Jason Waseq
ECE 103L
Assignment 4

1. During lab 4, we have seen numerical implementation of Fourier Series for periodic signals. As first part of this assignment, you need to write a Matlab function that would take an array representing a single period of a signal (x), corresponding time array (t), and return the Fourier Series coefficients (Ck) in exponential form. The function should also be able to take two (2) optional input arguments: number of Fourier coefficients (Nk) and plot option (p). Use the template ‘fourier_series_exp.m’ for this problem.

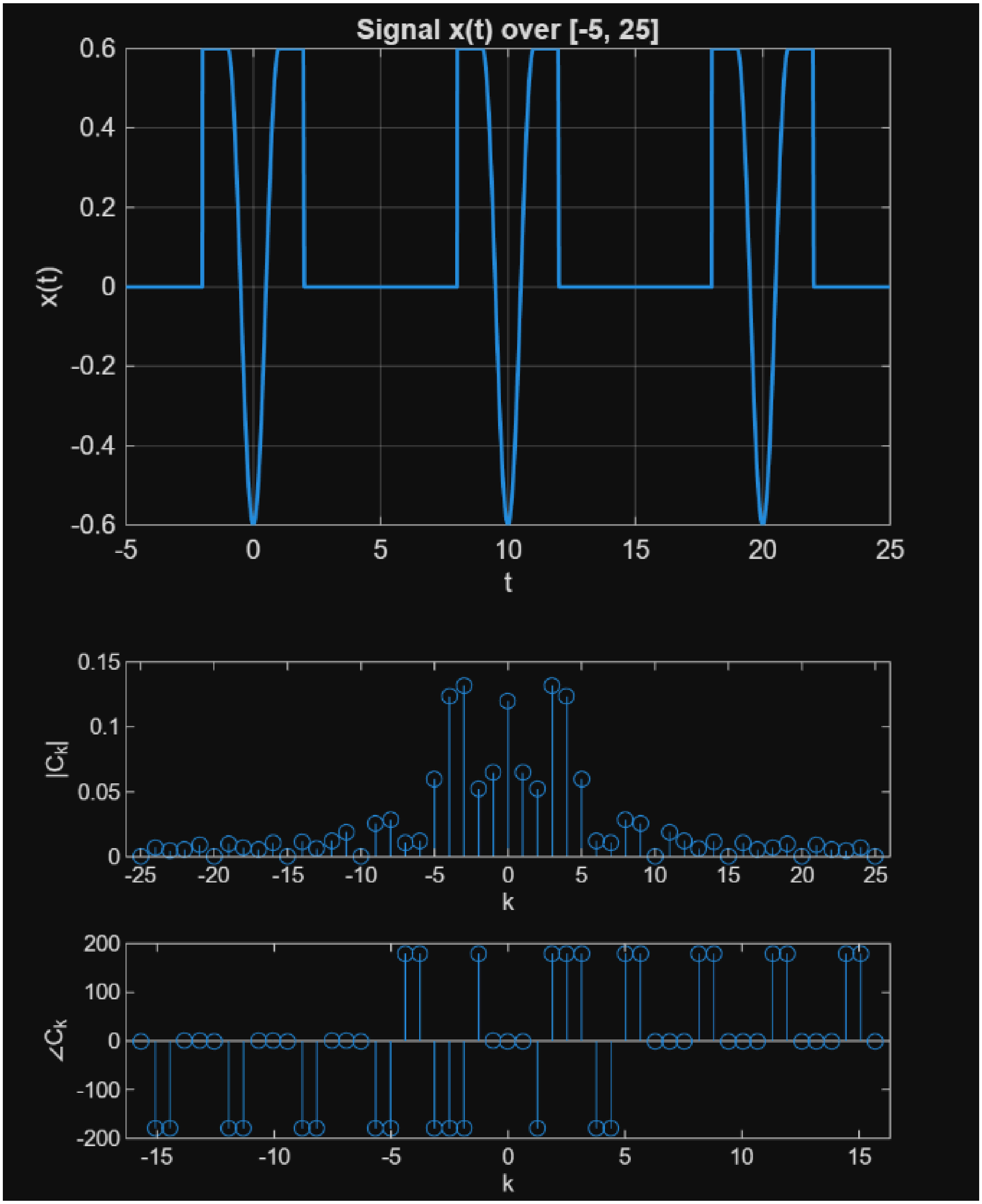
MATLAB Code:

```
function [Ck]=fourier_series_exp(x,t,Nk,p)

% Ck = exponential fourier series cofficient
% x = single period of a signal
% t = time corrosponding to 'x'
% Nk = (optional input) number of exponential terms
% p = plotting option ; p=0, no plots, p = 1 plot Ck vs k and reconstructed signal
% dT = t(2)-t(1) = temporal resolution of signal (x)
% T = peiod of signal 'x'
% w0= angular frequency of signal 'x'


  dT=t(2)-t(1);
  T= dT*length(t);
  w0=2*pi/T;

  % Check the number of inputs, 'nargin' returns number of input arguments
  if nargin <2
     error('Not enough input argument!')
  elseif nargin == 2
     Nk=101;  % you can set any default value you like
     p=0;  % not plots
  elseif nargin ==3
     p=0; % not plots
  end
  k=-floor(Nk/2):floor(Nk/2); % if Nk=11, k=-5:5; if Nk=12, k=-6:6


  %% evaluate Ck
  for ii=1:length(k)
     Ck(ii)=(1/T)*trapz(t, x.*exp(-j*k(ii)*w0*t));
  end


%% plot spectrum and reconstructed signal
```

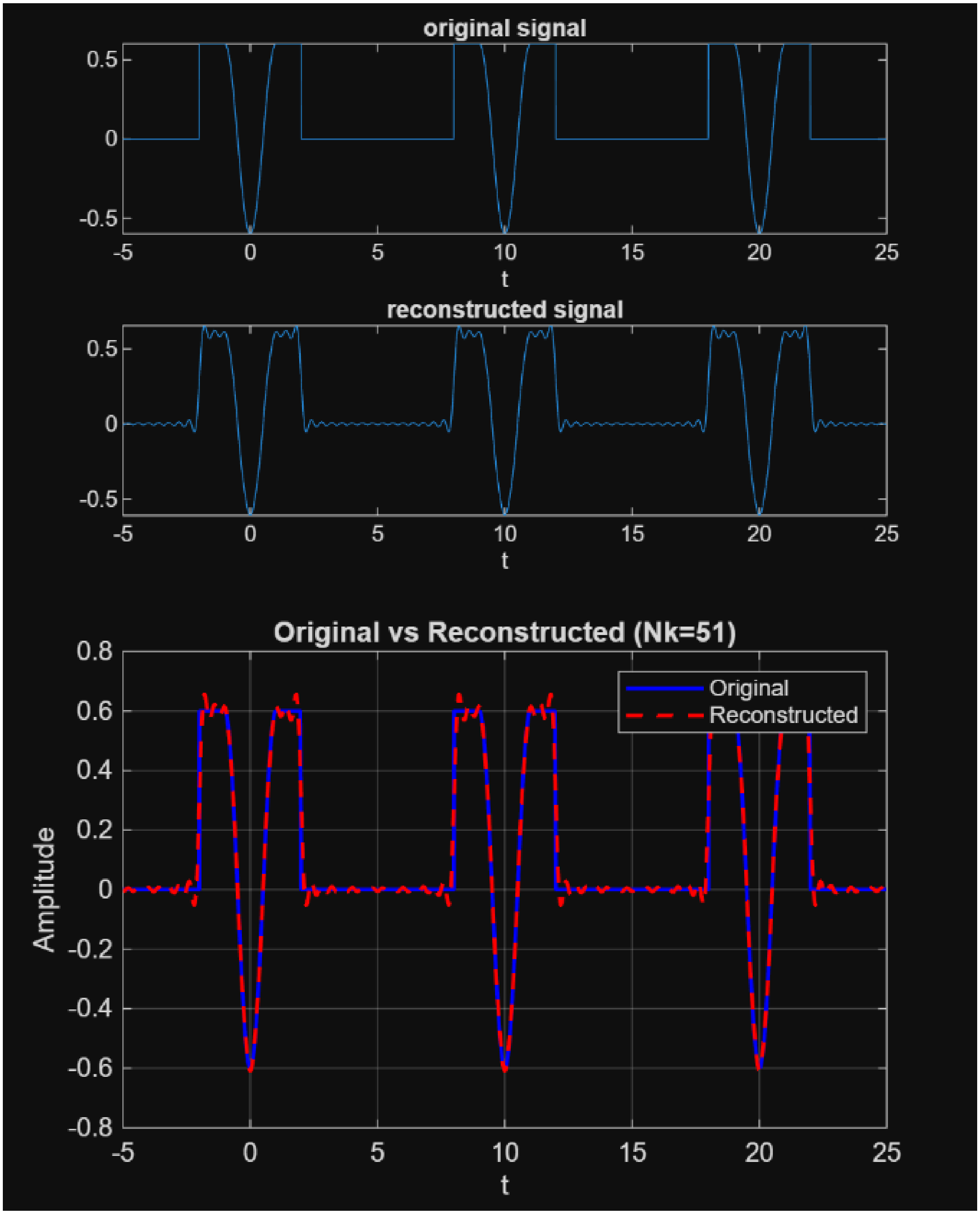
```
  if p==1
     % plot abs(Ck) vs k and angle(Ck) vs k
     figure
     subplot(211)
     stem(k,abs(Ck))
     xlabel('k')
     ylabel('|C_k|')

     w0k = w0*k;
     subplot(2,1,2)
     stem(w0k,angle(Ck)*180/pi);
     xlabel('k')
     ylabel('\angleC_k')

     % plot 3 cycles of the signal 'x' and the reconstructed signal
     x_ext=repmat(x,1,3);
     t_ext=t(1):dT:t(1)+(3*length(t)-1)*dT;

     x_reconstructed=zeros(size(t_ext));
     for ii=1:length(k)
        x_reconstructed=x_reconstructed+Ck(ii)*exp(j*k(ii)*w0*t_ext);
     end
     figure
     subplot(211)
     plot(t_ext,x_ext)
     xlabel('t');
     title('original signal')
     subplot(2,1,2)
     plot(t_ext,x_reconstructed)
     xlabel('t');
     title('reconstructed signal')
  end

end
```

2. A signal $x = 0.6\,\{u(t + 2) - (\cos(\pi t) + 1)\,[u(t + 1) - u(t - 1)] - u(t - 2)\}$ with a period $-5 \le t \le 5$ controls the location of the light source in an optical scanner. Plot the signal for the interval $-5 \le t \le 25$, its spectrum (C_k vs ω and $\angle C_k$ vs ω), and reconstructed time domain signal using 51 Fourier Series coefficients. Use the function you have written in problem 1 for solving this problem.

Final answer:



Signal x(t) over [-5, 25]
x(t)
t
|C_k|
k
∠C_k
k



The Fourier coefficients Ck describe the spectral content of x(t) with strong low-frequency terms due to the step components and weaker higher harmonics from the cosine segment. The reconstructed signal with Nk=51matched the original almost perfectly except for slight edge artifacts from periodic extension. Plots included x(t) over three periods, ∣Ck∣ vs ω, ∠Ck vs ω, and original vs reconstructed x(t).

Detailed solution:

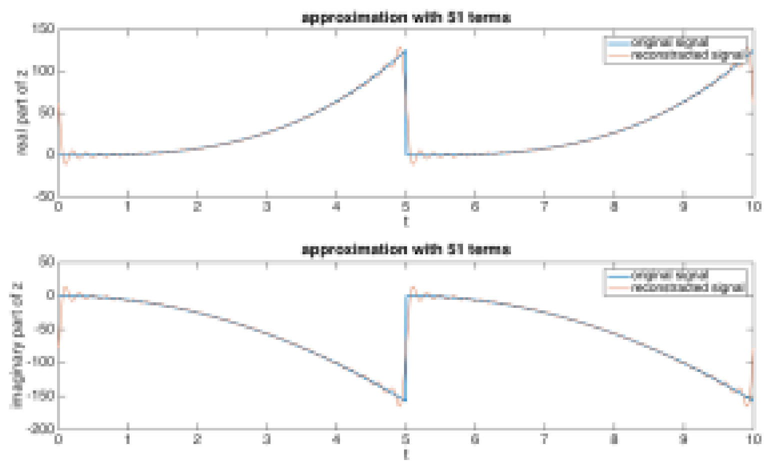
We expressed x(t) as a piecewise function over one period [−5,5] by evaluating each unit step term and cosine segment separately. The period was found to be T=10, so the fundamental frequency is ω0=2π/10. Using the provided fourier_series_exp function, we computed 51 exponential Fourier coefficients Ck via numerical integration over one period, then reconstructed the signal over three periods (−5 to 25) from those coefficients. The magnitude and phase spectra were plotted from Ck, and the reconstructed signal was compared to the original.

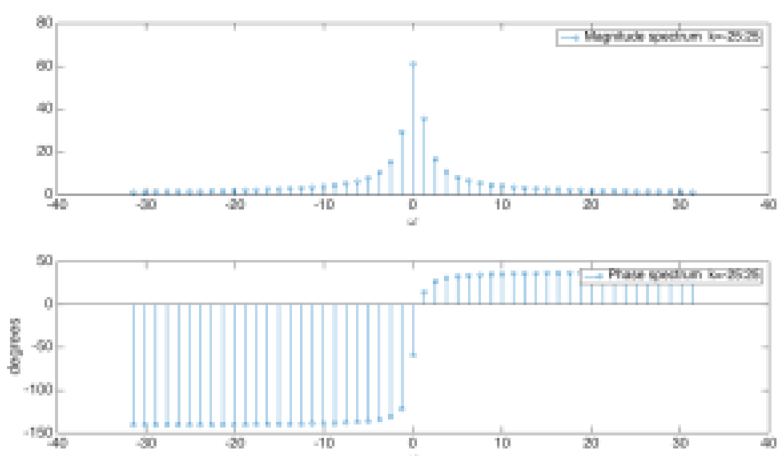
MATLAB Code:

```
% Define time for one period
T = 10;                  % Period
Fs = 1000;               % Sampling frequency (points per second)
dt = 1/Fs;
t_period = -5:dt:5-dt;   % One period [-5,5)

% Define the signal for one period
x_period = 0.6 * ( ...
  heaviside(t_period + 2) ...
  - (cos(pi*t_period) + 1) .* (heaviside(t_period + 1) - heaviside(t_period - 1)) ...
  - heaviside(t_period - 2) ...
);

% Plot signal from -5 to 25 (3 periods)
x_extended = repmat(x_period, 1, 3);   % Repeat 3 times
t_extended = -5:dt:25-dt;

figure;
plot(t_extended, x_extended, 'LineWidth', 1.5);
xlabel('t');
ylabel('x(t)');
title('Signal x(t) over [-5, 25]');
grid on;

% Compute Fourier Series coefficients & plot spectrum + reconstruction
Nk = 51;  % Number of coefficients
Ck = fourier_series_exp(x_period, t_period, Nk, 1);

% OPTIONAL: Reconstructed signal using Nk coefficients
w0 = 2*pi/T;
k = -floor(Nk/2):floor(Nk/2);
x_rec = zeros(size(t_extended));
for ii = 1:length(k)
  x_rec = x_rec + Ck(ii)*exp(1j*k(ii)*w0*t_extended);
end

figure;
plot(t_extended, real(x_extended), 'b', 'LineWidth', 1.5); hold on;
plot(t_extended, real(x_rec), 'r--', 'LineWidth', 1.5);
xlabel('t');
```

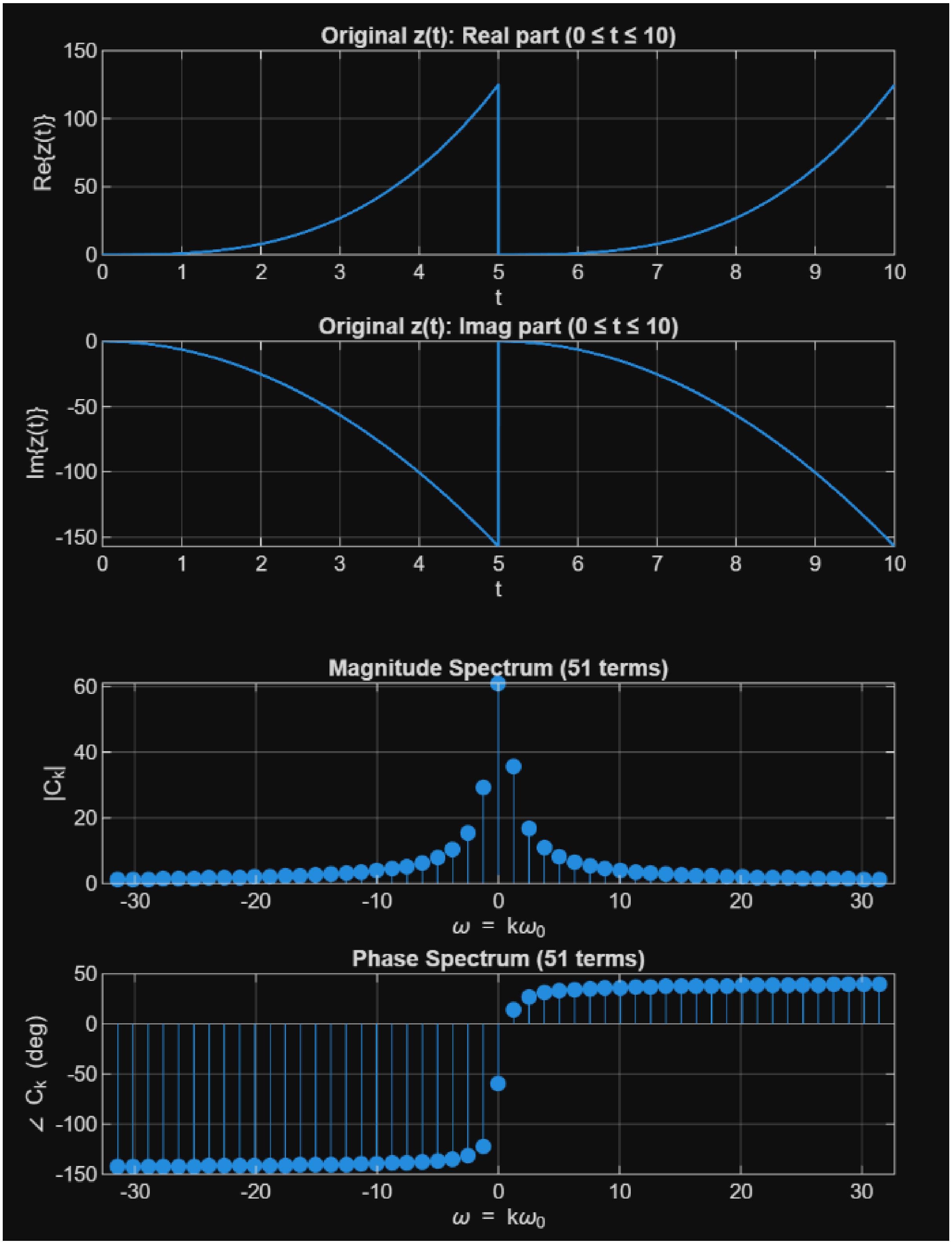
```
ylabel('Amplitude');
legend('Original','Reconstructed');
title('Original vs Reconstructed (Nk=51)');
grid on;
```

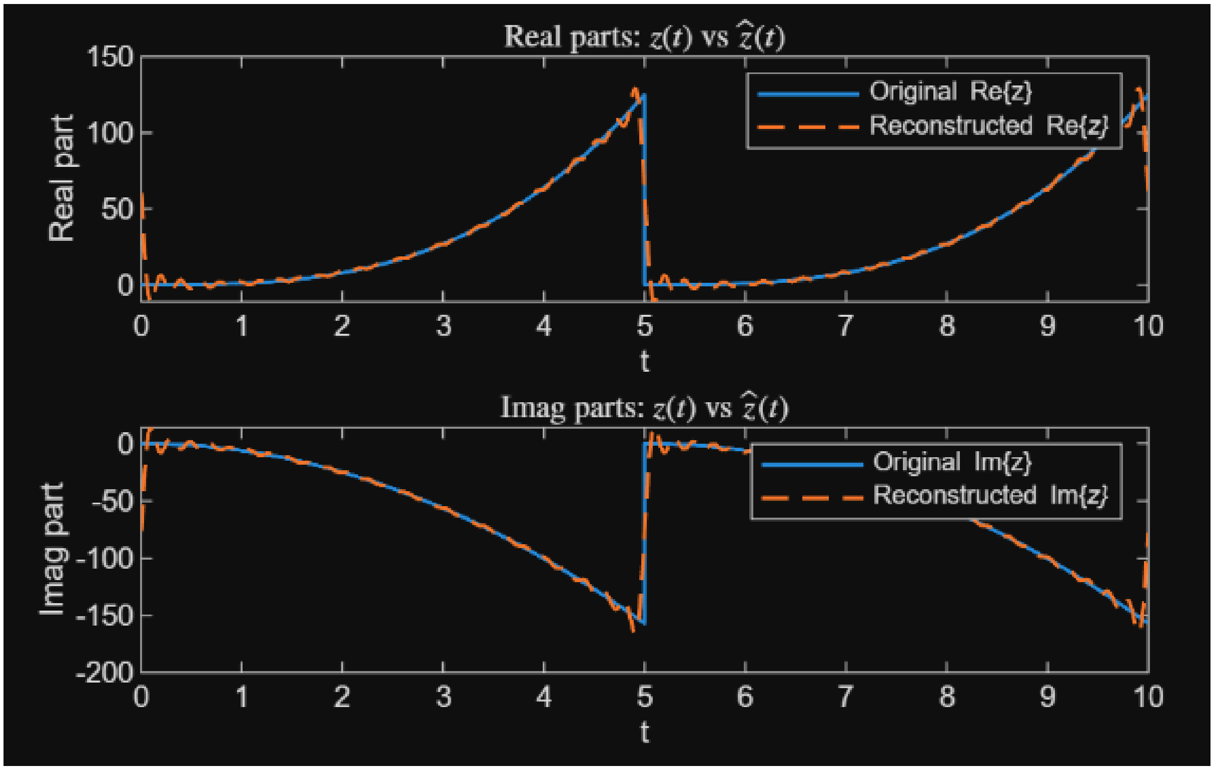
3. So far all the signals we have handled in this course are real signals. However, we can also use complex numbers to represent signals (complex signals). Let's consider a single period of a periodic signal $z(t) = t^3 - j2\pi t^2$, $0 < t \leq 5$. Calculate 51 Fourier Series coefficients (C_k) for this signal and reconstruct the time domain signal$\hat{z}(t)$ using these Fourier Series coefficients. Plot the spectrum (C_k vs ω and $\angle C_k$ vs ω) and the real and imaginary part of $z(t)$ and $\hat{z}(t)$for an interval of $0 \leq t \leq 10$. You can modify the Matlab file 'fs_numerical.m' which was used during the lab for solving this problem. Following are the sample plots for your reference.





Final answer:


Original z(t): Real part (0 ≤ t ≤ 10)
Re{z(t)}
t
Original z(t): Imag part (0 ≤ t ≤ 10)
Im{z(t)}
t
Magnitude Spectrum (51 terms)
|Ck|
ω = kω0
Phase Spectrum (51 terms)
∠ Ck (deg)
ω = kω0



The spectrum ∣Ck∣ showed decreasing magnitude with increasing ∣k∣, consistent with the smoothness of z(t), and the phase ∠Ck reflected the relative shifts between real and imaginary parts. The reconstructed z^(t) matched the original z(t) extremely closely for both real and imaginary components, confirming that 51 coefficients are sufficient to represent this smooth periodic complex signal.

Detailed solution:
We defined a single period of z(t)over 0 < t ≤ 5 with period T=5 and ω0=2π/5. The exponential Fourier series coefficients Ck for k=−25,…,25 were calculated numerically; for k=0, an exact closed form was derived to verify correctness. We then reconstructed the periodic signal z^(t) from these coefficients and plotted magnitude and phase spectra along with the real and imaginary parts of both the original and reconstructed signals over two periods (0 to 10).

MATLAB Code:

```
% z(t) = t^3 - j*2*pi*t^2 on 0 < t <= 5, periodic with T=5
% Compute 51 FS coefficients and reconstruct over 0 <= t <= 10

close all; clear; clc;

%% Single-period definition
T = 5;
w0 = 2*pi/T;

% Time grid for one period (exclude t=0 endpoint duplication at wrap)
dt = 1e-3;                % sufficiently fine for accurate trapz
t_single = 0:dt:T-dt;     % [0, T)
z_single = (t_single.^3) - 1j*2*pi*(t_single.^2);
```

```
%% Replicate to visualize original over 0..10 (two periods)
numPeriods = 2;
z_extended = repmat(z_single, 1, numPeriods);
t_extended = linspace(t_single(1), ...
  t_single(1) + (length(z_extended)-1)*dt, length(z_extended));

figure(1); clf;
subplot(2,1,1);
plot(t_extended, real(z_extended), 'LineWidth', 1.2);
xlabel('t'); ylabel('Re\{z(t)\}');
title('Original z(t): Real part (0 ≤ t ≤ 10)'); grid on; xlim([0 10]);

subplot(2,1,2);
plot(t_extended, imag(z_extended), 'LineWidth', 1.2);
xlabel('t'); ylabel('Im\{z(t)\}');
title('Original z(t): Imag part (0 ≤ t ≤ 10)'); grid on; xlim([0 10]);

%% Exponential FS: 51 terms (k=-25:25)
Nk = 51;
k = -floor(Nk/2):floor(Nk/2);

Ck = zeros(size(k));
for ii = 1:length(k)
  % Numerical integral over one period (trapz)
  Ck(ii) = (1/T) * trapz(t_single, z_single .* exp(-1j*k(ii)*w0*t_single));
end

% (Optional) exact DC term override for extra accuracy:
C0_exact = (T^3)/4 - 1j*(2*pi*T^2)/3;
Ck(k==0) = C0_exact;

%% Spectrum plots: |Ck| and angle(Ck) vs omega = k*w0
w = w0 * k;

figure(2); clf;
subplot(2,1,1);
stem(w, abs(Ck), 'filled'); grid on;
xlabel('\omega = k\omega_0'); ylabel('|C_k|');
title('Magnitude Spectrum (51 terms)');

subplot(2,1,2);
stem(w, angle(Ck)*180/pi, 'filled'); grid on;
xlabel('\omega = k\omega_0'); ylabel('\angle C_k (deg)');
title('Phase Spectrum (51 terms)');

%% Reconstruction on 0..10 with the same dt and number of samples
```

```
t_rec = t_extended;                   % match 0..10 timeline
z_rec = zeros(size(t_rec));
for ii = 1:length(k)
   z_rec = z_rec + Ck(ii) * exp(1j*k(ii)*w0*t_rec);
end

%% Plot reconstructed vs original (real and imag)
figure(3); clf;
subplot(2,1,1);
plot(t_rec, real(z_extended), 'LineWidth', 1.2); hold on;
plot(t_rec, real(z_rec), '--', 'LineWidth', 1.2);
xlabel('t'); ylabel('Real part');
legend('Original Re\{z\}','Reconstructed Re\{\itz\}');
title('Real parts: $z(t)$ vs $\hat{z}(t)$', 'Interpreter', 'latex');

subplot(2,1,2);
plot(t_rec, imag(z_extended), 'LineWidth', 1.2); hold on;
plot(t_rec, imag(z_rec), '--', 'LineWidth', 1.2);
xlabel('t'); ylabel('Imag part');
legend('Original Im\{z\}','Reconstructed Im\{\itz\}');
title('Imag parts: $z(t)$ vs $\hat{z}(t)$', 'Interpreter', 'latex');
```