Problems Chapter 15

Problem 15.1

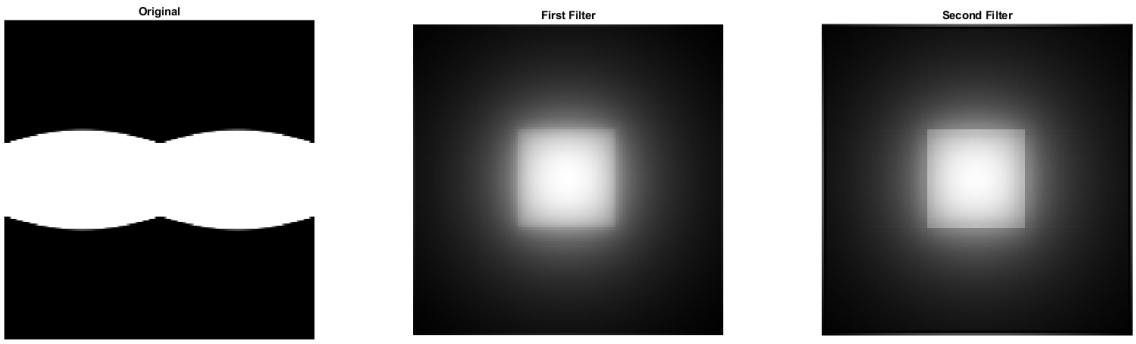
Load the proj_prof1 which contains a projection profile. Use an unfiltered inverse radon function. Deblur the image using 2 unsharp filters and display all 3 images.

Solution

```
p = load('C:\dev\biomedeng\Associated Files\Chapter 15\proj_prof1.mat')
p = p.p

[I_b,f] = iradon(p, 1, "none")% Inverse Radon Transform
I_b = mat2gray(I_b)

b = fspecial('unsharp')%Unsharp filter
I_f1 = imfilter(I_b, b)
I_f2 = imfilter(I_f1, b)%Apply twice

subplot(1, 3, 1)
imshow(p)
title('Original')

subplot(1, 3, 2)
imshow(I_f1)
title("First Filter")

subplot(1, 3, 3)
imshow(I_f2)
title("Second Filter")
```

Results

After two iterations of the unsharp filter the reconstructed image is still blurry but the square has well defined borders.



Problem 15.3

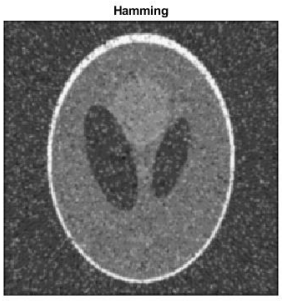
Load proj_prof3 which contains a projection profile. Reconstruct the image using four different filtered inverse radon functions: Ram-Lak, Hamming, Shepp-Logan and cosine filters. Note the differences and filter strengths.

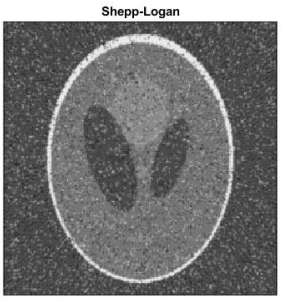
Solution

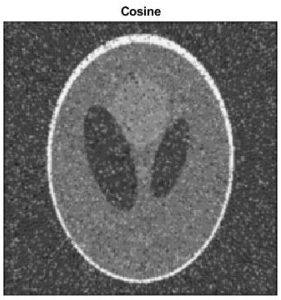
```
p = load('C:\dev\biomedeng\Associated Files\Chapter 15\proj_prof3.mat')
p = p.p

I_b1 = iradon(p, 1, "Ram-Lak")% Ram-Lak filter
I_b1 = mat2gray(I_b1)

I_b2 = iradon(p, 1, "Hamming")% Hamming filter
I_b2 = mat2gray(I_b2)

I_b3 = iradon(p, 1, "Shepp-Logan")% Shep-logan filter
I_b3 = mat2gray(I_b3)

I_b4 = iradon(p, 1, "Cosine")% Cosine filter
I_b4 = mat2gray(I_b4)

I_b5 = iradon(p, 1, "none")% defualt filter
I_b5 = mat2gray(I_b5)


subplot(2, 3, 1)
imshow(p)
title('Original')

subplot(2, 3, 2)
imshow(I_b1)
title('Ram-Lak')

subplot(2, 3, 3)
imshow(I_b2)
title('Hamming')

subplot(2, 3, 4)
imshow(I_b3)
title('Shepp-Logan')

subplot(2, 3, 5)
imshow(I_b4)
title('Cosine')

subplot(2, 3, 6)
imshow(I_b5)
title('none')
```

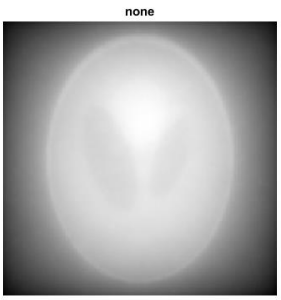
Results

The filters do not produce vastly different images and very little, if any, noise is filtered out. However, using no filter is very blurry and undefined compared to the filtered images.









Problem 15.8

Simulate an RF pulse and take its FFT and note its narrow bandwidth around the base frequency. Use both 64 and 32 MHz as base frequencies.

Solution

This uses a simple for loop to avoid writing everything twice, where f1 will be 64 and 32. The time vector goes from -0.5s to 0.5s.

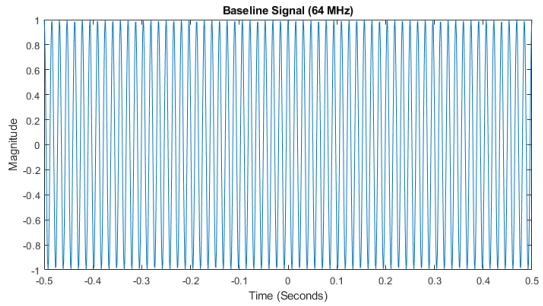
Code:

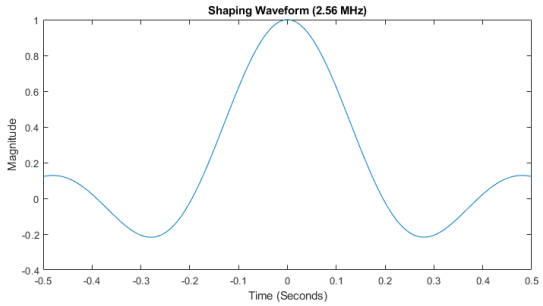
```
for f1 = [64 32] %MHz

    f2 = f1 / 25 %MHz
    fs = 1000 %MHz

    t = -0.5:1/fs:0.5 %Time vector, sinc is symmetric at t=0
    baseline = cos(2*pi*f1*t)

    shapewave = sinc(2*f2*t)

    shapedpusle = baseline .* shapewave
    sp_fft = abs(fft(shapedpusle))

    subplot(2, 2, 1)
    plot(t, baseline)
```

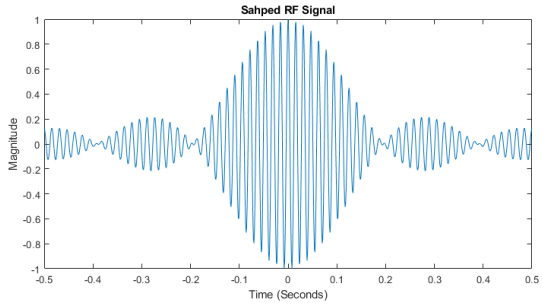
```
    xlabel('Time (Seconds)'); ylabel('Magnitude')
    title(['Baseline Signal (', num2str(f1), ' MHz)'])

    subplot(2, 2, 2)
    plot(t, shapewave)
    xlabel('Time (Seconds)'); ylabel('Magnitude')
    title(['Shaping Waveform (', num2str(f2), ' MHz)'])

    subplot(2, 2, 3)
    plot(t, shapedpusle)
    xlabel('Time (Seconds)'); ylabel('Magnitude')
    title('Sahped RF Signal')

    subplot(2, 2, 4)
    plot(sp_fft)
    xlim([0 150])
    xlabel('Frequency (MHz)'); ylabel('Magnitude')
    title('RF Signal Frequency Domain')
    figure;
end
```

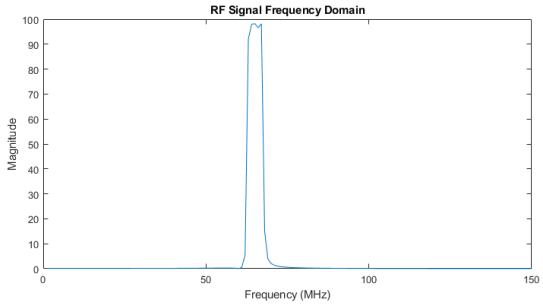
Results

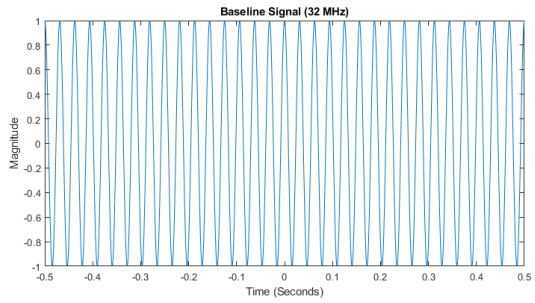
The Fourier transforms correctly shows a very narrow bandwidth around 64 and 32 MHz

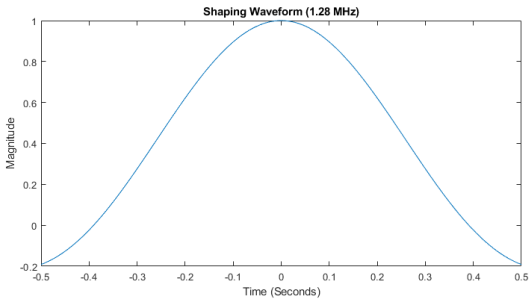


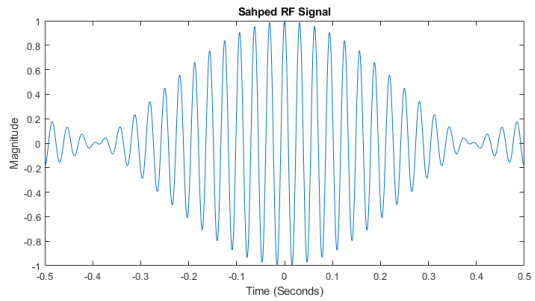


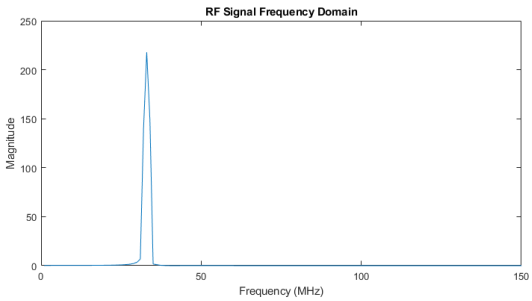












Problem Aliasing

Load frame 18 of mri.tif. Use a 2D FFT and plot it. Downsample the signal by zeroing every second row of the frequency matrix. Inverse it and plot it together with the original image. Note the difference and what happens.

Solution

The operation fft_2(2:2:end)=0 sets every column in every second row to zero.

Code:

```
I = imread('C:/dev/biomedeng/Associated Files/Chapter 15/mri.tif')
```

```
I = im2double(I)
```

```
fft = fft2(I)% Take 2D FFT
fft_p = fftshift(abs(fft))% Shift and abs

fft_2 = fft%Modified fft
fft_2(2:2:end) = 0 %Zero out every other row
fft_p2 = fftshift(abs(fft_2))% Shift and abs

I2 = ifft2(fft_2)%Inverse of the modified

subplot(2, 2, 1)
imshow(I)
title('Original')

subplot(2,2,2)
imshow(I2)
title('Reconstructed')

subplot(2,2,3)
mesh(fft_p)
```

```
title('2-D Forier Transform')

subplot(2,2,4)
mesh(fft_p2)
title('Modified 2D FFT')
```

Results

The image was downsampled and therefore we can now see the aliasing on the reconstructed image. 2 aliases can be seen, shifted halfway up and down the height of the image.