Real-Time Solutions for Pursuit-Evasion Games in the Presence of Obstacles

A Project Report

submitted by

Vineeth Ravi

EE13B122

in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2017

THESIS CERTIFICATE

This is to certify that the thesis titled **Real-Time Solutions for Pursuit-Evasion Games in the Presence of Obstacles**, submitted by **Vineeth Ravi**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Bharath Bhikkaji
Research Guide
Assistant Professor

Dept. of Electrical Engineering
IIT Madras, 600 036

Place: Chennai

Date: 8th May, 2017

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Prof. Bharath Bhikkaji for his timely guidance and the opportunity to explore different aspects of Robotics & Control Systems, throughout my undergraduate degree. Over and above the technical guidance, I thank him for his positive and encouraging outlook towards carrying out research.

I would also like to thank Jitesh Mohanan, for his guidance throughout the project and in making my BTP days a memorable experience.

Finally, I thank my family, friends and the IIT-Madras for providing me with support, opportunity and resources to carry out this research project.

ABSTRACT

KEYWORDS:

Pursuit-Evasion Games, Target-Guarding Problem, Dominance Regions, Differential Games, Path Planning, MATLAB Lego NXT toolkit, Opti-Track Camera, Isochrones.

Pursuit-Evasion Games, have been studied extensively by game theorists, engineers and scientists; since insights into these predator-prey systems enables us to develop mathematical models of real-world problems accurately. The Target-guarding problem posed by Issacs in his seminal textbook is revisited here in this project. We complicate the original plain pursuit evasion problem with the addition of obstacles which affects the players asymmetrically, hence providing higher accuracy to model rescue scenarios, biological behaviours, anti-poaching drones and so on. We implement the time-optimal strategies for the pursuer to catch the evader; We discuss optimal strategies for the pursuer and evader, with and without the presence of obstacles. We assume that the intercept happens only when there is a strict coincidence of their positions. We implement the optimal strategies as simulations in MATLAB for the pursuer to capture the evader in minimum time

The algorithm implementation for capturing the evader is based on the concept of dominance regions. The obstacles simulated here in this project are line-segments, and the simulated code can be implemented into Lego NXT robots or other anti-poaching unmanned vehicles and so on for solving various other pursuit-evasion problems.

The main goal of this project is to construct dominance regions, where a point in the plane is said to be dominated by one of the players, if he is able to reach that point earlier than the opposing player, irrespective of the strategies or actions of the opposing player. We show that, the idea of dominance provides a complete solution to this pursuit-evasion game and this can be extended, if there are obstacles present as well. We compare the dominance regions both with and without the presence of obstacles and simulate experiments, for robots in the presence of a line obstacle. We have also made use of a 3 player PE game, which can be used to model rescue scenarios, animal behaviour and other PE games which we encounter in the real world.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	**-03**
ABSTRACT	**-04**
LIST OF FIGURES & TABLES	**-06**
ABBREVIATIONS & NOTATION	**-07**
INTRODUCTION	**-08**
PROBLEM DESCRIPTION	**-11**
THEORY OF DOMINANCE REGION	**-12**
MATHEMATICAL BACKGROUND	**-15**
SIMULATION RESULTS	**-19**
THE BASIC LOS SOLUTION	**-19**
DOMINANCE REGION BASED STRATEGY	**-21**
OPTIMAL STRATEGY WITH THE PRESENCE OF OBSTACLES	**-22**
CONCLUSIONS	**-26**
FUTURE WORKS & APPLICATIONS	**-28**
APPENDIX A	**-32**
APPENDIX B	**-33**
APPENDIX C	**-36**
REFERENCES	**-40**

LIST OF FIGURES & TABLES

Fig 1: Guarding a target problem from (Isaacs, 1999). a) *P* and *E* plays optimally **-08**

Fig. 2: Bundles of Isochrones, for a single player **-13**

Fig. 3: Introducing obstacles, changes the locus of dominance regions **-15**

Fig. 4: Game geometry when vp ≠ ve and capture is the coincidence of P and E as shown **-16**

Fig. 5: Geometry of final segments of time-optimal paths. (Kabamba & Oyler,2015) **-17**

Fig. 6: The dominance region based strategy without obstacles **-18**

Fig. 7: Implementation of the Line-of-Sight algorithm **-20**

Fig 8: Implementation of the Dominance-Region based algorithm for optimal solution. **-21**

Fig. 9: This shows us the comparison between the two paths which the pursuer can follow in the presence of an obstacle **-23**

Fig 10: This figure shows a comparison among all possible strategies discussed till now with and without the presence of the obstacle **-24**

Fig. 11: Exceptional Calibration of the Cameras of the Opti-Track System. This system is used to stream the coordinates of the Pursuer and Evader to MATLAB, and we can control the pursuer to follow the time-optimal route, using the algorithm discussed till now **-28**

Fig. 12: The cameras mounted gives us an overhead picture of the robot trajectory paths **-29**

Fig. 13: Picture Depicting the Opti-Track system in the Lab. The Blue rigid body is the pursuer. The Red rigid body is the evader. The Yellow Markers represent the cameras of the Opti-Track system **-29**

Fig. 14: The Green marker represents the pursuer, and the red marker on top represents the evader. The line shown, is the obstacle in place. **-30**

Table 1: Results of a particular set of simulations with given initial conditions **-27**

ABBREVIATIONS & NOTATION

IIT-Madras - Indian Institute of Technology Madras

P3 - Predator, Protector, Prey

RRT - Rapidly exploring Random Trees

SLAM - Simultaneous Localization and Mapping

PE - Pursuit Evasion

Fig. - Figure

LOS - Line of Sight

Vp – Velocity of pursuer

Ve – Velocity of evader

T – Time

X – x coordinate in field

Y – y coordinate in field

R – Radius of the circle which represents the dominance region

INTRODUCTION

Robotics has a wide range of applications to Game-theoretical models, particularly to Pursuit-Evasion Games. Pursuit-Evasion games refer to the class of games, where there is at least 1 pursuer trying to catch an evader. An autonomous guard robot can be used in a variety of fields ranging from anti-poaching systems, where autonomous robots protect the habitat of wild animals from intruding poachers, to border patrol problems, where teams of autonomous robots are tasked with protecting the border of a secluded campus or region of interest. A major challenge in building such a guard robot is developing path planning algorithms that can compute optimal trajectories for the robot in real-time. Rufus Isaacs, in his pioneering work on differential games (Isaacs, 1999), presented optimal strategies for both the guard and the intruder, assuming both agents to have equal speed, see Fig. 1.

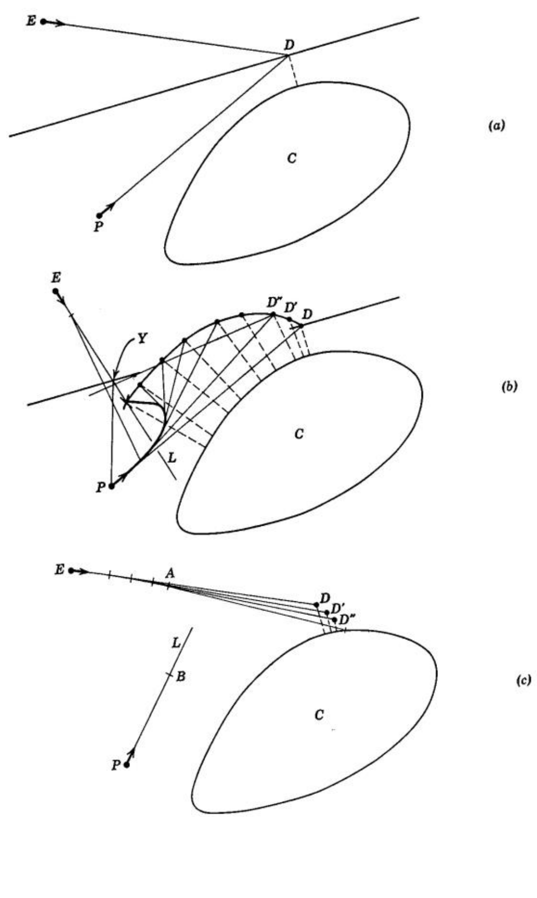


Figure 1: Guarding a target problem from (Isaacs, 1999). a) *P* and *E* plays optimally. b) *P* plays optimally, but *E* does not. c) *E* plays optimally, but *P* does not.

The target guarding problem is a two-player game wherein the intruder robot (evader *E*) attempts to reach a secluded region (target area *C*) while evading capture by the guard robot (pursuer *P*). The objective of the participating agents is opposite, thus making it a differential game. While the intruder attempts to minimize the distance to the target area at the end of the game, the guard robot attempts to maximize it. The game ends if the intruder safely reaches the target area or if the guard robot successfully intercepts the intruder before the intruder reaches the target area. Rufus Isaacs showed that, for the simple case where both agents have equal speeds, if the target region is closer to the guard, then the optimal strategy for both agents is to move towards that point *D*, on the perpendicular bisector of their initial positions, which is closest to that target area. In particular, this implies that if the intruder follows a pure-pursuit strategy (i.e.) head directly towards the target, he is playing sub-optimally. A key characteristic of a differential game is that if either of the agents adopt a policy other than the optimal strategy, it results in an advantage for the other agent. Therefore, an optimal guidance law for the guard robot should compute the desired heading for the robot at each instant using the current positions of the agents.

In the simple version of the game analysed by Rufus Isaacs, the velocities of both the agents were assumed to be equal. Further, the criterion for capture is the coincidence of the two robots. In reality, both these assumptions are incorrect. The velocity of the guard robot is usually greater than that of the intruder robot. Further, the robots are not point objects. Therefore, their physical size implies that capture occurs if the agents come within a certain distance δ whose value depends on the size of both the agents. In (Venkatesan and Sinha, 2014), Raghav et. al. extended the work of Rufus Isaacs to remove these two assumptions. Further, they provide a fast computation method to compute the optimal heading for the guard robot at every instant. In (Lau and Liu, 2014), Liu and Lau present an autonomous border patrol system that considers a single fast guard robot protecting the target region from multiple non-cooperating intruders. A major limitation of these works is that they assume the agents to be point objects that can instantaneously move in any direction on the 2D plane. But such robots belong to a special class of robots called holonomic robots. Most robots in reality are nonholonomic and have a minimum turning radius. Even robots with zero minimum turning radius, such as differential drive robots, have constrained motion such as inability to move laterally. Such constraints on the dynamics of the robots can potentially change the outcome of the game. In this work, we validate the optimal strategies for the guard robot proposed in (Venkatesan and Sinha, 2014) and study the effect of the vehicle dynamics on the optimality of the proposed strategies.

This project is motivated by the prevalence of pursuit–evasion games with obstacles and the corresponding lack of a study of the dominance regions for these games. PE games can be used to model a variety of scenarios, including predator/prey relationships in biology, military confrontations, and worst-case scenarios for rescue missions. The addition of obstacles enables the study of more complex and realistic scenarios. For example, it provides a way to study the impacts of human infrastructure on predator/prey relationships. There are also motion planning applications for teams of robots in obstacle-rich environments, such as urban search and rescue scenarios. Finally, PE with obstacles provides a framework to study military troop movements and surveillance missions in urban conflicts. In some circumstances, obstacles have asymmetric effects on the players. For example, if an unmanned aerial vehicle (UAV) on a surveillance mission is attempting to take a picture of a mobile ground target, then the ground target is constrained by buildings and other ground obstacles that do not affect the UAV. Similarly, airspace structures such as no-fly zones affect the UAV, but not the ground target. Games in these types of environments cannot be analysed with current methods. Multiplayer PE games are also of current interest, and an understanding of dominance leads to valuable insights into these games. For example, the P3 game is applicable to combat search and recovery scenarios, its solution is obtained through the use of dominance regions.

To solve this pursuit-evasion game, many strategies are simulated. We discover optimal solutions to solve the Target-Guarding problem, based on theory of dominance regions and simulate real-time algorithms, so these can be implemented in actual robots. In this Project, we extend the game for inclusion of obstacles and find the optimal way to solve the pursuit-evasion problem. The introduction of obstacles help us to model real-world scenarios effectively, which has not been done previously in other Target-Guarding Problem models.

The code used for simulation in MATLAB, can be implemented in real-time robots, in the presence of obstacles to solve other Pursuit-Evasion problems described in the below section.

PROBLEM DESCRIPTION

This project primarily addresses the problem of finding an optimal solution to existing Pursuit-Evasion games in the Real-World. We use the concept of the dominance regions to develop optimal strategies in the presence of obstacles for the pursuer to capture the evader in minimum time.

We first analyse the theory behind creating dominance region. Given a pursuer P, with velocity Vp, and evader E, with velocity Ve, both moving in a plane, with or without the presence of obstacles, we construct the dominance regions. We assume that the robots can turn instantly, and each player knows the location of the other players all the time. Hence we find the locus of the points that are dominated by the evader.

We try to develop an algorithm, which is computationally efficient, and can be used for quickly computed in various scenarios for capturing the evader, as opposed to other algorithms, developed till now based on the dominance approach, which have very computationally expensive, i.e. taking a long time to compute the optimal path. This is not optimal because, in the real-world, we do not get the luxury of time, to compute for large amounts of time, the optimal path to reach the evader, as a pursuer.

We first identify, how the theory of dominance, is established as an intersection of isochrone bundles. We then compute the mathematical equations behind the construction of dominance regions, and implement these equations for the pursuer to compute, in the least time as possible, for capturing the evader in minimum time.

We do this for both the cases where the obstacle is present and as well as absent.

Finally, we can also implement these simulations into actual LEGO EV3 robots, and experiment with other PE games and computation strategies, as shown in the Future Works Section of the Thesis.

THEORY OF DOMINANCE REGIONS

BACKGROUND

The following theorems are used as mentioned in the respective sections: -

Theorem 1. In the absence of obstacles, the time-optimal paths are straight lines, and the isochrones are concentric circles centered at the agent's initial location.

Theorem 2. In the presence of a set of polygonal obstacles, the timeoptimal paths are broken lines, breaking at obstacle vertices, and the isochrones form arcs of concentric circles centered at generating points, where a generating point is either an obstacle vertex or the agent's initial location.

Theorem 3. The curves that separate the plane into regions with unique generating points are either line segments or arcs of hyperbola.

Theorem 4. In the absence of obstacles, the dominance regions of the PE game are divided by an Apollonius circle.

Theorem 5. In a two player PE game of degree with no obstacles where the payoff is the time to capture, which P seeks to minimize and E seeks to maximize, the optimal strategies are such that capture occurs at the point, C, on the Apollonius circle that is farthest from E's initial position, and optimal play dictates that P and E both travel to C in minimum time.

Theorem 6. In the PE game with obstacles, the plane is exhaustively divided into three disjoint regions:

(1) A region where a player strictly dominates, (2) A region where the other player strictly dominates,
(3) A region where neither player dominates.

Moreover, the third region is obtained by intersecting bundles of isochrones.

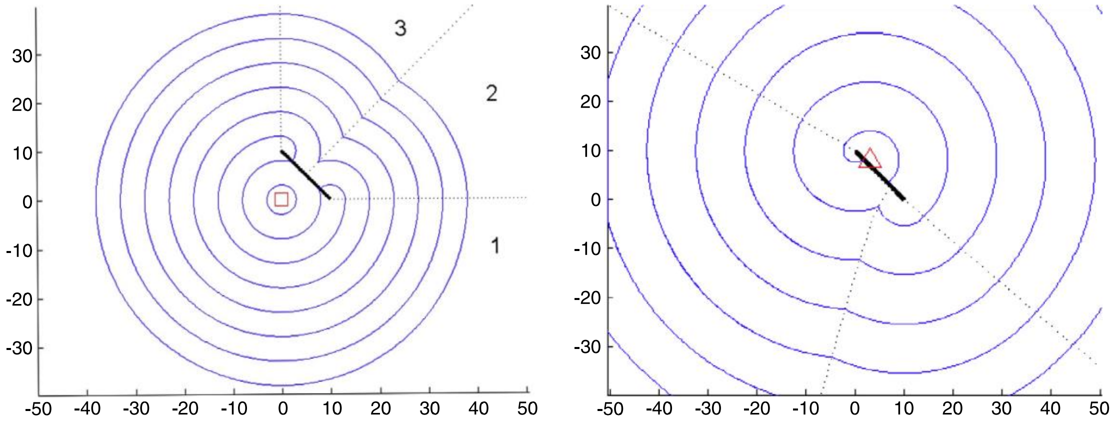
Theorem 7. Each portion of the dominance boundary satisfies the following condition for a specific value of t_B and d:

$$(\gamma^2 - 1)\, r^2 + 2\left(d\cos\theta - \gamma^2 v_A t_B\right) r + \left(\gamma^2 v_A^2 t_B^2 - d^2\right) = 0. \qquad (7)$$

Single Player moving in the presence of obstacles

The time-optimal control problem can be stated as follows: given an agent moving with simple motion and speed v, initial and final locations, (x_i, y_i) and (x_f, y_f), and a set of known obstacles, S, find a path that connects (x_i, y_i) to (x_f, y_f) without intersecting S such that the time required for the agent to reach (x_f, y_f) is minimized.

We use the theorems 1,2 and 3, to get the dominance regions for a single player in the presence of obstacles, as show below: -



(a) Prey. (b) Protector.

Figure 2: (Oyler and Kabamba, 2015) Bundles of Isochrones, for a single player.

The thick line is the obstacle, and the thin lines are the isochrones and the dotted lines are the curves which separate the regions, with different generating points. Region 1 is unaffected by the obstacle, but regions 2 and 3 are affected by the line obstacle.

Our Simulations later, analyse specifically to regions 1,2 and 3. We then compare and contrast the optimal paths, for regions 1,2 and 3.
The region 2 and 3, pertain to having an obstacle in front, whereas region 1 implies, no obstacle present, which has been described in the simulations.

Pursuit-Evasion Games

The game which we analyse in the simulations pertain to a single pursuer and a single evader. The complete solution is establish through the strategy of dominance region i.e. constructing an Apollonius circle.

It is the of the form

$$(\gamma^2 - 1)\, r^2 + (2d\cos(\theta))\, r - d^2 = 0,$$

where (r,θ) are polar coordinates with origin at the evader's initial location and the direction of zero azimuth along the line of sight to the pursuer; $\gamma = v_P/v_E$ is the speed ratio, and d is the initial distance between the players.

The above equation is derived in detail below, in the Mathematical Background Section of the Thesis.

Dominance boundary as an intersection of isochrone bundles

Here, we present the theory of dominance region as an intersection of bundles of isochrones, as mentioned in the single player case. This method, agrees with Theorem 4, when no obstacles are present. We extend this and use Theorem 5, to construct dominance regions in the presence of obstacles.

We construct isochrones for a specified time duration. These curve's parametrized by time t, is formed for each player. We eliminate the common variable t, which leads to the region, where players meet using time-optimal paths.

We now use theorem 6, to obtain a mathematical expression for this case.

As an example, consider the PE game with no obstacles in a Cartesian coordinate system with origin at the evader's initial location. If (x_E, y_E) and (x_P, y_P) are the positions of the evader and the pursuer, respectively, and if the pursuer's initial location is $(x_{P,0}, y_{P,0})$, then for a given time, t, the isochrones for each player are given by

$$x_E^2 + y_E^2 = v_E^2 t^2,$$
$$(x_P - x_{P,0})^2 + (y_P - y_{P,0})^2 = v_P^2 t^2.$$

After eliminating the common parameter, t, the intersections of the isochrones are the points where $x_E = x_P$ and $y_E = y_P$, so the subscripts are dropped, yielding

$$\frac{x^2 + y^2}{v_E^2} = \frac{(x - x_{P,0})^2 + (y - y_{P,0})^2}{v_P^2}.$$

Substituting $x_{P,0} = d$, $y_{P,0} = 0$, and $\gamma = v_P/v_E$ gives

$(\gamma^2 - 1)(x^2 + y^2) + 2dx - d^2 = 0$

This gives us the equation of the dominance region, with the presence of line obstacles in the plane. This equation would need to be modified for other nonlinear and polygonal obstacles, for implementation, for the scenarios mentioned in the Future Works Section.

The below figure, shows the dominance region, in the presence of line obstacles based on the intersection of isochrones.

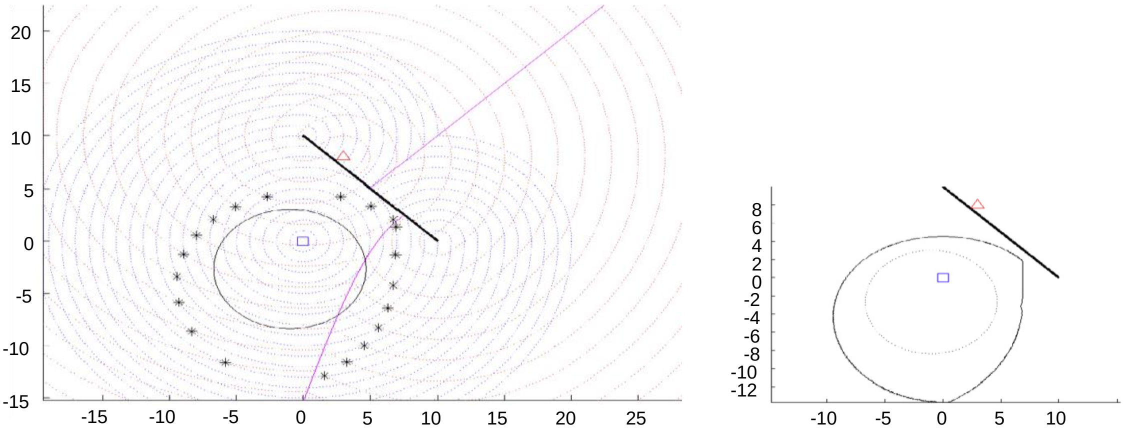


Figure 3: Introducing obstacles, changes the locus of dominance regions, as shown in Kabamba and Oyler, 2015

We realize that, implementing the dominance region strategy, using this approach, would be infeasible, because it would take a lot of time for the pursuer, to compute the dominance region, and this can be implemented in the real-world.

Hence, we develop equations of the dominance region in closed form, as shown in the mathematical background section below

MATHEMATICAL BACKGROUND

Issacs analysed the target guarding problem for a single pursuer and a single evader. We develop equations for constructing the dominance regions in closed form expression, So we can implement them in computationally efficient algorithms for use in real-world pursuit-evasion games or problems.

In this case, the perpendicular bisector of the line connecting their initial positions divides the playing space into regions that either of them can reach prior to the other. In other words, when both *P* and *E* travel straight towards the same point on the perpendicular bisector, they arrive at the same instant, and capture occurs. However, if the speeds of *P* and *E* are different, capture would not occur on the perpendicular bisector. To find the curve on which capture occurs, let

us assume that the ratio of the maximum speeds of *P* and *E* be $p : e$. Therefore, let the maximum speed of *P* be $v_p = pv$, and that of *E* be $v_e = ev$, where v is some common factor. Let t be the time elapsed until capture. Both *P* and *E* will travel straight at their maximum speeds because decreasing their speed or traveling in a curved path would benefit the opponent. Let $E_0\,(x_e,y_e)$ be the initial position of E and $P_0\,(x_p,y_p)$ be that of P.

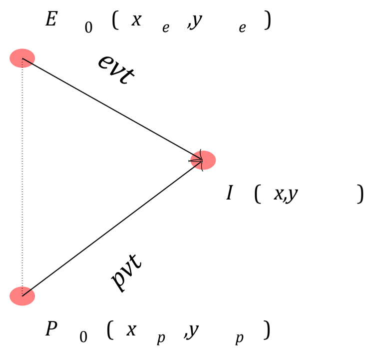


Figure 4: Game geometry when vp ≠ ve and capture is the coincidence of P and E as shown in Raghav and Sinha, 2014

Let us assume that the condition for capture is the coincidence of positions of P and E. Let $I(x,y)$ be the point where capture occurs, as shown in Fig. 2. Then, the distances traveled by P and E until capture would be *pvt* and *evt* respectively, and can be expressed mathematically as:

$$(x - x_e)^2 + (y - y_e)^2 = (v_e t)^2 = (evt)^2 \qquad (1)$$
$$(x - x_p)^2 + (y - y_p)^2 = (v_p t)^2 = (pvt)^2 \qquad (2)$$

Dividing (1) by (2), we get

$$\frac{(x - x_e)^2 + (y - y_e)^2}{(x - x_p)^2 + (y - y_p)^2} = \frac{e^2}{p^2} \qquad (3)$$

$$\begin{aligned} \Longrightarrow p^2 &\left[\left(x^2 - 2xx_e + x_e^2\right) + \left(y^2 - 2yy_e + y_e^2\right)\right] \\ &= e^2 \left[\left(x^2 - 2xx_p + x_p^2\right) + \left(y^2 - 2yy_p + y_p^2\right)\right] \end{aligned} \qquad (4)$$

$$\begin{aligned} \Longrightarrow\ & \left(p^2 - e^2\right) x^2 + \left(p^2 - e^2\right) y^2 \\ & + 2\left(x_p e^2 - x_e p^2\right) x + 2\left(y_p e^2 - y_e p^2\right) y \\ & + p^2\left(x_e^2 + y_e^2\right) - e^2\left(x_p^2 + y_p^2\right) = 0 \end{aligned} \qquad (5)$$

$$\begin{aligned} \Longrightarrow x^2 &+ 2\frac{\left(x_p e^2 - x_e p^2\right)}{\left(p^2 - e^2\right)} x \\ &+ y^2 + 2\frac{\left(y_p e^2 - y_e p^2\right)}{\left(p^2 - e^2\right)} y \\ &+ \frac{p^2\left(x_e^2 + y_e^2\right) - e^2\left(x_p^2 + y_p^2\right)}{\left(p^2 - e^2\right)} = 0 \end{aligned} \qquad (6)$$

Let $x_c = \frac{(x_e p^2 - x_p e^2)}{(p^2 - e^2)}$ and $y_c = \frac{(y_e p^2 - y_p e^2)}{(p^2 - e^2)}$. Then, (6) becomes

$$x^2 - 2xx_c + y^2 - 2yy_c + \frac{p^2\left(x_e^2 + y_e^2\right) - e^2\left(x_p^2 + y_p^2\right)}{(p^2 - e^2)} = 0 \quad (7)$$

$$\Longrightarrow (x - x_c)^2 + (y - y_c)^2 + \frac{p^2\left(x_e^2 + y_e^2\right) - e^2\left(x_p^2 + y_p^2\right)}{(p^2 - e^2)} - x_c^2 - y_c^2 = 0 \quad (8)$$

$$\Longrightarrow (9) \quad \boxed{(x - x_c)^2 + (y - y_c)^2 = R^2}$$

where

$$R = \sqrt{x_c^2 + y_c^2 - \frac{p^2(x_e^2 + y_e^2) - e^2\left(x_p^2 + y_p^2\right)}{(p^2 - e^2)}}.$$

Equation (9) is that of a circle with center $C(x_c,y_c)$ and radius R.

The above equation (9) converted to parametric form yields us

$$\left(\gamma^2 - 1\right) r^2 + (2d \cos(\theta))\, r - d^2 = 0,$$

where (r,θ) are polar coordinates with origin at the evader's initial location and the direction of zero azimuth along the line of sight to the pursuer; $\gamma = v_P/v_E$ is the speed ratio, and d is the initial distance between the players.

Now, we derive closed form expression in parametric form for the dominance regions with the presence of the obstacle:

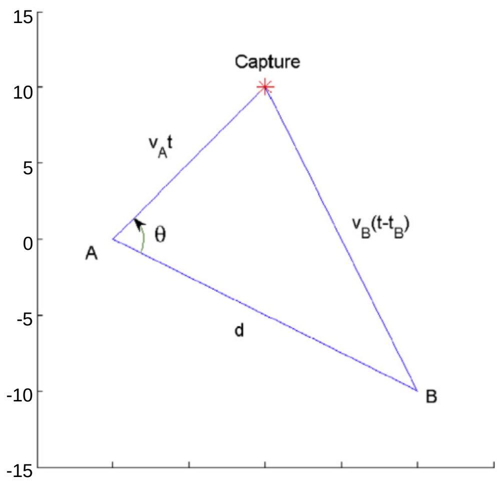


Figure 5: Geometry of final segments of time-optimal paths. (Kabamba & Oyler,2015)

The locus of intersections of the isochrone bundles can be determined from Figure 5 where *d* is the distance between the starting points of the final segments. From the law of cosines:

$$v_B^2(t - t_B)^2 = v_A^2 t^2 + d^2 - 2v_A td\cos(\theta).$$

We use theorem 7, with the above derived equations to yield us

$$(\gamma^2 - 1)\, r^2 + 2\left(d\cos\theta - \gamma^2 v_A t_B\right) r + \left(\gamma^2 v_A^2 t_B^2 - d^2\right) = 0. \quad (7)$$

Which gives us the closed form expression of the dominance region in the presence of obstacles. We have implemented these equations in the algorithm for obtaining the time optimal path.

We now create simulations based on the closed form expressions derived till now:

We expect them to look like the below figure, in the absence of obstacles: -

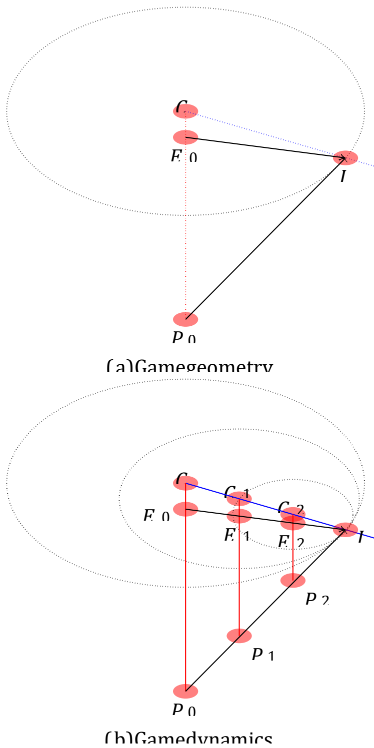


Figure 6: Raghav & Sinha,2014

The dominance region based strategy without obstacles

We have also done simulations, for the algorithm based on Line of Sight (LOS), we now compare the traditional LOS algorithm with the dominance region based strategy, to compare the minimum time obtained and validate the optimality of the dominance-region based strategy.

The simulation figures and results, along with the advantages and limitations of each algorithm are shown below.

SIMULATION RESULTS

Experimental Simulations have been carried out in MATLAB, (the code can be referred to in APPENDIX A,B and C), for carrying out other simulations or extending the project as show in the Future Works Section.

The below simulations have the initial conditions: -

Vp = 3m/s

Ve= 1m/s

(predator) E - Evader starting location = (0,0) (The location where the evader starts is taken as origin)

(protector) P- Pursuer starting location = (20,0)

(prey) T- immobile target location at (0,10)

Line obstacle coordinates {(10,2), (15,4)}

THE BASIC LOS SOLUTION

Please refer to the code for obtaining the figure shown below in APPENDIX A

This picture below illustrates the LOS algorithm for the pursuer to capture the evader. At every instant, the pursuer, checks the heading angle ‘theta’, between the pursuer and evader, i.e. the Line of Sight angle, and proceeds to orient itself and go in that direction. It keeps re-orienting itself, based on this angle, dynamically at every time instant, till it finally reaches the evader. We assume that the pursuer speed is greater the evader’s speed, because only then capture occurs through this algorithm in finite time.

The figure below illustrates this algorithm:

The red dotted line refers to the evader starting at origin.

The blue dotted line refers the pursuer starting at (20,0).

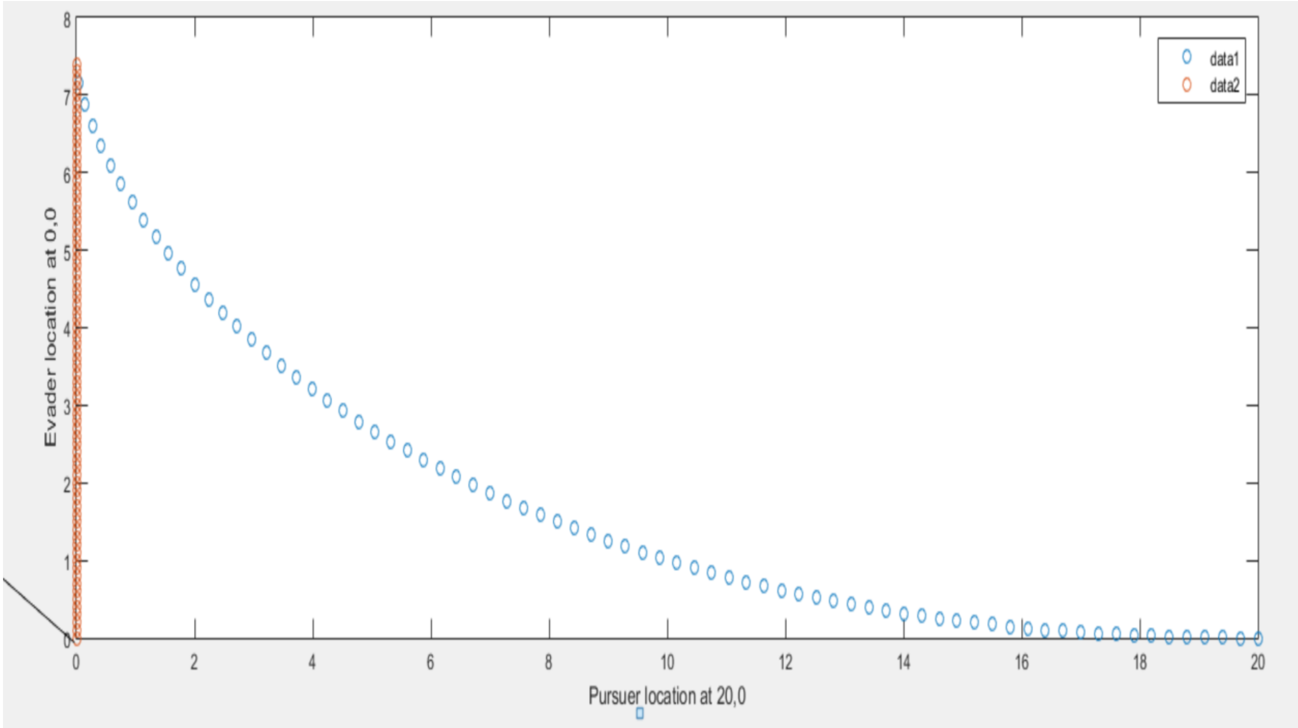


Figure 7: Figure Representing the LOS approach

Pursuer & Evader meet at (0, 7.5), i.e. capture occurs here. The time for capture is 7.5 seconds.

Advantages:

1) Intuitive to understand

2) Easy to Implement

3) Models animal behaviour accurately

Disadvantages:

1) Not Optimal solution

2) Fails in the case of obstacles, as LOS is lost.

3) Cannot be extended when multiple pursuers and evaders are present

Hence, we need to develop and implement optimal strategies to overcome the limitations of the LOS solution.

One such optimal solution is through the use of dominance regions, which has been discussed in detail now; The simulation results of this strategy is shown below

DOMINANCE REGION BASED STRATEGY

Please refer to the code for obtaining the figure shown below in APPENDIX B

The below figure illustrates the comparison between the LOS strategy and the Dominance region based strategy.

The yellow dotted line refers to the strategy based on the dominance region approach. Here we compute the dominance region circle, and optimally decide our heading direction as our pursuer. This dominance region algorithm has been discussed in detail above.

The red dotted line refers to the primitive LOS approach

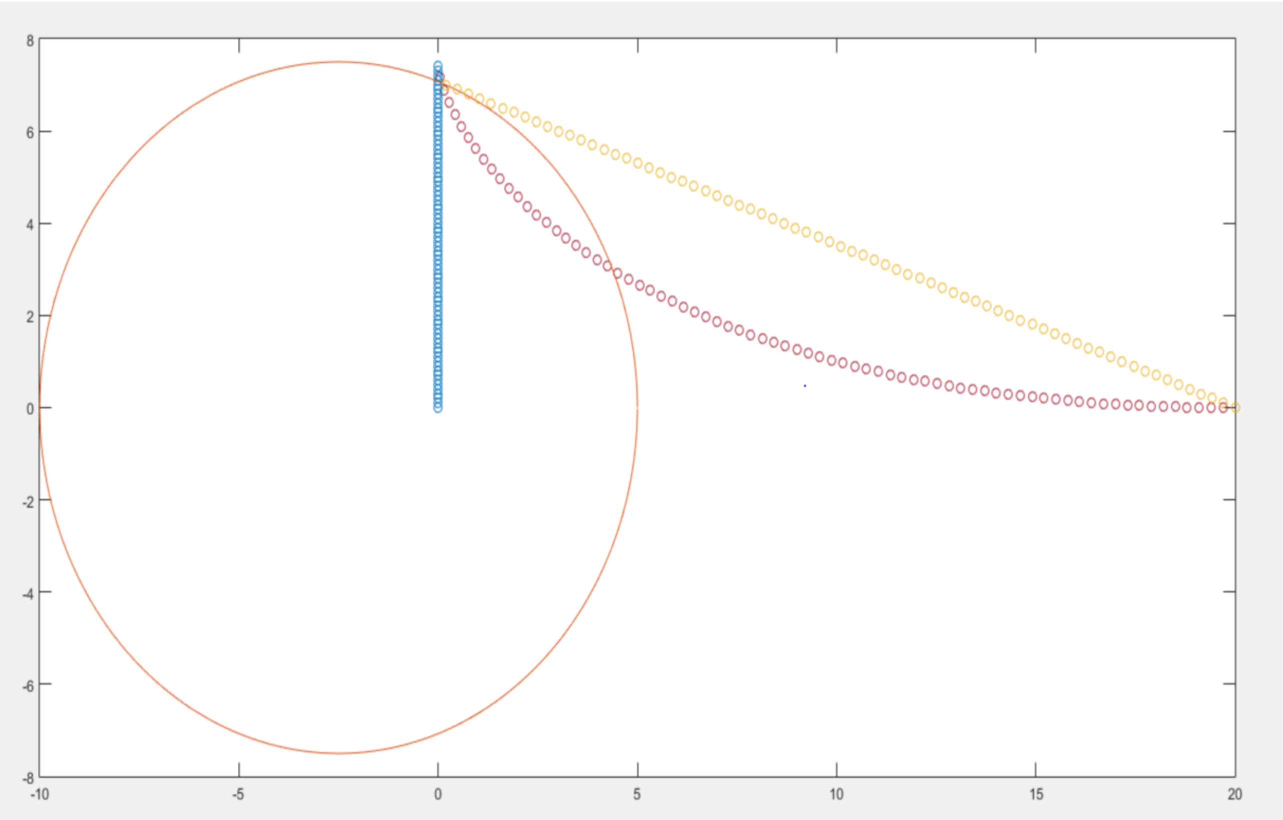


Figure 8: Implementation of the Dominance-Region based algorithm for optimal solution.

The capture here occurs in 7.1 seconds, lesser compared to the LOS approach which is 7.5 seconds. Hence, it is better than the previous LOS approach.

If, the Target T, was present at (7.3,0) instead of (10,0) the LOS approach, would have failed, and the evader would have won the game, since the pursuer would be too late in protecting the target.

Advantages:

1) Optimal Algorithm for pursuer to catch the evader

2) Easily extended to multiple pursuer-evader systems

3) LOS is not required

4) Gives a complete solution of the game

Disadvantages:

1) Needs to be improvised in the case of obstacles

2) Cannot work, if initial location of evader or velocity of evader is not known.

3) Harder to implement compared to LOS

We now implement the same algorithm of dominance region based strategy, modified for the case where obstacles are present and affect both the pursuer and evader asymmetrically.

OPTIMAL STRATEGY WITH THE PRESENCE OF OBSTACLES

Please refer to the code for obtaining the figure shown below in APPENDIX C

The dominance region approach is the optimal strategy to be followed even in the presence of obstacles. The pursuer computes, two dominance circles, based on the number of edges of the obstacle, this leads to multiple paths available of the pursuer. Amongst these paths, the pursuer chooses the path, which takes the least time to travel.

Path 1 - Denoted by the green and red dotted line

Path 2 - Denoted by the orange and purple dotted line

The obstacle is a line segment, starting at (10,2) and ending at (15,4).

The first path, i.e. Tpath1 takes 7.2 seconds.

The second path, i.e. Tpath2 takes 7.4 seconds.

Hence, we make the pursuer choose the 1st path, since, this can be pre-calculated by the pursuer in dynamic environments.

The figure below illustrates the two dominance region circles which arise, because of the present of the line obstacle.

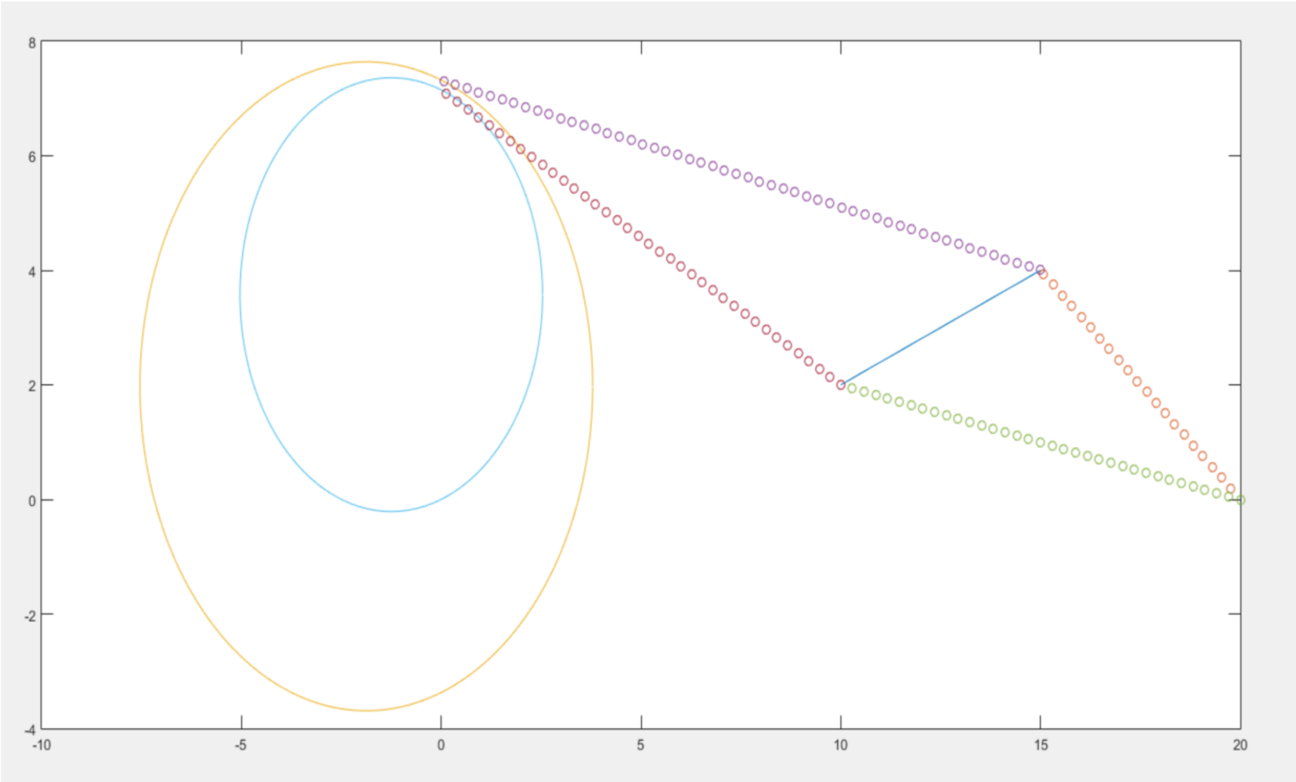
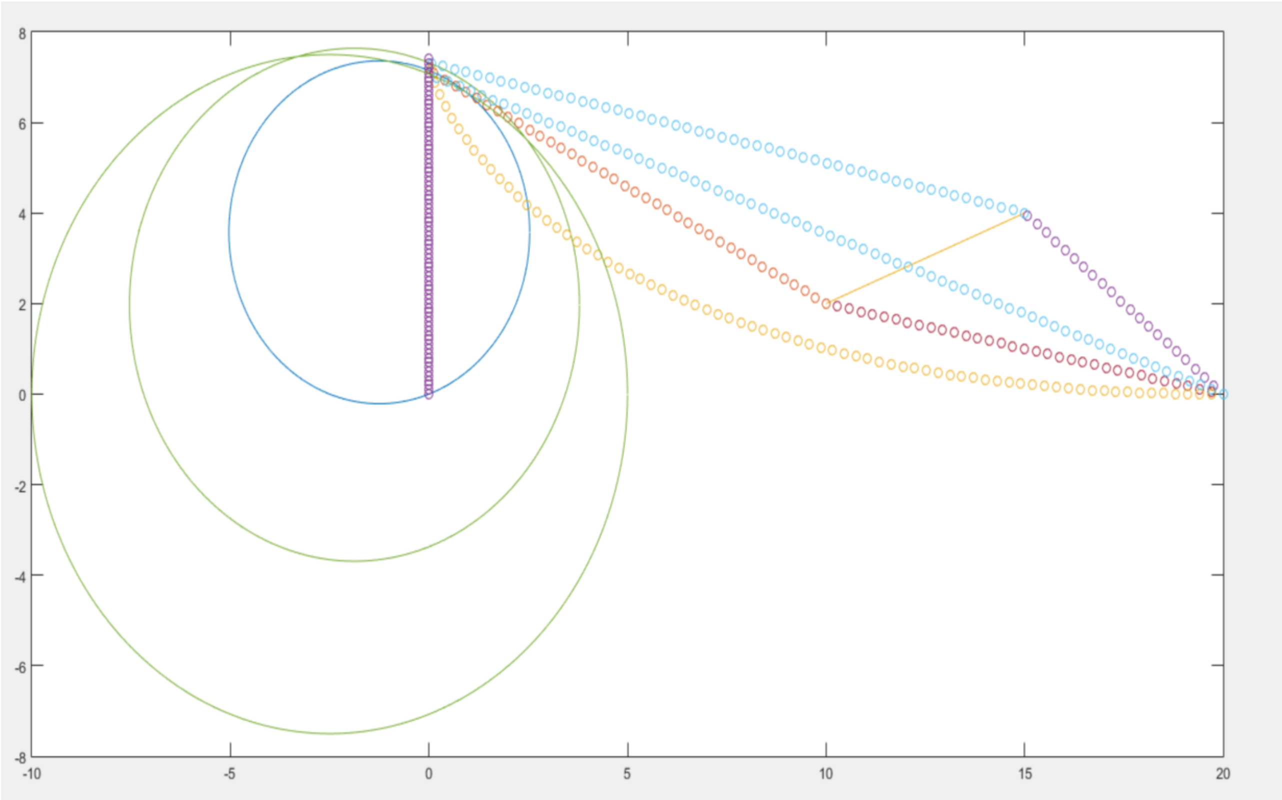


Figure 9: This shows us the comparison between the two paths which the pursuer can follow in the presence of an obstacle

This algorithm can be implemented in real-time robots like anti-poaching drones, and other mobile UAV's since its computationally fast as well as gives a time-optimal algorithm in the presence of obstacles. It can also be used to simulate other complex mathematical models of pursuit-evasion games, since this takes into account of obstacles as well for calculating the time optimal path for the capture to happen.

Let us now take a look at the summary of all strategies discussed/simulated till now as shown in the figures below: -

a) Pursuer computes all possible paths in advance and chooses the time optimal path



b) The path 1 is the time optimal path for the pursuer to follow

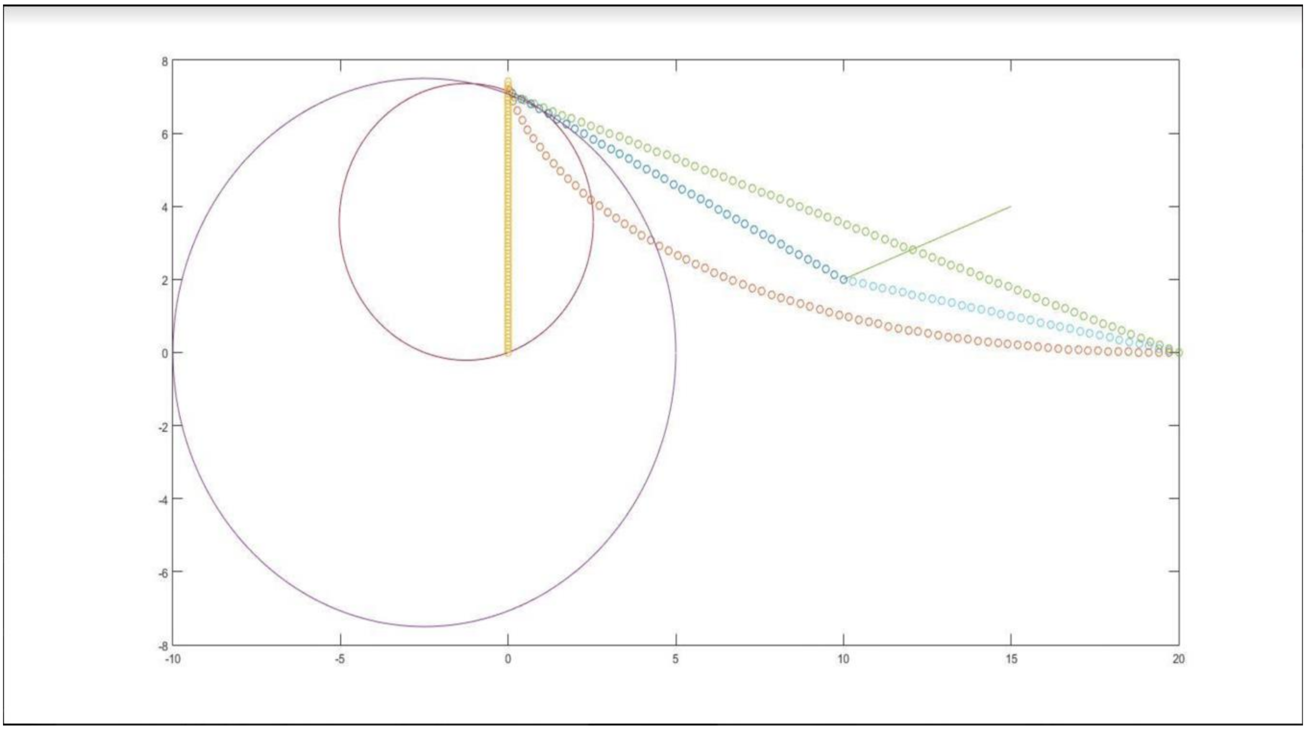


Figure 10: This figure shows a comparison among all possible strategies discussed till now with and without the presence of the obstacle.

Hence the optimal solution to the PE dominance region and P3 game is now obtained even with the presence of obstacles.

In Figure 10 a)

1) The purple dotted line is the movement of the evader to the stationery target T.

2) The yellow dotted line is the LOS approach.

3) The full blue dotted line is the dominance region based strategy, in the absence of the obstacle.

4) The orange-red and blue-pink dotted lines are the optimal paths available to the pursuer in the presence of a line obstacle.

The orange-red path, Path 1 is the time-optimal path with T = 7.2 seconds for capture,

In Figure 10 b)

1) The yellow dotted line is the movement of the evader to the stationery target T.

2) The orange dotted line is the LOS approach, in the absence of the obstacle.

3) The green dotted line is the dominance region based algorithm strategy in the absence of obstacle.

4) The blue dotted line is the dominance region based algorithm strategy in the presence of obstacle. This is the time-optimal path (path1), which has capture time of 7.2 seconds.

The given initial conditions are Ve=1m/s, Vp=3m/s, E0= (0,0), P0= (20,0) in the presence of obstacles. The dominance region based algorithm, is the optimal strategy for both the pursuer and evader to follow. Any other strategy will be sub-optimal and will lead to a disadvantage for that agent.

The initial conditions, i.e. location and speed of the pursuer and evader, the location and the size of the obstacle can all be changed, in MATLAB, by doing this, we can truly appreciate the optimality of the dominance-region based algorithm over other strategies for both the Pursuer and Evader. We can see that the differences in time for capture is now more prominent, on changing the initial conditions and running iterations of simulations based on the dominance region algorithm with the presence of line obstacles.

This algorithm can be implemented into actual real-time robots, as shown in the Future Works section of this project report.

CONCLUSIONS

The results, gives us a theoretical and experimental proof that the algorithm used in MATLAB, gives us the optimal time for the pursuer to catch the evader. We can compute the trajectory which the pursuer needs to follow, to catch the evader in minimum time, as shown in the graphs of the simulations results. Though further work needs to be done, to practically test the computational speed of the algorithm, as mentioned in the Future Works Section of the Thesis, this algorithm works well in ensuring minimum time of capture in the presence of line obstacles. In this project, the target guarding problem was solved for the pursuer and evader having simple motions with different speeds. The target guarding problem is fundamental to systems that protect a target from enemies, for example, unmanned vehicles trying to protect endangered animals from poachers. Results discussed in this project could be applied to a single drone trying to protect a static animal, under threat from a single poacher.

The optimal strategy for the drone would depend on the ratio of its speed to that of the poacher. If the drone's speed is higher than that of the poacher (which is usually the case), it would be possible for the drone to keep the poacher at bay, depending on their initial locations. However, if the poacher's speed is more than that of the drone, an interesting result from the analysis done from the simulations shows that it would not be possible for the drone to keep the poacher away from the animal if the poacher plays optimally.

The best that the drone could do in such a case is to go to the animal's location and wait there for the poacher to arrive. To build field-fit drones that would help in anti-poaching operations, the analysis done in this project has to be extended for mobile targets because, in reality, the animals would be moving, as mentioned in the Future Works Section. The effects of obstacles are studied by comparing the dominance regions in the presence of obstacles with the dominance regions in the absence of obstacles, and obstacles are shown to have potential benefits and drawbacks for both the faster and the slower players.

Finally, dominance regions are shown to provide the complete solution to PE games. Though in the simulations, we have introduced only line obstacles, we find that the dominance region based algorithm, ensures that the pursuer, can reach in lesser time than other traditional strategies like the LOS approach. Also, if interception is achieved when the drone gets within a certain proximity of the poacher, calculations for optimal strategies could become intense, and it may not be possible to do such time consuming mathematical computations in real-time. Therefore, algorithms like Rapidly exploring Random Trees (RRT) will have to be explored.

Advantages:

- The use of dominance region based algorithm gives us the minimal time required for the pursuer to capture the evader.
- This strategy works well in the presence of line obstacles and can be extended to other non-linear obstacles easily.
- This concept of dominance regions can be used even when there are multiple pursuers and evaders present in the field as well.

Limitations:

- If the location of the evader or the pursuer is not known, this algorithm will fail.
- This algorithm requires, the pursuer to know the presence of obstacles in advance to compute the minimum time for capture. If the map is not known to the pursuer, the minimum time will definitely increase, as it has to explore the map initially using techniques like SLAM.
- If the communication abilities of the protector and the prey are restricted as in the real world, it may lead to sub-optimal strategies from either side, hence this algorithm may not lead to the minimum time of capture.

SUMMARY

The below results are tabulated for the initial conditions: -

Vp = 3m/s

Ve= 1m/s

(predator) E - Evader starting location = (0,0) (The location where the evader starts is taken as origin)

(protector) P- Pursuer starting location = (20,0)

(prey) T- immobile target location at (0,10)

Line obstacle coordinates {(10,2), (15,4)}

TABLE 1: Results of a particular set of simulations with given initial conditions

S.No.	Strategy Type	Obstacle Present	Minimum Time for Capture
1	LOS	No	7.5 seconds
	Dominance Region based	No	7.1 seconds
2	LOS	Yes	*No Capture Possible*
	Dominance Region based	Yes	7.2 seconds

If we change these initial conditions and run the simulations for these test cases, we find that Dominance region based strategy always gives us the Minimum Time for capture.

This proves, that the theory is correct, based on these experimental simulations, and that the dominance region based algorithm gives us the optimal solution to the game.

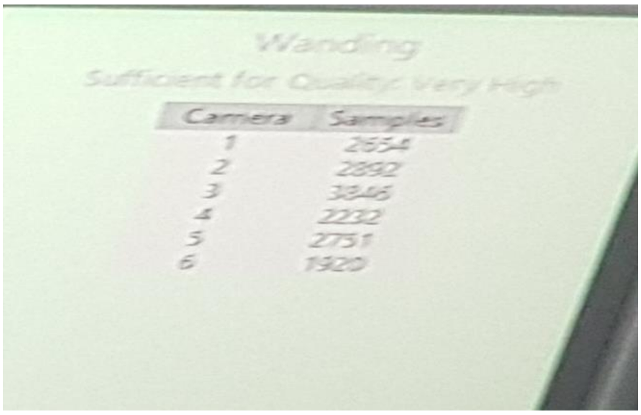
FUTURE WORKS & APPLICATIONS

The MATLAB simulated code can be implemented into actual real-time robots. Using the Opti-track system available at the Control Engineering Lab at IIT-Madras, We can implement the P3 game, and validate the experimental results simulated in MATLAB.

The Opti-track cameras have been calibrated as shown below in the figure, The LegoEv3 Robots have also been designed such that the pursuer moving at 0.4 m/s and the evader moves at 0.2 m/s. In the simulated code, we have pursuer moving at 3 times the evaders speed, here we have implemented the pursuer to have 2 times the evaders speed for more rigorous validation of the algorithm code.

The cameras of the Opti-Track system are calibrated through the process of Wanding.

a)



b)

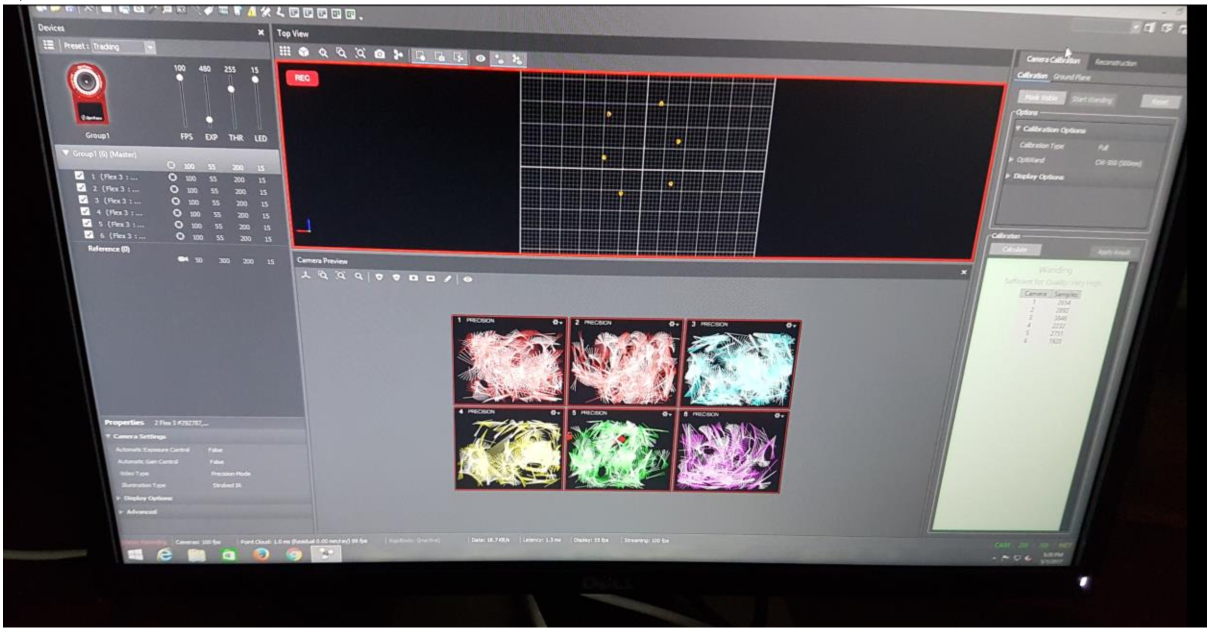


Figure 11: Exceptional Calibration of the Cameras of the Opti-Track System. This system is used to stream the coordinates of the Pursuer and Evader to MATLAB, and we can control the pursuer to follow the time-optimal route, using the algorithm discussed till now.

Figure 12: The cameras mounted gives us an overhead picture of the robot trajectory paths

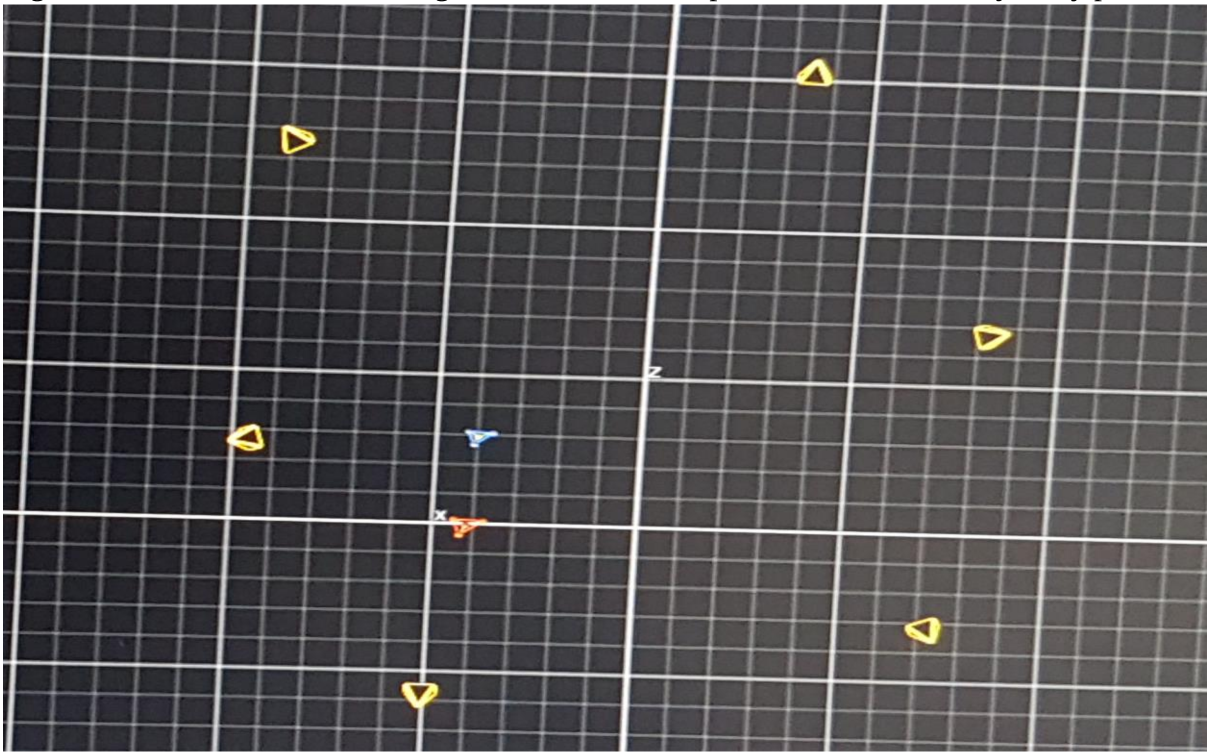


Figure 13: Picture Depicting the Opti-Track system in the Lab. The Blue rigid body is the pursuer. The Red rigid body is the evader. The Yellow Markers represent the cameras of the Opti-Track system

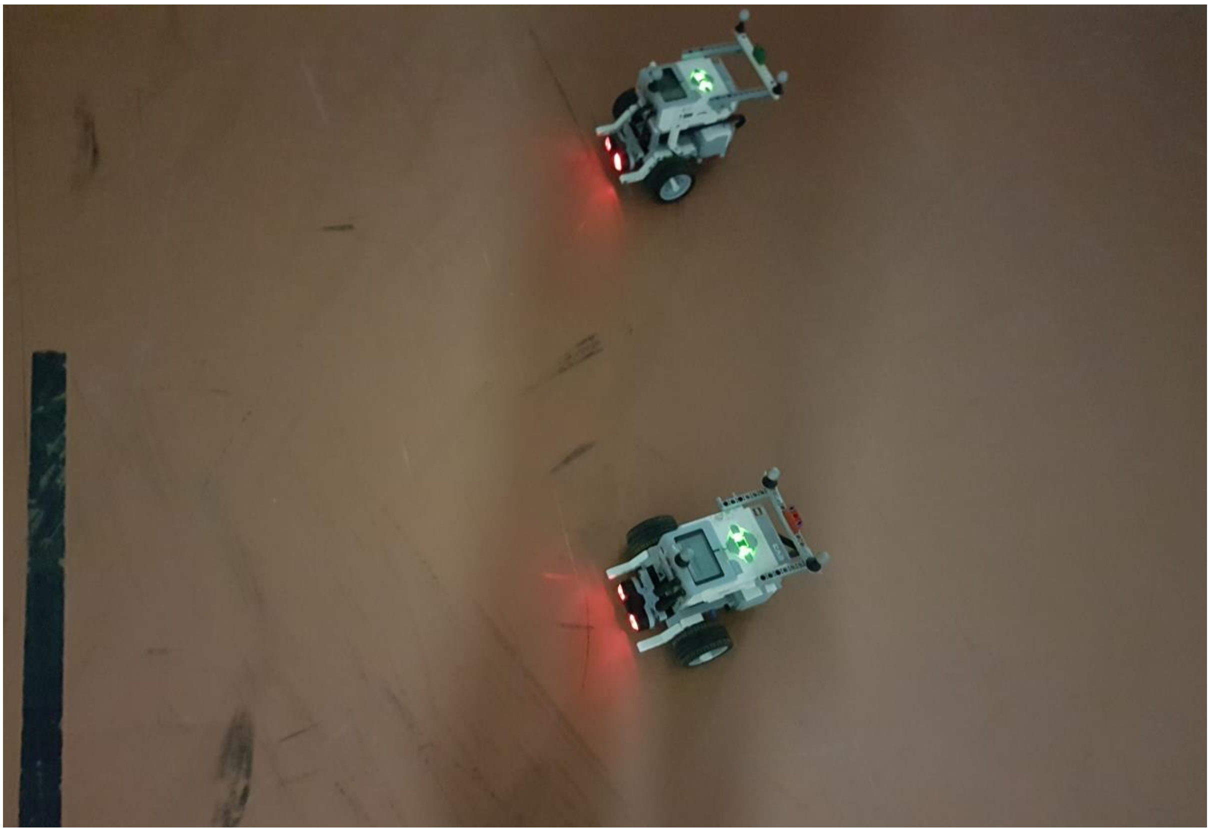


Figure 14: The Green marker represents the pursuer, and the red marker on top represents the evader. The line shown, is the obstacle in place.

- The existing algorithm is for finding the time-optimal path in the case of line obstacles.
- In the future, this algorithm can be improvised for non-linear and polygonal obstacles. This will increase the accuracy of the mathematical model.
- In addition, the simulated MATLAB code, can be implemented into the LEGO EV3 robots shown above, as mentioned in the future works section till now.
- This will help verify the accuracy and robustness of the code to real-world objects.
- By practically implementing the code, we can thus see if the existing algorithm is computationally cheap, or if we need to improvise the algorithm, for non-linear obstacles.
- Further modifications can be done, for the case, where capture occurs, if the pursuer reaches a certain distance of the evader, instead of a strict coincidence of their positions. The importance of this study is that, it is fundamental to building autonomous systems that would be used for protecting a target, for example, unmanned vehicles used in anti-poaching operations.
- The model assumes, the robots turn instantly, this might lead to small errors, this assumption may be relaxed, and the code may be improvised to account for these errors.
- If the computation becomes expensive for nonlinear obstacles, algorithms like RRT can be explored.

APPENDIX A

THE BASIC LOS SOLUTION

MATLAB Code for Simulation:

```
V1=1;
V2=3;
t=0.1;
flag=0;
y1=0;
x2=20;
y2=0;
T=0;
X2=[];
Y1=[];
Y2=[];
X1=[];
while (flag==0)
T=t+T;
X2=[X2 x2];
Y2=[Y2 y2];
Y1=[Y1 y1];
X1=[X1 0];
theta=atan((y1-y2)/x2);
x2=x2-V2*cos(theta)*t;
y2=y2+V2*sin(theta)*t;
y1=y1+V1*t;
if(x2<=0)
flag=1;
end
end
plot(X2,Y2,'o')
hold on
plot(X1,Y1,'o')
```

T – is updated every iteration to keep track of the time of capture or collision between the pursuer and the evader.

APPENDIX B

DOMINANCE REGION BASED STRATEGY

MATLAB Code for Simulation:

a) Dominance region generating code

```
function [x,y,r]=circle(xe,ye,xp,yp,V1,V2)
%x and y are the coordinates of the center of the circle
%r is the radius of the circle
%0.01 is the angle step, bigger values will draw the circle faster but
%you might notice imperfections (not very smooth)

x=((xe*V2*V2)-(xp*V1*V1))/(V2*V2-V1*V1);
y=((ye*V2*V2)-(yp*V1*V1))/(V2*V2-V1*V1);

r=sqrt(x*x + y*y - ((V2*V2*(xe*xe+ye*ye))-(V1*V1*(xp*xp+yp*yp)))/(V2*V2-V1*V1));

ang=0:0.01:2*pi;
xa=r*cos(ang);
ya=r*sin(ang);
plot(x+xa,y+ya);
hold on
end
```

b) Collision and Capture code

```
V1=1;V2=3;
t=0.1;
xe=0;ye=0;
xp=20;yp=0;
flag=0;
x1=0;y1=0;
x2=20;y2=0;
TL=0;
X2=[];
Y1=[];
Y2=[];
X1=[];
while (flag==0)
```

```
TL=t+TL;
X2=[X2 x2];
Y2=[Y2 y2];
Y1=[Y1 y1];
X1=[X1 0];
theta=atan((y1-y2)/x2);
x2=x2-V2*cos(theta)*t;
y2=y2+V2*sin(theta)*t;
y1=y1+V1*t;
if(x2<=0)
flag=1;
end
end
plot(X2,Y2,'o')
hold on
plot(X1,Y1,'o')

[xc,yc,r]=circle(xe,ye,xp,yp,V1,V2);
[TimeD]=predator(r,xc,yc,V1,V2);
```

c) Pursuer Movement Code

```
function [TimeD]=predator(r,xc,yc,V1,V2)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
xf=0;
yf=yc+sqrt(r*r-abs(xf*xf-xc*xc));

XP=[];
YP=[];

Td=0;
t1=0.1;

x2i=20;y2i=0;
flag1=0;

theta=abs(atan((yf-y2i)/(xf-x2i)));

while (flag1==0)
Td=t1+Td;
XP=[XP x2i];
YP=[YP y2i];
x2i=x2i-V2*cos(theta)*t1;
y2i=y2i+V2*sin(theta)*t1;
if(y2i>=yf)
flag1=1;
```

```
end
end

TimeD=Td;
plot(XP,YP,'o');
hold on
end
```

TimeD is used here to measure the time of capture between the pursuer and the evader using the Dominance region based algorithm.

APPENDIX C

OPTIMAL STRATEGY WITH THE PRESENCE OF OBSTACLES

MATLAB Code for Simulation

a) Creating new dominance region

```
function [x,y,r]=circle(xe,ye,xp,yp,V1,V2)
%x and y are the coordinates of the center of the circle
%r is the radius of the circle
%0.01 is the angle step, bigger values will draw the circle faster but
%you might notice imperfections (not very smooth)

x=((xe*V2*V2)-(xp*V1*V1))/(V2*V2-V1*V1);
y=((ye*V2*V2)-(yp*V1*V1))/(V2*V2-V1*V1);

r=sqrt(x*x + y*y - ((V2*V2*(xe*xe+ye*ye))-(V1*V1*(xp*xp+yp*yp)))/(V2*V2-V1*V1));

ang=0:0.01:2*pi;
xa=r*cos(ang);
ya=r*sin(ang);
plot(x+xa,y+ya);
hold on
end
```

b) Introducing Obstacle between Predator & Evader

```
V1=1;V2=3;
xe0=0;ye0=0;
xp0=20;yp0=0;

xobs1=10;yobs1=2;
xobs2=15;yobs2=4;

Xobs=[xobs1,xobs2];
Yobs=[yobs1,yobs2];
plot(Xobs,Yobs);
hold on;

Tpath1=0;Tpath2=0;
```

```
tpath2=0.1;
XPath2=[];
YPath2=[];

xpath2=xp0;ypath2=yp0;
flagp2=0;
theta=(atan((yobs2-ypath2)/(xobs2-xpath2)));
while (flagp2==0)
Tpath2=tpath2+Tpath2;
XPath2=[XPath2 xpath2];
YPath2=[YPath2 ypath2];
xpath2=xpath2-V2*cos(theta)*tpath2;
ypath2=ypath2-V2*sin(theta)*tpath2;
if(ypath2>=yobs2)
flagp2=1;
end
end

plot(XPath2,YPath2,'o');
hold on

xe=xe0;ye=ye0+V1*Tpath2;
xp=xobs2;yp=yobs2;

[xc,yc,r]=circle(xe,ye,xp,yp,V1,V2);
[Time2]=predator(r,xc,yc,V1,V2,xp,yp);

Tpath2=Tpath2+Time2;

% End of 1st type of path

tpath1=0.1;
XPath1=[];
YPath1=[];

xpath1=xp0;ypath1=yp0;
flagp1=0;
theta=(atan((yobs1-ypath1)/(xobs1-xpath1)));

while (flagp1==0)
Tpath1=tpath1+Tpath1;
XPath1=[XPath1 xpath1];
YPath1=[YPath1 ypath1];
xpath1=xpath1-V2*cos(theta)*tpath1;
ypath1=ypath1-V2*sin(theta)*tpath1;
if(ypath1>=yobs1)
flagp1=1;
end
end
```

```
plot(XPath1,YPath1,'o');
hold on

xe=xe0;ye=ye0+V1*Tpath1;
xp=xobs1;yp=yobs1;

[xc,yc,r]=circle(xe,ye,xp,yp,V1,V2);
[Time2]=predator(r,xc,yc,V1,V2,xp,yp);

Tpath1=Tpath1+Time2;

% End of second type of path


t=0.1;
xe=0;ye=0;
xp=20;yp=0;
flag=0;
x1=0;y1=0;
x2=20;y2=0;
TL=0;
X2=[];
Y1=[];
Y2=[];
X1=[];
while (flag==0)
TL=t+TL;
X2=[X2 x2];
Y2=[Y2 y2];
Y1=[Y1 y1];
X1=[X1 0];
theta=atan((y1-y2)/x2);
x2=x2-V2*cos(theta)*t;
y2=y2+V2*sin(theta)*t;
y1=y1+V1*t;
if(x2<=0)
flag=1;
end
end
plot(X2,Y2,'o')
hold on
plot(X1,Y1,'o')

[xc,yc,r]=circle(xe,ye,xp,yp,V1,V2);
[Time2]=predator(r,xc,yc,V1,V2,xp,yp);

% Normal way of going about
```

c) New algorithm for Pursuer Movement

```
function [Time2]=predator(r,xc,yc,V1,V2,xp,yp)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
xf=0;
yf=yc+sqrt(r*r-abs(xf*xf-xc*xc));

XP=[];
YP=[];

T2=0;
t2=0.1;

x2i=xp;y2i=yp;
flag1=0;

theta=abs(atan((yf-y2i)/(xf-x2i)));

while (flag1==0)
T2=t2+T2;
XP=[XP x2i];
YP=[YP y2i];
x2i=x2i-V2*cos(theta)*t2;
y2i=y2i+V2*sin(theta)*t2;
if(y2i>=yf)
flag1=1;
end
end

Time2=T2;
plot(XP,YP,'o');
hold on

end
```

Tpath1 and Tpath2 compute the time required for capture in their respective paths.

We now make the pursuer choose, the path which requires lesser time to travel and capture the evader. Hence, we now have an optimal algorithm for Pursuit-Evasion games in the presence of obstacles.

REFERENCES

1. **Dave W. Oyler, Pierre T. Kabamba 1 , Anouck R. Girard**, Pursuit–evasion games in the presence of obstacles,2015, ELSEVIER, Automatica. University of Michigan, Department of Aerospace Engineering, United States

2. **Raghav Harini Venkatesan, Nandan Kumar Sinha**, The Target Guarding Problem Revisited: Some Interesting Revelations,2014, The International Federation of Automatic Control Cape Town, South Africa. Department of Aerospace Engineering, Indian Institute of Technology

3. **R. Isaacs**. Differential Games: A Mathematical Theory with Application to Warfare and Pursuit, Control and Optimization. John Wiley and Sons, Inc. 1965.

4. **I. Rusnak.** The Lady, the Bandits and the Body Guards - A Two Team Dynamic Game. In Proceedings of the 16th IFAC World Congress. 2005.

5. **G. Lau** and **H. H. T. Liu**. Real-Time Path Planning Algorithm for Autonomous Border Patrol: Design, Simulation, and Experimentation. Journal of Intelligent Robot Systems. July 2013.

6. **Bakolas, E**., & **Tsiotras, P**. (2012). Relay pursuit of a maneuvering target using dynamic voronoi diagrams. Automatica, 48(9), 2213–2220

7. **Basar, T**., & **Olsder, G. J**. (1998). Dynamic noncooperative game theory (2nd ed.). Society for Industrial and Applied Mathematics.

8. **Bhadauria, D., Klein, K., Isler, V., & Suri, S**. (2012). Capturing an evader in polygonal environments with obstacles: The full visibility case. International Journal of Robotics Research, 31(10), 1176–1189.

9. **Bhattacharya, S., Hutchinson, S., & Basar, T**. (2009). Game-theoretic analysis of a visibility based pursuit-evasion game in the presence of obstacles. In American control conference. June (pp. 373–378).

10. **Bopardikar, S. D., Bullo, F., & Hespanha, J. P**. (2008). On discrete-time pursuitevasion games with sensing limitations. IEEE Transactions on Robotics, 24(6), 1429–1439.

11. **Boyell, R. L.** (1976). Defending a moving target against missile or torpedo attack. IEEE Transactions on Aerospace and Electronic Systems, AES-12(4), 522–526.

12. **Chung, T. H., Hollinger, G. A., & Isler, V**. (2011). Search and pursuit-evasion in mobile robotics: A survey. Autonomous Robots, 31(4), 299–316.

13. **Foley, M. H., & Schmitendorf, W. E**. (1974). A class of differential games with two pursuers versus one evader. IEEE Transactions on Automatic Control, 19(3), 239–243.

14. **Getz, W. M., & Pachter, M**. (1981). Two-target pursuit-evasion differential games in the plane. Journal of Optimization Theory and Applications, 34(3), 383–403.

15. **Giovannangeli, C., Heymann, M., & Rivlin, E**. (2010). Pursuit-evasion games in presence of obstacles in unknown environments: Towards an optimal pursuit strategy. In Cutting edge robotics 2010. InTech.

16. **Hershberger, J., & Suri, S**. (1999). An optimal algorithm for euclidean shortest paths in the plane. SIAM Journal on Computing, 28(6), 2215–2256.

17. **Hwang, I., Stipanovic, D. M., & Tomlin, C. J**. (2005). Polytopic approximations of reachable sets applied to linear dynamic games and a class of nonlinear systems. In Systems and control: foundations and applications, Advances in control, communication networks, and transportation systems (pp. 3–19). Boston: Birkhäuser.

18. **Isaacs, R.** (1965). Differential games. New York: John Wiley and Sons.

19. **Jin, S., & Qu, Z.** (2010). Pursuit-evasion games with multi-pursuer vs. one fast evader. In Proceedings of the 8th world congress on intelligent control and automation. WCICA. Jinan, China, July (pp. 3184–3189).

20. **Kabamba, P. T., & Girard, A. R.** (2014). Fundamentals of aerospace navigation and guidance. Cambridge University Press.

21. **Karaman, S., & Frazzoli, E.** (2011). Incremental sampling-based algorithms for a class of pursuit-evasion games. In Springer tracts in advanced robotics: Vol. 68. Algorithmic foundations of robotics IX (pp. 71–87). Berlin, Heidelberg: Springer.

22. **LaValle, S.M., Lin, D., Guibas, L.J., Latombe, J.C., & Motwani, R.** (1997). Finding an unpredictable target in a workspace with obstacles. In Proceedings of the IEEE international conference on robotics and automation, Albuquerque, NM, Vol. 1. April (pp. 737–742).

23. **Merz, A. W.** (1985). To pursue or to evade-that is the question. Journal of Guidance, 8(2), 161–166.

24. **Mitchell, J.S.B.** (1993). Shortest paths among obstacles in the plane. In Proceedings of the ninth annual symposium on computational geometry. San Diego, CA (pp. 308–317).

25. **Oyler, D.W., Kabamba, P.T., & Girard, A.R.** (2014). Pursuit-evasion games in the presence of a line segment obstacle. In 53rd IEEE conference on decision and control, Los Angeles, CA, December (pp. 1149–1154).

26. **Perelman, A., Shima, T., & Rusnak, I**. (2011). Cooperative differential games strategies for active aircraft protection from a homing missile. Journal of Guidance, Control, and Dynamics, 34(3), 761–773.

27. **Rusnak, I**. (2005). The lady, the bandits and the body guards: A two team dynamic game. In Proceedings of the 16th IFAC world congress, Vol. 16, Czech Republic, July (p. 934).

28. **Schaudt, B. F.** (1991). Multiplicatively weighted crystal growth voronoi diagrams. (Ph.D. thesis), Dartmouth College.

29. **Shima, T**. (2011). Optimal cooperative pursuit and evasion strategies against a homing missile. Journal of Guidance, Control, and Dynamics, 34(2), 414–425.

30. **Shinar, J., & Silberman, G.** (1995). A discrete dynamic game modelling anti-missile defense scenarios. Dynamics and Control, 5(1), 55–67.

31. **White, K. S., & Berger, J.** (2001). Antipredator strategies of Alaskan moose: Are maternal trade-offs influenced by offspring activity? Canadian Journal of Zoology, 79(11), 2055–2062.

32**. P.J Nahin** , Chases & Escapes: The Mathematics of Pursuit and Evasion.