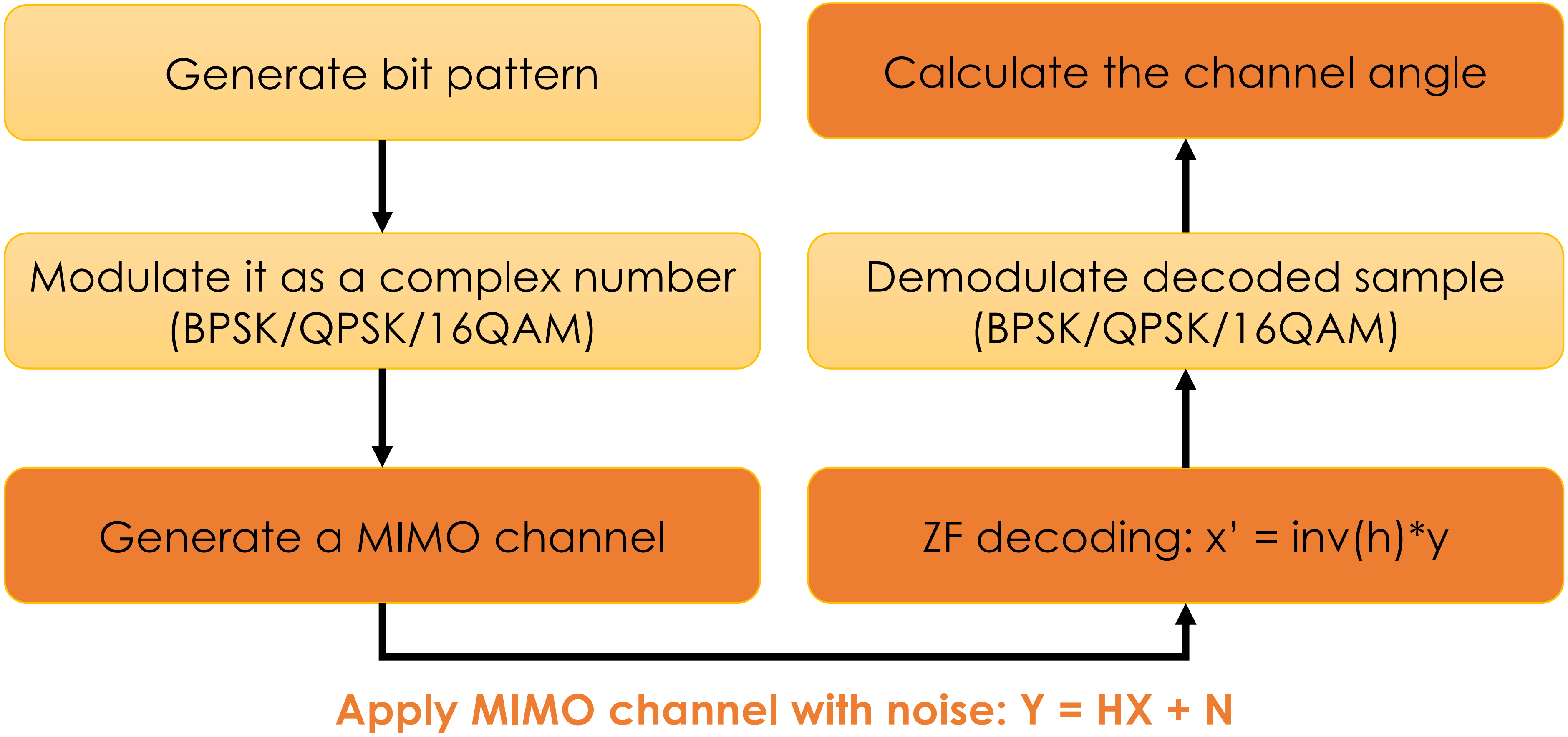
Network Systems Capstone @CS.NYCU

Lab7: MIMO ZF Equalization

Example of Wireless Transmission

`pre_lab7_mod.m`


Generate bit pattern
Modulate it as a complex number (BPSK/QPSK/16QAM)
Generate a MIMO channel
ZF decoding: x’ = inv(h)*y
Demodulate decoded sample (BPSK/QPSK/16QAM)
Calculate the channel angle
Apply MIMO channel with noise: Y = HX + N

Example Code

1. Generate a sequence of data bits
2. Modulate the bits into BPSK samples
3. Generate 2x2 MIMO channel
4. Analyze channel correlation
5. Simulate the reception over the channel with AWGN
   - Y = HX + N
6. ZF decoding
   - X' = inv(H)Y
7. SNR and BER calculation
8. Analyze SNR and compare them based on the channel correlation

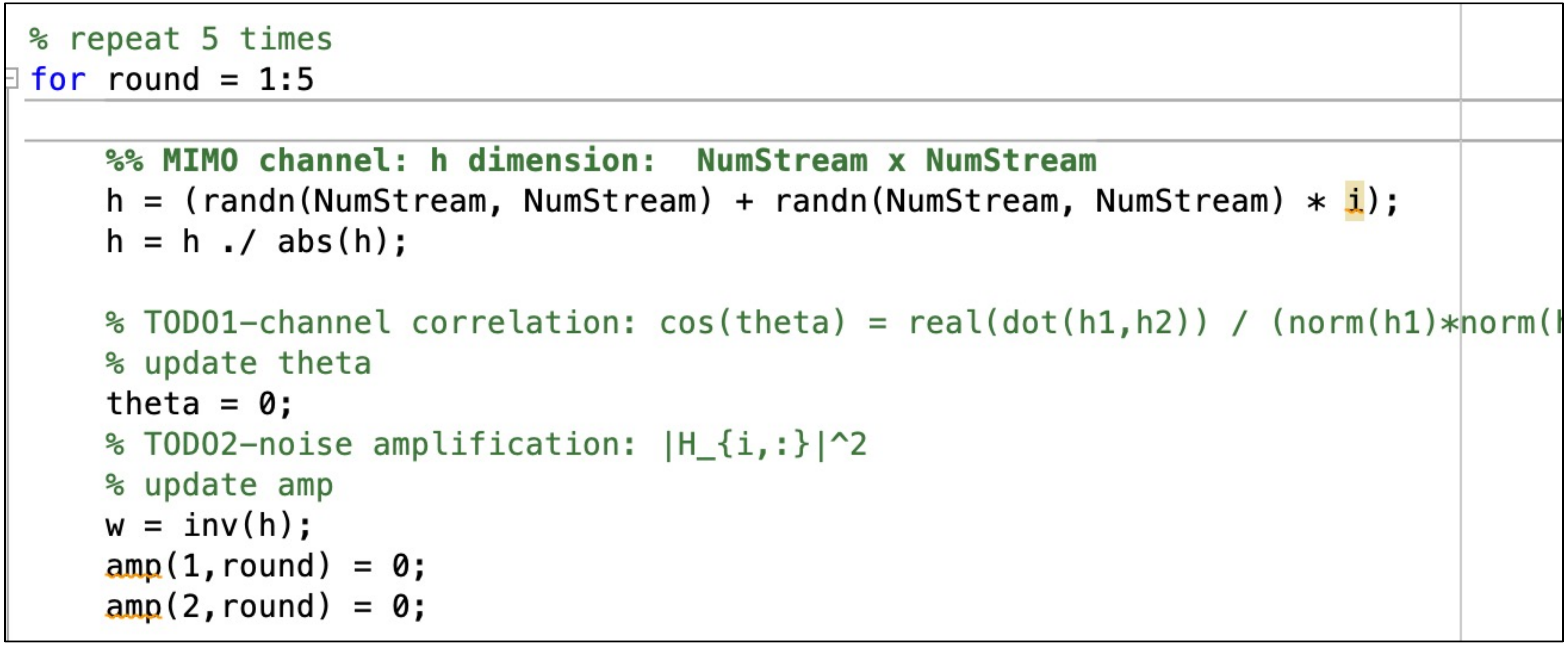
Example Code

- Search "**MIMO**" to check what have been modified in the example code for MIMO transmissions

- Search "**TODO**" to check what you should modify

Snapshot of Example Code

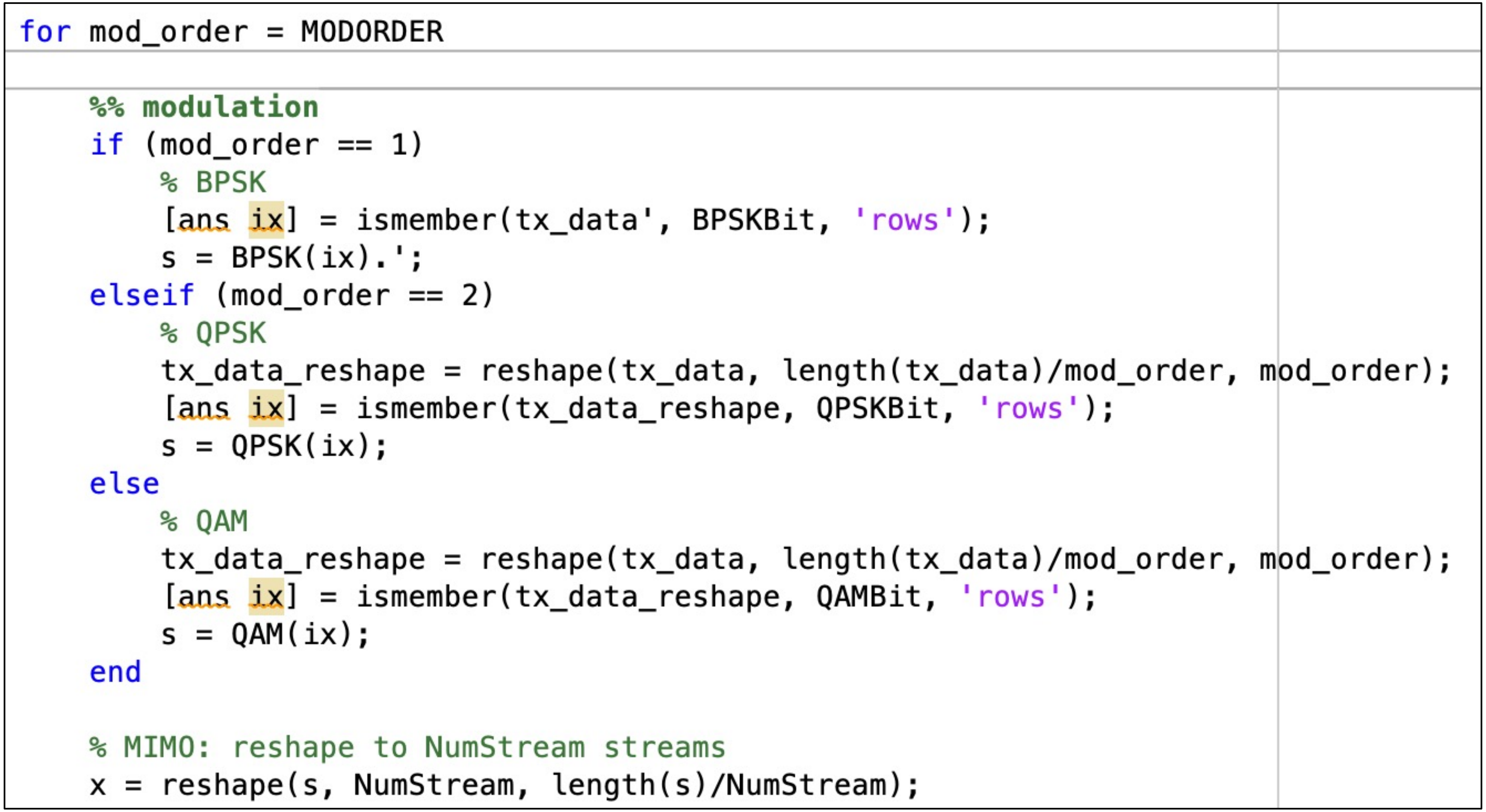
- lab7-mimo(): Channel generation



```
% repeat 5 times
for round = 1:5

    %% MIMO channel: h dimension:  NumStream x NumStream
    h = (randn(NumStream, NumStream) + randn(NumStream, NumStream) * i);
    h = h ./ abs(h);

    % TODO1-channel correlation: cos(theta) = real(dot(h1,h2)) / (norm(h1)*norm(
    % update theta
    theta = 0;
    % TODO2-noise amplification: |H_{i,:}|^2
    % update amp
    w = inv(h);
    amp(1,round) = 0;
    amp(2,round) = 0;
```

Snapshot of Example Code

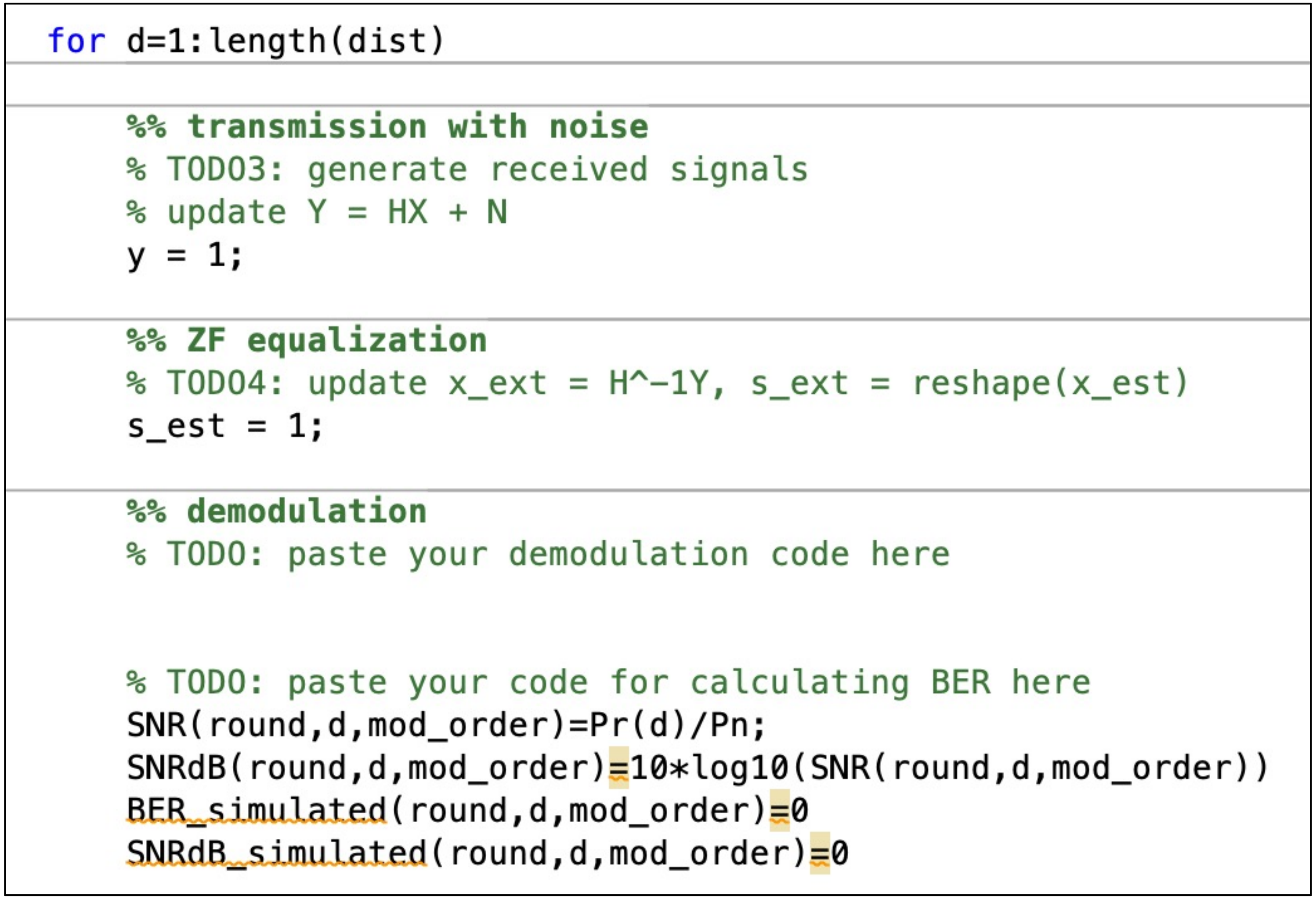
- lab7-mimo(): Modulation



```
for mod_order = MODORDER

    %% modulation
    if (mod_order == 1)
        % BPSK
        [ans ix] = ismember(tx_data', BPSKBit, 'rows');
        s = BPSK(ix).';
    elseif (mod_order == 2)
        % QPSK
        tx_data_reshape = reshape(tx_data, length(tx_data)/mod_order, mod_order);
        [ans ix] = ismember(tx_data_reshape, QPSKBit, 'rows');
        s = QPSK(ix);
    else
        % QAM
        tx_data_reshape = reshape(tx_data, length(tx_data)/mod_order, mod_order);
        [ans ix] = ismember(tx_data_reshape, QAMBit, 'rows');
        s = QAM(ix);
    end

    % MIMO: reshape to NumStream streams
    x = reshape(s, NumStream, length(s)/NumStream);
```

Snapshot of Example Code

- lab7-mimo(): Transmission and ZF decoding

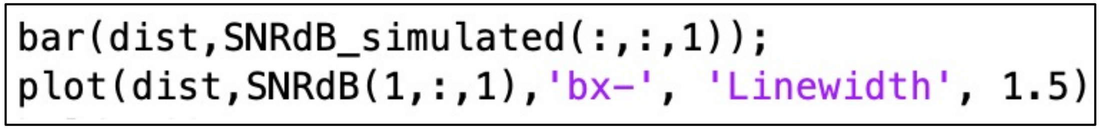


```
for d=1:length(dist)

    %% transmission with noise
    % TODO3: generate received signals
    % update Y = HX + N
    y = 1;

    %% ZF equalization
    % TODO4: update x_ext = H^-1Y, s_ext = reshape(x_est)
    s_est = 1;

    %% demodulation
    % TODO: paste your demodulation code here


    % TODO: paste your code for calculating BER here
    SNR(round,d,mod_order)=Pr(d)/Pn;
    SNRdB(round,d,mod_order)=10*log10(SNR(round,d,mod_order))
    BER_simulated(round,d,mod_order)=0
    SNRdB_simulated(round,d,mod_order)=0
```

Snapshot of Example Code

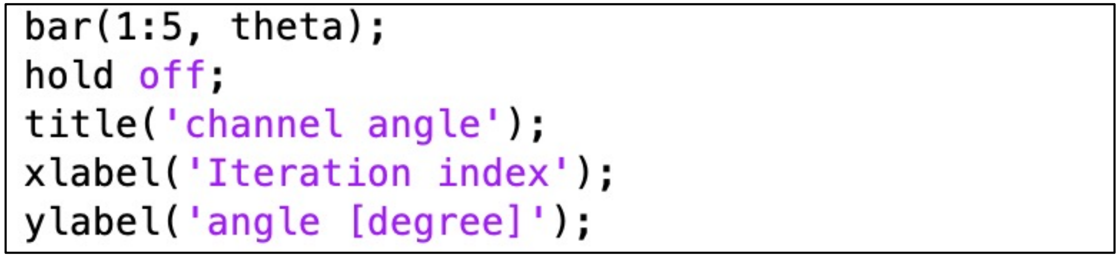
- lab7-mimo(): Plot figures

SNR



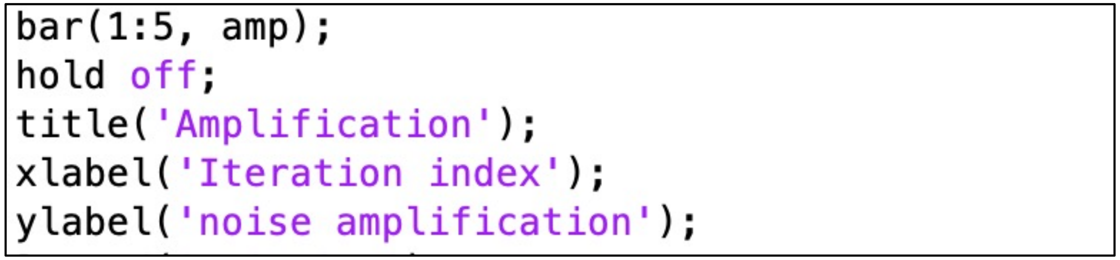
```
bar(dist,SNRdB_simulated(:,:,1));
plot(dist,SNRdB(1,:,1),'bx-', 'Linewidth', 1.5)
```

angle



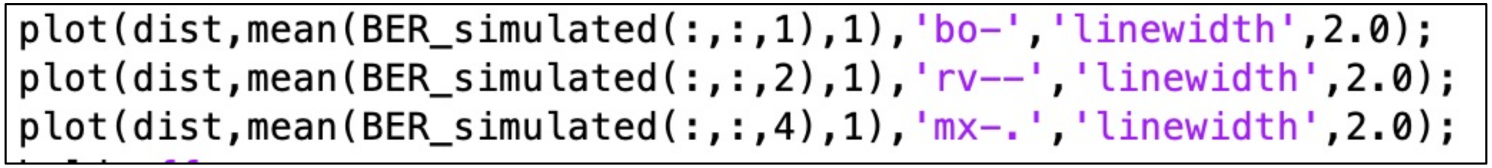
```
bar(1:5, theta);
hold off;
title('channel angle');
xlabel('Iteration index');
ylabel('angle [degree]');
```

amp



```
bar(1:5, amp);
hold off;
title('Amplification');
xlabel('Iteration index');
ylabel('noise amplification');
```

BER



```
plot(dist,mean(BER_simulated(:,:,1),1),'bo-','linewidth',2.0);
plot(dist,mean(BER_simulated(:,:,2),1),'rv--','linewidth',2.0);
plot(dist,mean(BER_simulated(:,:,4),1),'mx-.','linewidth',2.0);
```

TODO – Pre-Lab7-mod

Input and Output

- Input
  - Link distances: 100m ~ 400m
  - Tx Power: 10dBm
  - Noise Power: -85dBm
- Output
  - SNR, BER
  - Plot the figures
    - BER bar graph (x-axis: distances, y-axis: BER)
    - Channel angle bar graph
    - Power of inv(H) bar graph
    - SNR bar graph (x-axis: distances, y-axis: SNR)

TODO

Given a link distance and1,000 random samples

1. Calculate the path loss and drive the receiving power
2. Modulate the bit stream `tx_data` to `x` using BPSK/QPSK/16QAM
3. [TODO1] Calculate channel correlation (15%)
4. [TODO2] Calculate noise amplification (15%)
5. [TODO3] Calculate the received MIMO sample y (15%)
6. [TODO4] Decode the received sample x’ using ZF (15%)
7. Demodulate `x’` back to `rx_data`
8. [TODO5] Plot figures and analyze based on channel correlation in the report (40%)

How to Calculate Channel Angle?

$$\cos(\theta) = \frac{|real(\vec{h}_1^* \cdot \vec{h}_2)|}{||\vec{h}_1||||\vec{h}_2||}$$

$$\theta = \arccos(\cos(\theta))/\pi * 180$$

Angle should be in between 0-90 degrees

Map 90-180 degrees to 0-180 degrees

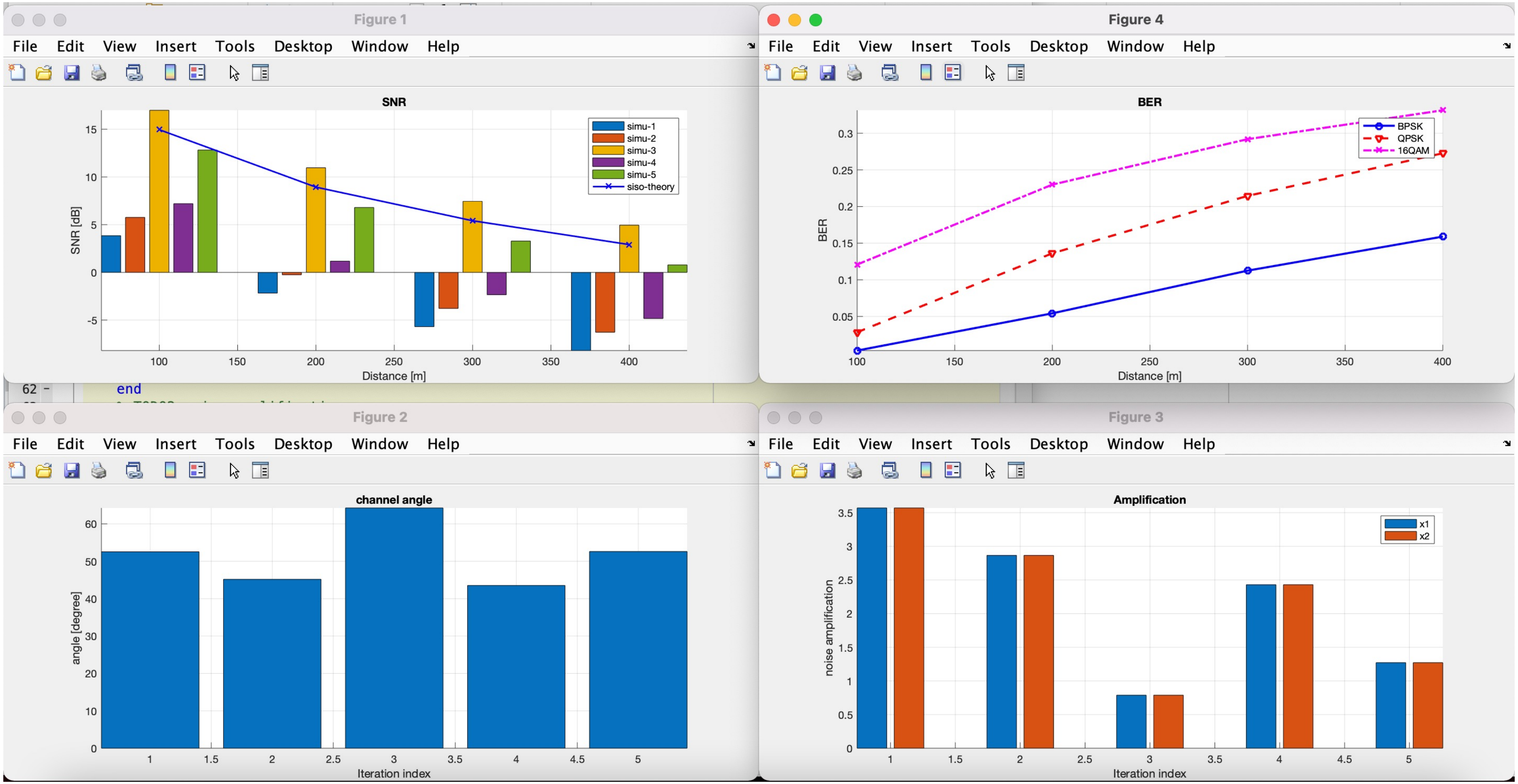
How to Calculate Noise Amp?

$$|x'|^2 = |H^{-1}|^2 * |x|^2$$

$$invH = H^{-1}$$

$$amp_i = |invH_{i,:}|^2$$

Example Results



```
end
```

Grading

- Code (60%)
  - Channel angle (15%)
  - Noise amplification (15%)
  - Generate received signal (15%)
  - ZF equalization (15%)
- Report and discussion (40%)
- Late policy
  - 20% off within 1 week of the deadline
  - After a week, 20% off per day
- Cheating
  - Cheaters equally share the score

Code Submission

- Deadline: Jun. 7 (Mon.) 23:59
- Submit to new E3
    - Source code: lab7_mimo_<studentID>.m
    - Figures:
        - lab7_SNR_<studentID>.jpg
        - lab7_ angle_<studentID>.jpg
        - lab7_ amp_<studentID>.jpg
        - lab7_ BER_<studentID>.jpg
    - report: lab7_mimo_<studentID>.pdf, including the following four figures and **your observation**!
        - SNR
        - Channel angle
        - Noise amplification
        - BER