Journal of Computational Physics 545 (2026) 114451

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

On devising more competitive strategies for creating and avoiding collisions along differentiable trajectories

Trevor Maxfield [a,*], Ronald Fedkiw [b]

[a] Stanford University, 475 Via Ortega, Institute for Computational and Mathematical Engineering, Stanford, 94305, CA, United States
[b] Stanford University, 353 Jane Stanford Way, Gates Computer Science, Stanford, 94305, CA, United States

ARTICLE INFO

Keywords:
Collision avoidance
Trajectory optimization
Pursuit-Evasion games

ABSTRACT

Given the popularity of signed (and unsigned) distance functions, both for decades in computational physics and more recently in machine learning approaches to computer vision problems, we (re)consider their use in collisions in the context of distance-based pursuit-evasion scenarios of differential game theory. Although the positions and orientations of the various participating entities are governed by the laws of physics and thus require approximation via either numerical simulation or machine learning models, we make the reasonable ansatz that the resulting trajectories can be modeled to any desired accuracy with fully differentiable piecewise cubic polynomial interpolation; as such, the positions and orientations can be specified by smooth cubic polynomials determined by values at their endpoints, which are interpolated from a physical simulation or machine learning model that depends on various control parameters. Notably, this sort of abstraction also enables modeling of more diverse scenarios such as stock trading (and other economic models) as long as one can describe the underlying system by a set of time varying states suitable for piecewise polynomial interpolation. After developing a basic framework, we show that distance function based strategies are quickly derailed by more interesting objectives. The interesting features of the distance function are the critical points of the sixth order polynomial obtained by squaring the cubic spline trajectories, and controlling these critical points is equivalent to controlling the roots of the fifth order polynomial obtained by differentiating the sixth order polynomial; unfortunately, robustly differentiating the roots of a fifth order polynomial with respect to its coefficients remains beyond the scope of existing work. We then show that an alternative formulation, based on continuous collision detection (CCD), a popular alternative to distance functions for contact and collision problems, avoids the squaring required by the distance function formulation. This reduces the problem to the control of the roots of a third order polynomial; importantly, this problem can be addressed via prior work. Our new approach leads to a wider range of tractable objective functions and thus a greater number of strategies that agents may consider; obviously, this facilitates winning the so-called game.

1. Introduction

Real-world events are often viewed as being discontinuous, where an event either occurs or does not occur: a collision does or does not happen, a stock is or is not traded, etc. This has led to, for example, the theoretical ideas behind decision trees and probabilistic graphs, which are popular in classical artificial intelligence. Although one might aim to smooth decisions via mollified Heaviside

* Corresponding author.
E-mail address: maxfit@stanford.edu (T. Maxfield).

https://doi.org/10.1016/j.jcp.2025.114451
Received 30 April 2025; Received in revised form 13 September 2025; Accepted 10 October 2025
Available online 15 October 2025

functions, that leads to averaging outcomes across mutually exclusive branches, hindering the determination of control parameters via gradient-based optimization. The surging efficacy of machine learning techniques, particularly advances in numerical optimization for training neural networks, has promoted broad interest in the topic of differentiability, including that of discontinuous events.

Generally speaking, the machine learning and artificial intelligence communities do not have a long-and-storied history studying discontinuities and the mollification of discontinuities for the sake of differentiability; in fact, they have developed a number of naive approaches and software packages (e.g., PyTorch [1], TensorFlow [2], Torch [3], Caffe [4], Theano [5], Jax [6], etc.) for computing derivatives for the sake of training neural networks. A recent paper [7] showed that these software packages have a fundamentally flawed view of the chain rule, ignoring the importance of L'Hôpital's rule and related techniques. On the other hand, the computational physics community has been working with discontinuities since the beginning of the field (and the beginning of computers) due to the importance of shock waves, particularly in nuclear weapons research. This includes developments such as ENO [8], mollified Heavisides and sharp-interface methods (see e.g., [9]), level set methods [9], ghost fluid methods [10–12], immersed interface methods [13], XFEM [14], etc.

Although collision detection is an important component in a myriad of computational physics problems, its discontinuous nature has rarely been addressed (most research is focused on collision response). The crude way of detecting collisions consists of sampling geometry in both space and time to ascertain potential interpenetrations; of course, this approach is problematic when objects lack an interior, have thin features, are moving quickly, etc. The first-principles based approach to collision detection is typically referred to as continuous collision detection (CCD), see e.g., [15–17]. In CCD, time-continuous trajectories of geometry are used to formulate a root-finding problem, where the smallest non-negative real root (if it exists) represents the time of collision; moreover, the lack of a real root in a prescribed time interval indicates the absence of a collision. Noting that collisions or the lack of collisions may be important for obvious reasons motivates the desire to control the roots of CCD problems. As discussed in [18], this would require differentiating both real and imaginary roots of the CCD problem with respect to the parameters of the problem.

In order to bootstrap considerations of CCD differentiability in computational physics it makes sense to first consider an area where creating and avoiding collisions has received considerable attention in prior works, i.e., pursuit-evasion problems in differential game theory. In the context of the pursuit-evasion problem, a pursuer aims for real roots within a desirable time interval, while an evader aims to make those real roots either large or non-positive (assuming that zero corresponds to the current time). For more information on differential game theory and pursuit-evasion problems see the seminal work [19] as well as [20–24]. Numerical solutions of these games are considered in [25–28] and more recent works applying machine learning and artificial intelligence techniques (in particular, reinforcement learning) to these games include [29–35]. In addition, see e.g., [36–39] for prior work that references computational physics and game theory.

In addition to the competition between the pursuer and evader over the sign and value of the real roots, complex-valued roots should also be considered. Turning a pair of real-valued roots into complex conjugates, or vice versa, can directly benefit the evader or pursuer, respectively. More generally, consider a multiverse model where temporal trajectories have a real-valued root when an event occurs and a complex-valued root when it does not. Decisions do not necessarily lead to different outcomes, but rather work to influence outcomes by pushing real-valued roots to become complex-valued so that an event does not happen, or by pushing complex-valued roots to become real-valued so that an event does happen. Formulating viable strategies requires differentiating both real and complex-valued roots with respect to various potential controls, actions, or decisions.

In order to add mathematical rigor to the examples and discussions, the trajectories of the pursuer and evader will be modeled by piecewise cubic polynomial splines that enforce physically appropriate continuity of position and velocity at their endpoints. Notably, [18] proposed robust techniques for differentiating both real and complex-valued roots of cubic polynomials with respect to their parameters, facilitating the differentiability of events occurring along piecewise cubic spline trajectories. Given the marked differences between minimization and root-finding (not only mathematically, but strategically from the standpoint of game theory), both distance function and root-finding approaches to the pursuit-evasion problem will be considered. Distance function approaches are quite popular in prior work and seem quite natural at first glance; however, the pursuit of more realistic objective functions quickly becomes intractable, since it requires careful consideration and control of the roots of fifth order polynomials. Notably, our CCD-motivated approach replaces these fifth order polynomials with simpler and tractable (via [18]) third order polynomials. In contrast to prior work in both the computational physics and the computational geometry literature, replacing the distance function formulation with a CCD-based approach leads to a less complicated (as opposed to a more complicated) formulation.

Since we have chosen to illustrate our work via path planning and trajectory optimization examples, it is worth briefly considering prior related works. These include sampling-based methods such as [40–43], convex optimization methods such as [44–47], methods leveraging or providing numerically computed or estimated gradients such as [48–51], and collision avoidance methods utilizing Markov decision processes such as [52–55]. The most similar prior work is [56], which uses implicit differentiation and subgradients to plan trajectories that locally avoid obstacles. Their objective directly considers the time of a collision, and while they note the discontinuity between colliding and not colliding, there is no explicit formulation of this as a goal (as we do, with making roots complex). Other interesting and relevant works include [57–60].

2. Preliminaries

Assume that an agent has a set of controls $\vec{\lambda}$ that can be used to affect its state $\vec{X}(t;\vec{\lambda})$, e.g.,

$$\vec{X}(t;\vec{\lambda}) = \vec{X}(0) + \int_0^t \vec{V}(\xi;\vec{\lambda})d\xi \tag{1}$$

where $\vec{\lambda}$ represents all of the relevant controls, and $\vec{V}$ represents the temporal derivative of $\vec{X}$; in the context of a pursuit-evasion problem, $\vec{X}$ and $\vec{V}$ are position and velocity respectively. If the agents have no spatial extent, the displacement between them is

$$\vec{S}(t;\vec{\lambda}^P,\vec{\lambda}^E) = \vec{X}^P(t;\vec{\lambda}^P) - \vec{X}^E(t;\vec{\lambda}^E) \tag{2}$$

where P and E superscripts represent the pursuer and evader. If the agents have spatial extent, the displacement between any two particular points (one on the pursuer and one on the evader) is given by

$$\vec{S}_{\text{point}}(t;\vec{\lambda}^P,\vec{\lambda}^E) = \vec{X}^P(t;\vec{\lambda}^P) + \vec{r}^P - \left(\vec{X}^E(t;\vec{\lambda}^E) + \vec{r}^E\right) \tag{3}$$

where the offsets $\vec{r}^P/\vec{r}^E$ might also vary with t and $\vec{\lambda}$. Note that redefining $\vec{X}+\vec{r}$ as $\vec{X}$ makes Eq. (3) equivalent to Eq. (2). If interactions depend on proximity, a discretization of proximity boundaries would utilize Eq. (3) (equivalently Eq. (2)) for each relevant pair of discretized points (perhaps preferring the pair with minimum distance). Note that proximity boundaries could instead be interpreted as uncertainty (in position, proximity, etc.) boundaries. When the boundaries are represented by spheres, discretization can be replaced by the displacement

$$\begin{aligned}\vec{S}_{\text{sphere}}(t;\vec{\lambda}^P,\vec{\lambda}^E) &= \vec{X}^P(t;\vec{\lambda}^P) + \frac{\vec{X}^E(t;\vec{\lambda}^E) - \vec{X}^P(t;\vec{\lambda}^P)}{\left\|\vec{X}^E(t;\vec{\lambda}^E) - \vec{X}^P(t;\vec{\lambda}^P)\right\|_2} r^P \\ &\quad - \left(\vec{X}^E(t;\vec{\lambda}^E) + \frac{\vec{X}^P(t;\vec{\lambda}^P) - \vec{X}^E(t;\vec{\lambda}^E)}{\left\|\vec{X}^P(t;\vec{\lambda}^P) - \vec{X}^E(t;\vec{\lambda}^E)\right\|_2} r^E\right) \qquad (4a)\\ &= \left(\vec{X}^P(t;\vec{\lambda}^P) - \vec{X}^E(t;\vec{\lambda}^E)\right)\left(1 - \frac{\eta}{\left\|\vec{X}^P(t;\vec{\lambda}^P) - \vec{X}^E(t;\vec{\lambda}^E)\right\|_2}\right) \qquad (4b)\end{aligned}$$

where $\eta = r^P + r^E$ is the sum of the radii of the spheres. Note that $\vec{S}_{\text{sphere}}$ flips direction when the spheres overlap which isn't necessarily sensible (unless the goal is to obtain an exact separation of η).

Typically, an agent is not in complete control of its state (position, velocity, acceleration, etc.), but rather aims to control its state by accounting for or reacting to external forces outside of its control (these could be non-physical as well, e.g. economic models). For example, when these external forces cause some acceleration $\vec{A}$ on an agent, a continuous representation of the physics can be written as

$$\vec{X}(t_{n+1}) - \vec{X}(t_n) - (t_{n+1}-t_n)\vec{V}(t_n) = \int_{t_n}^{t_{n+1}}\int_{t_n}^{\xi_1} \vec{A}(\xi_2)\,d\xi_2\,d\xi_1 \tag{5}$$

where $d\vec{X} = \vec{V}\,dt$ and $d\vec{V} = \vec{A}\,dt$. Since only the most trivial physics models have analytic solutions, Eq. (5) needs to be replaced with a discretization in order to facilitate a numerical approach. Recently, it has become popular to replace classical discretizations with surrogate models based on machine learning or other reduced order techniques, e.g., [61–63]. While these surrogates decrease the accuracy of the physics, they often increase the efficiency of numerical approaches, thereby improving the feasibility of optimization methods. Moreover, these surrogates can provide the required differentiability with respect to the controls that may not exist in more complex models and solvers.

In spite of discretizing for the sake of solvability, it is useful to have a continuous representation for the sake of differentiability and control. This can be achieved via piecewise polynomial interpolation of the discrete values output by the numerical method. Noting that it is well known that higher order polynomials tend to oscillate and overshoot, we utilize piecewise cubic interpolation emphasizing that it is the lowest order piecewise polynomial interpolation that can be both continuous and differentiable.

Let $\vec{x}(t)$ be the piecewise cubic polynomial trajectory of a point under consideration. Given a time interval $[t_{\text{left}}, t_{\text{right}}]$, the cubic polynomial spline

$$\vec{x}(t) = \vec{q}t^3 + \vec{a}t^2 + \vec{b}t + \vec{c} \tag{6}$$

has parameters described by the following conditions

$$\begin{bmatrix} t_{\text{left}}^3 & t_{\text{left}}^2 & t_{\text{left}} & 1 \\ 3t_{\text{left}}^2 & 2t_{\text{left}} & 1 & 0 \\ t_{\text{right}}^3 & t_{\text{right}}^2 & t_{\text{right}} & 1 \\ 3t_{\text{right}}^2 & 2t_{\text{right}} & 1 & 0 \end{bmatrix}\begin{bmatrix}\vec{q}\\ \vec{a}\\ \vec{b}\\ \vec{c}\end{bmatrix} = \begin{bmatrix}\vec{x}_{\text{left}}\\ \vec{v}_{\text{left}}\\ \vec{x}_{\text{right}}\\ \vec{v}_{\text{right}}\end{bmatrix} \tag{7}$$

meaning that

$$\begin{bmatrix}\vec{q}\\ \vec{a}\\ \vec{b}\\ \vec{c}\end{bmatrix} = \begin{bmatrix} \frac{2}{(t_{\text{right}}-t_{\text{left}})^3} & \frac{1}{(t_{\text{right}}-t_{\text{left}})^2} & \frac{-2}{(t_{\text{right}}-t_{\text{left}})^3} & \frac{1}{(t_{\text{right}}-t_{\text{left}})^2} \\ \frac{-3t_{\text{right}}-3t_{\text{left}}}{(t_{\text{right}}-t_{\text{left}})^3} & \frac{-2t_{\text{right}}-t_{\text{left}}}{(t_{\text{right}}-t_{\text{left}})^2} & \frac{3t_{\text{right}}+3t_{\text{left}}}{(t_{\text{right}}-t_{\text{left}})^3} & \frac{-t_{\text{right}}-2t_{\text{left}}}{(t_{\text{right}}-t_{\text{left}})^2} \\ \frac{6t_{\text{right}}t_{\text{left}}}{(t_{\text{right}}-t_{\text{left}})^3} & \frac{t_{\text{right}}^2+2t_{\text{right}}t_{\text{left}}}{(t_{\text{right}}-t_{\text{left}})^2} & \frac{-6t_{\text{left}}t_{\text{right}}}{(t_{\text{right}}-t_{\text{left}})^3} & \frac{2t_{\text{right}}t_{\text{left}}+t_{\text{left}}^2}{(t_{\text{right}}-t_{\text{left}})^2} \\ \frac{t_{\text{right}}^3-3t_{\text{right}}^2t_{\text{left}}}{(t_{\text{right}}-t_{\text{left}})^3} & \frac{-t_{\text{right}}^2t_{\text{left}}}{(t_{\text{right}}-t_{\text{left}})^2} & \frac{3t_{\text{right}}t_{\text{left}}^2-t_{\text{left}}^3}{(t_{\text{right}}-t_{\text{left}})^3} & \frac{-t_{\text{right}}t_{\text{left}}^2}{(t_{\text{right}}-t_{\text{left}})^2} \end{bmatrix}\begin{bmatrix}\vec{x}_{\text{left}}\\ \vec{v}_{\text{left}}\\ \vec{x}_{\text{right}}\\ \vec{v}_{\text{right}}\end{bmatrix} = T\begin{bmatrix}\vec{x}_{\text{left}}\\ \vec{v}_{\text{left}}\\ \vec{x}_{\text{right}}\\ \vec{v}_{\text{right}}\end{bmatrix} \tag{8}$$

with $t_{\text{left}} \neq t_{\text{right}}$. Note that the time intervals used to resolve trajectories for the pursuer and evader will necessarily differ from the time steps used to obtain a stable and accurate numerical discretization of the underlying physics. This means that $\vec{x}_{\text{left}}$, $\vec{v}_{\text{left}}$, $\vec{x}_{\text{right}}$, and $\vec{v}_{\text{right}}$ will be interpolated from $\vec{X}$ and $\vec{V}$; in addition, $\vec{x}_{\text{left}}$, $\vec{v}_{\text{left}}$, $\vec{x}_{\text{right}}$, and $\vec{v}_{\text{right}}$ are functions of the controls $\vec{\lambda}$, since $\vec{X}$ and $\vec{V}$ are functions of $\vec{\lambda}$. When a discussion is facilitated by nondimensionalizing time, T can be factored in order to define $\hat{t} = \frac{t - t_{\text{left}}}{t_{\text{right}} - t_{\text{left}}} \in [0,1]$ via

$$\begin{bmatrix} \hat{t}^3 & \hat{t}^2 & \hat{t} & 1 \end{bmatrix} = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{(t_{\text{right}} - t_{\text{left}})^3} & 0 & 0 & 0 \\ \frac{-3t_{\text{left}}}{(t_{\text{right}} - t_{\text{left}})^3} & \frac{1}{(t_{\text{right}} - t_{\text{left}})^2} & 0 & 0 \\ \frac{3t_{\text{left}}^2}{(t_{\text{right}} - t_{\text{left}})^3} & \frac{-2t_{\text{left}}}{(t_{\text{right}} - t_{\text{left}})^2} & \frac{1}{t_{\text{right}} - t_{\text{left}}} & 0 \\ \frac{-t_{\text{left}}^3}{(t_{\text{right}} - t_{\text{left}})^3} & \frac{t_{\text{left}}^2}{(t_{\text{right}} - t_{\text{left}})^2} & \frac{-t_{\text{left}}}{t_{\text{right}} - t_{\text{left}}} & 1 \end{bmatrix} \tag{9}$$

with

$$\hat{T} = \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & t_{\text{right}} - t_{\text{left}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & t_{\text{right}} - t_{\text{left}} \end{bmatrix} \tag{10}$$

remaining; then, $\begin{bmatrix} \hat{t}^3 & \hat{t}^2 & \hat{t} & 1 \end{bmatrix} \hat{T}$ replaces $\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} T$.

In each $[t_{\text{left}}, t_{\text{right}}]$ interval, each dimension has a separate and independent cubic polynomial

$$x_i(t; \vec{p}_i(\vec{\lambda})) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \vec{p}_i(\vec{\lambda}) \tag{11}$$

where

$$\vec{p}_i(\vec{\lambda}) = \begin{bmatrix} q_i(\vec{\lambda}) \\ a_i(\vec{\lambda}) \\ b_i(\vec{\lambda}) \\ c_i(\vec{\lambda}) \end{bmatrix} = T \begin{bmatrix} x_{i,\text{left}}(\vec{\lambda}) \\ v_{i,\text{left}}(\vec{\lambda}) \\ x_{i,\text{right}}(\vec{\lambda}) \\ v_{i,\text{right}}(\vec{\lambda}) \end{bmatrix} \tag{12}$$

is a vector of parameters. Assuming that the pursuit-evasion problem is in $\mathbb{R}^3$, let $\vec{\mathbf{p}}(\vec{\lambda})$ be a 12×1 vertical concatenation of the three $\vec{p}_i(\vec{\lambda})$. This allows one to write $\vec{x}(t; \vec{\mathbf{p}}(\vec{\lambda}))$ in order to reflect the dependencies; then, $\frac{\partial \vec{x}}{\partial \vec{\mathbf{p}}}$ is a 3×12 block diagonal matrix with a $\frac{\partial x_i}{\partial \vec{p}_i}$ in each block. Let $\vec{s}(t)$ be the piecewise cubic polynomial displacement between the piecewise cubic trajectories of the pursuer and evader points under consideration. In each interval, Eq. (2) (and equivalently Eq. (3)) becomes

$$\vec{s}(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) = \vec{x}^P(t; \vec{\mathbf{p}}^P(\vec{\lambda}^P)) - \vec{x}^E(t; \vec{\mathbf{p}}^E(\vec{\lambda}^E)) \tag{13}$$

where in each dimension

$$s_i(t; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) = x_i^P(t; \vec{p}_i^P(\vec{\lambda}^P)) - x_i^E(t; \vec{p}_i^E(\vec{\lambda}^E)) \tag{14a}$$

$$= \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \left(\vec{p}_i^P(\vec{\lambda}^P) - \vec{p}_i^E(\vec{\lambda}^E) \right) \tag{14b}$$

$$= \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) \tag{14c}$$

noting that Eqs. (13) and (14) (especially Eqs. (14b) and (14c)) depend on the ability to subdivide the pursuer's and evader's splines so they are defined on coincident time intervals. Note that

$$\vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) = T \begin{bmatrix} x_{i,\text{left}}^P(\vec{\lambda}^P) - x_{i,\text{left}}^E(\vec{\lambda}^E) \\ v_{i,\text{left}}^P(\vec{\lambda}^P) - v_{i,\text{left}}^E(\vec{\lambda}^E) \\ x_{i,\text{right}}^P(\vec{\lambda}^P) - x_{i,\text{right}}^E(\vec{\lambda}^E) \\ v_{i,\text{right}}^P(\vec{\lambda}^P) - v_{i,\text{right}}^E(\vec{\lambda}^E) \end{bmatrix} \tag{15}$$

and that $\vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E) = \vec{\mathbf{p}}^P(\vec{\lambda}^P) - \vec{\mathbf{p}}^E(\vec{\lambda}^E)$ is 12×1. In addition, $\frac{\partial \vec{s}}{\partial \vec{\mathbf{p}}^s}$ is a 3×12 block diagonal matrix with a $\frac{\partial s_i}{\partial \vec{p}_i^s}$ in each block. Similarly, Eq. (4) becomes

$$\vec{s}_{\text{sphere}}(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) = \vec{s}(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) \left(1 - \frac{\eta}{\left\| \vec{s}(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) \right\|_2} \right) \tag{16}$$

where $\vec{s}$ is defined via Eq. (13) using $\vec{x}^P$ and $\vec{x}^E$ as the centers of the two spheres.

In order to utilize numerical optimization on the various and competing objectives in pursuit-evasion games, differentiability of the objectives with respect to the controls ($\vec{\lambda}^P$ and $\vec{\lambda}^E$) is required. Since our proposed objectives are primarily concerned with controlling, creating, and/or eliminating roots (i.e., collisions), complex values of t need to be considered. In order to distinguish the obvious restriction that $t \in \mathbb{R}$ for trajectories from the complex values of t that need to be considered for the sake of roots, the

notation t_{root} with $t_{\text{root}} \in \mathbb{C}$ is utilized. Throughout this work $\mathbb{C}$ will be represented via $\mathbb{R}^2$ for ease and clarity in implementation; thus, complex numbers will generally be written as 2×1 column vectors of the real and complex parts. Occasionally, multiplying complex numbers will require writing them as 2×2 matrices.

In order to succinctly cover both real and complex cases, Eq. (14c) can be rewritten as

$$s_i(t; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) = \begin{bmatrix} (t^R)^3 - 3t^R(t^I)^2 & (t^R)^2 - (t^I)^2 & t^R & 1 \\ 3(t^R)^2 t^I - (t^I)^3 & 2t^R t^I & t^I & 0 \end{bmatrix} \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) \tag{17}$$

where $t = t^R + it^I$, stressing that $t^I = 0$ identically along any trajectory. This leads to

$$ds_i(t; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) = \frac{\partial s_i(t; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t} dt + \frac{\partial s_i(t; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} d\vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) \tag{18}$$

for the differentials, where

$$\frac{\partial s_i(t; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t} = \begin{bmatrix} \begin{pmatrix} 3(t^R)^2 - 3(t^I)^2 & 2t^R & 1 & 0 \end{pmatrix} \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) & \begin{pmatrix} -6t^R t^I & -2t^I & 0 & 0 \end{pmatrix} \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) \\ \begin{pmatrix} 6t^R t^I & 2t^I & 0 & 0 \end{pmatrix} \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) & \begin{pmatrix} 3(t^R)^2 - 3(t^I)^2 & 2t^R & 1 & 0 \end{pmatrix} \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) \end{bmatrix} \tag{19a}$$

$$\frac{\partial s_i(t; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} = \begin{bmatrix} (t^R)^3 - 3t^R(t^I)^2 & (t^R)^2 - (t^I)^2 & t^R & 1 \\ 3(t^R)^2 t^I - (t^I)^3 & 2t^R t^I & t^I & 0 \end{bmatrix} \tag{19b}$$

define the partial derivatives. Combining Eqs. (12) and (14) leads to

$$\frac{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^P} = T \begin{bmatrix} \frac{\partial x^P_{i,\text{left}}(\vec{\lambda}^P)}{\partial \vec{\lambda}^P} \\ \frac{\partial v^P_{i,\text{left}}(\vec{\lambda}^P)}{\partial \vec{\lambda}^P} \\ \frac{\partial x^P_{i,\text{right}}(\vec{\lambda}^P)}{\partial \vec{\lambda}^P} \\ \frac{\partial v^P_{i,\text{right}}(\vec{\lambda}^P)}{\partial \vec{\lambda}^P} \end{bmatrix} \quad \text{and} \quad \frac{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^E} = -T \begin{bmatrix} \frac{\partial x^E_{i,\text{left}}(\vec{\lambda}^E)}{\partial \vec{\lambda}^E} \\ \frac{\partial v^E_{i,\text{left}}(\vec{\lambda}^E)}{\partial \vec{\lambda}^E} \\ \frac{\partial x^E_{i,\text{right}}(\vec{\lambda}^E)}{\partial \vec{\lambda}^E} \\ \frac{\partial v^E_{i,\text{right}}(\vec{\lambda}^E)}{\partial \vec{\lambda}^E} \end{bmatrix} \tag{20}$$

which, with Eq. (18), leads to

$$\frac{\partial s_i(t; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{\lambda}^{P/E}} = \pm \begin{bmatrix} (t^R)^3 - 3t^R(t^I)^2 & (t^R)^2 - (t^I)^2 & t^R & 1 \\ 3(t^R)^2 t^I - (t^I)^3 & 2t^R t^I & t^I & 0 \end{bmatrix} T \begin{bmatrix} \frac{\partial x^P_{i,\text{left}}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}} \\ \frac{\partial v^P_{i,\text{left}}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}} \\ \frac{\partial x^P_{i,\text{right}}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}} \\ \frac{\partial v^P_{i,\text{right}}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}} \end{bmatrix} \tag{21}$$

making the dependence on the controls more explicit.

2.1. Method and strategies

Our proposed method is as follows: An agent has some model of how the displacement $\vec{S}$ between itself and its opponent evolves over time with respect to its own, and potentially its opponent's, controls. This displacement can be interpolated by piecewise smooth cubic splines, which leads to a consistent framework for considering various strategies, i.e., objective functions, that encourage or avoid collisions, typically described in the language of pursuit-evasion problems. These strategies are applied via gradients of the corresponding objective functions with respect to the interpolation points of the cubic splines and ultimately with respect to the controls via the chain rule (with derivatives, or estimates of the derivatives, of the model of the displacement). Our work is primarily concerned with the different strategies that consider collisions. When an opponent's trajectory cannot be directly modeled or is uncertain, we consider methods for bounding their trajectory so that similar strategies for creating or avoiding collisions can be applied.

There are two quite disparate strategies for addressing the pursuit-evasion problem. One could focus on distance and aim to minimize or maximize a norm of $\vec{s}$, or one could focus on collisions by considering the roots of the three equations obtained by setting each s_i equal to zero. Different strategies arise depending on which approach is considered. In particular, there are many strategies that avoid collisions while the distance decreases. When entities are relatively far apart, distance is probably more appropriate given the uncertainties in trajectory approximation; however, when entities are closer together, a root-finding approach introduces significantly more strategies. Generally speaking, an agent is at a disadvantage when it ignores or is otherwise unaware of potential strategies of an opponent; thus, we flush out some of the details associated with using our formulation in a distance-based approach (in Section 3) before moving on to the consideration of collisions (in Section 4). Both approaches are then considered in a simple example (in Sections 5 and 6).

For an agent to formulate the displacement between itself and another agent requires some estimation of its opponent's trajectory. This is largely dependent on what an agent knows and believes about its opponent, so the specifics of the estimation ought to be tuned to the problem at hand. One option is for an agent to treat its opponent's strategy as entirely unknown or uncertain, and in

such cases the agent can place a bound on its opponent's possible positions. An agent seeking to collide with its opponent can aim for the center of a bounding volume or a discretization of its surface. Conversely, an agent seeking to avoid collisions with its opponent must avoid the bounding region entirely. This is accomplished by the application of axis-aligned bounding boxes (in Section 9), where avoiding collisions with the box reduces to a root-finding problem involving derivatives and optimization of the roots in at most two dimensions, regardless of the overall dimensionality of the game. Physical plausibility of the interpolating splines is also considered (in Sections 6 and 7) and all of these components are later combined in a simple pursuit-evasion game (in Section 10).

3. Distance formulation

A distance function can be defined from the displacement,

$$\phi(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E)) = \left\|\vec{s}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))\right\|_2 \tag{22}$$

for Eq. (13) and

$$\phi_{\text{sphere}}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E)) = \left|\left\|\vec{s}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))\right\|_2 - \eta\right| \tag{23a}$$

$$\tilde{\phi}_{\text{sphere}}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E)) = \left\|\vec{s}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))\right\|_2 - \eta \tag{23b}$$

for Eq. (16), where Eq. (23b) calculates signed distance, facilitating discernment between being too close with $\tilde{\phi}_{\text{sphere}} < 0$ and being too far with $\tilde{\phi}_{\text{sphere}} > 0$. Eq. (23a) is useful when η is the preferred separation distance, whereas Eq. (23b) is useful when all points inside the sphere are treated equivalently.

The time when the distance is minimized

$$t_{\min} = \arg\min_{t\geq 0} \phi(t) \tag{24}$$

represents a critical point of the problem. To motivate strategies utilizing distance functions, consider a potential strategy for the pursuer

$$\min_{\vec{\lambda}^P} \; H(\phi(t_{\min}))\,\phi(t_{\min}) + H(-\phi(t_{\min}))\,t_{\min} \tag{25}$$

where H is the Heaviside function. The first term aims to minimize the distance when it is positive, while the second aims to minimize $t_{\min}$ when $\phi \leq 0$. Note that Eq. (25) is applicable for both ϕ_{sphere} and $\tilde{\phi}_{\text{sphere}}$ as well. The evader would work against this via

$$\max_{\vec{\lambda}^E} \; H(\phi(t_{\min}))\,\phi(t_{\min}) + H(-\phi(t_{\min}))\,t_{\min} \tag{26}$$

where the first term maximizes the minimal distance when it is positive, while the second term aims to maximize $t_{\min}$ when $\phi \leq 0$. Typically, Eqs. (25) and (26) are combined and modified to

$$\min_{\vec{\lambda}^P}\max_{\vec{\lambda}^E} \; H(\phi(t_{\min}))\,\phi(t_{\min}) + H(-\phi(t_{\min}))\,t_{\min} \tag{27a}$$

$$\max_{\vec{\lambda}^E}\min_{\vec{\lambda}^P} \; H(\phi(t_{\min}))\,\phi(t_{\min}) + H(-\phi(t_{\min}))\,t_{\min} \tag{27b}$$

respectively, in order to account for the pursuer and evader being aware of each other's strategies.

As is typical, the global minimum defined in Eq. (24) is often replaced by the consideration of local minima. Since differentiating ϕ requires division by $\|\vec{s}\|_2$ with $\|\vec{s}\|_2 \to 0$ as $\phi \to 0$, it is better to differentiate $\frac{1}{2}\phi^2$; moreover, in the $\phi \geq 0$ case, ϕ and $\frac{1}{2}\phi^2$ have the same minima. Thus,

$$\psi(t_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E)) = \frac{\partial}{\partial t}\frac{\phi(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))^2}{2}\bigg|_{t=t_{\min}} = \vec{s}(t_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))^T\frac{\partial\vec{s}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t}\bigg|_{t=t_{\min}} = 0 \tag{28}$$

implicitly defines the $t_{\min}$ as critical points. Note that $t \in \mathbb{R}$, $t_{\min} \in \mathbb{R}$, $\phi \in \mathbb{R}$, and $\vec{s} \in \mathbb{R}^3$ means that each s_i is defined by only the top row of Eq. (17) with $t^I = 0$; similarly, the scalar $\frac{\partial s_i}{\partial t}$ is defined by the upper left entry of the 2×2 matrix in Eq. (19a) with $t^I = 0$. The inner product between $\vec{s}$ and $\frac{\partial\vec{s}}{\partial t}$ is the sum of fifth order polynomials, each of which is formed by multiplying a cubic by a quadratic. Differentiating $\frac{1}{2}\phi^2$ can be used to identify potential $t_{\min}$ for ϕ_{sphere} and $\tilde{\phi}_{\text{sphere}}$ as well, noting that when $\tilde{\phi}_{\text{sphere}} < 0$ local minima become local maxima (and vice versa) and that new local minima are created (if they don't already exist) wherever $\tilde{\phi}_{\text{sphere}} = 0$. For both Eqs. (23a) and (23b),

$$\psi_{\text{sphere}}(t_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E)) = \left(1 - \frac{\eta}{\left\|\vec{s}(t_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))\right\|_2}\right)\psi(t_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E)) = 0 \tag{29}$$

implicitly defines the $t_{\min}$ as critical points. Note that division by a problematic $\|\vec{s}\|_2 = 0$ would only occur away from the spherical interaction boundary at the center of the sphere. The solutions of Eq. (29) include the solutions of Eq. (28) in addition to points on the sphere where $\|\vec{s}\|_2 = \eta$. Note that $\|\vec{s}\|_2^2 = \eta^2$ is a sixth order polynomial.

Remark 3.1. *Solving Eq. (28) for t_{min} requires finding the roots of a fifth order polynomial. Thus, controlling t_{min}, as is suggested by Eq. (27), requires differentiating a real root of a fifth order polynomial with respect to the coefficients of the polynomial. Solving for this derivative requires inverting $\frac{\partial \psi}{\partial t}$ at t_{min}, which is problematic when t_{min} is nearly a repeated root and $\frac{\partial \psi}{\partial t} \to 0$. Analysis of the different ways (in the paths of the polynomial coefficients) that a root can become repeated can lead to suitable directions for the gradient, as in the cubic polynomial case addressed in [18]. However, a robust numerical method for fifth order polynomials does not currently exist. This inability to properly treat t_{min} is a roadblock to distance function formulations.*

Converting a real-valued root $t_{\min}$ to be complex-valued or vice versa removes or adds, respectively, local extrema. In order to facilitate this strategy, Eq. (28) needs to be converted into a two equation system with both the real and imaginary parts set equal to zero; thus, we rewrite Eq. (28) as

$$\psi(t_{\min}; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) = \sum_i \left. \frac{\partial s_i(t; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t} \right|_{t=t_{\min}} s_i(t_{\min}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) = 0 \tag{30}$$

where $t \in \mathbb{C}$, $t_{\min} \in \mathbb{C}$, $\psi \in \mathbb{C}$, and each $s_i \in \mathbb{C}$ along the lines of Eqs. (17) and (19a). For the sake of complexifying $\|\vec{s}\|_2 = \eta$,

$$\|\vec{s}\|_2^2 = \sum_i s_i^* s_i = \sum_i (s_i^R)^2 + (s_i^I)^2 \tag{31}$$

as is standard. Note that consideration of $t_{\min} \in \mathbb{C}$ can be made consistent with $\phi \in \mathbb{R}$ by utilizing $\phi(t_{\min}^R)$ instead of $\phi(t_{\min})$ when $t_{\min} \in \mathbb{C}$.

Remark 3.2. *Technically, $t \in \mathbb{R}$ in Eq. (28) dictates that the derivative in Eq. (30) is with respect to $t \in \mathbb{R}$ not $t \in \mathbb{C}$. After differentiating with respect to $t \in \mathbb{R}$, $t_{min} \in \mathbb{C}$ is substituted into the result before $\frac{\partial s_i}{\partial t} \in \mathbb{C}$ and $s_i \in \mathbb{C}$ are multiplied via*

$$\frac{\partial s_i}{\partial t} s_i = \begin{bmatrix} Re\, \frac{\partial s_i}{\partial t} & -Im\, \frac{\partial s_i}{\partial t} \\ Im\, \frac{\partial s_i}{\partial t} & Re\, \frac{\partial s_i}{\partial t} \end{bmatrix} \begin{bmatrix} Re\, s_i \\ Im\, s_i \end{bmatrix} \tag{32}$$

as is standard. To see that the 2×2 matrix in Eq. (32) is identical to the 2×2 matrix in Eq. (30) (and Eq. (19a)), consider differentiating

$$s_i(t^R + it^I) = s_i^R(t^R, t^I) + i s_i^I(t^R, t^I) \tag{33}$$

with respect to t^R and t^I to obtain

$$s_i'(t^R + it^I) \begin{bmatrix} 1 & i \end{bmatrix} = \begin{bmatrix} \frac{\partial s_i^R(t^R,t^I)}{\partial t^R} + i \frac{\partial s_i^I(t^R,t^I)}{\partial t^R} & \frac{\partial s_i^R(t^R,t^I)}{\partial t^I} + i \frac{\partial s_i^I(t^R,t^I)}{\partial t^I} \end{bmatrix} \tag{34a}$$

$$\begin{bmatrix} Re\, s_i'(t^R + it^I) & -Im\, s_i'(t^R + it^I) \\ Im\, s_i'(t^R + it^I) & Re\, s_i'(t^R + it^I) \end{bmatrix} = \begin{bmatrix} \frac{\partial s_i^R(t^R,t^I)}{\partial t^R} & \frac{\partial s_i^R(t^R,t^I)}{\partial t^I} \\ \frac{\partial s_i^I(t^R,t^I)}{\partial t^R} & \frac{\partial s_i^I(t^R,t^I)}{\partial t^I} \end{bmatrix} \tag{34b}$$

where s_i', computed for the sake of the chain rule, is identical to $\frac{\partial s_i}{\partial t}$ with $t \in \mathbb{R}$. Note that this result is expected, since polynomials are analytic.

3.1. Differentials

Eqs. (22) and (23b) have differential

$$d\phi(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) = \frac{\vec{s}(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))^T}{\left\|\vec{s}(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))\right\|_2} d\vec{s}(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) \tag{35}$$

as does Eq. (23a) when $||\vec{s}||_2 \geq \eta$, but multiplication by -1 is required when $||\vec{s}||_2 < \eta$. Straightforward substitution of $d\vec{s}$ from Eq. (18) into Eq. (35) leads to the partial derivatives of ϕ, ϕ_{sphere}, or $\tilde{\phi}_{\text{sphere}}$ with respect to t and $\vec{\lambda}^{P/E}$. From Eq. (30),

$$d\psi(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) = \frac{\partial \psi(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t} dt + \sum_i \frac{\partial \psi(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} d\vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) \tag{36}$$

where the partial derivatives can be found in Eq. (A.1). For $t_{\min}$ a root of ψ, $\psi(t_{\min}) = 0$ implies $d\psi(t_{\min}; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) = 0$; thus, substituting $t_{\min}$ into Eq. (36) leads to

$$\frac{\partial \psi(t_{\min}; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\min}} dt_{\min} + \sum_i \frac{\partial \psi(t_{\min}; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} d\vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) = 0 \tag{37a}$$

$$\frac{\partial \psi(t_{\min}; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\min}} \frac{\partial t_{\min}}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} = -\frac{\partial \psi(t_{\min}; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} \tag{37b}$$

as an implicit equation for each $\frac{\partial t_{\min}}{\partial \vec{p}_i^s}$.

Eq. (37b) is an implicit equation for the derivatives of the roots of a fifth-order polynomial with respect to its coefficients. As mentioned in Remark 3.1, [18] only addresses third-order polynomials and given the intricacies discussed therein, robustly solving

Eq. (37b) seems rather difficult. In particular, the 2×2 coefficient matrix $\frac{\partial\psi}{\partial t_{\min}}$ in Eq. (37b) has the same form as the 2×2 coefficient matrix on the righthand side of Eq. (34b) after replacing s_i with ψ. This matrix is difficult to invert when the determinant approaches zero. The lefthand side of Eq. (34b) illustrates that the determinant is simply the magnitude squared of the complex number $\frac{\partial\psi}{\partial t_{\min}}$, computed in a straightforward way by assuming $t_{\min}\in\mathbb{C}$ and $\psi\in\mathbb{C}$.

A collision, with $\phi=0$, is a local and global minimum of $\frac{1}{2}\phi^2$ with $\frac{\partial\phi}{\partial t}=0$ as well. The corresponding $\frac{\partial\psi}{\partial t}=\frac{\partial\phi}{\partial t}\frac{\partial\phi}{\partial t}+\phi\frac{\partial^2\phi}{\partial t^2}=0$ is problematic for Eq. (37b). When a collision is not forecast, setting $\psi=\phi\frac{\partial\phi}{\partial t}=0$ with $\phi\neq 0$ means that a time at which $\frac{\partial\phi}{\partial t}=0$ is considered. Note that $\frac{\partial\phi}{\partial t}=0$ would be considered even if ϕ were used in the objective rather than $\frac{1}{2}\phi^2$. When there is no relevant time with $\frac{\partial\phi}{\partial t}=0$, the pursuer could either aim to move a real root into the domain of interest or aim to merge two complex roots in order to create a local minimum and maximum. Merging roots causes a repeated root with both $\frac{\partial\phi}{\partial t}=0$ and $\frac{\partial^2\phi}{\partial t^2}=0$ resulting in a problematic $\frac{\partial\psi}{\partial t}=0$. Given a time at which $\frac{\partial\phi}{\partial t}=0$, the pursuer aims to merge complex conjugate roots of ϕ onto this root of $\frac{\partial\phi}{\partial t}$ in order to create a collision, which has a problematic $\frac{\partial\psi}{\partial t}=0$. The evader resists these aforementioned strategies by aiming to keep roots of $\frac{\partial\phi}{\partial t}$ irrelevant or complex and $\phi^2>0$. Unfortunately, the game is played at or near $\frac{\partial\psi}{\partial t}=0$ where it is difficult to robustly solve Eq. (37b) numerically.

Remark 3.3. *Whenever the pursuer is successful at getting closer to the evader, the forecast distance decreases to a relative minima with $\frac{\partial\phi}{\partial t}=0$ before increasing again. If the evader successfully disallows this pursuer progress, then any relevant roots of $\frac{\partial\phi}{\partial t}$ would be complex. Thus, the key point of contention in distance function formulations is when $\frac{\partial\phi}{\partial t}=0$ has a repeated root (making $\frac{\partial\psi}{\partial t}=0$), highlighting the fundamental difficulty in distance function formulations.*

When using spherical spatial extent, the derivation of Eq. (37b) from Eq. (36) still holds with ψ replaced by ψ_{sphere}. However, the partial derivatives in Eq. (A.1) are replaced by

$$\frac{\partial\psi_{\text{sphere}}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t}=\left(1-\frac{\eta}{\left\|\vec{s}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))\right\|_2}\right)\frac{\partial\psi(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t}+\frac{\eta\,\psi(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))^2}{\left\|\vec{s}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))\right\|_2^3} \tag{38a}$$

$$\begin{aligned}\frac{\partial\psi_{\text{sphere}}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)}&=\left(1-\frac{\eta}{\left\|\vec{s}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))\right\|_2}\right)\frac{\partial\psi(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)}\\&\quad+\frac{\eta\,\psi(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\left\|\vec{s}(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))\right\|_2^3}s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))\frac{\partial s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)}\end{aligned} \tag{38b}$$

leveraging Eq. (28) for the definition of ψ in order to simplify the second term in Eq. (38a). Note that Eq. (37b) with these partial derivatives of ψ_{sphere} holds for both ϕ_{sphere} and $\tilde{\phi}_{\text{sphere}}$, since ψ_{sphere} in Eq. (29) is the same for both. Similar to the issues discussed in Remarks 3.1 and 3.3, the various difficulties associated with $\frac{\partial\psi_{\text{sphere}}}{\partial t_{\min}}$ would need to be addressed in order to solve Eq. (37b) for $\frac{\partial t_{\min}}{\partial\vec{p}_i^s}$.

Remark 3.4. *Note that ψ^2 in Eq. (38a) needs to be converted into a 2×2 matrix in order to be consistent with the notation used throughout this paper. This can be accomplished along the lines of Eq. (32), except by converting both complex numbers into 2×2 matrices. The justification would proceed similarly to Remark 3.2 and conclude with the fact that polynomials are analytic. In fact, if we had abstained from changing the order of s_i and $\frac{\partial s_i}{\partial t}$ in Eqs. (28) and (30), then we would have been forced to turn a complex s_i into a 2×2 matrix making ψ a 2×2 matrix in Eq. (28). Similarly, both ψ and s_i need to be converted into 2×2 matrices in Eq. (38b).*

Substituting Eq. (20) into Eq. (37a) leads to

$$\frac{\partial\psi(t_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t_{\min}}\frac{\partial t_{\min}}{\partial\vec{\lambda}^{P/E}}=-\sum_i\frac{\partial\psi(t_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)}\frac{\partial\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)}{\partial\vec{\lambda}^{P/E}} \tag{39}$$

and substituting Eq. (37b) into Eq. (39) leads to

$$\frac{\partial\psi(t_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t_{\min}}\frac{\partial t_{\min}}{\partial\vec{\lambda}^{P/E}}=\frac{\partial\psi(t_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t_{\min}}\sum_i\frac{\partial t_{\min}}{\partial\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)}\frac{\partial\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)}{\partial\vec{\lambda}^{P/E}} \tag{40}$$

which gives

$$\frac{\partial t_{\min}}{\partial\vec{\lambda}^{P/E}}=\sum_i\frac{\partial t_{\min}}{\partial\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)}\frac{\partial\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)}{\partial\vec{\lambda}^{P/E}} \tag{41}$$

when one can reliably invert $\frac{\partial\psi}{\partial t_{\min}}$. Since Eq. (37) also holds in the spherical spatial extent case, Eqs. (39)–(41) can be derived in the spherical spatial extent case as well. Eq. (41) dictates solving Eq. (37b) for each of the $\frac{\partial t_{\min}}{\partial\vec{p}_i^s}$ and subsequently multiplying each result by the partial derivatives in Eq. (20) before summing.

3.2. Greedy objectives

A greedy approach aims to decrease/increase ϕ as quickly as possible, i.e., aiming to decrease/increase $\frac{\partial \phi}{\partial t}$. Typically, controls are applied in a time interval $[0, t_{\text{lag}}]$, aiming to reduce $\phi(t_{\text{lag}})$. Minimizing $\phi(t_{\text{lag}})$ is equivalent to minimizing the slope of the secant line between the endpoints of $[0, t_{\text{lag}}]$, which is equivalent to minimizing the derivative as $t_{\text{lag}} \to 0$. Minimizing $\phi(t_{\text{lag}})$ is also equivalent to minimizing the integral of $\frac{\partial \phi}{\partial t}$ from 0 to t_{lag}, illustrating that a greedy approach to minimizing $\frac{\partial \phi}{\partial t}$ could be obtained by minimizing $\phi(t_{\text{lag}})$ even when t_{lag} is not necessarily approaching 0.

Define the energy

$$E_\phi(t; \vec{\lambda}^P, \vec{\lambda}^E) = \frac{1}{2}\phi(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))^2 \tag{42}$$

with an equivalent definition for ϕ_{sphere} and $\tilde{\phi}_{\text{sphere}}$. Utilizing Eq. (35),

$$\frac{\partial E_\phi(t; \vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} = \phi(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) \frac{\partial \phi(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{\lambda}^{P/E}} = \vec{s}(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))^T \frac{\partial \vec{s}(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{\lambda}^{P/E}} \tag{43}$$

and for the spherical case

$$\frac{\partial E_{\phi_{\text{sphere}}}(t; \vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} = \frac{\phi_{\text{sphere}}(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\left\| \vec{s}(t; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) \right\|_2} \frac{\partial E_\phi(t; \vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} \tag{44}$$

where replacing ϕ_{sphere} with $\tilde{\phi}_{\text{sphere}}$ requires multiplication by -1 when $||\vec{s}||_2 < \eta$. Substituting Eq. (21) (with $t \in \mathbb{R}$, and ignoring the row for t^I entirely) into Eq. (43) leads to

$$\frac{\partial E_\phi(t; \vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} = \pm \sum_i s_i(t; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} T \begin{bmatrix} \frac{\partial x_{i,\text{left}}^{P/E}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}} \\ \frac{\partial v_{i,\text{left}}^{P/E}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}} \\ \frac{\partial x_{i,\text{right}}^{P/E}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}} \\ \frac{\partial v_{i,\text{right}}^{P/E}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}} \end{bmatrix} \tag{45}$$

valid for any $t \in [t_{\text{left}}, t_{\text{right}}]$.

Suppose the pursuer and evader each work with $E_\phi(t_{\text{lag}})$, perhaps each with a different t_{lag} (and even a different t_{right}, although we ignore this for simplicity). The pursuer aims to minimize the distance at their t_{lag} via $-\frac{\partial E_\phi(t_{\text{lag}})}{\partial \vec{\lambda}^P}$, while the evader aims to maximize the distance at their t_{lag} via $\frac{\partial E_\phi(t_{\text{lag}})}{\partial \vec{\lambda}^E}$. As one might expect, these derivatives have the same form for both players: the pursuer moves directly towards the evader and the evader moves directly away from the pursuer, so they both move in the same vector direction (assuming that they both use the same t_{lag}). Denote the steepest descent/ascent directions for the pursuer/evader respectively by $\mp \nabla^{P/E} E_\phi(t_{\text{lag}})$, which is the transpose of Eq. (45) with a " - " in place of the "$\pm$".

Suppose the game is composed of a single time interval and let both players' initial states $\vec{x}_{\text{left}}$ and $\vec{v}_{\text{left}}$ be given so that $\frac{\partial \vec{x}_{\text{left}}}{\partial \vec{\lambda}^{P/E}}$ and $\frac{\partial \vec{v}_{\text{left}}}{\partial \vec{\lambda}^{P/E}}$ are identically zero matrices. This allows simplification of the gradient to

$$\mp \nabla^{P/E} E_\phi(t_{\text{lag}}; \vec{\lambda}^P, \vec{\lambda}^E) = \begin{bmatrix} \frac{\partial \vec{x}_{\text{right}}^{P/E}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}}^T & \frac{\partial \vec{v}_{\text{right}}^{P/E}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}}^T \end{bmatrix} \begin{bmatrix} (\hat{t}_{\text{lag}})^2(2\hat{t}_{\text{lag}} - 3)\vec{s}(t_{\text{lag}}; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) \\ (\hat{t}_{\text{lag}})^2(t_{\text{right}} - t_{\text{left}})(1 - \hat{t}_{\text{lag}})\vec{s}(t_{\text{lag}}; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E)) \end{bmatrix} \tag{46}$$

where, along the lines of Eq. (9), $\hat{t}_{\text{lag}} = \frac{t - t_{\text{left}}}{t_{\text{right}} - t_{\text{left}}}$ has been used for the sake of simplicity. Let the controls for each player be their values of $\vec{x}_{\text{right}}$ and $\vec{v}_{\text{right}}$ so that

$$\frac{\partial \vec{x}_{\text{right}}^{P/E}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}}^T = \begin{bmatrix} I_{3\times 3} \\ 0_{3\times 3} \end{bmatrix}, \quad \frac{\partial \vec{v}_{\text{right}}^{P/E}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}}^T = \begin{bmatrix} 0_{3\times 3} \\ I_{3\times 3} \end{bmatrix} \tag{47}$$

and $\vec{\lambda}^{P/E} \in \mathbb{R}^6$. Substituting Eq. (47) into Eq. (46) turns the first matrix on the right-hand side into a 6×6 identity leaving only the second matrix, which is a 6×1 column vector. Since $2\hat{t}_{\text{lag}} - 3 < 0$, the pursuer aims for a positional change at t_{right} opposite the direction of $\vec{s}(t_{\text{lag}})$, which would decrease the distance since $\vec{s}(t_{\text{lag}})$ points from the evader to the pursuer. The evader also aims for a positional change opposite the direction of $\vec{s}(t_{\text{lag}})$, which would increase the distance. Since $1 - \hat{t}_{\text{lag}} > 0$, the pursuer aims for velocity change at t_{right} in the direction of $\vec{s}(t_{\text{lag}})$, which is the wrong direction. This occurs because increasing the derivative at t_{right} causes the values on $(t_{\text{left}}, t_{\text{right}})$ to decrease, even though the values on the next interval would increase. See Fig. 1. Thus, the pursuer mistakenly aims for a velocity change in the wrong direction in order to improve its displacement in $(t_{\text{left}}, t_{\text{right}})$. The evader has the same incorrect behavior. On the other hand, this strategy does lead to a local improvement in $(t_{\text{left}}, t_{\text{right}})$ that might be desirable.

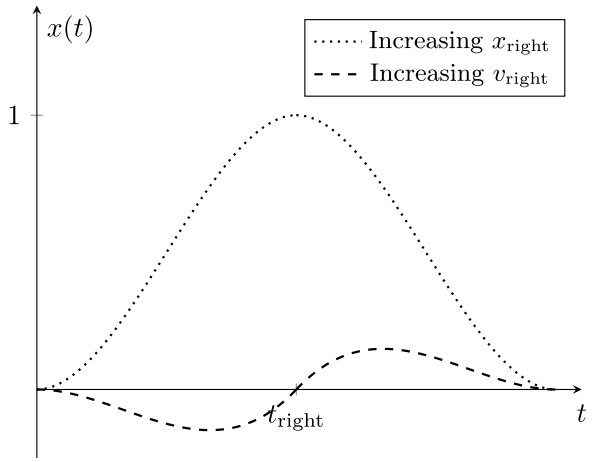

Fig. 1. A plot of one-dimensional cubic splines $x(t)$ that represent additive changes to x_{right} and v_{right}. Additive changes to x_{right} can be expressed by adding a scalar multiplier times a cubic that evaluates to zero at x_{left}, v_{left}, and v_{right}, while being equal to 1 at t_{right} (dotted line). This cubic is strictly positive in the open interval. Notably, continuity implies that the interval after t_{right} would similarly see an increase in the values of $x(t)$. Additive changes to v_{right} utilize a cubic that evaluates to zero at x_{left}, v_{left}, and x_{right}, while having a slope of 1 at t_{right} (dashed line). This lowers the values of $x(t)$ to the left of t_{right}, but raises the values of $x(t)$ to the right of t_{right}.

Most physical systems would have a velocity subservient to their positional goals. For example, let the controls for each player be their values for $\vec{x}_{\text{right}}$ and let

$$\vec{v}_{\text{right}}(\vec{\lambda}^{P/E}) = \frac{\vec{x}_{\text{right}}(\vec{\lambda}^{P/E}) - \vec{x}_{\text{left}}}{t^{P/E}_{\text{right}} - t^{P/E}_{\text{left}}} = \frac{\vec{x}_{\text{right}}(\vec{\lambda}^{P/E}) - \vec{x}_{\text{left}}}{\Delta t^{P/E}} \tag{48}$$

be subservient to $\vec{x}_{\text{right}}$; then,

$$\frac{\partial \vec{x}_{\text{right}}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}}^{T} = I_{3\times 3}, \quad \frac{\partial \vec{v}_{\text{right}}(\vec{\lambda}^{P/E})}{\partial \vec{\lambda}^{P/E}}^{T} = \frac{1}{\Delta t^{P/E}} I_{3\times 3} \tag{49}$$

replaces Eq. (47). Substituting Eq. (49) into Eq. (46) leads to

$$\mp \nabla^{P/E} E_{\phi}(t_{\text{lag}}; \vec{\lambda}^{P}, \vec{\lambda}^{E}) = (\hat{t}_{\text{lag}})^2 (\hat{t}_{\text{lag}} - 2) \vec{s}(t_{\text{lag}}; \vec{\mathbf{p}}^{s}(\vec{\lambda}^{P}, \vec{\lambda}^{E})) \tag{50}$$

which is always in the correct direction of $-\vec{s}(t_{\text{lag}})$ since $\hat{t}_{\text{lag}} - 2 < 0$. The gradient vanishes when $\vec{s}(t_{\text{lag}}) = 0$, which is correct for the pursuer, but undesirable for the evader; driven by gradient ascent, the evader can move in any direction to avoid the potential collision at $t = t_{\text{lag}}$. Instead of a constant velocity, the trajectory could be modeled with a constant acceleration via

$$\vec{x}^{P/E}_{\text{right}}(\vec{\lambda}^{P/E}) = \vec{x}^{P/E}_{\text{left}} + \Delta t^{P/E} \vec{v}^{P/E}_{\text{left}} + \frac{(\Delta t^{P/E})^2}{2} \vec{a}^{P/E} \tag{51a}$$

$$\vec{v}^{P/E}_{\text{right}}(\vec{\lambda}^{P/E}) = \vec{v}^{P/E}_{\text{left}} + \Delta t^{P/E} \vec{a}^{P/E} = 2 \frac{\vec{x}^{P/E}_{\text{right}}(\vec{\lambda}^{P/E}) - \vec{x}^{P/E}_{\text{left}}}{\Delta t^{P/E}} - \vec{v}^{P/E}_{\text{left}} \tag{51b}$$

where the players each still control their $\vec{x}_{\text{right}}$. The velocity derivative in Eq. (49) then includes an extra 2 on the numerator; then, $\hat{t}_{\text{lag}} - 2$ is replaced by -1 in Eq. (50) and the gradient once again points in the correct direction.

3.3. Root-forecasting objectives

Instead of a greedy approach, it might make more sense to increase or decrease ϕ at the time at which it is smallest. This leads to analyzing the real roots of ψ. Although the complex roots of ψ correspond to monotonic behavior, complex roots can merge to form real roots implying that they need to be considered as well. Thus, $\psi(t_{\text{min}}) = 0$ with $t_{\text{min}} \in C$ is considered in this subsection.

Since merging complex roots produce local extrema with values close to that of $\phi(t^{R}_{\text{min}})$ just before the complex roots merged, it makes sense to aim to increase/decrease $\phi(t^{R}_{\text{min}})$ even when $t_{\text{min}} \in \mathbb{C}$. Consider minimizing/maximizing

$$E_{\phi}(t^{R}_{\text{min}}; \vec{\lambda}^{P}, \vec{\lambda}^{E}) = \frac{1}{2} \phi(t^{R}_{\text{min}}; \vec{\mathbf{p}}^{s}(\vec{\lambda}^{P}, \vec{\lambda}^{E}))^2 \tag{52}$$

so that ϕ is either close to or far from zero if t^{I}_{min} does become zero making $\psi(t^{R}_{\text{min}}) = 0$. Importantly, t^{R}_{min} depends on the roots and thus depends on $\vec{\lambda}$, whereas t_{lag} from Section 3.2 was chosen arbitrarily and thus is not dependent on $\vec{\lambda}$. Utilizing the definition of ψ

in Eq. (28), Eq. (52) leads to

$$\frac{\partial E_\phi(t^R_{\min};\vec{\lambda}^P,\vec{\lambda}^E)}{\partial\vec{\lambda}^{P/E}} = \phi(t^R_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))\frac{\partial\phi(t^R_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial\vec{\lambda}^{P/E}} + \psi(t^R_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))\frac{\partial t^R_{\min}}{\partial\vec{\lambda}^{P/E}} \tag{53}$$

where $\frac{\partial t^R_{\min}}{\partial\vec{\lambda}^{P/E}}$ is defined in Eq. (41). One can similarly derive

$$\frac{\partial E_{\phi_{\text{sphere}}}(t^R_{\min};\vec{\lambda}^P,\vec{\lambda}^E)}{\partial\vec{\lambda}^{P/E}} = \frac{\phi_{\text{sphere}}(t^R_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\left\|\vec{s}(t^R_{\min};\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))\right\|_2}\frac{\partial E_\phi(t^R_{\min};\vec{\lambda}^P,\vec{\lambda}^E)}{\partial\vec{\lambda}^{P/E}} \tag{54}$$

illustrating that $E_{\phi_{\text{sphere}}}$ behaves similarly to E_ϕ (as before, replacing ϕ_{sphere} with $\tilde{\phi}_{\text{sphere}}$ requires multiplying by -1 when $||\vec{s}||_2 < \eta$). Regarding the tractability of this objective function, finding $t^R_{\min}$ requires computing the roots of a fifth order polynomial ψ. In addition, finding each $\frac{\partial t_{\min}}{\partial\vec{p}^s_i}$ in Eq. (41) requires solving Eq. (37b), which aims to find the derivatives of the roots of a fifth order polynomial with respect to its coefficients (no robust method for doing so yet exists).

When dealing with real roots of ψ where $\psi(t^R_{\min}) = 0$, the last term in Eq. (53) vanishes making it identical to Eq. (43). The only difference between this and the objective function for t_{lag} is that $t^R_{\min}$ is constrained to have $\psi(t^R_{\min}) = 0$ forcing $t^R_{\min}$ to be a relative extrema of ϕ. This means that the pursuer is working to minimize distance at the relative extrema of ϕ. Minimizing distance at relative minima of ϕ is the quickest way to reduce the minimal separation distance. Minimizing distance at a relative maximum of ϕ reduces the separation distance required to turn around and start getting closer once again, perhaps eliminating some evasive maneuvers of the evader. The evader opposes these goals.

Remark 3.5. *The reader may have noticed that we do not differentiate inflection points from relative minima/maxima, since we treat an inflection point as a coincident minimum and maximum consistent with a repeated real root.*

Consider the first term in Eq. (53). When dealing with complex roots of ψ where $\psi(t^R_{\min}) \neq 0$, ϕ has a nonzero slope $\psi(t^R_{\min})$ at $t^R_{\min}$. When $\psi(t^R_{\min}) > 0$, the pursuer prefers merging $t_{\min}$ with its complex conjugate in order to create a relative maximum so that ϕ flips from increasing to decreasing. When $\psi(t^R_{\min}) < 0$, the pursuer prefers to keep ϕ decreasing by maintaining $t^I_{\min} \neq 0$. In both cases, minimizing ϕ at $t^R_{\min}$ according to the first term in Eq. (53) is beneficial to the pursuer. The evader has competing goals, and would similarly aim to maximize ϕ at $t^R_{\min}$ according to the first term in Eq. (53). That is, the first term on the right-hand side of Eq. (53) is well motivated whether or not $\psi(t^R_{\min}) = 0$.

Consider the second term in Eq. (53). When $\psi(t^R_{\min}) > 0$, the pursuer prefers to decrease $t^R_{\min}$ so that the aspirational real-valued root (which would make ϕ start to decrease) occurs as early as possible; thus, the steepest descent direction for $t^R_{\min}$ should be (and is) followed, via the negative transpose of Eq. (53). When $\psi(t^R_{\min}) < 0$, the pursuer prefers to increase $t^R_{\min}$ so that any potential real-valued root (which would make ϕ start to increase) is delayed to be as far out into the future as possible; thus, the steepest ascent direction for $t^R_{\min}$ should be (and is) followed, via the negative transpose of Eq. (53). The evader has competing goals, which are achieved via the (positive) transpose of Eq. (53).

Remark 3.6. *Note that the transpose of $\frac{\partial t^R_{\min}}{\partial\vec{\lambda}^{P/E}}$ in Eq. (53) could be used to move both real roots and the real parts of complex roots around, moving desirable roots into intervals under consideration and undesirable roots out of such intervals.*

3.4. Root-controlling objectives

Modifying the naive greedy approach in Section 3.2 to more strategically consider the relative extrema of ϕ in Section 3.3 led quite naturally to a desire to control the roots of ψ. Although the objective function in Section 3.3 provided insight on how to move real roots and the real parts of complex roots around in time, a mechanism for merging complex roots to become real-valued and vice versa did not emerge. Thus, root-controlling objectives are discussed in this section.

Remark 3.7. *The greedy approach from Section 3.2 aims to minimize/maximize $\phi(t_{lag})$. When the player working against the current monotonicity is successful, ϕ flips from increasing to decreasing or vice versa. This happens when complex conjugate roots of ψ merge into a pair of real roots, representing a new relative maximum and minimum. Thus, the greedy approach is also concerned with root merging; however, it only considers the roots implicitly, likely missing out on various potential strategies.*

When aiming to control complex-valued roots, consider minimizing/maximizing

$$\tilde{E}_{t_{\min}}(t_{\min};\vec{\lambda}^P,\vec{\lambda}^E) = \frac{1}{2}t^*_{\min}t_{\min} = \frac{1}{2}(t^R_{\min})^2 + \frac{1}{2}(t^I_{\min})^2 \tag{55a}$$

$$E_{t_{\min}}(t_{\min};\vec{\lambda}^P,\vec{\lambda}^E) = t^R_{\min} + |t^I_{\min}| \tag{55b}$$

where Eq. (55b) provides more uniform gradients. Note that the absolute value on $t^I_{\min}$ properly treats whichever complex conjugate is being considered. Eq. (55b) leads to

$$\frac{\partial E_{t_{\min}}(t_{\min};\vec{\lambda}^P,\vec{\lambda}^E)}{\partial\vec{\lambda}^{P/E}} = \begin{bmatrix}1 & S(t^I_{\min})\end{bmatrix}\frac{\partial t_{\min}}{\partial\vec{\lambda}^{P/E}} \tag{56}$$

where $\frac{\partial t_{\min}}{\partial \vec{\lambda}^{P/E}}$ is defined in Eq. (41) and requires finding the derivatives of the roots of a fifth order polynomial with respect to its coefficients (by solving Eq. (37b)). When aiming to merge complex conjugates, the $\frac{1}{2}(t^I_{\min})^2$ term in Eq. (55a) might be preferred over the $|t^I_{\min}|$ term in Eq. (55b), since replacing $S(t^I_{\min})$ with $t^I_{\min}$ in Eq. (56) can help to prevent overshoots as $t^I_{\min} \to 0$; alternatively, a smoothed sign function (or a soft ℓ_1) could be used. When instead aiming to separate complex conjugates, this smoothing can lead to erroneously small gradients; thus, is should be avoided with $S(0) = \pm 1$ depending on which complex conjugate is being considered. When aiming to control real-valued roots, consider minimizing or maximizing

$$E_{\text{pair}}(t^+_{\min}, t^-_{\min}; \vec{\lambda}^P, \vec{\lambda}^E) = -\left|t^{R,+}_{\min} - t^{R,-}_{\min}\right| \tag{57}$$

as suggested in [18]. Typically, the pursuer would aim to minimize E_{pair} when the roots are close together in order to avoid a potential merge. On the other hand, the evader may work to merge roots even when they are quite far away, creating a slight asymmetry. The boundary between the two aforementioned cases (Eqs. (55) and (57)) is the repeated root case, which requires special treatment as discussed in [18].

As discussed in Section 3.3 the sign of $\psi(t^R_{\min})$ for a complex-valued root $t_{\min}$ indicates whether ϕ is locally increasing or decreasing. When $\psi(t^R_{\min}) > 0$, the pursuer aims to merge the complex conjugate roots to change ϕ from increasing to decreasing. When $\psi(t^R_{\min}) < 0$, the pursuer aims to resist such merging in order to maintain a decreasing ϕ. The evader works against these goals. The sign of ψ can be incorporated into the last term in Eq. (55b) by modifying it to

$$E_{t^I_{\min}}(t_{\min}; \vec{\lambda}^P, \vec{\lambda}^E) = S\left(\psi(t^R_{\min}; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))\right)|t^I_{\min}| \tag{58}$$

where the sign of zero could be treated as identically zero, since it corresponds to a real root where both the pursuer and evader will instead temporarily only focus on the first term on the righthand side of Eq. (53); on the other hand, it could make more sense to use the sign of ψ at $t^R_{\min} - \epsilon$ when the roots become real, since that represents the monotonicity that ϕ would have if $t^I_{\min}$ were to become nonzero again. Eq. (58) leads to

$$\frac{\partial E_{t^I_{\min}}(t_{\min}; \vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} = S\left(\psi(t^R_{\min}; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))\right) S(t^I_{\min}) \frac{\partial t^I_{\min}}{\partial \vec{\lambda}^{P/E}} \tag{59}$$

utilizing the fact that the derivative of $S(\psi)$ is 0 almost everywhere (and that the discontinuity is unimportant, since the treatment of Eq. (59) would typically be split into two separate cases). When $\psi(t^R_{\min}) > 0$, the pursuer uses the negative transpose of Eq. (59) following steepest descent on the magnitude of $t^I_{\min}$ to merge the complex conjugate roots as desired. When $\psi(t^R_{\min}) < 0$, the pursuer utilizes steepest ascent. The evader works against these goals.

4. Root-finding formulation

A collision (or interaction based on proximity) between two agents occurs in some interval when all of the components of $\vec{s}$ in that interval satisfy

$$s_i(t_{\text{root}}; \vec{p}^s_i(\vec{\lambda}^P, \vec{\lambda}^E)) = 0 \tag{60}$$

with $t_{\text{root}} \in \mathbb{R}$. The roots of Eq. (60) can be complex with $t_{\text{root}} \in \mathbb{C}$ and thus $s_i \in \mathbb{C}$ (although $s_i \in \mathbb{R}$ along any real trajectory). The pursuer would aim for a $t_{\text{root}} \in \mathbb{R}$ while the evader would aim to prevent this. To motivate strategies utilizing rootfinding, a pursuer seeking a collision as quickly as possible might aim to minimize $t^R_{\text{root}} \geq 0$ via

$$\min_{\vec{\lambda}^P} \max_{\vec{\lambda}^E} \left(\alpha^+ H(t^R_{\text{root}})\delta(t^I_{\text{root}}) - \alpha^- H(-t^R_{\text{root}})\right) t^R_{\text{root}} + \beta |t^I_{\text{root}}| \tag{61}$$

where $\beta > 0$ encourages a collision with $t^I_{\text{root}} = 0$, the $H(-t^R_{\text{root}})$ term with $\alpha^- > 0$ encourages $t^R_{\text{root}} \geq 0$, and the delta function $\delta(t^I_{\text{root}})$ term with $\alpha^+ > 0$ encourages a small $t^R_{\text{root}} \geq 0$ when a collision is predicted with $\delta(0) = 1$. The evader would work against this, via

$$\max_{\vec{\lambda}^E} \min_{\vec{\lambda}^P} \left(\alpha^+ H(t^R_{\text{root}})\delta(t^I_{\text{root}}) - \alpha^- H(-t^R_{\text{root}})\right) t^R_{\text{root}} + \beta |t^I_{\text{root}}| \tag{62}$$

encouraging t^I_{root} to be far from zero, a negative t^R_{root} to be far from zero, and a positive t^R_{root} to be far from zero when a collision is predicted.

Eq. (60) requires $t_{\text{root}} \in \mathbb{C}$ to simultaneously solve all three cubics, which might not be possible. Solution nonexistence contradicts our premise that unobtainable events can be modeled simply via complex roots. This can be remedied by considering

$$s_i(t_{\text{root},i}; \vec{p}^s_i(\vec{\lambda}^P, \vec{\lambda}^E)) = 0 \tag{63}$$

where each cubic is solved by a separate $t_{\text{root},i} \in \mathbb{C}$. The pursuer would aim to cluster the three $t_{\text{root},i}$ into a coincident $t_{\text{root},i} \in \mathbb{R}$, while the evader would aim to prevent this.

When considering spherical spatial extent, setting $\vec{s}_{\text{sphere}}$ from Eq. (16) to be identically zero results in

$$\left\|\vec{s}(t_{\text{root}}; \vec{\mathbf{p}}^s(\vec{\lambda}^P, \vec{\lambda}^E))\right\|_2 = \eta \tag{64}$$

by setting the scalar component of the unit vector $\frac{\vec{s}}{\|\vec{s}\|_2}$ to zero. This leads to a sixth order polynomial

$$\sum_i s_i(t_{\text{root}}; \vec{p}^s_i(\vec{\lambda}^P, \vec{\lambda}^E))^2 = \eta^2 \tag{65}$$

for $t_{\text{root}} \in C$. When $\eta = 0$, all of the s_i are fully decoupled from each other leading to Eq. (60). When $\eta \neq 0$, the three cubics are coupled since it is unclear how much of η should be appropriated to each of them. Eq. (65) leads to

$$\sum_i \left(s_i^R(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \right)^2 - \left(s_i^I(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \right)^2 = \eta^2 \tag{66a}$$

$$\sum_i s_i^R(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) s_i^I(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) = 0 \tag{66b}$$

after independently setting its real and imaginary components to zero.

Remark 4.1. *One might be tempted to utilize Eq. (31) in order to define the norm in Eq. (64). This violates the spirit of our proposed methodology, which consists of specifying equations that define an event and using real-valued and complex-valued roots to determine whether the event does or does not occur. If one were to utilize Eq. (31) in Eq. (64), it would lead to*

$$\sum_i s_i^R(t_{root}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))^2 + s_i^I(t_{root}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))^2 = \eta^2 \tag{67}$$

for $t_{root} \in \mathbb{C}$. Not only does Eq. (67) differ from Eq. (66a), but Eq. (67) is only a single equation for two unknowns (t_{root}^R and t_{root}^I). There are a few ways to proceed, for example, one might set

$$\sum_i s_i^I(t_{root}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))^2 = 0 \tag{68}$$

in order to specify that all of the s_i^I should be identically zero. This can also be written as

$$s_i^I(t_{root}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) = 0 \tag{69}$$

in order to emphasize that it represents three separate constraints. Remark 4.3 illustrates the issues with this approach.

A more tractable approach that avoids sixth order polynomials for spherical spatial extent consists of choosing a point $\vec{\eta}$ on the sphere and aiming to collide with it, i.e.,

$$s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) = \eta_i \tag{70a}$$

$$\sum_i \eta_i^2 = \eta^2 \tag{70b}$$

noting that acceptable solutions have $\eta_i \in \mathbb{R}$. Substituting Eq. (70a) into Eq. (70b), to eliminate the dependence on, and thus the need for the knowledge of, the individual η_i results in Eq. (65) as expected. As the sphere shrinks to a point, i.e. $\eta \to 0$, the η_i naturally disappear leading to Eq. (60). Eq. (70a) can be replaced by

$$s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) = \eta_i \tag{71}$$

with the pursuer working to cluster the three real roots. Note that the η_i could be treated as parameters that are updated each iteration, so that the more tractable cubic polynomials in Eq. (71) can be used in place of the sixth order polynomial in Eq. (65). In fact, it would make more sense to navigate the surface of the sphere by treating only η_1 and η_2 as parameters, where the sign in $\eta_3 = \pm\sqrt{\eta^2 - \eta_1^2 - \eta_2^2}$ should be chosen to be consistent with the intended hemisphere.

Remark 4.2. *The pursuer can discretize a sphere around the evader and consider each discretized point as a separate $\vec{\eta}$. The evader, however, cannot similarly discretize a sphere around the pursuer, since avoiding every discretized point still allows collision with the sphere.*

4.1. Differentials

Eq. (63) implies that $ds_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) = 0$ which leads to

$$\frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root},i}} dt_{\text{root},i} + \frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} d\vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) = 0 \tag{72a}$$

$$\frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root},i}} \frac{\partial t_{\text{root},i}}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} = -\frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} \tag{72b}$$

where $\frac{\partial s_i}{\partial t_{\text{root},i}}$ and $\frac{\partial s_i}{\partial \vec{p}_i^s}$ are defined via Eq. (19). This formulation for the derivatives of the roots of a cubic polynomial with respect to its coefficients is discussed in detail in [18]. Substituting $d\vec{p}_i^s$ from Eq. (20) into Eq. (72a) leads to

$$\frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root},i}} \frac{\partial t_{\text{root},i}}{\partial \vec{\lambda}^{P/E}} = -\frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} \frac{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} \tag{73}$$

and substituting Eq. (72b) into Eq. (73) leads to

$$\frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root},i}} \frac{\partial t_{\text{root},i}}{\partial \vec{\lambda}^{P/E}} = \frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root},i}} \frac{\partial t_{\text{root},i}}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} \frac{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} \tag{74}$$

which gives

$$\frac{\partial t_{\text{root},i}}{\partial \vec{\lambda}^{P/E}} = \frac{\partial t_{\text{root},i}}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} \frac{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} \tag{75}$$

when one can reliably invert $\frac{\partial s_i}{\partial t_{\text{root},i}}$. Eq. (75) dictates solving Eq. (72b) for $\frac{\partial t_{\text{root},i}}{\partial \vec{p}_i^s}$ and subsequently multiplying the result by the partial derivative in Eq. (20).

For spherical spatial extent, the total derivatives of Eq. (66) can be expressed compactly as the system

$$\sum_i \mathcal{S}_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \left(\frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root}}} dt_{\text{root}} + \frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} d\vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) \right) = 0 \tag{76}$$

where

$$\mathcal{S}_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) = \begin{bmatrix} s_i^R(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) & -s_i^I(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \\ s_i^I(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) & s_i^R(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \end{bmatrix} \tag{77}$$

is the matrix equivalent for multiplying by a complex number. From Eq. (76), the partial derivatives are defined implicitly as

$$\left(\sum_i \mathcal{S}_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root}}} \right) \frac{\partial t_{\text{root}}}{\partial \vec{p}_j^s(\vec{\lambda}^P, \vec{\lambda}^E)} = -\mathcal{S}_j(t_{\text{root}}; \vec{p}_j^s(\vec{\lambda}^P, \vec{\lambda}^E)) \frac{\partial s_j(t_{\text{root}}; \vec{p}_j^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_j^s(\vec{\lambda}^P, \vec{\lambda}^E)} \tag{78}$$

for $j = 1, 2, 3$. Solving Eq. (78) for $\frac{\partial t_{\text{root}}}{\partial \vec{p}_j^s}$ poses significant difficulties, since the 2×2 coefficient matrix contains fifth order polynomials; in fact, $\frac{\partial t_{\text{root}}}{\partial \vec{p}_j^s}$ is the derivative of the root of the sixth order polynomial in Eq. (65) with respect to the coefficients of one of the third order polynomials that are squared and summed to compose it.

Note that a nonzero $\mathcal{S}_j$ is required in order to formally derive Eq. (78) from Eq. (76); otherwise, $d\vec{p}_j^s$ drops out of Eq. (76). This occurs on the three great circles obtained by intersecting a sphere of radius η with each of the three coordinate planes. To see this, suppose that the t_{root} are constrained to lie on the great circle given by the two dimensions $i = 1, 2$; then, Eq. (76) only sums over $i \neq 3$ and Eq. (78) is correct for $j = 1, 2$. In order to remain on the circle, the third dimension requires the constraint $s_3(t_{\text{root}}; \vec{p}_3^s(\vec{\lambda}^P, \vec{\lambda}^E)) = 0$ implying that $\frac{\partial t_{\text{root}}}{\partial \vec{p}_3^s}$ satisfies Eq. (72) with $t_{\text{root},i}$ replaced by t_{root}. That is, the third dimension only uses the $i = 3$ term in Eq. (76), meaning that multiplication by $\mathcal{S}_3$ can be ignored. From a practical standpoint, temporarily rotating the coordinate system can be used to alleviate this issue.

Remark 4.3. *The total derivatives of Eqs. (67) and (68) from Remark 4.1 also lead to Eqs. (76) and (78) except with*

$$\tilde{\mathcal{S}}_i(t_{root}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) = \begin{bmatrix} s_i^R(t_{root}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) & 0 \\ 0 & s_i^I(t_{root}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \end{bmatrix} \tag{79}$$

replacing Eq. (77). The constraint that $\sum_i s_i^I(t_{root})^2 = 0$ implies that each $s_i^I(t_{root})^2 = 0$, meaning that the second row of Eq. (78) using Eq. (79) is trivially satisfied. Another way to see this is that $\sum_i s_i^I(t_{root})^2 = 0$ defines t_{root} as points where the implicit function theorem fails, since

$$\frac{\partial}{\partial t} s_i^I(t)^2 \Big|_{t=t_{root}} = 2 s_i^I(t_{root}) \frac{\partial s_i^I(t)}{\partial t} \Big|_{t=t_{root}} = 0 \tag{80}$$

and $s_i^I(t_{root}) = 0$ is always problematic. Alternatively, Eqs. (67) and (69) lead to

$$\begin{bmatrix} \sum_i s_i^R(t_{root}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \frac{\partial s_i^R(t_{root}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{root}} \\ \frac{\partial s_j^I(t_{root}; \vec{p}_j^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{root}} \end{bmatrix} \frac{\partial t_{root}}{\partial \vec{p}_j^s(\vec{\lambda}^P, \vec{\lambda}^E)} = -\begin{bmatrix} s_j^R(t_{root}; \vec{p}_j^s(\vec{\lambda}^P, \vec{\lambda}^E)) & 0 \\ 0 & 1 \end{bmatrix} \frac{\partial s_j(t_{root}; \vec{p}_j^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_j^s(\vec{\lambda}^P, \vec{\lambda}^E)} \tag{81}$$

replacing Eq. (78). The gradient direction in Eq. (81) would keep $\sum_i s_i^R(t_{root})^2$ constant as well as $s_j^I(t_{root}) = 0$, but would vary the other $s_k^I(t_{root})$ for $k \neq j$ (since t_{root} couples the s_i), violating the constraint in Eq. (69).

Substituting $d\vec{p}_i^s$ from Eq. (20) into Eq. (76) leads to

$$\begin{aligned} &\left(\sum_i \mathcal{S}_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root}}} \right) \frac{\partial t_{\text{root}}}{\partial \vec{\lambda}^{P/E}} \\ &\quad = -\sum_i \mathcal{S}_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} \frac{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} \end{aligned} \tag{82}$$

and substituting Eq. (78) into Eq. (82) three times leads to

$$\left(\sum_i \mathcal{S}_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root}}} \right) \frac{\partial t_{\text{root}}}{\partial \vec{\lambda}^{P/E}}$$

$$= \left(\sum_i S_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)) \frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root}}} \right) \sum_i \frac{\partial t_{\text{root}}}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} \frac{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} \tag{83}$$

which gives

$$\frac{\partial t_{\text{root}}}{\partial \vec{\lambda}^{P/E}} = \sum_i \frac{\partial t_{\text{root}}}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} \frac{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} \tag{84}$$

when one can reliably invert the coefficient matrix. Eq. (84) dictates solving Eq. (78) separately for each of the $\frac{\partial t_{\text{root}}}{\partial \vec{p}_i^s}$ and subsequently multiplying each result with the partial derivative in Eq. (20) before summing.

For the more tractable approach to spherical spatial extent given in Eq. (70),

$$\frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root}}} dt_{\text{root}} + \frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} d\vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) = \begin{bmatrix} d\eta_i^R \\ d\eta_i^I \end{bmatrix} \tag{85a}$$

$$\sum_i \begin{bmatrix} \eta_i^R & -\eta_i^I \\ \eta_i^I & \eta_i^R \end{bmatrix} \begin{bmatrix} d\eta_i^R \\ d\eta_i^I \end{bmatrix} = 0 \tag{85b}$$

noting that Eq. (85b) is identical to Eq. (76) based on the definitions of η_i and $d\eta_i$. Making the leap to consider the η_i as independent degrees of freedom as opposed to removable by constraint would lead to

$$\frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root}}} \frac{\partial t_{\text{root}}}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} = -\frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} \tag{86a}$$

$$\frac{\partial s_i(t_{\text{root}}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root}}} \frac{\partial t_{\text{root}}}{\partial \eta_i} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{86b}$$

which is significantly simpler than Eq. (78). However, perturbing the parameters in Eq. (70a) while keeping η_i constant, which is required in order to define the partial derivative according to Eq. (86a), perturbs t_{root} in the i-th equation to be inconsistent with the t_{root} in the other two equations (obviously, since Eq. (78) dictates the correct perturbation of t_{root} with respect to perturbations in the parameters). Thus, this approach only works when a separate $t_{\text{root},i} \in \mathbb{C}$ is employed for each cubic equation, replacing Eq. (70a) with Eq. (71). This leads to

$$\frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root},i}} dt_{\text{root},i} + \frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} d\vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E) = \begin{bmatrix} d\eta_i^R \\ d\eta_i^I \end{bmatrix} \tag{87}$$

in place of Eq. (85a) and

$$\frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root},i}} \frac{\partial t_{\text{root},i}}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} = -\frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E)} \tag{88a}$$

$$\frac{\partial s_i(t_{\text{root},i}; \vec{p}_i^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root},i}} \frac{\partial t_{\text{root},i}}{\partial \eta_i} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{88b}$$

in place of Eq. (86). Note that Eq. (88a) is identical to Eq. (72b) and thus tractable. Moreover, Eq. (88b) has the same coefficient matrix as Eq. (88a) and thus should be similarly quite tractable.

Assuming η_1 and η_2 are the degrees of freedom with η_3 specified via a constraint, Eq. (85b) can be rewritten as

$$\begin{bmatrix} d\eta_3^R \\ d\eta_3^I \end{bmatrix} = \frac{-1}{(\eta_3^R)^2 + (\eta_3^I)^2} \begin{bmatrix} \eta_3^R & \eta_3^I \\ -\eta_3^I & \eta_3^R \end{bmatrix} \sum_{i \neq 3} \begin{bmatrix} \eta_i^R & -\eta_i^I \\ \eta_i^I & \eta_i^R \end{bmatrix} \begin{bmatrix} d\eta_i^R \\ d\eta_i^I \end{bmatrix} \tag{89}$$

and substituted into Eq. (87). This leads to a modification of the $i = 3$ version of Eq. (88b)

$$\frac{\partial s_3(t_{\text{root},3}; \vec{p}_3^s(\vec{\lambda}^P, \vec{\lambda}^E))}{\partial t_{\text{root},3}} \frac{\partial t_{\text{root},3}}{\partial \eta_i} = \frac{-1}{(\eta_3^R)^2 + (\eta_3^I)^2} \begin{bmatrix} \eta_3^R & \eta_3^I \\ -\eta_3^I & \eta_3^R \end{bmatrix} \begin{bmatrix} \eta_i^R & -\eta_i^I \\ \eta_i^I & \eta_i^R \end{bmatrix} \tag{90}$$

governing the dependence of $t_{\text{root},3}$ on η_1 and η_2.

Remark 4.4. *Substituting Eq. (20) into Eq. (87) leads to Eq. (73), and substituting Eq. (88a) into Eq. (73) leads to Eqs. (74) and (75). That is, the dependencies of the $t_{root,i}$ on the controls $\vec{\lambda}$ does not fundamentally change when new controls η_1 and η_2 are added to navigate a point of interest on the sphere.*

4.2. Root-finding objectives

The root-finding formulation is arguably the best way to continue the discussion on objective functions from Sections 3.2–3.4. Rewriting $E_{t_{\text{min}}}$ from Eq. (55b) with $t_{\text{root},i}$ gives

$$E_{t_{\text{root},i}}(t_{\text{root},i}; \vec{\lambda}^P, \vec{\lambda}^E) = t_{\text{root},i}^R + |t_{\text{root},i}^I| \tag{91}$$

where the absolute value on $t^I_{\text{root},i}$ properly treats whichever complex conjugate is being considered. Note that ψ (see Eq. (58)) is not considered, since the pursuer always prefers real-valued roots and the evader always prefers complex-valued roots in the root-finding formulation. Similar to Eq. (57), the evader may aim to merge two distinct real roots $t^{R,+}_{\text{root},i}$ and $t^{R,-}_{\text{root},i}$ by maximizing

$$E_{\text{pair}}(t^{R,+}_{\text{root},i}, t^{R,-}_{\text{root},i}; \vec{\lambda}^P, \vec{\lambda}^E) = -\left| t^{R,+}_{\text{root},i} - t^{R,-}_{\text{root},i} \right| \tag{92}$$

while the pursuer would aim to minimize E_{pair} when the real roots are too close together.

In addition to controlling individual roots, a term is required for clustering (or resisting clustering) of the roots in order to create (or avoid) a coincident real root and thus a collision. For example, the pursuer might minimize

$$E_{\text{cluster}}(\vec{t}_{\text{root}}; t_{\text{cluster}}, \vec{\lambda}^P, \vec{\lambda}^E) = \frac{1}{2}\sum_i (t^R_{\text{root},i} - t_{\text{cluster}})^2 + \sum_i |t^I_{\text{root},i}| \tag{93}$$

for some target $t_{\text{cluster}} \in \mathbb{R}$. Instead of treating t_{cluster} as a parameter it can be defined via

$$t_{\text{cluster}} = \frac{1}{3}\sum_{i=1}^{3} t^R_{\text{root},i} \tag{94}$$

so that it depends on the $t_{\text{root},i}$. Eq. (94) is more appropriate for an evader's attempts to prevent clustering, since it is unlikely that they would know a pursuer's target t_{cluster} value. Since the $t^R_{\text{root},i}$ terms in Eqs. (91) and (93) can conflict, it is more appropriate to replace the first term in Eq. (91) with

$$E_{t_{\text{cluster}}}(t_{\text{cluster}}; \vec{\lambda}^P, \vec{\lambda}^E) = S(t_{\text{cluster}}) t_{\text{cluster}} \tag{95}$$

with t_{cluster} defined via Eq. (94). $S(t_{\text{cluster}})$ has been added on behalf of the pursuer in order to aim for an increasing t_{cluster} when $t_{\text{cluster}} < 0$. Note that the second term in Eq. (91) already appears in Eq. (93). This leads to a combined objective

$$E_c(\vec{t}_{\text{root}}; t_{\text{cluster}}, \vec{\lambda}^P, \vec{\lambda}^E) = \frac{1}{2}\sum_i (t^R_{\text{root},i} - t_{\text{cluster}})^2 + \sum_i |t^I_{\text{root},i}| + S(t_{\text{cluster}}) t_{\text{cluster}} \tag{96}$$

composed of E_{cluster} and $E_{t_{\text{cluster}}}$.

The derivative of Eq. (93) is

$$\frac{\partial E_{\text{cluster}}(\vec{t}_{\text{root}}; t_{\text{cluster}}, \vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} = \sum_i (t^R_{\text{root},i} - t_{\text{cluster}}) \left(\frac{\partial t^R_{\text{root},i}}{\partial \vec{\lambda}^{P/E}} - \frac{\partial t_{\text{cluster}}}{\partial \vec{\lambda}^{P/E}} \right) + \sum_i S(t^I_{\text{root},i}) \frac{\partial t^I_{\text{root},i}}{\partial \vec{\lambda}^{P/E}} \tag{97}$$

whether t_{cluster} is constant or defined by Eq. (94). Eq. (97) can be rewritten as

$$\frac{\partial E_{\text{cluster}}(\vec{t}_{\text{root}}; t_{\text{cluster}}, \vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} = \sum_i \left[t^R_{\text{root},i} - t_{\text{cluster}} \quad S(t^I_{\text{root},i}) \right] \frac{\partial t_{\text{root},i}}{\partial \vec{\lambda}^{P/E}} \tag{98}$$

when t_{cluster} is constant or equivalently

$$\frac{\partial E_{\text{cluster}}(\vec{t}_{\text{root}}; t_{\text{cluster}}, \vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} = \sum_i \left[t^R_{\text{root},i} - \tfrac{1}{3}\textstyle\sum_{j=1}^{3} t^R_{\text{root},j} \quad S(t^I_{\text{root},i}) \right] \frac{\partial t_{\text{root},i}}{\partial \vec{\lambda}^{P/E}} \tag{99}$$

when t_{cluster} is defined in Eq. (94). Note that arriving at Eq. (99) requires some algebraic simplification. Note that $\frac{\partial t_{\text{root},i}}{\partial \vec{\lambda}^{P/E}}$ is defined in Eq. (75) and requires solving Eq. (72b) via the approach proposed in [18]. In the spherical spatial extent case, $\frac{\partial t_{\text{root}}}{\partial \vec{\lambda}^{P/E}}$ defined via Eq. (84) requires solving Eq. (78), which would require a robust extension of the approach proposed in [18] and is thus not yet tractable. On the other hand, treating the η_i as constant by discretizing the sphere is tractable, since it can utilize the approach proposed in [18]. In addition, treating the η_i as an independent variable used to navigate target points on the sphere is also probably tractable (see Remark 4.4).

Consider a slight modification of derivative of the combined objective from Eq. (96),

$$\frac{\partial E_c(\vec{t}_{\text{root}}; t_{\text{cluster}}, \vec{\lambda}^P, \vec{\lambda}^E)}{\partial \vec{\lambda}^{P/E}} = \sum_i \left[t^R_{\text{root},i} - t_{\text{cluster}} + S(t^R_{\text{root},i}) \quad S(t^I_{\text{root},i}) \right] \frac{\partial t_{\text{root},i}}{\partial \vec{\lambda}^{P/E}} \tag{100}$$

where t_{cluster} is either constant or defined by Eq. (94). Here $S(t_{\text{cluster}})$ has been replaced by individual $S(t_{\text{root},i})$ both to give more fine grained control and so that Eq. (100) is still valid when t_{cluster} is constant. Weighting parameters can be added to the $S(t^R_{\text{root},i})$ in order to specify the relative importance of clustering the roots versus moving them around; similarly, weighting parameters can be added to the $S(t^I_{\text{root},i})$ in order to specify the relative importance of creating real roots versus clustering and moving them.

5. A simple 1D example

Consider a game played in $\mathbb{R}^1$ with a target C, a pursuer P, and an evader E arranged from left to right. The evader's goal is to get as close as possible to the target without colliding with the pursuer. For simplicity, let both the target and pursuer be stationary. The evader initially heads to the left towards the target before turning around to avoid the pursuer. See Fig. 2. The displacement s is

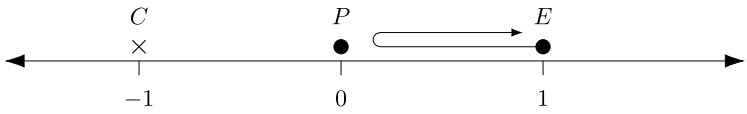

Fig. 2. The target, pursuer, and evader are arranged sequentially from left to right with the pursuer initially at the origin.

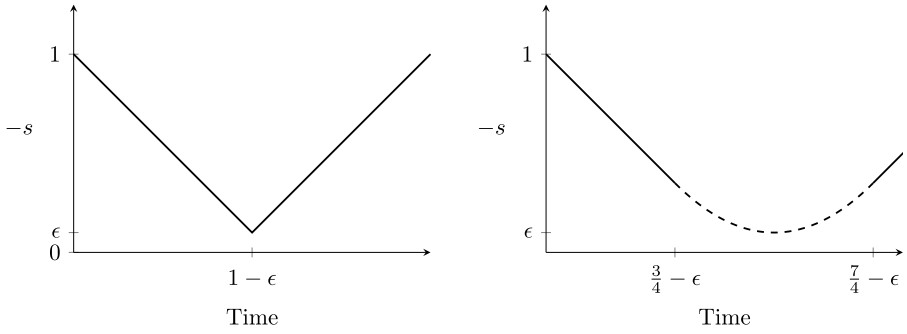

Fig. 3. Left: A plot of the negative displacement between the pursuer and evader, $|t-(1-\epsilon)|+\epsilon$, representing the behavior of an evader with a maximum speed of 1 and the ability to instantaneously modify their velocity. Right: The negative displacement for the piecewise smooth trajectory of an evader with a maximum speed of 1 (solid) and some finite turn around time (dashed) described by Eq. (101).

Table 1
Negative displacements $-s(t)$ for the evader when choosing $x^E_{\text{right}} = .26, .25$, and $.24$. See Fig. 4.

x^E_{left}	v^E_{left}	x^E_{right}	v^E_{right}	$-s(t)$	Roots	Extrema
.25	-1	.26	1	$-.02t^3+1.03t^2-t+.25$	$.49245 \pm .07i$, 50.52	.49250, 33.84
.25	-1	.25	1	$t^2-t+.25$	.5, .5, ∞	.5, ∞
.25	-1	.24	1	$.02t^3+.97t^2-t+.25$	.44, .58, -49.51	.51, -32.84

the negative of the evader's trajectory (from Eq. (13)), and the closest the evader can get to the target is to asymptotically reach the pursuer at $x=0$. Let the evader start at $x^E_{\text{left}}=1$ with $v^E_{\text{left}}=-1$ and maximum speed $v^E_{\max}=1$.

Consider a nearly optimal solution for the evader, where they reach a minimum distance of ϵ from the pursuer. If they can modify the direction of their velocity instantly (while maintaining their maximum speed $v^E_{\max}=1$), their (negative) displacement s is the absolute value of a linear function shifted vertically by ϵ. See Fig. 3, left. More realistically, if the evader requires finite time to turn around the absolute value function is interrupted. Consider a sample trajectory where the evader moves towards the pursuer with their maximum velocity of -1 until time t_n where the displacement is $1-t_n$. At this point an acceleration of 2 is applied until the evader reaches a velocity of 1, meaning that the displacement during this time is

$$-s(t) = (1-t_n) + (t-t_n)(-1) + \frac{(t-t_n)^2}{2}(2) \tag{101}$$

according to first order physics. In order for the parabola to have a global minimum of ϵ, i.e., at $(\frac{1}{2}+t_n, \epsilon)$, $t_n = \frac{3}{4}-\epsilon$. The evader will reach their maximum speed of 1 at $t=\frac{7}{4}-\epsilon$, meaning that Eq. (101) describes the displacement for $t \in [\frac{3}{4}-\epsilon, \frac{7}{4}-\epsilon]$. See Fig. 3, right.

Assume $\epsilon = 0$ for simplicity (the situation is similar when $\epsilon > 0$) and limit the discussion to the turn-around interval where $t \in [.75, 1.75]$. For simplicity, shift the time domain to $t \in [0,1] = [t_{\text{left}}, t_{\text{right}}]$ so that the evader's position at t_{left} is fixed at $x^E_{\text{left}} = 0.25$. Since only the interrupt of the absolute value is being considered, the evader has fixed $v^E_{\text{left}} = -1$ and $v^E_{\text{right}} = 1$ at t_{left} and t_{right} respectively. When modeling the evader's trajectory with a cubic spline, x^E_{right} is only remaining parameter and is thus considered the evader's only control. The optimal solution occurs when choosing $x^E_{\text{right}} = .25$, resulting in a repeated root with the pursuer. This is shown in Fig. 4 and Table 1 along with perturbations yielding a pair of either complex or real roots.

5.1. Greedy objectives

Let s_1 be the displacement from the evader to the pursuer and let s_2 be the displacement from the target to the evader,

$$s_1(t; \vec{\mathbf{p}}^s(\vec{\lambda}^E)) = -x^E(t; \vec{\mathbf{p}}^E(\vec{\lambda}^E)) \tag{102a}$$

$$s_2(t; \vec{\mathbf{p}}^s(\vec{\lambda}^E)) = x^E(t; \vec{\mathbf{p}}^E(\vec{\lambda}^E)) + 1 \tag{102b}$$

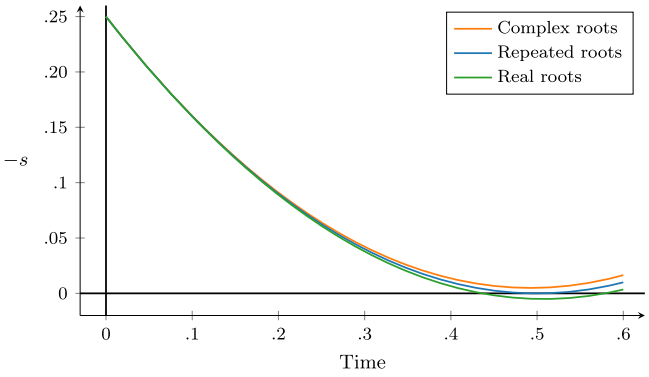

Fig. 4. Negative displacements $-s(t)$ from Table 1 representing the optimal solution (blue, a repeated root) and perturbations of the optimal solution (orange and green, for complex and real roots respectively) in the interrupt region. Note that the time axis is clipped to highlight the roots. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

noting that the evader has a pursuer role in s_2 when dealing with the target. An evader utilizing greedy objectives from Section 3.2 might consider maximizing

$$E_{\text{greedy}}(t_{\text{lag}};\vec{\lambda}^E) = \gamma^P E_\phi^P(t_{\text{lag}};\vec{\lambda}^E) - \gamma^C E_\phi^C(t_{\text{lag}};\vec{\lambda}^E) \tag{103a}$$

$$= \gamma^P \frac{1}{2}\left(x^E(t_{\text{lag}};\vec{\mathbf{p}}^E(\vec{\lambda}^E))\right)^2 - \gamma^C \frac{1}{2}\left(x^E(t_{\text{lag}};\vec{\mathbf{p}}^E(\vec{\lambda}^E)) + 1\right)^2 \tag{103b}$$

$$= \frac{1}{2}(\gamma^P - \gamma^C)(x^E(t_{\text{lag}};\vec{\mathbf{p}}^E(\vec{\lambda}^E)))^2 - \gamma^C x^E(t_{\text{lag}};\vec{\mathbf{p}}^E(\vec{\lambda}^E)) - \frac{1}{2}\gamma^C \tag{103c}$$

based on Eq. (42). When $\gamma^C = 0$, the optimal solution for the evader ignores the target and flees the pursuer with $x^E_{\text{right}} = \infty$. Increasing γ^C eventually causes the evader to stop targeting ∞ and to start targeting x^E_{right} towards $-\infty$. Once $\gamma^C = \gamma^P$, Eq. (103c) becomes linear with the optimal solution at $x^E_{\text{right}} = -\infty$. Increasing $\gamma^C > \gamma^P$ causes the maximum value of the quadratic in Eq. (103c) to occur at $x^E < 0$, meaning that the pursuer always wins the game. Unfortunately, larger penalties on the pursuer prevent the evader from approaching it, and smaller penalties on the pursuer allow the evader to cross over it. The only viable strategy is to keep γ^C large enough to allow the evader to approach the target until it is almost too late to avoid the pursuer; then, γ^P needs to be increased rapidly (while decreasing γ^C), hoping that there is enough time left to avoid the pursuer. Obviously, it would be better if the objective function captured the trajectories and the collision time; however, the relevant distance function formulations in Sections 3.3 and 3.4 are currently intractable.

Remark 5.1. *In multiple spatial dimensions the evader would aim to go around the pursuer instead of through it. At first this seems liked a marked improvement, but E_ϕ^P rules out an entire sphere of spatial locations around the pursuer; in fact, this sphere increases in size whenever the evader is successful in increasing its distance from the pursuer. The pursuer can use this sphere to cover the target, restricting the evader's strategies similar to the case in $\mathbb{R}^1$. Multiple pursuers can arrange their spheres to provide an even better covering of the target.*

5.2. Root-forecasting objectives

An evader utilizing root-forecasting objectives from Section 3.3 would use t^R_{min} (the real part of a root of ψ) instead of t_{lag} in Eq. (103). Notably, E_ϕ^P and E_ϕ^C utilize different t^R_{min} variables each depending on a different spline. This means that the two squared terms in Eq. (103b) each use a different x^E variable and thus cannot be merged into a single quadratic in x^E in Eq. (103c). Since the optimal solution is to set $x^E_{\text{right}} = .25$ in order to obtain a repeated root, the evader would set its target x^E_{right} position to its right aiming to turn around as soon as the game started being played with $t > 0$. Thus, the evader is adjusting γ^C and γ^P so that E_ϕ^C dominates when $t < 0$ and E_ϕ^P dominates when $t > 0$. This is the same undesirable strategy that the greedy formulation led to.

5.3. Root-controlling objectives

Since E_ϕ was shown to be problematic in Section 5.2, consider only using E_{pair} and $E_{t^I_{\text{min}}}$ from Eqs. (57) and (58). There are five roots of $\psi(t) = s(t)s'(t)$ since s is cubic, and Fig. 4 shows the three roots of interest (two of s and one of s'). Note that any two adjacent distinct real-valued roots of s are separated by a real-valued root of s'; moreover, any two adjacent distinct real-valued roots of ψ contains at least one (if not two) extrema. Thus, we consider merging a real root of s with a real root of s' via

$$E_{\text{triple}}(t_{\text{root}}, t_{\text{extrema}};\vec{\lambda}^E) = -|t^R_{\text{root}} - t^R_{\text{extrema}}| - |t^I_{\text{root}}| \tag{104}$$

stressing that that this doesn't actually happen; instead, Eq. (104) ends up merging two distinct real-valued roots while leaving the extrema intact (as can be seen in Fig. 4). In Section 5.4 where there are no issues associated with ψ mixing s and s', we will consider

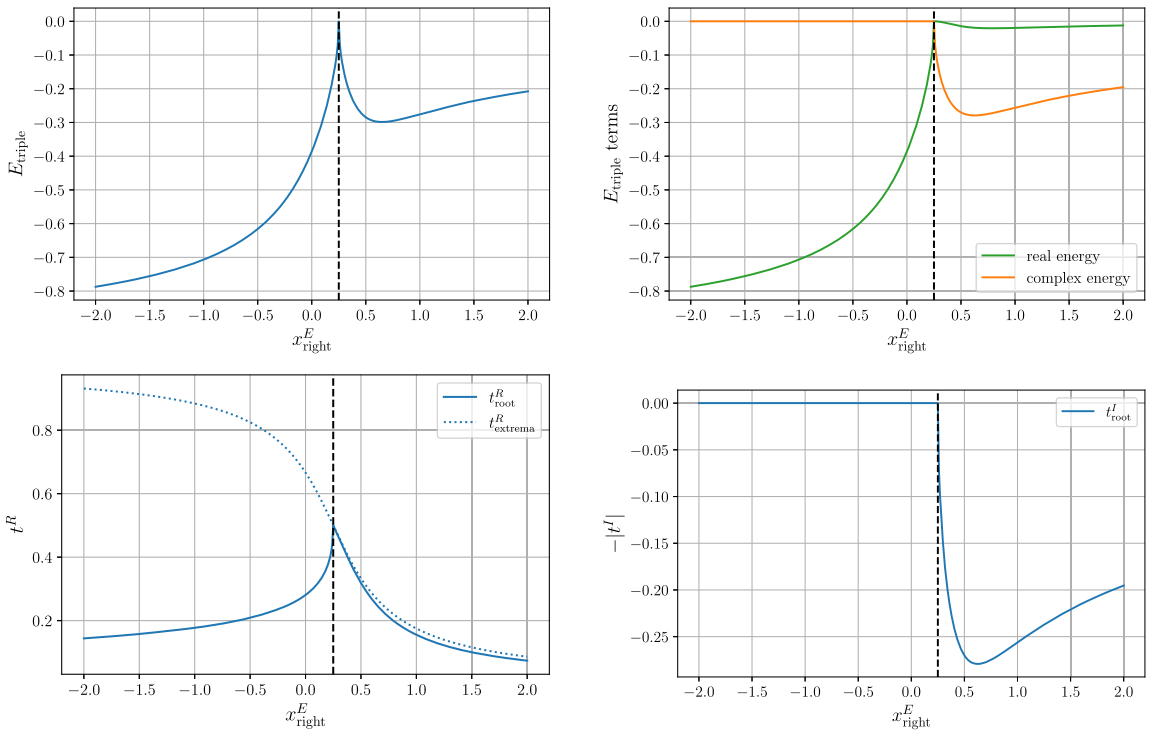

Fig. 5. The root-controlling E_{triple} has a global maximum at $x^E_{\text{right}} = .25$, corresponding to a triply repeated root (upper left). The first (upper right, green) and second (upper right, orange) terms in E_{triple} are calculated from the real (lower left) and the complex (lower right) parts of the roots, respectively. All plots share the same horizontal axis, the control x^E_{right}, and the dashed vertical line indicates the optimal $x^E_{\text{right}} = .25$.

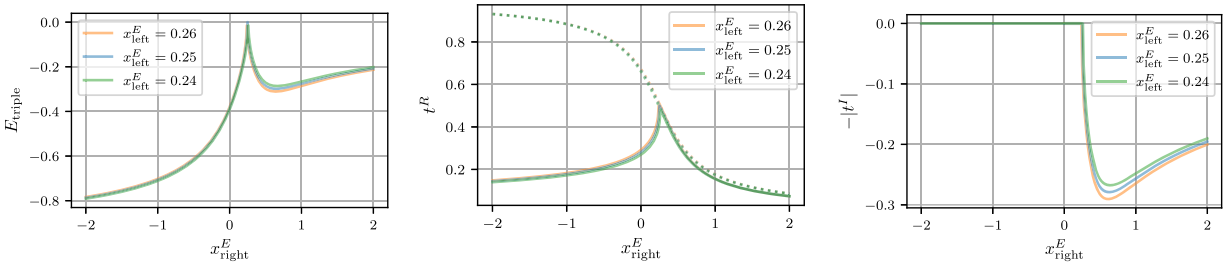

Fig. 6. Perturbing $x^E_{\text{left}} = .25$ (blue, as was shown in Fig. 5) to $x^E_{\text{left}} = .24$ (green) and $x^E_{\text{left}} = .26$ (orange) makes the expected perturbation in the energy E_{triple} (left), the real and complex parts of the roots (center and right respectively), and the corresponding extrema. All plots share the same horizontal axis, the control x^E_{right}. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the merging of two distinct real roots of s. Maximizing the first term in Eq. (104) both creates and maintains a triple root between two roots of s and one root of s', since a nonzero $t^I_{\text{root}} \neq 0$ implies $t^R_{\text{root}} \neq t^R_{\text{extrema}}$. However, as can be seen in Fig. 5, the separation of t^R_{root} and t^R_{extrema} is small for $t^I_{\text{root}} \neq 0$; thus, the second term in Eq. (104) has been added in order to create a more robust energy.

Fig. 6 shows that the formulation of the problem (i.e., energy, the roots, and the extrema) remains unchanged under small perturbations of x^E_{left}. Also note that it is straightforward to maintain a safe offset distance of ϵ merely by reformulating with the pursuer at ϵ instead of at 0. In hindsight, the root-forecasting strategy of Section 5.2 could have been made viable by ensuring that $t_{\min}$ is always chosen to be the extremum in $(0, 1)$; then, maximizing $-\phi(t_{\min})^2$ leads to the optimal strategy. The greedy strategy of Section 5.1 could have been made viable by choosing t_{lag} in the same fashion.

5.4. Root-finding objectives

An evader leveraging the root-finding formulation only considers the roots of the displacement with the pursuer s, ignoring the roots of s'. This leads to a slight modification of E_{triple} to

$$E_{\text{triple}}(t^+_{\text{root}}, t^-_{\text{root}}; \vec{\lambda}^E) = -|t^{R,+}_{\text{root}} - t^{R,-}_{\text{root}}| - |t^{I,+}_{\text{root}}| \tag{105}$$

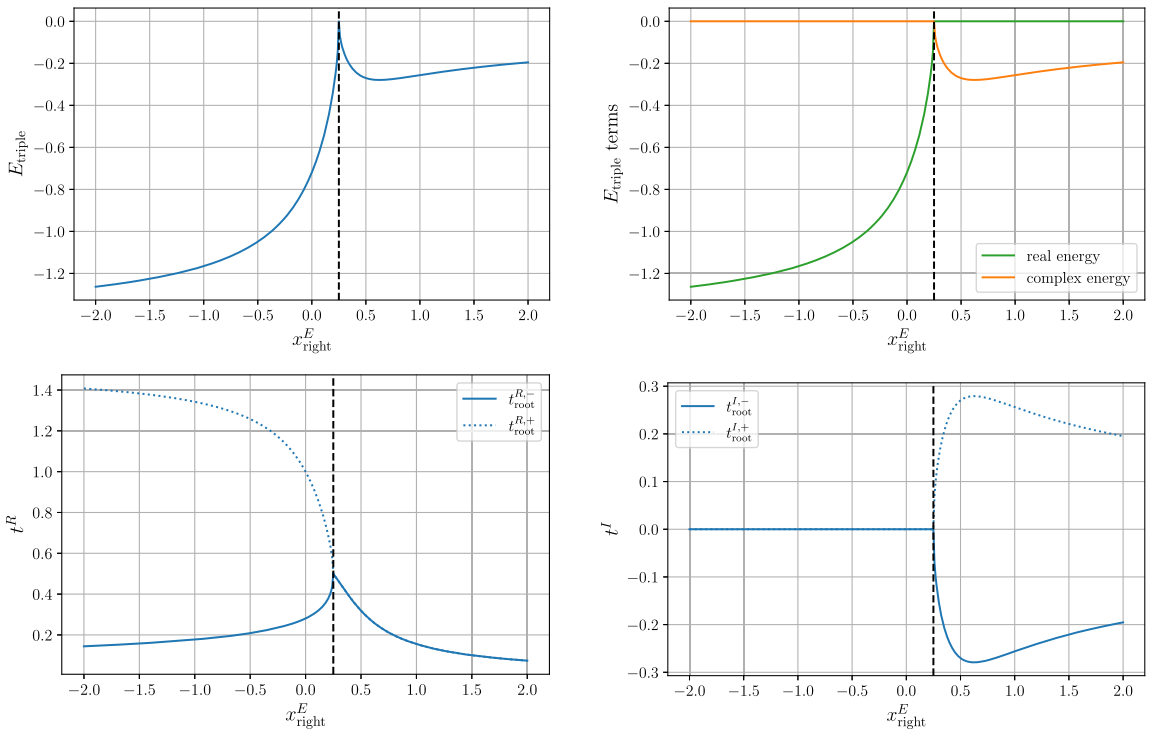

Fig. 7. The same as Fig. 5, except for the root-finding form of E_{triple}, Eq. (105). The global maximum of E_{triple} still occurs at $x^E_{\text{right}} = .25$ (upper left). The first (upper right, green) and second (upper right, orange) terms in E_{triple} behave similarly, following from the real (lower left) and complex (lower right) parts of the roots, respectively. All plots share the same horizontal axis, the control x^E_{right}, and the dashed vertical line indicates the optimal $x^E_{\text{right}} = .25$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where t^-_{root} and t^+_{root} are the first two roots encountered, respectively, for $t > 0$. See Fig. 7. Alternatively, consider maximizing

$$E_c(t_{\text{root}}; \vec{\lambda}^E) = t^R_{\text{root}} + \beta |t^I_{\text{root}}| \tag{106}$$

motivated by Eq. (91). The first term pushes the collision (or potential collision when $t^I_{\text{root}} \neq 0$) to happen at a later time and the second term aims to prevent the collision from happening. Note that t_{root} needs to be chosen to be the first root in $(0, 1)$. After real roots merge, the real part of the complex root starts moving back to the left (see Fig. 7, bottom left); thus, the first term in Eq. (106) aims for the triple root. The second term in Eq. (106) pushes the real part of the complex root to the left and thus competes with the first term; thus, β controls the size of the gap between the pursuer and evader. See Fig. 8.

Remark 5.2. *It is worth noting that dropping consideration of the target did not lead to the evader immediately heading off to $+\infty$ as quickly as possible. This is important in the context of inertia of game play. It makes little sense for an agent to panic and take the most extreme action possible to achieve a goal that is achievable with minimal effort. Moreover, using minimal effort to achieve a particular goal allows an agent to continue progressing towards other goals with minimal disruption. In the example under consideration, this translates into the evader continuing to proceed along their path to the left up until the point at which evasive action is actually necessary.*

6. Agnostic pursuer example

Consider the extension of the game outlined in Fig. 2 to $\mathbb{R}^2$ with $\vec{x}^E_{\text{left}} = (1, 0)$, $\vec{v}^E_{\text{left}} = (-1, 0)$, a stationary pursuer at $(0, y^P)$, a target defined via the line $x = -1$, and $t \in [0, 1] = [t_{\text{left}}, t_{\text{right}}]$. Fix the x-component of $\vec{x}^E_{\text{right}}$ to be -1, and choose $\vec{v}^E_{\text{right}} = (-1, 0)$ so that the evader crosses the goal line with a velocity perpendicular to the goal; then, the y-component of $\vec{x}^E_{\text{right}}$, denoted y^E_{right}, is the only control. This leads to

$$x^E(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} = 2t^3 - 3t^2 - t + 1 \tag{107a}$$

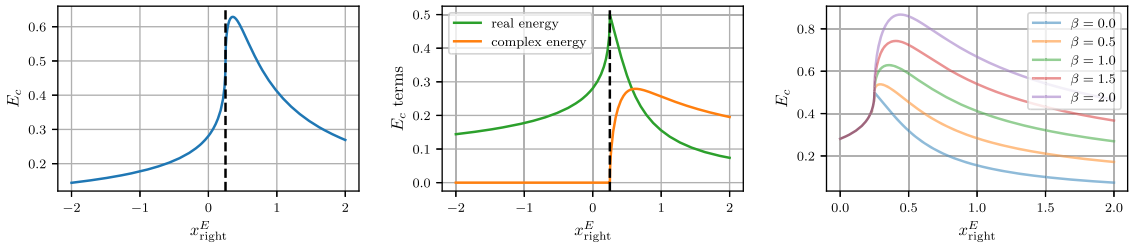

Fig. 8. The global maximum of E_c occurs when an $x^E_{\text{right}} > .25$ maintains a finite-sized gap between the pursuer and the evader (left, for $\beta = 1$). The contribution of the real part of the root (middle, green) is fixed, and decreasing the contribution of the imaginary part (middle, orange) by decreasing β moves the maximum closer to $x^E_{\text{right}} = .25$ (right, colors blue to purple for β from 0 to 2). Note that all plots share the same horizontal axis, the control x^E_{right}, and the dashed vertical line indicates the optimal $x^E_{\text{right}} = .25$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$y^E(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ y^E_{\text{right}} \\ 0 \end{bmatrix} = -2y^E_{\text{right}} t^2 \left(t - \frac{3}{2} \right) \tag{107b}$$

for the evader's trajectory (following from Eq. (11) and (12)). Given the tractability and other (e.g., see Remark 5.1) issues with the distance function formulations, we limit discussion to root-finding formulations going forward.

6.1. Stationary pursuer

With $(0, y^P)$ as the position of the stationary pursuer, the displacement is

$$s_x^{P-E}(t) = -x^E(t) = -2t^3 + 3t^2 + t - 1 \tag{108a}$$

$$s_y^{P-E}(t) = y^P - y^E(t) = y^P + 2y^E_{\text{right}} t^2 \left(t - \tfrac{3}{2} \right) \tag{108b}$$

where s_x^{P-E} has roots $\frac{1}{2}$, $\frac{1}{2} \pm \sqrt{\frac{5}{4}}$, implying that a collision happens when s_y^{P-E} has a root at $\frac{1}{2}$. Setting $s_y^{P-E}(\frac{1}{2}) = 0$ gives $y^E_{\text{right}} = 2y^P$ as the only invalid strategy for the evader. Substituting $y^E_{\text{right}} = 2y^P$ into Eq. (108b) gives

$$s_y^{P-E}(t) = 2y^P (t - \tfrac{1}{2})(2t^2 - 2t - 1) \tag{109}$$

which has roots $\frac{1}{2}$, $\frac{1 \pm \sqrt{3}}{2}$, unless $y^P = 0$ in which case all $t \in \mathbb{R}$ are roots.

The $y^P = 0$ case is a removable singularity in the family of splines. This can be seen by noting that s_y^{P-E}/y^P is the same cubic for all $y^P \neq 0$. Thus, the roots of s_y^{P-E} are ill-conditioned as $y^P \to 0$ and not degenerate. This can also be seen by rotating the coordinate system by an angle θ, from (x, y) to $(\hat{x}, \hat{y})$, giving

$$\begin{bmatrix} s_{\hat{x}}^{P-E}(t) \\ s_{\hat{y}}^{P-E}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} s_x^{P-E}(t) \\ s_y^{P-E}(t) \end{bmatrix} \tag{110}$$

where Eq. (109) causes the second column to vanish when $y^P = 0$ leaving only s_x^{P-E} from Eq. (108a) multiplying the first column; then, both the $\hat{x}$ and $\hat{y}$ coordinate splines have the same three roots as long as the rotation is not a multiple of $\frac{\pi}{2}$.

Changing the x location of the pursuer modifies Eq. (108a) causing the relevant root of s_x^{P-E} to deviate from $t_{\text{root}} = \frac{1}{2}$. After solving $s_y^{P-E}(t_{\text{root}}) = 0$ in order to obtain y^E_{right},

$$s_y^{P-E}(t) = \frac{-y^P}{t_{\text{root}}^2 (t_{\text{root}} - \frac{3}{2})} (t - t_{\text{root}}) \left(t^2 + t(t_{\text{root}} - \tfrac{3}{2}) + t_{\text{root}}^2 - \tfrac{3}{2} t_{\text{root}} \right) \tag{111}$$

replaces Eq. (109). Once again, one can divide s_y^{P-E} by y^P.

6.2. Stationary pursuer with spatial extent

Let the pursuer have spatial extent in one dimension, represented by a line segment $[-y_p, y_p]$ along $x = 0$. The x-displacement s_x^{P-E} still has a root at $\frac{1}{2}$. If $y^E_{\text{left}} \in [-y_p, y_p]$, then the first non-repeated root of either $y_p - y^E(t)$ or $-y_p - y^E(t)$ indicates the evader leaving the pursuer's y-extent; otherwise, it indicates the evader entering the pursuer's y-extent. Thus, the evader would aim to have 1, 3, or 5 real roots of $y_p - y^E(t)$ and $-y_p - y^E(t)$ combined in $t \in (0, \frac{1}{2})$. Having 0, 2, or 4 real roots leads to failure, and it is not possible to have 6 or more total real roots between the two cubics.

Suppose the evader aimed for exactly one real root in $t \in (0, \frac{1}{2})$ and preferred to have that root with $y_p - y^E(t)$, shown in Eq. (108b). Since $t^2(t - \frac{3}{2})$ is negative in $(0, \frac{1}{2})$, $y^E_{\text{right}} > 0$ is required in order for a root of $y_p - y^E(t)$ to exist in $(0, \frac{1}{2})$. With $y^E_{\text{right}} > 0$ and $y_p = 0$, $y_p - y^E(t)$ has roots at $t = 0, 0, \frac{3}{2}$ and a local minimum at $t = 1$. Increasing y_p shifts $y_p - y^E(t)$ upwards, causing one of the repeated roots to move to the right and eventually merge with the largest root at $t = 1$. Increasing y^E_{right} slows the rightward movement of the repeated root caused by increasing y_p; thus, since $y_p - y^E(\frac{1}{2}) = 0$ gives $y^E_{\text{right}} = 2y_p$, $y^E_{\text{right}} \in (2y_p, \infty)$ guarantees that the rightward moving root remains in $t \in (0, \frac{1}{2})$ as desired. Note that aiming to instead have the root be with $-y_p - y^E(t)$ would have given $y^E_{\text{right}} \in (-\infty, -2y_p)$; thus, the $[-2y_p, 2y_p]$ subset of the goal line represents invalid strategies.

Modifying the situation to add spatial extent in the x-direction by creating a box $[-x_p, x_p] \times [-y_p, y_p]$ around the pursuer means that the evader needs the rightward moving root to be earlier then $t = \frac{1}{2}$. The time interval can be found by adding x_p to the right-hand side of Eq. (108a). Since the rightward moving root is the only one that can occur in $t \in (0, 1)$, the evader can ignore potential intersections with the box when approaching $y = y_p$ from $y > y_p$ (or similarly from below).

6.3. Constant velocity pursuer

Consider a pursuer starting at the origin and moving with constant velocity $\vec{v}^P_{\text{left}} = \vec{v}^P_{\text{right}} = (0, -2)$ so that $\vec{x}^P_{\text{right}} = (0, -2)$; then,

$$s_y^{P-E}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -2 \\ -2 - y^E_{\text{right}} \\ -2 \end{bmatrix} = -2t + 2y^E_{\text{right}} t^2 \left(t - \frac{3}{2} \right) \tag{112}$$

replaces Eq. (108b). Setting $s_y^{P-E}(\frac{1}{2}) = 0$ gives $y^E_{\text{right}} = -2$ as the only invalid strategy for the evader. Substituting $y^E_{\text{right}} = -2$ into Eq. (112) gives $s_y^{P-E}(t) = -4t(t-1)(t-\frac{1}{2})$.

Suppose that this moving pursuer has spatial extent $[-y_p, y_p]$ in the y-direction so that the roots of $y_p - 2t - y^E(t)$ and $-y_p - 2t - y^E(t)$ are relevant. The x-displacement $s_x^{P-E}(t)$ still has a root at $\frac{1}{2}$, and if the evader aims for one real root of $y_p - 2t - y^E(t)$ in $t \in (0, \frac{1}{2})$ then

$$s_y^{P-E}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -2 & 1 \\ -3 & -2 & 3 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_p \\ -2 \\ y_p - 2 - y^E_{\text{right}} \\ -2 \end{bmatrix} = y_p - 2t + 2y^E_{\text{right}} t^2 (t - \tfrac{3}{2}) \tag{113}$$

replaces Eq. (112). Consider

$$s_y^{P-E}(t) = 2y^E_{\text{right}} t \left(t - \frac{3}{4} + \sqrt{\frac{9}{16} + \frac{1}{y^E_{\text{right}}}} \right) \left(t - \frac{3}{4} - \sqrt{\frac{9}{16} + \frac{1}{y^E_{\text{right}}}} \right) + y_p \tag{114a}$$

$$\frac{ds_y^{P-E}(t)}{dt} = 6y^E_{\text{right}} \left(t - \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{1}{3y^E_{\text{right}}}} \right) \left(t - \frac{1}{2} - \sqrt{\frac{1}{4} + \frac{1}{3y^E_{\text{right}}}} \right) \tag{114b}$$

in order to understand the perturbation of Eq. (108b) by the $-2t$ term to obtain Eq. (113).

When $y^E_{\text{right}} > 0$, the $-2t$ term perturbs one of the repeated roots at $t = 0$ to the left and the root at $t = \frac{3}{2}$ to the right. Increasing y_p causes the root at $t = 0$ to move to the right and eventually merge with the largest root at some $t > 1$. When $y^E_{\text{right}} = 0$, increasing y_p still moves the root at $t = 0$ to the right; however, there is no other root to merge with. When $-\frac{4}{3} \le y^E_{\text{right}} < 0$, the monotone decreasing s_y^{P-E} maintains the same behavior as in the $y^E_{\text{right}} = 0$ case with the root at $t = 0$ moving to the right. Setting $s_y^{P-E}(\frac{1}{2}) = 0$ gives $y^E_{\text{right}} = 2y_p - 2$. Since a larger y^E_{right} makes s_y^{P-E} more negative in $t \in (0, \frac{3}{2})$ and s_y^{P-E} is decreasing in $(0, 1)$, a larger y^E_{right} moves the root in $(0, 1)$ to the left; therefore, valid strategies are represented by $y^E_{\text{right}} > 2y_p - 2$.

When $y^E_{\text{right}} < -\frac{4}{3}$, the critical points $t_{c_1} = t_{c_2} = \frac{1}{2}$ that appear when $y^E_{\text{right}} = -\frac{4}{3}$ separate with $t_{c_1} \to 0$ and $t_{c_2} \to 1$ as $y^E_{\text{right}} \to -\infty$. In this case, setting $s_y^{P-E}(\frac{1}{2}) = 0$ to obtain $y^E_{\text{right}} = 2y_p - 2$ is the condition on the second (not the first) root in $(0, 1)$. Since a larger y^E_{right} makes s_y^{P-E} more negative in $t \in (0, \frac{3}{2})$ and s_y^{P-E} is increasing around $t = \frac{1}{2}$, a larger y^E_{right} moves the second root to the right of $t = \frac{1}{2}$; therefore, valid strategies are still represented by $y^E_{\text{right}} > 2y_p - 2$.

Adding spatial extent in the x-direction creates a box $[-x_p, x_p] \times [-y_p, y_p]$ around the pursuer, meaning that the evader needs to act earlier than $t = \frac{1}{2}$. This earlier time $t_1 < \frac{1}{2}$ is the unique root in $t \in (0, \frac{1}{2})$ found after adding x_p to the right-hand side of Eq. (108a). In addition, the evader needs to avoid re-intersecting $y = y_p$ from $y > y_p$ before some time $t_2 > \frac{1}{2}$, where t_2 is the unique root in $t \in (\frac{1}{2}, 1)$ found after subtracting x_p from the right-hand side of Eq. (108a).

When $y^E_{\text{right}} \ge -\frac{4}{3}$, solving for $s_y^{P-E}(t_1) = 0$ gives $y^E_{\text{right}} = \frac{-2t_1 + y_p}{t_1^2(3 - 2t_1)}$; therefore, valid strategies are given by $y^E_{\text{right}} > \frac{-2t_1 + y_p}{t_1^2(3-2t_1)}$. For $y^E_{\text{right}} < -\frac{4}{3}$, the condition on the first root is no longer automatically satisfied by the condition on the second root when $t_1 < t_{c_1}$. From

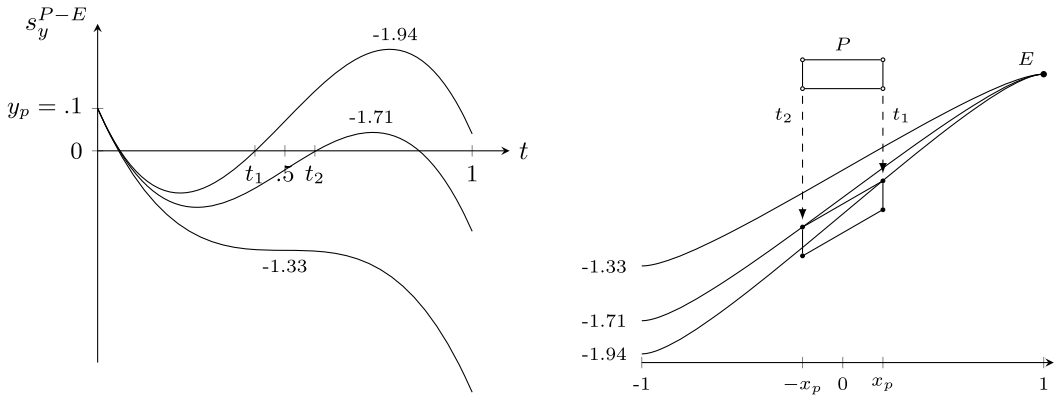

Fig. 9. Left: Plot of $s_y^{P-E}(t)$ with $y_p = .1$ for various y^E_{right} (values of y^E_{right} are indicated along their respective curve). Increasing y^E_{right} ensures that a second root does not occur before time $t = t_2$. Right: Visualization of the evader's trajectories corresponding to the figure to the left. The parallelogram represents the portion of the time evolving boundary of the pursuer that the evader's trajectories need to avoid. The $y^E_{\text{right}} = -1.94$ trajectory collides with the top right corner of the pursuer, whereas the $y^E_{\text{right}} = -1.71$ just misses the top left corner of the pursuer.

Eq. (114b), $t_1 < t_{c_1}$ implies $\frac{-1}{3t_1(1-t_1)} < y^E_{\text{right}}$. This means that $y^E_{\text{right}} > \frac{-2t_1+y_p}{t_1^2(3-2t_1)}$ extends from $y^E_{\text{right}} \geq -\frac{4}{3}$ to $y^E_{\text{right}} > \frac{-1}{3t_1(1-t_1)}$ when $\frac{-1}{3t_1(1-t_1)} < -\frac{4}{3}$. When $t_2 \leq t_{c_2}$, there is a condition on the second root; otherwise, when $t_2 > t_{c_2}$ the condition is on t_{c_2} instead of t_2. That is, the condition on the second root is $y^E_{\text{right}} > \frac{-2\min(t_{c_2},t_2)+y_p}{\min(t_{c_2},t_2)^2(3-2\min(t_{c_2},t_2))}$. From Eq. (114b), $t_{c_2} \leq t_2$ is relevant when $\frac{-1}{3t_2(1-t_2)} \leq y^E_{\text{right}} < -\frac{4}{3}$ and $t_2 < t_{c_2}$ is relevant when $y^E_{\text{right}} < \frac{-1}{3t_2(1-t_2)}$. Note that the condition as written is slightly circular when the $\min(t_{c_2}, t_2)$ is t_{c_2}, since $t_{c_2} = \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{1}{3y^E_{\text{right}}}}$ depends on y^E_{right}; however, the inequality can be manipulated to obtain a non-circular condition on y^E_{right}. See Fig. 9.

Adding a velocity u_p in the x-direction would modify the right-hand side of Eq. (108a) to include $u_p t$ in addition to $\pm x_p$. Although this causes both t_1 and t_2 to occur sooner/later for positive/negative values of u_p respectively, the analysis is similar.

Remark 6.1. *The reasoning and calculations for reaching the goal line by going below the pursuer are similar in spirit to what was discussed above for going above the pursuer.*

7. Predatory pursuer

Consider a pursuer with fixed speed v_p and velocity that always aims directly at the evader (i.e., pure pursuit [64]); then,

$$\vec{x}^P(t) = \vec{x}^P(t_n) - v_p \int_{t_n}^{t} \frac{\vec{x}^P(\tau) - \vec{x}^E(\tau)}{\left\|\vec{x}^P(\tau) - \vec{x}^E(\tau)\right\|_2} d\tau \tag{115}$$

gives the pursuer's position for $t > t_n$. Uncertainty over the evader's trajectory hinders the pursuer's ability to always head in the optimal direction; instead, an estimate of the evader's position $\vec{x}^E(t_n + \Delta t)$ at time $t_n + \Delta t$ would be used; then, assuming that the pursuer traveled in a straight line from time t_n to time $t_n + \Delta t$, Eq. (115) can be replaced by

$$\vec{x}^P(t) = \vec{x}^P(t_n) - (t - t_n) v_p \frac{\vec{x}^P(t_n) - \vec{x}^E(t_n + \Delta t)}{\left\|\vec{x}^P(t_n) - \vec{x}^E(t_n + \Delta t)\right\|_2} \tag{116}$$

for $t \in (t_n, t_n + \Delta t)$.

8. Constraining velocities

Before proceeding with a predatory pursuer example, the velocity of the evader needs to be constrained in order to make escaping the pursuer non-trivial. The more general framing of this problem consists of considering a spline on $[0, t_{\text{right}}]$ where Eq. (8) gives

$$\vec{q} = \frac{1}{t_{\text{right}}^3}(-2\Delta\vec{x} + t_{\text{right}}(\vec{v}_{\text{left}} + \vec{v}_{\text{right}})) \tag{117a}$$

$$\vec{a} = \frac{1}{t_{\text{right}}^2}(3\Delta\vec{x} - t_{\text{right}}(2\vec{v}_{\text{left}} + \vec{v}_{\text{right}})) \tag{117b}$$

$$\vec{b} = \vec{v}_{\text{left}} \tag{117c}$$

with $\Delta\vec{x} = \vec{x}_{\text{right}} - \vec{x}_{\text{left}}$ and setting out to minimize t_{right} while bounding the magnitude of the velocity. For discussion and analysis of other possible constraints, such as a constraint on arc length, see [65].

Assuming that $\vec{v}_{\text{left}}$ and $\vec{v}_{\text{right}}$ are appropriately bounded, only relative extrema of the speed in $(0, t_{\text{right}})$ need to be considered. The square of the speed and its derivatives

$$\left\|\vec{s}'(t)\right\|_2^2 = \vec{s}'(t)\cdot\vec{s}'(t) = (3\vec{q}t^2 + 2\vec{a}t + \vec{b})\cdot(3\vec{q}t^2 + 2\vec{a}t + \vec{b}) \tag{118a}$$

$$\frac{d}{dt}\frac{\left\|\vec{s}'(t)\right\|_2^2}{2} = \vec{s}''(t)\cdot\vec{s}'(t) = (6\vec{q}t + 2\vec{a})\cdot(3\vec{q}t^2 + 2\vec{a}t + \vec{b}) \tag{118b}$$

$$\frac{d^2}{dt^2}\frac{\left\|\vec{s}'(t)\right\|_2^2}{2} = \vec{s}'''(t)\cdot\vec{s}'(t) + \vec{s}''(t)\cdot\vec{s}''(t) = 6\vec{q}\cdot(3\vec{q}t^2 + 2\vec{a}t + \vec{b}) + (6\vec{q}t + 2\vec{a})\cdot(6\vec{q}t + 2\vec{a}) \tag{118c}$$

can be used to determine the minimum t_{right} that bounds the speed at internal relative extrema.

When $\vec{q} = 0$, Eq. (118a) is at most quadratic and $\vec{a}\cdot\vec{a} \geq 0$ implies that there are no relative maxima in the interval. When $\vec{q} \neq \vec{0}$, Eq. (118a) is quartic with a positive leading coefficient meaning that a relative maximum only occurs when there are three distinct extrema; in particular, the middle extrema would be the relative maximum. Thus, a relative maximum exists if and only if the cubic in Eq. (118b) has three distinct real roots. Note that $\vec{s}' = 0$ can be ruled out, since that would be a relative minimum, not a relative maximum, of Eq. (118a); thus, the root of interest has either $\vec{s}'' = 0$ or $\vec{s}''$ orthogonal to $\vec{s}'$. In addition, $\vec{s}''\cdot\vec{s}'$ must be strictly decreasing near the root, implying that the quadratic in Eq. (118c) must be strictly negative at the root, i.e., $\vec{s}'''\cdot\vec{s}' < -\vec{s}''\cdot\vec{s}'' < 0$ at the root. This can only happen if the concave up quadratic is negative at its minimum. Since the minimum occurs at $t = \frac{-\vec{q}\cdot\vec{a}}{3\vec{q}\cdot\vec{q}}$, $\vec{s}'''\cdot\vec{s}' < -\vec{s}''\cdot\vec{s}'' < 0$ is also required here. Notably, both quadratics in Eq. (118c) have the same minimum, meaning that $\left\|\vec{s}''\right\|_2$ is minimized here as well; interestingly, setting $\vec{s}'' = 0$ and minimizing the residual via the normal equations leads to $\vec{s}''\cdot\vec{q} = 0$ at the minimum of the quadratic.

At this point, t_{right} can be included as an additional spline parameter along with $\vec{x}_{\text{left}}$, $\vec{v}_{\text{left}}$, $\vec{x}_{\text{right}}$, and $\vec{v}_{\text{right}}$. In the context of our pursuit-evasion discussions, the evader would minimize t_{right} subject to some (soft) constraints on the maximum speed in $(0, t_{\text{right}})$. See Figs. 10 and 11. As discussed above, this means bounding the speed at the middle root (when it exists) of the three-distinct-root cubic in Eq. (118b). In order to increase or decrease the speed at the local maximum, the straightforward derivatives of the quartic in Eq. (118a) with respect to both its parameters and its argument are required. Since the argument is a root of the cubic in Eq. (118b), the chain rule dictates derivatives of the roots of a cubic with respect to its parameters. Although this could be accomplished via [18], the chain rule is not actually necessary at the middle root of the cubic since the derivative of the quartic with respect to its argument is zero there. Finally, note that the derivatives of $\vec{q}$, $\vec{a}$, $\vec{b}$, and $\vec{c}$ with respect to the spline parameters can be computed from Eq. (117). This modifies Eq. (20) to include $\frac{\partial\vec{p}_i^s}{\partial t_{\text{right}}}\frac{\partial t_{\text{right}}}{\partial\vec{\lambda}}$ terms.

Remark 8.1. *The notion of t_{right} ties into the root-finding formulation. When t_{root} is a preferred collision, using t_{root} in place of t_{right} and aiming to shorten the time of collision can lead to a nonphysical speed extrema in $(0, t_{root})$ introducing constraints on the ability to reduce t_{root}.*

Remark 8.2. *It could be worth considering whether analysis of Eqs. (118a)–(118c) might lead to a simpler formulation. Define*

$$\vec{a}_{left} = \vec{s}''(0) = 2\vec{a} = \frac{2}{t_{right}^2}(3\Delta\vec{x} - t_{right}(2\vec{v}_{left} + \vec{v}_{right})) \tag{119a}$$

$$\vec{a}_{right} = \vec{s}''(t_{right}) = 6\vec{q}t_{right} + 2\vec{a} = \frac{2}{t_{right}^2}\left(-3\Delta\vec{x} + t_{right}(\vec{v}_{left} + 2\vec{v}_{right})\right) \tag{119b}$$

which satisfy

$$\frac{\vec{a}_{left} + \vec{a}_{right}}{2} = \frac{\vec{v}_{right} - \vec{v}_{left}}{t_{right}} \tag{120}$$

illustrating their simple dependence on the original parameters; then, assuming that the middle root of the cubic is not appearing or disappearing but simply moving in time,

$$\vec{a}_{left}\cdot\vec{v}_{left} = \left(3\Delta\vec{x} - (2\vec{v}_{left} + \vec{v}_{right})t_{right}\right)\cdot\vec{v}_{left} = 0 \tag{121a}$$

$$\vec{a}_{right}\cdot\vec{v}_{right} = \left(-3\Delta\vec{x} + (\vec{v}_{left} + 2\vec{v}_{right})t_{right}\right)\cdot\vec{v}_{right} = 0 \tag{121b}$$

with solutions

$$t_L = \frac{3\Delta\vec{x}\cdot\vec{v}_{left}}{(2\vec{v}_{left} + \vec{v}_{right})\cdot\vec{v}_{left}} \quad \text{and} \quad t_R = \frac{3\Delta\vec{x}\cdot\vec{v}_{right}}{(\vec{v}_{left} + 2\vec{v}_{right})\cdot\vec{v}_{right}} \tag{122}$$

for t_{right} in Eqs. (121a) and (121b) respectively can be used to determine when the constraints may need to be turned on and off. We leave further analysis to future work (except for the 1D case addressed in Section 8.1, and some analysis on existence in the 3D case in Section 8.2) since bounds on the splines are only a proxy for bounds on the physics anyways.

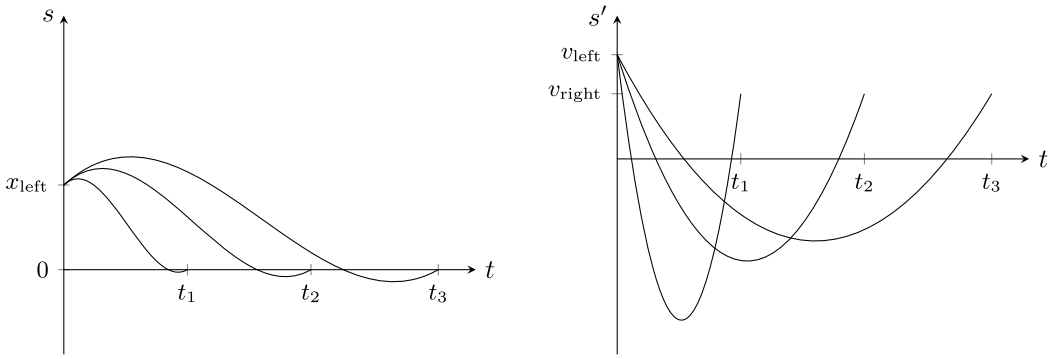

Fig. 10. One-dimensional splines (left) with corresponding velocities (right) for various t_{right} with all other parameters identical.

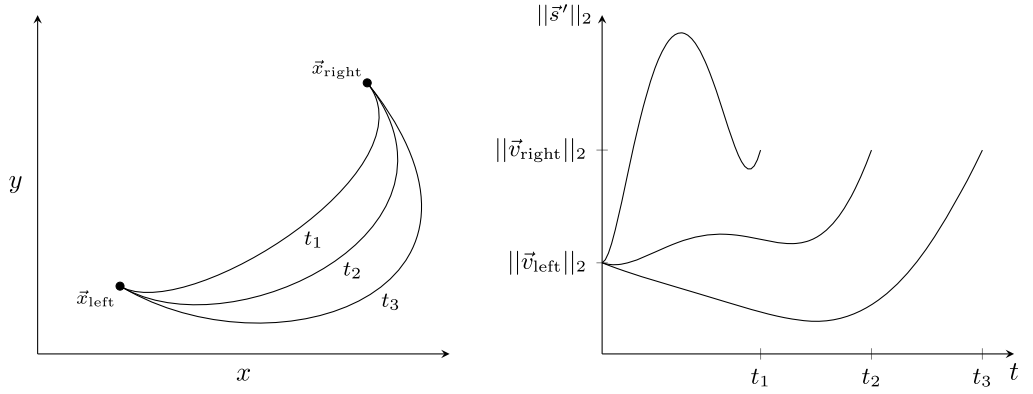

Fig. 11. Two-dimensional splines (left) with corresponding speeds (right) for various t_{right} with all other parameters identical. Note how decreasing t_{right} creates a local maximum in $||s'||_2$ in the time interval of interest.

8.1. One-dimensional case

In one spatial dimension, s' and s'' cannot be orthogonal meaning that relative extrema only occur when $s' = 0$ or $s'' = 0$. When the roots of s' are repeated, they are coincident with the root of s''; moreover, the root of s'' is a relative maximum when the roots of s' are real-valued and distinct. The root of s'' can be written as

$$t_{\text{inflect}} = -\frac{a}{3q} = \frac{t_{\text{right}} a_{\text{left}}}{a_{\text{left}} - a_{\text{right}}} \tag{123}$$

with

$$s'(t_{\text{inflect}}) = -\frac{a^2}{3q} + b = \frac{t_{\text{right}} a_{\text{left}}^2}{2(a_{\text{left}} - a_{\text{right}})} + v_{\text{left}} = \frac{t_{\text{right}} a_{\text{right}}^2}{2(a_{\text{left}} - a_{\text{right}})} + v_{\text{right}} \tag{124}$$

where the last equality is derived with the help of Eq. (120).

Since s'' is continuous, $t_{\text{inflect}} \in (0, t_{\text{right}})$ if and only if a_{left} and a_{right} have opposite signs. When t_{right} is slightly positive, a_{left} and a_{right} differ in sign via the $\pm 3\Delta x$ terms; thus, $t_{\text{inflect}} \in (0, t_{\text{right}})$. Increasing t_{right} may cause either or both of a_{left} and a_{right} to change signs. The sign changes at

$$t_L = \frac{3\Delta x}{2v_{\text{left}} + v_{\text{right}}} \quad \text{and} \quad t_R = \frac{3\Delta x}{v_{\text{left}} + 2v_{\text{right}}} \tag{125}$$

are relevant when $t_L > 0$ and $t_R > 0$ respectively. The smallest positive value between t_L and t_R provides an upper bound on the values of t_{right} that have have $t_{\text{inflect}} \in (0, t_{\text{right}})$. Choosing t_{right} to be equal to this upper bound guarantees that the velocity extrema occurs at either 0 or t_{right}. In addition, it may be possible to shrink t_{right} even further based on the behavior of $s'(t_{\text{inflect}})$.

Using t_L and t_R,

$$v_{\text{left}} = \Delta x\left(\frac{2}{t_L} - \frac{1}{t_R}\right), \qquad v_{\text{right}} = \Delta x\left(\frac{2}{t_R} - \frac{1}{t_L}\right) \tag{126a}$$

$$a_{\text{left}} = \frac{6\Delta x}{t_{\text{right}}^2}\left(1 - \frac{t_{\text{right}}}{t_L}\right), \quad a_{\text{right}} = \frac{-6\Delta x}{t_{\text{right}}^2}\left(1 - \frac{t_{\text{right}}}{t_R}\right) \tag{126b}$$

and t_{inflect} can be written as

$$t_{\text{inflect}} = \left(\frac{1 - \frac{t_{\text{right}}}{t_L}}{1 - \frac{t_{\text{right}}}{t_L} + 1 - \frac{t_{\text{right}}}{t_R}} \right) t_{\text{right}} \tag{127}$$

which leads to three variations of $s'(t_{\text{inflect}})$,

$$s'(t_{\text{inflect}}) = v_{\text{left}} + \frac{3\Delta x}{t_{\text{right}}} \frac{(1 - \frac{t_{\text{right}}}{t_L})^2}{1 - \frac{t_{\text{right}}}{t_L} + 1 - \frac{t_{\text{right}}}{t_R}} \tag{128a}$$

$$s'(t_{\text{inflect}}) = v_{\text{right}} + \frac{3\Delta x}{t_{\text{right}}} \frac{(1 - \frac{t_{\text{right}}}{t_R})^2}{1 - \frac{t_{\text{right}}}{t_L} + 1 - \frac{t_{\text{right}}}{t_R}} \tag{128b}$$

$$s'(t_{\text{inflect}}) = \frac{\Delta x}{t_{\text{right}}} \frac{(1 - \frac{t_{\text{right}}}{t_L})^2 + (1 - \frac{t_{\text{right}}}{t_R})^2 + 1 - \frac{t_{\text{right}}}{t_L} \frac{t_{\text{right}}}{t_R}}{1 - \frac{t_{\text{right}}}{t_L} + 1 - \frac{t_{\text{right}}}{t_R}} \tag{128c}$$

with derivative

$$\frac{\partial s'(t_{\text{inflect}})}{\partial t_{\text{right}}} = \frac{-6\Delta x}{t_{\text{right}}^2} \frac{(1 - \frac{t_{\text{right}}}{t_L})(1 - \frac{t_{\text{right}}}{t_R})}{\left(1 - \frac{t_{\text{right}}}{t_L} + 1 - \frac{t_{\text{right}}}{t_R}\right)^2} \tag{129}$$

which all facilitate conditions on $s'(t_{\text{inflect}})$. All $t_{\text{right}} > 0$ that are smaller than the smallest positive value between t_L and t_R (i.e., all t_{right} under consideration) lead to a positive value for the second fraction on the righthand side of Eq. (129); thus, $s'(t_{\text{inflect}})$ is monotone in t_{right} for all t_{right} under consideration. In addition, the second fraction in Eq. (128c) is positive for all t_{right} under consideration; thus, $s'(t_{\text{inflect}})$ is the same sign as Δx for all t_{right} under consideration. Moreover, $s'(t_{\text{inflect}}) \to S(\Delta x)\infty$ as $t_{\text{right}} \to 0$.

When t_L is the smallest positive value between t_L and t_R, setting $t_{\text{right}} = t_L$ gives $t_{\text{inflect}} = 0$ and $s'(t_{\text{inflect}}) = v_{\text{left}}$. Decreasing t_{right} into the interval $(0, t_L)$ yields $t_{\text{inflect}} \in (0, t_{\text{right}})$. Since $s'(t_{\text{inflect}})$ changes monotonically for $t_{\text{right}} \in (0, t_L)$, setting Eq. (128a) to be equal to $S(\Delta x)v_{\max}$ leads to a quadratic equation for t_{right}

$$v_{\text{left}} + \Delta x \frac{3(1 - \frac{t_{\text{right}}}{t_L})^2}{t_{\text{right}}(1 - \frac{t_{\text{right}}}{t_L} + 1 - \frac{t_{\text{right}}}{t_R})} = S(\Delta x)v_{\max} \tag{130}$$

which has at most one solution in the interval $(0, t_L)$. It has no solution when $v_{\text{left}} = S(\Delta x)v_{\max}$; in this case, $t_{\text{right}} = t_L$ is the minimum allowable t_{right}.

When t_R is the smallest positive value between t_L and t_R, setting $t_{\text{right}} = t_R$ gives $t_{\text{inflect}} = t_{\text{right}}$ and $s'(t_{\text{inflect}}) = v_{\text{right}}$. Decreasing t_{right} into the interval $(0, t_R)$ yields $t_{\text{inflect}} \in (0, t_{\text{right}})$. Since $s'(t_{\text{inflect}})$ changes monotonically for $t_{\text{right}} \in (0, t_R)$, setting Eq. (128b) to be equal to $S(\Delta x)v_{\max}$ leads to a quadratic equation for t_{right}

$$v_{\text{right}} + \Delta x \frac{3(1 - \frac{t_{\text{right}}}{t_R})^2}{t_{\text{right}}(1 - \frac{t_{\text{right}}}{t_L} + 1 - \frac{t_{\text{right}}}{t_R})} = S(\Delta x)v_{\max} \tag{131}$$

which has at most one solution in the interval $(0, t_R)$. It has no solution when $v_{\text{right}} = S(\Delta x)v_{\max}$; in this case, $t_{\text{right}} = t_R$ is the minimum allowable t_{right}.

When $t_L = t_R > 0$, $v_{\text{left}} = v_{\text{right}}$ and they have the same sign as $S(\Delta x)$. Setting $t_{\text{right}} = t_{LR} = t_L = t_R$ gives $q = a = 0$ leading to $s'' = 0$ for all t. Decreasing t_{right} into the interval $(0, t_{LR})$ yields $t_{\text{inflect}} = \frac{1}{2} t_{\text{right}}$. Since $s'(t_{\text{inflect}})$ changes monotonically for $t_{\text{right}} \in (0, t_{LR})$, either Eq. (130) or identically Eq. (131) can be used to determine the minimum allowable t_{right}. These equations degenerate to be linear (instead of quadratic) in this case; in addition, $t_{\text{right}} = t_{LR}$ is the minimal allowable t_{right} when $v_{\text{left}} = S(v_{\text{left}})v_{\max}$ or equivalently $v_{\text{right}} = S(v_{\text{right}})v_{\max}$.

When $t_L < 0$ and $t_R < 0$, the parenthetical term multiplying t_{right} in Eq. (127) is always in the interval $(0, 1)$ implying that $t_{\text{inflect}} \in (0, t_{\text{right}})$. Eq. (129) still states that $s'(t_{\text{inflect}})$ changes monotonically, and Eq. (128c) still states that $s'(t_{\text{inflect}}) \to S(\Delta x)\infty$ as $t_{\text{right}} \to 0$. When $t_R \le t_L < 0$, Eq. (128a) leads to

$$\lim_{t_{\text{right}} \to \infty} s'(t_{\text{inflect}}) = v_{\text{left}} - \frac{3\Delta x}{t_L + \frac{t_L^2}{t_R}} = -v_{\text{left}} \left(\frac{3}{2 + \frac{t_L}{t_R}\left(1 - \frac{t_L}{t_R}\right)} - 1 \right) \tag{132}$$

where the parenthetical term lies in the interval $[\frac{1}{3}, \frac{1}{2}]$ with the right endpoint achieved when $t_R = -\infty$; thus, Eq. (130) can be used. When $t_L \leq t_R < 0$, Eq. (128b) leads to

$$\lim_{t_{\text{right}}\to\infty} s'(t_{\text{inflect}}) = v_{\text{right}} - \frac{3\Delta x}{\frac{t_R^2}{t_L} + t_R} = -v_{\text{right}}\left(\frac{3}{2 + \frac{t_R}{t_L}\left(1 - \frac{t_R}{t_L}\right)} - 1\right) \tag{133}$$

where the parenthetical term lies in the interval $[\frac{1}{3}, \frac{1}{2}]$ with the right endpoint achieved when $t_L = -\infty$; thus, Eq. (131) can be used. Note that $t_L = t_R = -\infty$, has $s'(t_{\text{inflect}}) \to 0$ as $t_{\text{right}} \to \infty$ and either Eq. (130) or (131) can be used.

8.2. Three-dimensional case

Motivated by the one-dimensional analysis, we take the limit as $t_{\text{right}} \to \infty$ in order to bound $\|\vec{s}'\|_2$ on $t \in (0, t_{\text{right}})$, implying existence of a minimum t_{right} that limits the speed to not exceed $v_{\max}$. Let

$$\lim_{t_{\text{right}}\to\infty} \vec{s}(t) = \hat{s}(t) = t_{\text{right}}\left(\hat{q}(\tfrac{t}{t_{\text{right}}})^3 + \hat{a}(\tfrac{t}{t_{\text{right}}})^2 + \vec{b}\tfrac{t}{t_{\text{right}}} + \vec{c}\right) = t_{\text{right}}(\hat{q}\hat{t}^3 + \hat{a}\hat{t}^2 + \vec{b}\hat{t} + \vec{c}) \tag{134}$$

where t_{right} has been pulled out of Eq. (117) to obtain

$$\hat{q} = \vec{v}_{\text{left}} + \vec{v}_{\text{right}} \tag{135a}$$

$$\hat{a} = -2\vec{v}_{\text{left}} - \vec{v}_{\text{right}} \tag{135b}$$

where the change of variable from t to $\hat{t} = \frac{t}{t_{\text{right}}}$ makes $[0, 1]$ the interval of interest. Eqs. (118a)and (118b) can be rewritten as

$$\|\hat{s}'(t)\|_2^2 = \frac{\hat{s}'(\hat{t})}{t_{\text{right}}} \cdot \frac{\hat{s}'(\hat{t})}{t_{\text{right}}} = (3\hat{q}\hat{t}^2 + 2\hat{a}\hat{t} + \vec{b}) \cdot (3\hat{q}\hat{t}^2 + 2\hat{a}\hat{t} + \vec{b}) \tag{136a}$$

$$\frac{d}{dt}\frac{\|\hat{s}'(t)\|_2^2}{2} = \frac{\hat{s}''(\hat{t})}{t_{\text{right}}^2} \cdot \frac{\hat{s}'(\hat{t})}{t_{\text{right}}} = \frac{1}{t_{\text{right}}}(6\hat{q}\hat{t} + 2\hat{a}) \cdot (3\hat{q}\hat{t}^2 + 2\hat{a}\hat{t} + \vec{b}) \tag{136b}$$

noting that $\hat{s}'(t) = \frac{1}{t_{\text{right}}}\hat{s}'(\hat{t})$. Let $\hat{t}_{\text{inflect}} = \frac{t_{\text{inflect}}}{t_{\text{right}}}$ be the solution to

$$(6\hat{q}\hat{t}_{\text{inflect}} + 2\hat{a}) \cdot (3\hat{q}\hat{t}_{\text{inflect}}^2 + 2\hat{a}\hat{t}_{\text{inflect}} + \vec{b}) = 0 \tag{137}$$

corresponding to the location of the speed extrema; then, multiplying Eq. (137) by $\frac{1}{2}\hat{t}_{\text{inflect}}$ and subtracting it from Eq. (136a) gives

$$\|\hat{s}'(t_{\text{inflect}})\|_2^2 = (\hat{a}\hat{t}_{\text{inflect}} + \vec{b}) \cdot (3\hat{q}\hat{t}_{\text{inflect}}^2 + 2\hat{a}\hat{t}_{\text{inflect}} + \vec{b}) \tag{138}$$

reducing Eq. (136a) from a quartic to a cubic. Substituting in $\hat{q}$ and $\hat{a}$ leads to

$$\begin{aligned}\|\hat{s}(t_{\text{inflect}})\|_2^2 = {} & \vec{v}_{\text{left}} \cdot \vec{v}_{\text{left}}\left(-6\hat{t}_{\text{inflect}}^3 + 11\hat{t}_{\text{inflect}}^2 - 6\hat{t}_{\text{inflect}} + 1\right) \\ & + \vec{v}_{\text{left}} \cdot \vec{v}_{\text{right}}\left(-9\hat{t}_{\text{inflect}}^2 + 11\hat{t}_{\text{inflect}} - 3\right)\hat{t}_{\text{inflect}} \\ & + \vec{v}_{\text{right}} \cdot \vec{v}_{\text{right}}\left(-3\hat{t}_{\text{inflect}} + 2\right)\hat{t}_{\text{inflect}}^2\end{aligned} \tag{139}$$

as a weighted sum of cubics in $\hat{t}_{\text{inflect}}$. It remains to show that Eq. (139) is properly bounded for any $\hat{t}_{\text{inflect}} \in [0, 1]$ that satisfies Eq. (137), assuming that $\|\vec{v}_{\text{left}}\|_2$ and $\|\vec{v}_{\text{right}}\|_2$ are also properly bounded.

Remark 8.3. *Setting $\hat{t}_{inflect} = 1$ in Eq. (137) leads to $-\vec{v}_{left} \cdot \vec{v}_{right} = 2\|\vec{v}_{right}\|_2^2$; then, setting $\hat{t}_{inflect} = 0$ and $\hat{t}_{inflect} = 1$ in Eq. (139) gives $\|\vec{v}_{left}\|_2^2$ and $\|\vec{v}_{right}\|_2^2$ respectively.*

Eq. (139) can be bounded via

$$\frac{\|\hat{s}(t_{\text{inflect}})\|_2^2}{\|\vec{v}_{\text{left}}\|_2^2} \leq -6\hat{t}_{\text{inflect}}^3 + 11\hat{t}_{\text{inflect}}^2 - 6\hat{t}_{\text{inflect}} + 1 + \alpha\left|-9\hat{t}_{\text{inflect}}^3 + 11\hat{t}_{\text{inflect}}^2 - 3\hat{t}_{\text{inflect}}\right| + \alpha^2(-3\hat{t}_{\text{inflect}}^3 + 2\hat{t}_{\text{inflect}}^2) \tag{140}$$

with $0 \leq \alpha = \frac{\|\vec{v}_{\text{right}}\|_2}{\|\vec{v}_{\text{left}}\|_2} \leq 1$ when $\|\vec{v}_{\text{right}}\|_2 \leq \|\vec{v}_{\text{left}}\|_2$ and

$$\frac{\|\hat{s}(t_{\text{inflect}})\|_2^2}{\|\vec{v}_{\text{right}}\|_2^2} \leq \beta^2(-6\hat{t}_{\text{inflect}}^3 + 11\hat{t}_{\text{inflect}}^2 - 6\hat{t}_{\text{inflect}} + 1) + \beta\left|-9\hat{t}_{\text{inflect}}^3 + 11\hat{t}_{\text{inflect}}^2 - 3\hat{t}_{\text{inflect}}\right| - 3\hat{t}_{\text{inflect}}^3 + 2\hat{t}_{\text{inflect}}^2 \tag{141}$$

with $0 \leq \beta = \frac{\|\vec{v}_{\text{left}}\|_2}{\|\vec{v}_{\text{right}}\|_2} < 1$ otherwise. Note that the middle terms would normally have a $\cos\theta$ when rewriting Eq. (139) using an equality; however, Eqs. (140) and (141) assume $\cos\theta = \pm 1$ wherever necessary in order to obtain the absolute value function in the

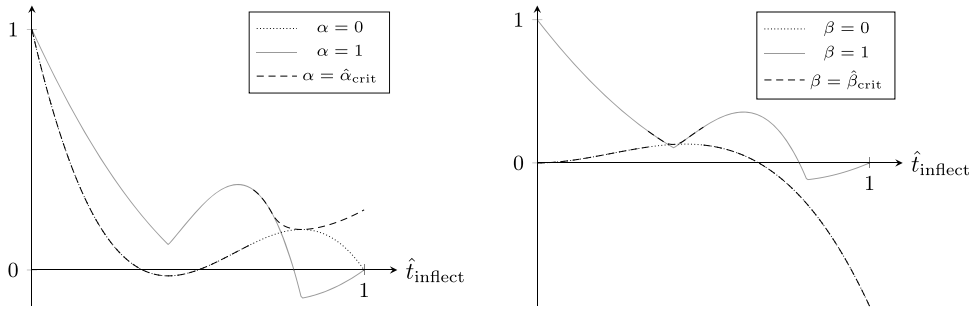

Fig. 12. Left: Plot of the bound in Eq. (140) for $\alpha = 0$, $\alpha = 1$, and $\alpha = \hat{\alpha}_{\text{crit}}$. Right: Plot of the bound in Eq. (141) for $\beta = 0$, $\beta = 1$, and $\beta = \hat{\beta}_{\text{crit}}$.

bound. Since the right-hand side of Eq. (140) is a quadratic in α, it is straightforward to find the global extremum α_{crit} as a function of $\hat{t}_{\text{inflect}}$. Since $\alpha \in [0, 1]$, $\hat{\alpha}_{\text{crit}} = \min(\max(\alpha_{\text{crit}}, 0), 1)$ can be plugged into Eq. (140) in order to determine the global maximum or minimum of the quadratic at each $\hat{t}_{\text{inflect}}$ defaulting to one of the boundary values (either $\alpha = 0$ or $\alpha = 1$) when $\alpha_{\text{crit}} \notin (0, 1)$. Fig. 12 left illustrates that the right-hand side of Eq. (140) is bounded from above by a value of 1 for all $\hat{t}_{\text{inflect}} \in [0, 1]$ for $\alpha = 0$, $\alpha = 1$, and $\alpha = \hat{\alpha}_{\text{crit}}$ implying that $\left\|\hat{s}(t_{\text{inflect}})\right\|_2 \leq \left\|\vec{v}_{\text{left}}\right\|_2$. Fig. 12 right similarly illustrates that $\left\|\hat{s}(t_{\text{inflect}})\right\|_2 \leq \left\|\vec{v}_{\text{right}}\right\|_2$ after applying the same process to β.

9. Safe sets

Consider a temporal discretization of the pursuer-evader problem with individual time steps $\Delta t = t_{n+1} - t_n$. Assuming an evader has uncertainty over the pursuer's strategy and trajectory, they might aim to avoid a sphere of radius $v_p \Delta t$ as long as they can bound the pursuer's maximum speed by v_p. As pointed out in Remark 4.2, discretizing the sphere works for the pursuer but not for the evader. Thus, we instead consider an axis-aligned bounding box with side length $2v_p\Delta t$ around the pursuer. Note that the axis-aligned bounding box overapproximates the sphere, limiting potential strategies. As future work, it is worth considering other bounding volumes as well as bounding volume hierarchies and spatial coverings, see e.g., [56,66–70]. In fact, object oriented bounding boxes are straightforward to use, simply by changing the coordinate system to make them axis aligned.

An axis-aligned bounding box is the interior region formed by the intersections of three sets of parallel planes. Deducing collisions with the box can be decomposed into subcases noting that the ordering of the collisions with the planes dictates which of the 27 subregions the evader lies in. Importantly, this facilitates root analysis in each dimension independently. Using time-varying corners leads to a continuous collision detection (CCD) formulation similar in spirit to what was discussed in Sections 6.3 and 9. Temporally discretizing the corner locations simplifies the formulation, since only a stationary box needs to be considered; however, the stationary box teleports from one location to another every time step. Given the uncertainty of the purser's location inside the axis-aligned box, the evader needs to avoid ever being inside of it. In order for the evader to avoid being inside the new stationary box at the beginning of the next time step, it must end the time step outside a larger stationary box with side length $4v_p\Delta t$. See Fig. 13.

When the evader starts a time step outside of the larger box, it merely needs to avoid collisions with it; otherwise, the evader needs to escape the larger box while avoiding collisions with the smaller box. A straightforward way to deal with escaping the larger box is to decompose the evader's trajectory into two splines which have a common endpoint on the larger box. The first spline needs to avoid collisions with the smaller box, and the second spline needs to avoid collisions with the larger box. Importantly, the evader needs to traverse the entire first spline during the time step. See Fig. 13. Notably, this two spline approach ensures the evader only seeks to avoid collisions, since the sought-after collision with the larger box is captured via the endpoint between the two splines. In the more general case when the evader's trajectory is modeled by a large number of splines that interpolate a simulation of the evader's physics, we are advocating the splitting of the multi-spline interpolation so that no spline crosses the larger box. Note that root-finding would need to be subsequently addressed on every relevant spline.

Assuming that the velocity between the two splines is continuous, there are six degrees of freedom on the common point between the splines: the three-dimensional velocity, the two-dimensional position, and the time it takes to traverse the first spline (as discussed in Section 8 and Remark 8.1). In the more general case where the common point between two splines is not constrained in any way, there are seven degrees of freedom at the common point. As usual, any extra degrees of freedom can be chosen to best satisfy the evader's goals.

Remark 9.1. *It is not necessary for the evader to avoid collisions with the larger box when it is outside of it, since an even number of collisions is sufficient; however, it is important that the evader avoid collisions with the smaller box any time they are inside the larger box.*

9.1. Avoiding axis-aligned bounding boxes

Regardless of whether the evader is trying to avoid collisions with the smaller or larger box, the problem is identical. The interior of an axis-aligned bounding box is the intersection of the interiors of a pair of parallel planes in every dimension. The evader is free to enter and leave the interior regions between pairs of planes as long as it stays outside of at least one of them at all times.

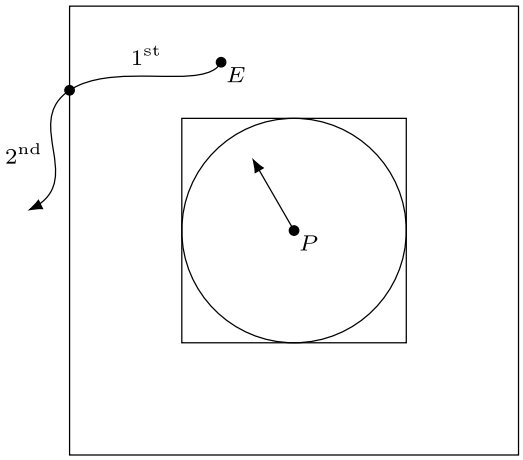

Fig. 13. In $\mathbb{R}^2$, the circle indicates potential locations of the pursuer in $t \in [t_n, t_{n+1}]$, and the smaller square is an axis-aligned overapproximation to the circle. The larger square represents the set of locations that the smaller square could contain at time t_{n+1} if the pursuer were able to reach any location in the smaller square. The evader's first spline needs to avoid collisions with the smaller square, and its second spline needs to avoid collisions with the larger square. Importantly, the evader needs to traverse the entire first spline during the time step.

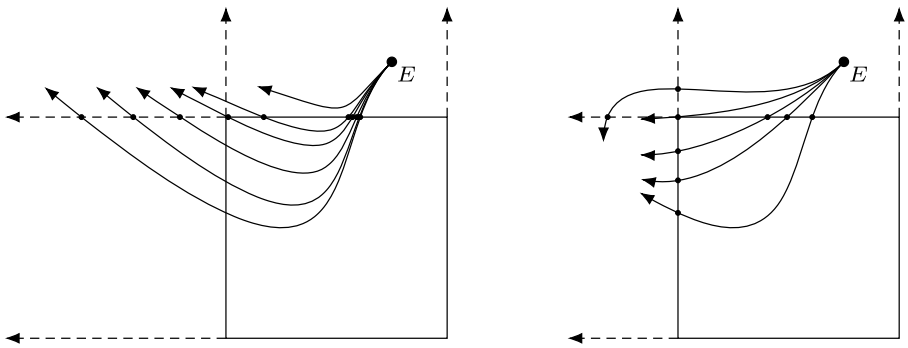

Fig. 14. Left: The evader is inside the vertical pair of lines and is aiming to stay outside of the horizontal pair. It accomplishes this by iterating on its degrees of freedom in order to merge the two real roots with the top horizontal line together in order to make them complex. (Six iterations are shown, progressing from the bottom spline to top spline.) Right: The evader is inside the vertical pair of lines and is aiming to exit them before entering the horizontal pair. It accomplishes this by iterating on its degrees of freedom in order to force an exiting-an-interior-region real root before the problematic entering-an-interior-region real root. When the roots are equal, the evader touches the corner; moreover, forcing a gap between the roots creates extra clearance with the corner. (Five iterations are shown, progressing from the bottom spline to top spline.)

When the evader is inside of all but one interior region, it needs to avoid entering the last interior region before exiting another one. Staying outside of the last interior region means avoiding collisions with the plane in the relevant dimension that is closest to the evader. This can be accomplished by either pushing all of the real roots (indicating collisions with that plane) to be outside of the time interval of interest or by merging real roots inside the time interval of interest with each other to make them complex. See Fig. 14 left. Alternatively, the evader can exit one of the other interior regions before entering the last one. This can be accomplished by forcing at least one exiting-an-interior-region real root to occur before the problematic entering-an-interior-region real root. See Fig. 14 right.

Remark 9.2. *Note the similarity between Fig. 14 left and Fig. 4. The one-dimensional problem considered in Section 5 is essentially equivalent to the vertical direction in Fig. 14 left, with both aiming to merge two real roots into a complex conjugate pair so that the evader can turn around before colliding. The ability to simplify more difficult problems by orthogonalizing constraints and/or modifying coordinate systems should not be underestimated.*

9.2. Escaping axis-aligned bounding boxes

As discussed above, the evader needs to escape the larger box before the end of the time step. This can be accomplished by creating an intersection with any one of the planes defining the interior of the box. For the sake of simplicity, each plane can be considered separately before choosing the one that leads to the best strategy. As discussed above, a straightforward way to address this is via a two spline strategy where the first spline has an endpoint on the plane of interest and the second spline avoids collisions with the

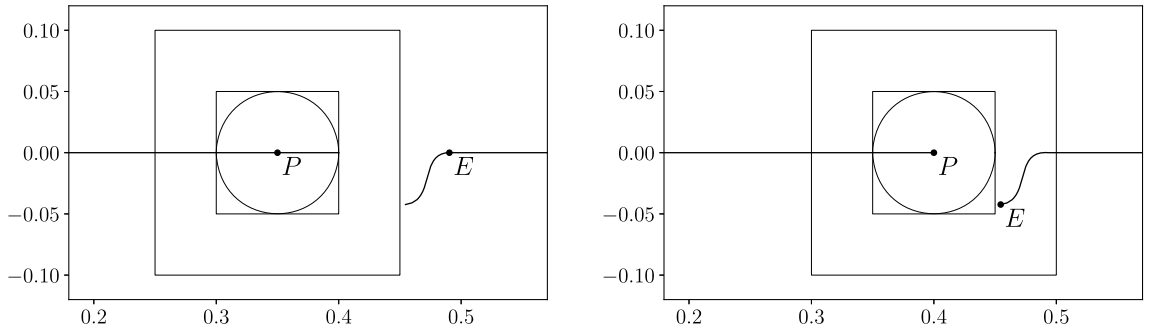

Fig. 15. Two successive time steps (left and right, respectively), each illustrating the current and prior locations of the pursuer and evader. The first figure also shows the planned trajectories during the time step, and the second figure shows the pursuer and evader's locations after moving through that planned trajectory. The pursuer and its fictitious boxes move directly to the right putting the evader into a precarious position, even though its predicted spline adequately avoided intersecting the larger box.

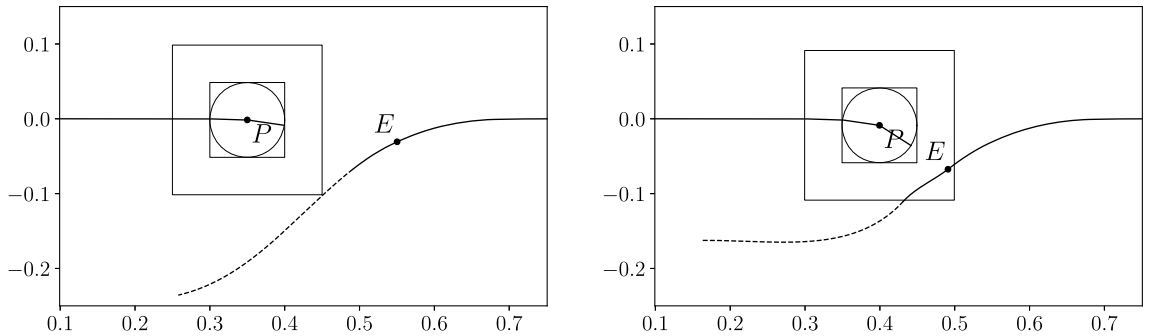

Fig. 16. Using long term planning (dashed lines), instead of a greedy short-term approach, allows the evader to begin reacting earlier (left) than in Fig. 15 and thus be in a better position as it gets closer to the pursuer (right).

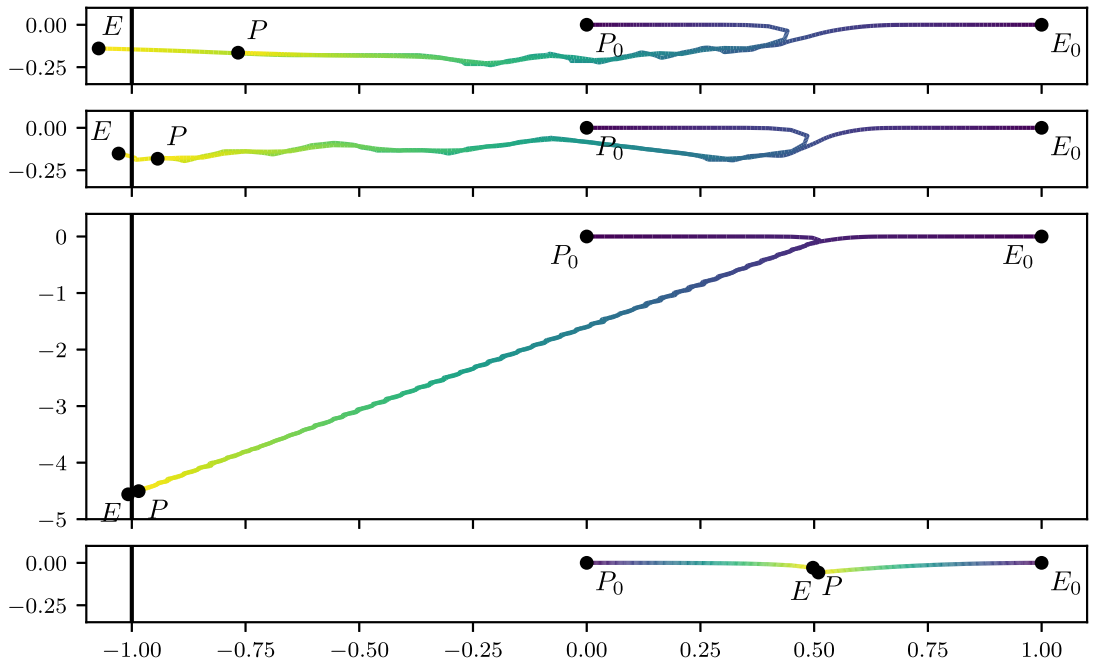

Fig. 17. Sample game trajectories when the evader has $v_{\max} = 1.5$, 1.25, 1.10 and 1.0 respectively (top to bottom). In spite of increasing the long term planning window the evader always lost the game with $v_{\max} = 1.0$ in our simulations (as might be expected). The color gradient of the lines indicates time.

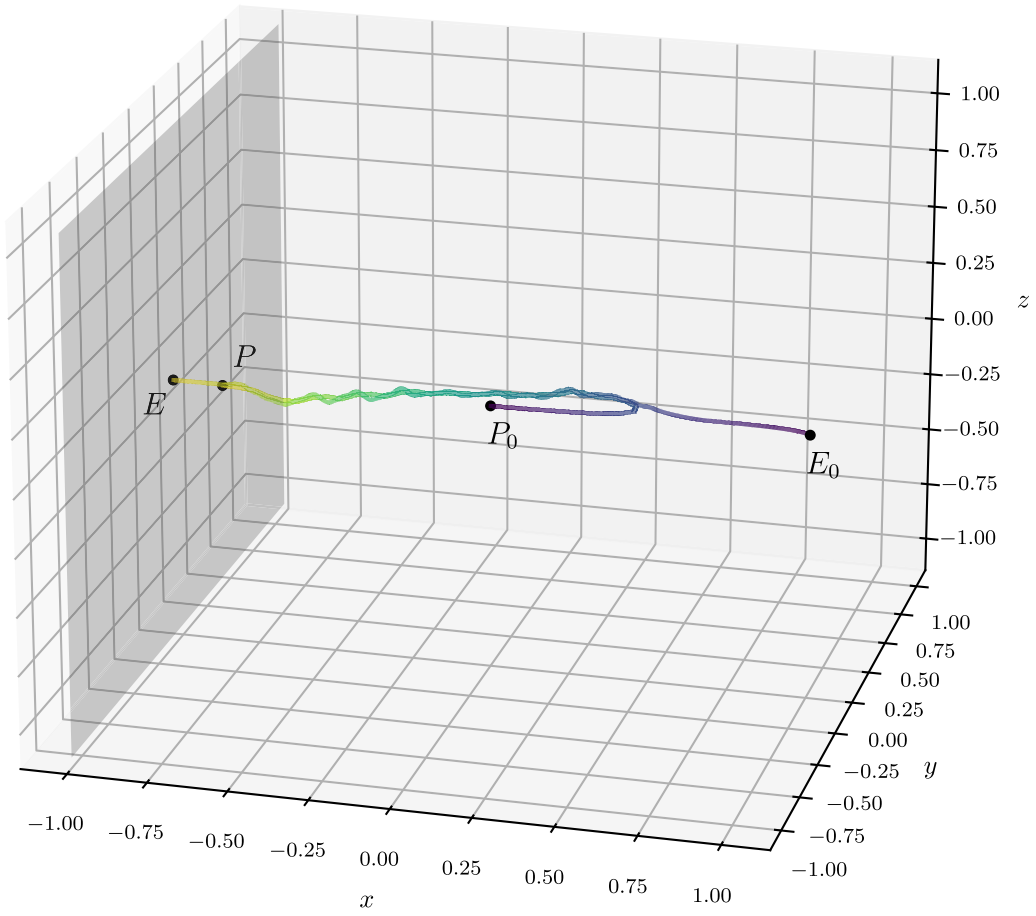

Fig. 18. Same example as in Figs. 16 and 17, except in three spatial dimensions. The pursuer starts at the origin, and the evader starts at $(1, 0, 0)$ and aims at $(-1, 0, 0)$. In order to prevent the game from degenerating into two spatial dimensions, the evader's initial velocity is chosen as $(-0.9, 0.25, 0.25)$. The evader's maximum speed is $v_{\max} = 1.5$. The color gradient of the lines indicates time. See also Fig. 19.

box. The first spline is restricted to have its endpoint lie on the plane of interest and its velocity point outwards from the interior region between the two planes in the dimension of interest.

Since the evader needs to traverse the entire first spline during the time step, t_{right} should be minimized while also appropriately bounding the speed below $v_{\max}$ as discussed in Section 8. A greedy strategy would be for the evader to travel in a straight line. In this case, the analysis in Section 8.1 could be used to determine t_{right}; however, there is no guarantee that this t_{right} occurs before the end of the time step. In fact, although Section 8.2 showed that a t_{right} does exist in every case, it is possible that none of them occur before the end of the time step. This means that minimizing t_{right} might not lead to the evader escaping the box, but it at least puts them in a good position to do so soon. However, it is important to keep in mind that the first spline needs to avoid collisions with the smaller box.

Remark 9.3. *Similar to the discretization of the sphere proposed for the pursuer in Remark 4.2 the evader could consider a discretization of the box; moreover, discretization allows the evader to replace the larger box with a more accurate sphere (even though the smaller box cannot be replaced). Exiting velocity direction and magnitude could also be discretized, at the risk of increasing the dimension of the discretization.*

10. Combined objectives

Consider a game where the pursuer, with constant speed $v_p = 1$, starts at $(0, 0)$ and uses the evader's position and velocity to aim for a predicted position of the evader according to Eq. (116). The evader starts at $(1, 0)$ and its strategy is to cross the goal line at $x = -1$; as opposed to fleeing towards $y = \pm\infty$ goal line locations, the evader aims for a point as close as possible to $(-1, 0)$ in order to minimize some concept of urgency. The evader has an initial velocity $(-1, 0)$, and a maximum speed $v_{\max} = 1.5$, allowing the evader to move faster than the pursuer when their spline is roughly linear, but limiting their speed to be closer to that of the pursuer when maneuvering.

In order to maintain safety from the pursuer and embrace uncertainty along the lines of Section 9, the evader uses the two spline strategy with its first spline (when it exists) ending on the boundary of the larger box and its second spline progressing towards the

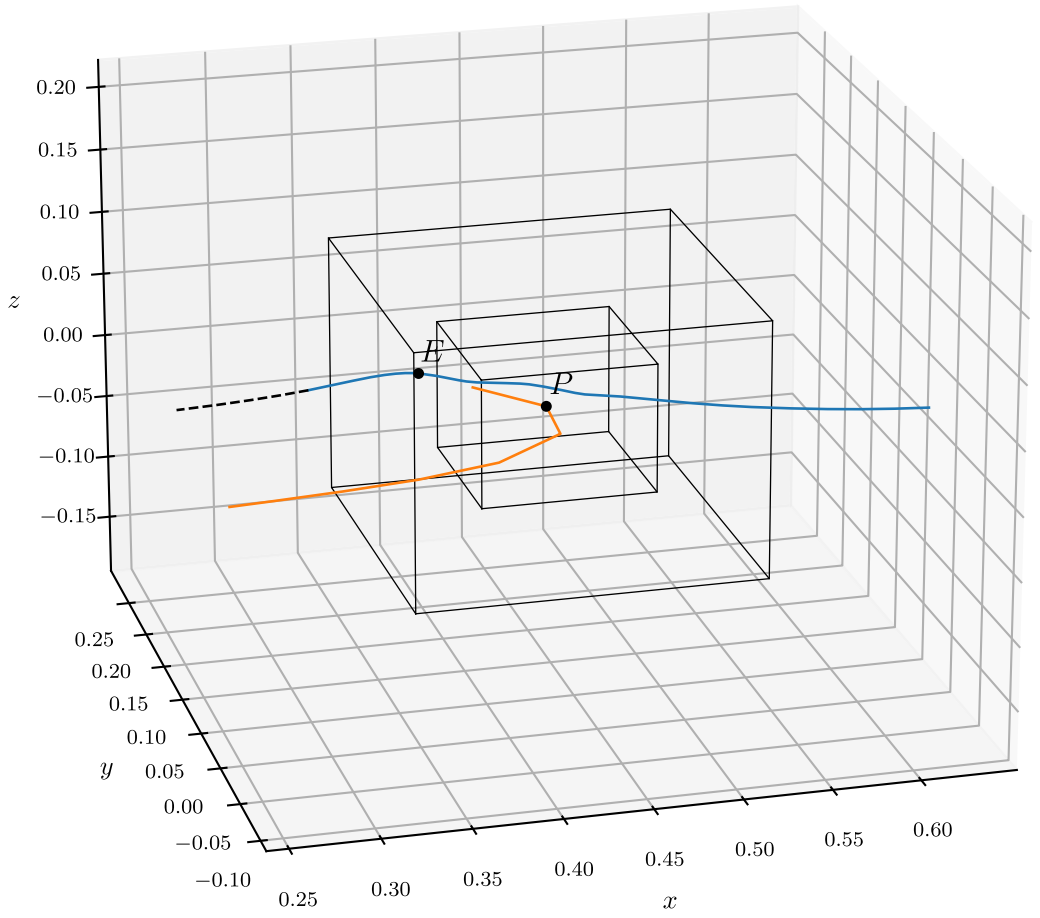

Fig. 19. A zoomed in view of a time step from Fig. 18 in order to illustrate the fictitious boxes in three spatial dimensions. The evader's plan is shown as the dashed line, with the solid orange and blue lines indicating the pursuer's and evader's past and current-step trajectories, respectively.(For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

goal line. Let Δt_1 and Δt_2 be the time allocated to the first and second spline respectively with $\Delta t_1 + \Delta t_2 = \Delta t$ and $\Delta t_1 = 0$ when the evader starts outside the larger box. As discussed in Sections 8 and 9, the evader aims to minimize Δt_1 in order to escape the larger box as soon as possible. This is accomplished by sampling a discretization of the faces of the box to find possible exit points and a discretization of the hemisphere normal to each face to find possible exit velocities. The corresponding t_{right} for each sample pair is minimized (via bisection) subject to the constraint that $v_{\max}$ is not exceeded on the spline; then, the sample pair that minimizes t_{right} (or satisfies some other criteria) can be chosen. The endpoint of the second spline, determined by $\vec{x}_{\text{right}}$ and $\vec{v}_{\text{right}}$, should be chosen to progress towards to the goal line while avoiding collisions with the larger box.

Fig. 15 illustrates two subsequent time steps in the game. In the first time step, the evader can no longer continue heading straight towards the goal line, and instead calculates a trajectory that bends downward so that it does not intersect with the larger box. In the second time step, after the evader gets to the end of its predicted spline and the pursuer and the fictitious boxes around it have moved, the evader ends up in a precarious situation where it will almost assuredly lose the game. Of course, a number of other considerations could be added to the evader's strategy in order to guarantee a more favorable outcome at the end of the time step. One of the most straightforward ways to address this issue without introducing a number of other strategies is to simply switch from greedy and myopic short-term planning to long-term planning. Notably, this can be accomplished in a straightforward manner with only a minor modification of our approach. Fig. 16 illustrates how long-term planning improves upon the dilemma shown in Fig. 15 without needing to introduce any other strategies or considerations. Instead of considering the second spline on Δt_2 it is extended to be considered on a $\Delta t_{\text{plan}} > \Delta t_2$. Nothing else need change.

Fig. 17 illustrates the trajectories obtained playing the game with various maximum velocities for the evader. Note that the evader's primary goal is to reach $x = -1$ and that it only starts aiming for $y = 0$ once it safely gets around the pursuer. Fig. 18 shows the trajectories obtained in three spatial dimensions. Notably, our formulation remains unchanged and the code is straightforward to implement. The root-finding and intersection code scales linearly with dimension, in comparison to the sampling code that scales more poorly due to the Newton-Cotes approach. The scaling of the sampling code can be fixed via Monte Carlo sampling.

Remark 10.1. *More generally, in $\mathbb{R}^N$, collisions with an axis-aligned bounding box can be interpreted as being inside N pairs of parallel planes; thus, once inside $N-1$ pairs of parallel planes, the goal is to exit one pair before entering the last pair. Choosing one of the $N-1$ pairs to exit can obviously be accomplished in linear time, and the problem degenerates to $\mathbb{R}^2$ once this pair has been chosen.*

11. Conclusion

We illustrated the efficacy of using cubic splines in order to interpolate physically simulated trajectories for the purpose of collision avoidance and path planning as well as the complimentary problem that seeks collisions during pursuit. Although more interesting objectives turned out to be intractable using distance-based formulations, a novel root-finding approach was shown to be tractable with the aid of [18]. Collisions are created by merging complex roots and avoided by merging real roots. The ability to explicitly and robustly handle the discontinuity between colliding and not colliding allows for gradient-based optimization between these two regimes, leading to a larger set of path planning strategies. As is typical, bounding volumes were used to represent uncertainty and spatial extent. This led to a natural asymmetry, since colliding requires intersecting some point on a bounding volume surface while collision avoidance requires missing every point. A pursuer can readily sample a surface of any type, whereas an evader may struggle to formulate algorithms for collision avoidance of complex volumes/surfaces. Notably, axis-aligned bounding boxes readily facilitate collision avoidance since they can be conceptualized as pairs of parallel planes thus decomposing collision avoidance into one-dimensional subproblems.

After illustrating the efficacy of our approach to pursuit and collision avoidance, we combined it with other goals and objectives in order to emphasize its potential for real-world impact. Most importantly, the ability to use gradient-based optimization allows our formulation to be incorporated into larger machine learning pipelines. It also facilitates the use of more complex controls via the chain rule. In addition, it is straightforward to incorporate traditional control and game theory concepts to provide more optimal estimates of an opponent's trajectory, such as iterated best response or bounded rationality [71,72]. Although our relatively simple examples already showed interesting emergent behavior, our current framework can also be used in more complex scenarios including (moving) obstacles, boundaries, etc. via the axis-aligned bounding box framework.

For the sake of exposition, we used a bound on the maximum velocity as a surrogate for the various limitations that would be imposed by the physics. An arc length bound is another useful surrogate, see [65]. It would be interesting to pursue more detailed coupling with numerical simulations of the physics. Although the axis-aligned bounding boxes facilitated incorporation of both uncertainty and spatial extent, overapproximations of safe sets restrict strategies. While it is straightforward to rotate the frame to consider object-oriented bounding boxes, our work could be improved by extending it to consider collisions with triangulated surfaces. There are robust CCD methods in the literature (dating as far back as [16]) that decompose triangulated surface conditions into coplanarity considerations, suggesting that our approach may be extendible to this case. Finally, a number of theoretical questions remain open. In order to extend [18], higher order polynomials would need to be understood better. It is unclear whether the sixth order polynomial in the distance function formulation has special structures that can be exploited due to it being the sum of the squares of third order polynomials. We have not settled on an optimal strategy for preserving complex-valued roots outside of merely aiming to keep the imaginary part large. Etc.

CRediT authorship contribution statement

Trevor Maxfield: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Ronald Fedkiw:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Data availability

No data was used for the research described in the article.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ronald Fedkiw reports financial support was provided by Office of Naval Research. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Research supported in part by ONR N00014-24-1-2644, ONR N00014-21-1-2771, and ONR N00014-19-1-2285. T.M. is supported in part by the Stanford Graduate Fellowship. The authors would like to thank Rohan Bansal for his contributions in developing the code and Obed Camacho for various useful discussions.

Appendix A. Reference equations

For reference, the partial derivatives of ψ, defined in Eq. (28), are

$$\frac{\partial \psi(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t} = \sum_i \left(\frac{\partial s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t} \frac{\partial s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t} + s_i^R(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)) \frac{\partial^2 s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t\, \partial t^R} + s_i^I(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)) \frac{\partial^2 s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t\, \partial t^I} \right) \tag{A.1a}$$

$$\frac{\partial \psi(t;\vec{\mathbf{p}}^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)} = \frac{\partial s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t} \frac{\partial s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)} + s_i^R(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)) \frac{\partial^2 s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)\, \partial t^R} + s_i^I(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)) \frac{\partial^2 s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)\, \partial t^I} \tag{A.1b}$$

where $\frac{\partial s_i}{\partial t}$ and $\frac{\partial s_i}{\partial \vec{p}_i^s}$ are given in Eq. (19) and the second derivatives

$$\frac{\partial^2 s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t\, \partial t^R} = \begin{bmatrix} (6t^R \;\; 2 \;\; 0 \;\; 0)\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E) & (-6t^I \;\; 0 \;\; 0 \;\; 0)\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E) \\ (6t^I \;\; 0 \;\; 0 \;\; 0)\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E) & (6t^R \;\; 2 \;\; 0 \;\; 0)\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E) \end{bmatrix} \tag{A.2a}$$

$$\frac{\partial^2 s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial t\, \partial t^I} = \begin{bmatrix} (-6t^I \;\; 0 \;\; 0 \;\; 0)\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E) & (-6t^R \;\; -2 \;\; 0 \;\; 0)\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E) \\ (6t^R \;\; 2 \;\; 0 \;\; 0)\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E) & (-6t^I \;\; 0 \;\; 0 \;\; 0)\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E) \end{bmatrix} \tag{A.2b}$$

$$\frac{\partial^2 s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)\, \partial t^R} = \begin{bmatrix} 3(t^R)^2 - 3(t^I)^2 & 2t^R & 1 & 0 \\ 6t^R t^I & 2t^I & 0 & 0 \end{bmatrix} \tag{A.2c}$$

$$\frac{\partial^2 s_i(t;\vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E))}{\partial \vec{p}_i^s(\vec{\lambda}^P,\vec{\lambda}^E)\, \partial t^I} = \begin{bmatrix} -6t^R t^I & -2t^I & 0 & 0 \\ 3(t^R)^2 - 3(t^I)^2 & 2t^R & 1 & 0 \end{bmatrix} \tag{A.2d}$$

follow directly by differentiating Eq. (19) directly with respect to t^R and t^I (assuming that $t \in \mathbb{C}$).

References

[1] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: an imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, 2019, pp. 8026–8037.

[2] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., {TensorFlow}: a system for {Large-Scale} machine learning, in: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 265–283.

[3] R. Collobert, K. Kavukcuoglu, C. Farabet, Torch7: a matlab-like environment for machine learning, in: BigLearn, NIPS Workshop, 2011.

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM International Conference on Multimedia, 2014, pp. 675–678.

[5] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, et al., Theano: a Python framework for fast computation of mathematical expressions, arXiv (2016) arXiv–1605.

[6] J. Bradbury R. Frostig P. Hawkins M.J. Johnson C. Leary D. Maclaurin G. Necula A. Paszke J. VanderPlas S. Wanderman-Milne Q. Zhang JAX: composable transformations of Python + NumPy programs,(2018). http://github.com/jax-ml/jax

[7] D. Johnson, T. Maxfield, Y. Jin, R. Fedkiw, Software-based automatic differentiation is flawed, arXiv:2305.03863 (2023), in review.

[8] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (2) (1988) 439–471.

[9] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer New York, 2003.

[10] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (2) (1999) 457–492.

[11] X.D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (1) (2000) 151–178.

[12] M. Kang, R. Fedkiw, X.D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (2000) 323–360.

[13] Z. Li, K. Ito, The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains, SIAM, 2006.

[14] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, Int. J. Numer. Methods Eng. 46 (1) (1999) 131–150.

[15] S. Redon, A. Kheddar, S. Coquillart, Fast continuous collision detection between rigid bodies, in: Computer Graphics Forum, 21, Wiley Online Library, 2002, pp. 279–287.

[16] R. Bridson, R. Fedkiw, J. Anderson, Robust treatment of collisions, contact and friction for cloth animation, in: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, 2002, pp. 594–603.

[17] T. Brochu, E. Edwards, R. Bridson, Efficient geometrically exact continuous collision detection, ACM Trans. Graph. 31 (4) (2012) 1–7.

[18] D. Johnson, R. Fedkiw, Addressing discontinuous root-finding for subsequent differentiability in machine learning, inverse problems, and control, J. Comput. Phys. 497 (2024) 112624.

[19] R. Isaacs, Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization, Control and Optimization, John Wiley and Sons, New York (1965).

[20] L.S. Pontryagin, On the theory of differential games, Russ. Math. Surv. 21 (4) (1966) 193.

[21] L.A. Petrosjan, Differential Games of Pursuit, 2, World Scientific, 1993.

[22] E. Dockner, Differential Games in Economics and Management Science, Cambridge University Press, 2000.

[23] A. Friedman, Differential Games, Courier Corporation, 2013.

[24] A. Mehlmann, Applied Differential Games, Springer Science & Business Media, 2013.

[25] M. Bardi, M. Falcone, P. Soravia, Numerical methods for pursuit-evasion games via viscosity solutions, in: Stochastic and Differential Games: Theory and Numerical Methods, Springer, 1999, pp. 105–175.

[26] T. Raivio, H. Ehtamo, On the numerical solution of a class of pursuit-evasion games, in: Advances in Dynamic Games and Applications, Springer, 2000, pp. 177–192.

[27] M. Falcone, Numerical methods for differential games based on partial differential equations, Int. Game Theory Rev. 8 (02) (2006) 231–272.
[28] M. Pontani, B.A. Conway, Numerical solution of the three-dimensional orbital pursuit-evasion game, J. Guid. Control Dynam. 32 (2) (2009) 474–487.
[29] E. Garcia, D.W. Casbeer, A. Von Moll, M. Pachter, Multiple pursuer multiple evader differential games, IEEE Trans. Automat. Control 66 (5) (2020) 2345–2350.
[30] D. Chen, Y. Wei, L. Wang, C.S. Hong, L.C. Wang, Z. Han, Deep reinforcement learning based strategy for quadrotor UAV pursuer and evader problem, in: 2020 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2020, pp. 1–6.
[31] J. Bertram, P. Wei, An efficient algorithm for multiple-pursuer-multiple-evader pursuit/evasion game, in: AIAA Scitech 2021 Forum, 2021, p. 1862.
[32] J. Silveira, K. Cabral, C.A. Rabbath, S. Givigi, Deep reinforcement learning solution of reach-avoid games with superior evader in the context of unmanned aerial systems, in: 2023 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2023, pp. 911–918.
[33] P. Li, S. Li, X. Wang, J. Cernỳ, Y. Zhang, S. McAleer, H. Chan, B. An, Grasper: a generalist pursuer for pursuit-evasion problems, in: Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, 2024, pp. 1147–1155.
[34] Y. Guan, W. Xu, G. Liu, Hybrid-pursuit strategies in multiple pursuer-evader games using reinforcement learning, IEEE Access 12 (2024) 187709–187721.
[35] Y. Chen, Y. Shi, X. Dai, Q. Meng, T. Yu, Pursuit-evasion game with online planning using deep reinforcement learning, Appl. Intell. 55 (6) (2025) 1–17.
[36] C. Cleveland, D. Liao, R. Austin, Physics of cancer propagation: a game theory perspective, AIP Adv. 2 (1) (2012) 11202.
[37] D. Greiner, J. Periaux, J.M. Emperador, B. Galván, G. Winter, Game theory based evolutionary algorithms: a review with nash applications in structural engineering optimization problems, Arch. Comput. Methods Eng. 24 (2017) 703–750.
[38] F. Borra, L. Biferale, M. Cencini, A. Celani, Reinforcement learning for pursuit and evasion of microswimmers at low Reynolds number, Phys. Rev. Fluids 7 (2) (2022) 023103.
[39] K. Weinberg, L. Stainier, S. Conti, M. Ortiz, Data-driven games in computational mechanics, Comput. Methods Appl. Mech. Eng. 417 (2023) 116399. https://doi.org/10.1016/j.cma.2023.116399
[40] S.M. LaValle, J.J. Kuffner, Jr, Randomized kinodynamic planning, Int. J. Rob. Res. 20 (5) (2001) 378–400.
[41] P. Falcone, F. Borrelli, J. Asgari, H.E. Tseng, D. Hrovat, Predictive active steering control for autonomous vehicle systems, IEEE Trans. Control Syst. Technol. 15 (3) (2007) 566–580.
[42] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, Int. J. Rob. Res. 30 (7) (2011) 846–894.
[43] G. Williams, N. Wagener, B. Goldfain, P. Drews, J.M. Rehg, B. Boots, E.A. Theodorou, Information theoretic mpc for model-based reinforcement learning, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2017, pp. 1714–1721.
[44] J. Schulman, J. Ho, A.X. Lee, I. Awwal, H. Bradlow, P. Abbeel, Finding locally optimal, collision-free trajectories with sequential convex optimization, in: Robotics: Science and Systems, 9, Berlin, Germany, 2013, pp. 1–10.
[45] X. Zhang, A. Liniger, F. Borrelli, Optimization-based collision avoidance, IEEE Trans. Control Syst. Technol. 29 (3) (2020) 972–983.
[46] M. Lutz, T. Meurer, Efficient formulation of collision avoidance constraints in optimization based trajectory planning and control, in: 2021 IEEE Conference on Control Technology and Applications (CCTA), IEEE, 2021, pp. 228–233.
[47] K. Tracy, T.A. Howell, Z. Manchester, Differentiable collision detection for a set of convex primitives, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 3663–3670.
[48] M. Yamamoto, Y. Isshiki, A. Mohri, Collision free minimum time trajectory planning for manipulators using global search and gradient method, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94), 3, IEEE, 1994, pp. 2184–2191.
[49] D. Rakita, H. Shi, B. Mutlu, M. Gleicher, Collisionik: a per-instant pose optimization method for generating robot motions with environment collision avoidance, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2021, pp. 9995–10001.
[50] Y. Kim, J. Kim, D. Park, Graphdistnet: a graph-based collision-distance estimator for gradient-based trajectory optimization, IEEE Rob. Autom. Lett. 7 (4) (2022) 11118–11125.
[51] Y. Zhi, N. Das, M. Yip, Diffco: autodifferentiable proxy collision detection with multiclass labels for safety-aware trajectory optimization, IEEE Trans. Rob. 38 (5) (2022) 2668–2685.
[52] S. Temizer, M. Kochenderfer, L. Kaelbling, T. Lozano-Pérez, J. Kuchar, Collision avoidance for unmanned aircraft using Markov decision processes, in: AIAA Guidance, Navigation, and Control Conference, 2010, p. 8040.
[53] H. Bai, D. Hsu, M.J. Kochenderfer, W.S. Lee, Unmanned aircraft collision avoidance using continuous-State POMDPs, in: Robotics: Science and Systems, Los Angeles, CA, 2011, pp. 1–8.
[54] X. Yu, X. Zhou, Y. Zhang, Collision-free trajectory generation for UAVs using Markov decision process, in: 2017 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2017, pp. 56–61.
[55] J. Bertram, P. Wei, J. Zambreno, A fast Markov decision process-based algorithm for collision avoidance in urban air mobility, IEEE Trans. Intell. Transp. Syst. 23 (9) (2022) 15420–15433.
[56] B. Davis, I. Karamouzas, S.J. Guy, NH-TTC: a gradient-based framework for generalized anticipatory collision avoidance, in: 16th Robotics: Science and Systems, RSS 2020, MIT Press Journals, 2020.
[57] N. Ratliff, M. Zucker, J.A. Bagnell, S. Srinivasa, CHOMP: gradient optimization techniques for efficient motion planning, in: 2009 IEEE International Conference on Robotics and Automation, IEEE, 2009, pp. 489–494.
[58] Z. Zhang, Y. Zhang, R. Han, L. Zhang, J. Pan, A generalized continuous collision detection framework of polynomial trajectory for mobile robots in cluttered environments, IEEE Rob. Autom. Lett. 7 (4) (2022) 9810–9817.
[59] T. Zhang, J. Wang, C. Xu, A. Gao, F. Gao, Continuous implicit sdf based any-shape robot trajectory optimization, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2023, pp. 282–289.
[60] T. Marcucci, P. Nobel, R. Tedrake, S. Boyd, Fast path planning through large collections of safe boxes, IEEE Trans. Rob. 40 (2024) 3795–3811.
[61] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. 91 (4) (2019) 045002.
[62] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. 51 (1) (2019) 357–377.
[63] S. Markidis, The old and the new: can physics-informed deep-learning replace traditional linear solvers?, Front. Big Data 4 (2021) 669097.
[64] P. Bouguer, Sur de nouvelles courbes auxquelles on peut donner le nom de lignes de poursuite, Mémoires de mathématique et de physique tiré des registres de l'Académie royale des sciences (1732) 1–15.
[65] T. Maxfield, On Devising More Competitive Strategies for Creating and Avoiding Collisions along Differentiable Trajectories, Ph.D. thesis, Stanford University, 2025. in preparation.
[66] J.K. Hahn, Realistic animation of rigid bodies, ACM Siggraph Comput. Graph. 22 (4) (1988) 299–308.
[67] R. Webb, M. Gigante, Using dynamic bounding volume hierarchies to improve efficiency of rigid body simulations, in: Visual Computing: Integrating Computer Graphics with Computer Vision, Springer, 1992, pp. 825–842.
[68] G. Barequet, B. Chazelle, L.J. Guibas, J.S.B. Mitchell, A. Tal, BOXTREE: a hierarchical representation for surfaces in 3D, in: Computer Graphics Forum, 15, Wiley Online Library, 1996, pp. 387–396.
[69] S. Gottschalk, M.C. Lin, D. Manocha, OBBTree: a hierarchical structure for rapid interference detection, in: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, 1996, pp. 171–180.
[70] M. Lin, S. Gottschalk, Collision detection between geometric models: a survey, in: Proc. of IMA Conference on Mathematics of Surfaces, 1, 1998, pp. 602–608.
[71] M.J. Kochenderfer, T.A. Wheeler, K.H. Wray, Algorithms for decision making, MIT press, 2022.
[72] N.M.T. Kokolakis, A. Kanellopoulos, K.G. Vamvoudakis, Bounded rationality in differential games: a reinforcement learning-based approach, in: Handbook of Reinforcement Learning and Control, Springer, 2021, pp. 467–489.