HYBRID CONTROL FOR AGGRESSIVE MANEUVERING OF AUTONOMOUS AERIAL VEHICLES

Marc W. McConley, Michael D. Piedmonte, Brent D. Appleby, Draper Laboratory, Cambridge, MA

Emilio Frazzoli, Eric Feron, Munther A. Dahleh, Massachusetts Institute of Technology, Cambridge, MA

Abstract

New advances in control theory are required to enable aggressive maneuvering of autonomous vehicles, while adapting in real time to changes in the operational environment. A hybrid control architecture, the states of which represent feasible trajectory primitives, is constructed to reduce the complexity of the motion-planning problem for a nonlinear, high-dimensional system such as an aerial vehicle. Any feasible trajectories in the primitive list are available to the automatic control system; these may include a complete set of transitions between pairs of trim trajectories in addition to pilot-inspired behaviors recorded during manual flight tests with a human pilot. This paper describes the structure of a hybrid automaton that solves a time-optimal motion-planning problem by sequencing maneuvers in real time from such a primitive list. The algorithm can be used in a free workspace, or in the presence of fixed or moving obstacles. We present simulation results showing the effectiveness of this approach for a behavior library generated by a combination of analysis and live flight tests with a small remote-controlled helicopter.

Introduction

Operation of future autonomous vehicles in high-stress mission environments, such as air combat, suppression of enemy air defenses, and urban warfare, requires high maneuverability and adaptation to uncertain dynamics and environmental conditions. Traditional control algorithms impose performance limitations that fall far short of what can be achieved by skilled human pilots (Figure 1). New advances in control theory are required to overcome these limitations in order to enable aggressive autonomous vehicle maneuvering while adapting in real time to changes in the operational environment.

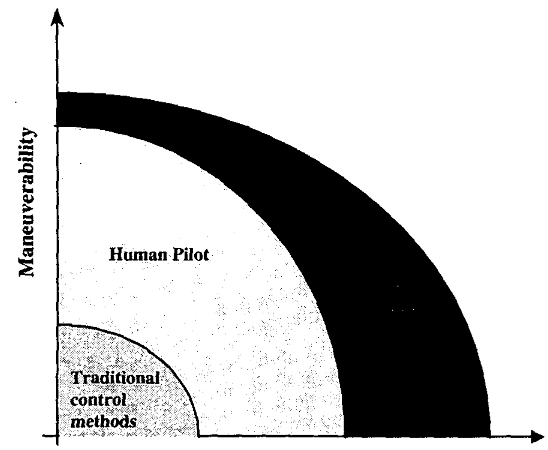

Figure 1. Human pilots far outmaneuver traditional automatic control systems, yet lack the bandwidth to achieve the full vehicle maneuvering capability.

Motion planning for a nonlinear high-dimensional system such as an aerial vehicle is a highly complex problem. Our approach to reducing this complexity is based on quantization of the system dynamics, leading to a hybrid control architecture. The states of the hybrid automaton represent feasible trajectory primitives for the vehicle. Given a set of feasible trajectory primitives, the control problem becomes one of "stitching" trajectories together with pre-defined maneuvers that start and end at trim trajectories.

The power of this approach is that any sort of feasible trajectories can be incorporated in the primitive list and made available to the automatic control system. In particular, it is possible to include pilot-inspired behaviors by conducting manual flight tests with the vehicle and recording control and state trajectory information while a human pilot executes a desired set of aggressive maneuvers. The behavior library thus constructed can be augmented by generating additional maneuvers to transition among a prescribed set of trim trajectories via analysis of the dynamic model of the vehicle (Figure 2).

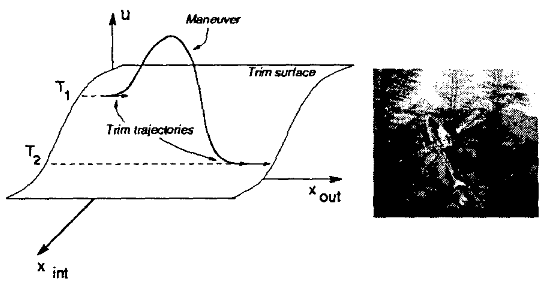

Figure 2. Maneuvers between trim trajectories of the vehicle that exercise the full vehicle maneuver envelope are generated by analysis of the vehicle dynamics.

This paper describes the structure of a hybrid automaton that solves a time-optimal motion-planning problem by sequencing maneuvers in real time from the combined list of pilot-inspired behaviors and model-generated maneuvers (Figure 3). The motion-planning algorithm can be used in a free workspace, or in the presence of fixed or moving obstacles. We present a case study in which we have generated a behavior library via a combination of analysis and live flight tests with a small remote-controlled helicopter. Given this behavior library, we have constructed a hybrid automaton for motion planning and a nonlinear control law for maneuver execution. We present simulation results showing the effectiveness of this approach.

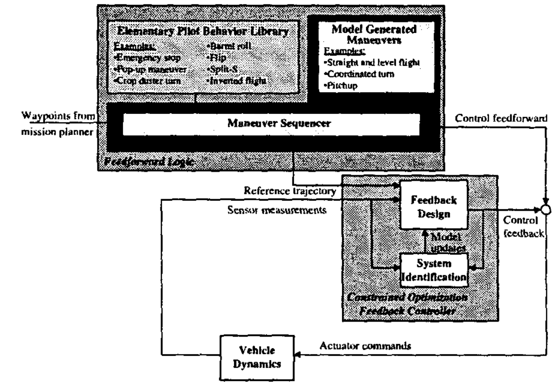

Figure 3. A hybrid automaton sequences maneuvers in real time from a list of pilot-inspired behaviors and model-generated maneuvers. In practice, this automaton is combined with an inner-loop feedback controller in a larger control system architecture.

Hybrid Automaton

Our approach to reduce the computational complexity of the motion-planning problem for a nonlinear, high dimensional system is based on a quantization of the system dynamics. Specifically, we restrict the feasible nominal system trajectories to a family of time-parameterized curves that can be obtained by the interconnection of appropriately defined primitives. These primitives then constitute a "maneuver library" from which the nominal trajectory is constructed. Instead of solving an optimal control problem over a high-dimensional, continuous space, we solve a mixed-integer-programming problem over a much smaller space.

At the core of the control architecture lies a hybrid automaton, the states of which represent feasible trajectory primitives for the vehicle. The task of the automaton is the generation of complete, feasible and "optimal" trajectories via the interconnection of the available primitives. In addition to reducing computational complexity, this approach provides a mathematical foundation for generating a provably stable hierarchical system and for developing the tools to analyze robustness in the presence of uncertainty in the process as well as in the environment [1]. Rigorous mathematical definitions of all these concepts can be found in [2].

We characterize trajectory primitives in order to capture the relevant characteristics of the vehicle dynamics, to allow for the creation of complex behaviors from the interconnection of primitives (to obtain "good" approximations to optimal solutions), and to determine the *minimal* set of key parameters identifying the state of the system. This is even more important for extension to multi-vehicle operations or for more complex systems.

Equilibrium points and trim trajectories

Equilibrium points trivially represent the simplest possible motion primitive. In a system with multiple equilibrium points, each equilibrium point can be chosen as a trajectory primitive. A closely related and more interesting class of primitives is given by trim trajectories. In an autonomous vehicle setting, these can be seen as those trajectories along which the velocities in body axes and the control input are constant. For aerial vehicles, such trim trajectories are usually described by steady-state values of the velocity magnitude (V), the flight-path angle (γ), the turn rate (r), and the sideslip angle (β) [3]. For many kinds of vehicles (such as fixed-wing aircraft), a zero sideslip angle is desired, but this is not always true for vehicles like helicopters that can move sideways and backward.

The first step in the design of our control architecture is the selection of a number of trim trajectories. The selection of trim trajectories can be carried out by gridding the set of attainable values of the steady-state parameters. This set is compact and can be identified with the flight envelope in the case of aerial vehicles.

This class of trajectory primitives has been used widely to construct control systems that switch through a sequence of controllers, progressively taking the system closer to a desired equilibrium point [4-7]. The ideas of gain scheduling and of linear parameter varying (LPV) system control can also be brought into this class [8], as well as other integrated guidance and control systems for UAV applications [9]. However, such a design choice generally results in relatively poor performance, and in "slow" transitions, as the system is required to stay in some sense close to the trim surface. Moreover, the absence of any information on the transient behavior can lead to undesirable effects, such as limit cycles.

Maneuvers

For more aggressive maneuvering, it is necessary to characterize trajectories that move "far" from the trim surface. In this paper, a *maneuver* is defined as a finite-time transition between any two trim trajectories, as shown in Figure 2. This may include transitions between two instances of the *same* trajectory. For example, acrobatic maneuvers like loops and barrel rolls can be considered as transitions out of and back into straight and level flight.

A problem in the off-line generation of trajectories is the large amount of storage memory required. Compressing the trajectory data can solve this problem. In this case, we identify some relevant parameters on the basis of which the onboard processor can compute "easily" (in real time) a reference trajectory to track. An efficient representation of trajectories can be achieved by exploiting the properties of differentially flat systems [10,11].

The design of the nominal trajectories must be carried out in a way that ensures that the vehicle does not violate constraints on its dynamic envelope (for example, maximum velocity) [12]. In this sense, the objective of envelope protection is ensured implicitly by the maneuver definition.

Motion planning

Given a defined set of trim trajectories and maneuvers specified off line, the on-line control problem becomes one of selecting, while in a given trim trajectory, what the next maneuver should be and when it should occur. All the relevant information while in a trim trajectory is defined by the hybrid automaton state and the vehicle position and heading at the inception of the trim trajectory. On each trim trajectory, the discrete control set can be identified with the subset of all the maneuvers that start at that trim state. Moreover, the hybrid controller must decide the timing of the jump.

Let q represent the trim state in the automaton and let h be a vector containing the position and heading of the vehicle. Assume we want to control the system to a desired trim state (q^*), and position and heading (h^*), and define a running cost function $\gamma(q, h)$ with $\gamma(q^*, h^*) = 0$. Given a control policy μ we can define a total cost function

$$J_\mu(q_0, h_0) = \int_0^\infty \gamma(q, h) dt.$$

We are interested in computing the optimal policy μ^* that minimizes the total cost for all initial conditions. Following dynamic programming theory, we can compute the optimal policy by solving Bellman's equation [13]. The optimization requires the solution of a mixed-integer program with one continuous variable (the jump time) and one discrete variable (the final state of the automaton at the end of the selected maneuver). In general, the optimal cost function is not known, but approximate dynamic programming algorithms such as value or policy iteration can be used. Moreover, since the dimension of the state space has been reduced to one discrete variable and four continuous variables, neuro-dynamic programming approximation techniques for a compact representation of the cost function can be effectively used, making the control algorithms suitable for real-time applications [14].

Obstacle avoidance

An important problem in autonomous vehicle control is motion planning in the presence of fixed or moving obstacles. We use the expression "motion planning" as opposed to the traditional "path planning" to emphasize the role of dynamics or non-holonomicity constraints on the allowable feasible trajectories. This problem is also known in the literature as kinodynamic planning [15].

The state space of non-trivial systems is typically very large, and the "curse of dimensionality" makes the solution of motion planning problems in such large-dimension spaces computationally intractable. The alternative that we propose, through the introduction of our hybrid control architecture, can be seen as a maneuver automaton space, in which we discretize the system dynamics, not the state space.

Using this representation, the hybrid automaton encodes all the relevant information about the system dynamics and dynamic constraints. The dynamic constraints must then be complemented by configuration constraints. We can define a set of primitives and starting configurations, such that the resulting trajectory is collision free. In general, the computation of such a set is very challenging. In the hybrid system literature, an approach that has been applied to several problems is based on the definition and the computation of the level sets of an appropriately defined Hamiltonian function [12,16]. A possibly conservative computation of the safe set can be carried out quite easily in several cases. For example, in the case of fixed obstacles, every trajectory that allows for a complete stop along a collision-free trajectory is safe.

Since the set of primitives is considerably smaller than the complete state, solution of optimal control problems for the hybrid automaton is computationally less expensive than solutions on the full state space. The hybrid automaton model lends itself in a straightforward manner to the implementation of a new class of motion-planning algorithms based on a randomized approach [17-19]. In particular, we use a form of rapidly exploring random trees (RRT's), introduced in [20,21].

A concise statement of our version of the RRT algorithm is depicted in Figure 4. We start by building a tree and assigning the initial state as its root. As the next step, we compute the optimal path in the obstacle free case to generate a sequence of states approaching the target. If no collisions are found on the tree, the algorithm terminates. Otherwise, we add all the states up to the one just before the collision to the tree. Choosing (at random) a new target state, we compute the corresponding optimal obstacle-free path that starts from the "closest" state in the tree. In this case, we can use as a distance function the unconstrained optimal cost function. For the new path, we add the states up to the target or the first collision to the tree as a child of the last state in the previous path. We repeat this process recursively until a path to the target is obtained, or until the algorithm terminates with a failure upon exceeding a prescribed maximum iteration count.

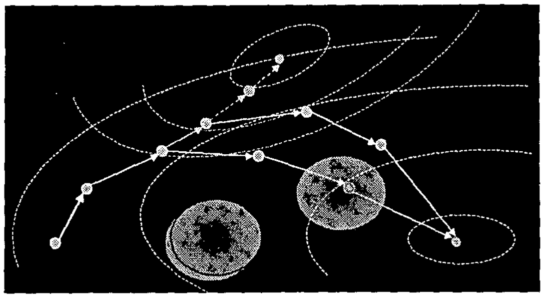

Figure 4. Motion planning approach with RRTs.

The RRT algorithm is not complete, in the sense that it could fail to generate a feasible trajectory, even if one exists. However, it can be shown that the RRT algorithm presented above is complete in a probabilistic sense for fixed obstacles [22]. A modified algorithm that is provably complete for moving obstacles appears in [23]. Randomized algorithms have been found to perform well in many applications of interest.

A variation of the approach just described has been implemented to allow for on-line selection of the minimum-time feasible trajectory. The modification consists in back-propagating the time to target each time a feasible child trajectory is added to the tree. By back-propagating we mean that we have to climb the tree back towards the root, labeling each node as the time to target of the child tree plus the time required for the transition from the node to the root of the child tree. This climbing process is repeated until the label on the node under examination in the current trajectory is found to have higher cost than a previous one or until we get to the maneuver tree root, in which case the current trajectory represents the best choice.

Randomized motion planning has been tested in several examples, including cases with moving obstacles, and proved to be fast and reliable. In the two cases depicted in Figures 5 and 6, the randomized motion planner succeeded in finding 50-100 feasible trajectories, for a running time ranging between 2 and 6 seconds. Figure 5 shows the optimal trajectory to avoid several fixed cylindrical obstacles, and Figure 6 shows an example of traversing a maze.

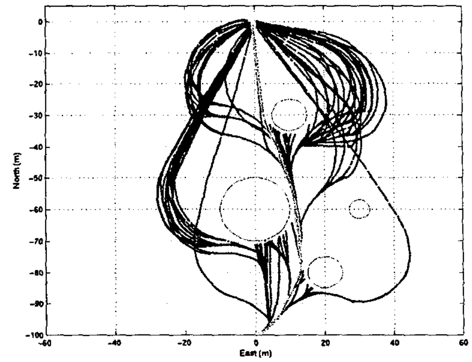

Figure 5. Cylindrical obstacle avoidance.

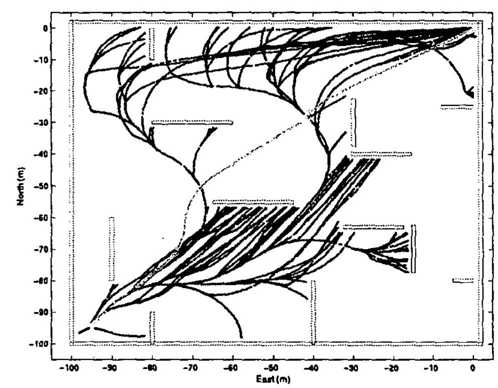

Figure 6. Maze traversal.

Pilot-inspired Maneuvers

The maneuver list used by the hybrid automaton can be augmented with more aggressive pilot-inspired maneuvers in addition to the analysis-based transitions between trim trajectories. Any feasible maneuver that the vehicle can perform can be included in the maneuver list. This provides a mechanism whereby live flight test data, recorded while a human pilot executes the desired aggressive maneuvers with the real vehicle, can be incorporated directly into the automatic control system.

Our approach to implementing aggressive maneuvers in an automatic system is a discrete-event rule-based approach [24]. Given recorded pilot inputs and measurements of vehicle states, some simple transition rules can be devised and tested in simulation to determine whether they reproduce the desired behaviors.

We shall demonstrate this approach using the example of a barrel roll executed on a small remote-controlled helicopter. Figure 7 shows a section of recorded flight data corresponding to a barrel roll maneuver executed by a human pilot during a live flight test. From analysis of the flight data and conversations with the pilot [25], it is clear that the pilot follows a few simple rules. The roll cyclic input (used to provide roll acceleration) is pushed to a fixed location and held there throughout the maneuver. In addition, the main rotor collective is used to reverse the direction of lift during the portion of the maneuver in which the vehicle is inverted.

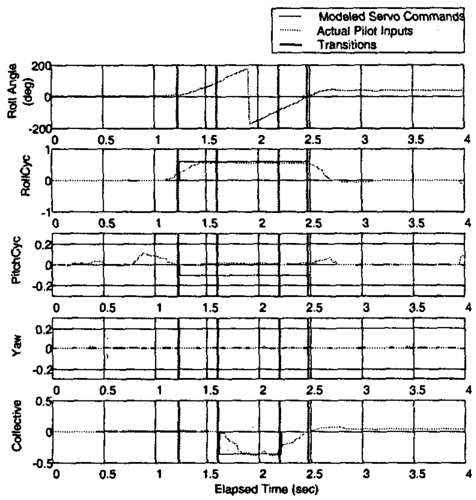

Figure 7. Recorded flight data for a barrel roll.

These rules can be encapsulated in a state flow diagram (Figure 8). At the beginning of the maneuver, the roll cyclic input is set to a constant value that is extracted from the recorded pilot inputs. This state is held until the vehicle is at a ninety-degree roll angle. At this point, the direction of lift must be reversed in order to maintain the proper lift through inverted flight, so the main rotor collective is set to a constant value extracted from the flight data. During this intermediate state, we continue to hold the roll cyclic input from the starting state. When the roll angle reaches ninety degrees in the opposite direction, the collective is restored to its original value (the original lift direction is resumed), and we continue to hold roll cyclic. Finally, we transition back to the nominal closed-loop control system at some small roll angle (for example, 5 degrees). The roll angle for the final transition is selected to ensure that the initial condition at the transition is within the region of stability of the closed-loop control system.

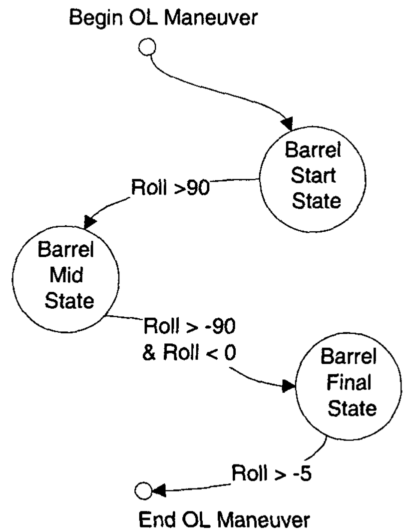

Figure 8. State flow diagram for a barrel roll.

Figures 9 and 10 show recorded flight data and the corresponding control inputs for two other pilot-inspired maneuvers: a flip and a "split-S" maneuver. All three of these maneuvers have been successfully executed in simulation using a discrete-event rule-based approach.

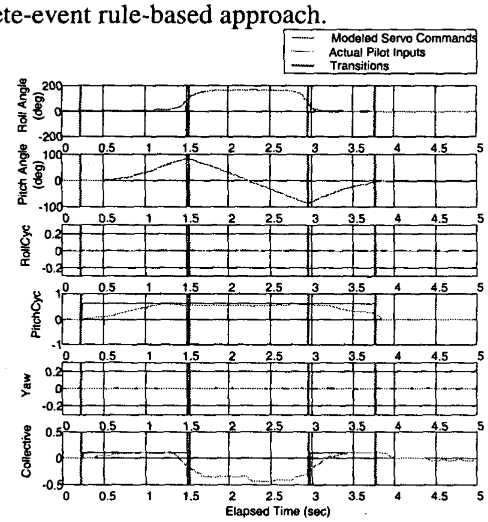

Figure 9. Recorded flight data for a flip.

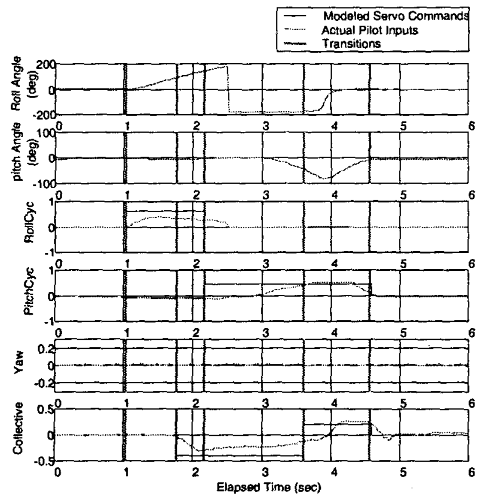

Figure 10. Recorded flight data for a split-S.

Figure 11 shows an example of how pilot-inspired maneuvers would be incorporated into a typical waypoint guidance strategy. The hybrid automaton is used to determine which pilot-inspired maneuvers achieve the desired optimization result and to schedule when they occur. (In the figure, "H" represents a hovering waypoint; "T" means to fly through the waypoint.)

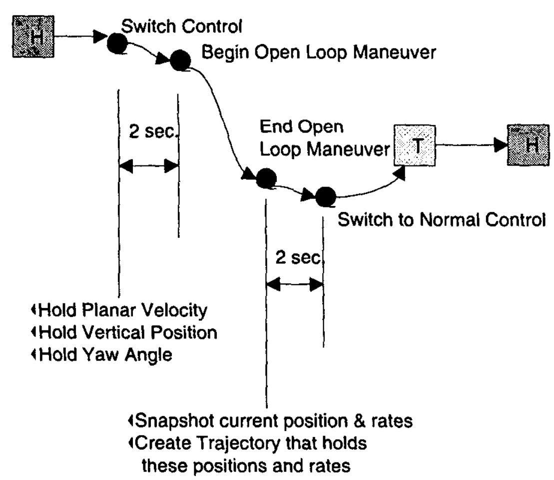

Figure 11. Example of waypoint guidance with pilot-inspired maneuvers.

Conclusions

This paper presented a hybrid automaton architecture applicable to autonomous vehicles. Algorithms have been given to solve (perhaps approximately) the time-optimal motion-planning problem in a free workspace as well as in the presence of fixed and moving obstacles. The hybrid automaton structure has been found to provide a flexible, computationally effective tool for motion planning for autonomous vehicles. Simulation results were presented for a specific example involving a small autonomous helicopter.

Current work includes the extension of the hybrid automaton to multi-vehicle operations and to real-time motion planning in an uncertain environment. We are also in the process of refining both the automaton and the pilot-inspired maneuver execution to be sufficiently numerically robust for safe flight operation. We plan to conduct autonomous flight tests in the coming year to evaluate this approach in real flight.

Acknowledgment

The authors are grateful to Alex Lob for piloting the remote-controlled helicopter and executing aggressive maneuvers during live flight tests.

[1] Frazzoli, E., M. A. Dahleh, E. Feron, 1999, *Robust Hybrid Control for Autonomous Vehicle Motion Planning*, Technical Report LIDS-P-2468, Massachusetts Institute of Technology.

[2] Frazzoli, E., M. A. Dahleh, E. Feron, 1999, "A Hybrid Control Architecture for Aggressive Maneuvering of Autonomous Helicopters," *IEEE Conf. on Decision and Control.*

[3] Stevens, B. L., F. L. Lewis, 1992, *Aircraft Control and Simulation*, John Wiley & Sons.

[4] Leonessa, A., V. S. Chellaboina, W. M. Haddad, 1997, "Globally Stabilizing Controllers for Multi-mode Axial Flow Compressors via Equilibria-dependent Lyapunov Functions," *Proc. IEEE Conf. Dec. Contr.*, San Diego.

[5] Tittus, M., B. Egardt, 1998, "Control Design for Integrator Hybrid Systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 491-500.

[6] McConley, M. W., B. D. Appleby, M. A. Dahleh, E. Feron, 2000, "A Computationally Efficient Lyapunov-Based Scheduling Procedure for Control of Nonlinear Systems with Stability

Guarantees," *IEEE Transactions on Automatic Control*, vol. 45, no. 1, pp. 33-49.

[7] Burridge, R. R., A. A. Rizzi, D. E. Koditscheck, 1999, "Sequential Decomposition of Dynamically Dexterous Robot Behaviors," *International Journal of Robotics Research*, vol. 18, no. 6, pp. 534-555.

[8] Shamma, J. S., J. R. Cloutier, 1993, "Gain-scheduled Missile Autopilot Design Using Linear Parameter Varying Transformations," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 16, no. 2, pp. 256-263.

[9] Kaminer, I., A. Pascoal, E. Hallberg, C. Silvestre, 1998, "Trajectory Tracking for Autonomous Vehicles: An Integrated Approach to Guidance and Control," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 29-38.

[10] Van Nieuwstadt, M. J., R. M. Murray, 1998, "Real-time Trajectory Generation for Differentially Flat Systems," *International Journal of Robust and Nonlinear Control*, vol. 8, no. 11, pp. 995-1020.

[11] Koo, T. J., S. Sastry, 1998, "Output Tracking Control Design of a Helicopter Model Based on Approximate Linearization," *IEEE Conference on Decision and Control.*

[12] Tomlin, C., J. Lygeros, S. Sastry, 1998, "Aerodynamic Envelope Protection Using Hybrid Control," *Proc. American Control Conf.*, vol. 3, pp. 1793-1796.

[13] Bertsekas, D. P., 1995, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, MA.

[14] Bertsekas, D. P., J. T. Tsitsiklis, 1996, *Neuro-dynamic Programming*, Athena Scientific, Belmont, MA.

[15] LaValle, S. M., 2000, "Robot Motion Planning: A Game-theoretic Foundation," *Algorithmica*, to appear.

[16] Tomlin, C., G. J. Pappas, S. Sastry, 1998, "Conflict Resolution for Air Traffic Management: A Study in Multi-agent Hybrid Systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4.

[17] Kavraki, L. E., P. Svestka, J. C. Latombe, 1996, "Probabilistic Roadmaps for Path Planning in High-dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566-580.

[18] Kavraki, L. E., M. N. Kolountzakis, J. C. Latombe, 1996, "Analysis of Probabilistic Roadmaps for Path Planning," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp. 3020-3025.

[19] Kavraki, L. E., J. C. Latombe, 1998, "Probabilistic Roadmaps for Robot Path Planning," K. Gupta, A. del Pobil, ed., *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, John Wiley, pp. 33-53.

[20] LaValle, S. M., J. J. Kuffner, 1999, "Randomized Kinodynamic Planning," *Proceedings of the 1999 IEEE International Conference on Robotics and Automation.*

[21] LaValle, S. M., 1999, "Rapidly-exploring Random Trees: A New Tool for Path Planning," available on the Internet at http://janowiec.cs.iastate.edu/~lavalle/papers/rrt.ps.

[22] Kuffner, J. J., S. M. LaValle, 2000, "RRT-Connect: An Efficient Approach to Single-query Path Planning," *IEEE International Conference on Robotics and Automation.*

[23] Frazzoli, E., M. A. Dahleh, E. Feron, 2000, "Real-time Motion Planning for Agile Autonomous Vehicles," *AIAA Guidance, Navigation, and Control Conference*, AIAA-2000-4056.

[24] Piedmonte, M., E. Feron, 1999, "Aggressive Maneuvering of Autonomous Aerial Vehicles: A Human-Centered Approach," *International Symposium on Robotics Research*, Snowbird, UT.

[25] Lob, Alex, 1999, Personal communication.