JOURNAL OF UNMANNED AERIAL SYSTEMS

PEER-REVIEWED PAPER

HEURISTIC NEAR-OPTIMAL UAS PATH PLANNING ALGORITHM FOR CONVOY OVERWATCH

Riley A. Livermore
First Lieutenant, USAF, Engineer, AFRL, Autonomous Control Branch

Garrison J. Lindholm
Captain, USAF, Engineer, AFRL, Autonomous Control Branch

Charles J. Neal
Captain, USAF, Test Director, AFNWC, Futures Division

Richard G. Cobb
Professor, Department of Aeronautics and Astronautics

John M. Colombi
Associate Professor, Department of Systems Engineering and Management

Abstract

The optimal path to fly a small unmanned aerial system (SUAS) for convoy overwatch was calculated, heuristically approximated onboard a SUAS autopilot, and demonstrated with hardware in-the-loop simulations and flight test. The optimal path minimized a cost functional consisting of the SUAS's control effort and deviation from a desired slant range. Due to several hardware and software limitations, the SUAS autopilot was incapable of implementing the optimal controller onboard. This paper introduces a novel heuristic-based algorithm developed in three steps and implemented onboard the autopilot to approximate the optimal solution. The first step manipulated the autopilot loiter logic to allow target tracking. The second step identified three parameters in the loiter logic that were tuned using a Design of Experiments (DOE) methodology. Lastly, a finite state machine (FSM) was created based on the DOE results to further optimize the real-world convoy overwatch algorithm. Each step was tested to evaluate how well it approximated the optimal path. The final algorithm using the FSM exhibited a 65% improvement in tracking performance and demonstrated an implementable, near-optimal convoy overwatch algorithm.

Heuristic Near-Optimal UAS Path Planning Algorithm for Convoy Overwatch Nomenclature

α = Cost functional weight factor
ϕ, θ, ψ = Roll, Pitch, and Yaw angles, deg
$\dot{\phi}$= Roll rate, deg/s
ψw = Wind direction, deg
f = Update frequency, Hz
g = Gravitational acceleration, m/s^2
h = SUAS altitude (AGL), m
J = Cost function value
J_{th} = Cost function threshold
SR = Slant range, m
SR_{D} = Slant range desired, m
t_0 = Initial time, sec
t_f = Final time, sec
t_{look} = Look-ahead time, sec
V = SUAS airspeed, m/s
V_{G} = SUAS ground speed, m/s
V_w = Wind speed, m/s
X_{GV}, Y_{GV} = Ground vehicle components in inertial frame
X_{SUAS}, Y_{SUAS} = SUAS components in the North, East, Down (NED) inertial frame

Introduction

Unmanned aerial systems (UASs) constitute a rapidly developing platform of American military air power. Since 2002, the number of operational UAS has increased over 40 fold and now UAS comprise an astounding 41% of the military's aircraft inventory (Gertler, 2012). Currently, both manned and unmanned platforms are used to provide real-time, video surveillance of friendly vehicle convoys in wartime environments. The Air Force Research Lab's (AFRL) goal is to demonstrate autonomous convoy overwatch using small unmanned aerial systems (SUASs) that can stream continuous, full motion video to the convoy commander without any direct input from the ground station operator. Using autonomous SUAS platforms for convoy surveillance allows for improved allocation of personnel and resources, saving the Air Force time, money, and lives. In addition, the technology has applications to search and rescue operations, armored car overwatch, wildlife tracking and monitoring, and border patrol applications.

Accomplishing the feat of autonomous target tracking with a SUAS is not without its challenges. The specific surveillance requirements for successfully performing convoy overwatch, coupled with the limited processing power of the onboard autopilot and power supply of the SUAS provide the greatest obstacles. To best achieve autonomous target tracking, first the optimal flight path for the SUAS must be determined. The subsequent challenge is to fly the optimal path onboard a real SUAS. If the optimal path cannot be determined and flown in real time, then a heuristic-based approximation of the optimal path should be demonstrated and compared.

A variety of research has been done using optimal control to develop flight paths for unmanned tasks. Smith, Cobb, Pierce, and Raska (2013) used optimal control to generate paths for UAS collision avoidance. Kim, Oh, and Tsourdos (2013) used optimal control to develop a nonlinear model-predictive coordinated standoff tracking of a moving target. Their optimal path formulation was compared with the Lyapunov vector field approach presented by Frew, Lawrence, and Morris (2008). The optimal solutions more accurately maintained the desired standoff distance from the target and kept the desired separation between the SUASs. Geiger et al. (2008) incorporated optimal control for a SUAS to track a moving ground vehicle. Their research used a fixed orientation camera and their cost functional maximize the SUAS time on target for simplified ground vehicle paths. Jodeh, Coon, Masternak, Cobb, and Agte (2014) used optimal control to plan a path for maximizing the ground coverage of unattended ground sensors. Prévost, Théiault, Desbiens, Poulini, and Gagnon (2009) developed an extended Kalman filter to predict the future path of a moving target and generated an optimal path based on that information. All of these optimal path

formulations required a cost functional, optimizer, and future knowledge of some kind. The processing time required to determine the optimal solutions often exceeds the mission time, therefore disqualifying them from real-time applications.

There have been numerous research efforts accomplished that use unmanned platforms to track moving targets that are applicable in a real-world environment. Rysdyk (2006) developed a path planning algorithm used for target observation in the presence of wind. His guidance law was developed using "good helmsman" techniques, which output a relative course heading as a function of cross-track error. Yoon, Park, and Kim (2012) developed a guidance law that used multiple SUAS to perform coordinated standoff tracking. Similarly, work done by Frew et al. (2008) and Lawrence, Frew, and Pisano (2008) performed target tracking by creating globally stable, Lyapunov vector fields that allowed a SUAS to autonomously and efficiently track a moving target. Quigley, Goodrich, Griffiths, Eldridge, and Beard (2005) improved the Lyapunov vector field approach through introducing a Hopf bifurcation method. This specific type of limit cycle has a faster convergence rate to the circular orbit than the Lyapunov approach and was implementable real-time onboard the SUAS. Each one of these methods represent a sub-optimal, target tracking algorithm that were efficient enough to execute real-time onboard the SUAS autopilot.

The two objectives of this research were: develop an optimal path planning algorithm and demonstrate a real-world execution of it for autonomous convoy overwatch. An optimal flight path algorithm is presented that calculates the necessary SUAS control inputs for any given ground vehicle path. More specifically, the algorithm determined the minimum control inputs required for a SUAS to maintain a desired slant range from the ground vehicle. This algorithm used future knowledge of the exact ground vehicle location to calculate the optimal path. The path planning capability was demonstrated through computer simulations using time-stamped, GPS locations of a ground vehicle driving along a path. Secondly, the optimal path algorithm was approximated onboard the SUAS, demonstrating a real-world, near-optimal tracking capability. The finalized heuristic method did not rely on future knowledge of the ground vehicle information, instead it was able to adjust and track based on current ground vehicle information. The heuristic approximation was developed using a three step process built upon the preexisting loiter logic of the SUAS autopilot.

Optimal Path Formulation

The first objective was determining the optimal SUAS flight path to perform convoy overwatch for a given ground vehicle profile. In the optimal path formulation, the ground vehicle profile information was known prior to solving the optimization problem. Therefore, the resulting optimal path was used as a metric to compare the accuracy of the heuristic-based approximations. This section presents the SUAS equations of motion, a multi-objective cost functional to minimize the slant range error with the minimal amount of control input, and the optimal control algorithm used to determine the optimal path.

Equations of Motion

The SUAS was modeled as a modified 2D Dubins (Dubins, 1957) airplane and included a level turn assumption. It was also assumed that the SUAS automatically used the required pitch control to maintain a constant altitude, reducing the degrees of freedom in the optimal control problem. The aircraft's roll rate ($\dot{\phi}$) was the only control variable used in the optimization. Assuming the SUAS only executed level turns allowed the roll angle to dictate turn rate and effectively described the SUAS's lateral-directional motion. For the optimization the SUAS and ground vehicles' ground velocity was an important parameter. While the SUAS's airspeed was fixed at the velocity corresponding to maximum endurance. However, the presence of wind (V_w) impacted the SUAS's groundspeed thus the SUAS groundspeed had to be calculated according to Equation 1:

$$V_G(i) = \sqrt{\left[V\cos\psi(i) - V_w(i)\cos\psi_w(i)\right]^2 + \left[V\sin\psi(i) - V_w(i)\sin\psi_w(i)\right]^2} \quad (1)$$

The SUAS equations of motion are defined in Equations 2-5:

$$\phi(i+1) = \phi(i) + \dot{\phi}(i)\,\Delta t \quad (2)$$

$$\psi(i+1) = \psi(i) + \left[\frac{g}{V_G(i)}\tan\phi(i)\right]\Delta t \quad (3)$$

$$X_{\text{SUAS}}(i+1) = X_{\text{SUAS}}(i) + \left[V_G(i)\cos\psi(i)\right]\Delta t \quad (4)$$

$$Y_{\text{SUAS}}(i+1) = Y_{\text{SUAS}}(i) + \left[V_G(i)\sin\psi(i)\right]\Delta t \quad (5)$$

The equations of motion were discretized based on a fixed time step, $\triangle t$. For a fixed $\triangle t$, these equations were used as first-order approximations to find the state information at the succeeding time step. The discrete form was consistent with the form sent to the optimization algorithm discussed later. The locations of the SUAS and the ground vehicle were required for calculating the slant range at each time step. Since the ground vehicle was friendly, as is the case for convoy overwatch, the exact location was known. For the optimal control algorithm, the location history of the ground vehicle was an exogenous input, passed directly into the optimizer. During flight test, the ground vehicle GPS coordinates were transmitted directly to the SUAS. The slant range at the current time step is defined as:

$$SR(i) = \sqrt{\left[X_{\text{GV}}(i) - X_{\text{SUAS}}(i)\right]^2 + \left[Y_{\text{GV}}(i) - Y_{\text{SUAS}}(i)\right]^2 + h^2} \quad (6)$$

Note that the SUAS altitude was held constant and the ground vehicle path was flat *(h(i) = h)*, which meant that a 3D slant range maps directly to a 2D standoff distance.

The cost functional:

$$J = \sum_{i=0}^{N}\left[\alpha\left(\frac{SR(i) - SR_{\text{desired}}}{SR_{\text{desired}}}\right)^2 + (1-\alpha)\left(\frac{\dot{\phi}(i)}{\dot{\phi}_{\max}}\right)^2\right]\Delta t \quad (7)$$

was the summation of the cost at each step. The optimal control was characterized by the control vector ($\dot{\phi}$) that minimized the scalar value *J* in Equation 7. This effectively minimizes control effort while maintaining sufficient slant range for the mission objective.

A weight factor (α) was used to gauge the relative importance placed on each term and ranges from 0 to 1. A low value of emphasized minimizing control effort, while a high value of favored reducing deviation from the desired slant range. The main goal of the convoy overwatch was to maintain the desired slant range from the ground vehicle as closely as possible. Not penalizing control effort resulted in aggressive, stop-to-stop saturated control inputs, correlating to erratic SUAS flight paths and decreased endurance. Therefore, α was set to 0.95 for the convoy overwatch scenario, which struck the balance between maintaining the desired slant range while still considering control effort. This level of α produced trajectories that were smooth enough to be tracked but still prioritized minimizing the deviation from the desired slant range.

For this optimal control problem to be representative of the real-world convoy overwatch scenario, the mathematical formulation included system constraints. The equations of motion, defined in Equations 2-5, are called dynamic constraints and act as equality constraints, forcing the optimizer to satisfy each state equation at each time step. Specific to this optimization, several path constraints were placed on the system to model the dynamics of the real-world flight test SUAS, discussed in detail later. Table 1 highlights the path constraints for roll angle (ϕ) and roll rate ($\dot{\phi}$). The roll angle was constrained to match the actual minimum turn radius of the flight test SUAS. Given the level turn assumption, constraining the roll angle simultaneously constrained the SUAS turn radius. Lastly, the maximum and minimum roll rates were determined based on real-world telemetry data. The roll rate constraint was rarely active because of its inclusion in the cost functional.

Table 1

Path Constraints for Optimal Control Problem

Variable	Min	Max
ϕ	$-40°$	$40°$
$\dot{\phi}$	-100 deg/s	100 deg/s

The optimizer used a finite time horizon method, named the "look-ahead" algorithm, for generating the optimal paths. In the look-ahead algorithm, shown in Algorithm 1, a finite time history of the ground vehicle's future path was made available to the optimizer. The number of nodes (N) were determined by multiplying the update frequency (f) by the look-ahead time (t_{look}). The look-ahead method started at the SUAS's initial conditions. The optimal control solution was only computed for the time interval specified by t_{look}. Then the SUAS state equations are propagated forward for $\triangle t$ seconds ($\triangle t = 1/f$) until it reached the next node. Next, a new solution interval was created using the same t_{look} and used the current states of the system as the new initial conditions. This process was repeated at each node until reaching the final time (t_f). It is important to note that the new optimal path could be solved at some multiple of $\triangle t$ if more computational time was required. In this case, there would be multiple nodes between the solution intervals.

Algorithm 1 : Look-ahead Algorithm

Determine total number of nodes: $N = t_{look} \times f$

Set initial conditions

for $t_0 : \Delta t : t_f$ **do**

  Acquire ground vehicle path and wind information (used as fixed parameters)

  Guess control vector $(\dot{\phi})$

  Propagate state equations forward (Eqns 2 - 6)

  Optimizer:

    Find $\dot{\phi}$ that minimizes cost functional (Eqn 7)

    Satisfy dynamic constraints (Eqns 2-5)

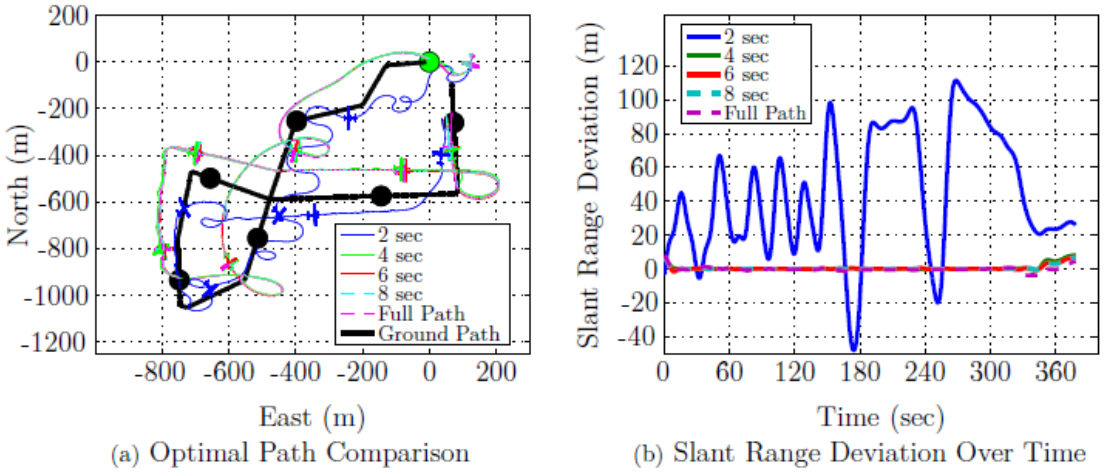
    Satisfy path constraints (Table 1)

  Set optimal states at 2nd time step as new initial conditions

end

Result: Optimal control and states correspond to the initial conditions of each time step from $t_0 : \Delta t : t_f$

The optimal control algorithm assumed the availability of the ground vehicle's path for a finite time horizon (t_{look}) and planned the optimal path based on that knowledge. Understanding how different values of t_{look} affected the optimal solution aided in algorithm selection for eventual implementation. There was a tradeoff between t_{look} and the realism of the model. In reality, the future path of the ground vehicle was not known with perfect certainty and therefore an estimate was required. A shorter t_{look} correlated with a more realistic estimate of the ground vehicle's location. Conversely, larger values of t_{look} typically resulted in a lower cost but were less feasible because of the uncertainty related with predicting the ground vehicle's path. To determine the appropriate t_{look}, several different look-ahead times are plotted and compared in Figure 1. Every optimal flight path was calculated using the same ground vehicle profile, zero wind speed, and the same initial conditions. MATLAB 2012b was the software package used to solve the optimal path using the fmincon function.



(a) Optimal Path Comparison (b) Slant Range Deviation Over Time

Figure 1 : Comparison of Optimal Paths Using Different Look-ahead Times and Full Path Information

In Figure 1a, an aircraft is plotted every 60 seconds to represent the location of the SUAS and a black circle is plotted to represent the ground vehicle location. The green circle represents the ground vehicle's starting point and the desired standoff distance from the ground vehicle was 150 meters. Figure 1b shows the slant range deviation over time. The x-axis is incremented by 60 seconds to correspond with the SUAS locations in Figure 1a. For each t_{look} value, Table 2 displays the corresponding cost, number of nodes, update rate, and run time per update.

Evaluating the flight paths in Figure 1a and 1b revealed a distinct difference between the optimal path calculated with $t_{look} = 2$ seconds and the flight paths created using a $t_{look} \geq 4$ seconds. Table 2 shows that the cost of the 2 second solution is over 480 times greater than all of the other solutions. Figure 1b portrays how the slant range of the 2 second optimal path immediately deviated from the desired slant range and grew in amplitude with time. Given the initial starting position and orientation, 2 seconds did not provide sufficient future knowledge for the optimizer to account for the ground vehicle behavior.

t_{look} (sec)	Cost (J)	# of Intervals	Δt (sec)	Interval Run Time (sec)	Total Run Time (sec)
2	47.12	554	0.67	0.13	72.0
4	0.098	554	0.67	0.17	94.2
6	0.075	554	0.67	0.25	138.5
8	0.071	554	0.67	0.31	172.7
Full Path	0.067	1	—	1845.6	1845.6

Table 2. Optimization Results for Figure 1

The required time for path convergence to the optimal path occurred for a lookahead time greater than 4 seconds. In fact, the paths with t_{look} ranging from 4 seconds to full path knowledge were nearly equivalent. This finding was critical because for $t_{look} \geq 4$ seconds the optimal path was not improved with additional increases in look-ahead time. The proximity of the various optimal path solutions suggested that the solution space near the global minimum was flat. The convergence criteria of the optimizer were not specific enough to discriminate between the solutions near the optimal path. This explained why the optimal paths look nearly identical in Figure 1a but had slightly different cost values in Table 2. A flat solution space was desirable because it allowed minor deviations to still result in the near-optimal flight path.

The timeliness of the solution was as equally important as its accuracy. The update frequency used for calculating the optimal solution was 1.5 Hz, meaning that $\triangle t = 0.67$ seconds. To be viable in real-time, the solution must converge to an optimal path at a rate faster than the update frequency. As a general rule of thumb, speeds twice as fast as the update frequency were desired because it allowed time for the autopilot to implement the controls. Therefore, t_{look} ranging from 4 8 seconds constituted viable solutions because of their speed and accuracy.[1] This meant that the look-ahead algorithm had the potential to determine the optimal path for convoy overwatch onboard the real-world, time-sensitive environment of a SUAS. Despite the speed of the look-ahead algorithm, the required advanced knowledge was not available to the SUAS in a real-time, dynamic environment. Therefore, a heuristic-based approximation of the optimal path devoid of prior knowledge of the ground vehicle information was developed.

Heuristic Approach

The second objective of this research was to demonstrate a near-optimal convoy overwatch in real time. A tracking algorithm called "follow-me" was engaged to execute a real-world, ground target tracking solution, agnostic to any notion of path optimality. The follow-me mode was native to the ArduPilot Mega (APM) 2.5, the flight test autopilot. This mode was the initial step in approximating the optimal path generated by the lookahead algorithm. Follow-me worked by collocating the current SUAS loiter point with the current GPS location of the ground vehicle. Successfully tracking a moving ground vehicle established a baseline performance and guaranteed feasibility of the problem. The next step was to identify parameters within the follow-me logic and determine which affected the optimality of the flight path. A Design of Experiments (DOE) was used to evaluate the impact of the selected parameters. Finally, to better decrease the cost of the heuristicbased path a Finite State Machine (FSM) was constructed to allow for dynamic path planning depending on the state of the SUAS.

Parameter Definition

Examinations of the existing APM fixed-wing aircraft firmware yielded three parameters, existing or readily implementable, directly utilized by the follow-me navigation logic. These parameters represent the effort to minimize roll rate and deviation from a desired slant range (Equation 7). The loiter radius, loiter range, and the navigation point lead time, were the three parameters used to build the heuristic approximation.

Loiter radius was the horizontal distance from the target point that the SUAS attempted to maintain. For a stationary ground vehicle, this represented a circular loiter. When the ground vehicle was inside or on the loiter radius, updates to desired heading (which were subsequently fed into lower level control loops) accounted for the ratio of the current target distance to the desired distance. The level of effort applied to achieve that distance directly represented the balance between control effort and slant range.

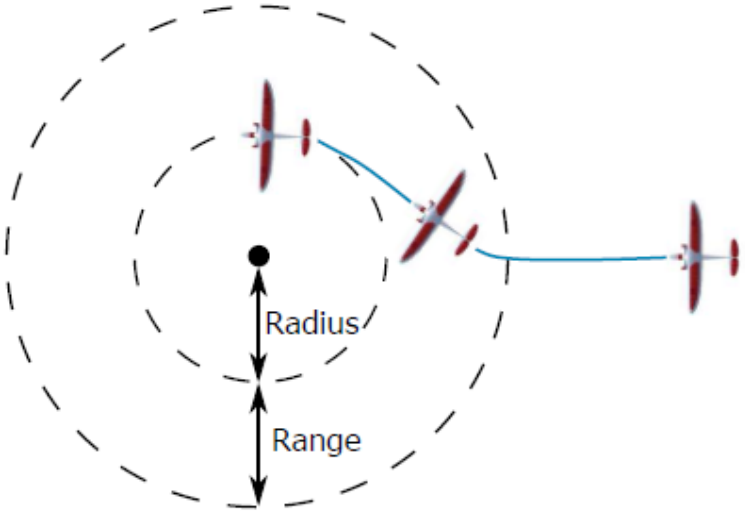


Figure 2 : Definition of Loiter Radius and Loiter Range

Loiter range was an additional distance beyond the loiter radius inside which the SUAS began a gradual transition from straight flight towards the target point to circular flight around the target point. Similar to the effects of loiter radius, control effort was directly based on a ratio representing relative SUAS position inside the range. Therefore, the loiter range parameter was a direct factor in control effort determination which impacted the optimality of any given flight. Figure 2 demonstrates the role of both loiter radius and range in the APM navigation logic.

Finally, the point to which the SUAS was navigating towards must be considered. An additional APM firmware modification called "lead time" was incorporated to allow for a simple estimation of the ground vehicle future location. The parameter represented a time constant that was multiplied by the ground vehicle velocity to adjust the navigation point ahead of the ground vehicle. The lead time was similar to t_{look}, in that it only considered a finite portion of the ground vehicle's future location. The lead time parameter, which was evaluated along with loiter radius and loiter range, defined the three parameters that were experimentally tuned to approximate the optimal path for ground vehicle tracking.

Finite State Machine Construction

The last step was to develop navigation logic in the FSM that was responsive to real-time SUAS conditions. The existing performance was examined to identify any com- bination of SUAS and ground vehicle states that warranted alternative behavior. However, rather than performing this analysis on data from the unmodified follow-me mode, it was important to first adjust any relevant system settings to maximize the flight path's opti- mality. Flight data garnered from these settings resulted in a more appropriate determina- tion of state definitions.

Test Setup

SUAS Testbed The SUAS used to flight test the heuristic approach was the Sig Rascal (Figure 3). The aircraft had a 110" wingspan, was 75.75" long, weighed 11 pounds empty, and was propelled by an electric motor. The motor was powered by three, 4-cell batteries which allowed for 40 minutes of flight time. The video surveillance payload, shown in the inset of Figure 3, was located underneath the Rascal's fuselage which allowed the camera to have an unimpeded 360◦ of pan and 90◦ of tilt. The camera placement maximized the camera's possible field of view, giving the SUAS increased flexibility for performing surveillance and reconnaissance missions.

A HackHD camera was used for gathering the video data during flight test. The camera recorded video in 1080P high definition, stored it onboard a Micro-SD card, and was outfitted with the stock 160◦ wide angle lens. The HackHD was attached to the pan- tilt gimbal, powered by two Hitec servos, and was commanded to continuously point at the current GPS coordinates of the ground vehicle.

Figure 3 : Gimbaled Camera Apparatus Affixed to SUAS Testbed

The APM was the autopilot used for the hardware in-the-loop (HIL) simulations and flight tests. APM was a full-featured, open source autopilot that incorporates all of the necessary functions for a variety of flight modes. The autopilot included a 3-axis accelerometer, rate gyros that measured the orientation changes of the aircraft, a 16 MHz Atmega328 processor onboard and could control 8 different channels. The APM 2.5 had a variety of flight modes that could be engaged through the MissionPlanner ground station software interface. The follow-me flight mode served as the launching point for the heuristic approximation algorithm. Because all of the APM 2.5 source code was available, the native follow-me code was used as a template to create both the DOE and FSM mode.

Flight Test Location

All flight tests were conducted at Camp Atterbury Joint Maneuver Training Center, IN. This Army installation had restricted airspace, which did not require a FAA issued certificate of authorization for SUAS flight test. Camp Atterbury had extensive ranges and multiple launch sites for SUASs, which made the location prime for flight test. The range had numerous roads that allowed for simulating real-world convoy operations. Figure 4 shows a view the SUAS airstrip and the ground path driven to simulate a convoy path.

The path had a "figure 8" configuration that featured both left and right turns, as well as straightaways. This path was specifically chosen because of its complexity and variation in ground vehicle behavior. The goal was to develop a heuristic-based approximation based on an overtly complex convoy route, with the assumption that tracking performance would naturally improve with simpler ground vehicle paths.

Figure 4 : Camp Atterbury Test Range, Indiana. Base Image: ©DigitalGlobe, IndianaMap Framework Data, USDA Farm Service Agency, ©2015 Google

The heuristic-based approximation of the optimal path was evaluated through a mixture of flight test and HIL simulations. The purpose of using HIL simulations was to supplement the flight test efforts, not to replace them. Therefore, the HIL environment was constructed to mimic the flight test setup as closely as possible. The HIL simulations featured an APM 2.5 and its MissionPlanner ground station software controlling a virtual Sig Rascal flying in the FlightGear simulator. A representative ground vehicle profile was driven on the same path used for the flight tests. A wind environment typical to Camp Atterbury was used in the simulation and the same wind conditions were used for each of the HIL simulated flights. Lastly, validation flights were conducted to ensure the similarity of control effort between the real-world and the virtual Sig Rascal. Reducing these independent variables to known quantities allowed the cost functional values from each HIL simulation to be directly compared. HIL simulations were essential due to the logistical constraints of flight test, the quantity of data required to build the heuristic approximation of the optimal solution, and reducing the variability of the wind conditions and ground vehicle path. There was confidence in the HIL results because of the ability to test the custom autopilot firmware modifications in a representative, high fidelity environment.

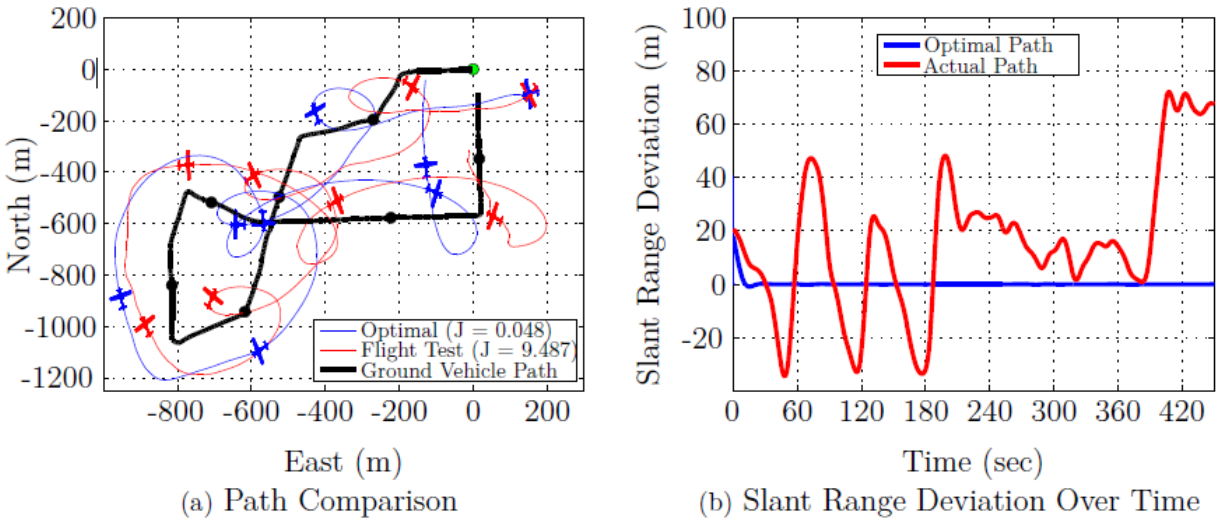
Results

The results of the flight test and HIL simulations are discussed in this section. The progression from the follow-me mode, to the DOE results mode, to the FSM mode are shown and their impact on the path optimality is discussed. The results displayed in this section are generated from both flight test and HIL simulation data. The optimal path was calculated with the look-ahead algorithm (Algorithm 1) and used the same ground vehicle profile, initial conditions, wind speed, wind direction, and SUAS airspeed measured by the autopilot during the flight test. The goal was to create an optimal solution that incorporated all of the specific test conditions experienced by the SUAS during the flight test. This allows a realistic comparison between the flight test and simulation result with the optimal path. Finally, all three modes are compared and analyzed to determine their ability to approximate the optimal path generated using the look-ahead method.

Follow-Me Mode

The first vehicle tracking function implemented onboard the SUAS was the followme mode. In this mode, the SUAS loiter point was set to be the ground vehicle's location. The flight test results, shown in Figure 5, compare the performance of the flight tested follow-me mode with its corresponding optimal path.

During the flight test, the autopilot controlled the SUAS to maintain a 150 meter loiter around the moving ground vehicle as it drove the "figure 8" ground path. The loiter radius was chosen based on engineering judgment.



(a) Path Comparison (b) Slant Range Deviation Over Time

Figure 5 : Results from the Follow-me Flight Test Compared to Optimal Path

The performance of the follow-me mode was sensitive to the initial conditions of the SUAS. Therefore, the ground vehicle began driving when the SUAS was located behind the vehicle at a similar heading angle. The starting SUAS configuration minimized the initial amount of control effort required to achieve the desired slant range. In Figure 5a, an aircraft is plotted every 60 seconds to represent the location of the SUAS for both the optimal and the flight test paths and a black circle to represent the ground vehicle location. The starting ground vehicle location is represented with a green circle.

Figure 5b compares the slant range deviation of the optimal path versus the flight test path as a function of time, showing how close the follow-me mode was to maintaining the desired slant range. The time intervals are set at 60 seconds to correlate with the aircraft locations in Figure 5a. A 0.1 Hz low-pass filter was used on both slant range data sets to eliminate the higher frequency noise.

Comparing the performance between the two paths reveals drastic differences and inadequacies of the flight-tested follow-me mode. The values of the cost functional, displayed in the legend, show a discrepancy between the optimal path cost and follow-me mode, which cost 200 times greater. Due to the multi-objective nature of the cost functional, the specific cost values do not have physical meaning and therefore are not intuitive. However, the magnitude of the cost functional difference reveals the disparity between the optimal path and the follow-me mode. This discrepancy between the optimal and follow-me paths can be largely attributed to the simplicity of the follow-me mode.

Using the follow-me mode, the SUAS successfully tracked the moving ground vehicle. However, there was a substantial difference between the optimal path and the path created by the follow-me mode. This makes sense because the optimal path specifically minimized slant range error and control effort, while the follow-me mode only sought to loiter about a moving point. For the first 240 seconds, the SUAS continually overshot the desired slant range as it aggressively tried to maintain the desired standoff. The following 120 seconds showed the SUAS flying closer to the desired slant range with less overshoots. This region of performance occurs when the ground vehicle turns to head east and drove straight for 800 meters. Coincidentally, the subtle slant range deviation defining this period occurred when the SUAS predominantly had a tail wind. The benefit of the tail wind for this period degraded the SUAS performance after the ground vehicle made the final turn. The ground vehicle's final left-hand turn occurred while the SUAS begins a righthand loiter. Immediately before the turn, the SUAS was at the desired standoff, but immediately following the turn its slant range error jumped significantly. To continue loitering around the ground vehicle, the SUAS had to turn back into the wind. The head wind drastically reduced the SUAS ground speed, causing a drastic increase in slant range error. In the optimal path, after the SUAS converged to the desired slant range there were no deviations because the future location of the ground vehicle was available to the optimizer.

The initial flight test successfully demonstrated the follow-me autopilot function. However, this method insufficiently approximated the optimal path and the two orders of magnitude difference between the costs highlighted the need for a better approximation. The goal of the following heuristic method iteration was to minimize the slant range error over the entire flight path through statistical analysis.

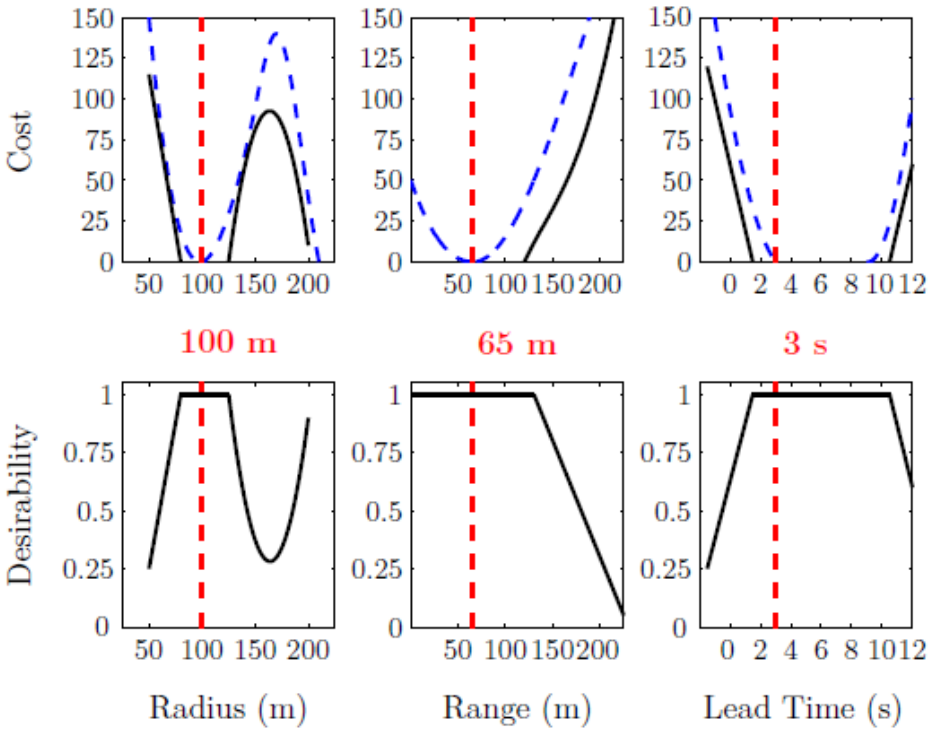


Figure 6 : Factor Profiler for Combined Regression Model during DOE

Design of Experiments Results Mode

A DOE was used to determine the best parameter values for the follow-me tracking task. A total of 26 HIL simulations run at various loiter ranges, loiter radii, and lead times were used to build a traditional regression model. Each of the simulated flights were run using the same ground vehicle profile, with the exact same wind model. All of the test points were entered in as non-coded (engineering) units and an analysis of variance was performed on all of the significant terms using the statistical software JMP.

A factor profiler, shown in Figure 6, was constructed for the model to determine the best combination of parameter settings. Figure 6 shows the relative cost and desirability for each of the parameters at a range of values. The desired parameter values minimized the cost, which simultaneously maximized the desirability. Each of the recommended parameters was highlighted graphically in Figure 6 with a vertical, red dashed line. The recommendations from the combined regression model were 100 meter loiter radius, 65 meter loiter range, and 3 second lead time. In this case, all three parameters were determined significant at the 0.05 level. Since the DOE was tuned for a specific path, it was unknown how these parameters would change for another ground vehicle path. This uncertainty was the motivation for using the "figure 8" path, with the assumption that tuning the algorithm to a complex ground vehicle path would allow for robustness towards simpler paths.

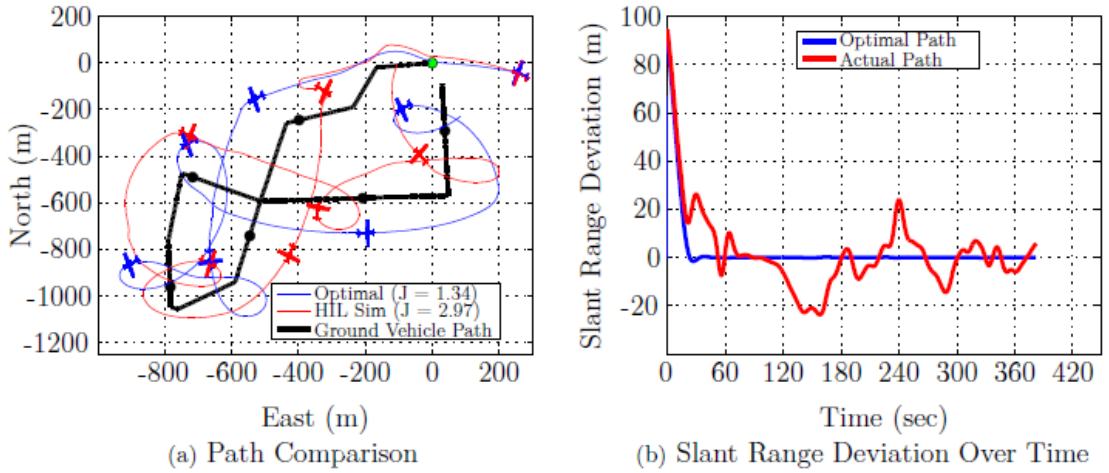


Figure 7 : HIL Simulation Results of DOE Suggested Parameters Compared to Optimal Path

A HIL simulation was performed at the DOE recommended lead time of 3 seconds, loiter range of 65 meters, and loiter radius of 100 meters. Figure 7a shows the path comparison of the HIL simulated DOE results and the optimal path. Figure 7b shows the slant range as a function of time. Higher frequency noise was eliminated from the data sets using a 0.1 Hz low pass filter.

The cost functional values for both paths are shown in the legend in Figure 7a. The cost of the HIL simulated path was only twice the cost of the optimal path. This was a substantial improvement from the follow-me mode shown in Figure 5a. Additionally, comparing the slant range behavior with time in both Figures 5b and 7b, demonstrates how the results from the DOE allowed the SUAS to adhere more closely to the desired slant range. Despite all of the improvements, there were still several noticeable overshoots in slant range deviation. These deviations resulted from the lack of future knowledge avail able to the APM. The sharp ground vehicle turns at 150 seconds and 240 seconds constituted drastic changes in the ground vehicle's direction. The DOE mode had the lead time parameter, which is a simple linear estimator, but was unable to account for these sudden turns. Contrast this performance with the optimal path generated using the look-ahead method which had perfect future knowledge. The optimizer anticipated sharp vehicle turns and adjusted the SUAS's roll rate accordingly to maintain the desired slant range throughout the entire ground vehicle profile. The perfect, future knowledge of the ground vehicle path assumption makes the optimal path infeasible to fly in real-time, but it serves as a best case for judging the optimality of the heuristic algorithm.

Finite State Machine Mode

The greatest increase to the path cost occurred during large deviations from the desired slant range. Setting the cost functional weight factor to α =0.95 heavily emphasized minimizing slant range error. Adding logic to supplement the existing DOE tuned parameters minimized the negative impact caused by large slant range error. The FSM added two additional parameters to the flight firmware and categorized the SUAS performance into three different states. The two additional parameters added were an instantaneous cost threshold ($J_{\text{threshold}}$) and a loiter radius buffer ($\triangle LR$). The slant range error ($SR - SR_{\text{desired}}$) and the instantaneous cost (J_i) were the two variables that dictated the active state within the FSM. Depending on the values of these two variables, the SUAS was in one of the three states: Standard tracking, High Range tracking, or Low Range tracking. Figure 8 shows a graphical representation of the FSM.

Standard tracking was the predominant state flown in simulation and was active when J_i was less than J_{th}. In this state, the DOE suggested parameters were flown. The High Range tracking state activated when J_i and SR were greater than J_{th} and SR_{D} respectively. In this instance, the loiter radius was reduced by $\triangle LR$. The smaller loiter radius increased the turn rate of the SUAS, allowing for the SUAS to quickly reduce the slant range error. The Low Range tracking state activated when J_i was greater than J_{th} and SR was less than SR_{D}. In this state, the SUAS was too close to the ground vehicle and as a result its loiter radius was increased by $\triangle LR$. The High and Low Range tracking states were designed to temporarily adjust the desired loiter radius, enabling the SUAS to quickly reacquire SR_{D} and fall back under J_{th}. Once the SUAS was under J_{th}, it would return to the Standard tracking state.

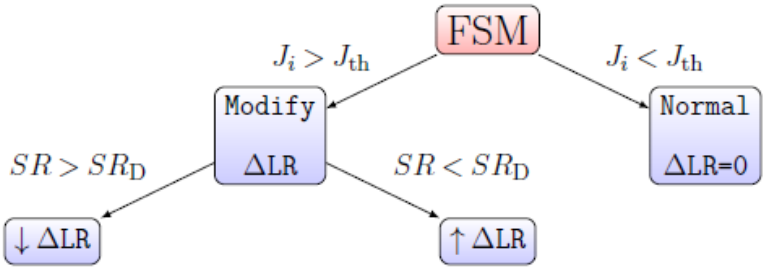
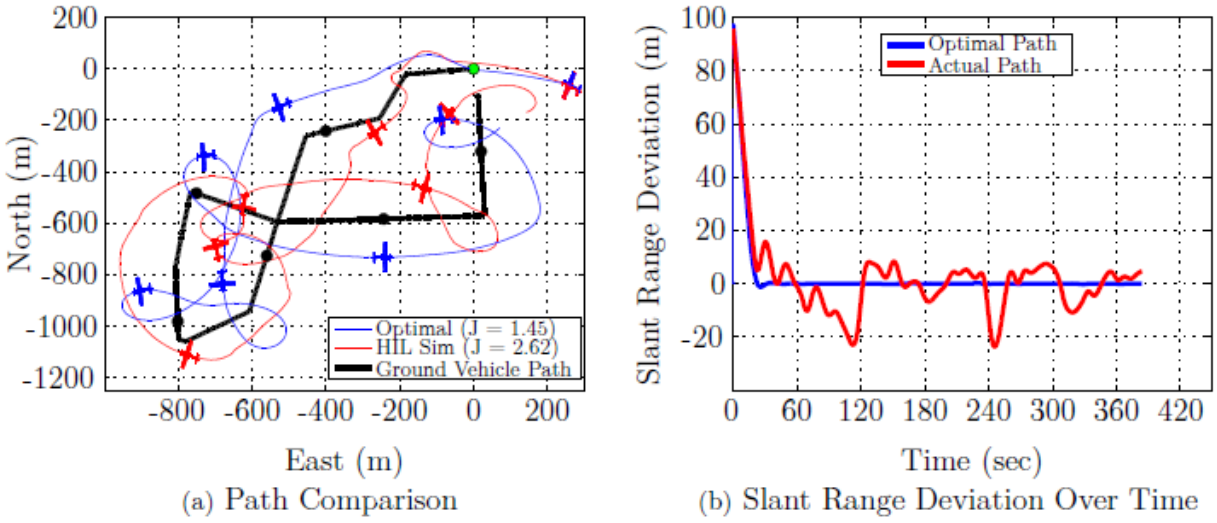


Figure 8 : Flowchart representing the logic that creates the FSM.

For the HIL simulations, the FSM was implemented on the APM with $J_{\text{threshold}}$ set to 0.003. This $J_{\text{threshold}}$ value activated the FSM logic when the to a slant range error was greater than 9 meters. $J_{\text{threshold}}$ was tuned based on analysis of a J_i profile and aimed to execute state transitions when necessary but not excessively. The loiter radius buffer ($\triangle LR$) was set to 35 meters, which represented a ±23% change of the loiter radius parameter. $\triangle LR$ was determined using engineering judgment with the intent to affect measurable changes while avoiding unsafe behavior if flown in real life. Figure 9a compares the flight path generated during a HIL simulation of the FSM mode to the optimal path calculated using the look-ahead algorithm. Figure 9b shows the slant range as a function of time for both the FSM simulation and the optimal path. A 0.1 Hz low pass filter was used on both slant range data sets to eliminate the higher frequency noise.



(a) Path Comparison (b) Slant Range Deviation Over Time

Figure 9 : HIL Simulation Results of FSM Compared to Optimal Path

Comparing the costs for the DOE results flight to the FSM flight reveals only a marginal improvement in performance. However, the benefit of the FSM is observed when comparing Figures 7b and 9b. Both the DOE and FSM flights had noticeable overshoots around 120 seconds, 240 seconds, and 300 seconds. These overshoots occurred because the SUAS reached the desired loiter radius attached to the ground vehicle (see Figure 2) and started to loiter. This was problematic because as the SUAS turned to loiter, the ground vehicle continued traveling along its path. The spikes in slant range error occurred during the short time when the ground vehicle and SUAS were traveling in opposite directions. With the FSM enabled, the SUAS quickly reacted to the high slant range error, increased its turn rate, and returned back to the desired slant range. The aggressive turning resulted in shorter duration spikes in slant range. The FSM behavior is seen graphically in Figure 9b by the distinct spikes that plunge away from the desired slant range, but quickly return. Without the FSM enabled (Figure 7), the SUAS was slower to reacquire the desired slant range, therefore decreasing the optimality of the flight path. The optimal path avoided the slant range overshoots encountered in both the DOE results and FSM mode by flying ahead of the ground vehicle and then turning around. By passing the ground vehicle instead of turning once the desired loiter radius was reached, the optimal path maintained the desired slant range throughout the turn. This optimal behavior was only possible because of the future ground vehicle path knowledge made available to the look-ahead algorithm.

Analysis of Each Mode

The second objective of this research was achieved through the development and simulation of the FSM mode. The FSM was developed iteratively starting with the followme functionality.

The graphical representation of both the flight paths and slant range deviation charts in Figures 5, 7, and 9 are beneficial for model verification. However, the improvement from the follow-me mode to the DOE results to the FSM is difficult to discern from simply comparing Figures 5, 7, and 9 to each other. Table 3 compares the cost functional values of the three different modes of the heuristic-based algorithm.

Replicate	Follow-Me Cost (J)	DOE Results Cost (J)	FSM Cost (J)
Initial test	9.732	2.967	2.620
1	6.747	5.249	1.966
2	5.656	2.799	2.513
3	4.915	1.985	2.361
Average	6.763	3.250	2.365
Std Deviation	2.118	1.400	0.286

Table 3
HIL Results for Follow-Me, DOE Results and FSM

The follow-me mode, the DOE settings mode, and the FSM mode were all simulated four times in HIL. Each of the 12 flight paths used the same ground vehicle profile, similar initial conditions, and the same wind profile. Controlling all of these factors allowed for each of the cost functional values to be directly compared to one another. The 2nd to last row of Table 3 displays the average cost functional value for each mode. The last row shows the standard deviation of the costs from the HIL tests. Both the average cost and the corresponding standard deviation decreased significantly from the followme mode to the FSM mode. Completing the DOE and accepting the suggested settings yielded an average cost reduction of 52%. Incorporating the FSM logic further increased the algorithm's optimality by another 27%. The final algorithm that used the FSM logic on top of the DOE results accounted for a 65% increase in performance compared to followme.

The variation between the different replicate data points is largely caused by the slight differences in the initial conditions. To appropriately capture the variance of the data, Figure 10 displays a 95% confidence interval for the cost of each mode and their corresponding optimal paths.

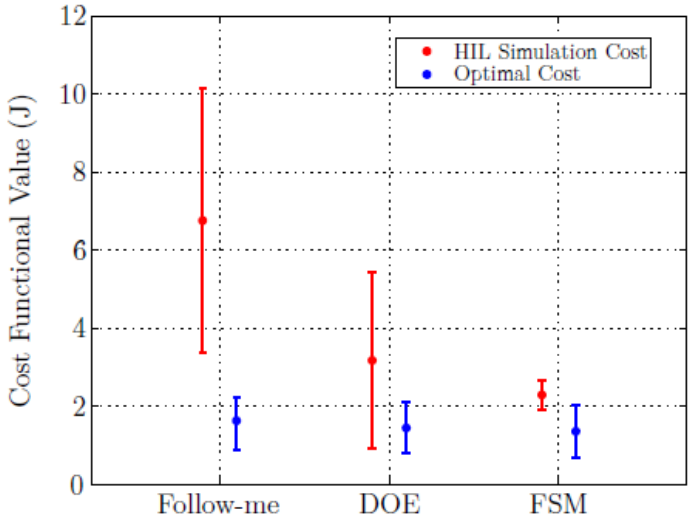


Figure 10 : 95% Confidence Intervals for Each Heuristic Algorithm Mode with Corresponding Optimal Paths

There are two important results shown in Figure 10. First, the iterative nature used to develop the algorithm successfully decreased the average cost and lowered the variation of the cost functional values. The order of magnitude decrease of the standard deviation from the follow-me mode to the FSM revealed a significant increase in the algorithm robustness. The FSM's superior performance, coupled with a lower standard deviation increased the confidence in the final version of the algorithm. Secondly, the proximity of the FSM results to its corresponding optimal path indicated that it accurately and consistently approximated the optimal solution for the given ground vehicle path.

The progression from follow-me to the DOE results to the FSM resulted in a closerto-optimal path planning algorithm. This trend validated the logic used when refining the algorithm because the DOE was used to narrow in on specific parameter values that correlated to a closer-to-optimal loiter logic. Additionally, incorporating the FSM increased the robustness of the path planner using the DOE suggested parameters.

Conclusion

A novel approach to autonomous convoy overwatch was presented that used optimal control-based techniques to build a heuristic algorithm. First, a cost functional was developed to model the optimal behavior of a SUAS tracking a cooperative, moving ground vehicle. The multi-objective cost functional primarily focused on minimizing the deviation from a desired slant range, while also considering the control effort. Using the look-ahead algorithm, the optimal path was developed for a given ground vehicle path. This algorithm was not able to be implemented onboard the SUAS and therefore a heuristic-based algorithm was developed to approximate the optimal path. Developing this algorithm consisted of three primary phases: the follow-me mode, the DOE results mode, and the FSM mode. The progression from the follow-me mode to the DOE results to the FSM resulted in a 65% increase in optimality. This trend validated the logic used when refining the algorithm because the DOE was used to narrow in on specific parameter values that correlated to an improved optimal loiter logic. While some of the flights were completed during flight test, the majority of the data were collected via HIL simulations. Therefore, it is recommended to flight test the three modes to validate the results shown in Figure 10.

The FSM was able to approach the cost of the optimal path without the benefit of perfect, future knowledge of the ground vehicle's location. The relatively low cost of these flights coupled with the considerable performance increases over default capabilities indicate that near-optimal flight paths for convoy overwatch are operationally feasible using a real-time, heuristic strategy implemented onboard the autopilot.

References

Dubins, L. E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79 (3), 497-516.

Frew, E., Lawrence, D., & Morris, S. (2008). Coordinated standoff tracking of moving targets using lyapunov guidance vector fields. *Journal of Guidance, Control and Dynamics , 31* (2), 290-306.

Geiger, B., Horn, J., Sinsley, G., Ross, J., Long, L., & Niessner, A. (2008). Flight test- ing a real-time direct collocation path planner. *Journal of Guidance, Control and Dynamics, 31* , 1575-1585.

Gertler, J. (2012). *U.S. Unmanned Aerial Systems* (Tech. Rep.). 101 Independence Avenue SE, Washington DC, 20540: Congressional Research Service.

Jodeh, N., Coon, T., Masternak, T., Cobb, R., & Agte, J. (2014). Optimal airborne tra- jectories for data collection from emplaced ground sensor arrays. *Proceedings from AIAA Guidance, Navigation and Control Conference*, National Harbor, MD.

Kim, S., Oh, H., & Tsourdos, A. (2013). Nonlinear model predictive coordinated standoff tracking of a moving ground vehicle. *Journal of Guidance, Control and Dynamics, 36* (2), 557-566.

Lawrence, D., Frew, E., & Pisano, W. (2008). Lyapunov vector fields for autonomous UAV control. *Journal of Guidance, Control and Dynamics, 31* (5), 1220-1229.

Prévost, C., Théiault, O., Desbiens, A., Poulini, E., & Gagnon, E. (2009). Receding hori- zon model-based predictive control for dynamic target tracking: a comparative study. *Proceedings from AIAA Guidance, Navigation and Control Conference,* Chicago, IL.

Quigley, M., Goodrich, M., Griffiths, S., Eldridge, A., & Beard, R. (2005). Target acqui- sition, localization, and surveillance using a fixed-wing mini-UAV and gimbaled cam- era. *Proceedings from IEEE International Conference of Robotics and Automation,* Barcelona, Spain.

Rysdyk, R. (2006). Unmanned aerial vehicle path following for target observation in wind. *Journal of Guidance, Control and Dynamics, 29* (5), 1092-1100.

Smith, N., Cobb, R., Pierce, S., & Raska, V. (2013). Optimal collision avoidance trajec- tories via direct orthogonal collocation for unmanned/remotely piloted aircraft sense and avoid operations. *Proceedings from AIAA Guidance, Navigation and Control Conference*, Boston, MA.

Yoon, S., Park, S., & Kim, Y. (2012). Guidance law for standoff tracking of a moving tar- get with leader-follower unmanned aerial vehicles. *Proceedings from AIAA Guidance, Navigation and Control Conference,* Minneapolis, MN.

Notes

[1]All solutions computed on a Samsung ATIV Smart PC pro (Intel i5 processor) using MATLAB 2012b under default settings