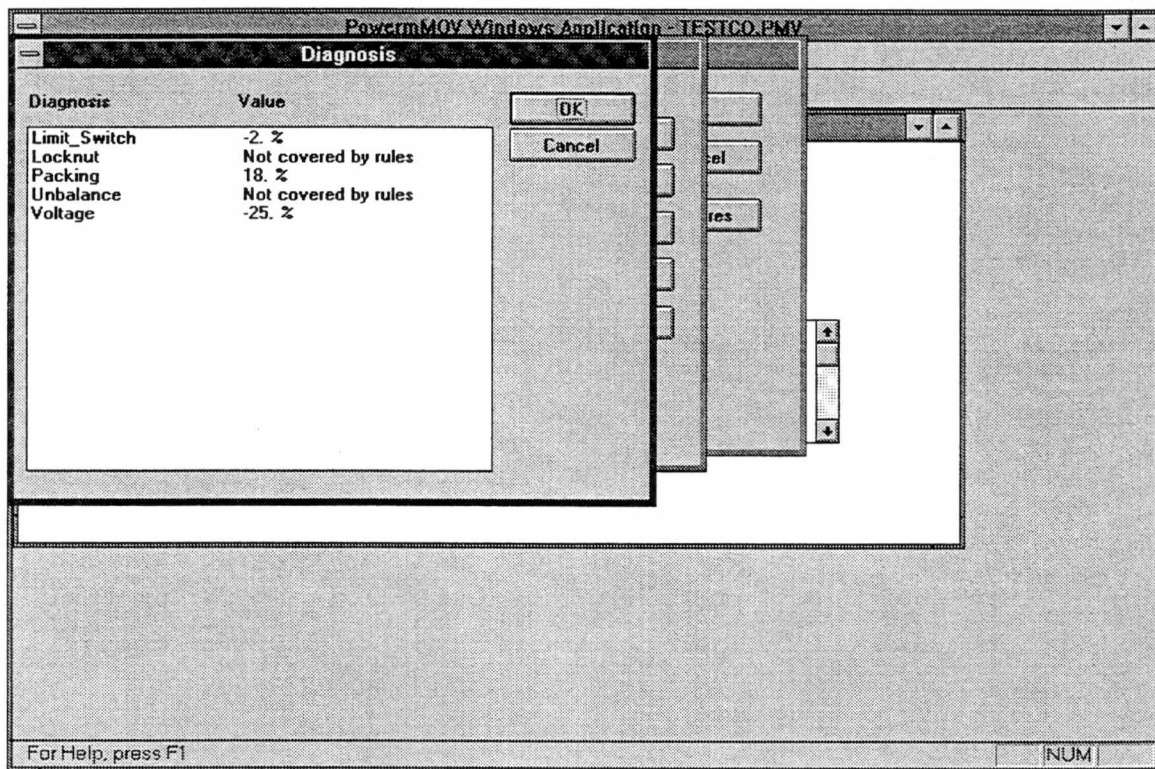


Figure 6.56 Diagnosis window.

All the rules must be defined before performing the diagnosis. This is done in the rule editor (Figure 6.57). This editor allows basic editing techniques as copy and paste. This makes it easy to create databases for new valves. If the new valve type is processed, the rules may be copied from the existing rule base and modified if necessary. Rules used in this example are listed in Table 6.3. The variables on the left hand side must have the same names as the diagnostic features, since they represents the input variables of the inference engine. The output variables represents possible malfunctions and must be defined separately before running inference engine. This can be done using Output variables dialog box shown in Figure 6.58.

A permanent record may be created for a particular valve analysis. This information is displayed in the report window, shown in Figure 6.59. This may be created in two ways: using report button in the main form and using report command from the File menu. The printing option is implemented and Print and Print Setup commands from File menu are enabled as long as the Report window is active. The information displayed in the report window consists of the information obtained to date. Therefore, if the report is generated before the diagnostic features are found, the report will not show any diagnostic feature.

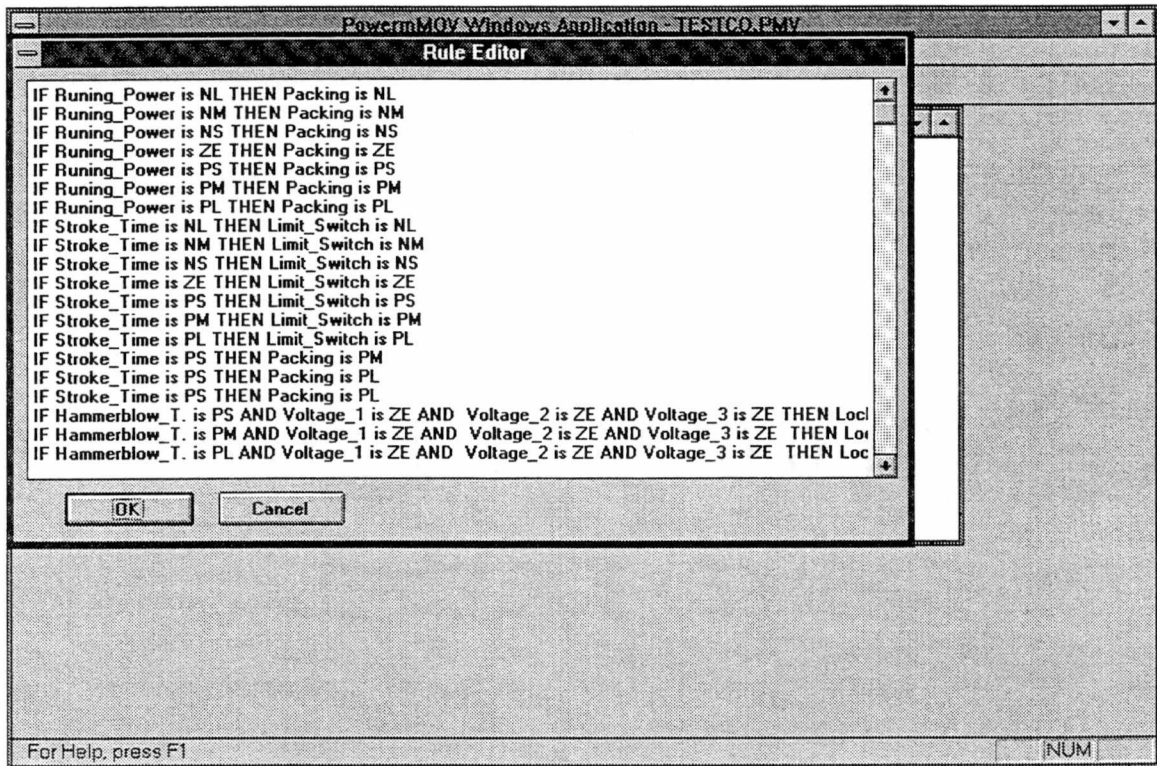


Figure 6.57 Rule editor.

Table 6.3 Rules used for MOV diagnosis

Rule #	Rules
1.	IF Runing_Power is NL THEN Packing is NL
2.	IF Runing_Power is NM THEN Packing is NM
3.	IF Runing_Power is NS THEN Packing is NS
4.	IF Runing_Power is ZE THEN Packing is ZE
5.	IF Runing_Power is PS THEN Packing is PS
6.	IF Runing_Power is PM THEN Packing is PM
7.	IF Runing_Power is PL THEN Packing is PL
8.	IF Stroke_Time is NL THEN Limit_Switch is NL
9.	IF Stroke_Time is NM THEN Limit_Switch is NM
10.	IF Stroke_Time is NS THEN Limit_Switch is NS
11.	IF Stroke_Time is ZE THEN Limit_Switch is ZE
12.	IF Stroke_Time is PS THEN Limit_Switch is PS
13.	IF Stroke_Time is PM THEN Limit_Switch is PM
14.	IF Stroke_Time is PL THEN Limit_Switch is PL
15.	IF Stroke_Time is PS THEN Packing is PM

Table 6.3 (cont.)

16	IF Stroke_Time is PS THEN Packing is PL
17.	IF Seating_Margin is NL THEN Torque_Switch is NL
18.	IF Seating_Margin is NM THEN Torque_Switch is NM
19.	IF Seating_Margin is NS THEN Torque_Switch is NL
20.	IF Seating_Margin is ZE THEN Torque_Switch is ZE
21.	IF Seating_Margin is PS THEN Torque_Switch is PS
22.	IF Seating_Margin is PM THEN Torque_Switch is PM
23.	IF Seating_Margin is PL THEN Torque_Switch is PL
24.	IF Seating_Margin is NL THEN Limit_Switch is NL
25.	IF Voltage_1 is NL AND Voltage_2 is NL AND Voltage_3 is NL THEN Voltage is NL
26.	IF Voltage_1 is NM AND Voltage_2 is NM AND Voltage_3 is NM THEN Voltage is NM
27.	IF Voltage_1 is NS AND Voltage_2 is NS AND Voltage_3 is NS THEN Voltage is NS
28.	IF Voltage_1 is NS AND Voltage_2 is NS AND Voltage_3 is NS THEN Voltage is NS
29.	IF Voltage_1 is ZE AND Voltage_2 is ZE AND Voltage_3 is ZE THEN Voltage is ZE
30.	IF Voltage_1 is PS AND Voltage_2 is PS AND Voltage_3 is PS THEN Voltage is PS

Table 6.3 (cont.)

31.	IF Voltage_1 is PM AND Voltage_2 is PM AND Voltage_3 is PM THEN Voltage is PM
32.	IF Voltage_1 is PL AND Voltage_2 is PL AND Voltage_3 is PL THEN Voltage is PL
33.	IF Current_3 is ZE AND Current is PS THEN Unbalance is ZE
34.	IF Current_3 is NS AND Current is PS THEN Unbalance is PS
35.	IF Current_3 is NM AND Current is PS THEN Unbalance is PM
36.	IF Current_3 is NL AND Current is PS THEN Unbalance is PL
37.	IF Current_1 is ZE AND Current is PS THEN Unbalance is ZE
38.	IF Current_1 is NS AND Current is PS THEN Unbalance is PS
39.	IF Current_1 is NM AND Current is PS THEN Unbalance is PM
40.	IF Current_1 is NL AND Current is PS THEN Unbalance is PL
41.	IF Current_2 is ZE AND Current is PS THEN Unbalance is ZE
42.	IF Current_2 is NS AND Current is PS THEN Unbalance is PS
43.	IF Current_2 is NM AND Current is PS THEN Unbalance is PM
44.	IF Current_2 is NL AND Current is PS THEN Unbalance is PL

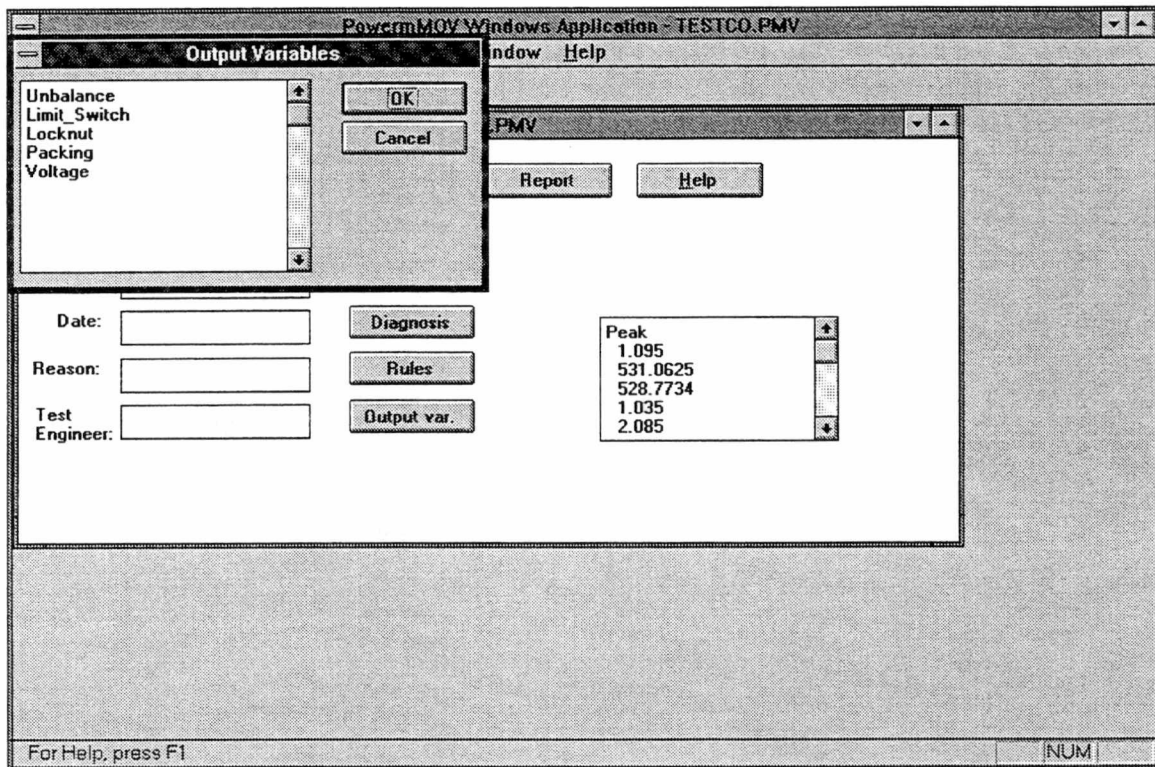


Figure 6.58 Output variable.

PowermMOV Windows Application - TESTCO.PMV				
File Edit Decomp Expert system View Window Help				
TESTCO.PMV:1				
TESTCO.PMV:2				
Valv	Voltage, mean [V]	96.1702	120 -0/+0	-19.8582 % *****
	Voltage 1 [V]	95.7942	120 -0/+0	-20.1715 % *****
Stro	Voltage 2 [V]	97.4697	120 -0/+0	-18.7753 % *****
	Voltage 3 [V]	95.2467	120 -0/+0	-20.6277 % *****
Da	Current, mean [A]	0.718152	0.8 -0/+0	-10.231 % *****
	Current 1 [A]	0.718219	0.8 -0/+0	-10.2226 % *****
Reas	Current 2 [A]	0.715689	0.8 -0/+0	-10.5388 % *****
	Current 3 [A]	0.720548	0.8 -0/+0	-9.93145 % *****
Test	Stroke Time [s]	14.52	15 -0/+0	-3.2 % *****
Engin	Runing Power [W]	87.9033	75 -0/+0	17.2044 % *****
	Hammerblow T. [s]	1.125	1 -0/+0	12.5 % *****
	Hammerblow P. [W]	208.305	225 -0/+0	-7.42014 % *****
	Unseating Time [s]	8.595	4 -0/+0	114.875 % *****
	Free Motion [s]	1.125	1 -0/+0	12.5 % *****

Figure 6.59 Report.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

7.1 Conclusions

The MOV diagnostic expert system demonstrated the ability to detect different conditions within the valve. The system is easy to implement. The emphasis has been placed on minimizing the user intervention. The system is capable of determining the anomalies that are currently present in MOVs. Additional improvement may be made in the system's capability to determine the degree of damage. This enhancement may be used in order to estimate the time until the next failure. The following are the significant contributions of this research.

- Development of a module for the detection of characteristic events during open-to-close and close-to-open valve strokes of an MOV using syntactic pattern recognition of motor power signature.
- Development of a Dynamic Data Exchange (DDE) link for data transfer between the data acquisition system and the expert system.
- Development of an expert system inference engine using fuzzy logic.
- Development of a user interface that utilizes the Windows operating system.

- Performance testing of the PC-based system using measurements on a Limitorque SMB000 valve actuator at a power company laboratory. Different conditions were simulated using a 1-inch globe valve and a 3-inch gate valve.
- Demonstration of the diagnostics system for on-line MOV anomaly detection.

7.2 Recommendations for Future Research

7.2.1 Program Improvement

The part of the program that will require constant changes is the rule base. Continued use of the program will show which malfunctions are well-covered by the rules and which ones need additional rules. There may also be malfunctions that are not covered by the current system, and the program therefore will require appropriate modifications so that the detection of these previously uncharacterized malfunctions is possible. The ease of rule editing was implemented in the current system with these potential changes in mind.

There is also a great diversity in MOV design. Malfunctions that are common in one MOV type, may never occur in another MOV. The parameters that are used for a particular MOV type to detect certain anomalies may not exist in some other MOV types. Therefore, new parameters may have to be added for some MOVs. Additional laboratory testing may be required to establish relationship between MOV anomalies and motor power signatures. In this version of the program, the parameter editor does not exist. The module that finds parameters is coded in the C++ program and changing of the analyzed

parameters would require a change in the code. Another method of defining parameters is possible, and when implemented, may increase the flexibility of the program.

The program is capable of detecting the present condition of an MOV. However, the maintenance work needed on an MOV must be decided by the maintenance personnel, using the valve condition as one of the decision parameters. The expert system may be expanded to incorporate a module for preparing a maintenance schedule for a particular MOV.

7.2.2 Database Support

One of the future improvements should consider signature trending and estimating operability lifetime. In order to achieve this, data obtained from previous measurements will be required. To accomplish this, the previously recorded data must be accessible by the trending algorithm. It is recommended that the tools of relational databases be used for this task.

The most commonly used database is the relational database. The concept of relational databases was developed as an answer to data centralization and fast and varied access to data information. The first prototype relational database system, called System R, was developed by IBM in the late 1960's and in the early 1970's [18].

The data generated by the diagnostics software can be written to an SQL database. The purpose of this database will be to collect data on valves in the same plant. A search of the database will give very important information about degradation of the valve system over time. The communication with the database may be achieved using the

Structured Query Language (SQL). SQL is widely used and the database created by the system will allow the use of certain commercial software packages, such as Access, Oracle, etc. By using the SQL syntax, the statement that extracts records according to specified criteria, can be constructed.

REFERENCES

1. R. C. Whitehouse, "The Valve and Actuator User's Manual," Mechanical Engineering Publication, London, 1993.
2. "Limitorque Type SMB Instruction and Maintenance Manual," Limitorque, Lynchburg, Virginia, 1982.
3. J. Eklund and B.R. Upadhyaya, "An Automated System for Motor-Operated Valve Diagnostics," Power Engineering, pp 38-41, December 1991.
4. H.D. Haynes, "Aging and Service Wear of Electric Motor-Operated Valves Used in Engineering Safety-Feature Systems of Nuclear Power Plants," NUREG/CR-4234, ORNL-6170/Vol. 2, 1989.
5. B.W. Roberts, "Power Signature Analysis for Motor Driven Device Monitoring and Diagnosis," Duke Power Company, 1993.
6. R.C. Hartley and A.E. Keller, "Test Quality," Second NRC/ASME Symposium on Pump and Valve Testing, 1992.
7. M. G. Say, "Alternating Current Machines," Halsted Press, New York, 1983.
8. S. A. Nasar, "Handbook of Electric Machines," McGraw-Hill, New York, 1987.
9. G. R. Slemon and A. Straughen, "Electric Machines," Addison-Wesley, Reading, Massachusetts, 1980.
10. J. M. Ivison, "Electric Circuit Theory," Van Nostrand Reinhold, New York, 1977.
11. L. H. Tsoukalas, "Fuzzy Logic," Notes for Tennessee Industry Week Short Course, August 15-19, 1994, Department of Nuclear Engineering, The University of Tennessee, Knoxville, August 1994.

12. G. Voit, "Fuzzy Logic: Concept to Constructors," AI Expert, pp. 26-33, November 1993.
13. J. Eklund and B. R. Upadhyaya, "Automatic Diagnostics of Motor-Operated Valves," Topical Report, Preventive Maintenance Engineering Laboratory, Department of Nuclear Engineering, the University of Tennessee, Knoxville, 1990.
14. R.J. Marshall, "The Determination of Peaks in Biological Waveforms," Computers and Biomedical Research, Vol .19, pp. 319-329, 1986.
15. J. Martin and S. Oxman, "Building Expert Systems," Prentice-Hall, Englewood Cliffs, 1988.
16. Visual C++ Programmers Guides, Microsoft, 1993.
17. S. Holtzner, "Microsoft Foundation Class Library Programming," Brady, New York, 1993.
18. S. Vang, "SQL and Relational Databases," Microtend Books, San Marcos, California, 1991.

VITA

Miodrag Glumac was born in Vinez, Croatia in 1965. In 1984 he received the high school diploma, and was accepted as a student in the Department of Electrical Engineering and Computer Sciences of the University of Ljubljana, Slovenia. He obtained the degree of graduated engineer (Bachelor of Science) in Electrical Engineering. After finishing his studies he worked for two years in Plomin, Croatia as an employee of the Croatian National Power Company. During this period he worked in the fields of automatic control, measurement and plant maintenance. In August 1993 he began graduate studies at The University of Tennessee, Knoxville, working as a graduate research assistant with Dr. Belle R. Upadhyaya, in the Nuclear Engineering Department.