A Distributed Operating Environment for Integrated Robotic Systems

LUKE NG, JAN P. HUISSOON
Department of Mechanical Engineering, University of Waterloo
200 University Avenue West, Waterloo, Ontario, N2L 3G1, Canada
l4ng@engmail.uwaterloo.ca, jph@surya.uwaterloo.ca

Sensor-based control of multiple robot systems require a large number of sensors and robotic motor units to be integrated. As the demand for agile manufacturing cells to become more adaptable and reconfigurable in terms of both hardware and software increases, data management and coordination of the numerous connected units become difficult. To attain this flexibility or agility, an operating system environment tailored for complex robot systems has been investigated and developed, based on the low-cost/high performance computing hardware made available by the personal computer industry. This paper begins with the description of the system architecture of our environment which promotes extensive integration, reconfiguration and expansion. It then proceeds to describe the implementation on our chosen real-world application, autonomous robotic welding which involves 3 robotic arm manipulator units and sensor integration. Finally, we discuss the merits of our system as well as future possibilities.

INTRODUCTION

Agile manufacturing robotic systems require extensive integration of multiple sensors with robot motor units, high adaptability and reconfiguration. Huissoon and Strauss [1] at the University of Waterloo developed an autonomous robotic welding system, an early agile manufacturing system, which involved the integration of a line-laser sensor with a robot arm manipulator and workpiece positioner. Based on the technology of the early 1990's, the addition of sensors and robot units to the system required substantial programming effort and centered around the robot itself.

Research has been done in the area of distributed networks and information sharing for work cell based robots [2],[3]. Wurll et al [3] proposed a distributed network using the CAN bus protocol to interconnect units. However, the near commodity availability of ethernet hardware coupled with the widespread adoption of the internet make TCP/IP ethernet a more suitable and cost effective interconnect and protocol. Therefore, in this study, a distributed network architecture was chosen with each unit acting as a device coordinated by a central intelligence connected via ethernet.

The objective of this paper is to explain the development and implementation of a distributed operating environment for complex robotic systems. It is a data-centric architecture which treats each robot or sensor as a device or unit in a collective system controlled by a centralized intelligence. The issues involved with using this operating environment for a real-world application, autonomous robotic fixture-less welding will also be presented.

SYSTEM ARCHITECTURE

At the core of our distributed operating environment is the central intelligence computer referred to as “Brain”. Connected via high-speed ethernet are the multiple sensor and robot units which make up the agile manufacturing cell.

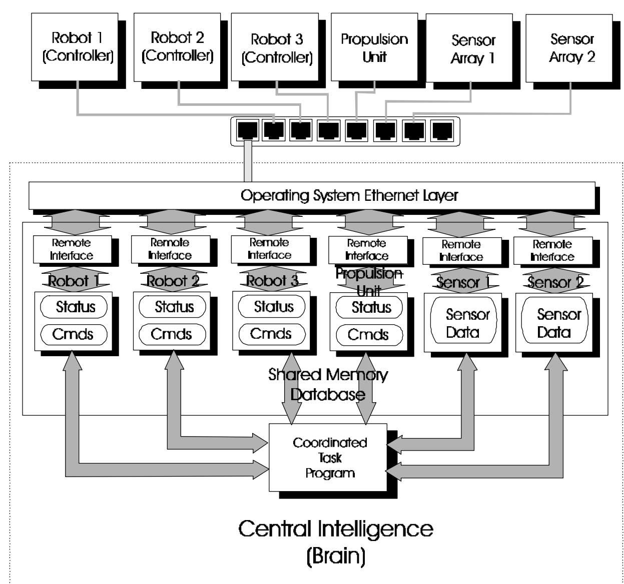

Figure 1. System Architecture

The “Brain” computer has software which manages the automatic detection and configuration of units attached to the ethernet backbone. It also has the capability of uploading updated system software to the robot or sensor itself.

Upon executing a particular task, the device manager mounts the required sensors or robotic units. It involves the creation of shared memory databases for each unit which includes status/sensor information and commands. As each units’ status data is refreshed, commands are sent to each robot unit. To synchronize the refreshing of shared memory, the "Brain" has a synchronization event referred to as the "Heart Beat" which is a 60Hz clock (application specific). It is in reference to this clock that all data transfers are initiated, guaranteeing a control action delay of no more than 33ms (2 cycles) on any sensor data. Since data is readily accessible on the "Brain", real-time monitoring and user intervention is possible.

This system architecture requires its operating system to have i) user assignable-priority to processes, ii) multi-tasking, iii) real-time performance, iv) low-level I/O access and v) a networking infrastructure (TCP/IP). The operating system we chose is QNX 4 [4], since it fulfills all the above requirements and contains a rich set of features which promotes real-time programming.

IMPLEMENTATION

In order to test the distributed operating environment on a real-world application, a task program has been developed. It is an autonomous welding program capable of welding a generic seam with no preplanning. This program consists of i) reading positional data from each robot client's shared memory area, ii) reading positional surface data from a line laser scanner, iii) using kinematic relationships to determine the positional commands of each robot in order to perform a coordinated weld on a generic part and iv) writes these commands to each robot client's shared memory area. We are presently in the preliminary stages of testing.

The existing welding robot, a Reis Robotstar V15 arm manipulator controller, was rebuilt specifically for the distributed operating environment. A key advantage of this robot's architecture is that it allows the motion controller to be completely implemented in software. Currently, an ethernet capable Intel Pentium processor board has replaced the aging dual Motorola 68020 processor boards which resided in the Reis Robotstar's VMEbus chassis. Its rebuilt system software now provides high performance positional motion control as well as ethernet communication to the distributed network. The software written for the Reis Robotstar V15 is modular, layered and generic so as to promote easy porting to other robot controllers.

Presently, a GM Fanuc S360arm manipulator for workpiece positioning is in the process of being integrated into the distributed operating environment. Due to the proprietary bus architecture of this robot, extensive modifications are necessary. These modifications involve routing the joint encoder signals to a secondary processor and injecting joint commands to the servo motors of each joint. Hence a secondary independent controller can control the robot manipulator and interface via ethernet with the distributed operating environment, while the primary controller is unaware of any motion from the robot. In addition, a workpiece positioning robot arm manipulator, the GM Fanuc S400, has also been incorporated into the network.

A line laser scanner has been integrated into the distributed environment. This scanner profiles a welding seam and determines the seam's location with respect to the scanner using computer vision techniques. The data made available to the "Brain" is in fact seam-tracking data in the form of seam coordinates and surface orientation.

In addition to sensors and robots, input and monitoring systems have been incorporated into the distributed environment. These include i) a joystick for real-tme user intervention, ii) TCP/IP server and JAVA remote clients for internet teleoperation,and iii) record and playback capabilities. A real-time remote 3D visualizer for the work cell is currently under development and will also be incorporated.

CONCLUDING REMARKS

As robotic systems become more complex, the management of data becomes crucial for mission success. Hence, a system such as the one described developed specifically for plug and play integration strives to make robot integration and reconfiguration simple in terms of hardware and software. Robots can be considered as devices whose system software can be completely reconfigured by the central intelligence if required.

For this implementation, the QNX network protocol was used to facilitate initial development. However, we will port to a TCP/IP implementation so that any unit equipped with modern operating system would be able to act as a device. Future research involves the development of additional task modules as well as expanding the sensor/robot database so that the "Brain" can acquire more skills and tools.

REFERENCES

[1] Huissoon, J.P; Strauss, D.L. Automatic Control of a Robotic Gas Metal Arc Welding Cell. Proceedings of Fifth World Conference on Robotics Research, Cambridge, Mass., 1994, pp 16-1 to 16-15.

[2] Cai, A; Fukado, T; Arai, F. Information Sharing Among Multiple Robots for Cooperation in Cellular Robotic System. IEEE Conference on Intelligent Robots and Systems. Vol 3 Sept 7-11, 1997, pp 1768-1773.

[3] Wurll, C; Henrich, D; Woern, H; Schoel, J; Damm, M; Meier, W. Distributed Planning and Control System for Industrial Robots. International Workshop on Advanced Motion Control, Jun 29-Jul 1, 1998, pp 487-492.

[4] QNX Software Systems Limited. QNX 4.25 Real-time Operating System. Kanata, Ontario, Canada. 1998.