Towards Real-Time Autonomous Target Area Protection: Theory and Implementation

Jitesh Mohanan, S.R. Manikandasriram, R. Harini Venkatesan, and B. Bhikkaji, *Member, IEEE*

***Abstract*—This paper considers the Target Guarding Problem (TGP) with a single pursuer P, a single evader E and a stationary target T. The goal of P is to prevent E from capturing T, by intercepting E as far away from T as possible. An optimal solution to this problem, referred to as Command to Optimal Interception Point (COIP), was proposed recently. This guidance law requires the positions of the agents involved. Typically, aerial sensors (such as GPS) used for obtaining these data may not always perform robustly on the field, thereby reducing the autonomy of the vehicles. The computational complexity of the expressions in the COIP law also make it difficult for a real time implementation. Here the TGP is revisited and the optimal solution is reformulated to expressions that are suitable for autonomous systems with ranging sensors mounted on them. These expressions also allow for seamless real-time implementation in robotic hardware. The reformulation enables the optimal solution to be coded as a lookup table requiring minimal memory to further increase the speed of computations. An experimental set up with mobile robots is then used to validate the claims. The case of T lying in E's dominance region is considered a lost game for P. However, this is true only if E plays optimally. If E plays sub-optimally P stands a chance to win the game. This case, which has not been analyzed earlier, is also discussed in this work, and an optimal strategy for P is presented.**

***Index Terms*—Target protection games; Pursuit-evasion games; Region of Dominance; Geometric approaches; Multi-agent systems; Autonomous mobile robots;**

I. Introduction

THIS paper studies optimal strategies for agents in the Target Guarding Problem (TGP). The problem chosen involves two mobile agents: the *pursuer* and the *evader*. Both the pursuer P and the evader E move on a two dimensional planar surface, as shown in Figure 1. The game also features a target, T, which is guarded by the pursuer P. The goal of the evader is to capture the target. The goal of the pursuer is to capture the evader before the target is attacked. It is usually desired that the pursuer intercepts the evader as far away from the target as possible. The dominance region of an agent constitutes the set of points in the play area that can be reached by that agent before the other agent. If the target is present in the dominance region of the pursuer, the game is won by the pursuer by using the optimal strategy. The aim is to derive an optimal strategy (guidance law) for the pursuer to intercept the evader in *real-time*.

This work was partly funded by IIT Madras under the ERP Grant.

Jitesh Mohanan, R. Harini Venkatesan and B. Bhikkaji are with the Department of Electrical Engineering, Indian Institute of Technology Madras, India, 600036 (e-mail: jiteshnov1@gmail.com; raghav.hv@gmail.com; bharath.bhikkaji@iitm.ac.in).

S.R. Manikandasriram is with the Robotics Institute, University of Michigan, Ann Arbor, USA 48105 (e-mail: srmanikandasriram@gmail.com).

An optimal strategy for the pursuer was presented in [1]. It was referred to as Command to Optimal Interception Point (COIP) guidance law. Though this strategy ensures victory for the pursuer P when the target is in P's dominance region, it cannot be easily implemented in real time. Besides that, the original COIP guidance law requires the instantaneous locations of the pursuer and the evader, which may not be obtainable always. For example, when using sensors such as GPS or satellite imaging systems, signals may either be jammed or not made available to the pursuer. Therefore, the COIP law, in its present form, is incapable of being applied in scenarios (for example, on battlefields) where complete autonomy is desired. Here, complete autonomy refers to the pursuer having access to an uninterrupted supply of relevant data, thereby enabling independent decision making throughout the game.

In situations where T lies in E's dominance region, the game is considered lost, assuming E plays optimally. If E plays sub-optimally P must take advantage and maximize his chances of winning.

The key focus of this paper is to derive an optimal strategy for the pursuer that is implementable in *real-time*. The optimal guidance law is reformulated to get to a form that can be coded as a lookup table, enabling an effective real-time implementation. The new form also makes the pursuer completely *autonomous* without having to rely only on global or aerial sensors, all the time. The lookup table based optimal guidance law is then experimentally validated using mobile robots. The case of the target lying in the evader's dominance region is also investigated and the optimal strategy for the pursuer is derived in such a case.

This paper is organized as follows. In Section II problems that are closely related to TGP are discussed. In Section III existing solutions to the TGP are presented. In Section IV, the expressions for the COIP guidance law derived in [1] are presented. In Section V, the problem statement and the contributions of this paper are made clear. A real time implementable form of the COIP law is derived in Section VI. A comparison of the computation speeds of the original COIP, the reformulated COIP and its lookup table implementation is presented in Subsection VII-A. Subsections VII-B and VII-C provides details on the suitability of the reformulated law for real time autonomous agents. Section VIII presents experiments that validate the reformulated COIP and the corresponding lookup table implementation. In Section IX the case where the target T lies in the evader E's dominance region is analyzed and an optimal guidance law for the pursuer P is derived. The paper concludes with a discussion of the results in Section X.

II. Motivation and Related work

The TGP was first studied by Isaacs [2] in his pioneering work on pursuit evasion games. Thereafter, research leading to real-time solutions for such games has been minimal. The primary reason being that the solution for these games are the saddle point equilibria of the Hamilton-Jacobi-Isaacs equation [3]. Determining the equilibrium points, especially if there are more than one, in a real time scenario would be computationally infeasible. More importantly, the solution of the differential game is not guaranteed to exist or be smooth everywhere on the state-space of the game [2].

Games similar to the TGP have also been of interest in several other studies. In [4], a variant of the TGP is analyzed, where the evader does not have information about the position or velocity of the pursuer, but has a better maneuverability than the pursuer. In [5], [6] and [7], games similar to TGP are analyzed with the special case that the pursuer and the target co-operate and form joint strategies for protection from the evader. The authors in [8] analyze the case with a moving target. In a limiting case, it is shown that this game is similar to a game played only by the target and the evader. The guidance laws presented in all of the studies mentioned above do not lend themselves to a seamless real-time implementation. Moreover, most of these studies were done for defense scenarios involving missile and aircraft engagements, the dynamics of which are different from autonomous ground robots protecting a target of interest.

In [9], the authors use Rapidly Exploring Random Trees (RRT) algorithm to generate optimal paths for the agents, for a game similar to the TGP. These techniques, being computationally expensive, restricts the use of the algorithm to slow agents.

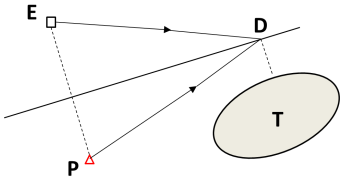
Vehicles that are completely autonomous use a host of on-board sensors for mapping and navigation. However, systems that are reliant only on GPS to estimate positions, suffer from the disadvantages of low resolution, susceptibility to jamming and spoofing, poor signal reception in certain geographies, and unsuitability for underwater measurements [10]. Recent studies on improving mapping algorithms by autonomous vehicles include [11], [12] and [13]. But, the algorithms presented in these studies are computationally expensive and focus on the dynamics of urban traffic, which can be very different from a target protection scenario in a battlefield.

Furthermore, it is not always possible for the pursuer to access the position of the evader using GPS. Techniques such as satellite imaging may prove to be computationally expensive to be processed on-board a mobile robot. This necessitates usage of distance measuring ranging sensors such as ultrasonic sensors, radars or lidars, to estimate the evader position. The original COIP law needs to be reworked to use data provided by these sensors.

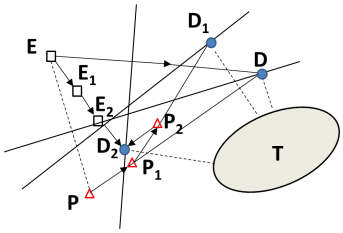
The work in this paper attempts to fill the above mentioned gaps in the literature.

III. The Target Guarding Problem: Existing Solutions

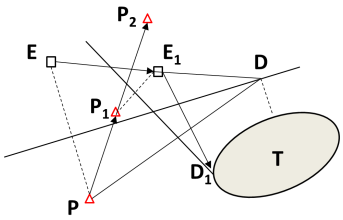
In the first reported analysis of the TGP in literature, Isaacs [2] considered the special case of the velocity of E being equal to that of P, *i.e.*, $v_e = v_p$, as shown in Figure 1. As $v_e = v_p$, the perpendicular bisector of the line joining the initial positions of P and E forms the boundary of their dominance regions. The optimal strategy proposed by Isaacs, for either players, is to head straight to the point D, which is the closest point to the target area T on the perpendicular bisector. If either of them deviates (either by change in *magnitude* or *direction* of velocity) from this optimal strategy, it benefits their opponent, as shown in Figure 1(b) and Figure 1(c).



(a) P and E play optimally



(b) P plays optimally, but E does not



(c) E plays optimally, but P does not

Fig. 1. Target Guarding Problem from [2].

When $v_p > v_e$, [14], the boundary of the dominance region is the Apollonian circle (with center C) shown using a dotted line in Figure 2. The optimal strategies for P and E in this case is to head straight to the point I (interception point), which is the point closest to T on the Apollonian circle. Based on this, the authors also proposed a new guidance law for the pursuer called the Command to Optimal Interception Point (COIP) guidance law [1], which will direct the pursuer to always pursue the point I.

IV. The Command to Optimal Interception Point (COIP) Guidance Law for a Two-Dimensional Scenario

In Figure 2, let P, E and T denote the instantaneous locations of the players, and O be the origin of the coordinate system describing the space in which the game is being played. Therefore, $\overrightarrow{OP}$, $\overrightarrow{OE}$ and $\overrightarrow{OT}$ are their position vectors respectively. Also, let $\frac{v_e}{v_p} = \frac{e}{p}$. Note that v_p and v_e are the instantaneous velocities of the pursuer and evader, which are

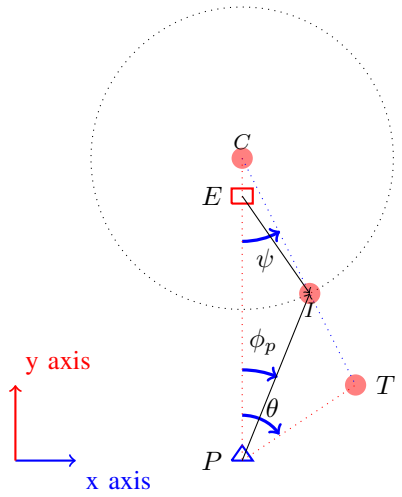


Fig. 2. Optimal strategies for P and E, for a static point target at T ($v_p > v_e$) [14]

assumed to be constant through one iteration of the algorithm. Through the guidance law, the aim is to calculate $\angle EPI = \phi_P$, the instantaneous heading angle off $\overrightarrow{PE}$, as shown in the figure.

For the TGP in two-dimensions, the following steps constitute the COIP guidance law [1]:

$$\overrightarrow{PE} = \overrightarrow{OE} - \overrightarrow{OP} \quad (1)$$
$$\overrightarrow{PT} = \overrightarrow{OT} - \overrightarrow{OP} \quad (2)$$
$$\overrightarrow{PC} = \frac{1}{1-\frac{e^2}{p^2}}\overrightarrow{PE} \quad (3)$$
$$R = \frac{e}{p}|\overrightarrow{PC}| \quad (4)$$
$$\overrightarrow{CT} = \overrightarrow{PT} - \overrightarrow{PC} \quad (5)$$
$$\overrightarrow{CI} = R\frac{\overrightarrow{CT}}{|\overrightarrow{CT}|} \quad (6)$$
$$\overrightarrow{PI} = \overrightarrow{PC} + \overrightarrow{CI} \quad (7)$$
$$\phi_P = \tan^{-1}\left(|\overrightarrow{PI} \times \overrightarrow{PC}|, \overrightarrow{PI}.\overrightarrow{PC}\right) \quad (8)$$

where ϕ_p is the required heading angle.

The speed of calculating the COIP guidance law can be greatly improved by computing (1)-(8) by a single lookup table and eliminating the operations of the cross product and the dot product. Constructing a lookup table for the COIP guidance law in its current form would be difficult, because the distances $|\overrightarrow{PE}|$ and $|\overrightarrow{PT}|$ can take on arbitrarily large values. Also, the co-ordinates of the agents, (1) and (2), are needed to compute the optimal heading angle, requiring a sensor like GPS.

V. The Problem Statement and Contributions

A. The Problem Statement

This paper addresses three problems that form part of the *real-time autonomous* solution for the TGP:

- Find the optimal guidance law (COIP) for the pursuer in the TGP, in real time.
- Reformulate the COIP for implementation on completely autonomous systems.
- Find out the optimal strategy for the pursuer when the target is located in the evader's dominance region.

For all the cases, it is assumed that both the pursuer and evader have all information pertaining to the environment and each other, at all instants.

B. Original Contributions

The original contributions of the paper are as follows:

- A new algorithm for computing the optimal COIP law which is amenable for implementation in real-time, and on completely autonomous systems.
- A lookup table strategy for the optimal COIP law, with minimal memory requirements, enabling a faster real-time implementation.
- Experimental verification of the optimal COIP law on a test bed consisting of mobile robots.
- Lemma 1 and Theorem 2, which give the optimal strategy of the pursuer when the target lies in the evader's dominance region.

Each of the contributions of this paper solves a specific part of the problem statement of interest, and thus are enablers for a real-time autonomous implementation of the solution to the TGP.

VI. Reformulating the COIP Guidance Law

A unique value for the heading angle of the pursuer, ϕ_P can be determined, given the following three variables:

$$k_0 = \frac{|\overrightarrow{PT}|}{|\overrightarrow{PE}|} \quad (9)$$
$$k_1 = \frac{e}{p} \quad (10)$$
$$\theta = \angle EPT \quad (11)$$

Equation (9) determines how far the pursuer is, from the target as compared to the evader, (10) determines the size of the dominance region of the evader, and (11) is the angle between the lines PE and PT, as shown in Figure 2.

The expression of $\phi_P = f(k_0, k_1, \theta)$ is derived as follows: Let the magnitude of vectors be represented without the arrow on the top, for example, $|\overrightarrow{PC}| = PC$, $|\overrightarrow{PE}| = PE$ and so on. Using (3), (4) and (10), PC can be written as

$$PC = \frac{PE}{1-k_1^2} \quad (12)$$
$$R = k_1 PC = \frac{k_1}{1-k_1^2}PE \quad (13)$$

In $\triangle CPI$, using the sine rule,

$$\frac{PI}{\sin\psi} = \frac{CI}{\sin\phi_P} = \frac{R}{\sin\phi_P}$$
$$\implies \sin^2\phi_P = \frac{R^2}{PI^2}\sin^2\psi$$
$$\implies \sin^2\phi_P = \frac{\sin^2\psi}{\left(\frac{PI}{R}\right)^2} \quad (14)$$

In $\triangle CPI$, using the cosine rule,

$$\begin{aligned} PI^2 &= CI^2 + PC^2 - 2.CI.PC.\cos\psi \\ &= R^2 + PC^2 - 2.R.PC.\cos\psi \end{aligned} \quad (15)$$

Dividing (15) by R^2, it is seen that

$$\left(\frac{PI}{R}\right)^2 = 1 + \left(\frac{PC}{R}\right)^2 - 2\left(\frac{PC}{R}\right)\cos\psi \quad (16)$$

Using (12) and (13),

$$\frac{PC}{R} = \frac{1}{k_1} \quad (17)$$

and therefore, (16) becomes

$$\left(\frac{PI}{R}\right)^2 = 1 + \frac{1}{k_1^2} - \frac{2}{k_1}\cos\psi \quad (18)$$

Substituting (18) in (14),

$$\begin{aligned} \sin^2\phi_P &= \frac{\sin^2\psi}{1 + \frac{1}{k_1^2} - \frac{2}{k_1}\cos\psi} \\ &= \frac{k_1^2\sin^2\psi}{k_1^2 - 2k_1\cos\psi + 1} \\ &= \frac{k_1^2\sin^2\psi}{k_1^2 \pm 2k_1\sqrt{1-\sin^2\psi} + 1} \end{aligned} \quad (19)$$

In $\triangle CPT$, using the sine rule,

$$\begin{aligned} \frac{CT}{\sin\theta} &= \frac{PT}{\sin\psi} \\ \implies \sin^2\psi &= \left(\frac{PT}{CT}\right)^2\sin^2\theta \\ \implies \sin^2\psi &= \frac{\sin^2\theta}{\left(\frac{CT}{PT}\right)^2} \end{aligned} \quad (20)$$

In $\triangle CPT$, using the cosine rule,

$$\begin{aligned} CT^2 &= PT^2 + PC^2 - 2.PT.PC.\cos\theta \\ \implies \left(\frac{CT}{PT}\right)^2 &= 1 + \left(\frac{PC}{PT}\right)^2 - 2\left(\frac{PC}{PT}\right)\cos\theta \end{aligned} \quad (21)$$

From (12),

$$\begin{aligned} \frac{PC}{PT} &= \frac{1}{1-k_1^2}\left(\frac{PE}{PT}\right) \\ &= \frac{1}{k_0(1-k_1^2)} \end{aligned} \quad (22)$$

by using (9). Let

$$k_2 = k_0(1-k_1^2). \quad (23)$$

Therefore, (22) becomes

$$\frac{PC}{PT} = \frac{1}{k_2} \quad (24)$$

and substituting (24) in (21),

$$\left(\frac{CT}{PT}\right)^2 = 1 + \frac{1}{k_2^2} - \frac{2}{k_2}\cos\theta \quad (25)$$

Substituting (25) in (20),

$$\begin{aligned} \sin^2\psi &= \frac{\sin^2\theta}{1 + \frac{1}{k_2^2} - \frac{2}{k_2}\cos\theta} \\ &= \frac{k_2^2\sin^2\theta}{k_2^2 - 2k_2\cos\theta + 1} \end{aligned} \quad (26)$$

Let

$$k_3 = k_2^2 - 2k_2\cos\theta + 1. \quad (27)$$

Therefore, (26) becomes

$$\sin^2\psi = \frac{k_2^2\sin^2\theta}{k_3} \quad (28)$$

Substituting (28) in (19),

$$\begin{aligned} \sin^2\phi_P &= \frac{k_1^2k_2^2\sin^2\theta}{k_3\left[k_1^2 \pm 2k_1\sqrt{1-\left(\frac{k_2^2\sin^2\theta}{k_2^2-2k_2\cos\theta+1}\right)}+1\right]} \\ &= \frac{k_1^2k_2^2\sin^2\theta}{k_3\left[k_1^2 \pm \frac{2k_1|k_2\cos\theta-1|}{\sqrt{k_3}}+1\right]} \\ \implies \sin\phi_P &= \sqrt{\frac{k_1^2k_2^2\sin^2\theta}{k_3\left[k_1^2 \pm \frac{2k_1|k_2\cos\theta-1|}{\sqrt{k_3}}+1\right]}} \\ \implies \phi_P &= \sin^{-1}\left(\sqrt{\frac{k_1^2k_2^2\sin^2\theta}{k_3\left[k_1^2 \pm \frac{2k_1|k_2\cos\theta-1|}{\sqrt{k_3}}+1\right]}}\right) \end{aligned} \quad (29)$$

where both k_2 and k_3 are functions of (k_0, k_1, θ) given by (30) and (31) respectively. Although the expression of ϕ_P in (29) appears complex, it is of the form $\phi_P = f(k_0, k_1, \theta)$, using which values of ϕ_P can be tabulated for different values of k_0, k_1 and θ.

Summarizing, given the three variables k_0, k_1 and θ, the expressions used for calculating a unique value for ϕ_P are:

$$k_2 = k_0(1-k_1^2) \quad (30)$$

$$k_3 = k_2^2 - 2k_2\cos\theta + 1 \quad (31)$$

$$\phi_P = \sin^{-1}\left(\sqrt{\frac{k_1^2k_2^2\sin^2\theta}{k_3\left[k_1^2 \pm \frac{2k_1|k_2\cos\theta-1|}{\sqrt{k_3}}+1\right]}}\right) \quad (32)$$

VII. Performance Comparison and Discussion

A. Comparison of Computation Speeds

In order to verify the improved speed of computations for the reformulated COIP law, all three implementations of the law (the original COIP law (1)-(8), the reformulated COIP law (9), (10) and (30) - (32) and the look up table), are compared here.

Since most of these algorithms are used in remote or mobile embedded systems, computations involving trigonometric functions are carried out using look up tables, instead of math libraries. Hence, only basic operations, namely, addition (subtraction), multiplication, division and square roots are considered for comparison purposes.

The basic computations involved in the original and the reformulated laws are summarized in Table I and Table II

respectively. The number of computations involved were determined as discussed below:

1) Equations (1) and (2) each consist of 2 addition operations, one along the x co-ordinate and another along the y co-ordinate.
2) Each evaluation of a vector magnitude, such as in (4) and (6) consists of 1 square root operation, 2 multiplications and 1 addition.
3) Expressions which are evaluated once are not re-evaluated in subsequent steps. For example, in (3), the value of $\frac{e}{p}$ is calculated, hence the same is not re-calculated in (4). In (32), k_1^2 has been already evaluated (in (30)) and so also k_2^2 and $k_2 \cos\theta - 1$ (in (31)).
4) Since $|\overrightarrow{PT}|$, $|\overrightarrow{PE}|$ and θ are measured by sensors on the robot (discussed in the next Subsection), (9) and (10) comprise of only 2 division operations. There are no computations involved in (11).
5) In (32) the square root is calculated using a look up table, because the value within the square root lies in the interval $[0, 1]$.

As an example, consider determining the number of computations in $\overrightarrow{CI}$, (6). The x co-ordinate of $\overrightarrow{CI}$ is

$$CI_x = \left(\frac{R}{\sqrt[2]{CT_x^2 + CT_y^2}} \right) CT_x, \tag{33}$$

where CT_x and CT_y denote the x and y co-ordinates, respectively, of $\overrightarrow{CT}$, while the y co-ordinate CI_y is (33) with CI_x, CT_x and CT_y replaced by CI_y, CT_y and CT_x respectively. Observe that CI_x involves 1 addition (in the denominator), 3 multiplications (2 in the denominator and 1 in the numerator), 1 division (for the entire fraction) and 1 square root, while CI_y involves only 1 multiplication, as the quantity inside the brackets has already been computed. Hence, the total number of computations for $\overrightarrow{CI}$ is 1 addition, 4 multiplications, 1 division and 1 square root. This process has been followed for other equations as well.

Simulations are run on the Intel "Skylake" Core i7-6700 CPU. This processor takes 3 clock cycles for addition, 5 clock cycles for multiplication, $7-27$ clock cycles for division and 27 clock cycles for square root [15]. An average value of 17 clock cycles is assumed for the division process in the comparisons. Thus the original COIP law takes around 258 clock cycles, while the reformulated takes only 143 clock cycles.

Equation No.	Additions	Multiplications	Divisions	Square Root
1	2	0	0	0
2	2	0	0	0
3	1	3	2	0
4	1	3	0	1
5	2	0	0	0
6	1	4	1	1
7	2	0	0	0
8	2	4	1	1
TOTAL	13	14	4	3
Clock Cycles (Intel)	39	70	68	81
Total Clock Cycles	258			

TABLE I
BASIC COMPUTATIONS IN THE ORIGINAL COIP LAW

Equation No.	Additions	Multiplications	Divisions	Square Root
9	0	0	1	0
10	0	1	1	0
30	1	2	0	0
31	2	3	0	0
32	2	4	1	1
TOTAL	5	10	3	1
Clock Cycles (Intel)	15	50	51	27
Total Clock Cycles	143			

TABLE II
BASIC COMPUTATIONS IN THE REFORMULATED COIP LAW

The experiments involved 100 trials of recording the computational time for all the three different implementations of the COIP guidance law, in MATLAB. Note from Figure 3 that the new algorithm and the lookup table is about 1.8 times and 6.7 times faster than the original COIP, respectively.

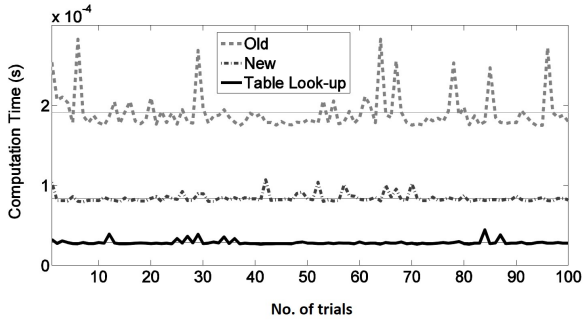


Fig. 3. Comparison of the computational times for the three different implementations of COIP guidance law over 100 trials

B. Real Time Processing and Autonomy

The reformulated COIP law is a function of the variables k_0, k_1 and θ. Variable k_0 can be measured by ranging sensors mounted on the pursuer robot. k_0 can be calculated as the inverse of the times-of-flights t_{PE_f} and t_{PT_f} of its ranging signal between the evader and the target, *i.e.,*

$$k_0 = \frac{t_{PE_f}}{t_{PT_f}} = \frac{PT}{PE}. \tag{34}$$

Since both the pursuer and the evader will always move with their maximum velocities to play optimally, k_1 becomes a known constant. Even in cases where the evader is not moving at his maximum speed (which is playing sub-optimally), the relative velocity between the two moving agents can be measured with the same ranging sensors [16], using which k_1 can be calculated. k_1 is assumed to be a known constant for the rest of this work.

The variable θ can also be measured using the same ranging sensors mounted on the robot. If one sensor is locked to the evader and the other to the target, θ is given by the angle between the orientation of both the sensors (Figure 5).

Hence, the reformulated COIP law can be computed using ranging sensors mounted on the pursuer, instead of aerial sensors, like GPS. The reduced clock cycles in computing the new COIP law also means that the processor draws lesser power from the battery. This keeps the robot running on the field for a longer time, contributing to increased autonomy.

C. Look up Table

As detailed in Section VI, in order to further improve the computational speed of the algorithm, a look up table is used, which depends on the three variables k_0, k_1 and θ. Since k_1 is a constant, the look up table for the reformulated COIP law depends only on k_0 and θ, $i.e.$ two variables. However, a look up table formulation for the original COIP law ((1) - (8)) would depend on four variables: x and y co-ordinates of $\overrightarrow{PI}$ and $\overrightarrow{PC}$. Note that this condition is true even under the assumption of the variable k_1 being a constant. Thus, the order of the look up table for the reformulated COIP law is only half the order for the original COIP law.

Moreover, since k_0 is a ratio of two distances, tabulation of the optimal strategy becomes easier and scalable to larger play areas. Note that the maximum value of k_0 is $\frac{\max|PT|}{\min|PE|}$. The minimum value of $|PE|$ is the sum of the approximate radius of the individual agents. Thus, k_0 is bounded.

Also, the variable θ lies in the interval [0, 360]. For 100 samples of k_0 and 360 samples of θ (a resolution of 1° per sample), the look up table generated for the reformulated COIP law for the experiments consumed only 300KB of memory.

VIII. Experimental Validation

Two differential drive robots built using LEGO EV3 Mindstorms kits (Figure 5) were used as the pursuer and evader robots. The evader robot was manually controlled using a joystick while the proposed guidance law was used to autonomously control the pursuer robot. A stationary target with a predetermined location was used throughout the game. The distances PE and PT were determined using an array of rotating ultrasonic sensors mounted on the pursuer robot. A simple proportional controller is used to maintain the desired orientation of each robot. The velocity of the pursuer was set to be double that of the velocity of the evader. The algorithm for the guidance law was implemented in SIMULINK. The result of the experiment is shown in Figure 4. The co-ordinates of P and E in Figure 4, were determined using an Optitrack motion capture system. These coordinates were used only for validation and were not part of the algorithm.

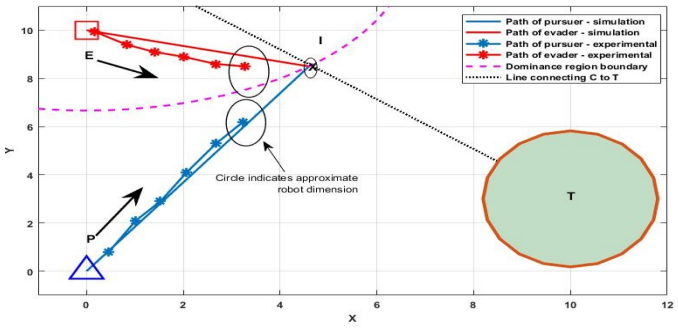


Fig. 4. *P* successfully intercepts *E* before *E* captures *T* - Experimental results

The close match between both the sets of trajectories in Figure 4 verifies that the proposed new algorithm successfully controls the pursuer agent autonomously to intercept the evader before the target is captured. A video demonstrating the above experiment can be seen at [17].

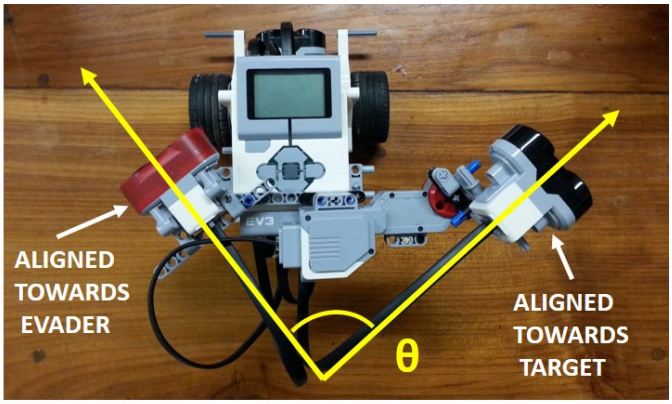


Fig. 5. The LEGO EV3 Robot mounted with rotating ultrasonic sensor array

IX. Guarding a Target Located in the Evader's Dominance Zone

If T were to be located inside E's dominance region, and if E plays sub-optimally, P's strategy should capitalize on this mistake. An example scenario is shown in Figure 7, where E wrongly interprets T to be outside his dominance region. If E operates based on this misinterpretation for sufficiently long, it might be possible for P to protect T, despite starting at a disadvantage. It is in view of this that the following section analyze what should P's strategy be, when T is in E's dominance region.

Lemma 1. *In the TGP game, the pursuer P's strategy is always to move in a direction such that α is minimized, where α is defined as:*

$$\alpha \quad = \quad R - |\overrightarrow{CT}|, \tag{35}$$

where R is the radius of the circle defining the dominance region of the evader, C is the center of the same circle and T is the target location (refer to Figure 6).

Proof. At any instant, let the positions of the players P, E and T be $(x_p, y_p), (x_e, y_e)$ and (x_t, y_t) respectively. Ideally, P would want to move in a direction that makes the target T lie inside his dominance region. This is achieved by minimizing the distance between T and the boundary of the dominance region of both players, as shown in Figure 6. In other words, P minimizes $|TI|$, which is nothing but α. □

Theorem 2. *In the TGP, if the target T lies within the evader E's dominance region, the instantaneous optimal strategy for the pursuer P is be to head towards the interception point I.*

Proof. Let

$$\overrightarrow{\Delta P} \quad = \quad \Delta r \left(\cos\beta\hat{i} + \sin\beta\hat{j}\right) \tag{36}$$

be the vector that results out of P's instantaneous strategy, where Δr is the instantaneous magnitude of P's direction vector, and β is P's heading angle off the positive x-axis, as shown in Figure 6.

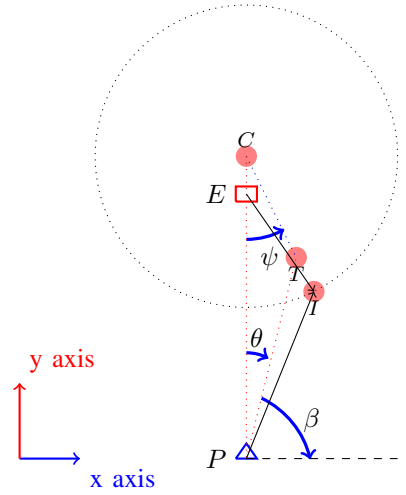


Fig. 6. Target in evader's dominance region

Let $\frac{d}{d\beta}$ be denoted by '$'$'. From Lemma 1, β should be chosen such that:

$$\alpha' = R' - \frac{\overrightarrow{CT}.\overrightarrow{CT}'}{|\overrightarrow{CT}|} = 0 \tag{37}$$

This results in P choosing a heading angle β, where

$$\tan\beta = \frac{y_p - y_i}{x_p - x_i} \tag{38}$$

where (x_p, y_p) is the present co-ordinate of the pursuer and (x_i, y_i) is the co-ordinate of the intercept point I. Derivation of (38) is presented in the Appendix. □

Equation (38) implies that P should move towards the point I, which is the same strategy as the case when T is in P's dominance region.

The case of T lying inside E's dominance region is now simulated. In this case, under optimal conditions, E will be able to capture T before being intercepted by P. This is shown in Figure 7 by dotted line paths.

Now, E follows a sub-optimal path EI instead of ET. This results in P intercepting E before E reaches T. Thus, the target is protected.

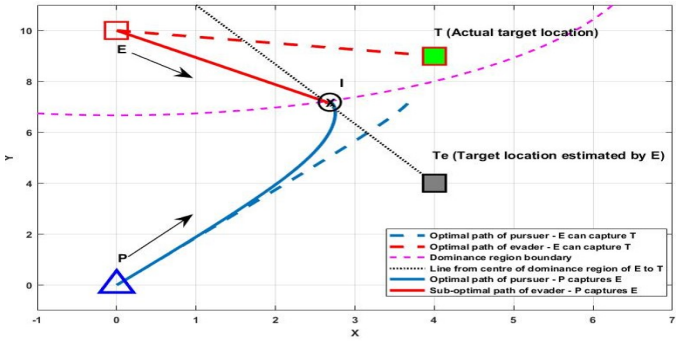


Fig. 7. P intercepts E before E captures T, if E plays sub-optimally

X. Conclusions

This brief studies the TGP with one pursuer, one evader and a stationary target. A new real time algorithm for deriving the instantaneous optimal path for the pursuer has been presented. This algorithm is an improvement to the previously proposed COIP guidance law, because it is almost two times faster. Moreover, the actual co-ordinates of the agents are not needed, obviating the need of global sensors like GPS, thereby making the robots completely autonomous. This also facilitates a look up table implementation of minimal size, which when used for computing optimal heading in real-time, is about seven times faster than the original COIP algorithm. This has been tested and validated on an experimental set-up consisting of LEGO robots.

The case of the target lying in the evader's dominance region was also discussed. The analysis suggests that the optimal heading angle for the pursuer remains the same as that of the case of the target lying inside the pursuer's dominance region.

Appendix
The heading angle of P when T lies in E's dominance region

Using (1) in (3), it is seen that

$$\overrightarrow{PC} = \frac{1}{1-\frac{e^2}{p^2}}(\overrightarrow{OE} - \overrightarrow{OP}) \tag{39}$$

Therefore,

$$\begin{aligned}\overrightarrow{OC} &= \overrightarrow{OP} + \overrightarrow{PC} \\ &= \frac{1}{1-\frac{e^2}{p^2}}\overrightarrow{OE} - \left(\frac{1}{1-\frac{e^2}{p^2}} - 1\right)\overrightarrow{OP}\end{aligned} \tag{40}$$

Let $a = \frac{1}{1-\frac{e^2}{p^2}}$ and $b = \frac{1}{1-\frac{e^2}{p^2}} - 1$. Therefore, (40) becomes

$$\overrightarrow{OC} = a\overrightarrow{OE} - b\overrightarrow{OP} \tag{41}$$

and

$$a = b + 1 \tag{42}$$

$$\frac{e^2}{p^2} = \frac{b}{a} \tag{43}$$

From (4),

$$\begin{aligned}R^2 &= \frac{b}{a}\overrightarrow{PC}.\overrightarrow{PC} \\ &= \frac{b}{a}\left(\overrightarrow{OC} - \overrightarrow{OP}\right).\left(\overrightarrow{OC} - \overrightarrow{OP}\right) \\ &= \frac{b}{a}\left(|\overrightarrow{OC}|^2 + |\overrightarrow{OP}|^2 - 2\overrightarrow{OP}.\overrightarrow{OC}\right) \\ &= |\overrightarrow{OC}|^2 + \frac{b}{a}\left(|\overrightarrow{OC}|^2 + |\overrightarrow{OP}|^2 - 2\overrightarrow{OP}.\overrightarrow{OC}\right) \\ &\quad -|\overrightarrow{OC}|^2\end{aligned} \tag{44}$$

Substituting (41) for $\overrightarrow{OC}$ in the second and third terms on the R.H.S in (44),

$$R^2 = |\overrightarrow{OC}|^2 + b|\overrightarrow{OP}|^2 - a|\overrightarrow{OE}|^2 \tag{45}$$

Now, $$\overrightarrow{\Delta P} = \Delta r\left(\cos\beta\hat{i} + \sin\beta\hat{j}\right) \tag{46}$$

is the vector that results out of P's instantaneous strategy. This implies

$$\overrightarrow{OP}_{\text{new}} = \overrightarrow{OP} + \overrightarrow{\Delta P} \tag{47}$$

Let $\frac{d}{d\beta}$ be denoted by ''. From (46), the objective for P is to choose β to minimize (35), which means β should be chosen such that:

$$\begin{aligned} \alpha' &= R' - \frac{\overrightarrow{CT}.\overrightarrow{CT}'}{|\overrightarrow{CT}|} = 0 \qquad (48)\\ \implies & R' - \frac{\overrightarrow{CT}.\left(\overrightarrow{OT}' - \overrightarrow{OC}'\right)}{|\overrightarrow{CT}|} = 0 \\ \implies & R' + \frac{\overrightarrow{CT}.\overrightarrow{OC}'}{|\overrightarrow{CT}|} = 0 \\ \implies & R' = -\frac{\overrightarrow{CT}.\overrightarrow{OC}'}{|\overrightarrow{CT}|} \qquad (49) \end{aligned}$$

because $\overrightarrow{OT}' = \overrightarrow{0}$ as the target is not moving.

Differentiating (45) w.r.t β,

$$RR' = \overrightarrow{OC}.\overrightarrow{OC}' + b\overrightarrow{OP}.\overrightarrow{OP}' - a\overrightarrow{OE}.\overrightarrow{OE}' \tag{50}$$

Substituting (49) in (50) and rearranging,

$$\left(\overrightarrow{OC} + R\frac{\overrightarrow{CT}}{|\overrightarrow{CT}|}\right).\overrightarrow{OC}' = a\overrightarrow{OE}.\overrightarrow{OE}' - b\overrightarrow{OP}.\overrightarrow{OP}' \tag{51}$$

But from (6), $R\frac{\overrightarrow{CT}}{|\overrightarrow{CT}|} = \overrightarrow{CI}$, and therefore, (51) becomes

$$\overrightarrow{OI}.\overrightarrow{OC}' = a\overrightarrow{OE}.\overrightarrow{OE}' - b\overrightarrow{OP}.\overrightarrow{OP}' \tag{52}$$

Substituting (41) in (52),

$$\overrightarrow{OI}.\left(a\overrightarrow{OE}' - b\overrightarrow{OP}'\right) = a\overrightarrow{OE}.\overrightarrow{OE}' - b\overrightarrow{OP}.\overrightarrow{OP}' \tag{53}$$

But $\overrightarrow{OE}' = \frac{d\overrightarrow{OE}'}{d\beta} = 0$. Therefore, (53) becomes

$$b\left(\overrightarrow{OP} - \overrightarrow{OI}\right).\overrightarrow{OP}' = 0 \tag{54}$$

Since $b \neq 0$,

$$\begin{aligned} \left(\overrightarrow{OP} - \overrightarrow{OI}\right).\overrightarrow{OP}' &= 0 \qquad (55)\\ \implies \left(\overrightarrow{OP}_{\text{new}} - \overrightarrow{OI}_{\text{new}}\right).\overrightarrow{OP}'_{\text{new}} &= 0 \qquad (56) \end{aligned}$$

From (47), $\overrightarrow{OP}'_{\text{new}} = \frac{d}{d\beta}\overrightarrow{\Delta P}$, as $\frac{d}{d\beta}\overrightarrow{OP} = 0$.

Therefore, using (46)

$$\overrightarrow{OP}'_{\text{new}} = \frac{d}{d\beta}\overrightarrow{\Delta P} = \Delta r\left(-\sin\beta\hat{i} + \cos\beta\hat{j}\right) \tag{57}$$

In the limiting case of $\Delta r \to 0$, $\overrightarrow{OP}_{\text{new}} \to \overrightarrow{OP}$ and $\overrightarrow{OI}_{\text{new}} \to \overrightarrow{OI}$. Using these and (57) in (56),

$$\left(\overrightarrow{OP} - \overrightarrow{OI}\right).\Delta r\left(-\sin\beta\hat{i} + \cos\beta\hat{j}\right) = 0 \tag{58}$$

Now, since $\Delta r \to 0$ ($\Delta r \neq 0$), (58) becomes

$$\left[(x_p - x_i)\hat{i} + (y_p - y_i)\hat{j}\right].\left(-\sin\beta\hat{i} + \cos\beta\hat{j}\right) = 0 \tag{59}$$

Therefore,

$$\begin{aligned} -(x_p - x_i)\sin\beta + (y_p - y_i)\cos\beta &= 0 \qquad (60)\\ \implies \tan\beta &= \frac{y_p - y_i}{x_p - x_i} \qquad (61) \end{aligned}$$

REFERENCES

[1] R. Harini Venkatesan and N. K. Sinha, "A new guidance law for the defense missile of non-maneuverable aircraft," *Control System Technology, IEEE Transactions on*, vol. 23, no. 6, pp. 2424–2431, 2015.

[2] R. Isaacs, *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. New York: John Wiley and Sons, Inc., 1965.

[3] G. J. O. Tamer Basar, *Dynamic Noncooperative Game Theory*. Philadelphia: SIAM., 1999.

[4] J. Shinar and G. Silberman, "A discrete dynamic game modelling anti-missile defense scenarios," *Dynamics and Control*, vol. 5, no. 1, pp. 55–67, 1995.

[5] A. Perelman, T. Shima, and I. Rusnak, "Cooperative differential games strategies for active aircraft protection from a homing missile," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 3, pp. 761–773, 2011.

[6] V. Shaferman and T. Shima, "Cooperative multiple-model adaptive guidance for an aircraft defending missile," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 6, pp. 1801–1813, 2010.

[7] T. Shima, "Optimal cooperative pursuit and evasion strategies against a homing missile," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 2, pp. 414–425, 2011.

[8] I. Rusnak, H. Weiss, and G. Hexner, "Guidance laws in target-missile-defender scenario with an aggressive defender," in *Proceedings of the 18th IFAC World Congress*, vol. 18. Elsevier Milan, 2011, pp. 9349–9354.

[9] G. Lau and H. Liu, "Real-time path planning algorithm for autonomous border patrol: Design, simulation and experimentation," *Journal of Intelligent and Robotic Systems*, pp. 517–539, 2014.

[10] Rickard Karlsson and Fredrik Gustafsson, "The future of automotive localization algorithms: Available, reliable, and scalable localization: Anywhere and anytime," *IEEE Signal Processing Magazine*, pp. 60 – 69, 2017.

[11] A. D. McIntyre, W. Naeem and S. S. A. Ali, "Underwater surveying and mapping using rotational potential fields for multiple autonomous vehicles," in *IEEE International Conference on Underwater System Technology: Theory and Applications (USYS), Malaysia*. IEEE, 13-14 December 2016.

[12] X. W. Liang Wang, Jun Wang and Y. Zhang, "3d-lidar based branch estimation and intersection location for autonomous vehicles," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 11-14 June 2017.

[13] A. N. Catapang and M. Ramos, "Obstacle detection using a 2d lidar system for an autonomous vehicle," in *Control System, Computing and Engineering (ICCSCE), 2016 6th IEEE International Conference on*. IEEE, 25-27 November 2016.

[14] R. Harini Venkatesan and N. K. Sinha, "The target guarding problem revisited: Some interesting revelations," in *Proceedings of the 19th World IFAC Congress 2014, Cape Town, South Africa*. IFAC, 24-29 August 2014, pp. 1556–1561.

[15] A. Fog, "Instruction Tables," http://www.agner.org/optimize/instruction_tables.pdf, 2017.

[16] M. Darms, P. Rybski, C. Baker, and C. Urmson, "Obstacle detection and tracking for the urban challenge," *IEEE Transactions on Intelligent Transportation Systems, Vol.10, No.3*, pp. 475–485, 2009.

[17] J. Mohanan, "Real Time Autonomous Target Area Protection," https://youtu.be/7GoMwNeDj0w, 2017.