Matlab Simulation Example 2: PHY abstraction under 11ax OFDM SISO system

Matlab code part is colored in orange.

Setup:
MCS = 4
num of transmit antenna = 1, num of receive antenna = 1
Channel: Model-D, bandwidth = **40MHz**
APEP length = 1000
Channel coding = LDPC

Step 1: running full PHY simulation
1.1 Set the above parameters in 1 full PHY/fullPHY.m:

```
mcs = [4]; % Vector of MCS to simulate between 0 and 9
numTxRx = [1 1]; % Matrix of MIMO schemes, each row is [numTx numRx]
chan = "Model-D"; % String array of delay profiles to simulate
% maxnumberrors = 40*1e3;  % The maximum number of packet errors at an SNR point
% maxNumPackets = 40*1e3; % The maximum number of packets at an SNR point
maxnumberrors = 1e1;  % The maximum number of packet errors at an SNR point
maxNumPackets = 1e2; % The maximum number of packets at an SNR point

% Fixed PHY configuration for all simulations
cfgHE = wlanHESUConfig;
cfgHE.ChannelBandwidth = 'CBW40'; % Channel bandwidth
bandwidth = cfgHE.ChannelBandwidth;
cfgHE.APEPLength = 1000;          % Payload length in bytes
cfgHE.ChannelCoding = 'LDPC';     % Channel coding
```

Using parfor to speed up the simulation (this require installing Matlab parallel computing toolbox):

```
% Simulate each configuration
results = cell(1,numel(simParams));
parfor isim = 1:numel(simParams)  % Use 'parfor' to speed up the simulation
   results{isim} = box0Simulation(simParams(isim));
end
```

Run fullPHY.m. This takes a few minus or less.

See if you can produce PER-VS-SNR curve from 1 down to 10^{-2}, as exemplified in Figure 1. If so, move on to step 1.2. If not, change snr for that MCS and channel in getBox0SimParams.m and redo this step.

1.2 set the number of packets to 40000:

```
mcs = [4]; % Vector of MCS to simulate between 0 and 9
numTxRx = [1 1]; % Matrix of MIMO schemes, each row is [numTx numRx]
chan = "Model-D"; % String array of delay profiles to simulate
maxnumberrors = 40*1e3;  % The maximum number of packet errors at an SNR point
maxNumPackets = 40*1e3; % The maximum number of packets at an SNR point
% maxnumberrors = 1e1;  % The maximum number of packet errors at an SNR point
% maxNumPackets = 1e2; % The maximum number of packets at an SNR point
```

```
% Fixed PHY configuration for all simulations
cfgHE = wlanHESUConfig;
cfgHE.ChannelBandwidth = 'CBW40'; % Channel bandwidth
bandwidth = cfgHE.ChannelBandwidth;
cfgHE.APEPLength = 1000;          % Payload length in bytes
cfgHE.ChannelCoding = 'LDPC';     % Channel coding
```

run fullPHY.m. This takes a long time (around a few hours).

1.3 You can see the output:
snrPer_CBW40_Model-D_1-by-1_MCS4.mat
and a PER-VS-SNR figure in Figure 1.
Save this figure in fig format.

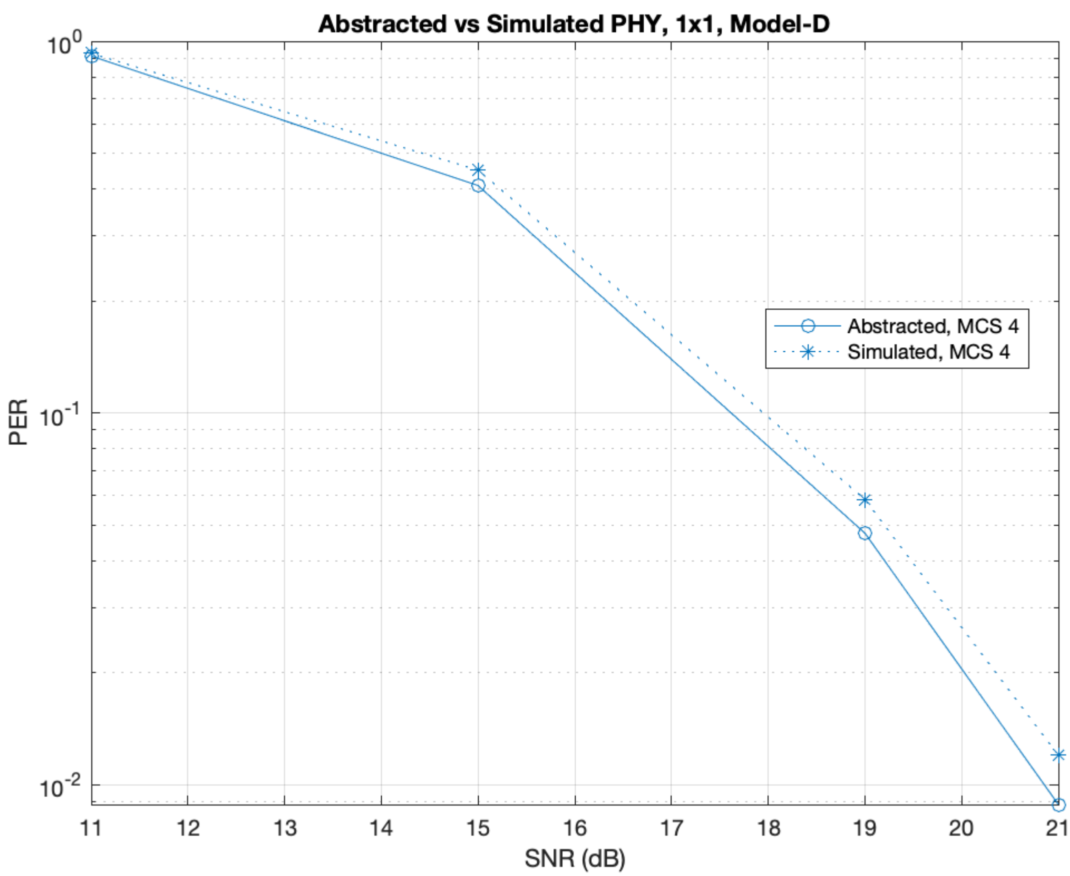

Figure 1. The dotted line is the full-PHY average PER- RXSNR curve under full PHY simulation. Ignore the solid line.

Step 2: optimize EESM parameter beta
2.1 Copy snrPer_CBW40_Model-D_1-by-1_MCS4.mat into the second folder: 2 EESM parameter optimization

2.2 Open eesmAbstractionPerVsEffSnr.m

Correctly load snrPer_CBW40_Model-D_1-by-1_MCS4.mat in eesmAbstractionPerVsEffSnr.m:

load('snrPer_CBW40_Model-D_1-by-1_MCS4.mat');

Randomly choose an initial beta value (usually the larger MCS value, the larger initial beta value):

% Initialize EESM parameters
beta = 8;

2.3 run eesmAbstractionPerVsEffSnr.m

2.4 You can see output
eesmEffSnr_CBW40_Model-D_1-by-1_MCS4.mat
This file includes optimized eesm parameter beta

Step 3: EESM-log-SGN PHY abstraction
3.1 Copy snrPer_CBW40_Model-D_1-by-1_MCS4.mat and eesmEffSnr_CBW40_Model-D_1-by-1_MCS4.mat into the third folder: 3 log-SGN method

3.2 Open skewGeneralizedNormalApp.m
Correctly load snrPer_CBW40_Model-D_1-by-1_MCS4.mat and eesmEffSnr_CBW40_Model-D_1-by-1_MCS4.mat in skewGeneralizedNormalApp.m:

load(‘snrPer_CBW40_Model-D_1-by-1_MCS4.mat');
load('eesmEffSnr_CBW40_Model-D_1-by-1_MCS4.mat')

There are 4 SNR values, stored in the variable snrs.
Set for example:

snrIdx = 1;

This means choose the 1st SNR value:

snrs(1) = 11

This value means the currently tested SNR is 11dB.

3.3 Run skewGeneralizedNormalApp.m. You’ll quickly see Figure 2

If the red curve fits with the blue histogram, then record the log-SGN parameters in the vector xBest in a table in your document (for example, Table A below)

xBest = [1.80977222255655 0.382958831273698 0.627649764888263 0.568670930775513]

These values are the average PER under SNR = 0dB.
Also, record the value in perFastAbs

perFastAbs = 0.9178

into perFastAbsVec in the file absAvgPerVsTxSnr.m.

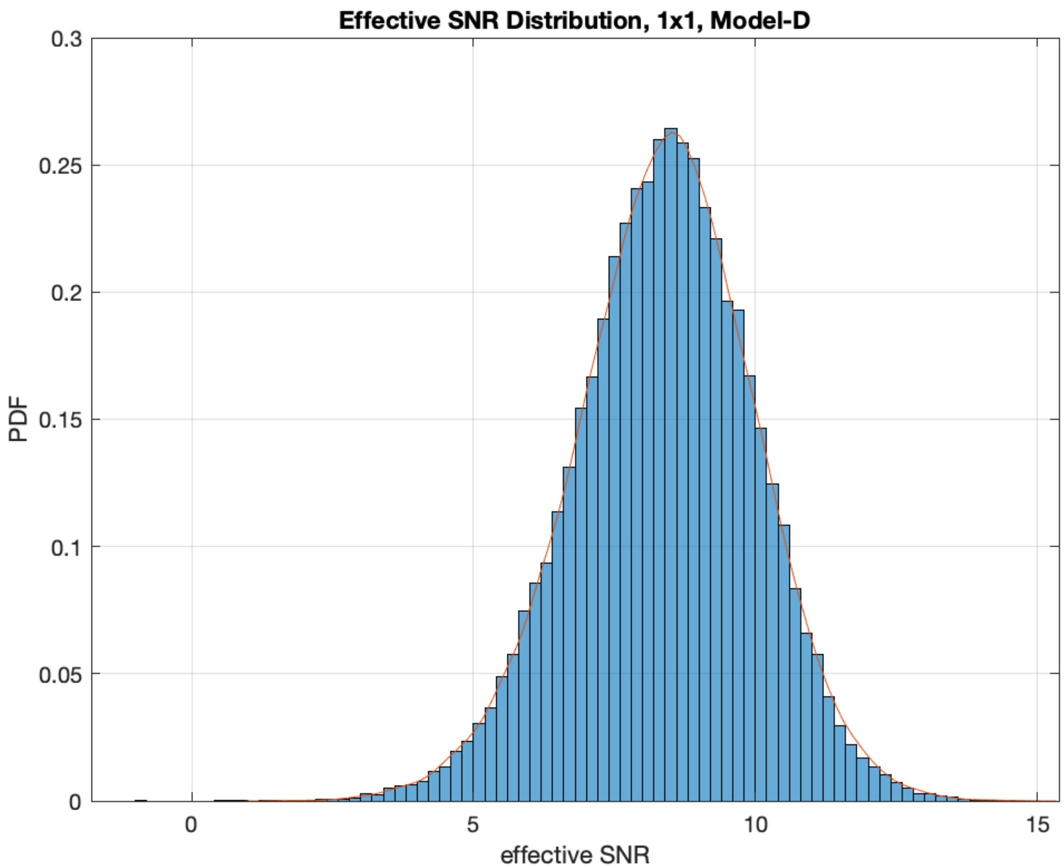

Figure 2. Effective SNR distribution under MCS4, RX SNR = 11dB

This value is the average PER under SNR = 11dB.

3.4 Repeat the process in step 3.2 to 3.3 for different SNR indices.
Fill in the table that records the log-SGN parameters, as exemplified in Table A.
Finish recording average PER under each SNR in the file absAvgPerVsTxSnr.m.

3.5 Run the file absAvgPerVsTxSnr.m
Copy the curve into figure 1, see whether it matches the full PHY simulation result (the dotted line) well.

Finally, you can produce the result in Table A

Table A. Average PER and Log-SGN parameters under 11ax **40MHz** OFDM SU-SISO, **Model-D, MCS4,** with payload length 1000. The

parameters are obtained by running 40000 packet simulations. Under such setup, the optimized EESM parameter beta = 7.1100.

Rx SNR	Full PHY Avg PER	Log-SGN Avg PER	ns-3	mu	sigma	lambda 1	lambda 2
11dB	0.9272	0.9178	0.9127	1.8098	0.3830	0.6276	0.5687
15dB	0.4501	0.4286	0.4171	2.2907	0.3815	1.0077	0.2751
19dB	0.0586	0.0598	0.0573	3.0827	0.3596	-1.3206	1.5074
21dB	0.0120	0.0090	0.0086	2.7503	0.4908	2.4674	0.3490