A PROJECT REPORT

Submitted by

ARUN KUMAR.V	71201105005
BALAJI.B	71201105007
PRAKASH.P	71201105032
SURESH KUMAR V.T	71201105062

In partial fulfillment for the award of degree

Of

BACHELOR OF ENGINEERING

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 006

ANNA UNIVERSITY::CHENNAI - 600025

MAY-2005

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641006

ANNA UNIVERSITY::CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report "**VIRTUAL INSTRUMENTATION BASED APPLICATION FOR MOTOR – PUMP PERFORMANCE MONITORING SYSTEM**" is a bonafide work of

ARUN KUMAR.V	71201105005
BALAJI.B	71201105007
PRAKASH.P	71201105032
SURESH KUMAR V.T	71201105062

who carried out the project work under my supervision

HEAD OF DEPARTMENT
(Prof. K.Regupathy Subramanian)

SUPERVISOR
(R.K.Pongiannan)

CERTIFICATE OF EVALUATION

College : **KUMARAGURU COLLEGE OF TECHNOLOGY**

Branch : **Electrical & Electronics Engineering**

Semester : **Eighth Semester**

Sl.No.	Name of the Students	Title of the Project	Name of the Supervisor with Designation
01	V.ARUN KUMAR	"A VIRTUAL INSTRUMENTATION BASED APPLICATION FOR AC MOTOR –PUMP PERFORMANCE MONITORING SYSTEM"	MR.R.K.PONGIANNAN M.E. Senior Lecturer
02	B.BALAJI		
03	P.PRAKASH		
04	V.T. SURESH KUMAR		

The report of the project work submitted by the above students in partial fulfillment for the award of Bachelor of Engineering degree in Electrical & Electronics Engineering of Anna University were evaluated and confirmed to be report of the work done by the above students and then evaluated.

18/4/08

(INTERNAL EXAMINER) (EXTERNAL EXAMINER)

Acknowledgement

ACKNOWLEDGEMENT

We express our deepest sense of gratitude to our principle Dr.K.K.Padmanabhan, M.Sc (engg), M.Tech, Ph.D. under whose blessings this project has achieved its final shape. We are indebted to Prof.K.Regupathy Subramanian B.E., M.Sc., Head of the department, and Electrical and Electronics Engineering for his valuable guidance all through the project.

Our wholehearted thanks to Mr.K.Rajan M.E., Assistant Professor, Electrical and Electronics Engineering department who has supported us throughout our project tenure.

We express our sincere gratitude to Mr.R.K.Pongiannan M.E., Senior Lecturer, and Department of Electrical and Electronics Engineering, who has guided us throughout the project.

We have immense pleasure in placing on record, our deep sense of gratitude to our dear parents and beloved friends who helped us to make our project successful.

A VIRTUAL INSTRUMENTATION BASED APPLICATION FOR AC MOTOR –PUMP PERFORMANCE MONITORING SYSTEM

Synopsis

Synopsis

Currently industrial applications require suitable monitoring systems able to identify all the performance parameters of machineries to operate the plant at higher efficiency as well as to perform the adequate maintenance. Most of the continuous process plants employ electrical equipments like motors, heaters, pumps and so on. Hence we made an attempt to monitor the parameters of a motor-pump system. The information coming from a general purpose monitoring system can be usefully exploited to realize a monitoring instrument for the monitoring of an AC Motor – Pump system and can be fed to a diagnostic tool for providing useful preventive measures.

This project "A virtual Instrumentation based application for AC Motor –Pump monitoring system" has been designed and implemented in LabVIEW 6.1 for 0.5 kW, 230V, 2880 rpm, 50Hz, single phase Motor-Pump system. The method is based on digital processing of the line signals acquiring by means of a virtual instrument. The system designed will monitor the operating parameters of Motor-Pump system and displays the performance parameters like current, voltage, speed, slip, efficiency and torque in motor side, operating head, discharge and overall conversion efficiency of the system.

Contents

CONTENTS

Introduction

1. Introduction

A monitoring system can be defined as a set of devices, procedures and diagnostic tools to track every signal step of a process[1]. Every industrial application requires a suitable monitoring system for its processes in order to identify any decrease in efficiency resulting in economic losses. Early detection of the deviation of the operating conditions from optimality may avoid subsequent faults, or even failures.

The option of combining diagnostic and monitoring operations on an AC Motor-Pump without using dedicated sensors can offer diagnosis as reliable as that provided by totally customized systems. The useful diagnostic indications can be obtained by this low-cost extension of the monitoring activity, and the reliability of the obtained indications can be significantly increased if the combination of advanced transducers and LabVIEW is used.

LabVIEW (Laboratory Virtual Instrument Engineering Workbench)[2],[3] is a development environment based on graphical programming. LabVIEW uses terminology, icons, and ideas familiar to technicians, scientists, and engineers, and relies on graphical symbols rather than textual language to describe programming actions.

Most commonly the motor –pump system[5],[6] are used in the industrial, agricultural and domestic sectors. These pumps operate inefficiently due to the variation in the supply parameters. Hence it is necessary to monitor the parameters to design the suitable controller. The motor pump system monitoring system is designed to monitor electrical parameters of the motor and pump side parameters.

Introduction to Lab VIEW

2. Introduction to Lab VIEW [2],[3]

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a development environment based on graphical programming. LabVIEW uses terminology, icons, and ideas familiar to technicians, scientists, and engineers, and relies on graphical symbols rather than textual language to describe programming actions.

2.1. LabVIEW

LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine program execution, LabVIEW uses dataflow programming, where data determine execution.

In LabVIEW, we build a user interface by using a set of tools and objects. The user interface is known as the front panel. We then add code using graphical representations of functions to control the front panel objects. The block diagram contains this code. If organized properly, the block diagram resembles a flowchart.We can purchase several add-on software toolsets for developing specialized applications. All the toolsets integrate seamlessly in LabVIEW.

LabVIEW is integrated fully for communication with hardware such as GPIB, VXI, PXI, RS-232, RS-485, and plug-in data acquisition devices. LabVIEW also has built-in features for connecting our application to the Internet using the LabVIEW web server and software standards such as TCP/IP networking and ActiveX.Using LabVIEW, we can create 32-bit compiled applications that give us the fast execution speeds needed for custom data acquisition, test, measurement, and control solutions. We also can create stand-alone executables and shared libraries, like DLLs, because LabVIEW is a true 32-bit compiler.

LabVIEW contains comprehensive libraries for data collection, analysis, presentation, and storage. LabVIEW also includes traditional program development tools. We can set breakpoints, animate program execution, and single-step through the program to make debugging and development easier.

LabVIEW also provides numerous mechanisms for connecting to external code or software through DLLs, shared libraries, ActiveX, and more. In addition, numerous add-on tools are available for a variety of application needs.

2.2 USES OF LabVIEW

LabVIEW empowers us to build our own solutions for scientific and engineering systems. LabVIEW gives us the flexibility and performance of a powerful programming language without the associated difficulty and complexity.

LabVIEW gives thousands of successful users a faster way to program instrumentation, data acquisition, and control systems. By using LabVIEW to prototype, design, test, and implement our instrument systems, we can reduce system development time and increase productivity by a factor of 4 to 10.

LabVIEW also gives us the benefits of a large installed user base, years of product feedback, and powerful add-on tools. Finally, National Instruments technical support and Developer Zone ensure successful development of our solutions.

2.3 Design procedure

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. Every VI uses functions that manipulate input from the user interface or other sources and display that information or move it to other files or other computers.

A VI contains the following three components:

- Front panel — Serves as the user interface.
- Block diagram — contains the graphical source code of the VI that defines its functionality.
- Icon and connector pane — identifies the VI so that we can use the VI in another VI. A VI within another VI is called a sub VI. A sub VI corresponds to a subroutine in text - based programming languages.

Front Panel

The front panel is the user interface of the VI. We build the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates.

Block Diagram

After we build the front panel, we add code using graphical representations of functions to control the front panel objects. The block diagram contains this graphical source code. Front panel objects appear as terminals on the block diagram. We cannot delete a terminal from the block diagram. The terminal disappears only after we delete its corresponding object on the front panel.

Every control or indicator on the front panel has a corresponding terminal on the block diagram. Additionally, the block diagram contains functions and structures from built-in LabVIEW VI libraries. Wires connect each of the nodes on the block diagram, including control and indicator terminals, functions, and structures.

Relationship between Front Panel Objects and Block Diagram Terminals

Front panel objects appear as terminals on the block diagram. Double-click a block diagram terminal to highlight the corresponding control or indicator on the front panel. We cannot delete a terminal from the block diagram. The terminal disappears only after we delete its corresponding object on the front panel.

Terminals are entry and exit ports that exchange information between the front panel and block diagram. Data we enter into the front panel controls enter the block diagram through the control terminals. When the VI finishes running, the output data flow to the indicator terminals, where they exit the block diagram, reenter the front panel, and appear in front panel indicators.

Palettes

LabVIEW palettes give us the options we need to create and edit the front panel and block diagram.

Tools Palette

The Tools palette is available on the front panel and the block diagram. A tool is a special operating mode of the mouse cursor. When we select a tool, the cursor icon changes to the tool icon. Use the tools to operate and modify front panel and block diagram objects.

Select Window» Show Tools Palette to display the Tools palette. We can place the Tools palette anywhere on the screen.

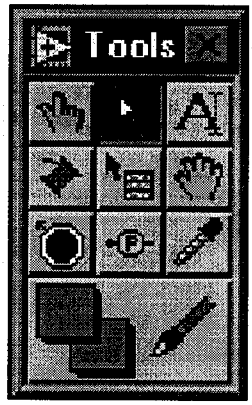


Fig. 2.1 Tools Palette

Controls Palette

The Controls palette is available only on the front panel. The Controls palette contains the front panel controls and indicators we use to create the user interface.

Select Window» Show Controls Palette or right-click the front panel workspace to display the Controls palette. We can place the Controls palette anywhere on screen.

Functions Palette

The Functions palette is available only on the block diagram. The Functions palette contains the VIs and functions you use to build the block diagram.

Select Window»Show Functions Palette or right-click the block diagram workspace to display the Functions palette. You can place the Functions palette anywhere on the screen.

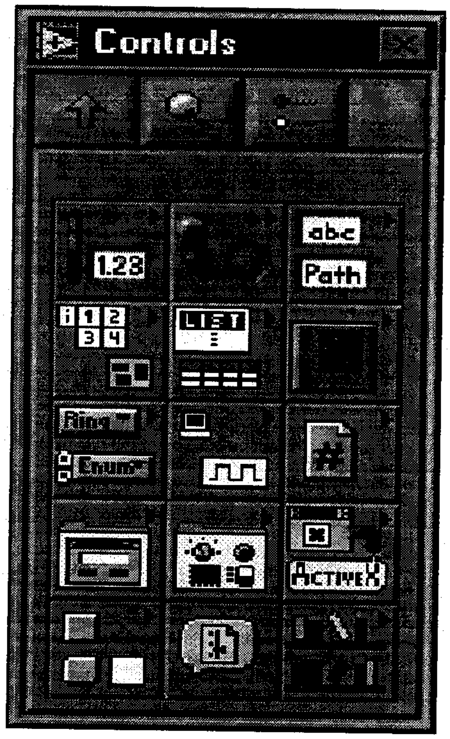


Fig. 2.2 Controls Palette

2.4 History of Instrumentation [4]

As a first step in understanding how instruments are built, consider the history of instrumentation. Instruments have always made use of widely available technology. In the 19th century, the jeweled movement of the clock was first used to build analog meters. In the 1930s, the variable capacitor, the variable resistor, and the vacuum tube from radios were used to build the first electronic instruments. Display technology from the television has contributed to modern oscilloscopes and analyzers. And finally, modern personal computers contribute high-performance computation and display capabilities at an ever-improving performance-to-price ratio.

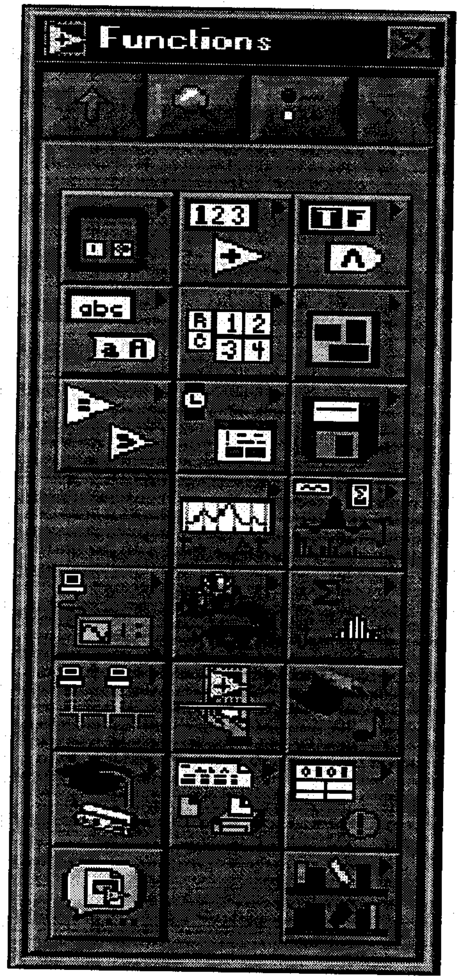


FIG. 2.3 Functions Palette

2.5 Introduction to Measurement and Virtual Instrumentation

We take measurements with instruments. Instrumentation helps science and technology progress. Scientists and engineers around the world use instruments to observe, control, and understand the physical universe. Our quality of life depends on the future of instrumentation—from basic research in life sciences and medicine to design, test and manufacturing of electronics, to machine and process control in countless industries.

Virtual Instrumentation

Virtual Instrumentation is defined as combining hardware and software with industry-standard computer technologies to create user-defined instrumentation solutions. National Instruments specializes in developing plug-in hardware and driver software for data acquisition (DAQ), IEEE 488 (GPIB), VXI, serial, and industrial communications. The driver software is the programming interface to the hardware and is consistent across a wide range of platforms. Application software such as LabVIEW, Lab Windows/CVI, Component Works, and Measure deliver sophisticated display and analysis capabilities required for Virtual Instrumentation.

System Components for Taking Measurements with Virtual Instruments

Different hardware and software components can make up our virtual instrumentation system. Many of these options are described in more detail throughout this manual. There is a wide variety of hardware components we can use to monitor or control a process or test a device. As long as we can connect the hardware to the computer and understand how it makes measurements, we can incorporate it into our system.

Comparing DAQ Devices and Special-Purpose Instruments for Data Acquisition

Measurement devices, such as general-purpose data acquisition (DAQ) devices and special-purpose instruments, are concerned with the acquisition, analysis, and presentation of measurements and other data we acquire.

Acquisition is the means by which physical signals, such as voltage, current, pressure, and temperature, are converted into digital formats and brought into the computer. Popular methods for acquiring data include plug-in DAQ and instrument devices, GPIB instruments, VXI instruments, and RS-232 instruments. Data analysis transforms raw data into meaningful information.

This can involve such things as curve fitting, statistical analysis, frequency response, or other numerical operations. Data presentation is the means for communicating with our system in an intuitive, meaningful format. Building a computer-based measurement system can be a daunting task. There is a wide variety of hardware components we can use to monitor or control a process or test a device. Should we build on traditional rack-and-stack IEEE 488 equipment or look to modular VXI-based solutions? Or maybe we should consider a PC-based plug-in board approach.

DAQ Devices versus Special-Purpose Instruments

The fundamental task of all measurement systems is the measurement and/or generation of real-world physical signals. The primary difference between the various hardware options is the method of communication between the measuring hardware and the computer. In this chapter we will separate the discussion into two categories: general purpose DAQ devices and special purpose instruments. General purpose DAQ devices are devices that connect to the computer allowing the user to retrieve digitized data values. These devices typically connect directly to the computer's internal bus through a plug-in slot. Some DAQ devices are external and connect to the computer via serial, GPIB, or Ethernet ports. The primary distinction of a test system that utilizes general purpose DAQ devices is where measurements are performed. With DAQ devices, the hardware only converts the incoming signal into a digital signal that is sent to the computer.

The DAQ device does not compute or calculate the final measurement. That task is left to the software that resides in the computer. The same device can perform a multitude of measurements by simply changing the software application that is reading the data. So, in addition to controlling, measuring, and displaying the data, the user application for a computer-based DAQ system also plays the role of the firmware—the built-in software required to process the data and calculate the result—that would exist inside a special purpose instrument. While this flexibility allows the user to have one hardware device for many types of tests, the user must spend more time developing the different applications for the different types of tests. Fortunately, LabVIEW comes with many acquisition and analysis functions to make this easy.

Instruments are like the general purpose DAQ device in that they digitize data. However, they have a special purpose or a specific type of measurement capability. The software, or firmware, required to process the data and calculate the result is usually built in and cannot be modified. For example, a multi-meter can not read data the way an oscilloscope can because the program that is inside the multi-meter is permanently stored and cannot be changed dynamically. Most instruments are external to the computer and can be operated alone, or they may be controlled and monitored through a connection to the computer. The instrument has a specific protocol that the computer must use in order to communicate with the instrument. The connection to the computer could be Ethernet, Serial, GPIB, or VXI. There are some instruments that can be installed into the computer like the general purpose DAQ devices. These devices are called computer-based instruments.

2.6. Interfacing DAQ Devices with PC

Before a computer-based system can measure a physical signal, a sensor or transducer must convert the physical signal into an electrical one, such as voltage or current. The plug-in DAQ device is often considered to be the entire DAQ system, although it is actually only one system component. Unlike most stand-alone instruments, we cannot always directly connect signals to a plug-in DAQ device. In these cases, we must use accessories to condition the signals before the plug-in DAQ device converts them to digital information.

The software controls the DAQ system by acquiring the raw data, analyzing the data, and presenting the results. FIG 2.4 shows two options for a DAQ system. In Option A, the plug-in DAQ device resides in the computer. In Option B, the DAQ device is external. With an external board, we can build DAQ systems using computers without available plug-in slots, such as some laptops. The computer and DAQ module communicate through various buses such as the parallel port, serial port, and Ethernet. These systems are practical for remote data acquisition and control applications.

DAQ System Components

A third option, not shown in FIG. 2.4, uses the PCMCIA bus found on some laptops. A PCMCIA DAQ device plugs into the computer, and signals are connected to the board just as they are in Option A. This allows for a portable, compact DAQ system.

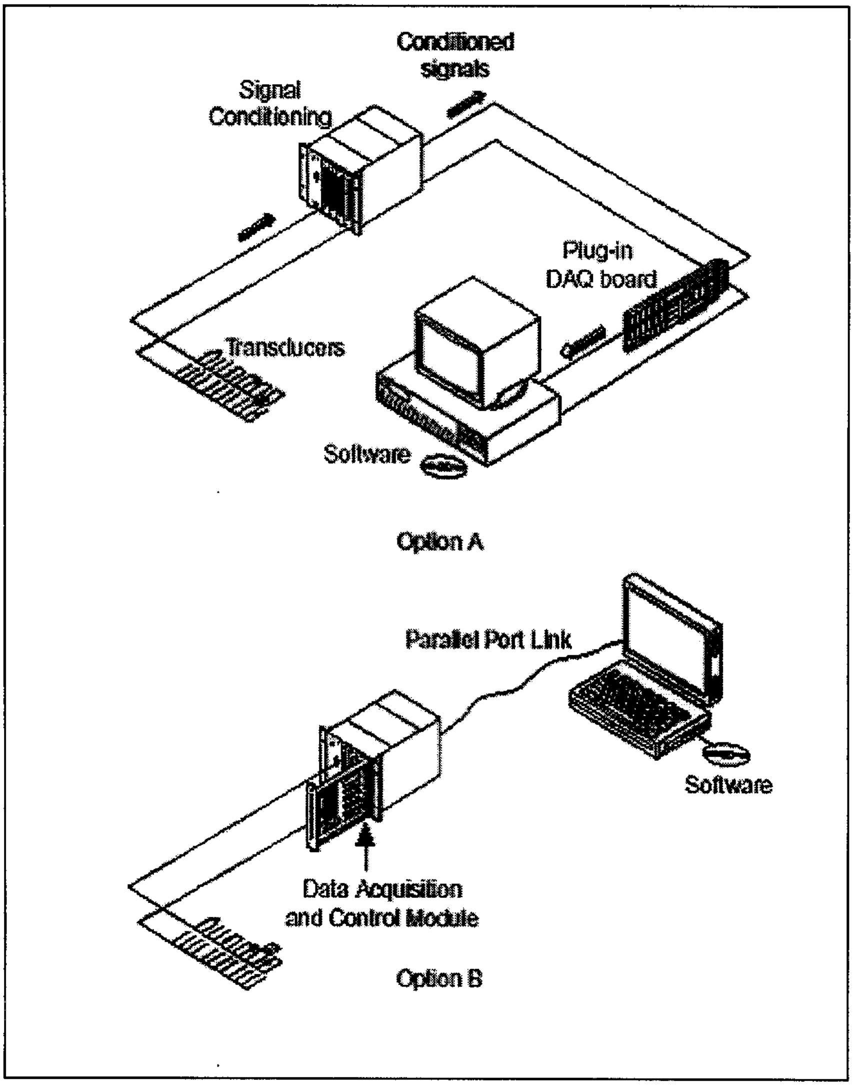


Fig. 2.4 DAQ System Components

Role of Software

The computer receives raw data. Software takes the raw data and presents it in a form the user can understand. Software manipulates the data so it can appear in a graph or chart or in a file for report. The software also controls the DAQ system, telling the DAQ device when to acquire data, as well as from which channels to acquire data.

Typically, DAQ software includes drivers and application software. Drivers are unique to the device or type of device and include the set of commands the device accepts. Application software (such as LabVIEW) sends the commands to the drivers, such as acquire a thermocouple reading and return the reading, then displays and analyzes the data acquired. LabVIEW includes a set of VIs that let we configure, acquire data from, and send data to DAQ devices. LabVIEW DAQ VIs makes call to the NI-DAQ Application Program Interface (API). The NI-DAQ API contains the tools and basic functions that interface to DAQ hardware.

Defining our Signal

Analog signals can be grouped into three categories: DC, time domain, and frequency domain. FIG. 2.5 illustrates which signals correspond to certain types of signal information.

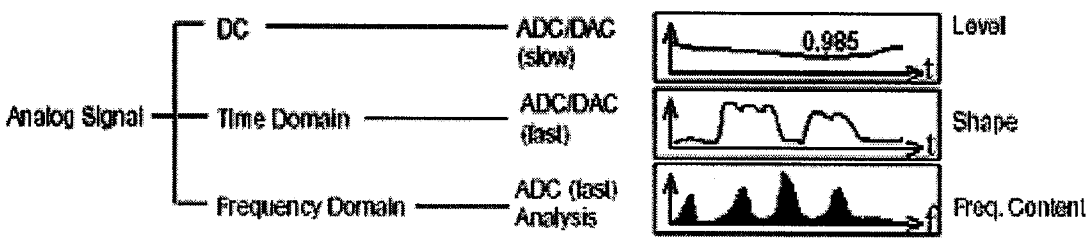


Fig. 2.5 Types of Analog Signal

We must define a few more signal characteristics before we can begin measuring. For example, to what is our signal referenced? How fast does the signal vary over time? We can treat a DC signal as a form of time domain signal. With a slowly-varying signal, we often can acquire a single point for our measurement. However, some DC signals might have noise, which varies quickly for time and frequency domain signals, we acquire several points of data at a fast scan rate.

The rate we sample determines how often the analog-to-digital conversions take place. A fast sampling rate acquires more points in a given time and, therefore, can often form a better representation of the original signal than a slow sampling rate. The sampling rate we should use depends on the types of features we are trying to find in our waveform. For example, if we are trying to detect a quick pulse in the time domain, we must sample fast enough that we do not miss the pulse. The time between successive scans must be smaller than the pulse period. If we are interested in measuring the rise time of a pulse, we must sample at an even faster rate, which depends on how quickly the pulse rises

If we are measuring frequency characteristics of a waveform, we often do not need to sample as fast as we do for time domain measurements. According to the Nyquist Theorem, we must sample at a rate greater than twice the maximum frequency component in a signal to get accurate frequency information about that signal. This is usually not a fast enough rate to recreate the shape of the signal in the time domain, but it does record the frequency information. The frequency at one half the sampling frequencies is referred to as the Nyquist frequency. Signals come in two forms: referenced and non-referenced signal sources.

More often, referenced sources are said to be grounded signals, and non-referenced sources are called floating signals.

Considerations for Selecting Analog Input Settings

The resolution and device range of a DAQ device determine the smallest detectable change in the input signal. We can calculate the smallest detectable change, called the code width, using the following formula.

$$\text{Code width} = \frac{\text{Device range}}{2^{\text{resolution}}}$$

2.7. Measuring Temperature

A RTD is formed with a temperature sensitive metal like platinum whose resistance changes with change in temperature. This section examines a simple approach to measuring temperature using a RTD.

In this example, we will learn how to measure a single temperature value following the diagram of FIG. 2.6.

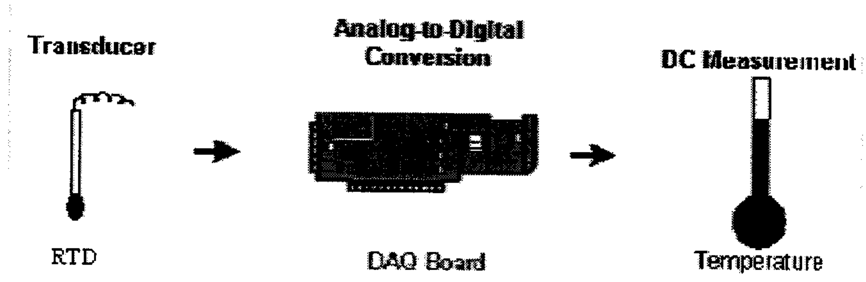


FIG. 2.6 Simple Temperature Measurement System

FIG. 2.7 shows the block diagram needed to measure the temperature if we are using DAQ Named Channels. In this case, the DAQ Named Channel handles all gain, and linearization required.

Fig. 2.7 Measuring Temperature Using DAQ Named Channel

If we do not want to use DAQ Named Channels to measure temperature, we must write a VI that determines the gain needed for our temperature range, read the RTD voltage and convert this information into a temperature.

2.8 Basic Lab VIEW Data Acquisition concepts

This section explains how data acquisition works with LabVIEW. Before we start building our data acquisition (DAQ) application, we should know some of the following basic LabVIEW DAQ concepts

- Where to find common DAQ examples?
- Where to find the DAQ VIs in LabVIEW?
- How the DAQ VIs are organized?
- Polymorphic DAQ VIs.
- VI parameter conventions.
- Default and current value conventions.
- The Waveform Control.
- Channel, port, and counter addressing.
- Limit settings.
- Other DAQ VI parameters.
- How DAQ VIs handle errors?
- Organization of analog data.

Channel and Port Addressing

The Analog Input and Analog Output VIs have a channels parameter where we can specify the channels from which the VIs read or to which they write. The Digital Input and Output VIs have a similar parameter, called digital channel list, and the equivalent value is called counter list for the Counter VIs.

Each channel we specify in the channels parameter becomes a member of a group. For each group, we can acquire or generate data on the channels listed in the group. VIs scan (during acquisition) or update (during generation) the channels in the same order they are listed. To erase a group, pass an empty channels parameter and the group number to the VI or assign a new channels parameter to the group. We can change groups only at the Advanced VI level.

DAQ Channel Name Control

The channels parameter in the Analog and Digital VIs is a DAQ Channel Name control. If we configured our channels in the DAQ Channel Wizard, the controls menu lists the channel names of our configured channels. Select a channel name from the menu or type a channel name or number into the string area of the control.

Channel Name Addressing

If we use the DAQ Channel Wizard to configure our analog and digital channels, we can address our channels by name in the channels parameter in LabVIEW. Channels can be an array of strings or, as with the Easy VIs, a scalar string control. If we have a channels array, we can use one channel entry per array element, specify the entire list in a single element, or use any combination of these two methods. If we enter multiple channel names in channels, we must configure all of the channels in the list for the same DAQ device. If we configure channels with names of temperature and pressure, both of which are measured by the same DAQ device, we can specify a list of channels in a single element by separating them by commas—for example, temperature, and pressure. If we configure channels with names of temp1, temp2, and temp3, we can specify a range of channels by separating them with a colon, for example, temp1:temp3. In specifying channel names, spelling and spaces are important, but case is not. Channel Controls When using channel names, we do not need to wire the device, input limits, or configure input parameters. LabVIEW always ignores the device input when using channel names. LabVIEW configures our hardware in terms of our channel configuration. In addition, LabVIEW orders and pads the channels specified in channels as needed according to any special device requirements.

Channel Number Addressing

If we are not using channel names to address our channels, we can address our channels by channel numbers in the channels parameter. The channels can be an array of strings or as with the easy VIs, a scalar string control. If we have a channels array, we can use one channel entry per array element, specify the entire list in a single element, or use any combination of these two methods. For instance, if our channels are 0, 1, and 2, we can specify a list of channels in a single element by separating the individual channels by commas—for example, 0, 1, and 2. Or we can specify the range by separating the first and last channels with a colon—for example, 0:2. Channel String Array Controls Some Easy and Advanced Digital VIs and Intermediate Counter VIs allow only one port or counter to be specified. Limit Settings Limit settings are the maximum and minimum values of the analog signal(s) we are measuring or generating. The pair of limit setting values can be unique for each analog input or output channel. For analog input applications, the limit setting values must be within the range for the device. Each pair of limit setting values forms a cluster.

Analog output limits have a third member, the reference source. For simplicity, LabVIEW refers to limit settings as a pair of values. LabVIEW uses an array of these clusters to assign limits to the channels in our channel string array. If we use the DAQ Channel Wizard to configure our analog input channels, the physical unit we specified for a particular channel name is applied to the limit settings. For example, if we configured a channel in the DAQ Channel Wizard to have physical units of Deg C, the limit settings are treated as limits in degrees Celsius. LabVIEW configures our hardware to make the measurement in terms of our channel name configuration.

Unless we need to overwrite our channel name configuration, do not wire this input. Allow LabVIEW to set it up for us. If we are not using the DAQ Channel Wizard, the default unit applied to the limit settings is usually volts, although the unit applied to the limit settings can be volts, current, resistance, or frequency, depending on the capability and configuration of our device. The default range of the device, set in the configuration utility or by LabVIEW according to the channel name configuration in the DAQ Channel Wizard, is used whenever we leave the limit settings terminal unwired or we enter 0 for our upper and lower limits. LabVIEW uses an array of strings to specify which channels belong to a group. Also, remember LabVIEW lists one channel to as many as all of the device channels in a single array element in the channels array. LabVIEW also assigns all the channels listed in a channels array element the same settings in the corresponding limit settings cluster array element.

Other DAQ VI Parameters

The device input on analog I/O, digital I/O, and counter VIs specifies the number the DAQ configuration software assigned to our DAQ device. Our software assigns a unique number to each DAQ device. The device parameter usually appears as an input to the configuration VIs. Another common configuration VI parameter, task ID, assigns our specific I/O operation and device a unique number that identifies it throughout our program flow. Some DAQ VIs performs either the device configuration or the I/O operation, while other DAQ VIs performs both configuration and the operation. Some of the VIs that handles both functions has an iteration input. When our VI has the iteration set to 0, LabVIEW configures the DAQ device and then performs the specific I/O operation. For iteration values greater than 0, LabVIEW uses the existing configuration to perform the I/O operation. We can improve the performance of our application by not configuring the DAQ device every time an I/O operation occurs.

2.9 Error Handling

Each Easy VI contains an error handling VI. A dialog box appears immediately if an error occurs in an Easy VI. Each Intermediate and Advanced VI contains an error in input cluster and an error out output cluster, as shown in FIG. 7. The clusters contain a Boolean indicator that indicates whether an error occurred, the code for the error, and source or the name of the VI that returned the error. If error in indicates an error, the VI passes the error information to error out and does not execute any DAQ functions.

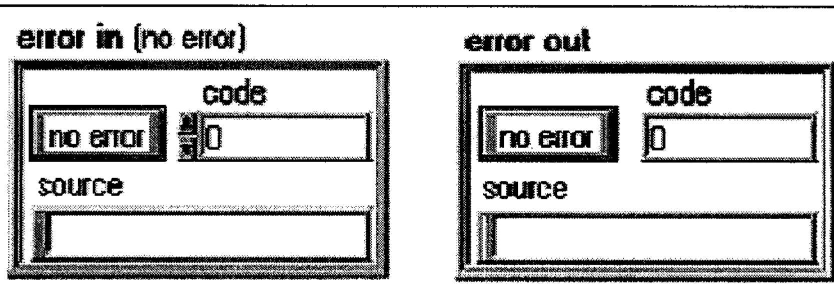


Fig. 2.8 Error In and Error out Clusters

2.10 Lab VIEW-based applications

Ultrasonic Measurement System for Nondestructive Evaluation of Advanced Structural Materials:

- The LabVIEW-based ultrasonic measurement system was designed to facilitate high-frequency (1 to 150 MHz) ultrasonic characterization of the internal structure of materials. Current materials the system is characterizing include those being developed for advanced high-temperature aero propulsion applications under supervision of NASA.
- The velocity and attenuation of ultrasonic waves are dependent upon both material type (chemical composition) and microstructure (number of voids/pores, grain size).
- Since material microstructure impacts physical properties
- (For example, stiffness, strength, electrical conduction and so forth, ultrasonic measurements can to some extent be used to predict the behavior of materials in service.
- The ultrasonic measurement system works as follows. A piezoelectric-based ultrasonic transducer is placed on top of the material sample to be characterized. The transducer is electrically excited by a voltage from a pulsar and
- Transforms the voltage into a high frequency mechanical wave (vibration).
- The wave is transmitted to and into the material sample where it bounces off of the sample surfaces. The various bounces or echoes return to the same
- Transducer where they are converted from mechanical back to electrical form. The echoes are sent to a PC-based 1 GHz analog-to-digital (A/D) converter for digitization. The digitized waveform is output from the A/D converter for digital processing and display.

- The LabVIEW program performs the major functions necessary to control the A/D converter (via a custom software driver).
- In addition, the software is responsible for analyzing the measurement data generated by the experiment, as well as displaying and storing the results of the analysis.

Single-phase motor-pump system

3. Single-phase motor-pump system

Single phase motors are designed to operate from a single-phase supply are manufactured in a large number of types to perform a wide variety of useful services in home, offices, workshops and in business establishment's etc. since the performance requirements of the various applications differ so widely, the motor manufacturing industry has developed many different types of such motors, each being designed to meet specific demands.

Single phase motors can be classified as under depending on their construction and method of starting:

- Induction Motors (split-phase, capacitor and shaded-pole etc.)
- Repulsion Motors (sometime called inductive-series motors)
- A.C. Series Motors
- Unexcited Synchronous Motors.

3.1 Types of capacitor-start Motors

- Single-voltage, externally-reversible type
- Single-voltage, non-reversible type
- Single-voltage reversible and with thermostat type
- Single voltage, non-reversible with magnetic switch type
- Two-voltage, non-reversible type
- Two-voltage, reversible type
- Single-voltage, three-lead reversible type
- Single-voltage, instantly-reversible type
- Two-speed type
- Two-speed with two-capacitor type

Capacitor Start and Run motor

This motor is similar to the capacitor-start motor except that the starting winding and capacitor are connected in the circuit at all times. The advantages of leaving the capacitor permanently in circuit are

- Improvement of over-load capacity of the motor
- A higher power factor
- Higher efficiency
- Quieter running of the motor which is so much desirable for small power drives in offices and laboratories

Some of these which start and run with one value of capacitance in the circuit are called *single-value* capacitor-run motors. Others which start with high value of capacitance but run with a low value of capacitance are known as *two-value* capacitor-run motors.

Single-value capacitor-run motor

It has one running winding and one starting winding in series with a capacitor. Since capacitor remains in the circuit permanently, this motor is often referred to as permanent-split capacitor-run motor and behaves practically like an unbalanced 2-phase motor. Obviously, there is no need to use a centrifugal switch which was necessary in the case of capacitor-start motors. Since the same capacitor in used for starting and running, its obvious that neither optimum starting nor optimum running performance can be obtained because value of capacitance used must be a compromise between the best value for starting and that for running. Generally capacitors of 2 to 20μF capacitance are employed and are more expensive oil or pyranol-insulated foil-paper capacitors because of continuous-duty rating. The low value of the capacitor results in small starting torque which is about 50 to 100 percent of the rated torque. Consequently, these motors are used where the required starting torque is low such as air-moving equipment i.e. fans, blowers and voltage regulators and also oil burners where quiet operation is particularly desirable.

One unique feature of this type of motor is that it can be easily reversed by an external switch provided its *running and starting windings are identical.* One serves as the running winding and the other as a starting winding for one direction of rotation. For reverse rotation the one that previously served as a running winding becomes the starting winding while the former starting winding serves as the running winding. Such reversible motors are often used for operating devices that must be moved back and forth very frequently such as rheostats, induction regulators, furnace controls, valves and arc-welding controls.

Two-valve capacitor-run motor

This motor starts with a high capacitor in series with the starting winding so that the starting torque is high, for running a lower capacitor is substituted by the centrifugal switch. Both the running and starting windings remain in the circuit.

The two values of capacitance can be obtained as follows:

- By using two capacitors in parallel at the start and then switching out one for low-value run
- By using a step-up auto-transformer in conjunction with one capacitor so that the effective capacitance value is increased for starting purposes

At the start when the centrifugal switch is closed, the two capacitors are put in parallel so that their combined capacitance is the sum of their individual capacitances. After the motor has reached 75 percent of full load speed, the switch opens and only one capacitor remains in the starting winding circuit. In this way, both optimum starting and running performance is achieved in such motors. If properly designed, such motors have operating characteristics very closely resembling those of two-phase motors.

Their performance is characterized by

- Ability to start heavy loads
- Extremely quiet operation
- Higher efficiency and power factor
- Ability to develop 25 percent overload capacity

Hence such motors are ideally suited where the load requirements are severe as in the case of compressors and fire strokes etc.

3.2 Centrifugal Pumps [5],[6]

The hydraulic machines which convert the mechanical energy into hydraulic energy are called pumps the hydraulic energy is in the form of pressure energy. If the mechanical energy is converted into pressure energy by means of centrifugal force acting on the fluid, the hydraulic machine is called centrifugal pump.

The centrifugal pump acts as a reversed of an inward radial flow reaction turbine. This means that the floe in centrifugal pumps is in radial outward directions. The centrifugal pump works on the principle of forced vortex flow which means that when a certain mass of liquid is rotated by an external torque, the rise in pressure head of a rotating liquid takes place. The rise in pressure head at any point of the rotating liquid is proportional to the square of tangential velocity of the liquid at that point. Thus the outlet of the impeller where radius is more, the rise in pressure head will be more and the liquid will be discharged at the outlet with a high pressure head. Due to this high pressure head, the liquid can be lifted to a high level.

Main parts of a centrifugal pump

- Impeller
- Casing
- Suction pipe with a foot valve and a strainer
- Delivery pipe

Design and implementation

4. Design and implementation

A online monitoring system requires continues monitoring of all the required input signals, which may be analog and digital. In this project" A virtual Instrumentation based application for AC Motor –Pump monitoring system" has been designed and implemented in LabVIEW 6.1 for 0.5 kW, 230V, 2880 rpm, 50Hz, single phase Motor-Pump system and the implementation is shown appendix I. The system architecture is shown in fig 4.1The method is based on digital processing of the line signals acquiring by means of a virtual instrument. The system designed will monitor the operating parameters of Motor-Pump system and displays the performance parameters like current, voltage, speed, slip, efficiency and torque in motor side, operating head, discharge and overall conversion efficiency of the system.

The Performance parameters include Efficiency, Torque, and Slip of induction motor and for centrifugal pump overall efficiency is analysed and the design of performance calculation is shown in fig 4.2. and 4.3 The line current drawn by the induction motor is analysed by monitoring the current drawn by the main winding and the starting winding, which is essential in designing the resistance and the inductance of both the windings, and the capacitance of start – run capacitor.

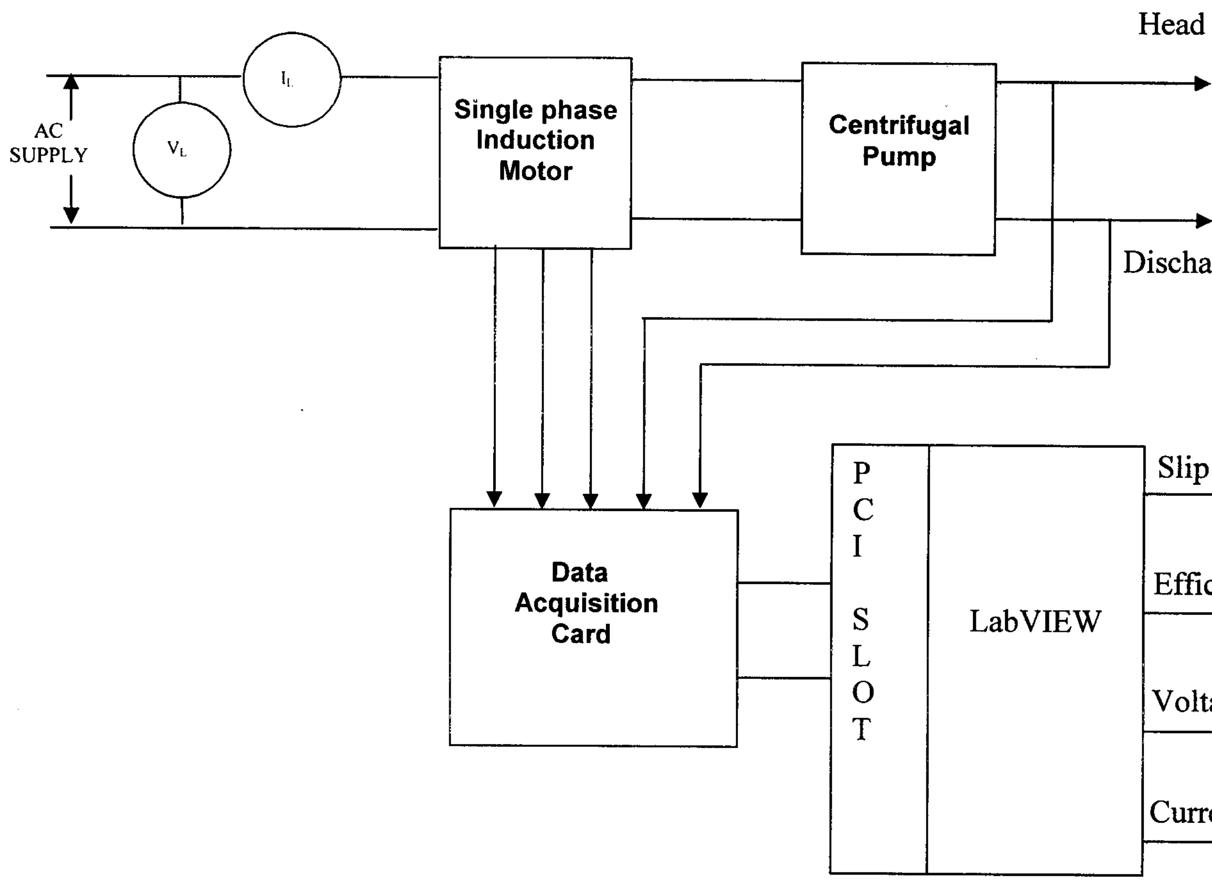


Fig 4.1 Motor –pump system monitoring

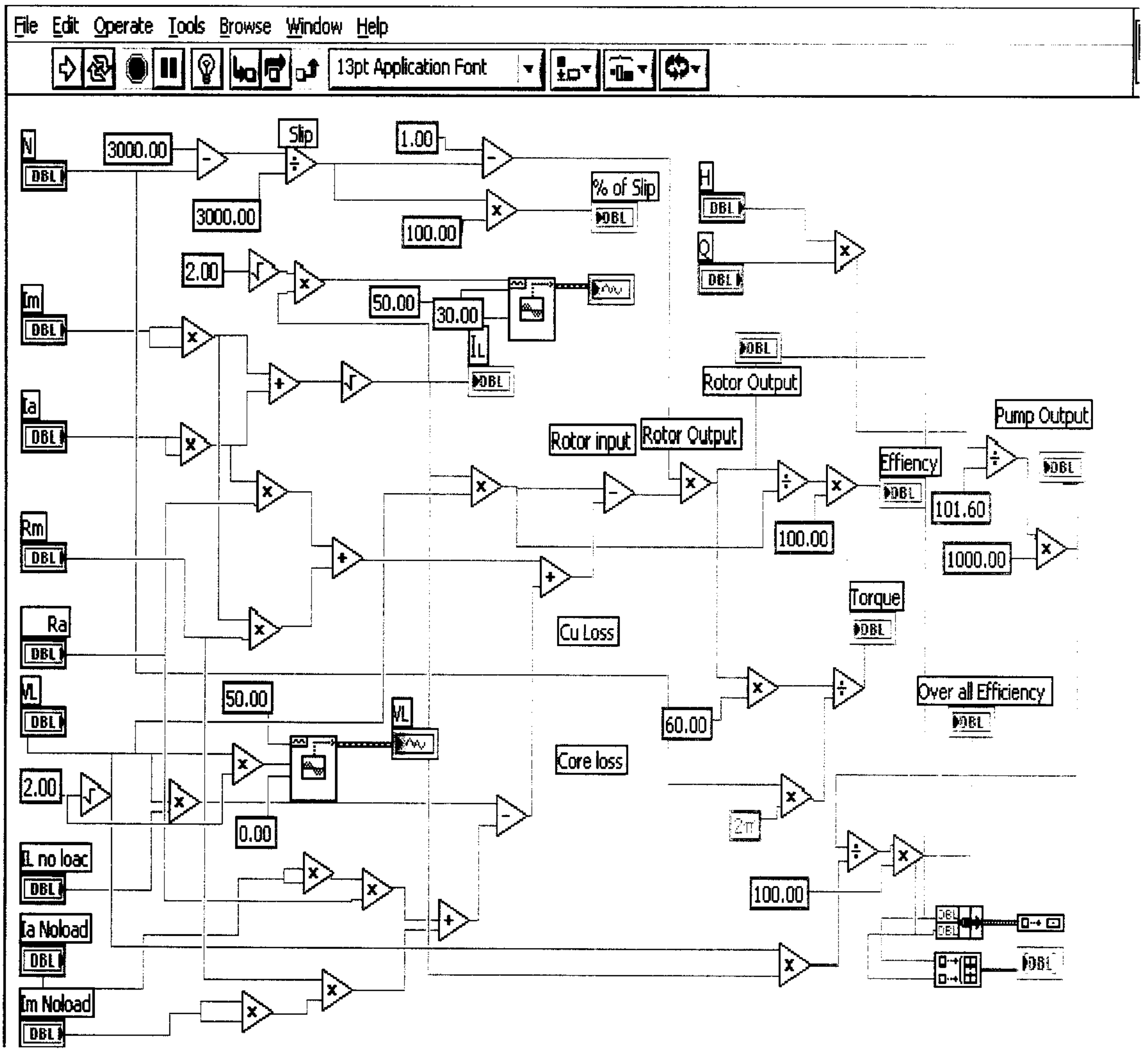


Fig 4.2 Block diagram of on line monitoring and Performance calculation

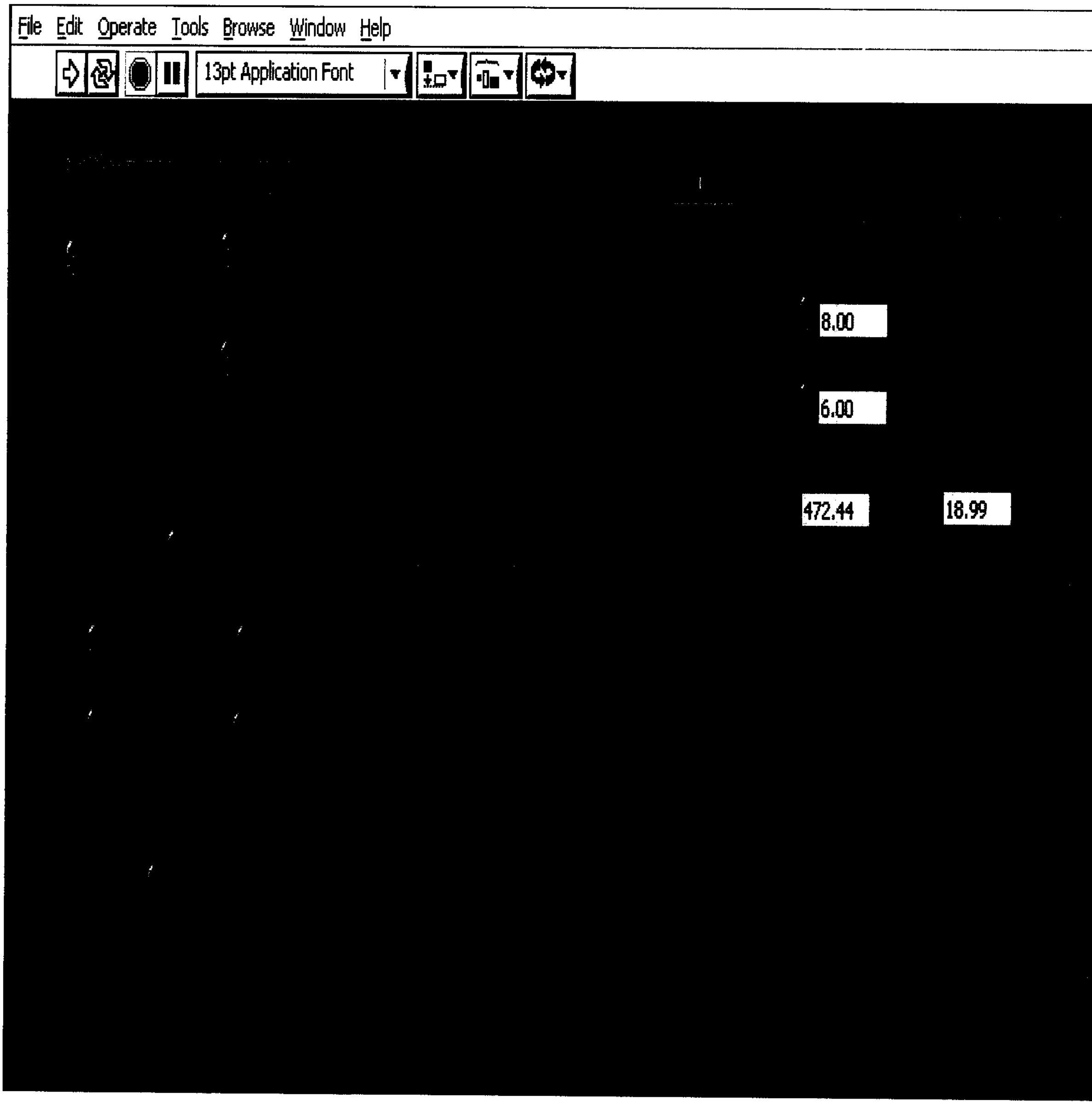


Fig 4.3 Performance Monitoring system- front panel

4.1. System hardware architecture

The single phase induction motor is started using a DOL starter and it drives the centrifugal pump. The input signals to be monitored are the line voltage, current through the main and auxiliary winding, the speed at which both the motor and centrifugal pump runs and the head and discharge of the pump. These input signals must be properly attenuated before they are connected to the DAQ (data acquisition) card. The DAQ card accepts a signal of maximum of ±10V and minimum of ±0.5V. The line voltage is attenuated to 6V using a step down potential transformer of turns ratio 38:1. A current transformer of turns ratio 1:25 is used to reduce the main winding current I_m and the starting winding current I_a. The speed of the set is converted as dc voltage using tacho generator. Current transmitters are used to monitor the head and discharge of the centrifugal pump. These input parameters are changed to 4 to 20mA and are converted to equivalent voltage using proper shunt. All the 6 input parameters are connected to the DAQ card via voltage probes. DAQ card converts all the analog quantities to equivalent digital quantities.

The LabVIEW interacts with the external world only through the DAQ card. The Peripheral Component Interconnect (PCI) bus (or slot) is used to connect the DAQ card, which feeds the attenuated input signals of the motor – pump set with the LabVIEW. The monitoring system should be programmed to recalculate the original values of the input signals from the attenuated signal. The acquired input signals must be multiplied by exact values of the turns ratio by which it is attenuated. The LabVIEW calculates the line current drawn from the values of the starting and running winding current, core loss and copper loss of the starting winding and running winding. Also it computes the total power consumed by the motor- pump set. The attenuated output voltage of the tacho generator is multiplied with suitable constant to read the correct speed of the set. For the two motor the slip is calculated and is indicated in the front panel also. The torque and the percentage efficiency of the single phase induction motor are calculated and also indicated in the front panel.

The centrifugal pump parameters, head and the discharge are measured using properly designed current transmitters which gives an output current of 4 – 20mA in accordance with the change in input. This current is converted to equivalent voltage, tolerable for DAQ card. The LabVIEW computes the out power and the overall efficiency of the motor-pump set. All the performance parameters of induction motor and the pump are shown in the front including no load current and the resistance of starting and running winding. The performance parameters are continuously plotted in wave chart.

4.2 Common device interfaces

Central to the implementation is the use of a common device interface for all virtual instrument components. All components respond to the same small set of commands-only the configuration parameter sets vary. Since configurations are part of the observation sequences the OCS can now pass configurations to the appropriate virtual instruments as needed, without having to understand the specific components within those instruments. Knowledge of the component configurations is isolated in the scripts, user interfaces and in the components themselves. Devices also implement a common state transition model. Commands move devices between states; actions are performed while in the *busy* state.

Results and Conclusion

5 Results and Conclusion

This project "A virtual Instrumentation based application for AC Motor – Pump monitoring system" has been designed and implemented in LabVIEW 6.1 for 0.5 kW, 230V, 2880 rpm, 50Hz, single phase Motor-Pump system. The method is based on digital processing of the line signals acquiring by means of a virtual instrument. The real time parameters of the motor and pump has been acquired using transducers and monitored and are shown in Appendix. II The system designed monitors the operating parameters of Motor-Pump system and displays the performance parameters like current, voltage, speed, slip, efficiency and torque in motor side, operating head, discharge and overall conversion efficiency of the system.

Bibliography

Bibliography

[1]. Loredana Cristaldi et al ," A Neurofuzzy application for AC motor drives monitoring systems" IEEE transactions on Instrumentation and Measurement, Vol 53, N0.4, pp-1020-1027, Aug 2004.

[2].User manual, National Instruments corporation, 2003.

[3]. www.ni.com

[4]. A.K.Sawhney "A course in Electrical and Electronics Measurements and Instrumentation "Dhanpatrai and sons, New Delhi, 2004.

[5]. R.K.Bansal," A Text Book of Fluid Mechanics and Hydraulic Machines" Laxmi Publications (p) ltd., New Delhi, 2002.

[6].G.K.Sakhu , "Pumps" , New Age International (p) Limited , Publishers , New Delhi.

Appendix

Appendix - Coding and Results

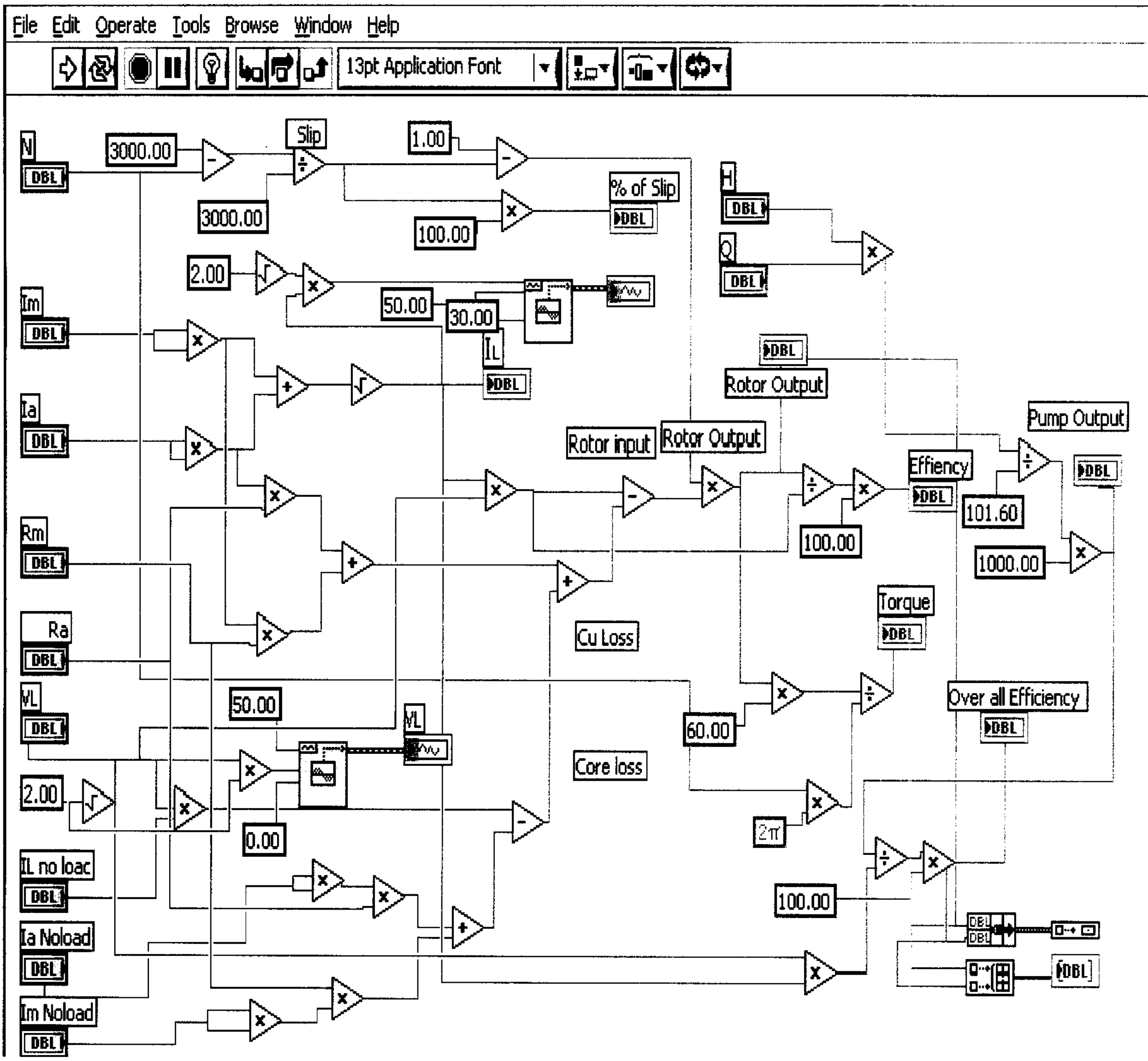


Fig 1 Block diagram of off line monitoring and Performance calculation

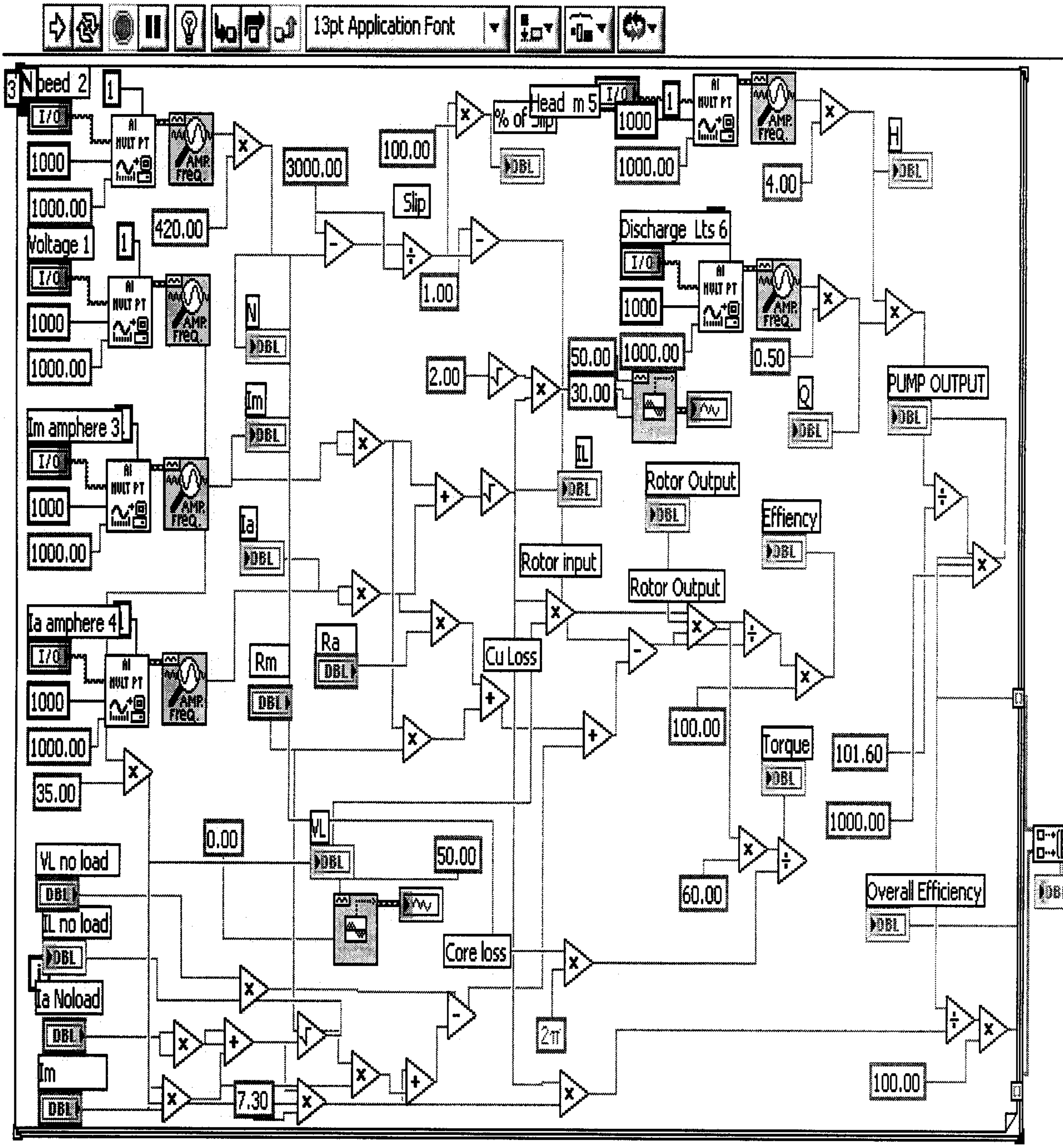


Fig 2 Block diagram of on line monitoring and Performance calculation

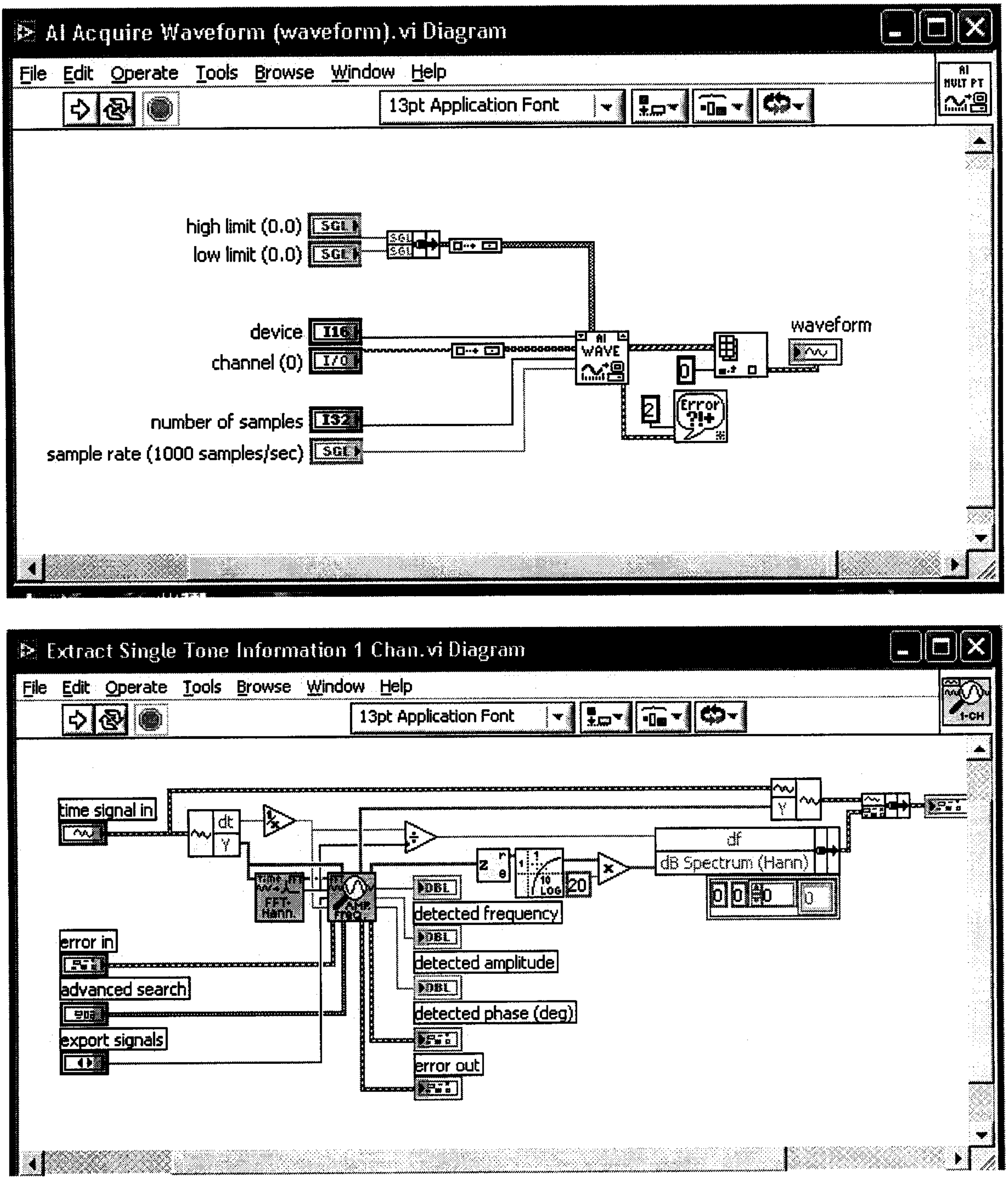
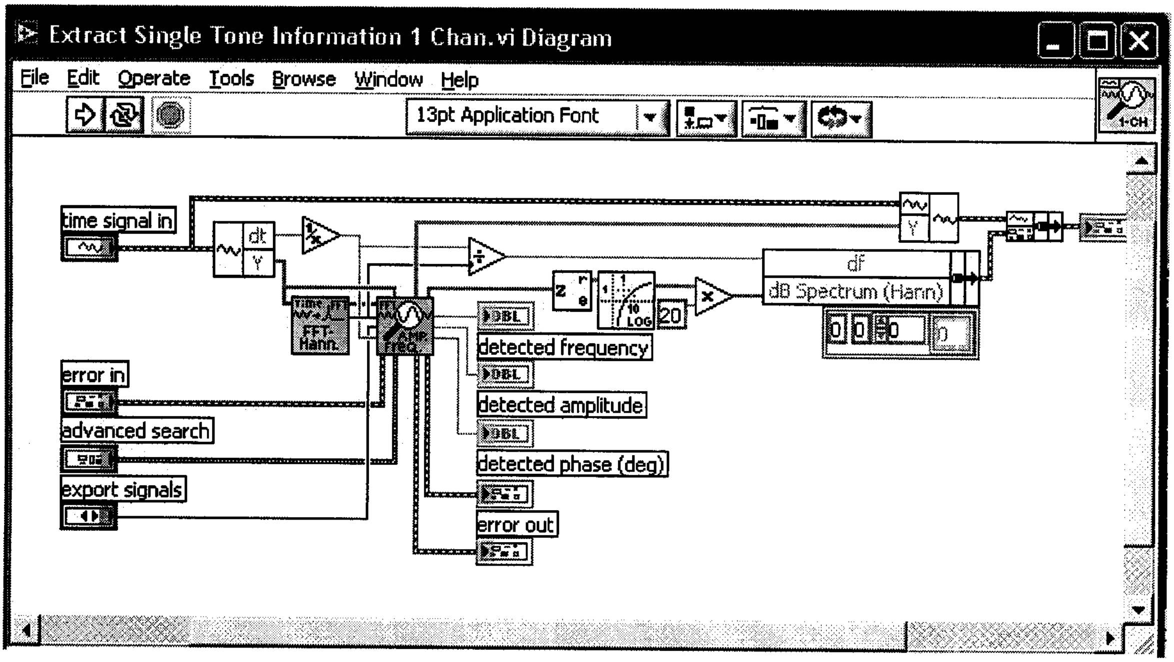


Fig 3 Block Diagram of single Tone Waveform Generator



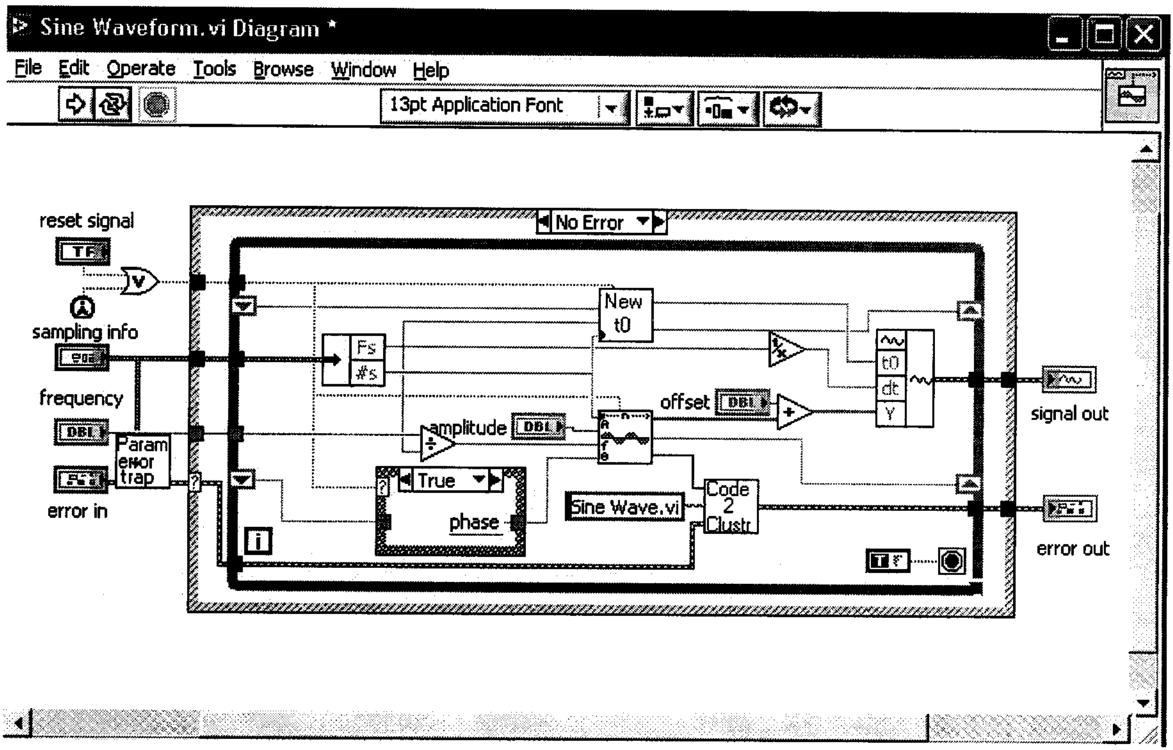


Fig 4 Block diagram Sine Waveform Generator

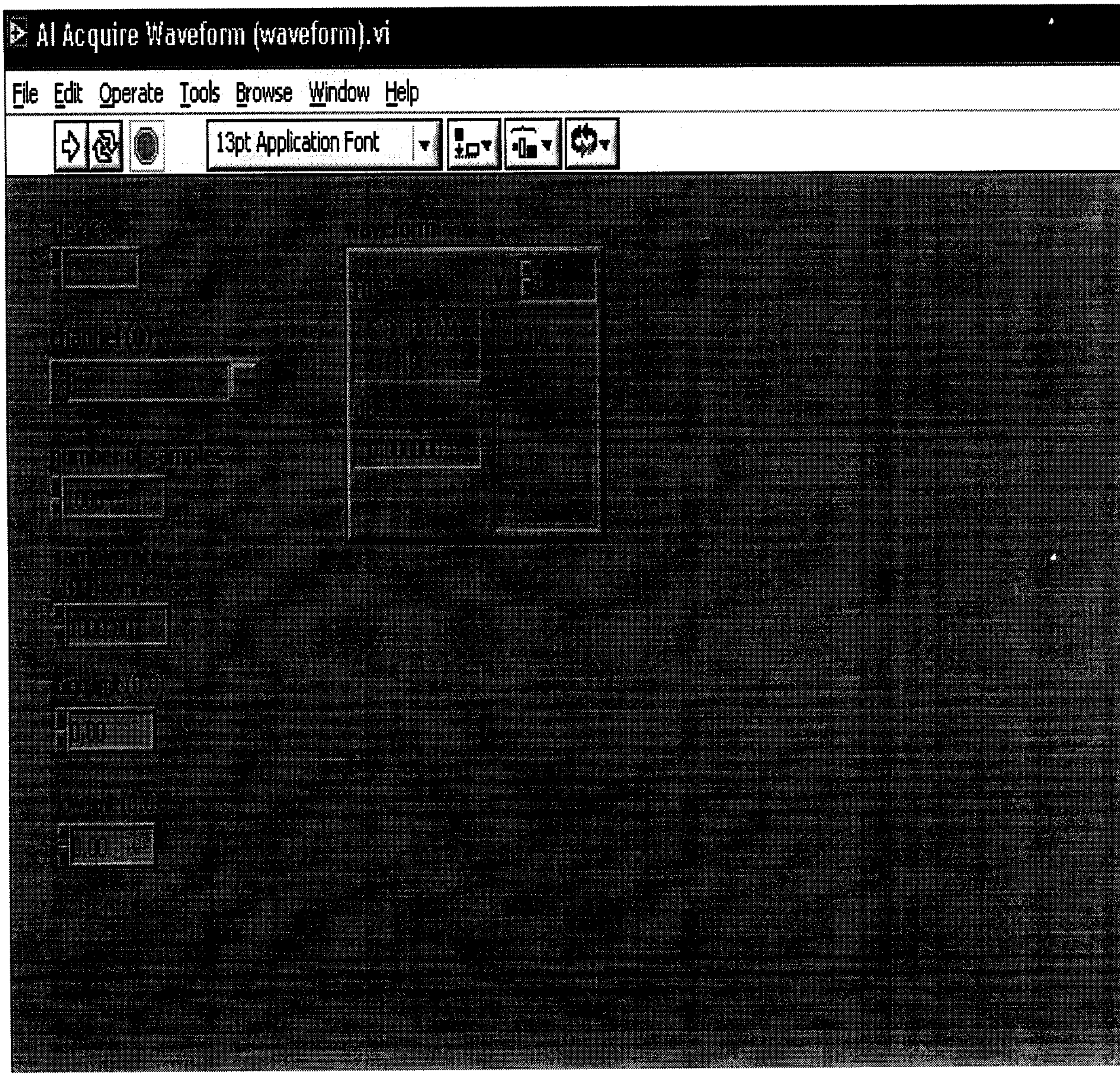


Fig .5 Front panel - AI Acquire waveform

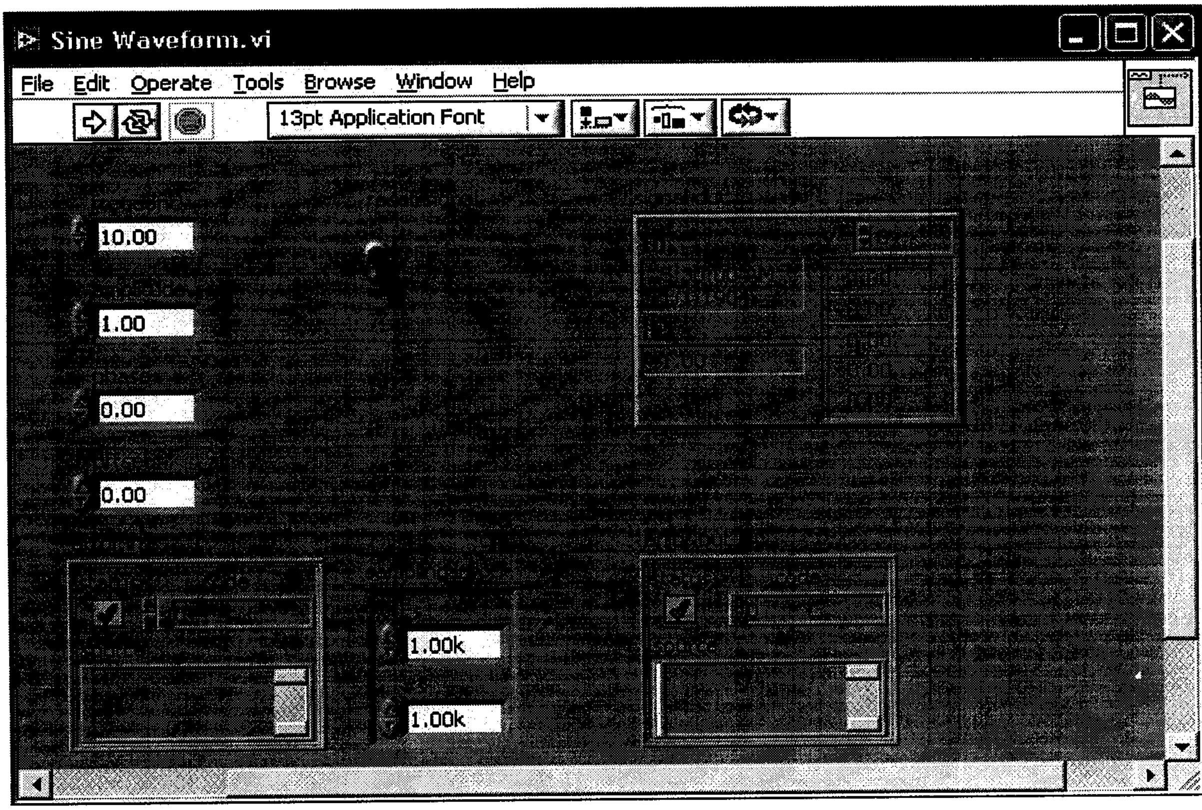


Fig .6 Front panel - Sine Waveform

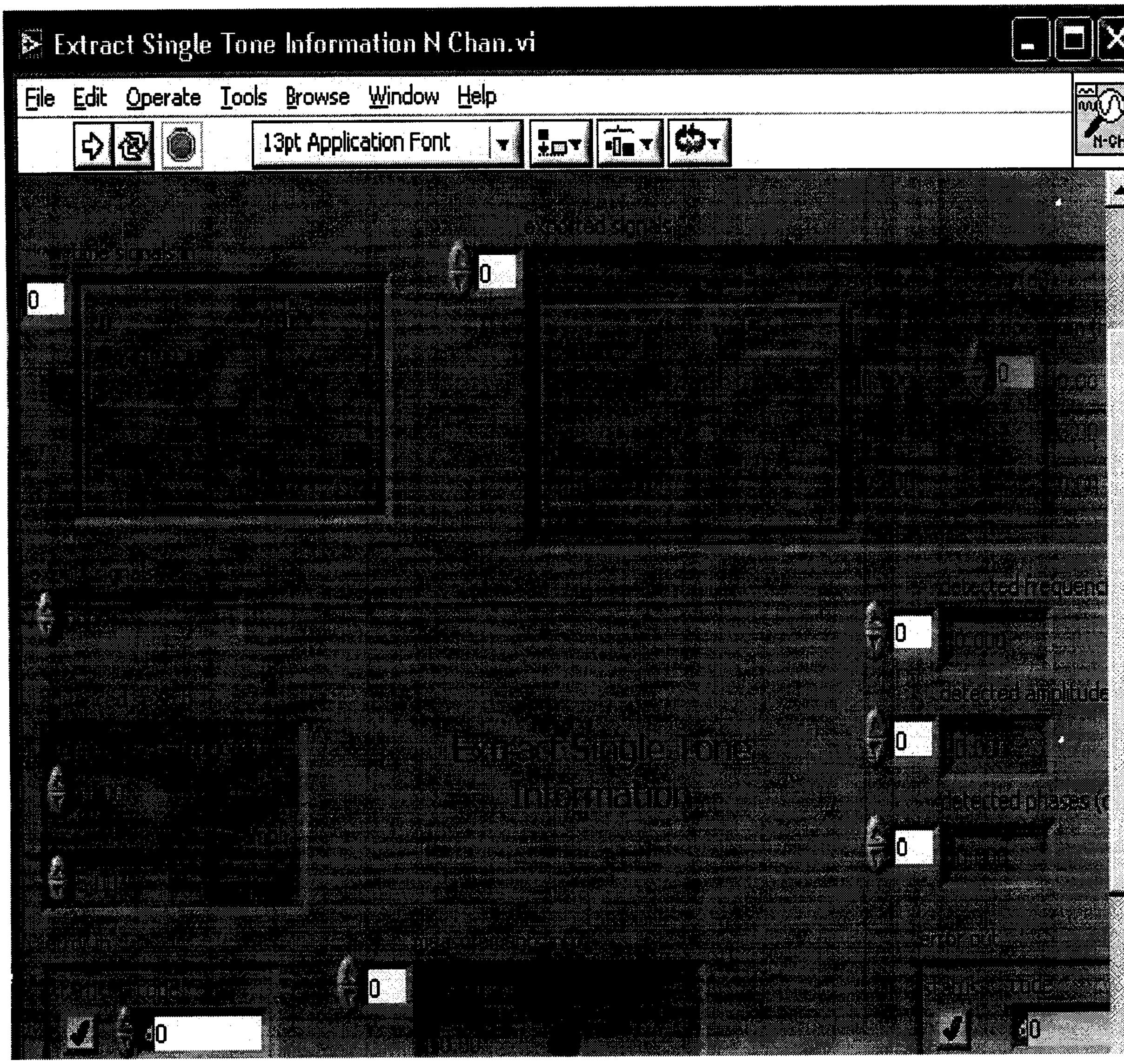


Fig .7 Front panel - Exact single Tone Information N channel

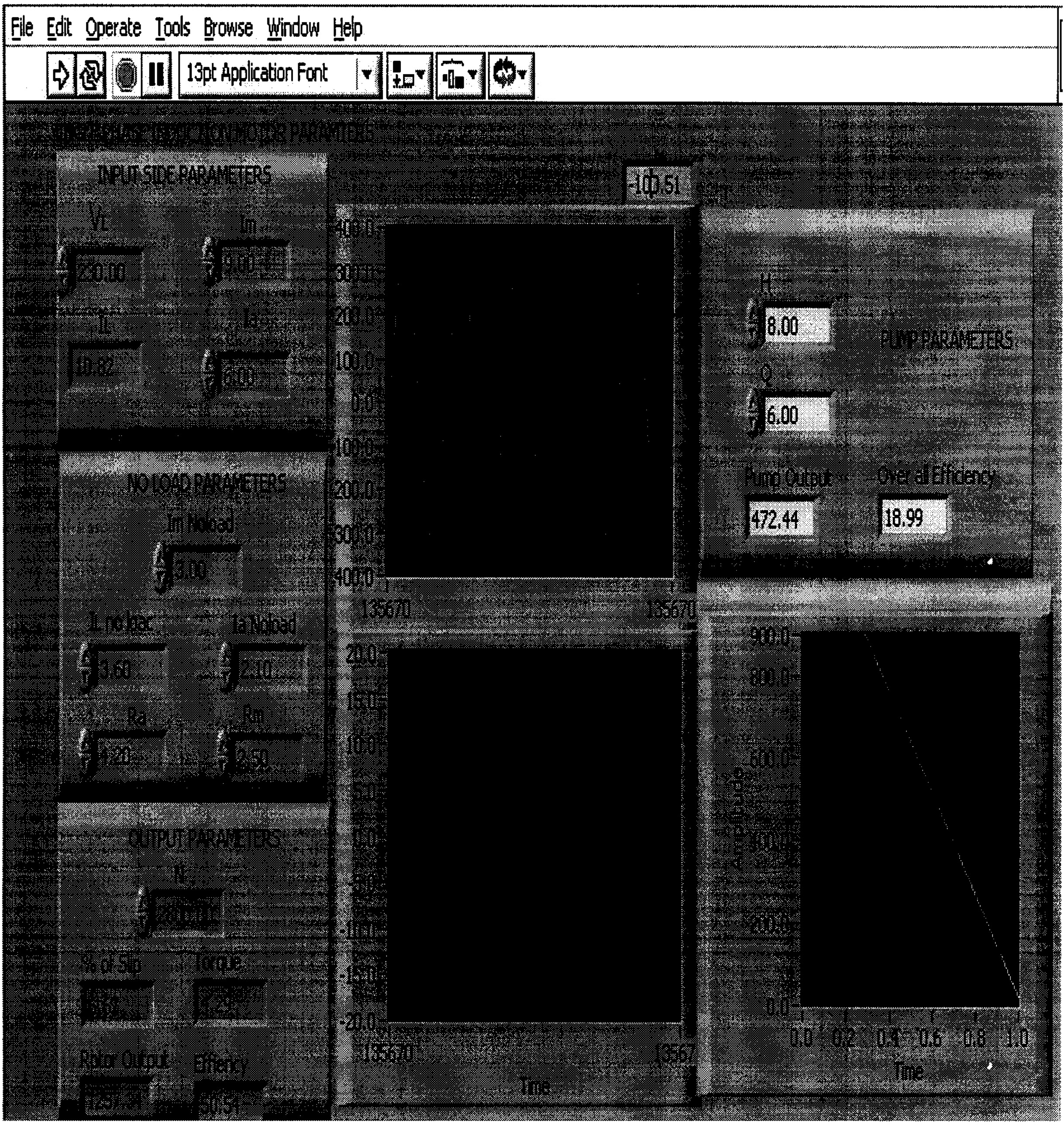


Fig.8 On line results