A Continuous-State Reinforcement Learning Strategy for Link Adaptation in OFDM Wireless Systems

João Paulo Leite, Robson Domingos Vieira, and Paulo Henrique Portela de Carvalho

***Abstract*—Adaptive modulation and coding is a link adaptation technique that exploits the knowledge of channel state information (CSI) to adapt the transmission parameters such as modulation order and coding rate in order do maximize the transmission throughput. Unfortunately the physical layer transmission parameters are not optimally adjusted due to the changing state of the wireless channel. This paper presents a machine learning approach based on the concept of reinforcement learning and Markov Decision Processes for link adaptation in orthogonal frequency-division multiplexing systems through adaptive modulation and coding. The proposed solution learns the best modulation and coding scheme for a given signal-to-noise ratio by interacting with the the radio channel on a online and real-time fashion and, therefore, a computationally intensive training phase is not required. Simulation results show that the proposed technique outperforms the well-known solution based on look-up tables for adaptive modulation and coding, and it can potentially adapt itself to distinct characteristics of the environment or the receiver radio frequency front end.**

***Index Terms*—Adaptive modulation and coding, continuous-state policy improvement, link adaptation, machine learning, orthogonal frequency-division multiplexing, reinforcement learning.**

I. Introduction

ADAPTIVE modulation and coding (AMC) has been of great interest as one alternative to increase the throughput of wireless communication systems, especially considering third generation (3G) and the next generation (4G) wireless communication systems, for which even higher data rates are expected [1]. AMC exploits the knowledge of channel state information (CSI) to adapt the transmission parameters in order do maximize the link throughput. Currently one of the approaches used for this purpose is the consultation of look-up tables [2]. The main drawback of this strategy when considering orthogonal frequency-division multiplexing (OFDM) and multiple-input multiple-output (MIMO) systems is the large error-rate variance that the tables exhibit for a fixed value of low dimensional link quality metrics (LQM). These metrics are particularly difficult to devise due to sensitivity of the link performance in terms of the defined metrics. Moreover, look-up tables are not obtained in real time, they may require a great amount of memory in order to be stored, and they do not reflect the unique radio-frequency characteristics of each device [3].

Recently, a shift in the paradigm was proposed. The authors of [4], [5] suggest the use of machine learning algorithms as a flexible framework to enable AMC. More specifically, the use of machine learning techniques is first considered in [5], where learning algorithms are envisioned to explore databases using classification algorithms. The databases would supply knowledge of past performance on packet transmissions as a function of physical layer parameters. As expected this approach has limited storage capacity and it requeries constant updating of the databases during the course of wireless network operation.

In [3], [4], [6], the link adaptation is formulated as a classification problem whose solution is obtained via the k-nearest neighbors (kNN) algorithm. The authors propose low dimensional feature set that enables machine learning to increase the accuracy of link adaptation in IEEE 802.11n systems. As a drawback, the authors have proposed an heuristic subcarrier ordering to achieve this feature set that may not be extensible to other systems or standards. Moreover the kNN approach heavily relies upon extensive trainning sets stored on databases.

In [7] support vector machines are used to solve the very same classification problem, while [8] uses an artificial neural network to deal with the link adaptation problem. The application of machine learning algorithms such as those previously cited, as well as other supervised learning approaches, rely heavily on training sets and require large samples of input-output pairs from the function to be learned. Therefore statistics such as the packet error rate or the bit error rate must be known *a priori*. Moreover, their training phase occurs off-line, what makes them not well suited for learning in an environment of high variability as the mobile radio channel. Furthermore, neural networks and support vector machines demand a computationally intensive training process [9].

It is often impractical to obtain examples of desired behavior that are both correct and representative of all the situations that the transmitter might be exposed to, e.g., the wireless channel behavior, impact of amplifier nonlinearities, oscillator phase noise and other radio-frequency (RF) imperfections [10], and non-Gaussian additive noise and interference. The latter is of special concern for cognitive radio scenarios, since the interference cognitive networks differs from that

João P. Leite and Paulo H. P. de Carvalho are with the Microwave and Wireless System Laboratory (MWSL), Department of Electric Engineering, University of Brasília, Campus Universitário Darcy Ribeiro, Asa Norte, CEP 709-900, P.O. Box 4386, Brasilia, DF, Brazil

Robson D. Vieira is with Nokia Technology Institute, SCS Quadra 1, Bloco F, 6o. andar, CEP 70.397-900 Brasília, DF, Brazil

Digital Object Identifier 10.14209/jcis.2015.6

of conventional networks due to the distinct transmission characteristic of a cognitive terminal and a conventional terminal [11]. In this situations the Gaussian assumption may not always hold [12]. In this sense, the techniques previously mentioned are infeasible for on-line learning. This suggests that other approaches should be considered.

Reinforcement Learning and Markov Decision Processes have recently attracted some attention to research in the communications field, specially in the context of cognitive radios. Instead of learning from examples provided by an external supervisor, learning is here accomplished by directly interacting with the environment. In this context, we propose a reinforcement learning (RL) approach to deal with the AMC problem. An external supervisor is not required, since the interactions with the environment provide the learning examples. By using past experiences obtained in real time, an agent can learn the best modulation and coding schemes to be used given the state of the channel and making minimal assumptions about the operating environment. The decision of choosing a modulation and coding scheme is treated as a Markov Decision Process whose objective is to maximize the spectral efficiency of the system.

In this paper, our contributions are: the modeling of adaptive modulation and coding as a k-armed bandit problem whose solution is based on Markov Decision Processes and its solution using a continuous-state reinforcement learning approach. To the best of the authors knowledge, this formulation for the AMC problem has not been presented yet.

Some considerations are required: we have opted for an approach based on a continuum of states. Since the state of the environment is described by the signal-to-noise ratio, determining the best partitioning of the state space can be problematical. Since the AMC thresholds are not known *a priori*, a coarse discretization of the state space may lead to throughput loss in a specific region of operation. A fine discretization leads to a very large number of states that must be dealt by the algorithm. This trade-off is not present in the continuous-state reinforcement learning. Moreover, in order to allow real time operation, this paper proposes a modification of the known On-line Least-Squares Policy Iteration algorithm (LSPI) [13]. The performance of our approach is then compared with the classical approach of look-up tables. Simulations show that the reinforcement learning technique can lead to throughput gains in scenarios with colored interference or uncompensated RF imperfections.

It is important to remind that the main purpose of the paper is not to investigate feature extraction to obtain optimized link quality metrics for link adaptation, like the technique of subcarrier ordering based on post-processing signal-to-noise ratio (SNR) presented in [4] (although this might a critical issue in OFDM systems). Instead, we are concerned mainly with an on-line approach for AMC in order to not depend on off-line training obtained from extensive simulations of the physical layer for each modulation and coding rate.

The remainder of this paper is organized as follows: Section II describes briefly the OFDM system model. Section III presents in detail the theory of Markov Decision Processes and its relation with the reinforcement learning problem. In Section IV the proposed algorithm for continuous-state reinforcement learning is presented. Section V is devoted to present the simulation results, and finally Section VI presents some concluding remarks.

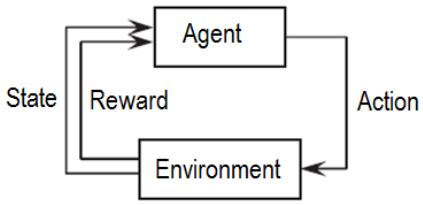


Fig. 1: Reinforcement learning interaction.

II. System Model

We consider an OFDM system model based on the communication procedures of wireless standards such as Third-Generation Partnership Project (3GPP) Long Term Evolution (LTE) and Worldwide Interoperability for Microwave Access (WiMAX). The transmission is done on a packet-basis. A cyclic redundancy check (CRC) field is attached to every packet before it is convolutionaly coded. The length of the bit stream is chosen so that data can be completely transmitted within the transmission time interval. The modulation is uniform in the sense that every subcarrier is modulated with the same M-QAM constellation for each transmitted frame. An adequate guard interval is inserted in each and every OFDM symbol so that intersymbol interference (ISI) can be eliminated at the receiver side.

We assume that the channel may vary considerably between different OFDM symbols depending on the correlation of fading between two successively transmitted symbol, but it does not vary within one OFDM symbol (quasi-static block fading model). At the receiver, the signal is equalized using a zero-forcing (ZF) equalizer and the data is decoded using the Viterbi algorithm. Specific details will be given in Section V.

III. Reinforcement Learning Theory

The basic framework for reinforcement learning (RL) problems is shown in Fig. 1. An *agent* interacts with the environment by selecting actions to take and then perceiving the effects of those actions. This effects are translated into a new state and a reward signal. The objective of the agent is to maximize some measure over the rewards [14]. Unlike supervised learning, the agent must learn from experiences generated by interacting with the environment.

In our system, we are interested in maximizing the throughput for a given state of the environment – determined by the mean (SNR) value over all subcarriers in an OFDM symbol – by selecting the modulation order and the convolutional coding rate. In practice there is only a finite set of admissible combinations between modulation order and coding rate. Every pair of this set is considered an action. The transmitter selects the best modulation and coding scheme just before

each packet transmission. Since we do not discretize the range of SNR values, the problem is classified as continuous-state reinforcement learning. In the next sections we formalize the RL framework using the theory of Markov Decision Processes (MDP), over which the solutions will be constructed.

A. Markov Decision Processes

Reinforcement learning problems can be formalized using the theory of Markov Decision Processes [15]. Initially it is assumed that the environment is a finite-state, discrete-time stochastic dynamic system. Latter a continuous-state extension will be presented.

A Markov Decision Process is defined as a 4-tuple $(\mathcal{S}, \mathcal{A}, P, R)$, where [13]:

- $\mathcal{S} = \{s_1, s_2, \dots s_n\}$ denotes the set of n possible states that describe the dynamics of the environment;
- $\mathcal{A} = \{a_1, a_2, \dots a_m\}$ is the finite set of m possible actions that an agent may choose;
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0; 1]$ is a Markovian transition model, where $P(s, a, s')$ is the probability of making a transition to state $s' \in \mathcal{S}$ when taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$;
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is a reward function, where $R(s, a, s')$ represents the immediate payoff of the environment for the transition from s to s' when taking action a.

It is common to express the transition function as $P(s, a, s') = \Pr(s_{t+1} = s' | s_t = s, a_t = a)$, where s_{t+1} represents the state of the process at time $t + 1$, s_t the state at time t and a_t the action taken after observing state s_t. The fact that there is not any time dependency on P or R as previously stated is due to the stationarity assumption of the MDP [16].

A stationary deterministic policy π defines the agent behavior and consists on a mapping from the states to the actions: $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The notation $\pi(s)$ indicates the action that the agent takes in state s. The state value of the policy π, $V^{\pi}(s)$, also referred as as V-function, is the expected cumulative reward that will be received while the agent follows the policy, starting from state s [9]. In the infinite horizon model, the value of the policy is defined as:

$$V^{\pi}(s) = \mathbb{E}\left\{\sum_{t=0}^{\infty} \gamma^t r_t\right\} \tag{1}$$

where r_t is the reward received at time instant t, $0 \leq \gamma \leq 1$ is a discount factor for future rewards with respect to the immediate reward. The discount factor determines the importance of future rewards. A value close to 0 makes the agent consider only the current reward, while a value close to 1 makes the agent prize a long-term high reward.

As might be expected, the reward depends on the state s of the environment at time t and the action that was taken. In reinforcement learning problems, the objective of the agent is to find an optimal policy $\pi^*(s) \in \mathcal{A}$ for each s that maximizes the cumulative measure of reward as defined in (1). In other words, a policy V^* must be found so that

$$\begin{aligned} V^*(s) &= V^{\pi^*}(s) \\ &= \max_{\pi} V^{\pi}(s), \forall s \in \mathcal{S} \end{aligned} \tag{2}$$

A more convenient way to characterize policies is by using the state-action value function (Q-function) instead of the V-function. The Q-function denotes how good is to perform action a when in state s [9]. It gives the return obtained when, starting on a given state, the agent takes a given action and then follows the policy π thereafter. It is defined as

$$Q^{\pi}(s, a) = \mathbb{E}\left\{\sum_{t=0}^{\infty} \gamma^t r_t | s_{t=0} = s, a_{t=0} = a\right\} \tag{3}$$

Using the fact that the environment is described by a Markovian transition model, (3) can be expressed as

$$\begin{aligned} Q^{\pi}(s, a) &= \mathbb{E}\{r_0 | s_{t=0} = s, a_{t=0} = a\} \\ &\quad + \mathbb{E}\left\{\sum_{t=1}^{\infty} \gamma^t r_t | s_{t=0} = s, a_{t=0} = a\right\} \\ &= \sum_{s' \in \mathcal{S}} P(s, a, s') R(s, a, s') \\ &\quad + \gamma \mathbb{E}\left\{\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_{t=0} = s, a_{t=0} = a\right\} \\ &= \mathcal{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s, a, s') Q^{\pi}(s', a'), \end{aligned} \tag{4}$$

known as Bellman equation, indicates that the Q-function of the current state-action pair can be expressed in terms of the expected immediate reward of the current state-action and the Q-function of the next state-action pair.

It is common to express (4) defining the Bellman operator T_{π} over $Q(s, a)$:

$$T_{\pi}[Q(s, a)] = \mathcal{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s, a, s') Q^{\pi}(s', a') \tag{5}$$

It is worthwhile to remark that for any initial value Q, successive applications of T_{π} over Q converge to the state-action value function Q^{π} of the policy π, since Q^{π} is the fixed point of the Bellman operator [17]. This fact will be used when we introduce the continuous-state reinforcement learning.

The optimal Q-function, $Q^*(s, a)$, is the one that satisfies $Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$. The Bellman's optimality principle states that any policy that selects at each state an action with the largest Q-value (i.e., a greedy policy) is optimal [18]. From (4), we can write that

$$Q^*(s, a) = \mathcal{R}(s, a) + \gamma \sum_{s' \in \mathcal{S}} P(s, a, s') \max_{a' \in \mathcal{A}} Q'(s', a') \tag{6}$$

As a consequence, (2) is written as

$$\begin{aligned} V^*(s) &= \max_{a\in\mathcal{A}} Q^*(s,a) \\ &= \max_{a\in\mathcal{A}} \left[\mathcal{R}(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s,a,s')\, V^*(s')\right] \end{aligned} \tag{7}$$

Once $Q^*(s,a)$ is known, the optimal policy can be determined by taking the action with the highest value among $Q^*(s,a)$ for each state $s \in \mathcal{S}$, i.e.

$$\pi^*(s) = \max_{a\in\mathcal{A}} Q^*(s,a) \tag{8}$$

When the transition model of the environment is known, the optimal policy can be obtained by solving the system of nonlinear equations generated in (7) using techniques such as dynamic programming [14].

A more realistic application of reinforcement learning is when the environment model is not available. In other words, we have no prior knowledge of $\mathcal{R}(s,a)$ and $P(s,a,s')$. In such cases, exploration of the environment is required to query the model. This is accomplished by algorithms such as SARSA and Q-learning [19]. For illustrative purposes we describe the operation of the latter. Q-learning find $Q^*(s,a)$ recursively using the 4-tuple (s,a,s',r), where s and s' are the states at time t and $t+1$, a is the action taken when in s and r is the immediate reward due to taking a at s. The updating rule is:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_a Q(s',a) - Q(s,a)\right] \tag{9}$$

where α is the learning rate.

As one might expect, these algorithms must balance the need to collect informative data by *exploring* novel action choices for the given state space with the need to control the process well enough by *exploiting* the available knowledge acquired so far. This is known as the *exploration vs. exploitation dilemma* [14].

As it was stated in section I, in order not to deal with the discretization of the space of states, it is necessary to consider a continuous-state approach to the algorithm of reinforcement learning. This is done in the following subsection.

B. Continuous-State Reinforcement Learning

At this point we change our attention to continuous-state reinforcement learning. Reinforcement learning in continuous state-space demands function approximation to allow continuous states and actions without discretization. We can no longer rely on a tabular representation of the Q-function since this method is impractical for large (or potentially infinite) state and action spaces. In this new framework, the exact representation of $Q^\pi(s,a)$ is replaced by a parametric function approximator $\hat{Q}^\pi(s,a)$ [13].

A common parametrization is given by a linear combination of l basis functions [16], [18]:

$$\begin{aligned} \hat{Q}^\pi(s,a) &= \sum_{k=1}^{l} \phi_k(s,a)\, w_k \\ &= \begin{bmatrix}\phi_1(s,a) & \cdots & \phi_l(s,a)\end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_l \end{bmatrix} \\ &= \boldsymbol{\phi}^T(s,a)\cdot \mathbf{w} \end{aligned} \tag{10}$$

The basis functions are fixed and, in general, nonlinear functions of s and a. A common basis scheme is the radial basis function (RBF). One advantage of radial basis functions is that they only generalize locally i.e., changes in one area of the state space do not affect the entire state space [14].

One must select the parameters $\mathbf{w}$ so that $\hat{Q}^\pi$ consists on a good approximation of Q^π. As derived in [13], [20], one way to find a good approximation is to force the approximate Q-function to be a fixed point under Bellman operator. First we write (4) in a matrix format:

$$\mathbf{Q}^\pi = \boldsymbol{\mathcal{R}} + \gamma \mathbf{P}\mathbf{Q}^\pi \tag{11}$$

where $\mathbf{Q}^\pi$ and $\boldsymbol{\mathcal{R}}$ are vectors of size $|\mathcal{S}|\,|\mathcal{A}|$, and $\mathbf{P}$ is a stochastic matrix of size $|\mathcal{S}|\,|\mathcal{A}| \times |\mathcal{S}|\,|\mathcal{A}|$ that contains the transition model of the process.

To find an approximation for the Q-function, we start with a projected form of the Bellman equation 4 and the matrix form (11)

$$T_\pi \hat{\mathbf{Q}}^\pi \approx \hat{\mathbf{Q}}^\pi \tag{12}$$

In a way similar to temporal difference learning algorithms [15], if a finite set of L samples (s_i, a_i, r_i, s_i'), $i = 1, 2, \ldots, L$, along with the policy π is provided, then we have all the information needed to implicitly find $\mathbf{P}$ in (11) and solve (12). Using this observation and plugging (10)–(11) into (12) and applying the definition provided in (5), (12) can be rewritten as [13]:

$$\mathbf{A}\mathbf{w} = \mathbf{b} \tag{13}$$

where

$$\begin{aligned} \mathbf{A} = \frac{1}{L}\sum_{i=1}^{L} &\left[\boldsymbol{\phi}(s_i,a_i)\,\boldsymbol{\phi}^T(s_i,a_i)\right] \\ &- \gamma\left[\boldsymbol{\phi}(s_i,a_i)\,\boldsymbol{\phi}^T(s_i',\pi(s_i'))\right] \end{aligned} \tag{14}$$

and

$$\mathbf{b} = \frac{1}{L}\sum_{i=1}^{L} \boldsymbol{\phi}(s_i,a_i)\, r_i \tag{15}$$

The matrices (14) and (15) can be update iteratively as the i-th sample is drawn. This is performed by calculating

$$\begin{aligned} \mathbf{A}_i &= \mathbf{A}_{i-1} + \boldsymbol{\phi}(s_i,a_i)\,\boldsymbol{\phi}^T(s_i,a_i) \\ &\quad - \gamma\boldsymbol{\phi}(s_i,a_i)\,\boldsymbol{\phi}^T(s_i',\pi(s_i')) \\ \mathbf{b}_i &= \mathbf{b}_{i-1} + \boldsymbol{\phi}(s_i,a_i)\, r_i \end{aligned} \tag{16}$$

The algorithm known as Leasts-Squares Temporal Difference Learning for the State-Action Value Function (LSTD-Q) [13] processes a batch of L samples using (16) and solves the linear system

$$\mathbf{A}_L \hat{\mathbf{w}} = \mathbf{c}_L \tag{17}$$

When the number of samples $L \to \infty$, we have $\mathbf{A}_L \to L\mathbf{A}$, $\mathbf{b}_L \to L\mathbf{b}$ and $\hat{\mathbf{w}} \to \mathbf{w}$. Substituting the solution of (17) in (10), we obtain an approximation for the Q-function of the current policy π. This is known as *policy evaluation.* The resulting values are used for a *policy improvement*, i.e., the search for the greedy policy as defined in (8). This procedure is repeated at the next iteration for a new batch of samples. This algorithm is called LSPI (Leasts-Squares Policy Iteration) [13].

IV. Proposed solution

A. The Algorithm

In spite of the fact that LSPI is considered the highest level of development for policy improvement [18], one of its main drawbacks is that it improves the policy only after it runs LSTD-Q on large bath of samples to obtain an accurate approximation for the Q-function, usually implying a quite large processing delay. On the other hand, one of the main objectives of reinforcement learning is to learn the environment and search for the optimal policy in an on-line fashion [21], and not by processing batches of information. On that ground we introduce a modified version of LSPI to evaluate the current policy using an adaptive ϵ-greedy exploration strategy to improve the policy [19]. With this modification, the policy improvement can be performed on-line.

The algorithm works as detailed in Algorithm 1. It is important to observe that step 5, which is not present in the original algorithm, implements the ϵ-greedy strategy to deal with the exploration vs. exploitation dilemma, and step 10 searches for the greedy policy.

Since a great amount of exploration is usually required, ϵ_t should no approach 0 too fast. Moreover, due to the variability of the wireless channel, it is interesting to allow a certain degree of exploration to keep track of possible changes in the policy. We consider the selection of ϵ_t according to:

$$\epsilon_t = \max\left(\epsilon_f, \epsilon_i^{\tau t}\right), \tag{18}$$

where $\epsilon_f < \epsilon_i \in [0;1]$, ϵ_i is a decay factor close to unity and τ is a constant to be chosen. On the first iterations of the algorithm, values of ϵ_t close to ϵ_i are selected – large values of ϵ_i lead to a more aggressive (random) exploration. As time advances, ϵ_t decays to values closer to ϵ_f and the exploitation is more aggressive. According to (18), choosing $\epsilon_f \neq 0$ always guarantees a certain amount of exploration. It is important to point out that Algorithm 1 differs from the version presented in [22]. Our approach is able to guarantee some degree of exploration even after the convergence of the algorithm, a desirable feature in systems with high variability ans temporal changes such as the wireless channel. Moreover, unlike [22], every new data is used to update the Q-function, accelerating the convergence of the method.

Algorithm 1 *Modified LSPI*

1. The current policy π is initialized randomly
2. The matrix $\mathbf{A}$ and the vector $\mathbf{b}$ in (13) are initialized with
   $\mathbf{A}_{-1} = \delta \mathbf{I}_{l\times l}$
   $\mathbf{b}_{-1} = \mathbf{0}_{l\times 1}$
   where δ is a small constant of order 10^{-6}, $\mathbf{I}_{l\times l}$ is the $l \times l$ identity matrix and $\mathbf{0}_{l\times 1}$ is the $l \times 1$ null vector
3. For $t \geq 0$:
   4. The agent senses the current state s_t
   5. A random action a_t is taken with probability ϵ_t, and the greedy action is taken with probability $1 - \epsilon_t$.
   6. As a result of the action, the environment might make a transition to state s_{t+1} and it generates a reward r_t
   7. Calculate $\Delta = \boldsymbol{\phi}(s_t, a_t)\boldsymbol{\phi}^T(s_t, a_t) - \gamma\boldsymbol{\phi}(s_t, a_t)\boldsymbol{\phi}^T(s_t, \pi(s_{t+1}))$
   8. $\mathbf{A}_t = \mathbf{A}_{t-1} + \Delta$
   9. $\mathbf{b}_t = \mathbf{b}_{t-1} + \boldsymbol{\phi}(s_t, a_t) r_t$
   10. $\hat{\mathbf{w}} = \mathbf{A}_t^{-1}\mathbf{b}_t$
   11. Improve the policy using $\pi(s) = \max_{a\in\mathcal{A}} \boldsymbol{\phi}^T(s, a)\hat{\mathbf{w}}$
12. End

B. Actions, States and Rewards

As mentioned in Section III, the set $\mathcal{A}$ of actions consists on the admissible combinations between modulation order and coding rate. The environment state is determined by the received SNR averaged over all subcarriers [2], which varies within a continuum of real values.

The considered reward function R is defined as the throughput achieved when taking action a when the environment is at state s, and it is given by

$$R(s, a, s') = \log_2(M_a)\,\rho_a\,[1 - PER(s, a)] \tag{19}$$

where M_a is the modulation order of action a, ρ_a is the coding rate of action a and $PER(s, a)$ is the packet error rate of the action a over channel state s. Since a CRC field is attached to every packet, the receiver can identify the packets that are received in error and the PER can be estimated directly through system measurements in a similar way to the one presented in [23] or [24]. This information is then used as feedback to the transmitter adjust the most compatible modulation and coding scheme (in terms of minimizing the PER).

C. Complexity

We briefly describe the complexity of the proposed framework, considering the number of complex multiplications as a complexity metric. From Algorithm 1, the most expensive operation involves a complexity of $\mathcal{O}\left(l^3\right)$ due to the linear system that must be solved (using naive approaches). The inner products between the basis functions exhibits computational

TABLE I: WINNER's SCM Parameters.

Parameters	Value
Carrier frequency	2.0 GHz
Mobile speed	10.8 m/s
Number of antennas at Base Station	1
Number of antennas at Mobile Station	1
Scenario	Suburban Macro
Number of paths	19

TABLE II: Modulation and Coding Schemes

Scheme Number (Action m)	Modulation	Code Rate
1	QPSK	1/2
2	QPSK	3/4
3	16QAM	1/2
4	16QAM	3/4
5	64QAM	2/3
6	64QAM	3/4

complexity of $\mathcal{O}(l)$. Hence the computational complexity of the algorithm is directly related to the dimension of the basis and it is most influenced by the resolution of the linear system in (13).

V. Simulation and Results

The performance of the proposed link adaptation scheme using a continuous-space reinforcement learning approach was evaluated through simulations and compared with the performance of look-up tables under several scenarios.

A. System Parameters

For simulation purposes, the transmission aspects are roughly based on those found in 3GPP-LTE standard. The transmission is performed in a 10 MHz bandwidth. The system operates in a frequency-division duplexing fashion (FDD), in which a radio frame is 10ms long and contains 10 subframes of 1ms. Each subframe is divided into 2 slots, each of which carrying 6 OFDM symbols. The subcarrier spacing is fixed at 15 kHz, and the cyclic prefix length was chosen to be 1/16 of the OFDM symbol duration (approximately 4.6 μs). The transmission is performed on basis of resource blocks, defined as 6 OFDM symbols in the time domain and 12 subcarriers in the frequency domain. The transmitter and the receiver are assumed to have single antennas. The set of allowable combinations of modulation and coding given in Table II. The forward error correction (FEC) is implemented through convolutional coding with the coding rates of 1/2, 2/3 or 3/4. The encoder consists of 1/2 rate coder with generators $[133, 171]$ (in octal), and subsequent puncturing process to obtain 2/3 or 3/4 rate.

In order to perform more realistic simulations, a time-varying multipath channel has been considered. The chosen channel model is the Spatial Channel Model (SCM), which generates channel coefficients based on 3GPP channel model specifications [25], as implemented by the scripts provided by WINNER SCM [26]. The parameters values are detailed in Table I and they were used in all simulations, unless indicated otherwise.

B. Look-up Tables

The technique known as RawBER mapping [2] was used to generate the look-up tables used for AMC link adaptation. In RawBER mapping, the LQM is found by averaging over all the probability of uncoded bit errors at each subcarrier. The link between RawBER and PER is a regression generated by simulations in the AWGN channel, which can be prepared beforehand [27] [28]. The SNR thresholds were defined using a PER constraint of 10%. For each simulation, it is necessary to fix the packet size and the channel model.

One main disadvantage of look-up tables, besides the large amount of memory and simulation time, it is the fact that the performance of the system depends also on the statistical behavior of the interference [29] and the Gaussian assumption (the interference and Gaussian noise can be modeled as having a single Gaussian distribution) may not hold [12]. Clearly it is impractical to generate data to predict all possible situations. This same observation is valid for supervised-learning-based approaches.

In a practical situation, the SNR thresholds are adjusted by hand using long-term data collected from the radio interface [30]. This approach requires not only some expertise from the operator but it also does not guarantee to maximize the throughput since a lot of different scenarios are taken into account to obtain reasonable values to be used as thresholds. This might lead to too optimistic or too pessimistic modulation and coding schemes selection.

C. Reinforcement Learning Approach

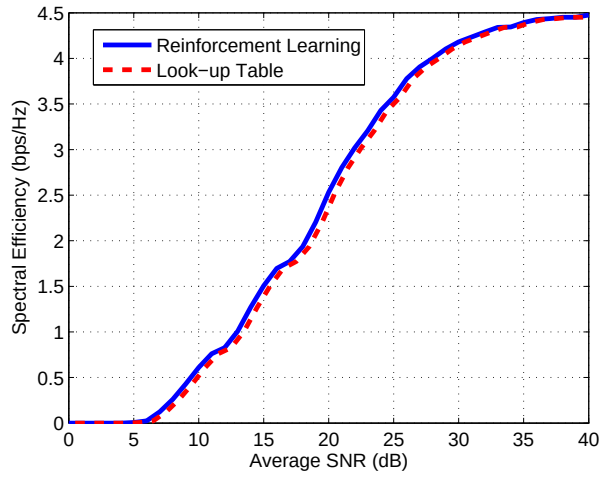
We applied the modified LSPI in a set of $l = 5$ basis functions for each of the 6 actions to approximate the value function. The basis were given by a constant term and 4 radial basis functions. For an action a,

$$\boldsymbol{\phi}(s,a) = \begin{bmatrix} 1 \\ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(s-\mu_1)^2}{2\sigma^2}} \\ \vdots \\ \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(s-\mu_4)^2}{2\sigma^2}} \end{bmatrix} \tag{20}$$

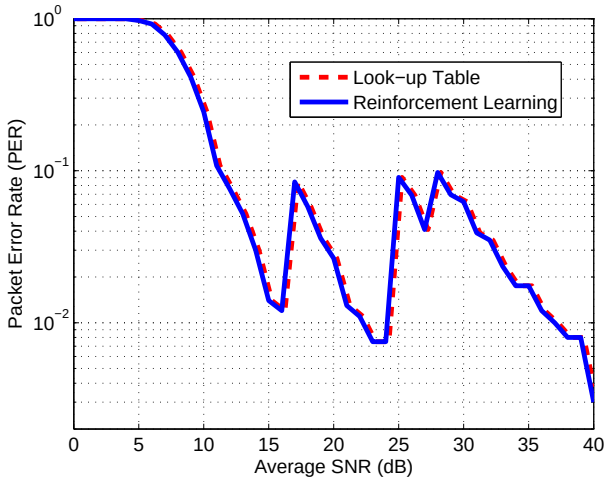
where the location parameters $\mu_1, \ldots, \mu_4$ are the centroids of the radial functions, equally spaced over the support $0 \leq SNR \leq 40$. The square scale parameter was chosen as $\sigma^2 = 2$ [13]. The set of possible actions are the $m = 6$ modulation and coding combinations given in Table II. The discount factor was set in $\gamma = 0.65$. To allow the exploration, we have set $\epsilon_f = 0.05$, $\epsilon_i = 0.95$ and $\tau = 0.01$. These choices will be justified latter.

D. Results

Fig. 2 shows the average spectral efficiency and packet error rate as a function of the SNR. Since the reinforcement learning approach was applied under the same circumstances that the



(a) Spectral Efficiency



(b) Packet Error Rate

Fig. 2: Average spectral efficiency and packet error rate of the look-up table and the reinforcement learning technique under the suburban macrocell scenario.

look-up table was obtained and uses the same link quality metric, they perform exactly the same. The main difference is that the reinforcement learning technique operates on-line and there is no need of an expert (teacher) or extensive simulations over different scenarios. The best modulation and coding scheme is selected by a non exhaustive trial and error procedure, requiring little programming effort for system training.

Fig. 3 and Fig. 4 consider the effect of some tunning parameters on convergence behavior of the algorithm. The mean square error (MSE) was calculated considering the throughput difference observed between the current improved policy (after the transmission of a given frame) and the optimal modulation and coding for a given SNR. This result was averaged over all the observed states. As shown in Fig. 3, the higher the discount factor, the faster the convergence at a cost of a higher MSE. As expected, a low value for the discount factor implies a myopic behavior since it values more

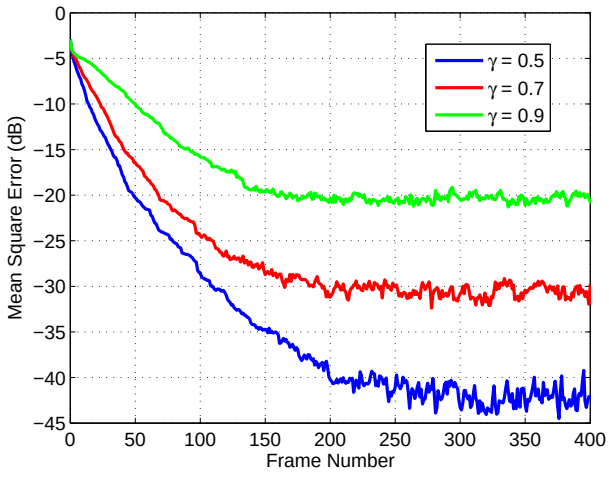
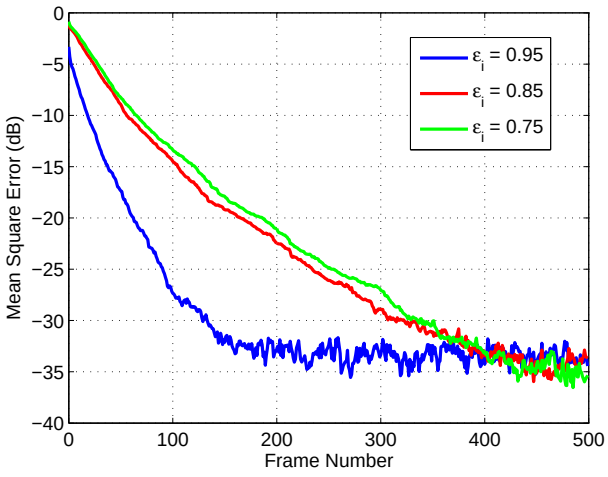


Fig. 3: Influence of the discount factor γ on the convergence of the RL algorithm for $\epsilon_i = 0.95$ and $\epsilon_f = 0.05$.

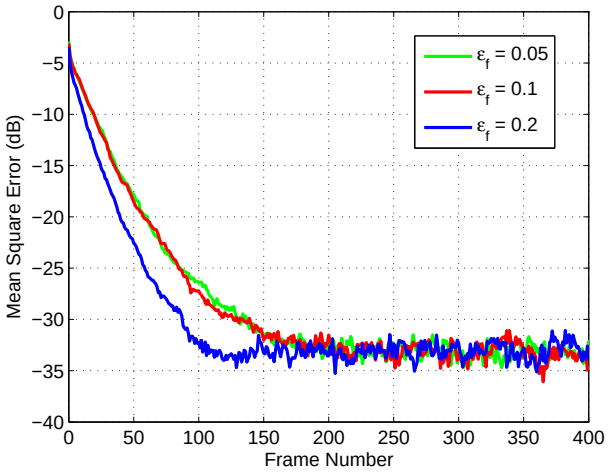
the immediate rewards and according to (16), the update steps of the matrices are smaller, justifying the larger number of frames to converge.

Fig. 4 shows the effect of the tunning values of the ϵ-greedy exploration strategy on the convergence of the reinforcement learning approach. As one can observe in Fig. 4(a), it is interesting to start the algorithm with an aggressive exploration strategy. Performing this way, the algorithm can learn faster what actions are suitable for each one of the states, implying in a faster convergence. As shown in Fig. 4(b), there is not a significant difference in the behavior of the algorithm for the values of ϵ_f. A larger value of ϵ_f is only appealing in a high variability scenario, where tracking capabilities are desirable (at the cost of not properly exploiting the optimal policy). This situation will be considered later in this paper.

Fig. 5 shows the average spectral efficiency and packet error rate of both approaches considering a scenario where colored interference is presented. This interference is composed of thermal noise (AWGN) and a second OFDM interference whose signal format is similar to the one found in 3GPP-LTE standard and whose power is three times higher than the white noise variance. Except for very low or very high values of signal to interference-plus-noise ratio (SINR), there is a gap of performance between the considered techniques. This difference can be larger than 1 bps/Hz depending on the SINR region. Here one of the problems of look-up tables (as well as other supervised learning approaches) is exposed: it can be very difficult to obtain the proper data through simulation in order to construct the tables or train the algorithms since they would depend on specific characteristics of the interfering signal. On the other hand, the proposed reinforcement learning scheme was able to learn from the environment, keeping the packet error rate under 10%. This fact is further confirmed in Fig. 6. It shows a second scenario of colored interference. This time, the interfering signal is composed of thermal noise and an OFDM signal whose power is eight times higher than the white noise variance. On the region of moderate values

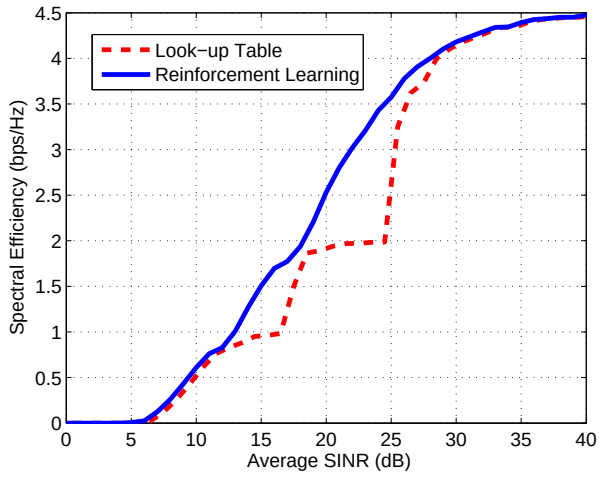


(a) Convergence behavior for different values of ϵ_i

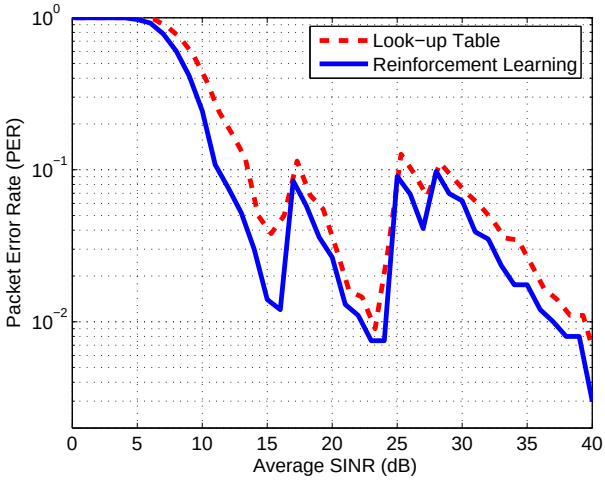


(b) Convergence behavior for different values of ϵ_f

Fig. 4: Influence of the initial (ϵ_i) and final values (ϵ_f) of ϵ-greedy exploration probabilities on the convergence of the RL algorithm keeping $\gamma = 0.65$.



(a) Spectral Efficiency



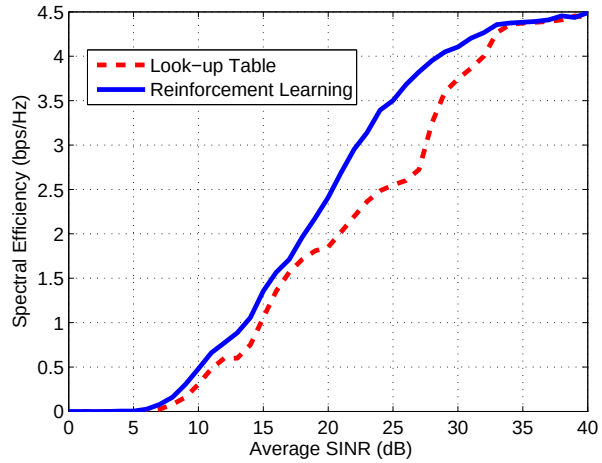
(b) Packet Error Rate

Fig. 5: Average spectral efficiency and packet error rate of the look-up table and the reinforcement learning technique under the suburban macrocell scenario and colored interference. The interference power is three times higher than the white noise variance.

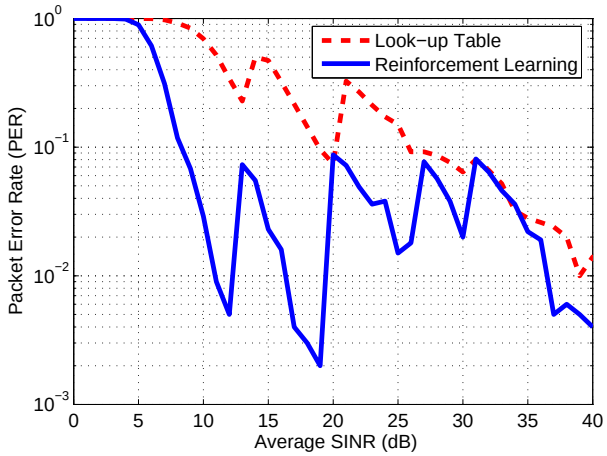
of SINR, we have not only a gap on the troughput but also higher values of packet error rate.

Next we investigate the possibility of applying the continuous-state reinforcement learning approach in situations where the channel characteristics vary over time [31]. Fig. 7(a) shows the tracking capabilities of our adaptation approach. We fixed the SINR value in 33 dB and the interference behavior is the same as described in the previous paragraphs. During the transmission of the first 300 frames, we consider the case where only additive white Gaussian noise is presented. From the frame 301 to the frame 500, colored interference is presented. Its power is three times higher than the noise power. At last, for the frames from 501 to 700, the interference power is eight times higher than the noise power. In the figure, the convergence time during the transitions is emphasized (50 frames and 30 frames, respectively). The convergence after the transition in the scenario is considerable faster than the initial convergence time since we do not recompute the policies from scratch, but we continue the learning using the previous Q-values. Fig. 7(b) shows how the values of PER and spectral efficiency vary in this situation. Although there is a slight increase in the PER, this behavior is due to the performance of the modulation and coding scheme on the given scenario, as shown by previous analysis. We remark that the value of $\epsilon_f = 0.05$ is able to provide enough exploration in this situation so that a new optimal policy can be obtained.

It may seen that the convergence intervals presented so far suggest that the proposed solution is not applicable to the time scale of the communication systems under study. It is important to remind that the duration of one LTE radio frame is 10 ms. The results show that convergence may be achieved within at most 5 seconds, which is very below the duration of a typical communication session. Moreover, the user equipment might periodically exchange control information with the base

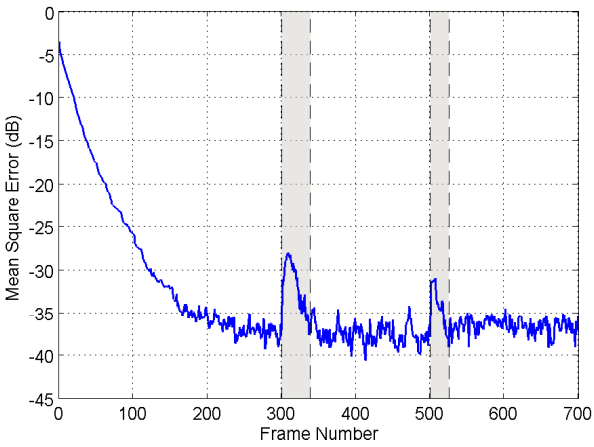


(a) Spectral Efficiency

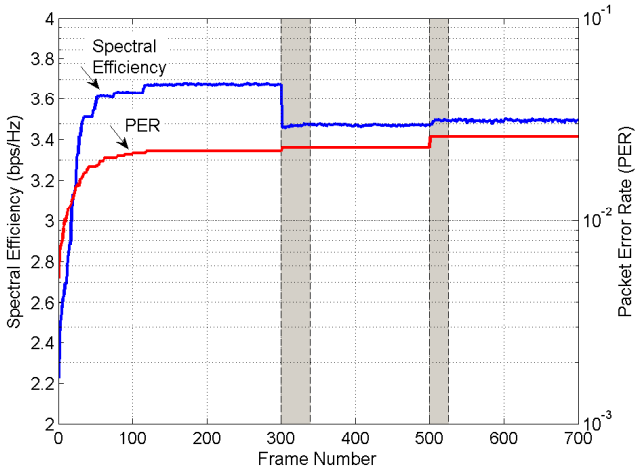


(b) Packet Error Rate

Fig. 6: Average spectral efficiency and packet error rate of the look-up table and the reinforcement learning technique under the suburban macrocell scenario and colored interference. The interference power in this case is eight times higher than the white noise variance.



(a) Convergence behavior



(b) Spectral efficiency and packet error rate

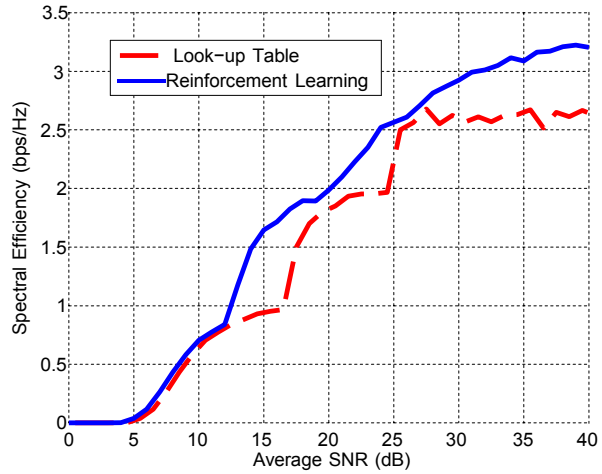
Fig. 7: Convergence behavior, spectral efficiency and packet error rate of the reinforcement learning technique under a time varying scenario for a SINR fixed at the value of 33 dB.

station and use this information to update the AMC mapping for a specific configuration set.

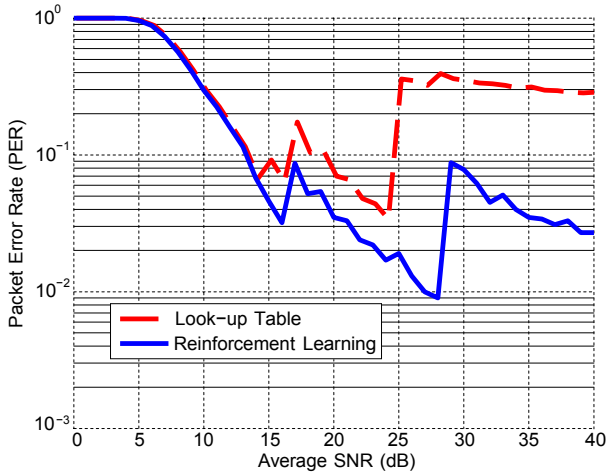
Finally we consider a scenario where RF imperfections (phase noise and I/Q imbalance) at the receiver side are introduced [32], both without compensation. More specifically we have phase noise energy of 0.013 rad^2, random phase imbalance of $3°$ and amplitude imbalance of 1.05. The results are shown in Fig. 8 As one can notice, an overall decrease of spectral efficiency of the system is observed, yet the look-up table exhibits poorer performance when compared to the reinforcement learning technique. The RL approach was able to learn that in a high SNR region the use of a high order modulation such as 64QAM would increase the packet error rate, decreasing the goodput. On the contrary, the look-up table has fixed thresholds, determined in advance in an off-line fashion, what does not allow it to adapt to the particularities of a given RF front end. It is also worthwhile to point out that in situations where RF imperfections are presented, the use of techniques based on supervised learning or the adjustment of a look-up table is almost impossible due to the great variety of scenarios and situation. Using the presented on-line reinforcement learning approach, this adaptation can be done for every receiver terminal.

VI. Conclusion

In this paper, we have presented a solution to the adaptive modulation and coding problem based on a machine learning framework using a continuous-state reinforcement learning algorithm. In this framework, the maximization of the spectral efficiency is treated as a Markov Process, where an unidimensional link quality metric (the mean SNR) was used to identify the state of the environment (the radio channel) and through interactions with the environment an optimal policy, i.e., an association between the states and the actions (given by the different combinations of modulation and coding) was found. The proposed scheme was shown appropriate for on-line and real time applications since it does not depend on any off-line



(a) Spectral Efficiency



(b) Packet Error Rate

Fig. 8: Average spectral efficiency and packet error rate of the look-up table and the reinforcement learning technique under the suburban macrocell scenario considering the presence of RF imperfections on the receiver side.

training phase. Moreover, it adapts to specific characteristics of the environment and the receiver. The look-up tables with fixed SNR thresholds tend to fail when applied in situations dissimilar from those of which they were obtained.

Issues that were not mentioned and are considered for further research is the presence of multiple antennas on the transmitter and/or receiver side, as well as the improvement of the dimensionality of the feature set and the use of nonuniform QAM modulation over different subcarriers.

REFERENCES

[1] Recommendation ITU-R M.1645, "Framework and overall objectives of the future development of IMT-2000 and systems beyond IMT-2000," Tech. Rep., 2003.

[2] S. Kant and T. L. Jensen, "Fast link adaptation for IEEE 802.11n," M. S. thesis, Aalborg University, Denmark, Feb. 2007.

[3] R. C. Daniels, C. M. Caramanis, and J. Robert W. Heath, "A Supervised Learning Approach to Adaptation in Practical MIMO-OFDM Wireless Systems," in *Proc. of IEEE Global Telecommunications Conference, 2008 - GLOBECOM 2008*, New Orleans, LO, Nov. 2008, pp. 1–5. doi: http://dx.doi.org/10.1109/GLOCOM.2008.ECP.878.

[4] ——, "Adaptation in Convolutionally Coded MIMO-OFDM Wireless Systems Through Supervised Learning and SNR Ordering," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 114–126, Jan. 2010 doi: http://dx.doi.org/10.1109/TVT.2009.2029693.

[5] R. Daniels and R. Heath, "An online learning framework for link adaptation in wireless networks," in *Information Theory and Applications Workshop, 2009*, Feb 2009, pp. 138–140. doi: http://dx.doi.org/10.1109/ITA.2009.5044935.

[6] ——, "Link adaptation in mimo-ofdm with non-uniform constellation selection over spatial streams through supervised learning," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, March 2010, pp. 3314–3317. doi: http://dx.doi.org/10.1109/ICASSP.2010.5496020.

[7] R. C. Daniels, C. M. Caramanis, and J. Robert W. Heath, "Online Adaptive Modulation and Coding with Support Vector Machines," in *Proc. of 2010 European Wireless Conference (EW)*, Lucca, Italy, Apr. 2010, pp. 718–724. doi: http://dx.doi.org/10.1109/EW.2010.5483527.

[8] H. Yigit and A. Kavak, "Adaptation using Neural Network in Frequency Selective MIMO-OFDM Systems," in *Proc. of 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC), 2010*, Modena, Italy, May 2010, pp. 390–394. doi: http://dx.doi.org/10.1109/ISWPC.2010.5483745.

[9] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. Cambridge, MA: MIT Press, 2010.

[10] T. Schenk, *RF Imperfections in High-rate Wireless Systems: Impact and Digital Compensation*. Dordrecht, the Netherlands: Springer, 2008.

[11] X. Hong, C.-X. Wang, and J. Thompson, "Interference Modeling of Cognitive Radio Networks," in *Vehicular Technology Conference, IEEE VTC Spring 2008*, Marina Bay, Singapore, May 2008, pp. 1851–1855. doi: http://dx.doi.org/10.1109/VETECS.2008.421.

[12] M. Aljuaid and H. Yanikomeroglu, "Investigating the Gaussian Convergence of the Distribution of the Aggregate Interference Power in Large Wireless Networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 9, pp. 4418–4424, Nov. 2010. doi: http://dx.doi.org/10.1109/TVT.2010.2067452.

[13] M. G. Lagoudakis and R. Parr, "Least-Squares Policy Iteration," *Journal of Machine Learning Research*, vol. 4, pp. 1107–1149, Dec. 2003.

[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[15] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, New Jersey: Wiley-Interscience, 2005.

[16] D. P. Bertsekas, *Dynamic Programming and Optimal Control - Vol. II*, 3rd ed. Cambridge, MA: Athena Scientific, 2007.

[17] C. Szepesvari, *Algorithms for Reinforcement Learning*. Morgan and Claypool Publishers, 2010.

[18] L. Busoniu, R. Babuska, B. de Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Boca Raton, FL: CRC Press, 2010.

[19] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 2009.

[20] M. G. Lagoudakis and R. Parr, "Model-Free Least-Squares Policy Iteration," in *Proc. of NIPS 2001: Neural Information Processing Systems: Natural and Synthetic*, Vancouver, BC, Dec. 2001, pp. 1547–1554.

[21] W. D. Smart, "Making Reinforcement Learning Work on Real Robots," Ph.D. dissertation, Brown University, Providence, Rhode Island, May 2002.

[22] L. Busoniu, D. Ernst, B. De Schutter, and R. Babuska, "Online least-squares policy iteration for reinforcement learning control," in *American Control Conference (ACC), 2010*, June 2010, pp. 486–491. doi: http://dx.doi.org/10.1109/ACC.2010.5530856.

[23] M. A. Haleem and R. Chandramouli, "Adaptive Stochastic Iterative Rate Selection for Wireless Channels," *IEEE Commun. Lett.*, vol. 8, no. 5, pp. 292–294, May 2004. doi: http://dx.doi.org/10.1109/LCOMM.2004.827389.

[24] A. Misra, V. Krishnamurthy, and R. Schober, "Stochastic Learning Algorithms for Adaptive Modulation," in *Proc. of IEEE 6th Workshop on Signal Processing Advances in Wireless Communications - SPAWC 2005*, New York, NY, Jun. 2005, pp. 756–760 doi: http://dx.doi.org/10.1109/SPAWC.2005.1506241.

[25] 3GPP, "3GPP TR 25.996 V8.5.0 - Spatial Channel Model for Multiple Input Multiple Output (MIMO) Simulations (Release 6)," Third Generation Partnership Project, Tech. Rep., Sep. 2003.
[26] J. Salo, G. Del Galdo, J. Salmi, P. Kysti, M. Milojevic, D. Laselva, and C. Schneider. (2005, Jan.) MATLAB implementation of the 3GPP Spatial Channel Model (3GPP TR 25.996). [Online]. Available: http://www.tkk.fi/Units/Radio/scm/
[27] M. Lamarca and F. Rey, "Indicators for PER Prediction in Wireless Systems: A Comparative Study," in *Vehicular Technology Conference, VTC Spring*, vol. 2, no. 30, Stockholm, Sweden, May 2005, pp. 792–796. doi: http://dx.doi.org/10.1109/VETECS.2005.1543413.
[28] F. Peng and J. Zhang, "Adaptive Modulation and Coding for IEEE 802.11n," in *IEEE Wireless Communications and Networking Conference*, Kowloon, Hong kong, Mar. 2007, pp. 656–661. doi: http://dx.doi.org/10.1109/WCNC.2007.126.
[29] S. Plass, A. Dammann, S. Kaiser, and K. Fazel, *Multi-Carrier Spread Spectrum 2007: Proceedings from the 6th International Workshop on Multi-Carrier Spread Spectrum*. Herrsching, Germany: Springer, 2007.
[30] S. C. Yang, *OFDMA System Analysis and Design*. Norwood, MA: Artech House, 2010.
[31] B. C. Csáji and L. Monostori, "Value Function Based Reinforcement Learning in Changing Markovian Environments," *Journal of Machine Learning Research*, vol. 9, pp. 1679–1709, Jun. 2008.
[32] G. Fettweis, M. Loehning, D. Petrovic, M. Windisch, P. Zillmann, and W. Rave, "Dirty RF: A New Paradigm," *International Journal of Wireless Information Networks*, vol. 14, no. 2, pp. 133–148, Jun. 2007. doi: http://dx.doi.org/10.1109/PIMRC.2005.1651863.

João P. Leite holds a Ph. D. in Electrical Engineering (2014). He received his B.S. degree in Communication Networks Engineering in 2007 and his M.Sc. degree in Electrical Engineering in 2009, both from the University of Brasilia, Brazil. Currently he is Professor of Electrical Engineering at University of Brasilia. His primary research interests include adaptive digital signal processing, simulation of communication systems, telecommunication software design, measurement of mobile radio channel and the application of machine learning techniques in wireless communication systems.

Robson D. Vieira received the B.S. degree in Electrical Engineering from the Federal University of Goiás, Brazil, in 1999, M.Sc. and the Ph.D. degree in Electrical Engineering from the Catholic University of Rio de Janeiro, Brazil, in 2001 and 2005 respectively. From March 1999 to March 2005, he worked as a researcher assistant at Centro de Estudos em Telecomunicações (CETUC), Rio de Janeiro, Brazil. Since 2005 he has worked at Nokia Technology Institute as a Telecommunication Specialist Researcher, where he is mainly working with White Space and Cognitive radio concepts and he is giving some support to GERAN standardization in the topic M2M. In the past he worked with WiMAX/LTE simulations and gave support for standardization activities on IEEE 802.16m. He is also Associated Researcher of University of Brasilia. His research interests include measurement and modeling of mobile radio channel, MIMO communication systems and cognitive radio.

Paulo H. P. de Carvalho received his B.S. in Electrical Engineering from University of Brasilia, Brazil, in 1988, his D.E.A. en Communications Optiques et Microondes from University of Limoges, France, in 1989, and his Ph.D. at Communications Optiques et Microondes from University of Limoges, France, in 1993. Currently he is Professor of Electrical Engineering at University of Brasilia. He has experience in the area of Electronic Engineering, with emphasis on telecommunication systems and microwave circuit simulation.