EESM-log-SGN - Quick Start Guide

Author: Sian Jin, sianjin@uw.edu

This document describes the MATLAB implementation of EESM-log-SGN method, and its low-storage complexity versions: EESM-log-SGN-LSC and EESM-log-SGN mixture model. The principles of these methods are shown in our IEEE TCOM paper: “Efficient PHY Layer Abstraction for Fast Simulations in Complex System Environments”.

Prerequisites:
MATLAB 2020b (or later version)
MATLAB WLAN toolbox in MATLAB 2020b (or later version)

MATLAB code:
https://github.com/sianjin/EESM-log-SGN

Introduction to the principles of EESM-log-SGN:
The log-SGN approach is based on traditional L2S mapping. We overview traditional L2S mapping as follows. As shown in the figure below, the traditional L2S mapping requires implementing the channel model within the network simulator to generate channel matrices at runtime. The channel matrix as well as the related MIMO precoding matrix and MIMO decoding matrix are used to calculate a post-equalization (post-MIMO-processing) SNR matrix that is mapped into a single metric effective SNR. The effective SNR summarizes all important link design features including channel frequency-selectivity, antenna correlation, the effect of MIMO precoding and decoding, modulation and coding, etc. Instantaneous PER can be predicted from the effective SNR using a PER-SNR lookup table under Additive White Gaussian Noise (AWGN) SISO channel with low complexity. We find, in Table 1, that this approach did not scale well to account for the increased dimensionality of more modern systems with higher MIMO dimensionality and channel bandwidth. In particular, the channel instance generation and the post-equalization SNR matrix generation steps that involve expensive matrix computations become prohibitively runtime expensive for a packet-level simulator. This motivates us to propose a new error model with low computational complexity for ns-3.

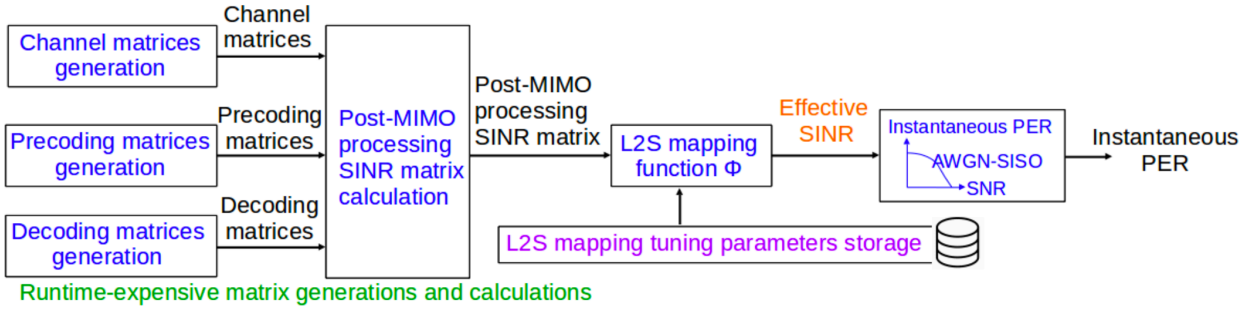

Fig. 1 Traditional L2S mapping principle

The Log-SGN approach herein considers Independent and Identically Distributed (i.i.d.) frequency-selective MIMO channel and is based on our discovery that the effective SNR

distribution under Exponential Effective SNR Mapping (EESM) L2S mapping can be well approximated by a 4-parameter distribution called *log-SGN distribution*. Therefore, under EESM L2S mapping, we only need to store 4 log-SGN parameters under a specific PHY layer setup (e.g., channel type, MIMO dimension, MCS, etc.) and draw effective SNR samples from this specific log-SGN distribution *directly*. The flow chart of Log-SGN is shown below.

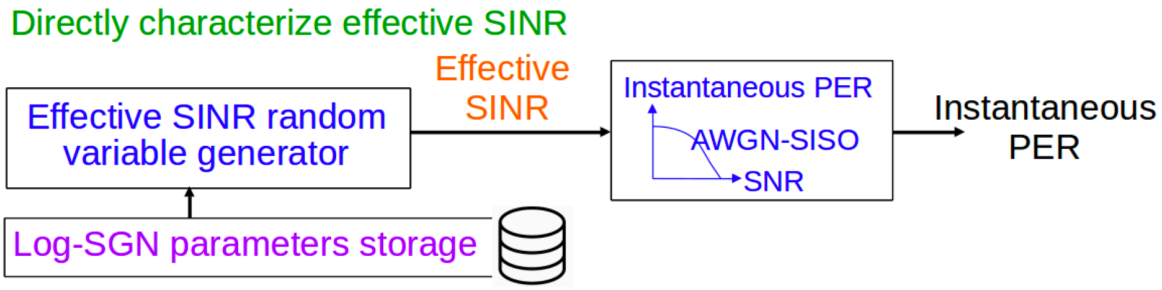

Fig. 2 Basic EESM-log-SGN principle

There are two storage-complexity issues related to the basic EESM-log-SGN. First, it's impossible to store log-SGN parameters for all interference scenarios. Based on our observation that the effective SNR and effective INR can be stored efficiently, as we will illustrate later, we propose a Low-Storage-Complexity (LSC) version of EESM-log-SGN called EESM-log-SGN-LSC, which estimates effective SINR using effective SNR and effective INR. The LSC effective SINR estimator is given by

$$\hat{\Gamma}_{eff,k}^{sinr} = \frac{\Gamma_{eff,k}^{snr}}{1+\theta\sum_{v\in\mathcal{V}}\Gamma_{eff,k}^{inr,v}}$$

where theta is the interference tuning parameter. The principle of EESM-log-SGN-LSC is indicated in Fig. 3.

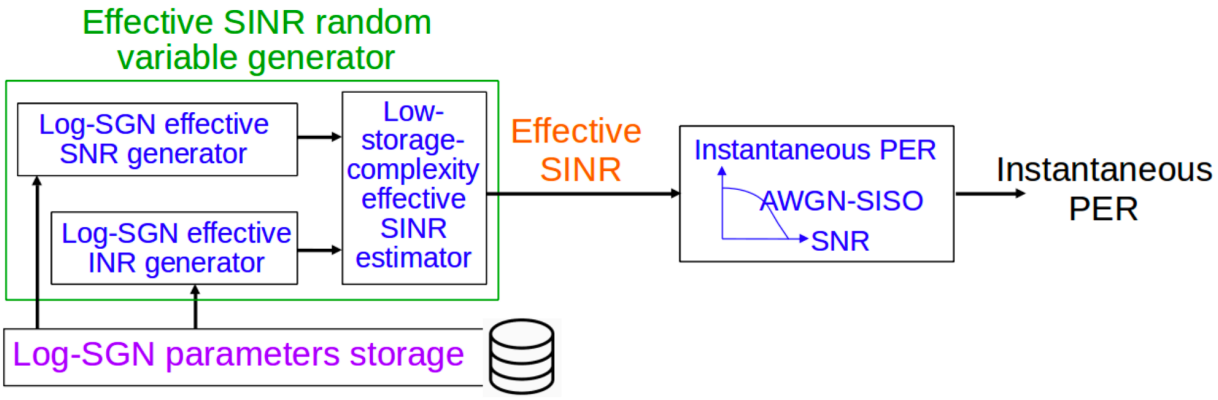

Fig. 3 EESM-log-SGN-LSC principle

Second, log-SGN parameters are sensitive to the change of the received (RX) SNR. In our TCOM paper, we propose to use the mixture model indicated in the following algorithm to output the realization of effective SNR. Using this algorithm, we can generate effective SNR random variable for any RX SNR.

Algorithm 3 Generate Effective SNR Random Variable for Any RX SNR $\gamma \in [\gamma_{min}, \gamma_{max}]$

Input: The RX SNR $\gamma \in [\gamma_{min}, \gamma_{max}]$ at receiver k.
Output: A realization of $\Gamma_{eff,k}^{snr}$.

1: Obtain RX SNR γ's minimum interval $\gamma \in [\gamma_1, \gamma_2]$, where the log-SGN parameters of RX SNRs γ_1 and γ_2 are stored.
2: Generate a realization of uniform random variable $u \sim \text{unif}(0, 1)$.
3: Calculate $\xi = \frac{\gamma - \gamma_1}{\gamma_2 - \gamma_1}$.
4: **if** $u < 1 - \xi$ **then**
5: Generate a realization of $\Gamma_{eff,k}^{snr}$ from the log-SGN PDF $f(\Gamma_{eff,k}^{snr}; \gamma_1)$;
6: **else**
7: Generate a realization of $\Gamma_{eff,k}^{snr}$ from the log-SGN PDF $f(\Gamma_{eff,k}^{snr}; \gamma_2)$.
8: **end if**

Introduction to EESM-log-SGN implementation workflow:
The implementation flow of EESM-log-SGN is shown in Fig. 4, which consists of an offline link simulation part using MATLAB and the actual network simulation part using a network simulator. The offline link simulation part generates log-SGN parameters for use at runtime in network simulations. The offline simulation implements a full PHY using MATLAB, and outputs post-MIMO processing SINR matrix as well as its associated binary packet error state. Using multiple full PHY simulation outputs, the EESM tuning parameter optimization block generates the optimized EESM tuning parameter. Using the optimized EESM tuning parameter and multiple post-MIMO processing SINR matrices, we obtain corresponding effective SINR and statistically fit the effective SINR histogram with the log-SGN distribution introduced. The network simulation part in Fig. 4 uses the EESM-log-SGN method to directly draw the instantaneous PER. The basic EESM-log-SGN method is introduced in Fig. 2 while the EESM-log-SGN-LSC method is introduced in Fig. 3.

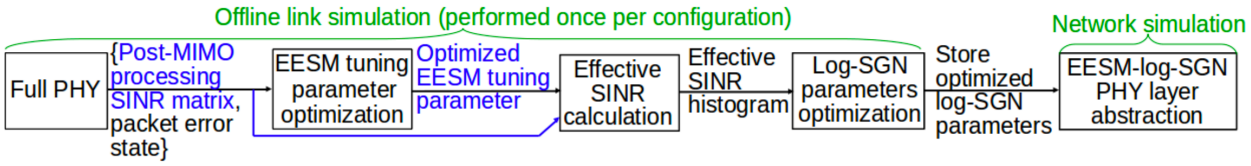

Fig. 4 EESM-log-SGN implementation flow

In our MATLAB code, the full PHY simulation code locates in subfolders with full PHY in the name. After running the full PHY simulation code, the full PHY simulation result in copied into the EESM parameter optimization folder. Depending on the PHY layer abstraction models, users can choose different subfolders. The basic EESM-log-SGN model locates in “MU code basic/3 Basic EESM-log-SGN method”; the mixture model locates in “MU code basic/4 EESM-log-SGN mixture”; the EESM-log-SGN-LSC model locates in “MU code LSC/3 EESM-log-SGN-LSC interference parameter optimization”. The full information of folders and subfolders are listed in the following table. The step-by-step examples of using the code in these folders are shown in MATLAB Example 1 to MATLAB Example 4.

Folders and subfolders in our MATLAB code:

Folder and subfolders	Description
MU code basic	Including files for implementing basic EESM-log-SGN model and EESM-log-SGN mixture model
MU code basic/1 Full PHY (no interferer)	Full PHY simulationwithout interferers
MU code basic/1 Full PHY (multi interferers)	Full PHY simulation under multiple interferers
MU code basic/2 EESM parameter optimization	Optimizing EESM parameter beta shown in Fig. 6 of our IEEE TCOM paper
MU code basic/3 Basic EESM-log-SGN method	Optimizing EESM-log-SGN parameter shown in Algorithm 1 of our IEEE TCOM paper and implementing the basic EESM-log-SGN shown in Fig. 2 of our IEEE TCOM paper
MU code basic/4 EESM-log-SGN mixture	Implementing EESM-log-SGN mixture model shown in Algorithm 3 of our IEEE TCOM paper
MU code basic/5 Traditional EESM (no interferer)	Implementing traditional EESM PHY layer abstraction shown in Fig. 1 of our IEEE TCOM paper
MU code LSC	Including files for implementing EESM-log-SGN-LSC model
MU code LSC/1 Full PHY (for effective SNR)	Full PHY simulation for the desired TX-desired RX pair for effective SNR estimation in EESM-log-SGN-LSC shown in Fig. 11 of our IEEE TCOM paper
MU code LSC/1 Full PHY (for effective INR)	Full PHY simulation for the interferer-desired RX pair for effective INR estimation in EESM-log-SGN-LSC shown in Fig. 11 of our IEEE TCOM paper
MU code LSC/2 EESM parameter optimization	Optimizing EESM parameter beta shown in Fig. 6 of our IEEE TCOM paper

MU code LSC/3 EESM-log-SGN-LSC interference parameter optimization	Optimizing EESM-log-SGN-LSC interference parameter theta shown in equation (6) of our IEEE TCOM paper
MU code LSC/4 EESM-log-SGN-LSC method	Implementing EESM-log-SGN-LSC estimator shown in Fig. 11 of our IEEE TCOM paper

Step-by-step examples:
4 MATLAB simulation example files are provided to show how to implement the basic EESM-log-SGN, EESM-log-SGN mixture model, and EESM-log-SGN-LSC **in a step-by-step manner**.

Simulation example files	Description
MATLAB Example 1: Reproducing Fig. 10 in our IEEE TCOM paper	Learn how to implement and use the **basic EESM-log-SGN** shown in Fig. 2 in our IEEE TCOM paper by reproducing Fig. 10 in our IEEE TCOM paper.
MATLAB Example 2: Reproducing Fig. 13 in our IEEE TCOM paper	Learn how to implement and use the **EESM-log-SGN mixture model** shown in Algorithm 3 of our IEEE TCOM paper by reproducing Fig. 13 in our IEEE TCOM paper.
MATLAB Example 3: Reproducing Fig. 12(a) in our IEEE TCOM paper	Learn how to implement and use the **EESM-log-SGN-LSC** shown in Fig. 11 of our IEEE TCOM paper by reproducing Fig. 12(a) in our IEEE TCOM paper.
MATLAB Example 4: Reproducing Table III in our IEEE TCOM paper	Learn how to implement **traditional EESM PHY layer abstraction** shown in Fig. 1 of our IEEE TCOM paper, and observe the runtime scaling issue caused by traditional EESM PHY layer abstraction.