A REFERENCE MODEL ARCHITECTURE
FOR
INTELLIGENT HYBRID CONTROL SYSTEMS

James S. Albus
and
Fred G. Proctor

Intelligent Systems Division
Manufacturing Engineering Laboratory
National Institute of Standards and Technology

Abstract: The Real-time Control System (RCS) is an intelligent systems architecture that integrates continuous and discrete control concepts in a hierarchically layered net of hybrid control nodes. In each node, discrete events sequence tasks and a feedback loop is closed at a cycle rate much faster than the dynamics of the process being controlled. At successively lower levels in the hierarchy, the rate of events and the feedback loop bandwidth increases by about an order of magnitude per level. RCS has been used in designing a number of hybrid control applications, most recently a four-axis machining center at the General Motors Powertrain plant in Pontiac, Michigan.

Key Words: Hybrid modes, real-time control, intelligent control, hierarchical control, reference architecture

1. INTRODUCTION

The Real-time Control System (RCS) developed at the National Institute of Standards and Technology and elsewhere over the past two decades (Albus, 1981, 1991, 1993; Meystel, 1993) is a reference model architecture for supporting the design of intelligent control systems and software. RCS combines discrete and continuous control concepts in a hierarchically layered set of processing nodes connected together by a network of communications pathways. A typical node in the RCS architecture is shown in Figure 1. At each layer of the RCS hierarchy, sensory data are processed, entities are recognized, world model representations are maintained, and tasks are decomposed into parallel and sequential subtasks, to be performed by cooperating sets of intelligent agents. Also at each level, feedback from sensors closes a control loop allowing each agent to respond and react in real-time to unexpected events. The result is a system that combines and distributes deliberative and reflexive features throughout the entire hierarchical architecture, with both discrete and continuous capabilities tightly integrated at all levels and time frames.

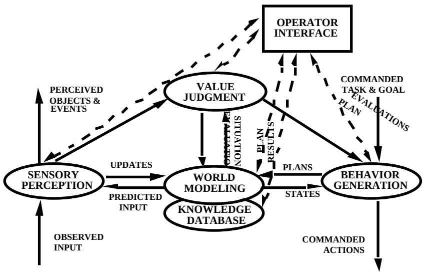

Figure 1. A typical node in the RCS reference model architecture.

Each node is constructed from four basic types of processing modules -- Behavior Generation (BG), World Modeling (WM), Sensory Perception (SP), and Value Judgment (VJ) -- plus a Knowledge Database (KD) and an Operator Interface.
In each node:

- Behavior Generation plans and controls action in order to achieve a commanded task goal. Behavior Generat uses information provided by World Modeling and Value Judgment to find the best assignment of tools and resources to agents, to select or generate the best plan of action, and to execute that plan.
- Sensory Perception scales and filters input from sensors, compares observed input with world model predictions, measures variance and uses correlation to detect events, and recognize objects and situations;
- World Modeling updates a Knowledge Database using variance and correlation results from Sensory Perception. World Modeling also answers queries, predicts sensory input, and simulates the result of hypothesized plans from Behavior Generation;
- Value Judgment computes cost, risk, and benefit. It assigns values to recognized objects and events, and to the results of simulated plans. Value Judgment also computes what is important (for attention), and what is rewarding or punishing (for learning).
- A Knowledge Database stores the data that support BG, WM, SP, and VJ functions in each node. Knowledge Database structures store information about the world in the form of state variables, entity and event frames, rules and equations, images and maps.
- An Operator Interface allows a human operator to interact with any of the functional processes, or to view the contents of the knowledge database.

2. THE COMMUNICATION SYSTEM

In addition to defining functional modules, the RCS reference architecture defines the communication system that moves information between them, both within the RCS nodes and between nodes. The communication system conveys commands from BG modules at one level to subordinates at the next lower level, and return status to supervisors at the next higher level. The communication system conveys tentative plans from BG planners to WM simulators. It transmits simulation results to VJ evaluators, and returns plan evaluations to BG Plan Selectors. It moves sensory data from sensors to SP filters, correlators, and detectors, and communicates detected entities and events to the WM for updating the KD. It transfers WM predictions, windows, and thresholds to SP comparators. It communicates perceived situations to the VJ modules for evaluation, and evaluations to the WM for updating the KD (Albus, 1993).

RCS does not specify a particular communication mechanism. Messages may be communicated by point to point message passing, network broadcast, shared common memory, or other mechanism depending on the preference of the system design engineer.

3. A HYBRID CONTROL ARCHITECTURE

Any node in the RCS architecture may implement a hybrid control system. Each of the functions within the nodes can be implemented by an augmented finite state machine. At the beginning of each compute cycle, each state machine exists in a particular state and reads input from a set of input buffers. It processes this information into a form suitable for a decision state-table, and searches the left hand side of the state-table for a line that matches the combination of input plus current state. When a match is found, the state machine goes to the next state, and executes whatever procedure is called for by that line in the state-table. It then writes an output to a set of output buffers. The communication system then moves information from the output buffers to the appropriate input buffers of other modules in the RCS system, and the cycle repeats. The sequencing of the state machines can be synchronous or asynchronous.

In RCS, the augmented state-machines are discrete event controllers. RCS arranges these discrete event controllers in a hierarchically layered architecture such that the frequency of discrete events is different at different levels. At all levels, the the cycle rate of the state-machines is designed to be about an order of magnitude faster than the highest dynamic frequency in the system being controlled. For example, at the bottom (servo) level of a RCS system for a high performance machine tool, the state-machine cycle rate might be as high as 30 kilohertz.

4. HIERARCHICAL LAYERING

The RCS architecture is organized as a hierarchical graph[1] in which nodes at the higher levels have broader scope with longer time horizons, and less detail. Lower level nodes have narrower scope with shorter time horizons, and more detail. Behavior Generation modules in nodes at the upper levels make long range plans consisting of very general goals, while at lower levels, Behavior Generation modules successively refine the long range plans into short term tasks with greater detail. At lower levels, Sensory Perception modules process data over local neighborhoods and short time intervals; while at higher levels, Sensory Perception modules integrate data over long time intervals and large spatial regions. At low levels, World Model data is short term and high speed with fine resolution; while at the higher levels it is broad in scope and general.

At all levels BG modules accept tasks and priorities from higher levels and make plans to accomplish those tasks. At all levels, feedback loops are closed to provide reactive behavior, with high-bandwidth fast-response loops at lower levels, and slower reactions at higher levels. At all levels, SP modules process input from sensors, or from lower level SP and WM modules, in order to provide feedback to close a control loop. At each level, there is a characteristic loop bandwidth, a characteristic planning horizon, a characteristic set of task skills, a characteristic range of temporal and spatial integration of sensory data. At the bottom of the hierarchy and external to the control system, are actuators that act on the world environment, and sensors that transform events in the world into information signals for the control system.

At every level, each node samples the state of the external world on each control cycle, and reacts as a discrete event controller. At each level, the rate of the control cycle is such that the effective closed loop bandwidth is high relative to the dynamics of the processes being controlled at that level. Thus, each node of the RCS architecture appears as though it is a continuous control system, even though it is implemented by a augmented state-machine with all the properties of a discrete event controller. The effect is that each node of the RCS system implements a hybrid control system. The discrete event control loop cycle is short enough so that the node acts as though it were a continuous controller – but it maintains all the properties of a discrete control system whose state and process parameters can be modified on any control cycle boundary.

5. APPLICATIONS

Over the past two decades, the RCS architecture has been used in the implementation of a number of experimental projects. These include:

5.1. A Horizontal Machining Workstation

This project was part of the NBS Automated Manufacturing Research Facility (AMRF) (Albus, *et al.*, 1982). It implemented a hybrid control system for a robot with a structured-light machine-vision system, a machine tool, an

---

[1] The structure is a graph, not a tree, because there are significant horizontal information pathways between modules and nodes at the same level.

automatic fixturing system, and a pallet shuttle. The robot included a quick change wrist, a part handling gripper with tactile sensors, and a tool handling gripper for loading and unloading tools in the machine tool magazine. The discrete event elements were represented as state-tables, and a wide variety of sensory interactive behaviors were demonstrated. These included locating and recognizing parts, determining the orientation of parts presented in trays, and automatically generating part handling sequences for part and tool loading and unloading (Wavering and Fiala, 1987).

5.2. A Cleaning and Deburring Workstation

This project was also part of the AMRF. It included two robots, a set of buffing wheels, a part washer/dryer machine, and a variety of abrasive brushes. Part geometry was input from a CAD database. Deburring parameters such as forces and feed rates were selected from a menu by an operator. Discrete event part handling sequences were automatically planned and executed for loading parts in a vise, and turning parts over to permit tool and gripper access. Continuous force sensing and control algorithms were used to modify the planned paths so as to compensate for inaccuracies in robot kinematics and dynamics (Murphy, *et al.*, 1988).

5.3. An Advanced Deburring and Chamfering System

This on-going project integrates off-line programming, real-time control, and active tool technologies in a hybrid control system. It automatically grinds precision chamfers on complex parts manufactured from hard materials such as aircraft jet engine components. The workstation consists of a grinding tool mounted on a micro positioner with computer controlled force and stiffness parameters, integrated with a 6 degree-of-freedom robot, and an indexing table for part fixturing. Part geometry is derived from standard IGES CAD data formats. Edge selection is performed by a human operator. Required tool force is automatically generated by formula using the cutting depth, feeds, and speeds input by the operator. Under a cooperative research and development agreement, a prototype production cell is currently being tested at Pratt & Whitney's East Hartford, CT site (Stouffer, *et al.*, 1993).

5.4. NBS/NASA Standard Reference Model Architecture for the Space Station Telerobotic Servicer (NASREM)

This project was sponsored by NASA Goddard Space Flight Center. NASREM was used by Martin Marietta to develop a hybrid control system for the space station telerobotic servicer. NASREM compliant algorithms have been developed for force servoing, impedance control, and real-time image processing of robotic and telerobotic systems at NIST, Martin Marietta Denver, Lockheed Palo Alto, Goddard, and in a number of university and industry labs in the United States and Europe (Albus, *et al.*, 1989).

5.5. Coal Mining Automation

This project transferred the RCS architecture and methodology to a team of researchers in the U.S. Bureau of Mines, and in turn, to the commercial mining industry. A comprehensive mining scenario was developed starting with a map of the underground region to be excavated, the machines to be controlled, and the mining procedures to be applied. Based on this scenario, a hybrid control system with simulation and animation was designed, built, and demonstrated. The same control system was later demonstrated with an actual mining machine and sensors (Huang, *et al.*, 1991).

5.6. An nuclear submarine maneuvering system

This ARPA sponsored project demonstrated the design and implementation in simulation of maneuvering and engineering support systems for a 637 class nuclear submarine. The maneuvering system involves an automatic steering, trim, speed, and depth control system. The system demonstrated the ability to execute a lengthy and complex mission involving transit of the Bering Straits under ice. Ice avoidance sonar signals were integrated into a local map using a CMAC neural network memory model (Albus, 1975). Steering and depth control algorithms were developed that enabled the sub to avoid hitting either the bottom or the ice while detecting and compensating for random salinity changes under the ice by making trim and ballast adjustments. The submarine engineering support system demonstrated the ability to respond to an emergency such as a lubrication oil fire by reconfiguring ventilation systems, rising in depth to snorkel level, and engaging the diesel engines for emergency propulsion (Huang, *et al.*, 1993).

5.7. A U.S. Postal Service Automated Stamp Distribution Center.

This RCS discrete event system demonstrated the ability to route packages through a series of carousels, conveyors, and storage bins, to maintain precise inventory control, provide security, and generate maintenance diagnostics in the case of system failure. The stamp distribution center was designed and tested first in simulation, and then implemented as a full scale system. The system contained over 220 actuators, 300 sensors, and ten operator workstations. An even larger and more complex RCS system for controlling a general mail facility is still under development (ATR Report, 1994).

5.8 Multiple Autonomous Undersea Vehicles

This hybrid control system was developed for controlling a pair of experimental vehicles designed and built by the University of New Hampshire. The RCS control system included a real-time path planner for sonar-guided obstacle avoidance, and a real-time map builder for constructing a topological map of the bottom. A series of tests was conducted in Lake Winnipasaki during the fall of 1987 (Herman and Albus, 1988).

5.9. Unmanned Ground Vehicles

Two versions of a hybrid control system for unmanned ground vehicles have been implemented on an Army HMMWV light truck. One version enables the vehicle to be driven remotely by an operator using TV images transmitted from the vehicle to an operator control station. This version has a retrotraverse mode that permits the vehicle to autonomously retrace paths previously traversed under remote control, using GPS and an inertial guidance system (Szabo, *et al.*, 1990).

A second version has demonstrated the ability to drive the HMMWV automatically using TV images processed through a machine-vision system with a real-time model matching algorithm for tracking lane markings. A World Model estimate of the lane markings is compared to observed edges in the image, and a new estimate is computed every 15 milliseconds, with pipeline latency of less than 150 milliseconds. The RCS real-time vision processing system has enabled this vehicle to drive automatically at speeds up to sixty miles per hour on the highway, and at speeds up to thirty-five miles per hour on a winding test track used by the county police for driver-training (Schneiderman and Nashman, 1994).

5.10. Planning and Control for a Spray Casting Machine.

The RCS hybrid architecture has been applied for planning and control of the automated Spray Casting Machine "OSPREY" which has been developed and manufactured by MTS Corporation (Minneapolis, MN) in cooperation with Drexel University. The system has three levels of resolution (Cleveland and Meystel, 1990).

5.11. An Autonomous Mobile Vehicle.

An autonomous vehicle was assembled and tested by Drexel University in 1984-1987. The goal of the effort was to investigate the RCS architecture with four levels of resolution "Planner-Navigator-Pilot" on the top of the lower level control of steering and propulsion. The results of this research are described in (Meystel, 1991).

5.12. An Open Architecture Enhanced Machine Controller

The RCS hybrid control model is currently being used as the basis for an open architecture Enhanced Machine Controller (EMC) for intelligent control of manufacturing equipment, such as machine tools, robots, and coordinate measuring machines. The EMC is a testbed for evaluating open architecture interface specifications. The EMC combines NASREM with the Specification for an Open System Architecture Standard (SOSAS) developed under the Next Generation Controller program sponsored by the Air Force and National Center for Manufacturing Sciences. In cooperation with the DoE TEAM (Technology for Enabling Agile Manufacturing) program, EMC functional modules have been defined, and Application Programming Interfaces (APIs) are being specified for sending messages between the functional modules. A prototype EMC has been installed and is being evaluated in the General Motors Powertrain prototype production facility in Pontiac Michigan as part of a DoE-TEAM/NIST-EMC government/industry consortium. The goal of this effort is to develop API standards for open architecture controllers (Proctor and Michaloski, 1993).

A block diagram of the EMC is shown in Figure 2. The EMC implemented at the GM Powertrain plant is on a 4-axis horizontal machining center with a tool changer and pallet shuttle system. It contains a commercial motion control board which closes the control loop on the X, Y, Z axes every 300 microseconds. At this rate, the output commands to the motor drives of the machine tool are indistinguishable from continuous control signals.

Higher level nodes in the EMC controller have a control cycle that runs every 20 milliseconds. These nodes provide input to the trajectory generator and spindle controller on the motion control board, as well as continuous motion output to the B axis motor drive, and discrete control signals to the tool changer, pallet shuttle, and miscellaneous actuators. All together the machining center has more than 100 discrete input/output points which must be sequenced precisely in order to effect proper loading and unloading of tools and materials. For these processes, 20 milliseconds is short compared to the dynamics of the system being controlled. Complete tool path motions, tool change operations, and pallet shuttle operations require many seconds to complete. At higher levels, events occur even less frequently. Machining tasks may take a number of minutes to execute. Therefore, to the machine, it appears as if the EMC is a continuous controller.

Yet the EMC can switch control modes on any 20 millisecond cycle boundary. During each computation cycle, the controller examines the input, matches it with the state-transition conditions in a state table, and if a match occurs, the system switches to a new state. In principle, each state could implement a different control algorithm. An operator can interrupt the system, switch control modes from automatic to manual, or give a new feed rate override command every 20 milliseconds. The RCS system is thus a hybrid controller in which all events, both continuous and discrete, are handled at rates that make the distinction between discrete events and continuous processes essentially disappear.

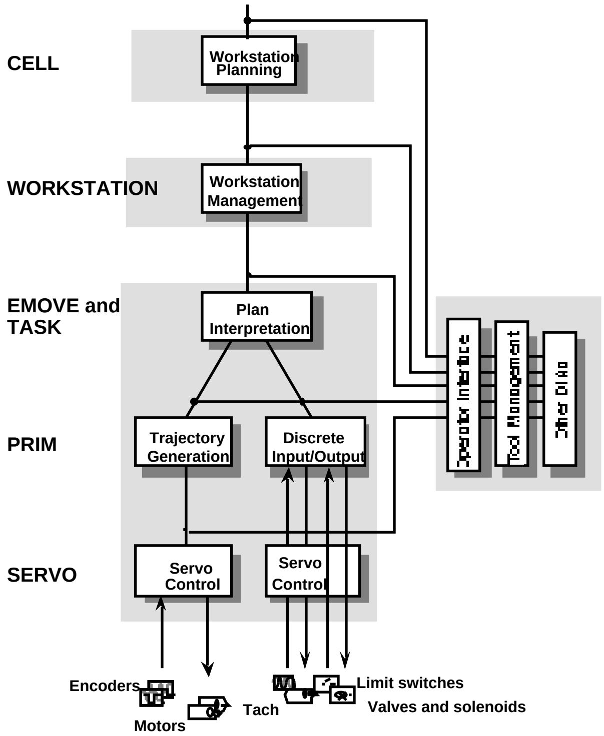

Figure 2. Behavior Generation hierarchy and Operator Interface for the Enhanced Machine Controller installed on a 4-axis flexible machining center at General Motors Powertrain. The EMC control hierarchy is composed of the Task, Elemental-Move, Primitive, and Servo levels. The Workstation and Cell level reside in existing design facilities at GM.

Current work at NIST and elsewhere is pursuing more complex implementations of RCS. For example, efforts to incorporate human operator interfaces into the RCS architecture that began with NASREM have continued with the Air Force/JPL/NIST Universal Telerobotic Architecture Project (UTAP), and the NIST RoboCrane. Work is also in progress to develop an engineering design methodology and a set of software engineering tools for developing RCS systems (Quintero and Barbera, 1993).

REFERENCES

Albus, J.S. (1975). A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC), and Data Storage in the Cerebellar Model Articulation Controller (CMAC), *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, September.

Albus, J.S. (1981). *Brains, Behavior, and Robotics,* Byte/McGraw-Hill, Peterborough, N.H.

Albus, J.S., McLean, C.R., Barbera, A.J. and Fitzgerald, M.L. (1982). Architecture for Real-Time Sensory-Interactive Control of Robots in a Manufacturing Facility. *Proceedings of the Fourth IFAC/IFIP Symposium -- Information Control Problems in Manufacturing Technology*.

Albus, J.S., McCain, H.G., and Lumia, R. (1989). NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM). *NISTTN 1235 (supersedes NBS Technical Note 1235, July 1987),* National Institute of Standards and Technology, Gaithersburg, MD.

Albus, J.S. (1991). Outline for a Theory of Intelligence. *IEEE Transactions on Systems, Man and Cybernetics,* **Vol. 21, No. 3**, pp. 473-509.

Albus, J.S. (1993). A Reference Model Architecture for Intelligent Systems Design. In: *An Introduction to Intelligent and Autonomous Control*, (Antsaklis, P.J., and Passino, K.M., (Ed.)), pp. 27-56, Kluwer Academic Publishers, Boston

ATR Report. (1993) Stamp Distribution Network, *USPS Contract Number 104230-91-C-3127 Final Report*, Advanced Technology & Research Corp, Burtonsville, MD., 20866-1172.

Cleveland, B., Meystel, A. (1990). Predictive Planning + Fuzzy Compensation=Intelligent Control. *Proceedings of the 5th IEEE International Symposium on Intelligent Control*, Philadelphia, PA.

Herman, M. and Albus, J.S. (1988). Overview of the Multiple Autonomous Underwater Vehicles (MAUV) Project. *Proceedings of IEEE International Conference on Robotics and Automation*, Philadelphia, PA.

Huang, H.M., Quintero, R. and Albus, J.S. (1991). A Reference Model, Design Approach, and Development Illustration toward Hierarchical Real-Time System Control for Coal Mining Operations. In: *Advances in Control & Dynamic Systems*, (C.T. Leondes (Ed.)), **Vol. 46, part 2 of 5,** pp. 173-254 Academic Press, San Deigo, CA.

Huang, H.M., Hira, R. and Quintero, R. (1993). A Submarine Maneuvering System Demonstration Based on the NIST Real-Time Control System Reference Model. *Proceedings of the 8th IEEE International Symposium on Intelligent Control*, Chicago, IL.

Meystel, A. (1991). *Autonomous Mobile Robots: Vehicles with Cognitive Control,* World Scientific, Singapore

Meystel, A. (1993). Nested Hierarchical Control. In: *An Introduction to Intelligent and Autonomous Control*, (Antsaklis, P.J., and Passino, K.M. (Ed.)), pp. 129-161, Kluwer Academic Publishers, Boston.

Murphy, K.N., Norcross, R.J. and Proctor, F.M. (1988). CAD Directed Robotic Deburring. *Proceedings of the Second International Symposium on Robotics and Manufacturing Research, Education, and Applications,* Albuquerque, NM.

Proctor, F. and Michaloski, J. (1993). Enhanced Machine Controller Architecture Overview, *NISTIR 5331,* National Institute of Standards and Technology, Gaithersburg, MD.

Quintero, R., Barbera, A.J. (1993). A Software Template Approach to Building Complex Large-Scale Intelligent Control Systems. *Proceeding of the 8th IEEE International Symposium on Intelligent Control*, Chicago, IL.

Schneiderman, H. and Nashman, M. (1994). Visual Tracking for Autonomous Driving. *IEEE Transactions on Robotics and Automation,* **Vol. 10, No. 6,** p.769-775.

Senehi, M.K., Kramer, T.J., Michaloski, J., Quintero, R., Ray, S.R., Rippey, W.G., Wallace, S. (1994). Reference Architecture for Machine Control Systems Integration: Interim Report. *NISTIR 5517,* National Institute of Standards and Technology, Gaithersburg, MD.

Stouffer, K., Michaloski, J., Russell, R. and Proctor, F. (1993). ADACS - An Automated System for Part Finishing. *NISTIR 5171*, National Institute of Standards and Technology, Gaithersburg, MD., and *Proceedings of the IECON '93 International Conference on Industrial Electronics, Control and Instrumentation*, Maui, Hawaii.

Szabo, S., Scott, H.A., Murphy, K.N. and Legowik, S.A. (1990). Control System Architecture for a Remotely Operated Unmanned Land Vehicle, *Proceedings of the 5th IEEE International Symposium on Intelligent Control,* Philadelphia, PA.

Wavering, A.J. and Fiala, J.C. (1987). Real-Time Control System of the Horizontal Workstation Robot, *NBSIR 88-3692*, National Institute of Standards and Technology, Gaithersburg, MD.