Line-of-Sight Tracking Based Upon Modern Heuristics Approach

Kevin McCarty
University of Idaho
Idaho Falls, Idaho USA
kmccarty@melaleuca.com

Milos Manic
University of Idaho
Idaho Falls, Idaho USA
misko@ieee.org

Abstract— **Any autonomous vehicle must be able to successfully navigate a wide variety of situations and terrain conditions. As a result, proposed solutions usually involve a sophisticated and expensive implementation of both hardware and software. In many situations, however, truly autonomous operation may not be necessary or practical. Instead, equipping and training a vehicle to automatically follow a human-controlled lead vehicle is a viable alternative. While still autonomous, the vehicle relies upon its leader to handle the complex decisions with regards to course and speed. This paper presents a simple and elegant configuration, called FLoST for Fuzzy Line of Sight Tracking, based on inexpensive line-of-sight devices controlled by a heuristic to determine direction and speed of a follower. Unlike the alternative approach where the follower needs to undergo a complex training process, the follower using the approach presented in this paper primarily relies upon a human leader to provide direction, allowing for a much simpler and less expensive vehicle implementation while still being able to match or exceed the effectiveness of the autonomous design under similar conditions. Finally, three boundary cases of lead vehicle maneuvers (circle, spiral and weave) are presented to show the efficacy of this approach.**

I. INTRODUCTION

There are situations where it is necessary or desirable to be able to rely upon autonomous, machine-guided vehicles to perform certain tasks. On the surface of Mars, for example, it is not practical to send a human driver while the crater of an active volcano may be deemed too hazardous. In less extreme conditions, autonomous vehicles can be called upon to perform a variety of tasks. Unfortunately both the difficulty and cost often prove to significant obstacles to implementation [1]. Innovative solutions have addressed this [2] but challenges in the form of urban terrain, road conditions, traffic "rules" and other obstacles continue to plague autonomous vehicles. Situations do exist, however, where the environment is tightly controlled enough not to require a fully autonomous solution but rather one which combines human leadership with the ability of a vehicle to follow.

For example, it may be necessary for an individual to require the help of a machine to transport material from point to point. An example could be an airport, where a person has to move luggage across a busy terminal, or a factory, where inventory is moved from production to shipping through small corridors or a junkyard where tons of metal has to guided around piles of debris. More extreme cases might involve the removal of hazardous waste or movement through a dangerous area where it would be preferable to use automated vehicles in lieu of human resources. Whereas it might otherwise be prohibitively expensive or simply too dangerous to trust to a truly automated vehicle, a hybrid system, with a human leader and an array of mechanical followers could prove a practical alternative.

"Following" technology, as opposed to a purely autonomous one, doesn't require sophisticated decisions with respect to direction, speed, hazards, or road conditions and as such requires less sophisticated sensory hardware and software. Additional reliance upon the judgment of the human leader can also mitigate the impact of obstacles and other issues which can make the operation of a purely autonomous vehicle difficult and hazardous.

Other solutions for automated following have been proposed, for example, by combining CCD cameras and neural networks for pattern recognition [3], motion sensors, GPS systems and standard communications [4] for platooning. This paper demonstrates that combining line-of-sight devices and a fuzzy algorithm for following is superior to the first solution in that it avoids much of difficulties associated with noise in the patterns and superior to the latter solution in that it employs a simpler array of devices and logic. An autonomous follower, using FLoST to mimic its human leader, presents a much more cost-effective solution and a potentially more effective one.

This paper presents a novel, fuzzy arithmetic based algorithmic approach to the problem of autonomous following. Using the analogy of a mother "Duck" and her "ducklings", the algorithm guides a series of mobile, autonomous units to follow a lead vehicle (or "Duck") and each other from a predetermined distance, mimicking both velocity and, to a greater or lesser degree depending upon conditions, direction traveled.

The proposed alternative technique relies upon a series of rapid angular scans to achieve location and distance measurements to the lead target. Technologies for line-of-sight tracking have been in use in both the commercial and military sectors for many years in various devices [5], [6]. These devices detect an object (such as a hostile aircraft) and relay information such as distance, direction and speed to other units. Such devices, mounted upon and directing the movement of some sort of mobile platform, following a human or mechanical leader, can thereby creating some new utility.

This paper is organized as follows: Section II presents a simple scenario and a series of applications for the algorithm presented in this paper. Section III presents the FLoST (Fuzzy Line of SighT) algorithm. Section IV lists test scenarios along with a discussion of the FLoST algorithm performance. Section V will present conclusions and future work.

978-1-4244-1718-6/08/$25.00 ©2008 IEEE

II. PROBLEM STATEMENT

For the purpose of demonstration, consider a lead vehicle ("Duck") and *N* followers ("ducklings"). Based on the FLoST algorithm presented in this paper, each "duckling" becomes the "Duck" to the subsequent "duckling", hence any configuration of 1 "Duck" to *N* "ducklings" is possible.

This technique has application in many areas where the movement of material is impractical for human agents. Consider the following scenario: an earthquake damages a chemical plant. Highly explosive chemicals must be moved to another facility immediately but it is deemed unsafe for anyone to get too near them while in transit. Applying FLoST in this scenario, the chemicals are loaded onto a series of FLoST-equipped vehicles following a lead vehicle with a human driver. The lead vehicle, or "Duck", is a heavily armored vehicle able to protect its human driver from the effects of a blast. The followers, or "duckling" are FLoST transports.

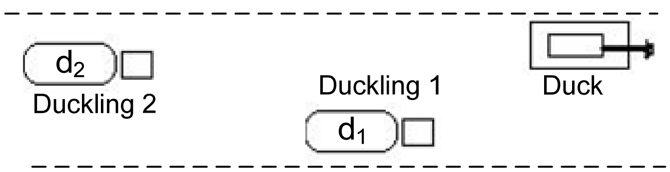

Fig. 1. 2-D surface, example with 1-Duck and 2 ducklings.

Typically the "Duck" will proceed in a determined, but not constant direction, and will not have to back-track at any point. The road surface may contain obstacles to move around, but otherwise allow the "ducklings" to maintain line-of-sight to the "Duck". "Ducklings" themselves are "daisy-chained" such that the "duckling" in front will serve as its follower's respective "Duck". It is reasonable to assume that the "Duck" will not intentionally try to evade the "ducklings" so its movement will be fairly consistent, though it may be necessary, at times, for more drastic maneuvers.

III. FLOST (FUZZY LINE-OF-SIGHT TRACKING) ALGORITHM

The FLoST algorithm will be presented on a generic problem of *N* "ducklings" following a "Duck", as illustrated by Fig 2:

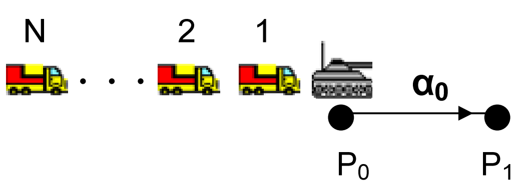

Fig. 2 Duck and ducklings at start

The heuristic of FLoST algorithm allows each "duckling" to follow the "Duck" as it proceeds from point to point on its journey. Each "duckling" accomplishes this by maintaining a knowledge base of "Duck" behavior. At each point, the "duckling" records the relative distance ΔP and relative direction change $\Delta\alpha$ of the "Duck" in its knowledge base, allowing it to determine its absolute position and direction as well as its desired velocity.

The "duckling" then applied the FLoST heuristic to better predict future behavior of the "Duck's" human driver as a function of its past behavior. In this example, the "duckling" creates three speed "zones" called Slow (*S*), Normal (*N*) and Fast (*F*) which are overlapping measures of the maximum speed of the "Duck" and used to assign direction changes, $\Delta\alpha$, along with an average direction change α_{Avg}, shown in Fig. 3.

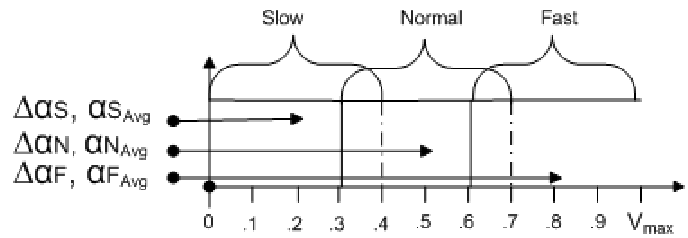

Fig. 3. Direction changes and averages in Duck speed zones

The "duckling" then assigns each $\Delta\alpha$ to its respective zone (or zones if it lies within an overlap):

$$\Delta\alpha_i \in \begin{cases} S, & V_i \le .4V_{max} \\ N, & .3V_{max} \le V_i \le .7V_{max} \\ F, & V_i \ge .6V_{max} \end{cases} \qquad (1)$$

where V_{max} is the maximum velocity of the "Duck". By taking the mean over all $\Delta\alpha$ in a given zone, the duckling calculates the α_{Avg}, which is then used to generate a turn rate coefficient for that zone. In this example the coefficients are:

$$TR_x = \tfrac{1}{2}\alpha_{xAvg} + \tfrac{1}{2}\alpha_{Last} \qquad \text{where } x \in \{S, N, F\} \qquad (2)$$

Where α_{Avg} is the average α for a given zone; α_{Last} is the last measurement taken for that zone; TR_x is a component used in a fuzzified function to calculate the search.

At the beginning of this scenario, the "Duck" starts at location P_0, followed by two "ducklings". The "Duck" will then proceed over time Δt in a direction and speed indicated by the angle α_0 to the point P_1 as shown in Fig 4. The first "duckling" will orient itself on and proceed to P_0, then using the FLoST algorithm, begin its first scan for the "Duck"using as its first search angle, Θ, the vehicle Maximum Turn Rate (MTR) based upon the fuzzy equation developed by Wu, Zeng, Chaing and Lee [7] for a given vehicle.

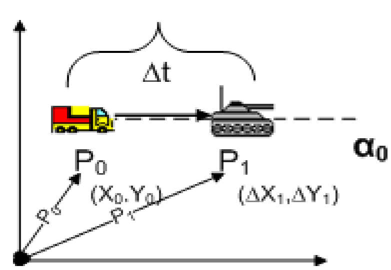

Fig. 4. Duck proceeds to first point

As it moves from point to point, the "duckling" updates its knowledge base of "Duck" behavior with information derived from each new "Duck" point. From the original speed zones, the duckling creates fuzzified versions of Slow (S), Normal (N), and Fast (F) that range from 0 to the maximum velocity of the Duck (V_{max}) as shown in Fig. 5.

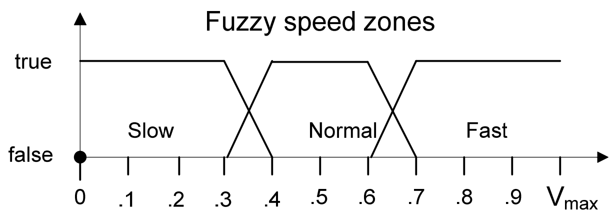

Fig. 5. Fuzzified speed of Duck.

The duckling applies the FLoST algorithm using the following steps:

Step 1. Scan for and locate "Duck". Apply FLoST to determine the search angle Θ, calculated as follows:

$$\Theta = Vf * TR = [Vs \quad Vn \quad Vf] * \begin{bmatrix} TR_s \\ TR_n \\ TR_f \end{bmatrix} \qquad (3)$$

where Θ has a minimum value of 1 degree. In the absence of any points for any TR_x, use the vehicle Maximum Turn Rate (MTR).

The search angle is drawn using the +/- Θ offset from the current angle α_i Search continues until either the "Duck" is located or it is determined the "Duck" is lost as illustrated for points by Fig 6.

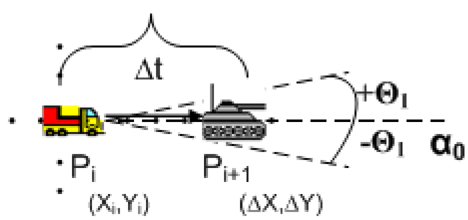

Fig. 6 Scan for Duck at P_1

Step 2. Use information about relative Duck's position to determine next point: P_1, new direction α_1.

The "duckling's" rangefinder provides a relative distance from the "duckling", ΔP, and the traversal mechanism provides the relative direction Δα. The new location P_1 and the new vector α_1 are defined as:

$$P_{i+1} = P_i + \Delta P_i \qquad (4)$$
$$\alpha_{i+1} = \alpha_0 + \Delta\alpha \qquad (5)$$

Step 3. Calculate the velocity of "Duck" to P_1 as:

$$V_i = |P_i - P_{i-1}| / \Delta t_i, \qquad \Delta t_i = t_i - t_{i-1} \qquad (6)$$

Step 4. Update the knowledge base as described by eqs. (1) & (2).

Step 5. Adjust course and speed and proceed to "Duck's" new known location.

As opposed to just applying the Maximum Turn Rate, MTR, the FLoST TR_x will usually generate a smaller search area than MTR. In this way, the "duckling" (d) can concentrate its search in the area the "Duck" (D) appears to be headed as illustrated in Fig 7.

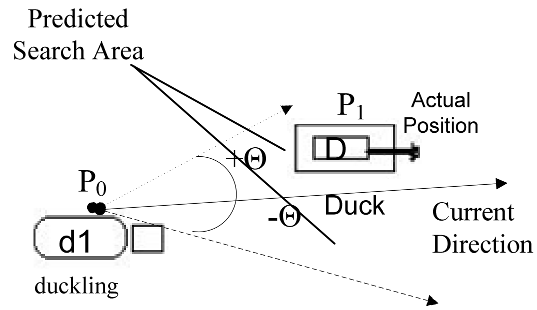

Fig. 7. Choosing a search area.

The major hardware components of the "duckling" consist primarily of an array of Line-of-Sight (LoS) sensors mounted on elements that can traverse the search area. A sample configuration is illustrated in Fig 8. These LoS mechanisms feed speed, direction and distance information to the navigation system which adjusts the "duckling"'s movement accordingly.

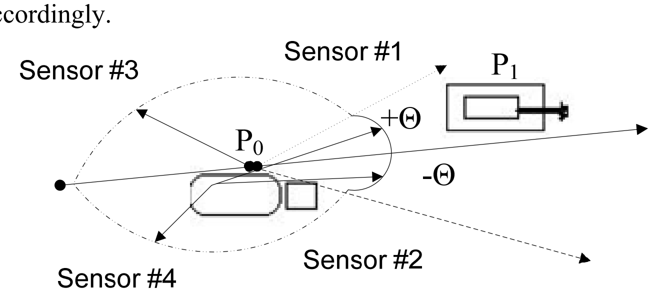

Fig. 8 Duckling searching with sensor array

Using the FLoST algorithm, the "duckling" follows its "Duck" from point to point, adjusting course and speed and updating its knowledge base of prior actions as shown in Fig. 6. The algorithm attempts to minimize the search area whenever possible, adjusting its search angles to account for variations in "Duck" movement at each iteration. Minimizing search angles allows for a more rapid scan rate which both reduces the search area and enables the "duckling" to make course and speed corrections on a more frequent basis - reducing the chance for future misdirection.

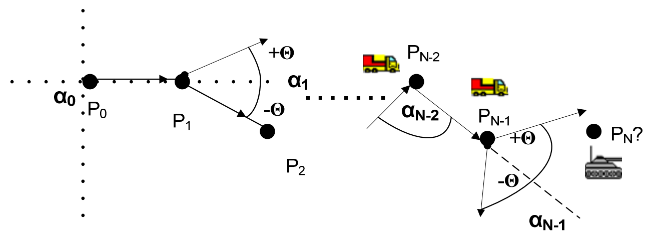

Fig. 9. Two ducklings using FLoST to follow Duck

This is possible when the actions of the "Duck" are generally consistent and relieves the "duckling" of having to take into account considerations regarding terrain, speed, safety, obstacles and other factors that would be very difficult and expensive to implement in an automated system. However, such factors are much more easily "inferred" by training the duckling to follow the behavior of the "Duck" precisely. Possible differences in human behavior at various speeds (and times) are addressed using the speed zone TRs with fuzzification so transitions are loosely defined.

IV. TEST EXAMPLES

Among the examples are both trivial cases and non-trivial cases. Trivial cases arise in the following three instances: 1) when velocity is zero; 2) when the change of direction is zero; 3) and when the sample time is zero. In each case, the search angle collapses. As trivial, these examples will not be discussed.

Non-trivial cases arise when the "Duck" moves in one direction for a period of time, establishing a knowledge base of very small direction changes, then executes a maximum turn in one direction or the other. For the "ducklings" this behavior is unexpected and will require additional iterations in order to learn and adapt to this new pattern. The performance of FLoST will be discussed in the following three boundary cases, representing three scenarios of sudden or unusual trajectory of "Duck" movements. These cases are: 1) the "Duck" is going in tight circle; 2) the "Duck" going in spiral; 3) the "Duck" weaving back and forth.

The first boundary case is where the "Duck" moves in a tight circle using the maximum turn rate (MTR) as shown in Fig 10.

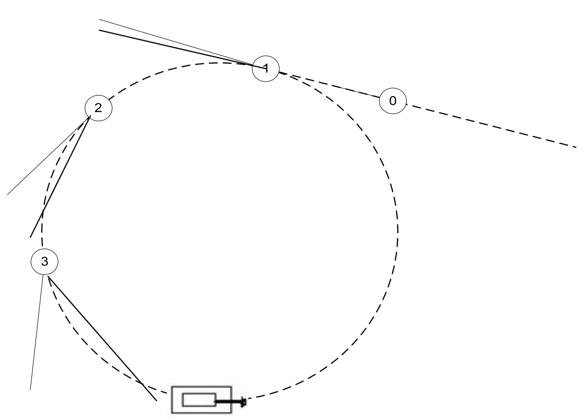

Fig. 10. Duck moves in tight circle

As the duckling has "learned" only small movements to this point, it will have to learn very different, previous behavior in order to generate a reasonable search rate coefficient TR.

Whereas a traditional search uses the full MTR to establish the search angle, the coefficients used by the "duckling" create a tight angle initially, growing larger with each sampling. However, while the standard search will always capture the "Duck" in its primary scan, the "duckling" will require many samples in order to generate a large enough search angle. A way to compare the two methods is to examine how efficient each is in its primary scan. This can be accomplished by comparing the scan Θ for each:

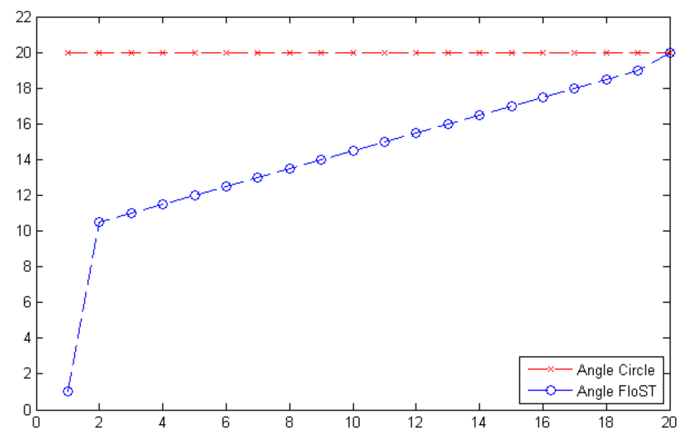

Fig. 11. Comparison of FLoST vs Maximum Turn Rate Search

For an TR = 20° the FLoST algorithm will begin at a significant disadvantage as it tries to unlearn previous behavior with the first measurement being off by a factor of 20. Very quickly, however, the FLoST adjusts the angle of search based upon the last recorded change in direction enabling it to rapidly approximate the increased difference between the directions of "duck" and "duckling" (Fig. 11). Problems of major directional change are not limited to FLoST; extreme maneuvers cause difficulty in other fuzzy tracking algorithms as well [11], [14].

The second boundary case is when the "Duck" moves in a widening spiral.

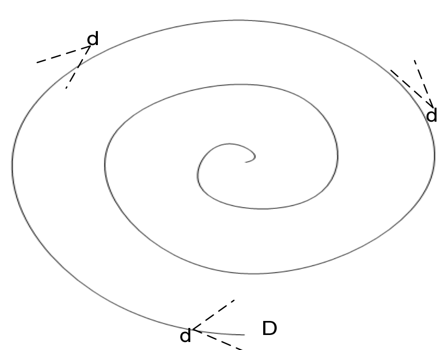

Fig. 12. Ducklings follow Duck in spiral.

The spiral motion starts as extreme as the circle but gradually reduces the turn rate. In the following example, the spiral starts out with an initial turn of 20° and loses a degree every two samples. It does not take long for the FLoST algorithm to catch up in this case as indicated by a graph comparing the search areas:

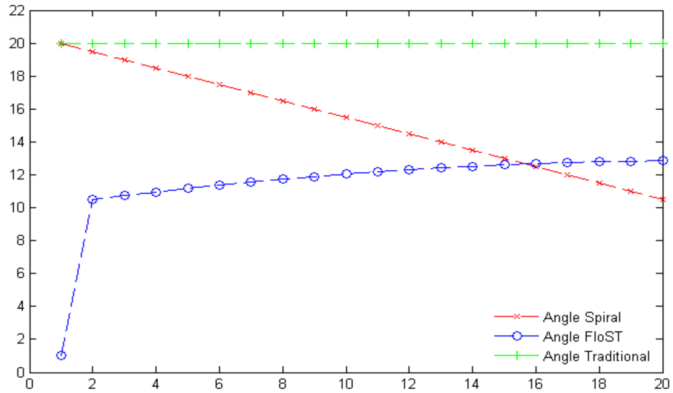

Fig. 13. Comparison of FLoST vs Maximum Turn Rate Search in a spiral

As with the tight circle, the "duckling" starts off at a significant disadvantage, however, the FLoST search angle quickly "catches up", surpassing the efficiency of a brute-force MTR calculation after a series of iterations allows it to expand and adapt, then narrow its search to accommodate the slowly degrading turn of the spiral.

A third boundary case is the weave. Like the circle, the weave utilizes maximum turn rate MTR, but in alternating directions.

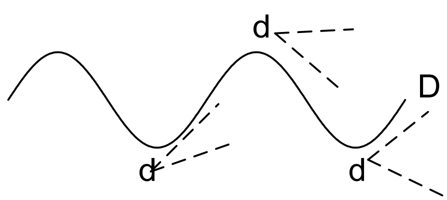

Fig. 14. Ducklings follow Duck in weave pattern

The weave attempts to perform the same extreme maneuver as the circle, although inertia in one direction will hinder its ability to exploit the full MTR in the other direction.

With an initial turn of 20° and subsequent weaves of 19°, FLoST quickly adapts to create a useful Θ as indicated by the graph in Fig. 15.

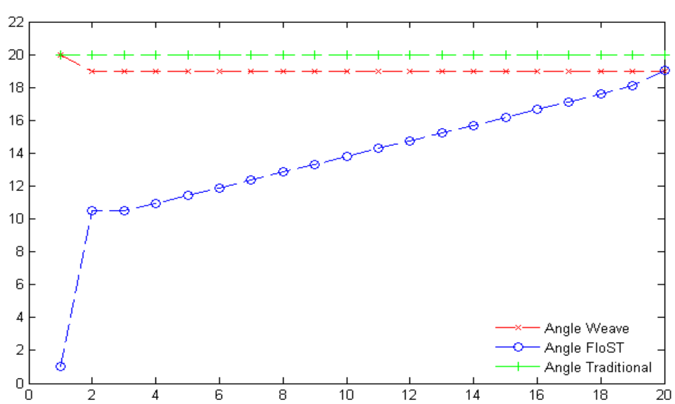

Fig. 15. Comparison of FLoST vs Maximum Turn Rate Search in a weave

Despite some initial trouble at boundary extremes, FLoST does very well under more less extreme, more "normal" operating conditions.

It is reasonable to assume a human operator will not deliberately attempt to evade a "duckling" or drive in an extreme circle, weave or spiral for any significant length of time. "Normal" operation then consists of relatively gentle turns at higher speeds or extreme turns at low speeds followed by sequences of relatively straight paths, with the FLoST "duckling" expanding or narrowing its search angle to accommodate, relative to the fuzzy zones (slow, normal, and fast) determined during operation. Using the fuzzy zones further optimizes the "normal" case since larger degree turns are more safely accomplished at slower speeds, reflected in the autonomous "duckling's" knowledge base. The "duckling" then knows that the slower speed has a greater incidence of wide turns and will adjust its search parameters accordingly.

In the original scenario of the chemicals that need to be move to a safe location, workers could load the dangerous chemicals and be well away from harm while the "Duck" calls the "ducklings" to marshall. The "Duck" driver can navigate a complex path at a safe speed knowing the automated "ducklings" will mimic the course and speed very precisely. Once to an area where they can be unloaded, the "ducklings" are dispersed automatically to safe areas and processed as needed.

V. Conclusion

Notions of an unmanned successful tracking system usually involve a complex array of devices and software and the assumption that no human direction is available. However, as this paper demonstrates, a simpler combination of human intelligence and machine algorithm could prove a worthwhile alternative, even, for all intents and purposes (via remote control), completely duplicating an autonomous process. The performance of FLoST, the algorithm presented in this paper, is discussed for three boundary cases ("Duck" moving in circle, moving in spiral and weaving back and forth) in which the FLoST algorithm quickly and successfully adjusted the search parameters to compensate.

Further work needs to be done to improve FLoST prediction and accuracy by incorporating a neural network based approach. Extending the FLoST to allow "ducklings" to properly couple and decouple from a train, sort themselves out and avoid conflicts with other "ducklings" would greatly improve the applicability and overall usefulness of FLoST. Finally there need to be processes to allow the "Duck" and "duckling" to respond and reestablish contact in the event line-of-sight tracking fails.

References

[1] D.K. Liu, X. Wu, A.K. Kulatunga, G. Dissanayake; *Motion Coordination of Multiple Autonomous Vehicles in Dynamic and Strictly Constrained Environments*; IEEE Conference on Cybernetics and Intelligent Systems, June 2006

[2] R.W. Wall, J. Bennett; G. Eis; *Creating a Low-Cost Autonomous Vehicle*; IEEE 2002 IECON02, Nov. 2002

[3] Y. Omura, S. Funabiki, T. Tanaka; *A Monocular Vision-Based Position Sensor Using Neural Networks for Automated Vehicle Following*; IEEE 1999 International Conference on Power Electronics and Drive Systems, PEDS'99, July 1999.

[4] K.S. Chang, W. Li, P. Devlin, A. Shaikhbahai, P. Varaiya, J.K. Hedrick, D. McMahon, V. Narendran, D. Swaroop, J. Olds; *Experimentation with a vehicle platoon control system*; Vehicle Navigation and Information Systems Conference, 1991, Oct. 1991

[5] C.J. Leigh-Lancaster, B. Shirinzadeh, Y.L. Koh; *Development of a laser tracking system*; Fourth Annual Conference on Mechatronics and Machine Vision in Practice, 1997.

[6] F. Michaud, P. Lepage, P. Frenette, D. Létourneau, N. Gaubert; *Coordinated Maneuvering of Automated Vehicles in Platoons*; IEEE Transactions On Intelligent Transportation Systems, Vol. 7, No. 4, December 2006.

[7] S. Wu, G. Zeng, H. Chiang, T. Lee; *Application of Optimal Fuzzy Tracking Control for Automotive Driving*, 2005 IEEE International Conference on Man and Cybernetics Systems Oct. 2005 Vol. 4.

[8] M. Hanss, M.; *Applied Fuzzy Arithmetic*, New York: Springer-Verlag, 2005.

[9] S. Kato, S. Tsugawa, K. Tokuda, T. Matsui, H. Fujii; *Vehicle Control Algorithms for Cooperative Driving With Automated Vehicles and Intervehicle Communications*; IEEE Transactions on Intelligent Transportation Systems, VOL. 3, NO. 3, September 2002.

[10] S. Wu, C. Lin; *Continuous time-invariant optimal fuzzy tracker design based on local concept approach*; The Ninth IEEE International Conference on Fuzzy Systems; Volume 1, May 2000.

[11] K.C. Chan, V. Lee, H. Leung; *Robust TargetTtracking Using a Fuzzy Filter*; IEEE International Conference on Systems, Man and Cybernetics, 1995, Volume 5, Oct. 1995

[12] S. Wu, C. Lin; *Global Optimal Fuzzy Tracker Design Based on Local Concept Approach*; IEEE Transactions on Fuzzy Systems, Vol. 10, No. 2, April 2002.

[13] R. Cucchiara. M. Piccardi; A. Prati, N. Scarabottolo; *Real-time Detection of Moving Vehicles;* 1999 International Conference on Image Analysis and Processing, Sept. 1999.

[14] C. C. Cheng; *Fuzzy tracking method with a switching grey prediction for mobile robot*; 10th IEEE International Conference on Fuzzy Systems; Dec. 2001.

[15] H Chia-Hung Hsu, A. Liu; *A Flexible Architecture for Navigation Control of a Mobile Robot*; IEEE International Conference on Systems, Man and Cybernetics; May 2007.

[16] D. Crump, D.L. Livingston; *Robot navigation using sensors with fuzzy characteristics*; IEEE SoutheastCon, 2007. Proceedings, March 2007