



**HAL**  
open science

## Limited Information Model Predictive Control for Pursuit-evasion Games

Mukhtar Sani, Bogdan Robu, Ahmad Hably

► **To cite this version:**

Mukhtar Sani, Bogdan Robu, Ahmad Hably. Limited Information Model Predictive Control for Pursuit-evasion Games. CDC 2021 - 60th IEEE Conference on Decision and Control, Dec 2021, Austin, United States. 10.1109/CDC45484.2021.9683016 . hal-03412855

**HAL Id: hal-03412855**

**<https://hal.science/hal-03412855>**

Submitted on 3 Nov 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Limited Information Model Predictive Control for Pursuit-evasion Games

Mukhtar Sani<sup>1,2</sup>, Bogdan Robu<sup>1</sup>, Ahmad Hably<sup>1</sup>

<sup>1</sup> Univ. Grenoble Alpes, CNRS, Grenoble INP\*, GIPSA-Lab, France

\*Institute of Engineering Univ. Grenoble Alpes

{mukhtar.sani,bogdan.robu,ahmad.hably}@gipsa-lab.grenoble-inp.fr

<sup>2</sup> Kano University of Science and Technology, Wudil, Nigeria

mukhtarsani@kustwudil.edu.ng

**Abstract**—This paper explores the use of model predictive control (MPC) in dealing with the pursuit-evasion game (PEG) problem where players have incomplete information on their opponents. This is different from most cases in the literature where each player knows all the information (states information and dynamics) on the opponent. The burden caused by such demand for the opponent’s full information induces the need for more sensors during physical implementation as well as high computation time. However, we found that only the current positions, i.e.  $x-y$  coordinate of the opponent, are indispensable. Thus, knowing the orientation and the dynamics of the opponent are insignificant to the performance of the game. We propose a new method to exploit a two-player PEG in the presence and absence of obstacles, where each player can only rely on the current position information of its opponent. Several simulation results show that the PEG problem can be handled and obstacles can be avoided using the proposed control protocol. We also show that our approach is robust to measurement noise and can perform better, in terms of the computation than the approach with full information.

**Index Terms**—MPC, Obstacles avoidance, Pursuit Evasion Games, Wheeled Robots

## I. INTRODUCTION

Pursuit and Evasion are non-cooperative game problems that are recently receiving a lot of attention especially in the field of mobile robotics. This was motivated by its numerous applicability in both military and civil applications. The game was basically concerned with solving a dynamic problem with a pursuer or group of pursuers in one team and an evader or a group of evaders in another team. The objective of one team conflicts with the objective of the other team. The formal solution of this type of problem, based on dynamic programming and variational techniques, was presented in [1] with several areas of applications and examples. Many other solutions are presented in the literature such as [2], [3], [4] and [5] which are based on safe-reachable area minimization using Voronoi partition for both holonomic and non-holonomic robots. The safe-reachable area method focuses on cooperative pursuit cases where a group of pursuers cooperate to capture a team of evading players.

A classification-based solution depending on the number of players, i.e. one-pursuer-one-evader, N-pursuers-one-evader, one-pursuer-M-evader, and N-pursuers-M-evaders were summarised in [6]. Paper [7] deals with the cooperative two-

pursuer one-evader differential game using a geometric approach. Based on the Hamiltonian formulation and the geometric solution using the point of intersection of two Apollonian circles, a candidate value function and its corresponding optimal strategy are obtained. However, all these approaches require current states information of the opponent to compute the optimal strategy of each player, but could not predict the future behavior of the opponent. In addition to its intricate computation, especially for nonholonomic systems, these methods could not incorporate obstacles avoidance.

Model predictive Control approaches are used to handle pursuit-evasion games in [8], [9] and [10] due to their capacity of handling constraints. In [8], Non-linear Model Predictive Tracking Control (NMPTC) was employed to deal with symmetric pursuit-evasion games for unmanned aerial vehicles. The game was symmetric in the sense that a pursuer can turn to be an evader while the evader can turn to be a pursuer. Both players are assumed to know all the information on the opponent. In [9], NMPC was used for pursuit-evasion between two heterogeneous players. The evading player is an unmanned aerial vehicle while the pursuing player is an unmanned ground vehicle. The paper considers the relative distance and orientation between the two players as states of the game and assumed that the players know full information on the opponent, no states constraints or obstacle avoidance can therefore be incorporated. Our previous paper [10] has considered current states information as the states of each player and was able to incorporate both states constraints and obstacles avoidance. Also, full information on the opponent is assumed to be known to each player. However, high computation time is consumed due to the double computation while searching for Nash equilibrium which is problematic during physical implementation.

The assumption of knowing the full on the opponent information by each player as in the case of [8], [9] and [10] is not feasible in real situations because each player will not reveal its full information. It is therefore pertinent to search for approaches that depend on the measured information. Also, during our physical experiments on pursuit-evasion games in Gipsa-lab, we have found that it is easier to obtain the position of players rather than obtain their orientation. Consequently,

we propose an approach for handling pursuit-evasion games such that each player only needs to obtain the current position of the opponent. The game-theoretic aspect of this approach is that the orientation of the opponent is predicted in advance using the position of both players as derived in [7] for a case of one pursuer one evader. For this purpose, we present the game description and all the preliminaries in Section II. The control algorithms are described in Section III, while the simulation results are presented in Section IV. Finally, we presented some conclusions and directions for future work in Section V.

## II. PRELIMINARIES

### A. Problem Statement

Consider a pursuit-evasion game between two nonholonomic robots of unicycle type, conducted in the interior of a polytope,  $\Omega \in R^2$  in the presence and absence of obstacles. The aim of the pursuer is to capture the evader in a shortest time possible by moving towards the evader while the aim of the evader is to avoid the pursuer or at worst maximize the capture time. Assuming that the pursuer is faster than the evader, capturing is guaranteed in finite time. Both players should have the ability to avoid an obstacle in the environment. The kinematic model of each wheeled mobile robot as represented in [11] is given by equation (1) below:

$$\begin{cases} \dot{x}_i &= v_i \cos \theta_i \\ \dot{y}_i &= v_i \sin \theta_i \\ \dot{\theta}_i &= \omega_i \end{cases} \quad (1)$$

The subscript  $i$  stands for  $i^{th}$  player such that  $i \in \{p, e\}$ , denotes pursuer and evader, respectively. The states  $(x_i, y_i) \in \Omega$  and  $\theta_i$  are the position and orientation of  $i^{th}$  player, respectively. The control inputs for the  $i^{th}$  player are the linear and the angular speeds, which are denoted as  $(v_i, \omega_i) \in U_i$ , respectively.  $U_i$  denotes the set of feasible control inputs for the  $i^{th}$  player, which is assumed to be bounded.

The relative distance between the two players is defined as:

$$r_{r.d}(t) = \sqrt{(x_p(t) - x_e(t))^2 + (y_p(t) - y_e(t))^2} \quad (2)$$

The capture condition can then be defined as:

$$r_{r.d}(t_c) \leq r_c, \quad t_c \geq 0 \quad (3)$$

where  $r_c$  is the capture radius (which is equal to the sum of the radii of the two robots) and  $t_c$  is the capture time.

a) *Pursuit Problem:* Find a set of feasible pursuer inputs  $u_p \in U_p$  such that the capture time is satisfied in finite time irrespective of the evader's strategy. Given an initial condition of the pursuer  $x_p(0), y_p(0), \theta_p(0) \in \Omega$ , the initial condition of the evader  $x_e(0), y_e(0)$  and that  $r_{r.d}(0) > r_c$ .

b) *Evasion Problem:* Find a set of feasible evader inputs  $u_e \in U_e$  such that the capture time is maximized irrespective of the pursuer's strategy. Given an initial condition of the evader  $x_e(0), y_e(0), \theta_e(0) \in \Omega$ , the initial condition of the pursuer  $x_p(0), y_p(0)$  and that  $r_{r.d}(0) > r_c$ .

### B. Nash Equilibrium

Game-theory modelize systems as intelligent rational decision-makers where an agent considers the opponent's strategy before deciding its strategy. The agent predicts the opponent's best response, which is the worst case from his point of view, and then computes its optimal strategy. The relationship between game theory and control was reviewed in [12]. A min-max dynamic games can be expressed as:

$$J_p(U_p, U_e) + J_e(U_p, U_e) = 0 \quad \forall U_p \in \mathcal{U}_p, U_e \in \mathcal{U}_e \quad (4)$$

where  $[J_p, J_e]$ ,  $[U_p, U_e]$  and  $[\mathcal{U}_p, \mathcal{U}_e]$  are the cost function, the control strategy and the admissible control strategies for the pursuer and evader respectively. The solution can be expressed as a double optimization:

$$\phi(U^{p*}, U^{e*}) = \min_{U_p} \max_{U_e} \phi(U^p, U^e) = \max_{U_e} \min_{U_p} \phi(U^p, U^e) \quad (5)$$

The strategy pair  $[U_p^*, U_e^*]$  is a Nash equilibrium if no player has an incentive to deviate.

$$\phi(U_p^*, U_e') \leq \phi(U_p^*, U_e^*) \leq \phi(U_e', U_e^*) \quad (6)$$

### C. Model Predictive Control

Model Predictive Control (MPC) is an online algorithm for computing sequence of optimal control variables using predicted system model. The system model can be linear or non-linear which translate to Linear Model Predictive Control (LMPC) and Non-linear Model Predictive Control (NMPC), respectively. MPC can be implemented by minimizing a quadratic cost function subject to some constraints which comprises the model of the system, state and input constraints. The output of the controller is a sequence of open loop controls predicted ahead over a prediction horizon,  $N$ . The first element of the sequence is applied to the system while the rest of the solutions are discarded. At every decision instant, the prediction horizon is shifted one step and the process is repeated to obtain the new optimal control sequence [9], [10], [13] and [14].

Mathematically, general MPC problem can formulated in discrete form to solve the following cost function:

$$\min_{\mathbf{u}} J = \sum_{k=0}^N \|\mathbf{x}(k) - \mathbf{r}(k)\|_Q^2 + \|\mathbf{u}(k)\|_R^2 \quad (7)$$

subject to the following constraints on the system's dynamics, states and inputs:

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k)), \quad k = 0, 1, \dots, N \quad (8a)$$

$$\mathbf{x}_{min} \leq \mathbf{x}(k) \leq \mathbf{x}_{max} \quad (8b)$$

$$\mathbf{u}_{min} \leq \mathbf{u}(k) \leq \mathbf{u}_{max} \quad (8c)$$

The reference signal  $\mathbf{r}(k)$  can be point vector or a trajectory, while the state variable  $\mathbf{x}(k)$  is a sequence of future trajectory of the system predicted using the system model in (8a).  $N$  is the prediction horizon,  $k$  stands for the instantaneous discrete time while  $Q$  and  $R$  are the states and control weighting matrices. The constraints in (8b) and (8c) represent states

and inputs bounds, respectively. At every decision instant, the systems are re-initialised using new measurement of states and references.

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (9a)$$

$$\mathbf{r}(0) = \mathbf{r}_0 \quad (9b)$$

The output is a sequence of predicted optimal controls:

$$\mathbf{u} = [\mathbf{u}(k), \mathbf{u}(k+1), \dots, \mathbf{u}(N)]^T \quad (10)$$

The first part  $\mathbf{u} = \mathbf{u}(k)$  is applied to the system, while the others are discarded. The controller parameters  $N$ ,  $Q$ , and  $R$  can be tuned to stabilize the system depending on the relative importance of the states and on the computation time.

### III. CONTROL STRATEGIES

#### A. Limited Information Model Predictive Control

As explained in the previous section, we want to use MPC technique to solve pursuit-evasion problem for nonholonomic mobile robots where each player has incomplete states information on the opponent.

1) *Prediction of Opponent's Heading Angle:* In order to obtain the heading angle of the opponent of each player, which is not available a priori, we can use the current position of the players to compute the optimal heading of the opponent. For this, we employ the optimal strategies for computing the heading angle of a player as presented in [3] and [7]. The method in [3] proposes pursuit and evasion strategies for holonomic systems using the concept of Voronoi partitions where two or more players cooperate to capture an evader assuming all the players have the same speed. The optimal heading angle for each pursuer is derived as:

$$\theta_i^* = \arctan\left(\frac{y_{ci} - y_i}{x_{ci} - x_i}\right) \quad (11)$$

While the optimal heading angle of the evader is derived as:

$$\theta_e^* = \pi + \arctan\left(\frac{y_{ci} - y_e}{x_{ci} - x_e}\right) \quad (12)$$

where  $(y_{ci}, x_{ci})$  is the center of the shared boundary. This same strategy was proposed in [7] to provide a solution to a cooperative pursuit-evasion game between with two faster pursuers and one evader using Hamiltonian formulation and the geometric properties of the game. The pursuers cooperate to capture a slower evader in minimum time. In this method, the cooperative strategy is only employed if none of the two pursuers can capture the evader faster than the other. The pursuit and evasion strategies are proposed based the positions of the players. Since our PEG involves one pursuer and one evader, the center of the shared boundary would be regarded as the position of the opponent. We can therefore use this method in a reciprocal manner to predict the optimal heading angle of the opponents.

2) *Pursuer's Controller:* The pursuer's controller can be designed to solve the pursuit problem using NMPC techniques as explained in section II-C. Assuming that the states of the pursuer and the position of the evader can be measured at every decision instant. The optimal control can be obtained by solving the following minimization cost function:

$$\min_{U_p} J(U_p, U_e^*) \quad (13)$$

Subject to:

$$X_p(k+1) = f_p(X_p(k), U_p(k)), \quad k = 0, 1, \dots, N \quad (14a)$$

$$\sqrt{(x_p^{k+1} - x_{obs})^2 + (y_p^{k+1} - y_{obs})^2} \geq (R_{obs} + R_{rob}) \quad (14b)$$

$$X_{p_{min}} \leq X_p(k) \leq X_{p_{max}} \quad (14c)$$

$$U_{p_{min}} \leq U_p(k) \leq U_{p_{max}} \quad (14d)$$

where

$$J = \|X_p(N) - X_e\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|X_p(k) - X_e\|_Q^2 + \|U_p(k)\|_R^2 \quad (15)$$

$X_e$  is a vector of evader's current position and it's predicted orientation which is computed by the pursuer in advance using (11). The first segment of the cost function (13) was added to stabilize the controller by tuning the weighting matrix  $Q_N$ . The constraint (14a) is the pursuer's predicted trajectory over the prediction horizon, (14b) is the obstacle avoidance constraint which depends on the position and radius of both the pursuer and the obstacle, respectively, while (14c) and (14d) are the pursuer's states and input constraints, respectively.

3) *Evader's Controller:* It can be designed to solve the evasion problem also using NMPC techniques, assuming that the states of the evader and the position of the pursuer can be measured at every decision instant. The optimal control can be obtained by solving the following maximization cost function:

$$\max_{U_e} J(U_e, U_p^*) \quad (16)$$

Subject to:

$$X_e(k+1) = f_e(X_e(k), U_e(k)), \quad k = 0, 1, \dots, N \quad (17a)$$

$$\sqrt{(x_e^{k+1} - x_{obs})^2 + (y_e^{k+1} - y_{obs})^2} \geq (R_{obs} + R_{rob}) \quad (17b)$$

$$X_{e_{min}} \leq X_e(k) \leq X_{e_{max}} \quad (17c)$$

$$U_{e_{min}} \leq U_e(k) \leq U_{e_{max}} \quad (17d)$$

where

$$J = \|X_e(N) - X_p\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|X_e(k) - X_p\|_Q^2 + \|U_e(k)\|_R^2 \quad (18)$$

$X_p$  is a vector of pursuer's current position and its predicted orientation which is computed by the evader in advance using (12). Similarly, the first part of the cost function (16) was added to stabilize the controller by tuning the weighting matrix  $Q_N$ . The constraint (17a) is the evader's predicted trajectory

over the prediction horizon, (17b) is the obstacle avoidance constraint which depends on the position and radius of both the evader and the obstacle, respectively while (17c) and (17d) are the evader's states and input constraints, respectively.

### B. Complete Information Model Predictive Control

In this method, as detailed in our previous paper [10], each player uses full information of the opponent to predict the next move. The solution of problem 1 (pursuer's controller) starts by predicting the evader's next move. The following optimal control problem is solved:

$$\max_{U_e} J(U_p, U_e) \quad (19)$$

subject to:

$$\begin{cases} X_e(k+1) = f_e(X_e(k), U_e(k)), & k = 0, 1, \dots, N-1 \\ \sqrt{(x_e^{k+1} - x_{obs})^2 + (y_e^{k+1} - y_{obs})^2} \geq (R_{obs} + R_{rob}) \\ X_{e_{min}} \leq X_e(k+1) \leq X_{e_{max}} \\ U_{e_{min}} \leq U_e(k+1) \leq U_{e_{max}} \end{cases} \quad (20)$$

where

$$J = \|X_e(N) - X_p\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|X_e(k) - X_p\|_Q^2 + \|U_e(k)\|_R^2 \quad (21)$$

The output of this computation  $X_e^*$  is used as reference to the following optimal control problem:

$$\min_{U_p} J(U_p, U_e) \quad (22)$$

subject to:

$$\begin{cases} X_p(k+1) = f_p(X_p(k), U_p(k)), & k = 0, 1, \dots, N-1 \\ \sqrt{(x_p^{k+1} - x_{obs})^2 + (y_p^{k+1} - y_{obs})^2} \geq (R_{obs} + R_{rob}) \\ X_{p_{min}} \leq X_p(k+1) \leq X_{p_{max}} \\ U_{p_{min}} \leq U_p(k+1) \leq U_{p_{max}} \end{cases} \quad (23)$$

where

$$J = \|X_p(N) - X_e^*\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|X_p(k) - X_e^*\|_Q^2 + \|U_p(k)\|_R^2 \quad (24)$$

The output of this computation is the pursuer's Nash equilibrium strategy, i.e.  $U_p^* = [v; \omega]^T$  which stands for the linear and angular speeds respectively.

The solution of problem 2 (Evader's controller) is computed in an similar but opposite format.

## IV. NUMERICAL RESULTS

The proposed control technique was implemented on MATLAB/Simulink environment. An open source symbolic framework for automatic differentiation and optimal control software called CasADi [15], was used for coding the controllers by converting the optimal control problems into nonlinear programming problems using multiple shooting approach. Another open source software, the Interior point optimizer

(IPOPT), was interfaced to provide a solution.

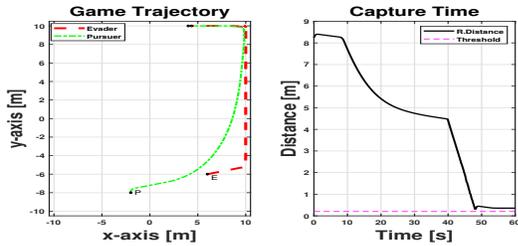
Several games are played in the presence and absence of obstacles. In each scenario (either with or without obstacles), the pursuer position is behind the evader with random initial conditions. The simulation environment is a squared polytope with a dimension of  $20m$  by  $20m$  which is translated to states constraints on both  $x$ -axis and  $y$ -axis from  $-10m$  to  $+10m$  and is applied to each player. In each game, the relative distance between the two players are computed at every time instant using equation (2) and the game terminates when the condition (3) is satisfied. The radius of each robot,  $R_{rob} = 0.1m$ , thus the sum of the radii of the two robots (which is the threshold distance for terminating the game) is  $0.2m$ . To ensure that the game terminates in a finite time, the pursuer's speed limit is selected to be slightly higher than that of the evader, thus the limits of pursuer's control variables are  $|v_p| = 0.5m/s$ ,  $|\omega_p| = \pi/3$  while the limits of the evader's control variables are  $|v_e| = 0.48m/s$ ,  $|\omega_e| = \pi/3$ .

The weighting matrices are tuned and the best values obtained are  $R = [1; 0.5]^T$ ,  $Q = \text{diag}(1; 1; 0.00001)$  and  $Q_N = 100000 * Q$ . The weight on the angular speed was less than the weight on the linear speed because agility of the robot is less important compared to the fastness. The weight of the position was more important than the weight on the angular difference. For stability, a large weight on the terminal cost was imposed. Three prediction horizons  $N = [5, 10, 20]$  are used for the sake of comparison while the sampling time was  $T_s = 0.1$ .

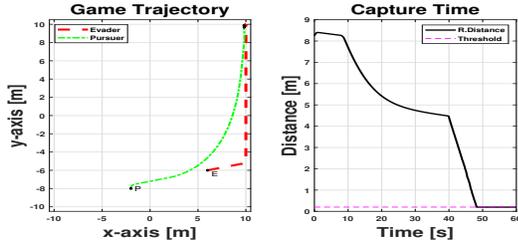
### A. Results of the Limited information MPC approach

Several simulations of PEG in an obstacles-free environment were conducted using random initial conditions. Each player firstly uses the opponent's current position to predict its optimal heading angle and then uses both the position and the predicted angle as a reference to the MPC. We present three different games with the same initial condition for the purpose of comparison, the only difference being the length of the prediction horizon. For each case, the initial condition of the pursuer is  $(-2, -8, \pi/2)$  while the initial condition of the evader is  $(6, -6, 0)$ . In Figure 1(a), the trajectory and capture time of the game was presented for the case where the prediction horizon is ( $N = 5$ ) while in Figure 1(b) and Figure 1(c), the same game was conducted with same initial conditions but using different prediction horizon of ( $N = 10$ ) and ( $N = 20$ ), respectively. Similar results are obtained nevertheless, the capture time is slightly affected by a small prediction horizon, as we see in Figure 1(a). The computation time of these three games are compared in Figure 2. The average computation time was by far less than the sampling time of  $T_s = 0.1s$  but one can notice a slight augmentation with the increase of the prediction horizons. Our proposed approach was found to be robust to the presence of measurement noise as shown in Figure 3. Some random noise was added to the feedback of each player, nevertheless close to similar results of Figure 1(a) were obtained.

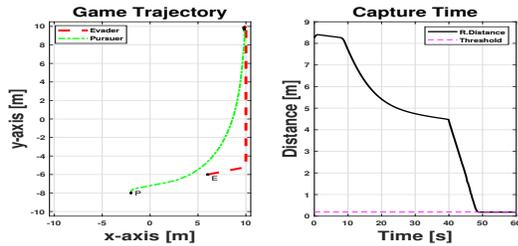
For the purpose of further improving the robustness of our



(a) PEG with Limited Information MPC,  $N = 5$



(b) PEG with Limited Information MPC,  $N = 10$



(c) PEG with Limited Information MPC,  $N = 20$

Fig. 1. PEG with Limited Information MPC in an obstacles free environment

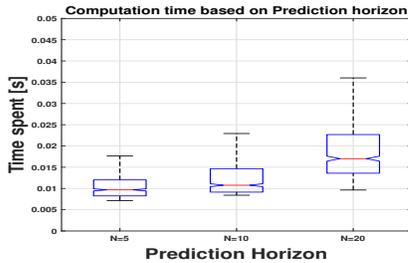


Fig. 2. Comparison of computation time, limited information MPC in an obstacle free environment

algorithms, we conducted three more simulations of the game in an environment that contains obstacles. The obstacles were placed strategically along the path of the game so that the players must encounter them. In Figures 4(a), 4(b) and 4(c), we present the trajectory and the capture time of the games for different prediction horizons. In each case, there is no collusion with any obstacle by either of the players. The computation time of each is computed for the comparison given in Figure 5. It was found that each computation time increases with increase in the length of the prediction horizon.

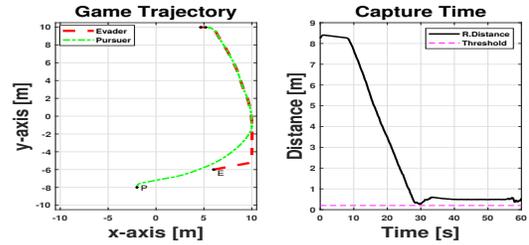
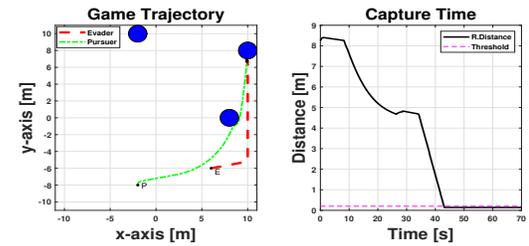
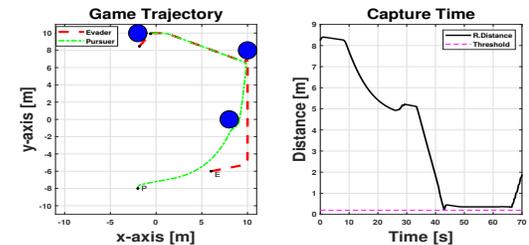


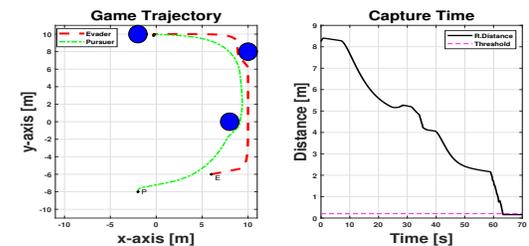
Fig. 3. PEG with Limited MPC with a noisy measurement,  $N=5$



(a) PEG with Limited Information MPC,  $N = 5$



(b) PEG with Limited Information MPC,  $N = 10$



(c) PEG with Limited Information MPC,  $N = 20$

Fig. 4. PEG with Limited Information MPC with obstacle avoidance

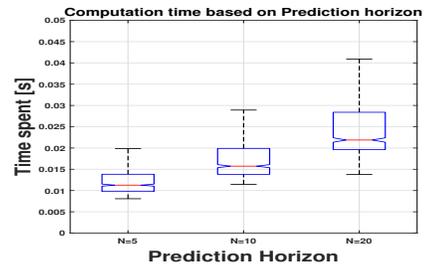


Fig. 5. Comparison of computation time, limited information MPC with obstacle avoidance

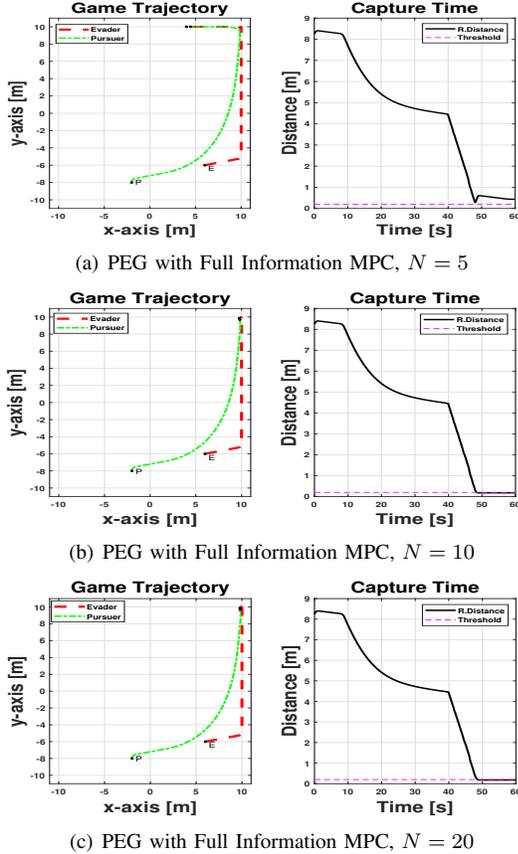


Fig. 6. PEG with Full Information MPC

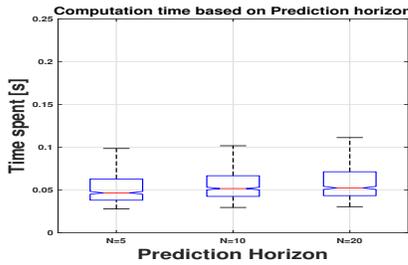


Fig. 7. Comparison of computation time, Full Information MPC in an obstacle free environment

## B. Comparative Study

We have compared the result of our method which have limited information with another approach in which each player has full information of the opponent. The results in Figure 6 depicts the trajectories and the capture time of the game under different prediction horizons. It can be seen that despite the limited information, our method performed satisfactorily in terms of the capture time. In addition, the computation of the method with full information, as presented in Figure 7 is much higher than the present method.

## V. CONCLUSION

This work proposes new game-theoretic way of using limited information MPC to handle pursuit-evasion games

for nonholonomic mobile robots where each player can only measure the current position of its opponent. Each uses its limited information to predict the future heading angle of its opponent. Instead of a conventional game-theoretic approach which requires each player to know the full information on its opponent which is unrealistic, the new strategy was able to handle the problem in the presence and absence of obstacles. The computation time required for this method is very small which is advantageous during physical implementation. It is also worth mentioning that the computation time is affected by the presence of obstacles. We would as a future work investigate the use of stochastic MPC in handling the same problem and then implement the algorithms on a physical systems. Finally, we would like to extend the work to cooperative pursuit problem where a team of pursuers cooperates to capture a single evader.

## REFERENCES

- [1] R. Isaacs, *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*, John Wiley and Sons Inc., New York originally published in (1965).
- [2] H. Huang, W. Zhang, J. Ding, D. M. Stipanovic, and C. J. Tomlin. "Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers" in the Proceedings of the Fiftieth IEEE Conference on Decision and Control and European Control Conference, pp. 4835–4840. IEEE, Orlando, FL (2011).
- [3] M. Kothari, J. G. Manathara and I. Postlethwaite "Cooperative Multiple Pursuers against a Single Evader" in Journal of Intelligent Robots and Systems 86, 551–567 (2017).
- [4] M. Kothari, J. G. Manathara and I. Postlethwaite "A cooperative pursuit-evasion game for non-holonomic systems" in the Proceedings of the Nineteenth IFAC World Congress, pp. 1977–1984, Cape Town, South Africa (2014).
- [5] Z. Zhou, W. Zhang, J. Ding, H. Huang, D. M. Sipanovic and C. J. Tomlin, "Cooperative Pursuit with Voronoi Partitions" in Automatica 72, 64-72, (2016).
- [6] I. E. Weintraub, M. Pachter and E. Garcia, "An Introduction to Pursuit-evasion Differential Games" in the proceedings of the American Control Conference (ACC), Denver, CO, USA, (2020), pp. 1049-1066.
- [7] E. Garcia, Z. E. Fuchs, D. Milutinovic, D. W. Casbeer, and M. Pachter. "A Geometric Approach for the Cooperative Two-Pursuer One-Evader Differential Game" in IFAC PapersOnline, 50-1(2017)-15214.
- [8] J. M. Eklund, J. Sprinkle and S. S. Shastry, "Switched and Symmetric Pursuit/Evasion Games Using Online Model Predictive Control With Application to Autonomous Aircraft" in IEEE Transactions on Control systems Technology, vol. 20, No. 3, May (2012).
- [9] G. Tzannetos, P. Marantos and K. J. Kyriakopoulos "A Competitive Differential Game Between an Unmanned Aerial and a Ground Vehicle Using Model Predictive Control" in 24th Mediterranean Conference on Control and Automation, Athens Greece, Jun 2016.
- [10] M. Sani, B. Robu and A. Hably "Pursuit-evasion Game for Nonholonomic Mobile Robots With Obstacle Avoidance Using NMPC" in the Proceedings of the 28th Mediterranean Conference on Control and Decision (MED), pp.978-983. Saint-Raphaël, France (2020),
- [11] L. Jualin. "Mobile Robotics", 6th Ed. Elsevier, (2015), pp 17-17.
- [12] J. R. Marden and J. S. Shemma, "Game theory and Control" in Annual Review of Control, Robotics and Autonomous Systems, (2018).
- [13] M. W. Mehrez, G. K. I. Mann and R. G. Gosine, "Stabilizing NMPC of wheeled mobile robots using open-source real-time software" in the 16th International Conference on Advanced Robotics (ICAR), Montevideo, (2013), pp. 1-6,
- [14] G. Diaz-Garcia , L. Burbano, N. Quijano and L. F. Giraldo "Distributed MPC and Potential Game Controller for Consensus in Multiple Differential-Drive Robots" in IEEE 4<sup>th</sup> Colombian Conference on Automatic control (CCAC) Medellin, Colombia (2019).
- [15] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings and M. Diehl. "CasADi - A software framework for nonlinear optimization and optimal control" in Mathematical Programming, in press (2018).