Automatic Generation of Ice Hockey Defensive Motion via Coverage Control and Control Barrier Functions

Kornvik Tanpipat, Takeshi Hatanaka, and Masatoshi Hiroura

Abstract A successful defensive strategy in ice hockey games is often designed empirically by an experienced professional. The majority of previous work on automating the strategy focuses on analyzing spatial data to decide the most optimal formation and action but cannot generalize the system to real games with real-time capabilities. We propose a novel control logic for generating real-time ice hockey defensive motion based on a control barrier function (CBF) and coverage control to extend our antecessors’ logic that succeeds in duplicating ideal formations for specific scenes. To this end, we first present an ellipsoidal CBF to overcome the drawbacks of the existing line-based CBF of our antecessors. We also tune and add a novel density function to reflect real specifications more precisely than the previous work. The control logic is then demonstrated through simulations with offensive motion in real games. It is confirmed that the present logic generates valid defensive movements without specification to these specific scenes. We further exemplify that the logic generates proper motion under ice hockey’s man-to-man and zone defense strategies and their intermediate strategies by tuning the logic. This would contribute to reducing the efforts of the practitioners to educate ice hockey players.

1 INTRODUCTION

Ice hockey is a sport in which players score goals in a six-on-six game on a field. Each team tries to achieve a goal by sending a puck, a black disk used in ice hockey, to the opponent’s goal. Currently, the defensive formation against the opposing team’s attack is determined empirically, and it is unclear whether it is indeed the optimal formation or not. Therefore, we seek to automate the creation of the defensive formation, utilizing the fact that the goal of the ice hockey defensive formation is to cover a large area while placing some defenders around the attacking players. Such action is similar to the objective of the coverage control problem, in which multiple robots form a network and move simultaneously while cooperating. Thus, we attempt to automate the simulation of ice hockey formations based on coverage control. Our predecessors have approached the problem by proposing the above idea and succeeded in imitating defensive strategies for empirically designed specific scenes [4]. They also simulate pass cut and goal block movements in these scenes by utilizing line-based control barrier function (CBF), which is a tool used to achieve a control’s objective subject to safety guarantees [5], [6]. However, their approaches are specific to some scenes. Moreover, Their line-based CBF is also not fully able to imitate pass cuts and goal blocks movements.

In this paper, we go beyond imitation to specific scenes. We propose a system to generate defensive formation for ice hockey in general scenes. We overcome the drawbacks of the previous logic regarding line-based CBF with a novel CBF model and also further improve the logic based on [1] and [2]. We also confirm that the novel system enables the generation of strategies in general scenes by inputting data from real scenes.

2 PRELIMINARY

In the following, we introduce the fundamentals of coverage control and control barrier functions for our simulation. These explanations are the same as mentioned in [1].

2.1 Coverage Control

The coverage control is a multi-agent system control method that governs agents to move toward the optimal placement based on a predetermined density function. Many applications of coverage control have been reported in many fields, such as [4].

2.1.1 System model

We consider the movement of n agents or defensive players \( V = \{1, \ldots, n\} \) on the 2-dimensional convex polyhedron field \( Q \subseteq \mathbb{R}^2 \). We also consider the defensive player model as the equation below

\[
\dot{x}_i(t) = u_i(t), \quad x_i(0) = x_{0i}
\]

where \( x_i = [x_{xi}, x_{yi}]^T \) is the position of the \( i^{th} \) defensive player and \( u_i = [u_{xi}, u_{yi}]^T \) is the input of the \( i^{th} \) player. Each area in the field also has its weight, defined by the weight function \( \phi(q) \).

Next, we divide the field area \( Q \) by the defensive players \( x_i \) into several Voronoi cells \( C_i \) according to Eq. [2]

\[
C_i(x) = \{ q \in Q : ||q - x_i||^2 \leq ||q - x_j||^2, \forall j \in V \} \quad (2)
\]

This implies that for any area \( q \in C_i(x) \), the area is subjected to the cell, corresponding to the closest defensive player. For example, the visualization of the Voronoi cells can be seen Fig. [1].
In Eq. 8,
\[
\mathbf{k} \text{ set. Thus, with the above controller, we can optimize}
\]
the equilibrium state as Eq. 8.

As the objective of coverage control is to optimize the coverage of the agents over the specified area. For example, from Fig. 1 and given that the weight function is some constant number over all the area, the eventual result of this setup is as shown in Fig. 2.

2.1.2 Evaluation function and control input

The evaluation function for the coverage control is defined as
\[
J(x, t) = \sum_{i=1}^{n} \int_{V_i(x)} ||q - x_i||^2 \phi(q) \, dq
\]  \hspace{1cm} (3)
where \( \phi(q) \) is the weight function of position \( q \). This evaluation equation quantifies how “well” the agents cover the area given the predetermined weight function.

With Eq. 3, we calculate the equilibrium state of the agents corresponding to the evaluation function by solving its derivatives as shown below.
\[
\frac{\partial J}{\partial x} |_{x=x^*} = 0
\]  \hspace{1cm} (4)
By solving the above equation, we can derive the equilibrium state which is
\[
x^*_i = \int_{V_i(x)} q\phi(q) dq \quad \int_{V_i(x)} \phi(q) dq
\]  \hspace{1cm} (5)
By considering the Voronoi cell \( C_i(x) \) as a rigid body, we use Eq. 6 to describe its properties.
\[
\text{mass}(C_i(x(t))) = \int_{C_i(x)} \phi(q) dq \quad \text{(6)}
\]
\[
\text{cent}(C_i(x(t))) = \frac{1}{\text{mass}(C_i(x(t)))} \int_{C_i(x)} q\phi(q) dq \quad \text{(7)}
\]
We can consider Eq. 6 as the cell’s mass representation and Eq. 7 as the cell’s center of mass representation. Consequently, the interpretation shows that we can rewrite and consider the equilibrium state of the agents as the center of mass of each Voronoi cells which is
\[
x^*_i = \text{cent}(C_i(x(t)))
\]
As the objective of coverage control is to optimize the coverage of the agents over the area, we can design a controller that drives the agents toward the equilibrium state as Eq. 8
\[
u_i(t) = -k(x_i(t) - \text{cent}(C_i(x(t))))
\]  \hspace{1cm} (8)
In Eq. 8, \( k \) is a constant gain that the designer can set. Thus, with the above controller, we can optimize the coverage of the agents over the specified area. For example, from Fig. 1 and given that the weight function is some constant number over all the area, the eventual result of this setup is as shown in Fig. 2.

2.2 Control Barrier Function (CBF)

As stated in [1], coverage control is necessary for the system to simulate goal block and pass cut movements. In this section, we introduce the fundamentals of CBF used in our simulation.

Considering the previously stated system dynamics from Eq. 11
\[
\dot{x} = f(x)
\]  \hspace{1cm} (9)
given \( x \in D \subseteq \mathbb{R}^m \) as the system state and \( f(x) \) as continuous Lipschitz function. Next, suppose we have a set \( C \) that satisfy the following equations
\[
C = \{ x \in \mathbb{R}^n : h(x) \geq 0 \}, \quad \text{(10)}
\]
\[
\partial C = \{ x \in \mathbb{R}^n : h(x) = 0 \}, \quad \text{(11)}
\]
\[
\text{Int}(C) = \{ x \in \mathbb{R}^n : h(x) > 0 \} \quad \text{(12)}
\]
given that \( h : \mathbb{R}^m \rightarrow \mathbb{R} \) is a continuously differentiable function, we recall the following theorems due to [1].

Theorem 1: From the system in Eq. 9 and set \( C \) in Eq. 10-12 if there exists class \( K \) function \( \alpha \), which satisfies the equation below, then \( h \) function is a control barrier function over \( C \subseteq D \subseteq \mathbb{R}^m \)
\[
\sup[L_fh(x) + \alpha(h(x))] \geq 0, \forall x \in D
\]  \hspace{1cm} (13)
As stated in [1], we also let \( C = D \) in this paper. Next, we recall the following theorem due to [9].

Theorem 2: For all time \( t \) and all initial state \( x_0 \in C \), if the state \( x(t, x_0) \) is in set \( C \) from Eq. 10-12, then set \( C \) is forward invariant.

Next, we define the set \( K_{cbf} \) as below,
\[
K_{cbf}(x) = \{ u \in U : L_fh(x) + \alpha(h(x)) \geq 0 \}
\]  \hspace{1cm} (14)
Thus, considering the continuous differentiable function $h$ and considering the set $C$ defined by Eq. 10.12, if the function $h$ is a control barrier function on the set $C$, according to Theorem 2, for any Lipschitz continuous input $u(x) \in K_{cbf}(x)$ which maps $C$ to $U$, set $C$ is a forward invariant.

In conclusion, using $u$ from Eq. 14 will guarantee that the state will not deviate from set $C$.

As we will apply the control barrier function on top of coverage control, which will produce $u_{nom}$, we can solve Eq. 15-16 to attain minimum change of $u$ that satisfy the constraint in the control barrier function.

$$u = \text{argmin}_u ||u - u_{nom}||^2$$

$$s.t. \quad Lf h(x) + \alpha(h(x)) \geq 0$$

4 CONTROL LOGIC FOR DEFENSIVE MOTION GENERATION

In the following, we describe the specifications for the control logic qualitatively. The detailed implementation of these specifications will be discussed in the next chapter.

4.1 Weight function design

We first recall the motivation to use coverage control for defense formation. Coverage control is used to make each agent, a defensive player, cover areas on the field, prioritizing significant regions. We can assign the significance of the area by designing the weight function. An area with a higher weight value implies higher importance. In our case, the defensive players should block the offensive players and try to protect the area in front of the goal.

For blocking the offensive players, defensive players should stay between offensive players and the goal while also moving according to the offensive players’ movement. Accordingly, we would design the weight function to be high where the defensive players should perform blocking. We will also consider the goal as another offensive player to make the defensive player move to protect the goal.

For covering the significant regions, the area closer to the goal should be more significant than the one further away. Moreover, the area in front of the goal should also be significantly important. Specifically, defensive players should protect the yellow area, which is shown in Fig. 4. According to [2], this area contributes to 75% of goals scored in a real game.

Next, the defensive players should pay more attention to the puck holder, which is the player who holds the puck. When the puck holder moves closer to the goal, we can also partially ignore the vertically opposite side of the field, omitting the goal’s weight on the vertical-opposite side but still paying attention to offensive players in the area.

Lastly, the middle area on the right side of the blue line empirically should also have lower importance. We can visualize that area in Fig. 5.

4.2 CBF design

The control barrier function is used in this problem to simulate goal block and pass cut movements ac-
In our simulation, we choose to focus only on performing a pass cut, which is the action of a defensive player to move in between two offensive players. In the previous CBF model in [1], a line-based CBF is proposed for realizing pass cut. However, this model does not guarantee to simulate such action.

### 4.2.1 Drawbacks of line-based CBF

In the previous model from [1], an orthogonal distance toward a line, running through two offensive players’ positions, is used as a model for a control barrier function for simulating a pass cut. In this case, CBF is non-negative when the defensive player is no further than $\delta$ from the line, as shown in Eq. 18.

$$h_{\text{cut}}^i = \delta^2 - \frac{(aX_x - X_y + b)^2}{a^2 + 1} \quad (18)$$

where $[X_x, X_y]^T$ is the position of the defensive player. Let other variables in Eq. 18 are defined as

$$a = \frac{X_{y1} - X_{y2}}{X_{x1} - X_{x2}} \quad (19)$$

$$b = \frac{X_{x1}X_{y2} - X_{x2}X_{y1}}{X_{x1} - X_{x2}} \quad (20)$$

where $[X_{xk}, X_{yk}]^T$ is the position of the $k^{th}$ offensive player for $k \in 1, 2$.

There are two problems in the previous model. First, Eq. 19 and 20 are inacalculable when $X_{x1} = X_{x2}$. Second, this control barrier function doesn’t guarantee that the defensive player moves between two offensive players as shown in Fig. 7.

Fig. 6 demonstrates the incorrect movement as the blue star dot, representing defensive player, does not move in between two red dots, representing static offensive players. This happens because the line, defined from two offensive players, stretches beyond the positions of two.

### 4.2.2 Ellipsoidal CBF

We propose to overcome this problem by introducing the ellipsoidal model as a control barrier function. The motivation of using the ellipsoidal model is to force the defensive players to move inside the ellipsoid, defined by the position of two offensive players, as shown in Fig. 8.

$$P_{wi} = P_i + \frac{P_{\text{goal}} - P_i}{|P_{\text{goal}} - P_i|} + \dot{P}_i \quad (21)$$

where $P_i$ is $i^{th}$ offensive players’ positions and $P_{\text{goal}}$ is the position of the goal which is $[6, 15]^T$. After determining the position to apply the Gaussian distribution, we use the following covariance matrix as a specification for applying the distribution to each
Fig. 8: Example of ellipsoidal control barrier function position defined above.

$$\Sigma W_i = \begin{pmatrix} 15 & 0 \\ 0 & 15 \end{pmatrix}$$ (22)

In conclusion, we have n+1 weight distributions on the field which are $W_1, W_2, ..., W_n$ and $W_{goal}$.

Once applied the distribution as a weight to each position stated above, the result can be seen in Fig. 9, given that the red dots are offensive players.

Fig. 9: Weight assigned for each offensive player and at the goal position

Next, we apply distance gain to each of the weights representing an offensive player, prioritizing the area closer to the goal with a higher weight value. We design the distance gain to have a higher value when applying to the position closer to the goal to achieve this specification. The updated weights are

$$W_i^*(x,y) = W_i^*(x,y) \times \frac{(25 - |P_{goal} - [x,y]^T|)}{25}, \forall i \in 1, 2, ..., n, \forall [x,y]^T \in Q$$ (23)

where $W_i^*(x,y)$ is the updated weight from the $i^{th}$ weight distribution at the position $(x,y)$ on the field $Q$. Fig. 10 demonstrates the result of the updated weights.

Fig. 10: Weight from Fig.9 when applying distance gain

Furthermore, another area in the field which defensive players should protect is the house area, as stated in the previous chapter. Therefore, we also imitate the house area and apply constant gain to raise the weights in the area. We design the house area as the inner area bounded by Eq. 24-26.

$$y_1 = -x_1 + 20$$ (24)
$$y_2 = x_2 + 10$$ (25)
$$(x_2 - 5)^2 + (y_2 - 15)^2 = 225$$ (26)

In this paper, we denote this bounded area as $Y$. We apply a constant gain $p$ to the weight inside this area. Furthermore, we also remove the goal weight outside the house zone. Using the above specification, we define the resulting weight as

$$W_{i,*}^*(x,y) = W_i^*(x,y) \times p, \forall (x,y) \in Y$$ (27)
$$W_{goal,*}^*(x,y) = \begin{cases} W_{goal,*}^*(x,y) \times p \times 1.5 & \forall (x,y) \in Y \\ 0 & \text{otherwise} \end{cases}$$

where $Y$ is the inner area described by Eq. 24-26 and $p$ is the gain constant. By setting $p = 2$, the resulting weight can be seen in Fig. 11.

Fig. 11: Weight from Fig.10 when applying gain on house area ($p = 2$)

To prioritize the puck holder when the puck holder moves close to the goal area, we selectively apply a constant gain to lower the significance of other offensive players on the field as shown in Alg. 1. We denote the puck holder’s index as $pu \in \{1, 2, ..., n\}$ and the x and y position of the puck holder as $P_{pu,x}$ and $P_{pu,y}$ respectively.

Given that the orange dot represents the puck holder, the result of the above weights can be visualized as Fig. 12.

Empirically, the green area in Fig. 3 is also less important than other areas. As a result, we apply a constant gain to lower this specific area. We can define the updated weight by applying the following equation

$$W_{i,**}^*(x,y) = \begin{cases} W_{i,**}^*(x,y) \times 0.1 & \forall x \in [14, 9.45, 21, 9.60] \\ W_{i,**}^*(x,y) & \text{and} \forall y \in [10, 20] \end{cases}$$ (28)

Fig. 12: Weight from Fig.10 when applying gain on house area ($p = 2$)
Algorithm 1 Weight adjustment algorithm

Result $W_{i}^{***}(x, y) \forall i \in \{1, 2, \ldots, n, \text{goal}\}$, a set of weights.

1: if $P_{pu,x} \leq 10$ and $P_{pu,y} \leq 15$ then
2: for $i \leftarrow 1$ to $n$ do
3: if $i \neq J$ then
4: $W_{i}^{**}(x, y) = 0.1 \cdot W_{i}^{**}(x, y)$ ∀$y < 15$
5: $W_{i}^{**}(x, y) = 0.05 \cdot W_{i}^{**}(x, y)$ ∀$y \geq 15$
6: end if
7: end for
8: $W_{\text{goal}}(x, y) = W_{\text{goal}}(x, y)$ ∀$y < 15$
9: $W_{\text{goal}}(x, y) = 0$ ∀$y \geq 15$ or $x \geq 10$
10: else if $P_{pu,x} \leq 10$ then
11: for $i \leftarrow 1$ to $n$ do
12: if $i \neq J$ then
13: $W_{i}^{**}(x, y) = 0.1 \cdot W_{i}^{**}(x, y)$ ∀$y \geq 15$
14: $W_{i}^{**}(x, y) = 0.05 \cdot W_{i}^{**}(x, y)$ ∀$y < 15$
15: end if
16: end for
17: $W_{\text{goal}}(x, y) = W_{\text{goal}}(x, y)$ ∀$y > 15$
18: $W_{\text{goal}}(x, y) = 0$ ∀$y \leq 15$ or $x \geq 10$
19: end if

Fig. 12: Weight from Fig.11 when applying constant gain.

$W_{i}^{***}(x, y) = W_{i}^{**}(x, y)$

for $i \in \{1, 2, \ldots, n\} \backslash pu$. In other words, this specification only applies to non-puck holder offensive players. Lastly, as we focus only on the left-hand side of the field, we omit all the weight on the right side of the field.

$W_{i}^{final}(x, y) = \begin{cases} 0 & \text{∀} x > 30 \\ W_{i}^{***}(x, y) & \text{otherwise} \end{cases} \forall i \in \{1, 2, \ldots, n, \text{goal}\}$

(29)

This is the finalized weights that we will use in the simulation.

5.2 Ellipsoidal CBF implementation

As stated in section 5.1, we design the new CBF as the equation below.

$h(X, X_{1}, X_{2}) = 1 - (X - X_{\text{goal}})^{T}P(X - X_{\text{goal}})$

(30)

given that $X = [x, y]^{T} \in Q$ is the position of the defensive player, $X_{\text{goal}} = [x_{\text{goal}}, y_{\text{goal}}]^{T} \in Q$ is the center of the ellipse formed by having the position of offensive players $X_{1,2}$ as vertices and the height and the semi-minor axes $d$, and $X_{k} = [x_{k}, y_{k}]^{T}$ is the position of the $k^{th}$ offensive player for $k \in 1,2$.

Furthermore, the parameters in Eq. (31) can be defined as the following.

$X_{\text{goal}} = \frac{X_{1} + X_{2}}{2}, l = ||X_{2} - X_{1}||$

(32)

where $l$ is the width of the semi-major axes and $\theta$ is defined as below

$\theta = \arctan\left(\frac{y_{2} - y_{1}}{x_{2} - x_{1}}\right)$,

(33)

$\cos(\theta) = \frac{x_{2} - x_{1}}{l}$,

(34)

$\sin(\theta) = \frac{y_{2} - y_{1}}{l}$

(35)

With the above ellipsoidal model, $h$ will be non-negative only when the defensive player moves inside the ellipse, between 2 offensive players.

As, we define the new ellipsoidal model, we rewrite the resulting constraint as Eq. (30) by applying it to Eq. (17)

$\dot{h}(X, X_{1}, X_{2}) = \frac{\partial h}{\partial X} \dot{X} + \frac{\partial h}{\partial X_{1}} \dot{X}_{1} + \frac{\partial h}{\partial X_{2}} \dot{X}_{2}$

(36)

$\geq -\alpha h(X, X_{1}, X_{2})$

The derivative of $h$ can be expanded into the following equation.

$\dot{h}(X, X_{1}, X_{2}) = a(\dot{x}_{1} + \dot{x}_{2} - 2\dot{x})$

$+ b(\dot{x}_{1} - \dot{x}_{2})$

$+ c(\dot{y}_{1} + \dot{y}_{2} - 2\dot{y})$

$+ d(\dot{y}_{2} - \dot{y}_{1})$

(37)

where $a, b, c$ and $d$ are

$\alpha = \left[\begin{array}{c}
\cos^{2}(\theta) + \sin^{2}(\theta) \frac{1}{l^{2}} - \frac{1}{d^{2}} \\
\frac{\cos^{2}(\theta)}{l^{2}} + \frac{\sin^{2}(\theta)}{d^{2}} \end{array}\right] (x - x_{0})$

$+ \cos\theta\sin\theta \left[\begin{array}{c}
\frac{1}{l^{2}} - \frac{1}{d^{2}} \end{array}\right] (y - y_{0})$

(38)
Pass cut is an action of blocking the player from passing the puck to other players. Alg. 2 is designed based on this motivation. Specifically, we aim to apply CBF when there is a possibility that offensive players, who are inside the house area, can receive the puck and subsequently score a goal. Therefore, we design an algorithm to only select defensive players inside the house area to perform a pass cut and prioritise block offensive players closer to the goal than those further away.

Algorithm 2 Online selection CBF algorithm

Result $P_{passcut}$, a set of pairs of defensive player indexes and offensive player indexes to perform pass cut.

1: Identify offensive players except puck holder in the house area as $O_{house}$ and defensive players in the house area as $D_{house}$.
2: Let $P_{passcut} = \emptyset$
3: while $O_{house} \neq \emptyset$ and $D_{house} \neq \emptyset$ do
4:    Select $i \in O_{house} = \text{argmin} ||x_i - x_{goal}||$ O,
5:    identify $d_i \in D = \text{argmax} h(P_d, P_i, P_{puck})$
6:    Remove $d_i$ from $D_{house}$ and $i$ from $O_{house}$
7: end while

From Alg. 2, we compute the pairs of offensive players that need to be blocked and the corresponding defensive players to perform pass cut. In the next section, we will apply the ellipsoidal CBF to each pair in the set $P_{passcut}$, preventing offensive players in the house area from receiving the puck from the puck holder.

5.4 Comprehensive algorithm

In the following, we describe the comprehensive algorithm in Alg. 3 which is used to calculate $u_i$ for each offensive player by applying the theorem, specifications, and algorithms stated above.

After computing the input for each defensive player $u_i$ from Alg. 3, we input the control inputs to the system and update the position of defensive players. Then, we do this process again with the new offensive player’s positions in the following frame until the simulation ends.

6 SIMULATION STUDY

In this chapter, we explain the method to attain the data from real scenes in this paper. Then, we describe the evaluation method that we will use to evaluate the simulation results. Lastly, we show the results of the simulation using the real scenes’ data.

6.1 Data extraction from real scene

For the data, we use ice hockey game scenes from Winter Olympic 2018. We selected two defensive
**Algorithm 3** Comprehensive algorithm

**Result** $u$, a set of inputs for defensive players $\{1, 2, ..., 5\}$

1: Initialize position of defensive players $i \in D = \{1, 2, ..., 5\}$ as $X_{di} \in Q$
2: Let the position of offensive player $j \in O = \{1, 2, ..., n\}$ as $X_{oj} \in Q$
3: Let the weight function as $\phi(x, y)$ with the design and implementation stated in section 5.1 using $X_{oj}$
4: Compute Voronoi cell $C_i$ using $X_{di}$ and $\phi(x, y)$ in Eq. 2
5: Compute cent($C_i$) using $C_i$ and $\phi(x, y)$ in Eq. 7 for $\forall i \in D$
6: Compute $u_i^{nom}$ using cent($C_i$) and $\phi(x, y)$ in Eq. 8 with $k = 1 \forall i \in D$
7: Let $P_{passcut}$ is the output of Algorithm 2.
8: for $i \leftarrow 1$ to 5 do
9: if $i$ is the first index of pair $p \in P_{passcut}$ then
10: Let $k =$ second index of the pair $p$
11: Compute $u_i$ by applying the ellipsoidal CBF with $x_i$ and the corresponding $x_k$ using quadratic programming to solve Eq. 16 subject to Eq. 17
12: else
13: $u_i = u_i^{nom}$
14: end if
15: if $||u_i|| > 3$ then
16: $u_i = ||u_i|| + 3$
17: end if
18: end for

scenes in which we can track offensive players almost all the time to attain the most accurate input for the simulation. The scenes are chosen among several scenes proposed by Takahito Suzuki, a retired head coach of the Japanese national team with who we cooperate within this research. The scenes used in this simulation are as the following:

- Canada and Czech republic’s game
- Canada and Finland’s game

For extracting the data from the scene, we have tried techniques according to [8]. However, we discovered two problems using these techniques. First, we are not able accurately track players’ poses using the pose estimation tool due to frequent collisions between players. Second, the court parsing process can not track the template images, needed to extract camera movement from these particular scenes because the number of the template images in these specific scenes is not high enough. As a result, we decide to extract the data manually. We extract the data frame by frame using the human eye to observe the position and movement of the players on the scenes. The example of the extracted data from Canada and Finland’s game scene can be seen in Fig. 13.

![Fig. 13: Example of extracted data from real scenes](image)

From figure 13, each coloured line represents one offensive player’s motion throughout the scene.

### 6.2 Evaluation Method

After surveying many papers and journals related to the evaluation of ice hockey defensive strategy, we found that there are no evaluation metric which can assess how well the generated formation is. The current evaluation metrics are based on statistical data such as [10] and [11]. Thus, we decide to use empirical method to evaluate our result by collaborating with Takahito Suzuki to evaluate the generated formations.

### 6.3 Results

This section shows results from simulating the defensive formation using the real data previously mentioned in this chapter. We use blue dots to represent defensive players, an orange dot to represent a puck holder, and red dots to represent other offensive players. This section also shows the simulation results using different $p$ values defined in Eq. 27.

The screenshots of the simulation from Canada and Czech republic’s game using $p = 2$ can be seen in Fig. 14-16.

![Fig. 14: The screenshots of the simulation from Canada and Czech republic’s game 1](image)

![Fig. 15: The screenshots of the simulation from Canada and Czech republic’s game 2](image)
Both simulations demonstrate the defensive movement of defensive players defending the area in front of the goal while also moving along with offensive players. These results are also being evaluated by the professional, as mentioned in section 6.2. He assesses that the generated movements of the defensive players are closed to the actual players. Thus, we can conclude that we achieve the generation of defensive formation for the scenes.

Furthermore, the results from the simulation using Canada and Finland’s game with different $p$ values, which are 0.5, 3 and 10 from top to bottom respectively, as shown in Fig. 20-22. Fig. 20-22 show that different $p$ value gives different results. With a lower $p$ value, the importance of the house area is relatively low, and defensive players will move close to each offensive player instead. A low $p$ value leads to movements that are close to a peer-to-peer defensive strategy in the real game. On the other hand, when $p$ grows bigger, the weight on the house area becomes higher, increasing the area’s significance. Consequently, defensive players exhibit defensive behavior that resembles zone defense strategy in the real game, where defensive players move to cover the important area rather than the players. We can also observe the intermediate strategies when we set $p = 3$.

### 7 CONCLUSION

In conclusion, we further develop the control logic from [2] and [1] by changing the weight function’s design in coverage control and improving the CBF model to an ellipsoidal CBF. Furthermore, we confirm that we can generate the defensive formation of ice hockey players using a new logic for two general scenes without adjusting the weight function specifically for the scene. We also discover that we can produce different strategies by changing the weight ratio parameter $p$, which affects the ratio between the players’ weights and the area’s weight in the house area. By adjusting $p$, we can generate several strategies, such as peer-to-
peer strategy when $p$ is small or zone defense strategy when $p$ is large. We can also observe the intermediate strategies by tuning parameter $p$ to be between the two strategies.

8 DISCUSSION

In this section, we discuss the problems that can be further improved for this paper.

1. We need to test the simulation with more scenes. In this paper, we have only tested the system with two general scenes. It is necessary to investigate the new logic further with several more scenes to confirm our assumption.

2. We need to automate the extraction of data from the videos. In this paper, we extract the input manually from the video. The method can be inaccurate as a human estimate the data. Developing a technique to automate this process is necessary to gain accurate input for testing the logic.

3. We need to take into consideration of the puck position in the future simulation. In the current simulation, we do not take into account the puck position. Although we assign the offensive player who holds the puck as a puck holder, it is undeniable that there exists a time when the puck is not with any players, such as when the puck holder sends the puck to the other player. Thus, the generated movement might not accurately reflect the actual game movement.

4. We need to create an evaluation metric to standardize our evaluation. In this experiment, we asked the professional to evaluate our experiment’s result. With ample experience in the field, there is some credibility to his evaluation. However, it is undeniable that there is a need to create a systematic evaluation metric for evaluating defensive movements.

References

[1] D. Matsumoto: Imitation and Control Logic Design of Defensive Formation of Ice Hockey Based on Coverage Control and Control Barrier Functions, Tokyo Institute of Technology 2021

[2] R. Ota : 被覆制御に基づくアイスホッケーの陣形シミュレーション, 東京工業大学工学院システム制御系特定課題研究報告書 2020

[3] 永原, 東 (編): マルチエージェントシステムの制御, コロナ社, 2015

[4] H. Dan, J. Yamauchi, T. Hatanaka, and M. Fujita: Control Barrier Function-Based Persistent Cover-
age with Performance Guarantee and Application to Object Search Scenario, Proc. 4th IEEE Conference on Control Technology and Applications, pp. 640-647, 2020

[5] A. D. Ames, J. W. Grizzle, and P. Tabuada: Control Barrier Function Based Quadratic Programs with Application to Adaptive Cruise Control, Proc. 53rd IEEE Conference on Decision and Control, pp. 6271–6278, 2014

[6] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada: Control Barrier Function Based Quadratic Programs for Safety Critical Systems, IEEE Transactions on Automatic Control, vol. 62, no. 8, pp. 3861–3876, 2017.

[7] —: Safety-critical Control for Dynamical Bipedal Walking with Precise Footstep Placement, Proc. IFAC on Analysis and Design of Hybrid Systems (ADHS), vol. 48, issue 27, pp. 147-154, 2015.

[8] N. Johnson: Extracting Player Tracking Data from Video Using Non-Stationary Cameras and a Combination of Computer Vision Techniques, MIT Sloan Conference 2020

[9] L. Wang, A.D. Ames and M. Egerstedt: Safety Barrier Certificates for Collisions-free Multirobot Systems, Proc. IEEE Transactions on Robotics, vol. 33, no. 3, pp. 661-674, 2017.

[10] Micheal E. S.: Statistical Evaluation of Ice Hockey Goaltending, Handbook of Statistical Methods and Analyses in Sports, Chapman and Hall/CRC, 2016

[11] Liu G. and Schulte O.: Deep Reinforcement Learning in Ice Hockey for Context-Aware Player Evaluation, Simon Fraser University 2018.