Synchronized Multi-Point Attack by Autonomous Reactive Vehicles with Simple Local Communication

CHIN A. LUA[1], KARL ALTENBURG[2], KENDALL E. NYGARD[1]
[1]Department of Computer Science and Operations Research
[2]Department of Accounting and Information Systems
North Dakota State University
Fargo, ND 58105
USA
Kendall.nygard@ndsu.nodak.edu, http://www.cs.ndsu.nodak.edu/~nygard

Abstract: We present a model consisting of a swarm of unmanned, autonomous flying munitions to conduct a synchronized multi-point attack on a target. The Unpiloted Air Vehicles (UAVs) lack global communication or extensive battlefield intelligence, instead, relying on passive short-range sensors and simple, inter-agent communication. The multi-point synchronized attack is successfully demonstrated in a simulated battlefield environment. The simulation results indicate that the reactive, synchronized, multi-point attack is effective, robust and scalable. It is especially well suited for numerous, small, inexpensive, and expendable UAVs.

Key-Words: Adaptive Flying Munitions, Autonomous Behaviors, Emergent Intelligence, Synchronized Multi-point Attack, Unpiloted Air Vehicles.

1. INTRODUCTION

> *"Attack him where he is unprepared, appear where you are not expected."*
> *"The spot where we intend to fight must not be made known; for then the enemy will have to prepare against a possible attack at several different points; and his forces being thus distributed in many directions, the numbers we shall have to face at any given point will be proportionately few."*
>
> -- Sun Tzu, The Art of War, 6th century B.C. [20]

Synchronized multi-point attack employs several tactical doctrines against an enemy target in a battlefield: 1) the element of surprise, 2) high probability of target destruction, and 3) diversion of enemy resources. However, it is difficult to attack a target synchronously. Previously developed approaches utilize high-level cooperative control that relies on long-range global communication, a global plan, and substantial *a priori* information about the battlefield, including locations of targets and threats [5], [15], [16].

The model that we present regulates a swarm of unmanned, autonomous, and adaptive flying munitions [1] to conduct a synchronized attack on a target. This model has the following characteristics:

- It functions as a swarm.
- Each UAV has a receiver for simple short-range messages from nearby UAVs, and long-range signals from a target.
- Each UAV has a transmitter to generate short-range messages.
- Each UAV has a small and simple set of common reactive behaviors that enable it to avoid obstacles, search, sense a target, and circle around a target or point, and attack.
- Each UAV has an internal compass and a short-range vision sensor that keep track of the relative distances from and directions to close obstacles.
- There are no designated leaders or followers [18].
- Each UAV makes its own decisions based on the local environment as perceived by its sensors.
- The UAVs are not required to assume any particular formation before a target is detected.
- It is scalable from a 1-point to much larger N-point synchronized attacks.

The UAVs are subjected to the following constraints:

- They must travel at a constant velocity throughout the attack.
- They must stay within a threshold distance of a detected target.
- They can depend on dead reckoning, need not know exactly where they are, and need not have GPS.
- They must strike the target at the same time, without knowledge of the locations of UAVs.
- There is no shared, global clock. They rely on their own timing for the synchronized attack, assuming some uniform, fixed speed.

2. RELATED WORK

Research in the area of flying munitions using autonomous behaviors, motor schema, and force fields as their control mechanism is in its infancy. Gillen and Jacques describe a simulator to evaluate control alternatives for intelligent munitions [10]. Passino et al. [17] explore a reactive biomimicry approach to developing a search map of a battlefield area with UAVs. Work in autonomous, multiple, mobile robotics aspires to similar control design goals: achieving a global behavior in a group of distributed robots using only local sensing, minimal communication, and behavior-based control mechanism. Three examples are given by Fredslund and Mataric [9].

Werger [22] demonstrated a robot soccer-playing team with a minimalist, behavior-based control system. By combining a few basic behaviors, two different group formations of three robots emerged. Mataric [14] showed how a set of simple behaviors, based on local sensing, can be combined so that a global behavior emerges. For example, a global flocking behavior emerges as each robot performs its local behavior of avoidance, aggregation, and dispersion. Kube and Zhang [13] demonstrated that only two basic local behaviors, avoidance and goal seeking, are enough for the physical robots to perform a collaborative box-pushing global behavior. Altenburg, Schlecht and Nygard [1] developed a framework for a simulator that incorporates the behaviors of the flying munitions developed here. This simulated battlefield environment is unique since it employs a swarm of UAVs with limited sensors and local behaviors to achieve the attack, and is arguably more robust than a deliberative approach.

The overall design philosophy of emergent intelligence through local interactions with neighboring individual members is inspired by other natural and synthetic swarms. Some of these other swarms include ants [6], [7], graphical turtles [18], *boids* [19], and fishes [21].

3. BEHAVIOR-BASED CONTROL ARCHITECTURE

The UAVs' control structure utilizes a decentralized behavior-based architecture inspired by the subsumption architecture [8], motor schema [3], [4], and force fields [2], [11], [12].

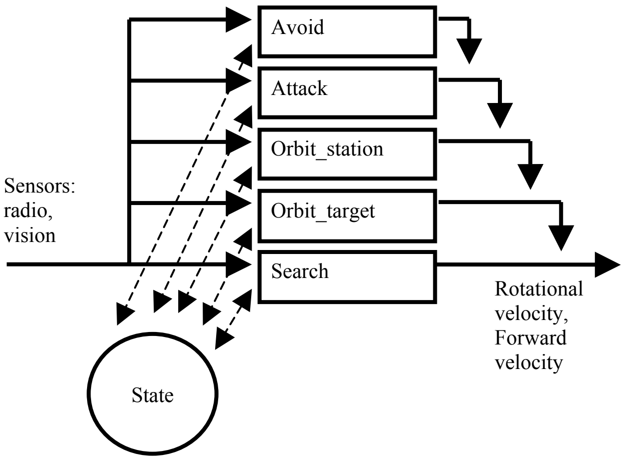


Figure 1. Control structure.

At each time step, the control architecture (Figure 1) reads both the external sensors' state and current internal state to invoke the appropriate behavior by which a UAV's direction and speed are set. However, if there is no information from the current state, the higher behavior layer of the architecture will inhibit and subsume the lower layer(s) as required by the subsumption scheme. The tight coupling between the sensors/state and behaviors is responsible for the UAVs' reactive maneuvers. The architecture is divided into three main subcomponents: sensors, actions, and behaviors.

4. SENSORS

The sensors are the "eyes" and "ears" of the UAV. A UAV's behavior is greatly dependant on its sensors. Nonetheless, the UAV's sensing capacity has limits as depicted in Figure 2.

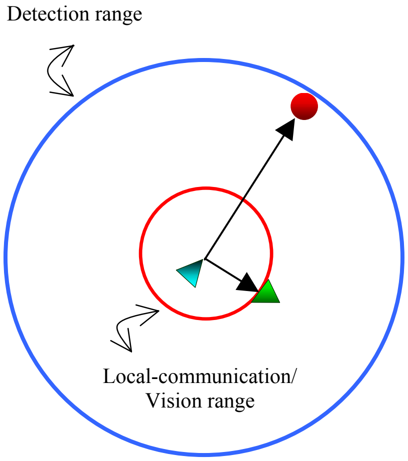


Figure 2. Relative sensory range limitations.

We assume that the target emits a detectable signal that can be sensed by a UAV at a relatively long range. The detection range (the larger circle in Figure 2) is a parameter that is dependant on the effectiveness of the UAV's target sensor. As the UAV detects the target signal, it is capable of determining the relative distance and direction to the target.

The ranges for local-communication and vision, for simplicity, are presumed to be the same (the smaller circle in Figure 2). The vision range is the distance at which the UAV can visually sense obstacles. The simulation assumes the imaging technology is able to determine the relative distances and directions of obstacles so the UAV can take evasive action. The local-communication range is the distance at which the UAV can receive or broadcast short-range messages from or to its neighbors. Normally the detection range is greater than the local-communication or vision range.

5. ACTIONS

The output of the architecture is a vector composed of two motor mechanisms: rotational and forward velocity. Using these two basic actions, the UAV can perform the following three maneuvers: *avoid* (Figure 3), *attract* (Figure 4), and *orbit* (Figure 5).

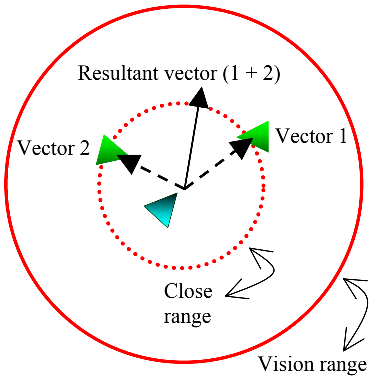


Figure 3. Vector calculation for UAV avoidance.

Avoid is activated when a UAV detects obstacles near itself. In Figure 3, the UAV senses two obstacles in its close range. The UAV acts to avoid the obstacles by traversing along the path of the resultant vector. It will continue to turn clockwise/counterclockwise and move forward until the desired path is satisfied.

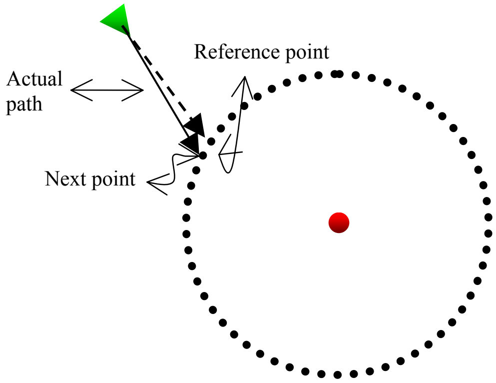


Figure 4. Attraction to a target.

In Figure 4, when a target is detected, the UAV plots a circular path as a set of waypoints about the target. It finds the nearest waypoint relative to itself – called the reference point. The UAV is attracted to the point next to the reference point, counterclockwise. As a result, it will turn and move directly to the next point.

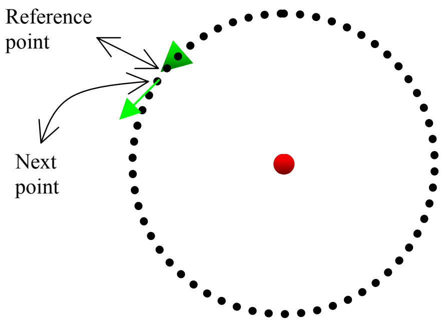


Figure 5. Entering an orbit around a target.

When the UAV touches the "orbit circle," it will orbit around the target as in Figure 5. This behavior is due to the next point always being one point ahead of the reference point, counterclockwise, and thus is the direction of motion. It will move the other way if the next point is placed clockwise. The action is the same for all orbiting UAVs in the simulation despite the radii of the circles.

6. BEHAVIORS

Sensors and actions are implemented in hardware, whereas behaviors are implemented in software. Hence, the behavior algorithms control the attack. To help with understanding the execution of the behaviors, we present an example of a 3-point attack.

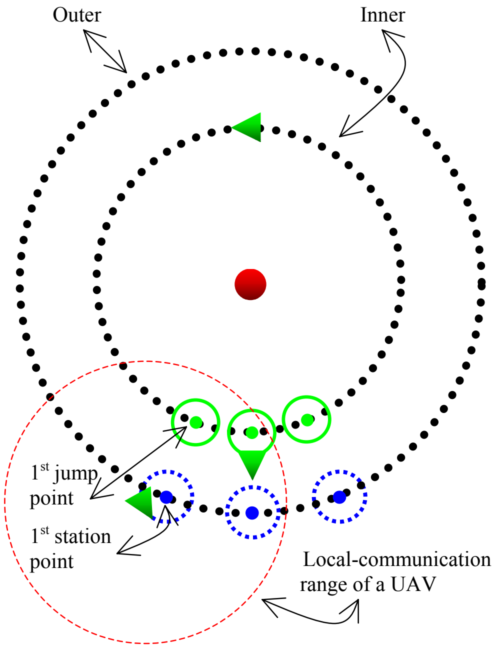


Figure 6. Elements of a 3-point synchronized attack.

The 3-point synchronized attack in Figure 6 has the following elements:

- Three UAVs (triangles).
- A target (large, center dot).
- Two larger dotted circles, outer and inner, each depicted as waypoints with the target as its center. We assume the two circles are a safe distance from the target's detection. They may or may not out of the target's strike range. In any case, if the UAVs are detected, they can immediately strike the target, avoid enemy's munitions, or maximize the strength of radio signals for long-range communication. The estimated radii of both outer and inner are computed by each UAV upon sensing the target signal.
- Three station points (larger dots on the outer circle) are needed for three UAVs to meet, loiter, and communicate. These points must not be far away from each other to honor the local communication constraint.

- Three jump points (larger dots on the inner circle) are needed for triggering the UAVs to "jump" toward the station points.
- Three jump zones (small circular zones with jump points as centers).
- Three station circles (portrayed as waypoints with station points as centers).
- The first jump point is known to all UAVs and determined by a shared, time-dependant formula. For example, assume that the formula is (X * X) Mod 360 where X is an 8-digit month, day, hour number. If the UAVs are deployed on May 1 (0501) at 1 AM (0100), then the 8 digits are 05010100. All UAVs, using the same time formula, know the first jump point is (05010100 * 05010100) Mod 360 = 40 degrees away from the 6 o' clock position on the inner circle, counterclockwise. The formula can be more complex and changed when needed. The first station point is inferred from the first jump point.
- Three messages are used: station number (integer), group number (integer), and strike (boolean). The station number is utilized to 1) take ownership of a station circle, 2) ask the other UAVs to go away, and 3) indicate the current value of the owned station point. For example, the 1st, 2nd and 3rd station points have values 1, 2, and 3. The group number states the highest station number a UAV has "heard" from its neighbors, including itself. A strike is transmitted to signal the attack has begun.
- A UAV who orbits about a station point has a local-communication range that covers its "left" and "right" neighbors, and some section of the inner circle.

The elements above coupled with state diagram below (Figure 7) are utilized to describe each UAV's behaviors occurred in each state.

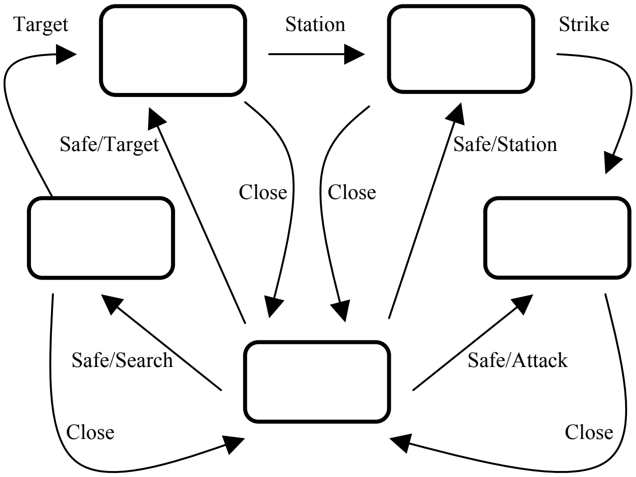


Figure 7. A summary of UAV behaviors.

The behavior in *Search* state:

- The UAVs will keep on searching in a battlefield if a target is not detected.

The behaviors in *Orbit target* state:

- The UAVs detect the target's signal and then move directly to the inner circle.
- They orbit counterclockwise upon touching the inner circle.
- Initially, all UAVs must set their station number to 1 and search for the first jump point. They move directly to the first station circle by entering the first jump zone. For those UAVs that fail to touch the first station circle and *hear* the station number is 1 broadcasted from their close competitors within the local-communication range, they must fall back to the inner circle without delay, increment their station numbers by 1, and repeat the same process for the next successive jump point and zone.
- The UAVs that fail to enter the jump zones are not allowed to jump toward the station circles, and their station numbers will not increment by 1. They have to circle again along the inner circle searching for the original jump zones. It is unlikely that the UAVs will need the second attempt if the jump zones are large enough and the number of neighboring UAVs is small.

The behaviors in *Orbit station* state:

- The UAV will orbit the station circle that it *owns*.
- A station circle can be owned by any UAV if it touches the circle and broadcasts *its* station number *first* before *its* nearby competitors.
- Only the station owner can continuously broadcast all three local messages to other UAVs that are within its local-communication range. Nevertheless, the station owner has to stop its transmission temporary if it is forced out of the circle, but its station number will not increment by 1 since the UAV knows it is the first one to transmit its station number, and other UAVs have already deserted that station circle. It will return to the station circle once the obstacles are gone.
- The group number (constantly updated upwards) and strike signal (not broadcasted initially) will be received and relayed by the neighboring station owners. The two messages will reach all station owners via their overlapping local-communication ranges. Initially, the group numbers may have different values, but they are the same before the strike is called. Any station owner who completes the predetermined number of orbits first will broadcast the strike signal. For example, if the number is 3600 degrees, then the station owner who completes 10 orbits first will order the strike. The station number, however, is not relayed by any neighboring station owners. The value of the station number stays constant since it reflects the current station point owned by the UAV.

The behaviors in *Attack* state:

- When the strike signal is received, each UAV that is still on the station circle will immediately apply the current group number (3 in this case) and its station number to determine its strike position on the inner circle. Each strike position has a fixed number of degrees that separates each UAV along the inner circle. This number is 120 degrees for a 3-point attack. The third UAV with station number 3 will leave its station first and fall back to the inner circle. It will travel 240 degrees around the target to reach its strike position. The second UAV with station number 2 will leave its station after the third UAV has had time to orbit 120 degrees. The first UAV with station number 1 will leave its station if it estimates the second UAV has traveled 120 degrees. All estimates are based on the fact that the inner circle is 8 times larger than the station circle and the UAVs' speed is constant.
- As each UAV arrives at the strike position, it will travel directly to the target and attack.

The behavior in *Avoid* state:

- The UAVs avoid all potential collisions from incoming, close obstacles in all situations. They will safely return to their previous state if the obstacles are gone.

For simplicity, we ignore some details that are not crucial in the performance of this simulation. All of them can be implemented in future work if desired:

- The simulation ignores that the UAVs are at different positions along the station circles when the strike is called. Thus, a timing error will occur in the strike. However, this error can be minimized if the size of the station circle orbits is small. A new local behavior can be added to eliminate the error by considering all UAVs' positions on the station circles in the delayed formula just before the strike.
- Those UAVs who drop out due to mechanical or other failures will not affect the attack significantly if they have not yet owned the stations; if they have, it will be less successful, but some UAVs will carry on the desynchronized attack anyway. This problem can be rectified by asking a UAV to periodically check the station number transmission of its "lesser" neighbor. If the UAV fails to hear its neighbor's station number, it could attempt to take ownership of the abandoned station by leaving its own station. Nonetheless, if the UAV hears its "lesser" neighbor again before taking ownership, it will fall back to the inner circle and search for the next jump zone.
- Other friendly UAVs can disturb those UAVs on the station circles or during the strike. The disturbance will not occur in this simulation if the number of UAVs is equal to or less than the number of stations and all of the UAVs are on the station circles before the strike. In future work, the disruption can be minimized if other UAVs will leave the battle area immediately once all the stations have been taken.
- The UAVs (in the air) and target (on the ground) are represented in the same two-dimensional plane; therefore, the actual slant distance is ignored. If necessary, this problem can be remedied by representing the UAVs in a three-dimensional simulation.

7. BEHAVIORAL DESIGN

The rationale behind the design of the behaviors is as follows:

- The local competition based on shortest relative distance is an effective strategy for determining the local paths and stations to the UAVs. For example, the UAVs compete among themselves to be the first to reach the inner circle, jump zones, station circles, and to issue the strike order if they have completed a predetermined number of station orbits. Hence, there is an emergent intelligence that is comparable to the global strategies despite the fact that the UAVs do not know the whereabouts of each other.
- The broadcasted station number is an effective, fast signal to inform all neighboring UAVs not to waste their time since the station they selected has been taken. Thus, they are able to save fuel for the next station.
- The UAVs can only own a station at a time through which all responsibilities as owners are properly executed.
- The UAVs are not allowed to transmit any messages until they become station owners. This policy ensures that all local messages are safely broadcasted with minimal detection from the target. Senders need not to have replies/confirmations about their messages from receivers, and thus it simplifies the communication.
- The UAVs that miss the jump zones and go for another round ensure that all available UAVs are lining up in all the contiguous stations starting from the first station. If they miss the jump, it is very likely that they have not heard the station number. The UAVs cannot assume others will take their places and responsibilities. For example, if only two UAVs fight for the first jump zone, they may both miss the zone due to their close proximity or because the zone is too small. If the design does not enforce them to check out the first jump zone again, they will both assume the first station has been taken; but in fact, it is empty. One way to avoid this is to ask both UAVs to seek the first jump zone again by orbiting another round. This time, one of them has a higher chance to fill up the first station. The other UAV can go for the next jump zone upon hearing the transmitted station number.
- The jump zone has desirable characteristics. For example, the size of the zone controls the traffic between the jump and station points. If the traffic is congested, the zone can be smaller so that some of the UAVs who miss the zone will have to go for another round. Hence, the diversion of traffic is achieved dynamically.

8. MACHINE LEARNING

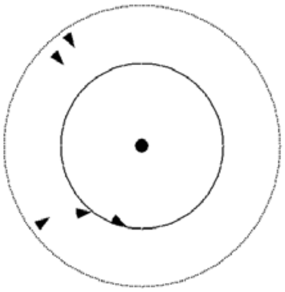
There are some machine learning techniques that are applied to this simulation. For example, a UAV constantly computes the relative distance to a station point. If the distance is equal or shorter than the radius of the station circle of that point, the UAV will broadcast the messages otherwise it will stop transmission. Thus, the UAV learns to transmit messages safely while it is on or inside the station circle.

Another example of learning is the UAV's timing for owning a station. Though a UAV can be close to a station circle, it does not have immediate clearance to broadcast its station number. The UAV needs to check whether it has received the same station number from its neighbors by checking its own. If the UAV's station number is increased by 1, then it knows one of its neighbors has already owned the station and will not broadcast its station number. By inspecting the changed value of its station number, the UAV learns whether it can own a station.

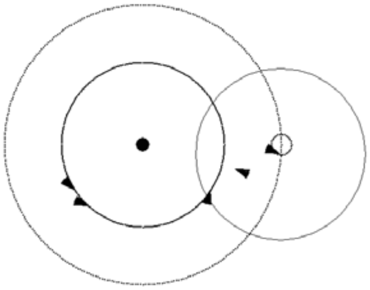
9. SIMULATION RESULTS

The simulation can handle the cases of 1-point to N-point synchronized attacks. We demonstrate how the 5-point and 18-point attacks are carried out in the snapshots that follow.

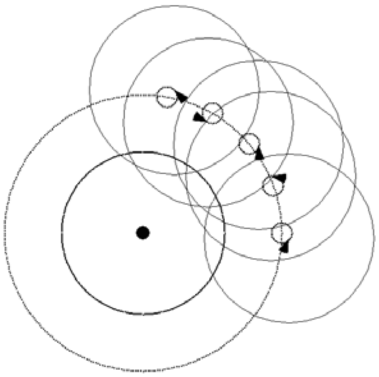
The 5-point synchronized strike is described in the following six scenes:



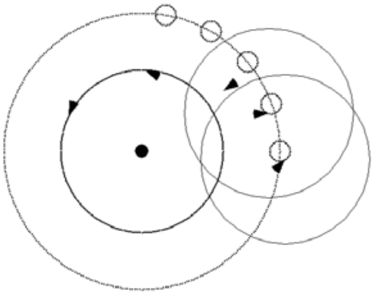
Scene 1. Five UAVs have sensed the target and are rushing to the inner circle around the target. One UAV touches the circle and orbits counterclockwise.



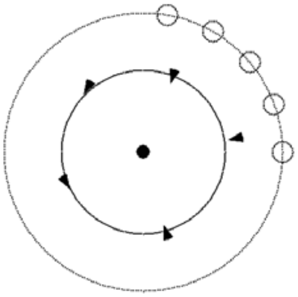
Scene 2. A UAV has touched the first station circle and immediately sent the short-range message (represented by the circle around it) to the two nearby UAVs that it has been owned. The two UAVs abandon the first station instantly and compete for the next jump zone upon receiving the message.



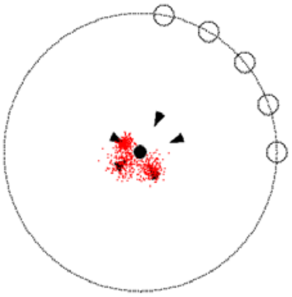
Scene 3. All 5 UAVs are circulating around the station points, broadcasting or relaying messages to their left and right neighbors. As a result, all UAVs (including any two furthest members) will have the updated group number before the strike.



Scene 4. One member has ordered the strike, and two UAVs have taken even positions (360/5 = 72 degrees apart) along the inner circle. All the strike positions have been computed while the UAVs are still station owners. There is no communication after they have left their stations.



Scene 5. All 5 UAVs are in even strike positions now.

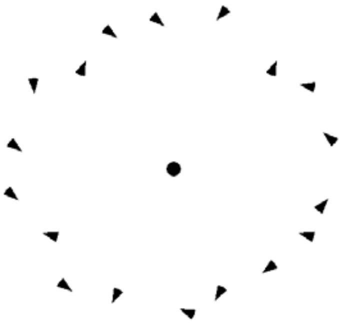


Scene 6. Three UAVs strike the target and explode as two others follow close behind.

The same control mechanism can incorporate 18 UAVs as easily as it handles 5. An 18-point strike is depicted in the following 4 scenes:

Scene 1. A swarm of wandering UAVs approach the target. They all compete to be the first to enter the inner circle while avoiding each other. Some UAVs begin traversing along the circle. Although they seem quite chaotic, they are able to sort themselves out reactively.

Scene 2. Some UAVs have been circulating around the station points (the first station is located at the 6 o'clock position). The rest will move on upon receiving the same station number. Again, the competition quickly facilitates who get the stations despite their chaotic interactions.



Scene 3. All 18 UAVs have filled up the stations.

Scene 4. After the strike positions are reached, the attack begins. Some UAVs explode upon hitting the target, and the mass assault has high probability on target destruction.

Demonstrations of these simulation results are available at: http://www.cs.ndsu.nodak.edu/~lua/ .

10. FUTURE WORK

There are several ways to make this model more robust and versatile:

- The target is static in the model. We would like to apply the multi-point synchronized attack on a moving target.
- The multi-point synchronized attack on several static targets that are in close proximity.
- The multi-point synchronized attack on several moving targets that are in close proximity.
- To handle those UAVs who have owned the stations but drop out due to mechanical or other failures.
- To experiment with noisy or imprecise sensors.
- Additional machine learning on *relative* zone size, jump or station point spacing, UAV's velocity, target velocity, number of UAVs, local-communication/vision range, target detection threshold, degree and strength of avoidance, etc.
- The UAVs' behaviors on direct assault as they are detected by the target.
- The future work mentioned at the end of section 6 above.

11. CONCLUSION

The model demonstrates that a reactive, synchronized, control strategy for multi-point attack can be effective, robust and scalable. In comparison with high-level global control approaches, the model requires modest UAV communication resources, is tolerant of loss of individual UAVs, and generally robust. It is especially well suited for numerous, small, inexpensive, and expendable UAVs. This tactic surprises the enemy target and diverts its resources, and, thus, increases the odds for destroying the target. These attack doctrines were true in Sun Tzu's time, and remain true today.

ACKNOWLEDGMENTS

The authors would like to thank Joseph Schlecht and Benzir Md Ahmed for their advice and helpful discussion on this work at various stages. This research was sponsored by the U. S. Air Force Office of Scientific Research.

REFERENCES

[1] K. Altenburg, J. Schlecht, and K. E. Nygard, "An Agent-based Simulation for Modeling Intelligent Munitions," Proceedings of the WSEAS Conference, Skiathos, Greece, September 2002

[2] J. R. Andrews, and N. Hogan, Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator, in David E. Hardt and Wayne J. Book, editors, Control Of Manufacturing Processes and Robotic Systems, ASME, Boston, pp. 243-251, 1983

[3] M. A. Arbib, Perceptual Structures and Distributed Motor Control, in Handbook of Physiology, Section 2: The Nervous System, Vol. II, Motor Control, Part 1 (V.B. Brooks, Ed.), American Physiological Society, pp. 1449-1480, 1981

[4] R. C. Arkin, "Motor Schema-based Mobile Robot Navigation," The International Journal of Robotics Research, pp. 92-112, August 1989

[5] R. W. Beard, T. W. McLain, M. Goodrich, and E. P. Anderson, "Coordinated Target Assignment and Intercept for Unmanned Air Vehicles," IEEE Transactions on Robotics and Automation

[6] E. Bonabeau, M. Dorigo, and G. Théraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford, University Press, 1999

[7] E. Bonabeau, and G. Théraulaz, "Swarm Smarts," Scientific American, pp. 72-79, March 2000

[8] R. A. Brooks, "A Layered Control System for a Mobile Robot," IEEE Journal of Robotics and Automation, Vol. 2:1, pp. 14-23, 1986

[9] J. Fredslund, and M. J. Mataric, "A General, Local Algorithm for Robot Formations," IEEE Transactions on Robotics and Automation, special issue on Multi Robot Systems, Vol. 18:5, October 2002

[10] D. P. Gillen, and D. R. Jacques, "Cooperative Behavior Schemes for Improving the Effectiveness of Autonomous Wide Area Search Munitions," Proceedings of the Cooperative Control Workshop, Florida, December 2000

[11] O. Khatib, "Real-time Obstacle Avoidance for Manipulators and Mobile Robots," The International Journal of Robotics Research, Vol. 5:1, 1986

[12] B. H. Krogh, "A Generalized Potential Field Approach to Obstacle Avoidance Control," International Robotics Research Conference, Bethlehem, Pennsylvania, August 1984

[13] C. R. Kube, and H. Zhang, "Collective Robotic Intelligence," Second International Conference on Simulation of Adaptive Behavior, pp. 460-468, December 7-11, 1992

[14] M. J. Mataric, "Designing and Understanding Adaptive Group Behavior," Adaptive Behavior, Vol. 4:1, pp. 51-80, December 1995

[15] T. McLain, and R. Beard, "Trajectory Planning for Coordinated Rendezvous of Unmanned Air Vehicles," Proceedings of the AIAA Guidance, Navigation, and Control Conference, Denver, Colorado, August 2000

[16] T. McLain, P. Chandler, S. Rasmussen, and M. Pachter, "Cooperative Control of UAV Rendezvous," 2001 American Control Conference, Arlington, Virginia, June 2001

[17] K. Passino, M. Polycarpou, D. Jacques, M. Pachter, Y. Liu, Y. Yang, M. Flint, and M. Baum, "Cooperative Control for Autonomous Air Vehicles," Proceedings of the Cooperative Control Workshop, Florida, December 2000

[18] M. Resnick, Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds, Cambridge, MA, MIT Press, 1994

[19] C. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model," Computer Graph, Vol. 21:4, pp. 25-34, 1987

[20] Tzu Sun, The Art of War, translated by Lionel Giles, London, Luzac & co., 1910

[21] D. Terzopoulos, X. Tu, and R. Grzeszczuk, "Artificial Fishes: Autonomous Locomotion, Perception, Behavior, and Learning in a Simulated Physical World," Artificial Life, Vol. 1:4, pp. 327–351, 1994

[22] B. B. Werger, "Cooperation Without Deliberation: A Minimal Behavior-based Approach to Multi-robot Teams," Artificial Intelligence, Vol. 110:2, pp. 293-320, June 1999