An Introduction and Literature Review of Fuzzy Logic Applications for Robot Motion Planning

Mr. Paul Yanik[1], Dr. George Ford[2], Dr. William McDaniel[3]

Abstract - This paper reviews fuzzy logic applications for robot motion planning. A short literature review is provided and three applications of fuzzy logic are considered: navigation and obstacle avoidance of a robotic vehicle in a 2D environment; a multilink industrial robot manipulator; and a robotic vehicle in a 3D environment. Scenarios are compared for their assignment of fuzzy sets, inference (rule base format and size), defuzzification technique, and the effectiveness and limitations of algorithms.

Keywords: fuzzy logic, fuzzy logic applications, robot motion planning, trajectory planning

INTRODUCTION

In the field of mobile robotics, the goal of autonomous robot operation is a topic of considerable research. Robots may frequently find themselves in environments for which a reliable map of obstacles and terrain is unavailable due to the dynamics of the environment, imprecise sensory data or simply a lack of prior knowledge of the environment [4]. In such situations, it may be necessary for a robot to recalculate its trajectory online. While an end-to-end motion plan for the robot in terms of gross motions over longer distances may remain valid, a response to a moving or unforeseen obstacle may force the robot to alter its path amid local obstacles while en route from the initial configuration to the final goal. As a possible means to meet such requirements, fuzzy logic based algorithms have found application in the area of robot motion planning because of their inherent ability to deal with imprecise inputs and their low computational complexity.

Fuzzy algorithms execute in three major stages: fuzzification, inference, and defuzzification. In the fuzzification stage, real world sensory inputs in a given universe of discourse are characterized on the closed interval [0, 1] according to their levels of membership in fuzzy sets. These sets are given names which express qualities of the input variable using easily understood linguistic terms. A membership function maps the value of the input variable to a degree of membership in each of the fuzzy sets. The fuzzified value then, represents the level of truth of each of these linguistic terms for a given input. For example, the angular direction of a near obstacle to a mobile robot might have a universe of discourse of -90° to 90° where 0° denotes the current heading of the robot. Figure 1 below shows a possible definition of fuzzy sets for how a "crisp" (real world) input angle θ might linguistically reflect the level to which the obstacle is left, in front, or right of the vehicle. Here, an angle of 30° represents a direction having fuzzified degrees membership in the set of angles to the right ("rightness") of μ_R and frontness of μ_F. In this manner, fuzzy sets are capable of handling imprecise inputs. Should an input angle be sensed inaccurately, its levels of truth according to the linguistic terms may vary while its relative membership levels in the sets remain qualitatively the same. Sets are often defined to have the piecewise-linear shape shown so as to reduce the computational complexity of determining set membership [3]. Typical shapes include triangular, square, singleton, Gaussian or asymmetric types. Other variables commonly of interest in robotic applications include distance to an obstacle and speed of the robot with respect to an obstacle.

The inference stage applies the fuzzified input value to a rule base to determine a [still fuzzy] command output. The rule base contains the operational intelligence of the system.

[1] Western Carolina University, Belk 161, Cullowhee, NC 28723. pyanik@wcu.edu
[2] Western Carolina University, Belk 161, Cullowhee, NC 28723. gford@wcu.edu
[3] Western Carolina University, Belk 161, Cullowhee, NC 28723. wmcdaniel@wcu.edu

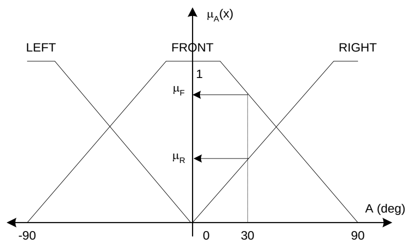

Figure 1 - Example fuzzy set definition with a fuzzified input

An example rule which might apply to a mobile robot is given below.

If ObstacleAngle is RIGHT and Distance is NEAR Then
SteeringDirection is LEFT_BIG (1)

A rule base must cover all permutations of input variables having degrees of truth in all possible linguistic terms. Hence the total number of rules *N* which must be represented in a rule base either by explicit statement or default action is given by in the relation:

$$N = \prod_{i=1}^{m} p_i \qquad (2)$$

where *m* is the number of input variables (angle, speed, distance, etc.), and p_i is the number of linguistic terms for the i^{th} variable [5]. Multiple rules in the rule base may have their predicate conditions satisfied to greater or lesser degrees by fuzzified input variables. Such rules are said to have *fired*. In fact, where fuzzy sets are defined to overlap on their universe of discourse, at least two rules are guaranteed to fire for any input value. Each fired rule, then, possesses an *adaptability* to the associated output command through the fuzzy operation (AND, OR, sum, bounded sum, product, etc.) stated by the rule [2]. Commonly, the AND (min) operation is used as shown in the example above. In such a case, the minimum fuzzified value of the predicate conditions becomes the adaptability of that rule to its consequence.

The defuzzification stage extracts a crisp command output from inferences drawn from fired rules. Techniques for defuzzification generally involve some analysis of the regions created by cutting the output fuzzy sets using adaptabilities from each fired rule. An example of such a region is given by Figure 2. Common methods for this operation include taking the centroid of largest area (CLA), and mean of maximum value (MOM). Numerous other approaches exist which are not considered here.

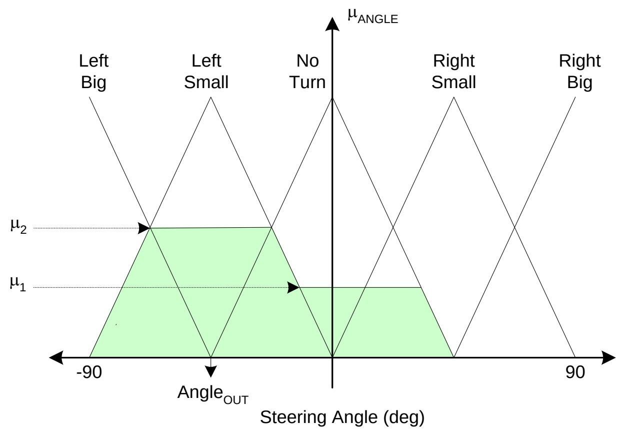

Figure 2 – Example of defuzzification (by MOM method).

This paper discusses three selected applications of fuzzy logic based algorithms to robot motion control and compares/contrasts their approaches. These applications include robotic vehicle navigation in a 2D environment, multilink robot manipulator guidance, and aerial robotic vehicle navigation in a 3D environment. Section 0 includes a concise summary of the articles which present each of these applications. Section 0 includes analysis of the approaches for their relative strengths and weaknesses.

LITERATURE REVIEW OF FUZZY LOGIC APPLICATIONS

Three examples of recent research are reviewed wherein methods are discussed for solutions to issues associated with robots autonomous robot navigation in the presence of incomplete or changing knowledge of their environments. Each of these examples employs a fuzzy logic based algorithm which allows the robot to make online decisions about its local trajectory without recalculation of its end-to-end motion plan.

Mobile Robot Navigation in a 2D Environment

Yang, Maollem and Patel [3] propose an augmentation to previous applications of fuzzy logic to 2D robot motion planning. All figures, equations and algorithm details in this section are attributed to this work unless otherwise stated. Prior work in this area focused on short range reactive control. That is, robots were navigated by simply reacting to near obstacles upon detection while taking into account a global goal direction. While such algorithms have frequently proven effective, they often encounter situations in which the goal configuration becomes unreachable by the robot despite the availability of a traversable path. More commonly, reactive fuzzy navigation may suffer from “shortsighted” behavior wherein the angle to the final goal influences all steering decisions in unison with local sensor data. Situations then arise in which short range sensors may not detect obstacles between the current configuration of the robot and the goal. In these cases, a path may be selected that is less desirable than others available. Figure 3 compares the results of shortsighted behavior to an end-to-end path plan. Through purely reactive short-range fuzzy control, the robot attempts to move in the direction of the goal at point B when obstacle 2 is still out of its perceptive range. The undesirable path ABCDEG is the result. Clearly, with the benefit of long-range planning, path AJKG would be seen as more desirable.

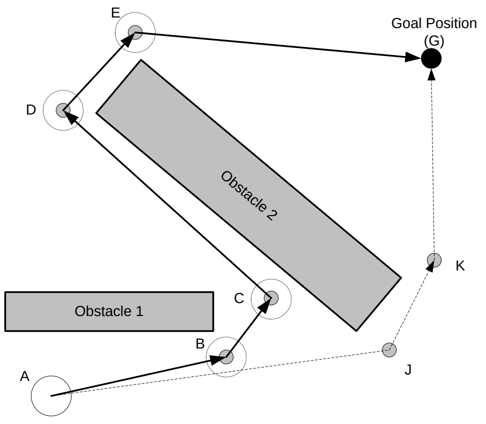

Figure 3 – Shortsighted goal-seeking versus long-range path planning.

The authors formulate an approach which attempts to overcome this tendency of fuzzy algorithms with a “layered, goal-oriented” navigation strategy. Two layers are proposed: long-range versus short-range information assessment. The first layer uses long-range sensor data and the global goal angle to determine a direction that is both traversable (free of obstacles) and desirable (toward the goal). The qualities of directional traversability and desirability are represented as fuzzy sets and fused to produce a way-point along the path to the goal. Sensors are positioned at intervals around the perimeter the robot body and detect distant obstacles. The signal strength of each sensor indicates the relative nearness of an obstacle. This strength is fuzzified through a collection of trapezoidal fuzzy sets for angles of -180° (left) to +180° (right). Where adjacent sensors detect an obstacle with strengths of μ_1 and μ_2, an untraversable area τ_i is the composed fuzzy set found by the bounded sum of the two strengths as given by the equation below.

$$\tau_i = \mu_1 \oplus \mu_2 = \min\{1, \mu_1 + \mu_2\} \qquad (3)$$

The *traversable* area Γ is given by the complements of all τ_i as defined by the equation below.

$$\Gamma = not \bigvee_{i=1}^{n} \{\tau_i\} = 1 - \max_{i=1}^{n}\{\tau_i\} \qquad (4)$$

where *n* is the number of activated sensors [3]. The desirability Ω of a potential steering direction is a fuzzified representation of the goal angle ϕ with respect to the current heading of the robot. It is determined by a composed set from the relative strengths (μ_1 and μ_2) of adjacent triangular fuzzy sets at 90° intervals on the same universe of discourse as traversability. It is defined by the equation below.

$$\Omega = \mu_1 \vee \mu_2 = sum\{\mu_1, \mu_2\} \qquad (5)$$

The direction to the way point is the fuzzy set $\tilde{\gamma}$ represented by the intersection of composed sets for traversability and desirability.

$$\tilde{\gamma} = \Gamma \wedge \Omega = \min\{\Gamma, \Omega\} \qquad (6)$$

Over the range of possible angles, the intersection set will have multiple peaks. Each peak will be separated from others by an interval of zero membership since the sets for untraversability have crossing points greater than 0.5. This intersection of composed sets is defuzzified by taking the mean of maximum of the largest area (MOMLA) to generate a crisp angle γ which is the direction to the way-point. This defuzzification technique is a combination of the mean of maximum (MOM) and centroid of largest area (CLA) techniques. It is shown to offer an improved outcome when finding the crisp target angle over the CLA method often used elsewhere. The position of the way-point and its orientation angle can then be found by

$$x_w = x_i + \rho\cos(\theta_i - \gamma) \qquad (7)$$

$$y_w = y_i + \rho\sin(\theta_i - \gamma) \qquad (8)$$

$$\theta_w = \theta_i - \gamma \qquad (9)$$

where (x_i, y_i, θ_i) is the initial configuration of the robot, (x_w, y_w, θ_w) is the configuration of the way-point, and ρ is the distance from the robot's current position to the way-point and

$$\rho = L + \delta \qquad \rho \le R \qquad (10)$$

where *L* is the distance between the robot and the obstacle closest to the way-point, δ is the size of the robot and *R* is the effective radius of the sensors.

The second layer takes the way-point produced by the first layer as a sub goal. This layer produces a local trajectory which guides the robot toward the way-point while avoiding collisions with obstacles. A ring of short-range sensors is used to implement a reactive navigation algorithm wherein local direction and speed commands are generated in response to near obstacles as the robot seeks the way-point. In a manner analogous to the first layer algorithm, the way-point is used as a desired direction for the robot while sensor inputs are used to infer a direction that allows the robot to avoid obstacles. The simultaneous objectives of sub goal seeking and obstacle avoidance are combined to produce a safe heading for the robot. The rule base for the second layer is of the general form shown below.

If sensor S is fired, Then
the direction indicated by S becomes fully disallowed. (11)

Here, a fired sensor denotes one in which the sensor reading for a near obstacle is above some threshold. By fully disallowing the direction associated with a fired sensor, the sub goal is pursued as aggressively as possible. The sub goal need not be attained precisely, however, since it is not the global objective. Further, the sub goal may become unreachable by dynamics of the environment or the robot may become lost or stuck while seeking the sub goal. In such situations, the second layer may abandon the sub goal and pursue a new one. This judgment can be made by assigning a time limit *T* for the robot to reach the way-point. The time limit can be found as

$$T = \frac{\rho}{v_b} \qquad (12)$$

where v_b is the typical speed of the robot in an unobstructed environment. The second layer of the algorithm can accept a new way-point from the first layer whenever the time limit expires.

A final feature of the algorithm is the implementation of a deadlock handling mechanism. Should the robot enter a situation in which obstacles block a direct path to the final goal and such obstacles are too large for long-range sensors to plan a path around, the fuzzy algorithm presented thus far could result in oscillatory motion that does not result in progress toward the goal. In such situations, the robot needs a strategy breaking the cycle of unproductive decisions. Typically, this is handled with a wall (or contour) following behavior until a safe trajectory is discovered. Instead, the authors temporarily replace the global goal angle with a new target angle. This allows the robot to make reasonable path alterations based solely on traversability when encountering convex obstacles (even though it may move in a direction away from the global goal) in order to extricate itself from a deadlock scenario.

The authors implement their algorithm experimentally on a Koala (6-wheeled) robot equipped a ring of sixteen infrared sensors each having a range of 5-20 cm and a field of perception of 10°. The infrared sensors are used as both long- and short-range detectors by setting their thresholds accordingly. Experimental results are divided into those for static versus dynamic environments. In a static environment, the algorithm determines only reachable way-points. Graphs of sensor readings show that the readings are generally low except when a way-point is close to the vertex of an obstacle. This behavior is interpreted as the algorithm's ability to avoid obstacles before coming close to them. Further, this behavior represents avoidance of the shortsighted behavior problem associated with a purely reactive fuzzy controller.
The algorithm's behavior in a dynamic environment involves movement of obstacles during the robot's passage so as to render some way-points unreachable. Here the robot is abandons such way-points upon expiration of its deadline to pursue new way-points until the final goal is attained.
Finally, in order to demonstrate the effectiveness of the layered algorithm over earlier methods, the Koala robot is programmed (separately) with only reactive direction-based and speed-based fuzzy algorithms. Compared to these algorithms, the layered approach produces improvements in navigation time of 27.6% and 16.3% respectively. Further, it is seen that the layered approach produces a smoother trajectory and avoids obstacles before coming in close proximity to them. This behavior allows wheel velocities to remain roughly constant with respect to one another resulting in less directional oscillation.

Multi-joint Industrial Robot Manipulator

Zavlangas and Tzafestas [4] present an application of fuzzy logic to the problem of autonomous operation for a jointed industrial manipulator. In particular, a fuzzy algorithm is proposed and validated for obstacle avoidance with a three-DOF RRP robot. All figures, equations and algorithm details in this section are attributed to this work unless otherwise stated. The proposed algorithm would allow the robot to operate without prior knowledge of its dynamic environment. The approach is based on sensory detection of only the manipulator's nearest obstacle. As in [3], the motivation for this research is to overcome the computational complexity of classical path planning, so that a manipulator may react to dynamic constraints online. Additionally, this approach is expected to provide benefit in industrial situations where path planning is conducted by tedious manual motion specifications. The authors hope to allow the user to move task specification for the robot to a higher level of abstraction: what is to be done rather than how, precisely, to do it.
A basic framework is presented in which an interrelationship between fuzzy control units for links of a SCARA (RRP) industrial manipulator is used for control. Each link is controlled by a separate fuzzy unit where an array of sensors present input to the controller. Essential inputs for a given link l_j, $j=1 \ldots n$ where n is the number of links, consist of a distance to its nearest obstacle d_j and an angular difference between its current and goal configurations. Sensory data for each link represents a scan over a cylindrical envelope surrounding the link.
The goal of the fuzzy controllers is to provide motor commands τ_j which allow all links to make progress toward their goal configuration (an attractive force) while reversing course when the current velocity would have them collide with the nearest obstacles (a repelling force). Since the robot is an RRP manipulator, the first two links only need consider such obstacles that lie with their plane of movement: to their left ($d_{j,\,left}$) or right ($d_{j,\,right}$). The final prismatic link must consider these plus any obstacle which lies below it ($d_{j,\,below}$). Because the motion of a proximal (closer to the global origin for the manipulator) link effects the motion of more distal links, the sensory information of more distal links must be considered by each link more proximal than itself. This relationship is shown in the simplified system diagram of Figure 4.

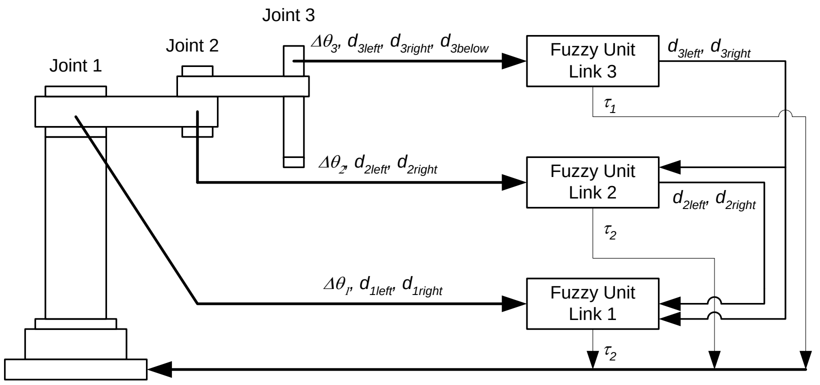
Joint 3
Joint 2
Joint 1
$\Delta\theta_3$, d_{3left}, d_{3right}, d_{3below}
Fuzzy Unit Link 3
d_{3left}, d_{3right}
τ_1
$\Delta\theta_2$, d_{2left}, d_{2right}
Fuzzy Unit Link 2
d_{2left}, d_{2right}
τ_2
$\Delta\theta_1$, d_{1left}, d_{1right}
Fuzzy Unit Link 1
τ_2

Figure 4 – Block diagram of fuzzy control system for an RRP manipulator.

The proposed fuzzification algorithm maps the distance to the nearest obstacle to its membership in a collection of fuzzy sets with linguistic terms denoting closeness to an obstacle (Far left, Left, Close left, Close right, Right, Far right) on a universe of discourse for proximity. Since these sets reflect closeness to an obstacle, they also represent a repelling force which the controller must attempt to minimize. The controller employs a second collection of fuzzy sets representing the attractive force which is the angle of the link with respect to its goal configuration. These sets, similarly, represent linguistic terms denoting direction toward the goal (Far left, Left, Contact, Right, Far right). All fuzzy sets are trapezoidal.

Again, the feedback from distal links to more proximal links, makes use of the distances and fuzzified membership values associated with the more distal links. Each fuzzy controller, then, makes use of both repelling and attractive fuzzy sets, their associated linguistic terms and levels of membership in each. So, if the linguistic terms for repulsion are $A_{pj}^{(j)}$ and those for attraction are $B_{pj}^{(j)}$, a generalized element in the inference rule base can be stated as

$$R_{r_j}^{(j)} : IF\ d_j\ is\ A_{p_j}^{(j)}\ AND \ldots d_n\ is\ A_{p_n}^{(j)}\ AND\ \Delta\theta_j\ is\ B_{q_j}^{(j)}\ THEN\ \tau_{r_j} \qquad (13)$$

where r_j is the number of rules for the fuzzy controller associated with link l_j. The value τ_{rj} is the aggregated fuzzy output for the controller. Mamdani's direct method [2] is used for defuzzification. The level of membership in the fuzzy sets for output is the intersection (*AND*) of linguistic term sets indicated by the rule. Such intersection can be calculated by either taking the minimum of all fuzzy outputs or the product of all fuzzy outputs. The authors use the product operator so as to allow all inputs to influence the final output. The crisp output τ_j of the controller is given by the centroid of the output fuzzy set area as determined by the rule base.

The algorithm is implemented using Microsoft C and tested with a simulated Adept1 industrial manipulator (RRP). Three obstacle arrangements are generated for both static and dynamic environments. Each link in turn was individually impeded by an obstacle in the scenarios. In each case, the algorithm provides collision-free motion. It is acknowledged that while the algorithm is largely effective, it is theoretically subject to becoming trapped at local minima due to the reactive nature of the fuzzy algorithm in the absence of global path information [4].

Robotic Vehicle in a 3D Environment

The research discussed above deals with mobile robots acting in planar environments. Various scenarios also exist in which autonomous robots must navigate in three dimensions as in aerial or underwater environments. Shi, Collins, and Dunlap [1] consider the application of an unmanned helicopter flying in a "cluttered" 3D environment. All figures, equations and algorithm details in this section are attributed to this work unless otherwise stated. Here the approach to 3D navigation extends concepts previously reviewed by decomposing the 3D problem into several 2D problems. The innovation claimed consists of the fusion of 2D outcomes to form a 3D fuzzy solution volume. The centroid of the largest component of this volume is defuzzified to yield the final output. Fuzzy controls are associated with preferred flying direction (yaw and pitch angles) as well as speed. Here again, fuzzy logic provides a viable solution to the 3D navigation problem by virtue of its tolerance to imprecise inputs and low computational complexity, both of which are valuable in high speed, dynamic environments.

The authors propose a fuzzy control algorithm based on "preference-based" behavior control. In this paradigm, steering command options receive fuzzy preference parameters for a set of behaviors: Goal seeking, obstacle avoidance, left wall tracking, and right wall tracking. Each behavior yields a fuzzy preference to a given command option: Large left turn (LLT), small left turn (SLT), no turn (NT), small right turn (SRT), and large right turn (LRT). Each command option, then, receives a vector of preferences from which a weighted "voting" process produces a composite desired steering correction angle. Additionally, a method is proposed for reducing the size of the rule base, which would otherwise increase dramatically with the addition of a third dimension to the typical 2D approach.

The practical range of directional change for the robotic helicopter application was constrained to be -52.5° to 52.5° for pitch and -90° to 90° for yaw. Decomposing the 3D navigation problem to a set of 2D navigation subproblems is accomplished by making each individual 2D fuzzy controller responsible for considering sensor data in the vertical angular range for pitch. This range Θ is based on a minimum safe distance d_s to the nearest obstacle and the height of the helicopter H_h as expressed by the equation below.

$$\Theta = 2\arctan\frac{H_h}{2d_s} \tag{14}$$

Hence, Θ is 15° where H_h is 1.8 m and d_s is 6.5 m. This result yields seven regions over the constrained range of pitch angles. A tradeoff is made here. Increasing a region's angular range to reduce computational complexity or adding a tolerance to the dimensions of the helicopter can potentially obscure passable spaces. Conversely, decreasing the angular range or overlapping the ranges to expose passable spaces increases the number of 2D subproblems and the associated computational cost. The seven vertical regions correspond to seven planar subproblems that can be solved by an individual fuzzy controller as shown in Figure 5.

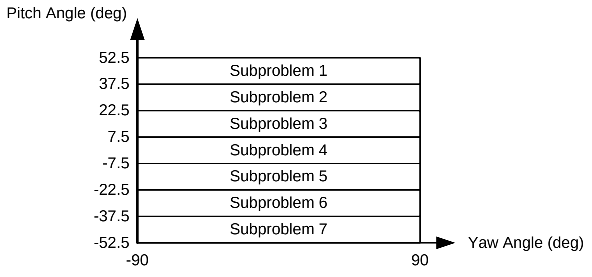

Figure 5 – Planar decomposition of a 3D navigation problem.

Horizontally, obstacles are detected within the constrained angular range for yaw (-90° to 90°). Determination of how narrow the angular range for which a specific sensor is responsible is determined in the same manner as for Θ (pitch) above. Nine horizontal sensors with overlapping cones of perception are used. The ranges for vertical sensing and horizontal sensing are also overlayed to produce a 7x9 array of perception regions. Four sensor readings are taken for each. The minimum of these readings indicates the nearest obstacle in each region as shown in Figure 6.

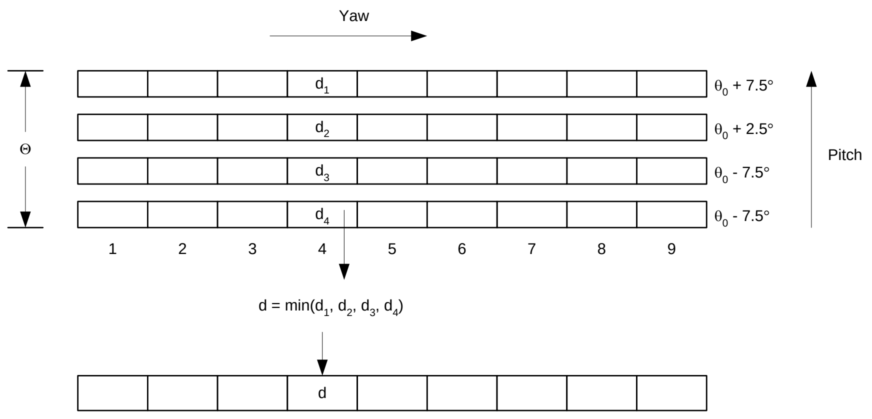

Figure 6 – Nine horizontal sensors per angular pitch range. The minimum of four readings in each region is used.

Readings were converted by the fuzzy controllers for each of the seven 2D subproblems to degrees preference for the five possible flight commands. From this conversion, a set of a preference vectors, $\alpha^{(k)}$ are produced which include preferences for each flight command from each of the seven 2D subproblems. In order to account for vertical preferences, a Goal Pitch Orientation (GPO) was used to produce a fuzzy weighting factor γ_k for each preference vector. Each vector is adjusted by this factor according to its compliance with the GPO as shown in assignment below.

In earlier 3D methodologies, command fusion was conducted for the preference vector by conventional defuzzification of each vector followed by vector summation to give a crisp output steering angle. This however, may produce problems in situations where the summed vectors generate a final output that essentially resolves safe paths that are near one another to a third path between the two that is not safe. Here an innovation is proposed in which the controllers for each of the subproblems are not allowed to execute their own defuzzification. Instead, the fuzzy preference vectors are juxtaposed to represent a 3D solution space that is defuzzified to determine a final steering orientation. Horizontal dimensions of the space are pitch angle and yaw [command] angle. The vertical dimension is the degree of preference. The result is a 3D vertical bar graph where tall bars are regions of free and desirable space as shown in Figure 7.

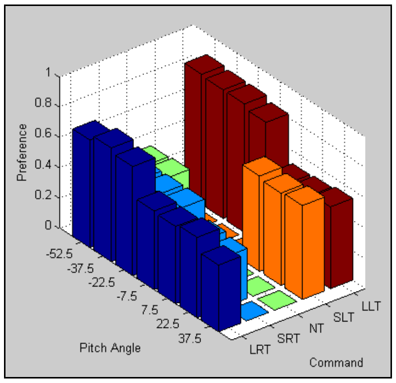

Figure 7 – Example 3D solution space for command preference defuzzification.

Defuzzification consists of computing the centroid of the largest contiguous volume (or contour) in the solution space. This calculation encounters problems when a contour is concave. The centroid of such a volume could lie close to the edge or even outside the contour. A threshold is applied to exclude regions of relatively low preference. The remaining candidate volumes are then decomposed into component convex volumes. The convex volume having the maximum summation of preferences is selected and its centroid is calculated to produce the final crisp steering command.
The algorithm is tested in a simulated environment using MATLAB Virtual Reality Toolbox. The subject vehicle (a helicopter) measured 4.0 m x 1.8 m. Start and goal configurations were specified by Cartesian coordinates involving varying horizontal translations and altitude changes. Various scenarios are described involving virtual buildings in an urban landscape, a floating object meant to emulate another craft (e.g. a blimp) and a virtual forest. The algorithm was exercised demonstrate various characteristics including a preference for a horizontal path (when ascending/descending paths are available), ascending, descending, wall tracking, and the ability to move through tight passages in cluttered spaces. Each of the simulations showed that the algorithm was effective in finding a safe path of reasonably efficient overall distance. Further, Shi, et al [1] tested the level of complexity in the algorithm by implementing it with a larger number of planar subproblems. When thirteen such decompositions (as opposed to the previous seven) were implemented so as to have greater overlap among sensory regions, the improvements (reductions) in total distance traveled by the robot were reduced minimally: from 0% to 1.99% across the range of scenarios. For the case in which no change in distance between scenarios was measured, the simulation showed that the robot actually chose different paths. The benefit seen by using a greater number of decompositions is correlated to the vertical complexity in the environment (buildings are simple vertical protrusions while trees vary in shape from top to bottom). However, the increase in decomposition (7 subproblems to 13 subproblems) approximately doubled the computing requirements.
The algorithm as it is implemented also neglects dynamics inherent to the vehicle. For example, the vehicle may not be capable of executing certain acceleration commands. This is especially important in situations where obstacle

collision is impending. A possible solution might involve removal of certain volumes from the 3D defuzzification contour which represent such hazards [1].

DISCUSSION

Similarities can be seen across the research summarized above in both the type of problems which can be solved using fuzzy logic based algorithms as well as for the manner in which the fuzzy technique is applied. Although nominal differences exist between these applications, the algorithms presented were effectively decomposed by the works reviewed to similar permutations of 2D obstacle avoidance and goal seeking. Hence, it can be stated generally that these fuzzy algorithms can be used to implement efficient control in a variety of settings with little conceptual difference in approach. The limitations of fuzzy algorithms exposed by these articles are, perhaps the more interesting results of the comparison.

In [3], the goal was to overcome the inherent susceptibility of reactive fuzzy algorithms to shortsighted behavior and deadlock by using two layers of fuzzy control. Fuzzy sets here are defined conventionally to be trapezoidal and triangular in shape. Unique among the algorithms studied is the rule base and fuzzy operator used here (bounded sum) to generate adaptability of rules for traversability and desired direction to long-range sub goal heading corrections (way-points). In other algorithms and in background reading, lambda cuts (see Figure 2) to output fuzzy sets by the rule adaptabilities appear to be most common. In this case, the strength of a sensor's detection of an obstacle normalized to the interval [0, 1] represents an adaptability of a rule for that sensor. A composed fuzzy set created by the bounded sum and its intersection with other composed sets (including that for the goal angle) influences the final defuzzified output. The defuzzification technique, mean of maximum of largest area (MOMLA) is a hybridization of the common techniques of mean of maximum and center of largest area (MOMCOLA). It is shown to provide a defuzzified output that was more accurate than either of the conventional techniques used alone. Apart from these lower level considerations, the layering of controllers to implement both short-range and long-range goal seeking represents a novel solution to the common tendency of reactive fuzzy control to become lost or stuck in pursuit of a goal. Here the use of long-range sensing allows the fuzzy algorithm to emulate a roadmap style method without the computational complexity of the analogous global planner. Additionally, the use of the deadline mechanism to prevent deadlocked motion is a notable difference of this algorithm and helps to overcome the stuck robot (local minima) problem. For these reasons, the layered approach seems to be highly practical one. Experimentation used actual hardware which lends added credibility to results.

Zavlangas and Tzafestas [4] implement an economical obstacle avoidance system for an industrial RRP robot. The formation of fuzzy sets and rules are conventional. Sets are configured conventionally as trapezoidal and are narrowed to react aggressively to the closest obstacles/goals. Rules are also conventional with an exhaustive combination of all variables and linguistic terms employed. Although not an especially complex adaptation to consider, the rules here take into account relationships between links. This feature presents a level of detail which vehicle navigation algorithms are not generally required to address.

The primary quality of the algorithm implemented is that only the single nearest obstacle is taken into account when planning the robot's local trajectory. The key benefit is in reduced computational complexity. However, an undesirable side effect of this reduction is an increased tendency to become trapped at local minima. Further, the algorithm may fail to find an obstacle-free path to the goal even if one exists. While the algorithm presents the common improvement to global path planning that reaction time and computing requirements are reduced in cluttered and dynamic environments, the potential for failure to find a solution path seems to be a significant flaw. For future work, the authors propose (without detail) a layered approach with higher level planning functions. Presumably, the layered algorithm presented above may provide a template.

In [1], the 3D helicopter navigation controller utilizes two features that are not shared by other algorithms studied: preference-based control, and 3D defuzzification. Unlike the two algorithms previously discussed which used input variables to generate a single output command, the preference-based system considers the preference of behaviors to multiple commands. Here, preferences perform similarly to rule adaptabilities to the command options. In this way, the preferences serve as votes for the final output behavior.

Set and rule formation are typical of other applications. Defuzzification, however, represents a primary innovation of this algorithm. Partitioning of a solution volume is conducted and the center of largest volume taken for final defuzzification. While it is unclear how this approach avoids local minima given that it is essentially a purely reactive system, presumably the combination of wall tracking and path selection based on altitude offer a greater range of possibility. Hence, the probability of the helicopter becoming trapped is small.

In general, it can be seen that the basic fuzzy algorithm is adaptable to various applications. Most useful research appears to be built upon conventional styles of set and rule base formulation and defuzzification t o achieve higher levels of abstraction. Coupled with the strengths of fuzzy approaches in simplicity and ease of implementation are the challenges of overcoming local minima. Applications which address these challenges would seem to hold the greatest promise for real world implementation.

CONCLUSION

The use of fuzzy logic for local navigation of robots is a much discussed topic in literature. Various implementations have been shown to be effective and efficient. The ability of fuzzy techniques to deal with imprecise data allows for smooth trajectory execution while their low computational complexity allows them to react quickly to dynamic environments without the need to alter the robot's end-to-end path.

Because algorithms based on fuzzy logic depend on sensory data to make navigational corrections, they are essentially reactive in nature. This quality can elicit suboptimal behaviors including shortsightedness and the local minima problem. A knowledge of longer range obstacles as from long-range sensory instruments or conventional model-based planning algorithms could allow the reactive behavior of a fuzzy controller to make quick trajectory adjustments while still approximating the most preferred path.

Despite the limitations and non-deterministic nature of fuzzy algorithms, they have been shown to produce robust local navigation control for robots in unknown and noisy environments. The literature reviewed here shows that, with modest differential complexity, fuzzy algorithms can be used practically in both 2D and 3D environments as well as with industrial manipulators. As such, they represent a complement to rather than a replacement for conventional motion planners across a wide range of applications where real time, autonomous obstacle avoidance is needed.

References

[1] D. Shi, E.G. Collins, and D. Dunlap, "Robot Navigation in Cluttered 3D Environments Using Preference-Based Fuzzy Behaviors," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 37, no. 6, pp. 1486-1499, Dec. 2007.

[2] K. Tanaka, *An Introduction to Fuzzy Logic for Practical Applications*, New York, NY, Springer-Verlag, 1997.

[3] X. Yang, M. Maollem and R.V. Patel, "A Layered Goal-Oriented Planning Strategy for Mobile Robot Navigation," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 35, no. 6, pp. 1214-1224, Dec. 2005.

[4] P.G. Zavlangas and S.G. Tzafestas, "Industrial Robot Navigation and Obstacle Avoidance Employing Fuzzy Logic," *Journal of Intelligent and Robotic Systems*, vol. 27, pp. 85-97, 2000.

[5] *LabVIEW PID Controller Toolkit User Manual*, National Instruments Corporation, Austin, TX, 2006.

Paul Yanik
Mr. Yanik is a visiting assistant professor and the program director of the program director of the Electrical and Computer Engineering Technology program at Western Carolina University. Mr. Yanik's professional interests include computer networking, electronics design automation, and digital design verification.

George Ford, Ed.D.

Dr. Ford is an assistant professor in the Department of Construction Management at Western Carolina University. Dr. Ford worked for over fifteen years in the corporate world in plant engineering and environmental engineering positions and managed numerous construction projects as a plant engineer in the paper, plastics and rubber industries including warehouses, manufacturing buildings and utilities infrastructures.

William McDaniel, Ed.D.
Dr. McDaniel is an assistant professor and the Distance Learning Program Coordinator at Western Carolina University. Dr. McDaniel's professional interests include Engineering Graphics & 3D Modeling & Animation, Engineering Materials, Polymer Science, and Distance Learning capability.