1(a). The following graphs were obtained by running Program P2_2 with the three specified frequencies:

\[ f = 0.05 \]
$f = 0.10$

Input Signal

Output signal

$f = 0.25$

Input Signal

Output signal
1(b). The output signals depend on the frequencies of the input signal according to:
\[ y[n] = \sin^2(\omega) = \sin^2(2\pi f). \]

**Mathematical Verification:**

Let \( \omega = 2\pi f \) and let the input signal be \( x[n] = \cos(\omega n) \). Then (2.14) on page 21 of the course Laboratory Manual becomes

\[ y[n] = x[n]^2 - x[n-1]x[n+1] \]
\[ = \cos^2(\omega n) - \cos[\omega(n-1)]\cos[\omega(n+1)]. \]  

(1)

We have \( \cos[\omega(n-1)] = \cos(\omega n - \omega) \) and \( \cos[\omega(n+1)] = \cos(\omega n + \omega) \), so

\[ x[n-1]x[n+1] = \cos(\omega n - \omega)\cos(\omega n + \omega). \]  

(2)

Applying the trigonometric identity \( \cos(A + B)\cos(A - B) = \frac{1}{2}\cos(2A) + \frac{1}{2}\cos(2B) \) to (2), we obtain

\[ x[n-1]x[n+1] = \frac{1}{2}\cos(2\omega n) + \frac{1}{2}\cos(2\omega). \]  

(3)

Upon applying the trigonometric identity \( \cos(2A) = 2\cos^2(A) - 1 \) to the first term on the right-hand side of (3), there results

\[ x[n-1]x[n+1] = \cos^2(\omega n) - \frac{1}{2} + \frac{1}{2}\cos(2\omega). \]  

(4)

Applying the trigonometric identity \( \cos(2A) = 1 - 2\sin^2(A) \) to the second term on the right-hand side of (4) then gives us

\[ x[n-1]x[n+1] = \cos^2(\omega n) - \frac{1}{2} + \frac{1}{2} - \sin^2(\omega) \]
\[ = \cos^2(\omega n) - \sin^2(\omega). \]  

(5)

Substituting (5) into (1), we then obtain

\[ y[n] = \cos^2(\omega n) - \cos^2(\omega n) + \sin^2(\omega) = \sin^2(\omega), \]

(6)

which verifies the desired result.
% Program Q2_6
% Modified version of P2_2 to make the input a cosine plus a constant.
clf;
N = 0:200;
K = 0.5;
f = 0.1;
arg = 2*pi*f*n;
x = cos(arg) + K;
% Compute the output signal
x1 = [x 0 0];  % x1[n] = x[n+1]
x2 = [0 x 0];  % x2[n] = x[n]
x3 = [0 0 x];  % x3[n] = x[n-1]
y = x2.*x2-x1.*x3;
y = y(2:202);
% Plot the input and output signals
subplot(2,1,1)
plot(n, x)
xlabel('Time index n');ylabel('Amplitude');
title('Input Signal')
subplot(2,1,2)
plot(n,y)
xlabel('Time index n');ylabel('Amplitude');
title('Output signal');
Graphs obtained by running the modified program:

\[
f = 0.1, \ K = 0.5
\]

2(b). From Eq. (3) on page 366 of the paper by Restrepo and Quiroga, we obtained

\[
y[n] = TK[\cos(\omega_0 n) + K] = \sin^2(\omega_0) + 2K[1 - \cos(\omega_0)]\cos(\omega_0 n). \tag{7}
\]

I will select choice (ii) and verify the correctness of (7) by starting with Eq. (2.14) on page 21 of the course Laboratory Manual.

Let \( x[n] = \cos(\omega_0 n) + K \). Eq. (2.14) of the course Laboratory Manual says that

\[
y[n] = x^2[n] - x[n - 1]x[n + 1]. \tag{8}
\]

We have that

\[
x^2[n] = \cos^2(\omega_0 n) + 2K \cos(\omega_0 n) + K^2. \tag{9}
\]

Working on the second term on the right-hand side of (8), we obtain

\[
x[n - 1] = \cos[\omega_0(n - 1)] + K = \cos(\omega_0 n - \omega_0) + K
\]
and
\[ x[n + 1] = \cos[\omega_0(n + 1)] + K = \cos(\omega_0 n + \omega_0) + K, \]
so
\[ x[n - 1]x[n + 1] = \cos(\omega_0 n - \omega_0) \cos(\omega_0 n + \omega_0) + K[\cos(\omega_0 n - \omega_0) + \cos(\omega_0 n + \omega_0)] + K^2. \quad (10) \]

Now, it was already shown in (2)-(5) above that
\[ \cos(\omega_0 n - \omega_0) \cos(\omega_0 n + \omega_0) = \cos^2(\omega_0 n) - \sin^2(\omega_0). \quad (11) \]
Substituting this into (10) and applying the trigonometric identity \( \cos(A + B) + \cos(A - B) = 2 \cos(A) \cos(B) \) to the term in square brackets, we get
\[ x[n - 1]x[n + 1] = \cos^2(\omega_0 n) - \sin^2(\omega_0) + 2K \cos(\omega_0 n) \cos(\omega_0) + K^2. \quad (12) \]
Substituting (9) and (12) into (8) then produces the final result
\[ y[n] = \sin^2(\omega_0) + 2K \cos(\omega_0 n) - 2K \cos(\omega_0) \cos(\omega_0) \]
\[ = \sin^2(\omega_0) + 2K[1 - \cos(\omega_0)] \cos(\omega_0), \quad (13) \]
thereby verifying the correctness of (7).

Equations (7) and (13) provide the answer to how the output signal \( y[n] \) depends on the DC value \( K \). Compared to the case where \( x[n] = \cos(\omega_0 n) \), there is an extra term added to \( y[n] \). This extra term is a cosine having the same frequency as the input and having an amplitude of \( 2K[1 - \cos(\omega_0)] \). This term adds a ripple to \( y[n] \) which may be plainly seen in the graph on the preceding page. The amplitude of the ripple is proportional to \( K \).
3(a). I selected the paper:


Another good one is:


3(b). Listing of Matlab program to implement DESA-1:

```matlab
% Program DESA1
% Implement the DESA-1 discrete energy separation algorithm
clear;
clf;
% Generate the input signal
n = 0:250;
A = 5.0;
f = 0.1;
w = 2*pi*f;
x = A*cos(w*n);

x2 = [0 x 0];                   % zero pad x[n]
y = [0 x 0] - [0 0 x];         % y[n] = x[n] - x[n-1]
z = [x 0 0] - [0 x 0];         % z[n] = x[n+1] - x[n]

% TK operator on y[n]
ynp1 = [y 0 0];       % y[n+1]
y  = [0 y 0];       % y[n]
ynm1 = [0 0 y];     % y[n-1]
Psi_y = y.*yn-ynm1.*ynp1;

% TK operator on z[n]
zn = [0 z 0];       % z[n]
znm1 = [0 0 z];     % z[n-1]
Psi_z = zn.*zn-znm1.*znp1;

% TK operator on x[n]
x2np1 = [x2 0 0];    % x2[n+1]
```

7
x2n = [0 x2 0]; % x2[n]
x2nm1 = [0 0 x2]; % x2[n-1]
Psi_x2 = x2n.*x2n-x2nm1.*x2np1;

% Resize TK outputs to original size of x[n]
Psi_y = Psi_y(3:length(Psi_y)-2);
Psi_z = Psi_z(3:length(Psi_z)-2);
Psi_x2 = Psi_x2(3:length(Psi_x2)-2);

% Estimate the IA and IF
IFhat = acos(1 - (Psi_y + Psi_z)./(4*Psi_x2))/(2*pi);
IAhat = sqrt(Psi_x2./(1 - (1 - (Psi_y + Psi_z)./(4*Psi_x2)).^2));

% Plot the input and output signals
figure(1);
plot(n,x);
axis([0 250 -6 6]);
xlabel('Time index n');ylabel('Amplitude');
title('Input Signal');

figure(2);
plot(n,IAhat);
xlabel('Time index n');ylabel('IA a[n]');
title('Instantaneous Amplitude (IA)');

figure(3);
plot(n,IFhat);
xlabel('Time index n');ylabel('IF \omega[n] / (2\pi)');
title('Instantaneous Frequency (IF)');
Plots obtained by running the program with the input signal $x[n] = 5 \cos[2\pi(0.1)n]$: 
3(c). Plots obtained by running the program on an AM-FM chirp signal: