

M229 Advanced Topics in MRI, Spring 2024

Homework 1: RF Pulse Design

Assigned: 2024.04.09; Due: 5 pm, Fri, 2024.04.19 by email

Questions? Email HoldenWu@mednet.ucla.edu

Turn in (1) a PDF with your simulation results and discussions, and (2) your MATLAB code. Include comments in your code to improve readability.

1. Consider a nonselective excitation in which a constant pulse of amplitude B_1 and duration τ is applied in the presence of B_0 . If the carrier frequency ω is not exactly tuned to ω_0 (i.e., $\omega \neq \omega_0$), determine the resultant transverse magnetization output at time τ . Use the small tip angle solution based in the rotating frame at the carrier frequency ω .

2. Adiabatic Full Passage Pulse Design using Hyperbolic Secant Pulse Equations

Design an adiabatic RF pulse using the method outlined in class:

$$\begin{aligned} B_1(t) &= A(t)e^{-i \int \omega(t')t'} \\ A(t) &= A_0 \operatorname{sech}(\beta t) \\ \omega_1(t) &= -\mu\beta \tanh(\beta t) \end{aligned}$$

a) amplitude modulation function

Design an amplitude modulation function, A_t , with *sech*. Use $\beta = 672$ [rad/s] and $A_0 = 0.12$ [G]. Use 512 samples for the pulse duration of 10.24 ms.

```
>> beta = 672;
>> pulseWidth = 10.24; % in [ms]
>> A0 = 0.12 % in [G]
>> nSamples = 512; % the number of samples
>> dt = pulseWidth/nSamples/1000; % time step in [sec]
```

Plot the amplitude modulation function in *Gauss*.

```
>> plot(time, A_t); grid on;
>> title('Amplitude Modulation Function'); xlabel('Time (ms)'); ylabel('A(t) (G)');
```

b) frequency modulation function

Design a frequency modulation function, $w1_t$, with *tanh*. Use $\mu = 5$ [dimensionless].

```
>> mu = 5; % [dimensionless]
```

Plot the frequency modulation function in *Hz*.

```
>> plot(time,w1_t); grid on;
>> title('Frequency Modulation Function'); xlabel('Time (ms)'); ylabel('\omega_1(t) (Hz)');
```

c) inversion profile using Bloch simulation

Combine amplitude and frequency modulation functions:

```
>> rf_pulse = A_t .* exp(1i .* cumsum(w1_t)*dt);
```

Or, you can use phase modulation function instead:

$$\phi(t) = \mu \ln(\operatorname{sech} \beta t)$$

```
>> rf_pulse = A_t .* exp(1i .* phi);
```

Simulate the inversion profile over a sufficient range of frequency (e.g. -4,000 Hz to 4,000 Hz) using Bloch simulation. Plot the inversion profile.

```
>> plot(freq_range, mz);  
>> title('Inversion Profile'); xlabel('Frequency (Hz)'); ylabel('M_z'); grid on;
```

d) inversion profiles with different B1+ variation

Simulate the inversion profiles with different B1+ variations (60% attenuated, 30% attenuated, and 150% amplified pulses). Plot the inversion profiles.

3. 2D EPI Pulse Design

Design a 2D separable EPI RF pulse using the method outlined in class.

a) gradient design

Design a blipped EPI trajectory with $k_{x,max} = k_{y,max} = 0.5$ cycles/cm, and nine lines ($L=9$). Assume the trapezoid ramps (τ_R) are 1/8 ms long, and the trapezoids themselves (τ) are 1 ms long. The blips are 1/4 ms long. What is the maximum amplitude of G_x and G_y ?

Sample the RF and gradient waveforms at 5 us (200 samples per trapezoid and 25 samples per trapezoid ramp). Include a refocusing lobe at the end to bring the trajectory back to $k_x = k_y = 0$, using 1 ms trapezoids on x and y . Plot the gradient waveforms, G_x and G_y , with the axes labeled.

b) RF pulse design

Design the RF waveform with $TBW = 4$ for the subpulses, and the envelope. This will produce a 4 cm by 4 cm excited volume.

```
>> tbw = 4;  
>> rf_fast = wsinc(tbw,samples);  
>> rf_slow = wsinc(tbw,L);
```

Apply the “flat-top only design” (RF only played flat part), and use the RF waveform to be zero during the refocusing gradient. Scale the RF to a flip angle of 1 radian (i.e. $\text{sum}(\text{rf}) = 1$). Plot the RF waveform in Gauss.

c) 2D Bloch simulation

Simulate the pulse over a sufficient range (e.g., -12cm to 12cm in x and y) at on-resonance. Plot the profile as an image using

```
>> imshow(abs(mxy),[]);
```

and cross-section plots along x (M_{xy} vs. x) and y (M_{xy} vs. y)

```
>> subplot(211); plot(x,abs(mxy(:,round(length(y)/2))));
```

```
>> subplot(212); plot(y,abs(mxy(round(length(x)/2),:)));
```