Proceedings IEEE/RSJ/GI International Conference on Intelligent Robots and Systems IROS, Sep. 1994.

Position and Force Control of a Multifingered Hand: A Comparison of Fuzzy Logic to Traditional PID Control*

Dimitris Hristu, Jon Babb, Harjit Singh, and Susan Gottschlich
NYS Center for Advanced Technology in Automation and Robotics
and
Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, NY 12180

Abstract

In this paper we will discuss the control of a five-fingered, 18 degree of freedom, tendon driven robot hand known as the Atlas Anthrobot. Developing a traditional PID controller for this hand has proved difficult because the behavior of the tendon drive system produces is highly nonlinear. This nonlinear nature of the system however does suggest it as a candidate for fuzzy control. Along these lines we have developed a PID control system and a fuzzy control system for the hand which are both capable of simultaneously controlling the position of the fingertips and the amount of force exerted through a contact point on the fingertip. In this paper we will present a comparison between the two systems in order to generally make statements about the appropriateness of each controller for controlling complicated tendon driven devices such as our robot hand.

1 Introduction

Our ultimate goal is to develop a control system for our robot hand capable of accurately controlling both the position of and force exerted by each robot fingertip. While force control is not necessary in the grasping phase of a robotic operation, it is imperative for the hand to be able to manipulate objects to insure that a stable grasp is maintained on the object during manipulation. The reasons for this have been justified in [Hristu 94].

*This work has been conducted in the New York State Center for Advanced Technology (CAT) in Automation and Robotics at Rensselaer Polytechnic Institute. The CAT is partially funded by a block grant from the New York State Science and Technology Foundation. This work has also been partially funded by NSF Grant number IRI-9211366.

Our original approach to this problem was to develop a PID controller for the hand. This control system has been described in [Singh 93]. However, the results of previous research on the PID controller [Zink 93] and of preliminary experiments with a fuzzy controller for just the position control loop led us to believe that a fuzzy controller may yield a better control system than was possible with just a PID controller. For this reason, we have developed a fuzzy controller and will compare it to the PID controller originally implemented. Our purpose in doing this is to determine which of the two paradigms is better suited for the robot hand. Of course, we are well aware that such an empiracle determination is not completely rigorous, however given the nature of fuzzy control systems and the divergence between the paradigms, a side-by-side theoretical comparison would be very difficult.

To further describe the requirements of a controller for the hand, a more extensive discussion on the robot hand and the problems that arise due to the tendon drive is necessary. The hand has 18 degrees of freedom, four for the thumb, three for each finger, and two associated with the wrist. The four degrees of freedom of the robot thumb roughly approximate a human thumb, which in reality, has five degrees of freedom. Thus rather than producing a three degree of freedom saddle joint at the base of the thumb, the robot hand instead been has a two degree of freedom joint system. The degrees of freedom associated with each finger is identical to the human fingers except that the last two joints on the finger are coupled together. Thus, if the middle finger joint rotates θ degrees, the last finger joint (known as the *distal finger joint*) also rotates θ degrees. The wrist degrees of freedom are currently not being controlled but rather are mechanically fixed by a bracket on the wrist.

Each finger joint on the hand is controlled by a servomotor attached to the joint by a tendon enclosed in a flexible conduit. The 18 servomotors needed to drive all 18 degrees of freedom are housed in a box behind the wrist of the Anthrobot. The flexible tendon conduit is not rigidly supported and moves during joint motion. Thus, the servomotor turns before the joint begins to move. The overall system thus exhibits hysteresis and spring effects which make it highly nonlinear. This hysteresis effect is illustrated in Fig. 1. The conduit may be approximately modeled as a spring be-

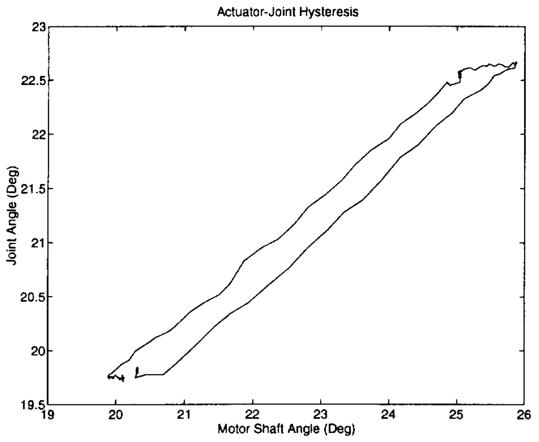

Figure 1: Actuator-Joint Hysteresis Curve

tween the servomotor and the joint. This spring action means that joint movement is not easily mappable to motor movement – there are delays between the motor movement and the corresponding joint movemet and nonlinearities in the relationship between the two.

Since knowledge of either the joint position or the servomotor position does not reveal the position of the other, both positions must be monitored. Potentiometers are mounted on the servomotor shaft and each finger joint. Velocity can be estimated by taking the difference between two consecutive position readings. Force data is supplied by a tactile sensor mounted on the fingertip. A more detailed system description of the hand is given in [Singh 93].

2 Review of Previous Work

The development of the current PID and fuzzy control systems has involved multiple iterations over a course of approximately three years. The first contribution to this effort was the development of the kinematics for the hand. While at first blush this problems may seem to be a straightforward extension to the usual work done in kinematics for robot arms, one significant difference arose. On a robot arm, the kinematics and inverse kinematics are always derived relative to some *tool point* at the end of the robot arm kinematic chain. On the robot hand, however, there are no obvious tool points on the digits. Generally speaking, the kinematics are necessary to relate the joint angles of the digits (fingers and thumb) to the position of the *point at which the digit contacts an object.* Unfortunately, however, this point can be any place at which the digit contacts the object. Thus the tool point is not fixed on the robot fingers.

In [Van Riper 92] was shown that the tool point for each finger can be parameterized and thus a moving tool point can be accounted for in the kinematics and inverse kinematics. A follow up to this work [Ali et al. 93] described the Jacobians of the hand, which again can be calculated so long as the parameters of the tool point are known.

The first effort to develop a position controller is described in [Zink 93]. The hand was originally only equipped with position sensors at the motors and not at the finger joints, so an extensive effort to model the tendon drive so that an accurate determination of the joint angless given knowledge of the motor shafts was carried out. The result of this investigation was that the relationship was so nonlinear (particularly due to the hysteris described previously) that developing a model that was mathematically tractible was not possible. Thus, it was determined that a position sensor at each end of the tendon was necessary to achieve reasonable performance. Toward this end, a potentiometer was mounted on the side of each joint and a PID position controller was developed.

Further work investigated the performance of a fuzzy position controller using the same setup as described in [Zink 93]. The results of this investigation showed that for the most part the fuzzy controller was superior to the PID controller, although there were of course some tradeoffs to be considered.

A complete PID hybrid force/position controller was presented in [Singh 93]. Preliminary work in force/position control was presented in [Zink 93], however at that time the requirements of the force/position controller were not fully known and thus the force/position control system presented in [Zink 93] did not satisfy the requirements established in [Hristu 94].

A major emphasis of the work presented in [Singh 93] was sensor development. In particular, the joint potentiometers used in [Zink 93] no longer were suitable because a newer version of the robot hand (the Anthrobot III rather than the Anthrobot II) was used and there was no longer enough room at the finger joints to properly mount the potentiometers. It was further desired to use a joint sensor that did not obstruct the lateral motions of the fingers. Toward this end, a sensor was developed that essentially converts the finger joint itself into a potentiometer – a semiconductive surface is mounted directly on the joint, and a wiper is attached to the following link so that as the finger joint moves, the resistance between the one end of the semiconductive surface and the wiper changes as well.

The tactile sensors to be used in the control system are presented in [Gery 93]. Unfortunately, at the time that these experiments were carried out, the tactile system presented in [Gery 93] was still not operational, so instead a simple force sensing resistor was used to measure the magnitude of the contact force exerted on the fingertip.

3 System Design

The general system diagram for controlling the hand is depicted in Fig. 2. The user provides to the system the parameters indicating the tool point on each digit[1],

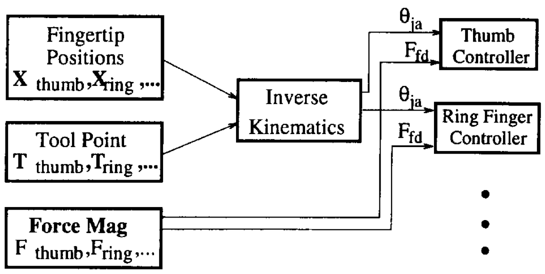

Figure 2: The PID force and position control loop of the hand.

the desired position of each tool point (fingertip position), and the desired force to be exerted at each tool point. The inverse kinematics are used to map the desired position of the tool point into a desired joint position vector θ_{jd} for each finger and a desired force magnitude F_{fd}. The joint position vector and the force magnitude are then fed into either the PID or fuzzy position/force servocontroller for each digit.

A simplified block diagram of the PID position/force servocontroller is depicted in Fig. 3. It is comprised of

[1]The tool point often corresponds to the contact point of the finger. Thus, this information can be obtained directly from tactile sensors if thye are capable of measuring the location of a contact point.

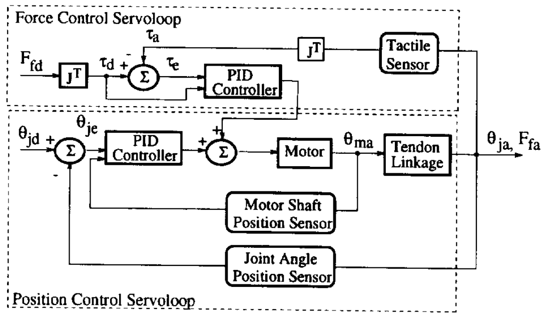

Figure 3: The PID force and position control loop of the hand.

a position control servoloop and a force control servoloop whose output is fed into the position control loop. The position control loop makes use of both the motor shaft position sensor (which measures the actual motor shaft position θ_{ma}) and the joint angle sensor (which measures the actual joint angle θ_{ja}) for each joint. Not explicitly shown in the figure is that the PID controller for the position control loop also makes use of the motor shaft velocity and joint velocity. These velocities are estimated by taking the discrete derivative of the motor shaft position and joint position respectively. Also not explicitly shown is that the tool point information is needed to calculate the Jacobian transposes J^T.

In this control system, a force command is essentially converted into a desired motor torque (τ_d) which is then converted into an incremental position command that is independent of the position command supplied by the user. The two commands are simply summed. The rationale here is that if the difference between the desired force and the actual force is zero, than the position command is followed. However, if this difference is not zero, the position of the finger must be adjusted to account for this difference. Generally speaking, the force command and the position command are consistent, so this procedure produces the desired result. Unfortunately, if they are inconsistent, the result will be a compromise between the force command and the position command. Along these lines, a weighting factor may be allowed to decide whether the position command is more important or the force command is more important. The higher the weighting factor associated with the force command, the more it takes precidence over the position command.

A block diagram of the Fuzzy servocontroller is depicted in Fig. 4. Here, a weighting scheme has been used where the weight K_c, which is referred to as a

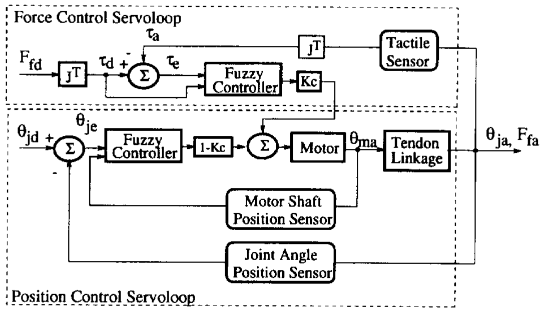

Figure 4: The fuzzy force and position control loop of the hand.

compliance index, is currently fixed by the user. The idea is that the user selects how compliant the system should be, in other words, how important satisfying the force command is relative to that of satisfying the position command. In the future, an additional fuzzy control loop could be used to alter this weight dynamically and automatically. In the following we will discuss the specifics of the fuzzy position servoloop and the fuzzy force servoloop in detail.

3.1 Fuzzy Position Control Servoloop

The fuzzy controller in the position control servoloop has three inputs, joint angle error $\theta_{\mathbf{je}}$, joint velocity $\dot{\theta}_{\mathbf{ja}}$ and motor velocity $\dot{\theta}_{\mathbf{ma}}$. Both velocities are calculated in a discrete fashion by differencing consecutive motor and joint position readings. The intuition the fuzzy joint position control is the following: When $\theta_{\mathbf{je}}$ is big, we can drive the motor at full speed. When $\theta_{\mathbf{je}}$ becomes small, we need to look at $\dot{\theta}_{\mathbf{ja}}$ and $\dot{\theta}_{\mathbf{ma}}$ to slow down or stop the joint because the non-linearities of the system become important. Also, since the hand is usually used for grasping objects, the controller must provide damping so that there is no overshoot in the joint angles or the object held may be dropped.

The joint error $\theta_{\mathbf{je}}$ was partitioned in five regions: positive big, positive small, zero, negative small and negative big. The joint and motor velocities were partitioned in three regions, positive zero and negative. Since they are obtained by differentiation, the velocities are somewhat noisy and are only used to determine the direction at which the joint is moving. Thus, three regions have proved to be sufficient. The output of the controller controls the motor current and is partitioned in five region just like $\theta_{\mathbf{je}}$. The fuzzy sets used for all variables are given in Appendix C.

3.2 Fuzzy Force Control Servoloop

A force sensor at the fingertip is used to measure the magnitude of the force exerted on the fingertip and the location of the contact point. The force sensor data is converted into joint torque by using the finger jacobian. The inputs to the controller are the joint torque error $\tau_{\mathbf{e}}$ and its derivative $\dot{\tau}_{\mathbf{e}}$ (calculated discretely). In essence, the controller regulates the fingertip force indirectly, by controlling the joint torques. This is necessary due to the kinematic redundancy of the finger (motion in a number of different joints can produce the same change in fingertip force). The inputs are partitioned in five regions each and so is the force controller output. Their fuzzy sets are given in Appendix C.

The center of gravity method was used to defuzzify the output. This way, any rule with a non-zero output contributes to the outcome. The output is computed via the equation

$$Output = \frac{\sum_{j=1}^{n} \mu_j \cdot Y_j}{\sum_{j=1}^{n} \mu_j} \tag{1}$$

where n is the number of rules, μ_j is the applicability of a rule and Y_j is its output.

In all 29 rules were used for the position control servoloop and 25 rules were used for force control servoloop. All rules are given in Appendices A and B.

4 Implementation

Both controllers were developed for the Atlas Anthrobot III and were coded in C, running on a 33 MHZ 68040 processor card. The voltage across the joint and motor potentiometers and across the force sensors are read using an A/D converter board. The software for both controllers relied upon a common core of routines to read sensor data, make necessary computations such as the finger Jacobians, etc.

Because the system must perform time-consuming computations for all five finger both for position and force control, the fuzzy rules were stored as look-up tables, i.e. arrays that had the input used as an index and the membership functions as stored data. At the start of every control cycle, the membership functions of all input variables are looked up in the table. Then, all rules are processed. Finally, the controller defuzzifies the outputs using either mean of maxima, or center of gravity. The controller output(s) are multiplied by a gain and sent to the hardware (motors).

Even without storing the rules symbolically, i.e. as functions of (a, b, α, β), the maximum frequency at which we could run the controller was around 50Hz. Therefore, software was written to compile the rules off-line, and store them or read them from a file. Thus, an array indexed by the inputs was created for each controller module (force and position) and the outputs for all possible inputs were stored. Using the universes of discourse shown in Appendix C, with a discretization of two counts per degree for joint angles, the overall storage requirement was about 2Mbytes out of the 16Mbytes available, which does not seem excessive. We were able to consider inputs that were outside the strict universe of discourse by using saturation detection. Thus, if an angle was bigger than 30 degrees (our universe of discourse limit for joint angles), it was simply considered to be *at* the limit, i.e. 30 degrees. This worked well and greatly reduced the space requirement for considering all inputs when compiling the rules. The improvement in the speed resulted in a maximum rate of about 110Hz. However, the option exists to evaluate the rules at run-time if saving memory is important. Finally, all variable ranges, memberships etc, were stored in terms of the servo loop speed, so no re-tuning is required when moving to different rates.

5 Experimental Results and Discussion

In the first experiment, the position controller was run by itself. The command called for a 45 degree move of one finger joint. The performance of the fuzzy controller was plotted against that of the PID controller for comparison. The PID controller required about 0.5 seconds to settle to within 1 degree of the desired angle. The fuzzy controller required 0.4 seconds to do the same. In both cases there was no overshoot, however, the fuzzy controller settled with near zero error. Fig. 5 shows the comparison, with the overall response, and a closer look around the ranges of interest.

Figure 6 shows the dominant rules that fired during the run (the rules that had the maximum degree of applicability at each time step), and plots of the position controller output, joint and motor velocities.

Next, the force controller was tested separately. While the finger was close (but not touching) to a rigid flat surface, the controller was given a desired force of 0.3. We must mention that the force sensor used returns a reading between zero and one. A force of 0.3 roughly corresponds to 200gr. Also, a reading of 0.05 is considered to be the threshold of meaningful detection. However, the sensor output to force characteristic

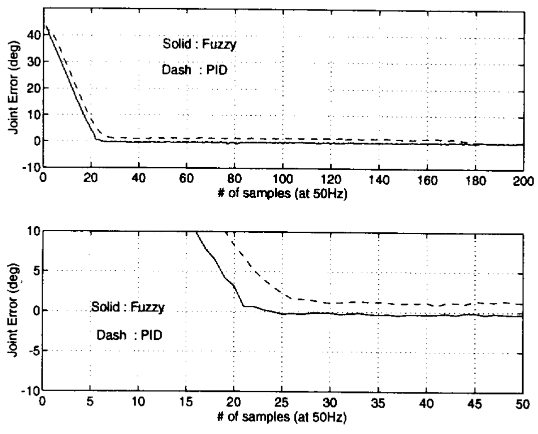

Figure 5: Joint Error Comparison

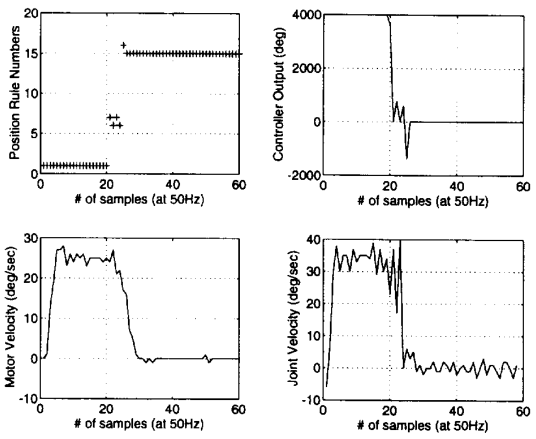

Figure 6: Fuzzy Position Controller Response

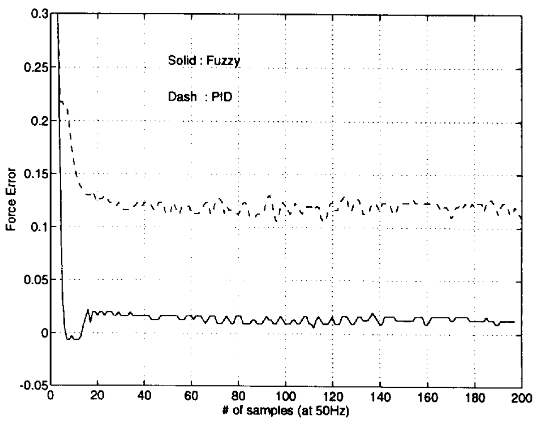

Figure 7: Force Error Comparison

is nonlinear.

Fig. 7 shows the comparison between the fuzzy and PID controllers. The settling time is the same for both controllers, however the fuzzy controller outperforms the PID in steady state error.

The reason for this is that the fuzzy controller is better able to handle the nonlinearities of the sensor because the nonlinearities of the sensor can be captured automatically in the thresholds used for the rules. To handle the nonlinearities in the PID controller, we must develop a sophisticated nonlinear calibration scheme, which has not been done.

With reference to the fuzzy controller, Fig. 8 shows the force control rules that had maximum applicability,

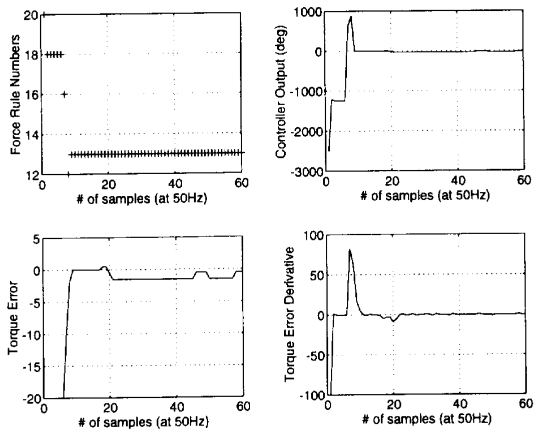

Figure 8: Fuzzy Force Controller Response

the controller output, the joint torque error, and its derivative.

Comparing the manner in which the position and force servoloops operate together on the two controllers is really not possible because they do not operate in the same fashion. Thus, we performed three experiments with the fuzzy controller only where the relative weights between the position and force controller were varied. Thus, the role of the compliance index K_c was demonstrated.

In Fig. 9 the controller was asked to provide zero force while maintaining the proximal, middle and distal joints of the finger at around 20 degrees each. Then, a force was exerted on the fingertip. The force and joint angles were monitored. The force readings were **magnified** by a factor of **20** in order to more clearly depict the force response on the graphs. A compliance index K_c of zero was set so that the position control dominates and force control is ignored. This corresponds to the finger(s) being very stiff. As we can see,

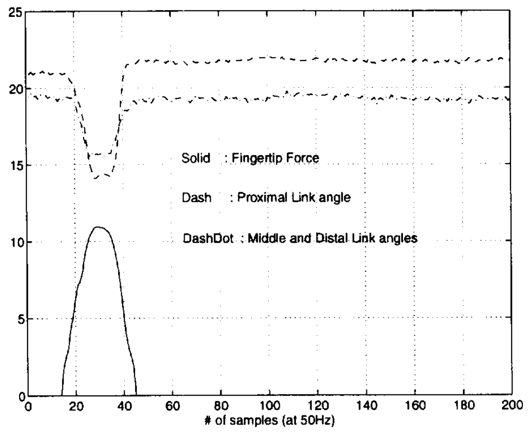

Figure 9: Controller Response: K_c=0

the force causes the finger to bend, but once removed, the joints return to their original position or very close to it. The maximum force applied was about 0.6 in sensor reading.

Another experiment was run with K_c set to 0.5, which means that the position and force controls are equally weighed. The controller was asked to maintain zero force and joint angles of 35 and 38 degrees. Figure 10 shows the results. This time, the joints are

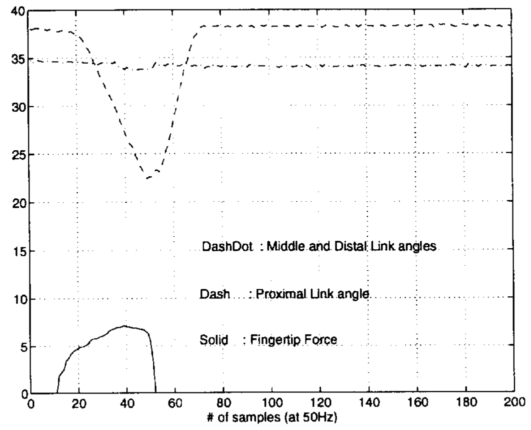

Figure 10: Controller Response: K_c=0.5

not as stiff (we can see it takes much less force to effect a larger joint angle change), but once the force is removed, the finger returns to position.

Finally, an experiment was run with K_c set to 1 (zero stiffness for the fingertip). The controller was asked to maintain a force of 0.5 and joint angles around 35 and 45 degrees, while being touched with some small force. As the finger tried to increase the force to the required level (ignoring position) the force moved away, causing the joints to move as well. As we can see in Fig. 11, initially the joints move so that the desired force is achieved, and continue to follow as the force is continually moved away slowly.

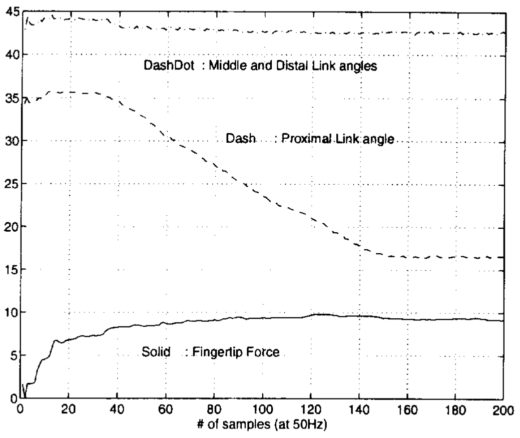

Figure 11: Controller Response: K_c=1.0

6 Conclusion

In this paper we have empirically compared a fuzzy position/force controller with a PID position/force controller. The results confirmed our suspicion – that the fuzzy controller outperformed the PID controller in tests for pure position control and pure force control. Experimental results concerning the hybrid controller were also presented to indicate the effects of weighting between position commands and force commands. The problem, of course, with such empirical tests is that the results are dependent on how well tuned each system is, particularly in the case of the fuzzy controller. Future work will involve the further tuning of these two systems and the continued evaluation of their performance, as our ultimate goal is a well-controlled robot hand.

References

[Ali et al. 93] M. S. Ali, K. J. Kyriakopoulos, and H. E. Stephanou. The Kinematics of the Anthrobot-2 Dextrous Hand. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1993.

[Gery 93] B. L. Gery. A Conductive Ink Tactile Sensing System with High Speed Data Acquisition and Automatic Calibration Capability. Master's thesis, Rensselaer Polytechnic Institute, November 1993.

[Hristu 94] Dimitris Hristu. Manipulation of Unknown Objects with a Multifingered Hand. Master's thesis, Rensselaer Polytechnic Institute, May 1994.

[Singh 93] H. Singh. Joint Control Of An Anthropomorphic Robotic Hand. Master's thesis, Rensselaer Polytechnic Institute, 1993.

[Van Riper 92] J. Van Riper. The Kinematics for an Anthropomorphic Robot Hand. Master's thesis, Rensselaer Polytechnic Institute, March 1992.

[Zink 93] Andrew Zink. Dynamic Modeling and Force/Position Control for the Anthrobot Dextrous Robot Hand. Master's thesis, Rensselaer Polytechnic Institute, 1993.

A Fuzzy Joint Torque Controller Rules

	Torque Error	Error Change	Output
1	pos. big	pos. big	pos. big
2	pos. big	pos. small	pos. big
3	pos. big	zero	pos. big
4	pos. big	neg. small	pos. small
5	pos. big	neg. big	zero
6	pos. small	pos. big	pos. big
7	pos. small	pos. small	pos. big
8	pos. small	zero	pos. small
9	pos. small	neg. small	zero
10	pos. small	neg. big	neg. small
11	zero	pos. big	pos. big
12	zero	pos. small	pos. small
13	zero	zero	zero
14	zero	neg. small	neg. small
15	zero	neg. big	neg. big
16	neg. small	pos. big	pos. small
17	neg. small	pos. small	zero
18	neg. small	zero	neg. small
19	neg. small	neg. small	neg. big
20	neg. small	neg. big	neg. big
21	neg. big	pos. big	zero
22	neg. big	pos. small	neg. small
23	neg. big	zero	neg. big
24	neg. big	neg. small	neg. big
25	neg. big	neg. big	neg. big

B Fuzzy Joint Position Controller Rules

	Joint Error	Motor Vel.	Joint Vel.	Output
1	pos. big	anything	anything	pos. big
2	pos. small	neg.	neg.	pos. big
3	pos. small	neg.	zero	pos. small
4	pos. small	neg.	pos.	zero
5	pos. small	zero	neg.	pos. big
6	pos. small	zero	zero	pos. small
7	pos. small	zero	pos.	zero
8	pos. small	pos.	neg.	pos. big
9	pos. small	pos.	zero	pos. big
10	pos. small	pos.	pos.	pos. small
11	zero	neg.	neg.	pos. big
12	zero	neg.	zero	pos. small
13	zero	neg.	pos.	zero
14	zero	zero	neg.	pos. small
15	zero	zero	zero	zero
16	zero	zero	pos.	neg. small
17	zero	pos.	neg.	zero
18	zero	pos.	zero	neg. small
19	zero	pos.	pos.	neg. big
20	neg. small	neg.	neg.	neg. small
21	neg. small	neg.	zero	neg. big
22	neg. small	neg.	pos.	neg. big
23	neg. small	zero	neg.	zero
24	neg. small	zero	zero	neg. small
25	neg. small	zero	pos.	neg. big
26	neg. small	pos.	neg.	zero
27	neg. small	pos.	zero	neg. small
28	neg. small	pos.	pos.	neg. big
29	neg. big	anything	anything	neg. big

C Fuzzy Membership Sets

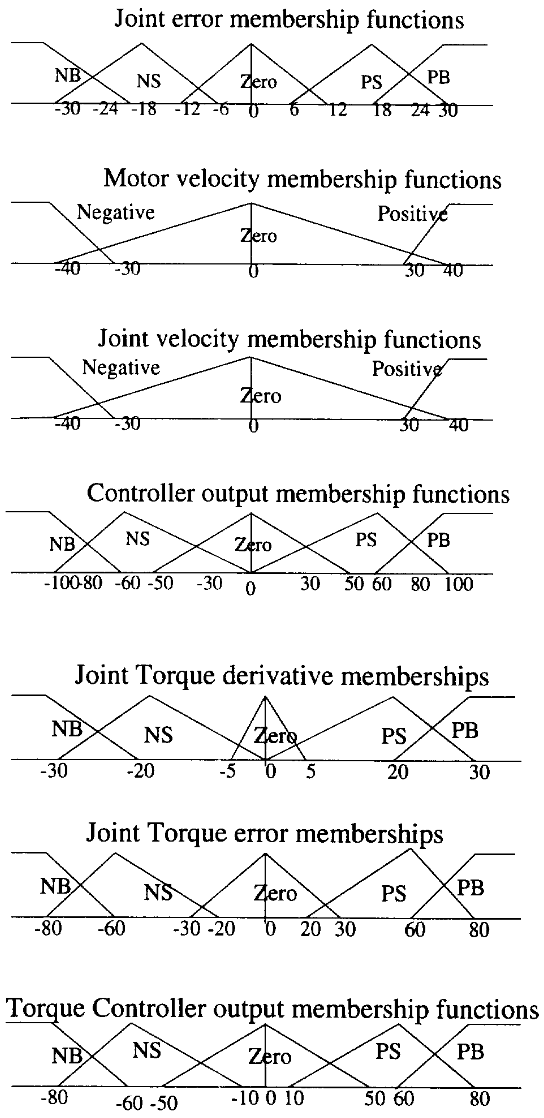

Figure 12: Membership Sets