Summer School on
Advanced Ultrasound Imaging

Part 1: Introduction to conventional velocity estimation

Jørgen Arendt Jensen
Center for Fast Ultrasound Imaging
Department of Health Technology
Technical University of Denmark

May 18, 2023

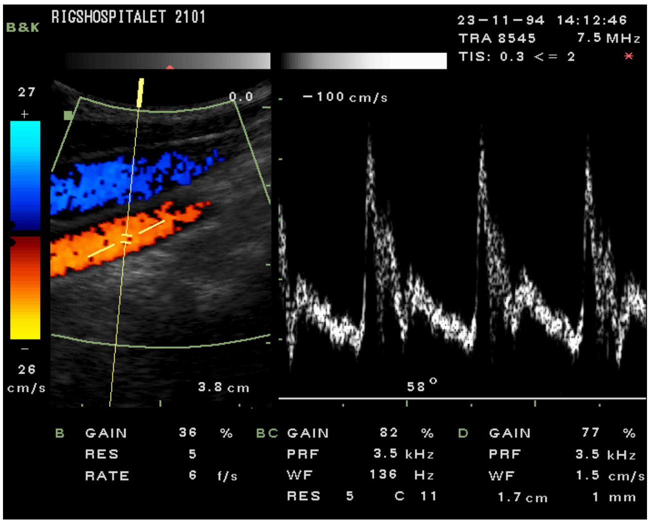
Outline

- Basic ultrasound and scattering from blood
- Model for ultrasound interaction with moving particles
- Physical effects and limitations in traditional velocity systems
- Color flow mapping - phase shift (autocorrelation) approach
- Color flow mapping - cross-correlation approach
- Stationary echo canceling
- Simulation of flow imaging systems

Characteristics of blood flow in humans

- Pulsating flow, repetition from 1 to 3 beats/sec
- Not necessarily laminar flow
- Short entrance lengths
- Branching
- Reynolds numbers usually below 2500, non-turbulent flow
- Very complicated flow patterns

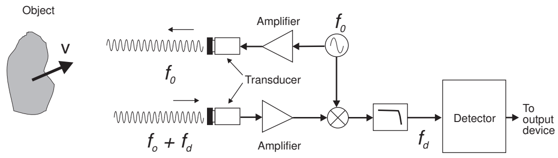
Conventional flow system



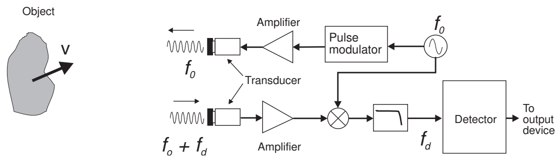
Triplex scan with B-mode image, color flow image, and spectral display. The square brackets indicate position and size of the range gate.

The classical Doppler effect

Continuous wave system



Pulsed wave system



Doppler shift

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

f_0 - Center frequency of transducer, c - Speed of sound,
v - Blood velocity

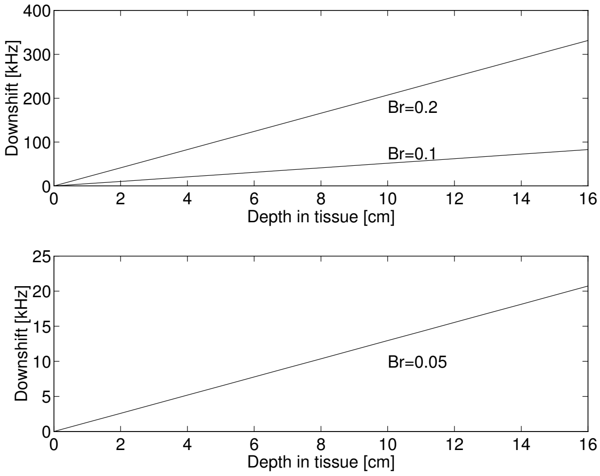
Effect of attenuation:

Down-shift in center frequency:

$$f_{mean} = f_0 - (\beta_1 B_r^2 f_0^2) z$$

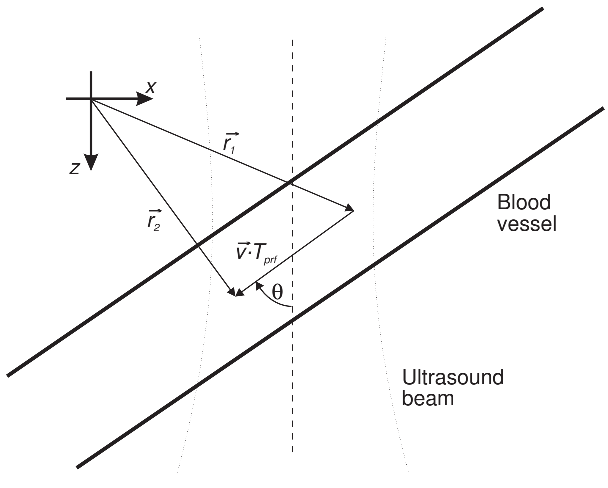
B_r - Gaussian bandwidth, β_1 - Attenuation coefficient [Np/Hz m]
z - Depth

Attenuation down shift



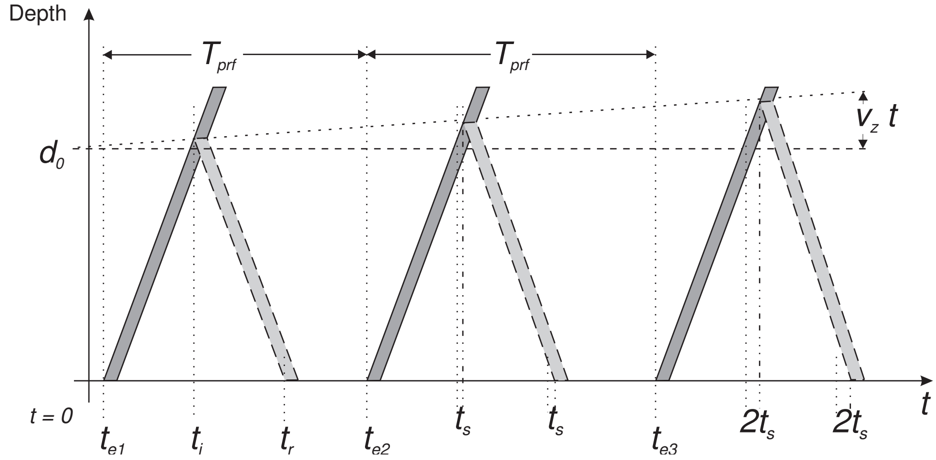
$f_0 = 3$ MHz, $\beta_1 = 0.5$ dB/[MHz·cm]
Typical Doppler shifts are 500 to 2000 Hz!

Basic measurement situation



$\vec{v}$ - Blood velocity, T_{prf} - Time between pulse emissions
$\vec{r}_1$ - Position at first emission, $\vec{r}_2$ - Position at second emission
θ - Angle between ultrasound beam and blood velocity vector

Time-space diagram for a number of pulse emissions and receptions



Time shift between emissions

$$t_s = \frac{2|v|\cos\theta}{c} T_{prf} = \frac{2v_z}{c} T_{prf}$$

Received signals

$$\begin{aligned} y_1(t) &= a \cdot e(t - \frac{2d}{c}) \\ y_2(t) &= a \cdot e(t - \frac{2d}{c} - t_s) = y_1(t - t_s) \end{aligned}$$

T_{prf} - Time between pulse emissions, v_z - Velocity along beam
c - Speed of sound, $e(t)$ - Emitted signal

Model for the received signals (single scatterer)

First emission:

$$r_0(t) = a \sin(2\pi f_0(t - \frac{2d}{c}))$$

Second emission:

$$r_1(t) = a \sin(2\pi f_0(t - \frac{2d}{c} - t_s))$$

i'th emission:

$$r_i(t) = a \sin(2\pi f_0(t - \frac{2d}{c} - t_s i))$$

Sampling at one depth

Measurement at one fixed time t_z or depth:

$$\phi = 2\pi f_0(t_z - \frac{2d}{c})$$

gives

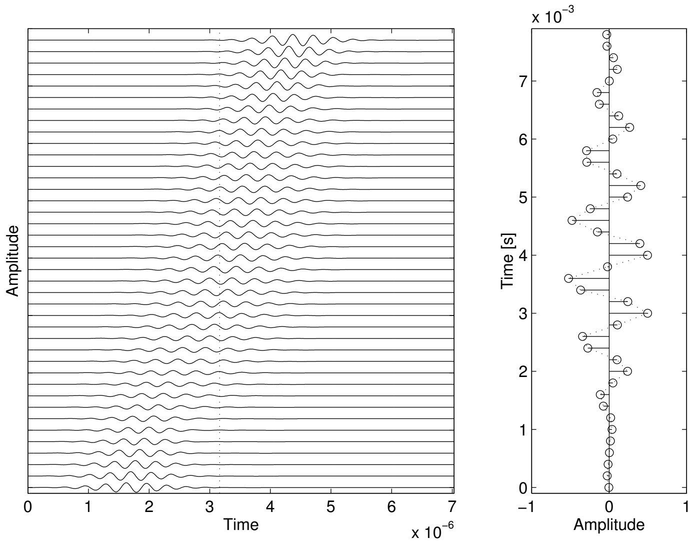
$$r_i(t_x) = -a \sin(2\pi f_0 t_s i - \phi) = -a \sin(2\pi \ \frac{2v_z}{c} f_0 \ (T_{prf} i) - \phi)$$

Frequency of sampled signal:

$$f_p = -\frac{2v_z}{c} f_0$$

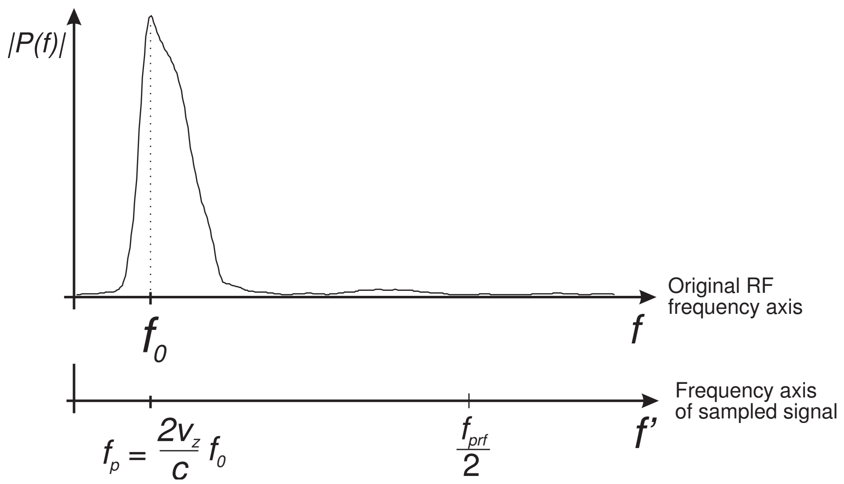
T_{prf} - Time between pulse emissions, v_z - Velocity along beam, c - Speed of sound, i - Emission number
f_0 - Center frequency

Signal from a moving scatterer crossing a beam



$$r_s(i) = -a \sin(2\pi \frac{2v_z}{c} f_0 T_{prf} i - \phi), \qquad \phi = 2\pi f_0 \left(t_z - \frac{2d}{c}\right)$$

Frequency axis scaling



Physical effects

Down shift in center frequency due to attenuation:

$$\Delta f = \beta_1 B_r^2 f_0^2 d_0$$

Down shift in resulting pulsed wave spectrum:

$$\Delta f_{pw,att} = \frac{2v_z}{c} \cdot \beta_1 B_r^2 f_0^2 d_0,$$

Doppler shift from motion of blood during pulse interaction:

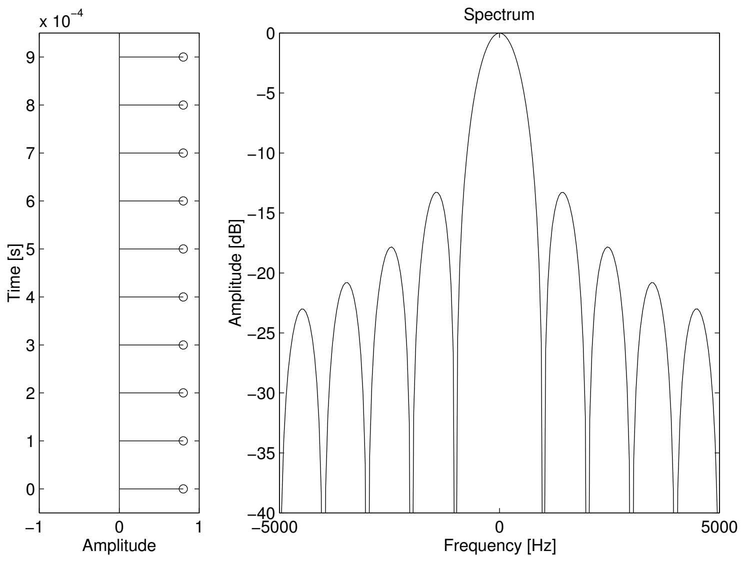
$$\Delta f_{pw,f_d} = \frac{2v_z}{c}\frac{2v_z}{c} f_0.$$

Non-linear components:

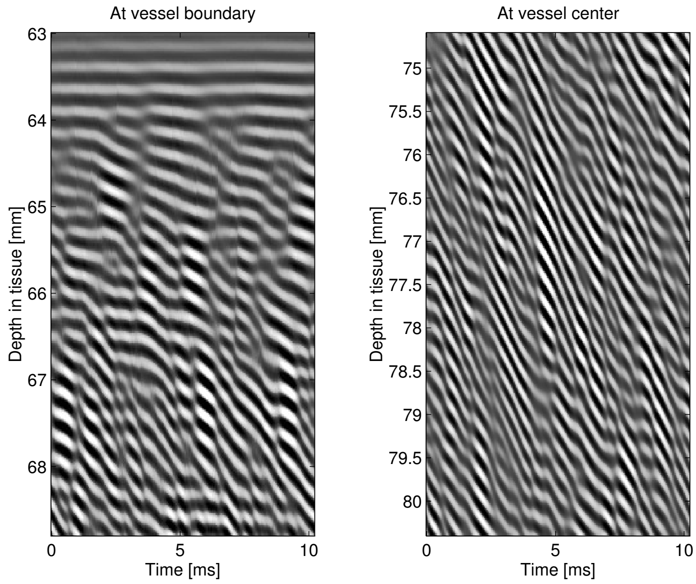
$$f_{\text{non-linear}} = \frac{2v_z}{c} f_{\text{har}}$$

Bias depends on whether $|f_{\text{non-linear}}| > f_{prf}/2$ or not.

Spectrum for stationary signal

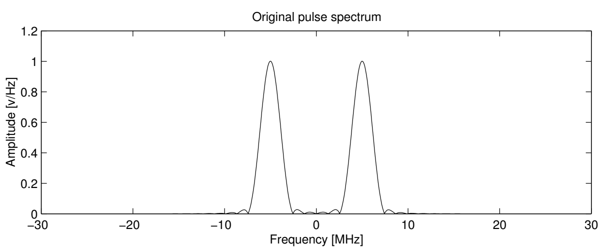


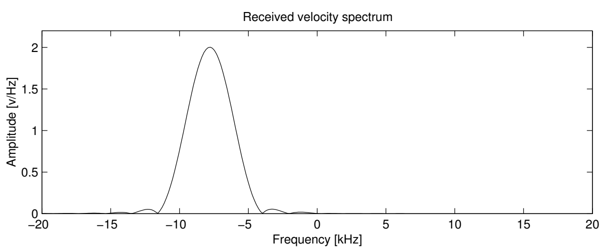
RF signal for vessel with parabolic flow

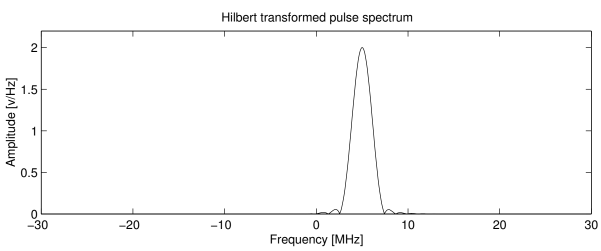


x-direction: Time between pulse emissions,
y-direction: Time since pulse emission

Hilbert transformation



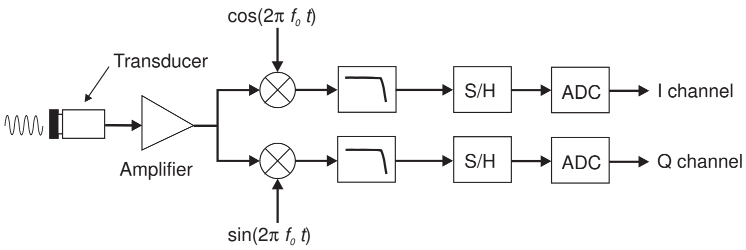




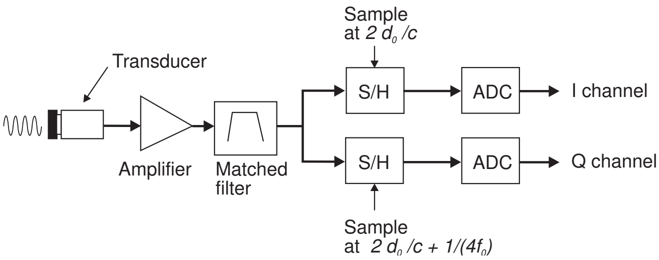
One sided spectrum created by Hilbert transforming the received signal. Thereby the sign of the frequency and velocity can be detected.

Pulsed wave systems using complex signals

Conventional analog demodulation

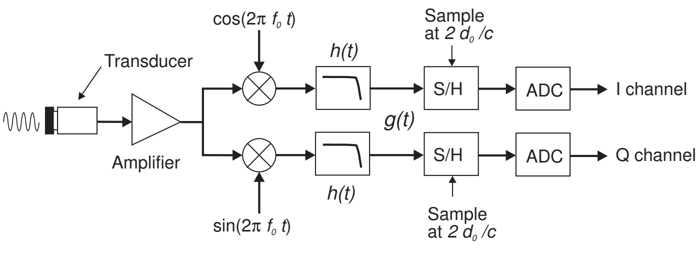


RF quadrature sampling



Pulsed wave system

Conventional analog demodulation



Demodulated signal:

$$g(t) = r(t) \cdot e^{j2\pi f_0 t} * h(t)$$

t - time since pulse emission

$$g(t) = \int_{-\infty}^{+\infty} h(\theta) r(t-\theta) e^{j2\pi f_0 (t-\theta)} d\theta = e^{j2\pi f_0 t} \int_{-\infty}^{+\infty} r(t-\theta)[e^{-j2\pi f_0 \theta} h(\theta)] d\theta$$

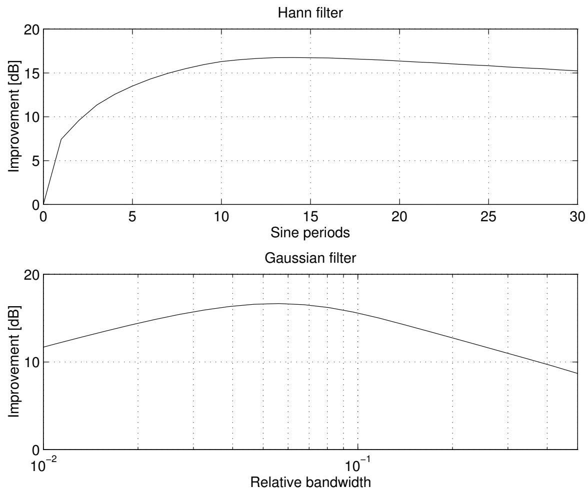
$e^{-j2\pi f_0 t} \cdot h(t)$ - Matched filter

$e^{j2\pi f_0 t}$ - Complex amplitude factor

Sampling operation: $t = t_x = \frac{2d_0}{c}$

If $t_x = \frac{K}{f_0}$ we get: $e^{j2\pi f_0 \frac{K}{f_0}} = 1$, (Note also $|e^{j2\pi f_0 \frac{K}{f_0}}| = 1$)

Effect of matched filter



Improvement in instantaneous signal power to mean noise power

Spectrum for a velocity distribution

Typical velocity distributions:

$$v(r) = v_0 \left[1 - \left(\frac{r}{R}\right)^{p_o}\right]$$

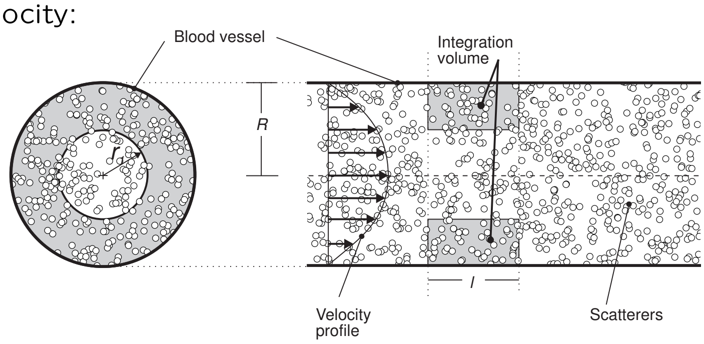
Parabolic flow for $p_o = 2$

Plug flow for $p_o \to \infty$

Frequencies received:

$$f_d(r) = \frac{2 v_0 f_0}{c} \left[1 - \left(\frac{r}{R}\right)^{p_o}\right] \cos(\theta)$$

Power density spectrum is found by calculating the number of scatterers that move at a particular velocity:



Distribution of scatterers in a tube

r radial position

v_0 maximum velocity found at center of vessel

R radius of vessel

Power density spectrum from scatterer distribution

Normalized power density:

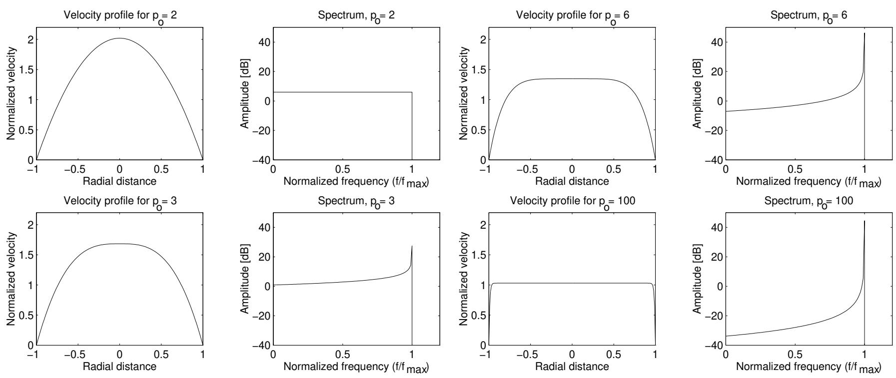
$$
\begin{aligned}
G(f_d) &= \begin{cases} \dfrac{2}{p_o \cdot f_{max}\left(1-\frac{f_d}{f_{max}}\right)^{1-\frac{2}{p_o}}} & \text{for } 0 < f_d < f_{max} \\ 0 & \text{else} \end{cases} \\
f_{max} &= \frac{2v_0 f_0}{c}\cos(\theta).
\end{aligned}
$$

Parabolic profile ($p_0 = 2$):

$$
G(f_d) = \frac{1}{f_{max}} \quad \text{for } 0 < f_d < f_{max}
$$

r	radial position	R	radius of vessel
v_0	maximum velocity found at center of vessel	ρ_p	particle density
v_1	velocity at the radial position r_1	f_0	ultrasound frequency
$\cos(\theta)$	angle between ultrasound beam and flow		

Examples of flow profiles and corresponding power density spectra

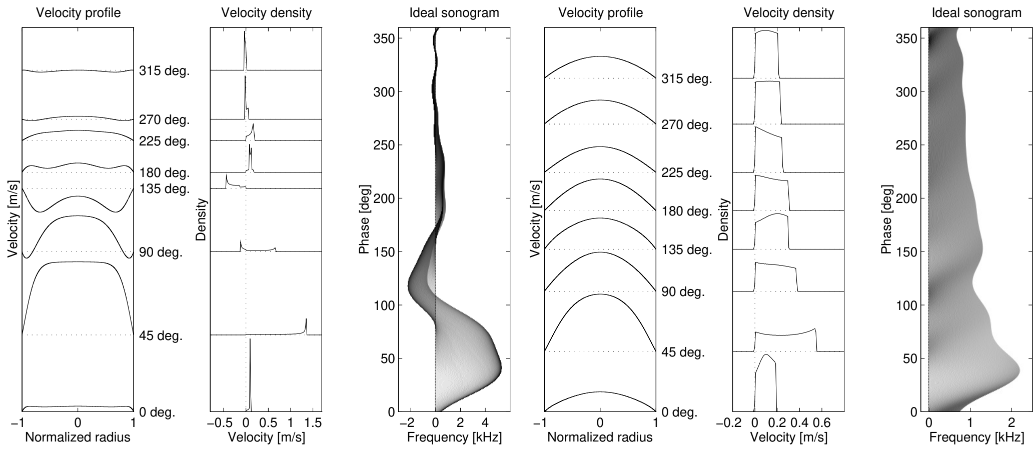


Idealized velocity profiles and corresponding normalized power density spectra.

Parabolic flow ($p_o = 2$) gives rectangular distribution of velocities and flat spectrum

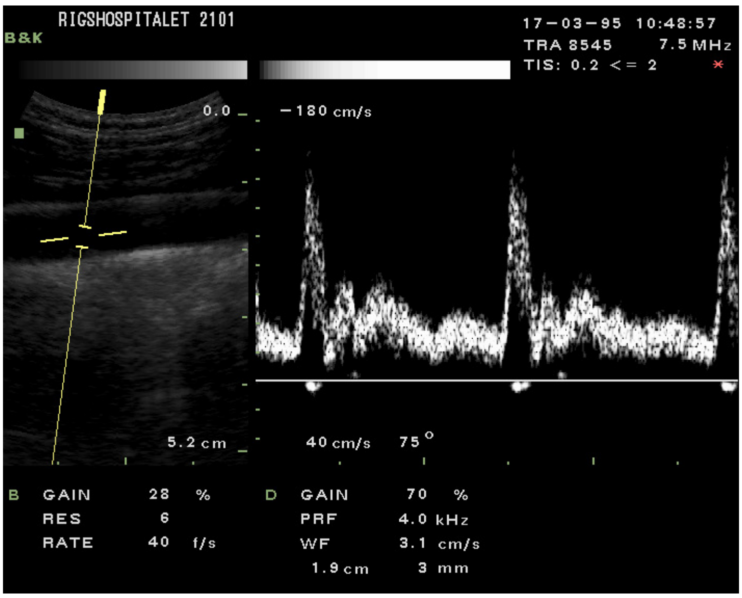
Plug flow ($p_o \to \infty$) the spectrum approaches a monochromatic shape, because nearly all scatterers are moving at the same velocity.

Examples of flow profiles and corresponding power density spectra for flow in carotis and femoralis

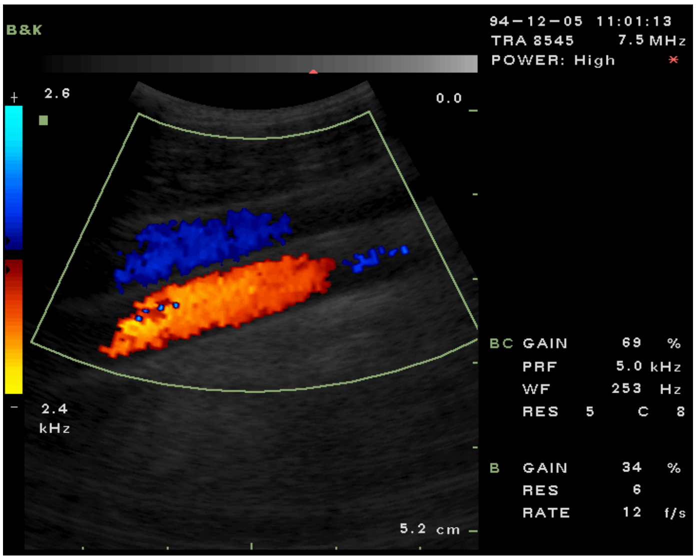


Series of velocity profiles for a common femoral artery (left) and common carotid artery (right) together with corresponding velocity densities and ideal sonograms. All curves are shown relative to the phase in the cardiac cycle.

Spectrogram from carotid artery



Color flow map



Blood supply to and from the brain (Carotid artery and jugular vein)

Color flow mapping using phase shift estimation

Received demodulated signal:

$$\begin{aligned} r_{cfm}(i) &= a \cdot \exp(-j(2\pi \frac{2v_z}{c} f_0 i T_{prf} + \phi_f)) \\ &= a \cdot \exp(-j\phi(t)) = x(i) + jy(i) \end{aligned}$$

Velocity estimation:

$$\frac{d\phi}{dt} = \frac{d\left(-2\pi \frac{2v_z}{c} f_0 t + \phi\right)}{dt} = -2\pi \frac{2v_z}{c} f_0$$

Find the change is phase as a function of time gives quantity proportional to the velocity.

Realization

$$\begin{aligned}\tan(\Delta\phi) &= \tan\left(\arctan\left(\frac{y(i)}{x(i)}\right) - \arctan\left(\frac{y(i-1)}{x(i-1)}\right)\right) \\ &= \frac{\frac{y(i)}{x(i)} - \frac{y(i-1)}{x(i-1)}}{1 + \frac{y(i)}{x(i)} \cdot \frac{y(i-1)}{x(i-1)}} \\ &= \frac{y(i) \cdot x(i-1) - y(i-1)x(i)}{x(i)x(i-1) + y(i)y(i-1)}\end{aligned}$$

using that

$$\tan(A-B) = \frac{\tan(A) - \tan(B)}{1 + \tan(A)\tan(B)}.$$

Then

$$\arctan\left(\frac{y(i)x(i-1) - y(i-1)x(i)}{x(i)x(i-1) + y(i)y(i-1)}\right) = -2\pi f_0 \frac{2v_z}{c} T_{prf}.$$

Color flow mapping using phase shift estimation

Using the complex autocorrelation:

$$R(m) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{i=-N}^{N} r^*_{cfm}(i) r_{cfm}(i+m),$$

Actual determination from the complex autocorrelation:

$$v_z = -\frac{cf_{prf}}{4\pi f_0} \arctan\left(\frac{\sum_{i=0}^{N_c-2} y(i+1)x(i) - x(i+1)y(i)}{\sum_{i=0}^{N_c-2} x(i+1)x(i) + y(i+1)y(i)}\right) = -\frac{cf_{prf}}{4\pi f_0} \arctan\left(\frac{\Im\{R(1)\}}{\Re\{R(1)\}}\right)$$

Corresponds to the mean angular frequency:

$$\bar{\omega} = \frac{\int_{-\infty}^{+\infty} \omega P(\omega) d\omega}{\int_{-\infty}^{+\infty} P(\omega) d\omega}$$

$P(\omega)$ is the power density spectrum of the received, demodulated signal.

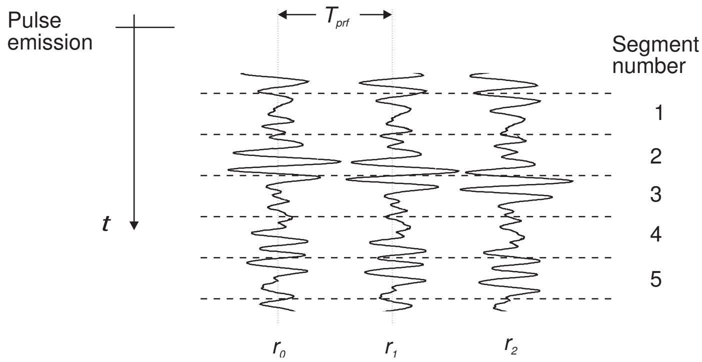
Phase shift estimation with RF sample averaging

Averaging of RF samples:

$$v_z = -\frac{cf_{prf}}{4\pi f_0} \arctan\left(\frac{\sum_{n=0}^{N_s-1}\sum_{i=0}^{N_c-2} y(n,i+1)x(n,i) - x(n,i+1)y(n,i)}{\sum_{n=0}^{N_s-1}\sum_{i=0}^{N_c-2} x(n,i+1)x(n,i) + y(n,i+1)y(n,i)} \right)$$

$x(n,i)$	RF sample for time index n and emission number i (in-phase component)
$y(n,i)$	Quadrature component
f_{prf}	Pulse repetition frequency
f_0	Center frequency of transducer
N_s	Number of samples for one pulse length
N_c	Number of emissions
c	Speed of sound

Color flow mapping using time shift estimation



Time shift between signals:

$$t_s = \frac{2\Delta z}{c} = \frac{2|\vec{v}|\cos(\theta)}{c} T_{prf}.$$

Signal relation between received signals

$$r_{s2}(t) = r_{s1}(t - t_s)$$

Cross-correlation yields

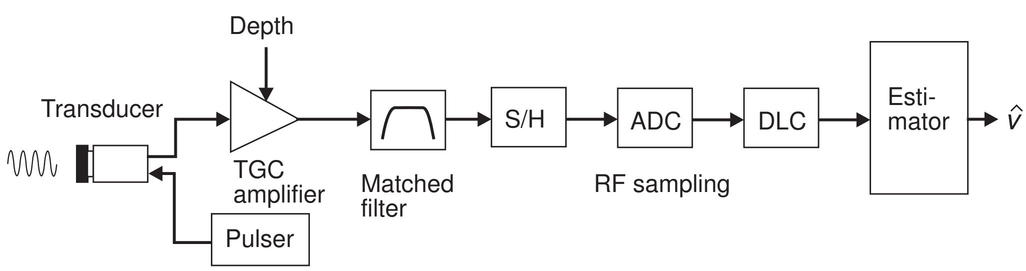
$$\begin{aligned} R_{12}(\tau) &= \frac{1}{2T}\int_T r_{s1}(t) r_{s2}(t+\tau) dt = \frac{1}{2T}\int_T r_{s1}(t) r_{s1}(t - t_s + \tau) dt \\ &= R_{11}(\tau - t_s) \\ R_{12}(\tau) &= R_{pp}(\tau) * \sigma_s^2 \delta(\tau - t_s) = \sigma_s^2 R_{pp}(\tau - t_s) \end{aligned}$$

Global maximum at $\tau - t_s = 0$.

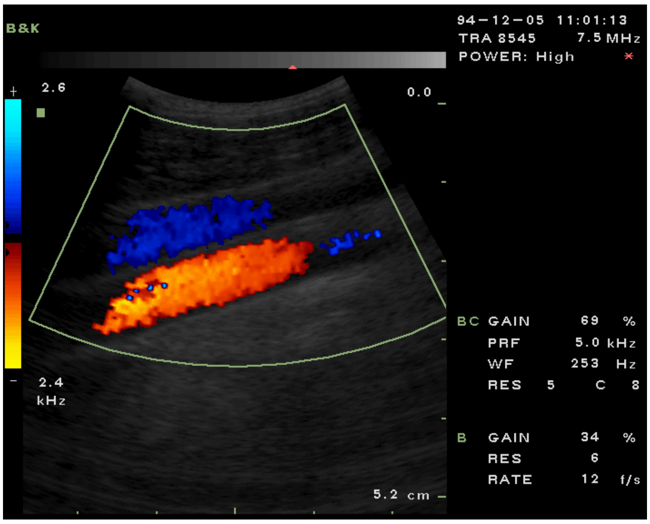
Velocity estimate is:

$$\hat{v}_z = \frac{c}{2}\frac{\hat{t}_s}{T_{prf}}.$$

Cross-correlation system



Color flow image



8 to 16 pulse emissions in each direction
Low accuracy for few emissions
Low frame rate for many emissions or image lines

Calculation of the cross-correlation

$$\hat{R}_{12d}(n, i_{seg}) = \frac{1}{N_s(N_c - 1)} \sum_{i=0}^{N_c-2} \sum_{k=0}^{N_s-1} r_{s_i}(k + i_{seg}N_s) r_{s_{i+1}}(k + i_{seg}N_s + n).$$

Largest detectable velocity:

$$v_{max} = \frac{l_g}{T_{prf}} = \frac{c}{2} N_s \frac{f_{prf}}{f_s}.$$

Minimum velocity due to time quantization:

$$v_{min} = \frac{c}{2} \frac{f_{prf}}{f_s}$$

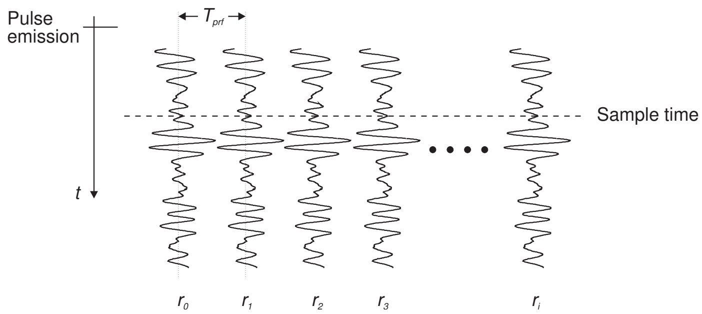
Interpolated peak by polynomial fit

$$n_{int} = n_m - \frac{\hat{R}_{12d}(n_m+1) - \hat{R}_{12d}(n_m-1)}{2(\hat{R}_{12d}(n_m+1) - 2\hat{R}_{12d}(n_m) + \hat{R}_{12d}(n_m-1))}$$

Interpolated estimate:

$$\hat{v}_{int} = \frac{c}{2}\frac{n_{int} f_{prf}}{f_s}.$$

Stationary echo canceling



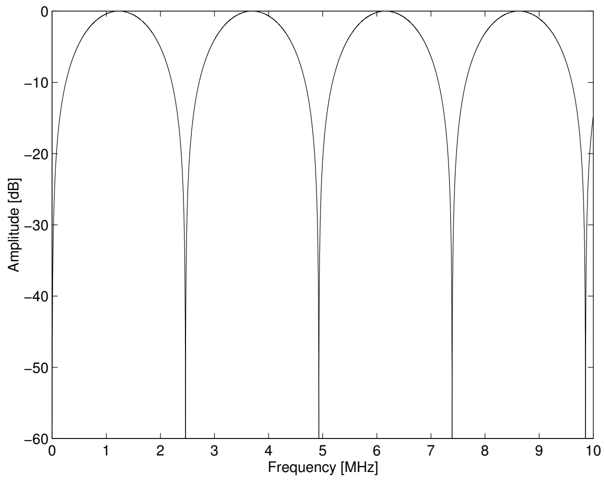
Canceling:

$$\begin{aligned} r_c(i) &= \frac{1}{2}(r(i-1) - r(i)) \\ y_i(t) &= y_{i-1}(t - t_s), \quad t_s = \frac{2v_z}{c} T_{prf} \\ y_c(t) &= \frac{1}{2}(y_{i-1}(t) - y_{i-1}(t-t_s)) \leftrightarrow Y_c(f) = \frac{1}{2} Y_{i-1}(f)(1 - e^{-j2\pi f t_s}) \end{aligned}$$

Transfer function of filter: $H(f) = \frac{1}{2}(1 - e^{-j2\pi f t_s})$

Transfer function of filter

$$H(f) = \frac{1}{2}(1 - e^{-j2\pi f t_s})$$



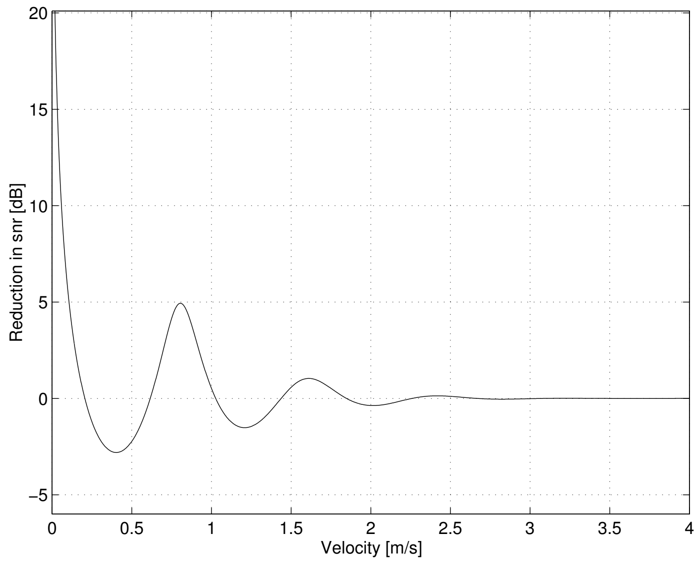
v=1 m/s, $f_{prf} = 3.2$ kHz.

Reduction in signal-to-noise ratio due to filter

$$R_{\text{snr}} = \frac{\text{snr}}{\text{snr}_f} = \frac{\sqrt{\dfrac{E[\{p(t) * s_c(t)\}^2]}{E[n^2(t)]}}}{\dfrac{1}{\sqrt{2}}\sqrt{\dfrac{E[\{p(t) * h(t;t_s) * s_c(t)\}^2]}{E[n^2(t)]}}}$$
$$= \sqrt{2}\sqrt{\frac{E[\{p(t) * s_c(t)\}^2]}{E[\{p(t) * h(t;t_s) * s_c(t)\}^2]}} = \sqrt{2}\sqrt{\frac{R_p(0)}{R_p(\tau) * R_h(\tau, t_s)|_{\tau=0}}}.$$

$p(t)$ - Pulse, $n(t)$ - Measurement noise, $s_c(t)$ - Scattering signal
$h(t;t_s)$ - Impulse response of filter, $R_p(\tau)$ - Autocorrelation of pulse,
$R_h(\tau, t_s)$ - Autocorrelation of filter

Reduction in signal-to-noise ratio due to stationary echo canceling

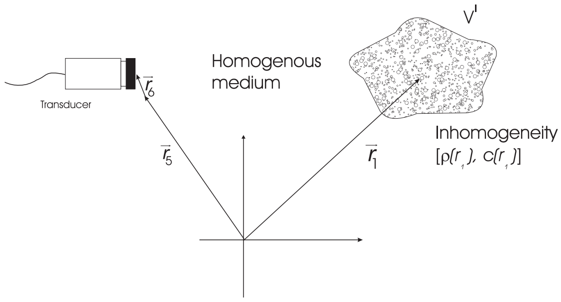


Gaussian 3 MHz pulse with relative bandwidth of 0.2, $f_{prf} = 3.2$ kHz.
Large reduction at low velocities.

Problems in current color flow imaging systems

- Frame rate is linked to number of lines in image
- More emissions gives lower frame rate
- Few emissions gives high standard deviation
- Only velocity along the ultrasound beam is found
- Stationary echo canceling is difficult due to few samples
- Slow moving flow is difficult to detect
- All these issues will be addressed in the lecture on SA flow imaging

Pulse Echo Ultrasound fields



Pulse echo field:

$$\begin{aligned} v_r(\vec{r}_1, t) &= v_{pe}(t) * f_m(\vec{r}_1) * h_{pe}(\vec{r}_1, t) \\ &= v_{pe}(t) * f_m(\vec{r}_1) * h_t(\vec{r}_1, t) * h_r(\vec{r}_1, t) \\ f_m(\vec{r}_1) &= \frac{\Delta\rho(\vec{r}_1)}{\rho_0} - \frac{2\Delta c(\vec{r}_1)}{c} \end{aligned}$$

Change in density: $\Delta\rho(\vec{r}_1)$, change in speed of sound: $\Delta c(\vec{r}_1)$

Simulation in Field II

Pulse-echo model:

$$p_r(\vec{r}_1, \vec{r}_2, t) = v_{pe}(t) \underset{t}{\star} f_m(\vec{r}_1) \underset{r}{\star} h_{pe}(\vec{r}_1, \vec{r}_2, t),$$

Neglect the Doppler effect and use:

$$\vec{r}_2(i+1) = \vec{r}_2(i) + T_{prf}\vec{v}(\vec{r}_2(i), t)$$

Scatterers are propagated between pulses according to their velocity

Field II: An Example: Flow simulation

```
%  Example of using Field II by Joergen Arendt Jensen, May 26, 2015

%  Start the system and initialize the path

path(path,'/home/jaj/programs/field_II/M_files')
path(path,'/home/jaj/programs/field_II/m_utilities');

%  Initialize the field system

field_init

%  Generate the transducer apertures for send and receive

f0=5e6;                  %  Transducer center frequency [Hz]
M=4;                     %  Number of cycles in emitted pulse
fs=100e6;                %  Sampling frequency [Hz]
c=1540;                  %  Speed of sound [m/s]
lambda=c/f0;             %  Wavelength [m]
pitch=lambda/2;          %  Pitch of transducer
element_height=5/1000;   %  Height of element [m]
kerf=width/10;           %  Kerf [m]
width=pitch-kerf;        %  Width of element
focus=[0 0 60]/1000;     %  Fixed focal point [m]
```

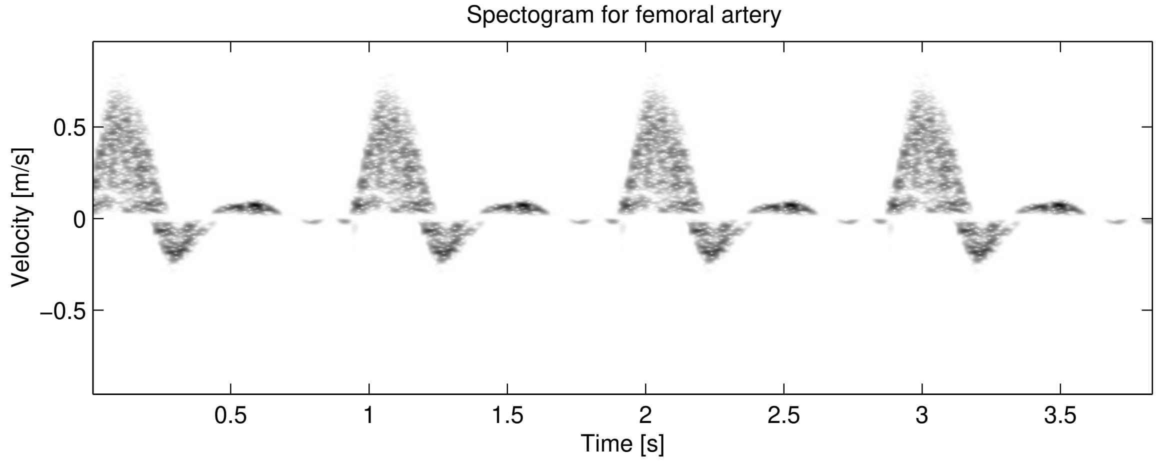


```
N_elements=64;           %  Number of physical elements
Nshoots=5000;            %  Number of shots to be processed

%  Set the sampling frequency

set_sampling(fs);

%  Generate aperture for emission

emit_aperture = xdc_linear_array (N_elements, width, element_height, kerf, 1, 1,focus);

%  Set the impulse response and excitation of the emit aperture

impulse_response=sin(2*pi*f0*(0:1/fs:2/f0));
impulse_response=impulse_response.*hanning(max(size(impulse_response)))';
xdc_impulse (emit_aperture, impulse_response);

excitation=sin(2*pi*f0*(0:1/fs:M/f0));
xdc_excitation (emit_aperture, excitation);

%  Generate aperture for reception

receive_aperture = xdc_linear_array (N_elements, width, element_height, kerf, 1, 1,focus)

%  Set the impulse response for the receive aperture
```

```
xdc_impulse (receive_aperture, impulse_response);

%   Set a Hanning apodization on the apertures

apo=hanning(N_elements)';
xdc_apodization (emit_aperture, 0, apo);
xdc_apodization (receive_aperture, 0, apo);

%  Make the flow simulation

for i=1:Nshoots

  %  Generate the rotated and offset block of sample

  theta=45/180*pi;
  xnew=x*cos(theta)+z*sin(theta);
  znew=z*cos(theta)-x*sin(theta) + z_offset;
  scatterers=[xnew; y; znew;]' ;

  %   Calculate the received response

  [v, t1]=calc_scat(emit_aperture, receive_aperture, scatterers, amp');

  %  Store the result
```

```
  image_data(1:max(size(v)),i)=v';
  times(i) = t1;

  %  Propagate the scatterers and aliaze them
  %  to lie within the correct range

  x1=x;
  x=x + velocity*Tprf;
  outside_range= (x > x_range/2);
  x=x - x_range*outside_range;
  end
```

Spectogram for femoral artery based on Womerlsy-Evans' model



Field II: CFM flow simulation

```
%  This example shows how a linear array B-mode system scans an image
%  when doing color flow mapping
%
%  This script assumes that the field_init procedure has been called
%  Here the field simulation is performed and the data is stored
%  in rf-files; one for each rf-line done. The data must then
%  subsequently be processed to yield the image. The data for the
%  scatteres are read from the file pht_data.mat, so that the procedure
%  can be started again or run for a number of workstations.
%
%  Version 2.2 by Joergen Arendt Jensen, May 27, 2015

%  Generate the transducer apertures for send and receive

f0=5e6;                 %  Transducer center frequency [Hz]
fs=100e6;               %  Sampling frequency [Hz]
c=1540;                 %  Speed of sound [m/s]
lambda=c/f0;            %  Wavelength [m]
width=lambda;           %  Width of element
element_height=5/1000;  %  Height of element [m]
kerf=0.05/1000;         %  Kerf [m]
focus=[0 0 70]/1000;    %  Fixed focal point [m]
N_elements=196;         %  Number of physical elements
```

```
rec_N_active=64;         %  Number of active elements in receive
xmit_N_active=64;        %  Number of active elements in transmit

%  Generate aperture for emission

xmit_aperture = xdc_linear_array (N_elements, width, ...
                   element_height, kerf, 1, 10,focus);

%  Set the impulse response and excitation of the xmit aperture

impulse_response=sin(2*pi*f0*(0:1/fs:2/f0));
impulse_response=impulse_response.*hanning(max(size(impulse_response)))';
xdc_impulse (xmit_aperture, impulse_response);

excitation=sin(2*pi*f0*(0:1/fs:2/f0));
xdc_excitation (xmit_aperture, excitation);

%  Generate aperture for reception

receive_aperture = xdc_linear_array (N_elements, width, ...
                       element_height, kerf, 1, 10,focus);

%  Set the impulse response for the receive aperture

xdc_impulse (receive_aperture, impulse_response);
```

```
%  Do for the number of CFM lines

Ncfm=10;
for k=1:Ncfm

%   Load the computer phantom

  cmd=['load sim_flow/scat_',num2str(k),'.mat']
  eval(cmd)

  %   Do linear array imaging

  no_lines=20;                    %  Number of lines in image
  image_width=40/1000;            %  Size of image sector
  d_x=image_width/(no_lines-1);   %  Increment for image

  %  Set the different focal zones for reception

  rec_zone_start=30/1000;
  rec_zone_stop=100/1000;
  rec_zone_size=10/1000;

  focal_zones_center=[rec_zone_start:rec_zone_size:rec_zone_stop]';
  focal_zones=focal_zones_center-0.5*rec_zone_size;
```

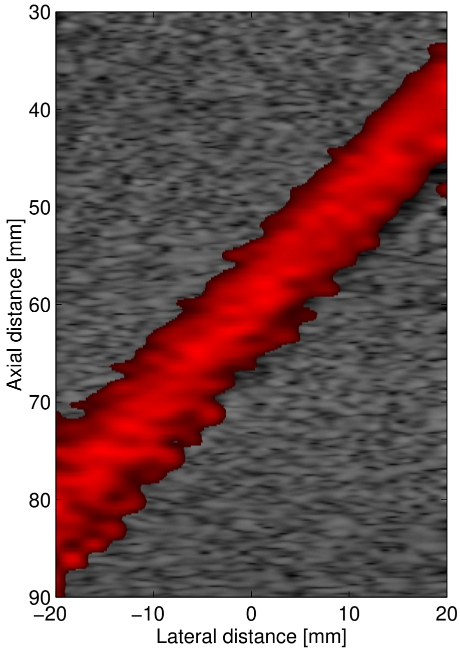
```
Nf=max(size(focal_zones));
focus_times=focal_zones/1540;

%  Set a Hanning apodization on the receive aperture
%  Dynamic opening aperture is used.

Fnumber=2.0;
rec_N_active_dyn=round(focal_zones_center./(Fnumber*(width+kerf)));

for ii=1:Nf
  if rec_N_active_dyn(ii)>rec_N_active
    rec_N_active_dyn(ii)=rec_N_active;
    end
  rec_N_pre_dyn(ii) = ceil(rec_N_active/2  - rec_N_active_dyn(ii)/2);
  rec_N_post_dyn(ii) = rec_N_active - rec_N_pre_dyn(ii) - ...
                             rec_N_active_dyn(ii);
  rec_apo=(ones(1,rec_N_active_dyn(ii)));
  rec_apo_matrix_sub(ii,:)=[zeros(1,rec_N_pre_dyn(ii)) rec_apo ...
                            zeros(1,rec_N_post_dyn(ii))];
  end

%  Transmit focus

z_focus=40/1000;
```

```
%   Set a Hanning apodization on the xmit aperture

xmit_apo=hanning(xmit_N_active)';

% Do imaging line by line

i_start=1;
x= -image_width/2 +(i_start-1)*d_x;

for i=i_start:no_lines
i
  %   Set the focus for this direction

  xdc_center_focus (emit_aperture, [x 0 0]);
  xdc_focus (xmit_aperture, 0, [x 0 z_focus]);
  xdc_center_focus (receive_aperture, [x 0 0]);
  xdc_focus (receive_aperture, focus_times,
              [x*ones(Nf,1), zeros(Nf,1), focal_zones]);

  %  Calculate the apodization

  xmit_N_pre  = round(x/(width+kerf) + N_elements/2 - xmit_N_active/2);
  xmit_N_post = N_elements - xmit_N_pre - xmit_N_active;
  xmit_apo_vector=[zeros(1,xmit_N_pre) xmit_apo zeros(1,xmit_N_post)];
```

```
    rec_N_pre(i) = round(x/(width+kerf) + N_elements/2 - rec_N_active/2);
    rec_N_post(i) = N_elements - rec_N_pre(i) - rec_N_active;

    rec_apo_matrix=[zeros(size(focus_times,1),rec_N_pre(i)) ...
                    rec_apo_matrix_sub zeros(size(focus_times,1), ...
    rec_N_post(i))];

    xdc_apodization (xmit_aperture, 0, xmit_apo_vector);
    xdc_apodization (receive_aperture, focus_times , rec_apo_matrix);

    %   Calculate the received response

    [rf_data, tstart]=calc_scat(xmit_aperture, receive_aperture, ...
                                positions, amp);

    %  Store the result

    cmd=['save sim_flow/rft',num2str(k),'l',num2str(i), ...
                      '.mat rf_data tstart']
    eval(cmd)

    %  Steer in another direction

    x = x + d_x;
```

```
    end  %  Loop for lines

  end  %  CFM loop

%   Free space for apertures

xdc_free (xmit_aperture)
xdc_free (receive_aperture)
```

Color flow imaging phantom



Exercise on ultrasound RF flow data

Basic model, first emission:

$$r_1(t) = p(t) * s(t)$$

$s(t)$ - Scatterer amplitudes (white, random, Gaussian)

Second emission:

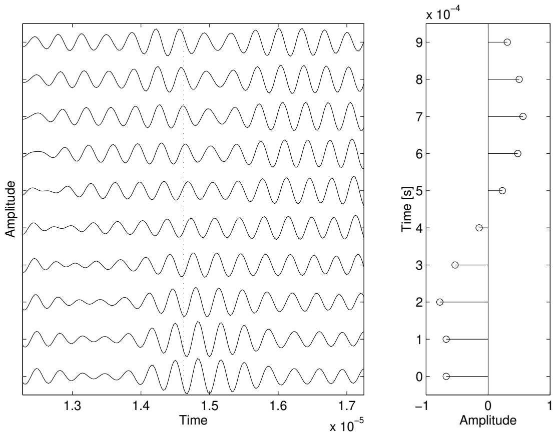
$$r_2(t) = p(t) * s(t - t_s) = r_1(t - t_s)$$

Time shift t_s:

$$t_s = \frac{2v_z}{c} T_{prf}$$

$r_1(t)$	Received voltage signal	$p(t)$	Ultrasound pulse
$*$	Convolution	v_z	Axial blood velocity
c	Speed of sound	T_{prf}	Time between pulse emissions

A simple interpretation - a collection of scatterers



Signal from a collection of scatterers crossing a beam from a concave transducer.

Collection of scatterers:

$$
\begin{aligned}
r_s(i) &= -\sum_{k=1}^{N} a_k \sin(2\pi \frac{2v_z(k)}{c} f_0 T_{prf} i - \phi_k) \\
\phi_k &= 2\pi f_0 \left(t_z - \frac{2d_k}{c} \right)
\end{aligned}
$$

k - Scatterer number

For a plug flow:

$$
\begin{aligned}
y_i(t) &= p(t) * e(t - it_s) = y_0(t - it_s) \\
t_s &= \frac{2v_z}{c} T_{prf}
\end{aligned}
$$

For a sampled system:

$$
\begin{aligned}
y_i(n) &= p(n) * e(n - i \cdot n_s) = y_0(n - i \cdot n_s) \\
n_s &= \frac{2v_z}{c} T_{prf} f_s
\end{aligned}
$$

Signal processing

1. Find ultrasound pulse (load from file)
2. Make scatterers
3. Generate a number of received RF signals
4. Study the generated signals
5. Compare with simulated and measured RF data
6. Make a function for velocity estimation using cross-correlation
7. Validate it on the simulated data and apply it to the femoral data

Signal relation between received signals

$$r_{s2}(t) = r_{s1}(t - t_s)$$

Cross-correlation yields

$$\begin{aligned} R_{12}(\tau) &= \frac{1}{2T}\int_T r_{s1}(t) r_{s2}(t+\tau) dt = \frac{1}{2T}\int_T r_{s1}(t) r_{s1}(t - t_s + \tau) dt \\ &= R_{11}(\tau - t_s) \\ R_{12}(\tau) &= R_{pp}(\tau) * \sigma_s^2 \delta(\tau - t_s) = \sigma_s^2 R_{pp}(\tau - t_s) \end{aligned}$$

Global maximum at $\tau - t_s = 0$.

Velocity estimate is:

$$\widehat{v}_z = \frac{c}{2}\frac{\widehat{t}_s}{T_{prf}}.$$

Calculation of the cross-correlation

$$\widehat{R}_{12d}(n, i_{seg}) = \frac{1}{N_s(N_c - 1)} \sum_{i=0}^{N_c-2} \sum_{k=0}^{N_s-1} r_{s_i}(k + i_{seg}N_s) r_{s_{i+1}}(k + i_{seg}N_s + n).$$

Interpolated peak by polynomial fit

$$n_{int} = n_m - \frac{\widehat{R}_{12d}(n_m+1) - \widehat{R}_{12d}(n_m-1)}{2(\widehat{R}_{12d}(n_m+1) - 2\widehat{R}_{12d}(n_m) + \widehat{R}_{12d}(n_m-1))}$$

Interpolated estimate:

$$\widehat{v}_{int} = \frac{c}{2}\frac{n_{int} f_{prf}}{f_s}.$$