Seeing the Future:
Optimizing mWidar Configurations with MDPs

Mark Boyer
University of Colorado, Boulder
Boulder, CO, USA
mark.boyer@colorado.edu

Anthony J. La Barca
University of Colorado, Boulder
Boulder, CO, USA
aj.labarca@colorado.edu

***Abstract*—The mWidar system is a proposed microwave imaging system whose geometry can be re-configured to interrogate different search volumes. This research investigated using a Markov Decision Process (MDP) to simulate searching a constrained 40×120 grid for targets. Deep Reinforcement Learning (DRL) was used to train a policy for configuration management to maximize rewards, considering (1) the number of targets in the search volume (Part 1), (2) the distance to each target, and (3) concurrent observations of targets (Part 2). Results show that a learned policy significantly outperforms a random or heuristic configuration policy given the positions and distance to targets. When considering repeated measurements, the increase in performance is doubled when using a DQN compared to a simple neural network. This work demonstrates the potential of using DRL for sensor configuration management, and provides a foundation for future work to improve the realism of the problem formulation.**

I. Introduction

A. mWidar System

Microwave arrays imaging has shown promise as a technique to quickly capture high resolution images. In simulation and controlled laboratory environments, this concept was able to perform real-time imaging and tracking with a range of tens of meters and spatial resolution of 0.1m [1]. Technology like this could be used for real-world applications like aircraft detection or spaceflight debris tracking. We hypothesized that a system that was operationalized for spaceflight orbital debris could be used to detect objects within a certain 2D area by physically changing the configuration of the transmitters to provide a low-power solution for real-time debris tracking and avoidance. Our challenge was to develop a policy for how the array should be configured in order to keep the spacecraft safe by looking out in front and also around the spacecraft within the 2D orbital plane of motion.

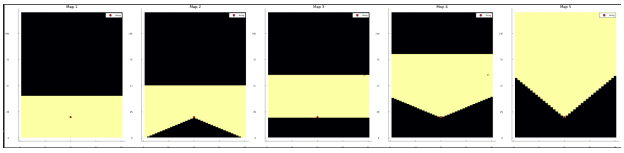

Fig. 1. mWidar Search Volumes by Configuration

B. Markov Decision Processes

We chose to formulate this as a Markov Decision Process (MDP), where the states are fully observable at each time step and the state is only dependent upon the previous time step state and action and a transition function between the two. When combined with a reward function and discount factor, this defines our general problem $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$.

C. Previous Work

Several others have used Deep Reinforcement Learning (DRL) on sensor optimization problems. Ewers et al. used DRL for sensor management on tracking and searching in a complex 3D environment with some success. You et al. used DRL for controlling a simulated unmanned combat air vehicle to detect targets with improved performance compared to a benchmark. Thornton et al. used DRL to vary the frequency and bandwidth of a simulated radar to improve target detection in congested spectral environments.

II. Simple MDP Formulation

A. Problem Setup

The problem we are attempting to solve is fundamentally about how to balance detection coverage for a system that can change configuration, similar to how an aircraft radar could change search patterns to search over a large volume of sky. We assumed that mWidar could be configured in 5 discrete possible configurations (1–5).

Configuration 1 allows it to search 360 degrees with a range of 20, while configuration 5 allows it to search to a range of 100 but with a max width of 40. See Figure 1 for visual depiction of the assumed search volumes. We assumed that our system could change configuration +/- 1 at each time step. Based upon the limitations of the search volumes, we constrained our state space to a grid that was 40×120, with the mWidar located at space (20,20). We generated tracks randomly around the edges that would travel linearly through the state space.

B. MDP Formulation

To formulate this problem as an MDP, we began with a simplified formulation of $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$. The state space $\mathcal{S}$ was defined for each grid square (-1=target without coverage, 0=no target, 1=target with coverage) plus a state for the mWidar configuration (1–5). The action space $\mathcal{A}$ was defined as $\mathcal{A} = [-1, 0, 1]$ for configurations $[1, 2, 3, 4, 5]$. The rewards $\mathcal{R}$ for each cell (i, j) were defined as:

$$\mathcal{R}_{ij} = \frac{k \cdot S_{ih}}{r_{ij}} \tag{1}$$

where k is a constant and r_{ij} is the range to the mWidar detector from the current cell.

The transition function was initially defined using the target and mWidar dynamics and a completely deterministic probability of detection (i.e. if there was coverage, it was a detect). Discount factor was set at $\gamma = 0.90$ initially.

C. Initial Solving

To solve the basic MDP, we trained a neural network using DQN [5]. The neural network contained an input layer of size 4803 (40×120 grid + radar x, y location + config), hidden layers of size 5000 and 128, and output layer of size 3 (action space). We used a Q target function of the maximum next state and Adam optimizer with a learning rate of 0.001. The neural network was trained using a replay buffer of maximum size 10000 and a batch size of 32. The training was done using an $\epsilon -$ greedy policy of 0.2. This was run for 500 time steps. A heuristic policy was also developed to keep the mWidar mostly in configuration 3, with semi-random explorations to configurations 1, 2, 4, and 5.

D. Evaluation

To evaluate our policy, we developed a simulation of the mWidar tracking objects within the designated area for a set amount of time t. The metrics we used to evaluate the system are based upon two criteria: number of tracks close to the system (Close Tracking = CT), and percentage of Tracked Targets (Tracked Targets = TT). These two metrics were measured as:

$$\text{CT} = \sum_t \sum_i \frac{\text{Detected targets}_i}{\text{All targets}_i * r_i}, \tag{2}$$

where r_i is the distance from the detector to the target, and

$$\text{TT} = \sum_t \sum_i \frac{\text{Detected targets}_i}{\text{All targets}_i}. \tag{3}$$

This simulation was then run n times and results were averaged for TT and CT. This was compared to a random policy as the baseline by using a random number generator with same seed.

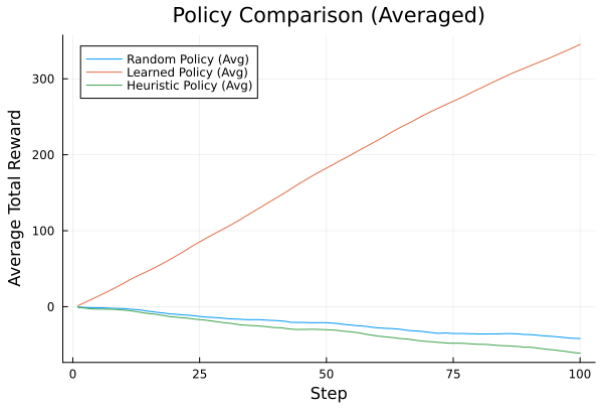

Fig. 2. Baseline Cumulative Rewards (CT) of DRL policy vs. random, heuristic policies

E. Performance

The performance of the neural network's learned policy was compared to a random policy and a heuristic

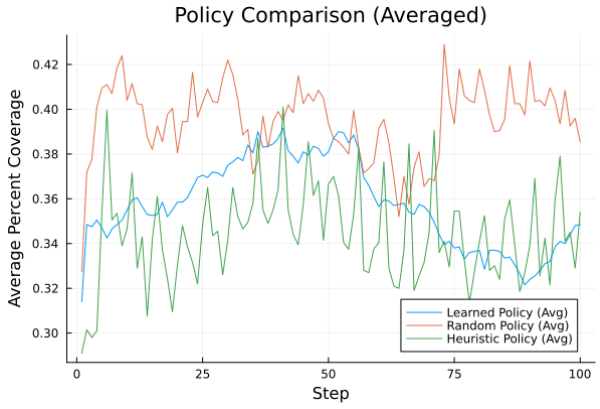

Fig. 3. Baseline Percent Coverage (TT) of DRL policy vs. random, heuristic policies

policy. Figure 2 shows the cumulative rewards (CT) of each policy averaged for 100 time steps of 100 runs each. The learned policy performed significantly better than the random and heuristic policies, which were close to zero over time. Figure 3 shows the percent coverage (TT) of each policy. There was no discernible difference between each policy over time, indicating the minimal gain in overall percentage of tracked targets for our learned policy. The model was also trained using a reward function that did not include the $1/r$ term and found no better performance on the TT metric.

III. Incentivizing Continuous Observation

A. MDP Modification

To encode additional time information, the MDP was modified by changing the values of tracked and untracked targets at each time step. Any target that went untracked for a time step had its state value decreased, while concurrent tracked targets had their state value increased. This was done to incentivize continuous observation of targets. This modification can be represented as an addition to our transition function, where the propagated state of each target at location (i, j) at time t is defined as:

$$\mathcal{S}_{ij}(t) = \begin{cases} \min(-1, \mathcal{S}_{ij}(t-1) - 6), & \text{if observed} \\ \max(2, \mathcal{S}_{ij}(t-1) + 2), & \text{if observed} \end{cases}$$

This encoding was combined with our overall reward function (Equation (1)), which inherently considers this time-dependent information. Since each targets score depends only on the previous time step, this does not break the Markov assumption.

The MDP was solved using two machine learning algorithms: a simple neural network to predict the Q-values, and the DQN algorithm described in Section II-C. Each algorithm was trained using the same hyperparameters as the previous section, but with the modified MDP.

B. Neural Network Performance

The neural network was trained for 1000 time steps and resulted in a net positive reward over time. The learned policy was able to outperform the random policy, leading to a $+40$ increase in reward. Figure 4 shows the performance of the learned policy compared to a random policy over time. When looking at the cumulative performance in Figure 5, the learned policy shows no significant improvement over time, just a net positive reward. Figures 4 and 5 show the performance of the learned policy

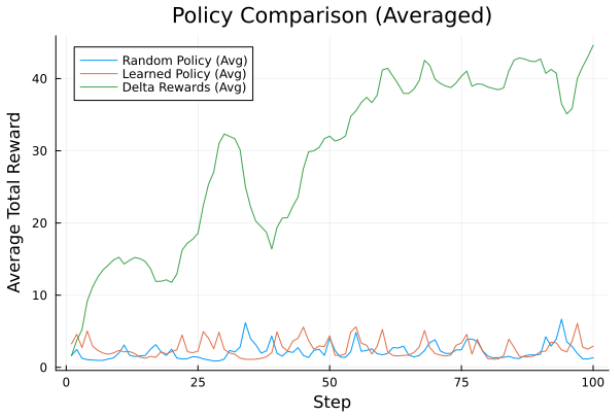

Fig. 4. Neural network-learned policy vs randomized policy

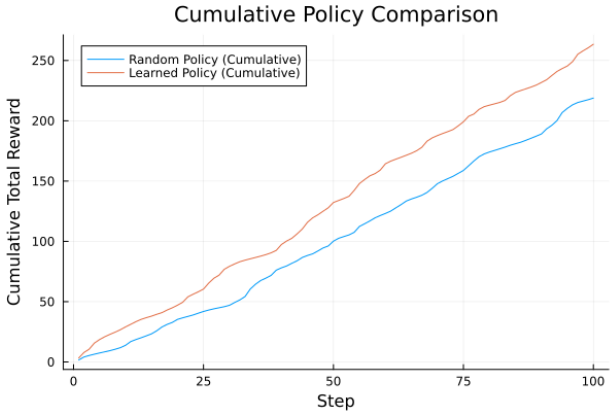

Fig. 5. Cumulative performance of neural network-learned policy vs randomized policy

C. DQN Performance

The DQN was trained for 1000 time steps and resulted in a net positive reward over time. However, as shown in Figure 6, the learned policy was able to outperform the random policy twice as much (+80). This is likely due to the increased complexity of the DQN algorithm, which allows for more complex function approximation. Figure 7 indicates that, while the random policy was able to achieve a net positive reward, the learned policy performed even better in the later stages of the simulation, indicated a true improvement in decision-making given the time-dependent information.

IV. Conclusion

This paper demonstrated a DRL approach to sensor configuration for a constrained tracking task. With a simple formulation and DQN implementation, the learned policy was able to significantly outperform a random or heuristic policy in tracking objects close to the radar,

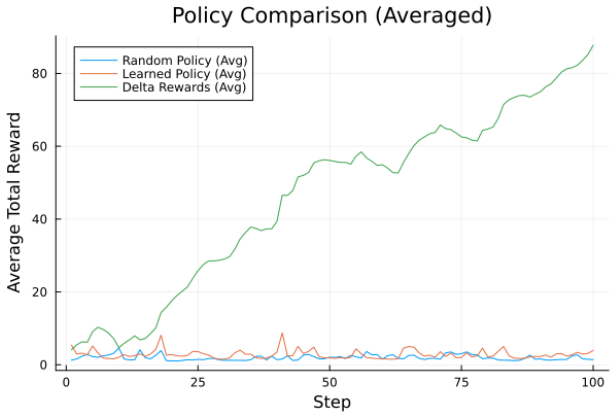

Fig. 6. DQN-learned policy vs randomized policy

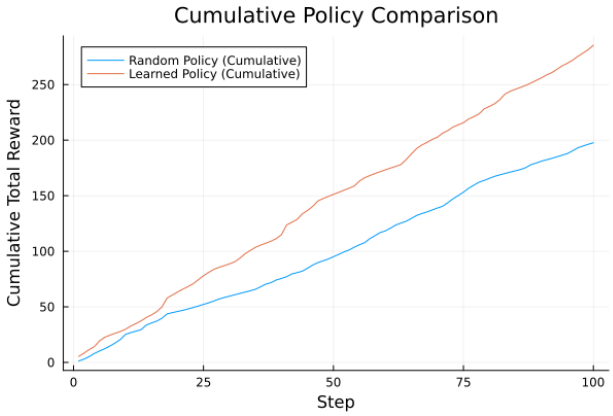

Fig. 7. Cumulative performance of DQN-learned policy vs randomized policy

but had no improvement in overall percentage of targets tracked. When additional time-dependent information was added into the state space, performance also increased significantly over a random policy. Unfortunately, we could not compare results between the two policies since the state formulation changed between the two MDPs, but both showed promising implementations of DRL for sensor configuration control.

A. Future Work

This project was completed as a limited investigation for a final project but may provide insights that may be applicable for engineers looking to improve sensor scheduling for advanced systems like electronically scanned arrays or scanning Infrared Search and Track systems. Future work could include adding additional complexity to the problem formulation in five ways: (1) incorporating real search volume dynamics and considering the true physics of the mWidar system for the observation masks, (2) reformulating the problem as a partially observable Markov Decision Process (POMDP), where the targets not in the detector are unobservable, (3) integrating a Kalman filter with observation uncertainty and adding filters to each of the targets to simulate true-tracking dynamics, basing the reward function on decreasing the uncertainty of the estimations of the targets, (4) using a continuous state space, where the mWidar detection are continuous, and (5) adding motion dynamics to the sensor location within a larger search volume. Each addition would increase the complexity and realism of the problem, bringing it closer to a real-world application of sensor configuration.

Acknowledgments

This work was supported by the University of Colorado Boulder, as part of the requirements for the course ASEN 5264: Decision Making Under Uncertainty. The authors would like to thank the course instructor, Dr. Zachary Sunberg, for his guidance and support throughout the project. The authors would also like to thank Dr. Fabio da Silva and Dr. Nisar Ahmed for their insight and support on the mWidar system.

V. Contributions and Release

Both authors contributed equally to the project formulation, development, and paper-writing. Mark developed the initial code for the POMDP formulation in Julia, implemented DQN in Part II (Simple MDP Formulation), and contributed to the abstract, background, related works, and conclusion. AJ structured the overall files, developed the initial problem idea, implemented all of the advanced MDP formulation in part III, and was the primary editor for the paper.

The authors grant permission for this report to be posted publicly.

References

[1] F. C. S. da Silva, A. B. Kos, G. E. Antonucci, J. B. Coder, C. W. Nelson, and A. Hati, "Continuous-capture microwave imaging," *Nature Communications*, vol. 12, no. 1, p. 3981, Jun. 2021.

[2] J.-H. Ewers, D. Cormack, J. Gibbs, and D. Anderson, "Multi-Target Radar Search and Track Using Sequence-Capable Deep Reinforcement Learning," *arXiv*, no. arXiv:2502.13584, Feb. 2025.

[3] S. You, M. Diao, and L. Gao, "Deep Reinforcement Learning for Target Searching in Cognitive Electronic Warfare," *IEEE Access*, vol. 7, pp. 37 432–37 447, 2019.

[4] 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 Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1335–1349, Dec. 2020.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.