The Fastest Arbitrary k-space Trajectories

Sana Vaziri[1], and Michael Lustig[1]
[1]Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA, United States

Introduction:

We present a method to compute the fastest possible gradient waveforms for a given k-space trajectory. In our design, we exploit the fact that each gradient set has its own limitation. This is an extension of our previous method described in [1]. In that solution, the worst-case magnitude gradient and slew constraints were considered. This has the advantage of resulting in waveforms that are invariant to any rotation of the trajectory, at the expense of sub-utilizing the hardware. Here, we present an improvement of the algorithm. We constraint the waveforms for each gradient set separately, while still requiring that the trajectory follows the desired k-space path. This produces the fastest waveforms. A redesign is needed when the trajectory is rotated, as the solution depends on the orientation of the curve. Our algorithm is fast and non-iterative and can compute waveforms on-the-fly.

Methods: The algorithm in [1] considers a gradient magnitude constraint of $(G_x^2+G_y^2)^{1/2} < G_{max}$ and slew-rate constraint of $(G'^2_x + G'^2_y) < S_{max}$. It takes any given k-space trajectory, C(p), and converts it to the Euclidian arc-length parameterization, C(s). It then determines a time function, s(t), with s(0) = 0 and s(T)=L, where T is the traversal time and L is the length of the curve. The slew and gradient constraints are then described in this parameterization. The time optimal solution is equivalent to determining the optimal velocity as a function of arc length. This solution is found by taking the minimum of a solution to an ordinary differential equation, solved forward and backwards. This is described in details in [1].

Here we modify the constraints in [1] to be separate for each gradient set, e.g., $|G_x| < G_{max}$, $|G_y| < G_{max}$, $|G'x| < S_{max}$ and $|G'_x| < S_{max}$. This results in the constraints and differential equation shown in Fig. 1.

The algorithm was implemented in the C programming language. Finite differences were used to approximate derivative operations. Integration was approximated using the trapezoid rule. Solutions to the ordinary differential equations were approximated using the 4th order Runge-Kutta method [2]. All necessary curve interpolations were done using cubic-spline interpolation.

Given: $C(p) = \{x(p), y(p), z(p)\}$

$$\alpha(s) = min\left\{\frac{\gamma G_{max}}{|x'(s)|}, \frac{\gamma G_{max}}{|y'(s)|}, \frac{\gamma G_{max}}{|z'(s)|}\right\}$$

$$\beta(s,\dot{s}) = min\left\{\frac{-|x''(s)|\dot{s}(t)^2+\gamma S_{max}}{|x'(s)|}, \frac{-|y''(s)|\dot{s}(t)^2+\gamma S_{max}}{|y'(s)|}, \frac{-|z''(s)|\dot{s}(t)^2+\gamma S_{max}}{|z'(s)|}\right\}$$

Forward ODE:

$$\frac{v_+(s)}{ds} = \begin{cases} \frac{1}{v_+(s)}\beta(s, v_+(s)), & \text{if } v_+(s) < \alpha(s) \\ \frac{d\alpha(s)}{ds}, & \text{otherwise} \end{cases}$$

Backward ODE:

$$\frac{v_-(s)}{ds} = \begin{cases} -\frac{1}{v_+(s)}\beta(s, v_-(s)), & \text{if } v_-(s) < \alpha(s) \\ -\frac{d\alpha(s)}{ds}, & \text{otherwise} \end{cases}$$

Figure 1 Gradient constraints and resulting differential equations

Results and Discussion: Fig. 2 shows examples of the fastest gradient waveforms for circular and spiral trajectories. For the circular trajectory [3], the fastest design resulted in 9.6% decrease in the readout duration compared to the rotationally invariant solution. For the spiral trajectory [4,5], we achieve a reduction of 5.5%. In conclusion, we presented an improvement to the time-optimal gradient design that exploits the separate constraints for each gradient set. This improvement can be substantial in fast acquisitions, such as SSFP for reduction of scan time and banding artifacts.

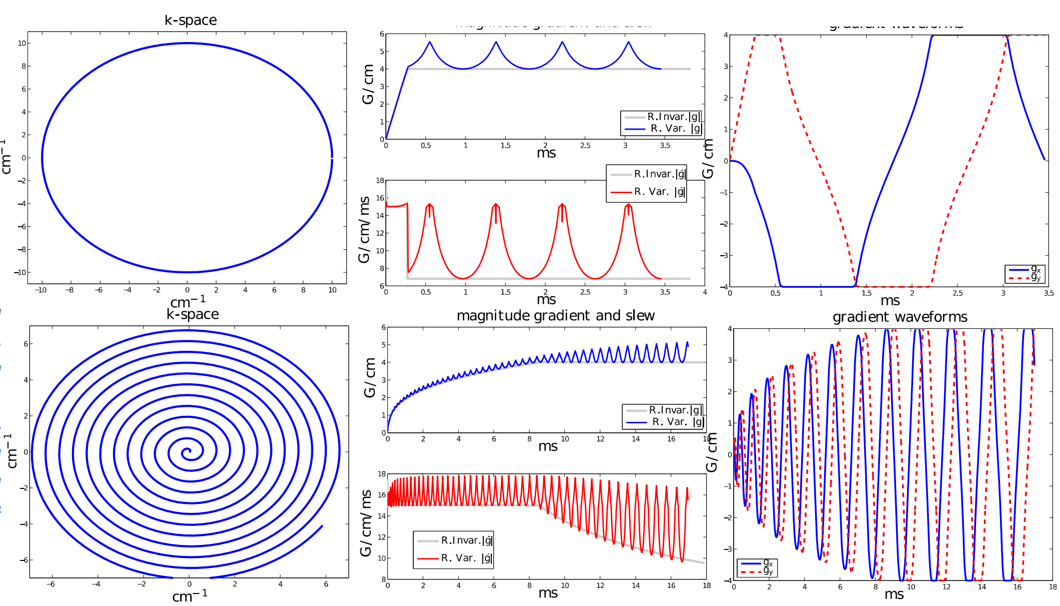

Figure 2 (a,d) Circular and spiral trajectories, (b,e) Gradient magnitude and slew-rate of our design compared to the rotationally invariant design. (c,f) The resulting gradient waveforms are rectified, yet are smooth and feasible to be played on a physical system. These achieve reduction of 9.6% and 5.5% respectively.

References:

[1] Lustig et. al ,IEEE-TMI 2008;6:866-73, [2] Boyce et. al, Elementary differential equations 6th edition, 1997. [3] Heid, ISMRM 2002; pp.2364 [4] King et. al, MRM 2004;51:81-92 [5] Meyer et. al, ISMRM 1996 pp:306