International Journal on Magnetic Particle Imaging
Vol 9, No 1, Suppl 1, Article ID 2303069, 4 Pages

Infinite Science Publishing

Proceedings Article

MPIMeasurements.jl: An Extensible Julia Framework for Composable Magnetic Particle Imaging Devices

Niklas Hackelberg [a,b,*] · Jonas Schumacher [c] · Justin Ackers [c] · Martin Möddel [a,b] · Fynn Foerger [a,b] · Matthias Graeser [a,b,c,d] · Tobias Knopp [a,b]

[a] Section for Biomedical Imaging, University Medical Center Hamburg-Eppendorf, Hamburg, Germany
[b] Institute for Biomedical Imaging, Hamburg University of Technology, Hamburg, Germany
[c] Fraunhofer Research Institution for Individualized and Cell-Based Medical Engineering IMTE, Lübeck, Germany
[d] Institute for Medical Engineering, University of Lübeck, Lübeck, Germany
[*] Corresponding author, email: n.hackelberg@uke.de

Abstract

Magnetic particle imaging (MPI) is a pre-clinical imaging modality, whose system design is still evolving, in particular towards human studies and clinical use. Therefore, many MPI scanners are custom-made distributed systems, both on the hard- and the software side. In this work we present the open-source Julia framework MPIMeasurements.jl, which implements a composable representation of imaging systems. It also offers flexible data structures that allow the implementation of specific imaging protocols, such as online/offline measurements, repeated measurements and system matrix calibrations. The project is designed to be expanded to new systems through community development and component reuse. To showcase the versatility of the software package, we give an overview of four very different MPI systems, which were realized with MPIMeasurements.jl.

I. Introduction

Devices for magnetic particle imaging (MPI) or magnetic particle spectroscopy (MPS), aside from commercial products, are usually custom-made [1–10]. The complexity and individuality of these systems can be a barrier, both for entering the field of MPI as a researcher and to software integration, maintenance and usage of the systems. On the hardware side, the OS-MPI project [11] introduced an open-source project that aimed to alleviate this barrier by providing hardware designs and supporting software for a field free line (FFL) imaging device. The software side of these scanners is usually as specialized as the hardware itself and features closed-source components with limited options for configuration. MPI scanners are commonly implemented as heterogeneous distributed systems, containing a wide range of devices such as power amplifiers, safety, sensor and control circuits, robots, pumps, motors and components for signal generation and acquisition.

During operation all these different components need to be coordinated for a scanner to perform as intended. This coordination consists of multiple steps with different requirements. Asynchronous steps such as establishing the initial connection to components can happen in any order, while synchronous steps require a specific order, for example to avoid unintended output currents caused by inrush currents. On top of that, the signal handling itself has hard real-time requirements during the actual measurements.

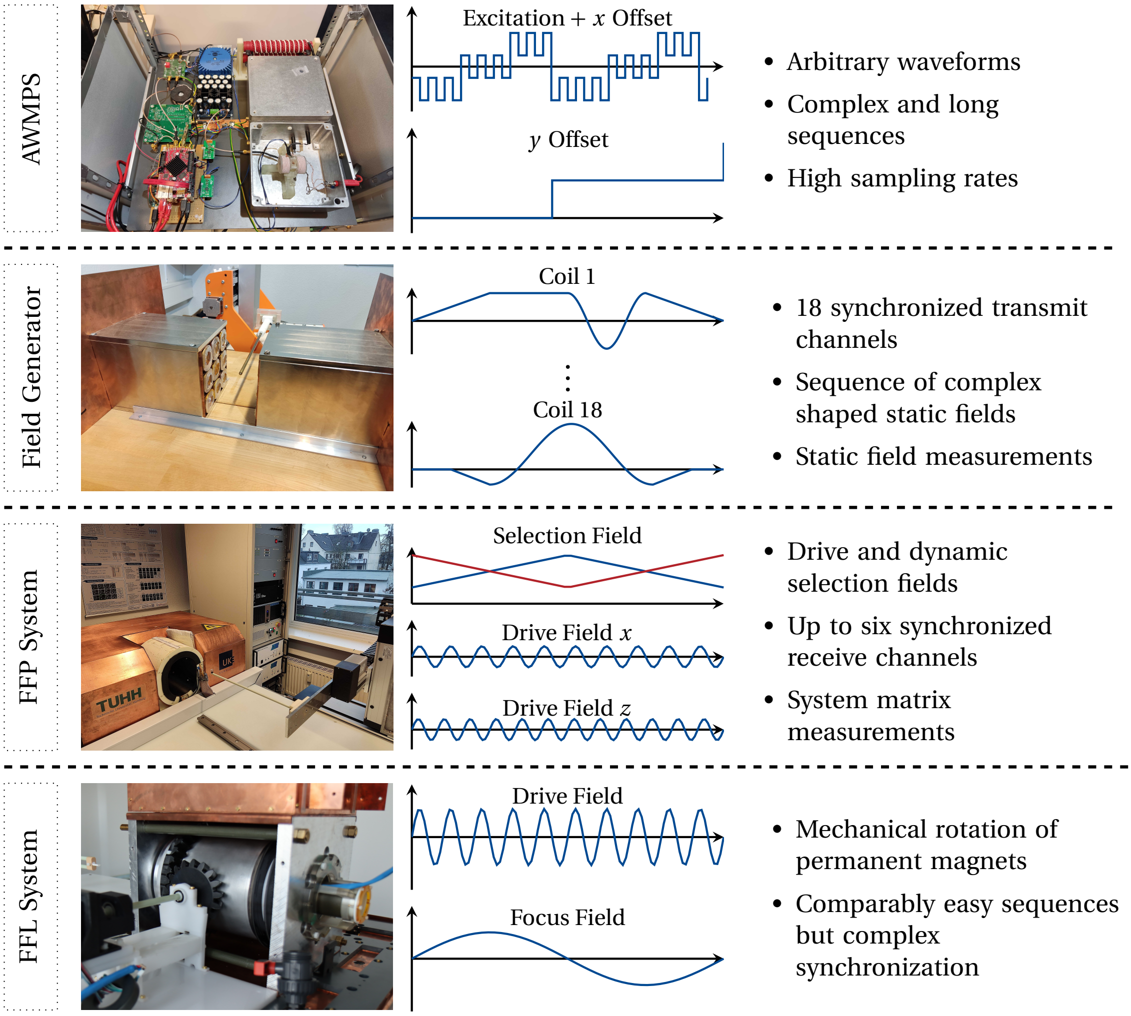

Figure 1: Systems using MPIMeasurements.jl. Each row shows a different system. The second column depicts an exemplary plot of a systems typical `Sequence`, while the third columns lists distinctive properties of a scanner or its operation that are covered by MPIMeasurements.jl.

In this work, we introduce the open-source Julia package MPIMeasurements.jl[1], which provides a framework for representing and working with MPI and MPS systems as composable data structures. It is intended as a flexible and community-developed project, that aims to cover the control of a wide range of system. The package features an extensible type hierarchy with generic interfaces for the devices of a system, as well as the infrastructure to plug-together and initialize a scanner as a collection of these devices. It also includes data structures for implementing custom scanner-agnostic imaging protocols.

[1] https://github.com/MagneticParticleImaging/MPIMeasurements.jl

II. Methods and Materials

The main data structures of MPIMeasurements.jl are the `Scanner`, its `Devices`, `Sequences` and `Protocols`, which can all be instantiated and automatically configured from text-based configuration files.

Devices: Each `Scanner` is composed of multiple `Devices` arranged in dependency trees. A `Device` is a configurable hard- or software component of a scanner, such as a robot. The `Device` types form a hierarchy with generic interfaces and implementations. In addition to the infrastructure for the hierarchy, MPIMeasurements.jl also contains a number of concrete `Device` implementations. In particular, it contains an implementation for a data acquisition (DAQ) system based on the RedPitayaDAQServer project [12], which are `Devices` responsible for signal handling.

Sequences: The implementation of a DAQ `Device` is tightly coupled to the `Sequence` data structure, because the latter is an abstract description of the magnetic fields being applied during an MPI experiment and the data and the acquisition parameters. The parameters are aligned with the MDF specification [13]. The magnetic fields are generated by electrical and/or mechanical channels, that can for instance be described by a sine wave, an arbitrary waveform or the rotation of a permanent magnet.

Protocols: The purpose of `Protocols` is to describe and implement complex measurement procedures that may involve several `Devices` and `Sequences`. For instance `Protocols` are used to realized repeated measurements or robot-based system-matrix calibration measurements. To achieve this, a `Protocol` controls all `Devices` of a scanner to perform all asynchronous and synchronous parts of its respective measurement. A `Protocol` acts as a running process and offers a communication channel, which can be used for user interaction, querying and storing of `Protocol` state and influencing the control flow through manual user interaction, scripts, console modes or even graphical user interfaces.

III. Results

The systems shown in Figure 1 are realized with a software stack composed of MPIMeasurements.jl, RedPitayaDAQServer. and MPIFiles.jl [14]. The selected systems showcase a variety of characteristics, devices and measurement scenarios, all of which could be handled by MPIMeasurements.jl. There is an additional example in the package documentation that shows a complete configuration of an MPS. The systems shown here are, in order, an arbitrary waveform magnetic particle spectrometer (AWMPS) [6], a scaled-down prototype of a power-optimized field generator [7] and two 3D MPI imaging systems [8, 9] based on a field free point (FFP) and an FFL, respectively.

The combined signal handling for these systems covers a wide range of requirements, met by the flexibility of the `Sequences` and the performance of the DAQ `Device`. Next to simple sinusoidal waveforms used for the drive fields of the imaging systems, the AWMPS, the field generator and the FFP systems also feature arbitrary waveforms in different frequency ranges, from 4 Hz up to 25 kHz. The sampling rate and sequence length of the systems differ in a similar vein, with the extreme case of the AWMPS measuring for up to 20 seconds at high sampling rates of 15.625 MHz, which requires high transmission rates. The number of send and receive channels also varies per system, with both the AWMPS and FFL system having two of each. The field generator has 18 send and no receive channels, as the prototype is primarily targeting force applications. And lastly, the FFP system has four send channels and up to six receive channels, the latter varying per measurement. In all systems, the signals are fully synchronized. In the case of the FFL system, the signals are additionally synchronized with the mechanical rotation of permanent magnets.

Due to the generic interfaces of the `Devices`, `Protocols` can be reused across systems, even if concrete `Devices`, such as the robots or the sensors of the systems, come from different vendors. This is true not only for basic measurement `Protocols` that can perform back- and foreground measurements with optional robot movements, but also for `Protocols` that were originally written for specific operations with a specific system. For example, the FFP system uses system matrices for image reconstruction and the field generator requires field measurements for calibration, both of which are measured with specific multi-threaded `Protocols`. These `Protocol` move either a delta sample or a field sensor to different positions with a robot. They have been optimized to reduce the runtime by processing the measurement results of a previous position, while simultaneously moving the robot to the next position and preparing the DAQ `Device` for the next measurement. Both `Protocols` work on different systems just by swapping out the `Scanner` and `Sequence` configuration files.

IV. Conclusion and Discussion

In this work we presented the Julia package MPIMeasurements.jl, which is intended as a flexible, extensible and maintainable community project for MPI and MPS measurement software. So far the project contains the framework itself and concrete `Device`, `Sequences` and `Protocol` implementations for the scanners we built in our research groups. In the future we plan to migrate old and realize new systems with MPIMeasurements.jl and extend the package with new `Devices` and `Protocols`.

The generic type hierarchy of the `Devices` and `Protocols` allows for code reuse across both imaging systems and working groups. That, together with the embedding of `Protocols` and Julia's ease of development, allowed for shorter development time.

Author's statement

The authors thankfully acknowledge the financial support by the German Research Foundation (DFG, grant number KN 1108/7-1 and GR 5287/2-1). The Fraunhofer IMTE is supported by the EU (EFRE) and the State Schleswig-Holstein, Germany (Project: IMTE – Grant: 124 20002/LPW-E1.1.1/1536). Conflict of interest: Authors state no conflict of interest.

References

[1] Z. W. Tay, P. W. Goodwill, D. W. Hensley, L. A. Taylor, B. Zheng, and S. M. Conolly. A high-throughput, arbitrary-waveform, mpi spectrometer and relaxometer for comprehensive magnetic particle optimization and characterization. *Scientific reports*, 6(1):1–12, 2016.

[2] P. Vogel, M. A. Rückert, P. Klauer, W. H. Kullmann, P. M. Jakob, and V. C. Behr. Traveling wave magnetic particle imaging. *IEEE Transactions on Medical Imaging*, 33(2):400–407, 2014, doi:10.1109/TMI.2013.2285472.

[3] P. Vogel, S. Lother, M. A. Rückert, W. H. Kullmann, P. M. Jakob, F. Fidler, and V. C. Behr. MRI Meets MPI: A Bimodal MPI-MRI Tomograph. *IEEE Transactions on Medical Imaging*, 33(10):1954–1959, 2014, doi:10.1109/TMI.2014.2327515.

[4] J. J. Konkle, P. W. Goodwill, O. M. Carrasco-Zevallos, and S. M. Conolly. Projection reconstruction magnetic particle imaging. *IEEE transactions on medical imaging*, 32(2):338–347, 2012.

[5] J. J. Konkle, P. W. Goodwill, D. W. Hensley, R. D. Orendorff, M. Lustig, and S. M. Conolly. A convex formulation for magnetic particle imaging x-space reconstruction. *PloS one*, 10(10):e0140137, 2015.

[6] F. Mohn, T. Knopp, M. Boberg, F. Thieben, P. Szwargulski, and M. Graeser. System Matrix Based Reconstruction for Pulsed Sequences in Magnetic Particle Imaging. *IEEE Transactions on Medical Imaging*, 41(7):1862–1873, 2022.

[7] F. Foerger, N. Hackelberg, M. Boberg, J.-P. Scheel, F. Thieben, L. Mirzojan, F. Mohn, M. Möddel, M. Graeser, and T. Knopp. Flexible Selection Field Generation using Iron Core Coil Arrays. *International Journal on Magnetic Particle Imaging*, 9(1 Suppl 1), 2023.

[8] F. Thieben, F. Mohn, F. Foerger, N. Hackelberg, M. Graeser, and T. Knopp. Safe and Rapid 3D imaging: Upgrade to human-sized brain Magnetic Particle Imaging. *International Journal on Magnetic Particle Imaging*, 9(1 Suppl 1), 2023.

[9] E. Aderhold, J. Schumacher, P. Stagge, M. Ahlborg, T. M. Buzug, and M. Graeser. Towards a Fully Integrated Preclinical Field-Free Line MPI Scanner. *International Journal on Magnetic Particle Imaging*, 9(1 Suppl 1), 2023.

[10] E. Mattingly, E. Mason, K. Herb, M. Śliwiak, J. Drago, M. Graeser, and L. Wald. A sensitive, stable, continuously rotating ffl mpi system for functional imaging of the rat brain. *International Journal on Magnetic Particle Imaging*, 8(2), 2022.

[11] E. Mattingly, E. Mason, K. Herb, M. Śliwiak, K. Brandt, C. Cooley, and L. Wald. OS-MPI: An open-source magnetic particle imaging project. *International Journal on Magnetic Particle Imaging*, 6(2 Suppl 1), 2020.

[12] N. Hackelberg, J. Schumacher, M. Graeser, and T. Knopp. A Flexible High-Performance Signal Generation and Digitization Plattform based on Low-Cost Hardware. *International Journal on Magnetic Particle Imaging*, 8(1 Suppl 1), 2022.

[13] T. Knopp, T. Viereck, G. Bringout, M. Ahlborg, A. von Gladiss, C. Kaethner, A. Neumann, P. Vogel, J. Rahmer, and M. Möddel. MDF: Magnetic particle imaging data format. *arXiv preprint arXiv:1602.06072*, 2016.

[14] T. Knopp, M. Möddel, F. Griese, F. Werner, P. Szwargulski, N. Gdaniec, and M. Boberg. MPIFiles.jl: A Julia Package for Magnetic Particle Imaging Files. *Journal of Open Source Software*, 4(38):1331, 2019.

10.18416/ijmpi.2023.2303069