JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 24, No. 2, March–April 2001

Hierarchical Control of a Space-Based Deployable Manipulator Using Fuzzy Logic

J. F. Goulet,* C. W. de Silva,† and V. J. Modi‡
University of British Columbia, Vancouver, British Columbia V6T 1Z4, Canada
and
A. K. Misra§
McGill University, Montreal, Quebec H3A 2K6, Canada

Application of the fuzzy logic intelligent control of space-based systems has received relatively less attention compared to conventional control procedures. Basic concepts involved in development of a hierarchical control system using fuzzy logic are explained. Next, fuzzy logic is applied to control a novel two-link robotic arm, composed of revolute and prismatic joints, supported by a mobile base traversing along an orbiting platform. The control system that is developed has three levels. A conventional controller, consisting of the feedback linearization technique combined with proportional-derivative control is used in the bottom level of the hierarchical system, to control the servomotors of the robot. A second layer consists of a servo-expert that preprocesses the high-resolution information coming from joint encoders and extracts the status of the system. A third, intelligent layer is added at the top of the hierarchy to complete the control structure. The main purpose of the top level is to tune the parameters of the crisp (conventional) controller to improve the response of the system. A decision table is developed offline, for tuning the parameters of the crisp (nonfuzzy) controller, which considerably reduces the real-time computational effort. Typical pick-and-place results from the robot are presented that show that extra tuning can significantly improve the performance of the manipulator.

I. Introduction

A HIERARCHICAL structure having a high-speed conventional (crisp or nonfuzzy) controller at the bottom layer and an intelligent tuner at an upper layer is developed in the present work and used in the specific context of controlling a space-based planar deployable manipulator. The robot, with deployable and slewing links, is positioned on an orbiting flexible platform (Fig. 1). The robot dynamics are modeled using an order N algorithm based on the Lagrangian approach and velocity transformations. The feedback linearization technique (FLT), with proportional-derivative (PD) control, is used to compensate for system nonlinearities and to regulate the rigid degrees of freedom. The intelligent tuner at the top level of the hierarchical system is developed from a validated set of linguistic rules for PD servos, using fuzzy system concepts. The performance of the hierarchical control system is evaluated on the basis of several test cases.

II. Motivation, Background, and Review

A. Motivation

One of the space projects that is receiving considerable attention is the International Space Station. The station will benefit from a mobile servicing system (MSS) for its construction, operation, and maintenance. The MSS consists of three parts: the space station remote manipulator system (SSRMS) having 7 degrees of freedom, the special purpose dexterous manipulator (SPDM) with 15 degrees of freedom for fine manipulations, and the mobile base system (MBS). The MBS is a platform that moves along the trail covering the length of the space station. The system would enable missions such as the launch and capture of satellites, the transportation of equipment, and the support of extravehicular activity. The SSRMS and the SPDM will normally operate under astronaut supervision. The astronauts will need sufficient training to use the robotic manipulator.

For systems of this nature that operate in a hostile environment, one important goal is to reduce the extravehicular activity by astronauts. Robotics and automation are identified as key technologies that are applicable in achieving that objective. Autonomous, semiautonomous, and dexterous space robots can reduce the workload of astronauts, improve safety, and increase the accuracy, repeatability, speed, and flexibility. Furthermore, they decrease the dependence on ground stations and communication links. Effective control strategies are essential for the robot system in realizing these objectives. There is a need for intelligent control and knowledge-based techniques, particularly due to the emphasis on nearly autonomous operation, with minimal human intervention.

Practical robots such as space manipulators are complex systems because they normally exhibit the following characteristics: nonlinearities, dynamic coupling, link and joint flexibility, model errors, parameter uncertainties, and unexpected disturbances.[1] Thus, space-based robotic manipulators represent a class of applications where knowledge-based control, which makes use of the operator experience, is particularly advantageous.[2]

B. Background

1. *Deployable Manipulator*

A variable-geometry manipulator has been designed and built at the Institute of Robotic and Intelligent Systems laboratory of the University of British Columbia.[3] It comprises two modules, each consisting of two links, one free to slew (revolute joint) and the other able to deploy and retrieve (prismatic joint). A general representation of the system is shown in Fig. 2. This manipulator presents the following attractive features compared to a system having revolute joints alone with the same number of degrees of freedom[3]: 1) reduced inertia coupling leading to simpler equations of motion, 2) improved capability to overcome obstacles, 3) simpler inverse-kinematics problem, and 4) smaller number of singular configurations.

2. *Conventional vs Intelligent Control*

In spite of the considerable progress that has been made in control theory, many industrial control problems are still solved through

Received 5 April 1999; revision received 7 February 2000; accepted for publication 9 June 2000.

*Graduate Fellow, Natural Sciences and Engineering Research Council of Canada, Department of Mechanical Engineering.

†Natural Sciences and Engineering Research Council Chair Professor of Industrial Automation, Department of Mechanical Engineering.

‡Professor Emeritus, Department of Mechanical Engineering. Fellow AIAA.

§Professor, Department of Mechanical Engineering. Associate Fellow AIAA.

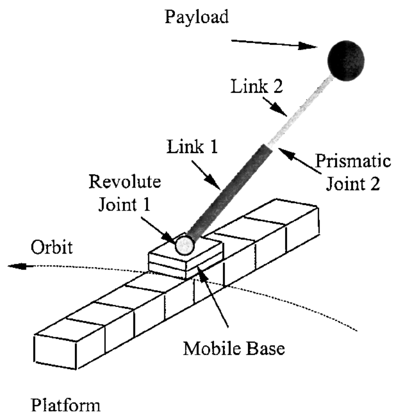


Fig. 1 Space-based deployable manipulator.

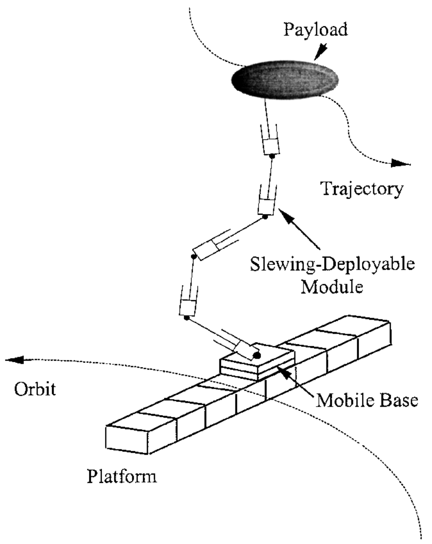


Fig. 2 Mobile flexible deployable manipulator with modules of slewing and deployable links.

empirical and heuristic approaches developed by experienced engineers. Intelligent control makes use of a knowledge base that is evolved through human experience or related means, for controlling those processes that cannot be effectively controlled by using conventional control techniques that are based on crisp algorithms. Given hereafter is a brief review of the transition from conventional control to intelligent control.

a. Conventional control. The expression conventional control refers here to those systems that utilize crisp control algorithms that are based on theories developed to control dynamic systems described by differential and difference equations. The term crisp is used here to mean nonfuzzy as is done routinely in the literature. There are several reasons why conventional controllers alone may be incapable of providing expected performance in complex plants:

1) A model-based controller may not give satisfactory results if the robot model is inaccurate, which will be the case generally because of a) parameter changes, b) unmodeled dynamics and time delays, c) nonlinearities, d) changes in operating point, and e) high order, particularly contributed by distributed inertia and flexibility.

2) The environment with which the robot interacts may not be completely predictable.

3) A conventional crisp algorithm, in general, cannot respond accurately to conditions that it cannot foresee or understand.

4) A crisp control algorithm will need a complete set of data, which may not be always available.

5) Sensor noise may make conventional controllers alone incapable of providing expected performance.

b. Adaptive control. When fixed-parameter feedback controllers do not give satisfactory results, then adaptive controllers may be used. In this case, the controller parameters are changed on line, based on error feedback and optimization of some performance index. Controllers can be designed to meet the performance requirements around an operating point where a linear model would be valid. Then, by using a scheduler, the controller can be made to meet the specifications over the whole operating range.[4] However, sometimes the operating range may have to be increased even further. The control system should be able to cope effectively with significant uncertainties in models of increasingly complex dynamic systems. It should be able to deal with significant unmodeled and unanticipated changes in the plant, in the environment, and in the control objectives. Under such circumstances, the effectiveness of conventional control will deteriorate and may necessitate intelligent control techniques for satisfactory control.

c. Intelligent control. An intelligent control system, in general, is able to acquire and represent knowledge about both the plant to be controlled and the environment in which the plant operates. It is able to 1) work with incomplete information, that is, incomplete model, incomplete rulebase, incomplete data, and so forth; 2) learn through experience; and 3) effectively operate in an unknown or unfamiliar environment, under parameter changes, disturbances, and so forth.

In other words, intelligent controllers are intended to replicate some of the human abilities such as adaptation and learning, planning under large uncertainty, and coping with large amounts of data.[5,6] They typically aim to attain higher degrees of autonomy. This has lead to functional architecture such as hierarchies. For instance, a supervisory layer can be added to a control system that uses discrete-event control. Such systems may be referred to as intelligent because the controller attempts to mimic high-level decision making processes of human operators.

Intelligent control builds on and enhances the conventional (crisp or nonfuzzy) control techniques to solve new challenging control problems. The research areas related to intelligent control include soft computing techniques such as neural networks, fuzzy logic, and genetic algorithms.[7] The three soft computing techniques mentioned in Table 1 can be directly utilized in intelligent control, either separately or synergistically.

d. Knowledge-based systems. The major advantage of a knowledge-based control procedure consists in its ability to use structured information available in nonnumeric form. It can, thus, deal with the accumulated knowledge in a set of linguistic statements. It is particularly attractive when a considerable amount of knowledge, expertise (specialized knowledge), and experience are available for controlling a particular plant, which may be incompletely known or too complex for accurate modeling.[6] One can, therefore, combine the advantages of a computer with some of the intelligent characteristics of a human for making inferences and decisions.

The main components of a knowledge-based system, as shown in Fig. 3, are a knowledge base, an inference engine, one or more databases, and a user interface. The knowledge base represents the available knowledge obtained from the linguistic description of human expertise in controlling a process. The inference engine tries to match the present data (context) with the condition part of the rules in the knowledge base (KB) to determine the control actions,

$$\text{Inference} = M[P(D), \text{KB}] \tag{1}$$

Table 1 Soft computing techniques

Example of application to robots	Strength
Neural networks (NN)	
NN-based controller taking the form of adaptive multilayer network; adaptable parameters are adjustable weights between nodes	With its learning and adaptation ability, NN is able to account for nonlinearities and uncertainties in robot dynamic model regardless of complexity
Fuzzy set theory	
Direct joint control; tuning of parameters of low-level controller	Easy implementation of expert knowledge, via fuzzy IF–THEN rules
Genetic algorithm	
Optimization and adaptive learning control of process plant	Derivative-free optimization and evolution through learning

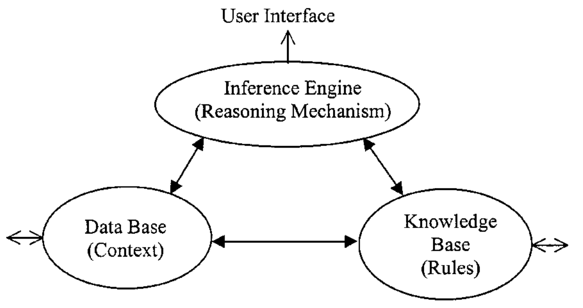


Fig. 3 Main components of a knowledge-based system.

where

P = preprocessing operator that converts the data to a form compatible with the KB, for example, a fuzzification operation in a fuzzy knowledge-based system
D = current data or information (context)
M = matching operator

The inference mechanism may also handle conflict resolution using methods such as first match, toughest match, privileged match, or most recent match.[6] There are many sources that lead to the development of a KB: human expertise, literature, experimentation, computer simulation, analysis, etc. Some areas of application of expert systems are design, interpretation, advising (legal, medical, etc.), forecasting, diagnosis, and prescription.

3. Fuzzy Logic and Fuzzy Control

Fuzzy logic allows the use of not-so-well-defined linguistic terms. Consider the following example. An integer is either odd or even. The classes odd and even are clearly defined. On the other hand, the class tall does not have such a clear (crisp) boundary. A person with a height of 1.65 m is not considered to be tall. A person with a height of 1.95 m will surely be considered tall. What about heights between 1.65 and 1.95 m? There is no universally accepted crisp boundary in this case. More closely related to engineering, an experienced operator could make the following statement about the behavior of a process: If the overshoot is big, then reduce the proportional gain moderately. Although the terms big and moderately do not have a crisp boundary, the operator's linguistic statement carries some knowledge. Zadeh,[8,9] the father of fuzzy logic, developed three decades ago a systematic framework to deal with fuzzy quantities such as tall and big. Fuzzy logic uses the concept of fuzzy sets that can model intermediate grades of belonging that occur in any concept.

Fuzzy controllers came into consideration at the beginning of the 1970s. They were inspired by the ability of human operators to interpret linguistic statements about a process under control and to reason for control actions, qualitatively, in the absence of precise and complete data as well as a mathematical model of the plant. Fuzzy control is a class of knowledge-based control procedures. It incorporates control knowledge in the form of a set of linguistic statements (rules) that may contain fuzzy quantities. Fuzzy logic control (FLC) has been applied to ship autopilots, subway cars, helicopters, pilot-scale steam engines, cement kilns, robotic manipulators, fish processing machines, household appliances, and many other systems.

C. Literature Review

In this section, to begin with, the literature relevant to the existing variable-geometry manipulator is briefly reviewed, followed by more important as well as appropriate contributions related to fuzzy logic and hierarchical control.

1. Variable Geometry Manipulator

Several studies have been conducted aimed at dynamics of structures with flexible deployable members. Lips and Modi[10,11] as well as Modi and Ibrahim[12] studied systems with deployable appendages directly connected to a central body. Similar studies, involving slewing and deployable appendages, were later conducted by Modi and Shen.[13]

A one-unit manipulator consisting of flexible revolute and prismatic joints, located on an orbiting flexible platform as shown in Fig. 1 was investigated by Marom and Modi.[14] The study of its planar dynamics and control showed significant coupling effects between the platform and the manipulator dynamics. A multimodule configuration was subsequently studied by Modi et al.,[15] as well as Hokamoto et al.[16] and Hokamoto and Modi.[17,18] Caron[19] presented an $\mathcal{O}(N)$ formulation for studying the planar dynamics and control of a multimodule manipulator consisting of several units, each having one revolute and one prismatic joint. The results showed interactions between flexibility, librational dynamics, and manipulator maneuvers. He also investigated the use of an FLT and linear quadratic regulator to control such manipulators. Chu[3] designed and constructed a variable-geometry manipulator prototype. He conducted simulation studies as well as experiments using the prototype, finding that significant coupling can exist among the base, joints, and payload vibrations.

2. Hierarchical and Knowledge-Based Control Using Fuzzy Logic

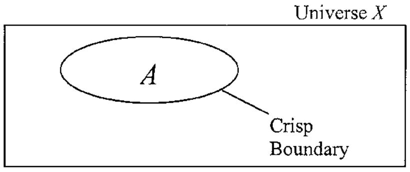
Zadeh[20] was the first to develop fuzzy logic as a tool for reasoning with imprecise concepts that arise in systems engineering. He proposed an approach for representing the vague reasoning of the human decision making process. The establishment of this theory has given birth to knowledge-based control and enriched the design of hierarchical control systems.

Mesarovic et al.[21] pioneered a theory for a hierarchical, multilevel system. They dealt with the levels of abstraction, complexity of decision making, and of priority of action. Zadeh[22] has discussed the role of fuzzy logic in the management of uncertainty in expert systems. He developed a systematic approach to the management of information that is imprecise or incomplete. Rasmussen[23] looked at the role of hierarchical knowledge representation in decision making and system management. This also includes the role of abstraction hierarchy in a supervisory control system and the extraction of significant information from high-resolution data. It was concluded that data abstraction is a necessary step in any scheme of hierarchical knowledge representation. De Silva and MacFarlane[24] developed a knowledge-based control structure for robotic manipulators. Their hierarchical structure has three levels, which integrate knowledge-based soft control with conventional hard control algorithms. The rulebase of the fuzzy controller consists of expert knowledge in the form of linguistic IF–THEN rules for servotuning. The main advantages of the proposed method are the ability of representing implicit knowledge of human experts and the flexibility of the control structure. Saridis[25] has developed an analytic formulation of the principle of increasing precision with decreasing intelligence for intelligent machines. De Silva[26] also investigated an analytic framework for knowledge-based tuning of servocontrollers. It involves a two-level control structure in which a conventional proportional–integral derivative (PID) servocontroller is tuned using fuzzy logic. It is demonstrated that a substantial reduction in computational effort is obtained through the analytical framework. De Silva[27] has discussed the information fuzziness and degree of resolution within the context of a control hierarchy and developed a criterion for decoupling a fuzzy KB.

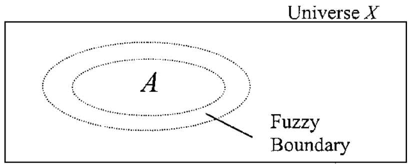
Barlev[28] successfully implemented an intelligent tuner for an ill-defined servomotor system. Wickramarachchi[29] developed a knowledge-based hierarchical control structure for process automation and concluded that, in the case of fish processing, its overall yield (recovery rate) could be increased.

III. Fuzzy Logic Control

This section focuses on the necessary theoretical background on fuzzy logic and its application to hierarchical systems. First, the concept of fuzzy sets is presented. Next, the mathematical tools such as the membership function and the compositional rule of inference are reviewed. Finally, the FLC is discussed in the context of hierarchical systems.



a) Crisp set A



b) Fuzzy set A

Fig. 4 Venn diagrams.

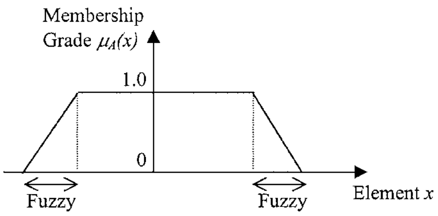


Fig. 5 Membership function of a fuzzy set A.

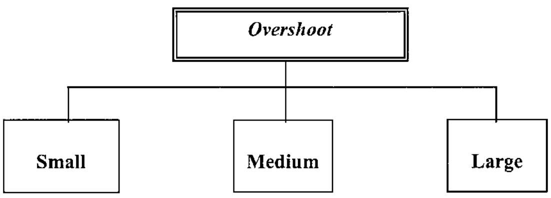


Fig. 6 Fuzzy states of the variable overshoot.

A. Fuzzy Sets and Membership Functions

The theory of fuzzy logic is based on the idea that the key elements in human thinking are not numbers but labels of fuzzy sets. A fuzzy set is a collection of elements that has no crisp boundaries. The elements belonging to a fuzzy boundary region neither completely belong to the fuzzy set nor are completely excluded from it. A fuzzy set is a tool for modeling intermediate grades of membership that naturally occur in any concept. The Venn diagrams of a crisp set and a fuzzy set are shown in Figs. 4a and 4b, respectively.[6]

A fuzzy set A can be represented mathematically by a membership function μ_A, which gives the degree of membership within the set of any element of the universe of discourse. The membership function μ_A maps the universe of discourse X to the interval [0, 1],

$$\mu_A(x)\colon X \to [0, 1] \tag{2}$$

A membership value of 1 states that the element considered is definitely in the set. On the other hand, a membership value of 0 states that the element considered is definitely outside the set. A typical membership function is shown in Fig. 5. A fuzzy set A in X may also be represented in a way that each element is paired with its corresponding grade of membership. For a discrete universe of discourse,

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \cdots + \frac{\mu_A(x_i)}{x_i} + \cdots = \sum_{x_j \in X} \frac{\mu_A(x_i)}{x_i} \tag{3}$$

Fuzzy sets are used to represent fuzzy states of a particular fuzzy variable. For example, consider the fuzzy variable overshoot pertaining to a step response. This variable could have three fuzzy states such as small, medium, and large (Fig. 6). Each fuzzy state is itself a fuzzy set having its own membership function (Fig. 7).

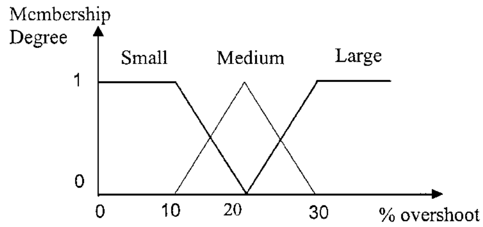


Fig. 7 Membership functions of the fuzzy states.

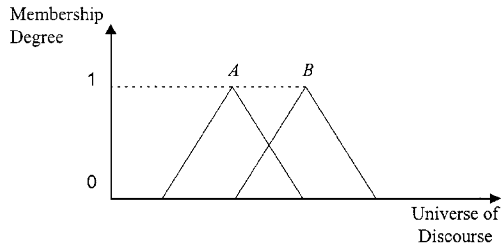


Fig. 8 Fuzzy sets A and B.

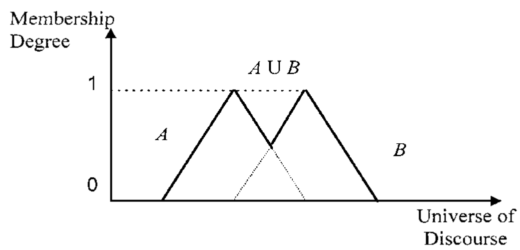


Fig. 9 Union of fuzzy sets A and B.

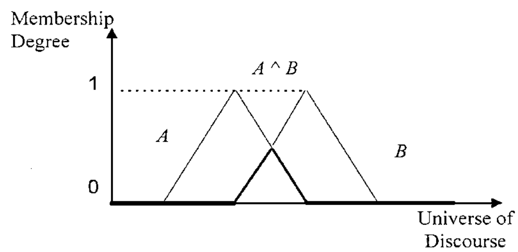


Fig. 10 Intersection of fuzzy sets A and B.

The fuzzy resolution represents the number of fuzzy states that a fuzzy variable can have. For the example just considered, the resolution is 3.

B. Fuzzy Logic Operations and Definitions

Consider two fuzzy sets A and B in the same universe of discourse (Fig. 8).

1. *Union (OR, ∪)*

The union of two fuzzy sets A and B in the same universe of discourse (Fig. 9) has a membership function given by

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \qquad \forall x \in X \tag{4}$$

2. *Intersection (AND, ∩)*

The intersection of two fuzzy sets A and B in the same universe of discourse (Fig. 10) has a membership function given by

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, \qquad \forall x \in X \tag{5}$$

C. Compositional Rule of Inference

In fuzzy control, the knowledge is embedded in the KB, which is a set of linguistic statements (rules) containing fuzzy quantities and employing the fuzzy implication (IF–THEN). An example of such a rule, related to the step-input response of a system, is IF rise time is too slow THEN proportional gain change is positive big.

Let the fuzzy sets R, D, and C denote the fuzzy relation that represents the collections of linguistic rules, the context data, and the control action, respectively. The membership function of the control action (fuzzy set C) is determined by matching the membership functions of the context data (fuzzy set D) with the KB (fuzzy rule base R):

$$\mu_C = \sup_Y \min(\mu_D, \mu_R) \tag{6}$$

Note that Y represents the space in which the data D are defined. The inference making procedure given by Eq. (6) is known as the compositional rule of inference, and clearly it is a special case of Eq. (1).

D. FLC

The basic steps in FLC may be as follows[6]:

1. *Offline*

1) Acquire and represent the control knowledge as a set of linguistic rules that may contain fuzzy quantities.

2) Develop a set of discrete membership functions for the process output variables and control input variables.

3) By the use of a fuzzy implication for each linguistic rule with the corresponding membership functions, obtain the multidimensional array R_i of membership values for that particular rule.

4) Get the fuzzy rule base R by combining the arrays R_i using fuzzy operations (AND, OR, and NOT).

2. *Online*

5) Fuzzify the measured process output variables as fuzzy singletons (fuzzy sets whose support is a single point).

6) Match the fuzzy measurements of step 5 with the fuzzy rule base R using the compositional rule of inference to get the fuzzy control inference.

7) Defuzzify the control inference.

Note that an alternative could be to construct a crisp decision table offline to reduce the online computation effort.

3. *Defuzzification*

Defuzzification is the process in which a representative crisp parameter is obtained for a given fuzzy set (membership function). One important method of defuzzification is the centroid method. Let us first define what a support set is. The support of a fuzzy set A is the crisp set formed by the collection of all elements $x_i \in X$ such that $\mu_A(x_i) > 0$. Now let the membership function of a control inference be given by $\mu_c(c)$, with a support set S. Then, using the centroid method, the crisp control action c' is given by, for discrete membership functions,

$$c' = \frac{\sum_{c_i \in S} c_i \mu_C(c_i)}{\sum_{c_i \in S} \mu_C(c_i)} \tag{7}$$

4. *Fuzzy Control Architectures*

FLC commonly has two different architectures: low-level direct control and high-level tuning control.

a. Low-level FLC. A large majority of the applications of fuzzy logic have been for low-level direct control where the fuzzy logic controller replaces the conventional controller. The control input is typically proportional to error and change in error of the output. However, when controlling fast processes, direct FLC often fails due to the inherent delay and imprecision introduced at the lowest level. This is because of the following:

1) Signal preprocessing, reasoning, and decision making time can be high.

2) Low-level sensory data can be noisy.

b. High-level FLC. It is intuitively more appealing and justifiable to use fuzzy logic at a higher level where the human operators have accumulated control knowledge. It has several attractive features:

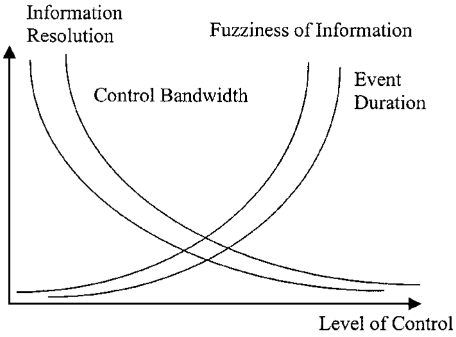


Fig. 11 Variation of information resolution and fuzziness with the level of hierarchy.

1) Speed of low-level (direct) control can be maintained high using conventional crisp techniques.

2) It improves accuracy of the low-level conventional controller.

3) Experience-based tuning protocols are inherently fuzzy in general, and hard control algorithms cannot use the knowledge contained by linguistic statements of human experts.

4) Human intervention in the control loop is supervisory.

This architecture leads naturally to a hierarchical structure in which the fuzzy controller performs system monitoring as well as other tasks such as tuning, adaptation, self-organization, and dynamic restructuring. Stability is an important consideration in any control system. For fuzzy control systems, one established approach for stability analysis is not available. Typically, in low-level FLC, the process and the controller are approximated by a nonlinear model and the conventional stability studies are carried out. In high-level FLC, stability of the fuzzy decision making process itself is taken into consideration in analyzing stability.[6] The topic of stability analysis is not addressed in the present paper.

E. Hierarchical Control

Hierarchical control systems were first introduced as a means of coping with the complexity of large and difficult control tasks.[21] They are particularly relevant when various control functions in the system can be conveniently ranked into different levels depending on their functionality. Some considerations for defining the levels of control are 1) the nature of tasks, 2) information resolution (decreases as level of control increases, Fig. 11), 3) fuzziness of information (increases as level of control increases), 4) event duration (increases as level of control increases), and 5) control bandwidth or control speed (decreases as level of control increases).

IV. Control System Development

With this as background for the fuzzy logic and hierarchical control, a practical example is presented now to illustrate its application to a novel space-based manipulator with a slewing link that carries a deployable member (Fig. 1). The two-link manipulator, mounted on a mobile base, is supported by a platform in an arbitrary orbit. The platform, robot links, and the revolute joint are treated as flexible. The objective is to design a knowledge-based hierarchical control for the system.

The equations governing dynamics of the system are highly nonlinear, nonautonomous, coupled, and can be expressed in the general form[30]

$$\boldsymbol{M}(\boldsymbol{q}, t)\ddot{\boldsymbol{q}} + \boldsymbol{F}(\dot{\boldsymbol{q}}, \boldsymbol{q}, t) = \boldsymbol{Q}(\dot{\boldsymbol{q}}, \boldsymbol{q}, t) \tag{8}$$

where $\boldsymbol{M}$ is the system mass matrix; $\boldsymbol{q}$ is the vector of generalized coordinates; $\boldsymbol{F}$ contains the terms associated with the gravitational, centrifugal, Coriolis, elastic, and internal dissipative forces; and $\boldsymbol{Q}$ represents generalized forces, including the control inputs. Equation (8) describes the inverse dynamics[1] of the system. For numerical simulation purposes, forward dynamics[1] of the system is required, and, therefore, Eq. (8) must be solved for $\mathrm{d}^2\boldsymbol{q}/\mathrm{d}t^2$:

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{Q} - \boldsymbol{F}) \tag{9}$$

For a system composed of N bodies, the solution of these equations of motion generally requires order (N^3) arithmetic operations.

To reduce the computational effort, the development of an order N formulation has been the focus of several studies. The method used here, based on the Lagrangian approach and involving velocity transformations, was developed by Caron[19] and Caron et al.[30] The formulation developed by Caron[19] and Caron et al.[30] is valid for serial manipulators consisting of an arbitrary number N of flexible, slewing, and deployable units. The formulation allows variation of geometric, inertia, and stiffness characteristics along the manipulator. The model accounts for flexibility and dissipation at the revolute joints. The coupling effects between the orbital, librational, slew, deployment, and vibrational degrees of freedom, associated with the platform and manipulator, are also taken into consideration.

A. Hierarchical Structure

An important aspect of the hierarchical structure developed in this paper is the combination of the advantages of a crisp high-bandwidth controller with those of a soft knowledge-based supervisory controller. Note that the focus of the present paper is fuzzy logic based hierarchical control. The system presented here contains conventional control in the lowest layer. Decision making based on fuzzy logic occupies upper layers. Accordingly, fuzzy logic is not used in the present application as a substitute for conventional control but rather as a complement. The structure can be decomposed into three main layers.

1. *First Layer*

The first (bottom) layer deals with high-resolution information such as the data coming from the sensors (optical encoders) attached to the closed-loop servomotors. This type of information is characterized by a large amount of individual data points, collected and produced at high frequency, that are precise and of high resolution. The servomotor at each joint of the manipulator is in closed-loop control. In the present case, a controller based on the FLT is designed to regulate the manipulator and platform rigid dynamics. This technique decouples the system, linearizes it, and uses a PD feedback loop to achieve the desired dynamic behavior. The FLT-based controller is designed using the rigid bodies model but is used to control the original, fully flexible system.

The crisp high-bandwidth controller used is of the PD type, whose control action is given by

$$C(t) = \mathrm{KP}\cdot e(t) + \mathrm{KD}\cdot \dot{e}(t) \tag{10}$$

where $e(t)$ is the feedback error signal, and KP and KD are proportional and derivative gains, respectively.

2. *Second Layer*

The data processing for monitoring and evaluation of the system performance occurs in this intermediate layer. Here, high-resolution, crisp data from sensors are filtered to allow an abstract representation of the current state of the manipulator. This operation is similar to interpretation of performance of a servosystem through direct observation by a human expert. This servo-expert layer acts as an interface between the crisp conventional controller that controls the servomotors at the bottom layer and the knowledge-based controller that tunes the controller at the top layer. The intermediate layer handles performance specifications, response processing, and computation of the performance indices. This stage involves, for example, averaging or filtering of the data points and computation of the rise time, overshoot, and steady-state offset.

3. *Top Layer*

The KB and the inference engine are used at the top layer to make decisions that will inform as to what changes would be needed in the system to achieve the overall control objective. This layer can serve such functions as monitoring the performance of the overall system, assessment of the quality of operation, tuning of the low-level controllers, and general supervisory control. In this layer, there is a high degree of information fuzziness and a relatively low control bandwidth. Figure 12 presents the hierarchical structure of the three-level control system. What is presented in the rest of the present section are the techniques and procedures that are used in the top layer in its knowledge-based decision making activities.

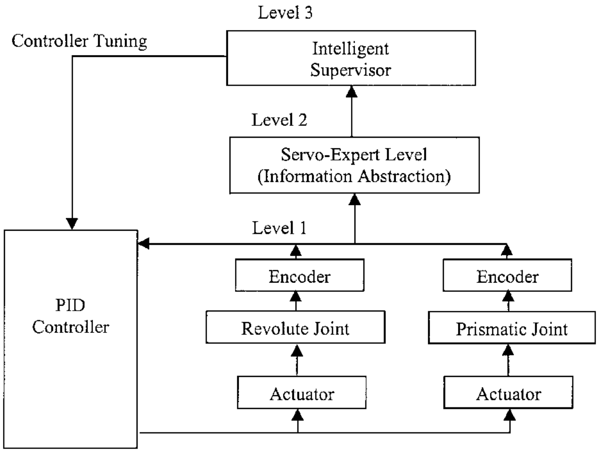


Fig. 12 Block diagram of the control system.

B. Performance Specification

The desired performance of the system is specified in terms of the parameters of a second-order reference model. Another way of specifying the performance would be to use time-domain performance parameters. Although the reference model implicitly represents the desired performance of the system, it is not an actual model of the system. The use of a reference model instead of the direct specification of time-domain parameters is preferred because it provides some buffer against unrealistic requirements. The reference model is expressed in terms of a damping ratio ζ, an undamped natural frequency ω_n, and a fixed offset G_d as

$$G_r(s) = \omega_n^2 \big/ \left(s^2 + 2\zeta\omega_n s + \omega_n^2\right) + G_d \tag{11}$$

From this model, the time response $y(t)$ of the system to a unit step input can be analytically obtained as

$$y(t) = 1 - \left[1\big/\left(\sqrt{1-\zeta^2}\right)\right] e^{-\zeta\omega_n t} \sin\left[\omega_n\sqrt{1-\zeta^2}\,t + \tan^{-1}\left(\sqrt{1-\zeta^2}/\zeta\right)\right] + e \tag{12}$$

where e represents the offset. Now, the following time-domain parameters are extracted based on the reference model, with the subscript M referring to the model with 1) rise time (RST_M), 2) overshoot, if underdamped (OVS_M), and 3) offset at steady state (OFS_M).

Thus, there are three parameters, RST_M, OVS_M, and OFS_M, representing the desired performance of the system. RST_M is chosen as the time it takes for the response to reach 95% of the desired steady-state response. OVS_M is calculated at the first peak of the response as a percentage of the desired steady-state response. The percentage steady-state offset is computed by taking the difference between the average of the last third of the response and the desired steady-state response.

The corresponding time-domain parameters are obtained from the response of the actual system (servomotor), with the subscript S referring to the servomotor as RST_S, OVS_S, and OFS_S. Once evaluated, the parameters of the model are compared with those of the real servomotors to get the index of deviation.

C. Performance Evaluation and Classification

For each performance attribute, an index of deviation is calculated using the following equation:

$$\text{Index of deviation of } i\text{th attribute} = 1 - \frac{i\text{th attribute of model}}{i\text{th attribute of servomotor}} \tag{13}$$

The index is defined in such a way that a value of 1 corresponds to the worst-case performance, whereas a value of 0 means the actual performance of the servomotor, for that particular attribute, exactly meets the specification.

Table 2 Mapping from the index of deviation to discrete performance indices

Index of deviation	Performance indices $K(i)$
$\mathrm{ERR}(i) < 0$	5
$0 \leq \mathrm{ERR}(i) \leq \mathrm{TH}(1)$	4
$\mathrm{TH}(1) \leq \mathrm{ERR}(i) \leq \mathrm{TH}(2)$	3
$\mathrm{TH}(2) \leq \mathrm{ERR}(i) \leq \mathrm{TH}(3)$	2
$\mathrm{TH}(3) \leq \mathrm{ERR}(i) \leq 1$	1

Table 3 Heuristics of PD tuning

Context	Action for performance improvement	
	KP	KD
RST	Increase	Decrease
OVS	Decrease	Increase
OFS	Increase	No change

The indices are calculated according to

$$\mathrm{RST}_{\mathrm{id}} = 1 - \mathrm{RST}_M/\mathrm{RST}_S = \mathrm{ERR}(1) \tag{14}$$

$$\mathrm{OVS}_{\mathrm{id}} = 1 - \mathrm{OVS}_M/\mathrm{OVS}_S = \mathrm{ERR}(2) \tag{15}$$

$$\mathrm{OFS}_{\mathrm{id}} = 1 - \mathrm{OFS}_M/\mathrm{OFS}_S = \mathrm{ERR}(3) \tag{16}$$

These indices represent the performance condition of the servomotors and, hence, should correspond to the context of the rulebase of system tuning. The index of deviation must, therefore, be fuzzified into membership values corresponding to the five selected primary fuzzy sets: highly unsatisfactory (HIUN), needs improvement (NDIM), acceptable (ACCP), in specification (INSP), and overspecification (OVSP).

To obtain a discrete set of performance indices $K(i)$, threshold values are defined for each index of deviation over the interval $-\infty$ to 1 as shown in Table 2. The performance indices obtained in this manner are the input to a precalculated decision table of tuning the PD parameters of a servomotor. The outputs from this decision table are the tuning actions that are used to update the controller parameters.

D. Fuzzy Tuner Layer

At the highest level of the hierarchical structure there is a KB for tuning a servocontroller. This tuning knowledge of human experts is expressed as linguistic rules using fuzzy terms. For each status of the system, a conceptual abstraction is computed, and the expert knowledge is transformed into a mathematical form by the use of fuzzy set theory and fuzzy logic operations. The end result is a decision table, calculated offline, that gives the tuning actions corresponding to the performance indices that represent the status of the process.

To obtain the decision table, one must first assign membership functions to each of the performance indices and tuning variables. Then, by applying the compositional rule of inference for each possible context parameter, fuzzy composite relation tables are computed that express the relations between the performance index and tuning variable. The last step consists of defuzzifying these composite tables to obtain crisp tuning actions corresponding to numerical context values of the system condition.

E. Fuzzy Linguistic Rules

The expert tuning knowledge for a simple PD controller in parallel form, that is, $G_c(s) = \mathrm{KP} + \mathrm{KD}s$, may utilize heuristics such as the ones given in Table 3. One may define the primary fuzzy sets for the performance indices for each context variable (i.e., RST, OVS, and OFS) as shown in Table 4.

The fuzzy tuning variables are defined as follows:

1) DKP is the change in proportional gain.

2) DKD is the change in derivative gain.

Table 4 Fuzzy labels of performance indices

Performance indices	Context fuzzy sets	
	Abbreviation	Fuzzy value
1	HIUN	Highly unsatisfactory
2	NDIM	Needs improvement
3	ACCP	Acceptable
4	INSP	In specification
5	OVSP	Overspecification

Table 5 Tuning fuzzy sets and representative numerical values

Tuning fuzzy sets		
Abbreviation	Fuzzy value	Integer value
PL	Positive large	3
PM	Positive moderate	2
PS	Positive small	1
ZR	Zero	0
NS	Negative small	−1
NM	Negative moderate	−2
NL	Negative large	−3

	If *RST* is *HIUN*	then *DKP=PL, DKD=NS,*
or	if *RST* is *NDIM*	then *DKP=PM, DKD=NS,*
or	if *RST* is *ACCP*	then *DKP=ZR, DKD=ZR,*
or	if *RST* is *INSP*	then *DKP=ZR, DKD=ZR,*
or	if *RST* is *OVSP*	then *DKP=NS, DKD=PS.*
	If *OVS* is *HIUN*	then *DKP=NL, DKD=PL,*
or	if *OVS* is *NDIM*	then *DKP=NM,DKD=PM,*
or	if *OVS* is *ACCP*	then *DKP=NS, DKD=ZR,*
or	if *OVS* is *INSP*	then *DKP=ZR, DKD=ZR,*
or	if *OVS* is *OVSP*	then *DKP=ZR, DKD=NS.*
	If *OFS* is *HIUN*	then *DKP=PM, DKD=ZR,*
or	if *OFS* is *NDIM*	then *DKP=PS, DKD=ZR,*
or	if *OFS* is *ACCP*	then *DKP=ZR, DKD=ZR,*
or	if *OFS* is *INSP*	then *DKP=ZR, DKD=ZR,*
or	if *OFS* is *OVSP*	then *DKP=NS, DKD=ZR.*

Fig. 13 Rulebase for servomotor tuning.

Each tuning variable may be expressed with the fuzzy sets and representative numerical values that are listed in Table 5. The rulebase for control parameter tuning is given in Fig. 13.

F. Decision Table Development

In developing the decision table for tuning, it is necessary to establish the membership functions for the fuzzy performance attributes RST, OVS, and OFS as well as for the fuzzy tuning actions DKP and DKD. The resolution of each fuzzy quantity is the number of fuzzy values a condition variable or an action variable can take. In other words, the number of discretization levels equals the number of possible fuzzy values that a variable may take. For a fine tuning action, a high resolution would be desired. The proper choice of fuzzy resolution would provide a compromise between the precision of tuning and computer memory storage requirements. The discrete representative values corresponding to the fuzzy quantities are normalized values; they represent a relative weight and can be scaled. The membership functions depend on the knowledge of the problem. In the case of high fuzziness, more robust membership functions with sufficient overlap between adjoining fuzzy states may be desired. Each fuzzy action or condition quantity has a representative value that is assigned a membership degree equal to unity. The decreasing membership degree around that reresentative value intorduces a degree of fuzziness as indicated in the Tables 6 and 7.

Table 6 Membership functions for the condition variables

Fuzzy set	RST, OVS, OFS 1	2	3	4	5
HIUN	1	0.6	0.2	0	0
NDIM	0.6	1	0.6	0.2	0
ACCP	0.2	0.6	1	0.6	0.2
INSP	0	0.2	0.6	1	0.6
OVSP	0	0	0.2	0.6	1

Table 7 Membership functions for the action variables

Fuzzy set	DKP, DKD −3	−2	−1	0	1	2	3
NL	1	0.9	0.8	0.7	0.6	0.5	0.4
NM	0.9	1	0.9	0.8	0.7	0.6	0.5
NS	0.8	0.9	1	0.9	0.8	0.7	0.6
ZR	0.7	0.8	0.9	1	0.9	0.8	0.7
PS	0.6	0.7	0.8	0.9	1	0.9	0.8
PM	0.5	0.6	0.7	0.8	0.9	1	0.9
PL	0.4	0.5	0.6	0.7	0.8	0.9	1

Table 8 Decision table for servo parameter tuning

Condition	Tuning action DKP	DKD
RST = HIUN	0.4	−0.2
RST = NDIM	0.3	−0.2
RST = ACCP	0	0
RST = INSP	0	0
RST = OVSP	−0.2	0.2
OVS = HIUN	−0.4	0.4
OVS = NDIM	−0.3	0.3
OVS = ACCP	−0.2	0
OVS = INSP	0	0
OVS = OVSP	0	−0.2
OFS = HIUN	0.3	0
OFS = NDIM	0.2	0
OFS = ACCP	0	0
OFS = INSP	0	0
OFS = OVSP	−0.2	0

A decision table is obtained offline, using the membership functions and the appropriate mathematical tools (Table 8).[6] Thus, for each condition-variable fuzzy value, we have a crisp inference or tuning action. These tuning actions may be scaled by a sensitivity factor before being used.

V. Results and Discussion

A pick-and-place operation was simulated using the hierarchical control structure described in the preceding section. The platform, revolute joint, and manipulator links are flexible. The FORTRAN program developed by Caron[30] was modified, and files representing the higher layers of the control hierarchy were added. The acceleration vector $d^2\boldsymbol{q}/dt^2$ was integrated numerically using Gear's method, which is well suited for stiff systems of ordinary differential equations. The numerical data used in the analysis are summarized as follows: The orbit is circular at an altitude of 400 km with a period of 92.5 min. The geometry of the platform is cylindrical, with axial to transverse inertia ratio of 0.005; mass, 120,000 kg; length, 120 m; and flexural rigidity $EI_p = 5.5 \times 10^8$ Nm^2. The manipulator revolute joint mass is 20 kg; moment of inertia, 10 kg m^2; and stiffness $K = 1 \times 10^4$ Nm/rad. The manipulator links (slewing and deployable) have a cylindrical geometry with axial-to-transverse inertia ratio of 0.005; mass, 200 kg; length, 7.5 m; and flexural rigidity EI_s and $EI_d = 5.5 \times 10^5$ Nm^2. The payload ratio is equal to two, that is, the mass of the payload is twice the mass of the unit (slewing plus deploying links), 800 kg. In the results presented here, the longitudinal elastic deformation of the bodies is neglected, as well as the dynamics of the mobile base. The platform is initially oriented perpendicular to the local vertical, that is, in an unstable equilibrium position.

Table 9 Desired initial, intermediate, and final positions of the robot

Desired values	Pick position	Way point	Place position
α, deg	0	36	40
d, m	0	6.57	7.3
End effector x, m	7.5	11.4	11.3
End effector y, m	1.5	9.8	11.0

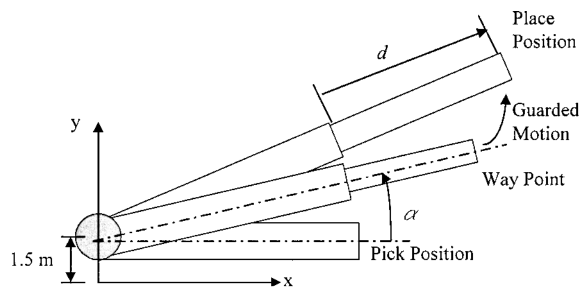


Fig. 14 Pick-and-place operation.

The desired position p in the joint space is generated by a sine-on-ramp profile [Eq. (17)]. It assures zero velocity as well as zero acceleration at the beginning and end of the maneuver:

$$p(\tau) = \frac{\Delta p}{\Delta\tau}\left[\tau - \frac{\Delta\tau}{2\pi}\sin\left(\frac{2\pi}{\Delta\tau}\tau\right)\right] \tag{17}$$

Here τ represents time and $\Delta\tau$ is the time allowed for the maneuver of amplitude Δp to take place.

A pick-and-place operation is a common task carried out by manipulators.[1] It consists of picking an object and bringing it to a desired position, within a specified time interval, with desired initial and final velocities and accelerations. Here, the manipulator is fully retracted in the pick position, parallel to the platform length, and is almost fully extended at the place position (Fig. 14). It stops at a way point, which is an intermediate position from which the guarded motion starts. At this position, further tuning of the controller will take place, if required. It is highly desirable that the motion from the way point to the final position be smooth (guarded motion). The initial, intermediate, and final positions are specified in the joint and Cartesian spaces as given in Table 9, d representing the length of the deployable link.

From the nature of the model, the traditional Ziegler–Nichols[31] method cannot be used to find a set of initial control gains. Moreover, the payload may not always have the same mass. Hence, the starting PD control gains were selected based on experience. Some tuning of the two joints should take place before the tracking starts at the pick position. The tuning consists of analyzing the response to a step input of 5 cm for the prismatic joint and of 5 deg for the revolute joint, with respect to the performance requirements of RST_M, OVS_M, and OFS_M as defined in Sec. IV.B. The gains are updated accordingly.

The manipulator is allowed 80 s to reach the way point, where it comes to a complete stop. As mentioned before, a sinusoid-on-ramp displacement profile is used for point interpolation in the joint space, that is, for α and d. The algorithm then decides if some further tuning is needed for the revolute or prismatic joint. Note that the control gains for the prismatic joint do not depend on the operation point. If the average error in the joint space during tracking exceeds a specified limit (0.5 deg, 3 cm), or if the maximum error is greater than a certain threshold (1 deg, 7 cm), extra tuning is judged necessary. The manipulator is then given 50 s to reach the place position. If, during a tuning sequence, the gains reach some steady-state values, the tuning sequence stops, but will be automatically resumed if the state becomes transient.

A. Response Without Tuning

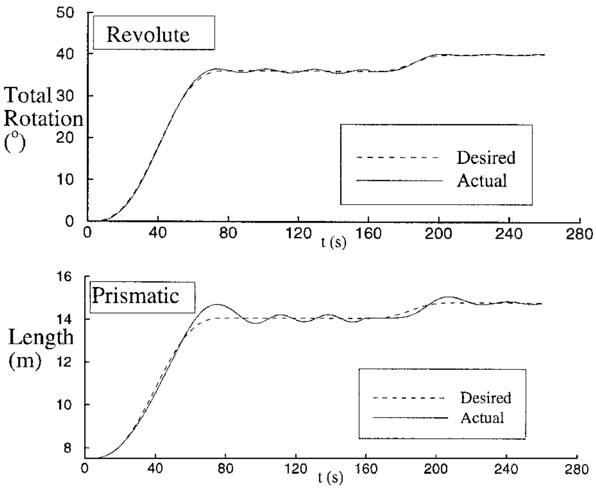
First, the manipulator is made to conduct a pick-and-place operation without any initial tuning. These gains may not be optimal over a wide range of the workspace, especially if the mass of the payload varies from one task to another. The time response in the joint space is shown in Fig. 15a, and the end-effector path in the Cartesian space is presented in Fig. 15b. The performance can be judged as poor and unacceptable, depending on the intended application. The maximum error in the Cartesian space for the guarded motion exceeds 28 cm and the average error 14 cm (Tables 10–12).

Table 10 Errors in the Cartesian and joint spaces without tuning

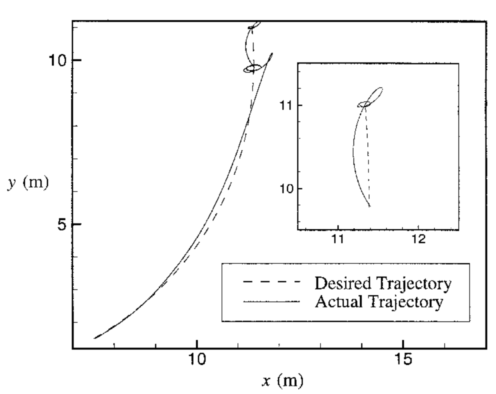
Parameter	Fast motion	Guarded motion
e_{av}, mm	217	149
$e_{\max}$, mm	661	289
Average error d, mm	201	133
Maximum error d, mm	647	289
Average error α, deg	$1.6E-4$	$2.5E-4$
Maximum error α, deg	$5.4E-4$	$5.2E-4$

Table 11 Errors in the Cartesian and joint spaces with initial tuning

Parameter	Fast motion	Guarded motion
e_{av}, mm	127	119
$e_{\max}$, mm	360	196
Average error d, mm	106	105
Maximum error d, mm	315	177
Average error α, deg	$5.5E-4$	$2.5E-4$
Maximum error α, deg	$6.7E-3$	$5.1E-4$



a) Slew and deployment time histories

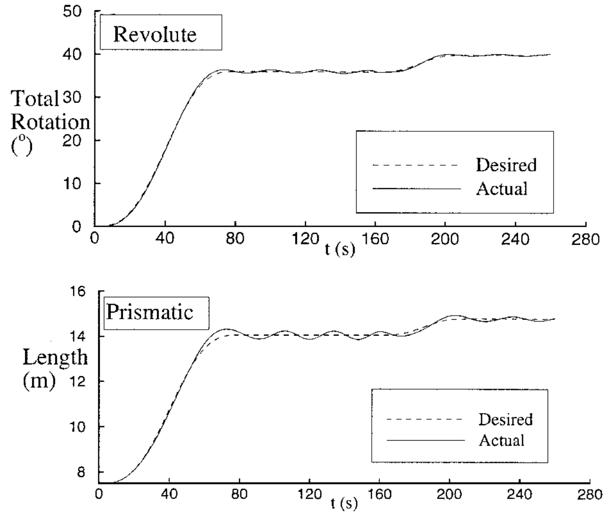


b) End-effector path in Cartesian space

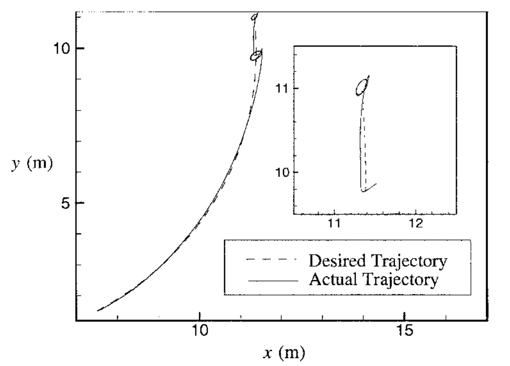
Fig. 15 Pick-and-place operation in absence of tuning.

Table 12 Errors in the Cartesian and joint spaces with autotuning

Parameter	Fast motion	Guarded motion
e_{av}, mm	127	91
$e_{\max}$, mm	360	171
Average error d, mm	106	70
Maximum error d, mm	315	171
Average error α, deg	$5.5E-4$	$2.5E-4$
Maximum error α, deg	$6.7E-3$	$5.2E-4$



a) Slew and deployment time histories



b) End-effector path in Cartesian space

Fig. 16 Pick-and-place operation with initial tuning.

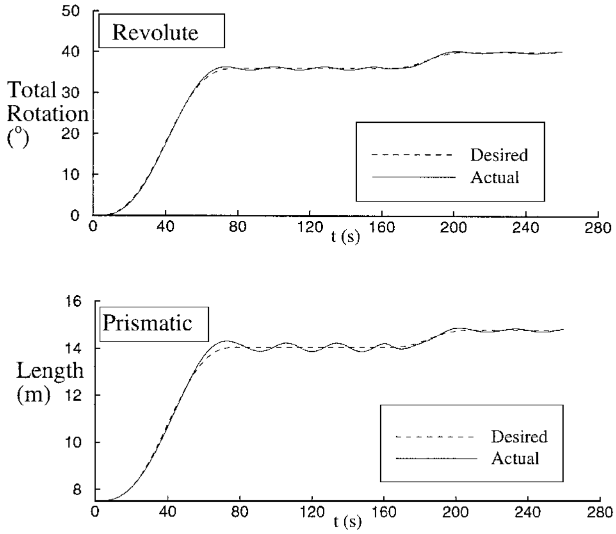
The errors in the joint space show that the prismatic joint is responsible for the poor performance rather than the slewing joint. When extra tuning is allowed at the way point, the software indeed decides to tune only the prismatic joint, as demonstrated later.

B. Response with Initial Tuning

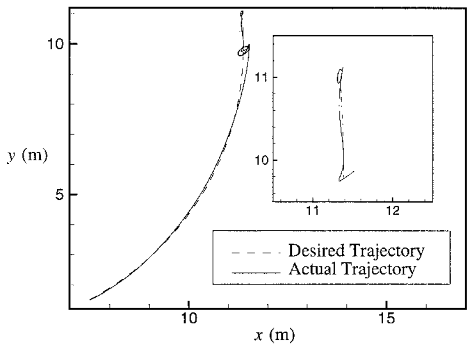
Next, the control parameters are tuned before the start of the operation. The response in the joint space is shown in Fig. 16a, and the end effector path in the Cartesian space is indicated in Fig. 16b. In this case the maximum and average errors for the guarded motion are about 19 and 11 cm, respectively (Table 11). Although the performance has improved, it may still be considered unacceptable if high precision is needed.

C. Response with Autotuning

In this case, the manipulator is controlled in the automatic tuning mode. That is, depending on the performance during the first part of the trajectory, some tuning may be done at the way point. In this case, tuning of the prismatic joint was required at the way point. Results, in the joint and the Cartesian spaces, are shown in Fig. 17. The guarded motion, from the intermediate way point to the place position, is smoother, and, thus, the performance is more satisfactory. The maximum and average errors for the guarded motion are



a) Slew and deployment time histories



b) End-effector path in Cartesian space

Fig. 17 Pick-and-place operation with autotuning.

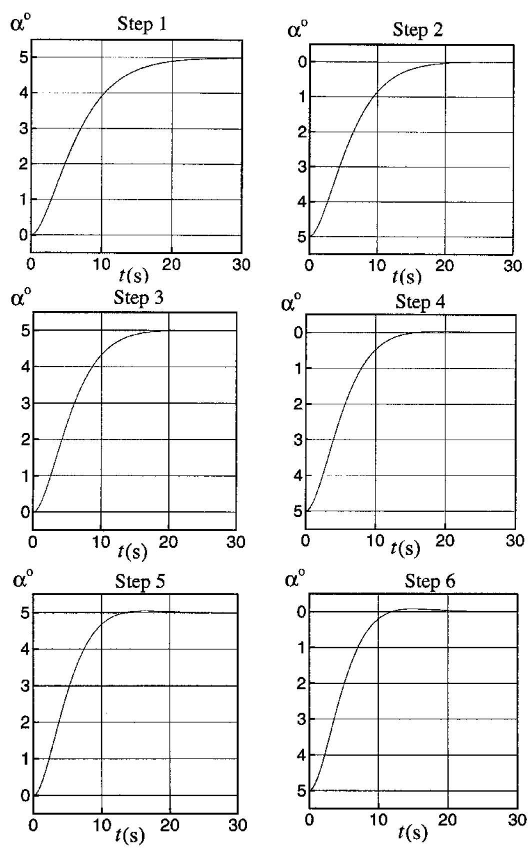


Fig. 18 Step responses at the pick position, with payload, during tuning of the revolute joint.

about 17 and 9 cm, respectively (Table 12). Autotuning seems to particularly help in the case of guarded motion, because more time is available for tuning in this case. Results show that some additional tuning helps to improve the tracking abilities of the manipulator.

The tuning of the controller parameters involves the analysis of the response of the joints to a series of step inputs. The response of the revolute joint to a sequence of six step inputs at the pick position is shown in Fig. 18. The response changes toward the desired one, as specified by the reference model. The evolution of the context variables and the PID gains is shown in Figs. 19 and 20, respectively. In the case of the revolute joint, the context value for the rise time increases from 1 to 2 whereas the context value for the overshoot is decreased from 5 to 4. Based on context values, the proportional gain is increased whereas the derivative gain is decreased. In the

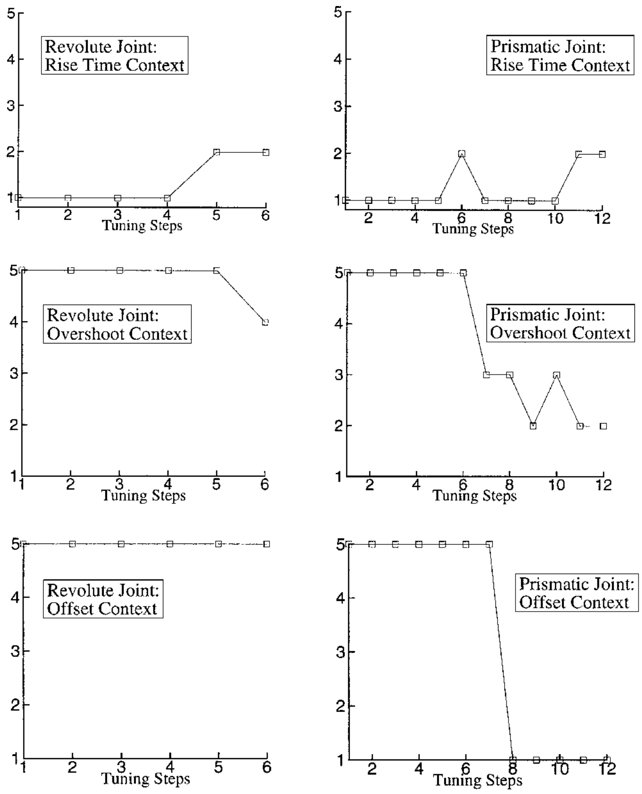


Fig. 19 Evolution of the context variables after each tuning step.

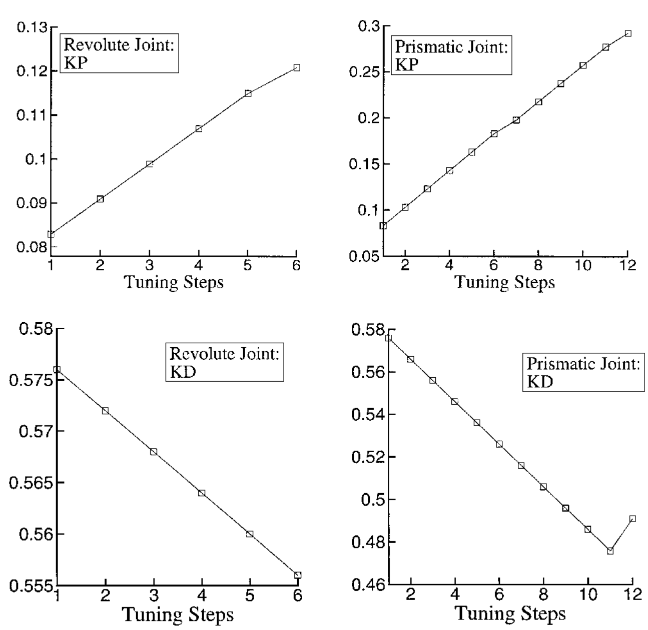


Fig. 20 Evolution of the PD gains after each tuning step.

case of the prismatic joint, the context value for the rise time also increases from 1 to 2 whereas that for the overshoot decreases from 5 to 2. The context value for the offset reduces because the steady state is not quite reached. Hence, the proportional gain is increased whereas the derivative gain is decreased.

VI. Concluding Remarks

The paper explains important aspects of the knowledge-based hierarchical control procedure and demonstrates its application to a flexible manipulator, with a deployable link, supported by an elastic platform. Based on the simulation results, the following general conclusions can be made:

1) Initial tuning of the parameters of a conventional low-level controller, as well as further tuning during operation, can improve the performance of the manipulator. The results have shown that the accuracy of positioning has been increased.

2) Fuzzy logic is useful in acquiring and representing the tuning knowledge available from experienced operators, as well as in making inferences based on the status of the plant. It is, therefore, an important step in the development of autonomous systems that are suitable in space-based applications.

3) A hierarchical control structure allows the advantageous combination of a conventional high-bandwidth and a knowledge-based low-bandwidth controller.

References

[1]Sciavicco, L., and Siciliano, B., *Modeling and Control of Robot Manipulators*, edited by S. Director, McGraw–Hill Series in Electrical and Computer Engineering, McGraw–Hill, New York, 1996, pp. 161, 162, and 171–175.

[2]Pedrycz, W., *Fuzzy Control and Fuzzy Systems*, etited by M. J. H. Sterling, Electronic and Electrical Engineering Research Studies: Control Theory and Applications Series, Research Studies Press, Somerset, England, U.K., 1989, pp. 61–67.

[3]Chu, M., "Design, Construction and Operation of a Variable-Geometry Manipulator," M.S. Thesis, Dept. of Mechanical Engineering, Univ. of British Columbia, Vancouver, Oct. 1997.

[4]Astrom, K. J., and Wittenmark, B., *Computer-Controlled Systems: Theory and Design*, 2nd ed., Prentice–Hall, Englewood Cliffs, NJ, 1990, pp. 445, 446.

[5]Antsaklis, P., "Defining Intelligent Control," *Proceedings of the 1994 IEEE International Symposium on Intelligent Control*, IEEE Control Systems Society, Piscataway, NJ, 1994.

[6]de Silva, C. W., *Intelligent Control: Fuzzy Logic Applications*, CRC Press, Boca Raton, FL, 1995.

[7]Jain, L. C., and de Silva, C. W., *Intelligent Adaptive Control: Industrial Applications*, CRC Press, Boca Raton, FL, 1999.

[8]Zadeh, L. A., "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-3, No. 1, 1973, pp. 28–44.

[9]Zadeh, L. A., "A Linguistic Approach to System Analysis," *Proceedings of the 1973 International Conference on Cybernetics and Society*, IEEE Systems, Man, and Cybernetics Society, Boston, 1973, pp. 259, 260.

[10]Lips, K. W., and Modi, V. J., "Flexible Appendages Deployment Dynamics as Applied to Satellite Attitude Motion," *Acta Astronautica*, Vol. 5, 1978, pp. 787–815.

[11]Lips, K. W., and Modi, V. J., "Three-Dimensional Response Characteristics for a Spacecraft with Deploying Flexible Appendages," *Journal of Spacecraft and Rockets*, Vol. 4, 1981, pp. 650–656.

[12]Modi, V. J., and Ibrahim, A. M., "A General Formulation for Librational Dynamics of Spacecraft with Deploying Appendages," *Journal of Guidance, Control, and Dynamics*, Vol. 7, 1984, pp. 563–569.

[13]Modi, V. J., and Shen, Q., "On the Dynamics of Spacecraft with Flexible, Deployable and Slewing Appendages," *Advances in the Astronautical Sciences*, edited by A. K. Misra et al., Univelt, Publisher for the American Astronautical Society, San Diego, CA, Vol. 85, Pt. I, pp. 877–896.

[14]Marom, I., and Modi, V. J., "On the Dynamics and Control of the Manipulator with a Deployable Arm," *Advances in the Astronautical Sciences*, edited by A. K. Misra et al., Univelt, Publisher for the American Astronautical Society, San Diego, CA, Vol. 85, Pt. 3, 1993, pp. 2211–2229.

[15]Modi, V. J., Chen, Y., Misra, A. K., de Silva, C. W., and Marom, I., "Nonlinear Dynamics and Control of a Class of Variable Geometry Mobile Manipulators," *Proceedings of the Fifth International Conference on Adaptive Structures*, edited by J. Tani et al., Technomic, Lancaster, PA, 1994, pp. 140–149.

[16]Hokamoto, S., Modi, V. J., and Misra, A. K., "Dynamics and Control of Mobile Flexible Manipulators with Slewing and Deployable Links," *Advances in the Astronautical Sciences*, edited by K. T. Alfriend et al., Vol. 90, AAS Publications, San Diego, CA, 1995, pp. 339–357.

[17]Hokamoto, S., and Modi, V. J., "Nonlinear Dynamics and Control of a Flexible Space-Based Robot with Slewing-Deployable Links," *Proceedings of the International Symposium on Microsystems, Intelligent Materials and Robots*, edited by J. Tani and M. Esashi, 1995, pp. 536–539.

[18]Hokamoto, S., and Modi, V. J., "Formulation and Dynamics of Flexible Space Robots with Deployable Links," *Transactions of the Japan Society for Mechanical Engineers*, Vol. 62, No. 596, 1996, pp. 1495–1502.

[19]Caron, M., "Planar Dynamics and Control of Space-Based Flexible Manipulators with Slewing Deployable Links," M.S. Thesis, Dept. of Mechanical Engineering, Univ. of British Columbia, Vancouver, Oct. 1996.

[20]Zadeh, L. A., "Fuzzy Sets," *Information and Control*, Vol. 8, 1965, pp. 338–353.

[21]Mesarovic, M. D., Macho, D., and Takahara, Y., "Theory of Hierarchical Multilevel Systems," *Mathematics in Sciences and Engineering*, edited by R. Bellman, Vol. 68, Academic, New York, 1970, pp. 34–108.

[22]Zadeh, L. A., "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems," *Fuzzy Sets and Systems*, Vol. 11, 1976, pp. 199–227.

[23]Rasmussen, J., "The Role of Hierarchical Knowledge Representation in Decision Making and System Management," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-15, 1985, pp. 234–243.

[24]de Silva, C. W., and MacFarlane, A. G. J., "Knowledge-Based Control Structure for Robotics Manipulators," *Proceedings of the IFAC Workshop on Artificial Intelligence in Real-Time Control*, 1988, pp. 143–148.

[25]Saridis, G. N., "Knowledge Implementation: Structure of Intelligent Control Systems," *Journal of Robotics Systems*, Vol. 5, No. 4, 1988, pp. 255–268.

[26]de Silva, C. W., "An Analytical Framework for Knowledge-Based Tuning of a Servo Controller," *Engineering Applications in Artificial Intelligence*, Vol. 4, No. 3, 1991, pp. 177–189.

[27]de Silva, C. W., "Fuzzy Information and Degree of Resolution Within the Context of a Control Hierarchy," *Proceedings of IECON'91*, 1991, pp. 1590–1595.

[28]Barlev, S., "Implementation and Evaluation of an Ill-Defined Servo-Motor System," M.S. Thesis, Dept. of Mechanical Engineering, Univ. of British Columbia, Vancouver, July 1992.

[29]Wickramarachchi, N., "Development of a Knowledge-Based Hierarchical Control Structure for Process Automation," Ph.D. Thesis, Dept. of Mechanical Engineering, Univ. of British Columbia, Vancouver, June 1995.

[30]Caron, M., Modi, V. J., Pradhan, S., de Silva, C. W., and Misra, A. K., "Planar Dynamics of Flexible Manipulators with Slewing Deployable Links," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 4, 1998, pp. 572–580.

[31]Ziegler, J. G., and Nichols, N. B., "Optimum Settings for Automatic Controllers," *Transactions of the ASME*, Vol. 64, 1942, pp. 759–768.