Sampling-based path planning: a new tool for missile guidance

P. Pharpatara * **B. Hérissé** * **R. Pepy** * **Y. Bestaoui** **

* *Onera - The French Aerospace Lab, Palaiseau, France (Tel: +33 1 80 38 66 48; e-mail: pawit.pharpatara@onera.fr)*
** *IBISC, Université d'Evry-Val-d'Essonne, Evry, France (Tel: +33 1 69 47 75 05; e-mail: Yasmina.Bestaoui@ufrst.univ-evry.fr)*

Abstract: A new missile midcourse guidance algorithm is proposed in this paper. It is a combination of sampling based path planning, Dubins' curves and classical guidance laws. Moreover, a realistic interceptor missile model is used. It allows to anticipate the future changes of flight conditions along the trajectory, especially the loss of maneuverability at high altitude. Simulation results are presented to demonstrate the substantial performance improvements over classical midcourse guidance laws.

Keywords: trajectory planning; optimal control; missiles; midcourse guidance.

1. INTRODUCTION

The guidance of an interceptor missile to a target includes two main stages after the launch: the midcourse guidance and the terminal guidance. The midcourse guidance consists in ensuring a proper collision course to the target so that the seeker can detect the target before engaging the terminal homing phase. While a Proportional Navigation (PN) law is often sufficient for the homing phase, trajectory shaping is necessary for the midcourse phase to fulfil the conditions for target detection. Furthermore, for long-range missile, an acceptable velocity before the endgame needs to be ensured. Therefore, the midcourse guidance problem is an optimal problem with constraints.

This optimal problem can be solved numerically using optimal control theory. However, such a nonlinear two-point boundary value problem cannot be solved in real time on a recent embedded computer. To avoid numerical problems, many closed-loop optimal guidance laws were proposed in the past using singular perturbation theory (Cheng and Gupta [1986], Menon and Briggs [1990], Dougherty and Speyer [1997]), linear quadratic regulators (Imado et al. [1990], Imado and Kuroda [1992]), analytical methods (Lin and Tsai [1987]) or modified proportional guidance (Newman [1996]). The kappa guidance detailed in (Lin [1991]) is certainly the most known of these guidance laws. It is designed using optimal control theory so that the missile final speed is maximized. Thus, gains of the control law are updated depending on current flight conditions. However, control limitations (saturations) are difficult to satisfy using this type of guidance law. Likewise, the kappa guidance is not suitable for a surface-to-air missile dedicated to high altitude interception since air density decreases with increasing altitudes. For such complex systems and missions, the optimal problem needs to be considered globally.
More recently, neural networks were trained with precomputed optimal trajectories (Song and Tahk [2001, 2002]) and used as a feedback control law to obtain almost optimal trajectories. However, neural networks cannot be trained with an exhaustive amount of optimal trajectories that would cover any possible configurations. Fuzzy logic techniques have also been investigated for the design of a midcourse guidance law (Lin and Chen [2000], Lin et al. [2004]). Nevertheless, control constraints remain difficult to satisfy.

In parallel with work on missile guidance problems, many studies in robotic field address optimal control problems and path planning. From the robotic point of view, a missile is a non-holonomic system moving forward. Although a missile is much more complex, the well-known Dubins' vehicle (Dubins [1957]) is the most similar system in mobile robotic. Very little work on midcourse guidance is based on Dubins' results to find minimum path length trajectories, though a recent work on reachability guidance is one of the first attempts (Robb et al. [2005]). However, the original result by Dubins only addresses non-varying systems while the maneuverability of a long-range missile can vary substantially along the trajectory due to varying control constraints. Therefore, Dubins' results are not sufficient to solve the path planning problems for a missile.
Sampling-based path planning methods, such as Rapidly-exploring Random Trees (RRT) (LaValle [2006]) or Probabilistic Roadmap Methods (PRM) (Kavraki et al. [1996]), offer solutions for trajectory shaping in complex environments while a classical optimal method often fails to find a solution. These are usually used for path planning of a Dubins' car in environments cluttered by obstacles (LaValle and Kuffner [2001]). The main advantage of these techniques is that even a complex system can be considered without the need for approximations (Pepy et al. [2006]). The idea of this paper is to use such a sampling-based method.

The proposed method in this paper consists in using the RRT algorithm for the trajectory shaping of a missile midcourse guidance stage. The principle is to use Dubins' results to define the metric of the algorithm. A PN law is used to steer the vehicle. Terminal constraints are

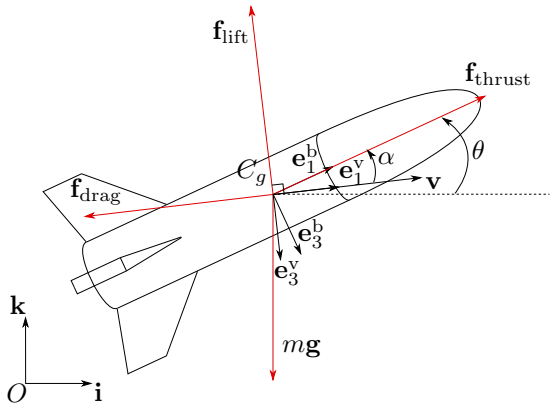


Fig. 1. Definition of reference frames

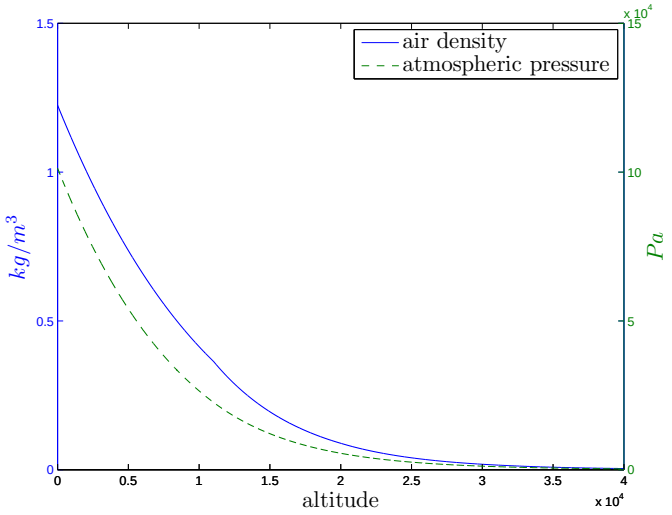


Fig. 2. The variation of air density and atmospheric pressure with altitude : US-76 model

defined with respect to the Predicted Interception Point (PIP) and the interceptor capabilities for the terminal guidance stage. Simulations results are obtained on a missile model consisting of one propulsion stage. The proposed approach demonstrates promising performance for trajectory planning in some high altitude cases where existing methods, such as kappa guidance, fail.

This paper is organized in three sections followed by a conclusion and some perspectives. Section 2 introduces the missile model and the problem. In Section 3, the RRT algorithm is described. Section 4 presents some simulation results obtained with the proposed approach.

2. PROBLEM STATEMENT

2.1 System model

The missile is modeled as a rigid body of mass m and inertia I maneuvering in a vertical 2D plane. A round earth model is used. Due to small flight times (less than one minute), the earth rotation has very little effect on the missile and is neglected in this paper. Three frames (Fig. 1) are introduced to describe the motion of the vehicle: an earth-centred earth-fixed (ECEF) reference frame $\mathcal{I}$ centered at point O and associated with the vector basis $(\mathbf{i}, \mathbf{k})$; a body-fixed frame $\mathcal{B}$ attached to the vehicle at its center of mass C_g with the vector basis $(\mathbf{e}_1^{\text{b}}, \mathbf{e}_3^{\text{b}})$; and a velocity frame $\mathcal{V}$ attached to the vehicle at C_g with the vector basis $(\mathbf{e}_1^{\text{v}}, \mathbf{e}_3^{\text{v}})$ where $\mathbf{e}_1^{\text{v}} \stackrel{\text{def}}{=} \frac{\mathbf{v}}{\|\mathbf{v}\|}$ and $\mathbf{v}$ is the translational velocity of the vehicle in $\mathcal{I}$. Position and velocity defined in $\mathcal{I}$ are denoted $\boldsymbol{\xi} = (x, z)^\top$ and $\mathbf{v} = (\dot{x}, \dot{z})^\top$. The translational velocity $\mathbf{v}$ is assumed to coincide with the apparent velocity (no wind assumption). The orientation of the missile is represented by the pitch angle θ from horizontal axis to $\mathbf{e}_1^{\text{b}}$. The angular velocity is defined in $\mathcal{B}$ as $q \stackrel{\text{def}}{=} \dot{\theta}$.

Translational forces include the thrust $\mathbf{f}_{\text{thrust}}$, the force of lift $\mathbf{f}_{\text{lift}}$, the force of drag $\mathbf{f}_{\text{drag}}$ and the force due to the gravitational acceleration $mg\mathbf{k}$ (Fig. 1). Aerodynamic torque is denoted τ_{aero} and perturbation torque is denoted τ_{pert}. Using these notations, the vehicle dynamics can be written as

$$\dot{\boldsymbol{\xi}} = \mathbf{v}, \tag{1}$$
$$m\dot{\mathbf{v}} = \mathbf{f}_{\text{drag}} + \mathbf{f}_{\text{lift}} + \mathbf{f}_{\text{thrust}} - mg\mathbf{k}, \tag{2}$$
$$\dot{\theta} = q, \tag{3}$$
$$I\dot{q} = \tau_{\text{aero}} + \tau_{\text{pert}}. \tag{4}$$

The aerodynamic forces are

$$\begin{aligned} \mathbf{f}_{\text{drag}} &= -\frac{1}{2}\rho v^2 S C_{\text{D}} \mathbf{e}_1^{\text{v}}, \\ \mathbf{f}_{\text{lift}} &= -\frac{1}{2}\rho v^2 S C_{\text{L}} \mathbf{e}_3^{\text{v}}, \end{aligned} \tag{5}$$

where ρ is the air density; S is the missile reference area, C_{D} is the drag coefficient, C_{L} is the lift coefficient, and $v \stackrel{\text{def}}{=} \|\mathbf{v}\|$. C_{L} and C_{D} both depend on the angle of attack α (Fig. 1) and on the longitudinal speed of the missile.

The thrust force is applied until the boost phase stops, *i.e.* as long as $t \leq t_{\text{boost}}$. It is assumed to be constant and is defined as

$$\mathbf{f}_{\text{thrust}} = \left(\text{I}_{\text{sp}}(\text{Vac}) g_0 q_{\text{t}} - A_{\text{e}} p_0\right) \mathbf{e}_1^{\text{b}}, \tag{6}$$

where $\text{I}_{\text{sp}}(\text{Vac})$ is the vacuum specific impulse, $q_{\text{t}} = -\dot{m}$ is the mass flow rate of exhaust gas, g_0 is the gravitational acceleration at sea level, A_{e} is the cross-sectional area of nozzle exhaust exit and p_0 is the external ambient pressure. Mass m and inertia I are time-varying values during the propulsion stage.

A hierarchical controller is used to control the lateral acceleration $\mathbf{a}^{\text{v}} = a^{\text{v}}\mathbf{e}_3^{\text{v}}$ perpendicular to $\mathbf{v}$: an inner loop stabilizes the rotational velocity q of the vehicle and an outer loop controls the lateral acceleration $\mathbf{a}^{\text{v}}$ (Devaud et al. [2000]). In the following, $\mathbf{a}_{\text{c}}^{\text{v}}$ denotes the setpoint of this control loop.

2.2 Atmospheric model

The US Standard Atmosphere, 1976 (US-76) is used in this paper. In the lower earth atmosphere (altitude $<$ 35 km), density of air and atmospheric pressure decrease exponentially with altitude and approach zero at about 35 km (Fig. 2). As we consider a missile with only aerodynamic flight controls, the maneuvering capabilities

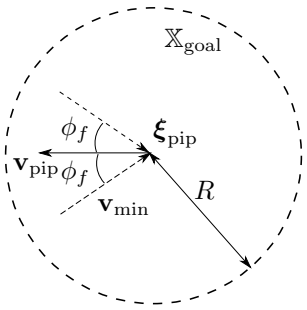


Fig. 3. $\mathbb{X}_{\text{goal}}$

are linked to the density of air (5) and approaches zero at 35 km.

2.3 Mission and requirements

Let $\mathbf{x}(t) = (\boldsymbol{\xi}(t), \mathbf{v}(t)) \in \mathbb{X} \subseteq \mathbb{R}^4$ be the state of the system, $\mathbf{a}_{\text{c}}^{\text{v}} \in \mathbb{U}(t, \mathbf{x}) \subseteq \mathbb{R}^3$ be an admissible control input and consider the differential system

$$\dot{\mathbf{x}} = f(t, \mathbf{x}, \mathbf{a}_{\text{c}}^{\text{v}}), \tag{7}$$

where f is defined in section 2.1.

$\mathbb{X} \subseteq \mathbb{R}^4$ is the state space. It is divided in two subsets. Let $\mathbb{X}_{\text{free}}$ be the set of admissible states and let $\mathbb{X}_{\text{obs}} = \mathbb{X} \backslash \mathbb{X}_{\text{free}}$ be the *obstacle region i.e.* the set of non-admissible states. In this paper, $\mathbb{X}_{\text{free}}$ is defined as

$$\mathbb{X}_{\text{free}} = \{\mathbf{x} : \text{altitude}(\boldsymbol{\xi}) > 0, \|\mathbf{v}(t > t_{\text{boost}})\| > v_{\text{min}}\}, \tag{8}$$

where v_{min} is the minimum tolerated interceptor speed at the interception point defined by the performance of the terminal phase of the interceptor.

The initial state of the system is $\mathbf{x}_{\text{init}} \in \mathbb{X}_{\text{free}}$.

The path planning algorithm is given a predicted interception point $\mathbf{x}_{\text{pip}} = (\boldsymbol{\xi}_{\text{pip}}, \mathbf{v}_{\text{pip}})$. In order to achieve its mission, the interceptor has to reach a goal set $\mathbb{X}_{\text{goal}}$, shown in Fig. 3, defined as

$$\mathbb{X}_{\text{goal}} = \{\mathbf{x} : \|\boldsymbol{\xi} - \boldsymbol{\xi}_{\text{pip}}\| < R, \|\mathbf{v}\| \geqslant v_{\text{min}}, \angle(\mathbf{v}, -\mathbf{v}_{\text{pip}}) < \phi_f\} \tag{9}$$

where R is the radius of a sphere centered at $\boldsymbol{\xi}_{\text{pip}}$ and ϕ_f is related to the maximum allowed aspect angle. Values of R, ϕ_f and v_{min} are defined by the homing loop performance of the interceptor.

$\mathbb{U}(t, \mathbf{x})$ is the set of admissible control inputs at the time t, when the state of the system is $\mathbf{x}$:

$$\mathbb{U}(t, \mathbf{x}) = \{\mathbf{a}_{\text{c}}^{\text{v}} : \alpha_c \leqslant \alpha_{\text{max}}(t, \mathbf{x})\} \tag{10}$$

where α_c is the needed angle of attack to obtain the control input $\mathbf{a}_{\text{c}}^{\text{v}}$ and $\alpha_{\text{max}}(t, \mathbf{x})$ is the maximum tolerated value of the angle of attack which depends on t and $\mathbf{x}$. It is defined by

$$\alpha_{\text{max}}(t, \mathbf{x}) = \min\left(\alpha_{\text{max}}^{\text{stb}}(t, \mathbf{x}), \alpha_{\text{max}}^{\text{struct}}(t, \mathbf{x})\right). \tag{11}$$

$\alpha_{\text{max}}^{\text{stb}}(t, \mathbf{x})$ is the maximum achievable angle of attack using tail fins. This value depends on altitude and speed of the missile. It is given by wind tunnel experiments. $\alpha_{\text{max}}^{\text{struct}}(t, \mathbf{x})$ is the structural limit which is given by the maximum lateral acceleration $\mathbf{a}_{\text{max}}^{\text{b}}$ in body frame that the missile can suffer before it breaks.

The motion planning problem is to find a collision free trajectory $\mathbf{x}(t) : [0, t_f] \to \mathbb{X}_{\text{free}}$ with $\dot{\mathbf{x}} = f(t, \mathbf{x}, \mathbf{a}_{\text{c}}^{\text{v}})$, that starts at $\mathbf{x}_{\text{init}}$ and reaches the goal region, *i.e.* $\mathbf{x}(0) = \mathbf{x}_{\text{init}}$ and $\mathbf{x}(t_f) \in \mathbb{X}_{\text{goal}}$.

Algorithm 1 RRT path planner

Function : build_rrt(in : $K \in \mathbb{N}$, $\mathbf{x}_{\text{init}} \in \mathbb{X}_{\text{free}}$, $\mathbb{X}_{\text{goal}} \subset \mathbb{X}_{\text{free}}$, $\Delta t \in \mathbb{R}^+$, out : G)

1: $G \leftarrow \mathbf{x}_{\text{init}}$
2: $i = 0$
3: **repeat**
4: $\quad \mathbf{x}_{\text{rand}} \leftarrow$ random_state($\mathbb{X}_{\text{free}}$)
5: $\quad \mathbf{x}_{\text{new}} \leftarrow$ rrt_extend($G, \mathbf{x}_{\text{rand}}$)
6: **until** $i++ > K$ or ($\mathbf{x}_{\text{new}} \neq$ null and $\mathbf{x}_{\text{new}} \in \mathbb{X}_{\text{goal}}$)
7: **return** G

Function : rrt_extend(in : G, $\mathbf{x}_{\text{rand}}$, out : $\mathbf{x}_{\text{new}}$)

8: $\mathbf{x}_{\text{near}} \leftarrow$ nearest_neighbour($G, \mathbf{x}_{\text{rand}}$)
9: $\mathbf{a}_{\text{RRT}}^{\text{v}} \leftarrow$ select_input($\mathbf{x}_{\text{rand}}, \mathbf{x}_{\text{near}}$)
10: $(\mathbf{x}_{\text{new}}, \mathbf{a}_{\text{c}}^{\text{v}}) \leftarrow$ new_state($\mathbf{x}_{\text{near}}, \mathbf{a}_{\text{RRT}}^{\text{v}}, \Delta t$)
11: **if** collision_free_path($\mathbf{x}_{\text{near}}, \mathbf{x}_{\text{new}}, \Delta t$) **then**
12: $\quad G$.AddNode($\mathbf{x}_{\text{new}}$)
13: $\quad G$.AddEdge($\mathbf{x}_{\text{near}}, \mathbf{x}_{\text{new}}, \mathbf{a}_{\text{c}}^{\text{v}}$)
14: $\quad$ **return** $\mathbf{x}_{\text{new}}$
15: **else**
16: $\quad$ **return** null
17: **end if**

3. ALGORITHM DESCRIPTION

3.1 Rapidly-exploring Random Trees

Rapidly-exploring Random Trees (RRT) (LaValle and Kuffner [2001]) is an incremental method designed to efficiently explore non-convex high-dimensional spaces. The key idea is to visit unexplored part of the state space by breaking its large Voronoi areas (Voronoi [1907]). Algorithm 1 describes the principle of the RRT when used as a path planner.

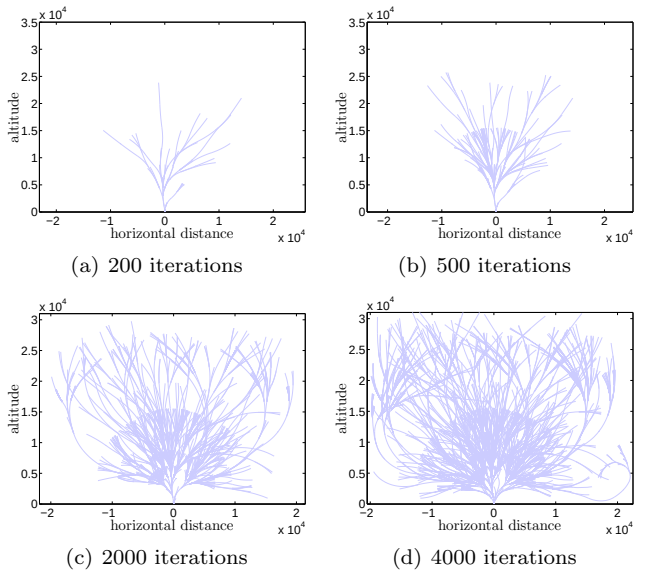
First, the initial state $\mathbf{x}_{\text{init}}$ is added to the tree G. Then, a state $\mathbf{x}_{\text{rand}} \in \mathbb{X}_{\text{free}}$ is randomly chosen. The `nearest_neighbor` function searches the tree G for the nearest node to $\mathbf{x}_{\text{rand}}$ according to a metric d (section 3.2). This state is called $\mathbf{x}_{\text{near}}$. The `select_input` function selects a control input $\mathbf{a}_{\text{RRT}}^{\text{v}}$ according to a specified criterion (section 3.3) to connect $\mathbf{x}_{\text{near}}$ to $\mathbf{x}_{\text{rand}}$. Equations (1) and (2) are then integrated on the time increment Δt using $\mathbf{a}_{\text{RRT}}^{\text{v}}$ and $\mathbf{x}_{\text{near}}$ to generates $\mathbf{x}_{\text{new}}$ (`new_state` function). At line 11, a collision test (`collision_free_path` function) is performed: if the path between $\mathbf{x}_{\text{near}}$ and $\mathbf{x}_{\text{new}}$ lies in $\mathbb{X}_{\text{free}}$ then $\mathbf{x}_{\text{new}}$ is added to the tree (lines 12 and 13).

These steps are repeated until the algorithm reaches K iterations or when a path is found, *i.e.* $\mathbf{x}_{\text{new}} \in \mathbb{X}_{\text{goal}}$.

Figure 4 illustrates RRT expansion.

3.2 `nearest_neighbour`

$\mathbf{x}_{\text{near}}$ is defined as the nearest state to $\mathbf{x}_{\text{rand}}$ according to a specified metric. While the euclidean distance is suitable for holonomic systems, this cannot measure the true distance between two states for non-holonomic vehicles. Indeed, initial and final orientations of the velocity vector $\mathbf{v}$ need to be taken into account. A metric that uses



(a) 200 iterations (b) 500 iterations

(c) 2000 iterations (d) 4000 iterations

Fig. 4. RRT expansion

Dubins' results (Dubins [1957], Boissonnat et al. [1991]) is proposed in this paragraph.

Let $\gamma = \theta - \alpha$ denote the flight path angle of the vehicle. Recall the translational dynamics (1) and (2) of the missile while ignoring the dynamics of the norm v of the velocity vector $\mathbf{v}$, then rewrite it as follows:

$$\begin{cases} \dot{x} = v\cos(\gamma) \\ \dot{z} = v\sin(\gamma) \\ \dot{\gamma} = \dfrac{a^{\mathrm{v}}}{v} \end{cases} \tag{12}$$

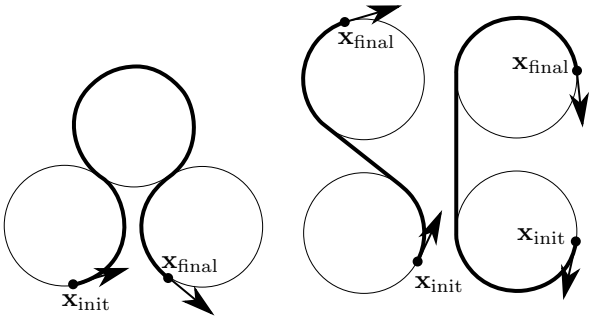
where $\mathbf{a}^{\mathrm{v}} = a^{\mathrm{v}}\mathbf{e}_3^{\mathrm{v}}$. Assume that the control of the lateral acceleration is perfect, that is $\forall t$, $\mathbf{a}^{\mathrm{v}} \equiv \mathbf{a}_{\mathrm{c}}^{\mathrm{v}} \in \mathbb{U}(t, \mathbf{x})$.

System (12) is similar to the system considered in (Dubins [1957]), often called the Dubins' car. This is a non-holonomic vehicle moving forward with a constant velocity and capable of maneuvering with a bounded curvature. The velocity v is not constant in (12) due to drag and thrust forces. In order to use Dubins' results, a change of variable from time t to curvilinear abscissa $s = \int_0^t v(u)\,du$ is made. Thus, (12) becomes

$$\begin{cases} x' = \dfrac{dx}{ds} = \cos(\gamma) \\ z' = \dfrac{dz}{ds} = \sin(\gamma) \\ \gamma' = \dfrac{d\gamma}{ds} = c \end{cases} \tag{13}$$

where $c = \frac{a^{\mathrm{v}}}{v^2}$ is the curvature of the vehicle trajectory. Since $\mathbf{a}^{\mathrm{v}} \in \mathbb{U}(t, \mathbf{x})$, the maximum curvature $c_{\max}(t, \mathbf{x})$ depends on t and $\mathbf{x}$ along the trajectory.

In (Dubins [1957]), $c_{\max}$ was considered as a constant along the trajectory and minimum length paths from an initial state $(x_{\mathrm{init}}, z_{\mathrm{init}}, \gamma_{\mathrm{init}})$ to a final state $(x_{\mathrm{final}}, z_{\mathrm{final}}, \gamma_{\mathrm{final}})$ were analyzed. It was shown that such paths are a sequence of circles C of maximum curvature and segment lines S. Furthermore, it was proved that minimum length paths are either of type CCC (Curve-Curve-Curve) or CSC (Curve-Segment-Curve). Later, using optimal control theory and some geometric arguments (Boissonnat et al. [1991]), the same result was obtained. An example of each type of paths is presented in Fig. 5. Thus, the length of the optimal path can be obtained analytically.



(i) CCC type (ii) CSC types

Fig. 5. Dubins' paths

The *Dubins' metric* used in the `nearest_neighbour` function is based on the Dubins' work as follows. Given a node $\mathbf{x}(t) \in G$, the maximum curvature $c_{\max}(t, \mathbf{x})$ is computed at $\mathbf{x}(t)$. Then, the minimum length path from $\mathbf{x}$ to $\mathbf{x}_{\mathrm{rand}}$ is computed using Dubins' results described previously. This procedure is repeated for all $\mathbf{x}(t) \in G$ and the state $\mathbf{x}_{\mathrm{near}}$ with the shortest optimal path to $\mathbf{x}_{\mathrm{rand}}$ is returned.

3.3 `select_input`

The `select_input` function returns a control input $\mathbf{a}_{\mathrm{RRT}}^{\mathrm{v}}$, using $\mathbf{x}_{\mathrm{near}}$ and $\mathbf{x}_{\mathrm{rand}}$. $\mathbf{a}_{\mathrm{RRT}}^{\mathrm{v}}$ can be chosen randomly or using a specific criterion. In this paper, to rapidly explore the state space, the control input $\mathbf{a}_{\mathrm{RRT}}^{\mathrm{v}}$ is chosen in order to create a new state $\mathbf{x}_{\mathrm{new}}$ as close as possible to $\mathbf{x}_{\mathrm{rand}}$. A Proportional Navigation (PN) guidance law is proposed in this paper:

$$\mathbf{a}_{\mathrm{RRT}}^{\mathrm{v}} = \left(\frac{N}{t_{\mathrm{go}}^2} (\boldsymbol{\xi}_{\mathrm{rand}} - \boldsymbol{\xi}_{\mathrm{near}} - \mathbf{v}_{\mathrm{near}} t_{\mathrm{go}}) \cdot \mathbf{e}_3^{\mathrm{v}} \right) \mathbf{e}_3^{\mathrm{v}} \tag{14}$$

where N is a constant gain and t_{go} is the estimated time-to-go.

To satisfy the missile constraints, the control input $\mathbf{a}_{\mathrm{c}}^{\mathrm{v}}$ is obtained after saturating $\mathbf{a}_{\mathrm{RRT}}^{\mathrm{v}}$ so that $\mathbf{a}_{\mathrm{c}}^{\mathrm{v}} \in \mathbb{U}(t, \mathbf{x})$.

The `new_state` function applies this control input during the time increment Δt to obtain $\mathbf{x}_{\mathrm{new}}$.

4. RESULTS AND ANALYSIS

A single-stage missile is used in this section. The boost phase of the missile lasts 20s. During the whole flight, the missile is only controlled aerodynamically using tail fins, the missile speed at t_{boost} is approximately 2000m/s. The control loop is assumed to be perfect ($\forall t$, $\mathbf{a}^{\mathrm{v}} = \mathbf{a}_{\mathrm{c}}^{\mathrm{v}}$).

In this section, two scenarios are analyzed. For both, the time increment $\Delta t = 2s$, the initial position $\boldsymbol{\xi}_{\mathrm{init}} = (0, 0)$, the missile is launched vertically, $\mathbb{X} = [-20\mathrm{km}, 20\mathrm{km}] \times [0\mathrm{km}, 30\mathrm{km}]$,

$$\mathbb{X}_{\mathrm{goal}} = \{\mathbf{x} : \|\boldsymbol{\xi} - \boldsymbol{\xi}_{\mathrm{pip}}\| < 500m, \|\mathbf{v}\| \geqslant 500m/s, \angle(\mathbf{v}, -\mathbf{v}_{\mathrm{pip}}) < \pi/8\},$$

$\boldsymbol{\xi}_{\mathrm{pip}} = (15\mathrm{km}, 15\mathrm{km})$ in scenario 1, $\boldsymbol{\xi}_{\mathrm{pip}} = (10\mathrm{km}, 25\mathrm{km})$ in scenario 2 and $\mathbf{v}_{\mathrm{pip}}$ is parallel to the ground.

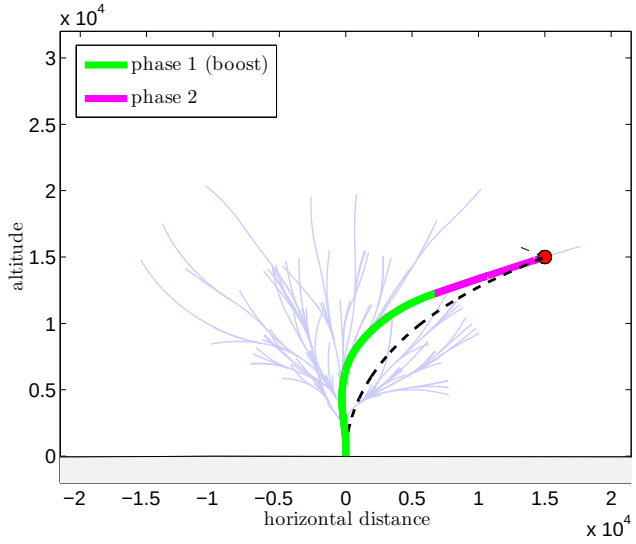


Fig. 6. Scenario 1 - Trajectories

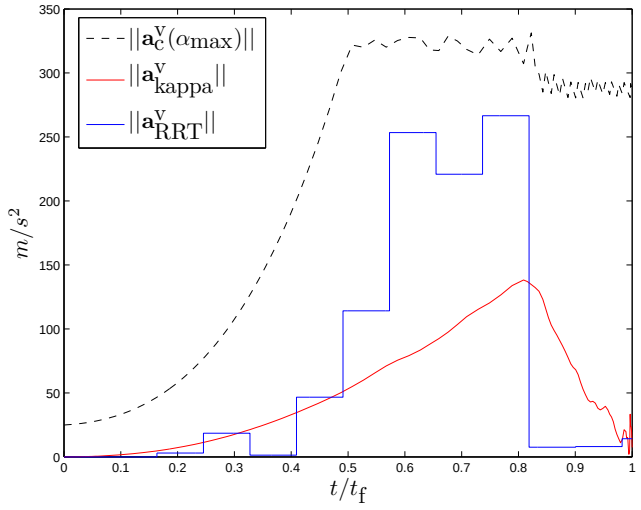


Fig. 7. Scenario 1 - Lateral accelerations along the solution trajectories

In both scenarios, trajectories generated by our RRT-based guidance law are compared to those obtained using kappa guidance (Lin [1991]). Control input returned by kappa guidance before saturation is denoted $\mathbf{a}_{\text{kappa}}^v$.

A bias toward the goal is introduced in the algorithm to reduce the number of needed nodes to reach $\mathbb{X}_{\text{goal}}$. This bias, called RRT-*GoalBias* (LaValle and Kuffner [2001]), consists in choosing $\mathbf{x}_{\text{pip}}$ as $\mathbf{x}_{\text{rand}}$ in the `random_state` function with a probability p. In this paper, $p = 0.01$.

On the following figures, the dashed curve is the trajectory obtained using the kappa guidance, the tree G is represented in blue, $\mathbb{X}_{\text{goal}}$ is represented as a red circle with two dashed line segments as in figure 3. The boost phase of the generated trajectory between $\mathbf{x}_{\text{init}}$ and $\mathbb{X}_{\text{goal}}$ is in green, the second phase in pink.

Figure 6 illustrates the trajectories obtained for scenario 1. Figure 7 illustrates the lateral accelerations $\mathbf{a}_{\text{kappa}}^v$ and $\mathbf{a}_{\text{RRT}}^v$ respectively returned by kappa guidance and computed in `select_input` function. $\mathbf{a}_c^v(\alpha_{\max})$ is the maximum lateral acceleration tolerated by the missile. Since the control inputs $\mathbf{a}_{\text{kappa}}^v$ and $\mathbf{a}_{\text{RRT}}^v$ for both guidance methods are always below the maximum values tolerated by the missile, both trajectories reach $\mathbb{X}_{\text{goal}}$ easily.

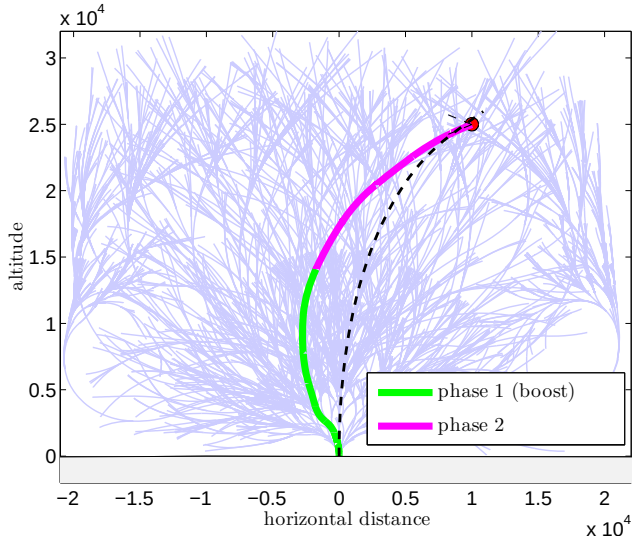


Fig. 8. Scenario 2 - Trajectories

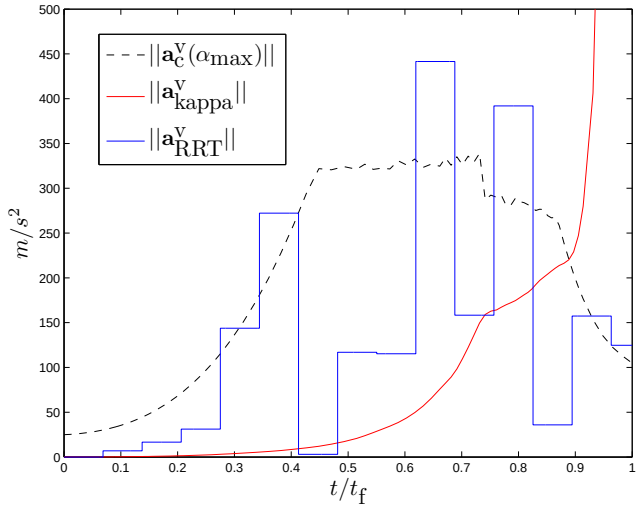


Fig. 9. Scenario 2 - Lateral accelerations along the solution trajectories

The number of generated nodes to find this solution is 1115. The final speeds are respectively 1717m/s and 1807m/s for kappa guidance and RRT-based guidance.

Scenario 1 illustrates a case where a solution can be found using a classical midcourse guidance law. In this case, our RRT-based guidance law is also able to provide a trajectory with similar performances.

Figure 8 illustrates the trajectories obtained for scenario 2. This case differs from scenario 1 since the target is higher and closer in terms of horizontal distance. The difficulty for an aerodynamically controlled missile lies in the low maneuverability at high altitude due to the low density of air. Therefore, it is hard to satisfy the constraint $\angle(\mathbf{v}(t_f), -\mathbf{v}_{\text{pip}}) < \phi_f$ where $\mathbf{v}_{\text{pip}}$ is parallel to the ground.

Kappa guidance trajectory (dashed curve) cannot satisfy this constraint since $\angle(\mathbf{v}(t_f), -\mathbf{v}_{\text{pip}}) = 0.62 > \pi/8$ rad. It does not anticipate the future lack of maneuverability and sends low control inputs until $t/t_f = 0.5$ (Fig. 9). Thus, at the end of the trajectory ($t/t_f > 0.9$), the guidance law tries to respect the aspect angle by sending huge control inputs to the controller. As the maneuvering capabilities are low, the missile cannot perform such lateral accelerations and fails to reach $\mathbb{X}_{\text{goal}}$.

On the contrary, as the RRT-based algorithm anticipates the loss of maneuverability near the PIP, the generated trajectory performs a back-turn. Indeed, it moves away from the line-of-sight at the beginning in order to reduce the curvature of the trajectory when approaching $\mathbb{X}_{\text{goal}}$. Thus, the needed lateral acceleration at the end of the trajectory remains lower than the maneuvering capabilities at these altitudes (Fig. 9). Since this problem is harder to solve than the previous one, the number of iterations increases and reaches 5953. Furthermore, $\|\mathbf{v}(t_f)\| = 1552\text{m/s} > v_{\min}$ is verified.

This scenario illustrates the effectiveness of our RRT-based guidance law compared to classical midcourse guidance laws, based on its capability to anticipate future flight conditions.

5. CONCLUSION AND PERSPECTIVES

A new and novel approach of missile guidance is introduced in this paper. This guidance scheme combines a sampling-based RRT path planner, Dubins' curves whose lengths are used as metric function, and Proportional Navigation guidance law to chose the appropriate control inputs. This method solves critical problems encountered in midcourse guidance that cannot be easily solved using classical guidance laws as it anticipates future flight conditions.

Results are promising and some possible extensions could be studied in future work. First, the optimality of the generated trajectories has to be studied. For example, it would be interesting to maximize the final velocity or to minimize the flight time. This could be performed by improving both metric function and control input selection. Next, the missile model could be more realistic including delays in the control loop. Then, as the generated trajectories currently lie in a 2-dimensional plan, the algorithm has to be generalized to a 3-dimensional space. Finally, the computing time could be improved by, for example, reducing $\mathbb{X}_{\text{free}}$ using preprocessing.

REFERENCES

J. D. Boissonnat, A. Cérézo, and J. Leblond. Shortest paths of bounded curvature in the plane. Technical report, Institut National de Recherche en Informatique et en Automatique, 1991.

V. H. L. Cheng and N. K. Gupta. Advanced midcourse guidance for air-to-air missiles. *Journal of Guidance, Control, and Dynamics*, 9(2):135–142, 1986.

E. Devaud, H. Siguerdidjane, and S. Font. Some control strategies for a high-angle-of-attack missile autopilot. *Control Engineering Practice*, 8(8):885–892, 2000.

J. J. Dougherty and J. L. Speyer. Near-optimal guidance law for ballistic missile interception. *Journal of Guidance, Control, and Dynamics*, 20(2):355–362, 1997.

L. E. Dubins. On curves of minimal length with a constraint on average curvature and with presribed initial and terminal position and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

F. Imado and T. Kuroda. Optimal guidance system against a hypersonic targets. In *AIAA Guidance, Navigation and Control Conference*, 1992.

F. Imado, T. Kuroda, and S. Miwa. Optimal midcourse guidance for medium-range air-to-air missiles. *Journal of Guidance, Control, and Dynamics*, 13(4):603–608, 1990.

L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12:566–580, 1996.

S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

C. F. Lin. *Modern Navigation Guidance and Control Processing*. Prentice-Hall, Inc., 1991.

C. F. Lin and L. L. Tsai. Analytical solution of optimal trajectory-shaping guidance. *Journal of Guidance, Control, and Dynamics*, 10(1):61–66, 1987.

C. L. Lin and Y. Y. Chen. Design of fuzzy logic guidance law against high-speed target. *Journal of Guidance, Control, and Dynamics*, 23(1):17–25, 2000.

C. L. Lin, H. Z. Hung, Y. Y. Chen, and B. S. Chen. Development of an integrated fuzzy-logic-based missile guidance law against high speed target. *IEEE Transactions on Fuzzy Systems*, 12(2):157 – 169, april 2004. ISSN 1063-6706. doi: 10.1109/TFUZZ.2004.825069.

P. K. Menon and M. M. Briggs. Near-optimal midcourse guidance for air-to-air missiles. *Journal of Guidance, Control, and Dynamics*, 13(4):596–602, 1990.

B. Newman. Strategic intercept midcourse guidance using modified zero effort miss steering. *Journal of Guidance, Control, and Dynamics*, 19(1):107–112, 1996.

R. Pepy, A. Lambert, and H. Mounier. Reducing navigation errors by planning with realistic vehicle model. In *IEEE Intelligent Vehicles Symposium*, pages 300–307, 2006.

M. Robb, B.A. White, A. Tsourdos, and D. Rulloda. Reachability guidance: a novel concept to improve midcourse guidance. In *American Control Conference*, pages 339 – 345, 2005.

E. J. Song and M. J. Tahk. Real-time neural-network midcourse guidance. *Control Engineering Practice*, 9 (10):1145 – 1154, 2001. ISSN 0967-0661.

E. J. Song and M. J. Tahk. Three-dimensional midcourse guidance using neural networks for interception of ballistic targets. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2):404 –414, 2002. ISSN 0018-9251.

G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *Journal fur die Reine und Angewandte Mathematik*, 133:97–178, 1907.