Dr. Norbert Cheung's Lecture Series

Level 5 Topic no: 8

Motion Control Platform

Contents

1. Introduction
2. Implementation of Real Time Control
3. Communication Networks and Protocols
4. Motion Control Interfaces
5. Development Cycles and Tools

Email: norbert.cheung@polyu.edu.hk
Web Site: www.norbert.hk

1. Introduction

The Work Cell Principle - an overall picture

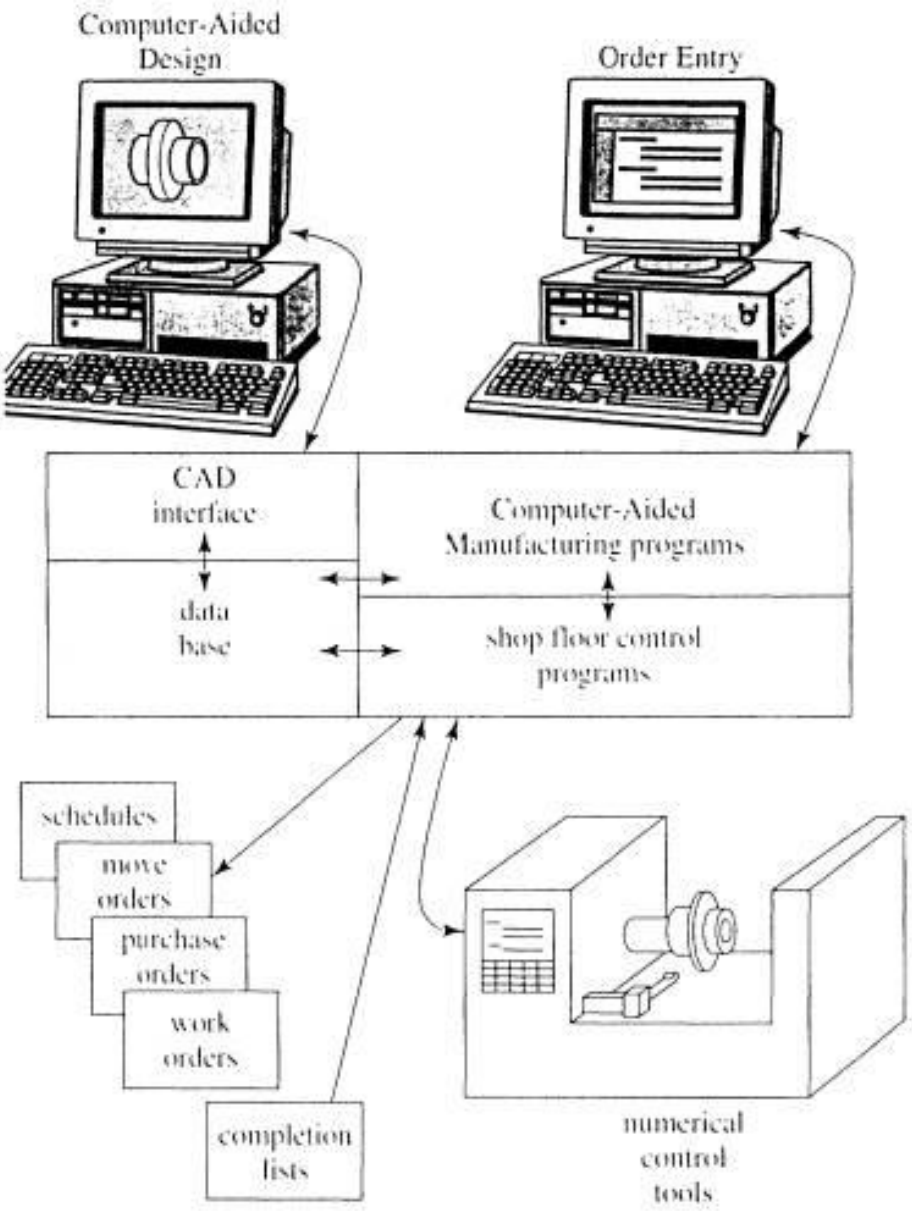
The focus of an industrial automation control system is at the work cell. Those devices that are monitored and controlled by computers are called peripherals. Peripherals may include:

- *Keyboards, monitors, printers, or disk drives.* While these are essential to a personal computer, an industrial controller can often do its work without any of them.

- *Sensors and actuators.* To control an industrial process, the computer must “see" the process through its sensors, and change the process through its actuators. Whole chapters are devoted to sensors and actuators in this text.

- *Intelligent peripherals such as remote sensors and actuators with communication adapters.* A computer can be connected to several communication adapters with a single set of conductors and can read selected sensors or write to selected actuators individually, specifying each by an address.

- *Other computerized controllers such as numerical control (NC) equipment, robots, vision systems, and other computers.* We can readily achieve 4-quadrant control of a dc machine by using a single converter, combined with either field or armature reversal. However, a great deal of switching may be required.

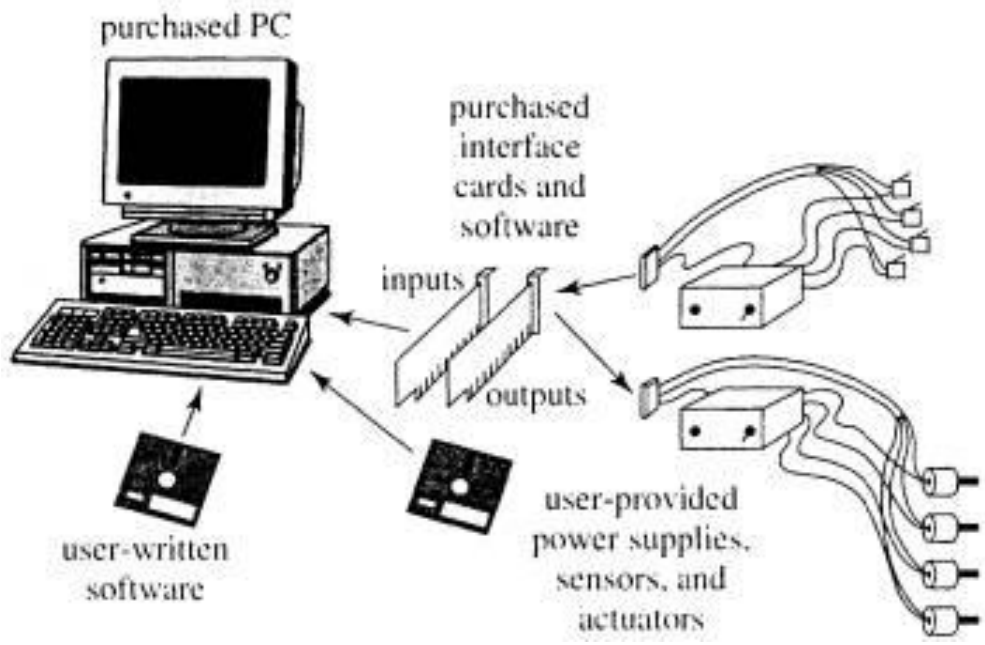


A CAD/CAM Production Work Cell

Work Cell Controller

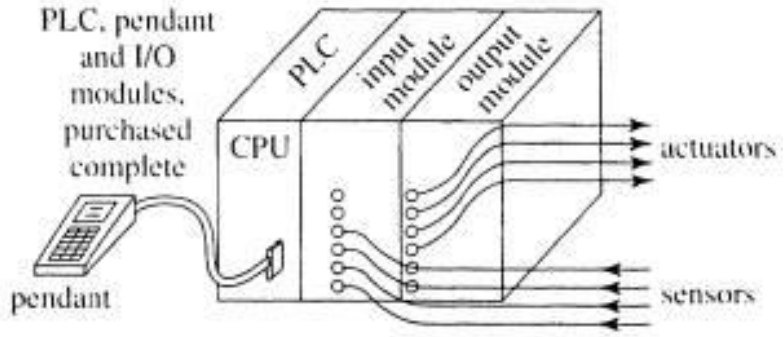
If none of the computers already in the work cell are powerful enough to control the work-cell's components, then a work cell controller is needed. An obvious choice for a computer to use as a cell controller is a standard personal computer (PC). A PC will require additional interface cards and programming. The set up person needs to be trained to select the hardware and software, to design the electrical connections, and to write the special-purpose programming.

The personal computer itself does not have to look like the standard office version. They can be purchased in "industrially hardened" versions with increased resistance to abuse and dirt, or purchased as single cards that can be assembled into user-built cases.



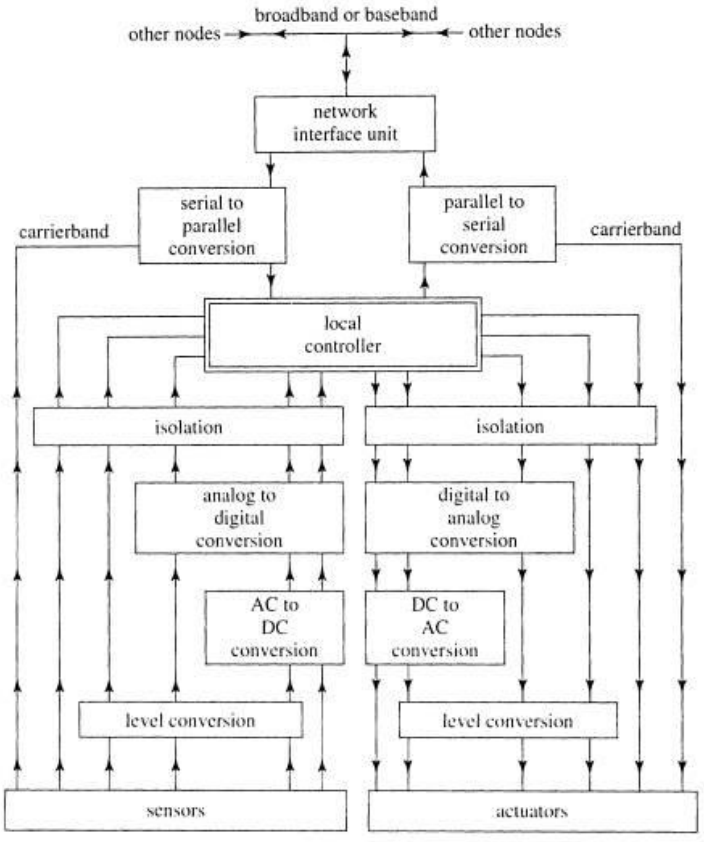
Using PC for Controller Purpose

Most new users of automation control start with a programmable logic controller (PLC). Even the simplest PLC comes complete with interface hardware, programming software, and a wide array of easily-connected expansion modules available from the PLC manufacturer. PLCs are usually programmed using a language known as ladder logic, which was developed to he easily understood by industrial technicians familiar with relay circuit design. PLCs were handicapped by the limitations of early ladder logic. Communication between proprietary PLCs and other controllers was difficult. Mean while. PLC suppliers expanded their offerings to allow networking of PLCs, improved the programming languages, and offered programming software so that PLC programs could he written at standard PCs.



Standard PLC Set up

Mainstream computer manufacturers have had a belated awakening and are now offering what they identify as "cell controllers." Cell controllers are primarily intended to handle inter-controller communications, to act as data storage devices, and to provide operators with a central control panel from which whole manufacturing systems can be controlled.



Hardware Block Diagram of a Typical Cell Controller

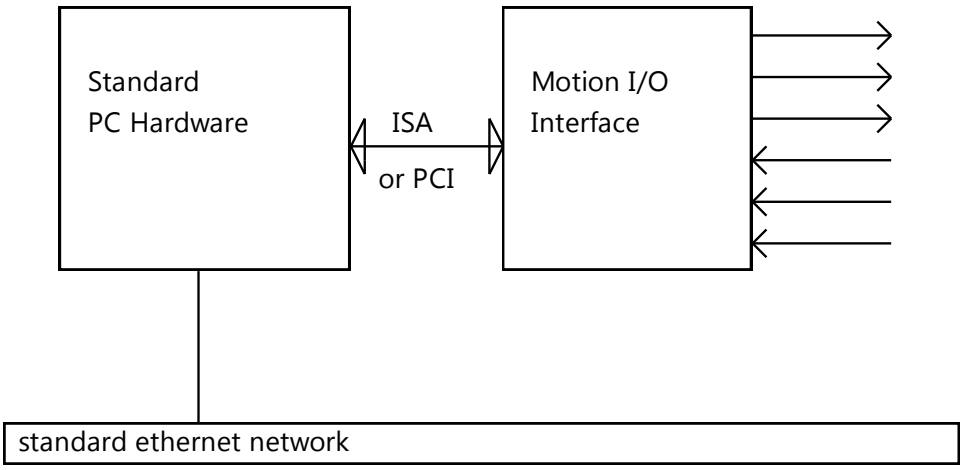
2. Implementation of Real Time Control

Some Important Issues:

- Window is not a real time operating system
- PC hardware is relatively cheap and powerful
- Motion Controller needs a tightly scheduled real time platform

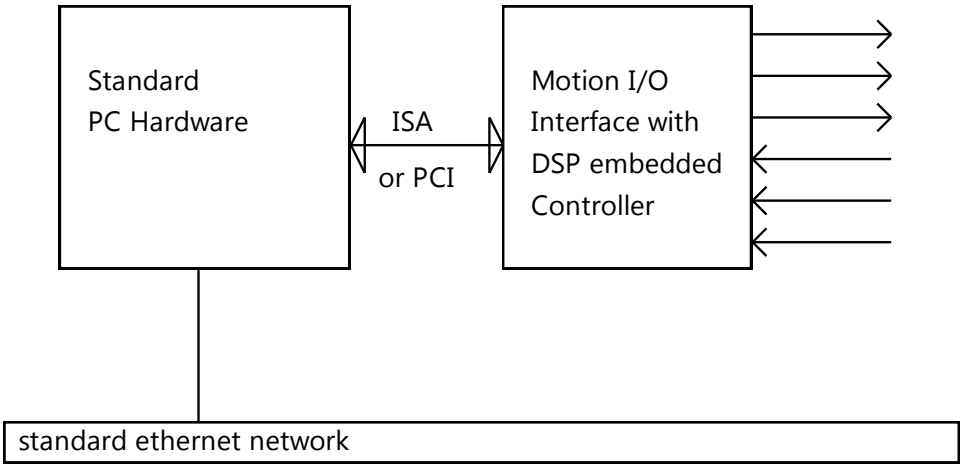
Some Possible Configurations:

Solution 1: use standard PC hardware and real-time OS



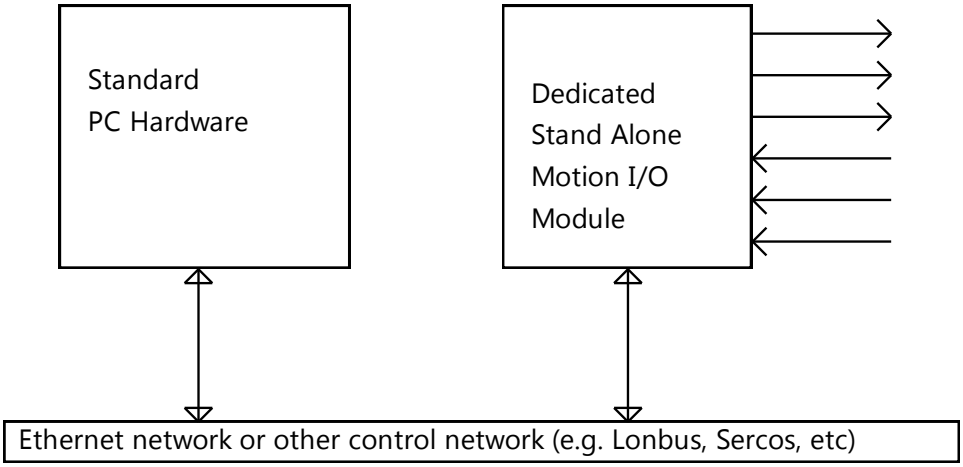
Comments:

Solution 2: use standard PC hardware for non-time critical control and DSP based embedded controller for time critical routines



Comments:

Solution 3: use dedicated controller module attached to the control network, use PC as a central monitor



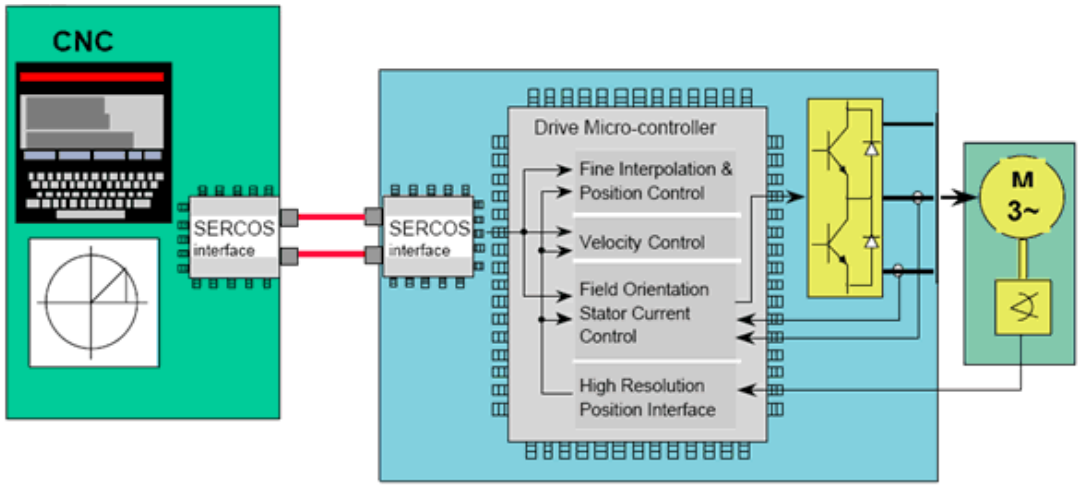
Comments:

3. Communication Networks and Protocols

Apart from standard TCP/IP internet interface, there are some communication bus standards useful to motion control. They are:

- SERCOS interface – for motion control synchronization
- LIN bus – for automobile slow speed device control
- LON bus – for distributed low-cost control points

SERCOS Interface



- **Highest Precision and Speed independent of Axes Quantity**

SERCOS is an acronym for SErial Realtime COmmunications System, a digital motion control bus that interconnects motion controls, drives, I/Os, sensors and actuators for numerically controlled machines and systems. It is an open controller-to-intelligent digital device interface, designed for high-speed serial communication of standardized closed-loop real-time data over a noise-immune, fiber optic ring (SERCOS I & II) or Industrial Ethernet cable (SERCOS III). SERCOS is an international standard.

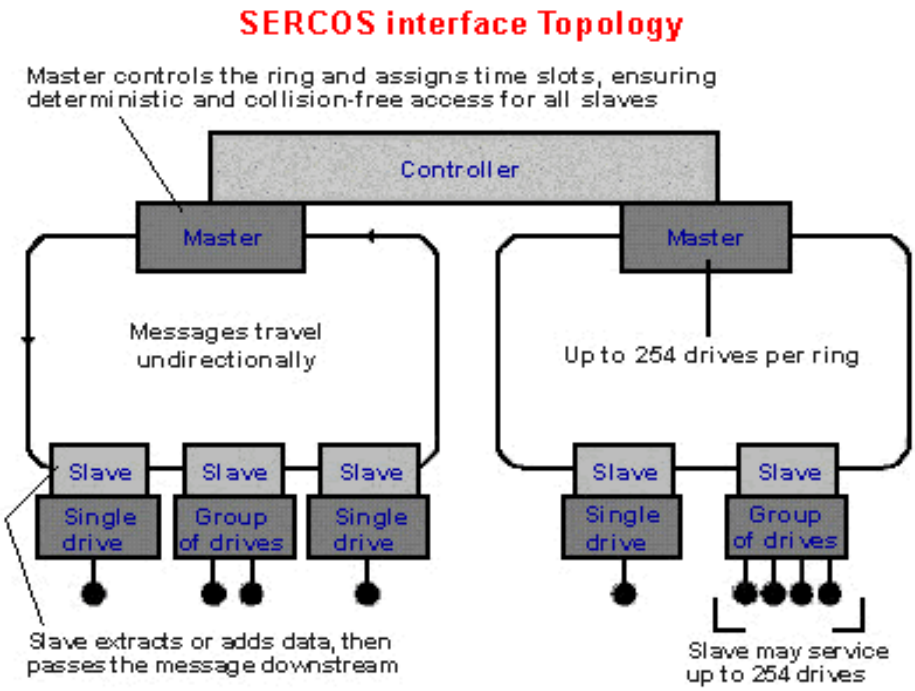
The SERCOS interface offers more than 500 standardized parameters that describe the interaction of drives and controls in terms independent of any manufacturer. This rich instruction set provides for communication of commands, parameters, status, and diagnostics, handing many different types of systems. SERCOS also

includes features for I/O control that often allow a machine builder to dispense with the need for a separate I/O bus.

The SERCOS interface greatly reduces connectivity problems in control systems. It can connect up to 254 drives to a control using one fiber optic cable ring or a single daisy-chained Industrial Ethernet cable. A comparable traditional analog servo system with 8 axes of motion may require over 100 wires between the drive and control.

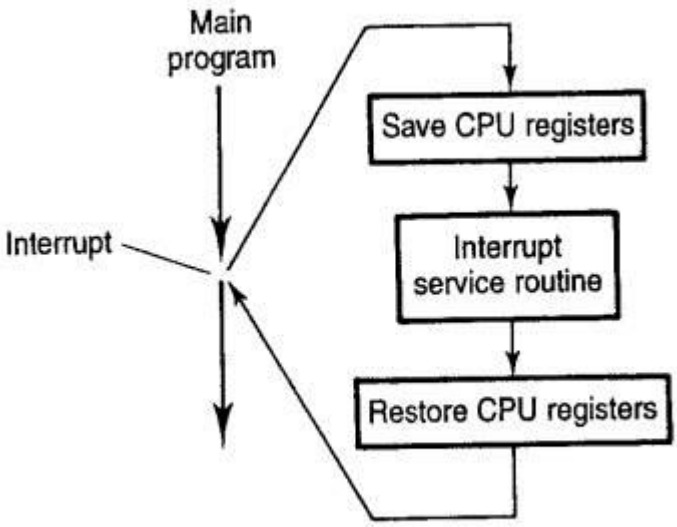
Functions of SERCOS

- Exchanges data between control and drives, transmitting command and actual values with extremely short cycle times
- Guarantees exact synchronization for precise coordinated moves with as many axes as required.
- Supports four operating modes: torque, velocity, position control and block mode. The transmission of nominal position has proven to be the best solution for fast, highly precise applications.
- Includes a service channel for non-cyclic data transmission, such as internal parameters, data and diagnostics. A complete set of drive parameters can be downloaded and uploaded for storage via the service channel.
- Enables the use of controls and drives from different manufacturers in a system, by standardizing all data, parameters, commands and feedbacks exchanged between drives and controls.

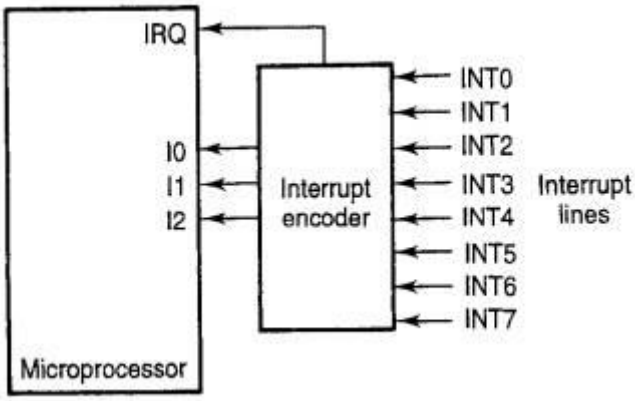


4. Motion Control Interfaces

The Interrupt Controller



Interrupt Operation



An interrupt priority encoder

5. Development Cycles and Tools

Development Cycle

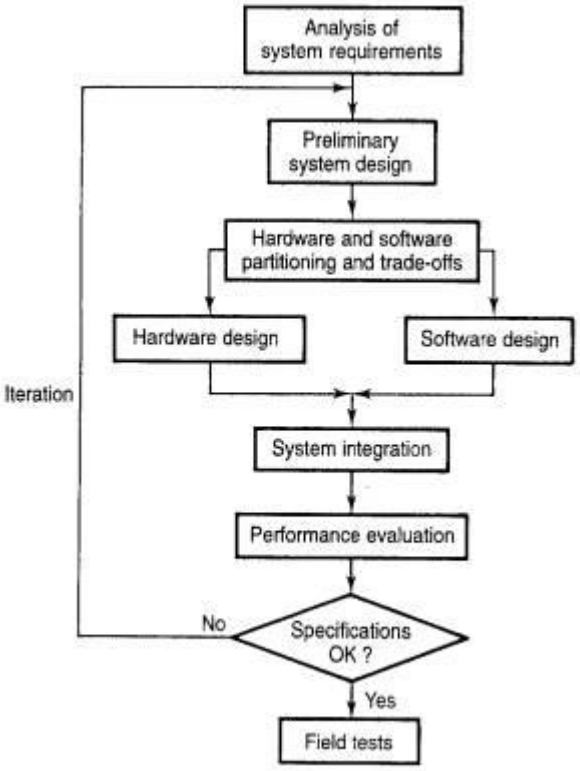


Figure 10-38. Development cycle of microprocessor-based real-time motion control systems.

Real Time Software

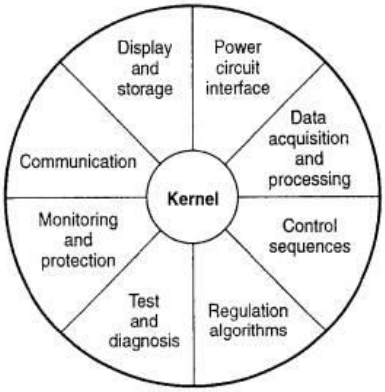


Figure 10-39. Typical structure of real-time control software.

Development Tools

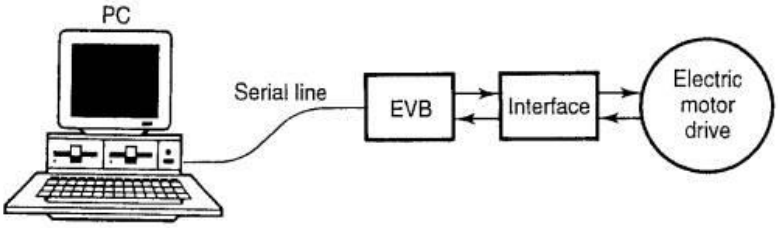


Figure 10-40. Low-cost development system based on an EVB.

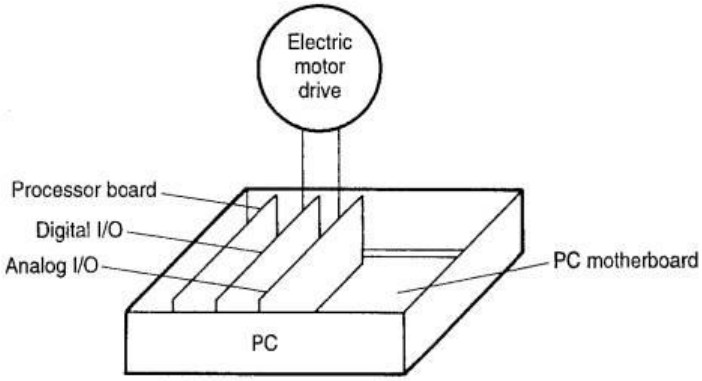


Figure 10-41. Development system based on a PC and add-on boards.

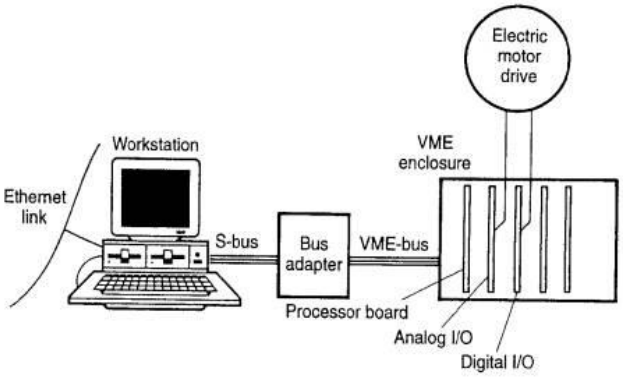


Figure 10-42. Development system based on a workstation and VME boards.

6. Optical Encoder Interface

A High Performance Motion Control System

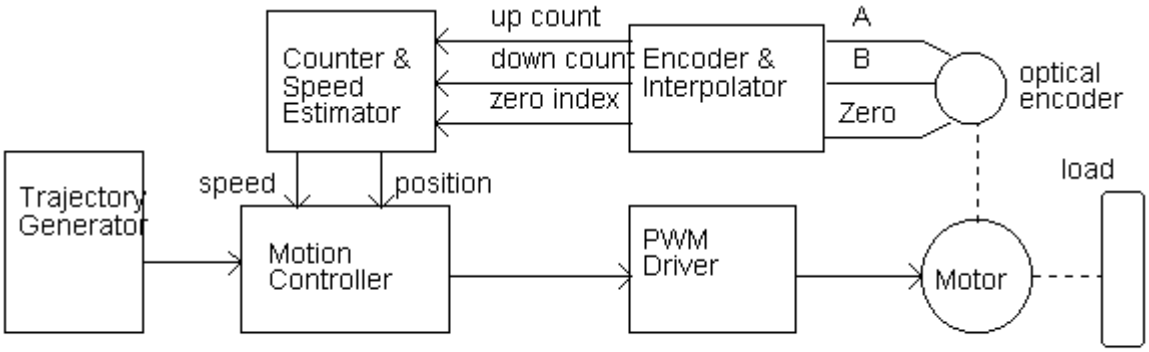


Fig. 2 Block Diagram of a typical high performance motion servo system

Block Diagram of a decoding circuit

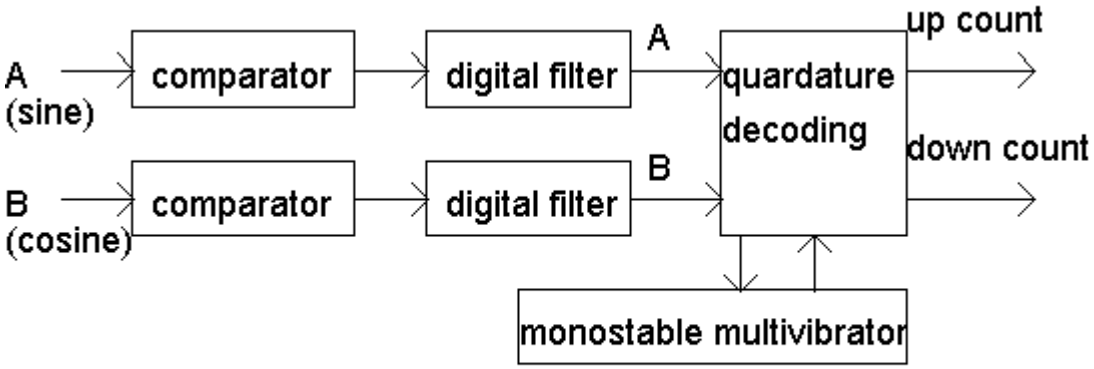
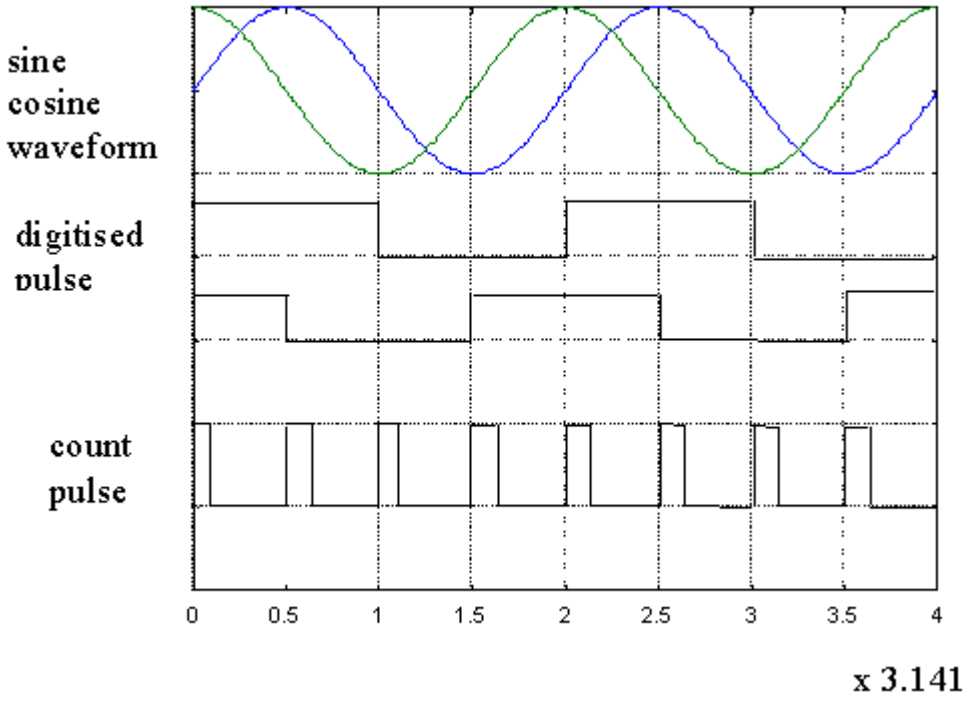


Fig.1 Typical block diagram of a decoder circuit (for optical encoder)

Optical Encoder Signals and Decoding Block Diagram



Increase the resolution of optical encoders - by interpolation

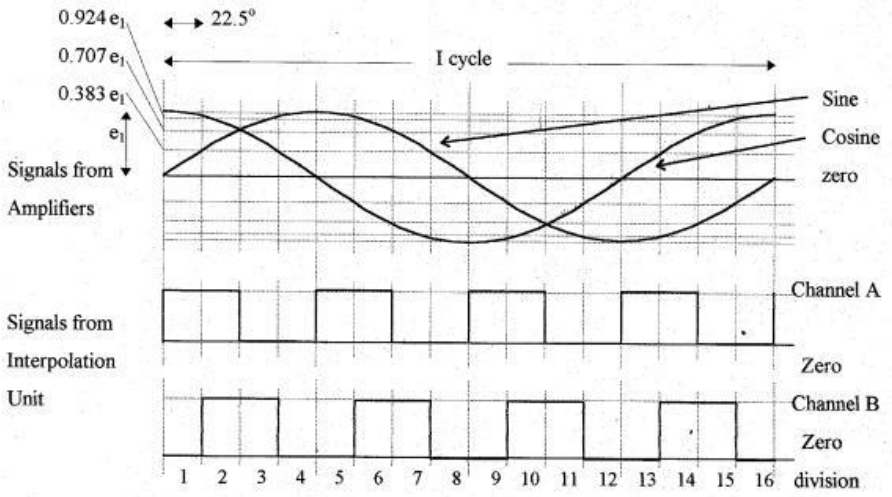
Presently, there are a few existing methods to perform interpolation of encoder waveforms. These include

(i) resistor based look-up table,

(ii) processor based ADC conversion, and

(iii) processor based ADC conversion with variable resolution.

Interpolation Signals



Use a resistor network to find the intermediate positions

There is a one-to-one mapping of these signals to the intermediate position values of the optical encoder. Essentially, the locus of the output signal is a circle, when they are plotted on a 2D plane. However, the magnitude of the signals decrease as the speed of the optical encoder is increased. A circular locus can only be created when the speed is low.

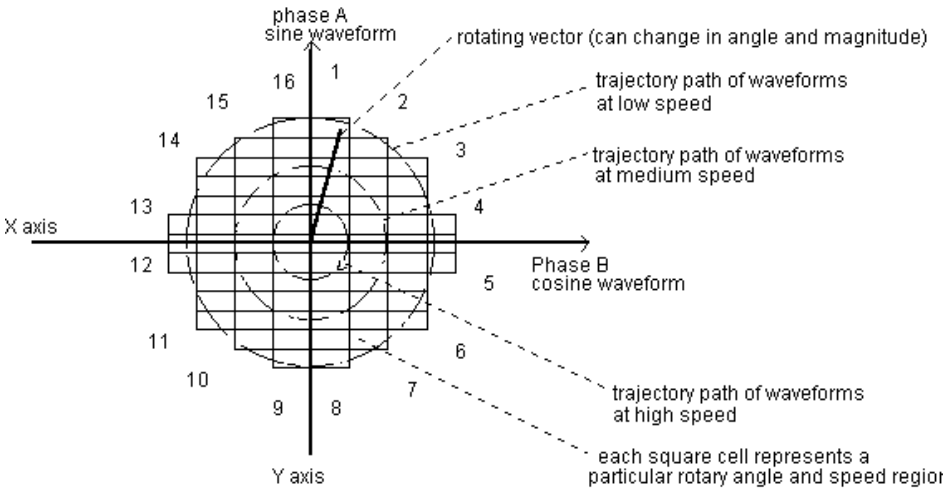


Fig. 5 Mapping the rotating angle by bounded regions

Use a two-channel data acquisition circuit

This is the most common interpolation method used in motion control systems. Rather than building a large resistor network with a large number of comparators, the interpolation process can be implemented by a high speed data acquisition circuit.

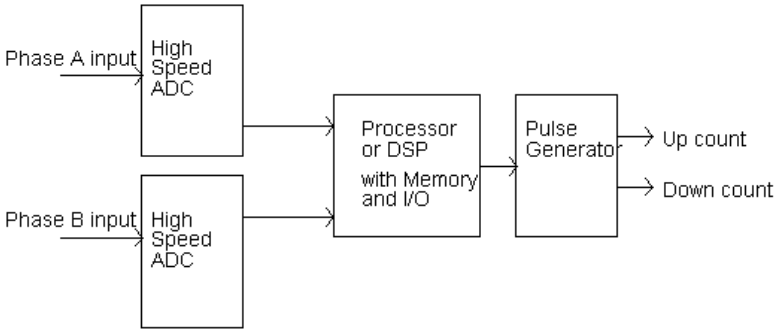


Fig. 6 Block diagram of a processor based interpolation unit

Disadvantages

1. Complex circuit and expensive component cost due the ADCs and computing unit.
2. Only suitable for low speed operation, due to the speed limitation of the computing unit and the data acquisition circuit.

The higher the resolution, the higher will be the pulse output rate, and the higher will be the computing requirement. Under this configuration, it is very expensive to implement a high speed and high-resolution interpolation decoder.

Variable resolution data acquisition circuit

For most applications, high resolution is not required for high-speed motions, but when the system's speed is very low or coming to a stop, high resolution is required. The previous method has the advantage of high resolution, but it cannot travel at very high speed.

To overcome this problem, many systems use a variable resolution scheme. When the speed is too fast for the computing unit to handle, its resolution decreases to reduce the computing burden.

However, this method has the disadvantages of complex hardware (to accommodate the flexible resolution scheme) and the problem of a “glitch” during resolution change.

A better method to perform sine-cosine interpolation

The method takes advantage of the fact that both sine and cosine waveforms decrease by the same ratio when the vector rotating speed is increased, and the shape of the circular locus and the vector angle remains the same.

Therefore, instead of comparing the waveforms to reference voltages, the two waveforms can be compared with each other to find out the angle of the rotating vector, and deduce the immediate position values. The problem of signal variation can be effectively eliminated.

The Method

To demonstrate the design process of the proposed method, a 16-fold interpolation unit is used as a design example. The interpolation unit divides the sine-cosine cycle into 16 sections, with an angular distance of 22.5° for each section.

The sectioning process can be represented conceptually as shown in Fig. 7. In this figure, 8 boundary lines (L0-L7) are used to divide the 16 sectors (S0-S15). The following table can express the boundary conditions of L0-L7:

L0: Y=0	*L1: X=0.414Y*	*L2: Y=X*	*L3: Y=0.414X*
L4: X=0	*L5: Y=-0.414X*	*L6: Y=-X*	*L7: X=-0.414Y*

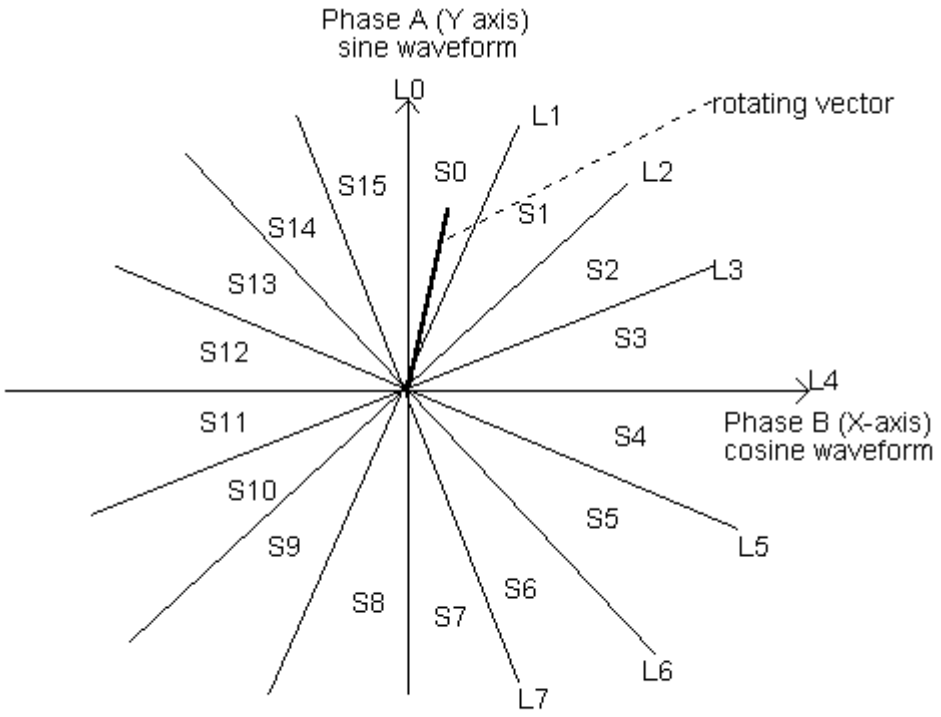


Fig. 7 Conceptual representation of a x16 interpolation unit

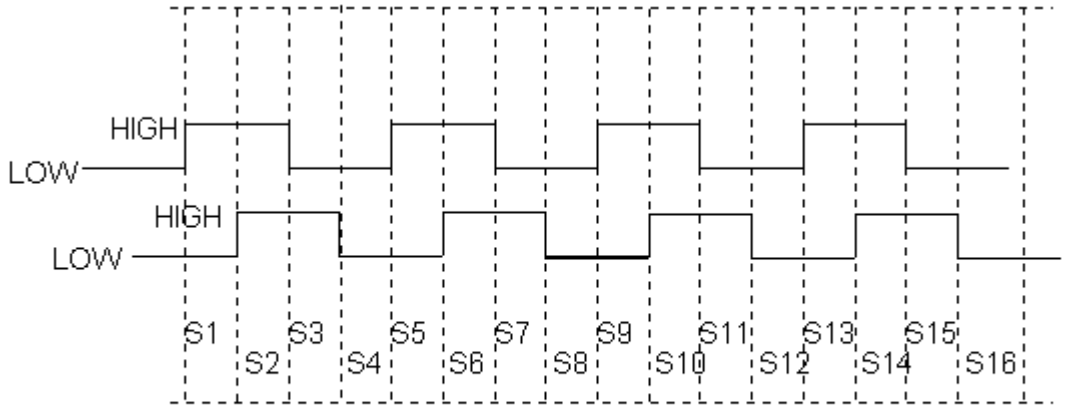


Fig.8 The quadrature output waveform of the 16-fold interpolation unit

By setting the conditions for each of the sector (e.g. S0 Þ Y>0 and X<0.414Y) all 16 sectors can be identified. The position of the optical encoder can be represented by a rotating vector in Fig. 7. In this way the change of signal amplitude will merely change the magnitude of the rotating vector; the angle of the rotating vector remains the same. By testing boundary conditions of the rotating vector, one can find out which sector the rotating vector resides.

Since the interpolation unit is designed to be interfaced with conventional quadrature decoding unit, the interpolation unit should output 90° phase-shifted A-B digital pulse-trains, as shown in Fig. 8. The figure shows that for channel A, output is high at S1, S2, S5, S6, S9, S10, S13, and S14. For channel B, output is high at S2, S3, S6, S7, S10, S11, S14, and S15. These output criteria can be matched with the boundary conditions of Fig. 7, to form a table shown in Fig. 9.

Output	Input	Output Sector
Channel A	(x>0) & (y>x) (y<0) & (-y<x) (y<0) & (y<x) (y>0) & (-y>x)	S1,S2 high S5,S6 high S9,S10 high S13,S14 high
Channel B	(0.414y<x) & (y>0.414x) (-y>0.414x) & (-0.414y<x) (0.414y>x) & (y<0.414x) (-y<0.414x) & (-0.414y>x)	S2,S3 high S6,S7 high S10,S11 high S14,S15 high

Once this is obtained, the 16 sectors can be translated into A-B pulse-trains (with ´4 resolution increase), which in turn can be translated into up/down count pulses (with ´16 resolution increase) using standard logic hardware or ASIC implementation.

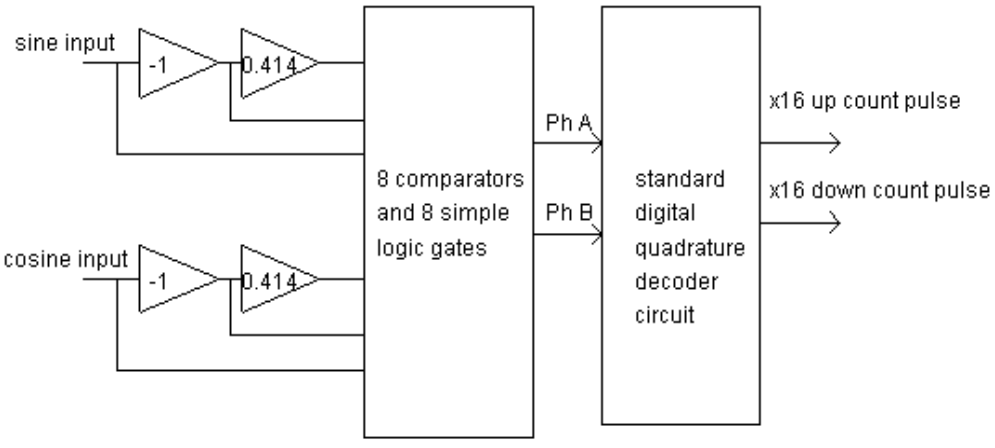


Fig. 11 Hardware block diagram of the x16 interpolation unit

--- END ---