Journal of Intelligent and Robotic Systems **30:** 209–226, 2001.

Design and Implementation of a Low Cost Multi-Fingered Robotic Hand Using a Method of Blocks

STEVEN M. SPANO and NIKOLAOS G. BOURBAKIS
Binghamton University, Department of Electrical Engineering, Intelligent Robotic and Applied AI Research Lab, Binghamton, NY 13902, U.S.A.

(Received: 26 April 2000; in final form 16 May 2000)

Abstract. The control of a mechanical robotic hand is plagued by an inability to derive accurate dynamic models of such a mechanism. The aspects of static and kinetic friction are major obstacles in the control of a mechanical hand. This paper presents a fuzzy-like based controller and its implementation for a low cost robotic hand. The controller has the ability to automatically regenerate the member set during the translation of any arbitrary joint in a robotic hand. A series of simulations has been conducted that illustrate the effectiveness of this controller for providing smooth translation independent of frictional forces.

The implementation of the controller used here is based on an IBM compatible computer using a custom designed acquisition/conversion interface and a mechanical hand assembly. A hand with a progressively linked finger structure has been used for simplicity. The acquisition system used here allows bidirectional communication with the sensors and actuators of the hand.

A series of experiments have been conducted which verify that the method of blocks was successful for controlling joint position. The simulation results include both fine movements, needed for dexterity, and gross movements that can be used for grasping. This robotic hand produces good results in a low cost implementation.

Key words: dexterous, position, velocity, blocks method, robotic hand, multi-finger hand.

1. Introduction

The goal of advanced robotic system is to develop a combined computer and mechanical structure that can perform operations in a manner analogous to human beings. Such a robotic system must have the ability to change its environment using a grasping end-effector. A mechanical five-finger replica of the human hand is the best choice for such a robot, because the world is designed around five-finger hands. A robot with such an end-effector would be capable of very robust operations. This goal of advanced robotics, according to classical control theory, is plagued by an inability to accurately model the dynamics of such a complex end-effector. The result is poor control of the end-effector [1].

Attempts have been made to enhance the control of robotic hands. These attempts lie in the realms of PID, Neural Network, and Fuzzy Logic Control. Fuzzy logic has been shown to provide better control results than PID [13]. Neural Network control system require time consuming training processes and are therefore not well suited for a system which must perform new tasks, with unknown operational constraints, at any point in time [5].

This paper presents the design, simulation and implementation of a low cost multi-figured robotic hand using a fuzzy-like control system implemented an IBM compatible computer via a custom interface. Simulations of the control system have been conducted to illustrate that the controller is capable of robust operation of hand in the presence of static and kinetic friction. A series of experiments have been also conducted to verify that this control method is capable of high-precision gross and fine translations.

2. Robotic Hand Design

A robotic hand is an electro-mechanical system comprised of many parts. The two main parts to such a device are the electrical components and the mechanical structure that allows motion.

The mechanical structure complexity is directly related to the degrees of freedom (DoF) that the hand may operate in. A typical mechanical structure for a hand includes rotational joints, actuator and joint connections, and the actuator component. The rotation joints (specified by DoF) are used to link mechanical fingers together [1]. These joints allow mechanical fingers to operate in a rotational plane that is similar to human hands. The joints must be connected to an actuator for motion to occur. The actuator is an analog device such as a motor or hydraulic slide.

The two main connection schemes are progressive and discrete. In a progressive actuation scheme, a group of linked joints will rotate is succession based on a primary joint. This primary joint is driven by an actuator. The discrete scheme allows a single actuator for each joint. This scheme allows greater control of the mechanical hand at the cost of increased weight and complexity.

The actuator must be connected to at least one joint (progressive) or all joints (discrete) to allow control of the hand. Joint and actuator connections can be achieved in many ways. The common methods are gear drives and tendon-type arrangements. A gear drive system allows direct connection between the actuator and a joint or between individual joints. This scheme is best suited for a progressive system due to the ability to use ratio-metric gearing to achieve successive joint rotations. A tendon arrangement is used for most discrete mechanical schemes and is typically connected to the primary joint in a progressive system. A tendon, which is a flexible wire or plastic strip, is connected between each joint (discrete) or the primary joint (progressive) and the actuator(s). This arrangement is similar to the anatomical features of the human body in that a tendon links a rotational join to a muscle (actuator). The tendon arrangement saves space by allowing rotation without complex gear systems.

The mechanical structure of the hand is complemented by a variety of electrical components. The primary group of electrical components include sensors and wires. The sensors of a robotic hand are capable of measuring angular position and velocity of a joint along with a degree of tactile feedback. Tactile sensors are used to acquire information ranging from a digital contact/no-contact through an analog representation of surface features which can be used to provide feedback for grasp control systems. The major sensors of the hand are position and velocity. If these two sensors are not present, primitive motion control of the hand could only be achieved through a vision/recognition system.

The secondary electrical system includes the discrete and integrated circuits. The discrete circuits usually provide analog power amplification and trimming. These processes allow for correct operation of the analog actuator. The integrated circuits form an interface between the amplification stage and the processing system. The processing system can be an onboard microcontroller or a separate computer system. The interface is responsible for quantizing sensor inputs and relaying these to the processing system. After the processing system has operated on the data, control outputs (signals for the actuators) are generated and the interface transforms these into the appropriate analog signals for the actuators.

These combinations of electrical and mechanical components form the basis for almost every type of mechanical robotic hand. The joints of the mechanical structure are operated, via gears or tendons, by an actuator and the status of these joints are monitored by electrical sensors. Tactile information may also be monitored by other sensors. An electronic interface is responsible for sampling sensor data and sending it to a processor. The processor provides digital outputs back into the interface. These outputs are transformed from digital to analog, via the interface, and are used to control the actuators. This arrangement forms a closed loop control system which includes the electro-mechanical parts of this structure.

2.1. INTERFACE DESIGN

The block diagram shown in the figure below represents the interface for the hand. This interface is capable of activating four dc motors in forward and reverse bias modes. Combined voltage/power amplifier is connected to each analog output to drive the dc motors. This interface is currently able to sample 8 analog inputs, however, it will be upgraded to 128 inputs. The A/D conversion system operates with a 15 KHz sampling rate Communication with the interface is achieve through the parallel port of an IBM compatible computer. (See Figure 1.)

The control software has the capability of independently adjusting the control efforts (output voltage) of each motor by loading a new 8-bit value into a d/a latch. The interface is biased such that 0–255 corresponds to voltage ranges between -12 and $+12$ VDC. The software can also select an independent analog source to be sampled via the a/d latch.

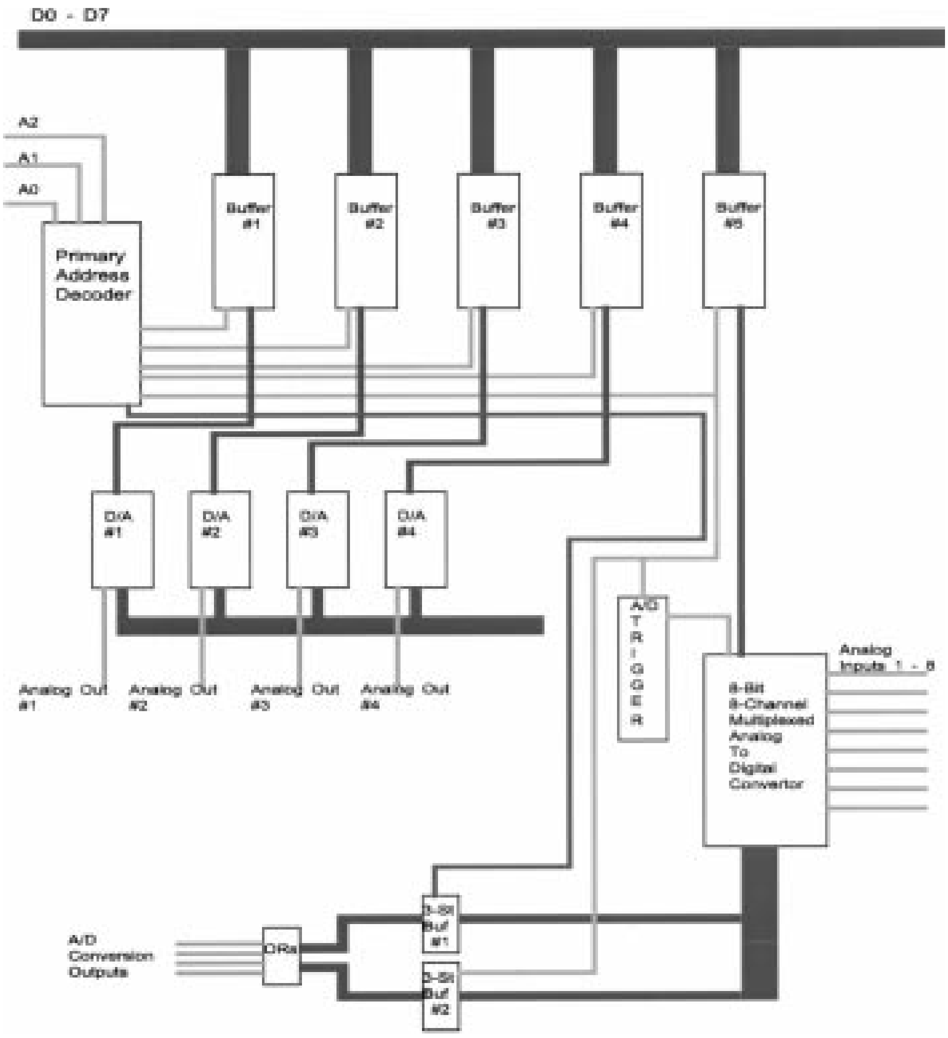

Figure 1. The block diagram of the interface.

2.2. MECHANICAL STRUCTURE

The diagram in Figure 2 represents an individual finger of the mechanic hand. The joints of the finger are progressively linked using a single tendon-type arrangement. The structure is such that angle T_3 increases followed by angle T_2 and finally angle T_1. A simplified force equation, where F_r refers to the elastic resistive force of the tendon and K refers to the tendon spring constant, is shown below.

$$F_r = K \sum L_i \sin(T_i), \quad i = 1, \ldots, 3.$$

A linear potentiometer is directly coupled with the end effector tendon. This potentiometer acts as a voltage divider and functions to provide position feedback

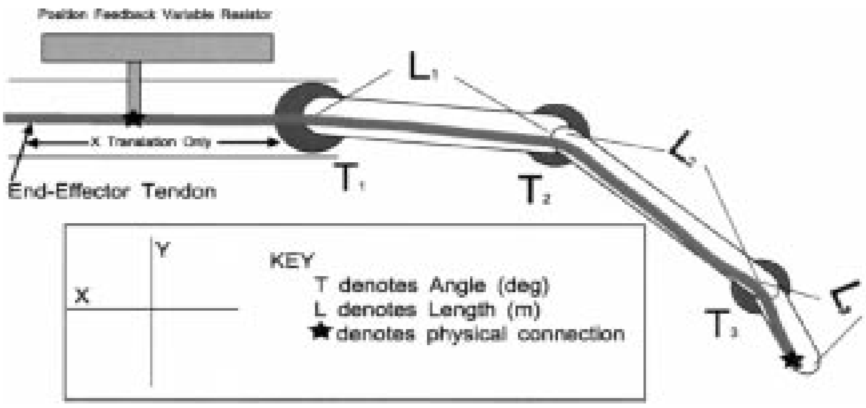

Figure 2.

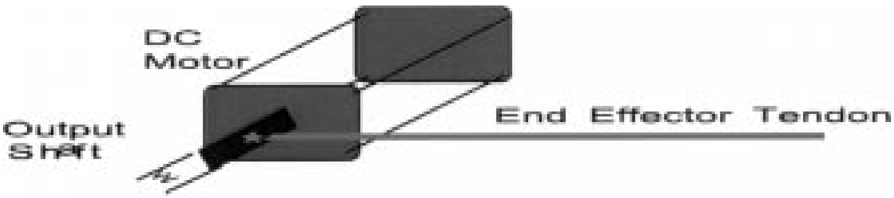

Figure 3.

for the control system. A +5 V potential is applied across this device and the voltage output is dependent upon the slide position. This voltage is sampled with the 8 bit D/A converter to provide integer numbers within the range of 0–255, where 0 represents full contraction and 255 represents full extension of each finger.

Each finger is actuated by a single dc motor arranged in the fashion shown in Figure 3.

2.3. MATHEMATICAL FORMULATION OF THE PROBLEM

Translation of any joint, in the robotic hand, will not occur until the following condition is obtained.

$$F_m > F_s.$$

Translation will occur only when the applied actuator force (F_m) is greater than the opposing force of static friction (F_s). Translation can be maintained through satisfying the following condition.

$$F_m \geqslant F_k.$$

The translation control problem exists at the point $F_m \sim> F_s$. The aspect of negative friction also affects the translation. Negative friction is the decrease in u_k (kinetic friction coefficient) due to an increase in rotational joint velocity ω and can be expressed through the following relation:

$$\operatorname*{Lim}_{\omega \to \omega_{\text{final}}} \Gamma(\omega, u_{\text{ki}}) = u_{\text{kf}}.$$

The identification of Γ is problematic and would require a very accurate value of u_{ki} (initial coefficient of kinetic friction). Such a value is not practical because very slow velocities would be needed to measure this term. The repeatability of these velocities is limited therefore poor values of u_{ki} would result.

Additional frictional terms can be introduced by considering the actuator dynamics. The actuator has frictional terms ε_{s} and ε_{k} (static and kinetic coefficients) and a pair of G_{s} and G_{k} terms which represent opposing static and kinetic frictional forces of the actuator. A Q function can also be used to represent negative friction of the motor

$$\operatorname*{Lim}_{\omega \to \omega_{\text{final}}} Q(\omega, \varepsilon_{\text{ki}})\varepsilon_{\text{kf}}.$$

The final frictional model for a joint in the robotic hand can be represented with the following model.

$$T_{\text{friction}} = F_{\text{s}} + F_{\text{k}} + G_{\text{s}} + G_{\text{k}} + \Gamma(\omega, u_{\text{ki}}) + Q(\omega, \varepsilon_{\text{ki}}).$$

Translation occurs when the following condition is satisfied

$$F_{\text{m}} \geqslant T_{\text{friction}}.$$

This model is highly nonlinear and any attempts at estimation of these parameters will yield in a incorrect, time consuming, pseudo-solution.

Linearization can be applied to this problem. The assumption will be made that the drift of the coefficient of kinetic friction for both the joint and the motor will be negligible. The second simplification that can be made is to assume that the motor torque (m_{t}) is much greater than its friction terms. These simplifications yield the following nonlinear model

$$T'_{\text{friction}} = F_{\text{s}} + F_{\text{k}} = f_{\text{s}}\text{sign}(\omega) + f_{\text{k}}\omega = T'_{\text{friction}}(f_{\text{s}}, f_{\text{k}}, \omega).$$

This nonlinear model is simplified by assuming certain operating constraints.

The goal of the control system is to achieve smooth joint translation regardless of the value of $T'_{\text{friction}}(f_{\text{s}}, f_{\text{k}}, \omega)$.

2.4. SIMULATED SOLUTION

The control problem is nonlinear in nature. Fuzzy logic control systems have been shown to provide good control of such nonlinear systems. Robustness issues still remain because a conventional fuzzy logic controller operates on a fixed set of rules that may not provide adequate control for all possible operating regimes.

The fuzzy logic controller that is proposed to solve the robustness issue is capable of regenerating its members based on a rule function. This rule function is divided into a group of non-overlapping blocks to provide a SISO control system. The input of this controller is joint position and the output is actuator response.

A normalized Gaussian function, bounded on an operating interval (p_i, p_f), was chosen to be the rule function for this controller. The parameter p_i represents the current position of the joint and parameter p_f represents the final desired position of the joint. The number of blocks (β) is calculated as $p_f - p_i$. For any given pair of p values, the Gaussian is bounded to exist only on that interval. The Gaussian is then divided into β blocks. The β value is calculated, on-line, as the joint translates. This value (assuming no external reactions) will continue to reduce to zero. A threshold value of $\beta = 30$ is used to maintain members within the fuzzy set throughout the translation period. Essentially, at the initial period of translation, many members exist and as translation occurs, the level of fuzziness increases.

The fuzzy control space is extended into a third dimension by multiplying each block by scaling factors. The scaling factors change values as the joint translates. The parameters affecting the scaling factors are joint velocity (ω), acceleration (α), rate of acceleration (α'), and position error (β).

Smooth motion is defined as follows

$$\left|\alpha'\right| \leqslant 1.$$

The control system also accepts a translation rate (r) input. This rate determines how quickly the joint will reach the final position (p_f).

The control system effectiveness can be observed by considering a translation example of a joint. The joint begins a $p_i = 0$ and the final desired position is $p_f = 220$ with a rate $r = 10$. Assuming no frictional terms, Figure 4 represents a simulation of the translation.

This plot represents the translation of the joint as a function of set re-generations. The initial β value was 220, therefore, 220 individual members (blocks of the Gaussian function) were generated. This high number of blocks allows for small changes in the control effort (motor). These small changes essentially mean a high control resolution similar to human control processes of lifting unknown weights. Typically, when lifting an unknown weight, a person will *pay* a lot of attention to his/her lifting effort at the beginning and at then end of the translation.

The simulation shown on the pervious page considers the translation of a joint under no external forces. The linearized model considers the frictional effects of the mechanical structure. The simulation was conducted for a static friction force requiring 15% of the total control effort and a kinetic friction force requiring 5% of the total control effort.

This simulation shows the translation response and the frictional changes as the control effort increases. The control effort increases to the point where translation occurs ($\sim$18 on X axis). At this point, the control system is capable of achieve smooth translation despite nonlinear frictional effects (see Figure 5).

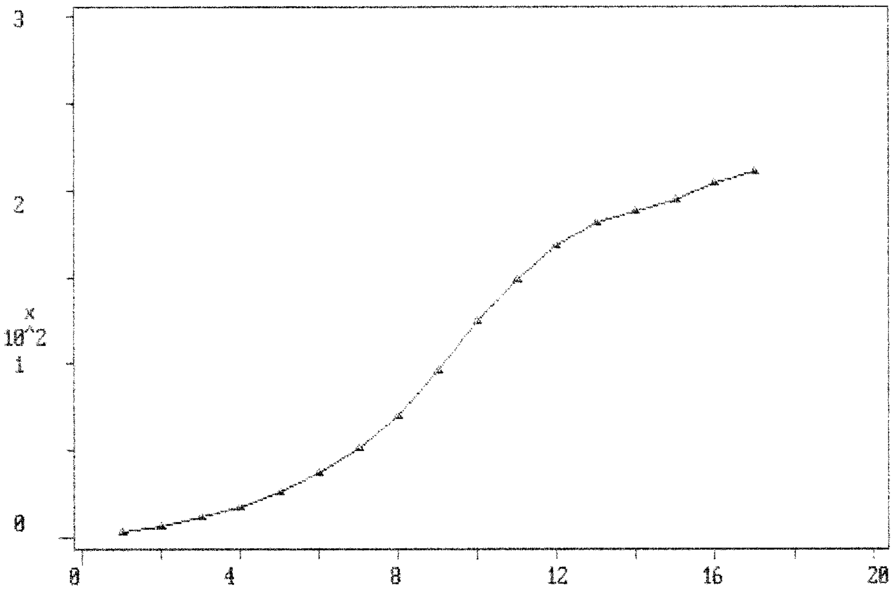

Figure 4.

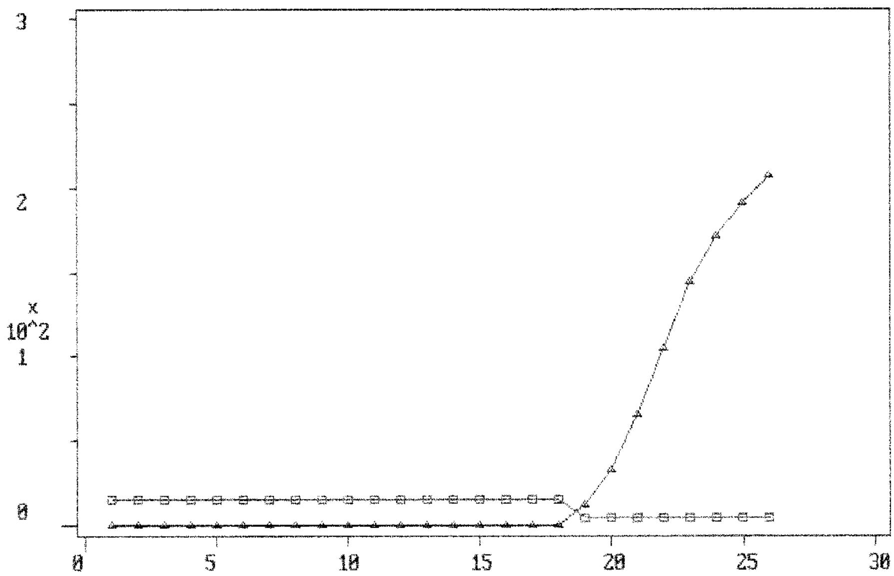

Figure 5.

This translation was simulated using rate values (r) ranging from 3 to 10. A rate of 10 represents the fastest mode and a rate of 3 represents a slow mode. The frictional forces were assumed to be the same. Figure 6 depicts the translation responses of the joint.

Figure 6 shows the translation responses for rate values ranging from 3 to 10. This plot is a representation of joint position vs. iteration. The simulation

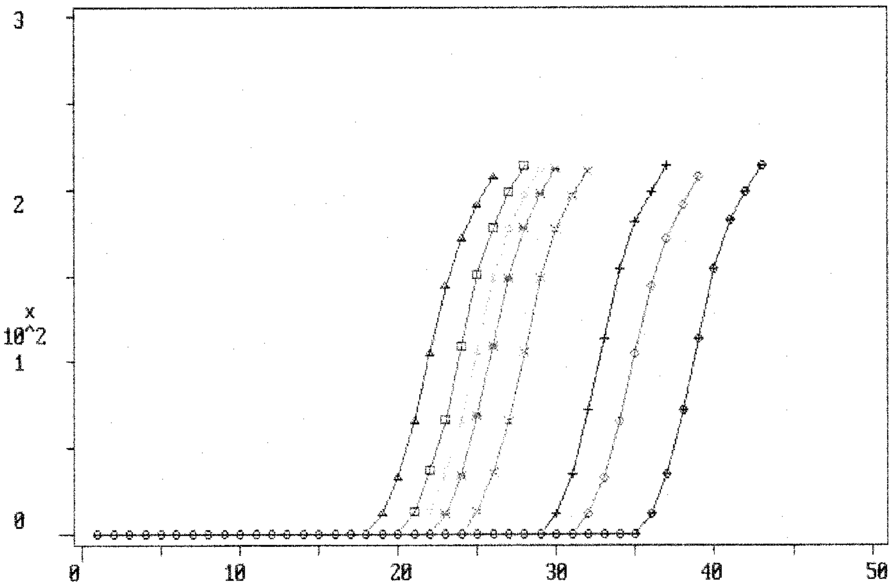

Figure 6.

graph shows that smooth translation occurs, despite frictional effects, at any of the selected rate values.

The translation time is a critical value for a practical application of this system. The translation time is depended upon the speed of the computer system that implements the controller. The simulations were conducted on a 486DX/25 Mhz and a Pentium 100 Mhz. The simulation time of the 486 was approximately 4 seconds while the simulation time on the Pentium was under 0.3 seconds. Practical applications of this system should be implemented on at least a Pentium 100 Mhz.

The effect of negative friction was also considered. A simulation was conducted for a frictional system. The conditions imposed were 20% of the total control effort to begin translation (static region) and 10% of the total control effort to maintain translation (kinetic region). The kinetic friction coefficient was forced to decrease 20% per iteration. Figure 7 shows the translation response. This plot represents a translation simulation of a frictional joint under negative kinetic friction. The simulation results confirm that smooth joint translation occurs in the presence of negative friction.

Figure 8 represents the translation between 18 and 27 iterations. This expanded plot shows that smooth translation is achieved at the price of fast translation when compared with the non-frictional model.

2.5. BLOCK REPRESENTATION OF THE CONTROL SYSTEM

Figure 9 represents the control system that was used to simulate the responses of the mechanical hand. The control system functions by determining the position

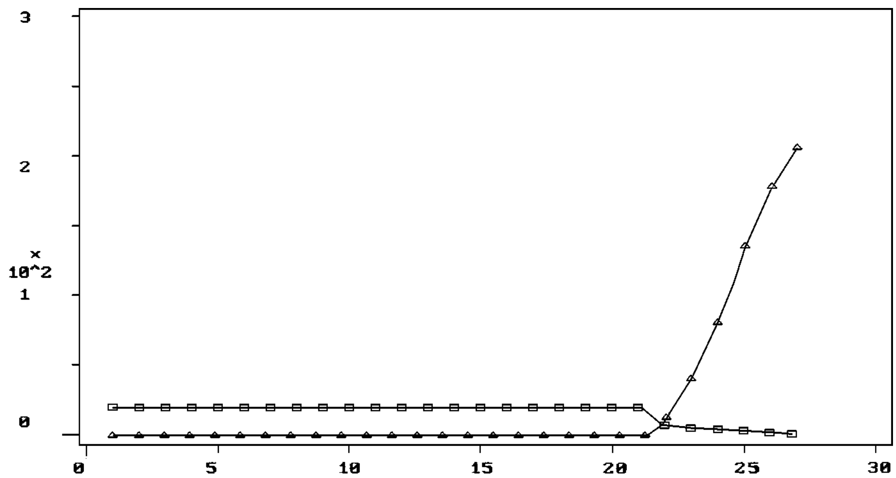

Figure 7.

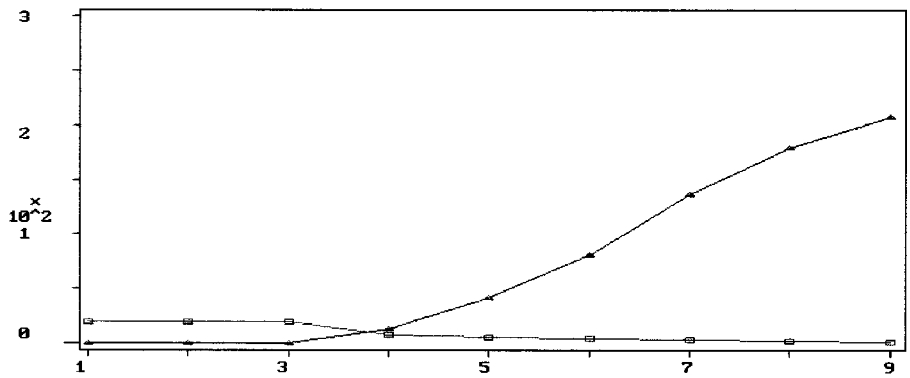

Figure 8.

error (E). This error signal is sent to a fuzzy scaler (conventional rule system) and the block generator that operates using the bounded normalized Gaussian. The fuzzy scaler generates the primary scaling factor. The values of ω, α, α' are derived from the position input. These values are received by the functional scaler. This device uses logical operations to adjust a secondary scaling factor. The primary and secondary scaling factors are multiplied within the Functional Event Block. This value is then multiplied by the output of the fuzzy block generator to produce a signal for the actuator. The actuator then drives the joint, which in turn drives the position encoder and provides a current position feedback.

2.6. RATE GENERATOR

A feedforward rate generator was used for on-line adjustment of the commanded rate (r). This method required that the user provide only the final position and

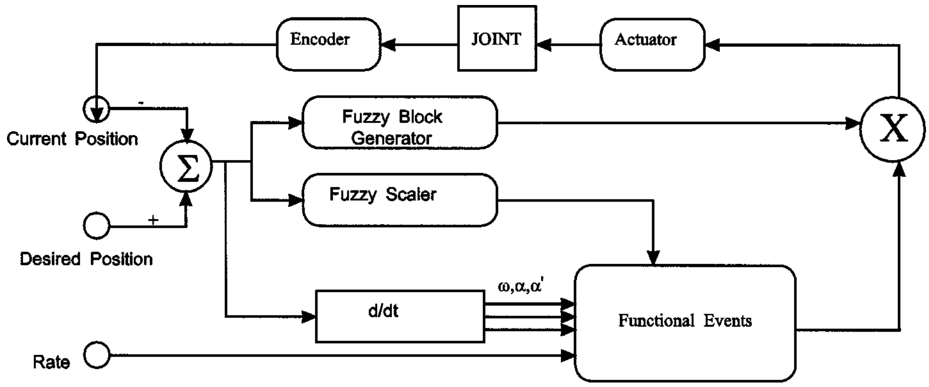

Figure 9.

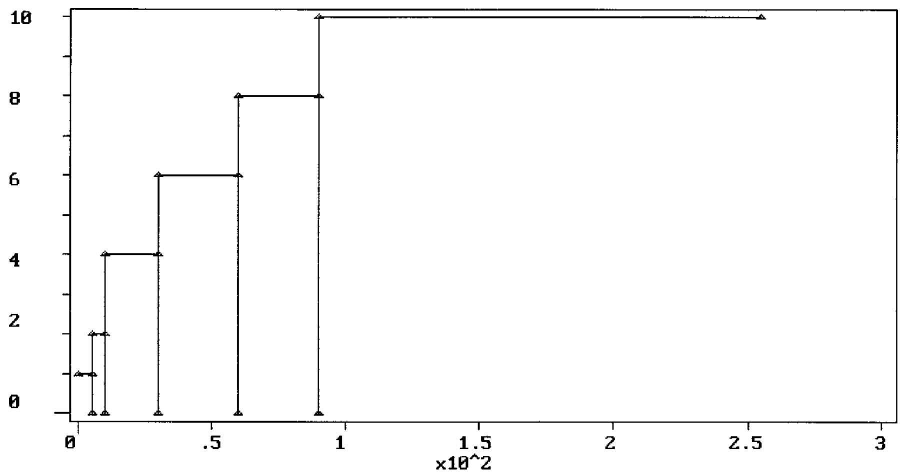

Figure 10. Rate generator transfer function (r vs. E)

allowed the system to perform tracking control with respect to a time varying external reference signal. The rate generator was based on the method of fuzzy blocks. The transfer function of the rate generator is shown at the top of the column of Figure 10.

3. Fuzzy Block Control Method

The control problem of a mechanical hand is highly nonlinear in nature. Fuzzy logic has been shown to provide excellent control of nonlinear systems. Robustness issues still remain because a conventional fuzzy logic controller operates on a fixed set of rules that may not provide adequate control for all possible operating regimes.

The fuzzy logic controller that is proposed to solve the robustness issue is capable of regenerating its members based on a rule function. This rule function is divided into a group of discrete blocks to provide a SISO control system. The input of this controller is joint position and the output is actuator response.

A normalized Gaussian function, bounded on an operating interval (p_i, p_f), was selected as the rule function for this controller. The parameter p_i represents the current position of the joint and parameter p_f represents the final desired position of the joint. The number of blocks (β) is calculated as $p_f - p_i$. For any given pair of p values, the Gaussian is bounded to exist only on that interval. The Gaussian is then divided into β blocks. The β value is calculated, on-line, as the joint translates. This value (assuming no external reactions) will continue to reduce to zero. A threshold value of $\beta = 30$ is used to maintain members within the fuzzy set throughout the translation period. Essentially, at the initial period of translation, many members exist and as translation occurs, the level of fuzziness increases.

The fuzzy control space is extended into a third dimension by multiplying each block by scaling factors. The scaling factors change values as the joint translates. The parameters affecting the scaling factors are joint velocity (ω), acceleration (α), rate of acceleration (α'), and position error (β).

The control system also accepts a translation rate (r) input. This rate determines how quickly the joint will reach the final position (p_f).

4. Experimental Results

The first experiment testing a gross translation between the mechanism limits of the hand. This translation was between 48 units and 180 units. The trajectory is shown in Figure 11.

The system response settled to within 1 unit of the desired final position (180) in $T_{ss} = 2.8$ s. The error of 1 unit is acceptable because the resolution of the A/D converter (ADC0809) is ± 1 bit that corresponds to ± 1 unit.

The second experiment was performed to test the translation response from an initial position of 155 to a final position of 80. The trajectory is shown in Figure 12.

The trajectory of the second experiment shows that the system settling time was $T_{ss} = 4.5$ s. The steady state deviations of ± 1 unit are due to A/D sampling errors and possibly circuit noise. No overshoot was observed in the response trajectory.

The fourth experiment was to command a fine translation between 91 and 85 units (see Figure 13). This trajectory shows that the system converged to 85 units in approximately 3.2 s. No overshoot was observed in this response curve. This fine motor translation example shows that this system can produce precision movements needed for dexterous manipulation tasks.

A fifth experiment tested a translation similar to the previous example. The task was to translate the finger from 90 units to 95 units (see Figure 14). The control system was capable of achieving fine motor control with no overshoot in the response. The settling time of this system was $T_{ss} = 2.2$ s. No steady state error was observed.

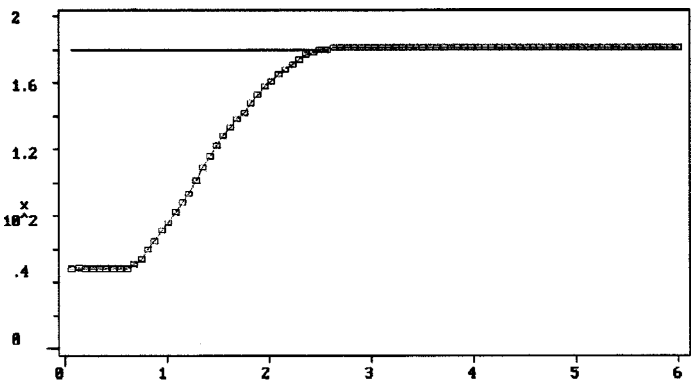

Figure 11.

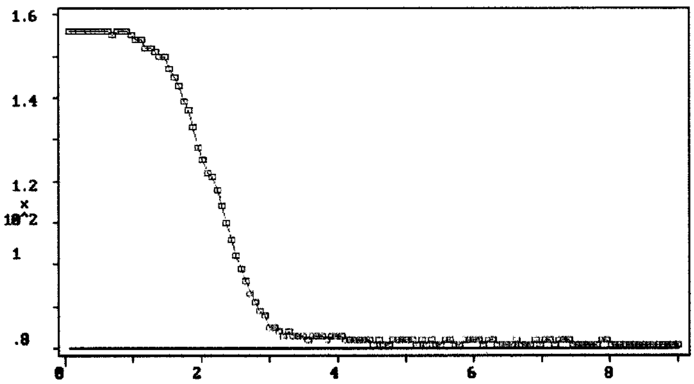

Figure 12.

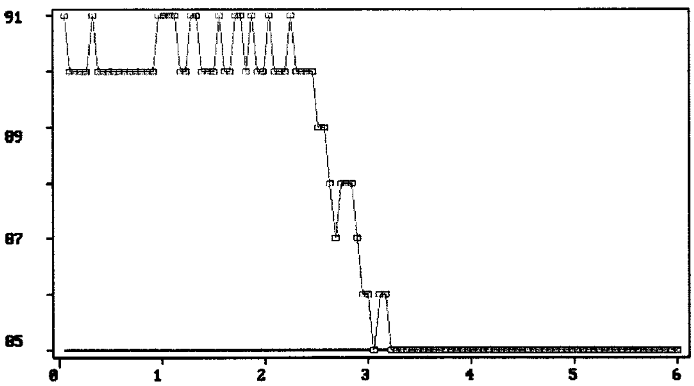

Figure 13.

The sixth experiment performed another fine motor translation between 171 units and 178 units. This experiment was used to test the response characteristics near the mechanical limits of the robotic hand (see Figure 15). This experiment confirmed that the control system was capable of fine motion near the mechanical limits of the system. Many nonlinearities, such as mechanism bending, exist at

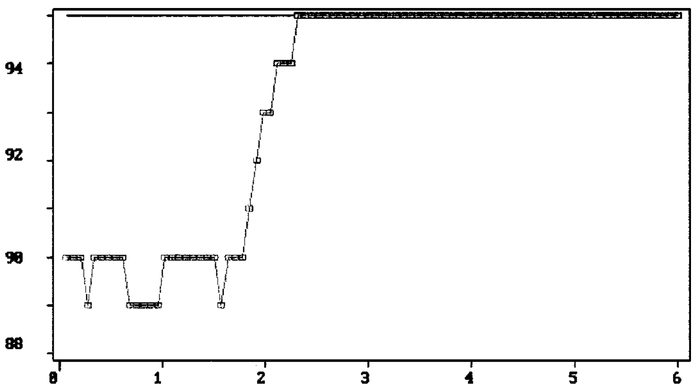

Figure 14.

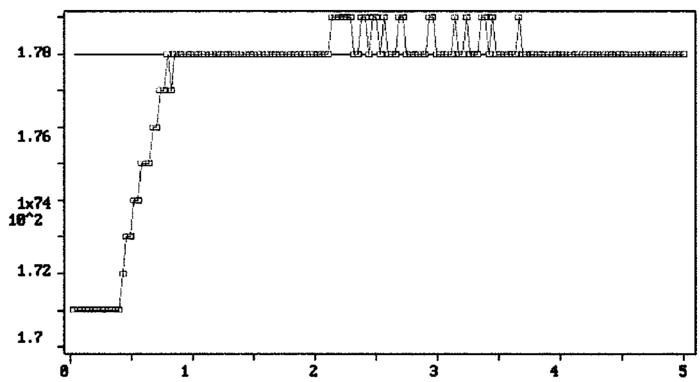

Figure 15.

these limits. The response settled at $T_{ss} = 1$ s and the deviations were due to sampling errors.

4.1. TRACKING CONTROL RESULTS

A key area in robotics research is tracking control (i.e., duplicating human motions with a machine). The remaining three experiments investigated tracking control examples. An external potentiometer was sampled to provide the reference position value for the control system.

Figure 16 represents the first tracking experiment. The solid line is the source signal and the line composed of squares is the system response. Observation of this plot will immediately show a very high correlation between the source and response paths. The response trajectory did not overshoot the source signal at any point. An analysis showed that the cross-correlation coefficient between these two data sets was $R_{sr} = 0.96$. The response characteristics were calculated to be within 99% of the source signal. This experiment showed that the control system performs well when tracking a refereed input.

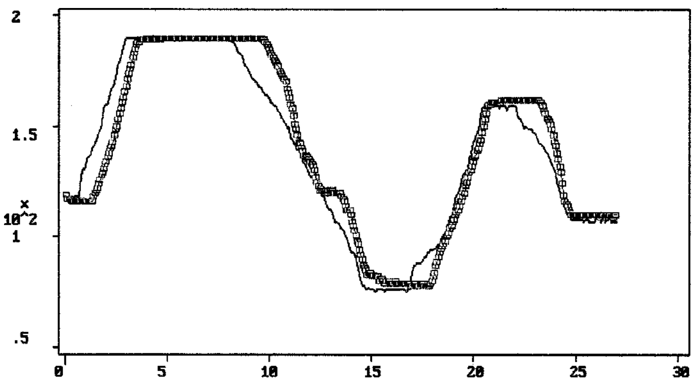

Figure 16. Tracking experiment #1.

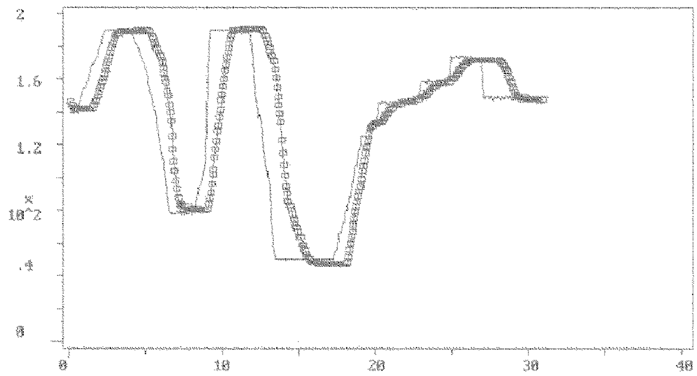

Figure 17. Tracking experiment #2

This experiment showed that the system response (square line) was capable of tracking a combined pulse and step source signal. Minimal deviations, most likely due to sampling errors, were observed in the response curve. These deviations were responsible for the small overshoots shown in the plot. Data analysis procedures yielded a cross-correlation coefficient of $R_{sr} = 0.81$. The system significance was calculated to be at least 99%. This experiment examined one of the difficult aspects of tracking control: response to a variable step input (see Figure 18). The source (solid line) consists of large and small step displacements. These displacements require fast acceleration changes in order to provide adequate trajectory tracking. The system response (square line) followed the source input, but some overshoots were present. These overshoots are caused by the large position variations in the source signal. The maximum overshoot was determined to be only 5 units. Data analysis showed that the cross-correlation coefficient was $R_{sr} = 0.8$ the system significance was 99%.

This experiment showed that the control system could provide trajectory tracking for a variable step source signal with an overshoot of less than 5 units.

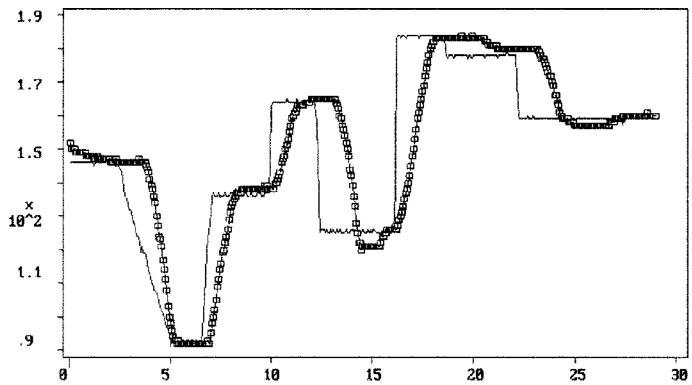

Figure 18. Tracking experiment #3.

5. Conclusions and Future Work

This paper presented the design, implementation and the simulated and experimental results of a multi-finger robotic hand using the fuzzy block control scheme. The hand was controlled using an IBM compatible computer via a custom designed acquisition/output interface. The hand mechanism consisted of five fingers in a progressive link arrangement.

The control system shows good potential of operating a mechanical hand under dynamic (or non-dynamic) friction.

The simulation results show that smooth translation of the joint occurs for the non-frictional model, fixed static and kinetic frictional model, and the fixed static with negative kinetic friction model. The control system is able to eliminate erratic translations, which affect other control attempts, due to frictional changes.

Enhancements need to be made in the functional event block. This block provides the control system with the ability to monitor fast accelerations in the joint dynamics and aids in adjusting the control effort to compensate for these unwanted conditions. The final two simulations (fixed friction and negative friction) showed that smooth translation was achieved, however, the translation curve is approaching a linear path. This linear path is dissimilar to human motion. The solution to this problem lies in optimization of the internal coefficients that operate in the functional event block. Optimization will enhance the control system performance by generating the best overall coefficients.

The simulations show that the control system is effective in reducing the effects of friction on the joint translation. On-line optimization of the functional coefficients should be applied to further increase the effectiveness of this system.

The control system was based on the concept of fuzzy-like logic. A fuzzy-like scheme, referred to as the method of blocks, was developed for the control experiment. This method is capable of regenerating the members in the fuzzy set by using a Gaussian rule function.

The experimental results conclusively showed that this method was capable of translating the fingers between any two arbitrary joints. The maximum steady state error observed was 1 unit (0.273 degrees). This system also performed excellent in tracking control experiments. The tracking control experiments demonstrated that this system could be used to duplicate human motion via an “input glove” fitted onto the human user. This capability would allow a robotic system to be designed to perform human tasks in a hostile environment.

The results of this method are very positive. The method of blocks was shown to be quite successful at controlling both fine and gross finger translations as well as tracking control. The next phase of this design is to implement high-resolution tactile sensors. These sensors will allow texture and pressure feedback that can be used for force control using a similar block method.

The control software, interface, and mechanical assembly produced high resolution translations in low cost. The total development cost, to date, is approximately $250 (US). Finally, the method presented here is cheap, reliable, for an accurate robotic manipulation system.

5.1. EXPERIMENTAL APPARATUS

The multi-finger hand with one actuator is shown in Figure 19. The I/O interface for robotic hand is shown in Figure 20.

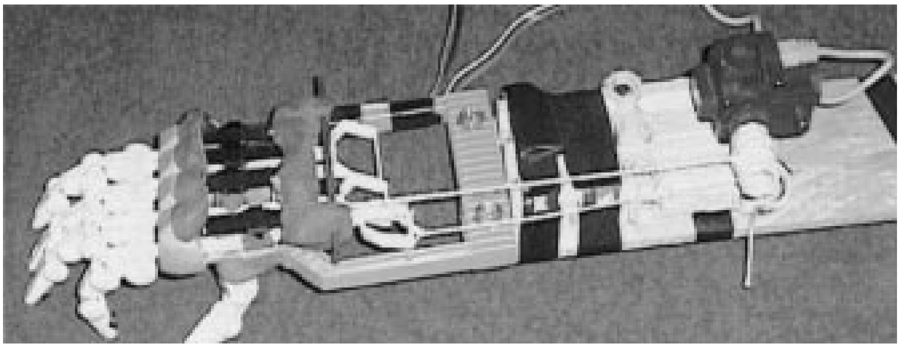

Figure 19.

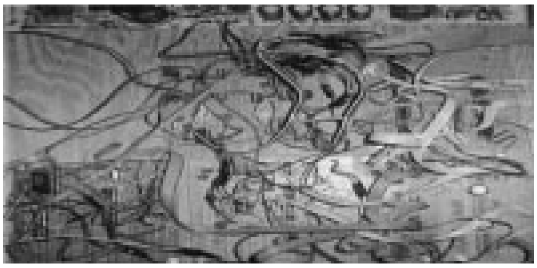

Figure 20.

Acknowledgement

This work is partially supported by an FRG grant 1997-98.

References

1. Bicchi, A.: Hands for dexterous manipulation and powerful grasping: A difficult road towards simplicity, in: *2nd Internat. Symposium on Robotics Research*, MIT Press, Cambridge, MA, 1985.
2. Dorsam, Th., Hammerschmidt, O., and Munoz, L. A.: A study on multifinger gripper control, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, April 1997, pp. 1112–1117.
3. Eggebrecht, L. C.: *Interfacing to the IBM Personal Computer*, H.W. Sams, 1983.
4. Ghausi, M. S.: *Electronic Devices and Circuits: Discrete and Integrated*, CBS College Publishing, 1985.
5. Hirzinger, G., Koeppe, R., Baader, A., Lange, F., Staudte, R., and Wei, G.-Q.: Perception and manipulation in robotics: Neural network approaches, in: *2nd Internat. Symposium on Robotics Research*, MIT Press, Cambridge, MA, 1985.
6. Hsu, Y.-C., Chen, G., and Sanchez, E.: A fuzzy PD controller for multi-link robot control: Stability analysis, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, April 1997, pp. 1112–1117.
7. Kircanski, M. and Kircanski, N.: Resoved-rate and resolved-acceleration-based robot control in the presence of actuator's constraints, in: *Proc of IEEE Internat. Conf. on Robotics and Automation*, April 1997, pp. 235–239.
8. Kumbla, K., Moya, J., and Baird, R.: Fuzzy control of robotic manipulator, in: *Fuzzy Logic Control: Software and Hardware Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1993.
9. Lea, R., Hoblit, J., and Jani, Y.: Fuzzy logic based robotic arm control, in: *IEEE Technical Activities Board: Fuzzy Logic Technology and Applications*, IEEE Press, New York, 1994.
10. Sicilano, B.: Parallel force/position control of robot manipulators, in: *2nd Internat. Symposium on Robotics Research*, MIT Press, Cambridge, MA, 1985.
11. Yoshikawa, T. and Sudou, A.: dynamic hybrid position/force control of robot manipulators: On-line estimation of unknown constraint, in: *1st Internat. Symposium on Experimental Robotics*, New York, 1990.
12. Wakerly, J. F.: *Digital Design*, Prentice-Hall, Englewood Cliffs, NJ, 1994.
13. Zand, M., Torab, P., and Bahri, A.: Hybrid position/force control of a dexterous hand based on fuzzy control strategy, in: *Proc. of IEEE Internat. Conf. on Robotics and Automation*, July 1997, pp. 133–139.