Deep Reinforcement Learning Based Adaptive Modulation with Outdated CSI

Shima Mashhadi, Niyousha Ghiasi, Shahrokh Farahmand, and S. Mohammad Razavizadeh

***Abstract*—The problem of adaptive modulation with outdated channel state information (CSI) is considered. Best existing approach to tackle this problem relies on using a (non-)linear auto-regressive moving average (ARMA) model to predict current CSI from outdated values. This approach is valid only if the wireless channel variations over time behave in a linear or smooth enough nonlinear fashion, which is not necessarily the case. We propose a deep reinforcement learning based adaptive modulation (DRL-AM) approach that can handle this limitation. While DRL-AM is more complex than (non-)linear AR(MA), it performs significantly better as corroborated via numerical results on real channel measurements. Furthermore, compared to capacity-achieving codes, complexity is moved from receiver to transmitter making this approach suitable for receiving nodes with limited resources such as internet of things (IoT) devices.**

***Index Terms*—Deep reinforcement learning, adaptive modulation, auto-regressive model, Wiener filter, outdated CSI**

I. Introduction

Adaptive modulation (AM) is a simple yet effective technique to obtain near-capacity rates for receiving devices that are limited in memory and processing power, e.g., internet of things (IoT) nodes, and thus can not decode complicated capacity-achieving codes [1]. Usually AM is chosen to either maximize throughput or rate subject to given bit error rate (BER) constraints. When a current estimate of channel is available, instantaneous and Ergodic throughput maximization amounts to an exhaustive search over modulation orders. Instantaneous BER constraints can easily be incorporated. A challenge arises in maximizing rate or throughput subject to average BER constraints. In this case, one needs to know the channel state information (CSI) distribution, and selecting the appropriate thresholds to switch modulations becomes a multi-dimensional exhaustive search problem which can become too complex even for offline processing.

Recently, this challenge has been tackled via deep reinforcement learning (DRL) in [2]. However, [2] assumes the distribution of CSI to be known in advance and perfect CSI to be available which limits the applicability of their algorithm. A simple RL for AM which removes the need for storing an offline calculated table is proposed by [3]. When current CSI is available but is not accurate, a robust RL for AM is proposed by [4]. Finally, AM design via DRL for heterogeneous networks (HetNets) in the presence of unknown multi-user interference is addressed by [5].

One main challenge in wireless communication systems exploiting CSI is the feedback delay. It takes a usually non-negligible delay to accurately estimate CSI at the receiver and fed it back to the transmitter. This delay can severely degrade the performance of AM schemes. Two remedies exist. The simple one is to just use outdated CSI to select the modulation. Second approach is to apply the linear minimum mean-square error (LMMSE) estimator to predict the current CSI from outdated measurements [6]. Given that for linear jointly Gaussian models LMMSE equals MMSE, the proposed estimator performs satisfactorily under these setting. However these linearity and Gaussian assumptions may not hold for realistic wireless communication channels.

Besides [4], neither of the aforementioned references consider the outdated CSI problem. While [2], [3] explicitly assume perfect CSI, [5] implicitly assumes so. Finally, [4] has considered this problem but assumes a low-quality estimate of current CSI is available which we do not assume. As our chief contribution, transmitter should perform AM with outdated CSI only and without any knowledge, whatsoever, of current CSI. Secondly, we apply a DRL method which is considerably more powerful than the proposed RL in [4]. Thirdly, the proposed DRL does not require labeled training data sets compared to supervised learning algorithms such as multilayer perceptron, long short-term memory, and recurrent neural network. Fourth, the proposed DRL is capable of detecting and exploiting nonlinear dependencies between CSI over time more accurately than both linear and nonlinear AR(MA) models for CSI variations [6], [7] as corroborated via numerical results.

The rest of this paper is organized as follows. Section II describes problem formulation. Section III reviews RL and DRL, then presents the proposed DRL-AM. Section IV provides numerical results and Section V concludes the paper.

II. Problem Formulation

Let us consider a single-input single-output (SISO) wireless communications channel. The frequency-flat channel changes over time and we represent the channel gain at time instant nT by h_n. Here, T is a single frame time consisting of F symbols and should be smaller than channel coherence time. Provided we know the channel gain at frame n, its capacity is given by

$$C_n = \log\left(1 + \frac{P_t|h_n|^2}{N_0}\right),$$

where P_t represents the fixed transmit power and N_0 denotes noise power. It is well-known that achieving capacity theoretically requires very long Gaussian codes [8]. A simple but practical way to approach capacity is by using adaptive

All authors are with the School of Electrical Engineering, Iran University of Science and Technology (IUST), Narmak, Tehran 16846-13114, Iran. emails:{sh_mashhadi,n_ghiasi}@elec.iust.ac.ir,{shahrokhf,smrazavi}@iust.ac.ir

modulation and coding (AMC) which offers a step-wise approximation of log function in capacity. Existing practical capacity-achieving codes still require a lot of processing power and long memories at the receiver side which is impractical for nodes with limited resources e.g., in IoT. Given that we target low-complexity receivers, we dispense with the adaptive coding part at a cost in achievable rate and focus on AM with uncoded data. To be more specific, suppose we can apply one of the K modulation schemes $\mathcal{M} := \{\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_K\}$ of corresponding constellation sizes $M_1, M_2, \cdots, M_K$. We define the throughput, in terms of bits per frame, at time n by

$$R_{h_n}(\mathcal{M}_i) = F \log_2(M_i)\left(1 - \mathrm{FER}_{h_n}(\mathcal{M}_i)\right). \qquad (1)$$

Here, FER is the frame error rate given by

$$\mathrm{FER}_{h_n}(\mathcal{M}_i) = 1 - \left(1 - \mathrm{SER}_{h_n}(\mathcal{M}_i)\right)^F, \qquad (2)$$

which assumes uncoded data as symbol errors occur independently. Symbol error rate (SER) for rectangular QAM of size M_i is given by [8, pp. 278]

$$\mathrm{SER}_{h_n}(\mathcal{M}_i) = 1 - \left(1 - 2\left(1 - \frac{1}{\sqrt{M_i}}\right) Q\left(\sqrt{\frac{3P_t|h_n|^2}{(M_i-1)N_0}}\right)\right)^2,$$

where Q-function is the integral of standard Gaussian tail. For every frame n, AM strives to maximize instantaneous throughput via optimum constellation selection

$$\mathcal{M}_n^* := \arg \max_{\mathcal{M}_i \in \mathcal{M}} R_{h_n}(\mathcal{M}_i). \qquad (3)$$

Since Ergodic throughput is the average of instantaneous throughput over channel gain pdf, optimizing instantaneous throughput also optimizes Ergodic throughput. The optimization problem (3) can be easily solved if the SNR of the current frame given by $P_t|h_n|^2/N_0$ is known to the transmitter. Since both P_t and N_0 are known in advance, optimum can be found via exhaustive search over modulation options provided h_n is known. Unfortunately, channel estimation is delay-prone meaning that by the time a new channel estimate is computed by the receiver and fed back to the transmitter, it has become outdated and true channel has changed.

Existing remedy is to use Jakes model [8, p. 809], whose auto-correlation can be well-approximated by the output of a linear auto-regressive (AR) system with white possibly Gaussian input noise. Hence, one can use an AR model of reasonable order to predict the current channel based on its available past values. A Wiener filter (WF) can be applied to estimate AR model coefficients [9]. This approach can be thought of as an iterative and adaptive version of LMMSE estimator in [6]. While theoretically sound and accurate, this approach falls apart when true channel variations are related in an unknown nonlinear fashion, where severe performance degradation can occur. Our proposed method aims to address this limitation. Accordingly, we propose a deep reinforcement learning (DRL) algorithm capable of learning the nonlinear dependencies between past and present channel values. As a characteristic of adaptive algorithms, it can also adapt to changes in channel statistics such as varying time dependencies. Compared to complicated coding schemes, complexity is shifted to BS transmitter from the IoT receiver. The Block diagram of the DRL-based adaptive modulation algorithm referred to as DRL-AM is shown in Fig. 1.

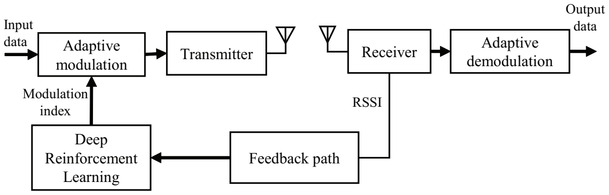

Fig. 1: Block diagram of the DRL-AM

III. DRL-Based Adaptive Modulation

First, we shortly review the RL and the DRL algorithms, then we propose a DRL-based modulation selection algorithm in order to maximize the instantaneous (and Ergodic) throughput.

A. Reinforcement Learning (RL)

To describe RL, definitions for five important elements are needed. They are expressed as follows [10].

1) State space S is the set of states s that are observed by an agent in the environment.
2) Action space A which is the set of actions a that can be taken by an agent in each state.
3) Immediate reward funcion $r(s,a)$ which is the immediate reward for taking action $a \in A$ in state $s \in S$.
4) Policy $\pi(s) \in A$ which is a mapping from the observed states to the actions that will be taken by an agent in those states.
5) State-action-value function $q_\pi(s,a)$ which is the long-term reward that is defined as the expected cumulative discounted reward in the future for the action $a \in A$ that is taken by an agent in the state $s \in S$ under policy π.

RL aims to select the optimal policy $\pi^*(s)$ for every state s that maximizes $q_\pi(s,a)$. Upon defining a discount factor $\eta \in [0,1]$, $q_\pi(s,a)$ is expressed as

$$q_\pi(s,a) = E\left\{\sum_{t=0}^{\infty} \eta^t r(s_t, a_t) | s_0 = s, a_0 = a\right\}, \qquad (4)$$

where expected value is taken over Markov transition probabilities $p_{s_t s_{t+1}}(a_t)$. Note that the policy $\pi(s)$ can be probabilities over the action set A in general or a single action $a \in A$ for simpler scenarios. Here, we look for deterministic, i.e., single action policies. The optimal policy per state is found by

$$\pi^*(s) = \arg \max_{a \in A} q_\pi(s,a). \qquad (5)$$

Due to unknown transition probabilities and partially observed states, an exact solution to (5) is impossible to obtain. Instead, a model-free algorithm called Q-learning has been proposed to approximately solve for the optimal policy [10]. In this algorithm the Q-function values are learned via trial and error and are updated as follows

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\Big[r(s,a) + \eta \max_{a' \in A} Q(s',a')\Big], \qquad (6)$$

where α is the learning rate and s' denotes the new state we move into after taking action a. To balance exploitation versus exploration, an ϵ-greedy algorithm can be utilized [10]. When the number of possible state-action pairs are very large or infinite, RL as derived above can not be applied because it will take too long to try every state-action pair. Thus, we need a complicated function with generalization capability that can map every unseen state to an optimal action. This function can be trained and made available via deep neural networks (DNN) leading to deep reinforcement learning (DRL) which will be discussed next.

B. Deep Reinforcement Learning

In DRL, the state-action-value function is approximated by $\tilde{Q}(s, a; \boldsymbol{\theta})$ where vector $\boldsymbol{\theta}$ is the weights of the DNN which mimics the true $Q(s, a)$ in some optimal sense. Besides the major advantage mentioned before, DRL doesn't need to store $Q(s, a)$ for each state and action pair and it only needs to store the DNN weights. To optimize, i.e., learn $\boldsymbol{\theta}$, definition of agent experience is needed, which is represented by $e =< s, a, r(s, a), s' >$. It consists of the state s, the action a taken in state s, the corresponding obtained reward $r(s, a)$ and the next state s'. These experiences are obtained via trial and error by the agent with the ϵ greedy method. Then, they are stored in the replay memory O and are later sampled uniformly to update $\boldsymbol{\theta}$. To ensure the stability of DRL, two NNs called the policy network (PN) with weight vector $\boldsymbol{\theta}$ and the target network (TN) with weight vector $\boldsymbol{\theta}^-$ are required. While TN generates new optimal policies to exploit, PN is trained by the experiences in O. Every T_{PN} iterations, TN becomes outdated and is replaced with the newly trained PN. The loss function that must be minimized in training PN is given by

$$L(\boldsymbol{\theta}) = E\left[\left(\text{Target Q(s,a)} - \tilde{Q}(s, a; \boldsymbol{\theta})\right)^2\right], \tag{7}$$

where

$$\text{Target Q(s,a)} = r(s, a) + \eta \ \arg \ \max_{a' \in A} \ \tilde{Q}(s', a'; \boldsymbol{\theta}^-). \tag{8}$$

For minimizing the loss function (7), stochastic gradient descent (SGD) is usually utilized.

C. DRL-AM Algorithm

We assume transmitter acts as an agent which wants to select an appropriate modulation order at each state which consists of outdated CSI. The main DRL-AM constituents are defined as follows.

1) State space at time n consists of received signal strengths (RSS) values for the previous τ transmitted frames measured by the receiver and fed back to the transmitter. In brief, the state in the current frame n is defined as
$$s(n) = \{\text{RSS}(n - \tau), \cdots, \text{RSS}(n - 1)\}, \tag{9}$$
where $\text{RSS}(n) := P_t |h_n|^2$.
2) Action space consists of all the available modulation orders denoted by
$$A = \{\mathcal{M}_1, \cdots, \mathcal{M}_K\}. \tag{10}$$
3) Immediate reward funcion is defined as the instantaneous throughput of the receiver for any given state $s(n)$ and action $\mathcal{M}_i$
$$r(s, a) = F \log_2 (M_i) \left(1 - \text{FER}_{h_n}(\mathcal{M}_i)\right). \tag{11}$$

The pseudocode of the proposed DRL-AM is provided in Algorithm 1.

Algorithm 1 DRL-Based Modulation Selection

```
Initialize replay memory $O$ to capacity $O_{max}$
Initialize the policy network with random weights $\theta$
Initialize the target network with weights $\theta^- = \theta$
for episode $\leftarrow 1$ to $I$ do
    for each $n$ do
        Select an action via $\epsilon$-greedy method, best action can be obtained via TN
        Take the selected action
        Feedback the measured RSS to the Tx by the Rx
        Observe the immediate reward and the next state
        Store the experience in replay memory $O$
        Sample random mini-batch from replay memory $O$
        Use SGD to update the weights of policy network
        After $T_{\text{PN}}$ frames, update the weights of the target network by the weights of policy network ($\boldsymbol{\theta}^- = \boldsymbol{\theta}$)
    end for
end for
Choose $a(n) = \arg \ \max_{a \in A} \ \tilde{Q}(s(n), a; \boldsymbol{\theta})$ afterwards
```

IV. Numerical Results

To emphasize the realistic nature of an indoor wireless channel, we have used the measured RSS data at a UCSB lab which is available online [11], [12]. As depicted in Fig. 2, position of the transmitter is highlighted with a triangular antenna while robot-mounted receiver moves along the various paths specified in Fig. 2.

It is assumed that the transmitter supports seven modulations of BPSK, 4-QAM, 8-QAM, 16-QAM, 32-QAM, 64-QAM and 128-QAM. The proposed DNN is composed of an input layer with τ neurons, which is related to elements of $s(n)$, five fully connected hidden layers with 64, 128, 256, 128, and 64 neurons, respectively, each with Relu activation function and an output layer with seven neurons, which relates to the seven available modulation orders. We set the value of the discount factor, η to 0.001, learning rate α to 0.003, and every 100 frames, TN is replaced with PN. Given the extensive processing power/time demand of optimal hyperparameter selection schemes [13], our hyperparameters were selected experimentally. We assume a human walking indoors at a speed of 5 kilometers per hour. Frame time T equals 36 milliseconds which amounts to a symbol rate of 28K symbols per second. Subsequently, the walking human moves about 5 cm within every frame transmission. Five algorithms are compared against. As benchmark, we consider an ideal case where current RSS is available without delay at the transmitter. Hence optimal modulation is selected via (3) with the knowledge of current CSI. Second Algorithm simply uses the outdated CSI

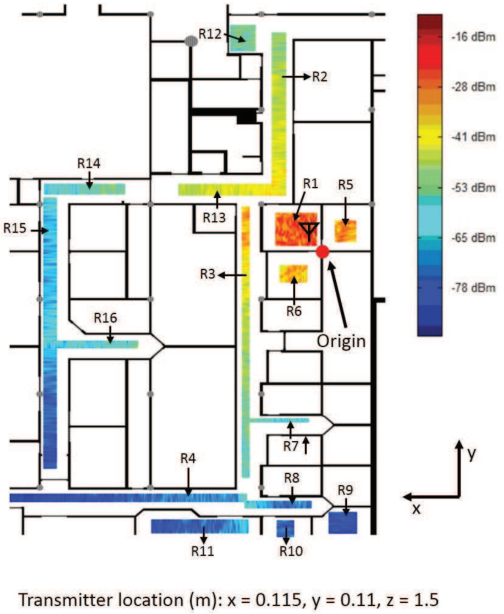

Fig. 2: Various paths for the receiving mobile node [12]

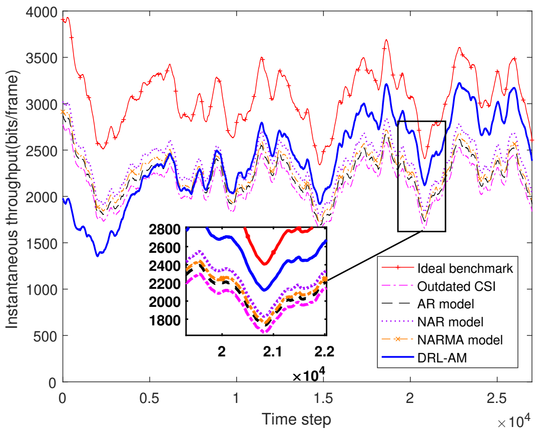

Fig. 3: Throughput versus time for deterministic route

to select the best modulation by placing h_{n-1} in (3) instead of h_n. Third algorithm uses a AR model of length τ to predict the current CSI from previous ones. A WF is exploited to optimize the AR model weights. Fourth algorithm uses a nonlinear autoregressive (NAR) model of length τ to predict the current CSI from previous ones [14]. Fifth algorithm uses a nonlinear auto-regressive moving average (NARMA) model of length τ for both AR and MA terms to predict the current CSI from previous ones [7]. We select τ equal to 10 for AR, NAR, NARMA, and DRL-AM.

A. *Deterministic Route*

Average performance on three different trajectories is investigated. The first route includes R14, R15, and R16, the second one includes R14, R15, R16, R4, and R8 and the third one includes R2, R13, R3, and R7 as in Fig. 2. The receiver moves in a deterministic fashion. Upon reaching an intersection, it decides a future direction randomly. Then, it moves along the selected path until reaching the next intersection. For this model, we set the batch size of experience samples to 512, O_{max} to 20000 and the initial value for ϵ to 1. At frame n, ϵ is computed as $\epsilon_n := 10^{-4} + (1 - 10^{-4}) \times e^{-\epsilon_{n-1} \times 3 \times 10^{-4}}$.

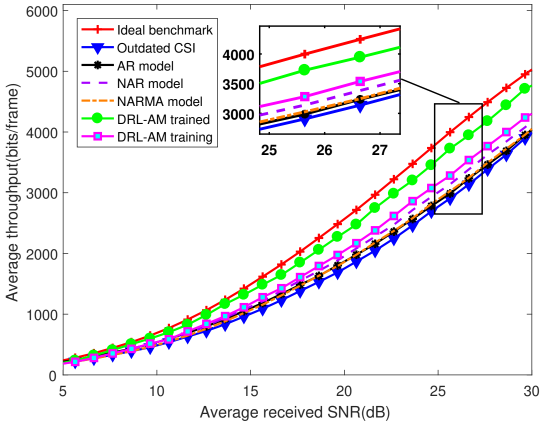

Fig. 4: Average throughput versus SNR for deterministic route

Fig. 3 depicts a 3000-wide moving average of instantaneous throughput in bits/frame over time. Average received SNR equals 22 dB. It is observed that while DRL-AM begins with a weak performance due to its initial random actions, it constantly improves over time. When enough time has elapsed, it significantly outperforms all four alternative algorithms. An improvement of about 300 to 400 bits per frame is observed at the enlarged segment. At certain points in time, DRL-AM gets very close to the ideal benchmark.

Fig. 4 plots average throughput versus received SNR. Evidently, as SNR increases, higher order modulations can be applied and utilizing the optimal decision produces a greater impact than low SNRs. Here, two curves are plotted for DRL-AM. The one labeled "trained" computes the average from time instant that ϵ reaches its minimum. That is the time where the algorithm has explored (or learned) enough and will only exploit afterwards. The curve labeled "training" computes the average from beginning to end thus taking into account the performance loss during training. For high enough SNR, the trained DRL-AM performs considerably better than the competition and close to the benchmark. It is noteworthy that even if we take the initial random actions of DRL-AM into account, DRL-AM still outperforms linear AR and NARMA methods by a visible margin. Minor improvement of linear AR model compared to outdated CSI reveals that channel variations are mostly nonlinear and hence a linear AR tracker can not effectively exploit the existing dependencies. A gap of about 700 bits per frame exists between the trained DRL-AM and linear AR model at an SNR of 25 dB. Compared to the trained DRL-AM, NAR and NARMA lack the flexibility to model the complex nonlinear channel variations. Hence, a gap of about 500 bits per frame exists between trained DRL-AM and NAR, which is the best performing alternative. We can conclude that once trained, DRL-AM outperforms both NAR and NARMA with a visible margin.

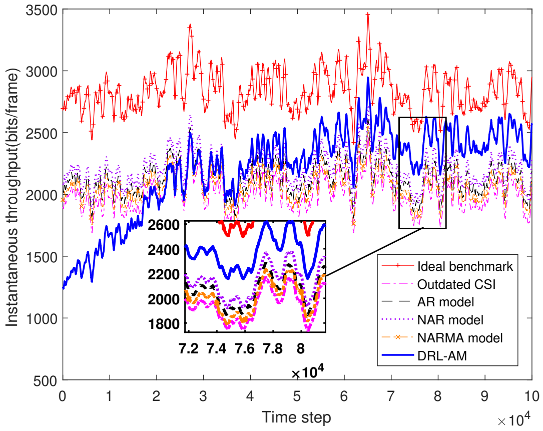

Fig. 5: Throughput versus time for random route

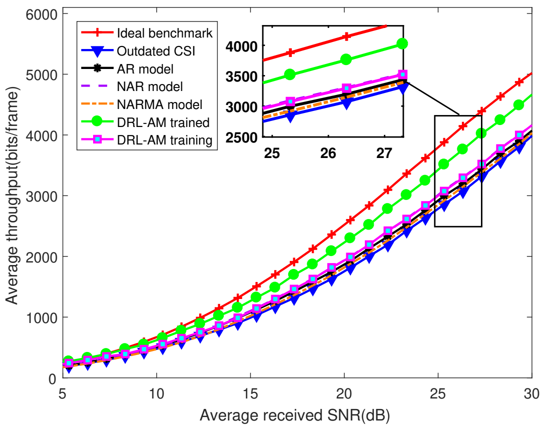

Fig. 6: Average throughput versus SNR for random route

B. Random Route

Average performance of all algorithms on three different trajectories is investigated. The routes are similar to the deterministic setting. However, the receiver movement pattern is different. In each trajectory, the receiver moves forward with probability 0.75 and changes direction with probability 0.25. For this model, we set the batch size of experience samples to 1000, the capacity of memory O_{max} to 50000 and the initial value for ϵ to 1. At frame n, ϵ is computed as $\epsilon_n := 10^{-4} + (1 - 10^{-4}) \times e^{-\epsilon_{n-1} \times 5 \times 10^{-5}}$. A 3000-wide moving average of the instantaneous throughput in bits per frame is plotted versus time in Fig. 5, which is the average results of three random routes. The average received SNR equals 22 dB. The average throughput versus SNR is plotted in Fig. 6. Same desirable behavior as in the deterministic route is observed. DRL-AM outperforms AR, NAR, NARMA, and outdated CSI and its performance comes close to the ideal benchmark.

Fig. 7 indicates the average throughput versus τ for the first random route at an average received SNR of 20 dB. As shown in Fig. 7, by increasing the value of τ from 1 to 4, the average throughput obtained by the DRL-AM method increases. By further increasing this value above 4, the average throughput becomes almost constant. It can be observed that for τ greater than four trained DRL-AM significantly outperforms the competitors. If the training episode is also included, DRL-AM performs similar to NAR and considerably better than NARMA, AR, and outdated CSI.

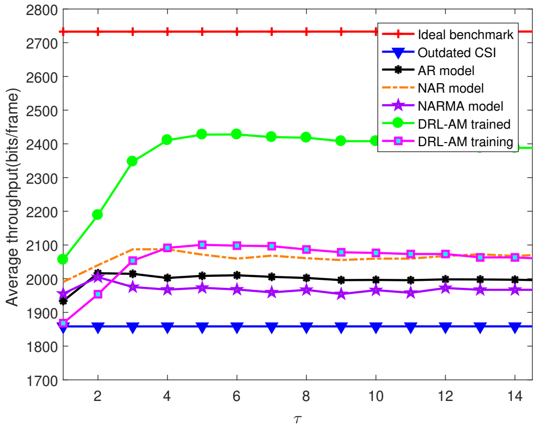

Fig. 7: Avg. throughput versus input size τ for random route

V. Conclusion

The problem of CSI feedback delay for adaptive modulation was investigated. The DRL-AM method was proposed to learn the best modulation order from past CSI measurements. It was revealed that the proposed approach outperforms algorithms based on AR, NAR, NARMA and outdated CSI.

References

[1] A. J. Goldsmith and Soon-Ghee Chua, "Variable-rate variable-power mqam for fading channels," *IEEE Transactions on Communications*, vol. 45, no. 10, pp. 1218–1230, 1997.

[2] D. Lee, Y. G. Sun, S. H. Kim, I. Sim, Y. M. Hwang, Y. Shin, D. I. Kim, and J. Y. Kim, "Dqn-based adaptive modulation scheme over wireless communication channels," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1289–1293, 2020.

[3] J. P. Leite, P. H. P. de Carvalho, and R. D. Vieira, "A flexible framework based on reinforcement learning for adaptive modulation and coding in ofdm wireless systems," in *Wireless Communications and Networking Conference (WCNC)*, 2012.

[4] R. Bruno, A. Masaracchia, and A. Passarella, "Robust adaptive modulation and coding (amc) selection in lte systems using reinforcement learning," in *Vehicular Technology Conference (VTC)*, pp. 1–6, 2014.

[5] L. Zhang, J. Tan, Y. Liang, G. Feng, and D. Niyato, "Deep reinforcement learning-based modulation and coding scheme selection in cognitive heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3281–3294, 2019.

[6] D. L. Goeckel, "Adaptive coding for time-varying channels using outdated fading estimates," *IEEE Transactions on Communications*, vol. 47, no. 6, pp. 844–855, 1999.

[7] M. Bhuyan, K. K. Sarma, and N. E. Mastorakis, "Nonlinear mobile link adaptation using modified flnn and channel sounder arrangement," *IEEE Access*, vol. 5, pp. 10390–10402, April 2017.

[8] J. G. Proakis, *Digital Communications*. McGraw-Hill, 4 ed., 2001.

[9] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1997.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, second ed., 2018.

[11] A. Gonzalez-Ruiz, A. Ghaffarkhah, and Y. Mostofi, "A Comprehensive Overview and Characterization of Wireless Channels for Networked Robotic and Control Systems," *Journal of Robotics*, vol. 5, p. 19, 2011.

[12] "Mostofi-Lab Wireless Channel Measurements (2009)." http://dx.doi.org/10.21229/M9159S.

[13] R. Liessner, J. Schmitt, A. Dietermann, and B. Bäker, "Hyperparameter optimization for deep reinforcement learning in vehicle energy management," in *Proc. of ICAART*, Feb 2019.

[14] L. Ljung, "System identification toolbox for use with matlab," Jan 2011.