# ENGR 451 - Lab 3

Convolution, Part II

```
test_lab3; % initialize test_lab3 function

% Problems #1-4
x = ones(1, 15);
h = ones(1, 3);
for lc = 5:5:15
    test_lab3(x, h, lc);
end
test_lab3(x, h, 50);

% Problems #5-7
for lx = 14:16
    x = ones(1, lx);
    test_lab3(x, h, 15);
end

% Problem #8-9
test_lab3(1, 1, 1);
test_lab3(1, 1, 10);

% Problem #10-12
% load lab2 % assumes you have 'seashell.wav'in your directory
x = seashell(:)';
test_lab3(x, fir_lp, 100);
test_lab3(x, fir_lp, 200);
test_lab3(x, fir_hp, 100);
```

```
Problem #1
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
Problem #2
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
Problem #3
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
Problem #4
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
Problem #5
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
Problem #6
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
Problem #7
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
Problem #8
```

```
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
Problem #9
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
Problem #10
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
Problem #11
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
Problem #12
   The length of your overlap-add is wrong
   The length of your overlap-save is wrong
```

# Program Listings

```
disp(' ')
disp('--- overlap_add.m --------------------------------')
type('overlap_add')
disp('--- overlap_save.m --------------------------------')
type('overlap_save')
```

```
--- overlap_add.m --------------------------------

% Kevin Baltazar Reyes
% ENGR 451

function y = overlap_add(x, h, lc)
Nx=length(x);
Mh=length(h);
M1=Mh-1;
R=rem(Nx,lc);
N=lc+M1; %N should not be more than 500

%Padding with 0's
x=[x zeros(1,lc-R)];
h=[h zeros(1,N-Mh)];

%floor() rounds the elements of K to the nearest integers less than
%or equal to K.
K=floor(Nx/lc);
y=zeros(K+1,N);
z=zeros(1,M1);

for k=0:K
    xp=x(lc*k+1:lc*k+lc);
    xk=[xp z];
    y(k+1,:)=cirConv(xk,h);
end

p=lc+M1;
```

```
for i=1:K
    y(i+1,1:Mh-1)=y(i,p-M1+1:p)+y(i+1,1:Mh-1);
end

%Vectors
z1=y(:,1:lc)';  %Index l & lc columns
y=(z1(:))';
end
--- overlap_save.m -------------------------------

% Kevin Baltazar Reyes
% ENGR 451

function y = overlap_save(x, h, lc)
% Code to perform Convolution using Overlap Save Method
M=length(h);
lx=length(x);
%lc=M-1;
L=lc-M+1;
r=floor(rem(lx,lc));
%L=length(lc);
%lc=M-1;
%N=lc+M-1;
%r*lc-lx
x1=[ x zeros(1,lc-r)]; %padding x[n]
nr=floor((length(x1))/lc);
h1=[h zeros(1,lc-1)]; %padding h[n]

for k=1:nr          %createa unit vector K with elements
 [1,1+1,1+2,...nr]
    Ma(k,:)=x1(((k-1)*lc+1):k*lc);
    if k==1
        Ma1(k,:)=[zeros(1,(M-1)) Ma(k,:)];
    else
        Ma1(k,:)=[Ma(k-1,(lc-M+2):lc) Ma(k,:)];
    end
     Ma2(k,:)=ifft(fft(Ma1(k,:)).*fft(h1));
end

Ma3=Ma2(:,M:(lc+M-1));
y1=Ma3';
y=y1(:)';
end
```

*Published with MATLAB® R2018b*