A Fault-Tolerant Transputer-based Parallel Controller for the Soft Arm Robot

Moenes Iskarous and Kazuhiko Kawamura
Center for Intelligent Systems
Box 1804, Station B
Vanderbilt University
Nashville, TN 37235
moenes@vuse.vanderbilt.edu

Abstract

This paper describes a transputer-based parallel controller that was developed in the Center for Intelligent Systems at Vanderbilt University to control the Soft Arm robot. The Soft Arm robot is used in a robotic aid system, called ISAC, for the physically challenged. The controller uses a network of T800 transputers as the parallel processing nodes. The controller is composed of a master node, a fuzzy node and joint nodes. The master node contains the kinematic model of the Soft Arm robot and the joint nodes contain the control algorithm for each joint. A fuzzy processor is connected to the master node to act as a supervisor for the lower level joint controllers. The controller hardware can be dynamically reconfigured through a programmable link switch, thus allow fault-tolerant operation of the controller.

1 Introduction

The ISAC (Intelligent Soft Arm Control) system is a robotic aid system that is being developed at the Center for Intelligent Systems at Vanderbilt University to aid the physically challenged [1, 2]. Figure 1 shows the robot used in the system. The arm is a five degrees of freedom robot called the Soft Arm. It has flexible and soft joints. This allows the robot to touch the user without doing harm. Thus, the Soft Arm robot is ideally suited for service robots [3].

The Soft Arm robot is pneumatically actuated by an actuators called the *Rubbertuators*. The rubbertuators are rubber pneumatic actuators that were developed by Bridgestone [4] which allows the joints to be flexible. A controller is used to drive the Soft Arm through the control of the air pressure fed to the rubbertuators.

The main objective of this research is to design and implement a robust and intelligent controller for the Soft Arm. Due to the high nonlinearities in the joints, ensuring soft motion of the arm requires the use of non conventional control techniques. The controller is based on transputers as the processing nodes. It also include a fuzzy processor that is used for high level control of the Soft Arm. Also having a fault-tolerant controller is very important when operating in a close contact with humans. One of the major goals is to design and implement a controller that can detect and recover from component faults.

The following section describes the parallel controller hardware and software and the concept of fault tolerance used in the parallel controller. Section 3 describes the modeling and control algorithm for the rubbertuator joint. Section 4 summarizes the conclusions and the possible future research directions. Finally, the appendix describes the Soft Arm robot.

2 The Transputer-based Parallel Controller

The transputer is a parallel processor, generally categorized as a Multiple Instruction Multiple Data (MIMD) computer. Transputers can be used to execute different operations on separate data at the same time. The

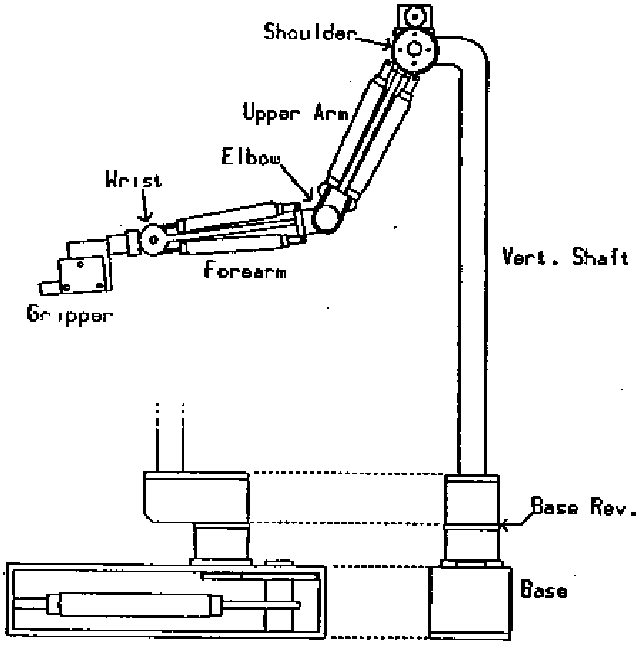

Figure 1: The Soft Arm Manipulator

transputer architecture directly implements the process model of concurrency to describe parallel systems. This logical model is the basis of occam, its native language. The transputer design philosophy was driven by C.A.R. Hoare's Communicating Sequential Processes (CSP) concept [5].

Communication between processes is achieved by means of channels. A channel between processes executing on the same transputer is a soft channel while a channel between processes executing on different processors is a hard channel. Transputers support the standard communication speeds of 5, 10 and 20 Mbits/sec. Link communication is not sensitive to clock phase. Thus, communication can be achieved between independently clocked systems as long as the communication frequency is the same.

The transputer-based parallel controller consists of two hardware modules; the transputer network and the interface circuit. The modules are connected through the high speed serial links. This allows the interface circuit to be connected to any transputer network connected to a host computer such as the SUN workstation or the PC.

2.1 Transputer Network

The transputer network developed contains the master transputer, the transputer connected to the fuzzy processor and transputer nodes for the joints control. It is connected to the SUN workstation via the serial link. To be able to connect the network to the SUN through the serial link, a transputer board is installed inside the SUN that interfaces a transputer through the *VMEbus*.

As shown in Figure 2, the transputer network on the TBE02 board consists of eight T800 transputers connected serially, each with 1 Mbyte of external RAM. There are two free links available for each transputer. These links are connected to the IMS C004 link switch. The programmable link switch can connect any two links together or connect any link to the edge connector of the board. The link switch can be configured at any time while the system is running to connect or disconnect the links. For the current application, the transputer links are connected to the interface circuits through the edge connector.

The architecture of the controller also increases the reliability and fault tolerance of the system. Having a dynamically configurable link switch is one of the advantages of using the transputer-based controller, since a faulty transputer can be isolated from the interface board and replaced by another transputer during the operation of the controller.

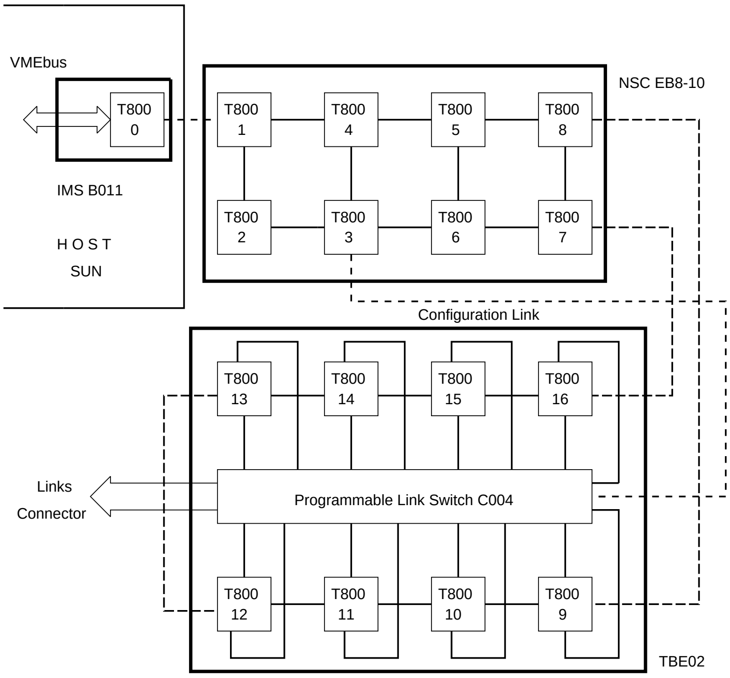

Figure 2: Transputer Network

2.2 Interface Circuit

Figure 3 shows the block diagram of the interface circuit. Each transputer is connected to one interface board that contains the interface circuit. Each interface Circuit allows one transputer to control a corresponding joint of the Soft Arm. The interface board accesses the hardware of the corresponding joint of the Soft Arm.

Each joint is driven by two current signals (4 - 20 mA) that set the pressure in the rubbertuators. An incremental encoder on the joint axis provides the interface with the joint position. The interface board contains the hardware required for converting the incremental position signal into absolute position. Extra analog input signals can be used to feedback the tension of the rubbertuators on each joint which can be used in force control.

The circuit is connected to the transputer via the high speed serial link operating at a speed of 10 Mbit/sec. As shown in Figure 3, the links are connected to two IMS C011 link adapters operating as input/output ports. Each link adapter has one eight-bit input and one eight-bit output port. One of the link adapters is used to transfer data and the other one is used to control the transfer operation.

The interface circuit can be disconnected from the joint controller without affecting the pressure and position signals. The pressure signal will be latched to the latest values of pressure because of the latches in the Digital to Analog Converter. The circuit will also continue reading the incremental encoder signals and update the joint position because the Quadrature Decoder/Counter can operate as a stand alone device until an external processor reads the absolute joint position. These features are very important in the case of isolating one of a faulty joint controllers and reconfiguring the system to replace the faulty node by another one.

2.3 Controller Software

All the controller software is written in C language. The programs are compiled and then loaded to the transputers through EXPRESS which is an environment used to download and run programs written for transputers [6]. Each transputer has its own program in its local memory. Programs can be divided into four main categories: the host node program, the fuzzy node program, the master node program and the joint nodes programs. Figure 4 shows the connections between different modules in the controller.

The master node is logically connected to all of the joint nodes. This is done through EXPRESS since the

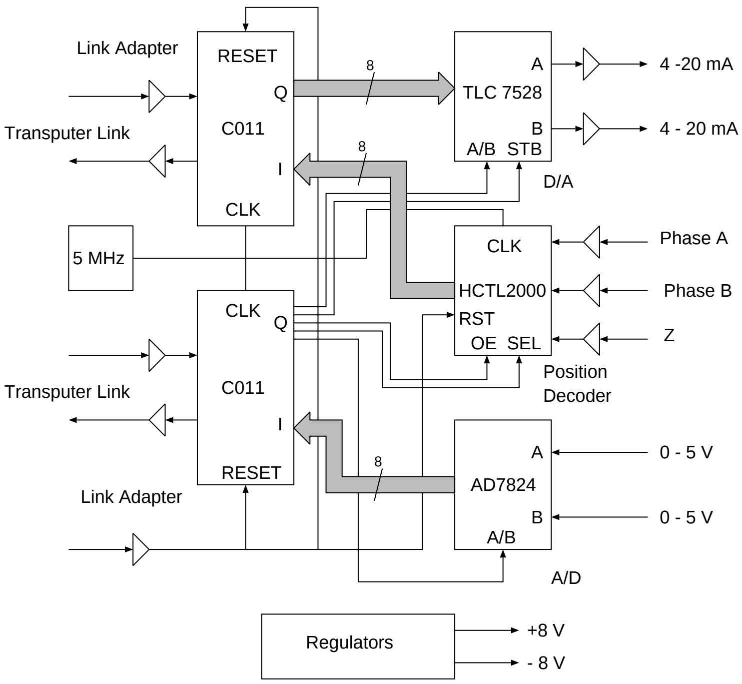

Figure 3: Interface Circuit Block Diagram

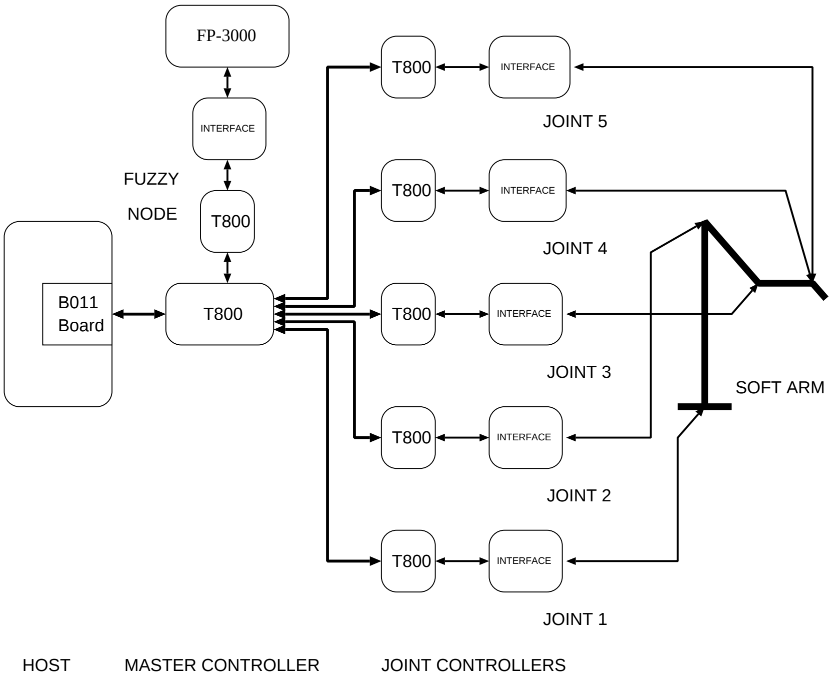

Figure 4: Connections Between Different Modules

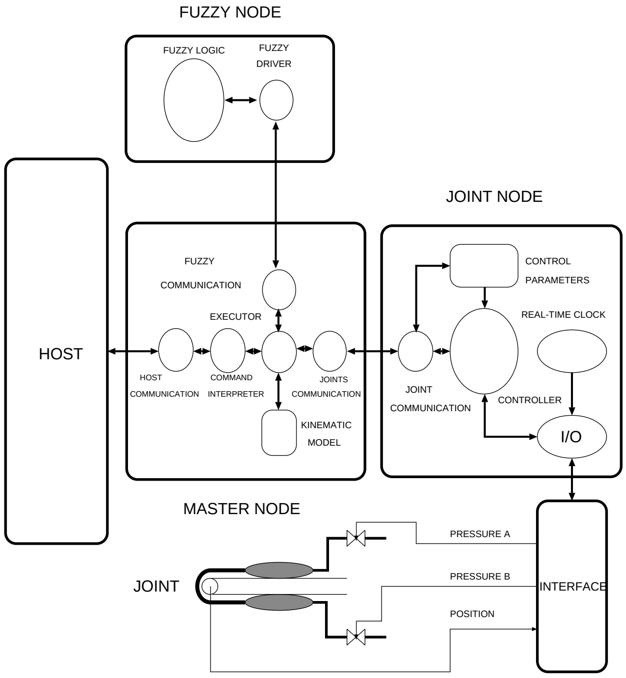

Figure 5: Software modules of the Controller

hardware links are not enough for all of the connections with the joint nodes. The following is a description of each node:

2.3.1 Host Node

The host node is the host computer used to send the control commands to the parallel controller. In this case, the host computer is a SUN workstation. This node is connected to the rest of the ISAC system via the Ethernet. The host computer can be replaced by any computer or terminal with a transputer serial link.

The host node initializes the transputer network and download programs for the master and joint controller nodes. It also downloads the knowledge base used by the fuzzy processor.

2.3.2 Master Node

This node is the one that communicates directly with the host computer. It accepts commands from the host computer to be performed by the Soft Arm robot. As shown in Figure 5, the master node contains the host communication module, the command interpreter, the Soft Arm kinematic model, the fuzzy communication module and the joints communication.

The master node is connected to the fuzzy processor that supervises the lower level PID controllers of the joint nodes. The fuzzy controller can dynamically changes the PID parameters so as to achieve better performance of joint control.

2.3.3 Fuzzy Node

The fuzzy node is a transputer connected with the OMRON FP-3000 fuzzy processor. The processor has its own memory where the fuzzy rules and membership functions are stored. The internal memory size is 1 Kbyte. Rules are stored internally in 208 bytes and can use external memory expansion up to 4.3 Kbytes. Each rule can have up to 8 antecedents and 2 consequents. The fuzzy processor can accept 8 inputs and 4 outputs. Each input may have up to 7 different membership functions that can be defined piecewise using 4 points. Also, each output can have up to 7 membership functions that are defined by a single value. This type of membership functions is called singleton [7]. The singelton is a bar-shaped membership function that have no width [7].

The processor can store 3 different rule groups in memory. The active rule group can be changed by sending a control word to the processor. The number of rules that the fuzzy processor can store depends on the number of antecedents and consequents in the rule group. Rule capacity can be increased by expanding the memory. In the case of our parallel controller, the rules were designed to fit in the memory inside the fuzzy processor. Each rule is associated with a weight that determines the effective contribution of that rule.

Defuzzification is done using one of three methods: the center of gravity, maximum height from the right and maximum height from the left. The method used in the parallel controller is the center of gravity.

The fuzzy node is used to initialize the fuzzy processor and control it to perform fuzzy inferencing. It receives inputs from the master node and run the fuzzy processor to generate the output which is returned back to the master node. This allows the fuzzy processor to run in parallel with the master and the joint nodes.

The fuzzy processor will be used for high level intelligent control of the joints. It will act as a supervisor to the lower level PID controllers. The main job of the fuzzy processor is to tune the PID parameters dynamically as the robot is moving to achieve a smooth stable motion. It will also be used in learning control through the modification of the rules and membership functions. The knowledge base of the fuzzy processor will be modified using a training mechanism (could be a neural network) so that it learns human-like trajectories.

2.3.4 Joint Nodes

Each joint node has its own program in its local memory. The programs are identical for all the joint nodes. The controller parameters of each joint are different from the rest since each joint has its own dynamic characteristics.

As shown in Figure 5, each joint node contains the master communication, the PID controller and the interface circuit driver. The PID parameters of the joint controller can be dynamically changed using the fuzzy supervisor to achieve a smooth motion of the arm. This operation is done through the master node.

2.4 Fault-Tolerant Operation

Using the dynamically configurable link switch, a faulty transputer can be isolated from the interface board and replaced by another transputer during the operation. This is similar to N-modular redundancy with spares [8]. Another enhancement in the operation is to use voting among three master nodes. This is used to achieve fault tolerance in the robot controller in case of failure of one of the master nodes. Figure 6 illustrates the proposed fault-tolerant architecture.

3 Modeling and Control Algorithm

3.1 Identification of the Joint Model

One of the problems encountered during the design of the controller is to develop an appropriate model that can be used to represent the joints. The identification of the joint model was performed using a rubbertuator test-bed.

The transfer function of the rubbertuator joint can be defined as the transfer function relating the pressure difference in the joint rubbertuators and the angle of rotation of the joint shaft:

$$G(s) = \frac{\Theta(s)}{\Delta P(s)} \tag{1}$$

Two linear discrete second-order models were tested for the system. The first model contains a delay, one zero and two poles. It is in the form

$$\frac{\Theta(z)}{\Delta P(z)} = z^{-1}\frac{b_1 + b_2 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{2}$$

The second model contains a delay, two zeros and two poles. It is in the form

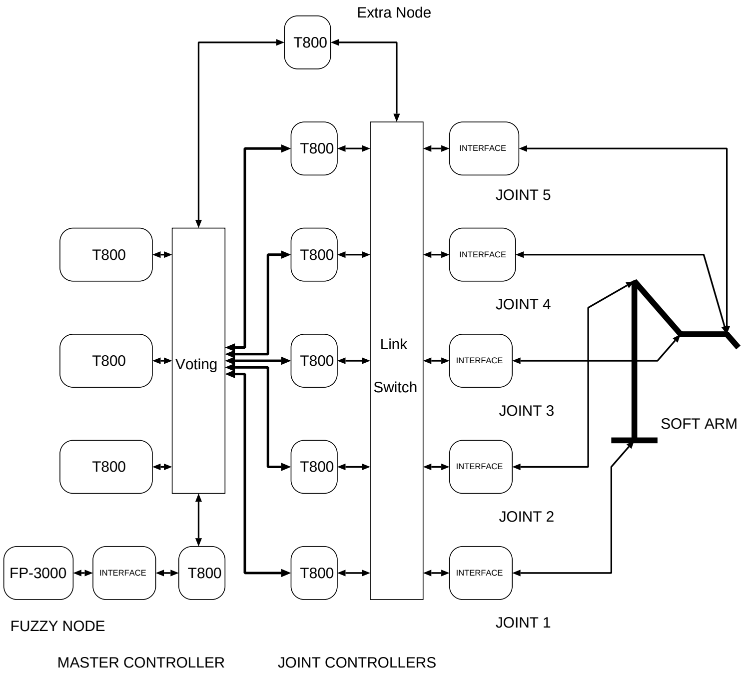

Figure 6: Fault-Tolerant Controller

$$\frac{\Theta(z)}{\Delta P(z)} = z^{-1}\frac{b_1 + b_2 z^{-1} + b_3 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{3}$$

The input signal (Δp) is the digital value of the pressure difference. The output signal is the digital position of the shaft.

Different identification techniques were used to identify the system parameters. The techniques used include the least squares estimation of an ARMA (Autoregressive Moving Average) model, the instrumental variable [9], the least squares estimation for an ARMAX model and the Box-Jenkins techniques. The first technique, which uses an ARMA model, was the most successful in reducing the error between the model simulation and the actual response. Also the first model had lower error between the model simulation and the actual response [10]. So the first model was chosen to represent the joint.

The other identification techniques used resulted in higher values of error between the actual response and the model simulation. So only the results of this technique are presented.

Figure 7 shows the input and output data. The identification was done using 1000 data values. The input was a pseudo random binary sequence of Δp in pressure increments and the output θ was in step values.

Figure 8 shows the frequency response of the actual system and of the model. The model is plotted by the continuous line and the actual system by the dotted line.

The final form of the discrete transfer function of the joint is

$$\frac{\Theta(z)}{\Delta P(z)} = z^{-1}\frac{0.6263 + 0.3936 z^{-1}}{1 - 0.0923 z^{-1} - 0.1046 z^{-2}} \tag{4}$$

3.2 Control Algorithm

A PID controller is used to control each joint of the Soft Arm robot. The discrete time PID controller is

$$\frac{P(z)}{E(z)} = K_p + \frac{K_i T(z+1)}{2(z-1)} + \frac{K_d(z-1)}{Tz} \tag{5}$$

where

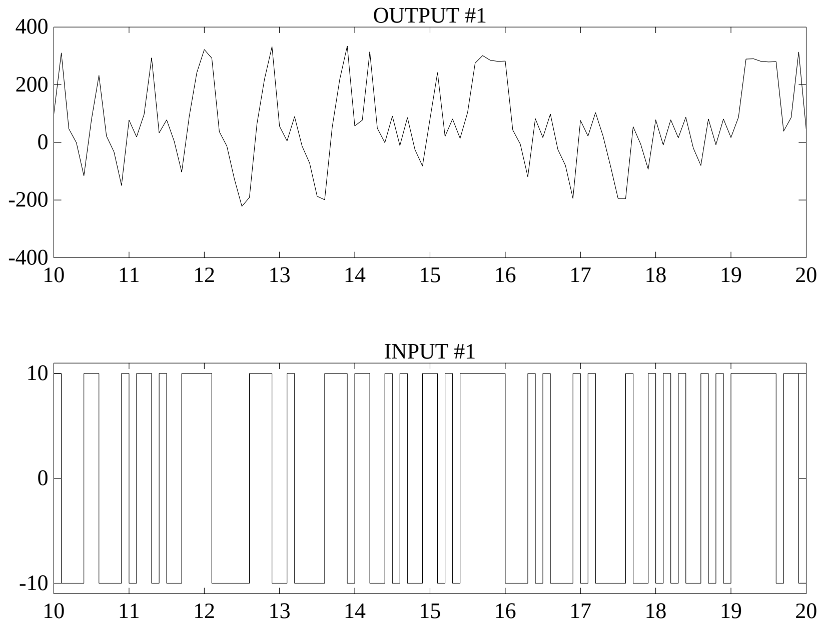

Figure 7: Input/Output Data

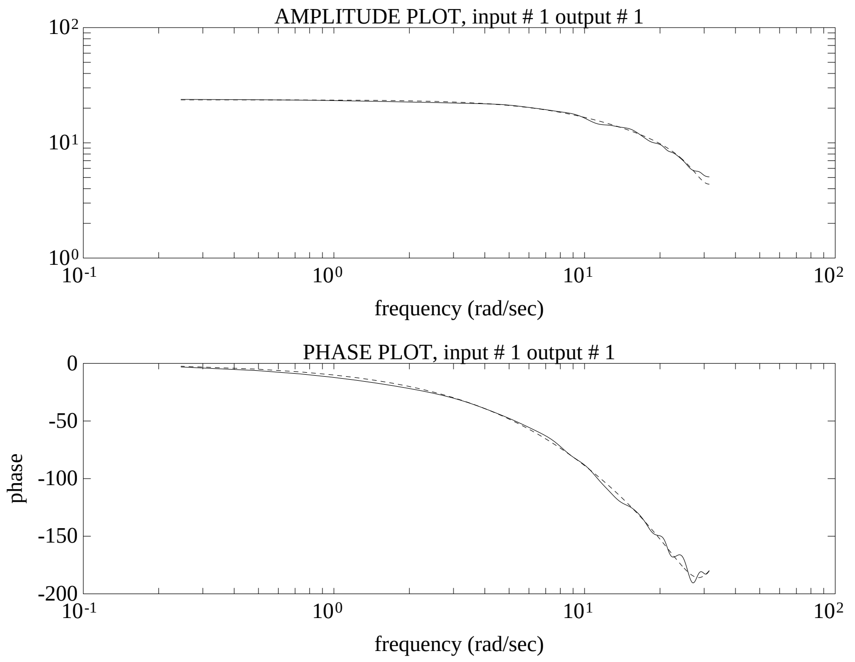

Figure 8: Frequency Response of the Model

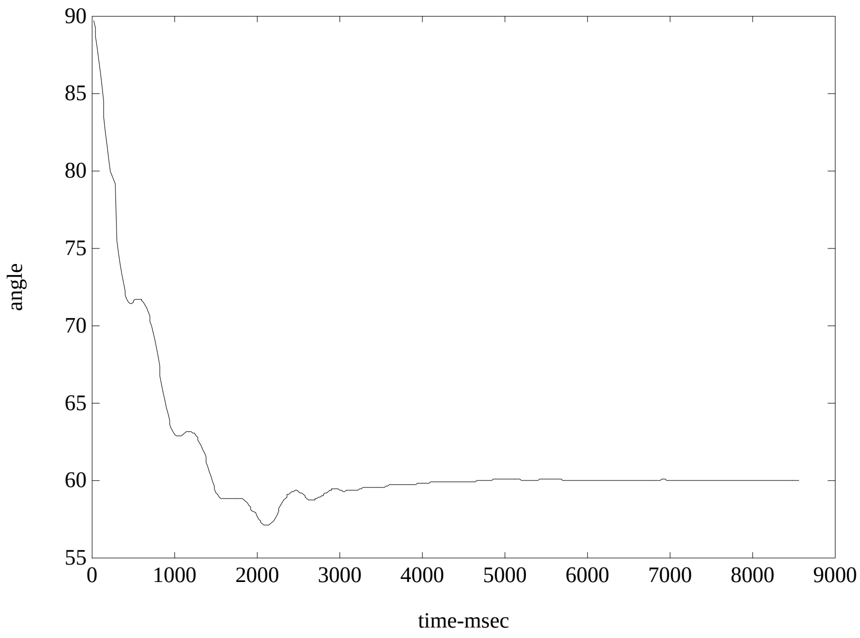

Figure 9: Step Response of the Shoulder Joint with speed = 100

T: Sampling time of the system

For each joint of the Soft Arm robot, different PID parameters are selected. This is due to the difference in the inertia, the friction and the rubbertuators that are associated with the joint. Tuning the parameters was done experimentally.

The control algorithm for Soft Arm robot was tested by plotting the response of the joints for a step command. Figure 9 shows the step response of the shoulder joint (joint 2) when a command is issued to move that joint from angle 90° to 60°.

As shown in Figure 9, the oscillations during the rise time are due to the nonlinearities in the rubbertuators and the joint dynamics. The joint also has a variable load due to the change in the inertia when the Soft Arm robot extends or retracts.

A ramp function is used to feed the position command to the PID controller. The ramp slope is determined by the speed parameter that can be adjusted for each joint. The transient response of joint is improved by using a speed of 20 where the maximum speed is 100. Figure 10 shows that using the ramp function improved the transient response of the joint without affecting the rise time.

Currently, a tuning algorithm using fuzzy logic to achieve a better transient response for each joint is under development. The algorithm will be dynamically changing the speed parameter according to the fuzzy rules so that the rise time is minimum with a smoother transient response.

4 Conclusions and Future Work

The main goal of this research was to design and implement a robust controller for the Soft Arm robot. This controller has been developed based on transputers as the parallel processing nodes.

One of the main criteria of evaluating the performance of the new controller was the speed of response to the commands. This aspect is important in the ISAC system since reflex control has been incorporated [3]. For example, reflex control requires a fast response to a *stop* command. This was measured with the parallel controller and it was found to be 800 μ sec compared to 100 msec of the SOFT ARM controller [3].

Research on combining neural networks with fuzzy logic in robot control is now conducted. The fuzzy system will be trained by modifying the rules to achieve high level tasks such as obstacle avoidance or path

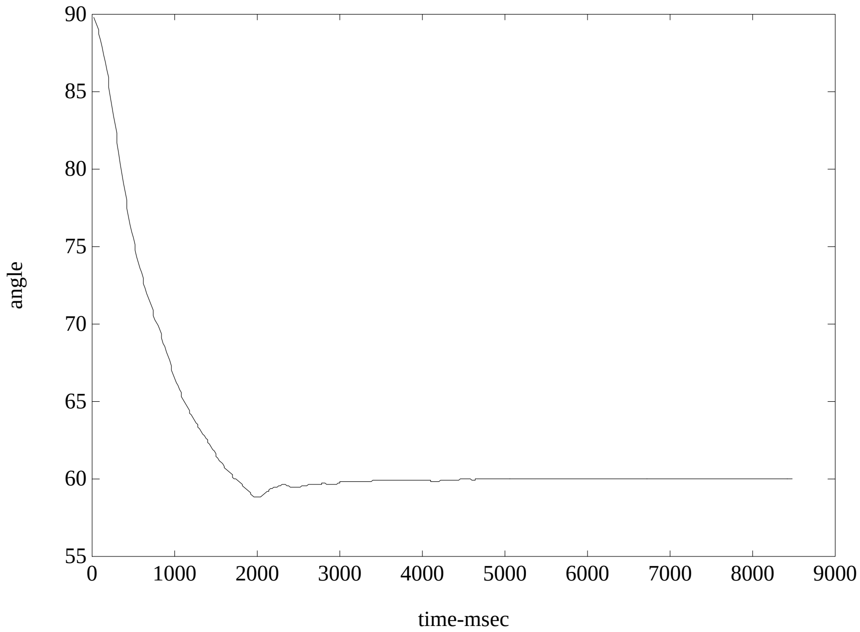

Figure 10: Step Response of the Shoulder Joint with speed = 20

generation [11] or human-like motion using neural networks [12].

Research on fault tolerance will be also conducted on the parallel controller. This research is possible by utilizing the programmable link switch in the transputer-based parallel controller. This may require more tight connection among different nodes which will lead to rewriting a dedicated message passing layer that can support the fault-tolerant operation of the controller.

Acknowledgement

The authors would like to thank all members of the ISAC group for their support and help during the design and implementation of the parallel controller. Special thanks to Pablo Martinez and Todd Pack for their contribution in the parallel controller software.

Appendix: The Soft Arm Robot System

The Soft Arm robot is a five degrees of freedom, pneumatically actuated robot developed by the Bridgestone Corporation of Japan [10]. Rubbertuators manipulate the robotic arm in much the same way as muscles manipulate the human arm. By varying the internal air pressure levels, the rubbertuators can be made to expand or contract in length. The circumference of the rubbertuators decreases during expansion and increases during contraction, adding to the robot's similar physical appearance to that of the human arm.

The structure of a rubbertuator is shown in Figure 11. It consists of a rubber tube covered with a sleeve of inter-twined fiber cord. Metal fittings are affixed to both ends of the tube.

Rubbertuators generate force only when they contract in the direction of contraction. The following formula is used to describe basic characteristics of the rubbertuator:

$$\frac{F}{\frac{\pi}{4}D_0^2} = P(a(1-\epsilon)^2 - b) \tag{6}$$

where

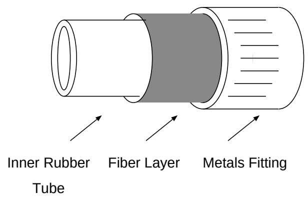

Figure 11: Rubbertuator Structure

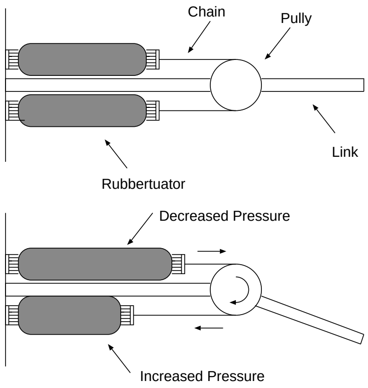

Figure 12: Example of One-axis Drive Rotation

F: Contraction force
D_0: Effective diameter before displacement
P: Internal pressure (gauge pressure)
ϵ: Contraction rate
a, b: Constants of the actuator

Rubbertuators feature hysteresis in their operation which is due to the original cord hysteresis inherent in the rubber, friction between the rubber and cords, and friction between the cords themselves [13]. This is a highly nonlinear system with complex behavior. Having this high degree of nonlinearity, a non-classical control technique would be more appropriate for the joint control. These techniques could be adaptive, neural or fuzzy control.

Two rubbertuators are used in the robot to form one degree of freedom. An example of one degree of freedom is illustrated in Figure 12. If the initial pressure in both rubbertuators is P_0. Then, if $+\Delta P$ is the pressure applied to one of the rubbertuators, and $-\Delta P$ is applied to the other, then rotation can be obtained in proportion to ΔP in the direction of $+\Delta P$ [10].

References

[1] A. Kara, K. Kawamura, S. Bagchi, and M. El-Gamal, "Reflex control of a robotic aid system for the physically disabled," *IEEE Control Systems Magazine*, vol. 12, pp. 71–77, June 1992.

[2] S. Bagchi and K. Kawamura, "An architecture for a distributed object-oriented robotic system," in *1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 711–716, July 1992.

[3] M. El-Gamal, A. Kara, K. Kawamura, and M. Fashoro, "Reflex control for an intelligent robotic system," in *1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1347–1354, July 1992.

[4] Bridgestone Corporation, *Rubbertuators and Applications for Robots.* Technical Report.

[5] C. Hoare, *Communicating Sequential Processes.* Prentice Hall, 1985.

[6] ParaSoft Corporation, *Express C Users's Guide*, 1990.

[7] Omron Corporation, *FP-3000 Digital Fuzzy Processor User's Manual*, 1991.

[8] B. W. johnson, *Design and Analysis of Fault-Tolerant Digital Systems.* Adison-Wesley Publishing Company, 1989.

[9] N. K. Sinha and B. Kuszta, *Modeling and Identification of Dynamic Systems.* Van Nostrand Reinhold Company, 1983.

[10] M. Iskarous, "A transputer-based parallel controller for the soft arm robot," Master's thesis, Vanderbilt University, May 1992.

[11] B. Kosko, *Neural Networks and Fuzzy Systems.* Prentice Hall, 1992.

[12] M. Kawato, Y. Maeda, Y. Uno, and R. Suzuki, "Hierarchical neural network model for voluntary movement with application to robotica," *IEEE Control Systems Magazine*, vol. 8, pp. 8–16, Apr. 1998.

[13] Bridgestone Corporation, *Rubbertuators and Applications for Robotics, Technical Guide No. 1*, 1986.