Example- In this example, we will use the FFT to approximate the Fourier transform of the continuous-time (CT) function -

$$x(t) = e^{-t}u(t).$$

Using the transform pair,

$$e^{-bt}u(t) \quad \leftrightarrow \quad \frac{1}{j\omega + b} \quad (b > 0) \;\; -\infty < \omega < \infty,$$

the exact/analytical Fourier transform of $x(t)$ is

$$X(\omega) = \frac{1}{j\omega + 1}, \quad -\infty < \omega < \infty.$$

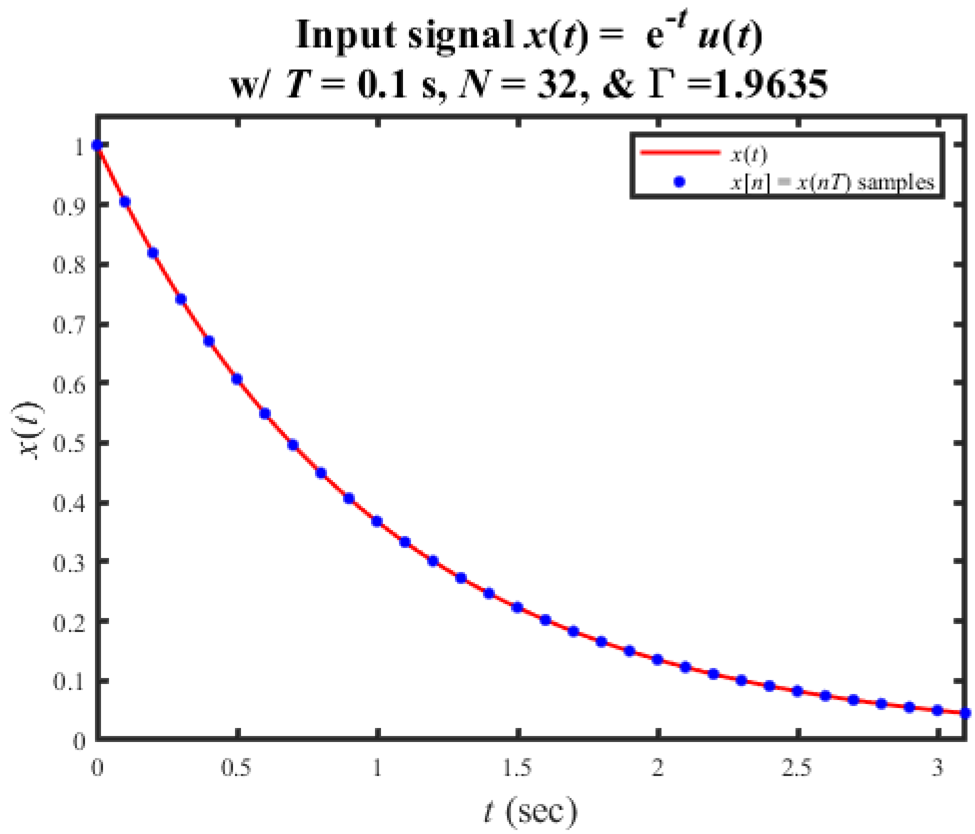
From the expression for $x(t)$, we see that $x(t \rightarrow \infty) \rightarrow 0$. This makes it possible to approximate the Fourier transform using the FFT/DFT.

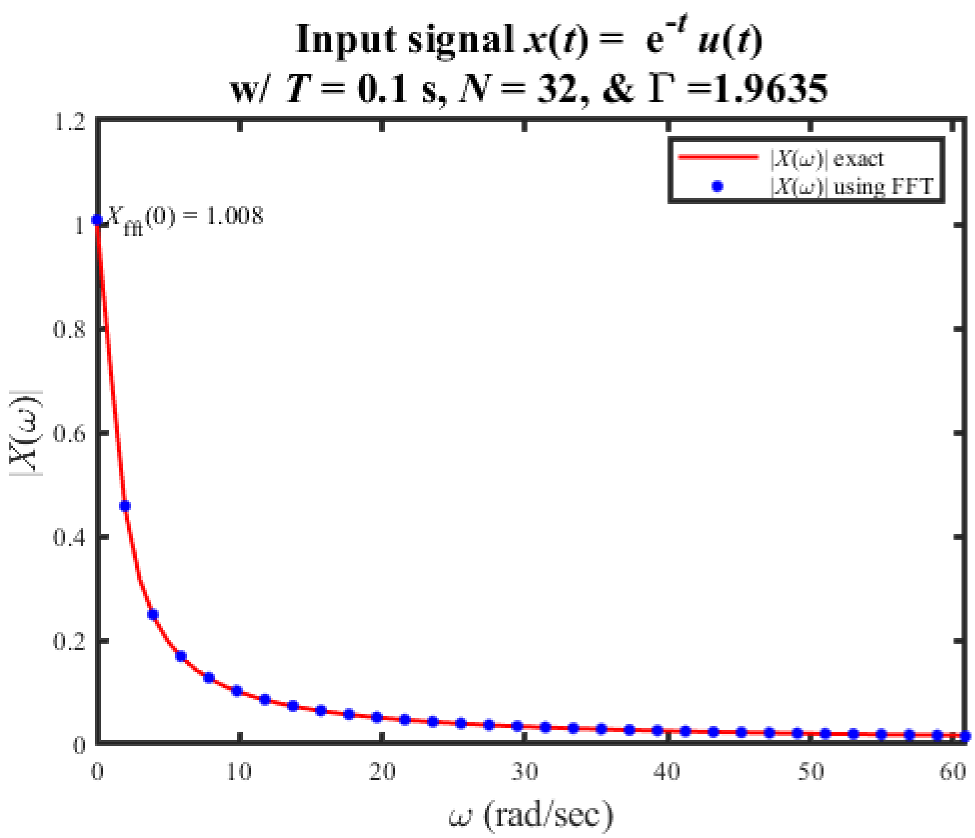
First, I will implement the approximation to the Fourier transform myself for N = 32 datapoints and a T = 0.1 s sampling period, using the MATLAB fft() function and class notes. The m-file is chap_04_fft_example_2a.m.

Second, I will use the MATLAB contfft() function for N = 32 and T = 0.1 s to demonstrate that we get the same results (with less coding needed). The m-file is chap_04_fft_example_2b.m

Then, I will use the MATLAB contfft() function for varying data points N and sampling periods T to illustrate how they can affect the results.

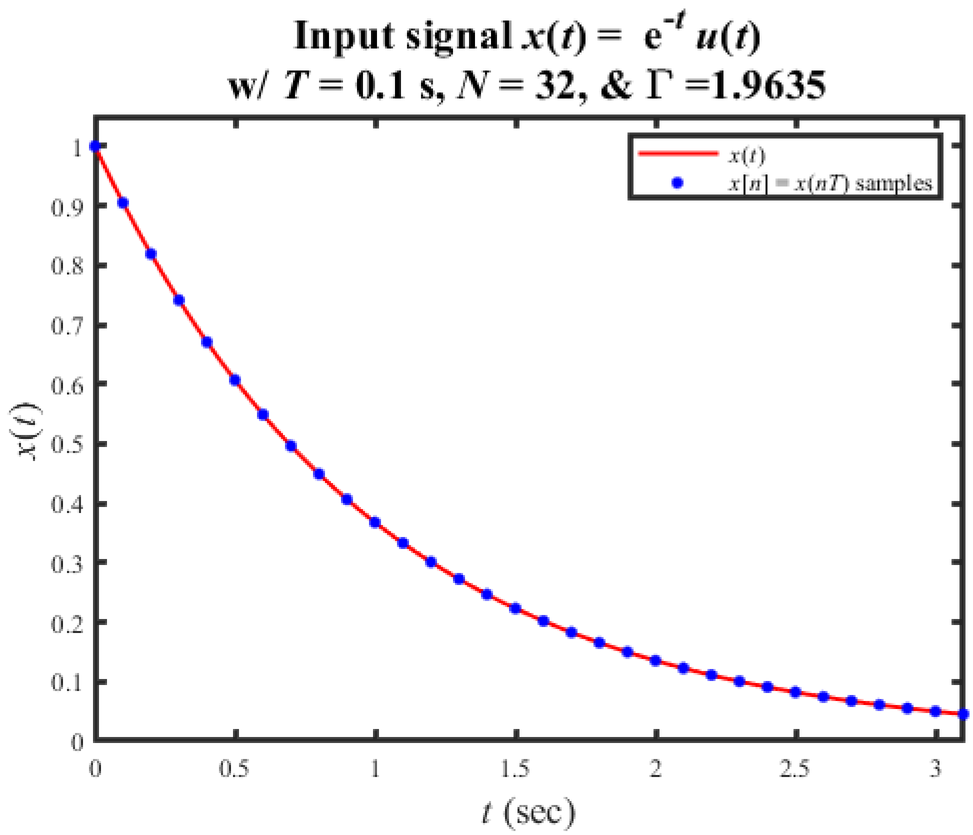
```
% Chapter 4 FFT Example 2 (chap_04_fft_example_2a.m)
% Use FFT to approximate the Fourier transform of
%            x(t) = e^-t u(t)
% at discrete frequencies wk = 2*pi*k/N/T and
% compare with the exact answer X[w] = 1/(jw+1)
% over the frequency range 0 < w < 2*pi*(N-1)/N/T.
%
% This m-file implements an FFT approximation of the
% Fourier transform X(w) directly.
%
clear;clc;close all;
T = 0.1; N = 32;     % Pick number of points & sampling rate
t = 0:0.02:(N-1)*T; % define time vector for exact function
x = exp(-t);         % exact input function
nT = 0:T:(N-1)*T;    % time vector at sampling rate
xn = exp(-nT);       % DT samples of exact x(t)
%
k = 0:1:N-1;  % FFT indices
k(1)= eps;    % replace k = 0 to avoid divide by zero error
Gamma = 2*pi/N/T;
wk = k*Gamma; % Discrete frequency points
w = 0:1:N*Gamma; % Continuous frequency points
X = 1./(j*w+1); % exact expression for Fourier transform
% Fourier transform of xn[n]=x(nT) & approx. to X(w)
Xk = fft(xn); Xfft = Xk.*(1-exp(-j*2*pi*k/N))./(j*2*pi*k/N/T);
Xmagfft = abs(Xfft);  % FFT approximate FT spectrum
Xmag = abs(X);        % exact FT spectrum
% Plot input signal
plot(t,x,'r-',nT,xn,'b.'),axis([0 (N-1)*T 0 1.05]),
ylabel('{\itx}({\itt})','fontsize',14,'fontname','times'),
xlabel('{\itt} (sec)','fontsize',14,'fontname','times'),
title({'Input signal {\itx}({\itt}) =  e^{-{\itt} }{\itu}({\itt})';...
    ['w/ {\itT} = ',num2str(T),' s, {\itN} = ',num2str(N),...
    ', & \Gamma =',num2str(Gamma)]},'fontsize',16,'fontname','times'),
legend(' {\itx}({\itt})',' {\itx}[{\itn}] = {\itx}({\itnT}) samples'),
% Plot Fourier transform of signal
figure,plot(w,Xmag,'r-',wk,Xmagfft,'b.'), axis([0 (N-1)*Gamma 0 1.2]),
text(wk(1),Xmagfft(1),[' {\itX}_{fft}(0) = ',num2str(Xmagfft(1))],...
       'horizontalalignment','left','verticalalignment','middle'),
ylabel('|{\itX}(\omega)|','fontsize',14,'fontname','times'),
xlabel('\omega (rad/sec)','fontsize',14,'fontname','times'),
title({'Input signal {\itx}({\itt}) =  e^{-{\itt} }{\itu}({\itt})';...
    ['w/ {\itT} = ',num2str(T),' s, {\itN} = ',num2str(N),...
    ', & \Gamma =',num2str(Gamma)]},'fontsize',16,'fontname','times'),
legend(' |{\itX}(\omega)| exact',' |{\itX}(\omega)| using FFT'),
set(findobj('type','line'),'linewidth',1.5,'markersize',14)
set(findobj('type','axes'),'linewidth',2,'fontname','times')
set(findobj('type','text'),'fontname','times')
```

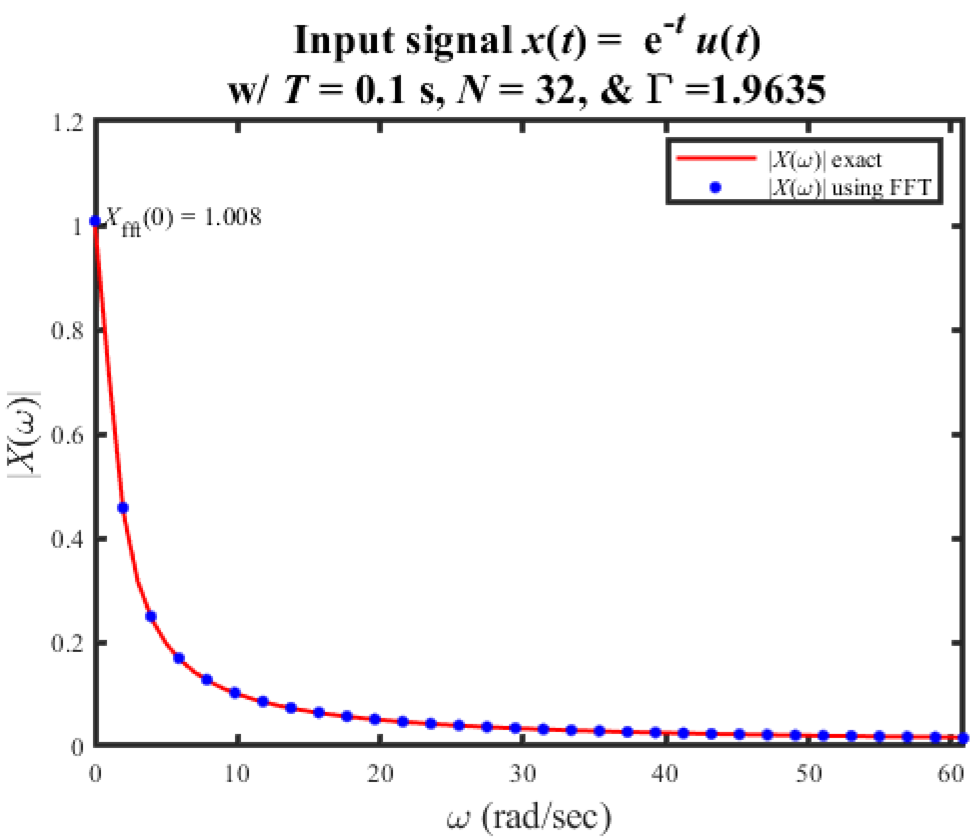




- Amazingly good agreement, this might be a ‘lucky sweet spot’.

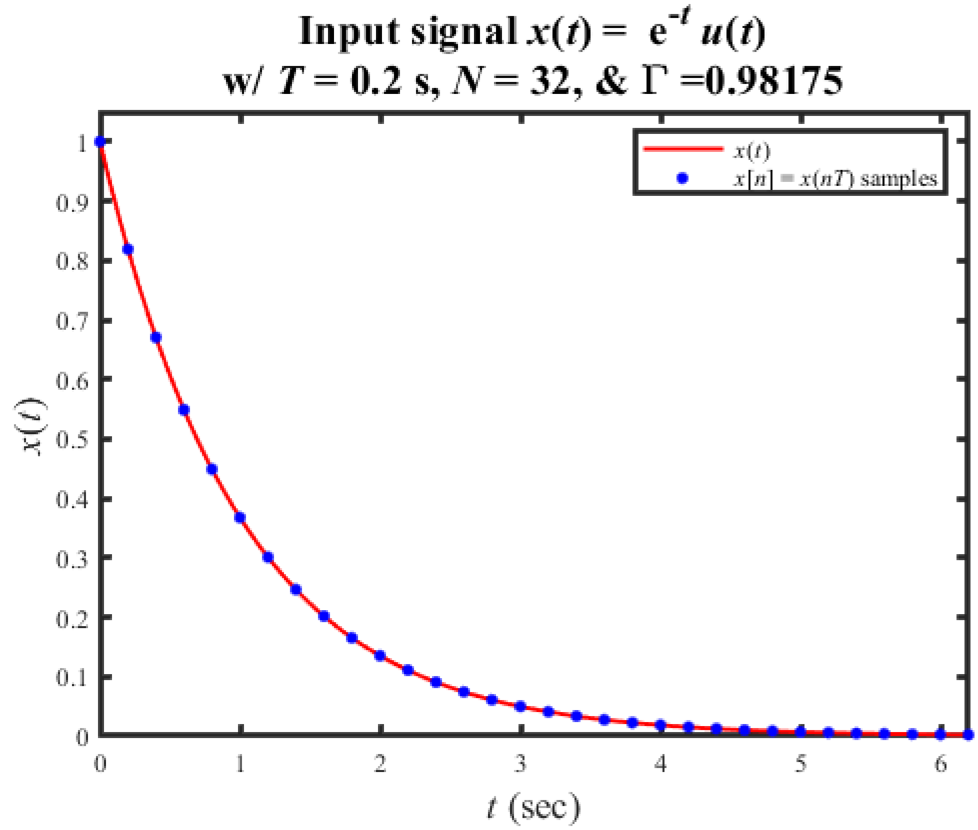
```
% Chapter 4 FFT Example 2 (chap_04_fft_example_2b.m)
% Use FFT to approximate the Fourier transform of
%           x(t) = e^-t u(t)
% at discrete frequencies wk = 2*pi*k/N/T and
% compare with the exact answer X[w] = 1/(jw+1)
% over the frequency range 0 < w < 2*pi*(N-1)/N/T.
%
% This m-file uses the function contfft() to
% compute the FFT approximation of X(w).
%
clear;clc;close all;
T = 0.1; N = 32;    % Pick number of points & sampling rate
t = 0:0.02:(N-1)*T; % define time vector for exact function
x = exp(-t);        % exact input function
nT = 0:T:(N-1)*T;   % time vector at sampling rate
xn = exp(-nT);      % DT samples of exact x(t)
%
Gamma = 2*pi/N/T;
w = 0:1:N*Gamma; % Continuous frequency points
X = 1./(j*w+1); % exact expression for Fourier transform
% Fourier transform of xn[n] = x(nT) & fft approx. to X(w)
[Xfft,wk] = contfft(xn,T);
Xmagfft = abs(Xfft);  % FFT FT approx. spectrum
Xmag = abs(X);        % exact FT spectrum
% Plot input signal
plot(t,x,'r-',nT,xn,'b.'),axis([0 (N-1)*T 0 1.05]),
ylabel('{\itx}({\itt})','fontsize',14,'fontname','times'),
xlabel('{\itt} (sec)','fontsize',14,'fontname','times'),
title({'Input signal {\itx}({\itt}) =  e^{-{\itt} }{\itu}({\itt})';...
    ['w/ {\itT} = ',num2str(T),' s, {\itN} = ',num2str(N),...
    ', & \Gamma =',num2str(Gamma)]},'fontsize',16,'fontname','times'),
legend(' {\itx}({\itt})',' {\itx}[{\itn}] = {\itx}({\itnT}) samples'),
% Plot Fourier transform of signal
figure,plot(w,Xmag,'r-',wk,Xmagfft,'b.'), axis([0 (N-1)*Gamma 0 1.2]),
text(wk(1),Xmagfft(1),[' {\itX}_{fft}(0) = ',num2str(Xmagfft(1))],...
        'horizontalalignment','left','verticalalignment','middle'),
ylabel('|{\itX}(\omega)|','fontsize',14,'fontname','times'),
xlabel('\omega (rad/sec)','fontsize',14,'fontname','times'),
title({'Input signal {\itx}({\itt}) =  e^{-{\itt} }{\itu}({\itt})';...
    ['w/ {\itT} = ',num2str(T),' s, {\itN} = ',num2str(N),...
    ', & \Gamma =',num2str(Gamma)]},'fontsize',16,'fontname','times'),
legend(' |{\itX}(\omega)| exact',' |{\itX}(\omega)| using FFT'),
set(findobj('type','line'),'linewidth',1.5,'markersize',14)
set(findobj('type','axes'),'linewidth',2,'fontname','times')
set(findobj('type','text'),'fontname','times')
```

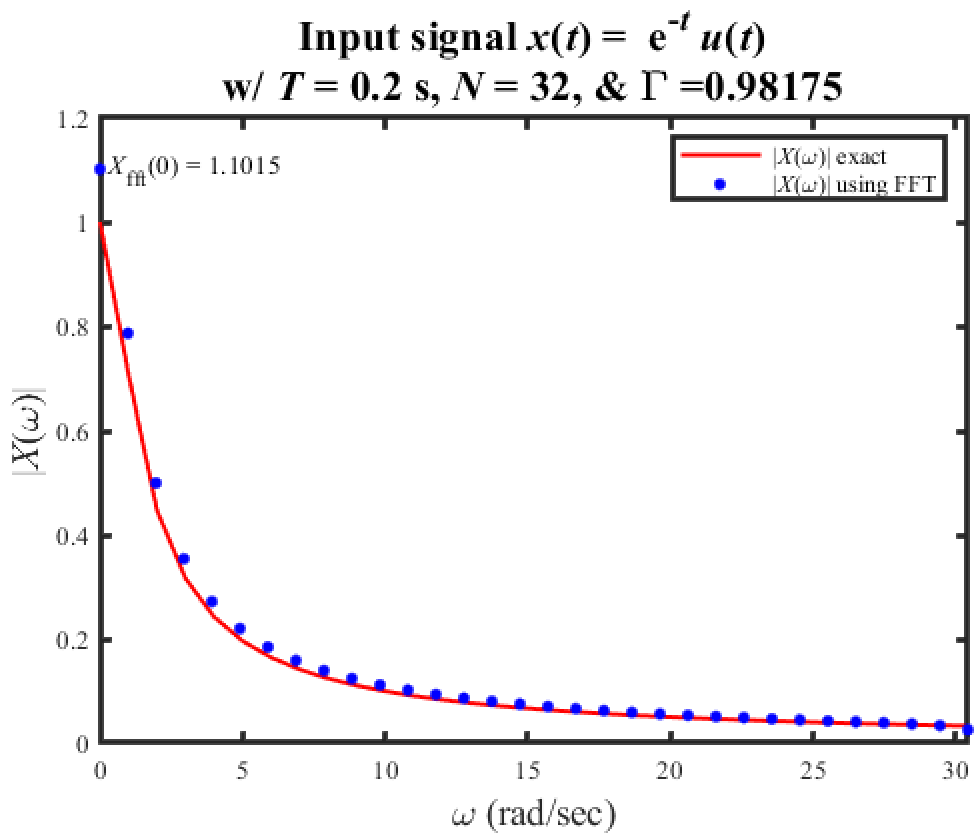




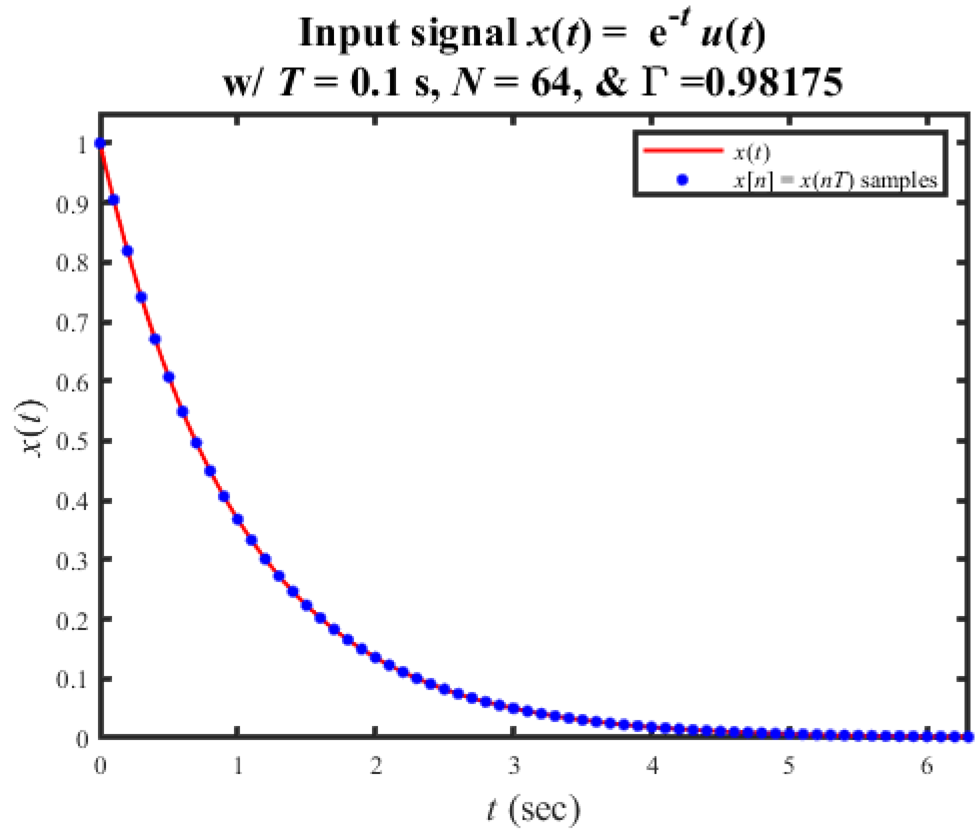
- **Same results!**

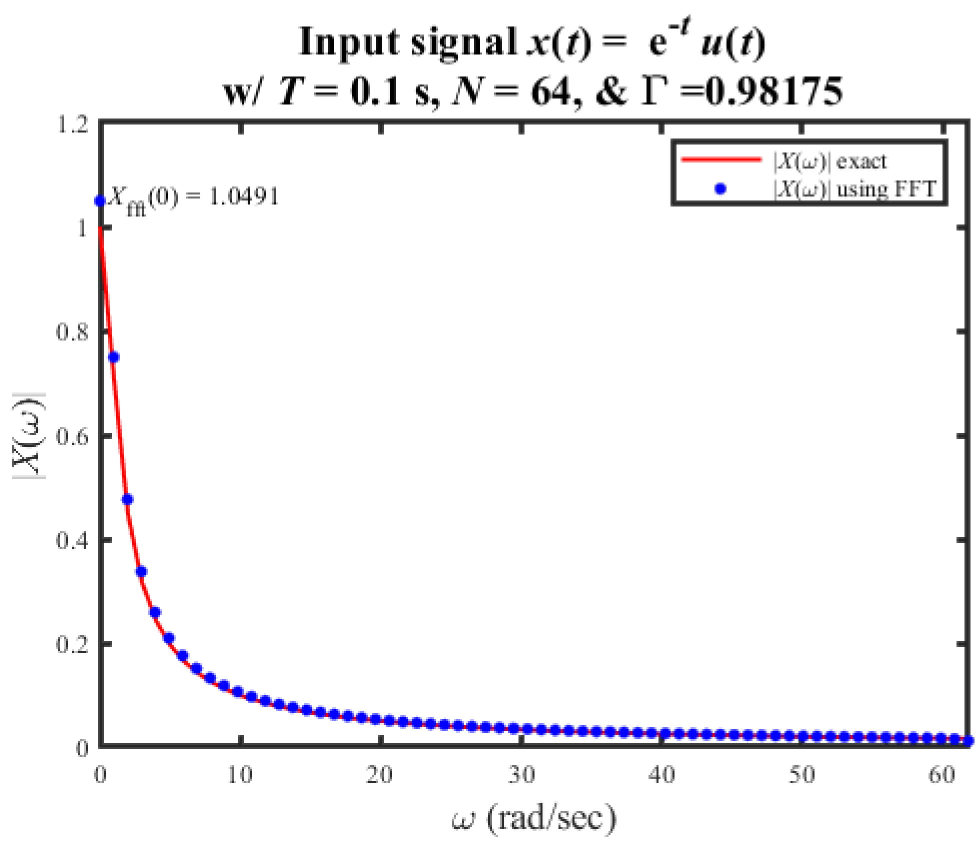
- **Try bigger sampling rate *T* with same number of points (worse).**





- **Try same sampling rate T with more points (nearly the same, except at DC).**





- **Try smaller sampling rate T AND more points (better at DC).**