A Comparison of Automatic Nap-of-the-Earth Guidance Strategies for Helicopters

Eric N. Johnson[*]
Lockheed Martin Professor of Avionics
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332
eric.johnson@ae.gatech.edu

John G. Mooney[†]
Graduate Research Assistant
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332
john.g.mooney@gatech.edu

Abstract

This paper describes updated results from a partnership between the Sikorsky Aircraft Corporation and the Georgia Institute of Technology to develop, improve, and flight test a sensor, guidance, navigation, control, and real-time flight path optimization system to support high performance Nap-of-the-Earth (NOE) helicopter flight. The emphasis here is on optimization for a combination of low height above terrain/obstacles and high speeds. Here, multiple obstacle mapping and guidance methods for generating the desired flight path are evaluated, including (1) a simple processing of each laser scan; (2) a motion primitive optimization method, and (3) a potential field based method. Simulation and flight test results have been obtained utilizing an onboard laser scanner to detect terrain and obstacles while flying at low altitude, and have successfully demonstrated obstacle avoidance in a realistic semi-urban environment at speeds up to 15 m/s while maintaining a miss distance of 15 m horizontally and vertically. Additionally, the performance of the selected LADAR model in multiple degraded environments was observed. The performance of the three guidance methods are compared to a published benchmark model. These results validate the above approaches and pave the way for extensions to this work, including multi-aircraft collaboration and other more complex scenarios.

1 Introduction

Unmanned aerial vehicles (UAVs) and optionally piloted aircraft are expected to play an increasingly important role in both civil and military applications. A specific challenge for military unmanned helicopters is reducing vulnerability of the aircraft during operations. Vulnerability reduction through Nap-of-Earth (NOE) flight (low altitude, high speed) is a currently accepted tactic for manned military helicopters and an appealing choice for unmanned variants. For a manned aircraft, NOE flight is characterized by the need for a skilled human operator utilizing their own eyes to provide both raw terrain information as well as the interpretation of that information for flight control. For optionally piloted or otherwise unmanned helicopters, there is a need to provide this same NOE capability with the inclusion of sensor(s) to gather terrain information, along with the appropriate guidance and control methods to make use of it.

Automatic flight of helicopters in the presence of obstacles has been explored by a number of researchers.

[*] http://uav.ae.gatech.edu
[†] Corresponding Author

During the early 1990s, researchers at NASA Ames implemented the Automated NOE Flight Program on habited helicopters to examine the feasibility and development of requisite technologies for automating NOE flight to reduce pilot work load (Cheng and Lam, 1994). As part of the DARPA Sandblaster program, Sikorsky Aircraft has flight demonstrated an integrated flight controls, sensor, and display system that is capable of automated approach to a point; but with some pilot intervention (Cowherd, 2007). Vision-based methods are of interest because they are potentially light weight, inexpensive, and passive (Ortiz and Neogi, 2007) (Zufferey et al., 2010). Larger aircraft, on the other hand, due to their payload capability can use active sensors, such as LADAR or radar. Due to their greater payload capability, larger aircraft can use active- based sensors, such as radar and LIDAR. Using two 24 Ghz pulse radar sensors, Kandil and Wagner of the University of Heidelberg, Germany implemented a collision avoidance algorithm using a potential field method on an Aeroscout B1-100 unmanned helicopter (Kandil et al., 2010). For obstacle avoidance in urban environments, Kim (Seoul National University) and Sastry (UC-Berkeley) implemented a model predictive control (MPC) framework into their Yamaha R-50 equipped with a Sick LMS-200 (Shim et al., 2005). Scherer et. al. (Scherer et al., 2008) specifically used a custom 3D laser scanner to fly in an urban setting at speeds up to 10 m/s. NASA Ames Aeroflightdynamics Directorate and the University of Minnesota have utilized a Yamaha RMAX helicopter equipped with a rotating Sick LMS291-S005 laser sensor as a testbed for 3D autonomous rotorcraft navigation in urban environments using both a fast A*-based 3D route planner and a 3D route planner via terrain plane slicing of a 2D planner (Tsenkov et al., 2008). Furthermore, the work by Mettler, Kong, Goerzen, and Whalley (Mettler et al., 2010) seeks to establish benchmarks that quantitatively and qualitatively measure the performance of aerial autonomous guidance systems.

Under this continuing effort (Johnson et al., 2011), a number of sensor modalities have been considered, including radar, sonar, LIDAR/LADAR, and vision (monocular and stereo) techniques. Our subsequent work has focused on LADAR, due to available accuracy, range, and update rate of existing off-the-shelf sensors to support flight test evaluation. To support evaluation of methods in simulation for trade studies and to prepare for flight testing, a detailed simulation model was developed for scanning LADAR systems, allowing several existing off-the-shelf models to be tested in a closed loop simulation environment (Hokuyo UTM-30LX, Sick LMS291-S05, and Sick LD-MRS). Based on factors such as maximum range, weight, power, and field of view, the Sick LD-MRS system was then selected for further development and flight test validation of an automatic NOE flight system on a small, unmanned helicopter.

The remainder of this paper is organized as follows. First, four of the methods for generating the desired path to avoid obstacles are described. Second, a description of the aircraft utilized for simulation and flight test evaluation is included. Third, simulation results are presented, including results of benchmark tests described by Whalley, et. al. in (Mettler et al., 2010). Finally, flight test results are discussed.

2 Guidance and Path Generation

Three methods for providing the guidance and path generation are explored here: (1) a simple processing of each laser scan, (2) a motion primitive optimization method, and (3) a potential-field method. The first is a relatively simple 2D method, working in the vertical plane. The latter two are computationally more expensive and operate in three dimensions.

2.1 Simple Processing of Single Scan Method

Here, the laser scanner is mounted such that a terrain profile is measured from directly below the helicopter to out in front of the helicopter, normal along the direction of travel, as far above the horizon as possible. In the case of the Sick LD-MRS, this translates to a field of view encompassing the bottom of the helicopter up to approximately 20 degrees above the horizon. In the simple processing method, each data set from the laser is first converted to a set of 3D points in an Earth fixed frame. The projected horizontal flight path is then compared to every point in the most recent laser scan. To ensure that the future path does not collide with any identified obstacle, a height restriction is then applied to each known point. The combination of observed obstacle points and height restrictions defines potential future trajectories. Altitude and vertical speed commands are then modified to achieve obstacle avoidance. The method pre-supposes an altitude control law that can track a specified altitude and vertical speed command. Here, altitude is the primary variable tracked. The vertical speed command is used to provide an additional feedforward signal to the controller for improved altitude tracking.

For scan points out in front of the aircraft, a minimum height restriction based on scan point i is found by:

$$h_{min,i} = h_i + \Delta h_{desired} - \frac{1}{2} a_{desired} \Delta t_i^2 \tag{1}$$

where $\Delta h_{desired}$ and $a_{desired}$ are specified vertical miss distance and desired maneuver vertical acceleration respectively, and Δt_i is the time remaining until the aircraft will be within the specified horizontal miss distance of scan point i. The commanded altitude is enforced as the maximum of the current command and the minimum from all scan points. A similar action occurs for vertical speed command as well:

$$\dot{h}_{min,i} = -\sqrt{4a_{desired}(h - h_i - \Delta h_{desired}) + 2(a_{desired}\Delta t_i)^2} + a_{desired}\Delta t_i \tag{2}$$

where h is the current altitude of the helicopter. This expression ensures both a smooth pull up at the desired maneuver acceleration and a push-over at the top with the same acceleration. Or, if the current altitude is low enough that the aircraft cannot smoothly pull up at the specified vertical acceleration level ($h < \max_i h_{min,i}$), then this same limit is found instead by:

$$\dot{h}_{min} = -a_{desired}\Delta t_i \tag{3}$$

For scan points within a specified horizontal miss distance of the aircraft (i.e., points below the aircraft) these same formulae are used, but the time remaining is calculated based on capturing the desired minimum altitude using the specified vertical acceleration:

$$\Delta t_i = \begin{cases} \sqrt{\left|\frac{2(h - h_i - \Delta h_{desired})}{a_{desired}}\right|}, & \text{if } h > h_i + \Delta h_{desired} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

When the range of the terrain sensor is sufficient for the given terrain profile and selected vertical acceleration levels, this simple method provides commanded altitude and vertical speed to meet prescribed miss distances and vertical acceleration levels.

As described, this method can be utilized to modify any guidance policy to ensure the vertical profile does not come within specified distance of terrain. That is, act as a ground collision avoidance system. For true NOE flight, the nominal profile is set to be a specified nominal vertical descent rate. This combination of a nominal descent rate and ground collision avoidance logic results in NOE flight, at least in the vertical plane.

2.2 Optimization of Motion Primitives Method

Due to limitations of the potential field method (discussed below in Section 4.3) another method was developed to be more straightforward and less computationally intensive. This method uses a simple set of six

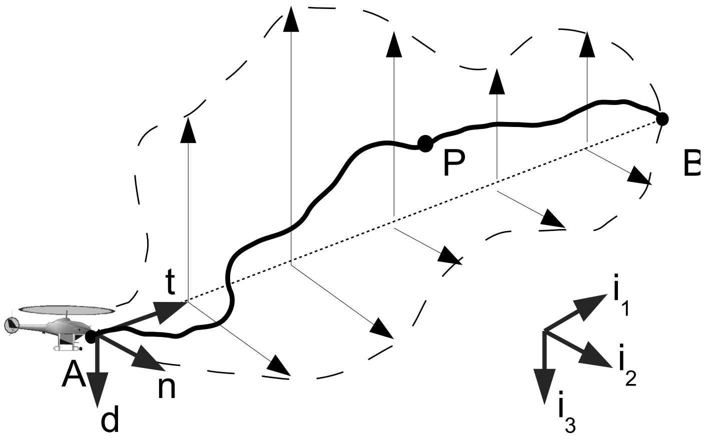


Figure 1: Trajectory defined by sinusoidal deviations from straight-line path

motion primitives to optimize lateral and vertical cross-tracking error from a straight-line path, while treating the obstacle field as a constraint set. The use of motion primitives for autonomous systems, including aircraft, is well known (LaValle, 2006) (Schouwenaars et al., 2003) (Frazzoli, 2001).

The unique contribution here is in the selection of a relatively small set of sinusoidal primitives. These primitives reduce the problem to six dimensions, the space of which contains a set of intuitive and good enough solutions. These primitives also have properties that make a number of later computations simpler. Figure 1 demonstrates how the three l_i parameters and three v_i parameters describe any point P on the planned trajectory between points A and B.

$$\vec{r}_p(s) = \vec{r}_A + s\vec{t} - l(s)\vec{n} - v(s)\vec{d} \tag{5}$$

$$D = \left\| \vec{r}_B - \vec{r}_A \right\|, \; \vec{t} = \frac{\vec{r}_B - \vec{r}_A}{D}, \; \vec{n} = \frac{\tilde{i}_3 \vec{t}}{\left\| \tilde{i}_3 \vec{t} \right\|}, \; \vec{d} = \tilde{t}\vec{n} \tag{6}$$

$$l(s) = l_1 \sin\left(\frac{s\pi}{D}\right) + l_2 \sin\left(\frac{2s\pi}{D}\right) + l_3 \sin\left(\frac{3s\pi}{D}\right) \tag{7}$$

$$v(s) = v_1 \sin\left(\frac{s\pi}{D}\right) + v_2 \sin\left(\frac{2s\pi}{D}\right) + v_3 \sin\left(\frac{3s\pi}{D}\right) \tag{8}$$

Each of the path parameters are a measure of the maximum deviation from the nominal path by the associated sinusoid. There are two sets of constraints on these parameters. First, we typically do not want a flight path that is substantially lower than the nominal straight line path. In order to ensure that most parts of the sinusoid stay above the straight line, the following linear constraints are imposed:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & -1 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & 1 \\ \sqrt{2} & 1 & \sqrt{2} \\ \sqrt{2} & -1 & \sqrt{2} \end{bmatrix} \left\{ \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array} \right\} \geq \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right\} \tag{9}$$

Note that these constraints will not strictly keep $v(s)$ positive, but they ensure that $v(s)$ is positive at each of the constituent sinusoids' minima and maxima, and thus will ensure that the path is mostly above the straight line path.

The second are the obstacle constraints—indeed, if there were no obstacle constraint, the optimal path would usually be a straight line. The obstacle constraint is checked such that no part of the path passes within a specified clearance distance of an occupied space. This constraint is highly nonlinear and in general non-convex.

The obstacle constraint defined by a grid-based map of the surrounding terrain features. Each element contains a measure of the probability (specifically the log-odds) that it is occupied. The cells are updated by applying sensor measurements via a probabilistic sensor model as in (Scherer et al., 2008). A simplified version of this mapping method was also developed in an attempt to save memory. In this case, two 2D grids are used; one with the terrain height at that location, and the other with a measure of confidence of the terrain height.

There are a number of factors which we would wish to consider in developing a fitness function for a given set of path parameters: path distance, path curvature, and path altitude are primary among them. Each of these elements is related to the magnitude of the path parameters. Since the fitness function will likely need to be evaluated many times, along with its gradient, an efficient form for it to take is

$$J = \frac{1}{2}\theta^T \begin{bmatrix} Q_L & \\ & \alpha_V Q_V \end{bmatrix} \theta + \alpha_s J_s + \alpha_e J_e \tag{10}$$

$$\theta = [l_1 \; l_2 \; l_3 \; v_1 \; v_2 \; v_3]^T \tag{11}$$

$$J_s = \left\| \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\frac{d\vec{r}_P(s=0)}{ds} \right) \right) \times \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{v}_0 \right) \right\|_2^2 \tag{12}$$

$$J_e = \left\| \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \left(\frac{d\vec{r}_P(s=D)}{ds} \right) \right) \times \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{1}{2} \left(\vec{r}_{A\to B} + \vec{r}_{B\to C} \right) \right) \right\|_2^2 \tag{13}$$

In equations 12 and 13, J_s and J_e are the norm squared of the path direction crossed with the current velocity vector and a vector oriented halfway to the next waypoint (Point C) respectively. These functions encourage the solver to avoid abrupt direction changes at the beginning and end of a flight plan segment. The α_i terms in 10 are positive constants to weight the cost function for particular mission parameters (e.g. to discourage vertical flight under masking conditions.) The Q_i terms are symmetric positive definite matrices which penalizes each harmonic relative to the others.

The optimization technique used is a steepest descent method, seeded by up to seven basic feasible solutions. These starting guesses are found by conducting seven different single direction searches from the origin (straight-line path) until a feasible solution is found.

The constraints are enforced by means of gradient projection. In short, this means that when a search direction is calculated, if there is an active constraint, then the search direction is projected onto the hyperplane defined by that constraint. For the linear constraints, the constrained hyperplane is the same throughout the search space, and analytically defined by a normal vector. The obstacle constraint is locally approximated as a hyperplane in the solution space.

Once an optimal path is found, a set of velocity commands is generated. Given a general twisting and turning obstacle-free path, movement along this path at a constant velocity will cause changes in acceleration due to path curvature. In determining the speed to fly a particular path, maximum speed and acceleration limits are satisfied by the algorithm. These limits may be basic aircraft limits, limits fed back to the algorithm from the inner-loop flight controller, or limits imposed by an operator based on a given mission scenario. It may be desirable to traverse a commanded trajectory slowly for a given mission while very rapidly for another. Here, it is assumed that the dynamic constraints, such as maximum flapping angle, power output, etc. can be mapped to a maximum velocity and a maximum acceleration of the vehicle. These values are known prior to start of flight or fed to the algorithm by the underlying flight controller.

Given the geometric path, the speed shaping algorithm seeks to find a speed profile that traverses the path in the shortest time without violating dynamic constraints. The first step is to find unit tangent vector, unit normal vector, and the curvature are found as a function of s:

$$\vec{t}(s) = \frac{\frac{d\vec{r}(s)}{ds}}{\left\| \frac{d\vec{r}(s)}{ds} \right\|} \tag{14}$$

$$\rho(s) = \left\| \frac{d\vec{t}(s)}{ds} \right\|^{-1} \tag{15}$$

$$\vec{n}(s) = \rho \left(\frac{d\vec{t}(s)}{ds} \right) \tag{16}$$

To compute the speed profile, an arbitrary number of evenly spaced points on the path are sampled. An initial guess for speed at each of the i^{th} samples is selected, typically just greater than zero to ensure that the initial guess does not violate any dynamic constraint. Using that initial guess, time is found as a function of path length.

$$\Delta s_i = \frac{v_i + v_{i-1}}{2} \Delta t_i \tag{17}$$

$$\Delta t_i = 2 \left(\frac{1}{v_i + v_{i-1}} \right) \Delta s_i \tag{18}$$

$$t_i = \sum_{j=1}^{i} \Delta t_j \tag{19}$$

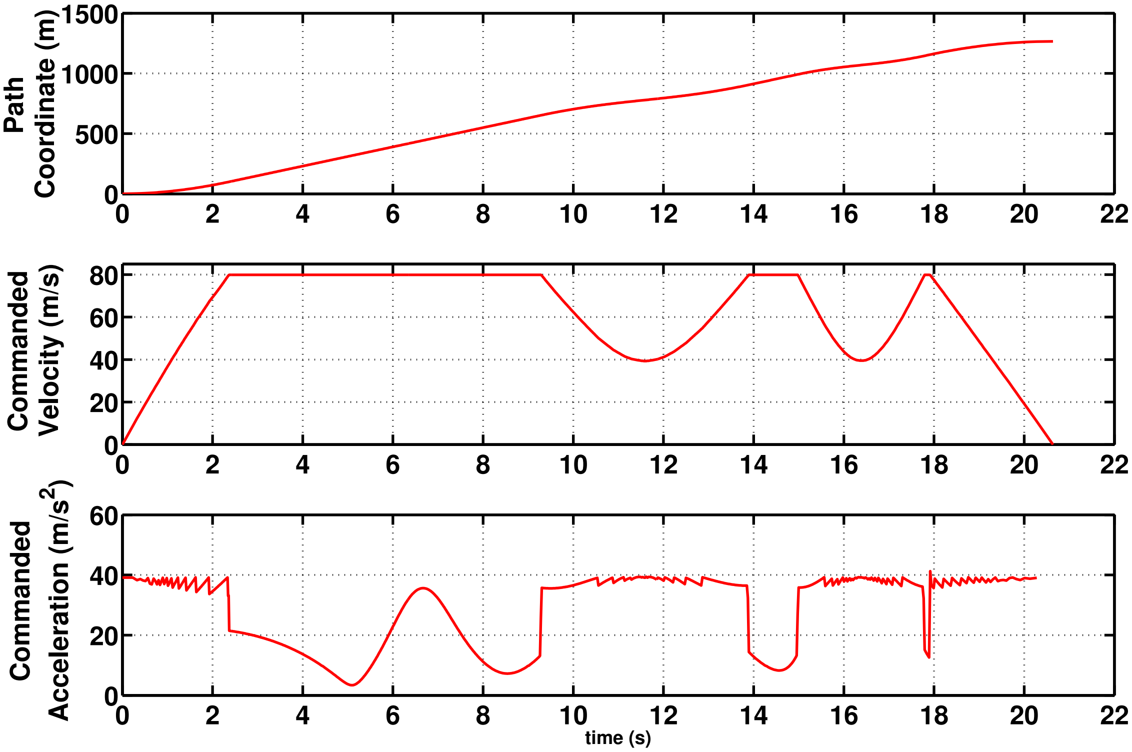


Figure 2: Speed shaping example: aircraft spends considerable time at maximum speed, but must at times decelerate to keep acceleration required to turn within limits

Finally, the non-gravitational acceleration over the curve is found and added to gravity:

$$\vec{a_i} = \frac{\Delta v_i}{\Delta t_i}\vec{t}(s_i) + \frac{v_i^2}{\rho_i}\vec{n}(s_i) + \vec{g} \tag{20}$$

The guess values for $v(s)$ are then iterated point by point using the following logic: Does the point exceed specified constraint conditions? If yes, reduce the velocity at that point. Does the point exceed the overall velocity limit? If yes, reduce the velocity at that point. Does $\|\vec{a}_i\|$ or $\|\vec{a}_{i-1}\|$ exceed acceleration limits? If yes, reduce the velocity at that point. If the answers to the previous questions are no, then increase the velocity of the point. Once all the points on the velocity profile have converged, then every point in the velocity profile has met a constraint and the optimum has been found. Figure 2 shows an example solution.

2.3 Potential-Field Method

Previous research under this partnership focused on the use of artificial potential fields to plan obstacle-free paths, detailed in (Johnson et al., 2011). A summary of this method is included here to aid the comparison of simulation results in Section 4.

Potential fields have been one of the most common approaches to autonomous avoidance. Early research on these methods (Warren, 1989; Khatib, 1986; Krogh and Thorpe, 1986; Warren, 1990a; Warren, 1990b; Koren and Borenstein, 1991) recognized the benefits and inherent limitations of the method. Subsequent research has focused on ways to overcome basic problems, such as local minima, and apply the technique to practical problems (Scherer et al., 2008; Li and Bui, 1998; Ge and Cui, 2002; Park et al., 2001). Both 2D (vertical plane) and 3D versions have been developed. Here, each obstacle is considered a source while the end goal is considered a sink. The aircraft then reacts to pseudo-forces acting on it by the sources and sinks. The method presupposes a control system that can track a desired position, velocity, and acceleration profile.

The path planner finds a smooth, continuous, obstacle-free path from the aircraft's current location to a desired waypoint. The mathematical machinery of potential theory provides a means to this end. In particular, the velocity field of an inviscid fluid flow around a body in the study of aerodynamics holds these characteristics. Generally speaking, artificial potential field techniques formulate this problem by representing the goal point and obstacles as known spatial boundary conditions. The goal point Dirchlet condition on the potential function is set at -1 and the obstacles and space boundaries at 0. This method uses the same obstacle information as detailed in 2.2.

The continuity equation, $\nabla \cdot V = 0$, reduces to the following after making a finite difference approximation in all three directions with an evenly spaced grid ($\Delta x = \Delta y = \Delta z$). Solving for $\phi_{i,j,k}$, one obtains:

$$\phi_{i,j,k} = \frac{\phi_{i+1,j,k} + \phi_{i,j+1,k} + \phi_{i,j,k+1} + \phi_{i-1,j,k} + \phi_{i,j-1,k} + \phi_{i,j,k-1}}{6} \tag{21}$$

In other words, the value of each point in the discrete potential field is equal to the average of the six points around it. This equation is applied to each cell iteratively, using a multigrid technique, to arrive at a solution as in (Scherer et al., 2008).

A straightforward approach to enforce nap-of-the-earth flight in this context is to impose a virtual obstacle in the form of an artificial ceiling and floor into the potential field. To avoid cutting off all feasible paths, we create a blanket region of permissible space around obstacles which overrides the imposed ceiling or floor. A ceiling height is selected via the relationship:

$$h_C = h_{FA} - \kappa(h_{FA} - h_G) \tag{22}$$

where h_C is the ceiling height, h_G is the average height of the ground, and κ is masking factor. κ is set by the operator with a value between 0 to 1, with 0 being no masking and 1 being maximum masking. Note that because of the boundary conditions in the path planning algorithm, there is always a ceiling at the top of the flight area, here denoted as h_{FA}.

Once the array specifying ϕ is found, the same finite difference approximation can be used to calculate the gradient vector at each discrete point in the field. The streamline is then calculated from the vehicle starting position by 4th order Runge-Kutta integration of the gradient vector field using linear interpolation. The trajectory follows the gradient to the point of lowest potential, the goal.

Note that the artificial potential field method provides only a path through the obstacle field but no details on what speed to use. The speed planner method described in Section 2.2 was originally developed for use with the potential field planner as described in (Johnson et al., 2011). However, the process is much less efficient than it is using the motion primitives approach as the unit tangent and unit normal of the curve must be found through finite difference approximations rather than analytically.

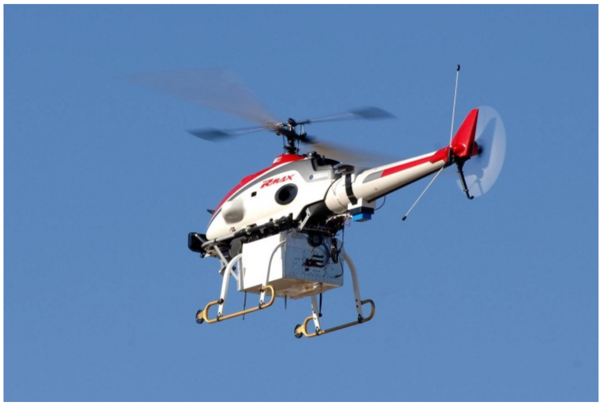


Figure 3: Yamaha RMAX-based research UAV utilized for this effort, 3130 mm main rotor

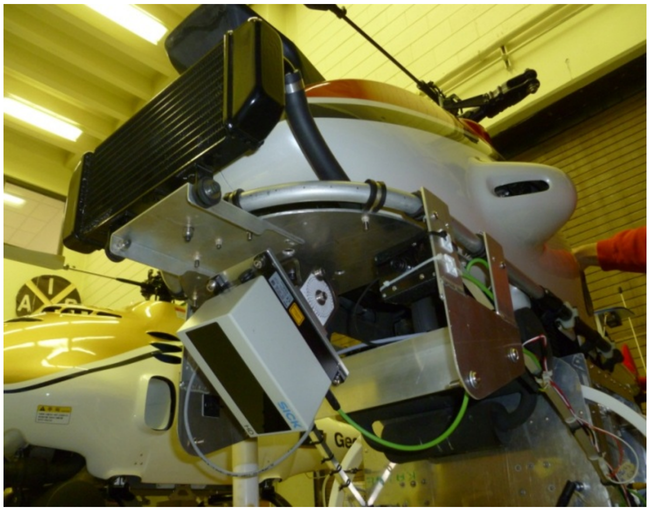


Figure 4: Sick LD-MRS Laser scanner mounted under the nose of the aircraft, able to see down and forward (sensor rotated 90 degrees in roll, 40 degrees nose down pitch)

3 Test Aircraft

A Yamaha RMAX based research UAV, Figure 3, was utilized for the simulation and flight test activities under this effort. The system consists of four major elements: the basic Yamaha RMAX airframe, a modular avionics system, baseline software, and a set of simulation tools.

The hardware components that make up the baseline flight avionics include general purpose processing capabilities and sensing. The research avionics configuration includes:

- 2 Embedded PCs
- Inertial Sciences ISIS-IMU Inertial Measurement Unit
- NovAtel OEM-4, differential GPS
- Sick LD-MRS laser scanner (Figure 4)
- Custom made ultra-sonic sonar altimeter

- Honeywell HMR-2300, 3-Axis magnetometer
- Actuator control interface
- Vehicle telemetry (RPM, Voltage, Remote Pilot Inputs, low fuel warning)
- 11 Mbps Ethernet data link and an Ethernet switch
- FreeWave 900 MHz MHz serial data link

The baseline navigation system running on the primary flight computer is a 17 state extended Kalman filter. The states include: vehicle position, velocity, attitude (quaternion), accelerometer biases, gyro biases, and terrain height error. The system is all-attitude capable and updates at 100 Hz (Dittrich and Johnson, 2002). The baseline flight controller is an adaptive neural network trajectory following controller with 18 neural network inputs, 5 hidden layer neurons, and 7 outputs for each of the 7 degrees of freedom (Johnson and Kannan, 2005). These 7 degrees of freedom include the usual 6 rigid-body degrees of freedom plus a degree of freedom for rotor RPM. The baseline flight controller and navigation system, which coupled with the simple baseline trajectory generator, is capable of automatic takeoff, landing, hover, forward flight up to the maximum attainable by the helicopter (around 26 m/s) and aggressive maneuvering.

To support evaluation of methods in simulation for trade studies and to prepare for flight testing, a detailed simulation model was developed for scanning LADAR systems, allowing several existing off-the-shelf models to be tested in a closed loop simulation environment (Hokuyo UTM-30LX, Sick LMS291-S05, and Sick LD-MRS). Based on factors such as maximum range, weight, power, and field of view, the Sick LD-MRS system was then selected for further development and flight test validation of an automatic NOE flight system on the autonomous RMax described above.

4 Simulation Results

Flight control software was developed utilizing the existing Georgia Tech UAV Simulation Tool (GUST), which is a set of C/C++ software that supports pure software, hardware-in-the-loop, and research flight test operations (Johnson and Schrage, 2004). GUST includes models of the sensors, aircraft, and aircraft interfaces down to the level of binary serial data (i.e., packets). It enables injection of model error and environmental disturbances. It includes a flexible scene generation capability and reconfigurable data communication routines, enabling a large number of possible hardware-in-the-loop simulation configurations.

In addition to simulating the expected flight test environment, both methods were tested in simulation against the simple terrain set of obstacle avoidance benchmarks established by Whalley, et al (Mettler et al., 2010). Although these benchmarks were not expressly intended to be flown on a live aircraft, nor do they directly measure NOE performance, it was deemed important enough to have a basis of comparison with which to analyze the performance of both algorithms.

4.1 Simple Processing of Single Scan Method

The single scan method was tested for a closed course at variety of speeds (6 to 15 m/s)[1] and desired altitudes above obstacles (6 to 15 m). For sake of comparison, only the 12 m/s / 15 m case is shown here; as these correspond to the flight test data also available.

Figure 5 shows a 3D plot of the recorded trajectory for two passes around the simulated closed course. Note that the single obstacle in the path, representing a tree line at the flight test location, results in significant

[1] This research was conducted using US customary units, and as such, many of the numbers used are not exact but rounded to the nearest whole unit or significant digit for convenience.

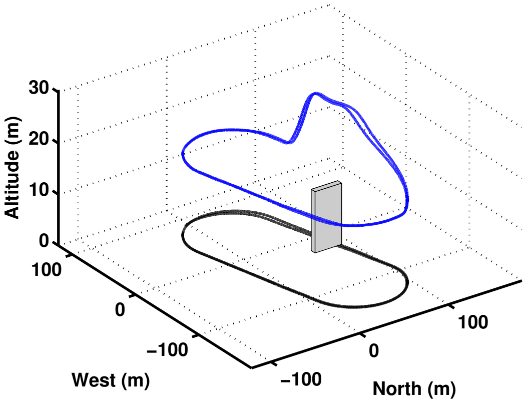


Figure 5: Simulation results with single scan algorithm 12 m/s, desired terrain height of 15 m, for two rounds of a closed course (horizontal projection of path shown on bottom); dominant feature is the avoidance of simulated tree line as the aircraft traverses clockwise in the plot

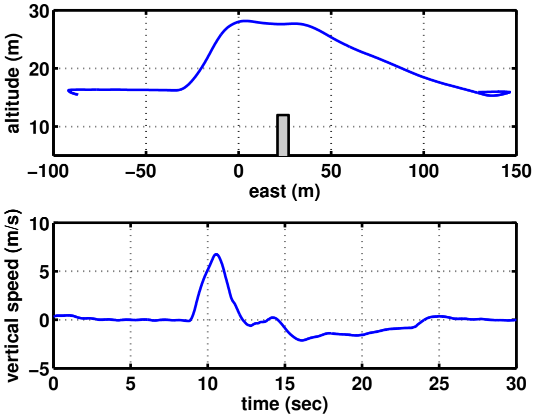


Figure 6: Simulation results with single scan algorithm 12 m/s, desired terrain height of 15 m, close up of pass of simulated tree line, desired descent rate after encounter was 1.52 m/s

flight path alteration, and the specified horizontal and vertical miss distances are satisfied (15 m each). 6 shows the altitude above a reference height (corresponding to approximately the terrain height for most of the field) and vertical speed vs. time profile for one of the passes over the simulated tree line (modeled as a box with appropriate length, width, height, and location).

All six benchmark simple terrain obstacle sets were flown, using similar parameters to the tests above and a mission speed of 3 m/s. Results are shown in Table 1 and Figure 8. The single scan method performed as expected in these tests. There are a few characteristics of the algorithm which these tests highlighted. First, the single scan method provides a viable reactive avoidance algorithm in all conditions tested, as there is nearly always open space available, provided the aircraft can climb high enough. Second, since it only considers vertical clearance, it fails to take advantage of lateral space to maneuver (most clearly shown in Figure 8b).

Finally, the single scan method while in NOE mode had a noteworthy response to benchmarks 5 and 6. The straight-line flight path passes directly over the corners of the two obstacles. Since the laser is fixed to scan in the longitudinal plane of the aircraft, it only intermittently sensed the corners of the obstacles (for

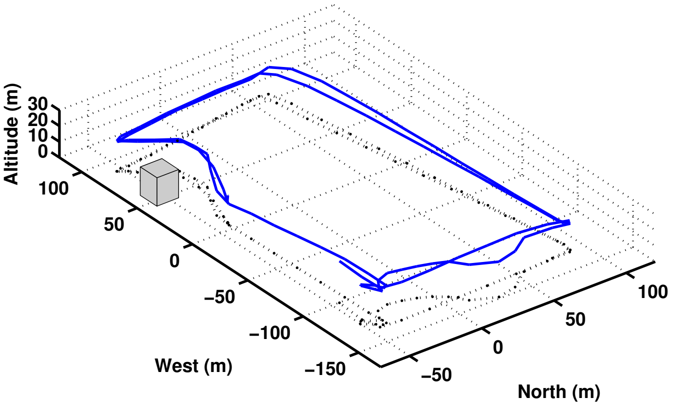


Figure 7: Simulation results with optimization algorithm 6 m/s, desired obstacle clearance of 9 m, for two rounds of a closed course (horizontal projection of path shown on bottom); dominant feature is the avoidance of simulated tree cluster in the southwest corner as the aircraft traverses clockwise in the plot.

example, when the aircraft was slightly nose left, it could not see the northwestern cube in benchmark 6.) Since the single scan method tries to maintain a vertical clearance over the terrain based solely on current sensor information, it would start to descend. This may indicate the need to switch to use of a persistent or otherwise filtered height map for NOE mode rather than raw sensor data.

4.2 Optimization of Motion Primitive Method

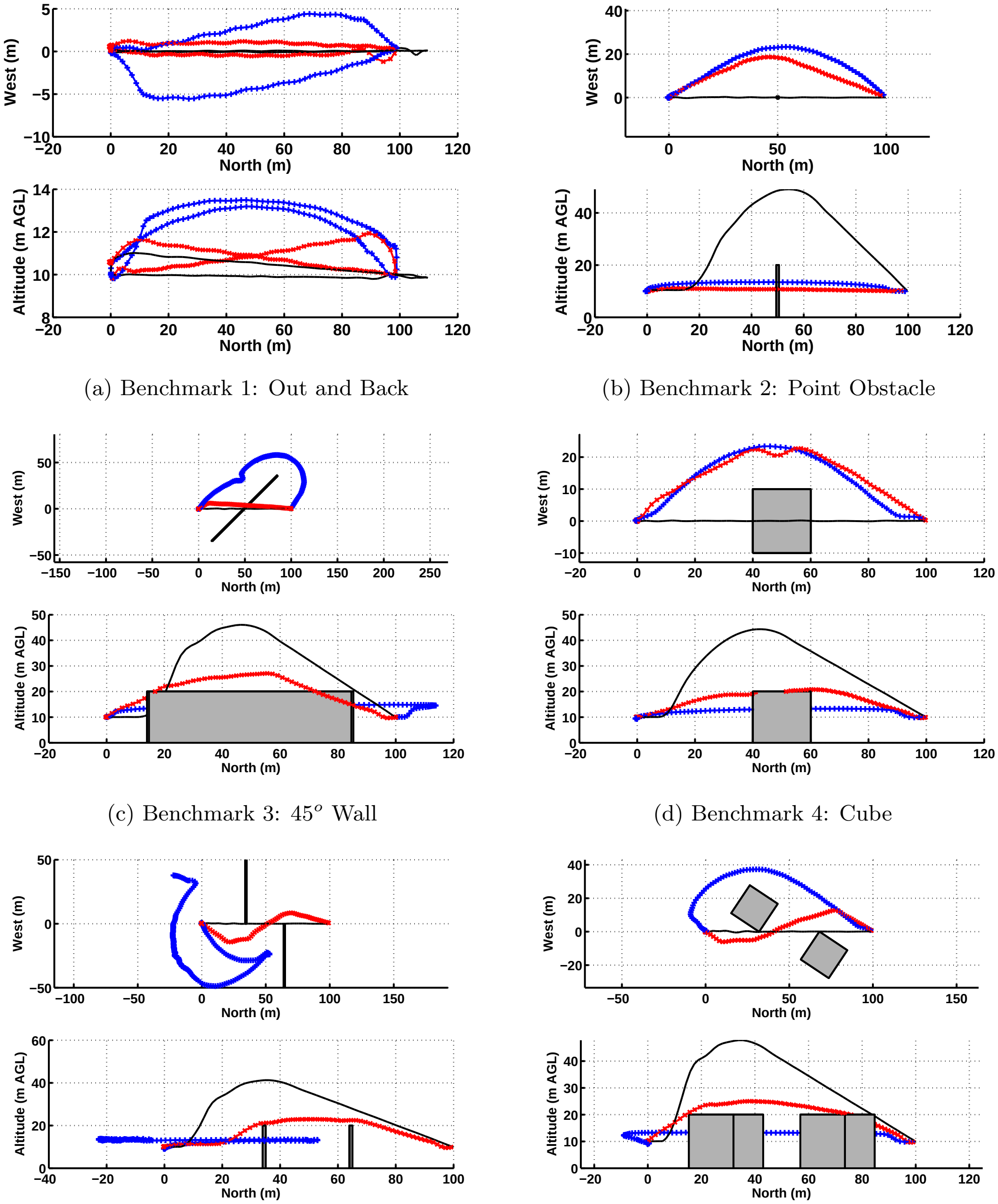
This method was tested in simulation for both open and closed courses from 3 m/s to 15 m/s, with obstacle clearances from 6 m to 15 m, using a variety of parameters to penalize vertical flight (hereafter referred to as the "masked configuration"). The obstacle set was selected to mimic the flight test area at McKenna MOUT site.

Figure 7 shows an example 3D plot of the flight path around a closed course. This course included a single large obstacle in the southwest corner, a small, low obstacle set in the north east corner, and a relatively dense but lower set of obstacles on the east side. The aircraft behaved largely as expected, and avoided obstacles while maintaining clearance, even at a mission speed of 6 m/s.

The optimization method also was tested in simulation against the benchmark obstacle sets. It performed similarly to the baseline and OFN algorithms in (Mettler et al., 2010), but with a climb in the center of the baffle obstacles. Note that the flight times in Table 1 are artificially fast due to overshoot in the speed planner.

4.3 Potential-Field Method

The potential field method was extensively tested in simulation, and results gained using a model of McKenna MOUT site were reported in (Johnson et al., 2011). Additional simulations were conducted using the benchmark tests from (Mettler et al., 2010) in order to compare the performance of the three NOE flight methods.



(a) Benchmark 1: Out and Back

(b) Benchmark 2: Point Obstacle

(c) Benchmark 3: 45^o Wall

(d) Benchmark 4: Cube

(e) Benchmark 5: Wall Baffle

(f) Benchmark 6: Cube Baffle

Figure 8: Benchmarking Results: single scan algorithm—straight line, potential field algorithm—denoted by the "+" (80% masking), optimization of maneuver primitives denoted by "×" (masking configuration)

Table 1: Simulation Results, Simple Terrain Set Benchmarks (flight times in s)

		Single Scan	Potential Method	Optimization of Maneuver Primitives	Published Benchmark
	Benchmark	Avoid Mode	80% Masking	Masked configuration	
1	Out and Back	82.28	76.2	68.92	78.8
2	Point Obstacle	42.46	40	35.54	39.3
3	45^o Wall	41.4	65.32	38.56	39.3
4	Cube	41.56	39.02	38.76	42.1
5	Wall Baffle	41.28	Algorithm Failed	38.5	41.7
6	Cube Baffle	43.28	51.54	37.92	39.8

Table 2: Weather Conditions at Fort Benning, GA (MFBGG1)

Parameter	Day 1	Day 2	Day 3
Mean Temperature (℃)	15.4	13.8	16.0
High Temperature (℃)	17.2	20.6	24.4
Mean Wind Speed (m/s)	3.0	1.2	2.1
Max Steady Wind Speed (m/s)	5.4	2.2	4.5
Max Gust Wind Speed (m/s)	8.9	4.5	8.9
Precipitation (mm)	22	0.25	0.0

The benchmark tests revealed three main shortcomings of the potential field method. First, the availability of open potential space above the aircraft resulted in a climb response to every obstacle, especially in the cases with zero masking factor. Second, when the sensor measured an obstacle that it had not previously been able to see, the resulting path was unpredictable especially at higher masking factors (particularly those shown in Figures 8c and 8e). Finally, the aircraft tends to fly unnecessarily wide, especially when it has relatively complete obstacle information.

5 Flight Test Results

A series of flight tests were conducted in late January 2011 and 2012 at McKenna MOUT site, Fort Benning, Georgia. The test goals were to certify the performance of both the single scan method of avoidance, the potential field method, and the optimization of motion primitives algorithm. Tests were all conducted in daylight with marginal to fair conditions. Weather conditions in the area of the test site are summarized in Table 2[2].

5.1 Laser Scanner Performance in Degraded Conditions

The Sick LD-MRS was tested in this flight test and several previous flight tests in degraded conditions which affected the performance of the sensor. The three conditions of note include dust, rain, and low-angle direct sunlight. Dust and rain generated false returns which negatively impacted the ability of the laser to accurately describe the terrain. The avoidance algorithms acted to evade a phantom obstacle, and in some cases, one that persisted to the immediate front of the aircraft. These two conditions, however, caused relatively diffuse false returns, and ultimately could be compensated for with built-in filters that could be turned on in software.

A more interesting development was the discovery of sensitivity in the laser scanner to direct sunlight. On several occasions, each of the planning and avoidance algorithms suddenly and spontaneously commanded

[2] Weather data courtesy Weather Underground, http://www.wunderground.com/

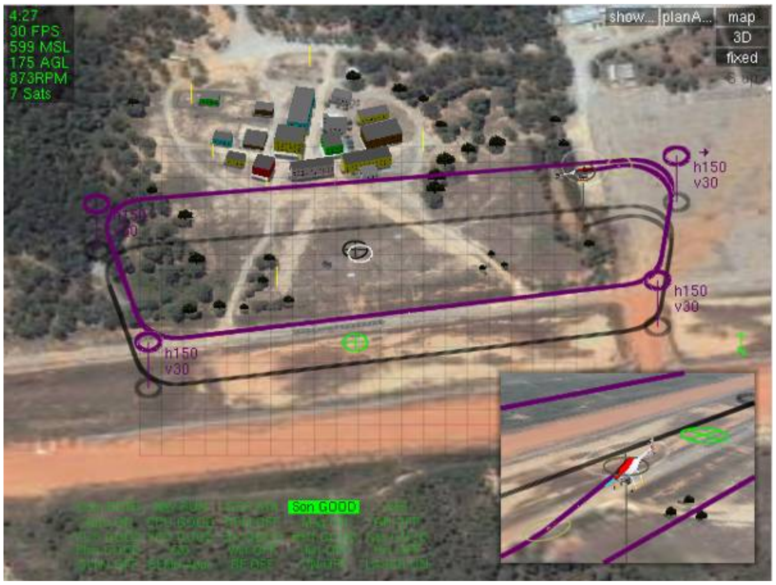


Figure 9: 3D view of ground path for first test of single scan algorithm (South up)

erratic paths, prompting mission abort or recovery by the safety pilot. After several instances, it was discovered that the common element in these events was the laser being pointed directly at the sun, typically in the early morning or late afternoon. Subsequent direct testing of the installed laser has resulted in a well-characterized model of this error Direct, non-occluded sunlight results in the laser seeing a cone of noisy false returns in the direction of the sun, which has the effect of making the avoidance algorithms react to phantom obstacle at situated at point blank range. Strategies to reject these errors are being pursued.

5.2 Simple Processing of Single Scan Method

The single scan algorithm was tested over multiple closed courses at the McKenna MOUT site. Results are summarized here for two representative tests. The ground path of the first is shown below in Figure 9. Commanded altitude was reduced to 15 m above all points within 15 m of desired path and up to a speed of 15 m/s. The path itself goes clockwise in the figure, encountering two long flat stretches in the center, and two tree lines, the taller on the left. The terrain height is also somewhat higher on the upper side of the image.

Figure 10 shows the altitude profile above a convenient reference height. Figure 11 shows the total speed with respect to the Earth. Figure 12 shows collective pitch angle. At 30 seconds, the speed is increased to 12 m/s. Throughout, the commanded height is 15 m above objects within 15 m of the path. From 5 to 50 seconds, the aircraft is traveling over the lower part of Figure 9 (the runway). The terrain here is somewhat lower that the reference height, and so the recorded altitude of between 9 m and 12 m is expected and corresponds to the expected 15 m above the local terrain. At 50 seconds, the aircraft turns sharply right and encounters both an increase in terrain height and a tree line. Over the next several seconds it rapidly climbs approximately 15 m to maintain the desired terrain and obstacle clearance. This saturates the collective, as shown in Figure 12. The onboard camera view at approximately 50 seconds is shown in Figure 13. A second smaller tree line is encountered approximately head on just after 80 seconds, shown in Figures 10 (altitude profile) and 14 (onboard camera).

A second test involved flying directly over the building in the McKenna MOUT village in a similar profile. This time the commanded speed was 9 m/s and the commanded height was 23 m above objects within 15 m horizontally. Altitude profile is shown in Figure 15, speed in Figure 16, and collective pitch in Figure 17.

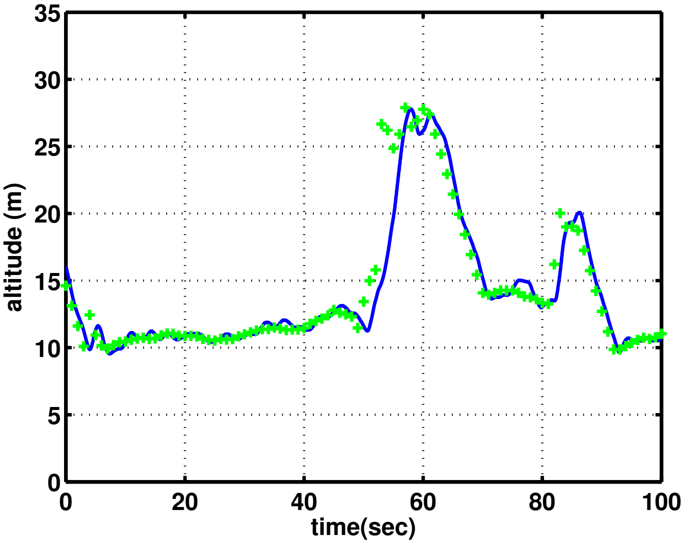


Figure 10: Altitude profile for single circuit of closed course, pattern starts to repeat at approximately 90 seconds. Green dots are recorded commanded position at 1 Hz. Blue best estimate of aircraft height above reference.

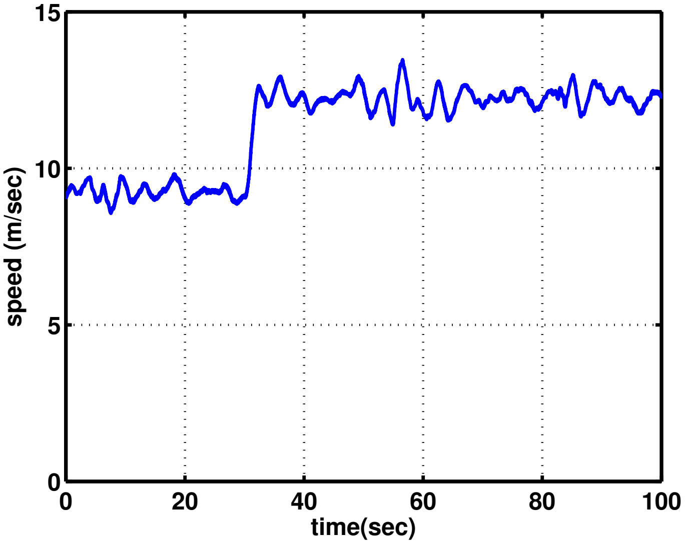


Figure 11: Speed with respect to the Earth during single closed course, set to 12 m/s during both enters with significant terrain features

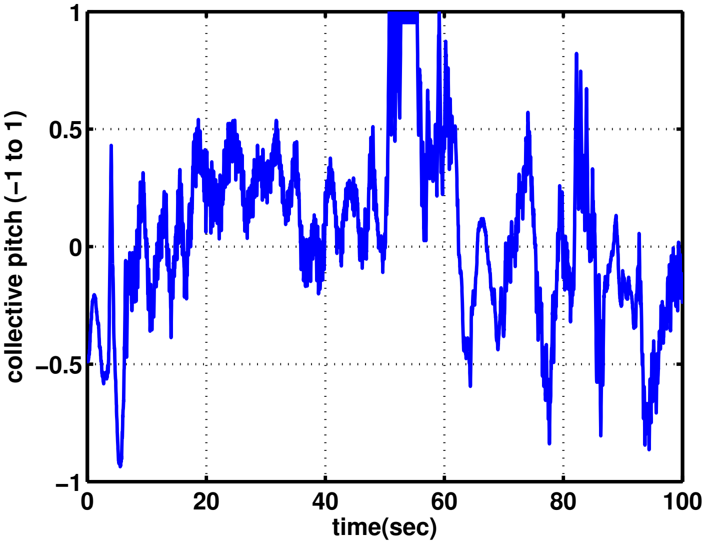


Figure 12: Collective pitch angle during a single circuit of closed course at 12 m/s per second. Note saturation from 50 to 55 seconds

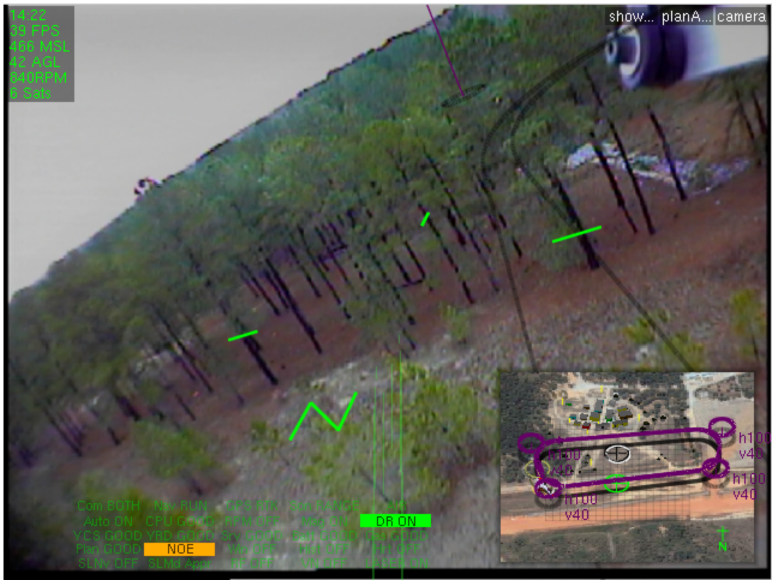


Figure 13: Onboard camera view as aircraft encounters increase in terrain height and tree line during turn

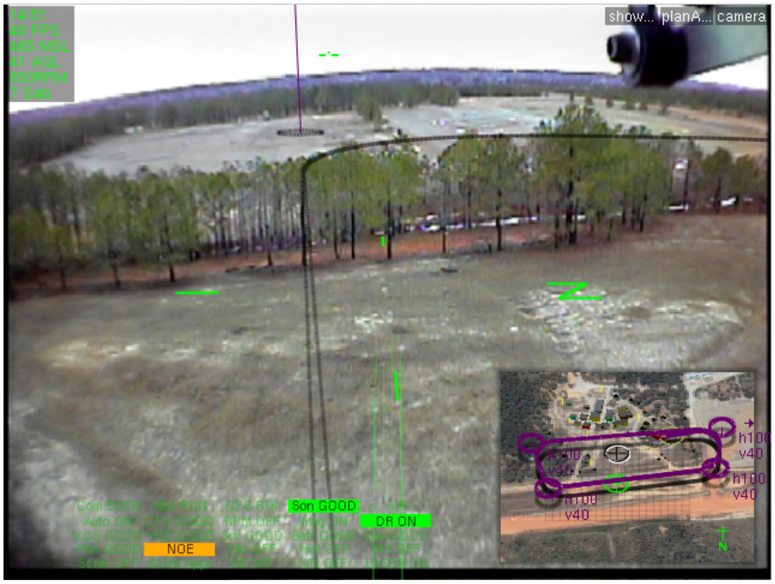


Figure 14: Onboard camera view as aircraft approaches smaller tree line

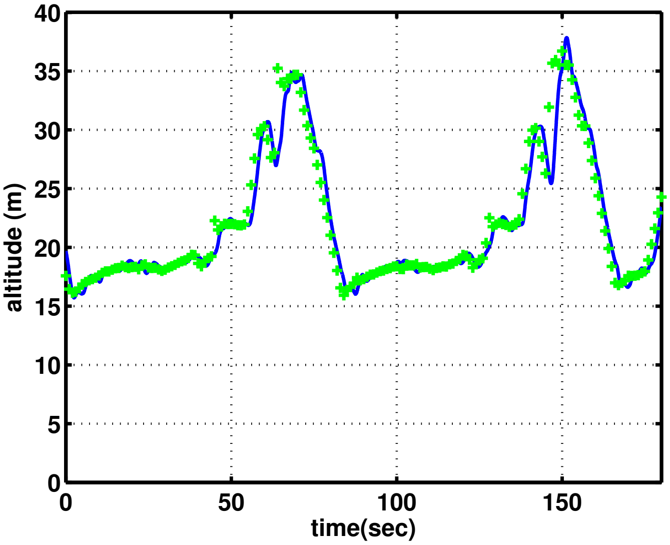


Figure 15: Altitude profile for two repeated circuits over buildings and trees at 9 m/s per second

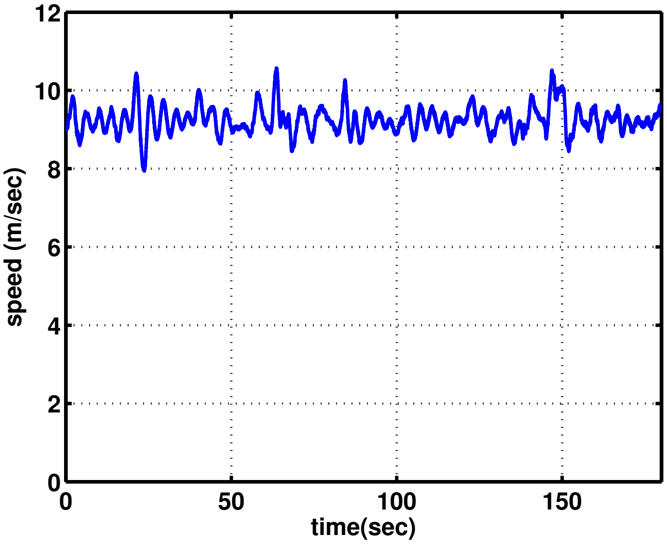


Figure 16: Speed profile for two repeated circuits over buildings and trees

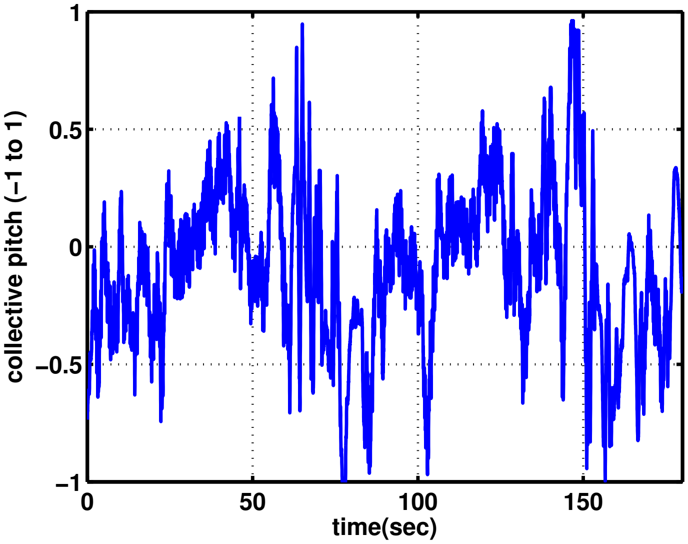


Figure 17: Collective pitch angle for two repeated circuits over buildings and trees

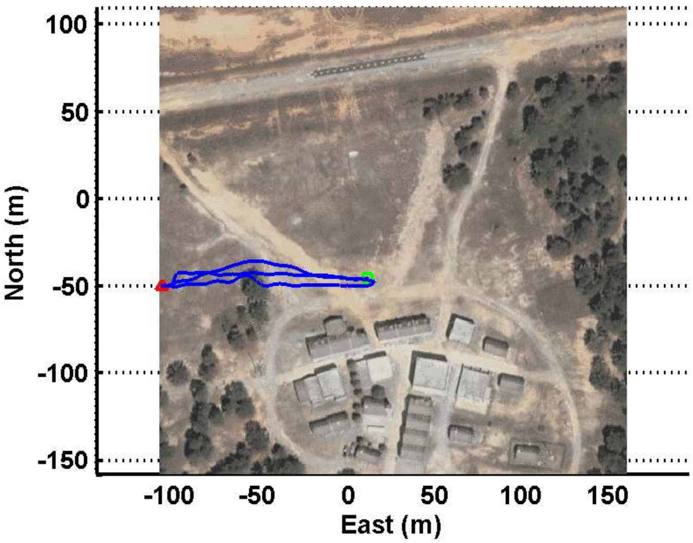


Figure 18: Flight test results with optimization algorithm 3 m/s, desired obstacle clearance of 3 m, for three passes against a single obstacle; starting position marked by green square, final position by red triangle.

Two complete circuits of the closed course are shown. The dominant feature is approaching the village from the North across an open field, and the aircraft rapidly climbs at 55 and 135 seconds provide selected clearance from the buildings. It climbs again at 65 and 145 to avoid even taller trees as in proceeds east out of the village area.

Over the two tests presented here and other tests, which included light rain, wind gusts over 8 m/s, and ground speeds of up to 15 m/s, it can be stated that the overall system behaved as expected from the simulation. Additional work is recommended to investigate maximizing the performance of the sensor while rejecting sensor errors, and then provide a basis to slow the aircraft down when the aircraft does not have sufficient climb performance to maintain specified miss distance, as was evident in Figure 12, where the collective pitch was saturated briefly.

5.3 Optimization of Motion Primitive Method

The motion primitives method was tested in three distinct scenarios. The first was a set of multiple passes against a singular obstacle at 3 m/s. The aircraft initially selected a route to the side, then on subsequent passes over, in general agreement with simulation results (Figure 18).

The second set of tests included a pair of flights around closed circuit course, again at 3 m/s (Figures 19 and 20). This flight resulted in a pair of anomalous maneuvers that ultimately required intervention by the safety pilot. The closed circuit course was flown both clockwise and counterclockwise and the anomalies occurred in different parts of the flight paths, but while the aircraft was pointing in the same direction, the azimuth to the sun. These faults were confirmed to be a result of faulty returns from direct sunlight shining in the laser.

The final series of tests was a series of passes near a relatively dense obstacle field, first at 4.5 m/s, then 6 m/s, with the obstacle map periodically reset (Figure 21). The algorithm behaved as predicted by simulation, with the exception of another anomaly that occurred as the aircraft pointed toward the sun.

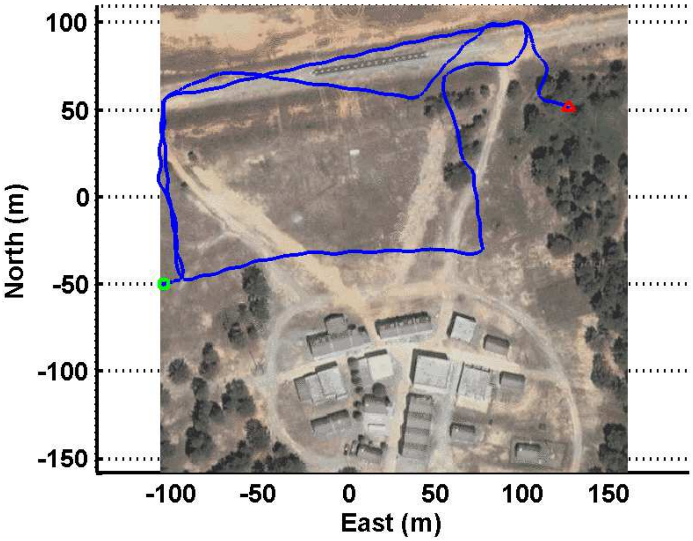


Figure 19: Flight test results with optimization algorithm 3 m/s, desired obstacle clearance of 9 m, for closed course; unusual flight path in northeastern corner caused by sensor error from direct sunlight; starting position marked by green square, final position by red triangle

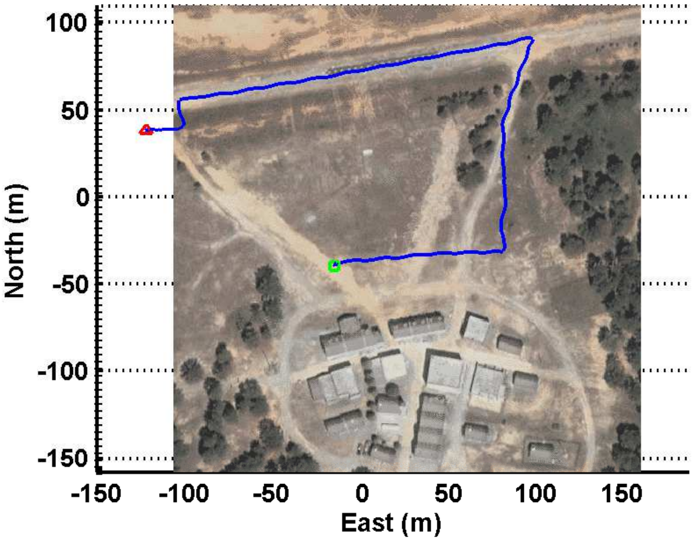


Figure 20: Flight test results with optimization algorithm 3 m/s, desired obstacle clearance of 9 m/s, for closed course; unusual flight path in northwestern corner caused by sensor error from direct sunlight; starting position marked by green square, final position by red triangle

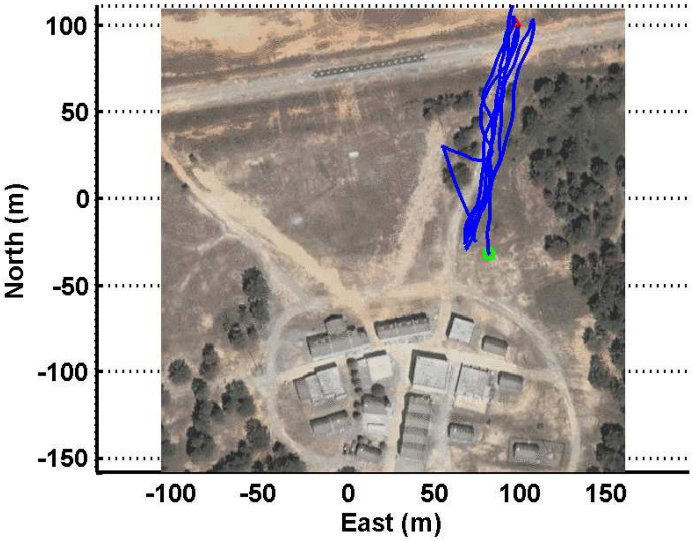


Figure 21: Flight test results with optimization algorithm 4-5.6 m, desired obstacle clearance of 9 m; one unusual flight path on southern end caused by sensor error from direct sunlight; starting position marked by green square, final position by red triangle

6 Conclusions

The efforts described in this paper include: (1) Flight testing of and installed ranging sensor, specifically the Sick LD-MRS; (2) Hardware-in-the-loop Simulation studies based on achieved sensor performance utilizing three methods for generating the desired NOE flight path; (3) Comparison of the performance of the systems against published benchmarks and (4) Flight testing of closed loop system performing autonomous unmanned NOE flight. Flight test results verify the effectiveness of the installed sensor, and validate the simulation results of the simpler algorithm up to 15 m/s at 6 m of terrain clearance. Future work includes extending these algorithms to enable cooperative guidance among multiple vehicles and other, more complex scenarios.

Acknowledgments

The authors would like to acknowledge support from Sikorsky Aircraft Corporation and the Georgia Institute of Technology, and the help of other contributors including: Jeong Hur, Brian McCabe, Russ Halstead, Suresh Kannan, Keeryun Kang, Chester Ong, Daniel Magree, Eohan George, and Claus Christmann.

References

Cheng, V. and Lam, T. (1994). Automatic guidance and control for helicopter obstacle avoidance. *Journal of Guidance, Control, and Dynamics*, 17(6).

Cowherd, C. (2007). Sandblaster 2 Support of See-Through Technologies for Particulate Brownout Task 5 Final Technical Report. Technical Report 110565, DARPA, Kansas City, MO.

Dittrich, J. S. and Johnson, E. N. (2002). Multi-sensor navigation system for an autonomous helicopter. *Proceedings of the 21st Digital Avionics Systems Conference.*

Frazzoli, E. (2001). *Robust hybrid control for autonomous vehicle motion planning.* PhD thesis, Massachusetts Institute of Technology.

Ge, S. S. and Cui, Y. J. (2002). Dynamic motion planning for mobile robots using potential field method. *Autonomous Robots*, 13(3):207–222.

Johnson, E. N. and Kannan, S. K. (2005). Adaptive Trajectory Control for Autonomous Helicopters. *AIAA Journal of Guidance, Control, and Dynamics*, 28(3):524–538.

Johnson, E. N., Mooney, J. G., Sahasrabudhe, V., and Hartman, J. (2011). Flight Testing of Nap-of-the-Earth Unmanned Helicopter Systems. In *AHS Annual Forum*.

Johnson, E. N. and Schrage, D. P. (2004). System Integration and Operation of a Research Unmanned Aerial Vehicle. *AIAA Journal of Aerospace Computing, Information, and Communication*, 1(1):5–18.

Kandil, A. A., Wagner, A., Gotta, A., and Badreddin, E. (2010). Collision Avoidance in a Recursive Nested Behaviour Control Structure for Unmanned Aerial Vehicles. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 4276–4281.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98.

Koren, Y. and Borenstein, J. (1991). Potential field methods and their Inherent Limitations for Mobile Robot Navigation. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 1398–1404. IEEE.

Krogh, B. and Thorpe, C. E. (1986). Integrated path planning and dynamic steering control for autonomous vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 1664–1669. Institute of Electrical and Electronics Engineers.

LaValle, S. M. (2006). *Planning Algorithms.* Cambridge University Press, Cambridge.

Li, Z. X. and Bui, T. D. (1998). Robot path planning using fluid model. *Journal of Intelligent & Robotic Systems*, 21(1):29–50.

Mettler, B., Kong, Z., Goerzen, C., and Whalley, M. (2010). Benchmarking of Obstacle Field Navigation Algorithms for Autonomous Helicopters. In *American Helicopter Society Annual Forum*.

Ortiz, A. E. and Neogi, N. N. (2007). Object Detection and Avoidance Using Optical Techniques in Uninhabited Aerial Vehicles. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, number August, pages 1–13.

Park, M., Jeon, J., and Lee, M. (2001). Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing. In *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, volume 3, pages 1530–1535. Ieee.

Scherer, S., Singh, S., Chamberlain, L., and Elgersma, M. (2008). Flying Fast and Low Among Obstacles: Methodology and Experiments. *The International Journal of Robotics Research*, 27(5):549–574.

Schouwenaars, T., Mettler, B., Feron, E., and How, J. P. (2003). Robust motion planning using a maneuver automation with built-in uncertainties. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 3, pages 2211–2216. Ieee.

Shim, D. H., Chung, H., Kim, H. J., and Sastry, S. (2005). Autonomous exploration in unknown urban environments for unmanned aerial vehicles. In *AIAA Guidance, Navigation, and Control Conference*, number August, pages 1–8.

Tsenkov, P., Howlett, J. K., Whalley, M., Schulein, G. J., Takahashi, M., Rhinehart, M. H., and Mettler, B. (2008). A system for 3d autonomous rotorcraft navigation in urban environments. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, pages 1–23, Reston, Virigina. American Institute of Aeronautics and Astronautics.

Warren, C. W. (1989). Global path planning using artificial potential fields. In *IEEE International Conference on Robotics and Automation*, pages 316–321.

Warren, C. W. (1990a). A technique for autonomous underwater vehicle route planning. *IEEE Journal of Oceanic Engineering*, 15(3):199–205.

Warren, C. W. (1990b). Multiple robot path coordination using artificial potential fields. *Robotics and Automation, 1990. Proceedings.*,, pages 500–505.

Zufferey, J.-C., Beyeler, A., and Floreano, D. (2010). Autonomous flight at low altitude with vision-based collision avoidance and GPS-based path following. In *IEEE International Conference on Robotics and Automation*, pages 3329–3334.