Fuzzy Logic Control of a Multifingered Hand Robot Using Genetic Algorithm Based on DSP

Elham Ataei\(^1\), Rouhollah Afshari\(^2\), Mohammad Ali Pourmina\(^1\)

**Abstract** – This paper presents the fuzzy control of a five fingered robot hand by using the genetic algorithm and TMS320F2812 chip. Because of its high-speed performance, its support for multi-motor control and its low power consumption, TMS320F2812 DSP from TI demonstrates itself as an ideal candidate for the five fingered robot hand.

There are five fingers and each finger has three degrees of freedom (DOF). All the actuators and electronics are integrated in the finger body and the palm. This robot has five fingers in which all joints are driven by DC motors. At the same time, the multisensory robot hand integrates position, force/torque and temperature sensors. The fuzzy controller dealing with the level of finger control is a multiple-input-multiple-output (MIMO) fuzzy learning controller and is implemented in AVRs microcontrollers. In this paper, the design of the fuzzy controller is based on the genetic algorithm. The whole weight of the hand is about 1.3Kg and the fingertips force can reach 8N.

**Keywords**: Multifinger, Fuzzy Logic, Genetic Algorithm, Hand Robot, DSP

---

**Nomenclature**

DOF = Degrees of freedom  
FLC = Fuzzy Logic Control  
GA = Genetic Algorithm  
DSP = Digital Signal Processor  
PID = Proportional–Integral–Derivative  
E = Error  
\( \theta_r \) = Reference Position  
\( \theta_a \) = Actual Position  
ec = change in errors  
U = Armature Voltage

---

**I. Introduction**

The five finger robot hands play an ever important role in the service robots and other challenge areas. Recently, various robot hands have been developed so far. Some nice robot hands have been built in the labs and companies, such as the NASA Robonaut Hand [1] the shadow Hand [2] the DLR Hand II [3] and the DLR/HIT Hand I [4]. Generally there are two kinds of hand, one is external actuation hand, where all the actuators are mounted in the forearm (NASA and Shadow), and another internal actuation hand (DLR, HIT), where there needs not any forearm and all the actuators and electronics are integrated in the finger body and the palm. Normally the internal actuation hand body is bigger than the external actuation hand.

The external actuation hands are driven by using the tendon cables. The elasticity of the tendon cable causes inaccurate joint angle control, and the long wiring of tendon cables may obstruct the robot motion when the hand is attached to the tip of the robot arm. Moreover, these hands have many problems on the product and the maintenance because it’s mechanical complexity. To solve these problems, robot hands in which the actuators are built into the hand (e.g., the Belgrade/USC hand by Venkataraman et al. [5], the Omni hand by Rosheim [6], the NTU hand by Lin et al. [7], and the DLR’s hand by Liu et al. [8]) have been developed. However, these hands have a problem in that their movement is unlike that of the human hand because the number of fingers and the number of joints in the fingers are insufficient. The designed robot is a five-fingered hand driven by built-in DC motors. The control of the multi-fingered robot hand is very complicated [9]. The reason is that the degrees of freedom (DOF) of the multi-fingered robot hand are too many and the dynamics of the hand are highly nonlinear and coupled. Once the tactile sensors are introduced, it becomes a much more complex control problem that deals with both position and force.

Since Zadeh’s paper on fuzzy set [10], fuzzy control has become one of the most active and fruitful areas in the applications of fuzzy set theory. Most fuzzy logic control relies on the operators’ experiences to design the knowledge bases since the pioneering work of Mamdani [11]. The adaptive fuzzy system equipped with a training algorithm is also applied to the control problem [12]-[14]. Due to the self-organized mechanism, the adaptive fuzzy control can resolve system uncertainty with less heuristic information. In the research of Bekey [15], a knowledge-based planner was proposed to select gasping postures by reasoning from symbolic information of the target object geometry and the nature of the task. However, the knowledge about grasping in the sense of control was not discussed.

Recently, the design of FLC has also been tackled with...
genetic algorithm (GA). These are optimization algorithm performing a stochastic search by iteratively processing ‘populations’ of solutions according to fitness [20], [22]. In control applications, the fitness is usually related to performance measures as integral error, setting time, etc. GA based FLC have been used in induction motor control system design successful [21], [24].

In this paper, DSP-based FLC of a five finger robot is implemented for force/position control on a fuzzy logic microcontroller using genetic algorithm. Heuristic knowledge is applied to define fuzzy membership functions and rules. The membership functions and rules are modified after initially knowledge from a PID controller developed from a simple linear model [16], [17], [23]. The hardware interface and the software algorithm are described.

II. Control System of Hand Robot

The multi fingered robot hand, which is designed and fabricated in our laboratory, has five fingers with fifteen degrees of freedom (DOF). All fingers have three joints. Each finger is equipped with tactile sensors to detect grasping force. The tactile sensors of the hand are attached to the inner sides of finger segments and the palm to detect the contact force, as shown in Fig. 1.

![Fig. 1. The designed hand](image)

We provide control architecture with a special designed hardware to control fifteen finger joints of the hand robot. The control system, as shown in Fig. 2, consists of a host computer, DSP board, microcontroller board, motors driver interface and sensors interface.

The DSP board contains a TMS320F2812 DSP chip and peripheral devices to perform the fuzzy control in terms of sensor data and commands from the host computer, and then distributes commands to each microcontroller’s board. Each microcontroller’s board contains an ATmega8 Atmel processor and motor driver interface. These boards are PID controllers. It performs the joint space control for each finger.

![Fig. 2. The hardware and control system of hand robot](image)

III. Fuzzy Control

Fuzzy logic provides an approximate effective mean of describing the behavior of some complex system. Unlike traditional logic type, fuzzy logic aims to model the imprecise modes of human reasoning and decision making, which are essential to our ability to make rational decisions in situations of uncertainty and imprecision.

While conventional controllers depend on the accuracy of the system model and parameters, FLCs use a different approach to control the position of DC motors. Instead of using a system model the operation of FLC is based on heuristic knowledge and linguistic description to perform a task. The effects from inaccurate parameters and models are reduced because a FLC does not require a system model. However, building a FLC from the ground-up may not provide good results or sometime even a worse result than a conventional controller if there is not enough knowledge of the system. Using GA to optimize the fuzzy controller and adjusting the rules and membership functions are described as follows.

III.1. Define inputs, outputs

To apply heuristic knowledge in the FLC, inputs and outputs are defined first. The most significant variables entering the fuzzy logic controller have been selected as the position error and its time derivative. The inputs are the errors (e) between the reference (θ_r) and actual position (θ_a), and the change in errors (e_c). The outputs are the change in armature voltage (U). The inputs and outputs illustrated in Fig. 3 are described by:

\[ e = e(k) = \theta_r(k) - \theta_a(k) \]  
\[ e_c = e(k) - e(k-1) \]  
\[ U = u(k) - u(k-1) \]

Where k is the time index.

![Fig. 3. Block diagram of the FLC](image)
III.2. Defining fuzzy membership function and rules

To perform fuzzy computation, the inputs and outputs must be converted from numerical or “crisp” value into linguistic forms. The terms such as “Small” and “Big” are used to quantize the inputs and outputs value. In this paper, the linguistic terms that used to represent the inputs and outputs value are defined by seven fuzzy variables as shown in Table I.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB</td>
<td>Positive Big</td>
</tr>
<tr>
<td>PM</td>
<td>Positive Medium</td>
</tr>
<tr>
<td>PS</td>
<td>Positive Small</td>
</tr>
<tr>
<td>ZE</td>
<td>Zero</td>
</tr>
<tr>
<td>NS</td>
<td>Negative Small</td>
</tr>
<tr>
<td>NM</td>
<td>Negative Medium</td>
</tr>
<tr>
<td>NB</td>
<td>Negative Big</td>
</tr>
</tbody>
</table>

Each fuzzy variable is a member of the subsets with a degree of between 0 (non member) and 1 (full member) as:

\[
\mu_A(x) = \begin{cases} 
1 & \text{if } \mu_A \in A \\
0 & \text{if } \mu_A \notin A 
\end{cases} \quad (4)
\]

A fuzzy membership function can contain several fuzzy sets depending on how many linguistic terms are used. Each fuzzy set represents one linguistic term. In this paper seven fuzzy sets are obtained by applying the seven linguistic terms. The member for indicating how much a crisp value can be a member in each fuzzy set is called a degree of membership. One crisp value can be converted to be “partly” in many fuzzy sets, but the membership degree in each fuzzy set may be different. In order to define fuzzy membership function, designers can choose many different shapes based on their preference or experience. The popular shapes are triangular and trapezoidal because these shapes are easy to represent designer’s ideas and require low computation time. For performing fine-tuning to improve the efficient of the controller, the adjacent of each fuzzy set value should overlap about 25% [18]. The initial membership functions are illustrated in Fig. 4.

To send out the armature voltage output, the output in the form of fuzzy sets must be converted to a crisp value. This process is called defuzzification. In this paper, the center of gravity method is chosen. The formula of this method is:

\[
z = \frac{\sum_{i=1}^{a} S_i F_i}{\sum_{i=1}^{a} F_i} \quad (5)
\]

Where \( z \) is the output from defuzzification, \( S_i \) is the specific position at the \( i^{th} \) fuzzy set, and \( F_i \) is the membership degree at the position. [19].

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>THE FUZZY LINGUISTIC RULES TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>( e_c )</td>
</tr>
<tr>
<td>NB</td>
<td>PB</td>
</tr>
<tr>
<td>NM</td>
<td>NB</td>
</tr>
<tr>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>ZE</td>
<td>PB</td>
</tr>
<tr>
<td>PS</td>
<td>PM</td>
</tr>
<tr>
<td>PM</td>
<td>PB</td>
</tr>
<tr>
<td>PB</td>
<td>NB</td>
</tr>
<tr>
<td>NS</td>
<td>NB</td>
</tr>
<tr>
<td>NM</td>
<td>NB</td>
</tr>
</tbody>
</table>

IV. Genetic Algorithm

GA is a stochastic optimization algorithm is originally motivated by the mechanisms of natural selection and evolutionary genetics. The GA serves, as a computing mechanism to solve the constrained optimization problem resulting from the motor control design where the genetic structure encodes some sort of automation. The basic element processed by a GA is a string formed by concatenating substrings, each of which is a binary coding (if binary GA was adopted) of a parameter. Each string represents a point in the search space.

The Selection, Crossover and Mutation are the main operations of GA.
Selection directs the search of GA toward the best individual. In the process, strings with high fitness receive multiple copies in the next generation while strings with low fitness receive fewer copies or even none at all.

Crossover can cause to exchange the property of any two chromosomes via random decision in the mating pool and provide a mechanism to product and match the desirable qualities through the crossover.

Although selection and crossover provide the most of the power skills, but the area of the solution will be limited. Mutation is a random alternation of a bit in the string assists in keeping delivery in the population.

The optimization step of GA is follow:

1. Code the parameter
2. The initialization of the population
3. Evaluate the fitness of each member
4. Selection
5. Crossover
6. Mutation
7. Go to step 1 until find the optimum solution.

V. GA based Fuzzy Controller

Since the fuzzy inference is time-consuming, and the DSP used in motor control is speed-limited, so real-time inference method cannot be chosen. Here by using the synthetic fuzzy inference algorithm, the computer makes a query table off-line in advance and stores it in the memory of DSP. In a practical control, the control value can be obtained according to the query table, and tuning the control coefficients (K_e, K_{ec}, and K_U) on-line.

The design of the fuzzy controller is base on the genetic algorithm. Fig. 5 shows the coding formulation when using GA to optimize the fuzzy controller. Here using 10 bits binary code to denote one fuzzy inference rule. The first binary code is the flag whether the rule is used. The 2-4, 5-7 and 8-10 refer to the error, change in error and the output variable. And 001, 010, 011, 100, 101, 110 and 111 refer to NB, NM, NS, ZE, PS, PM and PB respectively.

For example, the first rule of Fig. 5 shows that if e is PB and e is NB then U is PM, and the fist bit binary code ‘0’ indicate that this rule will be eliminated through optimization.

![Fig. 5. Coding method for GA](image)

In order to improve the speed of the optimization, this paper chooses 30 candidates as the initialization population, and these candidates are proved to be able to make the motor run steadily. Table III shows the parameter of GA used in this paper, and Table IV shows the fuzzy rules intimidated through GA method. Through the optimization, 6 rules are eliminated and 4 rules are optimized.

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>THE PARAMETERS OF GENETIC ALGORITHM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossover Possibility</td>
<td>0.85</td>
</tr>
<tr>
<td>Mutation Possibility</td>
<td>0.002</td>
</tr>
<tr>
<td>Generation Numbers</td>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE IV</th>
<th>THE LINGUISTIC RULE TABLE AFTER OPTIMIZATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>e_c</td>
</tr>
<tr>
<td>NB</td>
<td>PB</td>
</tr>
<tr>
<td>NM</td>
<td>PM</td>
</tr>
<tr>
<td>NS</td>
<td>PB</td>
</tr>
<tr>
<td>ZE</td>
<td>PB</td>
</tr>
<tr>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>PM</td>
<td>PM</td>
</tr>
<tr>
<td>PB</td>
<td>PM</td>
</tr>
</tbody>
</table>

VI. On-Line Tuning

In order to improve the dynamic performance of the multifingered hand robot, the elements of the query table need to be adjusted according to the input variables. To do this, the paper adjusts the coefficients (K_e , K_{ec} and K_U) to tuning the control system on-line.

The basic principle is the “rough adjustment” and “accurate adjustment”, namely, constantly adjusting the coefficients according to actual e, e_c. If the e and e_c are large, K_e and K_{ec} should be reduced while K_U should be increased because the main objective is diminishing the errors. When e and e_c are small, because the main aim is to diminish the overshoot and steady-state error, K_e and K_{ec} should be increased to increase the resolution of e and e_c while K_U should be reduced to obtain small control value to reduce the overshoot and steady-state error. The adjust function as follow

\[
K_e = \begin{cases} 
K_{e0} + K_1 \times e, & |e| \leq \frac{e_{max}}{2} \\
K_{e0} + K_1 \times \frac{e_{max}}{2}, & |e| > \frac{e_{max}}{2} 
\end{cases} \tag{6}
\]

\[
K_{ec} = \begin{cases} 
K_{ec0} + K_1 \times e, & |e| \leq \frac{e_{max}}{2} \\
K_{ec0} + K_1 \times \frac{e_{max}}{2}, & |e| > \frac{e_{max}}{2} 
\end{cases} \tag{7}
\]

Copyright © 2008 Praise Worthy Prize S.r.l. - All rights reserved
The feature work will be received on control theory: A...
Scientific, Educational & Researches institute from 2002 until now in the Research and Development (R&D) department. In 2006 he joined the electrical engineering department of Islamic Azad University (Abhar Branch) as a lecturer. He registered 8 patents during 2007-2011. Up to now he has had several outstanding positions such as “Director of the Robotic Groups of Zanjan University”, “Director of the Robotic Groups of Islamic Azad University, Abhar Branch”, “Director of the Robotic Groups of Zanjan PayamNoor University”. Currently, he is working on Solar and Bio Mass Power Plants at R&D Department, Atrin Parsian, Tehran, Iran. His research interests include robot control, humanoid robot hand, swarm bots and Solar Power Plants.

Mohammad Ali Pourmina received his B.S. and M.S. degrees in Electrical engineering (Telecommunication) from Islamic Azad University, Central branch, Tehran, Iran, in 1989 and 1991 respectively. He received his Ph.D. degree at Islamic Azad University, Science and research Branch, Tehran, Iran, in 1996. He joined Islamic Azad University, Science and research Branch, Tehran, Iran in 1996. He has been a member of Iran Telecommunication research center since 1992. He has performed research in the areas of packet radio networks and digital signal processing systems since 1991. His current research interests include spread-spectrum systems, cellular mobile communications, indoor wireless communications, DSP processors and wireless multimedia networks.