**LAPORAN KEMAJUAN TESIS**

### Perancangan parameter sistem

Parameter sistem ini dirancang untuk beroperasi di daerah padat penduduk (dense urban) dan mengikuti standar 5G. Spesifikasi parameter sistem *Massive* MIMO ditunjukkan pada Tabel 1.

|  |  |
| --- | --- |
| **Parameter** | **Suburban area** |
| *Frekuensi carrier* | 3.4GHz |
| *Spectral bandwidth* | 20MHz |
| Jumlah Antena BTS | 100 antena |
| Jumlah *user* | 30*user* |
| Gain antena BTS | 27dBi |
| Gain antena terminal | 2.2dBi |
| *Noise* figure BTS | 7dB |
| *Noise* figure perangkat *user* | 7dB |
| Temperatur *noise* | 290K |
| Kecepatan mobitilas *user* | 71km/h |
| Daya radias BTS | 1W |
| Daya radiasi perangkat *user* | 200mw |
| Delay spread urban macro | 0.65 |
| *Coherence Bandwidth* | 770 kHz |

Sesuai dengan paramater pada Tabel 1, perhitungan parameter sistem *Massive* MIMO dijabarkan sebagai berikut:

* Perhitungan jumlah pilot

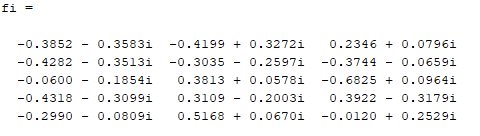
Asumsi jumlah pilot = 20% dari *coherence interval*, maka jumlah pilot :

### Estimasi CSI

Simulasi tahap awal yang dilakukan adalah dengan menggunakan jumlah antena array skala kecil terlebih dahulu. Jumlah antena BTS (*M*) sebanyak 10 dan jumlah *user* (*K*) yang dilayani sebanyak 3. Proses ini dilakukan untuk mengetahui apakah system komunikasi yang dibangun sudah benar dan sesuai dengan skema peneilitian. Berdasarkan perhitungan parameter sistem. Maka simulasi estimasi CSI diuraikan sebagai berikut:

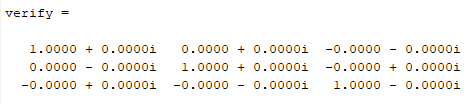
* Transmisi pilot dari *user* ke BTS.

Jumlah pilot yang ditransmisikan sebanyak 5 sample. Pilot yang ditransmisikan harus *orthogonal*. Oleh karena itu terlebih dahulu *user* harus membangkitkan *orthogonal* pilot sequence. Maka *user* akan mentransmisikan sinyal pilot yang berukuran . Matrix merupakan unitary matrix yang berukuran. Berikut ini merupakah hasil pembangkitan matrix

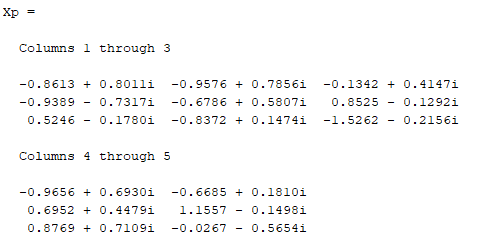


Matrix bersifat unitary, yang artinya jika dikalikan dengan matrix hermintiannya akan menghasilkan matrix identitas berukuran .

Pembuktian unitary matrix :



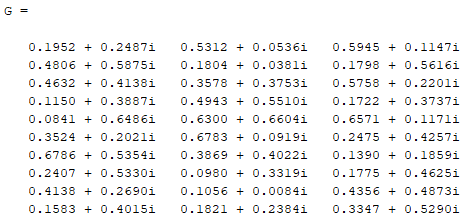
Maka pilot yang ditransmisikan oleh *user* adalah:



* Pembangkitan kanal *Rayleigh*.

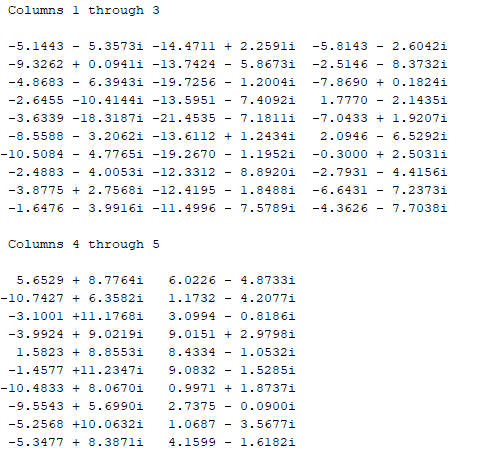
Untuk sementara, koefisien *large-scale fading* untuk setiap *user* diasumsikan sama yaitu bernilai 1. Sedangkan koefisien *small-scale fading* dibangkitkan secara random CN (0,1). Maka berdarkan persamaan berikut

Respon kanal *Rayleigh*  yang berukran [ M x K ] yang dibangkitkan adalah:



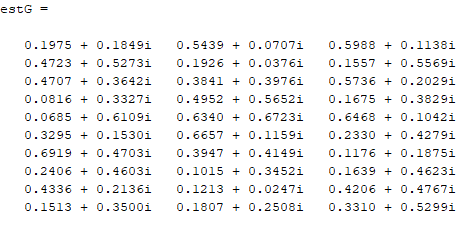
* *De-spreading* sinyal di BTS

Setelah *user* mentansmisikan sinyal pilot, BTS akan menerima sinyal pilot tersebut yang sudah dikalikan dengan gain kanal dan ditambah *noise*, sesuai persamaan (2.23). Sinyal yang diterima di BTS adalah:

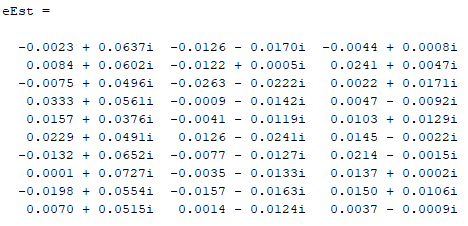


Kemudian BTS akan melakukan proses *de-spreading* sinyal pilot yang diterima. Sinyal hasil *de-spreading* ini digunakan untuk MMSE estimator. Sesuai dengan persamaan

Maka hasil estimasi kanal adalah:

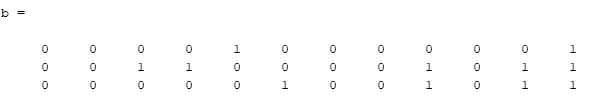


Hasil Estimasi kanal ini nantinya akan digunakan oleh BTS untuk membentuk precoding. Error estimasi kanal andalah

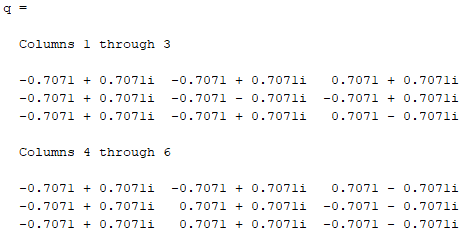


* Pembangkitan data oleh BTS

BTS membangkitkan data untuk semua user. Deretan bit biner yang dinagkitkan untuk semua user adalah



Deretan bit data kemudian dimodulasi menggunakan system modulasi 4QAM. Hasil modulasi 4 QAM adalah sebaga berikut:



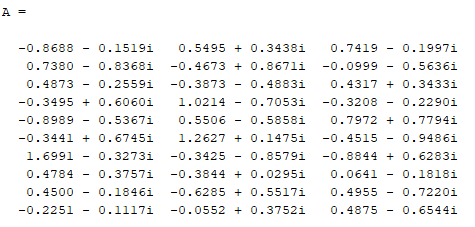
Data hasil modulasi QAM berukuran [K x M]. Sehingga baris pertama merupakan data untuk user 1. Baris kedua adalah data untuk user 2 dan baris ketiga adalah data untuk user ketiga.

* BTS membentuk precoding matrix

Setelah mendapatkan hasil estimasi kanal, BTS akan menggunakan hasil estimasi kanal untuk membentuk precoding matriks. Precoding berfungsi untuk meminimalisasi interferensi antar user. Teknik precoding yang digunakan adalah zero forcing. Seperti pada persamaan berikut

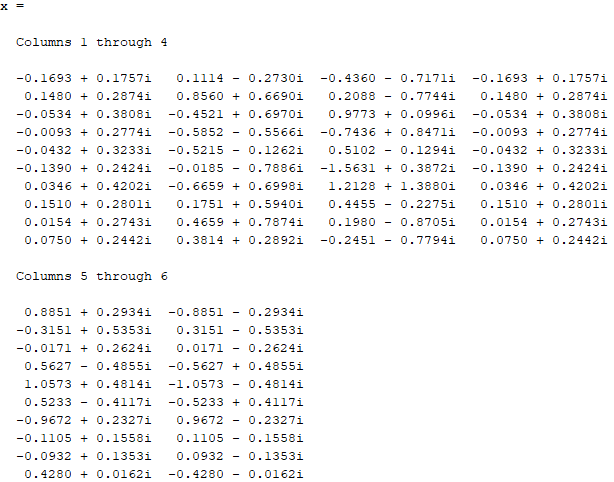
|  |
| --- |
|  |
|  |

Diasumsikan kanal bersifat resiprokal dan tidak berubah pada interval waktu tertentu (*coherence interval*). Maka hasil precoding matriks adalah



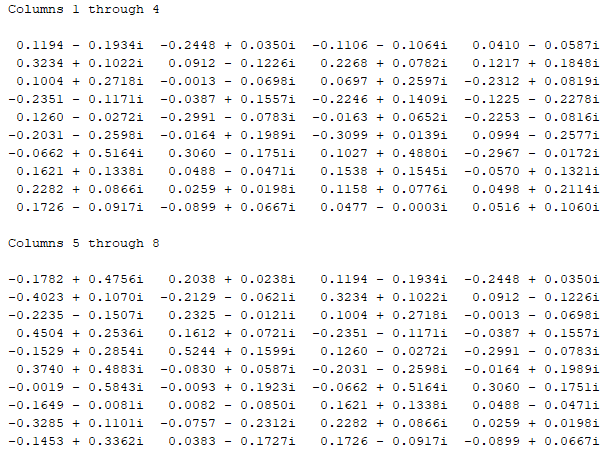
* Perkalian data symbol dengan precoding matriks

Sinyal yang akan dipancarkan oleh BTS sebelum masuk blok OFDM merupakan perkalian antara *precoding* matrix dan vector simbol QAM ().



* OFDM di sisi BTS

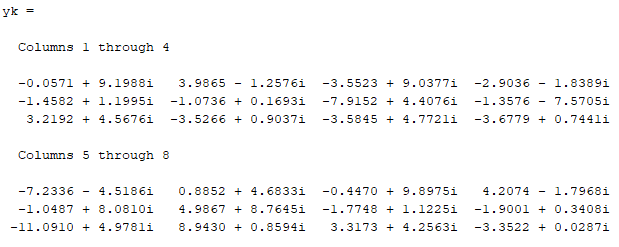
Sistem OFDM dapat mengatasi adanya *Inter Symbol Interference* (ISI) pada respon frekuensi *selective*. Sistem OFDM membagi bandwidth kanal menjadi bagian-bagian frekuensi dengan lebar pita sempit . Dalam system ini jumlah subcarriers sama dengan jumlah antenna pemancar. Proses OFDM di sisi pemancar dilakukan dengan menggunakan Teknik IFFT. Kemudian sinyal hasil IFFT diberi tambahan cyclic prefix yaitu dengan cara menambahkan 25% dari data terakhir ke awal data. Hasil dari proses OFDM adalah sebagai berikut:



* Transmisi Sinyal dari BTS ke UE

Sinyal hasil OFDM ini kemudian ditransmisikan ke user melalui sejumlah M-antena BTS. Diasumsikan proses transmisi ini masih dalam satu coherence interval dengan proses estimasi kanal oleh BTS. Sehingga kanal downlink sama seperti kanal uplink. Sehingga sinyal yang diterima oleh UE adalah perkalian antara sinyal yang ditransimikan dengan kanal dan ditambah noise, seperti ditunjukkan oelah persamaan berikut:

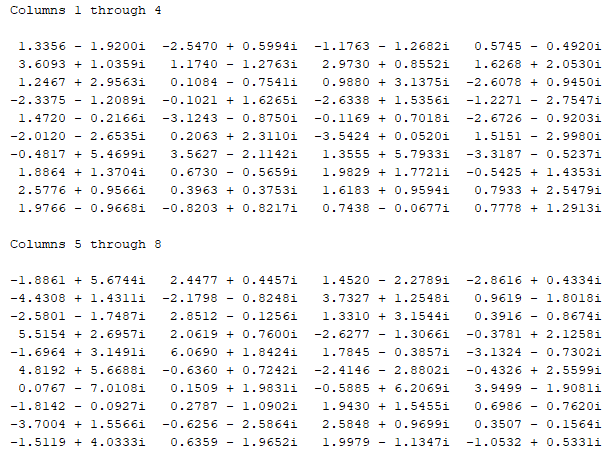
Hasil simulasi sinyal yang diterima adalah:



* Mengilangkan komponen kanal dari sinyal yang diterima

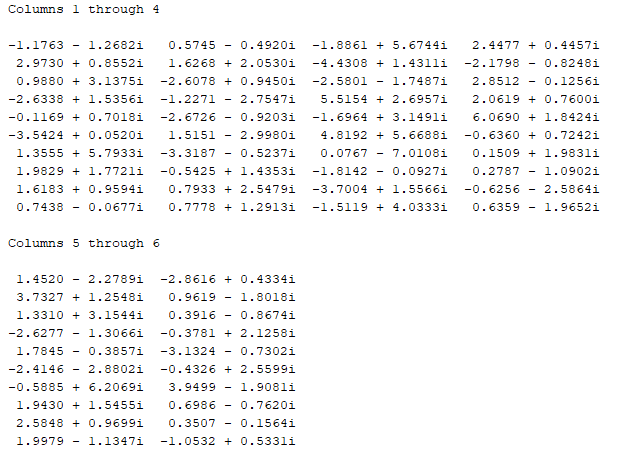
Karena sinyal yang diterima telah dikalikan dengan gain kanal, maka komponen kanal dihilangkan dengan cara mengalikan sinyal yang diterima dengan Teknik zero forcing yaitu dengan melakukan membentuk moonre pseudoinverse dari matriks kanal.

Sehingga sinyal yang diterima menjadi:

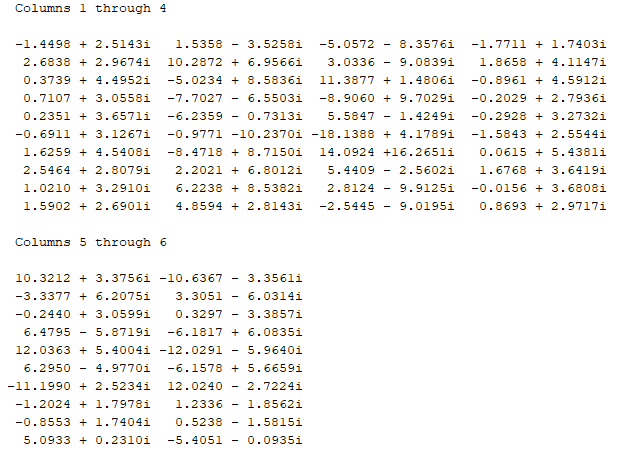


* Remove Cyclix Prefix

Sinyal yang diterima adalah sinyal yang masih megandung tambahan syclix prefix, sehingga perlu dihilangkan. Hasil sinyal tanpa cyclix prefix adalah sebagai berikut

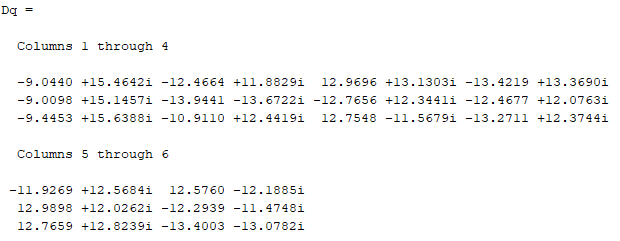


* Proses FFT di penerima

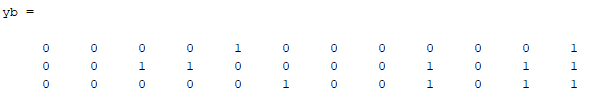
Sinyal tanpa cyclic prefix dari domain waktu kemudian diubah ke domain frekuensi dengan cara FFT. Hasil FFT dari sinyal adalah:  


* Deteksi symbol

Hasil sinyal yang telah diproses melalui blok FFT merupakan sinyal yang masih tercampur antara user 1 dan user lain, oleh karena itu harus dipisahkan untuk masing-masing user. Karena sinyal transmisikan sudah melalui proses precoding sebelumnya, maka untuk memisahkan ke masing-masing user hanya perlu dikalikan dengan matriks kanal saja. Sehingga sinyal yang diterima menjadi:



Sinyal di atas kemudian masuk ke proses deteksi symbol menggunakan skema demodulasi 4 QAM. Setelah proses deteksi symbol, selanjutnya akan diubah menjadi bit biner. Hasil biner dari sinyal yang diterima adalah sebagai berikut:



Bit biner yang diterima sama persis dengan bit biner yang dibangkitkan di sisi BTS. Sehingga erro bit nya adalah 0. Dalam hal ini tidak ada error bit karena noisenya rendah dan bit yang ditransmisikan hanya sedikit. Untuk selanjutnya akan disimulasikan untuk jumlah bit yang banyak dengan variasi noise, variasi lintasan kanal, dan variasi Jumlah antenna pemancar dan penerima.

**Lampiran program Matlab:**

% Generate Haar distributed random unitary matrices

% From book Acta Numerica

clear all; clc; % Clear screen

tau\_p = 5; % Pilot length

M = 10; % Number of Tx antenna (in one BS)

K = 3; % Number of Rx antenna (= number of UE)

L = 4; % Channel tap frequency selective

v = 1 / sqrt(L); % Variance channel for frequency selective

vn = 1; % Variance noise

mu = 0; % Mean of noise

beta = 1; % Large scale fadding Coefficient Of Rayleigh Channel

SNR\_ul = 128; % Uplink SNR (dB)

SNR\_dl = SNR\_ul; % Download SNR (dB)

nBit = 2; % Number of bits per chunk

QAM\_symbol = [-1 1; 1 1; 1 -1 ;-1 -1];

% ===================== UE SIDE ==========================

% GENERATE PILOT -----------------------------------------

% Generate a random complex matrix

X = complex(rand(tau\_p), rand(tau\_p)) / sqrt(2);

% Factorize the matrix

[Q, R] = qr(X);

% Unitary matrix M x N

fi = Q(:, 1:K);

% Verification of unitary matrix

verify = ctranspose(fi) \* fi;

% Generate the pilot

Xp = sqrt(tau\_p) \* ctranspose(fi);

% ===================== UE > BS ==========================

% Standard deviation

sigma = sqrt(v);

% Generate the Rayleigh channel from UE to BS

H = sigma .\* complex(rand(M, K), rand(M, K)) + mu;

% Add large scale fadding to the Rayleigh channel

G = sqrt(beta) \* H;

% Generate noise (AWGN)

Wp = sqrt(vn) .\* complex(rand(M, tau\_p), rand(M, tau\_p)) + mu;

% ===================== BS SIDE ==========================

% Received pilot signal

Yp = sqrt(SNR\_ul) \* G \* Xp + Wp;

% Despreading pilot

Yp\_aksen = Yp \* fi;

% Channel Estimation -------------------------------------

% MMSE channel estimation

estG = zeros(M,K);

for i=1:M;

for j=1:K;

% Estimated the chanel

estG(i,j) = (sqrt(tau\_p\*SNR\_ul)\*beta) / (1+tau\_p\*SNR\_ul\*beta) .\* Yp\_aksen(i,j);

end

end

eEst = G - estG; % Channel estimation error

%Mean-square channel estimation (unused, for verification only)

gamma = (tau\_p\*SNR\_ul\*beta^2) / (1+tau\_p\*SNR\_ul\*beta);

MSE = beta - gamma; % Mean Square Error

% Generate sample data for each UE -----------------------

jBit = nBit\*K\*2; % Total generated bits

% Pre-allocating

b = zeros(K, jBit);

for i=1:K;

% Generate streams of binary data (serial)

b(i,:) = randi([0 1], [1 jBit]);

end

% 4-QAM Modulation ---------------------------------------

symbol = QAM\_symbol / sqrt(2);

% Pre-allocating

q = zeros(K, jBit/nBit);

for i=1:K;

% Divide streams into chunks (Serial to Parallel)

pBit = transpose(reshape(b(i,:), nBit, []));

% Convert binary into decimal

pDec = bi2de(pBit);

% Use decimal as 4 QAM inputs

pComp = symbol(pDec+1, :);

% Combine real and imaginer

pQAM\_real = transpose(pComp(:, 1));

pQAM\_imag = transpose(pComp(:, 2));

pQAM = complex(pQAM\_real, pQAM\_imag);

% QAM symbol for each UE

q(i,:) = pQAM;

end

% Precoding ----------------------------------------------

% Identical power for each UE

power\_dl = [1/K 1/K 1/K];

% Zero Forcing Precoding Matrix

A = sqrt(M-K) \* conj(estG) / (transpose(estG)\*conj(estG));

% Signal to be transmitted (mixed)

x = A \* sqrt(diag(power\_dl)) \* q;

% OFDM ---------------------------------------------------

N = size(x, 1) ; % Number of subcarrier = number of Tx antenna

NFFT = size(x, 2) % Number point of FFT

CP = ceil(0.25\*NFFT); % Number of cyclic Prefix (25% of NFFT)

% IFFT process

IFFT\_SC = zeros(N, NFFT);

xCP = zeros(N, CP);

for i = 1:N;

% IFFT of transmitted signal

IFFT\_SC(i,:) = ifft(x(i,:), NFFT);

% Copy the end of signal to the begining of signal

xCP(i,:) = IFFT\_SC(i, NFFT-CP+1:NFFT);

end

% Add CP -------------------------------------------------

OFDM\_symbol = [xCP IFFT\_SC]; %CP + IFFT

% ===================== BS > UE ==========================

% Generate noise (AWGN)

W\_real = rand(K, size(OFDM\_symbol, 2));

W\_imag = rand(K, size(OFDM\_symbol, 2));

W = sqrt(vn) .\* complex(W\_real, W\_imag) + mu;

% ===================== UE SIDE ==========================

% Received Signal at UE

HA = transpose(estG);

yk = sqrt(SNR\_dl) \* HA \* OFDM\_symbol + W;

% Remove channel effect using zero forcing

yki = pinv(HA) \* yk;

% Remove CP ----------------------------------------------

% Received OFDM symbol with noise and channel

OFDM\_rem = yki(:, size(xCP, 2)+1:end);

fft\_symbol = zeros(N, NFFT);

for i=1:N

fft\_symbol(i,:) = fft(OFDM\_rem(i,:), NFFT);

end

xx = fft\_symbol;

% Detection -----------------------------------------------

DA = transpose(estG);

Dq = DA\*xx;

% 4-QAM De-Modulation ------------------------------------

yQAM = Dq; % Demodulation input

% Pre-allocating

yNumSymbol = size(yQAM, 2);

yb = zeros(K, nBit \* yNumSymbol);

yDec = zeros(yNumSymbol, 1);

% Iterate every UE

for i = 1:K;

% Separate real & imag from complex

z9 = [real(yQAM(i,:)); imag(yQAM(i,:))];

% Convert serial to parallel

z0 = transpose(z9);

% Get only the sign

z1 = sign(reshape(z0, [], 2));

% Iterate every symbol

for j = 1:size(z1, 1);

% Set true for matched QAM symbol

z2 = ismember(QAM\_symbol, z1(j, :), 'rows');

% Get index of true result

yDec(j) = find(z2) - 1;

end;

% Convert decimal to binary

yBit = de2bi(yDec);

% Paralel to serial

yb(i,:) = reshape(yBit', 1, []);

end

BER = yb-b