Pulse Sequence Programming in a Dynamic Visual Environment

J. Magland[1], F. W. Wehrli[1]

[1]Laboratory for Structural NMR Imaging, Department of Radiology, University of Pennsylvania Medical Center, Philadelphia, PA, United States

Introduction

MRI is unique in that once the hardware requirements are met, the gateway to new method development is the pulse sequence programmer. The complexity of the pulse sequencer's programming environment can be a major obstacle to implementation of new ideas. Generally, there is a tradeoff between user-friendliness and flexibility. A purely visual programming environment, for example one in which the programmer easily manipulates the sequence using mouse clicks, may lack the flexibility required to design pulse sequences that deviate substantially from common Cartesian or radial sampling schemes. On the other hand, in a code-based development environment which is flexible enough to cover the widest range of possible research sequences, pulse programs can be difficult to read and time-consuming to develop or debug. Here we introduce a visual pulse sequence programming environment (*SequenceTree*) that combines ease-of-use with flexibility and that can be interfaced to any scanner provided that an appropriate conversion tool is created. The aim is to provide software which is intuitive for basic users (i.e. MR researchers with little programming experience) as well as for advanced users (i.e. those familiar with the C programming language).

Methods

A SequenceTree pulse sequence is organized into a hierarchy of pre-defined nodes (see **Figure 1**). The lowest level nodes represent RF pulses, imaging gradients and data acquisition (ADC) events, or those events sent individually to the scanner. These objects are bundled together to form higher level nodes. For example, an excitation segment is a node consisting of an RF pulse together with a slice select gradient, and an acquisition segment is a node consisting of an ADC event together with a readout gradient. These segments are themselves organized into a higher level node such as a gradient-echo block, which would consist of four segments: an excitation segment, a phase encoding segment, an acquisition segment and a rewinder segment. In turn, the blocks (imaging blocks, navigator blocks, saturation blocks) can be organized by the programmer under a loop node which is responsible for specifying the order in which the blocks are played out during the scan. One example of a loop node is a Cartesian loop, which iterates over a set of phase encoding positions, calling a child node for imaging at each step. One or more loop objects are organized together under a single root node to form a sequence. The most standard sequence objects are built in to the software, and the advanced user can create custom objects using arbitrary C or C++ code.

The user interface displays the current sequence as an expandable tree (see **Figure 2**). Using the mouse, the programmer can insert standard or customized nodes into the sequence tree, and then browse the various sequence objects. The parameter pane shows those sequence parameters associated with the current node (for example TE and TR are parameters of a typical imaging block). When the user modifies any of these parameters, the simulation pane is instantly updated to reflect the modified sequence. Thus the user is able to visualize the sequence at all stages of development.

To maintain modularity in the sequence functionality, the logical interaction between nodes is minimized, allowing the user to modify selected portions of the sequence tree (such as changing an excitation pulse or switching from a Cartesian to radial loop) without affecting other parts of the tree. This also allows for copying and pasting of sequence objects. For example, a custom RF pulse, a specialized imaging block, or even an entire loop from one sequence can easily be incorporated into an entirely different sequence via copy and paste.

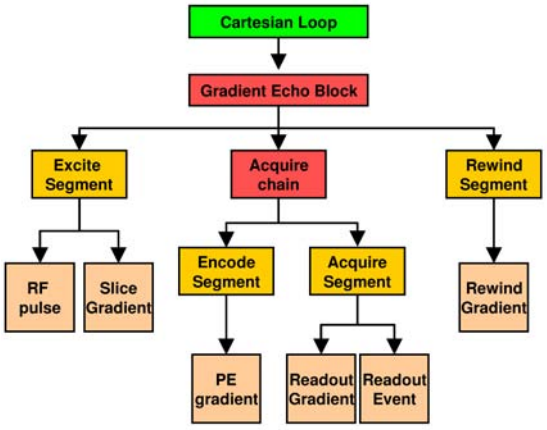

Figure 1: The tree structure of a typical SequenceTree pulse sequence. The lowest level nodes are the scanner events. The basic user controls only the top level nodes such as the loop or block objects, but the advanced user can create custom nodes at any level.

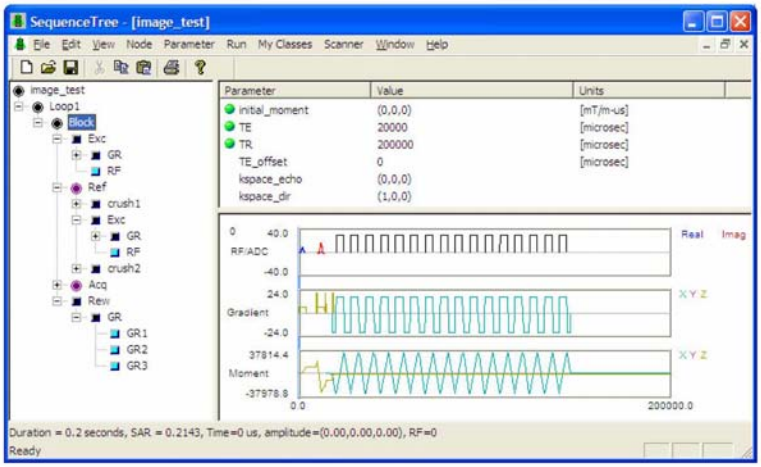

Figure 2: Screenshot of the SequenceTree software implementing the Interleaved Multiple Gradient Echo (IMGE) pulse sequence [2]. The sequence is displayed as an expandable tree (left pane). When the sequence parameters are modified (upper right pane), the simulation pane is updated in real time (lower right pane).

Results

Several dozen pulse sequences have been programmed with *SequenceTree* by students and researchers in the authors' laboratory. The sequences ranged in complexity from basic 3D gradient-echo and spin-echo imaging sequences to 3D hybrid radial [1] and a spectroscopic imaging sequence with 16 interleaved readouts per scan [2]. Sequences which used projection reconstruction, saturation pulses, phase cycling and inversion recovery were also programmed. A computer program was used to automatically convert the SequenceTree sequences into pulse programs compatible with Siemens scanners. The sequences were run on a Siemens Sonata 1.5T scanner and the raw data was processed offline. Progress has also been made interfacing SequenceTree to the Bruker DMX 400 microimaging system. All sequences implemented produced the expected images after reconstruction. Each pulse sequence was programmed within a matter of hours over the course of one or two days.

Conclusion

One of the obstacles to the development of novel NMR pulse sequences is the technical difficulty of pulse sequence implementation. SequenceTree is a visual programming environment which offers flexibility in a user-friendly environment. It spurs development of new unconventional sequences because it dramatically reduces the programmer's time and effort. Furthermore, it opens the MRI scanner to those researchers with little or no computer programming experience, and offers the advantage of a visual display of the sequence at every stage of pulse sequence development.

References:

[1] Magland et al. Proc ISMRM 13th Scientific Meeting (2005).

[2] Hilaire et al. Magn Reson Imag 18:777-786 (2000).

Acknowledgement: NIH Grant T32 EB000814