MaRGE: A graphical environment for MaRCoS

José M. Algarín [a], Teresa Guallart-Naval [a], José Borreguero [b], Fernando Galve [a], Joseba Alonso [a],*

[a] *MRILab, Institute for Molecular Imaging and Instrumentation (i3M), Spanish National Research Council (CSIC) and Universitat Politècnica de València (UPV), 46022 Valencia, Spain*
[b] *Tesoro Imaging S.L., 46022 Valencia, Spain*

ARTICLE INFO

ABSTRACT

The open-source console MaRCoS, which stands for "Magnetic Resonance Control System", combines hardware, firmware and software elements for integral control of Magnetic Resonance Imaging (MRI) scanners. Previous developments have focused on making the system robust and reliable, rather than on users, who have been somewhat overlooked. This work describes a Graphical User Interface (GUI) designed for intuitive control of MaRCoS, as well as compatibility with clinical environments. The GUI is based on an arrangement of tabs and a renewed Application Program Interface (API). Compared to the previous versions, the MaRGE package ("MaRCoS Graphical Environment") includes new functionalities such as the possibility to export images to standard DICOM formats, create and manage clinical protocols, or display and process image reconstructions, among other features conceived to simplify the operation of MRI scanners. All prototypes in our facilities are commanded by MaRCoS and operated with the new GUI. Here we report on its performance on an experimental 0.2 T scanner designed for hard-tissue, as well as a 72 mT portable scanner presently installed in the radiology department of a large hospital. The possibility to customize, adapt and streamline processes has substantially improved our workflows and overall experience.

1. Introduction

Until recently, the availability of Magnetic Resonance Imaging (MRI) has been strongly compromised by the initial investment required, as well as maintenance and infrastructural costs. A transformative shift is taking place, however, driven by a collective endeavor to reduce expenses and enhance affordability. Particularly, the advent of low-field MRI technology has significantly contributed to pushing down costs, making MRI more accessible to a broader population, transcending economic barriers and ensuring that the benefits of this medical imaging modality reach diverse communities [1–3].

The electronic control system or 'console' is pivotal to any MRI scanner. This component needs to control an intricate interplay of electromagnetic pulses with exquisite time resolution, as well as manage signal acquisition and the subsequent image reconstruction. Traditionally, MRI scanners have operated using proprietary consoles, but this constrains their adaptability and extensibility [4]. Therefore, as low-field MRI systems gain momentum, the need for an open-source, adaptable, and affordable console continues to grow. Among the numerous open-source initiatives to develop an affordable and widely spread control system [5–13], the recently released *Magnetic Resonance Control System* (MaRCoS) stands out for its high performance, versatility and low cost [14,15].

The core of MaRCoS is a Red Pitaya SDRLab 122–16, a commercial board with two analog inputs for digitizing received data and two analog outputs for generating radiofrequency (RF) transmit waveforms. It can control either a GPA-FHDO [16] or an OCRA1 [17] four-channel gradient board, providing flexibility for different experimental setups. The architecture of MaRCoS allows for unlimited, cycle-accurate sequences, handling rapid bursts of events, creating arbitrary waveforms, including RF and gradients. It supports the independent control of frequency, phase, and amplitude for different TX and RX channels in each repetition. MaRCoS is the first inexpensive system capable of these features.

Understandably, the first developments of MaRCoS have focused on hardware and firmware, advancing towards a robust and reliable technology while keeping the costs down and promoting accessibility. Nevertheless, MaRCoS is a live project developed collaboratively within an open international community and designed for massive use in a wide variety of MRI systems, so the software needs to eventually transition from machine-centered to user-centered. We consider that the next milestone in this regard is the release of a Graphical User Interface (GUI) that facilitates the installation of MaRCoS in MRI systems, regardless of their hardware, and that eases scanner operation both in laboratory and clinical settings. In this vision, it is critical that
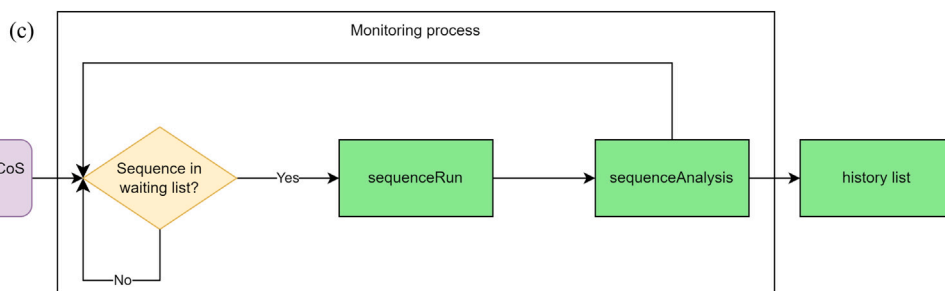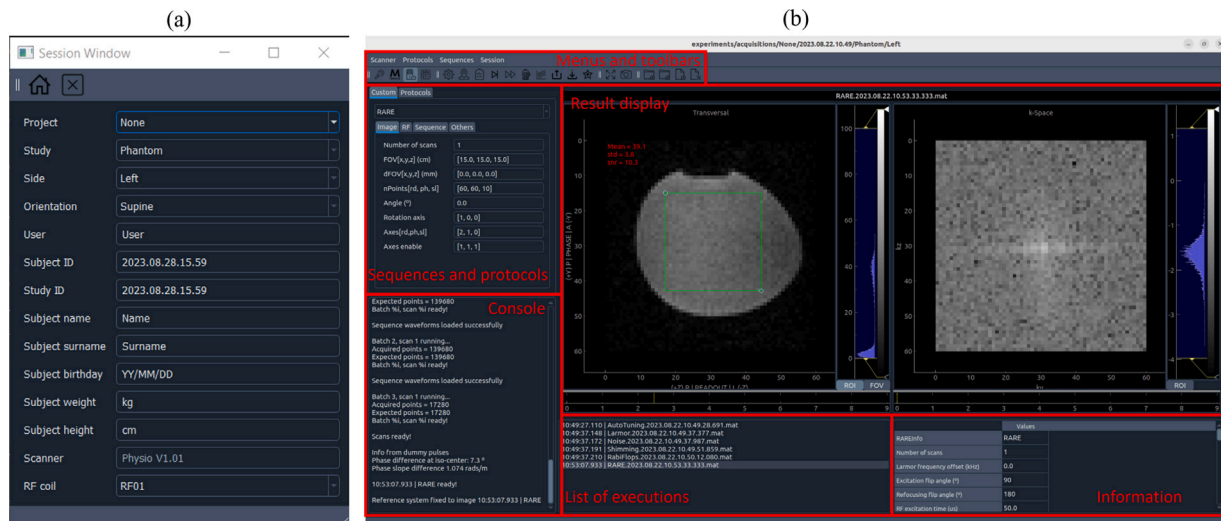

Fig. 1. Screenshot of the session (a) and main (b) windows of MaRGE GUI. (c) Workflow for sequence execution.

the interaction between human operators and the scanner is intuitive, but also that the GUI offers an accessible gateway for users to modify sequence parameters, apply imaging protocols, and visualize results.

In previous publications we introduced a rudimentary GUI designed to offer fundamental functionality for any system controlled by MaRCoS [14,15]. However, this first version had a number of limitations: (i) communication management with the MaRCoS server required users to establish a connection with the server through terminal commands before accessing the GUI; (ii) session management lacked information, leading to potential data mismanagement and disorganization; (iii) the GUI only allowed single sequence runs, as opposed to the multi-sequence protocols which MRI technologists and radiologists are used to; (iv) the absence of an automated calibration protocol for hardware components required users to manually calibrate the system parameters (e.g. Larmor frequency or gradient shimming) and introduce them in the next executed sequences, which was error-prone and severely limited the workflow; (v) the GUI lacked functionalities required for integration in clinical environments, e.g. DICOM compatibility, making it clash with established workflows; (vi) the inability to compare different reconstructions acquired during the same session hindered image analysis and diagnostic evaluations; and (vii) the GUI did not include tools for image post-processing.

In this paper, we present MaRGE (MaRCoS Graphical Environment) [18], a major upgrade with respect to the existing MaRCoS GUI, fully coded in Python 3, and designed to overcome the aforementioned limitations. A stable release of MaRGE can be found in https://github.com/marcos-mri, a GitHub organization that contains all the repositories required to make it work. MaRGE is released under the GNU General Public License version 3.0 (GNU GPL V3.0), emphasizing our commitment to open-source principles and fostering collaboration within the community. The released upgrades are aimed at ensuring the adaptability and usability of the system, both in controlled research environments and in clinical practice. This paper describes technical details of MaRGE and provides insight on its underlying Application Program Interface (API). Finally, we present results obtained from experiments conducted in a laboratory device dedicated to dental imaging [19,20] and a portable unit for extremity imaging [21,22] installed in the Medical Imaging Department of La Fe Hospital in Valencia, Spain.

2. Goals

The main goals behind MaRGE are three: first, to enhance user interaction with the MaRCoS system, fostering the experience for human operators with varying levels of technical expertise; second, to optimize the functionality of MaRCoS specifically for laboratory and medical environments, tailoring its capabilities for diverse research and clinical needs; and third, to provide a set of tools for image post-processing. To meet these objectives, we have tried to design the software placing the focus on users and their experience.

3. Graphical user interface

This section describes the features and functionalities encapsulated within MaRGE, and Fig. 1 shows a screenshot of the new GUI. The user interface builds on the underlying API, allowing seamless communication with the MaRCoS server, sequence initialization, parameter adjustments, and oversight of data acquisition.

3.1. Session window

The session window (Fig. 1(a)) in MaRGE opens up as the GUI is launched and serves as the starting point for the user. Here, they can input general information related to the imaging session, such as the research project or clinical trial, technical aspects about the scanner and protocols, and patient details. Through this interface, operators can
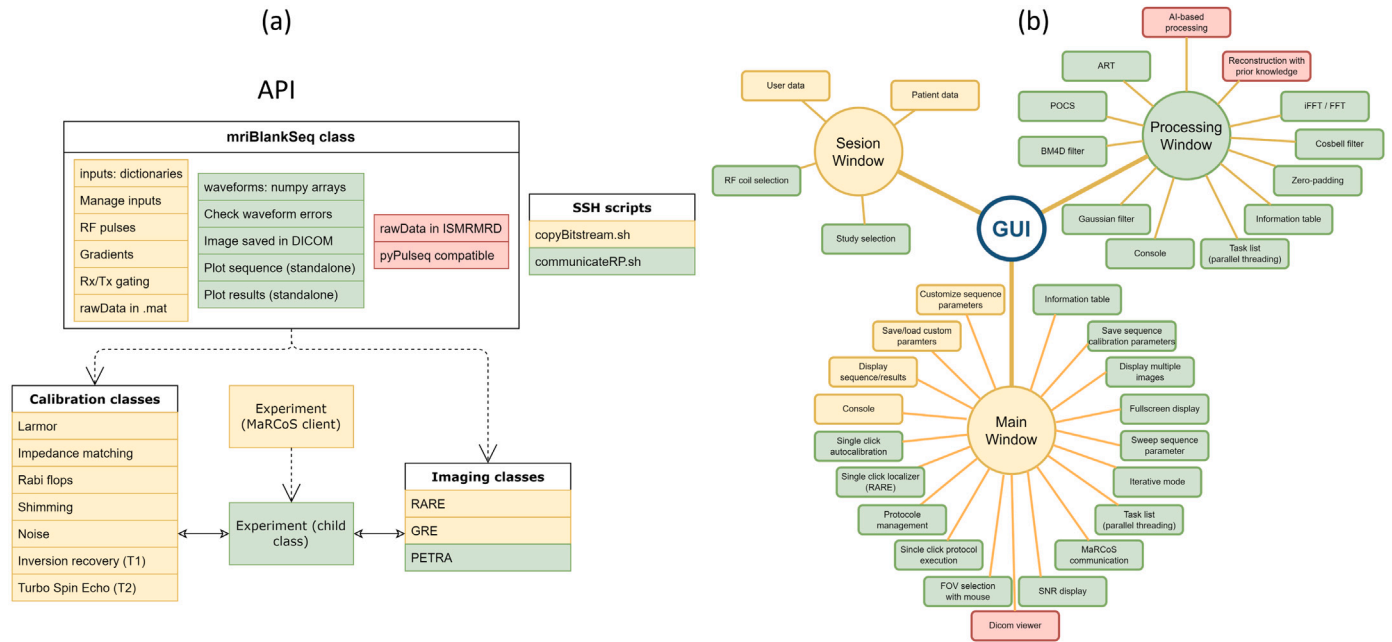
Fig. 2. Diagram of the API (a) and GUI (b). Colors represents what was done in first basic approach (orange), what has been implemented in current version (green) and some possible additional implementations (red). For the API sketch, dashed arrows represents inheritance while solid arrows represents communication. mriBlankSeq class box contains some of its attributes and methods.

ensure that images are linked to accurate metadata, enabling identification, tracking, and organization of imaging data. The session window also contributes to streamlining the workflow, as users can predefine session-specific parameters that are then automatically integrated into sequence setups and image acquisitions. All in all, the integration of user, patient, and project information within the MaRGE GUI is designed to promote efficient data management, facilitate traceability, and support adherence to established medical imaging protocols. Additionally, the information provided in the session window is incorporated into the resulting DICOM files, ensuring that imaging data retains its context and relevance for research or clinical purposes.

3.2. Main window

The main GUI window is shown in Fig. 1(b). Here, users can fine-tune sequence parameters, optimize timings, and personalize pulse patterns for their imaging needs. The GUI also allows protocol design from scratch, enabling researchers and clinicians to experiment with different sequence configurations. A history list provides access to previous reconstructions and results, to compare data and facilitate informed decision-making.

Integral to the MaRGE GUI is a dedicated area designed to display images and results processed by the system. In essence, this area is meant to become a platform for quality assessment, data analysis, and deciding whether a re-scan or a different sequence is required. To this end, users can zoom and crop the field of view, evaluate local and global signal-to-noise ratios automatically and gauge contrast levels.

The MaRGE GUI offers also a real-time console that serves as an informative dashboard. This presents users with relevant feedback, including system status, progress updates, and any potential alerts. Furthermore, an integrated box offers insights into items within the history list. A toolbar complements the GUI, providing dedicated functionalities to manage communication with the MaRCoS server, oversee sequence operations, and initiate new protocol creations.

Upon opening the GUI, a persistent process starts monitoring for sequences that pend execution. The monitoring process remains active as long as the main window is open. Once a sequence is detected in the queue, the monitoring process calls the `sequenceRun` and `sequenceAnalysis` methods (see Section 4.3), and the resulting data

is saved when the sequence terminates. A conceptual representation of this workflow is depicted in Fig. 1(c).

3.3. Post-processing window

The post-processing window (Fig. 5(f)) has a similar appearance to the main window, but it is intended for analyzing and refining the acquired data. Users can access the post-processing window either through the corresponding button located in the toolbar of the main window or by right-clicking on an item in the list of executions, which opens a context menu (see Fig. 1(b)). Here users can apply a range of image processing techniques both in *k*-space and in image space. The available tools are still few and basic, but we expect it to become richer as the number of MaRCoS and MaRGE users increases. For *k*-space we have zero-padding and cosine-bell filters, whereas images can be denoised with Block-Matching 4D (BM4D) [23] or Gaussian filters. The software can be expanded also with *ad hoc* reconstruction algorithms. By default, MaRGE includes standard Fast Fourier Transform (FFT), Partial Fourier by zero padding or projection onto convex sets (POCS) [24], and Algebraic Reconstruction Techniques (ART, [25–28]). All the iterative methods have been programmed to be compatible with GPU execution.

4. Application program interface

Custom sequences or acquisition schemes can be implemented through an API that the MaRGE package provides. The MaRGE API (Fig. 2(a)) has been programmed in Python and encapsulates functionalities intended to simplify the interaction between users and the MaRCoS system, including establishing communication with the MaRCoS server, executing pulse sequences, or managing data storage.

4.1. Setting up communication with MaRCoS server

Upon initialization, the GUI needs to establish communication with the MaRCoS server running in the Linux environment on the SDRLab board [15]. This process requires setting up the network parameters and ensuring the integrity of data transmission. Encrypted communication with the MaRCoS server is performed with two Secure Shell

(SSH) files: `copyBitstream.sh` to setup MaRCoS into the SDRLab, and `communicateRP.sh` opens and closes the bidirectional communication channels with the MaRCoS server. This communication allows users to send sequences from the client PC, controlling the system at a high level, or receiving data arrays back to the client PC once the sequence is finished [15].

4.2. *mriBlankSeq: the master class to create pulse sequences*

At the core of the API lies the `mriBlankSeq` class (Fig. 2(a)), an extensible parent class that serves as the base for generating arbitrary pulse sequences. In essence, this master class provides a structured framework for creating, executing, and managing complex MRI sequences. To that end, `mriBlankSeq` incorporates a number of attributes and methods (see mriBlankSeq box in Fig. 2(a)).

A first group of attributes include dictionaries to store sequence inputs and waveform data. They therefore function as containers for parameters that define the characteristics of an MRI sequence, including timing, pulse amplitudes, gradient settings and others. Additionally, the `mriBlankSeq` class integrates methods for defining RF pulses, gradient waveforms, and acquisition triggers. Leveraging these methods, users can assemble complex sequences by concatenating smaller waveform components.

In the current version of the API, the complete generated waveform is stored in NumPy arrays, which are subsequently transmitted to the MaRCoS client in the control computer. The communication between the sequence generator and the MaRCoS server on the SDRLab makes use of a dedicated method within the master class (`floDict2Exp`), which checks for inconsistencies on the sequence waveforms and orchestrates the transfer of waveform data to the MaRCoS server. These checks guarantee that the external electronics receive the right analog pulses, e.g. by identifying negative time steps or checking pulse amplitudes are within hardware bounds.

In MaRGE, reconstructions are saved in the DICOM 3.0 standard format for medical image storage and interoperability, as well as a MatLab native file (`.mat`) including the raw signal data acquired by the scanner, the reconstructed images, and relevant metadata related to executed sequences. This information facilitates subsequent analyses, comparisons, and validations of sequence performance. The corresponding `mriBlankSeq` method is called `saveRawData`.

`mriBlankSeq` further integrates two methods for data visualization: `sequencePlot` can be invoked to display the sequence instructions (Fig. 3(a)), offering a graphical overview of the designed pulse patterns; whereas `plotResults` provides a visual representation of the resulting data (Fig. 3(b)), which can take the form of one or two-dimensional plots depending on whether the user wants to visualize a signal or an image. These methods are called later by the GUI to visualize data (Figs. 1(b), 5 and 6), but may be run also in standalone execution (i.e. without the GUI), which we find useful during debugging (Fig. 3).

In conclusion, the `mriBlankSeq` class stands as the cornerstone of the API, encompassing user inputs, waveform generation, server communication, data preservation and visualization. Through its attributes and methods, `mriBlankSeq` allows users to generate complex MRI sequences and integrate them within the MaRCoS environment for execution and data management.

4.3. *Pulse sequences: child classes*

At the heart of the sequence hierarchy lies the principle of inheritance. In the MaRGE API, each pulse sequence class is conceived as a child of `mriBlankSeq`, thereby inheriting its attributes and methods. This design approach ensures a consistent and standardized foundation across all sequence classes, facilitating code management and ideally promoting best practices. By adopting this object-oriented inheritance model, MaRGE fosters modular development, allowing developers to
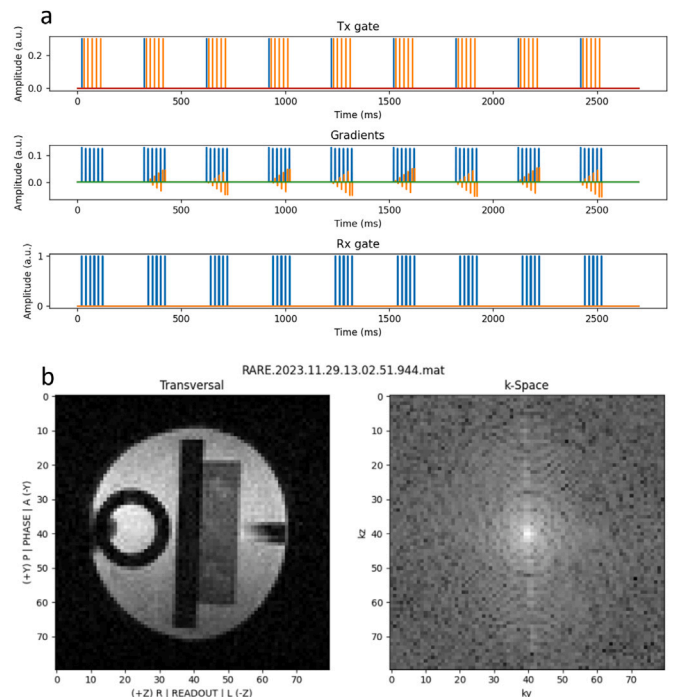

Fig. 3. Screenshot of the results displayed by a sequence executed in standalone. (a) Sequence waveform display after calling the `sequencePlot` method. (b) Image display after calling the `plotResults` method.

leverage the existing functionality provided by `mriBlankSeq` and build upon it to craft different sequences.

Child classes are executed by invoking the `sequenceRun` method. Fig. 4 showcases a simplified representation of the sequence structure for a 1D gradient echo. Once a pulse sequence is executed, the acquired data can be reconstructed into images for analysis and visualization by means of the `sequenceAnalysis` method. Sequences with a high number of acquired points can be executed in batches to prevent buffer overflow. The maximum allowable number of points before initiating a new batch can be configured through the hardware configuration file (Section 4.4).

4.4. *Hardware configuration file*

The hardware specifications of an MRI system determine to a large extent the quality and accuracy of the reconstructed images. The open, modular nature of MaRCoS means that the software needs adjustment whenever new hardware is encountered. To make this scalable, the MaRGE API includes a configuration file that stores parameters related to the scanner hardware components, acting as a centralized repository for information on RF coils (e.g. serial number), RF and gradient amplifiers (gain, input parameter ranges, etc.), or digital components such as the SDRLab boards or the control computer (IP addresses, etc.). This information is used by the API to initialize and interact with the MRI system's hardware, ensuring accurate control and synchronization during MRI scans. The hardware configuration file can be manually written following a simple structure [18].

5. Methods

For this work, MaRGE was installed in two distinct low-field MRI setups: a laboratory-based scanner for *ex vivo* dental imaging [19,20], and a portable extremity scanner [21,22] located in the experimental radiology department of a large hospital facility.

To operate MaRGE for the first time, users are advised to follow the setup procedure detailed in Ref. [18]. The initial step is the installation

```
# Initialize time
t0 = 20
t_ex = 20e3

# Shimming
self.iniSequence(t0, self.shimming)

# Excitation pulse
t0 = t_ex - hw.blkTime - self.rfExTime / 2
self.rfRecPulse(t0, self.rfExTime, rf_ex_amp, 0)

# Dephasing readout
t0 = t_ex + self.rfExTime / 2 - hw.gradDelay
self.gradTrap(t0, grad_rise_time, self.rdDephTime, rd_deph_amplitude * self.rdPreemphasis,
              hw.grad_steps, self.axis, self.shimming)

# Readout gradient
t0 = t_ex + self.echoTime - self.rdGradTime / 2 - grad_rise_time - hw.gradDelay
self.gradTrap(t0, grad_rise_time, self.rdGradTime, rd_grad_amplitude, hw.grad_steps, self.axis,
              self.shimming)

# Rx gate
t0 = t_ex + self.echoTime - self.rxTime / 2
self.rxGateSync(t0, self.rxTime)
rd_points += self.n_rd

self.endSequence(t_ex + self.repetitionTime)
```

Fig. 4. Example of sequence code for one dimensional gradient echo.

of the necessary software dependencies, including MaRCoS and the required Python packages. MaRGE can be then launched, opening the application. In our case, after downloading MaRGE from the repository [18] and installing on the control computers, running Windows for the dental system and Linux for the portable system, the hardware configuration files were adjusted for each system, and essential Python modules (such as PyQt5, pyqtgraph and others) were installed.

The Main Window has a button for running a set of automatic calibration sequences with a single click. This procedure includes the determination of: (i) the Larmor working frequency; (ii) the radio-frequency coil efficiency, which is then used to calibrate the amplitude of resonant pulses in a dimensionless scale between 0 and 1 to achieve a given flip angle for a given pulse duration; (iii) the gradient currents that actively shim the main magnetic field, estimated from the full-width-half-maximum (FWHM) of the spectrum of a spin-echo; and (iv) a noise measurement that is compared with Johnson limit.

In the dental scanner we typically start by manually impedance-matching the radio-frequency coil. However, the radio-frequency chain of the extremity scanner counts with a system to impedance-match the coil autonomously to $50\,\Omega$ at the working frequency [29]. This process can be triggered from MaRGE using a specialized method from the API, but it is also accessible from the main GUI as any other regular pulse sequence. Furthermore, this method is automatically integrated into the standard calibration process if the hardware for autonomous impedance matching is detected by MaRGE.

In the laboratory-based dental scanner [19,20], exclusively dedicated to *ex vivo* imaging, predefined imaging protocols are not a prerequisite. As a sample for these tests we used a cylindrical phantom with copper sulfate diluted in water with a concentration 1%. Here, the imaging process starts with manual impedance matching of the RF coil, followed by an initial calibration and, finally, the execution of customized pulse sequences. Depending on the application, we typically run Gradient Echo (GRE), Rapid Acquisition with Relaxation Enhancement (RARE), and Pointwise Encoding Time Reduction with Radial Acquisition (PETRA), which are all configured using the custom tab in the main GUI.

On the other hand, the portable scanner [21,22], operates in a clinical environment for *in vivo* extremity imaging and thus benefits from the possibility of executing imaging protocols. For instance, we created a specific protocol for knee imaging encompassing two localizers (RARE projection images in transversal and coronal orientation)

and four RARE sequences with different contrasts: coronal inversion recovery, sagittal T_1, sagittal T_2, and transversal T_1. The complete process therefore involves an initial calibration to adapt to the new patient or body part, the execution of the localizers to determine the field of view and angulation, and the subsequent execution of the corresponding imaging protocol.

6. Results

The following subsections evaluate specifically the system performance and the impact of the new developments on research and clinical protocols. These two use cases are arguably representative of the varied environments where low-field MRI is expected to play an increasingly important role.

6.1. Laboratory-based dental MRI scanner

Fig. 5 depicts the calibration process in the laboratory-based dental scanner, showcasing various results required for proper system performance: panel (a) shows a spin echo and its corresponding spectrum, required to measure the Larmor frequency, which, in this case, is measured at 11.1352 MHz; panel (b) shows a noise measurement and its spectrum, yielding an RMS noise value of $319\,\mu V$ at a bandwidth of 50 kHz and revealing distinct electromagnetic interference (EMI) at -24 and 7 kHz relative to the Larmor frequency (these are known to originate at the gradient power amplifier oscillator module); panel (c) displays the curve resulting from the active shimming calibration, along with the amplitude (top) and full width at half maximum (FWHM, bottom) of the spectrum at various shimming values (the optimal configuration achieves a FWHM of 500 Hz, corresponding to an homogeneity of 45 ppm); and panel (d) shows Rabi flops obtained by sweeping the excitation time of a single RF pulse, measured through a Free Induction Decay (FID, top) and an RF echo (bottom), where the maximum intensity ($\pi/2$ pulse) is achieved for $30\,\mu s$ with an input RF amplitude of 0.34 (arbitrary units between 0 and 1). Following these calibrations, the system is considered prepared for imaging.

In Fig. 5(e) we show images reconstructed from the signal detected during the execution of three distinct pulse sequences: RARE (left), GRE (center), and PETRA (right). The post-processed versions of these images are displayed in panel (f). Specifically, we applied a BM4D filter for denoising and rescaled the matrix to double its size.
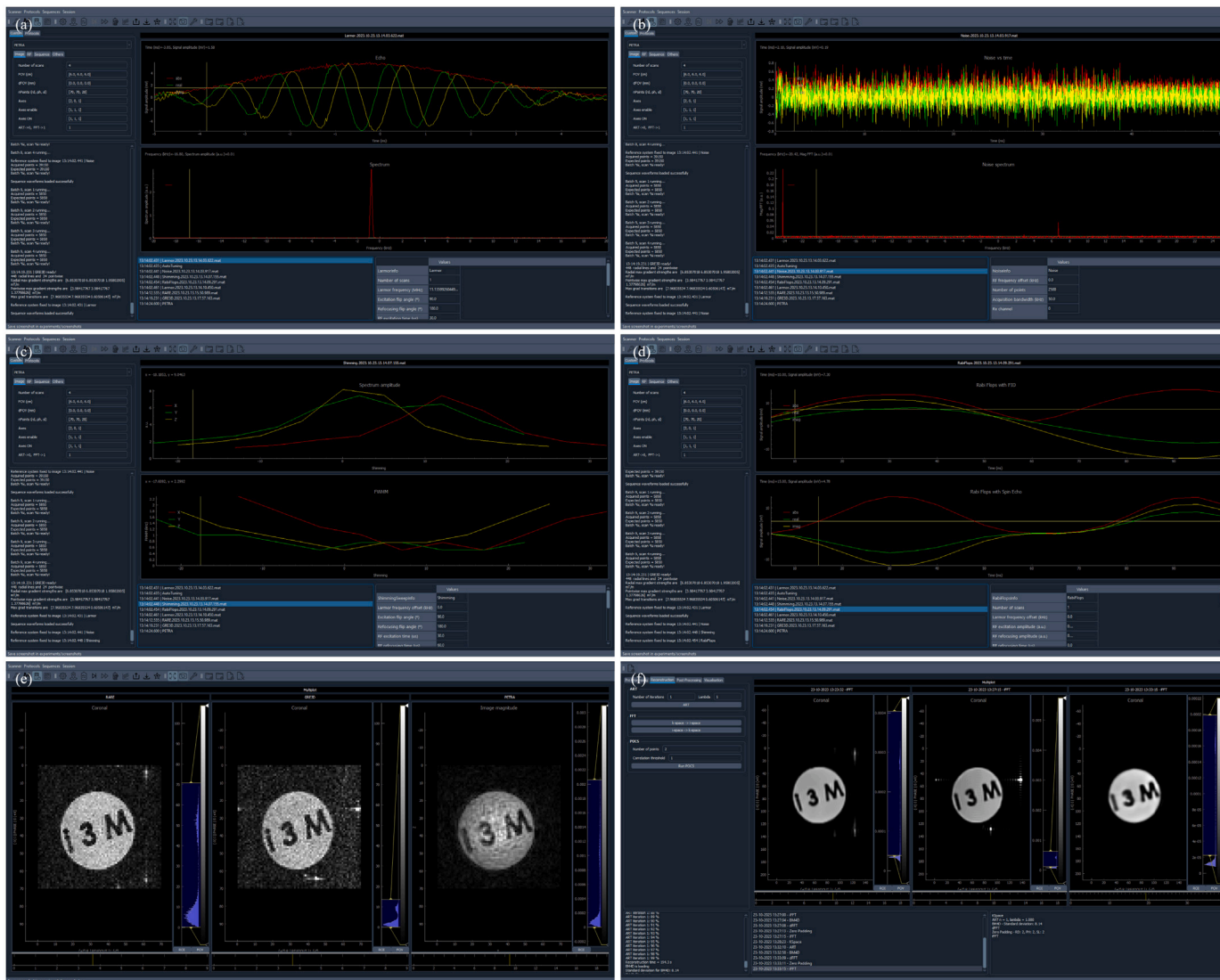
Fig. 5. Screenshots of the GUI in the laboratory-based dental MRI scanner: (a) echo to calibrate Larmor frequency, (b) noise measurement, (c) shimming to optimize the B_0 homogeneity, (d) Rabi flops to calibrate the B_1 efficiency, (e) and (f) from left to right, SE, GRE and PETRA images before and after post-processing.

6.2. Hospital-based MRI scanner for extremity imaging

Fig. 6 provides an overview of the calibration process in the hospital-based extremity MRI scanner: panel (a) shows the spin echo and its corresponding spectrum for the determination of the Larmor frequency at 3.0673 MHz; panel (b) displays the automatic impedance RF coil matching procedure in the Smith chart (left) and the resulting reflection coefficient after the impedance matching (right), reaching -27 dB with a bandwidth of 20 kHz, giving a Q factor $\approx$ 150; panel (c) shows the noise measurement and its spectrum, yielding an RMS noise value of $58\,\mu$V at 50 kHz, twice the expected Johnson noise level; panel (d) shows the shimming calibration curves, with an optimal value of FHWM $\approx$ 0.5 kHz (163 ppm); and panel (e) shows the Rabi flops, providing a maximum signal for an RF excitation time of $35\,\mu$s for an input RF amplitude of 0.30 (arbitrary units between 0 and 1).

Following calibrations, we present the acquired and post-processed images in Fig. 7. From left to right, the four images correspond to coronal inversion recovery, sagittal T_1, sagittal T_2, and transversal T_1. The top images are raw and post-processing in the bottom included only BM4D filtering, except for sagittal T_1, where a cos-bell filter was applied in the k-space readout direction to suppress artifacts observed in the raw image.

7. Discussion

The main qualitative conclusion we extract from operating with MaRGE is that the integration between GUI and API provides users with simplified but sufficient control over vastly different MRI scanners, as demonstrated in the above use cases. Notably, multi-sequence imaging protocols have been a crucial development for the hospital-based system for routine *in vivo* scans, and the possibility to fine-tune imaging parameters has enabled tailored sequences and, ultimately, an improvement in image quality also in our laboratory-based scanner, where we only worked with phantoms. Additionally, the one-click calibration process simplified significantly the optimization of the scanners' performance, reducing time and manual errors that often compromise the diagnostic potential of our images as well as the patient experience. As an illustrative example of the system's intuitive design, a newly arrived student recently learned to operate the scanner proficiently within just two hours, highlighting the ease of use provided by the MaRGE platform.

In summary, the enhancements aimed at improving user interaction with MaRCoS have been implemented and have led to a more intuitive experience. Perhaps most importantly, MaRCoS is now usable within medical environments.
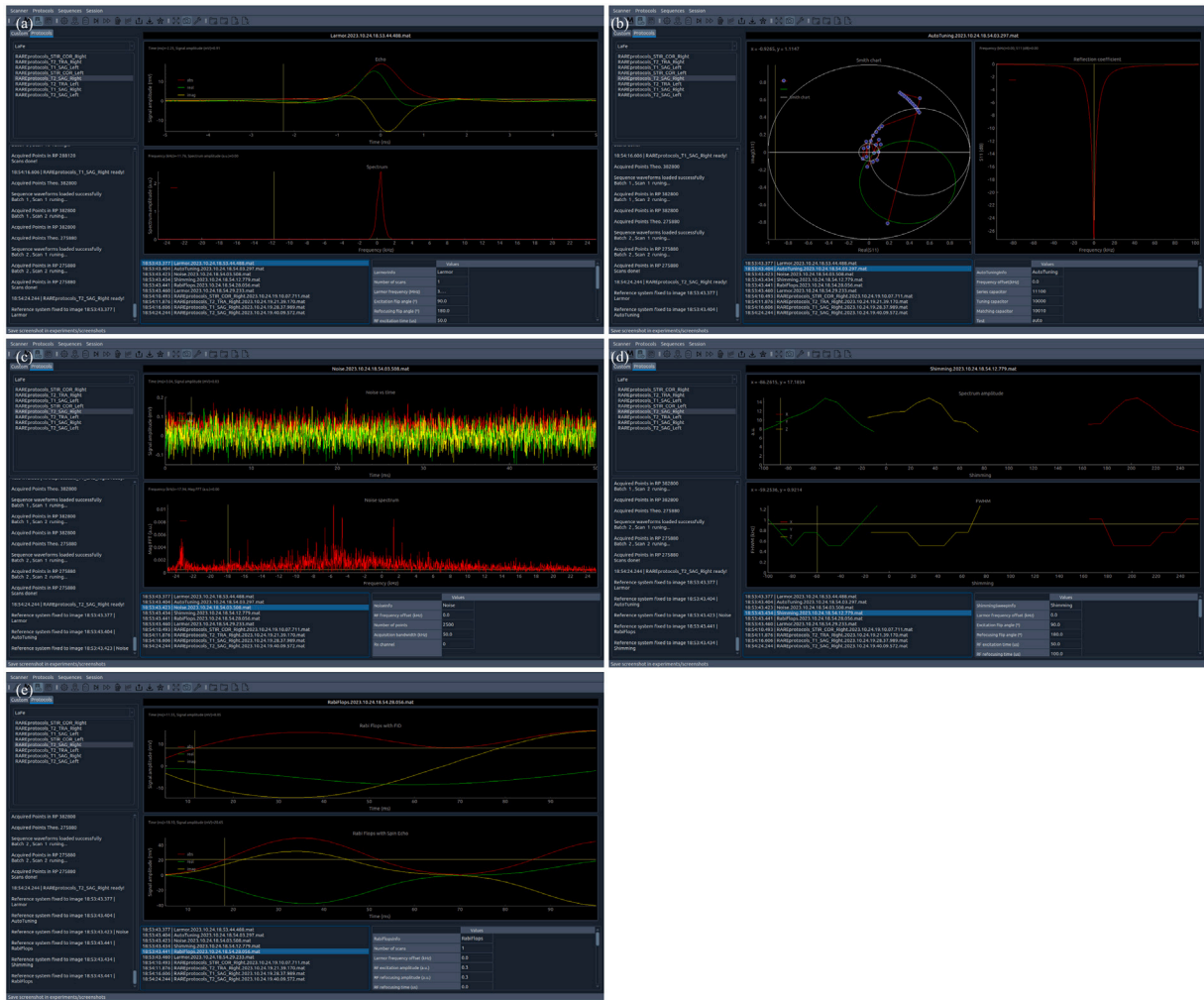
Fig. 6. Screenshots of the GUI in the hospital-based MRI scanner for extremity imaging: (a) echo to calibrate Larmor frequency, (b) automatic impedance tuning and matching; (c) noise measurement, (d) shimming to optimize the B_0 homogeneity, (e) Rabi flops to calibrate the B_1 efficiency.
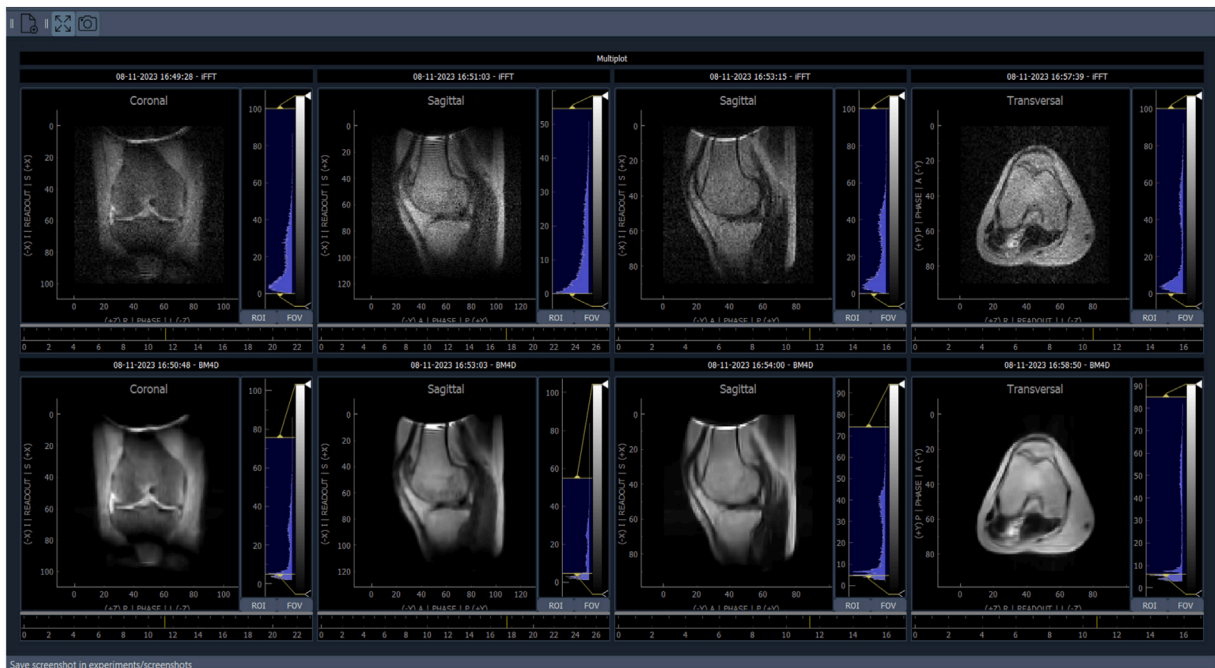

Fig. 7. Acquired images with the executed protocol. From left to right: coronal inversion recovery, sagittal T_1, sagittal T_2, and transversal T_1. Top (bottom) shows raw (post-processed) images.

8. Conclusion

The new MaRCoS GUI has been designed to emerge as a transformative element in the landscape of open-source MRI control, redefining user interaction and amplifying the potential for advanced medical imaging. This paper has explored the multifaceted dimensions of MaRGE, presenting its features and functionalities.

An overarching goal of MaRGE is to contribute to the democratization of MRI control and, in that way, to MRI access. The GUI's interactive components, including the image and results display area, are designed to allow users to validate data in real-time, make informed decisions, and collaborate effectively with colleagues.

Looking ahead, we plan to make MaRGE compatible with Pulseq, a widely embraced open-source framework written in Matlab and under development for python that is extensively used within the MRI community for programming pulse sequences [13]. Compatibility between MaRCoS and Pulseq has been already demonstrated [14], and transferring this to MaRGE should foster early adoption by a larger community. Similarly, we plan to expand MaRGE to save and operate with raw data in the ISMRMRD standard format [30], which was designed to encapsulate complex MRI data comprehensively. By adopting ISMRMRD, we shall enhance interoperability across MRI systems and software platforms.

Finally, official regulatory approval by regional governing bodies will eventually be required for the use of MaRCoS and MaRGE within healthcare systems. This is not our particular priority at the moment, but our developments are indeed influenced by the European Medical Device Regulation (Regulation (EU) 2017/745 of the European Parliament and of the Council) and United States Food & Drug Administration directives for medical devices (510(k), since the device is low to moderate risk and there is a legally marketed predicate), and we are making a conscious effort to adhere to the clinical standards defined therein.

CRediT authorship contribution statement

José M. Algarín: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Teresa Guallart-Naval:** Validation, Software, Formal analysis, Data curation. **José Borreguero:** Validation, Software, Formal analysis, Data curation. **Fernando Galve:** Visualization, Supervision, Project administration, Conceptualization. **Joseba Alonso:** Writing – original draft, Visualization, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

Authors have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

Ethical statement

All participants in this work were adults and provided written informed consent for this study. Ethical approval was obtained from the Ethics Committee (CEIm) of La Fe Hospital in Valencia (CEIm-F-PE-01-16, research agreement number 2022-187-1).

References

[1] M. Sarracanie, N. Salameh, Low-field MRI: How low can we go? A fresh view on an old debate, Front. Phys. 8 (2020) 172.

[2] T.C. Arnold, C.W. Freeman, B. Litt, J.M. Stein, Low-field MRI: Clinical promise and challenges, J. Magn. Reson. Imaging (2022) [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/jmri.28408.

[3] A. Webb, J. Obungoloch, Five steps to make MRI scanners more affordable to the world, Nature 2023 615:7952 615 (7952) (2023) 391–393.

[4] P.P. Stang, S.M. Conolly, J.M. Santos, J.M. Pauly, G.C. Scott, Medusa: A scalable MR console using USB, IEEE Trans. Med. Imaging 31 (2) (2012) 370–379.

[5] W. Tang, W. Wang, W. Liu, Y. Ma, X. Tang, L. Xiao, J.H. Gao, A home-built digital optical MRI console using high-speed serial links, Magn. Reson. Med. 74 (2015) 578–588.

[6] C.J. Hasselwander, Z. Cao, W.A. Grissom, Gr-MRI: A software package for magnetic resonance imaging using software defined radios, J. Magn. Reson. 270 (2016) 47–55, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1090780716300994.

[7] S. Anand, J. Stockmann, L. Wald, T. Witzel, A low-cost (500 USD) FPGA-based console capable of real-time control, in: Proceedings of the 2018 ISMRM and SMRT Annual Meeting and Exhibition in Paris, ISMRM, 2018.

[8] OCRA MRI. [Online]. Available: https://openmri.github.io/ocra/.

[9] K. Takeda, in: G.A. Webb (Ed.), Chapter 7 - Highly Customized NMR Systems Using an Open-Resource, Home-Built Spectrometer, in: Annual Reports on NMR Spectroscopy, Vol. 74, Academic Press, 2011, pp. 355–393, [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080970721000078.

[10] C.A. Michal, A low-cost multi-channel software-defined radio-based NMR spectrometer and ultra-affordable digital pulse programmer, Concepts Magn. Reson. B 48B (2018).

[11] D. Ariando, C. Chen, M. Greer, S. Mandal, An autonomous, highly portable NMR spectrometer based on a low-cost System-on-Chip (SoC), J. Magn. Reson. 299 (2019) 74–92, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1090780718303306.

[12] J. Zhen, C. Dykstra, G. Gouws, S. Obruchkov, R. Dykstra, Mobile low field magnetic resonance hardware development, J. Magn. Reson. 322 (2021) 106852, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1090780720301701.

[13] K.J. Layton, S. Kroboth, F. Jia, S. Littin, H. Yu, J. Leupold, J.F. Nielsen, T. Stöcker, M. Zaitsev, Pulseq: A rapid and hardware-independent pulse sequence prototyping framework, Magn. Reson. Med. 77 (2017) 1544–1552, [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/mrm.26235.

[14] T. Guallart-Naval, T. O'Reilly, J.M. Algarín, R. Pellicer-Guridi, Y. Vives-Gilabert, L. Craven-Brightman, V. Negnevitsky, V. Menküc, F. Galve, J.P. Stockmann, A. Webb, J. Alonso, Benchmarking the performance of a low-cost magnetic resonance control system at multiple sites in the open MaRCoS community, NMR Biomed. 36 (2022) 1–13.

[15] V. Negnevitsky, Y. Vives-Gilabert, J.M. Algarín, L. Craven-Brightman, R. Pellicer-Guridi, T. O'Reilly, J.P. Stockmann, A. Webb, J. Alonso, B. Menküc, MaRCoS, an open-source electronic control system for low-field MRI, J. Magn. Reson. 350 (2023) 107424, [Online]. Available: https://doi.org/10.1016/j.jmr.2023.107424.

[16] GPA-FHDO. [Online]. Available: https://github.com/menkueclab.

[17] OCRA1-SPI controlled 4 channel 18 BIT DAC and RF attenuator. [Online]. Available: https://zeugmatographix.org/ocra/2020/11/27/ocra1-spicontrolled-4-channel-18bit-dac-and-rf-attenutator/.

[18] MaRGE. [Online]. Available: https://github.com/marcos-mri/MaRGE.

[19] J.M. Algarín, E. Díaz-Caballero, J. Borreguero, F. Galve, D. Grau-Ruiz, J.P. Rigla, R. Bosch, J.M. González, E. Pallás, M. Corberán, C. Gramage, S. Aja-Fernández, A. Ríos, J.M. Benlloch, J. Alonso, Simultaneous imaging of hard and soft biological tissues in a low-field dental MRI scanner, Sci. Rep. 10 (1) (2020) 21470.

[20] J. Borreguero, F. Galve, J.M. Algarín, J.M. Benlloch, J. Alonso, Low field slice-selective ZTE imaging of ultra-short T2 tissues based on spin-locking, Sci. Rep. (2023) 1–12, [Online]. Available: https://doi.org/10.1038/s41598-023-28640-x.

[21] T. Guallart-Naval, J.M. Algarín, R. Pellicer-Guridi, F. Galve, Y. Vives-Gilabert, R. Bosch, E. Pallás, J.M. González, J.P. Rigla, P. Martínez, F.J. Lloris, J. Borreguero, Á. Marcos-Perucho, V. Negnevitsky, L. Martí-Bonmatí, A. Ríos, J.M. Benlloch, J. Alonso, Portable magnetic resonance imaging of patients indoors, outdoors and at home, Sci. Rep. 2022 12:1 12 (1) (2022) 1–11, [Online]. Available: https://www.nature.com/articles/s41598-022-17472-w.

[22] J. Algarín, T. Guallart-Naval, E. Gastaldi-Orquín, R. Bosch, F. Lloris, E. Pallás, J. Rigla, P. Martínez, J. Borreguero, R. Alamar, L. Martí-Bonmatí, J. Benlloch, F. Galve, J. Alonso, Portable MRI for major sporting events - a case study on the MotoGP World Championship, 2023, arXiv preprint [Online]. Available: http://arxiv.org/abs/2303.09264.

[23] M. Maggioni, V. Katkovnik, K. Egiazarian, A. Foi, Nonlocal transform-domain filter for volumetric data denoising and reconstruction, IEEE Trans. Image Process. 22 (1) (2013) 119–133.

[24] E.M. Haacke, E.D. Lindskog, W. Linz, A fast, iterative, partial-Fourier technique capable of local phase recovery, 1991, pp. 126–145.

[25] S. Kaczmarz, Angenäherte auflösung von systemen linearer gleichungen, Bull. Int. Acad. Pol. Sci. Let., Cl. Sci. Math. Nat. 35 (1937) 355–357.

[26] R. Gordon, R. Bender, G.T. Herman, Algebraic Reconstruction Techniques (ART) for three-dimensional electron microscopy and X-ray photography, J. Theoret. Biol. 29 (3) (1970) 471–481, [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022519370901098.

[27] R.M. Gower, P. Richtarik, Randomized iterative methods for linear systems, SIAM J. Matrix Anal. Appl. 36 (4) (2015) 1660–1690.

[28] F. Galve, J. Alonso, J.M. Algarín, J.M. Benlloch, Magnetic resonance imaging method with zero echo time and slice selection. ES Patent P202 030 504.

[29] RFAutoMaTE. [Online]. Available: https://github.com/mriLab-i3M/RFAutoMaTE.

[30] S.J. Inati, J.D. Naegele, N.R. Zwart, V. Roopchansingh, M.J. Lizak, D.C. Hansen, C.Y. Liu, D. Atkinson, P. Kellman, S. Kozerke, H. Xue, A.E. Campbell-Washburn, T.S. Sørensen, M.S. Hansen, ISMRM Raw data format: A proposed standard for MRI raw datasets, Magn. Reson. Med. 77 (2017) 411–421.