COGNITIVE RADAR MODE CONTROL: A COMPARISON OF DIFFERENT REINFORCEMENT LEARNING ALGORITHMS

Stephanie A Ford [1]* , Matthew Ritchie [2]

[1]Radar Systems Team, Sensing Group, CIS, Dstl, Porton Down, UK
[2]Radar Group, EEE department, UCL, London, UK
**E-mail: sford1@dstl.gov.uk*

Keywords: COGNITIVE RADAR, LEARNING (ARTIFICIAL INTELLIGENCE), DEEP Q-NETWORK, ADVANTAGE ACTOR-CRITIC, PROXIMAL POLICY OPTIMISATION

Abstract

This paper describes the use of deep reinforcement learning (RL) to apply the concept of cognition in sensing systems to the choice of operational radio frequency (RF) mode (active, bistatic receive, electronic surveillance (ES), electronic protection measures (EPM)) for a multi-function RF system (MFRS). This is investigated in a simulated air-to-air combat scenario, with the RL on a blue fast jet rewarded for successfully guiding a missile to the opposition, a red fast jet, and penalised if the red jet is successful. Three RL algorithms (deep Q-network (DQN), advantage actor-critic (A2C), and proximal policy optimisation (PPO)) are compared with baselines that include the 4 static modes and a set of fixed rulesets, and it is shown that - after hyperparameter tuning - the algorithms perform comparably to these baselines. It is suggested that PPO might be the optimal algorithm in this context.

1 Introduction

The congestion of the electromagnetic (EM) spectrum has negative impacts on the performance of traditional radar, but also presents an opportunity: that the RF already available in the environment could be used to detect and track targets without having to transmit oneself. A radar acting as the receive node in such a set up would thereby be more covert and more "green" with respect to the EM environment. One of the concerns about this type of radar is that it is reliant on there being good signals of opportunity available; it is best used in conjunction with active radar.

This work suggests that there are multiple RF operational modes (eg active, bistatic receive, ES, or EPM) whose advantages would be better leveraged as part of a set of capabilities within a single aperture, and applies a type of machine learning (ML) called RL to the choice of mode at any given time. It can be argued that this produces a system that can be defined as cognitive. An air-to-air simulation has been developed that outputs data to, and receives instructions from, a RL algorithm that controls the modes of a MFRS. This architecture has been formatted to match the Open AI Gym [1] environments, such that Stable Baselines [2] agents can be used to quickly test different RL algorithms.

The rest of this paper continues as follows. Section 2 offers a definition for cognition when applied to sensing systems, provides more detail on the modes that a MFRS might use, and introduces some RL concepts. Section 3 describes the architecture of the simulation and other code scripts that were used, the measurements of performance that were captured for use as baselines, and how the the RL algorithms were implemented. The results are presented and discussed in section 4. Section 5 suggests several steps to continue this work.

The key novel points of this work are: a detailed, quantitative analysis of the advantages and limitations offered by a MFRS and these modes in an air-to-air scenario; the application of RL to this situation; and the comparison of the performance of the RL algorithms to both the static modes and to fixed-rule solutions.

2 Context

2.1 Cognition in Sensing Systems

Haykin, in his seminal work on cognitive radar [3], takes generalised concepts of cognition from psychology and applies them to radar, defining a cognitive radar to require a perception-action cycle (PAC), memory, attention and intelligence. Other definitions [4–7] often include learning, and responsiveness to the environment, both on transmit and receive. The emphasis on transmit and receive is partially due to the original focus on waveform design in [3], and partially due to the inspiration drawn from the natural world, as bats and dolphins are hailed as "nature's masters of echolocation" [7]. As the field has developed, it has been recognised [8] that a clear definition of a cognitive radar *scale* is required. A classification system has been proposed, where degrees of cognition can be measured along 3 axes: planning, memory/learning, and decision.

The definition that will be used as a goal for this work encompasses many of the requirements presented above. It is acknowledged that it is a more stringent, binary, definition of cognition, that it is less subtle than that proposed in [8], and that it may not bring benefits at every level of the radar decision

chain. However, in this context it represents a (perhaps ambitious) target. The definition states that a cognitive system will learn from its experiences (ie maintains a memory), and uses that learning such that it will improve its performance when presented with the same situation again.

Although not a requirement for cognitive radar, most literature on the subject uses either sophisticated optimisation algorithms, or deep learning of some kind, for example RL, which has been used here. Section 2.3 gives an introduction to this topic.

2.2 Operational Modes

Airborne RF systems can have many roles to play. On the radar side, they may be required to survey a large area, detecting and tracking many targets at once, and passing the information to other "blue" parties, either cueing them to form their own tracks or to take some action regarding the targets. This role is often performed by an airborne warning and control system (AWACS). Or perhaps the radar will be cued to form and maintain a weapons-quality track whilst guiding a missile towards the target. ES systems, on the other hand, must analyse a wide frequency band for transmitted signals, and process this data into useful information on the EM environment and the transmitters found within it. The fulfillment of any of these tasks may be required in a congested and contested EM environment, with the spectrum taken up by both civilian, friendly, and unfriendly military signals. One of the key pressures towards more bistatic operation is the scarcity of free frequencies in the spectrum. Similarly, the steadily more contested EM environment has necessitated research on EPM.

The limited space and power on-board an aircraft, as well as the increasing flexibility of RF antennas, suggests that combining these roles into a single aperture might prove advantageous. Such an MFRS could then have, for instance, the following modes: active (traditional radar), bistatic (receive only), ES, EPM.

The new flexibility and potential advantages offered by the incorporation of new modes into a single aperture come at the price of additional decisions. When is each mode advantageous? How often should the MFRS change mode? Where is the tipping point at which remaining covert is no longer worth the reduction in available information? Because there is, as of yet, no system that operates in such a flexible way, there is no received wisdom or established protocol to aid operators in making these assessments. This work investigates whether artificial intelligence (AI), specifically RL, can be used to either produce this protocol, or be used in real-time to make those decisions.

2.3 Reinforcement Learning

RL is a type of ML in which the "correct" answer is not known, but an agent is able to interact with an environment. To do so, the agent observes the current state, $s_t \in \mathcal{S}$ of the environment, and chooses an action, $a_t \in \mathcal{A}$. The environment propagates the effects of that action, as well as any other dynamics that may affect the state, and returns a new observation and a reward, $r \in \mathcal{R}$ for the agent. This reward is an indication of the performance of the agent, and may be either calculated from values in the state, for example, the track accuracy, or based on events that occur within the environment, for example, a successful track update.

The agent's goal is to maximise its return, which is defined as the sum of discounted future rewards, $R_t = \Sigma_{k=0}^{K} \gamma^k r_{t+k}$, where K can be any value up to the length of the episode, and $\gamma \in [0, 1]$ is a discount factor to control the importance of immediate rewards against future rewards: if $\gamma = 1$, future rewards are worth as much as immediate rewards; if $\gamma = 0$, only immediate rewards are considered; typically, this value is > 0.9 but less than 1. The value of a state, $V(s)$ is the expected return; the expected return of a state-action pair is measured using the Q-value.

Some RL algorithms, such as DQN [9], use the Q-value to choose an action: the ϵ-greedy policy chooses the action with the highest Q-value with a probability of $(1 - \epsilon)$, and acts randomly (it explores) otherwise. Other algorithms, like A2C [10], instead maintain a policy, which is a probability distribution over the actions, and update this using a separate estimate of the advantage function, $A(s, a) = Q(s, a) - V(s)$, whereas policy gradient methods such as PPO [11] increase the probabilities of actions that lead to high returns directly.

RL can be considered a way of implementing cognition, although it is not the only possible way! A RL algorithm will maintain a replay buffer of experiences it has seen recently, and use these to update the parameters of its neural network (NN) (or NNs). These can be seen as short- and long-term memories, and their update mechanisms. The parameters, along with the current state of the environment, are used to select an action responsively; the basis on Q-values implicitly predicts the impact of those actions. And as the new experiences are used to train the NN, a RL algorithm will change its behaviour when presented with the same situation a second time. In this way, all of the requirements presented in the definition used here are met.

3 Method

A RL algorithm was trained to choose the operational mode of a MFRS within a simulation developed for this work. An air combat scenario was simulated, where a blue and a red fast jet were given track-based behaviours with the ultimate goal of guiding a missile to within 50m of the other jet. The initial positions are shown in figure 1. The AWACS shown on the right orbits in this location, providing RF illumination for the blue MFRS's bistatic mode. Radar detections in this simulation are probabilistic, that is, a random number is compared with the required probability of detection (derived from the target's signal to noise ratio (SNR)) to determine if a detection attempt is successful or not. Although this high-level modelling is not as accurate as fully modelling the complex RF signals, it is much faster, and is sufficiently representative for this work.

The red fast jet was equipped with a traditional radar, an ES system, and a jammer, which could all operate simultaneously.

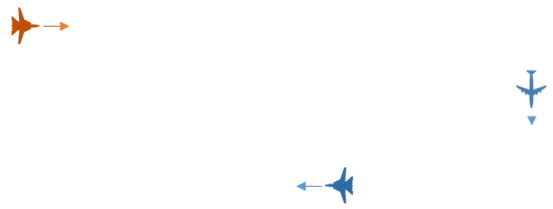


Fig. 1: Diagram of starting positions of the platforms simulated

The blue fast jet had only a MFRS with 4 available modes, as described above; the RL algorithm controlled these modes. A reward of +1 was assigned to the algorithm if the blue fast jet was successful; and a reward of -1 for when the red jet was successful. Note that this leaves open the possibility for neither side to win, and the reward to be zero (in fact, this was the case for 75% of episodes). This means that the rewards are particularly sparse, occurring only infrequently across many episodes; this will make learning more difficult. The parameters used in the simulation to represent the modes, and other key parameters such as the power of the red jet's jammer, were adjusted after an analysis of initial results showed that active mode was sufficiently superior to the others to prevent there being any benefit to switching mode. This adjustment aimed to more accurately represent the advantages of the alternative modes.

The simulation was linked to a PYTHON script, and wrapped in an Open AI Gym [1] class, such that Stable Baselines [2] algorithms could be readily applied. A `Pipe` object from PYTHON's multiprocessing toolbox was used to pass information and control between the simulation and the PYTHON script.

The action space for the algorithm contained the 4 modes of the MFRS; the state space contained track information such as range and velocity, track completeness; and contextual information such as whether the blue jet had a missile in the air, and if the illuminator for the bistatic mode (the AWACS radar) was operating.

Baseline performance measurements were captured for each of the 4 static modes (active, bistatic, ES, EPM); a random baseline; and 100 randomly-generated if-then rulesets. These were produced by selecting randomly from a specified set of possible values (based on the percentiles of values seen in the other baselines) for each variable in the state, thresholds, and comparators ("<", ">", "="). The results were measured over 20,000 episodes.

The Optuna package [12] was used to tune the hyperparameters of 3 algorithms: DQN [9], A2C [10], and PPO [11] from the Stable Baselines toolbox [2]. Tuning, although computationally expensive, was necessary as the out-of-the-box algorithms showed little or no learning.

The score for a particular set of hyperparameters was calculated as the sum of rewards, averaged over 3 trials of 100 episodes, after training for 1,480,000 (for the DQN, which otherwise did not show sufficient improvement to allow discrimination between the hyperparameter sets) or 740,000 (for A2C and PPO) timesteps*. The averaging was done in order to account somewhat for variations in (a) the initialisation of the algorithm and (b) the stochastic detections in the simulation.

4 Results

The baseline results are presented in table 1 - note that, as the jets are prevented from firing if they only have an ES track, the blue fast jet cannot win when its MFRS is in this mode.

Table 1 Baseline results.

Baseline	Average blue win rate	Average red win rate
Active	0.09	0.16
Bistatic	0.07	0.08
ES	0.00	0.31
EPM	0.11	0.10
Random	0.07	0.08
Average over rulesets	0.04	0.20
Highest blue win ruleset	0.11	0.10
Lowest red win ruleset	0.06	0.08
Highest red win ruleset	0.00	0.32

These baselines give an idea of the range of performances possible in this set-up: blue win rate can be as low as 0% (ie blue never wins) but only as high as 11%, whereas the red fast jet always seems to win at least 8% of playouts, or as many as 32%. The bias towards a red success in this particular set up is clear. The disadvantages of active mode (that it is a transmitting mode, and susceptible to jamming) are apparent in the high red win rate. The bistatic mode significantly reduces this red win rate and improves platform survivability, but does not offer a corresponding improvement in red win rate. The EPM mode does improve the blue win rate, but does not lower the red win rate as significantly as the bistatic mode.

Overall, there is a gentle, negative, linear relationship between blue and red win rates (higher blue win rate correlates to lower red win rate), as shown in figure 2; the triangles indicate static modes and crosses the rulesets, with the dashed line showing the line of best fit. This figure also helps to visualise good performance: the further down and right a result is, the better the performance; as the RL algorithms train, they might be expected to slide down the line of best fit.

As EPM mode is the only baseline to have a blue win rate higher than the red win rate, it can be described as the best baseline, although in some situations a decreased likelihood of a red missile being successful might be prioritised over high likelihood of a blue missile being successful.

The results from exemplar training runs of the tuned RL algorithms are shown in table 2. The initial and final win rates

**As each simulation playout ran for 300 seconds, and the mode selector was asked for a decision every 4 seconds, each simulation contains approximately 74 timesteps. 1,480,000 timesteps is therefore equivalent to approximately 20,000 episodes; 740,000 approximately 10,000 episodes.*

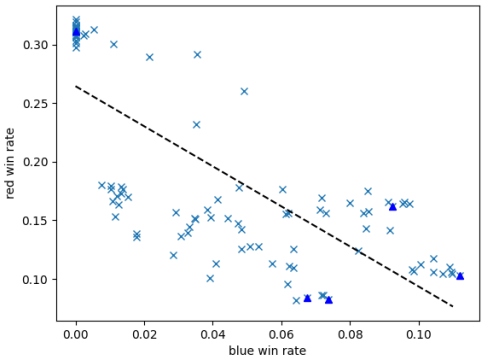


Fig. 2: Blue vs red win rates for the baseline results

Table 2 RL results.

Algorithm	Initial blue win rate	Final blue win rate	Initial red win rate	Final red win rate
A2C	0.02	0.09	0.24	0.13
PPO	0.01	0.10	0.21	0.13
DQN	0.01	0.07	0.24	0.16

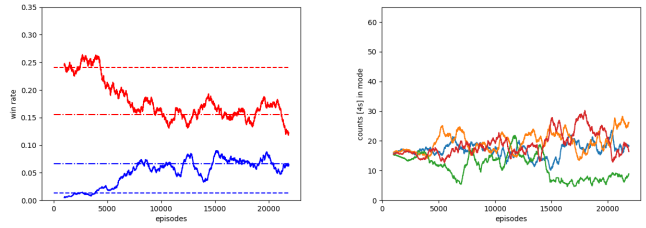
are calculated as the average across the first and last 5000 episodes respectively. The 1000-sample rolling averages of the blue and red win rates against the number of training episodes completed are plotted in figure 3. Also included are the 1000-sample rolling averages of the proportion of time spent in each mode, also against training episodes.

Although none of these tuned and trained algorithms surpass the EPM static baseline in overall performance, they do show evidence of learning, and the final performances fall in the bottom right hand side of the plot in figure 2.

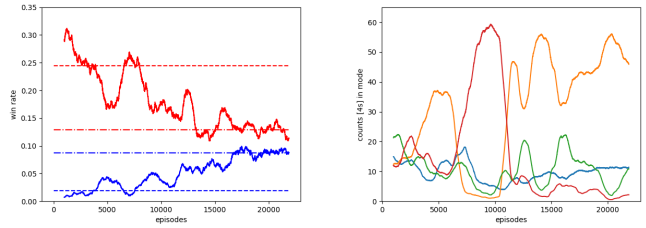
The left-hand plots in figure 3 show how performance is unstable during training, particularly for A2C, but seems to stabilise as the best performance is reached and the algorithm reduces the proportion of time spent exploring the state-action space. In addition to giving the best performance, the PPO algorithm also stabilises most quickly, followed by DQN - albeit at a worse performance - and then A2C. This pattern is consistent across all training runs for each algorithm.

The right-hand plots in figure 3 show how the relative proportions of modes selected vary with training. The DQN algorithm, the weakest of the 3, does not settle on any particular mode, although the tendency to select EPM or bistatic does seem to increase, whilst ES mode is selected less and less often. In contrast, both the A2C and PPO algorithms have a marked preference for bistatic and EPM modes respectively.

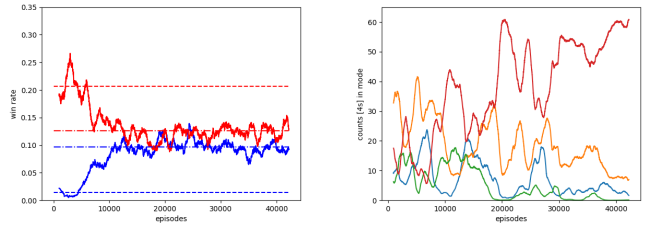
For the tuned PPO algorithm, the proportion of time still spent in bistatic mode does not seem to offer an advantage, unfortunately. However, the A2C algorithm, which only spends ∼65% of its time in its preferred mode, bistatic, significantly improves the blue win rate relative to this static baseline, from 7% to 10%, although it does not maintain the low red win rate of the baseline.



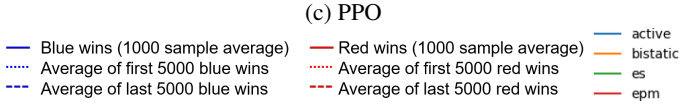
(a) DQN



(b) A2C



(c) PPO



(d) Legends

Fig. 3: Win rates and modes against training episodes for the tuned algorithms

On different training runs, the algorithms neither reach the same performance nor tend towards the same balance of modes. This is demonstrated forcefully by the example shown in figure 4, which shows a training run of the tuned PPO algorithm where it quickly converges to very high performance, relying mostly on bistatic mode, but then switches to using almost entirely ES mode. It may be that the algorithm found a local minimum, where there were fewer negative rewards for red successes, and possibly without sufficient memory of past episodes to encourage it to use other modes to garner positive rewards for blue successes.

5 Conclusion

This paper has discussed the use of deep RL to apply the concept of cognition in sensing systems to the choice of operational mode for a MFRS. It has been shown that RL algorithms, when tuned, can be trained to perform comparably to the baselines. The results suggest that DQN is the weakest of the 3 algorithms trialled, and suggest that the learning demonstrated by PPO converged more quickly and more stably than

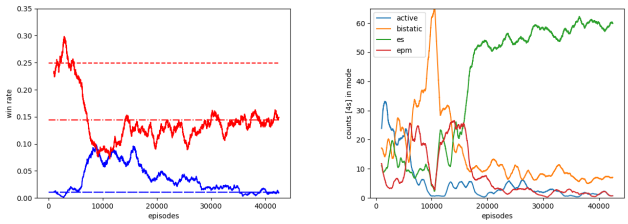


Fig. 4: Win rates and modes against episodes for tuned PPO

either DQN or A2C. However there is significant variation between training runs; more analysis using additional training runs is required to determine whether a medley of modes is advantageous over the best baseline (EPM) in this context.

Two variations on this setup are suggested for investigation: different reward systems, and different scenarios. Both of these could place emphasis on different elements of an airborne RF system's roles. For example, rewarding the cognitive agent controlling the MFRS for forming a track (an event-based reward) or proportionally to track accuracy (a metric-based reward) could encourage the agent to select either active or EPM mode, depending on the level of jamming present. In contrast, a reward system based on remaining covert by not transmitting ought to lead to greater use of bistatic and/or ES mode.

These emphases could also be achieved by changing the scenario within which the cognitive MFRS is operating. An illustrative comparison is the difference in behaviours that would be desirable in the case of (a) an aircraft (with a MFRS) defending a particular zone and (b) the same aircraft attempting to infiltrate an area defended by unfriendly platforms. Whereas in the first, an aggressive behaviour might be optimal (the use of active mode, for example, to alert the encroaching red aircraft that they have been spotted), in the second, this could be catastrophic. Instead, use of the covert modes would be desired. Even more interesting would be scenarios where it is not always obvious what the best mode would be, and where it changes over time. This could be achieved with multiple red targets in a more dynamic vignette.

Having investigated these variations, they could then be used to test the brittleness (or the contrary, robustness) of the learned solutions. How badly will a RL agent trained in for one situation, or with one reward scheme, perform in another? How long will it take to retrain, compared with training from scratch? These are vital questions that need answering before a cognitive system could be used in reality.

Acknowledgment

I would like to thank the Defence Science and Technology Laboratory (Dstl), for funding the work, and specifically my colleagues who have supported me - particularly Andy May, who has borne with my many questions with patience and humour.

References

[1] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016. see also `https://gym.openai.com`.

[2] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. see also `https://stable-baselines3.readthedocs.io/en/master/`.

[3] S. Haykin, "Cognitive radar: a way of the future," *IEEE signal processing magazine*, vol. 23, no. 1, pp. 30–40, 2006. IEEE.

[4] N. SET-227, "Cognitive radar," October 2020.

[5] A. Charlish, F. Hoffmann, C. Degen, and I. Schlangen, "The development from adaptive to cognitive radar resource management," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 6, pp. 8–19, 2020. IEEE.

[6] C. P. Horne, *An experimental investigation of cognitive radar*. PhD thesis, UCL (University College London), 2020.

[7] S. Z. Gurbuz, H. D. Griffiths, A. Charlish, M. Rangaswamy, M. S. Greco, and K. Bell, "An overview of cognitive radar: Past, present, and future," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 12, pp. 6–18, 2019. IEEE.

[8] C. Horne, M. Ritchie, and H. Griffiths, "Proposed ontology for cognitive radar systems," *IET Radar, Sonar & Navigation*, vol. 12, no. 12, pp. 1363–1370, 2018. IET.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[10] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, pp. 1928–1937, PMLR, 2016.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[12] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019. see also `https://optuna.org/`.