EE 321 Example of Truncated Fourier Series Fall 2012

The Matlab code below computes and plots a truncated Fourier series for a square wave, defined over one period of T seconds as $x(t)=\begin{cases}0, & \frac{-T}{2}<t\le 0\\ 1, & 0<t\le \frac{T}{2}\end{cases}$. The Fourier series coefficients are $a_0=0.5$, $a_n=0,\ \ n\ge 1$, $b_n=\dfrac{[1-\cos(n\pi)]}{n\pi},\ \ n\ge 1$.

```
function truncatedFS(N,T)
% This function computes an N-term truncated Fourier series
% and plots it over the interval [-T, 2T] seconds
% The Fourier series coefficients a0, a1,...,aN, and b1,...bN
% must be defined below. The example considers a square wave
% with x(t) = 0 for -T/2 < t < 0, x(t) = 1 for 0 <= t <= T/2.
a0=0.5;
n=[1:N];
a=zeros(1,N); % The an coefficients are all zero.
b=(1./(n*pi)).*(1-cos(n*pi)); % The bn coefficients.
t=[-T:T/100:2*T]; % Define time range with time step size T/100
x=a0*ones(1,length(t)); % zeroeth-order truncted Fourier series
for k=1:N % loop through all N Fourier coefficients
   x=x+a(k)*cos(2*pi*k*t/T)+b(k)*sin(2*pi*k*t/T);
end
% Repeat using magnitude/phase form of Fourier series
y=a0*ones(1,length(t));
A=sqrt(a.*a + b.*b);
theta=atan2(b,a); % phase in radians
for k=1:N
   y=y+A(k)*cos(2*pi*k*t/T-theta(k));
end
figure(1)
plot(t,x,t,y,'.')
xlabel('Time, t, sec')
ylabel('x(t)')
title('Truncated Fourier Series (Sine/Cosine and Magnitude/Phase Forms)')
```

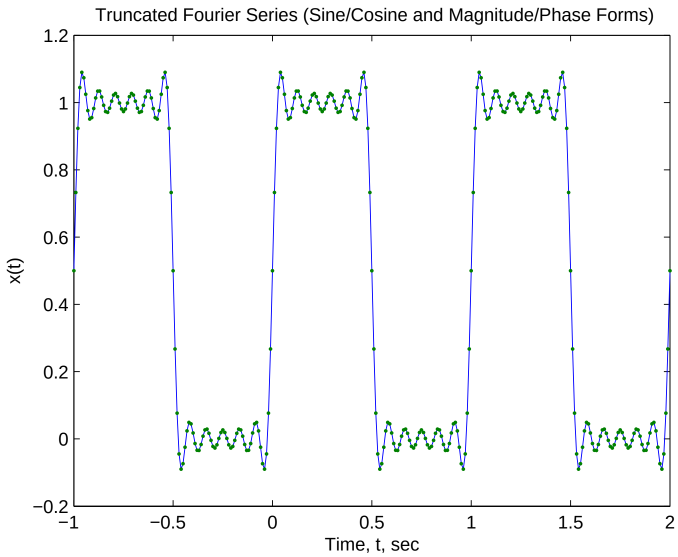


Figure 1. Truncated Fourier series, N = 11, T = 1.

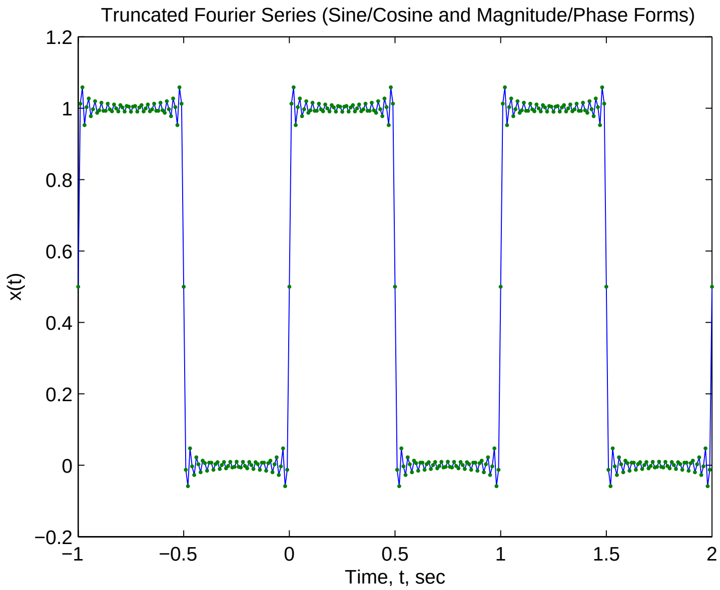


Figure 2. Truncated Fourier series, N = 31, T = 1.

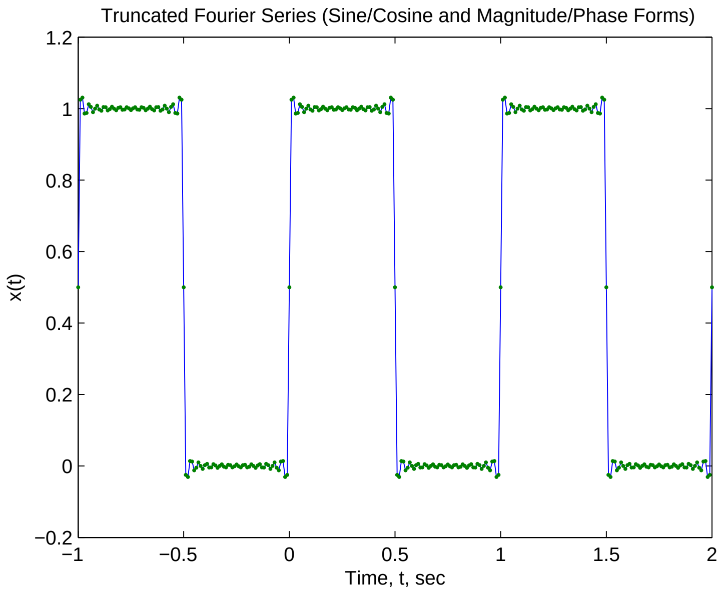
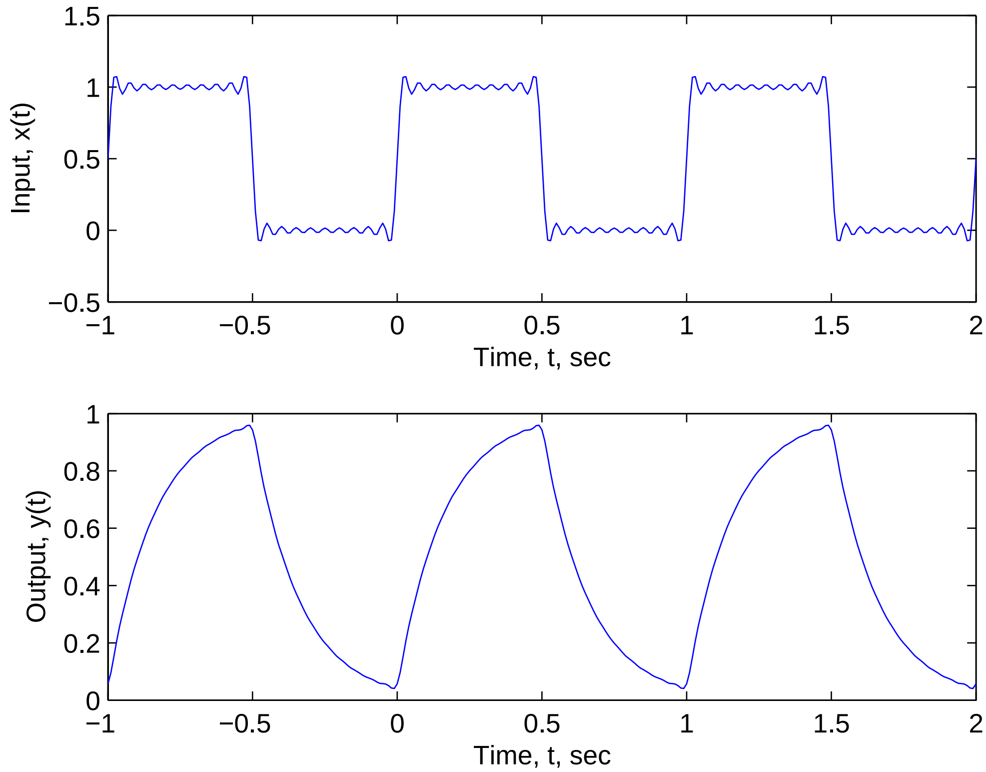


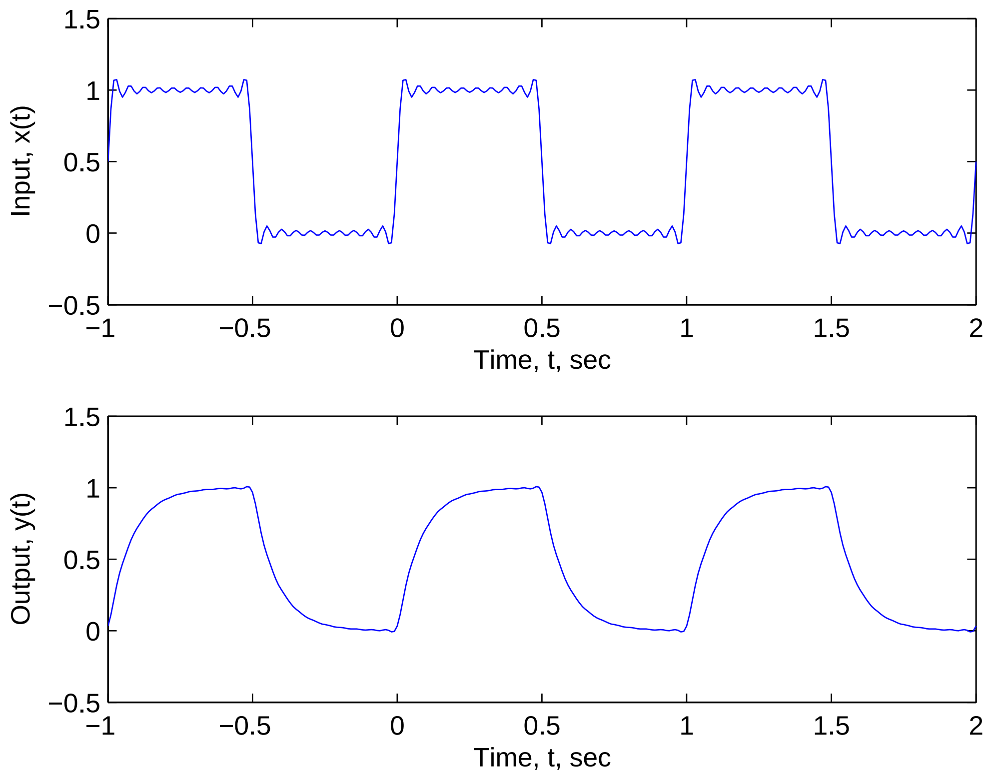
Figure 3. Truncated Fourier series, N = 71, T = 1.

Now, consider filtering a periodic waveform. If the input waveform, $x(t)$, has magnitude/phase Fourier series $x(t) = a_0 + \sum_{n=1}^{\infty} A_n \cos(n\omega_0 t - \theta_n)$, where $A_n = \sqrt{a_n^2 + b_n^2}$, $\theta_n = \tan^{-1}\frac{b_n}{a_n}$, then the output of a filter with input x(t) and transfer function $H(s)$ is $y(t) = H(0)a_0 + \sum_{n=1}^{\infty} A_n |H(jn\omega_0)| \cos(n\omega_0 t - \theta_n + \beta_n)$, where $\beta_n = \text{phase}\left(H(jn\omega_0)\right)$.

```
function truncatedFS(N,T,alpha)
% This function computes an N-term truncated Fourier series and plots it over the
% interval [-T, 2T] seconds. The Fourier series coefficients a0, a1,...,aN, and b1,...bN
% must be defined below. The example considers a square wave
% with x(t) = 0 for -T/2 < t < 0, x(t) = 1 for 0 <= t <= T/2.
% The Fourier series is then filtered using a transfer
% function H(s) = (alpha*w0)/(s + alpha*w0), where alpha > 0.
% For the example, the value of w0 is set to 2*pi/T, so the filter - 3 dB
% frequency is set at alpha times the first harmonic frequency of the input square wave
a0=0.5; n=[1:N];
a=zeros(1,N); % The an coefficients are all zero.
b=(1./(n*pi)).*(1-cos(n*pi)); % The bn coefficients.
t=[-T:T/100:2*T]; % Define time range with time step size T/100
x=a0*ones(1,length(t)); % zeroeth-order truncted Fourier series
for k=1:N % loop through all N Fourier coefficients
   x=x+a(k)*cos(2*pi*k*t/T)+b(k)*sin(2*pi*k*t/T);
end
% Repeat using magnitude/phase form of Fourier series
x1=a0*ones(1,length(t));
A=sqrt(a.*a + b.*b);
theta=atan2(b,a); % phase in radians
for k=1:N
   x1=x1+A(k)*cos(2*pi*k*t/T-theta(k));
end
figure(1)
plot(t,x,t,x1,'.')
xlabel('Time, t, sec')
ylabel('x(t)')
title('Truncated Fourier Series (Sine/Cosine and Magnitude/Phase Forms)')
% Now filter the input waveform, using the magnitude/phase form of FS
w0=2*pi/T;
wc=alpha*2*pi/T;
H0=1;H=wc./(j*n*w0 + wc);
y=H0*a0*ones(1,length(t)); % filter the a0 term
for k=1:N
   y=y+abs(H(k))*A(k)*cos(2*pi*k*t/T-theta(k)+phase(H(k)));
end
figure(2)
subplot(2,1,1)
plot(t,x1)
xlabel('Time, t, sec')
ylabel('Input, x(t)')
subplot(2,1,2)
plot(t,y)
xlabel('Time, t, sec')
ylabel('Output, y(t)')
```



Filtered square wave, N = 19, alpha = 1 so cutoff frequency = 1st harmonic frequency.



Filtered square wave, N = 19, alpha = 2, so cutoff frequency = 2nd harmonic frequency.