Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 989051, 14 pages
doi:10.1155/2012/989051

Research Article

Fuzzy Logic Unmanned Air Vehicle Motion Planning

Chelsea Sabo and Kelly Cohen

Department of Aerospace Engineering, University of Cincinnati, Cincinnati, OH 45221, USA

Correspondence should be addressed to Chelsea Sabo, sabocm@mail.uc.edu

Received 6 February 2012; Revised 4 June 2012; Accepted 4 June 2012

Academic Editor: Rustom M. Mamlook

There are a variety of scenarios in which the mission objectives rely on an unmanned aerial vehicle (UAV) being capable of maneuvering in an environment containing obstacles in which there is little prior knowledge of the surroundings. With an appropriate dynamic motion planning algorithm, UAVs would be able to maneuver in any unknown environment towards a target in real time. This paper presents a methodology for two-dimensional motion planning of a UAV using fuzzy logic. The fuzzy inference system takes information in real time about obstacles (if within the agent's sensing range) and target location and outputs a change in heading angle and speed. The FL controller was validated, and Monte Carlo testing was completed to evaluate the performance. Not only was the path traversed by the UAV often the exact path computed using an optimal method, the low failure rate makes the fuzzy logic controller (FLC) feasible for exploration. The FLC showed only a total of 3% failure rate, whereas an artificial potential field (APF) solution, a commonly used intelligent control method, had an average of 18% failure rate. These results highlighted one of the advantages of the FLC method: its adaptability to complex scenarios while maintaining low control effort.

1. Introduction

Increasingly over the years, autonomous robots and vehicles are being used to perform missions that are considered "dull, dirty, and dangerous" in both military and civil operations, such as operations in nuclear power plants, for the exploration of Mars, to investigate behind enemy lines in battle, wild-fire surveillance, border patrols, and weather forecasting [1, 2]. As the trend develops toward the increasing use of UAVs, it becomes necessary to allocate and control them effectively. Currently, unmanned air vehicles (UAVs) are incapable of being rerouted or retasked in flight, which is crucial as the mission objectives change, threats evolve, environment changes, or where there is no prior information about the scenario. Due to these limitations, it has been identified that a main concern for UAV growth is autonomous and intelligent control [3]. From an operational perspective, it is imperative that these UAVs are able to conduct its mission with a certain level of autonomy. A fundamental ability of autonomous UAVs is motion/path planning. Furthermore, incorporating intelligence would help UAVs to act and react more like their manned counterparts.

Additionally, there are a variety of scenarios in which the mission objectives rely on a UAV being capable of maneuvering in a dynamic, changing environment or in an environment in which there is no prior knowledge of the surroundings. Many military applications depend on UAVs avoiding threats due to antiaircraft missiles or enemy warplanes. In these situations, not only are these obstacles dynamic, but there is no way to plan ahead of the mission to avoid them due to uncertainty associated with the "fog of war." An important civilian application concerns wildfires, which cause destruction of thousands of acres, millions of dollars in damage, and cost many people their lives [4]. Because forest fires can change rapidly depending on environmental conditions and become difficult to combat, people are trying to develop new ways to deal with these fires. Real-time decision-making would also be essential when combating wildfires, as situations can change rapidly due to fluctuating environmental conditions (wind, fuel, terrain, etc.) [2].

In all these scenarios, mission effectiveness is dependent upon, or could be improved by, these vehicles autonomously path planning in their environment. Because most effective path planning algorithms rely on *complete* prior knowledge

Table 1: Advantages and disadvantages of several, common path planning techniques.

Visibility graphs	Voronoi method	Approx. cell decomp.	Artificial potential fields	Fuzzy logic	
Optimal	XX				
Complex obstacles			XX	X	XX
Higher dimensions			XX	XX	XX
Computable online				XX	XX
Uncertainty		X	X	XX	XX
Finds impossibilities	XX	XX			
Ease to implement				XX	XX

of the environment, it severely limits the capabilities of UAVs. Furthermore, these algorithms cannot attain solutions for complex environments (these problems are known to be NP-hard), and fairly accurate, real-time solutions are the reasonable expectation. Therefore, a more sophisticated method to two-dimensional path planning is needed to not only allow for maneuvering in an unknown environment, but is necessary to move in a dynamic and changing environment in real time, an important ability for realistic missions. To do this, UAVs must be able to motion plan in a dynamic sense.

1.1. The Objective of This Research Is to Propose an Algorithm for Path Planning in Two-Dimensions for an UAV That Would Allow It to Operate in Real Time and in an Unknown, Obstacle Environment. In the scenario studied in this research effort, an UAV uses basic heuristics to travel to a known target while avoiding various, static obstacles. A sensor system is used to provide feedback about the obstacles in the UAV's local environment. This information is then processed at each sampling time and used as one of the factors to determine the motion of the UAV for obstacle avoidance and path planning to the target.

Given a starting point and a target location, a path planning strategy must take into account dynamic and kinematic constraints of the vehicle, as well as information about obstacles, to generate a path. Therefore, this work was done to

(1) model an unmanned air vehicle including kinematic and dynamic constraints,

(2) develop a control strategy for motion planning: path planning to a target while using sensor information about the local environment for obstacle avoidance,

(3) validate the control strategy in a MATLAB simulation,

(4) analyze and interpret contributions and limitations of the methodology.

2. Literature Review

Path-planning and motion planning algorithms encompass a variety of applications and needs. For the most part, path planning can be simplified to navigating a robot to a distance point safely, that is, avoiding collisions along the way. To make scenarios realistic, algorithms must be able to handle varying numbers of objectives, complex tasks, different constraints, and uncertainty [5]. In this research objective, the robot is a UAV (unmanned aerial vehicle), and the point can be any number of tasks (basically involves visiting spatially different locations). It is necessary that the UAV will be able to avoid obstacles and no fly zones, and in reality, these will always have some degree of uncertainty. Thus, an approach is needed that can provide the required navigational decision-making given the inherent uncertainty associated with this class of problems. In this problem statement, uncertainty occurs when the vehicle has no prior knowledge of its environment, there are dynamic obstacles, and the target itself is dynamic. While there is much research in the field of motion planning, much of it can be boiled down into several categories of approaches. Therefore, the focus of this survey is on roadmap methods like visibility graphs and Voronoi diagrams, cell decomposition, potential fields, and fuzzy logic, as it is a basis for the approach used here.

Because path planning and motion planning algorithms have been studied for many years, excellent strategies have been developed for path planning in obstacle environments. Those methods that have been studied and utilized most tend to fall under roadmap method category: visibility graphs, Voronoi diagrams, and cell decomposition. Roadmap methods create a graph that connects an initial, starting point to a final, target location around obstacles within the environment. Roadmaps utilize the following definitions: the total space in the environment in called the C_{space}, the free space not containing any obstacle is called C_{free}, and the space containing the obstacles is called C_{obstacle}. Once the environment is defined, a graph is then created that connects vertices in the C_{free} enclosed within C_{space} [6]. Finally, this map can then be searched for a path from start to finish, generating a feasible route using a brute force method or an algorithm like A* [7].

While roadmap methods have their advantages, they also have their downfalls (see Table 1). The biggest benefit to using visibility graphs is that they are optimal. This is due to the fact that all paths are formed along the edges of obstacles. However, this is one of the disadvantages of this method also. If there is any uncertainty to the obstacles size or shape, the path can easily become infeasible. Furthermore, once extended to the three-dimensional case, these solutions no longer guarantee optimality. On the other hand, Voronoi

diagrams are able to handle a little uncertainty well, because they create paths that inherently steer clear of the obstacles. However, they are not optimal and can often create paths that intuitively do not make much sense. Finally, both exact and approximate cell decomposition methods have been studied. Again, the downfalls to these methods require complete prior information and all obstacles to be polygons. While this is more practical to implement above 2D than roadmap methods, it can be difficult for very complex environments. Furthermore, if the problem is not solvable, this method will not recognize that [8]. For example, if all possible paths are blocked by obstacles, this algorithm will iteratively search for a solution (creating smaller cells and searching for a path) indefinitely.

Much focus has also been on intelligent strategies like potential fields [9–12] as they have shown excellent results for comparable situations. While these methods cannot guarantee optimality, it has shown to produce comparable results for good algorithms. Furthermore, it is excellent in the case of uncertainty and easy to extend to 3D. Also, this method is quick to compute and therefore, practical to do online. All of these benefits, and the fact that optimal solutions are computationally intractable for complex scenarios, confirm intelligent control methods as the practical solution to these problems.

Potential fields for motion planning were originally used for on-line collision avoidance for when a UAV does not have prior knowledge of the obstacle but senses them in real time. The relatively simple concept treats the vehicle as a point under the influence of an artificial potential field where the variations in the space represent the structure of the environment. The attractive potential reflects the pull of the vehicle to the goal and the repulsive potential reflects the push of the UAV from the obstacles [8]. Therefore, the environment is decomposed into a set of values where high values are associated with obstacles and low values are associated with the goal.

One of the major disadvantages to this method is the tendency to get caught in local minima, causing them to fail. This is due to the fact that the methodology basically acts as a fastest descent approach. For example, this occurs when a vehicle encounters a C-shaped obstacle. While research has been done to work around this problem, this singular problem is the main focus creating a path planning algorithm based on the artificial potential field approach [13]. For example, different techniques try inserting intermediate nodes around the obstacles to help the vehicle navigate out.

In addition to artificial potential fields (APFs) as an intelligent control method, fuzzy logic has shown excellent results for path tracking and collision avoidance for both mobile robots and UAVs [14–25]. Fuzzy logic, based on multi-valued logic, provides a unique method for encoding knowledge about continuous variables. Fuzzy logic was proposed by Zadeh [26] in 1965 as a way to more accurately capture the real world. Experience from the past decade, with the successful marketing of a wide variety of products based on the Fuzzy Logic, has shown that for certain applications this approach can lead to lower development costs, superior features, and better end product performance. Additionally, this system has the ability to utilize expert heuristic knowledge of operation of controlled systems; capacity to successfully handle uncertainties and nonlinearities; and the existence of a variety of tools that assist in studying and building efficient fuzzy systems in relatively short times. In recent times, the advantages of fuzzy logic systems have made them attractive candidates for use in expert systems. Due to these advantages, fuzzy logic excels in circumstances where the environment is continually changing and incomplete information is available.

Additionally, fuzzy logic has similar benefits to APFs (i.e., adaptability to uncertainty, near-optimal results, ability to compute online, etc.), as well as the flexibility to work around some of the issues APF methods have with local minima. Path tracking for mobile robots show high-quality solutions [16–20] using the path error as inputs and the torque of wheels as outputs. Additionally, there has been some work that uses the benefits of fuzzy logic combined with other intelligent control techniques like genetic algorithms, potential fields, and neural networks for path planning of robots [15, 22–25]. When introducing stationary obstacles, and exploiting information about these obstacles (i.e., distance and angle) path tracking using fuzzy logic for UAVs has showed promising results [14].

Motion planning in real-time is becoming an increasingly studied field as mobile robots and Unmanned Aerial Vehicles become more autonomous to keep up with growing demands [27–29]. However, many of the optimal strategies require complete information of the environment apriori [8, 30, 31] and fail in the presence of uncertainty, both requirements of autonomous vehicles expecting to operate in realistic environments. For example, Zelek [12] has shown that a robot can use potential fields to navigate around an unknown environment in real-time successfully. Furthermore, methods like rapidly-exploring random trees and probabilistic roadmap methods [27, 32] have shown successful motion planning in real-time. These methods are sample-based and create random nodes toward the general direction of the goal vertex. One downfall to most of these real-time planning techniques is they require the user to recompute a solution when new information is presented [31, 33]. Often this can be computationally difficult and not very realistic to do online (especially in complex environments).

While the literature shows the advantages of using fuzzy logic for path tracking and planning for robots and path tracking for UAVs, to the best of the author's knowledge it does not address motion planning for UAVs in real-time.

2.1. Contributions of the Work. Similar strategies used for path tracking, motion planning, and obstacle avoidance of mobile robots and UAVS using fuzzy logic are exploited here in this paper. While there has been extensive work done for mobile robots, the applications to UAVs have been limited. Therefore, this work extends previous research on path tracking with obstacle avoidance and is motivated by the fact that these methodologies have shown excellent results in the past. This strategy takes information about obstacles and

a target location to motion plan in real time. The contributions of this work can be summarized as follows.

(i) The solution takes UAV dynamics into account.

(ii) It is a robust system in that it makes no assumptions about the environment a priori. Therefore, it works well with limited or no information.

(iii) The system is not just reactionary to obstacles: it continues to path plan towards the target while avoiding obstacles. This allows for better solutions and makes it less likely to get caught in local minima.

(iv) No assumptions are made about the target location: it can be static or dynamic.

(v) Fuzzy inference systems (FISs) are fast and can be implemented online with limited onboard processor capability.

(vi) The fuzzy system has a very low failure rate.

(vii) The fuzzy system yields near optimal performance in real time.

3. Problem Formulation

The problem of path planning in two-dimensional space is formulated by generating a path between a known initial state, $O(x_o, y_o)$, and final state, $T(x_t, y_t)$. Therefore, the kinematic equations for a UAV in this circumstance are a function of the inertial position (x, y), the cruise velocity (v), and the heading angle (Θ) as shown in Figure 1.

Second-order differential equations describing the aircraft autopilot system were developed by Buzogany et al. [34]. Since we are concerned with two-dimensional path planning, the altitude response was dropped. Therefore, we are only interested in controlling the velocity and heading angle, where v_c and Θ_c are the control inputs, respectively. The motion of the UAV can be described as follows in (1) [35], where τ_v and τ_θ are the time delays associated with controlling the velocity and heading angle, respectively,

$$\dot{v} = \frac{1}{\tau_v}(v_c - v), \qquad \dot{\theta} = \frac{1}{\tau_\theta}(\theta_c - \theta). \tag{1}$$

Further development by Dong et al. [14] illustrates the UAV in the inertial reference frame (Figure 1), and the position of the UAV can be defined using the heading angle (Θ) and distance from the origin (l) as follows:

$$X = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} l\cos\theta \\ l\sin\theta \end{pmatrix}. \tag{2}$$

It follows that

$$\dot{X} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} v\cos\theta \\ v\sin\theta \end{pmatrix} \tag{3}$$

and that

$$\ddot{X} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} \dot{v}\cos\theta - v\dot{\theta}\sin\theta \\ \dot{v}\sin\theta + v\dot{\theta}\cos\theta \end{pmatrix}. \tag{4}$$

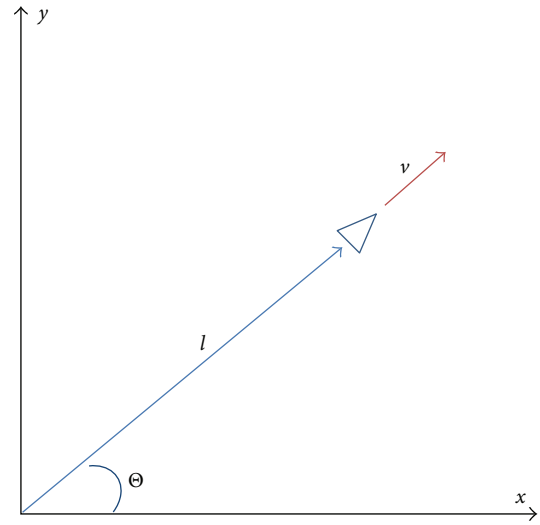

Figure 1: UAV in inertial reference frame.

By combining (1) and (4), the kinematic equations can be written in the following form:

$$\ddot{X} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} \frac{1}{\tau_v}(v_c - v)\cos\theta - v\frac{1}{\tau_\theta}(\theta_c - \theta)\sin\theta \\ \frac{1}{\tau_v}(v_c - v)\sin\theta + v\frac{1}{\tau_\theta}(\theta_c - \theta)\cos\theta \end{pmatrix}, \tag{5}$$

where the control inputs are the velocity (v_c) and heading angle (Θ_c).

Both the velocity and heading angle [34] are constrained as follows as shown below in (6) and (8), respectively. Additionally, the acceleration and heading angle rate is bounded to prevent instantaneous changes as shown below in (7) and (9), respectively,

$$v_{\min} \le v \le v_{\max}, \tag{6}$$

$$-a_{\max} \le \dot{v} \le a_{\max}, \tag{7}$$

$$-\theta_{\max} \le \theta \le \theta_{\max}, \tag{8}$$

$$-\omega_{\max} \le \dot{\theta} \le \omega_{\max}. \tag{9}$$

In this problem setup, the agent has a sensing range that is considered able to sense obstacles within $\pm 90^\circ$ and within a certain sending radius as shown in Figure 2.

3.1. Assumptions. For the purpose of this investigation, several simplifying assumptions were made without taking away from the applicability of the developed approach. While the control methodologies developed in this research can be extended to be used in three dimensions, it is assumed that all motion is in two dimensions; that is, the altitude is removed as a degree of freedom (altitude is constant). It is also assumed that wind is negligible (the UAV cruise speed is much greater than wind speed). Furthermore, the aircraft is a point mass, and therefore, no moment effects are considered. Additionally, the dynamic constraints of the aircraft are assumed to be known.

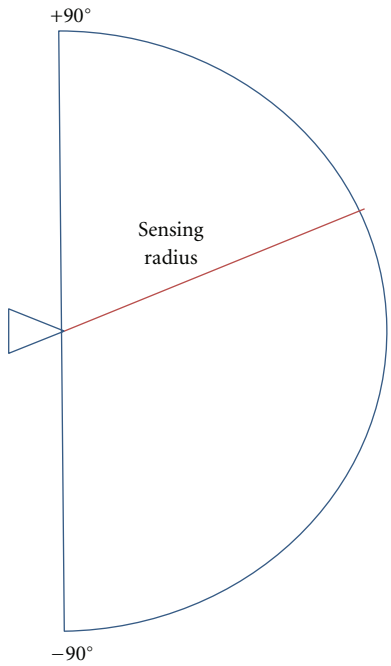

Figure 2: Sensing radius of agent.

To isolate the performance of the control methodology, several assumptions were made on the capabilities of the sensor system. The agent is assumed to be able to detect obstacles within its sensing range in real time and define them accordingly. The capabilities of sensors are discussed briefly by Mujumdar and Padhi [27], and while further work needs to be done in this area to extract information from the sensors accurately, this is a field of extreme interest for ongoing research.

The position of the UAV and the target location is assumed to be known by a GPS (global positioning system) and to be accurate. Furthermore, this information is updated at a reasonable sampling time. Additionally, the start location of the UAV and the target positions are given by some outside planner (known GPS locations), and there are no obstacles at these locations. Finally in this work, it is assumed that a feasible solution exists. That is, the UAV can navigate around the obstacles safely to the target location. This includes the assumption that the target location is a "safe" distance away from an obstacle.

4. Fuzzy Methodology

Due to the advantages that fuzzy logic presents in circumstances where the environment is continually changing and incomplete information is available, it is the basis behind the control strategy presented here for motion planning. In this section, the overall control system is described in detail, including control inputs and outputs, the decision-making rule base, and defuzzification calculation, and the methodology used for practical implementation issues.

Like previously discussed, fuzzy logic manipulates inputs using heuristic knowledge and typically converts them to outputs in three stages: fuzzification, decision-making logic, and defuzzification. The fuzzy logic controller can be visualized in Figure 3. Fuzzification takes analog inputs and converts them to a continuous value between 0 and 1 based on their degree of membership in each function. A knowledge base is then used to form a set of rules relating the inputs and outputs in the form of if-then statements. The outputs are finally converted back to crisp numbers using a defuzzification method.

4.1. Input and Output Membership Functions. For this problem setup, four inputs are used for the fuzzification interface and two outputs are given after defuzzification. Inputs into the system are as follows: distance from the UAV to the obstacle, angle between the UAV and the obstacle, the distance to the target, and the error between the current heading angle of the UAV and the angle of the target in relation to the inertial reference frame. The obstacle inputs were used by Dong et al. [14] for fuzzy path tracking and showed good results for obstacle avoidance. The target inputs were chosen based on the assumption that there is only minimal amount of information about the target; that is, GPS coordinates. The outputs for the system were also used by Dong et al. [14] and based on much literature that uses these as the control inputs to a UAV system.

The distance to the obstacle (Figure 4) is described by four membership functions: Close, Medium, Far, and Very Far (Out of Sensing Range). The angle between the obstacle and the UAV (Figure 5) is described by six membership functions: Negative Big, Negative Medium, Negative Small, Positive Small, Positive Medium, and Positive Big. Similar to the obstacle distance, the distance to the target (Figure 6) is described by three membership functions: On Top, Medium, and Far. Finally, the error between the heading angle and the target angle (Figure 7) is described by seven membership functions: Negative Big, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium, and Positive Big.

With these inputs and a rule base, the control input for the system is obtained. That is, the outputs of the fuzzy inference system, the percent of the maximum velocity and the heading angle change, are used as the control into the system. Therefore, the outputs of the FIS are the percent of maximum velocity and the heading angle change. The Mamdani method is used here, which expects the output membership functions to be fuzzy sets (as opposed to crisp values). The output velocity (Figure 8) can be broken into four membership functions: Very Slow, Slow, Fast, and Very Fast. The output angle change (Figure 9) is parallel to the target angle. Therefore, there are seven membership functions described by: Negative Big, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium, and Positive Big.

4.2. Decision-Making Rule Base. Rules relating the inputs and outputs for the fuzzy logic controller are set up in the form of if-then statements and are based on heuristics and human experience with navigating through an environment

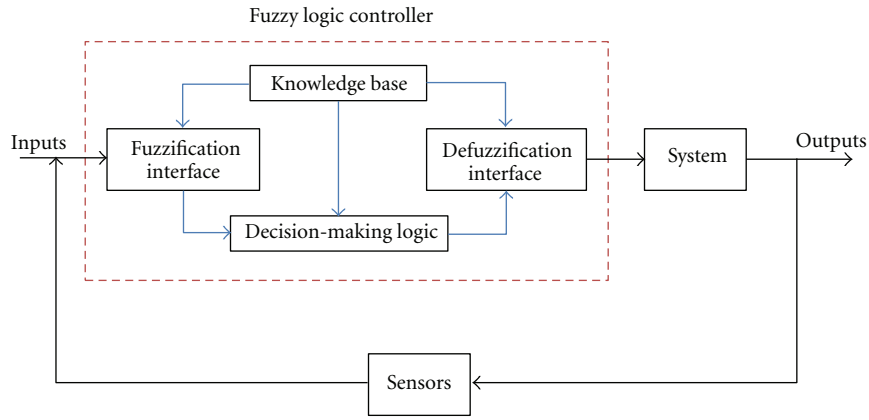

Figure 3: Fuzzy logic control system.

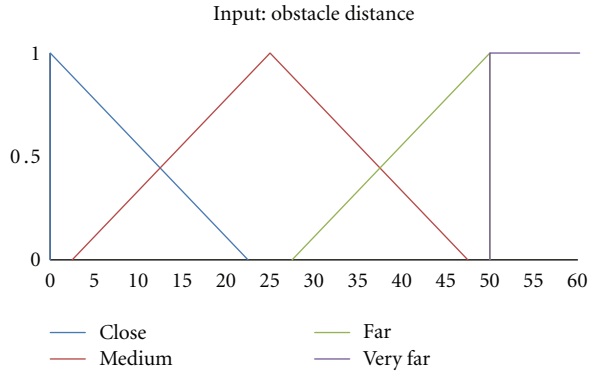

Figure 4: Input: distance between obstacle and agent (very far is for an obstacle out of sensing range).

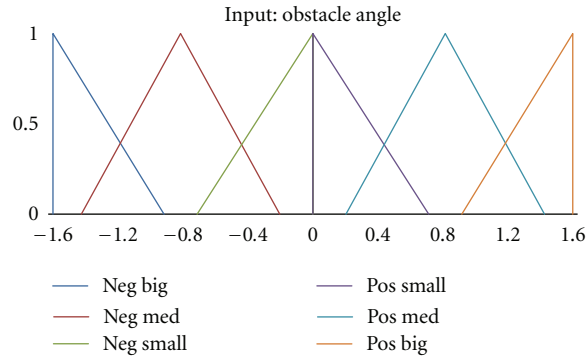

Figure 5: Input: angle between obstacle and agent heading angle.

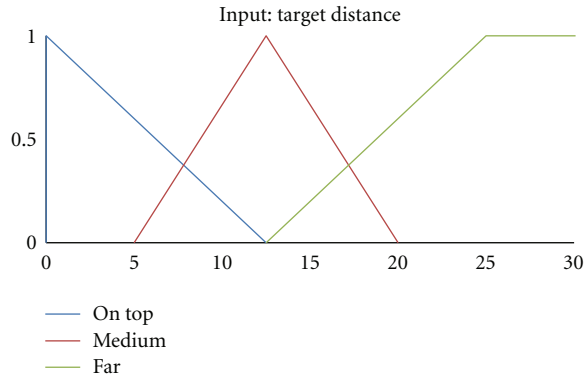

Figure 6: Input: distance between target and agent.

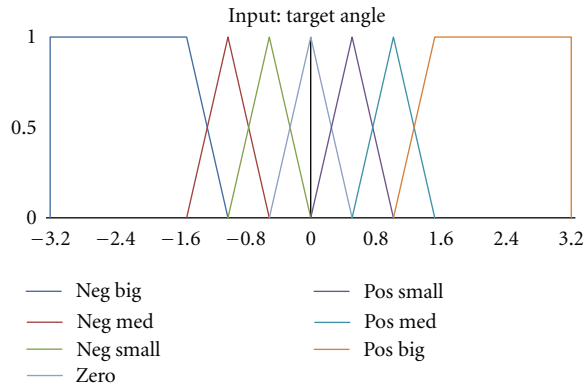

Figure 7: Input: angle between target and agent heading angle.

(similar to driving a car). The rules for the fuzzy inference system can be summed up in some simple decision-making logic. There are a total of 40 rules for this setup, and the rules can be broken up into two situations: if there is an obstacle within the sensing range or not. Once the rules were formed, the membership functions were tuned based on the distance travelled when compared to optimal solutions until the quality of the solutions reached a plateau.

The main objective of the controller when there is no obstacles within its sensing range becomes to plan a direct path to the agent's primary target. In this case, the distance to the obstacles is set to "very far" away (much larger than the sensing radius). Path planning is then done by altering the UAV's heading angle to match that of the angle of the target in the inertial reference frame or by driving the error between the two angles to zero. For the cases in which there are no obstacles in range and the target is very far away, the UAV

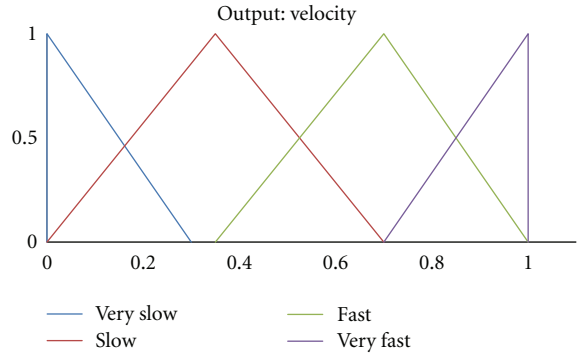

Figure 8: Output: percentage of maximum velocity.

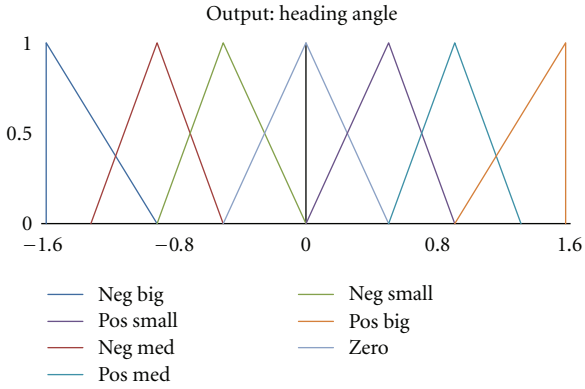

Figure 9: Output: heading angle deviation.

tends towards its maximum operating speed. When the agent reaches its target location, the agent alters its velocity to slow down and complete mission objectives while still driving the angle error to zero. Again, these rules stem from the rules used to drive a car, and can be summarized as follows.

(1) *If* D_o = Very Far *and* D_t = Far, *then* V = Very Fast.

(2) *If* D_o = Very Far *and* D_t = Med., *then* V = Slow.

(3) *If* D_o = Very Far *and* D_t = On Top, *then* V = Very Slow.

(4) *If* D_o = Very Far *and* Θ_t = Neg. Big, *then* Θ_c = Neg. Big.

(5) *If* D_o = Very Far *and* Θ_t = Neg. Med., *then* Θ_c = Neg. Med.

(6) *If* D_o = Very Far *and* Θ_t = Neg. Small, *then* Θ_c = Neg. Small.

(7) *If* D_o = Very Far *and* Θ_t = Zero, *then* Θ_c = Zero.

(8) *If* D_o = Very Far *and* Θ_t = Pos. Small, *then* Θ_c = Pos. Small.

(9) *If* D_o = Very Far *and* Θ_t = Pos. Med, *then* Θ_c = Pos. Med.

(10) *If* D_o = Very Far *and* Θ_t = Pos. Big, *then* Θ_c = Pos. Big.

When obstacles are detected within the sensing range, the agent alters its velocity and heading angle using information about obstacle distance, obstacle angle, and target location to avoid, and then recover, from the obstruction. The agent must slow down and change course to avoid it (again, similar to driving a car). This involves driving the heading angle error to around 90°. Once it is clear of this obstacle, it can continue its path toward the target. The set of rules that describe the change in heading speed and angle when an obstacle is detected can be summed up below:

(11) *If* D_o = Far *and* Θ_o = Pos. Med, *then* V = Very Fast *and* Θ_c = Zero.

(12) *If* D_o = Far *and* Θ_o = Pos Small, *then* V = Fast *and* Θ_c = Neg. Small.

(13) *If* D_o = Far *and* Θ_o = Neg. Small, *then* V = Fast *and* Θ_c = Pos. Small.

(14) *If* D_o = Far *and* Θ_o = Neg. Med, *then* V = Very Fast *and* Θ_c = Zero.

(15) *If* D_o = Medium *and* Θ_o = Pos. Big, *then* V = Fast *and* Θ_c = Zero.

(16) *If* D_o = Medium *and* Θ_o = Pos. Med., *then* V = Slow *and* Θ_c = Neg. Small.

(17) *If* D_o = Medium *and* Θ_o = Pos. Small, *then* V = Slow *and* Θ_c = Neg. Med.

(18) *If* D_o = Medium *and* Θ_o = Neg. Small, *then* V = Slow *and* Θ_c = Pos. Med.

(19) *If* D_o = Medium *and* Θ_o = NM, *then* V = Slow *and* Θ_c = Pos. Small.

(20) *If* D_o = Medium *and* Θ_o = NB, *then* V = Fast *and* Θ_c = Zero.

(21) *If* D_o = Close *and* Θ_o = PB, *then* V = Slow *and* Θ_c = Neg. Small.

(22) *If* D_o = Close *and* Θ_o = PM, *then* V = Very Slow *and* Θ_c = Neg. Med.

(23) *If* D_o = Close *and* Θ_o = PS, *then* V = Very Slow *and* Θ_c = Neg. Big.

(24) *If* D_o = Close *and* Θ_o = NS, *then* V = Very Slow *and* Θ_c = Pos. Big.

(25) *If* D_o = Close *and* Θ_o = NM, *then* V = Very Slow *and* Θ_c = Pos. Med.

(26) *If* D_o = Close *and* Θ_o = NB, *then* V = Slow *and* Θ_c = Pos. Small.

The last set of rules is for when obstacles are at extreme angles (D_0 = Far) and pose no threat of collision. In these conditions, the UAV can head toward the target but at a reduced speed (to avoid making sharp turns towards an obstacle). This was done so that motion planning to the target is the priority when obstacles are not in range, and that avoiding obstacles is the priority when obstacles are in range.

(27) *If* D_o = Far *and* Θ_o = Pos. Big *and* Θ_t = Neg. Big, *then* V = Fast *and* Θ_c = Neg. Big.

(28) *If* D_o = Far *and* Θ_o = Pos. Big *and* Θ_t = Neg. Med, *then* V = Fast *and* Θ_c = Neg. Med.

(29) *If* D_o = Far *and* Θ_o = Pos. Big *and* Θ_t = Neg. Small, *then* V = Fast *and* Θ_c = Neg. Small.

(30) *If* D_o = Far *and* Θ_o = Pos. Big *and* Θ_t = Zero, *then* V = Fast *and* Θ_c = Zero.

(31) *If* D_o = Far *and* Θ_o = Pos. Big *and* Θ_t = Pos. Small, *then* V = Fast *and* Θ_c = Pos. Small.

(32) *If* D_o = Far *and* Θ_o = Pos. Big *and* Θ_t = Pos. Med, *then* V = Fast *and* Θ_c = Pos. Med.

(33) *If* D_o = Far *and* Θ_o = Pos. Big *and* Θ_t = Pos. Big, *then* V = Fast *and* Θ_c = Pos. Big.

(34) *If* D_o = Far *and* Θ_o = Neg. Big *and* Θ_t = Neg. Big, *then* V = Fast *and* Θ_c = Neg. Big.

(35) *If* D_o = Far *and* Θ_o = Neg. Big *and* Θ_t = Neg. Med, *then* V = Fast *and* Θ_c = Neg. Med.

(36) *If* D_o = Far *and* Θ_o = Neg. Big *and* Θ_t = Neg. Small, *then* V = Fast *and* Θ_c = Neg. Small.

(37) *If* D_o = Far *and* Θ_o = Neg. Big *and* Θ_t = Zero, *then* V = Fast *and* Θ_c = Zero.

(38) *If* D_o = Far *and* Θ_o = Neg. Big *and* Θ_t = Pos. Small, *then* V = Fast *and* Θ_c = Pos. Small.

(39) *If* D_o = Far *and* Θ_o = Neg. Big *and* Θ_t = Pos. Med, *then* V = Fast *and* Θ_c = Pos. Med.

(40) *If* D_o = Far *and* Θ_o = Neg. Big *and* Θ_t = Pos. Big, *then* V = Fast *and* Θ_c = Pos. Big.

The defuzzification stage takes the outputs from the if-then rule base and converts it to a crisp number. Because the Mamdani method is used here [36], outputs are also in the form of membership functions. Once the total area from all the rules is summed, an appropriate method for determining a crisp output is chosen. Here, the centroid method, a common technique for defuzzification, is used.

4.3. Implementation Issues. One of the disadvantages to an intelligent control method like artificial potential fields for path planning is the tendency to get caught in local minima. For fuzzy logic, difficulty can occur at symmetrical or large, concave obstacle. However, the advantage to using fuzzy logic is that it is adaptable to incorporating additional rules to circumvent these issues. In this circumstance, the FIS uses only one point as the input to calculate the distance to the obstacle and always uses the closest point to the obstacle. Due to this, additional logic must be included.

To avoid this issue, a simple logic was implemented and verified to redefine obstacles that come within the agent's sensing range. If the agent appropriately evaluates the obstacle within the sensing range, it can overcome these issues. That is, if the agent encloses the areas within a certain minimum, "safe" radius (i.e., "redefines" its environment and what it sees as an obstacle), it can navigate out of the situations that cause these minima.

The mathematical formulation is as follows: if the $\Re^2$ global map is discretized, $O = \{O_1, O_2, \ldots, O_n\} \subset \Re^2$ is the set of polygon (both concave and convex) obstacles where E_O makes up the obstacle edges. In the discretized space, the E_O obstacle edges can be described by a set of points $P = \{p_1, p_2, \ldots, p_m\}$ that make up the edges.

Let (x_V, y_V) denote the position of the UAV at a given time. Then, the set of points that are within the sensing range, r_s, of the UAV is described by $P_I = \{p_{I1}, p_{I2}, \ldots, p_{Il}\}$, where $P_I \subset P$ for which the following is true:

$$\sqrt{\left(p_{Il,x} - x_V\right)^2 + \left(p_{Il,y} - y_V\right)^2} \leq r_s. \tag{10}$$

For this set of obstacle points within the UAV's sensing range, the agent then redefines the space as follows: for 1 *to* l, where $i, j \in l$ and $i \neq j$, if

$$\sqrt{\left(p_{Ii,x} - p_{Ij,x}\right)^2 + \left(p_{Ii,y} - p_{Ij,y}\right)^2} \leq r_f, \tag{11}$$

where r_f is the "safe" distance from the UAV to an obstacle, then $P_N = \{p_{N1}, p_{N2}, \ldots p_{Nk}\}$ becomes a set of points that creates a new obstacle edge between points p_i and p_j, where the set of points P_N is defined by

$$m = \frac{\left(p_{Ii,y} - p_{Ij,y}\right)}{\left(p_{Ii,x} - p_{Ij,x}\right)}, \qquad b = p_{Ii,y} - p_{Ii,x} \cdot m,$$
$$y_k = m \cdot x_k + b \quad \text{for } x_k = p_{Ii,x} \text{ to } p_{Ij,x}. \tag{12}$$

While this is relatively simple logic, it holds under several assumptions and conditions. This holds as long as the minimum "safe" radius is large enough in comparison to the vehicle minimum turning radius (often a valid assumption). Therefore, if the sensing range is large, the vehicle will have enough time to turn to avoid the obstacle.

This logic holds for not only concave obstacles, but also for convex obstacles. If a concave obstacle is too small for the UAV to turn out of, the algorithm would define the whole area as an obstacle. This would make sense since the UAV would not be able to navigate in and out of the obstacle, and, therefore, the UAV will never make its way into the obstacle in the first place. For convex obstacles, the agent would just be redefining an area inside the obstacle (an area that the agent would already be unable to reach).

If the concave obstacle is large enough for the UAV to venture into, the UAV will be able to redefine the obstacle and navigate out, even with a minimum turn radius. This is shown in Figure 10, where the blue is the sensing range and red is the obstacle (both original and redefined as it carries on).

For symmetrical obstacles, the UAV redefines as the area between the two as an "obstacle." Therefore, as the UAV approaches the obstacle and it comes into view, if multiple pieces (or two different obstacles here) appear in the sensing range and the distance between them is less than a safe distance, the UAV defines the entire distance between them as an obstacle. This implies that the area is "unsafe" to travel through, and the UAV would navigate around the two obstacles as though they were one. Several steps that the agent would take in this scenario are shown in Figure 11.

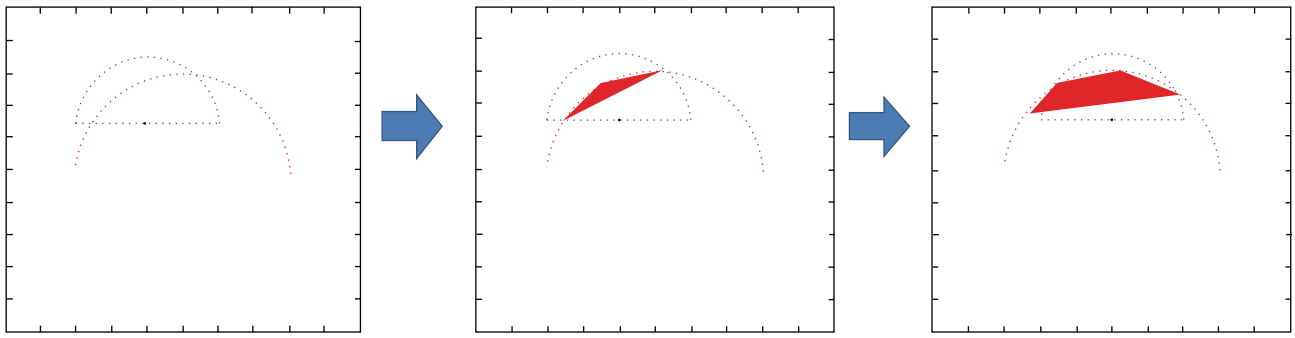

Figure 10: Agent Redefining a Large, Concave Obstacle.

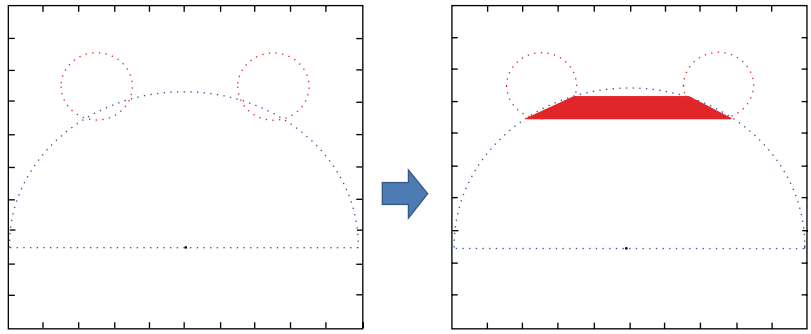

Figure 11: Agent Redefining Two, Symmetrical Obstacles.

5. Simulation and Results

In this section, the simulations performed and their results are described. Seeing as potential fields have been used frequently as an intelligent control method for motion planning, it the basis for comparison for methodology developed here. Furthermore, the fuzzy logic controller is compared to an optimal path planning algorithm that uses visibility graphs. This was done to analyze the overall performance of the FLC. In this section, the testing environment used in the Monte Carlo runs is described. All simulations described and presented here were implemented in MATLAB programming language and on a Windows machine with an Intel Core i5 2.4 GHz processor and 4.0 GB of RAM. Finally, results comparing methodologies are presented.

5.1. Simulation. The testing environment is an enclosed grid where each vertex is identical and indicates the presence of an obstacle. If there is an obstacle at that vertex, it is marked as 1. If not, it is marked as 0. Therefore, all obstacles are constructed as a series of vertices. The vehicle is assumed to be able to sense and define obstacle appropriately. That is, the vehicle searches its sensing range at each time step for obstacles and if it comes upon a vertex defined as a 1, it knows it is an obstacle. While this is a simple representation of an environment, it has several useful advantages that are exploited. First, the FLC utilizes the closest point of the obstacle within the sensing range, and this is much easier to compute if the environment is discretized. Furthermore, the volume of the space that is composed of obstacles can help define the complexity of the environment.

Both polygonal and nonpolygonal obstacles are tested in this research, but the nonpolygonal obstacles are only used for the validation of the methodology. This is due to the fact that the optimal approach used for comparison is capable of handling only polygons.

Again, Monte Carlo testing was completed to evaluate the performance of the solutions obtained. Each simulation contained a fixed number of polygon obstacles, a fixed starting position for the UAV, and a fixed number of targets. There were three types of obstacles the total, the size, and shape of each of the three were fixed. In each simulation, the obstacle type (so it could be any one of the three types), the location of the targets, and the location of the obstacles were all random. Three categories of obstacle density were used for comparing each method, and 300 runs of each of the three categories of obstacle density were used (900 simulations total). Simple (least dense) cases had 5 obstacles and an obstacle area percentage of less than 30%, moderate (medium density) had 10 obstacles and an obstacle area of less than 50% and greater than 30%, and complex (very dense) had 15 obstacles and an obstacle area of greater than 50%. The reason that the obstacle area was not consistent throughout the simulations is that the random aspect of the location of the obstacles allowed them to overlap.

5.1.1. UAV Operating Envelope. In the simulations described here, the parameters of the UAV maneuverability are summarized in Table 2. These parameters are described in

Table 2: UAV and simulation parameters.

Parameter	Value
r_s	50 m
r_f	25 m
$v_{\max}$	10 m/s
$v_{\min}$	3 m/s
$a_{\max}$	3 m/s^2
$\theta_{\max}$	30°
$\omega_{\max}$	25 rpm or 2.618°/s
τ_θ	0.4 secs
τ_v	4 secs
T	0.05 secs
T_{exec}	Time step, k, that last target is serviced

the previous section as the constraints of the UAV kinematics [30, 35]. In Table 2, r_s is the sensing range of the UAV, r_f is the safe range for a UAV from an obstacle, and T is the sampling rate of the UAV, target, and obstacle information.

5.1.2. Model Discretization. In addition to discretizing the environment, the UAV dynamics described earlier were discretized. The sampling rate, T, is the time that the UAV position, UAV heading angle, UAV velocity, UAV sensing range, and target location were updated. Furthermore, a step in the simulation time is represented by $k = 0, 1, 2\ldots$, each cell in the environment is equal to 1 m, and area of each map is 3.5 km × 3.5 km. Therefore, the model is updated according to the following:

$$\begin{aligned} x[(k+1)T] &= x(kT) + Tv(kT)\cos\theta(kT), \\ y[(k+1)T] &= y(kT) + Tv(kT)\sin\theta(kT), \\ \theta[(k+1)T] &= \theta(kT) + T\omega(kT), \\ v[(k+1)T] &= v(kT) + \frac{T}{\tau_v}[v_c(kT) - v(kT)], \\ \omega[(k+1)T] &= \omega(kT) + \frac{T}{\tau_\theta}[\theta_c(kT) - \theta(kT)], \end{aligned} \tag{13}$$

where the difference between the control input and the actual need to satisfy the constraints for acceleration and angular acceleration (14) is as follows

$$\begin{aligned} |v_c(kT) - v(kT)| &\le a_{\max}, \\ |\theta_c(kT) - \theta(kT)| &\le \omega_{\max}. \end{aligned} \tag{14}$$

5.1.3. Performance Metrics. Monte Carlo experiments were completed to analyze the performance of the different methods and gain insight to the contributions and limitations of the control method presented here. To do so, various performance metrics were calculated for each run. The performance error with respect to the optimal solution was calculated for the distance travelled as this is a common performance metric for path planning algorithms. Also, the control effort was computed as the sum of the square of the heading angle change at each time step. The heading angle rate was used to determine a quantitative metric for the required path changes of each control method. Finally, the number of errors was recorded over all the simulations to determine failure rate. A failure was considered if the UAV never reached the target and/or the UAV "crashed" into an obstacle. For each of the three control methodologies implemented, the previous was recorded where fuzzy logic control has subscript *flc*, potential field method has subscript *pf*, and optimal has subscript *o*. These values and their calculation are shown in Table 3.

5.2. Results. In this chapter, the results of testing the Fuzzy Logic Controller are presented. This includes validation of the methodology and analysis of the performance. Therefore, it is shown here that the system meets the specified constraints; is capable of operating in complex environments and capable of target retasking in-flight; performs excellent when comparing the distance traversed and control effort to optimal and potential field solutions. It is shown that the FLC navigates to the targets in real time while avoiding obstacles. Like mentioned previously, all environments were formed such that a plausible solution could be found. This means that the UAV can adequately navigate around obstacles and that targets are at a "safe" distance from the obstacles.

5.2.1. Constraint Verification. Beyond validation that the technique allowed for motion planning in an unknown environment, it was essentially that this logic works within the specified constraints of the vehicle. Therefore, examples were run to verify that the system was doing so. Shown below is an example of the fuzzy logic controller (Figure 12) and the corresponding acceleration (Figure 13) and heading angle rate (Figure 14). It was verified that the system never operated outside the constraints.

Furthermore, the controls for this example are shown (Figures 15 and 16). It is seen that there is a short time delay for both the velocity and heading angle change as expected.

5.2.2. Performance. As described in the previous section, the fuzzy inference system was compared to both an optimal path planning methodology and an artificial potential field method. As potential fields are widely used and the leading intelligent control method for dynamic path planning, it is the technique used for comparison here. Potential fields have been shown to provide near-optimal solutions and a viable alternative to optimal control strategies. A parabolic function is adopted to represent the artificial potential on the vehicle from an object where the function is quadratic in the Euclidean distance (so that it is differentiable) and multiplied by a scaling factor (differs for attractive and repulsive forces). From the gradient of the function, the force on the vehicle is calculated.

The methodology used here for optimal control solution is a roadmap method, visibility graph, and assumes the environment is completely known prior to the mission and that it is unchanging throughout. Again, it is implemented as a comparison for performance (i.e., distance traversed).

Table 3: Control solution parameters.

Parameter	Symbol	Calculation
Distance traversed	D	$\sum_{k=0}^{T_{\text{exec}}} \sqrt{(x[k+1]-x[k])^2+(y[k+1]-y[k])^2}$
Control effort	C	$(1/T_{\text{exec}}) \sum_{k=0}^{T_{\text{exec}}} \omega(k)^2$
Failure rate	F	# of failures/300
Obstacle area	OA	Area of obstacles (km^2)/12.25 (km^2)
Optimality	J	$D_{\text{pf}}/D_o \quad D_{\text{flc}}/D_o$

Table 4: Performance results—percentage of failures and control effort.

	Failures: simple environments	Failures: moderate environments	Failures: complex environments	Average control effort
Optimal	0.00	0.00	0.00	0.0158
FLC	0.02	0.03	0.05	0.0621
PF method	0.03	0.19	0.34	0.5169

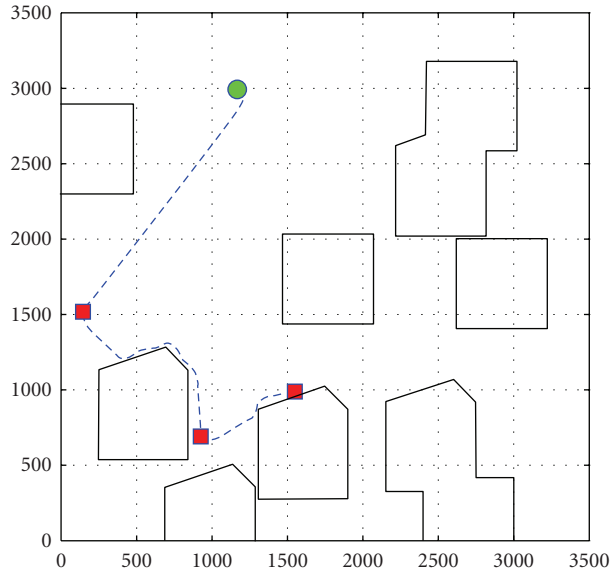

Figure 12: Example of the fuzzy logic controller for constraint verification.

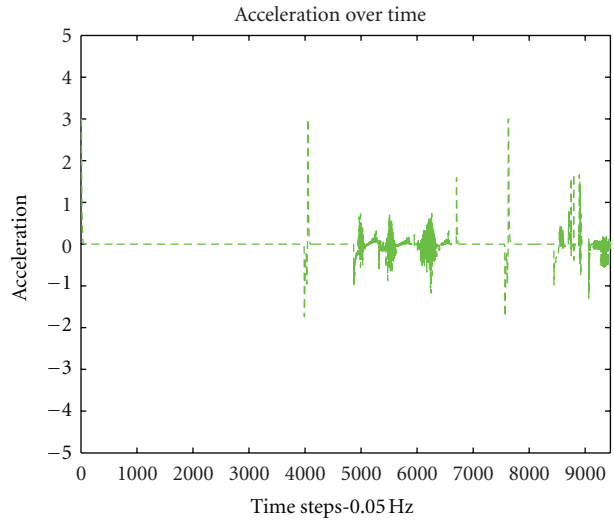

Figure 13: Acceleration versus time of the example for constraint verification.

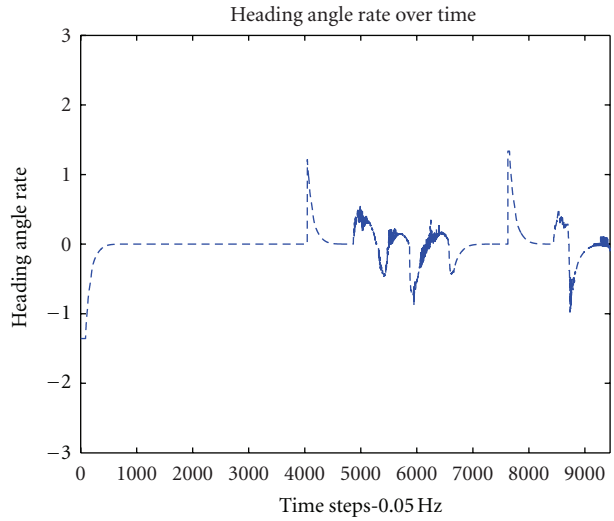

Figure 14: Heading angle rate versus time of the example for constraint verification.

Since the complete environment is known a priori, no-fly zones and other obstacles can all be represented as polygons.

While it is important that the FLC is at least somewhat comparable to other methods in terms of distance, the most significant contribution of this method is the reduced failure rate. It can be seen in Table 4 that the percentage of failures for the PF method (average of about 18% overall) over the FLC (average of about 3% overall) is significantly higher. Furthermore, as the environment gets increasingly complex, the number of failures for the PF method is about 1/3 of the total cases (as opposed to 5% for the FLC). These high numbers of failures for the PF method (especially for complex environments) make it unreliable to implement realistically. Additionally, although the optimal solution has no failures, it should be noted that if there is any uncertainty in the environment this percentage would increase dramatically. On the other hand, it is likely that the fuzzy logic

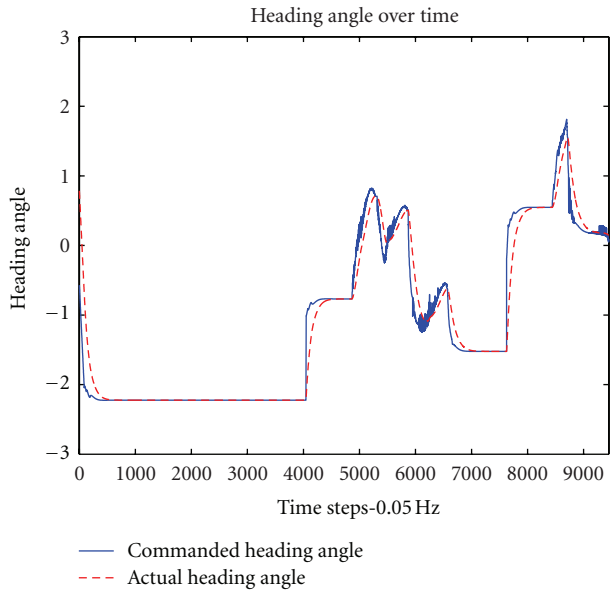

Figure 15: Controlled heading angle and actual heading angle versus time of the example.

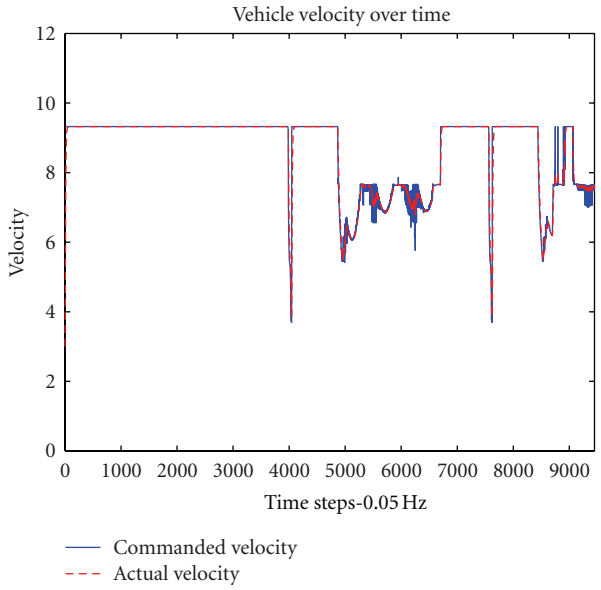

Figure 16: Controlled velocity and actual velocity versus time of the example.

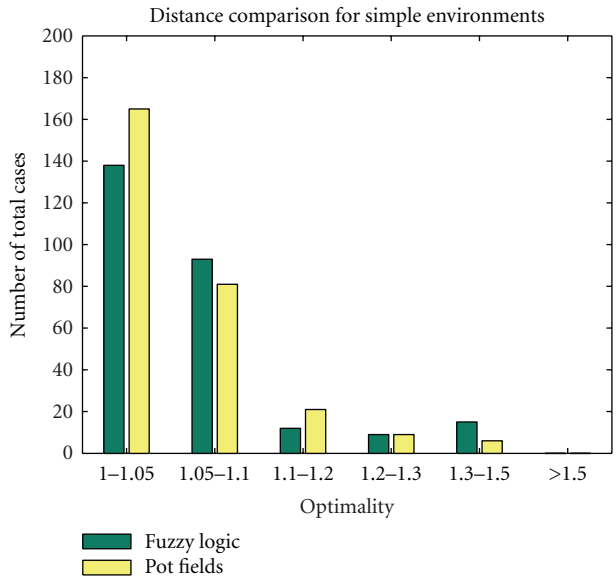

Figure 17: Extra distance traversed for the approximate solutions in simple (least dense) environments.

controller would perform the same as it makes no assumptions about the environment a priori.

Also, the control effort across methods is compared. The effort shown in Table 4 is the average over all simulations for each method. As can be seen, the FLC outperforms the PF method by almost 10 times. Additionally, the FLC is on the same order of magnitude as the optimal in terms of the control effort. This detail, as well as the significantly less failures, makes it a feasible approach to utilizing in real time.

Because path planning algorithms are generally used to solve for the minimum distance travelled, this was used for comparison for performance and as a measure to tune the membership functions. This is also done because the fuzzy inference system developed for path planning only relies on local information and a set of coordinates for the target, it is likely that the UAV will not always take the optimal path (and could possibly take a very suboptimal path). The question becomes as to how suboptimal this path is, and how often these very suboptimal solutions take place. That is, if hundreds of random obstacle environments are produced, how much further the UAV uses the FIS travelling than the branch-and-bound method. Also of interest is how much longer the UAV take to traverse its path (not necessarily the same as the distance for the FIS and PF method, because the velocity is not constant). As mentioned previously, the environments were broken down into three categories of density: simple, moderate, and complex. Figures 17, 18, and 19 show the number of cases (out of 300 for each category) as the optimality of the solution decreases in terms of distance traversed for each category, respectively.

As can be seen in the previous plots, the fuzzy logic solutions perform very well when compared to optimal solutions. 66% of cases lie within 90% of the optimal in terms of distance traversed. Also, all of the cases have an average of being within 7.6% over all cases (as compared to 9.7% for the PF method). It should also be noted that the optimal solution does not take vehicle dynamics into consideration and assumes that it can travel along the edge of an obstacle. Realistically, the optimal solution would be altered slightly to allow for a little "padding" around the obstacles. The fuzzy logic solution already does this. Therefore, the fuzzy logic solution is closer to the optimal cases than is indicated here.

6. Conclusions and Future Work

Presented here is the formulation and validation of a fuzzy logic controller (FLC) for motion planning in real-time

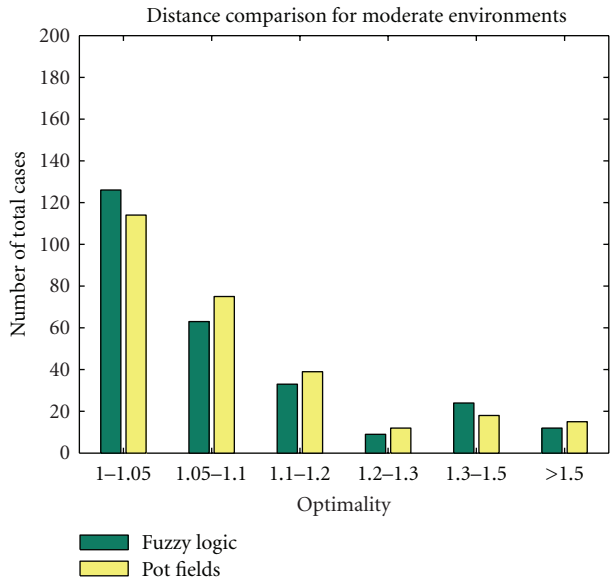

Figure 18: Extra distance traversed for the approximate solutions in moderate (medium density) environments.

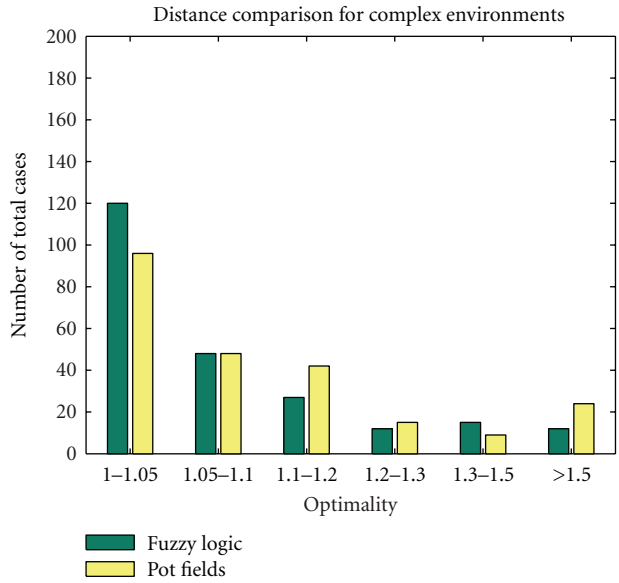

Figure 19: Extra distance traversed for the approximate solutions in complex (very dense) environments.

in a two-dimensional, unknown environment. The fuzzy inference system was verified on various, stationary obstacles and for moving targets (retasking mid flight). The FIS was compared to an optimal approach and an algorithm based on potential fields (PFs) that also motion plans in real time.

The comparison to other path planning methods showed that the FLC gets good results, and they can be obtained online with any type of environment. This was shown to be very reliable, with only about a 3% error rate over all cases. The errors that did occur were when the UAV navigated into an area that did not have an outlet, and it could not appropriately maneuver out. While this method is not faultless, the APF method showed a much higher failure rate at about 18% overall (34% for complex environments). Additionally, the APF method used significantly more control effort (about 10 times more overall) than the FLC.

It was also shown that in most cases when the environment is relatively simple (only contains polygons or "closed" obstacles) the FIS produces near-optimal solutions consistently. That is, the FLC was able to perform within 5.7% of the optimal solution in simple environments, within 10.7% for moderate environments, and within 6.5% for complex environments for distance travelled (7.7% overall). This beat the overall performance for the APF method at 9.7%. While the performance is less reliable for obstacle enriched environments, the results are still able to be obtained in real time and with reasonable closeness to optimality. As environments get increasingly complex, the optimal solution will be harder to obtain in a reasonable time. This will make it impossible to rely on acquiring an optimal solution even if the environment is completely known prior to flight.

The results presented in this work show that the FL algorithm outperforms the APF method in reliability of success of a solution, in near-optimality, and amount of control effort used. What is more, it was able to do so in real time with local information, which the optimal solution was not. While quantitatively it is obvious that the FLC outperforms the APF and optimal methods, there are several qualitative advantages to the fuzzy controller. The reduced failure rate is a distinguishing attribute of fuzzy logic. Fuzzy Logic allows the user to capture much more information about the environment in a more efficient manner. Additionally, the available tools allow the user to easily develop and manipulate FISs. This makes it a very powerful tool that allows the user to see the effects of incorporating additional information with minimal effort.

The advantages of using Fuzzy Logic (FL) for a motion-planner were demonstrated in this research: few failures, near-optimal paths, and low control effort. It is these advantages that make it a good tool for further development. Some possible obvious future works that were not explored in this research are motion-planning in three dimensions, avoidance of dynamic obstacles, and more robustness testing.

References

[1] S. S. Wegener, S. M. Schoenung, J. Totah et al., "UAV autonomous operations for airborne science missions," in *Proceedings of the 3rd AIAA Unmanned Unlimited Technical Conference*, pp. 302–311, Illinois, Ill, USA, September 2004.

[2] S. Tsach, A. Peled, D. Penn, B. Keshales, and R. Guedj, "Development trends for next generation UAV systems," in *Proceedings of the AIAA Infotech at Aerospace Conference and Exhibit*, pp. 490–503, Israel Aircraft Industries, May 2007.

[3] "Air Vehicles Vision 2009," Air Vehicles Directorate, Air Force Research Laboratory, 2009.

[4] "California Wildfires at a Glance," *Associated Press*, 2007, http://www.boston.com/news/nation/articles/2007/10/27/california_wildfires_at_a_glance/.

[5] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robots*, The MIT Press, 2000.

[6] A. Tsourdos, B. White, and M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles*, John Wiley & Sons, West Sussex, UK, 2011.

[7] S. Kambhampati and L. S. Davis, "Multiresolution path planning for mobile robots," *IEEE Journal of Robotics and Automation*, vol. 2, no. 3, pp. 135–145, 1986.

[8] J. C. Latombe, *Robot Motion Planning*, Springer, New York, NY, USA, 1991.

[9] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[10] P. Khosla and R. Volpe, "Superquadratic artificial potentials for obstacle avoidance and approach," in *Proceedings of IEEE Conference on Robotics and Automation (ICRA '88)*, April 1988.

[11] B. Krogh, "A generalized potential field approach to obstacle avoidance control," SME–RI Technical Paper MS84-484, 1984, Robotics Research Conference.

[12] J. S. Zelek, "Dynamic path planning," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Intelligent Systems for the 21st Century*, vol. 2, pp. 1285–1290, McGill University, October 1995.

[13] Q. Jia and X. Wang, "Path planning for mobile robots based on a modified potential model," in *Proceedings of IEEE International Conference on Mechatronics and Automation (ICMA '09)*, pp. 4946–4951, Changchun, China, August 2009.

[14] T. D. Dong, X. H. Liao, R. Zhang, Z. Sun, and Y. D. Song, "Path tracking and obstacle avoidance of UAVs—fuzzy logic approach," in *Proceedings of the 14th IEEE International Conference on Fuzzy Systems (FUZZ '05)*, pp. 43–48, North Carolina A&T State University, May 2005.

[15] D. K. Pratihar, K. Deb, and A. Ghosh, "A genetic-fuzzy approach for mobile robot navigation among moving obstacles," *International Journal of Approximate Reasoning*, vol. 20, no. 2, pp. 145–172, 1999.

[16] V. M. Peri, *Fuzzy logic controller for an autonomous mobile robot [M.S. thesis]*, Jawaharlal Nehru Technological University, Hyderabad, India, 2002.

[17] L. Astudillo, O. Castillo, P. Melin, A. Alanis, J. Soria, and L. T. Aguilar, "Intelligent control of an autonomous mobile robot using type-2 fuzzy logic," *Engineering Letters*, vol. 13, no. 2, pp. 93–97, 2006.

[18] X. Yang, M. Moallem, and R. V. Patel, "A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 35, no. 6, pp. 1214–1224, 2005.

[19] M. M. El-Khatib and D. J. Hamilton, "A layered fuzzy controller for nonholonomic car-like robot motion planning," in *Proceedings of the 3rd IEEE International Conference on Mechatronics (ICM '06)*, pp. 194–198, July 2006.

[20] Y. Z. Chang, R. P. Huang, and Y. P. Chang, "A simple fuzzy motion planning strategy for autonomous mobile robots," in *Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON '07)*, pp. 477–482, Taipei City, Taiwan, November 2007.

[21] J. Zhang and J. Raczkowsky, "Robust subgoal planning and motion execution for robots in fuzzy environments," in *Proceeding of IEEE/RSJ/GI International Conference on Intelligent Robots and Systems (IROS '94)*, vol. 1, pp. 447–453, September 1994.

[22] W. Wei, J. B. Mbede, and Y. Zhang, "Neuro-fuzzy motion control for mobile robot," in *Proceedings of the International Joint Conference on eural Networks (IJCNN '02)*, vol. 1, pp. 507–512, Honolulu, Hawaii, USA, May 2002.

[23] M. A. K. Jaradat, M. H. Garibeh, and E. A. Feilat, "Dynamic motion planning for autonomous mobile robot using fuzzy potential field," in *Proceedings of the 6th International Symposium on Mechatronics and its Applications (ISMA '09)*, pp. 1–6, March 2009.

[24] N. B. Hui, V. Mahendar, and D. K. Pratihar, "Time-optimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approaches," *Fuzzy Sets and Systems*, vol. 157, no. 16, pp. 2171–2204, 2006.

[25] S. R. Amada, P. R. Vundavilli, and D. K. Pratihar, "Adaptive vs. conventional potential field approaches for solving navigation problems of a real car-like wheeled robot," *International Journal of Intelligent Defence Support Systems*, vol. 2, no. 4, pp. 290–318, 2010.

[26] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[27] A. Mujumdar and R. Padhi, "Evolving philosophies on autonomous obstacle/collision avoidance of unmanned aerial vehicles," *Journal of Aerospace Computing, Information and Communication*, vol. 8, no. 2, pp. 17–41, 2011.

[28] S. H. Kim and R. Bhattacharya, "Multi-layer approach for motion planning in obstacle rich environments," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit (GNC '07)*, pp. 2656–2666, South Carolina, SC, USA, August 2007.

[29] S. H. Kim and R. Bhattacharya, "Motion planning in obstacle rich environments," *Journal of Aerospace Computing, Information and Communication*, vol. 6, no. 7, pp. 433–450, 2009.

[30] Q. Gong, L. R. Lewis, and I. M. Ross, "Pseudospectral motion planning for autonomous vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 3, pp. 1039–1045, 2009.

[31] K. P. Bollino, L. R. Lewis, P. Sekhavat, and I. M. Ross, "Pseudospectral optimal control: a clear road for autonomous intelligent path planning," in *Proceedings of the AIAA InfoTech at Aerospace Conference and Exhibit*, pp. 1228–1241, California, Calif, USA, May 2007.

[32] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.

[33] J. Choi, R. Curry, and G. H. Elkaim, "Real-time obstacle-avoiding path planning for mobile robots," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, August 2010.

[34] L. E. Buzogany, M. Pachter, and J. J. D'Azzo, "Automated control of aircraft in formation flight," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, vol. 3, pp. 1349–1370, 1993.

[35] W. Proud, M. Pachter, and J. J. D'Azzo, "Close formation flight control," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 1231–1246, Portland, Ore, USA, August 1999.

[36] T. Ross, *Fuzzy Logic With Engineering Applications*, John Wiley & Sons, West Sussex, UK, 2010.