Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

# Implementation of a new PC based controller for a PUMA robot

FAROOQ M.[†], WANG Dao-bo
(*College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*)
[†]E-mail: muh_farooq1974@yahoo.com

Received June 13, 2007; revision accepted Aug. 20, 2007

**Abstract:** This paper describes the replacement of a controller for a programmable universal machine for assembly (PUMA) 512 robot with a newly designed PC based (open architecture) controller employing a real-time direct control of six joints. The original structure of the PUMA robot is retained. The hardware of the new controller includes such in-house designed parts as pulse width modulation (PWM) amplifiers, digital and analog controllers, I/O cards, signal conditioner cards, and 16-bit A/D and D/A boards. An Intel Pentium IV industrial computer is used as the central controller. The control software is implemented using VC++ programming language. The trajectory tracking performance of all six joints is tested at varying velocities. Experimental results show that it is feasible to implement the suggested open architecture platform for PUMA 500 series robots through the software routines running on a PC. By assembling controller from off-the-shell hardware and software components, the benefits of reduced and improved robustness have been realized.

**Key words:** Programmable universal machine for assembly (PUMA) robot, Computed torque control (CTC), Pulse width modulation (PWM) amplifier, Graphical user interface (GUI)
**doi:**10.1631/jzus.2007.A1962 **Document code:** A **CLC number:** TP2; TP311

## INTRODUCTION

Robots form an essential part of mechatronics and computer integrated manufacturing (CIM) systems. Robots are generally controlled by dedicated controllers. As upgrades become costly and interfacing becomes complex due to hardware and software conflicts, the flexibility of the robotic manipulators is reduced. Dedicated hardware and proprietary software which normally allows only high level programming by the users are costly and difficult to understand.

Since the early years of 1980's many projects have been carried out to develop an open architecture controller such as NGC (Sorenson, 1993), GISC (Miller and Lenox, 1991), ROBLINE (Leahy and Petroski, 1994), and so on, which try to realize an open architecture controller, and several prototype systems have been developed. However, they are not widely accepted due to the overly restrictive definitions and special standards.

The Unimate programmable universal machine for assembly (PUMA) 500 series robots mainly uses DEC LSI 11 processor running VAL (variable assembly language) robot control software (Unimation Robotics, 1983). Methods of bypassing VAL are discussed in literature, including Unimation technical reports (Vistness, 1982). However, most of these procedures have been confined to replacing the LSI 11 with another DEC computer, leaving peripheral hardware intact. A well-refined open structure architecture for industrial robot is discussed in (Pan and Huang, 2004). However, it is mainly based on common object request broker architecture (CORBA), leaving scope to simplify the hardware and software. A hardware retrofit for PUMA 560 robot is discussed in (Becerra *et al*, 2004), but it still relies on special purpose TRC041 cards installed on the backplane of Mark II controller.

The shift towards the personal computer open architecture robot controller and the impact of using these newer controllers for system integration are discussed in (Fiedler and Schlib, 1998). In fact, it is far more cost effective to develop a new hardware using less specific interfaces. An improved PC-based design for PUMA robot is presented in (Katupitiya *et al.*, 1997), but this hardware configuration purely depends on in-house built designs. In our paper, a flexible, modular hardware is developed for the PUMA robot, incorporating a PC, in-house as well as specialized hardware. Some technical problems in the previous design for velocity test profile of joints 1, 2 and 4 have also been addressed. The joints position tracking error at high velocities is also minimized in our design.

## ORIGINAL UNIMATE PUMA 500

The Unimation Mark II is an industrial robot controller as shown in Fig.1.

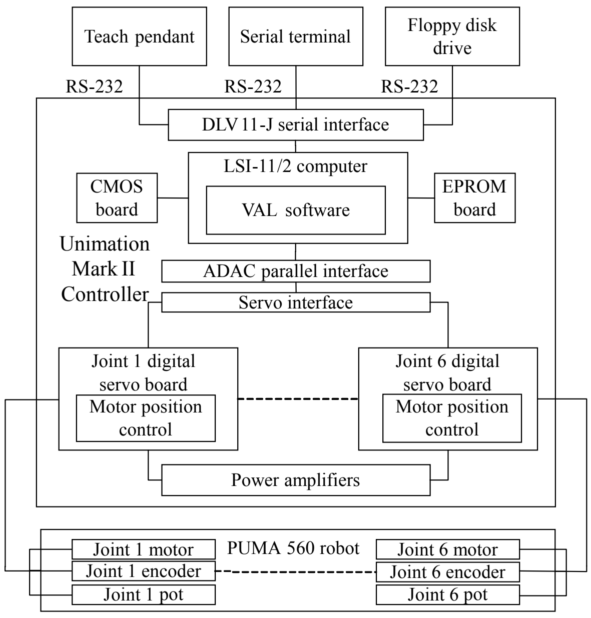


**Fig.1 Unimate PUMA 512 block diagram**

It consists of ten components (Corke, 1993):

(1) DEC LS11 computer with ADAC parallel interface board, DLV11-J serial interface board, CMOS board and EPROM board;

(2) Servo interface board;

(3) Six digital servo boards;

(4) Two power amplifiers assemblies;

(5) Power amplifier control board;

(6) Clock/terminator board;

(7) Input/output interface board;

(8) Two power supplies;

(9) High power function board;

(10) Arm cable board.

Description of the important components is given below.

1. DEC LSI-11/2 computer

The DEC LSI-11/2 computer is based on a 16-bit processor with up to 32 k words of memory. This 1970's era processor is used to compute the setpoints, which make up the desired robot joint trajectory. This processor communicates via the DLVII-J serial interface card to a floppy disk drive, a teach pendant, and a serial terminal. The setpoint can be updated by the LSI-11/2 based software at one of several user selectable periods. The default update period is 28 ms.

2. Digital servo boards

There are six digital servo boards, one per joint of the robot manipulator. Each servo board consists of a 6503 8-bit 1 MHz microprocessor with 2408 bytes of EPROM and 128 bytes of RAM. Each processor implements the position loop for its joint at about 1 kHz.

3. VAL-II software

VAL-II software for the Mark II controller allows the user to program robot motions. Once a VAL-II command is issued from the terminal, the LSI-11/2 computer computes the joint level trajectory and sends setpoints to the digital servo boards every 28 ms until the motion is complete.

### Disadvantages

There are several disadvantages or limitations of the VAL-II based Mark II controller system. The obsolete LSI-11/2 processor based computer with its limited computational abilities and memory does not permit a sophisticated trajectory generator to be implemented. The 1 kHz sampling frequency is fairly low by modern standards. Entering programs from a serial terminal and storing them to a low capacity floppy drive is cumbersome for software development. The greatest disadvantage may be the fact that this system does not allow the arbitrary sensors be integrated into the trajectory generator. For example, with a VAL-II based system it is not feasible to implement a visual based servo control architecture.

The original system used a large number of operational amplifiers and discrete components for conditioning the shaft encoder signals and amplifying the analog control voltages. This leaves considerable room for simplifying and compacting the controller design by substituting with modern components.

## NEW HARDWARE CONFIGURATION

The PUMA 512 robot is described in Fig.2. It is a member of the Unimate PUMA 500 series of robots with six joints. A simple feedback loop for a single robot joint is shown in Fig.3, which shows the position and speed loops.

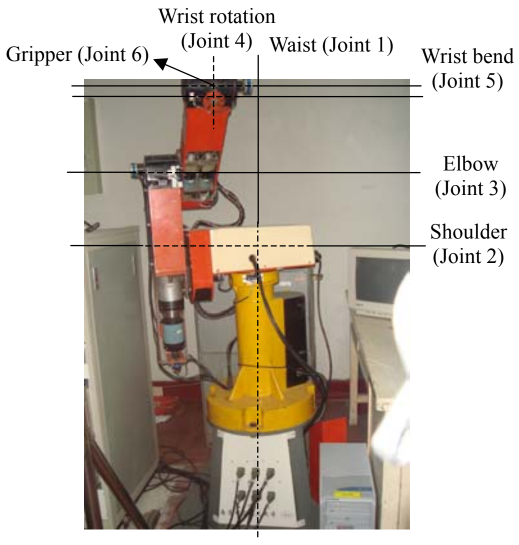


**Fig.2 PUMA robot 512 joints identification**

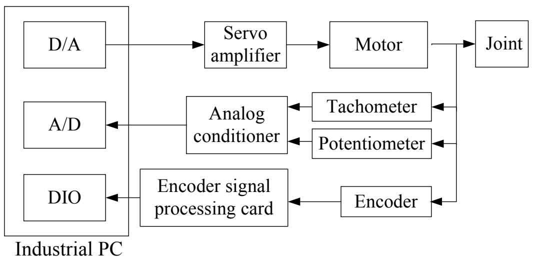


**Fig.3 Feedback loop of a robot joint**

Each of PUMA 512 joints is driven by a gear train with a permanent magnet DC servo motor which incorporates a rotary shaft encoder, a tachometer and a potentiometer. The potentiometers and shaft encoders use a common 5 V supply. The maximum and safe angle of joint movement and the specifications of motors, encoders, potentiometers and tachometers are shown in Table 1.

The new system's block diagram is shown in Fig.4. It mainly consists of a PWM (pulse width modulation) amplifier box, a control unit, a power supply box and an industrial computer. The PWM amplifier box contains 6 in-house built amplifiers employing SA01.

The SA01 amplifier is a PWM amplifier that can supply 2 kW to the load. It operates at 40 V DC control supply and ±15 V DC servo supply. This kind of amplifier technology provides particular benefit in the high power ranges where operating efficiencies as high as 90% can be achieved to dramatically reduce the heat sinking requirements.

The control box includes an internally designed digital conditioner card for shaft encoder signals and an analog conditioner card for potentiometer and tachometer. The in-house built encoder conditioner card uses ALTERA MAX 7256AETC100-10 CPLD (Chartrand, 2004) as shown in Fig.5. It belongs to MAX 7000A programmable device family. The card has 3 CPLDs, each for one shaft encoder.

Shaft encoders are sometimes supplied with an internal capacitor to drain electrical noise. A capacitor placed between the encoder case and the encoder electronics will couple this noise into the encoder or encoder wiring, where it can interfere with the normal operation. So, the encoders used in our design have no internal capacitor to avoid the interference between PWM motor drive currents and low current shaft encoders.

**Table 1 Specification of PUMA robot**

| Parameters | Joint | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Maximum operating range | ±150° | ±150° | ±150° | ±150° | ±180° | ±360° |
| Safe operating range | ±90° | ±90° | ±60° | ±90° | ±90° | ±180° |
| Servo motor | 90SZ53, 3000 rpm. 150 W, 110 V, 1.8 A | | | 55SZ53, 3000 rpm, 29 W, 48 V, 1.1 A | | 250 rpm, 15 W, 24 V, 0.6 A |

For all the 6 joints: Rotary encoder: DC +5 V, 600 P/R; Tachometer: CYH7-1.7 V/1000 rpm; Potentiometer: WHJ 9k ± 0.1%

The robotic arm needs two digital conditioner cards. The CPLDs are programmed using VHDL language. The signals A+, A−, B+, B−, Z+ and Z−, VCC and DGND are the eight signals from rotary shaft encoder which are interfaced to the CPLD via a differential line receiver MC3486. The 24 signals (D0_waist to D23_waist) go to 6-channel 722 DIO card. The other 5 joints' shaft encoders are connected to digital conditioner card in the same way.

A significant advantage of CPLDs is that they provide simple design through reprogramming. Unlike the commercially available encoder conditioner cards, the newly developed CPLD card is reconfigurable.

The power supply unit incorporates power supplies for PWM amplifier units, signal conditioner cards and an excitation 110 V power supply for 6 servo motors.

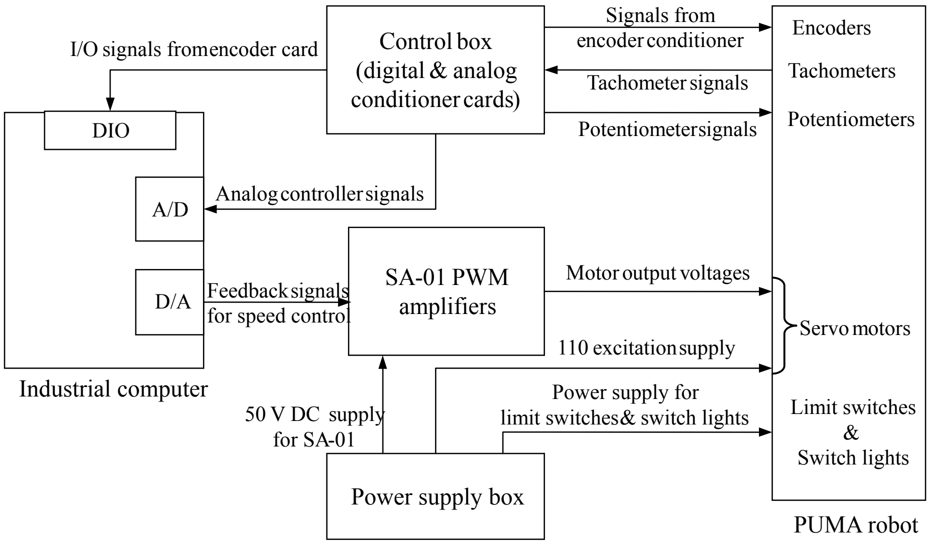

Control box (digital & analog conditioner cards)
I/O signals fromencoder card
DIO
A/D
D/A
Analog controller signals
Feedback signals for speed control
Industrial computer
SA-01 PWM amplifiers
50 V DC supply for SA-01
Power supply box
Signals from encoder conditioner
Tachometer signals
Potentiometer signals
Motor output voltages
110 excitation supply
Power supply for limit switches& switch lights
Encoders
Tachometers
Potentiometers
Servo motors
Limit switches & Switch lights
PUMA robot

**Fig.4 Schematic diagram of the new robot hardware**

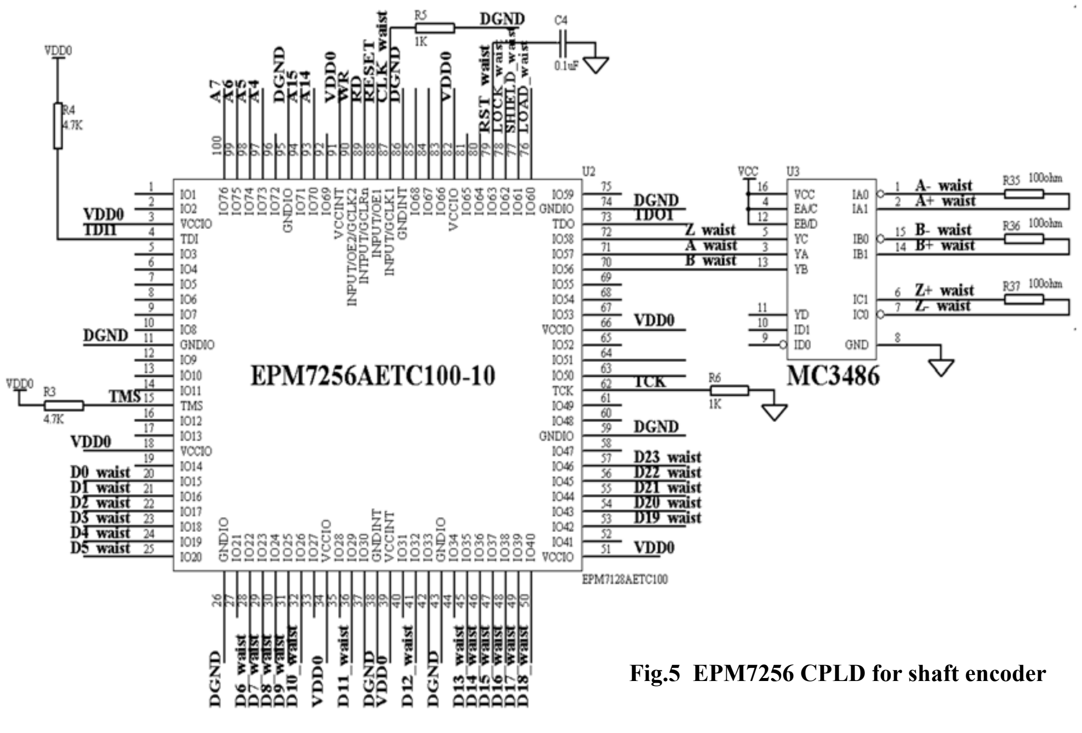


**Fig.5 EPM7256 CPLD for shaft encoder**

A Pentium IV industrial computer is used as a central controller. It has one 6-channel 722 DIO card, 16-bit 816 A/D and 6126 D/A cards.

The analog signals from tachometers and potentiometers are fed into a 6-channel analog conditioner card, which was designed in Simulation & Machine Control lab (S & MC). After conditioning, the signals are fed to an industrial PC (A/D card). The analog feedback signals from D/A are provided to PWM amplifiers for each joint to complete the speed loop.

**Advantages for new PC-based PUMA robot**

A new PC-based PUMA robot manipulator control system has three advantages: simplicity, flexibility and low cost. The hardware and software complexity of the Unimate Mark II robot controller is extensively reduced as discussed in Section 3. Flexibility refers to the ability to implement arbitrary control strategies which can easily integrate sensor information into low-level control. Flexibility also refers to the ability to easily use wide variety of sensors in the trajectory generator. The suggested PC-based platform has the ability to easily integrate such sensors as sonars, ranging lasers, cameras, etc. that would allow the user to implement complex control strategies (e.g. vision based control). With the current digital servo boards in the Mark II, none of these advanced modes are feasible.

Another potential advantage of the suggested platform is its cost effectiveness. Due to economies of scale and increased CPU power, personal computers have become viable and cost effective alternatives to workstations in engineering applications. An Intel processor easily outperforms a Spark IPX while costs less than a thousand dollars. To further elaborate, a short quotation of our suggested controller is given in the Appendix.

## DESCRIPTION OF CONTROL SCHEME

In this work, the reference torque for each joint of the arm is calculated using computed torque control (CTC) (Lewis *et al.*, 2004). This technique is used to remove the non-linearities of the PUMA by employing feedback linearization. The arm dynamics are given by

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}})+\boldsymbol{\tau}_{\mathrm{d}}=\boldsymbol{\tau}, \tag{1}$$

where $\boldsymbol{q}(t)\in\mathbb{R}^6$ is a vector of joint variables, $\boldsymbol{\tau}(t)\in\mathbb{R}^6$ is the control torque, $\boldsymbol{\tau}_{\mathrm{d}}(t)\in\mathbb{R}^6$ is a disturbance, $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}})$ represents nonlinear terms including coriolis/centripetal effects, friction and gravity.

Suppose that a reference trajectory $\boldsymbol{q}_{\mathrm{d}}(t)$ has been chosen for the arm motion. The tracking error is defined as

$$\boldsymbol{e}(t)=\boldsymbol{q}_{\mathrm{d}}(t)-\boldsymbol{q}(t). \tag{2}$$

If the tracking error is differentiated twice, then

$$\ddot{\boldsymbol{e}}=\ddot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{M}^{-1}(\boldsymbol{N}+\boldsymbol{\tau}_{\mathrm{d}}-\boldsymbol{\tau}). \tag{3}$$

The feedback input linearizing function may be defined as

$$\boldsymbol{u}=\ddot{\boldsymbol{q}}_{\mathrm{d}}+\boldsymbol{M}^{-1}(\boldsymbol{N}-\boldsymbol{\tau}), \tag{4}$$

and the disturbance function as

$$\boldsymbol{w}=\boldsymbol{M}^{-1}\boldsymbol{\tau}_{\mathrm{d}}. \tag{5}$$

Then the tracking error dynamics can be expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix}=\begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}_{6\times6}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix}+\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}_{6\times6}\end{bmatrix}\boldsymbol{u}+\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}_{6\times6}\end{bmatrix}\boldsymbol{w}. \tag{6}$$

Hence, as a result of using the feedback linearization transformation Eq.(4), the tracking error dynamics are given by a linear state equation with constant coefficients in Eq.(6).

The feedback linearization transformation can be inverted to give

$$\boldsymbol{\tau}=\boldsymbol{M}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_{\mathrm{d}}-\boldsymbol{u})+\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}}). \tag{7}$$

This is the computed torque control law. An outer loop controller is often used. The role of the outer loop controller is to provide the input $\boldsymbol{u}$. In this paper, PD (proportional-plus-derivative) computed torque controller has been used as an outer controller.

**PD outer-loop design**

The outer-loop signal $\boldsymbol{u}(t)$ is selected as the PD feedback:

$$\boldsymbol{u}=-\boldsymbol{K}_{\mathrm{D}}\dot{\boldsymbol{e}}-\boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}, \tag{8}$$

where $\boldsymbol{K}_\mathrm{P}$ is proportional gain, $\boldsymbol{K}_\mathrm{D}$ is differential gain.

Thus the overall robot arm input becomes

$$\boldsymbol{\tau}=\boldsymbol{M}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_\mathrm{d}+\boldsymbol{K}_\mathrm{D}\dot{\boldsymbol{e}}+\boldsymbol{K}_\mathrm{P}\boldsymbol{e})+\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}}). \tag{9}$$

The closed-loop error dynamics are

$$\ddot{\boldsymbol{e}}+\boldsymbol{K}_\mathrm{D}\dot{\boldsymbol{e}}+\boldsymbol{K}_\mathrm{P}\boldsymbol{e}=\boldsymbol{w}, \tag{10}$$

or in the state space form

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix}=\begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}_{6\times6}\\ -\boldsymbol{K}_\mathrm{P} & -\boldsymbol{K}_\mathrm{D}\end{bmatrix}\begin{bmatrix}\boldsymbol{e}\\ \dot{\boldsymbol{e}}\end{bmatrix}+\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}_{6\times6}\end{bmatrix}\boldsymbol{w}. \tag{11}$$

The closed loop characteristic polynomial is

$$\Delta_\mathrm{c}(s)=\left|s^2\boldsymbol{I}+\boldsymbol{K}_\mathrm{D}s+\boldsymbol{K}_\mathrm{P}\right|. \tag{12}$$

**Choice of PD gains**

It is usual to take the $n\times n$ gain matrices diagonal so that

$$\boldsymbol{K}_\mathrm{D}=\mathrm{diag}\{K_{\mathrm{D}i}\},\quad \boldsymbol{K}_\mathrm{P}=\mathrm{diag}\{K_{\mathrm{P}i}\}. \tag{13}$$

Then, using Eqs.(12) and (13),

$$\Delta_\mathrm{c}(s)=\prod_{i=1}^{n}\left(s^2+K_{\mathrm{D}i}s+K_{\mathrm{P}i}\right). \tag{14}$$

And the error system is asymptotically stable as long as $\boldsymbol{K}_\mathrm{D}$ and $\boldsymbol{K}_\mathrm{P}$ are all positive. Therefore as long as the disturbance is bounded, so is the error $\boldsymbol{e}(t)$.

The standard form for the second-order characteristic polynomial is

$$p(s)=s^2+2\zeta\omega_\mathrm{n}s+\omega_\mathrm{n}^2, \tag{15}$$

with $\zeta$ being the damping ratio and $\omega_\mathrm{n}$ the natural frequency. Therefore, desired performance in each component of the error $\boldsymbol{e}(t)$ may be achieved by selecting the PD gains as

$$K_{\mathrm{D}i}=\omega_\mathrm{n}^2,\quad K_{\mathrm{P}i}=2\zeta\omega_\mathrm{n}. \tag{16}$$

It is undesirable for the robot to exhibit overshoot, since this could cause impact if, for instance, a desired trajectory terminates at the surface of a workpiece. Therefore, the PD gains are selected for critical damping $\zeta$=1. In this case, Eq.(16) becomes

$$K_{\mathrm{D}i}=2\sqrt{K_{\mathrm{P}i}},\quad K_{\mathrm{P}i}=K_{\mathrm{D}i}^2/4. \tag{17}$$

The tuning of the CTC controller for PUMA robot using Eq.(17) leads to the gains as shown in Table 2. The CTC technique is known to perform well when the robotic arm parameters are known fairly accurately. Fortunately, the dynamics of PUMA 560 manipulator are well known and have been reported. The inverse dynamics and Denavit-Hartenberg arm parameters employed in this work are those reported in (Maciejowski, 2002; Lewis *et al.*, 2004).

**Table 2 Controller gains for Joints 1~6**

| | Joint | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| $\boldsymbol{K}_\mathrm{P}$ | 18000 | 10000 | 8900 | 6200 | 10500 | 5500 |
| $\boldsymbol{K}_\mathrm{D}$ | 19 | 16 | 16 | 14 | 18 | 16 |

## SOFTWARE DESIGN FOR CONTROLLER

To implement the control algorithms developed in Section 3, a real-time software was developed using C++/VC++. The GUIs developed for robot are shown partially in Fig.6 and Fig.7.

Fig.6 shows different options for robot control. The "Position-Control" and "Rate-Control" are used to control the 6 joints' position and speed respectively. The "Signal-Generator" is designed mainly for testing the robot position-trajectory performance. The "Data-View" and "Data-Curve" display the joints position and speed data. Fig.7 demonstrates "Position-Control" window only.

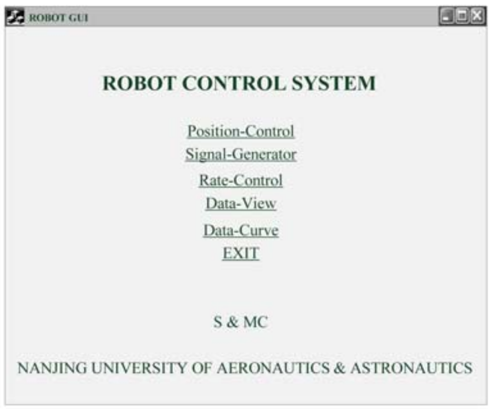
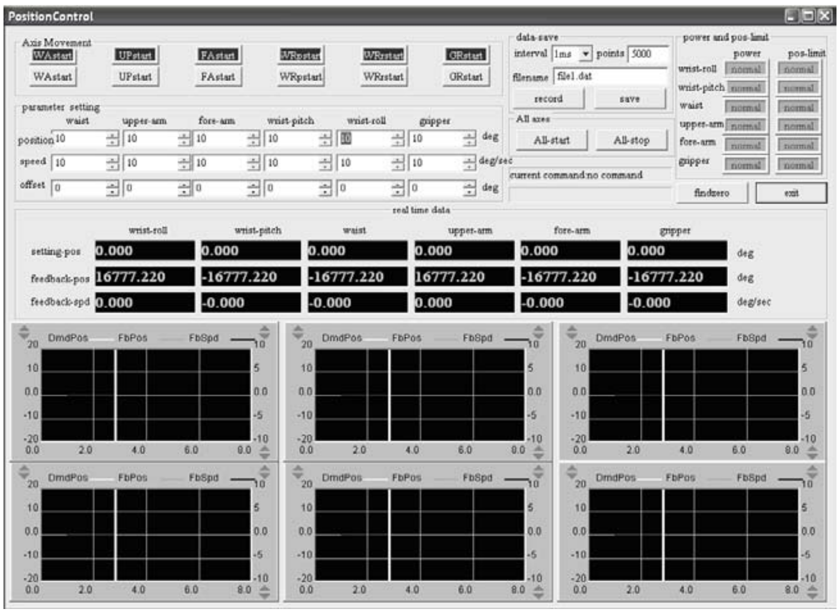


**Fig.6 GUI of PC-based PUMA robot**


PositionControl

Axis Movement

| | | | | | |
|---|---|---|---|---|---|
| WAstart | UPstart | FAstart | WRpstart | WRrstart | GRstart |
| WAstart | UPstart | FAstart | WRpstart | WRrstart | GRstart |

parameter setting

| | waist | upper-arm | fore-arm | wrist-pitch | wrist-roll | gripper | |
|---|---|---|---|---|---|---|---|
| position | 10 | 10 | 10 | 10 | 10 | 10 | deg |
| speed | 10 | 10 | 10 | 10 | 10 | 10 | deg/sec |
| offset | 0 | 0 | 0 | 0 | 0 | 0 | deg |

data-save: interval 1ms, points 5000, filename file1.dat, record, save

All axes: All-start, All-stop

current command:no command

power and pos-limit

| | power | pos-limit |
|---|---|---|
| wrist-roll | normal | normal |
| wrist-pitch | normal | normal |
| waist | normal | normal |
| upper-arm | normal | normal |
| fore-arm | normal | normal |
| gripper | normal | normal |

findzero, exit

real time data

| | wrist-roll | wrist-pitch | waist | upper-arm | fore-arm | gripper | |
|---|---|---|---|---|---|---|---|
| setting-pos | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | deg |
| feedback-pos | 16777.220 | -16777.220 | -16777.220 | 16777.220 | -16777.220 | -16777.220 | deg |
| feedback-spd | 0.000 | -0.000 | -0.000 | 0.000 | -0.000 | -0.000 | deg/sec |

**Fig.7 Robot Position-Control GUI layout**

**Control buttons**

The different control buttons located in the GUI are listed below.

Find zero: This button causes the robot to return to its zero position. Before running any programs, the arm should be sent back to zero position.

Axes movement: There are six control buttons for controlling the joint movement, each for one axis. To move one or more joints simultaneously to the specified positions, the relevant control button for that axis is used to start or stop the motion.

Gripper control: The gripper can be opened and closed using GRstart and GRstop buttons, respectively.

Parameter setting: The user can set the desired joint angular position, speed and offset.

Real-time data: The user can view the current joint angular position and speed in 'Position-Control' and 'Rate-Control' interfaces respectively.

Scope: A noteworthy feature of our GUI is the 'scope' for each axis. The scope gives the graphical interpretation of real-time joint movement. It can also be used effectively for trouble shooting during and after the robot design.

Parameter setting in 'Signal-Generator' Mode: To check the real-time performance of robot arms, various parameter setting options are given in 'Signal-Generator' interface, i.e. signal, amplitude, frequency and offset.

Data save: The 'Data save' button in the "Position-Control" interface is used to save data for one or more joints and display them in 'Data-View' and 'Data-Curve' in the main GUI.

## RESULTS

The suggested system was developed and applied to PUMA 560 robot. The original proprietary controller was replaced with it. Fig.8 depicts the experimental layout of the developed system. The PC runs the Windows 2000 operating system.

The PC-based controller is evaluated from two different aspects. The first aspect is to examine how easy the system integrations and modifications are. The second is to examine the performance of the control. The first aspect is obvious. The suggested PC-based hardware and software ensure that the extensibility and scalability are available. The second aspect is evaluated by the trajectory-tracking experiment.

To verify the effectiveness of the new controller, experiments were performed to test the tracking control of the robot manipulator. Firstly, each joint is separately requested to follow a desired trajectory. In this test, each joint is asked to move to a specified destination while following a predetermined path. Fig.9 and Fig.10 show the desired position

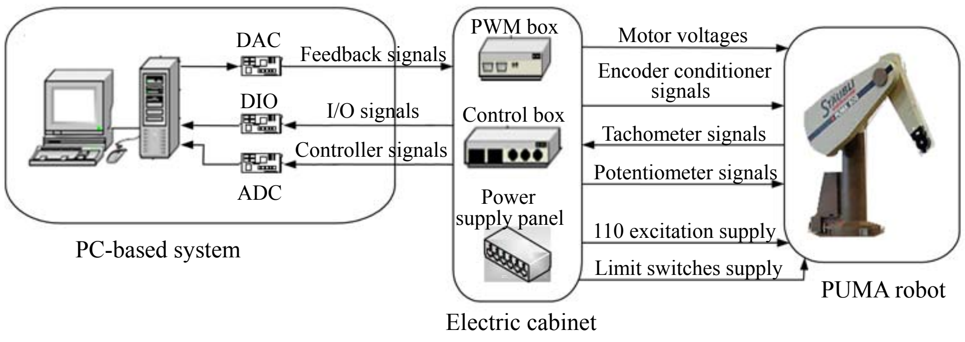


**Fig.8 The experimental platform for suggested open architecture controller**

trajectories and position tracking errors respectively for six joints. The desired profile consists of two parts: (a) linear acceleration from rest to maximum velocity for joints 1, 2, 3 and 5 while linear deceleration for joints 4 and 6; (b) Decelerate linearly to rest in 400 ms. The position tracking errors of joints 1, 3, 4, 5 and 6 are in the acceptable range of 0.005 rad to 0.01 rad except joint 2 for which the tracking error is 0.02 rad but still in the desirable range. All these tests were performed with PD gains as shown in Table 2. A careful selection of $\boldsymbol{K}_D$ and $\boldsymbol{K}_P$ is very important.

Two further tests were performed to examine the simultaneous joints movement: (1) The maximum velocity was set at 15000 counts/s. To test the simultaneous joints movement, all six joints were asked to move at their fastest speeds. (2) The linear interpolation mode test in which the individual joints should arrive at their respective destinations at the same time.

The same tests were performed with varying joints' velocities. Joints 1 and 3 showed higher position tracking errors at higher velocities, however, all the remaining joints showed satisfactory performance at high velocities. All the joints show satisfactory performance at low velocity and exhibit low position tracking error while following a velocity profile at high speeds.

The control method CTC used in this paper is a scheme for cancelling the nonlinearities in the dynamics to yield a linear error system. It works well if all the parameters of the robotic arm are known exactly. However, the PUMA robot in S & MC lab may have some divergence from the standard parameters due to the effect of aging. This minor deviation in the parameters adds further to position tracking error.

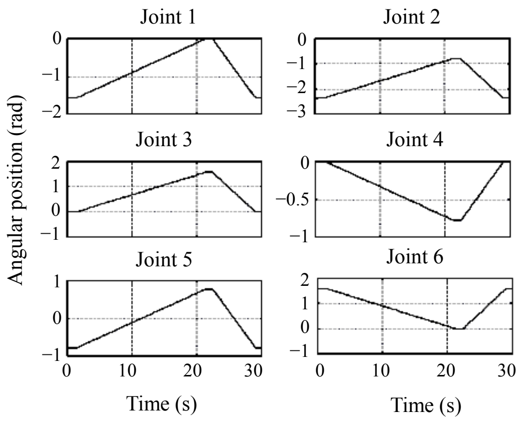


**Fig.9 Desired position trajectories**

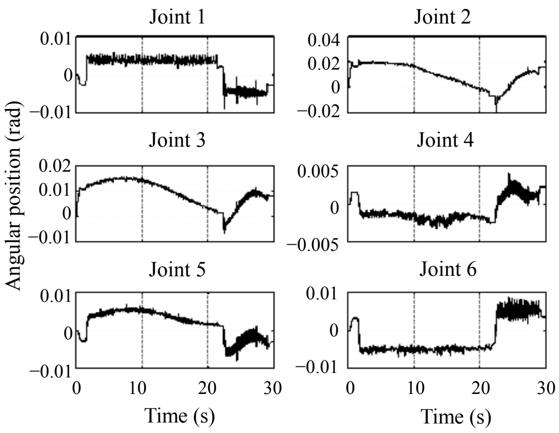


**Fig.10 Position tracking error**

## CONCLUSION AND DISCUSSION

In this paper, the preliminary results are achieved in the development and implementation of a new simple PC-based replacement controller for programmable universal machine for assembly (PUMA) 512 robot. By assembling controller from the off-the-shell hardware and software components, the benefits of reduced and improved robustness have been realized.

Although the experiments were performed at educational and research institution, the research is oriented towards industrial applications.

The software graphical user interface for the robot was developed using VC++. It encompasses all the features needed to control an industrial robot.

The experimental results show that it is feasible to implement modern control methods for PUMA 500 series robots through the software routines running on a PC.

Presently the authors are working on implementing and adapting the system for an industrial CNC milling machine.

## ACKNOWLEDGEMENT

The authors would like to thank Miss Ting-ting Bai who helped us in robot hardware design and Mr. Guo-hai Ye for his contribution in mechanical design and assembly. We would also like to thank Dr. Huang for helping us in debugging the GUI software in VC++.

## References

Becerra, V.M., Cage, C.N.J., Harwin, W.S., Sharkey, P.M., 2004. Hardware retrofit and computed torque control of a Puma 560 robot updating an industrial manipulator. *IEEE Control Systems Magazine*, **24**(5):78-82. [doi:10.1109/MCS.2004.1337867]

Chartrand, L., 2004. Advanced Digital Systems and Concepts with CPLD's (1st Ed.). Thomson Delmar Learning.

Corke, P.I., 1993. Operational Details of the Unimation Puma Servo System. Report, CSIRO Division of Manufacturing Technology. Australia.

Fiedler, P., Schlib, C., 1998. Open architecture robot controllers and workcell integration. *Robotics Today*, **11**(4):1-4.

Katupitiya, J., Radajewski, R., Sanderson, J., Tordon, M., 1997. Implementation of a PC Based Controller for a PUMA Robot. Proc. 4th IEEE Conf. on Mechatronics and Machine Vision in Practice. Australia, p.14-19. [doi:10.1109/MMVIP.1997.625229]

Leahy, M.B.Jr., Petroski, S.B., 1994. Unified Telerobotic Architecture Project Status Report. IEEE Int. Conf. Systems, Man and Cybernetics. San Antonio, Texas, **1**:249-253.

Lewis, F.L., Abdallah, C.T., Dawson, D.M., 2004. Robot Manipulator Control: Theory and Practice (2nd Ed.). Marcel Dekker.

Maciejowski, J.M., 2002. Predictive Control with Constraints (1st Ed.). Prentice Hall.

Miller, D.J., Lenox, R.C., 1991. An object oriented environment for robot system architecture. *IEEE Control System Magazine*, **11**(2):14-23. [doi:10.1109/37.67671]

Pan, L.D., Huang, X.H., 2004. Implementation of a PC-based Robot Controller with Open Architecture. Proc. IEEE Int. Conf. on Robotics and Biometrics, p.790-794.

Sorenson, S., 1993. Overview of Modular, Industry Standard Based Open Architecture Controller. Proc. Int. Conf. Robots and Vision Automation. Detroit Michigan, USA.

Unimation Robotics. 1983. User's Guide to VAL 398P2A: A Robot Programming and Control System. Unimation Inc., Danbury, Connecticut, USA.

Vistness, R., 1982. Breaking away from VAL. Technical Report, Unimation Inc. Danbury, Connecticut, USA.

## APPENDIX

A short quotation of our suggested controller is given here.

**Table A1 Quotation of the suggested controller**

| No. | Item | Qty. | Price (US$) | Total price (US$) |
|---|---|---|---|---|
| 1 | PWM amplifier card | 6 | 120 | 720 |
| 2 | Digital conditioner card | 2 | 310 | 620 |
| 3 | Analog conditioner card | 1 | 70 | 70 |
| 4 | 722 DIO card | 1 | 150 | 150 |
| 5 | 816 ADC card | 1 | 295 | 295 |
| 6 | 6126 DAC card | 1 | 310 | 310 |
| 7 | Power supply card | 3 | 70 | 210 |
| 8 | Industrial PC, P-IV | 1 | 750 | 750 |
| 9 | Miscellaneous components & cards | | 65 | 65 |
| 10 | Software (Windows NT, Visual C++) | 1 | 550 | 550 |
| | Total cost | | | 3740 |