



HAL
open science

Real-time Game-theoretic Model Predictive Control for Differential Game of Target Defense

Mukhtar Sani, Ahmad Hably, Bogdan Robu, Jonathan Dumon

► **To cite this version:**

Mukhtar Sani, Ahmad Hably, Bogdan Robu, Jonathan Dumon. Real-time Game-theoretic Model Predictive Control for Differential Game of Target Defense. *Asian Journal of Control*, 2023, 25 (5), pp.3343-3355. 10.1002/asjc.3087 . hal-04050443

HAL Id: hal-04050443

<https://hal.science/hal-04050443>

Submitted on 29 Mar 2023

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.

Real-time Game-theoretic Model Predictive Control for Differential Game of Target Defense

Mukhtar Sani | Ahmad Hably | Bogdan Robu | Jonathan Dumon

Univ. Grenoble Alpes, CNRS, Grenoble-INP^{*}, GIPSA-Lab, 38000 Grenoble, France.

¹Univ. Grenoble Alpes, CNRS, Grenoble-INP^{*}, GIPSA-Lab, 38000 Grenoble, France.

Correspondence

Mukhtar Sani, Univ. Grenoble Alpes, CNRS, Grenoble-INP^{*}, GIPSA-Lab, 38000 Grenoble, France.
Email: mukhtar.sani2707@gmail.com

Present address

[†]Gipsa-lab, 11 rue des Mathematiques, St Martin d'Heres, 38400, France

Funding information

This research was funded by a bilateral agreement between kano state government of Nigeria and French government through Campus France.

Grant/Award Number: 905212G/935965F
Also it has been partially supported by ROBOTEX 2.0 (Grants ROBOTEX ANR-10-EQPX-44-01 and TIRREX ANR-21-ESRE-0015) funded by the French program Investissements d'avenir. Also it has been partially funded by the French LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

This paper evaluates experimentally a novel strategy for solving a variant of the differential game of target defense in presence of obstacles. The game is widely applied in the areas of military defense for protecting important equipment such as a ship, an aircraft, a moving vehicle, or a sensitive installation from a malicious attacker. The state-of-the-art approaches mostly employ an offline optimization strategy that is only applicable to holonomic robots. Moreover, most of the approaches could not autonomously avoid obstacles or take into account uncertainties. As a consequence, this paper presents an online optimization technique, by designing a trade-off parameter that integrates game theory with the model predictive control which allows a non-holonomic defender to intercept the attacker while simultaneously defending the target. Simulations under different conditions as well as several indoor laboratory experiments validate the proposed approach. Moreover, performance is compared with a standard model predictive control approach.

KEYWORDS

Differential game, MPC, mobile robots, obstacles avoidance, Game-theory

1 | INTRODUCTION

Differential game is a powerful tool for analyzing decision-making processes in which multiple dynamic agents interact with each other either in a conflict or in a cooperative scenario. Several variants of the differential game exist in the literature, including the pursuit-evasion differential game where a pursuer or a group of pursuers strives to capture a single evader or group of evaders. Unlike in one-pursuer-one-evader game, where a single pursuer tries to capture a single evader, multi-player pursuit-evasion game entails cooperation between two or more pursuers with in order to capture a single evader. Cooperative strategies that ensure capture in a finite time have been widely reported in the literature. For instant, in [1, 2, 3, 4, 5] a safe-reachable area minimization approach was employed to pursue a single evader by multiple pursuers. The evader's safe-reachable area is computed using the Voronoi partition and then a cooperative strategy would be designed to minimize the safe-reachable area of the evader until it is captured. Also, In [6, 7, 8], the geometric approach is used to tackle the cooperative pursuit problem. When deciding whether to use pure pursuit or cooperative pursuit, the geometric location of the players is crucial. The method works in such a way that any pursuer that is closer to the evader will be the one to pursue the evader while the other remains stationary. When both players are equidistant from the evader, a cooperative strategy using an Apollonian circle would be employed so that both pursuers will pursue the evader concurrently. However in designing cooperative strategies, several works consider the pure pursuit of a single evader by a single pursuer. In [9, 10, 11, 12] for example, a 2-player pursuit-evasion game is solved using reachable sets and potential fields respectively. Model predictive tracking control is employed in [13] that provides evasive maneuvers for a UAV assuming a known pursuer's input. In [14, 15, 16, 17] however, an unknown pursuer's input is considered, thus having to be predicted using double optimization. It is pertinent to note in all the cases of pursuit-evasion game problems, the evader is only striving to avoid capture or at most maximize the capture time. Therefore, incorporating a target whether as a dynamic target or as a static target in the pursuit-evasion problems produces another variant, a *differential game of target defense*. For a static target, a *differential game of static target defense* is produced while for a dynamic target, a *differential game of active target defense* is produced.

The differential game of target defense consists of two pursuit-evasion games that are coupled to produce a game with three players; the defender, an attacker, and a target. The objective of the defender is to intercept an attacker before reaching a target, while the objective of the attacker is to reach the target while avoiding the defender. The objective of the target is to avoid the attacker by cooperating with the defender. This scenario is presented as a differential game of active target defense in [18, 19, 20, 21] for aircraft and missile applications, where the closed-form solution of the game, the players' optimal strategies, and verification of the closed-loop solution using Hamilton-Jacobi-Bellman equations are detailed. These strategies are designed to be applied on a team of slower target aircraft and defender missiles so that they cooperate to intercept a faster attacker missile. However, these approaches are limited to holonomic robots with simple equations of motion without considering obstacle avoidance.

Moreover, another variant of the differential game, known as the reach-void differential game is presented in [22] which involves one-defender-one-attacker. The game environment is splitted into a goal region and a play region such that the attacker strives to enter the play region without being captured, while the defender tries to capture the attacker in the play region. The paper focused on the defense problem where the defender is a Dubin's car with non-zero capture radius and the attacker has a simple-motion model. In this case, the defense strategy is to track the heading reference of the attacker from an evasion space. However, the approach looks trivial because the considered attacker is a holonomic system and is not playing any game against the defender. Other related works on reach-void games could be found in [23, 24, 25].

In this present paper, the differential game of static target defense is addressed. It involves scenario where a

single defender strives to intercept the attacker before it reaches an uncooperative static target, while the attacker is pursuing the target and avoiding the defender simultaneously. This problem is also known as the game of guarding target which was firstly explained by the famous US scientist, Rufus Isaacs in his book [26] as a special case of the differential game of active target defense. The underlying assumption is that both the defender and the attacker know their positions and that of the target. We consider a situation where the speed limits of the players are the same and the distance between each player and the target is equal. This scenario improves our previous work in [15, 16, 17] in the sense that the defender's objective is not limited to the pursuit of the opponent but as well includes protection of the target. Also, the attacker's objective is more specific. We propose an improved formulation of nonlinear model predictive control (NMPC) which we termed as game-theoretic model predictive control (GTMPC) to handle the differential game of static target defense (DGSTD) in both simple and complex environments. From the perspective of the defender, GTMPC contains a parameter that makes a trade-off between interception of the attacker and protection of the target.

The contribution of this paper is that it provides an intelligent solution to Isaacs's problem of guarding a target that could work for both holonomic and nonholonomic robots. The method is robust to the presence of obstacles and most importantly, our method could achieve interception in a situation that requires multiple pursuers with a single one. The paper is organised as follows. Section 2 presents the DGSTD as well as the termination conditions. The proposed game-theoretic model predictive control approach is presented in section 3. Simulation and experimental results are presented in sections 4 and 5 respectively. Finally, in section 6 several conclusions are drawn and suggestion for future perspective is made.

2 | DIFFERENTIAL GAME OF STATIC TARGET DEFENSE

The attacker (A) and the defender (D) have turn radius constraints, that is, they are nonholonomic while the target (T) is represented by a Cartesian position. While the position of the static target is regarded as the center of a particular object approximated a sphere, the kinematic model of nonholonomic wheeled mobile robot can be represented in equation (1), for more details the reader can refer [27, 28].

$$\begin{cases} \dot{x}(t) &= v(t) \cos \theta(t) \\ \dot{y}(t) &= v(t) \sin \theta(t) \\ \dot{\theta}(t) &= \omega(t) \end{cases} \quad (1)$$

where the state variable $\mathbf{x} = [x, y, \theta]^T$ represent the position of the robot in chassis frame of reference and the heading angle, while the control variables $\mathbf{u} = [v, \omega]^T$ represents the linear speed and the angular speed which are also bounded as $\mathbf{v}_{min} \leq \mathbf{v} \leq \mathbf{v}_{max}$ and $\omega_{min} \leq \omega \leq \omega_{max}$ respectively. Since the robots used in this work are not a point mass, the sizes of the defender and the attacker are defined as R_D and R_A respectively. The nonholonomic nature of this type of system imposes limitations on acceptable system velocity, which turns out to be useful as it limits the number of control inputs while maintaining the full controllability of the system in state-space [29]. This advantage, however, imposes a difficulty that is connected with the point-stabilization task, that is the task cannot be accomplished with a pure feedback algorithm due to Brockett's condition which implied that the linearized nonholonomic model is not stabilizable [30].

In the context of our DGSTD, the state variables of the defender and the attacker are given by $\mathbf{x}_D = [x_D, y_D, \theta_D]^T$ and $\mathbf{x}_A = [x_A, y_A, \theta_A]^T$ respectively, while the position of the target was given as $\mathbf{x}_T = [x_T, y_T]^T$. Also, the defender's

control variables are the linear speed v_D and angular speed ω_D , thus $\mathbf{u}_D = [v_D, \omega_D]^T$, while the attacker's control variables would then be represented as $\mathbf{u}_A = [v_A, \omega_A]^T$.

The complete set of the DGSTD is specified by $\mathbf{x} = (x_D, y_D, \theta_D, x_A, y_A, \theta_A)$ defined within the game environment, the Euclidean plane. The game dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}_A, \mathbf{u}_D)$ are defined as in terms of eq.1 with the following initial conditions: $\mathbf{x}_0 = (x_{D_0}, y_{D_0}, \theta_{D_0}, x_{A_0}, y_{A_0}, \theta_{A_0})$. Both the states and the controls are constrained unlike in [18].

Having defined the players of the game, models, and the game environment, we will define players' objectives and the game termination conditions. The objective of the defender is to pursue the attacker and protect the target while the objective of the attacker, on the other hand, is to reach the target and avoid the defender. Since the defender and the attacker have the same capabilities and the distance between each other and from the target is equidistant. Then the defender will never win the game when it blindly tracks the attacker. Without any cooperation from the target, the question is how can the defender intelligently pursue the two objectives explained herein and win the game while simultaneously avoiding all the obstacles whose position and size are defined as x_{obs}, y_{obs} while having a size R_{obs} .

To define the termination of the game, the notions of a game of kind and a game of degree are revised. Game of kind has two finite outcomes depending on whether or not the player can achieve its objective. For example, in a pursuit game, the objective might be capture and in a battle game, the objective could be the destruction of the opponent. Game of degree on the other hand has a continuum of possible payoffs. The players then try to maximize or minimize the established pay-off. Consider a given DGSTD where the initial condition is in a part of the state space where a strategy for the defender exists such that, under optimal play, the defender intercepts the attacker, i.e the attacker was not able to reach the target, then the DGSTD game of degree is played where the attacker tries to minimize the terminal distance between the attacker and the target while the defender tries to maximize it. When our interest lies on whether the defender can capture the attacker or whether the attacker reaches the target, that way, we are dealing with a game of kind. To end our game, we used the concept of a game of kind, which gives birth to two termination criteria. The first condition is when the attacker was intercepted by the defender. This happens when the distance between the defender and the attacker is equal to some threshold value as given in eq.(2).

$$C_1 : \sqrt{(x_A - x_D)^2 + (y_A - y_D)^2} = l_1 \quad (2)$$

where $l_1 = R_A + R_D$.

The second termination condition is when the attacker captures the target. This occurs when the attacker eventually reaches the target without being intercepted by the defender as given in eq.(3).

$$C_2 : \sqrt{(x_T - x_A)^2 + (y_T - y_A)^2} = l_2 \quad (3)$$

where $l_2 = R_T + R_A$.

Therefore, the termination condition of the game can be defined as (4) which is the combination of the two conditions (2) and (3).

$$C = C_1 \cup C_2 \quad (4)$$

3 | GAME-THEORETIC MODEL PREDICTIVE CONTROL

Game-theoretic model predictive control (GT-MPC) approach proposed in this section combines the notions of game-theory and model predictive control. Game theory is a mathematical tools that model systems as intelligent rational decision-makers where an agent considers the opponent's strategy when selecting its optimal strategy. Each player in a game seek to find its Nash equilibrium that is considered as the best response a particular player can give knowing that the opponent is also doing his best. The relationship between game theory and control was reviewed in [31].

Information on the opponent strategy or intention is critical to obtaining a Nash equilibrium. Since in real game, a player cannot share his strategy to the opponent, prediction is possible via double optimization as presented in [14, 15, 16]. If the intention of the opponent is known, then a player has to leverage on that in order to increase the chances of winning.

Model predictive control (MPC) also known as receding horizon control (RHC) is a feedback implementation of optimal control using finite prediction horizon and online optimization. The technique allows predicting the future states of the systems using its mathematical model and an initial state so as to obtain the future control sequence. A measurement of the current state is used at each decision instant to reinitialize the system. Only the first part of the control sequence is applied to the system, the remaining are discarded. The advantages of MPC comprises its conceptual ability to handle control of multi-variable coupled dynamical systems, constraints on the states, constraints on control inputs and nonlinearities in the systems model. In addition, MPC has systematic design approach and has a well understood tuning parameters, i.e prediction horizon length and weighting matrices [32, 33, 34, 35, 36]. A discrete nonlinear model predictive control (NMPC) can be formulated by solving the following optimal control problem (OCP):

$$\min_{\mathbf{u} \in \mathbb{R}^{n \times N}} J_N(\mathbf{x}_0, \mathbf{u}) \quad (5)$$

subject to

$$\begin{cases} \mathbf{x}(0) = \mathbf{x}_0, \\ \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)); \quad k \in \{0, 1, \dots, N-1\}, \\ \mathbf{x}_{min} \leq \mathbf{x}(k) \leq \mathbf{x}_{max} \quad k \in \{1, 2, \dots, N\}, \\ \mathbf{u}_{min} \leq \mathbf{u}(k) \leq \mathbf{u}_{max} \quad k \in \{0, 1, \dots, N-1\} \end{cases} \quad (6)$$

where:

$$J_N(\mathbf{x}_0, \mathbf{u}) = \sum_{k=0}^{N-1} V(\mathbf{x}(k), \mathbf{u}(k)) + W(\mathbf{x}(N)) \quad (7)$$

The term $V(\mathbf{x}(k), \mathbf{u}(k))$ is called the running cost which can be computed by penalizing the deviation of the system's state $\mathbf{x}(k)$ and control input $\mathbf{u}(k)$ from the reference state $\mathbf{x}^r(k)$ and reference control input $\mathbf{u}^r(k)$ respectively. Generally, the running cost is defined as:

$$V(\mathbf{x}(k), \mathbf{u}(k)) = \|\mathbf{x}(k) - \mathbf{x}^r(k)\|_Q^2 + \|\mathbf{u}(k) - \mathbf{u}^r(k)\|_R^2 \quad (8)$$

where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are positive definite symmetric weighting matrices. N is the prediction horizon, assuming that the length of the prediction and control horizon is the same. The choice of N is not straightforward, as

a large value would imply a non-realistic behavior where all the future movements of the agent are known, whereas a small value would imply a reaction based only on the current measurements. If the sampling time is small compared to the robot's dynamics, we can suppose that the estimation of few further positions would not be very far from the reality. As previously explained, in point stabilization problem, the state reference $\mathbf{x}^r(k)$ is a fixed value, thus the control, reference $\mathbf{u}^r(k) = 0$. In the case of trajectory tracking problem, the state reference $\mathbf{x}^r(k)$ is time varying, therefore the deviation of control input from the reference can be penalized due to computational advantages such as rendering the optimal control problem easier, avoiding control values with expensive energy [29].

The term $W(\mathbf{x}(N))$ is referred to as terminal cost which is used for stability purpose. It can be computed by penalizing the last entry from the state prediction $\mathbf{x}(N)$ from its reference $\mathbf{x}^r(N)$. Terminal cost can be defined as:

$$W(\mathbf{x}(N)) = \|\mathbf{x}(N) - \mathbf{x}^r(N)\|_P^2 \quad (9)$$

where $P \in \mathbb{R}^{n \times n}$ is a positive definite weighting matrix. The solution of the optimal control problem (5) is the optimal control sequence of the form:

$$\mathbf{u}^* = (\mathbf{u}^*(0), \mathbf{u}^*(1), \dots, \mathbf{u}^*(N-1)) \quad (10)$$

By combining the concepts of game theory and MPC together, the defender already has information on the attacker's intention. The first objective of the defender is to intercept the attacker before it reaches the target, while the second objective is to track the attacker. To satisfy both objectives concurrently, the defender is made to track a reference that depends on the parameter α which made a trade-off between the attacker's and the target's positions. This parameter is designed and tuned discretely. Eventually, the defender's control is formulated such that at every sampling time, the following optimal control problem is solved:

$$\min_{\mathbf{u}_D} J = \|\mathbf{x}_D(N) - \mathbf{x}_R\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_D(k) - \mathbf{x}_R\|_Q^2 + \|\mathbf{u}_D(k)\|_R^2 \quad (11)$$

where,

$$\mathbf{x}_R = (1 - \alpha)\mathbf{x}_A + \alpha\mathbf{x}_T \quad (12a)$$

$$\alpha \in (0, 1) \quad (12b)$$

Subject to:

$$\mathbf{x}_D(k+1) = f(\mathbf{x}_D(k), \mathbf{u}_D(k)), \quad k = 0, 1, \dots, N \quad (13a)$$

$$\sqrt{(x_D(k) - x_{obs})^2 + (y_D(k) - y_{obs})^2} \geq (R_D + R_{obs}) \quad (13b)$$

$$\mathbf{x}_{D_{min}} \leq \mathbf{x}_D(k) \leq \mathbf{x}_{D_{max}} \quad (13c)$$

$$\mathbf{u}_{D_{min}} \leq \mathbf{u}_D(k) \leq \mathbf{u}_{D_{max}} \quad (13d)$$

As we notice from (12b), the cost function depends on the parameter α which varies from 0 to 1. As the parameter increases from 0 towards 1, we are giving more weight to protecting the target than pursuing the attacker. Furthermore, in equations (13a to 13d) we have included some constraints in the formulation which comprises the defender's model, obstacles' avoidance, states, and input constraints. Due to the fact that nonholonomic systems are not linearly stabilizable, a terminal cost was added to asymptotically stabilize the controller. Conversely, the attacker is interested in reaching the target as quickly as possible while avoiding the defender, the problem can be formulated using conventional NMPC such that the target's state is regarded as the reference signal while the defender is taken as a dynamic obstacle. Therefore, the attacker will be solving the following OCP at every time instant.

$$\min_{\mathbf{u}_A} J = \|\mathbf{x}_A(N) - \mathbf{x}_T\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_A(k) - \mathbf{x}_T\|_Q^2 + \|\mathbf{u}_A(k)\|_R^2 \quad (14)$$

Subject to:

$$\mathbf{x}_A(k+1) = f(\mathbf{x}_A(k), \mathbf{u}_A(k)), \quad k = 0, 1, \dots, N \quad (15a)$$

$$\sqrt{(x_A(k+1) - x_D)^2 + (y_A(k+1) - y_D)^2} \geq (R_A + R_D) \quad (15b)$$

$$\sqrt{(x_A(k) - x_{obs})^2 + (y_A(k) - y_{obs})^2} \geq (R_A + R_{obs}) \quad (15c)$$

$$\mathbf{x}_{A_{min}} \leq \mathbf{x}_A(k) \leq \mathbf{x}_{A_{max}} \quad (15d)$$

$$\mathbf{u}_{A_{min}} \leq \mathbf{u}_A(k) \leq \mathbf{u}_{A_{max}} \quad (15e)$$

This cost function (14) comprises the running cost which is the sum of stages predicted over the prediction horizon and the terminal cost for stabilizing the controller. \mathbf{x}_A and \mathbf{u}_A respectively denote the states and control variables of the attacker whereas \mathbf{x}_T represents the position of the target taken as a reference to go. The constraint given by equation (15a) represents the nonlinear model of the robot. Two obstacle avoidance constraints are added to the formulation. The constraint (15b) is employed to make the attacker avoid the defender as a dynamic obstacle, while the second constraint (15c) is employed to avoid static obstacles that might be encountered. The weight matrices are tuned and the best values that stabilize the controller are chosen. The Q and R matrices are found to be diagonal matrices with diagonal elements obtained as (1, 1, 0.001) and (1, 1) respectively. The weight on the terminal penalty cost is found to be $1000 * Q$. In a general case, as we are on a problem of distance minimization between the defender and attacker, and not on an angle minimization problem, the weights of the first two elements in Q and R matrices should be the same. In this case we chose them to be both equal to 1 as it is the simplest case. Concerning the third element in the state space vector, namely θ , it was previously show in [17], [15] or [37] that it's weight should be much smaller in order to avoid unrealistic behavior where the defender, even though very far, will first turn around

itself, so to match the angle, before deciding to pursue the attacker.

4 | SIMULATION RESULTS

Simulations of several scenarios of the proposed differential games of static target defense are implemented in a Matlab/Simulink environment. An open-source symbolic framework for automatic differentiation and optimal control software, CasADi [38] is interfaced to code the algorithms using a multiple shooting approach, which convert optimal control problems into a nonlinear programming problem. The nonlinear model of a nonholonomic robot is used to implement the control where the initial conditions are selected randomly such that both the defender and the attacker are at the same distance from the target with both having the same the speed limits. This condition make the game interesting and different from the those presented in the literature where one is faster or more/less agile. For the purpose of comparison, we presented all the results for the same initial conditions, even though the algorithms is applicable under different ones with similar scenario. The initial condition of the defender is given as $x_{D_0} = [-4m; 4m; 0rad]$, the initial condition of the attacker is given as $x_{A_0} = [-4m; -4m; 0rad]$, and the position of the target is given as $x_T = [4.5m; 0m]$. Each robot is approximated as a circle with a diameter of $0.2m$, thus the value of the R_D and R_A are both $0.1m$, while the radius of the target is defined as $0.05m$. The positions and sizes of the three circular obstacles are given as $[-2m; 0m]$, $[0m, -2m]$, $[0m, 2m]$ while the size of the obstacle is given as $R_{Obs} = 0.5m$. Moreover, the bounds on the control variables which is translates into inequality constraints are defined as $v_{max} = 0.3 m/s$, $v_{min} = -0.3 m/s$, $\omega_{max} = \pi/3 rad/s$, and $\omega_{min} = -\pi/3 rad/s$ for both robots.

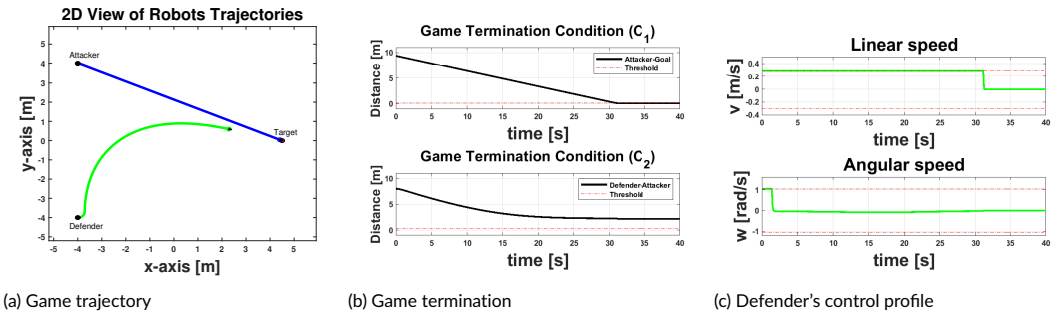


FIGURE 1 Differential game of static target defense using NMPC

We started by employing NMPC controller on the defender to track an attacker that is using NMPC with dynamic obstacle (defender) avoidance in addition to its aim of reaching the target. The result is depicted in Figure 1, where the game trajectory presented in Figure 1a show that the defender's attempt was unsuccessful as the attacker reached the target in 31secs . As it can be seen at the top Figure1b, the game was terminated using the termination conditions presented in (2), while the control profile of the defender is presented in Figure 1c. It can be seen that the attacker was able to reach the target without the defender preventing it. This is because the defender's NMPC strategy is just a tracking problem, and it would follow the attacker blindly. Similarly, we added some obstacles in the game environment and play the game under the same situation with the same initial conditions, yet the defender could not intercept the attacker before the latter reaches the target. The result of this scenario is presented in Figure 2.

GTMPCC approach is then employed to tackle the situation from the perspective of the defender. However, the

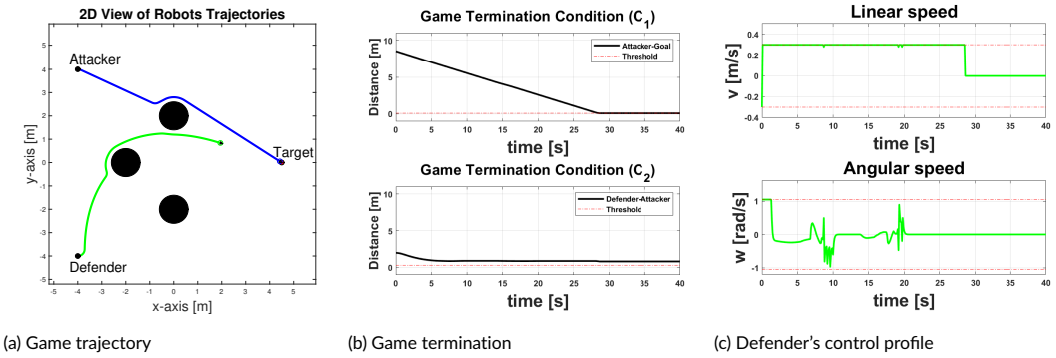


FIGURE 2 Differential game of static target defense with obstacles avoidance using NMPC

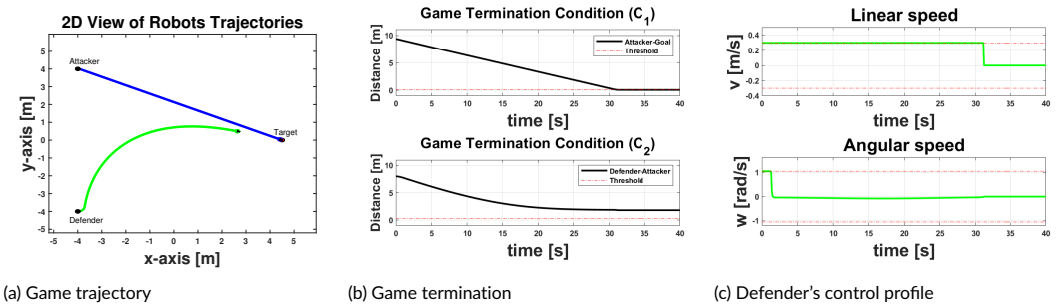


FIGURE 3 Attacker-defender-target game using GT-NMPC, $\alpha = 0.10$

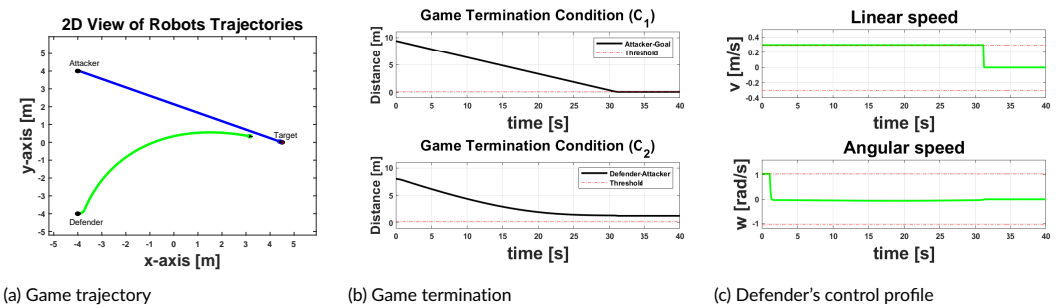


FIGURE 4 Attacker-defender-target game using GT-NMPC, $\alpha = 0.25$

value of the parameter α is paramount and would determine whether the defender will put emphasis on attacker tracking or target protection. To obtain the best value of α in this tricky situation, several numerical simulations are

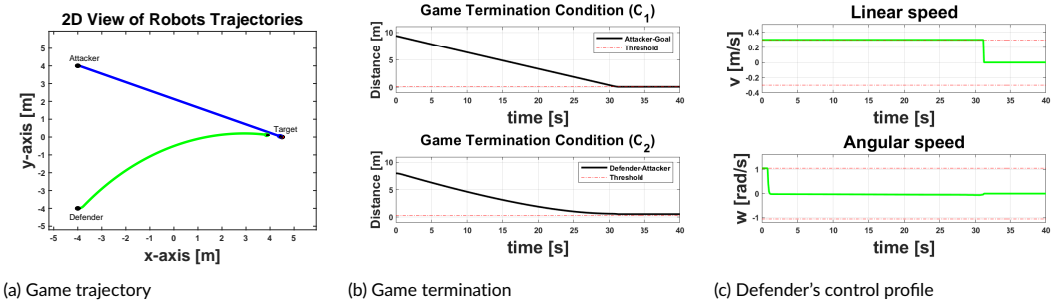


FIGURE 5 Attacker-defender-target game using GT-NMPC, $\alpha = 0.5$

conducted using several values ($\alpha = 0.10, 0.25, 0.50, 0.75$ and 0.90). In Figure 3, the defender (green player) employed GTMPC with $\alpha = 0.10$ tried to intercept the attacker (blue player) that is using a conventional NMPC, before it reaches the target (red). The game trajectory presented in Figure 3a shows that the defender's attempt was unsuccessful, as the attacker reached the target in 31seconds . As it can be seen at the top Figure 3b, the game was terminated using the termination conditions presented in (2). The control profile of the defender is presented in Figure 3c. With this value of $\alpha = 0.1$, tracking the attacker is given more importance than protecting the target. Similar situation was repeated in Figure 4 and Figure 5 where the values of α are 0.25 and 0.5 respectively. It can be seen in both cases that the defender could not intercept the attacker.

Besides, in Figure 6, the defender (green player) employed GTMPC with $\alpha = 0.75$ tried to intercept the attacker (blue player) that is using a conventional NMPC, before it reaches the target (red). The game trajectory presented in Figure 6a shows that the defender's attempt was successful this time as the defender captured the attacker in 28.5seconds . As it can be seen at the top Figure 6b, the game was terminated using the termination conditions presented in (2). The control profile of the defender is presented in Figure 6c. A similar outcome played out in Figure 7 when the defender employed GTMPC with $\alpha = 0.90$. It is evident that capture is guaranteed when target protection is given more priority than attacker tracking.

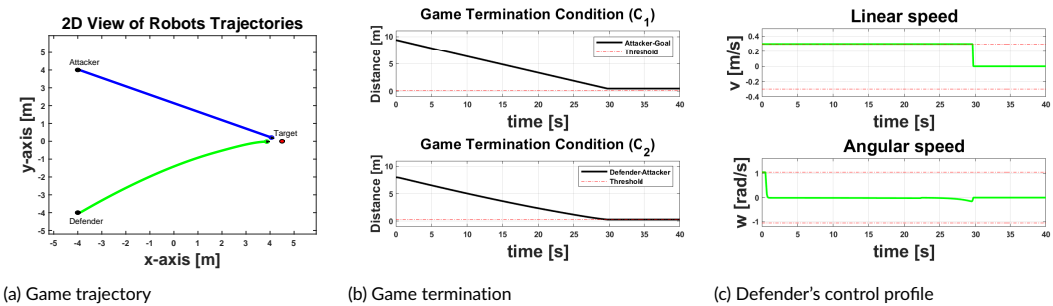


FIGURE 6 Attacker-defender-target game using GT-NMPC, $\alpha = 0.75$

Since we are looking for the minimal value of α smaller than 1 which can also guarantee capture, we adopted $\alpha = 0.75$ as a good value and use it for further testing in more complex situations including obstacles avoidance

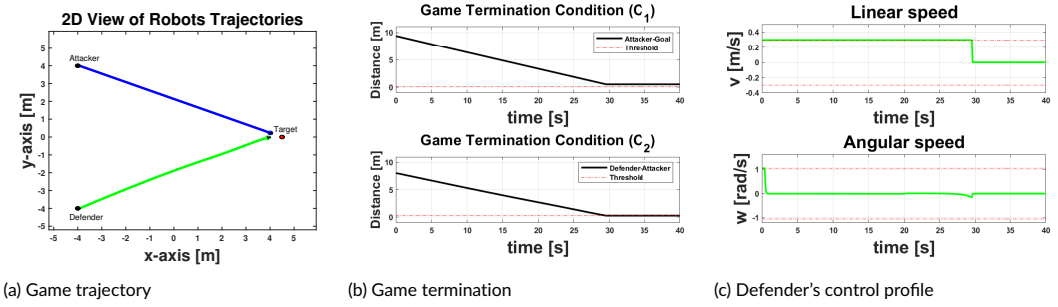


FIGURE 7 Attacker-defender-target game using GT-NMPC, $\alpha = 0.90$

incorporation and in experimental validations. In Figure 4, the defender employed GTMPC to intercept the attacker before it reaches the target. The game trajectory is presented in Figure 8a in which the attacker (blue player) aimed at the target while the defender (green player) pursued the attacker wisely and was able to capture it before reaching the target (red) in the presence of three obstacles (black circles). The termination conditions are presented in Figure 8b which also shows that the second condition given by equation (3) holds and the capture time is 17 seconds. The control profile of the defender was presented in Figure 8c.

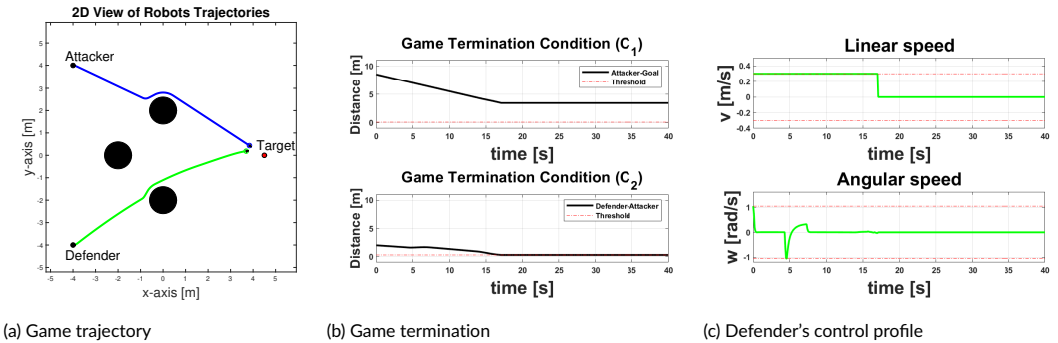


FIGURE 8 Differential game of static target defense with obstacles avoidance using GTMPC, $\alpha = 0.75$

5 | EXPERIMENTAL VALIDATION

This section presents the physical implementation of the proposed control algorithm on the mobile robots with real-time measurements and real transmission of information to the actual system. Despite implementing the algorithms on a nonlinear Simulink model, physical experiment is the most sure way approach for validating controllers.

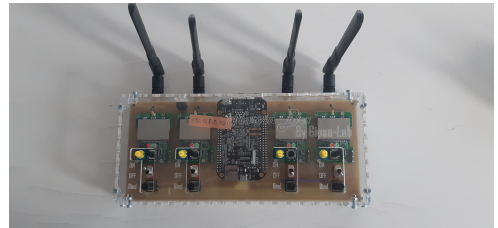
5.1 | Experimental set-up

Two nonholonomic mobile robots (uni-cycle type) depicted in Figures 9c and 9d are fabricated at Gipsa-lab and named Robot 1 and Robot 2 respectively. Each robot consists of two controlled wheels at the left and right sides of the robot and a freewheel at the front to support the robot. Two continuous rotation servo motors are coupled to the left and right wheels. Each motor takes in the speed command signal (ω_r or ω_l) which commands the robot how fast to rotate and in which direction (forward or backward). The speed commands of each servo-motor are simultaneously received wirelessly from the IRC board in Figure 9b which communicate with the robot's Spectrum DSMX receiver at 2.4GHz, for extra interference protection and faster reconnection times. The communication between the Simulink and the IRC board is done via the UDP-protocol block in the Simulink Real-Time™.

The motion capture system used in this work consists of 8 Miquis M3 cameras from Qualisys company, which is used as a sensor to track the movement of the robots. The Miquis cameras are designed to capture accurate data with very low latency. Each 2MP camera has a resolution of 1824×1088 pixels as well as a recording speed of 340 frames per second. The cameras are light and portable (around 700 grams each) and have been properly positioned to cover all the experimental ground, as their maximum capture distance is around 15 meters, which is close to the maximum size of the experimental ground of $16m^2$. The cameras, depicted in Figure 9a, are synchronized using an integrated Qualisys software (namely Camera Sync Unit) which communicates with a Raspberry-pi board containing a python code for converting the quaternions to Euler angles. Camera Sync Unit supports a wide range of synchronization and communication protocols including PTP, SMPTE, IRIG. The states of both robots, as well as any moving obstacle, are then sent in real-time to a Matlab™ environment through Simulink Real-Time™ via UDP-protocol.



(a) Overview of the motion capture system



(b) IRC Communication system



(c) Robot 1



(d) Robot 2

FIGURE 9 Robots, communication and motion capture systems.

The controllers then use the states and other information to compute the control inputs v and ω which are con-

verted to the angular speeds of the left (ω_l) and right (ω_r) wheels of the robot by the following equation:

$$\begin{cases} \omega_r = (2v + \omega R_{rob})/2r \\ \omega_l = (2v - \omega R_{rob})/2r \end{cases} \quad (16)$$

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

Each experiment can terminate in either three ways, i.e when the attacker reaches the target or when the defender captured the attacker, or when the game time elapses (120 seconds) which happened when the defender prevented the attacker from reaching the target. Also, the initial positions of each game were selected so that the players are equidistant from the target. Unlike in the case of simulations, the game environment for the experiments is between -2m to +2m on both axis. Moreover, the bounds on the control variables which translates into inequality constraints are defined as $v_{max} = 0.065m/s$, $v_{min} = -0.065m/s$, $\omega_{max} = 0.05rad/s$, and $\omega_{min} = -0.05rad/s$ for both robots. The sample time is 0.1 seconds and the prediction horizon N is equal to 10.

5.2 | Experiments in an obstacle-free environment

GT-MPC was implemented on the defender in figure 10 in a bounded-free environment. The game trajectory in figure 10a shows that the defender prevented the attacker from reaching the target until the game terminated. The attacker also considers the defender as a dynamic obstacle, that is why it moves away from the defender to avoid capture. The strategy profiles of the defender and the attacker are shown in figures 10b and 10c respectively. The slight oscillations seen on the real speeds is because the speeds are not measured directly but rather obtains off-line from the real-time state measurements.

To show the advantage of the proposed controller, the scenario was repeated with the defender employing the conventional NMPC in figure 11. The game trajectory in figure 11a shows that the defender tracks the attacker but could not intercept or prevent the attacker from reaching the target. This means that the defender blindly tracks the attacker without any form of intelligence, thus unless with speed advantage, the defender can never intercept the attacker in this scenario. The control strategy profiles of the defender and the attacker are respectively shown in figures 11b and 11c.

5.3 | Experiments in an environment with obstacles

Obstacles avoidance was incorporated in the formulation of the proposed controller. Three virtual obstacles are placed at strategic positions within the game environment as shown in Figure 12. The trajectories of the robots in figure 12a show that the defender intercepted the attacker before it reaches the target. The strategy profiles of the players are shown in figures 12b and 12c. However, when the game was repeated with the defender using NMPC in figure 13, the game trajectory in figure 13a shows that the defender could not intercept the attacker. The strategy profiles of both players are presented in figures 13b and 13c respectively.

6 | CONCLUSION

This paper studied differential games under the context of static target defense and proposed a solution using the combination of nonlinear model predictive control and game theory. The drawbacks of the previous approaches are

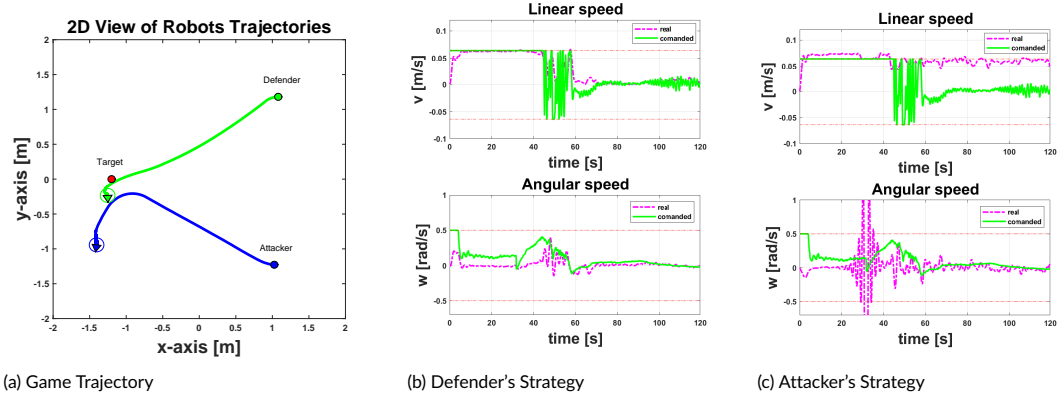


FIGURE 10 Differential Game of Static Target Defense Using GT-NMPC

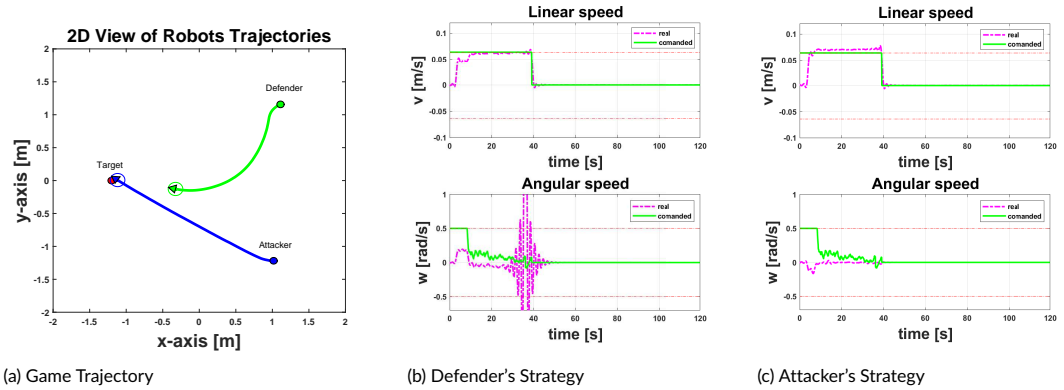


FIGURE 11 Differential Game of Static Target Defense Using NMPC

leveraged to propose a novel game-theoretic model predictive control. A parameter α which provides a trade-off between the pursuing objective and the defending objective was introduced and tuned such that the best value of the parameter that makes it possible for a single defender to intercept the attacker before reaching the target. Even though the algorithms are applied on a nonholonomic mobile robot, it can still be useful for other mobile robots such as unmanned aerial vehicles, unmanned surface vehicles with known dynamics. Several simulations and real-time experiments in the presence and the absence of obstacles are conducted. It can be concluded that the proposed approach performed better than tracking methods such as NMPC. As for future perspectives, it will be interesting to employ learning based model predictive control approach. Also, cooperative strategy between the defender and a moving target should be exploited based on the approach presented. On the theoretical side, a very important improvement would be the in finding analytically the optimal value of the α parameter, as the overall behavior seems to be monotonous to this parameter.

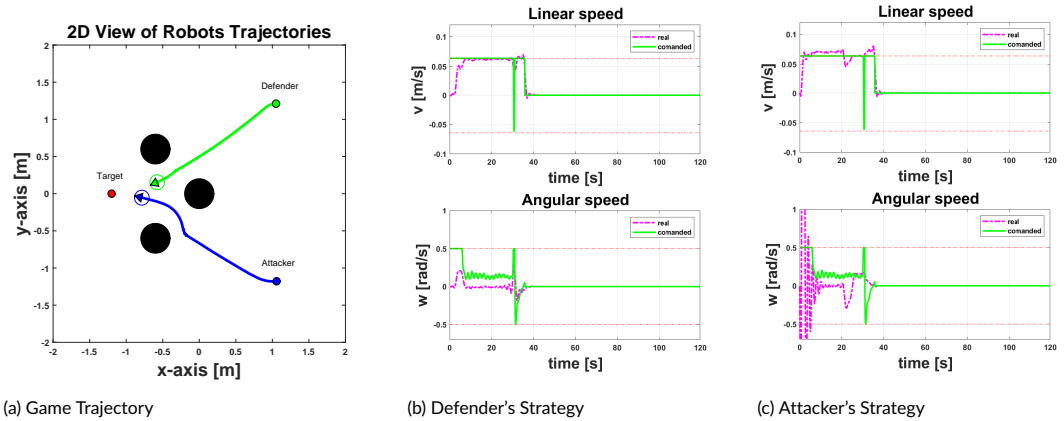


FIGURE 12 Differential Game of Static Target Defense Using GT-NMPC

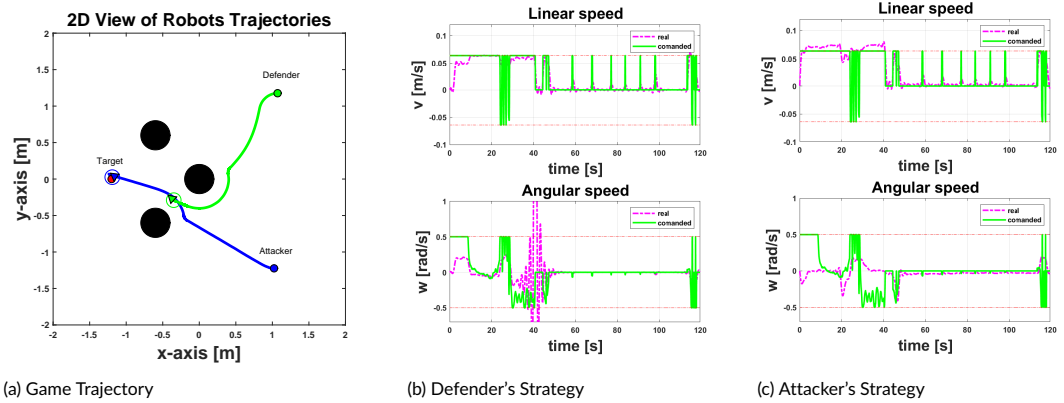


FIGURE 13 Differential Game of Static Target Defense Using NMPC

References

- [1] Huang H, Zhang W, Ding J, Stipanović D, Tomlin C. Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers. Proceedings of the IEEE Conference on Decision and Control; 2011. p. 4835–4840.
- [2] Kothari M, Manathara JG, Postlethwaite I. Cooperative Multiple Pursuers against a Single Evader. Journal of Intelligent and Robotic Systems 2017;86(3):551–567. <https://doi.org/10.1007/s10846-016-0423-3>.
- [3] Zhou Z, Zhang W, Ding J, Huang H, Stipanović DM, Tomlin CJ. Cooperative pursuit with Voronoi partitions. Automatica 2016;72:64–72. <https://www.sciencedirect.com/science/article/pii/S0005109816301911>.
- [4] Bakolas E, Tsiotras P. Optimal pursuit of moving targets using dynamic Voronoi diagrams. In: 49th IEEE Conference on Decision and Control (CDC); 2010. p. 7431–7436.

- [5] Pierson A, Wang Z, Schwager M. Intercepting Rogue Robots: An Algorithm for Capturing Multiple Evaders With Multiple Pursuers. *IEEE Robotics and Automation Letters* 2017;2(2):530–537.
- [6] Garcia E, Fuchs ZE, Milutinovic D, Casbeer DW, Pachter M. A Geometric Approach for the Cooperative Two-Pursuer One-Evader Differential Game. *IFAC-PapersOnLine* 2017;50(1):15209–15214. <https://www.sciencedirect.com/science/article/pii/S240589631733183x>, 20th IFAC World Congress.
- [7] Garcia E, Casbeer DW, Von Moll A, Pachter M. Cooperative Two-Pursuer One-Evader Blocking Differential Game. In: 2019 American Control Conference (ACC); 2019. p. 2702–2709.
- [8] Von Moll A, Casbeer DW, Garcia E, Milutinović D. Pursuit-evasion of an Evader by Multiple Pursuers. In: 2018 International Conference on Unmanned Aircraft Systems (ICUAS); 2018. p. 133–142.
- [9] Mitchell IM, Bayen AM, Tomlin CJ. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control* 2005;50(7):947–957.
- [10] Sun W, Tsiotras P, Lolla T, Subramani DN, Lermusiaux PFJ. Pursuit-evasion games in dynamic flow fields via reachability set analysis. In: 2017 American Control Conference (ACC); 2017. p. 4595–4600.
- [11] Dong J, Zhang X, Jia X. Strategies of Pursuit-Evasion Game Based on Improved Potential Field and Differential Game Theory for Mobile Robots. In: 2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control; 2012. p. 1452–1456.
- [12] Kumar A, Ojha A. Experimental Evaluation of Certain Pursuit and Evasion Schemes for Wheeled Mobile Robots. *International Journal of Automation and Computing* 2019;16:491–510. <https://doi.org/10.1007/s11633-018-1151-x>.
- [13] Eklund JM, Sprinkle J, Sastry SS. Switched and Symmetric Pursuit/Evasion Games Using Online Model Predictive Control With Application to Autonomous Aircraft. *IEEE Transactions on Control Systems Technology* 2012;20(3):604–620.
- [14] Tzannetos G, Marantos P, Kyriakopoulos KJ. A competitive differential game between an unmanned aerial and a ground vehicle using model predictive control. In: 2016 24th Mediterranean Conference on Control and Automation (MED); 2016. p. 1053–1058.
- [15] Sani M, Robu B, Hably A. Pursuit-evasion Game for Nonholonomic Mobile Robots With Obstacle Avoidance using NMPC. In: 2020 28th Mediterranean Conference on Control and Automation (MED); 2020. p. 978–983.
- [16] Sani M, Robu B, Hably A. Pursuit-evasion Games Based on Game-theoretic and Model Predictive Control Algorithms. In: ICCAD 2021: 5th IEEE-International Conference on Control, Automation and Diagnosis; 2021. .
- [17] Sani M, Robu B, Hably A. Limited Information Model Predictive Control for Pursuit-evasion Games. In: 60th Conference on Decision and Control (CDC2021); 2021. .
- [18] Garcia E, Casbeer DW, Pachter M. Design and Analysis of State-Feedback Optimal Strategies for the Differential Game of Active Defense. *IEEE Transactions on Automatic Control* 2019;64(2):553–568.
- [19] Liang L, Peng Z, Zhang F, Li X. Two coupled pursuit-evasion games in target-attacker-defender problem. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC); 2017. p. 5596–5601.
- [20] Getz WM, Leitmann G. Qualitative differential games with two targets. *Journal of Mathematical Analysis and Applications* 1979;68(2):421–430. <https://www.sciencedirect.com/science/article/pii/0022247X79901264>.
- [21] Getz WM, Pachter M. Capturability in a two-target 'game of two cars'. *Journal of Guidance and Control* 1981;4(1):15–21. <https://doi.org/10.2514/3.19715>.
- [22] Lai H, Yan R, Zhang W, Shi Z, Zhong Y. Reach-Avoid Differential Games via Finite-Time Heading Tracking. In: 2021 60th IEEE Conference on Decision and Control (CDC); 2021. p. 1656–1662.

- [23] Yan R, Shi Z, Zhong Y. Task Assignment for Multiplayer Reach-Avoid Games in Convex Domains via Analytical Barriers. *IEEE Transactions on Robotics* 2020;36(1):107–124.
- [24] Zhou J, Wu X, Lv Y, Li X, Liu Z. Recent progress on the study of multi-vehicle coordination in cooperative attack and defense: An overview. *Asian Journal of Control* 2022;24(2):794–809. <https://onlinelibrary.wiley.com/doi/abs/10.1002/asjc.2685>.
- [25] Yan R, Duan X, Shi Z, Zhong Y, Bullo F. Matching-based capture strategies for 3D heterogeneous multiplayer reach-avoid differential games. *Automatica* 2022;140:110207. <https://www.sciencedirect.com/science/article/pii/S0005109822000528>.
- [26] Isaacs R. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. Dover Books on Mathematics, Dover Publications; 1956. <https://books.google.fr/books?id=GPjBAgAAQBAJ>.
- [27] Lynch KM, Park FC. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press; 2017. <https://books.google.fr/books?id=8uS3AQAACAAJ>.
- [28] Jualin L. *Mobile Robotics*. Dover Books on Mathematics, Elsevier Publications; 2015.
- [29] Mehrez MW. *Optimization Based Solutions for Control and State Estimation in Non-holonomic Mobile Robots: Stability, Distributed Control, and Relative Localization*. PhD thesis, Memorial University of Newfoundland, Canada; 2017.
- [30] Brockett RW. Nonlinear Control theory and Differential Geometry,. In: *International Congress of Mathematics*; 1983. p. 1357–1368.
- [31] Marden JR, Shamma JS. *Game Theory and Control*. *Annual Review of Control, Robotics, and Autonomous Systems* 2018;1(1):105–134. <https://doi.org/10.1146/annurev-control-060117-105102>.
- [32] Wang L. *Model Predictive Control Systems Design and Implementation Using MATLAB*. Springer; 2009.
- [33] Alamir M. *Stabilization of Nonlinear Systems Using Receding-horizon Control Schemes: A Parametrized Approach for Fast Systems*. Springer; 2006.
- [34] Rawlings JB, Mayne DQ, Diehl MM. *Model Predictive Control: Theory, Computation and Design*. 2nd Edition, Nob Hill Publishing, LLC; 2019.
- [35] Farag W. Real-time NMPC path tracker for autonomous vehicles. *Asian Journal of Control* 2021;23(4):1952–1965. <https://onlinelibrary.wiley.com/doi/abs/10.1002/asjc.2335>.
- [36] Jia C, Xu H, Wang L. Robust nonlinear model predictive control for automatic train operation based on constraint tightening strategy. *Asian Journal of Control* 2020;.
- [37] Sani M. *Real-time Control of mobile robots using Model Predictive Control and Game-theoretic approaches*. PhD thesis, Université Grenoble Alpes [2020-....]; 2022.
- [38] Andersson JAE, Gillis J, Horn G, Rawlings JB, Diehl M. CasADi: A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation* 2019;11. <https://doi.org/10.1007/s12532-018-0139-4>.



Mukhtar Sani (Member, IEEE) received B.S degree in electrical engineering from Kano University of Science and Technology, Nigeria, in 2015. He then received masters in systems, control and information technologies, and a PhD in real-time control of mobile robots from Université Grenoble Alpes in 2018 and 2022 respectively. His current research interest covers learning-based control of mobile robots, machine learning, fault detection and clean energy systems.



Ahmad Hably is senior Member, IEEE. He has received the B.S. degree in electrical engineering from the Lebanese University in 2004, and the M.S. and the Ph.D. degrees in automatic control from the Grenoble Institute of Technology, Grenoble, France, in 2005 and 2007, respectively. After one year as a Researcher with the LIRMM Laboratory he joined the Université Grenoble Alpes, Grenoble, as an Associate Professor with the Grenoble Image Parole Signal Automatique Laboratory (GIPSA-Lab). His research interests include electrical vehicle integration, nonlinear control, predictive control for energy systems, and airborne wind-energy systems.



Bogdan Robu is an Associate Professor at the University Grenoble Alpes, France and a researcher in the GIPSA-lab laboratory since 2012. Prior to this, he held a one-year teaching position at Toulouse University, France. He received his PhD in control engineering from Toulouse University in 2010. His research interests include the control of complex systems using classical control techniques (optimal, predictive, adaptive) as well as learning techniques (machine learning, artificial intelligence). He applies these techniques mostly to distributed computing systems (Cloud, IoT, High Performance Computing) as well as mechatronic systems and economics.



Jonathan Dumon has received the M.S. degree from Ecole Centrale de Nantes with a system, image and signal processing specialization in 2002. After 4 years in the manufacture of special machines for industry, he joined CNRS as instrument design engineer in 2006, in charge of aerial robotic platforms in the Grenoble Image Parole Signal Automatique Laboratory (GIPSA-Lab). His technical and research interests include mechatronics, robotics, wind turbines and airborne wind-energy systems.