

Simulation and Harmonic Analysis of k -Space Readout (SHAKER)

John C. Bodenschatz*

and

Daniel B. Rowe

Department of Mathematical and Statistical Sciences, Marquette University

March 17, 2026

Abstract

In the realm of neuroimaging research, the demand for efficient and accurate simulation tools for functional magnetic resonance imaging (fMRI) data is ever increasing. We present SHAKER, a comprehensive MATLAB package for simulating complex-valued fMRI time series data that will advance understanding and implementation of the MR signal equation and related physics principles to fMRI simulation. The core objective of the package is to provide researchers with a user-friendly MATLAB graphical user interface (GUI) tool capable of generating complex-valued fMRI time series data. This tool will allow researchers to input various parameters related to the MRI scan and receive simulated k -space data with ease, facilitating a deeper understanding of the intricacies of the generation and interpretation of fMRI data.

Keywords: fMRI, MATLAB, Simulator

*The authors would like to thank the generosity of the Wehr Foundation and the Northwestern Mutual Data Science Institute for partially funding this work.

1 Introduction

Functional magnetic resonance imaging (fMRI) is a non-invasive imaging technique that allows trained physicians and scientists to observe functionality of organs, in particular the human brain. This is done by exciting protons in the various molecules that make up the different tissues of the organ, then determining a net change in magnetization as determined by an induced current in a loop of wire surrounding the patient. This net magnetization in different voxels of the region of interest (ROI) is associated with complex-valued spatial frequencies that fill an array of k -space; a high order approximation of the Fourier transform of the voxel image of the organ. The k -space array is then inverse discrete Fourier transformed (IDFT) to reconstruct an image. Figure 1a shows a simple MRI machine and the major axes, Figure 1b depicts an example of the magnitude of a measured complex-valued k -space array, and Figure 1c shows the magnitude of the complex-valued image reconstructed from complex-valued k -space, the magnitude of which is in Figure 1b. To perform experiments in the machine is both financially and temporally costly; demanding machine time and obtaining Institutional Review Board (IRB) approvals can slow down the process of investigating new statistical techniques to extract information from fMRI data.

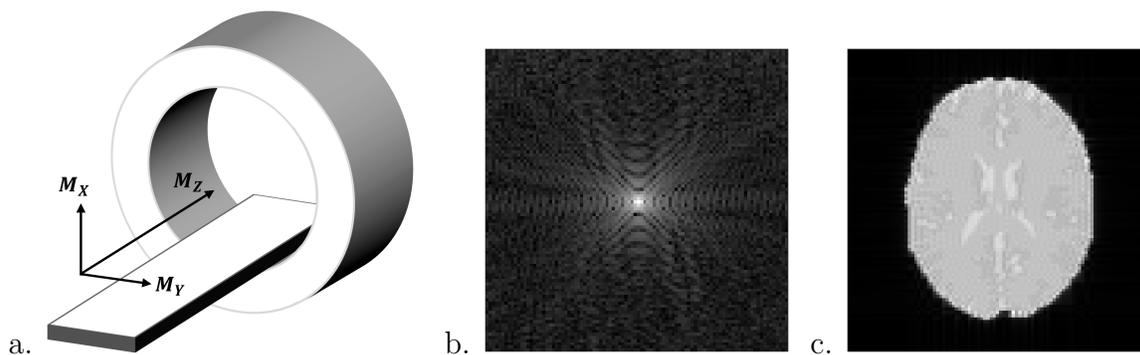


Figure 1: The process of obtaining an image from the machine. a) MRI machine with main axes indicated; the z -direction is referred to as longitudinal, the xy -plane is referred to as the transverse plane. b) Acquired k -space array of spatial frequencies. c) Inverse discrete Fourier transform of k -space; a reconstructed image.

Consequently, researchers will test developing methods on simulated data as a cost-effective way of measuring potential. Currently, simulated fMRI data are largely developed in-house for each researcher using a variety of methods. There has been work to develop a more standardized method to simulate fMRI time series data using various languages such as Python and R. However, many such methods disregard the complex-valued nature along with the true statistical and physical properties of the data output by the machine, in addition to returning magnitude-only images from simulations (Welvaert et al. 2011). Some of these methods may also demand some form of in-line coding or require external files to support simulation (Comby et al. 2024). It will be beneficial to provide a complete software tool to researchers that allows the simulation of complex-valued fMRI time series data with the ability to tune various parameters relating to the scan to match future experimental data, that will allow for proper testing of developing models. We present the current work on such an fMRI simulation software tool titled *Simulation and Harmonic Analysis of k-Space Readout* (SHAKER). SHAKER, a GUI simulator built on the MATLAB programming language, is built on the physics-based principles of the MRI machine and is designed so that both new and well-versed researchers in the field can simulate data with ease. The sections that follow will give a brief overview of the physics being applied in the simulator, followed by an in-depth description of each of the parts of the simulator. This will be examined in an example simulation study at the end.

1.1 Nuclear magnetic resonance

A primary aim of SHAKER is to provide a realistic simulation of fMRI measurements. It is important to understand the physical principles and phenomena that determine the measured signal which is later reconstructed into an image. The MR machine creates a very strong magnetic field B_0 along the z direction of the scanner as indicated in Figure 1a (1.5, 3, and 7 Tesla are common). This field aligns the spins of hydrogen nuclei within

the body to become parallel and anti-parallel with the direction of the scanner. A slightly higher number of spins align parallel to the field, resulting in a net magnetization M_0 which is measured in imaging. These hydrogen nuclei precess (resonate) at the Larmor frequency which is proportional to the external magnetic field they are exposed to,

$$f_0 = \gamma B_0, \tag{1}$$

where γ is the gyromagnetic ratio, a constant unique to each nucleus ([Larmor 1897](#)). In the case of hydrogen, we have $\gamma = 42.58 \text{MHz}/T$. To excite these nuclei, a radio frequency (RF) burst of energy is sent into the system at this resonant frequency. The nuclei enter a higher energy state where their spins tip against the main magnetic field B_0 at some flip angle α determined by the length of the RF pulse. An $\alpha = 90^\circ$ or less flip angle is common for fMRI. In the time that follows the RF pulse, these nuclei emit energy through two relaxation processes- T_1 and T_2 . The longitudinal or spin-lattice relaxation time, T_1 , is the recovery time for the parallel component of M_0 , M_Z , back to equilibrium. The transverse or spin-spin relaxation time, T_2 , is the decay of M_{XY} , the transverse component of M_0 to zero. In practice, T_2^* is what is actually measured. The relationship between T_2 and T_2^* is defined by

$$1/T_2^* = 1/T_2 + 1/T_2', \tag{2}$$

where $1/T_2' = \gamma \Delta B$ is the dephasing of the hydrogen nuclei as a result from hydrogen nuclei precessing at slightly different frequencies due to inhomogeneities in the magnetic field, ΔB . The two effects, T_1 and T_2^* are visualized in [Figure 2b-c](#). [Figure 2a](#) shows the net magnetization change, a vector sum of the T_1 and T_2^* relaxivities. These relaxivities result in a changing magnetic field within the tissue that is measured through current via Faraday's law of induction in one or more coils of wire that are placed locally to the region of interest. This measured signal is then later transformed into complex-valued images via the inverse

discrete Fourier transform.

In fMRI, the blood-oxygen-level-dependent (BOLD) signal is interrogated to determine regions of activation (Ogawa et al. 1990, Bandettini et al. 1992, Kwong et al. 1992). The BOLD signal is a measure of localized brain blood level and oxygenation changes which are correlates for neuronal activity. These changes occur as a result of certain stimuli or tasks, e.g., right-hand finger tapping, that activate known regions of the brain. The BOLD signal presents itself as a T_2^* effect since the change in magnetic properties of oxygenated and deoxygenated hemoglobin in blood causes a perturbation in the local magnetic field, ΔB . Hence, fMRI time series are T_2^* -weighted images, highlighting regions of the brain with significant T_2^* effects.

Every image from an MRI machine comes from a predetermined “pulse sequence” of RF bursts and changing of magnetic gradients within the machine. In fMRI, images are most often collected via single shot echo planar imaging (EPI); “single shot” meaning only one RF excitation is applied per k -space array. The most commonly used pulse sequence used in EPI is gradient echo (GRE) (Kumar et al. 1975, Bernstein et al. 2004). In general, a given signal equation yields complex signal s received at a given point (k_x, k_y) in k -space.

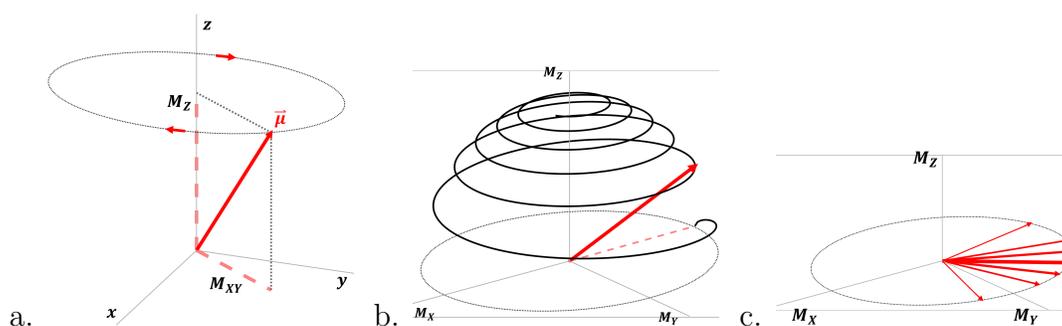


Figure 2: Depicted in (a) is the net magnetization and precession of the magnetic moment about the central axis. Representations of the T_1 (b) and T_2^* (c) relaxivities. T_1 is relaxation back into the longitudinal direction, T_2^* is relaxation in the transverse direction.

The GRE signal equation is given by

$$s(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{M_0 \sin(\alpha)}{(1 - \cos(\alpha)e^{-TR/T_1})} \left(1 - e^{-TR/T_1}\right) e^{-t/T_2^*} e^{i\gamma\Delta B t} e^{-i2\pi(k_x x + k_y y)} dx dy, \quad (3)$$

where $M_0(x, y)$, $T_1(x, y)$, $T_2^*(x, y)$, and $\Delta B(x, y)$ are functions of locations (x, y) within the physical object (or phantom) and $t(k_x, k_y)$ is the time at which the point (k_x, k_y) in k -space is scanned. The simplification of replacing $t(k_x, k_y)$ with TE , echo time, is often used and is equivalent to assuming that all data are acquired simultaneously at the TE . SHAKER does not impose this assumption. The repetition time, TR , is the time between successive RF pulses of the same slice, or equivalently, the time between successive measured k -space arrays of the same slice. The flip angle α is commonly set to 90° , which simplifies the first term in Equation 3 to just M_0 . While GRE is most commonly used because of the high signal it provides, some higher strength scanners (7 T+) may opt to use the spin echo (SE) pulse sequence to detect BOLD signal (Chen & Glover 2015, Nencka & Rowe 2005). The SE signal equation is given by

$$s(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} M_0 \left(1 - e^{-TR/T_1}\right) e^{-t/T_2} e^{i\gamma\Delta B t} e^{-i2\pi(k_x x + k_y y)} dx dy, \quad (4)$$

noting the use of T_2 instead of T_2^* . It has been shown that SE pulse sequences correct for the large scale dephasing caused by larger veins, which may not be as closely related to activation as capillaries (Kida et al. 2000). Closely related to the SE pulse sequence, but not generally used for fMRI experiments, is the inversion recovery (IR) pulse sequence. The signal equation for IR is given by

$$s(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} M_0 \left(1 - 2e^{-TI/T_1} + e^{-TR/T_1}\right) e^{i\gamma\Delta B t} e^{-i2\pi(k_x x + k_y y)} dx dy, \quad (5)$$

where TI is the inversion time. The IR pulse sequence is more commonly used for T_1 -

weighted images, as compared to the T_2^* -weighted image that is standard in fMRI, but is still included in SHAKER. Both the SE and IR signal equations assume a 90° flip angle from the initial RF pulse. Other pulse sequences such as diffusion weighted imaging (DWI-fMRI), and saturation recovery (SR) may be included in the future plans for SHAKER development.

1.2 k -Space and the Fourier transform

The signal equations from Section 1.1 are measured in the spatial-frequency domain called k -space (magnitude images of k -space are presented in Figure 3). Each of Equations 3-5 could be condensed to

$$s(k_x, k_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho_0(x, y) e^{-i2\pi(k_x x + k_y y)} dx dy. \quad (6)$$

In this form, we can see that the signal equation is the Fourier transform of ρ_0 , the net magnetization after having been weighted by the relevant relaxivities. In practice, however, k -space is only measured at a finite set of discrete points. So, we can discretize Equation 6 into

$$s(k_x, k_y) = \frac{1}{N_x N_y} \sum_{m=0}^{N_x-1} \sum_{n=0}^{N_y-1} \rho_0(x, y) e^{-i2\pi\left(\frac{k_x}{N_x} x_m + \frac{k_y}{N_y} y_n\right)}, \quad (7)$$

where N_x and N_y are the number of points in image-space in the x and y directions, respectively. In fMRI, it is common that $N_x = N_y \in \{64, 96, 128\}$. Thus, we arrive at k -space equating to the *discrete* Fourier transform of image-space.

The objects and phantoms being imaged are composed of real-valued voxels only. So, ideally, the Fourier transform of these objects would result in a k -space that maintains Hermitian symmetry. However due to the terms involving T_2^* and ΔB , the expected Hermitian symmetry of k -space is broken. In fact, when looking at Equations 3-5, the inclusion of these terms necessarily implies that k -space is only a very close *approximation* to the

Fourier transform of image-space, since the terms are time-dependent. This leads to possible distortions and artifacts when reconstructing k -space into images using the inverse discrete Fourier transform

$$\rho_0(x, y) = \frac{1}{N_{k_x} N_{k_y}} \sum_{m=0}^{N_{k_x}-1} \sum_{n=0}^{N_{k_y}-1} s(k_x, k_y) e^{-i2\pi \left(\frac{x}{N_{k_x}} k_{xm} + \frac{y}{N_{k_y}} k_{yn} \right)}, \quad (8)$$

however it is still the most common method of image reconstruction in MRI. The incorporation of prior knowledge regarding the relaxivities as well as magnetic field inhomogeneity has been implemented to enhance image reconstruction ([Karaman et al. 2015](#)).

As a result of physical limitations, k -space must be scanned, or traversed, in one continuous path. The most conventional method is to scan horizontal rows, often referred to as the frequency-encoding direction, in alternating directions working up (or down) k -space in the phase-encoding direction as shown in Figure 3a. This involves a set of “turnaround points” at the end of each row that are often discarded or not measured, resulting in dead scan time that decreases the rate of useful data acquisition. This has proven to be a convenient way to scan k -space as it results in a Cartesian encoding of the spatial frequencies which allows for the simple inverse discrete Fourier transform to reconstruct k -space back into an image. Other k -space trajectories, including non-Cartesian methods such as PROPELLER, radial

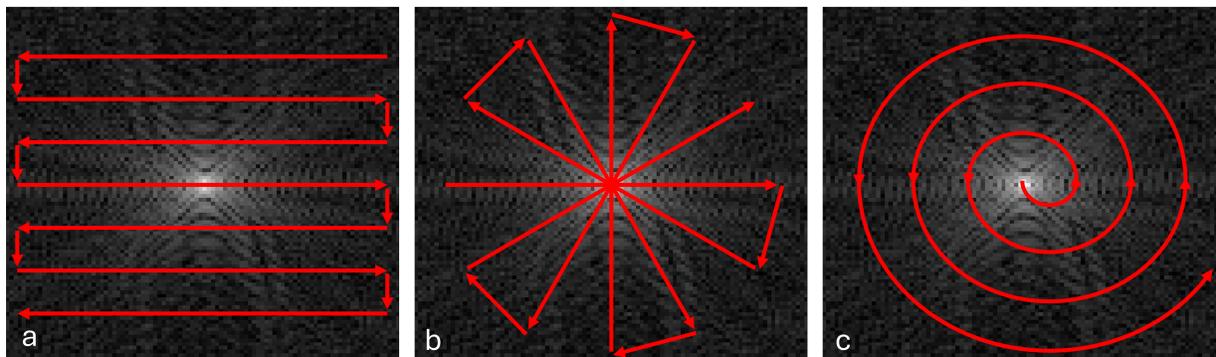


Figure 3: Three examples of k -space trajectory. a) Standard Cartesian encoding. b) Radial encoding. c) Spiral encoding.

as in Figure 3b (equivalent to PROPELLER with blade width 1), and spiral as in Figure 3c have been implemented for various reasons such as reducing scan time and increasing robustness to artifacts due to motion (Pipe 1999, Block et al. 2014).

2 Design

SHAKER is an all-inclusive fMRI simulation software package built with the user in mind. SHAKER is built using the MATLAB programming language and presents as a GUI (Figure 4), with no scripting or external data required (The MathWorks Inc. 2022). In the top left pane of Figure 4, users can view the pre-loaded digital phantom. The bottom left pane involved customization of MRI parameters and fMRI experimental design options. The

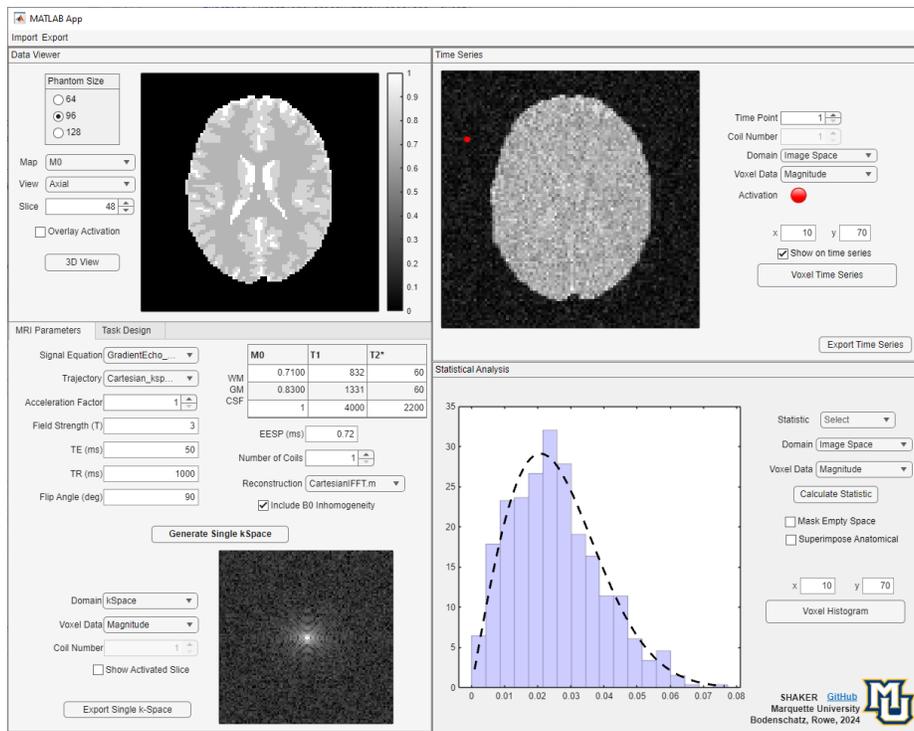


Figure 4: A screenshot of the working SHAKER GUI. In the top left pane, users can view the raw phantom data that will be input to the signal equation. In the bottom left pane are the adjustable MRI parameters and the fMRI experimental setup. The right pane displays two views: the top presents unaltered data from the simulated time series, and the bottom reflects an example of a statistical data set created from the time series data (in this case: a histogram of one voxel’s magnitude time series).

top right pane presents a view of the simulated time series data. The bottom right pane is where statistical maps and measurements from the time series data can be observed. This is also where any models in development may be tested on the simulated data. All code and data used to operate SHAKER are publicly available on GitHub to encourage a better understanding and allow customizations to be made. The contents of this section explain in detail the functionality of SHAKER in each of these panes.

2.1 Digital phantom

SHAKER comes pre-loaded with a full volume digital phantom that was simulated with realistic M_0 , T_1 , and T_2^* values based on a 3 T machine (Karaman 2016). The tissues included in the phantom are gray matter (GM), white matter (WM), and cerebrospinal fluid (CSF). The ΔB map was considered as a gradient along each of the dimensions of the scanner, combined with some biological detail from the T_2^* map. This included digital phantom is stored as a MATLAB structure array: `Phantom: M0, T1, T2, deltaB`, and users may also load in their own maps.

An axial slice of the phantom is shown in Figure 5. In some cases, a higher or lower sampling density requirement is needed in k -space. To support this, the digital phantom can be rescaled to $64 \times 64 \times 64$ or $128 \times 128 \times 128$ by changing the *Phantom Size* option. Should one

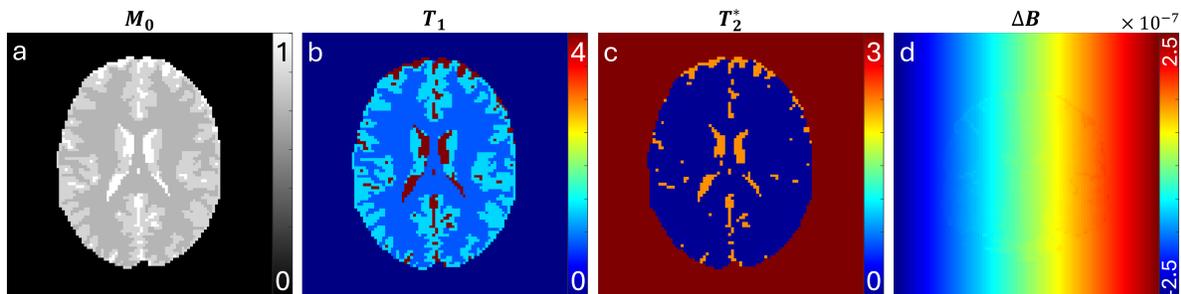


Figure 5: Maps taken from an axial slice of the digital phantom. a) Net Magnetization, M_0 (dimensionless). b) Longitudinal relaxation, T_1 (seconds). c) Transverse relaxation, T_2^* (seconds). d) Field inhomogeneity, ΔB (Tesla).

want to implement their own digital phantom, it will be necessary to create a structure with the same naming convention, having four maps whose dimensions all agree. This custom phantom can then be imported as a `.mat` file from the toolbar located at the top of the GUI. Additionally, there is an activation map included with SHAKER that carries the same dimension as the phantom. The activation map is a binary array, with ones only at the intended location(s) of simulated activation. It is designed such that it roughly resembles the left primary motor cortex region of the brain- the area that is expected to be active during right-hand finger tapping. Custom activation maps can also be imported as `.mat` files with type `double` and name `ActMap` and should share dimensions with the phantom being used for simulation.

At present, SHAKER is equipped to handle single-slice excitations in any of the three major planes: axial, sagittal, or coronal. Support for echo-volume imaging could be supported in future versions. Slice selection and orientation are both chosen and viewed in the top left pane of the GUI, titled *Data Viewer* as in Figure 6. The size of the phantom is also adjusted from this pane. Two other options that have no effect on the simulation: viewing each of the maps and visualizing where the activation is expected, are available from this panel as well. Making use of the recently developed `volshow()` function in MATLAB, users can get a 3-D view of the four maps that make up the phantom, sliced at the indicated location and orientation.

2.2 MRI parameters

Found in the bottom left pane of the GUI are tabs for *MRI Parameters* and *Task Design* as shown in Figures 7 and 9. *MRI Parameters* is the part of SHAKER where users will make selections similar to that of an MRI technician. *Task Design* will be discussed in Section 2.3. All settings in this pane should be confirmed before initializing any simulations. From

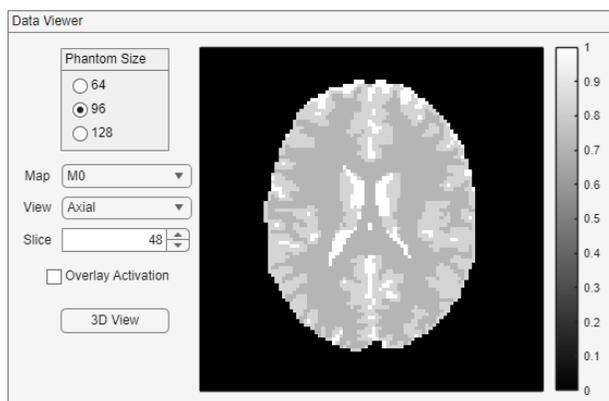


Figure 6: *Data Viewer*: the top left pane of SHAKER. Here the phantom size can be selected, each of the four maps can be viewed, and slice orientation/location can be selected.

the top toolbar of SHAKER, the MRI structure being used for simulation can be saved into a `.mat` file which contains a MATLAB structure array named `MRI`. This file may be imported to future instances of SHAKER for ease of reproducibility of results.

The first two options for MRI parameters are *Signal Equations* and *Trajectory*. These refer to a choice of signal equation as described in Section 1.1 and a k -space trajectory from Section 1.2. The k -space trajectory functions are designed to receive the MRI object inherent

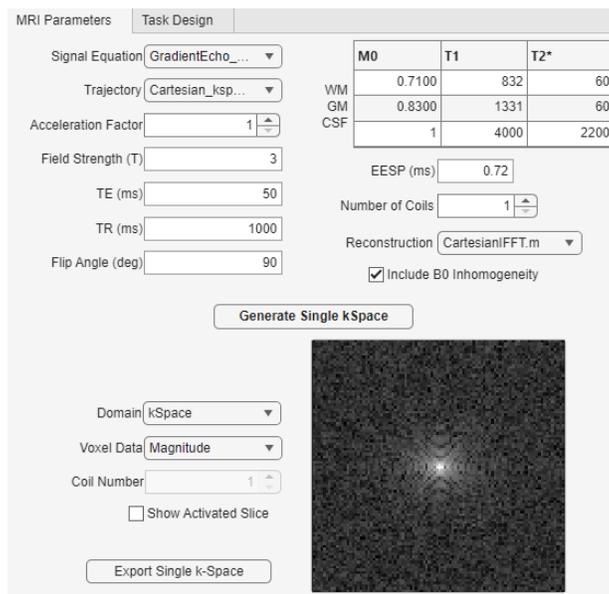


Figure 7: *MRI Parameters*: the first tab of the bottom left pane of SHAKER. Here relevant MRI parameters can be set. A single k -space can be simulated and observed to check that settings are correct before simulating an entire time series.

to SHAKER that contains all pertinent information regarding the scanner properties. The trajectory functions then return three arrays: one each for the k_x and k_y locations at which k -space is sampled as well as an array noting the time at which the points are sampled $t(k_x, k_y)$. These arrays are stored in an object within SHAKER for later reference. Following this, the user can select their choice of signal equation function, which receives input data about the phantom, k -space sampling, and the MRI object, then return a simulated array of k -space measurements. Both the signal equation and k -space trajectory are two files that users can create their own version of, using the templates provided by SHAKER, to sample k -space in their own preferred way. Further details regarding inputs/outputs can be found in the appendix.

The next option is *Acceleration Factor*. This can mean different things depending on the context of the k -space sampling method. For example, in Cartesian trajectories of k -space an acceleration factor of n_a is often implemented as a measurement of every n_a lines in the frequency-encode direction. In the single-spoke radial trajectory of k -space, this is commonly the measurement of every n_a spokes. There is no restriction on how this might be implemented in one's own k -space trajectory file. Since SHAKER currently supports single-slice imaging, this acceleration factor should be interpreted as an *in-plane acceleration* (IPA). Following this is the choice to change the simulated magnetic field strength. The values found in the table located at the top right of Figure 7 can be altered to produce an effect on other k -space features at the will of the user. The four options that follow, *TE*, *TR*, *Flip Angle*, and *EESP* are direct inputs to the signal equation as described in Section 1.1.

Parallel imaging in fMRI has received a lot of attention recently due to its ability to accelerate the rate at which images are acquired in fMRI experiments (Pruessmann et al. 1999, Griswold et al. 2002). In practice, each coil measures a sensitivity-weighted image

of the brain, or phantom, at no additional temporal cost. SHAKER supports the use of a single, uniform coil, or multiple coils aligned with the bore of the machine. Users may specify any number of coils to simulate their data by changing the value for the *Number of Coils* option. The n_c coils have sensitivity matrices that match the dimensions of the phantom. In SHAKER, each of the n_c coil sensitivities is constructed by placing a point at a each coil location, all of which are equidistant from the center of image-space and angularly equidistant from each other. The sensitivity of each coil array then decreases proportional to the inverse of the distance from this point. An example of the simulated coils, coil sensitivity weighted images, and averaged image is shown in Figure 8.

The last option is the choice of *Reconstruction Algorithm*. While SHAKER is a k -space simulation tool, the reconstruction of images is supported for the more common k -space trajectories. Similar to the k -space trajectory and signal equation, the choice of reconstruction algorithm can be user-created based on an included template. These templates can be found in the subdirectories for each of the respective steps in the simulation process. More detail on this can be found in the appendix.

2.3 FMRI experimental design and noise

Task-based fMRI generally starts with an initial set of rest images that allow the tissue to reach a steady state in the magnetic field (Elster et al. 2001). Following this, many epochs of rest / task images are taken. For example, an experiment may include: 16 initial rest images followed by 19 epochs of 16 task images followed by 16 rest images for a total of 624 images. It is often the case that some or all of the initial rest images are discarded for fMRI analysis due to the fact that they yield a higher signal than the remaining steady state images. This can be circumvented by increasing the flip angle for the first few images so that the amount of transverse magnetization excited in each image is approximately the

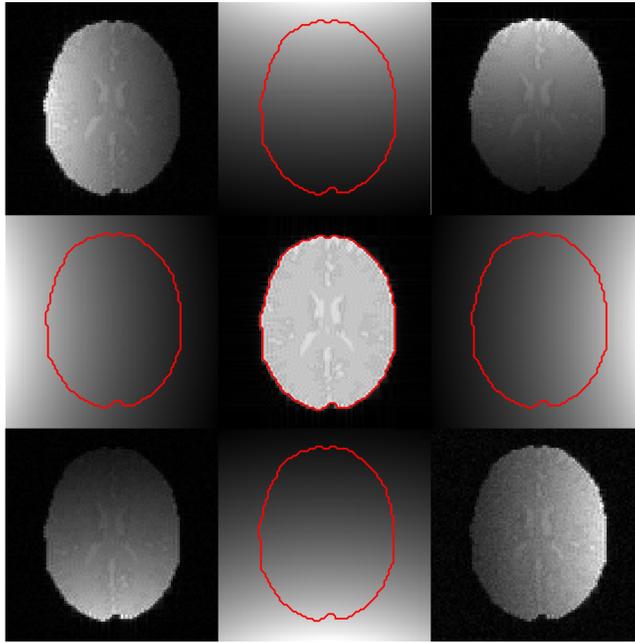


Figure 8: Coil sensitivities (top, right, bottom, left) and sensitivity weighted images (corners) for the case of $n_c = 4$. The center image is the true, unweighted slice image. The red outlines indicate the location of the slice as seen by each of the coil sensitivities.

same (Haase et al. 1986). For this example, discarding the first 16 from analysis would give $n_{IMG} = 608$ images in the fMRI time series. It has been shown, however, that the first few images can be used to aid in analysis of the measured fMRI data, e.g., T_1 map estimation. In SHAKER, users may choose a set number of initial rest images, the number of epochs, and number of rest/task images per epoch. This is then stored as a design vector that can be used for later analysis of the simulated time series. This is all done from the second tab of the bottom left pane of SHAKER, as in Figure 9.

Complex-valued measurements k_c in fMRI are composed of both a real and imaginary part, $k_c = k_R + ik_I$ where $i = \sqrt{-1}$. The measured magnitude r_k and phase ϕ_k of the voxels come from the transformation $k_R = r_k \cos(\phi_k)$ and $k_I = r_k \sin(\phi_k)$. To better model the process of the machine, SHAKER adds noise to k -space directly rather than to reconstructed images as is often done. Since the analog-to-digital converters (ADCs) sample k -space measurements independently, it is understood that the real and imaginary parts of k -space measurements

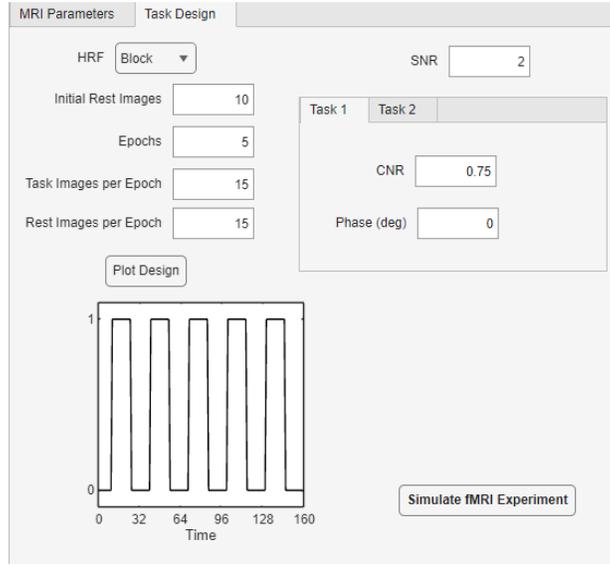


Figure 9: *Task Design*: the second tab of the bottom left pane of SHAKER. This is where users may adjust the experimental design of the time series, as well as specify SNR and CNR .

are independent and identically distributed (*iid*) normally for each spatial frequency. Thus, the joint distribution is given as

$$f(k_R, k_I) = \frac{1}{(2\pi\sigma_k^2)^{1/2}} \exp\left[-\frac{(k_R - \mu_{k,R})^2}{2\sigma_k^2}\right] \frac{1}{(2\pi\sigma_k^2)^{1/2}} \exp\left[-\frac{(k_I - \mu_{k,I})^2}{2\sigma_k^2}\right], \quad (9)$$

where $\mu_{k,R}$ and $\mu_{k,I}$ are the *true* real and imaginary components of the spatial frequency (Henkelman 1985, Lindquist 2008). The inverse Fourier transform of the real and imaginary components of the noise from k -space into image space will also be normally distributed with a scaled variance. This relationship is given by

$$\sigma^2 = \frac{\sigma_k^2}{n_x n_y}, \quad (10)$$

where σ^2 is the variance of the normally distributed real/imaginary noise in image space, σ_k^2 is the variance of the normally distributed real/imaginary noise in k -space, and n_x, n_y are the dimensions of reconstructed image space (Rowe 2016). This fact reveals that the joint distribution of a voxels real and imaginary parts in image-space can be written similarly to

Equation 9 as

$$f(y_R, y_I) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left[-\frac{(y_R - \mu_R)^2}{2\sigma^2}\right] \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left[-\frac{(y_I - \mu_I)^2}{2\sigma^2}\right]. \quad (11)$$

The true real and imaginary components, μ_r and μ_I , can be expressed in terms of the true magnitude and phase, ρ and θ , by the transformation $\mu_R = \rho \cos(\theta)$ and $\mu_I = \rho \sin(\theta)$. Since the magnitude of voxels is preferred over the real/imaginary values when looking at an image, we can transform the measured random variables (y_R, y_I) to (r, ϕ) where $y_R = r \cos(\phi)$ and $y_I = r \sin(\phi)$ (Rowe 2023). Calculating the Jacobian to be $J = r$, this gives the joint distribution

$$f(r, \phi) = \frac{r}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2} [r^2 + \rho^2 - 2r\rho \cos(\phi - \theta)]\right). \quad (12)$$

By integrating out ϕ from Equation 12, we get a Ricean marginal distribution for the voxel's magnitude r (Rice 1944, Gudbjartsson & Patz 1995, Rowe 2005, Adrian et al. 2013),

$$f(r) = \frac{r}{\sigma^2} \exp\left[-\frac{r^2 + \rho^2}{2\sigma^2}\right] I_0\left(\frac{r\rho}{\sigma^2}\right). \quad (13)$$

Here I_0 is the zeroth order modified Bessel function of the first kind. The mean of the Ricean distribution is $\sigma\sqrt{\pi/2}L_{1/2}(-\rho^2/2\sigma^2)$ where $L_{1/2}$ is a Laguerre polynomial. The variance of the Ricean distribution, denoted as σ_r^2 , has the following relationship with the variance of the real and imaginary components of voxels in image space, σ^2 ,

$$\sigma_r^2 = 2\sigma^2 + \rho^2 - \frac{\pi\sigma^2}{2}L_{1/2}^2(-\rho^2/2\sigma^2). \quad (14)$$

The subscript r is used to indicate the observed magnitude. In regions of empty space where the true signal ρ is small, $\rho \approx 0$, this is reduced to the Rayleigh distribution with

mean $\sigma\sqrt{\frac{\pi}{2}}$ and variance $\frac{4-\pi}{2}\sigma^2$ (Rayleigh 1880). In regions of space with high true signal ρ , this becomes the normal distribution with mean ρ and variance σ^2 . Integrating out the magnitude r from the joint distribution in Equation 12 gives the unnamed non-normal distribution marginal distribution for the phase ϕ ,

$$f(\phi) = \frac{1}{2\pi} \exp\left[-\frac{\rho^2}{2\sigma^2}\right] \left[1 + \frac{\rho}{\sigma}\sqrt{2\pi} \cos(\phi - \theta) \exp\left[\frac{\rho^2 \cos^2(\phi - \theta)}{2\sigma^2}\right] \Phi\left(\frac{\rho \cos(\phi - \theta)}{\sigma}\right)\right], \quad (15)$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. When the signal ρ is near zero, the phase will be uniformly distributed on $[-\pi, \pi]$ with mean 0 and variance $\frac{\pi^2}{3}$. When the signal ρ becomes large, the distribution of the phase becomes normal with mean θ and variance $\frac{\sigma^2}{\rho^2}$.

Task-based fMRI for an individual voxel's noiseless magnitude time series ρ_t can be expressed as the linear equation

$$\rho_t = \beta_0 + \beta_1 x_t. \quad (16)$$

Here, $\beta_0 \in \mathbf{R}^+$ is the baseline signal which determines the signal-to-noise ratio $SNR = \beta_0/\sigma_r$, and $\beta_1 \in \mathbf{R}$ is the task-related signal change which determines the contrast-to-noise ratio $CNR = \beta_1/\sigma_r$. The design vector $x_t \in \{0, 1\}^{n_t}$ has length equal to the number of reconstructed images in the time series, n_t . In x , indices corresponding to a non-task image have an element of 0, while indices corresponding to a task-active image have an element of 1; this is equivalent to a block design hemodynamic response function (HRF). At present, SHAKER supports block design HRFs only. Future development considerations include the option to convolve user-defined functions with the block design to create custom HRFs.

When multiple coils are introduced to support parallel imaging, the statistical properties of the data are altered. Considering C receiver coils, the composite magnitude signal is given by $M_C = \sqrt{\sum_{j=1}^C [M_{jR}^2 + M_{jI}^2]}$ where $M_{jR} = \rho_{jC} \cos(\theta_{jC})$ and $M_{jI} = \rho_{jC} \sin(\theta_{jC})$ denote

the real and imaginary signals reconstructed from the j^{th} receiver coil. It has been shown that the probability density of this composite magnitude is the non-central Chi distribution, which can be written as:

$$f(M_C) = \frac{\rho_C}{\sigma^2} \left(\frac{M_C}{\rho_C} \right)^C \exp \left(-\frac{(\rho_C^2 + M_C^2)}{2\sigma^2} \right) I_{C-1} \left(\frac{M_C \rho_C}{\sigma^2} \right) \quad (17)$$

where ρ_C is the true signal magnitude and I_k is the k^{th} order modified Bessel function (Koay & Bassar 2006). The mean of this non-central Chi distribution is given by

$$\mu_{M_C} = \sigma \beta_C {}_1F_1(-1/2, C, -\rho_C/(2\sigma^2))$$

where $\beta_C = \sqrt{\pi/2} \frac{(2C-1)}{2^{C-1}(C-1)!}$ and ${}_1F_1$ is the confluent hypergeometric function (Koay et al. 2009). The variance is given by

$$\sigma_{M_C} = \xi(\rho_C|\sigma, C)\sigma^2,$$

where ξ is a scaling factor given by

$$\xi(\rho_C|\sigma, C) = 2C + \frac{\rho_C^2}{\sigma^2} - \left[\beta_C {}_1F_1(-1/2, C, -\rho_C/(2\sigma^2)) \right]^2.$$

When designing an fMRI experiment in SHAKER, users may specify an *SNR* and *CNR* which will be used to calculate the baseline signal β_0 and magnitude signal increase β_1 , respectively¹. In addition to specifying *SNR* and *CNR*, users may choose to specify some amount of task-related-phase-change (TRPC) in degrees. It has been shown that there is biological information contained in the phase of an image, and as such it may be desirable

¹While not previously mentioned, now is a good time to acknowledge that SHAKER should be used with caution if studying quantitative MRI, as the final values have been through at least one layer of scaling and it may be difficult to extract quantitative meaning.

to consider it in statistical models (Rowe 2005, Rowe et al. 2007). In SHAKER, magnitude activation is determined through the CNR , and an additional phase angle in the activated areas can be specified. Once all MRI parameters and fMRI options are confirmed, users may select the *Generate Time Series* button found in the *Task Design* tab to initialize the simulation.

2.4 Time series data analysis

The top right panel of SHAKER as in Figure 10, labeled as *Time Series*, allows for visualization and examination of the simulated time series data. Users may look through individual images in the time series, in either k -space or image-space (if reconstructed), viewing the real, imaginary, magnitude, or phase parts of an image. There is also the option to monitor the time series of individual voxels which can be helpful to determine regions of activation and activation structure (magnitude/phase).

Below this, in the bottom right panel of SHAKER as in Figure 11, labeled *Statistical Analysis*, is where statistical maps and models of the simulated time series may be assessed. True to the MRI machine, SHAKER supports analysis of complex-valued data. Users may closely examine the real and imaginary part of an image, or magnitude and phase part

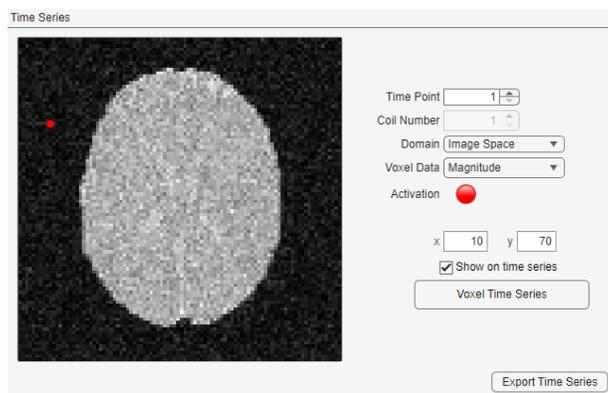


Figure 10: *Time Series*: the top right pane of SHAKER. Here the simulated time series can be observed and checked for expected results.

of image. This can be done in both k -space and reconstructed image-space. SHAKER comes preloaded with two simple statistical measures: a voxel-wise t test for activation detection and an SNR calculation to evaluate image fidelity (Ardekani & Kanno 1998). The t test compares the task-active images in a simulated time series to the mean rest image in order to determine some change in magnitude or phase over some ROI. This is what is expected with the BOLD signal due to the increased T_2^* effect. The SNR calculation estimates the SNR of each voxel throughout the time series. This helps determine the quality of k -space trajectory and reconstruction method by highlighting any regions of leakage (signal leaking between previously aliased voxels) or other artifacts. Statistical maps can be superimposed onto an anatomical image of the excited slice for better viewing of the activated regions or other ROIs. Additionally, there is the option to look at the histogram of any voxel’s magnitude/phase/real/imaginary component throughout the time series with theoretical probability density functions (PDF) overlain. This may be used to confirm expected distributions of voxel’s time series as described in Section 2.3. As described in the appendix, SHAKER supports the use of custom statistical methods and models to analyze the simulated data. At present it is recommended that for advanced models the data be exported and examined in a more controlled environment.

3 Example fMRI experiment

This section will carry out an example simulated fMRI experiment, detailing all steps taken in SHAKER. Suppose that a right-hand finger tapping experiment was performed on a subject. The 3T machine is set to scan an axial slice in the center of the brain. The data matrix size is 96×96 , with $TE = 60.4ms$, $EESP = 0.832ms$, $TR = 1s$, and no acceleration factor ($n_a = 1$). The experimental timing starts with an initial 16s of rest followed by 19 epochs of 16s of task alternating with 16s of rest for a total of $n_{IMG} = 624$ images.

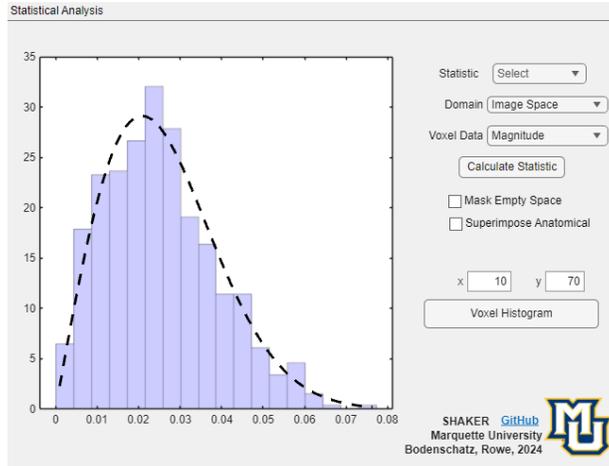


Figure 11: *Statistical Analysis*: the bottom right pane of SHAKER. This is where users may analyze statistical properties from the data, such as: t statistic for activation, SNR maps, voxel histograms, etc.

Firstly, the phantom size and slice orientation and number would be set. This is done in the top left pane of SHAKER by setting the options **Phantom Size: 96**, **View: Axial**, and **Slice: 48**. Following this, the MRI parameters should be set. Without further knowledge of the experiment, it may be safe to assume that a GRE pulse sequence is used and measured along the standard Cartesian trajectory. So the options **Signal Equation: GradientEcho_SigEq.m** and **Trajectory: Cartesian_kspace.m** should be selected. Following this, the acceleration factor, field strength, TE , TR , and $EESP$ can be input directly from the experimental setup data. It can be assumed that the flip angle is $\alpha = 90^\circ$ (this is not always the case experimentally, but, unless other information is known, is a reasonable simplification). For simplicity it can also be assumed that the machine is using a single, uniform coil. Since the k -space trajectory is the standard Cartesian path, images can be reconstructed by setting **Reconstruction: CartesianIFFT.m**. To better represent the machine, the box for B_0 inhomogeneity may be checked to include the ΔB effect into the simulation.

In the *Task Design* tab next to *MRI Parameters*, the options for the fMRI experimental design can be set. The HRF can be set to block and the four values that follow: initial

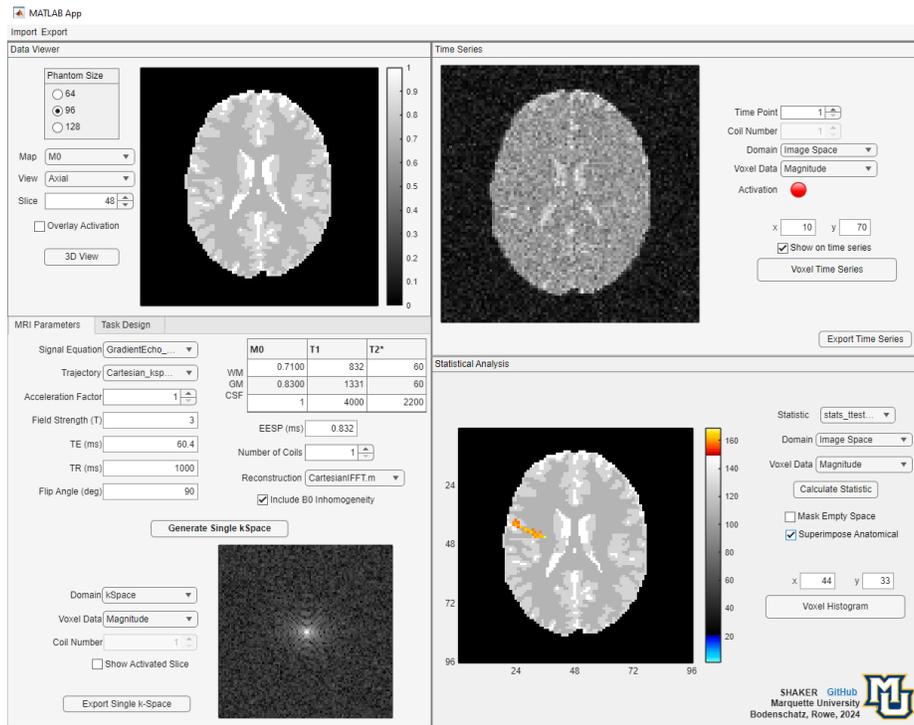


Figure 12: Screenshot of the input to SHAKER for the example simulation described in Section 3. Not pictured: the fMRI time series and SNR/CNR options set in the second tab of the bottom left panel.

rest images, epochs, and task/rest images per epoch, can be filled in directly from the experimental setup. The *Plot Design* button can be pressed to visualize and ensure the experimental timing is setup correctly. To be consistent with empirical data it is recommended that the SNR is set somewhere in the range of $[1, 10]$ and the CNR is set somewhere between $[0.1, 2]$. For this example, the two are set to be $SNR: 5$ and $CNR: 0.5$. There will be no phase activation added in this example, so $Phase: 0$. Once all inputs are confirmed to be correct, the time series is simulated by pressing the *Generate Time Series Data* button.

Once the simulation is complete, a summary of simulation will be displayed and the *Time Series* panel will be populated with data. This summary can be copied and pasted into a paper for publishing. From here, the images can be observed and sorted through to check for any errors in simulation. A time series of a voxel in an active region may be generated to

observe any noticeable patterns. The data may then be analyzed using custom or built-in tools. This can be done for task detection, noise analysis, etc. The results for this example are shown in Figure 12. The summary of simulation for this example reads:

The following fMRI time series data was simulated on 11-Nov-2024 at 17:07:23. The simulated time series is of slice 48 from a size 96 phantom in the Axial plane. The MRI parameters were set to be the following: Acceleration Factor = 1, Field Strength = 3T, TE = 60.4ms, TR = 1000ms, Flip Angle = 90deg, EESP = 0.832ms, and Number of Coils = 1. The data was simulated with the GradientEcho_SigEq.m signal equation using the Cartesian_kspace.m k-space trajectory. The experimental design involved an initial 16 rest images followed by 19 epochs, each consisting of 16 task images followed by 16 rest images for a total of 624 images. The SNR was set to 5 and the CNR was set to 0.5. There were 0 degrees of phase added to the activation. Images were reconstructed using the CartesianIFFT.m algorithm.

4 Discussion and future work

SHAKER is a one-stop shop for fMRI simulation. The GUI-based approach to the simulator allows for specially simulated data to be placed quickly into the hands of researchers as compared to long wait times for experimental data. The physics-based approach entrusts that the simulated data is representative of what a proper MRI machine might theoretically output. The most recent version of SHAKER can be found on GitHub at the following URL: <https://github.com/bodensjc/SHAKER>. Due to the public availability of SHAKER, it can remain in a constant state of development as users contribute ideas and needs for the simulation tool. All thoughts and considerations for future development are asked to be sent to the corresponding author.

Future consideration for this work involve the addition of more MRI features and parameters as well as additional fMRI experimental design components. Control of Field-of-View (FOV) and bandwidth could be helpful for generating zoomed images. More control of the noise generation through temporal variation as well as spatial. The inclusion of intra- and inter-imaging motion for fMRI experiments would help test registration and motion correction algorithms. Standard reconstruction methods such as SENSE and GRAPPA can be implemented to facilitate accelerated parallel imaging ([Pruessmann et al. 1999](#), [Griswold et al. 2002](#)). Proprietary non-uniform inverse Fourier transforms may also be implemented to facilitate the reconstruction of non-Cartesian based k -space trajectories. Simultaneous multi-slice (SMS) techniques such as CAIPIRINHA ([Breuer et al. 2005](#), [Xu & Rowe 2025](#), [Karaman 2016](#)) could be supported in the future as well as other full- or partial-volume imaging methods. Additional support for the processing of archival data can be added, including techniques such as: Nyquist ghost correction, zero-filling, apodization, motion correction, static B_0 correction, etc.

CRedit authorship contribution statement

John C. Bodenschatz: Writing - original draft, Visualization, Software, Investigation, Formal analysis. **Daniel B. Rowe:** Writing - review & editing, Supervision, Resources, Project administration, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

None declared.

References

- Adrian, D. W., Maitra, R. & Rowe, D. B. (2013), ‘Rician over Gaussian Modelling in Magnitude fMRI Analysis-Added Complexity with Negligible Practical Benefits’, *Stat* **2**(1), 303–316.
- Ardekani, B. A. & Kanno, I. (1998), ‘Statistical Methods for Detecting Activated Regions in Functional MRI of the Brain’, *Magnetic Resonance Imaging* **16**(10), 1217–1225.
URL: <https://www.sciencedirect.com/science/article/pii/S0730725X98001258>
- Bandettini, P. A., Wong, E. C., Hinks, R. S., Tikofsky, R. S. & Hyde, J. S. (1992), ‘Time Course EPI of Human Brain Function During Task Activation’, *Magnetic Resonance in Medicine* **25**(2), 390–397.
- Bernstein, M., King, K. & Zhou, X. (2004), *Handbook of MRI Pulse Sequences*, Elsevier Inc. Publisher Copyright: © 2004 Elsevier Inc. All rights reserved.
- Block, K., Chandarana, H., Milla, S., Bruno, M., Mulholland, T., Fatterpekar, G., Hagiwara, M., Grimm, R., Geppert, C., Kiefer, B. & Sodickson, D. (2014), ‘Towards Routine Clinical Use of Radial Stack-of-Stars 3D Gradient-Echo Sequences for Reducing Motion Sensitivity’, *Journal of the Korean Society of Magnetic Resonance in Medicine* **18**, 87.
- Breuer, F. A., Blaimer, M., Heidemann, R. M., Mueller, M. F., Griswold, M. A. & Jakob, P. M. (2005), ‘Controlled Aliasing in Parallel Imaging Results in Higher Acceleration (CAIPIRINHA) for Multi-Slice Imaging’, *Magnetic Resonance in Medicine* **53**(3), 684–691.
- Chen, J. E. & Glover, G. H. (2015), ‘Functional Magnetic Resonance Imaging Methods’, *Neuropsychology review* **25**(3), 289–313.
- Comby, P.-A., Vignaud, A. & Ciuciu, P. (2024), ‘SNAKE-fMRI: A Modular fMRI Data Simulator from the Space-Time Domain to k-Space and Back’, *arXiv* .
URL: <https://arxiv.org/abs/2404.08282>

- Elster, A. D., Burdette, J. H. & Field, A. (2001), *Questions & answers in magnetic resonance imaging*, second edition. edn, Mosby, St. Louis.
- Griswold, M. A., Jakob, P. M., Heidemann, R. M., Nittka, M., Jellus, V., Wang, J., Kiefer, B. & Haase, A. (2002), ‘Generalized Autocalibrating Partially Parallel Acquisitions (GRAPPA)’, *Magnetic Resonance in Medicine* **47**(6), 1202–1210.
- Gudbjartsson, H. & Patz, S. (1995), ‘The Rician Distribution of Noisy MRI Data’, *Magnetic Resonance in Medicine* **34**(6), 910–914.
- Haase, A., Frahm, J., Matthaei, D., Hanicke, W. & Merboldt, K.-D. (1986), ‘FLASH Imaging. Rapid NMR Imaging Using Low Flip-Angle Pulses’, *Journal of Magnetic Resonance (1969)* **67**(2), 258–266.
URL: <https://www.sciencedirect.com/science/article/pii/0022236486904336>
- Henkelman, R. M. (1985), ‘Measurement of Signal Intensities in the Presence of Noise in MR Images’, *Medical Physics* **12**(2), 232–233.
URL: <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.595711>
- Karaman, M. M. (2016), Improving fMRI Analysis and MR Reconstruction With the Incorporation of MR Relaxivities and Correlation Effect Examination, PhD thesis, Marquette University.
- Karaman, M. M., Bruce, I. P. & Rowe, D. B. (2015), ‘Incorporating relaxivities to more accurately reconstruct MR images’, *Magnetic Resonance Imaging* **33**(4), 374–384.
URL: <https://www.sciencedirect.com/science/article/pii/S0730725X15000041>
- Kida, I., Kennan, R. P., Rothman, D. L., Behar, K. L. & Hyder, F. (2000), ‘High-Resolution CMRO₂ Mapping in Rat Cortex: A Multiparametric Approach to Calibration of BOLD Image Contrast at 7 Tesla’, *Journal of Cerebral Blood Flow & Metabolism* **20**(5), 847–860. PMID: 10826536.

Koay, C. G. & Basser, P. J. (2006), ‘Analytically Exact Correction Scheme for Signal Extraction From Noisy Magnitude MR Signals.’, *Journal of magnetic resonance* **179** **2**, 317–22.

URL: <https://api.semanticscholar.org/CorpusID:17979879>

Koay, C. G., Özarlan, E. & Basser, P. J. (2009), ‘A Signal Transformational Framework for Breaking the Noise Floor and its Applications in MRI’, *Journal of Magnetic Resonance* **197**(2), 108–119.

Kumar, A., Welti, D. & Ernst, R. R. (1975), ‘NMR Fourier Zeugmatography’, *Journal of Magnetic Resonance* **18**(1), 69–83.

Kwong, K. K., Belliveau, J. W., Chesler, D. A., Goldberg, I. E., Weisskoff, R. M., Poncelet, B. P., Kennedy, D. N., Hoppel, B. E., Cohen, M. S. & Turner, R. (1992), ‘Dynamic Magnetic Resonance Imaging of Human Brain Activity During Primary Sensory Stimulation.’, *Proceedings of the National Academy of Sciences* **89**(12), 5675–5679.

Larmor, J. (1897), ‘A Dynamical Theory of the Electric and Luminiferous Medium.— Part III. Eelations With Material media’, *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **190**, 205–300.

URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1897.0020>

Lindquist, M. A. (2008), ‘The Statistical Analysis of fMRI Data’, *Statistical Science* **23**(4), 439 – 464.

URL: <https://doi.org/10.1214/09-STS282>

Nencka, A. S. & Rowe, D. B. (2005), ‘Complex Constant Phase Activation Model Removes Venous BOLD Contribution in fMRI’.

Ogawa, S., Lee, T. M., Kay, A. R. & Tank, D. W. (1990), ‘Brain Magnetic Resonance

- Imaging with Contrast Dependent on Blood Oxygenation’, *Proceedings of the National Academy of Sciences U. S. A.* **87**(24), 9868–9872.
- Pipe, J. G. (1999), ‘Motion Correction with PROPELLER MRI: Application to Head Motion and Free-Breathing Cardiac Imaging’, *Magnetic Resonance in Medicine* **42**(5), 963–969.
- Pruessmann, K. P., Weiger, M., Scheidegger, M. B. & Boesiger, P. (1999), ‘SENSE: Sensitivity Encoding for Fast MRI’, *Magnetic Resonance in Medicine* **42**(5), 952–962.
- Rayleigh, J. (1880), ‘On the Resultant of a Large Number of Vibrations of the Same Pitch and of Arbitrary Phase’, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **10**(60), 73–78.
- Rice, S. O. (1944), ‘Mathematical Analysis of Random Noise’, *Bell System Technical Journal* **23**(3), 282–332.
- Rowe, D. B. (2005), ‘Modeling Both Magnitude and Phase of Complex fMRI Data’, *NeuroImage* **25**, 1310–24.
- Rowe, D. B. (2016), *Image Reconstruction in Functional MRI*, first edition. edn, CRC Press.
URL: <https://doi.org/10.1201/9781315373652>
- Rowe, D. B. (2023), ‘Statistics of Intrinsic FMRI Data’, *Proceedings of the Joint Statistical Meeting* .
- Rowe, D. B., Meller, C. P. & Hoffmann, R. G. (2007), ‘Characterizing Phase-Only fMRI Data with an Angular Regression Model’, *Journal of Neuroscience Methods* **161**(2), 331–341.
- The MathWorks Inc. (2022), ‘MATLAB version: 24.1.0 (R2024a)’.
URL: <https://www.mathworks.com>
- Welvaert, M., Durnez, J., Moerkerke, B., Berdoolaege, G. & Rosseel, Y. (2011), ‘neuRosim:

An R Package for Generating fMRI Data', *Journal of Statistical Software* **44**(10), 1–18.

URL: <https://www.jstatsoft.org/index.php/jss/article/view/v044i10>

Xu, K. & Rowe, D. B. (2025), 'A Bayesian CAIPIVAT Approach with Through-Plane Acceleration to Enhance Efficiency of Simultaneously Encoded Slice Acquisition in FMRI', *Magnetic Resonance Imaging* **124**, 110540.

Appendix A: SHAKER Default Values

This section quickly defines the default values entered for SHAKER upon startup. In the *Data Viewer* pane, we have:

```
Phantom Size: 96
Map: M0
View: Axial
Slice: 48
```

The *MRI Parameters* are set to:

```
Signal Equation: GradientEcho_SigEq.m*
Trajectory: Cartesian_kspace.m*
Acceleration Factor: 1
Field Strength: 3
TE: 50
TR: 1000
Flip Angle: 90
EESP: 0.72
Number of Coils: 1
Reconstruction: CartesianIFFT.m*
Include B0 Inhomogeneity: 1
```

*: or first file alphabetically in respective folders. Lastly, the default *Task Design* options are preset to

```
HRF: Block
Initial Rest Images: 10
Epochs: 20
Task Images per Epoch: 15
Rest Images per Epoch: 15
```

SNR: 5
Task 1
CNR: 0.75
Phase: 0

Appendix B: SHAKER Structure and Objects

The SHAKER directory is structured in the following way:

```
main
  data                : contains phantom and activation data
    Phantom96.mat     : 4x[96x96x96] struct digital phantom
    ActMap96.mat      : [96x96x96] logical double
  functions           : internal helper functions of SHAKER
  kspaceTrajectories  : folder of custom k-space trajectories
  reconstructionMethods : folder of custom reconstruction methods
  signalEquations     : folder of custom signal equations
  statisticsFunctions : folder of statistical tests
  ksim.mlapp          : main SHAKER application
```

As shown, the `data` folder contains the digital phantom and default activation map that were described in Sections 2.2 and 2.3. The `functions` folder contains `generateTimeSeries.m` and `generateCoils.m`. The `generateTimeSeries.m` file takes in a noiseless rest and task k -space image as well as the experimental design, then outputs the complex-valued time series array with size $[n_{k_x} \times n_{k_y} \times n_c \times n_{IMG}]$. Other “helper” functions will continue to be placed here.

The `kspaceTrajectories` folder contains the trajectories described in Section 1.2. The `reconstructionMethods` folder contains all reconstruction method files (simple Cartesian IFT being the most commonly used). The `signalEquations` folder contains the signal equations described in Section 1.1. The `statisticsFunctions` folder contains statistical maps used to analyze the time series data. All four of these folders are where the relevant custom user files should be placed. There is a provided template in each of these folders to help support the creation of custom files. The input/output structure of each is given below. Note that while the templates expect each of the described inputs and custom files should be structured this way, they need not be used in the functions themselves. It is recommended for advanced statistical techniques and models that the simulated time series data be exported from data and examined in a more controlled setting.

```

template_kspace IN
    MRI : struct object from SHAKER
template_kspace OUT
    [kx, ky, timeMap] : array of single k-space measurement orders
        kx           : real double of measured k-space x locations
        ky           : real double of measured k-space y locations
        timeMap      : real double of time at k.s. points

template_reconstruction IN
    MRI           : struct object from SHAKER
    kSpace        : struct object from SHAKER
    kSpaceTimeSeries : [nkx x nky x nc x nimg] complex double of k.s.
template_reconstruction OUT
    imageTimeSeries : [nkx x nky x nc x nimg] cplx dbl of img space

template_sigEq IN
    M0      : [nx x ny] real double of M0 slice
    T1      : [nx x ny] real double of T1 slice
    T2star  : [nx x ny] real double of T2* slice
    deltaB  : [nx x ny] real double of dB slice
    kSpace  : struct object from SHAKER
    MRI     : struct object from SHAKER
template_sigEq OUT
    kspace : [nkx x nky] complex double of single k.s. array

template_stats IN
    TS      : [n(k)x x n(k)y x nc x nimg] cplx dbl of img or k-s t.s.
    kSpace  : struct object from SHAKER
    MRI     : struct object from SHAKER
template_stats OUT
    statsMap : [n(k)x x n(k)y x nc x nimg] real dbl of stat map

```

The two main objects referenced in many of the functions and operability of SHAKER are the `MRI` and `kSpace` objects. These struct objects contain the relevant information about the MRI settings and the *k*-space settings, respectively. The objects are made up of the following variables:

```

MRI (struct)
    SignalEquation (int)      : value indicating selected sig. eq.
    Trajectory (int)         : value indicating selected traj.
    AccelerationFactor(int)  : n_a acceleration factor
    FieldStrength (double)   : magnetic field strength
    EchoTime (double)        : TE stored in SI (seconds)

```

RepititionTime (double) : TR stored in SI (seconds)
 EESP (double) : effective echo spacing (ms)
 FlipAngle (double) : flip angle in deg
 NumberOfCoils (int) : n_c number of coils
 CoilSensitivities (double) : [nx x ny x nc] coil sensitivities
 gamma (double) : gyromagnetic ratio in MHz/T
 RelaxivityTable (double) : table of real values
 IncludeB0Inhomogeneity (int): indicator of if dB was applied

kSpace (struct)

- kX (double) : measured k-space x locations
- kY (double) : measured k-space x locations
- timeMap (double) : time (s) at measurement of k.s. points.