RESEARCH ARTICLE

Open-source magnetic resonance imaging: Improving access, science, and education through global collaboration

Lukas Winter[1] | João Periquito[2] | Christoph Kolbitsch[1] |
Ruben Pellicer-Guridi[3] | Rita G. Nunes[4] | Martin Häuer[1,5] | Lionel Broche[6] |
Tom O'Reilly[7]

[1]Physikalisch-Technische Bundesanstalt (PTB), Braunschweig and Berlin, Germany

[2]Department of Infection, Immunity & Cardiovascular Disease, The University of Sheffield, Sheffield, UK

[3]Asociación de investigación MPC, San Sebastián, Spain

[4]Institute for Systems and Robotics and Department of Bioengineering, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal

[5]Open Source Ecology Germany e.V. (nonprofit), Berlin, Germany

[6]Biomedical Physics, University of Aberdeen, Aberdeen, UK

[7]Leiden University Medical Center (LUMC), Leiden, The Netherlands

Correspondence
Lukas Winter, Physikalisch-Technische Bundesanstalt (PTB), Biomedical Magnetic Resonance – 8.1, Abbestr. 2-12, 10587 Berlin, Germany.
Email: lukas.winter@ptb.de

Abstract

Open-source practices and resources in magnetic resonance imaging (MRI) have increased substantially in recent years. This trend started with software and data being published open-source and, more recently, open-source hardware designs have become increasingly available. These developments towards a culture of sharing and establishing nonexclusive global collaborations have already improved the reproducibility and reusability of code and designs, while providing a more inclusive approach, especially for low-income settings. Community-driven standardization and documentation efforts are further strengthening and expanding these milestones. The future of open-source MRI is bright and we have just started to discover its full collaborative potential. In this review we will give an overview of open-source software and open-source hardware projects in human MRI research.

Abbreviations: AC, alternating current; ADC, apparent diffusion coefficient; AFNI, fMRI analysis software; ASCII, data format; BART, image reconstruction software; BIDS, Brain Image Data Structure; CAD, computer-aided design; CEST, chemical-exchange saturation-transfer; CIL, Core Imaging Library; CNC, computerized numerical control; Dafne, image segmentation software; DC, direct current; DNP, dynamic nuclear polarization; erwin, quantitative image postprocessing software; ESR, electron paramagnetic resonance; FAIR, Findability, Accessibility, Interoperability, and Reusability of data; FFT, fast Fourier transform; fMRI, functional MRI; FOV, Field of View; FreeSurfer, image registration and segmentation software; FSF, Free Software Foundation; Gadgetron, image reconstruction software; GOSH, Gathering for Open Science Hardware; GUI, graphical user interface; hMRI, quantitative image postprocessing software; ImageJ, image processing software; IP, intellectual property; ISMRM, International Society for Magnetic Resonance in Medicine; ISMRMRD, ISMRM raw data format; ITK, image registration software; JEMRIS, MR simulation software; Julia, programming language; KiCAD, electronic design software; MARCOS, Red Pitaya SDR-based MR control software; MARIE, EM simulation software; MIRTK, image processing software; MONAI, image segmentation software; MRILab, MR simulation software; MT, magnetization transfer; NiftyReg, image registration software; OCRA, Red Pitaya SDR-based MR control software; OSHWA, Open Source Hardware Association; OSI, Open Source Initiative; OSI² ONE, low-field MRI scanner; OSI², Open Source Imaging Initiative; PCB, printed circuit board; POSSUM, MR simulation software; pTx, parallel transmission; Pulseq, pulse sequence design software; PyQMRI, quantitative image postprocessing software; QMRITools, quantitative image postprocessing software; qMRLab, quantitative image postprocessing software; QUIT, quantitative image postprocessing software; RF, radiofrequency; Scanhub, MR control, acquisition, and postprocessing software; SDR, Software-Defined Radio; SigPy, RF pulse design software; SIRF, image reconstruction software; Spinach, MR simulation software; TOPPE, pulse sequence design software; VeSPA, image simulation software; Yarra, image reconstruction framework.

1 | INTRODUCTION

Magnetic resonance imaging (MRI) serves as a stark reflection of our global disparities. While MRI machines are indispensable in clinical practice within resource-rich hospitals, this technology appears to be nonexistent in low-income areas, rendering it practically inaccessible. One of the most powerful and versatile noninvasive medical imaging devices is also one of the most complex systems encountered in today's clinics. While some challenges of MRI research, development, and production are inherently tied to its fundamental physics and the realization through electrical, mechanical, and digital components, many other obstacles preventing affordable machines and reproducible research are human-made.

Black boxes, that is, inaccessible software, data, hardware, and documentation, continue to impede collaboration and progress in this vibrant interdisciplinary field of MRI. Within the realm of science, missing information prevents researchers from accessing, reproducing, and validating published results, hampering scientific progress and its translation into clinical applications.[1,2]

In machine development, deliberate concealment of information aims to thwart competitors from easily reproducing machines, effectively leading to higher investments required for already established technologies, subsequently raising costs for healthcare systems. Limited access to operational knowledge, such as parameters influencing imaging performance, poses challenges in advancing MRI from a qualitative to a more precise quantitative imaging technique.[3] Inaccessible information for the operation and maintenance of MRI machines creates difficulties for third parties attempting to repair and operate the equipment without vendor assistance. This intentional dependency on vendors can result in significant financial burdens or, at worst, render the equipment inoperable. Many countries heavily rely on donated or funded healthcare equipment (in some cases 80% of the total healthcare equipment is being donated).[4] However, only a fraction (10%–30%) of these donations become fully operational, largely because of a lack of user training and technical support.[4]

At practically all levels of healthcare product development, validation, documentation, and operation, whether for scientific purposes or clinical applications, the lack of access to information and/or the limited freedom to utilize available information unnecessarily complicates workflows: for example, efforts are being directed into time- and resource-intensive nondisclosure agreements, reverse engineering, and reinventing the wheel. Within healthcare systems, these challenges eventually manifest as exorbitant costs for machines and maintenance, a lack of interoperability, vendor lock-in effects, and high barriers to market entry. Consequently, significant disparities arise between regions and patients in terms of access to critical medical technology.[5,6]

Fortunately, opposing trends are gaining strength. The importance of nonexclusive information sharing has been identified as a vital element for efficient global research practices, leading funding agencies in the European Union (EU) to make open-access publications mandatory for receiving public funding.[7] Similarly, a recent US government policy announced the implementation of zero-embargo open-access publications by the end of 2025.[8,9] What initially began with free-to-read publications has now extended to data sharing, aiming to enhance the quality of results, foster global collaboration, accelerate innovation, and improve scientific transparency. Open-source software is increasingly being recognized as a fundamental component of an open science strategy, reinforcing officially a working culture that is predominantly founded on the principles of open source,[10] which are the free access, use, modification, and redistribution of code by anyone.[11] While still in its early stages, open-source hardware is also gaining momentum and is expected to follow a similar trend on a much broader scale. In a research setting, open-source hardware is an excellent fit, because laboratory designs can be (re)produced for only a fraction (∼10%) of the cost of proprietary counterparts.[12,13] Additionally, open-source hardware allows for customization, which is an asset for inventors with limited research budgets. Open-source hardware has been identified as a key factor in accelerating science and technology towards the United Nations' sustainable development goal of "building resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation".[14]

The progress towards open-source ecosystems is backed by various organizations such as the Open Source Initiative (OSI),[11] the Free Software Foundation (FSF),[15] the Open Source Hardware Association (OSHWA),[16] and the Gathering for Open Science Hardware (GOSH).[17] These organizations play a crucial role in establishing the necessary infrastructure for open-source collaborations. Recent initiatives include standards for data sharing such as the FAIR— Findability, Accessibility, Interoperability, and Reusability of data—principles,[18] certification, and standardization efforts for open-source hardware documentation such as the DIN SPEC 3105,[19,20] open-source hardware licenses such as the CERN OHL v2,[21] and open hardware labs created at universities, which grant access to tools and machines for open hardware production.[22]

The private sector has been quick to recognize and respond to these developments, adapting their business strategies to incorporate open source into their products. Open-source software has evolved into a multibillion-dollar market, and interestingly, the quality of the software has become the primary reason for companies to adopt open-source solutions.[23] Major tech companies, including Linux Red Hat, Microsoft, IBM,

Amazon, Google, SAP, Intel, OpenAI, and many others, actively utilize and contribute to open-source software. Likewise, open-source hardware designs are increasingly being commercialized by companies. Prominent examples include Arduino, Sparkfun, Adafruit, and Prusa.

These trends are also reflected in the MRI community. The International Society for Magnetic Resonance in Medicine (ISMRM) has increasingly been incorporating open-source content into its annual meetings. Multiple ISMRM study groups actively promote open-source collaboration and various initiatives have emerged to establish open-source standards within MRI. MR-Hub (https://ismrm.github.io/mrhub/) serves as a platform for open-source software, while the Open Source Imaging Initiative (OSI², https://www.opensourceimaging.org) focuses on both software and hardware.[24,25] Furthermore, thematically specialized initiatives have emerged to address specific research interests and improve open science practices. Examples include the Open Science Initiative for Perfusion Imaging,[26,27] MRSHub with open-source resources for MR spectroscopy,[28] and the Renal MRI Initiative.[29]

One of the most ambitious goals was set by OSI² upon its foundation in 2016: to develop a complete open-source MRI system. Since then, remarkable progress has been made with the first open-source hardware prototypes being presented in 2022, such as the low-field MRI system OSI² ONE.[30,31] This rapid advancement highlights the potential of collaborative efforts within the research community. Currently, these systems are diffusing to a wider user range, including industry collaboration. An exciting milestone was achieved when the first OSI² ONE system was successfully reproduced in Uganda, a remote place on the MR map.[33] To ensure that these technologies eventually reach patients, it is crucial to include sharing of documentation that can be used to facilitate regulatory approval as a medical device. Currently, such documentation blueprints are not publicly available yet, but work is underway to address this challenge.[32] The expansion of open-source hardware into the realm of open-source medical technologies offers numerous potential advantages: cost savings for the healthcare systems by reduced acquisition and maintenance expenses, accelerated innovation, global collaboration in product development, local adaptation and production, and more transparent processes in the reimbursement for medical services.[37–39]

The future of open-source MRI is bright, and we have only begun to discover its full collaborative potential. In this review, we aim to provide an overview of current open-source software and open-source hardware in human MRI research. We will delve into the concept of open-source, including its corresponding licensing and documentation models, and discuss the challenges involved in implementing open-source practices. While there are numerous exciting open-source projects in existence, it is not possible to cover them all comprehensively within this review. For those seeking further information, websites like https://ismrm.github.io/mrhub/ and https://www.opensourceimaging.org serve as excellent starting points. Other very valuable resources around open science practices and reproducibility in imaging are also available.[40–43]

Compressed lists of all open-source software and hardware projects covered in this review are given in Table 1 and Table 2. These lists are also available online (https://gitlab.com/osii-one/osii-orga) to provide a dynamic resource that can be updated with new project information or new projects by everyone.

1.1 | Open-source licenses and documentation

The term "open-source" was defined by the OSI and is based on the Free Software Definition by the FSF.[129] It means that the user is granted full control to freely use, study, modify, redistribute, and also commercially exploit the shared material. These definitions were primarily written for software source code, and software licenses must meet those requirements to qualify as free or open-source software. Both the OSI and FSF maintain a list of officially endorsed licenses.[130] However, the "freedoms" or "rights of open source" granted by free/open-source licenses can be applied to any material, including hardware. The OSHWA has published a definition clarifying the requirements for respective licensing terms for open-source hardware.[131] Today, there is a large variety of open-source copyright licenses available, optimized for different types of materials (software, hardware, images, fonts, etc.) and specific use cases (e.g., domain-specific liability or associated patents). However, especially for hardware, other domains of intellectual property (IP) rights may apply, most notably patent law or industrial design rights. To make hardware open source, it must be "freed" from all applicable domains of IP restrictions. Most commonly this is done by publishing the hardware design (1) under an open-source license, and (2) in a way that it is easily found on the internet (also called "defensive publishing"). The open-source license in turn ensures that the design files can be freely used, studied, modified, and redistributed. Open-source licensing schemes are often classified by their reciprocality, which determines the preservation of rights in derivative works (Table 1). Version 2.0 of the popular hardware-specific CERN Open Hardware License (CERN OHL) is available in three variants, reflecting these licensing schemes.[21,132] Licenses that prohibit commercial use or modification of the material (such as CC BY-NC or CC BY-ND) do not meet the licensing requirements of any of the abovementioned definitions and hence are not open-source licenses.

For effective reuse and widespread decentralized development of open-source hardware designs, complete and comprehensive technical documentation is crucial.[19,133] This includes the bill of materials, design files in their native format (e.g., 3D computer-aided design [CAD] models) and manufacturing-oriented export formats (e.g., STL files for 3D printing). It is important to note that the open-source hardware projects listed in this review may not adhere to a unified documentation standard, and often files that are needed to fully reproduce the designs are missing. Nevertheless, important pioneering steps of sharing designs have been performed and current community work is focusing on establishing documentation standards to improve documentation quality.

10991492, 0, Downloaded from https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/10.1002/nbm.5052 by University Of Aberdeen The Uni, Wiley Online Library on [08/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 List of open-source software resources, corresponding license, available files, published platform, and evaluated reference. A full regularly updated list can be found at https://gitlab.com/osii/cab.

Project name	Description	License	License type	Trapped?	Platform	Reference
3D Slicer	Image computing platform	3D Slicer license	permissive	No	Github.com	v5.2.2[44]
AFNI	Image postprocessing	Apache 2.0	permissive	No	Github.com	v0.3.8[45]
ANTs	Image postprocessing	Apache 2.0	permissive	No	Github.com	v2.4.4[46]
ANTsPy	Image postprocessing	Apache 2.0	permissive	No	Github.com	commit 931b124[47]
BART	Image reconstruction	BSD-3-Clause	permissive	No	Github.com	v0.8.00[48]
PyCoilGen	Gradient coil design	GPL-3.0	strong copyleft	No	Github.com	v0.1.3[49]
CoilGen	Gradient coil design	GPL-3.0	strong copyleft	Yes	Github.com	commit d8e6440[50]
CoSimPy	Circuit co-simulation	MIT	permissive	No	Github.com	v1.4.0[51]
Dafne	Image segmentation	GPL-3.0	strong copyleft	No	Github.com	v1.3-alpha2[52]
EPTlib	Electric properties tomography	MIT	permissive	No	Github.com	v0.3.3[53]
erwin	Quantitative MRI	MIT	permissive	No	Github.com	v1.0.2[54]
FreeSurfer	Image postprocessing	Apache 2.0	permissive	No	Github.com	v7.4.1[55]
Gadgetron	Image reconstruction	GADGETRON Software License (modified MIT license)	permissive	No	Github.com	v4.1[56]
GNURadio MRI	Pulse sequences	GPL-3.0	strong copyleft	No	bitbucket.org	v3.7[57]
HalbachOptimisation	Magnet simulation	MIT	permissive	No	Github.com	commit b5edf2e[58]
hMRI-Toolbox	Quantitative MRI	GPL-2.0	strong copyleft	Yes	Github.com	v0.5.0[59]
ImageJ	Image postprocessing	Public domain	permissive	No	Github.com	v2023.07.10[60]
ISMRMRD	MR raw data format	ISMRMRD software license (modified MIT license)	permissive	No	Github.com	v1.13.4[61]
ITK	Image postprocessing	Apache 2.0	permissive	No	Github.com	v5.3.0[62]
JEMRIS	MRI simulations	GPL-2.0	strong copyleft	Yes	Github.com	v2.9.0[63]
Keras	Deep learning	Apache-2.0	permissive	No	Github.com	v2.13.1[64]
KomaMRI.jl	MRI simulations	MIT	permissive	No	Github.com	v0.7.3[65]
LUMC Gradient Design Tool	Gradient coil simulation	MIT	permissive	No	Github.com	1a2dd56[66]
Madym	Quantitative MRI	Apache-2.0	permissive	No	Gitlab.com	v4.23.0[67]
MARCOS	Console software	GPL-3.0	strong copyleft	No	Github.com	commit 35c599e[68]
MARIE	EM field simulations	GPL-2.0	strong copyleft	Yes	Github.com	commit 3,254,454[69]
MIRTK	Image postprocessing	Apache 2.0	permissive	No	Github.com	v2.0[70]
MONAI	Deep learning	Apache-2.0	permissive	No	Github.com	v1.2.0[71]
MRILab	MRI simulations	BSD-2-Clause	permissive	Yes	Github.com	v1.3[72]
MRIReco.jl	Image reconstruction	MIT	permissive	No	Github.com	v0.7.1[73]
NiftyReg	Image postprocessing	BSD-3-Clause	permissive	No	Github.com	v1.3.9[74]

TABLE 1 (Continued)

Project name	Description	License	License type	Trapped?	Platform	Reference
OCRA	Console software	GPL-3.0	strong copyleft	No	Github.com	commit aac1907[75]
ODIN	Development interface	GPL-2.0	strong copyleft	No	sourceforge.net	v2.0.5[76]
Pulseq	Pulse sequences	MIT	permissive	Yes	Github.com	v1.4.0[77]
PyPulseq	Pulse sequences	AGPL-3.0	strong copyleft	No	Github.com	v1.4.0[78]
PyQMRI	Quantitative MR	Apache-2.0	permissive	No	Github.com	v1.2.2[79]
PyTorch	Deep learning	own license	permissive	No	Github.com	v2.0.1[80]
QMRITools	Quantitative MR	BSD-3-Clause	permissive	Yes	Github.com	v3.3.0[81]
qMRLab	Quantitative MR	MIT	permissive	No	Github.com	v2.5.0b[82]
QUIT	Quantitative MR	MPL-2.0	weak copyleft	No	Github.com	v3.3[83]
Scanhub	Console software	GPL-3.0	strong copyleft	No	Github.com	commit 0075c29[84]
SEPIA	Susceptibility mapping	MIT	permissive	Yes	Github.com	v1.2.2.4[85]
Shimming Toolbox	B1/B0 shimming	GPL-3.0	strong copyleft	No	Github.com	v0.1[86]
SigPy	MRI simulations	BSD-3-Clause	permissive	No	Github.com	v0.1.25[87]
SimpleElastix	Image postprocessing	Apache 2.0	permissive	No	Github.com	v0.10.0[88]
SIRF	Image reconstruction	Apache-2.0	permissive	No	Github.com	v3.4.0[89]
Spinach	MRI simulations	MIT	permissive	Yes	own website	v2.7.6049[90]
SPM	Quantitative MRI	GPL-2.0	strong copyleft	Yes	Github.com	commit 81dab04[91]
ThinWire – MRI Gradient Coil Design	Gradient coil simulation	MIT	permissive	Yes	Github.com	commit 9f020c9[92]
TOPPE	Pulse sequences	MIT	permissive	Yes	Github.com	v3.0[93]
VeSPA	MRI simulations	BSD-3-Clause	permissive	No	Github.com	commit a944347[94]
Virtual MRI scanner	MRI simulations	AGPL-3.0	strong copyleft	No	Github.com	v2.0[95]
Yarra	Image reconstruction	GPL-3.0	strong copyleft	No	bitbucket.org	client v0.62, server v0.97[96]

TABLE 2 List of open-source hardware resources, corresponding license, available files, published platform, and evaluated reference. Native file formats may be using proprietary software. Export file formats (e.g., STEP) can be processed using open-source software, making the underlying information extractable. They can also support replication if they are delivered in a production-ready format (e.g., STL for 3D printing). A full regularly updated list can be found at https://gitlab.com/osii/cab.

Project name	Description	License	License type	Native files	Export files	Bill of materials	Instructions	Platform	Reference
32-channel RF receive chain amplifiers	modular 32-channel RF signal amplifier and attenuator	CERN-OHL-1.2	strong copyleft	Yes	Yes	Yes	Yes	Github.com	commit 2cae1f8[35]
8ch scalable 19F coil	Scalable 3 T RF Coil	none	-	Yes	Yes	No	No	Github.com	commit db55480[97]
8ch TX/RX RF coil 3 T	8-channel 3 T pTx head coil	CERN-OHL-W-2.0	weak copyleft	Yes	Yes	Yes	Yes	PTB-hosted Gitlab	commit d02caee3[98]
8ch TX/RX RF coil 7 T	8-channel 7 T pTx head coil	CERN-OHL-W-2.0	weak copyleft	Yes	Yes	No	Yes	PTB-hosted Gitlab	commit e6033e92[99]
ASL phantom	Perfusion phantom	none	-	Yes	Yes	No	No	Github.com	commit 11d1293[100]
COSI measure	3-axis measurement system	CERN-OHL-1.2	strong copyleft	Yes	Yes	Yes	Yes	Github.com	v1.0[101]
COSI transmit	RF power amplifier 1 kW	CERN-OHL-1.2	strong copyleft	Yes	Yes	Yes	Yes	Github.com	commit fee73e7[102]
Current driver for local B_0-shim coils	Current driver, 8 A per channel	none	-	Yes	Yes	Yes	Yes	self-hosted Mediawiki	v7[103]
EASY_MRI_RFPA	BASIC RFPA for MRI applications	none	-	Yes	Yes	Yes	Yes	Github.com	v1.00[104]
GPA-FHDO	B0 shim or gradient amplifier	CERN-OHL-1.2	strong copyleft	Yes	Yes	Yes	Yes	Github.com	v1.3[105]
GPA-RP-Adapter	Red pitaya shield to connect GPA-FHDO	CERN-OHL-W-2.0	strong copyleft	Yes	No	Yes	No	Github.com	commit a19ebca[106]
H-field RF probe	magnetic field probe	CERN-OHL-1.2	strong copyleft	Yes	Yes	Yes	-	Github.com	v1.0[107]
High peak/average power TR-switch for thermal MR	Transmit/receive Switch	CERN-OHL-1.2	strong copyleft	Yes	Yes	Yes	Yes	Github.com	v1.0[108]
LimeSDR	NMR console hardware	CC-BY-3.0	permissive	Yes	Yes	Yes	Yes	Github.com	1v4[109]
Maker-gradient-amplifier	Gradient amplifier	MIT	permissive	No	Yes	Yes	No	Github.com	commit 0f4e5e3[110]
MGH tabletop MRI	Tabletop MRI scanner	None	-	No	Yes	Yes	Yes	self-hosted Mediawiki	Rev 23.03.2020 12:59 UTC, Jaystock[111]
MR dilution phantom	Contrast agent phantom	None	-	No	Yes	No	Yes	Github.com	commit 9fb1e97[112]
MRI cage test set	Antenna set to measure attenuation of Faraday RF cages	CERN-OHL-W-2.0	weak copyleft	No	Yes	No	Yes	Github.com	commit 3452e02[113]

TABLE 2 (Continued)

Project name	Description	License	License type	Native files	Export files	Bill of materials	Instructions	Platform	Reference
MRI compatible wireless sensors for motion tracking	MRI compatible wireless sensor for motion tracking	MIT	permissive	Yes	-	Yes	Yes	Github.com	commit 1dd0414[114]
MRI distortion phantom	Modular phantom to measure distortion	CERN-OHL-W-2.0	weak copyleft	Yes	No	Yes	Yes	Github.com	commit 4,896,369[115]
NMR relaxometry system	Tabletop NMR	None	-	Yes	No	Yes	Yes	Github.com	commit 7bfa320[116]
OCRA tabletop GPA	Gradient power amplifier	CERN-OHL-W-2.0	weak copyleft	No	Yes	Yes	Yes	self-hosted download link	Rev003[117]
OCRA1	4-channel digital-to-analog converter to play out gradient waveforms	CERN-OHL-W-2.0	weak copyleft	No	Yes	Yes	Yes	self-hosted download link	Rev003[118]
Open source NIST phantom	MR phantom	GPL-3.0	strong copyleft	No	Yes	Yes	Yes	Github.com	commit 251e5bd[119]
Opencore NMR	NMR spectrometer	none	-	Yes	Yes	Yes	Yes	Github.com	v2.0.0[120]
OpenForce MR	MR compatible force sensor	GPL-3.0	strong copyleft	Yes	Yes	Yes	Yes	Github.com	commit d0c5d28[121]
OSI² ONE Magnet	MR magnet, 48 mT, 30 cm bore	CERN-OHL-W-2.0	weak copyleft	Yes	Yes	No	No	Gitlab.com	v1.0.0[122]
OSI² ONE RFPA	RF power amplifier	CERN-OHL-W-2.0	weak copyleft	Yes	Yes	Yes	Yes	Gitlab.com	commit 1f1f1fd4[123]
Pneumatic phantom	Inflation system to mimic respiration-induced MRI artifacts	MIT	permissive	Partially	Partially	No	No	OSF.io	rev 30.03.2022 18:07 UTC[124]
pTx implant safety testbed	Parallel transmission hardware	CERN-OHL-W-2.0	weak copyleft	Yes	-	Yes	Yes	PTB-hosted Gitlab	commit 2e66f5db[125]
Solenoid RF coil	20.5 cm diameter RF coil	CERN-OHL-W-2.0	weak copyleft	Yes	Yes	No	No	Gitlab.com	commit da7e5dd4[126]
TU Delft GA	Gradient amplifier	CERN-OHL-W-2.0	weak copyleft	No	Yes	Yes	Yes	Gitlab.com	commit cdb509a2[127]
Wireless reference implant	Wireless reference implant hardware	CERN-OHL-W-2.0	weak copyleft	Yes	No	Yes	Yes	PTB-hosted Gitlab	commit 055ad252[128]

2 | OPEN-SOURCE SOFTWARE PROJECTS FOR MRI

Making software available as open source has become increasingly popular in MRI. Partly because of community efforts to make research methods used in publications transparent and reproducible, and partly because of online platforms, such as GitHub,[134] Bitbucket,[135] and GitLab,[136] which facilitate the sharing and collaborative development of software. The increased use of open-source high-level programming languages such as Python[137] has also contributed to the ease of code sharing. The quality and reusability of open-source software can vary greatly, ranging from simple scripts with limited documentation to large software projects with comprehensive documentation and community-based support. In recent years, the application of machine learning techniques in MRI has led to a proliferation of small open-source software projects. The majority of these consist of the code necessary to reproduce the results of a scientific publication and they provide a very valuable starting point for further developments. In this review we are going to focus on open-source software projects that do not just reproduce the results of a single scientific paper, but provide a wider range of functionality. In addition, we only include software projects that provide reasonable long-time support, that is, which are compatible with up-to-date operating systems and software versions.

We will cover three main areas of open-source software: (i) simulation of MR data acquisition and design of vendor-independent MR sequences; (ii) image reconstruction; and (iii) postprocessing of reconstructed images. It is worth noting that several of the software packages discussed cover multiple areas.

2.1 | MR simulations and sequences

Several open-source tools have been developed to simulate different aspects of the MR acquisition process. These tools enable the evaluation of the impact of pulse sequences on the measured data, resulting from changing a simple acquisition parameter such as the echo time to making more profound changes to specific pulse sequence elements, such as the characteristics of a radio frequency (RF) pulse.

2.1.1 | JEMRIS

JEMRIS, written in C++ but taking advantage of a MATLAB GUI, enables fast and efficient MR simulations.[63,138] Pulse sequences can be generated by concatenating multiple sequence objects, such as RF and gradient objects, each having its own set of attributes. It enables the simulation of B1 + transmit field inhomogeneity and of multichannel RF transmit systems; the latter is done by providing multiple RF waveforms, each assigned to a given coil channel. Bio-Savart transmit loops can be simulated, and the option to read external coil files is also provided. Besides predefined waveforms, this software can simulate arbitrary RF and gradient waveforms that can be defined using symbolic analytical formulas. Potential applications include simulating nonlinear gradient encoding or the impact of concomitant fields. This tool has been extended to simulate realistic flow motion.[139] More recently, JEMRIS has been integrated within an end-to-end open-source MR imaging workflow.[140] This workflow enables running a simulation with JEMRIS or to acquire data on a real scanner using the exact same pulse sequence, making it easier to carry out sequence prototyping and debugging. To execute the pulse sequence on different scanners and reconstruct images, Pulseq interpreters,[77] the ISMRMRD open data format,[61] and open-source reconstruction tools such as BART[48] are utilized (Figure 1).

2.1.2 | MRILab

MRILab also enables carrying out numerical simulations of MRI experiments.[72,141] The user interfaces run on Matlab but interface with C++ code and take advantage of GPU capabilities to enable running on a standard PC. When it was first made available, the main differentiating feature of this software was to model tissues composed of multiple compartments. This makes it possible to simulate quantitative magnetization transfer (MT), multicomponent T_1 and T_2 relaxometry, as well as chemical-exchange saturation-transfer (CEST) experiments. This feature is currently also available in JEMRIS.

2.1.3 | KomaMRI.jl

An alternative MRI simulation platform has been implemented recently using the Julia programming language.[65,142] It has the advantage of providing cross-platform compatibility and high CPU and GPU parallelization, enabling fast computations. It can read pulse sequence files in Pulseq format (see below) and outputs raw data using the ISMRMD standard. Furthermore, it provides integration with the MRIReco.jl reconstruction software package and other reconstruction programs compatible with the ISMRMD raw data file format.
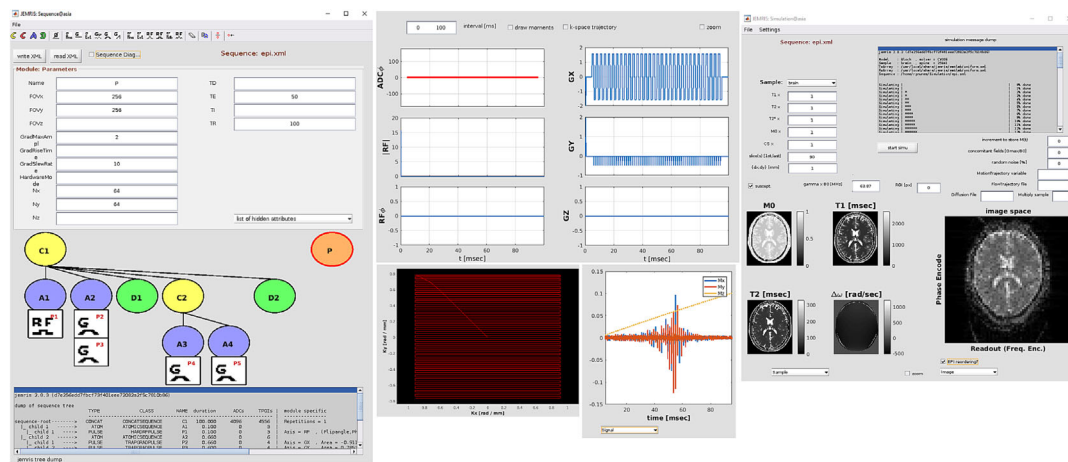
FIGURE 1 MRI simulation using JEMRIS.[63] Left: a pulse sequence is built by adding the RF and gradient objects to the tree and defining all the parameters required for the calculations (e.g., field of view, echo time). Middle: the sequence diagram and the k-space trajectory can be visualized for debugging. Right: having chosen the sample properties and selected the sequence file, the simulation can be run, and the obtained signal (bottom right of the middle column) processed to reconstruct an image. Here, an echo planar imaging sequence is simulated in a brain phantom, including susceptibility effects. The produced image appears distorted as expected.

2.1.4 | ODIN

ODIN is a GUI-supported C++ software framework to develop and simulate MR sequences.[76] Contemporary MRI techniques are available: for example, sequence modules for echo planar imaging and spiral-imaging, parallel imaging with GRAPPA reconstruction, 2D-pulses and field map-based distortion correction.

2.1.5 | Pulseq, PyPulseq, and TOPPE

Pulse sequence design using vendor-specific code requires a steep learning curve because it typically utilizes low-level programming languages. This can make the testing of new ideas and the translation of a given pulse sequence to systems produced by other manufacturers very complicated. The goal of these tools was to simplify these procedures by relying on high-level programming languages such as MATLAB (Pulseq[77,143] and TOPPE[93,144]) or the freely available Python language (PyPulseq[78,145]).

Pulseq proposed a new compact file format (".seq") for describing pulse sequences, based on a hierarchical representation consisting of a timing table that defines when different sequence blocks occur (Figure 2). The seq files can be generated by running high-level scripts or programmed using the JEMRIS GUI and can currently be interpreted and executed on Siemens, GE (via TOPPE), and Bruker commercial scanners. The development of interpreters specific to other manufacturers' platforms or open-source scanners can be expected in the future.

2.1.6 | Virtual MRI scanner

Virtual MRI scanner,[95,146] a result from the ISMRM 2019 Junior Fellow Challenge,[147] is a web-based platform that enables running virtual MRI experiments, simulating the interaction with a console, patient registration, and selection of the imaging protocol and sequence parameters. Once the experiment is set up, a ".seq" file is written with PyPulseq and Bloch simulations are carried out using a discrete event approach. An important feature of this software is to outsource computationally heavy tasks to a different computer other than that running the GUI.

2.1.7 | Spinach

Spinach is a spin dynamics library that carries out calculations using the quantum description of magnetic resonance phenomena.[90,148] Computations of complex systems are made possible in reasonable times by keeping only the main contributions from all possible spin-states in the simulated system. This software has the particularity of being able to support simulations of all forms of MR spectroscopy, including not only nuclear
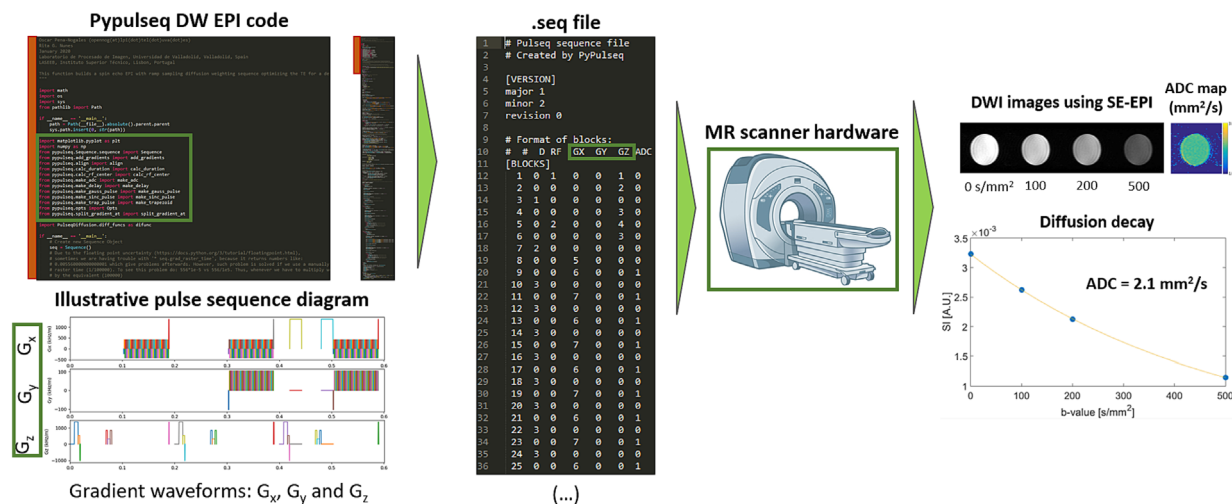
FIGURE 2 Pulseq[77] and Pypulseq[78] workflow, illustrating the design and testing of a diffusion-weighted echo planar imaging (EPI) sequence. Left: the pulse sequence is designed using high-level Matlab (Pulseq) or Python (Pypulseq) code as in this example, and its diagram is displayed below. Middle: a seq file is written, containing all the instructions concerning the RF, gradient, and sampling objects, and is imported into the MR scanner for running the sequence. Right: a set of images with different levels of diffusion-weighting (b-values) is acquired and used to estimate the apparent diffusion coefficient (ADC) map through monoexponential fitting of the signal decay in each voxel.

magnetic resonance (NMR), but also electron paramagnetic resonance (ESR) and dynamic nuclear polarization (DNP), among others. This is possible by splitting the code so that all core operations can be performed using a general algebraic abstraction applicable to any finite-state quantum system, whereas a separate part of the code is designed to deal with the specific assumptions required to model each spectroscopy modality.

2.1.8 | VeSPA

The Versatile Simulation, Pulses and Analysis (VeSPA) software enables users to create RF pulses, simulate their behavior and compare their performance for use in MR applications, including MR spectroscopy.[94,149] The software contains a database of pre-existing RF pulse designs that can be refined by the user, and the designed RF pulses can be exported using multiple file formats including ASCII and vendor-specific formats (Siemens). It is also possible to simulate and analyze spectroscopic data using VeSPA.

2.1.9 | SigPy

SigPy is a Python tool that includes toolboxes for MRI reconstruction and RF pulse design.[87,150] The latter (SigPy.RF) enables designing, for example, adiabatic, multiband, and B_1^+-selective RF pulses, including k-space trajectory design and parallel transmit capabilities. It also performs Bloch simulations to predict the performance of the designed RF pulses.

2.2 | Image reconstruction

Image reconstruction commonly describes the process of solving the inverse problem of obtaining an image from the acquired MR raw data. Often this problem is ill-posed and prior knowledge needs to be included in the form of additional regularization terms, which can require iterative methods. In addition, image reconstruction also involves other tasks such as prewhitening of the MR raw data, k-space trajectory estimation, calculation of coil sensitivity maps, and more. Here, we focus on image reconstruction software that covers the entire pipeline from the raw data to the final image.

2.2.1 | ISMRMRD

Most MR vendors have their own MR raw data format that can only be interpreted using proprietary tools, which makes it cumbersome to create vendor-agnostic image reconstruction and postprocessing software. To facilitate software compatibility regardless of the scanner, at an ISMRM workshop in 2013, a group of MR scientists defined an open-source MR raw data format named ISMRM raw data (ISMRMRD).[61,151] The ISMRMRD project also provides a standard for reconstructed image data and open-source converters to translate proprietary raw data formats to ISMRMRD. This community-driven initiative was a huge step towards reproducible science and enabled the harmonization of image reconstruction software.

2.2.2 | Gadgetron

In parallel to the development of ISMRMRD, the image reconstruction pipeline Gadgetron was created.[56,152,153] The main aims of Gadgetron are to be flexible and modular by utilizing a chain of individual gadgets. These modules carry out tasks such as noise prewhitening, sorting of the data, iterative image reconstruction, and estimation of quantitative parameters. The input and output of each gadget are standardized so that gadgets can be easily combined to generate tailored image reconstruction pipelines. MR raw data (e.g., ISMRMRD) can be sent to Gadgetron via a Transmission Control Protocol/Internet Protocol connection. This allows for the streaming of data directly from the MR scanner to a custom image reconstruction pipeline. The reconstruction can be carried out on a separate reconstruction server with multiple cores of modern CPUs and GPUs that ensure short reconstruction times. This enables, for example, the reconstruction of highly undersampled non-Cartesian MR raw data with iterative image reconstruction approaches during image-guided interventions. Gadgetron is implemented in C++, but high-level interfaces using, for example, Python, are also available, which merge computational speed with rapid prototyping.

2.2.3 | BART

BART focuses on image reconstruction providing a range of solvers and regularization functionals (e.g., total variation, sparsity in the wavelet domain, or low-rank approaches).[48,154] It is mainly implemented in C++ and image reconstruction can be carried out via the command line. High-level interfaces using, for example, Python, are also available. BART uses its own format for raw and image data but some support for ISMRMRD is already available.

2.2.4 | SIRF

SIRF is a framework that allows for the reconstruction of multimodality medical imaging data such as PET-MR.[89,155] Conveniently, the framework provides a unified interface to already existing image reconstruction packages (e.g., Gadgetron for MR and Software for Tomographic Image Reconstruction [STIR] for PET[156]). In addition to image reconstruction, SIRF also provides image registration capabilities utilizing NiftyReg and advanced iterative optimization techniques using Core Imaging Library (CIL).[157,158] These capabilities also allow for motion-corrected image reconstruction.[159]

2.2.5 | MRIReco.jl

A more recent software package is MRIReco.jl, which enables MR image reconstruction using Julia.[73,160] It provides algorithms for iterative regularized image reconstruction, is ISMRMRD compatible, and also contains an MR simulation module.

Figure 3 shows the reconstruction of the same MR raw data file with Gadgetron, BART, SIRF, and MRIReco.jl. This should not be seen as a comparison of different reconstruction packages but an illustration that all these reconstruction packages can process real MR raw data (a docker file providing all the necessary reconstruction packages and a script to reproduce this figure can be found at https://github.com/ckolbPTB/OpenSourceMrRecon). It can also be used as a starting point to try out and compare different reconstruction packages, avoiding a lengthy installation process.

Apart from Gadgetron, all the other image reconstruction frameworks are aimed at offline image reconstruction, that is, the MR raw data need to be exported from the scanner first. This can be challenging to apply in clinical practice. Yarra is a project that provides a general framework to integrate custom image reconstruction methods in the standard clinical workflow for easy translation of novel developments.[96]
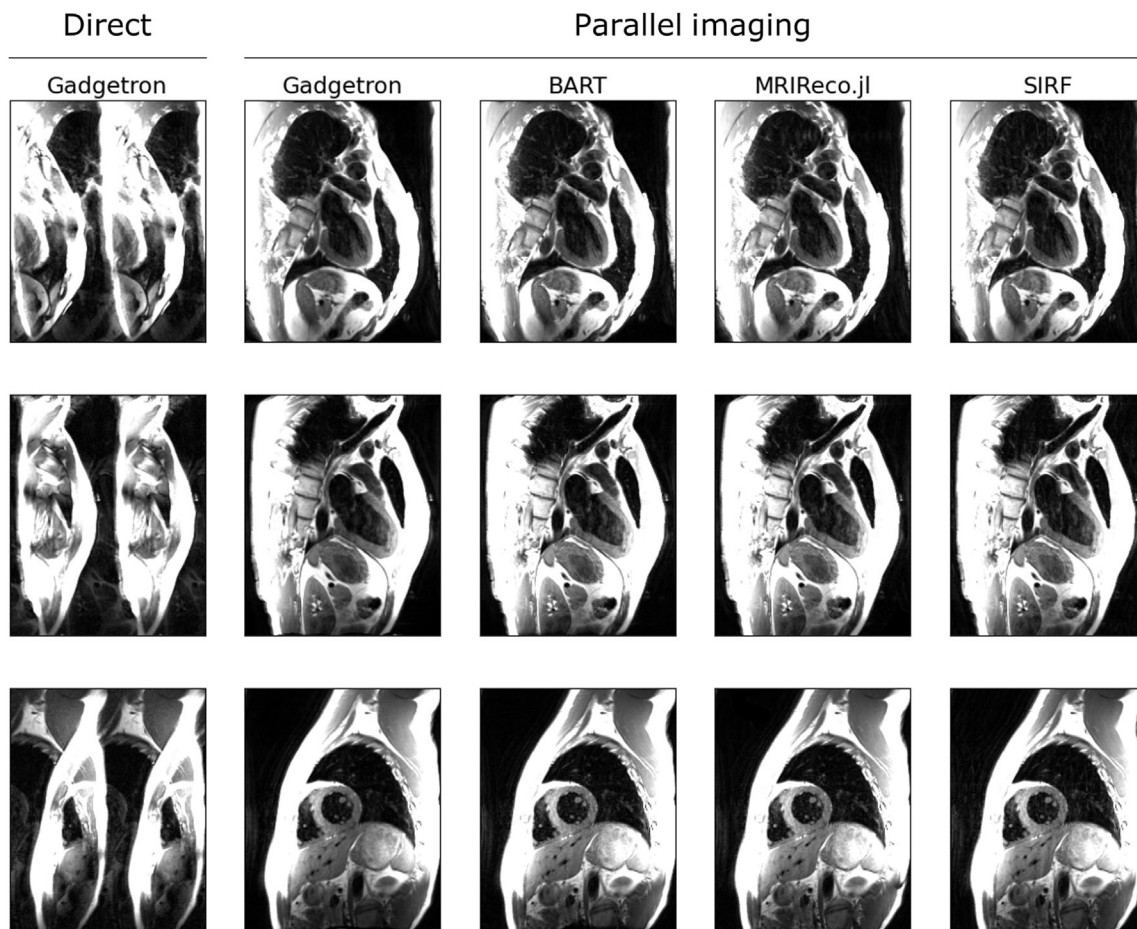
FIGURE 3 Reconstruction results of a turbo spin echo black-blood acquisition on a 3-T Siemens scanner using Gadgetron, BART, MRIReco.jl, and SIRF in three different orientations of the heart. The dataset was undersampled by a factor of ∼2. The direct reconstruction corresponds to a simple FFT followed by coil-combination and was carried out with Gadgetron. For parallel imaging, the information from the different receiver coils was utilized to remove undersampling artifacts. The type of algorithm used by each method varies but the final results are comparable. See https://github.com/ckolbPTB/OpenSourceMrecoRecon for more details.

2.3 | Image postprocessing

Image postprocessing is a method to extract information from an MR image. For general image analysis and postprocessing with a wide range of options, ImageJ is one of the most widely used open-source programs.[60] ImageJ can display, edit, save DICOM images, and perform deconvolution, registration, segmentation, and many more features that can be embedded via Plugins. More specialized MR analysis techniques such as more complex image registrations, automatic segmentations or quantitative analysis require more specialized software, details of which are presented below.

2.3.1 | Image registration

Image registration aligns two images so that common features overlap and can be used to compensate for misalignments due to subject motion and geometric distortions, align images from different modalities, timepoints (follow-up scans), or subjects (population studies). Among others, registration makes it possible to combine structural images with functional information and improves tumor delineation for surgical planning.

ITK

ITK emerged in 2002 and is one of the most used open-source tools for image registration in medical imaging.[62,161] Numerous MR scientific papers use ITK and often ITK algorithms are used as standard references for implementations of different registration techniques.[162–166] Several

scientific groups expanded ITK's core functions. Prominent examples are Elastix and its Python, Java, R, Octave, Lua implementation SimpleElastix[88,167]; ANTs[46,168] and its Python implementation ANTsPy.[47]

AFNI

AFNI is a software for the analysis and visualization of 3D human brain functional MRI (fMRI).[45,169–172] It enables visualizing neural activation maps within higher resolution anatomical scans. Slices in each cardinal plane can be viewed simultaneously. Manual placement of markers on anatomical landmarks allows transformation of anatomical and functional scans into stereotaxic (Talairach-Tournoux) coordinates.

FreeSurfer

FreeSurfer is a set of software tools for the study of cortical and subcortical anatomy.[55,173] In the cortical surface stream, the tools construct models of the boundaries between white and cortical gray matter, and the pial surface. Once these surfaces are known, several anatomical measures become possible, including cortical thickness, surface area, curvature, and surface normal at each cortical location. Surfaces from individuals can be aligned via an atlas with a high-dimensional nonlinear registration algorithm.

MIRTK

MIRTK is a research-focused image processing toolkit.[70] It provides a collection of libraries and command-line tools to assist in processing and analyzing imaging data. The main application of MIRTK is in adult and neonatal brain MR image registration, as well as the reconstruction of cortical surface meshes.

NiftyReg

NiftyReg is a suite of image registration and alignment tools primarily intended for use with MRI and other brain scan images.[74] NiftyReg is able to perform rigid, affine, and nonlinear registration of 2D and 3D images stored as Nifti or Analyze (nii or hdr/img).

2.3.2 | Image segmentation

Image segmentation techniques extract image features by subdividing the image into different regions. Manual MRI segmentation is still a common practice; however, it suffers from poor reproducibility, user variability, and is a very time-consuming task. Automatic segmentation tools have emerged in recent years using different segmentation techniques (e.g., threshold, clustering, edge, region or atlas-based image segmentation and, more recently, deep learning-based segmentation).[174]

3D Slicer

3D Slicer supports versatile visualizations and provides advanced functionality such as automated segmentation and registration for a variety of application domains.[44,175] Unlike a typical radiology workstation, 3D Slicer is not tied to specific hardware. An example of the 3D Slicer workflow using different open-source postprocessing tools is illustrated in Figure 4.

Dafne

Dafne is a program for MR image segmentation that includes advanced deep learning models (based on Keras[64]) for automatic segmentation.[52] The user can refine the automated results, and the software will learn from the improvements and modify its internal models accordingly.

MONAI

MONAI is based on PyTorch and Numpy libraries especially created for deep learning in healthcare.[71,80,178] It aims to capture the best AI development practices for healthcare researchers, focusing on medical imaging. MONAI has a repository (Model Zoo) that hosts medical imaging segmentation models developed by the scientific community.

2.3.3 | Quantitative parameter estimation

Commonly, MRI is used as a qualitative imaging modality, that is, the pixel values of reconstructed images are unitless and individually do not provide diagnostic information. Diagnosis is only possible due to the contrast between pathological and healthy tissue. In recent years this has started to change, and more and more quantitative MR techniques have been proposed, which directly provide quantitative (bio-)physical parameters (e.g., T1 relaxation time or blood flow). These parameters can be estimated by fitting a model to a sequence of qualitative images with different
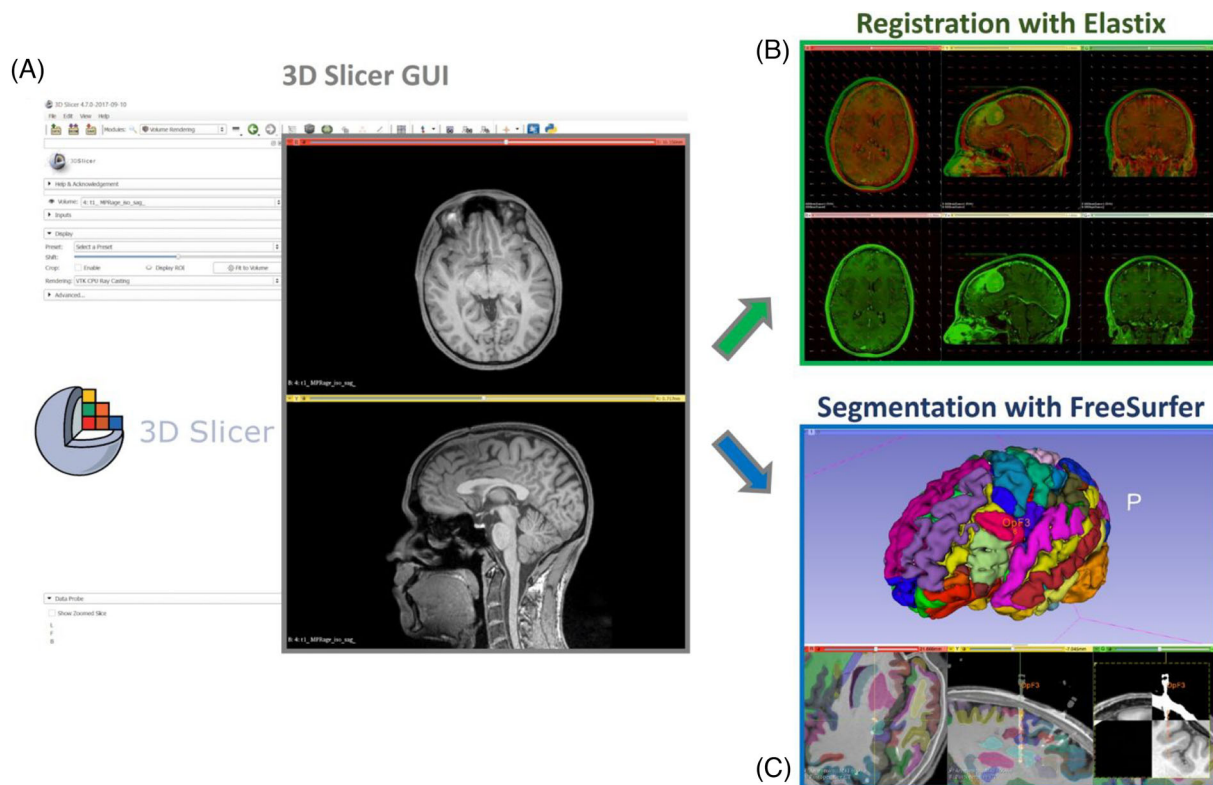
FIGURE 4 3D Slicer and workflow examples for MR image registration and segmentation. (A) 3D Slicer GUI.[44] (B) MRI brain registration using the Elastix extension for 3D Slicer.[88] Moving images are registered to fixed images. As an output the software provides the registered images and deformation fields. Image source from SlicerElastix.[176] (C) Visualization of FreeSurfer[55] segmentation for the assessment of stereotactic surgery in epilepsy. Figure adapted from previous works.[177]

contrasts. Examples of the numerous quantitative tools are electric properties tomography,[53,179] blood perfusion quantification,[67,180] and susceptibility mapping.[85,181] Here, we focus on software packages for quantitative analysis that are actively maintained and documented.

qMRLab

qMRLab provides a framework implemented in MATLAB/Octave to estimate quantitative parameters such as T1 and T2 relaxation times, B_1 and B_0 field maps, and magnetic transfer ratios for a wide range of MR sequences.[82,182] Parameter estimation often requires additional information about data acquisition and image reconstruction, which is frequently not part of the data exported from the MR scanner. qMRLab also advocates the development of qMRI-BIDS, an open-source data format for qMRI.[183] This format is derived from the Brain Image Data Structure (BIDS),[184] aiming to provide all the necessary metadata for accurate parameter estimation.

hMRI-toolbox

Another software package for quantitative parameter estimation implemented in MATLAB is hMRI-toolbox.[59,185] It relies on the Statistical Parametric Mapping (SPM) software package also available for MATLAB[91] and is targeted for qMRI of the brain.

QMRITools

QMRITools is a software package written in Mathematica.[81,186] Similar to qMRLab, its functionality is very broad and ranges from T1 and T2 mapping to fat-water quantification.

PyQMRI

A Python implementation for quantitative parameter estimation is provided by PyQMRI.[79,187] This software package is specifically aimed at both image reconstruction and parameter estimation.

QUIT

QUIT is a C++-based software package that allows for quantitative susceptibility mapping, B_1 mapping, perfusion estimation, and many more.[83,188]

erwin

A similar range of parameter estimates is provided by erwin, a Python-based software package for quantitative parameter estimation.[54] erwin can be imported as a Python package or used directly from the command line.

In addition to these software packages mainly aimed at processing reconstructed images, quantitative parameter estimation can also be carried out with the software packages discussed in "Image Reconstruction" (section 2.2). Gadgetron and SIRF allow for parameter estimation after image reconstruction. BART provides reconstruction techniques that directly reconstruct (bio-)physical parameters from the MR raw data without an intermediate step of reconstructing qualitative images.[189]

3 | OPEN-SOURCE MR SCANNERS

Designing the hardware required for a complete MR system involves a great deal of expertise in a wide range of engineering areas. The combination of multidisciplinary expertise, costly infrastructure, and materials costs have long restricted the handling of this technology to a handful of privileged manufacturers. Therefore, designing a complete open-source MR system was, for a long time, elusive. Recently, portable, permanent magnet-based low-field ($B_0 < 75$ mT) MRI systems have been shown to provide useful diagnostic value and address the MRI accessibility issue through a lower cost of ownership compared with more conventional higher field systems.[190–194] These low-field scanners are the most suitable candidates to exercise an open-source strategy, for several reasons:

1. Lower material and instrumentation costs, facilitating access to a larger community on limited budgets and in different places around the globe.
2. Construction feasible with easily accessible manufacturing methods.
3. Minimal siting requirements (low stray-field, no RF-shielded room, low power requirements).
4. Safer to produce and use: the lower magnetic field strengths and power usage largely reduce hazards presented to builders and practitioners.
5. Reduced risk on operating a scanner safely on human subjects (overall greatly reduced specific absorption rate [SAR] risks, simplified SAR modeling and monitoring).
6. Low-field MR companies would benefit from an open ecosystem helping with scanner development.

Within this strategy, the OSI^2 ONE low-field MR scanner, which consists largely of open-source software and hardware components, has been presented (Figure 5).[30,31,195] The MR magnet optimization is performed by an open-source Python code that finds the optimal location of small cubic permanent magnets in a ringlike arrangement.[58,190,196] A 47-mT magnet for imaging of the head and extremities was designed with an inner bore diameter of 30 cm using $12 \times 12 \times 12$ mm^3 N48 NdFeB magnets.[122] The low-field magnet can be constructed by hand using relatively affordable production machines such as CNCs, laser cutters, or 3D printers. The inevitable deviations through this low-cost manufacturing process led to a substantial difference between the simulated and realized B_0 field distribution of the magnet, which, however, can be corrected for by B_0-shimming routines. A three-axis robotic stage COSI Measure[101,197] equipped with a magnetic field sensor can map the initial magnetic field distribution. This information is then applied to run various shimming algorithms to calculate the position and orientation of smaller cubic magnets to homogenize B_0.[198,199] After a few iterations, B_0 homogeneities of $\sim$450 ppm over a 20 cm diameter spherical volume can be achieved.

For the gradient coils, several open-source software tools exist to calculate the wiring paths based on predefined requirements such as linearity over a target volume or coil efficiency.[49,50,66,92,200] These digital models can then be exported to CAD software to design a holder for the gradient coil wire. For low-field head-only systems, the gradient coils are much more efficient, experience substantially lower Lorentz forces, and can be driven by much lower currents than for whole body high-field systems. This enables relatively easy construction, using 3D printed holders to glue the wires into place. The gradient coils for the OSI^2 One have efficiencies around 0.5 mT/m/A, meaning that for typical imaging experiments no more than 10 A per gradient coil is needed. OSI^2 ONE currently uses a three-channel linear gradient amplifier able to deliver +/-15 A at approximately 12 V, with the voltage chosen so that the system can run off two car batteries for improved portability.[127,201] Other designs of linear amplifiers exist with a maximum output current of around 8 A per channel that can be used to drive gradients or B_0-shimming coils,[103,105,202–205] and experimental designs with 20 A/100 V[110,206] are also available.

For RF transmission, solenoidal coils wired on 3D printed forms can be used. Solenoids are more sensitive for low fields than other RF coil designs such as birdcage or saddle coils and are very simple to construct. The design files for a 20.5 cm diameter solenoid for OSI^2 ONE are available here.[126] The RF coil versus body noise dominance at low frequencies also implies a simpler receiver chain because coil arrays (> 2 receive channels) are less beneficial for increasing the signal-to-noise ratio. To amplify the RF pulses, several RF power amplifier designs with up to 1 kW
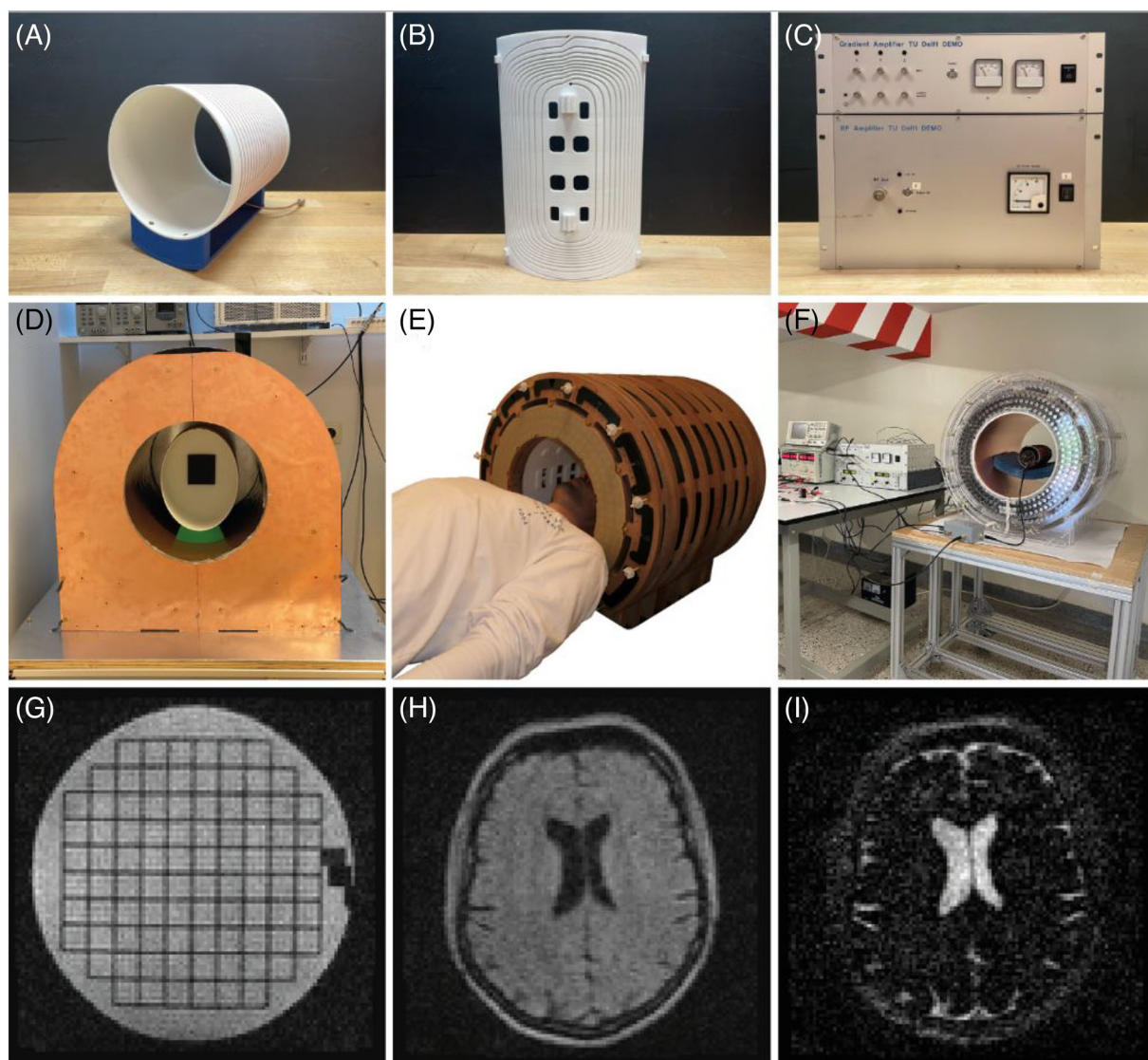
FIGURE 5 Photographs and components of the open-source low-field MR scanner OSI² ONE.[195] (A) Photograph of a 20.5-cm diameter solenoidal coil wound on a 3D printed former.[126] (B) One of four 3D printed gradient segments of the Y gradient. (C) Open-source gradient and RF amplifiers.[123,127] (A–C) have all been designed for the OSI² ONE. OSI² ONE scanners constructed at (D) The Leiden University Medical Center (LUMC) in Leiden, The Netherlands, (E) At the Physikalisch-Technische Bundesanstalt (PTB) in Berlin, Germany, and (F) At the Mbarara University in Mbarara, Uganda. (G) 3D turbo spin-echo (TSE) of the ACR phantom (FOV: $220 \times 220 \times 200$ mm^3, resolution: $2 \times 2 \times 2$ mm^3, TR/TE: 400/20 ms, ETL: 8, acquisition time: 10 min 5 s). (H) 3D TSE of an adult head (FOV: $220 \times 220 \times 225$ mm^3, resolution: $2 \times 2 \times 3$ mm^3, TR/TE: 600/15 ms, ETL: 7, elliptical k-space mask, acquisition time: 9 min 3 s). (I) 3D inversion recovery TSE of an adult head (FOV: $220 \times 220 \times 180$ mm^3, resolution: $2 \times 2 \times 3$ mm^3, TI/TR/TE: 200/6000/18 ms, ETL: 110, acquisition time: 6 min 0 s). A TI time of 200 ms was chosen to suppress signal from brain tissue. A center-out k-space trajectory was used with the first echo after 18 ms to maximize the CSF signal. All images have been corrected for B_0 distortions and gradient nonlinearity, no denoising has been applied.

of peak power with good linearity are available.[102,104,123,207] This power output is more than enough to drive the highly efficient low-frequency RF coils that generate much higher $B_1^+/\sqrt{kW}$ than their high-field counterparts, because component losses increase and tissue losses dominate at higher frequencies.

Regarding the console hardware, there are two open-source hardware-based NMR spectrometers with the capability to transmit RF and receive and digitize MR signals, the Opencore NMR[120,208] and LimeSDR.[109,209] However, to the best of our knowledge, no open-source console designs exist yet with the capabilities to control gradient pulses with the fidelity needed for imaging. Nonetheless, relatively affordable proprietary electronic boards have been used for MR imaging by dedicated open-source software packages such as gr-MRI,[57,210] OCRA,[75,211] and MAR-COS.[68,212] gr-MRI can be used with GNU Radio[213] compatible software-defined radios (SDRs), thus not limiting the software development to a single piece of hardware, being able to upgrade or change hardware designs without major modifications of the software. OCRA and MARCOS operate on the Red Pitaya hardware, which offers two independent transmit and receive channels and relies on external add-on boards to

generate the gradient waveforms.[106,118] A similar hardware-agnostic approach to gr-MRI is followed by the compatibility of MARCOS to Pulseq,[77,78] allowing independent and parallel pulse sequence and hardware developments, an important feature to successful open-source developments of a larger decentralized community.

In addition to these specific software solutions to drive the console hardware, more general software tools for scanner control, acquisitions, and postprocessing, such as Scanhub,[84,236] are being developed. Scanhub is a cloud-based generic and modular solution enabling integration of many of the aforementioned software packages (e.g., image reconstruction or postprocessing) to be used with actual MR scanner hardware such as the OSI^2 ONE system. These cloud-based platforms will gain further importance, beyond their core functionality to operate the scanner, by connecting users and experts from various backgrounds, exchanging imaging protocols and postprocessing routines while reducing the local computational requirements, or performing remote quality checks and maintenance of the scanners. An interesting aspect of the OSI^2 ONE project is the goal to also produce and provide documentation that follows the requirements for regulatory approval as a medical device according to EU regulation 2017/745. Such documentation, if publicly available, will greatly facilitate the translation from research prototypes to products, while simultaneously providing valuable documentation blueprints for other medical devices that are currently unavailable.

While the OSI^2 ONE project eventually targets in vivo imaging, other lower cost tabletop systems for NMR relaxometry[116,214] or fully capable MRI scanners[111,117,215] are successfully applied in the educational field (Figure 6). These tabletop scanners come with a full set of components to learn, understand, and perform MRI experiments of small volumes: magnet, console, RF and gradient coils, amplifiers, transmit/receive switches, GUI, pulse sequences, etc. Such educational systems offer stimulating practical learning experiences for a wide audience. Physics students can design pulse sequences to gain insight into the physical and technical principles that generate an MR image. Medical students can obtain hands-on experience of different contrast mechanisms of MRI and their interpretation. Chemistry students can exercise chemical analysis through relaxometry and spectroscopy techniques. Electrical engineering students can learn small signal, power, and digital electronics. Mathematics and Information Technology students can explore the information encoding/decoding power of linear algebra, information compression and machine learning with data generated by themselves.

3.1 | Open-source resources for MR safety, quality assurance, and novel applications

Within proprietary MR systems, customized hardware components are often needed to evaluate the feasibility of a novel application, investigate, and extend currently applied safety procedures, and advance science through novel experimental procedures. On the RF transmission side,
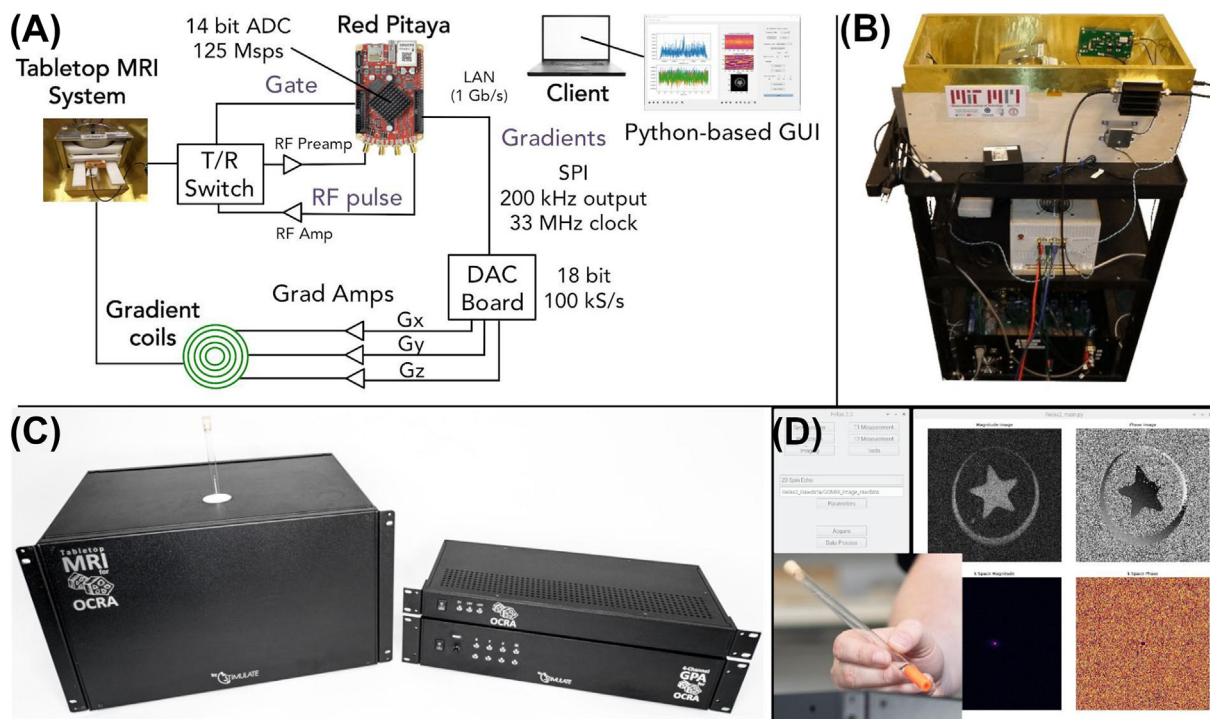

FIGURE 6 Two different educational tabletop systems. (A) Schematic, and (B) Photograph of the hardware setup of the instructional tabletop system from MGH/MIT.[111] (C) Photograph of the hardware, and (D) GUI and MR images from the OCRA Tabletop MRI System from Magdeburg University.[117]

particularly at ultrahigh-field (UHF) or for X-nuclei MRI, dedicated RF coils are often missing, leading to many customized research designs. Similarly, many research sites have already engaged in building their own phantoms. Recently, more open-source resources are becoming available, allowing easier reproduction, evaluation, and improvement by other groups.

For RF coils, an eight-channel transmit/receive design is available[99,216] to, for example, perform 7-T parallel transmission (pTx) experiments. A similar sized eight-channel RF coil was also constructed for 3 T, allowing an easy measurement-based comparison of pTx techniques between these field strengths.[98,217] Another scalable 3-T pTx RF coil design has been applied for ^{19}F imaging and could in principle also be tuned to ^{1}H frequency.[218,219]

To characterize and investigate the electromagnetic fields of these or similar RF coils, the open-source electromagnetic simulation package MARIE is available.[69,220] While MARIE might be an alternative to costly proprietary EM simulation software, other open-source software tools expand the functionality of proprietary software, such as CoSimPy,[51,221] which allows speedy circuit co-simulations (e.g., for fast RF coil optimizations). To homogenize B_1 for multichannel transmit RF coils, the shimming toolbox that implements static B_1 shimming routines can be used.[86,222] The shimming toolbox is not restricted to B_1 shimming; it can also be applied for real-time B_0 shimming and B_0 dynamic shimming using AC/DC coils, effectively operated by open-source hardware current drivers.[103,105,202]

Because parallel transmission hardware is not readily available, pTx systems have been assembled (e.g., for implant safety testing).[125,223] This setup allows a relatively flexible configuration of transmission and reception channels (up to 32 channels) over a wide frequency range, covering practically all relevant transmission frequencies from low-field to UHF MR at 7 T. Together with broadband RF power amplifiers and an open-
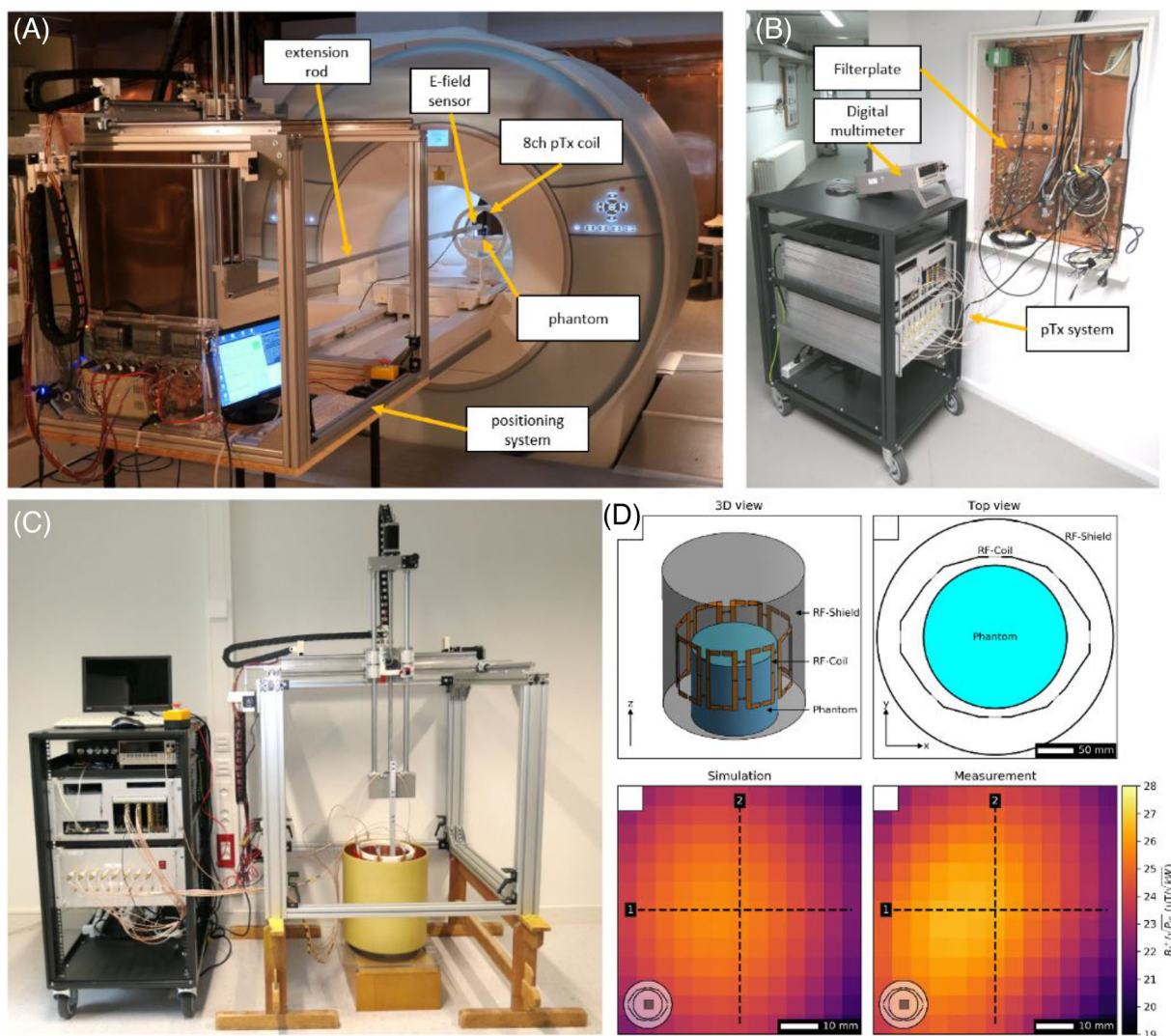

FIGURE 7 Open-source hardware for RF safety testing in and outside the MR scanner. (A and B) Parallel transmission medical implant safety testbed[125] used to measure E-fields inside the MRI system. (C and D) Parallel transmission (pTx) testbed setup outside the scanner to validate EM simulations by measuring B_1^+ with time-domain H-field probes at various locations using COSI Measure[101] in an open-source 7-T RF coil.[99] Figures adapted from previous works.[223]

source positioning system,[101,197] automated RF safety experiments can be performed outside or inside the MR scanner (Figure 7). This setup has been used to validate electromagnetic and temperature simulations, for RF safety measurements and sensor calibrations of implants. These configurations are extended by providing wireless reference implant hardware that can communicate with the pTx system over Bluetooth and exchange measured implant sensor information.[128,224] Other MR-compatible low-cost sensors have been presented, for example, wireless sensors for motion tracking,[114,225] or sensors to measure force during muscular exercises inside the MR scanner.[121,226]

Not all MR systems are equipped with highest end technology and specialized open-source hardware configurations can help to close this gap. A good example is a modular RF preamplifier/attenuator system, which was used to expand the initial four receive channels of a 4.7-T MRI to 32, potentially adjustable to other frequencies (Figure 8).[35,36] A similar approach that extends the standard performance of an available MR hardware configuration, enabling novel applications is, for example, a transmit/receive switch design capable of handling much higher average RF powers for thermal MR RF heating experiments.[108,228] Simple but useful open-source hardware tools for the laboratory or MR environment are, for example, RF field probes, which can be used to measure the frequency response of an RF coil, electromagnetic interferences, or RF shield attenuations.[107,113]

For phantoms, several designs are available. A very simple phantom, which can be assembled by mostly 3D printed parts, is the open-source NIST phantom.[119,229] Phantoms for very specific applications are also available to assess image distortions, for example, in image-guided interventions such as MR-guided radiation therapy,[115,230] 3D printed phantoms to evaluate contrast agent solutions,[112,231] for perfusion imaging,[100] or pneumatic phantoms to mimic respiration-induced artifacts.[124,232] A very useful tool to calculate gel ingredients for sucrose and polyvinylpyrrolidone (PVP)-based phantoms with a targeted permittivity and electric conductivity is available online.[233–235] While sucrose gives strong chemical shift artifacts, PVP is often preferred for imaging phantoms.

3.2 | Challenges and suggestions for open-source MRI

This section presents a summary of the observations we have made over the years regarding current open-source practices and areas that we believe require further improvement to enhance the generation of useful results.

3.2.1 | Communication and project management

Community is at the heart of open-source and its main workforce. Open-source projects are often inclusive, interdisciplinary, and intercultural. Communities are typically decentralized, and nonhierarchical, and community members often work voluntarily without monetary compensation and without strict deadlines. Because IP is organized as a common good, open-source technologies live independently from their originators or primarily distribution entities (e.g., companies or research institutes). Consequently, conventional communication and management structures with mostly clear responsibilities, goals, and decision-making hierarchies, are often missing. There is yet no standard solution for community governance. For large complex pioneer projects, such as open-source MRI technology, this needs to be piloted and iterated as the project grows. Today,
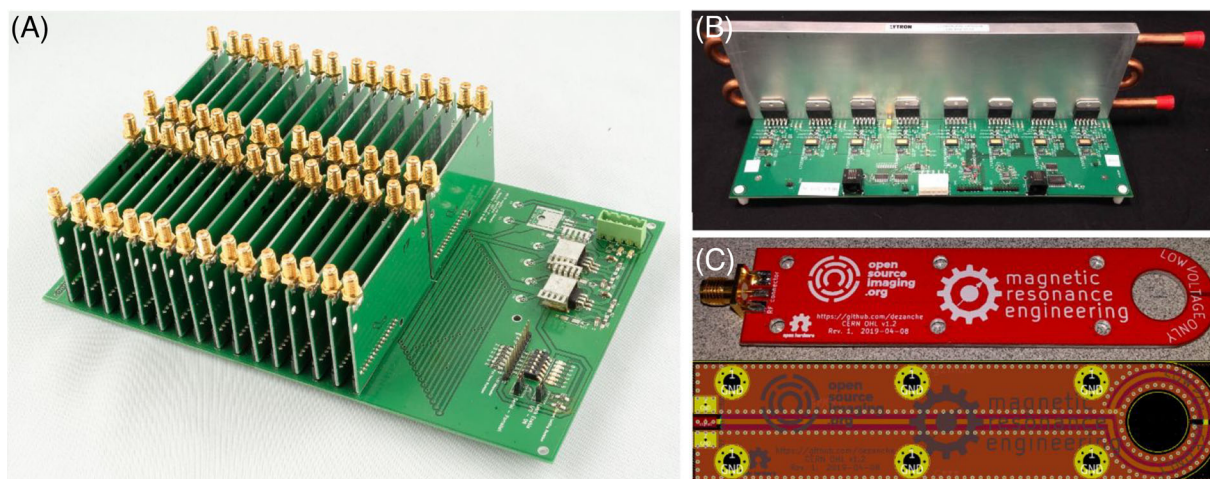

FIGURE 8 Open-source hardware for customized applications. (A) Modular 32-channel preamplifier/attenuator system.[36] (B) Low-cost eight-channel digitally programmable current driver with 8 A per channel to drive B_0-shimming coils.[103] (C) Simple RF probe for the laboratory in its manufactured form[107] and its PCB layout designed with the open-source software KiCAD.[227]

multiple software platforms support decentralized project development and community management (e.g., GitLab and OpenProject), but they are not yet fully streamlined for efficient use in a corporate scenario and also primarily cater to software development.

This knowledge helps to manage expectations of newcomers to the field who seek rapid results, and leave under the impression of slow, chaotic, and inefficient processes. Open source is not a sprint, it is a marathon, and so is the development of the underlying infrastructure.

3.2.2 | Documentation and licensing

Clear open-source licensing and complete, comprehensive documentation is crucial for the successful reuse of open-source resources. Numerous projects lack clear licensing terms, thus the use of these resources might lead to copyright infringement. In addition, the quality of documentation across open-source projects varies greatly. Particularly for open-source hardware, the stakes and the risks are much higher because monetary investment and time to build and test a device need to be invested. It is illusive that documentation standards will be unified across all technological disciplines. Therefore, technology-specific documentation criteria (TsDC, DIN SPEC 3105-1[19]) are defined and evaluated in standardization efforts to improve open-source hardware documentation. As experience shows, well documented projects can be reproduced without any or with minimal input from the original creator.

3.2.3 | Funding and infrastructure

Currently, there are few funding opportunities specifically available for open-source software projects, and even fewer (if any) for open-source hardware projects. Typically, researchers must find alternative paths by incorporating open-source technologies (e.g., as part of novel research proposals). Overall, more scientific results could be produced if essential proprietary technologies are available open source. A prerequisite for that is dedicated funding schemes. Other mechanisms to free designs would be to extend current open science guidelines and the requirements of public funding to open-source software and hardware.

In addition, no infrastructural support exists for developers after the release of a software or hardware. Even though extra efforts must be taken to create an open-source tool, the investment in the development and documentation of the tool is still reimbursed to some extent for the developer (e.g., through journal publications and increased visibility). However, after the release, there are no incentives and no support for ongoing development, bug fixes, maintenance, and user requests. This is particularly frustrating if the tool developed is successful and needed, because the developer is flooded with user requests. Institutional resources that exist, for example, for technology transfer and the costs associated with expensive patent applications or legal offices dealing with nondisclosure agreements, could be extended or partly transformed to strengthen open-source outputs.

3.2.4 | Open-source products

Currently, no, or very few products exist around the open-source projects presented in this review. To accelerate the distribution of technology and results produced with these, more participation is needed by the private sector. Not everyone wants and not everyone is able to reproduce open-source technology and there is strong interest in open-source products.

The challenge is that the principles of open source are difficult to conciliate with standard business models because protection of IP is a common practice to attract investment and to secure sufficient funds for product development. This is especially important for the commercialization of medical devices, which usually take several years to reach the market because of the lengthy certification processes. For such long development times, IP brings some security to investors, at the expense of slower development time, limited adoption, and dramatically reduced collaboration. In open-source–based business models the expertise on product development, support, and customization could be outsourced to the open-source community. Companies are curious but still hesitant to make the effort or take the risk to adapt or even connect with these new ways of value creation. To break the ice and accelerate the process, cooperation with innovation-driven businesses, which traditionally rely on IP restrictions, may need dedicated funding schemes.

3.2.5 | Medical devices

Open-source hardware cannot be used as a medical device on humans unless regulatory approval has been given. Regulatory approval and its documentation are lengthy processes that require substantial resources for companies. Currently, no blueprints are available to facilitate the roadmap towards regulatory approval. It has to be evaluated to what extent and in which documentation format open-source hardware and

software documentation can be aligned along the regulatory pathway. This would greatly ease the transition from research prototypes to actual products and would require less investment by the private sector. It is already difficult to establish open-source hardware documentation, let alone introduce documentation standards that go beyond current levels. This, together with the missing infrastructural support from institutions, seems to be a challenging task. However, based on observations from open-source software development over the last decades, professional documentation practices are being increasingly diffused into state-of-the art research. This trend was largely supported by dedicated software platforms (e.g., Github or Gitlab) allowing widespread collaboration, while enforcing certain documentation standards through built-in Wikis, project pages, git versioning, project management tools, etc. Software tools and blueprints to support documentation structures for regulatory approval are needed and would close the current gap between research prototypes and medical technologies used on patients.

4 | DISCUSSION AND CONCLUSION

Open-source practices and resources in MRI have substantially increased in recent years, indicating a clear trend towards a culture of sharing and establishing nonexclusive global collaborations to advance the field. These highly favorable advancements continue to enhance the reproducibility and reusability of code and designs, while providing a more inclusive approach to low-income settings. Despite these important steps, the documentation quality of current open-source projects is still strongly variable. Developers and scientists are mostly left alone in their efforts to make open-source releases, with little guidance on the publishing process, the documentation requirements, postrelease maintenance, and community management. Community-driven standards and initiatives are forming to address these infrastructural needs, but more dedicated support is needed (e.g., at the institutional level), by funding agencies, and by widely visible academic publishers. Here, in particular, open-source research hardware has the potential to substantially lower the cost and access to MR technology and, in close collaboration with the industry, enable an easier translation from research prototypes to actual products benefiting patients. Open source can be more than an accelerator for science, it can be the foundation for sustainable global collaboration in the development of medical devices, making MRI more accessible and transparent.

ORCID

Lukas Winter https://orcid.org/0000-0002-4381-275X
Christoph Kolbitsch https://orcid.org/0000-0002-4355-8368
Rita G. Nunes https://orcid.org/0000-0001-7425-5717

REFERENCES

1. Baker M. 1,500 scientists lift the lid on reproducibility. *Nature.* 2016;533(7604):452-454. doi:10.1038/533452a
2. Stikov N, Trzasko JD, Bernstein MA. Reproducibility and the future of MRI research. *Magn Reson Med.* 2019;82(6):1981-1983. doi:10.1002/mrm.27939
3. Karakuzu A, Biswas L, Cohen-Adad J, Stikov N. Vendor-neutral sequences and fully transparent workflows improve inter-vendor reproducibility of quantitative MRI. *Magn Reson Med.* 2022;88(3):1212-1228. doi:10.1002/mrm.29292
4. World Health Organization (WHO). Medical device donations: considerations for solicitation and provision. 2011. Accessed December 13, 2022. https://apps.who.int/iris/handle/10665/44568
5. World Health Organization (WHO). Baseline country survey on medical devices, 2014 update: Medical Equipment—total density per million population. Accessed January 7, 2023. http://gamapserver.who.int/gho/interactive_charts/health_technologies/medical_equipment/atlas.html
6. European Union (EU). Healthcare resource statistics—technical resources and medical technology. Accessed September 30, 2022. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Healthcare_resource_statistics_-_technical_resources_and_medical_technology
7. European Commission. Horizon. 2020 Online Manual. Accessed December 18, 2022. https://ec.europa.eu/research/participants/docs/h2020-funding-guide/cross-cutting-issues/open-access-data-management/open-access_en.htm
8. White House Office of Science and Technology Policy (OSTP). Desirable Characteristics of Data Repositories for Federally Funded Research. Executive Office of the President of the United States; 2022. 10.5479/10088/113528

9. Tollefson J, Van Noorden R. US government reveals big changes to open-access policy. *Nature*. 2022;609(7926):234-235. doi:10.1038/d41586-022-02351-1

10. European Commission. Open Source Software Strategy. 2020-2023. Think Open; 2020.

11. Open Source Initiative (OSI). Accessed December 18, 2022. https://opensource.org/about

12. Pearce JM. Return on investment for open source scientific hardware development. *Sci Public Policy*. 2016;43(2):192-195. doi:10.1093/scipol/scv034

13. Pearce JM. Economic savings for scientific free and open source technology: A review. *HardwareX*. 2020;8:e00139. doi:10.1016/j.ohx.2020.e00139

14. Gathering for Open Science Hardware (GOSH). Open Hardware: a key for accelerating Science and Technology towards the U.N. sustainable development goals (SDGS). 2021.

15. Free Software Foundation. Accessed December 18, 2022. https://www.fsf.org/about/

16. Open Source Hardware Association (OSHWA). Accessed December 18, 2022. https://www.oshwa.org/about/

17. Gathering for Open Science Hardware (GOSH). Published March 1, 2016. Accessed December 18, 2022. https://openhardware.science/about/

18. FORCE11. Guiding Principles for Findable, Accessible, Interoperable and Reusable Data Publishing version b1.0. 2022. https://force11.org/info/guiding-principles-for-findable-accessible-interoperable-and-re-usable-data-publishing-version-b1-0/

19. DIN SPEC 3105-1:2020-07—Open Source Hardware—Part 1: Requirements for technical documentation. *Beuth Publishing DIN*; 2020. https://www.din.de/en/wdc-beuth:din21:324805763

20. DIN SPEC 3105-2:2020 07—Open Source Hardware—Part 2: Community-Based Assessment. Beuth Publishing DIN; 2020. https://www.din.de/en/wdc-beuth:din21:324805750

21. CERN. CERN OHL version 2. Published 2020. Accessed August 24, 2021. https://ohwr.org/project/cernohl/wikis/Documents/CERN-OHL-version-2

22. The rise of open hardware at TU Delft. Accessed December 18, 2022. https://www.tudelft.nl/open-science/articles-tu-delft/the-rise-of-open-hardware-at-tu-delft

23. Black Duck by Synopsys. 2016 Future of Open Source Survey Results. Accessed December 18, 2022. https://de.slideshare.net/blackducksoftware/2016-future-of-open-source-survey-results

24. Winter L, Han H, Barghoorn A, et al. Open Source Imaging Initiative. *Proc Intl Soc Magn Reson Med*. 2016;3638.

25. Arndt F, Aussenhofer S, Behrens E, et al. Open source imaging initiative (OSI^2)—update and roadmap. In: *Proc Intl Soc Mag Reson Med*; 2017;4463.

26. Bell L, Henk M, Fedorov A, et al. The open source initiative for perfusion imaging (OSIPI). *Proc Intl Soc Mag Reson Med*. 2021;3820.

27. Open Science Initiative for Perfusion Imaging. Accessed December 20, 2022. https://www.osipi.org/index.html

28. MRSHub. Published April 9, 2020. Accessed December 20, 2022. https://mrshub.netlify.com/

29. RENALMRI. Accessed December 20, 2022. http://renalmri.org/

30. O'Reilly T. ISMRM workshop on low field MRI. In: *Flexible Magnet design at low field*; 2022.

31. Winter L. ISMRM workshop on low field MRI. In: *OSI^2 ONE: an open-source low-field MR scanner*; 2022.

32. Affordable low-field MRI reference system (A4IM). Accessed October 16, 2023. https://www.a4im.ptb.de/home

33. Obungoloch J, Muhumuza I, Teeuwisse W, et al. On-site construction of a point-of-care low-field MRI system in Africa. *NMR Biomed*. 36(7):e4917. doi:10.1002/nbm.4917

34. First open-source MRI scanner presented: the OSI^2 ONE. Accessed January 8, 2023. http://www.opensourceimaging.org/2023/01/08/first-open-source-mri-scanner-presented-the-osii-one/

35. 32-channel RF system. Published May 12, 2023. Accessed July 28, 2023. https://github.com/dezanche/32-channel_RF_system

36. 32-channel RF receive chain amplifiers. Accessed January 3, 2023. https://www.opensourceimaging.org/project/32-channel-rf-receive-chain-amplifiers/

37. Winter L, Pellicer-Guridi R, Broche L, et al. Open source medical devices for innovation, education and global health: case study of open source magnetic resonance imaging. In: *Co-Creation*. Springer; 2019:147-163. doi:10.1007/978-3-319-97788-1_12

38. Moritz M, Redlich T, Günyar S, Winter L, Wulfsberg JP. On the economic value of open source hardware—case study of an open source magnetic resonance imaging scanner. *Journal of Open Hardware*. 2019;3(1):2. doi:10.5334/joh.14

39. Pearce J. Distributed manufacturing of open-source medical hardware for pandemics. *Engineering*. 2020; 4(2), 49. doi:10.20944/preprints202004.0054.v1

40. Niso G, Botvinik-Nezer R, Appelhoff S, et al. Open and reproducible neuroimaging: From study inception to publication. *Neuroimage*. 2022;263:119623. doi:10.1016/j.neuroimage.2022.119623

41. eLife. eLife. Accessed July 22, 2023. https://elife-container.pubpub.org/about

42. MR-Pub. Accessed July 22, 2023. https://ismrm.github.io/mrpub/

43. NeuroLibre. Accessed July 22, 2023. https://neurolibre.org

44. 3D Slicer image computing platform. Accessed December 20, 2022. https://slicer.org/

45. Analysis of Functional NeuroImages (AFNI). Accessed December 20, 2022. https://afni.nimh.nih.gov/

46. Advanced Normalization Tools (ANTs). Accessed December 20, 2022. https://stnava.github.io/ANTs/

47. Advanced Normalization Tools in Python (AntsPy). Accessed December 20, 2022. https://github.com/ANTsX/ANTsPy

48. BART Toolbox. Accessed December 18, 2022. https://mrirecon.github.io/bart/

49. pyCoilGen. GitHub. Accessed October 16, 2023. https://github.com/kev-m/pyCoilGen/

50. CoilGen. December 2022. Accessed December 21, 2022. https://github.com/Philipp-MR/CoilGen

51. CoSimPy: An open-source python library for MRI radiofrequency Coil EM/Circuit Cosimulation. Accessed January 3, 2023. https://umbertozanovello.github.io/CoSimPy/

52. Deep Anatomical Federated Network (Dafne). Accessed December 21, 2022. https://dafne.network/

53. EPTlib—an open-source, extensible C++ library of electric properties tomography methods. Accessed December 20, 2022. https://eptlib.github.io/

54. erwin. Accessed December 21, 2022. https://erwin.readthedocs.io/en/latest/

55. FreeSurfer. Accessed December 20, 2022. https://surfer.nmr.mgh.harvard.edu

56. Gadgetron. https://gadgetron.github.io/

57. GNURadio-MRI. Accessed July 27, 2023. https://bitbucket.org/wgrissom/gnuradio-mri/src/master/
58. LUMC-LowFieldMRI. HalbachOptimisation. July 2023. Accessed July 27, 2023. https://github.com/LUMC-LowFieldMRI/HalbachOptimisation
59. A toolbox for quantitative MRI and in vivo histology using MRI (hMRI). Accessed December 21, 2022. https://hmri-group.github.io/hMRI-toolbox/
60. ImageJ. Accessed December 20, 2022. https://imagej.nih.gov/ij/
61. ISMRMRD. Accessed December 18, 2022. https://github.com/ismrmrd
62. ITK. The Insight Toolkit. Accessed December 20, 2022. https://github.com/InsightSoftwareConsortium/ITK
63. JEMRIS. Accessed December 23, 2022. https://www.jemris.org/
64. Keras: deep learning for humans. Accessed December 21, 2022. https://github.com/keras-team/keras
65. KomaMRI.jl: an open-source framework for general MRI simulations with GPU acceleration. Accessed July 22, 2023. https://github.com/cncastillo/KomaMRI.jl
66. LUMC. Gradient Design Tool. Accessed December 21, 2022. https://github.com/LUMC-LowFieldMRI/GradientDesignTool
67. QBI MADYM. Accessed December 21, 2022. https://gitlab.com/manchester_qbi/manchester_qbi_public/madym_cxx
68. MARCOS. Accessed December 19, 2022. https://github.com/vnegnev
69. MARIE: MAgnetic Resonance Integral Equation suite. Accessed July 27, 2023. https://github.com/thanospol/MARIE
70. Medical Image Registration ToolKit (MIRTK). Accessed January 3, 2023. https://mirtk.github.io/
71. Medical Open Network for Artificial Intelligence (MONAI). Accessed January 3, 2023. https://monai.io/
72. MRILab. —A numerical MRI simulator. Accessed July 22, 2023. https://leoliuf.github.io/MRiLab/
73. MRIReco. Accessed December 20, 2022. https://github.com/MagneticResonanceImaging/MRIReco.jl
74. NiftyReg. Accessed January 3, 2023. https://github.com/KCL-BMEIS/niftyreg
75. OCRA MRI. Accessed December 18, 2022. https://openmri.github.io/ocra/
76. ODIN. —Object Oriented Development Interface for NMR. Accessed July 27, 2023. https://od1n.sourceforge.net/index.html
77. Open-source pulse sequences: Pulseq. Accessed December 23, 2022. https://pulseq.github.io/
78. PyPulseq. A Python Package for MRI Pulse Sequence Design. Accessed December 23, 2022. https://github.com/imr-framework/pypulseq
79. PyQMRI. March 2023. Accessed July 27, 2023. https://github.com/IMTtugraz/PyQMRI
80. PyTorch. Accessed January 3, 2023. https://www.pytorch.org
81. QMRITools. Accessed July 27, 2023. https://www.qmritools.com/
82. qMRLab. Accessed December 20, 2022. http://qmrlab.org/
83. Quantitative imaging tools (QUIT). Accessed December 21, 2022. https://github.com/spinicist/QUIT
84. Scanhub. Accessed January 3, 2023. https://github.com/brain-link/scanhub-ui
85. Susceptibility pipeline tool for phase images (SEPIA). Accessed December 20, 2022. https://github.com/kschan0214/sepia
86. Shimming toolbox. Accessed January 7, 2023. https://shimming-toolbox.org
87. Sigpy MRI RF Design. Accessed January 6, 2023. https://sigpy.readthedocs.io/en/latest/mri_rf.html
88. SimpleElastix. —Medical Image Registration Library. Accessed January 5, 2023. https://simpleelastix.github.io/
89. Synergistic Image Reconstruction Framework (SIRF). Accessed January 4, 2023. https://github.com/SyneRBI/SIRF
90. Spinach. Accessed December 23, 2022. https://spindynamics.org/
91. SPM—Statistical Parametric Mapping. Accessed December 21, 2022. https://www.fil.ion.ucl.ac.uk/spm/
92. Sebastian-MR. ThinWire_MRIGradientCoilDesign. September 2022. Accessed December 21, 2022. https://github.com/Sebastian-MR/ThinWire_MRIGradientCoilDesign
93. TOPPE. —A framework for rapid prototyping of MR pulse sequences. Accessed December 23, 2022. https://toppemri.github.io/
94. VeSPA. Accessed January 3, 2023. https://vespa-mrs.github.io/vespa.io/
95. Virtual Scanner. Accessed January 3, 2023. https://github.com/imr-framework/virtual-scanner
96. Yarra Framework. Accessed January 5, 2023. http://yarra-framework.org/
97. Scalable 19 RF Coil. February 2023. Accessed July 28, 2023. https://github.com/ozenEEE/UKF_19FCOIL
98. PTB MRI Lab|8ch pTx headcoil 3T. Published December 8, 2022. Accessed July 28, 2023. https://gitlab1.ptb.de/mri-lab/8ch-ptx-headcoil-3t
99. PTB MRI Lab|8-channel transmit/receive headcoil for 7T. Accessed December 21, 2022. https://gitlab1.ptb.de/mri-lab/8ch-ptx-headcoil-7t
100. ASL Phantom. Accessed December 20, 2022. https://github.com/mgabr001-git/ASL_Phantom_3D_DesignFiles
101. COSI Measure. Accessed May 23, 2022. https://www.opensourceimaging.org/project/cosi-measure/
102. COSI-Transmit. May 2023. Accessed July 28, 2023. https://github.com/opensourceimaging/cosi-transmit
103. Current Driver for Local B0-Shim Coils. Accessed December 19, 2022. https://www.opensourceimaging.org/project/current-driver-for-local-b0-shim-coils/
104. EASY_MRI_RFPA. Published May 8, 2022. Accessed December 21, 2022. https://github.com/mtwieg/EASY_MRI_RFPA
105. GPA-FHDO. Accessed December 19, 2022. https://www.opensourceimaging.org/project/gpa-fhdo/
106. GPA-RP-Adapter. Accessed January 4, 2023. https://github.com/menkueclab/GPA-RP-Adapter
107. H-field RF probe. Accessed December 21, 2022. https://github.com/dezanche/H-field_RF_probe
108. High Peak/Average Power TR-Switch for Thermal MR. Accessed January 4, 2023. https://www.opensourceimaging.org/project/thermal-mr-tr-switch/
109. LimeSDR Spectrometer. Accessed January 3, 2023. https://www.opensourceimaging.org/project/limesdr-spectrometer/
110. Maker-Gradient-Amplifier. Accessed December 19, 2022. https://github.com/mtwieg/Maker-Gradient-Amplifier
111. Instructional tabletop MRI scanner. Accessed January 3, 2023. https://www.opensourceimaging.org/project/instructional-tabletop-mri-scanner/
112. Preclinical Phantom for Quantitative Solution Comparison. Accessed December 19, 2022. https://morgridge.org/research/labs/fablab/designs/preclinical-phantom-for-quantitative-solution-comparison/
113. Kretov. MRI Cage test set. Published June 11, 2022. Accessed July 28, 2023. https://github.com/Astrei/MRI_Cage_test_set
114. MRI-compatible-sensors. Accessed July 28, 2023. https://github.com/nyu-cbi-sensors/MRI-compatible-sensors
115. MRI distortion phantom: a large-volume, lightweight, modular MRI phantom with solid signal sources. Accessed July 28, 2023. https://github.com/dezanche/MRI_distortion_phantom

116. NMR relaxometry platform. Published December 6, 2021. Accessed July 28, 2023. https://github.com/mtwieg/NMR

117. OCRA. —Open-source console for real-time acquisition. Accessed May 3, 2021. https://zeugmatographix.org/ocra/

118. OCRA1—SPI controlled 4 channel 18bit DAC and RF Attenuator. Accessed January 4, 2023. https://zeugmatographix.org/ocra/2020/11/27/ocra1-spi-controlled-4-channel-18bit-dac-and-rf-attenutator/

119. Open Source NIST Phantom. Accessed December 19, 2022. https://github.com/kalinared/open-source-nist-phantom

120. Opencore NMR. Accessed January 3, 2023. https://www.opensourceimaging.org/project/opencore-nmr/

121. OpenForceMR: an MR-Compatible force sensor composed of cheap, off-the shelf components. Accessed July 28, 2023. https://github.com/fsantini/OpenForceMR

122. OSI2 ONE Magnet. GitLab. Published May 7, 2022. Accessed July 26, 2023. https://gitlab.com/osii-one/magnet/lumc-30cm-v2/30cm-halbach-magnet

123. TU Delft 1kW peak RFPA. Accessed December 21, 2022. https://gitlab.com/osii-one/rf-system/rf-power-amplifier/1kw-peak-rfpa

124. Pneumatic Phantom for Mimicking Respiration-Induced MRI Artifacts. Accessed December 21, 2022. https://www.opensourceimaging.org/project/pneumatic-phantom-for-mimicking-respiration-induced-artifacts/

125. PTB MRI Lab|pTx Implant Safety Testbed. Published January 11, 2023. Accessed July 28, 2023. https://gitlab1.ptb.de/mri-lab/ptx-implant-safety-testbed

126. Solenoid RF coil. Accessed January 6, 2023. https://gitlab.com/osii-one/rf-system/rf-coils/solenoid-d205mm-n20-3seg

127. TU Delft GA2. Accessed December 19, 2022. https://gitlab.com/osii-one/gradient-system/gradient-power-amplifier/tu-delft-15a/tu-delft-ga2

128. PTB MRI Lab|Reference wireless implant. Published May 12, 2023. Accessed July 28, 2023. https://gitlab1.ptb.de/mri-lab/reference-wireless-implant

129. GNU Project—Free Software Foundation. What is Free Software? Accessed December 18, 2022. https://www.gnu.org/philosophy/free-sw.en.html

130. Software Package Data Exchange (SPDX) License List. Accessed December 18, 2022. https://spdx.org/licenses/

131. Open Source Hardware Association (OSHWA). Open Source Hardware Definition (English). Published May 26, 2012. Accessed April 1, 2019. https://www.oshwa.org/definition/

132. Svorc J and Katz A. Breathe In, Breathe Out: How Open Hardware Licensing Can Help Save the World. *J Open Law Tech Soc.* 2020;11(1):49.

133. Mies R, Häuer M, Hassan M. Introducing readiness scales for effective reuse of open source hardware. *Procedia CIRP.* 2022;109:635-640. doi:10.1016/j.procir.2022.05.306

134. GitHub. Accessed December 18, 2022. https://github.com/

135. Atlassian. Bitbucket. Accessed December 18, 2022. https://bitbucket.org/product

136. GitLab. Accessed December 18, 2022. https://about.gitlab.com/

137. Python. Accessed December 18, 2022. https://www.python.org/

138. Stöcker T, Vahedipour K, Pflugfelder D, Shah NJ. High-performance computing MRI simulations. *Magn Reson Med.* 2010;64(1):186-193. doi:10.1002/mrm.22406

139. Fortin A, Salmon S, Baruthio J, Delbany M, Durand E. Flow MRI simulation in complex 3D geometries: application to the cerebral venous network. *Magn Reson Med.* 2018;80(4):1655-1665. doi:10.1002/mrm.27114

140. Veldmann M, Ehses P, Chow K, Nielsen JF, Zaitsev M, Stöcker T. Open-source MR imaging and reconstruction workflow. *Magn Reson Med.* 2022;88(6):2395-2407. doi:10.1002/mrm.29384

141. Liu F, Velikina JV, Block WF, Kijowski R, Samsonov AA. Fast realistic mri simulations based on generalized multi-pool exchange tissue model. *IEEE Trans Med Imaging.* 2017;36(2):527-537. doi:10.1109/TMI.2016.2620961

142. Castillo-Passi C, Coronado R, Varela-Mattatall G, Alberola-López C, Botnar R, Irarrazaval P. KomaMRI.jl: An open-source framework for general MRI simulations with GPU acceleration. *Magn Reson Med.* 2023;90(1):329-342. doi:10.1002/mrm.29635

143. Layton KJ, Kroboth S, Jia F, et al. Pulseq: A rapid and hardware-independent pulse sequence prototyping framework. *Magn Reson Med.* 2017;77(4):1544-1552. doi:10.1002/mrm.26235

144. Nielsen JF, Noll DC. TOPPE: A framework for rapid prototyping of MR pulse sequences. *Magn Reson Med.* 2018;79(6):3128-3134. doi:10.1002/mrm.26990

145. Ravi KS, Geethanath S, Vaughan JT. PyPulseq: A python package for mri pulse sequence design. *J Open Source Softw.* 2019;4(42):1725. doi:10.21105/joss.01725

146. Tong G, Geethanath S, Jimeno M, et al. Virtual Scanner: MRI on a Browser. *J Open Source Softw.* 2019;4(43):1637. doi:10.21105/joss.01637

147. Fillmer A, Qiuyun F, Fang L, Winter L. MR from a global perspective: the underrepresented part of the MR spectrum. In: ISMRM. 2019.

148. Hogben HJ, Krzystyniak M, Charnock GTP, Hore PJ, Kuprov I. Spinach—a software library for simulation of spin dynamics in large spin systems. *J Magn Reson.* 2011;208(2):179-194. doi:10.1016/j.jmr.2010.11.008

149. Soher BJ, Semanchuk P, Todd D, et al. Vespa: integrated applications for RF pulse design, spectral simulation and MRS data analysis. *Magn Reson Med.* 2023;90(3):823-838. doi:10.1002/mrm.29686

150. Martin JB, Ong F, Ma J, Tamir JI, Lustig M, Grissom WA. SigPy.RF: comprehensive open-source RF pulse design tools for reproducible research. In: *Proc. 28th annual meeting of ISMRM*; 2020.

151. Inati SJ, Naegele JD, Zwart NR, et al. ISMRM Raw data format: a proposed standard for MRI raw datasets. *Magn Reson Med.* 2017;77(1):411-421. doi:10.1002/mrm.26089

152. Hansen MS, Sørensen TS. Gadgetron: An open source framework for medical image reconstruction. *Magn Reson Med.* 2013;69(6):1768-1776. doi:10.1002/mrm.24389

153. Xue H, Inati S, Sørensen TS, Kellman P, Hansen MS. Distributed MRI reconstruction using gadgetron-based cloud computing. *Magn Reson Med.* 2015;73(3):1015-1025. doi:10.1002/mrm.25213

154. Uecker M, Ong F, Tamir J, et al. Berkeley advanced reconstruction toolbox. *Proc Intl Soc Mag Reson Med.* 2015;2486.

155. Ovtchinnikov E, Brown R, Kolbitsch C, et al. SIRF: Synergistic Image Reconstruction Framework. *Comput Phys Commun.* 2020;249:107087. doi:10.1016/j.cpc.2019.107087

156. Thielemans K, Tsoumpas C, Mustafovic S, et al. STIR: software for tomographic image reconstruction release 2. *Phys Med Biol.* 2012;57(4):867-883. doi:10.1088/0031-9155/57/4/867

157. Core Imaging Library—Part I: a versatile Python framework for tomographic imaging|Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. Accessed December 18, 2022. doi:10.1098/rsta.2020.0192

158. Papoutsellis E, Ametova E, Delplancke C, et al. Core Imaging Library—part II: multichannel reconstruction for dynamic and spectral tomography. *Phil Trans Royal Soc A: Math Phys Eng Sci.* 2021;379(2204):20200193. doi:10.1098/rsta.2020.0193

159. Brown R, Kolbitsch C, Delplancke C, et al. Motion estimation and correction for simultaneous PET/MR using SIRF and CIL. *Phil Trans Royal Soc A: Math Phys Eng Sci.* 2021;379(2204):20200208. doi:10.1098/rsta.2020.0208

160. Knopp T, Grosser M. MRIReco.jl: An MRI reconstruction framework written in Julia. *Magn Reson Med.* 2021;86(3):1633-1646. doi:10.1002/mrm.28792

161. Avants BB, Tustison NJ, Stauffer M, Song G, Wu B, Gee JC. The Insight ToolKit image registration framework. *Front Neuroinformatics.* 2014;8:8. Accessed December 20, 2022. doi:10.3389/fninf.2014.00044

162. van Dalen JA, Vogel W, Huisman H, Oyen WJ, Jager GJ, Karssemeijer N. Accuracy of rigid CT–FDG-PET image registration of the liver. *Phys Med Biol.* 2004;49(23):5393-5405. doi:10.1088/0031-9155/49/23/014

163. Van Ginneken B, Stegmann MB, Loog M. Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database. *Med Image Anal.* 2006;10(1):19-40. doi:10.1016/j.media.2005.02.002

164. Floca R, Dickhaus H. A flexible registration and evaluation engine (f.r.e.e.). *Comput Methods Programs Biomed.* 2007;87(2):81-92. doi:10.1016/j.cmpb.2007.04.009

165. Chen M, Lu W, Chen Q, Ruchala KJ, Olivera GH. A simple fixed-point approach to invert a deformation fielda. *Med Phys.* 2008;35(1):81-88. doi:10.1118/1.2816107

166. Cheung MR, Krishnan K. Interactive deformation registration of endorectal prostate MRI using ITK thin plate splines. *Acad Radiol.* 2009;16(3):351-357. doi:10.1016/j.acra.2008.09.011

167. Lowekamp B, Gabehart, Blezek D, et al. SimpleElastix: SimpleElastix v0.9.0. June 2015. doi:10.5281/zenodo.19049

168. Avants BB, Tustison NJ, Song G, Cook PA, Klein A, Gee JC. A reproducible evaluation of ANTs similarity metric performance in brain image registration. *Neuroimage.* 2011;54(3):2033-2044. doi:10.1016/j.neuroimage.2010.09.025

169. Cox RW. AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. *Comput Biomed Res.* 1996;29(3):162-173. doi:10.1006/cbmr.1996.0014

170. Chen G, Shin YW, Taylor PA, et al. Untangling the relatedness among correlations, part I: Nonparametric approaches to inter-subject correlation analysis at the group level. *Neuroimage.* 2016;142:248-259. doi:10.1016/j.neuroimage.2016.05.023

171. Chen G, Taylor PA, Shin YW, Reynolds RC, Cox RW. Untangling the relatedness among correlations, Part II: Inter-subject correlation group analysis through linear mixed-effects modeling. *Neuroimage.* 2017;147:825-840. doi:10.1016/j.neuroimage.2016.08.029

172. Chen G, Taylor PA, Qu X, et al. Untangling the relatedness among correlations, part III: Inter-subject correlation analysis through Bayesian multilevel modeling for naturalistic scanning. *Neuroimage.* 2020;216:116474. doi:10.1016/j.neuroimage.2019.116474

173. Fischl B. FreeSurfer. *NeuroImage.* 2012;62(2):774-781. doi:10.1016/j.neuroimage.2012.01.021

174. Shirly S, Ramesh K. Review on 2D and 3D MRI image segmentation techniques. *Current Medical Imaging.* 2019;15(2):150-160. doi:10.2174/1573405613666171123160609

175. 3D Slicer as an image computing platform for the Quantitative Imaging Network—ScienceDirect. Accessed January 7, 2023. Accessed December 20, 2022. https://www.sciencedirect.com/science/article/abs/pii/S0730725X12001816

176. SlicerElastix. https://github.com/lassoan/SlicerElastix

177. Surgical Planning In Stereotaxy—NAMIC Wiki. Accessed January 7, 2023. https://www.na-mic.org/wiki/Project_Week_25/Surgical_Planning_In_Stereotaxy

178. Cardoso MJ, Li W, Brown R, et al. MONAI: an open-source framework for deep learning in healthcare. arXiv:2211.020701. 2022. doi:10.48550/arXiv.2211.02701

179. Arduino A. EPTlib: an open-source extensible collection of electric properties tomography techniques. *Appl Sci.* 2021;11(7):3237. doi:10.3390/app11073237

180. Berks M, Parker G, Little R, Cheung S. Madym: a C++ toolkit for quantitative DCE-MRI analysis. *J Open Source Softw.* 2021;6(66):3523. doi:10.21105/joss.03523

181. Chan KS, Marques JP. SEPIA—Susceptibility mapping pipeline tool for phase images. *Neuroimage.* 2021;227:117611. doi:10.1016/j.neuroimage.2020.117611

182. Karakuzu A, Boudreau M, Duval T, et al. qMRLab: Quantitative MRI analysis, under one umbrella. *J Open Source Softw.* 2020;5(53):2343. doi:10.21105/joss.02343

183. Karakuzu A, Appelhoff S, Auer T, et al. qMRI-BIDS: An extension to the brain imaging data structure for quantitative magnetic resonance imaging data. *Sci Data.* 2022;9(1):517. doi:10.1038/s41597-022-01571-4

184. Gorgolewski KJ, Auer T, Calhoun VD, et al. The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Sci Data.* 2016;3(1):160044. doi:10.1038/sdata.2016.44

185. Tabelow K, Balteau E, Ashburner J, et al. hMRI—a toolbox for quantitative MRI in neuroscience and clinical research. *Neuroimage.* 2019;194:191-210. doi:10.1016/j.neuroimage.2019.01.029

186. Froeling M. QMRTools: a Mathematica toolbox for quantitative MRI analysis. *J Open Source Softw.* 2019;4(38):1204. doi:10.21105/joss.01204

187. Maier O, Spann S, Bödenler M, Stollberger R. PyQMRI: An accelerated Python based Quantitative MRI toolbox. *J Open Source Softw.* 2020;5(56):2727. doi:10.21105/joss.02727

188. Wood CT. QUIT: QUantitative Imaging Tools. *J Open Source Softw.* 2018;3(26):656. doi:10.21105/joss.00656

189. Wang X, Tan Z, Scholand N, Roeloffs V, Uecker M. Physics-based reconstruction methods for magnetic resonance imaging. *Phil Trans Royal Soc A.* 2021;379(2200):20200196. doi:10.1098/rsta.2020.0196

190. O'Reilly T, Teeuwisse WM, Webb AG. Three-dimensional MRI in a homogenous 27 cm diameter bore Halbach array magnet. *J Magn Reson.* 2019;307:106578. doi:10.1016/j.jmr.2019.106578

191. Cooley CZ, McDaniel PC, Stockmann JP, et al. A portable scanner for magnetic resonance imaging of the brain. *Nature Biomed Eng.* 2020;5(3):229-239. doi:10.1038/s41551-020-00641-5

192. Guallart-Naval T, Algarín JM, Pellicer-Guridi R, et al. Portable magnetic resonance imaging of patients indoors, outdoors and at home. Scientific Reports. 2022. 121:13147.

193. Liu Y, Leong ATL, Zhao Y, et al. A low-cost and shielding-free ultra-low-field brain MRI scanner. *Nat Commun*. 2021;12(1):7238. 10.1038/s41467-021-27317-1

194. Shah J, Cahn B, By S, et al. Portable, bedside, low-field magnetic resonance imaging in an intensive care setting for intracranial hemorrhage. *Neurology*. 2020;94(15 Supplement):270.

195. OSI^2 ONE. Accessed May 23, 2022. https://gitlab.com/osii

196. O'Reilly T, Teeuwisse WM, de Gans D, Koolstra K, Webb AG. In vivo 3D brain and extremity MRI at 50 mT using a permanent magnet Halbach array. *Magn Reson Med*. 2021;85(1):495-505. doi:10.1002/mrm.28396

197. Han H, Moritz R, Oberacker E, Waiczies H, Niendorf T, Winter L. Open source 3D multipurpose measurement system with submillimetre fidelity and first application in magnetic resonance. *Sci Rep*. 2017;7(1):13452. doi:10.1038/s41598-017-13824-z

198. Wenzel K, Alhamwey H, O'Reilly T, Riemann LT, Silemek B, Winter L. B0-shimming methodology for affordable and compact low-field magnetic resonance imaging magnets. *Front Phys*. 2021;9. doi:10.3389/fphy.2021.704566

199. O'Reilly T, Teeuwisse W, Webb A. A new approach to shimming Halbach arrays using higher order Halbach Array Inserts. *Proc Intl Soc Mag Reson Med*. 2022;63.

200. Amrein P, Jia F, Zaitsev M, Littin S. CoilGen: Open-source MR coil layout generator. *Magn Reson Med*. 2022;88(3):1465-1479. doi:10.1002/mrm.29294

201. de Vos B, Parsa J, Abdulrazaq Z, et al. Design, characterisation and performance of an improved portable and sustainable low-field MRI system. *Front Phys*. 2021;9. doi:10.3389/fphy.2021.701157

202. Arango N, Stockmann J, Witzel T, Wald L, White JK. Open-source, low-cost, flexible, current feedback-controlled driver circuit for local B0 shim coils and other applications. Proc Intl Soc Mag Reson Med. 1157.

203. Guallart-Naval T, O'Reilly T, Algarín JM, et al. Benchmarking the performance of a low-cost magnetic resonance control system at multiple sites in the open MaRCoS community. *NMR Biomed*. 2022;36(1):e4825. doi:10.1002/nbm.4825

204. Small Scale 600W Gradient Power Amplifier—OCRA. Accessed December 19, 2022. https://zeugmatographix.org/ocra/2021/12/15/small-scale-600w-gradient-power-amplifier/

205. Twieg M. The RF & Gradient amplifiers. Proc Intl Soc Mag Reson Med. E8829.

206. Twieg M. Gradient controller tutorial. Proc Intl Soc Mag Reson Med. 2018.

207. Blücher C, Han H, Hoffmann W, et al. COSI Transmit: open source soft- and hardware transmission system for traditional and rotating MR. Proc Intl Soc Mag Reson Med. 2017;184.

208. Takeda K. Highly customized NMR systems using an open-resource, home-built spectrometer. In: *Annual Reports on NMR Spectroscopy*. Vol.74. Elsevier; 2011:355-393. doi:10.1016/B978-0-08-097072-1.00007-8

209. Michal CA. A low-cost multi-channel software-defined radio-based NMR spectrometer and ultra-affordable digital pulse programmer. *Conc Mag Reson B: Mag Reson Eng*. 2018;48B(3):e21401. doi:10.1002/cmr.b.21401

210. Hasselwander CJ, Cao Z, Grissom WA. gr-MRI: A software package for magnetic resonance imaging using software defined radios. *J Magn Reson*. 2016;270:47-55. doi:10.1016/j.jmr.2016.06.023

211. Anand S, Stockmann JP, Wald LL, Witzel T. A low-cost (<$500 USD) FPGA-based console capable of real-time control. *Proc Int Soc Magn Reson Med*. 2018;948.

212. Negnevitsky V, Vives-Gilabert Y, Algarín JM, et al. MaRCoS, an open-source electronic control system for low-field MRI. 2022. Accessed December 19, 2022. http://arxiv.org/abs/2208.01616

213. GNU Radio—The Free & Open Source Radio Ecosystem. Accessed December 18, 2022. https://www.gnuradio.org/

214. Twieg M, Riffe M, Gudino N, Griswold MA. An open source mobile nmr relaxometry platform. *Proc Int Soc Magn Reson Med*. 2013; 139.

215. Cooley CZ, Stockmann JP, Witzel T, et al. Design and implementation of a low-cost, tabletop MRI scanner for education and research prototyping. *J Magn Reson*. 2020;310:106625. doi:10.1016/j.jmr.2019.106625

216. Seifert F, Pfeiffer H, Mekle R, Waxmann P, Ittermann B. 7T 8-Channel PTx head coil with high B1+ efficiency optimized for MRS. *Proc Intl Soc Mag Reson Med*. 2016;3545.

217. Aigner C, Winding M, Winter L, et al. An open-source, low-cost 8-channel transmit/receive 3T head coil for pTx developments, MR safety testing and standardization. *Proc Intl Soc Magn Reson Med*. 2023;2695.

218. Özen AC, Spreter F, Schimpf W, et al. Scalable and modular 8-channel transmit and 8-channel flexible receive coil array for 19F MRI of large animals. *Magn Reson Med*. 2023;89(3):1237-1250. doi:10.1002/mrm.29490

219. Scalable and Modular RF Coil. Accessed December 18, 2022. https://www.opensourceimaging.org/project/scalable-and-modular-rf-coils/

220. Villena JF, Polimeridis AG, Eryaman Y, et al. Fast electromagnetic analysis of MRI transmit RF coils based on accelerated integral equation methods. *IEEE Trans Biomed Eng*. 2016;63(11):2250-2261. doi:10.1109/TBME.2016.2521166

221. Zanovello U, Seifert F, Bottauscio O, Winter L, Zilberti L, Ittermann B. CoSimPy: An open-source python library for MRI radiofrequency Coil EM/-Circuit Cosimulation. *Comput Methods Programs Biomed*. 2022;216:106684. doi:10.1016/j.cmpb.2022.106684

222. D'Astous A, Cereza G, Papp D, et al. Shimming toolbox: an open-source software toolbox for B0 and B1 shimming in MRI. *Magn Reson Med*. 2022; 89(4):1401-1417. doi:10.1002/mrm.29528

223. Winter L, Silemek B, Petzold J, et al. Parallel transmission medical implant safety testbed: Real-time mitigation of RF induced tip heating using time-domain E-field sensors. *Magn Reson Med*. 2020;84(6):3468-3484. doi:10.1002/mrm.28379

224. Silemek B, Seifert F, Petzold J, Brühl R, Ittermann B, Winter L. Wirelessly interfacing sensor-equipped implants and MR scanners for improved safety and imaging. *Magn Reson Med*. 2023;90(6):2608-2626. doi:10.1002/mrm.29818

225. Pancoast L, Brantner D, Wiggins R, Walczyk J, Brown R. Wireless body sensor data acquisition platform for motion tracking. *Proc Int Soc Magn Reson Med*. 2021;566.

226. Santini F, Bieri O, Deligianni X. OpenForce MR: a low-cost open-source MR-compatible force sensor. *Conc Mag Reson B: Mag Reson Eng*. 2018;48B (4):e21404. doi:10.1002/cmr.b.21404

227. KiCad EDA. Accessed January 6, 2023. https://www.kicad.org/

228. Ji Y, Hoffmann W, Pham M, et al. High peak and high average radiofrequency power transmit/receive switch for thermal magnetic resonance. *Magn Reson Med.* 2018;80(5):2246-2255. doi:10.1002/mrm.27194

229. Jordanova K The phantom. *Proc Intl Soc Mag Reson Med*. 2022:E8570.

230. Gardiner M, Wachowicz K, De Zanche N. Distortion measurements using a large-volume, Lightweigt, modular MRI phantom with solid signal sources. *Proc Intl Soc Mag Reson Med*. 2022:5032.

231. Cox BL, Ludwig KD, Adamson EB, Eliceiri KW, Fain SB. An open source, 3D printed preclinical MRI phantom for repeated measures of contrast agents and reference standards. *Biomed Phys Eng Express*. 2018;4(2):027005. doi:10.1088/2057-1976/aa9491

232. De Tillieux P, Topfer R, Foias A, et al. A pneumatic phantom for mimicking respiration-induced artifacts in spinal MRI. *Magn Reson Med*. 2018;79(1): 600-605. doi:10.1002/mrm.26679

233. Duan Q, Duyn JH, Gudino N, et al. Characterization of a dielectric phantom for high-field magnetic resonance imaging applications: characterization of a dielectric phantom for high-field MRI. *Med Phys*. 2014;41(10):102303. doi:10.1118/1.4895823

234. Ianniello C, de Zwart JA, Duan Q, et al. Synthesized tissue-equivalent dielectric phantoms using salt and polyvinylpyrrolidone solutions: synthesized tissue-equivalent dielectric properties using a water-soluble polymer. *Magn Reson Med*. 2018;80(1):413-419. doi:10.1002/mrm.27005

235. Dielectric phantom recipe generator. https://amri.ninds.nih.gov/cgi-bin/phantomrecipe

236. Schote D, Behrens J, Winter L, Kolbitsch C, Dinh C. ScanHub: Open-Source Platform for MR scanner control, acquisition and postprocessing. *Proc Int Soc Magn Reson Med*. 2023;2391.