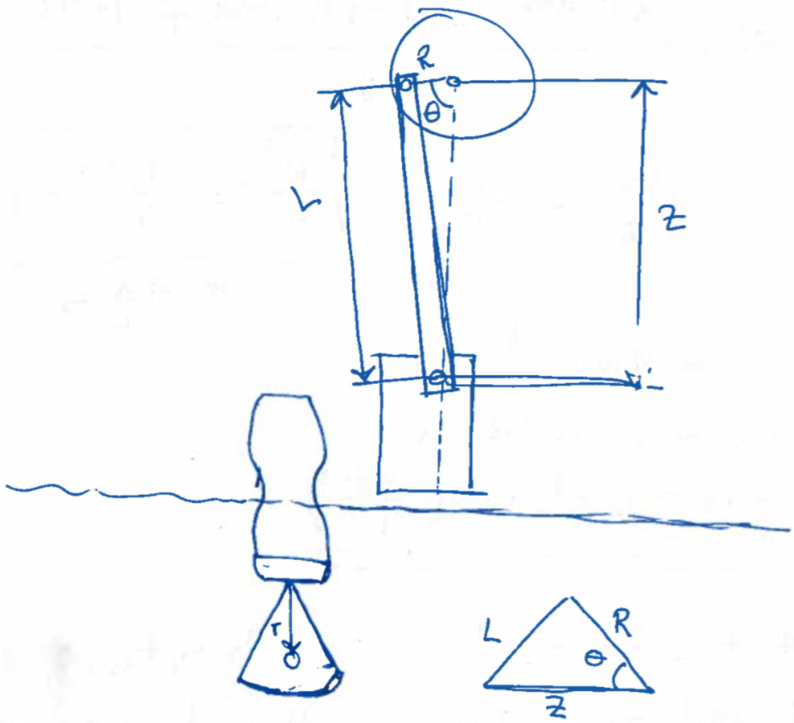
Exercise 8 - Ultrasound signals from moving targets

Part 1 - Analytic expression for velocity

Part 1



θ is a function of time and period of the rotation:

$\theta = \omega t$ ← Angular frequency

$\omega = 2\pi f$

$f = 1/T$ ← Rotation period

$$\theta(t) = \frac{2\pi t}{T}$$

Law of cosines:

$$L^2 = R^2 + z^2(t) - 2R \cdot z(t) \cdot \cos\left(\frac{2\pi t}{T}\right)$$

$$0 = z^2(t) - z(t)\left(2R\cos\left(\frac{2\pi t}{T}\right)\right) - (L^2 - R^2)$$

$\Downarrow$ $L >> R$

$$0 = z^2(t) - z(t)\left(2R\cos\left(\frac{2\pi t}{T}\right)\right) - L^2 \quad (1)$$

$$z(t) = \frac{2R\cos\left(\frac{2\pi t}{T}\right) \pm \sqrt{R^2\cos\left(\frac{2\pi t}{T}\right) + 4L^2}}{2}$$

$$= \frac{2R}{2}\cos\left(\frac{2\pi t}{T}\right) \pm L\underbrace{\sqrt{\frac{R^2}{4L^2}\cos\left(\frac{2\pi t}{T}\right) + 1}}_{L>>R \to \frac{R}{L} \to 0}$$

$$= R\cos\left(\frac{2\pi t}{T}\right) \pm L$$

Use first solution:

$$z(t) = L + R\cos\left(\frac{2\pi t}{T}\right)$$

At the time of "bottom, t_0", z will equal $L+R$ and the displacement will have maximum value.

$$v(t) = \frac{dz(t)}{dt} = \frac{d}{dt}\left(L + R\cos\left(\frac{2\pi t}{T}\right)\right)$$

$$= -R \cdot \frac{2\pi}{T}\sin\left(\frac{2\pi t}{T}\right)$$

$$= \underline{\underline{-\frac{2\pi R}{T}\sin\left(\frac{2\pi t}{T}\right)}}$$

with reference velocity being downwards direction

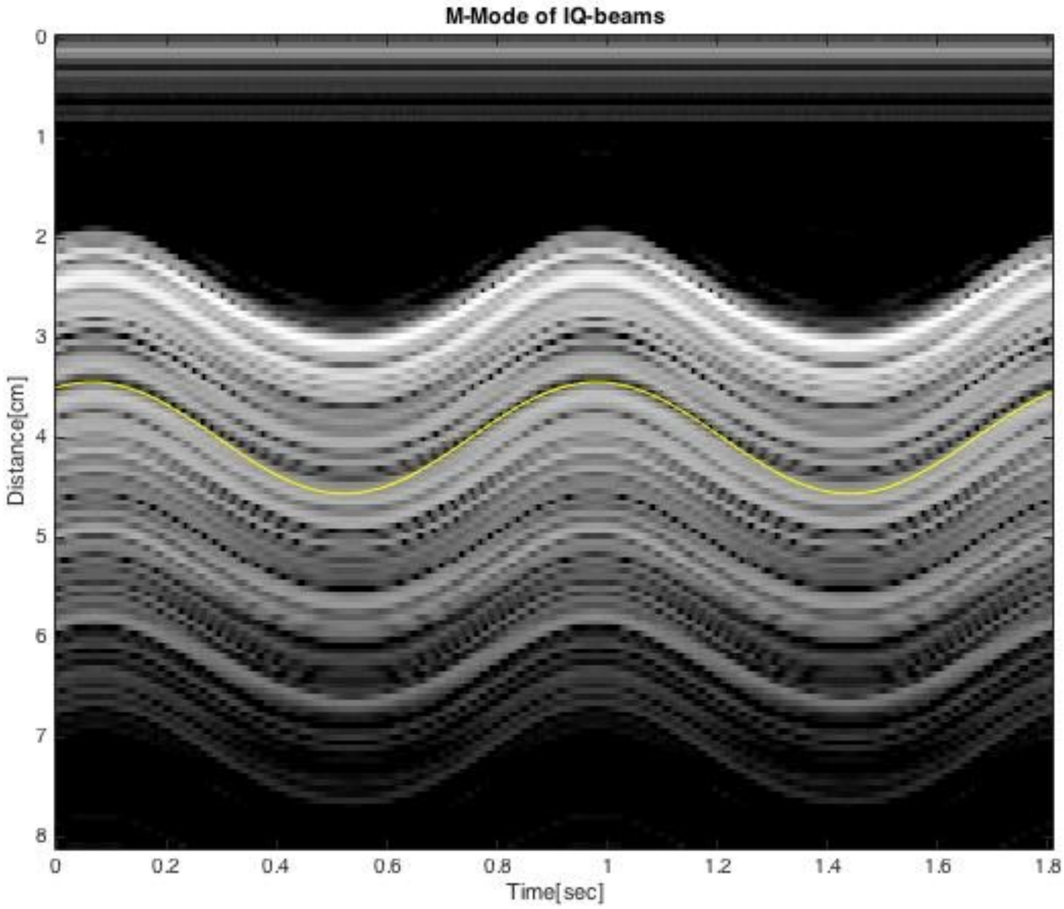
Find relationship between $r'(t)$ and $v(t)$:

Changes in $r(t)$ by $r'(t)$ is a product of the movement $v(t)$, but in the opposite direction

$$\underline{\underline{r'(t) = -v(t)}}$$

Part 2 - M-mode

Estimate parameters based on image



Estimate rotation period T by measuring the distance between two tops in the M-mode image:

T = 0.917 seconds

Estimate the time t0 by measuring the time the point in the phantom is at it's lowest point:

t0 = 0.53 seconds

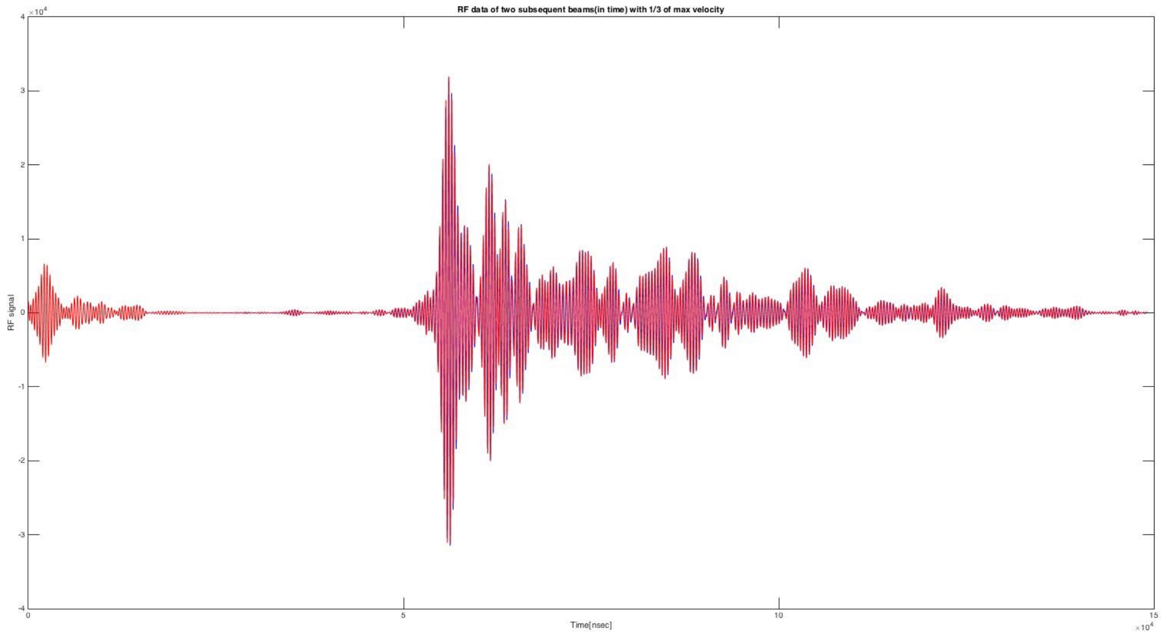
Estimate the excenter distance R by measuring half the height of the sin-wave in the M-mode image:

R = 0.487 cm

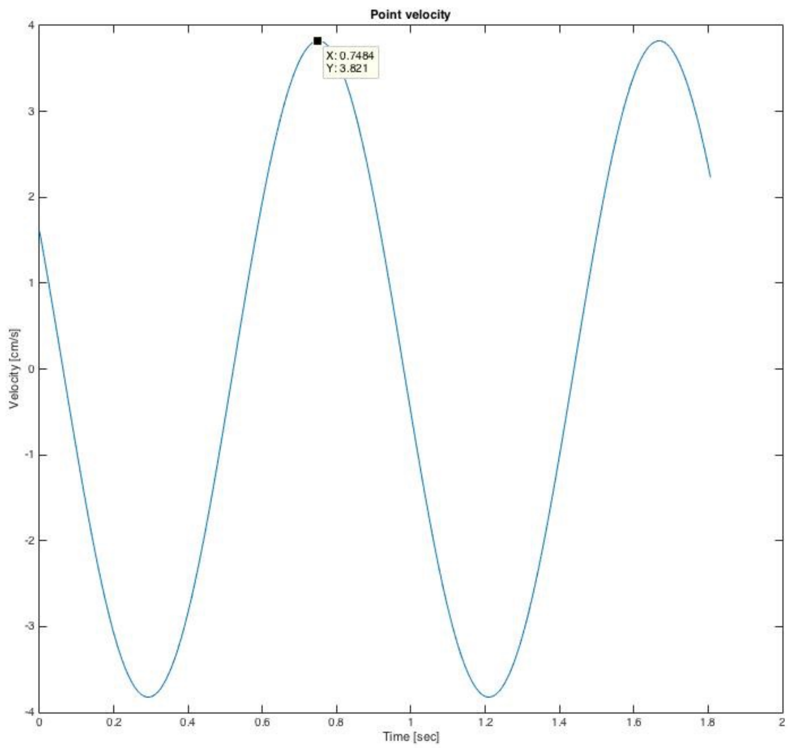
Compare calculated movement to the observed movement:

The calculated movement(yellow line) fits well with the observed movements in the image.

Part 3 - Time-shift in the RF-data



Plot the analytically calculated r'(t)



Calulate radial displacement and time shift

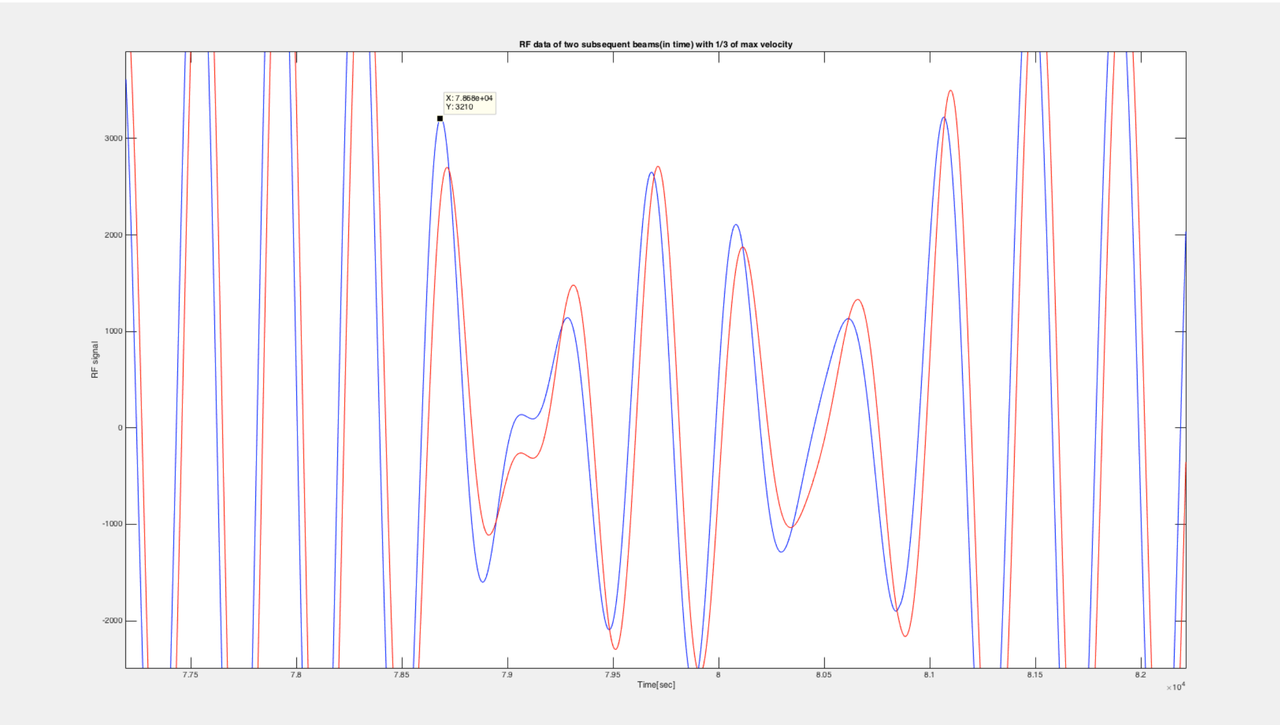
Analytic expression for the movement gives velocity: 1.148 cm/s

Calculated radial displacement: 0.00328 cm

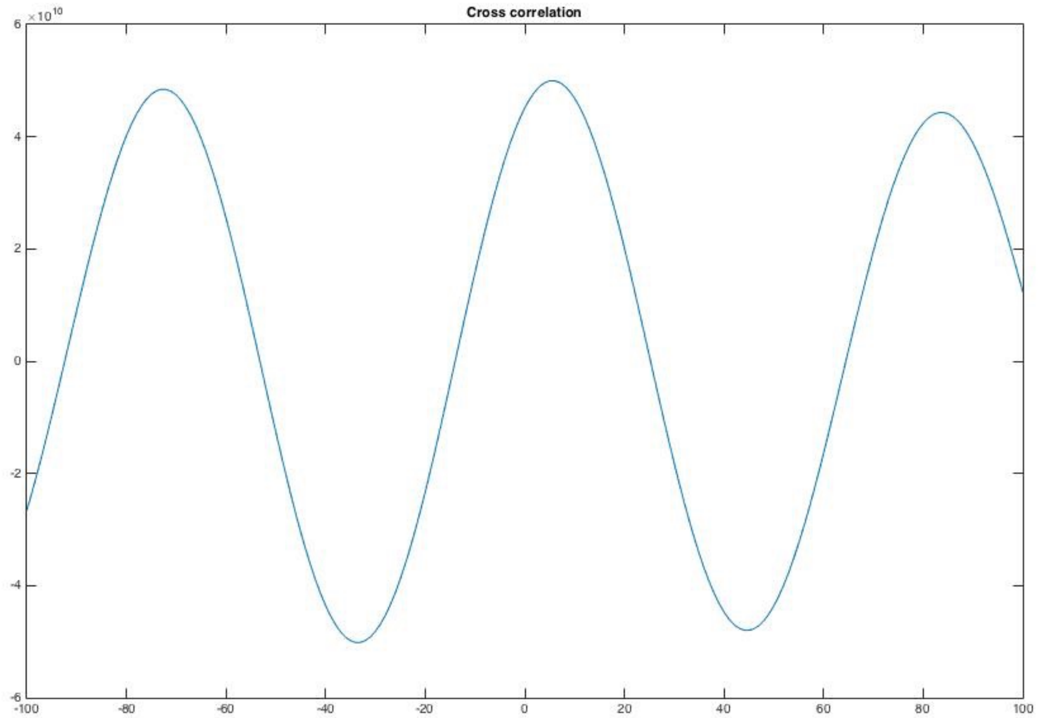
Calculated time shift: 42.6839 nsec

Convert to RF-data and plot both beams

Measured time shift in the RF-plot : 40 ns



Part 4 - Cross-correlation and phase shift in the RF-data



After 6 samples (e.g 30 nsec) the cross-correlation has its maximum

Analytical time shift: 42.753 nsec

Calculated phase shift: 0.471239 radians

Analytical phase shift: 0.671563 radians

Part 5 - The autocorrelation method

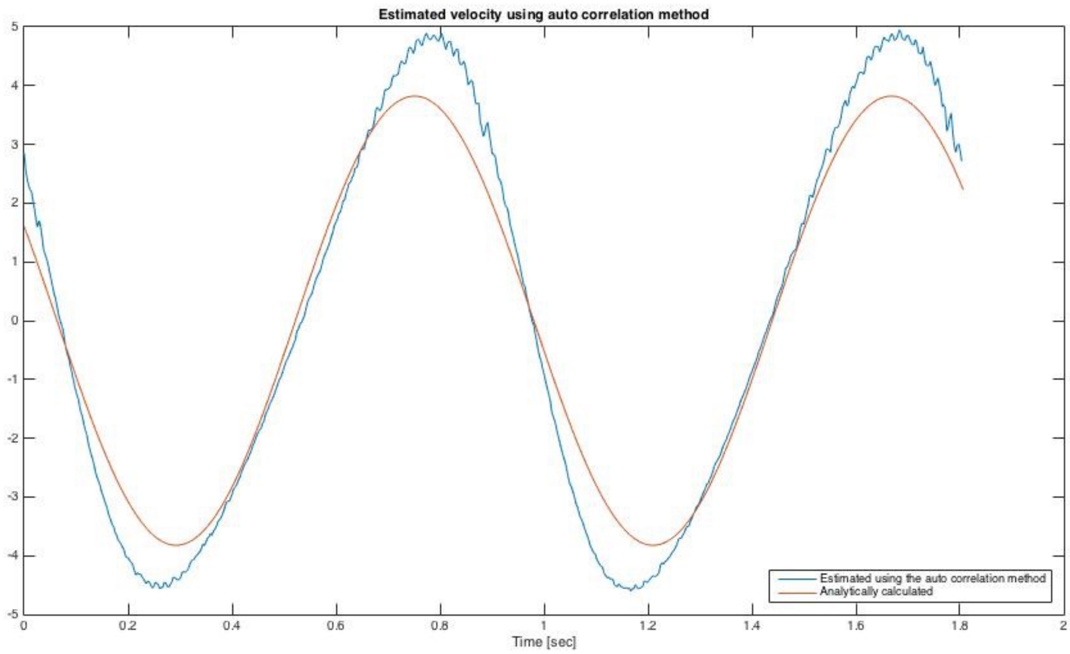
Compare phase shift with phase shift from previous part

Calculated phase shift using autocorrelation method: 0.453893 radians

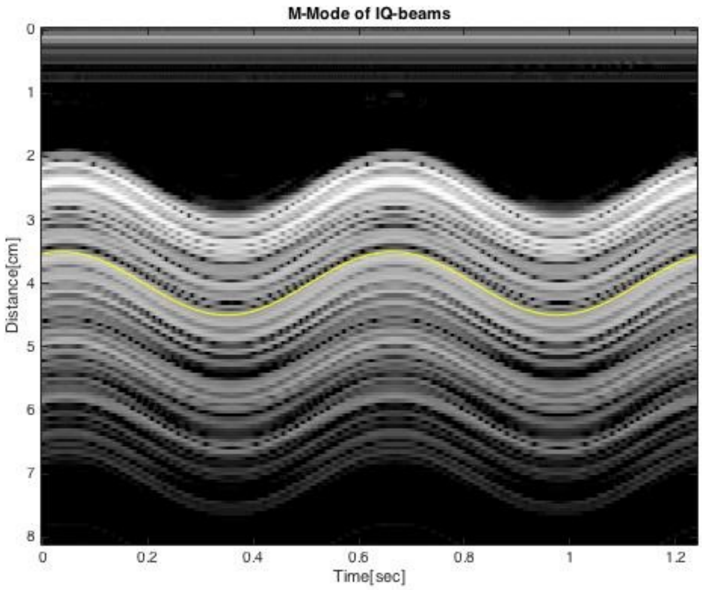
Give an expression for relation between velocity, frame rate, wavelength and phase-shift

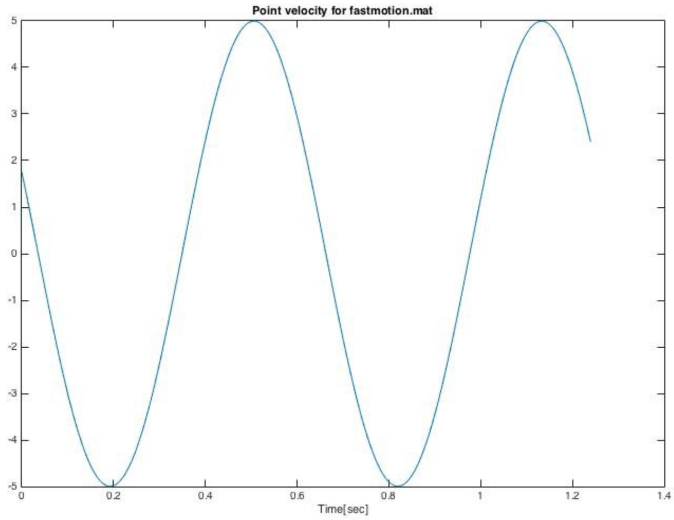
$$\Delta r = \frac{v}{FR}$$
$$\Delta t = 2\frac{\Delta r}{c} = 2\frac{v}{FR * c}$$
$$\Delta\Theta = 2\pi\frac{\Delta t}{T} = 2\pi\Delta t f = 2\pi\frac{\Delta t c}{\lambda}$$
$$\Delta\Theta = 4\pi\frac{v}{FR * \lambda}$$
$$v = \frac{\lambda FR}{4\pi}\Delta\Theta$$

Estimate phase-shift for all pairs of neighbouring beams, and plot velocity



Part 6 - Aliasing



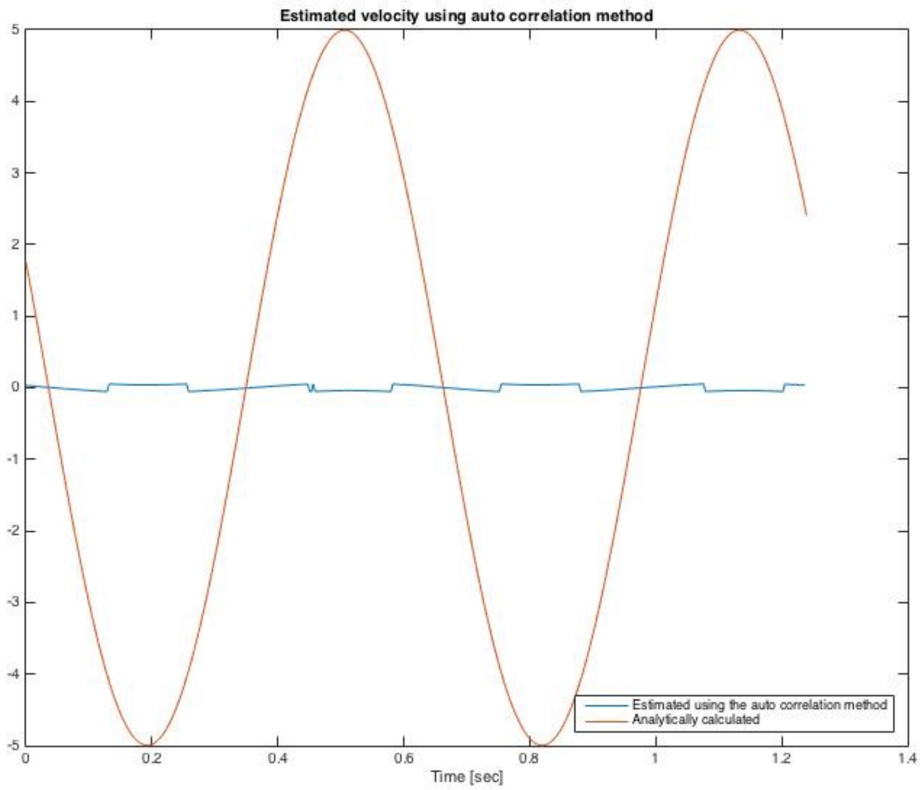


Measured rotation period:

T = 0.63 seconds

Measured time offset:

t0 = 0.35 seconds



In the "fastmotion" test case the velocity of the movement is too fast compared with the given frame rate. The fast motion leads to aliasing, which can be seen in the plot where the estimated velocity is shown as considerably smaller than the analytic. Using the auto correlation method for estimating the velocity fails in the case of aliasing, as the measured signal will not match the produced movements, and we lose too much essential information.

Doppler frequency shift is given by:

$$f_d = \frac{2f_0}{c} v$$

$$f_d < \text{Framerate}$$

$$FR > \frac{2f_0}{c} v \Rightarrow v < \frac{c \cdot FR}{2 f_0}$$

For this case:

$FR = 349.32\,Hz$, $f_0 = 2.5\,MHz$, $C = 1540\,m/s$

$$v < \frac{349.32\,Hz \cdot 1540\,m/s}{2 \cdot 2.5 \cdot 10^6\,Hz} = 10.76\,cm/s$$

If $v < 20\,cm/s \Rightarrow FR = ?$:

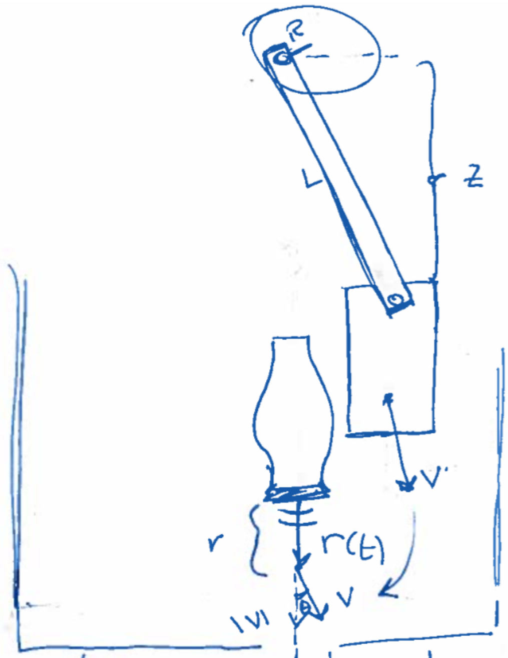
$$20\,c\cancel{m/s} = \frac{FR \cdot 1540 \cdot 100\,\cancel{cm}/s}{2 \cdot 2.5 \cdot 10^6\,Hz}$$

$$FR = \frac{20 \cdot 2 \cdot 2.5 \cdot 10^6\,Hz}{1540 \cdot 100} = 649.4\,Hz$$

If $v < 20\,cm/s$ & $f_0 = 1.67\,MHz \Rightarrow FR = ?$:

$$FR = \frac{20}{1540 \cdot 100} \cdot 2 \cdot 1.67 \cdot 10^6\,Hz = 433.8\,Hz$$

Part 7 - Angle between velocity and beam



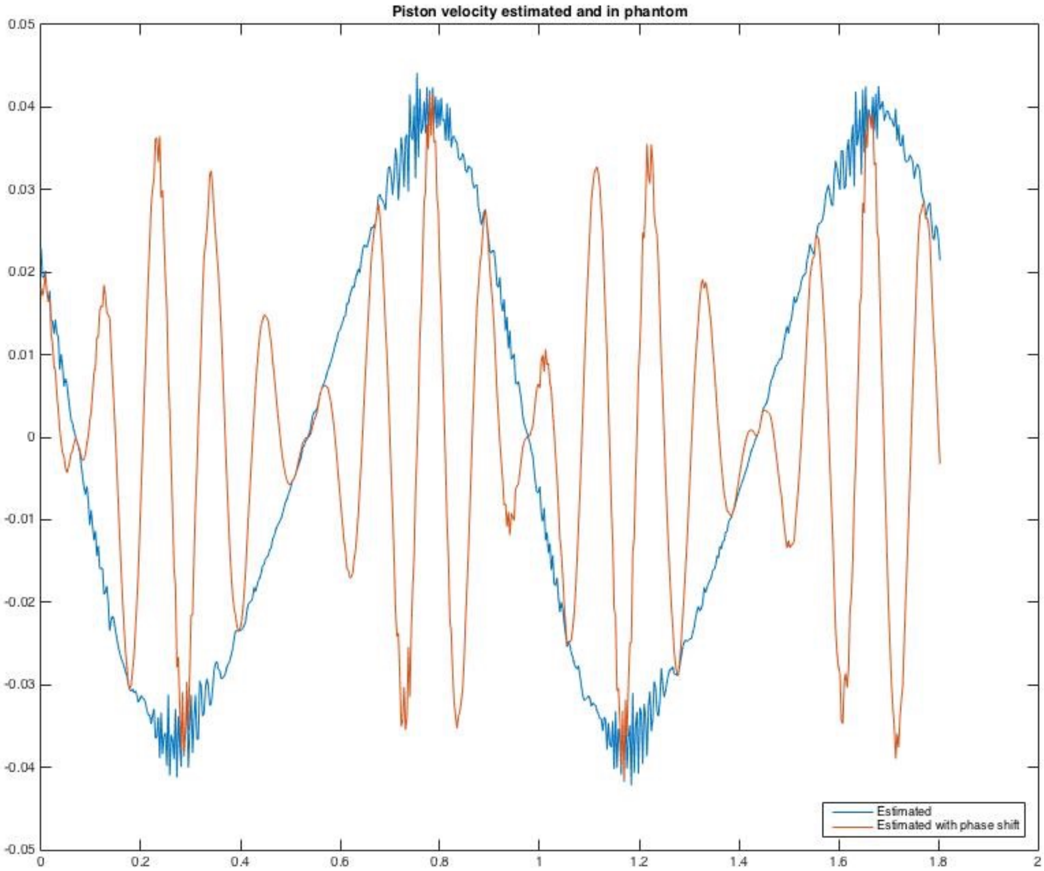
Change in frequency due to the movement of the piston is:

$$f_d = -\frac{2f_0}{c} v$$

where v is:

$$v = |v| \cos\theta$$

where |v| is the amplitude of v in direction of r. θ is the difference in angle between r and v.



Introducing phase-shift in the estimated velocity shows that the velocity which hits the phantom in the direction of the ultrasonic beam is the envelope of the velocity of the piston.

Appendix I - Source code

```
% Ex8.m
% Purpose:
%   Script to answer tasks in exercise 8 given in course
%   TTK4165 Medical Signal Processing
%
% Made by:
%   Even Florenes NTNU 2016
%
% Last changes:
%   2016-03-08: First attemptsp
%

%% PART 2
load slowmotion

middleBeamIq = squeeze(iq(:,4,:));

frameRate = s.Framerate_fps; % nFrames/seconds

nFrames = size(middleBeamIq,2); %nFrames

nSamples = size(middleBeamIq,1);

nSeconds = nFrames/frameRate;   %seconds = (nFrames/(nFrames/seconds))

time = 0:nSeconds/(nFrames-1):nSeconds;

distanceLength = s.iq.DepthIncrementIQ_m;

distance = 0:distanceLength/(nSamples-1):distanceLength;

amplitude = abs(middleBeamIq);
power = amplitude.^2;

gain = -50;
dynamicRange = 50;

logarithimicPower = imagelog(power,gain,dynamicRange);

figure,imagesc(time,10^4*distance,logarithimicPower),colormap(gray),...
   title('M-Mode of IQ-beams'),xlabel('Time[sec]'),ylabel('Distance[cm]');
hold on
%% Subpart 1:Measure parameters
[x,y] = ginput(3);

rotationPeriod = x(2)-x(1); % T [s]

t0 = x(3);

excenterDistance = (y(3)-mean([y(1),y(2)]))/2;
%y(1) and y(2) are supposed to
%be the same height, but
%measurements may misaligne.
```

```
%Use mean of the measurements
%to get best possible precision

fprintf('T= %g sec\n',rotationPeriod);
fprintf('t0= %g sec\n',t0);
fprintf('R= %g cm\n',excenterDistance);
%% Subpart 2: Plot point position in M-mode image
pistonLength = 10;
fixedPoint = 4;
pistonAngularFrequency = (2*pi*(time-t0))/rotationPeriod;

%pistonVelocity = pistonVelocityAmplitude*sin(pistonVelocityAngularFrequency);
pistonPosition = pistonLength+excenterDistance*cos(pistonAngularFrequency);
pointPosition = fixedPoint+excenterDistance*cos(pistonAngularFrequency);

plot(time,pointPosition,'y'),hold off;

%% PART 3
pistonVelocityAmplitude = -(2*pi*excenterDistance)/rotationPeriod;
pistonVelocity = pistonVelocityAmplitude*sin(pistonAngularFrequency);

pointVelocity = -pistonVelocity;

figure,plot(time,pointVelocity),title('Point velocity'),xlabel('Time [sec]'),...
   ylabel('Velocity [cm/s]');

%% Subpart 1: Choose frame and calulate parameters
maxVelocity = max(pointVelocity);
maxVariance = (1/30)*maxVelocity;

framesBelowOneThirdVelocity = find(pointVelocity < (1/3)*maxVelocity+maxVariance);

framesWithThirdVelocity = find(pointVelocity(framesBelowOneThirdVelocity) > (1/3)*maxVelocity ...
      - maxVariance);

frameWithThirdVelocity = median(framesWithThirdVelocity);

imageFrames = middleBeamIq(:,frameWithThirdVelocity-1:frameWithThirdVelocity);

figure,plot(10^4*distance,imageFrames),title('Two subsequent beams(in time) with 1/3 of max
velocity'),...
xlabel('Positon[cm]');

imageFramesVelocity = mean(pointVelocity(frameWithThirdVelocity-1:frameWithThirdVelocity));

radialDisplacement = imageFramesVelocity/frameRate;% [cm]

speedSound = 1540*100; %[cm/s]

timeShiftAnalytically = 2*(radialDisplacement/speedSound);

fprintf('Analytic velocity: %g cm/s\n',imageFramesVelocity);
fprintf('Calculated radial displacement = %g cm \n',radialDisplacement);
fprintf('Calculated time shift = %g nsec\n',10^(9)*timeShiftAnalytically);

samplingFrequencyRF=200e6; %RF sampling freq.
```

```
demodulationFrequency=s.iq.fDemodIQ_Hz; %IQ demodulation freq.
samplingFrequencyIQ=s.iq.frsIQ_Hz; %IQ sampling freq

firstRFBeam=iq2rf(imageFrames(:,
1),demodulationFrequency,samplingFrequencyIQ,samplingFrequencyRF);
secondRFBeam=iq2rf(imageFrames(:,
2),demodulationFrequency,samplingFrequencyIQ,samplingFrequencyRF);

timeRF=[1:length(firstRFBeam)]/samplingFrequencyRF;
timeRF = 10^9*timeRF;
figure,plot(timeRF,firstRFBeam,'b'),title('RF data of two subsequent beams(in time) with 1/3 of max
velocity'); hold on;
plot(timeRF,secondRFBeam,'r'); hold off;
ylabel('RF signal');
xlabel('Time[nsec]');

%% PART 4
crossCorrelation = zeros(1,201);
i = 1;
for l = -100:100

   for t = 15000:20000
      crossCorrelation(i) = crossCorrelation(i)+...
         firstRFBeam(t)*secondRFBeam(t+l);
   end
   i = i+1;
end

figure,plot(-100:100,crossCorrelation),title('Cross correlation');

sampleShiftMax = find(crossCorrelation  == max(crossCorrelation));
sampleShiftMax = (sampleShiftMax-101);
timeShiftCalculated = sampleShiftMax/samplingFrequencyRF;
timeShiftAnalytically = 2*abs(pointPosition(frameWithThirdVelocity-1)- ...
   pointPosition(frameWithThirdVelocity))/speedSound;

fprintf('After %d samples (e.g %g nsec) the cross-correlation has its maximum\n',sampleShiftMax,
10^9*timeShiftCalculated);
fprintf('Analytical time shift: %g nsec\n',timeShiftAnalytically*10^9);
phaseShiftCalculated = timeShiftCalculated*2*pi*demodulationFrequency;
phaseShiftAnalytically = timeShiftAnalytically*2*pi*demodulationFrequency;

fprintf('Calculated phase shift: %g radians \n',phaseShiftCalculated);
fprintf('Analytical phase shift: %g radians \n',phaseShiftAnalytically);

%% PART 5

phaseShiftAutoCorr = angle(mean(conj(imageFrames(60:80,2)).*imageFrames(60:80,1)));

fprintf('Calculated phase shift using autocorrelation method: %g radians \n',phaseShiftAutoCorr);

firstMiddleBeamIq = middleBeamIq(:,1:nFrames-1);
secondMiddleBeamIq = middleBeamIq(:,2:nFrames);

totalPhaseShift = angle(mean(conj(secondMiddleBeamIq).*firstMiddleBeamIq));
```

```
conversionValue = (speedSound*frameRate)/(4*pi*demodulationFrequency);

calculatedVelocity = conversionValue*totalPhaseShift;

figure(9),plot(time(1:630),calculatedVelocity,time,pointVelocity);
title('Estimated velocity using auto correlation method');
xlabel('Time [sec]');
legend('Estimated using the auto correlation method',...
    'Analytically calculated','Location','southeast');

%% PART 6

load fastmotion.mat
middleBeamIq = squeeze(iq(:,4,:));

frameRate = s.Framerate_fps; % nFrames/seconds

nFrames = size(middleBeamIq,2); %nFrames

nSamples = size(middleBeamIq,1);

nSeconds = nFrames/frameRate;   %seconds = (nFrames/(nFrames/seconds))

time = 0:nSeconds/(nFrames-1):nSeconds;

distanceLength = s.iq.DepthIncrementIQ_m;

distance = 0:distanceLength/(nSamples-1):distanceLength;

amplitude = abs(middleBeamIq);
power = amplitude.^2;

gain = -50;
dynamicRange = 50;

logarithimicPower = imagelog(power,gain,dynamicRange);

figure,imagesc(time,10^4*distance,logarithimicPower),colormap(gray),...
    title('M-Mode of IQ-beams'),xlabel('Time[sec]'),ylabel('Distance[cm]');
hold on
%% Subpart 1:Measure parameters
[x,y] = ginput(3);

rotationPeriod = x(2)-x(1); % T [s]

t0 = x(3);

excenterDistance = (y(3)-mean([y(1),y(2)]))/2;
%y(1) and y(2) are supposed to
%be the same height, but
%measurements may misalign.
%Use mean of the measurements
%to get best possible precision

fprintf('T= %g sec\n',rotationPeriod);
```

```
fprintf('t0= %g sec\n',t0);
fprintf('R= %g cm\n',excenterDistance);

%% Subpart 2: Plot position and velocity
pistonLength = 10;
fixedPoint = 4;

pistonAngularFrequency = (2*pi*(time-t0))/rotationPeriod;
pistonPosition = pistonLength+excenterDistance*cos(pistonAngularFrequency);
pointPosition = fixedPoint+excenterDistance*cos(pistonAngularFrequency);

plot(time,pointPosition,'y'),hold off;
pistonVelocityAmplitude = -(2*pi*excenterDistance)/rotationPeriod;
pistonVelocity = pistonVelocityAmplitude*sin(pistonAngularFrequency);

pointVelocity = -pistonVelocity;

figure,plot(time,pointVelocity),title('Point velocity for fastmotion.mat');
xlabel('Time[sec]');

%% Subpart 3: Calculate velocity by auto-correlation method
firstMiddleBeamIq = middleBeamIq(:,1:nFrames-1);
secondMiddleBeamIq = middleBeamIq(:,2:nFrames);
speedSound = 1540;
demodulationFrequency = s.iq.fDemodIQ_Hz;
totalPhaseShift = angle(mean(conj(secondMiddleBeamIq).*firstMiddleBeamIq));

conversionValue = (speedSound*frameRate)/(4*pi*demodulationFrequency);

calculatedVelocity = conversionValue*totalPhaseShift;

figure,plot(time(1:432),calculatedVelocity,time,pointVelocity);
title('Estimated velocity using auto correlation method');
xlabel('Time [sec]');
legend('Estimated using the auto correlation method',...
    'Analytically calculated','Location','southeast');


%% PART 7

load slowmotion.mat

firstIqBeam = squeeze(iq(:,1,:));
nFrames = size(firstIqBeam,2);

firstBeamIq = firstIqBeam(:,1:nFrames-1);
secondBeamIq = firstIqBeam(:,2:nFrames);
speedSound = 1540;
demodulationFrequency = s.iq.fDemodIQ_Hz;
frameRate = s.Framerate_fps;
totalPhaseShift = angle(mean(conj(secondBeamIq).*firstBeamIq));

conversionValue = (speedSound*frameRate)/(4*pi*demodulationFrequency);

calculatedVelocity = conversionValue*totalPhaseShift;
```

```
startAngle = s.iq.StartAngleIQ_rad;
angleIncrement = s.iq.AngleIncrementIQ_rad;

velocityInPhantom = zeros(1,nFrames-1);

for i = 1:nFrames-1

    velocityInPhantom(i) = abs(calculatedVelocity(i))*cos(startAngle+(i-1)*...
        angleIncrement);
end % for i

nSeconds = nFrames/frameRate;   %seconds = (nFrames/(nFrames/seconds))

time = 0:nSeconds/(nFrames-1):nSeconds;
time = time(1:630);

figure,plot(time,calculatedVelocity,time,velocityInPhantom),...
    title('Piston velocity estimated and in phantom');

legend('Estimated',...
    'Estimated with phase shift','Location','southeast');
```