Sensors **2015**, *15*, 6668-6687; doi:10.3390/s150306668

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Beamforming and Power Control in Sensor Arrays Using Reinforcement Learning

Náthalee C. Almeida [1,*], Marcelo A.C. Fernandes [2,†] and Adrião D.D. Neto [2,†]

[1] UFERSA—Federal Rural University of the Semi-Árido, Pau dos Ferros 59900-000, Brazil

[2] DCA-CT-UFRN, Federal University of Rio Grande do Norte, Natal 59072-970, Brazil;
E-Mails: mfernandes@dca.ufrn.br (M.A.C.F.); adriao@dca.ufrn.br (A.D.D.N.)

[†] These authors contributed equally to this work.

[*] Author to whom correspondence should be addressed; E-Mail: nathalee.almeida@ufersa.edu.br;
Tel.: +55-84-9604-1927.

Academic Editor: Vittorio M.N. Passaro

Received: 31 October 2014 / Accepted: 4 March 2015 / Published: 19 March 2015

Abstract: The use of beamforming and power control, combined or separately, has advantages and disadvantages, depending on the application. The combined use of beamforming and power control has been shown to be highly effective in applications involving the suppression of interference signals from different sources. However, it is necessary to identify efficient methodologies for the combined operation of these two techniques. The most appropriate technique may be obtained by means of the implementation of an intelligent agent capable of making the best selection between beamforming and power control. The present paper proposes an algorithm using reinforcement learning (RL) to determine the optimal combination of beamforming and power control in sensor arrays. The RL algorithm used was Q-learning, employing an ε-greedy policy, and training was performed using the offline method. The simulations showed that RL was effective for implementation of a switching policy involving the different techniques, taking advantage of the positive characteristics of each technique in terms of signal reception.

Keywords: beamforming; power control; sensor arrays; Q-learning

1. Introduction

Sensor arrays have been widely used in a variety of applications including estimation of the direction of arrival (DOA) of signals [1,2], tracking systems, location of sources [3], and suppression of interference signals [4], amongst others. Adaptive array systems are able to locate and track signals (of users and interferences), dynamically adjusting the sensor alignment to maximize reception, and minimizing interference using signal-processing algorithms [5]. In applications where the aim is to suppress interference signals associated with various different sources, adaptive sensor arrays can be used together with power control techniques. The development of adaptive sensor arrays using beamforming together with power control can be used to achieve better system performance, with lower consumption of energy for transmission [6–17].

Methods used to resolve the problem of combining beamforming with power control have been described [6,7], where an algorithm has been proposed that is capable of beamforming in the uplink channel, followed by the adjustment of power in this channel and in the downlink channel. In this case, the weights for the downlink channel are considered the same as those for the uplink channel. This algorithm improves the performance of the system using a Signal to Interference plus Noise Ratio (SINR).

In [8], a duality constrained least-mean-square (DCLMS) algorithm was proposed that utilizes LMS to find the optimum beamforming weights, while at same time controlling the power in both the uplink and downlink channels. A reference signal is used for beamforming, avoiding the use of additional algorithms for computation of the arrival directions of the signals. Since it is based on LMS, the algorithm presents low computational complexity, and the adaptation step for control of convergence can be selected empirically in order to better address the objectives of application of the algorithm. Updating of the transmission power only occurs after convergence of the beamforming process, taking account of successive tests of convergence and loss of performance in non-stationary environments.

A similar algorithm is presented in [9], but with proportionality between the uplink and downlink weights. In other work [10], minimization of the transmission power in the channels is performed for each antenna individually. All these approaches assume a priori knowledge of the channel, so that it is possible to calculate the reception SINR, which then enables calculations for updating of the transmission powers in the uplink and downlink channels.

The work described in [11] presents a joint optimization of beamforming and power control in a coordinated multicell downlink system which attends multiple users per cell to maximize the minimum weighted signal-to-interference-plus-noise ratio. The optimal solution and distributed algorithm with a fast convergence rate are obtained using the nonlinear Perron-Frobenius theory and the multicell network duality. Despite operating in a distributed manner, the iterative algorithm requires instantaneous power update in a coordinated cluster by means of backhaul.

In [12], an algorithm is proposed that explores techniques of beamforming at the source and destination nodes, together with control of transmission power, in order to minimize the total transmission power of the source so that a minimum SINR threshold can be maintained in each receiver. This objective is achieved using an iterative algorithm that combines these techniques.

The work described in [13] proposes an algorithm for power control together with beamforming in the receiver, with multiple adaptive base stations, for communication in the uplink channel. An iterative optimization algorithm is proposed, and the results show that the transmission power can be significantly reduced, even with a smaller number of multiple base stations, which is of considerable interest for uplink channel communications.

In [14], an algorithm was developed to optimize the sum rate of the network under the interference constraints of primary users using beamforming and power control for each secondary user in a multiuser cognitive radio system. The interior-point method was used to solve the problem, employing a second-order cone programming (SOCP) approach.

The work described in [15] proposes an algorithm that utilizes the combination of beamforming and power control for a cognitive radio system with a base station with multiple antennas. This algorithm employs an iterative water-filling method that seeks to maximize the total rate of the secondary users, without affecting the quality of service (QoS) of the primary link, in other words, with the restriction of protecting the primary network from interferences in the cognitive radio system.

Another paper [16] considers the benefits of combining beamforming by means of multiple cells in a multiple input and multiple output (MIMO) system, where the multiple base stations can act together to optimize the corresponding beamforming in order to improve the overall performance of the system. The duality between the uplink and downlink channels is generalized for multi-cell cases using Lagrange’s theorem applied to the criterion: minimize the total transmission power subject to SINR for remote users.

The algorithm presented in [17] provides a combination of power control and beamforming in ad hoc wireless transmission networks with multiple antennas subject to constant QoS restriction. The proposed algorithm reduces the mutual interference in each node. The total transmission power of the network is minimized, while ensuring constant SINR in each receiver. Comparison was made of the performance of cooperative (COPMA) and non-cooperative (NPMG) iterative algorithms. In the case of the COPMA algorithm, users update their beamforming vectors in order to minimize the total transmission power in the network. In the case of NPMG, all the beamforming transmission vectors are updated at the start of the iteration, followed by power control.

Artificial intelligence (AI) techniques have been widely used in problems involving beamforming and power control [18–23]. The proposal of the present work is to develop an intelligent algorithm that utilizes reinforcement learning (RL) to establish an optimum policy for combination of the two techniques, beamforming (BF) and power control (PC), in sensor arrays, benefiting from the individual characteristics of each technique in accordance with SINR threshold. In this case, the great challenge of RL is to select the action (BF or PC) that, based on the learning state (SINR), is most suitable for the system. The RL algorithm used was Q-learning with an ε-greedy policy, trained using the offline method. In this case, acquirement of the most suitable techniques for indication of the action to select during each state of learning is obtained by reinforcement, employing a structure of adaptive parameters on which the algorithm operates.

It is important to emphasize that the technique proposed in this paper is not limited to use of the LMS (least mean square) procedure for fitting of the beamforming parameters. Other AI methodologies such as fuzzy logic and artificial neural networks, amongst others, can be used to improve the performance of the parameter fitting [24–26].

Reinforcement learning is based on the capacity of the agent to be trained to obtain knowledge while interacting with the unknown environment in which it is inserted. One of the great advantages of reinforcement training is precisely the capacity of the agent, while interacting with the unknown environment, to evolve by means of identification of the characteristics of this environment, dispensing of any need for the teacher present in supervised learning [27]. Thus, the main contributions of this work are:

- Reinforcement learning ensures that the optimal policy is found, with the most suitable technique being executed in preference to others.
- Independence in execution between beamforming and power control-these techniques are not performed sequentially, as in the above work, but alternately.
- Reduced complexity of the method, with fewer operations required, because one technique is not followed by another.
- The RL methodology proposed here is independent of the beamforming parameter fitting technique and the power control algorithm.

The paper is structured as follows: Section 2 provides a basic description of the functioning of the adaptive sensor arrays, together with a model of the input and output signals of the array, and a discussion of resolution of the beamforming problem using the LMS and power control; Section 3 presents the Q-learning algorithm, based on RL, which is the main focus of this work; Section 4 presents a proposal for an intelligent agent, employing RL; Section 5 shows the results obtained for simulations involving the agent and the sensor array; Section 6 provides the main conclusions of the work.

2. Adaptive Sensor Arrays

Figure 1 shows a functional diagram of an adaptive sensor linear array with *K* elements. Each *k*-th element of the antenna array is spacing d should generally be equal to $\lambda/2$ (λ is the wavelength). The signal $x_{i,k}(n)$ received by the *k*-th antenna element is given by:

$$x_{i,k}(n) = \sum_{i=1}^{M} \rho_i v_i(n) e^{-j(k-1)\left(\frac{2\pi}{\lambda}\right) d\, cos(\theta_i)} + r_k(n) \quad (1)$$

where $v_i(n)$, ρ_i, and θ_i are the signal, the angle of arrival (AOA) and the attenuation of the i-th source, respectively. All sources have the same wavelength and $r_k(n)$ is the noise associated with each antenna element. The signal, $v_i(n)$, is modeled as a narrowband signal.

In many cases, overall reception performance is measured effectively using the SINR in the output of the array. The SINR estimates the ratio between the signal of interest with a view to noise plus interference, providing a measure of the quality of communication.

In an array of sensors, the adaptive process is conducted by means of adjusting the coefficients (weights) associated with each of the elements of the array. This adjustment is performed using a signal processor and considers a performance criterion established for the system, which could be the SINR, the mean square error, the bit error rate (BER), or any other parameter [28–30].

The desired characteristics of irradiation/reception, as well as the spatial filtering in an sensor array, are configured by convenient manipulation of certain array parameters, such as the number of elements (sensors), the geometry (spatial arrangement and spacing between the elements), types of antennas, and the coefficients (weights) used to adjust the signal amplitude and phase in each element.

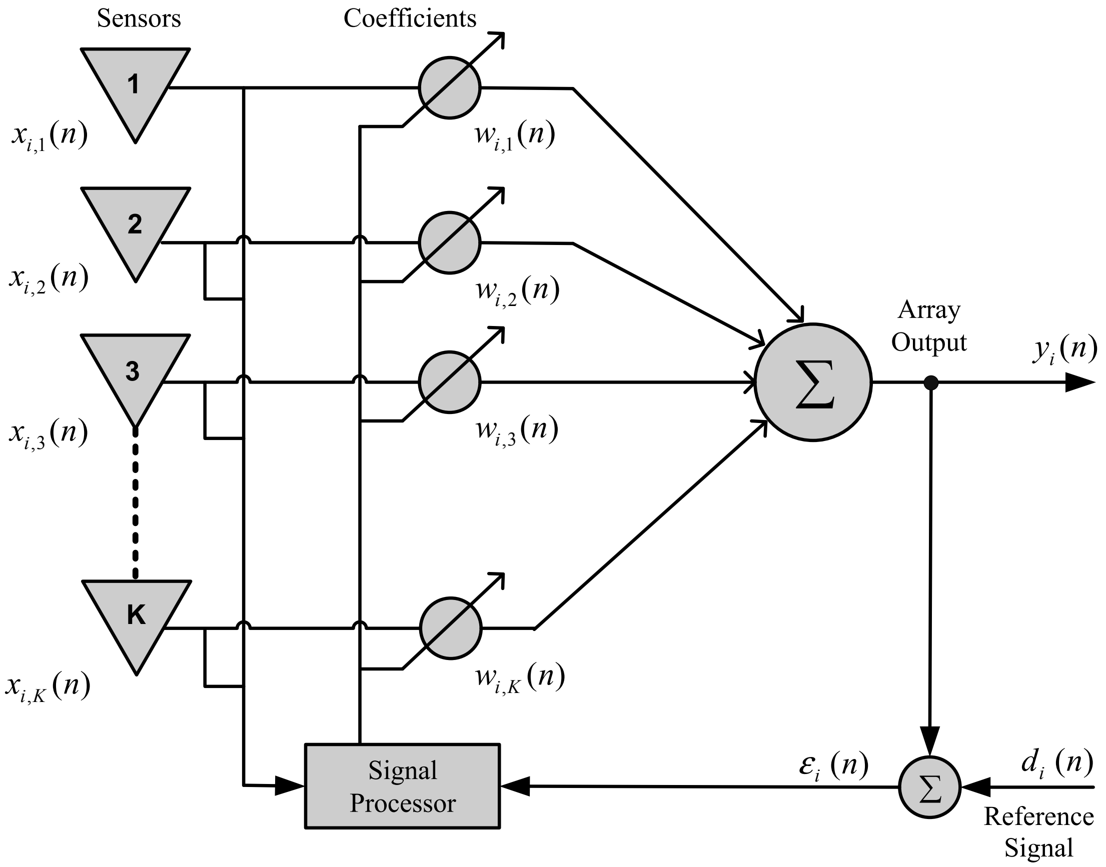


Figure 1. Functional diagram of an adaptive array.

In the system shown in Figure 1, the array output signal for the *i*-th user is given by:

$$y_i(n) = \boldsymbol{w}_{i,K}^H(n)\boldsymbol{x}(n) \tag{2}$$

where $\boldsymbol{w}_{i,K}^H(n)$ is the vector of weights for the i-th user of the system, and the index H represents the Hermitian conjugate transpose. The array output signal $y_i(n)$ is compared to the desired response $d_i(n)$, the difference between them is called the estimation error, as illustrated in Figure 1. As presented in [28], the reference signal (or desired signal) is a training sequence understood by the sensor array, sent periodically by the sources.

2.1. Beamforming

The objective of beamforming is to adjust the weight vectors in order to obtain the maximum SINR in the output of the array. This can be achieved by minimizing the total interference in the array output, while maintaining a constant gain for the signal of interest [5,29,30]. Taking the array output given in Equation (2), the average total output power (in W) for the *i*-th user can be written as:

$$P_i = E[y_i(n)^2] = E[\boldsymbol{w}_{i,K}^H(n)\boldsymbol{x}(n)\boldsymbol{x}^H(n)\boldsymbol{w}_{i,K}(n)] \tag{3}$$

where *E* is the mean operator. Defining:

$$R = E[\boldsymbol{x}(n)\boldsymbol{x}(n)^H] \tag{4}$$

as the autocorrelation matrix of the input signal, the average total output power of the array is given by:

$$P_i = \boldsymbol{w}_{i,K}^H(n) R \boldsymbol{w}_{i,K}(n) \tag{5}$$

The weight vectors that maximize the SINR of the array output can then be found using the following minimization problem:

$$\min_{(\boldsymbol{w}_{i,1},\dots,\boldsymbol{w}_{i,K},P_1,\dots,P_k)} \left(\sum_{i=1}^{K} P_i \right) \\ \text{subject to } SINR_i \geq \delta_i \tag{6}$$

where δ_i, P_i, and $\mathbf{w}_{i,K}$ are, respectively, the smallest allowed SINR value in dB (the pre-established threshold level), the transmission power, and the beamforming vector for the i-th user. The aim is to obtain an optimum pair of the weight and transmission power vectors, with minimization of the total transmission power, maintaining the SINR above a pre-established threshold (δ_i).

2.2. LMS Algorithm

The LMS algorithm is a method based on gradient search techniques, applied to mean square error functions, employing optimum solution of the Wiener-Hopf equation. The algorithm is based on the steepest descent method [31], in which changes in the weight vectors are made along the contrary direction of the estimated gradient vector. This can be described by:

$$\boldsymbol{w}_{i,K}(n+1) = \boldsymbol{w}_{i,K}(n) - \mu \hat{\nabla}(n) \tag{7}$$

where μ is a scaling constant that controls the rate of convergence and stability of the algorithm (adaptation step), and $\hat{\nabla}(n)$ is the vector gradient estimated from the quadratic error in relation to $\boldsymbol{w}_{i,K}(n)$.

The error is obtained between the output of the filter, ($y_i(n) = \boldsymbol{w}_{i,K}^H(n)\boldsymbol{x}(n)$), and the reference signal, $d_i(n)$, so that:

$$\varepsilon_i(n) = d_i(n) - \boldsymbol{w}_{i,K}^H(n) x(n) \tag{8}$$

The stop criterion or convergence criterion of the weights using LMS algorithm is the variation of the mean square error at each iterarion, *i.e.*:

$$|\varepsilon_i{}^2(n) - \varepsilon_i{}^2(n-1)| < \xi \tag{9}$$

where ξ ia threshold defined by the designer based on their application that limits the number of iterations.

2.3. Power Control

An important benefit derived from beamforming and consequent increase in the SINR is the possibility of reducing the signal transmission power. This improves the energy efficiency of the system and reduces interference between the users, ensuring that each signal is transmitted with the lowest power required to maintain a good quality connection.

The control of power in sensor arrays is based on a selected quality criterion, which in the present case was the SINR. During this process, the power is reduced in order to satisfy the restrictions shown in Equation (6). Based on papers presented at [8–11], the updating of the power is given by:

$$P_i(n+1) = P_i(n)\frac{\delta_i}{SINR_i} \tag{10}$$

where it can be seen that convergence is achieved when $SINR_i = \delta_i$. In the calculation of Equation (10), it is necessary to know the value of the $SINR_i$ in the moving terminal, which can be estimated from the minimum mean square error of the beamforming step (LMS), as described by:

$$SINR_i = \frac{1 - min\, E[\varepsilon_i{}^2(n)]}{min\, E[\varepsilon_i{}^2(n)]} \tag{11}$$

where $SINR_i$ represents the estimated SINR for each user.

Rearranging Equation (10) as a function of the minimum mean square error gives the final expression for calculation of the power:

$$P_i(n+1) = \delta_i P_i(n)\frac{min\, E[\varepsilon_i{}^2(n)]}{1 - min\, E[\varepsilon_i{}^2(n)]} \tag{12}$$

3. Reinforcement Learning

Reinforcement learning is a technique whereby an apprentice agent attempts to maximize a performance parameter based on the reinforcement it receives while interacting with an unknown environment. Its use is recommended when there are no a priori models available, or when it is not possible to obtain appropriate examples of situations to which the apprentice agent will be exposed. The agent that lacks previous knowledge learns by means of interaction with the environment, being rewarded for its actions and thereby discovering the optimum policy [32].

In a reinforcement learning system, the state of the environment is represented by a set of variables, known as the state space, which are perceived by the senses of the agent. An action chosen by the agent changes the state of the environment, and the value of this transition of states is passed to the environment by means of a scalar reinforcement signal (reward signal). The objective of the technique is to lead the agent to selection of the sequence of actions that would tend to increase the sum of the reward signal values.

The agent moves autonomously in the state space, interacting with the environment and learning about it through experimentation. Each time that the agent performs an action, an external training entity (critic), or even the environment, can give it a reward or a penalty, indicating how desirable it would be to reach the resulting state [33]. Hence, the reinforcement does not always signify an advance, as it can also inhibit the agent in relation to the action executed. Figure 2 provides a generic scheme of the notion of learning by reinforcement.

The goal of the RL method is to guide the agent towards taking actions that would result in maximizing (or minimizing) the sum of the reinforcement signals (numerical reward or punishment) received over the course of time, known as the expected return, which does not always signify maximizing the immediate reinforcement to be received [34].

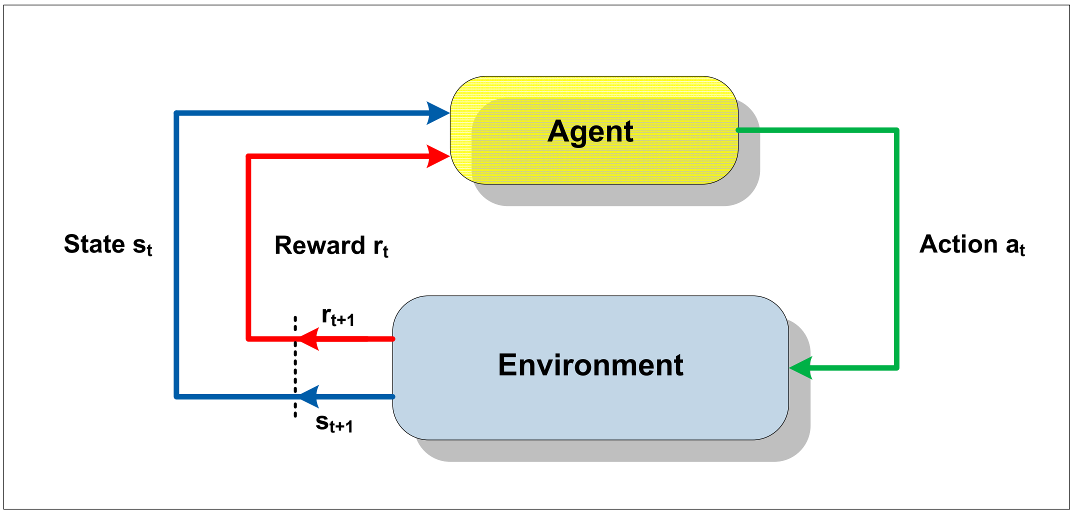


Figure 2. Scheme of interaction between the agent and environment.

The expression describing the sum of the reinforcement signals in an infinite horizon is given by:

$$R_t = r_{t+1} + \gamma . r_{t+2} + \cdots = r_{t+1} + \sum_{k=1}^{\infty} \gamma^k . r_{t+k+1} \tag{13}$$

where R_t represents the return (sum of the reinforcements) received over the course of time, r_{t+k} is the immediate reinforcement signal, and γ is the discount factor, defined in the interval $0 \leq \gamma \leq 1$, that ensures that R_t is finite. If $\gamma = 0$, the agent has a myopic view of the reinforcements, maximizing only the immediate reinforcements. If $\gamma = 1$, the reinforcement view covers all future states giving the same importance to gains at the moment and any future gain.

The behavior that the agent should adopt in order to achieve maximization (or minimization) of the return is known as the policy and can be expressed by π. According to [34], a policy *π (s,a)* is a mapping of states (*s*) in actions (*a*) taken in that state, and represents the probability of selecting each one of the possible actions, in such a way that the best actions correspond to the greatest probabilities of selection. When this mapping maximizes the sum of the rewards, the optimum policy has been achieved.

Evaluation of the quality of the actions taken by the agent involves application of the concept of state-action value function, *Q(s,a)*, which is a value that provides an estimate of how good it is for the agent to be in a given state (*s*) and take a given action (*a*), when it is following any policy π. The term *Q(s,a)* represents the expected value of the total return for the state $s_t = s$ (the present state), which is the sum of the reinforcements, taking into account the rate of discount (γ), as described in expression:

$$Q^{\pi}(s,a) = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right\} \tag{14}$$

Two central questions in relation to reinforcement learning are presented in [32]:

- Given a policy *π(s,a)*, what is the best way to estimate *Q(s,a)*?
- Given an affirmative response to the preceding question, how can this policy be modified so that *Q(s,a)* approaches the optimum value of this function, and how can the consequent corresponding optimum policy be obtained?

The literature suggests various algorithms that can be used in replying to these questions. However, in the present work it was decided to use the Q-learning algorithm developed by [35], which offers advantages including the fact that it can directly approach the optimum value of *Q(s, a)*, irrespective of the policy employed. The values of *Q(s, a)* are updated according to:

$$Q(s,a) = Q(s,a) + \alpha[r_{t+1} + \gamma.max_a Q(s_{t+1},a) - Q(s,a)] \tag{15}$$

where α is the learning rate $(0 \leq \alpha < 1)$ and γ is the discount rate $(0 \leq \gamma < 1)$.

The Q-learning algorithm is presented in Algorithm 1. The episode mentioned in this algorithm is characterized by a sequence of states ending in a final state.

Algorithm 1. Q-learning algorithm.

```
Initialize $Q(s, a)$ randomly;
Repeat (for each episode)
    Initialize $s$;
    Repeat
        Choose $a$ for $s$ using the policy $\pi$;
        Given the action $a$, watch $r$, $s'$;
        $Q(s,a) = Q(s,a) + \alpha[r_{t+1} + \gamma.max_a Q(s_{t+1},a) - Q(s,a)]$
        $s \rightarrow s'$;
    until the final state is reached;
until the number of episodes is reached
```

Given that convergence of the algorithm can only be guaranteed if all the state-action pairs are visited an infinite number of times, selection of the policy to be used in the Q-learning algorithm must ensure that all the pairs have non-null probability of being visited. This can be achieved using an ε-greedy policy, defined by:

$$\pi(s,a) = \begin{cases} 1 - \varepsilon + \dfrac{\varepsilon}{|A(s)|}, if\ a = a^* = arg \max_a Q(s,a) \\ \dfrac{\varepsilon}{|A(s)|}, a \neq a^* \end{cases} \tag{16}$$

This policy consists in a choice of the associated action to the highest value of Q with $1 - \varepsilon + \frac{\varepsilon}{|A(s)|}$ probability and random selection of any other action with $\frac{\varepsilon}{|A(s)|}$ probability, where $|A(s)|$ is the number of possible actions to be executed from s, and ε is the control parameter between greed and randomness.

4. Proposed Solution (Intelligent Agent Project)

In this problem, the aim of the reinforcement learning modeling is to find an optimum policy able to indicate the most suitable techniques that the agent should select among the actions available, considering beamforming and power control.

From the point of view of reinforcement learning, the problem can be modeled as follows: the state of the environment is represented by a discrete set of SINR values, so that S = {$SINR_1$, $SINR_2$, …, $SINR_m$}, where $SINR_m$ represents the maximum value defined for the SINR.

In each state $s \in S$, the deciding agent must select an action a from a set of actions available in the state s, denoted by $A(s)$. The possible actions available for each state are beamforming (BF) and power control (PC). The Q-learning algorithm governs the decision to explore or take advantage, using the policy known as ε-greedy. This policy is defined in the algorithm for selection of the action that possesses the highest value for the utility of the state (greedy criterion), with probability $(1 - \varepsilon)$, and for random action, with probability ε.

As a consequence of selection of an action $a \in A(s)$, starting from state s and at instant of decision t, the deciding agent receives a reward $r_t(s,a)$. The selection of action a alters the perception of the agent in relation to the environment, leading to a new state s_{t+1} that conducts it to a new instant of decision $t + 1$. When the reward is positive, it is seen as a profit or prize, and when it is negative, it is seen as a cost or punishment. In definition of the return function (reward), an indication should be provided of the objective to be achieved by the algorithm. In this problem, an attempt is made to minimize the total transmission power, maintaining constant the gain of the desired signal, and keeping the SINR above a pre-established threshold δ. The reward was therefore defined as being positive when the SINR approached the threshold, and negative when the opposite was true.

Once the states, actions, and return function had been defined, the next step was the process of training the agent, as shown in Algorithm 2. At the start of the learning process, the agent has no knowledge of the result obtained by choosing a particular action, so it performs various actions and observes the results. For a while, the agent explores many actions that result in increasingly greater rewards, and gradually tends to repeat them (exploration); after this, it acquires knowledge from the actions, and can sometimes learn to repeat those that result in greater rewards (exploitation).

Algorithm 2 requires information for the parameters (line 1): learning coefficient (α), parameter for regulation of the greedy criterion (ε), discount rate (γ), and LMS stop criterion (ξ). Subsequently, the matrix Q is initiated using random values (line 2), and the initial state is determined randomly within the values defined in s (line 4). An available action is chosen for s (line 6), and according to the action selected, the corresponding technique (BF or PC) is executed. If the action selected is BF, the algorithm will adjust the weights while the determined condition (line 8) remains satisfied, and if not, will update the transmission power in accordance with the equation presented in Algorithm 2 (line 16).

Assuming that the initial action PC is chosen, a random value was given for the *min_mse* variable equal to 2 (which is the minimum value of the error) and then the powers were updated. Otherwise, *i.e.*, if the chosen initial action is BF, the *min_mse* used to update the powers is based on the last iteration of the error vector ε_i in BF operation. The power is a vector of positions, where M is the number of signals that are addressing the array of sensors. In executing the available actions for s, the reward (r) (line 19) and a new state (s') are reached, according to the equation in line 20, and the matrix Q is updated (line 21). The process (lines 6–22) repeats itself until a final state is found, which in this work was defined by the SINR threshold. After attaining the final state, the algorithm is executed (from line 4) until the defined number of repetitions is completed. The stages of the Q-learning algorithm applied to the sensor array in the agent training step are shown in Algorithm 2.

At the end of the training, a Q matrix is constructed, which is utilized in the functioning of the agent. It is important to note that for each BF action, there is storage of a matrix $\boldsymbol{W}(s, 1)$, containing all the optimum weight values, together with a matrix $\boldsymbol{P}(s, 2)$, for each PC action, corresponding to all the transmission power updates. Whenever the environment is changed, a new execution of Algorithm 2 is required.

Algorithm 2. Sensor array Q-learning algorithm.

```
Require: (α, ε, γ, $\xi, M$)
Initialize Q (s, a)
While maximum episodes are not reached Do
    Initialize s                                  % initial state
    While final state is not reached Do
        Choose a according to ε-greedy rule; % a =1(BF) or a=2 (PC)
        If a==1                               % Beamforming
            While $|{\varepsilon_i}^2(n) - {\varepsilon_i}^2(n-1)| < \xi$ Do      $\% \, threshold\ less\ than\ 10^{-4}$
                For i=1, 2, …, $M$
                    $w_i(n+1) = w_i(n) + \mu\boldsymbol{x}(n)\varepsilon_i(n)$
                End-For
            End-While
        End - If
        If a==2                               % Power Control
                For i=1, 2, …, $M$
                    $P_i = \delta P_i \frac{min\_mse}{1-min\_mse}$ %Updated powers
                End-For
        End - If
        Watch r % Reward
$s' = (1-min\_mse)/min\_mse$                      % New State
$Q(s,a) = Q(s,a) + \alpha[r_{t+1} + \gamma.max_a Q(s_{t+1},a) - Q(s,a)]$ % Updating Q-table
        s → s'
        End-While
End-While
return Q(s,a) % Q-values Matrix
```

Algorithm 3 shows the functioning of the agent, where *j* is the processing cycle, corresponding to selection of an initial state, execution of an action, and attainment of a new state.

Algorithm 3. Algorithm for functioning of the agent.

```
If (environment changes)
training % Algorithm 2
Else
    Choose an initial state s
    For j from 1 until max_value Do
Choose $a_{max}$ for s and run
        s → s'
    End-For
```

5. Simulation and Results

The functioning of the algorithm was demonstrated for two sources ($M = 2$) using simulation of a situation with angles of 90° and 30° for the desired and interferent signals, respectively. The target SINR was 2 dB, and the initial powers were set at 1 W for both sources. All sources were modeled with polar binary signals ($v_i(n) \in \{-1,1\}$) random uniform distribution. The noise at each antenna element was modeled as Additive white Gaussian noise (AWGN) with variance σ^2. The parameters used in the simulation are listed in Table 1.

Table 1. Simulation Parameters.

Parameters	Value
Number of elements in the array (K)	8
Number of signals (M)	2
Initial transmit power (P_0)	1 W
SINR threshold (δ)	2 dB
Step Adaptation (μ)	0.001
Noise Variance (σ^2)	0.1
Distance between each element of the array (d)	$\lambda/2$
Learning Rate (α)	0.1
Discount Factor (γ)	0.9
Greedy Rule (ε)	0.2
Attenuation of the i-th source (ρ_i)	1

In this paper, a linear array of sensors with eight elements and two signal sources was used for a desired source and an interfering source. A linear array with K elements can create up to $K - 1$ nulls in the direction of the interfering source. When the number of unwanted (interfering) sources is close to such a limit, the attenuation of unwanted signals is reduced and there are excess gains (greater than the gain attributed to the desired signal) in the proximity of the desired and undesired angles. Thus, the use of two sources is in the range between the limits established for the performance of the system.

The distance between the array elements (d) is limited by the value $\lambda/2$. This limitation avoids the production and overlapping of side lobes. The unit transmission powers were used to initialize the power control algorithm and to facilitate the calculations. The choice of μ in the adaptation step is experimentally determined in order to provide stability of the algorithm. The higher the adaptation step value, the higher the convergence speed. However, the excess error also becomes larger, which is undesirable. The α, γ, and ε values were obtained after several simulations.

The states of the environment were represented by discrete SINR values between −0.8 and 5, which corresponded to index values in the range 1–18. Each action was identified with the label 1 or 2, indicative of beamforming and power control, respectively. An ε-greedy policy was adopted, with 80% possibility of selecting the better action.

Two simulations were performed using the parameters shown in Table 1, but with the variation of the noise (σ^2) changed to 0.3 in the second simulation. In each simulation, the training was configured to execute 10, 50 and 250 episodes.

The results of the first simulation are shown in Table 2, indicating the policies obtained after each different episode. Each line corresponds to a SINR value, and the columns correspond to the beamforming (BF) or power control (PC) processes.

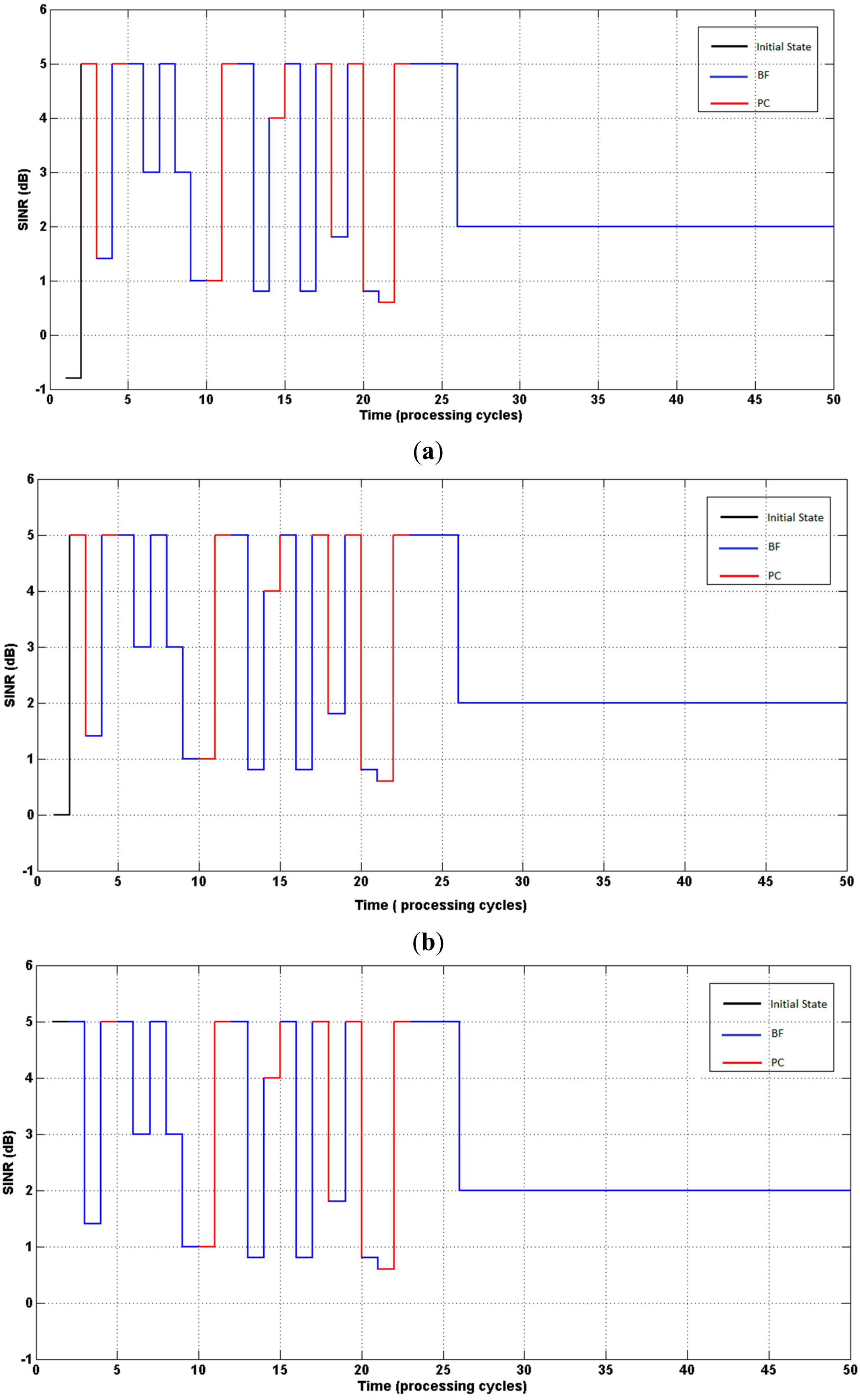
Table 2. Policy Improvement of the agent.

		Policy 10		**Policy 50**		**Policy 250**	
INDEX	**SINR(dB)**	**BF**	**PC**	**BF**	**PC**	**BF**	**PC**
1	−0.8	0.5	0.5	1	0	0	1
2	−0.6	0.5	0.5	1	0	1	0
3	−0.4	0.5	0.5	1	0	0	1
4	−0.2	0	1	1	0	1	0
5	0	0	1	0	1	0	1
6	0.2	0	1	0	1	0	1
7	0.4	0	1	0	1	0	1
8	0.6	0	1	0	1	0	1
9	0.8	1	0	1	0	0	1
10	1	0	1	0	1	0	1
11	1.2	0	1	1	0	0	1
12	1.4	0	1	0	1	0	1
13	1.6	0	1	0	1	0	1
14	1.8	0	1	0	1	0	1
15	2	**Destination**		**Destination**		**Destination**	
16	3	1	0	0	1	1	0
17	4	1	0	1	0	1	0
18	5	1	0	1	0	1	0

The values given in Table 2 correspond to the probability of selecting each technique, for the range of discretized SINR values. The optimum policy was obtained after 250 episodes, and was adopted for testing the agent.

Figure 3 present (on the ordinate axis) the states (SINR), and the curves indicate the evolution of the SINR until reaching the target value (δ = 2).

The switching sequence using an initial SINR of −0.8 is illustrated in Figure 4, with the actions executed (beamforming or power control) indicated on the ordinate axis, and the curve showing the order of execution in each processing cycle. Index 1 indicates execution of beamforming, and Index 2 indicates power control.



(**a**)

(**b**)

(**c**)

Figure 3. (**a**) System response. Agent started with SINR = −0.8 dB; (**b**) System response. Agent started with SINR = 0 dB; (**c**) System response. Agent started with SINR = 5 dB.

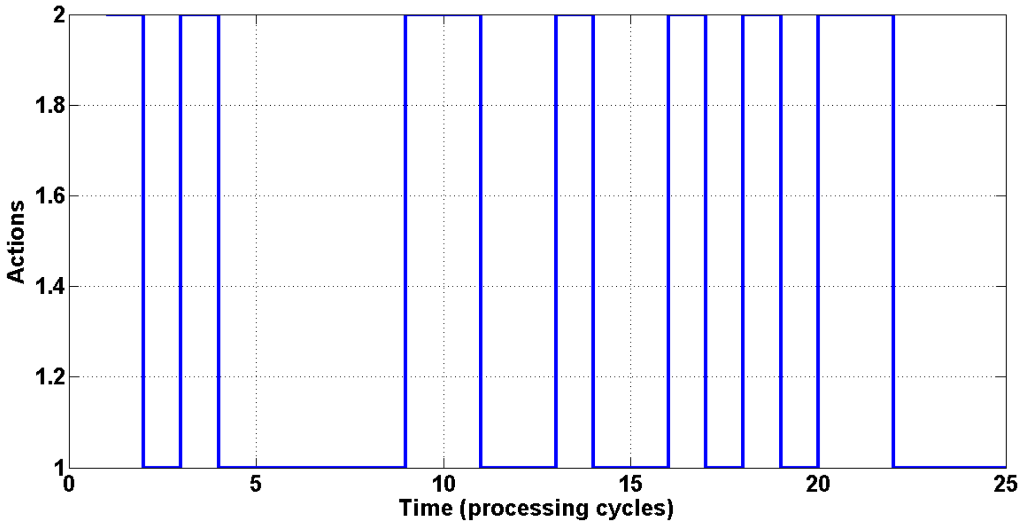


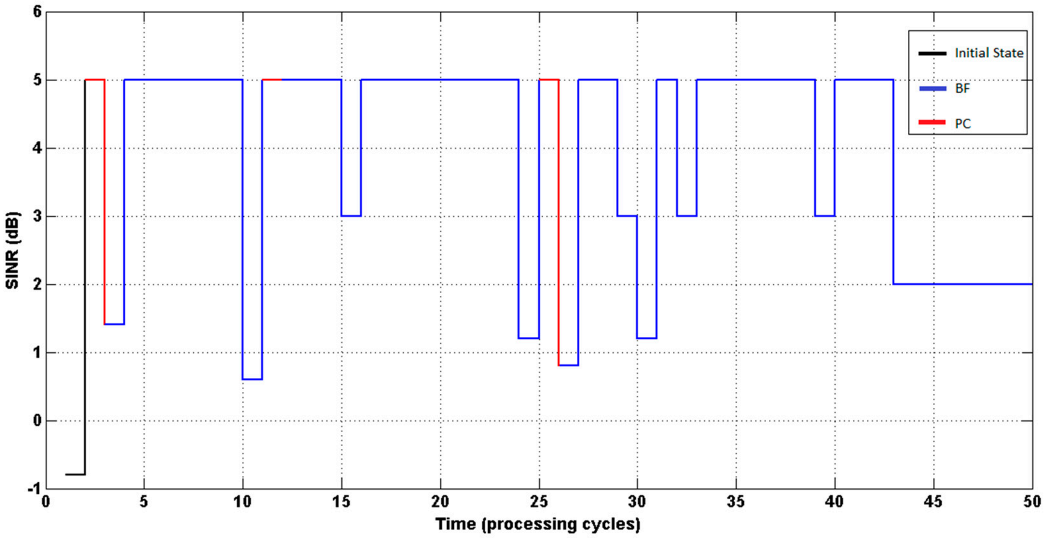
Figure 4. The switching sequence among the two techniques.

Table 3 presents the simulation results obtained using the same parameters, but with variance of the noise equal to 0.3.

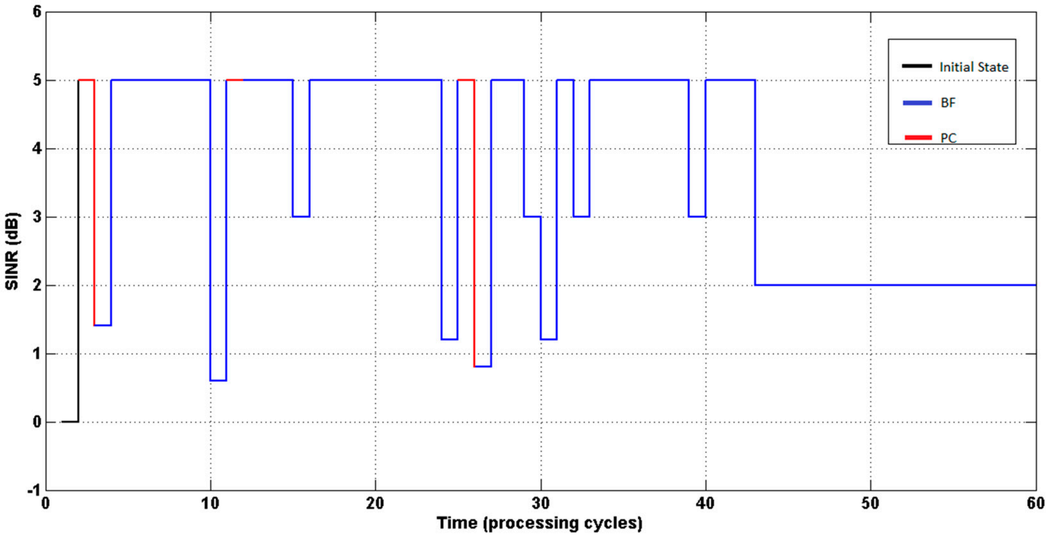
Table 3. Policy Improvement of the agent.

		Policy 10		Policy 50		Policy 250	
INDEX	**SINR(dB)**	**BF**	**PC**	**BF**	**PC**	**BF**	**PC**
1	−0.8	0.5	0.5	0	1	0	1
2	−0.6	0.5	0.5	0	1	0	1
3	−0.4	0.5	0.5	1	0	0	1
4	−0.2	0	1	1	0	0	1
5	0	0.5	0.5	0.5	0.5	0	1
6	0.2	0	1	0	1	0	1
7	0.4	0	1	0	1	0	1
8	0.6	0	1	0	1	0	1
9	0.8	1	0	0	1	0	1
10	1	0	1	0	1	0	1
11	1.2	0.5	0.5	0	1	0	1
12	1.4	0	1	0	1	1	0
13	1.6	0	1	0	1	0	1
14	1.8	0	1	0	1	0	1
15	2	**Destination**		**Destination**		**Destination**	
16	3	0	1	1	0	1	0
17	4	0	1	1	0	1	0
18	5	1	0	1	0	1	0

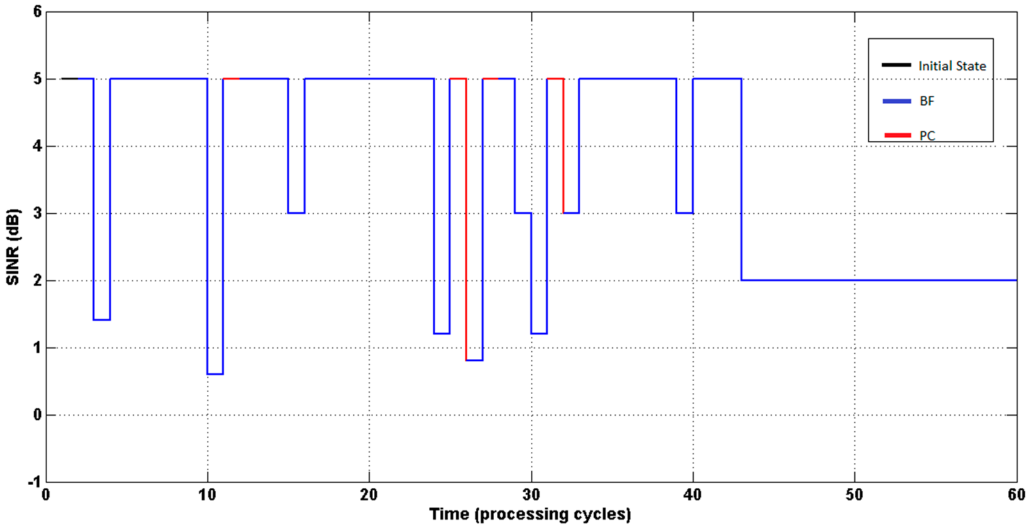
From Tables 2 and 3, it can be seen that power control was selected in more states, compared to beamforming, which demonstrates the independence of the algorithm in selecting the optimum policy, in contrast to other procedures [8–10,17] in which only the power is updated once beamforming convergence is reached. Figure 5 illustrate the evolution of the SINR until the target value is reached (second simulation).



(**a**)



(**b**)



(**c**)

Figure 5. (**a**) System response. Agent started with SINR = −0.8 dB; (**b**) System response. Agent started with SINR = 0 dB; (**c**) System response. Agent started with SINR = 5 dB.

Figure 6 presents the switching sequence for an initial SINR of 5, where the ordinate axis shows the actions executed (beamforming or power control) and the curve indicates the order of execution in each processing cycle. Index 1 indicates the execution of beamforming and Index 2 indicates the execution of power control.

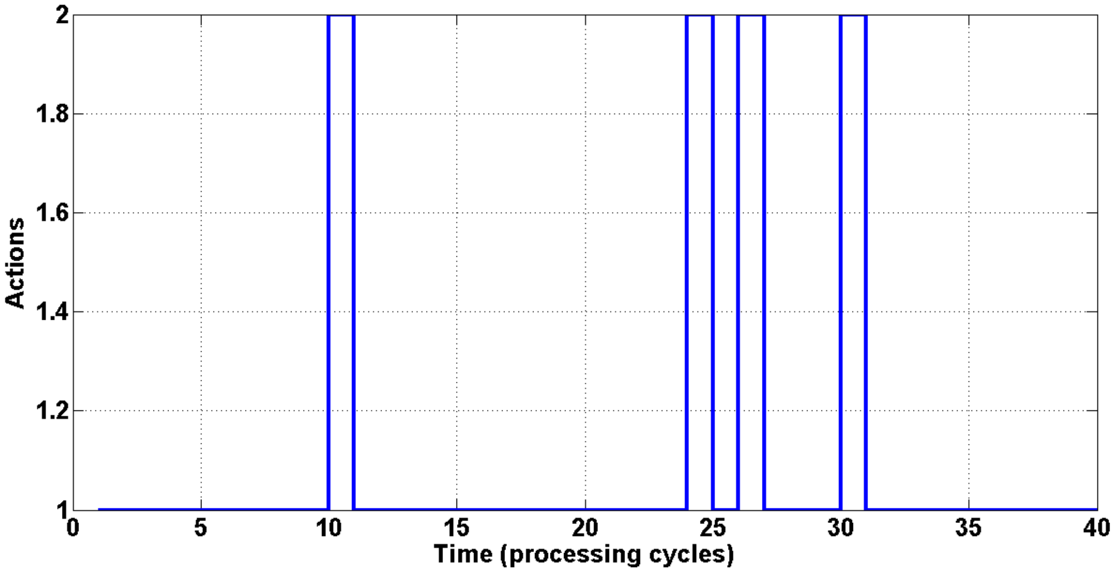


Figure 6. The switching sequence among the two techniques.

It can be seen that even with increase of the variance of the noise from 0.1 to 0.3, the proposed algorithm was effective in selecting the optimum policy, requiring only one new training, given that the environment was modified. This is an important point, and shows the robustness of the algorithm when faced with new noise conditions.

6. Conclusions

The use of beamforming and power control individually in sensor arrays has its benefits, but it can be seen that by employing them jointly there is an increase in system performance. Several studies have used beamforming and power control jointly, but this paper presents a new control method employing a combination of beamforming and power control. The algorithm presented uses the technique of reinforcement learning to obtain the optimum policy for selection between beamforming and power control in sensor arrays.

It can also be seen that the proposed technique reduces the computational cost, as the techniques are selected independently. For example (Table 2), using the optimum policy obtained after 250 episodes, it was found that in many cases it was not necessary to execute the LMS algorithm. This resulted in lower computational cost and reduced complexity of the proposed method.

From the simulations performed, it could be concluded that reinforcement learning offers an effective way of implementing a policy of switching between beamforming and power control in sensor arrays, benefiting from the advantages of both techniques.

Author Contributions

All the authors have contributed in various degrees to ensure the quality of this work. Náthalee C. Almeida, Marcelo A.C. Fernandes and Adrião D.D. Neto conceived the idea and experiments; Náthalee C. Almeida designed and performed the experiments; Náthalee C. Almeida, Marcelo A.C. Fernandes and Adrião D.D. Neto analyzed the data and wrote the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Rambach, K.; Yang, B. Direction of arrival estimation of two moving targets using a time division multiplexed colocated MIMO radar. In Proceedings of the 2014 IEEE Radar Conference, Cincinnati, OH, USA, 19–23 May 2014; pp. 1118–1123.
2. Zhang, W.; Liu, W.; Wu, S. Joint transmission and reception diversity smoothing for direction finding of coherent targets in MIMO radar. *IEEE J. Sel. Top. Signal Process.* **2014**, *9*, 115–124.
3. Gu, J.F.; Chan, S.C.; Zhu, W.P.; Swamy, M.N.S. Joint DOA estimation and source tracking with kalman filetring and regularized QRD RLS algorithm. *IEEE Trans. Circuits Syst. II Express Briefs* **2013**, *60*, 46–50.
4. Schmidt, J.F.; Lopez-Valcarce, R. Antenna competition to boost active interference cancellation in cognitive MIMO-OFDM. In Proceedings of the 8th Sensor Array and Multichannel Signal Processing Workshop (SAM), A Coruna, Spain, 22–25 June 2014; pp. 269–272.
5. Balanis, C.A. *Antenna Theory: Analysis and Design*, 3rd ed.; Wiley-Interscience: New York, NY, USA, 2005.
6. Rashid-Farrokhi, F.; Tassiulas, L.; Ray Liu, K.J. Joint optimal power control and beamforming in wireless networks using antenna arrays. *IEEE Trans. Commun.* **1998**, *46*, 1313–1324.
7. Rashid-Farrokhi, F.; Ray Liu, K.J.; Tassiulas, L. Transmit beamforming and power control for cellular wireless systems. *IEEE J. Sel. Areas Commun.* **1998**, *16*, 1437–1450.
8. Pitz, C.A.; Vanti, M.G.; Tobias, O.J.; Seara, R. Adaptive Beamforming for Antenna Arrays in Cellular Systems Based on a Duality between Uplink and Downlink Channels. In Proceedings of the 7th International Telecommunications Symposium (ITS), Manaus, Brazil, 6–9 September 2010.
9. Visotsky, E.; Madhow, U. Optimum beamforming using transmit antenna arrays. In Proceedings of the 49th IEEE Vehicular Technology Conference, Houston, TX, USA, 16–20 May 1999; Volume 1, pp. 851–856.
10. Yu, W.; Lan, T. Downlink beamforming with per-antenna power constraints. In Proceedings of the 6th IEEE Workshop on Signal Processing Advances in Wireless Communications, New York, NY, USA, 5–8 June 2005; pp. 1058–1062.
11. Huang, Y.; Tan, C.W.; Rao, B.D. Joint beamforming and power control in coordinated multicell: Max-min duality, effective network and large system transition. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 2730–2742.

12. Khandaker, M.R.A.; Rong, Y. Joint power control and beamforming for peer-to-peer MIMO relay systems. In Proceedings of the 2011 International Conference on Wireless Communications Signal Processing (WCSP), Nanjing, China, 9–11 November 2011.
13. Lu, X.; Li, W.; Tolli, A.; Juntti, M.; Kunnari, E.; Piirainen, O. Joint power control, receiver beamforming and adaptive multi base station coordination for uplink wireless communications. In Proceedings of the 21st IEEE International Symposium on Personal, Indoor and Mobile radio Communications Workshops, Instanbul, Turkey, 26–30 September 2010; pp. 446–450.
14. Hu, C.; Wang, F.; Wang, W. Joint beamforming and power control optimization by second-order cone programming approximation. In Proceedings of the 12th IEEE International Conference on Communication technology (ICCT), Nanjing, China, 11–14 November 2010; pp. 147–150.
15. You, S.; Noh, G.; Lee, J.; Wang, H.; Hong, D. Joint beamforming and power control algorithm for cognitive radio network with the multi-antenna base station. In Proceedings of the IEEE Communications and Network Conference (WCNC), Sydney, Australia, 18–21 April 2010; pp. 1–6.
16. Li, Z.; Yin, C.; Yue, G. A novel approach to joint beamforming and power control for the coordinated multicell multi-antenna system. In Proceedings of the 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), Shanghai, China, 21–23 September 2012; pp. 1–4.
17. Zeydan, E.; Kivanc-Tureli, D.; Tureli, U. Iterative beamforming and power control for MIMO ad hoc networks. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), Miami, FL, USA, 6–10 December 2010; pp. 1–5.
18. Gupta, N.; Reddy, A.L.N. Adaptive antenna using Fuzzy Logic Control. In Proceedings of the IEEE Applied Electromagnetics Conference (AEMC), Kolkata, India, 19–20 December 2007; pp. 1–4.
19. Noori, N.; Razavizadeh, S.M.; Attar, A. Joint beamforming and power control in MIMO cognitive radio networks. *IEICE Elect. Express* **2010**, *7*, 203–208.
20. Yoshida, J.; Hirose, A. Beamforming for impulse-radio UWB communication systems based on complex-valued spatio-temporal neural networks. In Proceedings of the International Symposium on Electromagnetic Theory, Hiroshima, Japan, 20–24 May 2013; pp. 848–851.
21. Zaharis, Z.D.; Skeberies, C.; Xenos, T.D.; Lazaridis, P.I.; Cosmas, J. Design of a novel antenna array beamformer using neural networks trained by modified adaptive dispersion invasive weed optimization based data. *IEEE Trans. Broadcast.* **2013**, *59*, 455–460.
22. Terabayashi, K.; Hirose, A. Ultra-short-pulse acoustic imaging using complex-valued spatio-temporal neural-network for null-steering: Experimental results. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 3410–3413.
23. Liu, Y.; Zhang, P.; Hain, T. Using neural network front-ends on far field multiple microphones based speech recognition. In Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 5542–5546.
24. Hung, J.-C. An adaptive robust fuzzy beamformer for steering vector mismatch and reducing interference and noise. *Inf. Sci.* **2014**, *266*, 160–170.
25. Anitha, M.; Kurahatti, N.G. Neural Fuzzy Inference Based Robust Adaptive Beamforming. *Int. J. Emerg. Technol. Adv. Eng.* **2013**, *3*, 641–648.

26. Song, X.; Wang, J.; Niu, X. Robust adaptive beamforming algorithm based on neural network. In Proceedings of the IEEE International Conference on Automation and Logistics, Qingdao, China, 1–3 September 2008.
27. Santos, J.P.Q.; Melo, J.D.; Neto, A.D.D. Reactive search strategies using reinforcement learning, local search algorithms and variable neighborhood search. *Expert Syst. Appl.* **2014**, *41*, 4939–4949.
28. Balanis, C.A.; Ioannides, P.I. *Introduction to Smart Antennas*; Morgan & Claypool Publishers: San Rafael, CA, USA, 2007.
29. Monzingo, R.A.; Miller, T.W. *Introduction to Adaptive Arrays*; Scitech Publishing Inc.: Raleigh, NC, USA, 2004.
30. Haykin, S. *Adaptive Filter Theory*, 3rd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 1996.
31. Widrow, B.; Mantey, P.E.; Griffiths, L.J.; Goode, B.B. Adaptive antenna systems. *IEEE Proc.* **1967**, *55*, 2143–2159.
32. Lima Junior, F.C.; Melo, J.D.; Dória Neto, A.D. Using the Q-learning algorithm in the constructive phase of the GRASP and reactive GRASP metaheuristics. In Proceedings of the IEEE International Joint Conferecence on Neural Networks, Hong Kong, China, 1–8 June 2008; pp. 4169–4176.
33. Sutton, R.; Barto, A. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
34. Peixoto, H.M.; Diniz, A.A.R.; Almeida, N.C; Melo, J.D.; Dória Neto, A.D.; Guerreiro, A.M.G. Modeling a system for monitoring an object using artificial neural networks and reinforcement learning. In Proceedings of the International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 2327–2332.
35. Watkins, C.J.C.H.; Dayan, P. *Q-Learning: Machine Learning*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1992; pp. 279–292.