Available online at www.sciencedirect.com

SCIENCE DIRECT®

Engineering Applications of Artificial Intelligence 17 (2004) 469–483

Engineering Applications of
ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

Development and implementation of a real-time open-architecture control system for industrial robot systems

J.S. Gu[a], C.W. de Silva[b,*]

[a] Pedestal Networks, Fremont, CA 94555, USA

[b] Department of Mechanical Engineering, University of British Columbia, 2324 Main Hall, Vancouver, BC, Canada V6T 1Z4

Accepted 12 March 2004

Abstract

This paper presents a real-time open-architecture control system (ROACS) which has been developed for flexible manipulation of industrial robots. Flexible manipulation refers to robot manipulation that handles tasks with uncertainties; hence, decision-making based on feedback data is essential in realtime operation. A real-time open-architecture control system with the capacity of parallel processing of realtime events, extraction of information from realtime data, and intelligent decision-making, is developed. The entire system consists of a real-time subsystem which manages robot hardware and executes path planning and data processing, and an intelligent subsystem which performs intelligent decision-making and feedback task control. In the context of intelligent task control, information extraction, fuzzy-logic-based interpretation and decision-making, and a novel design of associated real-time robot task language (RTTL) are developed. The conflicts between high bandwidth requirements for real-time services and the undeterministic time length for intelligent decision-making are managed in a cooperative real-time intelligent system model. Client–server architecture is found quite suitable for implementation of the system. The entire system has been successfully developed, implemented, and demonstrated for a robotic salmon slicing task which requires online determination of the backbone position.

Keywords: Robot control system; Task uncertainty; Intelligent task control; Mechanical impedance

1. Introduction

Robot manipulators are particularly suitable for flexible production operations where small batches of a range of different products are manufactured. A large majority of successful industrial robots are designed for carrying out tasks where accurate positioning would be important (De Silva, 2004; Unimation, 1985), and the positions have to be accurately specified in the robotic task commands. Force and mechanical impedance considerations are of minor significance here and are neither specified in the task description nor provided by the robot control system. However, there are many other process applications where tasks cannot be described in position specifications alone and force and impedance requirements become as important as the position requirements (De Silva, 2004; Gu, 1999). For example, in robotic tasks of interacting with a rigid environment (e.g., a parts assembly process), a very small change in displacement against a hard surface would result in a large force. In such tasks, position control can be viewed as an ill-posed problem, where exact positioning is almost impossible. Force control would be relatively easy and more appropriate in these circumstances. Fig. 1 illustrates a salmon slicing task, where a fish is sliced up to its backbone in the sequence a_1, to c_1, a_2 to c_2,..., and then the steaks are removed from the bone using a transverse cut sequence c_0 to c_1, to c_2, ..., to c_m. Steaks that are produced in this manner will be boneless.

Exact measurement of the bone positions as given by the locations c_1, c_2,...,c_m, is very difficult and may even be infeasible. Hence, online detection of bone by sensing the mechanical impedance of the process, and using some a priori knowledge of the impedance differences that are manifested in cutting meat and bone, would be

*Corresponding author. Tel.: +1-604-822-2781; fax: +1-604-822-2403.

E-mail address: desilva@mech.ubc.ca (C.W. de Silva).

0952-1976/$ - see front matter
doi:10.1016/j.engappai.2004.03.010

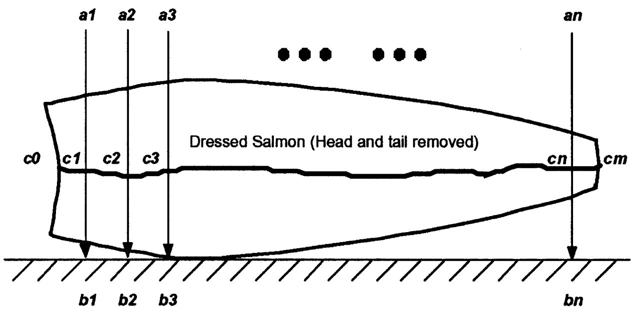

Fig. 1. Illustration of the motion sequences in a salmon slicing task.

much more efficient and effective. Execution of tasks of this type would benefit from flexible manipulation, which could handle uncertainties in task specifications through online feedback of sensory data and intelligent decision-making.

Flexible manipulation is defined in the present context to mean the robot manipulation where interpretation and high-level feedback of online sensory data along with process knowledge and intelligent decision-making would be employed to resolve uncertainties and variations in a "flexible" task specification. The associated "unknowns" may be positions that are required to specify the task for robot motion control, or even be process-dynamic characteristics. Flexible manipulation has two major features:

- *Sensor-based manipulation*: Without online sensory information, position unknowns may not be determined and corrected.
- *Intelligent manipulation*: Online detection of process characteristics would be based on process knowledge which also may incorporate high-level interpretation (abstraction) of low-level sensory data, and will require intelligent decision-making procedures.

Flexible manipulation of this type will challenge the traditional sensing and control technologies in robotics. First, the entire robot control system, including task-level control, not just the low-level robot servo control, will require process information and online feedback. Reliable and cost-effective sensing technologies are essential in flexible manipulation. Second, a traditional position-based task control system will not be effective in a wide range of robotic processing tasks, as noted before, and cannot generally handle high-level information feedback and intelligent decision-making. As will be emphasized in the sequel, impedance-based task control is particularly appropriate in this regard. An open-architecture control system is necessary for this purpose. The system should be both "real-time operational" for online control and "intelligent" for high-level decision-making. The conflicts between real-time performance and intelligent decision-making should be resolved properly in such a control system. The main requirements of the controller are summarized below:

- *Sensory feedback*: In most commercial designs of robot control systems, sensory data are fed back only to the position servo controllers. The high-level procedures of motion planning and task control adopt an open-loop architecture. As a result, tasks have to be programmed in motion commands, with no flexibility of modification and error resolution at the task level. The controller that is developed in the present work will process sensor feedback capabilities in levels of both behavior (motion) planning and task control, in addition to the servo level. This would provide the robot controller the capability to update online its operation and desired motion, based on sensory information, and thereby schedule new tasks accordingly, perhaps incorporating knowledge-based intelligent decision making.
- *Real-time intelligence*: In a robot control system, proper real-time performance of the low-level controller has to be guaranteed. At the same time, the control system has to perform some decision-making in order to ascertain the system performance and so as to plan the task properly. Again, knowledge-based intelligent decision making capability, in real time, will be important. The cooperation, synchronization and communication between the real-time and intelligent subsystems would be required here.
- *Open architecture*: Implementation of new functions when needed would be desirable in a flexible control system for robots. A client–server architecture can make a robot control system open to further development. Changes to robot functionality can be realized through modifying high-level modules, without altering the low-level, open-architecture control system.

In the salmon slicing task, it is quite difficult to measure the position of the backbone for each fish. It is also difficult to move the robotic cutter to the exact backbone position even if it is known, because the fish will be deformed during processing, and furthermore, some shifting of the fish will be possible due to soft handling. Position control alone is not adequate to properly perform this task. It is intuitively clear that the mechanical impedance of cutting a bone would be much higher than that of cutting meat. Hence, online detection of the backbone using impedance information at the cutting interface would be feasible, and this knowledge would be very helpful for commanding the robot cutter in carrying out the task. The uncertainties of the cutting position and the process environment in task description could be remedied by using impedance specifications.

The reminder of the paper is organized as follows: Section 2 will provide an hierarchical architecture in function levels of ROACS; Section 3 will present an