Pursuit-evasion Game for Nonholonomic Mobile Robots With Obstacle Avoidance using NMPC

Mukhtar Sani, Bogdan Robu, Ahmad Hably

Pursuit-evasion Game for Nonholonomic Mobile Robots With Obstacle Avoidance using NMPC

Mukhtar Sani, Bogdan Robu, Ahmad Hably

***Abstract*— In this work, non-cooperative competitive games between two unmanned ground robots using Nonlinear Model Predictive Control (NMPC) while incorporating obstacle avoidance techniques are studied. The objective of the first player (pursuer) is to minimize the relative distance and orientation between itself and the second player (evader) while avoiding obstacles, whereas the evader does the opposite. The Pursuit-Evasion Game (PEG) being a typical class of a differential game is formulated as a zero-sum game with two homogeneous players in five different game scenarios. The objective function of each player is formulated as a double optimization problem and is solved separately using NMPC techniques. The optimal trajectory of each player is computed iteratively by considering the best response of the opponent player. The level of information is assumed to be symmetric. Simulations of various scenarios show the winning possibility of each player.**

I. INTRODUCTION

Nowadays, game-theoretic paradigm has been getting a fascinating attention in the field of robotics and control despite its high computational complexities. This is due to its possibility to interpret a control law as a sort of intelligent rational decision maker that was designed to produce a desired effect [1]. Pursuit-evasion is an interesting class of differential games which is a typical non-cooperative game problem. It includes a number of mobile pursuers and evaders in a conflicting scenarios. Example of such problem involve a pursuer trying to catch the evader while the evader is trying to dodge in a real environment; this can be achieved by independently minimizing and maximizing the relative distance and orientation between them.

The application of the this setting was argued in [2] to be numerous. In surveillance and security, for tracking malicious evader that is trying to escape, the security robot must ensure that the evader didn't escape by maintaining visibility. Also, in home care setting, a tracking robot can follow an elderly person and alert caregivers in case of emergencies. It can also be useful in wildlife monitoring, where unmanned surface vehicle (USV) is required to navigate in a cluttered environment while tracking marine species.

Several approaches for solving differential games and in particular pursuit-evasion games have been presented in the past while earliest approaches are optimal control based techniques. In [3] variational techniques was used to solve differential games, conditions for capture and optimality are derived for a class of pursuit-evasion problem. Reachability-based approached are presented in [4] using Hamilton-Jacobi-Isaac's formulation to compute the reachable set of a continuous dynamic differential game. However, [5] argued that solutions of HJI equations are not readily available from the practical point of view especially for the problems with multiple agents and non-trivial dynamics, thus proposed conditions for the game to be terminated in terms of reachable sets inclusion. Numerical solutions to differential games by a sequence of finite state Markov games has been presented in [6] where players are assumed to be moving at constant speed. In [7], Improved Potential field method has been used for solving pursuit-evasion problem. A novel incremental sampling-based algorithm to compute the open-loop solutions for the evader assuming worst case scenario for the pursuer is presented in [8]. In [9] non-cooperative multi-agent planning problem was formulated as a stochastic game for a situation where there is an uncertainty from the sensor and actuator noise. Recently, a more intelligent approach was proposed in [10] where a tool from social psychology known as Social Value Orientation (SVO) was integrated into autonomous vehicle decision making. This was aimed to quantify the degree of agent's selfishness or altruism, thus allow for a better prediction of the agent's interaction and cooperation with others.

Pursuit-evasion often involves cooperation among agents. For example, a group of pursuers trying to capture a single evader as in [11] or the opposite case where a single pursuer against many evaders as in [12]. In other scenario where cooperation is involved in the pursuit-evasion game is Target, Defender and Attacker called (TAD) game. Two pursuit-evasion problems are coupled: Attacker-Target and Defender-Attacker as in [13]. The Attacker chases the Target whilst avoiding being captured by the defender and in parallel the Target cooperates with the Defender in order to dodge from the Attacker while helping the Defender to capture the Attacker. A practical application the TAD game is in active defense as proposed in [14] where an Attacker missile pursues a Target aircraft protected by a Defender missile which aims at intercepting the Attacker before the latter reaches the Target aircraft.

In most of the above situations Nash equilibrium of the game is computed using optimal control approaches, however, a much better technique is the so called "best response" where the Nash equilibrium is attained when all the agents are best responding. In [15] several information patterns are reviewed and then Best response is used to compute the Nash equilibrium for two quadrotor drone racing. The work was improved in [16] where sensitivity enhanced iterated best response algorithms was used to solve for the approximate Nash equilibrium in the space of feasible trajectories and applied to a car-like vehicle. In [17], the concept of iterated best response

[1] The auathors are with the Univ. Grenoble Alpes, CNRS, Grenoble INP*, GIPSA-lab, 38000 Grenoble, France. {mukhtar.sani, bogdan.robu, ahmad.hably}@grenoble.fr

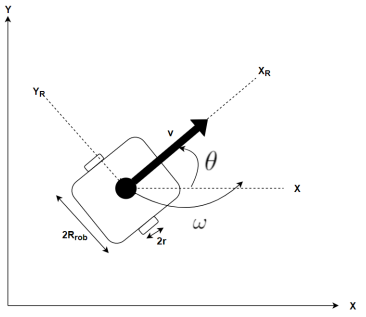

Fig. 1. Schematic diagram of mobile robot

and model predictive control was combined to solve for an agile interaction between two ground vehicles modeled in a semi-stochastic formulation.

Nonlinear Model Predictive Control approach has been applied to two aerial systems in [18]. Each aircraft computes its optimal strategy by firstly predicting the opponent's optimal trajectory assuming each have information about the dynamics and the current state of the itself and the opponent. The game can be switched due to its symmetrical nature, i.e the pursuer can become the evader and the vice-versa. Similar approach of using nonlinear Model predictive control (NMPC) have been presented in [19] for two heterogeneous systems. The unmanned ground vehicle (UGV) and the unmanned aerial vehicle (UAV). Two independent Model predictive controllers are designed using the dynamic model of UAV and the UGV. The PEG appear to be constant-sum instead of zero-sum game due to its heterogeneous nature. Nevertheless neither one of the two papers, [18] and [19], include obstacle avoidance which is a crucial aspect when dealing with the control of robots in a realistic environment.

In this paper, we intend to use NMPC techniques to compute optimal trajectories for two homogeneous autonomous ground vehicles in an cluttered environment and to analyse the various game scenarios in order to determine the win possibility for both players assuming equal level of information. The rest of the paper is organized as follows. In section II , we present the models of the non-holonomic robots, the obstacle avoidance techniques and formulation of the pursuit-evasion problem as a Nash equilibrium search. In section III, design of the game theoretic controllers is presented while the game scenarios and the results are presented in section IV. Finally conclusion and suggestions of future work are presented in section V.

II. Problem statement

A. Non-holonomic Mobile Robots

A robot is said to be holonomic if all its dynamics constraints are integrable into positional constraints. Another class of robots are called non-holonomic robots whose dynamic constraints are non-integrable. Such systems are also referred to as differential drive robots. This type of robots can be described as a cart having two controlled wheels with a control-free wheel at its front or as a tricycle with three controlled wheels. It can also be represented with a four-wheels configuration. The kinematic model of cart robots is given as [20]:

$$\begin{cases} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{cases} \tag{1}$$

Where the state vector denoted as $X = (x, y, \theta)$ represents the Cartesian (x, y) position of the robot in the earth frame of reference and the orientation angle θ, while v, ω are respectively the linear and the angular speeds of the robot measured in m/s and rad/s, contained in the input vector $U = (v, \omega)$. Moreover, the control inputs v and ω are related to the angular speeds of the left (ω_l) and right (ω_r) wheels of the robot by the following equation:

$$\begin{cases} \omega_r = (2v + \omega R_{rob})/2r \\ \omega_l = (2v - \omega R_{rob})/2r \end{cases} \tag{2}$$

where R_{rob} is the length of the robot's base from the center and r is the radius of the robot's wheels.

The discrete version (Euler) of the kinematic model in eq. (1) is used for designing the discrete NMPC for both robots.

$$\begin{cases} x^{k+1} &= x^k + T_s(v^k\cos\theta^k) \\ y^{k+1} &= y^k + T_s(v^k\sin\theta^k) \\ \theta^{k+1} &= \theta^k + T_s(\omega^k) \end{cases} \tag{3}$$

For the purpose of notation, lets represent the discrete kinematic equations of the two robots as follows, where the symbol "p" denotes pursuer and the symbol "e" denotes evader, T_s is the sampling time.

$$\begin{cases} X_p^{k+1} = f_p(X_p^k, U_p^k, Ts) \\ X_e^{k+1} = f_e(X_e^k, U_e^k, Ts) \end{cases} \tag{4}$$

Where the state variables $X_p^k = [x_p^k, y_p^k, \theta_p^k]^T$ and $X_e^k = [x_e^k, y_e^k, \theta_e^k]^T$ are measured using sensors at every decision instant.

B. Obstacle avoidance

One of the key issues associated to autonomous navigation of mobile robots is obstacle avoidance [21]. Mobile robots operating in a dynamic environment had to be enhanced with obstacle avoidance techniques for the safety of the objects around and that of the robot.

Literally, obstacle avoidance can be achieved via two stages, obstacle detection and collision avoidance. Some of the widely accepted obstacle avoidance techniques are:

- Bug Algorithms: These are basically the simplest algorithms [21]. The robot moves on the shortest path from its current position towards the goal until it come across an obstacle. The algorithm forces the robot to move tangentially around the obstacle's surface until it returns to its original path. The primitive bug algorithms makes the robot to circumnavigate the whole obstacle before returning to its original path.
- Artificial Potential Field (APF) Methods: In APF methods, the robot, obstacle and the goal are considered as electric charges such that the robot and the obstacle have the same polarity so that repulsive force is created while

the goal is assumed to have opposite polarity with the robot so that attractive force will be created [22].

- Bubble Band Technique: In these methods as firstly proposed in [23], the robot is surrounded by a "bubble" containing the maximum available free space which the robot can pass in any direction without collision.
- Vector Field Histogram: This method was firstly proposed in [24] in order to deal with the issue of sensor noise through constructing a polar histogram using the most recent readings of the sensor. In the histogram, a probability of obstacle's presence in a particular direction is plotted against the angle associated with the sonar sensor readings. A local occupancy grid map of the environment around the robot is created in order to compute the probabilities.
  The polar histograms is used to determine all the passages large enough to avoid collision with the obstacle. The passage to be followed by the robot is selected by evaluating the cost function defined for each passage which is a function of the alignment of robot's path with the goal and the difference between the current wheel orientation and the new direction. The passage having minimum cost function is selected.

In this work, the bug type algorithm is selected due to its simplicity and the intention to avoid increasing the computational cost of the NMPC. We consider M obstacles which are assumed to be spherical in shape with radius R_{obs} and each being positioned in a point described by it's Cartesian coordinates (x_{obs},y_{obs}). Detection of the obstacles can be achieved by measuring R_{obs}, x_{obs} and y_{obs} using sensors. Collision with the obstacles is avoided by including the following function as an inequality constraint in the NMPC formulation.

$$\sqrt{(x_{rob}-x_{obs})^2+(y_{rob}-y_{obs})^2} \geq (R_{obs}+R_{rob}+d_s) \tag{5}$$

where x_{rob} and y_{rob} are the position of the robot (pursuer or evader) in x-y plane, d_s is a safe distance between the robot and an obstacle. In this work, three spherical obstacles each of 2m in diameter are placed on strategic positions. Each Robot have a diameter of 0.4m while the safe distance selected is 0.2m.

C. Game Theoretic Formulation

In game theory, systems are modelled 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. If each agent plays its best response, Nash equilibrium is attained in which no player has incentive to deviate.

In our pursuit-evasion problem, the two agents have exactly the opposite intention. This depicts the zero-sum property in which an increase in utility/cost for one player results in a decrease in utility/cost for the other player by the same amount. The sum of the two objective function can be expressed as:

$$J_p(U_p,U_e)+J_e(U_p,U_e)=0 \quad \forall U_p \in \mathcal{U}_p,\ U_e \in \mathcal{U}_e \tag{6}$$

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 $\mathcal{J}$, thus

$$\mathcal{J}(U_p,U_e)=J_p(U_p,U_e)=-J_e(U_p,U_e) \tag{7}$$

Therefore the objective function of the pursuer and the evader can be defined in terms of the single objective function using double optimization as:

$$\begin{cases} \overline{\mathcal{J}}=\max_{U_e}\min_{U_p}\mathcal{J}(U_p,U_e) \\ \underline{\mathcal{J}}=\min_{U_p}\max_{U_e}\mathcal{J}(U_p,U_e) \end{cases} \tag{8}$$

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

$$\mathcal{J}(U_p^*,U_e^{'}) \leq \mathcal{J}(U_p^*,U_e^*) \leq \mathcal{J}(U_p^{'},U_e^*)$$

which indicates that U_p^* is the best response for the U_e^* and vice versa, thus $\overline{\mathcal{J}}=\underline{\mathcal{J}}$.
The value of the game corresponds to the saddle points of the game is given by:

$$VG=\mathcal{J}(U_p^*,U_e^*)$$

III. Control Design

Nonlinear Model Predictive Control (NMPC) technique was employed to compute the control inputs for both robots. The main motivations for using NMPC in our case are threefold: 1) Being an online optimization technique so that new controls can be computed at each decision instant. 2) Ability to handle Nonlinear dynamics of mobile robots. 3) Ability to handle MIMO systems such as the mobile robots.

In NMPC, a controller is obtained by minimizing a cost function subject to constraints which incorporate the nonlinear 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.

In the context of game theory, each player must firstly compute the worst case strategy of the opponent, then compute its own strategy based on it; which means that our problem has double stage optimization. Literally, the pursuer would first use the evader's dynamics and compute the maximum strategy (which is the pursuer's worst case), then compute its minimum strategy as a function of the latter. The evader does exactly the opposite by firstly computing the minimum strategy of the pursuer and then computes its maximum.

A. Controller Design for the Pursuer

The pursuer's controller is obtained by solving its Min-max problem. This was achieved by firstly estimating the evader's best move U_e^*:

$$\max_{U_e} J=\sum_{k=1}^{N}(X_e^k-X_p^k)^T Q_e(X_e^k-X_p^k)+(U_e^k)^T R_e U_e^k \tag{9}$$

subject to:

$$\begin{cases} X_e^{k+1} = f_e(X_e^k, U_e^k), \;\; k = 0, 1, \ldots, N-1 \\ f(X_e, X_{obs}(i)) \geq (R_{obs} + R_{rob} + d_s), \;\; i = 1, \ldots, M \\ X_{e_{min}} \leq X_e \leq X_{e_{max}} \\ U_{e_{min}} \leq U_e \leq U_{e_{max}} \\ X_p(0) = X_{p_0} \\ X_e(0) = X_{e_0} \end{cases} \tag{10}$$

Then compute U_p^* based on the value of U_e^* by solving.

$$\min_{U_p} J = \sum_{k=1}^{N} [X_p^k - X_e^k(U_e^*)]^T Q_p [X_p^k - X_e^k(U_e^*)] + (U_p^k)^T R_p U_p^k \tag{11}$$

subject to:

$$\begin{cases} X_p^{k+1} = f_p(X_p^k, U_p^k), \;\; k = 0, 1, ..N-1 \\ f(X_p, X_{obs}(i)) \geq (R_{obs} + R_{rob} + d_s), \;\; i = 1, \ldots, M \\ X_{p_{min}} \leq X_p \leq X_{p_{max}} \\ U_{p_{min}} \leq U_p \leq U_{p_{max}} \\ X_p(0) = X_{p_0} \\ X_e(0) = X_{e_0} \end{cases} \tag{12}$$

Where N is the level of thinking (technically called prediction horizon), M is the total number of obstacles in the game scenario, (X_e, U_e) and (X_p, U_p) denote the dynamical states and inputs of the evader and pursuer respectively, X_{p_0} and X_{e_0} are initial conditions, X_{obs} represent the x-y positions of the obstacles. The weighting matrices Q and R are chosen by the designer in order to achieve the satisfactory controller performance.

B. Controller Design for the Evader

Similarly, the evader's controller is obtained by solving its max-min problem which was is done by firstly computing U_p^* through solving:

$$\min_{U_p} J = \sum_{k=1}^{N} (X_e^k - X_p^k)^T Q_p (X_e^k - X_p^k) + (U_p^k)^T R_p (U_p^k) \tag{13}$$

subject to:

$$\begin{cases} X_p^{k+1} = f_p(X_p^k, U_p^k), \;\; k = 0, 1, ..N-1 \\ f(X_p, X_{obs}(i)) \geq (R_{obs} + R_{rob} + d_s), \;\; i = 1, \ldots, M \\ X_{p_{min}} \leq X_p \leq X_{p_{max}} \\ U_{p_{min}} \leq U_p \leq U_{p_{max}} \\ X_p(0) = X_{p_0} \\ X_e(0) = X_{e_0} \end{cases} \tag{14}$$

Then compute U_e^* based on the value of U_p^* by solving.

$$\max_{Ue} J = \sum_{k=1}^{N} [X_e^k - X_p^k(U_p^*)]^T Q_e [X_e^k - X_p^k(U_p^*)] + (U_e^k)^T R_e (U_e^k) \tag{15}$$

subject to:

$$\begin{cases} X_e^{k+1} = f_e(X_e^k, U_e^k), \;\; k = 0, 1, ..N-1 \\ f(X_e, X_{obs}(i)) \geq (R_{obs} + R_{rob} + d_s), \;\; i = 1, \ldots, M \\ X_{e_{min}} \leq X_e \leq X_{e_{max}} \\ U_{e_{min}} \leq U_e \leq U_{e_{max}} \\ X_p(0) = X_{p_0} \\ X_e(0) = X_{e_0} \end{cases} \tag{16}$$

The Q_p and Q_e are identity matrices of order 3, the R_p and R_e are identity matrices of order 2, the prediction horizon is 5 while the sampling time Ts is 0.1sec.

N.B: The reference trajectory in eq. (13) and (15) are computed using the opponent's dynamics as a function of the computed optimal control values.

IV. Simulation

A. Game Setup

The proposed controllers are implemented in MATLAB/SIMULINK coded using CasADi [26] interfaced with IPOPT solver. All simulations were done on a HP laptop with Intel Core i7 vPro 2.60GHz processor, 16G RAM and running 64-b windows 7 operating system in order to achieve fast computation. The Game was set up in five scenarios based on the input constraints. The state constraints are chosen to be a square shaped space for an indoor experiment defined by the opposed corner coordinates $[x_{min}, y_{min}]$ as $[-5m, -5m]$ and $[y_{max}, x_{max}]$ as $[5m, 5m]$. The initial conditions are randomly selected but with the pursuer always behind the evader. Below, we propose five different simulation scenarios in order to test the extreme cases:

Scenario 1: Both the pursuer and the evader have the same speed and agility. The linear and angular speeds limits for both robots are $-2ms^{-1} \leq v \leq 2ms^{-1}$ and $-2rads^{-1} \leq \omega \leq 2rads^{-1}$ respectively.

Scenario 2: Both the pursuer and the evader have the same speed but the pursuer is more agile. The linear speed limits of both robots is $-2ms^{-1} \leq v \leq 2ms^{-1}$ while the angular speed limits for the pursuer and evader are $-2rads^{-1} \leq \omega \leq 2rads^{-1}$ and $-1rads^{-1} \leq \omega \leq 1rads^{-1}$ respectively.

Scenario 3: Both the pursuer and the evader have the same speed but the evader is more agile .The linear speed limits of both robots is $-2ms^{-1} \leq v \leq 2ms^{-1}$ while the angular speed limits for the pursuer and evader are $-1rads^{-1} \leq \omega \leq 1rads^{-1}$ and $-2rads^{-1} \leq \omega \leq 2rads^{-1}$ respectively.

Scenario 4: Both the pursuer and the evader have the same agility but the pursuer is faster. The angular speed limits of both robots is $-2rads^{-1} \leq \omega \leq 2rads^{-1}$ while the linear speed limits for the pursuer and evader are $-2ms^{-1} \leq v \leq 2ms^{-1}$ and $-1ms^{-1} \leq v \leq 1ms^{-1}$ respectively.

Scenario 5: Both the pursuer and the evader have the same agility but the evader is faster. The angular speed limits of both robots is $-2rads^{-1} \leq \omega \leq 2rads^{-1}$ while the linear speed limits for the pursuer and evader are $-1ms^{-1} \leq v \leq 1ms^{-1}$ and $-2ms^{-1} \leq v \leq 2ms^{-1}$ respectively.

In each scenario, the relative distance and orientation between the two systems, $R.D$ is computed at each sampling

instant.

$$R.D = \sqrt{(x_p - x_e)^2 + (y_p - y_e)^2 + (\theta_p - \theta_e)^2} \quad (17)$$

A threshold distance of 0.2m is selected to stop the simulation. The game time in seconds is obtained to indicate how long it takes the pursuer to catch the evader. Thus a pursuer is said to win if the game time is very small while evader wins if the game time is very long.

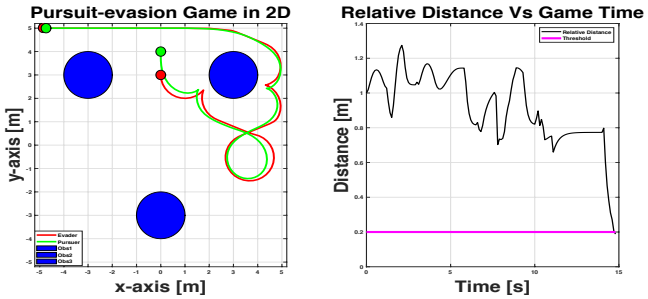

Fig. 2. Simulation of scenario 1, the blue circles represents the obstacles while the red and green lines represent the trajectory of the players.

B. Results

Several simulation experiments were conducted using several random initial conditions. The results presented here, for the purpose of consistency, are simulated with the same initial conditions for all the scenarios. The pursuer's initial conditions are $[x_p, y_p, \theta_p]^T = [0, 4, -\pi/2]^T$ while that of the evader are $[x_e, y_e, \theta_e]^T = [0, 3, -\pi/2]^T$.

For Scenario 1, where both robots have the same speed and agility, it takes the pursuer 14.8 seconds to catch the evader as depicted in figure 2. Both robots avoided the obstacles encountered and box constraints.

For scenario 2 and 3, the effect of agility is tested. The speed of both robots is the same. In scenario 2, the pursuer is more agile, thus catches the evader in just 5.4 seconds as in figure 3 while in scenario 3 the evader is more agile, thus it takes the pursuer 17.3 seconds to catch it as in figure 4. The agility differences has great effect in the game.

For scenario 4 and 5, the effect of speed is tested. The agility of both players is the same. In scenario 4, the pursuer is faster, it takes around 14 seconds to catch the evader as in figure 5

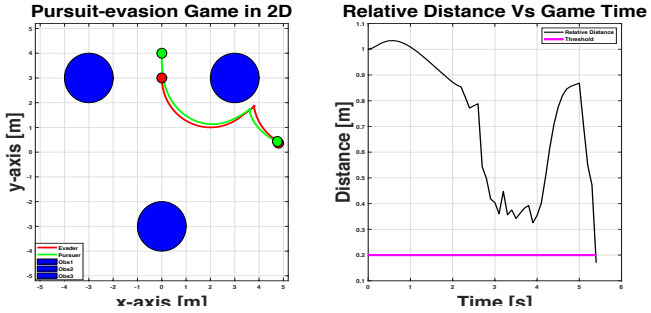

Fig. 3. Simulation of Scenario 2, the blue circles represent the obstacles,the red/green lines represent the trajectory of the players.

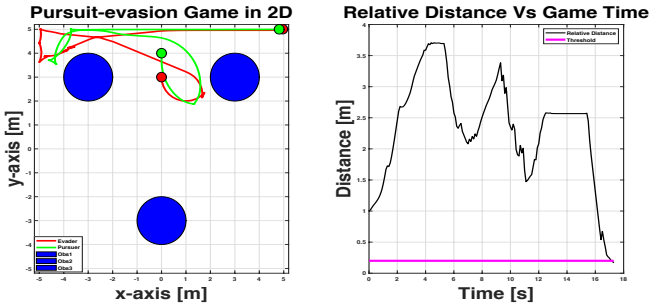

Fig. 4. Simulation of Scenario 3,the blue circles represent the obstacles,the red/green lines represent the trajectory of the players.

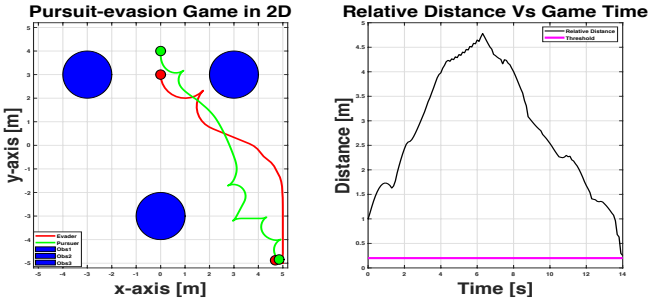

Fig. 5. Simulation of Scenario 4, the blue circles represent the obstacles,the red/green lines represent the trajectory of the players.

whereas in scenario 5 where the evader is faster, it takes 14.9 seconds to be captured by the pursuer. The speed has slight effect in the game.

In table I, the agility and speed effects are compared with the scenario 1. The percentage time gained are computed for each robot. It's observed that a more agile robot always has advantage.

The computation time for both controllers is in range of 0.01 to 0.02 seconds.

Player	More Agile	Faster
Pursuer	63.5%	5.4 %
Evader	16.9 %	0.7%

TABLE I

COMPARISON OF THE ADVANTAGE OF AGILITY AND SPEED.

V. CONCLUSION

In this work, we used game theory to design two Nonlinear Model Predictive Controllers for two homogeneous vehicles having opposing objectives in the presence of obstacles. Five scenarios have been examined, a robot that's more agile has better chance of winning the game in a cluttered environment that a faster one. Realistic game situations have been utilised and equal level of information has been assumed. Solution of min-max and max-min leads to Nash equilibrium in which any vehicle that deviate form it will eventually lose. Obstacles and constraints on the playground has been seen to slightly affect the winning. Thus in a tightly constraint environment, an agile player is likely to win despite the fact that the opponent is

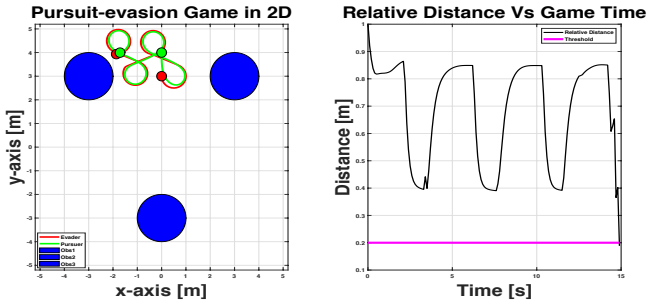

Fig. 6. Simulation of Scenario 5, the blue circles represent the obstacles,the red/green lines represent the trajectory of the players.

playing its Nash equilibrium.

The future work is to validate the simulation results using physical robots. Another trend is to develop a game of incomplete information, in which each robot has to estimate the dynamics of its opponent.

REFERENCES

[1] J. R. Marden and J. S. Shemma, " Game theory and Control" in Annual Review of Control, Robotics and Autonomous Systems, 2018.

[2] S. Bhattacharya, S. Hutchinson and T. Basar, "Game-theoretic Analysis of a Visibility Based Pursuit-Evasion Game in the Present of Obstacles", in American Control Conference, MO, USA, Jun, 2009.

[3] 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.

[4] I. M. Mitchell, A. M. Bayen and C. J. Tomlin, "A Time-Dependent Hamilton-Jacobi Formulation of Reachability Sets for Continuous Dynamic Games" in IEEE Transactions on Automatic Control, Vol 50, No. 7 Jul 2005.

[5] W. Sun, P. Tsiotras, T. Lolla, D. N. Subramani and P. F. J Lermusiaux, "Pursuit-Evasion Games in Dynamic Flow Fields via Reachability Set Analysis" In American Control Conference, Seattle, USA, May 2017.

[6] B. Tolwinski, "Numerical Solutions to Differential Games Based on Approximations by Markov Games", in the Proceedings of the 28th Conference on Decision and control, Florida 1989.

[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] S. Karaman, E. Frazzoli. "Incremental Sampling-Based Algorithms for a Class of Pursuit-Evasion Games" In: Hsu D., Isler V., Latombe JC., Lin M.C. (eds) Algorithmic Foundations of Robotics IX. Springer Tracts in Advanced Robotics, vol 68. Springer, Berlin, Heidelberg.

[9] W. Schwarting, A. Pierson, S. Karaman, and D. Rus, Stochastic Dynamic Games in Belief Space, arXiv preprint arXiv:1909.06963v1 [cs.RO] 16 Sep 2019.

[10] W. Schwarting, A. Pierson, S. Karaman, and D. Rus, Social behavior for autonomous vehicles, submitted, 2019.

[11] 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.

[12] Z. E. Fuchs. P. P. Khargonekar and J. Evers "Cooperative Defense within a Single-Pursuer, Two-Evader Pursuit Evasion Game" In 49th IEEE Conference on Decision and Control, GA, USA, DEC, 2010.

[13] L. Liang, Z. Peng, F. Zhang and X. Li, " Two Coupled Pursuit-Evasion Games in Target-Attacker-Defender Problem" In 56th IEEE Conference on Decision and Control, Melbourne Australia, Dec 2017.

[14] E. Garcia, D. W. Casbeer and M. Pachter, " Design and Analysis of State-Feedback Optimal Strategies for the Differential Game of Active Deffense" in IEEE Transaction on Automatic Control vol. 64, no 2, Feb 2012.

[15] R. Spica, D. Falanga, E. Cristofalo, E. Montijano, D. Scaramuzza and M. Schwager, "A real-time Game Theoretic Planner for Autonomous Two-Player Drone racing" in Proceedings of Robotics: Science and Systems, Jun 2018.

[16] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes and M. Schwager, Game Theoretic Planning for Self-Driving Cars in Competitive Scenarios. in Proceedings of Robotics: Science and Systems.2019.

[17] G. Williams, B. Goldfain, P. Drews, J. M. Rehg and E. Theodorou Best Response Model Predictive Control for agile Interactions Between Autonomous Ground Vehicles. in IEEE International Conference on Robotics and Automation (ICRA), Brisbane Australia, May 2018

[18] 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.

[19] 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.

[20] L. Jualin. "Mobile Robobtics", 6th Ed. Elsevier, 2015. pp 17-17.

[21] V. A. Bhavesh " Comparison of Various Obstacle Avoidance Algorithms" in International Journal of Engineering research and Technology (IJERT), Vol. 4 issue 12 Dec 2015.

[22] V. Lumelsky and T. Skewis, Incorporating Range Sensing in the Robot Navigation Function. IEEE Transactions on Systems, Man, and Cybernetics, 20:1990, pp. 10581068..

[23] Y. Koren and J. Borenstein. High-Speed Obstacle Avoidance for Mobile Robotics, in Proceedings of the IEEE Symposium on Intelligent Control, Arlington, VA, August 1988, pp: 382-384.

[24] O. Khatib and S. Quinlan. Elastic Bands: Connecting, Path Planning and Control, in Proceedings of IEEE International Conference on Robotics and Automation, Atlanta, GA, May 1993

[25] Borenstein, J. and Koren, Y., The Vector Field Histogram Fast Obstacle Avoidance for Mobile Robots. IEEE Journal of Robotics and Automation, 7, pp: 278288, 1991.

[26] 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.