Safe Receding Horizon Control of an Aerial Vehicle

David A. Anisi*
Optimization and Systems Theory
Royal Institute of Technology
100 44, Stockholm, Sweden

Petter Ögren and John W.C. Robinson
Department of Autonomous Systems
Swedish Defence Research Agency
164 90, Stockholm, Sweden

***Abstract*— This paper addresses the problem of designing a real time high performance controller and trajectory generator for air vehicles. The control objective is to use information about terrain and enemy threats to fly low and avoid radar exposure on the way to a given target. The proposed algorithm builds on the well known approach of Receding Horizon Control (RHC) combined with a terminal cost, calculated from a graph representation of the environment. Using a novel safety maneuver, and under an assumption on the maximal terrain inclination, we are able to prove safety as well as task completion. The safety maneuver is incorporated in the short term optimization, which is performed using Nonlinear Programming (NLP). Some key characteristics of the trajectory planner are highlighted through simulations.**

I. Introduction

THIS work deals with the problem of iteratively planning and executing a trajectory towards a given target. Given information about the terrain and positions of enemy threats, the trajectory should use the terrain to reduce exposure to enemy radar by flying low, while at the same time keeping control efforts, as well as total time of flight, small. Furthermore, we would also like to be able to formally verify that the air vehicle will in fact reach the target without crashing into the terrain.

The focus on iterative solutions is due to the fact that with imperfect information, the estimated location of target and threats might *change* during the course of flight. Furthermore, assuming that the problem originates from a complex real-world application, the existence of a truly optimal analytical solutions to the whole trajectory planning problem is deemed unlikely; thus we seek *"good enough" fast computational algorithms*.

In the field of trajectory planning and control, Receding Horizon Control (RHC) is a well known tool to achieve computationally efficient, "good enough", solutions to many unmanned vehicle control problems [1]–[8]. However, an important issue with RHC is to make sure that the greedy, short term optimization does not lead to long term problems. In the vehicle control domain, this often boils down to two things: not getting into situations where a collision is unavoidable, and making sure that the destination is actually reached.

Collision avoidance for ground vehicles or helicopters can be achieved by making sure that every planned trajectory ends in a standstill, [1]. Similarly, for fixed wing aerial vehicles a guaranteed obstacle free circular loitering pattern ensures safety, [2], [5]. Kuwata and How [6], [7] have considered safe RHC of autonomous vehicles in a 2D setting which rely on visibility graphs for environmental representation. However, task completion is not guaranteed.

Task completion has been considered by Richards and How [3], [4]. By augmenting the system with a binary "target state", that indicates whether the target set is reached or not, the authors end up with a hybrid system. Task completion is then guaranteed by imposing a hard terminal equality constraint on the target state at the end of the planning horizon. Although intuitive, this is indeed a very restrictive and computationally demanding constraint that require needlessly long planning horizons.

Most of the papers above address planar problems, while our formulation is in 3D. More importantly however, our work differs from the mentioned papers by the fact that in our case, safety and task completion are intimately connected. This is due to an elaborate choice of safety maneuver, at the end of the short time optimization. The safety maneuver is then augmented by a closed form goal terminating trajectory, that is collision free by design. By iteratively replacing old safe plans with new safe plans, there is always a safe plan available for execution if some step of the update procedure should fail.

The foundation of the present work was laid in [8]. The results there were however only applicable to helicopters and aerial vehicles with a thrust-to-weight ratio larger than one. Since many of today's UAVs have a value in the order of 0.5, the present paper extends the results to include this wider class of cases. Furthermore, the formal arguments in this paper are made on a higher level of abstraction, making the presentation somewhat more lucid.

In what follows, the trajectory optimization problem is presented in Section II. Section III then describes the proposed solution in detail. The properties of safety and task completion are proved in Section IV, followed by simulation examples in Section V. Finally, this paper is concluded in Section VI.

*The corresponding author (anisi@math.kth.se) would like to gratefully acknowledge the financial support by the Department of Autonomous Systems at the Swedish Defence Research Agency.

II. Problem Formulation

In this section we will state the problem in terms of vehicle model and control objectives in detail.

A. Vehicle Model

Let the aerial vehicle model be given in discrete time by

$$\begin{aligned} p_{i+1} &= p_i + h\ v_i \\ v_{i+1} &= v_i + h\ a_i \\ \|v_i\|_\infty &\leq v_{\max} \\ \|a_i\|_\infty &\leq a_{\max} \\ d(p_i) &\leq 0 \\ v_i^T(0,0,1) &\leq \|v_i\| \sin\bar{\gamma}, \end{aligned} \tag{1}$$

i.e. a discrete time, double integrator with time step h and upper bounds on magnitude of speed, $v_i \in \mathbb{R}^3$, and acceleration $a_i \in \mathbb{R}^3$. There is also a terrain collision constraint on position, $p_i \in \mathbb{R}^3$, as well as an upper bound $\bar{\gamma} \in (0, \pi/2]$ of the flight path angle γ. The collision constraint is

$$d(p_i) \triangleq H(x_i, y_i) + h_{\text{min}} - z_i \leq 0,$$

where $H(x_i, y_i)$ denotes the altitude of the terrain at the point $p_i = (x_i, y_i, z_i)$ and $h_{\text{min}} > 0$ denotes a minimum altitude clearance, set by the operator. It should be noted, that the theory below admits considerably more general dynamics and control constraints; as long as the NLP below terminates fast enough, any model can be used.

In order to state safety properties of the proposed controller, we make the following assumption about the terrain.

Assumption 1: The maximal terrain inclination α, is smaller than the vehicle maximum flight path angle $\bar{\gamma}$.

In instances where this assumption does not hold at isolated spots, a virtual "inclination-smoothed" terrain above the real one can be calculated and used instead of the original, thus fulfilling the assumption.

B. Control Objectives

Designing the aerial vehicle controller, we would like to meet the following objectives.

- Avoid ground collisions
- Arrive at target position
- Compute controls in real time
- Allow for information updates
- Achieve a short time of flight
- Use small control effort
- Achieve low threat and radar exposure

Ideally we would like to formally guarantee the first two items, satisfy the following two and minimize an objective function composed of the last three. The objective function might therefore be of the following form

$$\sum_i \|a_i\|_2^2 + bg(p_i) + c, \tag{2}$$

where p_i, a_i are defined above, $b, c > 0$ are scalar weights, g represents a measure of threat and radar exposure and the summation stretches over the whole mission.

C. Problem Statement

We formulate, somewhat loosely, the overall trajectory planning problem in the following way.

Problem 1: Iteratively and efficiently choose controls $\{a_i\}$ such that the model (1) starting from p_0 reaches the target p_{target}, while approximately minimizing the control objective (2).

The term approximately is added, since solving the problem to optimality turns out to be computationally intractable. If, however, we are willing to settle for a good, but not necessarily optimal solution in terms of the objective function (2) above, the problem becomes tractable.

III. Proposed Solution

In this section we will propose a solution to Problem 1 described above. The solution will be a Receding Horizon Control (RHC) scheme using Nonlinear Programming (NLP) for the short term planning. We will further more discuss the choice of terminal cost function for the long term planning, as well as how we can make sure that the vehicle actually arrives at the target without crashing into the terrain.

A. Solution Outline and Receding Horizon Control

As discussed in the introduction, RHC has proved to be a powerful tool to achieve good performance in a computationally tractable way. The main idea in RHC is to divide a planning problem into a short horizon and a long horizon part. Over the short horizon a detailed plan is calculated and over the long horizon only a coarse plan, or no plan at all is needed. The short term plan is then iteratively updated as time evolves, making the horizon of the plan *recede*, always extending a fixed amount of time into the future.

The way we apply RHC to this problem is illustrated in Figure 1. The figure depicts the aerial vehicle and corresponding plan at two different time instants. At the first instant a short term plan (solid and dotted lines) is calculated using NLP, as discussed below. This trajectory includes a safety maneuver (dotted) and is augmented with a goal terminating trajectory (dashed). The existence of the safety maneuver at the end of the short term plan is a *constraint* of the NLP. At the end of the safety maneuver the path is directed towards the target and has the climb angle α, which by definition is equal to the maximal terrain inclination. This fact makes the final part of the trajectory, a steady climb and a steady descent towards the target at α degrees inclination,

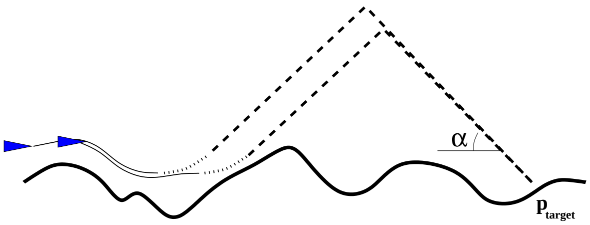

Fig. 1. Two consecutive plans. Note the short term plan (solid and dotted), safety maneuver (dotted), and goal termination trajectory (dashed), reaching the target, p_{target}, on the ground (thick solid).

collision free by design. That a collision free path is obtained also after replanning, i.e. that safety is retained, follows from the observation, [1], that if the new planning for some reason fails, the old plan is still valid and can be executed all the way to the target.

Having a plan, the vehicle now proceeds to execute the first part of the plan over an execution horizon T_e, which must be smaller than the N time step long planning horizon $T_p = Nh$. Figure 1 also depicts the situation after T_e, when the NLP planning is performed once more and a new plan is constructed. [1]

In order to apply the above solution strategy one has to find a way to negotiate the instant 2α turn at the top of the trajectory in some way, so that the constraints in model (1) are met. There are several ways of doing this, as discussed below (see Definition 2), but we shall make the standing assumption that the issue has been resolved for the first planning instance.

Having outlined the principle behind our formulation of the trajectory planning problem as an RHC problem, we now proceed to a somewhat more detailed look at its various components.

B. Short term planning and Nonlinear Programming

The NLP alluded to above, for the solution of the short term planning subproblem, can be described as follows.

At the time of planning, let the vehicle be in state p_c, v_c. Let all planning variables have two subscripts where the first represent the index of the plan, and the second represents the time scale on which the vehicle dynamics in (1) are defined, i.e. $p_{k,i}$ is the planned position of plan k at $h \cdot i$ time units after the plan was initiated.

Definition 1: By a short term plan, we mean a sequence $P_k = \{a_k\}$ that is the best known solution to the following NLP.

$$\begin{aligned}
\underset{a_k}{\text{minimize}} \quad & \sum_{i=1}^{N-1} h\|a_{k,i}\|_2^2 + \Psi(p_{k,N}) \\
\text{s.t.} \quad & p_{k,i+1} = p_{k,i} + h\, v_{k,i} && i = 0, \cdots, N-1 \\
& v_{k,i+1} = v_{k,i} + h\, a_{k,i} && i = 0, \cdots, N-1 \\
& d(p_{k,i}) \leq 0 && i = 0, \cdots, N \\
& \|v_{k,i}\|_\infty \leq v_{\max} && i = 0, \cdots, N \\
& \|a_{k,i}\|_\infty \leq a_{\max} && i = 0, \cdots, N \\
& v_{k,i}^T(0,0,1) \leq \|v_{k,i}\| \sin\bar{\gamma} && i = 0, \cdots, N \\
& p_{k,0} = p_c, \quad v_{k,0} = v_c \\
& \Psi(p_{k-1,N}) \leq \Psi(p_{k,N}) - \epsilon \\
& v_{k,N} = v_{\max}(\cos\phi\cos\alpha, \;\; \sin\phi\cos\alpha, \;\; \sin\alpha) \\
& \phi = \text{arctan2}((p_{target} - p_{k,N})^T(0,1,0), \\
& \qquad\qquad (p_{target} - p_{k,N})^T(1,0,0))
\end{aligned} \tag{3}$$

where k is plan index, $N = T_p/h$, and the terminal cost, $\Psi : \mathbb{R}^3 \to \mathbb{R}^+$, and cost decrement margin, $\epsilon > 0$, will be defined below. The final two constraints makes the last part of the trajectory climb in the direction of the target. All other variables are defined as their counterparts, without index k, in (1). If there is no feasible solution to (3) then the short term plan is undefined.

Remark 1: As part of the RHC scheme, we will always be on our way to execute an old plan when a new plan is constructed. Thus the old plan can often serve as a feasible starting solution in the optimization. This only fails when the descent part of the goal terminating trajectory falls within the new T_p and the α-constraint is not satisfied. If no new feasible trajectory is found, the execution continues along the old plan.

C. Goal termination trajectory

In this section we will see how the short term plan produced by the NLP can be canonically extended all the way to the target, in a way that is collision free by design, as suggested in Figure 1.

Definition 2: By an extended plan, we mean a short term plan $P_k = \{a_k\}$ over time horizon T_p, augmented with a straight line steady climb at α degrees towards the target, followed by a maximum rate turn and a straight line descent ending at the target. The dive is initiated at a point p_{top}, defined as the top vertex of a triangle, $\{p_{target}, p_{top}, p_{k,N}\}$, in the vertical plane with edge inclinations α, for the two upper edges. In instances when p_{top} and p_{target} are so close that the dive towards p_{target} violates the vehicles dive constraints, or when p_{target} is inside the maximum turn circle from p_{top} and cannot be reached, the extended plan is undefined. Likewise, if no feasible short term plan is found in the NLP, the extended plan is undefined.

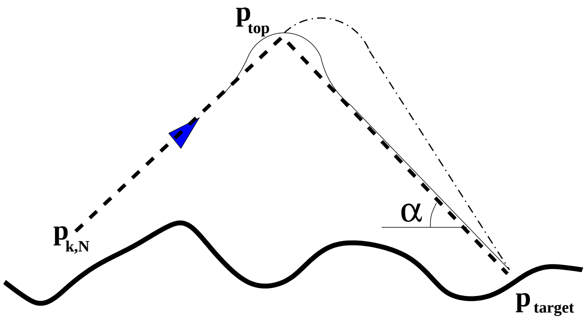
$\mathbf{p}_{\text{top}}$
$\mathbf{p}_{\text{k,N}}$
α
$\mathbf{p}_{\text{target}}$

Fig. 2. Two alternatives for forming a goal termination strategy. The dash-dotted curve illustrates the alternative described in Definition 2. Note that this is only one of many ways to reach the target while flying above the safe dashed line and satisfying the vehicle dynamics (1). If the gap between $\bar{\gamma}$ and α is big enough, the thin solid curve is a less conservative such option. The important thing is that the construction is of low computational complexity.

Remark 2: When planning a path on-line and either the NLP (3), or the formation of an extended plan according to Definition 2 fails, no new plan is selected and the aerial vehicle keeps executing the old plan.

[1] It is useful, but not necessary, that T_p, T_e are chosen so that the safety maneuver is shorter in duration than $T_p - T_e$ since then the replanning commences before the vehicle has started the safety maneuver. In fact, it is in general desirable that the vehicle actually *never* has to execute the safety maneuver.

Remark 3: Note that any choice of extension trajectory above the triangle $\{p_{target}, p_{top}, p_{k,N}\}$ is also collision free by design. By using information about the highest terrain altitude one can form more elaborate, and less conservative, extension trajectories. [2]

D. Long term planning and terminal cost

The purpose of the terminal cost, $\Psi : \mathbb{R}^3 \to \mathbb{R}^+$, in the NLP (3) is to guide the short term plan in directions that make the flown overall trajectory good, in terms of the objective function (2). It will also be used below to make sure there is not an infinite number of plan changes.

As in e.g. [1], [9], the terminal cost function, Ψ, is derived from a shortest path problem in a graph. This is done in a standard fashion and is therefore only described briefly here.

The graph is created by taking a horizontally equidistant mesh and using the terrain height $z = H(x, y)$ as values for the nodes. Above this set of nodes, four additional node layers are added at $z = H(x, y) + j600m,\ j = 1 \dots 4$. Each node is then connected by edges to its 8 neighbors in the same layer and the 9 neighbors above and below.

The edge cost is a weighted sum of Euclidean distance and threat exposure, in accordance with the choices of $g(\cdot), b, c$ in the objective function (2). In order to use Ψ to decide when to change plans, we demand that Ψ is always positive, hence $\Psi : \mathbb{R}^3 \to \mathbb{R}^+$.

Again, as in [1], [9] we calculate the optimal cost to go from each node using a Dijkstra type of algorithm and then interpolate the node values to find Ψ.

This is also a convenient point at which to mention that the possibility of updating the "off-line" computed cost to go should not be overlooked. The term "off-line" is rather to be interpreted as, at a much slower sampling rate than the control loop, *i.e.* in the order of tens of seconds (*cf.* [10]). As new information about the environment or mission objectives is gathered, it can be processed and fed back regularly to the vehicle through an updated terminal cost.

E. The Algorithm

In this section we state the proposed algorithm, after making an initial assumption.

Assumption 2: The first step in Algorithm 1 below returns a feasible solution.

In view of the discussion above, this assumption is very reasonable and only excludes cases like when the target position is extremely close to the launch position, and can not be reached even with a maximal turn.

Algorithm 1:

1) The vehicle is launched at inclination α and velocity $v_{\max}$ in the direction of the target. At launch the vehicle follows a default plan constructed by augmenting the initial state according to Definition 2.
2) The vehicle executes the given plan for time T_e
3) While executing, a new plan is sought according to Definitions 1 and 2.
4) If new informations arrive on e.g. threat positions, and Ψ needs to be recalculated, this is performed in the background. Once a new Ψ is calculated it is applied in Definition 1 as well as in 5 below.
5) Let $\epsilon > 0$ be fixed throughout the mission.
   If a new plan P_{k+1} according to Definitions 1 and 2 is found, then the new plan is activated instead of the old one.
6) If the target is not reached, go to 2.

Remark 4: The choice of proceeding along the old plan or switching to a new one, thereby incrementally improving the terminal cost, can be viewed as a conditional version of the satisficing control strategies used in some forms of RHC. In these RHC applications, the terminal cost acts as a control Lyapunov function with which one can construct a control with local optimality properties and guaranteed global stability. In the trajectory optimization problem considered here, the counterpart of stability is safety and task completion, which are both already guaranteed by any single plan. The terminal cost function here acts by providing a means of improving the cost for completing the mission by changing to a new plan (while retaining safety and task completion) which is executed on the condition that the cost to go is incrementally decreased.

In the next section we will analyze the performance of the algorithm.

IV. Theoretical Properties

In this section we will prove that the proposed solution will indeed solve Problem 1.

Proposition 1: Algorithm 1 solves Problem 1, i.e. the aerial vehicle will reach the target in finite time without colliding with the terrain.

Proof: First note that each accepted plan is safe and reaches the goal by construction. So if there is only a finite number of changes of plan the aerial vehicle will reach the target. Then note that the Ψ-value of each accepted plan must be at least ϵ better than the previous one. Since $\Psi \geq 0$, there can at most be $\lceil \Psi(p_0)/\epsilon \rceil$ changes.

Furthermore, the algorithm is iterative by design, and the RHC scheme makes it computationally efficient. Finally, the objective function (2) is approximately minimized since $||a_i||_2^2$ is present in the NLP and Ψ accounts for the path length and threat exposure, i.e. $bg(p_k)$ and c in (2).

This proves the proposition. ■

Remark 5: Finite time completion can also be proved using the above framework. If e.g. the vehicle starts out with a user defined upper bound on time over target, and only new plans that satisfy the bound are accepted, this property will be guaranteed. The argument works equally well with a given fuel limit.

V. Simulations

In this section, we present a small selection of the simulations made with the proposed trajectory planning algorithm.

For environmental representation, real terrain elevation data over the Cascade mountains, WA, have been used (see

[2] Since the planning is supposed to be updated iteratively, the extension part of the trajectories are never executed during a normal mission.

Figure 3). The dataset used is a subset extracted from the one

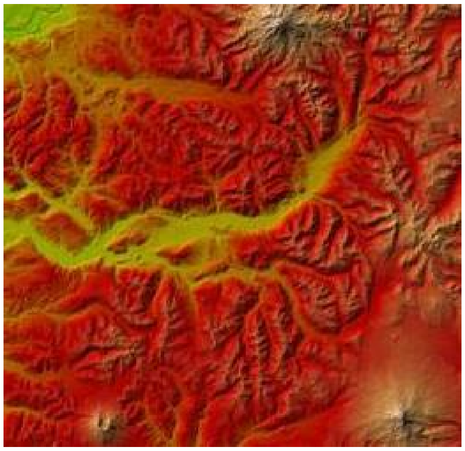

Fig. 3. The terrain elevation map used in the simulations represents an area of more than 82km $\times$82km taken from the Cascade range, WA. It contains the summits of Mt. Rainier, Mt. Adams and Mt. St. Helen's.

appearing in Reference [11]. The full-resolution elevation image, is made up of $16,385 \times 16,385$ nodes at 10 meters horizontal spacing. The vertical resolution is 0.1 meters. This dataset occupies roughly 5 GB on disk and is therefore impractical to work with. However, as will be seen from the simulation results, but also pointed out in [10], the environment should be decomposed in a manner that is consistent with the maneuvering capabilities of the vehicle. Therefore, this high level of accuracy is not needed to capture the global characteristics of the environment by the terminal cost, Ψ. The lower-resolution maps used in the simulations have therefore been sub-sampled at every 16^{th} and 256^{th} instance, resulting in a inter-pixel spacing of 160 and 2560 meters respectively. In the vertical direction, there are five horizontal layers with 600 meters in between. The vertical positions of each node depend on the altitude of the terrain at that particular point of the map, as explained above. The non-uniform grid built this way, can be seen as stretching out the layers of a uniform grid on the terrain surface.

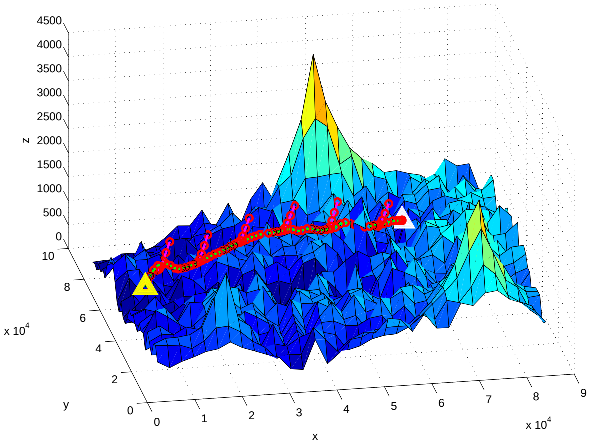

Fig. 4. The resulting trajectory, and a subset of the planned safety maneuvers.

Running the algorithm we get the trajectory shown in Figure 4. The plot also shows every tenth safety maneuver. Note that none of these were actually executed. New improved plans were iteratively replacing the old ones, before the vehicle reached the climbing part of the trajectories.

As discussed earlier, it is the terminal cost, $\Psi(p)$, that captures the global characteristics of both the environment and the mission objectives. This is readily done by varying the costs in the graph representation of the environment. Figure 5 shows the effect of switching on a radar having a detection radius of 10km. The position of the radar is marked with a black triangle, while yellow circles are used to map out the volume where the vehicle is visible to the radar. The path with circles, shows the outcome of the trajectory planner when the radar is not accounted for. Unaware of its existence, the generated path passes right through the detection area of the radar. The other path, namely the one marked with squares, shows the outcome when the terminal cost incorporates the radar. The threat exposure is now minimized by flying at a much lower altitude, utilizing the protection provided by the terrain and thereby avoiding radar detection.

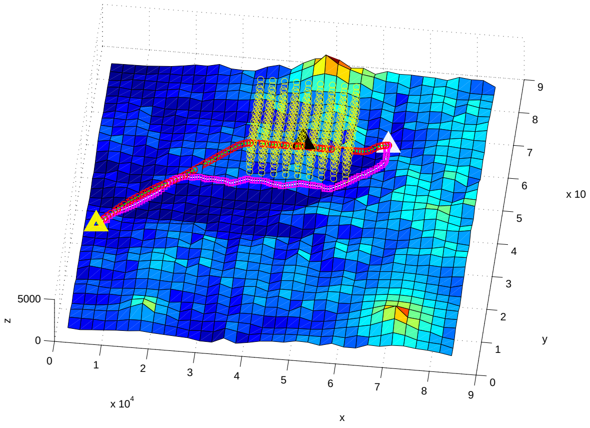

Fig. 5. The effect of threat exposure on the generated path.

VI. Conclusion

The solution to the on-line trajectory optimization problem presented in this paper uses a novel combination of ideas to guarantee safety and finite time task completion. A central idea is to use a conditional plan changing strategy, where, starting from a feasible solution, a new plan for the remaining part of the mission is only accepted if it gives an incremental decrease to the terminal cost. The key problem is then to retain safety and finite time task completion when the plan is changed. This problem is solved by making a safety maneuver, which guarantees both obstacle avoidance and existence of a goal termination trajectory, a condition which is incorporated into the short term planning NLP problem. The safety maneuver also has the beneficial side effect that it makes it easy to cope with hard real-time constraints. The choice of whether to change to a new plan or keep the old one is based on the terminal cost Ψ which is meant to act as a more easily computed (but conservative) alternative to the (optimal) cost to go. The terminal cost is determined off-line, or on a much more slowly evolving time scale than the short term planning, and this makes it possible to reduce the real-time computational burden drastically.

References

[1] P. Ögren and N. Leonard, “A convergent dynamic window approach to obstacle avoidance,” *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 188–195, 2005.

[2] T. Schouwenaars, J. How, and E. Feron, “Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees,” *AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island, August*, 2004.

[3] A. Richards and J. P. How, “Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility,” in *Proc. of the IEEE American Control Conference*, vol. 5, Jun. 2003, pp. 4034–4040.

[4] ——, “Robust variable horizon model predictive control for vehicle maneuvering,” *Internat. J. Robust Nonlinear Control*, vol. 16, no. 7, pp. 333–351, 2006.

[5] T. Schouwenaars, J. How, and E. Feron, “Receding horizon path planning with implicit safety guarantees,” in *Proc. of the IEEE American Control Conference*, vol. 6, 2004, pp. 5576–5581.

[6] Y. Kuwata and J. P. How, “Stable trajectory design for highly constrained environments using receding horizon control,” in *Proc. of the IEEE American Control Conference*, vol. 1, 2004, pp. 902–907.

[7] J. Bellingham, Y. Kuwata, and J. P. How, “Stable receding horizon trajectory control for complex environments,” in *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, Aug. 2003.

[8] D. A. Anisi, J. W. Robinson, and P. Ögren, “On-line trajectory planning for aerial vehicles: a safe approach with guaranteed task completion,” in *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, Aug. 2006.

[9] Y. Kuwata and J. How, “Three Dimensional Receding Horizon Control for UAVs,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 16–19, 2004.

[10] B. Mettler and E. Bachelder, “Combining on- and offline optimization techniques for efficient autonomous vehicle’s trajectory planning,” in *Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, CA, USA, Aug. 2005.

[11] P. Lindström and V. Pascucci, “Visualization of large terrains made easy,” in *Proc. of the IEEE Visualization, VIS ’01*, San Diego, CA, USA, Oct. 2001, pp. 363–574, [Online] http://www.cc.gatech.edu/projects/large_models/ps.html.