

ChiMS: Open-source instrument control software platform on LabVIEW for imaging/depth profiling mass spectrometers

Yang Cui (崔扬) and Luke Hanley^{a)}

Department of Chemistry, University of Illinois at Chicago, Chicago, Illinois 60607, USA

(Received 4 March 2015; accepted 2 June 2015; published online 24 June 2015)

ChiMS is an open-source data acquisition and control software program written within LabVIEW for high speed imaging and depth profiling mass spectrometers. ChiMS can also transfer large datasets from a digitizer to computer memory at high repetition rate, save data to hard disk at high throughput, and perform high speed data processing. The data acquisition mode generally simulates a digital oscilloscope, but with peripheral devices integrated for control as well as advanced data sorting and processing capabilities. Customized user-designed experiments can be easily written based on several included templates. ChiMS is additionally well suited to non-laser based mass spectrometers imaging and various other experiments in laser physics, physical chemistry, and surface science. © 2015 AIP Publishing LLC. [<http://dx.doi.org/10.1063/1.4922913>]

I. INTRODUCTION

The data acquisition and control software that is provided with commercial mass spectrometers generally performs well at the specific modes of operation for which such instruments have been designed. Software also plays an important role in the customization of commercial mass spectrometers as well as in the implementation of entirely “homebuilt” instruments, both of which can be used to prototype novel experimental strategies. However, commercial software is usually proprietary and often cannot be readily modified for alternate modes of operation without the express permission and cooperation of the instrument vendor. Cost, frequency of upgrades, and integration with evolving operating systems are among the additional issues that can arise with commercial data acquisition software.

Free open-source data acquisition software can lower the cost of owning, maintaining, and operating custom-built scientific instruments. Furthermore, community support of open-source software can improve interoperability and flexibility.¹ Open source software has been developed for the analysis of mass spectrometry (MS) datasets,^{2–4} including those produced in imaging experiments.^{5–7} Publically hosted, open-source instrument control software can permit similar advantages, especially given that so many MS imaging configurations are actually ion source modifications on commercial MS analyzers.

LabVIEW software (National Instruments, Austin, TX) has been widely adopted for data acquisition and control of customized scientific instrumentation,^{8,9} including mass spectrometers,^{10–13} and many data acquisition programs written in LabVIEW are open source. Some users consider LabVIEW to be interpreted and therefore limited by data transfer rates. However, LabVIEW is actually a compiled programming language that resides on top of the low-level virtual machine (LLVM) compiler which does allow high data transfer rates.¹⁴

This paper describes a versatile open-source instrument control software platform named ChiMS that was written within LabVIEW for imaging and depth profiling mass spectrometers. ChiMS was initially designed for a laser-based time-of-flight mass spectrometer optimized for imaging and depth profiling^{15–18} and some aspects of the software performance have been described previously.^{15,16} ChiMS was also designed for transferring large datasets from a digitizer to computer memory at high repetition rate; saving data to hard disk at high throughput; and automating data processing, imaging and depth profiling experiments. However, the ChiMS software has now evolved so that it is additionally well suited to non-laser based MS imaging and various other experiments in laser physics, physical chemistry, and surface science. The data acquisition mode generally simulates a digital oscilloscope, but with peripheral devices integrated for control as well as advanced data sorting and processing capabilities. Finally, customized user-designed experiments can be easily written based on several templates included in the ChiMS software.

II. SOFTWARE DESIGN AND INSTRUMENTATION

A. Software design

The structure of the ChiMS software is shown in Figure 1: it is a modular multiple-loop application framework with some parts involving a dynamic framework within a subprogram.¹⁹ ChiMS is an extension of the compound design pattern concept¹⁹ as it is constructed in a multiple single loop pattern composed of modules that communicate by messaging between loops. The hierarchy of modules includes: (1) a master subprogram for controlling all other modules in automatic mode, (2) high and low levels of user interface (UI) for flexible handling of peripheral devices, (3) module cores under a hardware abstraction layer (HAL) for communicating with hardware, and (4) processing and saving cores for simple signal processing and data storage.

The unique features enabled by this design include: (1) >1 kHz data acquisition repetition rate for MS imaging with

^{a)}Author to whom correspondence should be addressed. Electronic mail: lhlanley@uic.edu

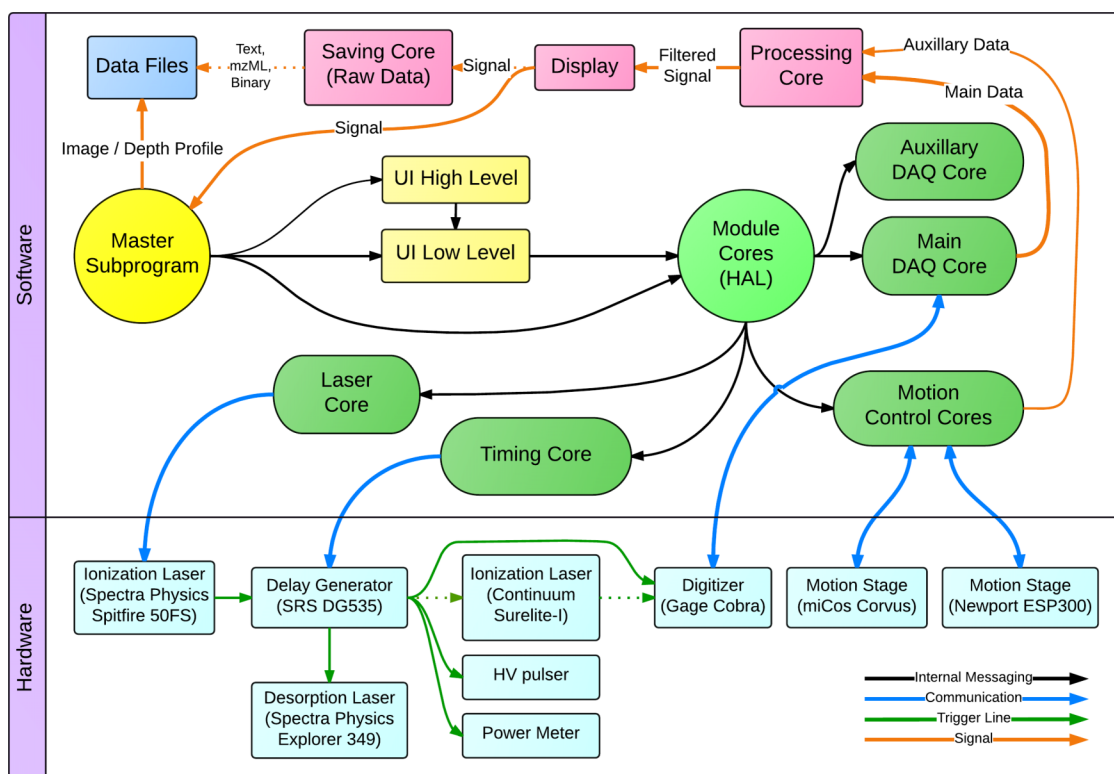


FIG. 1. Illustration of software concept/structure and hardware connections of the ChiMS data acquisition and control software package. Solid lines represent regular operating modes, and dotted lines show alternative modes. Dotted lines represent alternative operation modes. Cyan and green blocks represent actual hardware units and software communication cores, respectively. Colors for other blocks are chosen to group them by type. The general communication direction follows from dark to light yellow, then green, and finally cyan. Pink colored blocks do not interfere with the instrument control sequence. See text for further description.

the ability to collect generic, pulsed time-of-flight mass spectra without signal averaging or summing, (2) flexible peripheral device control, (3) real time data processing, and (4) ready implementation of dynamic or data dependent experiments.

Each block in Figure 1 represents a loop or module inside the software. There is a master subprogram on top of the high and low levels of the UI that runs the instrument in automatic acquisition mode to facilitate high volume data acquisition for applications such as MS imaging. The low level UI sends messages to the HAL modules that communicate bi- or unidirectionally with the hardware. Acquired data are fed into a processing core for simple digital filtering while the high level UI displays instrumental parameters. The processing core handles simple data filtering tasks such as pseudo pulse counting, noise suppressed accumulation, and/or baseline correction, then passes the processed filtered data to either a master subprogram for future data sorting, or the UI for display and streaming of raw data to storage media. The master subprogram is capable of processing raw data into images for display and/or saving to disk.

Module cores—colored green in Figure 1—are designed to encompass or wrap around the function of traditional LabVIEW instrument drivers, while additionally handling commands or other messages received from the upper level program. The module cores are usually single loop programs that respond to commands received from the queue structure, although some utilize two single loops (if the hardware device has both blocking and non-blocking operations). These module cores must be designed to handle all actions gracefully

while preventing unexpected module seizure. The module cores are also capable of running in standalone mode to facilitate testing and debugging. Running standalone also helps isolate potential issues from integration with the main program during the early prototyping stage of instrument development. A simple HAL is written on top of the LabVIEW instrument drivers to simplify switching between hardware with similar functionality from different vendors.

The user interface is separated into high and low levels. The high level UI accepts experimental parameters input by the user and converts these into hardware parameters that are then passed to the low level UI, where they are stored and passed to hardware without further modification. However, the hardware parameters in the low level UI are directly accessible to the user to permit trial experimental configurations without software modification.

Two types of master subprograms based on templates are included in ChiMS for different levels of customized experiments. The simpler, statically controlled master subprogram does not employ any online signal processing, but streams signal data to disk at the highest possible transfer rate. The more complicated, dynamically controlled master subprogram is used when online data processing is needed, such as in MS imaging. Thus, the dynamic master subprogram must control the operation, analyze the data, record results, and generate necessary instrumental parameters on the fly. Overall, general instrumental behavior is determined by the triggering scheme that is defined by hard-wired delay generator connections, timing values set in the hardware, and hardware action

sequences defined in the static or dynamic subprograms of ChiMS.

B. Instrumental details

The configuration of the laser desorption/ionization MS (LDPI-MS) instrument was detailed previously^{15–18} and is only briefly summarized here. This laser-based imaging/depth profiling mass spectrometer has the choice of two lasers available for desorption (Spitfire 50FS and Explorer 349, Spectra-Physics), and another for photoionization of desorbed neutrals (Surelite I-10, Continuum). The third harmonic from an ultrafast amplifier (Spitfire 50FS, Spectra-Physics) is also available for photoionization.^{16,20} A high precision vacuum motion stage (PLS-85, PI miCos) is used for sample movement. A motion stage with controller (ESP300, Newport) is used to control laser delay line and laser pulse energy. Customized ion optics were designed and assembled in house. A high speed digitizer (1 GS/s, 500 MHz, 8-bit, Cobra, Gage Applied) is used for signal acquisition from a multichannel plate ion detector. Several laser power/energy meters (EPM2000, Coherent, and Nova II, Ophir) are used for laser power/pulse energy measurement. Various digital delay gate generators (DG535, Stanford Research Systems) are used to control timing. The computer used in the reference instrument is equipped with a quad-core processor (Xeon E-1230, similar to Intel Core i5), 16 GB memory, a mechanical hard drive in a RAID 0 array, a 256 GB solid state hard drive (SSD), Windows 7, and LabVIEW 2013. ChiMS will also work (with minor changes) on computers with fewer hardware resources as well as those running Windows 8 or 8.1 and/or LabVIEW 2012 through 2014.

III. RESULTS AND DISCUSSION

A. High throughput MS imaging

Specialized hardware is required to achieve high throughput MS imaging: a high repetition rate laser, a digitizer with high data throughput and low latency such as a fast analog-to-digital converter (ADC) or time-to-digital converter (TDC), a motion stage with a rapid position update rate, and high slew rate pulse generators that can match the pulsed laser repetition rate. The instrument used here is a LDPI time-of-flight MS configured to run at 1 kHz repetition rate with a ns pulse length, UV Nd:YLF laser for desorption and an ultrashort pulse, Ti:sapphire laser outputting UV for photoionization. The remainder of this section further describes the ChiMS software in the context of how it facilitates 1 kHz MS imaging with this particular instrumental configuration. A 10 Hz MS image of a patterned pentacene sample using ChiMS running a similar instrumental configuration was recently published.¹⁸

Figure 2(a) shows the base UI of ChiMS, which mimics the function of an oscilloscope. Most of the functions of the base UI involve high level control of the instrument whereby mass spectra are displayed as long as the lasers are firing. The spectra are continually refreshed, which is particularly convenient for tuning instrumental operating parameters. The base UI provides control to all implemented hardware functionality.

The application menu guides the user to a master subprogram configuration page depicted in Figure 2(b) which allows users to provide parameters for MS imaging such as imaged area, pixel size, and number of averaged spectra per pixel. ChiMS calculates all the required hardware parameters from input values such as motion stage velocity, then combines them with queried instrumental status for input to the master subprogram. Figure 2(c) shows the master subprogram interface for MS imaging that controls the instrument in automatic mode and analyzes data online (see below).

B. Laser and timing cores

The laser and timing cores control the lasers and digital delay generators to fire or start at the expected time with an accuracy of nanoseconds. The amplifier of the Ti:sapphire laser employed here cannot be externally triggered and instead is used to drive the overall experimental timing sequence (see Figure 1). Thus, the Ti:sapphire laser triggers the digital delay gate generator (delay generator), which in turn must wait slightly less than 1 ms to trigger the desorption laser (no postionization signal is produced from the first ionization laser pulse). The delay generator also triggers the high voltage pulse generator to ramp the voltage on the ion optical electrodes to reject direct ions while extracting photoions. Finally, the digitizer is triggered by the time zero output pulse on the delay generator, which serves as the starting time of the time-of-flight spectrum.

Switching between different laser desorption/ionization schemes is done via reconfiguring the delay generator by load configuration files and swapping physical trigger lines (cables) between the lasers. The load configuration files (de)activate different delay generator channels and set both delay times and laser firing sequences. Drivers were written for two common delay generators (DG535, SRS and 500 series, BNC), allowing their rapid interchange despite hardware incompatibilities.

C. Main data acquisition (DAQ) core

The speed of the main DAQ core is responsible for acquiring and processing of the ion signal and is the central driving force of the ChiMS software data flow. The digitizer is typically a high speed ADC or TDC with an on-board histogrammer. The main DAQ core of ChiMS transfers binary integer data from the digitizer, then converts these data from integer to floating number format (in volts) for module compatibility. After this conversion, the data are sent to the processing core for simple signal processing tasks that include adjusting zero baseline, noise suppressed accumulation, pseudo pulse counting, and/or spectral averaging/summing. Two types of digitizers are currently supported by ChiMS: various high speed ADCs (100 MS/s to 4 GS/s, 8-bit to 16-bit PCI and PCIe, Gage Applied) and a low cost multipurpose device (100 MS/s, 14-bit, 10 MHz, dual channel, Analog Discovery, Digilent). However, many other types of data acquisition hardware can also be supported with only minor revisions to the ChiMS software.

ChiMS was specifically designed for rapid data collection and transfer rates. The typical sequence denoted in Figure 3

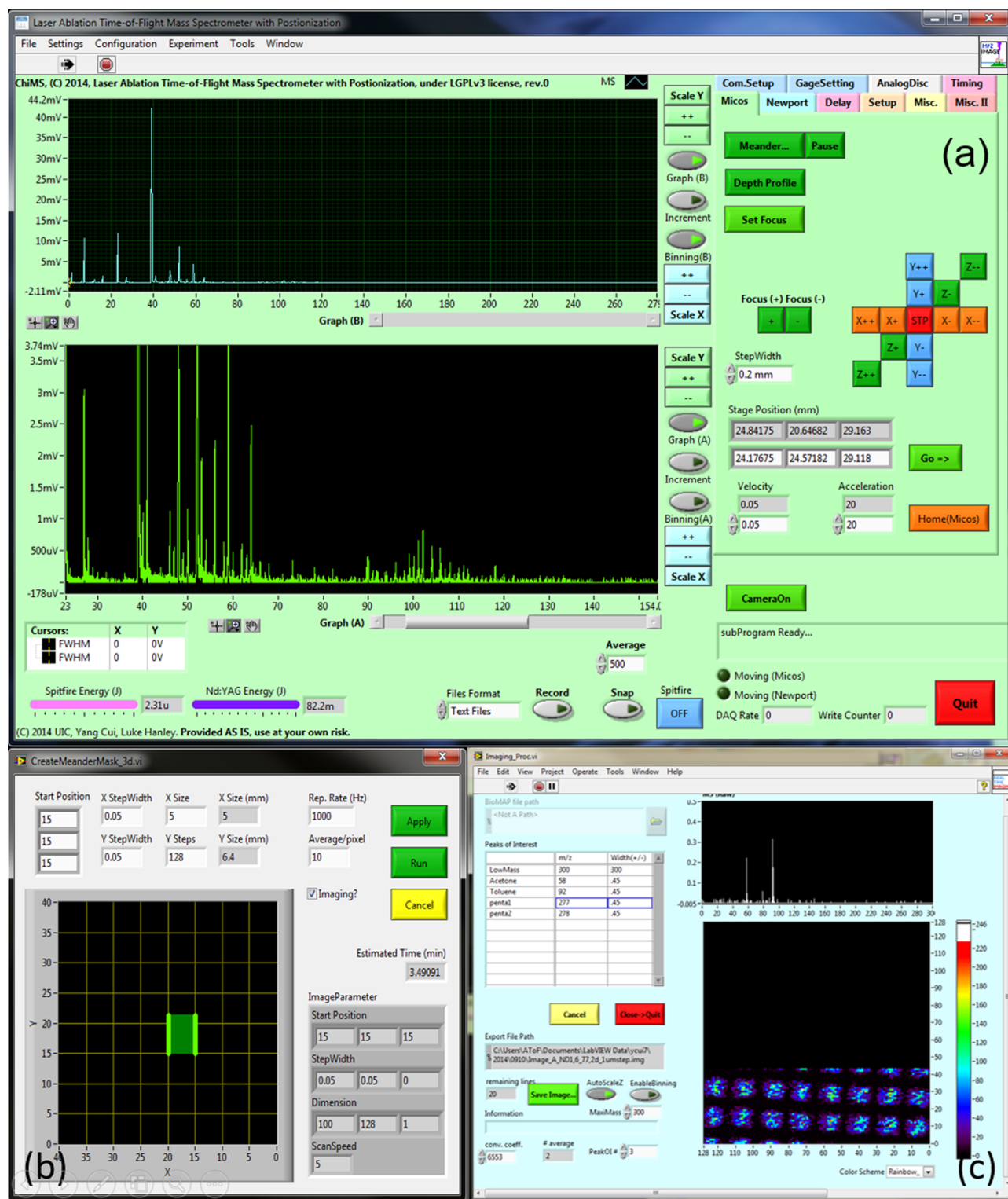


FIG. 2. (a) ChiMS base user interface for MS imaging experiment, (b) configuration master subprogram page for inputting experimental parameters, and (c) dynamic control master subprogram interface during an MS imaging experiment.

(top) as the “common simple method” is straightforward to write and deploy, but uses computer resources inefficiently. For example, this strategy cannot achieve a 1 kHz repetition rate because the software has to wait for the completion of raw data conversion, spectra processing, and (most time consuming of all) data saving before starting the next cycle of data acquisition. Even when file saving is offloaded to a thread in parallel,

data conversion and processing starts to slow the software at repetition rates >100 Hz because background applications can occasionally occupy all the resources of the computer’s central processing unit (CPU) in Windows 7. Thus, the CPU may delay action on time-sensitive hardware functions such as arming the digitizer to acquire signal before every laser pulse.

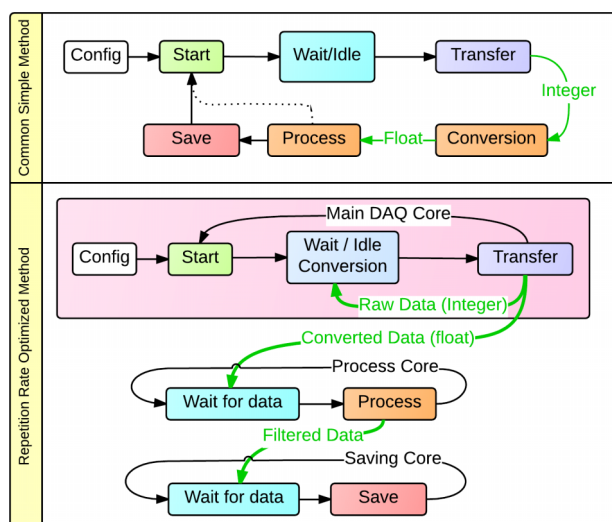


FIG. 3. Flowchart comparison of common simple method (top) versus repetition rate optimized method of data acquisition (bottom). See text for discussion.

ChiMS achieves rapid data acquisition by minimizing the digitizer idling time while offloading as many tasks as possible to parallel threads. This strategy is denoted in the pink block in Figure 3 (bottom) as the “repetition rate optimized method.” It is designed to both re-arm the digitizer as soon as data transfer is finished and convert raw digitizer output while the digitizer is recording signal for the next cycle. The general digitizer operation sequence starts with the armed hardware waiting for a trigger to acquire a spectrum. The program will run a digitizer status query in a loop with a no wait timer (a compromise due to the lack of sub-millisecond wait timer in Window 7 versions of LabVIEW). The main DAQ core switches to data transfer as soon as data acquisition finishes, then immediately re-arms the digitizer before performing data conversion, minimizing the chance of missed data acquisition cycles.

ChiMS has a triple data buffer structure to fully utilize the multicore CPUs ubiquitous in modern computers to decouple the data flow dependency between acquisition, processing, and saving. The ChiMS triple data buffer design consists of a raw integer conversion buffer in the main DAQ core, a floating number signal accumulation buffer in the processing core, and a final buffer in the master subprogram for data analysis. Utilization of LabVIEW’s data flow driven model²¹ allows computational work on the three buffers to be distributed across the multicore CPU to allow execution in parallel, improving main DAQ core responsiveness, and allowing higher overall data acquisition rates in ChiMS.

D. Motion control core

The motion control core is used to drive the motion stages used predominantly for scanning sample position with respect to desorption laser beam, a requirement for MS imaging experiments. A common, but slow, method for sample scanning is to repeat the sequence of moving then stopping the motion stage and firing the laser. Continuous motion scan imaging allows much higher repetition rates,²² but requires the highest posi-

tion update rate possible. The device driver was completely rewritten to optimize the overall ChiMS position update rate to 170 Hz. Prior to this update, a jitter in position referencing resulted in pixel misalignment during data processing.

The motion control core has both a non-blocking and a blocking loop and is based on the fundamental communication design of some motion stages controllers that will not respond to the computer if they are running a blocking command. Commands are sent to the non-blocking loop first which then decides whether to execute the command itself or pass the command to the blocking loop.

Two types of motion stage controller are currently supported (Corvus controller series, PI miCos and ESP series, Newport). Both share the same module interface with hardware differences hidden by the HAL (see above), facilitating replacement without modification of software.

E. Processing core

ChiMS has also a data processing core that receives and processes data acquired from the main DAQ core. Simple calculations can be performed with the data processing core, including digital filtering, binning, and signal averaging. The digital filter has three functions: zero offset adjustment, pseudo pulse counting, and noise suppressed accumulation.

F. Master subprogram for MS imaging

ChiMS can only generate and display simple mass spectra without higher level control. A dynamic master subprogram application programming interface was therefore devised for more complex experiments such as MS imaging, depth profiling, and laser beam profiling. The MS imaging master subprogram presented here is just one example of how dynamic master subprograms work in ChiMS. Parameters such as scan speed, average number of spectra per pixel, and overall image size are predefined through user input before the start of an experiment. However, the master subprogram application programming interface also allows modification of parameters during the experiment (i.e., in cases such as laser beam profiling, see below), where instrumental parameters are not known in advance of data collection and instead must be determined on the fly via calculation or feedback loop.

The dynamic master subprogram is provided as a template consisting of three major subroutines (known as subVIs in LabVIEW): sequence control, data sorter, and secondary parameter setter. The sequence control subVI manages higher level motion of the sample and sets other instrumental parameters by calling the secondary parameter setter subVI. The data sorter subVI averages raw spectra into pixels, integrates peak areas based on a user-defined list of m/z values of interest, generates images, and saves sorted images to disk in a standard format (Analyze v7.5 format, supported by the BioMAP image processing software, <http://www.maldi-msi.org/>). The data sorter subVI also allows users to view the image while data acquisition is still in process.

Figure 4 shows the overall topology and execution sequence of the MS imaging master subprogram. The master subprogram begins with the sequence control subVI moving

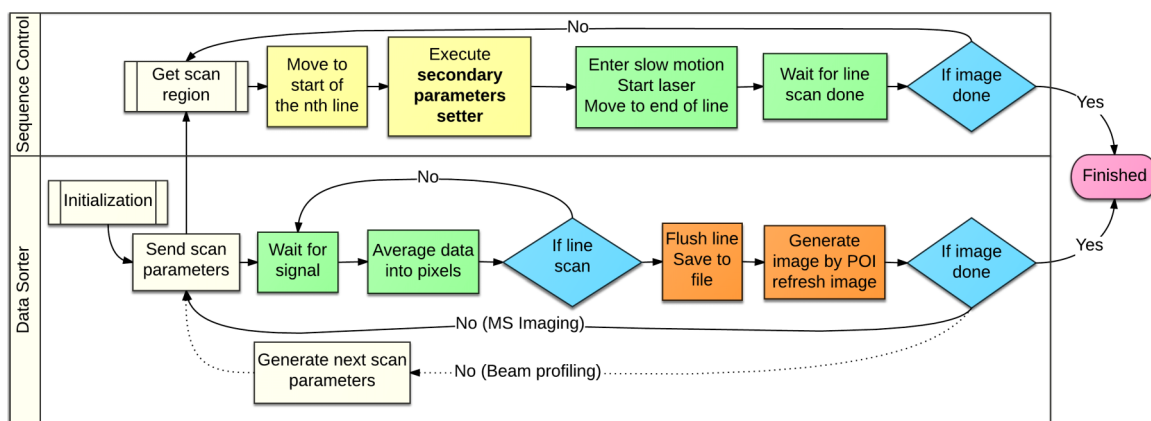


FIG. 4. Diagram of topology and execution sequence of master subprogram for MS imaging. Top and bottom boxes represent the sequence control subroutines (subVI) and data sorter subVI, respectively. The secondary parameters setter subVI is incorporated as a single file in the sequence control subVI. Dotted line represents an alternative routine.

the motion stage to the start location of the line scan, then waits for all ready signals from various experimental components (i.e., lasers, motion stages, and laser attenuator). Next, the sequence control subVI triggers the lasers and executes a line scan at the user pre-defined velocity until reaching the end of the line on the sample. During this process, the sync out signal of the fs laser triggers all equipment at 1 kHz via the delay generator while the digitizer acquires signal which it passes along to the processing modules (as described above). Simultaneous with these events, the data sorter subVI gathers signal filtered by the processing core and averages it into pixels, based upon the encoded motion stage position. Both subVIs run in a loop until the completion of each line scan. The sequence control subVI first checks if the full image is completed, then either starts the next line scan or terminates imaging. Finally, the data sorter subVI assembles all the acquired data into pixels, uses them to create one line of an image, and saves this line to disk. Preliminary images are generated in between line scans from the list of user-defined peaks of interest to provide a rapid image preview in the user interface while the overall experiment is still running.

Performance-wise, the MS imaging master subprogram acquires one 32k sample mass spectrum every 1 ms, then converts the raw 8-bit integer data to 32-bit single-precision floating point numbers for sorting to the image buffer. This process consumes hundreds of MB/s memory bandwidth. Both the Intel Xeon E3 or Intel Core i5 quad core processors can handle all these computations and higher repetition rates, provided such rates are allowed by the hardware.

G. Master subprogram for laser beam profiling

Precision laser beam profiling via the knife edge method²³ is used to evaluate lateral resolution in MS imaging.¹⁸ Laser beam focus size and focus location must be measured within submicron precision to successfully acquire ion images at the highest spatial resolution possible. Thus, the laser beam profiling master subprogram was created which represents an additional application for ChiMS beyond mass spectrometry. Details of the knife edge measurement have been published^{15,16,24} and are only summarized here. A knife edge is

scanned across a focused laser beam and records the light transmission via photodiode. The program fits the transmission curve to an algorithm, then repeats the scan at other axial positions to search for the minimum beam radius and Rayleigh range.

The benefits of automated beam profiling arise from the automatically determined variable scan velocity and length. Each beam radius measurement generates one transmission curve against scanning direction, then the curve fitting subVI calculates beam radius and location. Scan velocity and location calculated for each scan are used as guidance to assign parameters for the subsequent scan. By auto-fitting the parameters for each scan, idling time is minimized and useful data points are maximized. This strategy allows very rapid automated laser beam profiling while producing higher quality data resulting from collection of extra data points in the transition region.

H. Alternate modes, auxiliary tools, configuration saving, and portability

ChiMS can currently run several instrumental configurations without modification of its LabVIEW source code. One alternative instrument setup is discussed here to demonstrate software portability. Briefly, a laser front end was designed for a customized ion source coupled to a field deployable mass spectrometer with an atmospheric pressure interface (MTE50, MassTech).²⁵ The laser front end includes a motion stage with controller (ESP300, Newport) and a low cost multifunctional data acquisition device (Analog Discovery, Digilent) connected to an energy meter (J4-09, Molectron) for laser pulse energy measurement. ChiMS is used in this alternate setup for laser beam profiling and also sample rastering by the motion stage, with the acquisition of mass spectra handled by the software provided by the mass spectrometer vendor.

The Auxiliary DAQ core was created for slow data acquisition tasks such as monitoring power supply voltage outputs, vacuum pressure, and laser power (from analog output on power meter). The data acquired from the auxiliary DAQ core are fed into a control that is hidden on the front panel. The processing core regularly pulls data from the hidden control

and merges them with high speed signal from the main DAQ core, then sends merged signal to the processing core. The data flow and hardware are not critical, so they are omitted from Figure 2.

ChiMS also supports the saving to and loading from disk of configuration files which can set the instrument operational mode at the start of an experiment. This renders the instrument less prone to issues caused by incorrect hardware parameters, a common problem with customized instruments.

The ChiMS package also includes some auxiliary tools for mass calibration, text file viewing, and batch file processing. A file format converter is also available to generate MS imaging files from raw binary data files.

IV. CONCLUSIONS AND ChiMS AVAILABILITY

ChiMS is an open-source software developed for controlling custom-built imaging and depth profiling mass spectrometers and associated instrumentation. Optimized device drivers were incorporated to accommodate the highest laser repetition rates essential for high speed imaging. ChiMS is capable of running MS imaging experiments at up to 1 kHz repetition rates, hardware permitting. The associated instrument has been used to demonstrate $<2\ \mu\text{m}$ lateral resolution at 10 Hz repetition rate¹⁸ and is hardware limited to 200 Hz. The application of ChiMS has also been extended to depth profiling and laser physics experiments that include optical delay line, zero delay searching, and laser attenuator calibration.

A dynamically controlled master subprogram allows ChiMS to process and analyze signals online together with feedback loop capability to algorithmically adjust instrument parameters during experiments. One example of dynamic control is automated laser beam profiling, which requires a change of both scan region and velocity during the experiment. These parameters cannot be known before the experiment and must be calculated after each scan, so the devised master subprogram allows this auto-tuning. This strategy might be extended to automatic optimization of at least a few ion optical potentials for maximum ion transmission and mass resolution in simple mass spectrometers.²⁶ Although tandem MS is not implemented on the instrument used here, the interface can be used to programmatically select a precursor peak and run tandem MS, once such a capability is available. Finally, a similar laser-based imaging mass spectrometer has recently been reported that adapted ChiMS for yet higher speed data acquisition and control.²⁷

The ChiMS software package can be downloaded at <https://github.com/ycui7/ChiMS> and is licensed under a LGPL v3 license (<http://www.gnu.org/licenses/lgpl-3.0.html>).²⁸

ACKNOWLEDGMENTS

The authors acknowledge the contribution of Dr. Jerry F. Moore for his input on the various software features as well as for his occasional loans of useful equipment. Y.C. would also like to acknowledge the support of the University of Illinois at Chicago through a UIC Dean's Scholar Fellowship. The development of ChiMS was supported by the National

Institute of Biomedical Imaging and Bioengineering via Grant No. EB006532 and the U.S. Air Force Surgeon General's Office of Modernization under Contract No. FA7014-09-2-0003-T4T. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the National Institute of Biomedical Imaging and Bioengineering, the National Institutes of Health, the Air Force Surgeon General's Office, or the U.S. Government.

¹Wikipedia Contributors Open-Source Movement (Wikipedia, The Free Encyclopedia, 2015), see wikipedia.org, accessed 2014.

²M. Sturm and O. Kohlbacher, *J. Proteome Res.* **8**, 3760 (2009).

³R. Winkler, *J. Mass Spectrom.* **49**, 37 (2014).

⁴M. Strohalm, D. Kavan, P. Novák, M. Volný, and V. r. Havlíček, *Anal. Chem.* **82**, 4648 (2010).

⁵J. M. Fonville, C. L. Carter, L. Pizarro, R. T. Steven, A. D. Palmer, R. L. Griffiths, P. F. Lalor, J. C. Lindon, J. K. Nicholson, E. Holmes, and J. Bunch, *Anal. Chem.* **85**, 1415 (2012).

⁶C. Paschke, A. Leisner, A. Hester, K. Maass, S. Guenther, W. Bouschen, and B. Spengler, *J. Am. Soc. Mass Spectrom.* **24**, 1296 (2013).

⁷R. M. Parry, A. Galhena, C. Gamage, R. Bennett, M. Wang, and F. Fernández, *J. Am. Soc. Mass Spectrom.* **24**, 646 (2013).

⁸J. L. Hubisz, *Phys. Teach.* **52**, 318 (2014).

⁹D. J. Kim and Z. Fisk, *Rev. Sci. Instrum.* **83**, 123705 (2012).

¹⁰M. W. Little, J.-K. Kim, and K. K. Murray, *J. Mass Spectrom.* **38**, 772 (2003).

¹¹B. C. Laughlin, C. C. Mulligan, and R. G. Cooks, *Anal. Chem.* **77**, 2928 (2005).

¹²A. Akhmetov, J. F. Moore, G. L. Gasper, P. J. Koin, and L. Hanley, *J. Mass Spectrom.* **45**, 137 (2010).

¹³R. Hertz-Schünemann, R. Dorfner, C. Yeretizian, T. Streibel, and R. Zimmermann, *J. Mass Spectrom.* **48**, 1253 (2013).

¹⁴ni.com, NI LabVIEW Compiler: Under the Hood (2010), accessed 2014.

¹⁵Y. Cui, J. F. Moore, S. Milasinovic, Y. Liu, R. J. Gordon, and L. Hanley, *Rev. Sci. Instrum.* **83**, 093702 (2012).

¹⁶Y. Cui, C. Bhardwaj, S. Milasinovic, R. P. Carlson, R. J. Gordon, and L. Hanley, *ACS Appl. Mater. Interface* **5**, 9269 (2013).

¹⁷S. Milasinovic, Y. Cui, R. J. Gordon, and L. Hanley, *J. Phys. Chem. C* **118**, 28938 (2014).

¹⁸Y. Cui, I. V. Veryovkin, M. W. Majeski, D. R. Cavazos, and L. Hanley, *Anal. Chem.* **87**, 367 (2015).

¹⁹P. A. Blume, *The LabVIEW Style Book* (Prentice Hall, Upper Saddle River, NJ, 2007).

²⁰C. Bhardwaj, J. F. Moore, Y. Cui, G. L. Gasper, H. C. Bernstein, R. P. Carlson, and L. Hanley, *Anal. Bioanal. Chem.* **405**, 6969 (2013).

²¹Ni.com, Benefits of Programming Graphically in NI LabVIEW (2013), accessed 2015.

²²J. Spraggins and R. Caprioli, *J. Am. Soc. Mass Spectrom.* **22**, 1022 (2011).

²³M. Sato, in *Handbook of Charged Particle Optics*, edited by J. Orloff (CRC Press, Boca Raton, 2009), p. 392.

²⁴S. Milasinovic, Y. Liu, C. Bhardwaj, M. Blaze M. T., R. J. Gordon, and L. Hanley, *Anal. Chem.* **84**, 3945 (2012).

²⁵A. Misharin, K. Novoselov, V. Laiko, and V. M. Doroshenko, *Anal. Chem.* **84**, 10105 (2012).

²⁶A. Riedo, A. Bieler, M. Neuland, M. Tulej, and P. Wurz, *J. Mass Spectrom.* **48**, 1 (2013).

²⁷J. F. Moore, Y. Cui, V. Vaughn, E. Lewis, and J. A. Schultz, "Rapid imaging MS," in Proceedings of 63 ASMS Conference on Mass Spectrometry and Allied Topics MP163rd, America's Center, St. Louis, Missouri, 31 May–4 June 2015.

²⁸See supplementary material at <http://dx.doi.org/10.1063/1.4922913> for several features of ChiMS including its event driven user UI, general aspects of the high level UI, details of the digital filters, low speed DAQ core, auxiliary instrument control, file saving core, program configuration loading/saving, auxiliary toolkits, DAQ performance and limitations, and the list of currently supported hardware. The other software packages used and their interdependencies are also listed therein.