Connecting Run-Time Metrics to Outcome Performance of Team Attack and Defense

Jason Blum

CMU-RI-TR-16-24

May 2016

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Paul Scerri, Chair
Manuela Veloso
Michael Shomin

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Keywords: multi-agent teams, attack and defense, performance metrics

Abstract

Large underactuated ships have been shown to be vulnerable to attacks from smaller, faster boats due to their lack of maneuverability. This scenario can be addressed by a team of small "satellite" defensive boats designed to intercept and inspect approaching boats. Movements of defensive agents could be exploited by feinting attackers, drawing aggressive defenders out of an advantageous position as they move to intercept. Moving to defend one region may create vulnerabilities in others. Intelligent strategies are required to perform well in this adversarial environment, and developing these defensive schemes requires effective evaluation of their performance. Unfortunately, defensive strategies are typically evaluated solely by final outcome statistics, which do not carry any information about why a particular outcome was reached.

This work seeks to connect outcome performance with run-time metrics that describe the state of a team attack and defense scenario. The teams in this scenario represent the small boats attacking a large ship and the satellite defensive boats used to intercept them. To this end a novel family of run-time metrics, called Time-To-Arrive (TTA), is introduced. TTA can be used by both attack and defense to delegate tasks amongst agents in a team and execute strategies. TTA can be calculated offline to allow for a low run-time complexity. TTA data can be collected for fixed points around the defensive area and manipulated to describe the state of the defense as a whole.

Several thousand simulation runs with various combinations of attack and defense strategies, in addition to other varying parameters, were completed and compared. Attackers that exploit TTA information were shown to outperform random attackers. Dynamic defense provides robustness against such exploitation for defenders that can match their opponents' speed, but this turns into a disadvantage for slower defenders.

Lower values of TTA at the defensive perimeter showed a strong correlation with higher defensive winning percentage. Lower values of volatility, a scalar measure of how rapidly TTA values changed, also showed a strong correlation with higher defensive winning percentage. These two run-time metrics show a connection to outcome performance, and as such, could serve as a basis for future work as a run-time optimization objective function.

Acknowledgments

Committee Members: Paul Scerri, Manuela Veloso, Michael Shomin

Platypus

James Edmondson

Chris Tomaszewski

Contents

1 Introduction **1**

1.1 Related Work 3

1.1.1 Offensive and Defensive Tactics 3

1.1.2 Patrol, Security, and Surveillance 4

1.1.3 Measures of Effectiveness in a Combat Scenario 4

1.1.4 Role/Task Assignment 6

1.1.5 Reward Shaping 6

1.1.6 Predator-Prey 7

2 Methods **9**

2.1 Lutra Platform 9

2.2 Simulation 10

2.2.1 Dynamic Model 11

2.2.2 Control 12

2.2.3 Simulated Scenario 14

2.2.4 Time-To-Arrive 16

2.2.5 Offline Time-To-Arrive Calculation and Model 19

2.3 Offensive and Defensive Strategies 21

2.3.1 Offensive 22

2.3.2 Defensive 23

2.4 Minimum Time-To-Arrive Smoothness 28

2.4.1 Volatility 28

2.4.2 Circularity 29

3 Results **31**

3.1 Defender Performance and Metric Histograms 31

3.1.1 Metric Histogram Legend 32
3.1.2 Fast Defender, Outcome Performance 32
3.1.3 Fast Defender, Final Attack Maximum minTTA 33
3.1.4 Fast Defender, Volatility 33
3.1.5 Fast Defender, Circularity 35
3.1.6 Fast Defender, Final Attack Circularity 35
3.1.7 Slow Defender, Outcome Performance 36
3.1.8 Slow Defender, Final Attack Maximum minTTA 37
3.1.9 Slow Defender, Volatility 37
3.1.10 Slow Defender, Circularity 38
3.1.11 Slow Defender, Final Attack Circularity 38
3.1.12 Defender Count 40

4 Conclusions **41**
4.1 Discussion 41
4.2 Future Work 42

Bibliography **43**

List of Figures

2.1 The Lutra platform 10
2.2 Individual boat state geometry 11
2.3 Line-of-sight geometry for a straight line and smooth curve 13
2.4 Example screenshot from the simulation scenario 14
2.5 Example attacker removal event 15
2.6 A successful attack 16
2.7 Body frame Time-To-Arrive Contours, 5 and 10 seconds 17
2.8 Fixed radii Time-To-Arrive geometry 18
2.9 Polar Grid for Offline Time-To-Arrive Calculation 19
2.10 Offline least-squares TTA model error 21
2.11 Feinting attacker 23
2.12 Initial state of static defense 24
2.13 Initial state of turned defense 25
2.14 Initial state of dynamic defense 26
2.15 Interception and Recovery 27

3.1 Metric Histogram Legend 32
3.2 Defense winning % for fast defenders 33
3.3 Fast Defender, Final Attack Maximum minTTA 34
3.4 Fast Defender, Volatility 34
3.5 Fast Defender, Circularity 35
3.6 Fast Defender, Final Attack Circularity 36
3.7 Defense Winning % for Slow Defenders 36
3.8 Slow Defender, Final Attack Maximum minTTA 37
3.9 Slow Defender, Volatility 38
3.10 Slow Defender, Circularity 39
3.11 Slow Defender, Final Attack Circularity 39

List of Tables

2.1 Least-squares linear fit coefficients for TTA 20
2.2 Six primary combinations of offense and defense strategies 21

3.1 Simulation parameters 31

Chapter 1

Introduction

Large, underactuated ships are a mainstay in commercial and military fleets. High capacity vessels carry large loads over long ranges. They utilize a long, thin hull to maximize their forward speed and efficiency by minimizing drag. One drawback of this elongated form is a larger minimum turning radius. The combination of an underactuated design, high mass due to their size and cargo, and an elongated hull leads to long acceleration periods and large minimum turning radii. Put simply, large ships have severly limited mobility, even in the open water. When space becomes cramped (e.g. inside of a harbor or a canal), this is exacerbated. It can become such a challenge that an entire class of supporting boats, called the tugboat, is used to assist larger ships' movements in the tight confines of a harbor.

This limited mobility can lead to vulnerability to attacks by smaller boats. The smaller boats have a tighter turning radius and much shorter acceleration periods, allowing them to approach the large ship from several angles before the larger ship can maneuver in response. In recent years, these attacks often began with the smaller boats masquerading as harmless commerical boats [6]. For example, boats used for fishing are so ubiquitous that they do not draw suspicion even as they approach another ship. Small boat traffic is also higher in the areas with more limited space to maneuver a large ship, such as harbors, coastal areas, and canals.

For military vessels, this vulnerability is usually offset by advanced long range weapons - the large ship doesn't need to maneuver to defend itself. Application of deadly force may be viable when an attacking boat is clearly hostile, but one cannot assume that every approaching boat is truly hostile. A non-lethal solution to determine if a nearby boat is not an actual threat is required. One possible solution is to have a team of small "satellite" boats escort the larger ship. These defensive boats would intercept approaching boats to inspect them and defend against any real threats that they find. Using autonomous satellite boats would reduce the danger to human operators.

The ability of a defensive agent to physically intercept a possible attacker is highly dependent on their relative maneuverability. A slow boat with a large turning radius would have difficulty catching a fast, nimble target. Each defender can only cover a limited region of the environment at any given moment in time. Moving to intercept an approaching boat is a significant commitment to cover one region over another because the underactuated boats cannot instantly recover to their previous position. Movements of defending agents could be exploited by feinting attackers, drawing aggressive defenders out of an advantageous position as they move to intercept. Moving to defend one region may create vulnerabilities in others.

An overly aggressive defensive strategy, such as choosing to intercept far away from the asset being defended, may be of dire disadvantage if the defender cannot nimbly recover toward a newly-vulnerable position. Intelligent strategies are required to perform well in this adversarial environment, and developing these defensive schemes requires effective evaluation of their performance. Unfortunately, defensive strategies are typically evaluated solely by final outcome statistics. A successful strategy may involve a complex series of behaviors and movements, yet the evaluation of its effectiveness commonly comes down to just two questions: "Did it work?", and the slightly more advanced, "How often did it work?". What's missing is "Why did it work?".

In the realm of sports, performance statistics can be as simple as win vs. loss percentage or point differential, but they have grown much more complicated in recent years. For example, in baseball Sabermetrics [15], the goal is to identify behaviors during a game that are correlated with team victories. Wargames have been approached with game theory [32], and attempts have been made towards developing combat scenario run-time metrics [28], [10]. However, aside from information availability and its effects, run-time metrics for a combat defense have not reached a level of complexity near that employed in sports analytics.

Basic outcome performance statistics do not carry any information of the details that contributed to such an outcome. They do not create a connection between the run-time state of the system and the final result. Without more detailed feedback, the designers of defensive algorithms must seek out methods to help bridge the gap between agent behaviors and outcome performance. Taken to an extreme, the designers may even remove themselves from the design process entirely by relying on evolutionary or reinforcement learning methods. Even after sucessfully applying these techniques, it may be difficult to quantify or explain why exactly the resulting strategies are more or less effective.

This work seeks to connect outcome performance with run-time metrics that describe the state of a team attack and defense scenario. The teams in this scenario represent the small boats attacking a large ship and the satellite defensive boats used to intercept them. To this end a novel family of run-time metrics, called Time-To-Arrive (TTA), is introduced. In a literal sense, TTA describes the instantaneous physical readiness of a team of agents to defend regions of their environment. The maneuvering capability of an agent is described in its body frame, estimating the amount of time required to navigate to any given location around the agent. TTA inherently encodes the agents' dynamics and control. The TTA data from individual team members can be combined to describe the system state.

A simple simulator was built to run thousands of instances of this scenario with six combinations of attack and defensive strategies, along with variations in several other critical parameters such as defender top speed. Both the outcome performance and TTA data were collected and used to create an intuitive explanation for why one strategy outperformed the others.

1.1 Related Work

While this work focuses specifically on a metric to describe the run-time state of a defense, it shares aspects with various areas of research.

1.1.1 Offensive and Defensive Tactics

Some of the work explicitly attempting to develop metrics of tactics appears in the domain of sports, be it humans or robots on the field of play.

Clemente et al. [8] developed a method to inspect the tactical position of soccer players through a novel metric to describe the area covered by triangulations of players, called the Defensive Play Area. They perform statistical analysis on the outcome of soccer games against their metric. Their work is specific to the physical space of a soccer field, and their metric is a good example of a focus on position information rather than the potential positions that an agent may travel to.

Soccer is not the only sport to utilize more advanced player position metrics. The National Basketball Association has a advanced analytics arm that uses the SportVu optical tracking system to generate a massive data set. Maymin [23] sought to create a taxonomy of discrete basketball plays by augmenting player state with acceleration.

Other work in robot soccer does not have a metric as its direct focus, but certainly utilizes them to improve play. Biswas et al. [3] developed an advanced two-stage planner to explicitly attempt to coerce opposing robots into leaving openings in their defense. Their work focused on choosing actions to maximize the probability of goals and minimize risks of goals and serves as an example of expert-designed attack and defense. They also consider potential events, in that they address opponents that might receive a pass, not just the robot that currently holds the ball. The most recent detailed description of their small size league team reveals the complexity required to outperform in this space [24]. Unlike this work, their focus was on an aggressive offense rather than the defense.

Interception behavior has been established in the robot soccer domain for some time, e.g. [9]. They consider simple equations of motion for each instance, follow a ball on a straight line, and do not calculate this time of arrival information offline. The time-to-arrive developed for this work uses offline calculation and involves both both translation and rotational control actuation.

1.1.2 Patrol, Security, and Surveillance

Robots have been proposed for applications in security and surveillance. Homeland Security has taken interest in UAVs [4]. Guo et al. [16] developed a distributed localization and multi-agent path planning algorithms for autonomous security robots to reduce the need for modifcations to the environment to accomodate them. Additional focus has been placed on optimizing various aspects of the task and teamwork. Elmaliach et al. [12] attempts to create a multi-agent patrol such that the points of interest receive uniform coverage. Nilsson et al. [26] attempt to optimize line-of-sight coverage of a perimeter with small ground vehicles, and Anisi et al. [1] take this one step further to plan paths that minimize the required time to achieve this optimum. Chung et al. [7] optimize the probability of detecting an underwater intruder in a small channel of water. Performance in coverage and patrol can be quantified in such a way that optimization can be performed directly. This work seeks to quantify the state and quality of a defense, hopefully to serve as a basis for future work in optimizing the algorithms of attack and defense.

1.1.3 Measures of Effectiveness in a Combat Scenario

Perry et al. [28] focus on latency before actions are taken and the effectiveness of information exchanged between an offensive agent and a decision maker. They propose an information theoretic estimate for the expected time required to attack a surfacing submarine with a missile or a UAV given the effects of collaboration and network complexity. Darilek et al. [10] take a more game theoretic viewpoint, but still focus on the effects of knowledge on the outcome. They outline the Lanchester Laws [19] (quality vs. quantity of agents in a combat scenario) in a modern context, now with information availability as the primary measure of agent quality. Perry and Darilek did not simulate the actual agents involved and focus more on infrastructure.

A kamikaze style UAV team attack against a single Navy Destroyer was considered in [29]. They performed a Monte Carlo simulation of a single asset first sensing the incoming attack and defend itself with long range weaponry before the UAV's reached the ship. Better sensors improved sensing range, and better weapons improved the probability of destroying a UAV. They produced a cost-benefit analysis of improving sensors vs. improving weapons. In contrast, this work focuses on a helpless asset and its team of defenders, and variations in tactics rather than equipment (with the exception of defender speed).

Tiwari [33] unsuccessfully attempted to simulate a scenario that closely matches the real-world context to the one presented in the introduction here - small boat swarm attacks against Arleigh Burke Class Guided Missle Destroyers that initiates with small boats masquerading as harmless civilians. Ding [11] successfully implemented a simulation that examines different interception/guidance strategies in the same scenario used in this work. A helpless, stationary asset being attacked by a swarm of small boats must be protected by satellite defensive boats. Ding implemented a pursuit guidance interception algorithm inspired by missile control to track the centroid of a group of attackers. The algorithm seeks to keep an attacker in front of the defender before, during, and after an interception attempt; like this work, Ding emphasizes that a missed interception opportunity might leave the asset undefended. The defender speed and number of defenders were also varied. However, Ding did not evaluate the run-time state of the defense, focusing only on the winning percentage. Also, the attackers did not attempt feinting or deception, instead relying on superior numbers as they charge the asset, totally ignoring any defenders.

Seo et al. [31] simulate an anti-torpedo combat system. Similar to Ding, they adhere to probability of survival as their measure of effectiveness (MOE). Hong et al. [17] make an effort to connect their MOE with the required equipment measure of performance (MOP) in their simulation of the same torpedo defense problem. The MOPs are parameters that govern how often a particular system works, e.g. how often a decoy successfully misleads a torpedo. They begin with a desired outcome performance, MOE*, representing a simulated ship surviving at that desired probability, and design their simulation experiments to find the minimum necessary MOPs.

1.1.4 Role/Task Assignment

Like other multi-agent applications, a team of robots engaged in attack and defense will need to divide tasks between the members of the team. Vail and Veloso [34] used a distributed bidding system to delegate the roles of attacker, offensive support, and defensive support in the RoboCup-2002 competition, using potential fields to describe the relative positions of the ball, walls, goals, and agents. Michael et al. [25] also used a distributed bidding system, using it to coordinate several robots into formations.

In this work task assignment is limited to assigning a specific defender to intercept a specific attacker based on minimum time to the interception. Pereira et. al [27] develop a "reachability" map for task assignment and path planning, where robots of varying size and actuation capability must be able to physically reach an area of the map before it is assigned tasks there. The concept of physical reachability is analogous to this work's TTA, though TTA has an additional time requirement component.

1.1.5 Reward Shaping

Reinforcement learning requires a source of reward for the learning agents. Complex, even heterogeneous [21], team behaviors can be created from basic rewards, such providing a reward of $+1$ when a goal is scored by a soccer team and -1 when a goal is scored against the team. [2] explored role specialization in learning robot teams, focusing on the domain of robot soccer. Their rewards are limited to outcome performance, i.e. goal events and do not consider behaviors in between these events. The concept of reward shaping deals with speeding up reinforcement learning through intermediate rewards, also called "progress estimators" [22]. This work views TTA as a potential candidate for reward shaping, as a progress estimator for learning behaviors that exhibit robustness against intelligent adversaries. While the form of TTA metrics presented here is a designed heuristic, some work in reward shaping has shifted toward learning agents generating their own shaping rewards [18].

1.1.6 Predator-Prey

The predator-prey domain can be analogous to the attack and defense presented here. Though the most basic systems deal exclusively with predators chasing prey, Gomes et al. [14] presents a system of three types of agents: fox, sheep, and shepherds. These play similar roles to the attackers, asset, and defenders in this work respectively, though here only one asset is used and it is stationary. Martinson and Arkin [21] constructed a scenario with a team of robots taking on heterogenous roles in order to sweep for mines while dealing with other adversarial robots.

Chapter 2

Methods

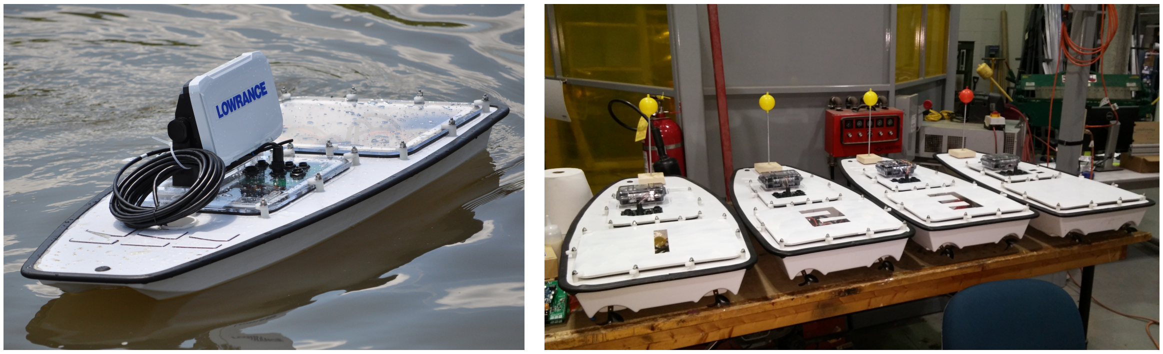
A 2D simulation of offensive and defensive teams of small robotic boats was used to demonstrate the use of TTA metrics and their connection to outcome performance. Instead of a high-score or zero-sum game, the simulated scenarios focus on the protection of a single "asset". The details of this scenario are discussed in Section 2.2.3.

2.1 Lutra Platform

The boat model used in the simulation for all agents is based on the latest generation of the Lutra series of small robotic boats, typically used for environmental monitoring and bathymetry[20]. The boats were developed for the Cooperative Robotic Watercraft project [35].

Each boat is approximately 1 meter in length and slightly less than 6 kg in mass. A Lutra boat can be fitted with a large fan that sits on a rotating base for an airboat design or two smaller propellers below the water line. In either case the result is an underactuated vessel that is fairly maneuverable, in that rotating in place is a viable type of movement. For a boat with a longer hull, designed for straight line speed, rotating in place is either inefficient or slower than changing the boat's heading by traveling forward along a curve.

While this low-cost commerical design is unlikely to appear in any real combat scenario, it is an appropriate platform to demonstrate multi-agent tasks on small scales.



(a) A boat deployed with bathymetry hardware (b) A group of Lutra boats

Figure 2.1: The Lutra platform

2.2 Simulation

The dominant factors that determines the manueverability of these small boats are the forces that appear in the plane of the water's surface, most notably drag. While buoyancy, water displacement, and pitching and rolling of the boat will all influence the movement and performance, these phenomenon are viewed as higher order effects that do not need to be included in a simulation that captures the gross movement and strategic principles at hand. By neglecting these effects in the simulation's simplified physics model, one can focus computational effort on evaluating the advanced metrics proposed by this work.

Environmental disturbances such as flowing water, wind, and wave action are also not modeled. While these could have significant impact on maneuverability and strategies, these are not included in initial testing for simplicity. With a simplified dynamic model, an ordinary differential equation integration scheme was used to propagate the effects of control actuation (i.e. motor thrust) and environmental effects (i.e. drag) forward in time, thus updating the state of every boat appearing in the simulation.

The simulation is written in Python, built around the SciPy ODE integration package [13]. The simulation updates at a rate of 20 Hz. The state information of all agents is immediately available to all agents, both offensive and defensive. While this is also unrealistic in terms of a real-world scenario, it allows the availability of information to be isolated from the performance related to maneuverability and interception strategies.

2.2.1 Dynamic Model

A state-space model of the boat dynamics is used for ODE integration. The state $\bar{q}$ is defined as

$$\bar{q} = \begin{bmatrix} x \\ y \\ u \\ w \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \text{inertial frame x coordinate} \\ \text{inertial frame y coordinate} \\ \text{surge velocity} \\ \text{sway velocity} \\ \text{heading} \\ \text{heading rate of change} \end{bmatrix}$$

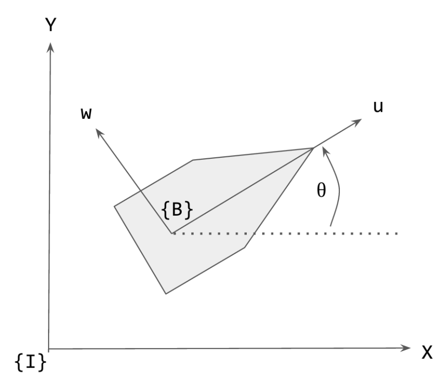


Figure 2.2: Individual boat state geometry

The time derivative of the state, $\dot{\bar{q}}$ is defined as

$$\dot{\bar{q}} = f(\bar{q}, t)$$

$$\dot{\bar{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{u} \\ \dot{w} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} u\cos\theta - w\sin\theta \\ u\sin\theta + w\cos\theta \\ \frac{F_u}{m} - \frac{1}{2m} a_u c_u \rho |u| u \\ \frac{F_w}{m} - \frac{1}{2m} a_w c_w \rho |w| w \\ \dot{\theta} \\ \frac{\tau}{J} - \frac{1}{2J} a_\theta c_\theta \rho |\dot{\theta}| \dot{\theta} \end{bmatrix}$$

symbol	**definition**
m	mass
J	rotational intertia
F_i	thrust, $i \in \{u, w\}$
τ	moment
a_i	drag area, $i \in \{u, w, \theta\}$
ρ	water density

Note that an airboat may have a nonzero sway thrust ($F_w \geq 0$), but a propeller boat can have a strictly zero sway thrust ($F_w = 0$). The moment τ is produced by the sway thrust in an airboat and by the force couple created by the difference between the two propellers in a propeller boat.

The boat parameters were based on the Lutra platform such that max speeds and stopping distances in the simulator were roughly similar to that of the real platform. This included adding a small linear drag term at low velocities, as the turbulent drag term (reliant on the square of velocity) became too small at low velocities. Exact matching would be problematic with a simple simulation, but this was not necessary for an sufficiently accurate simulation.

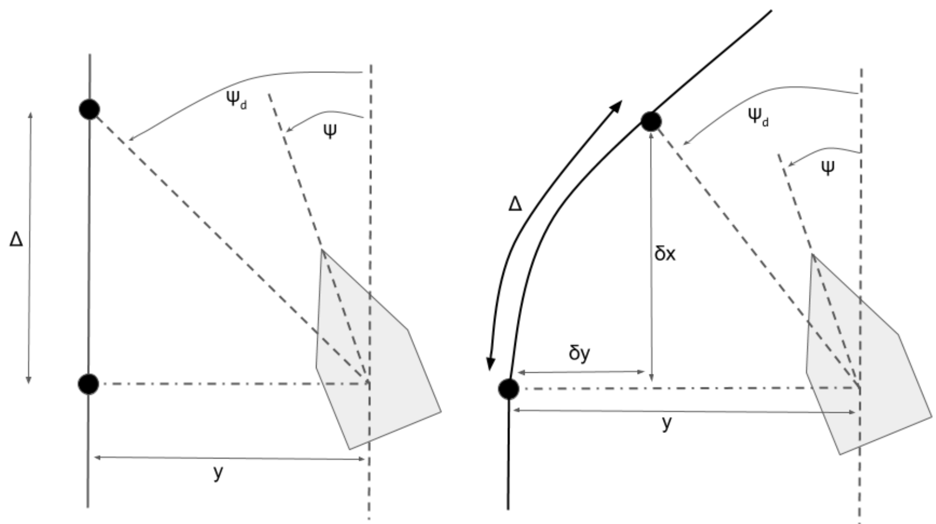
2.2.2 Control

Instead of trying to maintain a complex controller for all types of movement, and because complex behavior can be created by chaining several primitive behaviors together, several simple controllers were constructed for a library of primitives. Each primitive was attributed its own specific controller, and as chains of behaviors played out, the boat initialized the appropriate controllers. For example, strictly changing heading (turning in place) has a distinct, less complex PID controller compared to a full "point-and-shoot" PID used to navigate toward a goal location. All controllers iterate at 20 Hz, and full state information was available with no noise.

Simple location-based PID controllers like the point-and-shoot do not allow for direct velocity control. For curve following with velocity control, a line-of-sight (LOS) control designed for straight lines [5] was adapted to handle any parameterized curve (including lines, circles, and splines).

LOS Control

Line-of-sight control is simply described as aiming the vehicle at some point lying on the desired path, usually at some fixed “look-ahead” distance relative to the vehicle’s current position. This point will be referred to as the look-ahead point. This can be broken down into a simple trigonometry problem in the Frenet-Serret frame, located tangent to the desired path with its origin at the lookahead point. 2.3 shows this geometry.



(a) LOS triangle for a straight line (b) LOS triangle for a curve

Figure 2.3: Line-of-sight geometry for a straight line and smooth curve

[5] uses an novel integral term to deal with constant disturbances, such as water flow, but this is not included here due to absence of disturbances. That simplifies the equation for the linear path case to

$$\psi_d = -\tan\left(\frac{y}{\Delta}\right)^{-1}$$

Generalizing this to a curve requires the look-ahead Δ to be a path length along the curve. A triangle is still required, and the y component of the look-ahead along the curve is subtracted out to produce a triangle and the equation

$$\psi_d = -\tan\left(\frac{y - \delta_y}{\delta_x}\right)^{-1}$$

Calculating the look-ahead point for simply parameterized curves, such as lines and circles, is a simple task. Applying this to splines required more complex calculations.

A closed-form smooth spline generator was used to generate chains of splines [30], and a hybrid form of quadratic minimization and Newton's method minimization to find the point on the spline closest to the vehicle [36]. Once this closest point was found, the splines formulas quickly produce the look-ahead point.

2.2.3 Simulated Scenario

The specific scenario presented in this work is the defense of a single, stationary asset in open water. There are no obstacles or limitations on movement. The asset has no means to defend itself and relies entirely on a small team of defensive agents. The scenario is centered on the position of the asset.

The scenario begins with the defensive agents arrayed around the asset in a ring of defense. A team of attackers appears in the region around the asset and attempts to approach the asset. The defenders must prevent the attackers from getting too close. An example simulation screenshot from this scenario is shown in Figure 2.4.

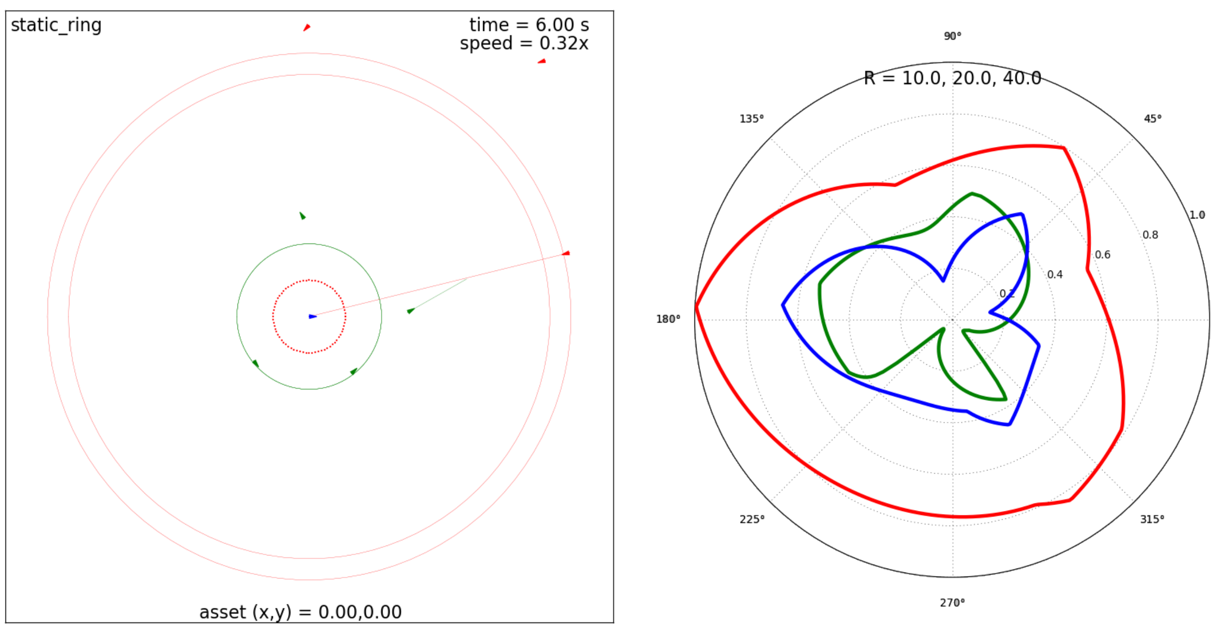


Figure 2.4: Example screenshot from the simulation scenario

In Figure 2.4 left side: The asset is the blue arrow in the center, the green arrows are the defenders, and the red arrows are the attackers. The dotted red circle around the asset indicates where a successful attack must reach. The other lines indicate current desired paths. In Figure 2.4 right side: Defender minimum time-to-arrive around the asset. Green ring is the minTTA along the perimeter of the 10 m radius circle, blue is the minTTA along the perimeter of the 20 m radius circle, and red is the minTTA along the perimeter of the 40 m radius circle. See section 2.2.4 for more details.

The scale of the environment and range of attack is proportional to the size of the Lutra platform. If any attack boat reaches within 5 meters of the asset, the asset is considered to be successfully attacked, the defenders lose the scenario, and the scenario is terminated. If any defender is within 2 meters of an attacker, there is a probability < 1 that the attacker will be removed from the scenario. If all attackers are removed, the defense wins the scenario, and the scenario is terminated.

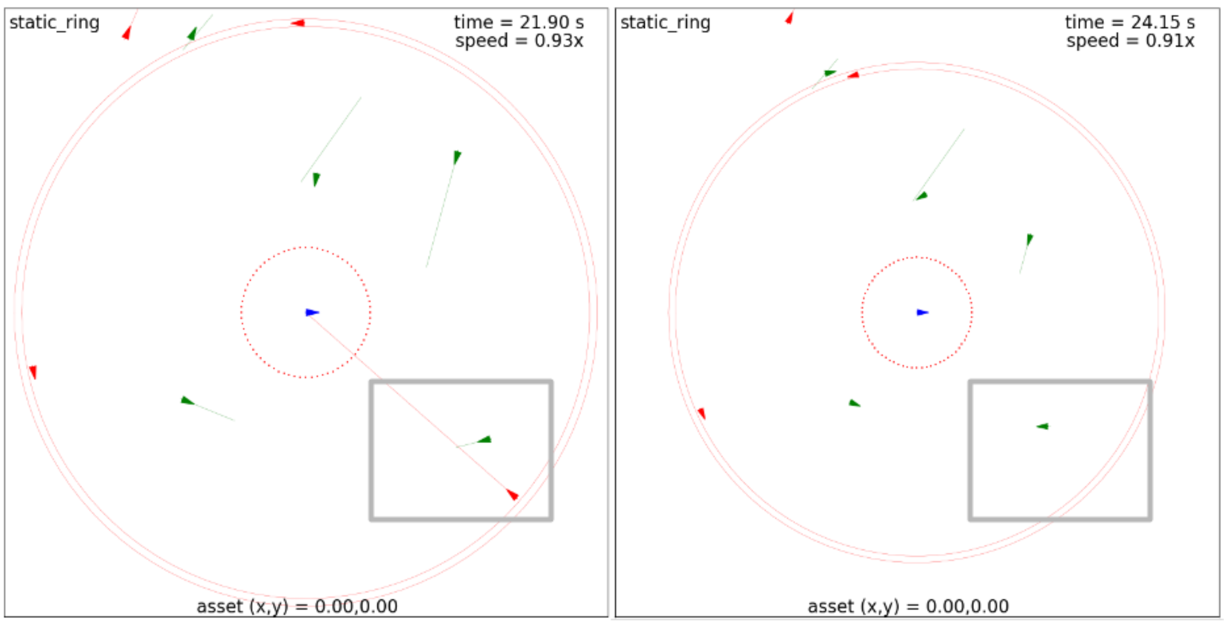
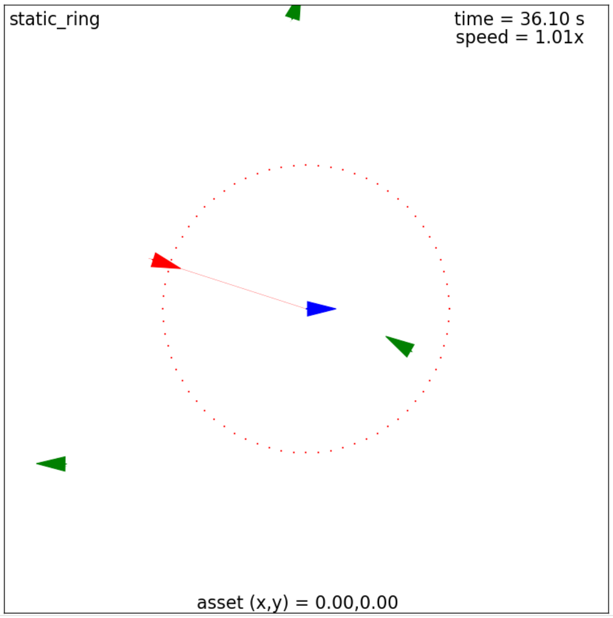


Figure 2.5: Example attacker removal event

The larger attacker vs. asset removal distance and the probability < 1 is to force the disadvantage of sparsity onto the defenders. The defenders must cover more area than they can cover by sitting still, preventing them from just sitting right on top of the asset as attackers approach.

All state information for all attackers and defenders is available to both the offensive and defensive teams. Asymmetric and/or imperfect availability of information is not in the scope of this work, as it deals primarily with the physical layout of the scenario.



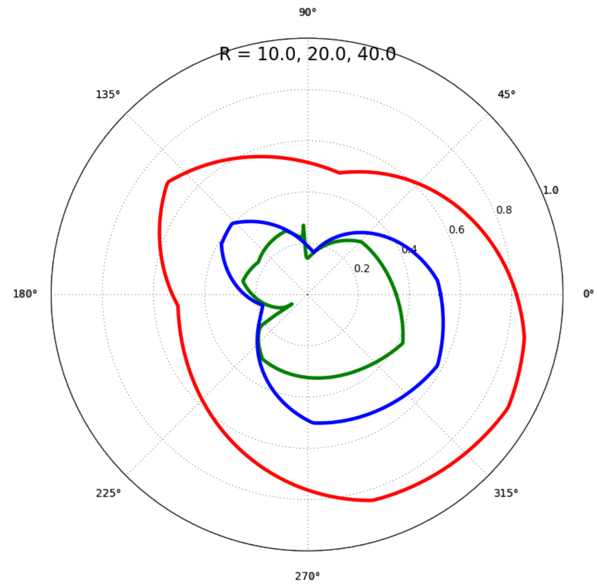


Figure 2.6: A successful attack

2.2.4 Time-To-Arrive

The name time-to-arrive (TTA) is quite literal - TTA is the time it would take an agent to arrive at a point in space, given that agent's initial state and typical control input. TTA represents the physical readiness of one or more agents to move into regions of their environment, and it inherently encodes the agents' dynamics. This is not the same as just having knowledge of an agent's state, but rather includes additional information on the availability of the agent and where the agent could possibly reach within some window of time.

This simple concept can be extended to a team of agents, and can be used to describe the state of a defense as a whole. This work seeks to show a connection between TTA and a defense's outcome performance. If such a connection exists, then TTA may prove a valuable means of evaluating the instantaneous quality of a defense, something that outcome performance statistics (such as winning percentage) cannot provide.

Body-Frame Time-To-Arrive Contours

TTA usually takes the form of some contour in space. The simplest form is a contour centered on the body frame of an agent, displaying the locations that the agent might reach along any relative heading within some fixed window of time.

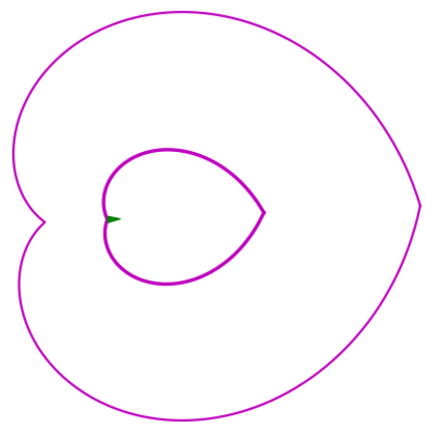


Figure 2.7: Body frame Time-To-Arrive Contours, 5 and 10 seconds

Note how, in Figure 2.7, the contours behind the agent are much closer together than in front of it, and that the 10 second contour is still very close to the back of the agent. In other words, a defender cannot address the region behind it nearly as easily as it can the region in front of it. As underactuated defenders move through space, they leave vulnerabilities in the defense behind them. Thus, shifting from one defensive formation to another can open transient holes in a defense that might not be readily apparent by focusing on defender position alone.

These contours can be treated as polygons, which enables a useful application. Judging whether a defender can arrive at a desired location within a time window is equivalent to checking if that point is within the interior of the TTA contour polygon corresponding to the time window. Union and intersection operations can be applied to a team's polygons, showing the region where any defender can arrive and where multiple defenders can arrive together respectively. This would be useful in scenarios where redundancy is required, i.e. when multiple defenders are needed in the same spot.

However, the polygon approach is limiting in that it requires a fixed time window to construct it. Instead, it is far more flexible to directly calculate time to arrive for any point of interest. The additional complexity of that calculation is addressed in section 2.2.5

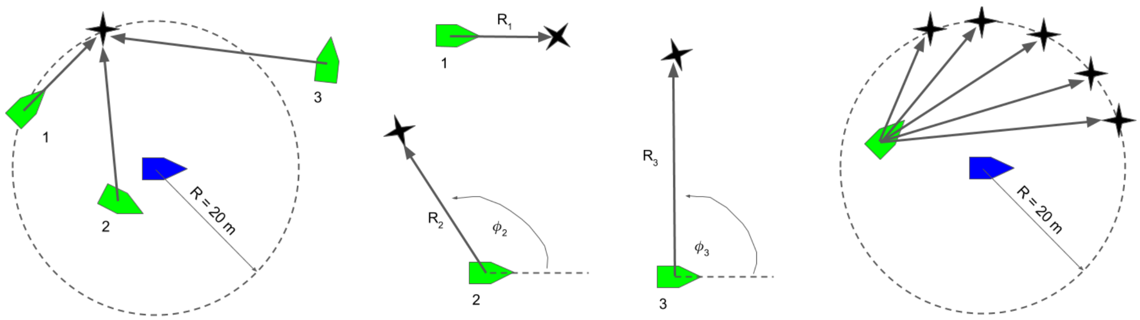
Minimum Time-To-Arrive

For a given point in space, each defender has its own TTA. Amongst the able defenders, there will be a minimum TTA. This will be abbreviated to "minTTA". Identifying which defender has the minTTA is useful for task allocation, as it allows the defender with the smallest intercept time to be assigned the task of intercepting some target. The smaller the minTTA, the easier the task.

Fixed Radii Time-To-Arrive Contours

More complex measures of TTA can be constructed to provide additional insight into the specific scenario of team defense of a single asset. This work will focus on a single metric: fixed-radii TTA contours.

An asset can be attacked from any angle in this simulation. As such, if the defenders' minTTA was calculated at points on a circle centered on the asset, this would provide information about the quality of defense surrounding the asset. A set of several fixed-radii circles were chosen, specifically 10 m, 20 m, and 40 m from the asset. The minTTA was calculated at 1 degree increments for each circle. The geometry of this calculation is shown in Figure 2.8. An example of the output is shown on the right hand side of Figure 2.4. A smaller value is desired, as this represents a shorter time until the arrival of a defender. Section 2.4 discusses further details these rings and desirable characteristics.



(a) (b) (c)

Figure 2.8: Fixed radii Time-To-Arrive geometry

TTA at Time of Final Attack

When the last attacker in a simulation instance charges at the asset, whether it reaches the asset or not, this sets the "time of the final attack" to the moment it begins its charge. An attacker that exploits a vulnerable defense makes the decision to do so at this moment. This moment in time is interesting to isolate, as it captures the attack decision that led to the final outcome of the simulation.

Maximum minTTA

For a specific fixed-radii ring of minTTA values, the maximum value of minTTA around the perimeter is referred to as the "max minTTA". This value represents the minTTA value of the defense at the most vulnerable heading angle with respect to the asset. The value at the 20 m ring was used for the analysis in this work. This ring was selected as it is in the region where the majority of interception interactions occur.

2.2.5 Offline Time-To-Arrive Calculation and Model

Calculating TTA for each defender at every time step in real-time requires a simplified model, as it is infeasible to apply a physics simulation repeatedly. To address this, TTA information can be calculated offline if the system of actuation and control is relatively simple.

TTA is calculated in the body frame of the agent. It represents the time required to arrive at a specific point given the initial state of the agent. An even further simplified form of the ODE-integration simulator can be used to generate this information for a large set of points around an agent. A polar grid of points in a semicircle $\phi \in [0, \pi]$ is used as the set of goal locations. Only a semicircle is required due to symmetry. Each iteration represents a combination of initial state, and some point in polar space around the agent $x = (R, \phi) \in \mathbb{R}^2$. As this is in the body frame of the agent, only the initial velocities have any influence on the TTA.

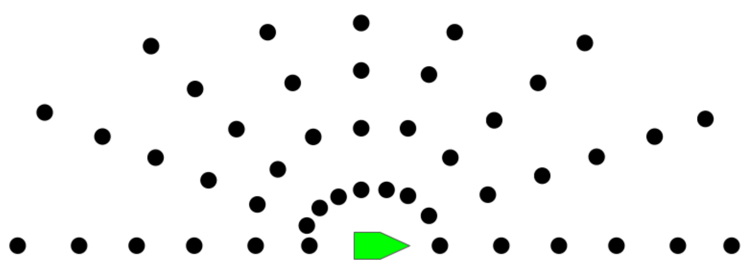


Figure 2.9: Polar Grid for Offline Time-To-Arrive Calculation

The simulation runs as the agent navigates to one of the goal points. The amount of time it took to arrive at the goal was collected for each goal location. After collecting this data for the boat design and control to be used in the defensive simulation, it was observed that the TTA was a nearly linear function of R, ϕ, and initial surge velocity, $u0$. A least-squares hyperplane was fit to the data (following the over-constrained matrix formulation in equation (2.1), thus producing a bare-bones complexity model of TTA that can be involved in thousands of calculations per time step in the simulator. The coefficients are listed in Table 2.1. These values are specific to the physical nature of the boat model and the control used to navigate to the goal locations.

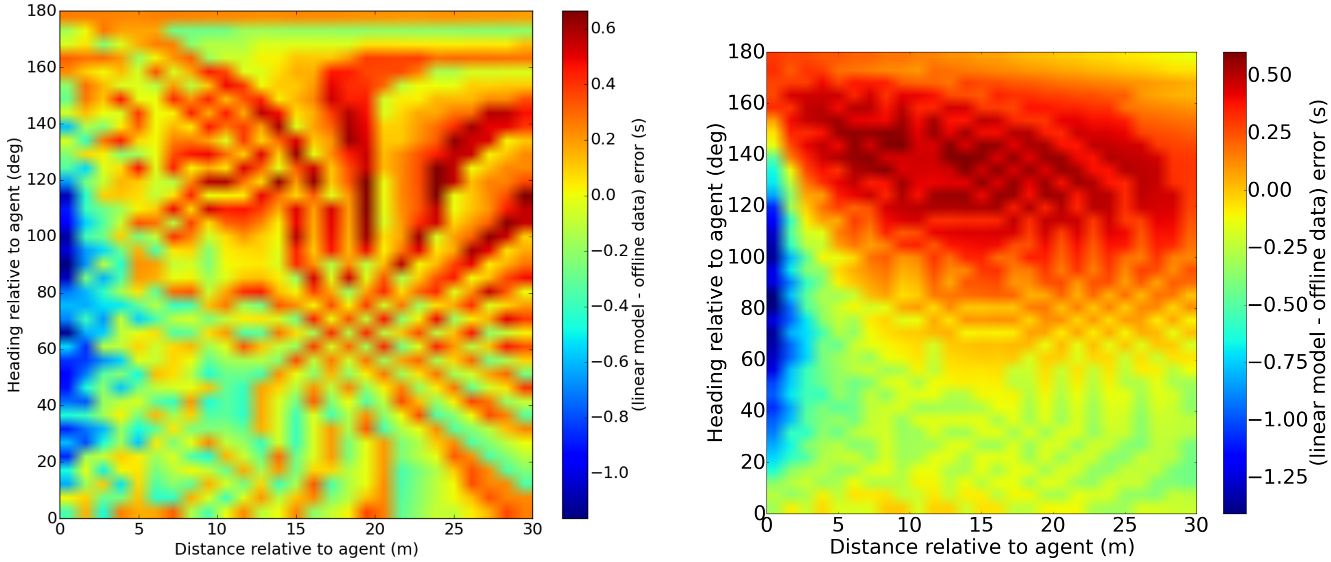
$$\begin{bmatrix} \theta^{(1)} & R^{(1)} & u_0^{(1)} \\ \theta^{(2)} & R^{(2)} & u_0^{(2)} \\ \vdots & \vdots & \vdots \\ \theta^{(n)} & R^{(n)} & u_0^{(n)} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \mathrm{TTA}_1 \\ \mathrm{TTA}_2 \\ \vdots \\ \mathrm{TTA}_n \end{bmatrix} \tag{2.1}$$

	Fast Defender	**Slow Defender**
a	2.54833	3.54372
b	0.40135	0.55509
c	0.09148	0.08567

Table 2.1: Least-squares linear fit coefficients for TTA

Note that the point-and-shoot behavior used in these offline calculations is not necessarily optimal, but it is the control most typically applied in the simulation, and thus provides the most utility in this situation. Figure 2.10 shows the fitting error of the linear model used in the simulation. Seeing that the TTA values could be off by a full second, any defenders using this model to intercept an attacker would include a 2 second buffer of extra time.

In an environment without any non-conservative forces, i.e. the drag present in this simulation, it would be possible to generate optimal feed-forward control and avoid a physics simulation altogether.



(a) Fast defender

(b) Slow defender

Figure 2.10: Offline least-squares TTA model error

2.3 Offensive and Defensive Strategies

Two variations of the attack and three variations of the defense were combined with various other parameters (e.g. number of attackers and defenders) in each iteration of the scenario. Table 2.2 lists all six combinations.

Case 1 Attack: Random Defense: Static	**Case 2** Attack: TTA-Expliot Defense: Static
Case 3 Attack: Random Defense: Turned	**Case 4** Attack: TTA-Expliot Defense: Turned
Case 5 Attack: Random Defense: Dynamic	**Case 6** Attack: TTA-Expliot Defense: Dynamic

Table 2.2: Six primary combinations of offense and defense strategies

2.3.1 Offensive

The offensive can either have access to defender TTA information, labeled "TTA-Exploit", or simply try to approach the asset at random times, labeled "Random". The TTA-Exploit attackers have access to the 20 m fixed-radius defender minTTA metric, and they seek to travel to the azimuth around the asset corresponding to the nearest of the 3 largest minTTA values. Once they arrive there, the attackers charge straight at the asset. In other words, they are seeking out a line of attack that is more difficult for defenders to intercept according to information they have at the time they initiate their charge.

The random attackers travel in a circle around the asset, out of reach of the defenders, until they commit to a charge straight at the asset at some random time.

The following characteristics are shared between the TTA-Exploit and the random attack:

1. All attackers start at random locations around the asset, though all these locations will be far from the asset, well out of the defender's reach.
2. Both types of offense have a single attacker that starts the scenario by feinting, attempting to draw defenders out of their initial positions. Note that the feinting attacker does not dodge or "juke" around the defender that tries to intercept it and immediately attack the asset. That initial dodge was extremely effective against the simple defensive interception algorithm, so much so that it skewed the results, obscuring some interesting outcomes based on less capable attackers. Therefore the feinting attacker feints outside of the defenders' range and waits for some time before behaving like the other attacking boats in the simulation.
3. Once an attacker decides to attack the asset, it approaches on a straight line and will not try to dodge any defenders trying to intercept it. That way, any results will be focused on the effectiveness of when and where the attacker commits to its charge instead of cat and mouse mobility games.

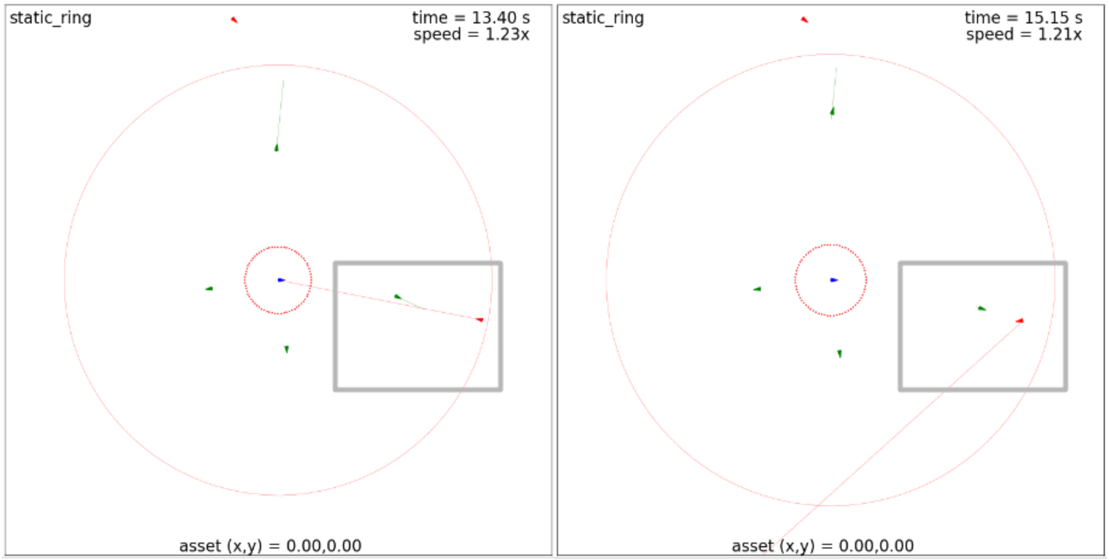


Figure 2.11: Feinting attacker

2.3.2 Defensive

Defensive strategies for each defender consist of three phases

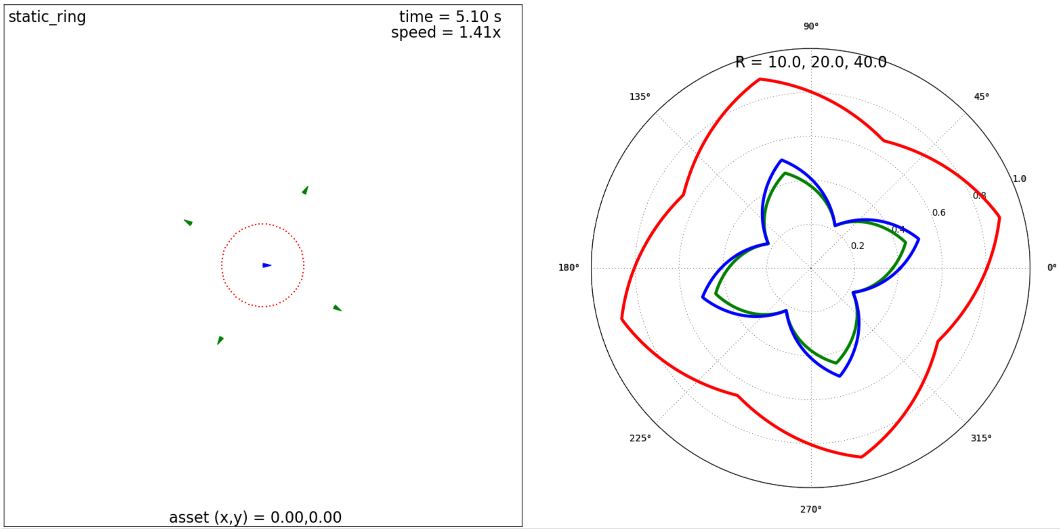
1. Preparation
2. Interception
3. Recovery

Preparation represents the defenders selecting and moving to positions around the asset to maximize their collective interception capability. Interception represents the algorithms governing the decisions of when and where to intercept an attacker in the space around the asset. Recovery is very similar to preparation, in that the defender returns to some advantageous position or movement around the asset after attempting to intercept an attacker.

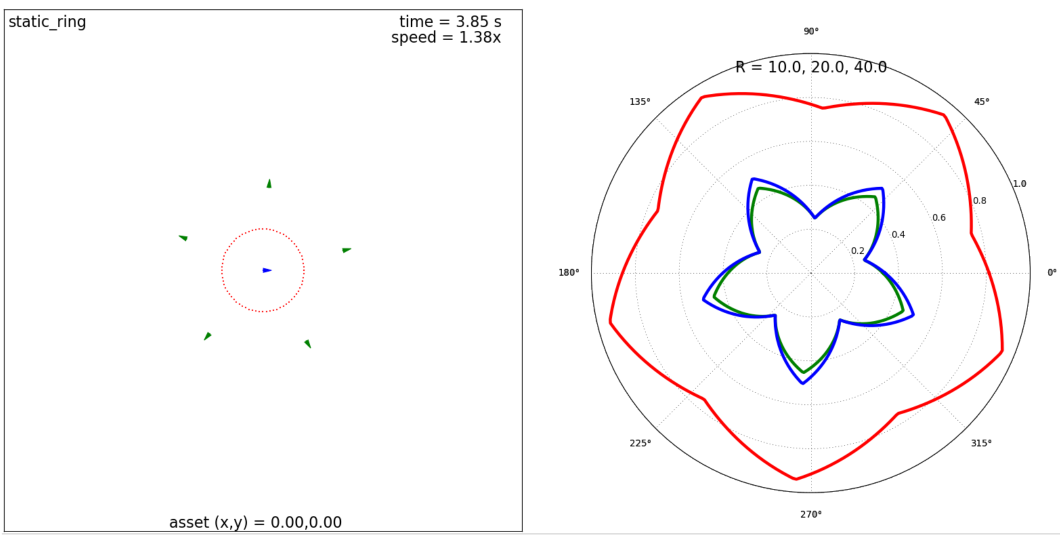
The defense is “Static”, “Turned”, or “Dynamic”. The differences between them only influence preparation and recovery. Interception behavior is the same for both static and dynamic defense.

In the static defense case, the defenders’ intial state is a ring around the asset, facing away from the asset, and with zero initial velocity. Once an interception attempt is made by a defender, it will always return to its initial position. Figure 2.12 shows the initial state of the static defense.

The turned defense case is the same as the static case, except that the initial heading of the defenders are turned 90 degrees such that the defenders are tangent to a circle centered on the asset. This defense type was included to isolate the motion component of the dynamic defense. Figure 2.13 shows the initial state of the turned defense.

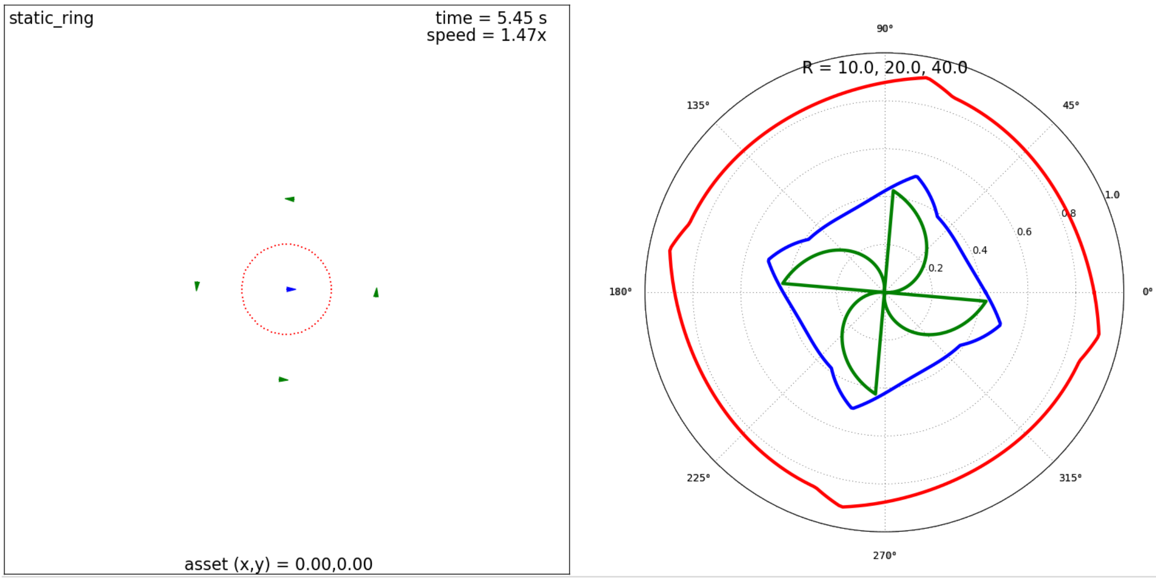


(a) 4 defenders

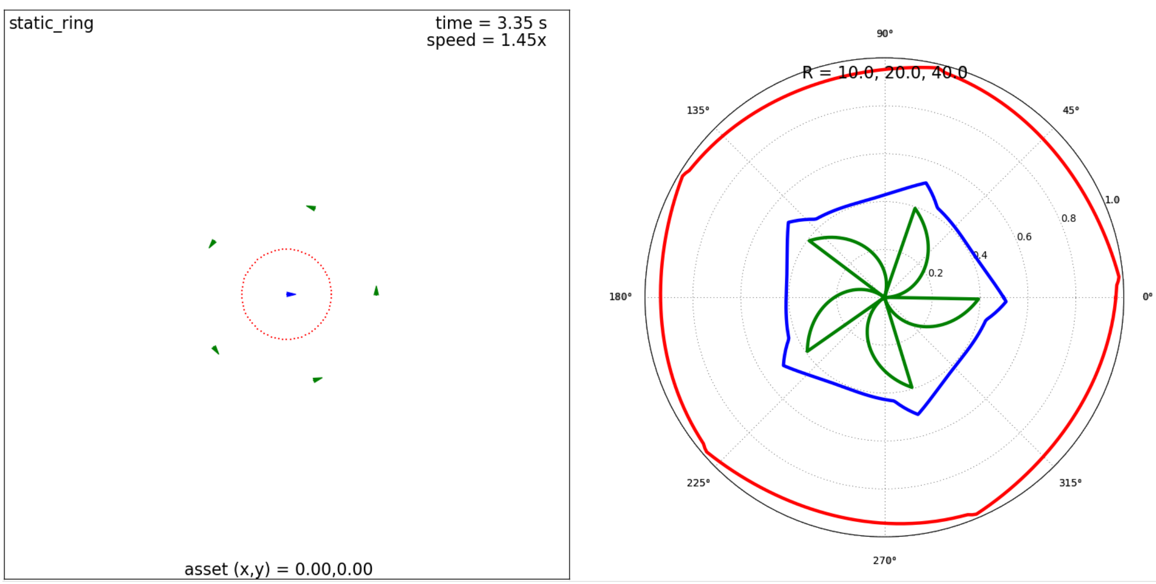


(b) 5 defenders

Figure 2.12: Initial state of static defense



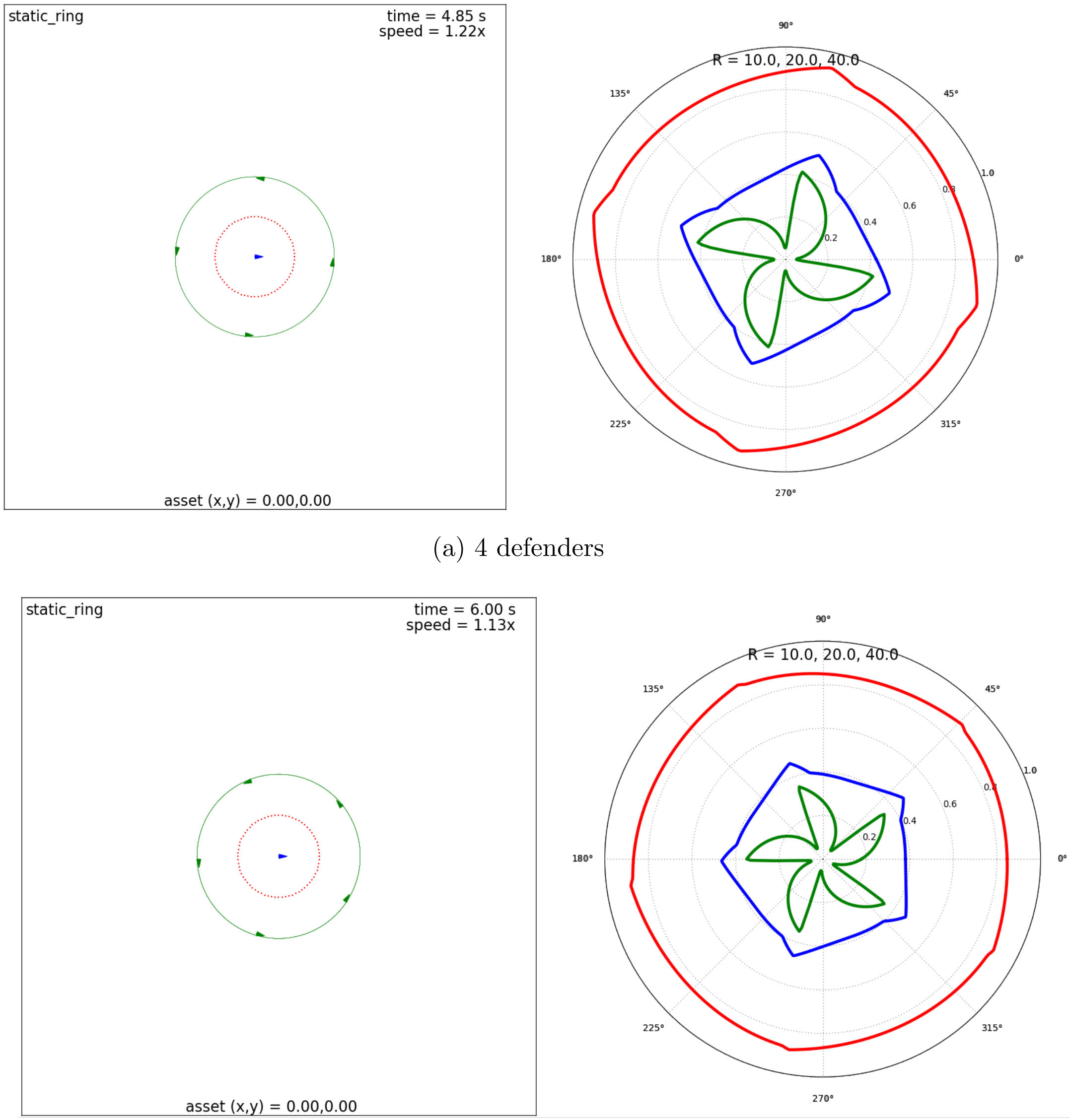
(a) 4 defenders



(b) 5 defenders

Figure 2.13: Initial state of turned defense

In the dynamic defense case, the defenders initial state is traveling around a 10 m circle centered on the asset, moving at their maximum speed. They are initially spaced out evenly around the azimuth. Figure 2.14 shows the initial state of the dynamic defense. Once an interception attempt is made by a defender, it will return to circling, but makes no attempt to spread out. It will simply begin following the initial circle immediately without regard to other defender positions.

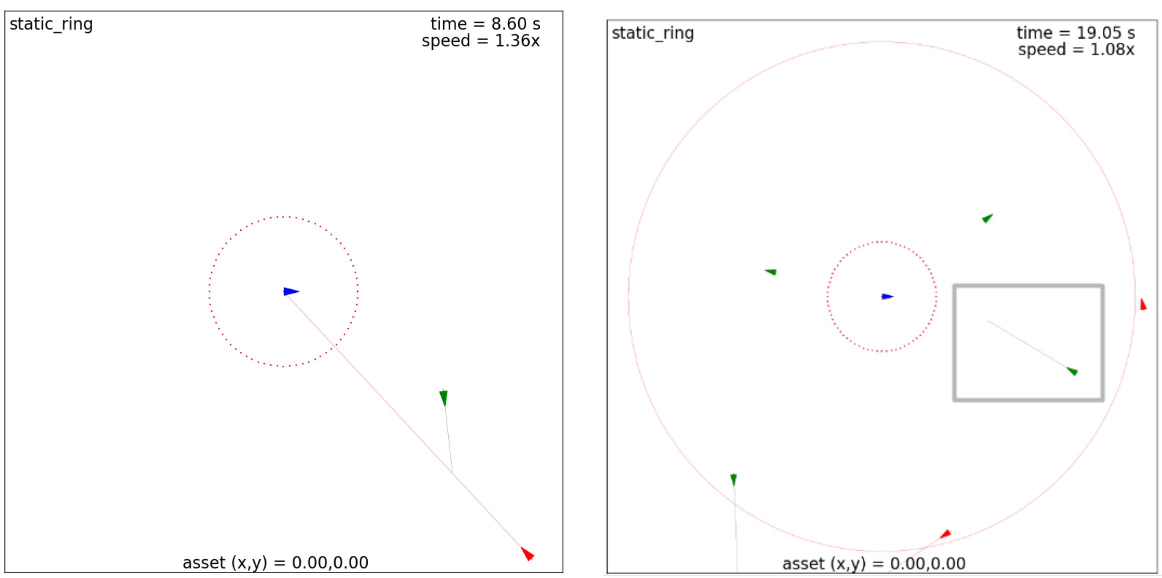


(a) 4 defenders

(b) 5 defenders

Figure 2.14: Initial state of dynamic defense

The interception behavior is the same for both static and dynamic defense. At each time step in the simulation, the defenders predict a straight line course for each attacker, assuming that the attacker's current surge velocity holds. If they determine, after comparing the attacker's assumed TTA against their own TTA along this line, that they can arrive before an attacker, they are labeled as eligible to intercept. The defender with the minimum intercept time is assign the task of interception. However, there are some restrictions to this assignment. The defender cannot be busy, i.e. they cannot be attempting to intercept another attacker. The time window for prediction is limited to 20 seconds, and the maximum distance away from the asset that they are allowed to intercept is also limited. These limitations prevent an overly aggressive defense that continuously tries to intercept and puts itself in a disadvantageous position. The maximum interception time window and maximum distance from asset can be thought of as measures of defensive aggressiveness.



(a) Interception (b) Recovery

Figure 2.15: Interception and Recovery

Once a defender has removed its targeted attacker, or if the attacker is no longer on the predicted intercept course, the defender is no longer busy and recovers according to the static or dynamic defense strategies. Note that a defender can be assigned a new interception immediately upon initiating recovery. Interception and recovery behavior is shown in Figure 2.15.

2.4 Minimum Time-To-Arrive Smoothness

As discussed in Section 2.2.4, three fixed-radii (10 m, 20 m, 40 m) minTTA rings were used to describe the state of the defense during the simulation. These values were stored into a large tensor at the rate of 5 Hz and 1 degree increments. If there were T total measurements over the course of a simulation instance, this generates a tensor minTTA $\in \mathbb{R}^{T\times 3\times 360}$. The tensor element minTTA_{ijk} represents the minimum TTA value at the i-th time step, for the j-th ring, at the k-th azimuth angle around the asset. Each simulation instance may have a different total run time, and therefore T will vary between simulation instances.

This is a large amount of information to handle after the production of thousands of simulation instances. Any individual minTTA_{ijk} value will not carry useful information by itself, but transformations of the tensor involving maxima, means, etc. can shed light on the overall state of the defense. In order to process this information and allow for intuitive interpretation of results of the four cases shown in Table 2.2, two measures of "smoothness" are introduced here, where smoothness refers to how much the minTTA rings change in value through time and around the azimuth. The two measures are called "volatility" and "circularity".

These values are calculated for various combinations of offense and defense strategies and collected into histograms for relative comparison. The absolute magnitude of both measures serves little purpose on its own.

2.4.1 Volatility

Volatility, V, measures how much the average minTTA around the perimeter of the ring changes over the course of the simulation. It is generated by first calculating the mean of the TTA along the perimeter of the rings, $\overline{TTA}^{\theta}$. Then the absolute values of discrete differences in time are summed and normalized by the total time of a particular simulation run. The values for each fixed-radii ring are pooled together into one scalar value. The smaller this value is, the less the mean TTA of the entire perimeter changes with time.

$$\overline{TTA}^{\theta}_{ij} = \frac{1}{360}\sum_{k=0}^{359} \text{minTTA}_{ijk}$$

$$V = \frac{1}{T}\sum_{j=0}^{2}\sum_{i=1}^{T-1}\left|\overline{TTA}^{\theta}_{ij} - \overline{TTA}^{\theta}_{(i-1)j}\right|$$

The resulting scalar is a single count collected in a histogram that represents the volatility's probability density function across a collection of simulation runs.

2.4.2 Circularity

The circularity, C, measures how much the time average corresponding to each point on the perimeter changes as you traverse the perimeter. It is generated by first calculating the mean (over time) value of minTTA at each location around the perimeter, $\overline{TTA}^{T}$. Then the absolute value of the discrete differences along the perimeter are summed and normalized by the number of degrees (i.e. 360). The values for each fixed-radii ring are pooled together into one scalar value. The smaller this value is, the less the mean minTTA varies along the perimeter. This can also be thought of as how warped away from a perfect circle the minTTA rings are on average over time.

$$\overline{TTA}^{T}_{jk} = \frac{1}{T}\sum_{i=0}^{T-1} \mathrm{minTTA}_{ijk}$$

$$C = \frac{1}{360}\sum_{j=0}^{2}\sum_{k=1}^{359}\left|\overline{TTA}^{T}_{jk} - \overline{TTA}^{T}_{j(k-1)}\right|$$

The resulting scalar is a single count collected in a histogram that represents the circularity's probability density function across a collection of simulation runs.

Chapter 3

Results

Table 3.1 shows the parameters that define a instance of the simulation.

Parameter	**Possible Values**
Number of defenders	4, 5
Number of attackers	3, 4
Attack type	"random", "TTA-Exploit"
Defense type	"static", "turned", "dynamic"
Max intercept distance away from asset	30 m, 40 m
Defender top (surge, rotation) speed	(1.75 m/s, 0.3 rad/s), (2.5 m/s, 0.4 rad/s)

Table 3.1: Simulation parameters

The simulation was run at least 300 iterations for each possible combination of parameters. All attackers retained a max surge speed of 2.5 m/s and a max rotation speed of 0.4 rad/s. The defenders with the same speed as the attackers are referred to as the "fast" defenders. The defenders with the reduced speed are referred to as the "slow" defenders.

3.1 Defender Performance and Metric Histograms

After pooling the results and statistics from all simulation runs, outcome performance and minTTA data were compared. The following sections describe differences across specific categories of parameters.

3.1.1 Metric Histogram Legend

The value of the TTA metrics were generated from the minTTA tensor and collected for each simulation instance. The results were divided into 6 primary subgroups, each one corresponding to a combination of the 6 attack and defense strategies. Then each of these was further divided into instances where the defenders won and where the defenders lost. The values of the metric were collected into bins of a histogram, and the defensive winning percentage was also calculated for each bin. Figure 3.1 shows a guide to interpreting the compound histogram representation of this data. In the following figures, 6 histograms will be shown in each figure. The top row contains results where random attackers were used. The bottom row contains results where TTA-Exploit attackers were used. The left-most column contains results where static defense was used. The middle column contains results for turned defense. The right-most column contains results for the dynamic defense.

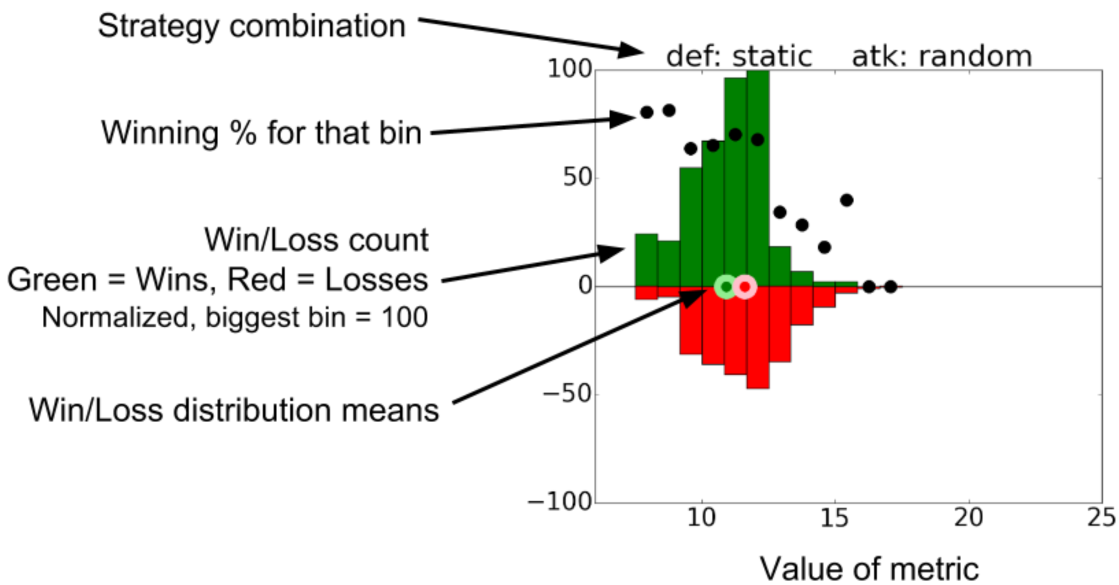


Figure 3.1: Metric Histogram Legend

3.1.2 Fast Defender, Outcome Performance

Winning percentages (winning referring to a defense's victory) are shown in Figure 3.2. Note how the TTA-Exploit attack outperforms relative to the random attack against all three defenses, but this outperformance was somewhat mitigated by the dynamic defense. The drop in winning percentage was 22.8% and 21.7% for static and turned, but only 13.6% for dynamic defense.

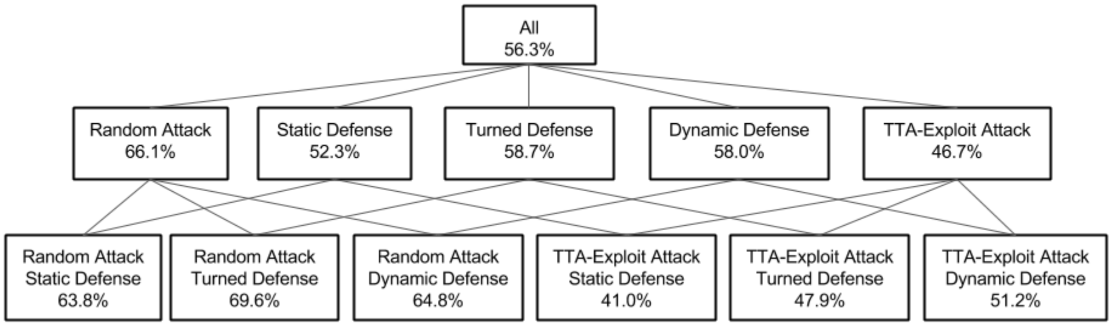


Figure 3.2: Defense winning % for fast defenders

3.1.3 Fast Defender, Final Attack Maximum minTTA

The maximum minTTA value shows a clear correlation with winning percentage, shown in Figure 3.3. Lower values of max minTTA correlate with an increase in defensive winning percentage. This is expected, as a smaller max minTTA value implies that a defense can reach all points along the perimeter in a shorter amount of time.

3.1.4 Fast Defender, Volatility

The volatility also shows a clear correlation with winning percentage, shown in Figure 3.4. Lower values of volatility correlate with an increase in defensive winning percentage. This is less intuitive as max minTTA, as it is measuring how quickly TTA values are changing rather than the values directly.

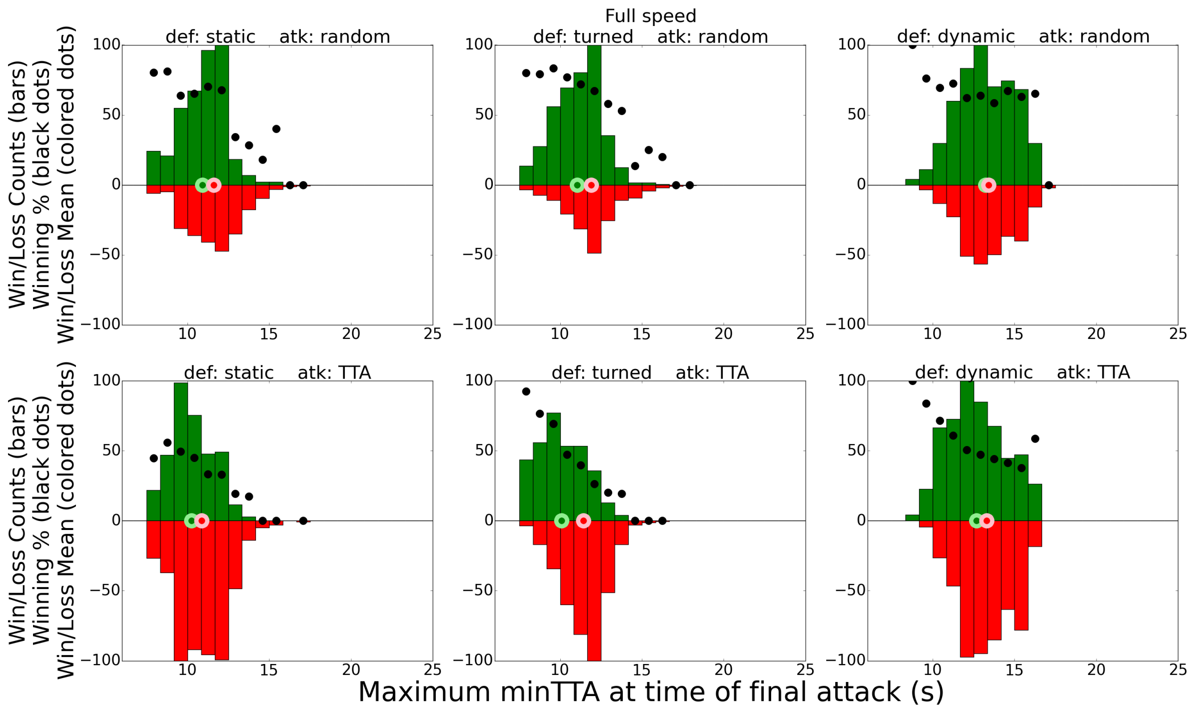


Figure 3.3: Fast Defender, Final Attack Maximum minTTA

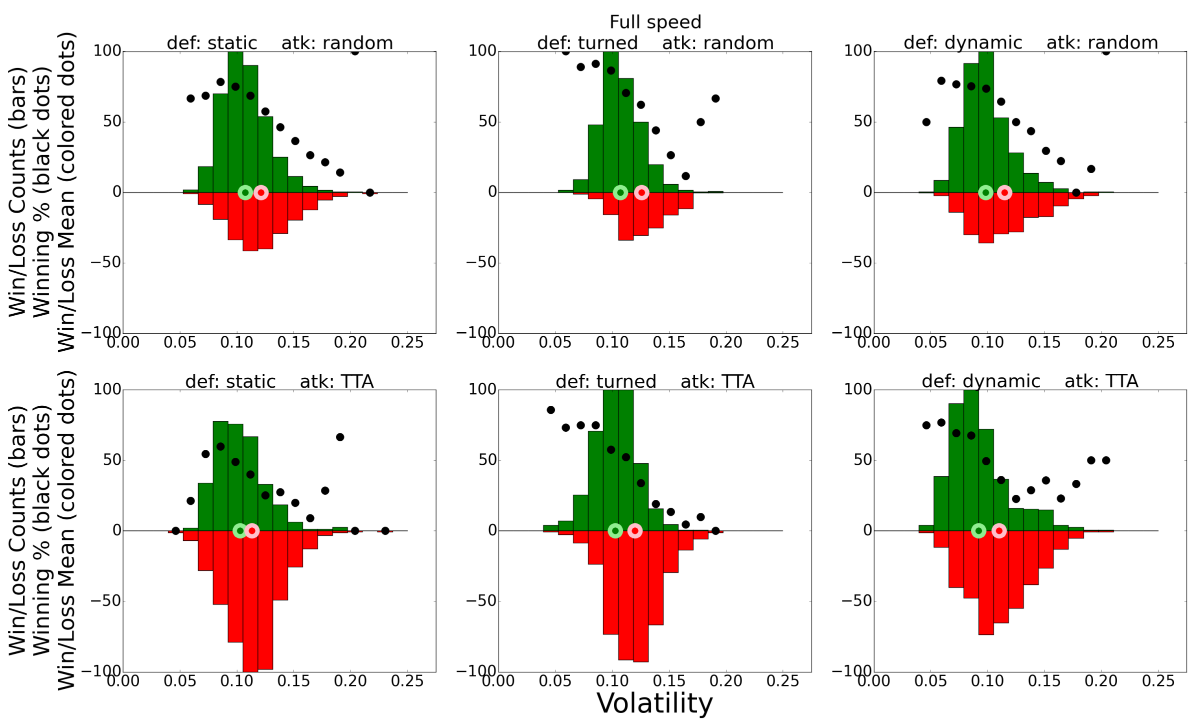


Figure 3.4: Fast Defender, Volatility

3.1.5 Fast Defender, Circularity

There is not a clear correlation between circularity and winning percentage, shown in Figure 3.5. As circularity is a measure of how warped the minTTA rings become (where a lower value is closer to a circle and higher values are more warped), a lower circularity was expected to be correlated with a higher winning percentage. This is especially true against TTA-Exploit attackers, because they are exploiting warped defenses. However, this was not observed.

Note how the dynamic defense's histograms are more tightly concentrated around the mean. By itself this does not provide much insight, but when compared against the same data for the slow defenders, this density around the mean seems to be related to the additional robustness provided by dynamic defense against TTA-Exploit attckers.

3.1.6 Fast Defender, Final Attack Circularity

The final attack circularity, shown in Figure 3.6, further highlights the lack of a clear relationship between circularity and the winning percentage.

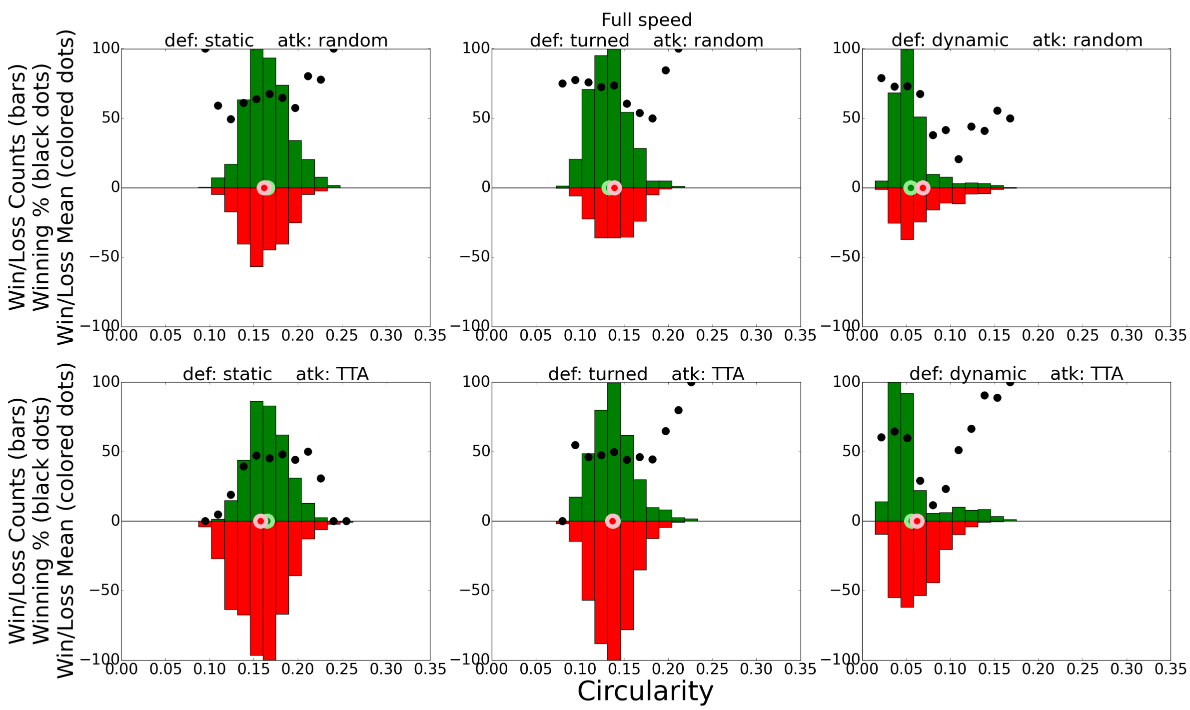


Figure 3.5: Fast Defender, Circularity

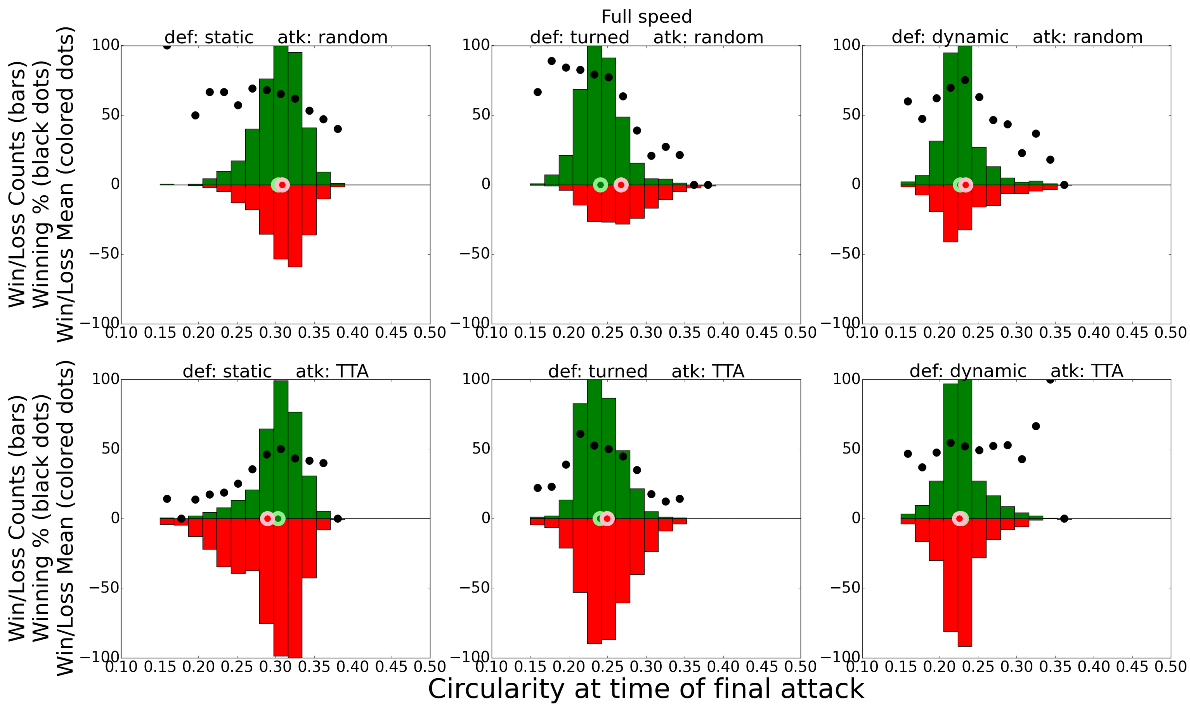


Figure 3.6: Fast Defender, Final Attack Circularity

3.1.7 Slow Defender, Outcome Performance

Winning percentages for slow defenders are shown in Figure 3.7. The TTA-Exploit attack outperforms relative to the random attack against all three defenses. With fast defenders, dynamic defense mitigates this outperformance, but the opposite is true with slow defenders. The drop in winning percentage was 9.2% and 14.8% for static and turned, but jumps to 17.1% for dynamic defense.

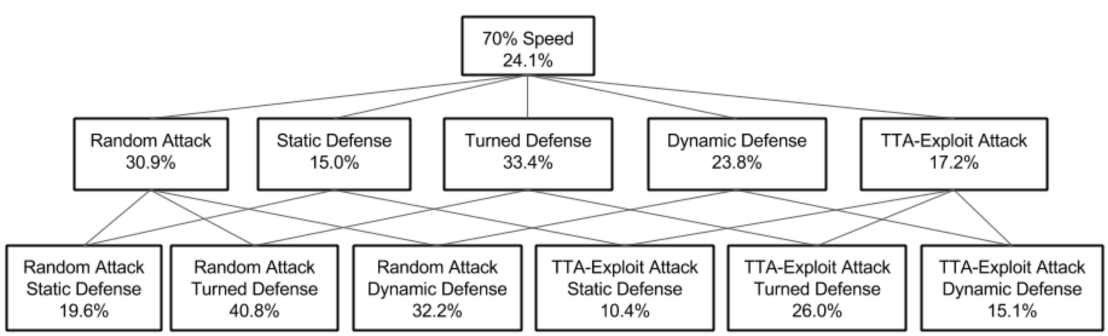


Figure 3.7: Defense Winning % for Slow Defenders

3.1.8 Slow Defender, Final Attack Maximum minTTA

Overall, the slower defenders have significantly higher minTTA values, shown in Figure 3.8. The slower defenders lose approximately twice as often as fast defender, so this is expected. Though the correlation between max minTTA isn't as clear with slow defenders, lower max minTTA values still correlate with higher defensive winning percentage.

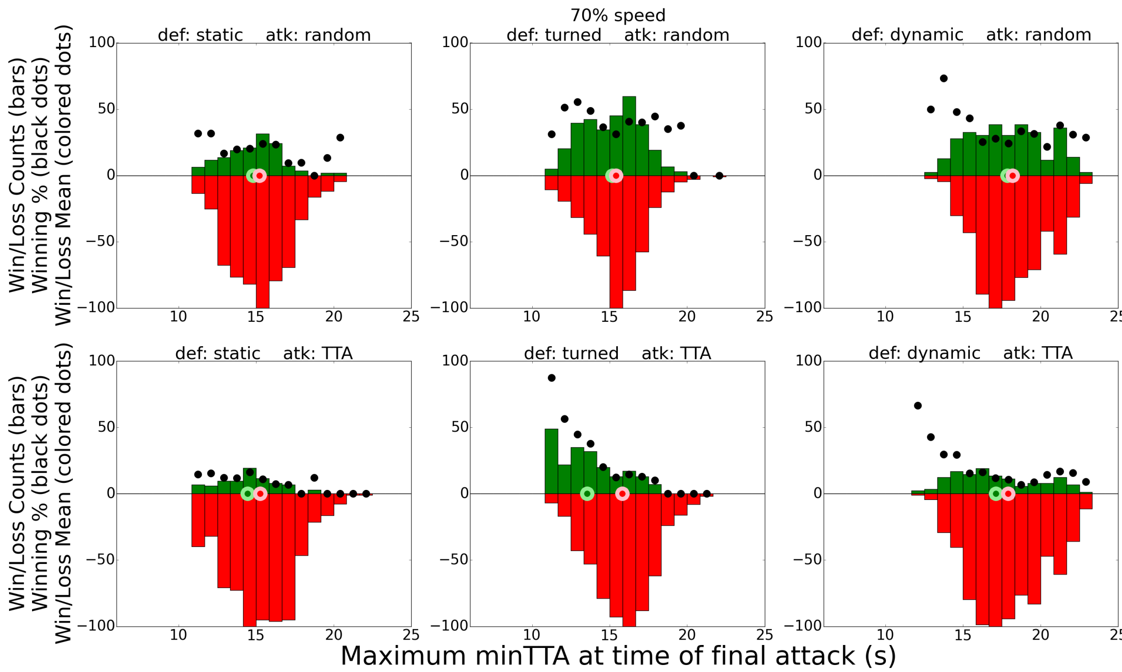


Figure 3.8: Slow Defender, Final Attack Maximum minTTA

3.1.9 Slow Defender, Volatility

The magnitude of the volatility does not significantly shift for the slow defenders, shown in Figure 3.9, and the correlation between lower volatility and higher defensive winning percentage is preserved.

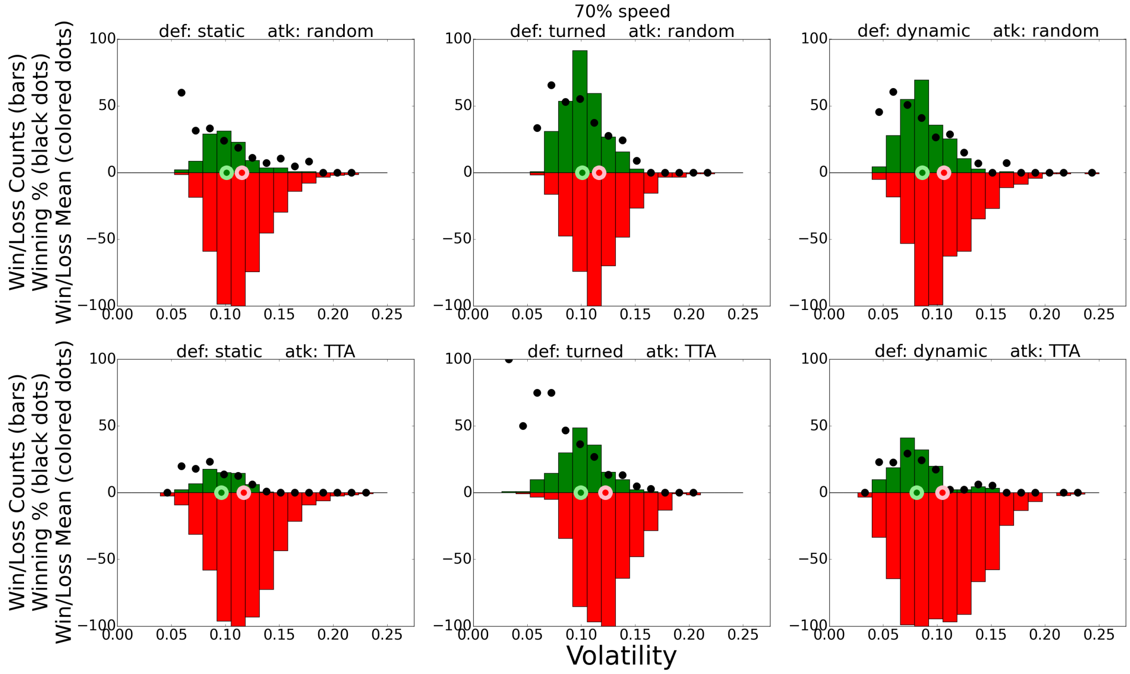


Figure 3.9: Slow Defender, Volatility

3.1.10 Slow Defender, Circularity

Overall, the circularity distributions have shifted to higher values with slow defenders, shown in Figure 3.10. For the dynamic defense histograms, with fast defenders there was a high density around the mean of the distribution, and dynamic defense mitigated the outperformance of TTA-Exploit attackers. With slow defenders, the dynamic defense now has a very wide spread distribution, and the dynamic defense exacerbated the outperformance of TTA-Exploit attackers.

3.1.11 Slow Defender, Final Attack Circularity

The final attack circularity for slow defenders has shifted to higher values as well, shown in Figure 3.11. The slow defender dynamic defense distributions are more spread out compared to the fast defenders.

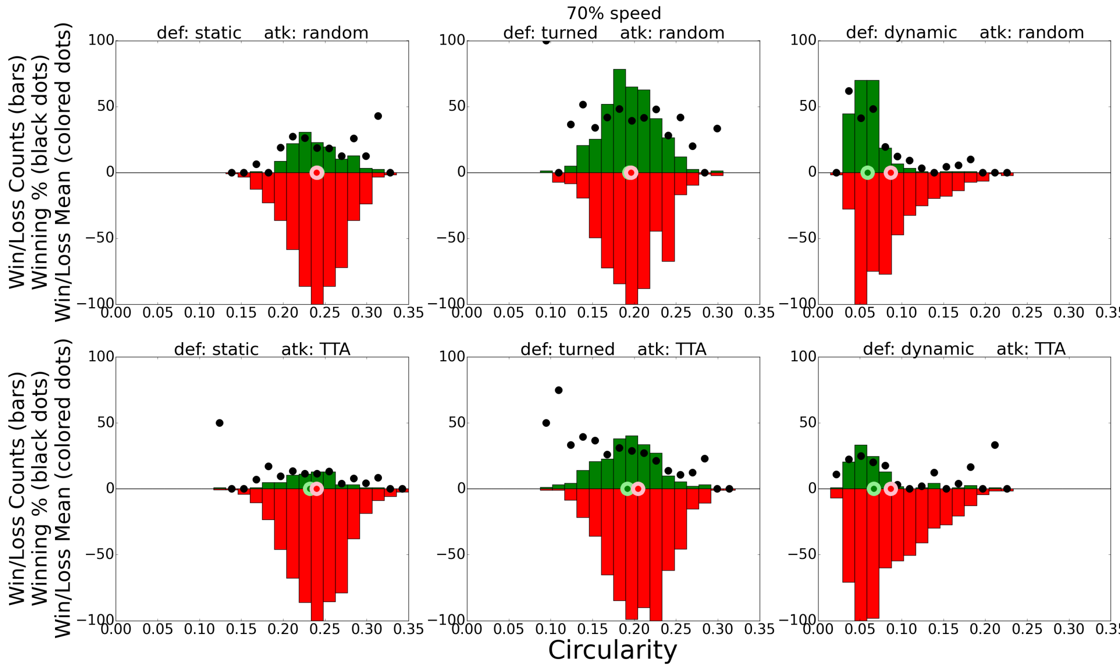


Figure 3.10: Slow Defender, Circularity

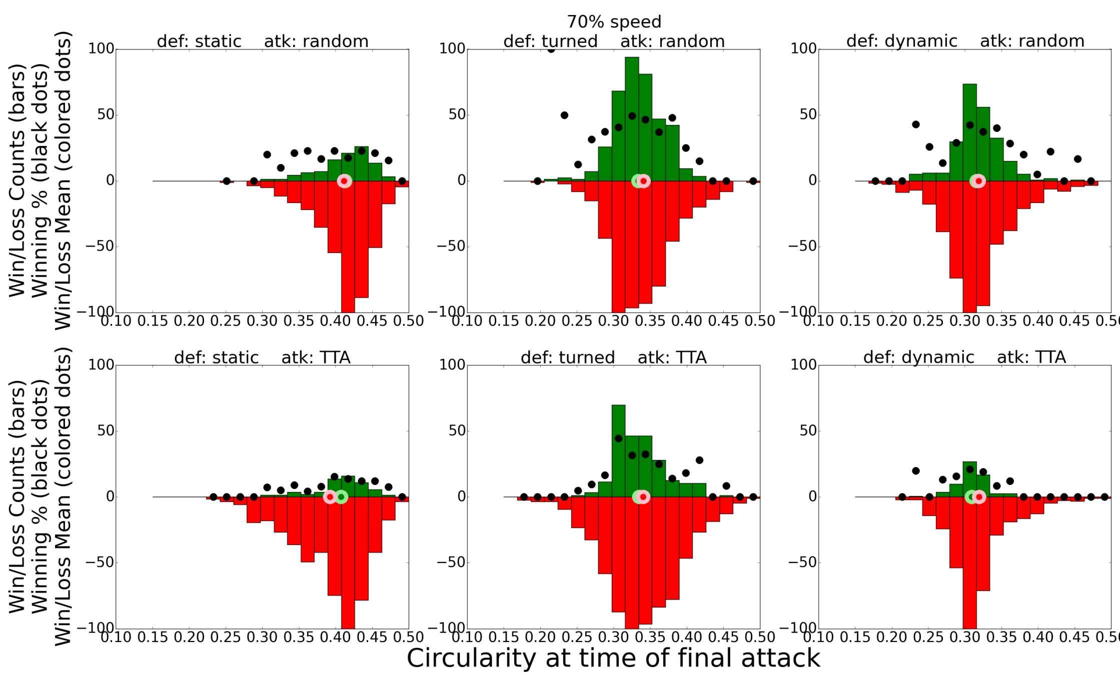


Figure 3.11: Slow Defender, Final Attack Circularity

3.1.12 Defender Count

As expected, adding extra defenders caused a monotonic rise in outcome performance. For fast defenders, having zero extra defenders (4 defenders, 4 attackers) resulted in 44.2% wins, one extra defender (4 defenders, 3 attackers or 5 defenders, 4 attackers) resulted in 56.6% wins, and two extra defenders (5 defenders, 3 attackers) resulted in 68.1% wins. For slow defenders, the same respective winning percentages were 14.0%, 23.8%, and 34.6%.

Chapter 4

Conclusions

4.1 Discussion

This work has constructed a simple simulator for a multi-agent attack and defense scenario involving small boats, and proposed a family of novel metrics to connect run-time characteristics of a defense to outcome performance in the simulated scenario. Time-To-Arrive (TTA) can be used by both attack and defense to delegate tasks amongst agents in a team and execute strategies. TTA can be calculated offline to allow for a low run-time complexity, ensuring that it can be used in real-time. A more detailed TTA metric was created to describe the run-time state of the defense as a whole, fixed-radii minimum TTA (minTTA) rings. The smoothness measures of volatility and circularity were applied to the tensor representation of the minTTA rings. The values of minTTA magnitude, volatility, and circularity from each simulation instance were collected into histograms for analysis.

Several thousand simulation runs with various combinations of attack and defense strategies, in addition to other parameters, were completed and compared. Attackers that exploit TTA information were shown to outperform random attackers, but this outperformance was mitigated by using a dynamic defense rather than a static defense. One caveat for dynamic defense is that its effect is strongly dependent on relative mobility. Dynamic defense provides advantage to defenders that can match their opponents' speed, but this turns into a disadvantage for slower defenders.

Circularity did not demonstrate a strong correlation, though the width of the histogram distribution appeared to be related to the robustness of dynamic defense against attackers exploiting TTA. A tighter distribution for the fast defenders coincided with increased robustness, while a wider distribution for the slow defenders coincided with decreased robustness.

Lower values of maximum minTTA at the defensive perimeter showed a strong correlation with higher defensive winning percentage. Lower values of volatility also showed a strong correlation with higher defensive winning percentage. These two run-time metrics show a connection to outcome performance, and as such, could serve as a basis for future work as a run-time optimization objective function.

4.2 Future Work

Now that lower values of maximum minTTA and volatility run-time metrics have been shown to correlate with effective defense, the next step is to use them as minimization objectives during run-time. In other words, the defensive agents would seek to use behaviors and movement that minimize these metrics during run-time. If this can be successfully demonstrated alongside increased defensive outcome performance, this will serve as further proof that the effectiveness of a defense can be optimized during run-time rather than over the course of a trial-and-error sequence of binary pass/fail tests.

In a mirror of optimization of a defense, a similar optimization could be applied to a team of attackers. Here, the maximum minTTA and volatility would serve as maximiation objectives, and the attackers would attempt to induce behavior and movement in the defenders that would increase these metrics during run-time.

While lower maximum minTTA being correlated with higher defensive effectiveness is expected and intuitive, volatility's similar correlation is not as easy to understand. Future work would explore the meaning behind rapid changes in minTTA and identify details of the behaviors that lead to different levels of minTTA volatility.

The same is true for why low circularity did not display a strong correlation with effective defensive, especially against TTA-Exploit attackers. If exploitative attackers seek out uneven, warped regions of the defense, a lower circularity should indicate additional robustness. This was not observed here, but future work should explore deviation from a defense's nominal (at rest) circularity instead of raw circularity magnitude. The width of the circularity histogram distribution seemed to be related to the varying levels of robustness displayed by the dynamic defense against exploitative attackers, but further investigation is required before conclusions can be made.

Bibliography

[1] David A Anisi, Petter Ögren, and Xiaoming Hu. Cooperative minimum time surveillance with multiple ground vehicles. *Automatic Control, IEEE Transactions on*, 55(12):2679–2691, 2010.

[2] Tucker Balch et al. Learning roles: Behavioral diversity in robot teams. *College of Computing Technical Report GIT-CC-97-12, Georgia Institute of Technology, Atlanta, Georgia*, 73, 1997.

[3] Joydeep Biswas, Juan Pablo Mendoza, Danny Zhu, Benjamin Choi, Steven Klee, and Manuela Veloso. Opponent-driven planning and execution for pass, attack, and defense in a multi-robot soccer team. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 493–500. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[4] Christopher Bolkcom. Homeland security: Unmanned aerial vehicles and border surveillance. DTIC Document, 2004.

[5] E. Borhaug, A. Pavlov, and K. Y. Pettersen. Integral LOS Control for Path Following of Underactuated Marine Surface Vessels in the Presence of Constant Ocean Currents. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4984–4991, Dec 2008.

[6] James Carafano. Small Boats, Big Worries: Thwarting Terrorist Attacks from the Sea. 2007. The Heritage Foundation, No. 2041. http://www.heritage.org/research/reports/2007/06/small-boats-big-worries-thwarting-terrorist-attacks-from-the-sea.

[7] Hoam Chung, Elijah Polak, Johannes O Royset, and Shankar Sastry. On the optimal detection of an underwater intruder in a channel using unmanned underwater vehicles. *Naval Research Logistics (NRL)*, 58(8):804–820, 2011.

[8] Filipe M Clemente, Fernando ML Martins, Micael S Couceiro, Rui S Mendes, and

António J Figueiredo. Developing a tactical metric to estimate the defensive area of soccer teams: The defensive play area. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*, page 1754337115583198, 2015.

[9] João Cunha, Nuno Lau, and João Rodrigues. Ball interception behaviour in robotic soccer. In *RoboCup 2011: Robot Soccer World Cup XV*, pages 114–125. Springer, 2011.

[10] Richard Darilek, Walter Perry, Jerome Bracken, John Gordon, and Brian Nichiporuk. Measures of effectiveness for the information-age army. Technical report, DTIC Document, 2001.

[11] Sze Yi Ding. *On Distributed Strategies in Defense of a High Value Unit (HVU) Against a Swarm Attack*. PhD thesis, Monterey, California. Naval Postgraduate School, 2012.

[12] Yehuda Elmaliach, Noa Agmon, and Gal A Kaminka. Multi-robot area patrol under frequency constraints. *Annals of Mathematics and Artificial Intelligence*, 57(3-4):293–320, 2009.

[13] Pearu Peterson Eric Jones, Travis Oliphant. SciPy: Open source scientific tools for Python, 2001. http://www.scipy.org/.

[14] Jorge Gomes, Pedro Mariano, and Anders Lyhne Christensen. Cooperative coevolution of partially heterogeneous multiagent systems. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 297–305. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[15] David Grabiner. The sabermetric manifesto. *The Baseball Archive*, 1994.

[16] Yi Guo, Lynne E Parker, and Raj Madhavan. Towards collaborative robots for infrastructure security applications. In *Proceedings of International Symposium on Collaborative Technologies and Systems*, volume 2004, pages 235–240, 2004.

[17] Jeong Hee Hong, Kyung-Min Seo, and Tag Gon Kim. Simulation-based optimization for design parameter exploration in hybrid system: a defense system example. *Simulation*, page 0037549712466707, 2013.

[18] George Konidaris and Andrew Barto. Autonomous shaping: Knowledge transfer in reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 489–496. ACM, 2006.

[19] Frederick William Lanchester. *Aircraft in warfare: The dawn of the fourth arm*. Constable limited, 1916.

[20] Platypus LLC. 2010 Smallman St., Pittsburgh, PA 15222. senseplatypus.com.

[21] Eric Martinson and Ronald C Arkin. Learning to role-switch in multi-robot systems. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 2727–2734. IEEE, 2003.

[22] Maja J Matarić. Reinforcement learning in the multi-robot domain. In *Robot colonies*, pages 73–83. Springer, 1997.

[23] Philip Maymin. Acceleration in the NBA: Towards an Algorithmic Taxonomy of Basketball Plays. 2013. http://ssrn.com/abstract=2200560.

[24] JP Mendoza, J Biswas, D Zhu, R Wang, P Cooksey, S Klee, and M Veloso. Cmdragons extended team description paper, 2015.

[25] Nathan Michael, Michael M Zavlanos, Vijay Kumar, and George J Pappas. Distributed multi-robot task assignment and formation control. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 128–133. IEEE, 2008.

[26] Ulrik Nilsson, Petter Ö Gren, and Johan Thunberg. Optimal positioning of surveillance ugvs. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2539–2544. IEEE, 2008.

[27] Tiago Pereira, Manuela Veloso, and António Moreira. Multi-robot planning using robot-dependent reachability maps. In *Robot 2015: Second Iberian Robotics Conference*, pages 189–201. Springer, 2016.

[28] Walter Perry, Robert W Button, Jerome Bracken, Thomas Sullivan, and Jonathan Mitchell. Measurements of effectiveness for the information-age navy: The effects of network-centric operations on combat outcomes. Technical report, DTIC Document, 2002.

[29] Loc V Pham, Brandon Dickerson, James Sanders, Michael Casserly, Vicente Maldonado, Demostenes Balbuena, Stephen Graves, and Bhavisha Pandya. *UAV swarm attack: protection system alternatives for Destroyers.* PhD thesis, Monterey, California: Naval Postgraduate School, 2012.

[30] A. Piazzi, C. Guarino Lo Bianco, and M. Romano. η^3-splines for the smooth path generation of wheeled mobile robots. *IEEE Transactions on Robotics*, 23(5):1089–1095, Oct 2007.

[31] Kyung-Min Seo, Hae Sang Song, Se Jung Kwon, and Tag Gon Kim. Measurement of effectiveness for an anti-torpedo combat system using a discrete event systems

specification-based underwater warfare simulator. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 8(3):157–171, 2011.

[32] Boris Stilman, Vladimir Yakhnis, and Oleg Umansky. Winning strategies for robotic wars: defense applications of linguistic geometry. *Artificial Life and Robotics*, 4(3):148–155, 2000.

[33] Andre Tiwari. Small boat and swarm defense: A gap study. Technical report, DTIC Document, 2008.

[34] Douglas Vail and Manuela Veloso. Multi-robot dynamic role assignment and coordination through shared potential fields. *Multi-robot systems*, pages 87–98, 2003.

[35] Abhinav Valada, Prasanna Velagapudi, Balajee Kannan, Christopher Tomaszewski, George Kantor, and Paul Scerri. Development of a low cost multi-robot autonomous marine surface platform. In *Field and Service Robotics*, pages 643–658. Springer, 2014.

[36] Hongling Wang, Joseph Kearney, and Kendall Atkinson. Robust and Efficient Computation of the Closest Point on a Spline Curve.