TRACS: An Experimental Multiagent Robotic System

MS-CIS-90-60
GRASP LAB 230

Xiaoping Yun
Eric Paljug
Ruzena Bajcsy

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104

August 1990

TRACS: An Experimental Multiagent Robotic System[*]

Xiaoping Yun, Eric Paljug, and Ruzena Bajcsy
Department of Computer and Information Science
University of Pennsylvania
200 South 33rd Street
Philadelphia, PA 19104-6389

Abstract

TRACS (Two Robotic Arm Coordination System), developed at the GRASP Laboratory at University of Pennsylvania, is an experimental system for studying dynamically coordinated control of multiple robotic manipulators. The systems is used to investigate such issues as the design of controller architectures, development of real-time control and coordination programming environments, integration of sensory devices, and implementation of dynamic coordination algorithms. The system consists two PUMA 250 robot arms and custom-made end effectors for manipulation and grasping. The controller is based an IBM PC/AT for its simplicity in I/O interface, ease of real-time programming, and availability of low-cost supporting devices. The Intel 286 in the PC is aided by a high speed AMD 29000 based floating point processor board. They are pipelined in such a way that the AMD 29000 processor performs real-time computations and the Intel 286 carries out I/O operations. The system is capable of implementing dynamic coordinated control of the two manipulators at 200 Hz.

TRACS utilizes a C library called MO to provide the real-time programming environment. An effort has been made to separate hardware-dependent code from hardware-independent code. As such, MO is used in the laboratory to control different robots on different operating systems (MS-DOS and Unix) with minimal changes in hardware-dependent code such as reading encoders and setting joint torques.

TRACS utilizes all off-the-shelf hardware components. Further, the adoption of MS-DOS instead of Unix or Unix-based real-time operating systems makes the real-time programming simple and minimizes the interrupt latencies. The feasibility of the system is demonstrated by a series of experiments of grasping and manipulating common objects by two manipulators.

[*]Presented at the Workshop on Software Tools for Distributed Intelligent Control Systems, San Francisco, CA, July 17-19, 1990

1 Introduction

An intelligent robotic system consists of multiple agents such as manipulators, end effectors, sensory devices, controllers, environments, etc. The operation of such a system requires real-time coordination of multiple agents. Coordination involves many research issues and can be interpreted differently from different perspectives. A recent NSF sponsored workshop on coordination and coordination theory [8, 5] suggested five definitions of coordination. This lack of a unique definition of coordination signals a need for understanding and investigation of coordination problems. With respect to a multiagent robotic system, those problems include real-time dynamic coordinated control of multiple manipulators and real-time communication among agents of the multiagent robotic system. Underlying those two problems is the development of a real-time programming environment and hardware architecture for design and implementation of coordinated control algorithms.

This paper describes an experimental multiagent robotic system, called TRACS, developed at University of Pennsylvania. TRACS is aimed to primarily address real-time coordinated control of multiagent systems. Two PUMA 250 manipulators, together with end effectors and sensors, are utilized in a testbed system. An IBM PC/AT is chosen as the host computer for the development of real-time programming environment.

Other real-time robotic systems include Chimera II [10], Condor [7], SPARTA [3], SAGE [9]. TRACS has the advantage of being cost-effective, simple to use, free from staff support, and easy to transfer to other labs and field applications.

2 System Description

As discussed in the introduction section, we are interested in two problems related to coordination of a multiagent system: control and communication. Realizing the difficulty of the problems, two independent yet integrated approaches are taken towards understanding of coordination. In the first approach, the effort is focused on coordinated control algorithms while the effect of communication delays on coordination is intentionally minimized. This is made possible by considering fewer agents so that one processor provides adequate computational needs. Interprocessor communications are thus eliminated. TRACS mentioned early is a result of this approach and it will be addressed in detail in this section. In the second approach, the emphasis is placed on real-time communication of a distributed system. Towards this end, a real-time kernel for distributed system, called TIMIX, is developed in the GRASP laboratory [4].

2.1 Hardware

The principal agents of TRACS are two PUMA 250 robot manipulators. The goal of the system is to dynamically coordinate motions of the two manipulators to perform cooperative tasks in their common workspace. Due to their primitive computing resources, the original Unimate controllers of PUMA robots are not capable of implementing dynamic coordinated control algorithms. A new controller is built. However, to eliminate unnecessary hardware constructions, the power amplifiers, D/A converters, and encoder decoding/counter circuits of the Unimate controllers are retained.

The following considerations are taken into account before choosing a controller architecture. Firstly, distributed systems should be avoided to eliminate communication delays as the emphasis of this study is on coordinated control. Single processor and coprocessor architectures are preferred. Of course, the choice is constrained by the current technology of microprocessors (speed) as well as available budget. Secondly, a simple operating system is preferred which provides such basic functions as file server, editing, compiling, and debugging and which can provide, after minimal effort, a real-time programming environment. With this regard, Unix should be on the low end of preference list. Thirdly, interface to manipulators and sensors should be simple and standard.

After a thorough evaluation, an IBM PC/AT with the MS-DOS operating system is chosen as the host computer. This choice satisfies the above requirements, namely the PC-AT, even with a coprocessor, is not a distributed system, MS-DOS is a simple operating simple that provides basic functionality and a real-time environment, and off-the-shelf technology is readily available for sensor interfaces.

The real-time computational burden of the coordinated dynamic control algorithms requires the addition of a coprocessor to the host computer. An AMD 29000 high speed floating point coprocessor board by YARC, INC. was chosen to satisfy this requirement. It is a single processor board with shared memory and I/O space on the PC-AT system. It comes with a C compiler.

The hardware architecture of this system is shown in Figure 1. The PC-AT communicates with the Unimate controllers through a parallel interface. The exchanged information includes angular encoder readings from the joints to the host computer and manipulator motor voltage data from the host computer to the Unimate controller. The end-effector control information is communicated through an analog interface. The host computer can also communicate with other machines, if needed, through an ethernet interface.

The end-effectors are custom made for the experiments that are conducted. Currently, two end-effectors have been made, a multi-configurable gripper and an open palm. The open palm is now being used for experiments in coordinated control where the two arms must manipulate large objects by maintaining a specified internal force between the palms.

2.2 Software

The development of new control algorithms is the purpose of this system. The system's software environment must allow the programmer to easily incorporate existing software, such as kinematics and gravity compensation. It must also provide easy access to information from the agents and sensors, and to easily communicate the algorithm's results. It should be based on a portable language. The system's underlying operation should be transparent to the normal user. However, if a programmer must make changes to the system, because, for instance, by the addition of a new sensor board, then the underlying system software should be simple enough to allow this without a major effort.

These requirements are satisfied by using the MO' control structure which is written in the C language [2]. MO' provides simple routines to get information in and out. It provides access to software programs that have already been developed. What Mo' does not specify is the underlying real-time system software.

The PC-AT host computer is configured as a real-time system by use of its clock interrupt. Each interrupt will cause the PC-AT to execute the code that gathers new data from the manipulators and sensors into shared memory, signals the AMD 29000 coprocessor to begin calculations with this new data, and output the latest control result from the AMD 29000 to the manipulator actuators. The time between interrupts is used by the PC-AT to perform user interface tasks. The AMD 29000 executes its code, the control algorithm in the MO' control structure, at the servo rate. Figure 2 shows the pipeline timing table of the Intel 80286 and AMD 29000. Thus the underlying system is not formidable to the programmer that must reconfigure it.

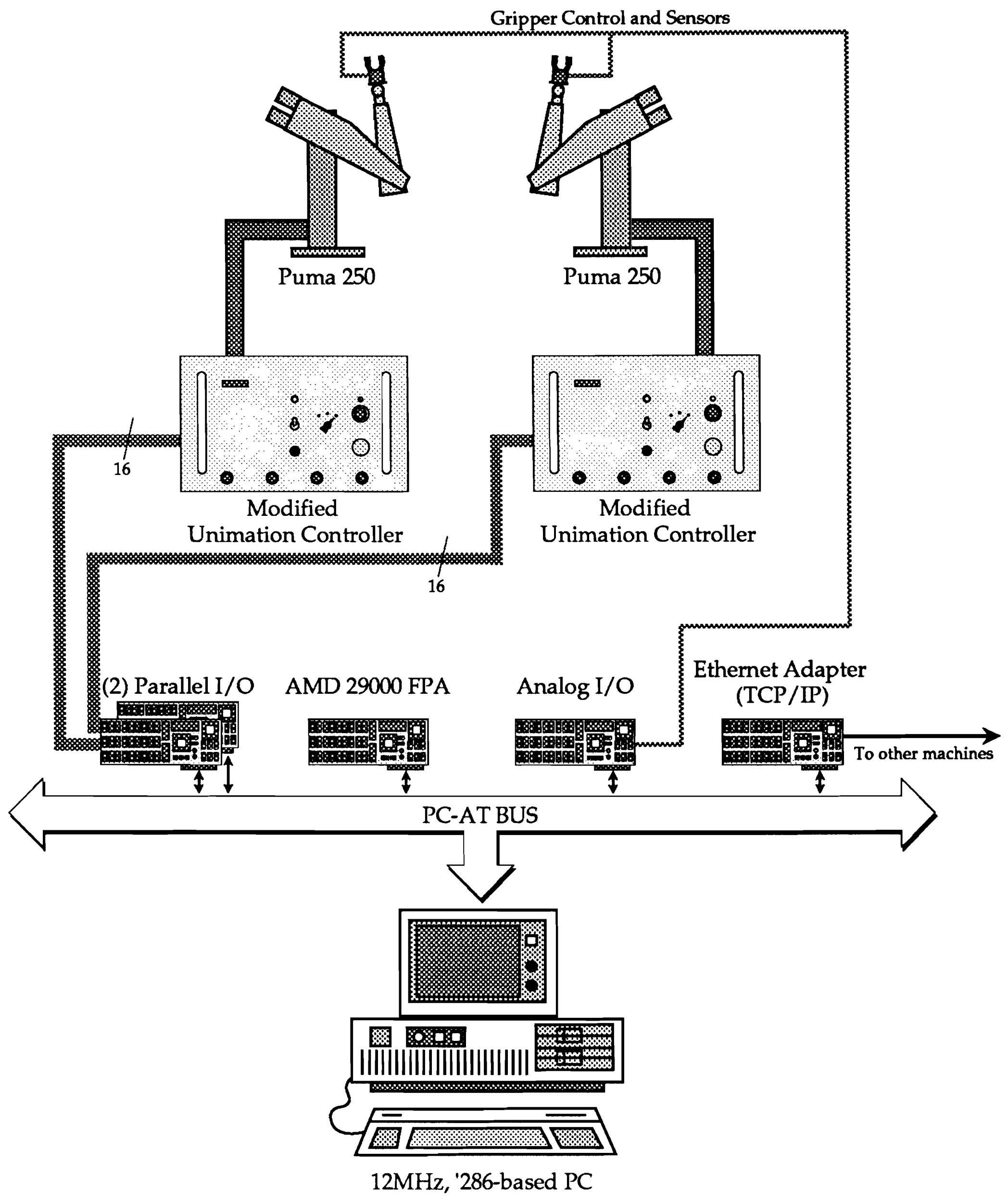
Gripper Control and Sensors
Puma 250
Puma 250
16
Modified Unimation Controller
Modified Unimation Controller
16
(2) Parallel I/O
AMD 29000 FPA
Analog I/O
Ethernet Adapter (TCP/IP)
To other machines
PC-AT BUS
12MHz, '286-based PC

Figure 1: TRACS Hardware Architecture

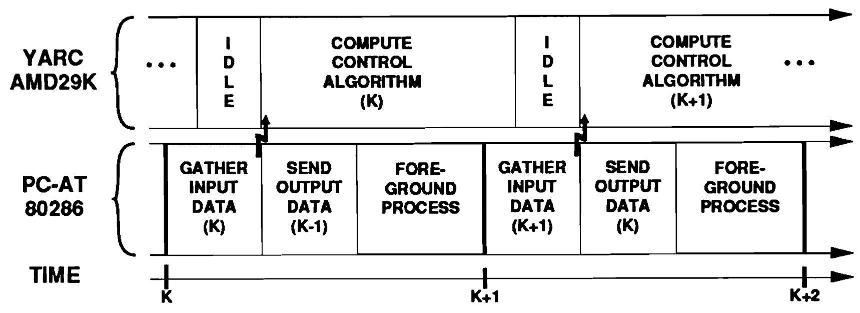

Figure 2: Timing Table of the Intel 80286 and AMD 29000

3 Implementation of Two-Arm Coordination

A series of experiments are conducted to test the functioning of the TRACS system. The tasks of the experiments are grasping, lifting and transporting objects by using two manipulators. The performance of those tasks requires dynamic coordination of the two manipulators, which in turn requires the understanding of two handed grasping and coordinated control of two manipulators.

Two handed grasping is concerned with the process of approaching the object, detecting contacts, evaluating the grasping configuration, determining the grasping force, and applying the grasping force to the object. An important step is the evaluation of the initial grasping configuration. Due to the dependence on friction, certain hands/fingers configurations can not guarantee stable grasping. In experiments, the end effectors (or hands) are simply flat surface palms instrumented by force sensors. Using such palms greatly reduces the uncertainty of contact normals. Regardless of the local geometry of the object at the contact point, the contact normal is that of the palm which is known to the controller. Given the coefficient of friction, an algorithm has been established to evaluate grasping configuration based on the relative position and orientation of the two palms. In the 2-dimensional case, the algorithm is based on the relative offsets of the two palms in x and y directions, and the relative angle between the two palms. It is noted that this algorithm does not dependent on the shape of the object and the exact contact points on the palms.

Having stably grasped the object, the next issue is coordinated control of the two manipulators. This problem has been theoretically studied by various groups including [13, 6, 1, 11]. Nevertheless, there is little experimental work in this area. Using TRACS, it is straightforward to implement and test any control algorithms. Coordination of two manipulators requires the simultaneous control of the Cartesian position and the interaction force [12]. The challenging problem here is the development of force control methods and the associated stability analysis. In the TRACS experiments, the problem is further complicated by the unilateral constraints, namely, each palm can push but can not pull the object. A solution that utilizes a nonlinear decoupling coordination algorithm which allows independent control of force and position is being studied. The experiments show that TRACS is capable of implementing dynamic coordinated control algorithms.

4 Conclusion

This paper described an experimental real-time control system, TRACS. Though the system was developed to control a multiagent robotic system, it is applicable to other real-time control systems. In particular, it is attractive to control of mobile platforms such as vehicles due to the portability of the system.

TRACS uses all off-the-shelf hardware components to reduce the cost and to expedite the technology transfer. The adoption of the simple MS-DOS operating system simplifies real-time programming, minimizes interrupt latencies, and reduces response overheads. Using C library to support the real-time programming environment and to implement real-time control algorithms makes the system portable and easy for distribution.

5 Acknowledgement

The authors wish to thank Filip Fuma and John Bradley for technical design support. This work was in part supported by: Airforce grant AFOSR F49620-85-K-0018, Army/DAAG-29-84-K-0061, NSF-CER/DCR82-19196 Ao2, NASA NAG5-1045, ONR SB-35923-0, NIH grant NS-10939 -11 as part of Cerebro Vascular Research Center, NIH 1-RO1-NS-23636-01, NSF INT85-14199, NSF DMC85-17315, ARPA N0014-88-K-0632, NATO grant No. 0224/85, DEC Corp., IBM Corp., and LORD Corp.

References

[1] Samad A. Hayati. Position and force control of coordinated multiple arms. *IEEE Transactions on Aerospace and Electronic Systems*, 24(5):584–590, September 1988.

[2] Gaylord Holder. Mo robot control software. December 1989.

[3] J. Ish-Shalom and P. Kazanzides. SPARTA: multiple signal processors for high-performance robot control. *IEEE Transactions on Robotics and Automation*, 5(5):628–640, October 1989.

[4] Robert B. King. *Design, Implementation, and Evaluation of a Distributed Real-Time Kernel for Distributed Robotics.* Technical Report MS-CIS-90-40, GRASP LAB 220, Dept. of Computer and Information Science, University of Pennsylvania, 1990.

[5] Joshua Lederberg and Keith Uncapher. *Towards a National Collaboratory.* Technical Report, National Science Foundation, The Rockefeller University, March 17-18, 1989. Report of an Invitational Workshop.

[6] Y. Nakamura, K. Nagai, and T. Yoshikawa. Mechanics of coordinative manipulation by multiple robotic mechanisms. In *Proceedings of 1987 International Conference on Robotics and Automation*, pages 991–998, Raleigh, North Carolina, 1987.

[7] S. Narasimhan, D. M. Siegel, and J. M. Hollerbach. CONDOR: an architecture for controlling the utah-MIT dexterous hand. *IEEE Transactions on Robotics and Automation*, 5(5):616–627, October 1989.

[8] NSF-IRIS Review Panel. *A Report by the NSF-IRIS Review Panel for Research on Coordination Theory and Technology, June 26, 1989.* Technical Report, National Science Foundation, 1989.

[9] Lou Salkind. The SAGE operating system. In *Proceedings of 1989 International Conference on Robotics and Automation*, pages 860–865, Scottsdale, Arizona, May 1989.

[10] David Stewart, D. Schmitz, and P. Khosla. Implementing real-time robotic systems using CHIMERA II. In *Proceedings of 1990 International Conference on Robotics and Automation*, pages 598–603, Cincinnati, Ohio, May 1990.

[11] T. J. Tarn, A. K. Bejczy, and X. Yun. New nonlinear control algorithms for multiple robot arms. *IEEE Transactions on Aerospace and Electronic Systems*, 24(5):571–583, September 1988.

[12] Xiaoping Yun, T.J. Tarn, and A.K. Bejczy. Dynamic coordinated control of two robot manipulators. In *The 28th IEEE Conference on Decision and Control*, Tampa, Florida, December 1989.

[13] Y. F. Zheng and J. Y. S. Luh. Control of two coordinated robots in motion. In *Proceedings of 24th IEEE Conference on Decision and Control*, pages 1761–1765, Ft. Lauderdale, Florida, December 1985.