

DETC2008-49820

DRAFT: TOWARDS SCALED DESIGNING AND TESTING OF UNMANNED AERIAL VEHICLE MISSIONS

Keith W. Sevcik

Drexel Autonomous Systems Laboratory
Department of Mechanical Engineering
Drexel University
Philadelphia, Pennsylvania 19104
Email: kws23@drexel.edu

Paul Y. Oh*

Drexel Autonomous Systems Laboratory
Department of Mechanical Engineering
Drexel University
Philadelphia, Pennsylvania 19104
Email: paul@coe.drexel.edu

ABSTRACT

Today's UAVs are being tasked to fly missions in increasingly difficult environments. Buildings, trees and thin wires form challenging terrain for the UAV to negotiate. The current paradigm of UAV research typically moves from computer simulation and individual subsystem testing to full integration in the field. The reactions of control algorithms to realistic sensor data are difficult to capture in simulation and can result in costly crashes. This paper introduces a methodic approach to developing UAV missions. A scaled down urban environment provides a facility to perform testing and evaluation (T&E) on control algorithms before flight. A UAV platform and test site allow the tuned control algorithms to be verified and validated (V&V) in real world flights. The resulting design methodology reduces risk in the development of UAV missions.

1 INTRODUCTION

The robotics community is faced with an ever increasing demand for robots that operate in cluttered outdoor environments. To perform tasks such as search and rescue and surveillance, the robots must operate in unstructured, dynamic environments. The nature of these environments often drives development to focus on sensing and control algorithms.

The current design paradigm begins with laboratory development and testing. Sensors are characterized in sterile, struc-



Figure 1. A SCALED MODEL ENVIRONMENT FOR TESTING UAV MISSIONS. THINGS SUCH AS TREES AND UNSTRUCTURED LIGHTING THAT ARE DIFFICULT TO CAPTURE IN COMPUTER SIMULATION ARE EASILY INCORPORATED HERE.

tured environments with the claim that the results are extensible to real life objects and conditions. While it is true that rigorous testing such as that presented in [1] and [2] helps one to understand the limitations of hardware, it is difficult to determine how the sensor and sensing algorithms will perform in unpredictable field conditions.

Similarly, computer simulations aid in the design of control algorithms. As the control code is refined, greater and greater detail can be incorporated into the model to approximate real world conditions. [3] investigated methods for simulating en-

*Address all correspondence to this author. This work is funded in part by the National Science Foundation CAREER award IIS 0347430.

environmental conditions such as wind gusts. Sensors have also been incorporated into computer simulation, as shown in [4]. However, present day computer models are unable to incorporate unstructured environments. In particular, objects such as trees and bushes are exceedingly difficult to accurately integrate into simulation.

Following lab development, the sensing hardware and control software are transferred to the robotic platform in order to perform real world tests. Many times the first test of the integrated sensing hardware and control software occurs in the field during these flights.

This design methodology invites costly, time consuming failures. Errors in programming, unforeseen design challenges, and unpredictable real world conditions lead to catastrophic crashes. To mitigate these risks, we propose a step in between lab development and real world flights where sensing and control can be tested and evaluated without having to fly the robotic platform.

The authors' previous work in this area [5] involved a full scale mock urban environment inside a 6 degree of freedom gantry. Sensor suites attached to the end effector of the gantry could be virtually flown through the environment. The motions of the gantry were governed by a high fidelity math model of the robotic platform. This allowed hardware-in-the-loop testing of the robot's sensing and control algorithms.

This approach proved useful for evaluating the robot's reactions to varying environmental conditions. However, physical limitations confined the testing area to a small slice of the urban environment. This limited testing to low-speed, low altitude maneuvers. While this technology could be implemented on a larger scale, it was ultimately unfeasible to evaluate entire missions, which could occur over several thousand square meters.

To solve these issues, inspiration was drawn from the early development of flight simulators. As described in [6], some of the first flight simulators utilized scaled models of terrain to provide visual feedback for pilots. These systems provided high fidelity, realistic visual cues for pilots. However, simulations were limited to the area of the models. This approach was abandoned in favor of computer based simulators which provided endless terrain maps, albeit at the sacrifice of realism.

The problem faced by simulation of UAV missions is quite the opposite. Missions are often confined to a defined region such as a town or group of buildings. Computer simulations attempt to model real world effects, but fail to capture the caveats of operating in real world environments. Scaled models such as that shown in Fig. 1 provide a means to test sensors and control algorithms against realistic environments.

This paper presents the design of a testing facility for UAV missions and its use to guide the development of a robotic helicopter. Section 2 describes the facility and its integration into the design process. Section 3 describes the robotic platform.



Figure 2. SATELLITE IMAGE OF THE HELICOPTER DEVELOPMENT RANGE AT PIASECKI AIRCRAFT. THE COMPLEX CONTAINS TYPICAL TERRAIN FOR UAV MISSIONS, SUCH AS URBAN AND WOODED ENVIRONMENTS. THE AREA SPANS SEVERAL HUNDRED METERS, ALLOWING AMPLE ROOM FOR UAV FLIGHT TESTS.

Section 4 describes the mission and algorithms being tested in the facility. Section 5 describes experimental results to date. Finally, conclusions and future work are presented in Section 6.

2 TESTING FACILITY

The goal of this research is to introduce a more sound design methodology to the field of UAV research. Through testing and evaluation (T&E), sensors and control algorithms can be tuned before flight. The refined hardware and software can then go through verification and validation (V&V) on board the actual robotic system. This affords a more robust end product and better management of risk during development. To guide the design of these T&E and V&V setups, the mission profiles must first be defined.

The types of missions under investigation are those typically executed by UAVs on-station after being deployed from a remote location. Such missions include reconnaissance, perch-and-stare and payload delivery. Many of these missions require investigation of more fundamental capabilities such as autonomous mapping and landing zone identification. Areas of interest are typically urban environments containing obstacles such as buildings, poles, trees and thin wires.

Such missions have been investigated in both [7] and [8]. In these experiments, the operational area was as large as $220m \times 220m$ flown at altitudes in the 10's of meters. The craft in these missions traverse the environment at speeds ranging from



Figure 3. THE V&V ENVIRONMENT COMPARED TO THE 1/87TH SCALE T&E ENVIRONMENT. THE T&E FACILITY WAS CREATED TO CLOSELY APPROXIMATE THE HELICOPTER DEVELOPMENT RANGE IN ORDER TO DRAW A CONTINUOUS PATH BETWEEN LABORATORY TESTING AND REAL-WORLD FLIGHTS.

4 – 10m/s. Furthermore, the ascent velocity in [7] is limited to 3m/s while the descent velocity is limited to 1m/s. These requirements are compiled in Table 1. From these criteria, the V&V environment selected by the authors was the helicopter development range at Piasecki aircraft. As can be seen in the satellite photo in Fig. 2, the area encompasses several hundred meters. Buildings and wooded regions provide a variety of terrain to test UAVs.

The focus of this design methodology is to create a continuous path from laboratory research to real world flights. The transition from T&E to V&V should therefore be as seamless as possible. As such, the T&E environment was created to closely approximate the Piasecki facility, as shown in Fig. 3. The facility was recreated at 1/87th scale, which is a common modeling scale. This permits access to a wide range of obstacles and terrain features which can be added in the future.

Assessment of UAV control algorithms required a testing facility capable of repeatable and controllable simulation of UAV dynamics and flight paths. The Systems Integrated Sen-



Figure 4. SYSTEMS INTEGRATED SENSOR TEST RIG (SISTR). SISTR PROVIDES A STAGE FOR TESTING AND EVALUATING SENSOR AND CONTROL ALGORITHMS IN A SCALED ENVIRONMENT. THE 6 DOF GANTRY THAT COMPRISES SISTR CAN BE PROGRAMMED THROUGH MODEL ADAPTIVE CONTROL TO MIMIC THE FLIGHT OF UAVS.

Table 1. CONSTRAINT VELOCITIES

Axis	Gantry	Scaled	Mission Required
X	0.012 - 0.61m/s	1.04 - 53.0m/s	4 - 10m/s
Y	0.019 - 0.61m/s	1.65 - 53.0m/s	4 - 10m/s
+Z	0.030 - 0.61m/s	2.61 - 53.0m/s	0 - 3m/s
-Z	0.101 - 0.61m/s	8.79 - 53.0m/s	0 - 1m/s

sor Test Rig (SISTR), shown in Fig. 4, is a National Science Foundation funded UAV testing facility that provides this capability. SISTR measures 19ft x 18ft x 20ft enclosing the scaled T&E environment.

As described in [5], the facility is surrounded by a 6 degree-of-freedom (DOF) computer controlled gantry. Using the math model of the UAV and model adaptive control, the gantry can be programmed to mimic the flight of an aerial vehicle. UAV sensor suites can be attached to the end effector of the gantry to provide real-time sensor feedback for testing sensor and control algorithms.

In mimicking the flight of a UAV, one of the most important design factors is that the velocities of the UAV can be accurately matched in the scaled down model. To accomplish this, the translational motions of the gantry must scale appropriately to fall within the operational velocity ranges of the UAV. Table 1 displays the maximum and minimum velocities achievable by the gantry, the scaled values of those velocities, and the corre-



Figure 5. THE SR100 HELICOPTER FROM ROTOMOTION, INC. THE SR100 IS SOLD AS A FULLY ROBOTIC PACKAGE CAPABLE OF AUTOMATED TAKE OFF, LANDING, AND GPS WAYPOINT FOLLOWING.

sponding required mission velocities. As can be seen, the velocity ranges required for the X-axis and Y-axis are easily achieved by the gantry. However, the Z-axis velocities of the gantry are faster than those required by the mission. This issue exists under the current software solution for controlling the gantry. The authors believe the gantry hardware is capable of achieving slower motions. This issue is currently being addressed by the authors.

Finally, the position in all translational axes of the gantry can be controlled to within $\pm 1cm$. This scales up to a resolution of $\pm 0.87m$. This position accuracy is well within the $\pm 2m$ accuracy of the typical GPS system. This provides a complete facility which can accommodate T&E of sensor and control algorithms for many different UAV platforms. To show the validity of this approach, the authors use a specific robotic system to show the complete design process incorporating T&E and V&V.

3 ROBOTIC PLATFORM

To perform V&V, a Rotomotion SR100 electric UAV helicopter was used, shown in Fig. 5. The SR100 is sold as a fully robotic helicopter capable of performing autonomous take off, landing, and GPS waypoint navigation when controlled from a laptop base station. Control from the base station to the helicopter is routed through an 802.11 wireless network adapter.

The SR100 has a rotor diameter of $\pm 2m$ allowing it to carry a payload of up to $\pm 8kg$. For these experiments, we outfitted the helicopter with custom landing gear, a custom camera pan/tilt unit, the SICK LMS200, a serial to Ethernet converter, and two $\pm 12V$ batteries for payload power. In total we added approximately $\pm 7kg$ of payload. This greatly reduces the flight time, which is up to 45 minutes without a payload.

The biggest attraction of this platform, however, is the fact that it is already outfitted with all of the necessary sensors to

calculate its pose. Gyros, an inertial measurement unit, and a magnetometer provide the craft's attitude and heading. This information is fused with a Novatel GPS system to provide position data. The position is reported as Cartesian coordinates relative to a global frame, whose origin is at the location where the helicopter was activated.

In selecting hardware to perform initial tests, the authors looked to previous experience designing UAV sensor suites. As a Future Combat Systems (FCS) One team member, the authors have gained extensive experience designing sensor suites for robots flying in near-Earth environments. The FCS Class II program focused on building a UAV to fly missions in areas such as urban terrain and forests. This project identified a few fundamental requirements for these sensor suites.

The sensor must detect a wide range of obstacles. In urban terrain, object size and composition can vary drastically, from buildings to telephone poles to thin wires and clothes lines. In particular, sparse objects such as trees and bushes are troublesome to detect.

The sensor must also be able to detect obstacles from far away and at oblique angles. The speed that a UAV can travel at is directly related to how far away it can detect obstacles. The greater the detection distance, the more time the UAV has to react and plan a new flight path.

These experiences in sensor suite design revealed that scanning laser range finders are the best suited sensor to meet these criteria. Preliminary experiments against the criteria stated above showed them to outperform common sensors such as sonar, computer vision and optic flow.

The biggest attraction of these sensors is their high fidelity and wide field of view. Their range is comparable if not better than many traditional sensors. Laser range finders are also able to clearly detect many different objects including sparse objects such as trees and bushes. Additionally, they are robust to varied lighting conditions, encountering difficulties only in extreme conditions such as direct sunlight measuring over $\pm 10,000lux$.

To illustrate the feasibility of this design methodology, a sensing algorithm must be exhibited on the scaled model, and the results must be replicated in the real world. A sensing algorithm must be utilized that tests the capabilities of the scaled T&E environment.

4 MISSION AND ALGORITHMS

One challenge faced by rotor-craft that is applicable across many UAV missions is the identification of a safe area to land. The process of detecting a safe landing zone can be broken into two parts. First, the terrain must be mapped. This is accomplished by fusing the laser scans with the pose measurements of the aircraft. Once a map has been generated, a landing zone

must then be extracted from the resulting terrain. The following sections describe the algorithms utilized to accomplish these steps.

4.1 MAPPING

To generate a terrain map, laser scans must be fused with relatively noisy pose measurements. This is accomplished using an implementation of the process presented in [9]. The fundamental concepts and their application are presented here in brief.

This algorithm produces a 3D map of the environment given noisy pose and terrain measurements. To find the corrected pose, a probabilistic model is constructed. This model is comprised of: the probability of the pose measurement, the probability of differential pose measurements, and the probability of consecutive scan alignment.

The probability of pose measurement is modeled as the probability of measuring the pose given the corrected pose. The system is taken to be 6 degrees of freedom, namely the 3 Cartesian coordinates and rotations about those axes. Their measurement at the current time step is the vector y_t , while the algorithm solves for the corrected pose x_t . Given the measurement covariance A , the probability of y_t given x_t as presented in [9] is then:

$$p(y_t|x_t) \propto \exp\left[-\frac{1}{2}(y_t - x_t)^T A^{-1}(y_t - x_t)\right] \quad (1)$$

The method also utilizes a differential model. Typically, the sensors onboard an aircraft measure rotational and translational rates. The pose is recovered through integration, making it susceptible to drift. The differential model is less affected by this error. Given D , the covariance of differential measurements, the differential model as derived in [9] is:

$$p(\Delta y_t|\Delta x_t) \propto \exp\left[-\frac{1}{2}(\Delta y_t - \Delta x_t)^T D^{-1}(\Delta y_t - \Delta x_t)\right] \quad (2)$$

where $\Delta y_t = y_t - y_{t-1}$ and $\Delta x_t = x_t - x_{t-1}$. As differential measurements are more accurate than absolute measurements, the covariance matrix D should represent a Gaussian with smaller standard deviation than A .

The final portion of the model is a representation of the likelihood of a scan. Rather than representing individual features as states as in traditional SLAM, the implementation in [9] models the consistency between consecutive scans as:

$$p(z_t|x_t, x_{t-1}, z_{t-1}) \\ \propto \prod_i \exp\left[-\frac{1}{2} \min[\alpha, \min_j (z_t^i - f(z_{t-1}^j, x_{t-1}, x_t))^T \\ B^{-1}(z_t^i - f(z_{t-1}^j, x_{t-1}, x_t))]\right]$$

The goal of this model is to align points from the current scan with points from the previous scan. The outer summation is over all points from the current scan. The i th point in the current scan z_t^i is compared to all points from the previous scan z_{t-1} . The inner summation is over all points from the previous scan. The function f maps the j th point from the previous scan z_{t-1}^j into the local coordinate system of the current scan x_t . The inner minimization identifies a point from the previous scan that is closest to the point from the current scan. The outer minimization thresholds this alignment to allow for local inconsistencies such as those from sparse objects. The matrix B is the measurement covariance.

Equations 1, 2 and 3 can be combined to form the probabilistic model for the entire problem [9]:

$$p(y_t|x_t)p(\Delta y_t|\Delta x_t)p(z_t|x_t, x_{t-1}, z_{t-1}) \quad (3)$$

The map and pose are recovered by finding the pose that maximizes this likelihood, or by minimizing the negative log likelihood given by [9]:

$$\text{const} + \frac{1}{2}((y_t - x_t)^T A^{-1}(y_t - x_t) \\ + (\Delta y_t - \Delta x_t)^T D^{-1}(\Delta y_t - \Delta x_t) \\ + \sum_i \min[\alpha, \min_j (z_t^i - f(z_{t-1}^j, x_{t-1}, x_t))^T \\ B^{-1}(z_t^i - f(z_{t-1}^j, x_{t-1}, x_t))])$$

This minimization is found by first minimizing to associate points from the current scan with those from the previous scan, and then performing hill-climbing to determine the pose that minimizes the negative log likelihood. These steps can be iterated until the negative log likelihood falls within a threshold.

4.2 SAFE LANDING ZONE ID

The algorithm presented in [10] provides robust detection of a safe landing zone based on the input of a point cloud terrain map from a LADAR scanner. This algorithm parameterizes a safe landing zone based on the slope of the landing area and the surface roughness. Costs are assigned to the terrain based

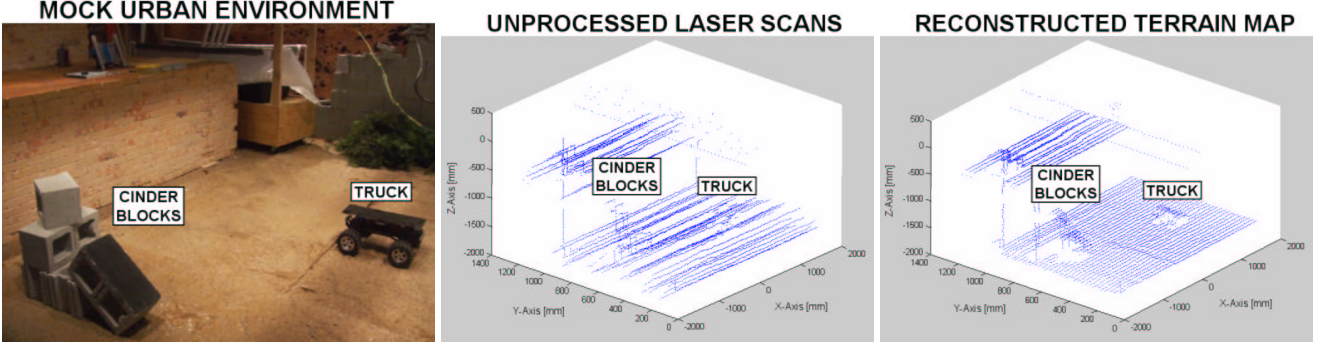


Figure 6. THE SENSOR WAS SCANNED THROUGH A MOCK URBAN ENVIRONMENT (LEFT). GAUSSIAN NOISE WAS ADDED TO THE POSE MEASUREMENT TO SIMULATE DATA GATHERED FROM A HELICOPTER (CENTER). THE TERRAIN MAP WAS THEN RECONSTRUCTED USING THE ALGORITHM DESCRIBED (RIGHT).

on these factors, and the lowest cost area which fits the helicopter rotor is selected. Our implementation of this algorithm is described below.

Because of the design of the laser scanner, the resulting point cloud is an irregularly spaced sampling of the scanned surface. A safe landing zone algorithm that uses this data would be intrinsically complicated and resource intensive. To simplify the implementation, the point cloud map is first converted into an image containing regularly spaced pixels. First the size of the grid cells must be determined. The width of each grid cell, C_w , is based on the angular resolution of the scanner θ and the average range to the surface R . The total width of the grid, G_w , is based on the field of view of the scanner f and the average range to the surface. The width of each cell, the total width of the grid and the total number of cells n is then:

$$\begin{aligned} C_w &= 2R \tan(\theta/2) \\ G_w &= 2R \tan(f/2) \\ n &= G_w / C_w \end{aligned} \quad (4)$$

The cell height was taken to be the same as the cell width. The total grid height G_h is determined from the distance traversed in the direction perpendicular to the scan plane. This formulation ensures there will not be multiple points per grid cell. After determining the cell and grid sizing, the (x, y) coordinates of data points in the point cloud must be transformed to (r, c) coordinates in the grid. This is accomplished using the relation [10]:

$$(r, c) = (y/C_w + G_h/2, x/C_w + G_w/2) \quad (5)$$

The value of each grid cell is based on the z -coordinate of the points. Interpolation is used to define cells that fall

in-between points. The resulting 2D array is analogous to a grayscale image whose pixel values correspond to the height of the terrain. This image is the raw elevation map.

To perform safe landing zone identification, the elevation map is separated into a surface roughness map and a landing incidence angle map. Both these maps require that an underlying smoothed surface first be determined. This surface is formed of by fitting square planes the size of the helicopter rotor diameter to the terrain map. Planes are represented as [10]:

$$\mathbf{n} \cdot \mathbf{x} + d = 0 \quad (6)$$

Where the fitted plane at cell $\mathbf{x} = (x, y, z)$ is described by $(\mathbf{n}, d) = (n_x, n_y, n_z, d)$. These planes are fit with an increment of $1/8$ the rotor diameter between planes. The resolution for the position of the chosen landing zone is therefore $1/8$ that of the helicopter rotor diameter. Smaller increments could be chosen to make this position more precise. This would come at the cost of processing speed. Due to error in the accuracy of the helicopter's pose measurement, the chosen resolution is believed to be sufficiently accurate.

The landing incidence angle α is calculated using the fitted planes and the geodetic normal of the surface n_g [10]:

$$\alpha = \cos^{-1}(\mathbf{n} \cdot n_g / \|\mathbf{n}\| \|n_g\|) \quad (7)$$

These fitted planes are also used to calculate the smoothed elevation map, where the smooth elevation z_s is given by [10]:

$$z_s = -(n_x x + n_y y + d) / n_z \quad (8)$$

Once the smoothed surface is generated, the roughness map can be determined. The roughness map $R(r, c)$ is calculated by subtracting the smoothed elevation map $Z_s(r, c)$ from the original elevation map $Z(r, c)$ and taking the mean [10]:

$$R(r, c) = |Z(r, c) - Z_s(r, c)| \quad (9)$$

A safe landing zone is chosen from a cost map. A cost is calculated for a region based on a weighted sum of the roughness and the slope. The weightings are chosen such that areas with a high roughness (and therefore obstacle rich) are avoided first. The remaining areas are then avoided if the landing incidence angle is too high. These weights are chosen based on the requirements of the platform.

5 EXPERIMENTAL RESULTS

The verification of this design methodology is an on going process. To demonstrate that the scaled model can be used to predict real-world results, the same mission must be executed on both the scaled model and in the real world. Specifically, the authors seek to show that a safe landing zone ID algorithm applied in both scenarios consistently identifies the same area for landing.

Before the latter can be demonstrated, the facility must be calibrated against a controlled real-world data set. This requires that the SR100 be flown over a predetermined path acquiring laser scans of the terrain. The flight data must then be used to generate the same path over the scaled environment, again acquiring data as the sensor is scanned over the scaled terrain.

To establish a “ground-truth” for the mapping and safe landing zone algorithms, the techniques were applied in a controlled indoor environment. The laser scanner was attached to the end effector of the gantry and traversed in a direction perpendicular to its scanning plain. This test was conducted over a full-scale mock urban environment. The results are shown in Fig. 6. Gaussian noise was added to the position data to simulate noisy pose measurements from a helicopter. As can be seen, the algorithm successfully recovers the terrain map in the form of a point cloud. This point cloud map can then be used in a safe landing zone algorithm.

The map generated by the laser scanner was then applied to the safe landing zone algorithm. Fig. 7 shows the cost maps produced by this process. Given the helicopter footprint, a landing zone was identified, marked with a cross in the figure. As can be seen, the area away from the cinder blocks and truck was chosen as the safest area to land. At this early stage, the algorithms have worked successfully using the LMS200 laser scanner in a full-scale mock urban environment.

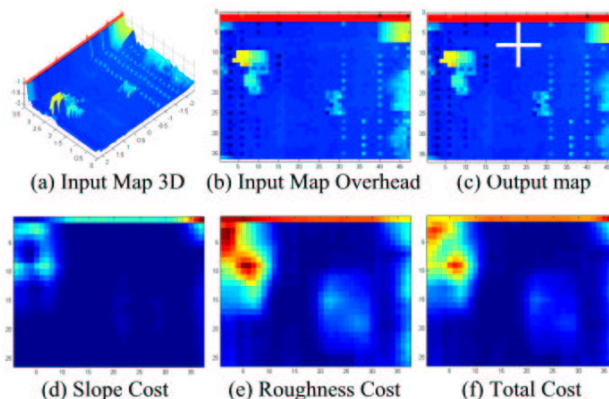


Figure 7. A TERRAIN IMAGE IS GENERATED FROM THE LADAR POINT CLOUD MAP. A COST MAP IS THEN CALCULATED BASED ON THE SLOPE OF THE TERRAIN AND THE LOCAL ROUGHNESS. THE SAFE LANDING ZONE, MARKED WITH A CROSS, IS DETERMINED AS THE LOWEST COST AREA THAT FITS THE HELICOPTER ROTOR DIAMETER.

6 CONCLUSIONS AND FUTURE WORK

Now that the mapping and safe landing zone algorithms have been shown to work, the next step is to conduct this experiment during an actual flight. The sensor must be flown in the Piasecki helicopter development range and used to generate a map of the facility. The path must be recorded and established relative to a known location. The mapping and safe landing zone algorithms must then be applied to this data set. In this experiment it will be crucial to avoid external influences to the data set, such as dust from downwash, strong sunlight which could dazzle the laser, and wind gusts which could blow the helicopter off course.

The experiment must then be recreated inside the scaled environment. The flight path must be scaled down along with the dynamics of the helicopter. Perhaps the most difficult aspect of the experiment will be appropriate scaling of the sensor. The authors acknowledge that the feasibility of scaling a laser scanner is an open problem. Areas of interest include scanning at higher resolutions and investigating comparable/smaller technologies.

Long terms goals for the project include investigating other common sensors and different missions. Sensors such as sonar and vision systems might scale better than laser scanners. Other missions such as path planning and autonomous navigation could be explored in a scaled environment. The ultimate agenda for this line of research is to identify what missions and platforms can be explored in a scaled environment to provide the UAV community with a continuous, risk managed path from laboratory development to real world flights.

ACKNOWLEDGMENT

The authors wish to thank Jesse Greenberg of Simulab Studios for the construction of the scaled testing environment. Thanks also go to Piasecki Aircraft for their continued support and use of their testing facilities.

REFERENCES

- [1] M. Alwan, M. B. Wagner, G. Wasson, and P. Sheth, "Characterization of Infrared Range-Finder PBS-03JN for 2-D Mapping", *International Conference of Robotics and Automation (ICRA)*, Barcelona, Spain, 2005.
- [2] C. Ye and J. Borenstein, "Characterization of a 2-D Laser Scanner for Mobile Robot Obstacle Negotiation", *International Conference on Robotics and Automation (ICRA)*, Washington, DC, 2002.
- [3] M. W. Orr, S. J. Rasmussen, E. D. Karni, and W. B. Blake, "Framework for Developing and Evaluating MAV Control Algorithms in a Realistic Urban Setting", *American Control Conference (ACC)*, Portland, OR, pp. 4096-4101, June 2005.
- [4] T. Netter and N. Franceschini, "A Robotic Aircraft that Follows Terrain Using a Neuromorphic Eye", *International Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, pp. 129-134, October 2002.
- [5] V. Narli, P. Oh, "A Hardware-in-the-Loop Test Rig for Designing Near-Earth Aerial Robotics", *International Conference on Robotics and Automation (ICRA)*, Orlando FL, pp. 2509-2514, May 2006.
- [6] D. J. Allerton, "Flight Simulation: Past, Present and Future", *The Aeronautical Journal*, vol. 104, no. 1042, December 2000.
- [7] S. Scherer, S. Singh, L. Chamberlain and S. Saripalli, "Flying Fast and Low Among Obstacles", *International Conference on Robotics and Automation (ICRA)*, Rome, Italy, pp. 2023-2029, April 2007.
- [8] M. A. Hsieh et. al. "Adaptive teams of Autonomous Aerial and Ground Robots for Situational Awareness". *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 991-1014, Nov. 2007.
- [9] S. Thrun, M. Diel, D. Hahnel, "Scan Alignment and 3-D Surface Modeling with a Helicopter Platform", *The 4th Int. Conf. on Field and Service Robotics*, Lake Yamanaka, Japan, pp. 14-16, July 2003.
- [10] A. Johnson et. al. "Lidar-based Hazard Avoidance for Safe Landing on Mars *AIAA Journal of Guidance, Control and Dynamics*, vol. 25, no. 5, October 2002.