

# Autonomous Guidance of Agile Small-Scale Rotorcraft

## Final Report on 2001-04 Activity

Bernard Mettler and Eric Feron  
Department of Aeronautics and Astronautics  
Laboratory for Information and Decision Systems  
Massachusetts Institute of Technology

To

The National Aeronautics and Space Administration  
Ames Research Center  
June 4, 2004

**Abstract:** This report describes a guidance system for agile vehicles based on a hybrid closed-loop model of the vehicle dynamics. The hybrid model represents the vehicle dynamics through a combination of linear-time-invariant control modes and pre-programmed, finite-duration maneuvers. This particular hybrid structure can be realized through a control system that combines trim controllers and a maneuvering control logic. The former enable precise trajectory tracking, and the latter enables trajectories at the edge of the vehicle capabilities. The closed-loop model is much simpler than the full vehicle equations of motion, yet it can capture a broad range of dynamic behaviors. It also supports a consistent link between the physical layer and the decision-making layer. The trajectory generation was formulated as an optimization problem using mixed-integer-linear-programming. The optimization is solved in a receding horizon fashion. Several techniques to improve the computational tractability were investigated. Simulation experiments using NASA Ames 'R-50 model show that this approach fully exploits the vehicle's agility.

## 1 Background

There exist numerous tasks and missions for unmanned aerial vehicles (UAVs) where agile maneuvering is necessary or represents a competitive advantage. Examples include: tracking of moving targets, flying in cluttered spaces (urban), tactical flight (nap of the earth). The execution of agile maneuvers typically involves both piloting skills (quickness and coordination) and higher-level thinking (knowledge of the

environment and the objectives of a mission). To benefit from their full performance potential, unmanned systems must be able to automatically determine their flight-path accounting for the dynamics of the vehicle and the information available about the physical environment and the mission or task.

In the following we first provide a short overview of key results obtained in our research on dynamics and control of miniature agile helicopters. These results establish the framework on which our guidance approach is built. In Section 2 we give an overview of the guidance problem for agile vehicles. We summarize earlier work using a maneuver automaton (MA) and introduce the more general linear-time-invariant MA (LTI-MA) used in our hybrid guidance approach. In section 3 we describe the formulation of the trajectory generation problem for the hybrid LTI-MA as an optimization problem using mixed-integer-linear-programming (MILP). We highlight some key developments that were necessary to make this method tractable in the MILP framework. Section 4 show simulation results using our guidance system on NASA Ames' small-scale R-50 helicopter model. Finally, in section 5 we provide conclusions to this work.

## 1.1 Dynamic Capabilities of Miniature Rotorcraft

Miniature rotorcraft are naturally endowed with dynamic capabilities that easily surpass those of full-size manned vehicles. A good display of these capabilities is provided by human pilots flying miniature acrobatic helicopters. We used dimensional analysis as a theoretical basis for understanding the effects of scaling and validate the vehicle behavior as seen in MIT's X-Cell helicopter and similar acrobatic hobby helicopter. We applied Froude (dynamic similarity) and Mach scaling hypothesis on models identified for Carnegie Mellon's Yamaha R-50 and MIT's X-Cell .60 [15, 14]. With this data we were able to show and validate the following trends: for a Froude-scale type model (R-50), the attitude rate sensitivity increases proportionally to  $N^{1/2}$  (the scale ratio  $N$  means  $1/N$  the rotor diameter of the full-scale prototype vehicle) and its thrust-to-weight ratio stays relatively unchanged; for a Mach-scale type model (X-Cell), the attitude rate sensitivity, as well as the thrust-to-weight ratio, increase proportionally to  $N$ . High rate sensitivity allows for quick changes in thrust and travel direction; high thrust-to-weight values are important to accelerate the vehicle, and compensate for gravitational forces. For example, the X-Cell exhibits an attitude rate sensitivity up to 200 deg/sec in pitch and roll, and a thrust-to-weight ratio exceeding 2.

## 1.2 Automatic Control of Agile Vehicles

The analysis of human control of a highly agile small-scale helicopters shows two distinct regimes: tracking of trim trajectories and maneuvering [17, 9]. The two domains are distinctly set apart in terms of control strategy and dynamic conditions:

- Tracking operations take place around trim trajectories; control around these trajectories involves continuous feedback; the dynamics are approximatively linear.
- Maneuvering actions are of finite durations and start and end on trim trajectories; the control activity typically involve large amplitude actions that result in large amplitude state changes (they often exploits the full available range of inputs and states); the dynamics accross this range is typically nonlinear; control is dominated by feed-forward actions, feedback may include discrete switching events triggered by state tresholds.

Tracking trim trajectories is a well researched area, maneuvering, however, is more challenging due to the highly nonlinear dynamics. Instead of applying traditional nonlinear control methods (e.g. feedback linearization) we developed a maneuvering control logic inspired by human control strategies [10]. The control logic combines angular rate controllers and a timing logic that allow tracking of pre-programmed reference trajectories. The amplitude and timing of these trajectories were extracted from piloted flight-test experiments; timing in the sequence is also triggered based on the vehicle states. Several acrobatic-type maneuvers were successfully implemented using this approach, including a snap roll, a hammerhead, and a split-S [11]. Trim tracking controllers (gain scheduled linear quadratic controllers) are used prior to and upon exit from the maneuver.

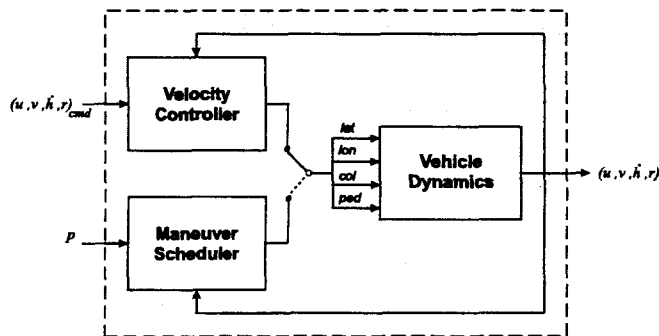


Figure 1: Hybrid control architecture: the helicopter can be controlled via velocity controllers or via a maneuver scheduler that implements pre-programmed maneuvers.

Combining trim tracking controllers and the maneuvering control logic enables a broad range of behaviors to be automatically executed. Figure 1 illustrate the implemented hybrid control architecture; Figure 2 shows this system in the form of a finite automaton: at any point in time the vehicle can either track trim trajectories using the linear tracking controllers or execute a finite number of pre-programmed maneuvers.

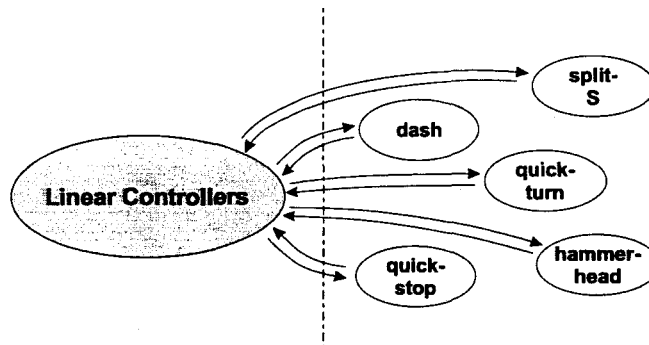


Figure 2: Abstraction of the hybrid control system as a finite state automaton.

## 2 Guidance of Agile Vehicles

Once the control architecture is available to implement a wide range of dynamic behaviors, the challenge is to exploit these capabilities to execute a task or mission in an autonomous fashion. A fundamental problem for enabling autonomous operation is that of guiding the vehicle between two geographic locations.

Motion planning for dynamic systems can be mathematically formulated using the framework of optimal control [4]. The complexity of a motion planning problem grows with the complexity of vehicle dynamics and the complexity of the environment they have to operate in. Highly agile unmanned aerial vehicles, such as small-scale rotorcraft, have complex dynamics and they may be expected to operate in the presence of stationary or even moving obstacles. Moreover, only part of the environment may be known accurately.

Solving the optimization problem in the full vehicle state-space is usually not computationally tractable for the conditions at hand. Therefore motion planning problems are often solved in some simplified or reduced form. These simplifications can easily introduce performance limitations.

### 2.1 Hybrid Guidance Philosophy

Our first experiments with a hybrid guidance approach [16], was based on the hybrid maneuver automaton (MA) [5, 7]. The main idea of the MA is to represent the vehicle dynamics through its maneuver space: the states coincide with trimmed flight, and the transitions among states coincide with maneuvers, that is, transitions between trim conditions. Taken together, the states and transitions determine the vehicle's trajectory. Such an automaton representation of the original nonlinear system represents a quantization of the original dynamics. The motion planning problem can then be solved through optimization on the maneuver space instead of the vehicle state-space.

Table 1: Description of sample maneuvers that could be implemented on a rotorcraft-type vehicle. Each maneuver has a specific utility.

Maneuver	Utility
dash to cruise	rapid acceleration from hover to one of the cruise conditions
quick stop	rapid transition from cruise to a full stop (hover)
quick Turn	rapid turn resulting in a pre-determined heading change
split-S	reversal of the flight direction with negative altitude loss
hammerhead	reversal of the flight direction with positive or no altitude loss

The state-space of a hybrid representation like the maneuver automaton is usually much smaller than that of the original system. It may thus be a more appropriate model to use for high-level activities such as trajectory planning or task selection. In some low-dimensional cases (and in the absence of obstacles) a policy based guidance solution that uses a pre-computed value function can be used [6]. This obviates the need to perform an optimization online. The value (or cost-to-go) function is calculated by applying dynamic programming techniques (value iteration [3]).

The maneuver automaton, however, has several drawbacks. These are primarily related to the fact that the vehicle dynamics are constrained to a finite set of motion primitives. In particular the continuous velocity mode is discretized into a finite number of discrete trim conditions with constant velocities. Restricting the vehicle behavior to these trims precludes the type of continuous tracking that is required for precise and smooth trajectories. In addition, to compute the value function the space in which the vehicle evolves has to be discretized. We found that the lack of continuous velocity modes and the space discretization used in the value function can be an issue when precise navigation is required [16, 23]. To achieve higher precision using a MA would require a finer quantization of its dynamics, as well as a finer discretization of the space for the value function. With such requirements, the complexity of the maneuver automaton, and the corresponding dynamic program and value function, would quickly increase beyond the practical limits.

## 2.2 Hybrid LTI-MA Representation

The combination of continuous tracking and maneuvering is key to agility. The former provides the flexible tracking capabilities required for smooth and precise flight; the latter provides dynamic capabilities that exceed what is possible under the LTI modes. Maneuvers are also interesting because they can be designed to accomplish specific operational effects, e.g., quick acceleration/deceleration, a sharp turn, or a direction reversal. Table 1 describes several maneuvers that could be used with a rotorcraft along with their operational utility. Figure 3 illustrates some of these maneuvers.

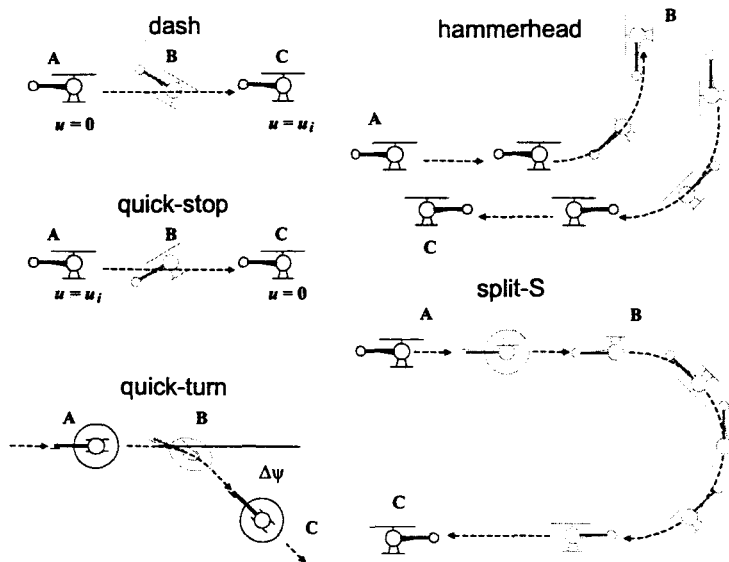


Figure 3: Illustration of maneuvers that can be used in the LTI-MA. The hammerhead and split-S have been implemented on MIT's helicopter.

The vehicle dynamics under our hybrid control architecture can be described by a finite-state automaton. Figure 4 shows a graph representation of such a machine with three LTI modes and seven maneuvers.

Flight-test experiments with MIT's hybrid control architecture [8] showed that first-order models already provide a good approximation of the body's translational (velocities  $u, v$ ) and rotational (heading rate  $r$ ) responses:

$$\begin{aligned}
 \dot{u} &= -\frac{1}{\tau_{u_i}} u + \frac{1}{\tau_{u_i}} u_{cmd} \\
 \dot{v} &= -\frac{1}{\tau_{v_i}} v + \frac{1}{\tau_{v_i}} v_{cmd} \\
 \ddot{h} &= -\frac{1}{\tau_{\dot{h}_i}} \dot{h} + \frac{1}{\tau_{\dot{h}_i}} \dot{h}_{cmd} \\
 \dot{r} &= -\frac{1}{\tau_{r_i}} r + \frac{1}{\tau_{r_i}} r_{cmd} \\
 \dot{\psi} &= r
 \end{aligned} \tag{1}$$

The vertical motion is described by the altitude rate ( $\dot{h}$ );  $\psi$  is the heading angle; the  $\tau_{\bullet,i}$  are the time constants for the different states  $\bullet$ . Several modes (indice  $i$ ) can be used to account for changing characteristics across the flight envelope.

The effect of maneuvering is represented by the state transition  $\Delta_{\bullet,p}$  incurred

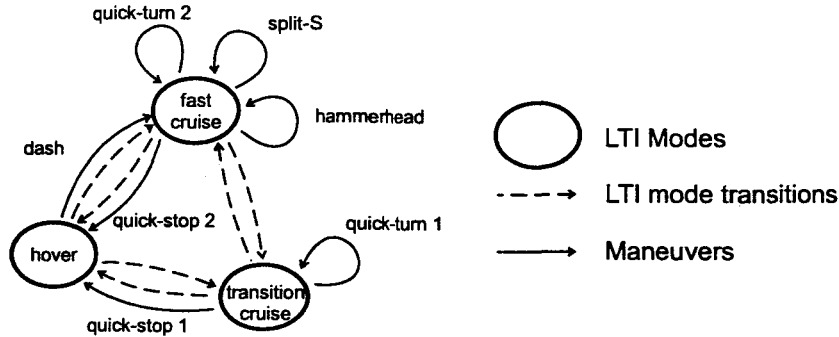


Figure 4: Graph representation of the hybrid LTI maneuver automaton. The vehicle is either in a LTI mode, tracking trim trajectories, or executing a finite duration maneuver.

during a maneuver (indice  $p$ ):

$$u(t + \Delta T_p) = u(t) + \Delta u_p \quad (2)$$

$$v(t + \Delta T_p) = v(t) + \Delta v_p \quad (3)$$

$$\dot{h}(t + \Delta T_p) = \dot{h}(t) + \Delta \dot{h}_p \quad (4)$$

$$r(t + \Delta T_p) = r(t) + \Delta r_p \quad (5)$$

along with the longitudinal  $\Delta x_p$ , lateral  $\Delta y_p$ , vertical  $\Delta h_p$  body-frame displacements, and heading change  $\Delta \psi_p$ .

The benefits of LTI-MA model are several. First, the model captures the essential vehicle behavior under our hybrid control architecture; the details about the physical layer and control system are abstracted out. Only the details that are relevant to the trajectory planning are represented. Moreover, compared to the MA, the modeling problem with the LTI-MA consist of selecting a few LTI-modes and a small set of maneuvers, thus it significantly simplifies the development of the motion primitive library.

### 3 Trajectory Generation with the LTI-MA

The generation of trajectories with a hybrid LTI-MA system can be formulated as an optimization problem using mixed-integer-linear programming (MILP). The MILP framework lends itself to the particular structure of the LTI-MA. Namely, the decision problem involves both continuous decision variable (velocity commands and turn rates in the LTI modes) and discrete decision variables (maneuvering), which are tied to logical-type conditions (e.g., whether the state satisfies the maneuvering conditions). The combination of continuous and binary (or integer) variables also

enables a number of other trajectory planning features to be explicitly accounted for in the optimization. These include, obstacles (and other terrain feature), LTI mode switching, or timing constraints. The MILP optimization problem can be solved using commercial solvers (CPLEX [13]).

### 3.1 MILP Trajectory Optimization

The general form of a mixed-integer-linear-program is:

$$\begin{aligned}
 & \min_x J(x) \\
 & \text{subject to:} \\
 & \quad \ell_1(x) \leq Mb \\
 & \text{AND } \ell_2(x) \leq M(1-b) \\
 & \quad b \in \{0, 1\}
 \end{aligned} \tag{6}$$

where  $J(x)$  is a (piecewise) linear objective function;  $\ell_1(x), \ell_2(x)$  are constraints (not limited to two), which must be linear; and  $b$  is a binary variable. We see here how logical decisions can be incorporated in the optimization problem. Namely, here the cost function  $J(x)$  has to be minimized subject to either one of two constraints  $\ell_1(x), \ell_2(x)$  on the continuous decision variable  $x$ . When  $b = 0$ , constraint  $\ell_1(x)$  must be satisfied, whereas  $\ell_2(x)$  is relaxed. Namely, if  $M$  is chosen sufficiently large,  $\ell_2(x) \leq M(1-b)$  is always satisfied independent of the value of  $x$ . The situation is reversed when  $b = 1$ . Since  $b$  can only take the binary values 0 or 1, at least one of the constraints  $\ell_1(x)$  and  $\ell_2(x)$  will be satisfied.

For the purpose of trajectory optimization the MILP minimizes a piecewise linear objective function subject to the vehicle dynamics and other constraints. We experimented with both minimum-time [22] and minimum position-error formulations for the objective function. We used the latter because it is computationally less expensive than the former, and does not dramatically affect the performance of the trajectory. The position-error objective function minimizes the 1 - norm of the error between a given waypoint location and the vehicle trajectory, over a finite  $N$ -step prediction horizon.

MILP has already been applied for trajectory optimizations. These past applications mainly focused on high-level operational aspects, such as multi-vehicle planning in the presence of known obstacles [19], mission coordination constraints [18], as well as path safety [20]. These problems were all entirely formulated in the inertial space using highly simplified vehicle dynamics. In order to account for the vehicle dynamics in the LTI-MA form we had to develop a linear approximation of the nonlinear vehicle kinematic equations.



### 3.2 Approximate Kinematics

The nonlinear vehicle kinematics relate the body-frame velocities  $u, v$  to the inertial-frame velocities  $v_N, v_E$ :

$$\begin{aligned}v_N &= u \cos \psi - v \sin \psi \\v_E &= u \sin \psi + v \cos \psi \\ \dot{\psi} &= r\end{aligned}\tag{7}$$

To obtain a linear approximation of these transformations, we segmented the continuous heading  $\psi$  in a finite number of sectors and approximated the kinematics in each sector by linear equations with constant values for the sine and cosine functions. With this approach, the nonlinear kinematic equations are replaced by a set of linear equations. Binary variables are used to track in which segment the heading is at each decision step, and enforce the corresponding set of linear kinematics. The vehicle trajectory is then computed through integration of the approximate inertial velocities.

Different levels of angular resolution can be used. The drawback is that the addition of each heading segment requires one additional binary variable (more efficient binary coding can be used [21], however, it is not clear that they improve computation time). Using this approach with few segments (8, i.e., 45 deg resolution), however, resulted in unsatisfactory errors in the predicted vehicle trajectory. We were able to reduce the prediction errors by forcing the vehicle heading to settle on the discrete heading directions at each sampling time.

### 3.3 Computational Considerations

A drawback of MILP is that the computation time increases at least polynomially with the number of variables and constraints. The optimization problem has to be solved in near real time. Solution times of about 1 sec were considered acceptable and used as criteria in our evaluation. In an effort to limit the complexity, tradeoffs inevitably take place. For example, a typical tradeoff is the one between the richness of the dynamic behavior and the granularity and length of the planning horizon.

We looked at ways to simplify the MILP formulation and reduce computation time. Increase in the computational resources may alleviate the need for such simplifications, however, we believe that techniques to reduce the computational complexity also have their merit in helping find formulations that are better tailored to the fundamental characteristics of the trajectory planning problem.

#### 3.3.1 Complexity factors and tradeoffs

For our obstacle-free MILP body-frame planner formulation, the complexity (solution time) is a function of the following dimensions: the number of decision steps in the planning horizon; the order of the vehicle's LTI dynamics; the number of continuous decision variables (control inputs); the number of maneuvers; and the number

of segments used in the heading quantization. Future formulations will include position constraints to encode obstacles, which will add to the above dimensions.

With first-order, closed-loop LTI dynamics with longitudinal, lateral and turn rate control (three decoupled first-order equations of motion, discretized at  $T_s = 1$  sec), we found that a planning horizon length of about 8 to 10 decision steps was a maximum for solution times under about one second.

Vehicles like rotorcraft exhibit a broad range of dynamic behaviors and can evolve in different types of environments. A planner that works over the various operating and environmental conditions would be ideal, however, a single formulation of the entire planning problem may not be computationally tractable. Making the trajectory planning real-time tractable involves limiting the number of variables, constraints, and length of planning horizon. To improve computational tractability we need to exploit the inherent characteristics of the planning problem.

### 3.3.2 Planning modes

The characteristics of the planning problem, over a short decision horizon, is largely a function of the flight conditions (e.g., hover or cruise) and the environment and operational context (e.g., how cluttered the space is). For example, in a cluttered space, precise trajectories that employ coordinated control actions are required; the maximum vehicle speed is typically limited. In contrast, in an obstacle-free space, the rotorcraft will typically employ a subset of the full dynamics, and it's speed will be on the faster side.

To simplify the MILP formulation we introduced planning modes that exploit the changing planning requirements as a function of the flight conditions and operational context. We designed the following three modes:

- Close-range, for full helicopter-like coordination and precise trajectories.
- Mid-range, for airplane-like flight coordination and maneuvering.
- Extended-range, for longer range planning in airplane-like flight.

The parameters for the different planning modes are shown in Table 2. The longer sampling interval used in the extended range planner allows to increase the planning range without increasing the number of steps in the planning horizon. This mode results allows a multi-resolution planning. The same principle applies to the heading quantization. All modes use the quantized kinematics with 9 segments to cover a range of  $\pm 180$ deg (45deg resolution). The mid-range maneuvering mode has 9 segments for a  $\pm 30$ deg front and rear range.

### 3.3.3 Maneuvering windows

To reduce the number of binary variables needed to keep track of the maneuvering actions along the planning horizon, maneuvering can be restricted to a specific region

Table 2: Description of the planning mode parameters. All modes use a prediction horizon of  $N = 8$  steps. The units for the control ranges are m/sec for velocities  $u$  and  $v$ , and deg/sec for turn rate  $r$ .

Mode	$T_s$	controls	resp. ranges	heading segments	maneuvers
Close-range	1 sec	$u, v, r$	$-1..3; \pm 3; \pm 90$	9 ( $\pm 180^\circ$ )	none
Mid-range	1 sec	$u, r$	$-1..10; \pm 90$	9 ( $\pm 180^\circ$ )	none
Ext.-range	2 sec	$u, r$	$-1..10; \pm 45$	9 ( $\pm 180^\circ$ )	none
Mid-range+	1 sec	$u, r$	$-1..10; \pm 90$	9 ( $\pm 30^\circ$ )	hammerhead

of the horizon. We experimented with limiting maneuvering to a small region (1-3 decision steps) in the beginning of the planning horizon. This is consistent with using maneuvers reactively, i.e., when a sudden change in the environment (such as an obstacle) requires an extreme reaction.

### 3.3.4 Planning mode selection: decision hierarchy

Each planning mode is characterized by a reachable set of destination points. Figure 5 shows a conceptual representation of the reachable sets for our different planner modes. With the extended planner reaching the furthest and the local planner with its reachable region concentrated around the origin. The conical regions correspond to the mid-range planner with maneuver, which has a tighter heading range. Note that these sets are function of the initial and final state. Figures 6 and 7 show the computed time-to-go for all destination points in the reachable sets of the close-range and mid-range with maneuver modes, respectively. The first is computed for the vehicle starting at rest, the second for the vehicle starting at 5m/sec.

In our experimental scheme the planning mode is selected based on the active reachable set (in which mode's reachable set the waypoint falls) and a factor that takes into account for the effect of vehicle state on the reachable set. If multiple modes are active, the one with the smallest reachable set is selected.

Taking some decisions outside of the actual trajectory optimization can spare significant computing time. Introducing context sensitive planning modes corresponds to decoupling certain aspects of the decision problem and introducing a decision hierarchy. Being able to recognize *a priori* certain features of the planning problem, from the vehicle state and environment features, allows to reduce the trajectory search space. For example, if it is possible to determine if the vehicle state and environment allows/precludes certain types of maneuvers or LTI behaviors, a planning mode with a specific set of maneuvers and LTI modes could be used, reducing the search space and time to compute the solution.

With our planning modes we were able to achieve computation times that are

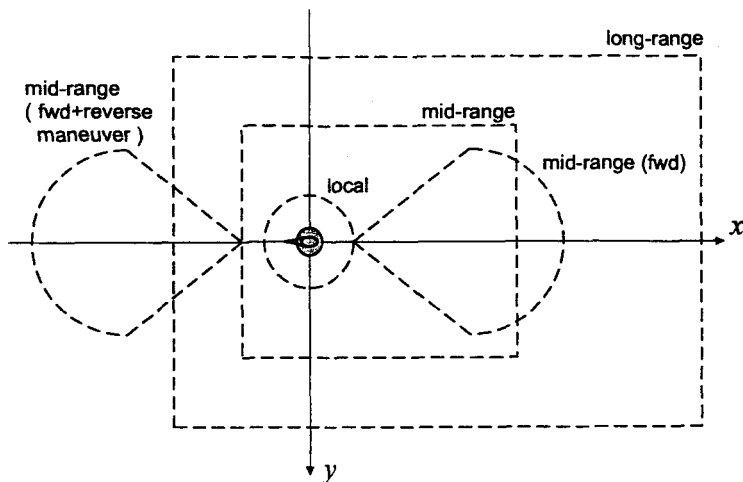


Figure 5: Conceptual representation of the reachable set of goal points for different planning modes. The modes are different in their use of control variables, commanded speed ranges, heading quantization, and available maneuvers.

within our performance criteria. Figures 8 and 9 show the solution times required for the destination points in the reachable sets of the close- and mid-range (with maneuver) modes, respectively. We can see that trajectories to all destinations in the reachable sets require less than 1 sec to compute.

## 4 Results

In this section we present simulation results that were obtained by applying our approach to the guidance of the small-scale Yamaha R-50 helicopter. The results are based on the closed-loop R-Max equations of motion provided by NASA Ames [12].

The purpose of the simulation is to demonstrate that the LTI-MA based guidance system enables trajectory planning that makes use of the vehicles agility, namely that it exploits the helicopter's multivariable control capabilities (longitudinal, lateral, and yaw motion) augmented with discrete maneuvering actions.

### 4.1 Implementation

The planner's task in our simulations is to guide the aircraft between a series of predetermined waypoints in an obstacle free environment. The planner switches from one waypoint to the next when an arrival criteria is satisfied. In the following experiments the arrival criteria, which is an attribute of each waypoint, consists of two parameters: a perimeter around the waypoint location specified by a radius

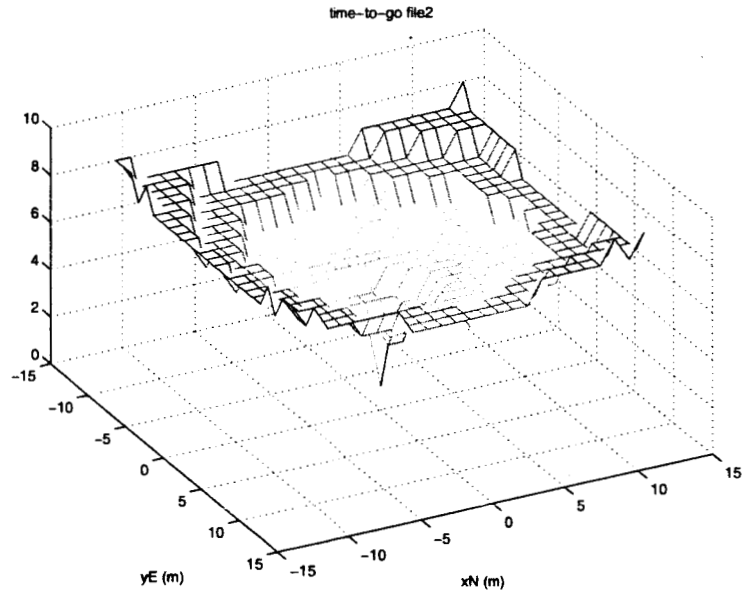


Figure 6: Reachable set for close-range mode, showing the time to reach different positions on a grid from rest at the origin  $(0,0)$ , heading north  $(x_N)$ .

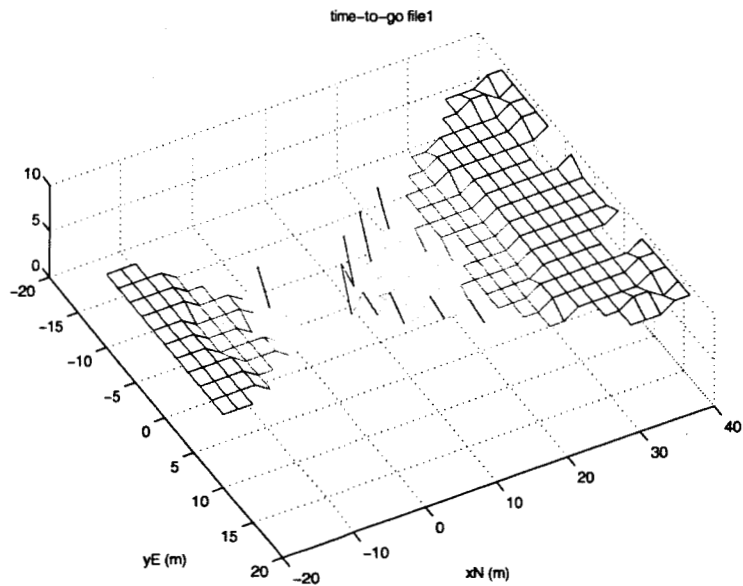


Figure 7: Reachable set for mid-range with maneuver mode, showing the time to reach different positions on a grid from the origin  $(0,0)$ , at an initial speed of 5m/sec, heading north  $(x_N)$ , with a direction reversal maneuver.

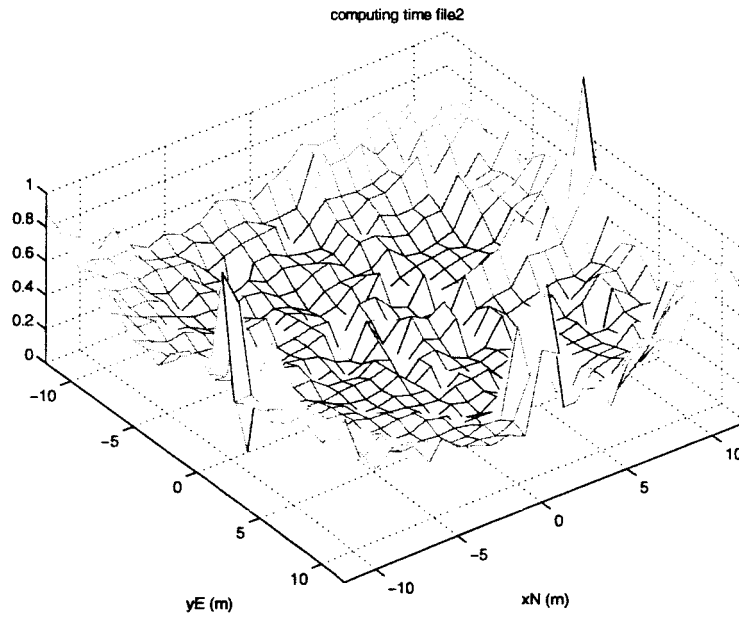


Figure 8: Solution time for the destination points in the planner's close-range mode's reachable set.

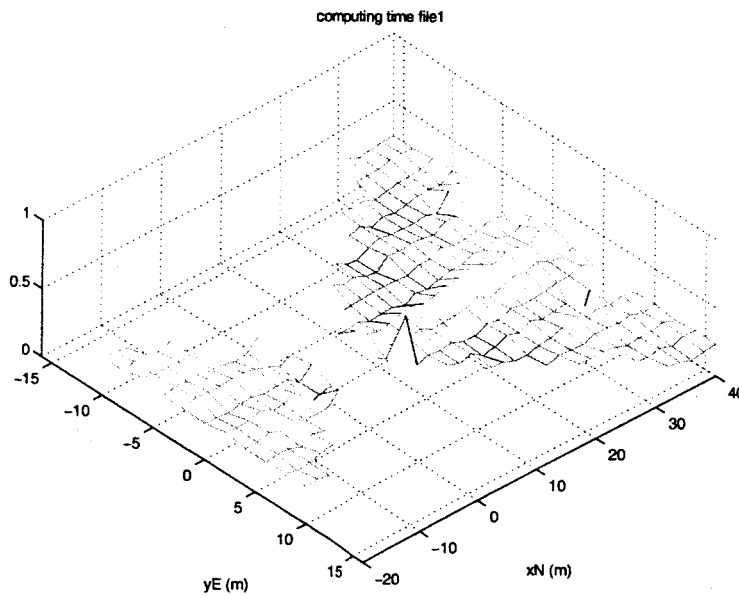


Figure 9: Solution time for the destination points in the planner's mid-range with maneuver mode's reachable set.

$R$ , and a maximum velocity, specified by a magnitude  $V$ ; both have to be satisfied simultaneously, at which instant the next waypoint coordinates are provided to the planner.

The planner computes the entire sequence of control inputs that drive the aircraft along a trajectory leading to the specified waypoint. The entire sequence could be implemented without being recomputed on the way to the waypoint. However, because of disturbances and modeling uncertainties (and approximations used in the planner) the trajectory may deviate from the nominal, planned one and miss the waypoint. To compensate for these effects, the planner is implemented in a receding horizon fashion: the planner replans a new trajectory, using the latest state information, as the vehicle moves toward the waypoint. In the following simulations, replanning was performed at every five time steps, or when a better planning mode was reached.

## 4.2 Simulations

Results from three simulation experiments are shown. In the first simulation, the helicopter is hovering at  $(0,0)$  and must go to a location 30m ahead  $(30,0)$ , and then back to the departure point  $(0,0)$ . Figure 10 shows the vehicle trajectory, figure 11 shows the commands and the active planning mode, and figure 12 shows the vehicle states. The arrival criteria for the waypoint  $(30,0)$  is  $R = 4$  meters and  $\sqrt{u^2 + v^2} \leq 2\text{m/sec}$ . The travel time for the entire trip is 18 seconds.

We can see that the planner exploits the full control input range (e.g.  $u_{cmd}$  up to 10m/sec). Also, when changing direction to go from the first waypoint  $(30,0)$  back to the departure location, the helicopter uses both heading and side slip to effectively turn back. This behavior illustrates the planner's ability to exploit the full command coordination of the vehicle's LTI velocity tracking controllers.

Figure 11 also shows how the planning mode changes as the vehicle moves toward the waypoints. The integers (1,2,3,4) correspond to the following modes: (1) close-range planner; (2) mid-range planner; (3) extended range; (4) mid-range with maneuver. We see that at the starting instant, the first waypoint falls within the mid-range mode. Then, as soon as the first few control actions are applied, the velocity increases (time step 3), the planner gets in the extended-range mode. Then, as the vehicle approaches the first waypoint, the close-range planner is enabled, allowing for more precise fully coordinated trajectories. When the return waypoint is provided, the planner goes back to extended-range mode until it approaches the final waypoint, where it goes back to the close-range mode.

The second simulation shows the same direction-reversal task, but here, the return waypoint is offset laterally from the origin  $(0,10)$ . The trajectory is shown in Figure 13, and the commands and states in Figures 14 and 15, respectively.

The last simulation shows the same direction reversal task with lateral offset, only here, the mid-range planning mode has a hammerhead maneuver available. The trajectory is shown in Figure 16, and the commands and vehicle states in

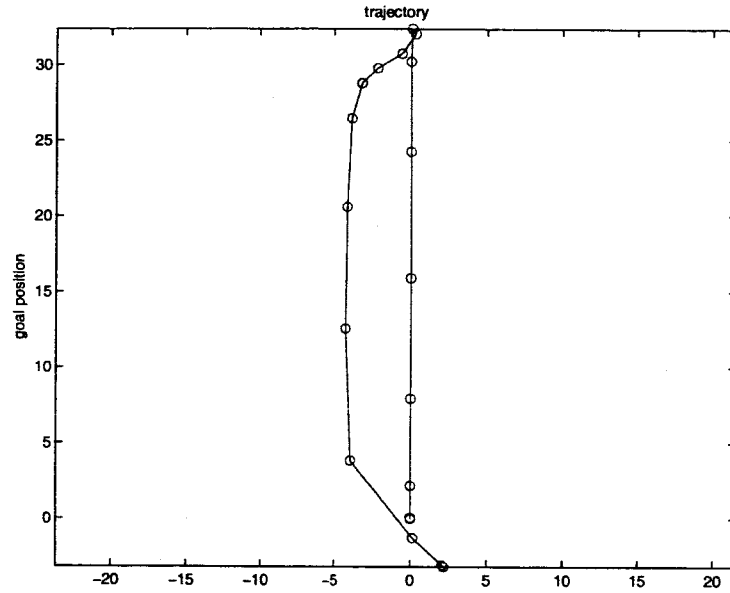


Figure 10: Vehicle trajectory for simulation 1: the helicopter starts from hover at (0, 0) and has to go to a location 30m ahead (30, 0), and then back to the departure point (0, 0).

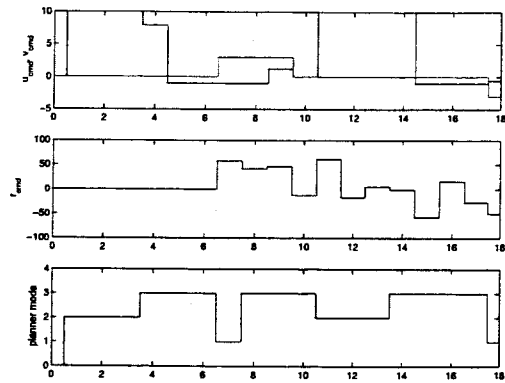


Figure 11: Commands and planner mode.

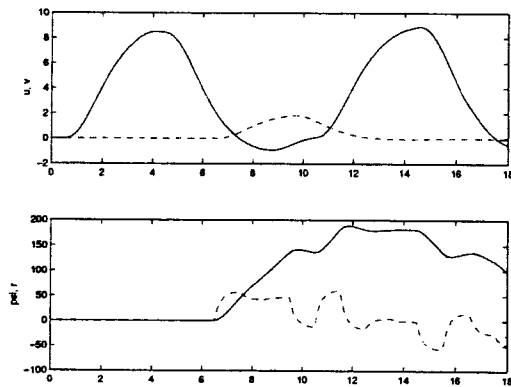


Figure 12: Helicopter states.



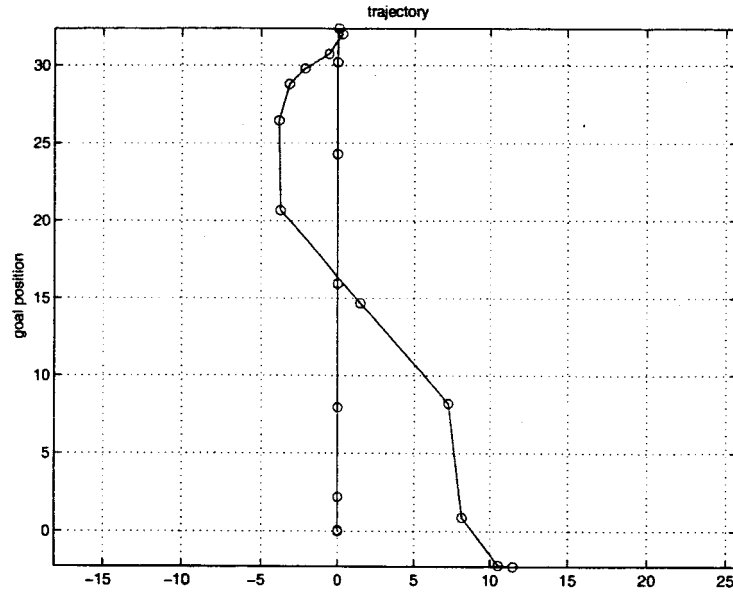


Figure 13: Vehicle trajectory for simulation 2: the helicopter starts from hover at (0,0) and has to go to a location 30m ahead (30,0), and then back to a destination offset 10m from the departure point (0,10).

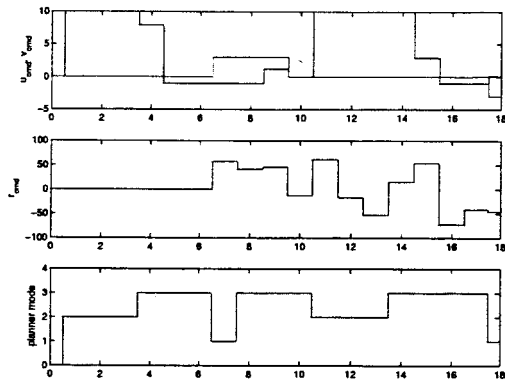


Figure 14: Commands and planner mode.

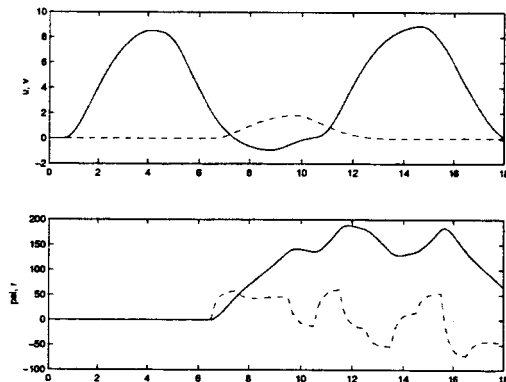


Figure 15: Helicopter states.

Figures 17 and 18, respectively. The arrival criteria for the first waypoint was set to  $R = 10$  meters and the maximum speed was relaxed, so that the vehicle approaches the waypoint with sufficient speed to initiate the hammerhead (which requires a minimum longitudinal speed  $u$  of 5m/sec). The maneuver is initiated at time step 6, while approaching the first waypoint. The maneuvering action is visible from the switching to planning mode 4 in Figure 17, and from the 180deg heading jump in Figure 18 .

### 4.3 Conclusions

The dynamics of agile vehicles can be efficiently described by a hybrid closed-loop model, combining linear time invariant (LTI) control modes with discrete maneuvering actions. This hybrid LTI-MA form was motivated by the tracking and maneuvering control modalities that were observed in the human pilot behavior during acrobatic flight experiments with a small-scale helicopter. This LTI-MA structure can be realized by a hybrid control system like the one developed at MIT, which combines LQ controllers for tracking, with a maneuvering control logic for the automatic execution of maneuvers.

Such a model of the vehicle dynamics is much simpler than the vehicle's full equations of motion, nevertheless it can describe a broad range of dynamic behaviors. In contrast to other approaches used to simplify the representation of the vehicle dynamics for purpose of motion planning, the LTI-MA reduces the size of the state-space in a meaningful way, i.e., by encoding the dynamics from a *utility* standpoint (how the vehicle is flown or used). This representation also supports a strong and consistent link between the physical layer (driven by ordinary differential equations of motion) and the high-level decision-making layer (driven by continuous and discrete, logical decision variables).

The trajectory generation problem with the LTI-MA was formulated as an optimization problem using mixed-integer-linear-programming (MILP). This approach is suited to the hybrid structure of the LTI-MA, i.e., the combination of continuous control variables (in the LTI modes) and discrete maneuvering decisions. The optimization problem has to be solved in real-time, hence computational tractability is a key issue. Several techniques were used to reduce the size of the optimization problem.

Simulation results show that the LTI-MA representation combined with the MILP optimization produces trajectories that successfully exploit the flexibility and precision of the LTI modes and the performance of the maneuvers. This approach has several other attractive attributes. The MILP framework allows to explicitly take into account other features that are relevant to trajectory generation, such as obstacles or no-fly areas. The LTI-MA/MILP formulation can also support reactive actions. As such it may also be well suited for guidance in the immediate, sensed environment.

Real-time computational tractability is a key issue for this type of guidance

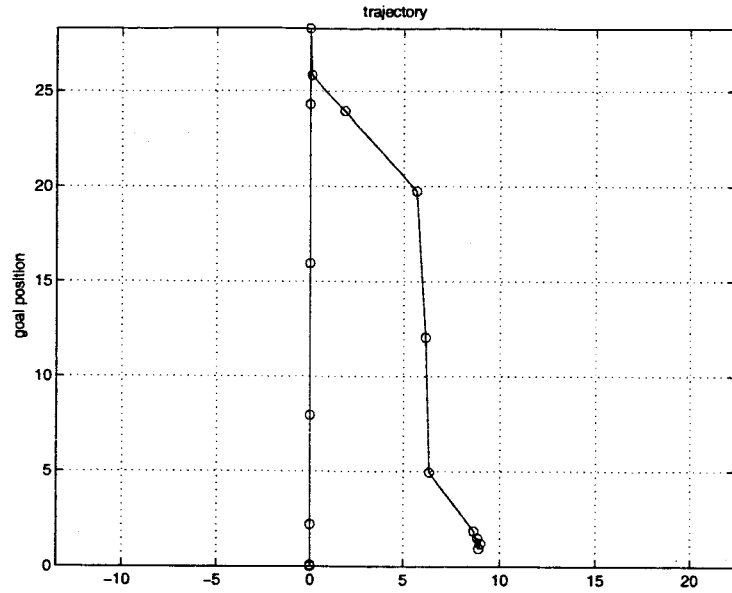


Figure 16: Vehicle trajectory for simulation 3: the helicopter starts from hover at  $(0, 0)$  and has to go to a location 30m ahead  $(30, 0)$ , and then back to a destination offset 10m from the departure point  $(0, 10)$ . A hammerhead maneuver is available to the planner.

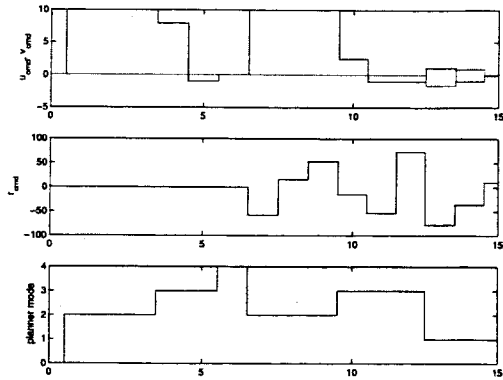


Figure 17: Commands and planner mode.

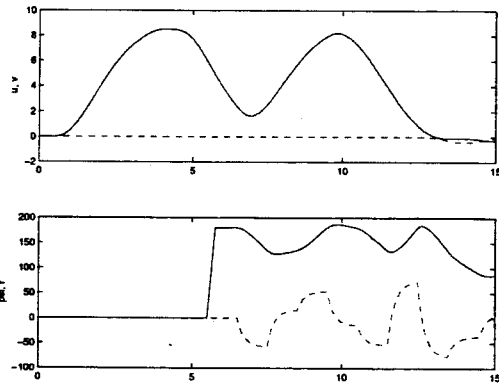


Figure 18: Helicopter states.

approach. The type of techniques we used in our system are based on exploiting the structure and specific characteristics of the agile trajectory planning problem. They work on the principle of decoupling certain aspects of the decision problem from the actual trajectory optimization, and introducing a decision hierarchy. Ongoing and future work focuses on formalizing these techniques.

## 5 Bibliography

### References

- [1] J. Albus. Outline for a Theory of Intelligence. *IEEE Transaction on Systems, Man, and Cybernetics*, 21(3):473–509, May/June 1991.
- [2] J. Albus, A. Meystel, and S. Uzzaman. “Multiscale Motion Planning”. Number 0-8186-8138-1/97, pages 19–26. IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA’97), 1997.
- [3] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, MA, 1995.
- [4] A. E. Bryson and Yu-Chi Ho. *Applied Optimal Control*. Taylor and Francis, Bristol, PA, 1975.
- [5] E. Frazzoli. “Robust Hybrid Control for Autonomous Vehicle Motion Planning”, PhD Thesis, MIT, June 2001.
- [6] 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.
- [7] E. Frazzoli, M.A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, 25(1):116–129, 2002.
- [8] V. Gavrillet, B. Mettler, and E. Feron. “Human-Inspired Control Logic for Automated Maneuvering of Miniature Helicopter”. *AIAA Journal of Guidance, Control, and Dynamics*, 2004.
- [9] V. Gavrillets, E. Frazzoli, B. Mettler, M. Piedmonte, and E. Feron. “Aggressive Maneuvering of Small Autonomous Helicopters: A Human-Centered Approach”. *International Journal of Robotics Research*, pages 795–807, October 2001.
- [10] V. Gavrillets, I. Martinos, B. Mettler, and E. Feron. “Control Logic for Automated Aerobatic Flight of Miniature Helicopter”. Monterey, CA, Aug. 2002. Proceedings of the AIAA Guidance, Navigation and Control Conference.

- [11] V. Gavrillets, B. Mettler, and E. Feron. "Control Logic for Automated Aerobatic Flight of a Miniature Helicopter". Monterrey, CA, August 2002. Proceedings of the American Institute of Aeronautics Guidance, Navigation, and Control Conference.
- [12] J.K. Howlett. Memorandum: Velocity-Heading Control Low-Order Equivalent Systems. Technical report, NASA Ames, Moffett Field, CA, June 2003.
- [13] ILOG. ILOG Optimization Suite; White Paper. Technical report, ILOG, Mountain View, CA, 2001.
- [14] B. Mettler, C. Dever, and E. Feron. "Scaling Effects and Dynamic Characteristics of Miniature Rotorcraft". *AIAA Journal of Guidance, Control, and Dynamics*, 2004.
- [15] B. Mettler, M.B. Tischler, and T. Kanade. "System Identification Modeling of a Small-Scale Unmanned Rotorcraft for Control Design". *Journal of the American Helicopter Society*, 47(1):50-63, January 2002.
- [16] 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.
- [17] M. Piedmonte and E. Feron. "Aggressive Maneuvering of Autonomous Aerial Vehicles: A Human-Centered Approach". Snowbird, UT, October 1999. International Symposium on Robotics Research.
- [18] Arthur Richards, John Bellingham, Michael Tillerson, and Jonathan P. How. "Co-ordination and Control of Multiple UAVs". Monterey, CA, Aug. 2002. Proceedings of the AIAA Guidance, Navigation and Control Conference.
- [19] 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.
- [20] 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.
- [21] T. Schouwenaars, B. Mettler, E. Feron, and J. How. "Hybrid Architecture for Receding Horizon Guidance of Agile Autonomous Rotorcraft". St. Petersburg, Russia, June 2004. 16th IFAC Symposium on Automatic Control in Aerospace.
- [22] T. Schouwenaars, B. Mettler, J. How, and E. Feron. Hybrid Architecture for Full-Envelope Rotorcraft Guidance. Phoenix, AZ, May 2003. Forum of the American Helicopter Society.

- [23] T. Schouwenaars, B. Mettler, J. How, and E. Feron. Robust Motion Planning Using a Maneuver Automaton with Built-in Uncertainties. Denver, CO, June 2003. American Control Conference.