The Journal of Open Source Software

SigPy.RF: A Python Toolkit for MRI RF Pulse Design

Jonathan B. Martin[1], **Heng Sun**[2], **Madison Albert**[3], **Jonathan I. Tamir**[4], **Frank Ong**[5], **and William A. Grissom**[6¶]

1 Vanderbilt University Institute of Imaging Science, Vanderbilt University Medical Center, Nashville, TN, USA **2** Department of Biomedical Engineering, Yale University, New Haven, CT, USA **3** Department of Biomedical Engineering, Vanderbilt University, Nashville, TN, USA **4** Department of Electical and Computer Engineering, University of Texas at Austin, Austin, TX, USA **5** Roblox Corporation, San Mateo, CA, USA **6** Department of Biomedical Engineering, Case Western Reserve University, Cleveland, OH, USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

We present SigPy.RF, an extensive set of open-source, Python-based tools for magnetic resonance imaging (MRI) radiofrequency (RF) pulse design. This toolbox extends the SigPy Python software package and leverages SigPy's existing capabilities for GPU computation, iterative optimization, and powerful abstractions for linear operators, proximal operators, and applications. Tools are available for all steps of the excitation design process including trajectory/gradient design, pulse design, and simulation. Our implemented functions for pulse design include advanced Shinnar-LeRoux (SLR), multiband, adiabatic, optimal control, B_1^+-selective and small-tip parallel transmission (pTx) designers.

Statement of need

The field of magnetic resonance imaging is currently experiencing rapid growth in open source imaging tools. Tools have been made freely available for MRI hardware development (Amrein et al., 2022; Anand, 2018), system simulation (Stöcker et al., 2010; Villena et al., 2014), pulse sequence programming (Layton et al., 2017), image reconstruction (Ong & Lustig, 2019; Uecker et al., 2015), and post-processing and analysis (Avants et al., 2014; Duval et al., 2018). However, one critical step of the imaging pipeline which has seen limited open-source tool development is RF pulse design. While RF pulse developers increasingly share code online in independent repositories, no unified set of common pulse design tools has been maintained in a rigorous and consistent manner with easy-to-read code and tutorials. This is despite the reality that in many cases, carefully designed or application-specific RF pulses are crucial to the success of MRI techniques. An open source pulse design code library would facilitate the development and dissemination of novel techniques and the comparison of approaches, similar to how BART (Uecker et al., 2015) and SigPy (Ong & Lustig, 2019) have made advanced parallel imaging and reconstruction methods widely accessible. To meet this need, we have developed a library of pulse design tools as part of the SigPy Python package for signal processing and image reconstruction (Ong & Lustig, 2019). We call this new package SigPy.RF. SigPy.RF is constructed as a nested package within the broader SigPy package.Figure 1 illustrates where the RF pulse design tools fit into the broader SigPy software, as a nested package within the package for general MRI tools.

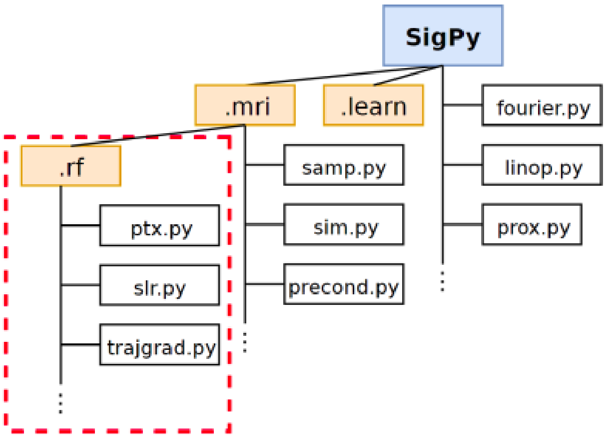

Figure 1: New RF pulse design tools (red) within the SigPy package hierarchy. The pulse design tools are grouped in a series of modules dedicated to a specific pulse class or type of RF pulse design utility.

About SigPy.RF

SigPy has a number of features that make it an ideal candidate to support a nested RF pulse design package. Image reconstruction and RF pulse design require many of the same operators and computational tools; for example, SENSE reconstruction (Pruessmann et al., n.d.) and small-tip parallel transmit pulse design (Grissom et al., 2006) are in many cases parallel processes moving in opposite directions between the spatial and frequency domains, with the same or similar linear operators and the solved with the same iterative optimization tools. Thus critical RF pulse design operations such as Fourier transforms, gradient-based iterative optimizers, and matrix manipulation methods are provided in SigPy (Ong & Lustig, 2019). SigPy was further built with a number of computational features that make it an ideal fit for RF pulse design. As a Python library, SigPy in general should not be expected to match the performance of comparable C libraries such as BART (Uecker et al., 2015). However, SigPy uses Numba (Lam et al., 2015) to translate many of its most commonly used tools, such as gridding functions, to optimized machine code at runtime. Optimizations that are dominated by FFTs, such as small-tip spatial domain pTx pulse designs, can closely match the performance of C toolboxes as the same low-level C FFT libraries are used (Ong & Lustig, 2019). Additionally, SigPy has a general unified CPU and GPU interface for most functions, allowing for easy movement of data between devices for computational flexibility and acceleration of computation with GPUs. Figure 2 shows the mean computation times for a small-tip-angle spatial domain parallel transmit pulse design (Grissom et al., 2006) on CPU and GPU. Design time was approximately an order of magnitude faster across all problem dimensions when the GPU was used versus CPU.

The Journal of Open Source Software

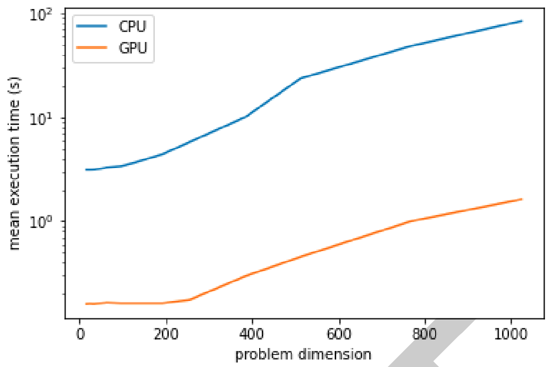

Figure 2: Mean execution time for small-tip spatial domain pulse design using SigPy linear operators and CuPy matrices across all considered problem dimensions, on CPU and GPU

Target Audience

The SigPy.RF toolbox has been developed for use by MRI researchers focusing on pulse sequence design, MRI physics, signal processing, and optimization. We believe that it will serve as an essential building block for more general image acquisition tools which require specialized RF pulses. The toolbox has already been incorporated into open-source sequence development software such as Pulseq (Layton et al., 2017) and PyPulseq (Sravan Ravi et al., 2019) to provide RF pulses critical to the performance of various pulse sequences. An example is shown in Figure 3. Finally, end-to-end optimization of MRI pulse sequences and reconstructions is being increasingly explored (Radhakrishna & Ciuciu, 2023; Wang et al., 2022); with the RF pulse and gradient waveform design functions provided by SigPy.RF, the SigPy package could facilitate this research.

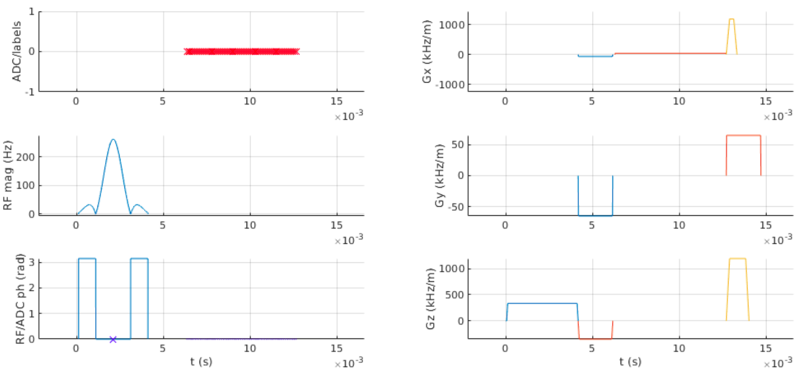

Figure 3: First TR of a GRE pulse sequence created in Pulseq (Layton et al., 2017) with SigPy.RF-designed SLR excitation pulse. Magnitude of the TB = 4 , 90 degree SLR pulse is plotted in the 'RF mag' plot, middle left.

We also envision SigPy.RF may be utilized for replicability and reproducibility studies, allowing for standardized conventions for RF pulses to be used across studies. The package could also serve as a hands-on teaching aid for researchers and students. We have developed several tutorials, which are accessible to a wide audience with minimal prior MRI knowledge.

Availability and Use

The latest version of SigPy includes the latest stable release of the pulse design tools and is available from the main repository. It can be installed through conda or pip- see the documentation for more details. The SigPy.RF fork includes pulse design tools that are still in progress, prior to being merged into the main codebase. Jupyter notebook pulse design tutorials for SigPy.RF are also available.

Preliminary development of this toolbox was presented in reference (J. Martin et al., 2020); the current version has been streamlined and expanded to include a larger collection of RF pulse design methods from the literature, as well as additional utility tools for I/O, gradient waveform design, and experimental B_1^+-selective pulse design algorithms which were prototyped using the initial SigPy.RF codebase, enabling the publication of Reference (J. B. Martin et al., 2022).

Acknowledgements

We gratefully acknowledge the support this study received from NIH Grant R01 EB 016695.

References

Amrein, P., Jia, F., Zaitsev, M., & Littin, S. (2022). CoilGen: Open-source MR coil layout generator. *Magnetic Resonance in Medicine*, *88*(3), 1465–1479. https://doi.org/10.1002/MRM.29294

Anand, S. M.). (2018). *OCRA : a low-cost, open-source FPGA-based MRI console capable of real-time control*. https://dspace.mit.edu/handle/1721.1/121619

Avants, B. B., Tustison, N., & Johnson, H. (2014). *Advanced Normalization Tools (ANTS)*.

Duval, T., Leppert, I. R., Cabana, J.-F., Boudreau, M., Gagnon, I., Berestovoy, G., Cohen-Adad, J., & Stikov, N. (2018). Quantitative MRI made easy with qMRLab. *Proc. Intl. Soc. Mag. Reson. Med.*, 2288.

Grissom, W., Yip, C. Y., Zhang, Z., Stenger, V. A., Fessler, J. A., & Noll, D. C. (2006). Spatial domain method for the design of RF pulses in multicoil parallel excitation. *Magnetic Resonance in Medicine*, *56*(3), 620–629. https://doi.org/10.1002/MRM.20978

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based Python JIT Compiler. *Proceedings of LLVM-HPC 2015: 2nd Workshop on the LLVM Compiler Infrastructure in HPC - Held in Conjunction with SC 2015: The International Conference for High Performance Computing, Networking, Storage and Analysis, 2015-Janua*. https://doi.org/10.1145/2833157.2833162

Layton, K. J., Kroboth, S., Jia, F., Littin, S., Yu, H., Leupold, J., Nielsen, J. F., Stöcker, T., & Zaitsev, M. (2017). Pulseq: A rapid and hardware-independent pulse sequence prototyping framework. *Magnetic Resonance in Medicine*, *77*(4), 1544–1552. https://doi.org/10.1002/MRM.26235

Martin, J. B., Abitha Srinivas, S., Vaughn, C. E., Sun, H., Griswold, M. A., Grissom, W. A., & Jonathan Martin, C. B. (2022). Selective excitation localized by the Bloch–Siegert shift and a B1+

$$B_1^+$$

gradient. *Magnetic Resonance in Medicine*. https://doi.org/10.1002/MRM.29271

Martin, J., Ong, F., Ma, J., Tamir, J., Lustig, M., & Grissom, W. (2020). SigPy.RF: Comprehensive Open-Source RF Pulse Design Tools for Reproducible Research. *Proc. Intl. Soc. Mag. Reson. Med.*, 1045.

Ong, F., & Lustig, M. (2019). SigPy: A Python Package for High Performance Iterative Recon- struction. *Proc. Intl. Soc. Mag. Reson. Med.*, 4819.

Pruessmann, K. P., Weiger, M., Scheidegger, M. B., & Boesiger, P. (n.d.). *SENSE: Sensitivity Encoding for Fast MRI*. https://doi.org/10.1002/(SICI)1522-2594(199911)42:5

Radhakrishna, C. G., & Ciuciu, P. (2023). Jointly Learning Non-Cartesian k-Space Trajectories and Reconstruction Networks for 2D and 3D MR Imaging through Projection. *Bioengineering*, *10*(2). https://doi.org/10.3390/BIOENGINEERING10020158

Sravan Ravi, K., Geethanath, S., & Thomas Vaughan Jr, J. (2019). PyPulseq: A Python Package for MRI Pulse Sequence Design. *Journal of Open Source Software*, *4*(42), 1725. https://doi.org/10.21105/JOSS.01725

Stöcker, T., Vahedipour, K., Pflugfelder, D., & Shah, N. J. (2010). High-performance computing MRI simulations. *Magnetic Resonance in Medicine*, *64*(1), 186–193. https://doi.org/10.1002/MRM.22406

Uecker, M., Ong, F., Tamir, J. I., Bahri, D., Virtue, P., Cheng, J. Y., Zhang, T., & Lustig, M. (2015). Berkeley Advanced Reconstruction Toolbox. *Proc. Intl. Soc. Mag. Reson. Med.*, 2486.

Villena, J. F., Polimeridis, A. G., Serrales, J. E. C., Wald, L. L., Adalsteinsson, E., White, J., & Daniel, L. (2014). MARIE a MATLAB-based open source software for the fast electromagnetic analysis of MRI systems. *Proc. Intl. Soc. Mag. Reson. Med.*, 0709.

Wang, G., Luo, T., Nielsen, J. F., Noll, D. C., & Fessler, J. A. (2022). B-Spline Parameterized Joint Optimization of Reconstruction and K-Space Trajectories (BJORK) for Accelerated 2D MRI. *IEEE Transactions on Medical Imaging*, *41*(9), 2318. https://doi.org/10.1109/TMI.2022.3161875