Spin Echo 2D

Example_SpinEcho2D_basic.m

for openMatlab interface

Pure Devices GmbH
Eisenbahnstr. 53
97084 Würzburg
GERMANY

Date: 15.11.2012

Preface

This document will describe a basic Spin Echo 2D sequence that can be used with the openMatlab interface from Pure Devices. All examples can be directly used in Matlab by copy and paste or by using the provided example files.

It is assumed that the reader is familiar in writing code for Matlab and to imaging sequence for magnetic resonance.

Preparation

To run the examples provided in this document we recommend to use a sample with a short T1-relaxation-time like e. g. an oil sample. Users that will use a magnet provided by Pure Devices may use the sample *10 mm Oil* or the *10 mm Structure* sample.

Frequency determination

The description of the example code *Example_SpinEcho2D_basic.m* is provided on the following pages.

```
%% Basic Spin Echo 2D

    % Preparations
    LoadSystem                                                  % load system parameters
    [HW,mySave]  = Find_Frequency_Sweep(talker, HW, mySave,0);  % determinning magnet frequency
    %HW.fLarmor = 22e6;                                         % uncomment for entering frequency
```

Each sequence that is written with the openMatlab interface needs to start with loading the system parameters. This ensures that all calibration values are loaded . It is recommended to load the parameters at the beginning of each sequence by calling the script `LoadSystem`.

The line: `[HW,mySave]  = Find_Frequency_Sweep(talker, HW, mySave,0);` will automatically determine the Larmor frequency of the system. The result will be stored in the variable `HW.fLarmor`. Please note that for homebuilt magnets it might be necessary to adapt the variable `HW.B0` either in the LoadMySystem file or before running `Find_Frequency_Sweep`. If the Larmor frequency of the system is well known one can also omit the determination function and enter the frequency manually into `HW.fLarmor`.

Frequency determination

The following lines can be directly executed in Matlab.

```
%% Example: Exact frequency determination for a 0.5 Tesla magnet

    % Preparations
    LoadSystem                                                   % load system parameters
    HW.B0 = 0.5;                                                 % set B0 to 0.5 Tesla
    [HW,mySave]  = Find_Frequency_Sweep(talker, HW, mySave,0);   % determining magnet frequency
    disp(HW.fLarmor);                                            % displays the frequency
```

Running the code above in Matlab will give the following result for a magspec magnet.

```
Command Window
Searching frequency...
Waiting 2 seconds after finding Frequency.
   2.1937e+07
```

Please note that the value of the frequency may slightly change due to temperature drifts. Therefore it is recommended to run the function for frequency determination before starting an experiment.

Parameter definition

The following variables are used for the spin echo 2D sequence calculations. Their values can be modified to experiment with this sequence.

```
% Define parameters for a basic spin echo 2D sequency
    nphase   = 64;                  % phase encoding steps
    nsamples = 64;                  % samples that will be acquired per acquisition
    echotime = 5e-3;                % the echo time in seconds
    fsample  = 100e3;               % sample frequency in Hz
    trep     = 0.2;                 % repetiotion time in seconds
    averages = 1;                   % number of averages (1=none)

    grad_read  = 3;                 % gradient channel for read gradient
    grad_phase = 1;                 % gradient channel for phase gradient

    grad_read_amp  = 150e-3;        % gradient amplitude read gradient in Tesla / meter
    grad_phase_amp = 180e-3;        % gradient amplitude phase gradient in Tesla / meter
    grad_ramptime  = 100e-6;        % ramp time for all gradients

    p90  = 45e-6;                   % rf-pulse duration 90° pulse in seconds
    p180 = 90e-6;                   % rf-pulse duration 180° pulse in seconds
```

Parameter assignment

The following variables are used for the spin echo 2D calculations.

```
% Parameters used for timing calculations
% Sequence parameters
Seq.plotSeq     =   [1 3];                % plots the sequence: RF, AQ, Grad(1), Grad(3)
Seq.tRep        =   ones(1,nphase)*trep; % assign repetition time to Seq structure
Seq.average     =   averages;            % assign averages to Seq structure

% RF transmitter parameters
TX.Start        =[  0;...            % start time of 90° rf-pulse
                    echotime/2];     % start time of 180° rf-pulse
TX.Duration     =[  p90;...          % duration of 90° rf-pulse
                    p180];           % duration of 180° rf-pulse
TX.Frequency    =[  HW.fLarmor;...   % frequency of 90° rf-pulse
                    HW.fLarmor];     % frequency of 180° rf-pulse
TX.Phase        =[  0;...            % phase of 90° rf-pulse
                    0];              % phase of 180° rf-pulse

% Aquisition parameters
AQ.fSample      = [ fsample ];                               % sample frequency of acquisition
AQ.Start        = [ echotime - ((nsamples / 2)/fsample)];    % start time of acquisition
AQ.nSamples     = [ nsamples ];                              % samples to acquire
AQ.Frequency    = [ HW.fLarmor];                             % frequency of acquisition
AQ.Phase        = [ 0 ];                                     % phase of acquisition

% Help variables
zero_matrix=zeros(1,nphase);
ones_matrix=ones(1,nphase);
```

Gradient calculations

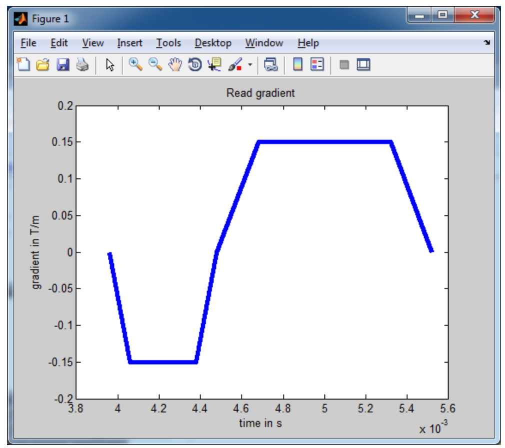
The following code shows the gradient assignment. Please refer to the file *Example_SpinEcho2D_basic.m* for the complete overview.

```
% Gradients time and amplitude calculations
% Gradient read
Grad(grad_read).Time = see Example_SpinEcho2D_basic.m

Grad(grad_read).Amp  = see Example_SpinEcho2D_basic.m

% Gradient phase
Grad(grad_phase).Time = see Example_SpinEcho2D_basic.m

% calculation of the amplitudes for the phase encoding steps
phase_amps = cumsum(ones_matrix * -grad_phase_amp/nphase)+grad_phase_amp/2;

Grad(grad_phase).Amp  = see Example_SpinEcho2D_basic.m

% Check if placement of read-gradient is after 2nd rf-pulse
if(Grad(grad_read).Time(1)<TX.Start(2));
    error('Echotime too short.');
end;
```

Gradient visualization - read

Running the following code will plot the gradient slope of the read-gradient.

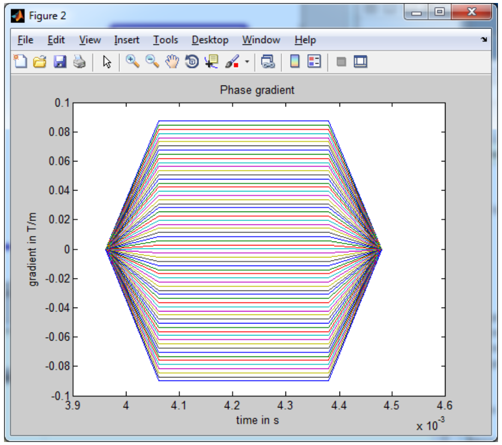
```
% Gradients time and amplitude calculations
figure(1);
plot(Grad(grad_read).Time,Grad(grad_read).Amp,'LineWidth',4);
xlabel('time in s');
ylabel('gradient in T/m');
title('Read Gradient');
ylim([-0.2 0.2]);
```



Gradient visualization - phase

The phase encoding steps of the phase gradient can be plotted the same way as the read gradient by executing the following lines:

```
% Gradients time and amplitude calculations
figure(2);
plot(Grad(grad_phase).Time,Grad(grad_phase).Amp,'LineWidth',1);
xlabel('time in s');
ylabel('gradient in T/m');
title(,Phase Gradient');
ylim([-0.1 0.1]);
```



Starting the sequency and preparing results

The last lines of the sequence script will start the measurement and prepare the results.

```
% Start measurement
[ Raw, SeqOut, data, data_1D ] = set_sequence(HW, Seq, AQ, TX, Grad, talker);

% Plot results
plot_data_1D(HW, data_1D);

% Displaying the result
kspace = squeeze(data.data);
imagespace = abs(fftshift(fft2(kspace)));

figure(4)
subplot(2,1,1);
imagesc(log(abs(kspace)));
title('k-space');
xlabel('phase direction');
ylabel('read direction');
colormap(gray);
subplot(2,1,2);
imagesc(abs(imagespace));
title('Image (2D FFT of the k-space)');
xlabel('phase direction');
ylabel('read direction');

% cleanup
clear ones_matrix;
clear zero_matrix;
```

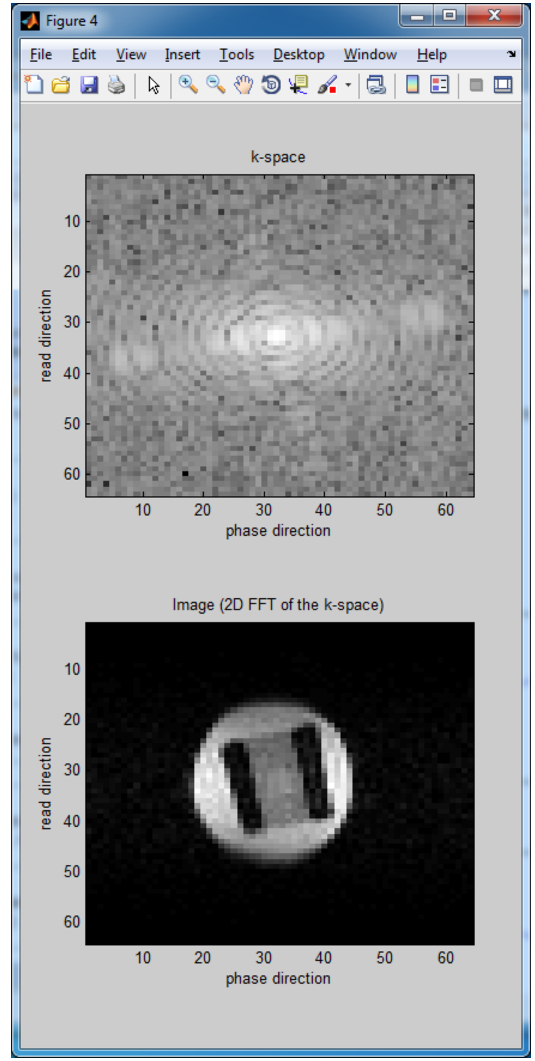


Figure 4 shows an exemplary result of running the complete sequence. The *10 mm Structure* sample was used for this experiment.

Example_SpinEcho2D_basic.m – complete sequence (1/3)

```
%% Demo Sequence "Example_SpinEcho2D_basic.m"
% This demo sequence demonstrates how to create a basic spin echo 2D
% sequence.

%% Basic Spin Echo 2D

% Preparations
LoadSystem                                                  % load system parameters
[HW,mySave]  = Find_Frequency_Sweep(talker, HW, mySave,0);  % determining magnet frequency
%HW.fLarmor = 21.947e6;                                     % uncomment for entering frequency

% Define parameters for a basic spin echo 2D sequency
nphase   = 64;                  % phase encoding steps
nsamples = 64;                  % samples that will be acquired per acquisition
echotime = 5e-3;                % the echo time in seconds
fsample  = 100e3;               % sample frequency in Hz
trep     = 0.2;                 % repetiotion time in seconds
averages = 1;                   % number of averages (1=none)

grad_read  = 3;                 % gradient channel for read gradient
grad_phase = 1;                 % gradient channel for phase gradient

grad_read_amp  = 150e-3;        % gradient amplitude read gradient in Tesla / meter
grad_phase_amp = 180e-3;        % gradient amplitude phase gradient in Tesla / meter
grad_ramptime  = 100e-6;        % ramp time for all gradients

p90  = 45e-6;                   % rf-pulse duration 90° pulse in seconds
p180 = 90e-6;                   % rf-pulse duration 180° pulse in seconds

% Parameters used for timing calculations
% Sequence parameters
Seq.plotSeq     =   [1 3];               % plots the sequence: RF, AQ, Grad(1), Grad(3)
Seq.tRep        =   ones(1,nphase)*trep; % assign repetition time to Seq structure
Seq.average     =   averages;            % assign averages to Seq structure

% RF transmitter parameters
TX.Start        =[  0;...           % start time of 90° rf-pulse
                    echotime/2];    % start time of 180° rf-pulse
TX.Duration     =[  p90;...         % duration of 90° rf-pulse
                    p180];          % duration of 180° rf-pulse
TX.Frequency    =[  HW.fLarmor;...  % frequency of 90° rf-pulse
                    HW.fLarmor];    % frequency of 180° rf-pulse
TX.Phase        =[  0;...           % phase of 90° rf-pulse
                    0];             % phase of 180° rf-pulse
```

Example_SpinEcho2D_basic.m – complete sequence (2/3)

```
% Aquisition parameters
AQ.fSample      = [ fsample ];                               % sample frequency of acquisition
AQ.Start        = [ echotime - ((nsamples / 2)/fsample)];    % start time of acquisition
AQ.nSamples     = [ nsamples ];                              % samples to acquire
AQ.Frequency    = [ HW.fLarmor];                             % frequency of acquisition
AQ.Phase        = [ 0 ];                                     % phase of acquisition

% Help variables
zero_matrix=zeros(1,nphase);
ones_matrix=ones(1,nphase);


% Gradients time and amplitude calculations
% Gradient read
Grad(grad_read).Time = [(echotime-((nsamples/2)/fsample)-2*grad_ramptime-1*grad_ramptime-((nsamples/2)/fsample)-1*grad_ramptime);...
                            (echotime-((nsamples/2)/fsample)-2*grad_ramptime-1*grad_ramptime-((nsamples/2)/fsample));...
                            (echotime-((nsamples/2)/fsample)-2*grad_ramptime-1*grad_ramptime);...
                        (echotime-((nsamples/2)/fsample)-2*grad_ramptime);...
                            (echotime-((nsamples/2)/fsample));...
                            (echotime+((nsamples/2)/fsample));...
                        (echotime+((nsamples/2)/fsample)+2*grad_ramptime)];

Grad(grad_read).Amp  = [0;...
                            -grad_read_amp;...
                            -grad_read_amp;...
                        0;...
                            grad_read_amp;...
                            grad_read_amp;...
                        0];

% Gradient phase
Grad(grad_phase).Time = [Grad(grad_read).Time(1)*ones_matrix;...
                             Grad(grad_read).Time(2)*ones_matrix;...
                             Grad(grad_read).Time(3)*ones_matrix;...
                         Grad(grad_read).Time(4)*ones_matrix];

% calculation of the amplitudes for the phase encoding steps
phase_amps = cumsum(ones_matrix * -grad_phase_amp/nphase) + grad_phase_amp/2;

Grad(grad_phase).Amp  = [0*zero_matrix;...
                            phase_amps;...
                            phase_amps;...
                         0*zero_matrix];
```

Example_SpinEcho2D_basic.m – complete sequence (3/3)

```
% Check if placement of read-gradient is after 2nd rf-pulse
if(Grad(grad_read).Time(1)<TX.Start(2));
    error('Echotime too short.');
end;

% Start measurement
[ Raw, SeqOut, data, data_1D ] = set_sequence(HW, Seq, AQ, TX, Grad, talker);

% Plot results
plot_data_1D(HW, data_1D);

% Displaying the result
kspace = squeeze(data.data);
imagespace = abs(fftshift(fft2(kspace)));

figure(4)
subplot(2,1,1);
imagesc(log(abs(kspace)));
title('k-space');
xlabel('phase direction');
ylabel('read direction');
colormap(gray);
subplot(2,1,2);
imagesc(abs(imagespace));
title('Image (2D FFT of the k-space)');
xlabel('phase direction');
ylabel('read direction');

% cleanup
clear ones_matrix;
clear zero_matrix;


%% -------------------------------------------------------------------------
% (C) Copyright 2012 Pure Devices GmbH, Wuerzburg, Germany
% www.pure-devices.com
%-------------------------------------------------------------------------
```