Wireless Communication Systems HW3

通訊所一年級 陳劭珩 *110064533*

November 7, 2021

1. Implement a Rayleigh fading channel simulator based on the Filtered Gaussian Noise method

– Plot the channel output for $f_m T = 0.01, 0.1$ *and* 0.5 $(t\,/\,T\ = 0\sim300)$

– Plot the channel output autocorrelation for $f_m T = 0.01, 0.1$ *and* 0.5 $(\,f_m\tau = 0\sim10)$

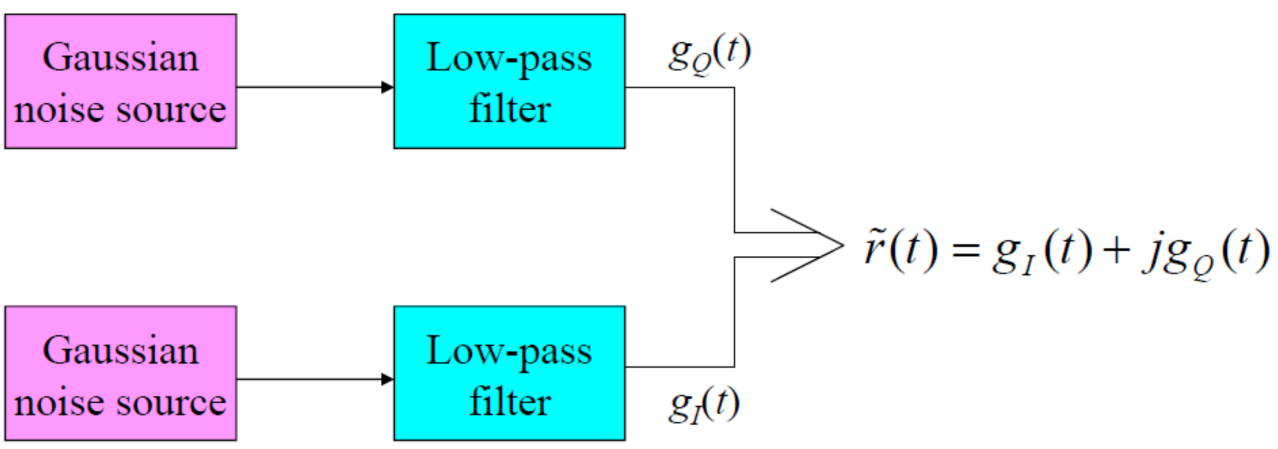
1.1 Answer

依照上課投影片的數學式及方塊圖來進行模擬，取$epoch\ K = 100000$，等於有10萬個$samples$、取$\Omega_p = 2$，功率 $power = \Omega_p/2\ = 1$，作圖時隨機取一個$time\ shift = 300\sim99700$，以避開初始不穩定的狀態。

$$\left(g_{I,k+1}, g_{Q,k+1}\right) = \zeta\left(g_{I,k}, g_{Q,k}\right) + (1-\zeta)\left(w_{1,k}, w_{2,k}\right)$$

$$\zeta\text{^}2\ -\ 2\zeta\left(2\ -\ cos\left(\frac{\pi f_m T}{2}\right)\right)\ +\ 1\ =\ 0$$

$$\rightarrow \zeta\ =\ 2\ -\ cos\left(\frac{\pi f_m T}{2}\right)\ -\ \sqrt{2\ -\ cos\left(\frac{\pi f_m T}{2}\right)^2\ -\ 1}$$



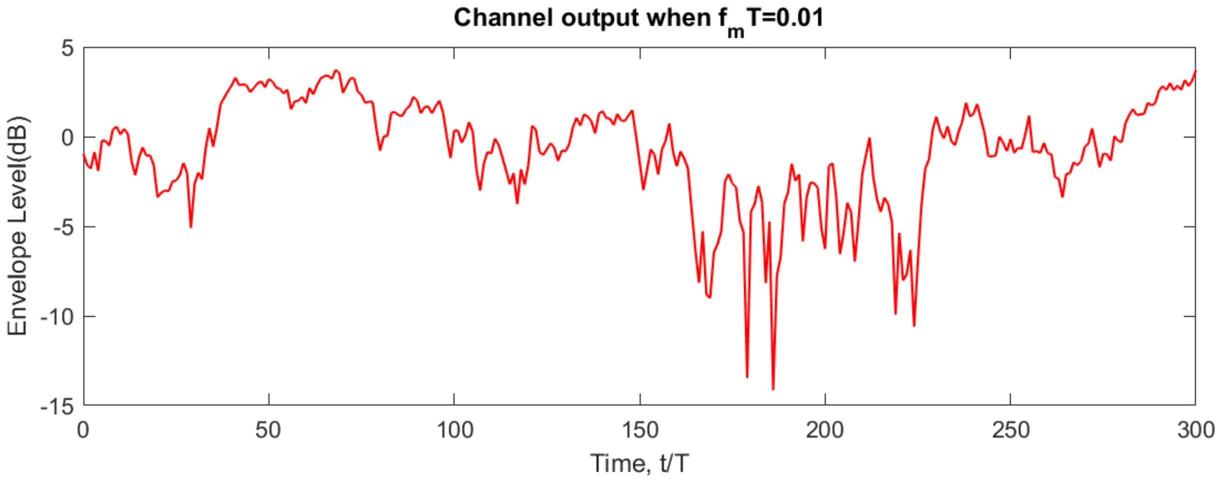
1.2 Code

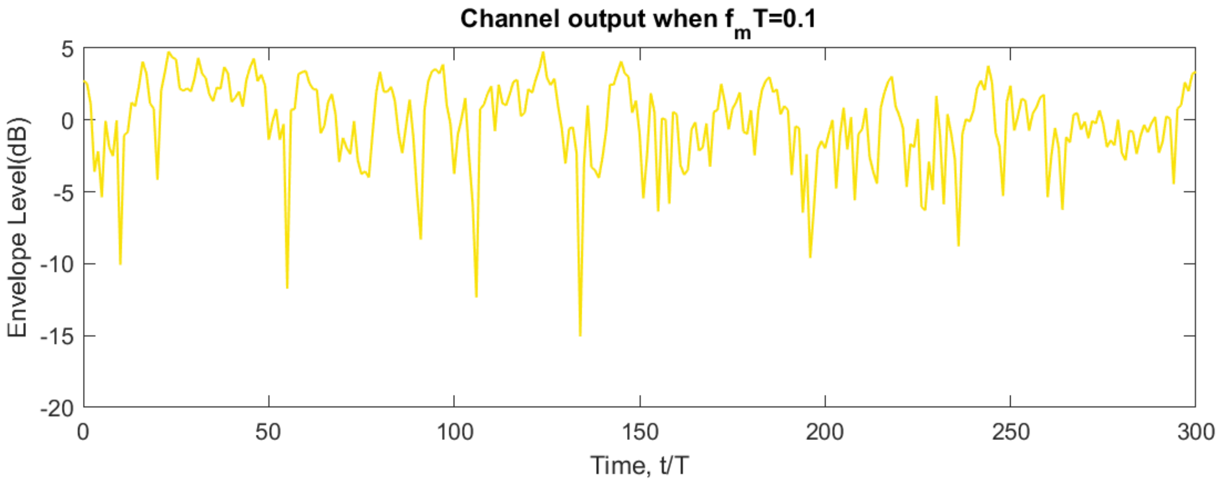
```
%
% Wireless Communication Systems HW3, 通訊所一年級 110064533 陳劭珩
%
% 1. Implement a Rayleigh fading channel simulator based on the Filtered
%    Gaussian Noise method
%    - Plot the channel output for fmT = 0.01, 0.1, 0.5 (t/T = 0~300)
%    - Plot the channel output autocorrelation for fmT = 0.01, 0.1, 0.5
%      (fmτ = 0~10)
%
clear;
clc;
%
fmT = [0.01, 0.1, 0.5]; % fmT = 0.01, 0.1, 0.5
t_over_T = 0:1:300;     % t/T = 0~300
%
K = 1e5;                % epoch k = 1, 2, 3, ... , K. 10^5 samples
omega_p = 2;            % total power Ω_p
power = omega_p / 2;    % power = σ_g_I^2 = σ_g_Q^2 = Ω_p/2
shift_t = round(300 + (K-300)*rand);
%
r = zeros(1, K);        % envelope r(t) = g_I(t) + j*g_Q(t)
g_I = zeros(1, K);      % g_I(t) = LPF(Gaussian noise)
g_Q = zeros(1, K);      % g_Q(t) = LPF(Gaussian noise)
envelope = zeros(1, K);
envelope_dB = zeros(1, K);
%
for i = 1:3
    %
    % ζ^2 - 2ζ(2 - cos(πf_mT/2)) + 1 = 0
    % ζ = 2 - cos(πf_mT/2) - √(2 - cos(πf_mT/2)^2 - 1)
    %
    zeta = 2 - cos(pi*fmT(i)/2) - sqrt((2-cos(pi*fmT(i)/2))^2 - 1);
    %
    % σ_g_I^2 = power = (1 - ζ)/(1 + ζ) * σ^2, σ^2 variance of w_1,k w_2,k
    % σ = √((1 + ζ)/(1 - ζ) * power)
    %
    sigma = sqrt((1 + zeta)/(1 - zeta) * power);
    %
    % Gaussian noise source. w_1,k and w_2,k
    %
```

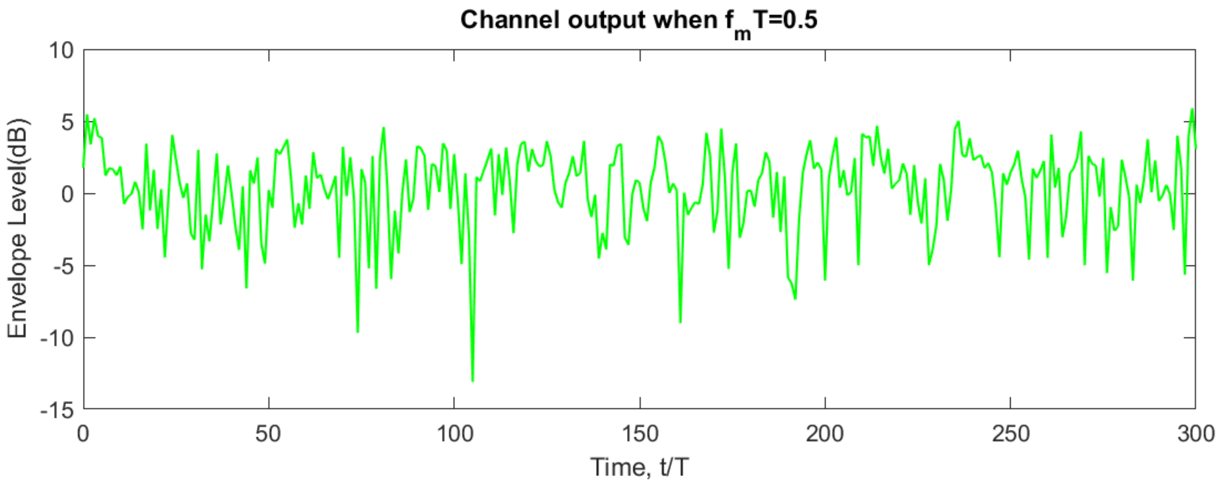
```
w_I = randn(1, K) * sigma; % w_I = w_1,k
w_Q = randn(1, K) * sigma; % w_Q = w_2,k
%
% pass the Gaussian noise w_1,k and w_2,k through the first-order LPF,
% we can obtain th real and imaginary parts of the complex envelope
%
g_I(1) = (1 - zeta) * w_I(1);
g_Q(1) = (1 - zeta) * w_Q(1);
%
envelope(1) = sqrt(g_I(1)^2 + g_Q(1)^2);
envelope_dB(1) = 10 * log10(envelope(1));
%
for k = 2:K
    %
    g_I(k) = zeta * g_I(k-1) + (1-zeta) * w_I(k-1);
    g_Q(k) = zeta * g_Q(k-1) + (1-zeta) * w_Q(k-1);
    %
    r(k) = g_I(k) + 1i*g_Q(k);
    envelope(k) = sqrt(g_I(k)^2 + g_Q(k)^2);
    envelope_dB(k) = 10 * log10(envelope(k));
    %
end
%
% plot and save
fig = figure(1);
if i == 1
    fig;
    subplot(3, 2, 1);
    plot1_1 = plot(t_over_T, envelope_dB(0+shift_t : 300+shift_t));
    set(plot1_1, 'LineWidth', 1, 'LineStyle', '-', 'Color', 'r');
    xlabel('Time, t/T');
    ylabel('Envelope Level(dB)');
    title('Channel output when f_mT=0.01');

    %
    autocorrelation = autocorr(r(:), 10/fmT(i));
    subplot(3, 2, 2);
    time_delay = 0 : fmT(i) : 10;
    plot1_2 = plot(time_delay, autocorrelation);
    set(plot1_2, 'LineWidth', 1, 'LineStyle', '-', 'Color', 'r');
    xlabel('Time delay, f_mτ');
```

```
        ylabel('Autocorrelation, ϕgg(τ)');
        title('Channel output autocorrelation when f_mT = 0.01');
    elseif i == 2
        fig;
        subplot(3, 2, 3);
        plot2_1 = plot(t_over_T, envelope_dB(0+shift_t : 300+shift_t));
        set(plot2_1, 'LineWidth', 1, 'Color', [0.9763 0.8831 0.0338]);
        xlabel('Time, t/T');
        ylabel('Envelope Level(dB)');
        title('Channel output when f_mT=0.1');
        %
        autocorrelation = autocorr(r(:), 10/fmT(i));
        subplot(3, 2, 4);
        time_delay = 0 : fmT(i) : 10;
        plot2_2 = plot(time_delay, autocorrelation);
        set(plot2_2, 'LineWidth', 1, 'Color', [0.9763 0.8831 0.0338]);
        xlabel('Time delay, f_mτ');
        ylabel('Autocorrelation, ϕgg(τ)');
        title('Channel output autocorrelation when f_mT = 0.1');
    elseif i == 3
        fig;
        subplot(3, 2, 5);
        plot3_1 = plot(t_over_T, envelope_dB(0+shift_t : 300+shift_t));
        set(plot3_1, 'LineWidth', 1, 'LineStyle', '-', 'Color', 'g');
        xlabel('Time, t/T');
        ylabel('Envelope Level(dB)');

        title('Channel output when f_mT=0.5');
        %
        autocorrelation = autocorr(r(:), 10/fmT(i));
        subplot(3, 2, 6);
        time_delay = 0 : fmT(i) : 10;
        plot3_2 = plot(time_delay, autocorrelation);
        set(plot3_2, 'LineWidth', 1, 'LineStyle', '-', 'Color', 'g');
        xlabel('Time delay, f_mτ');
        ylabel('Autocorrelation, ϕgg(τ)');
        title('Channel output autocorrelation when f_mT = 0.5');
    end
end
```

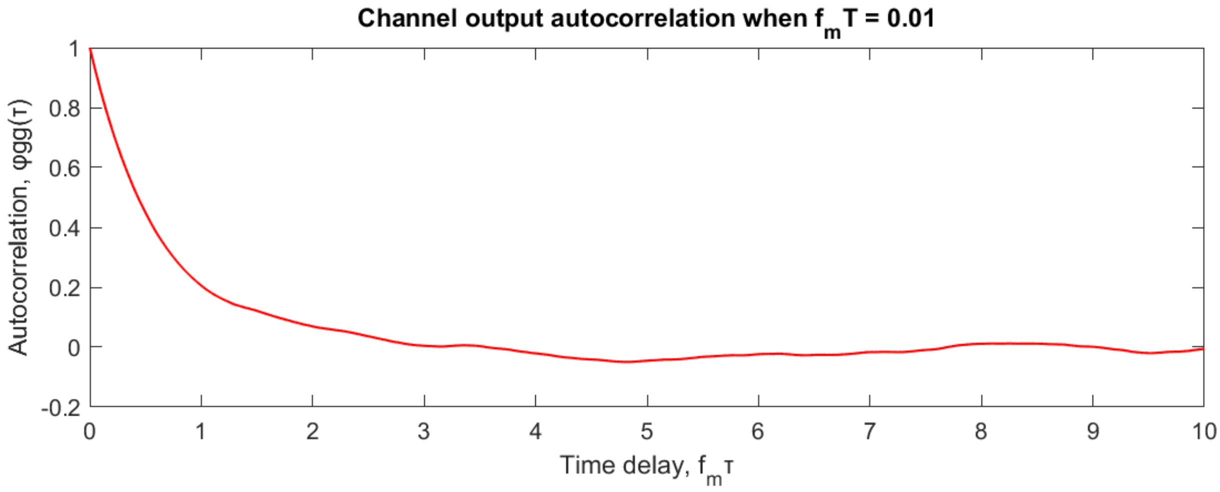
1.3.1 *Plot the channel output for* $f_m T = 0.01, 0.1$ *and* 0.5 $(t / T = 0\sim300)$

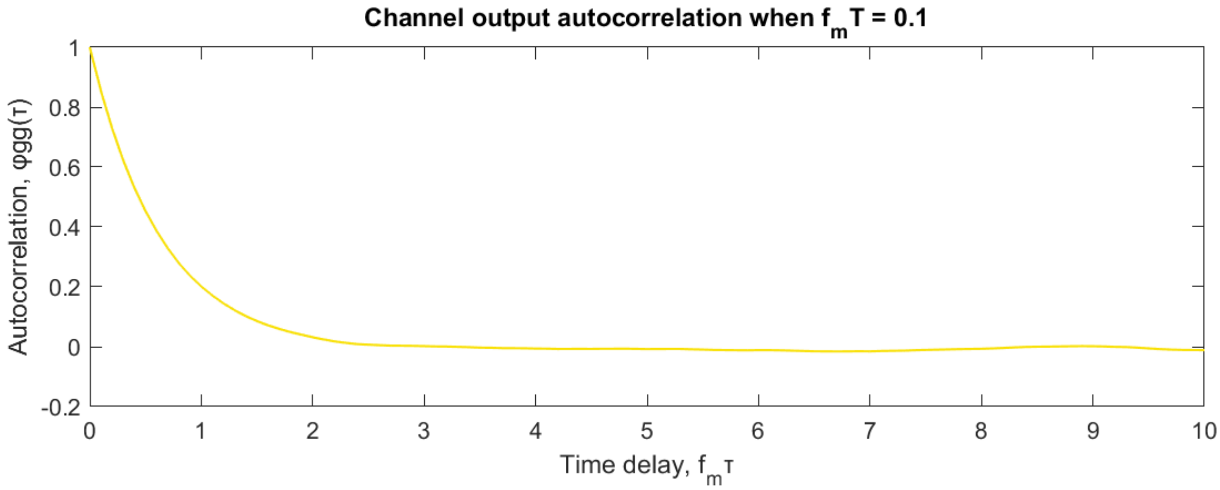


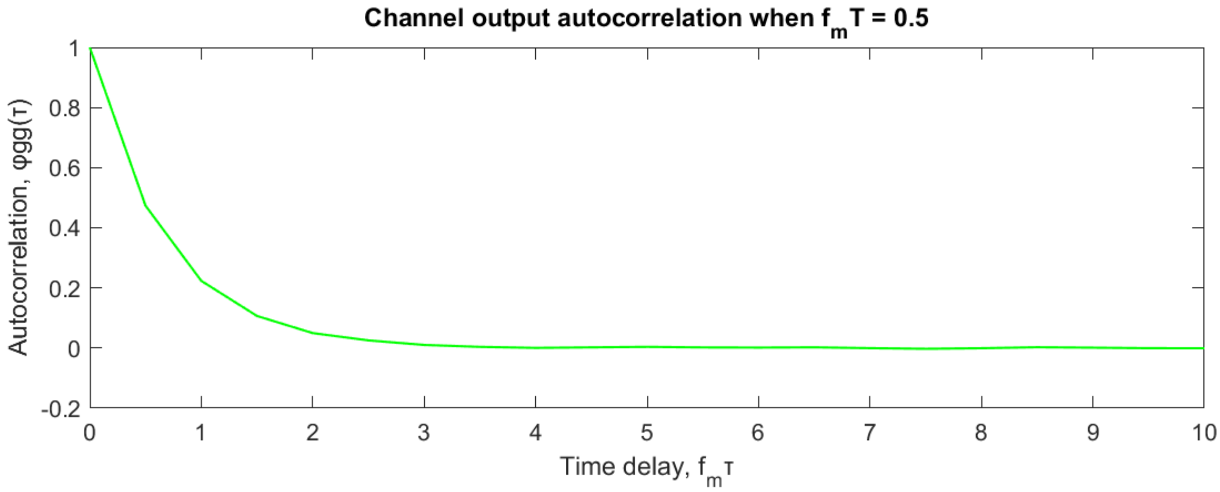




1.3.2 *Plot the channel output autocorrelation for* $f_m T = 0.01, 0.1, 0.5$ ($f_m \tau = 0 \sim 10$)







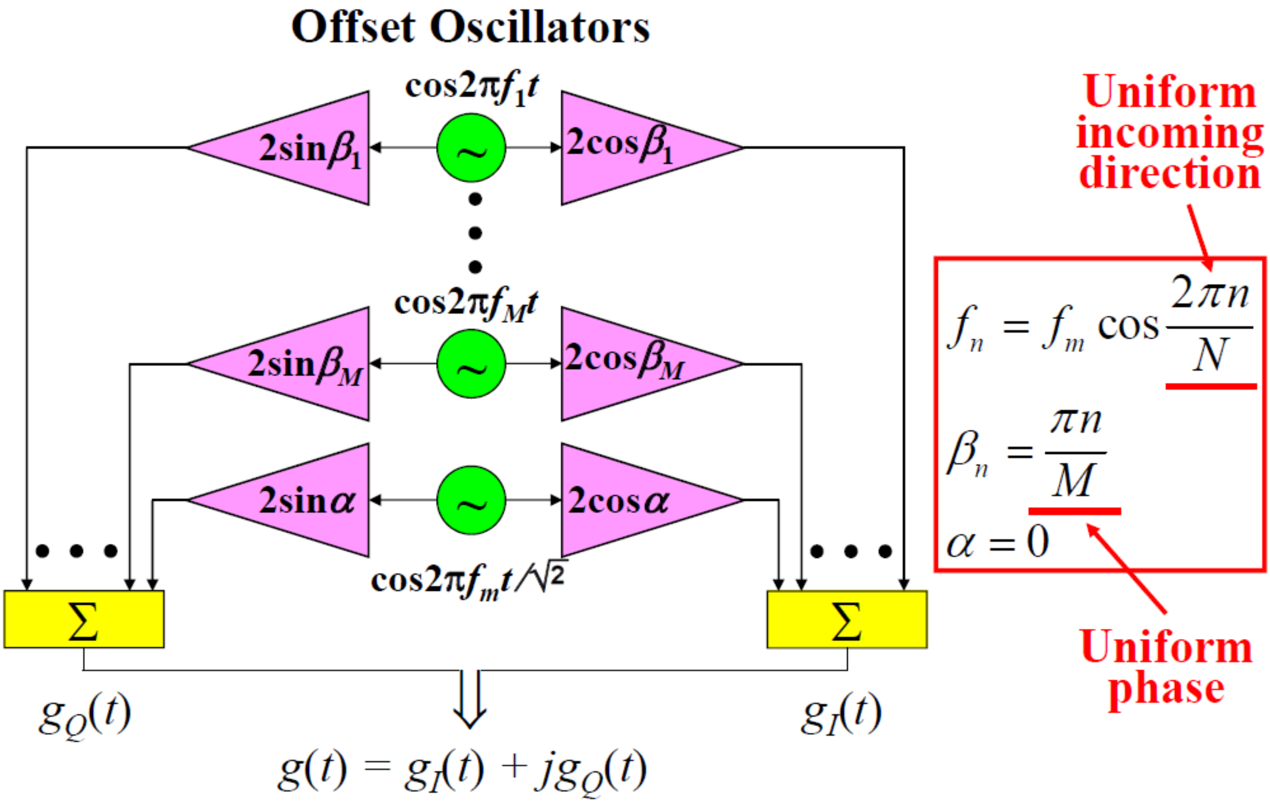
2. Implement a Rayleigh fading channel simulator based on the Sum of Sinusoids method (Jake's method)

– Plot the channel output for $M = 8, 16$ ($f_m T = 0.01, 0.1, 0.5$ *and* $t / T = 0 \sim 300$)

– Plot the channel output autocorrelation for $M = 8, 16$ ($f_m \tau = 0 \sim 10$)

2.1 Answer

依照上課投影片的數學式及方塊圖來進行模擬，一樣取10萬個$samples$，作圖時也一樣隨機取一個$time\ shift = 300 \sim 99700$，以避開初始不穩定的狀態。



$$g(t) = g_I(t) + g_Q(t)$$

$$= \sqrt{2}\left\{\left[2\sum_{n=1}^{M} cos\beta_n \cdot cos2\pi f_n t + \sqrt{2}cos\alpha \cdot cos2\pi f_n t\right]\right.$$

$$\left. + j\left[2\sum_{n=1}^{M} sin\beta_n \cdot cos2\pi f_n t + \sqrt{2}sin\alpha \cdot cos2\pi f_n t\right]\right\}$$

2.2 Code

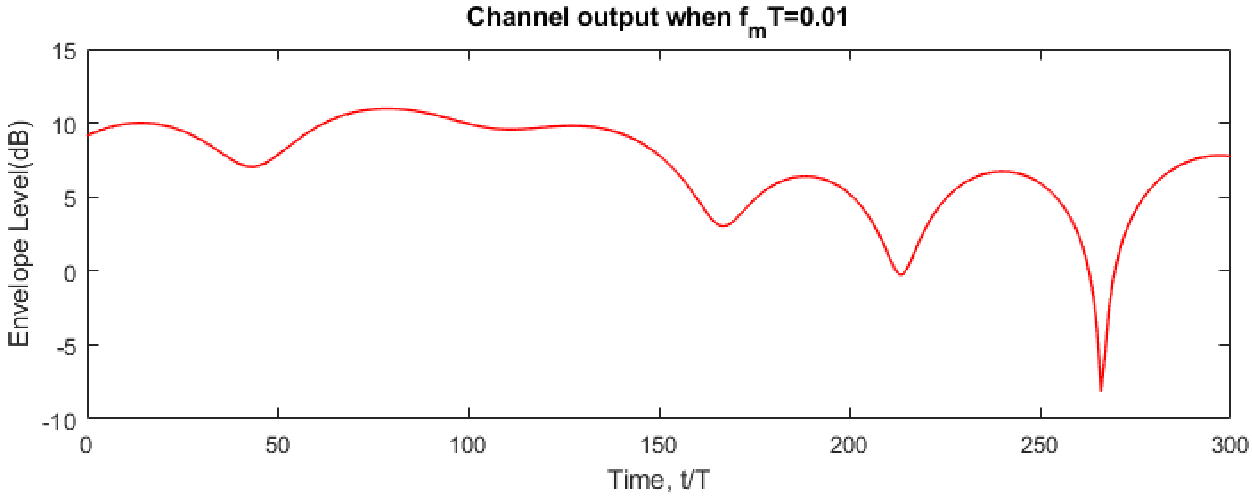
```
%
% Wireless Communication Systems HW3, 通訊所一年級 110064533 陳劭珩
%
% 2. Implement a Rayleigh fading channel simulator based on the Sum of
%    Sinusoids method
%    - Plot the channel output for M = 8, 16 (fmT = 0.01, 0.1, 0.5 and
%      t/T = 0~300)
%    - Plot the channel output autocorrelation for M = 8, 16 (fmτ = 0~10)
%
fmT = [0.01, 0.1, 0.5]; % fmT = 0.01, 0.1, 0.5
t_over_T = 0:1:300;     % t/T = 0~300
M = [8, 16];
N = 2 * (2*M + 1);
%
K = 1e5;                % 10^5 samples
fm = fmT;
shift_t = round(300 + (K-300)*rand);
%
g = zeros(1, K);        % envelope r(t) = g_I(t) + j*g_Q(t)
g_I = zeros(1, K);      % g_I(t) = LPF(Gaussian noise)
g_Q = zeros(1, K);      % g_Q(t) = LPF(Gaussian noise)
envelope = zeros(1, K);
envelope_dB = zeros(1, K);
%
for j = 1:2
    for i = 1:3
        %
        n = 1:M(j);
        theta = 2*pi*n / N(j); % uniform incoming direction incident angle
        fn = fm(i)*cos(theta);
        alpha = 0;
        beta_n = pi*n / M(j);  % uniform phase
        %
        for t = 0:K-1
            %
            g_I(t+1) = sqrt(2) * (2*sum(cos(beta_n).*cos(2*pi*fn*t)) + ...
                sqrt(2)*cos(alpha).*cos(2*pi*fm(i)*t));
            g_Q(t+1) = sqrt(2) * (2*sum(sin(beta_n).*cos(2*pi*fn*t)) + ...
                sqrt(2)*sin(alpha).*cos(2*pi*fm(i)*t));
            %
```

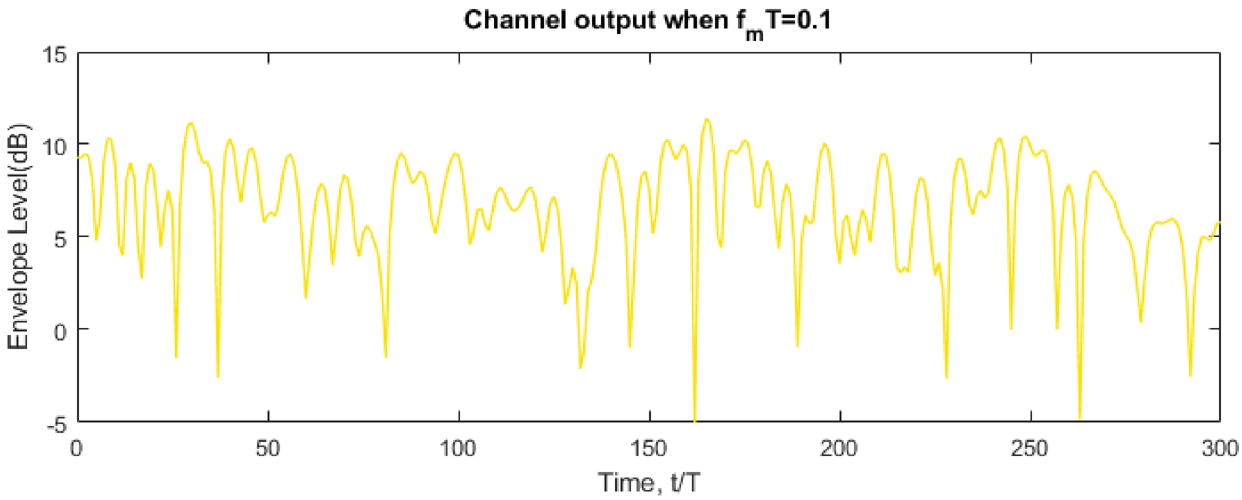
```
    g(t+1) = g_I(t+1) + 1i*g_Q(t+1);
    envelope_dB(t+1) = 10 * log10(sqrt(g_I(t+1)^2 + g_Q(t+1)^2));
end
%
% plot
fig = figure(j);
if i == 1
    fig;
    subplot(3, 2, 1);
    plot1_1 = plot(t_over_T, envelope_dB(0+shift_t : 300+shift_t));
    set(plot1_1, 'LineWidth', 1, 'LineStyle', '-', 'Color', 'r');
    xlabel('Time, t/T');
    ylabel('Envelope Level(dB)');
    title('Channel output when f_mT=0.01');
    %
    autocorrelation = autocorr(g(:), 10/fmT(i));
    subplot(3, 2, 2);
    time_delay = 0 : fmT(i) : 10;
    plot1_2 = plot(time_delay, autocorrelation);
    set(plot1_2, 'LineWidth', 1, 'LineStyle', '-', 'Color', 'r');
    xlabel('Time delay, f_mτ');
    ylabel('Autocorrelation, ϕgg(τ)');
    title('Channel output autocorrelation when f_mT = 0.01');
elseif i == 2
    fig;
    subplot(3, 2, 3);
    plot2_1 = plot(t_over_T, envelope_dB(0+shift_t : 300+shift_t));
    set(plot2_1, 'LineWidth', 1, 'Color', [0.9763 0.8831 0.0338]);
    xlabel('Time, t/T');
    ylabel('Envelope Level(dB)');
    title('Channel output when f_mT=0.1');
    %

    autocorrelation = autocorr(g(:), 10/fmT(i));
    subplot(3, 2, 4);
    time_delay = 0 : fmT(i) : 10;
    plot2_2 = plot(time_delay, autocorrelation);
    set(plot2_2, 'LineWidth', 1, 'Color', [0.9763 0.8831 0.0338]);
    xlabel('Time delay, f_mτ');
    ylabel('Autocorrelation, ϕgg(τ)');
    title('Channel output autocorrelation when f_mT = 0.1');
```

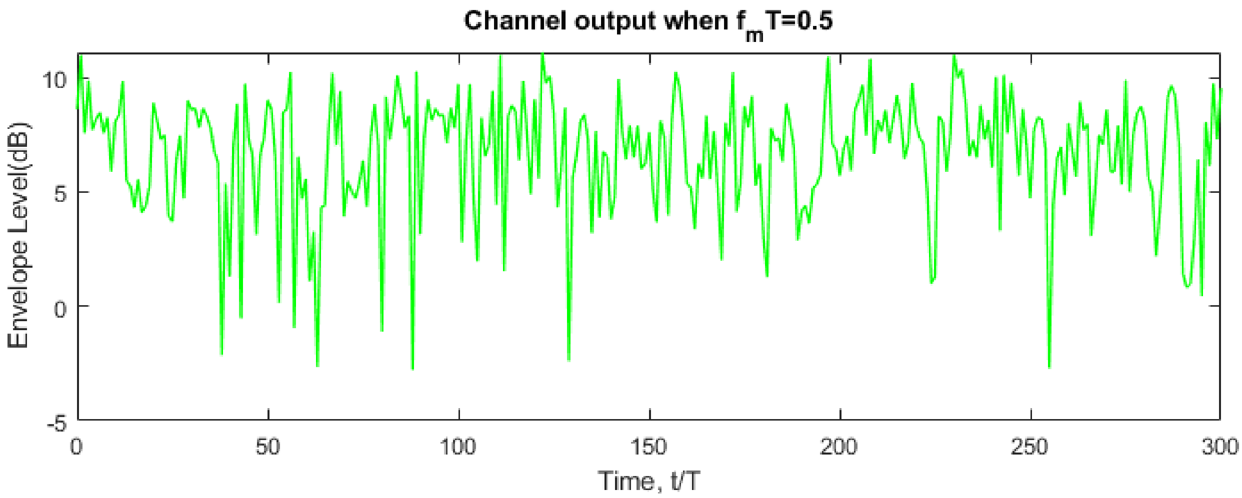
```
        elseif i == 3
            fig;
            subplot(3, 2, 5);
            plot3_1 = plot(t_over_T, envelope_dB(0+shift_t : 300+shift_t));
            set(plot3_1, 'LineWidth', 1, 'LineStyle', '-', 'Color', 'g');
            xlabel('Time, t/T');
            ylabel('Envelope Level(dB)');
            title('Channel output when f_mT=0.5');
            %
            autocorrelation = autocorr(g(:), 10/fmT(i));
            subplot(3, 2, 6);
            time_delay = 0 : fmT(i) : 10;
            plot3_2 = plot(time_delay, autocorrelation);
            set(plot3_2, 'LineWidth', 1, 'LineStyle', '-', 'Color', 'g');
            xlabel('Time delay, f_mτ');
            ylabel('Autocorrelation, ϕgg(τ)');
            title('Channel output autocorrelation when f_mT = 0.5');
        end
    end
end
```

2.3.1 *Plot the channel output for* $M = 8, 16$ $(f_m T = 0.01, 0.1, 0.5$ *and* $t\ /\ T = 0 \sim 300)$

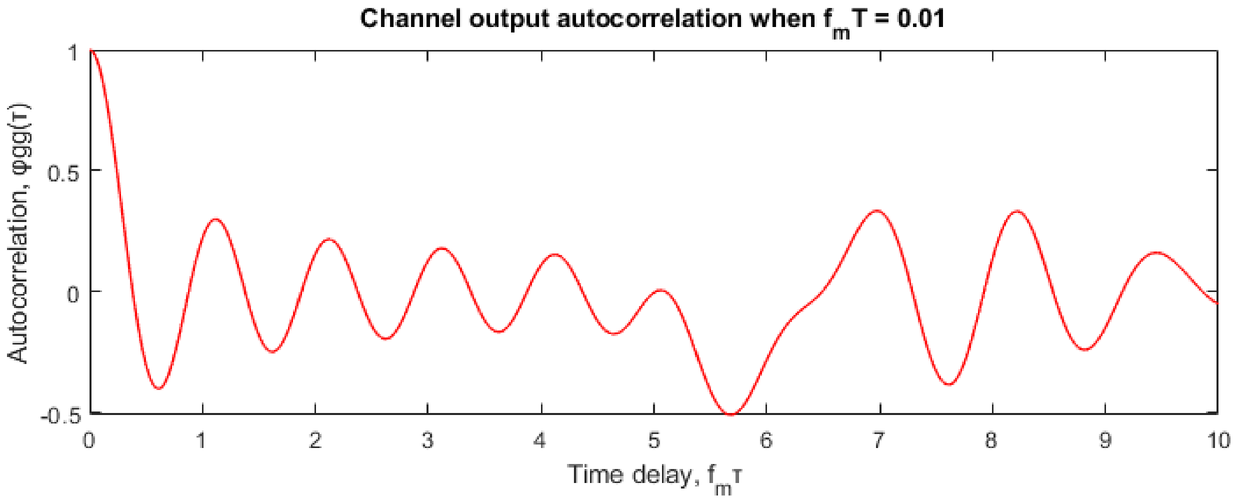
- M = 8

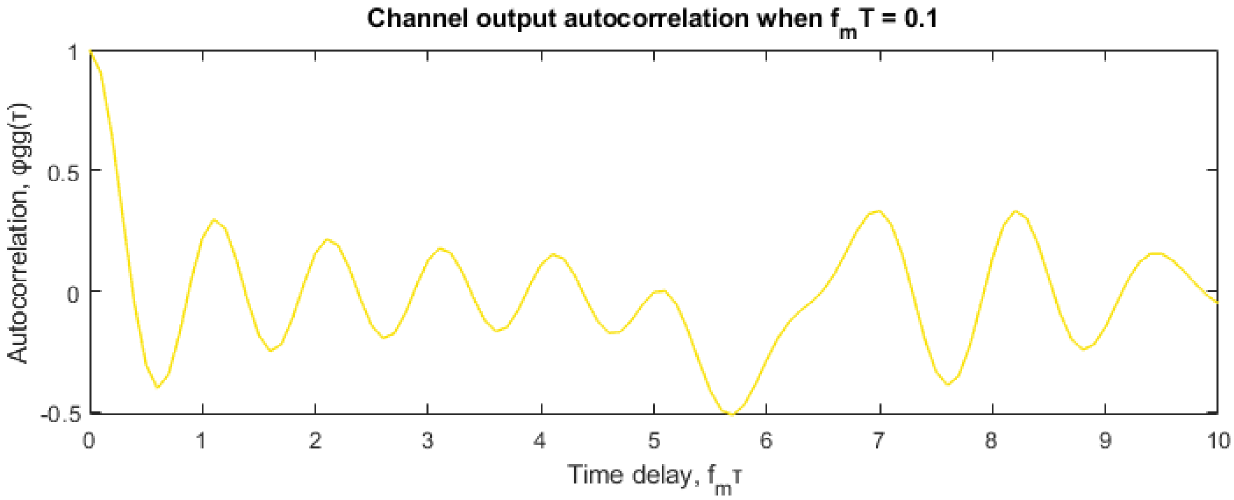


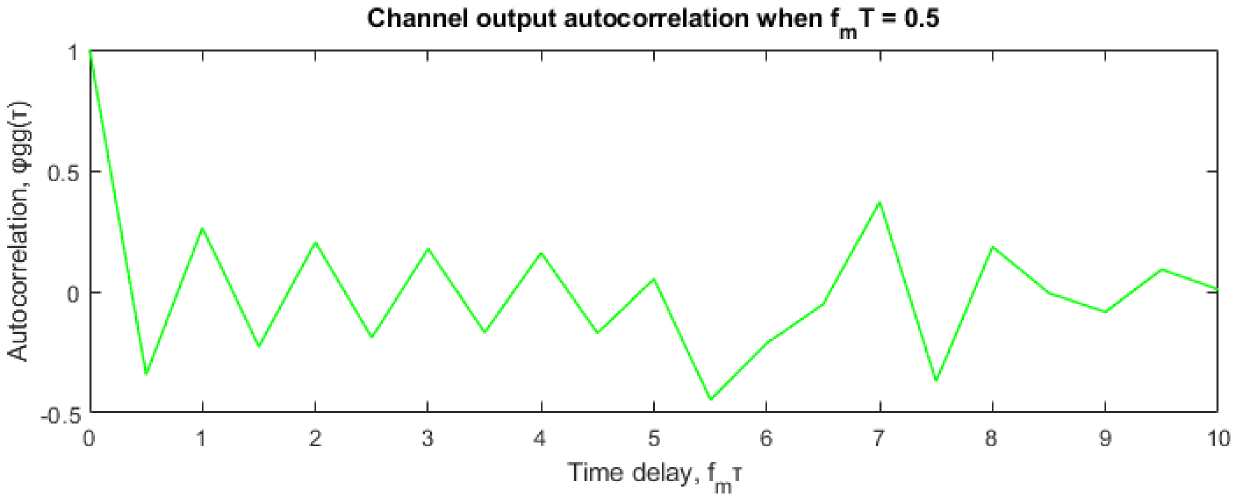




- M = 16

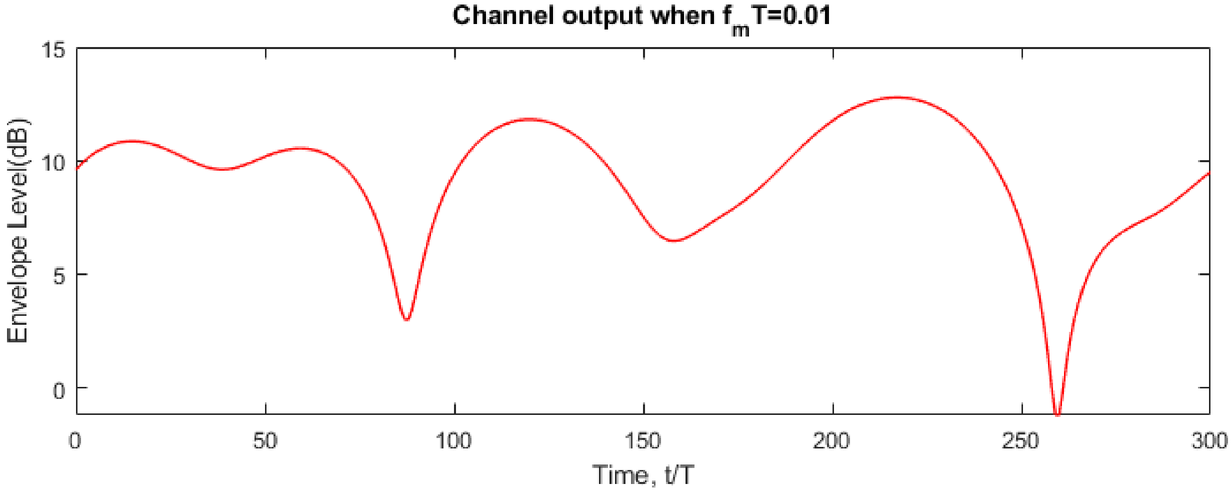

Channel output autocorrelation when $f_m T = 0.01$
Autocorrelation, φgg(τ)
Time delay, $f_m τ$

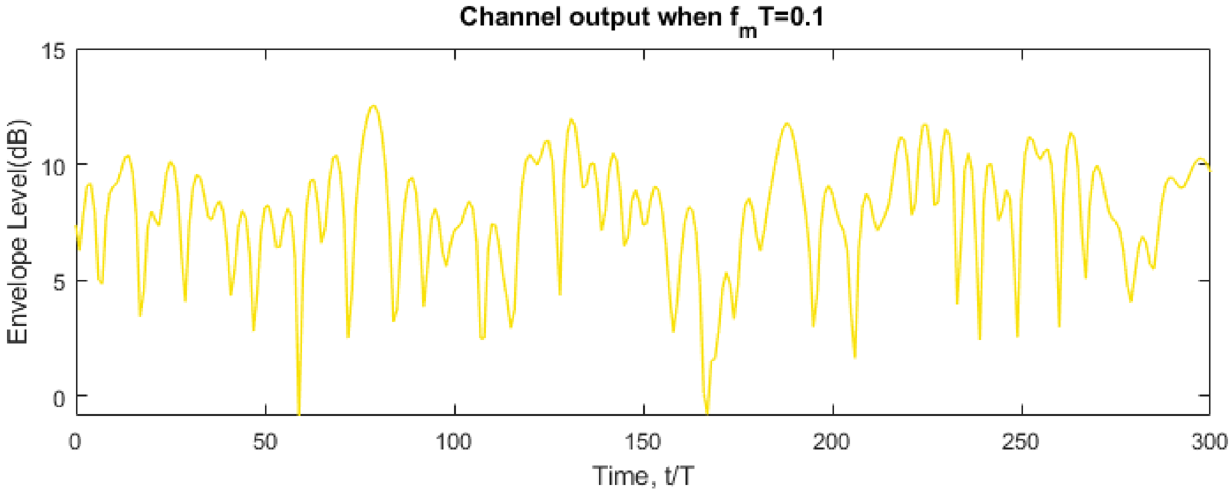

Channel output autocorrelation when $f_m T = 0.1$
Autocorrelation, φgg(τ)
Time delay, $f_m τ$

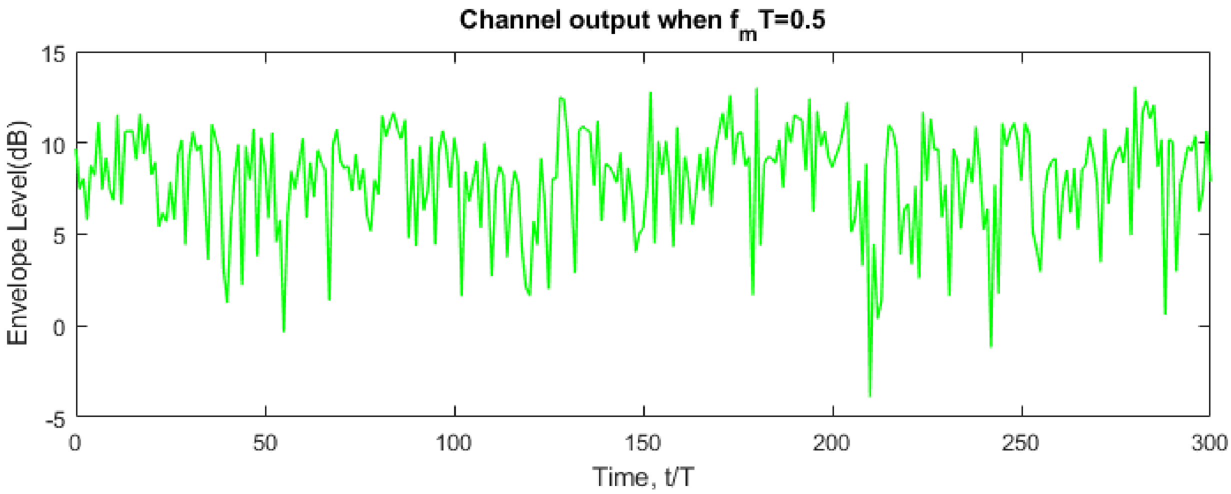

Channel output autocorrelation when $f_m T = 0.5$
Autocorrelation, φgg(τ)
Time delay, $f_m τ$

2.3.2 *Plot the channel output autocorrelation for* $M = 8, 16$ ($f_m\tau = 0\sim10$)

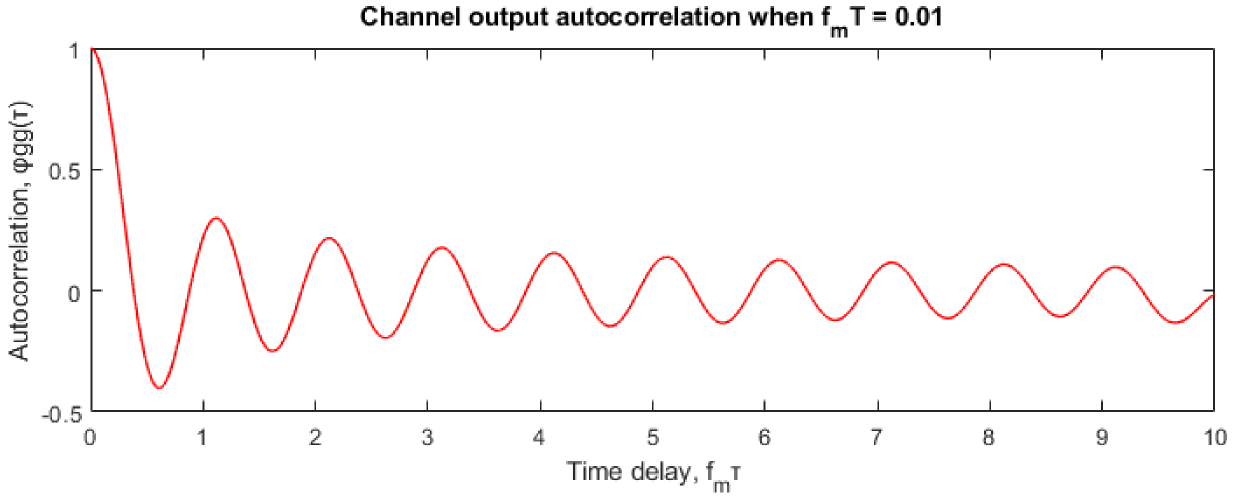
- M = 8

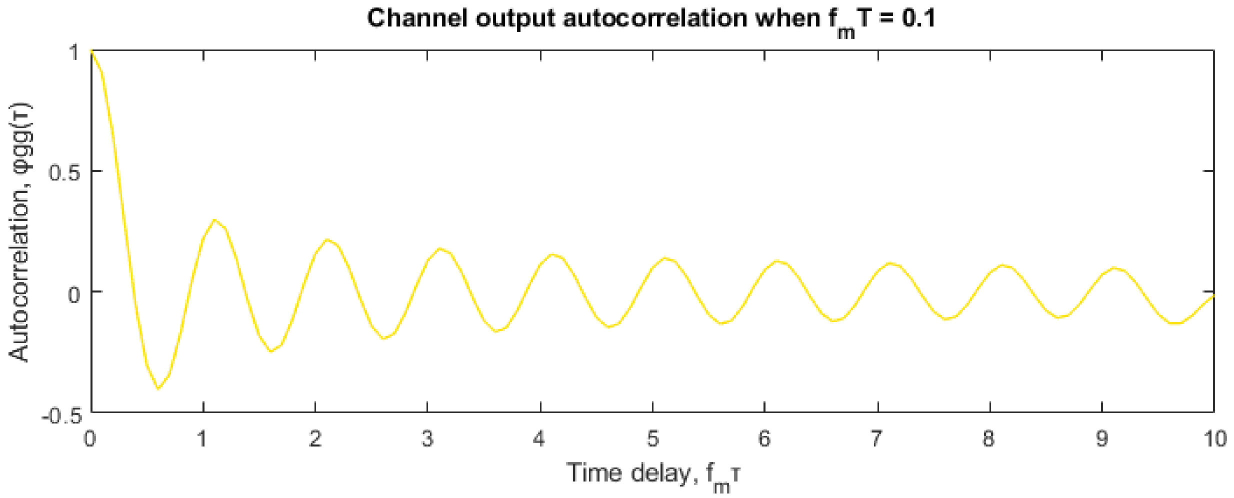


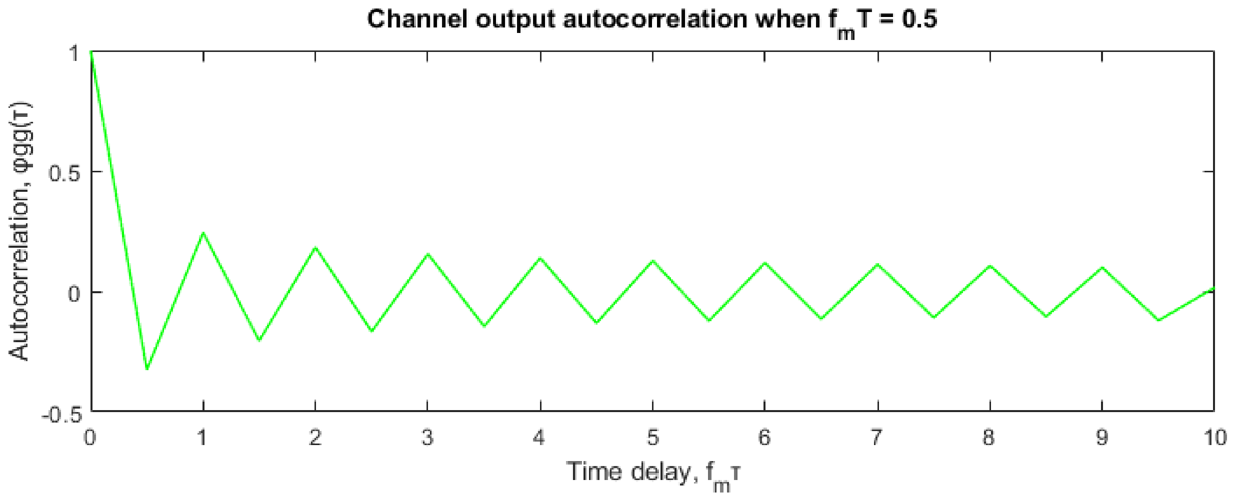




- M = 16


Channel output autocorrelation when f_mT = 0.01
Autocorrelation, φgg(τ)
1
0.5
0
-0.5
0 1 2 3 4 5 6 7 8 9 10
Time delay, $f_m\tau$


Channel output autocorrelation when f_mT = 0.1
Autocorrelation, φgg(τ)
1
0.5
0
-0.5
0 1 2 3 4 5 6 7 8 9 10
Time delay, $f_m\tau$


Channel output autocorrelation when f_mT = 0.5
Autocorrelation, φgg(τ)
1
0.5
0
-0.5
0 1 2 3 4 5 6 7 8 9 10
Time delay, $f_m\tau$

3. Discuss and compare the results of different cases.

3.1 Answer

在第1小題的 *Filtered Gaussian Noise method* 模擬，當 f_m越大，ζ越小，*complex envelope* 的雜訊成分越多自然會變化比較劇烈，而 *autocorrelation* 也慢慢變小。

第 2 小題的 *Sum of Sinusoids method (Jake's method)*和 Filtered Gaussian Noise method 一樣可看出隨著 f_m越大，*complex envelope* 一樣也變化的比較劇烈，且當 *M* 增加，也就是增加 *oscillators* 的數量，可以更貼近真實狀況，*autocorrelation* 較晚才出現失真。

Filtered Gaussian Noise method 產生的結果是離散的，且加上為降低演算法複雜度，實作是用 *first-order LPF* 來取代理想 *LPF*，所以想必會跟實際情況有差距。

Sum of Sinusoids method (Jake's method) 將所需的 *oscillators* 數量降為原本所需約 1/4，也就是降低 75%複雜度，且產生的結果是連續的，所以可以取任何想要的時間點來看，必須注意的是 *Sum of Sinusoids method* 並無隨機性，不過可以透過取不同時間部分來達到偽隨機的效果。