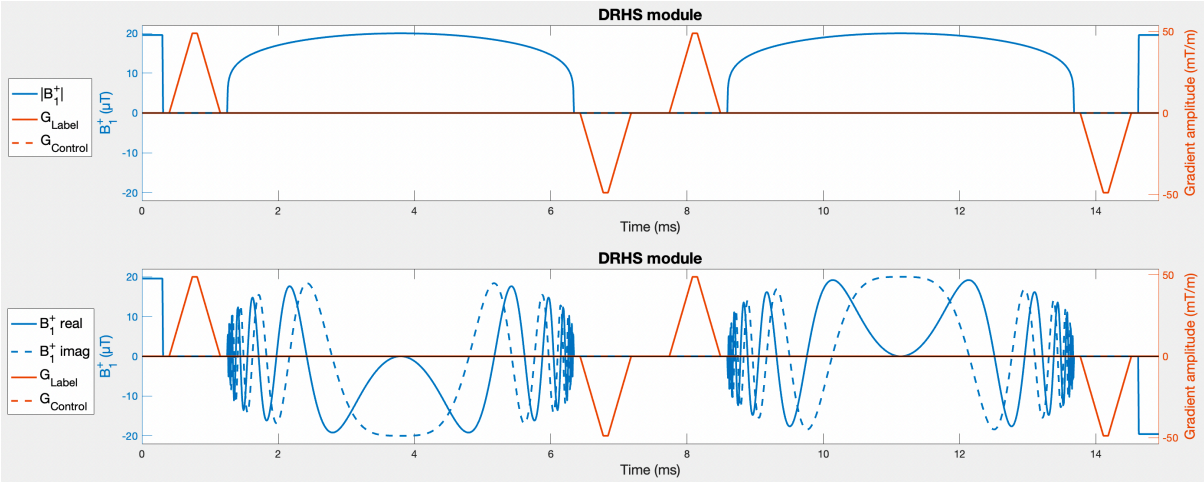
Velocity-Selective ASL: Bloch simulations

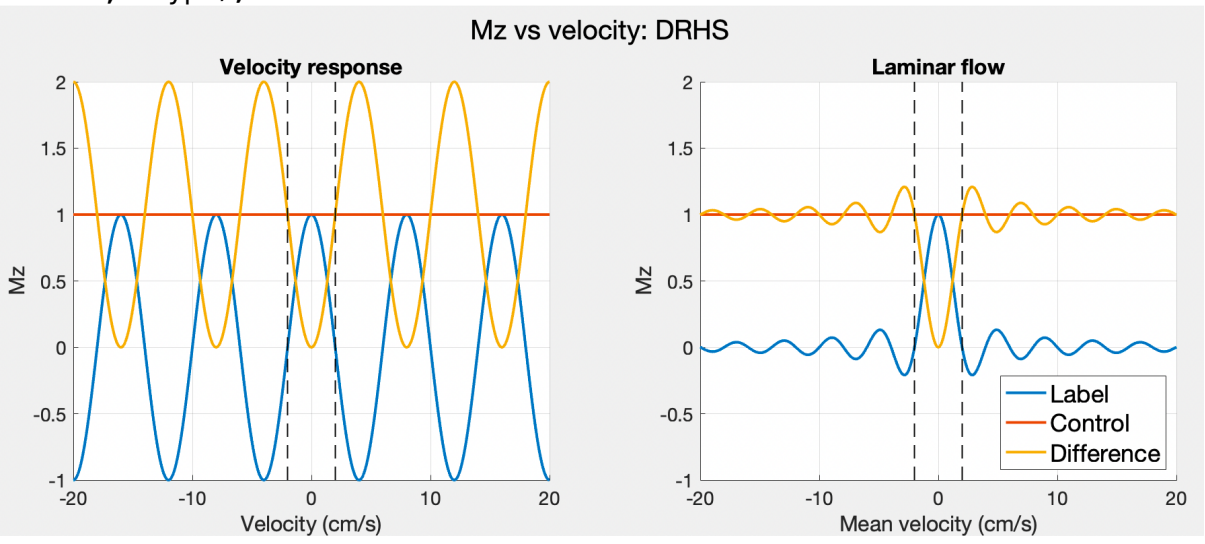
Walk through and example plots

```
%% Bloch simimulations: velocity-selective ASL modules
% Written by Joseph G. Woods and Dapeng Liu

% Add containing folder to MATLAB path and cd to it
filePath = fileparts(matlab.desktop.editor.getActiveFilename);
addpath(genpath(filePath));
cd(filePath)

% If you cannot use the pre-compiled bloch_Hz MEX functions, please try
% the following:
% 1. cd Bloch
% 2. mex -setup C
% 3. mex bloch_Hz.c -compatibleArrayDims
% 4. cd ..

%% General settings
Vcut    = 2;              % velocity cutoff (cm/s)
B1max   = 20 *1e-6;       % max B1+ amplitude      (μT    -> T)
Gmax    = 50 *1e-3*1e-2; % max gradient amplitude (mT/m  -> T/cm)
SRmax   = 150*1e-2;       % max gradient slew rate (T/m/s -> T/cm/s)
pad1    = 0.1;            % padding time before gradients (ms)
pad2    = 0.1;            % padding time after gradients (ms)
RFUP    = 10;             % RF update time (μs)
GUP     = 10;             % gradient update time (μs)
units   = 'T';            % Tesla
bplotVS = true;           % plot module
T1      = inf;            % longitudinal relaxation time (s)
T2      = inf;            % transverse relaxation time (s)
```

```
%% Generate the VSASL B1 and gradients

bvelCompCont = false; % velocity-compensated control (not for BIR-4!) -
                        false is zero-amplitude gradients in control
bcomposite   = true;  % composite refocussing pulses (FT-VSI only) - false
                        is hard refocussing pulses
bsinc        = false; % sinc modulation of excitation pulses (FT-VSI only)
                        - false is rectangular modulation
vsType = 'DRHS';

% Generate the B1 and gradients
[B1, GLabel, GCont, T] = genVSASL(vsType, Vcut, B1max, Gmax, SRmax, pad1, pad2,
RFUP, GUP, units, bplotVS, bvelCompCont, bcomposite, bsinc);

% Convert to Hz and Hz/cm for Bloch simulations
B1_Hz        = B1     * T.gam;
GLabel_Hzcm  = GLabel * T.gam;
GCont_Hzcm   = GCont  * T.gam;
```

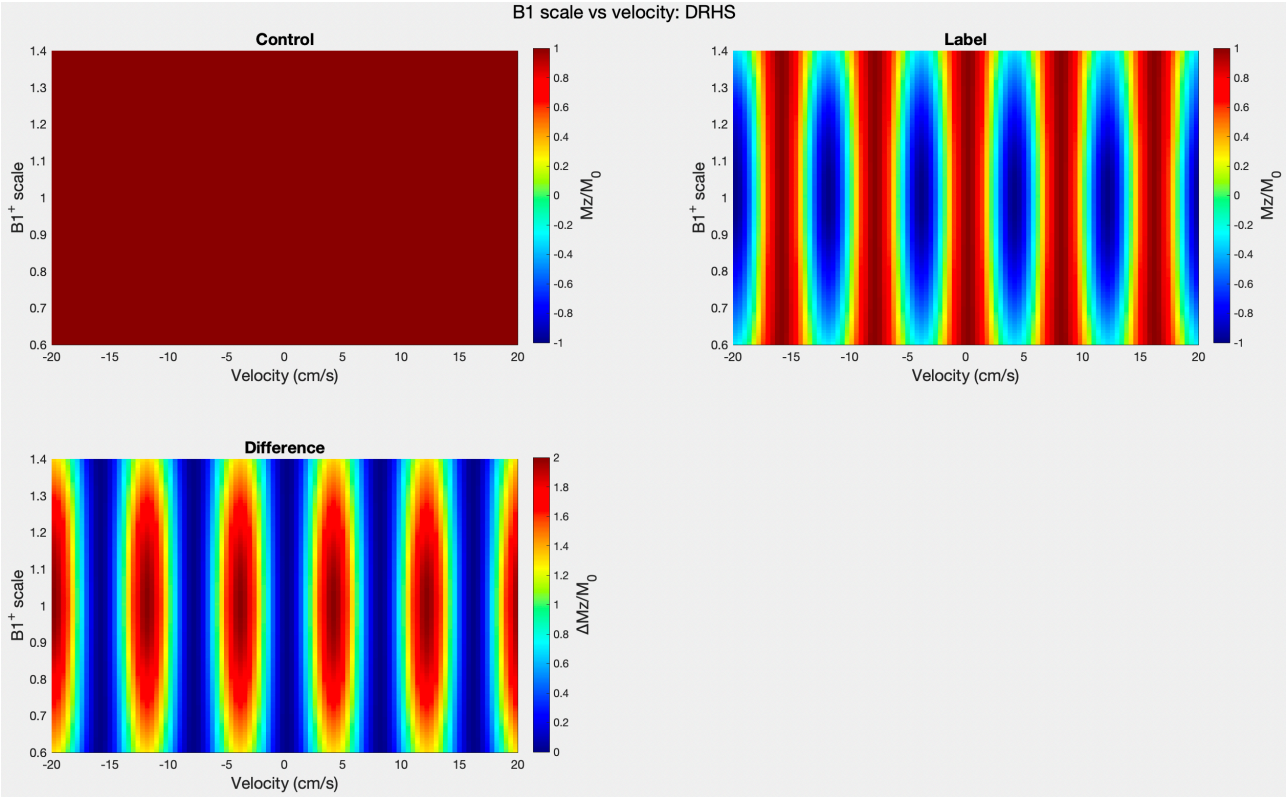


```
%% Mz vs velocity (velocity profile)

% Simulates label and control module for a range of velocities. Laminar
% flow integration is then performed.

% The velocity cut-off is defined as Vcut = π/(γ*2*m1) for
% DHRS/DRHT/BIR-4/BIR-8, where m1 is the 1st moment of the gradients in
% the VSASL modules. This correspondes to the velocity where the
% difference signal velocity profile first reaches a value of 1 (or the
% edge of the central sinc lobe for the laminar flow case).

% While Vcut for FT-VSI cannot be calculated using the same equation, we
% can empirically calculate the necessary m1 in each FT-VSI encoding
% period so that the "1-crossing" of the velocity profile is equal to Vcut
% (other definitions of Vcut are also possible).

% Laminar flow is often assumed when illustrating the labeling process in
% VSASL. Integrating the Mz profile over the laminar velocity distribution
% (uniform from 0 to Vmax) can be numerically approximated by calculating
% the mean Mz across a discrete set of velocities, where Vmean is Vmax/2
% (see Wong et al. MRM 2006. https://doi.org/10.1002/mrm.20906)

% Tasks: 1. Run the simulation for each VSASL module. What does the
%           velocity profile look like in each case?
%        2. What happens when Vcut is changed?

B1scale = 1;                    % B1-variation (fraction)
df      = 0;                    % off-resonance (Hz)
dp      = 0;                    % position (cm)
dv      = linspace(-40,40,1001); % velocity (cm/s)

% Run Bloch simulations
[~,~,mzlabel] = bloch_Hz(B1scale*B1_Hz, GLabel_Hzcm, RFUP*1e-6, T1, T2, df, dp,
dv, 0); % Label
[~,~,mzcont]  = bloch_Hz(B1scale*B1_Hz, GCont_Hzcm , RFUP*1e-6, T1, T2, df, dp,
dv, 0); % Control

% ASL subtraction
if strcmpi(vsType,'FTVSI'); dmz = mzlabel - mzcont;
else;                       dmz = mzcont - mzlabel; end

% Laminar flow integration
[mzLabel_laminar, ~   ] = laminarFlowInt(mzlabel, dv);
[mzCont_laminar , ~   ] = laminarFlowInt(mzcont , dv);
[dmz_laminar    , lind] = laminarFlowInt(dmz    , dv);

% Plot results
plot_VSprofile(Vcut,dv,lind,mzlabel,mzcont,dmz,mzLabel_laminar,mzCont_laminar,dm
z_laminar,vsType);
```

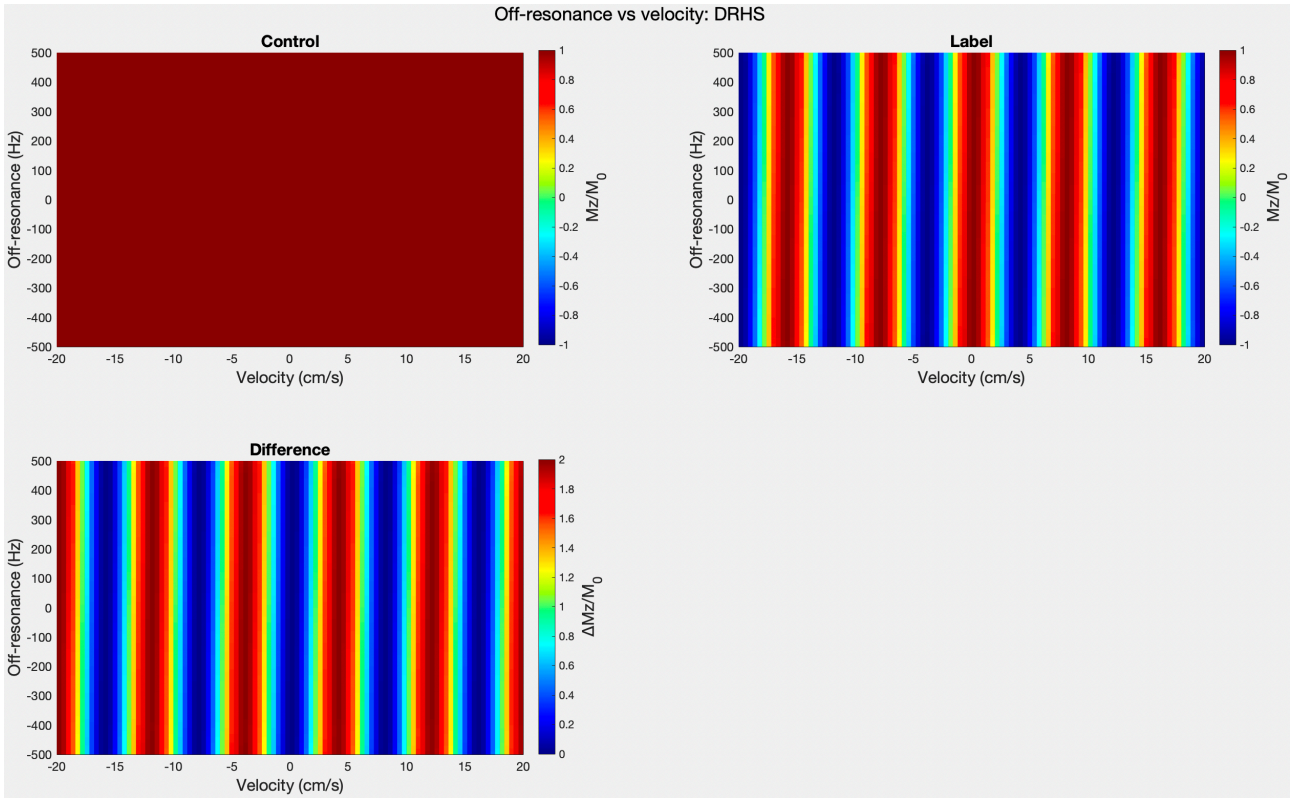


```
%% B1 vs velocity

% Simulates label and control module for a range of velocities and B1
% scaling.

% Some VSASL modules are more robust to B1-variation because they use
% adiabatic pulses for either refocussing or excitatition and refocussing.
%  - DRHS/DRHT modules use hard excitation pulses but adiabatic refocussing
%    pulses.
%  - BIR-4/BIR-8 use both adiabatic excitation and refocussing pulses.
%  - FT-VSI uses hard excitation pulses but composite refocussing pulses to
%    improve B1-robustness.

% Tasks: 1. Run the simulation for each VSASL module. Which are least/most
%           robust to B1 variation?
%        2. Are both the label or control module sensitive to B1?
%        3. For FT-VSI, try switching off the composite refocussing pulses
%           (hard refocussing pulses are used instead). What happens?

blaminar = false; % use laminar flow integration?

B1scale = linspace(0.6,1.4,101); % B1-variation (fraction)
df      = 0;                     % off-resonance (Hz)
dp      = 0;                     % position (cm)
dv      = linspace(-20,20,101);  % velocity (cm/s)

% Run Bloch simulations
mzlabel = zeros(length(B1scale), length(dv));
mzcont  = zeros(length(B1scale), length(dv));
for ii = 1:length(B1scale)
    [~,~,mzlabel(ii,:)] = bloch_Hz(B1scale(ii)*B1_Hz, GLabel_Hzcm, RFUP*1e-6,
T1, T2, df, dp, dv, 0); % Label
    [~,~,mzcont(ii,:)]  = bloch_Hz(B1scale(ii)*B1_Hz, GCont_Hzcm , RFUP*1e-6,
T1, T2, df, dp, dv, 0); % Control
end

% ASL subtraction
if strcmpi(vsType,'FTVSI'); dmz = mzlabel  - mzcont;
else;                       dmz = mzcont - mzlabel; end

if blaminar % Laminar flow integration
    mzlabeldisp = zeros(length(B1scale),ceil(length(dv)/2));
    mzcontdisp  = zeros(length(B1scale),ceil(length(dv)/2));
    dmzdisp     = zeros(length(B1scale),ceil(length(dv)/2));
    for ii = 1:length(B1scale)
        [mzlabeldisp(ii,:), ~   ] = laminarFlowInt(mzlabel(ii,:), dv);
        [mzcontdisp(ii,:) , ~   ] = laminarFlowInt(mzcont(ii,:) , dv);
        [dmzdisp(ii,:)    , lind] = laminarFlowInt(dmz(ii,:)    , dv);
    end
    dv = dv(lind);
else
    mzlabeldisp = mzlabel;
    mzcontdisp  = mzcont;
    dmzdisp     = dmz;
end

% Plot data
data1 = {dv,dv,dv};
data2 = {B1scale,B1scale,B1scale};
data3 = {mzcontdisp,mzlabeldisp,dmzdisp};
ti    = {'Control','Label','Difference'};
cbl   = {'Mz/M_0','Mz/M_0','ΔMz/M_0'};
```

```
surf_custom('data1',data1,'data2',data2,'data3',data3,...
            'name',['B1 scale vs velocity: ' vsType],...
            'xlabel',{'Velocity (cm/s)'},'ylabel',{'B1^+ scale'},'title',ti,...
            'color',{'jet'},'colorbarlabel',cbl,...
            'clim',{[-1,1],[-1,1],[0,2]});
```

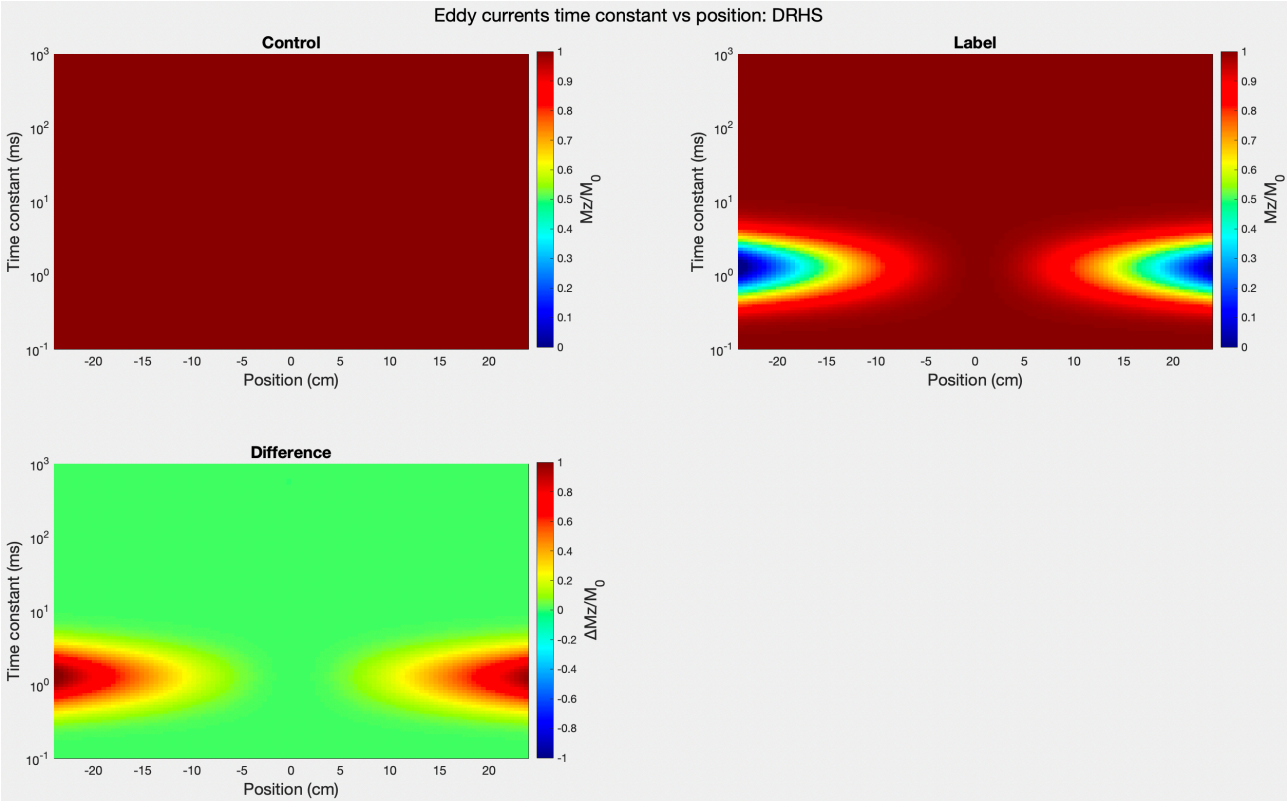


```
%% Off-resonance (B0) vs velocity

% Simulates label and control module for a range of velocities and
% off-resonances.

% All of the above modules use refocussing pulses, so should be fairly
% robust to off-resonance variation when B1scale = 1 (nominal B1).

% Tasks: 1. Run the simulation for each VSASL module to confirm they are
%           all robust to off-resonance.
%        2. Does the off-resonance robustness decrease when B1scale is
%           decreased? How does this compare for different modules?

blaminar = false; % use laminar flow integration?

B1scale = 1;                      % B1-variation (fraction)
df      = linspace(-500,500,101); % off-resonance (Hz)
dp      = 0;                      % position (cm)
dv      = linspace(-20,20,101);   % velocity (cm/s)

% Run Bloch simulations
[~,~,mzlabel] = bloch_Hz(B1scale*B1_Hz, GLabel_Hzcm, RFUP*1e-6, T1, T2, df, dp,
dv, 0); % Label
[~,~,mzcont ] = bloch_Hz(B1scale*B1_Hz, GCont_Hzcm , RFUP*1e-6, T1, T2, df, dp,
dv, 0); % Control

% ASL subtraction
if strcmpi(vsType,'FTVSI'); dmz = mzlabel - mzcont;
else;                       dmz = mzcont - mzlabel; end

if blaminar % Laminar flow integration
    mzlabeldisp = zeros(length(df),ceil(length(dv)/2));
    mzcontdisp  = zeros(length(df),ceil(length(dv)/2));
    dmzdisp     = zeros(length(df),ceil(length(dv)/2));
    for ii = 1:length(df)
        [mzlabeldisp(ii,:), ~   ] = laminarFlowInt(mzlabel(ii,:), dv);
        [mzcontdisp(ii,:) , ~   ] = laminarFlowInt(mzcont(ii,:) , dv);
        [dmzdisp(ii,:)    , lind] = laminarFlowInt(dmz(ii,:)    , dv);
    end
    dv = dv(lind);
else
    mzlabeldisp = mzlabel;
    mzcontdisp  = mzcont;
    dmzdisp     = dmz;
end

% Plot data
data1 = {dv,dv,dv};
data2 = {df,df,df};
data3 = {mzcontdisp,mzlabeldisp,dmzdisp};
ti    = {'Control','Label','Difference'};
cbl   = {'Mz/M_0','Mz/M_0','ΔMz/M_0'};

surf_custom('data1',data1,'data2',data2,'data3',data3,...
            'name',['Off-resonance vs velocity: ' vsType],...
            'xlabel',{'Velocity (cm/s)'},'ylabel',{'Off-resonance
(Hz)'},'title',ti,...
            'color',{'jet'},'colorbarlabel',cbl,...
            'clim',{[-1,1],[-1,1],[0,2]});
```


Off-resonance vs velocity: DRHS
Control
Label
Difference
Off-resonance (Hz)
Velocity (cm/s)
Mz/M0
ΔMz/M0

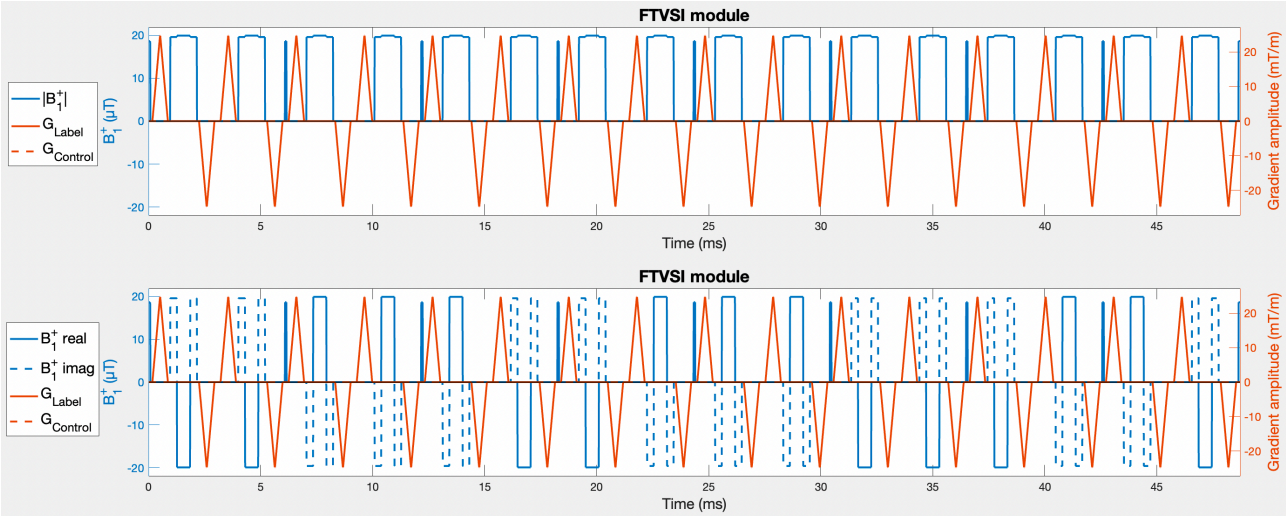
```
%% Eddy currents time constant vs position

% Simulates label and control module for a range of positions and
% eddy-current time constants.

% Eddy currents can be modelled as a convolution between the intended
% gradient and an exponential decaying gradient field, known as the eddy
% current field. The exponential decay is characterised by a time constant
% and amplitude.

% Eddy current fields can be non-spatially varying (B0 term) or have linear
% or higher order components. A Z-gradient can induce eddy current fields
% that vary spatially along Z as well as other directions, but here we just
% consider individual linear eddy currents terms parallel to the intended
% gradient.

% Eddy current artifacts in VSASL occur due to a non-zero 0th gradient
% moment resulting from the unbalanced phase accrual due to eddy currents.
% The dominant effect of this is the "labeling" of static tissue, which can
% result in a larger difference signal than the perfusion signal of
% interest.

% Some VSASL modules are less sensitive to eddy currents due to their
% design. However, eddy current artifacts can also be reduced by increasing
% pad2 or decreasing the maximum gradient amplitude (see Meakin and Jezzard
% MRM 2013. https://doi.org/10.1002/mrm.24302 and Guo et al. MRM 2015.
% https://doi.org/10.1002/mrm.25227). These approaches are explored in this
% simulation.

% More complicated methods for reducing eddy current artifacts include
% adjusting the gradient timings (see Woods et al. ISMRM 2021.
% https://cds.ismrm.org/protected/21MPresentations/abstracts/2720.html) or
% by using higher order pre-emphasis (see Zhao et al. ISMRM 2021.
% https://cds.ismrm.org/protected/21MPresentations/abstracts/3961.html).
% These are not explored in this simulation.

% Tasks: 1. Run the simulation for each VSASL module. Are some VSASL
%           modules more sensitive to eddy currents than others?
%        2. What happens when pad2 is increased to 2 ms?
%        3. What happens when Gmax is reduced to 20 mT/m?
%        4. What are the downsides of increasing pad2 or decreasing Gmax?
%        5. What happens when a velocity-compensated control module is used?
%        6. (ADVANCED) Alter code to plot eddy currents against velocity
%           for a fixed position of 24 cm to demonstrate how the velocity
%           profile is distorted by eddy current effects.

% Generate linear eddy currents parallel to VS gradients
tau = logspace(3,-1,100); % eddy current time constants (ms)
A   = 0.0025;             % eddy current amplitude (fraction of input gradient)
GLabel_EC_Hzcm = GLabel_Hzcm + gradec(GLabel_Hzcm, A, tau, GUP); % add eddy
currents to gradients
GCont_EC_Hzcm  = GCont_Hzcm  + gradec(GCont_Hzcm , A, tau, GUP); % add eddy
currents to gradients

B1scale = 1;                    % B1-variation (fraction)
df      = 0;                    % off-resonance (Hz)
dp      = linspace(-24,24,101); % position (cm)
dv      = 0;                    % velocity (cm/s)

% Run Bloch simulations
mzlabel = zeros(length(tau),length(dp));
mzcont  = zeros(length(tau),length(dp));
for ii = 1:length(tau)
    [~,~,mzlabel(ii,:)] = bloch_Hz(B1scale*B1_Hz, GLabel_EC_Hzcm(:,ii), RFUP*1e-
6, T1, T2, df, dp, dv, 0); % Label
```

```
    [~,~,mzcont(ii,:)]  = bloch_Hz(B1scale*B1_Hz, GCont_EC_Hzcm(:,ii) , RFUP*1e-
6, T1, T2, df, dp, dv, 0); % Control
end

% ASL subtraction
if strcmpi(vsType,'FTVSI'); dmz = mzlabel - mzcont;
else;                       dmz = mzcont - mzlabel; end

% Plot data
data1 = {dp,dp,dp};
data2 = {tau,tau,tau};
data3 = {mzcont,mzlabel,dmz};
ti    = {'Control','Label','Difference'};
cbl   = {'Mz/M_0','Mz/M_0','ΔMz/M_0'};
if strcmpi(vsType,'FTVSI'); clim = {[-1,0],[-1,0],[-1,1]};
else;                       clim = {[ 0,1],[ 0,1],[-1,1]}; end

surf_custom('data1',data1,'data2',data2,'data3',data3,...
            'name',['Eddy currents time constant vs position: ' vsType],...
            'xlabel',{'Position (cm)'},'ylabel',{'Time constant
(ms)'},'title',ti,...
            'color',{'jet'},'colorbarlabel',cbl,...
            'yscale',{'log'},'clim',clim);
```



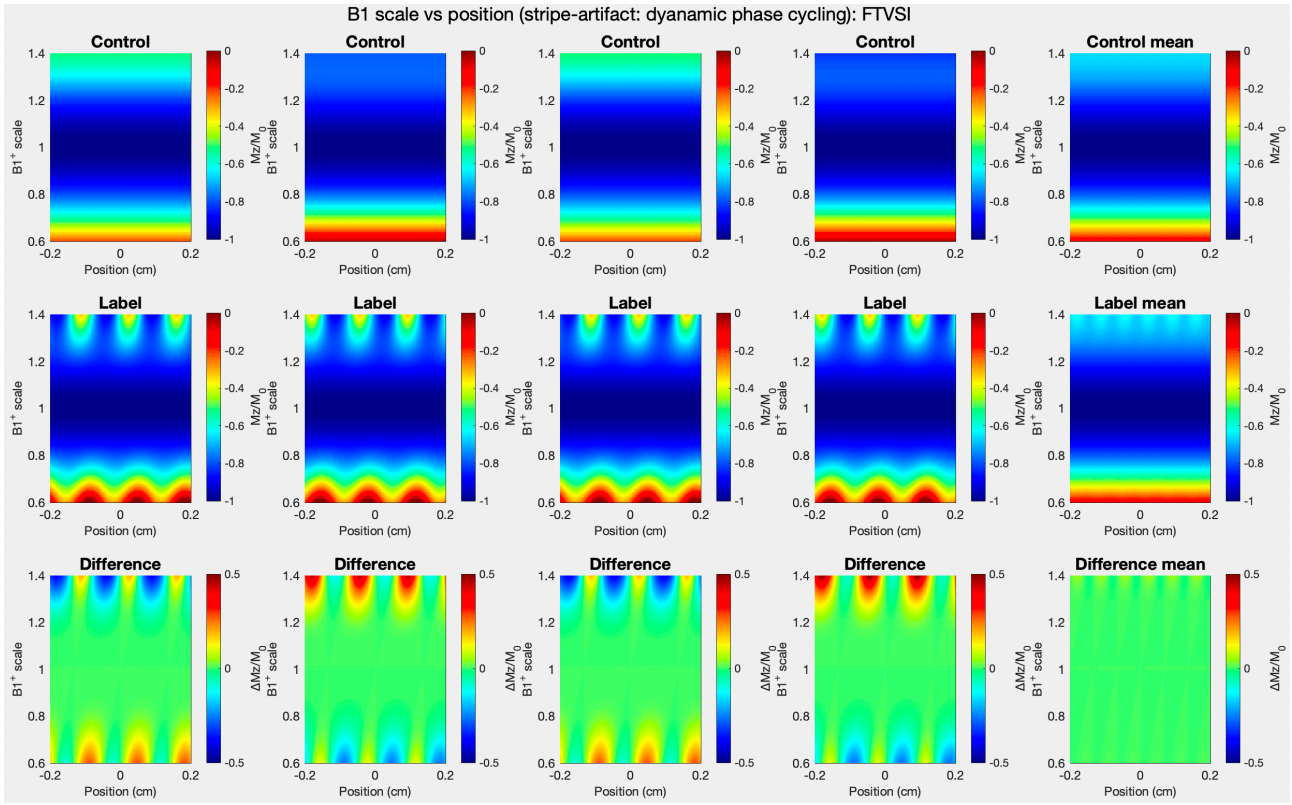
NOTE: Now running with:

```
vsType = 'FTVSI';
bvelCompCont = false;
bcomposite   = true;
bsinc        = false;
```




```
%% B1 vs B0 (dyanamic phase cycling)

% Simulates label and control module for a range of B0 and B1 scalings
% averaged across a 4 mm voxel of static spins.

% Devations from the nominal B1 can lead to inaccurate refocussing for
% FT-VSI (much less so for other VSASL modules that use adiabatic
% refocussing). This means that the phase accumulation by the
% velocity-encoding gradients is not perfectly reversed for static spins,
% creating spatial "stripe"-artifacts (see Liu et al. MRM 2021.
% https://doi.org/10.1002/mrm.28622).

% At the resolution used in perfusion imaging, these spatial stripes are
% averaged but do not cancel out. Instead, they create signal offsets in
% the static tissue, so called "DC-bias".

% Liu et al. proposed using dynamic phase cycling, where a 90° phase is
% added to the refocussing pulses over x4 TRs. This changes the stripe
% pattern in each acquisition so that it cancels out after averaging the
% separate acquisitions.

% This simulation demonstrates these spatial stripes and DC-bias and the
% effect of phase cycling.

% Tasks: 1. Run the simulations for using FT-VSI.
%        2. What happens to the spatial stripes when off-resonance is 200 Hz?
%        3. Is this as strong an effect for the other VSASL modules?

% The simulations have been split into two parts for speed:
%  Part 1: simulate spatial stripes (fix off-resonance)
%  Part 2: simulate DC-bias over B1scale and off-resonance
```

```
%% Part 1: simulate spatial stripes (fix off-resonance)

B1scale = linspace(0.6,1.4,101);  % B1-variation (fraction)
df      = 0;                      % off-resonance (Hz)
dp      = linspace(-0.2,0.2,101); % position (cm)
dv      = 0;                      % velocity (cm/s)

% Compatibility checks
if strcmpi(vsType,'BIR4'); error('Dynamic phase cycling not compatible with BIR-
4!'); end
if bvelCompCont; warning('Dynamic-phase cycling is not recommanded for velocity
compensated control!'); end

dynphase = size(T.B1,2); % x4 TR phase cycling
mzlabel  = zeros(length(B1scale),length(dp),dynphase);
mzcont   = zeros(length(B1scale),length(dp),dynphase);
dmz      = zeros(length(B1scale),length(dp),dynphase);
for jj = 1:dynphase

    % Update B1 waveform (convert to Hz)
    B1_Hz = T.B1(:,jj) * T.gam; % phase cycle refocussing pulses

    % Run Bloch simulations
    for ii = 1:length(B1scale)
        [~,~,mzlabel(ii,:,jj)] = bloch_Hz(B1scale(ii)*B1_Hz, GLabel_Hzcm,
RFUP*1e-6, T1, T2, df, dp, dv, 0);
        [~,~,mzcont(ii,:,jj)]  = bloch_Hz(B1scale(ii)*B1_Hz, GCont_Hzcm ,
RFUP*1e-6, T1, T2, df, dp, dv, 0);
    end

    % ASL subtraction
    if strcmpi(vsType,'FTVSI'); dmz(:,:,jj) = mzlabel(:,:,jj) - mzcont(:,:,jj);
    else;                       dmz(:,:,jj) = mzcont(:,:,jj) - mzlabel(:,:,jj);
end

end

% Mean over phase cycled TRs
mzlabelMean = mean(mzlabel,3);
mzcontMean  = mean(mzcont ,3);
dmzMean     = mean(dmz    ,3);

% Plot data across B1 and positions at max off-resonance (stripe-artifacts)
data1 = repmat({dp,dp,dp},1,dynphase+1);
data2 = repmat({B1scale,B1scale,B1scale},1,dynphase+1);
data3 = [squeeze(mat2cell(
mzcont,length(B1scale),length(dp),ones(1,dynphase)))',mzcontMean,...

squeeze(mat2cell(mzlabel,length(B1scale),length(dp),ones(1,dynphase)))',mzlabelM
ean,...
         squeeze(mat2cell(
dmz,length(B1scale),length(dp),ones(1,dynphase)))',dmzMean];
ti    = [repmat({'Control'},1,dynphase),{'Control mean'},...
         repmat({'Label'}  ,1,dynphase),{'Label mean'}   ,...
         repmat({'Difference'},1,dynphase),{'Difference mean'}];
cbl   = [repmat({'Mz/M_0'},1,2*(dynphase+1)),repmat({'ΔMz/M_0'},1,dynphase+1)];
clim  = [repmat({[-1,0]},1,2*(dynphase+1)),repmat({[-0.5,0.5]},1,dynphase+1)];

surf_custom('data1',data1,'data2',data2,'data3',data3,...
            'name',['B1 scale vs position (stripe-artifact: dyanamic phase
cycling): ' vsType],...
            'xlabel',{'Position (cm)'},'ylabel',{'B1^+ scale'},'title',ti,...
            'color',{'jet'},'colorbarlabel',cbl,...
            'clim',clim,'dim',[3,dynphase+1],...
            'labelfontsize',12);
```


B1 scale vs position (stripe-artifact: dyanamic phase cycling): FTVSI
Control
Control
Control
Control
Control mean
Label
Label
Label
Label
Label mean
Difference
Difference
Difference
Difference
Difference mean
B1+ scale
Position (cm)
Mz/M0
ΔMz/M0

```
%% Part 2: simulate DC-bias over B1scale and off-resonance

B1scale = linspace(0.6,1.4,41);  % B1-variation (fraction)
df      = linspace(-300,300,41); % off-resonance (Hz)
dp      = linspace(-0.2,0.2,11); % position (cm)
dv      = 0;                     % velocity (cm/s)

% Compatibility checks
if strcmpi(vsType,'BIR4'); error('Dynamic phase cycling not compatible with BIR-
4!'); end
if bvelCompCont; warning('Dynamic-phase cycling is not recommanded for velocity
compensated control!'); end

dynphase = size(T.B1,2); % x4 TR phase cycling
mzlabel  = zeros(length(B1scale),length(dp),length(df),dynphase);
mzcont   = zeros(length(B1scale),length(dp),length(df),dynphase);
dmz      = zeros(length(B1scale),length(dp),length(df),dynphase);
for jj = 1:dynphase

    % Update B1 waveform (convert to Hz)
    B1_Hz = T.B1(:,jj) * T.gam; % phase cycle refocussing pulses

    % Run Bloch simulations
    for ii = 1:length(B1scale)
        [~,~,mzlabel(ii,:,:,jj)] = bloch_Hz(B1scale(ii)*B1_Hz, GLabel_Hzcm,
RFUP*1e-6, T1, T2, df, dp, dv, 0);
        [~,~,mzcont(ii,:,:,jj)]  = bloch_Hz(B1scale(ii)*B1_Hz, GCont_Hzcm ,
RFUP*1e-6, T1, T2, df, dp, dv, 0);
    end

    % ASL subtraction
    if strcmpi(vsType,'FTVSI'); dmz(:,:,:,jj) = mzlabel(:,:,:,jj) -
mzcont(:,:,:,jj);
    else;                       dmz(:,:,:,jj) = mzcont(:,:,:,jj) -
mzlabel(:,:,:,jj); end

end

% Mean over phase cycled TRs
mzlabelMean = mean(mzlabel,4);
mzcontMean  = mean(mzcont ,4);
dmzMean     = mean(dmz    ,4);

% Mean over voxel
mzlabelVox    = squeeze(mean(mzlabel,2));
mzcontVox     = squeeze(mean(mzcont ,2));
dmzVox        = squeeze(mean(dmz    ,2));
mzabelMeanVox = squeeze(mean(mzlabelMean,2));
mzcontMeanVox = squeeze(mean(mzcontMean,2));
dmzMeanVox    = squeeze(mean(dmzMean,2));

% Plot data across B1 and positions at max off-resonance (stripe-artifacts)
data1 = repmat({df,df,df},1,dynphase+1);
data2 = repmat({B1scale,B1scale,B1scale},1,dynphase+1);
data3 = [squeeze(mat2cell(
mzcontVox,length(B1scale),length(df),[1,1,1,1]))',mzcontMeanVox,...

squeeze(mat2cell(mzlabelVox,length(B1scale),length(df),[1,1,1,1]))',mzabelMeanVo
x,...
         squeeze(mat2cell(
dmzVox,length(B1scale),length(df),[1,1,1,1]))',dmzMeanVox];
ti    = [repmat({'Control'},1,dynphase),{'Control mean'},...
         repmat({'Label'}  ,1,dynphase),{'Label mean'}   ,...
         repmat({'Difference'},1,dynphase),{'Difference mean'}];
cbl   = [repmat({'Mz/M_0'},1,2*(dynphase+1)),repmat({'ΔMz/M_0'},1,dynphase+1)];
clim  = [repmat({[-1,1]},1,2*(dynphase+1)),repmat({[-0.5,0.5]},1,dynphase+1)];
```

```
surf_custom('data1',data1,'data2',data2,'data3',data3,...
            'name',['B1 scale vs off-resonance (DC-bias, dyanamic phase
cycling): ' vsType],...
            'xlabel',{'Off-resonance (Hz)'},'ylabel',{'B1^+
scale'},'title',ti,...
            'color',{'jet'},'colorbarlabel',cbl,...
            'clim',clim,'dim',[3,dynphase+1],...
            'labelfontsize',12);
```