International Journal on Software Tools for Technology Transfer (2022) 24:371–394
https://doi.org/10.1007/s10009-022-00654-2

GENERAL

FMICS 2019/2020

Envelopes and waves: safe multivehicle collision avoidance for horizontal non-deterministic turns

Yanni Kouskoulas[1] · T. J. Machado[2] · Daniel Genin[3] · Aurora Schmidt[3] · Ivan Papusha[3] · Joshua Brulé[3]

Accepted: 8 March 2022 / Published online: 2 May 2022

Abstract
We present an approach to analyze the safety of asynchronous, independent, non-deterministic, turn-to-bearing horizontal maneuvers for two vehicles. Future turn rates, final bearings, and continuously varying ground speeds throughout the encounter are unknown but restricted to known ranges. We develop a library of formal proofs about turning kinematics and apply the library to create a formally verified timing computation. Additionally, we create a technique that evaluates future collision possibilities that is based on waves of position possibilities and relies on the timing computation. The result either determines that the encounter will be collision-free, or computes a safe overapproximation for when and where collisions may occur.

Keywords Formal methods · Collision avoidance · Proof assistant · Reachability · Non-deterministic horizontal turns · Coq

1 Introduction

Autonomous and semi-autonomous systems that control ground vehicles, boats, and aircraft all need to reason about horizontal turns in order to create plans for future motion that meet system objectives.

We are specifically motivated by aircraft collision avoidance maneuvers that combine vertical and horizontal advice to ensure multi-aircraft encounters are safely separated. These maneuvers advise aircraft to turn at the same time they change vertical velocity—the objective being to keep the aircraft separated in altitude during periods when their positions might coincide horizontally. This requires correctly computing the time interval that describes when in the future both aircraft might come into horizontal conflict.

This paper develops a formalization of *non-deterministic* turn-to-bearing motion, where a vehicle turns following a circular arc until reaching a certain bearing, and then follows a straight path thereafter. Turn-to-bearing motion is the building block for Dubins trajectories used in many different techniques in the literature (see Sect. 2), but here we consider that the parameters that describe our future path are non-deterministic and uncertain at the beginning of the turn.

The formalization is embodied in a library of proofs that are detailed descriptions of these kinematics and are machine-checked to guarantee correctness. Each theorem in this paper corresponds to a proof in the formalization.[1] We believe that the library can serve as a foundation for formal reasoning about horizontal turns in the Coq proof assistant, supporting the development of insight and correct reasoning for a wide variety of path planning and collision avoidance algorithms. Furthermore, we hope that it helps guarantee a high level of correctness and robustness for robotic systems' horizontal motion, and that it provides the basis for certifica-

This research was partially funded under the sponsorship of the Federal Aviation Administration Traffic Alert and Collision Avoidance System (TCAS) Program Office (PO) AJM-42 under Contract Number DTFAWA-11-C-00074 as well as internal funds from the Johns Hopkins University Applied Physics Laboratory.

✉ Yanni Kouskoulas
yxkous@gmail.com

T. J. Machado
tjm@nmsu.edu

Daniel Genin
daniel.genin@jhuapl.edu

[1] Seattle, WA, USA

[2] Las Cruces, NM, USA

[3] Johns Hopkins University Applied Physics Laboratory, 11100 Johns Hopkins Rd, Laurel, MD 20723, USA

[1] Coq proofs are at https://bitbucket.org/ykouskoulas/ottb-foundation-proofs

tion artifacts (i.e., proofs) that can be used to establish system algorithm and software correctness.

We develop an approach to evaluate collision possibilities during an encounter. We apply the library to develop and formally verify an exact pointwise timing computation, use it to create a sound approximation of the timing over an area, and quantify the timing computation over the reachable area of future motion.

This paper is an extension of [12], which presents the novel contributions of the original work, but also new ideas. The contributions of the original work are: the development of a Coq library for reasoning about non-deterministic Dubins-style paths; an additional Coq library defining a variety of two-argument arctangent functions with different branch cuts that are each sensitive to the quadrant and sign of their arguments; a new expression for computing the appropriate angle necessary for connecting Dubins paths to a destination waypoint;[2] and formally verified expressions of the timing constraints of uncertain turn-to-bearing motion. The contributions that are unique to this extension are: development of novel, sound approximations for the location of a vehicle within the reachable area using waves as moving boundaries; and a simple, efficient, piecewise approach to calculating the range of possible collision times between two vehicles each of which followings turn-to-bearing kinematics. It also improves the presentation of conjectures from the original paper that enhance ease of comprehension for readers who want to use the equations in practical applications.

The rest of this paper is organized as follows: Sect. 2 considers prior work in formalizing horizontal motion and analyzing potential collisions; Sect. 3 describes the details of our library and how we formalized non-deterministic, turn-to-bearing paths in Coq; Sect. 4 applies the library to derive exact, formally verified solutions for the timing of intersecting turns at a *given* point. The rest of the sections present new ideas that were not in the original paper: Sect. 5 develops a strategy for quantifying the collision timing computations over *all* points in the conflict area, providing a bookkeeping framework for managing the solution to different pieces of the exact timing equations separately; Sect. 6 develops formally verified approximations to the timing equations that are appropriate for quantification within their polygonal region; Sect. 7 presents a method for quantifying the timing equations over each piece of the domain, devising a sound solution for collision timing between two turning vehicles; and finally Sect. 8 discusses our conclusions and future work.

[2] There exist alternate expressions for this angle, but to our knowledge, the formulation in this paper is new.

2 Literature review

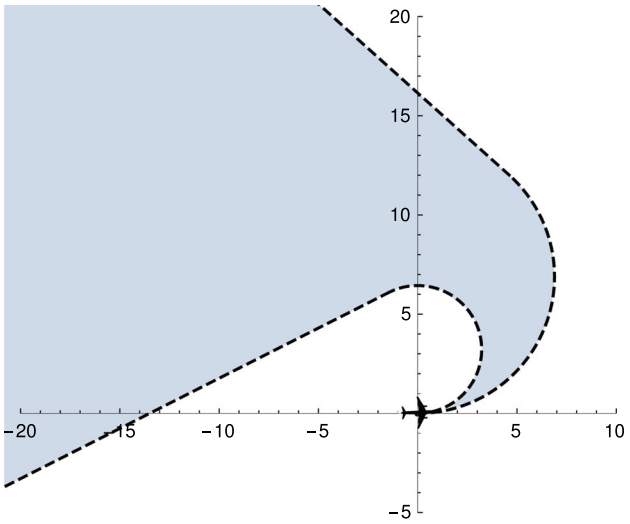
A number of efforts have gone on to formalize horizontal motion and prove properties about it, but all have characteristics that distinguish them from our work. For instance, [9] develops an approach for maneuvering and coordinating vehicles following Dubins paths utilizing Kripke models which is verified via a model checker, but does not incorporate non-determinism in the turning models, and does not consider timing characteristics of the turns. Examples such as [1] use differential dynamic logic with KeYmaera X to model collision avoidance in automobiles with skidding, but unlike our work they are concerned mainly with geometric properties of paths and do not consider timing. The work in [18] is an excellent treatment of collision avoidance in a wide variety of uncertain turning scenarios for ground robots. It assumes obstacles characterized by maximum velocity bounds, is not focused on timing analysis, and is not tailored for use in mixed vertical and horizontal collision avoidance. We develop a new expression for calculating allowable tangents to a turn; an alternate solution to this problem is reported in [20].

In [25], authors develop an algorithm for safe trajectories for robots and dynamic obstacles following constant speed trajectories with an upper and lower bounds on curvature, which encompasses turn-to-bearing maneuvers we consider. However, it uses a rather coarse approximation of the collision region in the velocity obstacle space, taking a union of all reachable regions over a time window, where we show how to compute a much tighter conflict region approximation over time. This has the advantage of allowing greater maneuverability which is particularly important in adversarial scenarios.

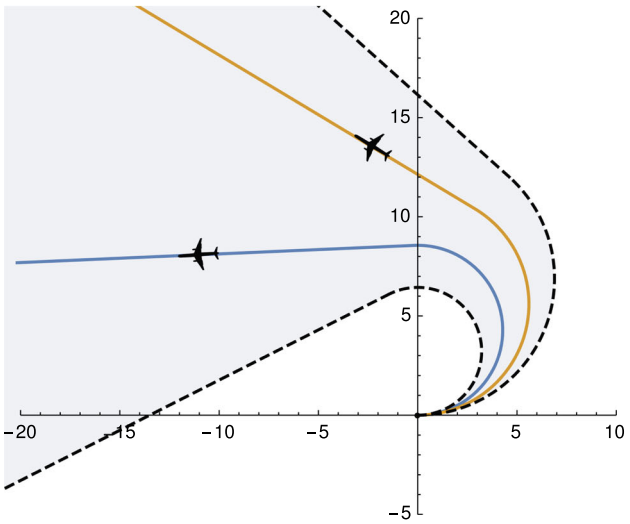
Closely related to this work is [21], which considers curved, horizontal aircraft avoidance maneuvers, but without combining them with vertical maneuvers; and [8], which considers vertical maneuvers, but with straight line horizontal kinematics, and does not allow combination with horizontal maneuvers.

Also closely related to this work is [10], which analyzes vertical maneuvers, but contains timing parameters that can be set to ensure safety for simultaneous horizontal maneuvers. Our timing computation can be used to set parameters that safely compose turn-to-bearing horizontal maneuvers with arbitrary bounded-acceleration vertical maneuvers.

Dubins paths, constructed of circular arc segments and straight lines, are used to model horizontal motion in many path planning and collision avoidance algorithms, such as [4,7,16,17,23,26]. These examples are not formally verified, and although some are created with aircraft in mind, they are not designed for timing analysis or adversarial collision avoidance assumptions in our work.



(a) Shaded area is reachable in the future.



(b) Paths show two possible trajectories.

Fig. 1 Visualizing turn-to-bearing motion

Many years of work have gone into the tools and libraries that we used for our development, including the Coq proof assistant [24] and the Coquelicot extensions for its real library [2]. Our libraries are intended to contribute to this toolbox.

3 Reasoning foundations for turn-to-bearing maneuvers

The first step in reasoning about the safety of turn-to-bearing maneuvers is to formalize the definition of a turn-to-bearing trajectory in a manner suitable for use in the Coq proof assistant. We consider the mathematical definition of turn-to-bearing kinematics, a library interface, and the formalization of geometric properties that are necessary for our safety analysis.

3.1 Non-deterministic Turn-to-bearing kinematics

We define *non-deterministic one-turn-to-bearing* motion as a set of trajectories representing a range of future motion possibilities that might be followed by the vehicle. We characterize this motion with a tuple that represents the set of future trajectories that are possible $(x_0, y_0, \theta_0, r_\alpha, r_\beta, \theta_\alpha, \theta_\beta, s_\alpha, s_\beta)$, where (x_0, y_0) and θ_0 are initial position and orientation of the vehicle; $r_\alpha, r_\beta, \theta_\alpha, \theta_\beta$ are bounds on the fixed turn radius and cumulative change in orientation after completing the circular turn, respectively; and s_α, s_β are bounds on the speed throughout the encounter, which unlike the other parameters is assumed to vary continuously as a function of time. We adopt the convention of using positive radii and bearing offsets to represent counterclockwise (left) turns, and negative radii and bearing offsets to represent clockwise (right) turns. Left turns are represented by $0 < r_\alpha \leq r_\beta$ and $0 < \theta_\alpha < \theta_\beta < 2\pi$, while right turns by $r_\alpha \leq r_\beta < 0$ and $-2\pi < \theta_\alpha < \theta_\beta < 0$. In all cases, we assume $0 < s_\alpha \leq s_\beta$. Realizing a specific future trajectory requires drawing from this sample space. Each trajectory has parameters $r, \theta_c, s(t)$ satisfying the constraint predicate $\chi(r, \theta_c, s) = \theta_c \in [\theta_\alpha, \theta_\beta] \wedge r \in [r_\alpha, r_\beta] \wedge (\forall u, s(u) \in [s_\alpha, s_\beta])$ which represents a path with initial turn that we model using a circular arc of radius r, followed by a linear path tangent to the turn whose bearing is offset by θ_c from θ_0. The path is traversed with continuously varying speed $s(t)$. Figure 1a plots a visualization of the turn-to-bearing envelope for $(x_0, y_0, \theta_0, r_\alpha, r_\beta, \theta_\alpha, \theta_\beta, s_\alpha, s_\beta) = (0, 0, 0, 3.22, 6.89, 2.41, 3.62, 1, 2)$, while Fig. 1b shows example trajectories consistent with that envelope.

Components of the vehicle's trajectory for these kinematics are given by

$$J_x(t) = \begin{cases} r\sin\left(\frac{\mathrm{d}(t)}{r} + \theta_0\right) - r\sin(\theta_0) + x_0 & \mathrm{d}(t) \leq r\theta_c \\ (\mathrm{d}(t) - r\theta_c)\cos(\theta_c + \theta_0) & \\ \quad + r\sin(\theta_c + \theta_0) - r\sin(\theta_0) + x_0 & \mathrm{d}(t) > r\theta_c \end{cases} \tag{1}$$

$$J_y(t) = \begin{cases} -r\cos\left(\frac{\mathrm{d}(t)}{r} + \theta_0\right) + r\cos(\theta_0) + y_0 & \mathrm{d}(t) \leq r\theta_c \\ (\mathrm{d}(t) - r\theta_c)\sin(\theta_c + \theta_0) & \\ \quad - r\cos(\theta_c + \theta_0) + r\cos(\theta_0) + y_0 & \mathrm{d}(t) > r\theta_c \end{cases} \tag{2}$$

for overall trajectory $J(t) = J_x(t)\hat{x} + J_y(t)\hat{y}$. The distance traveled on the path is related to speed during the trajectory in the usual way, $\mathrm{d}(t) = \int_0^t s(\gamma)\mathrm{d}\gamma$.

3.2 Library interface

The library we have developed is organized around the representation of a path in $\mathbb{R}^2$ and a predicate

$$\text{path\_segment}(D,\ f_x(d),\ f_y(d),\ (x_0,\ y_0),\ (x_1,\ y_1)) \tag{3}$$

which, when true, asserts: that $f_x(d)$ and $f_y(d)$ are parameterized functions describing the x and y positions of the path in the coordinate plane; that the resulting path is continuous and integrable; and that $f_x(d)$ and $f_y(d)$ are parameterized by the path distance, i.e., $\int_0^d \sqrt{(f_x'(\alpha))^2 + (f_y'(\alpha))^2}\mathrm{d}\alpha = \mathrm{d}$; that $(f_x(0),\ f_y(0)) = (x_0,\ y_0)$; and that $(f_x(D),\ f_y(D)) = (x_1,\ y_1)$. Parameterizing our path representation by path distance creates a canonical representation of the geometry for each path, isolating it from timing considerations associated with variations in speed during the maneuver. This allows us to analyze each aspect separately and combine them in the end.

Note that although the turn-to-bearing paths in the library define a starting and ending point separated by distance D, the paths continue indefinitely.

The library also contains piecewise functions parameterizing the x and y positions for turn-to-bearing paths $H_x(r, \theta_0, x_0, \theta_c, rtp, d)$ and $H_y(r, \theta_0, y_0, \theta_c, rtp, d)$, meant to be used with the path_segment predicate. The functions are equivalent to Eqs. (1) and (2), differing only in that they are parameterized by distance d instead of time t. The functions are curried before being used in path_segment, instantiated with starting point $(x_0,\ y_0)$, initial orientation θ_0, the turn radius r, and the angular offset for the final bearing θ_c. They also require an argument named rtp, which must be a proof object showing that $0 < r\theta_c < 2\pi|r|$, ensuring the signs of r and θ_c to be identical, and enforcing an upper bound on θ_c. The files *ttyp.v* and *tdyn.v* define the path_segment predicate, the parameterized turn-to-bearing paths, and prove lemmas about path continuity, differentiability, and path-length parameterization of H_x and H_y so they can be used with the path_segment predicate. Along with the parameterization, the library contains predicates straight and turning which indicate whether the parameters describing a path reach the final destination point while traveling in a straight line, or turning on a circular arc, respectively.

The rest of the library includes trigonometric definitions and identities that are missing from the Coq standard library (*atan2.v*, *strt.v* and *strt2.v*), lemmas that help the user introduce turn-to-bearing path_segment predicates into the context (*tlens.v*), lemmas that derive consequences and mathematical relationships from turn-to-bearing path_segment assumptions (*tlens.v*), lemmas about timing intervals (*ttim.v*), and theorems about the computation of timing properties based on pathlength (*dtlen.v*). The size of the development is significant, around 40k lines of proof scripts.

Because Coq allows expression in a higher-order logic, it permits quantification over any variable. This means we can hold the starting and ending points of the path fixed and quantify over the other parameters to reason about waypoints, or fix ranges of parameters and quantify over the radii and angles to reason about ranges of non-deterministic possibilities in turn radius and final bearing.

In this paper, for clarity, we present lemmas from the library in a standard position and orientation such that $(x_0,\ y_0) = (0, 0)$, $\theta_0 = 0$, and $(x_1,\ y_1) = (x,\ y)$. To analyze intersecting paths that are oriented and positioned arbitrarily with respect to one another, the more general form can be recovered by assuming that

$$x = (x_1 - x_0)\cos(\theta_0) + (y_1 - y_0)\sin(\theta_0) \tag{4}$$

$$y = -(x_1 - x_0)\sin(\theta_0) + (y_1 - y_0)\cos(\theta_0) \tag{5}$$

The library itself contains the translations and rotations to allow full generality when working with more than one path.

3.3 Trigonometric properties

Geometric intuition which might seem simple does not always translate naturally to formal analysis in a proving environment.

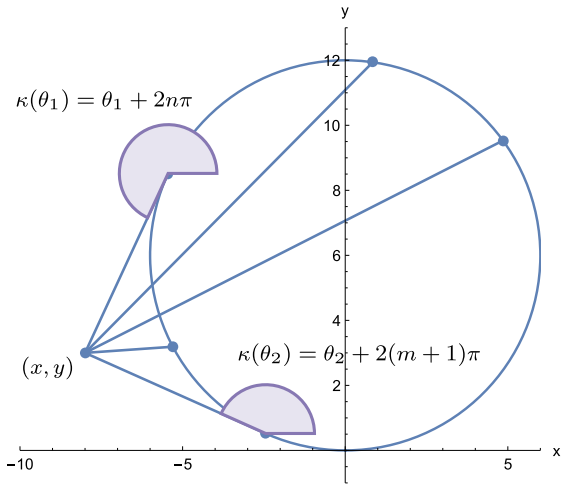
First we needed to encode in our proving environment a basic understanding of the way circular turns may be combined with straight paths that exit the turns on a tangent. There are two tangent lines to a circle, anchored at orientations θ_1 and θ_2, which arrive at any particular point $(x,\ y)$ outside the circle (see Fig. 2).

One of the tangents is not useful because for counter-clockwise turns, it always results in a path with a discontinuous derivative. This is geometrically obvious to a human by inspection, but somewhat challenging to formalize in Coq.

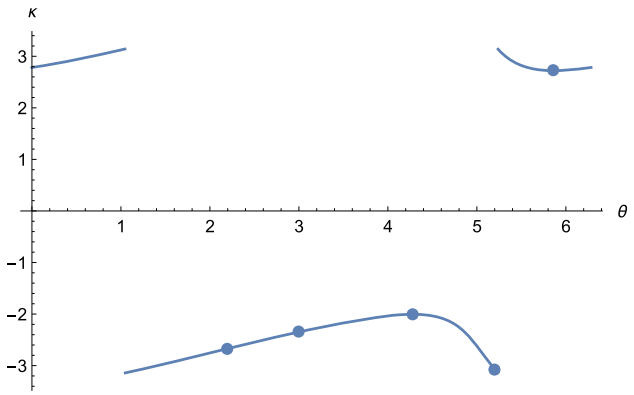
Using a chord lemma from geometry, we can infer that a vehicle approaching $(x,\ y)$ from a circular turn will do so at an angle of $\theta_m = 2\,\mathrm{atan}_2(y, x)$ and that the radius required to reach it will be $r_m = (x^2 + y^2)/(2y)$, see Fig. 3. We find that decreasing the radius of the turn decreases the angle, while increasing it makes $(x,\ y)$ unreachable via a tangent line. Thus, θ_m defines an upper bound on the approach angle. From inspection, we can see that the angle of the second tangent exceeds this boundary.

In order to formalize this geometric intuition, we define a function which given the orientation θ of the vehicle on a turning path of radius r would return the angle from the vehicle to the point $(x,\ y)$,

$$\kappa(\theta) = \mathrm{atan}\left(\frac{y - r(1 - \cos(\theta))}{x - r\sin(\theta)}\right). \tag{6}$$



(a) Angle of each line segment is κ



(b) Discontinuous κ_2; $\kappa' = 0$ points are tangent

Fig. 2 If we parameterize positions on a circular path using the vehicle orientation θ associated with the tangent, then $\kappa(\theta)$ (shaded) is the angle of the line connecting the point on the circle to a point (x, y) outside the circle

Both by construction, and by the periodicity of sin and cos, we note that κ is periodic with period 2π. For the remainder of this section we will restrict the domain of κ to $(0, 2\pi)$ for $r > 0$, and $(-2\pi, 0)$ for $r < 0$. As illustrated in Fig. 2b, the function κ is not continuous for all values of the destination point (x, y).

We define a series of functions based on a two-argument arctangent and different branch cuts, which have distinct, overlapping, and complementary domains upon which (x, y) yields a continuous function.

$$\kappa_2(\theta) = \operatorname{atan}_2\big(y - r(1 - \cos(\theta)), x - r\sin(\theta)\big) \tag{7}$$

$$\kappa_3(\theta) = \operatorname{atan}_2\big(-(y - r(1 - \cos(\theta))), -(x - r\sin(\theta))\big) + \pi \tag{8}$$

$$\kappa_4(\theta) = \operatorname{atan}_2\big(-(x - r\sin(\theta)), y - r(1 - \cos(\theta))\big) + \pi/2 \tag{9}$$

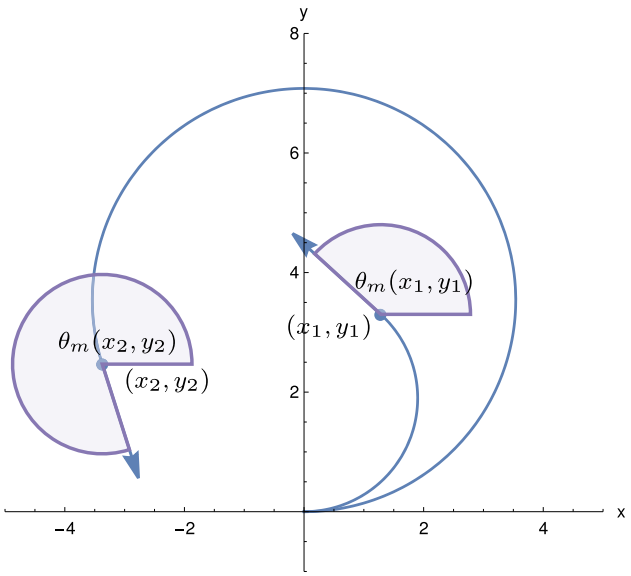


Fig. 3 Two example points illustrating the geometric intuition for r_m and θ_m using circular arcs that connect them with the origin. Each arc has a radius of r_m, and a tangent of θ_m at its end, where each parameter is calculated with the coordinates of the final point

Henceforth, when we refer to properties of κ, we are choosing a variant with the branch cut oriented so that there is no discontinuity for the given destination point (x, y).

When κ is continuous, we show that the unique maximum and minimum values $\kappa(\theta_1)$ and $\kappa(\theta_2)$ correspond to the angles of the correct and incorrect tangent lines, respectively (for $r > 0$, if $r < 0$ the maxima and minima are reversed). We prove that $\kappa(\theta) = \frac{\theta_m}{2}$ implies that $\theta = 0$ or $\theta = \theta_m$. Since κ takes on the value $\frac{\theta_m}{2}$, it must be that $\kappa(\theta_1) \geq \frac{\theta_m}{2} \geq \kappa(\theta_2)$.

Our choice of domain ensures that 0 is not between θ_1 and θ_2, so we can use the Intermediate Value Theorem to show that θ_m is in-between θ_1 and θ_2 in the domain. Because θ_m is a limiting value of the approach angle, we can eliminate θ_2, which is always outside of the allowable range, leaving θ_1 as the angle of approach that ensures path continuity.

We calculate extremal values of κ, θ_1 and θ_2, by setting the derivative of κ to zero, and solving for the argument. Fortunately, each variant of the κ function for which the destination point (x, y) yields a continuous function has the same derivative:

$$\kappa'(\theta) = \frac{r((2r - y)\tan^2(\theta/2) - 2x\tan(\theta/2) + y)}{D(\theta)}, \tag{10}$$

for $\theta \notin \{0, \pi\}$. The sign of the denominator

$$D(\theta) = 2(1 - \cos(\theta))/\sin^2(\theta) \cdot \Big((y - r(1 - \cos(\theta)))^2 + (x - r\sin(\theta))^2\Big) \tag{11}$$

is always positive, and so the sign of κ' is directly related to the sign of the quadratic function in the numerator; the task of calculating the maximum and minimum is reduced to the problem of solving a quadratic in $\tan(\theta/2)$. The solution associated with the maximum value of κ is given in Eq. (14).

Reasoning about the continuity of the κ variants, handling their derivatives as the angle crosses the branch cut, and ordering of roots and angles to establish what "in-between" means in an angular domain that is a clock system is contained within the file *strt.v* and its corresponding documentation.

3.4 Turn-to-bearing path properties

Parameters for turn-to-bearing trajectories must be selected in a way that the radius and angle of departure from the turn lead from the starting point to the ending point, and so that the distance is consistent with the path. In this section, we state basic results about paths, and select a few proofs about which we provide some details in order to give a flavor of the reasoning in the library.

We can construct a turn-to-bearing trajectory by first choosing an angle of approach θ for a point (x, y), and then computing the turn radius required to arrive there with that orientation. The angle of approach is constrained because of the initial position and angle of the aircraft and the required kinematics. Figure 4a shows the initial position of an aircraft and for a series of example reachable points, plots shows the angular extent of feasible approaches using pie slice-shaped circular segments; an example path is shown to illustrate one possibility to reach one of the points. Figure 4b graphically plots, for the lower left point, Eq. 13, i.e., what r must be for each allowable choice of θ at that point.

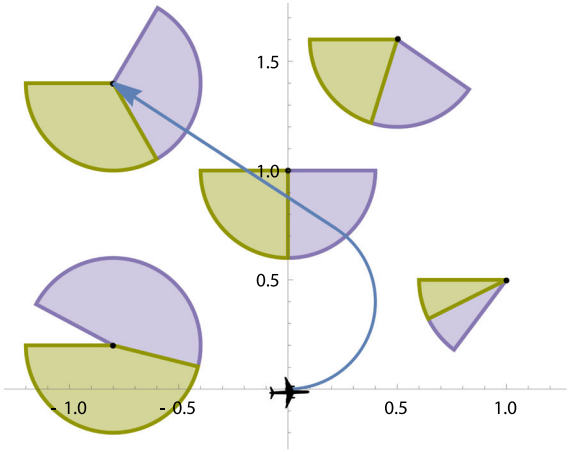
Theorem 1 (Turn-to-bearing dependent radius) *A vehicle following a turn-to-bearing trajectory can approach point* (x, y) *with a chosen angle* θ *when*

$$\begin{aligned}&(0 < \theta_m \wedge (\theta_m/2 < \theta \leq \theta_m \vee -2\pi < \theta < \theta_m/2 - 2\pi)) \vee \\ &\quad (\theta_m < 0 \wedge (\theta_m \leq \theta < \theta_m/2 \vee \theta_m/2 + 2\pi < \theta < 2\pi))\end{aligned} \tag{12}$$

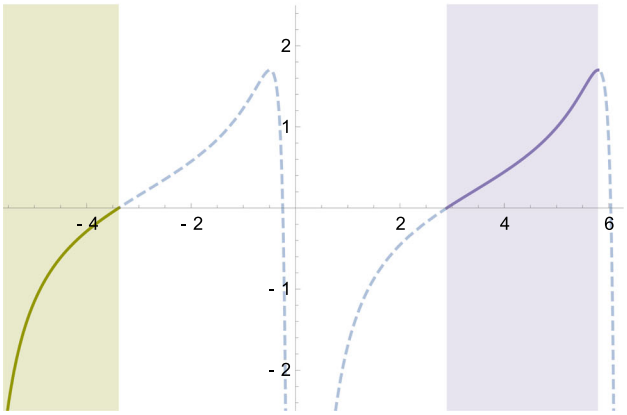
using radius

$$R(x, y, \theta) = \frac{x \sin(\theta) - y \cos(\theta)}{1 - \cos(\theta)}. \tag{13}$$

Similarly, we can also construct a turn-to-bearing trajectory by first choosing a turn radius r and then computing the angle of approach that the radius entails when we arrive at (x, y). The choice of radius is constrained if the target point is on the same side as the direction of the turn, because the turn must be rapid enough to orient the aircraft in the direction of the target point before it has passed it.



(a) Approaches for different points.



(b) r vs. θ for a single point $(-0.8, 0.2)$.

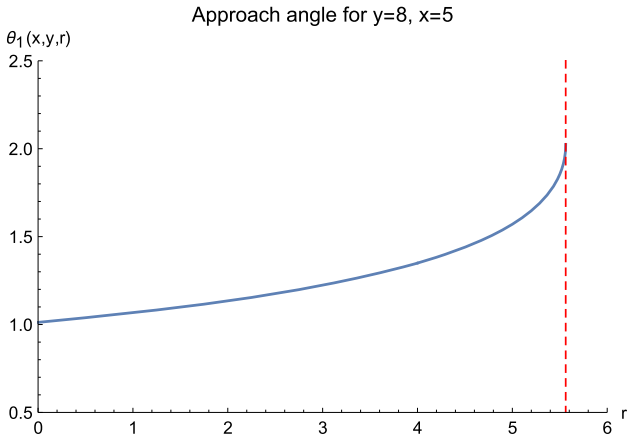
Fig. 4 Relationship between allowable angle of approach and required radius to achieve that angle. Choosing angular ranges of approach also entails a turn direction; left turns are marked with violet and right turns marked with green (color figure online)

Theorem 2 (Turn-to-bearing dependent approach angle) *A vehicle following a turn-to-bearing trajectory can approach point* (x, y) *using a turn with chosen radius* r *when*

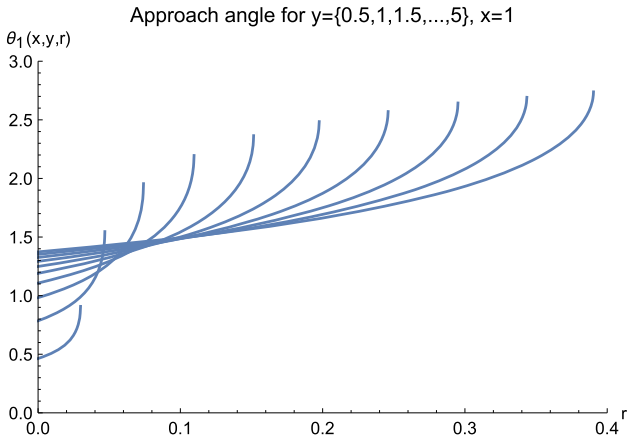
$$\left(0 < y \wedge r \leq \frac{x^2 + y^2}{2y}\right) \vee (y = 0 \wedge x < 0) \vee \left(y < 0 \wedge \frac{x^2 + y^2}{2y} \leq r\right)$$

and the angle of approach is

$$\Theta(x, y, r) = \begin{cases} 2\,\mathrm{atan}\left(\frac{x - \sqrt{x^2 - (2r - y)y}}{(2r - y)}\right) + P & 2r - y \neq 0 \\ 2\,\mathrm{atan}\left(\frac{y}{2x}\right) & 2r - y = 0 \wedge x > 0 \\ \pi\,\mathrm{sign}(r) & 2r - y = 0 \wedge x \leq 0 \end{cases} \tag{14}$$



(a) Varying r for a single (x, y).



(b) Varying r for a sequence of points.

Fig. 5 Plot of the first piece of $\Theta(x, y, r)$ from Eq. (14)

where $P = P(x, y, r)$ *is a phase correction given by*

$$P(x, y, r) = \begin{cases} 0 & (0 < r \wedge ((0 < x \wedge 0 < y) \vee x \leq 0 \wedge 2r < y)) \\ & \vee (r < 0 \wedge ((x < 0 \wedge y < 0) \vee y < 2r)) \\ 2\pi & 0 < r \wedge (0 \leq x \wedge y < 0 \vee x < 0 \wedge y < 2r)) \\ -2\pi & r < 0 \wedge (0 \leq x \wedge 0 < y \vee x < 0 \wedge 2r < y). \end{cases}$$

It is not surprising that for fixed (x, y), the first piece of $\Theta(x, y, r)$ is not differentiable or even always defined at $r = r_m$. What is surprising is that even if we define the endpoint to ensure the value of the function is finite, its rate of change is unbounded at the end of the interval. We initially expected that we could simply extend our θ_1 curve to create a function with a continuous derivative, but in the end had to settle for creating a piecewise continuous function using the limiting value at the cutoff point, which turned out to be enough for our purposes. This is illustrated in Fig. 5 and made formalizing the relationship between the length of circular arc path segments and the rest of the turn-to-bearing kinematics a longer process than we had expected.

This geometry appears in a variety of contexts, including [20, p. 15], which has another expression that may be used to solve for the angle. We leave it to the interested reader to show the equivalence between the result we have proved, and alternate formulations. We also found a remarkable simplification for the tangent path length:

Theorem 3 (Straight path segment expression) *For a turn-to-bearing trajectory given by* $(r, \Theta(x, y, r))$*, that starts at the origin with* $\theta_0 = 0$ *and passes through* (x, y)*, the square of the distance traveled on a straight line before we arrive at* (x, y) *is given by*

$$(x - r\sin(\Theta(x, y, r)))^2 + (y - r(1 - \cos(\Theta(x, y, r))))^2 = x^2 - (2r - y)y. \tag{15}$$

4 Reasoning about the timing of intersecting turns

This section describes the application of our turn-to-bearing Coq library to formalize and formally verify an exact, non-trivial timing property of these trajectories.

Having formalized turn-to-bearing paths, we need to reason about when (timing) and where (geometry) collisions might occur. The geometry of the reachable envelope for a turn is bounded by edges that are combinations of circular arcs and straight lines; the intersection of these areas can be computed in a straightforward manner. In other words, it is straightforward to overapproximate the conflict area as shown in Fig. 6.

In general, a collision can occur if there exists a point such that both aircraft can reach that point at the same time. This section considers the theorems and equations necessary to compute the earliest and latest possible times that aircraft can be at a *given* point. Intuitively, the possible locations of an aircraft are contained within an area that moves over time, a propagating wave within the reachable envelope with a leading and lagging edge. This wave of position possibilities can be computed via piecewise equations. Although it is intractable to exhaustively search over all points for the exact collision times, these theorems will permit us to calculate a sound overapproximation for collisions in later sections.

4.1 Pointwise collision timing

We define the reachable envelope

$$E = \{p \mid \exists\, (\theta_c, r, s, u),\ \chi(r, \theta_c, s) \wedge u > 0 \wedge J(u) = p\} \tag{16}$$

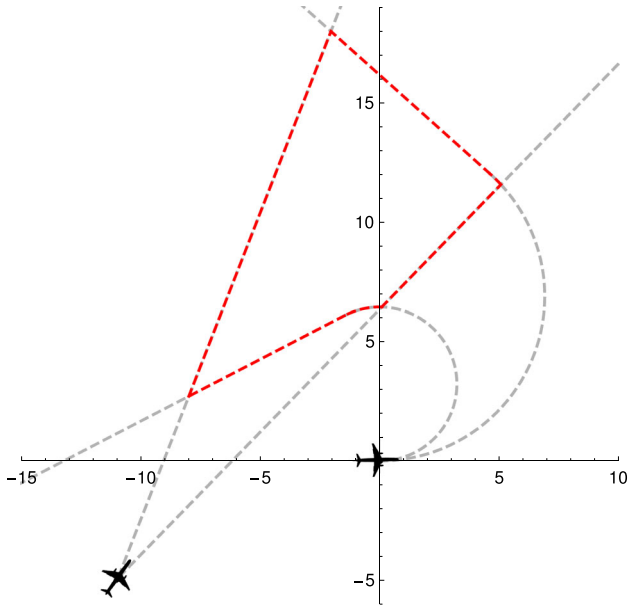


Fig. 6 A two-aircraft encounter. Dashed lines show the edges of the reachable area for each aircraft over all time, i.e., all possible future positions throughout the entire encounter. The intersection of reachable areas—outlined in red—is an overapproximation of the conflict area, containing all possible future collisions (color figure online)

for a vehicle to be the set of points that are reachable over the range of possible future trajectories. For any point in the reachable envelope $p \in E$, there is a set of trajectories $\mathcal{T}(p) = \{J(\cdot) \mid \exists\,(\theta_c, r, s, u),\ \chi(r, \theta_c, s) \wedge u > 0 \wedge J(u) = p\}$ that can reach that point. Each trajectory $J \in \mathcal{T}(p)$ corresponds with a different choice of radius and final bearing (which determine the path), and future ground speed $s(t)$. Figure 7a, b illustrates two different points in the reachable envelope of the ownship from Fig. 1 and possible paths taken from the family of trajectories that could reach each point.

There is a corresponding set of arrival times $I(p) = \{t_a \mid J \in \mathcal{T}(p) \wedge J(t_a) = p\}$ at which a vehicle can arrive at p. The earliest and latest arrival times for a single vehicle at a point p are then given by

$$t_{\mathrm{e}}(p) = \inf I(p) \tag{17}$$
$$t_{\mathrm{l}}(p) = \sup I(p). \tag{18}$$

To determine whether a collision between two aircraft is possible, we must look at the earliest and latest arrival times for each aircraft at each point in its reachable envelope. To organize this analysis, we first define four logical predicates that express whether the earliest and latest arrival time at point p in the reachable area occur when the other vehicle may also be located at that point. Each time variable t in the subsequent equations has a subscript indicating whether the time is the earliest possible ($_{\mathrm{e}}$) or latest possible ($_{\mathrm{l}}$) time of arrival, and a superscript indicating which aircraft timing is referenced, $^{\mathrm{i}}$ for intruder or $^{\mathrm{o}}$ for ownship.

$$W^{\mathrm{i}}_{\mathrm{e}}(p) = t^{\mathrm{o}}_{\mathrm{e}}(p) \le t^{\mathrm{i}}_{\mathrm{e}}(p) \le t^{\mathrm{o}}_{\mathrm{l}}(p) \tag{19}$$
$$W^{\mathrm{i}}_{\mathrm{l}}(p) = t^{\mathrm{o}}_{\mathrm{e}}(p) \le t^{\mathrm{i}}_{\mathrm{l}}(p) \le t^{\mathrm{o}}_{\mathrm{l}}(p) \tag{20}$$
$$W^{\mathrm{o}}_{\mathrm{e}}(p) = t^{\mathrm{i}}_{\mathrm{e}}(p) \le t^{\mathrm{o}}_{\mathrm{e}}(p) \le t^{\mathrm{i}}_{\mathrm{l}}(p) \tag{21}$$
$$W^{\mathrm{o}}_{\mathrm{l}}(p) = t^{\mathrm{i}}_{\mathrm{e}}(p) \le t^{\mathrm{o}}_{\mathrm{l}}(p) \le t^{\mathrm{i}}_{\mathrm{l}}(p) \tag{22}$$

We combine these to define two predicates to evaluate safety, one using the earliest arrival time, and the other using the latest arrival time,

$$W_{\mathrm{e}}(p) = W^{\mathrm{i}}_{\mathrm{e}}(p) \vee W^{\mathrm{o}}_{\mathrm{e}}(p) \tag{23}$$
$$W_{\mathrm{l}}(p) = W^{\mathrm{i}}_{\mathrm{l}}(p) \vee W^{\mathrm{o}}_{\mathrm{l}}(p) \tag{24}$$

For two aircrafts, we define a conflict area $C = E^{\mathrm{o}} \cap E^{\mathrm{i}}$ to reflect the geometry of the intersection of future paths without timing considerations. We prove:

Theorem 4 (Leading Lagging Equivalence) *For all points $p \in C$, the predicates $W_{\mathrm{e}}(p) = W_{\mathrm{l}}(p)$ are equal, so we can drop the subscript.*

Theorem 5 (Pointwise Safety) *$W(p)$ correctly establishes safety at point p: when it is true, there exist circumstances that lead to collision at p, and when it is not there are no circumstances that lead to collision at p.*

Theorem 6 (Collision Timing) *For each point $p \in C \wedge W(p)$, a collision may only occur in the time interval $[\max\left(t^{\mathrm{i}}_{\mathrm{e}}(p), t^{\mathrm{o}}_{\mathrm{e}}(p)\right), \min\left(t^{\mathrm{i}}_{\mathrm{l}}(p), t^{\mathrm{o}}_{\mathrm{l}}(p)\right)]$, and under the assumptions, no collision may occur outside this time interval.*

We can directly relate timing of a trajectory between two points to the range of path lengths for different possible paths connecting the points. The earliest arrival time to reach a point p, $t_{\mathrm{e}}(p)$ is achieved by the trajectory following the shortest path and the highest ground speed, i.e., $\inf I(p) = \frac{d_{\min}(p)}{s_\beta}$, where $d_{\min}(p)$ is the length of the shortest path from the starting point to p. The latest arrival time $t_{\mathrm{l}}(p)$ is achieved by the trajectory following the longest path with the slowest ground speed, i.e., $\sup I(p) = \frac{d_{\max}(p)}{s_\alpha}$, where $d_{\max}(p)$ is the length of longest path from the starting point to p. In this way, we convert the problem of computing collision timing into a problem computing the range of possible path lengths between two points.

4.2 Path length properties

We can define a function that computes the length of the path for a deterministic, left-turning turn-to-bearing trajectory starting from the origin with orientation $\theta_0 = 0$, passing through (x, y) with orientation θ, using a turn of radius r:

$$L(x, y, \theta, r) = r\theta + \|(x, y) - r(\sin\theta, 1 - \cos\theta)\|\,. \tag{25}$$

As discussed in Sect. 3.4, turn-to-bearing kinematics constrain the parameters for L, i.e., its arguments cannot all be chosen independently. Assume we fix the point we wish to reach, (x, y). We can independently choose the angle of approach θ to the final point, and that determines the turn radius of the maneuver. Alternatively, we can choose the radius of the turn, and compute the angle of approach to the point.

A central insight here is that for paths with the same starting and ending points, the path with a larger angle of approach will have a larger radius; and the path with a larger radius will be longer. More precisely:

Theorem 7 (Approach angle orders *turn-to-bearing* path radii) *Given two turn-to-bearing paths,* (r_1, θ_1) *and* (r_2, θ_2) *that pass through the same point* (x, y)*, if* $\theta_1 > \theta_2 > 0$*, then the radius of the first path* r_1 *is longer than the radius of the second path* r_2*, i.e.,* $r_1 > r_2$*:*

$$(\theta_1 > \theta_2 > 0) \rightarrow R(x, y, \theta_1) > R(x, y, \theta_2) \tag{26}$$

Theorem 8 (Radius orders turn-to-bearing path lengths) *Given two turn-to-bearing paths,* (r_1, θ_1) *and* (r_2, θ_2) *that pass through the same point* (x, y)*, if* $r_1 > r_2 > 0$*, then the first path length* L_1 *is greater than the second path length* L_2*, i.e.,* $L_1 > L_2$*:*

$$\begin{aligned}&(r_1 > r_2 > 0) \rightarrow \\ &\quad L(x, y, \Theta(x, y, r_1), r_1) > L(x, y, \Theta(x, y, r_2), r_2)\end{aligned} \tag{27}$$

4.2.1 Maximum and minimum path lengths

At each point in the reachable area, we can use the ordering of path lengths implied by Theorems 7 and 8 to find the minimum and maximum length path possible for uncertain turn-to-bearing motion constrained by non-deterministic bounds.

Theorem 9 (Minimum bearing-constrained path length) *For turn-to-bearing kinematics, given interval constraints on final bearing* $[\theta_\alpha, \theta_\beta]$ *and turn radius* $[r_\alpha, r_\beta]$ *where* $0 < r_\alpha$ *and* $0 < \theta_\alpha$*, and a reachable point* (x, y)*, the minimum path length is given by Eq.* (28)*.*

Theorem 10 (Maximum bearing-constrained path length) *For turn-to-bearing kinematics, given interval constraints on final bearing* $[\theta_\alpha, \theta_\beta]$ *and turn radius* $[r_\alpha, r_\beta]$ *where* $0 < r_\alpha$ *and* $0 < \theta_\alpha$*, and a reachable point* (x, y)*, the maximum path length is given by Eq.* (29)*.*

$$d_{\min}(x, y) = \begin{cases} L(x, y, \Theta(x, y, r_\alpha), r_\alpha) & \theta_\alpha \le \Theta(x, y, r_\alpha) \le \theta_\beta \\ L(x, y, \theta_\alpha, R(x, y, \theta_\alpha)) & (x^2 + y^2 > 2r_\alpha y \wedge (0 \le y(r_\alpha \le r_m \vee y = 0) \wedge \theta_\alpha < \theta_m) \vee \\ & \quad (y < 0 \wedge \theta_m < 0)) \wedge \Theta(x, y, r_\alpha) < \theta_\alpha \\ L(x, y, \theta_m, r_m) & r_\alpha \le r_m \le r_\beta \wedge \theta_m \le \max(\theta_\alpha, \Theta(x, y, r_\alpha)) \end{cases} \tag{28}$$

$$d_{\max}(x, y) = \begin{cases} L(x, y, \Theta(x, y, r_\beta), r_\beta) & x^2 + y^2 > 2r_\beta y \wedge \Theta(x, y, r_\beta) \le \theta_\beta \\ L(x, y, \theta_\beta, R(x, y, \theta_\beta)) & (x^2 + y^2 > 2r_\beta y \wedge \theta_\beta < \Theta(x, y, r_\beta)) \vee (r_\alpha \le r_m \le r_\beta \wedge \theta_\beta < \theta_m) \\ L(x, y, \theta_m, r_m) & r_\alpha \le r_m \le r_\beta \wedge \theta_m \le \theta_\beta \end{cases} \tag{29}$$

4.2.2 Right and uncertain turns

So far we have looked only at left turns, where the circle that defines our turn radius is positioned to the left of the vehicle, and the change in bearing is a relative angle in radians, positive according to the usual counter-clockwise convention. For non-deterministic left turns, $0 < r_\alpha \le r_\beta$ and $0 \le \theta_\alpha \le \theta_\beta$.

We can handle other types of turns via symmetry. For right turns, we choose the convention of identifying turning trajectories using radii with negative numbers, and giving relative bearing with negative numbers as well. We describe non-deterministic right turns using parameters such that $r_\alpha \le r_\beta < 0$ and $\theta_\alpha \le \theta_\beta < 0$. For this convention, the path length for right turns is given by:

$$L^{\text{right}}(x, y, \theta, r) = L(x, -y, -\theta, -r). \tag{30}$$

The function that determines the maximum and minimum distance for right turns is related to that for left turns in the following way:

$$d^{\text{right}}(x, y, \theta_\alpha, \theta_\beta, r_\alpha, r_\beta) = d(x, -y, -\theta_\beta, -\theta_\alpha, -r_\beta, -r_\alpha) \tag{31}$$

for both minimum and maximum distance.

We can compute the distances associated with non-deterministic forward motion that might include either a left

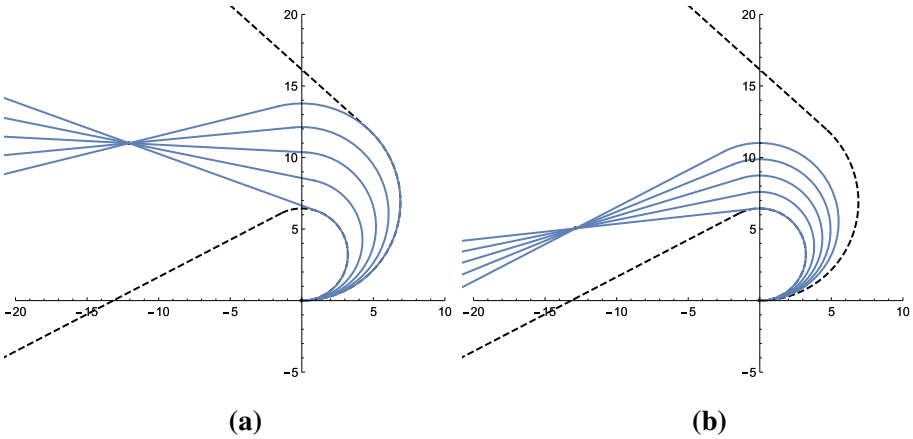


Fig. 7 Paths from the set of possible turn-to-bearing trajectories that reach two example points in space. The edges of the reachable envelope for a non-deterministic left turn are shown as a set of dashed lines; any point in the reachable envelope is reachable via these kinematics

or a right turn, by requiring $r_\beta < 0 < r_\alpha$ and $\theta_\alpha \leq 0 \leq \theta_\beta$. The distance function then relates to the left and right distance functions:

$$d^{\text{either}}(x, y, \theta_\alpha, \theta_\beta, r_\alpha, r_\beta) = \begin{cases} d(x, y, 0, \theta_\beta, r_\alpha, \infty) & y > 0 \\ d^{\text{right}}(x, y, \theta_\alpha, 0, -\infty, r_\beta) & y < 0 \\ x & y = 0 \end{cases} \tag{32}$$

4.3 Exact timing wavefront

The observations in Theorems 9 and 10 allow us to subdivide the reachable envelope into different areas, using a piecewise function to describe the timing. Figures 8 and 9 illustrate, for a single vehicle and a particular choice of parameters, the different strategies that maximize and minimize path length, and the areas associated with each strategy. The bounding areas that enclose uniform strategies are shown with dashed lines that illustrate the limits where each strategy is appropriate for finding minimum and maximum length. For turns with different parameters, these shapes change accordingly.

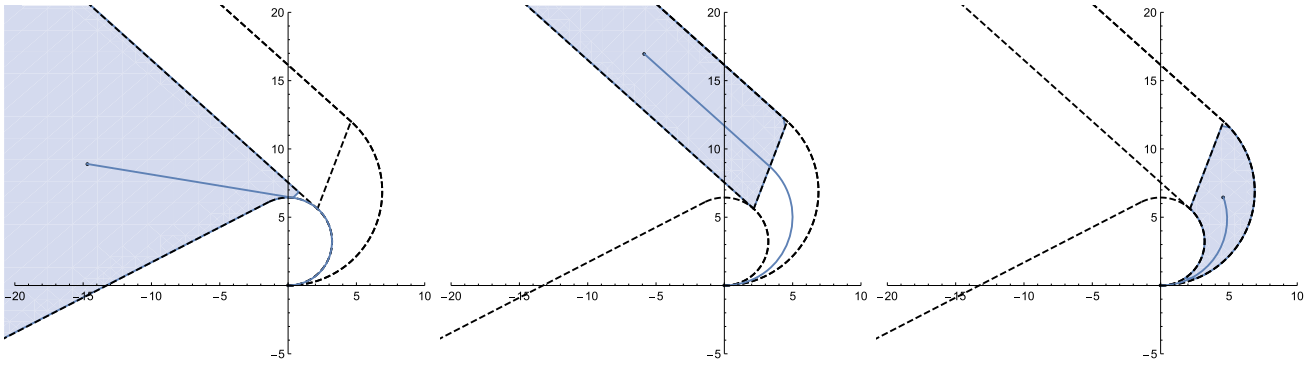
This means that if we want to find the shortest and longest paths to a point, we first consider paths with the smallest and greatest radii, r_α and r_β. Figures 8a and 9a illustrate individual trajectories that have minimum and maximum length for our example maneuver, constructed by using the minimum and maximum radii allowed. For some points, the most extreme turns could not produce trajectories that arrive at p, because the final bearings required by such trajectories are outside the parameters set for the motion, or because the points are inside the turning circle. Figures 8b and 9b illustrate individual trajectories that have minimum and maximum length for our example maneuver in this case. These are constructed by choosing radii that lead to most extreme values of bearing, so that the trajectory both reaches p, and does so with an orientation that is allowed by the parameters of our turn. Finally, there are points in the reachable envelope that are reachable as part of the initial turn. For these points, this initial turn is the maximum-length path. If the bearing at point p is outside the allowable range, then this is also the minimum-length path. Figures 8c and 9c illustrate individual trajectories that have minimum and maximum length for our example maneuver, which must be constructed as circular arcs.

In Fig. 10, we use the parameters of our example maneuver in Fig. 1, combining all of the results from Theorems 9 and 10 together into a single contour plot of the earliest and latest times to reach each point in the reachable envelope. The contour lines plotted in Fig. 10a, b can be thought of as the outer and inner boundaries (respectively) of the irregular annulus at the instant corresponding to the value of the contour. As time progresses, this annulus expands, so we can treat this like a propagating wave, an area that moves over time and encompasses all the possible positions that the vehicle may be in at each future moment.

The contours of equal timing for the minimum and maximum arrival times represent the shape of the leading and trailing edge of this wave, respectively. In addition to modeling and analyzing ranges of possibilities for turn-to-bearing kinematics, we will find that adding non-determinism also allows us to evaluate timing safety for small perturbations of turn-to-bearing—types of motion whose combination of trajectory and speed is sufficiently close, but not exactly the same.

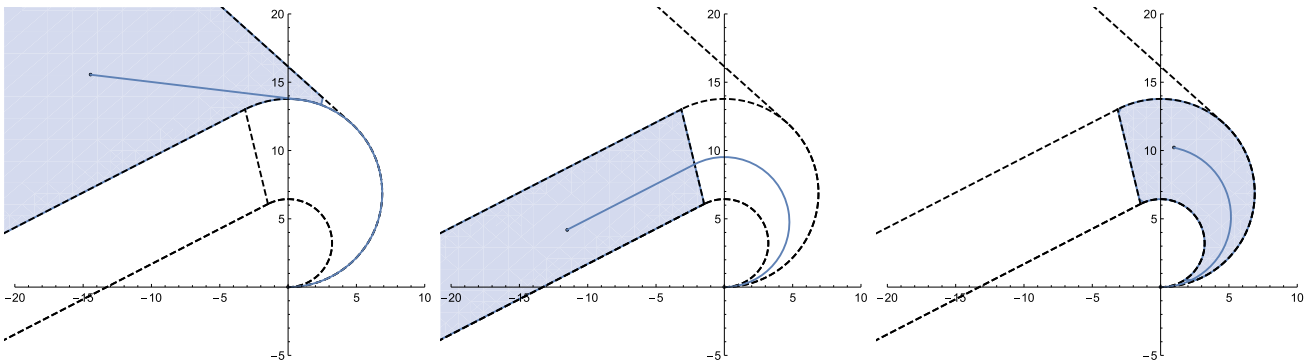
The theorems and equations within this section give exact, formally verified expressions that describe how to compute collision timing parameters $t_e(p)$ and $t_l(p)$ for two turning vehicles. Backed up by Theorem 6, these parameters describe the earliest and latest times that the vehicles may collide at a *given* point p that is in the conflict area C.

The next three sections that follow develop the mechanics for quantifying these calculations over *all* the points p in C. Quantifying the timing computation over the points in C is an important, practical step because it provides a parameter that can be used to fully characterize encounter timing.



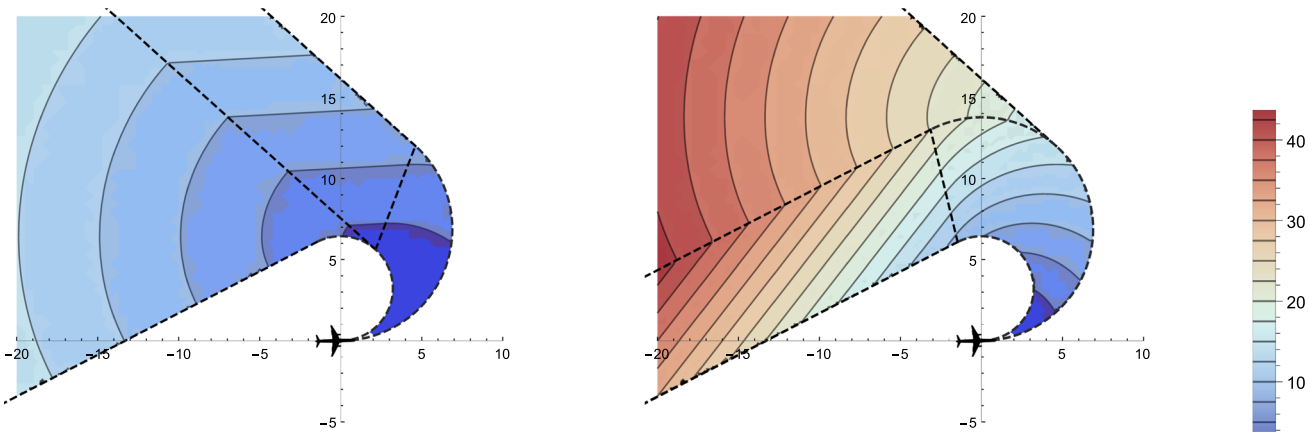
(a) $r = r_\alpha$ **(b)** $\theta_c = \theta_\alpha$ **(c)** $r = r_m \wedge \theta_m \leq \theta_c$

Fig. 8 Strategies described by Eq. (28) to find the minimum distance turn-to-bearing trajectory from the origin to a particular point for each of three possible regions. Example trajectories illustrate the strategy for a single point indicated in each region



(a) $r = r_\beta$ **(b)** $\theta_c = \theta_\beta$ **(c)** $r = r_m \wedge \theta_m \leq \theta_c$

Fig. 9 Strategies described by Eq. (29) to find the maximum length turn-to-bearing trajectory from the origin to a particular point when the point is in each of three possible regions. Example trajectories illustrate the strategy for a single point indicated in each region



(a) Minimum time to arrive, $t_e(p)$. **(b)** Maximum time to arrive, $t_l(p)$.

Fig. 10 Contour plot describing timing for a left turn with parameters used in Fig. 1. Each (x, y) position in the Cartesian plane is associated with a time to arrive at that position starting from $(0, 0)$ with orientation $\theta_0 = 0$, following turn-to-bearing kinematics

5 Quantifying collision timing computations over the conflict area

In this section, we develop an overall approach to quantifying the timing computations over the conflict area. We will describe the objective and intuition behind the quantification step, discuss the problems we encounter in more detail, and then create a framework for solving it. The following two sections fill in the remaining details of how we executed the approach.

To form some intuition about our algorithm, recall the example two-aircraft encounter (Fig. 6) where both aircraft follow uncertain turn-to-bearing trajectories. Dashed lines show the edges of the reachable envelopes E^{o} and E^{i} that contain the positions of each aircraft during the encounter, and illustrate how the conflict area C, outlined in red, is found by taking the intersection of the envelopes. We observe that the earliest and latest times can be computed by appropriately tracking the propagating timing wavefronts. To see how this works, refer to Fig. 11, which shows a series of snapshots of the future of the encounter from Fig. 6 (at times $t = 8$, 16, and 24) with shaded areas showing the set of possible positions in which each aircraft can be found at that moment. These moving areas are precisely the waves of possible positions described in Sect. 4.3. Their front and back edges are defined by the level sets of the timing computation, i.e., the contours of Fig. 10. Our objective is to solve for the time interval during which these propagating regions overlap.

Writing down the expression for timing parameters that span the entire encounter is straightforward. Recall from Theorem 6 that the earliest and latest collision times possible in an area C are given by

$$t_{\mathrm{e}} = \inf_{p \in C \wedge W(p)} \max\left(t_{\mathrm{e}}^{\mathrm{i}}(p), t_{\mathrm{e}}^{\mathrm{o}}(p)\right) \tag{33}$$

$$t_{\mathrm{l}} = \sup_{p \in C \wedge W(p)} \min\left(t_{\mathrm{l}}^{\mathrm{i}}(p), t_{\mathrm{l}}^{\mathrm{o}}(p)\right) \tag{34}$$

From the relationship between the timing of a trajectory at a point and the range of possible path lengths to arrive at that point, it suffices to analyze the path lengths (quantified over the conflict region), instead of considering timing directly (Sect. 4.1).

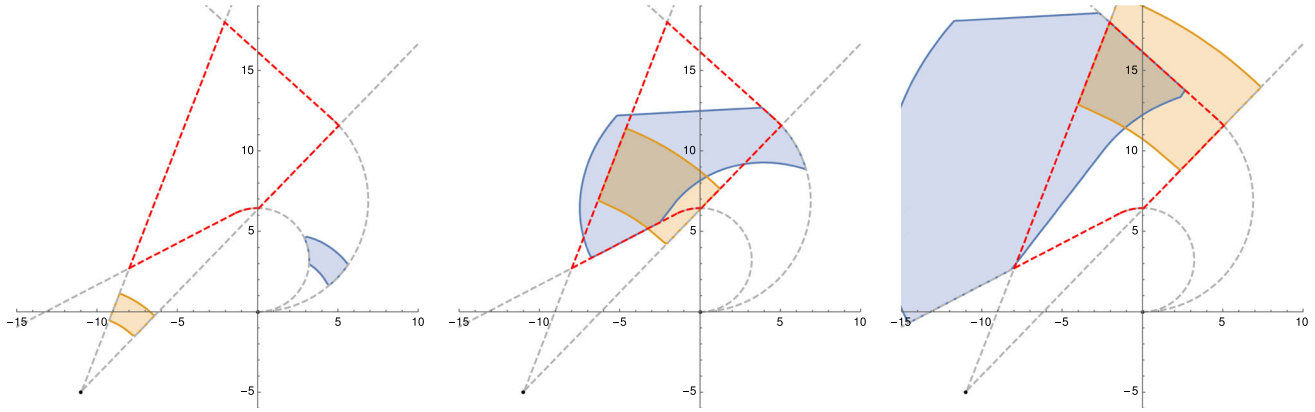
However, it is not obvious how to solve for t_{e} and t_{l} over every point in a region by using the minimum and maximum path lengths because there are a number of problems. The first problem we encounter is that our path length expressions in Eqs. (28) and (29) are piecewise, so we need some strategy for treating each piece separately. The second problem is that the domain over which the path length expressions apply is not convex and thus makes optimization more challenging. Figure 12 illustrates this with several examples of conflict areas C with different initial conditions. The conflict areas over which we need to quantify our timing calculation are neither convex nor necessarily simply connected. The framework we develop in this section will address the first two problems directly. The third problem is that the parameters x and y over which we are quantifying are in multiple places in the path length expressions, sometimes sprinkled within and between nested layers of transcendental functions. And it is not obvious how to enforce appropriate constraints on the domain of x and y. The straightforward approach of sampling of the conflict region and checking appropriate constraints would be both unsound (i.e., might indicate safety at times when collisions are possible) and computationally expensive. We discuss this problem further at the end of the section, and present a solution in Sects. 6 and 7 that is sound and computationally efficient.

To address the first two problems above, we divide the reachable envelope (domain) into different areas, according to the expressions that compute the minimum and maximum length paths, and then subdivide these pieces further to account for the conflict area and the motion possibilities of the other vehicle. The result is a covering of polygons where each is convex, and contains only a single uniform path length expression corresponding to earliest and latest timing for each vehicle at each point within the domain.

First, we create a sound overapproximation for the geometry of the conflict area using a set of convex polygonal sets in which the four different expressions for the timing computation (i.e., front and back edges of the waves for both vehicles) each belong to only one piece of the timing equations, Eqs. (28) and (29). For each vehicle with reachable envelope E, we create partitions using the domains $\{D_j\}$ of the timing equations matching the parameters of its kinematics, i.e., $F_j = \{D_j \cap E, E \backslash D_j\}$. Then, we create a refined partition R of the F_j domains that satisfies $\forall j$, $R \leq F_j$. Here the $\leq$ operator applied to partitions indicates that the left hand side partition is a refinement of the right hand side. Figure 13 shows what R looks like for the maneuver in Fig. 1.

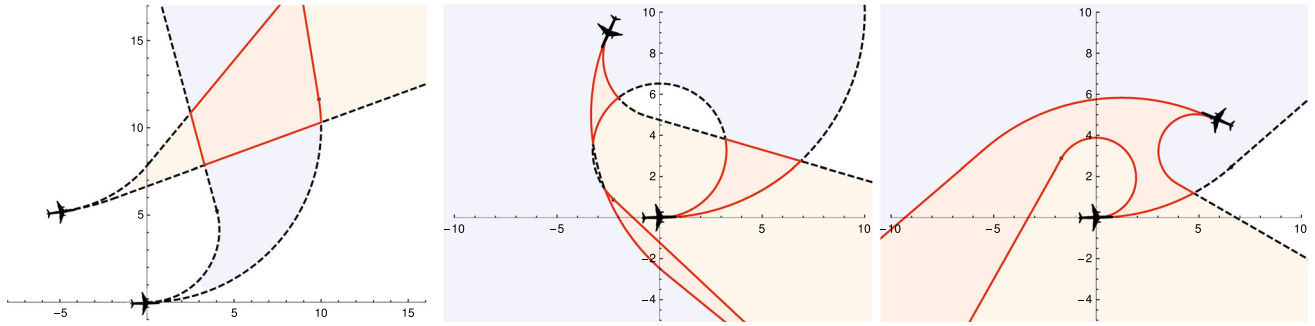
We then find a partition P, satisfying $P \leq (R^{\mathrm{i}} \cap E^{\mathrm{o}})$ and $P \leq (R^{\mathrm{o}} \cap E^{\mathrm{i}})$, that refines the partitions produced by the intersection of the reachable area partitions with the conflict area C. To create a sound polygonal overapproximation for this final partition P, we can follow the procedure above using polygonal approximations that describe each of the domains in the timing equation. In doing this, we relax the requirement that the sets subdividing the reachable area be partitions, allowing polygonal sets used for the initial part of the turn to overlap to ensure convexity. We describe these approximations and the adjustments they require in the next subsection.

For each polygon, we create a sound approximation of the edges of the two position waves within it using expressions that are circular or linear, so that we have an analytically tractable approach to solving for their overlap time. The



(a) $t = 8$ **(b)** $t = 16$ **(c)** $t = 24$

Fig. 11 Timing for a single scenario where two vehicles will make simultaneous, independent, non-deterministic turns with different turn-to-bearing assumptions. The reachable areas for the vehicles are shown at the moments in time indicated. The instantaneous reachable area is a propagating wave that moves outwards over time, encompassing the all possible positions that the vehicle may be located at each moment



(a) **(b)** **(c)**

Fig. 12 Geometry of three different scenarios where two vehicles will make simultaneous, independent, non-deterministic turns with different turn-to-bearing assumptions. The blue and yellow shading shows the reachable areas for the vehicles quantified over all time; it is equivalent to the union of instantaneous reachable areas shown in Fig. 11 for each future moment. The conflict area, where collisions might occur, is the intersection of the reachable areas, shaded in red (color figure online)

circular edge approximations are like the edges of circular ripples that result from dropping a pebble in a pond. They have the form:

$$(x - x_0)^2 + (y - y_0)^2 = (st + c)^2 \tag{35}$$

where s is the speed of propagation of the wavefront in the plane, c determines the timing of its initiation, and (x_0, y_0) is the point at which the wavefront originates.

The linear edge approximations are more like the straight edges of waves one might find sweeping across the open ocean. They have the form:

$$x \cos\phi + y \sin\phi = (st + c) \tag{36}$$

where c is the constant determining its position at specific times, and ϕ is the direction of propagation.

The purpose of creating these partitions and approximations is to break apart the conflict area C into polygonal regions, so the timing expressions are no longer piecewise in each polygon. Once these polygons are obtained, we can form polynomial—and in some cases, linear—optimization problems to find the collision interval.

These tailor-made approximations and refinements are each individually easier and more accurate to reason about because the timing expression is no longer piecewise inside each polygon. Table 1 shows which edge approximations we use for each part of the timing equations, referring to both a figure showing the domain, and the expression for the function's value being approximated.

Whenever these waves overlap, it means there exist trajectories that bring both aircraft into those positions at that moment, and thus create a collision. We can consider the timing intervals of each polygon independently, which gives us

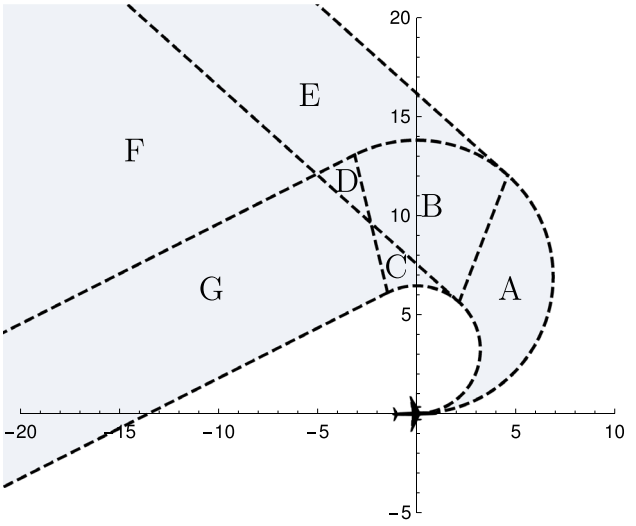


Fig. 13 A plot of four trajectories, with initial position $p_0 = (0, 0)$ at the origin, and initial orientation $\theta_0 = 0$ along the x-axis. These correspond to the limiting radii and bearings allowed for a non-deterministic left turn with $[\theta_\alpha, \theta_\beta] = [2.41, 3.62]$, and $[r_{\min}, r_{\max}] = [3.22, 6.89]$. All radii and bearings in between these limits are also possible

Table 1 Each region has a combination of edge shapes that describe the front and back edges of the waves

Region	Front edge	Back edge
A	Circular (28) Figure 8c	Circular (29) Figure 9c
B	Linear (28) Figure 8b	Circular (29) Figure 9c
C	Circular (28) Figure 8a	Circular (29) Figure 9c
D	Linear (28) Figure 8b	Linear (29) Figure 9b
E	Linear (28) Figure 8b	Circular (29) Figure 9a
F	Circular (28) Figure 8a	Circular (29) Figure 9a
G	Circular (28) Figure 8a	Linear (29) Figure 9b

some localization of where the collision may occur during that interval, or we can combine them together. The supremum and infimum of the union of the time intervals of wave intersection for each polygonal domain is a sound overapproximation of t_e and t_l, giving us the time interval in which collisions may occur during the encounter.

Using appropriate approximations and the approach we describe, we can analytically solve for the intersection of these envelopes and waves in an efficient manner, eliminating the quantification over time, and can establish the future safety of different horizontal maneuvers in unbounded time. In effect, our analysis allows us to ask and accurately answer: "If the aircraft find themselves in this configuration and the pilots restrict themselves to these turns, bearings, and speed limits during the encounter, can we guarantee they definitively do not collide?" Section 7 describes how we solve for the timing interval during which the waves overlap using these approximations.

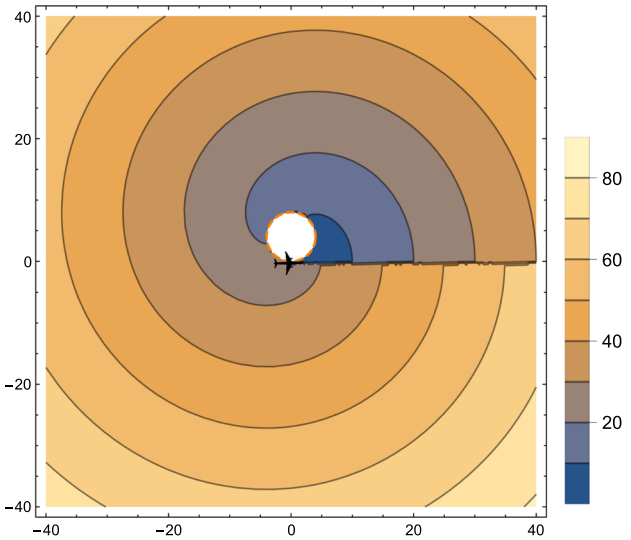


Fig. 14 Trajectory length for fixed-radius turn-to-bearing motion, for left turns with $r = 4$. Level sets of length are circle involutes

6 Sound approximation of collision timing equations

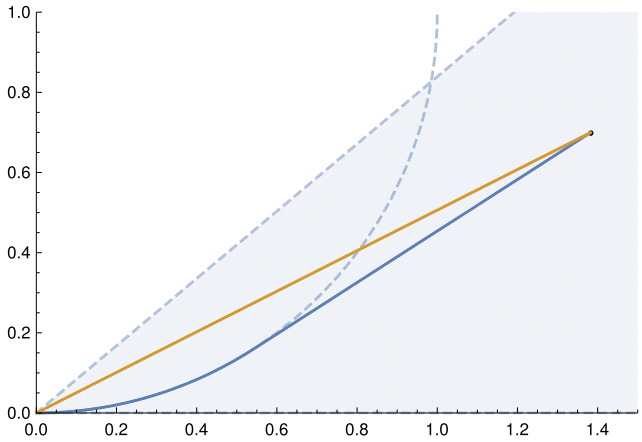
This section develops an approximation for path length whose accuracy can be controlled that allows analytical solution for and efficient computation of collision timing under non-deterministic, turn-to-bearing motion. Because of the approximation's simpler form, we will be able to use it to develop a solution for collision timing that is quantified over the envelope of future positions possible for that maneuver in each polygonal domain.

6.1 Fixed-radius turn-to-bearing approximation

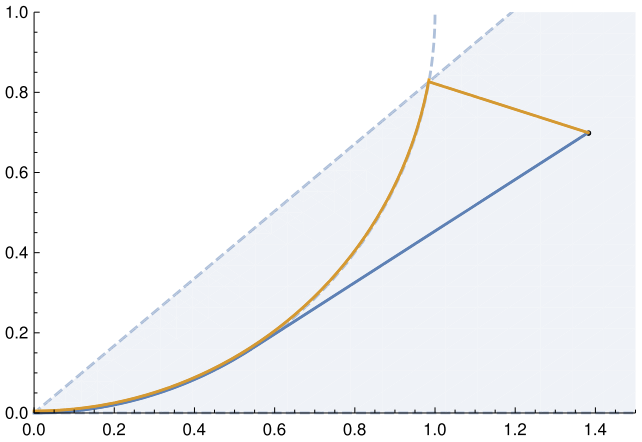
Points in the shaded area in Fig. 8a (or Fig. 9a) are reachable by left-turning paths whose radii span the full range of non-deterministic possibilities. The shortest (or longest) path to a point in the shaded area is the one with the minimum (or maximum) radius r_α(or r_β), so the function that expresses that bound matches the path length for vehicles that turn at exactly that radius, and leave at an appropriate tangent to reach the desired point. The expression is given by the second piece of Eq. (28) (or (29)).

Figure 14 is a contour plot where the color indicates the path length. Each path follows a circular arc of a particular radius, and then leaves the turn at a tangent to reach the destination point following a straight path thereafter. The plot has a cut, a discontinuity on the positive x-axis. The level set of path length for these pieces is a circle involute, which we will approximate in a limited area using a circular arc.

We prove:



(a) Path length underapproximation



(b) Path length overapproximation

Fig. 15 Turn-to-bearing paths (in blue) have length approximated by the length of the yellow paths. Path length for yellow paths can be expressed as functions of the length of their linear components, leading to contour plots with circular level sets. This approximation is valid within the domain indicated by the shaded wedge, whose angle in the approximating equations is ϕ_2 (color figure online)

Theorem 11 *For left turn-to-bearing motion with* $0 < \phi_2 < 2\pi$*, the length of a path starting from the origin initially oriented in the direction of the positive x-axis and arriving at a point* (x, y) *following one-turn-to-bearing motion with a fixed radius is bounded by*

$$\sqrt{x^2 + y^2} \leq L(x, y, \Theta(x, y, r), r) \leq r\phi_2 + \sqrt{(x - w_x)^2 + (y - w_y)^2} \tag{37}$$

where $w_x = r\sin(\phi_2)$ *and* $w_y = r(1 - \cos(\phi_2))$.

We know that the shortest path between two points in a Cartesian plane is a line segment between those points. We can use this to create both an underapproximation of the shortest path and an overapproximation of the longest path for this region.

Consider a turn-to-bearing path starting from the origin, traveling in a circular arc, leaving the arc at a tangent at point T, and traveling in a straight line thereafter to reach (x, y). The length of the line segment from the origin to (x, y) is a lower bound for the path distance, since the true path does not follow a straight line to reach (x, y). For an upper bound, we can use the length of a path that follows the circular arc but continues past T to another point U on the arc and then follows a straight (non-tangent) line from U to (x, y). This is longer than the turn-to-bearing path, because both paths stay together until point T, and from that point the turn-to-bearing path follows a straight line to (x, y). Every path that diverges at or after the tangent point must thus be longer. Figure 15a, b illustrates this strategy.

The domain of Figs. 8a and 9a is nearly a polygonal boundary, but not quite. We create a polygonal domain for our approximation of path distance in the second pieces of Eqs. 28 and 29 for a vehicle starting at the origin using simple linear boundaries. We define an area

$$S = \{(x, y) \mid y \geq 0 \wedge \text{atan}(y, x) \leq \phi_2\} \tag{38}$$

This area is equivalent to a domain

$$G_v^R(x, y) = y \geq 0 \wedge (\cos(\phi_2)y \leq \sin(\phi_2)x) \tag{39}$$

This wedge-shaped domain can be rotated and translated to allow us to approximate other parts of the circle. If we want to adjust it so that the initial bearing is $2\phi_1$, we can translate the domain so the vertex is at $(r\sin(\phi_1), r(1-\cos(\phi_1))$, and rotate it so the clockwise-most linear boundary is tangent to the circle at that point.

6.2 Fixed-bearing turn-to-bearing approximation

There are a range of possible turn-to-bearing trajectories that reach from the origin to each point in the shaded area in Fig. 8b. The minimum path length for each point–given by the second piece of Eq. (28)—corresponds uniformly to a trajectory of whose final bearing θ_β is at the end of the allowable range, and the radius that achieves that bearing, which varies depending on the point.

Turn-to-bearing trajectories reaching shaded points in Fig. 9b have a similar property. The maximum possible path length to reach these points—given by the second piece of Eq. (29)—also corresponds uniformly to trajectories of fixed final bearing (this time θ_α) at the other end of the allowable range, and the radius that achieves that bearing, which differs depending on the point.

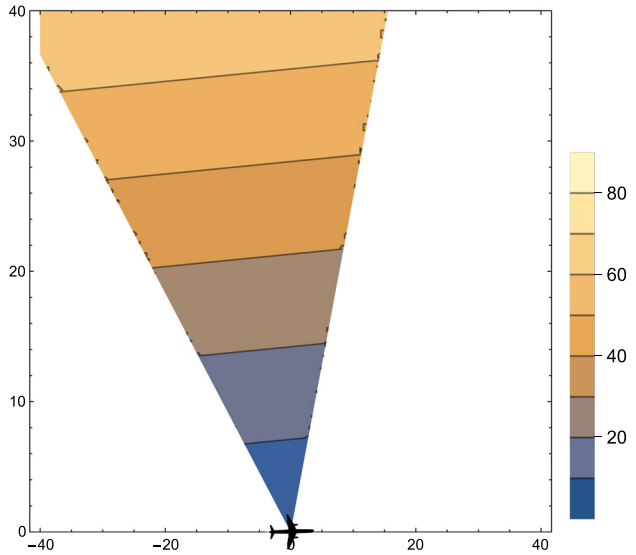


Fig. 16 Trajectory length for fixed-bearing turn-to-bearing motion, for left turns with final bearing $\theta = 2.4$. Level sets of length are lines

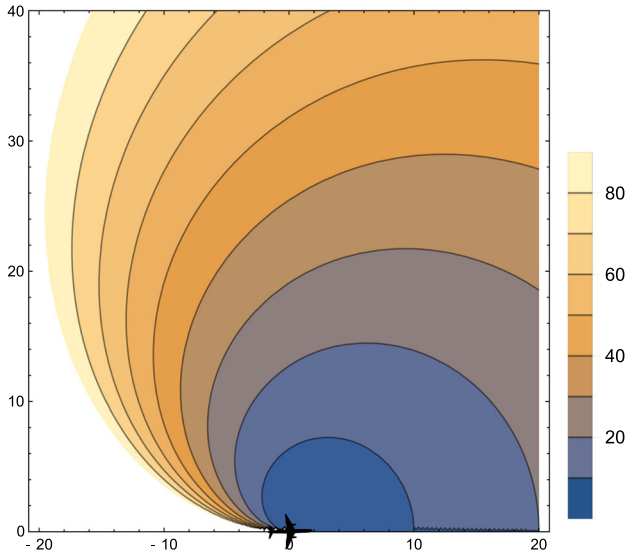


Fig. 17 Trajectory length for a left-turning circular arc. Level sets of length are cardioids

The level set of path length in the plane for these pieces is a line. Thus, the boundary of the area that contains the non-deterministic possibilities of our motion is a linear wave, an isoline in the plane with a fixed orientation that is moving over time.

Figure 16 shows the exact length of the turn-to-bearing path reaching each point in the plane of motion when those paths are constrained to end with uniform orientation. There is only one such path that reaches each point for a particular choice of bearing, and each point is colored according to the path length. We do not consider paths that end during the initial turn unless their bearing matches the motion we are analyzing, so although some points to the right of this wedge are traversed by these paths, the paths are circular at that stage, and their lengths are not shown in this figure. Points on the left side of the wedge are not reachable or traversable by this type of motion. For this section, each contour is linear and this type of motion does not require an approximation to express the path lengths at each point. We prove:

Theorem 12 *For left turn-to-bearing motion, path distance starting from the origin with orientation* 0 *(facing the direction of the positive x-axis) and arriving at a point* (x, y) *with orientation* θ *is given by*

$$L(x, y, \theta, R(x, y, \theta)) = x\left(\cot\left(\frac{\theta}{2}\right)\theta - 1\right) + y\left(\cot\left(\frac{\theta}{2}\right) - \frac{\cos(\theta)}{1-\cos(\theta)}\theta\right) \quad (40)$$

The key insight here is that for each region, the angle θ required to find the minimum or maximum path length is constant, and thus the expression for path distance has the form of a plane wave as given by Eq. (36). A plot of path lengths produces linear level sets, which can be interpreted as a linear wavefront in the plane.

The domains of the second pieces of Eqs. (28) and (29) are each convex, open sets with linear boundaries. They need no approximation to ensure convexity and can be subdivided into convex polygons to localize the timing of potential future collisions within them.

6.3 Turning approximation

Each turn-to-bearing maneuver begins with a turn that follows a circular arc. There is only one path that leads from the origin to each of the points in the shaded area in Fig. 8c, so that path is the shortest length path possible. The path length is given by the third piece of Eq. (28). The domain of Fig. 9c is a superset of Fig. 8c, where some of the points have more than one way to approach them. The longest possible path to reach each of these points also follows a circular arc, given by the same expression shown in the second piece of Eq. (29).

The level set of path length for these pieces is a cardioid, which we will approximate using a circular wave.

Figure 17 shows the lengths of turn-to-bearing paths starting at the origin as shown, reaching each point in the plane of motion, when each path is following a circular arc from beginning to end.

The key insight here is that if we examine the length of circular paths that end at points whose location is at a constant angle in the plane (and thus have a constant final orientation), we can create a function that has the form of a circular wavefront that matches the path lengths along that ray. We prove:

Lemma 1 (Circular path length) *The length of a left-turning circular path equals the distance from the origin scaled by a factor of* $\mathrm{sinc}(x) = \sin(x)/x$*:*

$$r_m\theta_m = \frac{\sqrt{x^2+y^2}}{\mathrm{sinc}(\frac{\theta_m}{2})}. \tag{41}$$

If we solve the implicit equation of a circle for the radius, we will find that it is proportional to $\sqrt{x^2+y^2}$. We are using the radius to compute the path lengths, so the bounds will have this term in them as well. Because $\mathrm{sinc}(\cdot)$ decreases monotonically over the interval $(0, \pi)$, we know that a circular function that matches one angle is a lower bound for angles with the same radii above it.

We define an area

$$\begin{aligned} S = \{(x, y) \mid \exists\, r \in [r_1, r_2], \theta \in [\theta_1, \theta_2], \\ x = r\sin(\theta) \wedge y = r(1-\cos(\theta))\} \end{aligned} \tag{42}$$

whose boundary is limited by circular turning arcs with different radii and straight lines radiating from the starting point at different angles, as shown in Fig. 18a. We can create circular wavefronts centered at the origin that match the path lengths at the straight edges, and serve as upper and lower bounds for all of the points within the shaded area.

These boundaries define a closed region, and the reachable area of circular turning motion can be tiled by these sets. In Theorem 13, we prove upper and lower bounds for path length in this type of region.

Theorem 13 *For left turn-to-bearing motion with* $0 < \theta_1 < \theta_m < \theta_2 < 2\pi$*, the length of a path starting from the origin initially oriented in the direction of the positive x-axis and arriving at a point* (x, y) *following a circular arc is bounded by*

$$\frac{\sqrt{x^2+y^2}}{\mathrm{sinc}\left(\frac{\theta_1}{2}\right)} \le L(x, y, \theta_m, r_m) \le \frac{\sqrt{x^2+y^2}}{\mathrm{sinc}\left(\frac{\theta_2}{2}\right)}. \tag{43}$$

The partition has elements of the form of S, but with different parameters r_1, r_2, θ_1, and θ_2. Elements have curved boundaries and are not convex, so we create a polygonal overapproximation of these elements which serves as the domain. Figure 18b shows this approximation in the first quadrant, where the region is defined by

$$\begin{aligned} G^C(x, y) = \big(&\cos(\theta_1/2)(y-\nu_y) \ge \sin(\theta_1/2)(x-\nu_x) \wedge \\ &\cos(\theta_2/2)(y-\iota_y) \le \sin(\theta_2/2)(x-\iota_x) \wedge \\ &(\iota_x-\nu_x)(y-\nu_y) \le (\iota_y-\nu_y)(x-\nu_x) \wedge \\ &\cos(\theta_1)(y-\upsilon_y) \ge \sin(\theta_1)(x-\upsilon_x) \wedge \\ &\cos(\theta_2)(y-\omega_y) \ge \sin(\theta_2)(x-\omega_x)\big), \end{aligned} \tag{44}$$

where

$$\begin{aligned} (\nu_x, \nu_y) &= r_1(\sin(\theta_1), (1-\cos(\theta_1))) \\ (\iota_x, \iota_y) &= r_1(\sin(\theta_2), (1-\cos(\theta_2))) \\ (\upsilon_x, \upsilon_y) &= r_2(\sin(\theta_1), (1-\cos(\theta_1))) \\ (\omega_x, \omega_y) &= r_2(\sin(\theta_2), (1-\cos(\theta_2))). \end{aligned}$$

When this approximation is used instead of a partition for refinement, the other areas that also are part of the refinement may be duplicated twice, and need to be evaluated with more than one wavefront combination.

This approximation can be used to safely represent positions for the highly nonlinear area at the beginning of the circular turn.

7 Sound solution for collision timing between two turning vehicles

In Sects. 5 and 6, we showed how to subdivide the reachable region of turn-to-bearing maneuvers into a covering with polygonal regions. Recall that the polygonal regions are sound overapproximations of refinements of the conflict area C, the region where two vehicles could both potentially reach.

In each of these polygonal regions, the vehicle location over time is bounded by the region, which is fixed in time, and linear or circular curve segments that propagate in time. These boundaries move and form the front and back edges of propagating waves—the front edge bounds the location of the vehicle in the longest paths that could be followed for a given time and the back edge bounds the location by the shortest paths that could be followed.

Guaranteeing the absence of collisions now equates to ensuring that, for each of these polygonal regions, there are no locations simultaneously contained by the front and back edges of the waves of position possibilities for both vehicles at the same time.

We represent each region as a polygon, which is an intersection of finitely many half-planes. A compact notation for the region is

$$\mathcal{P} = \left\{\mathbf{x} \in \mathbb{R}^2 \mid A\mathbf{x} \le \mathbf{b}\right\} \tag{45}$$

where $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^2$ and $\mathbf{b} \in \mathbb{R}^k$. We do not require the region be bounded.

The overlap property Evaluating timing for positions to overlap is necessary to establish collision possibilities. This was first discussed for individual points in Theorem 6. A necessary condition for there to be an overlap between the

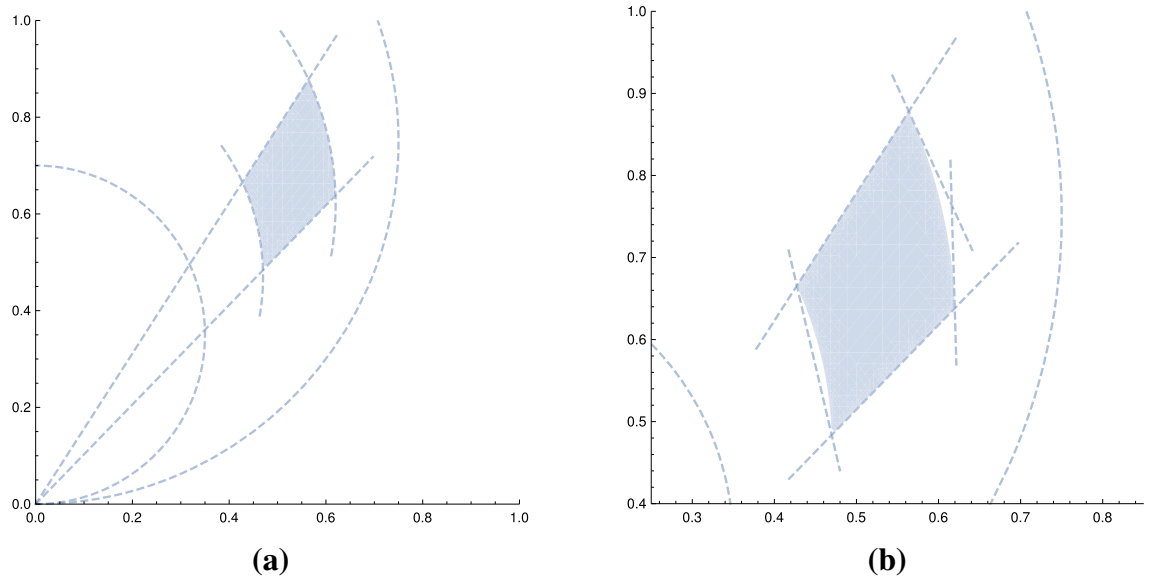


Fig. 18 Boundaries of a pixel that can be used to tile the reachable area for motion in a circular turn

time that ownship and intruder are in a particular point in x, y space is that the waves that constrain where each vehicle is must overlap at that point for some time t. If there exists a point and time that is within the polygon and within the wavefronts of both the ownship and the intruder at a common time, then a collision is possible at that place and future time.

In Sect. 4.1, Eqs. (17) and (18) define the earliest and latest times that the vehicle could reach point p. Our objective is to compute over the polygonal region, defined in (45), the earliest and latest times where the intersections of the ownship and intruder waves can meet. The intersection of the ownship and intruder waves form another wave, a set of points at each instant in time that for that future instant might produce a collision. We will call the wave resulting from the intersection the conflict wave. Since the over- and under-approximations for vehicle locations are sound in the polygonal region of interest, if the conflict wave intersects any part of our polygon, then that denotes a potential collision. The overall earliest and latest times for collision in this region will be denoted $t_e(\mathcal{P})$ and $t_l(\mathcal{P})$.

7.1 The case with purely linear wavefronts

We first treat the case where the front and back edges of both ownship and intruder position waves are all described with the linear boundary given in Eq. (40). This case happens in region D (see Fig. 13 and Table 1). The front and back of the linear edges are formed using the upper and lower bounds on the vehicle speed, s_β and s_α. Each "wavefront" is defined by a tangent and a velocity of propagation. The ownship front wave is

$$\begin{bmatrix} f_1^{\mathrm{o}} & f_2^{\mathrm{o}} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = s_\beta^{\mathrm{o}} t, \tag{46}$$

and the ownship back wave is

$$\begin{bmatrix} b_1^{\mathrm{o}} & b_2^{\mathrm{o}} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = s_\alpha^{\mathrm{o}} t. \tag{47}$$

The intruder's front wave, which starts at point $\mathbf{q} = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^{\mathrm{T}}$, is

$$\begin{bmatrix} f_1^{\mathrm{i}} & f_2^{\mathrm{i}} \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \mathbf{q} \right) = s_\beta^{\mathrm{i}} t, \tag{48}$$

and the intruder back wave is

$$\begin{bmatrix} b_1^{\mathrm{i}} & b_2^{\mathrm{i}} \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \mathbf{q} \right) = s_\alpha^{\mathrm{i}} t. \tag{49}$$

To relate these coefficients to those in Eq. (40), we define

$$f_1^{\mathrm{o}} = \cot(\theta_\alpha/2)\theta_\alpha - 1 \tag{50}$$

$$f_2^{\mathrm{o}} = \cot(\theta_\alpha/2) - \frac{\cos(\theta_\alpha)}{1 - \cos\theta_\alpha}\theta_\alpha \tag{51}$$

$$b_1^{\mathrm{o}} = \cot(\theta_\beta/2)\theta_\beta - 1 \tag{52}$$

$$b_2^{\mathrm{o}} = \cot(\theta_\beta/2) - \frac{\cos(\theta_\beta)}{1 - \cos\theta_\beta}\theta_\beta \tag{53}$$

where the θ_α and θ_β parameters correspond to the bounds on angle that apply to this particular ownship refinement.

We then do the same for the intruder front and back wave coefficients.

We now write the full set of constraints over the search space of feasible, x, y, and t. The joint set of constraints are:

$$\begin{bmatrix} f_1^{\mathrm{o}} & f_2^{\mathrm{o}} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq s_\beta^{\mathrm{o}} t \tag{54}$$

$$\begin{bmatrix} b_1^{\mathrm{o}} & b_2^{\mathrm{o}} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \geq s_\alpha^{\mathrm{o}} t \tag{55}$$

$$\begin{bmatrix} f_1^{\mathrm{i}} & f_2^{\mathrm{i}} \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \mathbf{q} \right) \leq s_\beta^{\mathrm{i}} t \tag{56}$$

$$\begin{bmatrix} b_1^{\mathrm{i}} & b_2^{\mathrm{i}} \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \mathbf{q} \right) \geq s_\alpha^{\mathrm{i}} t \tag{57}$$

$$A\mathbf{x} - \mathbf{b} \leq 0. \tag{58}$$

With purely affine inequality constraints, the problem of finding the earliest and latest times of a potential collision can be solved by minimizing or maximizing t over these constraints. For example, $t_{\mathrm{e}}(\mathcal{P})$ is the solution to the optimization problem

$$\begin{aligned} &\text{minimize } t \\ &\text{subject to } (54)\text{–}(58), \\ &\qquad t \geq 0, \end{aligned} \tag{59}$$

over the variables x, y, and t. If the problem (59) is infeasible, then collision is not possible in $\mathcal{P}$. Similarly, we can find $t_{\mathrm{l}}(\mathcal{P})$ by forming a maximization problem.

7.2 Case including only circular wavefronts

We now consider the cases where the front and back waves of ownship and intruder involve only circular waves. These would correspond to intersections of polygons where both the ownship and the intruder are in regions A, C or F (see Fig. 13 and Table 1). As described in Sect. 6, these turning sections are also subdivided (or tiled) into smaller regions and then overapproximated as polygons (Fig. 18b).

The timing bounds for these circular wavefronts are given in Theorem 13, Eq. (43). They have the form:

$$(x - x_0)^2 + (y - y_0)^2 = (vt + c_0)^2 \tag{60}$$

where v is the speed of propagation of the wavefront in the plane, c_0 determines the timing of its initiation, and (x_0, y_0) is the point at which the wavefront originates. We note that the front and back waves may have different parameters, x_0, y_0, and c_0, and will be denoted with subscript f for the front wave and with b for the back wave.

With appropriate coefficients for the ownship and intruder front and back wave edges, the constraints for a collision become the following.

$$(x - x_f^{\mathrm{o}})^2 + (y - y_f^{\mathrm{o}})^2 \leq (s_\beta^{\mathrm{o}} t + c_f^{\mathrm{o}})^2 \tag{61}$$

$$(x - x_b^{\mathrm{o}})^2 + (y - y_b^{\mathrm{o}})^2 \geq (s_\alpha^{\mathrm{o}} t + c_b^{\mathrm{o}})^2 \tag{62}$$

$$(x - x_f^{\mathrm{i}})^2 + (y - y_f^{\mathrm{i}})^2 \leq (s_\beta^{\mathrm{i}} t + c_f^{\mathrm{i}})^2 \tag{63}$$

$$(x - x_b^{\mathrm{i}})^2 + (y - y_b^{\mathrm{i}})^2 \geq (s_\alpha^{\mathrm{i}} t + c_b^{\mathrm{i}})^2 \tag{64}$$

$$A\mathbf{x} - \mathbf{b} \leq 0 \tag{65}$$

The four constraints (61)–(64) are quadratic in the decision variables x, y, and t, but they are not convex constraints. In fact, only the linear case results in a convex linear program. All other cases can be solved, in general, using polynomial optimization. Exact algorithms based on the Cylindrical Algebraic Decomposition (CAD) [3], such as [5,15], as well as approximation techniques involving moments [14], or sums-of-squares and semidefinite relaxations [19,22] can be used to determine the timing boundaries on each region.

7.3 Visualizing the solutions of wave intersections

Using informal, geometric, and CAD-inspired arguments, we present an approach to evaluating collision safety and solving for collision timing. We leave it as future work to formalize this proof. In Fig. 11 we saw that at each future moment in time, each vehicle has an area in which it will be found, encompassing the uncertainty in its motion between the present and that moment. This area is a propagating wave of position possibilities that changes shape and moves forward as time progresses. It is bounded by the curves that define the reachable area, and irregularly shaped front and back edges that move orthogonally to the reachable area boundaries, expanding over time. We use the path distance approximations from Sect. 6 to represent the front and back edges of the wave. Each encounter has two waves and four different edges, i.e., the leading and trailing edges of the area describing the possible positions for each vehicle.

In the previous section, we created a covering of convex polygons, each with a positional wave whose edges (level sets of path length) are represented by simple polynomials of at most order two. This breaks the problem up into a set of simpler problems. In this section, we will focus on the problem of solving for wave intersection within one polygonal region.

For now, we eliminate the polygonal boundaries so that we can more clearly see the geometry and timing of overlap of different position possibility waves. Later we will add the polygonal boundaries back into the problem.

In Sect. 4.1, we developed the overlap criteria for pointwise collision, namely that the time intervals when two vehicles might arrive at a point must overlap for there to be a

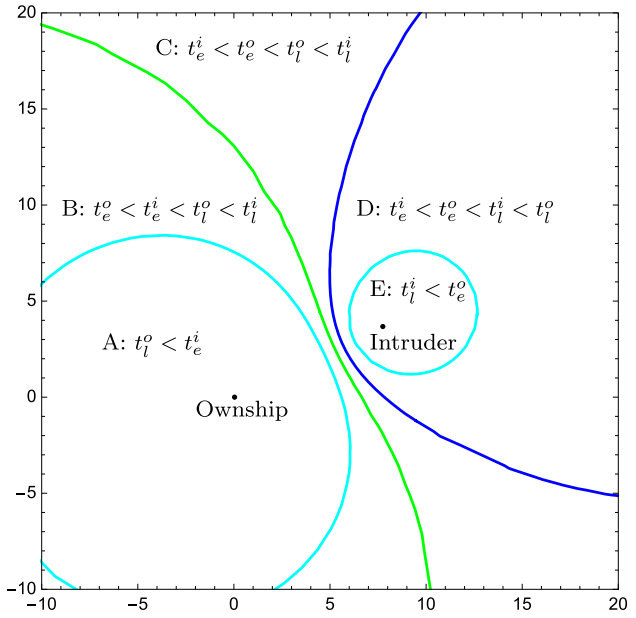


Fig. 19 Loci of intersections between leading and trailing edges of circular wavefronts are plotted together for a specific, example geometry. The loci impose an ordering of earliest and latest arrival times for each vehicle for the points in each region of the plane

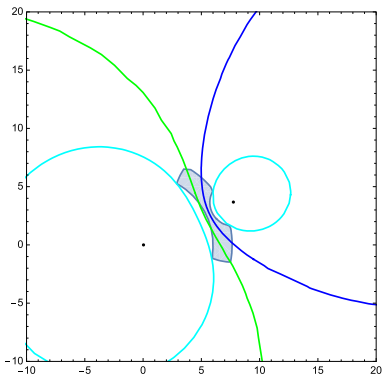
collision. We can now apply that criteria to the geometry of uncertain position waves.

One key insight is that we can order the arrival of wave edges at each point by creating loci of the moving edges of the position waves. Each locus is a curve consisting of all the points of intersection over time between two different position wave edges, one from each vehicle. The loci divide the horizontal plane into regions that identify the order the earliest arrival and latest departure times for each vehicle at each point in the region.

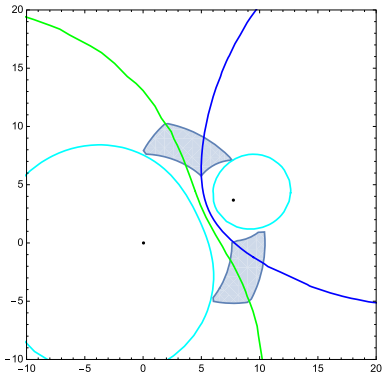
We consider an example encounter shown in Fig. 19 between two aircraft whose positions are localized by annuli whose outer and inner curves are expanding circular waves. The green curve is the locus of the two leading edges; the blue curve is the locus between trailing edges; the cyan curves are the loci of one leading and one trailing edge from each vehicle. Points labeled with vehicle designations mark the center of the circular waves. The regions are labeled and the ordering for points in each region imposed by the annuli are indicated.

In regions A and E of Fig. 19, there is no possibility of collision; the conflict wave never enters these regions. For regions B, C, and D, we can choose any point, and identify the earliest and latest times that the vehicles can collide at that point by looking at the ordering, and calculating the beginning and ending of the collision timing interval using Eqs. (17)–(18) and Eqs. (28)–(29).

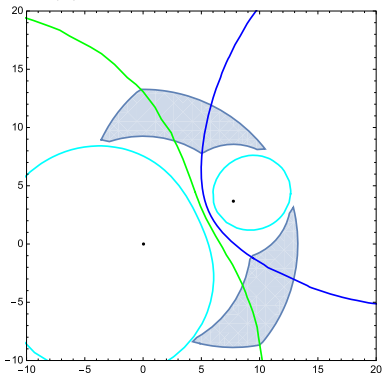
The intersection of the position waves for two vehicles is an area that moves over time and represents where collisions



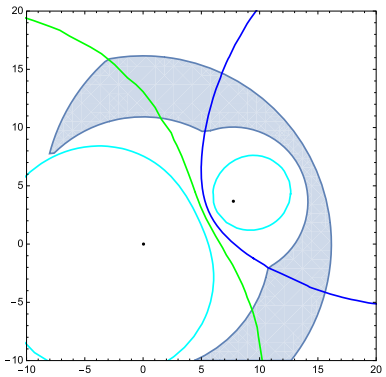
(a) Conflict wave at $t = 6.1$



(b) Conflict wave at $t = 10.5$



(c) Conflict wave at $t = 14.9$



(d) Conflict wave at $t = 19.3$

Fig. 20 Using the example geometry, we plot areas in which the aircraft can collide at different instants in time–the conflict wave

may occur at each moment; we call this the conflict wave. Figure 20 shows the conflict wave for our example geometry at different snapshots in time by shading the area of overlap between the areas of position possibilities. Our position wave edges, and thus the loci are based on the circular edge approximations from Sect. 6, so the shaded conflict waves are also approximations. This approach also works if one or more of the position-wave edges is a linear boundary.

With a visual understanding of conflict waves, we can reintroduce the convex polygonal boundaries we created for our approximations.

To compute the earliest or latest time that a collision is possible, we do not need to evaluate every single point in a polygonal area; we can compute earliest and latest collision times, i.e., Eqs. (33)–(34), by looking at the timing of the collision wave with respect to a finite set of points that we call critical points.

This approach corresponds to something akin to Cylindrical Algebraic Decomposition, which can be used to find extrema of polynomials on sets whose boundaries are polynomial equations and inequalities. CAD can be applied algorithmically, using techniques such as [5]; however, the order of decomposition is often important, and it may be necessary to ensure that the polynomial coefficients are rational.

7.3.1 Computing the earliest conflict time

To find the earliest possible collision time (and the location of that potential collision), we need to evaluate when and where the collision wave first touches the area of the polygon. Figure 21 uses the example geometry from the previous section, adding a triangle to represent the convex polygon that defines the boundary of this area, and plotting the critical points for computing t_e in red. One of these critical points will represent the point where the earliest possible collision may occur.

Each panel illustrates a type of critical point by moving the triangle to show a geometry and earliest moment at which that critical point is the first one in the polygon that comes in contact with the conflict wave. Figure 21a shows that there is a special case of a critical point that is not on the edges of the polygon at the location of the point where the leading wavefronts of each of the vehicles first touch, if that point is in the collision area. This is where the conflict wave first "appears." Figure 21b shows that the vertices of the polygon are also critical points—this is a consequence of the convexity of the polygon and front edge of the conflict wave. Figure 21c shows that points where the loci intersect the edges of the polygon are also critical points and may be the earliest locations at which the conflict wave comes into contact with the polygon. Figure 21d shows that points on a segment of the polygon that are tangent to the leading wavefront are also critical points.

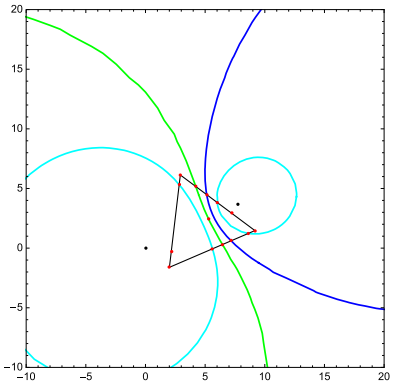
Critical points can be identified automatically, and once this is done, evaluation of the earliest time of collision is a matter of evaluating the earliest time of collision for each point that contacts the conflict wave, and choosing the minimum time overall.

7.3.2 Computing the latest conflict time

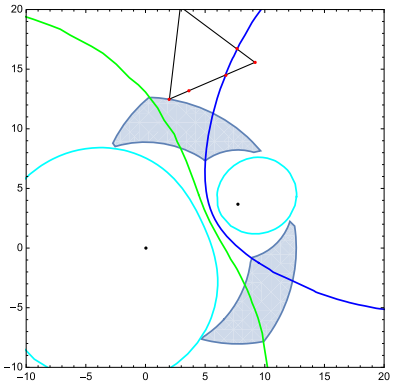
To find the latest possible collision time (after which there can be no further possibility of collision) and location, we need to calculate when and where the conflict wave last touches the area of the polygon. Figure 22 uses the example geometry from the previous section, again adding a triangle to represent the convex polygonal that defines the domain of our approximations, and plotting critical points in red. One of these critical points will be the point where the latest possible collision may occur.

As before, each subfigure illustrates a type of critical point by moving the triangle to show a geometry and timing where that critical point is the last one the polygon comes into contact with, before the conflict waves moves away entirely. The types of critical points used in computing t_l are a subset of the types used for computing t_e. Figure 22a shows that the vertices of the polygon are critical points. Figure 22b shows that points where the loci intersect the line segments that make up the polygon are also critical points. In some cases, the polygon that contains the approximation is open, i.e., the edges do not create a closed circuit. In these cases, the polygonal boundary has rays at the end of it extending to infinity. We use the angles of these rays to identify a range of angular directions by which we can approach infinity and still remain in the polygon. We then consider the angular position of the loci in the limit as we approach infinity, and the ordering of arrival and departure times at each angle as we approach infinity. By comparing these angular intervals, we can identify angular intervals and angular points in the region of validity where collisions may or may not occur. If a collision may not occur, then we can safely guarantee that t_l must have some finite value. If there exist angular directions where a collision may occur, then we cannot guarantee safety, there can be collisions that occur in unbounded time. In this case, these angles indicate that there is no upper bound to the interval in which collisions may occur; we can say the latest collision time is at infinity.

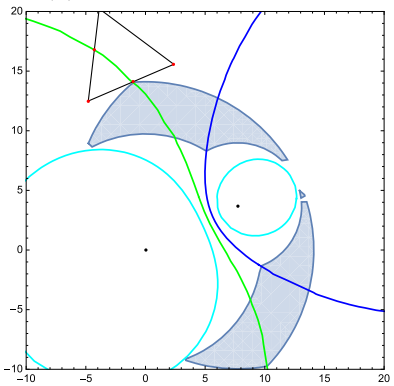
Once all critical points are identified, evaluation of the latest collision time is a matter of evaluating the latest collision time for each point that contacts the conflict wave, and choosing the maximum time overall. If we have an unbounded region with a latest collision time that is also unbounded, then the correct latest collision time is infinity.



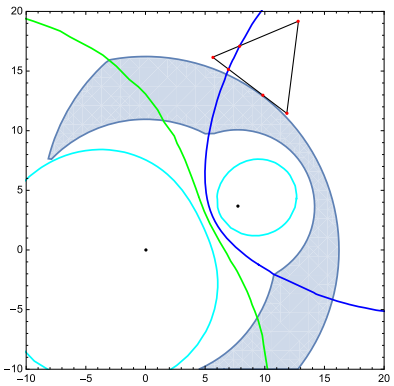
(a) Conflict wave at $t = 19.3$



(b) Conflict wave at $t = 14.9$

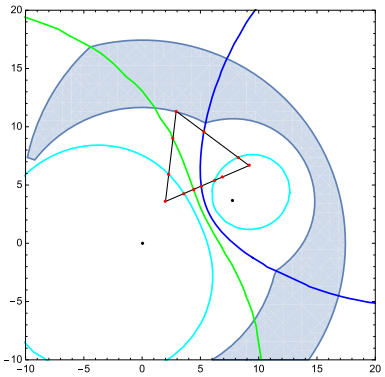


(c) Conflict wave at $t = 10.5$

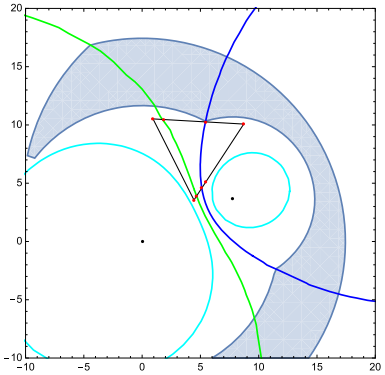


(d) Conflict wave at $t = 6.1$

Fig. 21 Types of critical points for computing t_e



(a) Conflict wave at $t = 19.3$



(b) Conflict wave at $t = 14.9$

Fig. 22 Types of critical points for computing t_l

8 Future work

There are many ways to improve on this work, some of which we have begun to explore.

The most straightforward continuation of this work is to move it closer to practical application. To this end, we have implemented the algorithm in the Julia language to compute horizontal conflict intervals and plan to synthesize a safety controller that uses it based on [13]. We are also developing a formalization and proofs of correctness specifically for the controller synthesis, which would allow us to extract a correct-by-construction controller implementation. We have started to experiment with the calculation in a few different simulations: to evaluate the safety of maneuvers made by autonomous boats; and also as part of a mixed vertical/horizontal collision avoidance system for aircraft, computing horizontal conflict intervals for [11].

Another enhancement would be to explore the tradeoff in our approximations between accuracy and computational efficiency. The calculations and maneuvers based on them are provably safe, but coarse approximations lead to maneuvers that are more conservative and might restrict the system unnecessarily, and tight approximations increase the computational burden. We feel that the approximations we have

described strike a good balance, but different applications may require more accuracy or more computational speed from the analysis. The approach in this paper could be used to adjust the fineness/coarseness of the covering polygons and thus set the tradeoff according to requirements of a particular application.

To make the system more useful in application, we have experimented with representing position uncertainty in the vehicles—which could represent sensor error or unexpected variations in future trajectories—by expanding polygons to contain shapes created by convolving a circular disk with each of the position waves, and by extension, the envelopes and collision waves. This can be done with minimal additional computational effort, but needs further proof and formal verification.

In addition, we are pursuing methods for constraining the learned policies of safe neural network controllers using the safety predicates from this work, such as in [6]. The safety predicates developed here may be used to guide the training of such networks, verify the correctness of the network policies, and could someday be used directly in the optimization of such neural controllers.

While the model developed in this work applies only to turn-to-bearing kinematics, due to the non-determinism that we incorporate there is a family of trajectories that are also encompassed by these proofs. The extension representing position uncertainty allows additional flexibility. Future research could include characterizing this family of trajectories.

9 Conclusion

In this work, we have created a formally verified library that describes uncertain turn-to-bearing kinematics and allows us to reason about the timing of such maneuvers without approximation. The representation allows non-determinism in all turn parameters by quantifying over state variables.

We have applied the library to compute timing intervals during which the intersecting turns of two vehicles might collide. These timing computations can be used to determine whether two aircraft will ever travel close enough to each other (under the range of assumed kinematics) to be in horizontal conflict, and, if so, what the earliest and latest times of the horizontal conflict can be. By combining horizontal conflict timing with reasoning about the vertical separation of aircraft, we can ensure that the aircraft are not simultaneously in horizontal and vertical conflict and guarantee the absence of collisions. To find the horizontal conflict time range, we first developed expressions of time intervals without approximation, for a given point accounting for non-deterministic horizontal maneuvers for each aircraft. We then applied the library to create approximations of the position waves that are useful for calculating the intersection between two position waves in subregions of the envelope in which collisions may occur. Finally, we showed a method for tiling and fitting sound polygonal approximations of each subregion, resulting in computationally efficient methods for solving for the earliest and latest horizontal conflict times.

References

1. Abhishek, A., Sood, H., Jeannin, J.B.: In: Formal verification of braking while swerving in automobiles. Association for computing machinery. In: Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control, HSCC '20. Association for Computing Machinery, New York, NY, USA (2020)
2. Boldo, S., Lelay, C., Melquiond, G.: Coquelicot: a user-friendly library of real analysis for Coq. Math. Comput. Sci. **9**(1), 41–62 (2015)
3. Collins, G.E.: Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In: Automata Theory and Formal Languages, pp. 134–183. Springer (1975)
4. Cons, M.S., Shima, T., Domshlak, C.: Integrating task and motion planning for unmanned aerial vehicles. Unmanned Syst. **02**(01), 19–38 (2014). https://doi.org/10.1142/S2301385014500022
5. Fotiou, I.A., Rostalski, P., Parrilo, P.A., Morari, M.: Parametric optimization and optimal control using algebraic geometry methods. Int. J. Control **79**(11), 1340–1358 (2006)
6. Genin, D., Papusha, I., Brulé, J., Young, T., Mullins, G., Kouskoulas, Y., Wu, R., Schmidt, A.: Formal verification of neural network controllers for collision-free flight. In: 14th International Workshop on Numerical Software Verification (NSV) (2021)
7. Isaiah, P., Shima, T.: A task and motion planning algorithm for the Dubins travelling salesperson problem. IFAC Proc. Vol. **47**(3), 9816–9821 (2014). (**19th IFAC World Congress**)
8. Jeannin, J., Ghorbal, K., Kouskoulas, Y., Schmidt, A., Gardner, R., Mitsch, S., Platzer, A.: A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system. STTT **19**(6), 717–741 (2017). https://doi.org/10.1007/s10009-016-0434-1
9. Jeyaraman, S., Tsourdos, A., Żbikowski, R., White, B.A.: Formal techniques for the modelling and validation of a co-operating uav team that uses dubins set for path planning. In: Proceedings of the 2005, American Control Conference, 2005, vol. 7, pp. 4690–4695 (2005)
10. Kouskoulas, Y., Genin, D., Schmidt, A., Jeannin, J.: Formally verified safe vertical maneuvers for non-deterministic, accelerating aircraft dynamics. In: Ayala-Rincón, M., Muñoz, C.A. (eds.) Interactive Theorem Proving—8th International Conference, ITP 2017, Brasília, Brazil, September 26–29, 2017, Proceedings, pp. 336–353. Springer (2017)
11. Kouskoulas, Y., Genin, D., Schmidt, A., Jeannin, J.: Formally verified safe vertical maneuvers for non-deterministic, accelerating aircraft dynamics. In: Ayala-Rincón, M., Muñoz, C.A. (eds.) Interactive Theorem Proving—8th International Conference, ITP 2017, Brasília, Brazil, September 26–29, 2017, Proceedings, Lecture Notes in Computer Science, vol. 10499, pp. 336–353. Springer (2017)
12. Kouskoulas, Y., Machado, T.J., Genin, D.: Formally verified timing computation for non-deterministic horizontal turns during aircraft collision avoidance maneuvers. In: ter Beek, M.H., Nickovic, D. (eds.) Formal Methods for Industrial Critical Systems—25th International Conference, FMICS 2020, Vienna, Austria, September

2–3, 2020, Proceedings, Lecture Notes in Computer Science, vol. 12327, pp. 113–129. Springer (2020)
13. Kouskoulas, Y., Schmidt, A., Jeannin, J.B., Genin, D., Lopez, J.: Provably safe controller synthesis using safety proofs as building blocks. In: IEEE 7th International Conference on Software Engineering Research and Innovation, CONISOFT 2019, October 23–25, 2019, Mexico City, Mexico, pp. 26–35 (2019)
14. Lasserre, J.B.: Global optimization with polynomials and the problem of moments. SIAM J. Optim. **11**(3), 796–817 (2001)
15. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
16. Ma, X., Castanon, D.A.: Receding horizon planning for Dubins traveling salesman problems. In: Proceedings of the 45th IEEE Conference on Decision and Control, pp. 5453–5458 (2006)
17. McGee, T.G., Hedrick, J.K.: Path planning and control for multiple point surveillance by an unmanned aircraft in wind. In: 2006 American Control Conference, pp. 4261–4266 (2006)
18. Mitsch, S., Ghorbal, K., Vogelbacher, D., Platzer, A.: Formal verification of obstacle avoidance and navigation of ground robots. Int. J. Robot. Res. **36**(12), 1312–1340 (2017). https://doi.org/10.1177/0278364917733549
19. Parrilo, P.A.: Semidefinite programming relaxations for semialgebraic problems. Math. Program. **96**(2), 293–320 (2003)
20. Platzer, A.: Differential hybrid games. ACM Trans. Comput. Log. **18**(3), 19:1-19:44 (2017). https://doi.org/10.1145/3091123
21. Platzer, A., Clarke, E.M.: Formal verification of curved flight collision avoidance maneuvers: a case study. In: Cavalcanti, A., Dams, D. (eds.) FM, LNCS, vol. 5850, pp. 547–562. Springer (2009). https://doi.org/10.1007/978-3-642-05089-3_35
22. Prajna, S., Papachristodoulou, A., Parrilo, P.A.: Introducing SOSTOOLS: a general purpose sum of squares programming solver. In: IEEE Conference on Decision and Control, vol. 1, pp. 741–746 (2002)
23. Song, X., Hu, S.: 2d path planning with Dubins-path-based A⋆ algorithm for a fixed-wing UAV. In: 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE), Beijing, China, pp. 69–73 (2017)
24. The Coq proof assistant. https://coq.inria.fr (2020). Accessed 24 May 2020
25. Wu, A., How, J.: Guaranteed infinite horizon avoidance of unpredictable, dynamically constrained obstacles. Auton. Robots **32**(3), 227–242 (2012)
26. Zhao, Z., Yang, J., Niu, Y., Zhang, Y., Shen, L.: A hierarchical cooperative mission planning mechanism for multiple unmanned aerial vehicles. Electronics **8**, 443 (2019). https://doi.org/10.3390/electronics8040443