LEARNING MODEL-FREE ROBUST PRECODING
FOR COOPERATIVE MULTIBEAM SATELLITE COMMUNICATIONS

Steffen Gracla   *Alea Schröder*   *Maik Röper*
Carsten Bockelmann   *Dirk Wübben*   *Armin Dekorsy*

Dept. of Communications Engineering, University of Bremen, Bremen, Germany
Email: {gracla, schroeder, roeper, bockelmann, wuebben, dekorsy}@ant.uni-bremen.de

ABSTRACT

Direct Low Earth Orbit satellite-to-handheld links are expected to be part of a new era in satellite communications. Space-Division Multiple Access precoding is a technique that reduces interference among satellite beams, therefore increasing spectral efficiency by allowing cooperating satellites to reuse frequency. Over the past decades, optimal precoding solutions with perfect channel state information have been proposed for several scenarios, whereas robust precoding with only imperfect channel state information has been mostly studied for simplified models. In particular, for Low Earth Orbit satellite applications such simplified models might not be accurate. In this paper, we use the function approximation capabilities of the Soft Actor-Critic deep Reinforcement Learning algorithm to learn robust precoding with no knowledge of the system imperfections.

Index Terms— Multi-user beamforming, 3D networks, Low Earth Orbit (LEO), Machine Learning (ML), deep Reinforcement Learning (RL)

1. INTRODUCTION

Integrating Non Terrestrial Networks (NTN), e.g. satellites and Unmanned Aerial Vehicles (UAVs), into current terrestrial infrastructure is one of the important pillars in the development of the sixth-generation standard of mobile communication networks [1]. So-called holistic 3D networks will enable ubiquitous global coverage, provide capacity for temporally and locally varying traffic demands and enhance the robustness of terrestrial network infrastructure [2, 3]. However, a host of challenges is introduced due to the high dynamism of NTN devices. Low Earth Orbit (LEO) satellites have especially fast-changing Line-of-Sight (LoS) channels, which are mainly characterized by the relative positions between satellites and users. To increase spectral efficiency through frequency reuse, precoding based Space-Division Multiple Access (SDMA) is used in satellite communications [4]. However, the performance suffers due to imperfect positional information [5, 6].

Finding a precoding algorithm that maximizes performance metrics such as the sum rate for multiple geometric constellations while also showing robustness against imperfect positioning and channel estimates can prove challenging. In light of this, Machine Learning (ML) methods present themselves as an attractive choice. ML can be used to approximate a viable algorithm where the optimum is either infeasible to determine or wholly unavailable. Deep Learning (DL) in particular has demonstrated tremendous potential on such problems in the past decade, [7, 8]. Applying DL to precoding has recently started to gather more attention, primarily in terrestrial communications, e.g., [9] use supervised learning to approximate a lower complexity Minimum Mean Squared Error (MMSE) precoder, [10] show the ability of Reinforcement Learning (RL) precoders to optimally learn on toy scenarios without interference and [11] use an autoencoder structure to learn robust precoding and decoding under imperfect channel knowledge. In the LEO satellite context, [6] have extended their work on robust precoding pertaining imperfect positional knowledge for a single satellite scenario by a supervised low-complexity approximation. In this paper, we will use model-free deep RL. In RL, an agent probes the environment (i.e., selecting a precoding matrix and observing the result), thereby generating data to learn from, to gain understanding of the system dynamics and adjust their behavior to maximize an objective. By using this approach, no assumptions about the error modeling need to be made; it is instead discovered and inferred from the data. However, data-driven learning requires data samples containing high information content to learn efficiently. For this reason, we use the Soft Actor-Critic (SAC) learning algorithm [12]. SAC encourages exploring new data samples where the algorithm's understanding is low, generating the necessary high quality data faster than random exploration.

In the next section, we introduce the system model of cooperative multibeam satellite communication, the applied Channel State Information at Transmitter (CSIT) error models, typical precoding approaches, and the sum rate maximization problem. Following that, we explain the SAC method as it is used in this paper. We then apply SAC to maximize the sum rate in the presence of CSIT error, and discuss & contrast the performance against common MMSE and Orthogonal Multiple Access (OMA) precoding. Finally, we briefly discuss the scalability of the model case presented in this paper. For brevity, we assume prior knowledge of deep Neural Networks (NNs) [13].

Notations: Lower and upper boldface letters denote vectors $\mathbf{x}$ and matrices $\mathbf{X}$ with $\mathbf{I}_N$ being an identity matrix of size $N \times N$. Transpose and Hermitian operators are indicated by $\{\cdot\}^\mathrm{T}$ and $\{\cdot\}^\mathrm{H}$, whereas $\circ$ is the Hadamard product. $|\cdot|$ and $\|\cdot\|$ signify the absolute value and Euclidean norm, respectively.

2. SATELLITE COMMUNICATION SETUP & NOTATIONS

This section introduces the applied LoS channel model with errors regarding the measurement of user and satellite positions. Further, we explain two common precoding techniques for satellite communications.

2.1. System Model

In this paper, we consider a multi-user downlink scenario with M LEO satellites, each equipped with a Uniform Linear Array (ULA) of N antennas, serving K handheld users with one receive antenna each and low receive antenna gain G_Usr. It is assumed that all satellites are provided with the data symbols of all users and perform joint precoding, i.e., all satellites perform the precoding together. The data symbol s_k of user k is weighted by a precoding vector $\mathbf{w}_k \in \mathbb{C}^{MN \times 1}$ and transmitted over the LoS channel $\mathbf{h}_k \in \mathbb{C}^{1 \times MN}$ with Additive White Gaussian Noise (AWGN) $n_k \sim \mathcal{CN}(0, \sigma_n^2)$. Hence, the receive signal y_k follows as

$$y_k = \mathbf{h}_k \mathbf{w}_k s_k + \mathbf{h}_k \sum_{l \neq k}^{K} \mathbf{w}_l s_l + n_k. \tag{1}$$

The LoS channel vector $\mathbf{h}_{k,m} \in \mathbb{C}^{1 \times N}$ from satellite m to user k with $\mathbf{h}_k = [\mathbf{h}_{k,1} \ldots \mathbf{h}_{k,M}]$ is given by

$$\mathbf{h}_{k,m}(\nu_{k,m}) = \frac{\lambda \sqrt{G_\mathrm{Usr} G_\mathrm{Sat}}}{4\pi d_{k,m}} \mathrm{e}^{-j\varphi_{k,m}} \mathbf{v}_{k,m}(\cos(\nu_{k,m})), \tag{2}$$

where G_Sat denotes the satellite antenna gain, λ is the wavelength and $d_{k,m}$ is the distance between the m-th satellite and the k-th user. The overall phase shift from satellite m to user k is given by $\varphi_{k,m} \in [0, 2\pi]$. The relative phase shifts between the N antennas of a satellite m to user k are described by the steering vector $\mathbf{v}_{k,m}(\cos(\nu_{k,m})) \in \mathbb{C}^{1 \times N}$, where $\nu_{k,m}$ is the Angle of Departure (AoD) from satellite m to user k. The n-th entry of the steering vector $\mathbf{v}_{k,m}(\cos(\nu_{k,m}))$ is given as

$$v_{k,m}^n(\cos(\nu_{k,m})) = \mathrm{e}^{-j\pi \frac{d_n}{\lambda}(N+1-2n)\cos(\nu_{k,m})}, \tag{3}$$

where d_n is the inter-antenna-distance between the N antennas per satellite. Given that the inter-antenna-distance d_n is fixed and known, the phase differences in (3) between antenna n and n' are only determined by the k-th user position, reflected in the space angle $\cos(\nu_{k,m})$. In practice, a precise estimation of the user's position may not be feasible due to the high velocities of LEO satellites. We assume that user k's position can be well estimated within a certain region. Therefore, we model the estimation error on the space angle $\cos(\nu_{k,m})$ as a uniformly distributed additive error $\varepsilon_{k,m} \sim \mathcal{U}(-\Delta\varepsilon, +\Delta\varepsilon)$ [14]. Since the error $\varepsilon_{k,m}$ is added on the cosine of the AoDs and given the definition of the steering vector (3), the erroneous channel vector $\tilde{\mathbf{h}}_{k,m}^{(1)}(\nu_{k,m})$ can be expressed as an overall multiplicative error $\mathbf{v}_{k,m}(\varepsilon_{k,m}) \in \mathbb{C}^{1 \times N}$ on the true channel $\mathbf{h}_{k,m}(\nu_{k,m})$

$$\tilde{\mathbf{h}}_{k,m}^{(1)}(\nu_{k,m}, \varepsilon_{k,m}) = \mathbf{h}_{k,m}(\nu_{k,m}) \circ \mathbf{v}_{k,m}(\varepsilon_{k,m}) \tag{4}$$

with the n-th entry of the error vector $\mathbf{v}_{k,m}(\varepsilon_{k,m})$

$$v_{k,m}^n(\varepsilon_{k,m}) = \mathrm{e}^{-j\pi \frac{d_n}{\lambda}(N+1-2n)\varepsilon_{k,m}}. \tag{5}$$

Joint precoding with multiple satellites requires tight inter-satellite synchronization. If such synchronization is imperfect, the signals from different satellites arrive with different phases. To model such synchronization mismatch, we assume an error $\zeta_{k,m} \sim \mathcal{N}(0, \sigma_\zeta^2)$ on the overall phase shift $\varphi_{k,m}$. The resulting erroneous channel estimation with imperfect synchronization and position knowledge is given as

$$\tilde{\mathbf{h}}_{k,m}^{(2)}(\nu_{k,m}, \varepsilon_{k,m}, \zeta_{k,m}) = \mathrm{e}^{-j\zeta_{k,m}} \cdot \tilde{\mathbf{h}}_{k,m}^{(1)}(\nu_{k,m}, \varepsilon_{k,m}). \tag{6}$$

In this paper, we analyze the influence of both error models, (4) and (6), on the precoding performance. We evaluate the performance of our different precoding techniques by comparing their corresponding sum rates

$$R = \sum_{k=1}^{K} \log \left(1 + \frac{|\mathbf{h}_k \mathbf{w}_k|^2}{\sigma_n^2 + \sum_{l \neq k}^{K} |\mathbf{h}_k \mathbf{w}_l|^2} \right) \tag{7}$$

Our goal is to maximize the sum rate (7) for various satellite-to-user constellations while enhancing robustness against imperfect user position knowledge at the satellites (4) as well as imperfect satellite position knowledge (6). We propose to learn a robust precoding algorithm using the SAC method, to be introduced in Section 3. Further, we analyze two common approaches explained in the subsequent section.

2.2. Baseline Precoding Techniques

This subsection presents two common precoding approaches for satellite downlink communications: the conventional MMSE precoder for SDMA, and an OMA approach assuming orthogonal time or frequency resources.
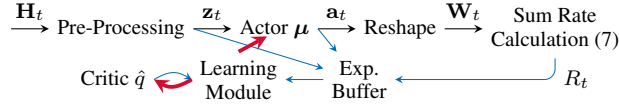
Fig. 1. SAC Precoder process flow. Top row describes inference, bottom row describes learning. Blue arrows relate to learning, red arrows show NN parameter updates.

2.2.1. MMSE

For optimal SDMA precoding, the precoding vector $\mathbf{w}_k$ steers a beam with maximal power into the direction of the corresponding user k while minimizing Inter-User Interference (IUI). The MMSE precoder has been proven to be a reliable precoder in this manner and is widely used [15, 16]. For MMSE, the precoding matrix $\mathbf{W}^{\text{MMSE}} = \left[\mathbf{w}_1^{\text{MMSE}} \ldots \mathbf{w}_K^{\text{MMSE}} \right]$ for a given channel estimate $\tilde{\mathbf{H}} = [\tilde{\mathbf{h}}_1 \ldots \tilde{\mathbf{h}}_K]^{\text{T}}$ is calculated as follows

$$
\mathbf{W}^{\text{MMSE}} = \sqrt{\frac{P}{\text{tr}\{\mathbf{W}'^{\text{H}} \mathbf{W}'\}}} \cdot \mathbf{W}'
$$

$$
\mathbf{W}' = \left[\tilde{\mathbf{H}}^{\text{H}} \tilde{\mathbf{H}} + \sigma_n^2 \cdot \frac{K}{P} \cdot \mathbf{I}_{MN} \right]^{-1} \tilde{\mathbf{H}}^{\text{H}}
$$

(8)

where P denotes the total amount of transmit power. In this paper, we further constrain the maximum transmit power per satellite P_m to be equally distributed among all M satellites, such that $P_m \leq P/M$ applies. Note that the MMSE precoding approach does not necessarily maximize the sum rate R (7).

2.2.2. OMA

In contrast to SDMA, OMA uses orthogonal time or frequency resources. In this case there is no IUI between the user channels. Therefore, the optimal precoder is a Maximum Ratio Transmission (MRT) precoder that solely maximizes the transmit power steered into the direction of a given user k

$$
\mathbf{w}_k^{\text{MRT}} = \sqrt{P} \cdot \frac{\tilde{\mathbf{h}}_k^{\text{H}}}{\|\tilde{\mathbf{h}}_k\|}.
$$

(9)

Here the total transmit power P is used for user k. Due to the absence of IUI, the sum rate for OMA calculates as

$$
R^{\text{OMA}} = \frac{1}{K} \sum_{k=1}^{K} \log \left(1 + \frac{|\mathbf{h}_k \mathbf{w}_k|^2}{\sigma_n^2} \right).
$$

(10)

To guarantee a fair comparison between OMA and the different SDMA approaches, the overall rate is divided by the number of users K, taking into account that in SDMA time and frequency resources are shared between all users. The MMSE and the OMA approach will serve as baselines for our proposed learning algorithm.

In the next section, we will discuss how to iteratively learn a robust precoding algorithm that approximately optimizes the sum rate (7) based on the current channel state estimate.

3. LEARNING TO PRECODE USING SOFT ACTOR-CRITIC

In deep RL, an agent learns to approximately maximize a performance metric by interacting with a system and adjusting its behavior according to the system feedback. Here, the metric is the sum rate R (7) and the behavior corresponds to the precoding algorithm. In this section, we introduce the Soft Actor-Critic (SAC) [12] RL method, which is characterized by rewarding exploration where uncertainty is high, thereby generating high quality data samples.

Our SAC implementation uses five components, shown in Fig. 1: 1) a Pre-Processing step, 2) two Critic Neural Networks (CrNNs) $\hat{q}_1$ and $\hat{q}_2$, 3) an Actor Neural Network (AcNN) $\boldsymbol{\mu}$, 4) an Experience Buffer, and 5) a Learning Module. We denote the NN weights as $\boldsymbol{\theta}_{\hat{q}_1}, \boldsymbol{\theta}_{\hat{q}_2}, \boldsymbol{\theta}_{\mu}$ for $\hat{q}_1, \hat{q}_2, \boldsymbol{\mu}$, respectively. The AcNN will form the precoding matrix $\mathbf{W}$. In SAC, it produces stochastic output using the reparametrization-trick: For the i-th entry a_i of an output vector $\mathbf{a}$, the AcNN produces two outputs, one of which is considered a mean $\mu_{i,\bar{\mu}}$ and the other a scale $\mu_{i,\sigma}$. The outputs $a_i \sim \mathcal{N}(\mu_{i,\bar{\mu}}, \mu_{i,\sigma})$ are then sampled from a Normal distribution.

A discrete inference and learning step t looks as follows: First, the Pre-Processing step prepares the amplitude and phase components of the complex valued erroneous CSIT $\tilde{\mathbf{H}}_t \in \mathbb{C}^{K \times MN}$ in a flat state vector $\mathbf{z}_t \in \mathbb{R}^{1 \times 2MNK}$ as input for the NN. Based on this input vector $\mathbf{z}_t$ and its current parameters $\boldsymbol{\theta}_{\mu,t}$, the AcNN $\boldsymbol{\mu}$ will output an action vector $\boldsymbol{\mu}_{\boldsymbol{\theta}_{\mu,t}}(\mathbf{z}_t) = \mathbf{a}_t \in \mathbb{R}^{1 \times 2MNK}$. This real-valued vector $\mathbf{a}$ is then reshaped into a complex-valued precoding matrix $\mathbf{W}_t \in \mathbb{C}^{MN \times K}$ with entries

$$
w_{t,k,m}^n = \mathbf{a}_{t,k+m+n} + j \mathbf{a}_{t,N+k+m+n},
$$

Table 1. Selected Parameters

Noise Power σ_n^2	6e-13 W	Transmit Power P	100 W
Sat. Altitude	600 km	Ant. per Sat. N	2
Sat. Nr. M	2	Gain per Sat. Ant G_{Sat}	14 dBi
Sat. Distance	10 km	Gain per User G_{Usr}	0 dBi
User Nr. K	3	Inter-Ant.-Distance d_n	$3\lambda/2$
Wavelength λ	15 cm	Hidden Layers $\times$ Nodes	4×512
Learning Steps t	3e4	Critic Learning Rate	1e-5
Exp Buffer Size	1e4	Actor Learning Rate	1e-6
Entropy Target α	1.0	Data Batch Size B	512

and is normalized to the available transmit power per satellite P/M, same as the MMSE precoder. Subsequently, precoding is performed, the resulting sum rate R_t (7) is calculated. The data sample of state $\mathbf{z}_t$, action $\mathbf{a}_t$ and result R_t is saved in the Experience Buffer. This concludes the inference part.

Next, to improve the quality of the AcNN's outputs, its weights $\boldsymbol{\theta}_{\mu,t}$ must be tuned. The Actor-Critic method operates as follows: We wish to update the weights to maximize the mapping of $(\mathbf{z}, \mathbf{a}) \rightarrow R$, outputting precoding that maximizes the sum rate. This mapping is, however, unknown to the learning method. Given the collected data samples from the Experience Buffer, the CrNNs can approximate a mapping $\hat{q}(\mathbf{z}, \mathbf{a}) = \hat{R}$. If the true mapping is approximated sufficiently well, the AcNN's weights $\boldsymbol{\theta}_\mu$ can be updated to maximize the known mappings $\hat{q}$. Weight updates are performed once per training iteration t on both CrNNs $\hat{q}_1, \hat{q}_2$ and AcNN $\boldsymbol{\mu}$. The CrNN $\hat{q}_1, \hat{q}_2$ are updated in a supervised learning manner, minimizing the squared distance

$$\mathcal{L}_{\hat{q}} = (\hat{q}_{\boldsymbol{\theta}_{\hat{q},t}}(\mathbf{z}_t, \mathbf{a}_t) - R_t)^2 = (\hat{R}_t - R_t)^2$$

between the approximation $\hat{R}_t$ given current weights $\boldsymbol{\theta}_{\hat{q},t}$ and the data sample target R_t. We then construct an AcNN loss

$$\mathcal{L}_{\mu,1} = -\min(\hat{q}_{1,\boldsymbol{\theta}_{\hat{q}_1,t}}(\mathbf{z}_t, \boldsymbol{\mu}_{\boldsymbol{\theta}_{\mu,t}}(\mathbf{z}_t)), \hat{q}_{2,\boldsymbol{\theta}_{\hat{q}_2,t}}(\mathbf{z}_t, \boldsymbol{\mu}_{\boldsymbol{\theta}_{\mu,t}}(\mathbf{z}_t)))$$

that minimizes the negative approximate expected results $\hat{R}_t$, thereby maximizing the approximate expected results. Selecting conservatively from multiple, independently initialized mappings $\hat{q}$ has been shown to stabilize the learning process.

This loss, however, gives no incentive for the AcNN to maintain its stochastic output's scales $\mu_{i,\sigma}$, i.e., incentive to explore new data samples where uncertain. During the learning process, insufficient exploration might lead to an incomplete data set that lacks information required to discover solutions near the global optimum. To prevent the AcNN from contracting its scales $\mu_{i,\sigma}$ too rapidly, the SAC method additionally adds an entropy loss

$$\mathcal{L}_{\mu,2} = \frac{1}{2MNK} \sum_{i=1}^{2MNK} \exp(\alpha) \log(\pi(\mu_{i,\boldsymbol{\theta}_{\mu,t}}(\mathbf{z}_t), \boldsymbol{\theta}_{\mu,t})),$$

where $\pi(\mu_{i,\boldsymbol{\theta}_{\mu,t}}(\mathbf{z}_t), \boldsymbol{\theta}_{\mu,t})$ is the probability of $\mu_{i,\boldsymbol{\theta}_{\mu,t}}(\mathbf{z}_t)$ given the current weights $\boldsymbol{\theta}_{\mu,t}$ of the AcNN $\boldsymbol{\mu}$ and α is a weighting factor. In the mean sense, this loss is directly minimized by increasing the AcNN output variance and therefore represents the necessary counterweight to the first loss $\mathcal{L}_{\mu,1}$. The scaling factor α is updated iteratively to keep the entropy roughly constant over the duration of training, i.e., it is adjusted whenever the entropy falls above or below a heuristic target value. Summarily, the AcNN $\boldsymbol{\mu}$ updates its weights $\boldsymbol{\theta}_\mu$ to minimize a composite loss $\mathcal{L}_\mu = \mathcal{L}_{\mu,1} + \mathcal{L}_{\mu,2}$. Both AcNN and CrNN parameter updates are performed using Stochastic Gradient Descent (SGD)-like optimization on batches of B data samples drawn from the Experience Buffer with uniform probability.

In the following section, we evaluate specific implementation details and the performance of the learned precoder for given scenarios.

4. EVALUATION

In this section, we cover specific implementation details and compare the performance of SAC-learned precoders to the MMSE and OMA baselines.

4.1. Implementation Details

The code is implemented in Python using the TensorFlow library and is available online at [17]. Table 1 lists the key communication and learning parameters. For each iteration t, user positions are updated around a mean inter-user-distance $\bar{D}_{\text{Usr}} = 1$ km following a uniform distribution of ± 30 m. This is done to prevent the learning process from honing in on a fixed user position regardless of the information it is fed, but it also highlights the method's ability to learn a single precoding algorithm that serves different satellite and user constellations.

We train three SAC precoders named SAC1, SAC2, SAC3. SAC1 is trained with perfect CSIT, SAC2 is trained on error model 1 (4) with a bound of $\Delta\varepsilon = 0.1$, and SAC3 is trained on error model 2 (6) with a bound of $\Delta\varepsilon = 0.1$ and a scale of $\sigma_\zeta = 0.01$. All precoders are evaluated in terms of the achieved sum rate R. In cases with imperfect CSIT, the sum rate $\bar{R}$ is averaged over 10 000 Monte Carlo iterations.
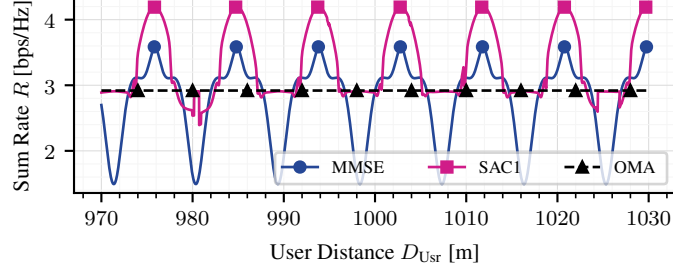
Fig. 2. The sum rate (7) for different user distances D_{Usr} with perfect CSIT clearly highlights the performance gains of the SAC precoder over MMSE, particularly when channels are highly correlated (valleys) or nearly orthogonal (peaks). The SAC precoder shows glitches, e.g., around $980\,\mathrm{m}$; it achieves its performance despite not having reached full convergence during training.
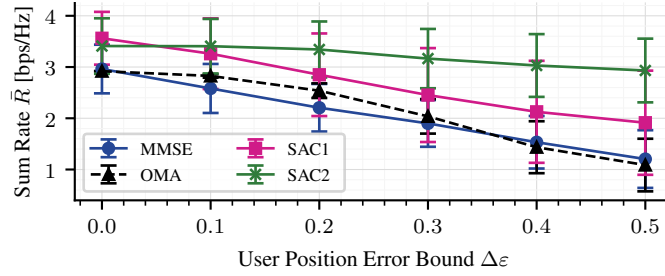

Fig. 3. Mean sum rate performance of different precoding approaches in the presence of an error on CSIT according to the first error model (4). Error bars represent the standard deviation. Note that user distances are also varied within each evaluation, hence the variance even at zero error.
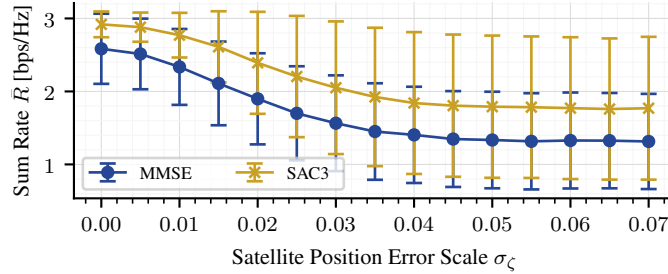

Fig. 4. Mean sum rate performance on the second error model, (6). We fixate the user position error bound to $\Delta\varepsilon = 0.1$ and sweep over the satellite position error scale σ_ζ. Again, the learned algorithm achieves higher sum rates than the MMSE precoder, as well as slightly increased robustness at higher errors.

4.2. Results

We first investigate the details of the precoding algorithm SAC1 for perfect CSIT in Fig. 2. As the user distances D_{Usr} change, the user correlation in user channels alternates periodically, significantly impacting the achievable performance. The learned precoder shows gains over the MMSE precoder especially in the extreme areas, with either highly correlated or nearly orthogonal channels. When the channels are highly correlated, the learned precoder achieves its gain by focusing as much power as possible on a single user, thereby decreasing interference, but sacrificing fairness. Crucially, the learned precoder outperforms OMA almost everywhere, demonstrating the increased spectral efficiency by frequency reuse. In further investigations, the learned algorithms generalize well to user distances up to at least $\pm\,300\,\mathrm{m}$ for the given scenario. We attribute this to the learning algorithm capturing the locally periodic nature of the optimization objective during training.

Fig. 3 compares SAC1 and SAC2 to MMSE and OMA precoding with unreliable user position measurements (4) for increasing error bound $\Delta\varepsilon$. As expected, since MMSE is not sum rate optimal, SAC1 achieves the best sum rate performance with perfect CSIT, with SAC2 slightly behind and both outperforming the baselines MMSE and OMA. With increasingly unreliable user position estimates, the performance of all four precoders degrades. However, SAC2, which already encountered unreliable information during training, shows resilience even at very high values of $\Delta\varepsilon$.

We then confirm in Fig. 4 that the same learning approach also works for imperfections of satellite positions (6), demonstrating the adaptability of model-free data-driven optimization.

5. CONCLUSIONS

We have applied the SAC RL method to the problem of sum rate optimal precoding for cooperative multibeam satellite communications in the presence of erroneous channel knowledge. Our results show that we are able to learn both effective and robust precoding algorithms with no assumptions on the underlying error model. We therefore expect this approach to scale well on real life satellites, particularly with sights on dedicated ML hardware being deployed on future satellites [18].

6. REFERENCES

[1] 3GPP TR 38.863, "Technical specification group radio access network; solutions for nr to support non-terrestrial networks (NTN): Non-terrestrial networks (NTN) related RF and co-existence aspects (release 17)," Sep. 2022.

[2] Israel Leyva-Mayorga, Beatriz Soret, Maik Röper, Dirk Wübben, Bho Matthiesen, Armin Dekorsy, and Petar Popovski, "LEO small-satellite constellations for 5G and beyond-5G communications," *IEEE Access*, vol. 8, pp. 184955–184964, 2020.

[3] Zhicheng Qu, Gengxin Zhang, Haotong Cao, and Jidong Xie, "LEO Satellite Constellation for Internet of Things," *IEEE Access*, vol. 5, pp. 18391–18401, 2017.

[4] Miguel Ángel Vázquez, MR Bhavani Shankar, Charilaos I Kourogiorgas, Pantelis-Daniel Arapoglou, Vincenzo Icolari, Symeon Chatzinotas, Athanasios D Panagopoulos, and Ana I Pérez-Neira, "Precoding, scheduling, and link adaptation in mobile interactive multibeam satellite systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 5, pp. 971–980, 2018.

[5] Maik Röper, Bho Matthiesen, Dirk Wübben, Petar Popovski, and Armin Dekorsy, "Beamspace MIMO for Satellite Swarms," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2022, pp. 1307–1312.

[6] Yanhao Liu, Yibiao Wang, Jue Wang, Li You, Wenjin Wang, and Xiqi Gao, "Robust downlink precoding for leo satellite systems with per-antenna power constraints," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 10694–10711, 2022.

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, "Playing Atari with Deep Reinforcement Learning," *arXiv:1312.5602*, 2013.

[8] Hayssam Dahrouj, Rawan Alghamdi, Hibatallah Alwazani, Sarah Bahanshal, Alaa Alameer Ahmad, Alice Faisal, Rahaf Shalabi, Reem Alhadrami, Abdulhamit Subasi, Malak T. Al-Nory, Omar Kittaneh, and Jeff S. Shamma, "An Overview of Machine Learning-Based Techniques for Solving Optimization Problems in Communications and Signal Processing," *IEEE Access*, vol. 9, pp. 74908–74938, 2021.

[9] Shaoqing Zhang, Jindan Xu, Wei Xu, Ning Wang, Derrick Wing Kwan Ng, and Xiaohu You, "Data augmentation empowered neural precoding for multiuser mimo with mmse model," *IEEE Communications Letters*, vol. 26, no. 5, pp. 1037–1041, 2022.

[10] Heunchul Lee, Maksym Girnyk, and Jaeseong Jeong, "Deep reinforcement learning approach to MIMO precoding problem: Optimality and Robustness," June 2020, arXiv:2006.16646.

[11] Foad Sohrabi, Hei Victor Cheng, and Wei Yu, "Robust Symbol-Level Precoding Via Autoencoder-Based Deep Learning," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020, pp. 8951–8955, IEEE.

[12] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine, "Soft Actor-Critic Algorithms and Applications," *arXiv:1812.05905*, Jan. 2019.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2020.

[14] Maik Röper, Bho Matthiesen, Dirk Wübben, Petar Popovski, and Armin Dekorsy, "Robust precoding via characteristic functions for vsat to multi-satellite uplink transmission," 2023.

[15] Christoph Windpassinger, *Detection and precoding for multiple input multiple output channels*, Ph.D. thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2004.

[16] Symeon Chatzinotas, Gan Zheng, and Björn Ottersten, "Energy-efficient MMSE beamforming and power allocation in multibeam satellite systems," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2011, pp. 1081–1085.

[17] Steffen Gracla and Alea Schröder, "Learning Beamforming," https://github.com/Steffengra/2302_learning_beamforming_code, 2023.

[18] Gianluca Giuffrida, Lorenzo Diana, Francesco de Gioia, Gionata Benelli, Gabriele Meoni, Massimiliano Donati, and Luca Fanucci, "CloudScout: A Deep Neural Network for On-Board Cloud Detection on Hyperspectral Images," *Remote Sensing*, vol. 12, no. 14, pp. 2205, July 2020.