SK and Modulo-SK Matlab Environmet Manual

I. Background

A. The AWGN with Noisy Feedback Channel Model

The setup is depicted in Fig. 1. Terminal A and Terminal B are connected by a pair of independent AWGN channels, given by

$$Y_n = X_n + Z_n, \quad \widetilde{Y}_n = \widetilde{X}_n + \widetilde{Z}_n,$$

where X_n, Y_n (resp. $\widetilde{X}_n, \widetilde{Y}_n$) are the input and output of the feedforward (resp. feedback) channel at time n, respectively. The feedforward (resp. feedback) channel noise sequence $Z_n \sim \mathcal{N}(0, \sigma^2)$ (resp. $\widetilde{Z}_n \sim \mathcal{N}(0, \widetilde{\sigma}^2)$) is i.i.d., and independent of the input X_n (resp. $\widetilde{X}_n$). Terminal A wants to send a message W to Terminal B over N rounds of communication, where W is uniformly distributed over a set of cardinality M. To that end, at time n, the terminals send

$$X_n = \varphi_n(W, \widetilde{Y}^{n-1}), \quad \widetilde{X}_n = \widetilde{\varphi}_n(Y^n),$$

where $\varphi, \widetilde{\varphi}$ are predetermined functions, such that the average power constraint P (resp. $\widetilde{P}$) are satisfied:

$$\sum_{n=1}^{N} \mathbb{E}(X_n^2) \leq NP, \quad \sum_{n=1}^{N} \mathbb{E}(\widetilde{X}_n^2) \leq N\widetilde{P}.$$

This formulation is usually referred to in the literature as *active* feedback, where *passive* feedback implies $\widetilde{X}_n = Y_n$. We denote the feedforward (resp. feedback) signal-to-noise ratio by $\mathrm{SNR} \triangleq \frac{P}{\sigma^2}$ (resp. $\widetilde{\mathrm{SNR}} \triangleq \frac{\widetilde{P}}{\widetilde{\sigma}^2}$). The excess signal-to-noise ratio of the feedback over the feedforward is denoted by $\Delta\mathrm{SNR} \triangleq \frac{\widetilde{\mathrm{SNR}}}{\mathrm{SNR}}$. A communication scheme $(\varphi, \widetilde{\varphi})$ achieves a rate $R \triangleq \frac{\log M}{N}$ and an error probability p_e, which is the probability that Terminal B errs in decoding the message W at time N, under the optimal decision rule.

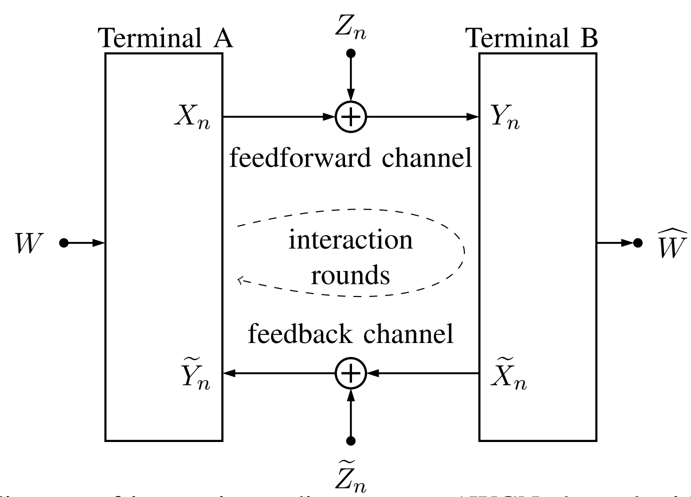

Fig. 1: Block diagram of interactive coding over an AWGN channel with noisy feedback

B. SK and Modulo-SK: A Brief Overview

Let us briefly review both the SK and Modulo-SK schemes. To that end, we first recall some basic facts from one-way communication. PAM is a simple and commonly used modulation scheme, where 2^R symbols are mapped (one-to-one) to the set

$$\{\pm 1\eta, \pm 3\eta, \cdots, \pm(2^R - 1)\eta\}, \tag{1}$$

where η is the normalization factor which is set so that the overall mean square of the constellation (assuming equiprobable symbols) is unity. While the Shannon capacity of and AWGN is know to be:

$$C(\mathrm{SNR}) = \frac{1}{2}\log(1 + \mathrm{SNR}), \tag{2}$$

it is well known, a uncoded-PAM can achieve a rate of

$$R = \frac{1}{2}\log\left(1 + \frac{\mathrm{SNR}}{\Gamma_0(p_e)}\right),$$

with an error probability not exceeding p_e, where $\Gamma_0(p_e)$ is the *PAM capacity gap*:

$$\Gamma_0(p_e) \triangleq \frac{1}{3}\left[Q^{-1}\left(\frac{p_e}{2}\right)\right]^2.$$

The operative meaning of the capacity gap is, that in order to achieve a rate R with error probability not exceeding p_e, uncoded PAM needs SNR which is $\Gamma_0(p_e)$ stronger than the minimal SNR derived from Shannon's limit (2). Below, we naturally extend the notion of capacity gap to the SK and Modulo-SK schemes.

The high level of the classical SK scheme [1], can be explained as follows: the message is mapped by Terminal A into a scalar message variable Θ using uncoded-PAM into. Terminal B (with the aid of Terminal A through the noiseless feedback and the feedforward channel) updates an estimate for Θ at every iteration round n, denoted by $\widehat{\Theta}_n$. Both terminals apply an iterative refinement scheme for $\widehat{\Theta}_n$ (see [2] for details) whose result is that the channel between Θ and $\widehat{\Theta}_N$ can be regarded as an AWGN with the following effective signal-to-noise ratio [1]:

$$\mathrm{SNR}_N = \mathrm{SNR} \cdot (1 + \mathrm{SNR})^{N-1}.$$

Now, the number of bits in the QAM constellation is nR, which for a target error probability p_e can be set according to (1) as follows:

$$NR = \frac{1}{2}\log\left(1 + \frac{\mathrm{SNR}_N}{\Gamma_0(p_e)}\right) = \frac{1}{2}\log\left(1 + \frac{\mathrm{SNR}\cdot(1+\mathrm{SNR})^{N-1}}{\Gamma_0(p_e)}\right). \tag{3}$$

On the other hand, the capacity gap of the SK scheme $\Gamma_0(p_e)$ is similarly defined as

$$R = \frac{1}{2}\log\left(1 + \frac{\mathrm{SNR}}{\Gamma^{\text{S-K}}(p_e)}\right), \tag{4}$$

Using (4) and (3) and solving for $\Gamma^{\text{S-K}}(p_e)$ demonstrates a rapid decay in the capacity gap with respect to the number of iterations, which can be regarded as another manifestation of the famous double-exponentially decay in the error probability of the SK scheme [3].

The SK scheme described above assumes a noiseless feedback link and is known to completely fail in the mere presence of noise in the feedback link. To remedy that, one can consider exponentially increasing the power over the feedback link to mimic the noiseless case – but this is of course not practical. Nevertheless, as we showed in [2], this naive approach can be made practical via the use of modulo-arithmetic.

[1]Note that it is also possible to achieve $\mathrm{SNR}_N = (1+\mathrm{SNR})^N - 1$, using a slightly different parameter setting [3, Chapter 17]

The modulo-SK scheme presented in [2] was shown to achieve an effective AWGN channel from Θ and $\widehat{\Theta}_N$ with:

$$\mathrm{SNR}_N = \frac{\mathrm{SNR}}{\Psi 3} \cdot \left(1 + \frac{\mathrm{SNR}}{\Psi_1 \Psi_2 \Psi_3}\right)^{N-1}.$$

where Ψ_1, Ψ_2, Ψ_3 are penalty terms, depending on p_e, SNR and ΔSNR which become manageable provided that ΔSNR is large enough. A characterization of the capacity gap and the performance is given in [2], [4].

II. The Matlab Environment

The main difference between standard point-to-point communication and the given schemes for feedback communication is the following: the current feedback schemes are tailored for a specific working point (namely, SNR, and ΔSNR, number of iterations N and target error probability p_e). Using a scheme in a different working point might not only result in exceeding error probability, but possibly to violating of the power constraints.

Therefore, for every working point, the scheme coefficients should be calculating beforehand. This is done by using the following function:

```
[snrShannondB,CapGapdB] = calcSNRworkPoint(N,R,DsnrdB,Petarget)
```

This function returns the standard SNR of Shannon's limit (`snrShannondB`), along with the capacity gap (`CapGapdB`) of the Modulo-SK for its specific inputs. Its inputs are the number or iterations N (`N`), ΔSNR in dB (`DsnrdB`), and the target error probability p_e (`Petarget`). Using the inputs and outputs of `calcSNRworkPoint` it is possible to use the simulation function:

```
function [BER,ffPower,fbPower] = ModuloSKenv(nbits,N,R,snrdB,DsnrdB,Petarget)
```

whose inputs are

- `nbits` the overall number of bits to be transmitted
- `N` the number of interaction rounds. Setting `N=1` represents uncoded PAM (without using the feedback).
- `snrdB` - SNR in dB
- `DsnrdB` - ΔSNR in dB. Setting `DsnrdB=inf` represents noiseless feedback and will activate the classic SK scheme.
- `Petarget` - the target error probability p_e Note that this is an upper bound to the actual simulated BER.

and whose outputs are

- `BER` - the simulated BER
- `ffPower` - a vector containing the average feedforward power per-iterations. Used in order to validate that the respective power constrains is not exceeded.
- `fbPower` - a vector containing the average feedback power per-iterations. Used in order to validate that the respective power constrains is not exceeded.

Examples for using these two functions are given in the file `runModuloSK.m`.

A. Plotting the figures

The environment contains three Matlab scripts which simulate the working points of the figures in the Deepcode paper [5], using the Modulo-SK analysis and simulation. The scripts are:

```
DrawFig2Left.m
DrawFig5Middle.m
```

`DrawFig5Left.m`

It is important to note that unlike in standard BER vs. SNR curves which usually apply the same communication system for all SNR points, the current scripts design a different system for every working points. This is done as follows:

- The rate is always fixed to $R = 1/3$.
- The number of iterations N is fixed for every curve.
- For every point, the graphs corresponds to a different combination of SNR and $\Delta\mathrm{SNR}$. If the feedback signal-to-noise ratio, $\widetilde{\mathrm{SNR}}$, is fixed then $\Delta\mathrm{SNR}_{\mathrm{dB}} = \widetilde{\mathrm{SNR}}_{\mathrm{dB}} - \mathrm{SNR}_{\mathrm{dB}}$.
- For every combination signal-to-noise ratios, R and N, a target error probability p_e is searched by a grid-search using `calcSNRworkPoint.m` which calculated the analytical performance bounds from [2].
- The overall setting is simulated, and the actual BER, along with the average transmitted power are calculated.
- `nPAMsyms` is the number of PAM symbols simulated, which is equal to the number of simulated instances of the Modulo-SK system.

Note: Throughout the analysis, p_e represents the average PAM *symbol* error probability which upper bounds the simulated bit-error-probability (BER). Note that the number of bits in every PAM symbol is $N \cdot R$, hence the overall number of simulated bits is `nPAMsyms*R*N`.

References

[1] J. P. M. Schalkwijk, "A coding scheme for additive noise channels with feedback part II: Band-limited signals," *IEEE Trans. Inf. Theory*, vol. IT-12, pp. 183–189, Apr 1966.

[2] A. Ben-Yishai and O. Shayevitz, "Interactive schemes for the AWGN channel with noisy feedback," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2409–2427, 2017.

[3] A. El Gamal and Y.-H. Kim, *Network information theory*. Cambridge University Press, 2011.

[4] A. Ben-Yishai and O. Shayevitz, "The Gaussian Channel with Noisy Feedback: Near-Capacity Performance via Simple Interaction," in *Proc. 52nd Allerton Conf. Communication, Control Computing*, Oct. 2014, pp. 152–159.

[5] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," 2018. [Online]. Available: https://arxiv.org/abs/1807.00801