Kyle Savidge

Miniproject #2 – EC508

10/19/16

# Code for (a) & (b):

%initialize variables, loop over 10 SNR values

L = 3;

fail = zeros(11, L);

probErrSim = zeros(11, L);

probErrExact = zeros(11,L);

probErrEst = zeros(11,L);

for SNRdb\_1 = 1:11

SNRdb = SNRdb\_1 - 1;

SNR = 10.^(SNRdb/10);

%simulate 10000 trials of sending bit x and getting y at the Rx

for trial = 1:10000

h = 1/sqrt(2) \* randn(L,1) + 1j/sqrt(2) \* randn(L,1);

w = 1/sqrt(2) \* randn(L,1) + 1j/sqrt(2) \* randn(L,1);

x = sqrt(SNR)\*((-1)^randi([0 1]));

y = h \* x + w;

%ML detection, determines whether the projection of magnitude h \* %value of y onto the real line is closer to x or the %complementary BPSK value of x

for l = 1:L

h\_temp = h(1:l);

y\_temp = y(1:l);

decider = real(((h\_temp)'\*y\_temp)/norm(h\_temp));

if (abs(decider - x) > abs(decider - (-1 \* x)))

fail(SNRdb\_1, l) = fail(SNRdb\_1, l) + 1;

end

%calculate the error rate

probErrSim(SNRdb\_1, l) = (fail(SNRdb\_1,l)/trial);

end

end

%exact error probablity for each L

mu = sqrt(SNR/(1 + SNR));

for l\_temp = 1:L

sum = 0; % reset the summation over l for each L

for l = 0:(l\_temp - 1)

sum = sum + (((1 - mu)/2))^l\_temp \* nchoosek(l\_temp - 1... + l, l) \* ((1 + mu)/2)^l;

probErrExact(SNRdb\_1,l\_temp) = sum;

end

end

%estimated error probability for each L

for l = 1:L

probErrEst(SNRdb\_1, l) = nchoosek(2\*l - 1, l)\*(1/((4 \* SNR)^l));

end

end

SNRdb = transpose(0:10);

semilogy(SNRdb,probErrSim);

hold on

semilogy(SNRdb,probErrExact);

semilogy(SNRdb,probErrEst);

hold off

legend('Simulated Error, L = 1', 'Simulated Error, L = 2', 'Simulated Error, L = 3', ...

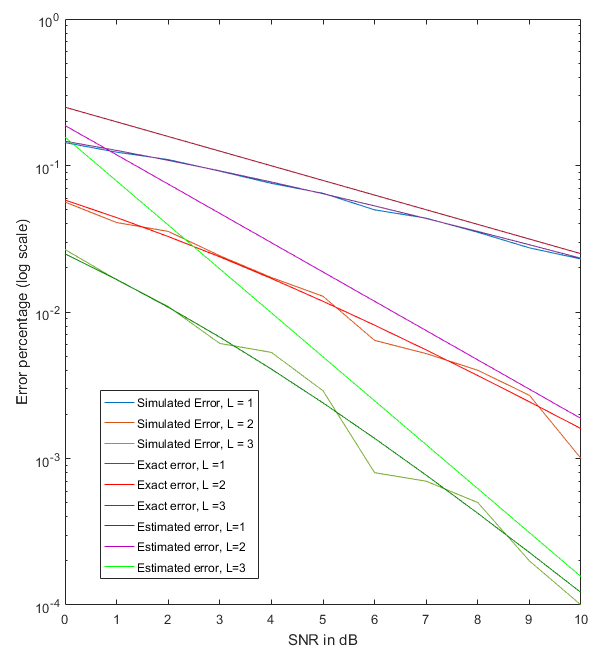
'Exact error, L =1', 'Exact error, L =2', 'Exact error, L =3', ...

'Estimated error, L=1', 'Estimated error, L=2', 'Estimated error, L=3');

xlabel('SNR in dB')

ylabel('Error percentage (log scale)')

# Plot from (a) & (b):



# Part (c):

%initialize variables, loop over 20 M\_train values

L = 3;

M\_train = 20;

M\_c = 100;

w = zeros(M\_train, L);

hhat\_l = zeros(L);

fail = zeros(M\_train, L);

probErrSim = zeros(M\_train, L);

probErrExact = zeros(M\_train,L);

SNRdb = 0;

SNR = 10.^(SNRdb/10);

for m\_temp = 1:M\_train

%simulate 1000 trials of sending bit x and getting y at the Rx

for trial = 1:10000

%generate h\_l

h = 1/sqrt(2) \* randn(L,1) + 1j/sqrt(2) \* randn(L,1);

%generate M\_train noise values

w = 1/sqrt(2) \* randn(L,M\_train) + 1j/sqrt(2) \* randn(L,M\_train);

%calculate h(hat)\_l using MMSE

sum = 0;

for m = 1:m\_temp

sum = sum + (sqrt(SNR)/(1 + M\_train \* sqrt(SNR))\*(h \* sqrt(SNR) + w(:,m)));

end

hhat\_l = sum;

%send x over the channel during the remaining time on the coherence

%interval

x = sqrt(SNR)\*((-1)^randi([0 1]));

w = 1/sqrt(2) \* randn(L,1) + 1j/sqrt(2) \* randn(L,1);

y = h \* x + w;

%ML detection, determines whether the projection of magnitude h \* value of y

%onto the real line is closer to x or the complementary BPSK value of x

for l = 1:L

h\_temp = hhat\_l(1:l);

y\_temp = y(1:l);

decider = real(((h\_temp')\*y\_temp)/norm(h\_temp));

if (abs(decider - x) > abs(decider - (-1 \* x)))

fail(m\_temp, l) = fail(m\_temp, l) + 1;

end

%calculate the error rate

probErrSim(m\_temp, l) = (fail(m\_temp,l)/trial);

end

end

%exact error probablity for each L

mu = sqrt(SNR/(1 + SNR));

for l\_temp = 1:L

sum = 0; % reset the summation over l for each L

for l = 0:(l\_temp - 1)

sum = sum + (((1 - mu)/2))^l\_temp \* nchoosek(l\_temp - 1 + l, l) \* ((1 + mu)/2)^l;

probErrExact(m\_temp,l\_temp) = sum;

end

end

end

x\_axis = transpose(1:M\_train);

semilogy(x\_axis,probErrSim);

hold on

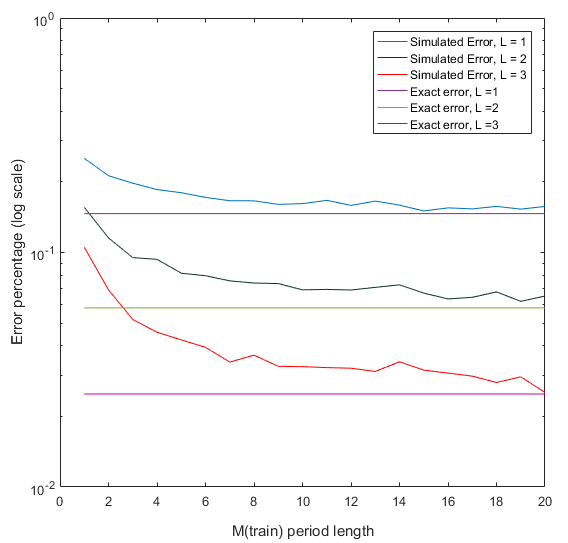
semilogy(x\_axis,probErrExact);

hold off

legend('Simulated Error, L = 1', 'Simulated Error, L = 2', 'Simulated Error, L = 3', ...

'Exact error, L =1', 'Exact error, L =2', 'Exact error, L =3');

xlabel('M\_train period length')

ylabel('Error percentage (log scale)')

%initialize variables, loop over 10 SNR values

# Part (d):

L = 3;

M\_train = 5;

M\_c = 100;

w = zeros(M\_train, L);

hhat\_l = zeros(L);

fail = zeros(11, L);

probErrSim = zeros(11, L);

probErrExact = zeros(11,L);

probErrEst = zeros(11,L);

for SNRdb\_1 = 1:11

SNRdb = SNRdb\_1 - 1;

SNR = 10.^(SNRdb/10);

%simulate 1000 trials of sending bit x and getting y at the Rx

for trial = 1:5000

%generate h\_l

h = 1/sqrt(2) \* randn(L,1) + 1j/sqrt(2) \* randn(L,1);

%generate M\_train noise values

w = 1/sqrt(2) \* randn(L,M\_train) + 1j/sqrt(2) \* randn(L,M\_train);

%calculate h(hat)\_l using MMSE

sum = 0;

for m = 1:M\_train

sum = sum + (sqrt(SNR)/(1 + M\_train \* sqrt(SNR))\*(h \* sqrt(SNR) + w(:,m)));

end

hhat\_l = sum;

%send x over the channel during the remaining time on the coherence

%interval

x = sqrt(SNR)\*((-1)^randi([0 1]));

w = 1/sqrt(2) \* randn(L,1) + 1j/sqrt(2) \* randn(L,1);

y = h \* x + w;

%ML detection, determines whether the projection of magnitude h(hat) \* value of y

%onto the real line is closer to x or the complementary BPSK value of x

for l = 1:L

h\_temp = hhat\_l(1:l);

y\_temp = y(1:l);

decider = real(((h\_temp')\*y\_temp)/norm(h\_temp));

if (abs(decider - x) > abs(decider - (-1 \* x)))

fail(SNRdb\_1, l) = fail(SNRdb\_1, l) + 1;

end

%calculate the error rate

probErrSim(SNRdb\_1, l) = (fail(SNRdb\_1,l)/trial);

end

end

%exact error probablity for each L

mu = sqrt(SNR/(1 + SNR));

for l\_temp = 1:L

sum = 0; % reset the summation over l for each L

for l = 0:(l\_temp - 1)

sum = sum + (((1 - mu)/2))^l\_temp \* nchoosek(l\_temp - 1 + l, l) \* ((1 + mu)/2)^l;

probErrExact(SNRdb\_1,l\_temp) = sum;

end

end

end

SNRdb = transpose(0:10);

semilogy(SNRdb,probErrSim);

hold on

semilogy(SNRdb,probErrExact);

hold off

legend('Simulated Error, L = 1', 'Simulated Error, L = 2', 'Simulated Error, L = 3', ...

'Exact error, L =1', 'Exact error, L =2', 'Exact error, L =3');

xlabel('SNR in dB')

ylabel('Error percentage (log scale)')

