Pursuit-evasion Games Based on Game-theoretic and Model Predictive Control Algorithms

Mukhtar Sani, Bogdan Robu, Ahmad Hably

▶ To cite this version:

Mukhtar Sani, Bogdan Robu, Ahmad Hably. Pursuit-evasion Games Based on Game-theoretic and Model Predictive Control Algorithms. ICCAD 2021 : 5th IEEE-International Conference on Control, Automation and Diagnosis, Nov 2021, Grenoble, France. hal-03412852

HAL Id: hal-03412852

https://hal.science/hal-03412852

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.

Pursuit-evasion Games Based on Game-theoretic and Model Predictive Control Algorithms

Mukhtar Sani[1,2], Bogdan Robu[1], Ahmad Hably[1]

[1] Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-Lab, France
{mukhtar.sani,bogdan.robu,ahmad.hably}@gipsa-lab.fr

[2] Kano University of Science and Technology, Wudil, Nigeria
mukhtarsani@kustwudil.edu.ng

***Abstract*—Non-cooperative game problems such as pursuit-evasion require a solution approach that takes into consideration the strategy of the opponents. To predict the strategy of an opponent in a game, its full information is required and more computation time would be spent. However, this requirement of the opponent's full information is not realistic. Also, the computation time required by the game-theoretic algorithm (GTA) could make the controller unimplementable for some systems. Conversely, Model Predictive Control (MPC) could be use to solve the same problem using only the states information on the opponent by solving minimization or maximization cost functions. In this paper, we compared the GTA and MPC algorithm using two autonomous nonholonomic ground robots. Several simulations were conducted in the absence and presence of obstacles, using different initial conditions. The results obtained showed that the MPC algorithm can achieve similar performance.**

***Index Terms*—Game-theory, MPC, Obstacles avoidance, PEG, Mobile Robotics.**

I. Introduction

Pursuit-evasion game is the most popular example of non-cooperative games which has several applications in the fields of mobile robotics such as surveillance, navigation, conflict and combat operations, and so on. In a two player pursuit-evasion game, the pursuer is trying to catch the evader in minimum time while the evader tries to avoid the pursuer or maximize the time of capture. Nash equilibrium is a best strategy pair for each player such that deviating from the equilibrium point would not enhance the player's chance of winning. Therefore each player strives to find its Nash equilibrium. Formal solution of pursuit-evasion game was introduced in [1] along with its descriptions and applications. However, the solution concepts was argued in [2] that it resembles the dynamic programming approach due to the inability of the authors to make excessive use of classical variational techniques. A solution based on variational techniques was then proposed as a solution of differential games without treating any specific example.

Solutions of pursuit-evasion games with more than one player in a team have been reviewed in [3]. In [4] and [5] safe reachable area minimization was presented to capture a single evader using a group of pursuers. Voronoi partition was employed to determine the safe reachable area of the evader and then propose cooperative pursuit strategy. The drawback of this method is the computational difficulties due to area computation especially for nonholonomic systems. Geometric solution was proposed in [6] using Apollonian circle for holonomic robots only. The authors in [7] developed a hybrid algorithm of dealing with pursuit-evasion problem depending on presence of an obstacles in the region of each player using the concept of potential field. In [8], several proportional navigation guidance schemes were described, analysed and applied on physical wheeled robots. Same approach was also compared with safe reachable area minimization method in [4].

The paper [9] employed Non-linear Model Predictive Tracking Control (NMPTC) to solve a symmetric PEG between two fixed wing air-crafts in 3-D space. The game was symmetric in the sense that the game can be switched such that pursuer can become an evader and vice-versa. Each aircraft predicts both its optimal trajectory and that of the opponent assuming to have full information of the game. In [10], the authors approached PEG between two heterogeneous players, the unmanned ground vehicle (UGV) and unmanned aerial vehicle (UAV) using NMPC. The states information of the players were not considered, rather it considered the states of the game which translate as the error between the positions of the two players. Both players are assumed to have full information about the opponent dynamics. However, states constraints cannot be imposed and obstacles avoidance cannot be incorporated. In our earlier paper [11], we used NMPC based game-theoretic algorithms to solve PEG in the presence of obstacles. Game-theoretic algorithms were developed for both players, thus knowledge of full information is assumed. The game was set-up based on the agility and speed difference of the robots. Results obtained showed that an agile robot has more advantage than a faster robot. The main drawback of game-theoretic algorithms is the computation time due to double optimization which is very important for physical implementation of algorithms. Also, the assumption of knowing full information of the opponent to find the Nash equilibrium of the game is not realistic. Therefore it is important to find out if it is possible to achieve the same performance with states based MPC alone. We employed the structure of game of kind

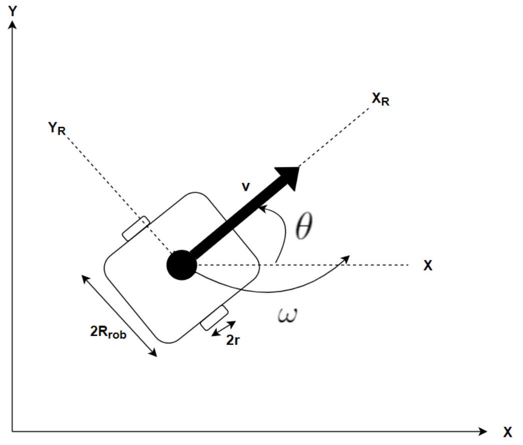


Fig. 1. Unicycle robot

where the capture time is the payoff of the game and assumed that the pursuer have higher speed and manoeuvrability for the purpose of terminating the game. We compare a game-theoretic algorithm with the MPC algorithm in simulation under several initial conditions with and without obstacles and compute the capture time in each game. For this purpose we present the robot's model and defined the problem statement in II. The game-theoretic and the MPC algorithms are presented in section III. Simulation results are shown in section IV. Finally, we present some conclusions and future work based on the obtained results in Section V.

II. Problem Statement

Consider a pursuit-evasion game between two non-holonomic robots of unicycle type in Figure 1. The game is to be 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 maximize the capture time. Assuming that the pursuer 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 [12] is given in (1).

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

where 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 θ_i are the position and orientation of i^{th} player respectively, while $(v_i, \omega_i) \in U_i$ represents the linear and the angular speeds respectively, which acts as a control inputs for the i^{th} player. U_i denotes the set of feasible control inputs for the i^{th} player and is assumed to be constrained.

To define the capture condition, let's denote t_c as the capture time, $r_{r.d}$ as the relative distance between the pursuer and the evader, r_c as capture radius (which is equal to the sum of the radii of the two robots) and $p(t)$ and $e(t)$ as the state vectors of the pursuer and the evader at time t respectively. The capture condition for the pursuer is given by

$$r_{r.d}(t_c) \leq r_c, \quad t_c \geq 0 \tag{2}$$

where

$$r_{r.d}(t) = ||p(t) - e(t)|| \tag{3}$$

Therefore, the pursuit problem can be defined as:

a) Problem 1: Given an initial configuration $p(0),\ e(0) \in \Omega$, with $r_{r.d}(0) > r_c$, find a set of feasible pursuer inputs $u_p \in U_p$ such that the capture time is minimised, assuming that the evader is playing its best strategy.

Conversely, the evasion problem can be defined as:

b) Problem 2: Given an initial configuration $p(0),\ e(0) \in \Omega$, with $r_{r.d}(0) > r_c$, find a set of feasible evader inputs $u_e \in U_e$ such that the capture time is maximized, assuming that the pursuer is playing its best strategy.

III. Control Algorithms

A. Model Predictive Control

Model Predictive Control is an online optimization technique which allow controller to compute new control variables at each decision instant. The main motivation for employing MPC technique was its ability to handle Nonlinear dynamics, Multi-Inputs Multi-outputs (MIMO) systems, states and inputs constraints as well as incorporating obstacles avoidance [11], [13]. In MPC, a controller is obtained by minimizing a cost function subject to constraints which comprises the dynamic model of the system. The controller output is a sequence of open loop controls predicted ahead over a finite horizon window called prediction horizon, N. Only first part of the controller is applied to close the loop at the particular decision instant T_s and the rest of the solution is ignored. At every decision instant, the prediction horizon is shifted one step and the process is repeated to obtain the new optimal control sequence.

Mathematically, the general form of the optimal control problem to solve is:

$$\min_u J = ||x(N) - r(N)||^2_{Q_N} + \sum_{k=0}^{N-1} ||x(k) - r(k)||^2_Q + ||u(k)||^2_R \tag{4}$$

subject to the following constraints:

$$x(k+1) = f(x(k), u(k)), \quad k = 0, 1, \ldots, N-1 \tag{5a}$$

$$x_{min} \leq x \leq x_{max} \tag{5b}$$

$$u_{min} \leq u \leq u_{max} \tag{5c}$$

The first part of the cost function is the terminal cost for stability purposes, while the second part is the running cost. x, r, and u stands for the states, reference and input parameters respectively. N is the prediction horizon, k stands for the instantaneous discrete time while Q, Q_N and R are the weighing matrices. The constraint in (5a) stands for the system dynamics which can be linear of non-linear, while the constraints in (5b) and (5c) represents states and inputs bounds respectively. At every decision instant, the systems is re-initialised using the new states measurements.

$$x(0) = x_0 \tag{6}$$

So that the output is a sequence of control inputs

$$u = [u(k), u(k+1), ...u(N-1)]^T \tag{7}$$

The first part $u = u(k)$ is applied to the system, while the others are discarded.
The controller parameters N, Q, Q_N and R can be tuned to stabilize the system depending on the relative importance of the states and on the computation time.
We can use this algorithm in our PEG in which each player only required states information of its rival to compute its strategy. The opponent's states is used as a reference $r(k)$. The nonlinear models in (1) is employed and used as the model of both systems. Thus, our control algorithm is Nonlinear Model Predictive Control (NMPC).

1) Pursuer's MPC Algorithm: The solution of problem 1 is the pursuer's controller which is a single NMPC block that accept states information of both players as an input and then give out the optimal strategy as an output by solving the following cost function:

$$\min_{U_p} J = ||X_p(N) - X_e||^2_{Q_N} + \sum_{k=0}^{N-1} ||X_p(k) - X_e||^2_Q + ||U_p(k)||^2_R \tag{8}$$

subject to:

$$\begin{cases} X_p(k+1) = f_p(X_p(k), U_p(k)), \quad k = 0, 1, ..N-1 \\ \sqrt{(x_e^{k+1} - x_{obs})^2 + (y_e^{k+1} - y_{obs})^2} \geq (R_{obs} + R_{rob}) \\ X_{p_{min}} \leq X_p(k+1) \leq X_{p_{max}} \\ U_{p_{min}} \leq U_p(k+1) \leq U_{p_{max}} \end{cases} \tag{9}$$

2) Evader's MPC Algorithm: The solution of problem 2 is the evader's controller which is also a single MPC block that accepts the states information of both players and give out the optimal strategy by solving the following the cost function:

$$\max_{U_e} J = ||X_e(N) - X_p||^2_{Q_N} + \sum_{k=0}^{N-1} ||X_e(k) - X_p||^2_Q + ||U_e(k)||^2_R \tag{10}$$

subject to:

$$\begin{cases} X_e(k+1) = f_e(X_e(k), U_e(k)), \quad k = 0, 1, ..N-1 \\ \sqrt{(x_e^{k+1} - x_{obs})^2 + (y_e^{k+1} - y_{obs})^2} \geq (R_{obs} + R_{rob}) \\ X_{e_{min}} \leq X_e(k+1) \leq X_{e_{max}} \\ U_{e_{min}} \leq U_e(k+1) \leq U_{e_{max}} \end{cases} \tag{11}$$

In order to include obstacles avoidance in the control algorithm, an additional constraint in (12) was imposed.

$$\sqrt{(x_i - x_{obs})^2 + (y_i - y_{obs})^2} \geq (r_{rob} + r_{obs}) \tag{12}$$

where x_i, y_i refer to the position of i^{th} player, x_{obs}, y_{obs} stand for the coordinate position of the obstacle and r_{obs} denotes the radius of the obstacle.

B. Game-theoretic Formulation

In game theory, systems are modeled as intelligent rational decision makers where an agent considers the opponent's move before deciding own strategy. The agent predict the opponent's best response which is the worst case from the agent's point of view and then computes its optimal strategy according to that. Game-theoretic solution is called Nash equilibrium which no player has an incentive to deviate from. In our pursuit-evasion problem, the sum of the two objective function can be expressed as:

$$J_p(U_p, U_e) + J_e(U_p, U_e) = 0 \quad \forall U_p \in \mathcal{U}_p, \; U_e \in \mathcal{U}_e \tag{13}$$

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. Given this special structure, zero-sum games are usually expressed in terms of a single objective function ϕ [14].

$$\phi(U_p, U_e) = J_p(U_p, U_e) = -J_e(U_p, U_e) \tag{14}$$

The solution of our differential game is double optimization:

$$\phi(U^{p^*}, U^{e^*}) = \min_{U^p} \max_{U^e} \phi(U^p, U^e) = \max_{U^e} \min_{U^p} \phi(U^p, U^e) \tag{15}$$

The strategy pair [U_p^*,U_e^*] is a Nash equilibrium if:

$$\phi(U_p^*, U_e^{'}) \leq \phi(U_p^*, U_e^*) \leq \phi(U_p^{'}, U_e^*) \tag{16}$$

1) Pursuer's Game-theoretic Control: In this algorithm, the control of each systems must firstly predict the strategy and the next move of the opponent so as to reduce the capture time. For this intelligent thinking, an additional MPC block is used based on assumption that the player knows full information about the opponent. Instead of sending the current position of the opponent to the control block, the predicted position is used. The computation times for each MPC block is summed at each instant. The solution of problem 1 is therefore in two steps:
Step 1: Predict evader's next move by solving the following cost function:

$$\max_{U_e} J = ||X_e(N) - X_p||^2_{Q_N} + \sum_{k=0}^{N-1} ||X_e(k) - X_p||^2_Q + ||U_e(k)||^2_R \tag{17}$$

subject to:

$$\begin{cases} X_e(k+1) = f_e(X_e(k), U_e(k)), \quad k = 0, 1, ..N-1 \\ \sqrt{(x_e^{k+1} - x_{obs})^2 + (y_e^{k+1} - y_{obs})^2} \geq (R_{obs} + R_{rob}) \\ X_{e_{min}} \leq X_e(k+1) \leq X_{e_{max}} \\ U_{e_{min}} \leq U_e(k+1) \leq U_{e_{max}} \end{cases} \tag{18}$$

The output of this block is a vector of predicted states of the evader denoted by X_e^*.

Step 2: Compute the pursuer's strategy by the solving the following cost function:

$$\min_{U_p} J = ||X_p(N) - X_e^*||^2_{Q_N} + \sum_{k=0}^{N-1} ||X_p(k) - X_e^*||^2_Q + ||U_p(k)||^2_R \tag{19}$$

subject to:

$$\begin{cases} X_p(k+1) = f_p(X_p(k), U_p(k)), \;\; k = 0, 1, ..N-1 \\ \sqrt{(x_e^{k+1} - x_{obs})^2 + (y_e^{k+1} - y_{obs})^2} \geq (R_{obs} + R_{rob}) \\ X_{p_{min}} \leq X_p(k+1) \leq X_{p_{max}} \\ U_{p_{min}} \leq U_p(k+1) \leq U_{p_{max}} \end{cases} \tag{20}$$

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

2) Evader's Game-theoretic Control: The solution of problem 2 is direct opposite to the above. In this case the evader first predict the optimal strategy and next move of the pursuer by solving minimization problem and then use the first block outputs to compute its optimal strategy by solving maximization cost function with its dynamics. Similarly, the solution is in two steps:
Step 1: Predict pursuer's next move by solving the following cost function:

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

subject to:

$$\begin{cases} X_p(k+1) = f_p(X_p(k), U_p(k)), \;\; k = 0, 1, ..N-1 \\ \sqrt{(x_e^{k+1} - x_{obs})^2 + (y_e^{k+1} - y_{obs})^2} \geq (R_{obs} + R_{rob}) \\ X_{p_{min}} \leq X_p(k+1) \leq X_{p_{max}} \\ U_{p_{min}} \leq U_p(k+1) \leq U_{p_{max}} \end{cases} \tag{22}$$

The output of this block is a vector of predicted states of the pursuer denoted by X_p^*.

Step 2: Then compute the evader's strategy by the solving the following cost function:

$$\max_{U_e} J = ||X_e(N) - X_p^*||^2_{Q_N} + \sum_{k=0}^{N-1} ||X_e(k) - X_p^*||^2_Q + ||U_e(k)||^2_R \tag{23}$$

subject to:

$$\begin{cases} X_e(k+1) = f_e(X_e(k), U_e(k)), \;\; k = 0, 1, ..N-1 \\ \sqrt{(x_e^{k+1} - x_{obs})^2 + (y_e^{k+1} - y_{obs})^2} \geq (R_{obs} + R_{rob}) \\ X_{e_{min}} \leq X_e(k+1) \leq X_{e_{max}} \\ U_{e_{min}} \leq U_e(k+1) \leq U_{e_{max}} \end{cases} \tag{24}$$

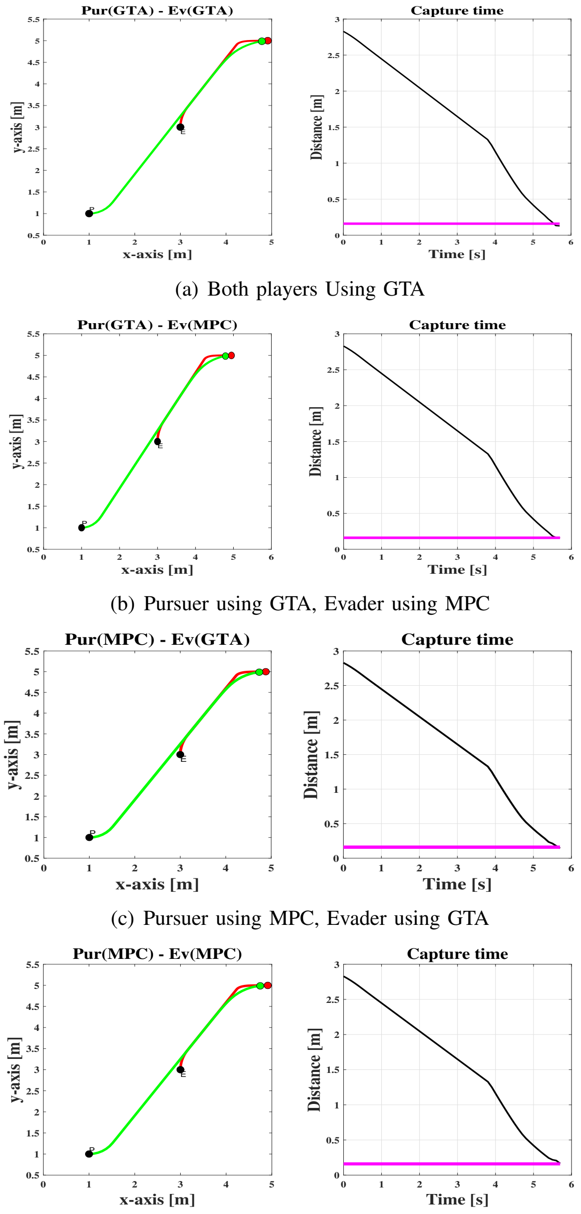
The output of this block is the evader's Nash equilibrium strategy, i.e $U_e^* = [v; \omega]^T$ which stands for the linear and angular speeds respectively.

IV. Results

The game was set up and implemented in MATLAB/Simulink environment. The controllers were coded using CasADi software [15], an open source symbolic framework for automatic differentiation and optimal control. An Interior point optimizer (IPOPT) was interfaced provide a solution. The game is played in the presence and absence of an obstacle. In each scenario (either with or without obstacle), four games are played such that in the first game, both players use the game-theoretic algorithms for control. In the second game, the pursuer uses game-theoretic algorithm while the evader use MPC algorithm. In the third game, the pursuer uses MPC algorithm while the evader uses game-theoretic algorithm. Finally, in the fourth game both players employ MPC algorithm. In each game, the relative distance between the two players are computed at every time instant as in (3), while the capture time is when (2) is satisfied.
The radius of each robot is $R_{rob} = 8cm$, the constraints on the pursuer's control variable are $|v_p| = 1m/s$, $|\omega_p| = \pi/3$ while the constraints on the evader's control variables are $|v_e| = 0.6m/s$, $|\omega_e| = \pi/4$. The pursuer has higher speed and manoeuvrability for the purpose of terminating the game in a finite time.
The weighting matrices were tuned and the best values obtained are $R = [1; 0.5]^T$, $Q = diag(1; 1; 0.001])$ and $Q_N = 100000 * Q$. The prediction horizon $N = 4$ while the sampling time $Ts = 0.1$. The states constraints is the dimension of the polytope $10m$ by $10m$, therefore the constraints on both $x - axis$ and $y - axis$ from $-5m$ to $+5m$ and is applied to both players.

A. Simulation in Obstacle-free environment

In Figure 2, PEG was played in the absence of obstacles but in a constrained environment. The initial conditions of the pursuer is $(1, 1, 0)$ while the initial condition of the evader is $(3, 3, \pi/2)$. These initial conditions are used for all the games in this scenario. In all the games, the left figure is the trajectory of the game while the figure at the right is the game payoff. The capture time in all the cases is obtained at the point of intersection of the relative distance $r_{r.d}$ between the two players and the constant threshold distance (sum of the radii of the two robots) $r_{th} = 0.16$. Both players used game-theoretic algorithms in 2(a), the capture time is 5.5 secs. Despite that the pursuer deviates from Nash equilibrium by using MPC algorithm in 2(b), the capture time is 5.5secs still. In Figure 2(c), the evader deviated from the Nash equilibrium and used MPC algorithm but still the capture time is 5.5 secs. Also in the figure 2(d), both payers deviated from the Nash equilibrium by employing MPC algorithm and still the capture time is 5.5 secs. Several simulation were conducted using many initial conditions with the pursuer behind the evader in each case. In all the simulations conducted, the capture time used to be approximately the same for all the four games.



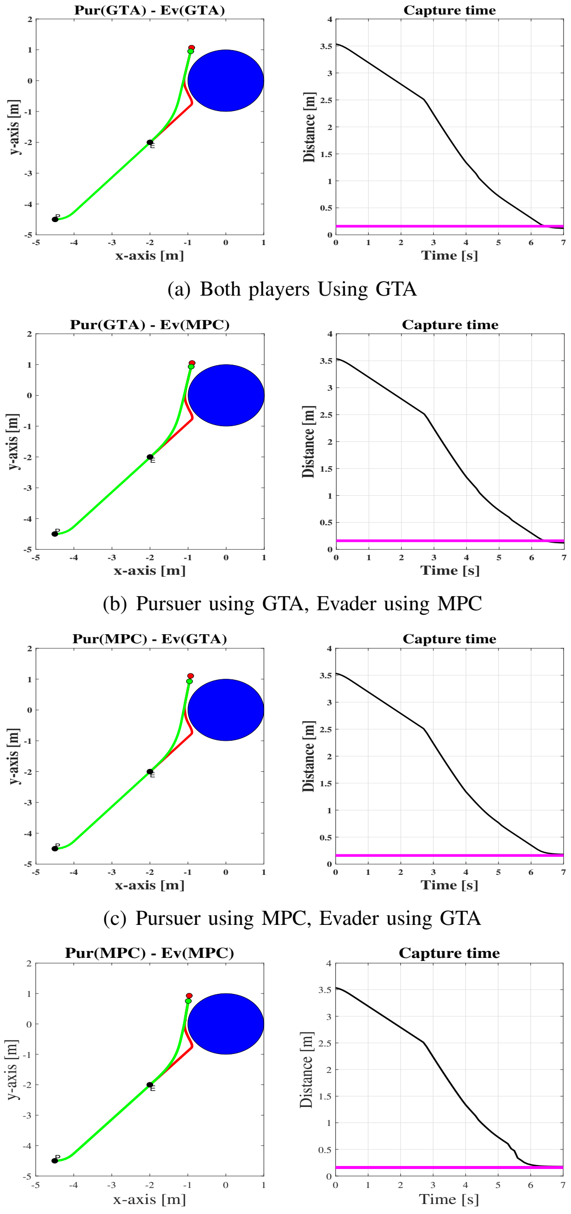
(a) Both players Using GTA

(b) Pursuer using GTA, Evader using MPC

(c) Pursuer using MPC, Evader using GTA

(d) Both players Using MPC

Fig. 2. PEG with Game-theoretic algorithm and MPC in an obstacle free environment



(a) Both players Using GTA

(b) Pursuer using GTA, Evader using MPC

(c) Pursuer using MPC, Evader using GTA

(d) Both players Using MPC

Fig. 3. PEG with Game-theoretic algorithm and MPC in the presence of an obstacles.

B. Simulation in an environment with obstacles

In order to improve the game condition to a more realistic situation, an obstacle avoidance capability was Incorporated. A spherical obstacle with 2m diameter is placed at the center of the game field (0,0). The players are placed in a strategic position so that the evader must encounter the obstacle before the end of the game. The results is presented in Figure 3. The initial conditions of the pursuer is $(-4.5, -4.5, 0)$ while the initial condition of the evader is $(-2, -2, \pi/4)$. These initial conditions are used for all the games in this scenario. In all the games, the left figure is the trajectory of the game while the figure at the right is the game pay-off. The capture time in all the cases is obtained at the point of intersection of the relative distance $r_{r.d}$ between the two players and the constant threshold distance (sum of the radii of the two robots) $r_{th} = 0.16$. Both players used game-theoretic algorithms in 3(a), the capture time is 6.3 secs. Despite that the pursuer deviate from Nash equilibrium by using MPC algorithm in 3(b), the capture time is 6.3secs still. In Figure 3(c), the evader deviated from the Nash equilibrium and used MPC algorithm but still the capture time is 6.3 secs. Also in the figure 3(d), both payers deviated from the Nash equilibrium by employing MPC algorithm and still the capture time is 6.3 secs.

V. Conclusion

This work studied pursuit-evasion games for nonholonomic mobile robots using game-theoretic and MPC algorithms. The game-theoretic algorithm required that each player must know the full information on its opponent to compute its Nash equilibrium which is difficult and consumes a lot of time. The MPC algorithm on the other hand, required only the current states information of the opponent. To show the performance of the algorithms, the game was played in the presence and absence of obstacles. In each scenario, four games are played with the players alternating their strategies between the game-theoretic and MPC algorithms. The game terminates when the relative distance between the two players reaches a threshold value, and the capture time is obtained at that point in each game. Based on simulations, the results obtained shows that MPC algorithm could achieve same performance.

As future work, we are presently working on implementing the controllers on physical systems to validate the results obtained in this work. For this purpose, the speed limits and some few modifications are inevitable. Also, we would like to extend the work to cooperative pursuit problem where a team of pursuers cooperate 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] Y. C. Ho, A. E. Bryson Jr, and S. Baron. “Differential Games and Optimal Pursuit-Evasion Strategies” in IEEE Transactions on Automatic Control Vol Ac-10, No.4 Oct 1965.

[3] I. E. Weintraub, M. Pachter and E. Garcia. “An Introduction to Pursuit-evasion Differential Games,” 2020 American Control Conference, Denver, CO, USA, (2020), pp. 1049-1066, doi: 10.23919/ACC45564.2020.9147205.

[4] M. Kothari, J. G. Manathara, and I. Postlethwaite. “Cooperative Multiple Pursuers against a Single Evader” in the Jounal of Intelligent Robots and Systems 86, 551–567 (2017). https://doi.org/10.1007/s10846-016-0423-3

[5] Z. Zhou, W. Zhang, J. Ding, H. Huang, D. M. Sipanovic and C. J. Tomlin. “Cooperative Pursuit with Vonoroi Partitions” in Automatica 72, 64-72, (2016).

[6] 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.

[7] J. Dong, X. Zhang and X. Jia. “Strategies of Pursuit-Evasion Game Based on Improved Potential Field and Differential Game Theory for Mobile Robots” in 2nd International conference on Instrumentation and Measurement, Computer, Communication and Control, Harbin, PR China (2012).

[8] A. Kumar and A. Ojha. “Experimental Evaluation of Certain Pursuit and Evasion Schemes for Wheeled Mobile Robots”. in the International Journal of Automation and Computation **16**, 491–510 (2019). https://doi.org/10.1007/s11633-018-1151-x

[9] 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).

[10] 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).

[11] 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).

[12] L. Jualin. “Mobile Robobtics”, 6th Ed. Elsevier, (2015). pp 17-17.

[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, doi: 10.1109/ICAR.2013.6766536.

[14] J. R. Marden and J. S. Shemma. “Game theory and Control” in Annual Review of Control, Robotics and Autonomous Systems, (2018).

[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 Computation, in press (2018).