# A fast method for designing time-optimal gradient waveforms for arbitrary k-space trajectories

**M. Lustig[1], S. Kim[1], and J. M. Pauly[1]**

[1]Electrical Engineering, Stanford University, Stanford, CA, United States

**Introduction:** The design of time-optimal gradient waveforms is an important, yet difficult problem. A common approach is to first design a k-space curve and then its associated gradient waveforms [1-4]. To design the gradients one must find the optimal switching times between slew-limited accelerations, decelerations and gradient limited regions, this can be difficult if there are many. Here, we provide a complete, fast and simple non-iterative solution for arbitrary multi-dimensional k-space trajectories that avoids this difficulty. The approach can be used to design waveforms for simple as well as complicated trajectories. The user must provide a path, and the algorithm will return the gradients that will traverse that path in minimum time. It is important to mention that some optimal gradient designs that exist in the literature either solve for 1D waveforms [5], or provide waveforms that traverse k-space from one point to another <u>not on a specific path</u> [6-7].

**Theory:** Designing the time optimal gradient waveform is equivalent to a similar problem in optimal control of navigating a robotic arm along a given path [8]. Finding the optimal switching times can be avoided by formulating the hardware and path constraints in the Euclidean arc-length parametrization. Since the path is already set, it suffices to find the right velocities, i.e., the gradient amplitudes for each point along it. In the algorithm specified in Fig. 1, we first describe the curve as a function of the Euclidian arc-length *s*, i.e., *C(s)*. We then express the hardware constraints (slew-rate and maximum gradient) in the arc-length parameterization and get a differential inequality, which expresses the allowed rate of acceleration *dv/ds* at points along the curve. It is shown in [8,9] that the optimal velocity for each point along the curve can be found by integrating an ordinary differential equation (ODE) forward and backward and taking the minimum solution. Finding the gradient waveform from the velocity is straightforward.

$g^* = TimeOptimalGradient(C(s)\ ,\ g_0,\ g_{fin},\ G_{max}\ ,\ S_{max})$ {

compute curvature: $\kappa(s) = |C_{ss}(s)|$

compute: $\alpha(s) = \min\{\gamma G_{max}\ ,\ [\gamma Smax/\kappa(s)]^{1/2}\}$

define: $\beta(s, s_t) = [\gamma^2 Smax^2 - \kappa^2(s) s_t^4]^{1/2}$

set $v_+(0) = \gamma g_0$ , integrate ODE forward:

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

set $v_-(L) = \gamma g_{fin}$ , integrate ODE backward:

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

set $v^*(s) = \min\{\ v_+(s), v_-(s)\ \}$

compute: $s^*(t)$ using the inverse of $t^*(s) = \int_0^S 1/v^*(\sigma)d\sigma$

compute: $C^*(t) = C(s^*(t))$

compute: $g^*(t) = 1/\gamma\, dC^*(t)/dt$

}

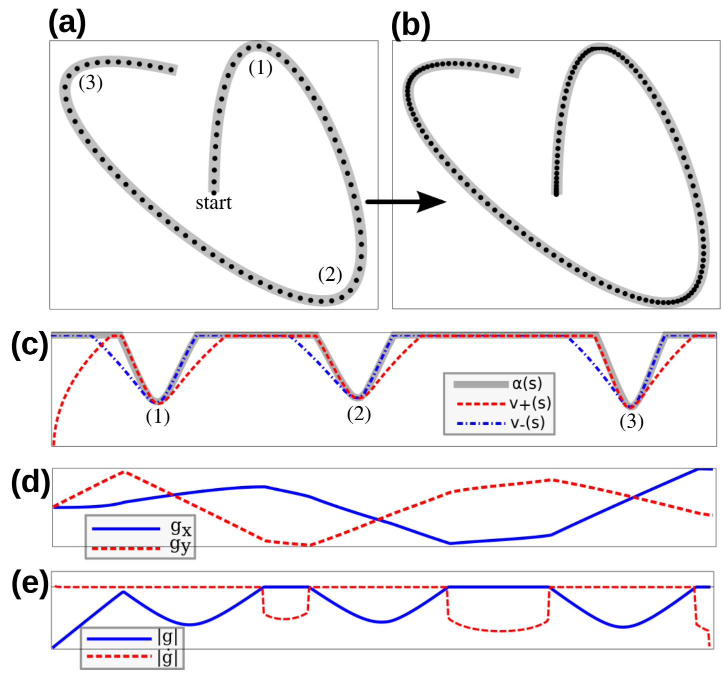


***Figure 1: Outline of the algorithm (left) and a simplified example (right). (a) Constant arc-length curve. (b) The time-optimal curve. (c) The H/W constraints and ODE solutions. (d) Resulting time-optimal gradient waveforms. (e) Either the magnitude gradient or slew-rate are maximized – a necessary condition for optimality.***

**Methods and Results:** The algorithm was implemented using Matlab™ and C programming language. Derivative operations were approximated by finite differences. Numerical integrations were approximated by the trapezoid method. The ordinary differential equations (ODEs) were solved using a 4th order Runge-Kutte method [10]. Cubic-spline interpolation was used for interpolating the curve when needed. As an example, we applied the method to design a dual density spiral and a rosette [4] trajectory. Both resulting waveforms are time-optimal. It is important to mention that the rosette design in [4] is NOT time optimal. The run-time for the C implementation was around 0.25 seconds.

**Discussion**: The proposed method provides a simple and fast general solution to design time-optimal gradient waveforms for any trajectory. This approach can also be used to design waveforms by prescribing a parametric curve and then designing the waveforms for it.

**References:** [1] Heid, Proc ISMRM, pp.2364, 2002 [2] King et al, MRM;51(1), pp. 81–92, 2004. [3] Meyer et al, Proc ISMRM, pp.306, 1996 [4] Noll, IEEE TMI; 52(4), pp. 831-841, 1997 [5] Simonetti et al, MRM;29(4), pp. 498-504, 1993 [6] Dale et al, ISMRM, p. 2361, 2002 [7] Hargreaves et al, MRM;51(1), pp.81-92, 2004 [8] Kim et al, IEEE Tran Auto Cont;50(7), pp. 967-983, 2005 [9] Lustig et al, "A fast method for designing time-optimal gradient waveforms for arbitrary k-space trajectories" IEEE TMI, submitted (http://www.stanford.edu/~mlustig/timeOptimalGradientDesign.pdf). [10] Boyce et al, Elementary Differential Equations. 6th ed 1997

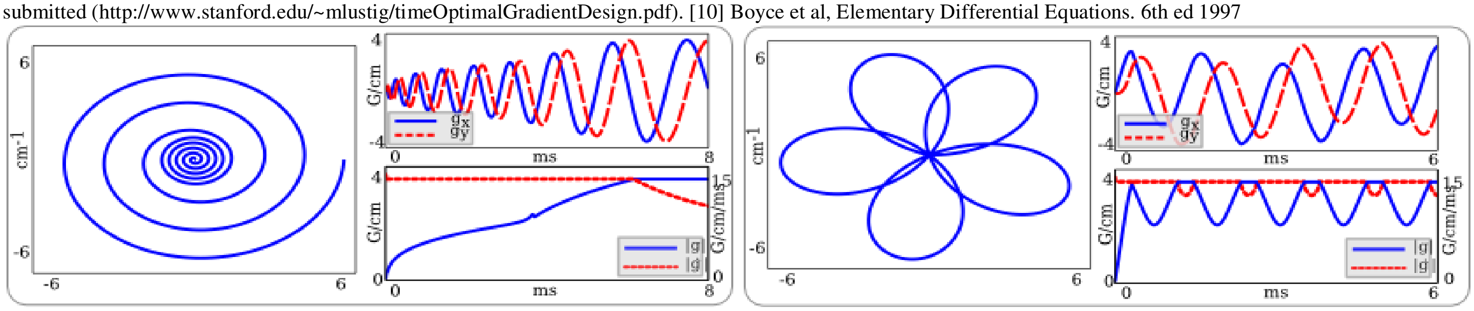


***Figure 2: Optimal dual density spiral design (left). The switch between densities is clearly seen in the magnitude gradient. Our waveform design for the rosette trajectory (right) is time optimal unlike the design in [4].***