Fifth IFSA World Congress (1993), 1410-1413

FUZZY CONTROL OF THREE LINKS OF A ROBOTIC MANIPULATOR

KISHAN KUMBLA and MO JAMSHIDI

CAD Laboratory for Intelligent and Robotics Systems
Department of Electrical and Computer Engineering
The University of New Mexico, Albuquerque, NM 87131, USA

ABSTRACT

This paper presents the application of fuzzy control to three links of a Rhino robot and compares its performance to traditional PD control. The dynamics of motion of robot links are governed by nonlinear differential equations. The fuzzy controller, being an adaptive technique, gives better performance than the traditional linear PD controller over a typical operational range. The fuzzy controller reaches the desired position with no overshoot, which is unlikely with the PD controller.

1. INTRODUCTION

Robot control requires the solution of nonlinear differential equations. Unfortunately, nonlinear differential equations are plagued by substantial requirements for computation and have an incomplete theory of solution. Thus, most approaches to robot controller design have suffered due to the complications of nonlinear effects.

Because of these complications, fuzzy logic offers a very promising approach to robot controller design. Fuzzy logic offers design rules that are relatively easy to use in a wide range of applications, including nonlinear robotic equations. Fuzzy logic also allows for design in cases where models are incomplete, unlike most design techniques. In addition, microprocessor-based fuzzy controllers have performed with data streams of 8 bits or less to allow for a simple design.

The potential for improvement offered by fuzzy logic warrants further investigation into its usefulness in realistic settings. This paper presents the hardware and software design, and subsequent testing and analysis, of a project to study control of a three link Rhino robot.

2. BASELINE HARDWARE DESIGN

Figure 1 provides an overall block diagram of the robot and control hardware. The Rhino robot has five links and a gripper. Fuzzy control was applied to three of the links: the shoulder, elbow, and wrist. Each link is moved with a DC servo motor rotating about an axis at each joint. The computer system used was a Packard Bell 386PC. The remaining hardware consisted of elements for data acquisition, power control (driver), and communications.

The DC motors were driven by a Fluke 4265A programmable power supply, which was interfaced to the motor through relays. These relays were computer controlled to switch power to only one motor at a time. Thus, at any point, only one of the links could be actuated. The power supply output voltage was controlled with an eight bit control word and could deliver voltages ranging from -16 to +16 volts in 0.25 volt increments.

The data acquisition hardware gathered angular position from optical encoders that pulsed a counter, giving a measurement of displacement. Each pulse corresponded to 0.12 degrees of movement. At every ten milliseconds a crystal clock triggered sampling of the counter. The count information was latched and transmitted to the computer through the RS-232C COM1 port in eight bit increments.

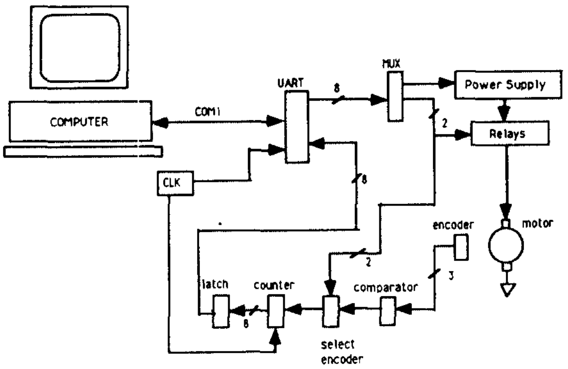


Figure 1. Schematic of the Robot Controller Hardware.

3. SOFTWARE IMPLEMENTATION OF THE FUZZY CONTROLLER

The controller software was divided into two primary modules written in the C language. The two modules are the fuzzy control module and the communications module.

3.1 Fuzzy Control Module

The first step in fuzzy controller design is the selection of input and output variables. The designated input variables were ERROR (robot link position error) and VELOCITY (velocity of the robot link). The output variable was chosen to be VOLTAGE (drive voltage output by the power supply). Each variable requires a set of membership functions. The membership functions are shown in Figure 2.

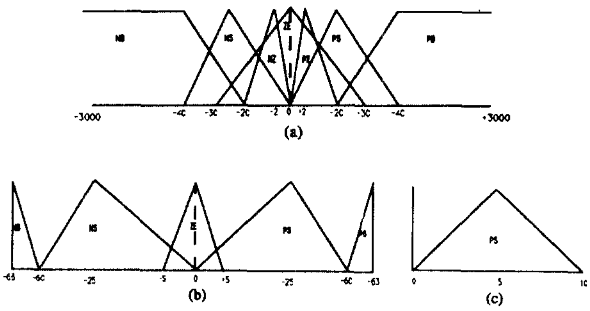


Figure 2. Fuzzy Membership Functions for (a) Position Error, (b) Output Voltage and (c) Velocity.

During actual operation, the computer reads in the robot position and then computes position error and velocity. The fuzzy controller fuzzifies the input quantities through algorithms that operate on the input data as specified by the membership functions. Next, the fuzzified input quantities pass through a series of IF-THEN decision rules that form the main body of the fuzzy controller. Thus, the fuzzy controller assesses the current state of the robot and determines which control action is most appropriate. Defuzzification is applied using the voltage output variable and a control action is selected. The control action is then passed through the interface to the external power supply driver.

The design of the fuzzy controller algorithm was accomplished with the Togai InfraLogic's fuzzy C Compiler [1]. Input to the Togai Fuzzy-C Compiler is a fuzzy source code consisting of membership functions and a set of IF-THEN decision rules. The Fuzzy-C Compiler converts the fuzzy source code into standard C source code which then passes through a C compiler to produce an executable code of the fuzzy controller. Trial experiments of the executable code were conducted to calibrate the membership functions until robot trajectory overshoot was suppressed and rise time was kept to a minimum.

3.2 Communications Module

The COM1 serial port of the PC was used to communicate between the computer and external hardware. A communications routine set the port at 9600 baud, eight data bits, two stop bits and even parity. The routine sampled the count information in the receive register every ten milliseconds. The fuzzy controller operates on this data and then outputs the result into a transmit register which was used to drive the DC motors.

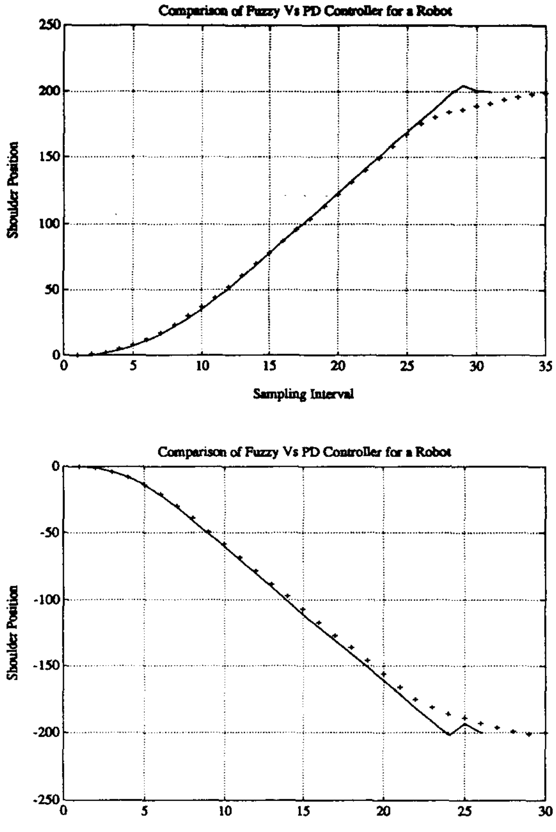
4. TESTING AND RESULTS

Two tests were conducted to compare controller performance on robot arm trajectory. In both tests, the gain of the PD controller was set to provide the best performance in all operating ranges. In the first test, each controller was set to traverse a trajectory of 200 counts in both positive (clockwise) and negative (counterclockwise) directions (Note: One count is equal to 0.12 degrees of rotation of the link). The test results for the trajectory of the shoulder, elbow and wrist are shown in Figure 3. The results show that the fuzzy controller is able to move the robot arm smoothly to the desired position without overshoot. The PD controller, however, generated overshoot in the trajectory. It is interesting to note that when the robot link approached its final destination, its velocity remained high (steep trajectory curve) with PD control, but began to decrease with fuzzy control.

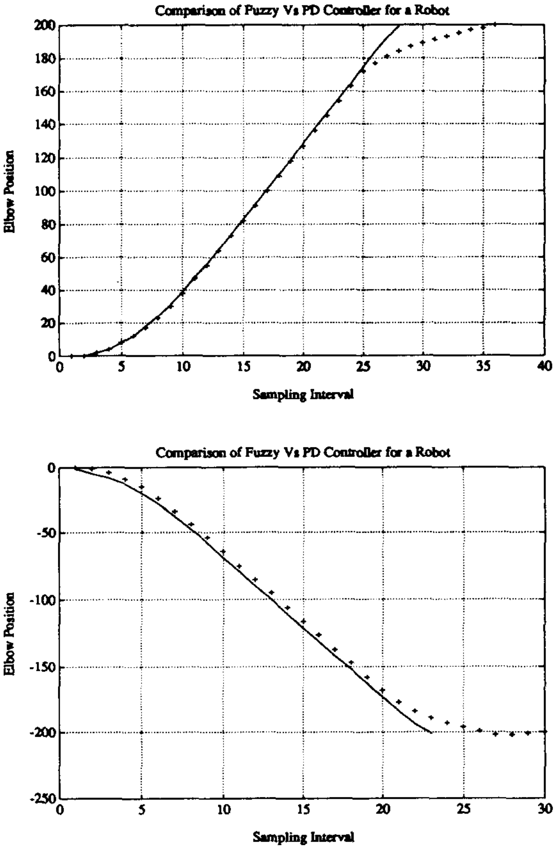
In the second test, the controllers were compared under various ranges of operation. The PD controller with suitable gain improved trajectory response by reducing the overshoot to less than four counts. However, the PD controller was unable to move the motor when the links were initially one or two counts away from the final destination. The primary cause for this effects was the nonlinear friction in the robot joints. One could overcome this resistance with an increase in controller gain, but then the links would overshoot even further when larger distances are traversed.

The fuzzy controller was able to overcome the problems encountered by traditional PD control and perform better in all conditions tested. Results showed that the maximum overshoot with fuzzy control was held to a count of one. The fuzzy controller also was able to actuate the motor and travel a short distance of one count, unlike the PD controller.

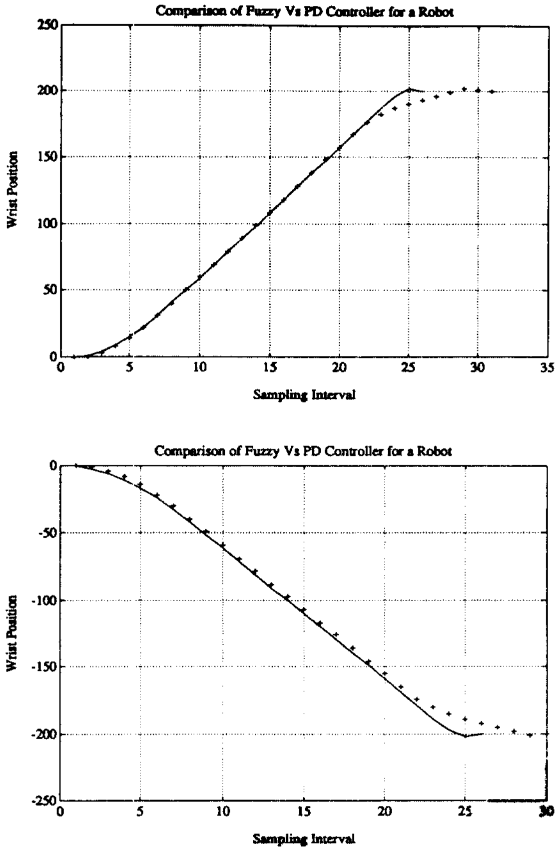
The fuzzy controller exhibited robustness in



'+++' : Fuzzy Controller. '----' : PD Controller.
Figure 3a. Shoulder Motion of the Robot.



'+++' : Fuzzy Controller. '----' : PD Controller.
Figure 3b. Elbow Motion of the Robot.



'+++' : Fuzzy Controller. '---' : PD Controller.
Figure 3c. Wrist Motion of the Robot.

performance against non-ideal effects like robot inertia, Coriolis effect and gravity. These effects alter the velocity, position, and acceleration of the robot links, which can degrade overall performance of conventional control. However, with fuzzy control, one set of fuzzy membership functions was sufficient to guard against these non-ideal effects and operate all links of the robot. This result is not shared by the PD controller which required different gains to accommodate for the differences between each link.

5. CONCLUSIONS

The fuzzy controller was able to suppress the robot trajectory overshoot and perform better than traditional PD control under all scenarios tested. Furthermore, one set of fuzzy membership functions was sufficient for handling the variations that occurred between the different links of the robot. This implies that fuzzy control is robust and can accommodate many unforeseen elements inevitable in any practical implementation.

6. REFERENCES

[1] Togai InfraLogic Inc. Fuzzy-C Development System User's Manual, Irvine, CA, 1991.

[2] Rhino, Inc., Rhino Robot XR Series Owner's Manual, Champaign, IL, 1988.

[3] Craig, John J. Introduction to Robotics. Addison Wesley, Reading, Mass., 1989.

[4] Larsen, D. G., Rony, P. R., Titus, J. A., and Titus, C. A. Interfacing and Scientific Data Communications Experiments. Howard W. Sams & Co., Inc., Indianapolis, Indiana, 1979.