3-2005

Analysis of Online-Delaunay Navigation for Time Sensitive Targeting

David Chow

ANALYSIS OF ONLINE-DELAUNAY NAVIGATION

FOR TIME-SENSITIVE TARGETING

THESIS

David Chow, Captain, USAF

AFIT/GOR/ENS/05-03

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/05-03

ANALYSIS OF ONLINE-DELAUNAY NAVIGATION FOR TIME-SENSITIVE TARGETING

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

David Chow, B.S.

Captain, USAF

March 2005

AFIT/GOR/ENS/05-03

ANALYSIS OF ONLINE-DELAUNAY NAVIGATION

FOR TIME-SENSITIVE TARGETING

David Chow, B.S.

Captain, USAF

Approved:

________________________________ ____________

Dr. Sharif H. Melouk Date
Thesis Advisor

________________________________ ____________

Dr. Stephen E. Baumert Date
Reader

AFIT/GOR/ENS/05-03

Abstract

Given the drawbacks of leaving time-sensitive targeting (TST) strictly to humans, there is value to the investigation of alternative approaches to TST operations that employ autonomous systems. This paper accomplishes five things. First, it proposes a short-hop abbreviated routing paradigm (SHARP) – based on Delaunay triangulations (DT), ad-hoc communication, and autonomous control – for recognizing and engaging TSTs that, in theory, will improve upon persistence, the volume of influence, autonomy, range, and situational awareness. Second, it analyzes the minimum timeframe need by a strike (weapons enabled) aircraft to navigate to the location of a TST under SHARP. Third, it shows the distribution of the transmission radius required to communicate between an arbitrary sender and receiver. Fourth, it analyzes the extent to which connectivity, among nodes with constant communication range, decreases as the number of nodes decreases. Fifth, it shows the how SHARP reduces the amount of energy required to communicate between two nodes. Mathematica 5.0.1.0 is used to generate data for all metrics. JMP 5.0.1.2 is used to analyze the statistical nature of Mathematica's output.

Acknowledgements

To Dr. M and Dr. B for guidance…to The Commodore for enlightenment…to Jen H for snacks…

Table of Contents

List of Figures

List of Tables

ANALYSIS OF ONLINE-DELAUNAY NAVIGATION FOR TIME-SENSITIVE TARGETING

1. Introduction

1.1 Overview

While the United States Air Force (USAF) has demonstrated a high proficiency at striking fixed targets, it generally lacks the ability to consistently engage time sensitive targets (TSTs) in single-digit minutes. TSTs are difficult to engage because they are usually unpredictable in location, typically involve mobile entities, and must be struck within a small window of opportunity. This thesis focuses on developing and analyzing a framework (SHARP) in which autonomous, cooperative unmanned air vehicles (UAVs) may not only work to identify and engage TSTs but also a) maintain their survivability, b) communicate securely and efficiently – that is: less energy, less equipment; at potentially larger bandwidths –, c) react quickly to a changing priority of targets, and d) maintain connectivity in the network despite a depletion in the number of UAVs in the network.

The following scenario highlights the basic idea and major players in SHARP. Some area of interest, say, a major city in the Middle East is determined to be a hotbed for TSTs – hit and run mortar teams or mobile missile launchers, for example. A constellation of UAVs – intelligence/surveillance/reconnaissance UAVs and strike UAVs – all of which operate under SHARP, are deployed to the area of interest. Twenty-four hours after the constellation arrives over the area of interest, an ISR UAV spots a team terrorists launching mortars into the face of a hotel that is populated with members of the press and foreign dignitaries. That ISR UAV sends an omni-directional signal, requesting a strike UAV; the signal propagates through the constellation until it is received by a strike UAV that is 30 km (a little over 18.5 miles) from the sender of the request. The strike UAV follows the *trail* of the request, through other ISR UAVs, back to the original sender. As soon as the strike UAV is within communication range of an ISR UAV in the *trail*, it gains knowledge of the previous ISR UAV in the *trail* , and immediately redirects itself to that ISR UAV. During the two minutes that the strike UAV will need to travel back to the original sender, the ISR UAV has been keeping track of the terrorist group. The strike UAV will have positive identification/location of the target by the time it has made its trip and much time still remains for further analysis regarding the proper application of force.

The four primary aims, as stated earlier, are best understood in the context of five important factors that make the successful engagement of TSTs possible: 1) *persistence*, 2) *volume of influence*, 3) *autonomy*, 4) *range*, and 5) *situational awareness*. Moreover, any new approach must enable the TST to be struck within a timeframe that is comparable to existing approaches. These factors shall now be discussed in further detail.

1.2 Persistence

This paper defines *persistence* as the continuous amount of time that an airborne asset can exert an influence over a given area. In general, a greater potential for *persistence* is better than less potential. For example, if surveillance of an area was required and an operator at a given air base had the option of two aircraft that were identical in every way except one (Type L) could stay airborne twice as long as the other (Type S), then the operator would choose the aircraft with the capacity to persist in the air twice as long as the other aircraft.

Without *persistence*, operations of aircraft can be degraded. For example, using the two types of aircraft mentioned earlier, if the aircraft type with the shorter *persistence* were used, then a) more than one Type S aircraft would need to be utilized, or b) the Type S aircraft would need to be maneuvered, in order to match the *persistence* of the Type L aircraft. Option A has the disadvantage of requiring a larger inventory of aircraft and increasing maintenance capability to service more aircraft. Option B may not be practical, depending on the type of aircraft. For instance, the speed of the aircraft may make higher levels of *persistence* more difficult, if not impossible – consider that a helicopter is more capable of maintaining persistence than a commercial jet. Calculating forces for a given turning radius and velocity shed light on this claim.

1.3 Volume of Influence

This paper defines volume of influence as the sub-space of the operational arena that can be measured/acted upon by the capabilities of an aircraft. Volume of influence is important because it defines the upper limit of space over which an asset can contribute to a mission.

1.4 Autonomy

This paper defines an autonomous system as one that can perform desired tasks in unstructured environments without human guidance. *Autonomy* could provide at least five benefits for TST-systems: 1) a smaller chance of making mistakes; 2) the delegation of mundane tasks; 3) removal of human risk; 4) removal of guidance limitations; 5) substitution of human limitations for machine-limitations.

The computational advantages that computers have over humans are well understood. Any function of a UAV that can be translated into software is better off performed – with less chance of error – by a machine. When the burden of mundane tasks is placed with machines, human resources are allocated to tasks that require creativity – something that computers, currently, are incapable of. Without human risk, autonomous systems are free to carry out missions that were once considered too perilous. In the absence of human-guidance limitations, TST can potentially take place over any part of the world. Finally, without humans to worry about, the performance of aircraft is only limited by the limitations of the materials and sub-systems that make the aircraft.

1.5 Range

Range is just as important as *persistence*. Greater *range* means an ability to survey land and airspace that is located farther from a home base. Two options exist for an operator who wants to survey land/air space beyond his current limit: 1) obtain aircraft with greater *range* or 2) build more bases at those distant locations. In virtually all instances, Option 1 is the only feasible course of action.

Currently, the USAF does not suffer from a lack of *range*. Over the past 20 years, it has consistently used strategic bombers to execute bombing missions that are thousands of nautical miles (nm) from home base. A good example of this capability is the B-2 bomber, which has an unrefuelled operating *range* that exceeds 6000 nm. The USAF has also demonstrated such *range* with intelligence, surveillance, and reconnaissance (ISR) assets. As far back as the early days of the Cold War, U-2 and SR-71 aircraft performed surveillance missions behind enemy lines. By itself, however, *range* is not very useful in finding/engaging TSTs.

1.6 Situational Awareness

Situational awareness, according to one definition, is a measure of the "amount of awareness that a pilot has about the tactical environment around him." Merriam-Webster's dictionary defines "awareness" as "having knowledge." In this context, I take knowledge to mean possession of facts that are relevant to avoiding a threat in a tactical environment. One source defined situational awareness as "knowledge of one's status relative to the threat in a tactical environment." So, given the previous definitions, situational awareness could reasonably be defined as "the amount of facts possessed by a pilot about a threat in a tactical environment."

The following scenario exhibits the importance of situational awareness. A pilot is being targeted by a surface to air missile (SAM). A pilot in the aircraft is quite limited by the knowledge that he has to avoid this threat in a tactical environment and some assumptions must be made to maximize the pilot's situational awareness. The first assumption is that the target is correctly identified. Secondly, the pilot must recognize this threat in a timely manner and perform some set of actions that he has studied in the tactics, techniques, and procedures (TT&P) and deal with avoiding such a threat. However, in a hazardous tactical environment, the pilot is concerned with many details of his environment: friendly aircraft flying in nearby airspace; other locations from which enemies could assail him; features of the terrain that must be avoided; the myriad of controls at his disposal; et cetera. Due to high stress levels and an imperfect memory, a human being may not perform adequately. The final assumption is that the methods being applied are suitable to the situation. TT&P against threats such as SAMs are heuristics that have demonstrated enough merit to warrant application in various situations and are not perfect. Thus, the situational awareness of a pilot is limited and room for improvement exists.

1.7 Existing Approaches to TST

Existing approaches to TST are highly dependent on manned aircraft and communications between the pilot and leadership on the ground. In general, the following must occur [9]:

First, the TST Cell within Combat Operations Division of an Air Operations Center (AOC) must make an assessment of the situation. The assessment would include ensuring collateral damage is minimized and determining the mobility, hardness, and self-defense capabilities of the TST to ensure a safe and effective combat capable aircraft is utilized against the TST. Second, a radio call under the authorization of the AOC Director will tell the pilot where to go, what to hit, and how much damage to inflict. There may be other aircraft engaged as well or providing close air support. Operators have experimented with pushing coordinates to combat aircraft, however this is of limited use with mobile TSTs. Finally, airspace coordination must occur to ensure altitude separation, or even evacuation of non-player aircraft. Ultimately, the pilot(s) is responsible to navigate to a coordinate or grid zone upon authorization and strike within rules of engagement upon authorization.

Reliance on manned aircraft has a few drawbacks. First, consider that TSTs are unpredictable in time of appearance and location. Thus, the best way to be ready for such a target is to be airborne at all times. Such a requirement is not practical for a pilot in an aircraft. Most of the pilot's time will be spent waiting for something to happen (if something happens). Also, long hours can lead to fatigue, which increases the probability that the pilot will make mental mistakes.

Not only are long hours a potential liability, but coordination between other manned aircraft that are flying in the airspace between the strike aircraft and the target is another hurdle. Considering that the other pilots likely have many tasks that they are performing under conditions of uncertainty, there is a chance that details related to the TST will be overlooked. Such oversights could easily compromise the TST mission.

1.8 Objective

Given the drawbacks of leaving TST strictly to humans, there is value to the investigation of alternative approaches to TST operations. This paper accomplishes six things. First, it proposes a short-hop abbreviated routing paradigm (SHARP) – based on Delaunay triangulations, ad-hoc communication, and autonomous control – for recognizing and engaging TSTs that, in theory, will improve major aspects of prosecuting TST. Second, it analyzes the minimum timeframe needed by a strike aircraft to navigate to the location of a TST under SHARP. Third, it shows the distribution of the transmission radius required to communicate between an arbitrary sender and receiver. Fourth, it analyzes the extent to which connectivity, among nodes with constant communication range, decreases as the number of nodes decreases. Fifth, it shows the how SHARP reduces the amount of energy required to communicate between two nodes. Finally, it determines whether or not the four aims (Section 1.1) can be reasonably obtained using SHARP.

The specific means of coverage – sensor capability – are not addressed in this study. This study focuses on the navigation and communication processes that occur, between UAVs, after a potential TST is identified until a strike UAV arrives in position to strike the potential TST. Not only are the UAVs assumed to be free of human control during those processes, but the UAVs only communicate among each other.

1.9 Thesis Outline

This thesis contains five chapters. Chapter 2 reviews concepts that are relevant to the study. Chapter 3 explains the tools – mathematical and programming – that were used to analyze SHARP. Chapter 4 presents the results of statistical analysis. The final chapter presents insights and conclusions, based on the research, and makes recommendations for further study.

2. Literature Review

The historical application of UAVs in military operations, as well as ideas from ad-hoc communications, computational geometry, and autonomous controls, were applied in this work. A review of certain aspects of each discipline is required in order to understand the rationale behind the methodology of this research.

2.1 Rise of UAVs

2.1.1 Brief History of UAVs in Military Operations

UAVs have a long history of military use [15]. UAVs for military purposes dates at least as far back as the American Civil War, when Confederate forces ineffectively used balloons, laden with explosives, to attack supply and ammunition depots. In World War II, UAVs were used as targets to train anti-aircraft gunners. Since then, UAVs have assumed a more prominent place in military operations. The United States recognized the UAVs utility as an ISR platform after the Israeli Defense Force successfully used the Israeli Aerial Industries Scout UAV during Operation Peace for Galilee, also known as the 1982 Invasion of Lebanon. During Operation Desert Storm, the USS Missouri used Pioneer UAVs as a spotter for Iraqi targets on Faylaka Island, a strategic location near Kuwait City. Most recently, in support of the Global War on Terrorism operations in Afghanistan, the Global Hawk and Predator platforms have been used extensively. Global Hawk has logged more than 1200 flight hrs in support of Operation Enduring Freedom. In operations against terrorists, Predator UAVs have successfully struck al-

Qaida operatives with Hellfire missiles and have been called upon to assist in search-and-rescue missions [19]. The expanding use of UAVs is due to the advantages that they have over manned aircraft.

2.1.2 Advantages of UAVs

First of all, UAVs are smaller and lighter; design considerations that usually have to be made with regard to the pilot are now moot. A smaller, lighter aircraft has some inherent advantages that include improved lift/drag ratio, increased acceleration, and reduced radar cross-section. Also, the cost of the air vehicle is typically a smaller percentage of the overall system when compared to the aircraft that requires onboard personnel [22].

Second, a UAV has the potential to perform maneuvers that would be impossible with personnel onboard. The forces under which a high-performance manned aircraft can operate are limited by the capacity of the onboard personnel to withstand such forces. Humans can experience visual impairment or total blackout if subjected to four to six multiples of gravity ("G's") after only a few seconds. Now, consider that a plane can be under as much as 9 G when pulling out of a dive, which typically takes more than a few seconds.

Third, UAVs can provide *persistence* over an area for much longer periods than an onboard pilot can endure. UAVs do not get tired, or hungry. Also, UAVs have the potential for far greater ranges since pilot specific constraints are no longer a factor. Since TSTs can appear virtually anywhere at any time, *persistence* and *range* need to be maximized [16].

Finally, operations can be conducted without considering the well being of onboard personnel. UAVs can be sent to search for surface-to-air missile sites without having to considering the event of a rescue mission. Even if a UAV is shot down, there is no chance of a loss of life [22].

2.2 Ad-Hoc Communications

Mobile nodes that wirelessly collect/process/send information among each other without centralized administration characterize an ad-hoc network. Work on such a communication paradigm can be traced as far back as 1968, when educational facilities in Hawaii were to be linked together by a one-hop, distributed channel-access management scheme. Later, in 1973, DARPA (Defense Advanced Research Projects Agency) began work on a multi-hop network that would be used in military applications. Today, the work that DARPA conducted has evolved into technology that enables individuals to connect their personal computer with their printer and wireless router [14].

Due to the mobility and growing use of UAVs on the battlefield, much research on connecting such assets within the framework of an ad-hoc network has been conducted. Work in [29] addressed the development of polynomial time algorithms to assess sensor coverage of an area. In anticipation of future battlefields that rely on UAVs within an ad-hoc network and have worked on developing new routing protocols [38]. [8] addresses the interaction of remotely controlled UAVs within an ad-hoc networking.

Ad-hoc wireless networks involve challenges that normally are not associated with traditional wireless networks. Nodes in an ad-hoc network are usually battery powered and have a limited memory. For this reason, creation of a network that minimizes energy consumption and memory requirements is of great value [26].

2.3 *Delaunay & Voronoi Geometry*

While application of Delaunay and Voronoi geometry does not require knowledge of every nuance, a few characteristics of these constructs are important to this thesis. Thus, some background knowledge on related mathematics, in addition to select details of Delaunay and Voronoi geometry, follows. Some ideas that are formally described later in this thesis are alluded to in order to highlight the relationship between DT or VD and those ideas. The terms "node" and "UAV" are synonymous in this context.

The following definitions are given by [37]. A triangulation is defined as a simple plane graph where every face boundary is a 3-cycle. A graph, *G*, is defined by a vertex set $V(G)$, an edge set $E(G)$, and a relation that associates with each edge two vertices (not necessarily distinct) called endpoints. In this study, all endpoints are distinct. A graph is planar if it has a drawing without crossings. Such a drawing is called a planar embedding of *G*. A plane graph is a particular planar embedding of a planar graph. Faces of a plane graph are the maximal regions of the plane that contain no point used in the embedding (maximal is equivalent to "no larger one contains this one"). An *n*-cycle is a cycle with *n* vertices. Triangulations are important in this study because they adequately model the partitioning of an *area of interest*. However, not all triangulations are created equal. Justification for the use of Delaunay triangulations and their dual graph, Voronoi diagram (VD), now follows. (Note: the *dual graph,* G^*, of a plane graph *G* is a plane graph whose vertices correspond to the faces of *G*. The edges of *G** correspond to the edges of *G* as follows: if *e* is an edge of *G* with face *X* on one side and face *Y* on the other side, then the endpoints of the dual edge $e^* \in E(G)$ are the vertices *x*, *y* of *G** that represent the faces *X*, *Y* of *G*. The order in the plane of the edges incident to $x \in V(G^*)$ is the order of the edges bounding the face *X* of *G* in a walk around its boundary).

According to [3], VDs and DTs are fundamental constructs defined by a discrete set of nodes in two and three dimensions. Figure 2 depicts both constructs.

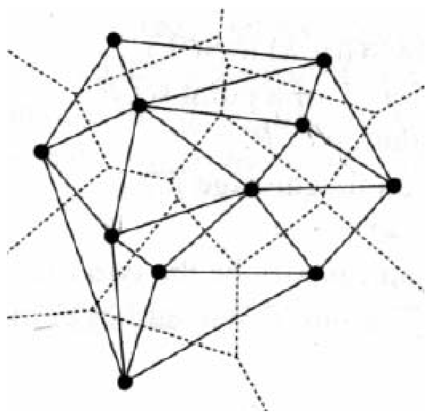

Figure 2.1: Voronoi diagram (dashed lines) and Delaunay triangulation (solid lines)

By definition, the VD, of a set of nodes, partitions the plane into a set of convex polygons (Voronoi regions) such that all points within a given polygon are closest to the node assigned to that polygon. In Figure 2.1, the nodes are the large dots and all points within the Voronoi region that is assigned to a particular node is closest to said node. Furthermore, points that lie on an edge are equidistant between exactly two nodes; points on a polygon's vertex are equidistant from exactly three nodes. The DT of a set of nodes is constructed by drawing edges between points whose Voronoi regions share an edge. In order for the DTs to be unique, the following three assumptions are made: 1) the nodes are not all located on a straight line; 2) the number of nodes in *V(G)* is three or more but finite; and 3) no more than three nodes are located on the same circle. If those assumptions are satisfied, then the points are said to be in *general position.* In fact, a triangulation is a DT if and only if no circle that goes through the points of the triangles of the DT contains other nodes. In a sense, each node is assigned an optimal region of *influence*. This idea can be extrapolated to the concept of *volume of influence*.

Given that large scale military operations, such as TST, usually entail voluminous amounts of data and frequent calculations, the use of any geometrical construct should be easily adaptable to computer processing. Fortunately, DT and VDs meet such criteria. [1] points out that representing VDs in a computer can be accomplished by standard data structures. Also, the memory required to store the constructs grows only as $O(n)$. Given the low cost of computer memory, even very complex geometries can be maintained with ease. [3] shows that the construction of n points, whose x-coordinates are pre-sorted, grows at $O(n \log n)$. Thus, the rapid construction of a constantly changing configuration of a given number of nodes is computationally friendly.

Because the nodes form and (in this thesis) work together as a network, the minimum spanning tree is an important characteristic. The minimum spanning tree is the set of edges, with minimum weight, that keeps the nodes connected. A graph is connected if a path between any two nodes exists. [1] points out that the minimum spanning tree (also called shortest connection network) is contained in the DT. Thus, once the minimum spanning tree is found, no connections above and beyond those that already exist in the graph need to be connected. The fact that the average degree of a vertex in DT is less than or equal to 6, as shown in [3], is also significant because in the event that communication routing is used without assistance of a network-wide, centralized processor, a given node is limited to the number of interactions that must take place between nodes in neighboring Voronoi regions. As a reminder, communication routing is not addressed in this thesis.

Not only can a VD can be created for a given set of nodes, but also [3] a given partition of a plane into convex hulls can be determined, via linear programming, as a VD in $O(n)$ time; if the partition is a VD, then the nodes for each partition can be determined in $O(n)$. While the theorem is quite powerful, it is beyond the scope of this paper. An option for including this theorem in future research is presented in Section 5.2.

2.4 Autonomous Routing

Autonomous routing is accomplished by a system that is able to vector itself between a starting location and an ending location by relying solely on some internal formula. Significant work in the realm of 1) perception and motion, 2) evolutionary behavior for navigation control, 3) coordination of fleets, 4) spatial navigation, and 5) distributed task allocation has been done. These elements all play a role in the composition of this research.

2.4.1 Perception and Motion

The study of the perception and motion of autonomous systems seeks to find better ways to enabling systems to sense and adapt to its environment without human intervention. In [10], some of the challenges of developing perception and motion capabilities for a planetary robot are discussed. First, due to its size, the robot's computing power, memory, and energy are limited. Second, the isolation of the robot makes constant contact with a control station difficult if not impossible. Thus, a high degree of *autonomy* is desirable. Similar challenges would also face an autonomous UAV.

More recently, [35] focused on enhancing a UAV's ability to "see and avoid" obstacles within predetermined limits of its environment. Schouwenaar et al focused on improving the practice of calculating safe routes based on a mixed-integer linear programming (MILP) routine. The challenge was to calculate the optimal path that an autonomous UAV can traverse between two points in space while avoiding obstacles. At least two glaring shortcomings exist in the MILP approach. First, computation time increases at, at least, a quadratic rate, making the approach impractical for large problems. Second, the MILP is computed *off-line*; thus, the approach is not robust against changes in the UAV's environment. In this context, *off-line* refers to some time before the UAV begins operations. *Off-line* is in contrast to the idea of *online* navigation/routing, where the UAV would have no predetermined knowledge of its environment and would have to gain such information during operations (see [7] and [35]). These problems are somewhat solved by applying a receding horizon (RH) planning strategy – in essence, a strategy that takes inputs from the environment up to a certain distance from the UAV. However, RH-MILP does not guarantee collision avoidance. The UAV could still maneuver too close to an obstacle that is just beyond the horizon of the current time step. The MILPs for UAV navigation do not explain how information about the aircraft's environment is obtained. SHARP could be a means of accessing different levels environment data, depending on which of the two major navigation philosophies are being used: or off-line (memory intensive) or online (computationally intensive). As state earlier, off-line navigation requires a predetermined, internal representation of the environment and any hazards to the UAV. In order for a computer to make sense of the environment data, the data must in the form of some non-volatile memory – hence, memory intensive. On the other hand, online navigation is done in real-time, without any previous knowledge of the environment. Thus, survivability of the UAV would be highly dependent on the ability of an on-board computer to calculate, in real-time, all the aspects of the UAV's environment that may be a hazard – hence, computationally intensive.

2.4.2 *Evolutionary Behavior for Navigation Control*

The study of evolutionary behavior for autonomous systems seeks to develop systems that can adjust to the environment without any internal representation of that environment – in other words, the actions of the autonomous system would be solely dependent on the inputs received from on-board sensors (also, reference Section 2.4.1). The work by [4] revolves around self-optimizing behavioral navigation controllers – via simulation – for autonomous, fixed wing UAVs using multi-objective GP (genetic programming). Essentially, the controllers have to be good enough to locate, navigate toward, and circle various kinds of radar without human assistance. [4] Is relevant to this thesis in at least three ways. First, it puts navigation – the focus of this paper – in the context of other aspects of behavioral control, such as obstacle avoidance, light seeking, game playing, et cetera. Secondly, his work helps to validate the idea that handing over tasks, which were normally reserved for human attention and were in the context of military operations, to a machine is feasible. Third, this work considers the possibility of UAVs having a degree of navigational dependency between autonomous machines and working cooperatively.

2.4.3 Coordination of a Fleet of UAVs

Often a team will perform a task more efficiently than an individual. Yet, a team may perform less efficiently than an individual if the dependencies between team members are not well defined. For this reason, the research by [20], regarding coordination and control of real multi-vehicle – rovers, blimps, and fixed-wing aircraft – scenarios is valuable. The authors emphasize that the workload and responsibilities of UAVs are increasing while the control structures have not been upgraded to account for such changes.

Their research takes advantage of receding-horizon task assignment to reduce the computation complexity (NP-Hard) of task assignment with precedence constraints – prioritized targets. Next, they optimize trajectories of vehicles using a MILP based receding-horizon planner, augmented by pruning and graph search algorithms. Finally, two test beds – 1) four blimps plus eight rovers; 2) eight autonomous UAVs – are created to test the feasibility of their coordination algorithms in a real-time network of autonomous agents. Furthermore, all vehicles were equipped with commercially available laptop computing technology; wireless communication was accomplished with off the shelf Ethernet components

Each test bed demonstrated the utility of autonomous vehicles. In the case of the blimps and rovers, the blimps were able to discover waypoints (in this case, arbitrary locations of interest) more quickly and with greater ease due to their line of sight and unimpeded direction of movement (obstacles were present on the ground). Rovers complemented the blimps because rovers, unlike blimps, had the ability to engage targets on the ground. In the case of fixed wing planes, the vehicles were able to successfully maneuver over a series of waypoints as information about waypoints was being uploaded to them in real-time. Advances have also been made in cooperative path planning under timing constraints. For example, [29] uses a coordination variable and coordination function to control autonomous agents. In that work, a coordination variable is a vector that captures all information that is needed for UAVs to accomplish a common objective; a coordination function is a means to quantify the cost, to a given UAV, of achieving goals that are common to all members of the group.

3. Methodology

3.1 Overview

SHARP is based on a configuration of ISR UAVs and a number of strike UAVs. When an ISR UAV spots a TST, the information about the target is relayed through a series of ISR UAVs and, ultimately, to the nearest strike UAV. Consider such a series of ISR UAVs and the edges that connect them to be the *path of information*. Then, the strike aircraft navigates to the vicinity of the ISR UAV that spotted the target, using assets in the *path of information* as a frame of reference. The path flown by the strike aircraft is a function of the path of information; as soon as the strike aircraft is within communication range of an ISR UAV, the strike aircraft redirects itself to the next node in the *path of information*. In order to model SHARP, UAVs are represented by nodes in a two-dimensional plane; the space in which the nodes exist is assumed to be an area in which TSTs are suspected. Each UAV has a certain radius of communication within which it can communicate with any other UAV. Further consideration regarding the key characteristics of SHARP now follows.

3.2 Nodes

Within an area of interest, a TST can appear virtually anywhere at any time. One way to maximize the probability of spotting such a target would be to have surveillance systems uniformly spaced within the airspace above that area. Because the exact configuration of nodes will vary, this characteristic is modeled as a random, uniform distribution of nodes over a square area. Furthermore, these nodes are assumed to be virtually stationary from the time that a request for a strike UAV is sent to the moment that the strike UAV receives the request, since the signal travels at the speed of light with some negligible delays when the signals get rerouted at the ISR UAVs.

3.3 Interaction between Nodes

3.3.1 Capabilities of Nodes

The maximum transmission range per node is a critical characteristic of SHARP. In general, greater transmission ranges increase the number of nodes that can communicate with each other. Yet, an engineering trade off is made between communication capability and flight performance. As two nodes move farther apart, more power is required to successfully send wireless communications. More power generally equates to heavier communication equipment, which will degrade the flight performance of the aircraft. More weight effects an aircraft in a few ways, including a) stall velocity increases; b) for a given lift to drag ratio, thrust must increase, which implies more fuel burned; and c) without a thrust to weight ratio greater than one, vertical acceleration is not possible. Of course, having a transmission radius that is too large may make the wireless network vulnerable to signals collection by an adversary.

3.3.2 Connectivity

In order for SHARP to work, the configuration of ISR UAVs and strike UAVs must be connected. In other words, any two nodes must be able to communicate along the path of information. This study does not put any restrictions on the transmission radii of the nodes. Such flexibility is useful since the detection of a TST is the first priority and, in general, the probability of detection increases as the number of UAVs increase. Thus, the number of UAVs for a given area must be determined before the problem of communication capability can be adequately defined. Without putting restrictions on transmission radius, a better sense of what transmission capability is necessary for a given number of nodes in an area of interest can be gained.

Connectivity and energy concerns (Section 3.3.4) are closely related. Assuming that UAVs have limited communication ability – especially with respect to having enough energy reserves to sustain communication over time – restricting communication to nodes that have adjacent Voronoi regions is considered. For this reason, the DTs are used to ensure that all communication is relayed through only the nodes with adjacent Voronoi regions. For a given *path of information*, the minimum transmission radius required is the longest edge in the *path of information*.

Connectivity between ISR UAVs and strike UAVs, in particular, is special for at least two reasons. First, without communication, a strike UAV that is en route to a target may needlessly travel to the target if the target has been deemed benign. Secondly, without constant communication, the strike UAV may not get rerouted to a higher priority TST that has been spotted in the strike UAV's vicinity.

3.3.3 Signal Routing

Applying routing algorithms to communication within a network is usually desirable when the sender of a message is targeting a specific node to receive the message. Routing could also help alleviate bandwidth constraints that will likely be present in a data/voice-intensive network within a theater of war. With many potential paths between any two UAVs in a constellation, communication could be rerouted along edges of the Delaunay triangulation that aren't currently being used to capacity. In the case of TST, though, such communication could be a disadvantage since the objective is to contact the closest strike aircraft, which has an unknown location. Thus, omni directional signaling between nodes is the best way to relay messages in a time sensitive scenario. Routing is not addressed by this study.

3.3.4 Energy Concerns

An energy efficient communication network is one that minimizes the amount of energy that is required to communicate a message between two nodes in the network. As [30] points out, the practice of multi-hop communication can effectively reduce the power consumption in an ad-hoc network. The energy required for communication between two nodes is strongly dependent on the distance between the nodes. The relationship between energy and distance is given by $E = B \cdot d^{y}$, where *E* is energy, *B* is a proportionality constant that describes overhead cost per bit; *d* is the distance between nodes; and *y* is the path loss exponent depending on the RF environment. In this thesis, $B = 2.3$ and $y = 2$ (assuming free space propagation). In general, greater energy efficiency is achieved when communication occurs over several short hops. Energy is also used when information is processed at each node, but it is about two orders of magnitude less than the energy that is required to communicate between nodes. The Delaunay triangulation provides a way to find the multi-hop path that is of minimum distance.

Transmitted signals from nodes are modeled as circles around nodes. The circles can be interpreted from at least two different perspectives. On one hand, they imply that each node is equipped with omni-directional antenna. On the other hand, they can be interpreted simply as limits on transmission ranges, without regard to the type of antenna used – directional or omni-directional. This study assumes that all nodes communicate via omni-directional antenna. Furthermore, all nodes are assumed to have the same maximum transmission distance.

3.4 Navigation

A node cannot communicate with other nodes unless it is within the maximum transmission distance of another node. Thus, having enough transmission capability is a concern. However, the ideal network would not have excessive communication capability. In this study, nodes are assumed to have a transmission distance that is at least as long as the longest leg in the shortest path between two arbitrary nodes.

The shortest path may not be unique, but only one path needs consideration for at least two reasons: 1) the location of a TST is always relayed from an ISR UAV to a strike UAV; 2) the time required to find the alternate shortest path would be an unnecessary delay.

The strike aircraft alters its direction as soon as it can receive commands from the ISR UAV to which it is headed. Once the strike aircraft is within transmission range of an ISR asset, the strike UAV is vectored to the next ISR UAV in the shortest path, until it reaches the ISR UAV that initiated the transmission. Such a path is valuable for at least three reasons. First, along such a path, strike UAVs are within communication radius of ISR UAVs. Second, it restricts strike UAVs to areas that are at least safe enough for ISR UAVs. The assumption that ISR UAVs are generally more vulnerable to attack is made. Third, in the event that the strike UAV is forced to the ground, it does so in close proximity to an ISR UAV, which would be used to easily assess the condition of the downed aircraft. Figure 3.1 depicts the essential elements of the navigation.

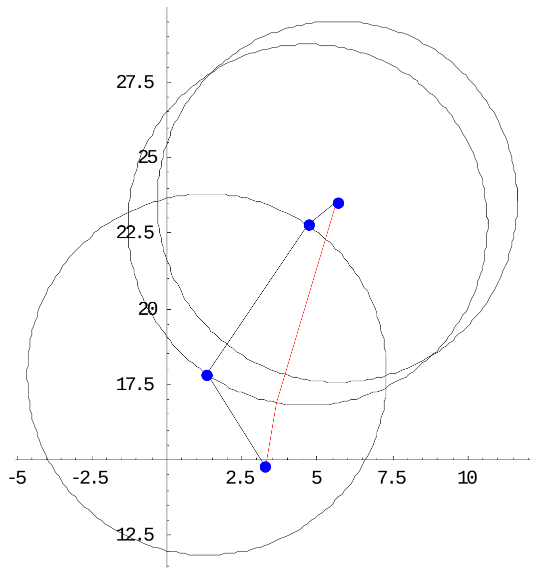

Figure 3.1: Path of Information (edges between nodes) and path via SHARP (red line)

Only edges between nodes with adjacent Voronoi regions make up the *path of information*. Communication regarding the location of a TST would follow the edges in the path of information. In Figure 3.1, the node farthest from the horizontal axis happens to be the node that initiates communication. The communication continues until a strike UAV is found – in this case, the node that is closest to the horizontal axis. In order to fly to the UAV that originated the communication, the strike aircraft follows the SHARP path, depicted as a red line in Figure 3.1. Following the SHARP path requires the strike UAV to have information about the nodes in the *path of information* – otherwise, the strike aircraft would not know how to travel to the ISR UAV that made the original request for assistance. Many feasible solutions exist. One mechanism for applying *path* *of information* data to the strike UAV, so that the correct SHARP path can be followed, will now be presented.

Somehow, the locations of the ISR UAVs, at the time the request for assistance was received, in the path of information must be available to the strike UAV. One way of ensuring this involves including cumulative geo-location data, for the ISR UAVs in the *path of information*, in the request, as it gets passed from one ISR UAV to another. Figure 3.2 depicts essential aspects this heuristic on a sub-graph of Figure 2.1.

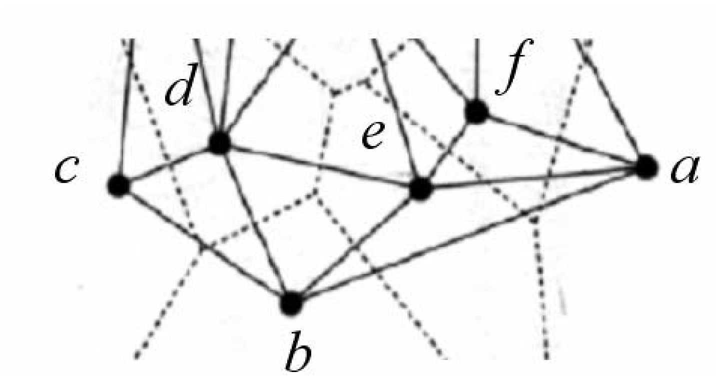

Figure 3.2: A graph where nodes $\{a, b, d, e, f\}$ represent ISR UAVs
{c} is the only strike UAV and {a} is the original requestor for a strike UAV

Before demonstrating the steps of the heuristic, some conventions should be considered: 1) "x" identifiers denote inactivity; 2) once a UAV has received a given TST signal, say 'TST_a_13:00:00', then any future reception of that signal will be discarded by that UAV. However, another TST signal, say 'TST_a_13:00:23', would not be discarded. 3) A signal that is killed by a UAV, because the same signal has already been forwarded, is denoted by KILL(…). 4) When the nearest strike UAV receives the TST-signal, the event is denoted by DONE(…) and sends a receipt to the initial ISR UAV. 5)

If the original signal has been relayed through a series of UAVs ($x_0, x_1, ..., x_n$), then this fact is denoted by a string in the form of “ $x_0, x_1, ..., x_n$ ”, where the originator is x_0 and x_n is the most recent receiver. That being said, the heuristic is now presented.

Table 3.1: Retrieving the *Path of Information* through inter-ISR communication

	0	1	2	3	4	5	6	7	8
a	INIT	x	x	x	x	x	x	x	x
b	x	x	x	x	x	a	KILL(a,e)	x	x
c	x	x	x	x	x	x	x	x	DONE(a,e,d)
d	x	x	x	x	x	x	x	a,e	x
e	x	x	a	KILL(a,f)	x	x	x	x	x
f	x	a	x	x	KILL(a,e)	x	x	x	x

Let the variables *a* through *f,* in Figure 3.2, signify the geo-locations of a set of UAVs at the moment they receive a TST-signal. Let all nodes in Figure 3.2, except *c,* be ISR UAVs; let *c* be a strike UAV; let *a* be the one that initiated the assistance of a strike UAV. The propagation of *a*’s request can be analyzed using a table such as Table 3.1, where rows represent nodes in the graph and columns are used to catalog the order in which various nodes received a specific TST-signal from other nodes. Assume that the TST signal is uniquely identified by notation ‘TST_<origSend>_<univTime>’, where <origSend> identifies the ISR UAV that originally requested a strike UAV, and <univTime> is Universal Time. The construction of Table 3.1 will now be addressed.

Column 0: Since *a* is the originator of the TST-signal, an INIT identifier is placed in cell (a,0). Column 1: *f* is the first receiver of the TST-signal, and this fact is represented by the “a” identifier in cell (f,1). Column 2: The next node to receive a signal, in this case from *a,* is *e*; thus, an “a” is placed in cell (e,2). Column 3: *e* is the next UAV to receive the TST-signal, which happened to arrive via the following sequence of nodes: (a,f). But since *e* has already received the TST-signal, the signal is killed in cell (e,3). Column 4: Shortly after, *f* kills the signal relayed through (a,e). Column 5: The next node to receive a signal from the TST-signal is *b*, thus an "a" is placed in column five. Column 6: By the time the signal via (a,e) has arrived at *b*, *b* has already received the signal along a shorter path, thus KILL(a,e) is placed in cell (b,6). Column 7: Shortly after, the TST-signal via (a,e) reaches *d*, so (a,e) is placed in cell (d,7). Column 8: Finally, since the path (a,e,d,c) is slightly shorter than (a,b,c), and since *c* is the strike UAV, DONE(a,e,d) is placed in cell (c,8). Note how *c* now has the geo-location waypoints for the path of information. Proper SHARP navigation can now be accomplished.

3.5 Variables

The model will vary two factors – 1) the number of nodes, and 2) the surveillance area – both at three levels. The number of nodes will be varied at levels of 25, 50, and 75. The sides of the squares, in which nodes may appear, will be tested at lengths of 25, 50, and 75 (assumed to be kilometers). Five hundred replications are performed for each combination.

3.6 Metrics

The methodology of this research enables analysis of at least four aspects of SHARP: 1) the ratio between the SHARP path and the Euclidean distance between an arbitrary sender and receiver – heretofore the *direct length*; 2) the distribution of the minimum transmission radius required to communicate between an arbitrary sender and receiver, along the shortest path of the Delaunay triangulation; 3) the difference of energy consumption, attributed to communication, between communicating along the *direct length* versus SHARP's *path of information*; and 4) the extent to which nodes, within a predefined space and with a given transmission radius, lose connectivity as the number of nodes decreases.

3.7 Two Procedures

The first three metrics are calculated by a procedure, Proc1, that a) uniformly distributes a predefined number of nodes in a predefined area; b) selects two random nodes and finds the shortest path, along the Delaunay triangulation, between said nodes; c) computes the minimum transmission radius that would allow the nodes in the shortest path to communicate; d) calculates the SHARP path.

The last metric is calculated by a second procedure, Proc2, that varies slightly from the first procedure. The second procedure allows the transmission radius to be set to arbitrary distances and calculates the number of shortest paths along the Delaunay triangulation that contained nodes that could not communicate with each other. Figure 3.3 is an example of such a path, where the rightmost node is not within the radii of another node.

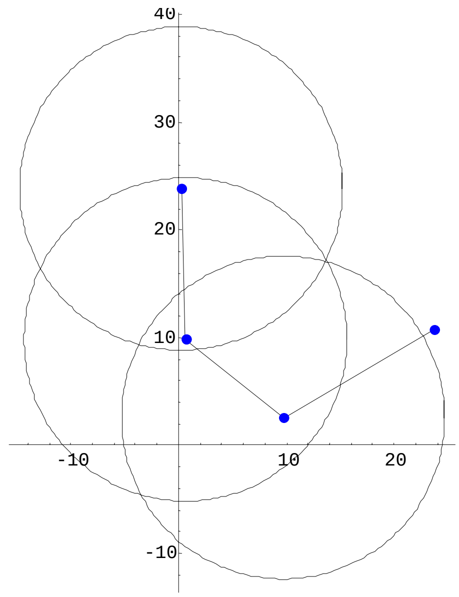

Figure 3.3: A SHARP Path with Inadequate Transmission Radii

Mathematica 5.0.1.0 is used to generate data for all metrics. JMP 5.0.1.2 is used to analyze the statistical nature of Mathematica’s output. The details of each procedure are in Appendix A.

4. Analysis/Results

The methodology was designed to address four aspects of SHARP: 1) the minimum distance need by a strike aircraft to navigate to the location of a TST under SHARP; 2) the distribution of the transmission radius required to communicate between an arbitrary sender and receiver; 3) the extent to which SHARP reduces the amount of energy required to communicate between two nodes; and 4) the extent to which connectivity, among nodes with constant communication range, decreases as the number of nodes decreases. The analysis of each metric, for all scenarios, will be discussed now.

4.1 Minimum Distance – SHARP path

Let x be a path followed by the strike aircraft under SHARP and let y be the corresponding *direct* length. The first metric is collected as x/y over the various combinations of nodes and areas. Percentile statistics best capture the significance of this metric. Percentiles reveal the percentage of paths that were below various values; in general, improvement occurs when a given percentile is matched against smaller SHARP paths.

Table 4.1: Percentiles for ratios between SHARP path and direct path

%	25n,25s	25n,50s	25n,75s	50n,25s	50n,50s	50n,75s	75n,25s	75n,50s	75n,75s
100	1.129	1.218	1.153	1.231	1.236	1.240	1.229	1.306	1.225
99.5	1.099	1.160	1.139	1.179	1.171	1.215	1.201	1.217	1.188
97.5	1.056	1.063	1.060	1.106	1.102	1.119	1.150	1.147	1.130
90.0	1.018	1.023	1.024	1.052	1.045	1.043	1.080	1.072	1.064
75.0	1.003	1.003	1.003	1.018	1.015	1.015	1.026	1.027	1.028
50.0	1.000	1.000	1.000	1.002	1.001	1.001	1.005	1.005	1.005
25.0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10.0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2.5	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.5	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.0	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

n: denotes number of nodes;
s: the length of a side of a square area

According to the percentile data in Table 4.1, the SHARP path will be nearly equal to the direct path in the vast majority of instances. In just a few instances, the SHARP paths had excessively long legs – one SHARP path was 1.240 times the direct path and another was 1.306 times the direct path. Long SHARP paths in this model are likely a result of the fact that the Delaunay triangulation always contains the convex hull of a set of points (see Figure 2.1). Fortunately, these very long edges do not occur frequently – say, once or twice in 500 replications. One solution to eliminating edges that are too long entails adding more UAVs to the area of interest. Another solution could involve the use of receivers to measure the relative strength of signals and then restrict movement of UAVs in the event that signals are received below some power threshold.

These percentiles would be of great interest in a real-world implementation of SHARP, because if a constellation of x UAVs has a high probability of delivering SHARP paths that are virtually equivalent to a constellation with cx UAVs, where c is some positive constant, then the option with fewer UAVs will cost less.

4.2 Distribution of Minimum Transmission Radius

In order for communication to occur between any two end nodes, along a series of inner nodes, the minimum transmission radius of the nodes must be at least as long as the longest edge between adjacent nodes in the *path of information*. For different configurations, randomly placed by a uniform distribution over an area of interest, and for randomly chosen end nodes, the minimum transmission radius will take different values. A histogram of such radii can be created, resulting in a distribution. Table 4.2 shows percentile data gathered from the histograms of each scenario that was tested.

Table 4.2: Percentiles of Minimum Transmission Distance
Required for Communication between Two Nodes

%	25n,25s	25n,50s	25n,75s	50n,25s	50n,50s	50n,75s	75n,25s	75n,50s	75n,75s
100.0	22.996	49.393	80.056	24.147	47.371	70.418	24.079	45.645	71.747
99.5	21.727	46.359	68.936	23.040	44.907	66.437	23.203	43.454	68.365
97.5	19.190	40.502	59.429	20.206	39.615	57.583	19.257	37.638	58.377
90.0	14.904	29.488	46.081	15.116	29.858	43.066	13.742	28.141	43.764
75.0	11.016	22.189	32.480	9.136	19.071	29.803	8.573	17.939	30.419
50.0	8.634	16.224	25.259	6.894	13.310	20.407	6.036	12.049	17.377
25.0	6.334	12.666	18.985	5.347	10.255	16.347	4.562	9.240	13.981
10.0	4.129	9.426	13.511	3.964	7.467	12.101	3.531	7.303	10.540
2.5	2.387	4.179	8.188	2.213	4.094	7.738	2.132	4.496	6.071
0.5	1.295	0.716	4.299	1.305	1.004	2.871	0.671	1.415	1.281
0.0	1.083	0.419	1.934	0.682	0.791	0.723	0.291	0.688	0.859

n: denotes number of nodes;
s: the length of a side of a square area

One may guess that, for a given area, the minimum transmission distance decreases as nodes increases. Likewise, minimum transmission distance should increases, for a given number of nodes, as area is increased. However, the degree of these changes is more important than the mere upswing or downswing of values. Due to randomness in the generated constellations and the relative increase in nodes or area, some scenarios do not follow the general trends.

For example, consider the data in Table 4.3 for all scenarios with 25^2 areas. Table 4.3 shows means and confidence intervals for the transmission distances under the various scenarios. Overlapping confidence intervals is more evidence that the distributions between any two scenarios are not statistically significant. When the number of nodes is increased from 25 to 50, one would expect the transmission distance to decrease. Such a decrease does occur, and the decrease is significant since the confidence intervals for the means of each scenario do not overlap. However, the same cannot be said for the (50n,25s) and (75n,25s) scenarios. This is an example of significant increase in nodes (a 50% increase) that does not result in appreciably shorter transmission distances. The interval of the mean for the (50n,25s) scenario is 7.630 – 8.389 and the interval of the mean for the (75n,25s) scenario is 6.966 – 7.721. A slight overlap of the two intervals exists, suggesting that the difference is not as statistically significant as, say, the difference between the means of the (50n,25s) and (50n,50s) scenarios, where no overlap exists.

Table 4.3: Mean; Low & High Confidence Intervals for
Minimum Transmission Radius ($\alpha = 0.95$)

	25n,25s	25n,50s	25n,75s	50n,25s	50n,50s	50n,75s	75n,25s	75n,50s	75n,75s
Mean	9.113	18.116	27.337	8.009	15.733	24.349	7.344	14.694	22.861
upper 95% Mean	9.477	18.857	28.448	8.389	16.499	25.451	7.721	15.423	24.052
lower 95% Mean	8.750	17.374	26.227	7.630	14.967	23.247	6.966	13.964	21.670

With a few negligible exceptions (see the data at the 0.5 and 0.0 percentile in Table 4.2), transmission distances, at the same percentile and over the same number of nodes, steadily and significantly increase. For example, the transmission distances at the 90 percentile for scenarios with 25 nodes were 14.904, 29.488, and 46.081. Data in

Table 4.3 shows that the confidence intervals are clearly separated in scenarios with the same number of nodes (much more so than in scenarios with the same area). The differences are much more clear because a given unit of change in the dimension of the squared area has a greater effect on density of nodes than the same unit of change in the quantity of nodes does.

Data regarding the relationship between density of nodes and the Delaunay triangulation appears to have significant consequences with respect to minimum transmission radius. Recall that the Delaunay triangulation always contains the convex hull of a set of nodes. This fact can be a liability if edges along the convex hull are too long for transmission. Also, recall that the scenarios with a higher density of nodes, for a given number of nodes, always yielded significantly lower transmission distances at a given percentile. With these facts in mind, one may jump to the conclusion that simply maintaining a particular density of nodes over larger areas would solve the problem of unacceptably large transmission radii. Data from the model suggests otherwise.

The model was run for two scenarios with the same densities. One scenario simulated 25 nodes in a 5^2 km^2 and the second scenario simulated 100 nodes in a 10^2 km^2. Table 4.4 reveals the percentile statistics collected for these two scenarios. Not only are the node densities equal, but also the larger area can be divided into four areas that are equal to the area of the smaller area.

Table 4.4: Percentiles of Minimum Transmission Distance Required for Communication between Two Nodes in Configurations with Equivalent Density of Nodes

%	25n,5s	100n,10s
100.0	4.732	9.742
99.5	4.611	9.336
97.5	3.936	8.405
90.0	2.928	6.180
75.0	2.270	3.889
50.0	1.619	2.205
25.0	1.237	1.712
10.0	0.850	1.278
2.5	0.425	0.910
0.5	0.182	0.405
0.0	0.013	0.180

n: denotes number of nodes;
s: the length of a side of a square area

Table 4.5: Mean; Low & High Confidence Intervals for Minimum Transmission Radius ($\alpha = 0.95$)

	25n,5s	100n,50s
Mean	1.801	3.041
upper 95% Mean	1.876	3.218
lower 95% Mean	1.726	2.864

n: denotes number of nodes;
s: the length of a side of a square area

As the data shows, the longest transmission that was required to communicate between two nodes in the (25n,5s) scenario was nearly half that of the (100n,10s) scenario – 4.732 versus 9.742. Moreover, at every percentile, a shorter transmission was required in the (25n,5s) scenario. Since the 10^2 km^2 area can be divided equally into four 5^2 km^2 areas, the maximum transmission radius can be shrunk considerably by substituting a large Delaunary triangulation with four smaller-scaled Delaunay triangulations.

4.3 *Energy Savings*

Let x be the energy consumed over a path in SHARP and let y be the energy consumed if transmission over the direct path were used. Energy savings is measured as x / y. Table 4.5 shows percentile statistics for energy savings under different scenarios.

Table 4.6: Percentiles of Energy Savings for SHARP

%	25n,25s	25n,50s	25n,75s	50n,25s	50n,50s	50n,75s	75n,25s	75n,50s	75n,75s
100.0	4.104	3.717	3.848	2.436	3.952	5.074	4.248	3.705	4.712
99.5	3.360	3.359	3.134	2.409	3.200	3.346	3.012	2.979	2.824
97.5	1.849	1.830	1.578	1.622	2.074	1.834	1.623	1.560	1.789
90.0	1.112	1.111	1.083	1.002	1.141	1.083	1.016	1.079	1.035
75.0	1.000	1.000	1.000	0.878	0.972	0.978	0.834	0.817	0.863
50.0	0.761	0.763	0.729	0.567	0.609	0.617	0.549	0.554	0.562
25.0	0.566	0.562	0.557	0.432	0.442	0.451	0.383	0.404	0.405
10.0	0.451	0.456	0.447	0.339	0.350	0.334	0.302	0.295	0.315
2.5	0.365	0.370	0.365	0.286	0.286	0.264	0.240	0.251	0.251
0.5	0.298	0.321	0.323	0.247	0.256	0.208	0.196	0.207	0.216
0.0	0.286	0.314	0.303	0.220	0.250	0.191	0.178	0.192	0.210

n: denotes number of nodes;
s: the length of a side of a square area

The data shows that energy savings occurred in approximately 75% of all paths taken. While SHARP rarely resulted in more energy consumed, compared to communication over the *direct path*, the degree of excessiveness for the few instances was quite large. For example, multiples of 4.104, 3.717, and 3.848 were the maximum ratios in scenarios with 25 nodes. The data also suggests that the difference between energy savings across all scenarios, at a given percentile level and given number of nodes, seems to be modest. For instance, at the 50 percentile, the low and high values for 25n scenarios were 0.729 and 0.761, respectively. However, a significant difference does appear to exist between scenarios with different numbers of nodes. In Table 4.6, with some exceptions, overlap of confidence intervals does not occur for scenarios with equal area but different numbers of nodes. The notable exceptions occur in scenarios with the same area but with either 50 or 75 nodes. Randomness in the data and insufficient difference between the number of nodes is a likely cause of overlapping confidence intervals between two pairs of scenarios: (50n,25s), (75n,25s); and (50n,75s), (75n,75s), respectively. Apparently, energy savings seems to be significantly influenced by the number of nodes in a network, as opposed to transmission distance.

Table 4.7: Mean; Low & High Confidence Intervals for Energy Savings ($\alpha = 0.95$)

	25n,25s	25n,50s	25n,75s	50n,25s	50n,50s	50n,75s	75n,25s	75n,50s	75n,75s
Mean	0.828	0.830	0.805	0.671	0.746	0.735	0.651	0.650	0.677
upper 95% Mean	0.864	0.866	0.837	0.701	0.787	0.774	0.685	0.683	0.714
lower 95% Mean	0.793	0.794	0.773	0.641	0.705	0.695	0.617	0.617	0.640

However, an important fact to remember is that such savings require some sort of routing algorithm to ferry signals between only the nodes in the shortest path along the Delaunay triangulation. This study does not consider the use of routing algorithms because the appearance of a TST is considered to be worth the excess energy consumption.

4.4 Connectivity and Reduction of Nodes

In a hazardous environment, the probability of an ISR UAV getting eliminated from the constellation is likely and of great concern. If an original constellation of UAVs is configured with a given maximum transmission radius, then the reduction in nodes will impact the connectivity of the remaining nodes. Thus, the effect of lost nodes and its impact on SHARP's *path of information* should be tested.

An important parameter in such tests is the maximum transmission radius. Ideally, the maximum transmission radius of all nodes should be large enough so that communications can continue between all nodes with high probability. The 90-percentile transmission radius for each base case scenario was arbitrarily chosen. Connectivity was tested for each reduction (-5, -10, -15) of the starting node values (25, 50, 75), for each area of interest (25^2 km^2, 50^2 km^2, 75^2 km^2). Note that this study does not consider the possibility that aircraft can reorient themselves in order to maintain connectivity. An example of one sensitivity test for a base case is now presented for clarification.

For the base case of (25n, 25s), the 90-percentile transmission radius was 15 km long. Three sub-scenarios (see left-most column of Table 4.7) were tested for connectivity: (20n, 25s), (15n, 25s), and (10n, 25s). One hundred replications were run for each sub-scenario. The same procedure was repeated for the remaining base cases with 25 nodes and for all base cases with 50 nodes and 75 nodes. The values shown in Table 4.7 are the percentages of paths, under each sub-scenario, over which communication would not have occurred due to inadequate transmission radii.

Table 4.8: Sensitivity of UAV Configurations to Reduced Node Population

n	25s	50s	75s
20	10.7	14.3	12.3
15	15.7	21.0	15.3
10	17.0	16.7	26.3
45	9.3	10.0	13.7
40	7.7	10.7	11.0
35	9.0	10.7	9.7
70	9.7	9.0	7.3
65	7.3	7.3	12.0
60	8.0	14.3	11.0

n represents the sub-scenarios that were derived from the three base cases; the 90 percentile radius for each Base Case is used

The data in Table 4.7 does not seem to show any strong patterns. Several minor characteristics, however, stand out: 1) only three scenarios of all cases derived from 75 nodes had lost paths in excess of 10%; 2) all scenarios derived from 25 nodes had percentage of lost paths in double digits; 3) upon inspection, the base case of 50 nodes seems to perform just as well as the base case of 75 nodes. While the first two characteristics are not surprising, the third is somewhat unusual. Intuition says that the base case with more nodes should be more resistant to decrements. However, the apparent parity could be attributed to the fact that while an increase from 25 to 50 nodes is 100%, the increase from 50 to 75 nodes is only 50%, and the 50% increase was not enough to overcome the fact that scenarios for a given area of interest had virtually the same 90 percentile values for minimum transmission radius (see Table 4.2). More data should be collected before further conclusions are reached.

The most significant cause for such similarity between percentages of lost paths is probably the use of a 90-percentile transmission radius. The results of sensitivity tests clearly show the advantages of being capable of transmitting over longer distances. For example, considering that sub-scenario (10n,75s) lost well over half of the original constellation and was still able to communicate over almost 75% of the randomly chosen paths is impressive. This kind of robustness would be of great value in an environment where survivability of a given UAV is low.

5. Conclusions and Future Research

5.1 Conclusions

The TST problem is one that presents many challenges for military systems that are highly dependent on human control. This study proposes an alternative system (SHARP) that, in theory, enables autonomous, cooperative unmanned air vehicles (UAVs) to a) maintain their survivability, b) communicate securely and efficiently, c) react quickly to a changing priority of targets, and d) maintain connectivity in the network despite a depletion in the number of UAVs in the network. The network is used not only to relay communication between nodes, but also to navigate strike aircraft to TSTs, using UAVs along the *path of information* as waypoints.

Next, a mathematical model for SHARP was created in order to addresses several key issues related to the coordination of UAVs in a TST scenario. Specifically, the model builds upon Delaunay triangulations, which were generated for uniformly distributed nodes in predefined areas of interest, and collects data regarding 1) the ratio between the *direct path* and the SHARP path; 2) the distribution of the minimum transmission radius required to communicate between an arbitrary sender and receiver, along the shortest path of the Delaunay triangulation; 3) the difference of energy consumption, attributed to communication, between communicating along the *direct length* versus SHARP's *path of information*; and 4) the extent to which nodes, within a predefined space and with a given transmission radius, lose connectivity as the number of nodes decreases. Each of the four metrics should be considered for a configuration of UAVs under SHARP. The first is important because a SHARP path that is too much longer than the direct length may make travel along the SHARP path too much in a time-sensitive operation; furthermore, it ensures that the strike UAVs stay within an airspace that is suitably safe. The second is critical because it reveals the amount of transmission power that is required to ensure connectivity between arbitrary points in the network. More transmission power, in general, requires more infrastructure in UAVs, which influences factors such as performance and cost of UAVs. Additionally, multiple paths between two nodes in the constellation provide flexibility of communication where bandwidth may be a limitation. The third metric reveals the degree to which energy consumption due to communication is saved or wasted via SHARP. In light of a UAV's limited energy supply, less energy consumption could significantly prolong the effectiveness of a UAV sortie. Minimizing the amount of energy consumed is closely related to the idea of transmitting signals only as far as they need to be transmitted; minimizing the signal strength of communication between nodes reduces the probability that such communication will be collected and exploited by adversaries. The fourth and final metric is especially crucial to military operations since UAVs will likely be lost in a hazardous, combat environment and connectivity in the network should not be affected radically as the sparseness in a network increases.

Delaunay triangulations are used as a base for the model for several reasons. First, the shortest path through the triangulation is equivalent to the shortest path that would be taken by an electromagnetic signal that is propagated between two arbitrary nodes. Second, such triangulations are useful when solving problems that deal with connectivity and/or routing in a wireless network. Third, the shortest path between two nodes in a Delaunay triangulation has a proven upper bound; thus, the path created by the model will have an even tighter upper bound.

The model uses a uniform distribution because it captures the fact that detection of a TST requires, in general, relatively uniform spacing over an area of interest, with no prior knowledge of where TST may appear. UAVs are assumed to be capable of maintaining relatively stable locations above an area of interest. In order to remove any bias that may be inherent in a particular Delaunay triangulation, each repetition of the model constructs a new set of points. Collecting data from many Delaunay triangulations also leaves open the possibility of revealing any underlying structures that are present in the triangulations.

The output from the model suggests that the first metric is frequently optimal and that the SHARP path is, for most intents and purposes, equivalent to the direct length. However, the prerequisite for such small differences between the SHARP path and the direct path is a sufficiently long transmission distance. Data for the second metric suggests that short transmission distances are attainable, in general, at the cost of more nodes. Even when node density is increased such that the vast majority of transmission radii are in single digits, there still exist paths that require large transmission radii. While very long transmission radii are a problem given the limitations/assumptions made in this thesis, the problem would likely be remedied should a different path, using shorter node separation, be used – in lieu of the positive skewness in the distribution of the transmission radii, in general, there is good chance of this occurring. The third metric, too, can be steadily reduced as node density increases. Finally, with the exception of the markedly poor sensitivity results for scenarios starting with 25 nodes, none of the scenarios differed considerably in terms of the fourth metric. The lack of differences could be attributed to the fact that increasing node density to reduce transmission distance is not enough to offset the reduced robustness that arises by virtue of a shorter transmission distance.

The four aims (Section 1.1) were satisfied to the following degrees. First Point: by restricting strike UAVs to airspaces that are occupied by ISR UAVs, the survivability of UAVs is likely kept at an acceptable level. Second Point: communication within the ad-hoc network is conducted in a secure and efficient fashion, in the sense that 1) for a given area and number of nodes, the transmission distance can be capped at an arbitrary value and result in a certain level of connectivity and robustness, in the event of lost nodes; and 2) energy savings occurs in the vast majority of communication along the *path of information*. Bandwidth savings due to smart routing of communication was not addressed in this thesis, but should be addressed in future research. Third Point: the potential for quick reaction to dynamically generated targets has been shown in lieu of the small differences between the SHARP path and the *direct length*. Likewise, the SHARP path ensures constant communication with ISR UAVs, thereby making a timely reroute to a different TST possible. Fourth Point: SHARP has demonstrated an ability to maintain connectivity, even in the event of a severe reduction of UAVs in the network.

While the mathematical model of SHARP is currently very simple, it can give the user a better perspective on the challenges/benefits of such a system of autonomous machines. In lieu of the USAF's move toward UAVs, this model provides a handy way to grasp the number of UAVs that may be required to detect and engage all TSTs with a high probability given certain constraints on, say, the four metrics that were analyzed in this study. Subsequently, a better perspective on where a particular type of UAV will fit in the present and future concept of operations of the USAF can be gained. Moreover, if the sensor and flight performance capabilities, as well as operating costs, of different UAVs are added to the current model of SHARP, then people who are responsible for acquiring UAVs for the USAF will be in a better position to find optimum trade-offs between, say, survivability, maintenance costs, communication capability, sensor capability, sensitivity to attack, and transmission energy savings. A mature version of the model may give field operators the ability to diagnose, in very short order, the health (based on some metric such as probability of disconnect) of a particular network of UAVs and then recommend a course of action (such as how many UAVs must be added to the network in order to reduce the probability of disconnect below some threshold).

5.2 Future Research

Further research could improve this model's ability to 1) capture additional costs of SHARP; 2) model the dynamics of multiple strike aircraft attacking several TSTs over time; 3) quantify the effects of routing algorithms for communication and the effects that they would have on energy costs, and 4) improve methods to autonomously maintain connectivity among nodes when the ISR UAVs are allowed to move. Finally, 5) an automated procedure that generates an optimal UAV distribution based on a user defined partition of an area of interest would be of great value.

First, a number of parameters could be added to this model of SHARP in order to gauge the costs associated with a given number of nodes in a given area. For example, if each transmission by a node is assigned a cost, then the rate at which energy, for communication, within particular nodes decreases could be analyzed. Such information could be compared to the average flight endurance of nodes and the difference would shed light on excessive capability, with respect to flight or communication, in a SHARP network. Furthermore, costs related to maintenance of UAVs could be compared to costs to maintain a similarly capable manned-aircraft. Also, since some UAVs are very portable and do not require a runway, various aspects of distributing such portable UAVs across many ground units in the field could be compared to launching UAVs from a few fixed runways.

Second, during wartime operations, autonomous aircraft would likely operate in a very dense airspace that is extremely unpredictable. Developing a means of equipping aircraft with an internal logic that would enable them to avoid each other without serious degradation to each other's operations would be a step toward making such wartime operation a reality. Conceivably, a distributed simulation with a different thread for each aircraft could be developed to test such internal logic and measure the effect of such logic based on metrics such mean delay per route or number of collisions/near-collisions per thousand hours of flight time.

Third, investigating the effects of routing algorithms for communications would shed light on the relationship between transmission savings and computation directly related to routing. If computation is too intense, then the latency of the network may increase to excessive levels. On the other hand, if no form of routing is used, then nodes may be fitted with transmission capability that needlessly degrades flight performance.

Fourth, a real-world implementation of autonomous UAVs should allow for each aircraft to pursue its own objectives without being a detriment to the network at large.

This vision cannot be realized unless every movement of a UAV is in some way dependent on all other UAVs. Analyzing the effects of allowing nodes to temporarily break connectivity with the network could lead to more flexible constellations of UAVs and ultimately more adaptable behaviors.

Finally, as mentioned in Section 2.3, the ability to derive optimal node location based on a user-determined partition of an area of interest would be a nice option in the event that a TST planner has reason to give certain areas more attention than others. For example, assume that a TST planner decides that *n* regions (forming a partition of, say, a city) should be the focus of attention. Whether the partition that he/she defines forms a VD is of concern, since the dual graph of that would-be VD forms a DT, which has a number of desire able properties that were described in Section 2.3. This ability could take the form of a software application that a) enables its user to download map information for areas around the world and define partitions on that map via some graphical user interface; b) determines if the partition is a VD; c) if the partition is a VD, then it would return optimal node locations; if the partition was not a VD, then the computer would apply some heuristic that would create a VD while maintaining, to the greatest extent, the characteristics of the initial partition.

Appendix A

Proc1:

****

Coding by Harry Calkins and David Chow

Loading Packages

```
<<DiscreteMath`

<<Graphics`
```

The Auxiliary Functions

dist

This is a function that will calculate the distance between two points. Input form: {{w,x},{y,z}}. See testInputs.nb for the test inputs.

```
dist[{pt1_?VectorQ,pt2_?VectorQ}]:= Sqrt[(pt1 - pt2).(pt1 - pt2) ]
```

getLongestEdge

This function (getLongestEdge) will return the longest edge if given an input that is a list of lists, where each sublist is a pair of vertices in 2-space. Input should be in the form { {{a1,b1},{c1,d1}}, {{a2,b2},{c2,d2}},... }

```
getLongestEdge[ptsList_]:=
  Module[{lgthList},
    lgthList={};
    For[i=1,i\[LessEqual] Length[ptsList],i++,
        lgthList=Append[lgthList,dist[ptsList[[i]]]];
        ]
      lgthList;
    Max[lgthList]
```

```
    ]
```

DGraph

This function creates the Delaunay triangulation.

```
DGraph[pts_]:= Module[{locs,rools,plt,ptprs},
    locs = Range[Length[pts]];
    rools=Thread[Rule[pts,locs]];
    plt= PlanarGraphPlot[pts,DisplayFunction \[Rule] Identity];
    ptprs=First/@Cases[plt,_Line,Infinity]/.rools;
     FromUnorderedPairs[ptprs]
    ]
```

An Auxiliary Function

This function contains the logic for a strike aircraft changing its flight path when it enters the transmission radius of a neighboring node.

```
ff[{x1_,y1_},{x2_,y2_},rad_?NumericQ]:=
  If[Norm[({x1,y1}-{x2,y2})]>
      rad,{rad Cos[\[Theta]],rad Sin[\[Theta]]}+{x2,
          y2}/.FindRoot[{(1-t) x1+t x2\[Equal]
            x2+rad Cos[\[Theta]],(1-t) y1+t y2\[Equal]
            y2+rad Sin[\[Theta]]},{t,.9},{\[Theta],
          ArcTan@@({x1,y1}-{x2,y2})}],
    {x1,y1}
    ]
```

The TwistedPath2 function

This function calculates the SHARP path

```
TwistedPath2[pts_?MatrixQ,rad_?NumericQ]:=
  FoldList[ff[#1,#2,rad]&,pts[[1]],Most@Rest[pts]]
```

PathLengths

This function calculates the path length that is determined by a series of vertices. Input should be in the form {{x1,y1},{x2,y2},...}

```
PathLengths[ptlis_?MatrixQ]:=
  Map[Norm[#,2]&,(Partition[ptlis,2,1]/.{a_,b_}\[RuleDelayed] b-a)]
```

ShowPath2

This function enables a visual representation of the key elements of SHARP: transmission radius, nodes in the shortest path, and SHARP path.

```
ShowPath2[ptlis_?MatrixQ,solis_?MatrixQ,rad_,opts___?OptionQ]:=
  Show[Graphics[{Line[ptlis],Circle[#,rad]&/@Rest[ptlis],Red,
      Line[Append[solis,Last[ptlis]]],Blue,AbsolutePointSize[6],
      Point/@ptlis}],{opts}]
```

Line Crossings

This gives 1 if the paths cross and 0 if they do not, taking the segments one pair at a time.

```
LineCross[v1_?VectorQ, v2_?VectorQ, v3_?VectorQ, v4_? VectorQ] :=
  Module[{t, r,
     val}, {t, r} = {t, r} /.
      FindRoot[(1 - t)v1 + t v2 == (1 - r) v3 +
          r v4, {t, .3, .6}, {r, .3, .6}];
    If[0 < t < 1 && 0 < r < 1, 1, 0] ]
```

Simulation Function

This function is set up to generate n vertices where each of the coordinates lies between 0 and 25. That can be changed by changing the parameters in the Random functions in the verts line.

If you do not want this function to generate the vertices and instead have a list of vertex sets then you will need to make a minor modification to this function to take the vertex list as a first argument and then use the Map function to get things to work.

"n" is the number of nodes in an area of interest; "pt1" and "pt2" are arbitrary indeces among the "n" nodes (note: neither pt1 nor pt2 can be an integer greater than n).

With a plot

```
dataList[n_Integer,{pt1_Integer,pt2_Integer}]:=
 Module[{vrts,gr1,shrtpth,shrtpth2,edgcnt,ptlis,regpthln,shrtpthln,
   crossCount,directLength,getTheLines,getTheLineEnds,theLongestEdge,
   theLongEdgeInShrtpth, edgesInRegPth,regpthEnergy,directLengthEnergy,
   overheadCost,pathLoss},
  verts = Table[{Random[Real,{0,25}],Random[Real,{0,25}]},{n}];
  getTheLines=PlanarGraphPlot[verts,DisplayFunction \[Rule] Identity];
  getTheLineEnds=First/@Cases[getTheLines,_Line,Infinity];
  theLongestEdge=getLongestEdge[getTheLineEnds];(*longest edge in DGraph*)
  directLength=dist[verts[[{pt1,pt2}]]];
  gr1= DGraph[verts];
  shrtpth= ShortestPath[gr1,pt1,pt2];
  edgcnt = Length[shrtpth]-1;
  ptlis = verts[[shrtpth]];
  theLongEdgeInShrtpth=Max[PathLengths[ptlis]];
  shrtpth2 = TwistedPath2[ptlis,theLongEdgeInShrtpth];
  regpthln =Plus@@ PathLengths[ptlis];
  shrtpthln =Plus@@ PathLengths[Join[shrtpth2,{ptlis[[-1]]}]];
  edgesInRegpth=PathLengths[ptlis];
  overheadCost=2.3;
  pathLoss=2;
  regpthEnergy=0;
  For[i=1,i\[LessEqual]Length[edgesInRegpth],i++,
   regpthEnergy=regpthEnergy+overheadCost*(edgesInRegpth[[i]]^pathLoss
       )];
  directLengthEnergy=overheadCost*(directLength^pathLoss);
  crossCount=
   Count[Apply[LineCross,
     Flatten[#,1]&/@
      Thread[{Partition[Take[ptlis,{2,-2}],2,1],
        Partition[Take[Append[shrtpth2,ptlis[[-1]]],{3,-1}],2,
```

```
            1]}],{1}],1];
   ShowPath2[ptlis,shrtpth2,theLongEdgeInShrtpth,
     AspectRatio \[Rule] Automatic,Axes\[Rule] True,ImageSize \[Rule] 4 72];
   {edgcnt,crossCount,regpthln,shrtpthln,shrtpthln/directLength,
     directLength, theLongestEdge,theLongEdgeInShrtpth,
     regpthEnergy/directLengthEnergy}
   ]
```

Without a plot

The only difference between this function and dataList is that this function does not display any graphs.

```
dataList2[n_Integer,{pt1_Integer,pt2_Integer}]:=
  Module[{vrts,gr1,shrtpth,shrtpth2,edgcnt,ptlis,regpthln,shrtpthln,
     crossCount,directLength,getTheLines,getTheLineEnds,theLongestEdge,
     theLongEdgeInShrtpth, edgesInRegPth,regpthEnergy,directLengthEnergy,
     overheadCost,pathLoss},
   verts = Table[{Random[Real,{0,25}],Random[Real,{0,25}]},{n}];
   getTheLines=PlanarGraphPlot[verts,DisplayFunction \[Rule] Identity];
   getTheLineEnds=First/@Cases[getTheLines,_Line,Infinity];
   theLongestEdge=getLongestEdge[getTheLineEnds];(*longest edge in DGraph*)
   directLength=dist[verts[[{pt1,pt2}]]];
   gr1= DGraph[verts];
   shrtpth= ShortestPath[gr1,pt1,pt2];
   edgcnt = Length[shrtpth]-1;
   ptlis = verts[[shrtpth]];
   theLongEdgeInShrtpth=Max[PathLengths[ptlis]];
   shrtpth2 = TwistedPath2[ptlis,theLongEdgeInShrtpth];
   regpthln =Plus@@ PathLengths[ptlis];
   shrtpthln =Plus@@ PathLengths[Join[shrtpth2,{ptlis[[-1]]}]];
   edgesInRegpth=PathLengths[ptlis];
   overheadCost=2.3;
   pathLoss=2;
   regpthEnergy=0;
   For[i=1,i\[LessEqual]Length[edgesInRegpth],i++,
     regpthEnergy=regpthEnergy+overheadCost*(edgesInRegpth[[i]]^pathLoss
          )];
   directLengthEnergy=overheadCost*(directLength^pathLoss);
   crossCount=
     Count[Apply[LineCross,
         Flatten[#,1]&/@
           Thread[{Partition[Take[ptlis,{2,-2}],2,1],
```

```
        Partition[Take[Append[shrtpth2,ptlis[[-1]]],{3,-1}],2,
         1]}],{1}],1];
  {edgcnt,crossCount,regpthln,shrtpthln,shrtpthln/directLength,
   directLength, theLongestEdge,theLongEdgeInShrtpth,
   regpthEnergy/directLengthEnergy}
  ]
```

Example

This important thing to keep in mind when looking at the plots here is the path starts at the point which does not have a circle around it.

In [dataList[a,{b,c}], {d}], "a" denotes number of nodes, "b" and "c" are arbitrary nodes selected from the "a" specified nodes, and "d" is the number of iterations.

```
TableForm[Simdata =Table[dataList[25,{5,23}],{25}],
 TableHeadings \[Rule] {None,{"Edges","Crosses","Path Length 1",
    "Path Length 2","Advantage","DirectLength","theLongestEdge",
    "theLongEdgeInShrtpth","commEnergySaved"}}]
Timing[Simdata2 =Table[dataList2[25,{5,23}],{500}];]
```

Exporting the Data

This function will export the results of the simulation to the default folder of the user.

```
Export["Step1_25n_25s_2.txt",Simdata2,"Table",
 ConversionOptions\[Rule]{"FormatType"\[Rule](NumberForm[#,{12,10},
  NumberPadding\[Rule]{" ","0"}]&)}]
```

Proc2:

****

Coding by Harry Calkins and David Chow

Loading Packages

```
<<DiscreteMath`

<<Graphics`
```

The Auxiliary Functions

dist

This is a function that will calculate the distance between two points. Input form: {{w,x},{y,z}}. See testInputs.nb for the test inputs.

```
dist[{pt1_?VectorQ,pt2_?VectorQ}]:= Sqrt[(pt1 - pt2).(pt1 - pt2) ]
```

getLongestEdge

This function (getLongestEdge) will return the longest edge if given an input that is a list of lists, where each sublist is a pair of vertices in 2-space. Input should be in the form { {{a1,b1},{c1,d1}}, {{a2,b2},{c2,d2}},... }

```
getLongestEdge[ptsList_]:=
 Module[{lgthList},
  lgthList={};
  For[i=1,i\[LessEqual] Length[ptsList],i++,
     lgthList=Append[lgthList,dist[ptsList[[i]]]];
     ]
   lgthList;
  Max[lgthList]
  ]
```

DGraph

This function creates the Delaunay triangulation.

```
DGraph[pts_]:= Module[{locs,rools,plt,ptprs},
  locs = Range[Length[pts]];
  rools=Thread[Rule[pts,locs]];
```

```
 plt= PlanarGraphPlot[pts,DisplayFunction \[Rule] Identity];
 ptprs=First/@Cases[plt,_Line,Infinity]/.rools;
  FromUnorderedPairs[ptprs]
 ]
```

An Auxiliary Function

This function contains the logic for a strike aircraft changing its flight path when it enters the transmission radius of a neighboring node.

```
ff[{x1_,y1_},{x2_,y2_},rad_?NumericQ]:=
 If[Norm[({x1,y1}-{x2,y2})]>
    rad,{rad Cos[\[Theta]],rad Sin[\[Theta]]}+{x2,
       y2}/.FindRoot[{(1-t) x1+t x2\[Equal]
         x2+rad Cos[\[Theta]],(1-t) y1+t y2\[Equal]
         y2+rad Sin[\[Theta]]},{t,.9},{\[Theta],
       ArcTan@@({x1,y1}-{x2,y2})}],
  {x1,y1}
  ]
```

The TwistedPath2 function

This function calculates the SHARP path

```
TwistedPath2[pts_?MatrixQ,rad_?NumericQ]:=
 FoldList[ff[#1,#2,rad]&,pts[[1]],Most@Rest[pts]]
```

PathLengths

This function calculates the path length that is determined by a series of vertices. Input should be in the form {{x1,y1},{x2,y2},...}

```
PathLengths[ptlis_?MatrixQ]:=
 Map[Norm[#,2]&,(Partition[ptlis,2,1]/.{a_,b_}\[RuleDelayed] b-a)]
```

ShowPath2

This function enables a visual representation of the key elements of SHARP: transmission radius, nodes in the shortest path, and SHARP path.

```
ShowPath2[ptlis_?MatrixQ,solis_?MatrixQ,rad_,opts___?OptionQ]:=
  Show[Graphics[{Line[ptlis],Circle[#,rad]&/@Rest[ptlis],Red,
        Line[Append[solis,Last[ptlis]]],Blue,AbsolutePointSize[6],
        Point/@ptlis}],{opts}]
```

Line Crossings

This gives 1 if the paths cross and 0 if they do not, taking the segments one pair at a time.

```
LineCross[v1_?VectorQ, v2_?VectorQ, v3_?VectorQ, v4_? VectorQ] :=
  Module[{t, r,
      val}, {t, r} = {t, r} /.
        FindRoot[(1 - t)v1 + t v2 == (1 - r) v3 +
              r v4, {t, .3, .6}, {r, .3, .6}];
    If[0 < t < 1 && 0 < r < 1, 1, 0] ]
```

Simulation Function

This function is set up to generate n vertices where each of the coordinates lies between 0 and 25. That can be changed by changing the parameters in the Random functions in the verts line.

If you do not want this function to generate the vertices and instead have a list of vertex sets then you will need to make a minor modification to this function to take the vertex list as a first argument and then use the Map function to get things to work.

"n" is the number of nodes in an area of interest; "pt1" and "pt2" are arbitrary indeces among the "n" nodes (note: neither pt1 nor pt2 can be an integer greater than n).

With a plot

```
dataList[n_Integer,{pt1_Integer,pt2_Integer},rad_?NumericQ]:=
  Module[{vrts,gr1,shrtpth,shrtpth2,edgcnt,ptlis,regpthln,shrtpthln,
      crossCount,directLength,getTheLines,getTheLineEnds,theLongestEdge,
      theLongestEdgeInShrtpth},
    verts = Table[{Random[Real,{0,25}],Random[Real,{0,25}]},{n}];
```

```
  getTheLines=PlanarGraphPlot[verts,DisplayFunction \[Rule] Identity];
  getTheLineEnds=First/@Cases[getTheLines,_Line,Infinity];
  theLongestEdge=getLongestEdge[getTheLineEnds];
  directLength=dist[verts[[{pt1,pt2}]]];
  gr1= DGraph[verts];
  shrtpth= ShortestPath[gr1,pt1,pt2];
  edgcnt = Length[shrtpth]-1;
  ptlis = verts[[shrtpth]];
  theLongestEdgeInShrtpth=Max[PathLengths[ptlis]];
  shrtpth2 = TwistedPath2[ptlis,rad];
  regpthln =Plus@@ PathLengths[ptlis];
  shrtpthln =Plus@@ PathLengths[Join[shrtpth2,{ptlis[[-1]]}]];
  crossCount=
   Count[Apply[LineCross,
     Flatten[#,1]&/@
      Thread[{Partition[Take[ptlis,{2,-2}],2,1],
        Partition[Take[Append[shrtpth2,ptlis[[-1]]],{3,-1}],2,
         1]}],{1}],1];
  ShowPath2[ptlis,shrtpth2,rad,AspectRatio \[Rule] Automatic,
   Axes\[Rule] True,ImageSize \[Rule] 4 72];
  regpthEdgeLengths=PathLengths[ptlis];
  For[i=1,i\[LessEqual]Length[regpthEdgeLengths],
   If[regpthEdgeLengths[[i]]>rad,(numberOutOfBounds=numberOutOfBounds+1;
     Break[])];
   i++];
  {edgcnt,crossCount,regpthln,shrtpthln,shrtpthln/directLength,
   directLength, theLongestEdge, theLongestEdgeInShrtpth}
  ]
```

Without a plot

The only difference between this function and dataList is that this function does not display any graphs.

```
dataList2[n_Integer,{pt1_Integer,pt2_Integer},rad_?NumericQ]:=
 Module[{vrts,gr1,shrtpth,shrtpth2,edgcnt,ptlis,regpthln,shrtpthln,
   crossCount,directLength,getTheLines,getTheLineEnds,theLongestEdge,
   theLongestEdgeInShrtpth},
  verts = Table[{Random[Real,{0,25}],Random[Real,{0,25}]},{n}];
  getTheLines=PlanarGraphPlot[verts,DisplayFunction \[Rule] Identity];
  getTheLineEnds=First/@Cases[getTheLines,_Line,Infinity];
  theLongestEdge=getLongestEdge[getTheLineEnds];
```

```
directLength=dist[verts[[{pt1,pt2}]]];
gr1= DGraph[verts];
shrtpth= ShortestPath[gr1,pt1,pt2];
edgcnt = Length[shrtpth]-1;
ptlis = verts[[shrtpth]];
theLongestEdgeInShrtpth=Max[PathLengths[ptlis]];
shrtpth2 = TwistedPath2[ptlis,rad];
regpthln =Plus@@ PathLengths[ptlis];
shrtpthln =Plus@@ PathLengths[Join[shrtpth2,{ptlis[[-1]]}]];
crossCount=
  Count[Apply[LineCross,
      Flatten[#,1]&/@
        Thread[{Partition[Take[ptlis,{2,-2}],2,1],
            Partition[Take[Append[shrtpth2,ptlis[[-1]]],{3,-1}],2,
              1]}],{1}],1];
regpthEdgeLengths=PathLengths[ptlis];
For[i=1,i\[LessEqual]Length[regpthEdgeLengths],
  If[regpthEdgeLengths[[i]]>rad,(numberOutOfBounds=numberOutOfBounds+1;
      Break[])];
  i++];
{edgcnt,crossCount,regpthln,shrtpthln,shrtpthln/directLength,
  directLength, theLongestEdge, theLongestEdgeInShrtpth}
]
```

Example

This important thing to keep in mind when looking at the plots here is the path starts at the point which does not have a circle around it.

In dataList[a,{b,c}, d], {e}], "a" denotes number of nodes, "b" and "c" are arbitrary nodes selected from the "a" specified nodes, "d" is the radius, and "e" is the number of iterations.

```
TableForm[Simdata =Table[dataList[15,{5,10},22],{20}],
  TableHeadings \[Rule] {None,{"Edges","Crosses","Path Length 1",
      "Path Length 2","Advantage","DirectLength","theLongestEdge",
      "theLongestEdgeInShrtpth"}}]
Print[numberOutOfBounds," paths had inadequate transmission radii."]
numberOutOfBounds=0;
Timing[Simdata2 =Table[dataList2[60,{5,10},15],{100}];]
Print[numberOutOfBounds," paths had inadequate transmission radii."]
```

SessionTime[]

Exporting the Data

This function will export the results of the simulation to the default folder of the user.

```
Export["Step2_60n_25s.txt",Simdata2,"Table",
  ConversionOptions\[Rule]{"FormatType"\[Rule](NumberForm[#,{12,10},
       NumberPadding\[Rule]{" ","0"}]&)}]
```

Acronyms/Keywords

AOC	Air Operations Center
Autonomy	See Section 1.4
Coordination variable/function	See Section 2.4.3
DARPA	Defense Advanced Research Projects Agency
Direct length	The Euclidean distance between the ISR UAV closest to a TST and the nearest strike UAV
G-force	Gravity force
ISR	Intelligence, Surveillance, Reconnaissance
MILP	Mixed integer linear programming
Off-line	Conducted before operations; not real-time
online	Conducted in real-time
Path of information	The shortest path, on the Delaunay triangulation, between the ISR UAV that is closest to a TST and the nearest strike UAV
Persistence	See Section 1.2
Range	See Section 1.5
RF	Radio frequency
RH	Receding-horizon
SAM	Surface to Air missile
Situational Awareness	See Section 1.6
SHARP	Short-hop abbreviated routing paradigm
TST	Time-sensitive Targetting/Targets
TT&P	Tactics, techniques and procedures
UAV	Unmanned Air Vehicle
Volume of Influence	See Section 1.3
Waypoint	A position in space, on an aircraft's flight plan

Bibliography

[1] Aichholzer, Oswin, and Franz Aurenhammer. "Voronoi Diagrams - Computational Geometry's Favorite." http://www.igi.tugraz.at/telematik/tele1-02_aich-favor.pdf . 2002. Institute for Theoretical Computer Science. 6 Oct 2004.

[2] Air Land and Sea Application Center. "Multi-Service Tactics Techniques and Procedures for Targeting Time Sensitive Targets." April 2004. Air Land and Sea Application Center. 4 Oct 2004.

[3] Aurenhammer, Franz, Rolf Klein. "Voronoi Diagrams." *Handbook of Computational Geometry*. Elsevier Science Publishing, 2000*: 201-290.*

[4] Barlow, Gregory J. "Autonomous Controller Design for Unmanned Aerial Vehicles using Multi-objective Genetic Programming." Master's thesis. North Carolina State University. Raleigh, NC. March 2004.

[5] Best, Richard A. "Intelligence, Surveillance, and Reconnaissance (ISR) Programs: Issues for Congress." Federation of American Scientists. 4 Aug 2004. Congressional Research Service. 1 Oct 2004. http://www.fas.org/irp/crs/RL32508.pdf

[6] Bigham, John, et al. "Using intelligent agents for managing resources in military communications." Computer Networks. Vol 46, Issue 5. Dec 2004: 709-741

[7] Bose, Prosenjit, and Pat Morin. "Online Routing In Triangulations." *Siam Journal of Computing, 33(4), 937-951, 2004.*

[8] Brown, Timothy, et al. "Test Bed for a Wireless Network on Small UAVs." University of Colorado. 2004. Accessed: 15 NOV 2004.

[9] Buckmaster, Gerald. Gerald.Buckmaster@dm.af.mil. Jan 5, 2005.

[10] *Chatila, Raja, Simon Lacroix, Stephane Betge-Brezetz, Michel Devy, and Thierry Simeon. "Autonomous Mobile Robot Navigation for Planet Exploration: The EDEN Project." http://citeseer.ist.psu.edu/cache/papers/cs/1268/http:zSzzSzwww.cs.cmu.eduzSzafszSzcs.cmu.eduzSzuserzSzepkzSzwwwzSzworkshopzSzchatila.pdf/chatila96autonomous.pdf . 1996.* Laboratory for Analysis and Architecture of Systems. 13 Oct 2004.

[11] Choudhury, Romit Roy, and Nitin H. Vaidya. "Performance of ad hoc routing using directional antennas." Sep 2004.

[12] Dobkin, David, et al. "Delaunay Graphs Are Almost As Good As Complete Graphs." Discrete and Computational Geometry, Vol 5, Issue 4 (July 1990): 399-407.

[13] Feller, S.D., et al. "Tracking and imaging humans on heterogeneous infrared sensor arrays for law enforcement applications." http://www.disp.duke.edu/~sfeller/Publications/Aerosense2002_Law.pdf . Duke Integrated Sensing and Processing. April 2002. Duke University. 4 Oct 2004.

[14] Frodigh, Magnus, Per Johansson, and Peter Larsson. "Wireless Ad-Hoc Networking – The Art of Networking Without A Network." Ericsson Review, No. 4, 2000.

[15] Garamone, Jim. "From U.S. Civil War to Afghanistan: A Short History of UAVs." DefenseLINK News. 16 April 2002. American Forces Information Service. 2 December 2004. <www.defenselink.mil>

[16] Hebert, Adam J. "Compressing The Kill-Chain." Air Force Magazine. March 2003. Air Force Association. 1 Oct 2004. http://www.afa.org/magazine/march2003/0303killchain.asp

[17] HQ Air Force Doctrine Center. "Intelligence, Surveillance, and Reconnaissance Operations." http://www.dtic.mil/doctrine/jel/service_pubs/afd2_5_2.pdf . Defense Technical Information Center. 21 Apr 1999. Defense Technical Information Center. 3 Oct 2004.

[18] Jackson, Sarah E. “Planning Coverage Of Points Of Interest Via Multiple Imaging Surveillance Assets: A Multi-Modal Approach.” Air University Research Web. March 2003. Maxwell Air Force Base. 29 Sep 2004.

[19] Kaufman, Gail. “Lessons Learned in Afghanistan will help improve UAVs.” Defense News Media Group. September 2002. Defense News Media Group. 2 December 2004. < http://www.defensenews.com/conferences/0902isr/isr_1320456.html >

[20] King, Ellis, et al. “Coordination and Control Experiments on a Multi-vehicle Testbed.” http://hohmann.mit.edu/papers/0949_FrP05.1.pdf. Sep 2004. Massachusetts Institute of Technology. 29 Sep 2004.

[21] Kinney, Gary W. “A Hybrid Jump Search And Tabu Search Metaheuristic For The Unmanned Aerial Vehicle Routing Problem.” Storming Media: Pentagon Reports and Documents. March 2000. Air Force Institute of Technology. 29 Sep 2004.

[22] Kumar, Rajesh. “Tactical Reconnaissance: UAVs Versus Manned Aircraft.” Air Command and Staff College, March 1997. Maxwell Air Force Base. Jan 2005

[23] Lemaire, Thomas, et al. “A Distributed Tasks Allocation Scheme in Multi-UAV Context.” http://www.laas.fr/~tlemaire/publications/lemaireICRA2004.pdf . Sep 2004. Laboratory for Analysis and Architecture of Systems. 29 Sep 2004.

[24] Li, Cuihong, Katia Sycara. “A Stable And Efficient Scheme For Task Allocation Via Agent Coalition Formation.” Intelligent Software Agents Lab. 23 Feb 2004. The Robotics Institute, Carnegie Mellon University. 29 Sep 2004.

[25] Li, Xiang-Yang, Yu Wang. “Efficient Localized Routing for Wireless Ad-Hoc Networks.” 23rd International Conference on Distributed Computing Systems Workshops. May 19 - 22, 2003.

[26] Li, Xiang-Yang, Peng-Jun Wan, Yu Wang, and Ophir Frieder. “Sparse Power Efficient Topology for Wireless Networks.” 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9. January 7-10, 2002.

[27] Lihui Gu, Daniel, Guangyu Pei, Henry Ly, Mario Gerla, Beichuan Zhang, Xiaoyan Hong, "UAV aided intelligent routing for ad-hoc wireless network in single-area theater", WCNC 2000 - IEEE Wireless Communications and Networking Conference, no. 1, September 2000 pp. 1220-1225.

[28] Marzolf , Gregory S. “Time Critical Targeting. Predictive Versus Reactionary Methods: An Analysis For the Future.” Air University Press . March 2004. Maxwell Air Force Base. 29 Sep 2004. http://aupress.au.af.mil/SAAS_Theses/Marzolf/Marzolf.pdf

[29] McLain, Timothy and Randal W. Beard. “Cooperative Path Planning for Time Critical Missions.” Proceedings of the American Control Conference, June 4-6, 2003.

[30] Meguerdichian, Seaphan, et al. “Coverage Problems in Wireless Ad-hoc Sensor Networks.” UCLA Gateway. April 2001. University of California, Los Angeles. 7 Nov 2004. http://www.cs.ucla.edu/~miodrag/papers/Meguerdichian_Infocom_01.pdf.

[31] Möhring, Rolf H., et al. “Solving Project Scheduling Problems by Minimum Cut Computations.” Massachusetts Institute of Technology Sloan School, Management Science Area. 29 Sep 2004. http://web.mit.edu/sloan-msa/Papers/1.9.pdf

[32] Nemeth, Gabor, Zolton Richard Turanyi, Andras Valko. “Throughput of Ideally Routed Wireless Ad Hoc Networks.” ACM SIGMOBILE Mobile Computing and Communications Review, Vol 5, Iss 4. (October 2001).

[33] Okabe A., B. Boots, K. Sugihara. Spatial Tessellations. West Sussex, England: John Wiley & Sons Ltd., 1992.

[34] Space Daily “First Production Model Fire Scout UAV Rolls Out.” http://www.spacedaily.com/news/uav-03c.html . 20 Jan 2003. Accessed: 9 Oct 2004.

[35] Schouwenaars, Tom, Eric Feron, and Jonathan How. “Safe Receding Horizon Path Planning for Autonomous Vehicles.” 40th Allerton Conference on Communication, Control, and Computation. October 2002.

[36] US Joint Forces Command. “Doctrine For Joint Urban Operations.” http://www.dtic.mil/doctrine/jel/new_pubs/jp3_06.pdf. 16 Sep 2002. US Joint Forces Command. 1 Oct 2004.

[37] West, Douglas B. Introduction to Graph Theory. New Jersey: Prentice Hall, 2001.

[38] Xu, Kaixin, and Xiaoyan Hong, Mario Gerla, Henry Ly, Daniel Lihui Gu, "Landmark routing in large wireless battlefield networks using UAVs", MILCOM 2001 - IEEE Military Communications Conference, no. 1, October 2001 pp. 230-234

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
11-05-2005	**Master's of Science Thesis**	Aug 2004 – Mar 2005

4. TITLE AND SUBTITLE	
ANALYSIS OF ONLINE-DELAUNAY NAVIGATION FOR TIME SENSITIVE TARGETING	**5a. CONTRACT NUMBER** NA
	5b. GRANT NUMBER NA
	5c. PROGRAM ELEMENT NUMBER NA
6. AUTHOR(S)	**5d. PROJECT NUMBER** NA
Chow, David., Captain, USAF	**5e. TASK NUMBER** NA
	5f. WORK UNIT NUMBER NA

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Street, Building 642 WPAFB OH 45433-7765	AFIT/GOR/ENS/05-03
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) NA	**10. SPONSOR/MONITOR'S ACRONYM(S)** NA
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) NA

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
NA

14. ABSTRACT

Given the drawbacks of leaving time-sensitive targeting (TST) strictly to humans, there is value to the investigation of alternative approaches to TST operations that employ autonomous systems. This paper accomplishes five things. First, it proposes a short-hop abbreviated routing paradigm (SHARP) – based on Delaunay triangulations (DT), ad-hoc communication, and autonomous control – for recognizing and engaging TSTs that, in theory, will improve upon persistence, the volume of influence, autonomy, range, and situational awareness. Second, it analyzes the minimum timeframe need by a strike (weapons enabled) aircraft to navigate to the location of a TST under SHARP. Third, it shows the distribution of the transmission radius required to communicate between an arbitrary sender and receiver. Fourth, it analyzes the extent to which connectivity, among nodes with constant communication range, decreases as the number of nodes decreases. Fifth, it shows the how SHARP reduces the amount of energy required to communicate between two nodes. Mathematica 5.0.1.0 is used to generate data for all metrics. JMP 5.0.1.2 is used to analyze the statistical nature of Mathematica's output.

15. SUBJECT TERMS

Delaunay Triangulation, online navigation, time sensitive targeting, unmanned air vehicles, Voronoi Diagram

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					David Chow, Capt, USAF (ENS)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	78	(937) 255-3636; e-mail: david.chow@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18