Deep Learning Assisted Rate Adaptation in Spatial Modulation Links

Anxo Tato, Carlos Mosquera
atlanTTic Research Center, Universidade de Vigo, Galicia, Spain
Email: {anxotato, mosquera}@gts.uvigo.es

***Abstract*—The adaptation of Spatial Modulation based links to the channel conditions is challenged by the complicated dependence between performance (either error rate metrics or theoretically achievable rates) and the multiple antenna channel description. In this paper a coding rate selection mechanism is presented based on a carefully selected set of channel features and the proper training of a deep neural network, which all together can satisfy a given error rate bound.**

***Index Terms*—Deep Learning, Link adaptation, MIMO, Neural Network, Spatial Modulation.**

I. Introduction

Spatial Modulation (SM) is being considered for future 5G systems [1], since it can increase the spectral efficiency with respect to single antenna systems, with simpler hardware requirements as compared with other multi-antenna techniques, reducing the power consumption. Its most basic implementation activates only one antenna at a time, with the information encoded into the index of the active antenna and the transmit symbol. This seemingly simple transmission scheme poses some challenges when designing the receiver or computing the achievable information rates.

In particular, the achievable rate shows a significantly complex dependence with the channel coefficients, as opposed to more conventional Multiple Input Multiple Output (MIMO) techniques. This is a significant drawback when applying some sort of Adaptive Coding and Modulation (ACM) mechanism, generically known as link adaptation. This consists typically in varying the modulation order and/or the coding rate of the channel encoder to track the varying channel conditions. The ultimate goal is to adjust the transmitted bit rate to the information that the channel can support for a given bit error probability.

Link adaptation makes it necessary for the transmitter-receiver pair to estimate somehow the rate that can be supported by the channel. In most practical cases, the receiver computes some metric which is sent back to the transmitter end. This metric can be in the form of the average or effective Signal to Interference and Noise Ratio (SINR), or some Channel Quality Indicator (CQI) specifically suited to the set of Modulation and Coding Schemes (MCS) available to the transmitter [2].

The authors presented in [3] some analytical approximations to the integral expression of the mutual information in an SM link. Its use in adaptive settings is jeopardized by the error of the approximations, and also by the need to estimate the achievable rate of practical MCSs. With this motivation in mind, in this work we explore the use of Machine Learning (ML) tools to determine at each moment the appropriate MCS to transmit. The use of ML at the physical layer of communication systems is gaining momentum, as recent surveys illustrate [4]. In particular, Neural Networks (NN) have been successfully used for channel estimation and equalization [5], signal recognition and modulation classification [6], [7], detection in MIMO Generalized SM [8], and learning of physical layer parameters in Cognitive Radio [9], among others. In [10], NNs are applied to perform link adaptation in multicarrier systems. A publication more related to this work is [11], where the authors use a NN to make codebook selection in SM systems.

The current work uses a Multilayer Feedforward Neural Network (MFNN) of three hidden layers to operate the adaptive MCS decision process at the receive end. This is based on some specifically selected input features which can be easily obtained from the MIMO channel matrix, together with the Signal to Noise Ratio (SNR). Thus, we will show how the mapping from the SNR and the channel matrix to the optimum MCS can be carried out with an affordable computational cost and excellent performance.

The paper is organized as follows. Section II presents the system model. Then, in Section III the steps for implementing the coding rate selection in an adaptive SM link are explained. Sections IV and V present the simulation settings and the results, and finally the conclusions are exposed.

II. System model

We consider a Spatial Modulation system, where information is not only conveyed by the selection of a symbol from a constellation $\mathcal{S}$, but also by the antenna choosen for sending that symbol. The system model equation of an SM link with N_t transmit antennas and N_r receive antennas for a given discrete time instant is

$$\mathbf{y} = \sqrt{\gamma}\mathbf{H}\mathbf{x} + \mathbf{w} \tag{1}$$

This work was funded by the Xunta de Galicia (Secretaria Xeral de Universidades) under a predoctoral scholarship (cofunded by the European Social Fund), and it was partially funded by the Agencia Estatal de Investigación (Spain) and the European Regional Development Fund (ERDF) under project MYRADA (TEC2016-75103-C2-2-R).

where $\mathbf{y} \in \mathbb{C}^{N_r \times 1}$ is the received vector, γ the average Signal to Noise Ratio (SNR), $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ the channel matrix, $\mathbf{x} \in \mathbb{C}^{N_t \times 1}$ the transmitted signal and $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_r})$ the Additive White Gaussian (AWGN) noise vector. Due to the specific nature of SM (activation of only one antenna per symbol period), $\mathbf{x}$ has only one component different from zero (component l) and its value is $s \in \mathbb{C}$, a symbol taken from a constellation $\mathcal{S}$ with M symbols. Therefore, (1) can be also expressed as

$$\mathbf{y} = \sqrt{\gamma}\mathbf{h}_l s + \mathbf{w} \tag{2}$$

where $\mathbf{h}_l$ denotes the l column of $\mathbf{H}$, $l \in \{1, 2, \ldots, N_t\}$. We assume a unit power constraint, i.e., $\mathbb{E}\left[\mathbf{x}^H\mathbf{x}\right] = \mathbb{E}\left[|s|^2\right] = 1$.

We consider a packet based transmission where the transmitter encodes a group of information bits into a codeword or FECFRAME. The coding rate can be chosen from a predefined set of codes for link adaptation purposes. K different coding rates are considered, with respective rates $r_1 < r_2 < \ldots < r_K$. Data is more protected with r_1, the rate closest to zero, although it has also the lowest spectral efficiency. Conversely, if the channel is particularly benevolent information bits can be encoded with a rate r_K, which provides the highest spectral efficiency at the expense of a lower protection of the information bits. In general, the spectral efficiency attained with a given code of rate r is $\eta = r\log_2(N_t M)$, where M is the number of symbols of the constellation.

The SM link adaptation problem can be formulated as

$$\begin{aligned} \underset{r}{\text{maximize}} \quad & r\log_2(N_t M) \\ \text{subject to} \quad & r \in \{r_1, r_2, \ldots, r_K\} \\ & \text{BER}(\gamma;\ r, \mathbf{H}) \leq p_0. \end{aligned} \tag{3}$$

The Bit Error Rate (BER) depends on the selected coding rate r, the channel matrix $\mathbf{H}$ and the SNR γ, and must be kept below p_0. As opposed to more conventional MIMO techniques, the channel capacity is a very involved function of $\mathbf{H}$ [3], which can be calculated in real time by using a neural network [12]. It is also the case that, for a given coding rate, the influence of $\mathbf{H}$ is such that SNR values which differ in as much as 10 dB may be needed to achieve the same performance as can be seen in Fig. 3. As a consequence, coding rate selection requires a practical mechanism to track both the SNR and the channel matrix.

Fig. 1 shows the block diagram of an adaptive SM system where the transmitter can encode the information bits with a variable rate. The coding rate to be used, r, is calculated by the receiver from γ and $\mathbf{H}$ and then signalled back to the transmitter through a feedback channel. At the receiver soft detection is performed to send to the channel decoder the log likelihood ratios (LLRs) of each bit. The exact LLRs are calculated following [13]. The difficult task of selecting the appropriate coding rate solving (3) is addressed with the aid of a neural network properly trained. The NN is trained off-line, so that the on-line computations can be easily embedded into a receiver as we will detail later.

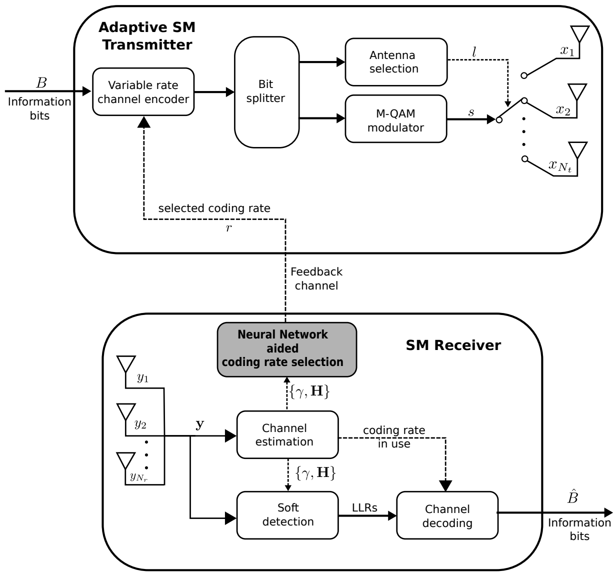


Fig. 1: Block diagram of an adaptive SM system with variable coding rate.

III. Coding rate selection

We detail in this section all the steps involved in the design of a receiver able to estimate the optimum coding rate as a function of the channel. This receiver will use machine learning principles, which will be illustrated later with some simulation results.

A. *Evaluation of the performance of the channel codes*

As the relation between the BER, channel matrix and SNR is quite involved for SM, system level simulations must be run to characterize the channel codes performance. For each of the available K codes, and for a large number of different channel matrices $\mathbf{H}$ following the same distribution as that expected in the practical deployment, the corresponding BER curve with respect to the SNR is obtained: $\text{BER} = f(\gamma;\ r, \mathbf{H})$. Fig. 2 shows, for illustration purposes, a collection of BER curves for several channel matrices. This first step is, by far, the most time consuming, taking even several days of execution time. Note that the remaining parameters and functions of the system remain fixed: number of antennas, family of codes, constellation, architecture of both transmitter and receiver, mapping of bits to SM symbols, and detection and decoding algorithms.

B. *Extraction of the SNR thresholds*

The channel codes BER curves are to be processed to extract the threshold (minimum) SNR to guarantee a given BER p_0 for each of the simulated channel matrices. Thus, a collection of stairwise plots, like those of Fig. 3, is produced. The selection of the appropriate coding rate for a given SNR can be now addressed with the support of a learning scheme, which can

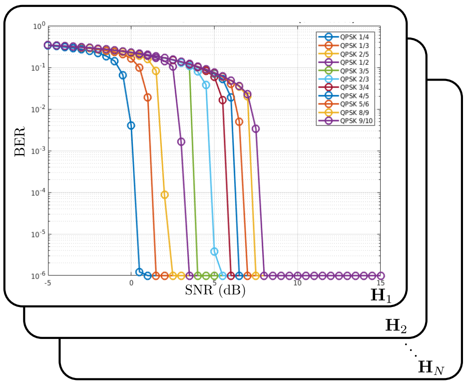


Fig. 2: The different channel codes performance must be evaluated for a large number of channel matrices.

generalize the performance curves to any arbitrary channel matrix $\mathbf{H}$.

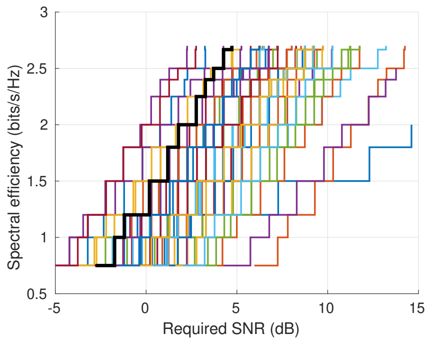


Fig. 3: The minimum required SNR to guarantee a given BER p_0 with each coding rate for a set of 20 different channel matrices.

C. Building the dataset for Machine Learning

The next step consists of building the dataset $\mathbb{X}$ used in the training and testing of the neural network. In [12] we have shown that it is possible to calculate the capacity constrained to a given constellation of an SM system with high accuracy by using a simple one-hidden layer feedforward neural network. However, this contrained capacity or mutual information is an upper bound of the achievable rate, and practical receiver implementations can be more or less far from this bound.

The selection of the neural network input features x has a paramount importance to obtain a good performance. The vector of features is obtained from the SNR γ and the channel matrix $\mathbf{H}$ by means of a transformation $\mathsf{x} = g(\gamma, \mathbf{H})$. In Table I of [12] several options for this function $g(\gamma, \mathbf{H})$ are given. There it is shown that the columns norms $\gamma \|\mathbf{h}_l\|^2$ and the angles between each pair of columns of $\mathbf{H}$ are a good selection for the input features.

The dataset $\mathbb{X} = \{(\mathsf{x}_i, \mathsf{y}_i), i = 1, 2, \ldots, m\}$ is made of all the input-output pairs. The vector x_i is the neural network input extracted from each pair of $(\gamma_i, \mathbf{H}_i)$ from our code performance evaluation. The scalar y_i, the neural network desired output or target variable, is a discrete real variable between 0 and 1 which represents the highest coding rate which can be used whilst meeting the BER p_0. Thus, $\mathsf{y}_i = r_k$ for coding rate r_k, and $\mathsf{y}_i = 0$ if no coding rate can guarantee the prescribed BER.

D. Neural network training

Once the Machine Learning dataset $\mathbb{X}$ is built, a neural network architecture has to be selected and trained to set its internal parameters $\boldsymbol{\theta}$. The goal is to provide good predictions $\hat{\mathsf{y}}$ from the input features x: $\hat{\mathsf{y}} = h(\mathsf{x}; \boldsymbol{\theta})$.

The design of the architecture of the network requires the selection of the number of hidden layers (*depth* of the model), of the number of neurons per layer (*width* of the model), the activation function and the specification of the output unit [14]. In this regard, although this is a classification problem, we have obtained better results with linear output units, so that the real-valued output of the network $\hat{\mathsf{y}}$ has to be quantified to obtain a value in the set $\{0, r_1, \ldots, r_K\}$ to yield a valid coding rate. Tangent hyperbolic is used as activation function for the hidden layer neurons. The network training is performed with the Levenberg-Marquardt (LM) backpropagation algorithm [15] to minimize the Mean Squared Error (MSE).

E. Performance evaluation

The NN performance evaluation is conducted on the testing portion of the dataset. The classification results are characterized by the confussion matrix, from which three key metrics can be extracted: (i) accuracy, defined as the ratio between the correct coding rates predictions and the total number of prediction made; (ii) ratio of underestimations, when a coding rate below the target is selected; (iii) ratio of overestimations, when higher rates than the target are chosen. The latter is the most critical since the selection of a coding rate beyond the receiver capabilities causes an outage, i.e., the frame cannot be correctly decoded, which results in a loss of throughput. On the other side, underestimation causes only a less efficient use of the link (a reduction of the average spectral efficiency), although the transmission is still successful. The selection of the coding rate is done with

$$r = Q\left(\hat{\mathsf{y}} - \Delta\right) = \arg\min_{r_k} \left|\hat{\mathsf{y}} - \Delta - r_k\right|, \qquad (4)$$

where Δ represents a positive back-off margin to be subtracted to the neural network output $\hat{\mathsf{y}}$ in order to reduce the outage probability.

F. Operation phase

Finally, during the system operation, receivers employ the neural network trained in the design phase to perform the coding rate selection. The transmitter receives through a feedback channel the choice made by the receiver and applies the corresponding coding rate for the next frame transmissions intended for that receiver. Although the neural network parameters $\boldsymbol{\theta}$ are fixed during the operation phase, the margin Δ introduced in equation (4) could be adapted in a similar way as that proposed in [16].

IV. System Simulation Setup

The previous procedure to perform coding rate selection with the assistance of Deep Learning was evaluated in an SM link with 2 transmit and 2 receive antennas. QPSK symbols are transmitted, mapped from bits which are encoded with the family of codes of the DVB-S2 standard [17]. These consist on the concatenation of a BCH (Bose-Chaudhuri-Hochquenghem) and a LDPC (Low Density Parity Check) code. There are $K = 9$ available coding rates, ranging from $1/4$ to $9/10$. The length of the FECFRAME is fixed to $64,000$ bits like in DVB-S2 standard. The number of information bits per codework is then variable and depends on the particular coding rate selected. The LDPC maximum number of iterations was set to 50. The mapping of bits to SM symbols follows Gray coding, by assigning the complementary bits sequences to the SM symbols which are more distant. We assume perfect Channel State Information (CSI) at the receiver, which includes the SNR and the channel matrix.

The system level simulations to obtain the performance of the codes are calculated for $N = 1,000$ different 2×2 channel matrices $\mathbf{H}$, generated by following a unit-variance Rayleigh distribution, i.e., $h_{ij} \sim \mathcal{CN}(0,1)$. For each matrix the average BER after the BCH decoding is calculated for 41 equispaced values of SNR between -5 and 15 dB. The average BER is calculated after simulating the transmission of 25 FECFRAMEs. The target BER for coding rate selection is $p_0 = 10^{-4}$. Table I sums up the main parameters of the system.

The NN input features x are calculated for each tuple $(\gamma, \mathbf{H})$ as

$$\mathsf{x} = g(\gamma, \mathbf{H}) = \left[\text{sort}\left(\gamma\|\mathbf{h}_1\|^2,\ \gamma\|\mathbf{h}_2\|^2\right),\quad \Theta_H,\quad \varphi\right]^t \tag{5}$$

following the results of [12], where the columns norms are sorted in ascending order. The parameters Θ_H and φ are two angles obtained from the scalar product between the two complex column vectors as

$$\mathbf{h}_1^H\mathbf{h}_2 = \|\mathbf{h}_1\| \cdot \|\mathbf{h}_2\| \cdot \cos\Theta_H \cdot e^{i\varphi}. \tag{6}$$

The so-called Hermitian angle Θ_H belongs to the interval $[0, \pi/2]$ whereas φ, named Kasner's pseudo-angle, takes values between $-\pi$ and π [18]. On the other hand, the neural network target output y is obtained as explained in the previous section for a target BER p_0 of 10^{-4}. The dataset is divided into three independent parts: 15% of the samples were reserved for the final test of the performance of the neural network. The remaining 70% and 15% were employed for training and validation of the neural network, respectively.

Several neural networks architectures were tested with a number of hidden layers between one and seven and a number of neurons per layer between 5 and 30. Each neural network was trained 20 different times with different sets of initial parameters. The training run for $1,000$ epochs, although it could be halted earlier if the network performance on the validation dataset stopped improving or remained the same for 6 epochs in a row. The default parameters of the *trainlm* function of Matlab® were used for the training.

TABLE I: System parameters

Paramter	Value
Transmit and receive antennas	$N_t = 2$, $N_r = 2$
Constellation	QPSK ($M = 4$)
Channel coding	DVB-S2 codes (BCH + LDPC)
Coding rate options	1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 5/6, 9/10
Target BER	$p_0 = 10^{-4}$
Channel matrices	1000 Rayleigh ditributed
SNR range	-5 to 15 dB (0.5 dB steps)

V. Simulation results

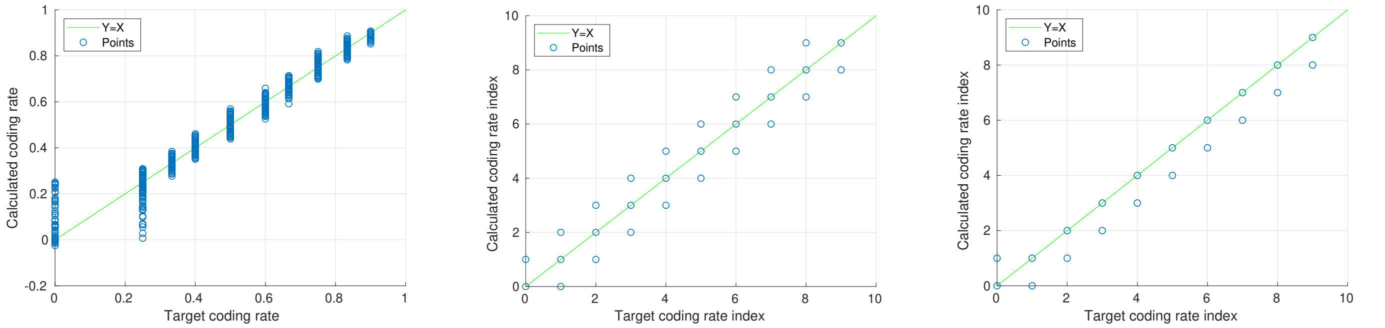
The results provided in this section correspond to a three hidden layers NN, with 20, 15 and 10 neurons per layer, respectively. Table II shows the raw classification performance obtained from the neural network output without applying any margin (for $\Delta = 0$) using equation (4). This table sums up the main information of the confusion matrix. The classification accuracy is typically better than 90.0% for almost all classes and the probability of selecting a coding rate higher than the target class (which causes an outage) is always below 9%. Whenever a wrong decision takes place, this is because the rate right above or below the optimal one is chosen instead. N/T corresponds to the class No-Transmission, i.e., no coding rate can guarantee the BER constraint.

The average accuracy, outage and underselection probabilities measured in the testing dataset are 96.2 %, 2.1 % and 1.7 %, respectively. Note that not all coding rates have to be chosen with the same probability, with N/T (no transmission) and 9/10 as the most likely in Table II. If we focus on the real output of the neural network ŷ, the Mean Square Error (MSE) is $4.75 \cdot 10^{-4}$ and the mean absolute error is 0.012.

Fig. 4 shows three regression plots with the target coding rate, the output of the neural network ŷ and the selected coding rate. Fig. 4a depicts how the output of the neural network is around the target values, except for the two lower classes, where there is more dispersion. However, as it can be seen in Fig. 4b, where the selected coding rate index is shown for a margin $\Delta = 0$, the miss classifications only happen with the two adjacent classes. Lastly, Fig. 4c shows the index of the target and the selected coding rate when a margin $\Delta = 0.03$ is employed in equation (4). With this value, no outage takes place at the expense of a reduction of the accuracy.

TABLE II: Classification performance (no margin is applied, $\Delta = 0$): reduced version of the confusion matrix.

Target coding rate	**N/T**	**1/4**	**1/3**	**2/5**	**1/2**	**3/5**	**2/3**	**3/4**	**5/6**	**9/10**
Accuracy (%)	98.7	95.9	91.9	94.2	93.8	91.8	94.4	89.3	89.7	99.5
Outage (%)	1.3	2.3	4.3	1.5	2.1	4.4	1.6	6.9	8.5	-
Underselection (%)	-	1.8	3.8	4.4	4.0	3.8	3.9	3.7	1.8	0.5



(a) Neural network ouput (b) Selected coding rate index, $\Delta = 0$ (c) Selected coding rate index, $\Delta = 0.03$

Fig. 4: Regression plot with target coding rate and calculated coding rate.

The average throughput η for each SNR value γ is computed using all the ML dataset as

$$\eta = \frac{1}{N}\sum_{i=1}^{N}\log_2(N_t M)\hat{r}_i(1-\epsilon_i) = \frac{1}{N}\sum_{i=1}^{N}3\hat{r}_i(1-\epsilon_i), \quad (7)$$

where $\hat{r}_i$ is the coding rate selected for the channel matrix i and ϵ_i is a binary variable which takes the value 1 in the case of an error event, i.e., when a coding rate higher than the target coding rate is selected, and which takes the value 0 when the correct coding rate or a lower one is selected. In this case, $N = 1,000$, since one thousand different channel matrices are simulated per each SNR value. With the previous definition of ϵ_i, the average outage probability per SNR is formulated as

$$\text{Average outage probability} = \frac{1}{N}\sum_{i=1}^{N}\epsilon_i. \quad (8)$$

For the average throughput and outage probability, the complete dataset $\mathbb{X}$ is used, and not only the part used in the neural network testing, in order to obtain smoother graphics without increasing the number of channel matrices.

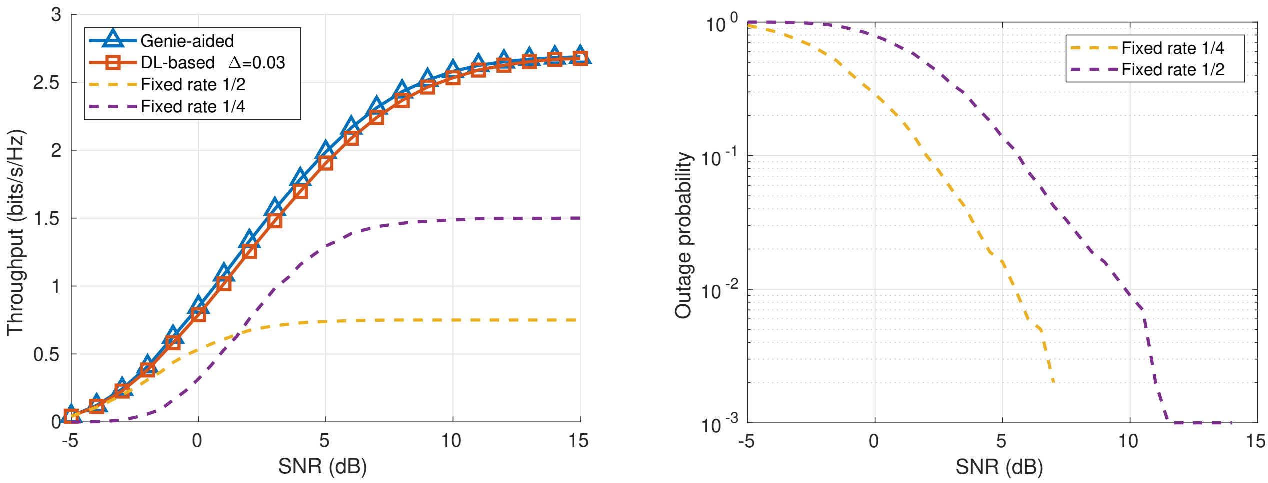
Finally, Fig. 5a shows the average throughput as a function of the SNR computed with the whole dataset, for the same Rayleigh distributed channel matrices. The blue triangles show the maximum achievable throughput, obtained with a genie who knows exactly which coding rate to choose. The red squares represents the throughput attained with our DL-based solution for the coding rate selection, employing a margin of $\Delta = 0.03$ for avoiding outage episodes. On the other hand, the two yellow and purple dashed lines are obtained with a fixed coding rate of $1/4$ and $1/2$, respectively.

The advantages of an adaptive SM system with variable coding rate are clearly exposed in Fig. 5a. Works like [19] and [20] consider only the adaptation of the modulation order and propose to employ a fixed and conservative coding rate for all the range of application of each constellation. This coarse rate adaptation procedure falls far from exploiting the achievable rate of the channel as Fig. 5a illustrates. Moreover, the proposed solution for selecting the coding rate based on DL is very close to the maximum achievable throughput (the genie-aided selection) and it outperforms the fixed coding rates curve for all the simulated SNRs. Lastly, for the sake of completeness, in Fig. 5b it is shown the average outage probability only for the fixed coding rate setup, since with both DL and genie-aided no outage takes place.

We have shown in [12] that a shallow neural network with a single hidden layer can obtain with high accuracy the theoretically achievable rate of SM, at a much lower computational cost than the previous numerical approximations of the MI expression. Going one step further, our results here show that by inserting two additional hidden layers, the neural network can learn to select the optimum coding rate in a real setup for a given family of channel codes and for a particular receiver implementation.

VI. Conclusion

The application of Adaptive Coding and Modulation to Spatial Modulation (SM) links requires receivers to be able to compute the achievable rates on-the-fly. The relation between the channel matrix and these rates is far from trivial, which poses major challenges to decide the most appropriate MCS for a given channel response. In this work we have shown how a deep neural network can be used if properly trained with all the available channel codes and a number of channel realizations. The achieved selection accuracy is quite high, and the outage caused by miss-classifications can be easily reduced



(a) Average throughput

(b) Average outage probability

Fig. 5: Average throughput and average outage probability per SNR in a 2×2 SM system with a QPSK constellation and Rayleigh distributed channel matrices.

by subtracting a judiciously chosen back-off margin from the neural network output. This fine adaptation of the coding rate assisted with a deep neural network allows to increase the throughput of the system, specially when compared with other approaches with a fixed coding rate. The extension of the tests shown here for higher number of antennas and additional constellations is planned for further steps.

References

[1] E. Basar, M. Wen, R. Mesleh, M. Di Renzo, Y. Xiao, and H. Haas. Index Modulation Techniques for Next-Generation Wireless Networks. *IEEE Access*, 5:16693–16746, 2017.

[2] Krishna Sayana and Jeff Zhuang. Link performance abstraction based on mean mutual information per bit (MMIB) of the LLR channel, 2007.

[3] P. Henarejos, A. Perez-Neira, A. Tato, and C. Mosquera. Channel Dependent Mutual Information in Index Modulations. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3261–3265, April 2018.

[4] O. Simeone. A very brief introduction to machine learning with applications to communication systems. *IEEE Transactions on Cognitive Communications and Networking*, 4(4):648–664, Dec 2018.

[5] M. M. A. Moustafa and S. H. A. El-Ramly. Channel estimation and equalization using backpropagation neural networks in OFDM systems. In *2009 IFIP International Conference on Wireless and Optical Communications Networks*, pages 1–4, April 2009.

[6] M. T. E. A. Elsoufi, X. Ying, W. Jun, and T. Bin. Fletcher-Reeves learning approach for high order MQAM signal modulation recognition. In *2016 7th International Conference on Information and Communication Systems (ICICS)*, pages 74–79, April 2016.

[7] M. Mirmohammadsadeghi, S. S. Hanna, and D. Cabric. Modulation classification using convolutional neural networks and spatial transformer networks. In *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pages 936–939, Oct 2017.

[8] A. Marseet and F. Sahin. Application of complex-valued convolutional neural network for next generation wireless networks. In *2017 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*.

[9] P. V. R. Ferreira, R. Paffenroth, A. M. Wyglinski, T. M. Hackett, S. G. Biln, R. C. Reinhart, and D. J. Mortensen. Multiobjective Reinforcement Learning for Cognitive Satellite Communications Using Deep Neural Network Ensembles. *IEEE Journal on Selected Areas in Communications*, 36(5):1030–1041, May 2018.

[10] J. Kassab and S. Nagaraj. Adaptive modulation in an OFDM communications system with artificial neural networks. In *2009 International Joint Conference on Neural Networks*, pages 1547–1551, June 2009.

[11] V. Saxena, B. Cavarec, J. Jaldn, M. Bengtsson, and H. Tullberg. A learning approach for optimal codebook selection in spatial modulation systems. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 1800–1804, Oct 2018.

[12] A. Tato, C. Mosquera, P. Henarejos, and A. Pérez-Neira. Neural Network Aided Computation of Mutual Information for Adaptation of Spatial Modulation. *arXiv e-prints*, page arXiv:1904.10844, Apr 2019.

[13] Y. S. Cho, Jaekwon K., W. Y. Yang, and Chung G. Kang. *MIMO-OFDM Wireless Communications with MATLAB*. Wiley Publishing, 2010.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[15] M. T. Hagan and M. B. Menhaj. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, Nov 1994.

[16] Alberto Rico-Alvarino, Anxo Tato, and Carlos Mosquera. Robust adaptive coding and modulation scheme for the mobile satellite forward link. In *Signal Processing Advances in Wireless Communications (SPAWC), 2015 IEEE 15th International Workshop on*, June 2015.

[17] Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 1: DVB-S2 . *ETSI EN 302 307-1 V1.4.1 (2014-11)*.

[18] K. Scharnhorst. Angles in complex vector spaces. *Acta Applicandae Mathematica*, 69(1):95–103, Oct 2001.

[19] P. Yang, Y. Xiao, L. Li, Q. Tang, Y. Yu, and S. Li. Link Adaptation for Spatial Modulation With Limited Feedback. *IEEE Transactions on Vehicular Technology*, 61(8):3808–3813, Oct 2012.

[20] P. Yang, M. Di Renzo, Y. Xiao, S. Li, and L. Hanzo. Design Guidelines for Spatial Modulation. *IEEE Communications Surveys Tutorials*, 17(1):6–26, Firstquarter 2015.