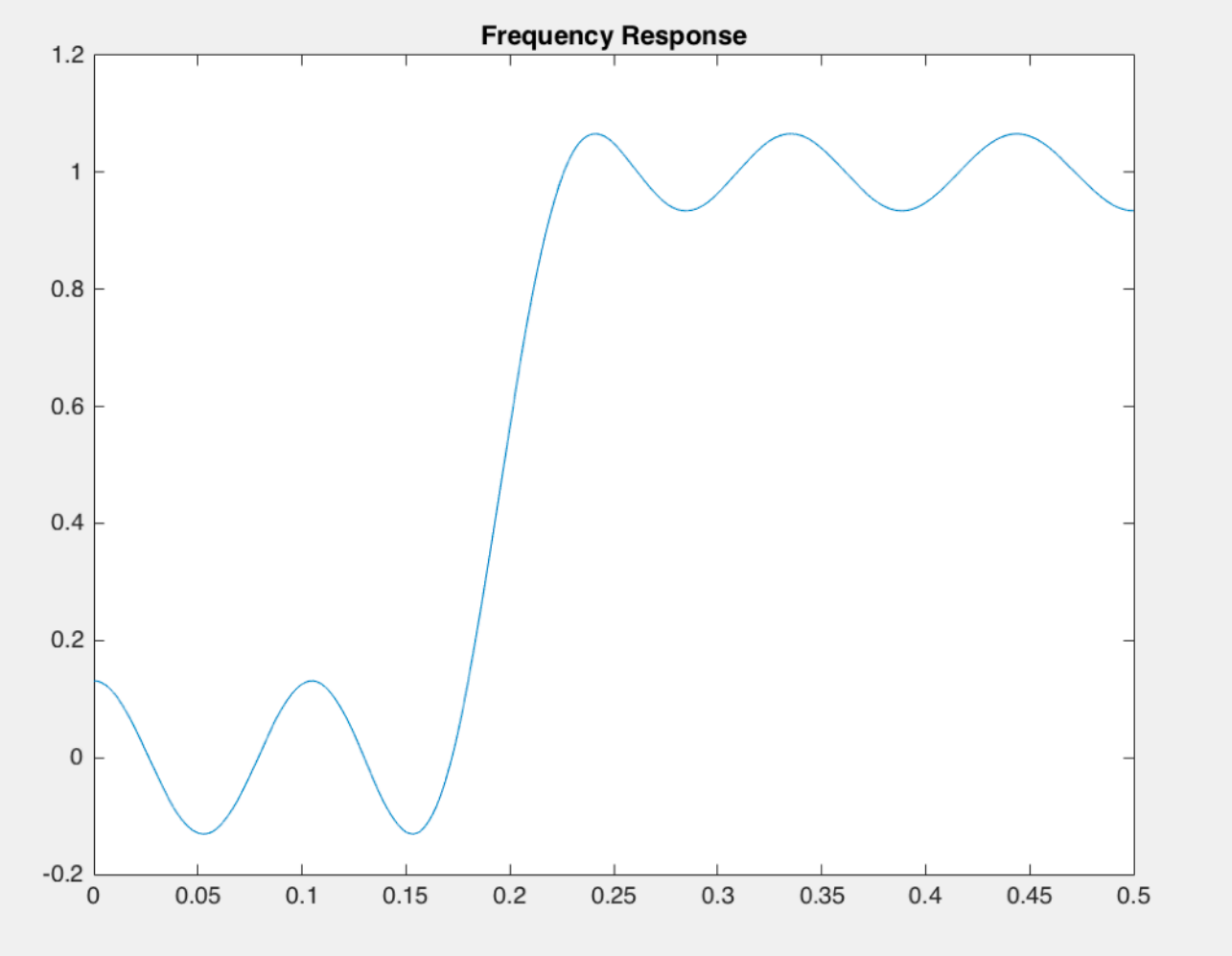
ADSP 高等數位訊號處理, Spring 2017
Homework #1

B03901023 電機三 許秉鈞
指導教授 丁建均

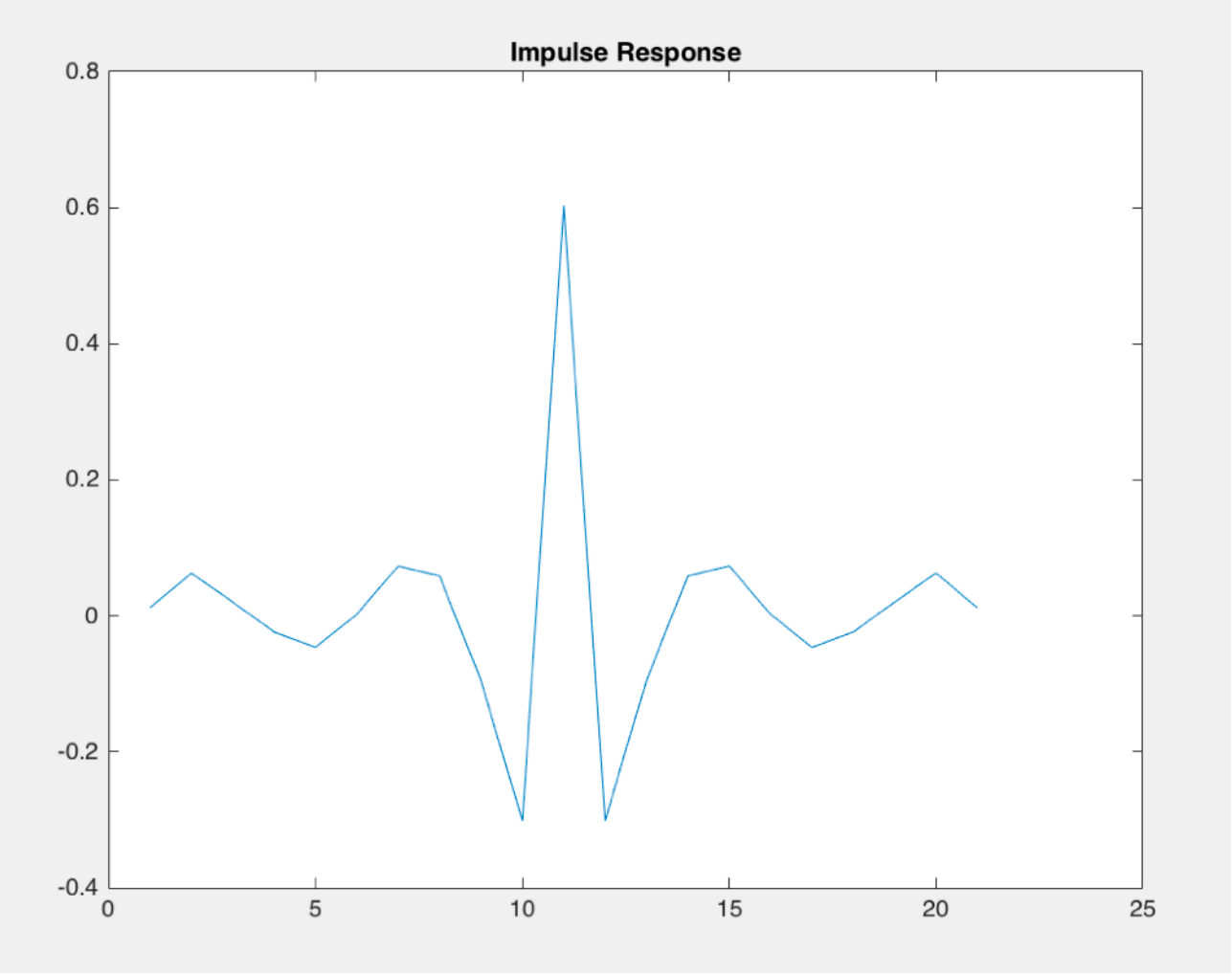
Problem 1:

. (a) MATLAB code （[更新] 請參考最底下）（附於信件）

. (b) The frequency response:



(c) The impulse response h[n]:



(d) The maximal error for each iteration:

一共6個rounds

```
it = 1, E0 = 0.1630
it = 2, E0 = 0.0967
it = 3, E0 = 0.0690
it = 4, E0 = 0.0657
it = 5, E0 = 0.0657
it = 6, E0 = 0.0657 (這個round可以不看)
```

[程式碼]

```
E0 = 0;
E1 = inf;
delta = 0.0001;

N = 21;
k = 10; % (N + 1) / 2
F = zeros(1, k+2); % normalized freq
P = zeros(1, k+2); % local max/min
size = 10000;
p_b = [1100 / 5000, 2500 / 5000]; % pass band
t_b = [900 / 5000, 1100 / 5000]; % transition band

% step 1
for i = 1:k+2
    F(i) = i * 0.04;
    P(i) = i * 0.04;
    if F(i) > t_b(1) && F(i) < t_b(2) % Exclude trans band
        F(i) = 0.33; %0.20 is in trans band-> 0.33
        P(i) = 0.33;
    end
end

it = 0; %iterator, # of iteration
% step 5, Case A requires iteration!
while ((E1 - E0 > delta) || (E1 - E0 < 0))
    it = it + 1;
    it
    F = P;
    E1 = E0;
    W = w(F);
    m = zeros(k+2, k+2);

    % step 2
    m(:,1) = 1;
    m(:,k+2) = (-1).^transpose(0:11)./transpose(W);
    m(:,2:k+1) = cos(2*pi*transpose(F)*(1:10));
    hd = transpose(h(F));
    s = m\hd; % left divide

    x = linspace(0 , 0.5 , size);
    R = zeros(1, size);

    % step 3
    for i = 1:k+1
        R = R + s(i) * cos(2 * pi * (i-1) * x);
    end
    H_F = h(x);
    W_F = w(x);
    err = (R - H_F).*W_F; %element-wise multiply

    % step 4
    [ymax , xmax , ymin , xmin] = extrema(err); % please see ref

    myx = sort_x(xmax, xmin);
    tmp = round(myx * 0.05) / 1000;
```

```
    mat = tmp * 2000 * 10;
    mat(1) = 1;
    E0 = max(abs((R(mat)-h(tmp)) .* w(tmp)));
    E0

    P = tmp;
end

% step 5, case B...
figure;
plot(x , R);
title('Frequency Response');

% step 6, impulse response
h_f = zeros(N , 1);
h_f(k+1) = s(0+1);

for i=1:k
    h_f(k+1+i) = s(1+i) / 2;
    h_f(k+1-i) = s(1+i) / 2;
end

figure;
plot(h_f);
title('Impulse Response');

function H = h(value)

    tmp = 1000 / 5000;
    H = zeros(size(value));
    H((0 <= value) & (value < tmp)) = 0;
    H((tmp <= value) & (value <= 0.5)) = 1;
end
function W = w(value)
    p_b = [1100 / 5000, 2500 / 5000]; % passband
    t_b = [900 / 5000, 1100 / 5000]; % transition band
    s_b = [0, 900 / 5000];

    W = zeros(size(value));
    W((0.0 <= s_b(1)) & (value <= t_b(1))) = 0.5;
    W((p_b(1) <= value) & (value <= p_b(2))) = 1.0;
end
function myx = sort_x(xmax, xmin)
    myx = sort([xmax , xmin]);
end
function [xmax,imax,xmin,imin] = extrema(x)
%ref: https://www.mathworks.com/matlabcentral/fileexchange/12275-extrema-
m
%--extrema2-m?focused=6267317&tab=function
% OPEN SOURCE CODE

%EXTREMA   Gets the global extrema points from a time series.
xmax = [];
imax = [];
xmin = [];
imin = [];
```

```
% Vector input?
Nt = numel(x);
if Nt ~= length(x)
 error('Entry must be a vector.')
end

% NaN's:
inan = find(isnan(x));
indx = 1:Nt;
if ~isempty(inan)
 indx(inan) = [];
 x(inan) = [];
 Nt = length(x);
end

% Difference between subsequent elements:
dx = diff(x);

% Is an horizontal line?
if ~any(dx)
 return
end

% Flat peaks? Put the middle element:
a = find(dx~=0);              % Indexes where x changes
lm = find(diff(a)~=1) + 1;    % Indexes where a do not changes
d = a(lm) - a(lm-1);          % Number of elements in the flat peak
a(lm) = a(lm) - floor(d/2);   % Save middle elements
a(end+1) = Nt;

% Peaks?
xa  = x(a);             % Serie without flat peaks
b = (diff(xa) > 0);     % 1  =>  positive slopes (minima begin)
                        % 0  =>  negative slopes (maxima begin)
xb  = diff(b);          % -1 =>  maxima indexes (but one)
                        % +1 =>  minima indexes (but one)
imax = find(xb == -1) + 1; % maxima indexes
imin = find(xb == +1) + 1; % minima indexes
imax = a(imax);
imin = a(imin);

nmaxi = length(imax);
nmini = length(imin);

% Maximum or minumim on a flat peak at the ends?
if (nmaxi==0) && (nmini==0)
 if x(1) > x(Nt)
  xmax = x(1);
  imax = indx(1);
  xmin = x(Nt);
  imin = indx(Nt);
 elseif x(1) < x(Nt)
  xmax = x(Nt);
  imax = indx(Nt);
```

```
  xmin = x(1);
  imin = indx(1);
 end
 return
end

% Maximum or minumim at the ends?
if (nmaxi==0)
 imax(1:2) = [1 Nt];
elseif (nmini==0)
 imin(1:2) = [1 Nt];
else
 if imax(1) < imin(1)
  imin(2:nmini+1) = imin;
  imin(1) = 1;
 else
  imax(2:nmaxi+1) = imax;
  imax(1) = 1;
 end
 if imax(end) > imin(end)
  imin(end+1) = Nt;
 else
  imax(end+1) = Nt;
 end
end
xmax = x(imax);
xmin = x(imin);

% NaN's:
if ~isempty(inan)
 imax = indx(imax);
 imin = indx(imin);
end

% Same size as x:
imax = reshape(imax,size(xmax));
imin = reshape(imin,size(xmin));

% Descending order:
[temp,inmax] = sort(-xmax); clear temp
xmax = xmax(inmax);
imax = imax(inmax);
[xmin,inmin] = sort(xmin);
imin = imin(inmin);

end
% Carlos Adrin Vargas Aguilera. nubeobscura@hotmail.com
```