Towards End-to-End Learning for Salinity-based Molecular Communication

Roya Khanzadeh*[1,2], Stefan Angerbauer*[1,2], Franz Enzenhofer[1], Andreas Springer[1], and Werner Haselmayr[1]

[1]Johannes Kepler University Linz, Institute for Communications Engineering and RF-Systems, Linz, Austria

[2]JKU LIT SAL eSPML Lab, Linz, Austria

I. Introduction

END-to-end learning of communication systems is a relatively new concept in which the whole communication system including the transmitter (TX), the channel, and the receiver (RX) is modeled as a Neural Network (NN), a so-called autoencoder (AEC), and jointly optimized to maximize the information rate. This concept has been already successfully applied in several wireless communication scenarios [1] and can also be a promising approach to molecular communication (MC) [2]. In MC, which uses molecules for information transmission, it is often difficult to drive an accurate yet tractable channel model and, thus, data-driven approaches can better handle the imperfection of the assumed models. Although there are some works on the application of NNs in MC, they mostly focus on detection algorithms [3], [4]. A deep reinforcement learning AEC has been proposed in [2], where a fully connected NN is trained for a MC channel by fixing the weights of the network at the TX and adjusting that of the RX and vice versa. In this work, we propose an AEC-based transceiver for a realistic MC channel with high intersymbol interference (ISI), where the channel model is taken from a recently published microfluidic testbed using different salinity levels for information transmission [5]. In contrast to [2], the proposed method jointly trains the weights of the NN both at TX and RX, utilizing back-propagation through the channel model and includes convolutional layers to combat the curse of dimensionality. The performance is also compared to a Viterbi sequence detection algorithm.

II. End-to-End Learning for Molecular Communication

Similar to wireless communication, the processing blocks in MC systems (including channel coding, modulation, equalization, demodulation, etc.) are mostly optimized individually, which does not certainly lead to the best possible end-to-end performance [1]. However, the entire MC system (from the TX to the RX) can be optimized using an AEC. AEC is a specific type of an artificial NN attempting to regenerate its input at the output, which is used for feature learning and dimensional reduction (or expansion). In the context of end-to-end learning in communication, AECs are used to learn the most important features of the data at the TX, which can then be propagated over the channel and reconstructed at the RX.

Fig. 1 shows the structure of the AEC for end-to-end learning of MC, consisting of three parts: a trainable encoder (TX), non-trainable layers with probabilistic and deterministic behavior (channel), and a trainable decoder (RX). Neural networks are typically trained using the well-known back-propagation algorithm, which requires calculating the gradient of the applied loss function with respect to all weights of the NN. In the case of AEC, back-propagation is possible for all layers except for the channel, which is a physical system and, thus, blocks a gradient flow to the TX. This issue can be addressed by taking the following two-step approach for training the AEC:

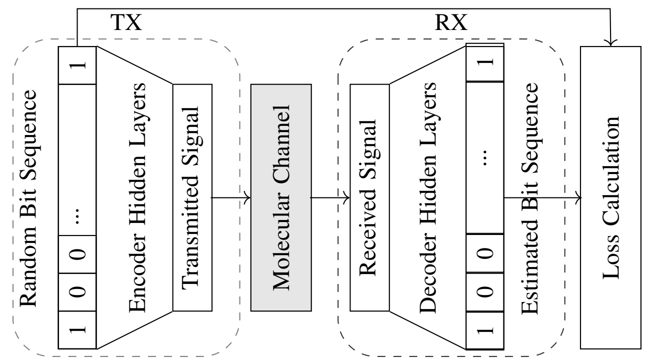

Fig. 1: AEC for end-to-end learning of MC.

1) Represent the channel by an estimated model (in our case an FIR filter), that allows differentiation of the channel output with respect to its input. Having access to this estimated channel model, training the network can be done without the need to access the physical system.
2) In the so-called "fine-tuning"-step the weights of the RX are refined by transmitting over the real channel and fixing the weights of the TX. Although this step is compulsory when the estimated channel model is not accurate or changes over time, dealing with that is left for future work.

To train our network, we used the estimated channel impulse response (CIR) of a salinity-based MC testbed presented in [5], given by

$$h(t) = \frac{1}{\sqrt{4\pi D_{\text{eff}} t}} e^{\frac{-(L-vt)^2}{4D_{\text{eff}} t}} \tag{1}$$

where $D_{\text{eff}} = 1.24 \times 10^{-4}\,\mathrm{m}^2/\mathrm{s}$ is the diffusion coefficient, $v = 0.055\,\mathrm{m/s}$ describes the velocity of the fluid flow and L is the length of the channel. The CIR is depicted in Fig. 2(a).[1]

III. Proposed Autoencoder Architecture

For the described MC system with deterministic distortion described in (1) and probabilistic noise effects (i.e., Gaussian

[1]In this work, we used this estimated model to show the effectiveness of AEC in MC, however, the full potential of AEC can be observed when the estimated model is not accurate, which will be investigated in future works.

*These authors contributed equally to this work.

TABLE I: The proposed AEC contains convolutional (Conv1d), linear (Linear1d) and pooling (Max-pooling) layers with normalization (BatchNorm1d) and non-linear mapping (ReLu or sigmoid).

Encoder (Transmitter)	
Type of layer	**Output size**
Input layer	M
Conv1d+BatchNorm1d+ReLu	$M \times 16$
Conv1d+BatchNorm1d+ReLu	$M \times 32$
Conv1d+ReLu+Pulse Shaping+Normalization	Mq
Decoder (Receiver)	
Conv1d+BatchNorm1d+ReLu	$Mq \times 16$
Adaptive average pooling	$Mq/2 \times 16$
Linear1d+BatchNorm1d+ReLu	$Mq/4 \times 16$
Conv1d+BatchNorm1d+ReLu	$Mq/4$
Max-pooling+sigmiod	M

noise), we propose an AEC which learns how to map a sequence of information bits to a sequence of symbols so that the RX can recover the original information bits from the distorted and noisy versions of the transmitted symbols. The architecture of the proposed AEC is summarized in Table I. Assuming independent and identically distributed (i.i.d) random bit sequences of length M as input, convolutional layers are used at the TX to only connect a few nearby neurons in the previous layer. Each convolutional layer of the encoder comes with a batch normalization and a non-linear activation function to be able to model any desirable function. For pulse-shaping, we use a conventional rectangular pulse with an oversampling ratio of q. The last normalization layer limits the average transmitted power. Please note that the proposed architecture in Table I is designed for binary transmission but it can be easily extended for non-binary communication.

A distorted and noisy version of the transmitted signal is received at the RX and the decoder is trained to recover the original bits. Inspired by linear equalization techniques, a convolutional layer with a kernel size proportional to the channel memory is used as the first layer in the decoder to combat the effect of ISI. Downsampling from the hidden features to the size of the transmitted bit sequences is done in four further steps, using two pooling layers as well as one linear fully connected layer and one convolutional layer. Binary cross-entropy is employed as a cost function for loss calculation (see Fig. 1) in order to maximize the information rate.

IV. Simulation Results

It is assumed that the CIR in (1) is fixed during the transmission and Gaussian noise is added to the received signal. In Fig. 2(b), the bit error ratio (BER) of the proposed method, i.e. AEC, is compared to Viterbi binary sequence detection [6], referred to as VIT, for various symbol durations (T_s) versus different signal to noise ratios (SNR). For the Viterbi algorithm, simple on-off keying modulation with rectangular pulse shaping is used. It can be observed that in low-ISI environments (i.e., $T_s = 0.5\,\mathrm{s}$), the AEC algorithm shows almost the same error performance as the VIT algorithm. On the other hand, when the symbol duration is decreased from $0.5\,\mathrm{s}$ to $0.3\,\mathrm{s}$, the AEC algorithm maintains its performance and outperforms the VIT algorithm. The superior performance of the proposed AEC method can be attributed to the additional degree of freedom it provides at the TX, allowing for joint encoding and modulation of data into multiple signal levels while maintaining the same transmit power and data rate compared to model-based approaches. Please note that the proposed AEC only requires training for a single SNR and can be evaluated for multiple SNRs. Both AEC and VIT algorithms exhibit linear growth in terms of time complexity as M increases. However, the AEC also requires additional time for the training phase, but this can be reduced by using parallel processing and specialized hardware like GPUs.

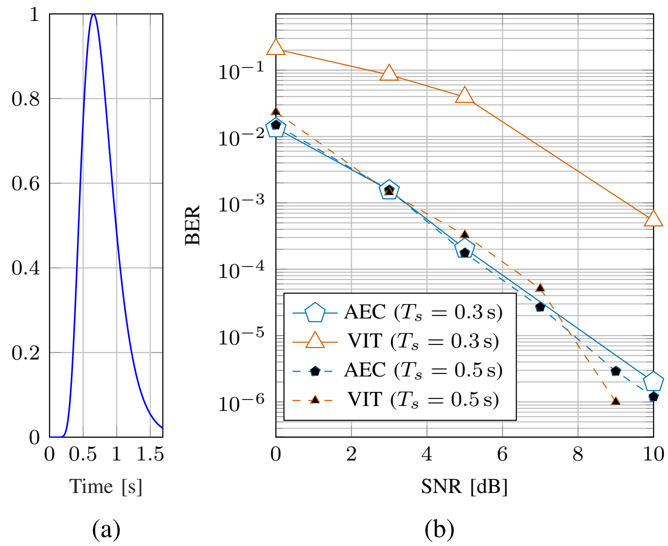

Fig. 2: (a) Normalized CIR (b) BER vs. SNR

V. Conclusion and Future Works

An AEC-based end-to-end learning for a salinity-based molecular communication testbed was proposed, in which TX and RX are jointly optimized to communicate over a channel with strong ISI effects. The proposed joint optimization showed very good performance in strong ISI environments and low-SNR regimes. Adapting the proposed method to be implemented on a real setup by applying fine-tuning on the RX side is the future direction of this research.

Acknowledgment

This work has been in part supported by the ”University SAL Labs” initiative of Silicon Austria Labs (SAL) and its Austrian partner universities for applied fundamental research for electronic-based systems.

References

[1] S. Cammerer *et al.*, “Trainable communication systems: Concepts and prototype,” *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5489–5503, 2020.

[2] S. Mohamed *et al.*, “Model-based: End-to-end molecular communication system through deep reinforcement learning auto encoder,” *IEEE Access*, vol. 7, pp. 70 279–70 286, 2019.

[3] N. Farsad *et al.*, “Neural network detection of data sequences in communication systems,” *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5663–5678, 2018.

[4] M. Bartunik *et al.*, “Deep learning to demodulate transmissions in molecular communication,” in *Workshop on MolComm*, 2022.

[5] S. Angerbauer *et al.*, “Salinity-based information transmission in microfluidic channels,” in *Proc. Int. Conf. on Nanoscale Computing and Communication*, 2022, pp. 1–2.

[6] J. Wang *et al.*, “Understanding and embracing the complexities of the molecular communication channel in liquids,” in *Proc. Int. Conf. on Mobile Computing and Networking*, 2020, pp. 1–15.