Real-time Online Learning for Pattern Reconfigurable Antenna State Selection

Xaime Rivas Rey, Geoffrey Mainland and Kapil Dandekar
Drexel University, Philadelphia, PA 19104, USA
Email: {xr39, gbm26, krd26}@drexel.edu

2020 7th NAFOSTED Conference on Information and Computer Science (NICS) | 978-0-7381-0553-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/NICS51282.2020.9335872

***Abstract*—Pattern reconfigurable antennas (PRAs) can dynamically change their radiation pattern and provide diversity and directional gain. These properties allow them to adapt to channel variations by steering directional beams toward desired transmissions and away from interference sources, thus enhancing the overall performance of a wireless communication system. To fully exploit the benefits of a PRA, the key challenge is being able to optimally select the antenna state in real time. Current literature on this topic, to the best of our knowledge, focuses on the design of algorithms to optimally select the best antenna mode with evaluation performed in simulation or post-processing. In this study, we have not only designed a real-time online antenna state selection framework for SISO wireless links but we have also implemented it in an experimental software defined radio testbed. We benchmarked the multi-armed bandit algorithm against other antenna state selection algorithms and show how it can improve system performance by mitigating the effects of interference taking advantage of the directionality PRAs provide. We also show that when the optimal state changes over time the bandit approach does not work very well. For such a scenario, we show how the Adaptive Pursuit algorithm works well and can be a great solution. We also discuss what changes could be done to the bandit algorithm to work better in this case.**

***Index Terms*—Machine Learning, Online Learning, Pattern Reconfigurable Antennas, Multi-Armed Bandit, Adaptive Pursuit, $\epsilon-$greedy, Wireless communications, Software Defined Radios**

I. Introduction

Over the last few decades, there has been an exponential growth in the number of connected wireless devices and the amount of data sent and received. This increasing demand is one of the main challenges in wireless communications, motivating the development of simple techniques to better utilize and share spectrum.

Traditionally, to mitigate adverse effects such as interference, multi-path fading, shadowing, spectrum scarcity, solutions like channel coding, power control, or MIMO communications were proposed. However, all these solutions come with the price of added complexity, overhead and/or cost. Recent studies have shown how Reconfigurable Antennas (RA) can be used to enhance wireless communications by altering the wireless channel that a radio perceives [1]. RA can dynamically change their radiation characteristics, including frequency, polarization and radiation pattern [2], [3]. This work focuses particularly on pattern reconfigurable antennas (PRA). This type of antenna provides pattern diversity gain and directional gain that can be leveraged to enhance the performance of a wireless communication system.

Multiple algorithms have been proposed but only tested via either simulation or in post-processing, not being able to run the algorithm in a real time experiment [4]–[6]. Some approaches were either based on channel estimation and prediction [6] or complex online learning where the search space needs to be pruned [4]. Specifically, Bahceci et al. [6] proposed a block MMSE based channel estimation scheme in which only a subset of patterns are trained and then the correlation among different states is used to predict the channels of the untrained states. The problem with this method is that it incurs a large delay and overhead due to the extensive channel estimation based training procedures. In [4], Zhao et al. proposed to use a Thompson Sampling (TS) framework as it converges faster when the number of states to choose from is large and also helps prune the search space. However, this technique is more complex than a simple Multi-armed bandit approach and only justifiable when the search space is large. Hasan et al. [7] used reconfigurable antennas based on parasitic tuning. The challenge with this approach is the large amount of possible states the RA can be set to, requiring an offline Genetic algorithm (GA) based search to determine which configurations generate optimal modes. Also, they had to assume the channel was quasi-static and extensive training is required, generating a lot of overhead and making this approach not very useful for real-time applications. The contribution of this paper is the demonstration of a low complexity selection algorithm for RA that operates in real-time.

The rest of this work is structured as follows: Section II covers the system model and pattern reconfigurable antenna used. Section III describes the selection algorithm used for this work and what other algorithms will be used for benchmarking. Section IV describes the experiments and evaluates the results obtained. Section V summarizes the paper and gives an overview of the results and future work that will follow up from this study.

II. System Model

This paper focuses on the downlink of a single cell single input single output (SISO) OFDM system where the transmitter is equipped with a conventional omnidirectional folded-dipole antenna and the receiver is equipped with a reconfigurable Alford loop antenna (RALA) [3]. Our work can easily be generalized to the case of a RALA on the transmitter side as well. The receiver node can be seen on Figure 2, where

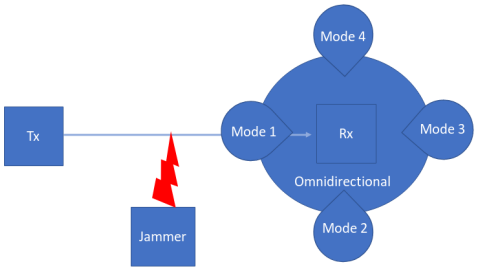

Fig. 1. **System model -** Unidirectional SISO link with a third node acting as a source of interference.

three main components are highlighted: reconfigurable Alford loop antenna, raspberry pi (RP) to physically switch the state of the antenna and a software defined radio (SDR).

The received signal y_t at time t is represented as follows:

$$\mathbf{y}_t = \mathbf{h}_{t,n}^T \mathbf{x}_{n,t} + \mathbf{n}_t \quad \text{for} \quad n = 1, ..., K \tag{1}$$

Where $\mathbf{h}_{t,n}^T \in \mathbb{C}$ represents the transposed channel response at time t for transmit antenna configuration n, $\mathbf{x}_t \in \mathbb{C}$ is the transmitted data at time t. The noise n_t is modeled as a zero-mean complex white Gaussian random variable.

The RALA has a total of $K = 5$ modes, 4 of which are directional and range 360^o in azimuth. The other mode is omnidirectional. The measured far field radiation patterns can be seen in Figure 3.

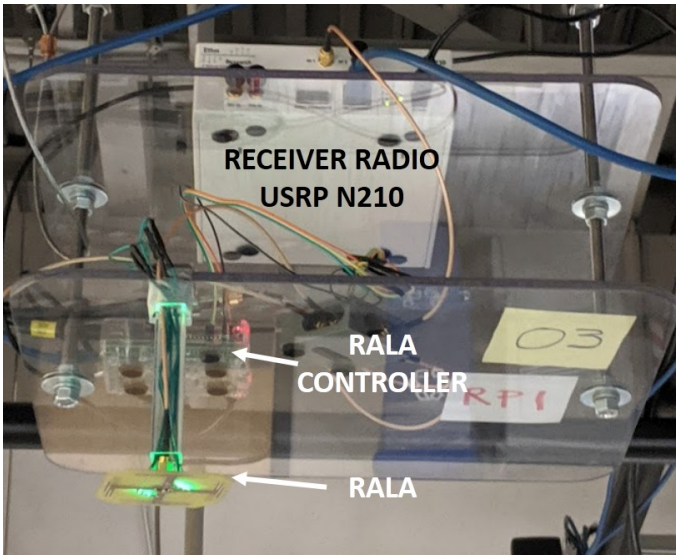

Fig. 2. **Receiver node -** A USRP N210 was used for both receiver and transmitter, the receiver can be seen in the figure with a Reconfigurable Alford Loop Antenna and its custom controller (Raspberry Pi).

III. Methodology

A. Multi-armed bandit theory

The Multi-Armed Bandit (MAB) problem is a fundamental mathematical framework for learning unknown variables. The classic formulation [9]–[11] states that there are N independent arms and a single player choosing arm i, where $i \in [1, 2, ...K]$. There is always a trade-off between exploiting

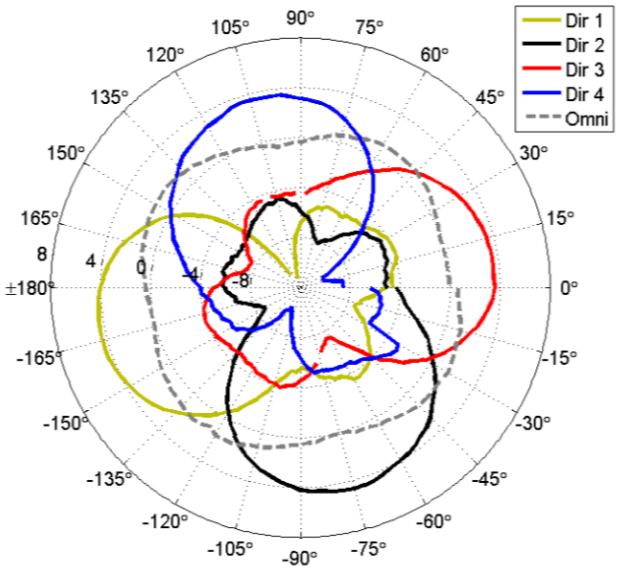

Fig. 3. **RALA radiation patterns -** Reconfigurable Alford Loop Antenna with 4 directional modes and an omnidirectional one [3], [8].

the arm with the highest expected payoff or exploring other arms to acquire more knowledge about their rewards. On each play of a single arm, the player receives a random reward and updates the expected payoff for that arm with the objective of maximizing the long term reward.

The formulation for this problem considers a transmitter with a conventional omnidirectional antenna and a receiver equipped with a RALA. In this context, "arm" and the "antenna state" refer to the selected radiation pattern for the PRA on the receiver side and are interchangeable. Every time period T, an arm is selected. If the transmission is successful, an instantaneous random reward $R_i(n)$ associated with that arm is achieved. The reward is assumed to be an i.i.d. random process with unknown mean that we are trying to learn. Since, at each time slot, only the selected arm generates a reward, this method does not require instantaneous information for each state, making it a practical real-time approach.

B. State selection strategies

We base our selection technique on two different policies given in [5], [12] and benchmarked them against other reference algorithms. A description of all selection strategies is given next.

1) Random selection: This strategy was used as a performance lower bound. Every time a state needs to be selected, one of the K antenna states is selected uniformly at random.

2) ϵ-greedy: When having to choose an arm, this policy will either select the arm with current highest mean reward with a probability of $1 - \epsilon$ or any of the K arms with probability ϵ.

3) Multi-armed bandit: The two algorithms used are described in detail in Algorithm 1 and Algorithm 2. For clarity,

a description along with a definition of all variables used by the algorithms is given below:

- UCB1 - this algorithm needs to keep track of two vectors containing mean rewards ($\overline{R}_i$) and how many times (n_i) any arm i has been visited, where $i \in [1, 2, ...K]$, being K the total number of possible states and n the time step being considered. The first step is to select each arm at least once and populate these variables. Once this initialization step is done, the algorithm enters an infinite loop where it selects the arm with higher upper confidence index, which in this case is a combination of the mean reward for the arm and the one sided confidence interval of the estimated mean reward: $\overline{R}_i + \sqrt{2ln(n)/n_i}$. This index allows arms with smaller mean reward to be selected after a certain amount of trials where the best arm has been exploited so we have a balance between exploitation and exploration.
- UCB1 Tuned policy - this algorithm keeps track of the same variables as the previous one, the main difference being that it accounts for the variance in reward, making a better state selection in the majority of cases. The tuned index is defined as: $\overline{R}_i + \sqrt{ln(n)/n_i min\{1/4, V_i(n_i)\}}$, where $V_i(x) = (1/x \sum R_{i,x}^2) - \overline{R}_{i,x}^2 + \sqrt{2ln(t)/x}$, where state i has been selected x times during the first t time slots. This algorithm is expected to perform better in scenarios where the rewards have high variance due to dynamic channel variations.

4) Adaptive Pursuit (AP): The main advantage of this algorithm is that it doesn't assume the problem to be time invariant, hence its ability to adapt to dynamic scenarios. To do so, instead of keeping track of a mean reward per state, adaptive pursuit will store a vector of expected rewards $\overline{\mathbf{Q}}$ and a vector with the probability of choosing each state $\mathbf{P}$. Each element of the probability vector is bounded by P_{min} and $P_{max} = 1 - (K-1)P_{min}$ and must add to $\sum_{i=1}^{K} P_i = 1$. Each one of these vectors will be updated with a learning rate, β, $\alpha \in (0, 1]$ respectively, using a low pass filter after every iteration. A more detailed explanation can be seen on Algorithm 3.

Algorithm 1: UCB1 Policy, Auer et al. [12]

```
initialization;
n_i, R̄_i ← 0;
Select each antenna state at least once and update n_i, R̄_i;
while True do
    Select antenna state i that maximizes upper confidence index R̄_i + sqrt(2ln(n)/n_i);
    Update n_i, R̄_i for antenna state i;
end
```

IV. Performance Evaluation

In this section we describe the experiments ran to benchmark in real-time the two different UCB policies against the other algorithms previously described on Section III. The hardware, software and test procedure are analyzed along with a discussion of results.

Algorithm 2: UCB1 Tuned Policy, Auer et al. [12]

```
initialization;
n_i, R̄_i ← 0;
Select each antenna state at least once and update n_i, R̄_i;
while True do
    Select antenna state i that maximizes upper confidence index R̄_i + sqrt(ln(n)/n_i · min{1/4, V_i(n_i)});
    Update n_i, R̄_i for antenna state i;
end
```

Algorithm 3: Adaptive Pursuit, based of Wolfe et al. [13]

```
initialization;
P̄ ← 1/K
Q̄ ← 0
while True do
    Select state with probability distribution P̄
    Update expected reward for selected state i*:
    Q̄_{i*} = (1 − α)Q̄_{i*} + αR_{i*}
    Update probability vector P:
    P_i = P_i + β(P_max − P_i), i = i*
    P_i = P_i + β(P_min − P_i), else
end
```

Each node uses an in-house full-stack SDR implementation called "Dragon Radio" [14] which leverages Liquid DSP for its OFDM Physical layer with Media Access Control (MAC) and options of Time Division Multiple Access (TDMA) and Frequency Division Duplexing along with a very flexible link layer [15]. This implementation is equipped with a high level python interface in which an antenna state selection controller was developed. The controller was programmed in a way such that, based on user defined policies, it will select the antenna state using any of the policies previously defined in Section III. The switch for the reconfigurable antenna was coded in C and it ran on a raspberry pi model 3 (RP) [16], controlled via socket commands. The general-purpose input/output (GPIO) pins of the RP selected between different arms of the PRA by connecting them to either ground or $3.3\ V$ (off and on respectively).

The experiment consisted of two Dragon Radio nodes: a transmitter and a receiver. Depending on the scenario, a third node acts as a source of interference. All nodes used were Ettus USRP N210 [17] software defined radios (SDR), equipped with a SBXv3 daughterboard that allows them to work at frequencies of up to $4.4\ GHz$.

The receiving node was equipped with a RALA and the transmitter was equipped with a commercial omni-directional folded-dipole antenna. Dragon Radio was configured to use a

TDMA protocol and its throughput performance was tested for 120 seconds and 5 identical trials that were averaged for each one of the algorithms. All algorithms chose antenna modes using Received Signal Strength Indicator (RSSI) as a reward, except for random selection which used none. The center frequency was $2.485\ GHz$, bandwidth was $4\ MHz$ and the modulation scheme was QPSK. Traffic was generated using iperf2 network test software, allowing us to record Packet Error Rate (PER). For every experiment the receiver logged RSSI.

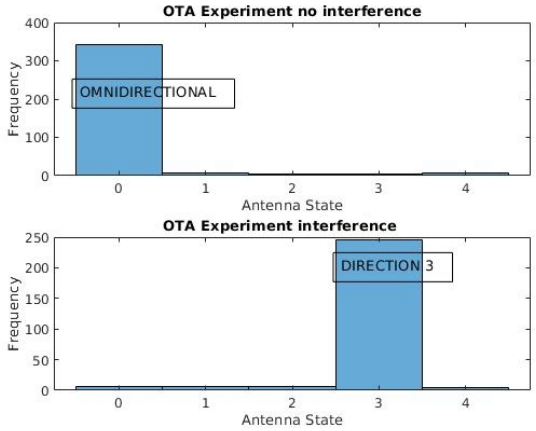

Fig. 4. **Histogram of state selection for UCB1 tuned.** This policy, after a initial exploration, exploits the omnidirectional mode 0 with no interference (top), and with static interference it selects directional mode 3 (bottom).

A. Over the air SISO experiment without interference

Figure 5 shows the over the air (OTA) frequency versus time waterfall plot captured by a third node while the two Dragon Radio nodes communicated under no interference. Since there was no interference, all antenna modes allowed the radios to have a successful communication. All algorithms chose the omnidirectional mode and had less than 1 % PER. The reason for this is that due to the placement of the nodes, none of the directional antenna modes pointed directly to the transmitter, therefore making the omnidirectional mode the most successful one. Figure 4 (top) shows how the UCB1 Tuned policy selected the nodes and mainly exploiting mode 0.

B. Over the air SISO experiment with a static source of interference

This experiment tests if any of the RALA patterns can spatially suppress the interference generated from a fixed location and nearby RF continuous wave (CW) jamming node while steering away from it and towards the transmitter. The experimental setup of transmitter and receiver was the same and a third node acting as a jammer was added. The physical placement can be seen on Figure 6. The jammer is equipped with the same conventional omnidirectional folded-dipole antenna as the transmitter and has an equivalent gain.

The performance results of this experiment, captured with the PER that each algorithm provided, are shown in Table I. In this case, both UCB and $\epsilon-$greedy policies selected a

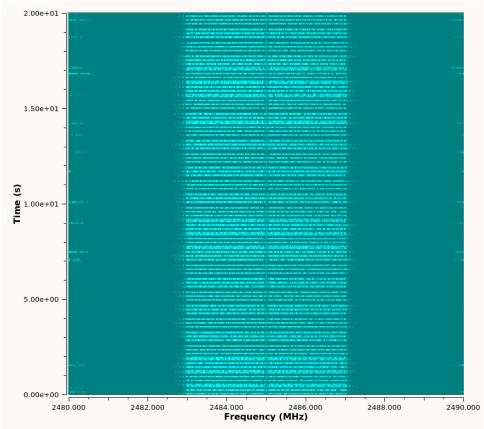

Fig. 5. **OTA experiment.** This waterfall plot shows the transmitted signal frequency vs time waterfall plot when the UCB1 Tuned policy was used in real time and no interference.

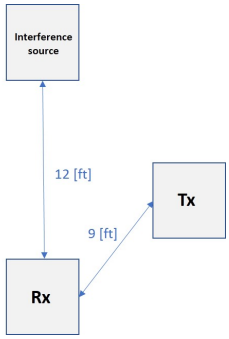

Fig. 6. **Physical Layout.** The three interference nodes shown represent the location at which the interfering node was located.

directional mode, particularly mode 3, as it steered away from the jammer and pointed towards the transmitter. This can be seen in Figure 4 (bottom) for the UCB1 Tuned policy. The differences in performance between algorithms comes from the fact that each policy exploits and explores the antenna arms differently, with UCB1 Tuned policy being the best one.

TABLE I
PER UNDER STATIC SOURCE OF INTERFERENCE

Algorithm	UCB1 Tuned	AP	UCB1	Random	$\epsilon-$greedy ($\epsilon = 0.2$)
PER [%]	1.20	2.82	2.97	8.88	3.76

Figure 7 shows the empirical cumulative distribution function (CDF) of RSSI for all algorithms. These results further confirm the differences seen in Table I, as the UCB policies have the least amount of errors and also have the highest RSSI values. Random selection acts as a lower bound and $\epsilon-$greedy is in between the random policy and MAB. Adaptive Pursuit performs similar to the MAB, which makes sense as in this case the interferer is always on the same location therefore AP has no advantage over MAB.

C. Over the air SISO experiment with a dynamic source of interference

This experiment tests how the algorithms perform when the source of interference changes locations halfway during the

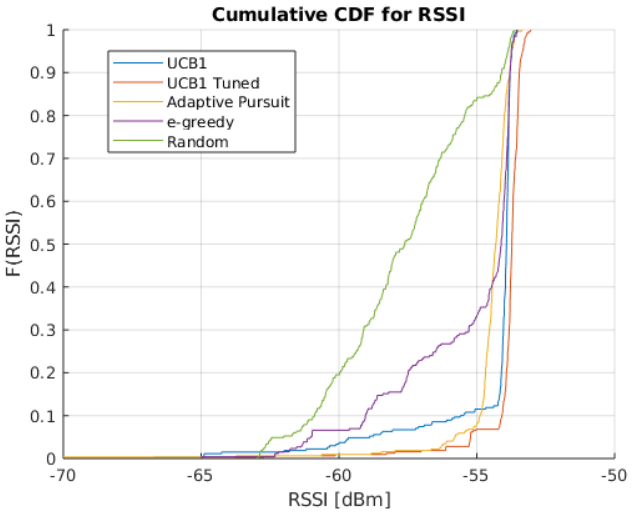

Fig. 7. **Empirical CDF -** this figure shows the empirical CDF of RSSI for all algorithms, further confirming the results shown on Table I.

experiment and therefore the optimal state for the PRA does as well. We can see from figure 8 how the MAB approach performs well once it converges to an optimal solution (4) but the moment the source of interference changes locations, it is unable to converge to a new optimal state, a total of 24.41% of packets were lost in the experiment. On the other hand, the Adaptive Pursuit starts on the optimal state (4) and once the source of interference changes locations, it converges over time to a new optimal state (2), allowing the radio to perform better overall and only lose a total of 4.94% of the packets sent.

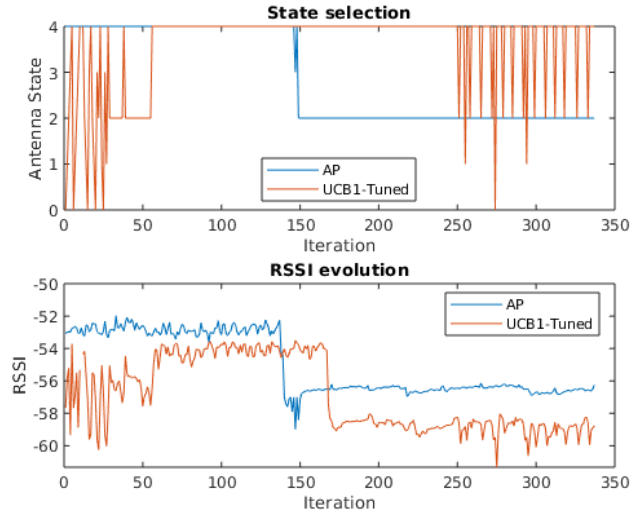

Fig. 8. **State and RSSI -** this figure shows how the states are selected and the effect this has on RSSI in the dynamic noise scenario for both AP and MAB algorithms. The interferer's location was changed halfway through the experiment, around iteration 150 of the algorithm.

The reason why AP is able to converge to a new optimal solution is that it selects the mode with the highest expected reward instead of the highest UCB, highly dependant on mean reward over time. In this case the MAB algorithm, after running for a while, in order for the mean reward of each state to decrease, it needs a large amount of time, whereas AP does not have that problem.

A possible way of avoiding this would be for the MAB approach to reset its parameters if the current rewards drop below a certain threshold or to use a moving average instead of an overall mean for the rewards.

V. Conclusion & Future Work

In this paper we demonstrated a software framework to test different algorithms for real-time pattern reconfigurable antenna state selection and showed the difference that it makes when there is interference. Particularly, we compared the cases where the source of interference is static and when it changes over time. We performed the experiments using USRP N210 SDRs with identical daughtercards and implemented our algorithms in python within the radio framework in order to run it in real-time.

First, we demonstrated that both the MAB and the AP algorithms can effectively be run in real-time to improve a SISO link by selecting the optimal PRA state while outperforming algorithms such as Random selection and $\epsilon-$greedy. Secondly, we showed how the bandit approached is not efficient when the sources of interference change locations over time, as it will exploit the first optimal state, even if it changes, once it has converged to a solution. On such a scenario we showed how the Adaptive Pursuit algorithm can be used and we also discussed how with some modifications, the MAB approach can be adjusted to achieve better results.

Now that the software framework is validated, we plan on expanding this work to multiple nodes and a higher number of antenna modes to select from by using a new model of the RALA.

Acknowledgment

The authors would like to acknowledge Alex Lackpour for his invaluable help, suggestions and feedback. This material is based upon work supported by the National Science Foundation under Grant No. CNS-1816387, CNS-1730140 and award CCF-1717088.

References

[1] N. Ojaroudi Parchin, H. Jahanbakhsh Basherlou, Y. I. A. Al-Yasir, R. A. Abd-Alhameed, A. M. Abdulkhaleq, and J. M. Noras, "Recent developments of reconfigurable antennas for current and future wireless communication systems," *Electronics*, vol. 8, no. 2, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/2/128

[2] M. K. Fries, M. Grani, and R. Vahldieck, "A reconfigurable slot antenna with switchable polarization," *IEEE Microwave and Wireless Components Letters*, vol. 13, no. 11, pp. 490–492, Nov 2003.

[3] D. Patron and K. R. Dandekar, "Planar reconfigurable antenna with integrated switching control circuitry," in *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*, April 2014, pp. 2737–2740.

[4] T. Zhao, M. Li, and G. Ditzler, "Online reconfigurable antenna state selection based on thompson sampling," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2019, pp. 888–893.

[5] N. Gulati, D. Gonzalez, and K. R. Dandekar, "Learning algorithm for reconfigurable antenna state selection," in *2012 IEEE Radio and Wireless Symposium*, Jan 2012, pp. 31–34.

[6] I. Bahceci, M. Hasan, T. M. Duman, and B. A. Cetiner, "Efficient channel estimation for reconfigurable mimo antennas: Training techniques and performance analysis," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 565–580, Jan 2017.
[7] M. Hasan, I. Bahceci, and B. A. Cetiner, "Downlink multi-user MIMO transmission for radiation pattern reconfigurable antenna systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6448–6463, 2018.
[8] S. Begashaw, J. Chacko, N. Gulati, D. H. Nguyen, N. Kandasamy, and K. R. Dandekar, "Experimental evaluation of a reconfigurable antenna system for blind interference alignment," in *2016 IEEE 17th Annual Wireless and Microwave Technology Conference (WAMICON)*, April 2016, pp. 1–6.
[9] V. Anantharam, P. Varaiya, and J. Walrand, "Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: I.i.d. rewards," *IEEE Transactions on Automatic Control*, vol. 32, no. 11, pp. 968–976, November 1987.
[10] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, may 2002. [Online]. Available: https://doi.org/10.1023/A:1013689704352
[11] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0196885885900028
[12] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, may 2002. [Online]. Available: https://doi.org/10.1023/A:1013689704352
[13] S. Wolfe, S. Begashaw, Y. Liu, and K. R. Dandekar, "Adaptive link optimization for 802.11 uav uplink using a reconfigurable antenna," in *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, Oct 2018, pp. 1–6.
[14] K. R. Dandekar, S. Begashaw, M. Jacovic, A. Lackpour, I. Rasheed, X. R. Rey, C. Sahin, S. Shaher, and G. Mainland, "Grid software defined radio network testbed for hybrid measurement and emulation," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2019, pp. 1–9.
[15] J. D. Gaeddert, "Liquid DSP - Software-Defined Radio Digital Signal Processing Library," http://liquidsdr.org/.
[16] "Raspberry Pi," https://www.raspberrypi.org/.
[17] National Instruments, "Ettus Research." https://www.ettus.com/.