Національний технічний університет України  
«Київський політехнічний інститут ім. І.Сікорського»  
Факультет електроніки  
Кафедра фізичної та біомедичної електроніки

Лабораторна робота №2

«**Вивчення та реалізація реконструкції томографічних зображень на основі перетворення Радона**»  
з курсу «Біомедичні електронні системи»

Підготувала  
ст. 5 курсу гр. ДМ-61м  
Нікітюк Н.

Київ  
2017

**Мета роботи**: ознайомлення зі застосуванням прямого та оберненого перетворення Радона в комп’ютерних томографах. Дослідження та набуття навичок обробки зображень з використанням перетворення Радона в середовищі MatLAB

**Робоче завдання та лістинг програми**

1. Дослідження перетворень Радона на моделі томографічного зрізу головного мозку.

1.1. Виконати пряме перетворення Радона моделі томографічного зрізу (рис. 5.3), вибравши кути проекцій від 0 до180 градусів.

P = phantom(200);

figure(1);

imshow(P);title('Модель томографічного зрізу')

figure(2)

G = flipud(gray); %перевернути матрицю палітри зверху донизу

imagesc(P);colormap(G);colorbar; axis image; %образ матриці Р

title('Модель томографічного зрізу')

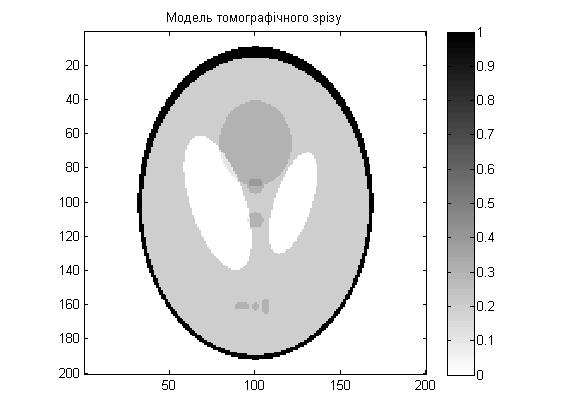
[R, xp] = radon(P, 0:180);

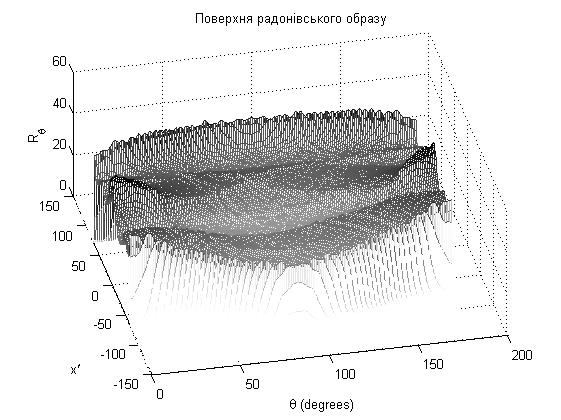
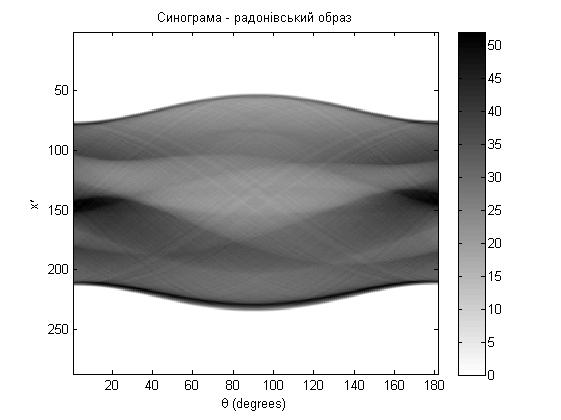
figure(3)

imagesc(R);colormap(G);colorbar;title('Синограма - радонівський образ'); xlabel('\theta (degrees)'); ylabel('x\prime')

figure(4); mesh(0:180, xp, R);colormap(G);

title('Поверхня радонівського образу'); xlabel('\theta (degrees)'); ylabel('x\prime'); zlabel('R\_\theta')



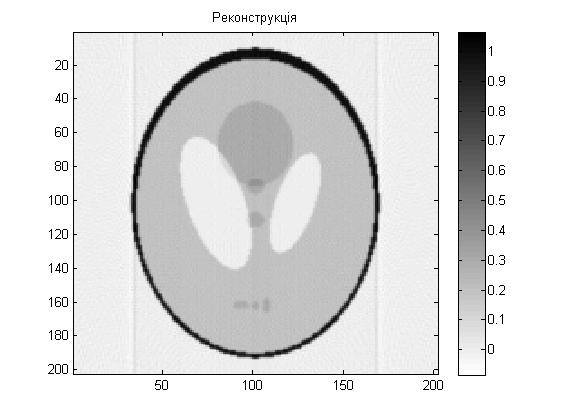
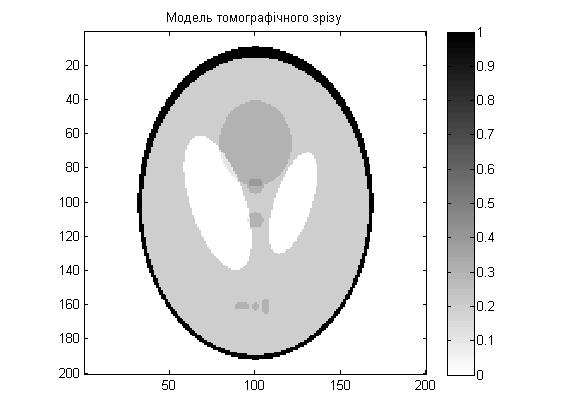


1.2. Відновити зображення за допомогою оберненого перетворення Радона за всіма значеннями проекцій. Порівняти отримане зображення з початковим. Зробити висновки.

I = iradon(R, 0:180);

figure(5)

imagesc(I);colormap(G);colorbar;title('Реконструкція');

  
1.3. Побудувати залежність часу відновлення зображення від кількості використаних проекцій. Зробити висновки.

I = zeros(size(I));

step = [1 3 5 12 18 20 30 36 45 60 90 180];

t = zeros(1, length(step));

for n = 1:length(step)

tic;

I = iradon(R, step(n));

proj(n) = 180./step(n) + 1;

t(n) = toc;

end

proj\_new = proj(1):-1:proj(end);

t\_new = interp1(proj, t, proj\_new, 'pchip');

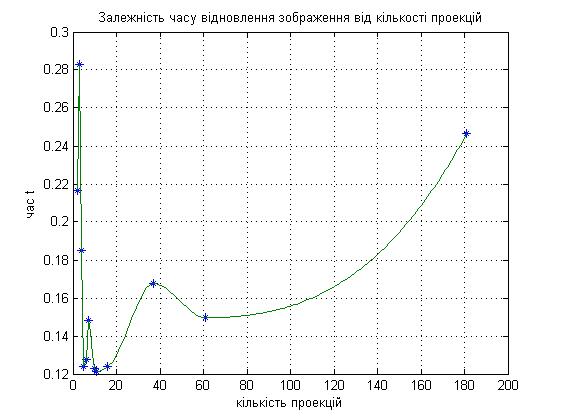
figure(6)

plot(proj, t, '\*', proj\_new, t\_new, '-')

title('Залежність часу відновлення зображення від кількості проекцій');

xlabel('кількість проекцій'); ylabel('час t')

grid on



1.4. Дослідити вплив параметрів оберненого перетворення Радона на схожість відновленого зображення з оригіналом для п'яти комбінацій виду інтерполяції і типу фільтра. Для цього визначити середньоквадратичну похибку, та побудувати графік її залежності від кількості проекцій, що використовуються. Зробити висновки

interp = {'nearest', 'spline', 'nearest', 'spline', 'nearest'};

filter = {'Shepp-Logan', 'Cosine', 'Hamming', 'Hann', 'Hann'};

for j = 1:length(interp)

for n = 1:length(step)

I = zeros(size(R));

I = iradon(R, step(n), interp{j}, filter{j});

proj(n) = 180./step(n) + 1;

deviation(n) = sqrt(sum(sum((I(1:200,1:200) - P).^2)));

end

figure()

proj\_new = proj(1):-1:proj(end);

deviation\_new = interp1(proj, deviation, proj\_new, 'pchip');

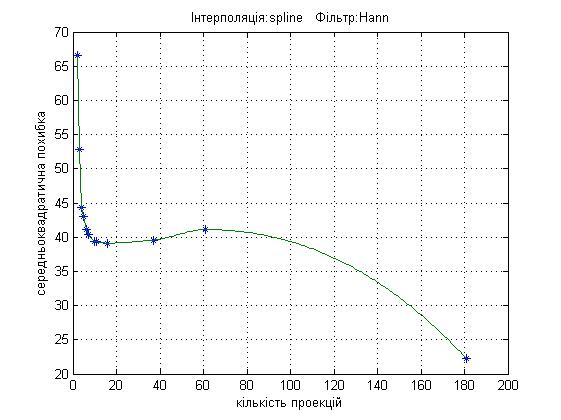
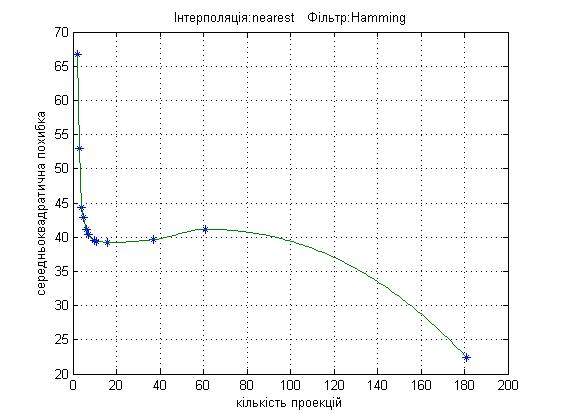
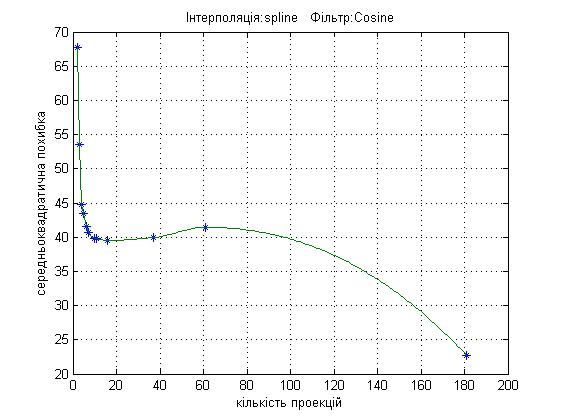
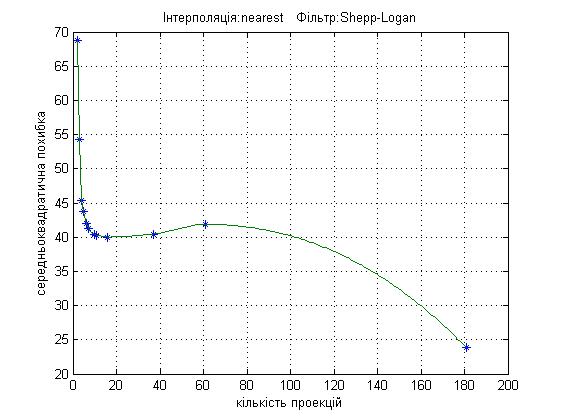
plot(proj, deviation, '\*', proj\_new, deviation\_new, '-')

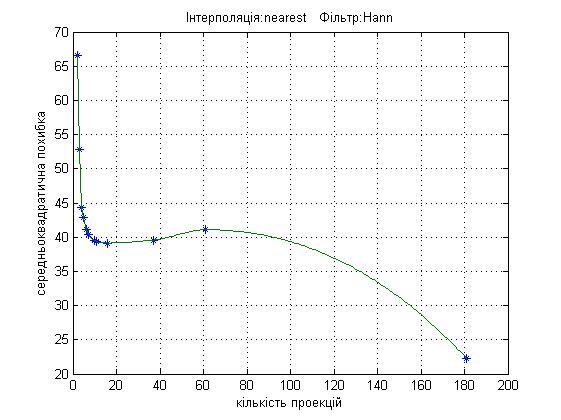
title(strcat({'Інтерполяція:'},{interp{j}}, {' Фільтр:'},{filter{j}}))

xlabel('кількість проекцій'); ylabel('середньоквадратична похибка')

grid on

end





2. Виконати пп. 1.1 – 1.4. для зображення реального зрізу головного мозку. Зробити висновки.

clc;clear all;close all;

% пункт 1

%%1.1

%P = phantom(200);

P = imread('image.jpg', 'jpg');

figure(1);

imshow(P);title('Зображення реального зрізу головного мозку'figure(2)

P = double(P);

G = flipud(gray); %перевернути матрицю палітри зверху донизу

imagesc(P);colormap(G);colorbar; axis image; %образ матриці Р

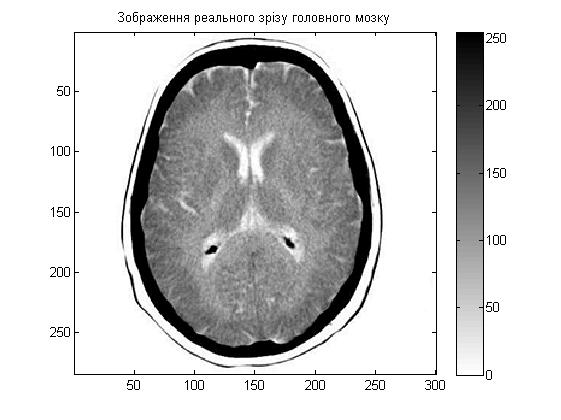
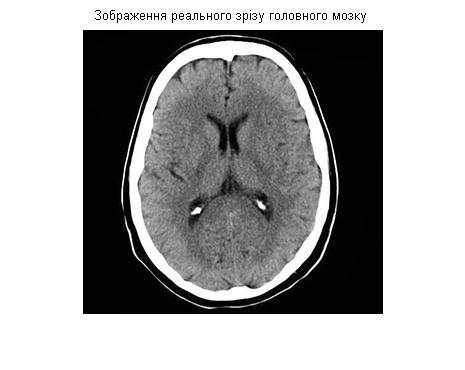
title('Зображення реального зрізу головного мозку' [R, xp] = radon(P, 0:180);

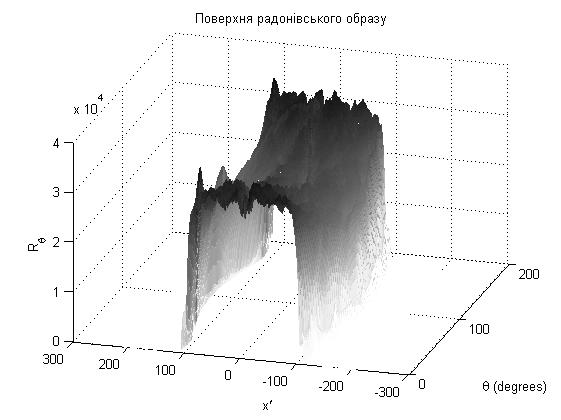
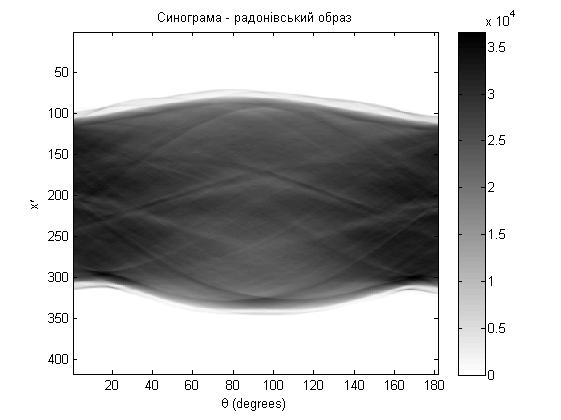
figure(3)

imagesc(R);colormap(G);colorbar;title('Синограма - радонівський образ'); xlabel('\theta (degrees)'); ylabel('x\prime')

figure(4); mesh(0:180, xp, R);colormap(G);

title('Поверхня радонівського образу'); xlabel('\theta (degrees)'); ylabel('x\prime'); zlabel('R\_\theta')



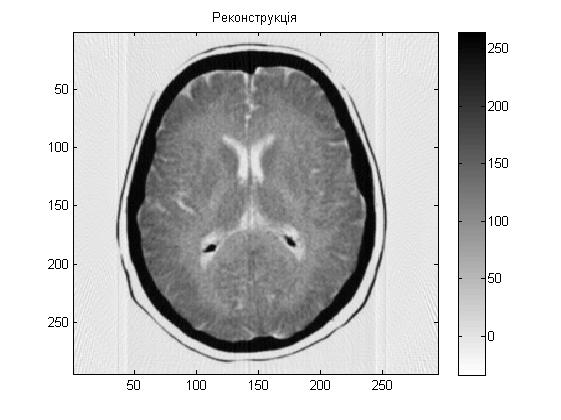
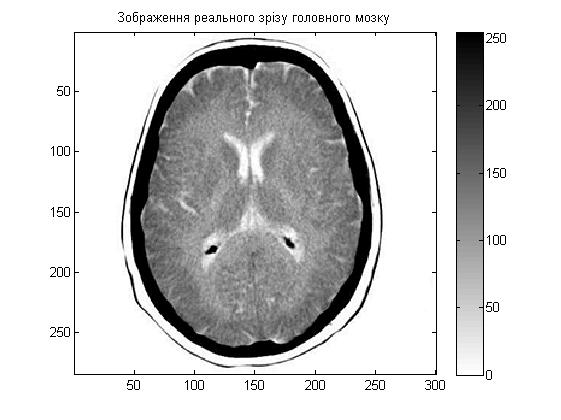


%%1.2

I = iradon(R, 0:180);

figure(5)

imagesc(I);colormap(G);colorbar;title('Реконструкція');

%%1.3

I = zeros(size(I));

step = [1 2 3 5 6 9 10 12 15 18 20 30 36 60 90 180];

t = zeros(1, length(step));

for n = 1:length(step)

tic;

I = iradon(R, step(n));

proj(n) = 180./step(n) + 1;

t(n) = toc;

end

proj\_new = proj(1):-1:proj(end);

t\_new = interp1(proj, t, proj\_new, 'pchip');

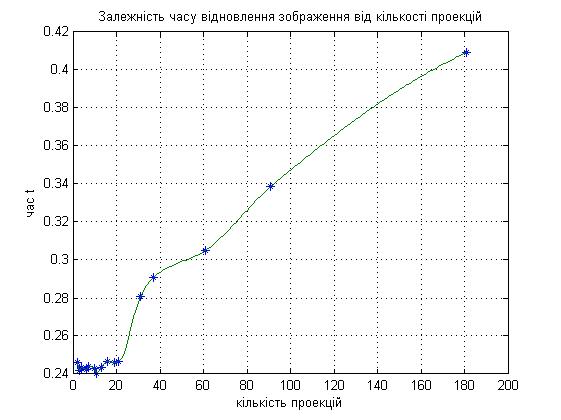
figure(6)

plot(proj, t, '\*', proj\_new, t\_new, '-')

title('Залежність часу відновлення зображення від кількості проекцій');

xlabel('кількість проекцій'); ylabel('час t')

grid on



%%1.4

step = [1 3 5 12 18 20 30 36 45 60 90 180];

interp = {'nearest', 'spline', 'nearest', 'spline', 'nearest'};

filter = {'Shepp-Logan', 'Cosine', 'Hamming', 'Hann', 'Hann'};

size\_P = size(P);

for j = 1:length(interp)

for n = 1:length(step)

I = zeros(size(R));

I = iradon(R, step(n), interp{j}, filter{j});

size\_I = size(I);

vectors = min([size\_P(1) size\_I(1)]);

columns = min([size\_P(2) size\_I(2)]);

proj(n) = 180./step(n) + 1;

deviation(n) = sqrt(sum(sum((I(1:vectors,1:columns) - P(1:vectors,1:columns)).^2)));

end

figure()

proj\_new = proj(1):-1:proj(end);

deviation\_new = interp1(proj, deviation, proj\_new, 'pchip');

plot(proj, deviation, '\*', proj\_new, deviation\_new, '-')

title(strcat({'Інтерполяція:'},{interp{j}}, {' Фільтр:'},{filter{j}}))

xlabel('кількість проекцій'); ylabel('середньоквадратична похибка')

grid on

end

