AIAA

Autonomous UAV Guidance Build-Up: Flight-Test Demonstration and Evaluation Plan

B. Mettler, T. Schouwenaars, J. How,
J. Paunicka and E. Feron

Massachusetts Institute of Technology,
Cambridge, MA 02139
Boeing Phantom Works, Seal Beach, CA 90740

Autonomous UAV Guidance Build-Up: Flight-Test Demonstration and Evaluation Plan

B. Mettler, * T. Schouwenaars, † J. How, ‡
J. Paunicka § and E. Feron ¶

Massachusetts Institute of Technology, Cambridge, MA 02139
Boeing Phantom Works, Seal Beach, CA 90740

This paper describes plans towards flight-test demonstrating an autonomous guidance system for unmanned aerial vehicles (UAVs). The goal of the experiment, which will take place on Boeing's UAV test-bed in 2004, is to demonstrate key autonomy capabilities that are critical for future UAV operation. These are enabled using a receding-horizon implementation of a trajectory planner formulated as a mixed-integer-linear-programming (MILP). Hybrid vehicle dynamics combining multiple linear time-invariant (LTI) modes are used to cover the scope of vehicle dynamics. In addition to the LTI modes, coordinated maneuvers that have inherent operational value can be invoked. The planner also reckognizes a number of tasks primitives. Trajectory safety is guaranteed by computing a rescue path a every time step. Prototype demonstration scenarios are designed to exercise and evaluate the guidance system's primary autonomy capabilities. The key milestones of the technology buildup and transition to the Boeing open control platform (OCP), and toward flight testing are described.

Overview

Unmanned aerial vehicles (UAVs) are increasingly being used in real-world applications, mostly military. Currently operating UAVs use rudimentary guidance technologies, such as following pre-planned or manually provided waypoints. Such operation modalities only allow limited operational flexibility and restrict applications to ones such as high-altitude reconnaissance, exploration, target assignment. Plans for future more elaborate roles, have been outlined (see the UAV Roadmap 2002-2027[8]). In order to allow more complex missions or task, UAVs will require more advanced guidance capabilities, in particular ones that increase the vehicle autonomy, i.e., have the ability to tackle a number of operational events without operator input, including, for example, trajectory replanning following the detection of a threat, or changes in the mission or environment configuration. Over the past years, advances in software and computational power have fuelled the development of a variety of new guidance methodologies for UAVs. The availability of various guidance technologies and the understanding of operational specification create opportunities to work towards refining and implementing such advanced guidance concepts. To this date very few have been flight-test evaluated on actual vehicles. Planning the UAV flight path, or trajectory, is the chief problem in autonomous UAV deployment. The basic difficulties include partially known and changing environments, extraneous factors, such as threats, evolving mission elements, tight timing and positioning requirements. Moreover, these must be tackled, while at the same time, explicitly accounting for the actual vehicle maneuvering capabilities,including dynamics and flight-envelope constraints. The trajectory planning problem translates into a mathematical optimization problem, Typically, however, it is too complex, i.e., solution computationally too demanding, to be solved and implemented in real-time. To address this issue, guidance problems can be simplified for computational tractability. Milam *et al.*,[?] propose a two-degree of freedom approach, consisting of a trajectory generator and a linear feedback controller. The trajectory generator provides a feasible feed-forward control- and reference trajectory. To allow computational tractability, the trajectory generator uses a lower dimensional representation of the system, allowing to map, the system equations, cost functional, and vehicle constraints into an output space. The output trajectories are parameterized in term of B-spline curves. The optimization in the output space, parameterized in terms of the B-spline coefficients, is solved using sequential quadratic programming. Hauser *et al.*[?] proposes a technique where the trajectory generation is solved in two steps. First the trajectory is generated for a sim-

*Post-doctoral associate, Laboratory for Information and Decision Systems, Department of Aeronautics and Astronautics bmettler@mit.edu; corresponding author.

†Doctoral candidate, Laboratory for Information and Decision Systems, Department of Aeronautics and Astronautics toms@mit.edu

‡Associate Professor of Aeronautics and Astronautics, Space Systems Laboratory. jhow@mit.edu

§Associate Technical Fellow, Boeing Phantom Works james.l.paunicka@boeing.com

¶Associate Professor of Aeronautics and Astronautics, Laboratory for Information and Decision Systems. feron@mit.edu

plified model of the actual vehicle, and subsequently, a morphing technique[7] is used to obtain the trajectory for the detailed model of the vehicle. The simplified model captures the essential dynamics and features of the full order system. In some cases a receding horizon scheme can be used for the morphing. Alternatively, they can be solved offline and then stored in the on-board computer in a practical form. Frazzoli *et al.*[4] propose an approach where the vehicle dynamics is represented through a hybrid maneuver automaton. This representation allows to capture broad vehicle maneuvering capabilities and lends itself to efficient real-time trajectory planning by precomputing an optimal cost-to-go function. This method was evaluated for a highly agile miniature rotorcraft using a high-fidelity simulator.[7] Most guidance methodologies fall either in a computational intensive category (real-time optimization), or a memory intensive category (pre-computation). Each, however, has its particular advantages and shortcomings resulting from the way the problem is formulated and solved. In the following we plan to demonstrate a planning technique that uses real-time optimization as well pre-computed elements. A large part of our effort will be to understand future UAV guidance requirements and effectively handling the computational limitations. Besides providing a concrete example of implementation of advancedguidance methods to a full-scale UAV, this exercise will also help us define meaningful directions for future research in UAV guidance algorithms and help define computational requirements. The activities described in this paper are part of the Defense Advanced Research Program Agency (DARPA) Software enabled control (SEC) program (see[1] for an overview). An advanced guidance system involves software components besides the actual trajectory generation algorithm. The goal of our activities is to transition, and eventually demonstrate, enhanced guidance capability on a full-scale vehicle equipped with actual future UAV avionics and software. The paper begins with a brief overview of UAV guidance requirements, starting with today's UAV operation, and likely directions for future guidance systems. We follow with a description of Boeing's UAV flight-test vehicle, including its control modalities and control interface, around which the guidance system will be designed. Then we describe the receding horizon MILP guidance system and the capabilities that are enabled. Following we give an outline of the flight test scenarios which will be used to demonstrate and evaluate our technology. We conclude with an outline of the technology build-up and transition milestones.

UAV Autonomous Guidance: Requirements and Build-Up

Today's UAVs are principally surveillance and reconnaissance vehicles that are operated remotely by a human operator from a ground control station; they have no on-board guidance capabilities that would give them some level of autonomy, for example, to re-plan a trajectory in the event of a change in the environment or mission. With such rudimentary capabilities, only simple tasks can be accomplished, and the operation is also limited to simple, largely known environments. UAV autonomy has to be developed to accomplish more complex tasks, operate in uncertain, changing environments, and eventually allow integration with other UAVs and manned vehicles to accomplish coordinated mission.

Ways to specify and determine the levels of autonomy have been proposed[3,8] [need to say more]. These, however, are difficult to apply; there seems to be no natural rule that dictates in which order the autonomy has to develop. UAV autonomy build up will happen in a gradual manner, to develop the appropriate technologies, we need to understand the basic UAVs operational modalities. In the following, we first layout a simple operational UAV framework, which will help us develop the guidance system and test scenarios that we are planning to use in our flight-test demonstration.

Autonomy Build Up

To layout the framework, we can look at a simple example in which a UAV has to execute a particular task. The following narratives describe different operational modalities in order of what we would consider a growing autonomy level:

1. Operator guides the UAV to the task area by placing waypoints. He monitors the mission continuously as it unfolds, looking at the information gathered from various sources. He takes direct actions in the events of threats or changes in the mission specification. At the task location he commands the payload. The same process is followed on the return flight.

2. UAV guides itself to the task area using an onboard autonomous guidance system, which determines the vehicle path based on the sensed environment and operational objectives. Operator is prompted in the event of threats, and/or changes in mission and before payload is activated. He monitors proper action and takes over in case of unresolved situations.

3. UAV guides itself to the task area using an onboard autonomous guidance system. UAV adapts its behavior to context, managing the risk/performance to maximize mission outcome. Operator is only present and active at the mission level.

The main elements in the example are the operational objectives, the environmental elements, and the vehicle capabilities. We can characterize the autonomy based on the complexity of these elements, the a priori knowledge about these elements in the particular mission to be executed and the uncertainty in the knowledge. As the autonomy increases, the UAV can operate in more complex environments, execute more complex tasks, and the vehicle can display a broader range of behaviors. Also, less a priori knowledge is necessary and some degree of uncertainty is tolerated. Finally, the amount and type of interaction required from the operator changes. Table 1 illustrates the principle based on the example's characteristics and the following sections. Bellow, we also give more details on the three operational elements.

Operator Modality

The amount and type of operator interaction changes as the autonomy capabilities increase. The operator modality shifts from a continuous, low level, direct, to an event based more abstract interaction. Since the operator workload and attention capacity is limited, less workload is important to enable more complex tasks to be executed. Also, a single operator to supervise multiple vehicles collaborating in a same mission.

Operation

From an operational viewpoint, a number of objectives may need to be satisfied, examples include: 4-D planning (3-D plus time, e.g., rendezvous task), radar signature minimization, or, if the UAV is enslaved to a payload (e.g., directed energy weapons), kinematic constraints will have to be satisfied in that particular stage of the mission. Also it is desirable to be able to adjust the dynamic behavior, i.e., flight performance, to a particular operational context. For example, a mundane traveling task may require a different behavior than a high-risk infiltration or reconnaissance task.

Environment

Both the characteristics of the environment and the knowledge about it, at a given point in time. For example, the UAV has to satisfy airspace constraints. These can originate from mission specifications or others like meteorological conditions. Also if operating at lower altitude, the terrain also becomes a relevant characteristic. Finally, as part of the environment we can also include other constraints, including enemy threats. These environmental features are not always perfectly known and some may change over time.

Vehicle Capabilities

Some UAVs will have to perform tasks that require higher maneuverability. This may be the case for unmanned combat aerial vehicles (UCAVs), or smaller UAVs operating in challenging terrain conditions like urban canyons. Such types of operations will require guidance systems to plan trajectories that can exploit the full vehicle capabilities.

Primary Autonomy Capabilities

Based on the state of the art in UAV autonomy, the immediate objective would be to gradually increase the complexity of the environment and tasks, and, at the same time, incorporate a stochastic components. Our primary requirements for the guidance system are:

- Exploit the vehicle capabilities
- Dynamically integrate task and mission elements
- Rapidly respond to changes in situational awareness

Autonomous control architectures typically have a hierarchical structure.[7] However, beyond the primary capabilities, e.g., multi-vehicle coordination, task allocation, a number of command and sensing structures could be used. For example centralized or distributed. These are typically imposed by the type of missions that are flown. In the following we focus at how our technology satisfies the primary requirements. For the higher level ones the main question is whether our technology provides enough flexibility to accommodate or support potential high-level command and sensing structure.

Description of UAV Test-Bed

Flight-Test Vehicle

The flight-test vehicle is a modified T-33 aircraft equipped with an autopilot and components of the actual Boeing's UAV avionics. Figure 1 shows a picture of the aircraft on the ground. The autopilot support different control modalities, including: waypoint following, as well as, turn-rate, altitude, and velocity set and hold. These control modalities are accessible through Boeing's open control platform (OCP).[9]

A standard laptop computer will serve as the onboard flight processor. It will host the OCP, and the different software components of the guidance algorithms. Our demonstration will concentrate on horizontal flight. The guidance system will produce the turn rate and velocity commands of the aircraft to follow the generated trajectory.

In the initial design and test phase, a simulation model of the T-33 with the autopilot provided by Boeing will be used. The guidance system will be designed based on a hybrid model of the vehicle's closed-loop dynamics as well as precise specifications on the control inputs and vehicle states constraints, which define the operational envelope. The OCP which also serves as software integration framework for software development and desktop simulations.

Table 1 Main elements in autonomous UAV operation and operator modality

	Operator Modality	Operational Objectives	Environmental Features	Vehicle Capbilities
1.	continuous and direct	none/operator	known/operator	waypoints
2.	event based, partially direct	basic	mostly known	maneuvers
3.	task/mission based	advanced	partially known	full envelope

Fig. 1 Picture of the Lockheed T-33 on the ground.

Description of the Guidance Technologies

MILP Technology Background

We plan to demonstrate a guidance technology based on mixed integer-linear programming (MILP). MILP is a mathematical programming technique that is used to solve problems combining linear and integer decision variables. This technology is extensively used in operations research, so several effective numerical solvers are available (e.g. the commercial package CPLEX by ILOG[6]).

Over the past several years, a number of vehicle path planning problems have been formulated using a MILP framework. This approach's key is that it allows to explicitly account for constraints and binary decision variables. These are used in the context of trajectory planning to take care of the vehicle maneuvering limitations and the obstacles in the environment to generate collision-free trajectories. The MILP framework can also account for a number of mission and operational constraints while coordinating multiple vehicles.

All this is contingent on formulating the problem in a way that is compatible with the MILP framework. The approach was first used to compute the full set of trajectories off-line, but a receding horizon formulation was developed to permit on-line planning. This step was accomplished by combining the real-time optimization over a short planning horizon with an estimate of the cost-to-go from various terminal points.[2] The cost-to-go estimate is developed on-line (but outside of the MILP calculation) using a coarse grid of the points associated with the obstacle corners. The MILP then joins the detailed dynamic part of the trajectory with the cost-to-go points using a projection of the terminal velocity.[2] This RH formulation enables real-time path planning and improves the responsiveness of the planner to changes in the vehicle team or environment. The MILP approach has recently been extended to guarantee the safety of the trajectory[14] and to include a hybrid representation of the vehicle dynamics which combines linear time invariant (LTI) models and agile maneuvers.[15] This modification enables trajectory planning for vehicles with a broad range of behaviors, such as rotorcraft. MILP-based trajectory planning has already been successfully implemented on several test-vehicles at MIT,[11] and has been simulated on spacecraft as well as other air and ground vehicles.

Key Technology Characteristics

For the flight demonstration we plan to implement an extended version of the MILP formulation. The main features of the trajectory planner are:

- Receding horizon planning. Planning is performed in real-time by solving the MILP over a finite prediction horizon (receding horizon mixed integer-linear programming (RH-MILP)). The terminal cost is extracted from a cost-to-go function that is computed based on the known environment configuration. This function is recomputed whenever changes to the environment configuration occur.

- Planning using a hybrid vehicle model. The model combines multiple, linear time invariant (LTI) modes that capture the vehicle flight dynamics in the portion used for the flight demonstration, (and agile maneuvers for trajectories beyond linear dynamics. At each time step, the planner can choose between staying on in LTI mode or invoke a more agile maneuver.)

- Safe trajectories are computed explicitly at each time step. These safety, or rescue/fail-safe trajectories, can be invoked in the event of a problem, such as failure to compute a feasible trajectory over the following sampling period.

- Tasks can be inserted at specified times and/or geographical locations. The trajectory planner generates a trajectory that satisfies the various task requirements (e.g. ingress and egress conditions). For the purpose of trajectory planning, the tasks are represented by primitives that are compatible with the MILP formulation. This allows complex tasks to be integrated in the trajectory planning while preserving computational efficiency.

In the following we outline these features in more details.

Receding Horizon MILP

Solving for the trajectory over the full mission length is computationally too expensive and is also not meaningful given that specifications may change, the knowledge of the environment may evolve and disturbances as well as uncertainties would affect the accuracy of the plan. To address this issue, the trajectory computation is broken into two portions: (a) a finite-duration segment obtained from solving a MILP in real-time over receding prediction horizon, and a trajectory approximation covering the rest of the duration. For the implementation, the trajectory approximation is represented as a terminal cost in (a), obtained from a cost-to-go function computed based on the known environment configuration and mission specifications. This function is recomputed whenever changes to the environment configuration occur. In the receding horizon (RH) MILP, at each decision step decide to stay in an LTI mode, OR execute maneuver (IF initial conditions are OK).

Planning with a Hybrid Vehicle Model

Preliminary simulation experiments have shown that the turn rate and velocity dynamics can be modeled using low order linear time invariant (LTI) models. Several of these models will be used to accurately model the dynamic responses over the aircraft speed range used in the flight experiment. Figure **??** shows simulated vehicle responses from Boeing's hi-fidelity simulator.

[put some plots of the vehicle response ???]

The LTI models describe the vehicle dynamic response under the autopilot in turn rate and velocity command mode. For example, with first-order response types for the turn-rate, velocity we would have:

$$\dot{\chi} = \frac{\tau_\chi}{s + \tau_\chi} \tag{1}$$

$$\dot{V} = \frac{\tau_V}{s + \tau_V} \tag{2}$$

The operational range of each LTI mode is described by constraints of the type

$$\dot{\chi} \leq \dot{\chi}_{max} \tag{3}$$

$$\dot{V} \leq \dot{V}_{max} \tag{4}$$

$$V_{min} \leq V \leq V_{max} \tag{5}$$

In addition to the multiple LTI modes, maneuver primitives may be used to represent the vehicle capabilities beyond its linear range. The hybrid LTI-Maneuver model achieves two purposes, first, it helps to cover a large portion of the flight envelope, second, maneuvers can be used to encapsulate actions that have a specific operational purpose (e.g. rapid turns, spiral dives). Figure 2 shows a graph representation of the hybrid UAV model and Figures 3 and 4 show two different turn maneuvers. The first, is a conventional 90 deg turn. Compared to a turn in LTI mode, it could be designed to best exploit the vehicle maneuverability, which, for example, could be used when the UAV has to avoid a threat area. The second, is an atypical turn which illustrates a maneuver that could be of particular operational interest. Maneuvers could be implemented through dedicated control logics that are tailored to the specific characteristics of the maneuver.[5]

For the purpose of planning, maneuvers are represented by primitives. These only contain the parameters that are relevant to planning, e.g. displacement incurred (i.e. discreet state transition), duration (which can be used as a measure of performance or cost). Finally, this representation is compatible with the MILP framework. Note that such a representation is compatible with many existing flight control systems.

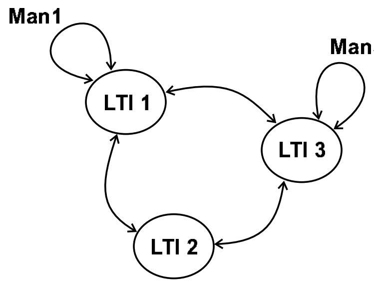

Fig. 2 Graph of the hybrid UAV model with multiple LTI modes and maneuvers.

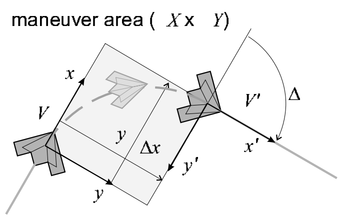

Fig. 3 Maneuver primitive for a sharp 90 deg turn.

Tasks Insertion

UAVs will be used in the future to accomplish more complex tasks. Possible task include the engagement of special sensors or weapons that require flying the object of interest or target.

Planning a trajectory to a task area results in additional constraints for the planner. Same as for the maneuvers, each task is represented by a primitive. These capture the structure and characteristics of the task that are relevant for the planning of the trajectory to and from the task. For example, the in- and egress conditions (e.g., heading, velocity, altitude), the required free airspace. Figures 5 and 6 show two examples of task primitives. Here again, we assume specific control logic is used.

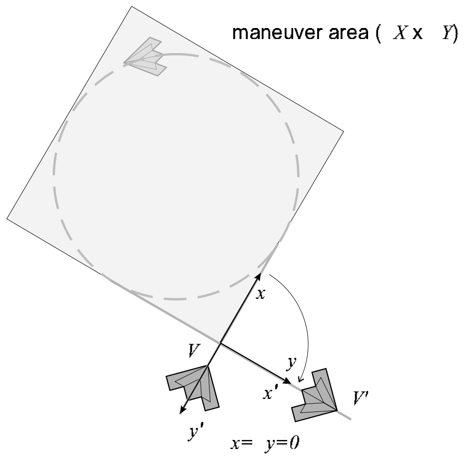

Fig. 4 Maneuver primitive for an atypical 90 deg **turn.**

The high precision trajectories that are used for the task are implemented by dedicated control logic, similar to agile maneuvers. This again significantly simplifies the complexity of the trajectory planning.

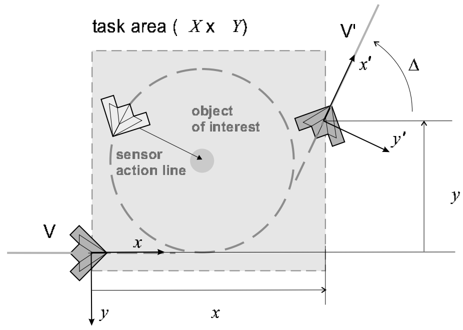

Fig. 5 Task primitive for a sensor engagement.

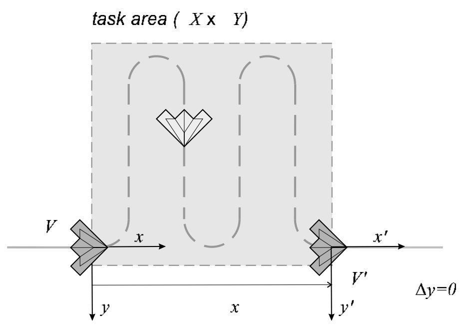

Fig. 6 Task primitive for an aerial search.

Maneuver and Task Primitives

Basically maneuver and task primitives perform a hierarchical break-down of the different mission stages: traveling, handling extraneous events, executing tasks. Maneuvers and Tasks can be organized by type. Within one type, some can be parameterized to cover a continuum of execution. Library of maneuvers or task primitives can be modified dynamically This allows to control which primitives can be used as a function of the context. A possible benefit of this is risk/performance management. Since the motion primitives are a repertoire of all behaviors available to the guidance system, it is straightforward to perform mode switching and even reconfiguration by switching between different primitives libraries. These switches can be triggered based on the operational context or based on failures. Figure **??** shows an example where a reduced set of maneuver primitives is used (e.g. in case of a failure limiting the maximum aircraft turn rate).

Trajectory Safety

At each trajectory computation, the feasibility of a rescue trajectory must be satisfied. This guarantees that given the knowledge of the environment, a rescue path can be executed. Basically, the UAV will only if it can safe. Figure 7 shows the concept.

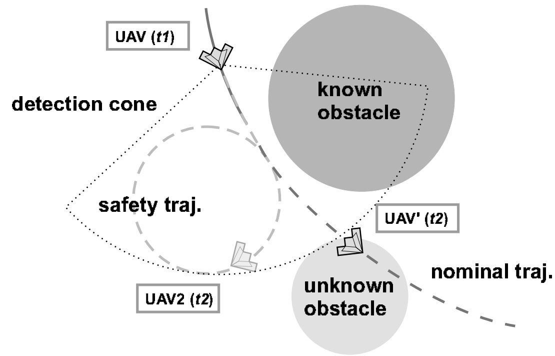

Fig. 7 Safety trajectory. Planning algorithm always enforces the feasibility of a rescue path within the known portion of the airspace (e.g. on-board sensor detection range).

For this demonstration we plan to use circular loiter, or holding patterns as safety path. Note that other patterns can be used. These depend on the vehicle capabilities. A rotorcraft for example will be able to perform a hover. These could also be represented by primitives and stored in a safety-state library, providing a range of possibilities.

MILP Formulation

To formulate the aircraft guidance problem into a linear program with mixed integer/linear constraints requires a series of transformation and approximations. The vehicle dynamics are expressed through discrete-time linear equations of motion, acting as an equality constraint in the problem. Typically, the dynamics are subject to flight envelope constraints that require the states and inputs to belong to a given interval.

Obstacles and other operational constraints that have non-convex feasible regions are accounted for using multiple linear constraints together with an integer slack variable. For example, in Schouwenaars *et al.*,[13] obstacles are represented by four sided rectangular shapes using four inequality constraints and four binary variables. In Richards *et al.*,[10] the velocity and turn rate (or force) constraints are expressed as circular constraints approximated by a M sides polygon. The addition of specification to the problem will result in the addition of several mixed integer/linear constraints, significantly increasing the size of the problem to solve.

The linear objective function can be formulated to result in minimum fuel or minimum time trajectories. The optimization problem is then solved over a fixed time horizon. The length of the planning horizon directly influences the solution time. If the environment is known a priori, the collision-free guidance problem can be simplified. One technique is to pre-compute an approximate cost-to-go function, which allows the prediction horizon to be reduced to a local region,[2] and has shown to offer significant advantage for computation.

Figure 8 shows an obstacle-free trajectory planned between two waypoints; at the second waypoint the aircraft must head north (heading constraint).

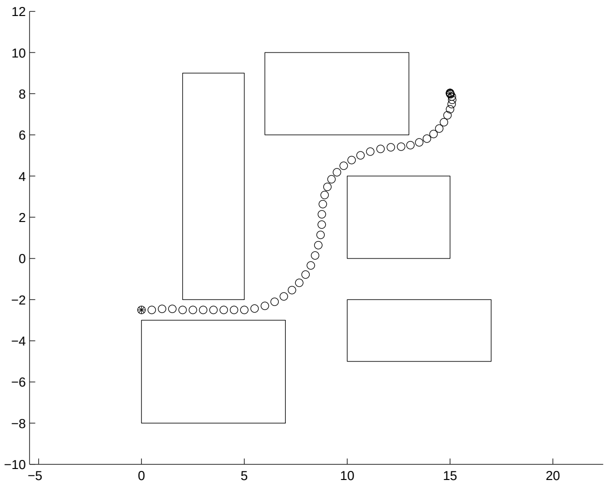

Fig. 8 Trajectory between two waypoints, where the airplane flies from left to right and must head north at the second waypoint (heading constraint).

Enabled Capabilities

By using a hybrid vehicle model, the RH-MILP planner can make full use of the vehicle dynamic capabilities when planning a trajectory. Moreover, if available, agile maneuvers can be invoked, allowing vehicle maneuvers beyond the linear range of the dynamics, thus enhancing the vehicles ability to react to unforeseen situations.

In regard to planning a trajectory to perform a task, or a mission, the RH-MILP technology can efficiently integrate information about the immediate and global environment and task or mission elements. This information can then readily be accounted for in the computation of the vehicle trajectory.

At the global level, the environment and mission elements are translated in a cost-to-go map. At the local level (within the sensing/perceptual range of the UAV), the information is taken into account explicitly in the planning horizon. These features enable the following primary autonomy capabilities (or skills), namely:

- Reactive trajectory planning. Any new information about the immediate vicinity of the UAV, e.g., provided by on-board sensors, is used explicitly for the trajectory computation. For example, in the event of a pop-up threat, assuming it is detected, the trajectory will avoid the threat area.
- Dynamic re-planning. Changes in the global mission parameters are integrated into the trajectory generation in real-time. This is performed by re-computing the global cost-to-go map. Thus, changes in the mission are immediately accounted for in the trajectory.
- Dynamic task insertion. A task is accounted for in the planner as an element that can be inserted or changed dynamically.
- Safe trajectory. At every time step, the trajectory can be aborted to enter a safe trajectory (loiter pattern). This feature can be used in the event no feasible trajectory could be computed within the next planning interval. The above capabilities are the foundation of the higher level ones, including the cooperative dynamic planning with two vehicles that will be shown as part of the flight demonstration.

Implementation

When developing a guidance technology, other aspects are also relevant. Some of these include:

- compatible with existing flight control facilities
- transparent (certification)
- supports gradual autonomy build-up
- integrate with evolving vehicle capabilities

Integration with the OCP

A key part of our effort will be to fit the guidance technology within the computing constraints, i.e., to take best advantage the resources to meet operational requirements. The computational constraints (data rates, processor speed, memory) will impose limitations on the amount of optimization that can be performed online.

For the online optimization based approach, we will have to determine the level of complexity that can be solved in real time, and how it impacts the resulting performance of the guidance system. The mixed linear/integer guidance programs were solved using the commercial CPLEX software package. Empirical studies of the computational requirements as a function of complexity (in terms of number of obstacles and number of steps in the prediction horizon) have been presented in the work of Bellingham *et al.*[2] and Richards *et al.*[12]

Receding Horizon API

To support receding horizon planning, the OCP will include a receding horizon application program interface (API).

Demonstration Scenarios

The flight testing will take place at Edward's Air force Base. The following provides the templates for these scenario prototypes. Two scenarios are planned. The first will be used to exercise the primary autonomy capabilities, and the second will be used to exercise extended capabilities.

First Scenario

The objective of the first scenario is to exercise and evaluate the primary autonomy capabilities. The scenario provides a template describing the key elements and events.

Mission: UAV has to accomplish a task at a predefined location. The environment is largely known. One or two unknown threat regions may be activated. The task is known a priori.

Assets: T-33 as UAV.

Timeline of Events

Starting condition: UAV1 loiters in predetermined pattern (sequence of waypoints). Environment is partially known; one or two pop-up threats are expected on route.

Narrative: (t_1) UAV leaves loiter area on a waypoint trajectory computed by the trajectory planner. On the way to the task area, the trajectory planning explicitly accounts for known threat areas (or no-fly zones), generating a trajectory that anticipates them. (t_2) If a pop-up threat is detected; the planner will generate a trajectory that avoids the threat area. (t_3) The UAV is guided to the task to meet the task's ingress conditions; executes the specified task (in the demonstration the task is not executed explicitly), while satisfying the task boundary conditions; (t_4) exits the task area satisfying egress conditions; (t_5) and returns to the loiter area, following the same process as to the task.

Second Scenario

The purpose of the second scenario is to exercise the primary autonomy capabilities in a more dynamic setting. This is demonstrated by introducing/modifying tasks and activating threat areas as the mission unfolds. Moreover, we use the F-15 to play the role of

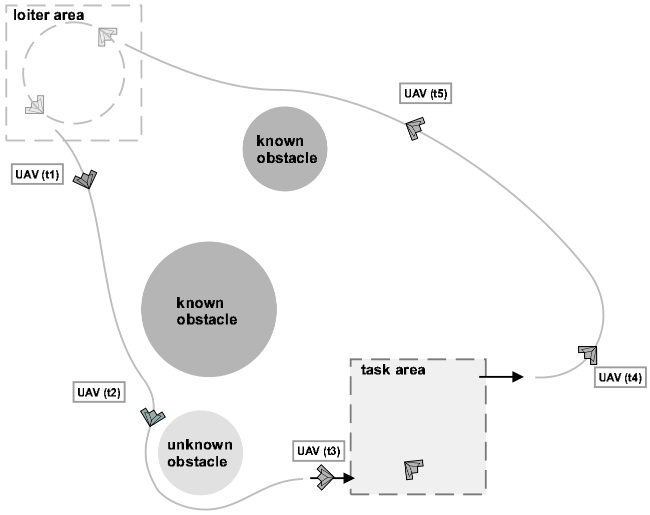

Fig. 9 Illustration of the first demonstration scenario prototype.

an exploratory UAV that can relay information about the environment and the mission back to the UAV.

Mission: Two UAVs are engaged to accomplish a task. One is an exploratory vehicle, used to gather information about the environment and the task to be accomplished. The second is setup to execute the task.

Assets: T-33 as UAV2 (task) and piloted F-15 as UAV1 (exploratory).

Timeline of Events

Starting condition: UAV1 and UAV2 loiter in predetermined pattern (sequence of waypoints). Environment is partially known. One or two pop-up threats are expected on route, also, the task location and type is not known precisely.

Narrative: (t_0) UAV1 (emulated by the F-15) leaves ahead of UAV2 (T-33) on exploratory flight towards task area. (t_1) UAV1 relays the coordinates of a threat unknown to UAV2 which just left the loiter area. (t_2) UAV2, integrates threat dynamically and generates a new trajectory that anticipates the threat. (t_2) UAV1 reaches the task area and identifies the exact task and location as well as surroundings; the information is relayed to UAV2 which integrates it dynamically in the trajectory generation. (t_3) UAV1 returns to loiter area; UAV2 executes task. (t_4) UAV1 reaches loiter area; UAV2 exits task area and returns to loiter area.

Activity Outline

MIT development phase: spring-summer 2003

May-June 03 Finalize demo scenarios based on identified vehicle characteristics and Boeing's specifications (geometry of the test area). Simulation of the demonstration scenarios in development environment and scaled down implementation on MIT's fixed wing test bed.

June-July 03 Identification of Boeing's T-33 UAV

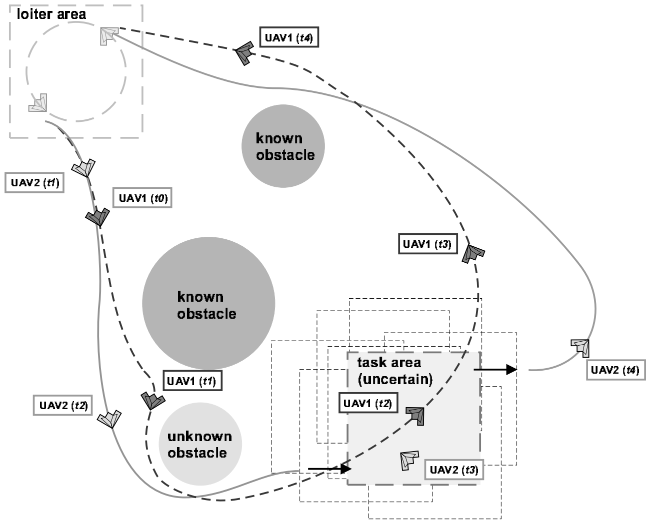

Fig. 10 Illustration of the second demonstration scenario prototype.

test bed response characteristics across experimental flight envelope from the DEMOSIM open vehicle simulation. Development of the RH-MILP Software modules for the integration with the RHC-API

July 03 Transition to the RHC-API (Flight experiments of a MA-MILP trajectory planner on MIT's autonomous helicopter)

Handoff and Integration

Aug. 03 Handover to Boeing for integration into flight-test environment (one grad student will spend the month at Boeing Saint-Louis). Desktop simulation using OCP and RHC-API

Sept. 03 Dome simulation. Modifications based on Boeing's feedback

Jan.-Mar 04 Integration into flight-hardware ground simulation for further testing (HILSIM)

Mar.-June 04 Scenario training and preview for pilot and Weapon System Officer (WSO) in domed simulations

June 04 Flight-test demonstration on actual aircraft

Acknowledgements

The projects is made possible thanks to the funding from DARPA SEC F33615-01-C-1850.

References

[1]Tariq Samad adn Gary Balas. *Software-Enabled Control; Information Technology for Dynamical Systems.* IEEE Press. John Wiley & Sons, Inc., Hoboken, NJ, 2003.

[2]J. Bellingham, A. Richards, and J.P. How. "Receding Horizon Control of Autonomous Aerial Vehicles". American Control Conference, 2002.

[3]B. T. Clough. "Metrics, Schmetrics! How Do You Track A UAV's Autonomy?". Monterrey, CA, Aug. 2002. AIAA Conference on Guidance Navigation and Control.

[4]E. Frazzoli, M. A. Dahleh, and E. Feron. "A Hybrid Control Architecture for Aggressive Maneuvering of Autonomous Helicopter. Phoenix, AZ, December 1999. IEEE Conference on Decision and Control.

[5]V. Gavrilets, I. Martinos, B. Mettler, and E. Feron. "Aggressive Maneuvering Flight Tests of a Miniature Robotic Helicopter". Sant'Angelo d'Ischia, Italy, July 2002. Proceedings of the 8th International Symposium on Experimental Robotics.

[6]ILOG. "ILOG Optimization Suite; White Paper". Technical report, ILOG, Mountain View, CA, 2001.

[7]B. Mettler, M. Valenti, T. Schouwenaars, E. Frazzoli, and E. Feron. "Rotorcraft Motion Planing for Agile Maneuvering". Montreal, Canada, June 2002. Proceedings of the 58th Forum of the American Helicopter Society.

[8]Office of the Secretary of Defense. "Unmanned Aerial Vehicles Roadmap 2002 - 2027". Technical report, Department of Defence, 2002.

[9]J.L. Paunicka, W.K. Koenig, and B.R. Mendel. "A Revolutionary Control Framework for UAVs and Other Embeded Systems". Phoenix, AZ, May 2003. Forum of the American Helicopter Society.

[10]A. Richards and J.P. How. "Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming". Monterrey, CA, Aug. 2002. AIAA Conference on Guidance Navigation and Control.

[11]A. Richards, Y. Kuwata, and J.P. How. "Experimental Demonstration of Real-Time MILP Control". Austin, TX, Aug. 2003. AIAA Conference on Guidance Navigation and Control.

[12]A. Richards, T. Schouwenaars, J.P. How, and E. Feron. "Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-Integer Linear Programming". *Journal of Guidance, Control, and Dynamics*, 25(4):755–765, July-August 2002.

[13]T. Schouwenaars, B. DeMoor, E. Feron, and J. How. "Mixed integer programming for multi-vehicle path planning". Porto, Portugal, Sept. 2001. Proceedings of the 2001 European Control Conference.

[14]T. Schouwenaars, E. Feron, and J. How. "Safe Receding Horizon Path Planning For Autonomous Vehicles". 40th Annual Allerton Conference on Communication, Control, and Computing, Nov. 2002.

[15]T. Schouwenaars, B. Mettler, J. How, and E. Feron. "Hybrid Control Architecture for Full-Envelope Vehicle Guidance". Phoenix, AZ, May 2003. Forum of the American Helicopter Society.