Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 58-11 (2024) 165–170

A Hybrid Dynamical System Formulation of Capture-the-Flag Games

Santiago Jimenez Leudo * **Philipp Braun** **
Ricardo G. Sanfelice * **Iman Shames** **

* *Department of Electrical and Computer Engineering, University of California, Santa Cruz, USA, (e-mail: {sjimen28,ricardo}@ucsc.edu)*
** *School of Engineering, Australian National University, Canberra, Australia (e-mail: {philipp.braun,iman.shames}@anu.edu.au).*

Abstract: In this paper, we derive a comprehensive hybrid system representation for capture-the-flag games and a corresponding zero-sum game formulation. The hybrid system formulation and implementation is a first step towards the analysis and design of (sub)optimal control laws for the game. A scenario-based controller is employed to test the model in a simulation tool where the rules of the game are encoded.

Keywords: Multi-player games; hybrid systems; game theory

1. INTRODUCTION

Capture the flag is a rule-based game allegedly dating back to the book *"Scouting For Boys"*, by Baden-Powell (1908), where two (or more) teams of players compete against each other, trying to capture the opponents flag and return it to their own base. This game describes a rich family of sub-problems, where team members (representing robots and/or humans) cooperate with each other to maximize their profits, while teams compete to outperform each other. While humans and robots operate in continuous time, the capture-the-flag game is governed by discrete-time events at unknown and possibly periodic time instances, making it a *hybrid dynamical game*.

Classical competitive and non-cooperative dynamic games, such as chess and go, have seen a recent success in the software and algorithm development of AlphaZero and AlphaGo, as in Bertsekas (2022), for example. However, in chess and in go, the game takes place in a discrete-time horizon, which allows for solution concepts that employ tools from the discrete-time systems literature. Game theory for hybrid systems is discussed in Tomlin et al. (2000), in Altman et al. (2000), in (Lin and Antsaklis, 2022, Ch. 6.5), and in Jimenez Leudo and Sanfelice (2022a,b). Recent controller designs for capture-the-flag games can be found in Garcia et al. (2018), Huang et al. (2015), and in Wang et al. (2023), among others.

While decision-making processes are often relatively easy to describe informally, mathematically-precise formulations are generally difficult.The framework of hybrid systems theory in Goebel et al. (2012) and Sanfelice (2021) provides a useful level of versatility and formalism to study many modern hybrid games. Accordingly, a main contribution of this paper is the formulation of capture-the-flag games as hybrid dynamical systems. This work is motivated by the Aquaticus competition [1] and the preliminary work in Braun et al. (2023) discussing a heuristic controller design for capture-the-flag games without referring to a hybrid systems formulation.

The remainder of this paper is organized as follows. Section 2 briefly summarizes the rules of a capture-the-flag game. Section 3 provides the derivation of a hybrid system formulation and Section 4 discusses objective functions to define a zero-sum game. Section 5 presents a numerical example. By considering the dynamics of capture-the-flag games, this paper is meant to be a stepping stone to investigate the foundations of multi-player decision making in dynamical systems with composite discrete and continuous-time components using the said hybrid systems formalism.

Notation. The symbol $\mathbb{N}$ denotes the set of natural numbers including zero. The symbol $\mathbb{R}$ denotes the set of real numbers and $\mathbb{R}_{\geq 0}$ denotes its nonnegative subset. The Euclidean norm of a vector $x \in \mathbb{R}^n$ is denoted by $|x| = \sqrt{x^\top x}$. Given vectors x and y in $\mathbb{R}^n$, we write $(x, y) = [x^\top, y^\top]^\top$. Given a nonempty set $\mathcal{A} \subset \mathbb{R}^n$, the distance from x to $\mathcal{A}$ is defined as $|x|_{\mathcal{A}} = \inf_{y \in \mathcal{A}} |x - y|$. We denote with $\mathbb{1}_{\mathcal{A}} : \mathbb{R}^n \to \{0, 1\}$ the indicator function of the set $\mathcal{A}$. A ball of radius γ centered at $p \in \mathbb{R}^2$ is denoted by $\gamma\mathbb{B}_p = \{q \in \mathbb{R}^2 | \ |p - q| \leq \gamma\}$.

2. CAPTURE-THE-FLAG GAMES

In this section, we introduce the rules of the game before they are translated into a hybrid system formulation in Section 3. The Aquaticus competition[1] consists of two teams, a blue team (B) and a red team (R). The blue team has $b \in \mathbb{N}$ robots and the red team has $r \in \mathbb{N}$ robots.

⋆ Research partially supported by NSF Grants no. CNS-2039054 and CNS- 2111688, by AFOSR Grants nos. FA9550-19-1-0169, FA9550-20-1-0238, FA9550-23-1-0145, and FA9550-23-1-0313, by AFRL Grant nos. FA8651- 22-1-0017 and FA8651-23-1-0004, by ARO Grant no. W911NF-20-1-0253, and by DoD Grant no. W911NF-23-1-0158.

[1] Aquaticus competion: https://oceanai.mit.edu/aquaticus

10.1016/j.ifacol.2024.07.442

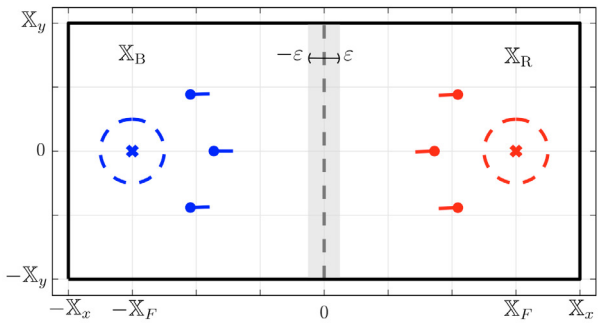


Fig. 1. Playing field of the Aquaticus competition with three blue and three red robots arbitrarily positioned.

The k-th robot in either team is modeled as the dynamical system

$$\dot{p}_k = \begin{bmatrix} \dot{p}_{k,1} \\ \dot{p}_{k,2} \end{bmatrix} = \begin{bmatrix} v_k \cos u_k \\ v_k \sin u_k \end{bmatrix} =: f(p_k, u_k), \tag{1}$$

where $p_k = (p_{k,1},\ p_{k,2}) \in \mathbb{R}^2$ is the position, $u_k \in \mathcal{U}_k \subseteq [-\pi, \pi]$ is an input representing the instantaneous heading angle, and v_k is the input velocity.

The k-th robot in team B is denoted k_B, with $k \in N_\mathrm{B} := \{1, 2, \ldots, b\}$, and the i-th robot in team R is denoted i_R, with $i \in N_\mathrm{R} := \{1, 2, \ldots, r\}$. The playing field[2] $\mathbb{X} := [-\mathbb{X}_x, \mathbb{X}_x] \times [-\mathbb{X}_y, \mathbb{X}_y] \subset \mathbb{R}^2$ is divided into the regions $\mathbb{X}_\mathrm{B} := [-\mathbb{X}_x, -\varepsilon] \times [-\mathbb{X}_y, \mathbb{X}_y]$, $\mathbb{X}_\mathrm{R} := [\varepsilon, \mathbb{X}_x] \times [-\mathbb{X}_y, \mathbb{X}_y]$, where $\varepsilon > 0$, and an arbitrarily small *neutral zone* $(-\varepsilon, \varepsilon) \times [-\mathbb{X}_y, \mathbb{X}_y]$. The neutral zone is introduced to ensure that $\mathbb{X}_\mathrm{B} \cap \mathbb{X}_\mathrm{R} = \emptyset$. This simplifies the presentation in the following by excluding situations that might occur on the zero-measure set defined by $\mathbb{X}_\mathrm{B} \cap \mathbb{X}_\mathrm{R}$. Each team has a flag that it protects from being captured by the opponent team. The flags' bases are located at $F_\mathrm{B} = (-\mathbb{X}_F, 0) \in \mathbb{X}$ and $F_\mathrm{R} = (\mathbb{X}_F, 0) \in \mathbb{X}$, where $\mathbb{X}_F > 0$ (see Figure 1).

The parameters of the game are the tagging radius $\gamma_c > 0$, the capturing radius $\gamma_F > 0$, and a timeout parameter $\bar{T} > 0$. Without loss of generality, considering the perspective of a robot $k_\mathrm{B} \in N_\mathrm{B}$ that competes with a robot $i_\mathrm{R} \in N_\mathrm{R}$, the rules of the game are as follows:

(a) *Tagging:* If k_B and i_R are in the blue region, namely, $p_{k_\mathrm{B}}, p_{i_\mathrm{R}} \in \mathbb{X}_\mathrm{B}$, and if $p_{i_\mathrm{R}} \in \gamma_c \mathbb{B}_{p_{k_\mathrm{B}}}$, then i_R is tagged by k_B, i.e., i_R is temporarily deactivated and the blue flag is instantaneously returned to F_B if i_R was carrying it.
(b) *Reactivation:* A robot i_R, which is temporarily deactivated by being tagged or by leaving the playing field, which corresponds to $p_{i_\mathrm{R}} \notin \mathbb{X}$, needs to satisfy $p_{i_\mathrm{R}} \in \gamma_F \mathbb{B}_{F_\mathrm{R}}$ to be reactivated. A robot leaving the playing field loses the flag if it was carrying it.
(c) *Disabled Tagging:* After tagging i_R, k_B loses its ability to tag another robot for time $\bar{T} > 0$.
(d) *Flag Capturing:* If k_B is not temporarily deactivated, and if $p_{k_\mathrm{B}} \in \gamma_F \mathbb{B}_{F_\mathrm{R}}$, then k_B grabs the red flag. Only blue robots can grab the red flag.
(e) Only one robot can carry the flag at a time.
(f) *Flag Return:* If k_B satisfies $p_{k_\mathrm{B}} \in \gamma_F \mathbb{B}_{F_\mathrm{B}}$ while carrying the red flag, then the blue team scores and the red flag is instantaneously returned to F_R.

[2] In Aquaticus, $\mathbb{X}_x := 80$m, $\mathbb{X}_y := 40$m, and $\mathbb{X}_F := 60$m.

The goal of team B is to capture the red flag and to successfully return it to its own base $\gamma_F \mathbb{B}_{F_\mathrm{B}}$ as many times as possible.

3. HYBRID SYSTEMS GAME FORMULATION

In this section, we formulate a hybrid system model of capture-the-flag games. Hybrid systems include states that evolve both continuously and discretely over time. Following the framework in Sanfelice (2021), a hybrid system is defined by its data (C, F, D, G), such that

- the continuous evolution of the state is governed by a differential equation defined by the *flow map* F and it occurs when in the *flow set* C; and
- the discrete evolution of the state is governed by a difference inclusion defined by the *jump map* G and it occurs when in the *jump set* D.

For the purpose of this paper, a hybrid dynamical system, which is denoted by $\mathcal{H}$, is given in terms of the hybrid inclusion with inputs (Sanfelice, 2021)

$$\mathcal{H} \begin{cases} \dot{x} &= F(x, u_\mathrm{B}, u_\mathrm{R}) \quad x \in C \\ x^+ &\in G(x) \quad\quad\ x \in D \end{cases} \tag{2}$$

where x is the state, and u_B and u_R are the inputs. We define a solution $(x(\cdot), (u_\mathrm{B}(\cdot), u_\mathrm{R}(\cdot)))$ to $\mathcal{H}$ from the initial condition x_0, as in (Sanfelice, 2021, Definition 2.29) and denote by $\mathcal{R}(x_0, (u_\mathrm{B}(\cdot), u_\mathrm{R}(\cdot)))$ the set of maximal state trajectories to $\mathcal{H}$ from x_0 for $(u_\mathrm{B}(\cdot), u_\mathrm{R}(\cdot))$.

3.1 Implementation of the Rules of the Game

From the description of the competition above, to encode the rules of the game given in (a)-(f), we propose a state vector that includes the position of the $k-$th robot, introduced in (1). For each $k_\mathrm{B} \in N_\mathrm{B}$ and each $i_\mathrm{R} \in N_\mathrm{R}$, the additional states with associated dynamics are described, without loss of generality, taking the perspective of an arbitrary robot $k_\mathrm{B} \in N_\mathrm{B}$, as follows. In addition to p_{k_B}, the state of the robot k_B has a timer τ_{k_B} to model its tagging ability and two logic variables, $(q_{k_\mathrm{B}}, \eta_{k_\mathrm{B}}) \in \{0, 1\}^2$ to model whether k_B is tagged and whether it carries the flag, respectively. The timer decreases according to $\dot{\tau}_{k_\mathrm{B}} = -1$, and takes values in $(-\infty, \bar{T}]$, where $\bar{T} > 0$ is as in rule (c). Thus, the state of k_B is

$$x_{k_\mathrm{B}} := (p_{k_\mathrm{B}}, \tau_{k_\mathrm{B}}, q_{k_\mathrm{B}}, \eta_{k_\mathrm{B}}) \in X_{k_\mathrm{B}} := \mathbb{R}^2 \times (-\infty, \bar{T}] \times \{0, 1\}^2. \tag{3}$$

We model the dynamics of k_B as a hybrid system, for which the flow and jump sets are defined to constrain the evolution of the state p_{k_B}, the timer τ_{k_B}, and the logic variables $(q_{k_\mathrm{B}}, \eta_{k_\mathrm{B}})$.

To model the tagging ability of k_B (item (c)), we define

$$\tau_{k_\mathrm{B}} \begin{cases} > 0: & \text{robot } k_\mathrm{B} \text{ does not have tagging ability,} \\ \leq 0: & \text{robot } k_\mathrm{B} \text{ has tagging ability.} \end{cases} \tag{4}$$

To model whether a robot is tagged or carrying a flag, the logic variables take the following values:

$$q_{k_\mathrm{B}} = \begin{cases} 0: & \text{robot } k_\mathrm{B} \text{ is active,} \\ 1: & \text{robot } k_\mathrm{B} \text{ is deactivated,} \end{cases} \tag{5}$$

$$\eta_{k_\mathrm{B}} = \begin{cases} 0: & \text{robot } k_\mathrm{B} \text{ does not carry the flag,} \\ 1: & \text{robot } k_\mathrm{B} \text{ carries the flag.} \end{cases} \tag{6}$$

The logic variables remain constant while the other states evolve in continuous time. In addition to the states introduced above for robot k_{B}, a flag state μ_{B} is introduced for each team, as follows:

$$\mu_{\mathrm{B}} = \begin{cases} 0 : \text{the blue flag is not at its base,} \\ 1 : \text{the blue flag is at its base.} \end{cases} \tag{7}$$

With these definitions, the rules outlined in Section 2 can be summarized through the following update laws.

The dynamics of the 'tagging ability' state: According to rule (c), a robot loses its tagging ability after tagging another robot. Hence, if

$$\exists i_{\mathrm{R}} \in N_{\mathrm{R}}, \ \exists k_{\mathrm{B}} \in N_{\mathrm{B}} \text{ s.t.} \begin{cases} p_{i_{\mathrm{R}}} \in \mathbb{X}_{\mathrm{B}}, \ q_{i_{\mathrm{R}}} = 0, \\ p_{i_{\mathrm{R}}} \in \gamma_C \mathbb{B}_{p_{k_{\mathrm{B}}}}, \tau_{k_{\mathrm{B}}} \leq 0, \\ p_{k_{\mathrm{B}}} \in \mathbb{X}_{\mathrm{B}}, \text{ and } q_{k_{\mathrm{B}}} = 0, \end{cases} \text{ then } \tau_{k_{\mathrm{B}}}^{+} = \bar{T}. \tag{8a}$$

Here, (8a) encodes that robot k_{B} is in the position to tag robot i_{R}, and thus, $\tau_{k_{\mathrm{B}}}$ is updated. By setting the flow of $\tau_{k_{\mathrm{B}}}$ as $\dot{\tau}_{k_{\mathrm{B}}} = -1$, we can ensure that a robot regains its tagging ability after $\bar{T}$ seconds by checking when $\tau_{k_{\mathrm{B}}} \leq 0$.

The dynamics of the 'tagged' state: To implement rule (a), we consider the following scenario and update law. If

$$\exists i_{\mathrm{R}} \in N_{\mathrm{R}}, \ \exists k_{\mathrm{B}} \in N_{\mathrm{B}} \text{ s.t.} \begin{cases} p_{i_{\mathrm{R}}} \in \mathbb{X}_{\mathrm{R}}, \ q_{i_{\mathrm{R}}} = 0, \\ p_{k_{\mathrm{B}}} \in \gamma_C \mathbb{B}_{p_{i_{\mathrm{R}}}}, \tau_{i_{\mathrm{R}}} \leq 0, \\ p_{k_{\mathrm{B}}} \in \mathbb{X}_{\mathrm{R}}, \text{ and } q_{k_{\mathrm{B}}} = 0 \end{cases} \text{ then } q_{k_{\mathrm{B}}}^{+} = 1 - q_{k_{\mathrm{B}}}. \tag{9a}$$

Here, (9a) reflects that robot i_{R} is in the position to tag robot k_{B}, and thus, $q_{k_{\mathrm{B}}}$ is updated to 1.

Two additional cases are considered for the update of $q_{k_{\mathrm{B}}}$. In particular,

$$\text{if } \{ p_{k_{\mathrm{B}}} \notin \mathbb{X} \text{ and } q_{k_{\mathrm{B}}} = 0, \quad \text{then} \quad q_{k_{\mathrm{B}}}^{+} = 1 - q_{k_{\mathrm{B}}}, \tag{9b}$$

and

$$\text{if } \{ p_{k_{\mathrm{B}}} \in \gamma_F \mathbb{B}_{F_{\mathrm{B}}} \text{ and } q_{k_{\mathrm{B}}} = 1, \quad \text{then} \quad q_{k_{\mathrm{B}}}^{+} = 1 - q_{k_{\mathrm{B}}}, \tag{9c}$$

cover rule (b), i.e., a robot leaving the playing field is deactivated and a deactivated robot returning to its flag base is reactivated. If none of these events occur, $q_{k_{\mathrm{B}}}$ remains constant and, thus, $\dot{q}_{k_{\mathrm{B}}} = 0$.

The dynamics of the 'carrying-the-flag' state: If a robot $k_{\mathrm{B}} \in N_{\mathrm{B}}$ carries the flag when it is tagged, then it loses the flag. This (see rule (a)) is encoded through the following update law. If

$$\exists i_{\mathrm{R}} \in N_{\mathrm{R}} \text{ s.t.} \begin{cases} p_{i_{\mathrm{R}}} \in \mathbb{X}_{\mathrm{R}}, \ q_{i_{\mathrm{R}}} = 0, \\ \tau_{i_{\mathrm{R}}} \leq 0, \ \mu_{\mathrm{R}} = 0, \\ p_{k_{\mathrm{B}}} \in \gamma_C \mathbb{B}_{p_{i_{\mathrm{R}}}}, \\ q_{k_{\mathrm{B}}} = 0, \text{ and } \eta_{k_{\mathrm{B}}} = 1, \end{cases} \text{ then } \eta_{k_{\mathrm{B}}}^{+} = 1 - \eta_{k_{\mathrm{B}}}. \tag{10a}$$

Similarly,

$$\text{if } \begin{cases} p_{k_{\mathrm{B}}} \notin \mathbb{X}, \ q_{k_{\mathrm{B}}} = 0, \\ \eta_{k_{\mathrm{B}}} = 1, \ \mu_{\mathrm{R}} = 0, \end{cases} \quad \text{then} \quad \eta_{k_{\mathrm{B}}}^{+} = 1 - \eta_{k_{\mathrm{B}}} \tag{10b}$$

encodes that $k_{\mathrm{B}} \in \mathbb{N}_{\mathrm{B}}$ loses the flag when leaving the playing field (see rule (b)). The update law

$$\text{if } \begin{cases} p_{k_{\mathrm{B}}} \in \gamma_F \mathbb{B}_{F_{\mathrm{R}}}, \\ q_{k_{\mathrm{B}}} = 0, \ \eta_{k_{\mathrm{B}}} = 0, \\ \mu_{\mathrm{R}} = 1, \end{cases} \quad \text{then} \quad \eta_{k_{\mathrm{B}}}^{+} = 1 - \eta_{k_{\mathrm{B}}} \tag{10c}$$

encodes that k_{B} captures the red team's flag (see rule (d)). If k_{B} returns the red flag to the blue flag base, then $\eta_{k_{\mathrm{B}}}$ is updated according to rule (f), which is

$$\text{if } \begin{cases} p_{k_{\mathrm{B}}} \in \gamma_F \mathbb{B}_{F_{\mathrm{B}}}, \\ q_{k_{\mathrm{B}}} = 0, \eta_{k_{\mathrm{B}}} = 1, \\ \mu_{\mathrm{R}} = 0 \end{cases} \quad \text{then} \quad \eta_{k_{\mathrm{B}}}^{+} = 1 - \eta_{k_{\mathrm{B}}}. \tag{10d}$$

A final rule is implemented to ensure that only one robot carries the flag at a time. According to (10c), more than one robot is able to update their η logic state at the same time, encoding that they all have "captured the flag." To address this, and ensure that only one blue robot is carrying the red flag according to rule (e), for all $k_{\mathrm{B}} \in N_{\mathrm{B}}$, the update law

$$\text{if } \left(\eta_{k_{\mathrm{B}}} \sum_{m_{\mathrm{B}}=1}^{k_{\mathrm{B}}} \eta_{m_{\mathrm{B}}} \right) \geq 2 \quad \text{then} \quad \eta_{k_{\mathrm{B}}}^{+} = 1 - \eta_{k_{\mathrm{B}}} \tag{10e}$$

ensures that only one robot (by convention the robot with the smallest index) has its η logic state equal to 1 after the update. As before, if none of these events occur, $\eta_{k_{\mathrm{B}}}$ remains constant and thus, $\dot{\eta}_{k_{\mathrm{B}}} = 0$.

The dynamics of the flag state: If a red robot grabs the blue flag according to rule (d), μ_{B} needs to be updated from 1 to 0. This is modeled by the following update. If

$$\exists i_{\mathrm{R}} \in N_{\mathrm{R}} \text{ s.t.} \begin{cases} q_{i_{\mathrm{R}}} = 0, \ \eta_{i_{\mathrm{R}}} = 0, \\ p_{i_{\mathrm{R}}} \in \gamma_F \mathbb{B}_{F_{\mathrm{B}}}, \\ \mu_{\mathrm{B}} = 1, \end{cases} \text{ then } \mu_{\mathrm{B}}^{+} = 1 - \mu_{\mathrm{B}}. \tag{11a}$$

Similarly, if the red flag is successfully carried to the blue flag base $\gamma_F \mathbb{B}_{F_{\mathrm{R}}}$ (see rule (f)), then, the flag is instantaneously returned to its base and μ_{B} is updated to 1, which is described by

$$\text{if } \exists i_{\mathrm{R}} \in N_{\mathrm{R}} \text{ s.t.} \begin{cases} q_{i_{\mathrm{R}}} = 0, \eta_{i_{\mathrm{R}}} = 1, \\ p_{i_{\mathrm{R}}} \in \gamma_F \mathbb{B}_{F_{\mathrm{R}}}, \\ \mu_{\mathrm{B}} = 0, \end{cases} \quad \text{then} \quad \mu_{\mathrm{B}}^{+} = 1 - \mu_{\mathrm{B}}. \tag{11b}$$

If the robot carrying the flag is deactivated by leaving the playing field, then the flag is instantaneously returned to its base, i.e.,

$$\text{if } \exists i_{\mathrm{R}} \in N_{\mathrm{R}} \text{ s.t.} \begin{cases} q_{i_{\mathrm{R}}} = 0, \eta_{i_{\mathrm{R}}} = 1, \\ p_{i_{\mathrm{R}}} \notin \mathbb{X}, \mu_{\mathrm{B}} = 0, \end{cases} \quad \text{then} \quad \mu_{\mathrm{B}}^{+} = 1 - \mu_{\mathrm{B}}, \tag{11c}$$

(see rule (b)). Similarly, if a red robot is tagged while carrying the blue flag, then the flag is instantaneously returned to its base (see rule (a)), as follows. If

$$\exists i_{\mathrm{R}} \in N_{\mathrm{R}}, \ \exists k_{\mathrm{B}} \in N_{\mathrm{B}} \text{ s.t.} \begin{cases} q_{i_{\mathrm{R}}} = 0, \eta_{i_{\mathrm{R}}} = 1, \\ p_{i_{\mathrm{R}}} \in \mathbb{X}_{\mathrm{B}}, \mu_{\mathrm{B}} = 0, \\ p_{i_{\mathrm{R}}} \in \gamma_C \mathbb{B}_{p_{k_{\mathrm{B}}}}, \\ q_{k_{\mathrm{B}}} = 0, p_{k_{\mathrm{B}}} \in \mathbb{X}_{\mathrm{B}}, \\ \tau_{k_{\mathrm{B}}} = 0, \end{cases} \text{ then } \mu_{\mathrm{B}}^{+} = 1 - \mu_{\mathrm{B}}. \tag{11d}$$

If none of these events occur, $\dot{\mu}_{\mathrm{B}} = 0$.

3.2 Flow and Jump Sets of the Hybrid Dynamical System

Using the scenario-based update laws in the previous section, we describe the game as a hybrid dynamical system $\mathcal{H}$. The state and input of the system are

$$\begin{aligned} x &= (x_{\mathrm{B}}, x_{\mathrm{R}}), & u &= (u_{\mathrm{B}}, u_{\mathrm{R}}), \\ x_{\mathrm{B}} &= (x_{1_{\mathrm{B}}}, \ldots, x_{b_{\mathrm{B}}}, \mu_{\mathrm{B}}), & u_{\mathrm{B}} &= (u_{1_{\mathrm{B}}}, \ldots, u_{b_{\mathrm{B}}}), \\ x_{\mathrm{R}} &= (x_{1_{\mathrm{R}}}, \ldots, x_{r_{\mathrm{R}}}, \mu_{\mathrm{R}}), & u_{\mathrm{R}} &= (u_{1_{\mathrm{R}}}, \ldots, u_{r_{\mathrm{R}}}), \end{aligned}$$

where each $x_{k_{\mathrm{B}}}$ and $x_{i_{\mathrm{R}}}$ is defined as in (3), and the state space and input space are defined as

$$\begin{aligned} X &= \left\{ x \,\middle|\, \begin{array}{l} x_{k_{\mathrm{B}}} \in X_{k_{\mathrm{B}}} \ \forall \ k_{\mathrm{B}} \in N_{\mathrm{B}} \\ x_{i_{\mathrm{R}}} \in X_{i_{\mathrm{R}}} \ \forall \ i_{\mathrm{R}} \in N_{\mathrm{R}} \\ \mu_{\mathrm{B}}, \mu_{\mathrm{R}} \in \{0, 1\} \end{array} \right\}, \\ U &= \left\{ u \,\middle|\, \begin{array}{l} u_{k_{\mathrm{B}}} \in \mathcal{U}_{k_{\mathrm{B}}} \ \forall \ k_{\mathrm{B}} \in N_{\mathrm{B}} \\ u_{i_{\mathrm{R}}} \in \mathcal{U}_{i_{\mathrm{R}}} \ \forall \ i_{\mathrm{R}} \in N_{\mathrm{R}} \end{array} \right\}, \end{aligned} \tag{12}$$

respectively, with $X_{k_{\mathrm{B}}}$ defined as in (3). The definitions of the flow map F, the jump map G, the flow set C, and the jump set D are presented next. With that aim, we introduce first the following case-based jump sets.

Define the set where robot $k_{\mathrm{B}} \in N_{\mathrm{B}}$ tags robot $i_{\mathrm{R}} \in N_{\mathrm{R}}$

$$D^{\mathrm{tag}}_{k_{\mathrm{B}},i_{\mathrm{R}}} := \left\{ x \in X \,\middle|\, \begin{array}{l} p_{k_{\mathrm{B}}} \in \mathbb{X}_{\mathrm{B}},\ q_{k_{\mathrm{B}}} = 0, \\ p_{i_{\mathrm{R}}} \in \mathbb{X}_{\mathrm{B}},\ q_{i_{\mathrm{R}}} = 0, \\ p_{i_{\mathrm{R}}} \in \gamma_C \mathbb{B}_{p_{k_{\mathrm{B}}}},\ \tau_{k_{\mathrm{B}}} \leq 0 \end{array} \right\}. \quad (13)$$

This set corresponds to the update (8a). In addition, to encompass the states where the red robot i_{R} is tagged while carrying the blue flag, we define

$$D^{\mathrm{tag}f}_{k_{\mathrm{B}},i_{\mathrm{R}}} := \left\{ x \in D^{\mathrm{tag}}_{k_{\mathrm{B}},i_{\mathrm{R}}} \,|\eta_{i_{\mathrm{R}}} = 1, \mu_{\mathrm{B}} = 0 \right\}, \quad (14)$$

encoding the set for the jump of μ_{B} in (11d). For any $i_{\mathrm{R}} \in N_{\mathrm{R}}$ and $k_{\mathrm{B}} \in N_{\mathrm{B}}$, the set $D^{\mathrm{tag}f}_{i_{\mathrm{R}},k_{\mathrm{B}}}$ is defined similarly.

Combining these definitions, denote

$$D^{\mathrm{tag}}_{k_{\mathrm{B}}} := \bigcup_{i_{\mathrm{R}} \in N_{\mathrm{R}}} D^{\mathrm{tag}}_{k_{\mathrm{B}},i_{\mathrm{R}}}, \quad (15) \qquad D^{\mathrm{tag}}_{\mathrm{B}} := \bigcup_{k_{\mathrm{B}} \in N_{\mathrm{B}}} D^{\mathrm{tag}}_{k_{\mathrm{B}}}, \quad (17)$$

$$D^{\mathrm{tag}f}_{k_{\mathrm{B}}} := \bigcup_{i_{\mathrm{R}} \in N_{\mathrm{R}}} D^{\mathrm{tag}f}_{k_{\mathrm{B}},i_{\mathrm{R}}}, \quad (16) \qquad D^{\mathrm{tag}f}_{\mathrm{B}} := \bigcup_{k_{\mathrm{B}} \in N_{\mathrm{B}}} D^{\mathrm{tag}f}_{k_{\mathrm{B}}}. \quad (18)$$

Here, (15) characterizes the set where the robot k_{B} can tag, and (17) characterizes the sets where the blue team can tag. Likewise, (16) and (18) are subsets of (15) and (17), respectively, where the tagged robot carries the flag.

Remark 1. Due to the definitions of $\mathbb{X}_{\mathrm{B}}$ and $\mathbb{X}_{\mathrm{R}}$, it is not possible that $k_{\mathrm{B}} \in N_{\mathrm{B}}$ and $i_{\mathrm{R}} \in N_{\mathrm{R}}$ tag each other at the same time since $\varepsilon > 0$ and $\mathbb{X}_{\mathrm{B}} \cap \mathbb{X}_{\mathrm{R}} = \emptyset$. ∘

A robot $k_{\mathrm{B}} \in N_{\mathrm{B}}$ captures the flag when in the set

$$D^{\mathrm{flag}}_{k_{\mathrm{B}}} := \{x \in X|\ q_{k_{\mathrm{B}}} = 0, \eta_{k_{\mathrm{B}}} = 0,\ p_{k_{\mathrm{B}}} \in \gamma_F \mathbb{B}_{F_{\mathrm{R}}}, \mu_{\mathrm{R}} = 1\},$$

which corresponds to the update in (10c). The set at which the blue team can capture the red flag is defined as the union of individual sets, namely,

$$D^{\mathrm{flag}}_{\mathrm{B}} := \bigcup_{k_{\mathrm{B}} \in N_{\mathrm{B}}} D^{\mathrm{flag}}_{k_{\mathrm{B}}}. \quad (19)$$

As modeled by (10e), multiple blue robots potentially could capture the red flag at the same time. To rule out having multiple blue robots carrying the red flag simultaneously, we define the sets

$$D^{\mathrm{flag},\mu}_{k_{\mathrm{B}}} := \left\{ x \in X \,\middle|\, \left(\eta_{k_{\mathrm{B}}} \sum_{m_{\mathrm{B}}=1}^{k_{\mathrm{B}}} \eta_{m_{\mathrm{B}}} \right) \geq 2 \right\}, D^{\mathrm{flag},\mu}_{\mathrm{B}} := \bigcup_{k_{\mathrm{B}} \in N_{\mathrm{B}}} D^{\mathrm{flag},\mu}_{k_{\mathrm{B}}} \quad (20)$$

which trigger a jump if multiple blue robots update their η state at the same time and ensure that only one has its η logic state equal to 1 at the same time.

The red flag has been successfully carried to the blue team's base if the state is in

$$D^{\mu}_{\mathrm{B}} := \bigcup_{k_{\mathrm{B}} \in N_{\mathrm{B}}} D^{\mu}_{k_{\mathrm{B}}}, \quad (21a)$$

where

$$D^{\mu}_{k_{\mathrm{B}}} := \{x \in X|\ p_{k_{\mathrm{B}}} \in \gamma_F \mathbb{B}_{F_{\mathrm{B}}}, \eta_{k\mathrm{B}} = 1, \mu_{\mathrm{R}} = 0, q_{k_{\mathrm{B}}} = 0\}. \quad (21b)$$

The corresponding jump is encoded by the update (10d), modeling that the blue robot no longer carries the red flag and the flag is instantaneously returned to the red base.

If a robot $k_{\mathrm{B}} \in N_{\mathrm{B}}$ is active and leaves the playing field, i.e., the state is in

$$D^{\mathbb{X}}_{k_{\mathrm{B}}} := \left\{ x \in X|\ p_{k_{\mathrm{B}}} \in \overline{\mathbb{R}^2 \backslash \mathbb{X}}, q_{k_{\mathrm{B}}} = 0 \right\}, \quad (22)$$

then, $q_{k_{\mathrm{B}}}$ is updated according to (9b). The closure of $\mathbb{R}^2 \backslash \mathbb{X}$ is used so that $D^{\mathbb{X}}_{k_{\mathrm{B}}}$ is closed.

If in addition, the corresponding robot is carrying the red flag when leaving the playing field, that is, the state is in

$$D^{\mathbb{X}f}_{k_{\mathrm{B}}} := \left\{ x \in D^{\mathbb{X}}_{k_{\mathrm{B}}} \,|\eta_{k_{\mathrm{B}}} = 1, \mu_{\mathrm{R}} = 0 \right\},$$

then the states $\eta_{k_{\mathrm{B}}}$ and $\mu_{k_{\mathrm{B}}}$ are updated according to (10b) and (11c), respectively. From the local jump sets, we define the sets at which B leaves the playing field as

$$D^{\mathbb{X}}_{\mathrm{B}} := \bigcup_{k_{\mathrm{B}} \in N_{\mathrm{B}}} D^{\mathbb{X}}_{k_{\mathrm{B}}}, \quad D^{\mathbb{X}f}_{\mathrm{B}} := \bigcup_{k_{\mathrm{B}} \in N_{\mathrm{B}}} D^{\mathbb{X}f}_{k_{\mathrm{B}}}, \quad (23)$$

If a robot k_{B} is tagged or has left the playing field, it needs to reach the set

$$D^{\mathrm{utag}}_{k_{\mathrm{B}}} := \{x \in X|\ q_{k_{\mathrm{B}}} = 1, p_{k_{\mathrm{B}}} \in \gamma_F \mathbb{B}_{F_{\mathrm{B}}}\} \quad (24)$$

to be reactivated, as encoded through (9c).

We define the set at which B regains its tagging ability as

$$D^{\mathrm{utag}}_{\mathrm{B}} := \bigcup_{k_{\mathrm{B}} \in N_{\mathrm{B}}} D^{\mathrm{utag}}_{k_{\mathrm{B}}}.$$

The sets above define the events that encode the rules of the game, which we use to define the jump set of $\mathcal{H}$ as

$$\begin{aligned} D := {} & D^{\mathrm{tag}}_{\mathrm{B}} \cup D^{\mathrm{tag}}_{\mathrm{R}} \cup D^{\mathrm{flag}}_{\mathrm{B}} \cup D^{\mathrm{flag}}_{\mathrm{R}} \cup D^{\mathrm{flag},\mu}_{\mathrm{B}} \cup D^{\mathrm{flag},\mu}_{\mathrm{R}} \\ & \cup D^{\mu}_{\mathrm{B}} \cup D^{\mu}_{\mathrm{R}} \cup D^{\mathbb{X}}_{\mathrm{B}} \cup D^{\mathbb{X}}_{\mathrm{R}} \cup D^{\mathrm{utag}}_{\mathrm{B}} \cup D^{\mathrm{utag}}_{\mathrm{R}}. \end{aligned} \quad (25)$$

Note that $D \subset X$ is closed since it is the union of closed sets. Correspondingly, the flow set is defined as

$$C := \overline{X \backslash D}. \quad (26)$$

The closure is used to ensure that C is closed, which is needed to guarantee that $\mathcal{H}$ is well-posed.

Remark 2. Note that the proposed modeling approach allows a robot $k_{\mathrm{B}} \in N_{\mathrm{B}}$ to tag multiple robots of R at the same time if the state is in multiple tagging jump sets simultaneously, e.g., $x \in D^{\mathrm{tag}}_{k_{\mathrm{B}},1_{\mathrm{R}}} \cap D^{\mathrm{tag}}_{k_{\mathrm{B}},2_{\mathrm{R}}}$. Likewise, a robot k_{B} can be tagged by multiple robots of R at the same time (if $x \in D^{\mathrm{tag}}_{1_{\mathrm{R}},k_{\mathrm{B}}} \cap D^{\mathrm{tag}}_{2_{\mathrm{R}},k_{\mathrm{B}}}$, for example), removing multiple tagging abilities for T seconds. ∘

3.3 Overall Hybrid System Model

Now, we use the expressions in Sections 3.1 and 3.2 to define a hybrid dynamical system $\mathcal{H}$ to model the capture-the-flag game. For each x in the flow set C as in (26) and each u in the set of inputs U as in (12), the state of a robot $k_{\mathrm{B}} \in N_{\mathrm{B}}$ evolves continuously according to

$$\dot{x}_{k_{\mathrm{B}}} = \begin{bmatrix} \dot{p}_{k_{\mathrm{B}}} \\ \dot{\tau}_{k_{\mathrm{B}}} \\ \dot{q}_{k_{\mathrm{B}}} \\ \dot{\eta}_{k_{\mathrm{B}}} \end{bmatrix} = F_{k_{\mathrm{B}}}(x_{k_{\mathrm{B}}}, u_{k_{\mathrm{B}}}) := \begin{bmatrix} f(p_{k_{\mathrm{B}}}, u_{k_{\mathrm{B}}}) \\ -1 \\ 0 \\ 0 \end{bmatrix},$$

and the flag state evolves according to $\dot{\mu}_{\mathrm{B}} = 0$. The maps $F_{i_{\mathrm{R}}}(x_{i_{\mathrm{R}}}, u_{i_{\mathrm{R}}})$, $i_{\mathrm{R}} \in N_{\mathrm{R}}$, are defined similarly.

Based on the robots individual flow maps, the game evolves continuously for all $x \in C$ according to

$$\begin{aligned} \dot{x} = {} & F(x, u_{\mathrm{B}}, u_{\mathrm{R}}) \\ & := (F_{1_{\mathrm{B}}}(x_{1_{\mathrm{B}}}, u_{1_{\mathrm{B}}}), \ldots, F_{b_{\mathrm{B}}}(x_{b_{\mathrm{B}}}, u_{b_{\mathrm{B}}}), 0, \\ & \quad F_{1_{\mathrm{R}}}(x_{1_{\mathrm{R}}}, u_{1_{\mathrm{R}}}), \ldots, F_{r_{\mathrm{R}}}(x_{r_{\mathrm{R}}}, u_{r_{\mathrm{R}}}), 0). \end{aligned} \quad (27)$$

The definition of the jump map is more complicated since it requires the union of case-based individual jump maps. Again, we focus on the derivation from the perspective

of B, while the definitions for R follow analogously. For $x \in D$, $k_{\mathrm{B}} \in N_{\mathrm{B}}$, we consider the following definitions. First, notice that the position of a robot does not change at a jump, i.e., $p_{k_{\mathrm{B}}}^{+} = p_{k_{\mathrm{B}}}$ for each $x \in D$. For the remaining state variables recall the set $D_{k_{\mathrm{B}}}^{\mathrm{tag}}$ in (15), where a robot k_{B} tags a robot i_{R}, and define the sets

$$D_{k_{\mathrm{B}}}^{(1)} := D_{k_{\mathrm{B}}}^{\mathrm{tag}},$$

$$D_{k_{\mathrm{B}}}^{(2)} := \overline{\left(\left(\bigcup_{i_{\mathrm{R}} \in N_{\mathrm{R}}} D_{i_{\mathrm{R}},k_{\mathrm{B}}}^{\mathrm{tag}}\right) \cup D_{k_{\mathrm{B}}}^{\mathbb{X}} \cup D_{k_{\mathrm{B}}}^{\mathrm{utag}}\right) \setminus D_{k_{\mathrm{B}}}^{(1)}},$$

$$D_{k_{\mathrm{B}}}^{(3)} := \overline{\left(\left(\bigcup_{i_{\mathrm{R}} \in N_{\mathrm{R}}} D_{i_{\mathrm{R}},k_{\mathrm{B}}}^{\mathrm{tag}f}\right) \cup D_{k_{\mathrm{B}}}^{\mathrm{flag}} \cup D_{k_{\mathrm{B}}}^{\mathrm{flag},\mu} \cup D_{k_{\mathrm{B}}}^{\mu} \cup D_{k_{\mathrm{B}}}^{\mathbb{X}f}\right) \setminus D_{k_{\mathrm{B}}}^{(1)} \cup D_{k_{\mathrm{B}}}^{(2)}},$$

$$D_{\mu_{\mathrm{R}}} := D_{\mathrm{R}}^{\mathrm{tag}f} \cup D_{\mathrm{B}}^{\mathrm{flag}} \cup D_{\mathrm{B}}^{\mu} \cup D_{\mathrm{B}}^{\mathbb{X}f},$$

where a jump is triggered to update the variables $\tau_{k_{\mathrm{B}}}$, $q_{k_{\mathrm{B}}}$, $\eta_{k_{\mathrm{B}}}$, and μ_{R}. This construction gives a sequential priority to the jumps that occur on $D_{k_{\mathrm{B}}1}$, then on $D_{k_{\mathrm{B}}2}$, and lastly on $D_{k_{\mathrm{B}}3}$. The definition of $D_{k_{\mathrm{B}}2}$ follows from the local jump sets (13), (22), and (24). The justification for the form of $D_{k_{\mathrm{B}}3}$ stems from (16), (19), (20), (21), and (23). The definition of $D_{\mu_{\mathrm{R}}}$ is consistent with the definition of the sets (18), (19), (21), and (23).

Based on these sets, we define the corresponding local jump maps $\hat{g}_{k_{\mathrm{B}},m} : D_{k_{\mathrm{B}}}^{(m)} \to X$, $m \in \{1,2,3\}$, where

$$\begin{aligned}
\hat{g}_{k_{\mathrm{B}},1}(x) &:= (z_{k_{\mathrm{B}}}, \bar{T}, q_{k_{\mathrm{B}}}, \eta_{k_{\mathrm{B}}}) && \text{if } x \in D_{k_{\mathrm{B}}1},\\
\hat{g}_{k_{\mathrm{B}},2}(x) &:= (z_{k_{\mathrm{B}}}, \tau_{k_{\mathrm{B}}}, 1 - q_{k_{\mathrm{B}}}, \eta_{k_{\mathrm{B}}}) && \text{if } x \in D_{k_{\mathrm{B}}2},\\
\hat{g}_{k_{\mathrm{B}},3}(x) &:= (z_{k_{\mathrm{B}}}, \tau_{k_{\mathrm{B}}}, q_{k_{\mathrm{B}}}, 1 - \eta_{k_{\mathrm{B}}}) && \text{if } x \in D_{k_{\mathrm{B}}3},\\
\hat{g}_{\mu_{\mathrm{R}}}(x) &:= 1 - \mu_{\mathrm{R}} && \text{if } x \in D_{\mu_{\mathrm{R}}}.
\end{aligned}$$

Depending on the local jump maps, the corresponding states are updated.

Simplifying the jump sets above by eliminating the disjoint sets via a scenario-based analysis, we define the sets

$$\begin{aligned}
\Delta_{k_{\mathrm{B}}}^{\tau} &:= D_{k_{\mathrm{B}}}^{\mathrm{tag}},\\
\Delta_{k_{\mathrm{B}}}^{q} &:= \overline{(D_{k_{\mathrm{B}}}^{\mathbb{X}} \setminus D_{k_{\mathrm{B}}}^{\mathrm{tag}}) \cup D_{k_{\mathrm{B}}}^{\mathrm{utag}}},\\
\Delta_{k_{\mathrm{B}}}^{\eta} &:= \overline{\left(D_{k_{\mathrm{B}}}^{\mathrm{flag}} \setminus \bigcup_{i_{\mathrm{R}} \in N_{\mathrm{R}}} D_{i_{\mathrm{R}},k_{\mathrm{B}}}^{\mathrm{tag}f}\right) \cup (D_{k_{\mathrm{B}}}^{\mu} \setminus D_{k_{\mathrm{B}}}^{\mathrm{tag}})},\\
\Delta_{k_{\mathrm{B}}}^{\mu} &:= \overline{D_{k_{\mathrm{B}}}^{\mathrm{flag},\mu} \setminus \bigcup_{i_{\mathrm{R}} \in N_{\mathrm{R}}} D_{i_{\mathrm{R}},k_{\mathrm{B}}}^{\mathrm{tag}}}.
\end{aligned}$$

To construct the jump map of the tagging ability states, we define, for $k_{\mathrm{B}} \in N_{\mathrm{B}}, i_{\mathrm{R}} \in N_{\mathrm{R}}$, the maps

$$\hat{G}_{k_{\mathrm{B}},i_{\mathrm{R}}}^{\tau}(x) := \begin{smallmatrix}(x_{1_{\mathrm{B}}},\ldots,\hat{g}_{k_{\mathrm{B}},1}(x),\ldots,x_{b_{\mathrm{B}}},\mu_{\mathrm{B}},\\ x_{1_{\mathrm{R}}},\ldots,\hat{g}_{i_{\mathrm{R}},2}(x),\ldots,x_{r_{\mathrm{R}}},\mu_{\mathrm{R}})\end{smallmatrix} \quad \text{if } x \in D_{k_{\mathrm{B}},i_{\mathrm{R}}}^{\mathrm{tag}}.$$

Putting the individual jump maps together, we define

$$\hat{G}_{k_{\mathrm{B}},\tau}(x) := \bigcup_{i_{\mathrm{R}} \in N_{\mathrm{R}}} \hat{G}_{k_{\mathrm{B}},i_{\mathrm{R}}}^{\tau}(x) \quad \text{if } x \in \Delta_{k_{\mathrm{B}}}^{\tau}.$$

Consider the jump map of the tagged states for which we define, for $k_{\mathrm{B}} \in N_{\mathrm{B}}$, the mappings

$$\hat{G}_{k_{\mathrm{B}},q}(x) := \begin{smallmatrix}(x_{1_{\mathrm{B}}},\ldots,\hat{g}_{k_{\mathrm{B}},2}(x),\ldots,x_{b_{\mathrm{B}}},\mu_{\mathrm{B}},\\ x_{1_{\mathrm{R}}},\ldots,x_{r_{\mathrm{R}}},\mu_{\mathrm{R}})\end{smallmatrix} \quad \text{if } x \in \Delta_{k_{\mathrm{B}}}^{q}.$$

Consider the jump map of the carrying-the-flag states for which we define, for $k_{\mathrm{B}} \in N_{\mathrm{B}}$, the mappings

$$\hat{G}_{k_{\mathrm{B}},\eta}(x) := \begin{smallmatrix}(x_{1_{\mathrm{B}}},\ldots,\hat{g}_{k_{\mathrm{B}},3}(x),\ldots,x_{b_{\mathrm{B}}},\mu_{\mathrm{B}},\\ x_{1_{\mathrm{R}}},\ldots,x_{r_{\mathrm{R}}},\hat{g}_{\mu_{\mathrm{R}}}(x))\end{smallmatrix} \quad \text{if } x \in \Delta_{k_{\mathrm{B}}}^{\eta},$$

and for the case of multiple robots capturing the flag, the mappings

$$\hat{G}_{k_{\mathrm{B}},\mu}(x) := \begin{smallmatrix}(x_{1_{\mathrm{B}}},\ldots,\hat{g}_{k_{\mathrm{B}},3}(x),\ldots,x_{b_{\mathrm{B}}},\mu_{\mathrm{B}},\\ x_{1_{\mathrm{R}}},\ldots,x_{r_{\mathrm{R}}},\mu_{\mathrm{R}})\end{smallmatrix} \quad \text{if } x \in \Delta_{k_{\mathrm{B}}}^{\mu}.$$

The constructions above lead to the jump map

$$G(x) := \left\{ \hat{G}_{s_\star,z}(x) \;\middle|\; \begin{array}{l} x \in \Delta_{s_\star}^{z}, \star \in \{\mathrm{B},\mathrm{R}\},\\ s \in N_\star, z \in \{\tau,q,\eta,\mu\} \end{array} \right\}. \tag{28}$$

The discrete evolution of the game is governed by

$$x^{+} \in G(x), \quad x \in D. \tag{29}$$

The overall game is modeled by $\mathcal{H}$ given in (25)–(29).

Before we introduce the objective function of the game in the next section, we highlight some important properties of the hybrid system.

Lemma 1. Consider the hybrid system defined in (25)–(29). Suppose f is continuous. Then, $\mathcal{H}$ satisfies the hybrid basic conditions (Sanfelice, 2021, Def. 2.20), that is, F is continuous, $G : D \rightrightarrows X$ in (28) is outer semicontinuous and locally bounded, and C and D are closed. ⌟

Following (Sanfelice, 2021, Definition 2.18), we say that a system $\mathcal{H}$ that satisfies the hybrid basic conditions is well posed. Well posedness of hybrid closed-loop systems guarantees key structural properties of solutions.

Lemma 2. Consider the hybrid system defined in (25)–(29). For each given input $t \mapsto (u_{\mathrm{B}}(t), u_{\mathrm{R}}(t))$ with domain $[0,\infty)$, and for each initial condition $x_0 \in C \cup D$, there exists a maximal solution (Sanfelice, 2021, Definition 2.29) to $\mathcal{H}$ that is complete and its domain is unbounded in the ordinary time variable. ⌟

4. ZERO-SUM FORMULATION

This section discusses the objective of the game that was described in Section 2 and it proposes a meaningful objective function to define a zero-sum game.

Based on the definition of D_{B}^{μ} and D_{R}^{μ} in (21), the overall goal of capturing the opponent's flag and returning it to the team's own base is encoded by the objective function

$$\begin{aligned}
\mathcal{J}_g(x_0,(u_{\mathrm{R}}(\cdot),u_{\mathrm{B}}(\cdot))) := \sup_{x(\cdot)\in\mathcal{R}(x_0,(u_{\mathrm{R}}(\cdot),u_{\mathrm{B}}(\cdot)))}\\
\sum_{(t_{j+1},j)\in\operatorname{dom} x(\cdot)} \left(\mathbb{1}_{D_{\mathrm{B}}^{\mu}}(x(t_{j+1},j)) - \mathbb{1}_{D_{\mathrm{R}}^{\mu}}(x(t_{j+1},j))\right)
\end{aligned} \tag{30}$$

where $\{t_j\}_{j=0}^{\sup_j \operatorname{dom} x}$ is a nondecreasing sequence associated to the definition of the hybrid time domain of $(x(\cdot),(u_{\mathrm{R}}(\cdot),u_{\mathrm{B}}(\cdot)))$ as in Sanfelice (2021) and $\mathcal{R}(x_0,u)$ is the set of maximal state trajectories to $\mathcal{H}$ from x_0 for $(u_{\mathrm{R}}(\cdot),u_{\mathrm{B}}(\cdot))$, as defined in Section 3. The cost $\mathcal{J}_g$ is defined as the worst-case cost over all solutions from x_0.

Based on this definition, the value function is given by

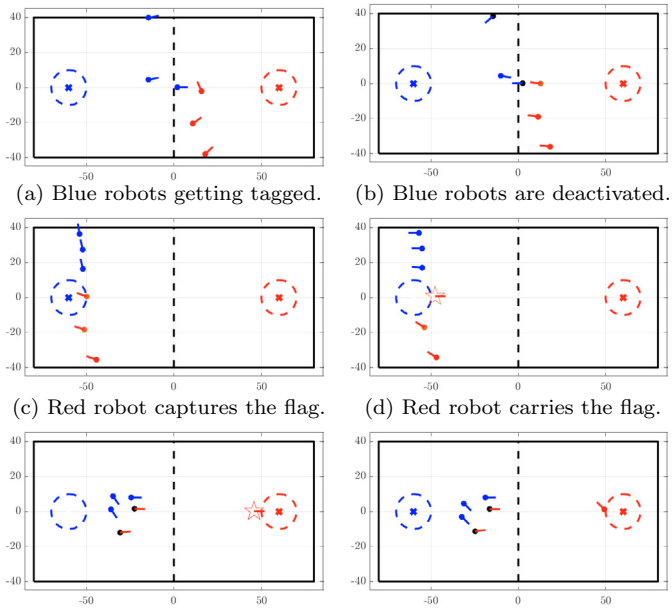
$$\begin{aligned}
\mathcal{J}_g^{\star}(x_0) &= \inf_{u_{\mathrm{R}}(\cdot)} \sup_{u_{\mathrm{B}}(\cdot)} \mathcal{J}_{\mathrm{g}}(x_0,(u_{\mathrm{R}}(\cdot),u_{\mathrm{B}}(\cdot))) && (31)\\
&= \sup_{u_{\mathrm{B}}(\cdot)} \inf_{u_{\mathrm{R}}(\cdot)} \mathcal{J}_{\mathrm{g}}(x_0,(u_{\mathrm{R}}(\cdot),u_{\mathrm{B}}(\cdot))) && (32)
\end{aligned}$$

where team B aims to maximize the cost while team R seeks to minimize it. If $\mathcal{J}_{\mathrm{g}}^{\star}(x_0) > 0$, team B wins, and if $\mathcal{J}_{\mathrm{g}}^{\star}(x_0) < 0$, team R wins.

Even though the cost function in (30) encodes the objective of each team, the synthesis of optimal control laws is an open problem of research.

5. NUMERICAL SIMULATIONS

Consider a scenario with three robots in each team, initialized as in Figure 1. A simulation tool has been developed [3] where the rules of the game are encoded by implementing the hybrid model in Section 3. This tool allows one to test different controllers. Some of the robots have been endowed with a switched controller following the derivations in Garcia et al. (2018).



(a) Blue robots getting tagged. (b) Blue robots are deactivated.
(c) Red robot captures the flag. (d) Red robot carries the flag.
(e) Red robot dropping the flag. (f) Flag back at blue home.

Fig. 2. Simulation of hybrid model as in Section 3. Each row depicts an update on the state due to a jump.

In Figure 2(a), a blue robot is about to be tagged and another blue robot is about to exit the playing field. Thus, $x \in D^{\text{tag}}_{i_{\text{R}},k_{\text{B}}} \cap D^{\mathbb{X}}_{k_{\text{B}}}$ as in (13) and (22), which trigger the state to jump according to (9). When the state $q_{k_{\text{B}}}$ is updated, denoting a robot has been deactivated, its marker color changes to black as in (b) and the controller leads them to head back to the base according to the law

$$\kappa_{k_{\text{B}}}(x) := \cos^{-1}\left(\frac{-\mathbb{X}_F - p_{k_{\text{B}},1}}{|F_{\text{B}} - p_{k_{\text{B}}}|}\right). \tag{33}$$

In addition, a robot that has a tagging timeout changes its color to green or orange, depending on whether it belongs to B or R, respectively. In (c), a red robot is about to capture the blue flag, so $x \in D^{\text{flag}}_{i_{\text{R}}}$ as in (19), which triggers a jump in the state according to (10c) and (11a). When the states $\eta_{i_{\text{R}}}$ and μ_{B} are updated, denoting that a red robot has captured the blue flag, its marker changes to a star as in (d), the marker denoting the position of the flag at its base turns white, and the controller leads the red robot to head back to its base according to the attack strategy design in Phase II in Garcia et al. (2018). In (e), the red robot carrying the flag arrives to its base, namely, $x \in D^{\mu}_{\text{R}}$ as in (21), which triggers a jump in the state according to (10d) and (11b). When the states $\eta_{i_{\text{R}}}$ and μ_{B} are updated, denoting the red team has scored because a red robot returns the blue flag to the red base, its marker changes back to a circle as in (f), the marker denoting the position of the base turns blue, and the controller $\kappa_{i_{\text{R}}}(x) := \pi$ leads the red robot to cross the midfield.

[3] Code at https://github.com/sjleudo/HybridCaptureTheFlag

6. DISCUSSION AND CONCLUDING REMARKS

This paper proposes a hybrid system formulation with a zero-sum hybrid games framework to describe exhaustively capture-the-flag games. While synthesizing a controller in an optimal fashion represents an unsolved challenge, a simulation tool is developed to implement the game model with a preliminary controller design. Future work will focus on suboptimal controller designs with performance guarantees using model predictive control approaches. This will be further explored following Braun et al. (2023).

REFERENCES

Altman, E., Başar, T., and Pan, Z. (2000). Piecewise-deterministic differential games and dynamic teams with hybrid controls. In *Advances in Dynamic Games and Applications*, 223–256. Springer.

Baden-Powell, R. (1908). *Scouting for Boys*. C. Arthur Pearson Ltd.

Bertsekas, D. (2022). *Lessons from AlphaZero for Optimal, Model Predictive, and Adaptive Control*. Athena Scientific.

Braun, P., Shames, I., Hubczenko, D., Dostovalova, A., and Fraser, B. (2023). Capture the flag games: Observations from the 2022 Aquaticus competition. *IFAC-PapersOnLine*, 56(2), 11363–11368. 22nd IFAC World Congress.

Garcia, E., Casbeer, D.W., and Pachter, M. (2018). The capture-the-flag differential game. In *IEEE Conference on Decision and Control*, 4167–4172.

Goebel, R., Sanfelice, R.G., and Teel, A.R. (2012). *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press.

Huang, H., Ding, J., Zhang, W., and Tomlin, C.J. (2015). Automation-assisted capture-the-flag: A differential game approach. *IEEE Transactions on Control Systems Technology*, 23(3), 1014–1028.

Jimenez Leudo, S. and Sanfelice, R.G. (2022a). Sufficient conditions for optimality and asymptotic stability in two-player zero-sum hybrid games. *25th ACM International Conference on Hybrid Systems: Computation and Control*, 1–11.

Jimenez Leudo, S. and Sanfelice, R.G. (2022b). Sufficient conditions for optimality in finite-horizon two-player zero-sum hybrid games. 3268–3273.

Lin, H. and Antsaklis, P.J. (2022). *Hybrid Dynamical Systems: Fundamentals and Methods*. Springer.

Sanfelice, R.G. (2021). *Hybrid Feedback Control*. Princeton University Press.

Tomlin, C., Lygeros, J., and Sastry, S. (2000). A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88(7), 949–970.

Wang, J., Zhou, Z., Jin, X., Mao, S., and Tang, Y. (2023). Matching-based capture-the-flag games for multi-agent systems. *IEEE Transactions on Cognitive and Developmental Systems*.