




A MATLAB-based Instrument Control (MIC) package for fluorescence imaging

Sajjad A. Khan ^{1,2}, Sandeep Pallikkuth ¹, David J. Schodt ¹, Marjolein B. M. Meddens ¹, Hanieh Mazloom-Farsibaf ¹, Michael J. Wester ^{1,3}, Sheng Liu ¹, Ellyse Taylor ¹, Mohamadreza Fazel ¹, Farzin Farzam ¹, and Keith A. Lidke ¹✉

¹ Department of Physics and Astronomy, University of New Mexico, Albuquerque, New Mexico, USA ² Nanoscience and Microsystems Engineering, University of New Mexico, Albuquerque, New Mexico, USA ³ Department of Mathematics and Statistics, University of New Mexico, Albuquerque, New Mexico, USA ✉ Corresponding author

DOI: [10.21105/joss.07275](https://doi.org/10.21105/joss.07275)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Adam Tyson](#)  

Reviewers:

- [@bencardoen](#)
- [@raacampbell](#)

Submitted: 30 July 2024

Published: 28 January 2025

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

MATLAB Instrument Control (MIC) is a software package designed to facilitate data collection for custom-built microscopes. Utilizing object-oriented programming, MIC provides a class for each low-level instrument. These classes inherit from a common MIC abstract class, ensuring a uniform interface across different instruments. Key components such as lasers, stages, power meter and cameras are grouped under abstract subclasses, which standardize interfaces and simplify the development of control classes for new instruments. Both simple and complex systems can be built from these lower level tools. Since the interoperation is developed by the end user, the modes or sequence of operations can be flexibly designed with interactive or automated data collection and integrated analysis. MATLAB provides the ability to create GUIs and therefore MIC allows for both rapid prototyping and for building custom, high-level user interfaces that can be used for production instruments.

Statement of need

Development of new microscopy methods often require integration of novel components and/or novel workflows between low-level hardware. Creating such software solutions for each new instrument or for prototyping can significantly slow down the development cycle. It is also often desirable to be able to quickly analyze incoming data during development. To address this critical need, we have developed the MATLAB Instrument Control (MIC) software package, freely available and specifically tailored for the customization and automation of complex, multi-component microscopy systems (Pallikkuth et al., 2018). MIC leverages MATLAB's (MathWorks Inc.) robust environment for object-oriented programming, allowing users to control diverse instrumentation through a unified interface. Each instrument class within MIC inherits from a common abstract class, ensuring consistency while enabling flexibility to accommodate a variety of components like lasers, stages, and cameras. The source code for MIC has been archived to GitHub: <https://github.com/LidkeLab/matlab-instrument-control>.

MIC not only supports the development and control of new instruments but also integrates seamlessly with MATLAB's comprehensive data and image analysis tools, as well as with Single Molecule Imaging Toolbox Extraordinaire (SMITE) (Schodt, Wester, et al., 2023) allowing researchers to process data in real-time during experiments. This capability is crucial for iterative testing and development in experimental setups. For example, the single molecule localization microscopy/single particle tracking (SMLM/SPT) analysis software suite SMITE

was developed in sync with MIC, so acquiring data and performing analyses flow smoothly. Additionally, MIC is designed to be user-friendly for those familiar with MATLAB, offering customizable control classes, extensive export methods, functional tests, and graphical user interfaces for each instrument component.

An example of MIC's utility is demonstrated through a custom-built Sequential microscope available on [Sequential SR Microscope](#) (Schodt, Farzam, et al., 2023) specifically designed for direct STochastic Optical Reconstruction Microscopy (dSTORM) and DNA Point Accumulation for Imaging in Nanoscale Topography (DNA-PAINT) based super-resolution. Another usage of MIC is demonstrated through a class designed for Total Internal Reflection Fluorescence (TIRF) based super-resolution imaging [TIRF SR Microscope](#) (Fazel et al., 2022). Each of these classes includes an intuitive graphical user interface that manages multiple excitation lasers and camera settings, simplifying complex data collection tasks. Several MIC GUIs are presented in Figure-1.

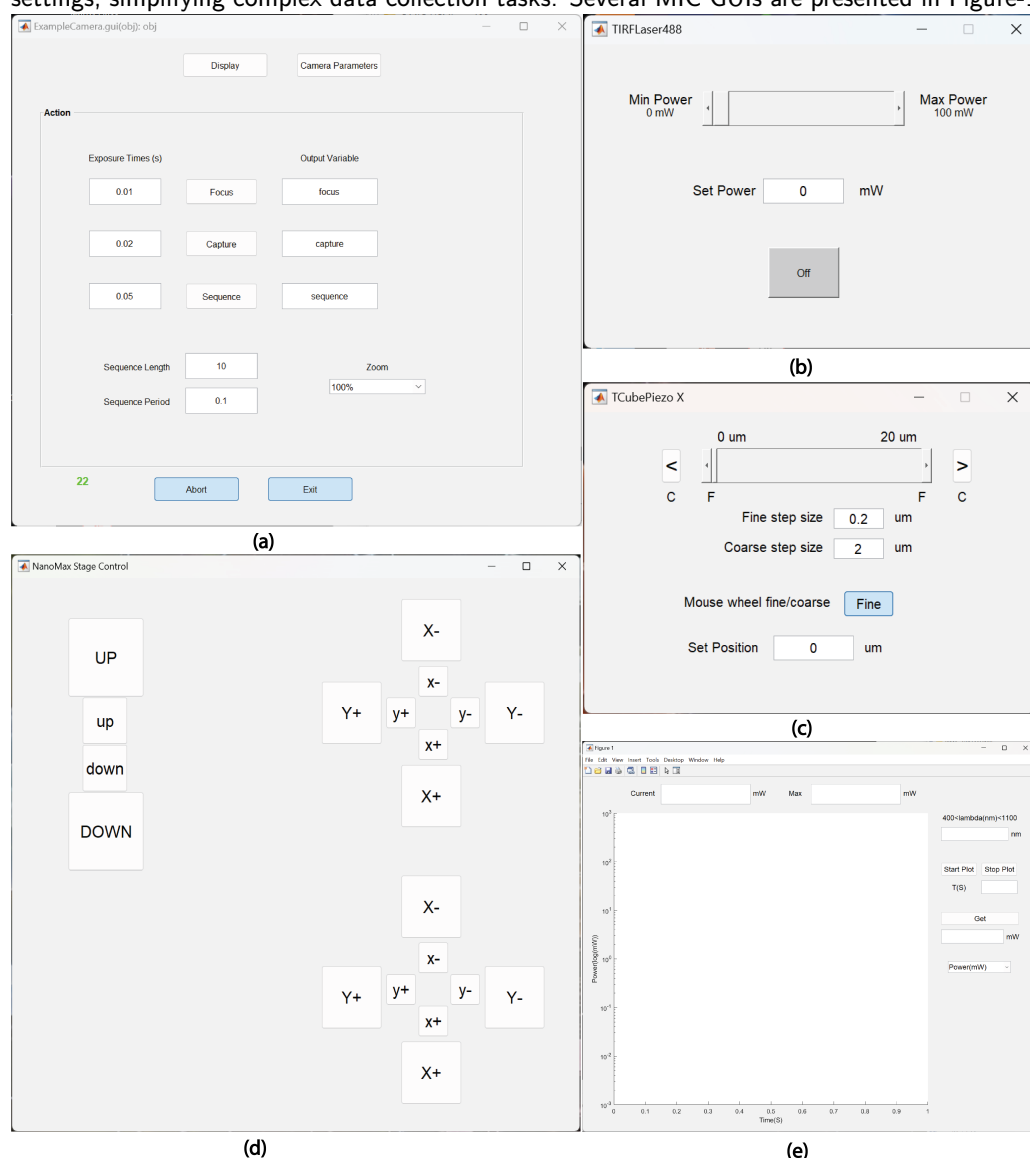


Figure-1: MIC Graphical User Interfaces (GUIs); (a) Camera control GUI (b) Light source (TIRF Laser488) GUI (c) Linear stage (TCubePiezo X) GUI (d) 3D stage (NanoMax Stage Control) GUI (e) Power meter GUI.

MIC is designed to operate with Hierarchical Data Format (HDF5) files which efficiently store very large datasets. The HDF5 format is particularly useful for storing large datasets, as it

allows for efficient data storage and retrieval. This is especially important for single molecule fluorescence imaging, where large amounts of data are generated during experiments. The HDF5 format is also supported by MATLAB, which makes it easy to import and analyze data stored in this format.

There are a few other software packages that allow users to control and synchronize multiple hardware components for microscopy applications. Notable examples include Micro-Manager (Edelstein et al., 2010), based in Java, PYME (the PYthon Microscopy Environment) (PYME Developer Community, 2020), based in the Python environment, and LSMAQ which is a lightweight and flexible laser scanning microscope acquisition software written in MATLAB. It supports National Instruments hardware for galvo-based scanning. Potential users of MIC are encouraged to compare and contrast MIC with these packages to assess what might be best for their particular development environment. Micro-Manager is a customizable platform for controlling microscopy systems, supporting a wide range of hardware devices, and is primarily built on Java. Micro-Manager comes with its own GUI. Micro-Manager can save files in three formats: separate image files, Image file stack (OME-Tiffs) and NDTiff. Micro-Manager may be a good choice for those who primarily use ImageJ/Fiji for image analysis. PYME is designed to facilitate image acquisition and data analysis in microscopy, with a focus on super-resolution techniques like PhotoActivated Localization Microscopy (PALM), STORM, and PAINT. It runs on multiple platforms, including Windows, Linux, and OSX. PYME comprises several key components: PYMEAcquire for microscope and camera control, PYMEVisualize for visualizing localization data sets, and PYMEImage for viewing and processing raster images. PYME is compatible with a variety of data formats, including its proprietary .pzf format as well as standard formats such as .tif. Additionally, PYME supports metadata in multiple formats, including .json, .md, and .xml.

Author Contributions

KAL conceived and supervised the development of MIC. SAK wrote the manuscript. SAK & MJW added GitHub documentation and converted MIC to a hierarchical organization. SAK added the example classes with their GUIs, and added functional tests. KAL added AndorCamera and KCubePiezo. SP added AndorCameraZyla and various laser classes. SAK tested the laser classes. SL added DCAM4Camera to control Hamamatsu cameras using the DCAM4 API. MBMM added DynamixelServo, GalvoAnalog, and HamamatsuLCOS classes. DJS added 3D registration, BiochemValve, CavroSyringePump, H5, and MCLMicroDrive. MF added Attenuator and StepperMotor. HMF added stage3D.abstract, CoherentLaser561, GalvoDigital, IRSyringePump, and MCLNanoDrive. ET added DMP40 and updated Triggerscope. FF added FlipMountTTL, ShutterTTL. SP & SAK added NanoMax. All authors reviewed the manuscript.

Acknowledgements

This work was supported by NIH grants 1R01GM140284, K12GM088021 (ASERT-IRACDA program), NCI P30CA118100, NCI R01CA248166, NIGMS R01GM109888, NIGMS R21GM104691, NIGMS R21GM132716, NIGMS R35GM126934, NCRR RR024438, NIGMS 1R01GM140284, NIBIB 1R21EB019589, NIGMS 5P50GM085273 (New Mexico Spatiotemporal Modeling Center); NSF grants 0954836 and 2039517; DOE grant DE-SC0019267; support from the University of New Mexico Office of the Vice President for Research Program for Enhancing Research Capacity; and supported by grants from NVIDIA while utilizing an NVIDIA A6000 GPU.

References

- Edelstein, A., Amodaj, N., Hoover, K., Vale, R., & Stuurman, N. (2010). Computer control of microscopes using μ Manager. *Current Protocols in Molecular Biology*, 92(1), 14.20.1–14.20.17. <https://doi.org/10.1002/0471142727.mb1420s92>
- Fazel, M., Wester, M. J., Schodt, D. J., Cruz, S. R., Strauss, S., Schueder, F., Schlichthaerle, T., Gillette, J. M., Lidke, D. S., Rieger, B., Jungmann, R., & Lidke, K. A. (2022). High-precision estimation of emitter positions using bayesian grouping of localizations. *Nature Communications*, 13(1), 7152. <https://doi.org/10.1038/s41467-022-34894-2>
- Pallikkuth, S., Meddens, M., Fazel, M., Farsibaf, H., Farzam, F., Wester, M., & Lidke, K. (2018). A MATLAB-based Instrument Control Package for Fluorescence Imaging. *Biophysical Journal*, 114(3, Supplement 1), 532a. <https://doi.org/10.1016/j.bpj.2017.11.2912>
- PYME Developer Community. (2020). *The PYthon Microscopy Environment*. <https://www.python-microscopy.org/>. <https://doi.org/10.5281/zenodo.4289803>
- Schodt, D. J., Farzam, F., Liu, S., & Lidke, K. A. (2023). Automated multi-target super-resolution microscopy with trust regions. *Biomed. Opt. Express*, 14(1), 429–440. <https://doi.org/10.1364/BOE.477501>
- Schodt, D. J., Wester, M. J., Fazel, M., Khan, S., Mazloom-Farsibaf, H., Pallikkuth, S., Meddens, M. B. M., Farzam, F., Burns, E. A., Kanagy, W. K., Rinaldi, D. A., Jhamba, E., Liu, S., Relich, P. K., Olah, M. J., Steinberg, S. L., & Lidke, K. A. (2023). SMITE: Single Molecule Imaging Toolbox Extraordinaire (MATLAB). *Journal of Open Source Software*, 8(90), 5563. <https://doi.org/10.21105/joss.05563>