UCL Mechanical Engineering 2020/2021

ENGF0004 48-hour Project

NCWT3

April 28, 2021

Contents

List of Figures

1 PDEs, Matrix applications

1.1 Developing mathematical model

1.2 Constant cross-sectional area

1.3 Solving equation

1.4 Finite difference numerical scheme

1.5 Comment on implications of stiffening of blood vessels with age

1.6 Neglecting derivatives of non-linear terms of unknown variables

2 Vector calculus

2.1 Proof that divergence of velocity equals zero

Proof. If the fluid is incompressible, our total derivative is zero:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = 0 \tag{2.1}$$

$$\tag{2.2}$$

We can start to derive the divergence of the velocity by rewriting the second term in 2.3:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \underline{u}) = 0 \tag{2.3}$$

$$\frac{\partial \rho}{\partial t} + \rho \left(\nabla \cdot \underline{u}\right) + \underline{u} \cdot \left(\nabla \rho\right) = 0 \tag{2.4}$$

Looking at the $\nabla \rho$ term:

$$\nabla \rho = \left(\frac{\partial \rho}{\partial x}, \frac{\partial \rho}{\partial y}, \frac{\partial \rho}{\partial z} \right) \tag{2.5}$$

We know that all derivatives of ρ are zero as ρ is a constant, hence:

$$0 + \rho \left(\nabla \cdot \underline{u}\right) + 0 = 0 \tag{2.6}$$

$$\nabla \cdot \underline{u} = 0 \tag{2.7}$$

□

2.2 Acceleration of fluid element

Fluid element acceleration is given by:

$$\frac{\mathrm{D}u}{\mathrm{D}t} = \frac{\partial \underline{u}}{\partial t} + \left(\underline{u} \cdot \nabla\right) \underline{u} \tag{2.8}$$

Flow is steady, hence

$$\frac{\mathrm{D}u}{\mathrm{D}t} = 0 + (\underline{u} \cdot \nabla)\,\underline{u} \tag{2.9}$$

$$= -\omega y \frac{\partial \underline{u}}{\partial x} + -\omega x \frac{\partial \underline{u}}{\partial y} + 0 \frac{\partial \underline{u}}{\partial z} \tag{2.10}$$

$$= -\omega y \frac{\partial \underline{u}}{\partial x} + \omega x \frac{\partial \underline{u}}{\partial x} \tag{2.11}$$

$$= -\omega y \begin{pmatrix} 0 \\ \omega \\ 0 \end{pmatrix} + \omega x \begin{pmatrix} -\omega \\ 0 \\ 0 \end{pmatrix} \tag{2.12}$$

$$= \begin{pmatrix} -\omega^2 x \\ -\omega^2 y \\ 0 \end{pmatrix} \tag{2.13}$$

2.3 Integral

2.3.1 Area of integration

2.3.2 Find the limits of integration

2.3.3 Calculation of triple integral

$$= \iiint_V (xyz)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x \tag{2.14}$$

$$= \int_0^1 \int_0^{1-x} \int_0^{1-x-y} (xyz)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x \tag{2.15}$$

Computing the z integral:

$$= xy \int_0^{1-x-y} (z)\,\mathrm{d}z \tag{2.16}$$

$$= xy \left[\frac{z^2}{2}\right]_0^{1-x-y} \tag{2.17}$$

$$= xy \left[\frac{(1-x-y)^2}{2} - \frac{0^2}{2}\right] \tag{2.18}$$

$$= \frac{xy}{2}\left(y^2 + x^2 + 2xy - 2x - 2y + 1\right) \tag{2.19}$$

$$= \frac{1}{2}\left(xy^3 + x^3 y + 2x^2 y^2 - 2x^2 y - 2y^2 x + xy\right) \tag{2.20}$$

Inputting 2.20 into 2.15:

$$\int_0^1 \int_0^{1-x} \left(\frac{1}{2}\left(xy^3 + x^3 y + 2x^2 y^2 - 2x^2 y - 2xy^2 + xy\right)\right) \mathrm{d}y\,\mathrm{d}x \tag{2.21}$$

Computing the y integral:

$$= \frac{1}{2}\int_0^{1-x} \left(xy^3 + x^3y + 2x^2y^2 - 2x^2y - 2xy^2 + xy\right) \mathrm{d}y \tag{2.22}$$

$$= \frac{1}{2}\left[\frac{xy^4}{4} + \frac{x^3y^2}{2} + \frac{2x^2y^3}{3} - x^2y^2 - \frac{2xy^3}{3} + \frac{xy^2}{2}\right]_0^{1-x} \tag{2.23}$$

$$= \frac{1}{2}\left[\frac{x\left(1-x\right)^4}{4} + \frac{x^3\left(1-x\right)^2}{2} + \frac{2x^2\left(1-x\right)^3}{3} - x^2\left(1-x\right)^2 - \frac{2x\left(1-x\right)^3}{3} + \frac{x\left(1-x\right)^2}{2}\right] \tag{2.24}$$

Expanding:

$$= \frac{1}{2}\left[\frac{x - 4x^2 + 6x^3 - 4x^4 + x^5}{4} + \frac{x^3 - 2x^4 + x^5}{2} + \frac{2x^2 - 6x^3 + 6x^4 - 2x^5}{3}\right.$$
$$\left. - \left(x^2 - 2x^3 + x^4\right) - \frac{2x - 6x^2 + 6x^3 - 2x^4}{3} + \frac{x - 2x^2 + x^3}{2}\right] \tag{2.25}$$

Simplifying

$$= \frac{x^5 - 4x^4 + 6x^3 - 4x^2 + x}{24} \tag{2.26}$$

$$= \frac{1}{24}\left(x^5 - 4x^4 + 6x^3 - 4x^2 + x\right) \tag{2.27}$$

Inputting 2.27 into 2.21:

$$\int_0^1 \left(\frac{1}{24}\left(x^5 - 4x^4 + 6x^3 - 4x^2 + x\right)\right) \mathrm{d}x \tag{2.28}$$

Computing the x integral:

$$= \frac{1}{24}\left[\frac{x^6}{6} - \frac{4x^5}{5} + \frac{3x^4}{2} - \frac{4x^3}{3} + \frac{x^2}{2}\right]_0^1 \tag{2.29}$$

$$= \frac{1}{24}\left[\frac{1}{6} - \frac{4}{5} + \frac{3}{2} - \frac{4}{3} + \frac{1}{2}\right] \tag{2.30}$$

$$= \frac{1}{720} \tag{2.31}$$

3 Transforms

3.1 Plot of data

```
clc
clear
close all

%import data
data = readmatrix('Section3_data.txt');

%plot data
plot(data(:,1), data(:,2))
title('Graph to show variation in signal over a period of 100 seconds')
xlim([0 100])
```

```
ylim([-5 5])
xlabel('Time/s')
ylabel('Pulse oximeter signal/arbitrary units')
grid on
```

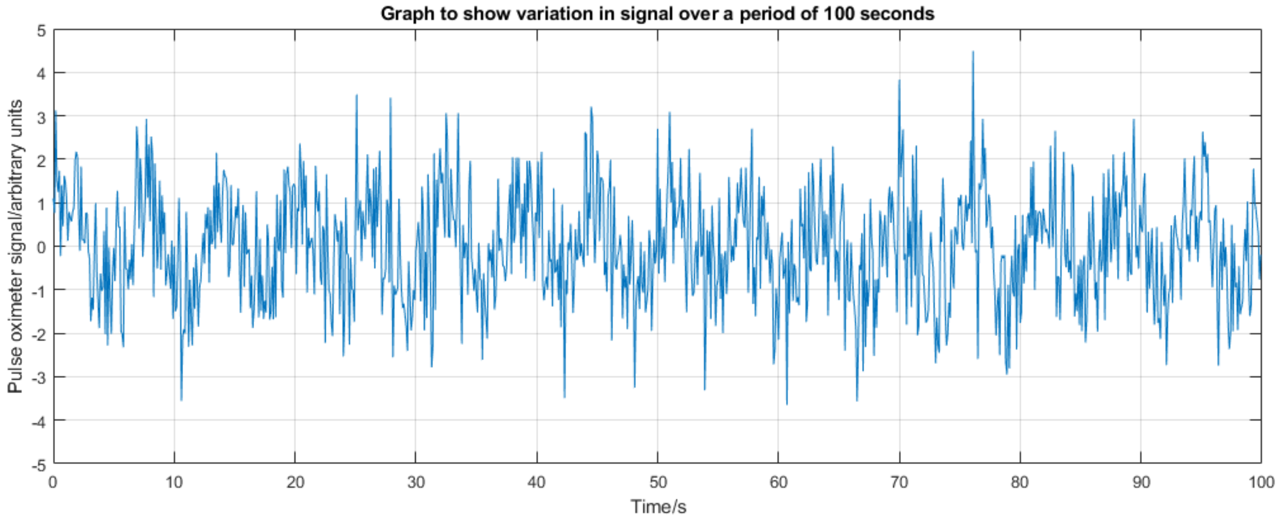

Figure 1: Graph to show variation in signal over a period of 100 seconds.

3.2 Plot of Fourier transform

```
clc
clear
close all

%import data
data = readmatrix('Section3_data.txt');

y = fft(data(:,2)); %compute discrete Fourier transform of data, (fast Fourier transform algorithim), indexing pulse oximeter data
n = length(data(:,2)); %find length of matrix
Fs = 10; % Sampling frequency (Hz)
f =(0:n-1)*(Fs/n); % Frequency range
fshift = (-n/2:n/2-1)*(Fs/n); %defines x-axis range for shifted transform
yshift = fftshift(y); %shifts zero-frequency component to centre of the array, this swaps the left and the right halves of x
figure;

%plot data
plot(fshift, abs(yshift))
title('Graph to show absolute values of transform in the frequency domain')
xlabel('Frequency/Hz')
ylabel('Fourier transform of signal data/arbitrary units')
axis square
grid on
```

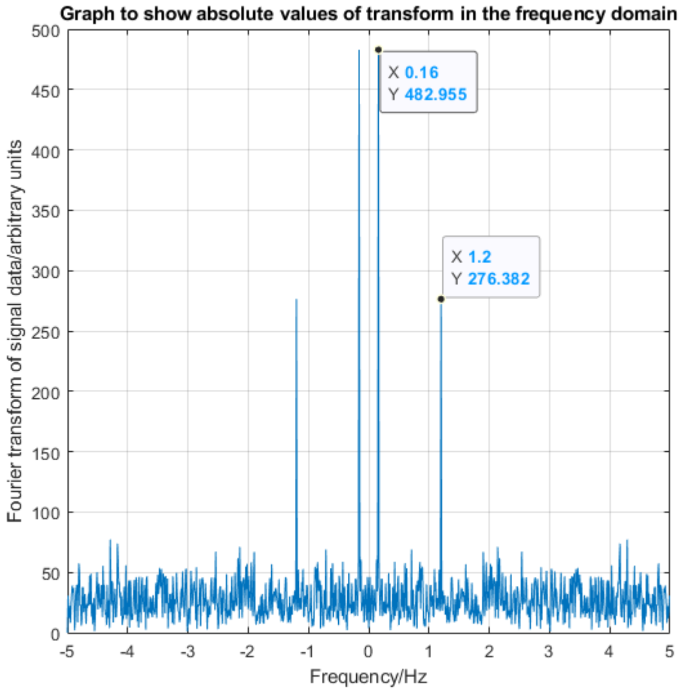

Figure 2: Graph to show absolute values of transform in the frequency domain.

3.3 Extraction of patient's cardiac and respiratory cycle

As seen from Figure 2, we can extract two values from our Fourier transform. The higher peak has a frequency of 0.16 Hz and a period of 6.25 s. This represents the breathing of the subject (9.6 breaths per minute). According to a Cleveland Clinic article on vital signs, the average human breathing rate for adults should be around 12-16 breaths per minute [1]. The lower peak has a frequency of 1.2 Hz and a period of 0.83 s. This represents the heartbeat of the subject (72 beats per minute). According to the British Heart Foundation, the average resting heart rate for adults is between 60-100 beats per minute [2].

3.4 Frequency filter

3.4.1 Gaussian functions

A Gaussian function was generated using MATLAB's "gaussmf" function. $\mu = \pm 1.2$. The value for σ was selected arbitrarily to de-noise the signal to an appropriate level

```
clc
clear
close all
%import data
data = readmatrix('Section3_data.txt');

y = fft(data(:,2)); %compute discrete Fourier transform of data, (fast Fourier transform algorithim), indexing pulse oximeter data
n = length(data(:,2)); %find length of matrix
Fs = 10; % Sampling frequency (Hz)
```

```
fshift = (-n/2:n/2-1)*(Fs/n); %defines x-axis range for shifted transform
z = [gaussmf(fshift , [0.01 1.2])' + gaussmf(fshift , [0.01 -1.2])'];% generate and add gaussians

%plot data
plot(fshift , z)
title('Graph to show filter , centred at positive and negative cardiac frequencies')
axis square;
grid on
xlabel('Frequency/Hz')
ylabel('Magnitude/arbritrary units')
```

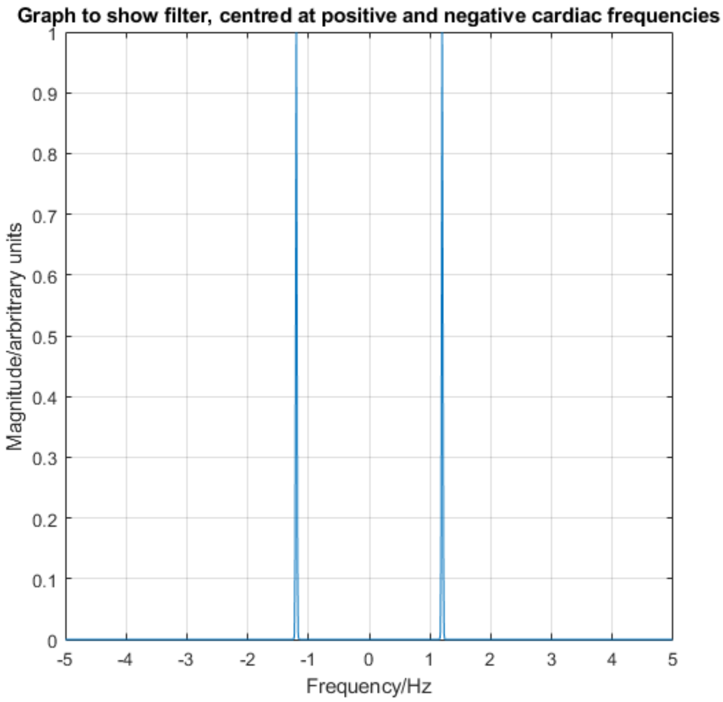

Figure 3: Graph to show filter, centred at positive and negative cardiac frequencies.

3.4.2 Filtered/unfiltered Fourier data comparison

```
clc
clear
close all

%import data
data = readmatrix('Section3_data.txt');

y = fft(data(:,2)); %compute discrete Fourier transform of data, (fast Fourier transform algorithim), indexing pulse oximeter data
n = length(data(:,2)); %find length of matrix
Fs = 10; % Sampling frequency (Hz)
f =(0:n-1)*(Fs/n); % Frequency range
fshift = (-n/2:n/2-1)*(Fs/n); %defines x-axis range for shifted transform
```

```
yshift = fftshift(y); %shifts zero-frequency component to centre of the
    array, this swaps the left and the right halves of x
z = [gaussmf(fshift, [0.01 1.2])' + gaussmf(fshift, [0.01 -1.2])'];%
    generate and add gaussians
filtData = abs(yshift).*z; %multiply FT signal data with gaussian
figure;

%plot data
plot(fshift, abs(yshift),fshift, filtData)
title('Graph to show comparison between filtered and unfiltered FT signal
    ')
xlabel('Frequency/Hz')
ylabel('Fourier transform of signal data/arbitrary units')
legend('Filtered data', 'Unfiltered data')
axis square
grid on
figure(2);
plot(fshift, filtData, fshift, abs(yshift))
xlim([1 2])
ylim([0 150])
title('Graph to show comparison between filtered and unfiltered FT signal
    ')
xlabel('Frequency/Hz')
ylabel('Fourier transform of signal data/arbitrary units')
legend('Filtered data', 'Unfiltered data')
axis square
grid on
```

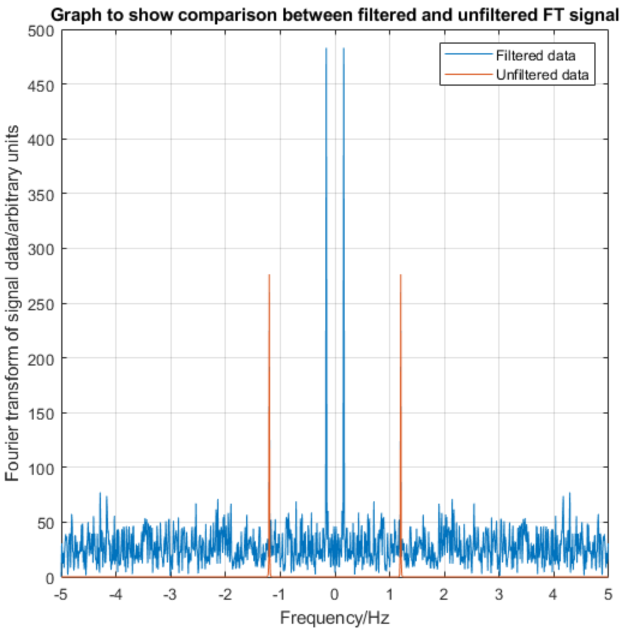

Figure 4: Graph to show comparison between filtered and unfiltered FT signal.

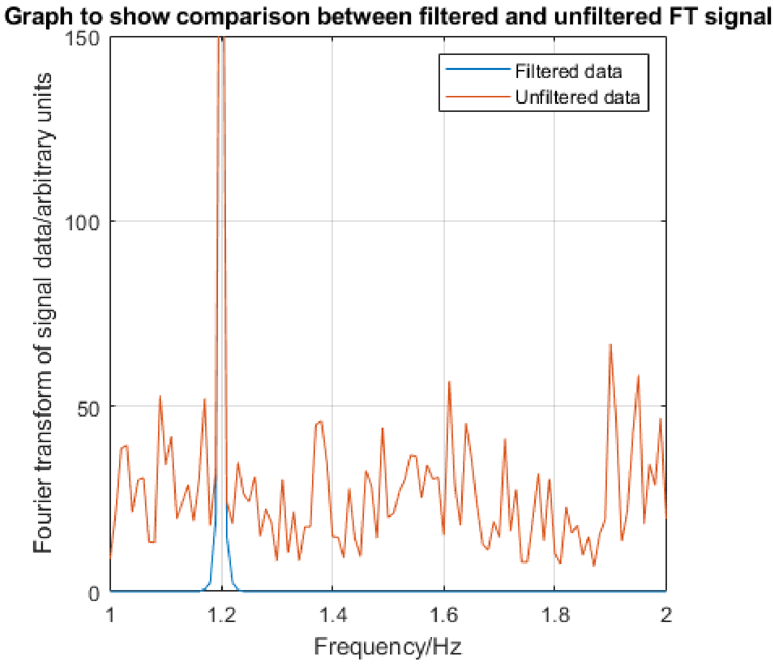

Figure 5: Graph to show comparison between filtered and unfiltered FT signal (close-up).

3.5 Filtered data

```
clc
clear
close all

%import data
data = readmatrix('Section3_data.txt');

y = fft(data(:,2)); %compute discrete Fourier transform of data, (fast Fourier transform algorithim), indexing pulse oximeter data
n = length(data(:,2)); %find length of matrix
Fs = 10; % Sampling frequency (Hz)
f =(0:n-1)*(Fs/n); % Frequency range
fshift = (-n/2:n/2-1)*(Fs/n); %defines x-axis range for shifted transform
yshift = fftshift(y); %shifts zero-frequency component to centre of the array, this swaps the left and the right halves of x
z = [gaussmf(fshift, [0.01 1.2])' + gaussmf(fshift, [0.01 -1.2])'];%generate and add gaussians
filtData = abs(yshift).*z; %multiply FT signal data with gaussian
y2 = ifftshift(filtData); %inverse zero frequency shift
x2 = ifft(y2); %inverse fourier
figure;

%plot data
plot(data(:,1), x2)
title('Graph to show filtered data from pulse oximeter')
xlabel('Time/s')
ylabel('Pulse oximeter signal/arbitrary units')
axis auto
grid on
```

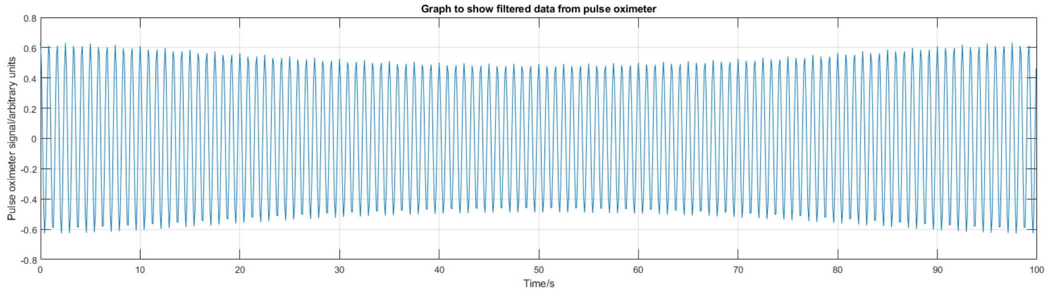

Figure 6: Graph to show filtered data from pulse oximeter.

3.6 Effect of varying the width of Gaussian function

The code was adjusted to created two additional cases, to make four in total:

- Unfiltered data
- Gaussian filter with $\sigma = 0.1$
- Gaussian filter with $\sigma = 0.01$
- Gaussian filter with $\sigma = 0.001$

```
clc
clear
close all

%import data
data = readmatrix('Section3_data.txt');

y = fft(data(:,2)); %compute discrete Fourier transform of data, (fast
    Fourier transform algorithim), indexing pulse oximeter data
n = length(data(:,2)); %find length of matrix
Fs = 10; % Sampling frequency (Hz)
f =(0:n-1)*(Fs/n); % Frequency range
fshift = (-n/2:n/2-1)*(Fs/n); %defines x-axis range for shifted transform
yshift = fftshift(y); %shifts zero-frequency component to centre of the
    array, this swaps the left and the right halves of x
z1 = [gaussmf(fshift, [0.01 1.2])' + gaussmf(fshift, [0.01 -1.2])'];%
    generate and add gaussians
z2 = [gaussmf(fshift, [0.1 1.2])' + gaussmf(fshift, [0.1 -1.2])'];%
    generate and add gaussians
z3 = [gaussmf(fshift, [0.001 1.2])' + gaussmf(fshift, [0.001 -1.2])'];%
    generate and add gaussians
filtData1 = abs(yshift).*z1; %multiply FT signal data with gaussian 0.1
filtData2 = abs(yshift).*z2; %multiply FT signal data with gaussian 0.01
filtData3 = abs(yshift).*z3; %multiply FT signal data with gaussian 0.001
y21 = ifftshift(filtData1); %inverse zero frequency shift 0.1
x21 = ifft(y21); %inverse fourier
y22 = ifftshift(filtData2); %inverse zero frequency shift 0.01
x22 = ifft(y22); %inverse fourier
```

```
y23 = ifftshift(filtData3); %inverse zero frequency shift 0.001
x23 = ifft(y23); %inverse fourier
figure;

%plot data
subplot(2,2,1)
plot(fshift, abs(yshift))
title('unfiltered')
xlim([0.7 1.7])
ylim([0 150])
axis square
grid on
subplot(2,2,2)
plot(fshift, filtData2)
title('stdev = 0.1')
xlim([0.7 1.7])
ylim([0 150])
axis square
grid on
subplot(2,2,3)
plot(fshift, filtData1)
xlim([0.7 1.7])
ylim([0 150])
title('stdev = 0.01')
axis square
grid on
subplot(2,2,4)
plot(fshift, filtData3)
xlim([0.7 1.7])
ylim([0 150])
title('stdev = 0.001')
axis square
grid on

figure(2)
subplot(4,1,1)
plot(data(:,1), data(:,2))
title('unfiltered')
axis auto
grid on
subplot(4,1,2)
plot(data(:,1), x22)
title('stdev = 0.1')
axis auto
grid on
subplot(4,1,3)
plot(data(:,1), x21)
title('stdev = 0.01')
axis auto
grid on
subplot(4,1,4)
plot(data(:,1), x23)
title('stdev = 0.001')
```

```
axis auto
grid on
```

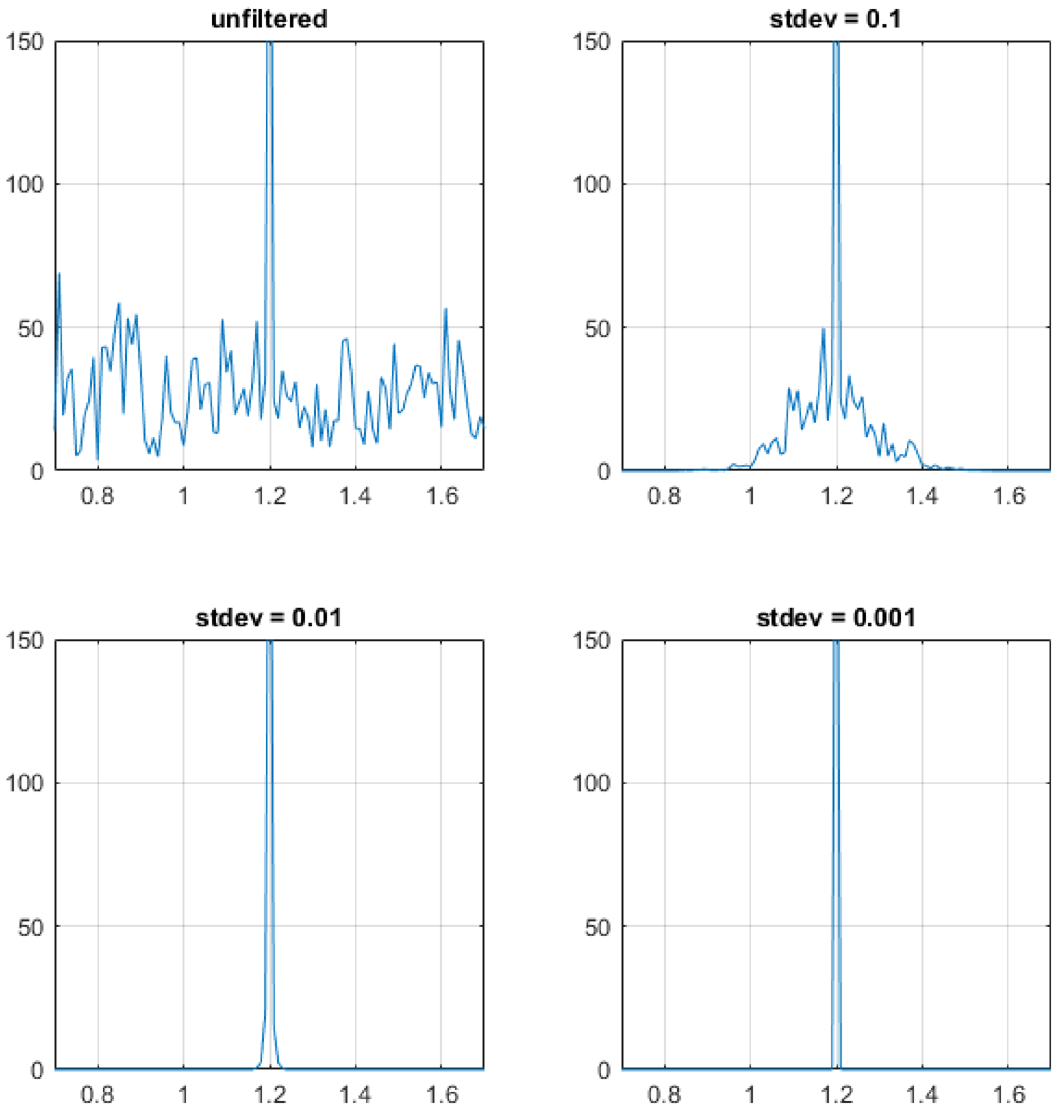

Figure 7: Graphs to compare the effect of varying Gaussian filter width on FT signal.

Here we can see that adjusting the value of σ effects the amount of noise that appears at the base of the peak in the Fourier transformed data. For $\sigma = 0.1$, there is still quite a bit of residual noise. $\sigma = 0.01$ and $\sigma = 0.001$ both do not exhibit any noise at the base, but we can see that for $\sigma = 0.01$, there is a slight flaring at the base.

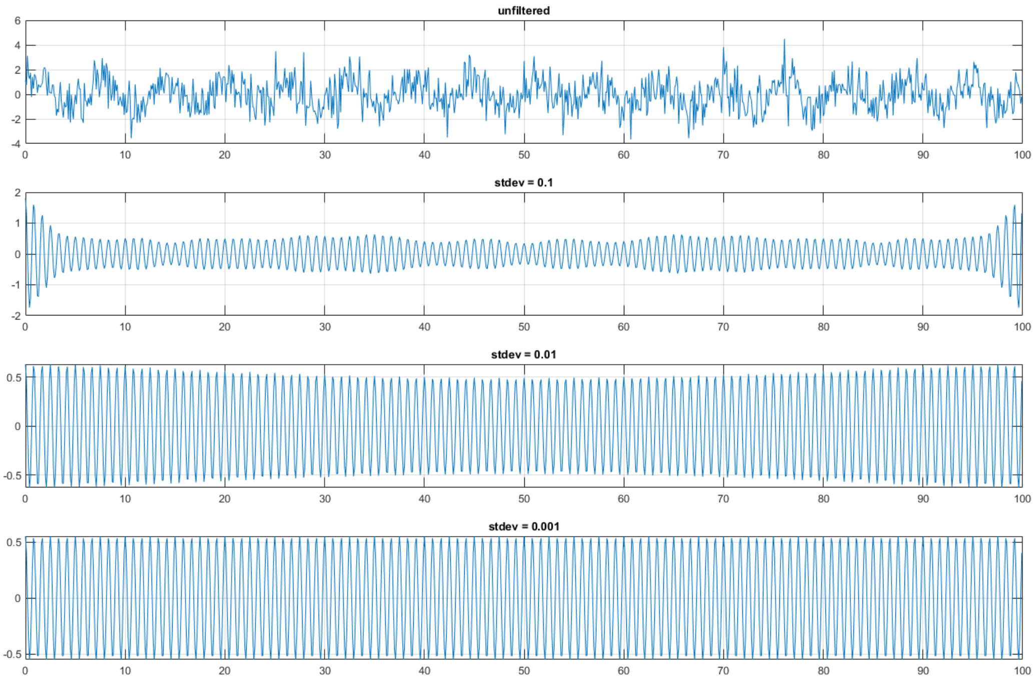

Figure 8: Graphs to compare the effect of varying Gaussian filter width on signal from pulse oximeter.

Here we can see the effect of the residual noise in the $\sigma = 0.1$ case, with relatively large variations in the amplitude of the signal. We can also see the effect of the flared base in the $\sigma = 0.01$ case as a smooth decrease and then increase in the amplitude of the signal. The $\sigma = 0.001$ case represents a virtually perfect signal with a frequency of 1.2 Hz.

4 Statistics

References

[1] Cleveland Clinic, "Vital Signs", https://www.hopkinsmedicine.org/health/conditions-and-diseases/vital-signs-body-temperature-pulse-rate-respiration-rate-blood-pressure#:~:text=Respiration%20rates%20may%20increase%20with,to%2016%20breaths%20per%20minute. Accessed 27/04/21 14:47

[2] British Heart Foundation, "What is a normal pulse rate?", https://www.bhf.org.uk/informationsupport/heart-matters-magazine/medical/ask-the-experts/pulse-rate#:~:text=A%20normal%20resting%20heart%20rate,rich%20blood%20around%20the%20body. Accessed 27/04/21 14:45