Use of Reinforcement Learning for Interference Avoidance or Efficient Jamming in Wireless Communications

Zachary A. Schutz

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Richard Buehrer, Chair

Daniel J. Jakubisin, Co-chair

John Ruohoniemi

May 7, 2024

Blacksburg, Virginia

Keywords: Reinforcement Learning, OFDM, Interference Avoidance, Jamming, Underwater Channel, 5G

Copyright 2024, Zachary A. Schutz

Use of Reinforcement Learning for Interference Avoidance or Efficient Jamming in Wireless Communications

Zachary A. Schutz

ABSTRACT

We implement reinforcement learning in the context of wireless communications in two very different settings. In the first setting, we study the use of reinforcement learning in an underwater acoustic communications network to adapt its transmission frequencies to avoid interference and potential malicious jammers. To that effect, we implement a reinforcement learning algorithm called contextual bandits. The harsh environment of an underwater channel provides a challenging problem. The channel may induce multipath and time delays which lead to time-varying, frequency-selective attenuation. These factors are also influenced by the distance between the transmitter and receiver, the subbands the interference is located within, and the power of the transmitter. We show that the agent is effectively able to avoid frequency bands that have degraded channel quality or that contain interference, both of which are dynamic or time-varying .

In the second setting, we study the use of reinforcement learning to adapt the modulation and power scheme of a jammer seeking to disrupt a wireless communications system. To achieve this, we make use of a linear contextual bandit to learn to jam the victim system. Prior work has shown that with the use of linear bandits, improved convergence is achieved to jam a single-carrier system using time-domain jamming schemes. However, communications systems today typically employ orthogonal frequency division multiplexing (OFDM) to transmit data, particularly in 4G/5G networks. This work explores the use of linear

Thompson Sampling (TS) to jam OFDM-modulated signals. The jammer may select from both time-domain and frequency-domain jamming schemes. We demonstrate that the linear TS algorithm is able to perform better than a traditional reinforcement learning algorithm, upper confidence bound-1 (UCB-1), in terms of maximizing the victim's symbol error rate. We also draw novel insights by observing the action states, to which the reinforcement learning algorithm converges.

We then investigate the design and modification of the context vector in the hope of increasing overall performance of the bandit, such as decreased learning period and increased symbol error rate caused to the victim. This includes running experiments on particular features and examining how the bandit weights the importance of the features in the context vector.

Lastly, we study how to jam an OFDM-modulated signal which employs forward error correction coding. We extend this to leverage reinforcement learning to jam a 5G-based system implementing some aspects of the 5G protocol. This model is then modified to introduce unreliable reward feedback in the form of ACK/NACK observations to the jammer to understand the effect of how imperfect observations of errors can affect the jammer's ability to learn.

We gain insights into the convergence time of the jammer and its ability to jam the victim, as well as improvements to the algorithm, and insights into the vulnerabilities of wireless communications for reinforcement learning based jamming.

Use of Reinforcement Learning for Interference Avoidance or Efficient Jamming in Wireless Communications

Zachary A. Schutz

GENERAL AUDIENCE ABSTRACT

In this thesis we implement a class of reinforcement learning known as contextual bandits in two different applications of communications systems and jamming. In the first setting, we study the use of reinforcement learning in an underwater acoustic communications network to adapt its transmission frequencies to avoid interference and potential malicious jammers. We show that the agent is effectively able to avoid frequency bands that have degraded channel quality or that contain interference, both of which are dynamic or time-varying. In the second setting, we study the use of reinforcement learning to adapt the jamming type, such as using additive white Gaussian noise, and power scheme of a jammer seeking to disrupt a wireless communications system. To achieve this, we make use of a linear contextual bandit which probabilistically models the jammer's observed context features as having a linear relationship with the reward function. We demonstrate that the linear algorithm is able to outperform a traditional reinforcement learning algorithm in terms of maximizing the victim's symbol error rate. We extend this work by examining the impact of the context feature vector design, LTE/5G-based protocol specifics (such as error correction coding), and imperfect reward feedback information. We gain insights into the convergence time of the jammer and its ability to jam the victim, as well as improvements to the algorithm, and insights into the vulnerabilities of wireless communications for reinforcement learning based jamming.

Dedicated to my friends and family.

Acknowledgments

I would first like to thank my parents and brother for supporting me in my journey through undergraduate and graduate school. There were many times I initially wanted to give up throughout my journey to become an engineer, but they always guided me to push myself and not accept defeat, even though I really wanted to. As a fellow engineer, my brother would listen to my gripes and would validate my feelings. I would like to thank the people behind the Department of Defense (DoD) Cyber Scholarship Program (CySP). Without this scholarship, I would not have been able to attend graduate school. The DoD CySP has opened up many opportunities for me and has been a blessing on my life. I would like to thank my committee for guiding me through my academic journey. All have inspired me and guided me at some point during my academic career. I'd like to thank Dr. Jakubisin for giving me a chance to participate in undergraduate research and being my inspiration for attending graduate school. I'd like to thank Dr. Buehrer for being a solid mentor in wireless communications. Both the digital communications class and the advanced digital communications class were both enlightening and testing of my spirit. I'd like to thank Dr. Ruohoniemi for being patient with both myself and my team as the subject matter expert for my senior design group. I'd like to thank Dr. Thornton for his mentorship in reinforcement learning. This work would not exist without his hard work and help in understanding reinforcement learning. For my friends, I'd like to thank Sean for participating in shenanigans with me. I'd like to thank Fiona and John for entertaining Sean and I after returning from the shenanigans. I'd like to thank Chris for putting up with and dealing with Sean and I's shenanigans on a daily basis. Lastly, I'd like to thank Casey for participating in Sean and I's shenanigans and introducing me to the TPB. Watching Casey work and go through the wringer consistently inspired me

to keep working on my thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Reinforcement learning (RL) is used for a wide range of applications from advertising to automation in robots [2, 3]. RL is a machine learning technique that trains an agent to optimize its actions through a trial-and-error process similar to how humans learn. A simple example might be to optimize a wireless communications system in terms of picking a modulation scheme to maximize the throughput. The agent learns by the rewards it receives from taking an action, such as the throughput achieved by selecting the modulation scheme. RL also has many applications to wireless communications and radio access networks. Some use cases for reinforcement learning in wireless communications include radio resource allocation and scheduling, network traffic optimization, power allocation of antennas, and interference avoidance and mitigation [4].

1.1 Motivation

RL can be leveraged to control a communications system's transmission frequency to avoid interference and jamming and be applied to control a jammer seeking to disrupt a communications system. We look at both in this thesis, with an emphasis on the latter. It is important to study the effects of physical layer jamming in order to determine effective strategies. Those strategies can then be used to disrupt unwanted communications that occur between adversaries. Convergence of the jammer to a useful strategy is also impor-

tant to study. The RL relevance to jamming will only provide benefits when the jammer converges to a meaningful solution in the desired time horizons. The studies of these issues often assume the jammer has perfect observations of the errors caused to the victim signal, but in reality, this is not the case. There is a need to study the behavior of the jammer under RL when reward feedback is unreliable.

Jamming has typically been studied in the context of spread spectrum communications systems since military applications use spread spectrum communications, but it has been increasingly important to study the impact of jamming in 5G networks since coexistence between civilian and military applications in the 5G spectrum have grown. Moreover, military applications are expected to use 5G capabilities in the future [5], so it is imperative to study and understand the effect of jamming.

The term wireless communications is not exclusive to terrestrial networks. Underwater acoustic communications (UAC) networks employ acoustic communications to transmit data through the medium of water. To enable the use of an underwater internet of things (IoT), prioritizing resiliency and data rate is of utmost importance to the communications system. Resiliency and data rate become especially important since the underwater channel is known to induce multipath and time delay which introduce time-varying, frequency-selective attenuation to the transmitted signal. Possible interference sources that may affect the signal include non-intentional interference sources, such as passing by ships and animals that use the acoustic communications frequency spectrum, and intentional interference sources such as a malicious jammer seeking to disrupt communications. A jammer or malicious node could introduce a wide-band jamming signal that would introduce additional interference to the frequency spectrum [6]. Another option a jammer could take would be using a singular tone or narrow band frequency to jam a signal that is known to be present in that spectrum [6]. An extension of the tone jammer is a frequency hopping (FH) jammer where the jammer

switches frequency bands for a certain amount of time to effectively jam the transmitter over a wide range of frequencies [6]. A way to combat this is the use of reinforcement learning for subband selection and interference avoidance.

This gives motivation to implement an algorithm to avoid the frequency bands that are heavily occupied by noise, degraded by the channel, or that are occupied by any potential interferers, such as other co-channel systems, or jammers. Large-scale pathloss increases with frequency as shown in Fig. 1.1, which typically makes transmissions at higher frequencies disadvantageous [7, 8]. However, when a jammer or interferer is present, higher pathloss may be advantageous if it affects the jammer or interferer more than the intended transmitter.

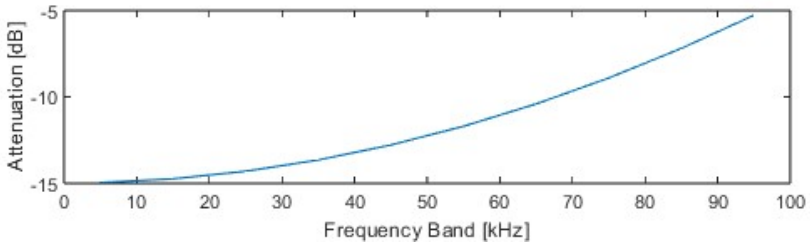

Figure 1.1: Relative channel attenuation as a function of frequency in the underwater channel at a distance of 100m.

1.2 Related Works

1.2.1 Jamming Against a Victim Signal

Prior work has derived the optimal waveforms for jamming orthogonal frequency division multiplexing (OFDM) [1]. However, to apply optimal jamming in practice requires *a priori* knowledge, whether it is coherence with the victim communication system or knowledge about the victim's transmission strategy. In a real-world scenario, the jammer will have little to no knowledge about the victim's transmission. In [9], the authors apply a multi-

armed bandit to learn a jamming strategy for the victim signal without knowledge *a priori*, but the learning scheme becomes cumbersome when the action space becomes large. This is because the bandit algorithm (upper confidence bound-1 (UCB-1)) must try every action at least once and must maintain separate confidence indexes for each arm. In contrast, a linear bandit only need maintain a single confidence index for all actions, leading to faster convergence. In [10], the authors deploy a linear bandit using Thompson Sampling (TS) in order to converge to an effective jamming solution quickly, while outperforming UCB-1 in a large action space. No information is assumed *a priori*, and the jammer is non-coherent with the victim's transmission strategy [10], but the work focuses on jamming single-carrier digital modulation.

In today's communications systems, multi-carrier transmission using OFDM is commonly used as the preferred method of data transmission, particularly in 4G, 5G, and Wi-Fi. In [11], the authors examine the effect of using reinforcement learning to jam a federated learning network in the context of signal classification of a network. They jam the uplink and/or downlink to reduce the accuracy of the trained global model of the network, and use reinforcement learning so the jammer can learn which user equipment (UE) or server is best to attack [11]. Again, this assumes the victim/jamming signals are single-carrier signals modulated with either BPSK or QPSK.

1.2.2 Jamming Avoidance and Subband Selection

Reinforcement learning has been applied to waveform optimization and subband selection in a variety of wireless transmission applications. In the context of radar, bandit algorithms have been studied for selecting a waveform from a series of linear frequency modulated (LFM) chirp waveforms and the center frequency and bandwidth of the chirp [12]. Thornton

et al. [12] shows that radar performance is enhanced when Thompson Sampling (TS) and exponential-weight algorithms are used to decide which waveforms, center frequency, and bandwidth to use based on feedback provided from spectrum observations and returns from a collocated receiver. Relatedly, Thornton *et al.* [13] showed that deep RL can also be used for varying the bandwidth and center frequency of the LFM waveform to mitigate interference from other systems and improve overall radar performance.

RL has also been studied in the context of Industry 4.0 Network for the purpose of spectrum sharing [14]. Bajrachrya and Jung [14] implement a TS approach of channel selection in which the arms of the contextual bandit problem can increase or decrease over time. Jones [15] modeled an environment in GNU Radio to implement a Q-learning approach to an OFDM signal. The subband was chosen by the Q-learning algorithm by feedback from the receiver that included the reward and sensing of the interferer band occupancy [15]. The work explored limitations of real-time implementations of RL and the feasibility for use in a real-world scenario [15].

In the context of UAC, Q-learning has been applied to optimization of energy efficient routing [16] to improve network lifetime, throughput, and energy consumption. However, to the best of our knowledge, RL applied to band of operation selection in UAC, where the channel conditions are highly frequency- and distance-dependent, has not been explored in prior work.

1.3 Contributions

Chapter 2 gives an overview of RL. We delve into the specifics of multi-armed bandits and contextual bandits to give an understanding on the applications of these different types of RL. We also cover the types of RL algorithms that are used in this thesis to provide a

foundation on their mechanisms to help understand the work in this thesis.

Chapter 3 introduces the use of RL in an underwater channel for interference avoidance. We introduce multiple RL algorithms to apply to avoiding interference and observe the signal-to-interference and noise ratio (SINR) and the percentage of the time the legitimate transmitter/receiver (Tx/Rx) pair is able to avoid the bands that are insufficient for transmission due to either degradation in the underwater channel or interference from external sources. We first apply a deterministic frequency hopping (FH) pattern to a jammer, and then apply a non-deterministic FH pattern by using a Markov decision process. Both jamming cases were initially examined with a deterministic channel model, and then an auto-regressive-1 (AR-1) process is introduced to account for the impact of multipath fading on the channel gain. We show jammer collision avoidance and successful transmission rates are increased compared to non-learning and random approaches to subband selection.

The outcome of this work has resulted in the publication of a research paper titled

- Contextual Bandits: Band of Operation Selection in Underwater Acoustic Communications. This work was accepted for publication at 2023 IEEE ICC Workshops [17].

Chapter 4 expands upon previous work of Thornton and Buehrer of using a linear RL algorithm for jamming a single-carrier communications system [10]. We now explore jamming a victim communications system employing OFDM. The bandit employs a linear algorithm, linear TS, to observe the symbol error rate (SER) achieved from this algorithm. The UCB-1 algorithm is also employed by the bandit to create a baseline of comparison against the performance of linear TS. The bandit has the ability to employ either single-carrier or multi-carrier waveforms to jam the victim signal. The bandit also has the ability to implement pulse jamming. The average power of the jammer is seen as constant at the victim, but the instantaneous power is modified by setting the pulse rate. We first explore the effect

of ordering of the action space to observe how this influences how the jammer learns and the effectiveness of the jammer in terms of SER achieved. We then explore the difference between linear TS and UCB-1 in terms of convergence rate and SER achieved. The SER of each of the learning algorithms is observed to show how effective linear TS is compared to UCB-1. We observe similar convergence results to those achieved in [10] but in the context of jamming an OFDM-modulated signal. We also confirm the bandit chooses the optimal schemes which were derived from [1], while drawing novel insights.

The outcome of this work has resulted in the publication of a research paper titled

- Linear Jamming Bandits: Learning to Jam OFDM-Modulated Signals. This work is accepted for publication at 2024 IEEE ICC [18].

Chapter 5 explores the design of the context vector to include or exclude certain features in order to improve performance in terms of convergence time, SER performance, or both. We use the context vector used in Chapter 4 as a baseline to observe how including or excluding certain features changes performance. Through this analysis we draw insights on the use of different context vectors for different transmission strategies. The definition of "success" or "frequency of success" for the bandit is then explored and whether this parameter significantly influences bandit convergence and SER caused to the victim. These results show the importance of choosing a numerically valid success value. Lastly, we explore the weights the bandit puts on each of the context features to explore how important the bandit weights a feature and how the weight changes depending on the context vector used. These results help support how one context vector might outperform another context vector, and the importance of selecting a numerically valid success value.

Chapter 6 expands upon the work presented in Chapter 4 by exploring jamming a system employing forward error correction (FEC) codes. We first explore the effect the pulse rate

has on an OFDM signal employing FEC to understand how spreading jamming power across codewords affects the block error rate (BLER) of the victim signal. We also explore different dimensions in which pulsed jamming can be applied such as pulsed jamming across subcarriers, across OFDM symbols, or both to see how targeting certain properties of an OFDM signal can affect the BLER. We draw insights on how different jamming methods may produce very different results in terms of block error rate. We then use this knowledge to apply a linear TS algorithm to jam a 5G-based system. The jammer has available options such as choice of modulation scheme, a power scaling/pulsing parameter, and targeting the victim signal to jam the physical downlink shared channel (PDSCH) data, demodulation reference signal (DMRS), or both. We observe the BLER achieved and the choices the bandit makes in order to effectively jam the system. We lastly implement unreliable reward feedback to characterize the impact of imperfect reward information on the bandit algorithm. We draw novel insights for the choices the jammer makes under these constraints.

Finally, in Chapter 7, the conclusion and future work are discussed.

Chapter 2

Overview of Reinforcement Learning

Reinforcement Learning (RL) is a subset of machine learning that trains an agent to optimize a problem through a trial-and-error process similar to that of what humans undergo. The agent is not told what actions to take, but must learn through trying actions and observing the numerical reward that is achieved through taking that action. There is a balance between "exploration" and "exploitation" that the agent must adhere to in order to fully understand the complexities of the underlying system and environment. The agent must explore actions over time to understand the environment, but does not want to explore too much because it is trying to maximize an overall numerical reward for the entire period of time. Fig. 2.1 demonstrates a simple RL feedback loop. RL has use cases in many fields such as robotics and advertising. In the advertising case, the agent already knows some information about the problem, such as, the demographic or age group for whom the advertisements are targeted towards. We refer to this information as "context" information. What makes using RL so interesting for wireless communications is there is little to no known prior context information about the problem. The agent has to learn this context information concurrently with the reward information as it explores actions.

An alternative option to RL that is used commonly in wireless communications are simple adaptive techniques. One example use case for interference avoidance and mitigation using a simple adaptive technique is a sense-and-avoid algorithm that senses the frequency spectrum and chooses a subband that has low interference. It was demonstrated by Thornton

and Buehrer in the case of radar performance, in highly stochastic environments learning approaches generally perform better than simple adaptive techniques [19]. Since the topics we cover in this thesis involve highly stochastic environments, we apply RL techniques instead of simple adaptive techniques for better performance.

We extend this use case in RL in communications systems to either avoid interference or cause interference to a legitimate system.

Figure 2.1: Diagram of a simple RL feedback loop.

2.1 Introduction to Contextual Multi-Armed Bandits

A multi-armed bandit is one such where the agent must repeatedly make decisions given a set of choices, or arms. *A priori*, the agent does not know the properties of each arm. The term "bandit" comes from the "one-armed" bandit machines in casinos. Each machine has a different probability of winning, and you don't know which machine will provide the best payout. Therefore, you want to explore the arms to observe the expected outcome of each machine, but you also want to exploit the machine that has the best payout over time.

This introduces the balance between exploration and exploitation. Given the advertising example previously discussed, example properties of an arm might be the number of clicks an advertisement might produce on a website. This can also be called the reward observed from the agent choosing to display that advertisement. At every round t, the bandit observes some information called a context that may be deterministic or random. This context helps the bandit make decisions based on the quality of the context information. After observing that context, the bandit may choose an exploring action or exploiting action. An exploring action is one where the bandit seeks to gain new information, where as an exploiting action is one where the bandit seeks to take advantage of past learned information. Over each iteration, the agent learns what actions are best to exploit the environment and return substantial rewards. It is important to balance exploring each arm to learn more about the reward distribution for each arm, and then using that knowledge to exploit the environment for the long term.

There are trade-offs between exploitation and exploration. If the agent begins to exploit the bandit quickly, the trade-off may be that the exploration does not find enough information about each arm and how the environment acts. The end result will lead to poor reward feedback and poor exploitation. If the bandit takes too long to learn, the agent may not exploit enough over a long term to take advantage of the environment. This is reflected in the agent selecting greedy and non-greedy actions to either exploit or learn about the system.

In the following subsections, we review the RL algorithms we used in this work.

2.1.1 Upper Confidence Bound Algorithm

The upper confidence bound (UCB) algorithm is a deterministic algorithm known for its optimism in the face of uncertainty. The algorithm allows for a balance between greedy and non-greedy actions. In its initial states, the agent will choose non-greedy actions until it learns enough about the environment to confidently exploit the environment with the choices it has available to it.

In every algorithm going forward, there is a value either calculated or sampled from a distribution that is associated with every action. That value represents the benefit received by the agent due to taking that action whether to explore or exploit the environment. In the UCB algorithm, that value is calculated using Eqn. 2.1 from Sutton and Barto [20, p. 32-36],

$$Q_{i(t+1)} = Q_{i(t)} + \frac{1}{t}\left[R_t - Q_{i(t)}\right] \tag{2.1}$$

where $Q_{i(t+1)}$ is the next estimated value of arm i at time step $t+1$, $Q_{i(t)}$ is the current value of arm i at time step t, and R_t is the observed reward at time step t. The term $\left[R_t - Q_{i(t)}\right]$ is an error in the estimate of $Q_{i(t+1)}$ [20, p. 31-32]. Every time an action is taken, the error term is reduced by coming closer to R_t [20, p. 31-32]. If the error is large, it is likely the bandit will choose to explore to gain additional information. Theoretically, as t approaches infinity, $Q_{i(t+1)} = Q_{i(t)}$. Therefore, the estimated value of the next action is proven as the value previously calculated.

The agent must also choose which action to play which is determined based on Eqn. 2.2 from Sutton and Barto [20, p. 32-36],

$$i_{t+1} = \arg\max_{i_{t+1}} \left[Q_{i(t+1)} + c * \sqrt{\frac{\ln t}{N_{t+1}(i_{t+1})}} \right], \tag{2.2}$$

where i_{t+1} is the next arm chosen to be played, $c > 0$ determines the degree of exploration, and $N_{t+1}(i_{t+1})$ is the number of times that arm has been played. If $c = 1$, then that denotes UCB-1. If $N_{t+1}(i_{t+1}) = 0$, then the next action is considered to be a maximizing action and will thus be chosen for the agent to play on the next time step [20, p. 36]. The square root term adds uncertainty to the action by adding variance to the estimated value of the action. If t is large and $N_{t+1}(i_{t+1})$ is small, this lets the agent know that the action has not been chosen enough and that it should be chosen by adding a large amount of uncertainty to the estimated value of the next action. In this case, if the arm has repeatedly returned subpar rewards the few times it has been chosen, it still may not be chosen because the estimated value of the next action is very small in comparison to the values of the other actions available. If t is large and $N_{t+1}(i_{t+1})$ is large, the calculated uncertainty will be small. This lets the agent know that this action is either trustworthy or untrustworthy depending on the estimated value of the next action. Trustworthy in this context means that the action will most certainly provide large rewards, and untrustworthy means the action will most certainly provide rewards that are low or undesirable. The agent will then choose the maximum of these calculations to determine the next action to take.

This leads to the agent allowing for exploration and exploitation in the system to fully understand the choices available to take advantage of the environment present. A sample algorithm to implement the UCB-1 algorithm is shown in Alg. 1.

Algorithm 1: UCB-1 Algorithm

Input $Q_{i(t)} = 0$, $c = 1$
for *Each time step* $t = 1, ..., T$ **do**
  (1) Play arm i_t selected using Eqn. 2.2.
  (2) Observe R_t.
  (3) Update $Q_{i(t)}$ using Eqn. 2.1.
  (4) Increment $N_{t+1}(i_{(t)})$
end

2.1.2 Thompson Sampling Algorithm

Thompson Sampling (TS) is also an algorithm known for its optimism in the face of uncertainty but is a non-deterministic algorithm. The arms are modeled using probability distributions that are updated after an action is taken. The updates are performed using Bayesian statistics using a prior distribution as follows to calculate a posterior distribution,

$$\Pi(\theta|X) \propto L(X|\theta)\Pi(\theta) \tag{2.3}$$

where $\Pi(\theta|X)$ is the posterior distribution, $L(X|\theta)$ is the likelihood function, and $\Pi(\theta)$ is the prior distribution on θ [21]. Our interest is in θ after observing or collecting observations on X, where θ is the sampled values of the arms after collecting observations on the environment X. The exploration in TS comes from the randomisation in the posterior probability density functions [22, p. 459]. If the posterior is not concentrated, the agent is more likely to explore the action space [22, p. 459]. As the posterior becomes more concentrated after data is collected, the agent is more likely to take actions that exploit the environment [22, p. 459]. In this respect, θ is very similar to $Q_{i(t)}$ in UCB where the posterior distribution is the Bayesian approach of accounting for uncertainty, while Eqns. 2.1 and 2.2 represent the frequentist approach to uncertainty in the actions selected by the bandit. Many distributions can be used for performing updates in TS, but the distributions we care about for this thesis are the Beta and Normal distributions.

TS: Beta Distribution

The Beta distribution is often used in TS to model the space as Bernoulli trials, i.e., successes and failures. A success is defined as the bandit choosing an action that meets a defined

threshold or has a positive effect on the system. In our case, this may be effectively avoiding interference or successfully causing interference on a victim Tx/Rx pair. A failure is defined as not meeting that threshold or not having an effect or enough of an effect in order to consider it a success. The likelihood function is defined in Eqn. 2.4,

$$L(X|\theta) \propto \theta^{\sum_{i=1}^{N} x_i} * (1-\theta)^{N-\sum_{i=1}^{N} x_i} \tag{2.4}$$

where this represents a Bernoulli probability mass function with parameter θ. The prior Beta distribution is defined as Eqn.2.5,

$$\theta_{i(t)} \sim B(\alpha_{i(t)}, \beta_{i(t)}) \tag{2.5}$$

where $\theta_{i(t)}$ is the sampled value from the Beta distribution of the ith arm at time step t, $\alpha_{i(t)}$ is the number of successes at time step t of the ith arm, and $\beta_{i(t)}$ is the number of failures at time step t of the ith arm [21]. After an action, the $\alpha_{i(t)}$ and $\beta_{i(t)}$ of that arm is updated to properly reflect the posterior distribution and knowledge gained after having taken that action as $\alpha_{i(t+1)}$ and $\beta_{i(t+1)}$. If $\alpha_{i(t)}$ and $\beta_{i(t)}$ are initialized to 1 to reflect a uniform distribution $U(0,1)$, and a success occurs due to taking action i, the posterior for action i will be reflected as follows,

$$\theta_{i(t+1)} \sim B(\alpha_{i(t+1)} = 2, \beta_{i(t+1)} = 1). \tag{2.6}$$

Similarly, if a failure occurs instead,

$$\theta_{i(t+1)} \sim B(\alpha_{i(t+1)} = 1, \beta_{i(t+1)} = 2). \tag{2.7}$$

A more general form of the posterior update is reflected in Eqn. 2.8,

$$\theta_{i(t+1)} \sim \beta(\alpha_{i(t+1)}, \beta_{i(t+1)}) \tag{2.8}$$

where,

$$\alpha_{i(t+1)}, \beta_{i(t+1)} = \begin{cases} \alpha_{i(t+1)} = \alpha_{i(t)} + 1, \ \beta_{i(t+1)} = \beta_{i(t)} \text{ if } R_{i(t)} = 1 \\ \alpha_{i(t+1)} = \alpha_{i(t)}, \ \beta_{i(t+1)} = \beta_{i(t)} + 1 \text{ if } R_{i(t)} = 0 \end{cases} \tag{2.9}$$

where $R_{i(t)}$ is the reward received by the ith arm at time step t [21].

After all arms are sampled, the arm that provides the highest value is utilized for the next time step, i.e.,

$$a_{(t+1)} = \max_i \theta_{i(t+1)} \tag{2.10}$$

where $a_{(t+1)}$ is the next action chosen [21]. An example algorithm is shown in Alg. 2 [21].

Algorithm 2: TS: Beta Distribution Algorithm

Input $\alpha_i, \beta_i = 1$
for *Each time step* $t = 1, ..., T$ **do**
  (1) Sample $\theta_{i(t)} \sim B(\alpha_{i(t-1)}, \beta_{i(t-1)})$.
  (2) Play arm $i(t) = \max_i \theta_{i(t)}$.
  (3) Observe R_t and update $\alpha_{i(t)}, \beta_{i(t)}$ according to Eqns. 2.8 and 2.9
end

TS: Normal Distribution

Similarly, the Normal distribution can also be used in TS. The updates are more complex than that of the Beta distribution when using TS. A Normal distribution is most useful when the rewards are no longer equal to 0 or 1. The rewards now have scaling to them that will shift the mean of the distribution depending on the reward returned. The shift of the distribution represents a shift in the average reward returned to the bandit on a particular

choice. A large mean represents a particular arm on average returning a large reward while a small mean represents a small reward being returned for that particular action. The estimate mean and variance of the prior distribution can be defined in Eqns. 2.11 and 2.12 [21],

$$\hat{\mu}_{i(t)} = \frac{\sum_{w=1:i(w)=i}^{t-1} R_{i(t)}}{N_i(t-1)+1} \tag{2.11}$$

$$\hat{\sigma}^2_{i(t)} = \frac{1}{N_i(t-1)+1} \tag{2.12}$$

where $\hat{\mu}_{i(t)}$ is the estimated mean of the prior distribution for arm i at time step t, $\hat{\sigma}^2_{i(t)}$ is the estimated variance of the prior distribution for arm i at time step t, $R_{i(t)}$ is the reward for arm i at time step t, and $N_i(t-1)$ is the total number of times arm i has been played up to time step $t-1$. Both the likelihood and prior will both be defined as the Normal distribution. The prior distribution is given as,

$$\mathcal{N}(\hat{\mu}_{i(t)}, \hat{\sigma}^2_{i(t)}). \tag{2.13}$$

The posterior can be found simply by updating the parameters $\hat{\mu}_{i(t)}$ and $\hat{\sigma}^2_{i(t)}$ [21],

$$\hat{\mu}_{i(t+1)} = \frac{\hat{\mu}_{i(t)} * N_i(t-1) + R_{i(t)}}{N_i(t)+1} \tag{2.14}$$

$$\hat{\sigma}^2_{i(t+1)} = \frac{1}{N_i(t)+1}. \tag{2.15}$$

The posterior is constructed by,

$$\mathcal{N}(\hat{\mu}_{i(t+1)}, \hat{\sigma}^2_{i(t+1)}). \tag{2.16}$$

An example algorithm is shown below in Alg. 3 [21],

Algorithm 3: TS: Normal Distribution Algorithm

Input $\hat{\mu}_i, N_i = 0$
for *Each time step* $t = 1, ..., T$ **do**
  (1) Sample $\theta_{i(t)} \sim \mathcal{N}(\hat{\mu}_{i(t-1)}, \frac{1}{N_{i(t-1)}+1})$.
  (2) Play arm $i(t) = \max_i \theta_{i(t)}$.
  (3) Observe R_t and update $\hat{\mu}_{i(t)} = \frac{\hat{\mu}_{i(t-1)} * N_{i(t-1)} + R_t}{N_{i(t)}+1}$, $N_{i(t)} = N_{i(t-1)} + 1$.
end

2.1.3 Linear Contextual Bandits

When an action space becomes large, it adds extra computational complexities to the problem since each arm's properties must be separately accounted for. This leads to large computations that slows down the processing decision time for the bandit since in conventional TS, the estimation for the reward distribution for each arm must be tracked and updated. In order to help alleviate this, linear bandits are used to assume a linear relationship between properties of the arms and reward or cost function. In other words, as Lattimore and Szepesvári state [22, p. 237], the rewards are assumed to have a linear relationship that allows learning to translate from one context to another.

From Lattimore and Szepesvári [22, p. 237-240], at each time step t, the bandit observes a context $C_t \in \mathcal{C}$. Based on context C_t and the past contexts observed, the bandit will play action $A_t \in [k]$. The reward received, R_t, satisfies

$$R_t = r(C_t, A_t) + \eta_t \tag{2.17}$$

where $r : \mathcal{C} \times [k] \to \mathbb{R}$ is the reward function and η is the noise where it is conditionally subgaussian [22, p. 237-240].

The vehicle through which the bandit or agent observes the environment and the consequences of the actions taken is the context vector, $\phi(c, a) : \mathcal{C} \times [k] \to \mathbb{R}^d$, where c is the

set of contexts that have been observed and a are the action(s) taken when those context(s) were observed [22, p. 237-240]. The context vector consists of context features that account for certain observations about the agent's actions, such as the average reward of that action. The parameter or weighting vector, $\theta \in \mathbb{R}^d$, is unknown to the bandit. It then holds that

$$r(c, a) = \langle \phi(c, a), \theta \rangle, \ \forall \ (c, a) \in C \times [k][22, p.237 - 240]. \tag{2.18}$$

Together, the context vector and weighting vector help the bandit determine a linear relationship between similar actions and contexts and the rewards the chosen actions provide. However, sub-optimal actions still need to be taken in order to gain information about those actions. Until an action has been explored, there will be unknowns about the context features and the results of choosing that action.

Linear Thompson Sampling

In particular, we are most interested in linear TS. The weighting vector θ, is now sampled from a posterior distribution every round. The weighting vector can be interpreted as the sampled statistics from the distribution we are using to determine how the bandit weights the importance of each context feature. A sample algorithm from Lattimore and Szepesvári is shown in Alg.4 [22, p. 465].

Algorithm 4: Linear Thompson Sampling Algorithm

for *Each time step* $t = 1, ..., T$ **do**
  (1) Sample from posterior: $\theta_{(t)} \sim \Pi(.|.)$.
  (2) Play arm $a(t) = \max_{a \in \mathcal{A}} \langle \theta_{(t)}, a \rangle$.
  (3) Observe R_t and update.
end

The algorithms in Sections 2.1.1, 2.1.2, and 2.1.3 may be adapted in future chapters to

accommodate the problem space.

Chapter 3

Use of Reinforcement Learning for Interference Avoidance in Underwater Communications

3.1 Introduction

In this chapter, we examine the performance of using reinforcement learning to avoid subbands that are being interfered with in an underwater communications link. We initially compare random frequency hopping (FH) to the learning algorithms in a deterministic channel. The performance of convergence and attainable signal-to-interference-and-noise ratio (SINR) is analyzed as well as the overlap between when the transmitter and receiver decide to use a subband that has severe interference in it. The jammer will first implement a deterministic frequency hopping pattern that is generated pseudo-randomly and then jammer hops will be determined by a Markov process. We then introduce an autoregressive-1 (AR-1) process to the deterministic channel model to emulate a stochastic channel that is time-varying and frequency selective. The performance of the reinforcement learning agent is then analyzed with the stochastic channel model in consideration.

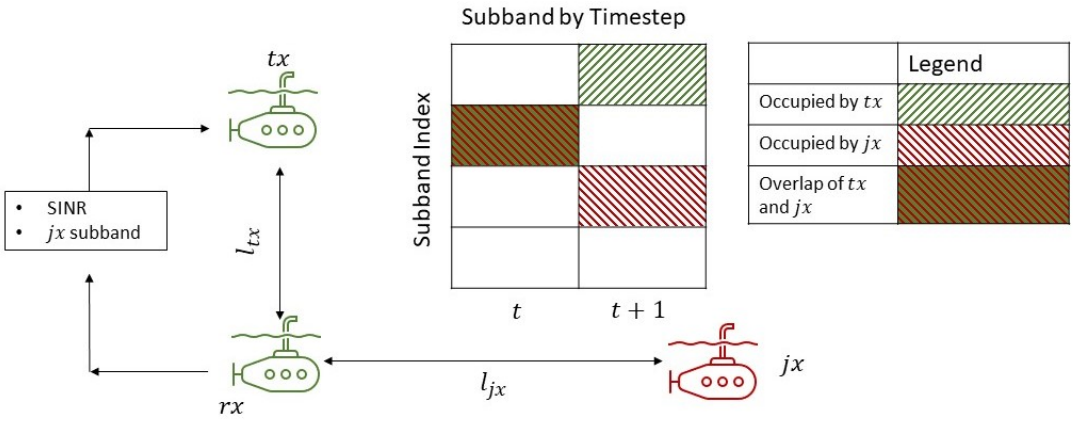

Figure 3.1: Overview of system diagram.

3.2 System Model

3.2.1 Channel Environment

In this work, we focus on modeling the large-scale channel conditions that impact average SINR. For this case, we chose to model the pathloss as follows [7]:

$$A(l, f) = l^k a(f)^l, \tag{3.1}$$

where $A(l, f)$ is the attenuation of the signal due to distance, l, in km and frequency, f, in kHz of the transmission. The term k represents the spreading factor of the signal in the channel where $k = 1$ models cylindrical spreading typically found in shallow water settings, and $k = 2$ models spherical spreading typically found in deep water settings. For the current use case, k was set to 1.5. The term $a(f)$ is the absorption coefficient and depends on frequency as follows [7]:

$$10 \log a(f) = 0.11 \frac{f^2}{1 + f^2} + 44 \frac{f^2}{4100 + f^2} + 2.75 * 10^{-4} f^2 + 0.003. \tag{3.2}$$

As frequency increases with a stationary distance between transmitter and receiver, the attenuation will increase on an exponential scale, as shown in Fig. 3.2.

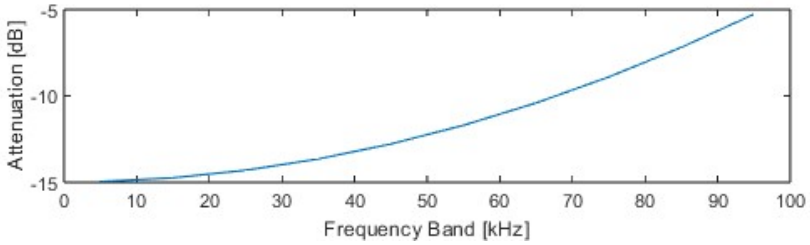

Figure 3.2: Relative channel attenuation as a function of frequency in the underwater channel at a distance of 100m.

In order to account for the impact of multipath fading on the channel gain, a stochastic term Δ is introduced into the path loss equation as follows:

$$A(l, f) = l^k a(f)^l * 10^{\Delta(t)/10}. \tag{3.3}$$

The stochastic term in the $t+1$ time step is computed using an AR-1 process as follows [23]

$$\Delta(t+1) = \Delta(t)\rho + (1-\rho)w(t), \tag{3.4}$$

where ρ is the correlation coefficient per-step, and $w(t)$ is a zero-mean Gaussian random process. We assume the fading in each frequency band is independent.

Following the work of [7, 8], we approximate the noise spectrum as follows:

$$10 \log N(f) \approx 50 - 18 log f. \tag{3.5}$$

This noise term is additive and frequency dependent and decreases at an exponential rate. A more rigorous additive noise term of the ambient noise can be constructed from the thermal, shipping, waves, and turbulence noise terms defined in [7, 8].

3.2.2 Communication Model

We model the transmitter and jammer channel occupancy as a discrete choice of transmission subband over the range of 0-100kHz, as shown in Table 3.1.

Table 3.1: Choices of Frequency Bands.

Choice Number	Frequency Band (kHz)	Center Frequency (kHz)
1	0-10	5
2	10-20	15
3	20-30	25
4	30-40	35
5	40-50	45
6	50-60	55
7	60-70	65
8	70-80	75
9	80-90	85
10	90-100	95

When the transmitter and jammer are located in the same subband, the SINR at the receiver with a jammer present can be calculated from:

$$SINR = \frac{\frac{P_{tx}}{A(l_{tx},f_c)}}{N(f_c) * B + \frac{P_{jx}}{A(l_{jx},f_c)}} \tag{3.6}$$

where the terms f_c correspond to the center frequency of the subband, B corresponds to the bandwidth, P_{jx} and P_{tx} correspond to the power of the jammer and power of the transmitter, respectively, and l_{jx} and l_{tx} correspond to the distance from jammer to receiver and transmitter to receiver, respectively.

For a jammer not located in the same subband as the transmitter, the SINR can be calculated from:

$$SINR = \frac{P_{tx}}{N(f_c) * B * A(l_{tx}, f_c)}. \tag{3.7}$$

We model deterministic jamming patterns as well as jamming patterns controlled by a

Markov decision process. The starting frequency for the transmitter and jammer is randomly chosen from the frequency bands uniformly as shown in Table 3.1. In the deterministic case, the jammer follows a predefined jamming sequence. In the non-deterministic case, the jammer follows the Markov decision process dictated by a parameter γ that controls the probability of the jammer performing a sequential sweep of the subbands. The parameter γ specifies the probability of the jammer moving to the next subband, while the other states will have an equal probability of occuring. If all states are summed, the total probability will be 1. An example matrix that was used is shown below:

$$\begin{bmatrix} \frac{(1-\gamma)}{M-1} & \gamma & \frac{(1-\gamma)}{M-1} & \dots & \frac{(1-\gamma)}{M-1} \\ \frac{(1-\gamma)}{M-1} & \frac{(1-\gamma)}{M-1} & \gamma & \dots & \frac{(1-\gamma)}{M-1} \\ \dots & \dots & \dots & \dots & \\ \frac{(1-\gamma)}{M-1} & \dots & \dots & \dots & \gamma \\ \gamma & \frac{(1-\gamma)}{M-1} & \dots & \dots & \end{bmatrix}, \tag{3.8}$$

where M is the number of subbands. If $\gamma = 1$, then the pattern would correspond to a deterministic sweep pattern where the jammer would go through the subbands sequentially until it repeats. In both the deterministic and non-deterministic cases, the transmitter and receiver have no prior knowledge of the jamming sequence.

3.3 Learning Problem

At each time step t, the transmitter chooses an action $a_t \in A$, and the jammer chooses an action $j_t \in J$, which are the subbands listed in Table 3.1. The transmitter and jammer will transmit on the subband that their respective actions, a_t and j_t, dictate. The receiver is able to monitor all subbands to observe the state of the jammer. The receiver will then send back

the received SINR and the subband the jammer was located in back to the transmitter. The value of the action just taken, a_t, is calculated based on j_{t-1}. The next action a_{t+1} is then determined, and then the jammer will update its state to j_{t+1} for the $t+1$ time step.

Each action will receive a reward R_t depending on the result of the action. A success occurs when the SINR returned by the receiver is above a defined threshold.

The transmitter will explore each subband for the first time steps of the simulation to gather initial data on whether that subband is a valid choice depending on the SINR that is returned to the system. The rest of the time steps of the simulation are spent with the transmitter exploring under-utilized actions versus exploiting the jammer, channel, and noise to take as many greedy actions as possible to maximize a high SINR for the system while avoiding the occupied frequency subbands. We explored the Upper-Confidence-Bound-1 (UCB-1), a UCB algorithm with specialized rewards, and two implementations of the Thompson Sampling (TS) algorithm to select transmitter actions [10, 24].

Since the previous jammer state is known to the system, this makes the problem a contextual bandit, which is reflected in the equations that follow. The number of possible jammer states is also important because each action, a_t, is determined by the previous state of the jammer, j_{t-1}. If there are 10 possible actions the transmitter can take for one jammer state, it is a 10-armed bandit. Since there are 10 jammer states, the situation becomes one where there are 10 contextual bandits running in parallel, with each bandit being a 10-armed bandit.

3.3.1 UCB-1 Algorithm

UCB-1 was chosen because of its optimism in the face of uncertainty. This algorithm is known for optimizing the choice between greedy and non-greedy actions. In its initial states, the algorithm will choose non-greedy actions to determine which actions are of the best

choice. In this case, the transmitter will spend its initial states choosing all of the subbands to determine which ones are most suitable. Suitability is determined by rewards given to the system in order to influence the decision of the next choice. The estimated value of the next action is:

$$Q_{t+1}(j_{t-1}, a_t) = Q_t(j_{t-1}, a_t) + \frac{1}{t}[R_t(j_{t-1}, a_t) - Q_t(j_{t-1}, a_t)]. \tag{3.9}$$

This will show that as the number of time steps increase, the closer the estimated value of the action gets to the true value of the action. Ideally, we want to obtain the true value of the action in order to exploit the system. The next action is chosen as the one with maximum value, conditioned on the observation of the current jamming state (j_t):

$$a_{t+1} = \arg\max_{a_{t+1}} \left[Q_{t+1}(j_t, a_{t+1}) + \sqrt{\frac{\ln t_{j_t}}{N_{t+1}(j_t, a_{t+1})}} \right], \tag{3.10}$$

where $N_{t+1}(j_t, a_{t+1})$ is the number of times action a has already been chosen in the context of j_t. If $N_{t+1}(j_t, a_{t+1})$ is zero, the algorithm will choose $N_{t+1}(j_t, a_{t+1})$ to be equal to 1, thus forcing it to have already taken at least one action. The term $\ln t_{j_t}$ is the natural log of the total times the jamming band j_t has been chosen by the jammer. Thus, it lets the system converge asymptotically as time steps increase letting the uncertainty of the chosen action decrease. The square root term provides uncertainty to the system, so the greater the uncertainty is to the system, the greater the chance that particular action will be taken in order to explore and find the true value of that action.

A UCB algorithm with specialized rewards in this thesis differentiates from a traditional UCB-1 algorithm with reward values of 0 and 1 for a failure and a success, respectively. A UCB algorithm with specialized rewards for this paper will have a reward system where the reward is a function of the returned SINR. This will be referred as "scaled UCB" from now

on. The reward at time t for scaled UCB is as follows

$$R_t = 20 * (SINR_t - 10), \tag{3.11}$$

where $SINR_t$ is the average SINR returned by the receiver over time step t.

3.3.2 Thompson Sampling Algorithm

TS was also considered for this problem because it has many of the same characteristics as a UCB algorithm, such as optimism in the face of uncertainty, but instead of being a deterministic algorithm, it is a probabilistic algorithm. TS here was modeled as separate Bernoulli Trials, either a success or a failure (reward values 1 and 0, respectively). When the rewards are binary, the Beta distribution is useful, where the mean of the distribution can be given by

$$\mu = \frac{\alpha}{\alpha + \beta} \tag{3.12}$$

where μ is the mean of the distribution, α is the success of the trial, and β is the failure of the trial. Each arm of the bandit will have its own Beta distribution based on its successes and failures, where each distribution is sampled at each time step, and the arm with the highest value sample will be chosen. TS uses prior information similar to how the UCB-1 algorithm uses it above.

To fairly compare scaled UCB and TS, a Gaussian prior distribution was used with the same reward function of scaled UCB as shown in Eqn. 3.11. Similar to using the Beta distribution, the arm with the highest sample will be chosen. The mean and variance of the

prior Gaussian distribution for each arm can be calculated as follows [21]

$$\mu_{i(t)} = \frac{\mu_{i(t-1)} * N_t(j_{t-1}, a_t) + R_t}{N_t(j_{t-1}, a_t) + 1} \tag{3.13}$$

$$\sigma_i^2(t) = \frac{1}{N_t(j_{t-1}, a_t)} \tag{3.14}$$

where $\mu_{i(t)}$ is the mean, and $\sigma_i^2(t)$ is the variance of each of the distributions. Each arm can then be sampled from its respective distribution $\mathcal{N}(\mu_{i(t)}, \sigma_i^2(t))$.

3.4 Simulations

This section examines the overall performance of the algorithms with respect to the number of times the transmitter and jammer landed on the same frequency subband for transmission, the number of successes in each trial, and the average SINR of each trial.

The average success, frequency of collision between transmitter frequency and jammer frequency, and the average SINR were recorded during the simulations. The transmission was considered a success when the SINR was greater than or equal to a threshold of 10 dB.

3.4.1 Frequency Hopping

For the FH jammer, the length of the sequence for the jammer was 10 frequency subbands. After the 10th subband, the jammer would repeat its sequence from the beginning. Each frequency subband in the sequence was uniformly chosen, with each subband choice being independent from each other.

In addition to the Traditional UCB-1, Scaled UCB, and TS techniques discussed above, we also considered non-learning algorithms for comparison. The first is Fixed Frequency Hop-

ping (Fixed FH) in which the transmitting frequencies were acquired by randomly selecting 10 frequencies uniformly to transmit the signal and then cycling through those frequencies regardless of what the jammer does. The second is Random Frequency Hopping (Random FH) in which the transmitting frequency values were randomly selected from Table 3.1 every time step, so the transmitter does not follow a pattern. A summary of the performance of these techniques can be found in Table 3.2.

Table 3.2: Summary of Transmitter Using Different Techniques

Technique	Collision Rate	Success Rate	Avg. SINR
Fixed FH	9.73%	32%	8.32 dB
Random FH	9.87%	63.7%	8.85 dB
Trad. UCB-1	4.81%	83.75%	10.57 dB
TS (Beta)	0.91%	96.53%	11.96 dB
Scaled UCB	0.28%	98.05%	11.51 dB
TS (Gaussian)	0.14%	99.59%	12.19 dB

Overall, the learning algorithms drastically improved performance of the system compared to non-use of the learning algorithms. Over the 200 trial runs, success rate was above 80%, jammer collision rate was below 5%, and the SINR threshold was met for each of the RL algorithms.

Comparing the learning algorithms, the traditional UCB-1 algorithm performed the worst out of all of the learning algorithms in all categories. We found that scaling rewards with the scaled UCB algorithm led to a noticeable improvement in performance, due to the emphasis that this would place on exploitation. Both TS (Beta) and the scaled UCB algorithms had

comparable performance where the scaled UCB algorithm had a slightly better performance in avoiding the jammer and higher success rate. TS (Beta) did have a higher average SINR performance. This is due to TS having a better balance between exploration and exploitation in this case. TS (Gaussian) outperformed all of the other RL algorithms in all categories. The reward function used in this case allowed for better exploitation of the subbands, and even outperformed the scaled UCB algorithm. The TS algorithm would allow for a better resource allocation in terms of transmission power, at the expense of computational power.

The performance of the system was also determined by average SINR over each time step as well as the convergence rate which can be seen in Fig. 3.3.

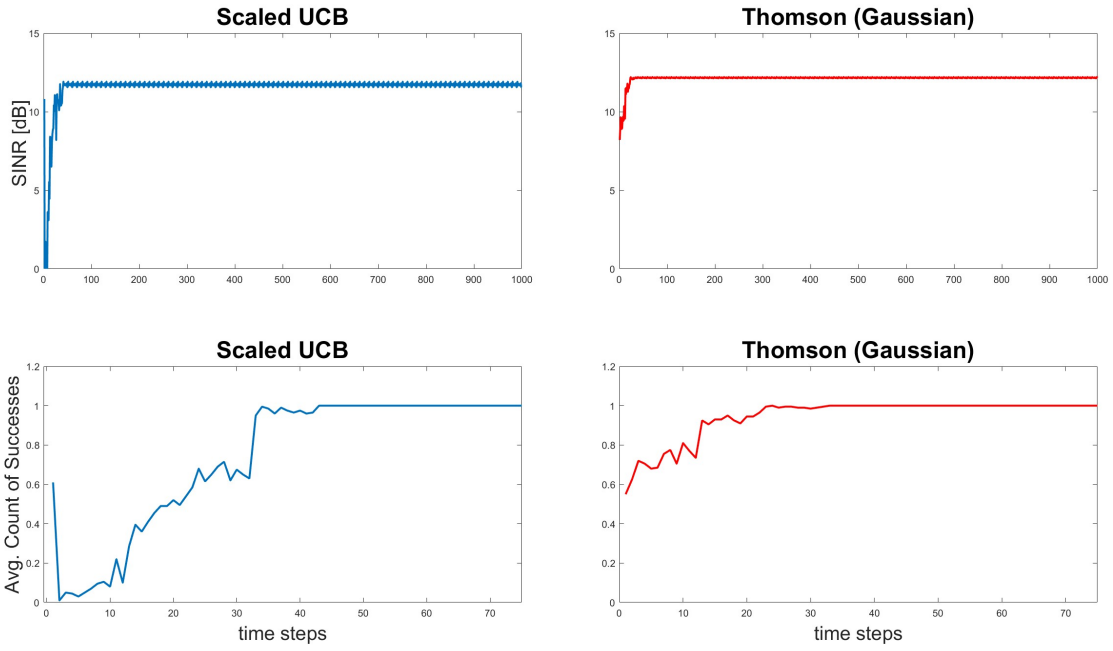

Figure 3.3: Average SINR (top graphs) and average number of successes (bottom graphs) over time step (200 simulations).

The average SINR of the scaled UCB levels out after about the 34th time step, which also follows the convergence graph. The SINR for TS levels out quickly in about 23 time steps, which also follows its respective convergence graph. In this respect, TS has better

performance converging to an SINR above the threshold level faster because of a better exploration and exploitation ratio.

The first 100 times steps and last 100 time steps of individual runs of the scaled UCB and TS algorithms were examined in terms of interference avoidance as shown in Figs. 3.4 and 3.5. A significant shift in behavior is observed after learning has converged.

The first 100 time steps is where both of the algorithms make use of exploring viable choices. Fig. 3.5 shows that TS will not often explore more of the arms than the scaled UCB in the first 100 time steps. TS with a Gaussian prior will immediately try to exploit the environment which leads to faster convergence to a policy. In this case, TS also has a higher overall SINR.

3.4.2 Markov Process

As mentioned in Section 3.2.2, we also model the jammer selection of subbands using a Markov decision process where the probability of picking a subband was controlled by γ. The initial subband for both transmitter and jammer were selected uniformly from the subband choices shown in Table 3.1. Each consecutive jammer subband was then selected with a certain probability depending on the previous state of the jammer.

TS converges to more successes on average faster than the scaled UCB algorithm, but will not have as high as a success rate in late time steps, which can be shown in Fig. 3.6. The scaled UCB algorithm takes 60-70 time steps to converge while the TS algorithm only takes 35-50 steps to converge. Scaled UCB stays consistent even when the probability γ is lowered (increased uncertainty in the jammer behavior). Even though TS has a slightly lower success rate, Fig. 3.6 shows that it will often have a higher SINR than scaled UCB.

Finally, the simulation of Fig. 3.6 was repeated for the stochastic channel model of Eqns. 3.3 and 3.4 as shown in Fig. 3.7. In this case, performance is lower across the board and

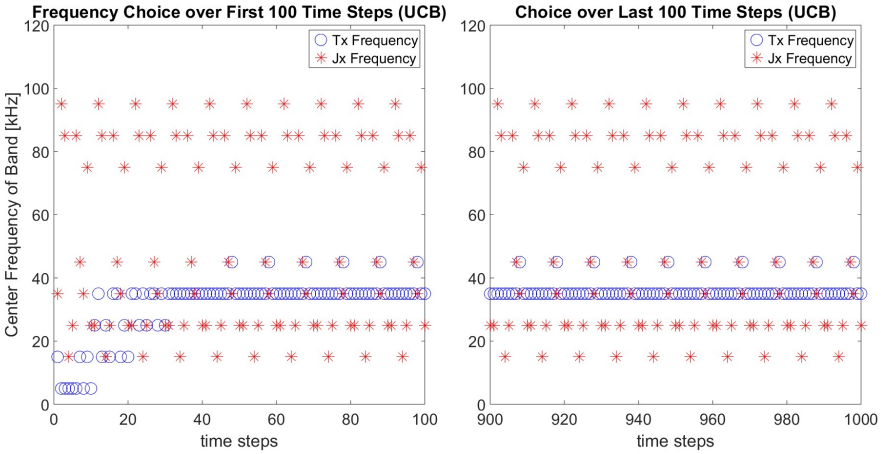

Figure 3.4: First and last 100 time steps of frequency choices of scaled UCB.

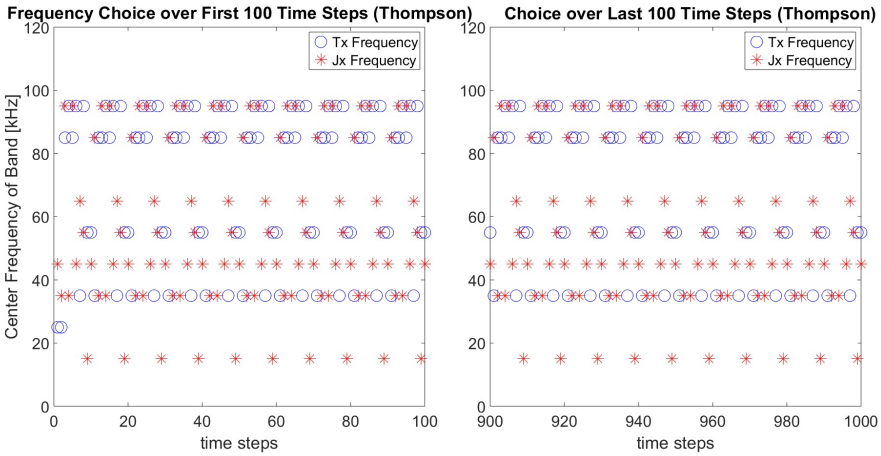

Figure 3.5: First and last 100 time steps of frequency choices of TS.

the learning algorithms are similar in performance to a system which has knowledge of the subband with the best *average* SINR. However, TS shows visible improvement in SINR levels for higher values of γ.

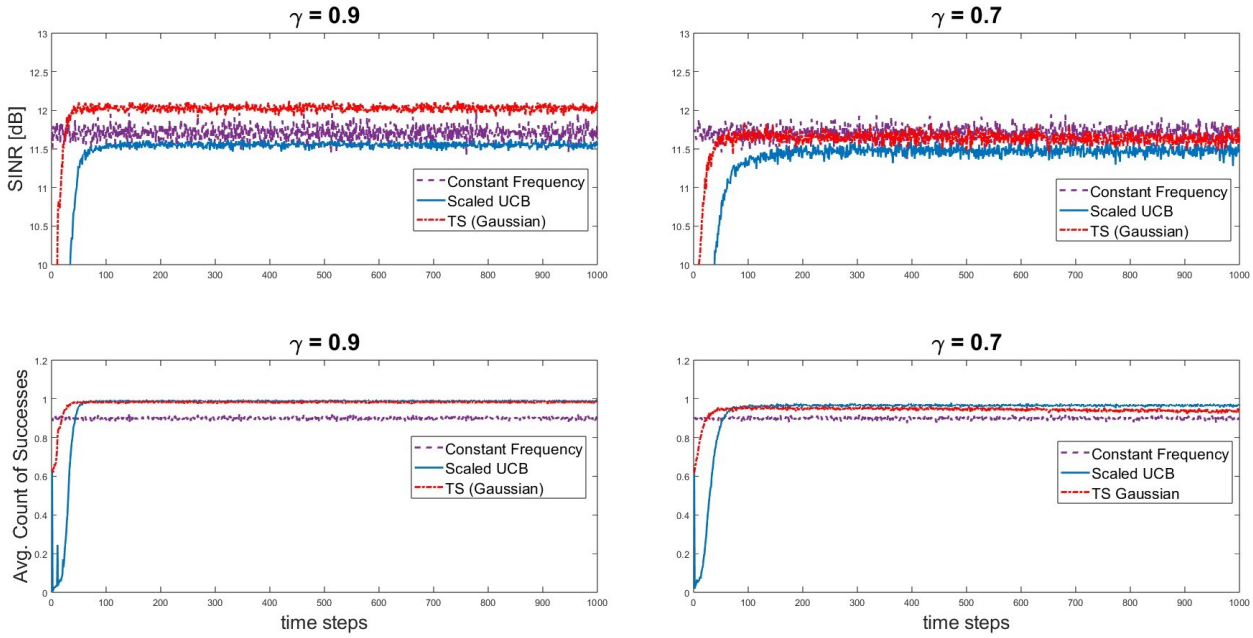

Figure 3.6: Convergence of the average success rate of Scaled UCB and TS (2000 simulations). Comparison (in purple) with a constant frequency corresponding to the best average SINR (absent any knowledge of the jammer).

3.5 Conclusion

There is obvious improvement with usage of the learning algorithms. High SINR and successful transmission rate is especially important in UAC networks because of the limited energy resources, and time for the transmission to reach the receiver. The transmitter should not spend all of its time trying to re-transmit because it is a waste of resources, as well as a waste of energy for both the transmitter and receiver.

When comparing the algorithms, the scaled UCB and TS algorithms perform better than the traditional UCB-1 algorithm. TS has a lower success rate than the scaled UCB algorithm, but it begins to converge at a faster rate and in some cases has a higher overall SINR. This is important for reliable and energy efficient transmission in the underwater environment.

Even when there is a high degree of randomness (e.g., $\gamma = 0.7$ and stochastic channel), it is observed that the learning algorithms are able to converge to a performance that is at least

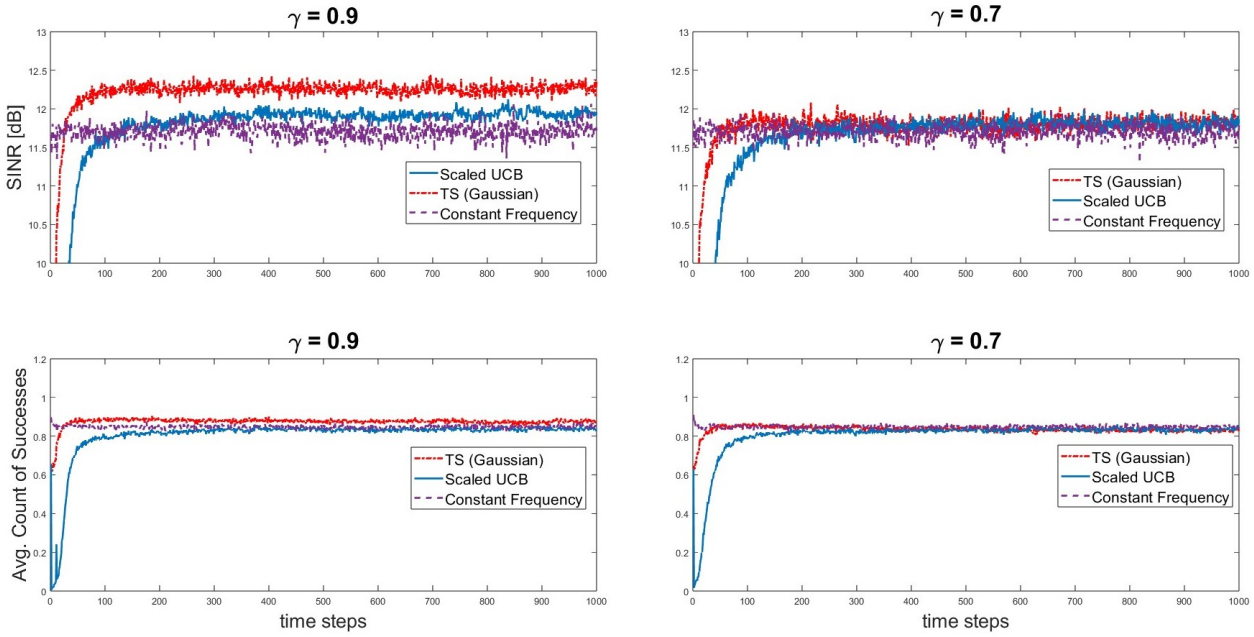

Figure 3.7: Convergence of Scaled UCB and TS with the stochastic pathloss model (2000 simulations). Comparison (in purple) with a constant frequency corresponding to the best average SINR. Time step interval for stochastic channel model was 10 seconds.

as good as a transmitter and receiver pair that are able to select the subband with the best average SINR.

Chapter 4

Linear Jamming Bandits: Learning to Jam OFDM-Modulated Signals

4.1 Introduction

In this chapter, we begin our investigation of the use of RL to jam a communications system. We start simply by trying to jam an OFDM-modulated signal. In subsequent chapters, more complexities will be introduced into this model. We will first analyze if the ordering of choices makes a difference in how the bandit learns. These results will then be used to analyze the performance of using linear TS to jam an OFDM-modulated signal and compare it to traditional UCB-1. The algorithm implements different discretization values of the action space in order to fully understand how the bandit is able to learn and exploit the system. We demonstrate that different jamming schemes may have similar impacts on the victim reception of the signal.

4.2 System Model

We model the legitimate signal between transmitter and receiver (i.e., the "victim signal") as an OFDM-modulated symbol. In the time-domain (TD), the victim signal is represented

as

$$v(t) = \sum_{k=1}^{N_{sc}} V(k) exp(j2\pi kt/N_{sc}), \forall 0 \leq t \leq N_{sc} - 1 \tag{4.1}$$

where $V(k)$ is the frequency-domain (FD) symbol that is typically a digital phase-amplitude-modulated signal, such as M-QAM or M-PSK, and N_{sc} is the total number of subcarriers used in the OFDM transmission. The IFFT operation converts the frequency-modulated symbols to TD signals before they are transmitted.

The jamming signal may employ either single-carrier or multi-carrier waveforms in order to jam the victim signal. The jamming signal in the TD using a form of digital phase-amplitude modulation is modeled as

$$j(t) = \sum_{i=\infty}^{\infty} \sqrt{P_j} j_i g(t - iT) \tag{4.2}$$

where P_j is the average jamming signal power as seen at the victim's receiver and j_i are the modulated jamming symbols. Since the victim signal is modulated in the FD, another viable jamming strategy would be to jam in the FD where the jammer uses an OFDM-modulated waveform. This jamming signal modeled in the TD is

$$j(t) = \sum_{k=1}^{N_{sc}} J(k) exp(j2\pi kt/N_{sc}), \forall 0 \leq t \leq N_{sc} - 1 \tag{4.3}$$

where $J(k)$ is the FD symbol that is typically a digital phase-amplitude-modulated signal, such as M-QAM or M-PSK, and N_{sc} is the total number of subcarriers used in the OFDM transmission. The IFFT operation converts the frequency-modulated symbols to TD signals before they are transmitted.

Here it is assumed that the victim signal, $v(t)$, is attacked by a jamming signal, $j(t)$, and is also effected by a zero-mean Gaussian noise term, $n(t)$, with variance σ^2. It is assumed that the noise has constant received power over the observation interval. The received signal at

the victim's receiver can be expressed as

$$y(t) = \sqrt{P_v}v(t) + \sqrt{P_j}j(t)exp(j\phi) + n(t) \tag{4.4}$$

where P_v and P_j are the power of the victim signal and power of the jamming signal, respectively, and $\phi \in (0, 2\pi]$ is a uniform random variable representing the jamming signal's phase-offset from the victim signal.

At the receiver, an FFT operation is performed to return the underlying frequency modulated victim symbols, $V(k)$,

$$\begin{aligned} Y(k) &= \sum_{t=0}^{N_{sc}-1} y(t)exp(-j2\pi kt/N_{sc}) \\ &= \sqrt{P_v}V(k) + \sqrt{P_j}j(t)exp(-j2\pi kt/N_{sc} + j\phi) + N(k) \end{aligned} \tag{4.5}$$

$\forall 0 \leq k \leq N_{sc} - 1$, where $N(k)$ is the FD version of $n(t)$ after the FFT. For FD jamming, Eqn. (4.5) simplifies to

$$Y(k) = \sqrt{P_v}V(k) + \sqrt{P_j}J(k)exp(j\phi) + N(k) \tag{4.6}$$

where it is assumed the jamming and victim signals operate in the same frequency band. The victim is not able to use a sense-and-avoid technique that allows it to avoid interference it sees in the channel. The main concern is to observe how the jammer learns to jam a victim employing an OFDM-modulated signal. When the jammer is coherent with the victim signal, the phase term reduces to 1 (i.e., $\phi = 0$).

The jammer is able to select jamming techniques that are either in the TD or FD, which is shown in the next section. It is assumed that the TD jamming signal spanned the entire

length of the victim signal. This spreads the power of the TD signal across an entire OFDM symbol which includes the guard bands and cyclic prefix. On the other hand, it is assumed for the FD jamming signal that all of its power is focused in the data-carrying subcarriers of the victim signal, and no power is added in the guard band or cyclic prefix.

The average power of the jamming signal as measured by the jammer-to-noise ratio (JNR) is seen as constant at the victim's receiver, but the instantaneous power may be increased by implementing a pulsed jammer. In this case, the jamming signal is modified by ρ where ρ is the probability the jammer is on and $(1-\rho)$ is the probability the jammer is off. The term ρ modifies the instantaneous power by

$$P_{j_{inst}} = \frac{10^{JNR/10}}{\rho} \tag{4.7}$$

where we assume $\sigma^2 = 1$ for the purposes of setting P_j and P_v.

4.3 Learning Problem

For a specific case of contextual bandits known as contextual linear bandits, the bandit is able to learn about similar actions from the linear relationships between the distribution of statistics of the action space and the contexts learned from the actions taken. In a continuous action space, this is important for the bandit to learn.

At each time step t[1], the bandit will chose an action $a_t \in \mathcal{A}$. The action space is composed of two components: Signaling Scheme and probability prarameter, ρ. The signaling scheme set is composed of TD AWGN, TD BPSK, TD QPSK, FD AWGN, FD BPSK, FD BPSK $\pi/4$, FD QPSK, and FD QPSK $\pi/4$. The power/jamming probability parameter is $\rho \in [0, 1]$,

[1]From this point forward, t represents the time step for the jammer, whereas in Section 4.2, t represents the sampling time of the real-valued signal.

and is discretized by M: $\{1/M, 2/M, ..., 1\}$. The victim is modeled as having a stationary modulation scheme to observe how the bandit learns to jam a particular modulation scheme that the victim employs.

After an action is selected, the bandit will attempt to jam the victim signal with the selected signaling scheme and ρ value. It is assumed the bandit is able to observe the SER achieved from the selected strategy from a feedback channel consisting of the victim's ACKs and NACKs. The bandit will then update the cost function C_t, which is unknown to the bandit *a priori*. The cost function is expressed as,

$$C_t = max(SER_t - SER_{target}, 0)/JNR_t \tag{4.8}$$

where SER_t is the observed SER at time step t, SER_{target} is the target symbol error rate, and JNR_t is the average JNR at time step t. JNR_t is included in this equation to capture the efficiency of resource usage [10], but for the current case, we use a constant JNR in order to examine the effects of the reinforcement learning under a fixed average power constraint. A more realistic cost function based on packet error rate (PER) can similarly be constructed [10].

From the cost function, a context vector is constructed in order to learn about multiple contexts from selecting one action. The context vector is constructed as,

$$\varphi_i(t) = \left[\frac{1}{t} \sum_{l=1}^{t} C_l(a_i), \frac{1}{t} \sum_{l=1}^{t} \mathbb{1}\{C_l(a_i) > \tau\}, \max_{l \leq t} C_l(a_i) \right], \tag{4.9}$$

where $\mathbb{1}$ is the indicator function, and τ is a threshold selected to indicate whether the victim's communication was disrupted [10]. The role of τ is to capture the frequency with which an action set produces a non-zero error rate. The other contexts monitor the average cost of the action selected and the largest disruption caused by the jammer. The contexts of Eqn.

4.9 are chosen because these features when grouped together have been shown to perform well, as shown by Thornton and Buehrer[10]. From the context vector, the bandit discovers the expected costs from the strategies that are chosen. The second feature in the context vector, which tracks the frequency of time steps with non-zero error rates, is particularly important when errors occur infrequently. For a stochastic linear bandit structure, we use the following relationship,

$$C_t(a_i) = \langle \varphi_i, \theta \rangle + \eta_t \tag{4.10}$$

where θ is the weighting vector and η_t is a conditionally sub-gaussian random variable conditioned on the jammer's knowledge of the history of costs, actions, and contexts [10]. This holds for all $a \in \mathcal{A}$ and $t \in \mathbb{N}^+$ [10]. The weighting vector is important because it forms a linear relationship between the actions and contexts gained from the actions chosen. This results in finding out information about similar expected costs from similar actions. We can now use the proposed linear jamming bandits algorithm from Thornton and Buehrer, shown in Alg. 5 [10].

The linear bandit algorithm proposed is TS, which comes from a Bayesian origin. This algorithm is known to provide well-known theoretical and practical results [10, 21, 40]. Since Alg. 5 employs Bayesian inference, it uses a likelihood function and prior distribution to calculate a posterior distribution, from which it samples to pick the actions. These sampled

Algorithm 5: Linear Jamming Bandits

Input Discretization factor M, Cost function $\mathcal{C}$, B $= I_d$, $\hat{\theta} = 0_d$, $f = 0_d$
for *Each time step* $t = 1, ..., T$ **do**
  (1) Sample $\tilde{\theta}_t \sim \mathcal{N}(\hat{\theta}, B^{-1})$
  (2) Assemble context vectors using (4.9)
  (3) Utilize jamming strategy $a_t = \arg\max_i \langle \varphi_i, \tilde{\theta}_t \rangle$
  (4) Update $B = B + \varphi_{a_t} \varphi_{a_t}^T$, $f = f + \varphi_{a_t} \mathcal{C}$, and $\hat{\theta} = B^{-1} f$
end

actions maximize $\langle \varphi_i, \tilde{\theta}_t \rangle$. Usually, the bandit would want to minimize $\langle \varphi_i, \tilde{\theta}_t \rangle$, but since the jammer is trying to maximize the SER caused to the victim, maximization is used instead. An uninformative prior is constructed using a normal distribution which estimates the weighting vector, $\theta \sim \mathcal{N}(\hat{\theta}, B^{-1})$, which are prior set to uninformative values.

The exploration of the TS algorithm comes from randomness in sampling from a distribution with initially high variance. The higher the variance the distribution has, the more the algorithm will explore its available options. Traditional TS requires separate parameterization of the distribution of statistics for each action. Since this is a linear bandit, only one parameterization of the distribution of statistics needs to be up kept with the assumption that an inner product relationship exists with the context vector. This leads to the discovery of a linear relationship with other actions. However, in order to learn features of a particular action, the action still needs to be chosen in order to learn the context vectors of an action set. However, asymptotic convergence to the optimal action is not guaranteed in TS, while it is guaranteed in UCB-1 [10]. In Section 4.4, we show that TS has faster convergence than UCB-1 and better performance overall in terms of SER achieved.

4.4 Analysis

For our purposes, the victim signal is modeled as an OFDM-modulated 16QAM signal with 64 subcarriers and 52 of those subcarriers having data modulated on them. The 12 remaining subcarriers are nulled (6 on each end of the 52 data-carrying subcarriers). A cyclic prefix of length 16 is added on to the OFDM signal.

4.4.1 Action Space Ordering

These particular simulations are performed over 10,000 time steps, with $M = 100$ and JNR = 10 dB. The bandit has 8 jamming choices to choose from: TD AWGN, FD AWGN, TD BPSK, FD BPSK, FD BPSK $\pi/4$, TD QPSK, FD QPSK, and FD QPSK $\pi/4$, where $\pi/4$ means that the constellation is rotated by a phase of $\pi/4$ (Fig. 4.1). In one time step, 200 OFDM symbols are sent by the victim transmitter.

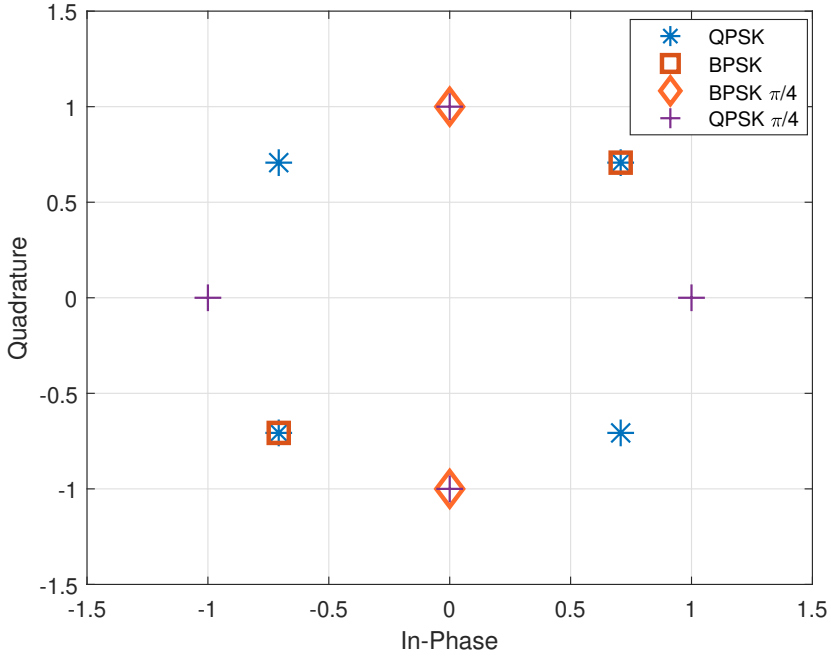

Figure 4.1: Constellation showing the difference between M-PSK and M-PSK $\pi/4$ rotation.

Each simulation is performed at certain SNRs for the victim signal, and the jamming signal is kept at a JNR of 10 dB. This is done in order to compare the results of actions chosen and SER achieved with Fig. 4.2 which shows the SER achieved by various jammers based on the work of [1] to derive optimal jamming schemes. In the simulations that follow, we evaluate performance at SNRs of 2 dB, 16 dB, and 25 dB. These are chosen to model the SER at a low SNR (2 dB), the transition point in optimal modulation schemes (16 dB), and

an obvious choice in optimal modulation scheme (25 dB).

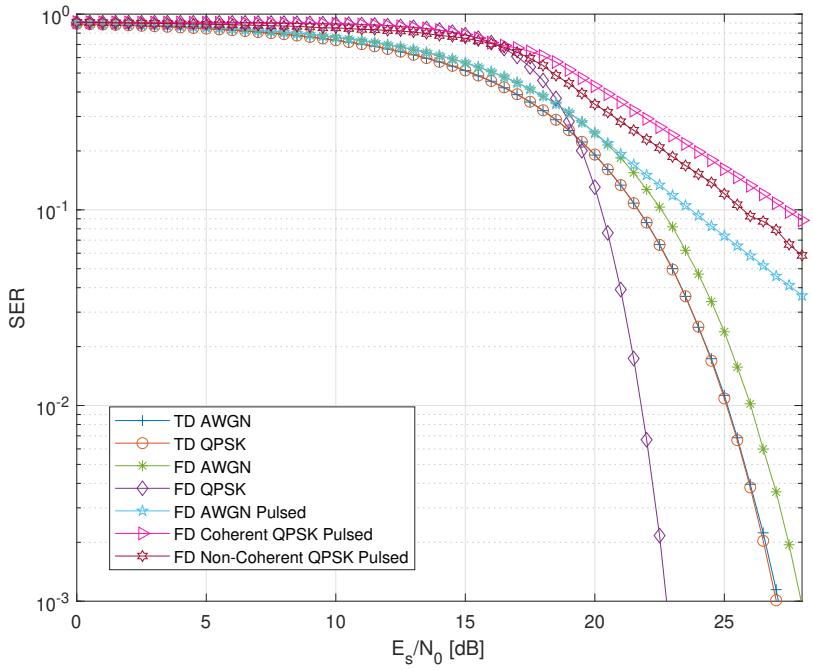

Figure 4.2: Comparing SER of OFDM-modulated 16QAM with different jamming techniques at a JNR of 10 dB [1].

Different orderings of the jammer modulation action space are also evaluated to observe if different groupings of modulation schemes performed better than others. Linear bandits gain information based on the linear relationship between the weighting vector and context vector, so ordering of actions can affect how the bandit learns. Actions that are grouped together have similarities between them where choosing them can affect what the bandit learned.

Order 1 is formed on the basis of grouping TD choices and FD choices together, Order 2 is formed by grouping modulation choices together, and Order 3 is formed by grouping modulation choices together and grouping the $\pi/4$ choices as a separate pair.

The SER is simulated for each of the orders at the specified SNRs to show how fast the

Table 4.1: Grouping of contexts for the Linear Bandit

Order 1	Order 2	Order 3
TD AWGN	TD AWGN	TD AWGN
TD BPSK	FD AWGN	FD AWGN
TD QPSK	TD BPSK	TD BPSK
FD AWGN	FD BPSK	FD BPSK
FD BPSK	FD BPSK $\pi/4$	TD QPSK
FD BPSK $\pi/4$	TD QPSK	FD QPSK
FD QPSK	FD QPSK	FD BPSK $\pi/4$
FD QPSK $\pi/4$	FD QPSK $\pi/4$	FD QPSK $\pi/4$

bandit converges to the optimal SER found from Fig. 4.2. The modulation choices and ρ values are also tracked to see if the bandit could reach the optimal choice(s) and optimal SER(s) found from Fig. 4.2.

The simulations at each SNR show that ordering of the modulation schemes did not matter. All orders performed similarly to each other, and the belief is that since ρ values were not scrambled out of numerical order, the action space led to relying on choosing the correct modulation scheme. This space only consisted of 8 modulation choices which is trivial for a contextual linear bandit. What was interesting was that the bandit was able to converge to the optimal modulation scheme and ρ value for each SNR when coherence between the jammer and victim is assumed $\phi = 0$. The case of SNR = 25 dB is provided in Fig 4.3 to show case this.

Examining Fig. 4.3, all orders perform similarly. All orders begin to converge at the same point at approximately 1600 time steps. By the end of the simulation, Order 2 seems to outperform both Orders 1 and 3, but that improvement is marginal. Going forward Order 2 will be used in the future sections of the paper.

Examining the modulation choices the bandit made, Fig. 4.4 shows all orders found that the optimal scheme was either FD BPSK $\pi/4$ or FD QPSK $\pi/4$. This coincides with what

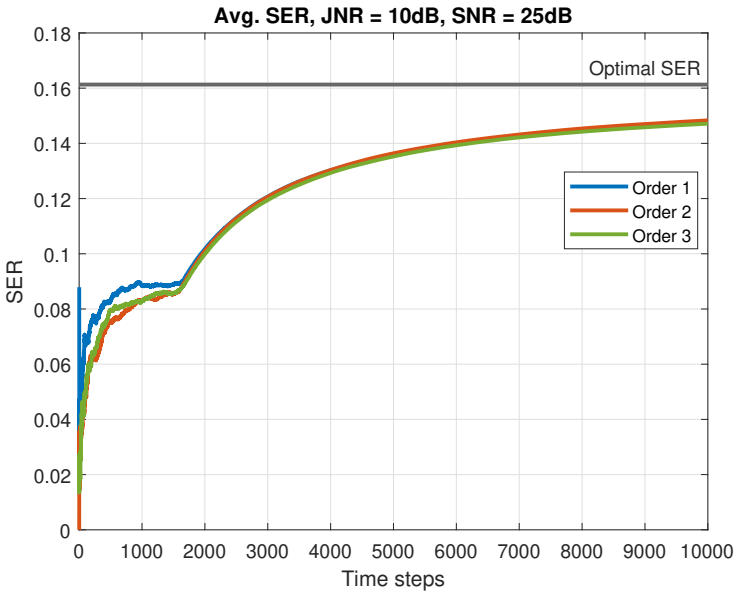

Figure 4.3: Average SER of linear TS with JNR = 10 dB and SNR = 25 dB.

is shown in Fig. 4.2 and derived in [1]. Orders 1 and 2 were very similar where FD BPSK and FD QPSK were also chosen after the beginning convergence period along with the $\pi/4$ options. Order 3 used only FD BPSK $\pi/4$ or FD QPSK $\pi/4$ after the beginning convergence period.

Lastly, the ρ value choices of the bandit were examined for an SNR of 25 dB. All orders converge to the optimal ρ value at the same rate, which is $\rho = 0.239$, as shown in Fig. 4.5.

The bandit is able to learn the optimal jamming choices to inhibit a OFDM-modulated 16QAM signal, but it learns at a slow rate. If jamming choices were taken away that had very similar results, such as removing TD/FD BPSK or TD/FD QPSK, the action space would reduce, and thus the bandit would be able to learn the optimal jamming method faster. Assuming that the bandit is able to distinguish between OFDM and single-carrier modulation using 4th order cumulants [44], the jamming choices can be further narrowed to only FD jamming options.

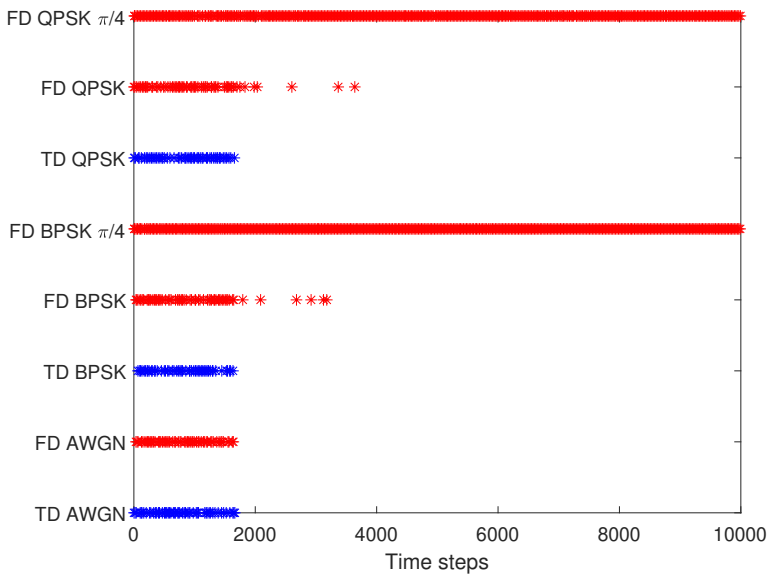

Figure 4.4: Order 2 jamming strategies over time showing whether TD or FD choices were chosen over the course of the simulation with SNR = 25 dB and JNR = 10 dB.

The modulation choices by the bandit draw out an interesting insight. Specifically, that QPSK and BPSK with a $\pi/4$ rotation are both selected by the bandit as it converges. While prior work has found QPSK to be an optimal signal for jamming quadrature constellations [1], the BPSK $\pi/4$ scheme appears to be a special case. It affects both the in-phase and quadrature dimensions due to the rotation. Its only disadvantage is that the in-phase and quadrature jamming is correlated. However, this fact is not exploited by the victim signal.

4.4.2 Comparison of Learning Algorithms

Again, we simulate SNR's of 2 dB, 16 dB, and 25 dB to examine the performance and convergence rates of linear TS and UCB-1. Since having coherence between the victim and jammer is unrealistic in a real-world scenario the jammer is assumed non-coherent to the victim. We examine how discretization of ρ affects the convergence of the algorithms. We let discretization factor, M, be 5, 50, and 100.

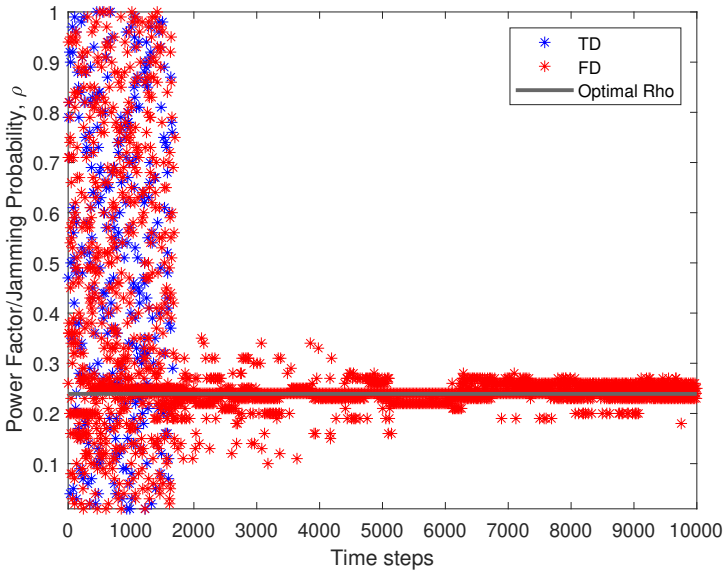

Figure 4.5: Order 2 ρ selection over time showing whether TD or FD choices were chosen over the course of the simulation with SNR = 25 dB and JNR = 10 dB.

$SNR = 2dB$

Both learning algorithms have similar performance in terms of SER, but as is seen in Fig. 4.2, there is little learning to be done since all modulation choices are viable for jamming at a high rate. In Fig. 4.6, TS does show that it learns at similar rates no matter the discretization size of the simulation. UCB-1 only shows asymptotic behavior where the behavior does change with discretization value. As M increases, the convergence is slower and the final SER decreases marginally. So far, discretization (and the resulting action size) does not majorly affect either algorithm.

$SNR = 16dB$

Fig. 4.7 shows signs of discretization having an effect on the learning algorithms. UCB-1 has trouble smoothly converging to a high SER compared to the optimal SER with fine dis-

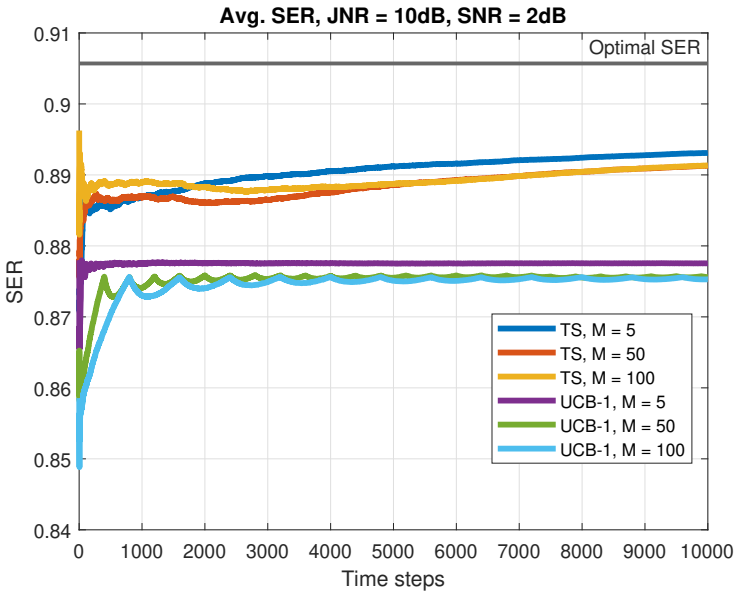

Figure 4.6: Comparison of Linear TS and UCB-1 at SNR = 2 dB and JNR = 10 dB.

cretization resolution. This is due to the large action space that occurs as the discretization resolution of ρ becomes finer. UCB-1 is unable to explore all of the space in order to exploit the victim signal, even after 10,000 steps. With smaller discretization, UCB-1 can smoothly converge to an SER, but it is not close to the optimal SER that is able to be achieved. Linear TS is able to smoothly converge to high SERs relative to the optimal SER. The linear nature of the bandit allows it to find similar statistical properties of similar actions at once, leading to faster convergence and higher SER impacting the victim more than UCB-1.

$SNR = 25dB$

As shown in Fig. 4.8, at high SNRs, both learning algorithms are not close to converging to the optimal SER, but linear TS still outperforms UCB-1. This is due to the non-phase coherence of the victim and jammer. UCB-1 still has trouble converging to a high enough SER to significantly impact the victim's transmission, but we see at low discretization that it can learn to approach SERs close to those achieved by linear TS. It is able to slowly learn

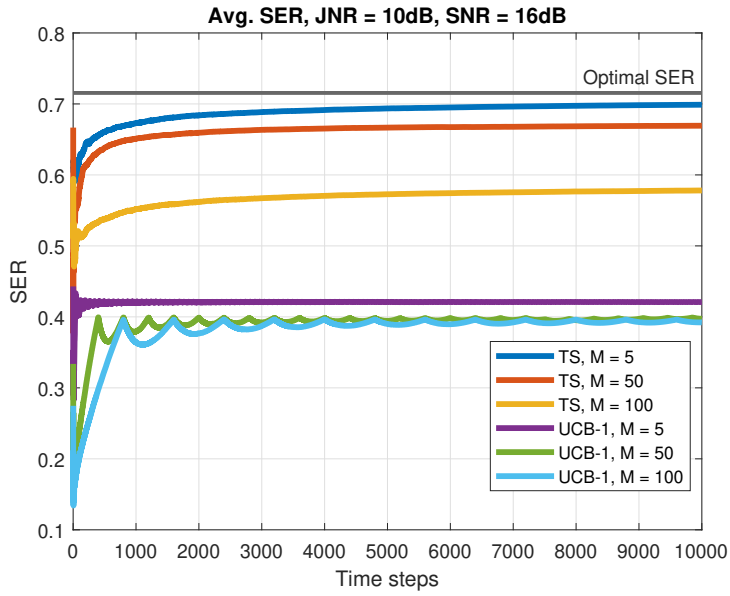

Figure 4.7: Comparison of Linear TS and UCB-1 at SNR = 16 dB and JNR = 10 dB.

and achieve higher SERs compared to higher discretization of the action space. The large discretization leads UCB-1 to stall because of the amount of actions it has to take in order to effectively learn the system and exploit it.

4.5 Conclusion

We show that linear TS overall outperforms UCB-1 in terms of convergence and the SER achieved in reasonable time horizons. Low discretization allows the linear bandit to fully exploit the system quickly and cause the most disruptions to the victim. This upholds the results of [10] in the context of OFDM-modulated signals when the jammer may select from the FD and TD jamming signals. Observing the actions selected by linear TS, we discover an unexpected insight which is that we find BPSK with a $\pi/4$ rotation to be a special case of the optimal jamming scheme for quadrature phase-amplitude victim signals.

If the victim were able to adapt it's modulation scheme to a more robust scheme to improve

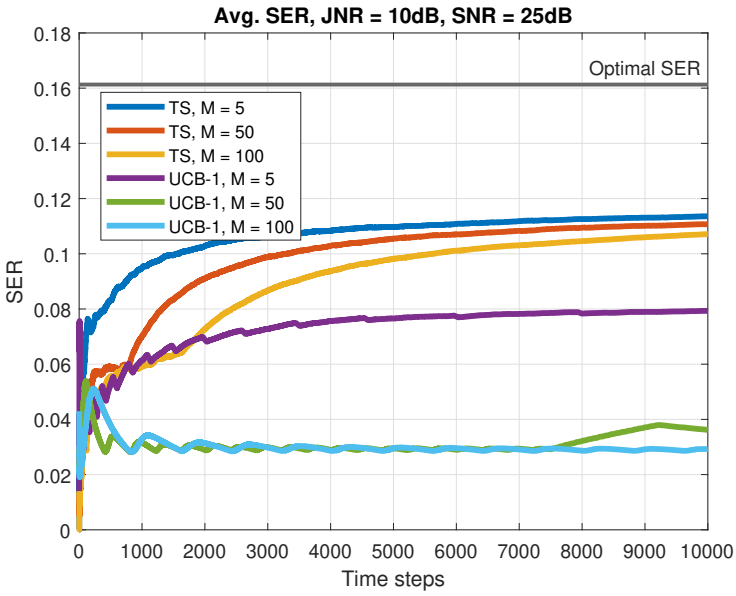

Figure 4.8: Comparison of Linear TS and UCB-1 at SNR = 25 dB and JNR = 10 dB.

performance, the jammer would see this as a success since it is effectively able to lower the victim's throughput. However, the jammer would have difficulty learning and applying effective jamming schemes since it would be seeing lower victim symbol errors than it previously would haven been.

Chapter 5

Understanding and Modifying the Context Vector

5.1 Introduction

In contextual linear bandits, the context vector plays an important role in how the bandit learns. The features of the context vector provide a way for the bandit to observe the effect of the actions taken against the victim. In this particular case, we implement a contextual linear bandit in a wireless communications space where the bandit is a jammer aiming to disrupt legitimate communications between a transmitter and receiver pair in an OFDM system. In the previous chapter, we show that using a Gaussian TS model is superior to using UCB-1 in terms of convergence rate and SER achieved to effectively jam an OFDM signal. We now wish to improve upon the learning algorithm by exploring different context vector feature sets used to either improve convergence time, improve the effectiveness of the bandit in terms of SER caused to the victim, or improve both the convergence time and effectiveness of the bandit.

5.2 Specifics of the Context Vector

The context vector is the vehicle through which the jammer observes the environment and consequences of the jammer's actions that are taken. With the combination of the weighting vector, as seen in Chapter 2 Section 2.1.3, the bandit can learn the relationship between similar actions and the effect it has on the victim under the assumption that this relationship is linear. This linear relationship is between the context and the cost/reward the bandit observes in the environment. However, the bandit still needs to take suboptimal actions to gain information about those particular actions. Until an action has been explored, there will be unknowns about the context and results of choosing that action.

For our particular model, the relationship may not necessarily be linear, but we are constraining our model to treat it that way by using this linear bandit structure.

5.2.1 Size of the Context Vector

The size of the context vector plays a large role in a realistic bandit model, such as, it may effect the computational efficiency of the bandit, i.e., the computational complexity required to make updates on the TS distribution parameters. In general, the larger the context vector is, the less computationally efficient it is. The opposite is also generally true. There may be advantages to using a larger context vector over a smaller context vector if it provides comparably faster convergence, or if it provides noticeably greater rewards over the smaller context vector. In general, there is a fundamental trade-off between size of the context vector and computational complexity.

5.2.2 Over-determined vs. Under-determined

Over-determined

This occurs when there are more context features than needed to either converge quickly to a solution, converge to the optimal solution, or both. The added features may improve the contextual bandit minimally, and, with extra features, it will add to the computational complexity of the system. Additionally, if the absolute value of the weights of the context features are close to zero, then the context vector is likely over-determined

Under-determined

This occurs when there are not enough context features to either converge quickly to a solution, converge to the optimal solution, or both. The context vector may not be taking advantage of all the information available to the bandit to approach the optimal solution, converge quickly, or both. Additionally, if the sampled values for the context features are examined, if the absolute value of the features are very large, but the bandit has poor performance, then the context vector is likely under-determined.

There is a balancing act to choosing the features that add the best value while considering convergence, effect of the bandit to the system, and computational complexity of the bandit. More features may provide faster convergence and higher overall effect, but it will cause higher computational complexity and may not provide enough of a benefit to be worth the strain on the system. Fewer features provides a way to be more computationally efficient, but this may not provide enough information for the system to effectively learn from its observations of the environment. Ideally, we want the absolute value of the weight of the context features to be large while providing good performance.

5.3 Context Vector Models

The original context vector used in the previous chapter, Chapter 4, and the paper by Thornton and Buehrer [10], is expressed as,

$$\varphi_{1i}(t) = \left[\frac{1}{t}\sum_{l=1}^{t} C_l(a_i), \frac{1}{t}\sum_{l=1}^{t} \mathbb{1}\{C_l(a_i) > \tau\}, \max_{l\leq t} C_l(a_i)\right] \tag{5.1}$$

where $\mathbb{1}$ is the indicator function, $C_l(a_i)$ is the cost of the current action, and τ is a threshold selected to indicate whether the victim's communication was disrupted. After observing the effects from both the previous chapter and [10], the choice of ρ, the jamming rate and power scaling factor, was found to be an important decision for the jammer to be able to reach the optimal SER [10]. By including ρ in the context vector, as given by,

$$\varphi_{2i}(t) = \left[\frac{1}{t}\sum_{l=1}^{t} C_l(a_i), \frac{1}{t}\sum_{l=1}^{t} \mathbb{1}\{C_l(a_i) > \tau\}, \max_{l\leq t} C_l(a_i), \rho\right] \tag{5.2}$$

our goal is to define a context vector that can learn a general relationship between ρ and the expected cost/reward. The bandit would be able to learn quicker and more efficiently to exploit the victim communications system better than Eqn. 5.1 would allow.

We also considered multiple non-linear transformations on ρ prior to forming the context vector. These are expressed as,

$$\varphi_{3i}(t) = \left[\frac{1}{t}\sum_{l=1}^{t} C_l(a_i), \frac{1}{t}\sum_{l=1}^{t} \mathbb{1}\{C_l(a_i) > \tau\}, \max_{l\leq t} C_l(a_i), \ln\rho\right] \tag{5.3}$$

$$\varphi_{4i}(t) = \left[\frac{1}{t}\sum_{l=1}^{t} C_l(a_i), \frac{1}{t}\sum_{l=1}^{t} \mathbb{1}\{C_l(a_i) > \tau\}, \max_{l\leq t} C_l(a_i), \ln\frac{1}{\rho}\right] \tag{5.4}$$

$$\varphi_{5i}(t) = \left[\frac{1}{t}\sum_{l=1}^{t} C_l(a_i), \frac{1}{t}\sum_{l=1}^{t} \mathbb{1}\{C_l(a_i) > \tau\}, \max_{l \leq t} C_l(a_i), e^{\rho}\right] \tag{5.5}$$

which capture ρ in a non-linear fashion to model and possibly capture information about ρ that we may be unaware of. By including non-linear functions of ρ, we seek to find a non-linear relationship between ρ and the expected cost/reward within constraints of our linear model.

We include one final context vector with a random component, which is shown as,

$$\varphi_{6i}(t) = \left[\frac{1}{t}\sum_{l=1}^{t} C_l(a_i), \tilde{f} \sim Beta(\alpha_i, \beta_i)\right] \tag{5.6}$$

where the thought was to use the equation for a probability density function (PDF) of a Beta distribution to capture information about ρ as such,

$$g(\rho) = \frac{\rho^{\alpha_i - 1}(1-\rho)^{\beta_i - 1}}{B(\alpha_i, \beta_i)}$$

where

$$\alpha_i = \sum_{l=1}^{t} \mathbb{1}\{C_l(a_i) > \tau\}$$

$$\beta_i = N_i - \alpha_i$$

and

$$B(\alpha_i, \beta_i) = \frac{\Gamma(\alpha_i)\Gamma(\beta_i)}{\Gamma(\alpha_i + \beta_i)}$$

where α_i and β_i are the α and β are related to the ith action, N_i is the total number of times the ith action is chosen, and $\Gamma(.)$ is the Gamma function. To avoid a continuity error due to division by zero whenever $B(\alpha_i, \beta_i) = 0$, the next step was to sample the Beta distribution

instead where the successes and failures were measured by the metric τ. This is similar to how frequency of success is measured by τ in Eqn. 5.1. This contains information on both the modulation scheme chosen and ρ value since they are encoded together in one action in the action space. This would lead to choosing the correct modulation scheme and ρ value more often than that of Eqn. 5.1. This also has the added benefit of less features than Eqn. 5.1.

The sampled variable is named as $\tilde{f}$ to reflect that the sampled Beta distribution will reflect a mean of the $frequency$ of the success of the action. Instead of directly plugging the frequency feature into the context vector like Eqn. 5.1, we found it better to sample a distribution to determine the validity of the frequency and to determine the next action.

Alg. 6 is presented below describing how Eqn. 5.6 is used with linear TS. In Alg. 6, f and

Algorithm 6: Alternate Linear Jamming Bandits

Input Discretization factor M, Cost function $\mathcal{C}$, B = I_d, $\hat{\theta} = 0_d$, $f = 0_d$
Initialize Sample $\tilde{f} \sim$ Beta(1,1) for all i
for *Each time step* $t = 1, ..., T$ **do**
  (1) Sample $\tilde{\theta}_t \sim \mathcal{N}(\hat{\theta}, B^{-1})$
  (2) Assemble context vectors using (5.6)
  (3) Utilize jamming strategy $a_t = \arg\max_i \langle \varphi_i, \tilde{\theta}_t \rangle$
  (4) Update α_i and β_i
  (5) Sample $\tilde{f}_i \sim$ Beta(α_i,β_i)
  (6) Update $B = B + \varphi_{a_t}\varphi_{a_t}^T$, $f = f + \varphi_{a_t}\mathcal{C}$, and $\hat{\theta} = B^{-1}f$
end

$\tilde{f}$ have no relation.

5.4 Context Vector Analysis

These simulations were conducted with a constant JNR of 10 dB, and for the SER, are a cumulative average over one simulation. This was done to simply compare results faster. They were also conducted with victim SNR = 2, 16, and 25 dB, similar to the previous chapter, Chapter 4. The value of τ was set to 0.5. This value was determined to work best over all SNR cases. Different values for τ depending on the SNR will be examined in the next section. A linear TS algorithm was used to conduct these simulations similar to that in Chapter 4 (Alg. 5). The equation for the PDF of a Beta distribution was also included in these simulations to show the evolution of reaching Eqn. 5.6.

Fig 5.1 shows the performance of all the context vectors at an SNR of 2 dB and JNR of 10 dB. The context vector that performs the best is clearly that of Eqn. 5.6 with Eqn. 5.2 and Eqn. 5.1 following closely, as shown in Fig. 5.1. Eqn. 5.6 has faster convergence and better performance in terms of SER caused to the victim compared to all other context vector options. As mentioned previously, the thought is that the feature that samples the Beta distribution keeps track of the modulation scheme and the ρ that is associated with that particular action. By updating and sampling Beta distributions related to the action space, the system is able to examine the paired modulation schemes and ρ values that have the highest probability to cause damage to the victim transmitter and receiver pair.

Repeating the simulation for an SNR of 16 dB, the results appear to be closer, as shown in Fig. 5.2. Eqn. 5.6 again outperforms the other context vectors in terms of convergence rate and SER caused to the victim.

Lastly, an SNR of 25 dB was examined, as shown in Fig. 5.3. Eqn. 5.6 again outperforms the other context vectors. The other context vectors have a longer exploration time compared to that of Figs. 5.1 and 5.2. Eqn. 5.6 is able to immediately exploit the system and cause a

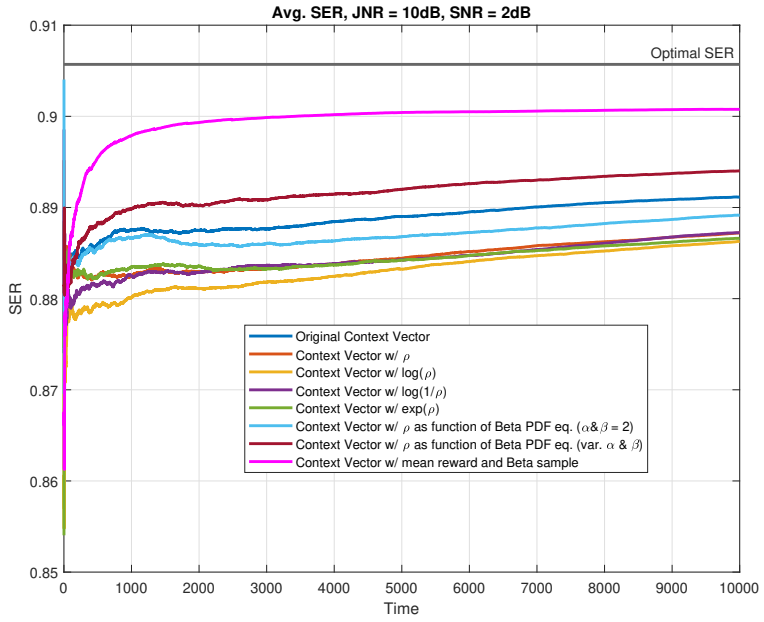

Figure 5.1: Cumulative average of SERs of different context vector over one simulation for SNR = 2 dB.

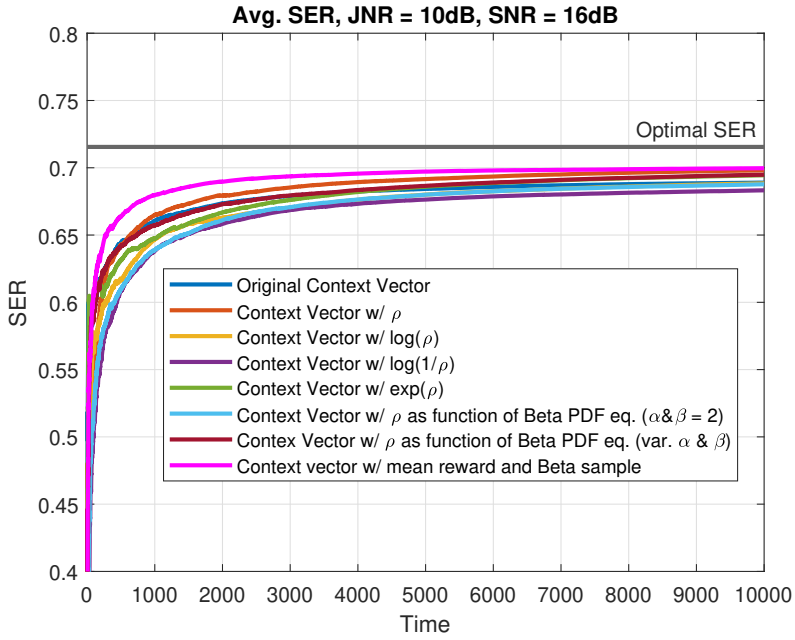

Figure 5.2: Cumulative average of SERs of different context vector over one simulation for SNR = 16 dB.

higher SER than the other context vectors.

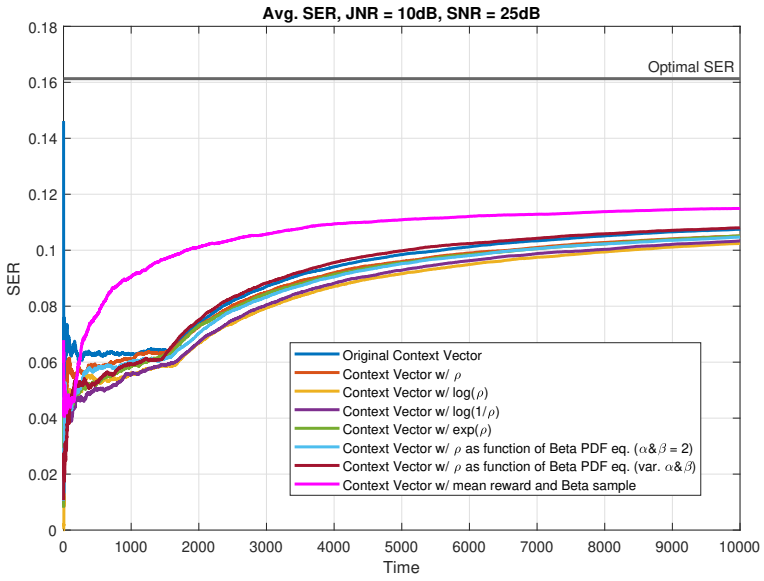

Figure 5.3: Cumulative average of SERs of different context vector over one simulation for SNR = 25 dB.

There are cases where Eqn. 5.1 does outperform Eqn. 5.6, as shown in Fig. 5.4. Since the algorithm is non-deterministic with Eqn. 5.6, multiple simulations were performed where the SER is now averaged over the simulations instead of being averaged cumulatively over one simulation to investigate overall performance. For each average, 200 simulations were performed, and only Eqn. 5.1 and Eqn. 5.6 were compared to give a frame of reference for Eqn. 5.6.

For Figs. 5.5, 5.6, and 5.7, Eqn. 5.6 performs just as good as or better than Eqn. 5.1. Eqn. 5.6 always converges faster, and has comparable or better SER jamming capabilities than that of Eqn. 5.1.

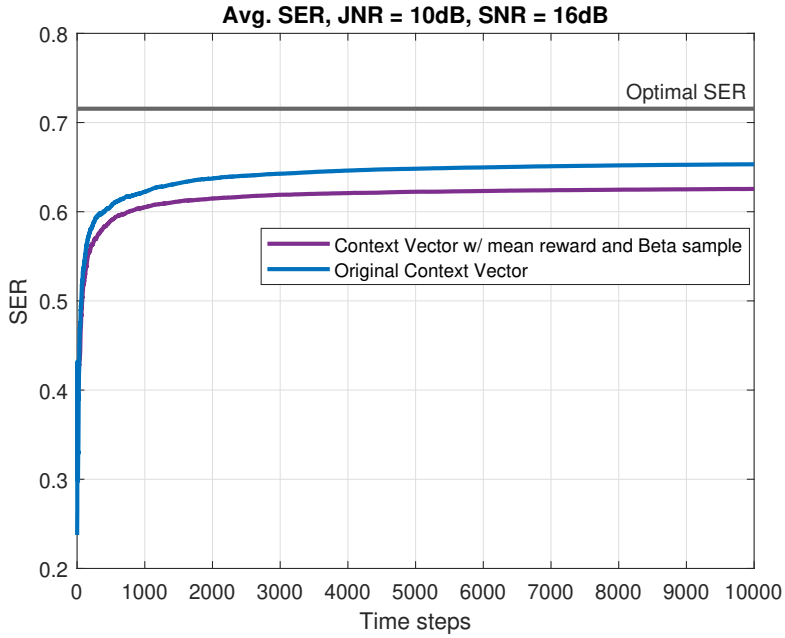

Figure 5.4: Case of original context vector outperforming Eqn. 5.6

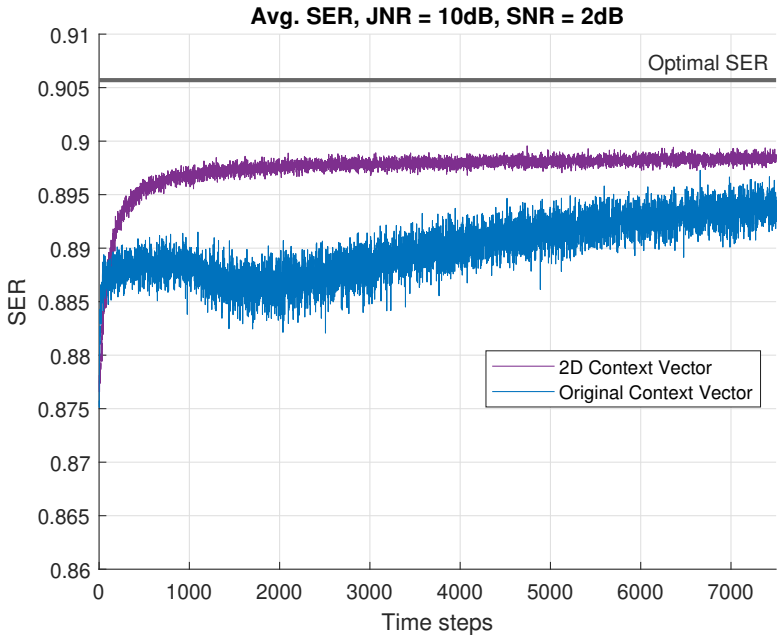

Figure 5.5: Comparison of original context vector (Eqn. 5.1) and 2D context vector with sampled Beta random variable (Eqn. 5.6) averaging over 200 simulations of SERs for SNR = 2 dB.

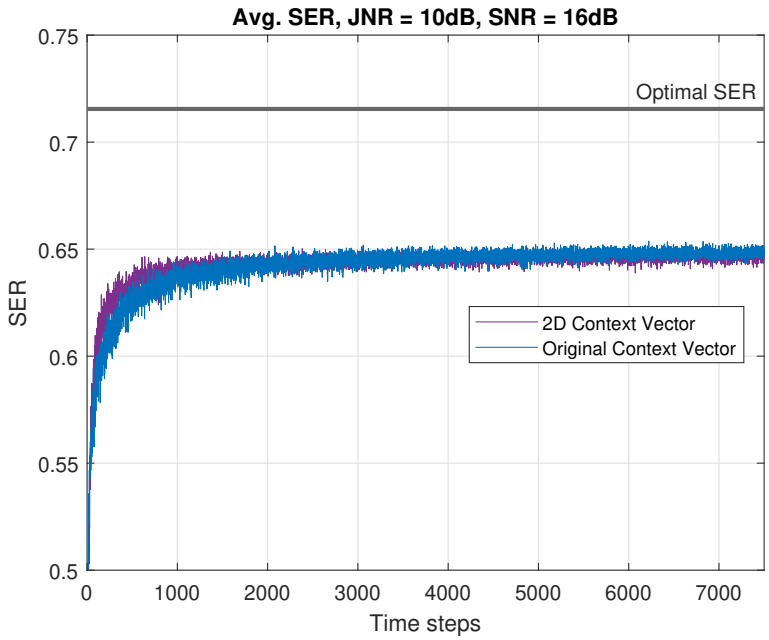

Figure 5.6: Comparison of original context vector (Eqn. 5.1) and 2D context vector with sampled Beta random variable (Eqn. 5.6) averaging over 200 simulations of SERs for SNR = 16 dB.

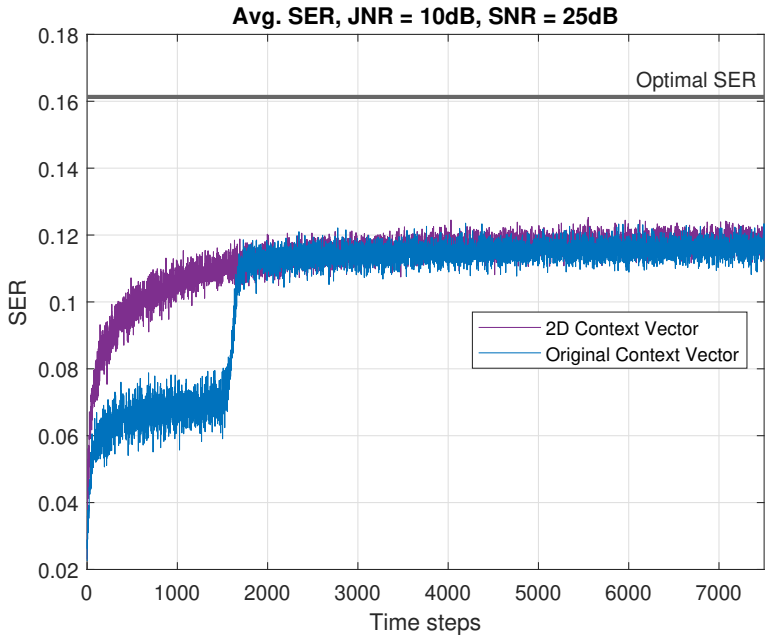

Figure 5.7: Comparison of original context vector (Eqn. 5.1) and 2D context vector with sampled Beta random variable (Eqn. 5.6) averaging over 200 simulations of SERs for SNR = 25 dB.

5.4.1 Choices of the Bandit

We also examined the choices the bandit made under Eqn. 5.6 which include the jamming schemes and the selection of ρ. These choices may give a deeper insight in how the bandit is able to converge quickly and effectively disrupt the victim Tx/Rx communications as good as or better than the bandit using Eqn. 5.1.

Figs. 5.8 and 5.10 show the modulation choices and ρ values selected by the bandit, respectively, under Eqn. 5.6 at an SNR = 2 dB and JNR = 10 dB. Figs. 5.9 and 5.11 show the modulation choices and ρ values selected by the bandit, respectively, under Eqn. 5.1 at an SNR = 2 dB and JNR = 10 dB. Figs. 5.8 and 5.10 are a large contrast to Figs. 5.9 and 5.11. In Figs. 5.9 and 5.11, they show that any jamming scheme and value for ρ work well at disrupting communications at low SNRs. This follows from Chapter 4 and Fig. 4.2 that shows at low SNR any jamming scheme is viable for causing a high SER to the victim. In the case of Eqn. 5.6, the bandit explores for a minimal amount of time steps and then chooses to stick to one jamming scheme and one ρ to exploit the victim Tx/Rx pair. The bandit does not explore all possible values for ρ either. It explores the available options and eventually lands on a ρ value that is close to the optimal value, marked by the grey line in Fig. 5.10. Once it finds that option, it rarely does exploring on the side, only aiming to exploit, which contrasts with what the bandit does when under Eqn. 5.1.

Figs. 5.12 and 5.14 show the modulation choices and ρ values selected by the bandit, respectively, under Eqn. 5.6 at an SNR = 25 dB and JNR = 10 dB. Figs. 5.13 and 5.15 show the modulation choices and ρ values selected by the bandit, respectively, under Eqn. 5.1 at an SNR = 25 dB and JNR = 10 dB. Figs. 5.12 and 5.14 are also interesting because they contrast the decision of the bandit under Eqn. 5.1. Under Eqn. 5.1, as shown in Figs. 5.13 and 5.15, the bandit will explore initially and then eventually land on a few main jamming

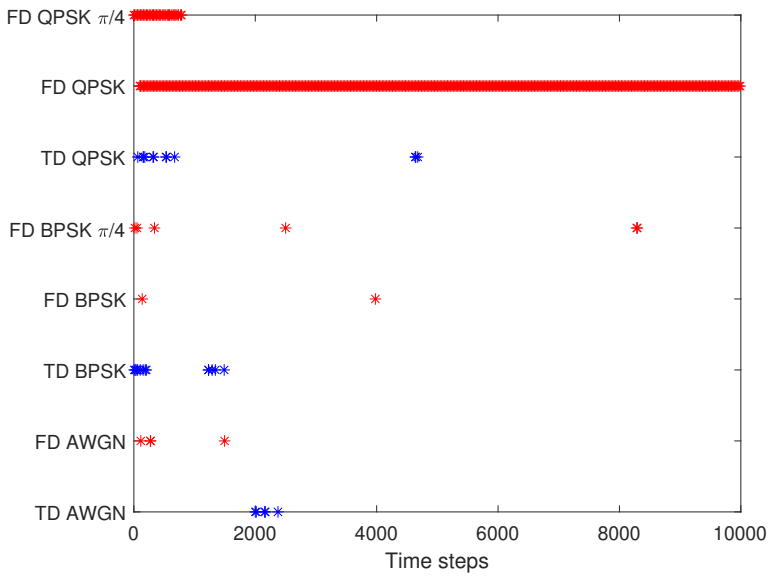

Figure 5.8: Bandit selection of jamming scheme over time at an SNR = 2 dB and JNR = 10 dB under Eqn. 5.6. The red markers are FD jamming options and blue markers are TD jamming options.

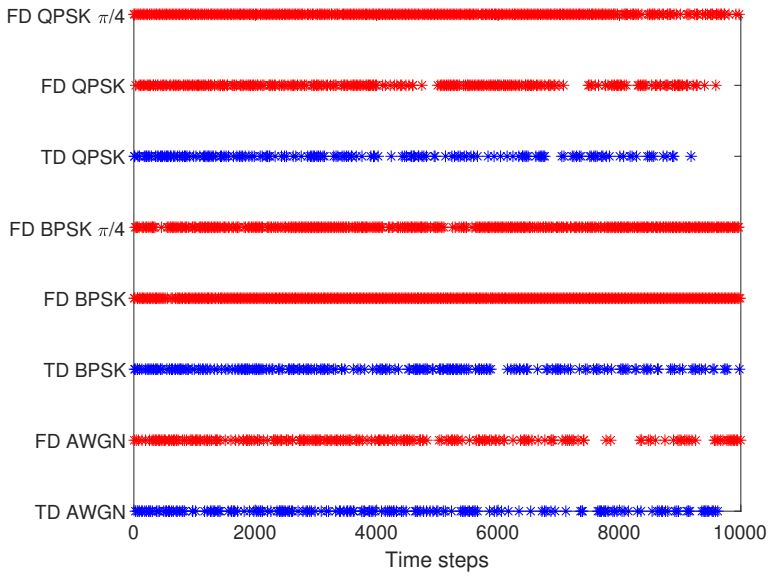

Figure 5.9: Bandit selection of jamming scheme over time at an SNR = 2 dB and JNR = 10 dB under Eqn. 5.1. The red markers are FD jamming options and blue markers are TD jamming options.

schemes to exploit the victim and a defined range of ρ values. Under Eqn. 5.6, the bandit will not strictly settle on a jamming scheme or ρ value to use. There is a clear jamming choice the bandit will make, but unlike Figs. 5.8 and 5.10, the bandit will explore on the side to check for better jamming options instead of strictly exploiting the victim. Even under these circumstances, Eqn. 5.6 outperforms Eqn. 5.1.

From this examination, the bandit under Eqn. 5.6 may do significantly less exploring overall than the bandit under Eqn. 5.1. The bandit finds a set of choices that exploit the bandit well and will take advantage of that choice until the simulation is over. The bandit still conducts side exploring, but will rarely change its initial choice. This is also not necessarily good because if the bandit finds a suboptimal choice and sticks with it, it will under perform Eqn. 5.1, as seen in Fig. 5.4.

What we typically look for in a context vector is a balanced system; one that is not over-determined or under-determined. Eqn. 5.1 is a good example of a balanced system since it performs consistently over multiple simulations while discovering the optimal choices to use. Eqn. 5.6 may not discover the optimal solutions to use, but it discovers effective ways to jam the victim in our desired time span while performing just as well as Eqn. 5.1. In this way, it is still a balanced system. For this case, the importance is the ability for the bandit to not find the best/optimal choice, but the ability for the bandit to quickly learn a meaningful solution. Eqn. 5.6 does not always converge to the optimal solution, but it is able to learn quickly to a solution that is close to optimal in order to quickly cause damage to the legitimate Tx/Rx pair. Eqn. 5.1 does a good job at effectively learning and applying the optimal solution(s), but as we have seen, the optimal solution may not provide the kind of performance we are looking for in the time span we desire.

The use of Eqn. 5.6 can be useful for systems that are computationally constrained and desire to cause a large jamming impact on average with high convergence rates to the optimal

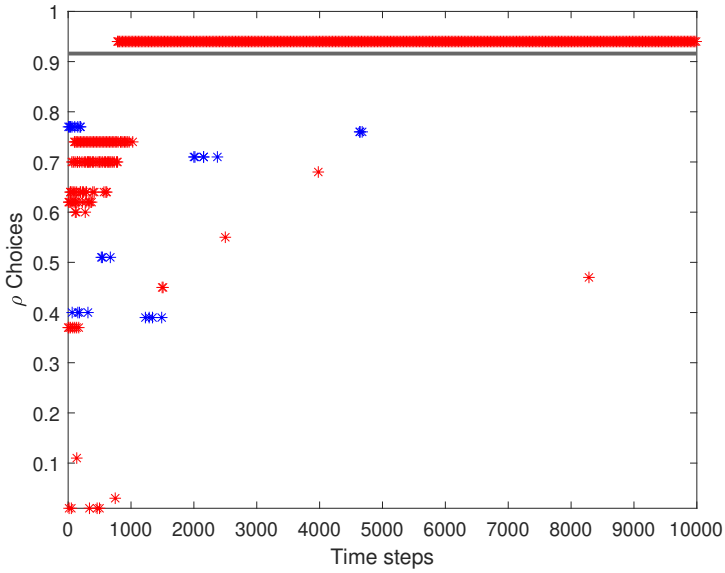

Figure 5.10: Bandit selection of ρ over time at an SNR = 2 dB and JNR = 10 dB under Eqn 5.6. The grey line marks the optimal value for ρ. The red markers are FD jamming options and blue markers are TD jamming options. The red markers are FD jamming options and blue markers are TD jamming options.

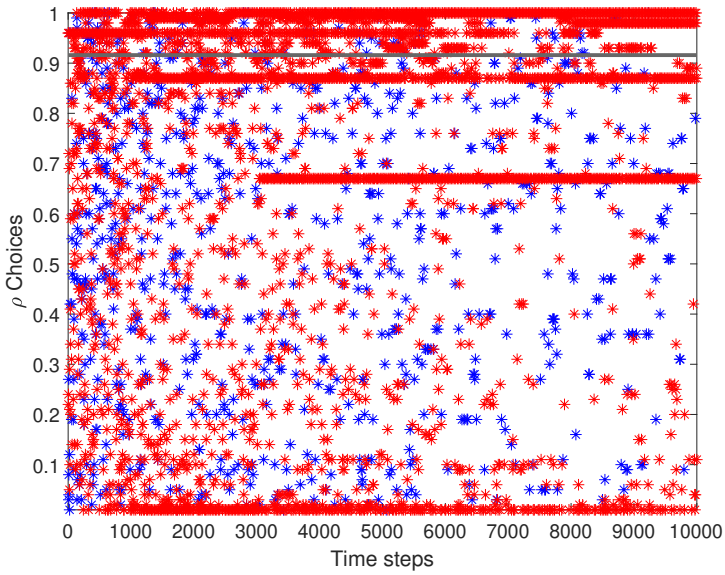

Figure 5.11: Bandit selection of jamming scheme over time at an SNR = 2 dB and JNR = 10 dB under Eqn. 5.1. The grey line marks the optimal value for ρ. The red markers are FD jamming options and blue markers are TD jamming options.

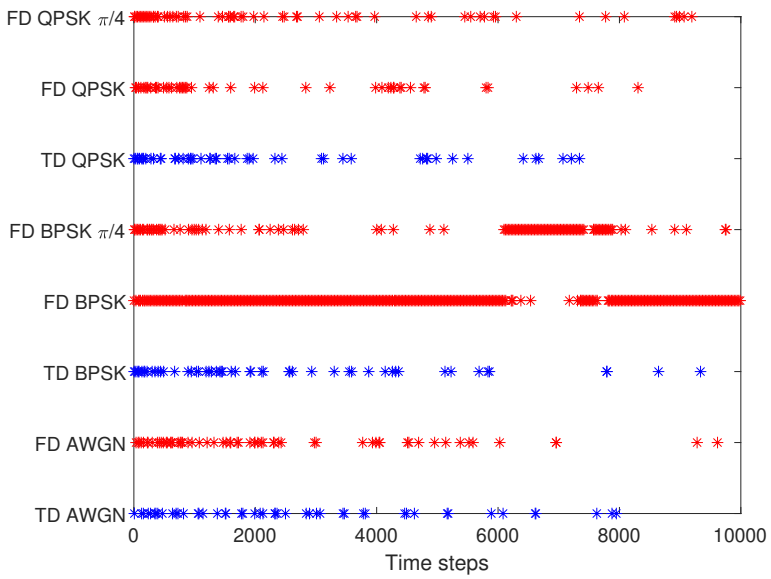

Figure 5.12: Bandit selection of jamming scheme over time at an SNR = 25 dB under Eqn. 5.6. The red markers are FD jamming options and blue markers are TD jamming options.

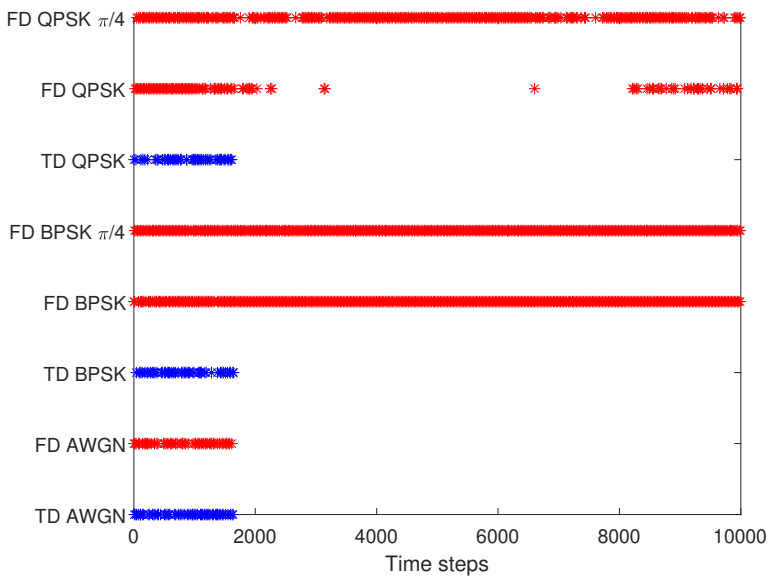

Figure 5.13: Bandit selection of jamming scheme over time at an SNR = 25 dB and JNR = 10 dB under Eqn. 5.1. The red markers are FD jamming options and blue markers are TD jamming options.

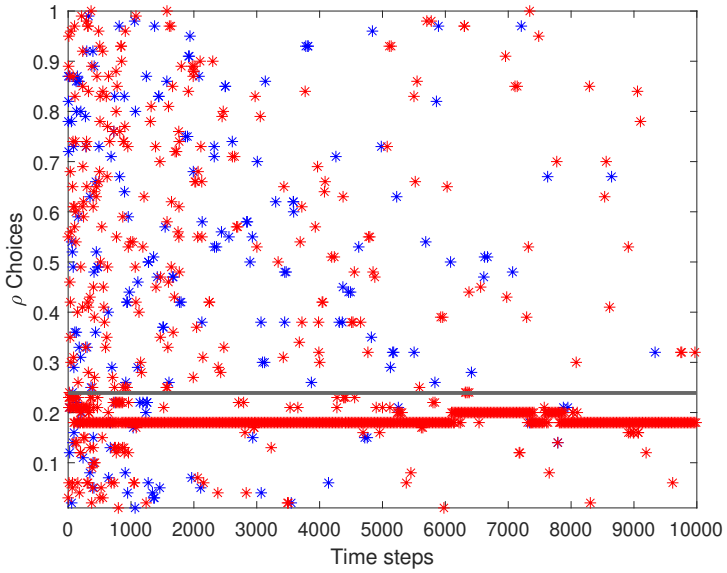

Figure 5.14: Bandit selection of ρ over time at an SNR = 25 dB under Eqn. 5.6. The grey line marks the optimal value for ρ. The red markers are FD jamming options and blue markers are TD jamming options.

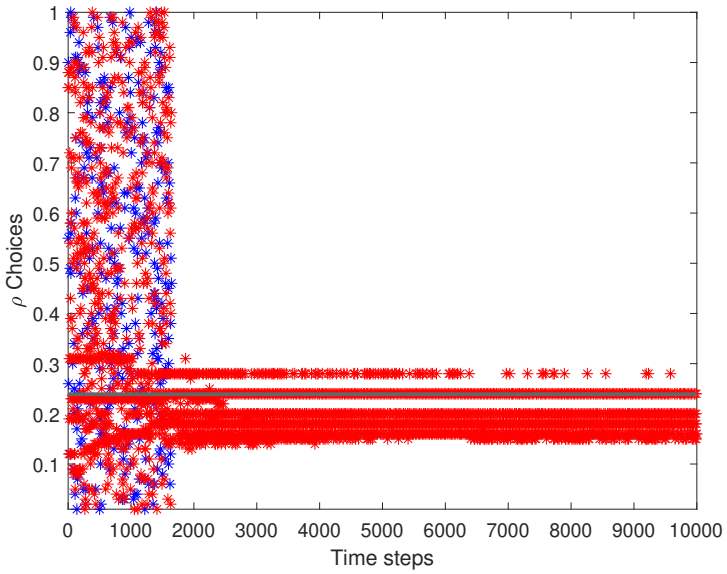

Figure 5.15: Bandit selection of ρ over time at an SNR = 25 dB and JNR = 10 dB under Eqn. 5.1. The grey line marks the optimal value for ρ. The red markers are FD jamming options and blue markers are TD jamming options.

solution. If computational complexity is not an issue, and a reliable system is desired, Eqn. 5.1 would be a solid option to implement in a jamming system.

The key takeaway is that different victim transmission strategies correspond to different "ideal" context vectors.

5.5 Impact of Jamming Success Metric

As previously mentioned, Eqn. 5.1 is the context vector originally used in the work by Thornton and Buehrer [10] and extended for use in Chapter 4. The second feature of the equation is defined as

$$\varphi_{2(i)}(t) = \frac{1}{t}\sum_{l=1}^{t} \mathbb{1}\{C_l(a_i) > \tau\}, \tag{5.7}$$

where $\varphi_{2(i)}(t)$ is the second feature of the context vector of arm i at time step t, $\mathbb{1}$ is the indicator function, $C_l(a_i)$ is the cost or reward received from taking the action a_i at the current time step l, and $\tau \in (0, 1)$ is the metric on whether the cost of taking that action is considered a success. If the cost returned is greater than τ, this will indicate a 1 for a success, and 0 for the opposite. This is summed over all time to obtain the frequency of success of arm i. Previously in Chapter 4, τ was used to capture the frequency of a non-zero error rate, but this introduces a dilemma on how to set τ. A low value of τ will capture all error rates, but will provide no feedback to the jammer on what action is useful. A high value may provide feedback on only the optimal schemes that achieve high error rates, but there may be cases where no schemes will achieve error rates as high as the set value of τ. Constructive feedback will not be provided in this case either. We provide an examination on τ to study how useful this metric is and what values are useful in providing feedback to the bandit. In this section, we strictly consider Eqn. 5.1 to examine the effect of using

different values for τ depending on the optimal SER value. Again, we examine SNRs of 2, 16, and 25 dB. We use both cumulative averages over one simulation and averages over multiple simulations to compare over one simulation vs. the average outcome of the bandit.

For an SNR of 2 dB, it is apparent the value of τ does effect the convergence of the bandit, shown in Fig. 5.16. The best value for τ is $\tau = 0.9$. For a single simulation, it has a fast convergence rate compared to the other values and achieves a high error rate. For multiple simulations, $\tau = 0.9$ still converges faster, but at 6,000 time steps, the other values of τ achieve a higher SER than $\tau = 0.9$. It can be argued that these values are better than $\tau = 0.9$, but they have a slow convergence rate. Since the SER caused to the victim system are high for all values of τ, the SER achieved when it surpasses that of $\tau = 0.9$ is considered negligible.

For an SNR of 16 dB, there is the same conclusion as before with the selection of τ being important in convergence and effectiveness of the jammer, as shown in Fig. 5.17. There are also suboptimal choices of τ, such as, choosing a value that is not possible to obtain. One such choice is choosing a value for τ that lies above the optimal SER ($\tau = 0.9$). It slowly converges to an effective SER and only converges after a large amount of time steps have elapsed. Setting τ larger than the optimal SER potentially makes this feature meaningless due to no information feedback if τ is never reached. On the other hand, too low of a value for τ and it will often be reached, rendering little to no information about schemes that provide good jamming results. A more viable choice is to choose a value for τ that is close to the optimal SER but is not the optimal SER. It is so the jammer will know which actions in the action space are most effective in obtaining the optimal SER or close to the optimal SER. This deems the action a success and will update the frequency of success feature of the context vector for that action reliably. In reality, τ cannot be set without knowing the SNR ahead of time. It can be argued that $\tau = 0.9$ is a viable choice if looking at the multiple

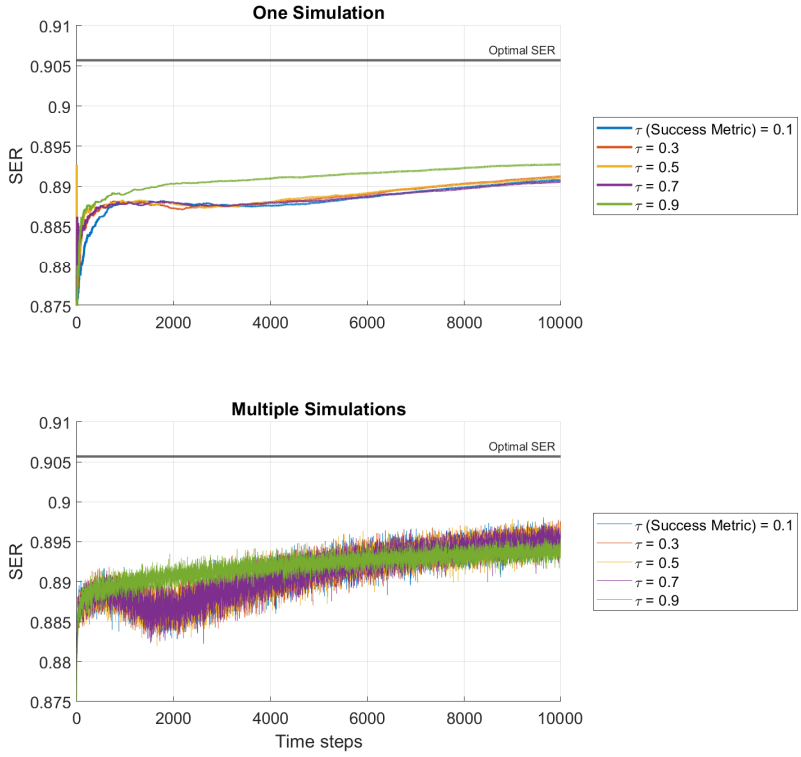

Figure 5.16: τ comparison at an SNR of 2 dB for both one simulation and multiple simulations.

simulations graph since it does come closest to the optimal SER, but the convergence time is too long, and compared to the next best choice of $\tau = 0.7$ the SER achieved is negligible between the two options.

Fig. 5.18 shows the convergence the bandit over different values for τ for an SNR of 25 dB. We see similar results for what we have seen in the previous figures in both one simulation and multiple simulation graphs. Overly high values for τ will produce suboptimal results, but balanced τ values will produce results that are better for convergence time and effectiveness of the jammer. Balanced in this case means using values for τ that are not low compared to the optimal SER and not high compared to the optimal SER. This is for the bandit to fairly measure "successes" and "failures" in jamming and promote more optimal actions for

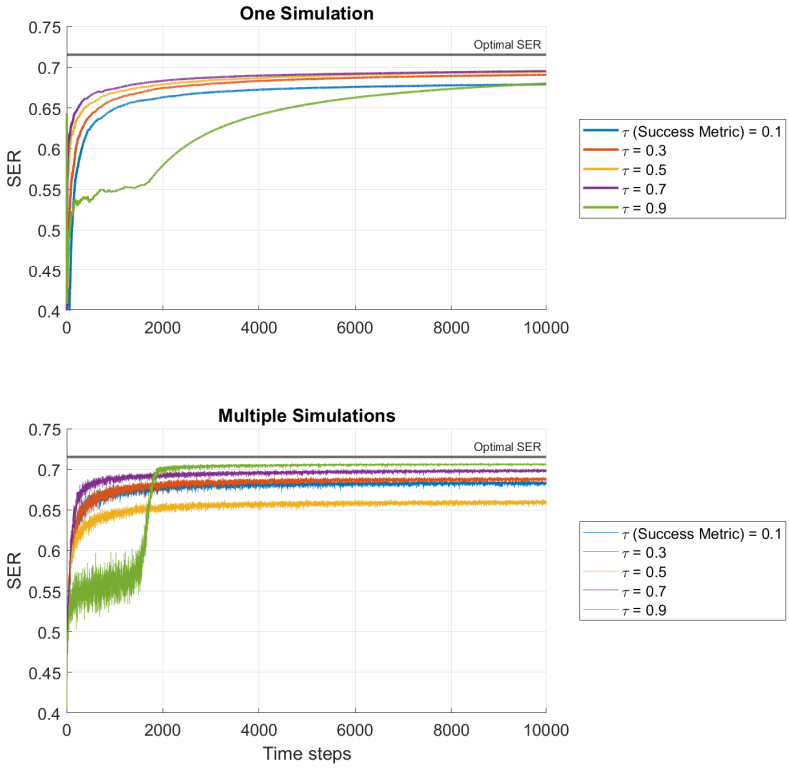

Figure 5.17: τ comparison at an SNR of 16 dB for both one simulation and multiple simulations.

the bandit to choose. If a suboptimal τ is used, it may promote more suboptimal actions depending on the actions it has already chosen and learned from. In this case, $\tau = 0.1$ is the best option. An argument can be made that the other τ values are better than $\tau = 0.1$ since they eventually achieve a higher SER, but the convergence time is slow and the SER achieved is negligible compared to $\tau = 0.1$. They are also less consistent than $\tau = 0.1$ since there may be cases such as the one simulation graph that shows $\tau = 0.1$ fully outperforms all other options.

This also may showcase why the bandit explores more than usual under Eqn. 5.6 in Section 5.4.1 when the SNR is 25 dB compared to when the SNR is 2 dB since the τ used for both of those simulation were equal to 0.5, as shown in Figs. 5.8 and 5.12, which is suboptimal

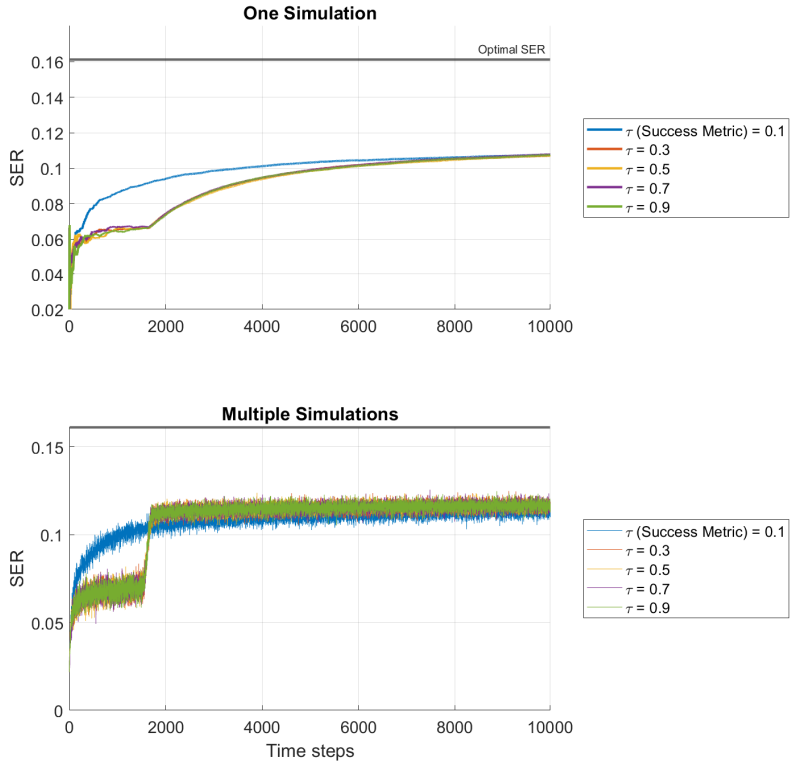

Figure 5.18: τ comparison at an SNR of 25 dB for both one simulation and multiple simulations.

for an SNR = 25 dB.

For all three SNR cases, it can be seen that the more reliable choices for τ have a smaller variance of SER achieved in multiple simulations, than the suboptimal choices. The suboptimal choices usually have a wider variance in SER achieved, which shows that these options are unreliable for SER achieved. This can also be seen in the single simulation graphs that show the suboptimal choices have slow convergence time and do not approach the optimal SER.

5.5.1 Weighting Vector Analysis

The weighting vector sampled from the posterior distribution represents how effective the actions taken and contexts observed were. A way to analyze the usefulness of the context features is to examine how the weighting vector puts weights on the context features. The greater the relative magnitude to the other context features the weighting value provides, the better the context feature is. The closer to 0 relative to the other context feature weighting values are, the worse that feature is. The only context vectors that are going to be examined are Eqns. 5.1 and 5.6 because they perform the best as previously seen in Section 5.4. $\tau = 0.5$ will be used on all simulations for fair comparison, and as in previous sections, these simulations will take place at SNRs = 2 dB, 16 dB, and 25 dB under the presence of a jammer with a JNR = 10 dB. The first 100 values were cutoff from the simulation to give the bandit time to settle on the value of the features.

Starting with the bandit under Eqn. 5.1, Fig. 5.19 shows the weighting values over time that the distribution gave the context vector features at an SNR = 2 dB. The average reward and max reward features are the most valuable to the bandit while the frequency of success of the action does not seem to matter to the bandit. This could be because the bandit always achieves a success no matter what modulation scheme, so the feature does not matter to the bandit. The level of τ was then increased to $\tau = 0.9$ because it was shown in section 5.5 that at that level, the bandit performs the best. We also wanted to test whether the frequency of success context did not matter because a success was always achieved in the case of $\tau = 0.5$. If τ was set to 0.9, a success would be achievable since it is below the highest possible SER that is able to be reached, but a success would not always be achieved since the threshold is high compared to the optimal SER, therefore possibly increasing the importance of this context. After this change, in Fig. 5.20, it was shown that the frequency of success context mattered even less to the bandit while the other two contexts mattered more.

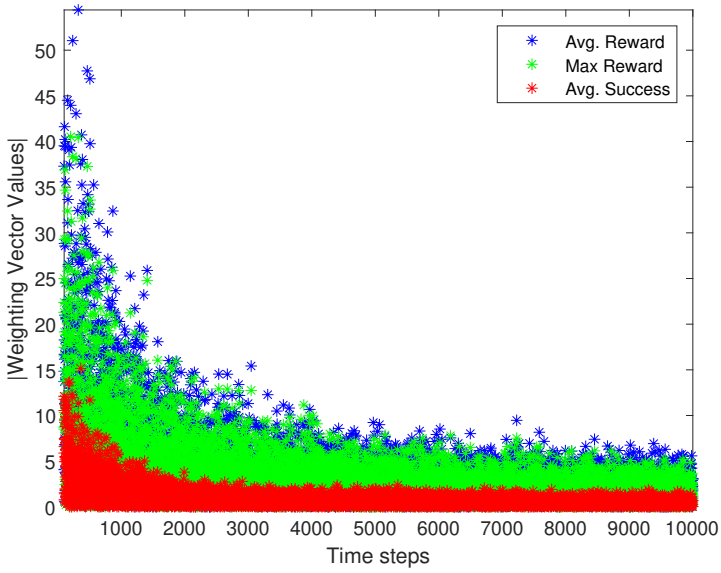

Figure 5.19: Weighting values of the context features over time at an SNR = 2 dB and JNR = 10 dB under Eqn. 5.1.

The other interesting result is when the SNR was set to 25 dB. When $\tau = 0.5$, the importance of frequency of success outweighs both context features, as shown in Fig. 5.21. This is because it is very unlikely for the bandit to achieve a jamming success result when the highest possible SER it can achieve is a rate of 16.13%. If $\tau = 0.5$, which equates to an SER of 50%, it is very unlikely to reach that level, so the bandit deems the frequency of success feature important and weighs the other features as not important. This may be due to the algorithm realizing that no matter how much it weighs the frequency of success feature nothing will change. In the extremely rare probability that it does experience a success, then the bandit will most surely choose the scheme that provided a success going forward. For this particular case, it seems that the bandit under Eqn. 5.1 can be classified as under-determined. The above statement can be supported by setting τ to an achievable level like 0.1. This is shown in Fig. 5.22 where the distribution of weighting is similar to that of Fig. 5.19. Again, we show that the value chosen for τ is important since it influences how the bandit learns and what features it deems important for it to learn.

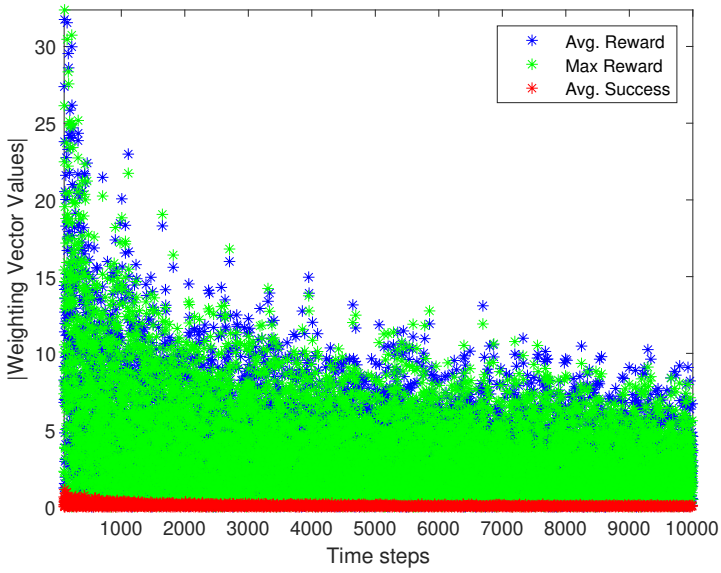

Figure 5.20: Weighting values of the context features over time at an SNR = 2 dB and JNR = 10 dB under Eqn. 5.1 with $\tau = 0.9$.

The frequency of success feature and sampling of the Beta distribution feature are compared because they both account for the successes and failures of the actions but in different ways. We are trying to gain insight as to why Eqn. 5.6 performs as good as or better than Eqn. 5.1. Now we examine how the bandit weights the context features under Eqn. 5.6. Fig. 5.23 shows the weights of the context features at an SNR of 2 dB. We see that mean reward is the more important feature in the context vector, but the sampling of the Beta distribution does help the bandit learn. The disparity between these features is not as wide as seen in Fig. 5.19 where the frequency of success feature is not needed at all. If τ is set to 0.9, as seen in Fig. 5.24, the weights have less variance in their values compared to Fig. 5.23. The importance of the beta sampling context feature in Eqn. 5.6 also decreases with the increase of τ from 0.5 to 0.9, which also follows from Fig. 5.19.

The other case examined was an SNR of 25 dB. In Fig. 5.25, similar results to the previous figure, Fig. 5.24, can be shown. If τ is changed to 0.1 and 0.9, respectively, the variance of

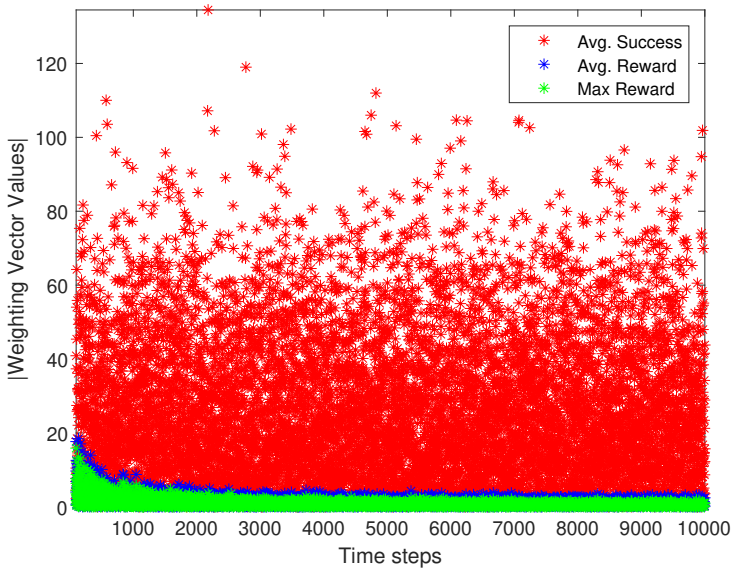

Figure 5.21: Weighting values of the context features over time at an SNR = 25 dB and JNR = 10 dB under Eqn. 5.1.

Table 5.1: Average Weights Under Eqn. 5.1, $\tau = 0.5$

	Avg. Reward	Freq. Success	Max Reward
2 dB	2.726	0.8147	2.339
16 dB	2.746	0.1306	2.622
25 dB	1.217	25.00	0.6403

the weights for each context changes, as shown in Figs. 5.26 and 5.27. In Fig. 5.26, the variance of the weight for the average reward becomes wider while the variance and average value for the Beta sampling becomes smaller. In Fig. 5.27, the variance of the weight for the average reward becomes smaller while the the variance and average value for the Beta sampling becomes larger. This is due to the fact where the success parameter is very unlikely to be achieved due to the constraint on the highest possible SER that could be achieved. This is not as drastic as the case for Eqn. 5.1 where the weight of the frequency of success is increased dramatically, as shown in Fig. 5.21. The average values of the weights for each SNR case can be found in Table 5.1 for Eqn. 5.1 and Table 5.2 for Eqn. 5.6.

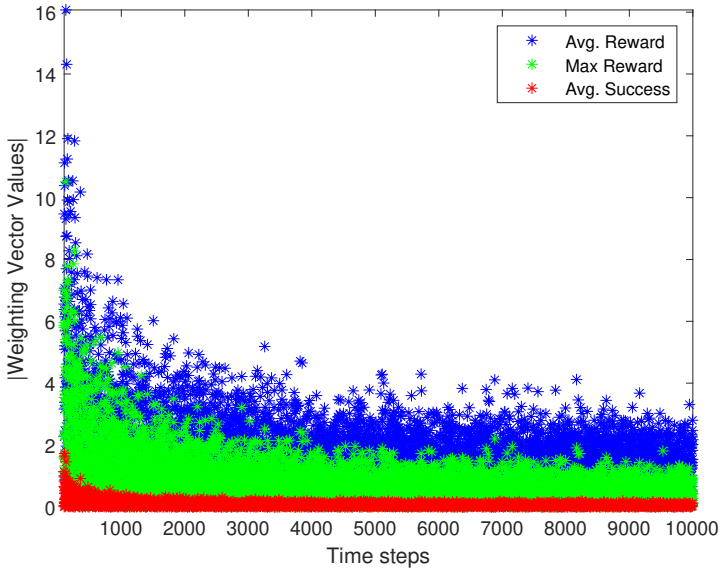

Figure 5.22: Weighting values of the context features over time at an SNR = 25 dB and JNR = 10 dB under Eqn. 5.1 and $\tau = 0.1$.

Table 5.2: Average Weights Under Eqn. 5.6, $\tau = 0.5$

	Avg. Reward	Beta Sample
2 dB	0.998	0.2365
16 dB	0.997	0.1390
25 dB	1.003	0.0742

Even though both context features consider success, they are considered in different ways. The second feature from Eqn. 5.6 being sampled lets Eqn. 5.6 perform as good as or better than Eqn. 5.1. As the samples become larger, the bandit is more sure that the jamming options it is selecting are the optimal ones for jamming. The average reward also being weighted more in Eqn. 5.6 than in 5.1 relative to the other features allows the bandit to find and stick with one jamming option the entire simulation. Since the average reward and max reward are weighted evenly in Eqn. 5.1, this leads to the jammer selecting multiple jamming schemes and not selecting just one. This leads to more time exploring options and possibly selecting schemes that are suboptimal compared to one that is more effective. It seems it is

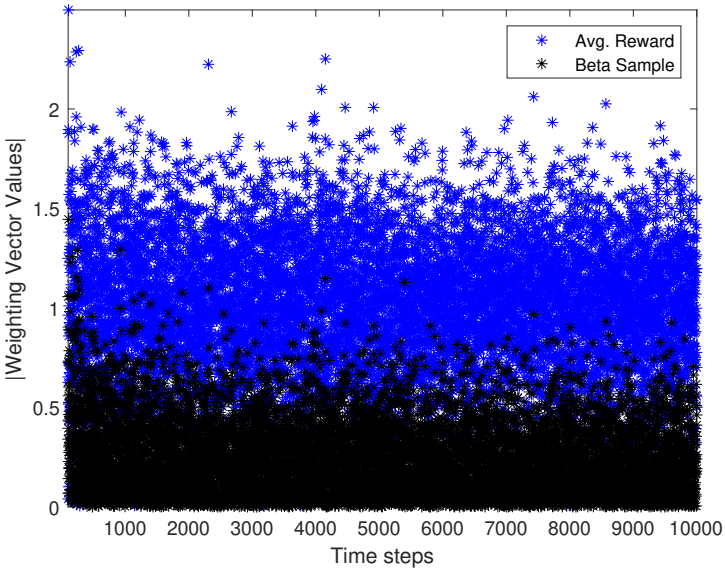

Figure 5.23: Weighting values of the context features over time at an SNR = 2 dB and JNR = 10 dB under Eqn. 5.6.

almost better for the bandit to do less exploring and only select one jamming scheme that for sure does well than for the bandit to do more exploring and select multiple schemes that appear to do well.

5.6 Conclusion

There might be multiple balanced context vectors that are effective to use for the bandit. For our use case, it is more important how fast the bandit learns to use an effective strategy than for the bandit to learn and use an optimal strategy or multiple optimal strategies. In the best case, the bandit will be able to learn and use the optimal strategy quickly and efficiently.

For different victim transmission strategies, there are corresponding "ideal" context vectors to use. A "one-fits-all" context vector may be suboptimal in the case of the victim having

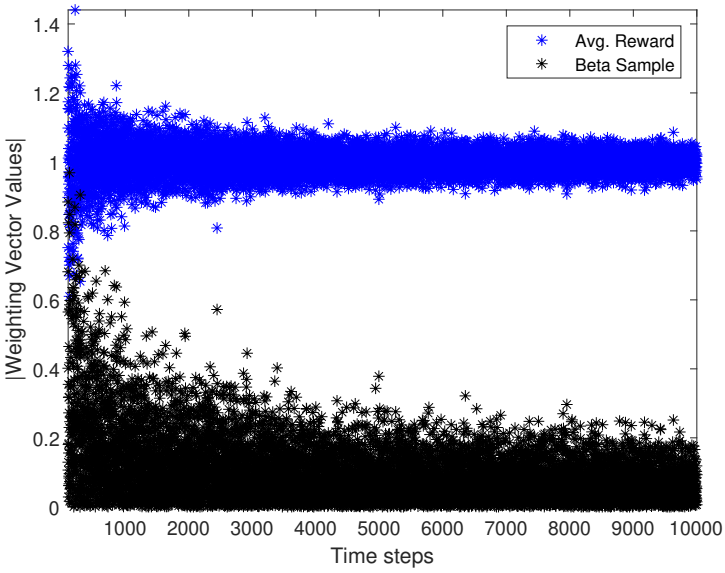

Figure 5.24: Weighting values of the context features over time at an SNR = 2 dB and JNR = 10 dB under Eqn. 5.6 with $\tau = 0.9$.

multiple different transmission strategies.

The selection of τ is more important than initially realized. It is a measure for success, and if overly high it might deem no action a success, and if overly low, it may deem every action a success. This may not allow the bandit to filter out suboptimal actions and lower the rate at which it converges to an effective SER to jam the victim. The suboptimal τ value is also unreliable because it was seen in the multiple simulation graphs that the suboptimal values usually had a wider variation in terms of SER achieved. There may be times where it eventually outperforms given a long time to converge, but there are also chances it may not converge as seen previously in the single simulation graphs.

We have shown how Eqn. 5.6 performs better by showing how the weights of the features change depending on the context vector used. Having only the average reward feature in the context vector allows the jammer to find one transmission strategy that works well and continues to choose that strategy. This also gives the sampling of success feature to have

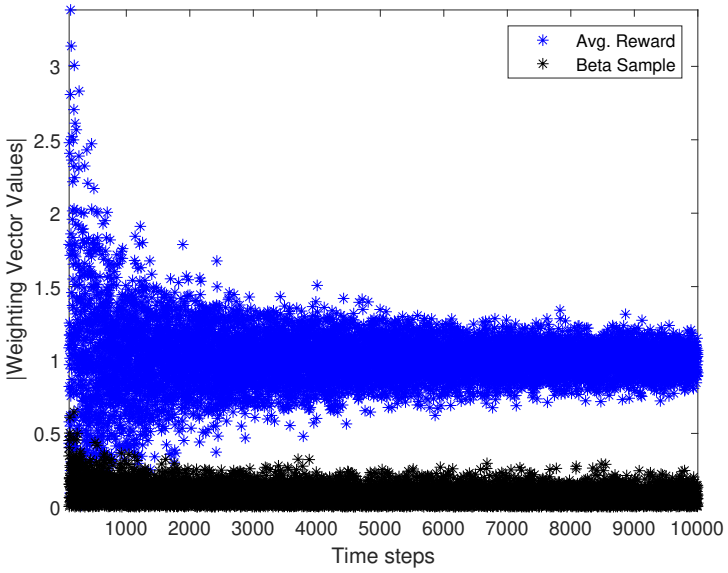

Figure 5.25: Weighting values of the context features over time at an SNR = 25 dB and JNR = 10 dB under Eqn. 5.6.

more weight relative to its partner feature. This jamming strategy either performs equal to or better than Eqn. 5.1 which is able to find multiple optimal solutions, but at the cost of more exploration and the jammer switching between jamming strategies instead of sticking to one jamming strategy. This is due to having both the max reward and average reward features "compete" with each other. The frequency of success feature gets weighted less in return. This supports both of the other conclusions on the context vector analysis and the analysis on τ.

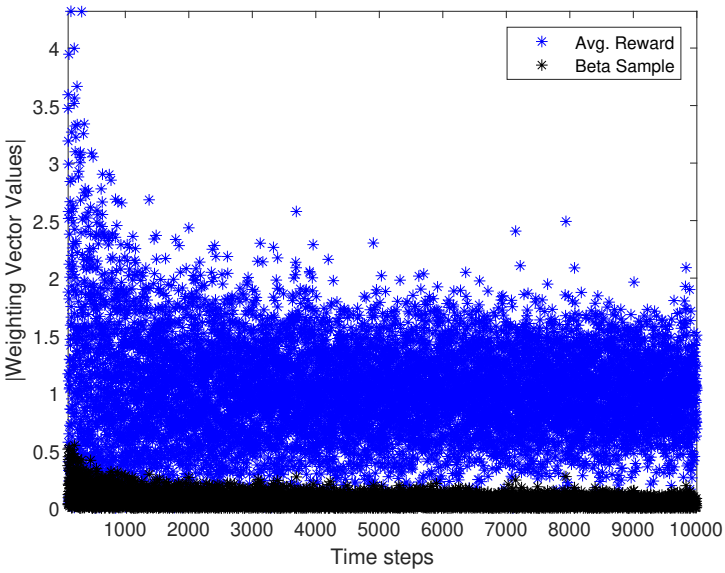

Figure 5.26: Weighting values of the context features over time at an SNR = 25 dB and JNR = 10 dB under Eqn. 5.6 with $\tau = 0.1$.

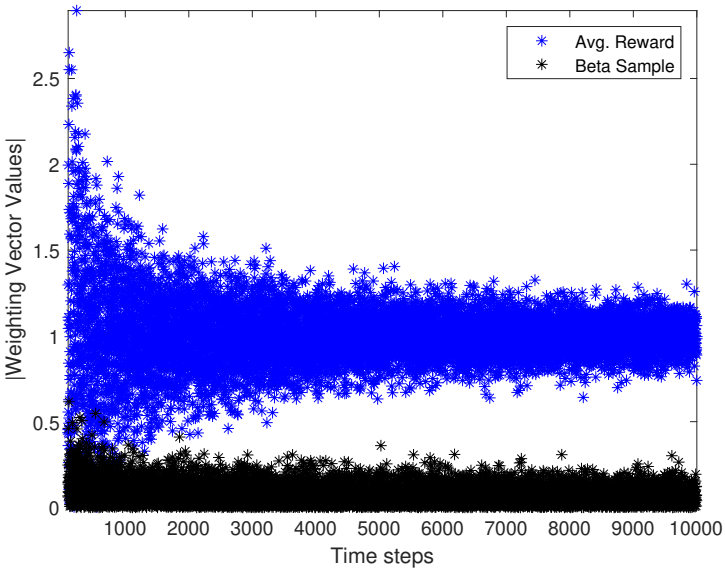

Figure 5.27: Weighting values of the context features over time at an SNR = 25 dB and JNR = 10 dB under Eqn. 5.6 with $\tau = 0.9$.

Chapter 6

Jamming Coded Systems

6.1 Introduction

Previously in Chapters 4 and 5, we spent time exploring using RL to jam an OFDM-modulated signal and examining the design of the context vector to see if we could possibly learn faster to jam an OFDM-modulated signal. We now explore jamming systems employing forward error correction (FEC) coding with the help of RL. Jamming a coded-system is more complex than that of just jamming an OFDM-modulated signal. FEC coding adds redundancy to the data stream to combat loss of data due to noise, fading, and multipath propagation. This allows the receiver to detect and correct errors in the bit stream using some sort of recovery process determined by the FEC code used. The redundancy in the data lowers throughput but ensures the data is sent correctly over the communications channel. By using FEC and OFDM, the system achieves both time diversity and frequency diversity adding to the robustness of the system. A 5G-based system will also employ an uplink and downlink channel with different scheduling techniques that make it difficult for a jammer to know what frequency and time the victim Tx/Rx pair is transmitting on. This in turn makes it difficult for the jammer to have perfect observations of the errors to see how effective its jamming strategies were. By exploring how RL will learn to jam a coded-based system, we hope to gain insight on how the agent will learn and how to adjust parameters, such as the context vector features, in order to learn fast and efficiently. Imperfect feedback observations

are later added to see how the bandit adapts to receiving unreliable information.

6.2 Understanding Jamming a Signal with Forward Error Correction

In order to give a full analysis of jamming a coded-based system, we first need to understand the effect of jamming a signal that employs FEC. Before, the jamming rate and power scaling parameter, ρ, was used to increase the instantaneous power of the jammer while keeping the average power the same across a moment in time. Increasing ρ allowed the jammer to take advantage of the victim by making sure it jammed an adequate amount of symbols, especially at higher SNRs where the victim signal had a higher chance of being decoded correctly due to the difference in average signal-to-jammer ratio (SJR) power. In the case of a coded system, we are unsure that focusing the jammer power through pulsing within the subcarriers, pulsing within the OFDM symbols, or randomly within the subcarrier-symbol block will affect the victim signal significantly because of the redundancy added to the signal due to FEC.

FEC today commonly uses block codes which correct fixed packets of data of a certain length. Each block is called a codeword where the code rate determines the number of bits dedicated to data and the number of bits dedicated to parity bits which help the FEC code correct bits. The code can only correct so many bits. The value of ρ chosen can impact the codeword if the jamming signal hits enough of the codeword to render it ineffective in correcting the affected bits or symbols. The jamming method, such as subcarrier or OFDM symbol jammming, is important in this aspect as well since the agent may not know how the codeword is applied to the signal, such as, across the subcarrier or across the symbols. Since

we are no longer looking at things on a symbol error basis this changes the impact of ρ. The codeword may experience the same amount of average jamming power over the duration of the codeword for a range of ρ values depending on the mapping. The change experienced by the victim is the concentration of that power into a subset of the coded bits.

The FEC is able to decode the received bits in two separate ways: soft- and hard-decision decoding. Soft-decision decoding represents a probability distribution of what the bits could be by calculating the likelihood or log-likelihood ratio (LLR) of what the bits could be. Soft-decision decoding uses the calculated minimum Euclidean distance to another codeword and extra information like the LLR in order to decode the data correctly. Soft-decision decoding is usually used when a large noise floor is present and where uncertainty of the received bits is greater. Hard-decision decoding uses the minimum Hamming distance between codewords in order to decode bits. Hard-decision decoding is usually used when a high SNR is achievable.

6.2.1 System Model

We consider two signals: a single-carrier victim signal and an OFDM-modulated victim signal, both employing an FEC. The single-carrier signal in the TD using a form of digital phase-amplitude modulation is represented as,

$$v(t) = \sum_{i=\infty}^{\infty} \sqrt{P_v} v_i g(t - iT), \tag{6.1}$$

where P_v is the power of the victim signal, v_i are the digitally modulated victim symbols, and $g(t-iT)$ is the pulse shape. Similarly, the OFDM-modulated victim signal in the TD is constructed by Eqn. 4.1. The code uses soft-decision decoding in order to decode the bits. The jammer generates an AWGN signal modified by ρ to examine how different values for ρ at a constant SNR and JNR affect the victim signal. For both single-carrier and OFDM

cases, we consider phase coherence between the jammer and victim. Both signals are also sent through an AWGN channel, as well as being affected by the jammer producing an AWGN signal.

6.2.2 Single-Carrier

For this section, the victim signal employs BPSK with 3/4 rate low-density parity-check (LDPC) code with block size 27. There is one codeword per 486 data symbols. Each value of ρ was simulated for 500 time steps where each time step consisted of 50 codewords being transmitted. The BLER was averaged for each time step, and then these values were averaged over 500 time steps for each value of ρ.

6.2.3 Analysis: TD BPSK Signal - TD AWGN

The block error rate (BLER) was first examined over ρ to see the effect ρ has on the victim signal's codewords. We initially simulate at SNR values of 2, 16, and 25 dB because we have previously used these values to give insight into the jamming of the signal. The JNR is set to 10 dB.

At an SNR of 2 dB, as shown in Fig. 6.1, shows that a BLER of 100% is achieved at any value of ρ. The SNR is low enough where no matter what the victim signal cannot recover from the jamming. This makes sense since the signal-to-jammer power level is low where the jammer power dominates the transmission. At an SNR of 16 dB, as shown in Fig. 6.2, the BLER significantly decreases. The codeword is able to correct bits for a satisifactory BLER for the system. Examining the BLER over ρ, the most effect transmission strategy for the jammer would be to use low values of ρ to effectively cause disruptions to the victim transmission signal. As ρ increases, and instantaneous power and pulsing of the jamming signal decreases,

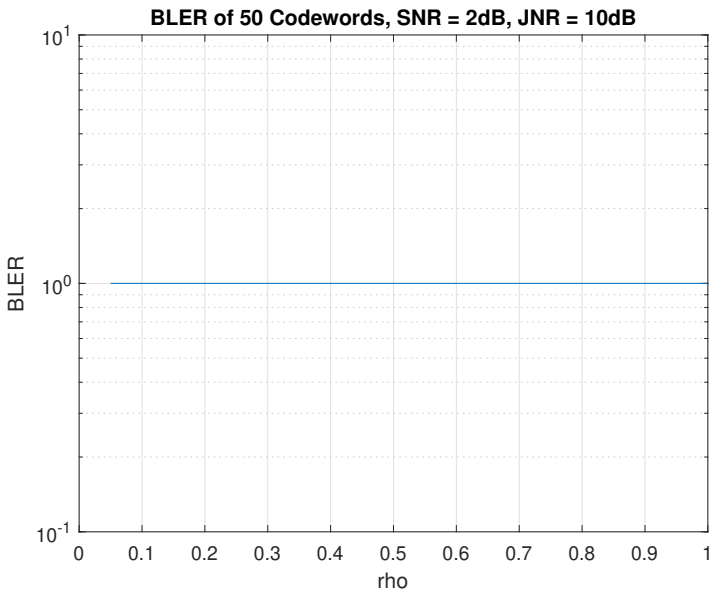

Figure 6.1: BLER of BPSK over ρ with 3/4 rate LDPC under TD AWGN with SNR = 2 dB and JNR = 10 dB.

the jammer becomes significantly less effective at causing block errors, showing that the average power of the jammer is not high enough to overcome the codewords. As seen in Fig. 6.2, low BLER are achieved, so as SNR is increased the victim signal dominates. At an SNR of 25 dB, it is obvious that no BLER is achieved for the jammer.

We also examine the distribution of LLRs to see the effect it has on decoding the received codeword. Both histograms and box plots were used to help visualize how ρ affects the incoming signal and the LLRs of the received bits. Fig. 6.3 display the LLRs across the three SNR values. As SNR increases, the LLR distribution displays a bimodal behavior signifying confidence the code has in the bits. An opposite statement can also be made where the distribution shows the LLRs are large in magnitude and shows bimodal behavior, but the introduced noise could have pushed the LLRs into the opposite decision regions making it near impossible for the code to recover the code words.

The box plots are now examined to give more insight into the histograms of LLRs. The box

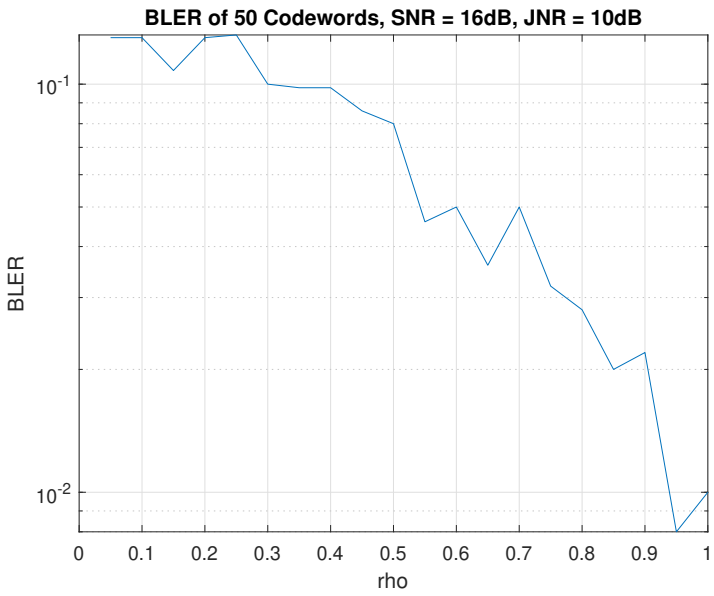

Figure 6.2: BLER of BPSK over ρ with 3/4 rate LDPC under TD AWGN with SNR = 16 dB and JNR = 10 dB.

plots are plotted for LLRs across values of ρ equal to 0.05, 0.2, 0.35, 0.5, 0.65, 0.85, and 1, which are shown in Fig. 6.4. In general, as ρ increases, BLER decreases. The smaller values of ρ have a large impact on the ditribution of LLRs. These distributions have a large variance in LLR values as well as containing many outliers showing the effect that ρ has on the codewords and the high BLER achieved. As ρ increases, the LLR distribution variance shrink as well as the BLER decreasing. In high SNR cases, the LLRs still display a large variance but do not have as many outlying LLR points as well as not impacting the victim signal. The code is "confident" in the LLR values in the bits as compared to low SNR cases where it is not sure in the values of the bits or has false "confidence" in the bits due to the noise affecting the signal by a large margin. This shows from the LLRs being large in magnitude despite BLER being high.

For this case, it is advantageous to only consider small values for ρ since they have the largest effect in disrupting the victim transmission. This knowledge would significantly decrease the

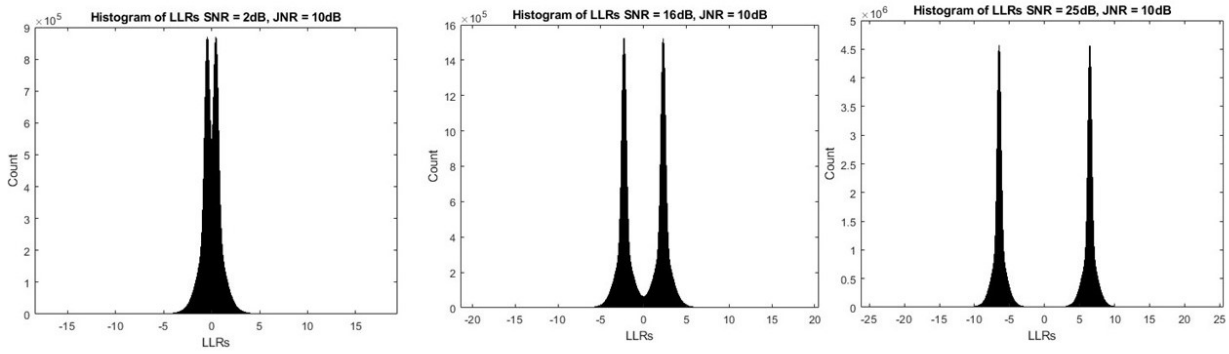

Figure 6.3: Histograms of LLRs across SNR (from left to right: 2 dB, 16 dB, and 25 dB).

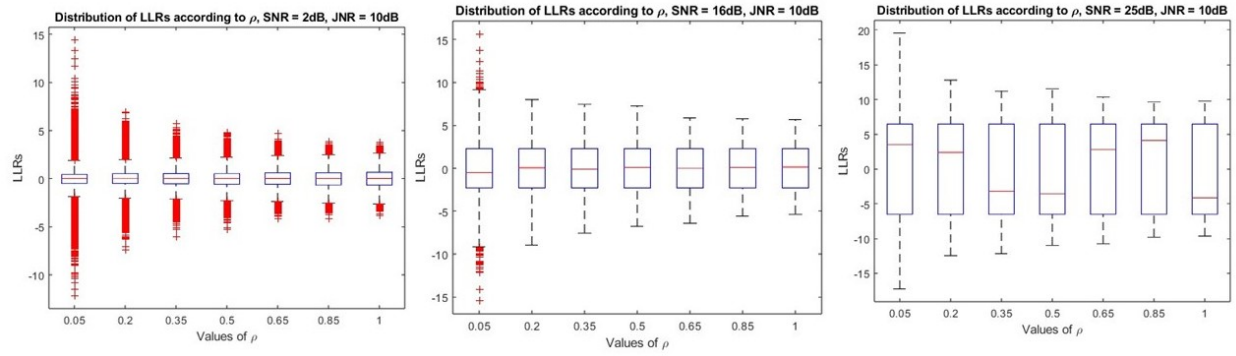

Figure 6.4: Box plots of LLRs across ρ (from left to right: 2 dB, 16 dB, and 25 dB).

action space since there are no advantages for the jammer to learn ρ since it would just choose the lowest value for ρ in each timestep.

6.2.4 Orthogonal Frequency Division Multiplexing

For this section, the victim signal employs an OFDM-modulated signal with BPSK modulated on the subcarriers with a 3/4 rate LDPC code with a block size of 27. The OFDM signal consists of 324 data subcarriers, 83 total guard bands, 41 guard bands on each end of the signal and a null on the center subcarrier, and a cyclic prefix length of 27. There are 2 codeword words per OFDM symbol. We examine the effect of using TD AWGN and FD

AWGN to jam the signal and see the difference between the two. We also implement 3 types of jamming: symbol, subcarrier, and randomly jamming the resource elements within the resource grid, which we will call random jamming. Symbol jamming applies the jammer on an OFDM symbol basis. When ρ is applied, it is on a symbol-by-symbol basis and adjusts power by jamming the subset of selected symbols at a higher instantaneous power, as seen in Fig. 6.5. Subcarrier jamming applies the jammer on a subcarrier basis only. The parameter ρ is applied on a subcarrier-by-subcarrier basis to the selected subcarriers where the power is distributed, and the non-selected subcarriers are not affected at all, as seen in Fig. 6.6. Random jamming pulses the jammer within the subcarrier-symbol grid, as seen in Fig. 6.7. These jamming types are examined such that they are the only jamming type used for a simulation to compare and contrast how jamming different parts of an OFDM signal may affect the block errors achieved by the jammer.

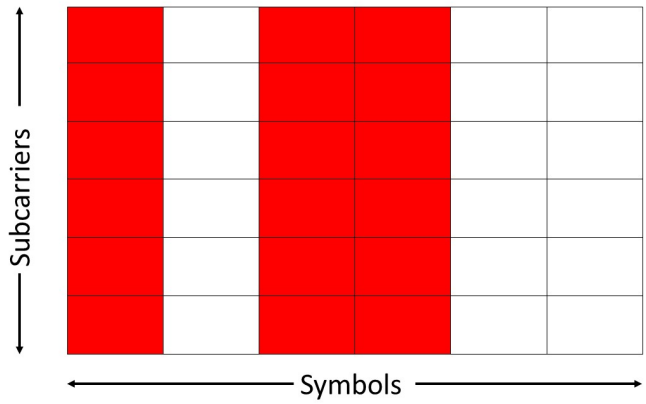

Figure 6.5: Example of symbol jamming where ρ distributes power over time.

6.2.5 Analysis: OFDM - TD AWGN

The BLER simulations shown are that of SNR values of 14 dB to 17 dB since we have previously observed the jammer to achieve 100% BLER with SNR of 2 dB and 0% BLER

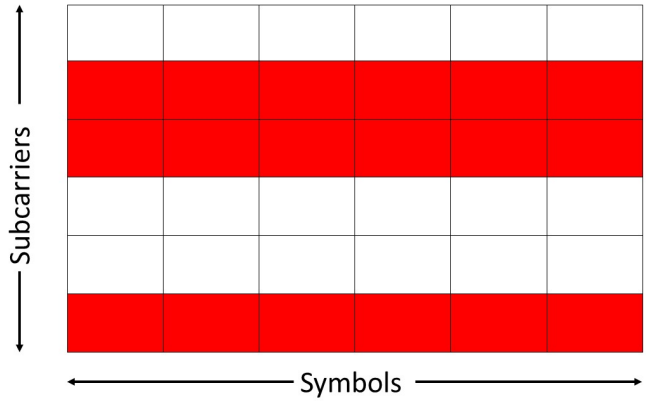

Figure 6.6: Example of subcarrier jamming where ρ distributes power over subcarriers.

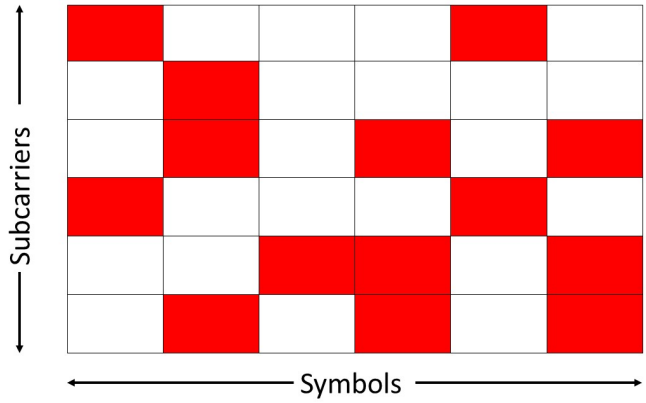

Figure 6.7: Example of random jamming where ρ distributes power over time and subcarriers.

with SNR of 25 dB. We also observed that all jamming types achieve BLERs of nearly 100% below an SNR of 14 dB. Similarly, above an SNR of 17 dB the jammer will achieve almost no block errors. The histograms will still use 2, 16, and 25 dB and any other extra values thought needed to examine the LLR distributions over ρ. All simulations use JNR = 10 dB.

Fig. 6.8 shows the BLERs of the victim at different SNR levels from 14 dB to 17 dB. In between values such as SNR = 15.5 dB were also included to show how small increases in SNR affect the BLER of the victim significantly depending on the jamming method used. Initially at 14 dB, the jamming methods of random and subcarrier jamming perform the best

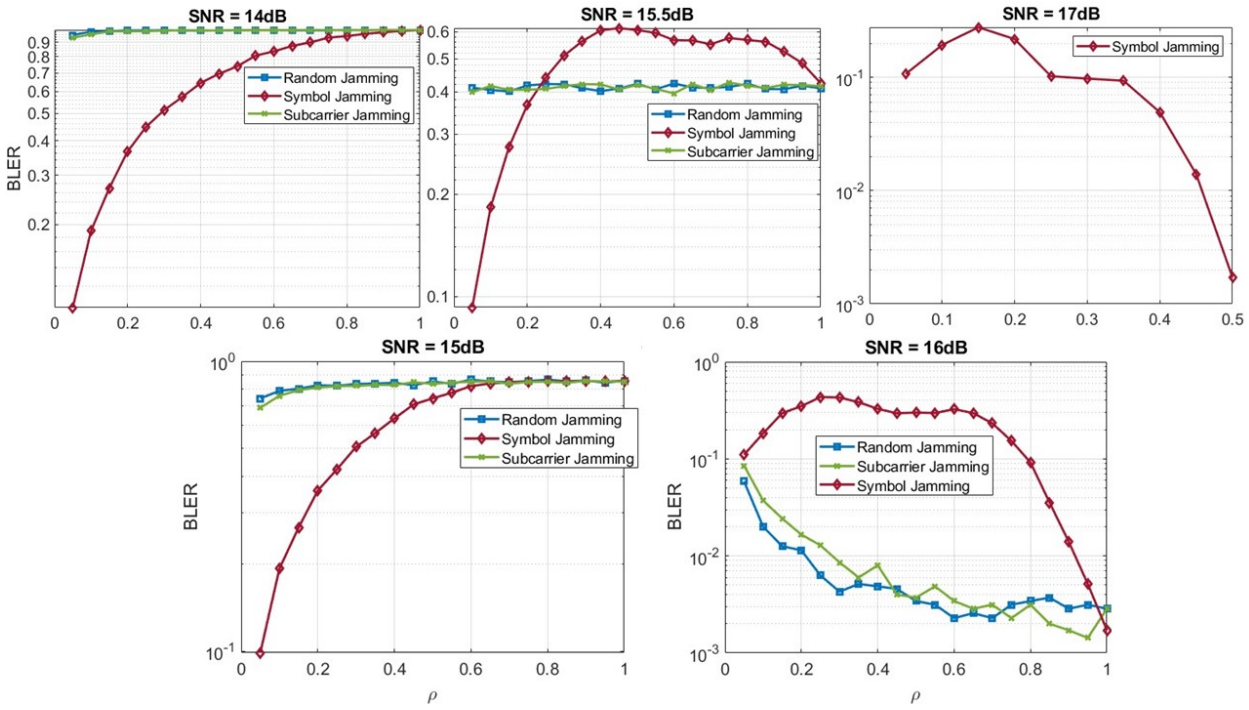

Figure 6.8: Average BLER across ρ using different jamming methods at JNR = 10 dB. Increasing SNR is displayed from left to right: 14 dB, 15 dB, 15.5 dB, 16 dB, and 17 dB.

since at any value for ρ a high BLER is achieved. Symbol jamming does not perform well at lower values for ρ since it is not hitting enough portions of codewords to have a large effect. As SNR increases, the BLER returns for random and subcarrier jamming diminsh quickly while symbol jamming is able to significantly outperform the other two jamming methods. For symbol jamming, initially only middle values of ρ are the most effective. When SNR increases to it's highest point at 17 dB, it is clear that low values of ρ perform the best. The phenomenon at SNR levels between 15-16 dB can be explained by how the codewords are placed in the subcarrier-symbol matrix and the rate of jamming used. Codewords are mapped across subcarriers for a specific OFDM symbol. Thus, symbol jamming concentrates the jammer's power into a subset of the transmitted codewords. At low ρ values, the symbol jamming is only able to jam a small amount of symbols because there are 14 OFDM symbols in each transmission compared to number of subcarriers where there are 435 total subcarriers.

For low ρ values the jammer can jam more subcarriers than symbols because ρ represents a percentage of the subcarriers or symbols being jammed. This explains the performance at low SNRs. Increasing the SNR to 15-16 dB, random and subcarrier jamming no longer have enough instantaneous power to have a large effects on the codeword. Random and subcarrier jamming also do not overlap with whole codewords like symbol jamming does. They will only hit parts of codewords while symbol jamming is guaranteed to hit two codewords per OFDM symbol. Symbol jamming has subpar performance at low and high ρ values because at low values it does not hit enough codewords to have an effect while at high ρ values it does not have enough instantaneous power to effect the codewords. The middle values for ρ therefore have the best combination of jamming rate and instantaneous power to effect the codewords. As SNR is increased more, only lower values of ρ work for symbol jamming because it has enough instantaneous power to effect the codewords. The other two jamming methods can no longer affect the victim signal for reasons discussed above.

The histograms of the different jamming methods over SNR are presented in Fig. 6.9. Both random and subcarrier jamming have similar LLR distributions. The distributions appear normal in shape, and as SNR increases the magnitude of the LLR values increase as well. Symbol jamming shows that the distributions of LLR are more dense in the center of the distributions and have less variance in LLR values. The same pattern emerges as SNR increases; the LLRs increase and are more distinct in their values.

The box plots of LLRs across ρ were also examined as shown in Figs. 6.10, 6.11, and 6.12. Again, random and subcarrier jamming have similar LLR distributions as SNR increases. They are not similar to the distribution shown in Section 6.2.3 where as ρ increases, the distributions of LLRs decrease in variance and have less outliers. Here, the distributions over ρ appear to be very similar and do not change. However, as SNR increases the magnitude of LLRs also increase demonstrating that the FEC code is more sure in the values of the bits

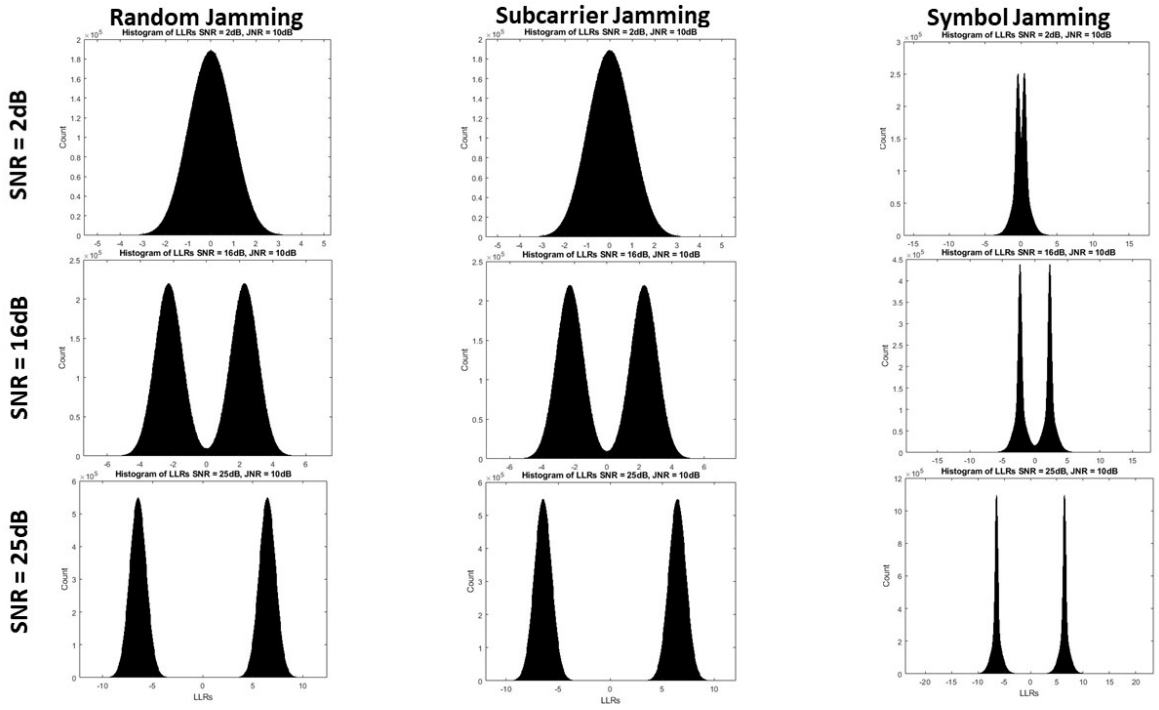

Figure 6.9: Histogram of LLRs using different jamming methods in the TD at JNR = 10 dB.

received. For symbol jamming, the distribution of LLRs over ρ are initially small and then increase and decrease as ρ increases. This may be because no symbols are being chosen to jam when $\rho < 0.1$ since ρ is the probability that a symbol is jammed. If the probabilities are small, and the sampling size of available symbols to jam are small, then no symbol may be chosen to jam. This is similar to what happens in Fig. 6.8 between 15-16 dB. These distributions for symbol jamming help explain the phenomenon which was explained earlier.

A normal LTE or 5G BLER is 2% while BLERs greater than or equal to 10% are unacceptable for successful transmissions. According to these simulations with a jammer operating at a JNR of 10 dB present, an SNR above 17 dB would be needed to achieve successful transmissions.

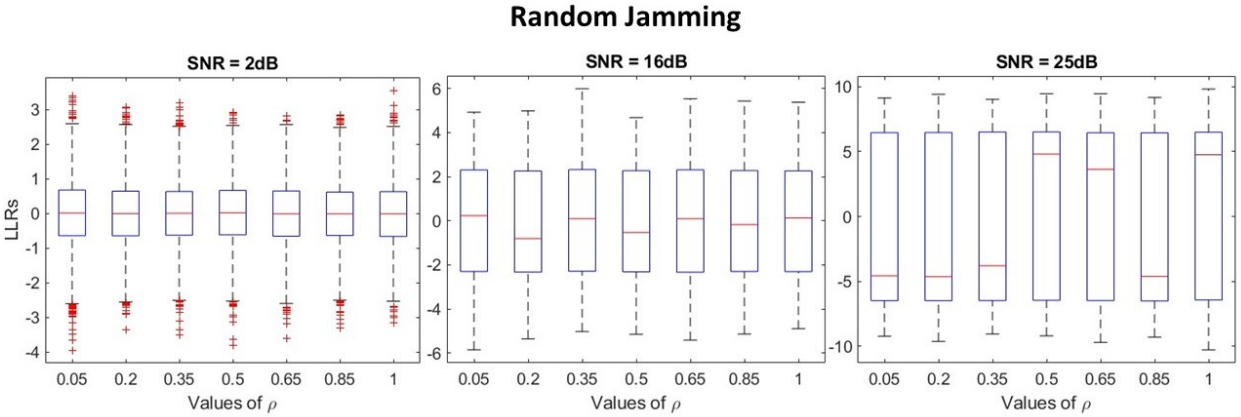

Figure 6.10: Box plots of LLRs across ρ using random jamming and TD AWGN with JNR = 10 dB.

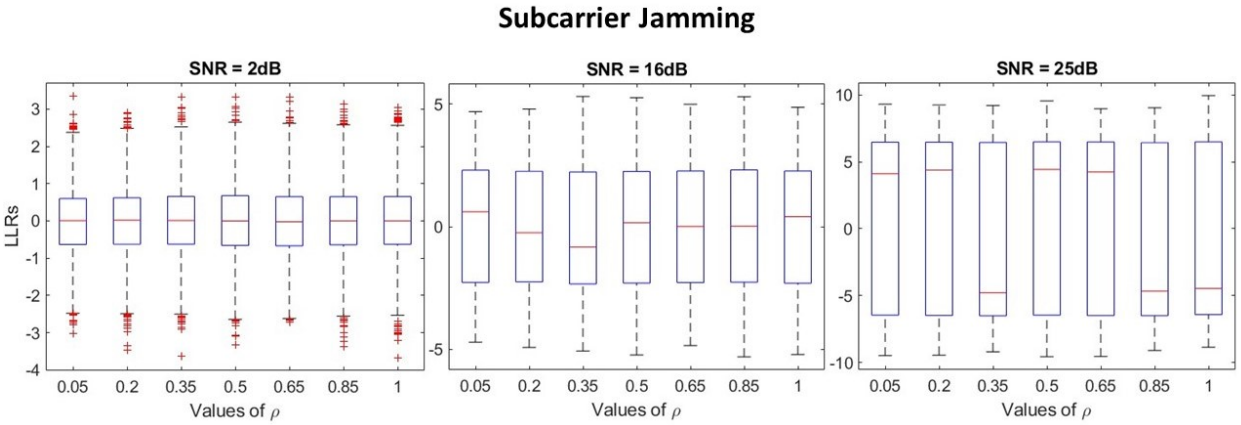

Figure 6.11: Box plots of LLRs across ρ using subcarrier jamming and TD AWGN with JNR = 10 dB.

6.2.6 Analysis: OFDM - FD AWGN

Similar to the previous section, the BLER simulations shown are that of SNR values of 15 dB to 17 dB since we have previously observed the jammer to achieve a 100% BLER with SNR of 2 dB and 0% BLER with SNR of 25 dB across all jamming types. We also observed that all jamming types achieve BLERs of nearly 100% below an SNR of 15 dB for this case. Similarly, above an SNR of 17 dB the jammer will achieve almost no block errors. The histograms will still use 2, 16, and 25 dB and any other extra values thought

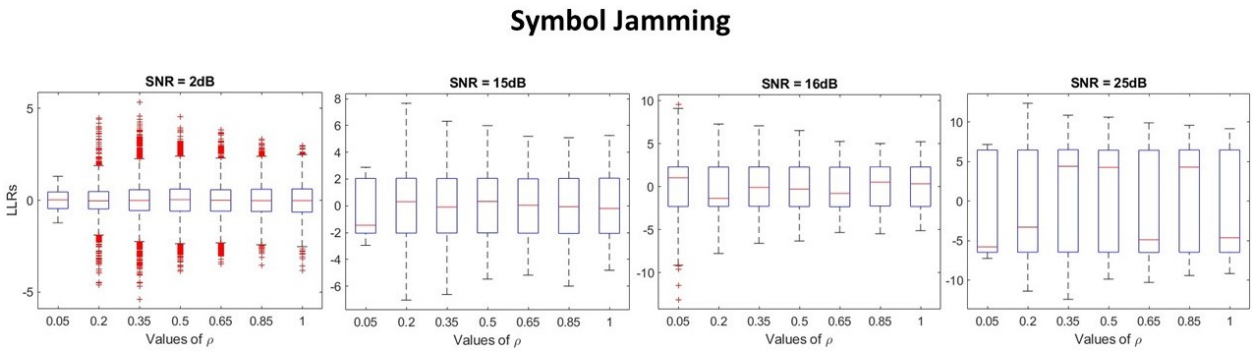

Figure 6.12: Box plots of LLRs across ρ using symbol jamming and TD AWGN with JNR = 10 dB.

needed to examine the LLR distributions over ρ. All simulations use JNR = 10 dB. Fig. 6.13 shows the BLER over ρ for multiple SNR values. As seen previously in Section 6.2.5, the jamming methods perform similarly in terms of BLER and shape of the curves. This is due to random and subcarrier initially having enough average power and hitting a satisfactory amount of parts of codewords to achieve high BLERs. As the SNR increases, the average power and instantaneous power are not enough to affect the codewords leading the code to be able to correctly decode the codewords. Again for symbol jamming, it is guaranteed to hit two codewords per OFDM symbol, so it guaranteed to hit whole code words instead of parts of codewords. As seen previously, the value for ρ also matters since too low of a value will not be able to affect large amount of codewords while too high of a value does not have enough instantaneous power to inhibit the codes decoding ability. The middle values for ρ place a good balance on the jamming power and amount of symbols hit by the jammer. As SNR increases, the decoding ability of the victim increases, so higher instantaneous power is required to jam the victim. In turn, the jammer hits less symbols and causes less block errors. Fig. 6.14 display the histograms of the LLRs over SNR. The histograms in Fig. 6.14 display similar traits to one another regardless of the jamming method. They are also strongly weighted in the center with low variance in the distribution. This may be an affect of AWGN that is applied in the FD as opposed to the TD since FD jamming is able to focus

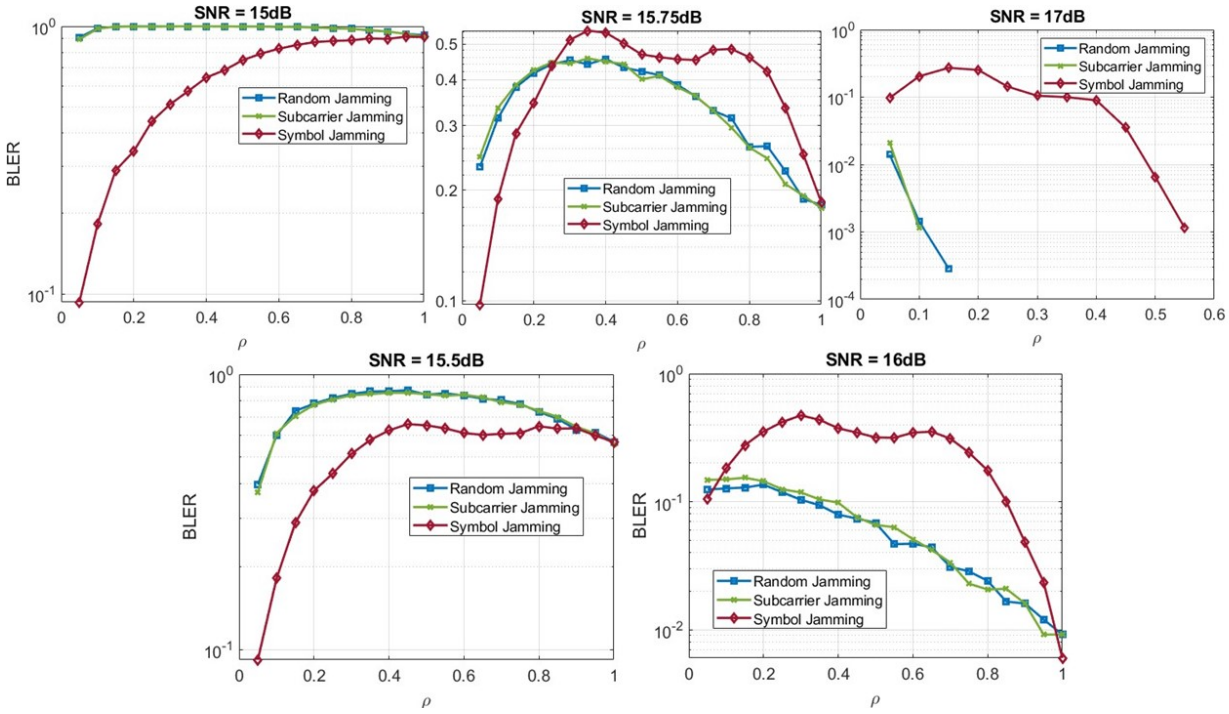

Figure 6.13: Average BLER across ρ using different jamming methods at JNR = 10 dB. Increasing SNR is displayed from left to right: 15 dB, 15.5 dB, 15.75 dB, 16 dB, and 17 dB.

power in the data-carrying subcarriers. However, TD jamming spreads the power across an entire OFDM symbol which includes the guard bands and cyclic prefix. This is similar to the system model seen in Chapter 4.

Figs. 6.15, 6.16, and 6.17 show the LLR distributions over ρ for all jamming methods. Random and subcarrier jamming perform similarly to each other and follow the same pattern as in Section 6.2.3 where as ρ increases, the distribution and outliers decrease. Symbol jamming follows the same distribution pattern seen in Section 6.2.5 where the distribution is initially small, but as ρ increases the distribution increases and then shrinks again. This help support our statements made above of the BLER curves by showing the trend in LLR distribution.

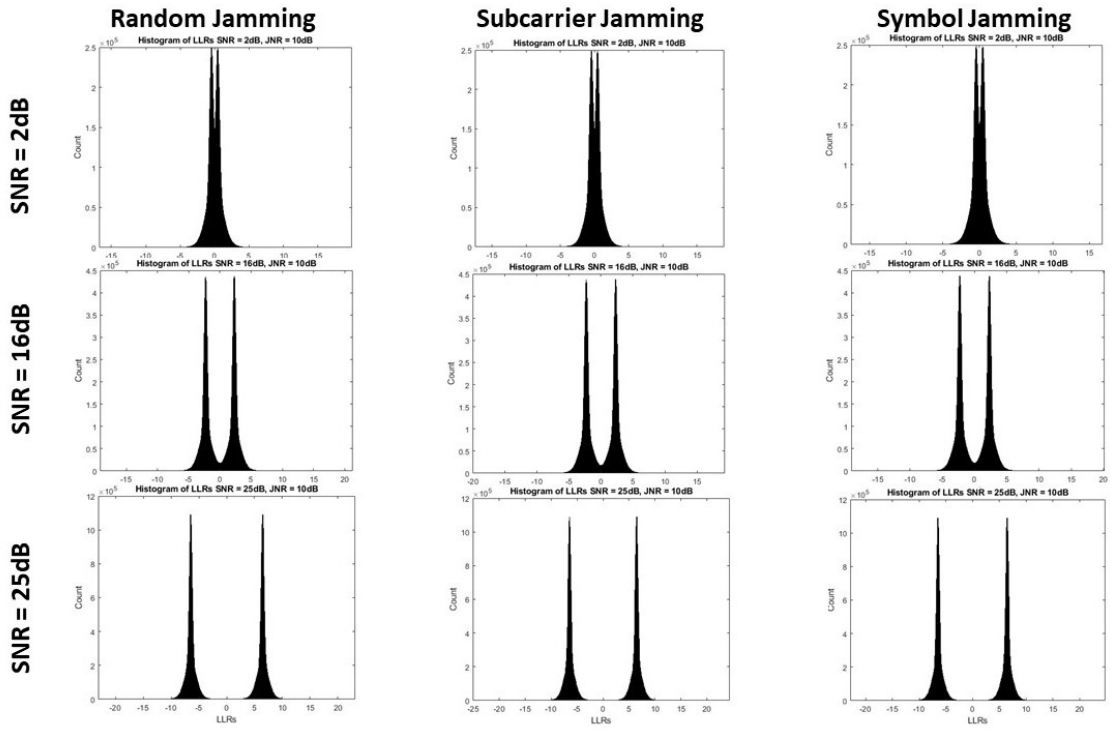

Figure 6.14: Histogram of LLRs using different jamming methods in the FD at JNR = 10 dB.

6.2.7 Conclusion of Jamming a Coded Signal

For the OFDM cases, the BLER has unique results depending on the jamming method and ρ value used. The lowest ρ value is only ideal in certain situations, so the bandit being able to learn about ρ should still be included in the action space. The above analysis gives insight in how jamming a coded system might work, but what is unique to these cases is how the codeword is layed out and how many codewords are in each symbol. The same results might not be achieved if one codeword was dedicated to one subcarrier-symbol block. These results are still useful because targeting whole codewords instead of parts of codewords seem to be the most effective in causing transmission errors.

If the victim attempts to keep the SNR at a specific value, the jamming may have a large impact on the victim signal. This depends on the signal-to-jammer ratio. For our case, we

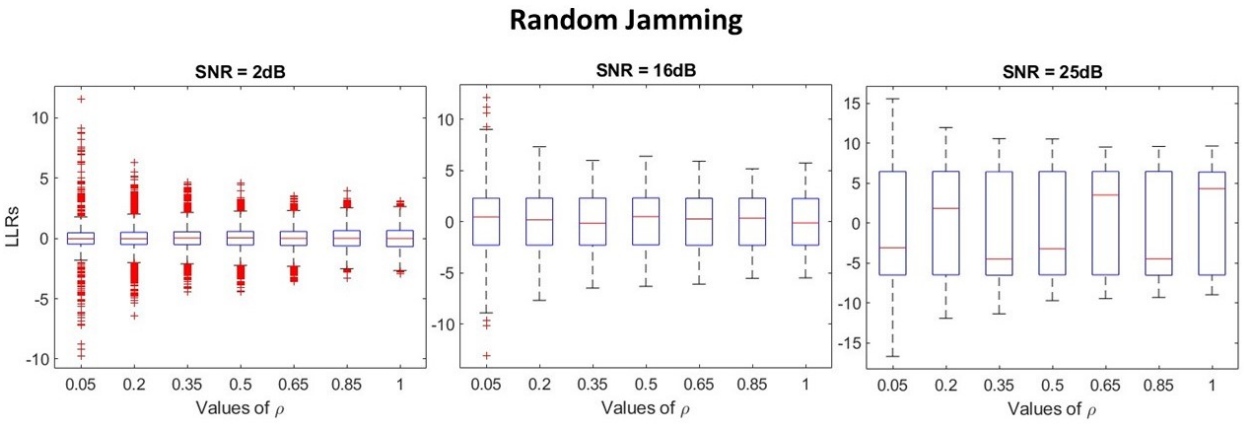

Figure 6.15: Box plots of LLRs across ρ using random jamming and FD AWGN with JNR = 10 dB.

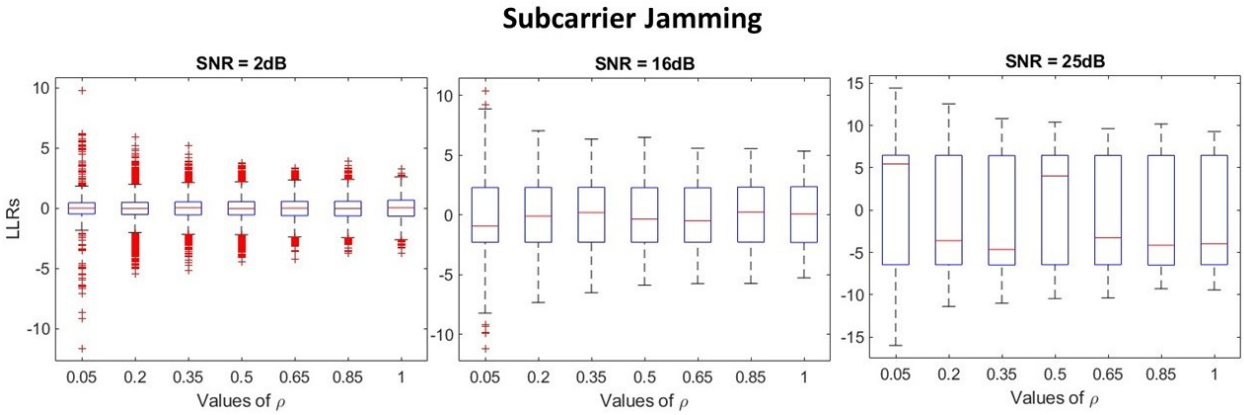

Figure 6.16: Box plots of LLRs across ρ using subcarrier jamming and FD AWGN with JNR = 10 dB.

demonstrated the jammer can have a large impact on the victim signal if the SNR is kept below 17 dB and the JNR is 10 dB. If the victim is able to control its modulation and coding scheme to adapt for a more robust modulation and coding scheme, this would significantly impact the jamming since the jammer would not be able to observe high BLERs that it was previously observing. It would still count this as a success since it would lower the victim's throughput. However, if reinforcement learning is used by the jammer, it would significantly impact the learning because of the lower BLERs seen as a result of the victim using a more robust scheme. If the victim implements hybrid automatic repeat request

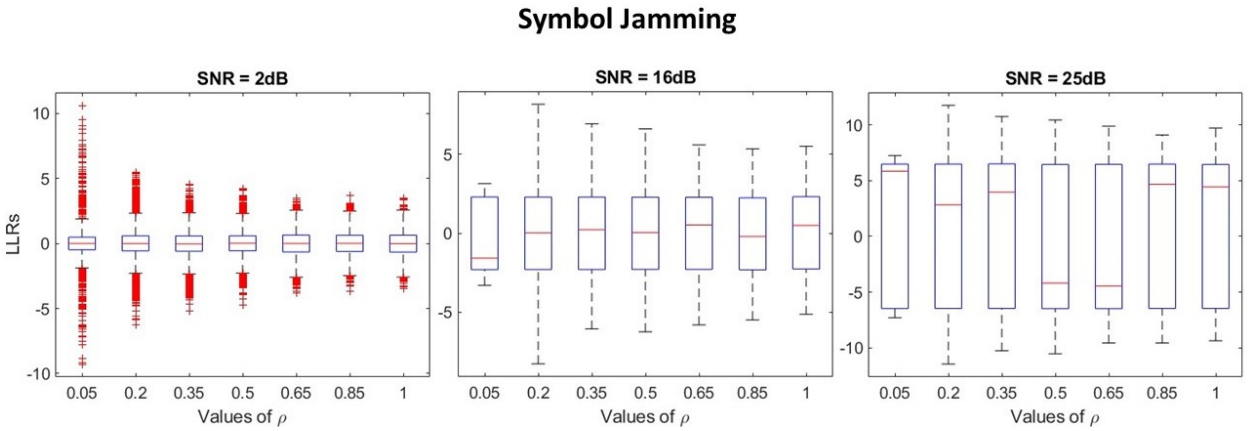

Figure 6.17: Box plots of LLRs across ρ using symbol jamming and FD AWGN with JNR = 10 dB.

(HARQ) processing, this would significantly improve the victim signal's overall performance and prevent the jammer from effectively disrupting the victim signal. We explore the victim being able to use HARQ processing in the following section.

6.3 Jamming a 5G-Based System

We now reintroduce the reinforcement learner to jam a 5G-based system. For this particular system, we consider the downlink shared channel (DL-SCH) that only transmits the physical downlink shared channel (PDSCH). The DL-SCH has many capabilities to increase performance of the system such as multi-antenna systems, beamforming, hybrid automatic repeat request (HARQ), use of demodulation reference signals (DMRS), and use of phase tracking reference signals. For our use case, we initially simplify the victim transmission system and then add capabilities back in to monitor the overall performance of the bandit. We also consider how a jammer might obtain reward feedback in the form of observed acknowledgment/non-acknowledgement (ACK/NACK) resulting from the victim's attempts to decode the jammed data transmission. Moreover, we consider unreliable observation of

this reward information (eg. ACK/NACK) as would be expected with practical detectors.

6.4 System Model

The victim signal in the TD is modeled by Eqn. 4.1. The SNR is modified accordingly by Eqn. 6.2 to account for the difference between signal bandwidth and sampling frequency:

$$SNR_{adjusted} = SNR_{initial} - 10\log_{10}(\frac{f_s}{BW}), \tag{6.2}$$

where $SNR_{adjusted}$ is the SNR relative to signal bandwidth, $SNR_{initial}$ is the designated SNR at the wideband sampling rate, f_s is the sampling frequency, and BW is the bandwidth.

Initially, the victim has the following capabilities: channel estimation, noise estimation, equalization, and scaling LLRs by the channel state information. We assume the victim has no timing or phase offset to correct. We later allow for the transmitter to use HARQ processing to examine how the bandit learns and performs.

The jammer signal in the TD is modeled by Eqn. 4.3. The jammer will only implement FD jamming schemes since that was proven from Chapter 4 to be the most effective schemes against OFDM. The FD received victim signal is represented as Eqn. 4.6.

An important question regarding practical implementation of RL for jamming is how the RL agent obtains its reward feedback. In a real life scenario, as shown in Fig. 6.18, a UE is blindly decoding every millisecond the physical downlink control channel (PDCCH) information. If decoded, an identifier will point to where the physical downlink shared channel is located and also the allocation information for the eventual uplink transmission by the UE. If the PDSCH is successfully decoded, the UE will send an ACK on the uplink transmission either located on the physical uplink control channel (PUCCH) or physcial uplink shared channel

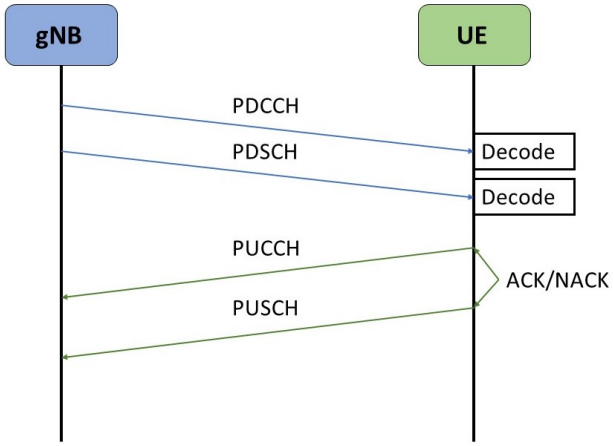

Figure 6.18: Overall transmission diagram between a base station (gNB) and a UE.

(PUSCH). If a block error occurs due to failure to decode the PDSCH, the UE will transmit a NACK. Depending on if HARQ processing is in use, the base station will either retransmit information or not retransmit the information. In the presence of extreme interference or jamming, the UE may not process that the PDCCH was sent. In this case, the UE will not be aware of the need to transmit an ACK/NACK since it did not receive the PDCCH, and thus was not aware of the PDSCH. To recap, a summary of the three scenarios is shown here:

1. Miss PDCCH. → No attempt to decode PDSCH → No ACK/NACK transmission.
2. Detect PDCCH → Detect PDSCH → Transmit ACK.
3. Detect PDCCH → Fail to decode PDSCH → Transmit NACK.

There are multiple options for the jammer to realistically observe ACK/NACK information. One option is to assume the jammer has knowledge of the victim signal and is able to decode uplink information in order to know whether an ACK or NACK was sent. This adds more complexity to the jammer model to make these observations. An energy detector can also be implemented and used in multiple methods. In a simple method, the jammer can use an

energy detector, as represented by Eqn. 6.3

$$R_j(\mathbf{x}) = \sum_{k=0}^{N-1} |v(k)|^2, \tag{6.3}$$

where $R_j(\mathbf{x})$ is the jammer receiving energy from observing an ACK/NACK and $v(k)$ is the victim's transmission signal of an ACK/NACK. Finally, the jammer can monitor the utilization drop of the uplink, as shown in Fig. 6.19. An abruptly lower utilization of the uplink channel during jamming would likely be associated with the UE not receiving the PDCCH. If drops in power or utilization of the uplink channel occur, the jammer assumes that the jamming was effective and has successfully disrupted the victim's communications, as shown by the red arrows in 6.19.

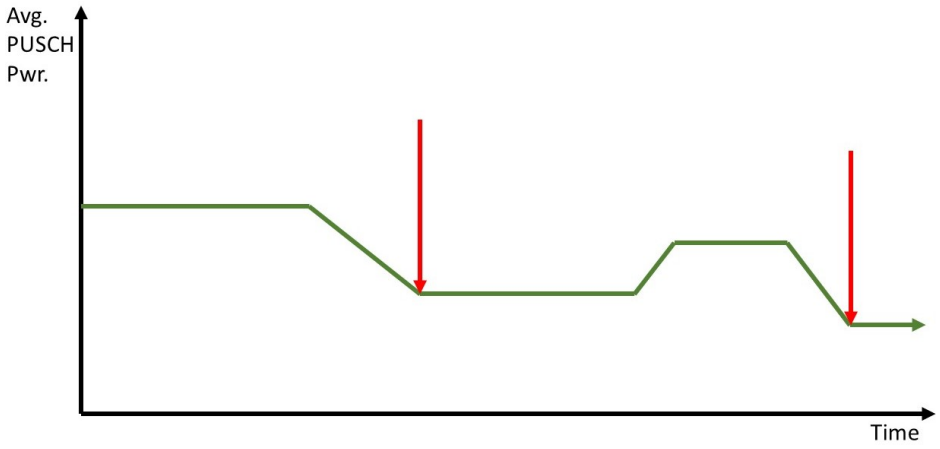

Figure 6.19: Example of monitoring average power of the uplink to determine successes in jamming schemes.

In all of the above scenarios, the jammer will receive some partial or noisy observation of the reward. To account for this, we simply implement the probability of the jammer observing an ACK as $(1 - \lambda)$ and not observing an ACK as λ. This can be viewed as the probability model, shown in Fig. 6.20. The goal is to characterize the impact of the unreliable reward information on the bandit algorithm as a function of the reliability parameter λ.

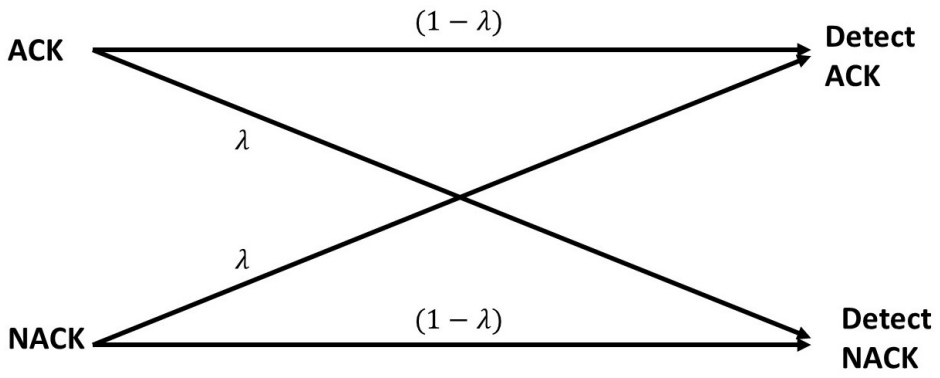

Figure 6.20: Probability diagram of the jammer correctly detecting an ACK or NACK.

6.5 Learning Problem

We again implement the contextual linear bandits algorithm seen in Chapter 4. At each time step t, the jammer will select an action $a_t \in \mathcal{A}$. The action space is composed of three components: ρ, modulation scheme, and jamming method. From Chapter 4, we saw that the FD jamming schemes performed the best in terms of SER. We now only include FD jamming schemes for the bandit since we were able to analyze previous performance. This signaling set is composed of AWGN, BPSK, BPSK $\pi/4$, QPSK, and QPSK $\pi/4$. We still consider ρ as part of the action space since we saw previously in the last section that ρ can have an impact on the jammer and what jamming modulation schemes it decides to jam with. ρ is defined as $\rho \in (0, 1]$, and is discretized by M: $\{1/M, 2/M, ..., 1\}$. We now consider different jamming methods to include in the action space. We assume the jammer has knowledge of the DMRS locations. The jammer now has the ability to jam the PDSCH data only, the DMRS only, or randomly within an entire time slot (i.e. both PDSCH data and DMRS). The instantaneous power provided by the selection of ρ will be adjusted accordingly depending on the jamming method selected to attain the same average JNR seen by the victim throughout the simulation. This is because when jamming only the PDSCH data or only the DMRS, these only account for a fraction of the subcarrier-symbol block. The power after scaling by

ρ must be adjusted to keep the same average power seen at the victim the same no matter the jamming method selected. Otherwise the power will be lower if not normalized.

After the action is selected, the jammer will try to jam the victim with the selected jamming scheme. We no longer consider SER as a metric for the cost function like in Eqn. 4.8. Since we are dealing with a coded system, we instead use BLER. Eqn. 4.8 is modified as

$$C_t = max(BLER_t - BLER_{target}, 0)/JNR_t \tag{6.4}$$

where $BLER_t$ is the observed BLER at time step t, $BLER_{target}$ is the target BLER, and JNR_t is the average JNR at time step t. As previously mentioned in Chapter 4, JNR_t is included in this equation to capture the efficiency of resource usage [10], but for the current case, we use a constant JNR in order to examine the effects of the RL under a fixed average power constraint. We also consider unreliable reward feedback by the jammer having the ability to observe victim ACKs/NACKs. As mentioned previously, there is a probability, $(1 - \lambda)$, associated with observing an ACK. If an ACK is observed the jammer considers that as no block error has occurred. The probability with incorrectly observing a NACK when an ACK is actually sent is represent by λ. If the jammer observes a NACK or nothing is observed, the jammer considers that a block error. We hope to observe the jammer's performance under this unreliable feedback, as it will effect the cost function and context vector. This will inhibit its ability to learn an effective strategy and its overall learning rate.

We then construct the context vector from Eqn. 4.9. We can then use the proposed linear jamming bandits algorithm from Thornton and Buehrer, shown in Alg. 5 [10].

6.6 Analysis

The victim signal is encoded by a LDPC code with a code rate of 0.54 and modulated with 16QAM. This is the lowest code rate available for 16QAM in the modulation coding scheme (MCS) table. A 15kHz subcarrier spacing is used to transmit the signal. The data of the signal is initially spread over 612 subcarriers in a PDSCH grid. A guard band of length 412 is added to the signal where each side of the data-carrying subcarries is attached with a guard band length of 206 guard bands. After OFDM-modulation has taken place, cyclic prefixes of either length 80 or 72 are added to each OFDM symbol. The sampling frequency of the signal is 15.36MHz while the bandwidth of the signal is 9.18MHz. The bandit and jammer are assumed coherent for these simulations. Each time step, 4 frames that are 10ms long are sent out by the victim. The BLER is averaged for those 4 frames and returned as an observation for the bandit. The value for τ, the success parameter, was set to 0.5 for all simulations to compare fairly. As seen previously in Chapter 5, the value of τ may affect the learning rate of the bandit. For these simulations, we cumulatively average one simulation between time steps, and then average 20 simulations across time steps. While averaging the BLER across time steps of multiple simulations provides a better sense of absolute performance of the jammer, we instead observe the average cumulative BLER over multiple simulations because this method allows to better visualize the comparison of different algorithms or scenario parameters such as λ. Our interest is in relative performance difference between two results.

We first consider no HARQ processing and perfect reward feedback of the ACK/NACK process. After, we add the ability for the victim to conduct HARQ processing and compare results. Lastly, we add unreliable reward feedback to observe the jammer learning ability and learning rate.

6.6.1 Perfect ACK/NACK Feedback without HARQ Processing

As mentioned previously, the following simulation results consider no HARQ processing and perfect observation of the ACK/NACK process. The victim signal has an SNR = 24 dB, and we simulate the jammer signal having JNRs of 5.2, 7.2, 9.2, and 11.2 dB.

We first examine the victim transmission under a JNR = 11.2 dB, and then decrease the JNR from there to observe the results. Fig. 6.21 shows these results. The jammer is able

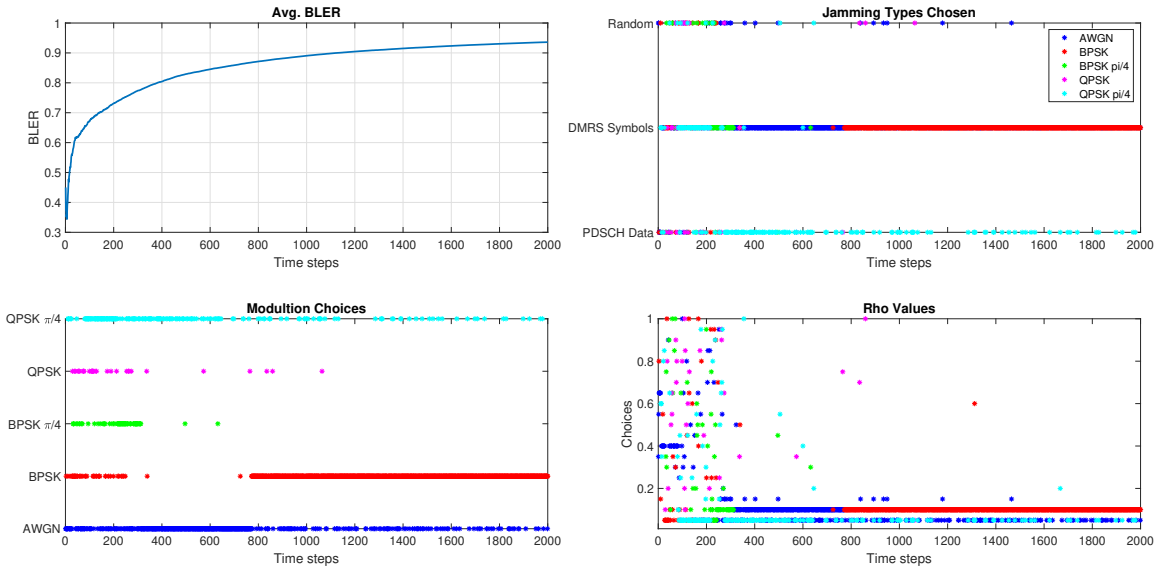

Figure 6.21: Collective results of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 11.2 dB and SNR = 24 dB.

to cause extremely high BLER to the victim transmission. For reference, a BLER of 2% is normal while BLER of 10% is unacceptable. Here the jammer is able to learn to cause a 95% BLER to the victim, intitially starting at 60% BLER. It begins to learn approximately 200 time steps into the simulation. This equates to approximately 8 seconds. While the BLER achieved is impressive, it is important to note of some key items: there is only one codeword per 1,024 subcarrier-14 OFDM symbol grid. In Section 6.2.4, there are 2 codewords per

OFDM symbol. This comes out to 28 codewords for the subcarrier-OFDM symbol grid. The victim will be able to decode more codewords in the latter scenario at smaller SJR than the former at a higher SJR. In this scenario, the jammer is also able to jam the DMRS directly. The jammer clearly found that jamming the DMRS led to higher BLER because it gave the victim receiver incorrect channel and noise estimation factors severely inhibiting equalization of the incoming signal. The BLER examination is important for this section because it shows the convergence rate and how high it converges to a particular BLER. However, the strategies the jammer chooses is of more interest because it shows how the jammer learned to reach that point. The jammer utilized the strategy of selecting to the jam the DMRS the most. The next selected strategy was to jam the PDSCH data itself. This strategy worked because a low SJR is not needed to be able to cause high BLER to this particular system due to the reasons stated above.

It is interesting how the jammer selected certain jamming methods with certain modulation schemes. For example in Fig. 6.21, when the jammer chose to jam the DMRS, the modulation scheme chosen with it was almost exclusively QPSK $\pi/4$ and BPSK. This makes sense since the DMRS in the victim signal uses QPSK to transmit over the AWGN channel. The jammer choosing QPSK $\pi/4$ and BPSK makes sense because the DMRS symbols will get pushed into other regions and will severely affect the channel and noise estimation because of this. We have seen previously that the best way to jam a QAM signal is for the jammer to also use a QAM signal [1]. On the other hand in Fig. 6.21, when the jammer chose to jam the PDSCH data, the jammer exclusively chooses to use AWGN to jam the victim signal. The values for ρ used are almost exclusively low values such as 0.05 and 0.1. In the case of DMRS jamming, this would be advantageous for the jammer to target a subset of symbols to throw estimation off with large power fluctuations. In the case of PDSCH jamming, it might be advantageous for the jammer to target small sets of data to have the LLRs of the received data be large.

As seen in Section 6.2.4, this may be enough to cause bits to not be corrected and cause the whole codeword to be in error. Random jamming was not chosen a lot because it may not be targeting sufficient DMRS symbols or PDSCH data symbols to cause the codeword to be in error. When the jammer chooses jamming PDSCH data or the DMRS, the jammer knows for sure that it is going to hit the DMRS or PDSCH data. For jamming randomly, the jammer does not know how many DMRS symbols it will hit, if any, to cause significant irreparable damage to the codeword.

We next simulated JNR = 9.2 dB, as shown in Fig. 6.22. Again, the jammer is able to achieve high BLERs in a relatively short amount of time. This time the exploiting begins about 400 time steps into the simulation or about 16 seconds into the simulation. The jammer almost exclusively jams the DMRS, and multiple schemes are chosen to do this. QPSK is the dominant choice in later time steps, but all of the QAM modulation schemes are chosen periodically to jam the DMRS. We have seen previously that it is advantageous to choose a type of QAM to jam a victim signal also employing a type of QAM from Amuru and Buehrer [1]. Chapter 4 also supports this and supports using types of BPSK to jam QAM. AWGN is chosen less consistently however possibly because there are many more opportunities for the jammer to choose QAM options to jam the victim. The values for ρ chosen vary on the lower end of the spectrum. They are either consistently 0.05, 0.25, or 0.3. Since the SJR is higher than previously seen, the jammer needs to make sure that more of the DMRS are affected to continue to make sure the estimation correction factors are significantly incorrect to impact the whole codeword.

We next simulated JNR = 7.2 dB, as shown in Fig. 6.23. The BLER is much lower comparatively to Figs. 6.21 and 6.22. The jammer also begins to learn approximately double that of when the JNR = 9.2 dB, but this may be attributed to the value of τ chosen. It is still able to cause considerable damage however because it is able to reach BLERs above 10%. Again,

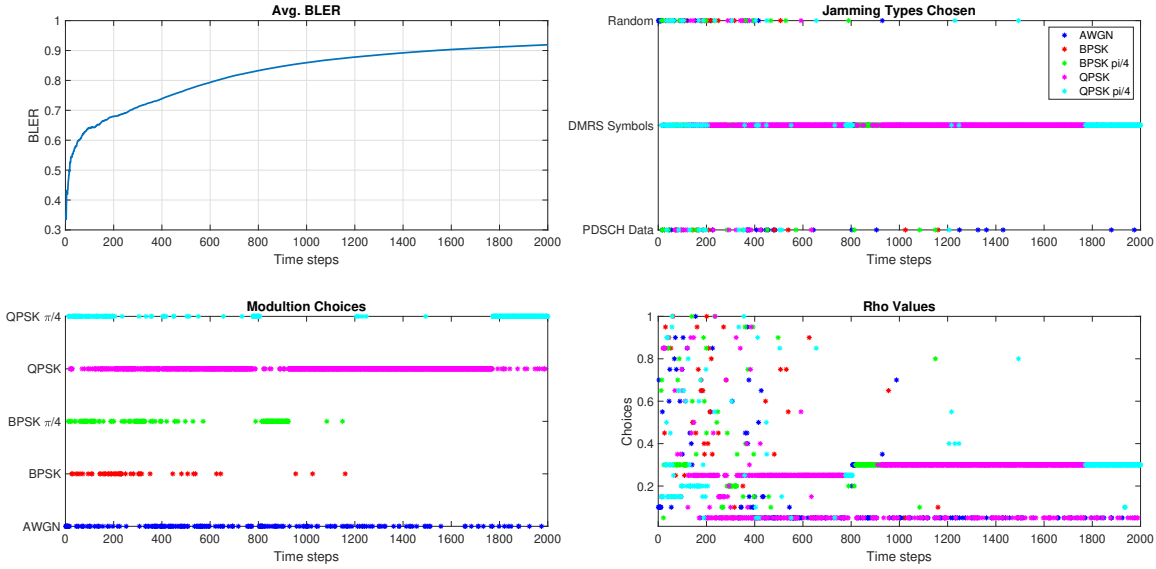

Figure 6.22: Collective results of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 9.2 dB and SNR = 24 dB.

DMRS jamming is chosen as the dominant action. However, the jammer begins to choose the QAM modulations less and AWGN more consistently. Lower values for ρ are also chosen to jam the system. This may be similar to the case in Section 6.2.4 where symbol jamming using AWGN and low values for ρ was the only effective route when the signal-to-jamming ratio became large. For this, DMRS jamming and AWGN at low values for ρ may be the best combination in order to jam most effectively.

Lastly for this section, we simulated at a JNR = 5.2 dB, as seen in Fig. 6.24. The BLER achieved is subpar compared to the other JNR levels, but the SJR is very high. The victim should be able to over come the jamming easily. The jammer exclusively used PDSCH jamming. It is possible that the JNR was too low for the jammer to learn to jam the DMRS. After the learning period, AWGN was exclusively used at a low ρ value. Learning period in this context refers to the actions in which the bandit clearly begins to exploit the victim. PDSCH jamming was used the most because the level for τ set was mainly unachievable for

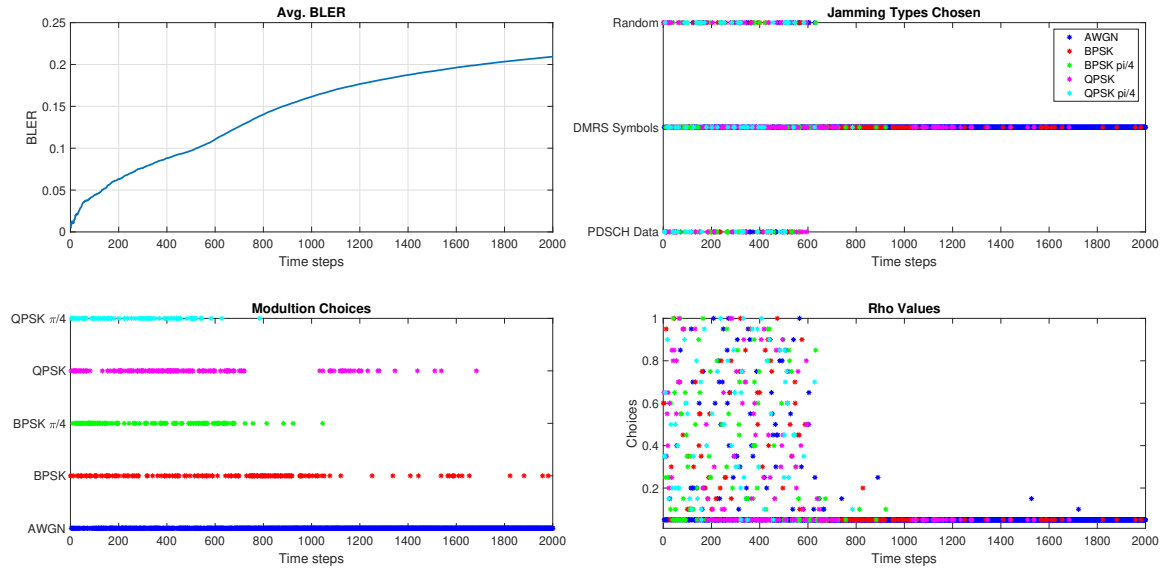

Figure 6.23: Collective results of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 7.2 dB and SNR = 24 dB.

the jammer, so no matter the jamming method chosen it would be highly unlikely to reach a BLER = 50%. However, AWGN does seem to be the best scheme for the jammer to use when presented with a situation where the victim has a considerably higher power level than it. This again goes back to Section 6.2.4 where symbol jamming used with AWGN and low values for ρ was the best option to affect the LLRs.

It is interesting to see how the jammer changes its jamming strategies to adapt to different SJR levels. When the power levels are closer the jammer will chose to jam DMRS using QAM schemes. As the gap in power widens, the jammer will chose to use AWGN, which follows from Section 6.2.4. The values for ρ may vary in the lower values as well depending on the SJR. As the gap widens in power levels between the victim and jammer, the lowest value for ρ will more consistently be chosen which was seen in both Chapter 4 and Section 6.2.4.

We now let the system use HARQ processing to see how the learning rates change, how the

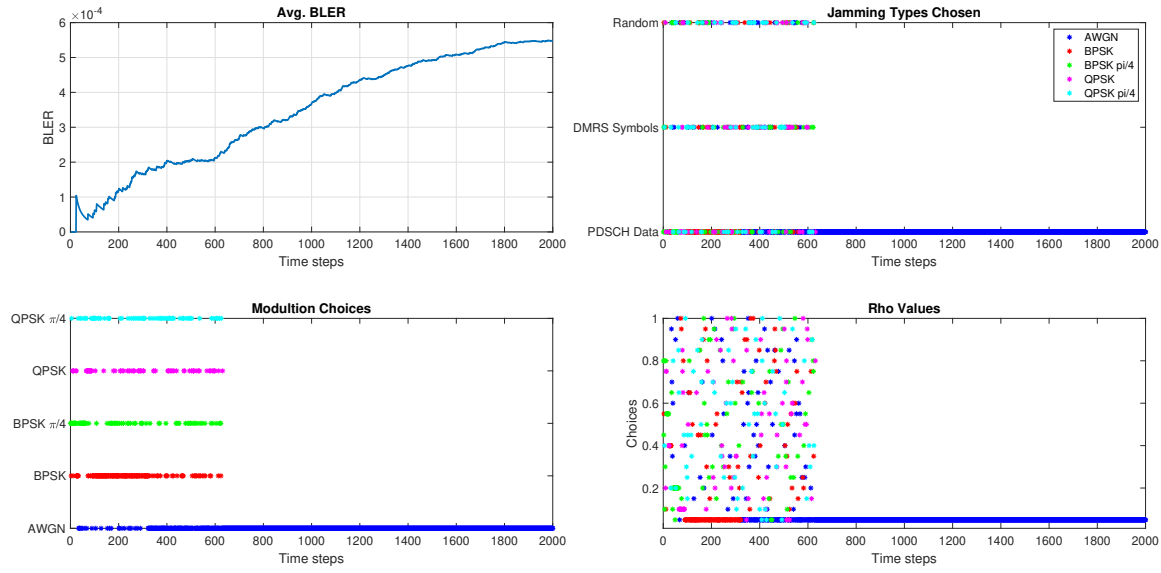

Figure 6.24: Collective results of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 5.2 dB and SNR = 24 dB.

bandit learns, and how BLERs change.

6.6.2 Perfect ACK/NACK Feedback with HARQ Processing

The following results consider the victim being able to use HARQ processing. The redundancy version (RV) sequence used for this process is 0, 2, 3, 1. Again, we start the simulations at a JNR = 11.2 dB, as seen in Fig. 6.25, and then lower the JNR to 9.2 dB, 7.2 dB, and 5.2 dB. HARQ processing gives a significant advantage to the victim because it is able to retransmit the signal and combines the results of that process with the previous received signal process. It is able to do this up to 4 times, which gives a significant advantage to it. We can see from the BLER curve presented in Fig. 6.25 that the jammer performance has significantly degraded, but it is still able to achieve high BLERs. The rate at which it achieves high block errors has dropped significantly as well. The dominant jamming method is jamming DMRS, while the jammer is not able to settle on a modulation scheme. This is

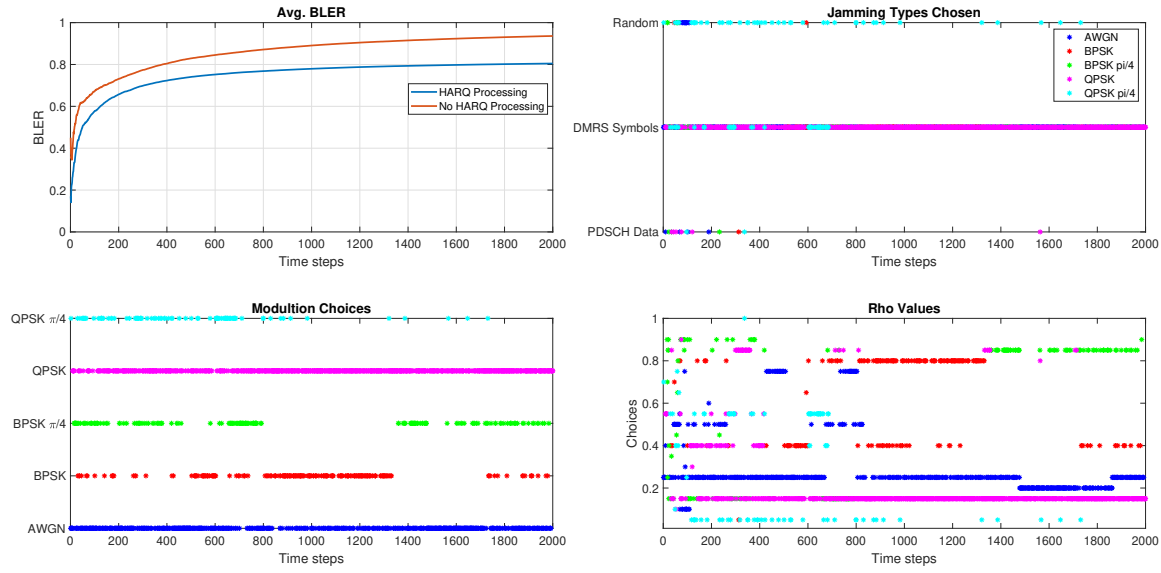

Figure 6.25: Collective results under HARQ processing of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 11.2 dB and SNR = 24 dB.

due to the retransmit process the victim employs. The jammer may think it chooses a viable scheme but since the victim is able to combine received signals, the victim BLER improves significantly while the feedback for the jammer degrades. This also inhibits the rate the bandit is able to learn at and it stagnates quickly compared to if no HARQ processing is used. If Figs. 6.21 and 6.25 are compared, when no HARQ processing is used the jammer is able to settle quickly on a scheme to use. When HARQ processing is used the jammer cannot settle on one scheme to effect the victim as severely compared to Fig. 6.21. This is also seen in how the jammer is not able to settle on a value for ρ. It keeps exploring on the side to see if the BLER improves with these choices.

Fig. 6.26 shows the results of HARQ processing when the JNR = 9.2 dB. Again, we come to similar results where the BLER has degraded significantly, the rate of creating block errors has degraded and stagnated more quickly than without HARQ processing, and the jammer is not able to land on one dominant choice of modulation scheme to use or ρ value to use. It

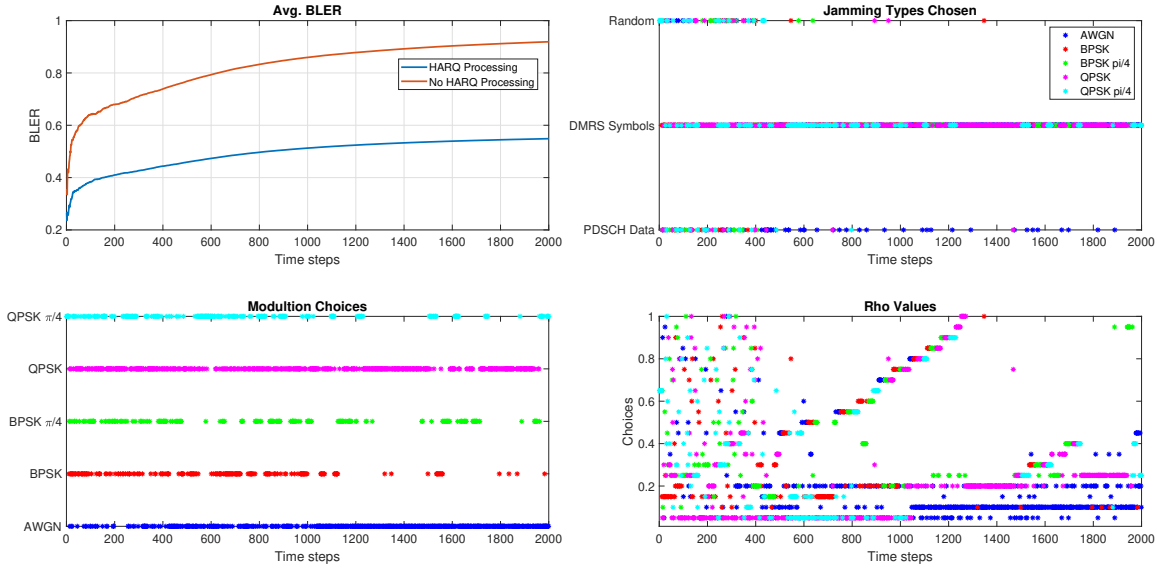

Figure 6.26: Collective results under HARQ processing of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 9.2 dB and SNR = 24 dB.

may have a main scheme it uses, but it still explores its options on the side in the hope that it is able to land on the scheme that significantly improves the BLERs it is able to cause to the victim.

Fig. 6.27 shows the results of HARQ processing when the JNR = 7.2 dB. For this case, the BLER curves are similar. The jammer initially performs better with HARQ processing due to the initial choices made. However, the rate of causing block errors quickly stagnates and the bandit eventually performs better when the victim employs no HARQ processing. The jammer makes similar choices if compared to Fig. 6.23 where the dominant jamming type is DMRS jamming, the dominant scheme is AWGN, and the lower values for ρ are used most often. Again the reason that the jammer under performs even with the similarities is because HARQ processing adds a significant advantage to the victim.

Lastly for this section, the results for a JNR of 5.2 dB are such that it will no longer will have

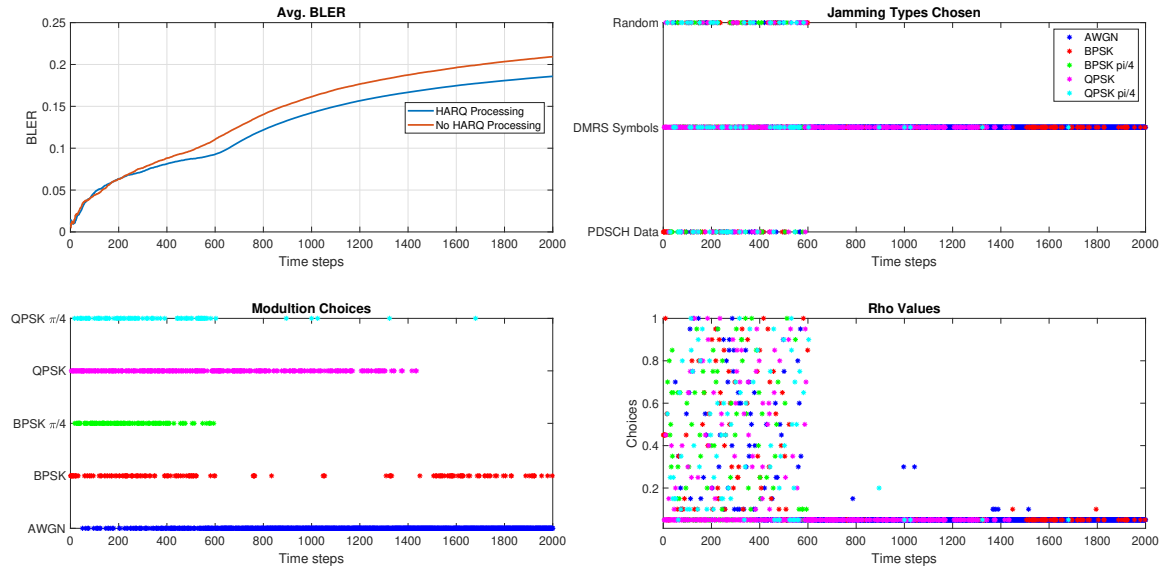

Figure 6.27: Collective results under HARQ processing of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 7.2 dB and SNR = 24 dB.

an effect on a typical 5G target because the BLERs produced are close to zero. The HARQ processing adds a large advantage to the jammer, and when the BLERs are very low and cannot reach the value for τ that was set; the jammer chooses a dominant jamming method and modulation scheme. This is because no matter the values it chooses, it is very unlikely it will reach the BLER that is considered a success. In this case, HARQ processing severely inhibits the jammer from achieving any BLERs, thus lowering its chances of learning any effective scheme. Since these simulations are cumulative, even causing one block error will not make a difference in the overall simulation.

6.6.3 Unreliable Feedback

For this section, the simulation is modified for one 10ms frame to represent a time step. We consider an SNR = 24 dB and JNR = 11.2 dB, 7.2 dB, and 6.2 dB. We examine 0.05,

0.1, and 0.15 for values of λ. We consider the probability of correctly observing an ACK and the probability of correctly observing a NACK to be symmetrical for this case (eg. $\lambda = \lambda_{ACK} = \lambda_{NACK}$). The victim still has the ability to use HARQ processing to attempt to increase its throughput and decrease its BLERs, as seen from the last section. The jammer is now non-coherent to the victim signal's phase.

We begin with analyzing a JNR of 11.2 dB and $\lambda = 0.05$ to show how the probability of observing an ACK/NACK incorrectly can affect the jammer reward function, as shown in Fig. 6.28. Fig. 6.28 not only shows the BLER the jammer believes it obtained, but also the BLER actually obtained from the actions it took, and the BLER obtained by a jammer operating on had perfect observations of ACKs/NACKs while non-coherent to the victim signal. Fig. 6.28 shows that the true BLER achieved from jamming is higher than what the jammer believes it is achieving. This is due to when the jammer achieves a NACK and incorrectly observes that the victim achieved an ACK. This scenario will occur when the JSR is high. The true BLER is also higher than the jammer with perfect observations of ACKs/NACKs. This may be attributed to a dominant selection of a jamming scheme. For the majority of the simulation AWGN is selected as the jamming modulation, as well as $\rho = 0.35$ being chosen and DMRS symbol jamming being chosen. With perfect observations of the ACKs/NACKs, the jammer would spend more time trying to find the optimal strategy or multiple optimal strategies as opposed to choosing an effective strategy to jam the victim in the desired time horizons. This follows from Chapter 5 where choosing an effective strategy in a timely manner appears to achieve better results than finding the optimal strategy or multiple optimal strategies where more time is spent exploring and not exploiting the victim. Regardless, the jammer is easily able to converge to high BLERs to effectively jam the victim.

The next value of λ simulated is 0.1, as shown in Fig. 6.29. Again, we see similar results

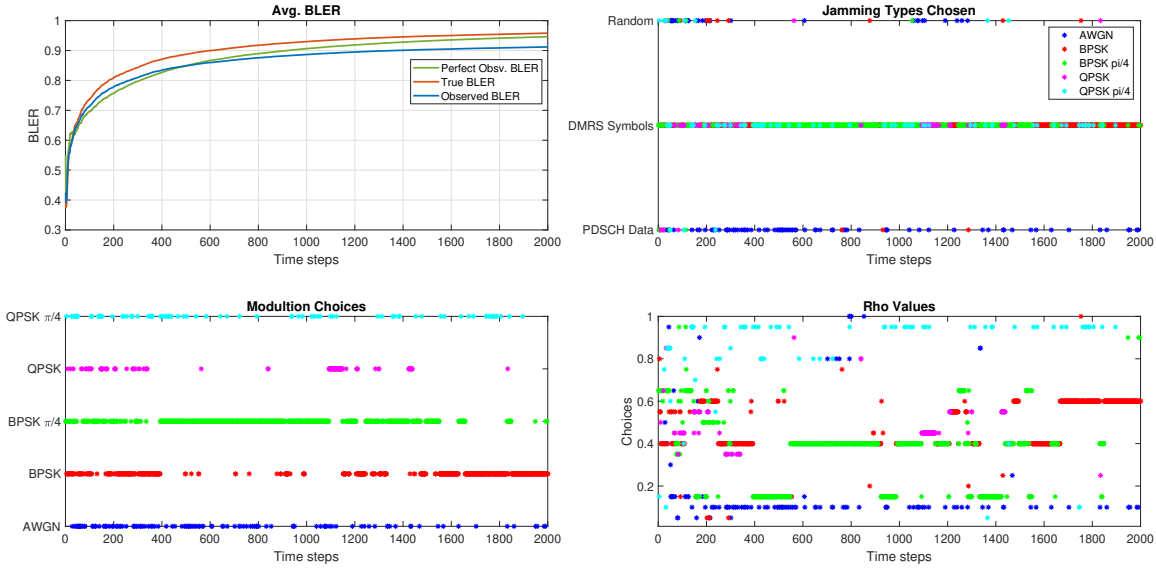

Figure 6.28: Collective results under HARQ processing and unreliable feedback ($\lambda = 0.05$) of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 11.2 dB and SNR = 24 dB.

to Fig. 6.28 where the observed BLER (unreliable feedback) and perfect observation BLER (perfect feedback) is lower than the true BLER the bandit achieves. However, the margin of error between the observed BLER and true BLER has increased, but the true BLER has not changed. This shows that at a high enough JNR relative to SNR, the performance of the jammer may be unaffected. However, the performance the jammer believes it achieves will deteriorate.

The last value of λ simulated is 0.15, as shown in Fig. 6.30. The margin of error has increased from Figs. 6.28 and 6.29. However, the true performance of the jammer has not degraded. The modulation choices appear to be random, but DMRS symbol jamming is still chosen dominantly through the course of the simulation. This shows that if the JNR is high enough relative to the SNR the performance of the jammer may appear to degrade, but will continue to achieve high BLERs.

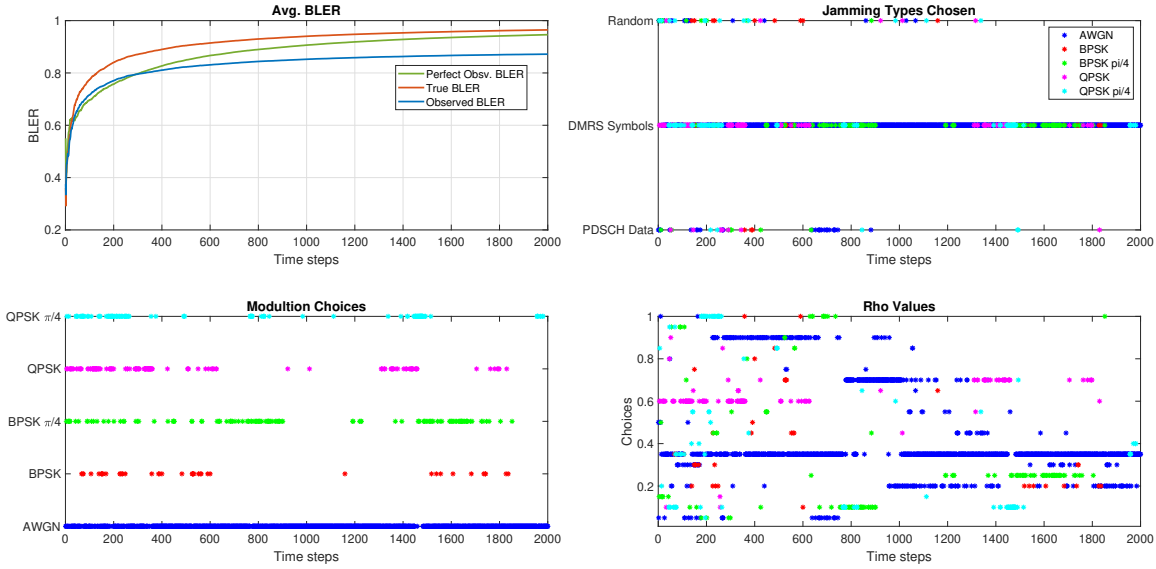

Figure 6.29: Collective results under HARQ processing and unreliable feedback ($\lambda = 0.1$) of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 11.2 dB and SNR = 24 dB.

We now analyze a JNR of 7.2 dB and $\lambda = 0.05$ to give a baseline of how probability of observing an ACK/NACK incorrectly can affect the jammer reward function, as shown in Fig. 6.31. The BLER curves show similar shape, learning periods, and exploitation rates. However, there is a margin of error between the true BLER and observed BLER that the jammer believes it caused. In this case, the margin of error is small, and the bandit still can effectively cause damage to significantly inhibit the victim's ability to communicate since the BLER achieved is above 10%. True NACKs are now infrequent compared to when the JSR is high, so the rewards feedback is now dominated by false NACKs. This leads to the observed BLER now being larger than the true BLER. This is opposite of what was observed when the JNR was 11.2 dB. The learning rate is also comparable to that of perfect observation capabilities and the victim using HARQ processing. In terms of the jammer's actions, there is very little uncertainty in the jammer's decision process meaning the jammer converges to a consistent set of strategies after the learning period. This can be observed in the modulation

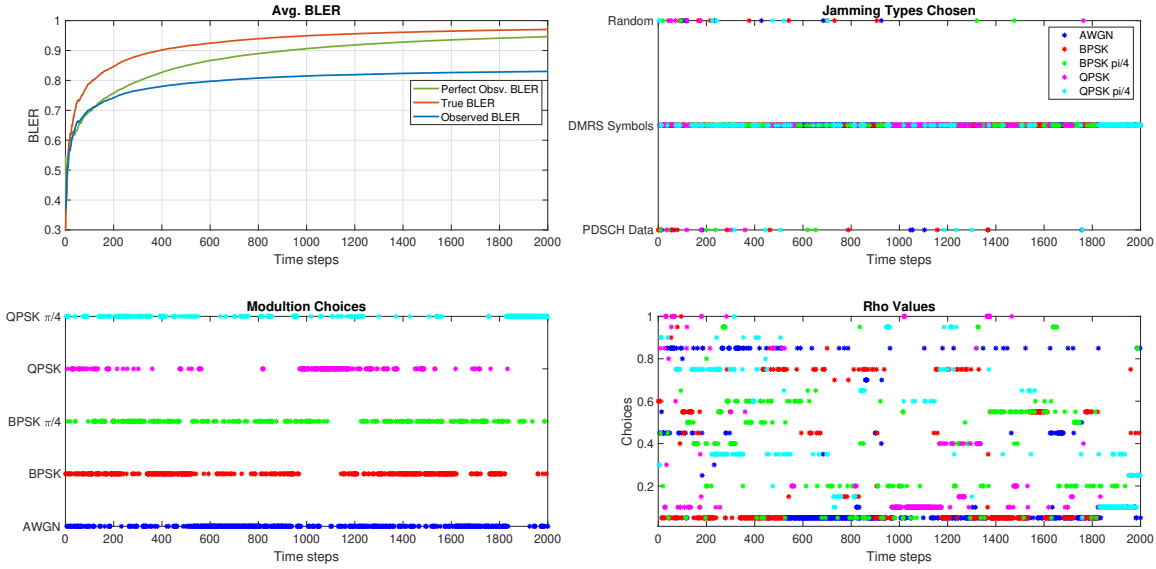

Figure 6.30: Collective results under HARQ processing and unreliable feedback ($\lambda = 0.15$) of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 11.2 dB and SNR = 24 dB.

choices and ρ choices in Fig. 6.31. Some side exploring does occur, such as occasionally choosing AWGN as the jamming modulation and a higher ρ value to use with AWGN, where this indicates the aforementioned uncertainty. This indicates some remaining uncertainty because after the learning period, the jammer tries other jamming strategies to see if it can obtain a higher BLER. The main jamming method is still chosen as DMRS jamming, so the jammer is still able to effectively differentiate the process between choosing the different available jamming methods of PDSCH data jamming, DMRS jamming, or both.

The next value of λ simulated is 0.1, as shown in Fig. 6.32. The BLER curve learning period and exploitation rate appear to have decreased compared to that of in Fig. 6.31. The margin of error between the observed BLER and the true BLER have increased, as well as the margin of error between the true BLER and perfect observation BLER have increased. Uncertainty in the choices in modulation and ρ values have also increased. Side exploring has increased in ρ and modulation scheme choice. The main modulation choices used are QPSK

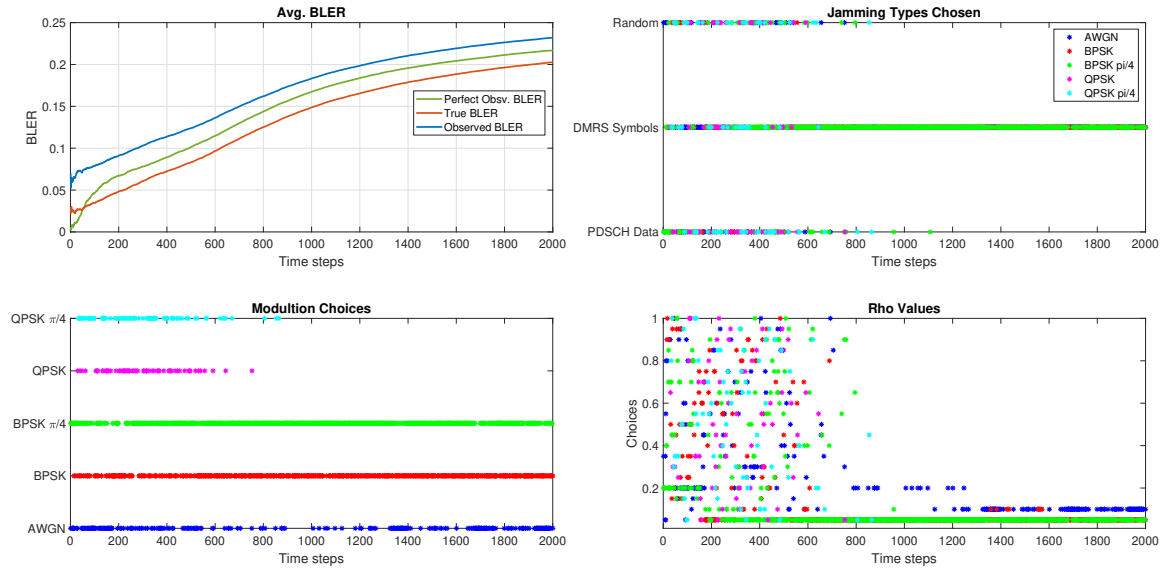

Figure 6.31: Collective results under HARQ processing and unreliable feedback ($\lambda = 0.05$) of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 7.2 dB and SNR = 24 dB.

and AWGN with partial usage of QPSK $\pi/4$ and BPSK $\pi/4$. The transition period between learning period and exploiting period is still clear in these plots where there is an obvious transition at 700 time steps. This means the jammer is still able to learn, but is more unsure of its choices because of the unreliability of the reward function. It will eventually converge to a BLER that will heavily impact the victim receiver (10% BLER in about 7 seconds). The main aspect is that it can still differentiate between jamming methods to effectively jam the victim by choosing DMRS jamming.

The last value of λ simulated for a JNR of 7.2 dB is 0.15, as shown in Fig. 6.33. Again, the margin of error between the observed and true BLER has increased, as well as the margin of error between the true BLER and the perfect observation BLER. This follows from the increase in uncertainty of the reward function, so the jammer does not know if the actions it takes are truly effective or not. The learning period appears unaffected, and the exploitation rate after the learning period has decreased, leaving the bandit less effective

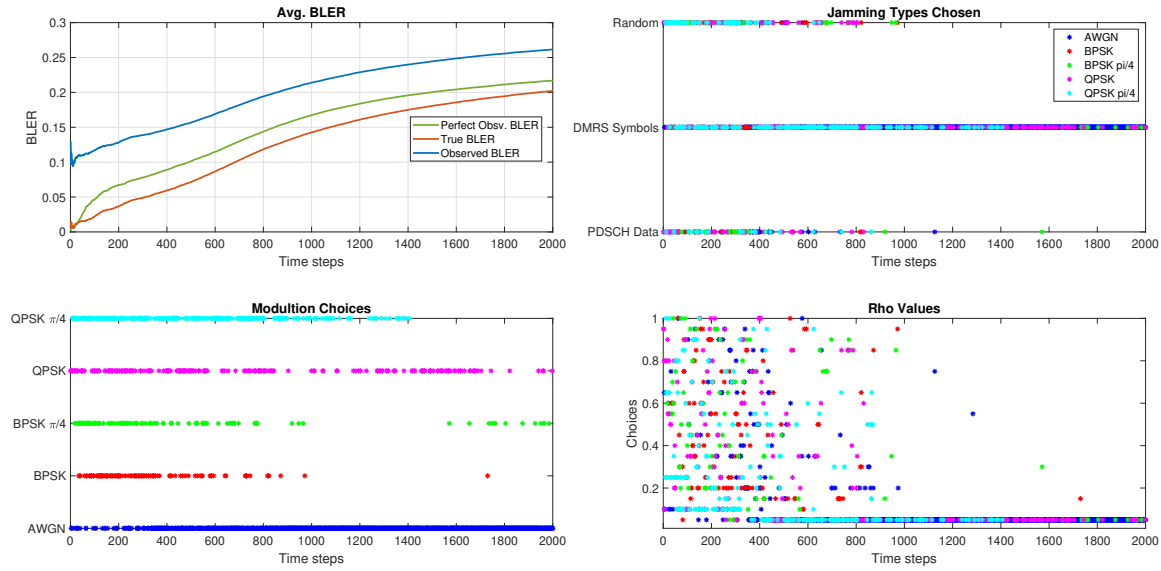

Figure 6.32: Collective results under HARQ processing and unreliable feedback ($\lambda = 0.1$) of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 7.2 dB and SNR = 24 dB.

than the previously discussed results in Figs. 6.31 and 6.32. The modulation and ρ choices still show that side exploring is occurring, and side exploring is starting to occur with the jamming method. Random jamming and PDSCH data jamming are being chosen on the side for the jammer to try different methods of obtaining a higher BLER. The main method of DMRS jamming is still being chosen, but side exploring is occurring for the bandit to discover if there are better options available to it. This may be because false detections of NACKs has occurred with these actions resulting in the bandit receiving a false sense of their possible effectiveness. Even with the randomness, the bandit is still able to prevail by achieving a BLER of 10% in 8 seconds.

We now simulate a JNR of 6.2 dB at $\lambda = 0.05, 0.1$, and 0.15. Fig. 6.34 shows the result of JNR = 6.2 dB and $\lambda = 0.05$. The margin of error between the observed and the true BLER has increased drastically. This BLER is ineffective against the victim since it achieves practically no block errors. Even with perfect observations, the bandit is unable to achieve

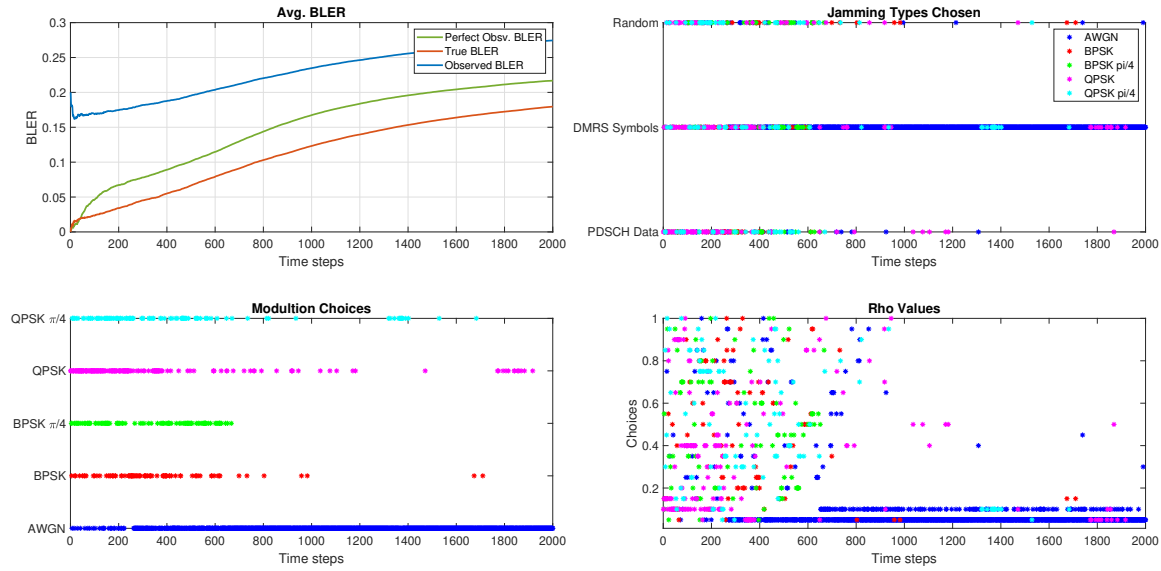

Figure 6.33: Collective results under HARQ processing and unreliable feedback ($\lambda = 0.15$) of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 4 dB and SNR = 24 dB.

a high enough BLER to affect the victim signal. The bandit is unable to learn because the BLER it observes is very high compared to the true BLER. It is unable to see successes because of the false feedback. Even if it were to see successes, since the perfect observation BLER is low it would be hard for it to converge normally under perfect conditions. This can be seen in the actions it takes. In both modulation scheme and jamming method, the choices almost appear random, and it cannot settle on one scheme to use. For the ρ values, it appears to land on consistent values such as mainly using AWGN at a low constant jamming rate and QPSK $\pi/4$ at a higher rate.

A value $\lambda = 0.1$ was then exmained, as shown in Fig. 6.35. The increase in uncertainty has widened the margin of error between the observed and true BLER curves again. This would increase the difficulty of learning an effective scheme to use if the jammer thinks that all schemes used give effective feedback. This goes back to Chapters 4 and 5 where at high levels of JNR relative to SNR, this particular context vector was seen picking every modulation

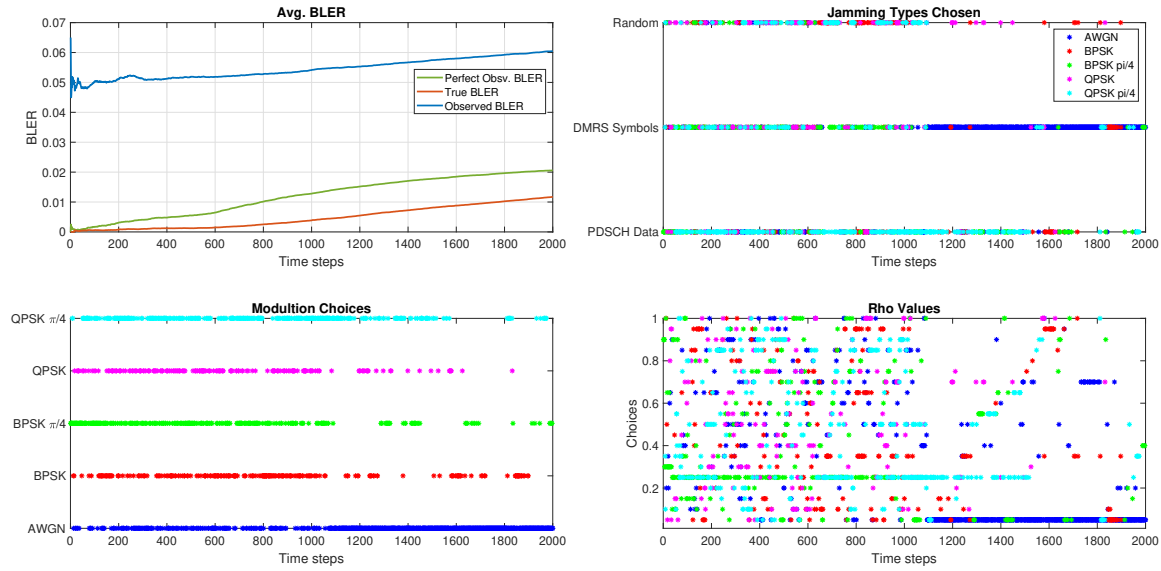

Figure 6.34: Collective results under HARQ processing and unreliable feedback ($\lambda = 0.05$) of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 6.2 dB and SNR = 24 dB.

scheme and value for ρ since all worked equally as well. This may be a similar case here where the imperfect reward feedback is creating a similar situation for the jammer. Since true NACKs are so infrequent, the rewards feedback is dominated by false NACKs and the jammer will see all combinations of jammer schemes as having roughly equal effectiveness. An approach to fix this may be to use the context vector suggested in Chapter 5, Eqn. 5.6. Remember, the results from Chapter 5 showed that Eqn. 5.6 would find one scheme that was effective and would employ it for a majority of the simulation and would occasionally conduct side exploring to see if other options were available. This could work in this particular situation. If it finds and selects a scheme that is effective to the bandit, it could employ that scheme to jam the victim while conducting occasional side explorations to see if it can find a more effective action. This could also work against the jammer if it settles on an undesirable scheme early and continues to stick with that scheme based on false early information. For this case, we have seen the context vector used in these simulations for this section have

suboptimal performance, so using a context vector that has a slight chance of working is better than causing no block errors if power constraints are a concern.

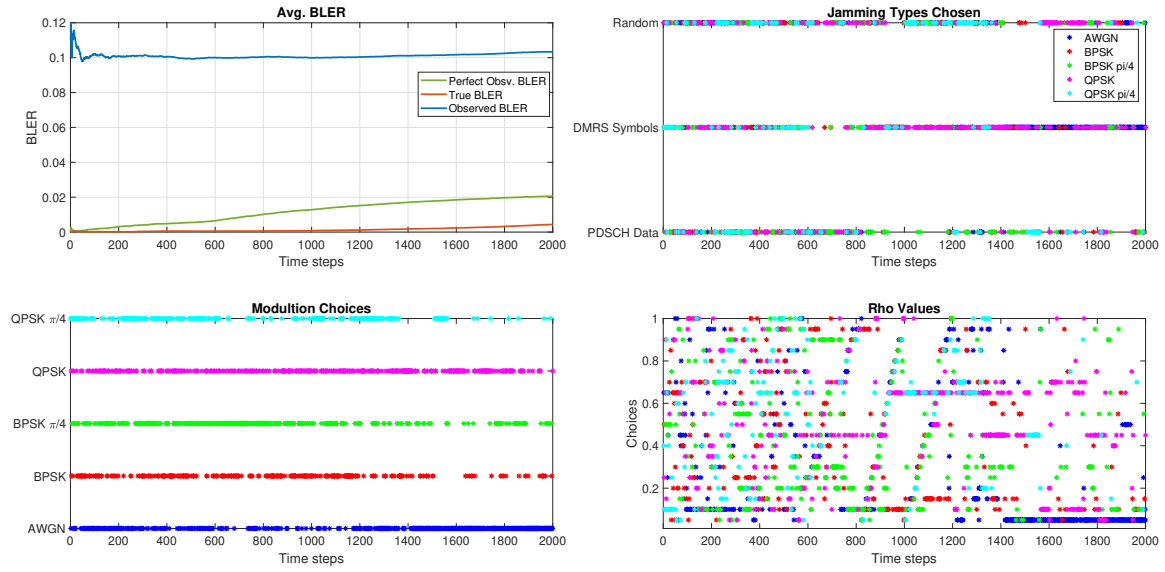

Figure 6.35: Collective results under HARQ processing and unreliable feedback ($\lambda = 0.1$) of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 6.2 dB and SNR = 24 dB.

Lastly, the result of simulating a value of $\lambda = 0.15$ is shown in Fig. 6.36. Again, the observed BLER has increased from Figs. 6.34 and 6.35. As seen previously, this makes the jammer unable to learn effective schemes since it thinks it is achieving high error rates and that all schemes are valid for jamming. The two options available are to either increase the JNR because the jamming was very ineffective or to switch the context vector if there are constraints on power usage for the jammer.

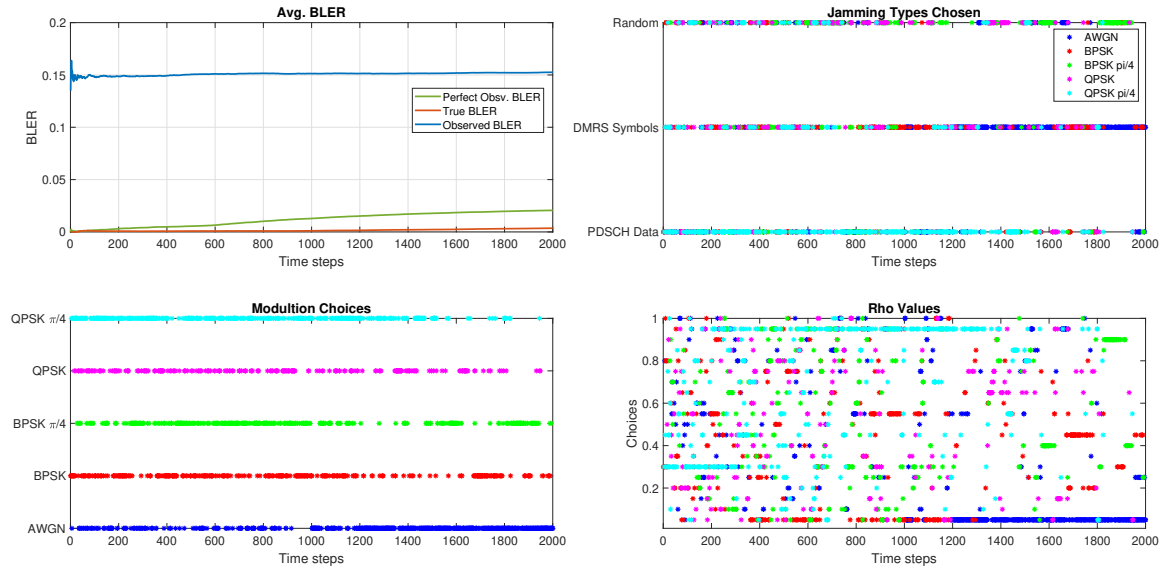

Figure 6.36: Collective results under HARQ processing and unreliable feedback ($\lambda = 0.15$) of BLER, types of jamming methods used, jamming modulation schemes used, and ρ values chosen at JNR = 6.2 dB and SNR = 24 dB.

6.7 Conclusion

Jamming a coded system introduces new complexities in terms of jamming method/modulation, pulse scheme, and pulse rate. The jammer must try and use these methods to distribute the LLRs in a way that is irrecoverable for the error correction code. From simulations, we have demonstrated that targeting whole codewords is the best way to do this. Depending on the SJR however, how the power is distributed in the codeword is important. Too much power and only hitting small parts of the codeword allows for the codeword to recover. Too little power while hitting all of the codeword does not spread the LLRs. The jammer must hit enough of the codeword with enough power to spread a majority of its LLRs so the code cannot recover.

When perfect observation of reward feedback is used, the jammer is able to efficiently and effectively introduce high BLERs to the victim receiver in a short amount of time. In order

to jam a 5G-based system, we found DMRS jamming to be most effective so the victim receiver's noise and channel estimates over-correct the received symbols. Depending on the SJR, different modulation schemes and ρ values are beneficial for the jammer in order to cause high BLERs. Even with HARQ processing being used to increase the victim's reliability, the jammer is still able to affect the victim, but at a reduced rate. When unreliable reward feedback is introduced, the jammers performance and ability to learn degrades significantly depending on the JNR. With high reliability (i.e. 95% correct ACK/NACK observation), the jammer is still able to learn to a high enough BLER to significantly effect the victim. When reliability is significantly reduced, the jammer's learning capabilities are reduced depending on the JNR it is using. If the JNR is high enough, it can overcome the unreliability. If the JNR is low, the jammer may need to increase power or switch context vectors if power usage is a constraint to effectively learn to jam the victim.

Chapter 7

Conclusion

For jamming avoidance, through our comprehensive analysis we show that there is obvious improvement with usage of the learning algorithms. The need for high SINR and successful transmission rate is especially important in UAC networks because of the limited resources available to the communications system, such as limited energy resources and time for the transmission to reach the receiver. Time should not be spent trying to re-transmit because of the waste of resources and energy for the communications system. The comparison of algorithms showed that scaled UCB and TS algorithms perform better than traditional UCB-1. TS has a lower success rate than scaled UCB, but it converges at a faster rate and in some cases achieves a higher overall SINR. This is especially important for reliable and energy efficient transmission in the underwater environment.

Through our comprehensive analysis of jamming a legitimate communications system using an OFDM-modulated signal, we initially show that linear TS outperforms traditional UCB-1 in terms of convergence and SER caused to the victim communications system in reasonable time horizons. Low discretization allows for the bandit to thoroughly exploit the system quickly and cause the most disruptions to the victim's communications. This upholds the result found in [10] in the context of OFDM-modulated signals when the jammer may select from TD and FD jamming signals. Observing the actions selected by linear TS, we discover an unexpected insight which is that we find BPSK with a $\pi/4$ phase rotation to be a special case of the optimal scheme for QAM victim signals.

Our analysis of the inner workings of the context vector provide that there may not be a context vector that works for all use cases of linear bandits in communications systems when the victim employs different transmission strategies and not a singular strategy. There may be better ways to construct the context vector that are tailored to the problem trying to be solved. This may involve constructing a context vector that exploits early knowledge of effective schemes rather than continuing to explore for optimal schemes for the sake of increasing convergence rate, decreasing the learning period, increasing the exploitation period, and in the case of jamming, increasing the overall errors caused to the victim system. The context feature vector of Eqn. 5.6 provided many of these benefits without necessarily learning the optimal modulation scheme and ρ value to use. The intuition behind this is that the sampled Beta distribution context feature provides a way to capture the average frequency of success of the paired modulation scheme and ρ values as a probability distribution function instead of a deterministic value. Depending on how the modulation scheme and ρ value chosen perform, the concentration of the distribution may have a small variance such that it performs well or inadequate. The large variance in the distribution provides that the modulation scheme has not been chosen enough. If the distribution is highly concentrated on a particular action (modulation scheme and ρ value pair), the jammer will continue to choose that scheme going forward, only conducting side exploration if returns are diminishing.

Lastly, our analysis of jamming OFDM-modulated signals using FEC coding showed how different jamming strategies can be employed to disrupt the communications and cause high BLERs. We showed a novel way of distributing enough power over the codewords to effectively jam them by causing the LLR distributions to be widely spread out. After understanding these results, similar principles were then used to apply this to jamming a 5G-based system employing a linear bandits algorithm initially studied in Chapter 4. With perfect observations of ACK/NACK feedback for the reward function, the system was easily

able to converge to high BLERs. The schemes and ρ values the jammer chooses depends on the value of the SJR, but the main jamming method to use is to jam the DMRS. By jamming the DMRS, the estimation of the noise and the channel become over-compensated. This over-compensation over-corrects the data sent, allowing for LLRs that are widely incorrect. When unreliable reward feedback is introduced, the jammer is severely inhibited from converging to actions with significant impact to the victim depending on the SJR. Depending on the SJR and the accuracy of the information, it may be better to raise the JNR to effectively jam the victim, or if there are power constraints, to switch to context vectors that converge to effective actions to jam the victim recognizing that these may not be optimal.

7.1 Future Work

For jamming avoidance in an UAC network, further investigation of real-time convergence of the algorithms is needed to understand the required computational and communications power required to support the exploitation stage of these algorithms, as well as the efficiency of the algorithms. Included in this would be to investigate if these algorithms would be feasible and applicable to real-time scenarios.

Investigation of a "game" using multi-armed bandits where the transmitter would have choices to switch frequency bands, move closer or farther from the intended receiver, or change transmission power is of interest. The jammer would also have options to move, change frequency bands, or change transmission power each time step. The goal would be to see how the transmitter would try to mitigate the jammer with these actions.

For jamming a legitimate communications system, in reality, communications systems will use multi-antenna systems to increase diversity and increase throughput of the system. This allows for techniques like beamforming and maximal ratio combining to be used to improve

the robustness of the signal. Multi-antenna systems also allow for more layers on the transmission channel, which translates to more codewords being able to be mapped to the signal. Testing the ability of a bandit employing reinforcement learning on this type of system and tracking performance would be the next step this work could go in.

Modern communications systems will also switch modulation schemes and coding rates if degradation in the system is observed. From our results, once a BLER of 10% is noticed by the victim signal the victim would have switched to a more robust modulation scheme and coding rate, such as QPSK with LDPC code rate of 0.54. A victim lowering its MCS could be considered a success from the jammer's perspective. However, the jammer will likely see a decrease in BLER as a result which will impact the ability to learn successful jamming strategies. Non-stationary tracking of the problem is important to discover performance of the jammer employing a reinforcement learning algorithm for implementation in real-life scenarios.

Bibliography

[1] S. Amuru and R. M. Buehrer, "Optimal jamming against digital modulation," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 2212–2224, Oct. 2015.

[2] Y. Lin, Y. Liu, F. Lin, L. Zou, P. Wu, W. Zeng, H. Chen, and C. Miao, "A survey on reinforcement learning for recommender systems," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2023.

[3] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *Journal of Intelligent & Robotic Systems*, vol. 86, pp. 153–173, Jan. 2017.

[4] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[5] L. Bastos, G. Capela, A. Koprulu, and G. Elzinga, "Potential of 5G technologies for military application," in *Proc. International Conference on Military Communication and Information Systems (ICMCIS)*, IEEE, May 2021.

[6] M. Domingo, "Securing underwater wireless communication networks," *IEEE Wireless Commun. Mag.*, vol. 18, pp. 22–28, Feb. 2011.

[7] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, pp. 34–43, Oct. 2007.

[8] D. E. Lucani, M. Stojanovic, and M. Medard, "On the relationship between transmission power and capacity of an underwater acoustic communication channel," in *OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean*, IEEE, Apr. 2008.

[9] S. Amuru, C. Tekin, M. van der Schaar, and R. M. Buehrer, "Jamming bandits—a novel learning method for optimal jamming," *J. IEEE Transactions on Wireless Communications*, vol. 15, pp. 2792–2808, Apr. 2016.

[10] C. E. Thornton and R. M. Buehrer, "Linear jamming bandits: Sample-efficient learning for non-coherent digital jamming," in *Proc. IEEE Military Communications Conference (MILCOM)*, IEEE, Nov. 2022.

[11] Y. Shi and Y. E. Sagduyu, "Jamming attacks on federated learning in wireless networks," 2022. arXiv:2201.05172.

[12] C. E. Thornton, R. M. Buehrer, and A. F. Martone, "Constrained contextual bandit learning for adaptive radar waveform selection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, pp. 1133–1148, Apr. 2022.

[13] C. E. Thornton, M. A. Kozy, R. M. Buehrer, A. F. Martone, and K. D. Sherbondy, "Deep reinforcement learning control for radar detection and tracking in congested spectral environments," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, pp. 1335–1349, Dec. 2020.

[14] R. Bajrachrya and H. Jung, "Contextual bandits approach for selecting the best channel in industry 4.0 network," in *2021 International Conference on Information Networking (ICOIN)*, IEEE, Jan. 2021.

[15] A. M. Jones and W. C. Headley, "Considerations of reinforcement learning within real-time wireless communication systems," in *IEEE MILCOM*, IEEE, Nov. 2022.

[16] Z. A. Khan, O. A. Karim, S. Abbas, N. Javaid, Y. B. Zikria, and U. Tariq, "Q-learning based energy-efficient and void avoidance routing protocol for underwater acoustic sensor networks," *Computer Networks*, vol. 197, p. 108309, Oct. 2021.

[17] Z. Schutz and D. J. Jakubisin, "Contextual bandits: Band of operation selection in underwater acoustic communications," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, May 2023.

[18] Z. Schutz, D. J. Jakubisin, C. E. Thornton, and R. M. Buehrer, "Linear jamming bandits: Leanring to jam OFDM-modulated signals," *IEEE International Conference on Communications (ICC)*, June 2024.

[19] C. E. Thornton and R. M. Buehrer, "On the value of online learning for radar waveform selection," *IEEE Transactions on Radar Systems*, vol. 1, pp. 505–519, 2023.

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, Nov. 2018.

[21] S. Agrawal and N. Goyal, "Further optimal regret bounds for thompson sampling," 2012. arXiv:1209.3353.

[22] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge University Press, July 2020.

[23] P. Qarabaqi and M. Stojanovic, "Statistical characterization and computationally efficient modeling of a class of underwater acoustic communication channels," *IEEE J. Ocean. Eng.*, vol. 38, pp. 701–717, Oct. 2013.

[24] D. Bouneffouf, S. Parthasarathy, H. Samulowitz, and M. Wistuba, "Optimal exploitation of clustering and history information in multi-armed bandit," in *Proceedings of the*

28th International Joint Conference on Artificial Intelligence (IJCAI), p. 2016–2022, AAAI Press, 2019.

[25] Z. Wu, T. Bicer, Z. Liu, V. De Andrade, Y. Zhu, and I. T. Foster, "Deep learning-based low-dose tomography reconstruction with hybrid-dose measurements," *arXiv preprint arXiv:2009.13589*, 2020.

[26] M. J. Bocus, A. Doufexi, and D. Agrafiotis, "Performance of OFDM-based massive MIMO OTFS systems for underwater acoustic communication," *IET Communications*, vol. 14, pp. 588–593, Mar. 2020.

[27] T. Qiu, Z. Zhao, T. Zhang, C. Chen, and C. L. P. Chen, "Underwater internet of things in smart ocean: System architecture and open issues," *IEEE Trans. Ind. Informat.*, vol. 16, pp. 4297–4307, July 2020.

[28] B. Li, J. Huang, S. Zhou, K. Ball, M. Stojanovic, L. Freitag, and P. Willett, "MIMO-OFDM for high-rate underwater acoustic communications," *IEEE J. Ocean. Eng.*, vol. 34, pp. 634–644, Oct. 2009.

[29] P. Zetterberg, F. Lindqvist, and B. Nilsson, "Underwater acoustic communication with multicarrier binary frequency shift keying," *IEEE J. Ocean. Eng.*, vol. 47, pp. 255–267, Jan. 2022.

[30] B. K. Patel, D. Roy, and S. R. K. Vadali, "Performance of BPSK and BFSK digital modulation schemes in colored noise scenario," in *2015 IEEE Underwater Technology (UT)*, IEEE, Feb. 2015.

[31] M. Stojanovic, "Low complexity OFDM detector for underwater acoustic channels," in *OCEANS 2006*, Sept. 2006.

[32] X. Feng, J. Wang, X. Kuai, M. Zhou, H. Sun, and J. Li, "Message passing-based impulsive noise mitigation and channel estimation for underwater acoustic OFDM communications," *IEEE Trans. Veh. Technol.*, vol. 71, pp. 611–625, Jan. 2022.

[33] X. Cai, L. Hu, W. Xu, and L. Wang, "Design of an OFDM-based differential cyclic-shifted DCSK system for underwater acoustic communications," in *26th IEEE Asia-Pacific Conference on Communications (APCC)*, Oct. 2021.

[34] D. J. Schott, A. Gabbrielli, W. Xiong, G. Fischer, F. Höflinger, J. Wendeberg, C. Schindelhauer, and S. J. Rupitsch, "Asynchronous chirp slope keying for underwater acoustic communication," *Sensors*, vol. 21, p. 3282, May 2021.

[35] S. Jiang, "On securing underwater acoustic networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 729–752, 2019.

[36] D. J. Jakubisin, Z. Schutz, and B. Davis, "Resilient underwater acoustic communications in the presence of interference and jamming," in *OCEANS 2022, Hampton Roads*, Oct. 2022.

[37] S. Hashima, K. Hatano, H. Kasban, and E. M. Mohamed, "Wi-Fi assisted contextual multi-armed bandit for neighbor discovery and selection in millimeter wave device to device communications," *Sensors*, vol. 21, p. 2835, Apr. 2021.

[38] D. Bouneffouf, I. Rish, and C. Aggarwal, "Survey on applications of multi-armed and contextual bandits," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, July 2020.

[39] D. Mihaylova, "An overview of methods of reducing the effect of jamming attacks at the physical layer of wireless networks," in *Lecture Notes of the Institute for Computer Sci-*

ences, Social Informatics and Telecommunications Engineering, pp. 271–284, Springer International Publishing, 2019.

[40] M. Abeille and A. Lazaric, "Linear thompson sampling revisited," *J. Electronic Journal of Statistics*, 2016. arXiv:1611.06534.

[41] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," 2012. arXiv:1209.3352.

[42] Y. Shi, Y. E. Sagduyu, T. Erpek, and M. C. Gursoy, "Jamming attacks on NextG radio access network slicing with reinforcement learning," in *Proc. IEEE Future Networks World Forum (FNWF)*, IEEE, Oct. 2022.

[43] M. Jacovic, X. R. Rey, G. Mainland, and K. R. Dandekar, "Mitigating RF jamming attacks at the physical layer with machine learning," *J. IET Communications*, vol. 17, pp. 12–28, Oct. 2022.

[44] M. Shi, A. Laufer, Y. Bar-Ness, and W. Su, "Fourth order cumulants in distinguishing single carrier from OFDM signals," in *Proc. IEEE Military Communications Conference (MILCOM)*, IEEE, Nov. 2008.

[45] M. Lichtman, R. M. Rao, V. Marojevic, J. H. Reed, and R. P. Jover, "5G NR jamming, spoofing, and sniffing: Threat assessment and mitigation," 2018. arXiv:1803.03845.