Fuzzy Mediation for Online Learning in Autonomous Agents

Giovanni Vincenti and Goran Trajkovski
Department of Computer and Information Sciences
Towson University, 8000 York Rd., Towson, MD 21204
E-mail: {gvince1, gtrajkovski} @towson.edu

Abstract

Information fusion is a field that deals with merging multiple inputs to create a single output. Several fusion mechanisms are in place, but none can be easily adapted to accommodate for online learning using a simulator. This paper analyzes the concept of fuzzy mediation as a novel approach to the creation of a framework through simulations of controllers interacting with a controllable object. We propose this new algorithm as a mean to improved methods of learning through positive reinforcement, observation, and direct interaction with the object to be controlled.

Introduction

Several application domains require a computer system to mediate in real-time between several sources of control, such as an agent that is trying to perform a set of actions to fulfill a task, while trying to navigate through obstacles and maze-like environments. We turn to information fusion when merging all the information and the drives in order to give the agent a single command that will make it go forward rather than backwards or in any other direction.

We introduce a novel method for information fusion, called fuzzy mediation. Most information fusion algorithms operate at the level of sensors, where they have to mediate and merge several pieces of information to give the agent a reliable understanding of its whereabouts. Fuzzy mediation instead is an algorithm that mediates between the desired controls of an agent from two distinct sources. Such sources can both be human, one human and one an automated controller, or both automated controllers.

In this article we report on several simulations that we performed using the Wang and Mendel algorithm (1992) for learning through a Fuzzy Associative Memory (FAM). We simulate an agent being controlled by a human expert and a novice, FAM-based automated controller, which learns as the human is manipulating the agent.

This article is organized as follows. The next section introduces the algorithm that we propose as a solution to the problem of dynamic information fusion. The third section analyzes the Artificial Intelligence algorithm that we used for the tests reported in this article. In this section we also review the training algorithms used with the Wang-Mendel learning model. The fourth section of this paper reports the findings of our simulation experiments, that are further discussed in the fifth section. The last section concludes the paper and states directions for futher research.

Fuzzy Mediation

Information fusion is a domain that deals with the integration of input from multiple sources in order to create a single output (Kokar et al., 2004). This task is particularly difficult in the event of conflicting information coming from different sources. The mediation that needs to take place is the heart of any information fusion mechanism.

One of the applications that may come to mind when talking about information fusion systems is the domain of shared control. As technology advances, more vehicles rely heavily on computing for their control (Grenoble O'Malley, 2005; Norman, 2005; Wallace, 2000). This form of control that does not allow a human to interact directly with the machine any more leaves room for the application of shared control concepts to a domain that, although cites shared control, is really based on supervised control with little to no shared interaction (Norman, 2005).

Fuzzy mediation is an innovative concept that allows two agents to interact in real-time while controlling an object. The majority of information fusion algorithms involve a static method of evaluation set at the time of system design (Kokar et al., 2004). The concept introduced in this article is based on a dynamic concept of mediation, assuming that the two controllers can interact and learn, thus the need for a reward system shifting control to the learner, as its knowledge allows for actions that resemble the trainer's inputs.

The first step of the fuzzy mediation algorithm is to receive two inputs, one coming from the trainer, T, and one from the trainee, t. Next we create a range [-10, 10] and we normalize the value of T to the 0 of this range. The range is divided into the following fuzzy sets: {Wide variation to the left, Slight variation to the left, Similar, Slight variation to the right, Wide variation to the right}. Thus, T belongs to the set of "Similar" with a membership of 1. The next task is to identify the difference between T and t and normalize it to the [-10, 10] range. This will allow us to identify to what set the value of t belongs, and with what degree of membership.

Before we go any further in the evaluation of the inputs, we have to define where control lays. In this system,

control can be mapped to a range [-1, 1], where –1 identifies complete ownership of the control given to the trainer, and 1 gives complete ownership to the trainee. Every simulation will start with a mediation value of –1, giving complete control to the trainer.

When both input values have been identified by one or more sets, we will compare these sets. In the case of the value of t having a membership of 1, we have several options. If the value of t is within the set "Similar", then the mediation value of control will shift towards 1 by a certain increment, thus giving more control to the trainee. If t belongs to one of the sets that identify a slight variation, then the value of control mediation will stay the same. If, instead, the value of t is identified by a set that indicates a wide deviation, the value of control mediation will shift in favor of the trainer, thus delegating more control to the trainer. If the value of t is identified by two sets, then the control mediation value will change accordingly to the sets that label the value t, also taking into consideration the membership value to each of the sets.

For example, if t belongs to set "Similar" with a membership value of 1, the control mediation index will shift in order to give more control to the trainee by 0.1 units. Should the value of t belong to two sets, "Similar" and "Slight variation to the right" with memberships of 0.25 and .075, then the total change in control mediation will be computed by multiplying the value associated with each set with the membership to that set. If a membership of 1 to the "Similar" set shifts the control mediation value by 0.1, then a membership of 0.25 to the same set will give a shift of 0.1 x 0.25, or 0.025, in the index of control mediation. The second set that t belongs to involves a shift of 0 in control mediation. Thus, the overall shift of control mediation for the situation just described is 0.025 in favor of relinquishing more control to the trainee.

Currently the control mediation function is dependent on the iterations of the simulation. Most likely the function will not be dependent on the frequency of the evaluation of the inputs, but it will be time-dependent.

In order to calculate the output, we will have to get the value of control mediation, normalized between [-1, 1]. This domain will be divided into two fuzzy sets, {Trainer, trainee}. The value of control mediation will be identified with one or two of these sets, with a different membership value. When the index is given membership values for both sets, the output value can be calculated by multiplying the input value of the trainer by the membership to the "Trainer" set and adding it to the input of the trainee multiplied by the membership to the "Trainee" setting. So, if the input of the trainer is of magnitude 50 and the input of the trainee is of magnitude 100, and the control mediation belongs to both "Trainer" and "Trainee" sets with equal membership, the output will be computed in the following manner: (50 x 0.5) + (100 x 0.5), giving a mediated output of 75.

Learning in agents based on the Wang-Mendel algorithm

Among many algorithms that emulate human intelligence, the Wang-Mendel algorithm (Wang and Mendel, 1992) is a very efficient one when applied to problems of vehicular control. Its first application, reported in Wang and Mendel (1992), employs this method to a truck "backer-upper" problem, where the target of the simulation was to place a truck into a virtual environment, and let a controller back it up to its loading station. In their experiments, Wang and Mendel used an expert to teach the program how to "back up" into the loading dock, and then let the application perform. They found that their algorithm was extremely efficient in solving this problem.

For this simulation we use Type-1 fuzzy sets. Although researchers argued that Type-2 fuzzy sets are more appropriate in some situations where uncertainty is more predominant (Mendel and John, 2002; Mendel, 2003), we feel that the setting of these experiments is simple enough where the uncertainty of the boundaries of the fuzzy sets can easily be overcome by increasing the number of fuzzy sets associated with the input or the output ranges in question.

FAM stores the relationship that exists between one or more events and a consequence. The details of this process are reported in (Wang and Mendel, 1992). One important aspect of this algorithm is how the system reacts when multiple events are associated with different consequences within the training pairs. When multiple examples lead to similar associations, or should the example show the FAM that an association is different from what has been previously stored, the algorithm takes into consideration the strength of the association already stored. If the new association should have a strength that is higher than the one already present, then the association will be changed, if necessary, and the new strength of the rule will be stored. Should the strength of the training example be lower than the association already stored, then the example will not be considered. The importance of an association is calculated by multiplying the membership values of the fuzzy sets that have the highest ownership of the input (event) and output (consequence) for the training pair.

For this simulation we will use this algorithm given its ideal use shown in the original work. We will simulate the control of an agent. Such control will be represented by equations 1 through 4.

$$Y = X \qquad \text{(eq. 1)}$$

$$Y = |X| \qquad \text{(eq. 2)}$$

$$Y = \sin(X) \qquad \text{(eq. 3)}$$

$$Y = X^2 \qquad \text{(eq. 4)}$$

These four equations will create training examples for the FAM to learn. In order to train the memory, we will use

several training algorithms, discussed in further detail shortly. After each training session, we will test the memory by asking it the Y for a given X. We will proceed in an ordered manner, with X starting at the lower boundary of the range and increasing until it reaches the higher boundary.

Training Algorithms

In order to carry out a simulation that is realistic, we have analyzed several training scenarios that would mimic an autonomous agent with imprecise knowledge at first that would improve as the operations continue. The training algorithms are quite different, and they are: "Greedy", "Random", "Core" and "Left to Right".

Greedy training. This type of training will feed to the fuzzy system random numbers until the FAM is just filled. The accuracy of this type of learning is highly dependent on the set of training numbers. If the set happens to include values that are right around the cores of the sets, then this algorithm will perform very well, otherwise the rules may not be the most suitable.

Random training. This training style is the one where the memory is presented with a fixed number of training pairs. If a rule has not been found for a particular association of sets, then the system will output the default value stored in the FAM. There is no assurance that the best example will be used for a certain association.

Core training. This type of training feeds to the system the values of the cores. Theoretically, once a rule reaches a point where the membership of the input value to a set is 1 (the value belongs only to one set), then the rule will not be replaced by any other one, because that sample is the best available for that set. This type of training is significant because we will be able to compare how many training sets will be necessary to achieve results similar if not equal to the results that we can achieve if the training set was the set of all core values for the input and output sets.

Left to Right training. This final algorithm for training the memory is used as control-type training. The training sets are given to the memory in an ordered manner, starting from the lower boundary of the range of the training set and ending at the high boundary. This type of training ensures that the FAM is given as many training pairs as possible, covering the entire range of the simulated environment.

Analysis of the various training algorithms

In order to evaluate the performance of the Wang-Mendel algorithm, we analyzed the performance of a hypothetical agent trained using the algorithms described above. Also, we wanted to evaluate how accurate the performance of an agent based on this algorithm is as we increase the granularity of the input and output sets associated with the fuzzification and defuzzification operations. We chose the average error as index in order to evaluate the performance of the various agents.

After the initial training that the agent receives in the form of (X, Y), where X is the input and Y is the expected output, computed by equations 1-4, depending on the scenario, when agent is then asked to return a Y value for every X it is presented. The calculations are made on the absolute difference between YEXP, the Y calculated by evaluating X by means of the equation used for training, and YFAM, the Y calculated in according to the Wang-Mendel algorithm and based on the trained agent. Each agent is tested using 201 data points. Table 1 shows the average error for all the simulations performed. The values show the average error for all agents set with a granularity between 7 input/output sets and 21 input/output sets in odd increments.

Eq	Left-to-Right	Core	Greedy	Random
1	7,61359E-15	7,61359E-15	7,61359E-15	7,61359E-15
2	9,91578E-15	9,91578E-15	0,126954513	9,91578E-15
3	0,194910328	0,189261003	0,287849826	0,203842502
4	0,539344649	0,458899104	0,747058728	0,561321949

Table 1: Average error for behaviors in Eqs 1-4, versus training algorithms

The data show that the agents learn effectively in situations where the relation between an X and a Y is linear, with no changes in the trend of the relationship, as in equation 1. The agents show a slightly greater error in a system where the linearity between X and Y is still present, but there is change, as the one associated with equation 2. We see the greatest values for error in the tests performed using equations 3 and 4. Any type of training will produce a greater error than in the case where the relationship between the two variables is linear.

Among the training algorithms we can see that the "Greedy" training is the method that yields the greatest error, because the agent starts working as soon as there is one association for each input/output region. This allows for some rules that have a very weak importance to stay in the system, and not be overwritten by stronger rules. The number of examples shown to the agent is highly dependent on how quickly the random examples cover the entire span of the input/output range.

"Random" training yields results with lower error. It is important to note that this is due to the fact that some weak rules are replaced by stronger ones found during successive observations. In this case the agent is shown 250 examples, even if all the input/output rules have already been filled.

The "Core" training shows the smallest error for the entire set of simulations. Theoretically, the strongest associations are the ones where the membership of a value to a set is closest to 1, or full membership to a set. In this particular system, the core of a TPE contains only one value, and that is the value that was used to train the agent. There is no

other example that will replace the given example, because the example given already maps to a membership of 1. In this case the agent is shown as many examples as the number of fuzzy sets that define the input/output range, thus the agents were shown between 7 and 21 examples, in odd increments.

The "Left-to-Right" training is considered a control type training because we show in a methodical manner all the examples starting from the lower boundary of the range going to the higher boundary. We have found that the error for this type of training is slightly higher than the "Core" training. The explanation lays in the fact that the example shown to the agent may not have the highest degree of importance, thus is not the best association.

Finally, the simulations related to the last situation, the one governed by equation 4, only reinforce the findings associated with what was discussed for equation 3.

The next aspect to analyze is the relationship between the average error, the type of training, and the number of sets associated with the input/output domains of the agents. Table 2 shows the results relative to this aspect of the experiment.

The results of this analysis are consistent with what would be expected. The higher the granularity of the input/output domains, the more accurate the agent's performance is. For this simulation we analyzed the experiments that were conducted using equation 3. We notice that in most cases, no matter what the training algorithm was, the average error is reduced as the number of fuzzy sets is increased. This is not true only for the "Greedy" training, as it reaches 21 fuzzy sets. The higher error is probably to trace back to the fact that the agent was shown only one example for some of the associations, without the possibility of replacing such bad example with another one with a higher significance. It is important to note that an agent performing with 21 fuzzy sets and that has been trained with a "Core" algorithm has the lowest error rate of all the simulations reported above.

Experiments

The set of simulations discussed above show that a situation where an agent is trained using the "Greedy" algorithm can be compared to a novice who is learning just enough about an environment to navigate through it, making some mistakes. As the novice learns more, thus observes more examples about the environment, we can relate to the results found in the agent trained using a "Random" algorithm. Finally, as a novice learns more about the environment and many of the original rules are replaced with others with higher importance, the novice becomes an expert. This situation is depicted in the simulations where the agent is trained using a "Core" algorithm. Moreover, we can assume that a novice's understanding of a specific domain is somewhat broad at first, thus reflecting to an agent with a low granularity for the input and output domains. As the agent goes from novice to expert, we can only assume that the understanding of the environment increases, with an increase in the granularity of the input and output ranges. This can be simulated by having an agent perform simulations first using a low number of sets, and gradually increasing the number of sets to reach the highest possible foreseen in the environment created for these experiments.

	Left-to-right	Core	Greedy	Random
7	0,627506065	0,675936524	0,650292194	0,650422003
9	0,237706944	0,237706944	0,400731338	0,230864971
11	0,197668687	0,171135865	0,252355939	0,241180776
13	0,139887356	0,138705757	0,246697204	0,154017493
15	0,108492532	0,115646522	0,249215771	0,107077011
17	0,074813892	0,064634848	0,141544446	0,096659755
19	0,112382012	0,061053242	0,167742751	0,075847614
21	0,060825138	0,04926832	0,194218968	0,074670394

Table 2: Average error for behavior in Eq. 3 across agents of different granularity and training approaches

This scenario will represent the sequence of events that we will simulate in this final round of experiments. When we introduced the concept of fuzzy mediation, we also addressed the analysis of the difference between the input of the trainer and the one of the trainee. For this simulation the difference between the inputs was normalized to [-10, 10], and then the range was broken down into the sets reported in Table 3, which also specifies the boundaries of each set.

Set name	Outer left boundary	Inner left boundary	Inner right boundary	Outer right boundary
Wide variation to the left	$-\infty$	$-\infty$	-4	-3
Slight variation to the left	-4	-3	-2	-1
Similar	-2	-1	1	2
Slight variation to the right	1	2	3	4
Wide variation to the right	3	4	∞	∞

Table 3: Boundaries of trapezoidal fuzzy sets used to compute the linguistic modifier describing the difference between the inputs of the trainer and the trainee

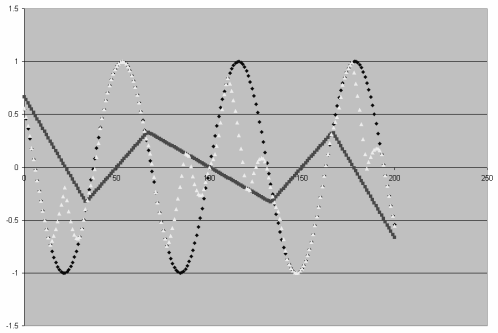
The fuzzy sets used map a value to a membership of 1 when the value falls between the two inner boundaries. If a value falls between an inner boundary and outer boundary on the same side, then it will have a partial membership to the set in question in according to the membership equation of choice. In this case, the equation chosen is a simple line equation. If a value falls outside of the outer

boundaries, then the membership to the set in question is 0. For these simulations, the maximum absolute difference between the trainer's input and the trainee's is of one unit that is then mapped to the domain described above.

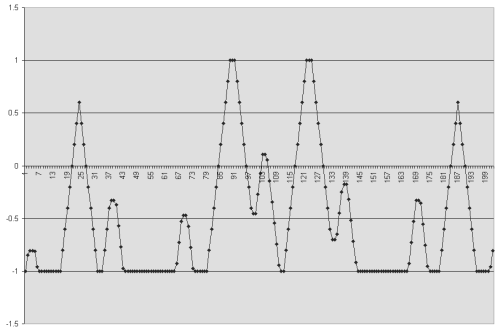
In the following simulations we will use as an index the average value of the mediator weight. As we described above, the control of an ideal object must be mediated between two controllers, a trainer and a trainee. The average value for this index shows which controller has most of the weight. A weight between –1 and 0 shows that the trainer has more control, with total control to the trainer if the weight is –1. If the weight is between 0 and 1, then the trainee has the majority of control, with control completely assigned to the trainee if the value is 1. If the value is 0 then the control is shared equally between the trainer and the trainee.

Sets	Avg. mediator weight
7	-0.517277228
11	0.603871287
21	0.945544554

Table 4: Core training for the sine environment with three different number of sets



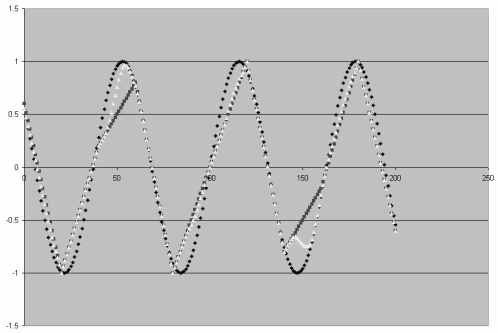
(a)



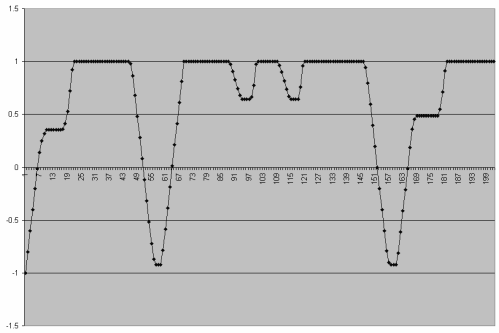
(b)

Figure 1: (a) Input and output values and (b) control mediation value for simulation using and agent with 7 sets and core training

The first aspect that we will discuss about the simulations performed is the influence of the number of fuzzy sets used for the input/output domain by the agents. Table 4 shows that, as the agent identifies the range in question with more fuzzy sets, also the weight of control shifts from a considerable average ownership of control to the trainer, as reported by the simulation with an agent based on seven fuzzy sets, to an almost complete ownership of control in favor of the trainee, as shown by the interaction with the agent based on 21 fuzzy sets.



(a)



(b)

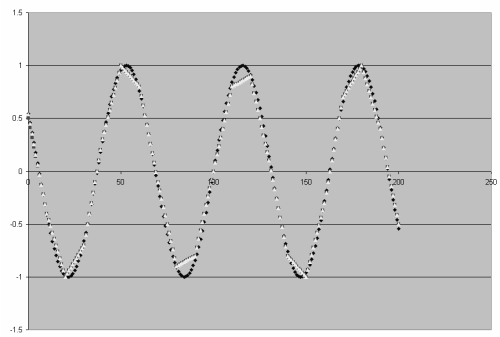
Figure 2: (a) Input and output values and (b) control mediation value for simulation using and agent with 11 sets and core training

Figures 1 through 3 show graphs reporting the input values of the trainer and the trainee as well as the mediated output. Next to each simulation we also show graphs of the value of the mediator weight after each pair of trainer/trainee input are evaluated by the fuzzy mediation system. The graphs correspond to the simulations reported in Table 4. These figures show how dynamic this process of mediation between two controllers can be.

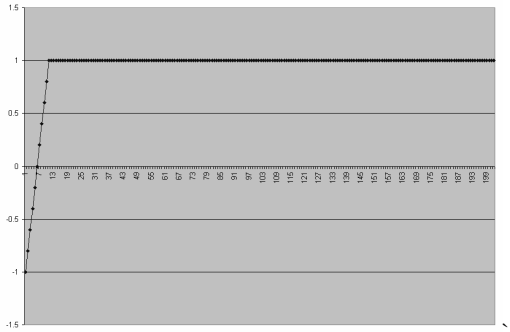
Training type	Avg. mediator weight
Greedy	0.149693069
Random	0.110108911
Core	0.603871287
L-to-R	0.300168317

Table 5: Different training algorithms for FAMs with the same amount of sets (11) within the sine environment

We also analyzed the performance of several training algorithms and the average weight of the mediator, reported in Table 5. For this particular example we chose an agent set with 11, and we can see that the agent that received "Core" training showed the best performance, keeping the average mediator value well over the side of the trainee for matters of control.



(a)



(b)

Figure 3: (a) Input and output values and (b) control mediation value for simulation using and agent with 21 sets and core training

Discussion of the results

Nowadays many domains thrive on the interaction between humans and machines. Fuzzy mediation seems the next logical step in order to destroy the barrier imposed by an automated controller that overlooks every action taken by a human before it is actually put in action. These types of systems can be found especially in the new fly-by-wire airplanes developed in recent years.

Not only will this algorithm create a middle ground for the control of vehicles, but it will also create a framework where learning is not executed by passive observation of someone else's performance, and at the same time is not as drastic as putting a novice pilot in control of a plane for the first time. Fuzzy mediation naturally uses positive reinforcement as the key that will enable trainers and trainees to interact through the first phases of the learning period, as to give the student a sense of control, but at the same time, not to relinquish control completely.

It is important to note that fuzzy mediation should allow for a paradigm shift in the type of learning environment when both operators reach a level of familiarity with the controls that is equal. At that point the fuzzy mediation algorithm will not use positive reinforcement as leverage to learning and operating a vehicle, but it will use collaborative learning, where two agents need to interact successfully to accomplish a task.

Other fields of application include monitoring systems that learn from the operator of a vehicle. The vehicle will observe the operator for a certain lapse of time, and it will continue doing so for the majority of the operational time of the machine. The vehicle will maintain the role of a trainee as long as the vehicle is operated within certain guidelines. When the boundaries for safe operations will be broken, then the trainee will be able to replace the trainer in operating the vehicle, effectively switching roles. The roles will be switched back to normal when the trainer, now trainee, will show that s/he can operate within safety guidelines for a certain amount of time.

Conclusion and Further Work

The results reported in this article make us believe that the concept of fuzzy mediation has great potential. The preliminary results show a remarkable performance using an equation as a trainer and an agent based on the Wang-Mendel algorithm as a trainee. The next step will involve the use of two human subjects as well as the use of a human interacting with an artificial agent in alternating roles, as we continue to explore the interaction between agents and the emergence of new behaviors in homogenous (Trajkovski, 2007) and heterogeneous multiagent systems.

References

Grenoble O'Malley, P. (2005, March/April). Troubleshooting Machine Control: Contractors Talk About Making the Most of 3D GPS. Grading & Excavator Contractor, 7(2).

Kokar, M., Tomasik, J. and Weyman, J. 2004. Formalizing Classes of Information Fusion Systems. *Information Fusion*. 5(3): 189-202.

Mendel, J. 2003. Type-2 Fuzzy Sets: Some Questions Answered. *Newsletter of the IEEE Neural Networks Society*. 1: 10-13.

Mendel, J. and John, R. 2002. Type-2 Fuzzy Sets Made Simple. *IEEE Transactions on Fuzzy Systems*. 10(2): 117-127.

Norman, D. A. (2005, November-December). There's an Automobile in HCI's Future. Interactions, 53-54.

Trajkovski, G. 2007. *An Imitation-Based Approach to Modeling Homogenous Agents Societies.* Hershey, PA: Idea Group.

Trajkovski, G., Schlosburg, J., Whitman, B. and Vincenti, G. 2005. Building Infrastructure for an Honors Research Robotics Lab. In Proceedings of the 6th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2005), 352-357. Towson, Maryland: IEEE Computer Society.

Vincenti, G. and Trajkovski, G. 2007. Fuzzy mediation as a novel information fusion method applied to learning-oriented systems. *Information Fusion* (submitted).

Wallace, J. (2000, March 20). Unlike Airbus, Boeing lets aviator override fly-by-wire technology. Seattle Post-Intelligencer.

Wang, L. and Mendel, J. M. 1992. Generating Fuzzy Rules by Learning from Examples. *IEEE Transactions on Systems, Man, and Cybernetics*. 22(6): 1414-1427.