The Journal of Open Source Software

PyPulseq: A Python Package for MRI Pulse Sequence Design

Keerthi Sravan Ravi[1], **Sairam Geethanath**[1], **and John Thomas Vaughan Jr.**[1]

1 Columbia Magnetic Resonance Research Center, Columbia University in the City of New York, USA

DOI: 10.21105/joss.01725

Software

- Review
- Repository
- Archive

Submitted: 21 August 2019
Published: 12 October 2019

Summary

Magnetic Resonance Imaging (MRI) is a critical component of healthcare. MRI data is acquired by playing a series of radio-frequency and magnetic field gradient pulses. Designing these pulse sequences requires knowledge of specific programming environments depending on the vendor hardware (generations) and software (revisions) intended for implementation. This impedes the pace of prototyping. Pulseq (Layton et al., 2017) introduced an open source file standard for pulse sequences that can be deployed on Siemens/GE via TOPPE (Nielsen & Noll, 2018)/Bruker platforms. In this work, we introduce `PyPulseq`, which enables pulse sequence programming in Python. Its advantages are zero licensing fees and easy integration with deep learning methods developed in Python. `PyPulseq` is aimed at MRI researchers, faculty, students, and other allied field researchers such as those in neuroscience. We have leveraged this tool for several published research works (Poojar, Geethanath, Reddy, & Venkatesan, n.d.; Ravi et al., 2018a, 2018b, 2019a, 2019b; Tong et al., 2019).

Statement of need

MRI is a non-invasive diagnostic imaging tool. It is a critical component of healthcare and has a significant impact on diagnosis and treatment assessment. Structural, functional and metabolic MRI generate valuable information that aid in the accurate diagnosis of a wide range of pathologies. A unique strength of MRI is the ability to visualise diverse pathologies achieved by the flexibility in designing tailored pulse sequences. MRI pulse sequences are a collection of radio-frequency and gradient waveforms that are executed on the scanner hardware to acquire raw data.

Research efforts on pulse sequence design are directed at achieving faster scan times, improving tissue contrast and increasing Signal-to-Noise Ratio (SNR). However, designing pulse sequences requires knowledge of specific programming environments depending on the vendor hardware (generations) and software (revisions) intended for implementation. Typically, MRI researchers program and simulate the pulse sequences on computers and execute them on MRI scanners. This typically involves considerable effort, impeding the pace of prototyping and therefore research and development. This also hampers multi-site multi-vendor studies as it requires researchers to be acquainted with each vendor's programming environment. Furthermore, harmonizing acquisition across MRI vendors will enable reproducible research. This work introduces an open source tool that enables pulse sequence programming for Siemens/GE/Bruker platforms in Python, based on the Pulseq standard (Layton et al., 2017).

Introduction to the Pulseq file format: `.seq`

The `.seq` file format introduced in Pulseq (Layton et al., 2017) is a novel way to capture a pulse sequence as plain text. The file format was designed with the following design criteria in mind: human-readable, easily parsable, vendor independent, compact and low-level (Layton et al., 2017). A pulse sequence comprises of radiofrequency pulses, magnetic field gradient waveforms, delays or analog-to-digital converter (ADC) readout *events*. A *block* comprises of one or more *events* occurring simultaneously. *Event* envelopes are defined by *shapes*, which are run-length encoded and stored in the `.seq` file. In a `.seq` file, each *event* and *shape* is identified uniquely by an integer. *Blocks* are constructed by assembling the uniquely referenced *events*. Therefore, any custom pulse sequence can be synthesised by concatenating *blocks*.

About `PyPulseq`

The `PyPulseq` package presented in this work is an open source vendor-neutral MRI pulse sequence design tool. It enables researchers and users to program pulse sequences in Python, and export them as a `.seq` file. These `.seq` files can be executed on the three MRI vendors by leveraging vendor-specific interpreters. The MRI methods have been reported previously (Ravi, Potdar, et al., 2018). The `PyPulseq` package allows for both representing and deploying custom sequences. This work focuses on the software aspect of the tool. `PyPulseq` was entirely developed in Python, and this has multiple advantages. Firstly, unlike existing C++ frameworks such as ODIN (Jochimsen & Von Mengershausen, 2004) and SequenceTree (Magland, Li, Langham, & Wehrli, 2016), `PyPulseq` does not require any compilation of the pulse sequence scripts. Secondly, it does not involve any licensing fees that are otherwise associated with other scientific research platforms such as MATLAB. Thirdly, there has been a proliferation of deep learning projects developed in Python in recent years. These advantages allow `PyPulseq` to be integrated with projects related to various stages of the MRI pipeline. For example - deep learning techniques for acquisition (intelligent slice planning in Ravi et al. (2018a)) and related downstream reconstruction. Finally, the standard Python package manager - PyPI - enables convenient installs on multiple OS platforms. These Python-derived benefits ensure that `PyPulseq` can reach a wider audience.

We have leveraged the `PyPulseq` library to implement acquisition oriented components of the Autonomous MRI (AMRI) package (Ravi et al., 2018a, 2019a, 2019b), Virtual Scanner (Tong et al., 2019), and the non-Cartesian acquisition library (Ravi et al., 2018b). Also, the `PyPulseq-gpi` branch integrates a previous version of `PyPulseq` with GPI to enable GUI-based pulse sequence design. This work has been previously reported (Ravi, Potdar, et al., 2018) and is not within the scope of this JOSS submission. Currently, `PyPulseq` does not support external triggers and interactive slice planning. Raw data acquired with pulse sequences designed with `PyPulseq` cannot be reconstructed vendor-supplied tools. `PyPulseq` is a translation of Pulseq from MATLAB (Layton et al., 2017).

Target audience

`PyPulseq` is aimed at MRI researchers focusing on pulse sequence design, image reconstruction, and MRI physics. We also envisage PyPulseq to be utilized for replicability and reproducibility studies such as those for functional MRI (multi-site, multi-vendor). The package could also serve as a hands-on teaching aid for MRI faculty and students. Beginners can get started with the bundled example pulse sequences. More familiar users can import the appropriate packages to construct and deploy custom pulse sequences.

The Journal of Open Source Software

Acknowledgements

This study was funded (in part) by the 'MR Technology Development Grant' and the 'Seed Grant Program for MR Studies' of the Zuckerman Mind Brain Behavior Institute at Columbia University (PI: Geethanath).

References

Jochimsen, T. H., & Von Mengershausen, M. (2004). ODIN—object-oriented development interface for nmr. *Journal of Magnetic Resonance*, *170*(1), 67–78. doi:10.1016/j.jmr.2004.05.021

Layton, K. J., Kroboth, S., Jia, F., Littin, S., Yu, H., Leupold, J., Nielsen, J.-F., et al. (2017). Pulseq: A rapid and hardware-independent pulse sequence prototyping framework. *Magnetic resonance in medicine*, *77*(4), 1544–1552. doi:10.1002/mrm.26235

Magland, J. F., Li, C., Langham, M. C., & Wehrli, F. W. (2016). Pulse sequence programming in a dynamic visual environment: SequenceTree. *Magnetic resonance in medicine*, *75*(1), 257–265. doi:10.1002/mrm.25640

Nielsen, J.-F., & Noll, D. C. (2018). TOPPE: A framework for rapid prototyping of mr pulse sequences. *Magnetic resonance in medicine*, *79*(6), 3128–3134. doi:10.1002/mrm.26990

Poojar, P., Geethanath, S., Reddy, A. K., & Venkatesan, R. (n.d.). Rapid prOtotyping of 2D non-cartesian k-space trajEcTories (rocket) using pulseq and gpi. *Critical Reviews™ in Biomedical Engineering*. doi:10.1615/CritRevBiomedEng.2019029380

Ravi, K. S., Geethanath, S., Jochen, W., & Vaughan, J. T. (2018a). MR value driven autonomous mri using imr-framework. In *ISMRM workshop on machine learning part ii*, Washington, D.C.

Ravi, K. S., Geethanath, S., & Vaughan, J. T. (2018b). Imr-framework for rapid design and deployment of non-cartesian sequences. In *I2i workshop*. Retrieved from cai2r.net/i2i

Ravi, K. S., Geethanath, S., & Vaughan, J. T. (2019a). Autonomous scanning using imr-framework to improve mr accessibility. In *ISMRM workshop on accessible mri for the world*. New Delhi, India.

Ravi, K. S., Geethanath, S., & Vaughan, J. T. (2019b). Self-administered exam using autonomous magnetic resonance imaging (amri). In *ISMRM 27th annual meeting and exhibition*. Montreal, Canada.

Ravi, K. S., Potdar, S., Poojar, P., Reddy, A. K., Kroboth, S., Nielsen, J.-F., Zaitsev, M., et al. (2018). Pulseq-graphical programming interface: Open source visual environment for prototyping pulse sequences and integrated magnetic resonance imaging algorithm development. *Magnetic resonance imaging*, *52*, 9–15. doi:10.1016/j.mri.2018.03.008

Tong, G., Geethanath, S., Qian, E., Ravi, K. S., Jimeno Manso, M., & Vaughan, J. T. (2019). Virtual mr scanner software. In *ISMRM 27th annual meeting and exhibition*. Montreal, Canada.