The interfacing technology of PC-based Factory Automation systems

Hung-Yuan Chung and Dar-Win Jang

Department of Electrical Engineering
National Central University
Chung-Li 320 Taiwan ROC

Abstract

The purpose of this paper is to use a 4 axis servo card to design a very friendly man-machine interface to set up a practical PC based industrial controller system. We also introduced a novel Pulse generator IC PG9210 which is powerful in multi axis motion control of servo motors.

Key Words: PC based, industrial controller, FA, PG9210, man-machine interface

1. Introduction

The automation technology has already played an important role in engineering applications. The factory automation (FA) is also widely applied in advanced nations to keep their competitive ability from falling down. The automation technology development is also the main policy planned by government [1,2].

FA equipment's such as Numerical Control [3,4] machine tools, industrial robots, and other automated machinery are used to save energy and promote automation, which allows for high productivity and also produces quality products. The aforementioned FA equipment's is becoming more technical with the application of servo motor and motion control technologies [5,6,7]. Rolf Johansson presents algorithms for continuous time quadratic optimization of motion control [8]. The proposed optimal control is useful both for motion control, trajectory planning, and motion analysis.

The main purpose of this paper is to present practical technology to fit the needs of users who are interested in FA. Here we must emphasize a very important idea about the FA technology [9]. The basic philosophy of the FA technology is "Choosing appropriate way in order minimum cost."

When developing the automation technology, an FA engineer must know that, we not only need the integration of related technology from different field, we also need to know the actual need of the industry who want to apply the FA technology. There are too many differences in applying FA to different cases. This is a major problem in designing a successful FA system.

Because the motivation of this paper is to present practical technology to fit the needs of users in Taiwan who are interested in FA. First, we must know the related information in FA market. A typical design approach of FA system can be summarized as in Fig 1-1.

![Diagram](image).

Fig 1-1 The design approach of a FA system

Step 1: Design precise and robust mechanism to fit the need of FA system. This is a very important step. Because without a precise and robust mechanism, it is useless to control it.

Step 2: Choose proper transmitting mechanism, such as ball screw.

Step 3: Choose appropriate motor and driver. Now AC servo motor are widely used.

Step 4: Design a controller. This is the main part that we will discuss.

Now let us look at the needs of users. For users they need a controller which have the following characteristics: high reliability, very easy to use, and low cost. So we try to design a very friendly man machine interface by using a domestic low cost PC-based high performance 4 axis servo control card (Appendix A) to fit the needs of users. Here we focus on PC-based industrial controllers. Next we will present a very special pulse generating IC PG9210 to develop a fully softwareized motion controller for pulse type servo driver. The major contribution is low cost and flexibility.

In addition, we have to take the machine set up environment, cost into consideration. We need to integrate different fields of technology such as mechanics, electronics, computer, to form the so-called mechatronics engineering [10]. Then we can get the best solution.

II. The man machine interface

1 Brief introduction

The man machine interface plays an important role in industrial controller system. B. P. [11] developed a man-machine interface for a computerized Numerical controller by Ada with object oriented method. If it is not easy for users to operate the system, it is not a practical one though it is very powerful.
When developing the software, compact and flexibility is our goal. We use the top-down technique to design the software which is easy for maintaining the program. We also want to make the best use of RAM, so we use dynamic memory allocation to reduce the use of memory as less as possible. The resulting EXE type files is 112638 bytes, and the program only use 127312 bytes RAM which is convenient for future developments. It is running under compatible IBM PC under MS DOS.

The basic operation of the interface is based on windows and function key with on-line information just like the operation of Turbo C. There is one thing special. The program can autodetect Chinese operating system. If it is in Chinese mode the text in the interface will be Chinese automatically otherwise it will be all English text. Because all the text information are stored in external file, it is an efficient way to change the interface in other language like Japanese or French just by writing an text file without recompiling the program.

2 Major functions

2.1 System parameters window

Because there are too many differences when applying FA to different factory. We have to set up proper system parameters to fit the various needs of different plant. The system parameter window can help us do this easily. The windows is shown as Fig 2-1. We can watch the system parameters of each axis.

If you want to change the value, you just move the cursor to the place where you want. Key in the correct value, then press F1 to update the value. You can write the updated value to EPROM, too.

![Fig 2-1 System parameter window](image)

We only list part of the system parameters. For further information please look at [12]. We have to emphasize that it is very important to set up system parameters carefully and correctly.

2.2 Run window

We develop a modified G-COMMAND (which will be discussed later) interpreter. We can use a text editor such as PE2 to edit a G-COMMAND file according our need of motion control. Then we can run the G file continuously or step by step by operation of windows. When the program is running, we can watch many messages on-line in the Watch Window such as running mode, step number, the position, the following errors, the limit switch status ... of each axis.

The G COMMAND is a motion control language widely used in NC system. Here is an example. G00 x y z : It means that the 3 axis of the machine will move to the position (x, y, z) at maxi speed. For further information please look at [6,7]. Here is the list of the modified G-COMMAND.

G01 : linear motion. G02 : clockwise circular motion.
G03 : counter clockwise circular motion. G04 delay operator.
G32 clockwise spiral motion.
G33 counter clockwise spiral motion.

d1, d2, d3, d4 distances or end point to go (\(\mu m\)) -2e3 - 2e3 -1.
speed : speed index. When speed is 1, it means that velocity is 20 \(\mu m/\text{ms}\) or 1.2M/min. It is a floating point between 1 - 500

c1, c2 : the point of a circle (\(\mu m\)).
R : Radius of a circle (\(\mu m\)).

G COMMAND list :

```
G01 [XY YZ ZX] d1 d2 speed

G01 [XW YW ZW] d1 d2 d3 speed

G04 delay (delay ms 0 \(\leftrightarrow\) 2\(^{31} - 1\))

INC (Change to incermetal mode)

ABS (Change to absolute mode)
```

G02 C [X Y] d1 d2 c1 c2 speed (with center specified)

G02 R [X Y] d1 d2 R speed (with radius specified)

G32 C [XY] d1 d2 d3 c1 c2 speed (with center specified)

G32 R [XY] d1 d2 R speed (with radius specified)

```
```

~ 268 ~

Authorized licensed use limited to: National Central University. Downloaded on August 31, 2009 at 22:27 from IEEE Xplore. Restrictions apply.
2.3 TEACH window

We can teach the machine by hand wheel. The teach window is in Fig 2-2. First, we must decide which axis we want to teach, then we use handwheel or set the QUICK move command. When we move to the desired position, we press F4 to save the position. After we finish teaching, we press F5. The machine will go to the position which we taught.

III. Results and discussion

Fig 3-1 is the experiment result for position control. The desired position is 123.456 mm. The final position is 123.456 mm. The tracking error is zero. This is the most important result for point to point position control. This means that the resolution of our system is μm. The rise time can be assigned by changing the speed for point to point position control in the system parameter window.

Fig 3-2 is the experimental result for linear interpolation. The desired linear motion is a line from (0,0) to (12.345,11.246) mm. The experimental result is matched with the desired.

Fig 3-3 is the experimental result for circular interpolation. The desired circular motion is an arc from (1.5,6) to (10.5,3) with center (5.5,3) mm. The experimental result is matched with the desired.

IV Position control of AC servo motor

4.1 Introduction of PID control theory

Despite the advent of many sophisticated control theories and techniques, the majority of industrial processes nowadays are still regulated by PID controllers. The reasons for its widespread use are that it requires very little knowledge of the plant dynamics and the methods of determining the controller parameters are well known and understood. It does reveal the rich potential of this extremely simple control strategy for meeting various specifications for vast variety of industrial processes.

4.2 The PID control algorithm: The basic algorithm

A typical control system is shown in continuous form in Figure 4.1a and discrete form in Figure 4.1b.
The PID controller can be expressed in the form

\[ G_c(s) = \frac{M(s)}{E(s)} = K_c(1 + \frac{1}{T_d s} + T_i s) \]  

where \( K_c \) = controller gain, \( T_i \) = integral action time, and \( T_d \) = derivative action time.

The controller in the form shown in equation 4.1 is known as the ideal or non-interacting three term controller in that each of the three terms is evaluated independently.

Equation 4.1 can be expressed in the time domain in the following form:

\[ m(t) = K_c \sum \frac{d\epsilon(t)}{dt} + K_i \int \epsilon(t) dt + K_d \sum \frac{\epsilon(t)}{T_d} \]  

In order to derive a control algorithm, equation 4.2 has to be converted to discrete form. One method of doing this is to use first order finite differences. We can write

\[ \frac{df}{dt} = \frac{f(t) - f(t - 1)}{\Delta t} \]

and hence equation 4.3 becomes

\[ m(t) = K_c \Delta t \sum e(t) + K_i \sum \frac{e(t)}{T_i} \]  

By introducing new parameters as follows:

\[ K_c = K_c \]
\[ K_i = K_i \times T_i / T_c \]
\[ K_d = K_d \times T_d / T_c \]

where \( T_c = \Delta t \) = the sampling interval, equation 4.5 can be expressed as an algorithm of the form:

\[ s_e = s_e + e \]
\[ m_e = K_p e_e + K_i s_e + K_d (\sum e - s_e) \]  

where \( s_e \) = the sum of errors.

4.3 Saturation's

In any practical application the value of the manipulated variable \( m \) is limited by physical constraints. If a PI controller is used, then because there is a standing error, the integral term will continue to grow, i.e., the value of \( s_e \) in equation 4.5 will be increased at each sample time. Consequently, the value of manipulated variable will increase and the demanded heat output will continually increase.

The effect is called integral wind-up or integral saturation and results in the controller having a poor response when it comes out of a constrained condition.

Here we use fixed limits method to solve this problem. A maximum value and minimum value for the integral summation is fixed and if the term exceeds this value it is reset to the maximum or minimum value of the manipulated variable.

4.4 Reference values

In the Figure 4.2 the system is characterized by forming an error that is the difference between the set point and the process output. The controller generates a control signal by operating on the error. Here we can use an extra parameter \( b \) to tune the response of the closed loop system. Which is explained below.

Figure 4.2 A simple feedback system with error feedback.

This means that the control law is not based on the pure error feedback. A straightforward extension of this idea is to choose a PID controller of the form

\[ u = K_p \left[ e + \frac{1}{T_i} \int e(t) dt + \frac{T_d}{T_i} \frac{d\epsilon(t)}{dt} \right] \]  

where the error in the proportional part is

\[ e_p = br - y \]  

and the error is

\[ e = r - y \]  

The controller obtained for different values of \( b \) will respond to load disturbances and measurement noise in the same way. The response to set point changes will, however, depend on the values of \( b \). We will show the effect of \( b \) later.

4.5 Introduction of motion control

Motion control means that we can control at will the position, velocity, acceleration, and force of a mechanical system in the space. This technology has contributed to providing high productivity on the production line and to producing high quality products which are the basis of modern industry development. Advanced motion control requires high speed, large allowable current reliable semiconductor switches, fast processor.

4.6 Position control of AC servo motor

There are two kinds of controls in positioning. One is called PTP (point to point control) and the other is CPC (continuous path control). In both cases, considerations in accuracy, response time, and cost are required. The choice of sensors is the most important factor in obtaining positioning accuracy.

The control system block diagram is shown in Figure 4.3.

~ 270 ~
The plant we want to control is a industrial digital AC servo driver AFS-6 and a AC servo motor. We use a PCL-726 which is a six channel D/A output card. It can generate a voltage signal between -10V and 10V with 12 bits which is compatible with the driver. We use a PCL - 833 which is a 3-axis quadrature encoder and counter card to feedback the position signal from the encoder of motor. Then the reference command is subtracted from the feedback signal to form the error signal e(k) which is the input of the PID position controller.

Now let's look at the result of experiments. We can see in Fig 4.4 which the response is fast with very small overshoot 0.46%. The following error is + one pulse which equal to \(360 \div 4000 = 0.09\) (per pulse) This proves that we can get good response with proper value of PID and b values. The sampling time is 22 ms. We can control two axis of servo motor with sampling time 22 ms. The results is in Fig 4.5 and 4.6. The result is quiet satisfactory.

Figure 4.4 Response of PID position control with P=0.2 I=0.02 D=0.06 b=0.96 Mp=0.46%

Figure 4.5 Response of Y axis PID position control with P=0.4 I=0.043 D=0.0021 b=0.98 Mp=2.71%

V. Conclusions

We have proposed a practical PC-based industrial controllers system with friendly man-machine interface to fit the domestic FA needs. The main contribution is the man machine interface. We use Turbo C with top-down technique to design a window-based interface which is compact and flexible.

We can see in section IV that PID control with modification of reference value is an easy but powerful control law. The problem is how to tune an optimal parameter. That is what we want to overcome.

The other contribution is that we can set up a hardware interfacing card by PG9210 [13]. PG9210 (9210) is a IC which generate pulses which is suitable for the position control of step motor or servo motor. In order to have multi axes control, 9210 can have a master IC which send synchronous signal to slave IC. Then we can get our goal for multi axes real time control.

We have to write the pulse data into 9210 within the interrupt loop, then we get the continuous pulse output signal. We can change the direction, frequency, acceleration, interrupt interval of pulses to fit different kind of CPU. The block diagram of PG9210 is in Appendix B Fig B-1.

With 9210, we can develop a servo motion control card with low cost but high performance. The main part of the system is software design. We can use the top-down technique to design more efficient and flexible software environment. The block diagram for multi axes control is shown in Appendix B Fig B-2. The main task to calculate the correct interpolation data for each axis in every interrupt internal with external I/O scanning. The structure of software planning is very important.
References:

12. The Uerro Electronics, "Manual of 4 axis servo card 4020/4060".
13. JWACO, Japan, "Manual for pulse generator IC PG9210".

Appendix A. Introduction of 4 axis servo card (SNC 4020)

A.1 Function

- 4 axes can be simultaneously controlled
- Precise position and speed servo control
- Linear and circular interpolation motion
- Hand wheel teaching
- Error detection in every step
- System parameters programmable which can be saved to EPROM
- Encoder A/B phase software programmable

A.2 System description

4020 can control 4 speed type (voltage type) servo drivers with servo motors plus a hand wheel for teaching. 4020 use 4 12-bit D/A (-10V -> +10V) which sends to the V command of driver then is amplified to drive servo motor. Here 4020 use optical encoder to get position feedback. It is shown in Fig A-1.

![Fig A-1 System block diagram](image)

A.3 System block diagram

Fig A-2 is the function block diagram of 4020. The position loop compensator uses proportional feedback control with velocity feed forward control. It can significantly reduce following errors while keeping the stability. If Kp is too large, response will be fast but system may become unstable. SKvf can reduce the following error. We should adjust these two parameters to get the best performance for each system. They can be changed in system parameter window.

A.4 Software environment

4020 use Motorola 64000 (64K) as the central processing unit. We use Common Data Memory (CDM) to hand shake PC with 4020. As we can see from Fig A-3. We can select 8K Bytes of RAM to avoid conflict with other program. Now the address is D04000-D05FFF.
Fig A-2 Functional block diagram

Fig A-3 CDM for handshaking

Figure 4.3 The block diagram of PID position controller