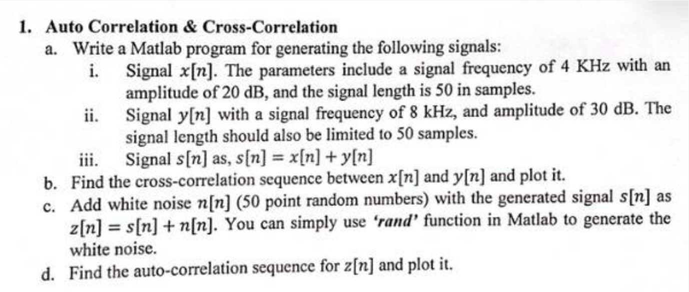
Matlab code for correlation



1. **Auto Correlation & Cross-Correlation**
   a. Write a Matlab program for generating the following signals:
      i. Signal $x[n]$. The parameters include a signal frequency of 4 KHz with an amplitude of 20 dB, and the signal length is 50 in samples.
      ii. Signal $y[n]$ with a signal frequency of 8 kHz, and amplitude of 30 dB. The signal length should also be limited to 50 samples.
      iii. Signal $s[n]$ as, $s[n] = x[n] + y[n]$
   b. Find the cross-correlation sequence between $x[n]$ and $y[n]$ and plot it.
   c. Add white noise $n[n]$ (50 point random numbers) with the generated signal $s[n]$ as $z[n] = s[n] + n[n]$. You can simply use ***'rand'*** function in Matlab to generate the white noise.
   d. Find the auto-correlation sequence for $z[n]$ and plot it.

find matlab code

Got it ✅
The problem is asking you to generate signals, compute **auto-correlation** and **cross-correlation** using MATLAB.

Here's the MATLAB code for the whole question:

```
matlab

clc;
clear;
close all;

%% Parameters
N = 50;                  % Signal length
fs = 32000;              % Sampling frequency (choose > 2*max freq for Nyquist)
n = 0:N-1;               % Discrete sample index

%% (a) Generate signals
% i. x[n] : 4 kHz, 20 dB
Ax = 10^(20/20);          % Convert dB to linear amplitude
fx = 4000;                % Frequency in Hz
x = Ax * sin(2*pi*fx*n/fs);

% ii. y[n] : 8 kHz, 30 dB
Ay = 10^(30/20);          % Convert dB to linear amplitude
fy = 8000;                % Frequency in Hz
y = Ay * sin(2*pi*fy*n/fs);

% iii. s[n] = x[n] + y[n]
s = x + y;

%% (b) Cross-correlation between x[n] and y[n]
rxy = xcorr(x,y);
lags_xy = -(length(x)-1):(length(x)-1);

figure;
stem(lags_xy, rxy, 'filled');
title('Cross-correlation r_{xy}[n]');
xlabel('Lag');
ylabel('Amplitude');
grid on;

%% (c) Add white noise to s[n]
n_white = rand(1,N);      % 50 random noise samples (0–1 uniform)
z = s + n_white;
```

```
%% (d) Auto-correlation of z[n]
rz = xcorr(z);
lags_z = -(length(z)-1):(length(z)-1);

figure;
stem(lags_z, rz, 'filled');
title('Auto-correlation r_{zz}[n]');
xlabel('Lag');
ylabel('Amplitude');
grid on;

%% Plot signals for clarity
figure;
subplot(3,1,1);
stem(n, x, 'filled'); title('Signal x[n]'); xlabel('n'); ylabel('x[n]');

subplot(3,1,2);
stem(n, y, 'filled'); title('Signal y[n]'); xlabel('n'); ylabel('y[n]');

subplot(3,1,3);
stem(n, s, 'filled'); title('Signal s[n] = x[n] + y[n]'); xlabel('n'); ylabel('s[n]');
```

🔑 Explanation of steps:

1. **Amplitude conversion**:
    - dB → linear: $A = 10^{\frac{dB}{20}}$
    - So 20 dB → 10, 30 dB → 31.62.
2. **Signal generation**:
    - $x[n] = A_x \sin(2\pi f_x n / fs)$
    - $y[n] = A_y \sin(2\pi f_y n / fs)$
3. **Cross-correlation**: `xcorr(x,y)`
    - Measures similarity between two signals at different lags.
4. **White noise**: `rand(1,N)` generates 50 uniform random samples.
5. **Auto-correlation**: `xcorr(z)`
    - Shows self-similarity of the noisy signal.

👉 Do you want me to also add **Cartesian-style grid plots for all signals** (as your earlier question asked)?