HAL
open science

Missile trajectory shaping using sampling-based path planning

Pawit Pharpatara, Romain Pepy, Bruno Hérissé, Yasmina Bestaoui

To cite this version:

Pawit Pharpatara, Romain Pepy, Bruno Hérissé, Yasmina Bestaoui. Missile trajectory shaping using sampling-based path planning. 26th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013), Nov 2013, Tokyo, Japan. pp.2533–2538, 10.1109/IROS.2013.6696713 . hal-00860192

HAL Id: hal-00860192

https://hal.science/hal-00860192v1

Submitted on 10 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Missile trajectory shaping using sampling-based path planning

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

***Abstract*— This paper presents missile guidance as a complex robotic problem: a hybrid non-linear system moving in a heterogeneous environment. The proposed solution to this problem combines a sampling-based path planner, Dubins' curves and a locally-optimal guidance law. This algorithm aims to find feasible trajectories that anticipate future flight conditions, especially the loss of manoeuverability at high altitude. Simulated results demonstrate the substantial performance improvements over classical midcourse guidance laws and the benefits of using such methods, well-known in robotics, in the missile guidance field of research.**

I. Introduction

The purpose of this paper is to define a new guidance algorithm for an endo-atmospheric interceptor missile which aims to destroy a flying target.

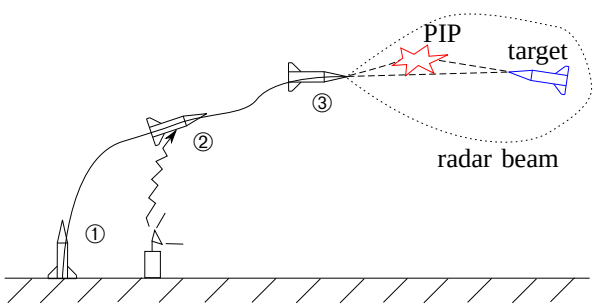


Fig. 1. Guidance of an interceptor

The guidance of an interceptor toward a target is illustrated in figure 1. It is divided into two different phases. The first one, called midcourse guidance, starts at the interceptor launch (see ①). During this stage (see ②), the missile only uses its proprioceptive sensors (most of the time an inertial measurement unit) to locate itself. The target trajectory is predicted by specific algorithms using its current state (position and velocity) returned by a radar. The position where the collision is supposed to happen is known as the Predicted Intercept Point (PIP). During midcourse guidance, this PIP is sent from the radar to the interceptor using a communication channel. These data are often refreshed during the flight. The aim of the midcourse guidance algorithm is to move the interceptor toward the PIP. Midcourse guidance stops when the interceptor approaches the PIP, *i.e.* when the radar sensor embedded on the interceptor is able to detect the target. Then, the terminal homing phase can start (see ③). Many works solve the terminal guidance problem with closed-loop guidance laws such as Proportional Navigation [18]. In this paper, we focus on the midcourse guidance problem.

Finding a path from the initial position to the PIP is a guidance problem with many constraints : the interceptor missile is a hybrid system (propulsive stage and non-propulsive stage), its speed and maneuvering capabilities vary during the flight. It flies in a heterogeneous environment and has to reach the PIP with a certain angle and enough speed to ensure that the embedded radar can find the target and that the lethal system can destroy it.

Many closed-loop optimal midcourse guidance laws were proposed in the past using singular perturbation theory [2], [14], [4], linear quadratic regulators [8], [7], analytical methods [13] or modified proportional guidance [15]. The kappa guidance detailed in [12] 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, this guidance law relies on some restrictive approximations. Moreover, control limitations (saturations) are difficult to satisfy. Likewise, the kappa guidance is not suitable for a surface-to-air missile dedicated to high altitude interception since variations of air density are large from low to high altitudes. For such complex systems and missions, the optimal problem needs to be considered globally.

Many studies in the robotic field, especially in path planning and control theory aims to find trajectories in complex environments. For example, sampling-based path planning methods, such as Rapidly-exploring Random Trees (RRT) [10] or Probabilistic Roadmap Methods (PRM) [9], 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 [11]. The main advantage of these techniques is that even a complex system can be considered without a need of approximations [16]. The idea of this paper is to use such a sampling-based method to plan a path for interceptor missiles.

The proposed method in this paper is based on Rapidly-exploring Random Trees for trajectory shaping during the midcourse phase. A metric function based on Dubins' curves is used to compute the distance between two states. A kappa guidance law is used to steer the missile. Simulation results are obtained for a missile using a single propulsive stage. Terminal constraints are defined with respect to the Predicted Intercept Point (PIP) and to the interceptor capabilities for the terminal guidance stage. The obtained results demonstrate

P. Pharpatara, R. Pepy and B. Hérissé are with Onera - The French Aerospace Lab, Palaiseau, France (email: pawit.pharpatara@onera.fr, romain.pepy@onera.fr, bruno.herisse@onera.fr).

Y. Bestaoui is with IBISC, Université d'Evry-Val-d'Essonne, Evry, France (e-mail: Yasmina.Bestaoui@ufrst.univ-evry.fr).

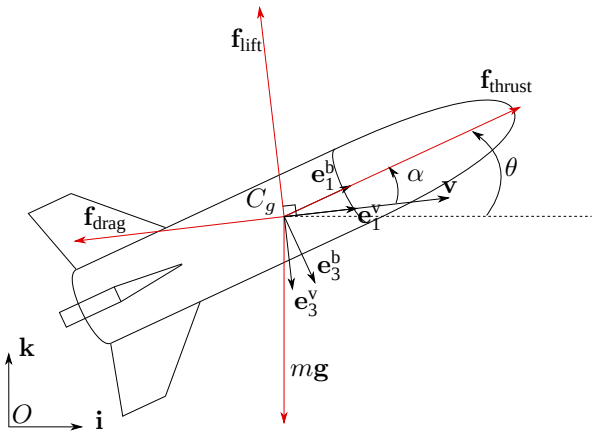


Fig. 2. Definition of reference frames

the potential of path planning algorithms in missile guidance especially in high altitude cases where classical closed-loop laws cannot find a proper solution.

This paper is divided into four parts. First, the system and environment modelings are presented in section II. Then, section III introduces the RRT path planner and its modifications for missile guidance. Then, some simulated results are presented (section IV). Finally, some concluding remarks are made in the last section.

II. Problem statement

A. System modeling

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 few effect on the missile and is neglected in this paper. Three frames (Fig. 2) 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 basis vectors $(\mathbf{i}, \mathbf{k})$; 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. 2). 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. 2), the axial force coefficient C_{A} and the normal force coefficient C_{N}. These can be expressed as

$$\begin{aligned} C_{\text{D}} &= C_{\text{A}} \cos\alpha + C_{\text{N}} \sin\alpha \\ C_{\text{L}} &= C_{\text{N}} \cos\alpha - C_{\text{A}} \sin\alpha \end{aligned} \tag{6}$$

C_{N} can be written as $C_{\text{N}} = C_{\text{N}\alpha}\alpha$ where $C_{\text{N}\alpha}$ is the normal force coefficient curve slope.

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

$$\mathbf{f}_{\text{thrust}} = (\text{I}_{\text{sp}}(\text{Vac}) g_0 q_{\text{t}} - A_{\text{e}} p_0)\, \mathbf{e}_1^{\text{b}}, \tag{7}$$

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 $\mathbf{a}^{\text{v}}$ [3]. In the following, $\mathbf{a}_{\text{c}}^{\text{v}}$ denotes the setpoint of this control loop.

B. Environment modeling

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. As we consider a missile with only aerodynamic flight controls, the maneuvering capabilities are linked to the density of air (5) and approach zero at 35 km.

C. Problem formulation

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

$$\dot{\mathbf{x}} = f(t, \mathbf{x}, \mathbf{a}_{\text{c}}^{\text{v}}), \tag{8}$$

where f is defined in section II-A.

$\mathbb{X} \subseteq \mathbb{R}^4$ is the state space. It is divided into two subsets. Let $\mathbb{X}_{\text{free}}$ be the set of admissible states and let $\mathbb{X}_{\text{obs}} = \mathbb{X} \setminus$

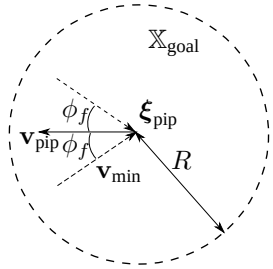


Fig. 3. $\mathbb{X}_{\text{goal}}$

$\mathbb{X}_{\text{free}}$ be the *obstacle region i.e.* the set of non-admissible states. In this paper, $\mathbb{X}_{\text{free}}$ is defined as

$$\mathbb{X}_{\text{free}} = \{\mathbf{x} : \text{altitude}(\boldsymbol{\xi}) > 0, \|\mathbf{v}(t > t_{\text{boost}})\| > v_{\text{min}}\}, \quad (9)$$

where v_{min} is the minimum tolerated interceptor speed at the intercept 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 intercept 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

$$\mathbb{X}_{\text{goal}} = \{\mathbf{x} : \|\boldsymbol{\xi} - \boldsymbol{\xi}_{\text{pip}}\| < R, \|\mathbf{v}\| \geqslant v_{\text{min}}, \angle(\mathbf{v}, -\mathbf{v}_{\text{pip}}) < \phi_f\} \quad (10)$$

where R is the radius of a sphere centered at $\boldsymbol{\xi}_{\text{pip}}$ and ϕ_f is an angle related to the maximum capabilities of the terminal guidance system. 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 $\mathbf{x}$:

$$\mathbb{U}(t, \mathbf{x}) = \{\mathbf{a}_{\text{c}}^{\text{v}} : \alpha_c \leqslant \alpha_{\text{max}}(t, \mathbf{x})\} \quad (11)$$

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 $\mathbf{x}$ and is defined by

$$\alpha_{\text{max}}(t, \mathbf{x}) = \min\left(\alpha_{\text{max}}^{\text{stb}}(t, \mathbf{x}), \alpha_{\text{max}}^{\text{struct}}(t, \mathbf{x})\right). \quad (12)$$

$\alpha_{\text{max}}^{\text{stb}}(t, \mathbf{x})$ 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 $\mathbf{x}(t_f) \in \mathbb{X}_{\text{goal}}$.

D. A challenging problem

To sum up this challenging problem:

- The system described in section II-A is hybrid with a propulsive phase and a non-propulsive phase;
- The speed of the system increases during the propulsive phase and then decreases due to drag forces;
- The environment is heterogeneous with density of air that decreases exponentially with altitude;
- Maneuvering capabilities of the system are linked to both speed and altitude and tend to zero when the altitude approaches 30 km;
- The goal area must be reached with a minimum speed in order to destroy the target.

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**

III. Motion planning framework

A. Rapidly-exploring Random Trees

Rapidly-exploring Random Trees (RRT) [11] 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 [17]. 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 III-B). 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 III-C). 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). On 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.* $\mathbf{x}_{\text{new}} \in \mathbb{X}_{\text{goal}}$.

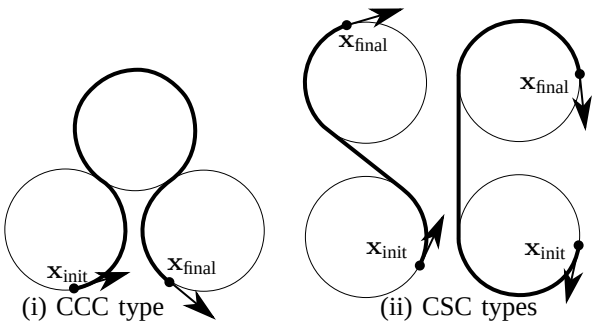


Fig. 4. Dubins' paths

B. `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 Dubins' results [5], [1] 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 and ignore the dynamics of the norm v of the velocity vector $\mathbf{v}$. Considering curvilinear $s = \int_0^t v(u)\,du$ instead of the time, the system can be rewritten as follows

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

where $c = \frac{a^{\text{v}}}{v^2}$ is the curvature of the vehicle trajectory and $\mathbf{a}^{\text{v}} = a^{\text{v}}\mathbf{e}_3^{\text{v}}$. Assume that the control of the lateral acceleration is perfect, that is $\forall t,\ \mathbf{a}^{\text{v}} = \mathbf{a}_{\text{c}}^{\text{v}} \in \mathbb{U}(t, \mathbf{x})$. Since $\mathbf{a}^{\text{v}} \in \mathbb{U}(t, \mathbf{x})$, the maximum curvature $c_{\max}(t, \mathbf{x})$ depends on t and $\mathbf{x}$ along the trajectory. The system (13) is similar to the system considered in [5] often called the Dubins' car. This is a non-holonomic vehicle moving forward with a constant velocity and capable of maneuvering within a bounded curvature.

In [5], $c_{\max}$ was considered as a constant along the trajectory and minimum length paths from an initial state $(x_{\text{init}}, z_{\text{init}}, \gamma_{\text{init}})$ to a final state $(x_{\text{final}}, z_{\text{final}}, \gamma_{\text{final}})$ 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 [1], the same result was obtained. An example of each type of path is presented in Fig. 4. Thus, the length of the optimal path can be obtained analytically.

The *Dubins' metric* used in the `nearest_neighbour` function is based on 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 $\mathbf{x}_{\text{rand}}$ is computed using Dubins' results described previously. This procedure is repeated for all $\mathbf{x}(t) \in G$ and the state $\mathbf{x}_{\text{near}}$ with the shortest optimal path to $\mathbf{x}_{\text{rand}}$ is returned.

C. `select_input`

The `select_input` function uses $\mathbf{x}_{\text{near}}$ and $\mathbf{x}_{\text{rand}}$ and returns a control input $\mathbf{a}_{\text{RRT}}^{\text{v}}$. $\mathbf{a}_{\text{RRT}}^{\text{v}}$ can be chosen randomly or using a specific criterion. A Kappa guidance law is proposed in this paper. This guidance law returns the control input

$$\mathbf{a}_{\text{RRT}}^{\text{v}} = ((\mathbf{a}_1 + \mathbf{a}_2) \cdot \mathbf{e}_3^{\text{v}})\mathbf{e}_3^{\text{v}}, \tag{14}$$

with

$$\begin{aligned} \mathbf{a}_1 &= \frac{K_1}{t_{\text{go}}}(\mathbf{v}_{\text{rand}} - \mathbf{v}_{\text{near}}) \\ \mathbf{a}_2 &= \frac{K_2}{t_{\text{go}}^2}(\boldsymbol{\xi}_{\text{rand}} - \boldsymbol{\xi}_{\text{near}} - \mathbf{v}_{\text{near}} t_{\text{go}}) \end{aligned} \tag{15}$$

choosing $||\mathbf{v}_{\text{rand}}|| = ||\mathbf{v}_{\text{near}}||$, K_1 and K_2 are the gains and t_{go} is the estimated time-to-go: $t_{\text{go}} = t_f - t$. The estimation of t_{go} is a difficult problem as it theoretically necessitates to estimate the entire flight time t_f to the target. There exists many techniques to provide an approximate of t_{go} [12].

The second term $\mathbf{a}_2$ is the proportional navigation term which closes out the heading error to the PIP while the first term $\mathbf{a}_1$ is the shaping term which rotates the flight path angle so that γ converges to γ_f.

The purpose of the optimal Kappa guidance law is to maximize the missile terminal speed v_f (equivalent to minimizing the total loss in kinetic energy) while satisfying the boundary conditions: no heading error, required terminal flight path angle $\gamma = \gamma_f$ and the distance between the missile and the target $D = ||\boldsymbol{\xi}_{\text{PIP}} - \boldsymbol{\xi}_{\text{near}}|| = 0$ at the PIP. Gains K_1 and K_2 satisfying these constraints can be approximated by [12]

$$\begin{aligned} K_1 &= \frac{2F^2D^2 - FD(e^{FD} - e^{-FD})}{e^{FD}(FD-2) - e^{-FD}(FD+2) + 4} \\ K_2 &= \frac{F^2D^2(e^{FD} + e^{-FD} - 2)}{e^{FD}(FD-2) - e^{-FD}(FD+2) + 4} \\ F^2 &= \frac{QSC_{\text{A}}(||\mathbf{f}_{\text{thrust}}|| + QS(C_{\text{N}\alpha} - C_{\text{A}}))^2}{m^2v^4(||\mathbf{f}_{\text{thrust}}|| + 2QSC_{\text{N}\alpha} - QSC_{\text{A}})} \end{aligned} \tag{16}$$

where $Q = \rho v^2/2$ is the dynamic pressure.

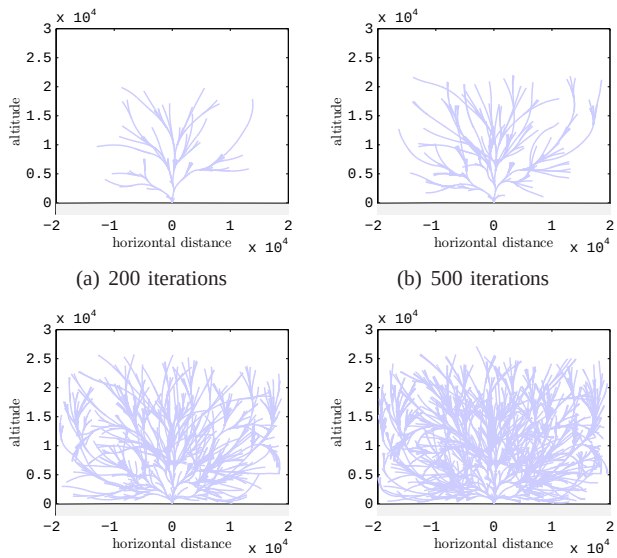
D. `new_state`

First, the `new_state` function saturates $\mathbf{a}_{\text{RRT}}^{\text{v}}$ to obtain the admissible control input $\mathbf{a}_{\text{c}}^{\text{v}} \in \mathbb{U}(t, \mathbf{x})$ that satisfies the missile constraints. Then, it applies this control input during the time increment Δt to obtain $\mathbf{x}_{\text{new}}$.

Figure 5 illustrates RRT expansion using these functions.

IV. Results and analysis

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



(a) 200 iterations (b) 500 iterations

(c) 2000 iterations (d) 4000 iterations

Fig. 5. RRT expansion

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

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

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

In both scenarios, trajectories generated by our RRT-based guidance law are compared to those obtained using kappa guidance with the desired flight path angle $\gamma_f = \pi - \phi_f$ and $\phi_f = \pi/8$. Control input returned by kappa guidance before saturation is denoted $\mathbf{a}^{\text{c}}_{\text{kappa}}$.

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* [11], consists in choosing $\mathbf{x}_{\text{pip}}$ as $\mathbf{x}_{\text{rand}}$ in the `random_state` function with a probability p. In this section, $p = 0.01$.

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 is in pink.

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

The number of generated nodes to find this solution is 391. The final speeds for kappa guidance and RRT-based guidance

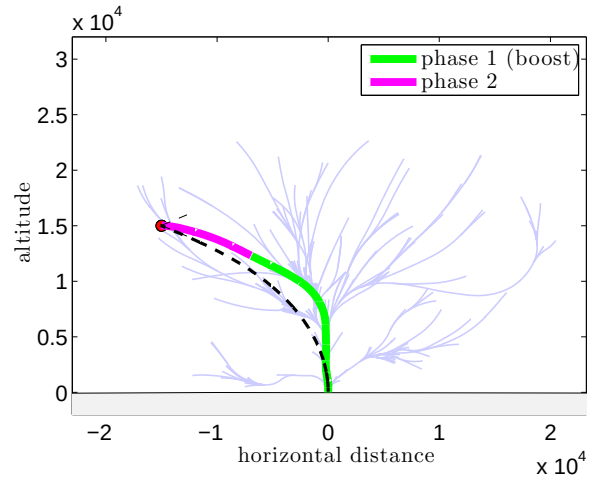


Fig. 6. Scenario 1 - Trajectories

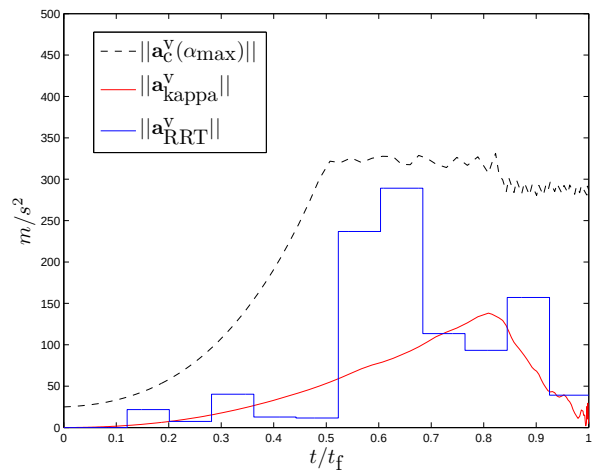


Fig. 7. Scenario 1 - Lateral accelerations along the computed trajectories

are respectively 1717m/s and 1586m/s. As a result, both guidance laws provide trajectories with similar performance.

Figure 8 illustrates the obtained trajectories for scenario 2. Since the target is higher and closer in term of horizontal distance, an aerodynamically controlled missile encounters difficulty to reach $\mathbb{X}_{\text{goal}}$. The difficulty lies in the low maneuverability at high altitude due to the low density of air. Therefore, the constraint $\angle(\mathbf{v}(t_f), -\mathbf{v}_{\text{pip}}) < \phi_f$ is hard to satisfy when $\mathbf{v}_{\text{pip}}$ is parallel to the ground.

The trajectory generated by Kappa guidance (dashed curve) cannot satisfy this constraint since $\angle(\mathbf{v}(t_f), -\mathbf{v}_{\text{pip}}) = 0.62$ rad $> \pi/8$. 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 satisfying the constraint $\angle(\mathbf{v}, -\mathbf{v}_{\text{pip}}) < \phi_f$ by sending huge control inputs to the controller. Since its 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 at high altitude near the PIP, a

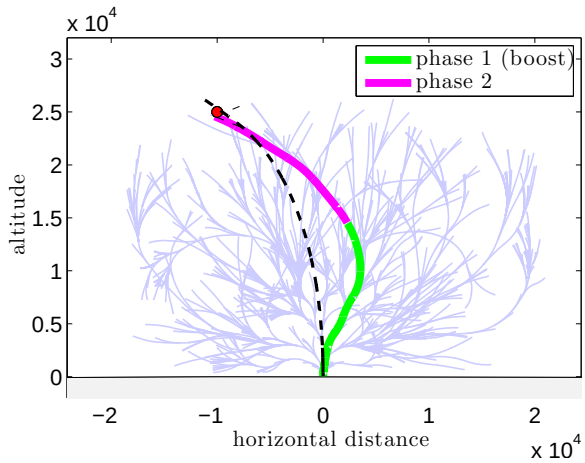


Fig. 8. Scenario 2 - Trajectories

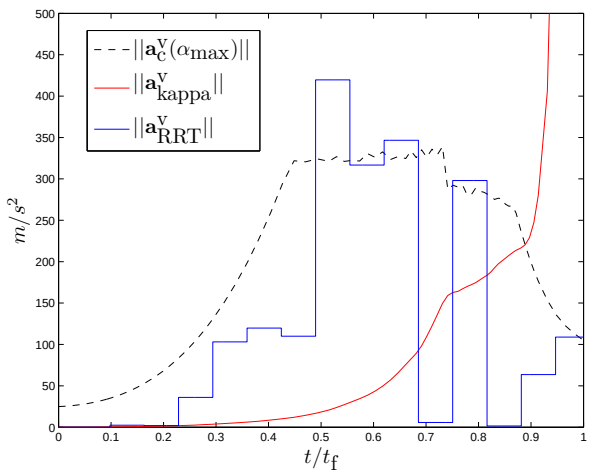


Fig. 9. Scenario 2 - Lateral accelerations along the computed trajectories

back-turn is performed by its generated trajectory. 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 1578 for this trajectory. Furthermore, $\|\mathbf{v}(t_f)\| = 1309\text{m/s} > v_{\min}$ 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.

V. Conclusion and Perspectives

This new and novel guidance law combines a sampling-based RRT path planner, Dubins' curves whose lengths are used as a metric function, and a Kappa guidance law to choose the appropriate control inputs. The critical midcourse guidance problems that cannot be solved easily using classical guidance laws can be solved by this method as it anticipates future flight conditions.

Results are promising and some possible extensions could be studied in future work. The study of the optimality of the generated trajectories is our first priority: it could be interesting to maximize the final velocity or to minimize the flight time. This could be done by improving both metric function and control input selection. Another solution would consist in preprocessing the state space as it could also reduce the computing time, for example using a simplified missile model [6]. Furthermore, a more realistic missile model should be used by including delays in the control loop. Finally, as the generated trajectories currently lie in a 2-dimensional plan, the algorithm has to be generalized to a 3-dimensional space.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] B. Hérissé and R. Pepy. Shortest paths for the Dubins' vehicle in heterogeneous environments. In *Proceedings of the IEEE Conference on Decision and Control*, Florence, Italy, 2013. To appear.

[7] F. Imado and T. Kuroda. Optimal guidance system against a hypersonic targets. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 1992.

[8] 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.

[9] 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.

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

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

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

[13] 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.

[14] 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.

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

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

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

[18] P. Zarchan. *Tactical and Strategic Missile Guidance*, volume 157 of *Progress in Astronautics and Aeronautics*. America Institute of Aeronautics and Astronautics, Inc. (AIAA), second edition, 1994.