

Towards Rapid Implementation of Adaptive Robotic Systems

Col. M. Leahy, PhD
USAF
Washington, DC

G. Oreind
ACG, Inc
San Jose, CA

F. DaCosta
ACG, Inc.
San Jose, CA

P. Pignon PhD
ACG, Inc.
San Jose, CA

R. Lumia, PhD
NIST
Gaithersburg, MD

Introduction

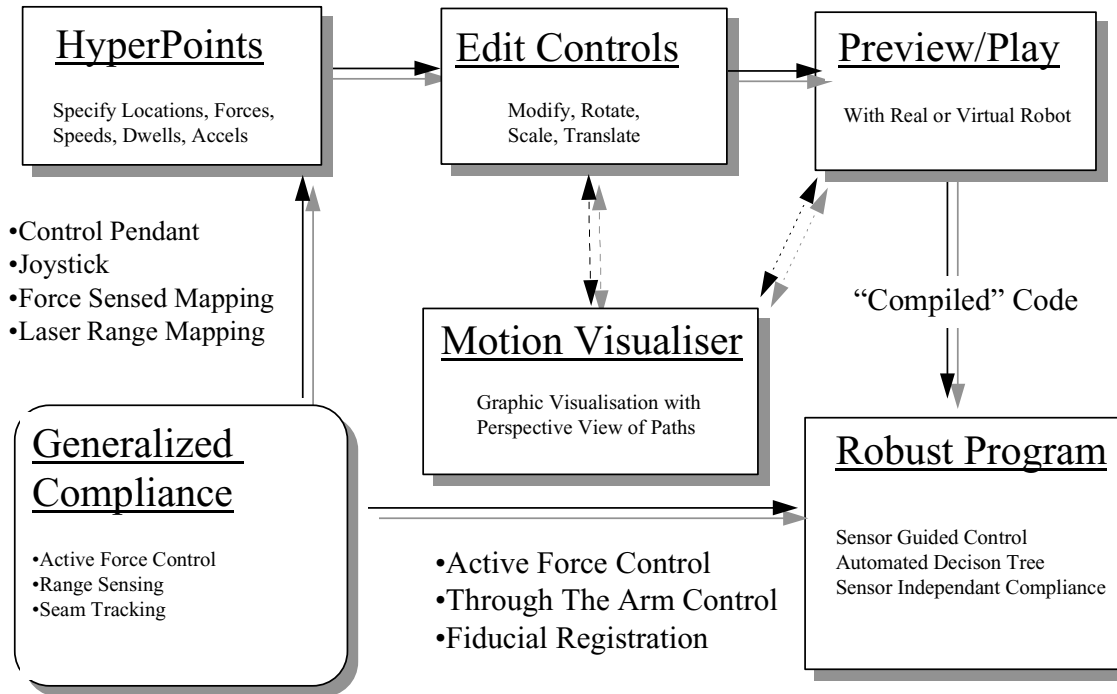


Figure 1: Lego block approach Adaptive Real time Embedded System Design

Current automation design practice produces expensive one-of-a-kind installations where the system cannot be easily modified to meet changing demands or advancements in technology. It is imperative that we design robot systems to be modular, portable and easily re-configurable in order to reduce the design lead times and life cycle costs of providing automation alternatives.

The Unified Tele-robotics Architecture Program (UTAP) was developed under the sponsorship of the US Air Force Robotics and Automation Center of Excellence. A goal of the program was to define and develop prototypes of commonly used software building blocks for sensor guided real time embedded control of telerobotic devices. Standard building blocks and a non-proprietary communication protocols would provide the Air Force and specifically the Logistic Centers with a support infrastructure designed to rapidly and efficiently build and maintain mission critical automation systems.

This paper restricts itself to unclassified applications of the technology. Three (unclassified) applications of interest to the Air Force were explored: Paint Stripping, Surface Finishing/Grinding and vision based inspection. The three applications were chosen because of striking commonalities:

- Some form of dynamic contour following needed, with motion parameters altered in real time
- A multi sensor situation is typical: Multiple inputs collectively define the new set point.
- The software modules are identical in function, if the data passed can be parameterized correctly.

Based on these observations, ACG implemented a "flexible" programming environment for sensor guided contour following tasks. All software, including the visualization sub system, was implemented in our real time OS. The software is being used at Schlage Lock to rapidly generate all the programs for robotic door handle polishing cells. A subset of the software system is in operation at Warner Robins AFB, where it is being used to strip paint off F-18 aircraft.

We briefly describe three modules that are applicable to non-military applications: the Hyper Point Data structure, The Generalized Compliance Module and the Robot Motion Visualization Module.

The Hyper Point data structure and usage

Contour following is the process of moving along a path while changing motion parameters regulates the motion along the path.

Each point along the path has a unique location and associated process value. Process parameters include the force, speed and peripheral device signals to be set at that point.

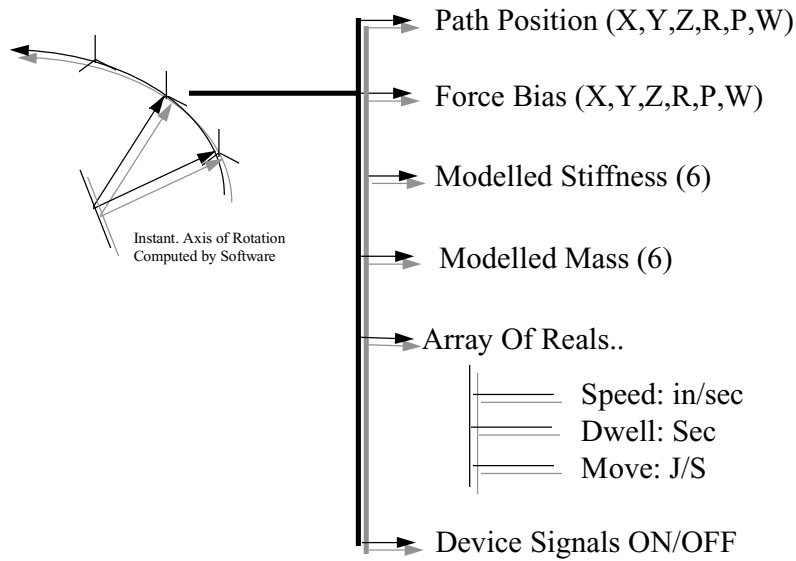


Figure 2: An Example of Hyper Point data structure applied to polishing tasks

The user teaches a few control points manually and the software then generates a smooth continuous path that links all the control points together. Both arc and line motions are created automatically to blend in smooth profiles. In the Edit mode, the operator can play back, fast forward, rewind and edit (*e.g.* insert, modify or delete) these Hyper Points. The Hyper Points are stored as program files by the system. These files contain all the data the robot controller needs to perform the polishing task for that part.

Generalizing Compliance

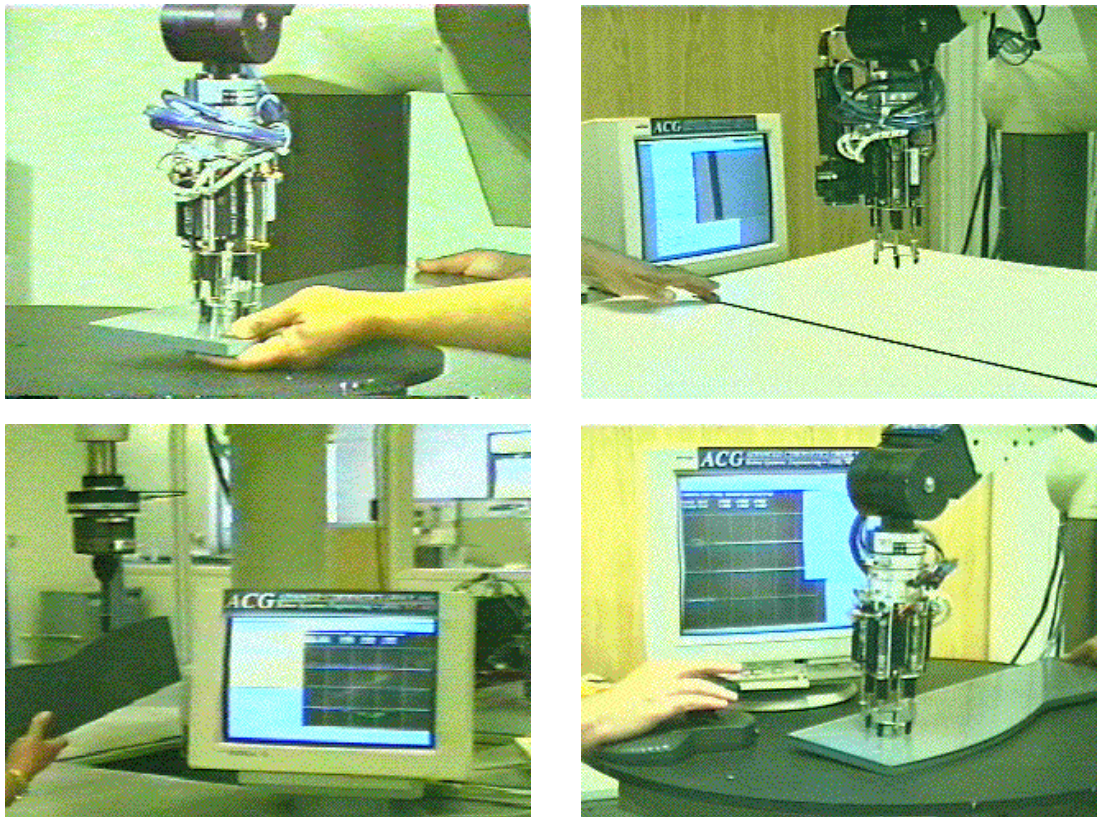


Figure 3: Applications of Sensor Independent Compliance Module

The compliance module has been implemented as a second order filter for any number of input channels. The system response is that of a spring mass damper, with the spring mass damper values being set by the control program. Each sensor has its own characteristics filter parameters. The module has been employed to (Figure 3 clockwise from top):

1. Maintain normalcy to the surface using range data (Paint stripping applications)
2. Follow a moving black wand (simulating seam tracking)
3. Using Force sensing to map the contour of the surface for the purposes of digitizing it.
4. Integrating Joystick Controls in order to move along a surface yet stay normal to it (Shared Control)

The Motion Visualiser

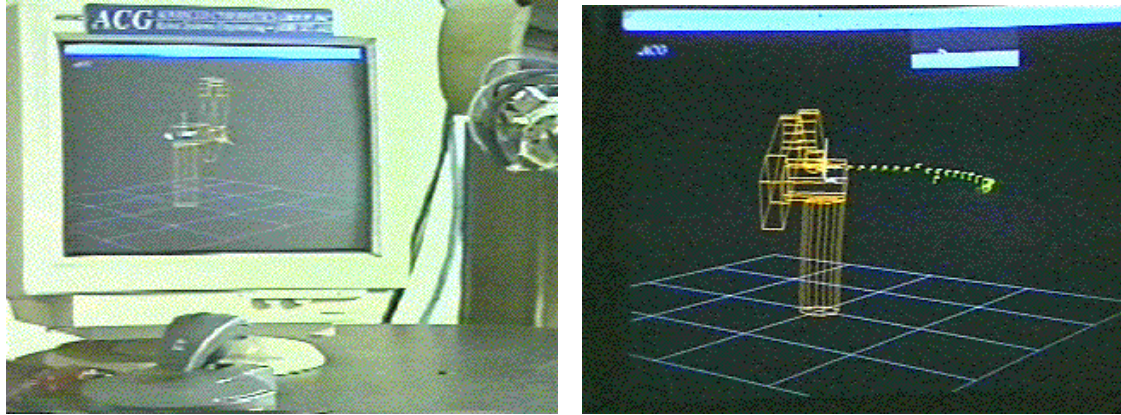


Figure 4: The Robot Motion Visualiser being used to check for reachability

The objective of motion visualization is to provide the operator with low cost tools to preview robot motions and check for joint envelope errors or out of reach conditions before downloading code to the robot. The visualiser shows an animation of a set of proposed motions, bounding box collision detection and path traces. It has a low level communications interface that continuously checks the current location of the robot and updates the simulation model. The simulation model, therefore, accurately reflects the current location of the robot. The motion visualiser is also integrated with the compliance module to ensure dynamic on-the-fly robot motion planning and collision avoidance. – Commands from the joystick are “checked” by the system before being passed on to the robot controller via the remote communications link.

The remote communications link between the simulation environment and the robot must work under “difficult” situations. A prototype system that guaranteed Isochronous network performance was implemented and tested. The core concept was to extend existing and understood self-healing mesh network approaches to support low latency, guaranteed delivery traffic.

A “motion highway map” is also integrated with the visualiser. The highway map eliminates the arduous task of specifying specific via motions to avoid obstacles. Instead, we supply the motion manager with a map of the highways for the current cell and a set of algorithms to compute, depending on the current robot location, the best route. If the cell is reconfigured, some highways became “closed” and the system automatically - and in real time - re-computes the new route. A potential field approach to collision avoidance was developed and implemented. A proof of concept prototype was developed. The pictures show our real time embedded controller controlling the Puma robot. Videotape footage available on request.

[End of disclosure](#)