PURSUIT AND EVASION GAMES: SEMI-DIRECT AND COOPERATIVE CONTROL METHODS

A Thesis

by

ALLEN SAYRE PARISH III

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2008

Major Subject: Computer Engineering

PURSUIT AND EVASION GAMES: SEMI-DIRECT AND COOPERATIVE CONTROL METHODS

A Thesis

by

ALLEN SAYRE PARISH III

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Co-Chairs of Committee,	John E. Hurtado
	Dezhen Song
Committee Members,	Ricardo Gutierrez-Osuna
Head of Department,	Valerie E. Taylor

May 2008

Major Subject: Computer Engineering

ABSTRACT

Pursuit and Evasion Games:

Semi-direct and Cooperative Control Methods. (May 2008)

Allen Sayre Parish III, B.S., Texas A&M University

Co–Chairs of Advisory Committee: Dr. John E. Hurtado
Dr. Dezhen Song

Pursuit and evasion games have garnered much research attention since the class of problems was first posed over a half century ago. With wide applicability to both civilian and military problems, the study of pursuit and evasion games showed much early promise. Early work generally focused on analytical solutions to games involving a single pursuer and a single evader. These solutions generally assumed simple system dynamics to facilitate convergence to a solution. More recently, numerical techniques have been utilized to solve more difficult problems. While many sophisticated numerical tools exist for standard optimization and optimal control problems, developing a more complete set of numerical tools for pursuit and evasion games is still a developing topic of research.

This thesis extends the current body of numeric solution tools in two ways. First, an existing approach that modifies sophisticated optimization tools to solve two player pursuer and evasion games is extended to incorporate a class of state inequality constraints. Several classical problems are solved to illustrate the efficacy of the new approach. Second, a new cooperation metric is introduced into the system objective function for multi-player pursuit and evasion games. This new cooperation metric encourages multiple pursuers to surround and then proceed to capture an evader. Several examples are provided to demonstrate this new cooperation metric.

To my wife Julie, whom I love more dearly every day

ACKNOWLEDGMENTS

I would first like to thank my wife Julie. She has been a constant source of encouragement since I first met her. She has been particularly helpful and uplifting during my graduate studies. I'm extremely blessed to have her as my companion and best friend.

This thesis work was conducted under the direction of Dr. John E. Hurtado. I can not thank him enough for his advisement and mentorship. I approach problems in a vastly improved way thanks to his teaching. Most of all I thank him for persevering with me and for me through the struggles of this research. He is a wonderful advisor.

I'd also like to thank my graduate research committee for their support during my studies. In particular, I'd like to thank Dr. Dezhen Song for his additional advisement.

Funding for my graduate studies was provided by the Department of Homeland Security through a graduate fellowship. I greatly appreciate the opportunities the fellowship provided me, including an internship opportunity at Sandia National Laboratories.

I also wish to thank my family for being so uplifting and supporting during my stay at Texas A&M University. In particular I'd like to thank my mom for being such a great mother. She's the best cheerleader a son could have.

Most of all I thank my Creator. I have had so many wonderful blessings and opportunities in my life. All the glory goes to Him.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Over the past half-century pursuit and evasion games have gained much research attention due to wide applicability in many research arenas such as civilian and defense problems. Rufus Issacs first proposed differential game theory to analyze interactions between a single pursuer and evader [1]. Subsequent extensions have expanded the general study to areas of cooperative control games, where multiple pursuers attempt to capture one or more evaders.

Differential game theory was met with much fanfare early in its existence. Much of the early work in the field relied on analytical solutions to solve relatively simple dynamic systems. More complex differential games often require sophisticated numerical techniques to solve. While numerical methods exist to directly minimize an objective function, these methods do not directly extend to differential game theory. Researchers have recently begun developing methods for employing traditional numerical techniques to solve differential games [2],[3]. One particularly promising result was recently developed by Horie and Conway [4]. These extensions are still being developed and have some fundamental limitations.

In addition to differential games with a single pursuer, a class of cooperative control games exist where multiple pursuers attempt to capture a single evader. Mathematical models for cooperation strategies in the system objective function are relatively sparse in cooperative control theory. In objective function-based cooperative control problems, multiple pursuers often do not actually devise a strategy before attempting capture of an evader. In other words, the system objective function remains

This thesis follows the style of *IEEE Transactions on Automatic Control.*

unchanged for multiple pursuers. This opportunity for future research is explored in this work.

This thesis contributes to existing research in two areas. First, an extension is provided to Horie and Conway's semi-direct approach to support a class of state inequality constraints. Second, a new elegant cooperative control method is introduced to encourage pursuers to surround an evader before capture. Numerical simulations are provided to illustrate both contribution areas.

This thesis first presents background material for differential game and optimal control theory. A relatively new semi-direct numerical technique for solving differential games is thoroughly introduced in chapter (III). A few examples are presented to illustrate that the technique can be used to successfully solve differential games. The semi-direct approach is then extended in chapter (IV) to a class of state inequality constraints. Several illustrative classical problems are solved to demonstrate the new extension. In chapter (V), a new cooperation metric is introduced for multi-pursuer pursuit and evasion games. The thesis concludes in chapter (VI) with a discussion of results and possible future work.

CHAPTER II

BACKGROUND

This research primarily relies on differential game and optimal control theory developed over the past half century. An overview of differential games and optimal control theory is presented first. An optimal control based approach to solve differential games is then presented. This type of approach, which uses analytical conditions for an optimal solution, is used heavily throughout the thesis. After an analytical method for solving differential games is presented, numerical techniques for optimal control and differential games are described. The background section concludes with a discussion of cooperative control games.

A. Differential Games

Pursuit and evasion games pit multiple players with opposing goals against each other. These games are relevant in real world situations, and are especially important in defense applications. One common civilian example is a suspect trying to avoid police capture. In this game, the evader is the suspect trying to avoiding police capture and the pursuer is the officer. The objectives of the two players in the game are opposite. As in this example, the evader generally seeks to maximize the amount of time before capture by the pursuer. One common military application that can be cast as a pursuit and evasion game is the missile interception problem. In this scenario, a guided missile seeks to hit an intended target, while an interceptor missile seeks to destroy it before the target is reached. The guided missile acts as an evader trying to avoid destructive capture by the interceptor.

Pursuit and evasion games became of particular interest during the start of the Cold War between the United States and the USSR. Isaacs originally developed differential game theory to solve such problems in 1951 [1]. Isaacs casts these problems as a pursuer and an evader trying to minimize or maximize an objective function, respectively. The resulting saddle point solution is often referred to as a “minimax” solution. Much of the early work in differential game theory solved pursuit and evasion games by using strictly analytical techniques. Although these traditional techniques were fruitful, more complicated pursuit and evasion problems can not always be solved with analytical approaches.

B. Optimal Control

While differential game theory solves the problem of simultaneously satisfying two opposing objectives, optimal control theory focuses on satisfying a single objective [5]. In a typical optimal control problem, controls are found to either minimize or maximize (hereafter, extremize) a specified objective function. For example, many optimal control problems find the control solution that minimizes the time or energy needed to achieve an objective. One such problem is the minimum time rest-to-rest problem for a satellite. In an optimal control problem such as this, analytical optimality conditions via Pontryagin’s theory are formed to solve the problem. These conditions ultimately result in solving a two-point boundary value problem. If a solution method utilizes all of the analytical optimality conditions to solve a two point boundary value problem (TPBVP), it is classified as an indirect method. Indirect methods that use the analytical optimality conditions require sophisticated knowledge of optimal control theory and are somewhat impractical for laymen practitioners. Strictly numerical approaches also exist to solve optimal control problems. These methods are often called direct methods, because they find controls that extremize the objective function *directly* via sophisticated numerical techniques. A direct approach does not use analytical necessary conditions to find an optimal solution. Direct approaches are more thoroughly discussed in section (C).

A method for finding optimal controls for a general optimal control problem without terminal constraints and a fixed final time is now discussed [5].

Initial conditions for the state variables, $\mathbf{x}$, are prescribed and the system evolves as follows.

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) \tag{2.1}$$

The objective function for the system then takes the following form.

$$J = \phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}, \mathbf{u}, t)dt \tag{2.2}$$

The objective function of this problem consists of soft constraints, ϕ, which penalize deviations from desired final values for some state variables, and an integral portion which penalizes functions of state and control variables throughout the duration of the game. Each optimal control problem will have a different objective function that depends on desired performance characteristics. The solution to this optimal control problem finds the control $\mathbf{u}$ that minimizes the objective function, J.

In order to form the analytical necessary conditions for an optimal solution, the Hamiltonian H is defined. The Hamiltonian consists of the integrand of the system objective function appended with adjoint, or costate variables, $\boldsymbol{\lambda}(t)$, multiplied by the system evolution equations.

$$H(\mathbf{x}, \mathbf{u}, t) = L(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T f(\mathbf{x}, \mathbf{u}, t) \tag{2.3}$$

One finds that the solution to the optimal control problem must satisfy the following set of equations via calculus of variations techniques.

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) \tag{2.4}$$

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}} \tag{2.5}$$

$$\frac{\partial H}{\partial \mathbf{u}} = 0 \tag{2.6}$$

Calculus of variations techniques also prescribe that final conditions for the costate variables must satisfy the following equation.

$$\boldsymbol{\lambda}(t_f) = \left.\frac{\partial \phi}{\partial \mathbf{x}}\right|_{t_f} \tag{2.7}$$

The boundary conditions for the state variables and the costate variables are split between initial conditions for $\mathbf{x}$ and terminal conditions for $\boldsymbol{\lambda}$. The solution to the optimal control problem is then reduced to solving the two point boundary value problem described by Eqns. (2.4) - (2.7). The solution of these equations yields the optimal values for $\mathbf{u}$ and is the solution to the optimal control problem.

Optimal control problems can not be solved analytically except for problems with relatively simple system dynamics. Numeric techniques are needed to solve all but the simplest of optimal control problems. As mentioned previously, numeric techniques can either classified as an indirect or direct approach depending on whether analytical necessary conditions are used.

C. Numerical Techniques for Optimal Control Problems

This section provides an introduction to some prevailing numerical techniques for solving optimal control problems. These methods are considered direct methods because they do not rely on analytical necessary conditions. The methods directly output controls that extermize the objective function using the governing equations for the system and any existing constraints on the system. Nonlinear programming tools such as SNOPT, NPSOL, or FMINCON are capable of solving difficult minimization problems for which no analytical solution exists [6] [7] [8]. These methods differ from *indirect* approaches since they do not require any analytical optimality conditions to perform function minimization.

A nonlinear programming problem consists of solving for free parameters (controls) that best satisfies a system objective function and constraints. Numerical methods generally attempt to work towards a satisfactory solution by forming an approximate model of the original problem at each iteration. Through successive iterations, numerical approaches can often converge on a solution to the original problem.

In FMINCON, an optimization problem is broken into *sequential quadratic programs* (SQP) in order to find a satisfactory solution [8]. A sequential quadratic programming approach utilizes a quadratic approximation for the system objective function and a linear model for system constraints. At each iteration, a quadratic problem is solved to progressively move towards a solution.

Solutions generated by most numerical approaches, including FMINCON, are not guaranteed to be global solutions. For example, in some cases extrema may look like the global solution to the numerical method. If all constraints are satisfied and perturbations near a local extrema do not produce a more fit solution, the numerical method will terminate and output the local extrema as a solution. Figure (1) illustrates a potential solution space to a one dimensional problem. It is very plausible that the numerical solution might conclude that x^* is the global minima if given x^0 as an initial solution, when in fact, x^{min} is the true global minima for the problem.

Tradeoffs exist when choosing between numerical solutions. Absolute global convergence is exchanged for ease of use and solution speed. Since global convergence can not be guaranteed, the analysis of solutions is somewhat difficult. Still, numerical approaches to optimization problems are reliable for many classes of systems.

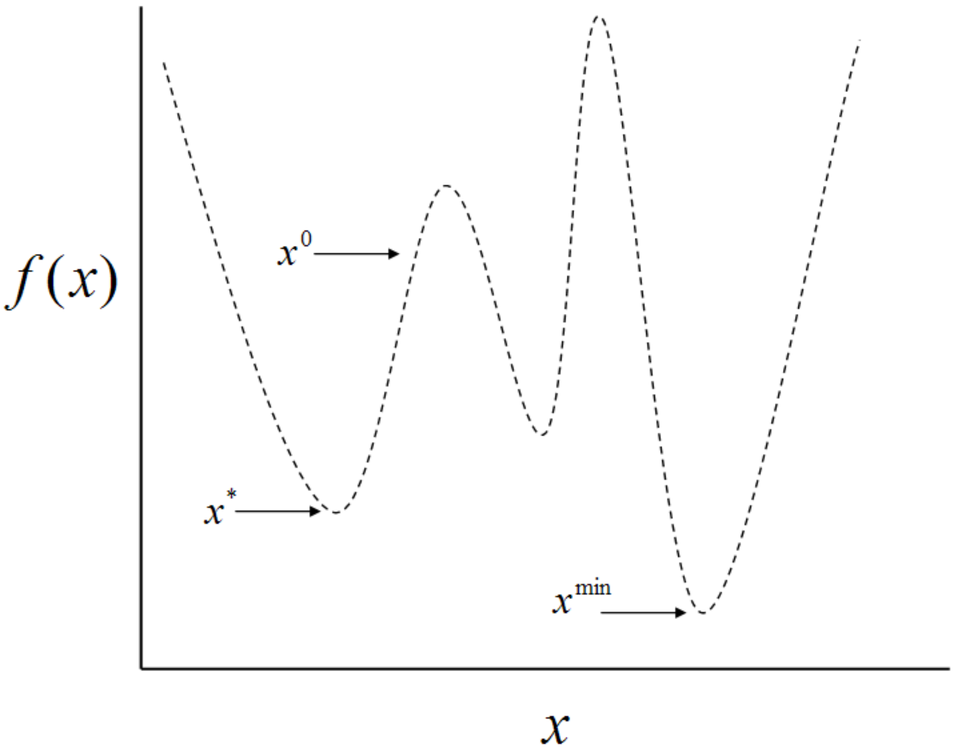

Fig. 1. Comparison of Global and Local Minimas

D. Optimal Control-based Solution to Differential Games

Much of the optimal control development discussed thus far also applies to solving a differential game. Bryson and Ho develop necessary conditions for the two player differential game that parallel those derived for the optimal control problem [5]. Consider the two player differential game of the following form.

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{w}, \mathbf{v}, t) \tag{2.8}$$

Here **w** and **v** are the pursuer and evader control variables, respectively. The state variables, **x**, of the system can often be broken up into variables that describe the pursuer's state and variables that described the evader's state. This separation is denoted as $\mathbf{x}_p$ and $\mathbf{x}_e$, respectively. The dynamic system is then rewritten as follows.

$$\dot{\mathbf{x}}_p = f_p(\mathbf{x}_p, \mathbf{w}, t) \tag{2.9}$$

$$\dot{\mathbf{x}}_e = f_e(\mathbf{x}_e, \mathbf{v}, t) \tag{2.10}$$

The objective function for this system can then be formed.

$$J = \phi(\mathbf{x}_p(t_f), \mathbf{x}_e(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}_p, \mathbf{x}_e, \mathbf{w}, \mathbf{v}, t)dt \tag{2.11}$$

The solution for this two player game is a saddle point [1],[5]. The value of the "minimax" solution can be written as follows.

$$V = \max_{\mathbf{v}} \min_{\mathbf{w}} J \tag{2.12}$$

The Hamiltonian and the costate equations are then the following.

$$H = \boldsymbol{\lambda}_p^T f_p + \boldsymbol{\lambda}_e^T f_e + L \tag{2.13}$$

$$\dot{\boldsymbol{\lambda}}_p = -\frac{\partial H}{\partial \mathbf{x}_p} \tag{2.14}$$

$$\dot{\boldsymbol{\lambda}}_e = -\frac{\partial H}{\partial \mathbf{x}_e} \tag{2.15}$$

The optimality conditions for the player controls can then be defined.

$$\frac{\partial H}{\partial \mathbf{w}} = 0 \tag{2.16}$$

$$\frac{\partial H}{\partial \mathbf{v}} = 0 \tag{2.17}$$

The final time values for the adjoint equations are then given as follows.

$$\boldsymbol{\lambda}_p(t_f) = \left.\frac{\partial \phi}{\partial \mathbf{x}_p}\right|_{t_f} \tag{2.18}$$

$$\boldsymbol{\lambda}_e(t_f) = \left.\frac{\partial \phi}{\partial \mathbf{x}_e}\right|_{t_f} \tag{2.19}$$

The preceding development identifies the necessary conditions for an optimal solution of a two player fixed time differential game without terminal constraints. An indirect approach uses all necessary conditions along with the system evolution equations and solves the two point boundary value problem. The next section describes solving differential games using direct optimization tools.

E. Numerical Techniques for Differential Games

Traditional direct methods can not simultaneously satisfy two objective functions [2],[9]. Accordingly, the tools can not be used as-is for pursuit and evasion differential games because these problems require multiple objective functions to be satisfied. As a result, many complicated pursuit and evasion games have previously remained unsolved. Providing a complete numerical tool for solving general pursuit and evasion games is still an active topic of research.

Researchers have recently developed new methods to adapt existing nonlinear programming techniques to solve pursuit and evasion games. Raivio and Ehtamo solve a complicated pursuit and evasion game by decomposing the problem into sequential one-sided games [9]. The first subproblem minimizes the system objective function while holding the evader's controls constant. The second subproblem maximizes the same objective function while keeping the pursuer's control constant. Since the subproblems reduce the differential game to a traditional optimal control problem,the smaller subproblems are solved using a direct nonlinear programming solution [3].

Horie and Conway develop an innovative approach that incorporates some of the analytical necessary conditions to solve a pursuit and evasion game with a nonlinear programming tool [2]. In this method, the optimal controls for one player are solved using the optimality conditions and expressed via adjoint variables for that player. By solving analytically for one set of controls, the differential game is transformed into one of finding an extrema for a single objective function. The technique can be thought of as a semi-direct optimization procedure since some of the analytical optimality conditions are used. Through this problem transformation, nonlinear programming techniques can be used to solve realistic pursuit and evasion games [2], [4], [10]. Horie and Conway's semi-direct method has successfully solved challenging two player differential games. However, the method they propose is unable to solve problems that have state inequality constraints. This limitation makes the method unable to solve many classical problems such as Breakwell's football problem or Isaacs's patrolling a channel problem [11], [1].

F. Cooperative Control

In a cooperative control problem, multiple entities work together to accomplish a common objective. A wide variety of applications exist including unmanned air vehicle formation control, search and rescue problems, and satellite configuration problems [12]. Generally, cooperative control solutions rely on communication between players to share state information in order to accomplish the objective. Sometimes cooperating players are teamed against a player that wishes to prevent them from accomplishing their objective.

Such is the case in multiplayer pursuit and evasion games. In these games, multiple pursuers generally attempt to capture or destroy one or more evaders. For example, multiple police officers may work cooperatively to apprehend a single suspect. The pursuers work together cooperatively to achieve the objective, but the evader(s) works to prevent the objective from being completed.

In real world multi-player pursuit and evasion games, pursuers often employ a common strategy to aid in capturing the evader. For example, a general might instruct his army to perform a flanking maneuver before attacking the opponent. Police officers attempting to apprehend a criminal might try to use their environment to corner a suspect before arresting him. While cooperation strategies are abundant in real world, they are often difficult to incorporate in mathematical problem formulations. This thesis presents a new cooperation strategy that can be directly incorporated into the system objective function in chapter (V).

CHAPTER III

SEMI-DIRECT METHOD

A. Development

As mentioned previously, numerical software tools for function minimization are not designed to simultaneously satisfy two objective functions. A numerical solver such as NPSOL extremizes a given objective function [7]. These tools are extremely useful for solving optimal control problems, but can not used without modification to solve differential games where a saddle point, or “minimax”, solution is required.

Horie and Conway sought to solve air combat problems with realistic aircraft dynamics. Analytical techniques can be used to solve very simple differential games, but are unable to solve problems with realistic dynamics. Numerical techniques are capable of solving more complicated problems, and can generally be characterized as either indirect methods or direct methods. Direct methods minimize an objective function without any knowledge of optimality conditions from first order necessary conditions or Pontryagin’s principle.

Indirect methods, on the other hand, use necessary conditions found via Pontryagin’s principle. A solution is found by solving the two-point boundary value problem associated with these necessary conditions and the system dynamics. One such TPBVP was presented in the previous chapter for a general differential game.

Horie and Conway sought a way to capture the benefits of using a numerical solver without having to implement all necessary conditions typically found in indirect solutions. The method uses some of necessary conditions and can be thought of a semi-direct approach to solving differential games because it uses some, but not all, of the necessary conditions found in a indirect approach. That is, the solution has characteristics of both indirect and direct numerical techniques. The semi-direct method is presented below.

Again consider a pursuit and evasion game with two players. The pursuer is attempting to minimize an objective function J while the evader seeks to maximize it. Initial conditions for both players are prescribed, and a fixed final time is assumed. The mathematical description of this game is presented previously, but is duplicated here for clarity.

$$\dot{\mathbf{x}}_p = f_p(\mathbf{x}_p, \mathbf{w}, t) \tag{3.1}$$

$$\dot{\mathbf{x}}_e = f_e(\mathbf{x}_e, \mathbf{v}, t) \tag{3.2}$$

$$J = \phi(\mathbf{x}_p(t_f), \mathbf{x}_e(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}_p, \mathbf{x}_e, \mathbf{w}, \mathbf{v}, t)dt \tag{3.3}$$

Recall that the solution of this two player game is a saddle point solution. The Hamiltonian for the system is formed, and the costate equation evolve as follows.

$$H = \boldsymbol{\lambda}_p^T f_p + \boldsymbol{\lambda}_e^T f_e + L \tag{3.4}$$

$$\dot{\boldsymbol{\lambda}}_p = -\frac{\partial H}{\partial \mathbf{x}_p} \tag{3.5}$$

$$\dot{\boldsymbol{\lambda}}_e = -\frac{\partial H}{\partial \mathbf{x}_e} \tag{3.6}$$

The optimality conditions for the player controls are then the following.

$$\frac{\partial H}{\partial \mathbf{w}} = 0 \tag{3.7}$$

$$\frac{\partial H}{\partial \mathbf{v}} = 0 \tag{3.8}$$

The final time conditions for the costate variables can be written as follows.

$$\boldsymbol{\lambda}_p(t_f) = \left.\frac{\partial \phi}{\partial \mathbf{x}_p}\right|_{t_f} \tag{3.9}$$

$$\boldsymbol{\lambda}_e(t_f) = \left.\frac{\partial \phi}{\partial \mathbf{x}_e}\right|_{t_f} \tag{3.10}$$

The semi-direct method uses the optimality condition for one player so that the numerical solver needs only to solve for the remaining player's controls. The problem is then transformed from finding a minimax type solution to extremizing the objective function as seen from a single player's perspective. To simplify the discussion, assume that the pursuer's control will be determined by the numerical solver. The above expressions for the evader's adjoint variables are incorporated so that the numerical solver solves the following problem. Minimize J subject to the system evolution equations, the evader costate variable equations, and the optimality condition for the evader.

The preceding problem can be solved by a nonlinear programming solver (or other direct numerical technique) because the problem has been transformed into one of finding the extrema for a single objective function. By utilizing the necessary conditions for one player in an innovative manner, Horie and Conway have successfully transformed the problem so that it can be solved by very sophisticated, widely-available numerical tools.

B. Problem Initialization

Horie and Conway note the importance of a feasible initial solution as input to a numerical solver. Without a sufficiently good initial solution, a numerical solver can not converge upon a solution. A means for acquiring an initial solution is then needed. For very simple problems, it is sometimes possible to provide an initial solution based on analytical insight. However, a more sophisticated method is often necessary. As a result, Horie and Conway use a genetic algorithm to generate the initial solution needed by their semi-direct method [10].

1. Genetic Algorithms

Genetic algorithms were first created by John Holland as a new search and optimization method that utilizes techniques derived from evolution theory [13]. Genetic algorithms, like other optimization techniques, find parameters that minimize or maximize a particular objective function.

Unlike many optimization techniques, however, genetic algorithms do not rely on gradient based approaches or other deterministic methods to iterate towards a optimal solution. A genetic algorithm uses an encoding of the parameter set of interest as it iterates towards an optimal solution. For an optimal control problem, this parameter set is the set of controls that minimizes the system objective function. A number of individual parameter sets, called a population, are used in each problem iteration to progress toward a solution. Population members are evaluated using a fitness function, which bears much similarity to the objective function in an optimal control problem. As the algorithm progresses iteratively, the members with the best fitness value have a greater chance of surviving to subsequent generations.

Between iterations, the best members of the previous iteration are probabilistically chosen based on the fitness function. After a new population has been selected, members generally undergo crossover and mutation transformation. Crossover is the process of probabilistically combining elements of two encoded parameter sets in the new population. Mutation is the process of randomly altering elements of a given population member [14].

The process of random mutation allows genetic algorithms to robustly solve many optimization problems. Additionally, the fact that the best members of a population carry forward to subsequent iterations is key to solving optimization problems. Horie and Conway were able to show that they are also very useful for finding good initial solutions to differential games [10]. For example, the authors use a genetic algorithm to provide an initial solution for the classical Homicidal Chauffeur problem [1]. The initial solution is then used with their semi-direct method for solving differential games via a NLP tool.

2. Example: Homicidal Chauffeur Problem

A well known differential game is the Homicidal Chauffeur game first proposed by Isaacs [1]. In this game, a “deranged” chauffeur is attempting to run over a pedestrian. The chauffeur, driving a car, plays the role of a less agile, but faster pursuer. The pedestrian has more agility but is not nearly as fast as the pursuer. The game ends when the chauffeur has successfully driven the car within a prescribed radius of the evader. The problem will now be formally developed.

The pursuer and evader have maximum speed of s_p and s_e, respectively. The pursuer motion in the x-y plane is controlled by a turning rate governed by a control. Whereas, the evader motion is controlled by a turning rate that can be changed instantaneously.

A relative coordinate model can be used to simplify the problem with the following system dynamics.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \left(-\frac{s_p}{R}w\right) y + s_e \sin v \\ \left(\frac{s_p}{R}w\right) x - s_p + s_e \cos v \end{bmatrix} \tag{3.11}$$

In the preceding equation, w is the turning angle of the pursuer, v is the turning angle of the evader, and R is the turning radius of the pursuer. The objective function for the homicidal chauffeur problem is simply the time to capture. The evader is trying to maximize the time to capture, while the pursuer is trying to minimize it. Additionally a soft constraint is added to penalize the terminal miss from the capture radius.

$$J = \phi(x(t_f), y(t_f), t_f) + t_f \tag{3.12}$$

$$\phi(x(t_f), y(t_f), t_f) = x(t_f)^2 + y(t_f)^2 - R_{cap}^2 \tag{3.13}$$

In order to find the necessary conditions we proceed as customary. The Hamiltonian is first formed.

$$\begin{aligned} H &= \boldsymbol{\lambda}^T f + L \\ &= \lambda_x \left(\left(-\frac{s_p}{R} w \right) y + s_e \sin v \right) \\ &\quad + \lambda_y \left(\left(\frac{s_p}{R} w \right) x - s_p + s_e \cos v \right) + 1 \end{aligned} \tag{3.14}$$

The adjoint variables are then governed as follows.

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}} \tag{3.15}$$

$$\dot{\lambda_x} = -\lambda_y \left(\frac{s_p}{R} w \right) \tag{3.16}$$

$$\dot{\lambda_y} = -\lambda_x \left(-\frac{s_p}{R} w \right) \tag{3.17}$$

$$\boldsymbol{\lambda}(t_f) = \left. \frac{\partial \phi}{\partial \mathbf{x}} \right|_{t_f} \tag{3.18}$$

$$\lambda_x(t_f) \;=\; 2x(t_f) \tag{3.19}$$

$$\lambda_y(t_f) \;=\; 2y(t_f) \tag{3.20}$$

Note that the preceding equations can be simplified as follows.

$$x(t_f)\lambda_y(t_f) = y(t_f)\lambda_x(t_f) \tag{3.21}$$

Horie and Conway use a genetic algorithm to generate initial conditions for their semi-direct method [10]. They solve Isaacs' homicidal chauffeur problem for a particular set of initial conditions. The following results duplicate the work of Horie and Conway in order to provide further verification that a genetic algorithm can be successfully used to provide a good enough initial guess for a semi-direct NLP solution. The problem parameters can be found in the table (I).

Table I. Homicidal Chauffeur Problem Parameters

Parameter Description	Abbreviation	Value
Chauffeur speed	s_p	1
Evader speed	s_e	0.1
Chauffeur turning rate	R	1
Capture Radius	R_{cap}	0.8

Figure (2) illustrates the numerical solution to the homicidal chauffeur problem. The initial genetic algorithm results are very similar to those found by Horie and Conway in [10]. The numerical results presented in figure (2) show that the initial solution found by the genetic algorithm is refined when used by the semi-direct approach. Additionally, the results produced by the semi-direct approach approach the optimal, analytical solution described by Isaacs in [1].

This section provides additional verification that a genetic algorithm can be used to find an initial solution to classic differential game. Additionally, the results show that the semi-direct method finds a solution consistent with analytical approaches for a problem with relatively simple system dynamics.

C. Blocking Problem

The section describes the blocking problem game posed by Issacs [1]. In this game, the evader is attempting to maximize its vertical position, while the pursuer desires

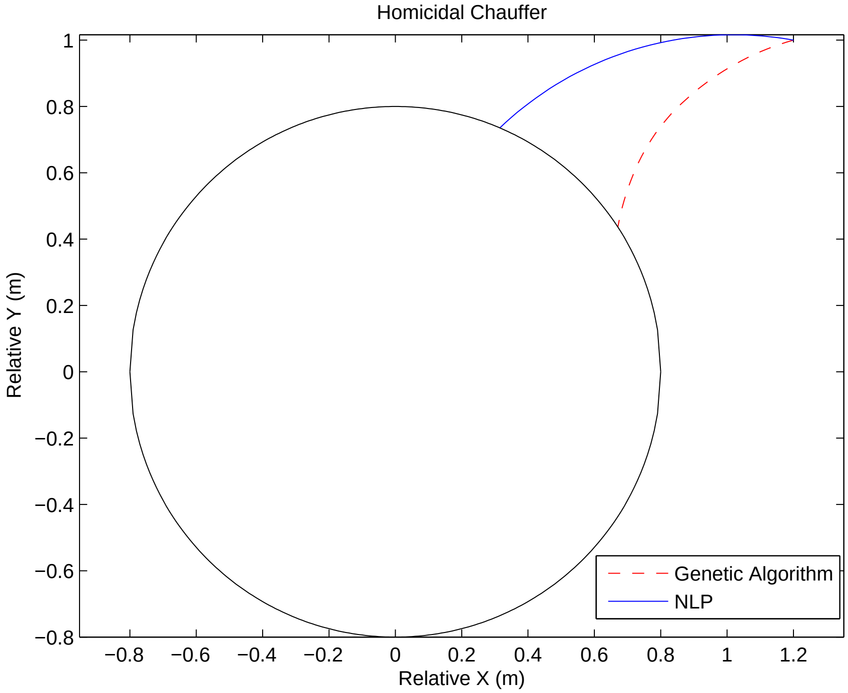

Fig. 2. Solution to Homicidal Chauffeur Problem

to minimize the advance. Additionally, the final time is fixed and a capture radius is specified. At the end of the game the pursuer and evader must be exactly a capture radius apart. Figure (3) illustrates this differential game. The pursuer and evader dynamics are respectively governed by the following set of equations.

$$\begin{aligned}\dot{x_p} &= \sin w \\ \dot{y_p} &= -\cos w\end{aligned}$$

$$\begin{aligned}\dot{x_e} &= -\sin v \\ \dot{y_e} &= \cos v\end{aligned}$$

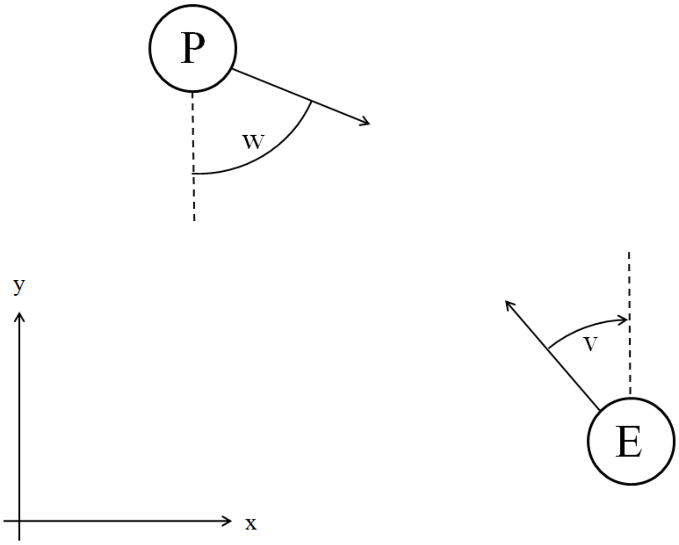

Fig. 3. Blocking Problem

To eliminate an unneeded system state, a relative state x is used to describe the relative positions between evader and pursuer in the x direction.

$$
\begin{aligned}
x &\equiv x_e - x_p \qquad (3.22)\\
\dot{x} &= -\sin v - \sin w \qquad (3.23)
\end{aligned}
$$

The system states are then reduced to the following.

$$\dot{y_p} = -\cos w \qquad (3.24)$$

$$\dot{y_e} = \cos v \qquad (3.25)$$

$$\dot{x} = -\sin v - \sin w \qquad (3.26)$$

Since the evader wishes to maximize the advancement of the evader in the y direction, y_e, and the pursuer wishes to minimize y_e, the system objective function is given as follows.

$$J = \int_{t_0}^{t_f} \cos v dt \qquad (3.27)$$

The solution to the differential game is then to find controls that specify a minimax solution of Eq. (3.27).

In order to find an optimal solution to the problem, necessary conditions are used. The development is similar to previous problems, except a terminal condition is specified. This terminal condition is a function of system states. The addition of a hard constraint alters the methodology for finding final time values for the adjoint variables, and is described below.

The Hamiltonian and costate variables follow.

$$\begin{aligned} H &= \cos v + \boldsymbol{\lambda}^T \dot{\mathbf{x}} \\ &= \cos v - \lambda_1 \cos w + \lambda_2 \cos v - \lambda_3 (\sin v + \sin w) \end{aligned} \tag{3.28}$$

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}} \tag{3.29}$$

The costate variables are shown to be constant by evaluating Eq. (3.29).

$$\dot{\lambda}_1 = 0 \tag{3.30}$$

$$\dot{\lambda}_2 = 0 \tag{3.31}$$

$$\dot{\lambda}_3 = 0 \tag{3.32}$$

The optimality conditions take the following form.

$$\frac{\partial H}{\partial w} = 0 \tag{3.33}$$

$$\frac{\partial H}{\partial v} = 0 \tag{3.34}$$

The preceding equations then yield the following relationships for the adjoint variables and the controls.

$$\lambda_1 \sin w - \lambda_3 \cos w = 0 \tag{3.35}$$

$$-(\sin v(1 + \lambda_2) + \lambda_3 \cos v) = 0 \tag{3.36}$$

Manipulating the above expression, new expressions for the player controls, w and v, are found.

$$\tan w = \frac{\lambda_3}{\lambda_1} \tag{3.37}$$

$$\tan v = -\frac{\lambda_3}{1 + \lambda_2} \tag{3.38}$$

Since a semi-direct approach will be used to solve the problem, one set of controls must be explicitly defined. Eq. (3.38) is used to evaluate the evader's controls.

$$v \equiv -atan2(\lambda_3, 1 + \lambda_2) \tag{3.39}$$

Terminal conditions must also be specified for the adjoint variables. In previous developments, the state variables were free at the final time and penalized via soft constraints. In this problem, a hard constraint is enforced at the final time; the players must be a capture radius R_{cap} away from one another at the final time.

The hard constraint ψ is illustrated in figure (4) and expressed formally by the following equation.

$$\psi(\mathbf{x}(t_f), t_f) = 0 = x(t_f)^2 + (y_p(t_f) - y_e(t_f))^2 - R_{cap}^2 \tag{3.40}$$

When hard constraints are present, the following equation is used to determine the final time values for adjoint variables [5].

$$\boldsymbol{\lambda}(t_f) = \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\psi}}{\partial \mathbf{x}}|_{t_f} \tag{3.41}$$

Here, $\boldsymbol{\nu}$ is a vector of free parameters that must be determined. Eq. (3.41) can be

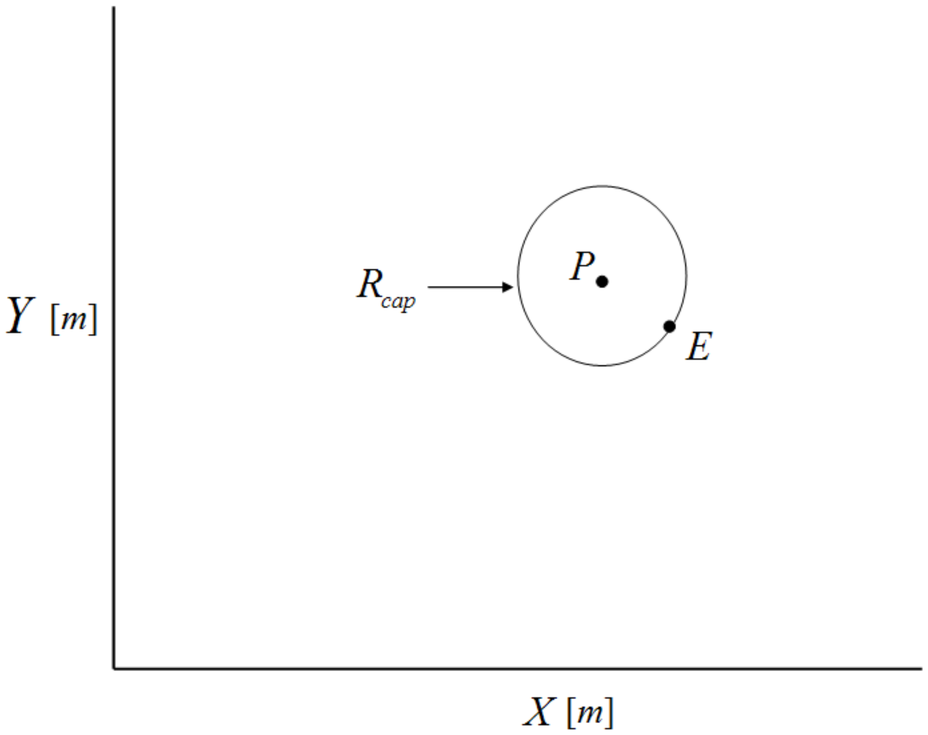

Fig. 4. Hard Constraint for Blocking Problem

expanded into three equations.

$$\lambda_1(t_f) = \nu_1(y_p(t_f) - y_e(t_f)) \tag{3.42}$$

$$\lambda_2(t_f) = \nu_2(y_e(t_f) - y_p(t_f)) \tag{3.43}$$

$$\lambda_3(t_f) = \nu_3 x(t_f) \tag{3.44}$$

The preceding development was used to solve the blocking problem using both a closed form and a semi-direct approach. The closed form solution incorporates all of the preceding necessary conditions. The semi-direct approach minimizes the objective function given in Eq. (3.27) subject to the state evolution equation, the adjoint variable specific to the evader's y position, the adjoint variable for the relative x position, and the optimality condition for the evader. Since the pursuers controls are found numerically by a nonlinear programming tool, λ_1 does not need to be found by the numerical solver. Additionally the optimality condition for the pursuer's control, Eq. (3.35) is not used. A slightly perturbed version of this closed form solution was used as input to a semi-direct method. The numerical tool FMINCON was then used to solve the problem in a semi-direct fashion [8].

The blocking problem was solved for the initial conditions given in table (II). Other simulation parameters are given in table (III).

Table II. Blocking Problem Initial Conditions

Player	X position (m)	Y position (m)
Pursuer	0.00	10.00
Evader	1.00	0.00

Table III. Blocking Problem Simulation Parameters

Parameter Description	Abbreviation	Value
Capture Radius	R_{cap}	2 m
Final Time	t_f	4.773 s

The closed form solution and the NLP solution match extremely well. A summary of results from both approaches is presented in table (IV). The values for the pursuer control and the evader control, $w(t)$ and $v(t)$, respectively were constant throughout the simulation, and are listed in the table as a single value.

A visual illustration of the results from the closed form solution is presented in figure (5). This figure illustrates the evader moving in the positive y direction before capture by the pursuer at the final time.

Table IV. Blocking Problem Simulation Results

Parameter	Closed Form	Semi-direct
Pursuer control $(w(t))$	0.524 rad	0.547 rad
Evader Control $(v(t))$	-0.524 rad	-0.522 rad
Final Time Vertical Position $(y_e(t_f))$	4.134 m	4.137 m

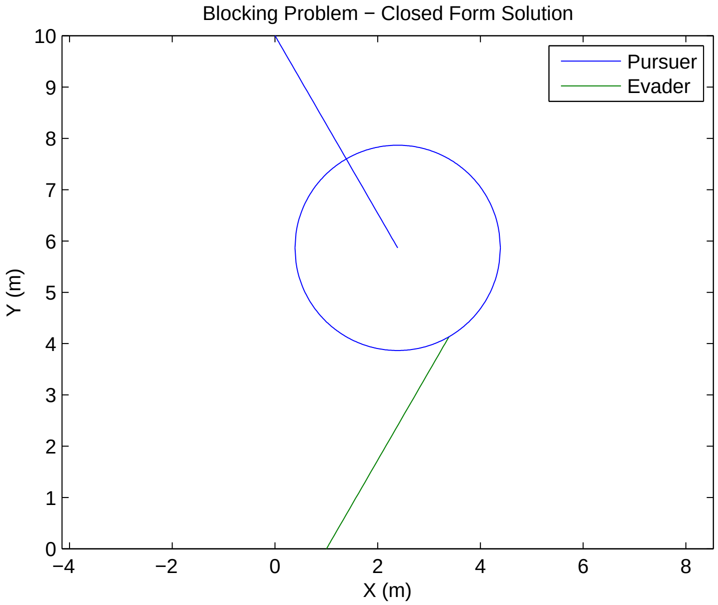

Fig. 5. Blocking Problem Closed Form Simulation Results

Figure (6) illustrates the results from the semi-direct solution. The numerical solution which uses only some of the necessary conditions, was able to nearly duplicate the results of the closed form solution. These results give further verification that the semi-direct method can be successfully used to solve differential games that are unsolvable by direct NLP techniques.

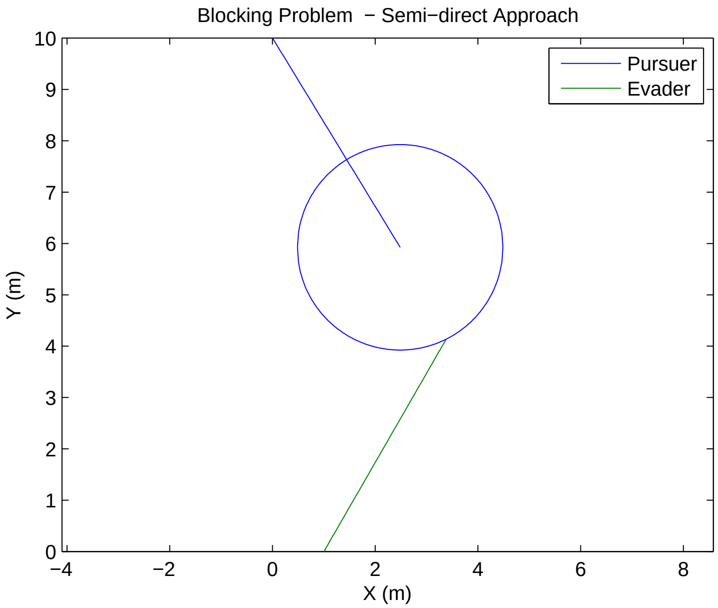

Fig. 6. Blocking Problem NLP Simulation Results

CHAPTER IV

SEMI-DIRECT METHOD WITH STATE INEQUALITY CONSTRAINTS

This chapter formally introduces a technique to extend the Horie and Conway's semi-direct method for solving pursuit-evasion games to a class of state inequality constraints. Several illustrative examples are provided to demonstrate the new technique.

Chapter III includes a detailed introduction to Horie and Conway's semi-direct method, a technique that uses nonlinear programming tools developed for numerical optimization problems to solve differential games. The method uses some of the analytical necessary conditions to directly specify controls for one player. This choice allows a nonlinear programming tool to then directly extremize the system objective function. Without any additional information, such as necessary conditions, traditional nonlinear programming techniques can not find the solution to the game.

One limitation of the semi-direct method is that it does not handle state inequality constraints. This limits the reach of the method from many problems that enforce conditions on state variables. One such classical problem is the football problem posed by Breakwell and Merz [11]. The football problem is solved using the method for a few illustrative cases after the extension method is introduced.

A. Method Development

In this section, a general dynamic system is used to introduce an extension to the semi-direct method for state inequality constraints. The system states are denoted as $\mathbf{x}$. The system controls for the pursuer and evader are $\mathbf{w}$ and $\mathbf{v}$, respectively. The system dynamics and objective function are as follows.

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{w}, \mathbf{v}, t) \tag{4.1}$$

$$J = \phi(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{x}, \mathbf{w}, \mathbf{v}, t)dt \tag{4.2}$$

Now consider the case of some state variables having a lower and upper bound.

$$\beta \leq x_i \leq \alpha \tag{4.3}$$

This form of state inequality constraint bounds the *ith* state variable to be greater than β and less than α. Let S represent an inequality constraint as a function of the system state variables $\mathbf{x}$ and time t.

$$S(\mathbf{x}, t) \leq 0 \tag{4.4}$$

An equivalent representation to the class of state constraints presented in Eq. (4.3) can be written as follows.

$$S_1 \equiv x_i - \alpha \leq 0 \tag{4.5}$$

$$S_2 \equiv -x_i + \beta \leq 0 \tag{4.6}$$

Bryson and Ho introduce methods for solving optimization problems with state inequality constraints [5]. For the class of state inequality constraints assumed in Eq. (4.3), the necessary conditions for the optimal control problem must be developed. First, form the Hamiltonian as follows.

$$H = L(\mathbf{x}, \mathbf{w}, \mathbf{v}, t) + \boldsymbol{\lambda}^T f(\mathbf{x}, \mathbf{w}, \mathbf{v}, t) + \boldsymbol{\mu}^T S^{(q)} \tag{4.7}$$

In this expression, L is the integrand of the objective function given in Eq. (4.2), f describes how the system evolves, and $\boldsymbol{\lambda}$ are adjoint variables. $S^{(q)}$ is the *qth* time derivative of S where the controls of the system first appear. It is found by taking consecutive time derivatives of S, substituting $f(\mathbf{x}, \mathbf{w}, \mathbf{v}, t)$ for $\dot{\mathbf{x}}$ where appropriate, and stopping when the controls of the system first appear. When a constraint boundary is encountered $S = 0$ and $S^{(q)} = 0$. The function $\boldsymbol{\mu}(t)$ only influences the system when a constraint is encountered, it is zero when a constraint is inactive.

As was the case for problems without state constraints, the adjoint equations are governed as follows.

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}} \tag{4.8}$$

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial L}{\partial \mathbf{x}} - \boldsymbol{\lambda}^T \frac{\partial f}{\partial \mathbf{x}} - \boldsymbol{\mu}^T \frac{\partial S^{(q)}}{\partial \mathbf{x}} \tag{4.9}$$

For brevity, let a subscript $\mathbf{x}$ denote a partial derivative with respect to $\mathbf{x}$. The previous expression then can be written as follows.

$$\dot{\boldsymbol{\lambda}} = -L_{\mathbf{x}} - \boldsymbol{\lambda}^T f_{\mathbf{x}} - \boldsymbol{\mu}^T {S^{(q)}}_{\mathbf{x}} \tag{4.10}$$

Note that since $\boldsymbol{\mu}(t)$ is 0 when the player is not on the inequality boundary, the above expression holds for all time.

The optimality conditions for this problem is broken into two parts, one for each player.

$$0 = H_{\mathbf{w}} = L_{\mathbf{w}} + \boldsymbol{\lambda}^T f_{\mathbf{w}} + \boldsymbol{\mu}^T {S^{(q)}}_{\mathbf{w}} \tag{4.11}$$

$$0 = H_{\mathbf{v}} = L_{\mathbf{v}} + \boldsymbol{\lambda}^T f_{\mathbf{v}} + \boldsymbol{\mu}^T {S^{(q)}}_{\mathbf{v}} \tag{4.12}$$

The final time constraints for $\boldsymbol{\lambda}$ are then given as follows.

$$\boldsymbol{\lambda}(t_f) = \left.\frac{\partial \phi}{\partial \mathbf{x}}\right|_{t_f} \tag{4.13}$$

The addition of this class of state inequality constraints can now be incorporated into the semi-direct approach. One way to incorporate the inequality constraints is to break the problem down into several sections, those when the constraint boundary is active and those where the constraint boundary is inactive. The numerical optimization tool is tasked with solving for adjoint variables for one player and free controls for the other. The addition of state inequality constraints requires the numerical solver to provide estimates for the influence functions.

The following sections provide several examples of solving differential games with state inequality constraints. The system is introduced, analytical necessary conditions are developed, and a numerical simulation is then provided.

B. Example: Equispeed Football Problem

In this game, a single pursuer is chasing an evader. The evader is attempting to maximize its vertical position, while the pursuer desires to capture the evader. Additionally, the pursuer is trying to minimize the final distance between players while the evader is trying to maximize this distance. Each player's horizontal position is bounded. The game models American football, where the offensive player is trying to get as far "up the field" as possible while staying in bounds. This problem was solved via an analytical approach by Breakwell and Merz [11].

The speed of each player is identical and prescribed. Each player's only control is a turn angle. An illustration of the game is provided in figure (7). The pursuer and evader states are governed by the following equations.

$$\dot{x_p} = \sin w \tag{4.14}$$

$$\dot{y_p} = -\cos w \tag{4.15}$$

$$\dot{x_e} = -\sin v \tag{4.16}$$

$$\dot{y_e} = \cos v \tag{4.17}$$

In these sets of equations, w is the control available to the pursuer, and v is the control available to the evader. These controls represent the turning angle of each

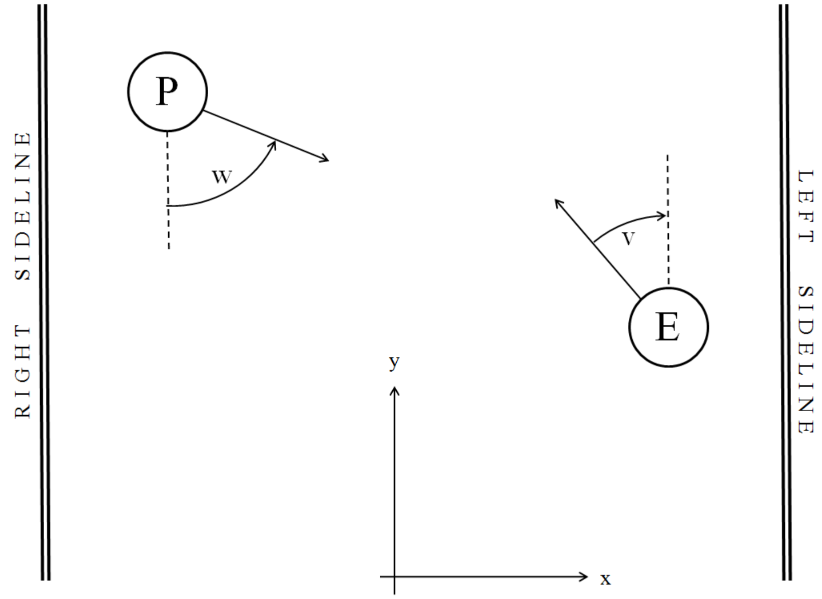

Fig. 7. Football Problem

respective player.

The objective function for the system is given as follows.

$$J = \frac{1}{2}\left((x_p(t_f) - x_e(t_f))^2 + (y_p(t_f) - y_e(t_f))^2\right) + \int_{t_0}^{t_f} \cos v dt \tag{4.18}$$

The integral portion of Eq. (4.18) represents the upfield position of the evader while the soft constraint portion of the equation represents the final time distance between the players. The evader seeks to maximize J, while the pursuer attempts to minimize it.

For this example, each player's x coordinate position is bound between -1 and 1. As a result, we have four simple state inequality constraints that bound the player's motion.

$$S_1(x_p, t) \leq 0; \qquad x_p - 1 \leq 0 \tag{4.19}$$

$$S_2(x_p, t) \leq 0; \qquad -x_p - 1 \leq 0 \tag{4.20}$$

$$S_3(x_e, t) \leq 0; \qquad x_e - 1 \leq 0 \tag{4.21}$$

$$S_4(x_e, t) \leq 0; \qquad -x_e - 1 \leq 0 \tag{4.22}$$

As previously outlined, It is necessary to find $S_i^{(q)}$ for each state inequality constraint. $S_i^{(q)}$ is the qth time derivative where the controls of the system first appear.

$$S_1^{(1)} = \sin w \tag{4.23}$$

$$S_2^{(1)} = -\cos w \tag{4.24}$$

$$S_3^{(1)} = -\sin v \tag{4.25}$$

$$S_4^{(1)} = \cos v \tag{4.26}$$

All of the inequality constraints for this example are first order inequality constraints, since only a single derivative is taken before the system controls appear. The analytic necessary conditions for the solution of this differential game follow below. The extension of the semi-direct method is used to incorporate the preceding state inequality constraints into the football problem. This development uses the assumption that the evader's control will be eventually found using necessary conditions and that J will then be minimized. An equally valid solution method would be to substitute for the pursuer's controls and then maximize the system objective function.

We first form the Hamiltonian.

$$H = L + \boldsymbol{\lambda}^T f(\mathbf{x}, w, v, t) + \boldsymbol{\mu}^T \mathbf{S}^{(q)} \tag{4.27}$$

$$= \cos v + \lambda_1 \sin w - \lambda_2 \cos w - \lambda_3 \sin v + \lambda_4 \cos v + \tag{4.28}$$

$$+ \mu_1 \sin w - \mu_2 \cos w - \mu_3 \sin v + \mu_4 \cos v \tag{4.29}$$

The costate equations are as follows.

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{x}} \tag{4.30}$$

$$\dot{\lambda}_1 = 0 \tag{4.31}$$

$$\dot{\lambda}_2 = 0 \tag{4.32}$$

$$\dot{\lambda}_3 = 0 \tag{4.33}$$

$$\dot{\lambda}_4 = 0 \tag{4.34}$$

Optimality conditions are given by the following equations.

$$\frac{\partial H}{\partial w} = 0 \tag{4.35}$$

$$\frac{\partial H}{\partial v} = 0 \tag{4.36}$$

$$(\lambda_1 + \mu_1) \cos w + (\lambda_2 + \mu_2) \sin w = 0 \tag{4.37}$$

$$\sin v (1 + \lambda_4 + \mu_4) + (\lambda_3 + \mu_3) \cos v = 0 \tag{4.38}$$

Rearranging to solve for w and v yields the following expressions.

$$\tan w = \frac{-(\lambda_1 + \mu_1)}{(\lambda_2 + \mu_2)} \tag{4.39}$$

$$\tan v = \frac{-(\lambda_3 + \mu_3)}{1 + \lambda_4 + \mu_4} \tag{4.40}$$

In order to solve the problem using a nonlinear programming tool, Eq. (4.40) is used for the evader's control. That is, v is defined by the following equation.

$$v \equiv -\text{atan2}(\lambda_3 + \mu_3, 1 + \lambda_4 + \mu_4) \tag{4.41}$$

Final time values for the adjoint variables then follow.

$$\boldsymbol{\lambda}(t_f) = \left.\frac{\partial \phi}{\partial \mathbf{x}}\right|_{t_f} \tag{4.42}$$

$$\lambda_1(t_f) = x_p(t_f) - x_e(t_f) \tag{4.43}$$

$$\lambda_2(t_f) = y_p(t_f) - y_e(t_f) \tag{4.44}$$

$$\lambda_3(t_f) = -\left(x_p(t_f) - x_e(t_f)\right) \tag{4.45}$$

$$\lambda_4(t_f) = -\left(y_p(t_f) - y_e(t_f)\right) \tag{4.46}$$

Utilizing the Horie and Conway semi-direct approach, Equations (4.14),(4.15), (4.16), (4.17), (4.33), (4.34), (4.45), (4.46), and (4.41) are used to minimize the system objective function specified in Eq. (4.18).

The football problem developed in the previous section was simulated using the initial conditions given in table (V). An initial estimate for the solution was generated using a genetic algorithm. The initial solution served as an input to the nonlinear programming solver, FMINCON.

Table V. Equispeed Football Problem Parameters

Player	X position (m)	Y position (m)
Pursuer	0.00	1.00
Evader	0.52	0.02

Figure (8) illustrates that the semi-direct method with inequality constraints

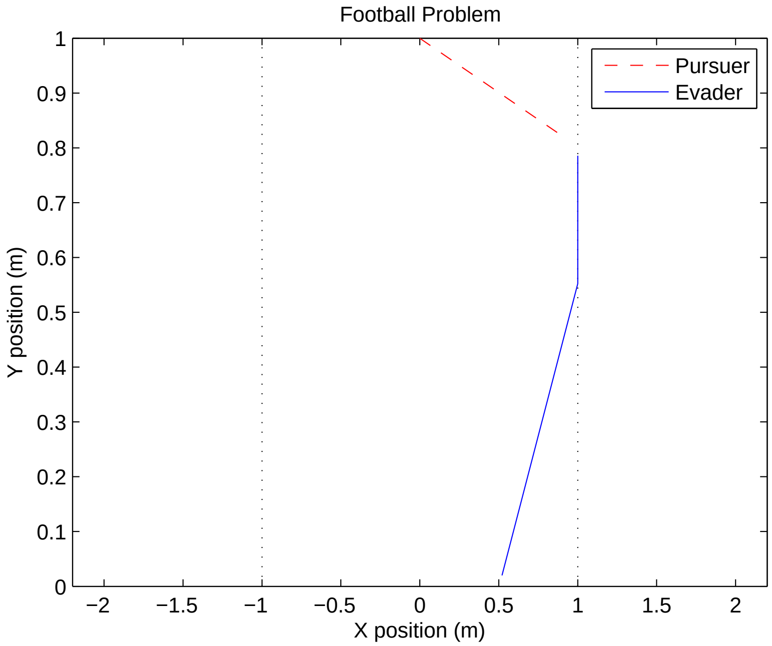

Fig. 8. Equispeed Football Problem Trajectory

approach can successfully solve the equispeed football problem. As is expected, the evader heads to the right most sideline to avoid early capture, and advances upfield until the simulation concludes. The pursuer makes a direct approach to the sideline and is within reach of the pursuer at the end of the simulation.

C. Example: Football Problem with Faster Evader

This section describes the football problem when the evader has a speed advantage. This problem is identical to that described in section (B), except that the players have no longer have equal capabilities. Much of the development for the faster evader problem is identical to the equispeed case. Key differences are identified in the following development. Again refer to figure (7) for a system illustration.

The pursuer dynamics are identical to those found in Eqns. (4.14) and (4.15). The evader dynamics are slightly modified from the equispeed case with a constant multiplier, γ, greater than unity that provides the speed advantage to the evader.

$$\dot{x_e} = -\gamma \sin v \tag{4.47}$$

$$\dot{y_e} = \gamma \cos v \tag{4.48}$$

The objective function for this game is also slightly altered from the equispeed case to reflect the speed advantage given to the evader.

$$J = \frac{1}{2}\left((x_p(t_f) - x_e(t_f))^2 + (y_p(t_f) - y_e(t_f))^2\right) + \int_{t_0}^{t_f} \gamma \cos v dt \tag{4.49}$$

The necessary conditions for the optimal solution of the game are identified as before. First the Hamiltonian is formed, and the evolution of the adjoint variables is identified. The optimality conditions and the final time values for the adjoint equations are then specified. Note that the state inequality constraints for this game are unchanged from the previous version. (See Eqns. (4.23)-(4.26))

The Hamiltonian for the system now follows.

$$H = \gamma \cos v + \lambda_1 \sin w - \lambda_2 \cos w - \lambda_3 \gamma \sin v + \lambda_4 \gamma \cos v + \tag{4.50}$$

$$+ \mu_1 \sin w - \mu_2 \cos w - \mu_3 \gamma \sin v + \mu_4 \gamma \cos v \tag{4.51}$$

The costate equations are identical to those in the equispeed game (See Eqns. (4.31) - (4.34)), as are the optimality conditions given by Eqns. (4.37) and (4.38). Also, the final time values for the adjoint equations are identical to those given in Eqns. (4.43), (4.44), (4.45), and (4.46). As before, the evader control is explicitly found by solving for v in Eq. (4.38). The evader's control is given by Eq. (4.41).

With the resulting set of equations, the semi-direct method is used to minimize the system objective function, (4.49), subject to (4.14), (4.15), (4.47), (4.48), (4.33), (4.34), (4.45), (4.46), and (4.41).

The nonlinear programming numerical solver FMINCON is used to numerically simulate this differential game. The initial positions of the players is given in table (VI). The evader is simulated to be ten percent faster than the pursuer. That is, γ is equal to 1.1.

Table VI. Faster Evader Football Problem Parameters

Player	X position (m)	Y position (m)
Pursuer	0.00	1.00
Evader	0.52	0.02

Figure (9) illustrates the results of the numerical simulation and shows that the strategies taken by the two players are similar to the equispeed case. However, the evader is able to reach a farther upfield position when it has a speed advantage over the pursuer. The evader first pursues a trajectory towards the sidelines. Once the boundary is reached, the evader then heads directly upfield until the simulation concludes. As before, the pursuer takes a direct approach towards the intersection of the two player's path.

This chapter shows how Horie and Conway's semi-direct method can be extended to the class of problems with state inequality constraints. The classical football problem, unsolved by the standard semi-direct approach, was presented to show the applicability of this new development.

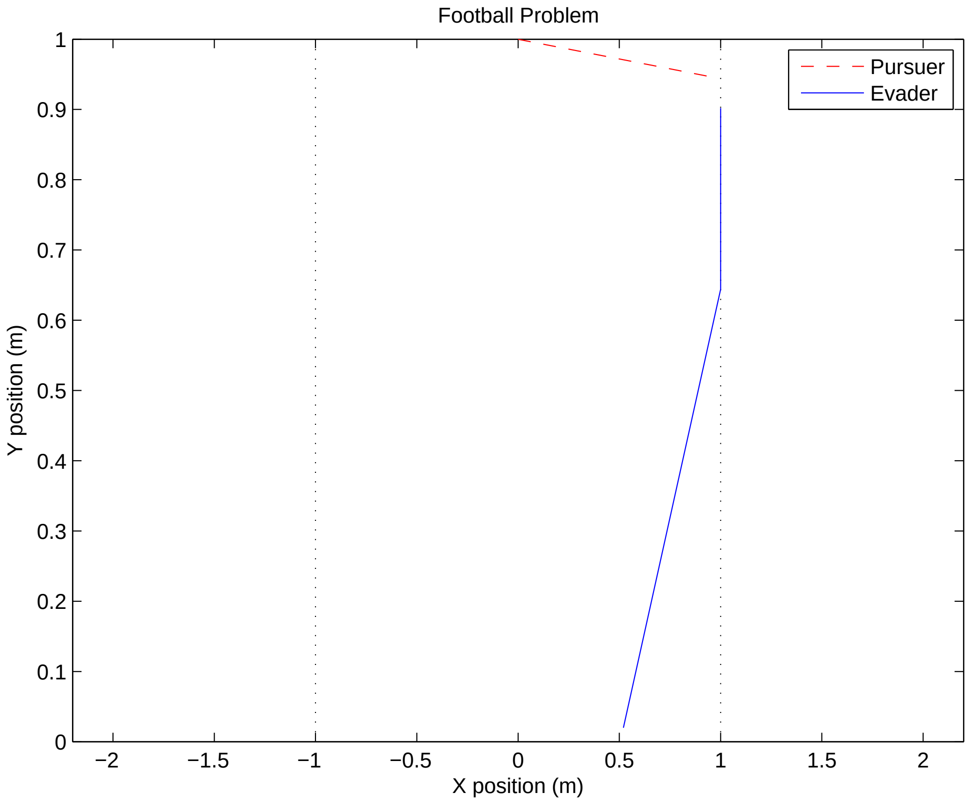

Fig. 9. Faster Evader Football Problem Trajectory

CHAPTER V

STRATEGY FOR COOPERATIVE CONTROL

If multiple pursuers are attempting to capture an evader, it is often desirable to agree upon a strategy for capture. For example, an army might perform a flanking maneuver before attacking enemy forces. While intuitive strategies for capture are easy to devise, elegant mathematical strategies are sparse.

A new and elegant cooperative control strategy is presented in this section. A relative center of mass metric is incorporated into the objective function integrand and system soft constraints. The intuition behind this strategy is that it encourages pursuers to surround an evader as soon as possible, since pursuers are trying to minimize the overall objective function. Additionally, since the center of mass relative to the pursuer is included as a soft constraint, pursuers also try to minimize this metric at the end of the game.

In general, this cooperation strategy could be used in differential games with nonlinear dynamics with a variety of system objective functions. However, to allow a straightforward feedback approach, linear system dynamics and a quadratic system objective function was assumed.

Section (A) presents a development for a feedback based solution to the cooperative pursuit and evasion game. Sections (B) - (D) present several examples of multiple pursuers attempting to capture an evader with and without the new strategy.

A. Feedback Control Development

This section presents the development of a general solution for a differential game with n pursuers and a single evader. More specifically, this class of problems involves a system objective function with an integrand consisting of quadratic components, quadratic soft constraints, and linear system dynamics.

Note that cooperative pursuit and evasion games can be studied with a variety of assumptions about state knowledge. For the games studied in this thesis, perfect knowledge of position and velocity information by all players is assumed.

The following equation describes the motion of the system.

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{5.1}$$

$$= A\mathbf{x} + C\mathbf{w} - D\mathbf{v} \tag{5.2}$$

Here, position and velocity level states are expressed relative to the evader. The objective function for the system is given by the following equation.

$$J = \phi(\mathbf{x}(t_f), t_f) + \int \frac{1}{2}\left(\mathbf{x}^T V \mathbf{x} + \mathbf{w}^T R \mathbf{w} - \mathbf{v}^T T \mathbf{v}\right) \mathrm{dt} \tag{5.3}$$

The soft constraints, ϕ, are expressed as follows, where S_1 is a matrix that penalizes terminal miss and S_2 penalizes the relative center of mass and the time rate change of the center of mass at the end of the simulation.

$$\phi = \frac{1}{2}(\mathbf{x}^T S_1 \mathbf{x})\bigg|_{t_f} + \frac{1}{2}(\mathbf{x}^T S_2 \mathbf{x})\bigg|_{t_f} \tag{5.4}$$

Recall that solving the optimal control pursuit and evasion problem includes identifying and satisfying the analytical necessary conditions of the problem. The Hamiltonian, adjoint equations, optimality conditions, and final time values for the costate variables must be constructed.

The Hamiltonian is found by appending adjoint variables multiplied to Eq. (5.2) to the integrand in Eq. (5.3).

$$H = \frac{1}{2}(\mathbf{x}^T V \mathbf{x} + \mathbf{w}^T R \mathbf{w} - \mathbf{v}^T T \mathbf{v}) + \boldsymbol{\lambda}^T (A\mathbf{x} + C\mathbf{w} - D\mathbf{v}) \tag{5.5}$$

The adjoint (costate) variables evolve as follows.

$$\begin{aligned}\dot{\boldsymbol{\lambda}} &= -\frac{\partial H}{\partial \mathbf{x}} \qquad (5.6)\\ &= -\frac{1}{2}(V+V^T)\mathbf{x} - A^T\boldsymbol{\lambda} \qquad (5.7)\end{aligned}$$

Since the matrix V is symmetric, $V = V^T$, the preceding equation can be rewritten more concisely.

$$\dot{\boldsymbol{\lambda}} = -V\mathbf{x} - A^T\boldsymbol{\lambda} \qquad (5.8)$$

The matrices R and T are also symmetric. The preceding simplifying assumption will be implicit from this point forward.

The optimality conditions and final costate values are as follows.

$$\begin{aligned}\frac{\partial H}{\partial \mathbf{w}} &= R\mathbf{w} + C^T\boldsymbol{\lambda} = 0 \qquad (5.9)\\ \frac{\partial H}{\partial \mathbf{v}} &= -T\mathbf{v} - D^T\boldsymbol{\lambda} = 0 \qquad (5.10)\end{aligned}$$

$$\begin{aligned}\boldsymbol{\lambda}(t_f) &= \left(\frac{\partial \phi}{\partial \mathbf{x}}\right)\bigg|_{t_f} \qquad (5.11)\\ &= (S_1 + S_2)\mathbf{x}\bigg|_{t_f} \qquad (5.12)\end{aligned}$$

For simplification, the control variables $\mathbf{w}$ and $\mathbf{v}$ can be expressed in terms of the costate variables.

$$\begin{aligned}\mathbf{w} &= -R^{-1}C^T\boldsymbol{\lambda} \qquad (5.13)\\ \mathbf{v} &= -T^{-1}D^T\boldsymbol{\lambda} \qquad (5.14)\end{aligned}$$

The evolution of the state and costate variables can then be expressed as follows.

$$\begin{aligned}\dot{\mathbf{x}} &= A\mathbf{x} - CR^{-1}C^T\boldsymbol{\lambda} + DT^{-1}D^T\boldsymbol{\lambda} \qquad (5.15)\\ \dot{\boldsymbol{\lambda}} &= -V\mathbf{x} - A^T\boldsymbol{\lambda} \qquad (5.16)\end{aligned}$$

These equations can also be written in matrix form.

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} A & -CR^{-1}C^T + DT^{-1}D^T \\ -V & -A^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix}$$

The matrix F is then defined to further simplify the matrix representation.

$$F = CR^{-1}C^T - DT^{-1}D^T \tag{5.17}$$

Then the preceding matrix equations is rewritten as follows.

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} A & -F \\ -V & -A^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{bmatrix} \tag{5.18}$$

Now consider a feedback solution with the assumption that the costate equations are a function of $\mathbf{x}$. Here, K is a time varying feedback matrix.

$$\boldsymbol{\lambda} = K\mathbf{x} \tag{5.19}$$

$$\dot{\boldsymbol{\lambda}} = \dot{K}\mathbf{x} + K\dot{\mathbf{x}} \tag{5.20}$$

Substituting equation set (5.18) for $\dot{\mathbf{x}}$ and $\dot{\boldsymbol{\lambda}}$ yields the following equations

$$-V\mathbf{x} - A^T K\mathbf{x} = \dot{K}\mathbf{x} + KA\mathbf{x} - KFK\mathbf{x} \tag{5.21}$$

Factoring out $\mathbf{x}$ and rearranging yields an expression for K that is independent of the system state variables.

$$\dot{K} = -V - KA - A^T K + KFK \tag{5.22}$$

Utilizing the information in Eq. (5.11) and the feedback form of $\boldsymbol{\lambda}$ in Eq. (5.20), gives the following expression for the final time values of the matrix K.

$$\begin{aligned} \boldsymbol{\lambda}(t_f) &= (K\mathbf{x})\bigg|_{t_f} && (5.23) \\ \boldsymbol{\lambda}(t_f) &= (S_1 + S_2)\mathbf{x}\bigg|_{t_f} && (5.24) \\ K(t_f) &= (S_1 + S_2)|_{t_f} && (5.25) \end{aligned}$$

The time evolution of the elements of the matrix K can be found via backwards integration of Eq. (5.22). Once the initial time values of the matrix K are found by completing the backwards integration described above, the system can be simulated forwards to solve the pursuit and evasion game. This is straightforward because Eq. (5.18) governs the evolution of the state and costate variables. Next, the relative center of mass metric is discussed.

As a final note, the relative center of mass strategy has been mentioned in this section, but not explicitly defined. The following two equations define the relative center of mass for position state variables.

$$c_{m_x} = \sum_{i=1}^{n} \frac{x_i}{n} \tag{5.26}$$

Here, c_{m_x} denotes the relative center of mass for the system in the x direction, n is the number of pursuers, and x_i is the position variable that describes the relative distance between the ith pursuer and the evader.

Similarly, the relative center of mass for the y direction is defined as follows.

$$c_{m_y} = \sum_{i=1}^{n} \frac{y_i}{n} \tag{5.27}$$

Additionally, one might be interested in the time rate change of the relative center of mass. These metrics are defined as follows.

$$\dot{c}_{m_x} = \sum_{i=1}^{n} \frac{\dot{x}_i}{n} \tag{5.28}$$

$$\dot{c}_{m_y} = \sum_{i=1}^{n} \frac{\dot{y}_i}{n} \tag{5.29}$$

In the simulations that use the new cooperation method, the relative center of mass for both the x and y directions (Eqns. (5.26)-(5.27)) were penalized in the integrand of the objective function and system soft constraints, while the time rate change of the center of mass in both directions was penalized in the system soft constraints(Eqns. (5.28) - (5.29)).

The following sections provide examples using the center of mass metric to encourage surrounding behavior. Direct comparisons between solutions that do not use the center of mass metric and those that do are also presented. As numerical results show, the inclusion of the center of mass metric into the system objective function clearly encourages surrounding behavior that is very desirable in multiplayer pursuit and evasion games.

B. Example: Two Pursuers and a Single Evader

This section provides an illustration for the case where two pursuers attempt to capture a single evader.

The initial positions for each player is given in table (VII). Pursuer positions and velocities were randomly chosen between [-50,50] m and [-0.5,0.5] m/s, respectively. For consistency, the evader position and velocity was chosen to be zero.

Figures (10), (11), and (12) illustrate the optimal solution when the center of mass metric is not included in the system objective function.

Table VII. Initial Conditions for Two Pursuers Game

Player	X pos. (m)	X vel. (m/s)	Y pos. (m)	Y vel. (m/s)
Pursuer 1	-12.84	-0.08	9.47	0.07
Pursuer 2	21.65	0.01	27.64	-0.01
Evader	0.00	0.00	0.00	0.00

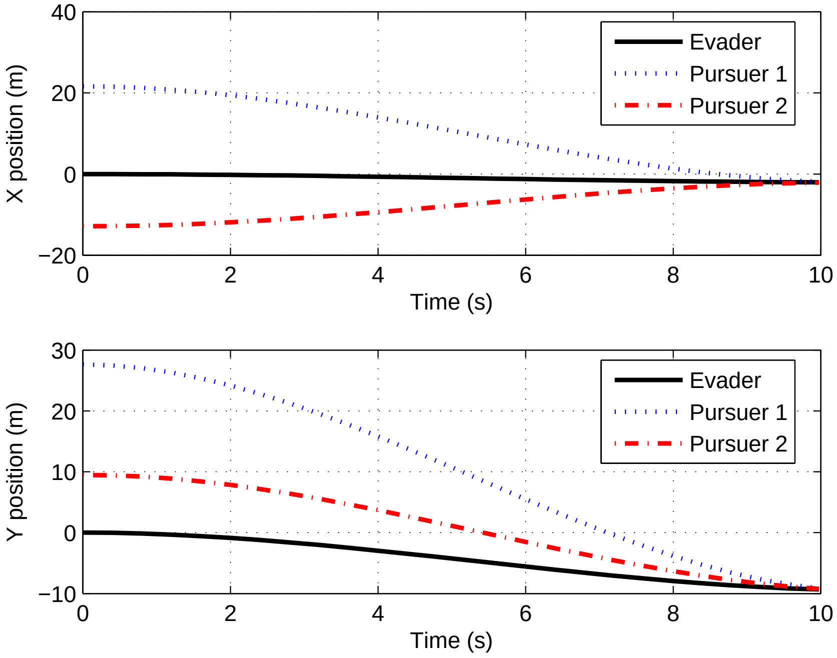

Fig. 10. Traditional Method Split View: Two Pursuers

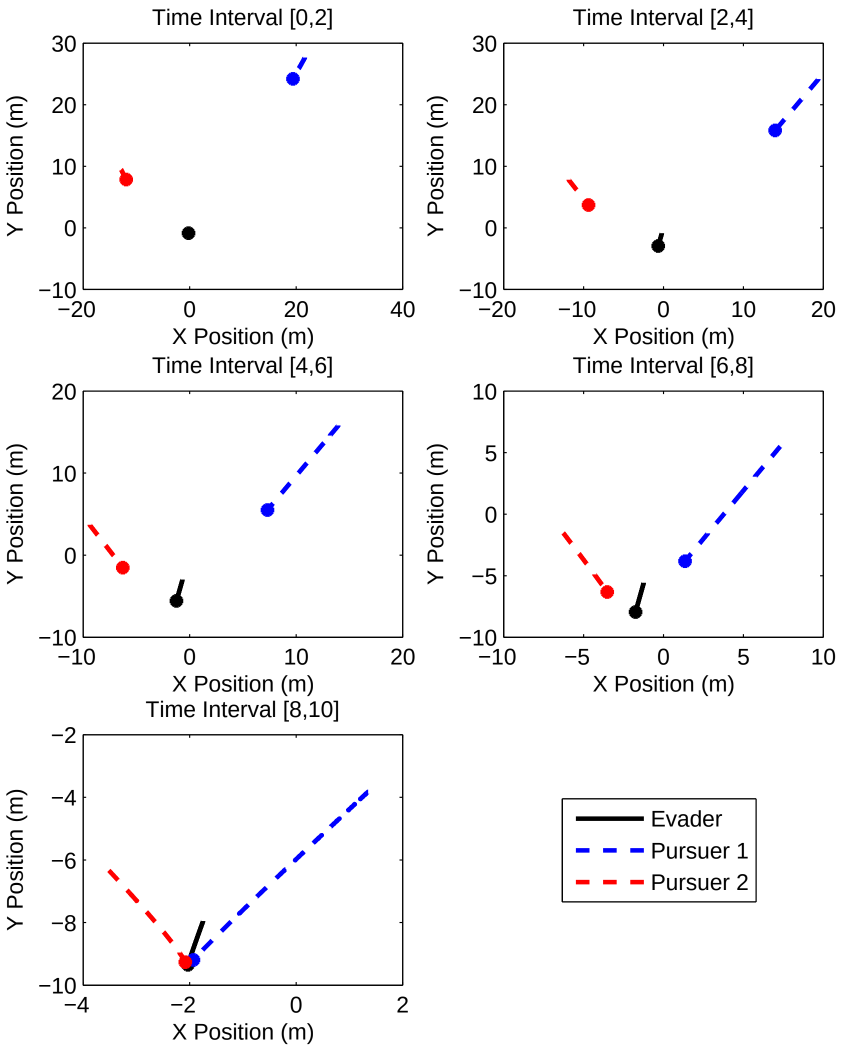

Fig. 11. Traditional Method Time Evolution: Two Pursuers

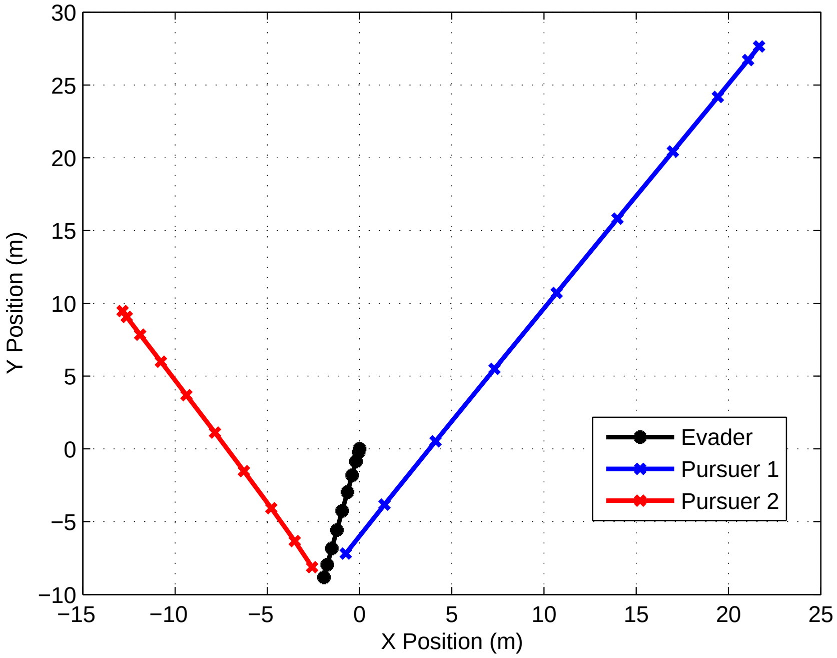

Fig. 12. Traditional Method Combined View: Two Pursuers

Numerical results show that the pursuers are able to capture the evader. While figure (11) shows that capture occurs, the pursuer players do not exhibit any observable cooperative behavior. The presence of a second pursuer aids capture, but because one pursuer is not mathematically aware of the other, true cooperation is not employed.

The same initial conditions are used again with the center of mass metric included in the objective functions. Figures (13), (14), and (15) illustrate the results of this cooperative strategy for the two pursuer, single evader case.

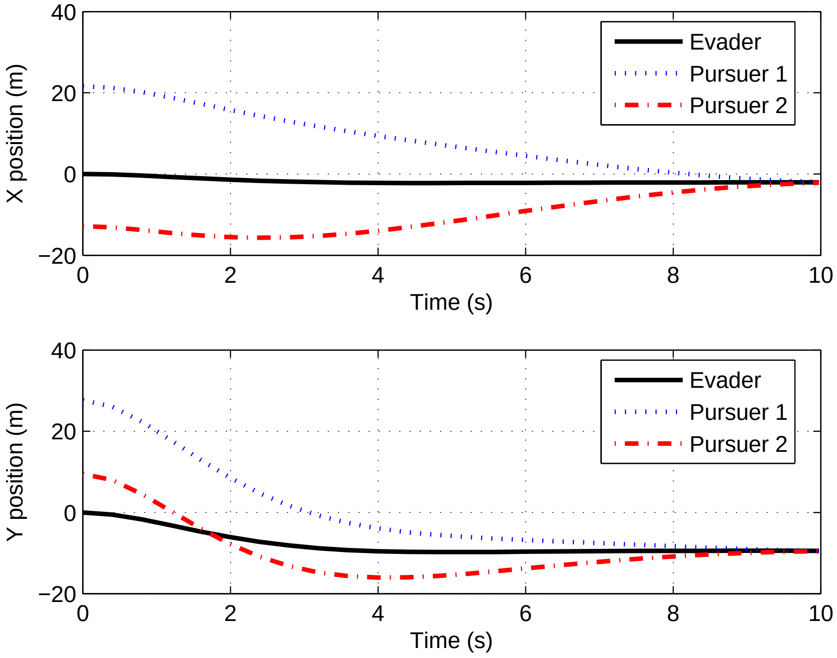

Fig. 13. New Method Split View: Two Pursuers

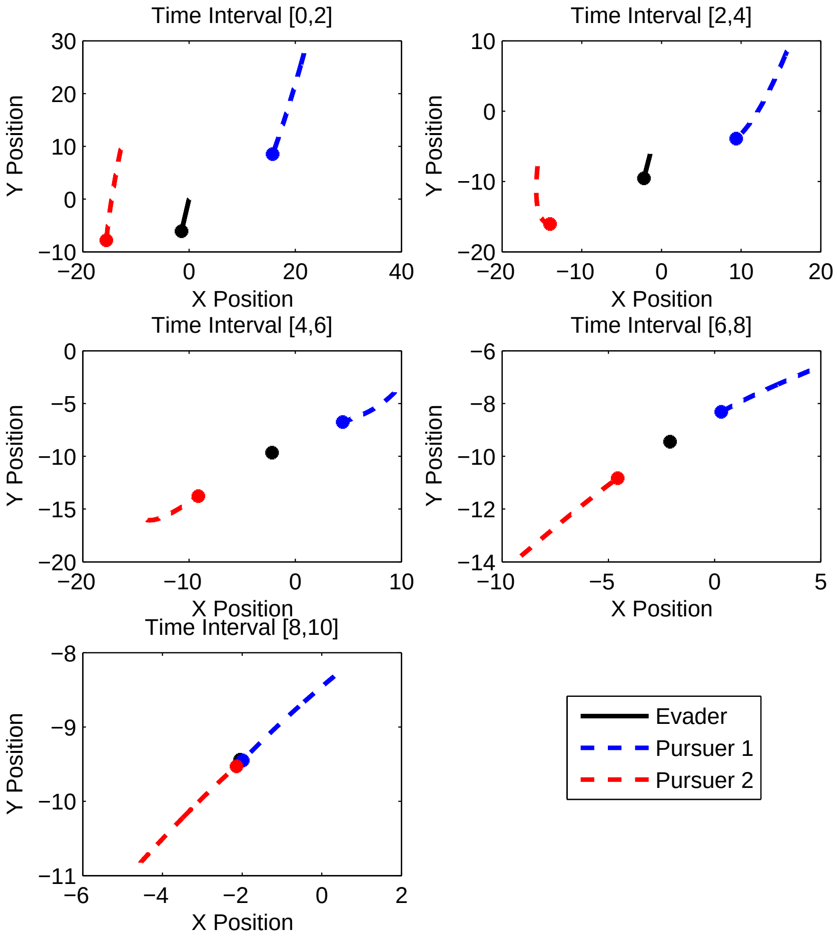

Fig. 14. New Method Time Evolution: Two Pursuers

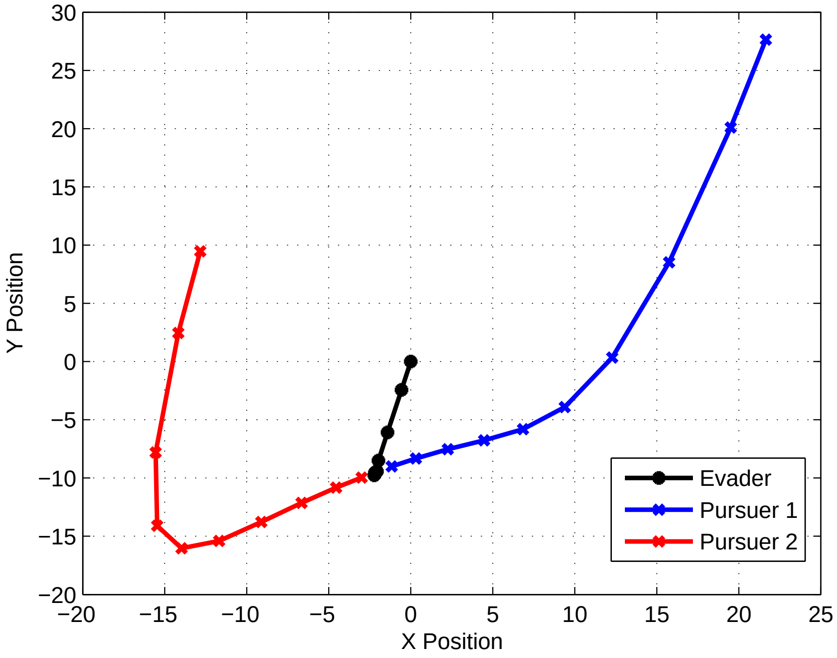

Fig. 15. New Method Combined View: Two Pursuers

While both the traditional cooperative control approach and the new approach result in capture, key differences exist in solution characteristics.

The difference between the two approaches is clear in the frame-by-frame view found in figures (11) and (14). In the traditional approach, the pursuers take a straight-line approach for the evader. In the new approach the pursuers clearly exhibit surrounding behavior. The pursuers immediately perform curving maneuvers in order to first surround, and then capture the evader. At roughly five seconds the pursuers have effectively surrounded the evader. Once the surrounding maneuver is complete, the evader is nearly stationary for the duration of the game. Once the evader is surrounded, a move in any direction leads the evader closer to capture. This behavior might be extremely useful in real world scenarios where the true intention is to destroy the evader—a stationary target is much easier to hit than a moving one.

C. Example: Three Pursuers and a Single Evader

The method previously detailed for solving a three-player pursuit and evasion games can be easily extended for additional pursuers. In this section, the case where three pursuers are attempting to capture a single evader is explored.

The initial conditions for each player are listed in table (VIII). As before, the initial positions and velocities for the pursuers are chosen randomly in the ranges $[-50, 50]m$ and $[-0.5, 0.5]m/s$, respectively. For convenience, the evader starts at the origin with zero velocity.

The differential game is simulated using traditional cooperative control techniques as well as with the new cooperation strategy. Results of the traditional approach are illustrated in figures (16), (17), and (18). Figure (16) provides a split view of the evolution of the x and y positions of the players. Figure (17) breaks the simula-

Table VIII. Initial Conditions for Three Pursuers Game

Player	X pos. (m)	X vel. (m/s)	Y pos. (m)	Y vel. (m/s)
Pursuer 1	-32.7	0.48	-22.86	-0.25
Pursuer 2	37.57	0.24	7.98	0.26
Pursuer 3	39.39	-0.30	-20.13	0.16
Evader	0.00	0.00	0.00	0.00

tion into five separate views, one for each two second interval. Figure (18) shows the position evolution of each player throughout the simulation. The tick marks represent the player's position at successive one second intervals.

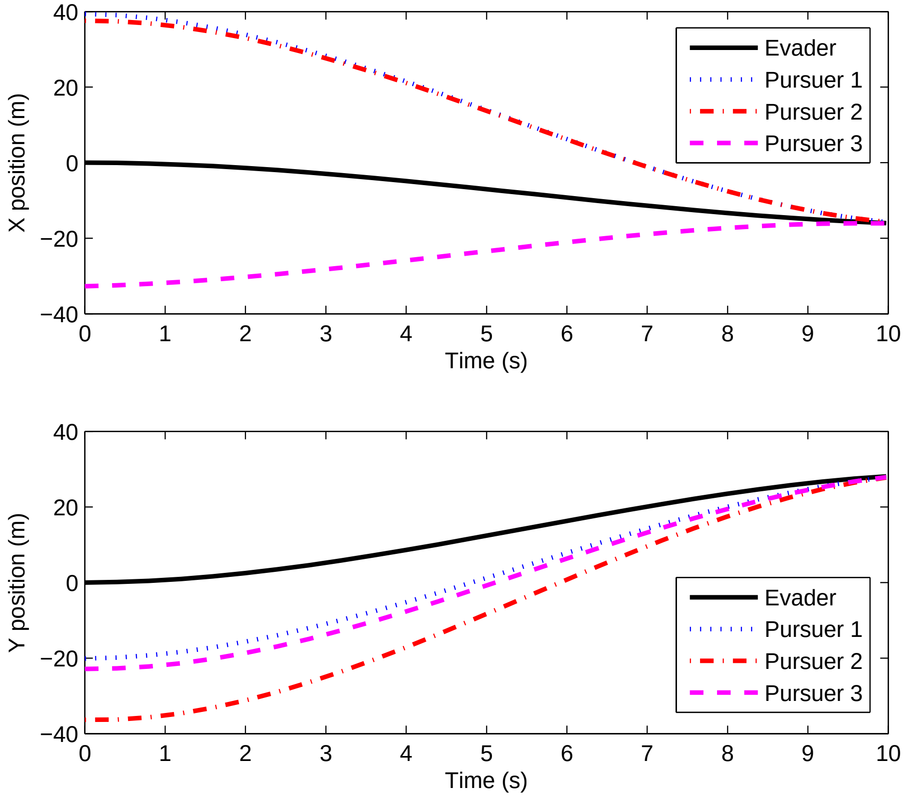

Fig. 16. Traditional Method Split View: Three Pursuers

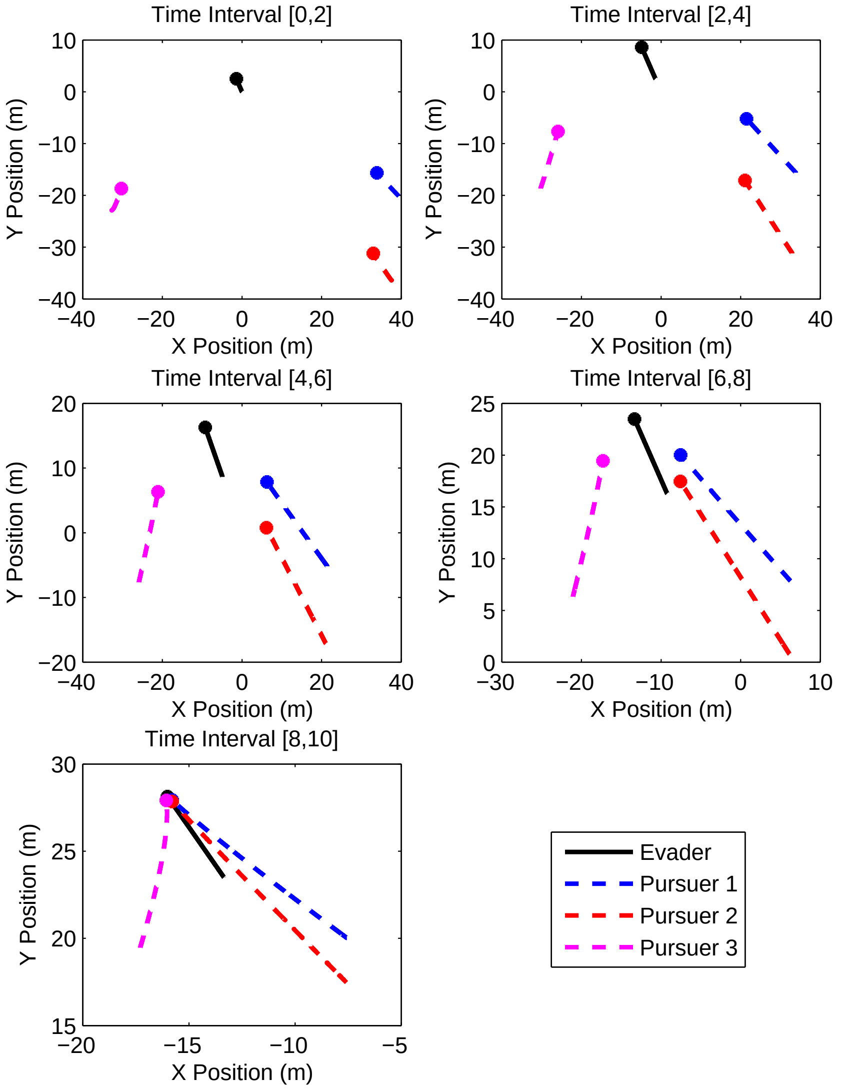

Fig. 17. Traditional Method Time Evolution: Three Pursuers

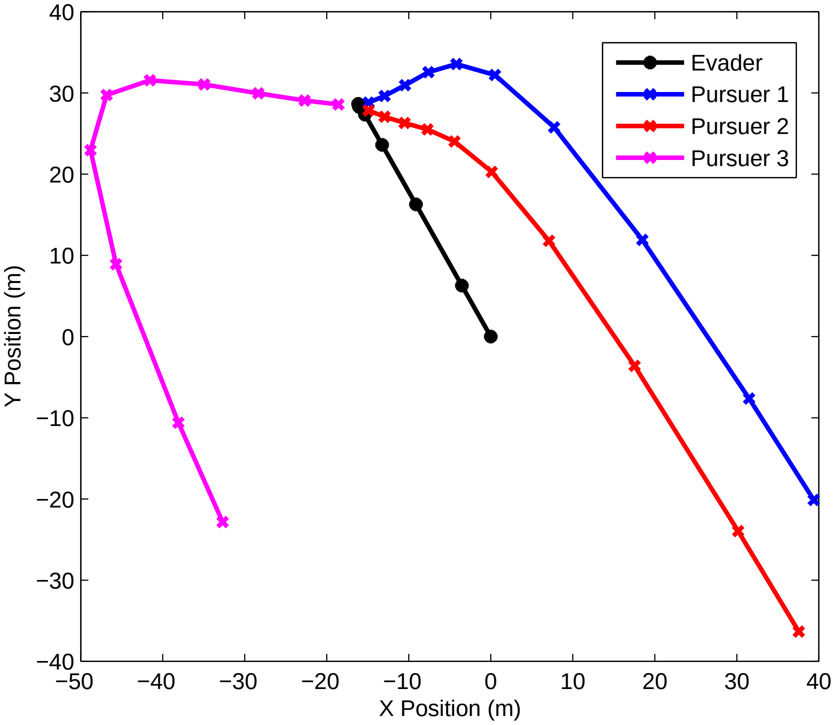

Fig. 18. Traditional Method Combined View: Three Pursuers

Results for the three pursuer problem are similar to the preceding section. The pursuers successfully chase and capture the evader, but no cooperation between pursuers is visible. Note that in the y coordinate subgraph of figure (16), all pursuers remain below the evader until capture. No strategic positioning is performed.

The problem is solved for the same initial conditions, while implementing the cooperation strategy. Results for this game are illustrated in figures (19), (20), and (21). Figure (20) shows the three pursuers working cooperatively to capture the evader. The pursuers first perform a surrounding maneuver and then close in to capture the evader. This surrounding behavior is clearly visible in the $[4, 6]$ time interval.

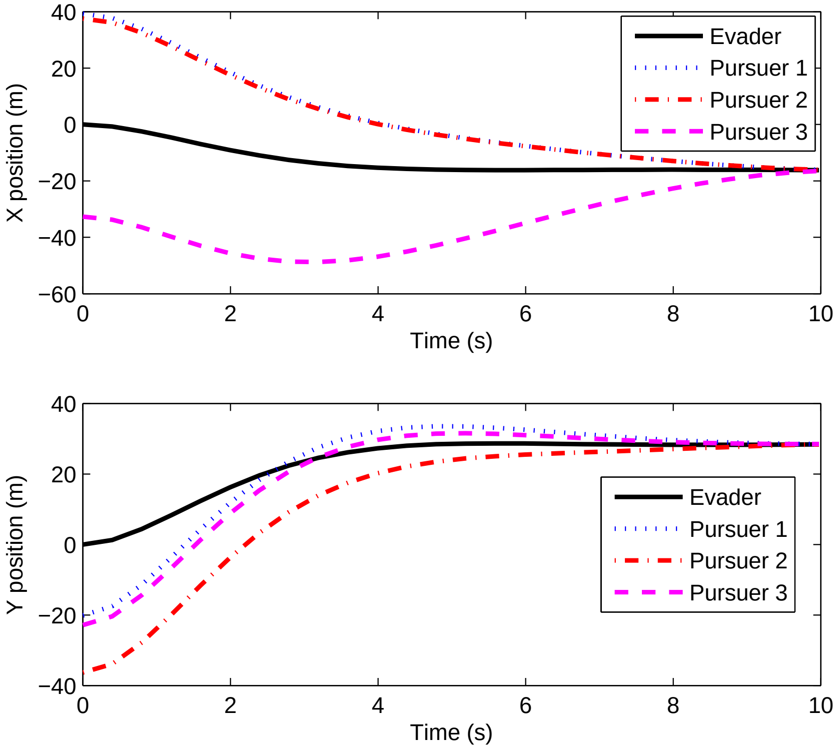

Fig. 19. New Method Split View: Three Pursuers

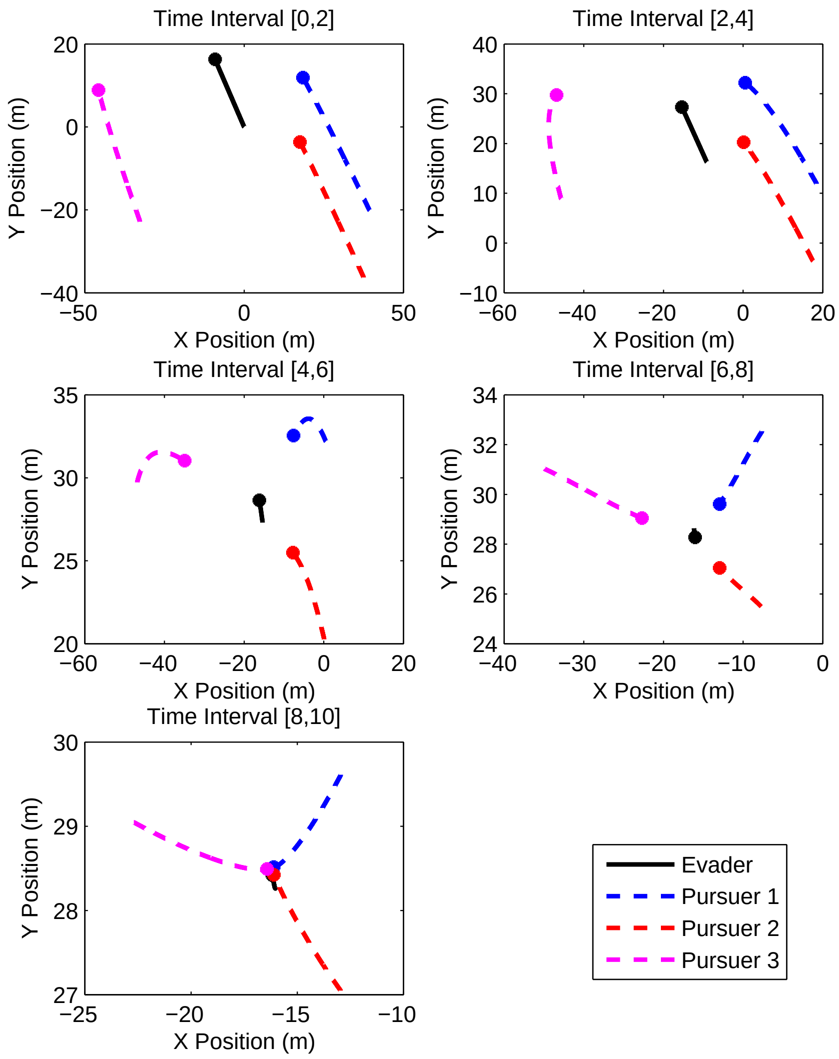

Fig. 20. New Method Time Evolution: Three Pursuers

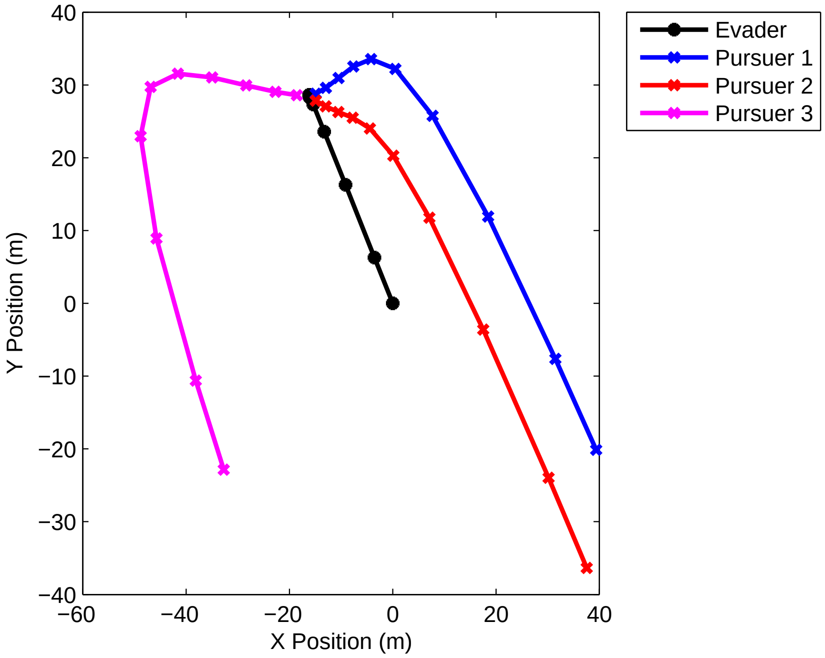

Fig. 21. New Method Combined View: Three Pursuers

As in the previous example, the evader becomes nearly motionless after the pursuers surround it. Figure (19) illustrates that the evader does not move significantly after it is surrounded. At this point there is again “nowhere to run” since any move would lead the evader closer to capture.

D. Example: Four Pursuers and a Single Evader

This section provides a final example of the new cooperation strategy. In this game, four pursuers attempt to capture an evader.

The initial positions and velocities are selected randomly in the ranges given previously and are presented in table (IX). The four pursuer problem is first solved without using the cooperation strategy. These results are illustrated in figures (22), (23), and (24).

Table IX. Initial Conditions for Four Pursuers Game

Player	X pos. (m)	X vel.(m/s)	Y pos. (m)	Y vel. (m/s)
Pursuer 1	-21.56	-0.03	-43.52	0.49
Pursuer 2	8.28	-0.08	1.55	-0.17
Pursuer 3	-6.71	-0.27	2.98	0.26
Pursuer 4	2.98	0.14	-29.09	-0.12
Evader	0.00	0.00	0.00	0.00

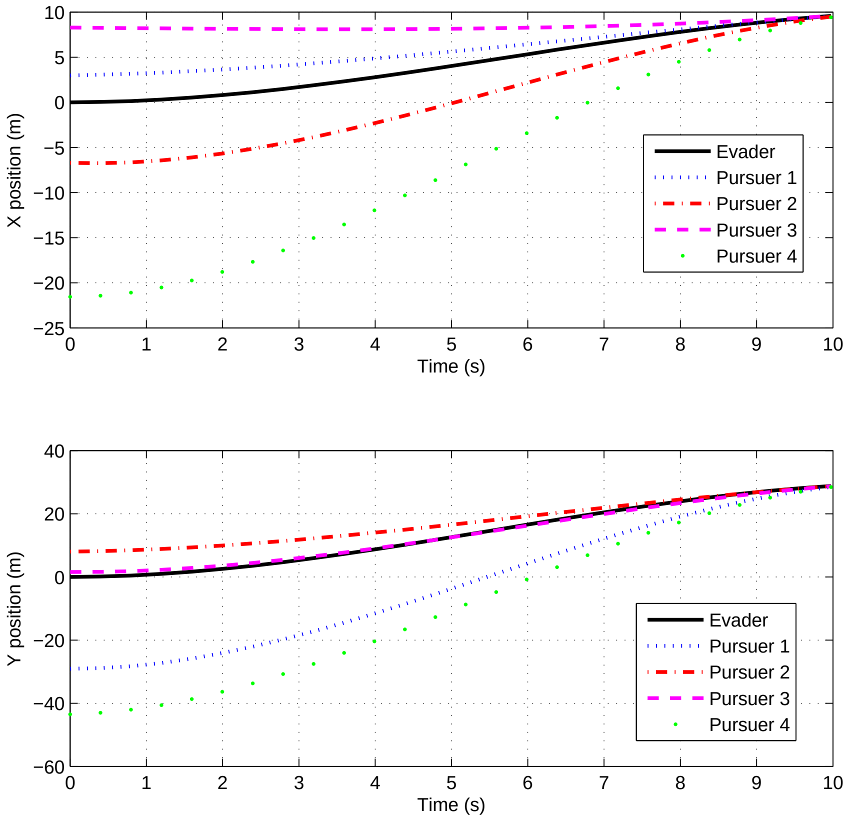

Fig. 22. Traditional Method Split View: Four Pursuers

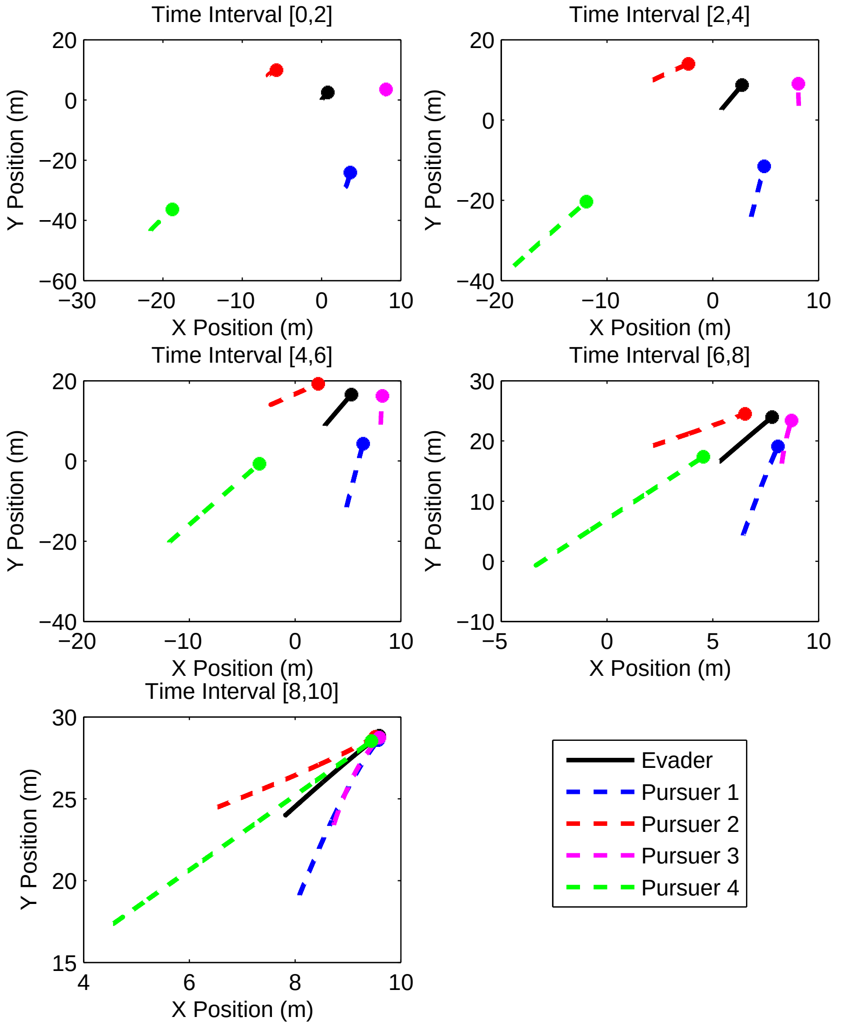

Fig. 23. Traditional Method Time Evolution: Four Pursuers

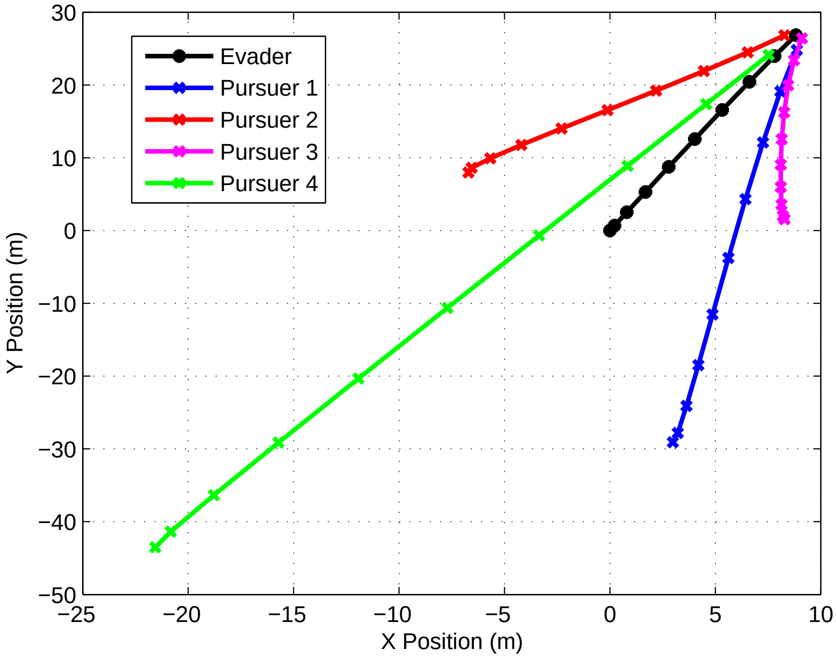

Fig. 24. Traditional Method Combined View: Four Pursuers

The four pursuers are able to capture the evader within the allotted capture time. Again the pursuers do not exhibit any signs of teamwork or cooperation in the traditional approach.

Interestingly, figure (23) shows that the evader appears to be surrounded by the pursuers in the $[4, 6]$ view. Not surprisingly, this surrounding behavior is not maintained in future intervals because it is not encouraged mathematically in the objective function.

The final example illustrates the results of four pursuers attempting to capture an evader using the new cooperation strategy.

Finally, the new cooperation strategy is used in the four pursuer, single evader game. The results for this game is illustrated in figures (25), (26), and (27). As in previous examples employing the cooperation strategy, the pursuers first surround the evader before moving to capture. Figure (26) illustrates that the evader is again nearly stationary after it has been surrounded by the evader.

This chapter introduced a general feedback based closed loop approach to solving multi-player pursuit and evasion games. A cooperation metric was introduced to encourage pursuers to first surround the evader before moving in for capture. Several numerical results illustrate differences between the traditional and new approaches, and show that the cooperation strategy does successfully encourage surrounding behavior.

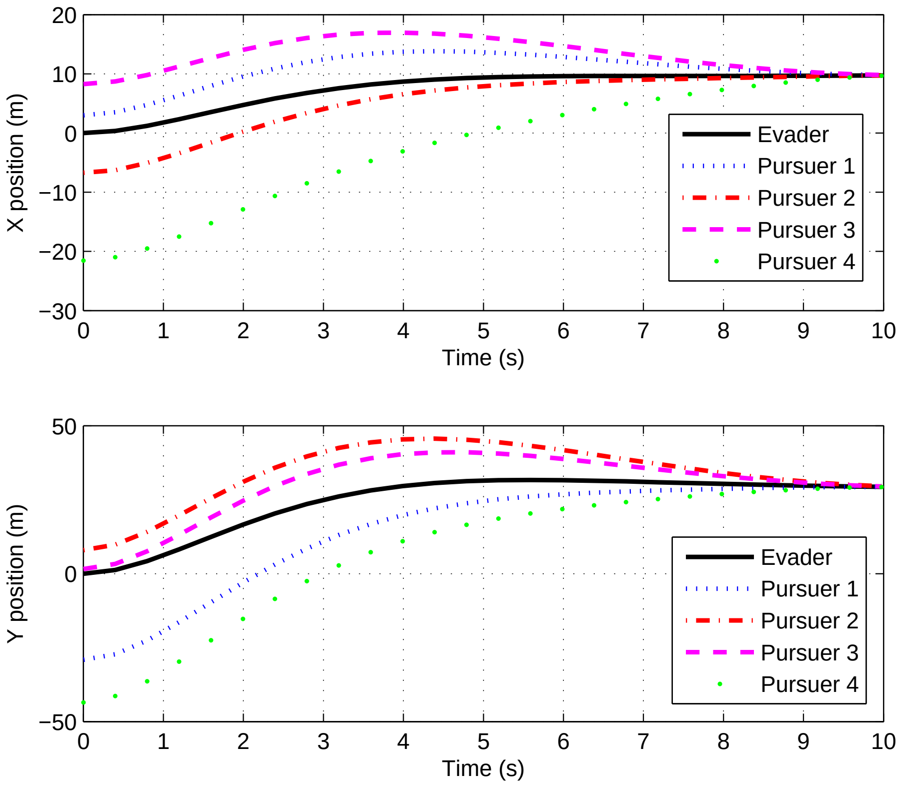

Fig. 25. New Method Split View: Four Pursuers

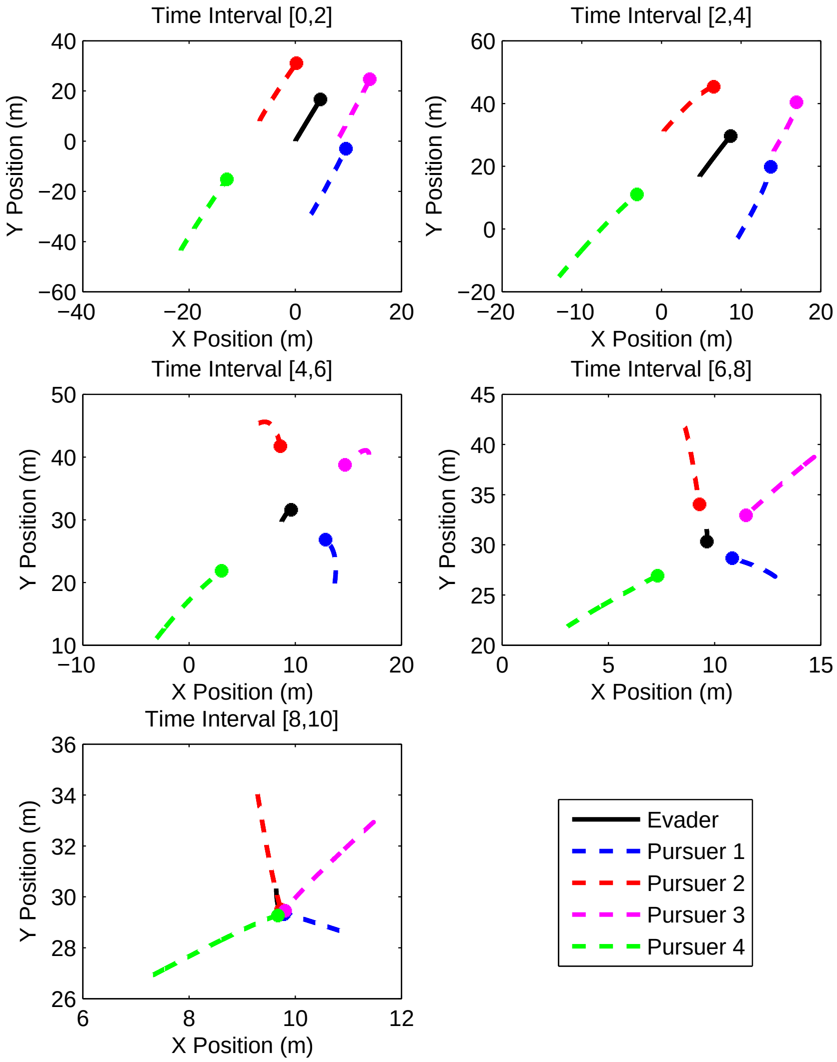

Fig. 26. New Method Time Evolution: Four Pursuers

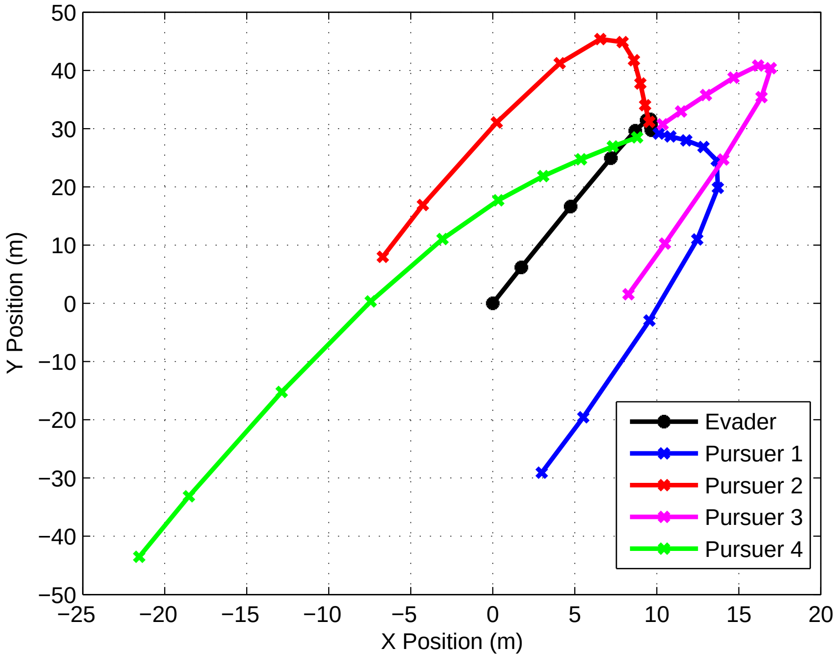

Fig. 27. New Method Combined View: Four Pursuers

CHAPTER VI

DISCUSSION AND SUMMARY

The semi-direct approach presented in this thesis shows how differential games can be solved using existing nonlinear programming (NLP) tools originally designed for optimization and optimal control problems. By incorporating some of the analytical necessary conditions, one can use a NLP tool without solving the two point boundary value problem. This technique, developed by Horie and Conway, allows more realistic differential games to be solved than previous analytical-based techniques.

The new extension of the semi-direct method presented in this thesis utilizes the semi-direct approach to solve a class of problems that include state inequality constraints. The new developments are based on existing optimal control approaches in that inequality constraints are incorporated using system influence functions. Note that, in order to solve the differential game, the NLP solution must determine the influence functions and possible discontinuities in the adjoint variables. The new extension to the semi-direct approach was successfully demonstrated by finding the solution to the classical football problem.

In addition to this extension, this thesis also presents a new cooperation strategy for multiplayer pursuit and evasion games. The incorporation of a center of mass based cooperation metric into the system objective function encourages pursuers to first surround the evader and then proceed to capture. The method was illustrated using a feedback control based framework for multi player pursuit evasion games.

Recall that the new cooperation method encouraged the pursuers to surround the evader (See figure (20)). Once the evader is surrounded, it remains nearly stationary for the remainder of the game. The game is essentially over at this point. The pursuers could take as much time as they desired to capture the opponent. Since additional movement only worsens the evaders situation by bringing it closer to a pursuer, it is left to sit and wait for the inevitable. A new definition of termination criteria for the game can then be proposed.

Generally a pursuit and evasion game is considered finished when either the pursuers capture the evader, or the evader escapes. In a cooperative game, the game can be considered over once an outmatched evader is entirely surrounded. Using this revised termination criteria, the new cooperative control method performed nearly twice as well as its more traditional counterpart. This new termination criteria might merit consideration in future cooperative control games.

In addition to this candidate for future work, several other avenues for future research are opened by the results presented in this thesis. The extensions to the semi-direct method might be used used in future work to solve a variety of differential games with state inequality constraints, such as the patrolling a channel problem with realistic dynamics. Future extensions may also encompass a wider class of state inequality constraints, such as inequality constraints that are a function of both state variables and control variables. One might also work to lessen the difficulty of finding a feasible initial solution for the NLP solver.

Ultimately, a complete numerical tool for pursuit and evasion games is desired. Such a tool would allow practising engineers to solve realistic pursuit and evasion games without having to understand all of the complexities involved in an optimal solution. Nonlinear programming direct methods for optimization and optimal control problems, such as the well-known SNOPT tool, have been widely successfully in part because of their reach to practitioners.

The new cooperative control strategy also opens up several avenues of future work. The surrounding behavior might easily be exploited in many real-world scenarios. For example, the military might wish to first surround and immobilize an enemy before an aerial attack. The surrounding cooperation technique may also be of particular applicability to cooperative herding games. Additionally, mathematical expressions for desired cooperation behavior can be developed to better simulate real world strategies. The ability to perform flanking or cornering maneuvers are two immediate candidates for this type of approach.

Even though pursuit and evasion games have been widely studied over the past half century, much work still remains to provide a more complete toolset for solving these challenging games. Due to their applicability to a wide variety of fields, it is not surprising that pursuit and evasion games have been and will continue to remain an extremely active topic of research.

REFERENCES

[1] R. Isaacs, *Differential Games.* New York: Wiley, 1965.

[2] K. Horie, "Collocation with nonlinear programming for two-sided flight path optimization," Ph.D. dissertation, University of Illinois, Urbana, IL, 2002.

[3] T. Raivio, "Computational methods for dynamic optimization and pursuit-evasion games," Ph.D. dissertation, Helsinki University of Technology, 2000.

[4] K. Horie and B. A. Conway, "Optimal fighter pursuit-evasion maneuvers found via two-sided optimization," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 29, no. 1, pp. 105–112, 2006.

[5] A. E. Bryson and Y. C. Ho, *Applied Optimal Control.* New York: Hemisphere, 1975.

[6] P. E. Gill, W. Murray, and M. A. Saunders, *User's Guide for SNOPT 7: Software for Large Scale Nonlinear Programming.* Stanford, CA: Stanford University, 2006.

[7] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for NPSOL 5.0: A Fortran Package for Nonlinear Programming.* Stanford, CA: Stanford University, 1998.

[8] *Optimization Toolbox 3 User's Guide*, 8th ed. Natick, MA: MathWorks Inc., September 2007.

[9] T. Raivio and H. Ehtamo, "Visual aircraft identification as a pursuit-evasion game," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 23, no. 4, pp. 701–708, 2000.

[10] K. Horie and B. A. Conway, “Genetic algorithm preprocessing for numerical solution of differential games problems,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 27, no. 6, pp. 1075–1078, 2004.

[11] J. V. Breakwell and A. Merz, “Football as a differential game,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 15, no. 5, pp. 1292–1294, 1992.

[12] R. Sharma, J. E. Hurtado, and S. R. Vadali, “Optimal strategies in cooperative games with terminal payoff,” in *AAS/AIAA Astrodynamics Specialist Conference*, August 2007.

[13] J. H. Holland, *Adaptaion in Natural and Artificial Systems.* Cambridge, MA: MIT Press, 1992.

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Boston, MA: Addison-Wesley, 1989.

APPENDIX A

MATRIX DEFINITIONS FOR COOPERATIVE CONTROL PROBLEMS

A concise representation of the matrices used in the cooperative control problems are listed below for reference. In the following development i denotes the matrix row while j denotes the column. Sizes of each matrix are given in terms of the number of pursuers, denoted by n. The matrix A of size $4n$ rows by $4n$ columns can be written as follows.

$$A_{i,j} = \begin{cases} 1 & , i \text{ is odd and } j = i + 1 \\ 0 & , \text{ otherwise} \end{cases} \tag{A.1}$$

The matrix C of size $4n$ rows by $2n$ columns can be written as follows.

$$C_{i,j} = \begin{cases} 1 & , i \text{ is even and } j = \frac{i}{2} \\ 0 & , \text{ otherwise} \end{cases} \tag{A.2}$$

The matrix D of size $4n$ rows by 2 columns can be written as follows.

$$D_{i,j} = \begin{cases} 1 & , i \text{ is even, } i \leq 2n, \text{ and } j = 1 \\ 1 & , i \text{ is even, } i > 2n, \text{ and } j = 2 \\ 0 & , \text{ otherwise} \end{cases} \tag{A.3}$$

The matrix V of size $4n$ rows by $4n$ columns is written as follows.

$$V_{i,j} = \begin{cases} \frac{v}{2n} & , i,j \text{ are odd and } i,j \leq 2n \\ \frac{v}{2n} & , i,j \text{ are odd and } i,j > 2n \\ 0 & , \text{ otherwise} \end{cases} \tag{A.4}$$

The matrix R of size $2n$ rows by $2n$ columns is written as follows.

$$R_{i,j} = \begin{cases} r & , i = j \\ 0 & , \text{ otherwise} \end{cases} \tag{A.5}$$

The matrix T of size 2 rows by 2 columns is written as follows.

$$T_{i,j} = \begin{cases} t & , i = j \\ 0 & , \text{otherwise} \end{cases} \tag{A.6}$$

Recall that S_1 penalizes terminal miss in position and velocity level state variables. It is a $4n$ by $4n$ matrix that is written as follows.

$$S_{1_{i,j}} = \begin{cases} s_1 & , i = j \\ 0 & , \text{otherwise} \end{cases} \tag{A.7}$$

S_2 penalizes the relative center of mass and the time rate change of the relative center of mass. It is a $4n$ by $4n$ that is written as follows.

$$S_{2_{i,j}} = \begin{cases} \frac{s_2}{2n} & , i{,}j \text{ are odd and } i{,}j \leq 2n \\ \frac{s_2}{2n} & , i{,}j \text{ are odd and } i{,}j > 2n \\ 0 & , \text{otherwise} \end{cases} \tag{A.8}$$

The user selected parameters v, r, t, s_1, and s_2 used in the new method examples given in chapter were $v = 1$, $r = \frac{1}{2}$, $t = 3$, $s_1 = 1$, and $s_2 = 1$. The traditional method parameters are $v = 0$, $r = \frac{1}{2}$, $t = 3$, $s_1 = 1$, and $s_2 = 0$.

The following matrices provide an explicit representation of the system matrices for the two pursuer case.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.9}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \quad D = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{A.10}$$

$$V = \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.11}$$

$$R = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix} \quad ; \quad T = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} \tag{A.12}$$

$$S_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.13}$$

$$S_2 = \begin{bmatrix} \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix} \tag{A.14}$$

VITA

Allen Sayre Parish III is the son of Margaret A. Parish and Dr. Allen S. Parish, Jr. He was born in Corpus Christi, Texas and lived there through high schoool graduation in 2001. Upon high school graduation, Allen enrolled at Texas A&M University to pursue his undergraduate degree. Allen graduated *summa cum laude* in May 2005 from Texas A&M University in College Station, Texas with a Bachelor of Science in Computer Engineering.

Mr. Parish's graduate studies have been generously funded through a Department of Homeland Security graduate fellowship. The research presented in this thesis was conducted under the direction and support of Dr. John E. Hurtado.

Allen married his wife Julie in May 2007. They live happily together in College Station, Texas.

Allen has previously worked at National Instruments, Lockheed Martin, and Sandia National Labortories. He currently works at Valtech Technologies in College Station, Texas as a software consultant.

Mr. Parish may be contacted via email at aparish@gmail.com. He may also be reached through Dr. John E. Hurtado, Department of Aerospace Engineering, Texas A&M University, College Station, TX 77843.