**Genetic Optimization of Fuzzy Logic Control for Coupled Dynamic Systems**

**Submitted To**

**The 2014 Academic Year NSF AY-REU Program**

**Part of**

**NSF Type 1 STEP Grant**

**Sponsored By**

**The National Science Foundation**

**Grant ID No.: DUE-0756921**

**College of Engineering and Applied Science**

**University of Cincinnati**

**Cincinnati, Ohio**

**Prepared By**

**Andrew Janson, Senior, Computer Science**

**Nicklas Stockton, Sophomore, Aerospace Engineering**

**Report Reviewed By:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Dr. Kelly Cohen**

**REU Faculty Mentor**

**University of Cincinnati**

**January 6 – April 18, 2013**

Genetic Optimization of Fuzzy Logic Control for Coupled Dynamic Systems

ANDREW JANSON1, NICKLAS STOCKTON1, KELLY COHEN1

1 College of Engineering and Applied Science, University of Cincinnati, Cincinnati, Ohio

ABSTRACT: This research project, in the field of control systems, was funded by the National Science Foundation through the Research Experience for Undergraduate (REU) students. The objective of this project was to combine the robustness of fuzzy logic control with the adaptability of genetic algorithms to produce a self-optimizing oscillation damping control mechanism. Once an initial fuzzy inference system (FIS) is developed by an expert for a given dynamic system, the genetic algorithm will be able to optimize the FIS for a range of similar systems with varying parameters. In order to evaluate the control mechanisms developed during this project, a simulation of a two cart spring-mass system was developed in MATLAB. The performance of the controllers was determined by how quickly it could approach a wall and how close it was able to settle the car system to the wall without crashing. The membership functions of the FIS were reduced to an array of real-valued parameters in order to be used in a genetic algorithm. Once the genetic representation of the FIS was defined, the selection, reproduction, and mutation methods were developed to complete the genetic algorithm. The best solution developed by the genetic algorithm was evaluated against the hand-tuned solution developed in the first phase of the project. In order to simulate varying parameters between similar dynamic systems, the mass of the car system in the simulation was varied from 3kg to 20kg. For each weight change the genetic algorithm was allowed to re-optimize the parameters of the FIS. The performance of the genetic algorithm, with respect to the theoretical best, varied up to 17%, while the unmodified FIS varied up to 300%. These results show that genetic algorithms are necessary to allow fuzzy control mechanisms to adapt to different systems with little external input from a general user.

INTRODUCTION

Fuzzy logic systems are used as control systems which are capable of handling the vagueness of the real world. Fuzzy logic can model and control nuances overlooked by the binary logic of conventional computers [1]. In fuzzy logic, the truth of any statement becomes a matter of degree. Take for example, determining the timespan of your weekend [2]. Binary logic only allows a yes/no answer to the question: “Is this day part of my weekend?” However, in the real world we do not consider exactly 12am Saturday to be the start of our weekend. We usually view most of Friday as a part of the weekend, determined by when we decide to quit working. Fuzzy logic allows us to specify that most of Friday is considered the weekend as well by assigning it a value of less than one but greater than zero. This distinction gives the fuzzy logic system much more robustness compared to binary logic as it can map a larger range of inputs.

Many structural dynamics problems may be represented by a coupled set of second-order dynamic systems. Coupled rigid-body and flexible body dynamics are sensitive to movement vibrations, which add instability to the structures. The best solution is to use active control to augment structural dynamics.

The objective for this research project is to develop an effective non-linear active structural control methodology to provide stability in large flexible structures. Such structures include robotic manufacturing arms that must perform tasks in a quick and accurate manner. This project also takes into consideration managing stability with minimum cost. Stability in these structures is obtained by damping oscillations that occur during rapid movement of the structure.

Optimization of the non-linear fuzzy logic controller is best accomplished using genetic algorithms. Fuzzy logic controllers are defined by a large set of parameters which greatly increase the search space to find optimum values for the controller. Genetic algorithms mimic natural evolution through Darwinian selection [3]. Individuals who are best suited to the environment survive and produce the next generation. Over a large number of generations individuals become stronger as the weaker individuals are filtered out. Starting with a large number of diverse solutions allows for larger portions of the search space to be investigated and to then converge on areas that provide the best solutions. The automation of the optimization process generalizes the fuzzy logic controller so that it may be easily applied to different systems with varying requirements and parameters.

RESEARCH GOALS AND OBJECTIVES

The primary goal of the research project is to develop an optimized non-linear active structural control methodology for coupled flexible-rigid body structures. The following objectives have been identified to achieve this goal:

1. Understand how fuzzy systems use active control on coupled dynamic systems.
2. Study several different proposed control solutions.
3. Develop a fuzzy control system from identified characteristics.
4. Learn and apply optimization techniques to the newly developed control system.

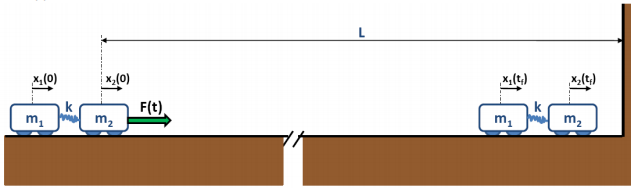
RESEARCH TASKS

The following research tasks are being conducted to achieve the objectives for this project:

1. Develop a simulation of a spring-mass system to provide a testing environment for the damping controllers.
2. Analyze several proposed solutions to determine the best characteristics and rule base for the fuzzy controller.
3. Develop and hand-tune a fuzzy logic damping controller to outperform all other controllers in the simulation.
4. Implement a genetic algorithm to further optimize the fuzzy logic damping controller.

COUPLED SPRING-MASS SIMULATION MODEL

A simulated environment is necessary to test the efficacy of the proposed fuzzy inference systems. The simulation consists of two cars connected by a spring that must traverse a given distance as fast as possible without crashing into a wall at the other end. The car system is propelled by a force on the front car. This force is the output of the developed controllers. At each time instant the controllers must determine how much force to exert on the carts, and in what direction, to get as close as possible to the wall in minimum time. The constants for the model are the weight of the cars, the spring constant, and the distance to the wall that must be traversed. A diagram of the simulation model is shown in Figure 1.



**Figure 1** Diagram of two rigid bodies connected by a spring traversing distance L in minimum time.

The modeled system contains displacement and velocity sensors on each cart. Therefore, the four inputs to the FIS are the distances traveled and the velocities of both car 1 and car 2. The output of the system is the force to be applied to car 2, limited to ±1N. At each time instant the FIS will use the four inputs and determine the force that must be applied to cart 2, governed by a set of fuzzy rules.

Input:

Output:

The initial conditions at time t=0 are that both car 1 and car 2 are at rest and are starting at position zero. The maximum runtime for the simulation was 500 seconds. The final conditions for the simulation stipulate that the distance traveled by the two cars should not exceed 100m and they should be at rest without oscillating.

,

The acceleration of car 1 is determined by the displacement between the two cars, the spring constant, and the mass of the car. The acceleration of car 2 is also determined by these factors in addition to the output force exerted on the car over its mass.

Car 1:

Car 2:

DATA EVALUATION

The efficiency of an FIS’s control of the system will be based upon how fast it traverses the distance to the wall and how close the cars settle to the wall without hitting it. The cost function J is then defined by the settling time *tf*, the time taken to settle the cars within 1m of the wall and the steady state error, the distance between the front car and the wall. The control system that produces the lowest value of *J* will be proven to be the best solution.

In order to provide a frame of reference for the performance of any control system the theoretical limits of the simulation were calculated to provide a lower bound. This lower bound is the minimum cost that an ideal, optimal solution can obtain. For a rigid body, the mass of the two cars is 3kg and the maximum possible force exerted on car 2 is 1N. From these values the maximum average acceleration of the system is defined as:

From the maximum acceleration of the system the minimum time to traverse the distance to the wall, 100m, can be calculated. There are two different scenarios to consider.

(1) Applying maximum force over the entire distance and instantaneously stopping the cars at the wall. This scenario is not feasible, but will serve as the absolute minimum time it will take for the cars to approach the wall.

The velocity is the average velocity of the cars over the distance traversed, 100m. If a constant force of 1N is applied to the cars and the initial velocity is zero, the traversal time can be calculated.

, vi = 0

(2) Applying maximum force over half the distance and then applying maximum negative force in the second half to slow the velocity of the cars to zero right at the wall.

First Half:

Decelerating back to rest over the next fifty meters takes the same amount of time; therefore the total time taken to reach 100m is 34.46 seconds. However, we want the time taken to reach the 99m mark. The time taken to travel the last meter is 2.45 seconds. Therefore, the best time to reach the 99m mark and reach zero velocity at the wall is:

If the car stops at exactly 100m the cost function evaluates to:

The latter scenario is the more realistic of the two and will be the limit that the control systems will try to achieve. The actual system is a flexible body structure that will oscillate, so the control systems will not reach this rigid body limit.

FUZZY INFERENCE SYSTEM

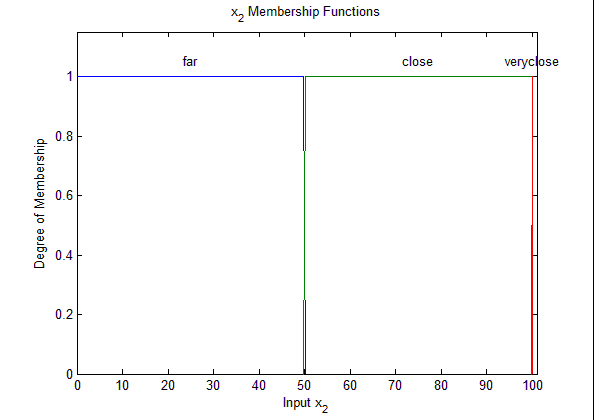
A fuzzy inference system (FIS) is a control system built on the basic principles of fuzzy logic [2]. It can take any number of analog inputs and map them to a set of logical variables ranging from 0 to 1. This mapping is performed by the membership functions, which determine the degree of membership each input has to that function. Each input typically has multiple membership functions. Based upon the value of the input, it will have a different value for each membership function. Each of these membership functions are used in a linguistic rule base of IF-THEN statements that determines the analog output variable. By using the set of rules and the membership functions the FIS is able to determine an appropriate analog output given a set of analog inputs.

The FIS built during this project uses two measured inputs, the position and velocity of car 2. The output is the force that should be exerted on car 2 at any point in the simulation.

**Position**

The first input variable, position, has three different membership functions to which it can map. The position of the car, within the range [0 100], can either be far away from the wall, close to the wall, or very close to the wall. If the simulation has just begun and the car is as far away as possible from the car, the far membership function will be 1 and the other two will be 0. Conversely, at the end of the simulation as the car approaches the wall the far membership function will be 0, the very close will be approaching

1, and close will be somewhere in between. The membership functions for position can be seen in Figure 2.

**Figure 2** Position membership functions.

Membership Function Parameters:

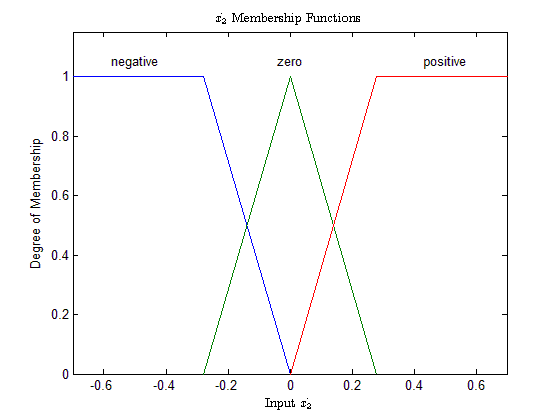
Far: Trapezoid [0 0 49.8 50.1]

Close: Trapezoid [49.8 50.1 99.9 100]

Very Close: Triangle [99.9 100 100.1]

**Velocity**

The second input variable, velocity, has three different membership functions. The velocity of the car, within the range [-6 6], can either be negative, zero, or positive. When the simulation starts, the car is at rest and the degree of membership to zero velocity will be 1 and the others will be 0. For a majority of the simulation the car is moving forward at a fast pace, therefore, the degree of membership to positive velocity will be 1. The negative membership function will not come into play until the end of the simulation as the car slows down and the oscillations between to the cars become more noticeable. At this point the vibrations will cause the car to bounce back and forth between positive and negative velocity as the car system tries to settle at the wall. The membership functions for velocity can be seen in Figure 3.

**Figure 3** Velocity membership functions.

Membership Function Parameters:

Negative: Trapezoid [-6 -6 -0.2792 0]

Zero: Triangle [-0.2792 0 0.2792]

Positive: Trapezoid [0 0.2792 6 6]

**Rule Base**

The rule base of an FIS is a series of IF-THEN conditions that use the membership functions of all the inputs in order to determine the output variable. There are also several membership functions for the output variable that the rule base maps too. Each of the rules has an influence on what the output of the system should be. The weight of these influences again varies from 0 to 1 and the final output value is determined by calculating the centroid of all of the rules. For instance, one rule says that if the car is far away then the output force should be positive and large so that the car may move towards the wall. Another rule will say that when the car is close to the wall the output force should be negative in order to slow the car down. All of these rules have influence over all input ranges determined by how the input variables map to the given membership functions in that specific rule.

The IF portion of the statement contains input membership functions of both variables and the THEN portion contains the output membership functions. The system developed during this project has seven distinct rules, shown below in Table 1.

IF Position is *Far* THEN Force is *Positive*

IF Position is *Close* and Velocity is *Negative* THEN

Force is *Positive*

IF Position is *Close* and Velocity is *Zero* THEN Force

is *Zero*

IF Position is *Close* and Velocity is *Positive* THEN Force

is *Negative*

IF Position is *Very Close* and Velocity is *Positive* THEN

Force is *Positive*

IF Position is *Very Close* and Velocity is *Zero* THEN

Force is *Zero*

IF Position is *Very Close* and Velocity is *Positive* THEN

Force is *Negative*

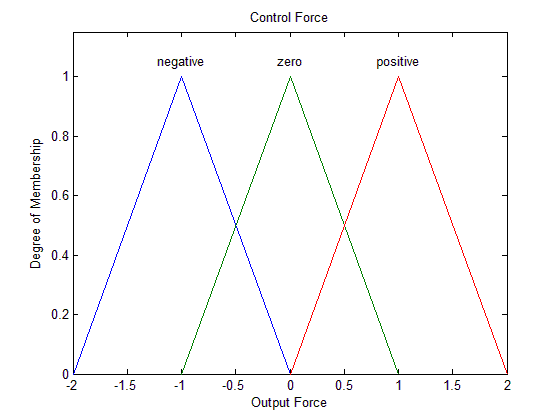
**Table 1** Rule base of the Fuzzy Inference System.

Velocity

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Negative** | **Zero** | **Positive** |
| **Far** | **Positive** | | |
| **Close**  Position | **Positive** | **Zero** | **Negative** |
| **VeryClose** | **Positive** | **Zero** | **Negative** |

**Output Force**

The output variable, force, also has three membership functions. The output force, within the range [-1 1], can either be negative, zero, or positive. The weight of these membership functions depends on the weight of each rule that maps to it. The centroid of these three membership functions is then computed to determine the final force output. At the beginning of the simulation the rules that want a positive output force will dominate and force the centroid to be 1. When the control system determines that the car system should begin slowing down, the centroid will be forced to negative 1. This switchover will occur over a very short time period around halfway to the wall. This is known as a bang-bang approach, with maximum acceleration and then maximum deceleration. As the car system decelerates and approaches the wall the oscillation damping rules will dictate the force to apply on the car system. Typically, the output force will be opposite of the motion of car 2. The membership functions for the output force are shown in Figure 4.

**Figure 4** Control force membership functions.

Membership Function Parameters:

Negative: Triangle [-2 -1 0]

Zero: Triangle [-1 0 1]

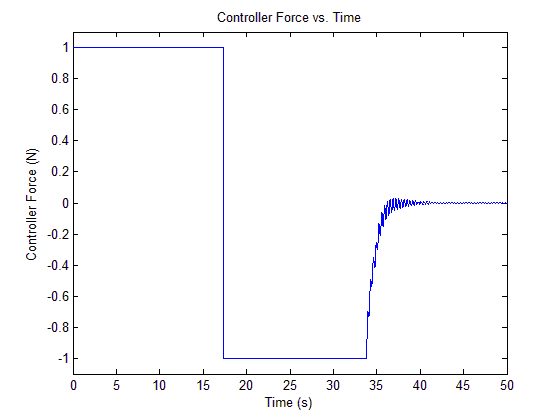
Positive: Triangle [0 1 2]

**Simulation Performance**

The efficiency of this FIS was determined by using it as the control mechanism in the simulation. The goal was to approach the realistic theoretical limits calculated above and reach this goal with minimum force. Table 2 shows the settling time, final position of car 2 and the calculated *J* value for the controller’s performance. Figure 5 shows the controllers output force over time for each step of the simulation. The controller’s output force is much lower than previously tested fuzzy solutions as well as an optimal linear controller.

**Table 2** Final results from implemented FIS.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 32.303s | 99.999821m | 0.32339 |



**Figure 5** System control force over time.

GENETIC ALGORITHM

The performance of the FIS depends directly on the value of each parameter of the membership functions. These values were hand-tuned by a time consuming process of trial and error. To quicken this process, a genetic algorithm was utilized to autonomously tune the membership functions and approach an optimal solution.

A genetic algorithm is a computational mechanism which imitates evolutionary behavior to achieve optimality. It consists of a population of individuals which undergoes a process similar to natural selection, reproduction, and mutation over the course of a number of generations [3]. Selection is attained by evaluating the fitness of each individual according to a fitness function. For the purposes of this research, each individual represents a FIS and the fitness function is simply the cost function used earlier. The individual which most effectively minimizes the cost function is considered most fit for the control environment.

In order to manipulate the FIS structure in an algorithmic manner, it is necessary to represent it as a genetic individual. Since the values of the parameters of the membership functions of the FIS have significant impact on the control performance, it was decided to manipulate only these parameters with the algorithm; however, of the thirty-one parameters which comprise this FIS model, many are trivial to the overall performance. It is desirable to reduce the number of parameters to facilitate the optimization process.

**Parameter Reduction**

To simplify the genetic tuning of the parameters, the symmetry of the system was exploited. The parameters were reduced from thirty-one to seven. The position membership functions were simplified to only three parameters by defining a center point (center) between far and close functions, a distance from the center point at which the far and close functions will be valued at 0 and 1 respectively (iTrap1), and half of the base of the triangular membership function which decides when the car is very close to the wall (iTriBase1). The velocity membership function parameters were reduced to two parameters by defining the one parameter for the distance from 0 that each of the membership functions reaches 1 for the negative and positive functions (iTrap2) and another to define half of the base of the zero velocity triangular membership function (iTriBase2). The output force membership functions were reduced similarly by allowing the negative and positive membership functions become trapezoidal (oTrap and oTriBase).

Position Membership Function Parameters:

Far: Trapezoid [0 0 (center-iTrap1) (center+iTrap2)]

Close: Trapezoid [(center-iTrap1) (center+iTrap1) 99.9 100]

Very Close: Triangle [(100-iTriBase1) 100 (100+iTriBase1)]

Velocity Membership Function Parameters:

Negative: Trapezoid [-6 -6 (0-iTrap2) 0]

Zero: Triangle [(0-iTriBase2) 0 (0+iTriBase2)]

Positive: Trapezoid [0 (0+iTrap2) 6 6]

Control Force Membership Function Parameters:

Negative: Trapezoid [-2 (-1-oTrap) (-1+oTrap) 0]

Zero: Triangle [(0-oTriBase) 0 (0+oTriBase)]

Positive: Trapezoid [0 (1-oTrap) (1+oTrap) 2]

These parameter reductions allow an individual to be defined by a string of seven numbers.

Individual Definition:

[iTrap1 center iTriBase1 iTrap2 iTriBase2 oTrap oTriBase]

**Population Initialization**

An initial population is generated by assigning random values to each of the individual parameters within given ranges. iTrap1, iTriBase1, and oTriBase are allowed to vary between 0.05 and 1. iTrap2 and iTriBase2 are allowed to vary between 0.05 and 2. Center values fall between 45 and 55, and oTrap between 0.05 and 0,95. Twenty individuals comprise a population. Each individual is evaluated for fitness and brought up for selection to produce a new generation.

**Parent Selection and Reproduction**

A new generation consists of three individuals which remain unchanged from the previous generation, called elite children, ten individuals which are created from recombination of two parents, five individuals which are created from mutating recombined children, and two individuals randomly defined from the previously defined ranges.

Parents are selected by selecting the three best fit individuals to both become parents and elite children. Seven more parents are selected by randomly choosing three individuals from the remaining population, selecting the most fit, and returning the other two. This tournament style of selection is repeated until all parents are selected.

Reproduction occurs by a process called blended crossover with an modification (BLX-), by selecting a new parameter from the range [], where:

and

Parents are defined as:

and .

and .

The interval is the user-defined range for that specific parameter. This mechanism allows the algorithm to create a child from two parents which is a blend of both parents, while still expanding the search space. As a population generally converges on a solution, so too do the children of the population.

**Mutation**

Five of the recombined children are selected. Two parameters are selected at random from each child to be mutated. Mutation is defined to be non-uniform such that the mutation has a smaller effect in later generations as follows:

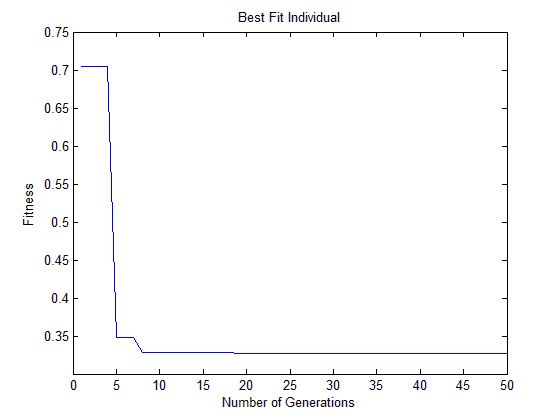
Where represents a coin flip such that .

Where is the current generation, and is the maximum number of generations. is a random value from the interval [0,1]. The function computes a value in the range of such that the probability of returning a zero increases as the algorithm advances. The value of determines the impact of time on the probability distribution of . is set to 1.5 for the algorithm for this research.

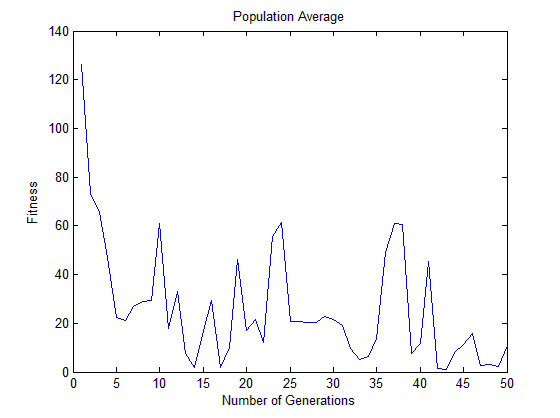
Two additional children are added to the population by random selection from the ranges in order to ensure that the search space is sufficiently large.

RESULTS

Running the algorithm for 50 generations yields a FIS which performs nearly as well as the hand-tuned FIS from above. Figure 6 shows the value of the best fitness from each population over time. Figure 7 shows the average fitness values from each population over time. It can be seen that the algorithm converges on a solution after only few iterations, while finally settling on the best fit individual after eighteen iterations. Table 3 shows the settling time, final position and calculated *J* value from the best fit individual FIS from the genetic algorithm.



**Figure 6** Best fitness by generation



**Figure 7** Average fitness values by generation

**Table 3** Final results from algorithm FIS

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 32.634s | 99.999999m | 0.32634 |

**Genetic Adaptability**

The above results demonstrate the ability of the genetic algorithm to tune a FIS to near-optimal performance. All tuning and development heretofore was done with one unchanged two-mass spring system. Changing the masses of each car significantly causes the FIS controller to perform very poorly; however, utilizing the genetic algorithm to generate an optimum FIS for each new system is an efficient method of developing good active controllers. This is demonstrated by changing the masses and to 2kg and 4kg respectively. They are again changed to 4kg and 8kg, and 4kg and 16kg. The algorithm was used for each case. After twenty generations the genetically optimized FIS performed within 10% of the theoretical rigid body limit in three cases out of four. With a total system mass of 20kg, the FIS performed within 20% of the theoretical limit. These results are shown in Table 4.

**Table 4** Genetic algorithm FIS performance compared to hand-tuned FIS and rigid body limit



It is easily seen in these results that the use of the genetic algorithm is advantageous in the autonomous development of near optimal FIS controllers. Whereas the hand-tuned FIS performs slightly better for the system for which it was tuned, the genetic algorithm is able to tune a FIS rapidly and accurately for a varied set of circumstances.

CONCLUSIONS

**Fuzzy Logic**

Fuzzy logic provides a robust framework for control. It has been demonstrated that proper fuzzy control is efficient and computationally inexpensive. The inherent vagueness of set membership and linguistic operation of fuzzy logic allows the controller to mimic expert human control. This superior control, however, comes with a steep cost in FIS development. Hand-tuning a FIS is time consuming and tedious.

**Genetic Algorithm**

The use of the genetic algorithm facilitates FIS development. Once a FIS has been developed for a general type of control situation, it is relatively simple to define the FIS as a genetic element and automate the tuning through the evolutionary process. These results imply that if a general fuzzy controller is developed for a family of control situations, then a genetic algorithm can be implemented to tune each FIS to its specific task. The tuning, therefore, can be accomplished by someone with no expertise in the control of the situation. As the computation is quick, efficient control could be widely distributed due also to the low-cost of development.

REFERENCES

[1] B. Kosko and S. Isaka, Fuzzy Logic, Sci Am 269 (1993), 76-81.

[2] MATLAB and Fuzzy Logic Toolbox Release 2012b, The Mathworks, Inc, Natick, Massachusetts, United States.

[3] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, Genetic Fuzzy Systems, Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific Publishing Co., MA, 2001