Time-Optimal Collaborative Guidance using the Generalized Hopf Formula

Matthew R. Kirchner, Robert Mar, Gary Hewer, Jérôme Darbon, Stanley Osher, Y.T. Chow

Abstract—Presented is a new method for calculating the time-optimal guidance control for a multiple vehicle pursuit-evasion system. A joint differential game of \( k \) pursuing vehicles relative to the evader is constructed, and a Hamilton-Jacobi-Isaacs (HJI) equation that describes the evolution of the value function is formulated. The value function is known such that the terminal cost is the squared distance from the boundary of the terminal surface. Additionally, all vehicles are assumed to have bounded controls. Typically, a joint state space constructed in this way would have too large a dimension to be solved with existing grid-based approaches. The value function is computed efficiently in high-dimensional space, without a discrete grid, using the generalized Hopf formula. The optimal time-to-reach is iteratively solved, and the optimal control is inferred from the gradient of the value function.

I. INTRODUCTION

One of the first successful implementations of control laws for pursuit problems is proportional navigation (PN) [27], which attempts to drive the rate of the line-of-sight vector between pursuer and evading target vehicle to zero. In this derivation, the target vehicle is assumed moving, but not maneuvering (turning). Generalizations of this concept attempt to estimate the vehicle maneuver [28], but these methods are not optimal since evasion strategy is not considered, i.e. not formulated as a differential game [29]. Additionally, this family of control laws does not account for control saturation. PN typically requires the magnitude of the control bound of the pursuer to be much greater than that of the evader to be successful, on the order of 3-5 times greater [28]. These guidance laws are strictly one-on-one in nature, and do not readily generalize to collaborative systems of multiple vehicles where the desired pursuit guidance is to ‘team’ together to capture a target. These early pursuit problems typically referred to controller designs as guidance laws, and in this letter we will use the terms controller and guidance interchangeably.

More recently, [29] proposed a solution to multi-vehicle pursuit evasion in a plane. In this case the problem was solved sub-optimally with heuristics in an effort to avoid the computational burden of direct solution to the Hamilton-Jacobi equation. Additionally, the method was based on simplified, single-integrator dynamics that require the vehicles to maneuver instantaneously to ensure capture.

A general alternative is to formulate the pursuit-evasion problem as a differential game, and derive a Hamilton-Jacobi-Isaacs (HJI) equation representing the optimal cost-to-go of the system. Traditionally, numerical solutions to HJI equations require a dense, discrete grid of the solution space [26], [24], [25]. Computing the elements of this grid scales poorly with dimension and has limited use for problems with dimension of greater than four. The exponential dimensional scaling in optimization is sometimes referred to as the “curse of dimensionality” [5], [4]. This phenomenon is seen clearly in [19], which formulated a differential game for a capture-the-flag problem and solved numerically on a four dimensional grid with [23]. The computational time was as much as 4 minutes, too slow for real-time application, even with a coarsely sampled grid of 30 points in each dimension and with low numeric accuracy. When the grid is increased to 45 points in each dimension and with high numeric accuracy, the computation time jumps to an hour.

Recent research [12], [10] has discovered numerical solutions based on the generalized Hopf formula that do not require a grid and can be used to efficiently compute solutions to a certain class of Hamilton Jacobi equations that arise in linear control theory and differential games. This readily allows the generalization with pursuit-evasion to collaborative guidance of multiple pursuing vehicles.

This letter presents new methods developed for multi-vehicle collaborative pursuit guidance of maneuvering targets. A joint system state space representing the kinematics of all pursuing vehicles relative to the target was constructed, the dimension of which makes it infeasible for traditional grid-based methods. This high-dimensional problem was then efficiently solved using the generalized Hopf formula, and included the constraint of time-varying bounds on the magnitude of available vehicle control, while ensuring intercept when starting within the reachable set. The rest of the paper is organized as follows. We derive the models used in the study in Sec. II, followed the presentation of efficient solution techniques that employ the generalized Hopf formula to solve the Hamilton-Jacobi equations for optimal control and differential games in Sec. III. The application of these methods to collaborative guidance is given in Sec. IV, followed by results on a planar, multiple vehicle pursuit-evasion game in Sec. V.

II. PURSUIT-EVASION MODEL

First consider the pursuit-evasion game with only a single pursuer. We construct a state space representation of the
position and orientation of the pursuer relative to the evader, with geometry shown in Fig. 1. With \( x = [\delta x, \delta y, \delta \theta] \), the relative system becomes

\[
\frac{dx}{dt}(t) = \begin{bmatrix} V_p \cos(\delta \theta) - V_e + \frac{\delta v_e}{V_e} \delta x \\ V_p \sin(\delta \theta) - \frac{\delta x}{V_p} - \frac{\delta v_e}{V_e} \delta y \end{bmatrix}, \tag{1}
\]

with \( V_p \) and \( V_e \) representing the forward speed of the pursuer and evader, respectively. The terms \( a_p \) and \( a_e \) are the lateral acceleration inputs to the system for each vehicle. These accelerations are limited by the physical maneuver capabilities of the vehicles. This system is based on the relative state [25] of two modified Dubin’s car [31, 13] models, with acceleration instead of the more common turn rate input. Additionally, we constructed this system to be evader centric, allowing for the addition of multiple pursuers. Introducing the new state vector \( x = [\delta x, \delta y, \delta v_x, \delta v_y]^T \), we proceed to linearize the system (1) with

\[
\dot{x}(t) = \begin{bmatrix} 0 & I_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ \pm 1 \end{bmatrix} a_p + \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} a_e, \tag{2}
\]

with the \( \pm \) sign needed depending on whether its tail-chase (+) or head-on (−) engagement. The linearization at first glance may seem extreme, but this linearization strategy is used when deriving proportional navigation, or its variants such as augmented proportional guidance and extended proportional guidance, using linear quadratic control techniques [28]. The controls for the pursuer are constrained to the set \( \mathcal{A}_p = \{ a_p : \|Q_p^{-1}(t) a_p\|_\infty \leq 1 \} \) and the controls for the evader are constrained to the set \( \mathcal{A}_e = \{ a_e : \|Q_e^{-1} a_e\|_\infty \leq 1 \} \). The infinity norm with diagonal matrix \( Q \), scales the control limit independently in orthogonal directions. \( Q_p \) is a function of time since some systems have control bounds that vary with time, and is needed to model aerodynamic control surfaces on decelerating vehicles. Both controls are considered symmetric (centered at zero) for this paper and all experiments, though it can be easily generalized with an offset term in the Hamiltonian.

We represent the capture set, \( \Omega \), as an ellipsoid

\[
\Omega = \{ x : \langle x, W^{-1} x \rangle \leq 1 \} , \tag{3}
\]

where \( W \) is the ellipsoid shape matrix. The elements of \( W \) are selected such that the pursuing vehicle must be within a distance \( r \)

\[
\| [\delta x \, \delta y] \| \leq r,
\]

and the velocity at intercept is within some large bound \( \epsilon \) (we don’t care what the velocity was at capture, just as long as capture has occurred). This gives

\[
W = \begin{bmatrix} r^2 & 0 & \cdots & 0 \\ 0 & r^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \epsilon^2 \end{bmatrix}.
\]

### A. Collaborative Pursuit-Evasion

For a multi-vehicle problem with \( k \) pursuers against a single evader, the joint state space with state vector \( \chi \in \mathbb{R}^{4 \times k} \) can be constructed as follows

\[
\chi = \begin{bmatrix} \chi_1 \\ \chi_2 \\ \vdots \\ \chi_k \end{bmatrix} = \begin{bmatrix} A & \cdots & 0 \\ B_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ B_k & \cdots & 0 \end{bmatrix} \begin{bmatrix} a_{p1} \\ a_{p2} \\ \vdots \\ a_{pk} \end{bmatrix}, \tag{4}
\]

Collaborative guidance is induced by noticing that capture can happen by any single vehicle of the \( k \) vehicles in the system. The capture set for the \( i \)-th vehicle in the joint system (4) is denoted as

\[
\Omega_i = \{ \chi : \langle \chi, W_i^{-1} \chi \rangle \leq 1 \},
\]

with the shape matrix defined as the block diagonal matrix

\[
W_i = \begin{bmatrix} e^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^2 \end{bmatrix}.
\]
This implies that the capture set for the joint system is
\[ \Omega = \cup_i \Omega_i. \] (6)
It is worth noting that even the collaborative guidance problem consisting of just 2 pursuers on a single target will have state dimension of 8, far too large to generate a numerically tractable solution using grid-based level set methods [26, 24].

III. Hamiltonian Jacobi Equations with Bounded Control

To compute optimal guidance, we use the generalized Hopf formula [12], [18], [21]. Consider system dynamics represented as
\[ \begin{cases} \frac{dx}{dt} (t) = f (u (t)) \quad \text{in } (t, T), \\ x (t) = x, \end{cases} \] (7)
where \( x \) is the system state, \( u (t) \in C \) is the control input, constrained to lie in the convex admissible control set \( C \). We consider a cost functional for a given initial time \( t \), and terminal time \( T \)
\[ K (x, t, u) = \int_t^T L (u (t)) \, dt + J (x (T)), \] (8)
where \( x (T) \) is the solution of (7) at terminal time, \( T \). We assume that the terminal cost function \( J : \mathbb{R}^n \to \mathbb{R} \) is convex. The function \( L : \mathbb{R}^n \to \mathbb{R} \cup \{ +\infty \} \) is the running cost, and is assumed proper, lower semicontinuous, convex, and 1-coercive. The value function \( v : \mathbb{R}^n \times (0, T] \to \mathbb{R} \) is defined as the minimum cost, \( K \), among all admissible controls for a given state \( x \), and time \( t < T \) with
\[ v (x, t) = \inf_{u \in C} K (x, t, u). \] (9)
The value function in (9) satisfies the dynamic programming principle [7], [12] and also satisfies the following initial value Hamilton-Jacobi (HJ) equation by defining the function \( \varphi : \mathbb{R}^n \times \to \mathbb{R} \) as \( \varphi (x, t) = v (x, T - t) \), with \( \varphi \) being the viscosity solution of
\[ \begin{cases} \frac{\partial \varphi}{\partial t} (x, t) + H (\nabla_x \varphi (x, t)) = 0 \quad \text{in } \mathbb{R}^n \times (0, +\infty), \\ \varphi (x, 0) = J (x) \quad \forall x \in \mathbb{R}^n, \end{cases} \] (10)
where the Hamiltonian \( H : \mathbb{R}^n \to \mathbb{R} \cup \{ +\infty \} \) is defined by
\[ H (p) = \sup_{c \in \mathbb{R}^m} \{ -f (c), p \} - L (c). \] (11)
To apply the constraint that the control must bounded, we introduce the following running cost \( L = I_C \), where
\[ I_C = \begin{cases} 0 & \text{if } c \in C \\ +\infty & \text{otherwise,} \end{cases} \]
is the indicator function for the set \( C \). This reduces the Hamiltonian to
\[ H (p) = \max_{c \in \mathbb{C}} (-f (c), p). \]
Solving the HJ equation (10) describes how the value function evolves with time at any point in the state space and from this, optimal control policies can be found.

A. Viscosity Solutions with the Hopf Formula

It was shown in [12] that an exact, point-wise viscosity solution to (10) can be found using the Hopf formula [18]. Moreover, since no discretization grid is constructed, the formula can provide a numerical method that is efficient even when the state space is high-dimensional. The value function can be found with the Hopf formula
\[ \varphi (x, t) = -\min_{p \in \mathbb{R}^n} \left\{ J^* (p) + tH (p) - \langle x, p \rangle \right\}, \] (12)
where the Fenchel-Legendre transform \( g^* : \mathbb{R}^n \to \mathbb{R} \) of a convex, proper, lower semicontinuous function \( g : \mathbb{R}^n \to \mathbb{R} \cup \{ +\infty \} \) is defined by [14]
\[ g^* (p) = \sup_{x \in \mathbb{R}^n} \{ \langle p, x \rangle - g (x) \}. \] (13)
Notice that the Hopf formula in (12) can be equivalently written [21, 18] as
\[ \varphi (x, t) = \sup_{p \in \mathbb{R}^n} \inf_{y \in \mathbb{R}^n} \left\{ J (y) + \langle p, x - y \rangle - tH (p) \right\}. \] (14)
Following the basic definition of the Fenchel-Legendre transform, (13) can be written [12] as
\[ \varphi (x, t) = (J^* + tH)^* (x). \]
This shows that value function is itself a Fenchel-Legendre transform. It follows from a well known property of the Fenchel-Legendre transform [11] that the unique minimizer of (12) is the gradient of the value function
\[ \nabla_x \varphi (x, t) = \arg \min_{p \in \mathbb{R}^n} \left\{ J^* (p) + tH (p) - \langle x, p \rangle \right\}, \]
provided the gradient exists. So by solving for the value function using (12), we automatically solve for the gradient. Note that no numeric approximation schemes are introduced to compute the gradient or the associated value function.

B. General Linear Models

Now consider the following linear state space model
\[ \frac{dx}{dt} (t) = Ax (t) + Bu (t), \] (15)
with state vector \( x \in \mathbb{R}^n \), and control input \( u \in C \subset \mathbb{R}^m \). We can make a change of variables
\[ z (t) = e^{-tA} x (t), \] (16)
which results in the following system
\[ \frac{dz}{dt} (t) = e^{-tA} BU (t), \] (17)
with terminal cost function now defined in \( z \) with \( \varphi (z, 0) = J_z (z) = J_z (e^{tA} z) \), which depends on terminal time, \( t \). Notice that the system is of the form presented in (10) with \( f (u (t)) = e^{-tA} B (t) u (t) \), with the exception that the system is now time-varying. It was shown by Lions in [21] that the Hopf formula in (12) can be generalized for a time-varying Hamiltonian to solve for the value function of the system in (17) with
\[ \varphi (z, t) = -\min_{p \in \mathbb{R}^n} \left\{ J_z^* (p, t) + \int_0^t H (p, s) \, ds - \langle z, p \rangle \right\}, \] (18)
with the time-varying Hamiltonian defined as
\[ H(p, t) = \max_{c \in \mathcal{C}} \left< -e^{-(T-t)A} B(T-t) c, p \right>. \]

The change of variable to \( T - t \) is required for time since the problem was converted to an initial value formulation from a terminal value formulation in \([10]\).

C. Linear Differential Games

Now consider the system
\[
\frac{dx}{dt}(t) = Ax(t) + B(t) u(t) + D(t) w(t),
\]
which is equal to \([15]\) with an extra term, \( D(t) w(t) \), added. We assume that the additional control input \( w(t) \) is adversarial and bounded by \( w(t) \in \mathcal{D} \subseteq \mathbb{R}^l \). The cost functional becomes
\[
G(x, t, u, w) = \int_t^T L(u(t), w(t)) dt + J(x(T)),
\]
where \( x(T) \) is the solution of \([19]\) at terminal time, \( T \). We assume that the goal of the adversarial control input \( w(t) \) is to increase the cost functional \([20]\), in direct contradiction with the input \( u(t) \), which we are designing in an attempt to minimize the cost. This system forms a differential game \([20]\), and has a corresponding lower value function
\[
V(x, t) = \inf_{u \in \mathcal{C}} \sup_{w \in \mathcal{D}} G(x, t, u, w),
\]
and upper value function
\[
U(x, t) = \sup_{u \in \mathcal{D}} \inf_{w \in \mathcal{C}} G(x, t, u, w).
\]

As derived in \([13]\), the upper and lower value functions are viscosity solutions of possibly non convex HJ equation. We can define the following upper and lower Hamiltonians as
\[
H^+(p, t) = \sup_{c \in \mathbb{R}^m} \inf_{d \in \mathbb{R}^l} \{ -f(c, d), p \} - L(c, d),
\]
\[
H^-(p, t) = \inf_{d \in \mathbb{R}^l} \sup_{c \in \mathbb{R}^m} \{ -f(c, d), p \} - L(c, d).
\]
The running cost becomes
\[
L = \mathcal{I}_c - \mathcal{I}_d.
\]
If the Hamiltonians \( H^+ \) and \( H^- \) coincide, then from \([15]\)
\[
H^+ = H^- = H^\pm \implies U = V.
\]
We can apply the same change of variables from \([10]\) to get
\[
\frac{dz}{dt}(t) = e^{-tA} B(t) u(t) + e^{-tA} D(t) w(t),
\]
and then we can solve for the value function \( \varphi(z, t) = U(z, t) = V(z, t) \) with the generalized Hopf formula
\[
\varphi(z, t) = -\min_{p \in \mathbb{R}^n} \left\{ J^*_p(p, t) + \int_t^T H^\pm(p, s) ds - \langle z, p \rangle \right\},
\]
with the time-varying, non convex Hamiltonian given by
\[
H^\pm(p, t) = \max_{c \in \mathcal{C}} \left< -e^{-(T-t)A} B(T-t) c, p \right> - \max_{d \in \mathcal{D}} \left< -e^{-(T-t)A} D(T-t) d, p \right>.
\]

IV. Time-Optimal Control with the Hopf Formula

Following the methods presented above in \([22]\), we have the transformed system \([11]\) as
\[
\frac{dz}{dt}(t) = e^{-tA} B_p(t) + e^{-tA} D_{ae}(t),
\]
and the Hamiltonian is the dual norm of the control set
\[
H(p, t) = \left\| -Q_p R(T-t) B_v e^{-A^\dagger} - P \right\|_1 - \left\| -Q_e D_v e^{-A} p \right\|_1.
\]
We choose a terminal cost function \( J(z) \) such that
\[
\begin{cases}
J(z) < 0 & \text{for any } z \in \text{int } \Omega, \\
J(z) > 0 & \text{for any } z \in (\mathbb{R}^n \setminus \Omega), \\
J(z) = 0 & \text{for any } z \in (\Omega \setminus \text{int } \Omega),
\end{cases}
\]
where \( \text{int } \Omega \) denotes the interior of \( \Omega \). The intuition behind defining the terminal cost function this way is simple. If the value function \( \varphi(z_0, T) < 0 \) for some \( z_0 \) and \( T \), then there exists a control \( u(t) \) that drives the state from the initial condition at \( z_0 \), to the final state, \( z(T) \) inside the set \( \Omega \). The smallest value of time \( T \), such that \( \varphi(z_0, T) = 0 \) is the minimum time to reach the set \( \Omega \), starting at state \( z_0 \). The control associated with the minimum time to reach is the time-optimal control. The ellipsoid terminal set defined in \([6]\) results in a quadratic terminal cost function
\[
J_x(x) = \langle x, W^{-1} x \rangle - 1,
\]
After variable substitution the cost function becomes
\[
J_z(z, t) = \langle z, V(t) z \rangle - 1,
\]
with \( V(t) = e^{tA^\dagger} W^{-1} e^{tA} \). Following the property that the Fenchel-Legendre transform of a norm function is the dual norm \([6]\), we have
\[
J_z^*(p, t) = 1 + \frac{1}{4} \langle p, V(t)^{-1} p \rangle.
\]
The generalized Hopf formula requires the integration of the Hamiltonian which is approximated by Riemann sum quadrature \([3]\) with step size \( h \)
\[
\int_0^T H(p, s) ds \approx h \sum_{s_k \in S} H(p, s_k),
\]
where \( S \) denotes the set of discrete time samples. Rectangular quadrature with fixed step size \( h \) was used to pre-compute the time samples \( s_k \) from time \( 0 \) to \( T \), which requires only a simple sum at run time to evaluate the integral. We can approximate the matrix exponential terms efficiently, with bounded error, using \([2]\).

A. Optimization

To solve the Hopf formula in \([13]\), we are performing an unconstrained minimization problem. This objective function is nonsmooth, due to the presence of the 1-norm in the integral of the Hamiltonian. Nonsmooth unconstrained minimization problems can be solved in a variety of ways. However, because we can explicitly derive the gradient and the Hessian, this
The capture radius is \[ r = \sqrt{\left( \sum_{i=1}^{n} (p_i(t) - z_i)^2 \right)} \leq 19 \text{ m} \leq 533 \text{ m} < r \leq 2 \text{ km} \].

To find the optimal control to the desired convex terminal set \( \Omega \), we proceed by solving for the \( T^* \), the minimum time to reach the boundary of the set \( \Omega \). This is solved numerically with

\[
T^* = \arg \min_{t \in \mathbb{C}} \varphi(z_0, t).
\]

If the minimum time to reach \( T^* \) is greater than total available time \( T \), then the set \( \Omega \) is not reachable in time \( T \). The optimal control, \( u^* \), can be computed at \( z_0 \) by noticing the relation

\[
\max_{c \in \mathbb{C}} \langle c, \nabla_z \varphi(z_0, T^*) \rangle = \langle u^*, \nabla_z \varphi(z_0, T^*) \rangle,
\]

whenever \( \varphi(z, t) \) is differentiable. Which gives the optimal control as

\[
\nabla_p H(\nabla_z \varphi(z_0, T^*), T^*) = e^{-t \hat{A}_{E}} \hat{B}(t) a_p^* + e^{-t \hat{A}_{D}} \hat{D}(t) a_e^*.
\]

1) Terminal Set: Let \( J_i \) represent terminal cost function of vehicle \( i \) with shape matrix \( W_i \), then the terminal cost function of the collaborative system is

\[
J(z, t) = \min_{i=1, \ldots, k} J_i(z, t).
\]

Darbon and Osher [12] showed that max/min-plus algebra [11], [16], [22] can be used to generalize the Hopf formula to solve for non-convex initial data that can be formed as the union of convex functions as in [40]. We solve the initial value problem

\[
\begin{cases}
\frac{\partial \varphi_i(z, t)}{\partial t} + H(\nabla_z \varphi_i(z, t)) = 0 & \text{in } \mathbb{R}^n \times (0, +\infty), \\
\varphi_i(z, 0) = \min_{i=1, \ldots, k} J_i(z) & \forall z \in \mathbb{R}^n,
\end{cases}
\]

by taking the pointwise minimum over the \( k \) solutions \( \phi_i(z, t) \), each of which has convex initial data with the generalized Hopf formula.

\[
\nabla_p \varphi(z, t) = \frac{V(t)^{-1}}{2} + \text{diracs}.
\]

To find the optimal control to the desired convex terminal set \( \Omega \), we proceed by solving for the \( T^* \), the minimum time to reach the boundary of the set \( \Omega \). This is solved numerically with

\[
T^* = \arg \min_{t \in \mathbb{C}} \varphi(z_0, t).
\]

If the minimum time to reach \( T^* \) is greater than total available time \( T \), then the set \( \Omega \) is not reachable in time \( T \). The optimal control, \( u^* \), can be computed at \( z_0 \) by noticing the relation

\[
\max_{c \in \mathbb{C}} \langle c, \nabla_z \varphi(z_0, T^*) \rangle = \langle u^*, \nabla_z \varphi(z_0, T^*) \rangle,
\]

whenever \( \varphi(z, t) \) is differentiable. Which gives the optimal control as

\[
\nabla_p H(\nabla_z \varphi(z_0, T^*), T^*) = e^{-t \hat{A}_{E}} \hat{B}(t) a_p^* + e^{-t \hat{A}_{D}} \hat{D}(t) a_e^*.
\]

V. RESULTS

The above control solution has been integrated into a closed loop 2-on-1 pursuit-evasion 3 degree of freedom (3DOF) simulation using MATLAB R2016a and Simulink at 120Hz with Euler integration (see next paragraph). This included using a third order autopilot for each pursuer, and using the gradient of the value function to find optimal evader control. Preliminary results solved for the optimal control on average 40 – 83 ms on a 3 GHz Intel Core i7 950. Performance could be improved further by computing guidance at a slower rate than the engagement dynamics and/or migrating to a lower level language.

As a post-process, the evader’s inertial state is found by solving the modified Dubin’s car initial-value problem [11] relative to a fixed origin with zero initial conditions and known inputs. To prevent numeric error accumulation, the initial value problem must be solved with the same information and method as the closed loop simulation. Without saving the intermediate evaluations, Euler’s method is the only solver that satisfies this. Adding the evader’s inertial state to the vehicle’s relative state and correcting for the induced rotational motion provides the vehicle’s inertial state.

The first example uses a simple geometry in the tail-chase scenario and the engagement trajectory is shown in Figure 2. The capture radius is \( r = 3m \), evader control is limited to \( ||a_e|| \leq 10 m/s^2 \), and both pursuers have control bounds that decrease in time with

\[
||a_p|| \leq \frac{(t - 40)^2}{40} m/s^2,
\]

when \( 0 \leq t \leq 40 \), and 0 otherwise. The evader is assumed to travel at speed \( V_e = 50 m/s \) and the pursuers at speed 255.225 m/s (0.75 Mach). Both pursuing vehicles, initially launched at 4000m from the evader, are simultaneously traveling directly at the evader. Notice that both pursuers separate as to surround and contain the evader. The miss distance was 0.879m < \( r = 3m \) and time to intercept was 19.533 seconds.
The second example utilized a similar engagement with head-on aspect (more dynamic than the tail-chase) configuration. The engagement parameters are the same as example 1, but a 6000m initial separation. Figure 3 shows the flyout paths. With both vehicles launched simultaneously, with a head-on aspect and at a range of 6000m, the miss distance was 2.54m and time to intercept was 20.158 seconds. Note that the initial conditions thus far were accurate to the linearization assumption and the initial state can be such that the linearization error in $\Omega$ is large. In this case, the solution of the zero level set time may be smaller than the available flight time $T$. This indicates the set $\Omega$ is not reachable (due to the linearization error) and in our simulations reverts to proportional navigation (PN) until the set $\Omega$ is considered reachable. This can easily be countered by increasing the control bound of the evader to account for linearization error.

VI. CONCLUSIONS AND FUTURE WORK

The generalized Hopf formula provide interesting and powerful new capabilities for solving high-dimensional optimal control and differential games, such as the pursuit-evasion guidance presented here. Additionally, the above work can be used for evasion strategies that could be of interest for collision avoidance problems. Future work will focus on extending the generalized Hopf formula for certain classes of non-linear systems, such as feedback linearizable systems [30], and apply splitting algorithms [12, 17, 8] for efficient optimization when the gradient and Hessian is not explicitly known.

REFERENCES

[1] M. Akian, R. Bapat, and S. Gaubert. Max-plus algebra. Handbook of Linear Algebra (Discrete Mathematics and its Applications), 39:10–14, 2006.
[2] A. H. Al-Mohy and N. J. Higham. Computing the action of the matrix exponential, with an application to exponential integrators. SIAM Journal on Scientific Computing, 33(2):488–511, 2011.
[3] H. Anton, S. Davis, and I. Bivens. Calculus: A New Horizon. Wiley New York, 1999.
[4] R. E. Bellman. Dynamic Programming, volume 1. Princeton University Press, 1957.
[5] R. E. Bellman. Adaptive Control Processes: A Guided Tour. Princeton University Press, 2015.
[6] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.
[7] A. R. Bravly and Y.-C. Ho. Applied Optimal Control: Optimization, Estimation and Control. CRC Press, 1975.
[8] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision, 40(1):120–145, 2011.
[9] S. C. Chapra and R. P. Canale. Numerical Methods for Engineers, volume 2. McGraw-Hill New York, 1998.
[10] Y. T. Chow, J. Darbon, S. Osher, and W. Yin. Algorithm for overcoming the curse of dimensionality for certain non-convex Hamilton-Jacobi equations, projections and differential games. Technical report, UCLA Tech. Rep. CAM 16-27, University of California, Los Angeles, Department of Mathematics, Group in Computational Applied Mathematics, 2016.
[11] J. Darbon. On convex finite-dimensional variational methods in imaging sciences and Hamilton-Jacobi equations. SIAM Journal on Imaging Sciences, 8(4):2268–2293, 2015.
[12] J. Darbon and S. Osher. Algorithms for overcoming the curse of dimensionality for certain Hamilton-Jacobi equations arising in control theory and elsewhere. Research in the Mathematical Sciences, 3(1):19, 2016.
[13] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. American Journal of Mathematics, 79(3):497–516, 1957.
[14] I. Ekeland and R. Temam. Convex Analysis and Variational Problems. SIAM, 1999.
[15] L. C. Evans and P. E. Souganidis. Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations. Technical report, DTIC Document, 1983.
[16] W. H. Fleming. Deterministic nonlinear filtering. Annali della Scuola Normale Superiore di Pisa-Classe di Scienze, 25(3-4):435–454, 1997.
[17] T. Goldstein and S. Osher. Split Bregman method for ${L}_1$-regularized problems. SIAM Journal on Imaging Sciences, 2(2):323–343, 2009.
[18] E. Hopf. Generalized solutions of non-linear equations of first order. Journal of Mathematics and Mechanics, 14:951–973, 1965.
[19] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin. A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag. In 2011 IEEE International Conference on Robotics and Automation (ICRA), pages 1451–1456. IEEE, 2011.
[20] R. Isaacs. Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization. Courier Corporation, 1999.
[21] P. L. Lions and J.-C. Rochet. Hopf formula and multitime Hamilton-Jacobi equations. Proceedings of the American Mathematical Society, 96(1):79–84, 1986.
[22] W. M. McEneaney. Max-Plus Methods for Nonlinear Control and Estimation. Springer Science & Business Media, 2006.
[23] I. Mitchell. A toolbox of level set methods. Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, Canada, http://www.cs.ubc.ca/~mitchell/Toolbox/LS/Toolbox/LS.pdf. Tech. Rep. TR-2004-09, 2004.
[24] I. Mitchell. The flexible, extensible and efficient toolbox of level set methods. Journal of Scientific Computing, 35(2):300–329, 2008.
[25] I. Mitchell, A. M. Bayen, and C. J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. IEEE Transactions on Automatic Control, 50(7):947–957, 2005.
[26] S. Osher and F. Fedkiw. Level Set Methods and Dynamic Implicit Surfaces, volume 153. Springer Science & Business Media, 2006.
[27] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd. Basic principles of homing guidance. Johns Hopkins APL Technical Digest, 29(1):25–41, 2010.
[28] N. F. Palumbo, R. A. Blauwkamp, and J. M. Lloyd. Modern homing missile guidance theory and techniques. Johns Hopkins APL Technical Digest, 29(1):42–59, 2010.
[29] S. Pan, H. Huang, J. Ding, W. Zhang, and C. J. Tomlin. Pursuit, evasion and defense in the plane. In American Control Conference (ACC), pages 4167–4173. IEEE, 2012.
[30] J.-J. Slotine and W. Li. Applied Nonlinear Control, volume 199. Prentice-Hall Englewood Cliffs, NJ, 1991.
[31] M. M. Stipanović, G. Inalhan, R. Teo, and C. J. Tomlin. Decentralized overlapping control of a formation of unmanned aerial vehicles. Automatica, 40(8):1285–1296, 2004.