

Robust Motion Planning Using a Maneuver Automaton with Built-in Uncertainties

Tom Schouwenaars,¹ Bernard Mettler,² Eric Feron,³ and Jonathan P. How⁴

Massachusetts Institute of Technology

Cambridge, MA 02139

Keywords: Autonomous Vehicles, Guidance, Motion Planning, Robust Optimization

Abstract

In this paper, we extend a recently introduced motion planning framework for autonomous vehicles based on a maneuver automaton representation of the vehicle dynamics. We bring robustness into the guidance system by accounting for the uncertainties in the motion primitives used by the maneuver automaton. The uncertainties are taken into account in the offline computation of a guidance function, as well as in a real-time planning policy. We illustrate our approach using a high-fidelity simulation model of MIT’s autonomous X-Cell miniature helicopter, and present an example that highlights the performance improvement over the original framework. We demonstrate that, when uncertainties are present, a nominal planning policy generates suboptimal trajectories in both open- and closed-loop guidance, and that trajectories obtained by applying the robust policy are less sensitive to perturbations in the motion primitives.

1 Introduction

An essential part of the autonomy of an unmanned aerial vehicle (UAV) consists of a path planning and guidance system that enables the vehicle to plan and execute a trajectory in a particular environment. This environment may contain obstacles around which the vehicle has to navigate, and whose positions may not be exactly known *a priori*. Moreover, the environment may change over time, requiring real-time motion planning and guidance algorithms.

However, it is known that the motion planning problem is intrinsically NP-hard [1], and that its complexity grows with that of the vehicle dynamics and the environment in which the vehicle has to operate. This is certainly the case for highly agile vehicles with fast and complex dynamics, such as small-scale rotorcraft [2]. For such vehicles, the state space and the set of possible control actions are extremely large, requiring simplifications to reduce the dimensionality when a solution has to be computed in real-time.

Such simplifications are usually based on approximate representations of the vehicle dynamics, such as the use of lower order equations of motion. Sometimes, only the vehicle kinematics are used for planning. These approximations, however, usually result in conservative performance. To address this shortcoming, representations based on motion primitives have recently been introduced [3]. By choosing an appropriate set of motion primitives, the state space can be significantly reduced without giving up the key performance and maneuverability of the vehicle [4].

Frazzoli *et al.* used the concept of hybrid automata [5] to model the dynamics of the vehicle based on motion primitives, and formulated the optimal motion planning problem as a dynamic program [6]. In our previous work [4], we used this so-called *maneuver automaton* framework to compute offline open-loop trajectories for MIT’s autonomous X-Cell miniature helicopter. We assumed perfect execution of the motion primitives, which allowed us to decouple the guidance system from the actual vehicle dynamics. This assumption was previously introduced in [6], and was termed the “consistency assumption”.

In reality, however, there exists uncertainty in the parameters of the motion primitives, due to both uncertainties in the model, as well as external disturbances acting on the vehicle. As a result, the consistency assumption is usually not valid in practice. As we show in this paper, these uncertainties result in perturbed and suboptimal trajectories, even when a path is computed online with a closed guidance loop. Closing the guidance loop based on a nominal policy, *i.e.*, a policy that assumes perfect execution of the motion primitives, is therefore not a sufficient condition for optimality of the

¹Research Assistant, Laboratory for Information and Decision Systems, toms@mit.edu

²Post-doctoral Associate, Laboratory for Information and Decision Systems, bmettler@mit.edu

³Associate Professor, Laboratory for Information and Decision Systems, feron@mit.edu

⁴Associate Professor, Space Systems Laboratory, jhow@mit.edu

trajectories.

To tackle this issue, we extend the maneuver automaton framework and dynamic programming formulation to explicitly account for the uncertainty in each of the motion primitives. As such, we obtain a robust version of the maneuver automaton and a corresponding robust, real-time guidance policy. Implementation of this robust policy results in trajectories that are less affected by mismodeling and disturbances, both in open- and closed-loop guidance.

2 Maneuver Automaton Representation

In this section, we give a brief outline of the *maneuver automaton* (MA) as introduced in [6]. The MA is a hybrid representation of the dynamics of a vehicle, consisting of a finite collection of two types of motion primitives: trim conditions and maneuvers. A trim condition is a dynamic equilibrium of the vehicle, a maneuver is defined as a finite time transition between two trim conditions. For a helicopter, for instance, two trim conditions could be hover mode and forward flight at $5m/s$; the corresponding maneuvers would be the two transitions between these flight regimes.

The MA is a dynamic system in its own right, with a hybrid quantity as control input. Its state is described by a hybrid vector which evolves according to precise laws. At each point in time, the vehicle is either in a trim condition or performing a maneuver between two trim conditions. The system's behavior for these two conditions is as follows:

- while in trim condition q , the vehicle can *coast* for an unlimited length of time $\tau[k]$. This time variable is the continuous component of the MA's hybrid control vector. The hybrid state vector (\mathbf{x}, q) consists of the continuous state \mathbf{x} and the discrete trim condition q . The hybrid state then evolves as:

$$\mathbf{x}[k+1] = \mathbf{x}[k] + \tau[k]\dot{\mathbf{x}}_q \quad (1)$$

$$q[k+1] = q[k] \quad (2)$$

$$t[k+1] = t[k] + \tau[k], \quad (3)$$

where $\dot{\mathbf{x}}_q$ is the time rate of change of the vehicle's continuous state variables.

- while performing a maneuver p , the vehicle leaves the trim trajectory $q_{\text{from},p}$ for a finite length of time before settling to the trim trajectory $q_{\text{to},p}$. The maneuver is initiated by the control action p , which is the discrete component of the hybrid control vector, and is described by a fixed duration T_p and displacement $\Delta\mathbf{x}_p$ in the continuous state space. The actual time history of the continuous

state and control variables during a maneuver is a function of the control law used in the execution of the maneuver. As such, when maneuvering, the hybrid state evolves as:

$$\mathbf{x}[k+1] = \mathbf{x}[k] + \Delta\mathbf{x}_p \quad (4)$$

$$q[k+1] = q_{\text{to},p} \quad (5)$$

$$t[k+1] = t[k] + T_p, \quad (6)$$

The hybrid control input at instant k can thus be described by a vector $(\tau, p)_k$. A coasting command is then represented as $(\tau, 0)_k$, a maneuver command as $(0, p)_k$.

By using the MA representation, the problem of finding a time-optimal trajectory from an initial position \mathbf{x}_0 and trim state q_0 , to a destination \mathbf{x}_f and trim state q_f , amounts to finding the sequence of coasting times and maneuvers $(\tau, p)_k$, $k = 1 \dots N$, that minimizes the total traveling time $T = \sum_{k=1}^N (\tau[k] + T_p[k])$. This optimization problem can be formulated as a dynamic program (DP):

$$\mathbf{J}(\mathbf{x}[k], q[k]) = \min_{\tau, p} \left[\tau[k] + T_p[k] + \mathbf{J}(\mathbf{x}[k+1], q[k+1]) \right], \quad (7)$$

with $(\mathbf{x}[0], q[0]) = (\mathbf{x}_0, q_0)$ and $\mathbf{J}(\mathbf{x}_f, q_f) = 0$.

The above expression is a form of the Bellman equation, and can be solved by running a value iteration offline [7]. The result of this computation is an optimal cost function $\mathbf{J}(\mathbf{x}, q)$, that gives the minimal time-to-go from all hybrid states (\mathbf{x}, q) on a discretization grid to the destination and end trim (\mathbf{x}_f, q_f) .

Note that this time-to-go function is computed only once. It is stored in the on-board computer of the vehicle, and can subsequently be used for real-time motion planning towards waypoints. The online planning phase then merely consists of evaluating all possible control actions from the current state $(\mathbf{x}[k], q[k])$: the vehicle can either keep coasting in trim $q[k]$, or can execute a maneuver with $q_{\text{from}} = q[k]$. The optimal action to be executed is the one that minimizes the right-hand side of Eq. (7), *i.e.*, the action that minimizes the sum of the coasting or maneuver time and the resulting time-to-go $\mathbf{J}(\mathbf{x}[k+1], q[k+1])$ from the new state $(\mathbf{x}[k+1], q[k+1])$. Since the number of possible actions is limited, and since the time-to-go can now be looked up or interpolated in the database, this optimal guidance policy can be implemented in real-time.

3 Robust Motion Planning

Although the MA technique as described above is suited for closed-loop guidance, previous work only considered offline (*i.e.*, open-loop) planning [4]. In open-loop planning, the entire sequence of control actions

that will guide the vehicle to the goal is generated beforehand, based on the characteristics of the motion primitives. As a result, because of disturbances in the environment and uncertainties in the MA model – *i.e.*, uncertainties in the trim and maneuver parameters –, one can expect the open-loop trajectory to miss the goal when the plan is executed. A closed-loop trajectory, on the other hand, is more likely to reach the destination.

However, if the uncertainties in the parameters of the motion primitives are significant, both open- and closed-loop paths will be affected. This is primarily caused by the fact that maneuvers are characterized by a fixed duration and a fixed displacement in the continuous state space. A perturbation in these values will guide the vehicle away from the optimal state “foreseen” by the automaton.

Although a closed-loop policy will correct for these perturbations, it will only do so after the maneuver has been executed. This is a consequence of the hybrid automaton setup: whenever a maneuver is initiated, it has to be executed until the end. Even though the guidance policy will optimize the rest of the trajectory from that perturbed state on, the total resulting trajectory from start to end will be suboptimal. Moreover, when the uncertainties in the motion primitives are of the order of magnitude of the grid resolution used in the offline computation of the value function, the trajectory can take the appearance of a random walk. Hence, if the target region around the goal is too small, the vehicle might keep “bouncing” around it, before converging to the goal.

To mitigate these effects, one needs to account for the uncertainties in the MA when formulating and implementing the dynamic program. The optimal action to be executed should be a trade-off between the cost of the action and the uncertainty in the cost, resulting from perturbations in the execution of the action. The cost uncertainty has two components: 1) the uncertainty in the duration of the action itself, and 2) the uncertainty in the time-to-go resulting from the uncertainty in the end state of the action. However, the two uncertainty components are not independent: they are coupled by the vehicle dynamics.

As in classic DP, the trade-off between cost and uncertainty can be formulated implicitly by taking the expected value over the uncertainties in the Bellman equation. In particular, we introduce the variables $\varepsilon_{\dot{\mathbf{x}}_q}$, $\varepsilon_{\Delta \mathbf{x}_p}$ and ε_{T_p} , which, respectively, represent the uncertainty in the trim parameters, maneuver displacement, and maneuver duration. Note that $\varepsilon_{\dot{\mathbf{x}}_q}$ and $\varepsilon_{\Delta \mathbf{x}_p}$ are vectors whose variables denote the uncertainty of each component of $\dot{\mathbf{x}}_q$ and $\Delta \mathbf{x}_p$ respectively. Moreover, consider an environmental disturbance vector \mathbf{w} that is acting on the vehicle. The hybrid equations governing

the MA then become:

$$\mathbf{x}[k+1] = \mathbf{x}[k] + \tau[k](\dot{\mathbf{x}}_q + \varepsilon_{\dot{\mathbf{x}}_q}) + \mathbf{w}[k] \quad (8)$$

$$q[k+1] = q[k] \quad (9)$$

$$t[k+1] = t[k] + \tau[k] \quad (10)$$

for a coasting action, and

$$\mathbf{x}[k+1] = \mathbf{x}[k] + \Delta \mathbf{x}_p + \varepsilon_{\Delta \mathbf{x}_p} + \mathbf{w}[k] \quad (11)$$

$$q[k+1] = q_{to,p} \quad (12)$$

$$t[k+1] = t[k] + T_p + \varepsilon_{T_p} \quad (13)$$

for a maneuver action.

Given an appropriate probability distribution of the different uncertainty components, a robust cost function $\mathbf{J}_{rob}(\mathbf{x}, q)$ and a corresponding optimal, robust guidance policy can then be computed by taking the expected value over the uncertainties in the offline value iteration. Accordingly, Eq. (7) should be generalized as follows:

$$\begin{aligned} \mathbf{J}_{rob}(\mathbf{x}[k], q[k]) &= \min_{\tau,p} E_{\varepsilon,w} \left[\tau[k] + T_p[k] + \varepsilon_{T_p} \right. \\ &\quad \left. + \mathbf{J}_{rob}(\mathbf{x}[k+1], q[k+1]) \right] \quad (14) \end{aligned}$$

in which $\mathbf{x}[k+1]$ is given by Eq. (8) or Eq. (11). Being an optimal trade-off between performance and uncertainty, the robust guidance policy is expected to produce “smoother” trajectories in closed-loop guidance, and to generate offline planned paths that are most likely to be perturbed the least when executed in open-loop.

4 Miniature Helicopter

We applied the robust framework described above to MIT’s X-Cell miniature helicopter [8]. The helicopter is equipped with a velocity control system that has the following command inputs: body axis forward velocity u_{cmd} , body axis side velocity v_{cmd} , altitude h_{cmd} and yaw rate r_{cmd} . The yaw rate command is mechanized to work as a coordinated turn rate command in forward flight. Since the helicopter also features an altitude hold mode, we could concentrate on the horizontal planning problem. Hence, the continuous state consisted of the distance to the goal and the relative heading to the goal, as previously introduced in [4]. Moreover, for simplicity, we considered a reduced library of 20 trims and 48 maneuvers that – although not covering the full flight envelope of the vehicle – were sufficient to illustrate the present research.

For our experiments, we used a high-fidelity simulation model of the helicopter. We closed the guidance loop by continuously applying the robust guidance policy in real-time on the estimated state of the helicopter,

Table 1: Parameters for sample trim conditions q

q	u_{cmd} m/s	v_{cmd} m/s	r_{cmd} rad/s	\bar{u} m/s	\bar{v} m/s	\bar{r} rad/s	ε_u m/s	ε_v m/s	ε_r rad/s	Description
1	0.0	0.0	0.0	0.0	-0.1	0.0	0.1	0.02	0.0	hover
3	1.0	0.0	0.0	1.0	-0.1	0.0	0.45	0.01	0.0	forward at 1 m/s
4	3.0	0.0	0.0	3.0	-0.1	0.0	0.15	0.01	0.0	forward at 3 m/s
6	0.0	1.0	0.0	0.06	0.91	0.0	0.07	0.01	0.0	right sideways at 1 m/s
8	0.0	-1.0	0.0	0.0	-1.1	0.0	0.1	0.02	0.0	left sideways at 1m/s
9	0.0	0.0	0.2	0.0	0.0	0.2	0.1	0.1	0.1	right turn on the spot
13	1.0	0.0	0.2	1.1	-0.2	0.2	0.1	0.1	0.1	right turn at 1 m/s
20	3.0	0.0	-1.2	3.50	0.50	-1.0	0.5	0.2	0.1	left turn at 3 m/s

and by commanding the corresponding u_{cmd} , v_{cmd} and r_{cmd} commands to the autopilot at (100Hz). As far as our knowledge goes, this was the first attempt to build a closed-loop guidance system based on a maneuver automaton as introduced in [6].

To characterize the uncertainty in the parameters describing the motion primitives, extra variables need to be introduced in the trim and maneuver library. For instance, if the uncertainty in a parameter is normally distributed around its average value, its variance could be included in the library. However, not having a precise mathematical model of the uncertainty in the motion primitives, we took a more practical approach of simulating many different sequences of trims and maneuvers, and recorded the average value and maximum deviation of the characterizing parameters.

Trim library

Each trim in the library is denoted by an index q , and is characterized by the three step input commands u_{cmd} , v_{cmd} and r_{cmd} . We considered both rectilinear trim conditions in the four principal directions – *i.e.*, forward flight, backward flight and right and left sideways flight–, as well as turns on the spot and steady turn trims at different speeds and turn rates. More specifically, for the turn trims, we considered both a fast and a slow turn rate.

To account for the uncertainty, we included the following parameters in the trim library: \bar{u} , \bar{v} , \bar{r} , ε_u , ε_v and ε_r . The first three parameters denote the average values, the last three indicate the maximal deviation from the average. Table 1 shows the parameters of a few sample trims used in our automaton.

Maneuver library

After selecting the trims, we synthesized 48 maneuvers using the high-fidelity simulation of the closed-loop control system. We considered two classes of maneuvers: 1) transitions between the several rectilinear trims, and 2) banking maneuvers between forward/hover flight and turn trims and vice versa. For all maneuvers, we first steered the helicopter to the

starting trim condition, and then commanded a reference step input to the desired end trim trajectory. We recorded the change in longitudinal position (Δx), lateral position (Δy) and heading ($\Delta\psi$) after the completion of the maneuver. All changes were defined with respect to the position and orientation of the body frame at the initialization of the maneuver. The execution time of the maneuver (T_p) was defined as the time after which the helicopter reached steady state in the desired trim trajectory.

A few sample maneuvers are given in Table 2. Apart from the maneuver index p and the trim indices q_{from} and q_{to} , we included the following parameters in the library: \bar{T}_p , $\bar{\Delta x}$, $\bar{\Delta y}$, $\bar{\Delta\psi}$, ε_{T_p} , $\varepsilon_{\Delta x}$, $\varepsilon_{\Delta y}$ and $\varepsilon_{\Delta\psi}$, again, indicating average values and maximal deviations respectively. Given these uncertainty parameters, we computed a robust cost function by approximating the expected value in Eq. (14) by a weighted average over the nominal and perturbed motion primitives:

$$\begin{aligned} \tilde{J}_{rob}(\mathbf{x}[k], q[k]) = \min_{\tau, p} \sum_{i=-1}^1 \sum_{j=-1}^1 \alpha_{ij} \left[\tau[k] + T_p[k] \right. \\ \left. + j\varepsilon_{T_p} + \tilde{J}_{rob}(\mathbf{x}[k+1](i\varepsilon_q, j\varepsilon_p), q[k+1]) \right] \end{aligned} \quad (15)$$

The indices i for the trims and j for the maneuvers indicate the nominal case ($i, j = 0$), the maximum positive perturbations ($i, j = 1$) and the maximum negative perturbations ($i, j = -1$) respectively. The α_{ij} are the weight factors, and $\varepsilon_q = (\varepsilon_u, \varepsilon_v, \varepsilon_r)$ and $\varepsilon_p = (\varepsilon_{\Delta x}, \varepsilon_{\Delta y}, \varepsilon_{\Delta\psi})$ are the uncertainty vectors as given in the trim and maneuver table respectively. The same expression was used in the real-time guidance policy to compute the optimal action at each time step. Note that we did not include the disturbances \mathbf{w} that are inherent to the environment. These disturbances, such as wind, are typically hard to model, and since they can change drastically over time, it makes sense to disregard them in the value iteration. When the disturbances can be estimated online, however, they could be incorporated in the real-time guidance policy.

Although the weighted average formulation in Eq. (15) is a rather crude approximation of the expected value

Table 2: Parameters for sample maneuvers p

p	q_{from}	q_{to}	\overline{T}_p sec	$\overline{\Delta x}$ m	$\overline{\Delta y}$ m	$\overline{\Delta \psi}$ deg	ε_{T_p} sec	$\varepsilon_{\Delta x}$ m	$\varepsilon_{\Delta y}$ m	$\varepsilon_{\Delta \psi}$ deg	Description
3	1	4	3.20	3.78	-0.30	-3.42	0.64	1.88	-0.11	-0.28	hover to forward
15	3	1	2.40	1.24	-0.23	-0.82	0.48	0.01	-0.05	-0.01	forward to hover
17	3	4	2.64	4.77	0.27	2.34	0.53	1.57	0.08	0.41	forward acceleration
18	3	6	1.50	1.14	0.50	-4.50	0.30	0.11	0.29	1.63	forward to right sideways
20	3	13	0.13	0.13	0.01	0.65	0.03	0.03	0.01	0.32	forward to slow right turn
30	4	20	0.34	1.03	-0.02	-6.27	0.07	0.21	-0.01	-3.77	forward to sharp left turn
41	13	3	0.16	0.19	0.03	1.24	0.03	0.04	0.01	0.02	slow right turn to forward
48	20	4	5.00	12.73	-10.02	-40.55	1.00	6.88	-2.14	-1.03	sharp left turn to forward

expression in Eq. (14), it is sufficient to illustrate the improvement in performance over the nominal (*i.e.*, non-robust) framework. For comparison, we also calculated the nominal cost function by considering the nominal trim and maneuver parameters in Eq. (7), and compared both policies in open-loop planning and closed-loop guidance.

5 Results

The following example highlights the improvement in performance and robustness of the robust method over the nominal approach. The helicopter has to fly from the origin to a waypoint $(24.9m, 5.1m)$, where it is supposed to hover. The target radius around the waypoint is $0.5m$, and the final heading is not prescribed. The helicopter is initially flying forward at $3m/s$ (trim 4) in the direction of the Y-axis (0° heading). Figure 1 shows the robust and nominal trajectories as planned in open-loop, *i.e.*, they are only based on the parameters of the trims and the maneuvers as stored in the libraries. The solid lines represent the coasting segments, the dashed lines indicate the maneuvers. Since the maneuver library only contains the final displacements, the maneuvers are plotted as straight lines. The nominal path takes $9.45s$, the robust trajectory arrives after $15.10s$.

As can be seen from the figure, the nominal planning algorithm chooses to make a sharp 90° turn to head straight to the goal, whereas the robust planning algorithm opts for a slower, longer turn. The reason is that the uncertainty in the actual turn angle is larger for a sharp turn, *i.e.*, for a turn with a higher turn rate. Namely, if the banking maneuvers and the actual turn time take shorter/longer than planned, the additional negative/positive change in angle will be larger than it would be in the case of a slow turn.

Figure 2 depicts the resulting trajectories when the planned paths are executed in open-loop in the X-Cell simulation. As can be expected, because of the optimal trade-off between time and uncertainty minimiza-

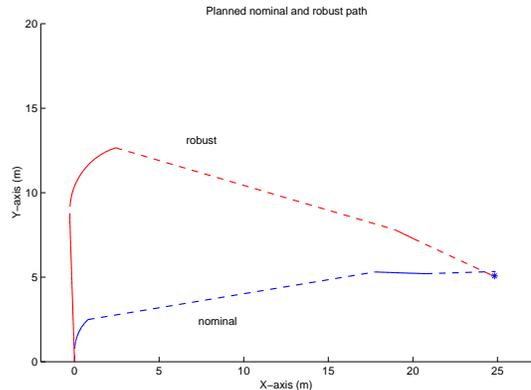


Fig. 1: Nominal and robust path as planned in open-loop. The helicopter is initially in the origin, flying at $3m/s$ North, and has to move to hover mode in position $(24.9m, 5.1m)$. The solid lines represent the coasting segments, the dashed lines indicate the maneuvers.

tion, the robust trajectory remains much closer to the planned path than the nominal trajectory. The latter is significantly perturbed by the uncertainties in the motion primitives.

Finally and most importantly, we compared the nominal and robust policy using closed-loop guidance, in which the guidance policies are applied in real-time based on the actual vehicle state. The resulting trajectories are plotted in Figure 3. Again, the robust policy yields a trajectory that resembles the open-loop planned path of Figure 1 much more closely than the closed-loop path produced by the nominal policy. The nominal trajectory now takes $20.63s$, whereas the robust one reaches the goal region after $18.92s$.

Since in the nominal case the guidance algorithm has to compensate for the perturbed 90° turn (see point A in Fig. 3), the nominal path now takes longer than the robust trajectory. Moreover, the compensating trim and maneuver sequence is again perturbed itself (segment A-B), resulting in an indirect path to the goal. As a result, the actual trajectory and arrival time using the nominal policy differ significantly from the pre-

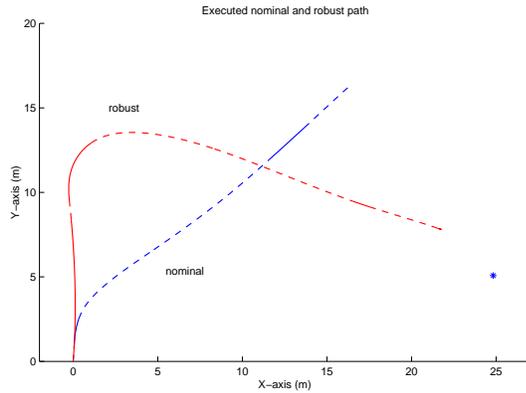


Fig. 2: Trajectories resulting from the open-loop execution of the planned nominal and robust path from Figure 1.

dicted path. In the robust case, however, although the predicted arrival time is greater than the nominal one, the prediction is more accurate. This clearly illustrates the trade-off between minimization of arrival time and uncertainty, and is representative for the typical performance of the two policies.

6 Conclusion

In this paper, we presented a robust guidance method for autonomous vehicles that are modeled as maneuver automata. We extended the hybrid automaton framework introduced by Frazzoli *et al.* to account for uncertainties in the motion primitives. These are accounted for both in the offline computation of an optimal guidance cost function, as well as in a real-time guidance policy. The resulting policy is an optimal trade-off between time-to-goal and uncertainty minimization. We applied the method to a high-fidelity simulation model of MIT’s autonomous X-Cell miniature helicopter, in which we closed the guidance loop by letting the robust planning policy issue optimal commands to the rotorcraft autopilot. We presented an example that highlights the performance improvement achieved with the robust policy: we showed that the original, nominal framework generates suboptimal trajectories in both open- and closed-loop guidance, and that the trajectories obtained by using the robust policy are less sensitive to perturbations in the motion primitives.

Acknowledgements

This work was carried out at the NASA Ames research facility. We would like to thank Vladislav Gavrillets (MIT) for providing the X-Cell helicopter simulation, and Mario Valenti (MIT), Dr. Mark Tischler, Hossein Mansur, Wei Lang and Greg Schulein (NASA) for their

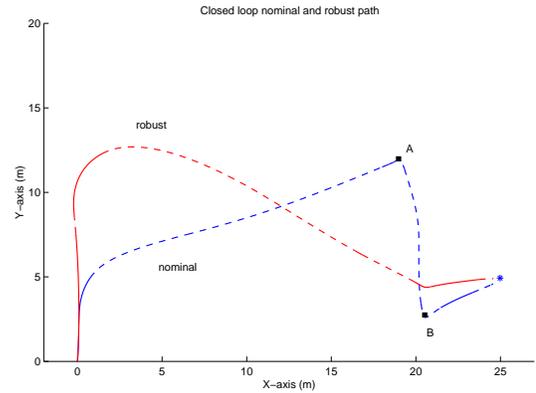


Fig. 3: Trajectories resulting from real-time closed-loop guidance with nominal and robust planning policy.

support during this project. We are also grateful to the reviewers for their useful feedback. Funding for this research was provided by NASA Grant NAG 2-1441. Tom Schouwenaars was partly supported by a Francqui Fellowship of the Belgian American Educational Foundation.

References

- [1] J. Reif. “Complexity of the Mover’s Problem and Generalizations”. *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science*, pages 224–241, 1979.
- [2] V. Gavrillets, B. Mettler, and E. Feron. “Non-linear Model for a Small-Size Acrobatic Helicopter”. Number AIAA 2001-4333, Montreal, QC, August 2001. Proceedings of the AIAA GNC Conference.
- [3] E. Frazzoli, M. A. Dahleh, and E. Feron. “A Hybrid Control Architecture for Aggressive Maneuvering of Autonomous Helicopters”. Phoenix, AZ, December 1999. IEEE Conference on Decision and Control.
- [4] B. Mettler, M. Valenti, T. Schouwenaars, E. Frazzoli, and E. Feron. “Rotorcraft Motion Planning for Agile Maneuvering”. Montreal, QC, June 2002. Proceedings of the 58th AHS Forum.
- [5] N. Lynch, R. Segala, and F. Vaandrager. “Hybrid I/O Automata Revisited”. *Lecture Notes in Computer Science, Springer-Verlag*, 2034:403–417, March 2001.
- [6] E. Frazzoli. “Robust Hybrid Control for Autonomous Vehicle Motion Planning”, PhD Thesis, MIT, June 2001.
- [7] D. Bertekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific, 2nd edition, 2000.
- [8] K. Sprague, V. Gavrillets, D. Dugail, B. Mettler, and E. Feron. “Design and Applications of an Avionics System for a Miniature Acrobatic Helicopter”. Daytona Beach, FL, 2001. AIAA Digital Avionics Systems Conference.