Monograph:

Wang, Q. and Zalzala, A.M.S. (1994) Real-Time Transputer Interface System for the PUMA560 Industrial Robotic Manipulator. Research Report. ACSE Research Report 530 . Department of Automatic Control and Systems Engineering

Real-time Transputer Interface System for the PUMA560 Industrial Robotic Manipulator

Q. Wang and A. M. S. Zalzala

Research Report #530

5 August 1994

RRG
Tel: +44(0)742 768555
Fax: +44(0)742 731729
Email: rrg@sheffield.ac.uk
Robotics Research Group
Dept. of Automatic Control & System Engineering
University of Sheffield

Abstract

This documentation reports on the first phase of the EPSRC project - "Advanced Real-time Control Algorithms for Robotic Systems with Application to the Resource Industries". The work has mainly been devoted to the design of a Transputer Interface Board (TIB) to establish a transputer link to the 6503 microprocessors of the PUMA arm controller. In addition to hardware implementation, program testing of the board was successfully accomplished, though improvement is still necessary and under way. The transputer system is now able to move the PUMA560 robotic manipulator joint by joint.

Table of Contents

1. Introduction 4

2. PUMA Interface Alternatives 5

3. The PUMA Transputer Interface Board (TIB) 10

3.1 Principle of operation 11

3.2 TIB details 13

4. Software Considerations 15

5. Conclusions 17

Acknowledgements 17

References 17

1. Introduction

The aim of this project is to implement new robot control structures using distributed algorithms, neural networks and genetic algorithms, and demonstrate their applicability by designing a real-time integrated robotic workcell. The project consists of mainly three phases, which are: 1). Hardware design of a transputer-based PUMA controller; 2). Implementation of parallel NNs and GAs algorithms based on this new control platform; and 3). Investigate the applicability of this new PUMA control system for real-time applications, such as in resources industries. This report details the work carried out for the last few months, marked as phase 1 for the project.

Recent developments in robotic applications have shown a trend towards precise and high speed motion to accomplish a specific task. However, the efficiency of the available industrial robots is severely lowered by the complexity of its operation. In mathematical terms, the planning and control of robot motion is a very heavy computational burden to be executed in real-time. Problems in the control of robots arises from the vast computational complexities associated with its mathematical formulations, in addition to the need for appropriate adaptive control methods to achieve the required precision and speed.

The robotics community has long been interested in applying the cognition theory as one possible solution to this demand. The use of distributed processing with transputers is a very attractive solution which has shown promising features. Basically, the computational burden is divided onto several processors and massive parallelism within the network can reduce the computational burden.

Neural networks consist of a massive number of simple neurons, which are basically simple processors doing very simple mathematical/logical operations, linked together. A neural network mimics the intelligence of advanced living creatures like a human being. Theoretically, it can learn arbitrary complicated mathematical relationships. In robotics, neural networks can be used to learn the inherently non-linear dynamics and kinematics relationships.

Genetic algorithm (GA) is essentially an optimization's method. Unlike conventional methods, it searches optimum solutions globally. By this way, it can avoid being trapped in a local minimum, which is the common handicap of the conventional methods. There are other benefits to use GA as an optimum search method. One of these is its feasibility to be parallelised on a distributed parallel processors. In robotics, it can mainly be used in optimum path planning and decision on collision avoidance.

To achieve the end-aim, the first phase of the project was to establish a physical link between the transputers and the PUMA controller. Further implementation of other advanced control algorithms, such as NNs and GAs outlined above, are dependant on this link to allow experiments.

2. PUMA Interface Alternatives

One of the basic features an intelligent system must posses is the ability to accommodate to changes in the environment. Although sensory feedback is a necessity to achieve this, most industrial robots lack this capability. The limitations lie in mainly three aspects. First, there are no channels for sensors' information to be incorporated. Second, there is no floating-point hardware to perform complex mathematical operations. And finally, the software source code is normally stored in EPROM, and can not be modified, which is necessary to include sensor information and other intelligences.

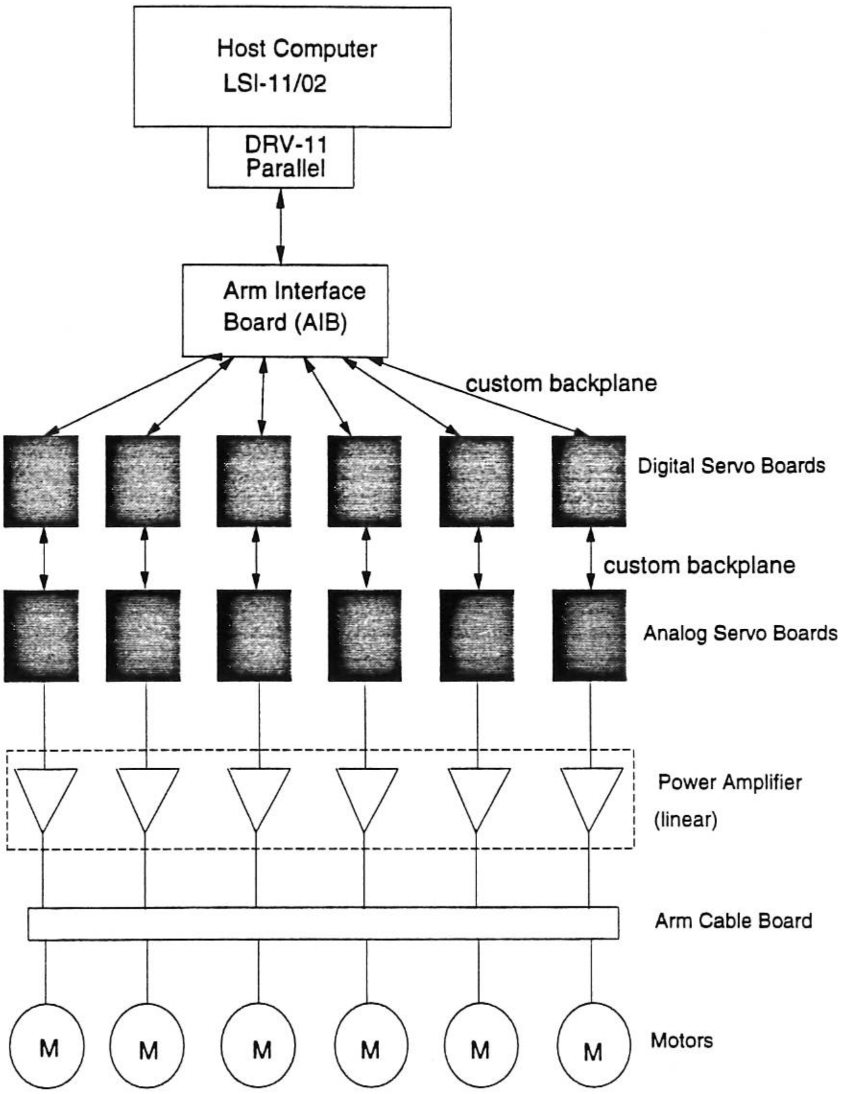

Fig.1 PUMA Controller Hierarchy

The controller of the PUMA560 robot can be enhanced by the addition of an external computer which does not presents those limitations listed above. Fig. 1 shows the PUMA controller hierarchy. Basically, there are two levels in the controller. One is the lower level which consists of a digital servo board - 6503 microprocessor, an analog servo board and a power amplifier for each joint. There is a simple PD feedback (see Fig. 3) controller in this lower level section. The feedback loop uses a sample period of 0.875 ms to ensure that all joints are servoed to where there are required. The microprocessor is also required to acknowledge back to the host computer in every 28 ms.

The higher level, which is also called the supervisory level, consists of a LSI-11 computer. This supervisory computer is mainly functioned as a management system, which is known as VAL (for Victor's Assembly Language), and ensures that new data for the lower level are sent in every 28 ms. Kinematics transformation, path planning, error handling and man-machine interfacing etc. are all processed by this computer.

Whole those boards are hold in a confined card cabinet as shown in Fig. 2. Serval alternative configurations, with or without VAL, are possible and some have already been implemented by various researchers. These alternatives differ in terms of the extent to which the existing controller hardware is replaced and the capabilities of the external computer used.

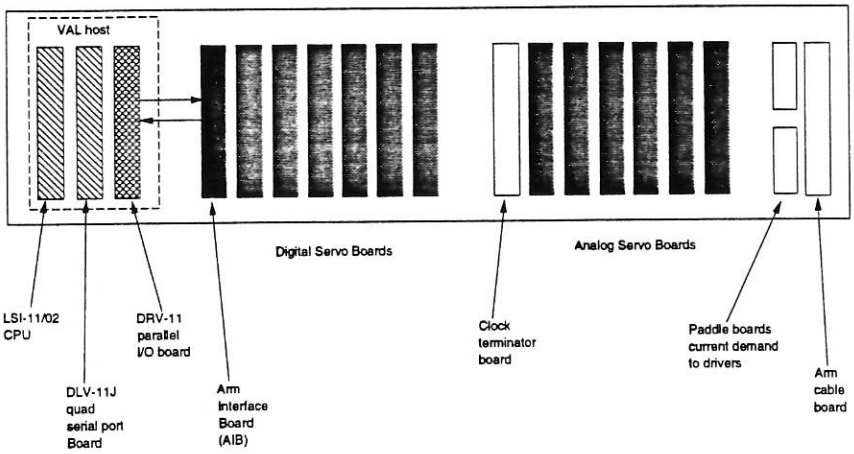

Fig.2 PUMA Control Cabinet

The approach labelled as (1) in Fig. 3, which allows for direct and total control of the joint servos, had been chosen by many researchers. Kazanzides et al. [1] used a multiple MC6800-based single board computers, the custom-developed Armstrong multiprocessor system, and two SUN 3/260 computers. Chen et al. [2] at University of Surrey used a transputer system together with a SUN workstation for the same purpose. Nagy [8] used a PC to control the PUMA robot. Though potentially very powerful and flexible, this option requires to develop a great deal of custom hardware and software.

The alternative labelled as (2) in Fig.3 was originally described by Visnes[3], and while the low-level hardware is kept to control the joint servoing, the supervisory level was eliminated. By this, a great deal of flexibilities can be acquired with much less complexity than option (1). Goldenberg and Chan [4] chose this option as well and developed a PUMA control system based on the TUNIS computer built at Toronto the University. Valvanis et al. [5] used a DEC's VAX computer using the same interface option.

Although this option is simple and uncomplicated, incomplete documentation has proved to be the main obstacle. As the low level joint controller still exists, the custom developed supervisory level must be fully compatible with it. Corke [6] had some description of how PUMA arm controller works. However, this description is based on their specific version of PUMA560, and is far from general.

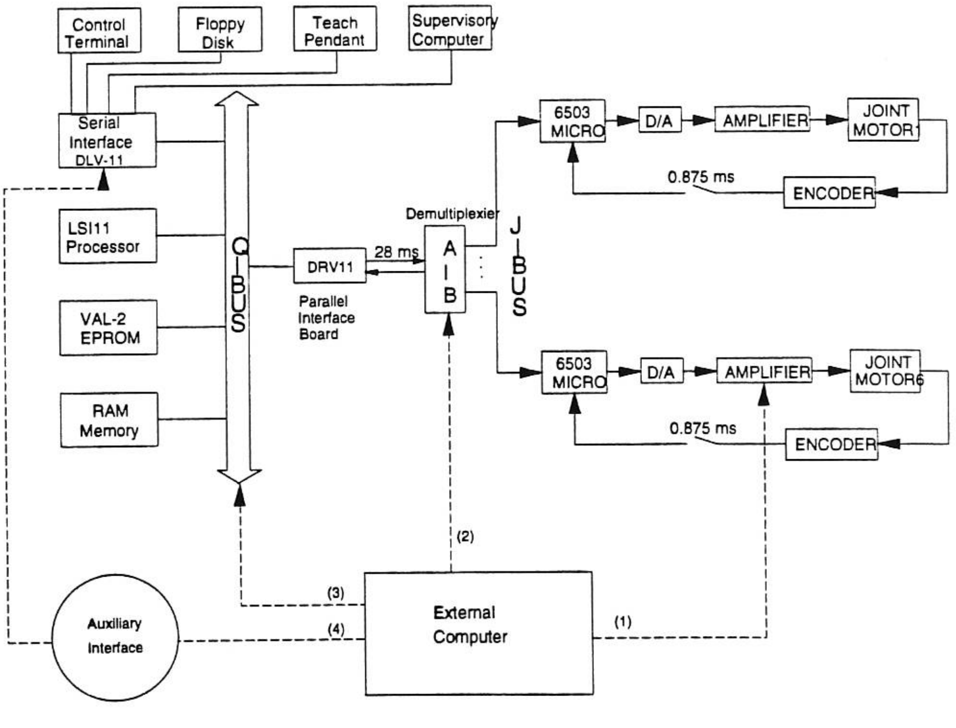

Fig.3 PUMA Interface Options

The option number (3) shown in Fig.3 directly connect the external computer to the LSI bus (Q-bus) within the existing PUMA controller. This connection is simple but DEC products have to be used. Other options exist, such as what is labelled as auxiliary interface in Fig. 3, which uses serial links. This option limits transferring speed from the external computer. And what's more, VAL is still used with its limitations.

3. The PUMA Transputer Interface Board (TIB)

The system described here is based on the option labelled (2) shown in Fig. 3, where the T805 transputer network is used to replace the LSI-11. A SUN-SPARC workstation is used to provide the man-machine interface to develop and store the control software system. The transputer system consists of a motherboard which could have up to 16 processors (TRAMs), linked together by high speed (20MBaud) serial links. One of the TRAMs is the master which provides communications with the host computer (SUN-SPARC), and also controls the slave TRAMs.

Transputer puts many processors together to increase computational power. And the processor are running in the parallel nature. Only data are exchanged using high speed serial links between processors. One of the its potentially powerful features is that it works in a way as neural networks, as each neuron in the network can be interpreted as a simple processor. New TRAMs can be added (together with other motherboards) if even more computational power is needed.

The interactions between the transputer and the out-side world (Eg other peripheral) are achieved using the Inmos link adaptor. The link adaptor is currently linked to the master TRAM at the same baud rate (20MBaud) as the internal links between TRAMs. The function of a link adaptor is to transfer serial signals to parallel ones, which are TTL compatible and can be manipulated to the user's requirements.

The T805 offers accurate control of the clock time. Even in a low priority process, the clock ticks are 15625 per second, providing a time period of 64 us. By controlling the clock ticks, it is possible to guarantee an efficient real-time response, so as to generate interrupts to the PUMA in every 28 ms. This is important as otherwise joints will jerk erratically when servoed if new data are not sent to the PUAM low-level control section in time.

The only piece of the custom-built hardware required for the new control system is the PUMA-Transputer Interface Board (TIB). Transputers differ significantly from LSI-11 in the hardware used to interface external devices. LSI-11 uses a DEC product, the DRV-11 parallel interface board, to communicate with the PUMA Arm Interface Board (AIB). In order to use the transputer system, the interface board TIB must be appeared, to the low-level control section AIB, as a DRV-11, to provide proper data and control signal transfer and buffering. Buffering allows for the transputer to be up to ten feet away from the PUMA robot without significant noise or transmission problems.

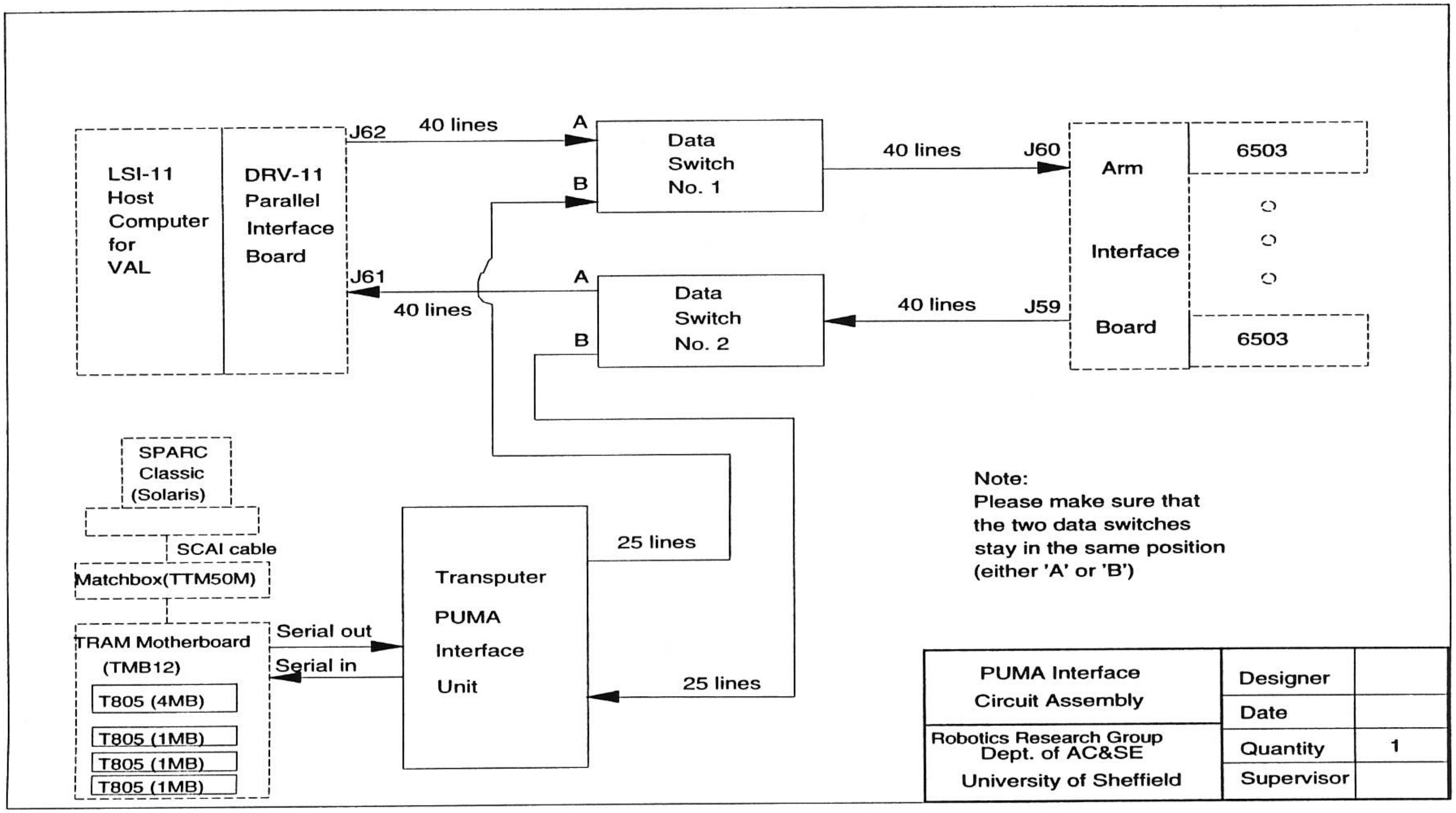

Fig. 7 General layer-out of the new controller

ensuring that the custom-built PUMA transputer interface board (TIB) really looks like a DEC DRV-11 board to the lower level control section (AIB). Reference [11] has some description of this DEC board.

The proper data and control signals must be sent to the PUMA to indicate when new commands and data become available and an acknowledgment, to indicates that a command has been received, from the PUMA must be passed back to TIB. This interface driver routines also incorporate error indicators that will inform the control programs and the user of the PUMA's failure to acknowledge any command or the user's failure to acknowledge any command sent to him.

PUMAUtil Routines within this layer are responsible for 1) Arm initialization; 2) Arm calibration; 3) Arm movement based upon joint space; 4) Kinematics (forward and inverse); and 5) Other miscellaneous utilities, Eg. transform between angles and encoder values.

The potentiometer constants were different for each individual PUMA robot, and had to be determined experimentally. Both the PUMA initialisation and calibration routines must be successfully completed before the PUMA arm can be moved safely and positioned accurately.

The movement of the PUMA arm is based on the use of optical encoders. The current means of moving the arm uses the methods of joint co-ordinates motion whereby each joint is moved at their new commanded positions at the same time. The magnitude of the encoder value determines its new position whereas the difference between the encoder's current value and the value just sent determine the speed of motion.

The new optical encoder values must be sent to the low level control hardware at every 28 ms intervals to avoid jerking the arm. If time required to calculate a set of six new encoder values is larger than 28 ms, a process has to be dedicated to sending data to AIB at the specific rate.

The forward and inverse kinematics routines are included in this layer and are still under development.

PUMACtrl The routines at this level are responsible for performing the high level task of moving the PUMA arm to whatever position and orientation the user has specified, providing they are within the PUMA's range.

5. Conclusion Remarks

A transputer based PUMA interface system had been designed, built and tested during the past few months. With this system, it is possible to use much more advanced control algorithms, like neural networks and genetic algorithms, for the operation of the PUMA560 industrial robot in real-time. Further improvement of this hardware/software system is still necessary and under way.

This design increases the PUMA's capability a great deal. The only custom-built hardware is a single interface board, TIB, while the remainder of the controller was implemented in easily accessible, portable and flexible software. Thus the new PUMA controller represents a firm foundation upon which new dynamic models, end-effector environment sensors and variety of other research topics related to the advanced control of robots may be studied experimentally.

Acknowledgement

The authors wish to thank Dr H. Thomason for his help and useful discussion in the initial construction of the design. Dr K. Yearby's and Dr H. Hu's of the RRG, Univ. of Oxford, expertise in transputers - especially the link adaptor - C011, contributed a lot in the testing. Graham Niklin and John Marsh provided various technical support, which is gratefully acknowledged. Mr Y. X. Ni and Mr. M. W. Chen helped a lot in understanding the electronic drawings of the PUMA Arm Interface Board. In a word, we would like to thank whoever helped, encouraged and offered discussion.

The financial support is provided by EPSRC (Grant No. GRJ/15797).

References

[1]. Kazanzides P., Wasti H. and Wolovich W. A., "A multiprocessor system for real-time robotic control: design and applications," in *Proc. IEEE Int. Conf. Robotics Automation*, pp. 1903-1908, 1987.

[2]. Chen N. & Parker G. A.: "An Open Architecture Robot Controller", *IEE Workshop Proceedings - Systems Engineering for Real Time Applications*, 13-14 September 1993, Cirencester, UK.

[3]. Visnes R., "Breaking away from VAL", Stanford University, 1982.

[4]. Goldenberg A. A. and Chan L., "An Approach to Real-Time Control of Robots in Task Space. Application to Control of PUMA 560 Without VAL-II", *IEEE Transactions on Industrial Electronics*, Vol. 35, No.2, pp.231-238, May 1988.

[5]. Valvanis K. P., Leachy M. B., Jr., and Saridid G. N., "On the real-time control of a VAX 11/750 computer-controlled PUMA-600 robot arm", *RPI, RAL Tech. Rep. 42,* April 1985.

[6]. Corke P. I., "Operational details of the PUMA560 robot manipulators", *CSRO tech. rep.*, 1991.

[7]. PUMA Robot (MkI) Technical Manual, Section 8.2:"Electrical Drawings", Unimation 11/82.

[8]. Nagy P. V.: "A New Approach to Operating a PUMA Manipulator without Using VAL", Carnegie Mellon University report, Canada.

[9]. Jim Q. Wang: "ART-CARS Progress Report No.4", March 14, 1994, University of Sheffield.

[10]. "Microcomputer Interface Handbook", Digital, 1980.