SciTech, 5 January 2015, Kissimmee, FL

Genetic Optimization of Fuzzy Logic Control for Coupled Dynamic Systems

Andrew Janson,[*] Nicklas Stockton,[†] and Kelly Cohen,[‡]

College of Enginering and Applied Science, University of Cincinnati, Cincinnati, Ohio

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 3%, 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.

Nomenclature

C^i	Genetic parent
F	Force, N
I	Selection interval
K	Spring Constant, N/m
L	Length, m
d	Interval buffer distance
m	Mass, kg
t	Time, s
$x(t)$	Position, m
x_i^j	Parameter of genetic individual
$\dot{x}(t)$	Velocity, m/s
$\ddot{x}(t)$	Acceleration, m/s^2

[*] Senior, Department of Electrical Engineering and Computing Systems

[†] Sophomore, Department of Aerospace and Engineering Mechanics

[‡] Associate Professor, Department of Aerospace and Engineering Mechanics

Copyright © 2014 by the American Institute of Aeronautics and Astronautics, Inc. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the copyright owner.

y	State vector

Subscripts

0	Denotes initial condition
1	Denotes property of trailing cart
2	Denotes property of leading cart
f	Denotes condition at settling time

Symbols

α	Parameter modifier
$\Delta(t, x)$	Mutation function
λ	Random multiplier
τ	Coin flip value

I. 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 the weekend.[2] Binary logic allows only a yes/no answer to the question: "Is this day part of the weekend?" However, human reasoning does not consider the "weekendness" of a day to immediately rise to a value of one from zero at exactly 12am Saturday. 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.[3] 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.[4] 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.

A. Goals and Objectives

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

- Understand how fuzzy systems exert active control on coupled dynamic systems.
- Study several different proposed control solutions.[5,6,7,8]
- Develop a fuzzy control system from identified characteristics.
- Learn and apply optimizations techniques to the newly developed control system.

B. Approach

We undertook the following research tasks to achieve the previously stated objectives:

1. Developed a simulation of a spring to provide a testing environment for the damping controllers.
2. Analyzed several proposed solutions to determine the best characteristics and rule base for the fuzzy controller.
3. Developed and hand-tuned a fuzzy logic damping controller to outperform all other controllers in the simulation
4. Implemented a genetic algorithm to further optimize the fuzzy logic damping controller.

II. Coupled Spring-Mass Simulation Model

A simulated environment is necessary to test the efficacy of the proposed fuzzy inference systems (FIS). The simulation consists of two cars connected by a spring; this system is expected to traverse a given distance as quickly as possible without exceeding a certain bound represented by a wall. The system is propelled by a force on the leading car which represents the actuating output of the controller. At each time instant, the controller must determine how much force to exert on the carts, and in which direction, to get as close as possible to the wall in minimum time. The constants for the model are the weight of each car, the spring constant, and the distance which must be traversed to the wall. A diagram of the simulation is shown in figure 1.

$$m_1 = 1\text{kg}, \quad m_2 = 2\text{kg}, \quad K = 250\text{N/m}, \quad L = 100\text{m}$$

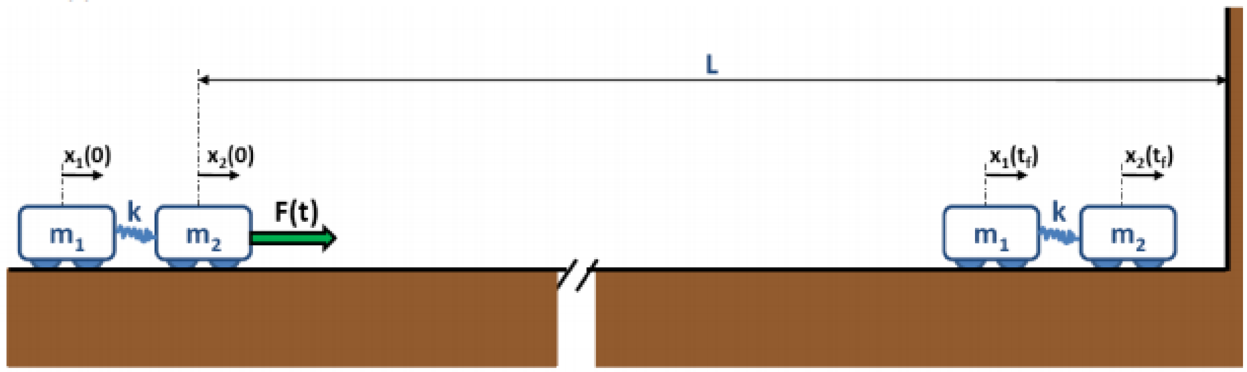


Figure 1. Diagram of two rigid bodies connected by a spring traversing distance L in minimum time.

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 carts. These iputs represent the state vector of the system. The output of the controller is the force ,$F(t)$, applied to cart 2, limited to ±1N. At each time step of the simulation, the FIS uses the four inputs to determine the force which must be exerted according to a fuzzy rule base.

$$\text{Input}: \quad \vec{y}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix}$$

$$\text{Output}: \quad |F(t)| \leq 1\text{N}$$

At time $t = 0$, both carts are at rest at position $x = 0$. The maximum allowed runtime of the simulation is 500 sṪhe system requirement for the final condition is that neither cart exceeds 100 m and should be at rest with negligible oscillation.

$$\vec{y}_0 = \begin{bmatrix} 0\text{m} \\ 0\text{m} \\ 0\text{m/s} \\ 0\text{m/s} \end{bmatrix}, \quad \vec{y}(500) = \begin{bmatrix} < 100\text{m} \\ < 100\text{m} \\ 0\text{m/s} \\ 0\text{m/s} \end{bmatrix}$$

The acceleration of cart 1 is determined by the displacement between the two carts, the spring constant, and the mass of the cart. The acceleration of cart 2 is also a function of these parameters as well as the control force. The equations of motion for the system are represented by simple second-order differential equations.

$$\text{Cart1}: \quad \ddot{x}_1 = \frac{K}{m_1}(x_2 - x_1) \tag{1}$$

$$\text{Cart2}: \quad \ddot{x}_2 = \frac{K}{m_2}(x_1 - x_2) + \frac{F}{m_2} \tag{2}$$

A. Data Evaluation

The efficiency of a FIS's control of the system is based upon the amount of time it expends traversing the distance to the wall and how close the carts are to the wall when they settle. Any breach of the wall results in immediate system failure. A cost function, J, is defined by the settling time t_f, the time taken to settle within 1 m of the wall and the steady state error, the distance between the leading carts and the wall. The control system that produces the lowest J value will be proven to be the most fit solution for the simulation.

$$t_f = |L - \bar{x}(t_f)| \leq 1\text{m} \tag{3}$$

$$J = \frac{t_f}{100} + 2[L - x_2(500)] \tag{4}$$

Where the constants 100 and 2 are scaling factors. 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 for the value of Eq. (4). Assuming a single rigid body ass with no dynamic coupling, the model is greatly simplified to a single equation where the acceleration of the body is a function of only the control force.

$$\ddot{x} = \frac{F}{M} \tag{5}$$

Where M is the total mass of the systemThe total mass of the system is 3 kg whereas the maximum input force is limited to 1 N, rendering Eq. (5)

$$\ddot{x} = \frac{1\text{N}}{3\text{m}} = \frac{1}{3}\text{m/s}^2$$

as the maximum acceleration. Given this acceleration and the distance to be traversed to the wall, the minimum time to complete the trip can be calculated. There are, however, two scenarios to consider.

1. Applying maximum force over the entire distance and instantaneously stopping the carts at (but not touching) the wall provides an absolute, if unfeasible, optimum simulation completed in minimum time. Given an initial velocity of zero and a constant accelerating force of 1 N, the traversal time is calculated. Note that once the carts have breached 99 m, the system may come to rest and be considered settled.

$$x(t) = \dot{x}_0 t + \frac{1}{2}\ddot{x}t^2, \quad \dot{x}_0 = 0$$

$$x(t) = \frac{1}{2}\ddot{x}t^2$$

Letting $x(t) = 99$

$$t = \sqrt{\frac{2x(t)}{\ddot{x}}} = \sqrt{\frac{2 \cdot 99\text{m}}{\frac{1}{3}\text{m/s}^2}} = 24.37\text{s}$$

2. Applying maximum force over half of the distance and then applying maximum negative force in the second half to slow the velocities of the carts to zero at the wall position.

   First half:

$$t = \sqrt{\frac{2x(t)}{\ddot{x}}} = \sqrt{\frac{2 \cdot 50\text{m}}{\frac{1}{3}\text{m/s}^2}} = 17.23\text{s}$$

   TRaversing the second half and stopping at the wall takes the same amount of time, therefore the total time expended in reaching 100 m is 34.46 s; however, we are interested in the time taken to breach the 99 m mark. The time taken to travel the last meter is 2.45 s, so the best possible time to reach the 99 m position is :

$$t = 32.19\text{s}$$

   As this is a much more realistic scenario, this is the limit used as the benchmark in this research. Using this time to evaluate Eq. (4 on the previous page)

$$J = \frac{32.19}{100} + 2[100 - 100] = 0.3219$$

   results in the minimum possible cost. The addition of harmonic oscillation and non-linear dynamics ensures that this limit will not be reached, but merely provides a standard against which a controller may be measured.

III. Fuzzy Inference System

A fuzzy inference system (FIS) is a control system built on the basic principles of fuzzy logic[1].[9] It can take an arbitrary number analog inputs and map them to a set of logical variables ranging from 0 to 1. This mapping is performed by membership functions, which determine the degree of membership each input has to a function. Each input typically has multiple membership functions. Based on the value of the input, it will have a different value for each membership function. Each of these membership functions are evaluated according to a linguistic rule base of IF-THEN statements which determines the analog output variable. By using the set of rules and the memberships functions, the FIS is able 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 cart 2. The output of the FIS is the force exerted on cart 2 at a given point in the simulation.

A. Membership Functions and Rule Base

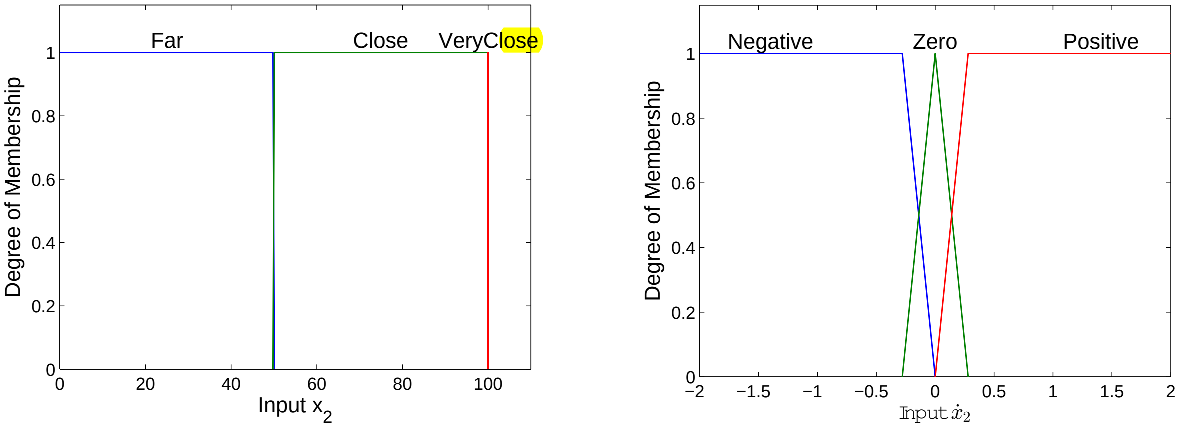
1. Position

The first input variable, position, is composed of three membership functions to which it can map. The position of the car, from 0 to 100, is described in human-understandable language as far away, close to, or very close to the wall. If the simulation has just begun and the cart is as far away from the wall as it can be, the "Far" membership function will map to 1 and the "Close" and "VeryClose" functions will evaluate to 0. Conversely, at the end of the traverse, as the car approaches the wall, "VeryClose" will map to 1 and "Far" to 0. "Close" may evaluate to some value in between. The membership functions are shown in figure 2(a) on the following page. As the system predominately behaves as a rigid body on a large scale, the membership functions mirror the ideal simulation of a rigid body for the majority of the carts' travel. Each function is represented by a vector of values expressing the points at which the function switches from 0 to 1 or 1 to 0. For the FIS used in this research, trapezoidal- and triangular-shaped membership functions are utilized. Trapezoids are represented by a four-element vector as they start at 0, rise to 1, remain at 1, and finally fall to 0. Likewise, triangular functions are expressed as three-element vectors. The parameters of the position membership functions shown in figure 2(a) on the next page are:

$$\text{Far}: \quad \text{Trapezoid}\begin{bmatrix} 0 \\ 0 \\ 49.8 \\ 50.1 \end{bmatrix}, \quad \text{Close}: \quad \text{Trapezoid}\begin{bmatrix} 49.8 \\ 50.1 \\ 99.9 \\ 100 \end{bmatrix}, \quad \text{VeryClose}: \quad \text{Triangle}\begin{bmatrix} 99.9 \\ 100 \\ 100.1 \end{bmatrix}$$

2. Velocity

The second input variable, velocity, is composed of three membership functions. The velocity of the cart, within a range from -6 to 6, can either be classified as "Negative", "Zero", or "Positive". When the simulation starts, the carts are at rest and the degree of membership to "Zero" velocity will be 1 whereas "Negative" and "Positive" will be 0. For the majority of the simulation, the carts are moving forward with a fast pace; therefore, the "Negative" membership function does not come into play until close to the very end of the simulation when the oscillatory effects dominate the motion of the cart system. It is at this point that the true dynamic nature of the system is exhibited and the controller does the most calculation in an attempt to damp the oscillations. The membership functions for $\ddot{x}_2$ are shown in figure 2(b).



(a) x_2 membership functions

(b) $\dot{x}_2$ membership functions

Figure 2. Input membership functions

The velocity membership function parameters are:

$$\text{Negative}: \quad \text{Trapezoid}\begin{bmatrix} -6 \\ -6 \\ -0.2792 \\ 0 \end{bmatrix}, \quad \text{Zero}: \quad \text{Triangle}\begin{bmatrix} -0.2792 \\ 0 \\ 0.2792 \end{bmatrix},$$

$$\text{Positive}: \quad \text{Trapezoid}\begin{bmatrix} 0 \\ 0.2792 \\ 6 \\ 6 \end{bmatrix}$$

3. Rule Base

The rule base of an FIS is a series of IF-THEN (antecedent-consequent) 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 to. 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 dictates that if the car is far away then the output force should be positive and large so that the car will 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 antecedent of each statement contains memberships of both input variables and the consequent maps to the output membership functions. The rules for this FIS were developed based upon intuitive decision making. The rules developed for our FIS are shown in Table 1 on the next page.

Table 1. Rule Base of the Fuzzy Inference System

		Velocity Measurement		
		Negative	Zero	Positive
Position Measurement	Far	Positive		
	Close	Positive	Zero	Negative
	VeryClose	Positive	Zero	Negative

4. Control Force

The output variable, force, is also composed of three membership functions. The output force is bounded to be between -1 and 1N, thus the membership functions allow for the output to be "Negative", "Zero", or "Positive". As the centroid of the area beneath each function is used to evaluate the control force, the upper and lower bounds are centered over 1 and -1 respectively. These functions represent the "defuzzification" stage of the FIS and these outputs are determined by evaluating each rule in the rule base (see § 3 on the preceding page) in parallel.[2] The controller emulates bang-bang control for the majority of the simulation lifetime, sharply transitioning from full acceleration to full deceleration. As the cart system nears the wall, the oscillation damping rules will dictate the force to apply on the system. Typically, the output force will be directed opposite to the velocity of cart 2. The membership functions for the control force are shown in figure 3.

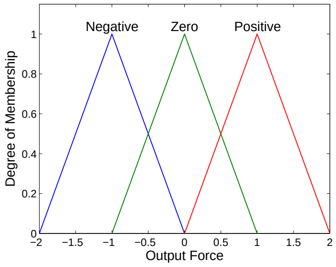


Figure 3. Control force membership functions.

$$\text{Negative}: \quad \text{Triangle}\begin{bmatrix}-2\\-1\\0\end{bmatrix}, \quad \text{Zero}: \quad \text{Triangle}\begin{bmatrix}-1\\0\\1\end{bmatrix}, \quad \text{Positive}: \quad \text{Triangle}\begin{bmatrix}0\\1\\2\end{bmatrix}$$

B. 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 on the following page shows the settling time, final position of cart 2 and the calculated J value for the controllers performance. Figure 4 shows the controllers output force over time for each step of the simulation. The controllers output force is much lower than previously tested fuzzy solutions as well as an optimal linear controller.

It is clear that the controller expends less energy than a traditional bang-bang type controller would in the oscillation damping process. It can be seen that the FIS responds quickly to control needs and applies the needed force with low-latency.

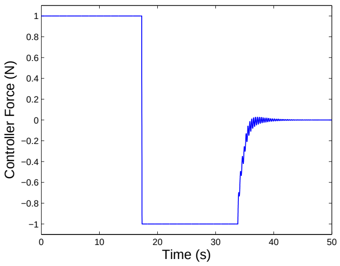


Figure 4. System control force over time.

Much work was done to hand-tune this controller to perform optimally, thus the work was undertaken to develop a genetic algorithm to produce similar results autonomously. Automating the tuning process ultimately produces a controller which is nearly as good as the hand-tuned controller, but does so at little to no effort to the programmer.

Table 2. Results from implemented FIS.

t_f	$x_2(500)$	J
32.303s	99.999821m	0.32328

IV. 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.[4] 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.

A. 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 Parameter

$$\text{Far}: \quad \text{Trapezoid}\begin{bmatrix} 0 \\ 0 \\ (center - iTrap1) \\ (center + iTrap1) \end{bmatrix}, \quad \text{Close}: \quad \text{Trapezoid}\begin{bmatrix} (center - iTrap1 \\ (center + iTrap1) \\ 99.9 \\ 100 \end{bmatrix},$$

$$\text{VeryClose}: \quad \text{Triangle}\begin{bmatrix} (100 - iTriBase1) \\ 100 \\ (100 + iTriBase1) \end{bmatrix}$$

- Velocity Membership Function Parameters

$$\text{Negative}: \quad \text{Trapezoid}\begin{bmatrix} -6 \\ -6 \\ (0) - iTrap2) \\ 0 \end{bmatrix}, \quad \text{Zero}: \quad \text{Triangle}\begin{bmatrix} (0 - iTriBase2) \\ 0 \\ (0 + iTriBase2) \end{bmatrix},$$

$$\text{Positive :}\quad \text{Trapezoid}\begin{bmatrix}0\\(0+iTrap2)\\6\\6\end{bmatrix}$$

- Control Force Membership Function Parameters

$$\text{Negative :}\quad \text{Trapezoid}\begin{bmatrix}-2\\(-1-oTrap)\\(-1+oTrap)\\0\end{bmatrix},\quad \text{Zero :}\quad \text{Triangle}\begin{bmatrix}(0-oTriBase)\\0\\(1+oTriBase)\end{bmatrix},$$

$$\text{Positive :}\quad \text{Trapezoid}\begin{bmatrix}0\\(1-oTrap)\\(1+oTrap)\\2\end{bmatrix}$$

These Parameter reductions allow an individual to be defined by a single vector of seven variables.

- Individual Definition

$$\begin{bmatrix}iTrap1\\center\\iTriBase1\\iTrap2\\iTriBase2\\oTrap\\oTriBase\end{bmatrix}$$

B. 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.

C. 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 blended crossover process with an α modification (BLX-α), by selecting a new parameter x'_i from the range $[x_{min} - I\alpha, x_{max} + I\alpha]$, where

$$x_{min} = min(x_i^1, x_i^2) \quad \text{and} \quad x_{max} = max(x_i^1, x_i^1)$$

Parents are defined as

$$C^1 = (x_1^1, x_2^1, \cdots, x_7^1) \quad \text{and} \quad C^2 = (x_1^2, x_2^2, \cdots, x_7^2)$$

$$I = \frac{x_{max} - xmin}{b_i - a_i}$$

$$\alpha = min(d^1, d^2)$$

$$d^1 = x_{min} - a_i \quad \text{and} \quad d^2 = b_i - x_{max}$$

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

D. Mutation

Five of the recombined children are selected by random sampling and then two randomly sampled parameters are selected from each of these child for mutation. Mutation is defined to be nonuniform such that the mutation has smaller effect in later generations as follows:

$$x_i' = \begin{cases} a_i + \Delta(t, x_i - a_i), & \text{if } \tau = 0 \\ b_i - \Delta(t, b_i - x_i), & \text{if } \tau = 1 \end{cases}$$

where τ represents a coin flip such that $P(\tau = 1) = P(\tau = 0) = 0.5$

$$\Delta(t, x) = x(1 - \lambda(1 - \frac{t}{t_{max}})^b)$$

where t is the current generation, and t_{max} is the maximum number of generations. The variable λ is a random value from the interval $[0, 1]$. The function Δ computes a value in the range $[0, x]$ such that the probability of returning a zero increases as the algorithm advances. The value of b determines the impact of the time on the probability distribution of Δ. The value of b is set to 1.5 for 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.

V. Results

Running the algorithm for 50 generations yields a FIS which performs as well as the hand-tuned FIS from B on page 7. This result converges out of the evolution process quickly as can be seen in figure 5(a) on the next page. Though the algorithm finds a near optimal solution quickly, it continues to search similar solutions, selecting the best individuals each time to produce progressively better fit children each generation. Figure 5(b) on the following page shows the average fitness of generation. It is clear from this plot that the algorithm produces many unfit children in its search for optimality. This satisfies the need of a good algorithm to expand the search area to eliminate premature convergence.

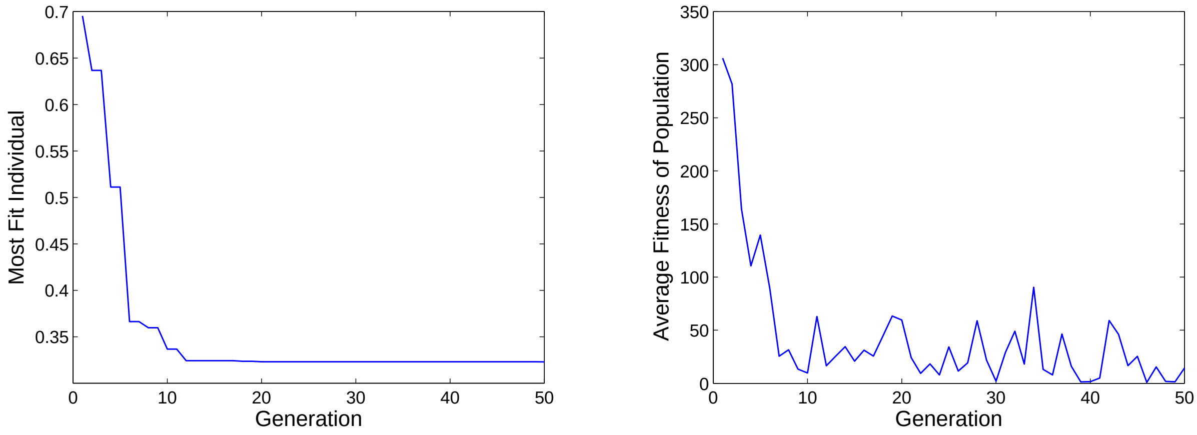
The results of the performance of the most fit individual produced by the genetic algorithm are shown in Table 3.

Table 3. Final Results from algorithm-generated FIS

t_f	$x_2(500)$	J
32.308s	99.999999m	0.32311

A. Genetic Adaptability

These results demonstrate the ability of the genetic algorithm to tune a FIS to near-optimal performance. All tuning and development heretofore was done with an unchanging system setup of known masses connected by a known spring. Although a robust controller,[3] introducing changes to the masses of each car significantly alters the performance of the FIS as it was carefully tuned to only a certain envelope; however, utilizing the genetic algorithm to generate an optimum FIS for each new system setup is an efficient method of developing good active controllers. This is demonstrated by changing the masses m_1 and m_2 to 2kg and 4kg respectively. They are again changed to 4kg and 8kg, and finally 4kg and 16kg. The algorithm was deployed for each case to optimize a controller for that envelope. After fifty generations of evolution, the genetically optimized



(a) Best fit individual by generation.

(b) Individual fitness average by generation.

Table 4. Genetic algorithm FIS performance compared the hand-tuned FIS and rigid body limit.

	Mass 1 (kg)kg, Mass 2 (kg)			
	1kg, 2kg	2kg, 4kg	4kg, 8kg	4kg, 16kg
Theoretical Limit	0.3191	0.4553	0.6438	0.8312
GA FIS	0.3231	0.4562	0.6579	0.8434
Hand-tuned FIS	0.3230	0.6125	5.3072	3.3240
GA Error	1.3%	0.2%	2.2%	1.5%
Hand-tuned Error	1.2%	34.5%	724.4%	299.9%

FIS performed within 3% of the theoretical rigid body limit in al four cases. These results are displayed in Table 4.

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. Given a generic control architecture, the genetic algorithm is able to tune a FIS rapidly and accurately for a varied set of circumstances.

VI. Conclusions

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.

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] Kosko, B., *Fuzzy Thinking, The New Sciencs of Fuzzy Logic*, Hyperion, NY, 1994.

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

[3] Cohen, K., Weller, T., and Ben-Asher, J., "Control of Linear Second-Order Systems by Fuzzy Logic-Based Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 3, 2001, pp. 494–501.

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

[5] Walker, A., "untitled," Class project.

[6] Stimetz, A., "untitled," Class project.

[7] Mitchell, S., "untitled," Class project.

[8] Vick, T., "untitled," Class project.

[9] Kosko, B. and Isaka, S., "Fuzzy Logic," *Scientific American*, , No. 269, 1993, pp. 76–81.