Integrating New Sensors into a NASREM Implementation

Dr. Ronald Lumia
Robot Systems Division
National Institute of Standards and Technology
Bldg 220, Rm B127
Gaithersburg, Maryland 20899

Abstract

It is rare when the requirements of a system remain constant between the conception of the implementation and its delivery. Changes in specification occur and these modifications cannot be ignored in the methodology of developing systems. One way to approach this problem is to employ a functional architecture which is supple enough to support the inevitable changes without requiring significant redesignof the system. This paper discusses the integration of a new vision sensor, which supports active vision, into the existing robot system implementation at NIST. The lab implementation uses the NASA/NIST Standard Reference Model for Telerobot Control System Architecture (NASREM) as its functional architecture. The four degree of freedom active vision platform is described along with the process of its integration.

1. Introduction

The Intelligent Controls Group of the Robot Systems Division at NIST has been implementing the NASREM architecture for several years. NASREM is a hierarchical system with six levels where each level is divided into task decomposition, world modeling, and sensory processing. A more detailed description of NASREM can be found in [1]. To implement NASREM, interfaces for the modules were required. The interfaces for the lower two levels can be found in [2-7]. The approach has been to develop the architecture independently of the application by creating a robot control system testbed.

Since the NASREM architecture has been developed in a technology independent manner, there are two types of modifications which must be allowed. First, the architecture must be able to support different application areas. The NASREM architecture has been use for underwater vehicles [8], mining vehicles [9], autonomous land vehicles [10], as a basis for the Next Generation Controller, as well as for its originally stated space application. Second, the architecture must facilitate the integration of new sensors even if the application does not change so that the implementation may evolve with technology. This paper describes the evolution of the testbed when the capability of active vision is added.

The paper is organized as follows. First, a short description and rationale for active vision is provided. Then, the NASREM testbed is described both before and after the addition of the active vision capability. Then, a tracking capability using active vision is described. Finally, some application areas are suggested for future research.

2. Active Vision

Active vision is a new research area where controlling the motion of the sensor is just as important as the capabilities of sensor processing. This leads researchers away from the search for the "correct" image processing algorithm, e.g., optimal edge or corner extractors, toward the development of an understanding of the relationship between feature extraction and motion of the sensor.

In order to explore this direction the Intelligent Controls Group has constructed The Real-time Intelligently Controlled Optical Positioning System (TRICLOPS), four degree of freedom, three camera device, as shown in Figure 1. The first degree of freedom rotates the cameras in a manner analogous to that of a human neck. The second degree of freedom tilts all three cameras, allowing the device to look up and down. The central camera is rigidly attached to this degree of freedom. Since it is equipped with a wide angle lens, this camera provides images with a wide field of view but with low spatial resolution. The remaining two degrees of freedom are the independently controlled vergence cameras. The focus of this paper is the integration of TRICLOPS into the NASREM testbed.

3. NASREM Testbed

The layout of the NASREM testbed before the addition of TRICLOPS is shown in Figure 2 in the unshaded area. The system consists of two major parts: the hardware associated with the robot and the hardware related to the vision processing. The robot is the Robotics Research Corporation[1] (RRC) K-1607, a seven degree of freedom manipulator, which incorporates joint torque loops to minimize the effects of drive train nonlinearities, especially motor friction and harmonic drive compliance, to reduce the apparent motor inertia seen at the actuator output. The RRC controller provides all of the power electronics to drive each joint motor. The RRC controller provides an interface which allows an external computer system to issue joint torque commands to the manipulator every 2.5 ms.

The desired torques for each joint are calculated by the NIST developed controller in the Manipulation rack, a multi-processor system based on Motorola 680x0 processor boards in a VME backplane. Currently, there are seven processors in the Manipulation target system which execute the Primitive and Servo level code. The interface to the RRC controller is through an IEEE 488 parallel link. Code development is in ADA on a SUN-3 host, and the cross-compiled code is downloaded, along with a runtime kernel, to the target system. The SUNs are connected through an ethernet and a variety of software is available on the system to assist in the code development process.

The camera image is processed by the Pipelined Image Processing Engine (PIPE), a special purpose computing architecture designed specifically for image processing. PIPE is a preprocessor for iconic (spatially indexed) images. The output of PIPE is interfaced to the Perception rack through a dedicated 680x0 computer. Currently, the

---

[1] Products named in this paper are listed for purposes of information only. There is no implied endorsement of any products or implication that they are the best available for the purpose.

Perception rack has four computers to process the output of PIPE and interface with the World Model. The Perception rack is interfaced with a SUN host in the same way as the Manipulation rack.

A connection between the Manipulation rack and the Perception rack must exist to use the two systems in concert. This connection is easily removed during the debugging phase of program development so as to reduce contention for the system hardware by programmers.

4. Integrating TRICLOPS into the NASREM Testbed

As stated in the introduction, the ability to integrate new sensors into the system is critical for the evolution of a system. The active vision sensor, however, is significantly more complex than most sensors. It is a system, complete with its own computers for control. In spite of this, the integration was rapid for several reasons. First, relatively little software had to change, and even where changes were required, the organization provided by the NASREM architecture structured the changes. Second, the complete computing infrastructure could be used unchanged. Third, the active vision system required the addition of only 20% of the software to adapt the current system for use as the active vision platform.

The addition of TRICLOPS into the NASREM testbed is shown in Figure 2 as the shaded region. The procedure used to integrate active vision into the system was:

Clone a VME rack of computers

Copy and modify the NASREM controller used for the RRC robot

Develop the interfaces which are unique to TRICLOPS

Develop the unique algorithms associated with TRICLOPS

Perform process to processor assignment to achieve design specifications

4.1. Clone a VME rack of computers

For flexibility in research, as opposed to economic incentives, it was decided to create another VME rack of 680x0 processors to control TRICLOPS. The reason is that the contention for hardware is minimized during the development of the capabilities of the system. The three VME racks, as shown in figure 2, can have programmers independently developing software. When the entire system is to run in an integrated fashion, all of the backplanes must be connected together. All that this required, other than the purchase of the equipment, is to connect a bus extender card from the TRICLOPS VME rack to the Perception rack. From the standpoint of camera output, the only requirement is the obvious need to connect the TRICLOPS cameras to the inputs of PIPE.

4.2. Copy and modify the NASREM controller used for the RRC robot

The NASREM controller code for the RRC robot was simply copied for use as the controller for TRICLOPS. Most of the modifications are obvious. The world model must changed in order to have new forward and inverse kinematics, number of joints (4 for TRICLOPS, 7 for RRC robot), Jacobian, inertia matrix. These changes are relatively easy but more important, it is clear from the structure of the code imposed by the NASREM architecture precisely which processes must change.

4.3. Develop the interfaces which are unique to TRICLOPS

The information in the interface between TRICLOPS and its controller is the same. It consists of desired joint torques, with actual joint torque and positions as feedback. Therefore, the robot controller and TRICLOPS controller from that standpoint are identical. The physical mechanism associated with the interface, however, is different for the devices. Consequently, the connection between TRICLOPS and its controller, a standard parallel port, had to be implemented.

4.4. Develop the unique algorithms associated with TRICLOPS

Several unique algorithms were required for TRICLOPS. First, the vergence control algorithms, where the outer two cameras would verge on a single point in space were required. This was implemented as a new trajectory generation algorithm at the primitive level of the new TRICLOPS controller. Also, an algorithm which computes position from triangulation was required in the perception branch.

4.5. Perform process to processor assignment to achieve design specifications

In order to achieve human-like motion performance, a design specification was to have TRICLOPS' closed loop bandwidth to be 50 Hz. To achieve this, the sampling frequency for the motors associated with TRICLOPS is 2 KHz. This requirement is much more stringent than that of the robot controller. Consequently, the process to processor assignment, a process described in [4], needed to change to achieve this goal.

5. Conclusion

A functional architecture is helpful because it provides the basis for the organization of the hardware and software which implements it. When new requirements for a system dictate the addition of new sensors or even new systems, the functional architecture provides guidance for a smooth and rapid integration.

6. References

[1] Albus, J.S., McCain, H.G., Lumia, R., "NASA/NBS Standard Reference Model Telerobot Control System Architecture (NASREM)," NIST Tech. Note 1235, NIST, Gaithersburg, MD, July, 1987.

[2] Fiala, J., "Manipulator Servo Level Task Decomposition," NIST Technical Note 1255, NIST, Gaithersburg, MD, October, 1988.

[3] Kelmar, L. "Manipulator Servo Level World Modeling," NIST Tech. Note 1258, NIST, Gaithersburg, MD, December, 1989.

[4] Fiala, J., "Note on NASREM Implementation," NISTIR 89-4215, NIST, Gaithersburg, MD, December, 1989.

[5] Chaconas, K. J., Nashman, M., "Visual Perception Processing in a Hierarchical Control System: Level 1," NIST Tech. Note 1260, June, 1988.

[6] Wavering, A. "Manipulator Primitive Level Task Decomposition," NIST Technical Note 1256, NIST, Gaithersburg, MD, October, 1988.

[7] Kelmar, L. "Manipulator Primitive Level World Modeling," NIST Tech. Note 1273, NIST, Gaithersburg, MD, December, 1989.

[8] Albus, J.S., "System Description and Design Architecture for Multiple Autonomous Undersea Vehicles," NIST Technical Note 1251, NIST, Gaithersburg, MD, September, 1988.

[9] Huang, H.M., "Hierarchical Real-Time Control System for Use with Coal Mining Automation," Fourth Conference on the Use of Computers in the Coal Industry, Morgantown, WV, June 1990.

[10] Szabo, S., Scott, H.A., Murphy, K.N., Legowik, S.A., "Control System Architecture for a Remotely Operated Unmanned Land Vehicle," Proceedings of the 5th IEEE International Symposium on Intelligent Control, Philadelphia, PA, September, 1990.

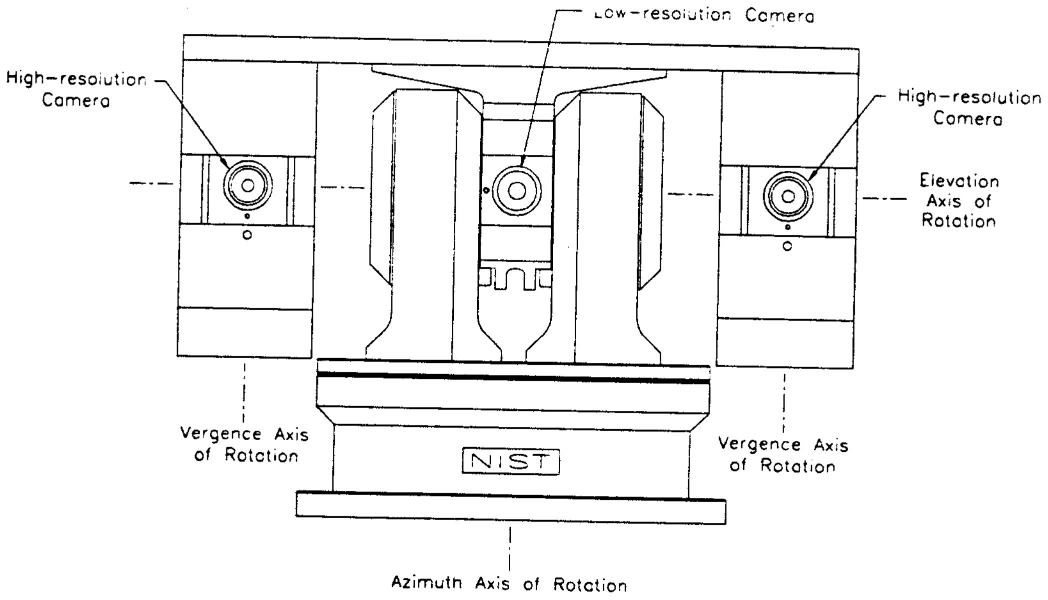

Figure 1. Front View of TRICLOPS

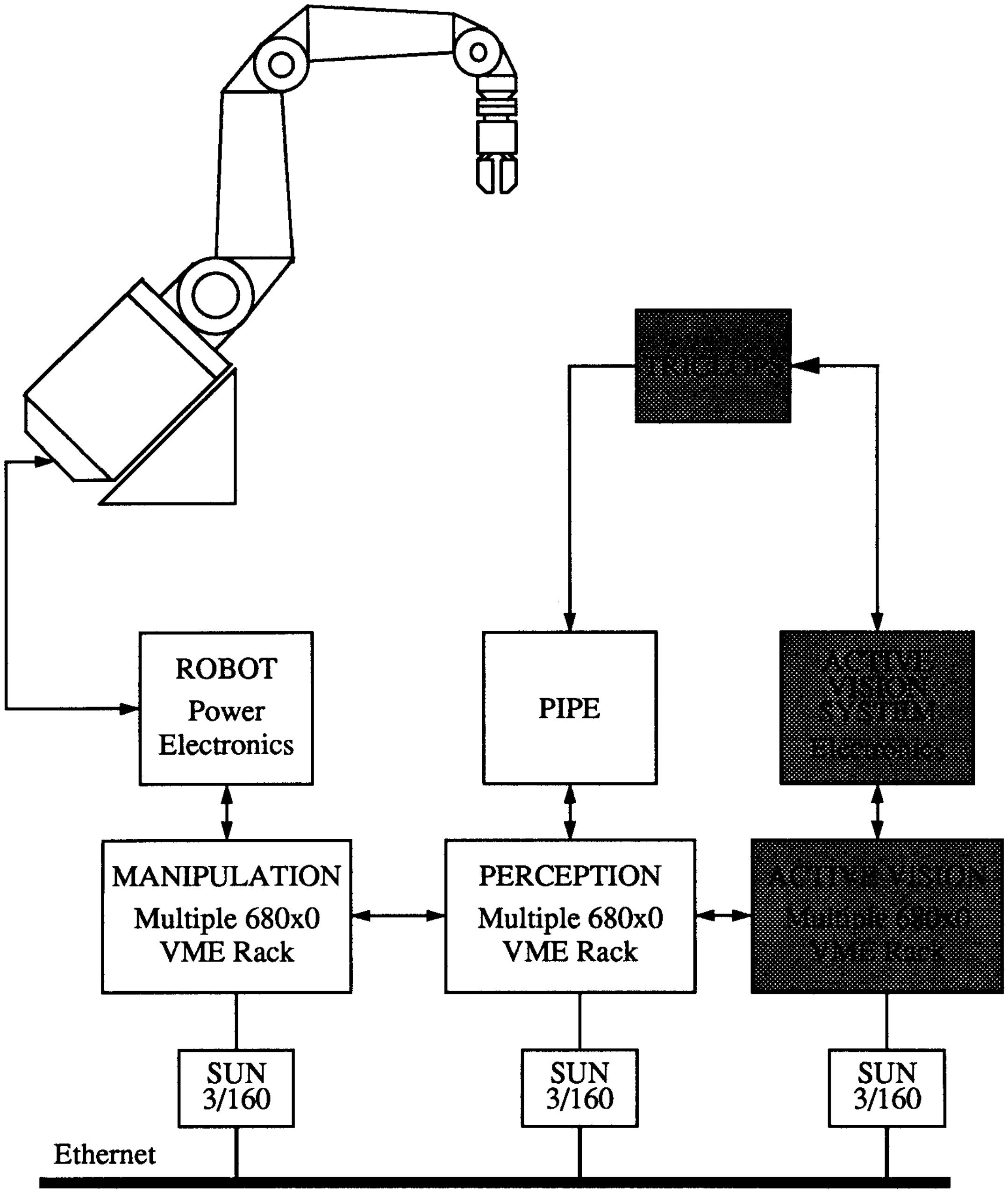

Figure 2. NASREM Testbed for Manipulation and Active Vision