Matlab Simulation Example 6: EESM-log-SGN-LSC under 11ax OFDM/ OFDMA MIMO/MU-MIMO system

Matlab code part is colored in orange.

Setup:
MCS = 4
num of transmit antenna = 4, num of receive antenna = 2
{106, 8x{2,2}:{2,2}} mixed MU-MIMO OFDMA (allocation index = 97)
Channel: Model-D, bandwidth = 20MHz
APEP length = 1000
1 interference
RX INR = RX SNR -30dB
Channel coding = LDPC

Step 1: running full PHY simulation for mixed desired signal and interference signal

Set the above parameters in 1 full PHY/3 full PHY mixed channles 1Ints/fullPHY.m

```
mcs = [4]; % Vector of MCS to simulate between 0 and 9
numTxRx = [8 2]; % 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

% Fixed PHY configuration for all simulations
cfgHE = wlanHEMUConfig(97);
for userIdx = 1:numel(cfgHE.User)
   cfgHE.User{userIdx}.APEPLength = 1000; % Payload length in bytes
end
```

In box0Simulation1IntUser1.m, set interference power to be 10dB smaller than desired signal transmit power:

intPathloss = 1/10^(30/10); % Interference path loss in linear scale

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

1.3 You can see the output: snrPer_config97_Model-D_8-by-2_MCS4Mix.mat.

Step 2: optimize EESM parameter beta for mixed desired signal and interference signal

2.1 Copy snrPer_config97_Model-D_8-by-2_MCS4Mix.mat.

into the second folder: 2 EESM parameter optimization

2.2 Open eesmAbstractionPerVsEffSnr.m
Correctly load snrPer_config97_Model-D_8-by-2_MCS4Mix.mat in eesmAbstractionPerVsEffSnr.m:

load('snrPer_config97_Model-D_8-by-2_MCS4Mix.mat');

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

% Initialize EESM parameters beta = 7;

2.3 run eesmAbstractionPerVsEffSnr.m

2.4 You can see output eesmEffSnr_Config97_Model-D_8-by-2_MCS4.mat This file includes optimized eesm parameter beta

2.5 rename it into eesmEffSnr_Config97_Model-D_8-by-2_MCS4Mix.mat

Step 3: running full PHY simulation for the desired signal

3.1 Set the above parameters in 1 full PHY/1 full PHY desired channels/fullPHY.m

mcs = [4]; % Vector of MCS to simulate between 0 and 9

```
numTxRx = [8 2]; % 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

% Fixed PHY configuration for all simulations
cfgHE = wlanHEMUConfig(97);
for userIdx = 1:numel(cfgHE.User)
   cfgHE.User{userIdx}.APEPLength = 1000; % Payload length in bytes
end
```

The RX SNRs in getBox0Params are set to be the same as those of Step1.

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

3.3 You can see the output: snrPer_config97_Model-D_8-by-2_MCS4Sig.mat.

Step 4: optimize EESM parameter beta for the desired signal

4.1 Copy snrPer_config97_Model-D_8-by-2_MCS4Sig.mat in step 3 into the second folder: 2 EESM parameter optimization

4.2 Open eesmAbstractionPerVsEffSnr.m
Correctly load snrPer_config97_Model-D_8-by-2_MCS4Sig.mat in eesmAbstractionPerVsEffSnr.m:

```
load('snrPer_config97_Model-D_8-by-2_MCS4Sig.mat');
```

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

```
% Initialize EESM parameters beta = 7;
```

4.3 run eesmAbstractionPerVsEffSnr.m

4.4 You can see output eesmEffSnr_Config97_Model-D_8-by-2_MCS4.mat This file includes optimized eesm parameter beta

Step 5: running full PHY simulation for the desired signal

5.1 Set the above parameters in 1 full PHY/1 full PHY desired channels/fullPHY.m

```
mcs = [4]; % Vector of MCS to simulate between 0 and 9
numTxRx = [8 2]; % 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

% Fixed PHY configuration for all simulations
cfgHE = wlanHEMUConfig(97);
for userIdx = 1:numel(cfgHE.User)
   cfgHE.User{userIdx}.APEPLength = 1000; % Payload length in bytes
end
```

The RX SNRs in getBox0Params are set to be the 30dB lower than those of Step1.

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

5.3 You can see the output: snrPer_config97_Model-D_8-by-2_MCS4Int.mat.

Step 6: optimize EESM parameter beta for the desired signal

6.1 Copy snrPer_config97_Model-D_8-by-2_MCS4Int.mat in step 5 into the second folder: 2 EESM parameter optimization

6.2 Open eesmAbstractionPerVsEffSnr.m
Correctly load snrPer_config97_Model-D_8-by-2_MCS4Int.mat in eesmAbstractionPerVsEffSnr.m:

```
load(‘snrPer_config97_Model-D_8-by-2_MCS4Int.mat');
```

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

% Initialize EESM parameters beta = 7;

6.3 run eesmAbstractionPerVsEffSnr.m

6.4 You can see output eesmEffSnr_Config97_Model-D_8-by-2_MCS4.mat This file includes optimized eesm parameter beta

6.5 rename it into eesmEffSnr_Config97_Model-D_8-by-2_MCS4Int.mat

Step 7: EESM-log-SGN-LSC PHY abstraction

7.1 Copy all the above generated 6 files into the third folder: 3 log-SGN method

7.2 correctly load above 6 files in getOptIntTuningParam1Int.m

7.3 Run getOptIntTuningParam1Int.m and obtain thetaOpt

7.4 Set the value of theta in effSINRModelLogSGNDataDivision1Int.m to be the thetaOpt; correctly load above 6 files in effSINRModelLogSGNDataDivision1Int.m

7.5 Run effSINRModelLogSGNDataDivision1Int.m