aerospace

MDPI

Article

A Geometrical, Reachable Set Approach for Constrained Pursuit–Evasion Games with Multiple Pursuers and Evaders

Olli Jansson * and Matthew W. Harris

Mechanical and Aerospace Engineering Department, Utah State University, Logan, UT 84322, USA
* Correspondence: olli.jansson@usu.edu

Abstract: This paper presents a solution strategy for deterministic time-optimal pursuit–evasion games with linear state constraints, convex control constraints, and linear dynamics that is consistent with linearized relative orbital motion models such as the Clohessy–Wiltshire equations and relative orbital elements. The strategy first generates polytopic inner approximations of the players' reachable sets by solving a sequence of convex programs. A bisection method then computes the optimal termination time, which is the least time at which a set containment condition is satisfied. The pursuit–evasion games considered are games with (1) a single pursuer and single evader, (2) multiple pursuers and a single evader, and (3) a single pursuer and multiple evaders. Compared to variational methods, this reachable set strategy leads to a tractable formulation even when there are state and control constraints. The efficacy of the strategy is demonstrated in three numerical simulations for a constellation of satellites in close proximity in low earth orbit.

Keywords: convex optimization; reachable set; differential game; pursuit–evasion game

Citation: Jansson, O.; Harris, M.W. A Geometrical, Reachable Set Approach for Constrained Pursuit–Evasion Games with Multiple Pursuers and Evaders. *Aerospace* **2023**, *10*, 477. https://doi.org/10.3390/aerospace10050477

Academic Editor: Fanghua Jiang

Received: 3 April 2023
Revised: 11 May 2023
Accepted: 15 May 2023
Published: 18 May 2023

1. Introduction

This paper studies time-optimal pursuit–evasion games with linear dynamics and convex state and control constraints. Cases with a single pursuer and single evader, multiple pursuers and a single evader, as well as a single pursuer and multiple evaders are considered. The paper takes a geometrical approach in solving these games by modeling them as reachable set problems as first introduced in [1]. By using modern, convex optimization techniques for the reachable set construction of constrained systems, it is now possible to solve constrained, multiplayer games not considered in [1].

Pursuit–evasion games commonly arise in aerospace engineering as well as in economics. The history of differential pursuit–evasion games begins with the seminal work by Isaacs [2]. His work describes several applied problems and their differential pursuit–evasion formulations. An overview of the development of differential pursuit–evasion games is given in [3], which covers seminal work conducted in the 1970s as well as current state-of-the-art work.

Aerial warfare is one of the typical problems that can be modeled as a pursuit–evasion game [4,5]. Multiple assumptions are often made to simplify dynamical complexity. Some common assumptions restrict the motion to be planar, model the aircraft as a point mass, linearize the dynamics, and require an instantaneous control response [6]. Examples of different aerial warfare problems modeled as pursuit–evasion games consist of missile versus aircraft [4,6–8], aircraft versus aircraft interception [9,10], and fighter maneuvering [11].

Pursuit–evasion games have also been studied in the context of astrodynamics, particularly collision avoidance and interception [12,13]. Rather than use a nonlinear two-body formulation, linearization and the use of the Clohessy–Wiltshire (CW) equations simplifies the problem so that minimax formulations lead to open-loop and closed-loop control strategies based on kriging [14]. Pursuit–evasion strategies have been used for the tracking of space objects and selection of sensor management strategies [15]. A hybrid global–local

technique has been developed for a two-phase (long distance and short distance) game [16]. The two-phase approach has been extended in [17] using reinforcement learning under assumptions of impulse control, incomplete information, and a quadratic objective.

A pursuit–evasion game can be seen as an optimal control problem where the pursuer(s) and evader(s) have different objectives. The addition of multiple agents with opposite objectives generally makes the optimal control problem more difficult to solve than those with an individual pursuer or evader would be. It should be noted that sometimes the single-agent optimal control problems can be complicated to solve as is. A technique utilizing calculus of variations may be used to solve differential games. This technique requires setting an optimal control problem for all the players with the underlying necessary conditions and then finding a solution simultaneously for all the players. Linear–quadratic as well as time-optimal scalar games were solved using this methodology in [18]. A case of one of the players not playing the game optimally based on the variational approach is also included in [18] as part of a short discussion on games with stochastic behavior.

Because of their complexity, it is often difficult to solve differential games using variational theory. This fact is aptly summarized in [17]: "However, for the space [pursuit-evasion] game, it is a challenge to solve the transformed [two-point boundary value problem] due to its high dimensionality and strong nonlinearity." This is particularly true when there are pointwise constraints on the states and controls. The pointwise constraints introduce switches as trajectories enter and leave the control boundaries, which may violate the smoothness assumptions required for the Newton method. Consequently, direct methods [19,20] and semidirect methods have been introduced to leverage finite-dimensional optimization packages. These methods involve the use of genetic algorithms and neural networks [8,11,21,22].

In contrast to the optimization-based approaches, Mizukami [1] has shown that a two-player time-optimal pursuit–evasion game terminates when the evader's reachable set is contained in the pursuer's reachable set, which motivates a geometric approach to solving the game based on reachable set computation. More precisely, the termination point in the state space lies on the boundary of the players' reachable sets. For problems with linear dynamics and integral-constrained (rather than pointwise-constrained) controls, analytical solutions can be derived for the termination time and optimal trajectories. Motivated by this work, reachable set analyses have been used in dynamic flowfields [23], coordinate control [24], missile/sensor trade studies [25], and other game scenarios [26–28].

In the CW setting and with control magnitude constraints, it is possible to approximate reachable sets with a sequence of analytical computations [29]. The presence of state constraints and other types of control constraints complicates the process of computing reachable sets, and in general, numerical methods are utilized to approximate them. Multiple numerical algorithms to calculate a reachable set for control affine systems are provided in [30]. An algorithm for the computation of reachable sets for linear systems with bounded inputs, which approaches the problem by finding inner and outer approximations, is introduced in [31]. Optimal control has also been used to calculate reachable sets [32–35]. A polytopic approximation of the actual reachable set is computed. In [34], a single semidefinite program (SDP) and multiple sequential second-order cone programs (SOCPs) are solved to find an inner approximation of the reachable set for linear dynamics and convex state and control constraints. An outer approximation is added in [33] as a heuristic to determine the precision of the approximation. Nonconvex control constraints are considered in [35]. The approach from [34] is used as a basis for solving multiplayer games in this paper.

The primary contribution of this work is the construction of reachable set approximations to solve constrained pursuit–evasion games. The inclusion of pointwise, convex constraints on the state and control make the variational approach difficult, while such constraints add minimal complexity to the geometric approach. The inclusion of multiple pursuers and evaders adds complexity to the problem as well, especially if variational

techniques were to be used. It is shown that with reachable set methods, constrained pursuit–evasion problems with multiple pursuers and evaders remain tractable. It is assumed that the pursuers and evaders have perfect information about the other players and all players perform optimally. Pursuit–evasion problems with (1) a single pursuer and single evader, (2) multiple pursuers and a single evader, and (3) a single pursuer and multiple evaders are considered.

The remainder of this paper is structured as follows. Section 2 describes a numerical method based in convex optimization for approximating reachable sets. The algorithm is based on the one introduced in [34]. The constrained pursuit–evasion game is introduced in Section 3 and an explanation on how reachable set theory can be used to solve it for a varying number of pursuers and evaders is explained. Section 5 applies the method to aerospace problems whose dynamics and constraints are explained in Section 4. The results are summarized and conclusions are drawn in Section 6.

2. Reachable Sets

An introduction to reachable sets is given in this section. Reachable sets for linear time-varying (LTV) dynamics with second-order cone control constraints and linear state constraints are considered. This system of interest is given as

$$\dot{x} = A(t)x + B(t)u \tag{1}$$
$$y = C(t)x \tag{2}$$
$$x \in \mathcal{X}(t) = \{x \in \mathbb{R}^n : D(t)x \leq b(t)\} \tag{3}$$
$$u \in \mathcal{U}(t) \tag{4}$$
$$x(t_0) = x_0. \tag{5}$$

The state vector is $x \in \mathbb{R}^n$, the control vector is $u \in \mathbb{R}^m$, the output vector is $y \in \mathbb{R}^p$, the system matrix is $A(t) : \mathbb{R}^n \to \mathbb{R}^n$, the control influence matrix is $B(t) : \mathbb{R}^m \to \mathbb{R}^n$, and the output matrix is $C(t) : \mathbb{R}^n \to \mathbb{R}^p$. The quantities $D(t) \in \mathbb{R}^{q \times n}$ and $b(t) \in \mathbb{R}^q$ define the linear state constraint $\mathcal{X}(t)$. The second-order cone control constraint is $\mathcal{U}(t)$. The state and control constraints limit the values that can be taken by the state and control pointwise in time. The initial state vector is x_0 at initial time t_0. A feasible control is any control satisfying (1)–(5).

Definition 1. *For the system in* (1)–(5)*, the reachable set* $\mathcal{R}(t)$ *is the set of all outputs the system can reach at time* t *from the initial state* x_0 *using feasible controls, i.e.,*

$$\mathcal{R}(t) = \{y \in \mathbb{R}^p \ : \ y = C(t)\Phi(t, t_0)x_0 + C(t)\int_{t_0}^{t} \Phi(t, \tau)B(\tau)u(\tau)\, d\tau \text{ for all feasible } u\} \tag{6}$$

where $\Phi(t_2, t_1)$ *is the state-transition matrix.*

The reachable set is the set of all points that can be reached from a given initial condition with a feasible control. The first term on the right-hand side of (6) is $C(t)\Phi(t, t_0)x_0$. Because it is independent of the control, this term is associated with the translation of the reachable set. The second term on the right-hand side is the integral term. This term depends on the control and is associated with the size and shape of the reachable set. In reachable set theory, sets that grow in size are said to be expanding. An illustration of a reachable set in two-dimensional space that is translating and growing in time is shown in Figure 1.

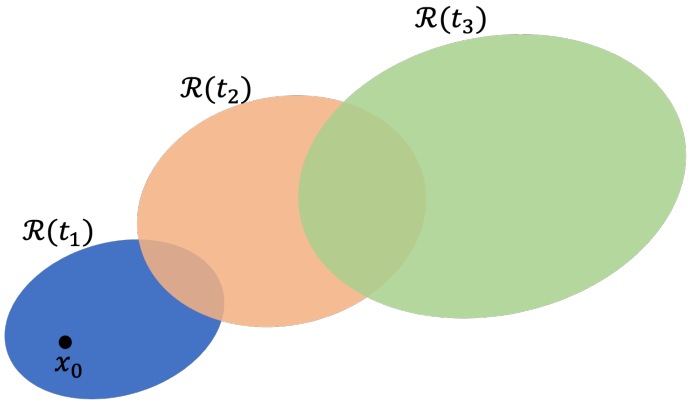

Figure 1. The initial point is labeled x_0. The reachable set at time t_1 is shown as the small blue ellipse. As time increases to t_2, the reachable set translates up and to the right as it grows. The reachable set at t_2 is shown as the orange ellipse. As time increases to t_3, the reachable set continues to translate and grow; it is shown as the green ellipse.

Reachable sets do not always expand, and the strategy presented herein has no such requirement. The simplest sufficient condition to ensure the expansion of the reachable set is time invariance [36]. For time-varying systems, a sufficient condition for expansion is properness (see Corollary 17.1 of [37]). Furthermore, because the dynamics of the system are linear and the state and control constraints are convex, the reachable set is convex [36].

2.1. Algorithm for Reachable Set Calculation

An algorithm to calculate an inner polytopic approximation of the reachable set for the system described in (1)–(5) is given next. The algorithm first solves an SDP to compute a largest possible size simplex that fits inside the reachable set. The algorithm then solves a sequence of SOCPs to compute a better inner approximation of the reachable set based on the initial simplex. A more detailed explanation of the algorithm can be found in [34].

Consider the discretized version of the continuous system given in (1)–(5). The discrete-time system is

$$x_{k+1} = \bar{A}_k x_k + \bar{B}_k u_k,\ k = 0, \ldots, N-1 \tag{7}$$
$$y_k = C_k x_k,\ k = 0, \ldots, N \tag{8}$$
$$x_k \in \mathcal{X}_k = \{x \in \mathbb{R}^n : D_k x \leq b_k\} \tag{9}$$
$$u_k \in \mathcal{U}_k \tag{10}$$
$$x_{k=0} = x_0 \tag{11}$$

where N represents the number of time intervals used in discretization. Appendix A explains how discretization is conducted in this paper to obtain the discrete, barred quantities based on their continuous counterparts and discretization time step. Discretization allows for the use of numerical optimization. The reachable set definition given in Definition 1 for the continuous system is analogous to that for the discrete system by replacing t with t_k.

2.1.1. Initial Simplex

The computation of the initial simplex is explained first. Computing the initial simplex requires solving a single SDP. Let $\mathcal{Z}$ be the set containing the $p+1$ vertices of an initial simplex in $\mathbb{R}^p$. Furthermore, define a matrix Q as

$$Q = \begin{bmatrix}(z_2 - z_1) & \ldots & (z_{p+1} - z_1)\end{bmatrix} : z_i \in \mathcal{Z}. \tag{12}$$

The volume of the simplex may then be calculated as

$$v = \frac{1}{p!}|\det Q|. \tag{13}$$

With an expression for the simplex volume, maximizing (13) gives an initial simplex with a maximum volume. This also leads to the vertices of the simplex being on the boundary of the system's reachable set:

$$z_i \in \partial\mathcal{R} \subset \mathcal{R}. \tag{14}$$

This maximum volume simplex is used as the initial simplex. The volume maximization problem can be written as the following SDP:

$$\begin{aligned} \min_{x_i, u_i, z_i} \quad & -\operatorname{logdet}(Q) \\ s.t. \quad & Q = Q^T = \begin{bmatrix}(z_2 - z_1) & \dots & (z_{p+1} - z_1)\end{bmatrix} \succeq 0 \\ & z_i = y_i(t_k), i = 1, \dots, p+1 \\ & \text{Equations (7)–(11).} \end{aligned} \tag{15}$$

The log determinant is denoted as logdet, which acts on square symmetric matrices. The initial simplex requires the calculation of $p+1$ trajectories emanating from the initial point, x_0, to $p+1$ final states. The initial simplex is constructed from the $p+1$ final state vectors which form the $z_1, \dots, z_{p+1}$ vertices of the initial simplex.

2.1.2. Growing Simplices

Following the construction of the initial simplex, improved approximations are achieved by computing additional simplices out of the open faces of the current polytopic approximation. Whereas the computation of the initial simplex required a solution of an SDP, the computation of additional simplices is achieved by solving an SOCP for each additional vertex in the polytopic approximation. This is possible because maximizing the volume of an additional simplex is analogous to maximizing the length of a vector that connects the open face to a point on the boundary of the reachable set and is orthogonal to the face. The point on the boundary of the reachable set is then added as a new vertex to the current polytopic approximation of the reachable set:

$$\begin{aligned} \min_{\alpha, \lambda, x, u} \quad & -\alpha \\ \text{s.t.} \quad & \nu = Z_i\lambda, \quad 1^\top\lambda = 1, \quad \lambda \geq 0 \\ & \nu + \alpha h_i = z \\ & z = y(t_k) \\ & \text{Equations (7)–(11).} \end{aligned} \tag{16}$$

The quantity to be maximized is the scaling factor α. The point on the open face ν is such that the minimum distance between it and the reachable set boundary is maximized. The matrix Z_i contains the p vertices corresponding to the open face i. The vector $\lambda \in \mathbb{R}^p$ identifies the point ν from the vertices in Z_i. A vector normal to the ith open face is given by h_i. A point on the boundary of the reachable set is z. Figure 2 gives a graphical representation of these variables with ν being represented as a vector from one vertex.

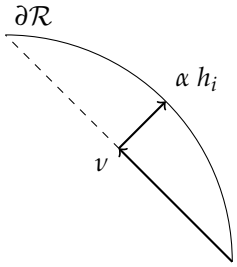

Figure 2. Illustration of variables used for polytopic approximation in (16).

Overall, the algorithm requires the solution of a single SDP defined in (15) to obtain an initial approximation of the reachable set. A tighter approximation is then reached by solving a sequence of SOCPs given in (16), with each SOCP adding a new vertex to the current polytopic approximation. The first three reachable set approximations are illustrated in Figure 3 along with the actual reachable set. Further explanation and illustration of the approximation process may be found in [34].

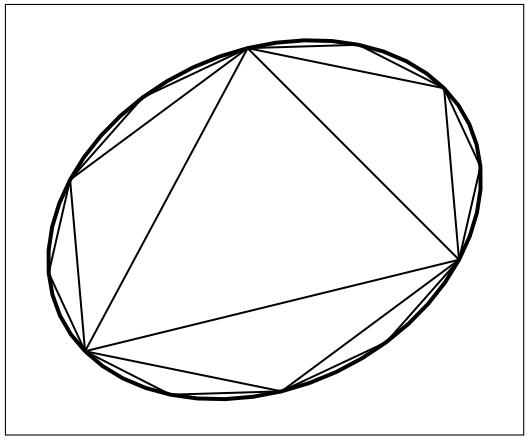

Figure 3. The initial simplex is the innermost triangle. The next two generations share vertices with the triangle and approximate the reachable set, which appears as the ellipse with a thicker line.

3. Game Theory

This section introduces the pursuit–evasion game of interest. The three pursuit–evasion games considered in this paper are games with (1) a single pursuer and single evader, (2) multiple pursuers and a single evader, and (3) a single pursuer and multiple evaders. The pursuers' objective is to minimize the capture time of all the evaders. The evaders' objective is to maximize the time it takes for the pursuers to capture all the evaders. It is assumed that a solution exists, i.e., capture occurs, such that it is a game of degree rather than a game of kind [2].

3.1. Single Pursuer and Single Evader

Consider the simplest case of a single pursuer and single evader. The game may be rewritten as two time-optimal control problems. The pursuer's problem is:

$$\begin{aligned} \min_{u_P(\cdot)} \quad & t^* \\ \text{s.t.} \quad & \dot{x}_P = A_P(t)x_P + B_P(t)u_P \\ & y_P = C_P(t)x_P \\ & x_P \in \mathcal{X}_P(t) = \{x \in \mathbb{R}^n : D_P(t)x \leq b_P(t)\} \\ & u_P \in \mathcal{U}_P(t) \\ & x_P(t_0) = x_{0,P} \\ & y_P(t^*) = y_E(t^*). \end{aligned} \tag{17}$$

The evader's problem is:

$$\begin{aligned} \max_{u_E(\cdot)} \quad & t^* \\ \text{s.t.} \quad & \dot{x}_E = A_E(t)x_E + B_E(t)u_E \\ & y_E = C_E(t)x_E \\ & x_E \in \mathcal{X}_E(t) = \{x \in \mathbb{R}^n : D_E(t)x \leq b_E(t)\} \\ & u_E \in \mathcal{U}_E(t) \\ & x_E(t_0) = x_{0,E} \\ & y_E(t^*) = y_P(t^*). \end{aligned} \tag{18}$$

The subscript P refers to the pursuer, the subscript E refers to the evader, and t^* is the termination time of the game. It is assumed that both the pursuer and evader have perfect information about the other.

It was shown by Mizukami [1] that such time-optimal games can be recast as reachable-set inclusion problems.

Theorem 1 ([1]). *If a solution to the game exists, then the game terminates in the least time such that*

$$\mathcal{R}_E(t) \subseteq \mathcal{R}_P(t). \tag{19}$$

The question of existence is not explored in this paper. It is assumed that a solution exists, and the theorem is used to find the solution by comparing the players' reachable sets. The termination time is the first time at which the evader's reachable set is contained in the pursuer's reachable set. The game is therefore reduced to a one-dimensional search for t^*. This search is performed in practice by using a bisection, which is a special case of the more generic Algorithm 1 presented in the next subsection, and the method of Section 2.1 for approximating the reachable sets. In the state space, the game terminates at a point on the boundary of both players' reachable sets, i.e., $y^* \in \partial R_E(t^*) \cap \partial R_P(t^*)$, where y^* is the capture point.

The simplest situation occurs when $C(t)\Phi(t, t_0)x_0$ is zero (so that the reachable sets do not translate) and the pursuer's control set is larger than the evader's (so that it enlarges at a faster rate). Prior to the termination time, the evader's reachable set is not contained in the pursuer's. After the termination time, the evader's reachable set is in the interior of the pursuer's. At the termination time, the evader's set is contained in the pursuer's set, and they share a boundary point. These relationships are demonstrated in Figure 4.

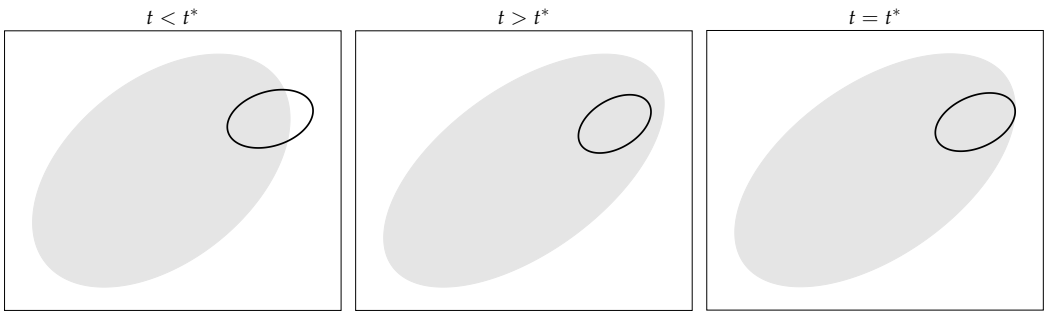

Figure 4. Pursuer's (shaded region) and evader's (solid line) reachable sets before capture ($t < t^*$), after capture ($t > t^*$), and at capture ($t = t^*$).

3.2. Multiple Pursuers and Single Evader

When there are multiple pursuers and a single evader, the game terminates when the evader's reachable set is contained in the union of the pursuers' reachable sets [23], i.e.,

$$\mathcal{R}_E(t) \subseteq \left\{ \bigcup_{i=1}^{N_P} \mathcal{R}_{P_i}(t) \right\} \tag{20}$$

where N_P is the number of pursuers. For later use, the bracketed term on the right-hand side is defined to be $\mathcal{R}_{\cup P}(t)$. If the game has only one pursuer, (20) reduces to (19) as expected. Some of the possible capture scenarios are shown in Figures 5 and 6. In Figure 5, the capture happens at the boundary of the two pursuers' and evader's sets. In Figure 6, the capture happens at the boundary of the four pursuers' sets but is in the interior of the evader's reachable set. In both cases, however, the capture happens at the first time instance when the evader's reachable set is contained in the union of the pursuers' reachable sets.

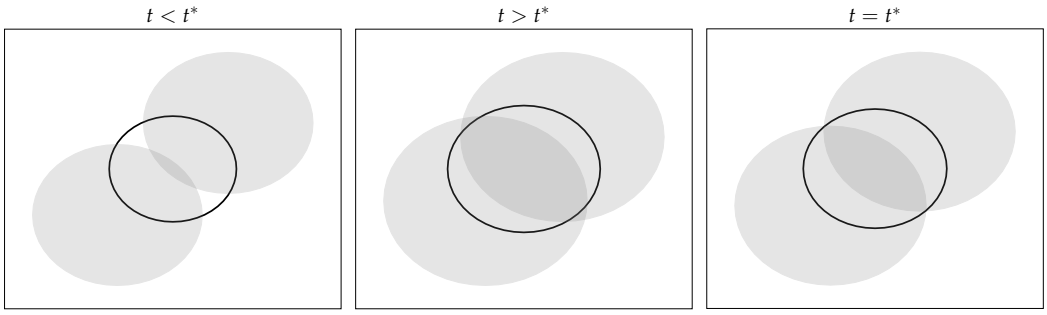

Figure 5. Pursuers' (shaded region) and evader's (solid line) reachable sets before capture, after capture, and at capture.

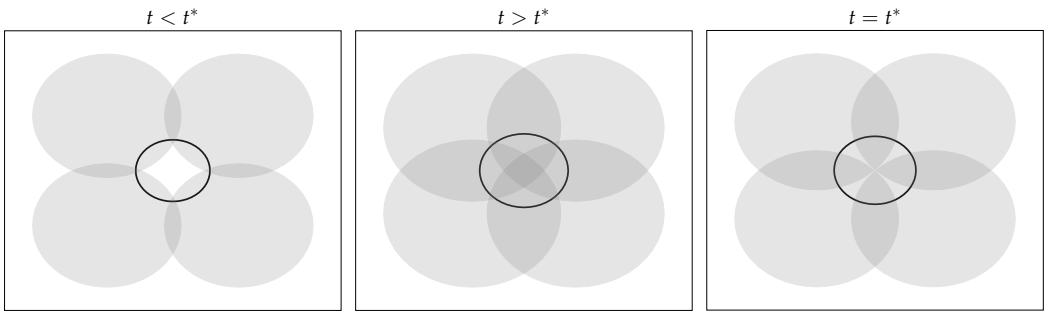

Figure 6. Pursuers' (shaded region) and evader's (solid line) reachable sets before capture, after capture, and at capture.

A line search is conducted to find the capture time. An algorithm that performs this line search is shown in Algorithm 1. The algorithm uses the method from Section 2.1 to generate the inner approximations of the pursuers' and evader's reachable sets and checks if (20) is satisfied or not. The algorithm uses a bisection method to find the capture time by checking the set inclusion and tightening the minimum and maximum allowable times.

Algorithm 1 Line Search for Capture Time in Multiple Pursuer and Single Evader Game

Input: Dynamics for evader and pursuers, initial states, ϵ, t_{min}, t_{max}
Output: t^*

Set $t_0 = (t_{min} + t_{max})/2$ and $i = 0$.
while $\|t_i - t_{i-1}\| < \epsilon$ **do**
  $i = i + 1$
  Calculate $\mathcal{R}_E(t_{i-1})$
  Calculate $\mathcal{R}_{P_1}(t_{i-1}), \ldots, \mathcal{R}_{P_{Np}}(t_{i-1})$ and $\mathcal{R}_{\cup P}(t_{i-1})$
  if $\mathcal{R}_E(t_{i-1}) \subset \mathcal{R}_{\cup P}(t_{i-1})$ **then**
    $t_i = (t_{min} + t_{i-1})/2$, $t_{max} = t_{i-1}$
  else
    $t_i = (t_{max} + t_{i-1})/2$, $t_{min} = t_{i-1}$
  end if
end while
$t^* = t_i$

3.3. Single Pursuer and Multiple Evaders

Lastly, a pursuit–evasion game with a single pursuer and multiple evaders is considered. This game is split into multiple single pursuer and single evader games where the pursuer's reachable set is reset at the capture point when a capture of one of the evaders happens. Figure 7 demonstrates the evolution of the pursuer's and evaders' reachable sets where the pursuer's reachable set is reset after capturing an evader.

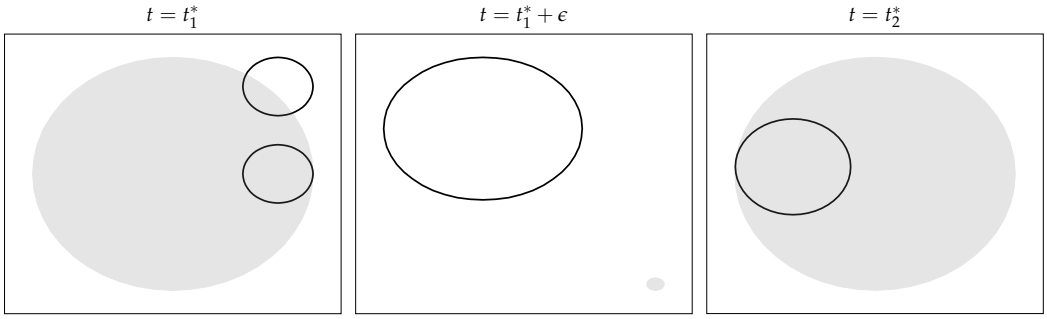

Figure 7. Pursuer's (shaded region) and evaders' (solid line) reachable sets at capture of the first evader, shortly after the capture of the first evader and at capture of the second evader. Notice that shortly after capture of the first evader, the pursuer's reachable set is reset at the capture point, whereas the second evader's set continues to grow.

To consider all the possible strategies for the pursuer to capture the evaders, it is in general required to consider $N_E!$ possibilities for capturing sequences, where N_E is the number of the evaders. This means that in a case of three evaders, the pursuer could capture the evaders in six different orders: 1,2,3; 1,3,2; 2,1,3; 2,3,1; 3,1,2; or 3,2,1. Figure 8 shows the possible capture order of the pursuer in a three evader game. t with subscripts represents the capture time of the evaders in the order of numeric values in the subscript. t^* is the termination time of the game associated with the shallowest branch, i.e., the branch offering the least time compared to all other branches. In this example, the selected order for capturing the evaders is 2,1,3.

Now consider a case where catching four evaders in the order 1,2,3,4 takes $t_{1,2,3,4}$ time units. If catching only evaders 4 and 3 in this order takes $t_{4,3}$ time units and $t_{1,2,3,4} < t_{4,3}$, then it is not optimal for the pursuer to first capture evader 4 and then evader 3. As a consequence, there is no need to further compute $t_{4,3,1,2}$ and $t_{4,3,2,1}$ since it is guaranteed that both are greater than $t_{1,2,3,4}$. This logic leads to a pruning strategy that may allow the termination time for the game to be found without enumerating all the capture strategies completely. This pruning strategy is analogous to that in branch and bound, which is used to solve integer optimization problems [38].

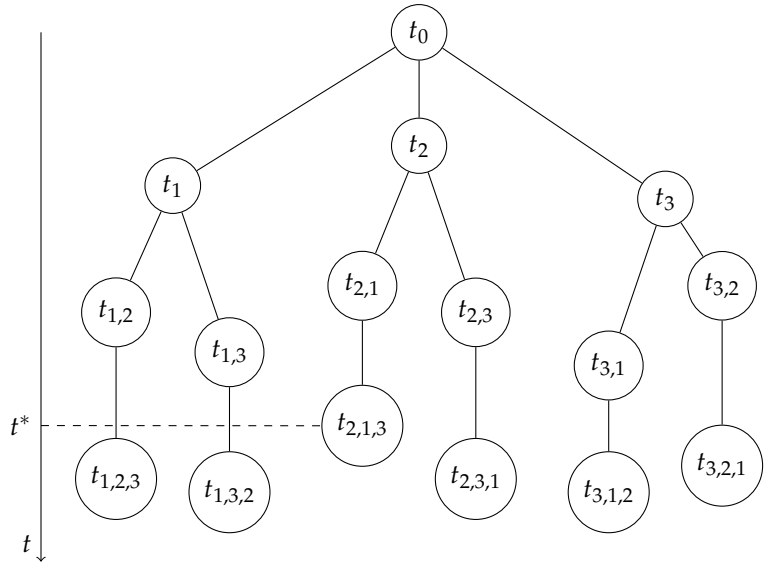

Figure 8. Tree graph showing the different possibilities (branches) on the order pursuer could capture the evaders. t^* is the termination time for the pursuer to capture all the evaders.

An algorithm to find the termination time t^* for the multiple evader and single pursuer game is provided in Algorithm 2. The algorithm begins by finding the termination time of the game by using a greedy approach when choosing the next evader to be captured. The other capturing sequences are then compared to this. If the capture time of an evader is larger than the current estimate for the termination time of the game $t^*_{current}$, that capturing sequence is excluded from further analysis and the branch is pruned. On the other hand, if the game termination time using that sequence is less than the current estimate for the game termination time, the current estimate for the game termination time is updated to that.

Algorithm 2 Termination Time in Single Pursuer and Multiple Evader Game

Input: Dynamics for evaders and pursuer, initial states
Output: t^*

Set $t^*_{current} = \infty$
for $i = 1$ **to** $N_E!$ **do**
  while Free evaders left **do**
    Find capture time, t of the next evader such that this capture order is not yet considered.
    Add the captured evader in the capture sequence.
    if $t \geq t^*_{current}$ **then**
      Set this capture sequence considered.
      break while
    end if
    Reset pursuer's reachable set at capture point.
  end while
  if $t < t^*_{current}$ **then**
    $t^*_{current} = t$
  end if
end for
$t^* = t^*_{current}$

4. Dynamics

The dynamics used for the numerical simulations in Section 5 are explained in this section. A planar constellation of satellites near a circular orbit is considered. The dynamics can be modeled linearly with the Clohessy–Wiltshire (CW) equations [39]

$$\begin{aligned} \dot{x} &= Ax + Bu = \begin{bmatrix} 0 & I \\ M_1 & M_2 \end{bmatrix} x + \begin{bmatrix} 0 \\ I \end{bmatrix} u \\ y &= Cx = \begin{bmatrix} I & 0 \end{bmatrix} x \end{aligned} \tag{21}$$

where $x \in \mathbb{R}^4$ is a state vector with the first two elements corresponding to a satellite's relative position in a local vertical local horizontal (LVLH) frame and the last two corresponding to a satellite's relative velocity in the LVLH frame. The external control acceleration is $u \in \mathbb{R}^2$. The matrices M_1 and M_2 are given by

$$M_1 = \begin{bmatrix} 3\omega^2 & 0 \\ 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 2\omega \\ -2\omega & 0 \end{bmatrix} \tag{22}$$

where ω is the mean motion of the circular reference orbit and is set to 4 rad/h for the remainder of this paper. This corresponds to a low earth orbit. Other relative motion models such as those based on time-varying relative orbital elements [40,41] are also applicable within the reachable set-based solution strategy. In all the simulations, the output vector y is to be the relative position of the players, which means that the output matrix is $C = \begin{bmatrix} I & 0 \end{bmatrix}$, with I being the 2×2 identify matrix and 0 being the 2×2 zero matrix. The relative position is chosen as the output because capture occurs in position space.

5. Numerical Simulations

Numerical simulations of the three cases introduced in Section 3 are provided in this section. All the agents evolve according to CW dynamics as explained in Section 4. Each agent's control set is of the form

$$\mathcal{U}(t) = \{u \in \mathbb{R}^2 \ : \ ||u|| \leq \rho\} \tag{23}$$

with ρ being a prescribed upper bound on the control magnitude. This is a second-order cone constraint. Additional constraints consistent with those in Equations (1)–(5) are also present and described in the following subsections.

5.1. Single Pursuer and Single Evader

Consider a single pursuer and single evader pursuit–evasion problem with the players' dynamics given in (21). The initial states of the pursuer and evader are

$$x_{0,P} = [0 \text{ km}, 1 \text{ km}, 0 \text{ km/h}, 0 \text{ km/h}]^\top \tag{24}$$

$$x_{0,E} = [0 \text{ km}, 0 \text{ km}, 0 \text{ km/h}, 0 \text{ km/h}]^\top. \tag{25}$$

The pursuer's and evader's states are constrained by a linear inequality as in (3) with

$$D = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \text{ km}. \tag{26}$$

The D matrix is dimensionless. The control of the pursuer is bounded by $\rho_P = 1 \text{ km/h}^2$, and the control of the evader is bounded by $\rho_E = 0.5 \text{ km/h}^2$.

Figure 9 shows the reachable sets of the pursuer and evader when the capture happens. The trajectories that the pursuer and evader take to reach the capture point are also shown in the figure. The capture happens at $t^* = 1.42$ h. It is evident from the figure that the capture happens at the boundary of both the pursuer's and evader's reachable sets.

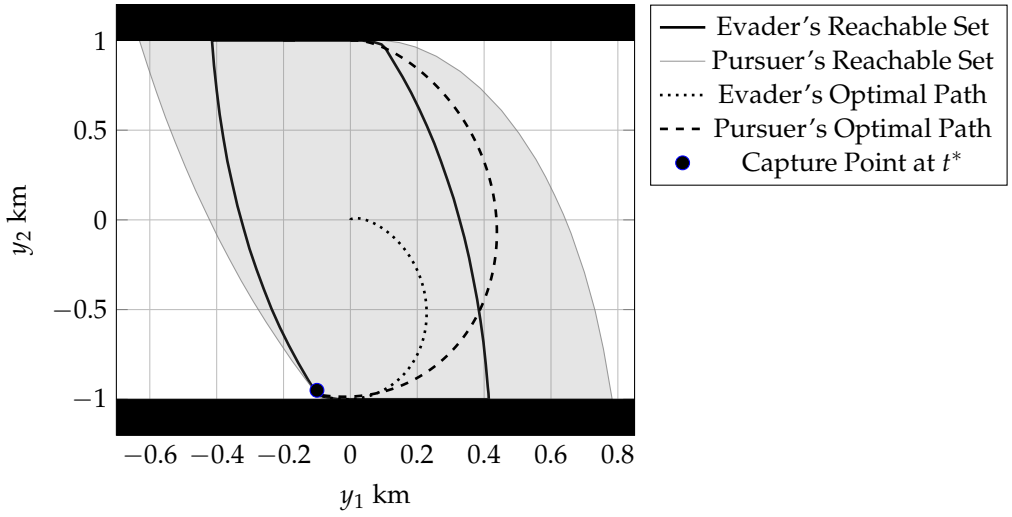

Figure 9. Pursuer's (shaded) and evader's (solid) reachable sets at capture time with the trajectories of the pursuer (dashed) and evader (dotted) from their initial position to the capture point (black circle). The black areas at the top and bottom represent keep-out zones generated by the linear constraints.

5.2. Multiple Pursuers and Single Evader

Now consider a pursuit–evasion problem with four pursuers and a single evader. The dynamics of the pursuers and evader are given by the CW Equation (21). The initial states of the players are

$$x_{0,P_1} = [0\text{ km}, 0.5\text{ km}, 0\text{ km/h}, 0\text{ km/h}]^\top, \tag{27}$$
$$x_{0,P_2} = [0\text{ km}, -0.5\text{ km}, 0\text{ km/h}, 0\text{ km/h}]^\top, \tag{28}$$
$$x_{0,P_3} = [\omega^{-2}\text{ km}, 0.5\text{ km}, 0\text{ km/h}, 0\text{ km/h}]^\top, \tag{29}$$
$$x_{0,P_4} = [-\omega^{-2}\text{ km}, -0.5\text{ km}, 0\text{ km/h}, 0\text{ km/h}]^\top, \tag{30}$$
$$x_{0,E} = [0\text{ km}, 0\text{ km}, 0\text{ km/h}, 0\text{ km/h}]^\top. \tag{31}$$

The states of all the players are constrained with the same D and b specified in (26) as in the previous simulation. The control input of all the pursuers is constrained by $\rho_P = 1\text{ km/h}^2$, and the control input of the evader is constrained by $\rho_E = 0.5\text{ km/h}^2$.

Figure 10 shows the reachable sets of all the pursuers as well as the evader. The capture happens at the origin. The evader could be captured by either pursuer 1 or pursuer 2, but only pursuer 1's trajectory from its initial point to the capture point is shown. Interestingly, the capture happens at the evader's initial location as that is the last point covered by the union of the pursuers' reachable sets. The capture happens at $t^* = 0.84$ h.

5.3. Single Pursuer and Multiple Evaders

Lastly, a game with three evaders and a single pursuer is considered. The dynamics of all the agents are again given by the CW equations. The initial states of the pursuer and evaders are

$$x_{0,P} = [0, 0, 0, 0]^\top, \tag{32}$$
$$x_{0,E_1} = [0, 0.5, 0, 0]^\top, \tag{33}$$
$$x_{0,E_2} = [0, -0.75, 0, 0]^\top, \tag{34}$$
$$x_{0,E_3} = [\omega^{-2}, 0, 0, 0]^\top. \tag{35}$$

There are no state constraints for the agents but the control inputs are constrained with $\rho_P = 1\text{ km/h}^2$ for the pursuer and $\rho_E = 0.25\text{ km/h}^2$ for the evaders.

Figure 11 shows the reachable sets of all the agents at the time instance when the first evader is captured. Figure 12 shows the reachable sets of the remaining two evaders as well as the pursuer's set when the second evader is captured. Figure 13 shows the reachable sets of the pursuer and the third evader at the time of the last capture, which terminates the game. The captures happen at $t_1 = 0.92$ h, $t_{1,2} = 2.32$ h, and $t_{1,2,3} = t^* = 6.24$ h.

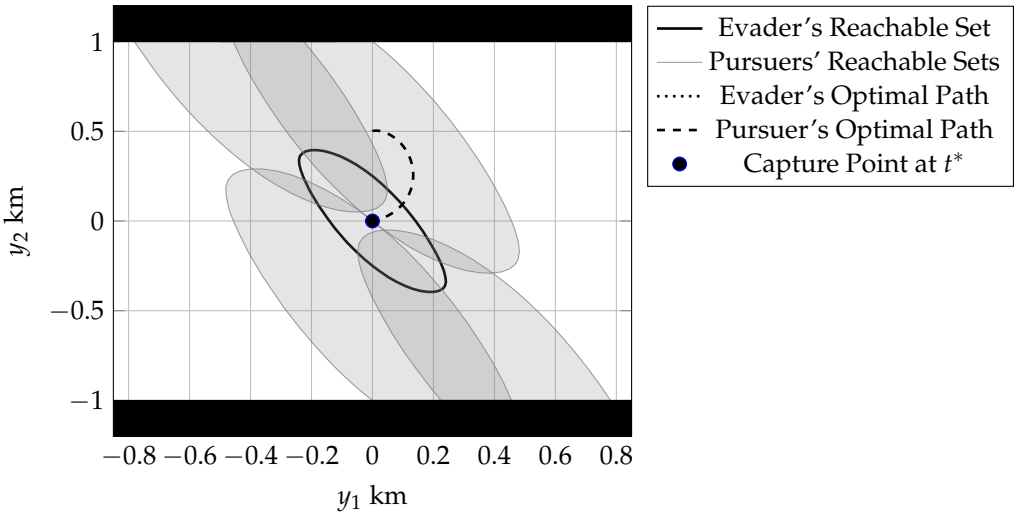

Figure 10. Pursuers' (shaded) and evader's (solid) reachable sets at capture time with the trajectory of the capturing pursuer (dashed) from its initial output to the capture point (black circle). The evader's trajectory (dotted but invisible to the eye) is stationary at the origin. The black areas at the top and bottom represent keep-out zones generated by the linear constraints.

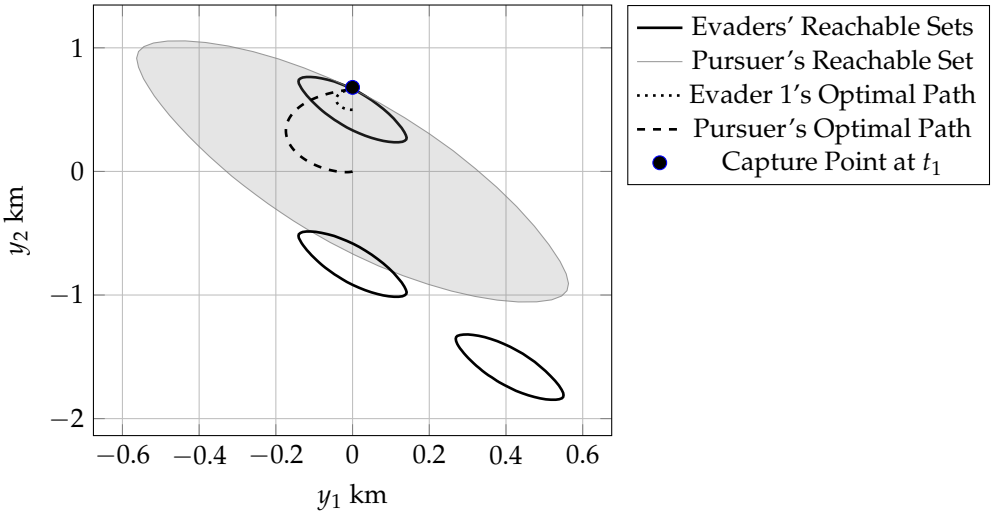

Figure 11. Pursuer's (shaded) and evaders' (solid) reachable sets at the time of the first capture with the trajectories of the pursuer (dashed) and first evader (dotted) from their initial positions to the capture point (black circle). Evader 1's reachable set is the uppermost. Evader 2's reachable set is the middle one. Evader 3's reachable set is the lowermost.

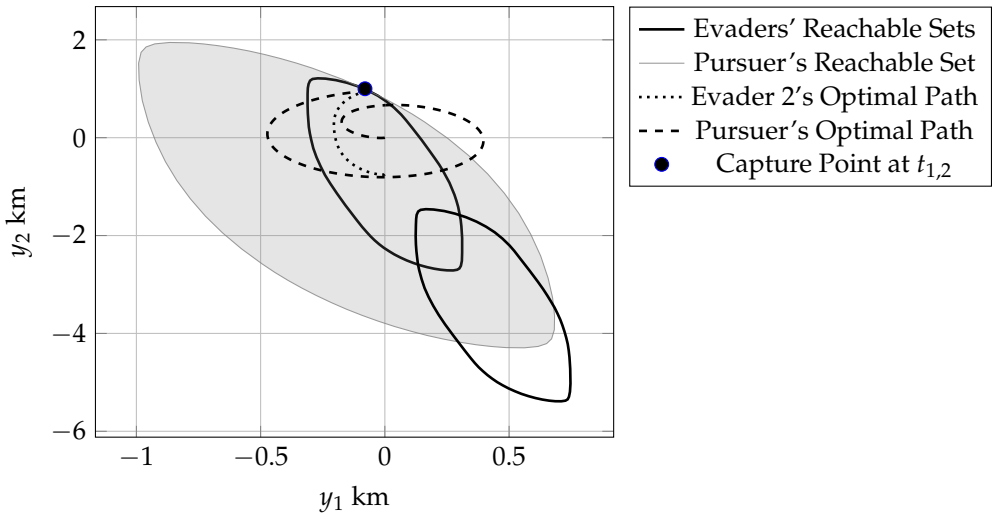

Figure 12. Pursuer's (shaded) and evaders' (solid) reachable sets at the time of the second capture with the trajectories of the pursuer (dashed) and first evader (dotted) from their initial positions to the capture point (black circle). Evader 2's reachable set is the uppermost. Evader 3's reachable set is the lowermost.

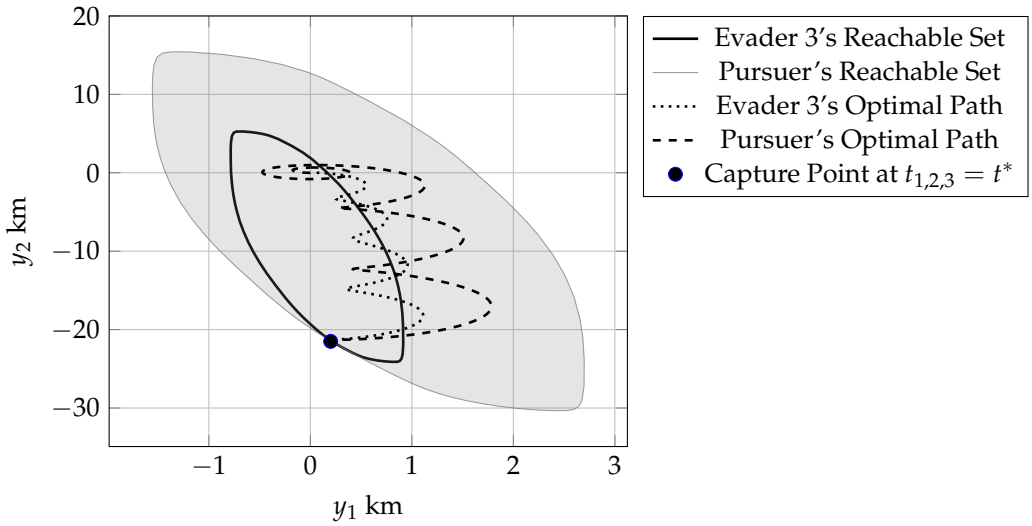

Figure 13. Pursuer's (shaded) and evader 3's (solid) reachable sets at the time of the final capture with the trajectories of the pursuer (dashed) and first evader (dotted) from their initial positions to the capture point (black circle). This final capture terminates the game.

5.4. Computational Performance

The reachable set strategy for solving time-optimal pursuit–evasion games relies upon bisection (see Algorithm 1) and reachable set calculations (see Section 2.1). At each time within a bisection, a reachable set calculation is required for each player in the game. Each reachable set calculation requires the solution of a single SDP (for the initial simplex) and multiple SOCPs (for growing the faces). SDPs are solved by using the software SDPT3 [42], and SOCPs are solved by using the software Gurobi [43]. Problems are parsed using YALMIP [44] in the MATLAB environment [45]. Table 1 shows the average SDP solve time, average SOCP solve time, average number of SOCPs solved, and average time to calculate each reachable set for the three simulations described in Sections 5.1–5.3. The computations were performed on an HP laptop with a 10th generation Intel i7 1.8 GHz processor.

Table 1. Reachable set calculation times.

Case	Average SDP Time (s)	Average SOCP Time (s)	Average SOCP Number	Average Total Time (s)
Section 5.1	0.59	0.0068	40	0.86
Section 5.2	0.49	0.0064	44	0.77
Section 5.3	0.34	0.0059	59	0.69

The overall run time of the algorithm is affected primarily by Algorithm 1 parameters t_{min}, t_{max}, and ϵ because these dictate the number of bisection iterations and hence number of reachable sets that must be calculated. The best way to reduce the total run time is to provide good estimates of the lower and upper bounding times t_{min} and t_{max}. Table 2 shows the total run time for the algorithm for the three simulations described in Sections 5.1–5.3 on instances requiring 5, 10, 25, and 100 bisection iterations (itr).

Table 2. Total run time for the reachable set algorithm.

Case	Total Time (s) for 5 itr	Total Time (s) for 10 itr	Total Time (s) for 25 itr	Total Time (s) for 100 itr
Section 5.1	8.6	17.2	43.0	172
Section 5.2	19.3	38.5	96.3	385
Section 5.3	13.8	27.6	69.0	276

To summarize, the reachable set strategy takes minutes (on a personal laptop) to solve constrained time-optimal pursuit–evasion games with two to five players.

6. Conclusions

This paper presented a technique using convex optimization to numerically construct reachable sets and to solve time-optimal pursuit–evasion games when the dynamics are linear and all the constraints are convex. These requirements are consistent with linearized relative orbital motion models such as the Clohessy–Wiltshire equations and relative orbital elements. Games with (1) a single pursuer and single evader, (2) multiple pursuers and a single evader, and (3) a single pursuer and multiple evaders were considered. Traditional formulations, such as those based on variational equations, are not tractable in the presence of practical actuator and state constraints. On the contrary, such constraints add minimal complexity to the reachable set strategy described herein. This is true because construction of the reachable sets has been reduced to a sequence of convex programs. Three numerical simulations were presented to demonstrate the strategy for a constellation of satellites in close proximity in low earth orbit. The computation times for the reachable sets were in subseconds, and the total run times for the algorithm were in minutes.

Author Contributions: Conceptualization, O.J. and M.W.H.; methodology, O.J. and M.W.H.; software, O.J.; validation, O.J. and M.W.H.; formal analysis, O.J. and M.W.H.; investigation, O.J. and M.W.H.; resources, O.J. and M.W.H.; data curation, O.J. and M.W.H.; writing—original draft preparation, O.J.; writing—review and editing, O.J. and M.W.H.; visualization, O.J.; supervision, M.W.H.; project administration, M.W.H.; funding acquisition, M.W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Office of Naval Research, grant number N00014-22-1-2131.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The continuous time system is

$$\dot{x} = A(t)x + B(t)u. \tag{A1}$$

Upon defining the state-transition matrix $\Phi(t_{k+1}, t_k)$ associated with $A(t)$ on the interval $[t_k, t_{k+1}]$, it can be shown by direct differentiation that a solution to Equation (A1) on the interval $[t_k, t_{k+1}]$ is

$$x(t_{k+1}) = \Phi(t_{k+1}, t_k)x(t_k) + \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)B(\tau)u(\tau)d\tau. \tag{A2}$$

For small time steps $\Delta t = t_{k+1} - t_k$, evaluation of the integral may be approximated by fixing u at its initial value $u(t_k)$. Upon defining

$$\bar{A}_k = \Phi(t_{k+1}, t_k), \quad \bar{B}_k = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)B(\tau)d\tau, \tag{A3}$$

a discrete-time approximation of the continuous-time system is

$$x_{k+1} = \bar{A}_k x_k + \bar{B}_k u_k, \quad k = 0, \ldots, N-1, \tag{A4}$$

which is the same as Equation (7). Upon fixing all other quantities at the node times, e.g., $C_k = C(t_k)$, the discretization is complete.

References

1. Mizukami, K.; Eguchi, K. A geometrical approach to problems of pursuit-evasion games. *J. Frankl. Inst.* **1977**, *303*, 371–384. [CrossRef]
2. Isaacs, R. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*; Dover Publications: Mineola, NY, USA, 1999.
3. Weintraub, I.E.; Pachter, M.; Garcia, E. An introduction to pursuit-evasion differential games. In Proceedings of the 2020 American Control Conference (ACC), IEEE, Denver, CO, USA, 1–3 July 2020; pp. 1049–1066.
4. Shinar, J.; Gutman, S. Recent advances in optimal pursuit and evasion. In Proceedings of the 1978 IEEE Conference on Decision and Control Including the 17th Symposium on Adaptive Processes, IEEE, San Diego, CA, USA, 10–12 January 1979; pp. 960–965.
5. Shinar, J.; Glizer, V.Y.; Turetsky, V. A pursuit-evasion game with hybrid pursuer dynamics. *Eur. J. Control* **2009**, *15*, 665–684. [CrossRef]
6. Shinar, J. Solution techniques for realistic pursuit-evasion games. In *Control and Dynamic Systems*; Elsevier: Amsterdam, The Netherlands, 1981; Volume 17, pp. 63–124.
7. Imado, F.; Kuroda, T. A method to solve missile-aircraft pursuit-evasion differential games. *IFAC Proc. Vol.* **2005**, *38*, 176–181. [CrossRef]
8. Carr, R.W.; Cobb, R.G.; Pachter, M.; Pierce, S. Solution of a Pursuit–Evasion game using a near-optimal strategy. *J. Guid. Control Dyn.* **2018**, *41*, 841–850. [CrossRef]
9. Greenwood, N. A differential game in three dimensions: The aerial dogfight scenario. *Dyn. Control* **1992**, *2*, 161–200. [CrossRef]
10. Calise, A.J.; Yu, X.m. An analysis of a four state model for pursuit-evasion games. In Proceedings of the 1985 24th IEEE Conference on Decision and Control, IEEE, Fort Lauderdale, FL, USA, 11–13 December 1985; pp. 1119–1121.
11. Horie, K.; Conway, B.A. Optimal fighter pursuit-evasion maneuvers found via two-sided optimization. *J. Guid. Control Dyn.* **2006**, *29*, 105–112. [CrossRef]
12. Shen, D.; Pham, K.; Blasch, E.; Chen, H.; Chen, G. Pursuit-evasion orbital game for satellite interception and collision avoidance. In Proceedings of the Sensors and Systems for Space Applications IV, International Society for Optics and Photonics, Orlando, FL, USA, 25–29 April 2011; Volume 8044, p. 80440B.
13. Blasch, E.P.; Pham, K.; Shen, D. Orbital satellite pursuit-evasion game-theoretical control. In Proceedings of the 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), IEEE, Montreal, QC, Canada, 2–5 July 2012; pp. 1007–1012.
14. Stupik, J.; Pontani, M.; Conway, B. Optimal pursuit/evasion spacecraft trajectories in the hill reference frame. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, MN, USA, 13–16 August 2012; p. 4882.
15. Shen, D.; Jia, B.; Chen, G.; Pham, K.; Blasch, E. Game optimal sensor management strategies for tracking elusive space objects. In Proceedings of the 2017 IEEE Aerospace Conference, IEEE, Big Sky, MT, USA, 4–11 March 2017; pp. 1–8.
16. Zeng, X.; Yang, L.; Zhu, Y.; Yang, F. Comparison of Two Optimal Guidance Methods for the Long-Distance Orbital Pursuit-Evasion Game. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *57*, 521–539. [CrossRef]
17. Yang, B.; Liu, P.; Feng, J.; Li, S. Two-Stage Pursuit Strategy for Incomplete-Information Impulsive Space Pursuit-Evasion Mission Using Reinforcement Learning. *Aerospace* **2021**, *8*, 299. [CrossRef]
18. Bryson, A.; Ho, Y. *Applied Optimal Control: Optimization, Estimation and Control*; CRC Press: Boca Raton, FL, USA, 1975.
19. Basar, T. Relaxation Techniques and asynchronous algorithms for online computation of non-cooperative equilibria. *J. Econ. Dyn. Control* **1987**, *11*, 531–549. [CrossRef]

20. Uryasev, S.; Rubinstein, R. On relaxation algorithms in computation of noncooperative equilibria. *IEEE Trans. Autom. Control* **1994**, *39*, 1263–1267. [CrossRef]
21. Johnson, P.A. Numerical Solution Methods for dIfferential Game Problems. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2009.
22. Horie, K.; Conway, B.A. Genetic algorithm preprocessing for numerical solution of differential games problems. *J. Guid. Control Dyn.* **2004**, *27*, 1075–1078. [CrossRef]
23. Sun, W.; Tsiotras, P.; Lolla, T.; Subramani, D.N.; Lermusiaux, P.F. Multiple-pursuer/one-evader pursuit–evasion game in dynamic flowfields. *J. Guid. Control Dyn.* **2017**, *40*, 1627–1637. [CrossRef]
24. Chung, C.F.; Furukawa, T.; Goktogan, A.H. Coordinated control for capturing a highly maneuverable evader using forward reachable sets. In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, IEEE, Orlando, FL, USA, 15–19 May 2006; pp. 1336–1341.
25. Salmon, D.M.; HElNE, W. Reachable sets analysis-an efficient technique for performing missile/sensor tradeoff studies. *AIAA J.* **1973**, *11*, 927–931. [CrossRef]
26. Chung, C.F.; Furukawa, T. A reachability-based strategy for the time-optimal control of autonomous pursuers. *Eng. Optim.* **2008**, *40*, 67–93. [CrossRef]
27. Zanardi, C.; Hervé, J.Y.; Cohen, P. Escape strategy for a mobile robot under pursuit. In Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century, IEEE, Vancouver, BC, Canada, 22–25 October 1995; Volume 4, pp. 3304–3309.
28. Jansson, O.; Harris, M.; Geller, D. A Parallelizable Reachable Set Method for Pursuit-Evasion Games Using Interior-Point Methods. In Proceedings of the 2021 IEEE Aerospace Conference (50100), IEEE, Big Sky, MT, USA, 6–13 March 2021; pp. 1–9.
29. Gong, H.; Gong, S.; Li, J. Pursuit-Evasion Game for Satellites Based on Continuous Thrust Reachable Domain. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 4626–4637. [CrossRef]
30. Colonius, F.; Szolnoki, D. Algorithms for computing reachable sets and control sets. *IFAC Proc. Vol.* **2001**, *34*, 723–728. [CrossRef]
31. Girard, A.; Le Guernic, C.; Maler, O. Efficient computation of reachable sets of linear time-invariant systems with inputs. In Proceedings of the International Workshop on Hybrid Systems: Computation and Control, Santa Barbara, CA, USA, 29–31 March 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 257–271.
32. Varaiya, P. Reach set computation using optimal control. In *Verification of Digital and Hybrid Systems*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 323–331.
33. Dueri, D.; Raković, S.V.; Açıkmeşe, B. Consistently improving approximations for constrained controllability and reachability. In Proceedings of the 2016 European Control Conference (ECC), IEEE, Aalborg, Denmark, 29 June–1 July 2016; pp. 1623–1629.
34. Dueri, D.; Açıkmeşe, B.; Baldwin, M.; Erwin, R.S. Finite-horizon controllability and reachability for deterministic and stochastic linear control systems with convex constraints. In Proceedings of the 2014 American Control Conference, IEEE, Portland, OR, USA, 4–6 June 2014; pp. 5016–5023.
35. Yang, R.; Liu, X. Reachable set computation of linear systems with nonconvex constraints via convex optimization. *Automatica* **2022**, *146*, 110632. [CrossRef]
36. Pecsvaradi, T.; Narendra, K.S. Reachable sets for linear dynamical systems. *Inf. Control* **1971**, *19*, 319–344. [CrossRef]
37. Hermes, H.; LaSalle, J.P. *Functional Analysis and Time Optimal Control*; Academic Press Inc.: New York, NY, USA, 1969.
38. Guenin, B.; Konemann, J.; Tuncel, L. *A Gentle Introduction to Optimization*; Cambridge University Press: Cambridge, UK, 2014.
39. Clohessy, W.; Wiltshire, R. Terminal guidance system for satellite rendezvous. *J. Aerosp. Sci.* **1960**, *27*, 653–658. [CrossRef]
40. Bennett, T.; Schaub, H. Continuous-Time Modeling and Control Using Nonsingular Linearized Relative-Orbit Elements. *J. Guid. Control Dyn.* **2016**, *39*, 2605–2614. [CrossRef]
41. Sullivan, J.; Grimberg, S.; D'Amico, S. Comprehensive Survey and Assessment of Spacecraft Relative Motion Dynamics Models. *J. Guid. Control Dyn.* **2017**, *40*, 1837–1859. [CrossRef]
42. Toh, K.C.; Todd, M.J.; Tutuncu, R.H. SDPT3—A Matlab software package for semidefinite programming, Version 1.3. *Optim. Methods Softw.* **1999**, *11*, 545–581. [CrossRef]
43. Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*; Gurobi Optimization, LLC: Houston, TX, USA, 2022.
44. Löfberg, J. YALMIP: A Toolbox for Modeling and Optimization in MATLAB. In Proceedings of the CACSD Conference, Taipei, Taiwan, 2–4 September 2004.
45. The Mathworks, Inc. *MATLAB 2023*; The Mathworks, Inc.: Natick, MA, USA, 2023.