HIERARCHAL PLANNING DEMONSTRATED IN AUTONOMOUS FLIGHT; INTEGRATION OF THE RISK-AWARE PATH PLANNER WITH A SENSE-AND-AVOID PLANNER ON A BLACK HAWK HELICOPTER

Chad Goerzen, Marc Takahashi, Matthew Whalley
U.S. Army Combat Capabilities Development Command Aviation & Missile Center, Moffett Field, California, USA

Gregory Schulein
San Jose State Research Foundation, Moffett Field, California, USA

Nathan Mielcarek
Universities Space Research Association, Moffett Field, California, USA

LTC Wesley Ogden, James Carr IV, Carl Ott, Kevin Thomazios, David Waldman
U.S. Army Combat Capabilities Development Command Aviation & Missile Center, Fort Eustis, Virginia, USA

Abstract

A hierarchal planning method with three levels was demonstrated in simulation and in flight on a Black Hawk helicopter. The software was flown on prior occasions and demonstrated the ability to autonomously sense and avoid previously unknown terrain while enroute to a destination. The new mission level software adds the ability to make mission-length plans over a stored terrain map, while simultaneously minimizing user-input risk functions. This system was tested with different levels of pilot input, ranging from simply entering a destination, to replanning while enroute, to adding lateral and longitudinal inputs to influence the flight path and altitude. The addition of this mission level improves the ability of the system to keep to a desired height above ground level, while improving the ability of the pilot or operator to predict the path of the helicopter and time to destination.

1. INTRODUCTION

One of the primary design patterns in autonomy is hierarchal planning: a large and complex planning problem is divided into different levels, of different time scales. Consider the aircraft planning problem: an aircraft flying near the ground must follow constraints dictated by the complex geometry of the terrain, process a high-bandwidth terrain sensor data stream, and ultimately give control commands to the four or more actuators. This is similar to many tasks in mobile robotics, and in this field hierarchal planning is the dominant technique for solving the control problem. By decomposing the complex problem into simpler tasks and establishing a hierarchy between the tasks, a solution can be computed and updated in real time that can approach optimal performance. Each level has a communications interface with the levels above and below it and provides certain safety guarantees. Hierarchical planning has been a dominant paradigm in operations even before the field of autonomous motion planning began. For example, in a piloted flight mission, the flight crew plans the flight and communicates the plan to relevant parties ahead of the flight. During the flight, a pilot senses and avoids terrain according to visual flight rules while still passing near waypoints mapped during pre-flight planning, while meanwhile maintaining a desired attitude, altitude profile, and power level that may vary from second to second. The above example can be formalized using a hierarchical control framework.

Hierarchical control has been suggested for autonomous motion planning for many years. In Ref. 1, a brain model is used as a basis of hierarchical control and lays out the fundamental theory of this ap-

Distribution Statement A: Approved for public release; distribution is unlimited. PR2023319.

proach. Two more recent approaches seek to optimize the framework: Ref. 2 uses an approach that optimizes multiresolution hierarchical control systems, and Ref. 3 optimizes the interfaces and communications between levels of control. Closely related are multiresolution planning approaches: Ref. 4 plans over a quadtree model of the world with smaller volume elements on the boundaries between occupied and free space. Refs. 5 and 6 both represent the world in fine detail close to the vehicle and coarse detail distant from the vehicle, allowing long-term planning while still ensuring safety in the immediate vicinity. Ref. 7 gives a proof of completeness of one refinement of this approach. Ref. 8 uses a dual resolution planner: the coarse resolution plan is over a terrain map, while the fine-resolution plan is over sensed terrain. The approach described in the main body of this paper can be considered a 3-dimensional generalization of this dual resolution approach.

US Army Combat Capabilities have accomplished research in helicopter autonomy for almost twenty years. Refs. 9 and 10 describes the control law and hardware development needed for autonomous flight testing of a 200 pound Yamaha RMAX helicopter, and Ref. 11 describes the first flights of the RiskMinOFN algorithm for sense-and-avoid flight of the same helicopter. Ref. 12 outlines the transition of the aforementioned software onto the full-authority fly-by-wire RASCAL JUH-60A helicopter, which achieved ground speeds of over 40 knots in sensor-guided flight. Improvements in the sensor allowed an increase in speed up to 114 knots in sensor-guided flight, as described in Ref. 13. The Autonomous Partial-Authority Flight Control System (APAFCS) allowed operations on an EH-60L Black Hawk helicopter, described in Ref. 14. This aircraft is similar to Black Hawk helicopters in the US Army fleet. These earlier flight tests focused on demonstrating safety and performance of a local sense-and-avoid planner and did not use any maps or prior knowledge of the terrain. More recently, we added a mission level planner to the existing system, in effect extending the hierarchal framework from two levels to three (see Table 1). These planning levels are categorized here by the time horizon involved but have other differences. For instance, the highest level is characterized by a complex environment at coarse resolution with a low dimensionality space of 2.5 dimensions (a three-dimensional terrain surface represented by a two-dimensional map). By contract, the lowest level is characterized by a simple spatial environment, but of a higher dimensionality state space consisting of position vector, velocity vector, attitude vector, and other state variables. The time horizons given are order-of-magnitude approximates that vary from one flight to another.

Table 1. Breakdown of planning levels

Level	Task	Planning Horizon	Process Name
High	Mission planning	~30 minutes	RAPP
Medium	Reactive planning	~3 minutes	RiskMinOFN
Low	Trajectory and inner-loop control	~30 seconds	APAFCS

The highest level is newly written software called the Risk Aware Path Planner (RAPP). It is designed to integrate with the RiskMinOFN process, acting as a mission level planner with prior knowledge of terrain and airspace maps. It creates a pre-plan for the entire mission and can be triggered to replan by the operator at any time during the flight. It is a quasi-3-dimensional planner that uses a digital terrain map as its base, and sends coarse waypoints to RiskMinOFN. By contrast, RiskMinOFN is a fully-three-dimensional search that operates over a limited time horizon of up to 3 minutes, which gives velocity commands each second to the Autonomous Partial Authority Flight Control System (APAFCS). This lowest level takes care of trajectory generation, waypoint and velocity command following, and inner loop stabilization. While the system has been flown under control of only the Middle and Low levels, this paper will show the effects of adding the High level to the architecture.

2. MAIN BODY

2.1. System architecture

The system architecture that follows this structure is shown in Figure 1. Each level reads data from external sensors or stored maps, including the digital elevation map, terrain sensor, and INS/GPS navigation sensor. The operator/pilot enters a destination command into RAPP: this may be a hover point, a landing zone, and approach direction may be restricted to a certain angle or free. The APAFCS gives actuator commands to the helicopter.

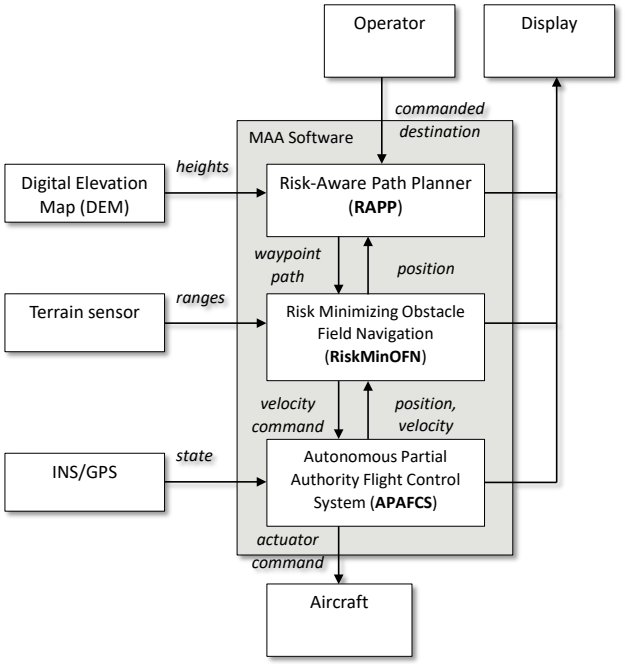

Figure 1: System block diagram showing the interaction of the levels of control.

2.2. Algorithm description

The algorithm used by the RAPP process is based on the A* search, an algorithm that is widely used for grid searches. However, this A* search is a step in a larger process:

1. *Initialize*: Set parameters, load terrain from storage, and load airspace threats from storage. Wait for a destination command and aircraft state data.
2. *Create a risk map*: Resample the terrain to desired planning resolution and limits, based on the information from step 1. Calculate risk values for each pixel – this risk assumes the aircraft will be at the desired height above ground level (AGL).
3. *Search*: Use the A* algorithm to search for the lowest risk path from the current state to the destination. Start state needs to be adjusted to account for vehicle motion while planning, and end state may need to be adjusted to control the ingress direction. This is the most computationally demanding step.
4. *Obtain path*: A backtrace through the A* solution results in a list of equally spaced points at the resolution of the risk map. This is the lowest risk path to the desired state.
5. *Simplify path*: To avoid over-constraining RiskMinOFN, waypoints falling in a straight line are eliminated. The remaining waypoints only are found at turns in the path.
6. *Send result*: Format the points and send them to RiskMinOFN. Repeat from Step #1.

A note on step 2 is important. One unique aspect of this research is that we use explicit modeling of mission risk factors in both RAPP and in RiskMinOFN. This allows automatic adaption to changes in the environment, such as higher winds and gusts that make control tracking less precise. It is also a unifying methodology used to weight differing mission constraints against each other. While some of the risk factors will be common between RAPP and RiskMinOFN, they will each have some factors more relevant to the scale at which they operate. RAPP uses three terms of risk, constant time risk, altitude risk, and threat site risk:

I. Constant time risk:

(1) $R_{time} = rl/v$

where *r* is risk per second, *l* is cell length, *v* is cruise velocity

II. Altitude risk:

(2) $R_{alt}(x,y) = h(x,y)a_{max}$

where a_{max} is maximum altitude risk, $h(x,y)$ is space-varying terrain height

III. Threat site risk:

(3) $R_{threat}(x,y)$

where R_{threat} is the space-varying user defined risk

Total risk: The above three factors are combine using probabilistic union:

(4) $R_{total} = R_{time} \cup R_{alt} \cup R_{threat}$

where the form for independent variables is used to calculate the probabilistic union:

(5) $A \cup B = A + B - AB$

The probabilistic union used in implementation to combine risk from time in operation, altitude, and threat sites assumes the three factors are independent. However, it is straight-forward to account for non-independence and other higher-order factors. Also, other risk factors can be factored in by using probabilistic union. For example, if a weather map is available, it can be expressed as a risk value and combined with the total risk using probabilistic union.

2.3. Interfaces

The interfaces between RAPP and the other software are an important aspect of this research. The software is designed to minimize system disruption, so the interface between the operator and RAPP is identical to that previous interface between the operator and RiskMinOFN. This consists of a list of one or more destinations which include Universal Transverse Mercator (UTM) coordinates and an approach angle constraint. These coordinates consist of easting, northing, and height, although height may be unspecified to allow OFN to determine the altitude. If the list contains only a single destination, OFN will come to a hover at this point. If the list contains multiple destinations, OFN will approach each destination on the list to within a threshold radius. These are called waypoints. Once this radius is achieved, the achieved waypoint is deleted from the list and OFN continues to the next destination. The threshold radius is set so that the helicopter will not begin slowing down for waypoints. As a result, if the waypoint path specifies a sharp turn, OFN will turn early and miss the waypoint by a large distance. The OFN software has ultimate responsibility for keeping the vehicle away from terrain and threats, and is designed to do so even if a waypoint is located within the terrain. This identical interface is taken over by RAPP. If the user enters a single destination, RAPP produces a waypoint path through low terrain to that destination, as outlined in the previous section. If the user enters multiple destinations, RAPP calculates a waypoint path for the list of destinations, resulting in a long waypoint path that passed through the desired waypoints.

The interface between RAPP and RiskMinOFN is largely similar to that between the user and RAPP. There are several parameters that need to be determined. First, spacing between waypoints. RAPP creates a waypoint list at grid spacing (here set to 100 meters between points) and simplifies it by eliminating redundant points – only points at the ends of straight lines need be included. The resulting points are unevenly spaced, with a minimum space of 100 meters between them, and no maximum space. Second, altitude designation. Initially, RAPP generates waypoints with altitude constraints, specifying a constant altitude above the terrain map. However, this constraint resulted in unsatisfactory behavior on the part of RiskMinOFN. For instance, if one of the RAPP points happened to fall in a deep valley, it would specify a spuriously low altitude. The result was that RiskMinOFN occasionally reduced speed and altitude for a valley that would have been quickly overflown at constant speed and altitude. A more profound example is over steep terrain, any low-resolution terrain map has significant altitude error. RAPP could specify an altitude that is too close to the terrain. While RiskMinOFN can recover from this error condition, in practice it tends to search for a lower-risk path to this inaccessible waypoint, resulting in unnecessary delays and possibly even an error condition resulting in RiskMinOFN taking the helicopter to a hover and sending an error message to the operator. To avoid these error conditions, RiskMinOFN now outputs 2-dimensional easting/northing coordinates while leaving altitude unspecified: RiskMinOFN attempts to fly through these waypoints at the desired height AGL according to the terrain picked up by the sensor. The final waypoint on the list does, however, keep the height constraint, to allow hover at a desired altitude.

The timing aspect of the interface is also important. While RiskMinOFN executes a complete replan every two seconds, RAPP only executes a replan when commanded by the operator. For longer missions of around 100 kilometers, a RAPP plan may take several seconds to complete. Many missions stick with the original RAPP plan, but it is trivial for the pilot to replan RAPP while enroute. Since RiskMinOFN has full control over terrain and threat separation, it is safe for RAPP to take a long time to calculate a replan. One important aspect of the replan is that RAPP uses the point of nearest hover calculated by RiskMinOFN as its initial point rather than the current position. This ensures the path produced by RAPP is flyable without backtracking. In practice, RiskMinOFN marks the first RAPP waypoint as “complete” within one second of the creation of the RAPP waypoint path.

An important feature facilitated by RAPP is the constrained approach direction. While this feature could be implemented in RiskMinOFN, the scale is such that it would be tricky to account for all possible scenarios. RAPP has a mission scale view, so is better suited to planning an approach with a constrained direction. This feature is important for landing at an airfield in coordination with other air traffic, as well as landing in high wind conditions.

An existing RiskMinOFN mode is GO-ARND, or constant direction mode. In this mode, the destination given to RiskMinOFN consists only of a direction. RiskMinOFN will fly generally in this direction, while still avoiding terrain and threats. It is used for takeoffs and for cancelling the current mission, giving the operator time to determine a new course of action.

To summarize, the original waypoint interface to RiskMinOFN was essentially split into two interfaces: coarse waypoints from the operator to RAPP, and fine waypoints from RAPP to RiskMinOFN. The RAPP waypoints are 2-dimensional waypoints with unconstrained altitude, except for the final point which may have a specified altitude. Both interfaces are event-based and can happen as rarely as once each mission, and as frequently as the operator presses the replan button.

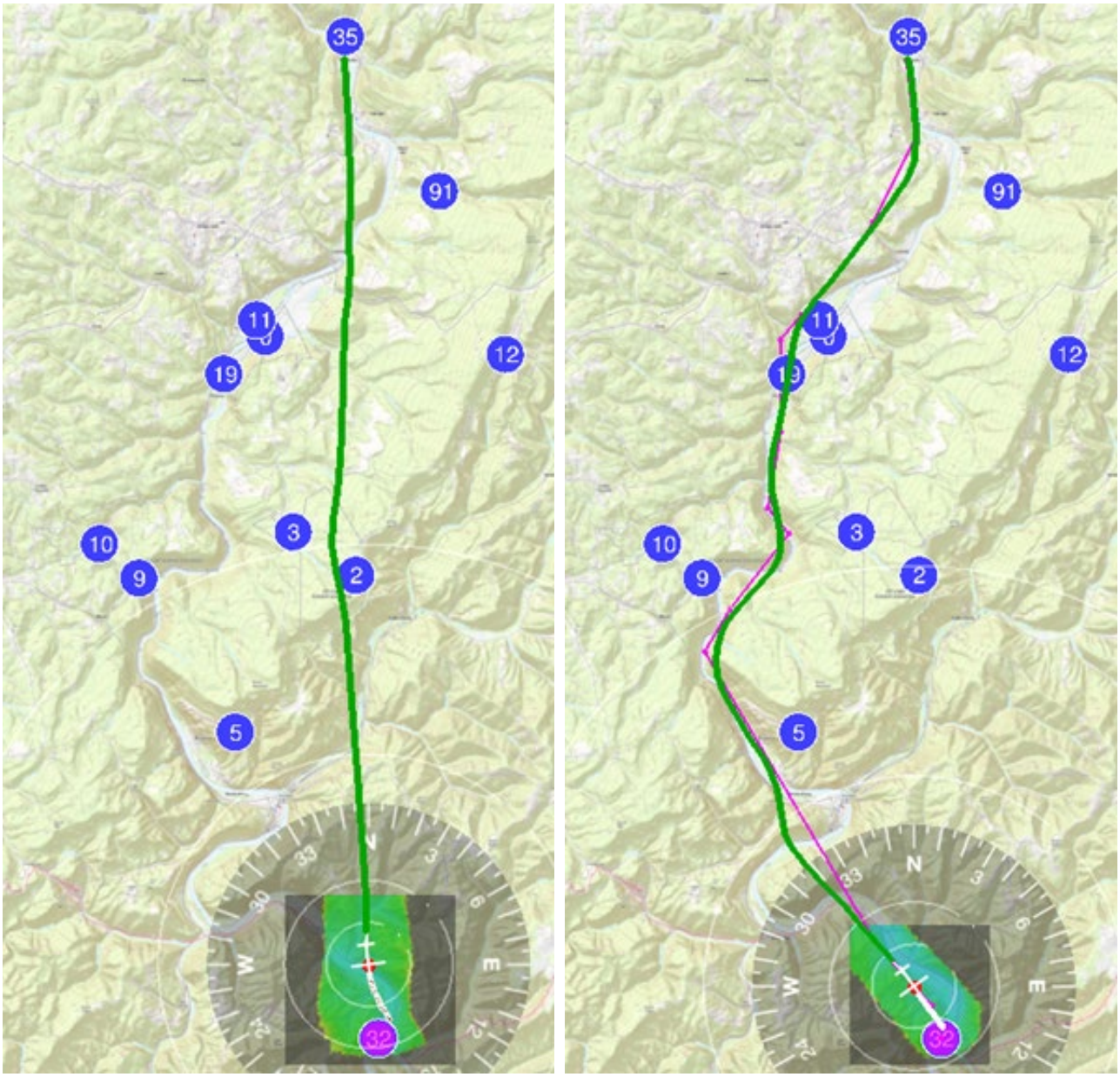

Figure 2: Comparison of routes flown with RAPP disabled (left) and with RAPP enabled (right). Terrain relief is dominated by the river going through waypoints 35 and 9 and a ridge going through waypoints points 12 and 5. Blue circles indicate waypoints used as destinations. The green line shows the path flown, while the thin magenta line and dots show the RAPP output. The colored square around the aircraft shows the RiskMinOFN view of recently sensed terrain.

3. SIMULATION TESTING

3.1. Simulation environment

The simulation environment accounts for many of the flight conditions the autonomy software is expected to be exposed to in flight. The flight model and control system simulator are integrated in a stand-alone desktop simulation model. This model consists of a linearized model with stability derivatives interpolated across the 0 to 120 knot airspeed. This model includes non-linear kinematics, including actuator models and limits. Bounding aircraft trim points are set to preserve stability derivatives that depend on velocity. In the desktop simulator, the Autonomous Partial-Authority Flight Control System (APAFCS) is fully integrated with the other autonomy software in the same way as it is for flight tests. Hardware specific interfaces, such as the 1553 bus, are emulated. This flight model accounts for an average level of navigation solution noise. It also uses a sensor simulator, a GPU-accelerated beam-wise simulation of a terrain range sensor. In this case it simulates the H. M. Burns Corp. Multi-Function Lidar (MFL) sensor. The sensor simulator emulates the time-varying geometric state of the sensor head relative to the aircraft and returns the range in the direction of each beam. Its noise model includes:

- Additive Gaussian noise on range returns: for Lidar systems, this is a relatively small contribution to noise, on the order of magnitude of 0.1 meters.
- Dropouts, or false negatives: the range-variant dropout function randomly falsely changes range to NO_RETURN; the dropout rate increases with increasing range. This rate becomes significant near maximum range.
- False positives, including wrap-around hits: this model takes additional GPU resources and is only run for calibration runs to ensure the evidence grid filters them out in the region near the aircraft.
- Sensor position and angle drift model: this is a Brownian motion processes with a correction term to represent the GPS corrections. It represents drift in the angle and position estimates of the sensor. It can be a significant source of error.

Care was taken to calibrate these four error models according to data acquired in flight. The additive noise model used a scan of a runway from hover, estimating standard deviation from a plane fit to determine noise level. Dropout rate was acquired by a flight in terrain and a matching flight in simulation – the range histogram from the flight data was compared against the range histogram from simulation. False positives rate was acquired by flight over a flat surface; any returns with altitude above the ground level higher than threshold were considered false positives. Sensor position and angle drift were estimated by looking at the difference between sensor position and angle estimates and aircraft angle and position estimates and verified by comparing simulated landing approach scans with scans from flight. The range at which a radio tower is detected by the fused terrain in flight is compared to detection range in simulation, to ensure they match.

This noise model has been matured over the years to include increasing numbers of sensor errors that were found to be significant. While a physics-based simulator would be better suited to deal with different surfaces, especially bodies of water, this would involve an increase in complexity of the sensor simulator and make it difficult to achieve real-time performance. To ensure safe flight over large bodies of water, a planar airspace limit (“threat”) is coded into the parameter file during any flight over a body of water with more than ~1 km maximum width.

3.2. Simulation procedure

Testing in simulation has been carried out to compare the system with and without RAPP. Figure 2 shows the result of one pair of these runs. This example is typical of the difference between the two modes. RiskMinOFN by itself has no concept of the overall view of the mission, only of what is in its line-of-sight and within sensor range. Since RAPP has access to terrain for the length of the entire route, it produces waypoints that result in a lower altitude profile and make more use of terrain masking.

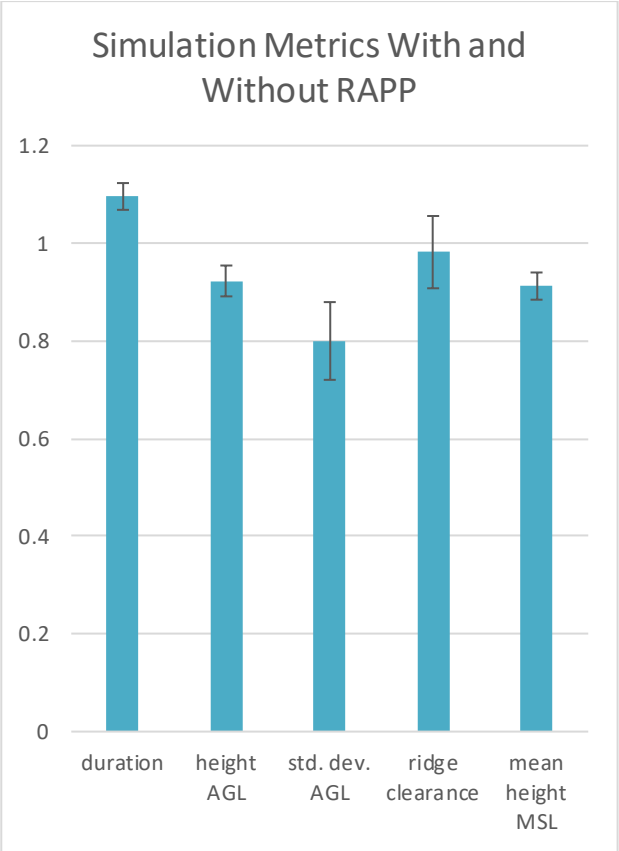

Figure 3: The ratio of performance with RAPP to performance without RAPP averaged over 21 flights for a variety of performance metrics.

For the averaged performance, a series of 21 missions was executed, both with RAPP and without RAPP. The mission length ranged from 8 to 30 minutes. Five performance metrics were recorded:

- *duration*: the time taken to execute the mission
- *height AGL*: root mean square average of the height above ground level
- *std. dev. AGL*: standard deviation of the height above ground level
- *ridge clearance*: safety metric that measures minimum clearance above terrain while at speed
- *mean height MSL*: averaged height above Mean Sea Level

Figure 3 shows metrics for the metrics for these runs. The largest difference between the configurations was in altitude: using RAPP reduced overall altitude by 10 percent, reduced height above ground level by 9 percent, and reduced the standard deviation of height above ground level by 20 percent. There was no significant change in ridge clearance, meaning that this reduced altitude did not impact the safety of the mission in a negative way. RAPP accomplished this by taking longer routes to better fly through low-altitude parts of the terrain, resulting in an increase of 10 percent in mission duration.

4. FLIGHT TESTING

4.1. Flight procedures

RAPP was integrated into the autonomy software stack for flight testing. With a limited number of flight hours available, the flight tests have some specific purposes:

- Validate the realism of the simulation environment by comparing the system in the simulator to the same system in flight.
- Investigate pilot interactions with the autonomous tools, and the usefulness of feedback of internal states to the pilot for situational awareness and process monitoring.

There are three types of flight test that were accomplished using RAPP:

1. Checkout: starting in a hover 20 – 50 meters above ground level, send to a landing zone at least 2 km straight ahead over flat terrain.
2. Traffic pattern: Starting in a hover 20 – 50 meters above ground level, press GO_ARND to takeoff. Once ground speed reaches 25 m/s (50 knots), select a landing zone with a constrained approach angle close to the takeoff point. The approach angle constraint causes RAPP to lay out a traffic pattern.
3. Terrain flight: Starting in a hover 20 – 50 meters above ground level, press GO_ARND to takeoff. Once the ground speed reaches 25 m/s (50 knots), select a distant landing zone. Use pilot input tools, including OFN speed up/slow down, pilot additive lateral and heave inputs through the 4-way switch, and replan to accomplish secondary tasks such as following soft constraints such as avoiding low-altitude overflight of residential areas.

All three of these flight types end with a Safe Landing Area Detection (SLAD) scan. Since the RAPP

algorithm is not used in final landing operations, this paper will not detail the landings, other than to give an overview of the landing procedure. When the aircraft is within 60 seconds of arriving at a hover point 40 meters above the center of the landing scan area, it points the sensor head down to focus on the landing area and slows down the scan rate to increase data density. Once enough data are collected, a landing suitability map is shown on the pilot displays, and candidate landing points selected by the SLAD algorithm are shown. If the pilot makes no selection, the aircraft will wait for a few minutes at the original hover point. If the pilot selects a landing point, OFN will guide the helicopter down to a hover 15 meters above the landing point. At this point OFN hands over control to the landing software, which takes the aircraft down to the ground.

There are four flight modes that the mission script and operator may put OFN into:

- *Cruise:* 59.3 m/s (115 knots), 350 m (1148 ft) AGL
- *TerrainPlus:* 51.44 m/s (100 knots), 80 m (262 ft) AGL, increased acceleration limits
- *Terrain:* 41.2 m/s (80 knots), 80 m (262 ft) AGL
- *Approach:* 30 m/s (58.3 knots), 60 m (197 ft) AGL, reduced acceleration limits

The GO-ARND command automatically sets the flight mode to *Terrain*. When a destination is entered, the flight mode is automatically set to *TerrainPlus*. On the event of approaching within 120 seconds of the landing zone, the mode is automatically set to *Terrain*. This has two purposes: first, if the operator had previously set flight to *Cruise* mode, it gives the helicopter time to reduce altitude in time for landing approach; second, it reduces speed more gradually to avoid encountering the lower collective limit on final approach. On the event of approaching within 60 seconds of the landing zone, the mode is automatically set to *Approach*. This allows more accurate flight closer to the terrain and gives the system enough time to scan the landing zone.

4.2. Comparison test: traffic pattern

The purpose of this test is to compare flight in the simulation environment with actual flight. The traffic pattern was used for this since it exercises the major parts of the autonomy stack while being of relatively short duration. It is also something that all pilots are familiar with, giving them a basis for evaluation.

This flight was executed on July 19, 2023, run number 011, at Felker Army Airfield, Virginia. The results of both the simulation and flight are coplotted in Figure 4, where black lines indicate flight data while red lines indicate simulation data. Figure 4 a). has time histories for speed, roll, pitch, yaw, and collective. The speed profile shows TerrainPlus mode used initially, with an automatic transition to Terrain mode at around 190 seconds. During the flight, the evaluation pilot decreased the OFN speed limit to 70 knots to ensure the system could turn tight enough; this was replicated in the simulation run. While the system roll limit is set to 30 degrees, in practice the limited sensor field-of-view limits the achieved roll to around 10 degrees for turn-about maneuvers. It is expected that mechanically pointing the sensor into the direction of the turn can allow a tighter turning radius and more closely achieve the roll limit. Pitch and roll histories align closely between simulation and flight, while collective shows a bias between simulation and flight. This is because the simulator is calibrated to a particular aircraft weight, while in flight the weight varies as fuel is consumed. Figure 4 b). shows the altitude profiles. The simulation shows similar timing, but a bias towards higher altitude; the flight height AGL more closely matched the desired height of 80 meters. This difference can be attributed to the sensor model; in flight, there are few to no Lidar returns from the water but in simulation the water is treated the same as land. While the horizontal plane threat representing the water is in place in both flight and simulation, the detected land plane in simulation causes RiskMinOFN to go for greater separation distance. Note also the height AGL, as measured by the radar altimeter, has rapid changes indicating forests, whereas the simulator in this case uses a flat earth model. Figure 4 c). shows paths that overall line up but show some differences. The actual path flown is sensitive to small changes in initial condition and timing of using inputs, and the differences in path are roughly equal to differences from one flight to another.

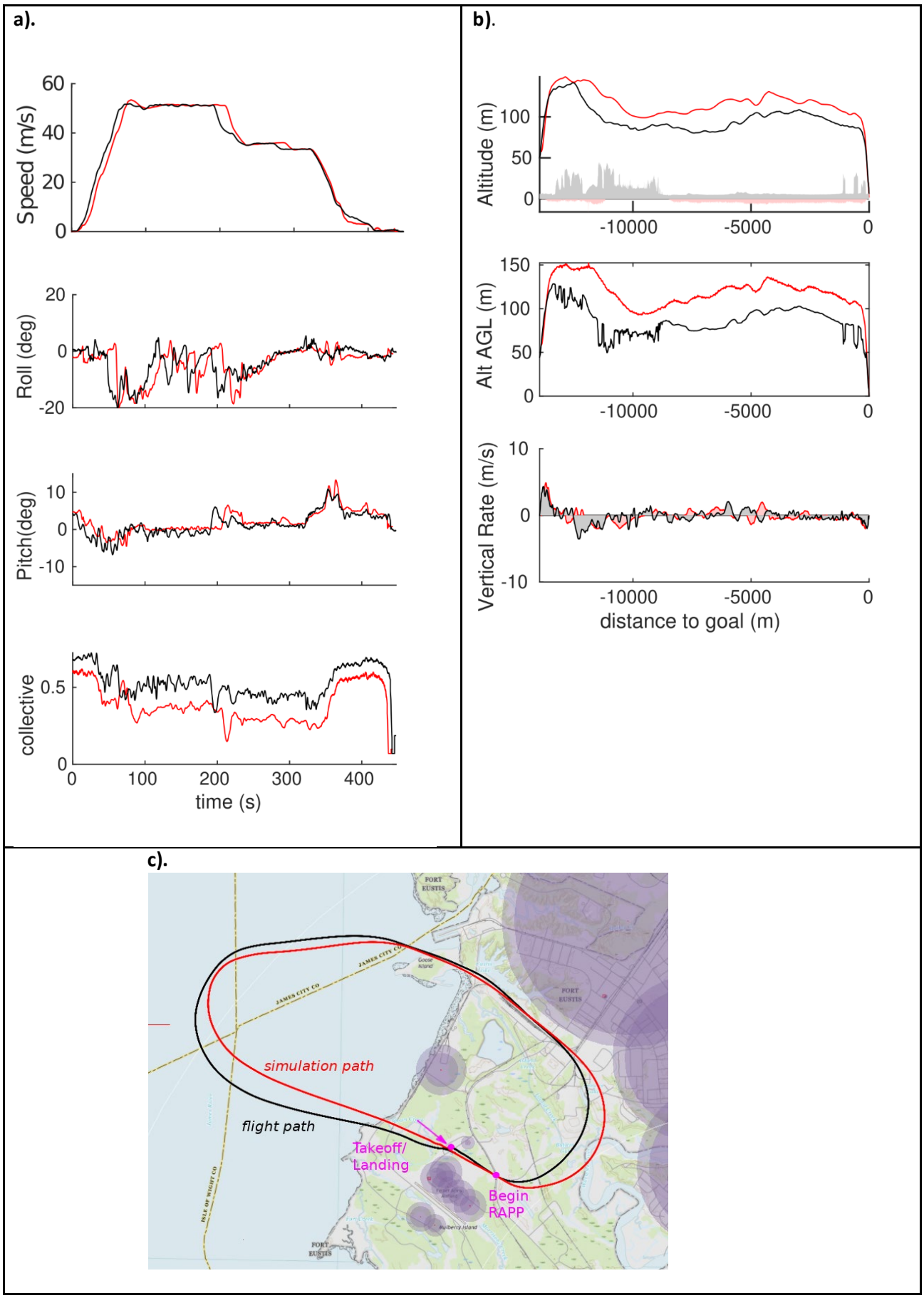

Figure 4: Comparison of flight data (black) with simulation data (red). Section **a)** shows time histories of speed, roll angle, pitch angle, and collective position. Section **b)** shows altitude profiles and vertical rate – this flight ended with a landing near sea level, so altitudes both end at 0. Section **c)** shows paths overlayed on the area map – purple circles indicate no-fly zones requested by the airfield, magenta circles indicate the start and end points given to RAPP, with the arrow indicating the landing direction constraint.

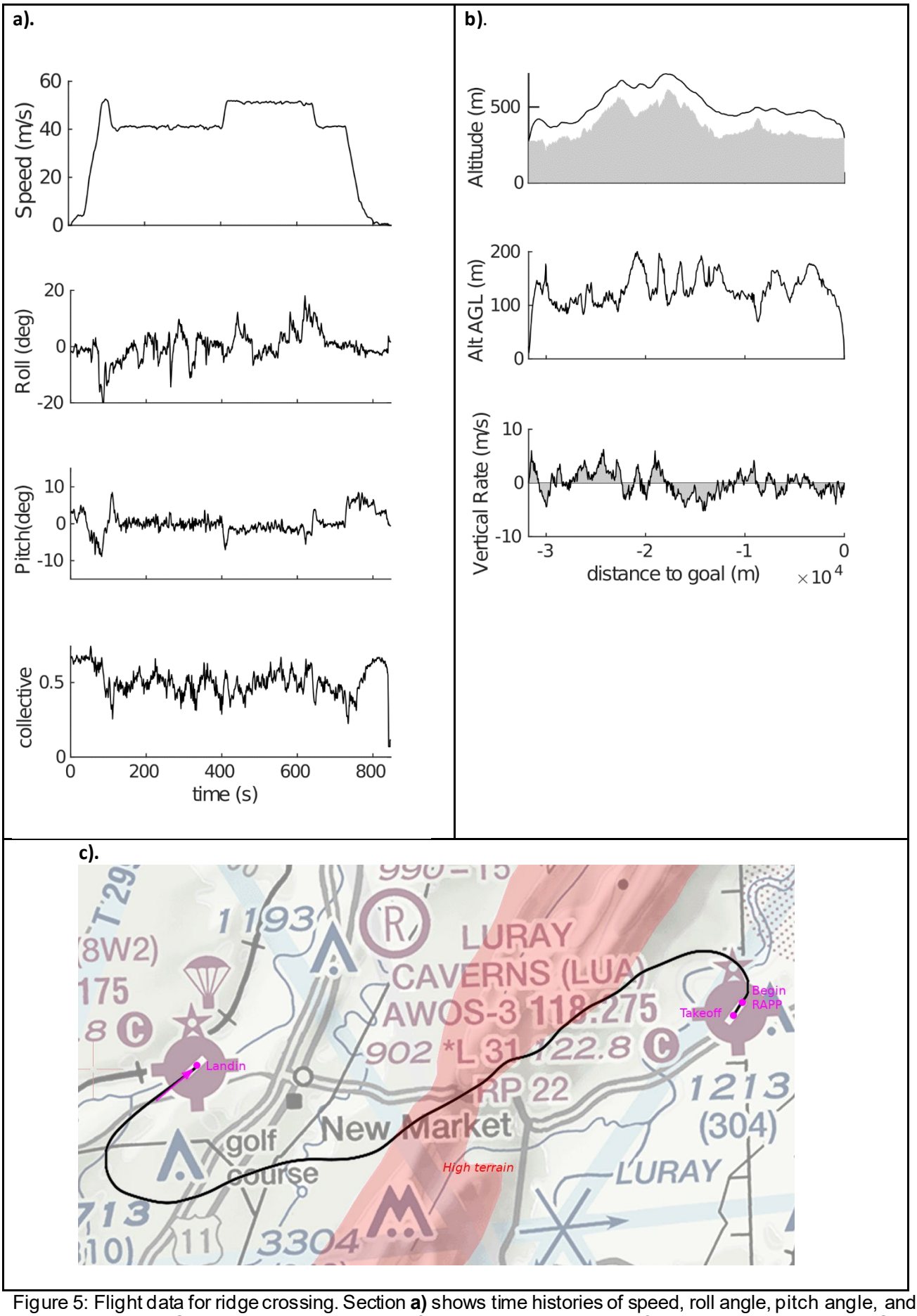

Figure 5: Flight data for ridge crossing. Section **a)** shows time histories of speed, roll angle, pitch angle, and collective position. Section **b)** shows altitude profiles and vertical rate – this flight ended with a landing. Section **c)** shows paths overlayed on the area map – magenta circles indicate the start and end points given to RAPP, with the arrow indicating the landing direction constraint.

Figure 6: Frame from flight record video. **Upper left:** pilot's moving map display, shows RiskMinOFN fused terrain in false color overlay with red indicating high terrain and blue indicating low, RAPP route as a magenta line. **Upper right:** pilot's first-person view display shows true color Lidar data, aircraft state information, flight path vector is a cyan aircraft symbol. **Bottom (left to right):** forward-facing camera, backward-facing camera, cockpit camera, tail camera.

4.3. Terrain test

A flight among terrain was executed in the area around Shenandoah Valley, Virginia, on November 8, 2022. Shown here is run number 007. This flight involved taking off from one airport, flying over the lowest gap in a sharp ridge with 300-500 meters of relief, and flying to a different airport with a constrained approach direction.

Figure 5 shows selected flight data records for this flight. Figure 5 a). has the time histories of ground speed, roll angle, pitch angle, and collective. From the ground speed it is evident that after achieving full speed, the evaluation pilot put OFN into *Terrain* mode with a maximum speed of 41.2 m/s (80 knots). Midway through the flight, after crossing the ridge, the evaluation pilot put OFN back into *TerrainPlus* mode. The mission system automatically slowed OFN back to *Terrain* mode at 120 seconds to destination, and *Approach* mode at 60 seconds to destination. It executed a landing area survey and the evaluation pilot selected a landing point, allowing the system to autonomously land. Figure 5 b). shows the altitude profile flown. Despite there being over 300 meters of relief, the system tracked the desired height AGL of 80 meters relatively consistently. Figure 5 c). shows the path flown. After taking off in the northeast direction from the airstrip on the east side of the map, the helicopter followed the RAPP path to gradually climb over the sharp ridge, cross at the lowest gap, and gradually descend the other side before turning around and approaching the airport shown on the west of the map from the desired approach heading.

An example frame from the flight video record of this flight is shown in Figure 6. This shows the helicopter under full autonomous control crossing the gap in the ridge. The moving map in the upper left of the image indicates a lateral pilot offset with the split magenta pointer: the pilot clicked the left direction of a four-way switch, causing the system to fly to the left of the RAPP route.

The system successfully executed 97 minutes of research flight in the terrain around the Shenandoah Valley. Due to a misconfiguration, the RAPP system failed to avoid high terrain as was desired from many of the flights, but instead minimized the

change in altitude. That caused RAPP to seek high terrain in some cases. This has since been corrected in the parameter files. It has additionally executed two traffic patterns of eight to nine minutes each and six checkout runs of three to five minutes each.

5. CONCLUSIONS

This paper represents ongoing work and further flights among terrain are planned for later this year. The most significant results of this paper are:

1. The Risk-Aware Path Planner (RAPP) added a level to the existing hierarchical control framework, increasing planning horizon from sense-and-avoid to mission timeframe.
2. Adding RAPP has improved the system's ability to keep a consistent height above ground level while reducing overall height.
3. Procedural start point and end point adjustments in the RAPP algorithm were found to handle replans in flight and constrained approach directions gracefully, and in fact can generate standard traffic patterns with minimal user input.
4. The existing waypoint interface was found to be sufficient to control the sense-and-avoid planner along a mission length path.
5. A waypoint path display on the moving map gives pilots/operators a better prediction of the vehicle's position.

6. REFERENCES

1. [Albus, 1979] Albus, J. S., "Mechanisms of Planning and Problem Solving in the Brain", Mathematical Biosciences, vol. 45, pp. 247-293, 1979.
2. [Maximov 1992] Maximov, Y., and Maystel, A., "Optimum design of multiresolutional hierarchical control systems," in Proceedings of IEEE Int'l Symposium on Intelligent Control, p. 514-420, Glasgow, 1992.
3. [Lacaze 2002] Lacaze, A., "Hierarchical Planning Algorithms," in Proceedings of SPIE 16th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, 2002.Johnson, W., *Helicopter Theory*, Princeton University Press, Princeton, NJ, 1980, pp. 808–813.
4. [Kambhampati 1986] S. Kambhampati and L. Davis, "Multiresolution path planning for mobile robots," in IEEE Journal on Robotics and Automation, vol. 2, no. 3, pp. 135-145, September 1986, doi: 10.1109/JRA.1986.1087051.
5. [Behnke, 2003] Behnke, S. (2004). Local Multiresolution Path Planning. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds) RoboCup 2003: Robot Soccer World Cup VII. RoboCup 2003. Lecture Notes in Computer Science(), vol 3020. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-25940-4_29.
6. [Tsiotras 2007] Tsiotras, P., and Bakolas, E.,"A Hierarchical On-Line Path-Planning Scheme Using Wavelets," European Control Conference, Kos, Greece, July 2-5, 2007.
7. [Cowlagi 2010] Cowlagi, R., and Tsiotras, P., "Multi-resolution Path Planning: Theoretical Analysis, Efficient Implementation, and Extensions to Dynamic Environments," 49th IEEE Conference on Decision and Control, Atlanta, GA, Dec. 15-17, 2010, pp. 1384-1390
8. [Ferguson 2006] Ferguson, D., Stentz, A. Multi-Resolution Field D*. In Proceedings of the International Conference on Intelligent Autonomous Systems (IAS), IOS Press, 2006.
9. [Whalley 2003] Whalley, Freed, Takahashi, Christian, Patterson-Hine, Schulein, and Harris,"The NASA/Army Autonomous Rotorcraft Project," presented at the American Helicopter Society 59th Annual Forum, Phoenix, AZ, May 2003.
10. [Takahashi 2008] Takahashi, Schulein, Whalley, "Flight Control Law Design and Development for an Autonomous Rotorcraft" presented at the 64th Annual Forum of the American Helicopter Society, Montreal, Canada, May 2008.
11. [Goerzen 2011] Goerzen, C. and Whalley, M., "Minimal risk motion planning: a new planner for autonomous UAVs in uncertain environments," presented at The AHS International Specialists Meeting on Unmanned Rotorcraft, Tempe, AZ, Jan. 2011.
12. [Whalley 2013] Whalley, Takahashi, Goerzen, Schulein, Fletcher, Moralez, Ott, Olmstead, Savage, Burns, and Conrad, "Flight Test Results for Autonomous Obstacle Field Navigation and Landing Site Selection on the RASCAL

JUH-60A," presented at the 69th Annual Forum of the American Helicopter Society, Phoenix, AZ, May 2013.

13. [Goerzen 2015] Goerzen, Whalley, Takahashi, Schulein, Ott, Minor, Savage, Burns, and Conrad, "High Speed Canyon Flight: Obstacle Field Navigation Envelope Expansion on the RASCAL JUH-60A Black Hawk," presented at the Sixth AHS International Specialists' Meeting On Unmanned Rotorcraft Systems, Chandler, AZ, January 2015.

14. [Takahashi 2021] Takahashi, Fujizawa, Lusardi, Whalley, Goerzen, Schulein, Mielcarek, Cleary, Carr, and Waldman, "Autonomous Guidance and Flight Control on a Partial-Authority Black Hawk Helicopter" Journal of Aerospace Information Systems Vol. 18, No. 10, 13 April 2021.