Proceedings of the World Congress on Engineering and Computer Science 2007
WCECS 2007, October 24-26, 2007, San Francisco, USA

A New Approach to Implementation of an Open Architecture Controller for a PUMA Robot

Farooq M, *Member, IAENG*, Wang Dao Bo and Dar N.U.

Abstract- **The paper describes the replacement of the controller for a PUMA 512 robot with a newly designed PC based controller employing real-time direct control of the six joints. The original structure of the PUMA robot has been retained. The hardware of the new controller includes in-house designed: PWM amplifiers, digital and analog controllers. The system uses digital I/O cards; signal conditioner cards for force sensor at end effector and tachometers; and 16 bits A/D and D/A boards. An Intel Pentium IV industrial computer is used as the central controller. The control software has been implemented using VC++ programming language. The trajectory tracking results show the validity of the new PC based controller.**

Index terms- **Computed torque control (CTC), Graphical user Interface (GUI), Pulse width modulation (PWM) amplifier, PUMA robot,**

I. 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.

The Unimate PUMA 500 series Robots mainly uses DEC LSI 11 processor running VAL robot control software [1]. Methods of bypassing VAL are discussed in literature, including Unimation technical reports [2] [3]. However, most of these procedures have been confined to replacing the LSI 11 with another DEC computer, leaving peripheral hardware intact. In fact, it is far more cost effective to develop new hardware using less specific interfaces. The shift towards personal computer open architecture robot controller and the impact of using these newer controllers for system integration is discussed in [4]. An improved PC-based design for Puma robot was presented in [5], 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 personal computer, in-house as well as specialized hardware. Some technical problems in the previous design for velocity test profile of joints1, 2 and 4 have also been addressed. The joints position tracking error at high velocities are also minimized in our design.

The work was supported by S & MC lab, College of Automation Engineering, Nanjing University of Aeronautics & Astronautics.

Farooq M is with Nanjing University of Aeronautics & Astronautics, Nanjing PR China as a PhD student. (Phone: 86-13851942994, fax: 86-25-84498069, e-mail: muh_farooq1974@yahoo.com)

Wang Dao Bo is with Nanjing University of Aeronautics & Astronautics, Nanjing PR China as a Professor. (E-mail: dbwangpe@nuaa.edu.cn)

Dar N. U. is with University of Science & Technology, Taxila, Pakistan as a PhD student. (E-mail: nudar@hotmail.com)

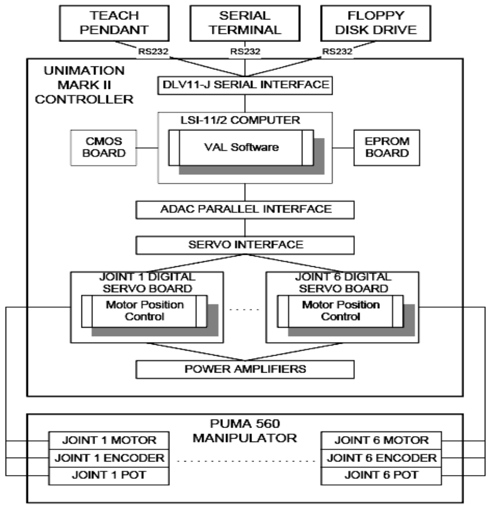

Fig.1. Unimate Puma 512 block diagram

II. UNIMATE PUMA 500

The Unimation Mark II is an industrial robot controller as shown as in Fig.1. It consists of ten components [6]:

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.

The original system used a large number of operational amplifiers and discrete components for conditioning of shaft encoder signals and amplification of analog control voltages. This leaves considerable scope to simplify and compact the controller design by substitution of more modern components.

II. NEW HARDWARE CONFIGURATION

The PUMA 512 robot used for work is described in Fig.2, is a member of the Unimate PUMA 500 series of Robots, having six joints.

Each of PUMA 512 joints is driven by through a gear train by a permanent magnet DC servo motor which incorporates a rotary shaft encoder, a tachometer and a potentiometer. The single turn potentiometer, which is housed in the end of each motor, is driven by planetary gears so that the potentiometer will move less than one revolution during the full range of movement of the joint. The maximum and safe angle of joint movement and specifications of motors, encoders, potentiometers and tachometers are shown in table 1.

The new system's block diagram is shown in the Fig.3. The PWM amplifier box contains 6 in-house built amplifiers employing SA01. The SA01 amplifier is a pulse width modulation amplifier that can supply 2KW to the load. The full bridge output amplifier can be operated from a single power supply over a wide range of voltages. An error amplifier is included which can provide gain for the velocity control loop.

The control box includes an internally designed digital conditioner card for shaft encoder's signals and an analog conditioner card for potentiometer and tachometer. The in-house built encoder conditioner card uses ALTERA MAX 7256AETC100-10 CPLD as shown in the Fig.4. It belongs to MAX 7000A programmable device family. The card has 3 CPLDs one for each shaft encoder.

Shaft encoders are sometimes supplied with an internal capacitor from circuit common to case ground to drain electrical noise from common to building ground. However

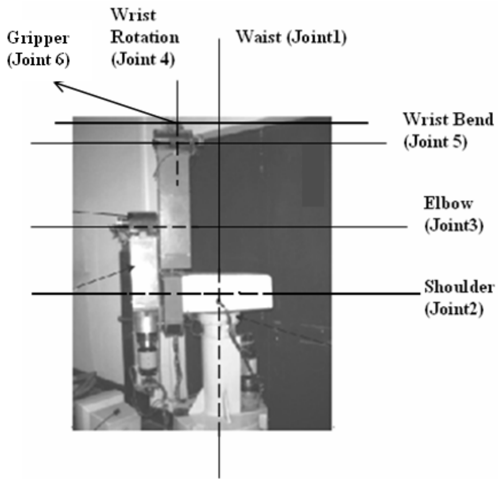

Fig.2 Puma 512 Robot

PWM drives have high frequency noise that is coupled to the frame and shaft of the motor. 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.

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 signals D0_waist to D23_waist go to 6 channels 722 DIO card. The other 5 joints' shaft encoders are connected

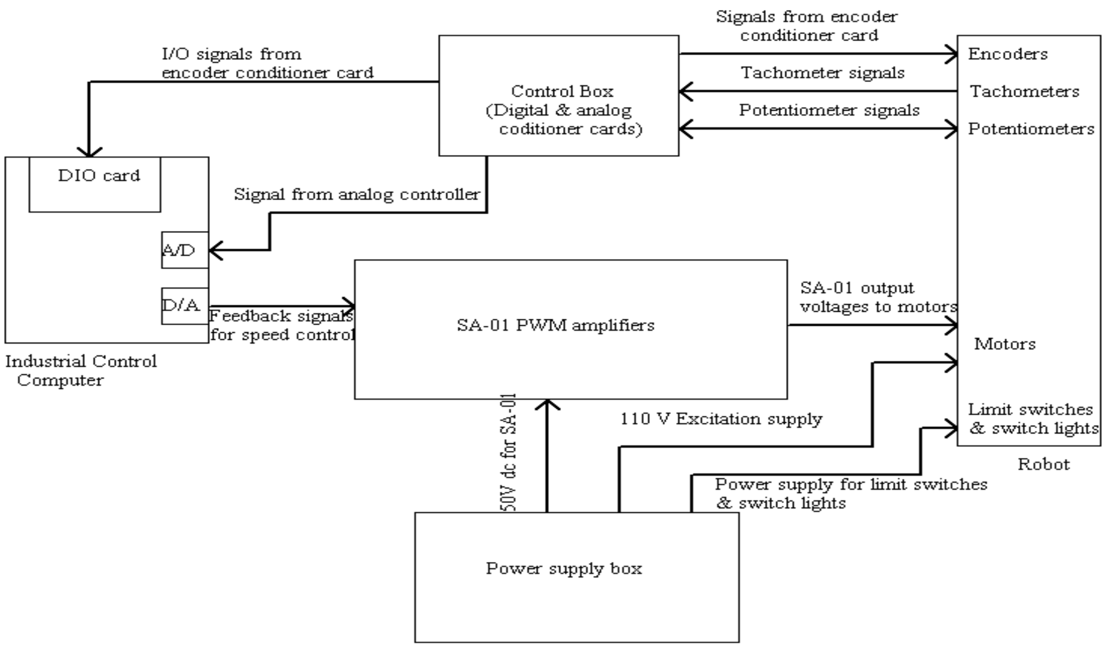

Fig.3 Schematic diagram of new robot hardware

ISBN:978-988-98671-6-4

TABLE 1
SPECIFICATIONS OF PUMA ROBOT

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Maximum angle of joint movement	±150º	±150º	±150º	±150º	±150º	
Safe angle of joint movement	±90º	±90º	±60º	±90º	±150º	
Rotary encoder	DC+5V, 600P/R					
Servo motor	90SZ53, 3000RPM. 150W, 110 V, 1.8A.			55SZ53, 3000RPM, 29W, 48V, 1.1A		2500 RPM ,15W ,24V, 0.6A
Tachometer	CYH7-1. 7V/1000RPM					
Potentiometer	WHJ 9K ± 0.1%					

to digital conditioner card in the same way

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

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

The analog signals from tachometers and potentiometers are fed into the analog conditioner card. The card was designed in Simulation & Machine Control lab (S & MC). After conditioning the signals, they are fed to 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.

III. DESCRIPTION OF THE CONTROL SCHEME

In this work, the reference torque for each joint of the arm is calculated using computed torque control [7]. This technique is used to remove the nonlinearities of the PUMA

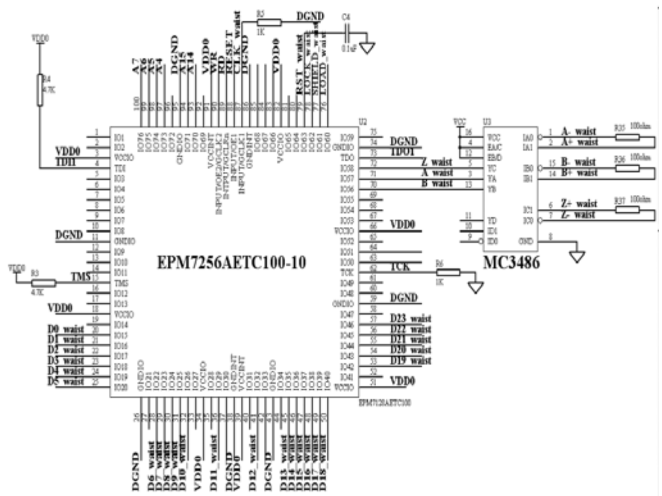

Fig.4 EPM7256 CPLD for shaft encoder

by employing feedback linearization. The arms dynamics are given by

$$M(q)\ddot{q} + N(q,\dot{q}) + \tau_d = \tau \quad (1)$$

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

Suppose that a reference trajectory $q_d(t)$ has been chosen for the arm motion. The tracking error is defined as:

$$e(t) = q_d(t) - q(t) \quad (2)$$

If the tracking error is differentiated twice, then

$$\ddot{e} = \ddot{q}_d + M^{-1}(N + \tau_d - \tau) \quad (3)$$

The feedback input linearizing function may be defined as:

$$u = \ddot{q}_d + M^{-1}(N - \tau) \quad (4)$$

and the disturbance function as

$$w = M^{-1}\tau_d \quad (5)$$

Then the tracking error dynamics can be expressed as

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I_{6\times6} \\ 0 & 0 \end{bmatrix}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I_{6\times6} \end{bmatrix} u + \begin{bmatrix} 0 \\ I_{6\times6} \end{bmatrix} w \quad (6)$$

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

The feedback linearization transformation can be inverted to give

$$\tau = M(q)(\ddot{q}_d - u) + N(q,\dot{q}) \quad (7)$$

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 *u*. In this paper, PID computed torque controller has been used as an outer controller. The stabilization of (6) is not difficult. In fact, the nonlinear transformation (4) has converted a complicated nonlinear controls design problem into a simple design problem for a linear system consisting of 6 decoupled subsystems, each obeying Newton's laws. The resulting control scheme is shown in Fig. 5.

The computed torque control technique is known to

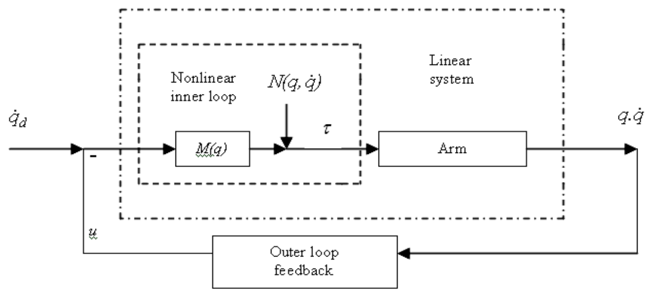

Fig. 5 CTC Control scheme

perform well when the robotic arm parameters are known fairly accurately. Fortunately, the dynamics of PUMA 560 manipulator are well known and reported. The inverse dynamics and Denavit-Hartenberg arm parameters employed in this work are those reported in [7] and [8].

IV. SOFTWARE DESIGN FOR THE CONTROLLER

To implement the control algorithms developed in section III, real time software was developed using C++/ VC++ [9]. The graphical user interface developed for robot has been shown partially in the Fig. 6 and Fig.7.

The Fig.6 shows different options for robot control. The "Position-Control" and "Rate-Control" are used to control the robot 6 joints' position and speed respectively. The "Signal-Generator" is designed mainly for testing the robot position trajectory performance. "Data-View" and "data-Curve" can display the joints position and speed data. Fig. 7 demonstrates "Position-Control" window only. It has control button for finding zero, parameters setting, displaying desired and feedback position. The six joints can be 'started' or 'stopped' using 6 control buttons. The gripper start or stop button will open or close it. The 'find

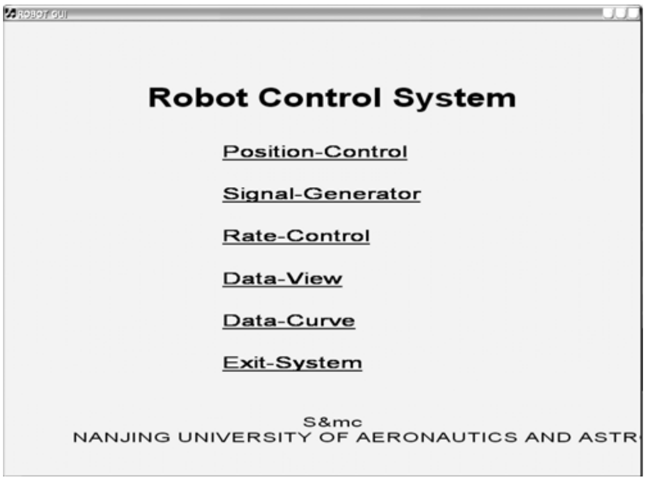

Fig.6. GUI of PC based PUMA Robot

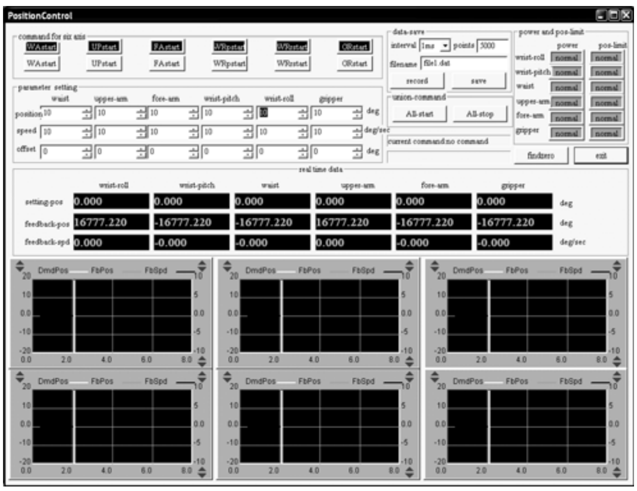

Fig.7. Robot Position-control GUI layout

zero' button will cause the PUMA robot to come to its zero position. The 'data curve' button in the "position-control" window is used to save data for one or more joints and display them in "data view" and 'data curve' in the main GUI.

V. RESULTS:

To verify the effectiveness of the new controller, some 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. The same tests were performed with varying joints' velocities. The maximum velocity was set to 15000 counts/sec. To test the simultaneous joints movement, all the six joints are asked to move at their fastest respective speeds. Fig. 8 and Fig.9 show the desired position trajectories and position tracking errors respectively for six joints. The position tracking errors of all six joints are quite satisfactory. Joint 1 and joint 3 showed higher position tracking errors at higher velocities, however, all the remaining joints showed satisfactory performance at high velocities.

VI. DISCUSSION & CONCLUSIONS

In this paper, the preliminary results achieved in the development and implementation of a new simple PC based replacement controller for PUMA 512 robot.

The control method, CTC, used in this paper, is a scheme for canceling 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. One of the reasons in position tracking error may also be some variance in those parameters as the PUMA robot at S & MC lab, which is used in this experiment, may have some divergence from the standard parameters because of aging.

Though, some technical problems were faced while performing the tests at higher velocities for joint 1 and joint 3, the new designed hardware and software works very well and overcomes the problems in the previous PC based design for PUMA robot. All the joints show satisfactory performance at low velocity as well as they exhibit low position tracking error while following a velocity profile at high speeds.

The previous work in this field relies only on the in-house built hardware. Our work uses both commercially available and in-house built hardware, as discussed in section I, to develop a high performance, modular yet cost effective design. The interference between PWM motor drive currents and shaft encoder signals has been avoided as confronted in the previous design. This is achieved by using shaft encoders without internal capacitors between electrical common and the case. Also, the shaft encoder casing was insulated from the motor to minimize encoder bearings currents and ground noise.

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 showed that it is feasible to

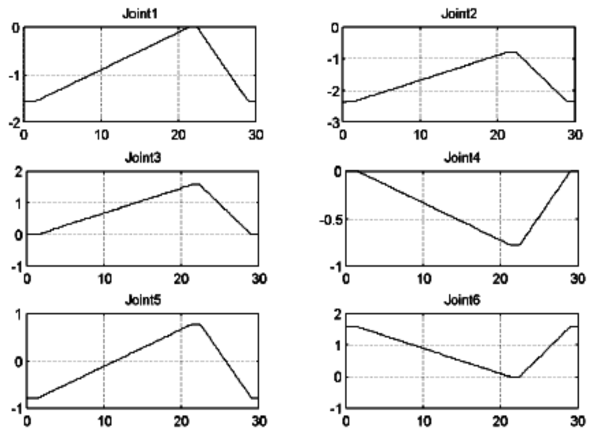

Fig.8 Desired position trajectories (Radians vs. Seconds)

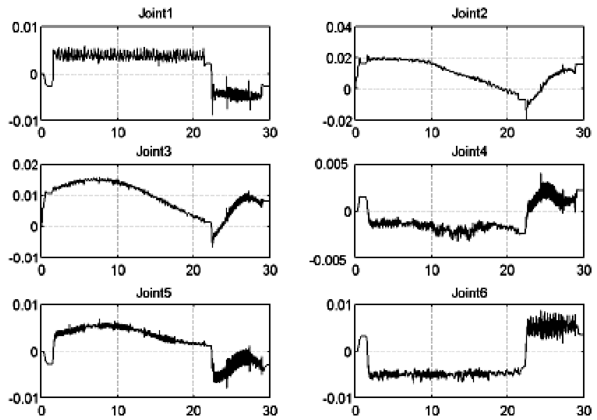

Fig.9 Position tracking error (Radians vs. Seconds)

implement modern control methods for PUMA 500 Series Robots through software routines running on a PC.

ACKNOWLEDGMENTS:

The authors would like to thank to Miss Bai Tingting who helped us in robot hardware design and Mr. Ye Guohai 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

[1] Unimation Robotics, “User’s Guide to VAL 398P2A: A Robot Programming and Control System,” Unimation Inc., Danbury, Connecticut, USA, 1983.
[2] R. Vistness, “Breaking away from VAL. Technical Report,” Unimation Inc. Danbury, Connecticut, USA, 1982.
[3] Unimation Robotics., “The Updated Breaking Away from VAL,” Technical report. Unimation Inc. Danbury, Connecticut, USA, 1982.
[4] P. Fiedler & C. Schlib, “Open Architecture Robot Controllers and Workcell Integration,” Robotics Today, Vol. 11, no.4, PP. 1-4, 1998.
[5] J. Katupitiya, R. Radajewski, J. Sanderson & M. Tordon, “Implementation of a PC based controller for a PUMA Robot,” Proc. IEEE Conference on Mechatronics and Machine Vision in Practice, 23-25, 1997, Australia, PP: 14-19.
[6] P.I. Corke, “Operational Details of the Unimation Puma Servo System,” Report, CSIRO Division of Manufacturing Technology, 1993, Australia.
[7] F.L. Lewis, C.T. Abdallah & D.M. Dawson. “Robot Manipulator Control, Theory and Practice,” 2nd ed., Marcel Dekker, 2004.
[8] J. M. Maciejowski, “Predictive Control with Constraints,” Prentice Hall, 2002.
[9] D. Chapman & J. Heaton, “Visual C++ in 21 days," Ist ed., Sams, 1998.