

SILIA: Software Implementation of a Multi-Channel, Multi-Frequency Lock-in Amplifier for the Extraction of Periodic Features From Data

Amrut Nadgir, Richard Thurston and Daniel S. Slaughter

Chemical Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, California, USA

Abstract

We describe a software implementation of a multi-channel, multi-frequency Lock-in Amplifier to extract modulated signals from noisy data distributed over collections of channels with arbitrary dimensionality and size. This software implementation (<https://github.com/AMOS-experiment/SILIA>) emulates the functionality of a multi-channel, multi-frequency lock-in amplifier in a post-processing step following data acquisition. We demonstrate the versatility and performance for extracting weak signals in spectroscopy and microscopy. We also discuss more general applications and exhibit a method to automatically estimate error from a lock-in result.

1. Introduction

Lock-in amplifiers are widely used to suppress noise from a periodic signal input with a known frequency. These applications include various types of spectroscopy and other fields including, but not limited to impedance spectroscopy (ZI Impedance Spectroscopy,; Sun Holmes 2007 microfluidic spectroscopy), deep level transient spectroscopy (Auret 1985 Lock-in DLTS), active reset of superconducting qubits (ZI qubit reset) and neuroscience (Giaconia Lock-in Brain Monitoring). Many multi-channel and multi-frequency lock-in amplifiers are expensive and offer little or no flexibility in the number of input channels and references. By implementing a software lock-in amplifier, we adopt the approach where the amplification is a post-processing step after the data acquisition is completed. A software implementation increases the adaptability of the software to the task at hand and simplifies the Lock-in process so it can be easily modified to different specifications.

This present Software Implementation of a Lock-In Amplifier (SILIA) can be used to analyze data from a suitably modulated system, having any number of signal channels and frequency references. As a result, it can serve as a simpler and more cost effective substitute for hardware lock-in amplifiers that are commonly used for multi-channel and multi-frequency applications. Many of the experiments and techniques involving lock-in amplification, such as lock-in imaging (Liu, Shen, Ma 2016 Lock-in Camera; Probst, Jaquier 1993 Lock-in PPM resolution; Foix Lock-in ToF Camera, Marriott 2008 live imaging in cell), pump-probe spectroscopy (Bourquin,Prasankumar 2003 femtosecond pump-probe spectroscopy; Kolarczik 2018 pump-probe quantum dot; Gilburd pump probe nanotubes; Fushitani Pump Probe Spectroscopy Applications) and pump-probe microscopy (Fischer, Wilson 2016 Review Pump-Probe Microscopy; Pu Ting Dong Pump Probe Microscopy Summary), require the use of specialty hardware that can perform the necessary computations during the course of the experiment.

With SILIA, researchers can use standard recording devices during the experiment and perform lock-in amplification afterwards with no additional cost. SILIA may also be used to extract periodic motion from pre-recorded videos without an explicit reference.

SILIA offers additional benefits not available to traditional lock-in amplification. One, it can interface with any time series in a specified format. Two, it has the capability to repeatedly retrieve the signal amplitude and relative phase of a weak signal after the initial data acquisition while modifying input parameters and performing post-processing analysis. Lastly, the software is customizable and readily adapted to a wide variety of spectroscopic or imaging instrumentation, and other devices. Since SILIA is applied as a post-processing step, the speed of the data acquisition is only limited by the digital instrumentation and there are no additional hardware optimizations required for execution. The software can also easily be manipulated to suppress noise from varied forms of measurement, such as frequency sweeps ([Sonaillon and Bonetto 2007 error correction frequency sweep](#)). SILIA is also able to estimate the error in the results by partitioning the input signal into intervals and locking into each of these subintervals.

In this article, we will first discuss the principles behind lock-in amplification and how SILIA implements them in software. After explaining how SILIA works, we move on to a basic multi-signal functionality test for the software where we showcase SILIA's ability to lock-into multiple channels, references simultaneously and produce accurate errorbars for its result. Then, we benchmark the runtime and noise suppression abilities of SILIA in the "Software Benchmarking" Section. After benchmarking, we apply SILIA to spectroscopy and a fluorescence microscopy simulation and present the results. Finally, we discuss the benefits, drawbacks of SILIA as well as its potential for improvement.

2. Lock-in Amplification Basics

The lock-in amplifier is able to accomplish noise suppression and phase detection by using phase-sensitive techniques to remove the fourier components of the input signal that are not modulated at the desired frequency. Lock-in amplifiers generally perform signal mixing and low-pass filtering ([SRS Lock-in tutorial; ZI WhitePaper Principles of Lock-in](#)). The signal mixing step multiplies the input signal by a reference. A sinusoidal reference ensures that the fourier components of the input signal that oscillate at the reference frequency are shifted to 0Hz in the frequency space representation of the mixed signal. A low-pass filter suppresses the frequency components of the mixed signal that are far from 0Hz. After the Lock-in process is completed, the filtered signal output will only contain the fourier components of the input signal that were oscillating at the initial reference frequency, thus suppressing noise. The phase of the signal relative to the reference can be extracted by comparing output between the in-phase and quadrature components of the reference ([SRS Lock-in tutorial; ZI WhitePaper Principles of Lock-in; Temple 1975; Scofield Frequency Domain Description](#)).

Early lock-in amplifiers were analog devices, while most modern lock-in amplifiers are fully digital or implemented on a field programmable gate array. Digital lock-in amplifiers tend to be simpler than

their analog counterparts, while also achieving wider bandwidth, greater adjustability and improved accuracy (Mandelis 1994; Wang, Wang, digital Lock-in Amplifier Scanning-Grating Spectrometer; Wang 1998 PC Digital Lock-in; Stimpson 2019 high freq lock-in; Giacconi 2017 FPGA Lock-in For Brain; Carminati 2016 limiting resolution digital lock-in). As a result of the apparent superiority of digital lock-in amplifiers, there has been great interest in the various possible implementations of these instruments.

3. SILIA: Software Implementation of a Lock-In Amplifier

Implementation

An overview of the software is shown in Figure 1. SILIA fits a sinusoidal function to the references and locks into each input channel with each reference frequency. If there are n input channels and k desired frequency references to lock-in to, then the software will output nk lock-in results.

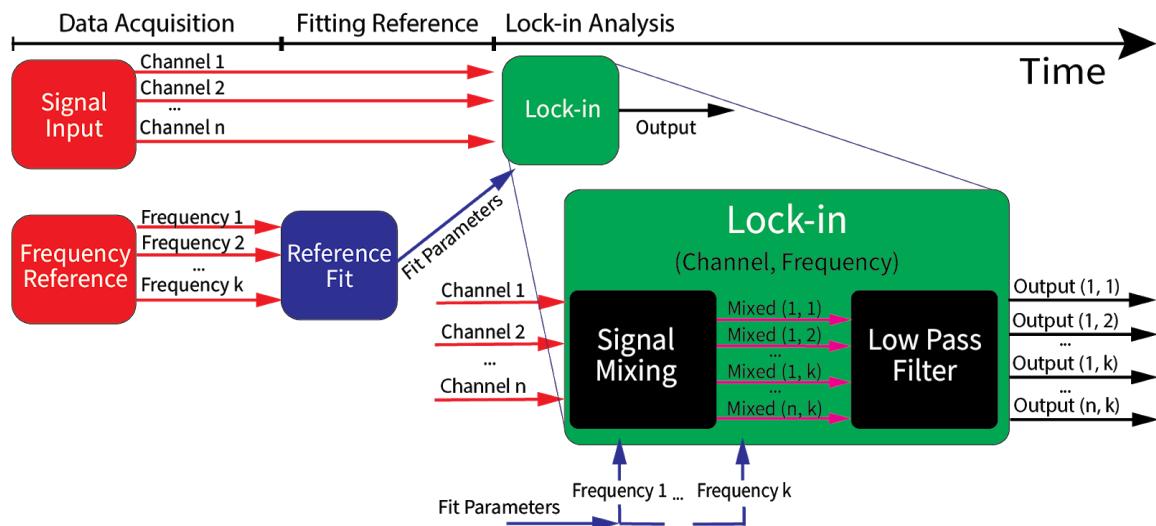


Fig 1: An overview of a single data acquisition cycle and lock-in process, showing the different steps required for the software lock-in analysis in temporal order. The final lock-in amplification step requires the signal data over all n channels and the k fitted frequency references as inputs. The signal mixing and low-pass filter steps are run on the input channels and references where every channel and reference pair is locked into separately. SILIA also has the option to skip the Reference Fit step.

SILIA requires the raw input signal for each channel, frequency references and a cutoff frequency for the low-pass filter. The raw input and frequency references are not passed into the software in real time and are processed after the data acquisition period. The data acquisition device will be required to store its measurements with timestamps in a format that will later be readable to the software.

For each reference, we compute the in-phase and quadrature mixed components for each signal channel. To mitigate error from nonuniform sampling rates, we performed cubic interpolation on the input signal to ensure evenly spaced samples. The interpolation is necessary because the Fast-Fourier Transform (FFT) algorithm we use in later parts of the analysis assumes the samples are regularly

spaced. After interpolation, the software applies a low-pass filter to each of the mixed signals. To implement a low-pass filter in software, we applied the real FFT with a Hanning window to the mixed signal and removed all components outside a symmetric interval centered on 0, with the width specified by a frequency cutoff parameter. This acts as a low-pass filter with a response curve that is essentially a step function, thus minimizing any error caused by the filtering step in conventional lock-in amplifiers, but also assumes that samples are evenly spaced (uniform sampling). Due to the discrete sampling of the signal input, the filter cannot differentiate between arbitrarily close frequency components of the signal. Therefore, the cutoff frequency is limited by the frequency resolution of the measured signal. Assuming nearly uniform sampling, the frequency resolution tends to improve with more data. In addition, the maximum frequency that can be locked into is given by $f/2$ and is equal to the Nyquist frequency of the data ([NI FFT tutorial](#)). Lock-in amplitudes and phases were computed from the measured in-phase and quadrature components of the filtered signals ([Bhattacharya Digital Lock-in](#)). Each data acquisition cycle produces an array of output values indexed by their respective channel and frequency reference (Fig. 1).

The fitted references are scaled so the output when locking into a purely sinusoidal signal that oscillates at the reference frequency is equal to the amplitude of the input. There can be additional, but predictable bias when the input waveform is non-sinusoidal. The most common example of this is with square wave inputs. The fourier series expansion of an even square wave with a peak to peak

amplitude of A and a duty cycle, d , is, $2A/\pi \sum_{n=1}^{\infty} 1/n \sin(n\pi d) \sin(2\pi nft)$ where t is time and f is the

frequency of the square wave in Hz. As long as the cutoff frequency of the lock-in is less than f , only the $n=1$ fourier component will remain after the low-pass filter. Therefore, the ideal lock-in output magnitude for a standard square wave signal with a duty cycle of $1/2$ would be $2A/\pi$. Since this type of bias is predictable, it can also easily be accounted for.

SILIA also has the ability to estimate inner products in frequency space between the reference and the input signal. To compute these projections, SILIA has the option to skip the reference fitting step and mix the signal input with a scaled raw reference. The raw reference is scaled by a factor of 2 so the inner product between two purely sinusoidal signals that oscillate at the same frequency is the product of their amplitudes. This feature gives SILIA the ability to lock-into non-sinusoidal references but it cannot extract phase information in this case due to a lack of knowledge about the structure of the reference Input.

SILIA has an additional built-in error estimation feature. Apart from the primary lock-in functionality, it can split the input signal into portions that may or may not overlap and lock into each part. After splitting and locking into the data, SILIA evaluates the standard deviation of the results. The size and number of these portions are specified by input parameters. There is a tradeoff to be made here since too little overlap and too many windows will result in an underestimation of the error, while too few windows with very large window sizes and high overlap will overestimate the error.

4. Multi-Signal Basic Functionality Test

Methods

We performed a series of simulations to test the lock-in capabilities of SILIA for multiple channels and multiple references acquired simultaneously, and to characterize the quality of the output lock-in signal and phase compared to the input signals and references.

For the data acquisition component of the simulation, we simulated 100 channels of data, some of which had only Gaussian noise while others had a synthetic signal, modulated at a specific reference frequency, as well as the noise. The signal was sinusoidal and had a power of $1/2$ while the noise had a variance of 1 (Fig. 2). We adopt the convention of SNR defined as $\frac{\text{signal power}}{\text{noise power}}$. Therefore, the SNR ratio in this simulation was $1:2$ for the channels that contained the input signals and the signal had an amplitude of 1.

The data acquisition is for a simulated time of 5 seconds with a sampling rate of 5000Hz and a frequency resolution of 0.2Hz. The cutoff frequency is also set to 0.2Hz. The references are 80Hz and 120Hz square waves.

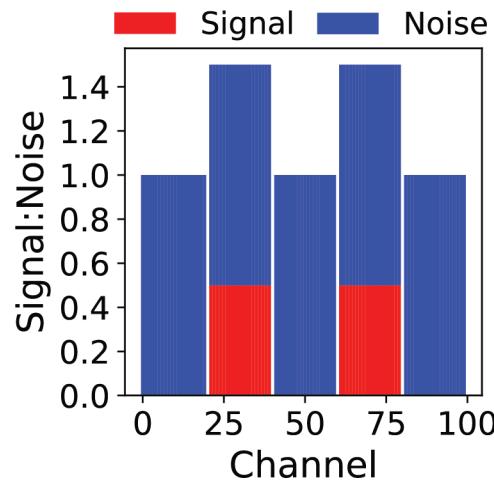


Fig 2: A graphical representation of the SNR for each channel of the input signal. We simulated 100 channels which we split into batches of 20 channels each. The first, third and fifth batch of 20 channels were Gaussian noise with a variance, or power of one. The second and fourth batches were sine waves with a power of $1/2$ and additional Gaussian Noise. One of the sine waves oscillated at 80Hz (channels 20-40) while the other had a frequency of 120Hz (channels 60-80).

Output values are generated by using SILIA to lock into 80Hz and 120Hz references. Uncertainties are estimated by separately analyzing four evenly spaced time-intervals with marginal overlap and computing the standard deviation of the output lock-in magnitude and phase across all intervals. The noise power is twice as large as the signal power (Fig. 2).

Results

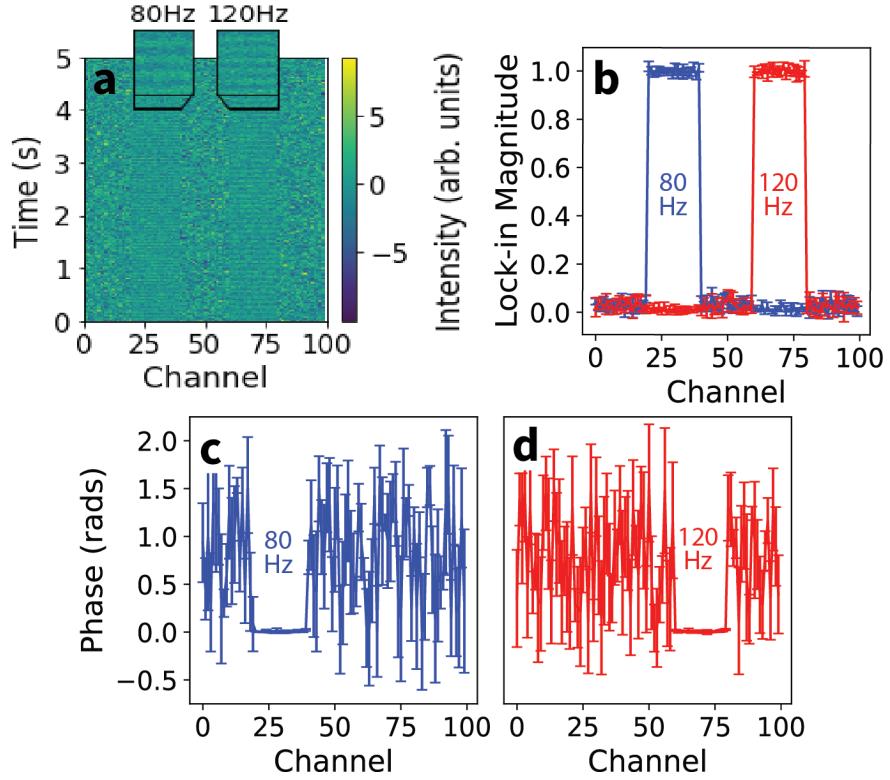


Fig 3: (A) Spectrogram of the simulated signal. The two zoomed insets show the noisy oscillating signals in channels 20-40 and 60-80, both signals oscillate at different frequencies. (B) The multi-frequency SILIA outputs. Both curves correctly identify the amplitudes of the 80Hz and 120Hz signals at each channel. The output for each channel at the peaks are equal, within the estimated uncertainties, and the uncertainties are greater when there is no signal present. (C, D) Phase output when locking into both frequencies. We see that the phase error is very small in the presence of signal and large for all noisy channels.

The input consisted of two sinusoidal signals oscillating at 80Hz and 120Hz respectively with an amplitude of 1 and a higher-amplitude additional Gaussian noise over each channel (Fig. 3A). We expect SILIA to suppress the Gaussian noise as well as any signal not oscillating at the reference frequency. When locking into the 80Hz or 120Hz signal, we should see an output magnitude near 1 for channels 20-40 or 60-80 respectively, and near 0 everywhere else. When locking into the 80Hz and 120Hz signals, we see mean peak values of 1.000 with standard deviations of 0.009 and 0.010 as well as mean outputs in channels without signal of 0.020 and 0.016 respectively which is almost exactly what we expect (Fig. 3B). The error bars at the peak values for the 80 and 120Hz results are, on average, 0.018 and 0.020 which are overestimations of the standard deviation of the output at peak signal by a factor of two but still give a useful estimate of fluctuation in output (Fig. 3B). The output phase of 0 in the presence of signal is the same as the relative phase between the input signal and reference. For the 80Hz and 120Hz outputs in the channels having signal, we observe a phase of 0.007 and 0.009 respectively with standard deviations of 0.006 and 0.008 and mean errors of 0.009 and 0.010 respectively (Fig. 3C, D). The phase of the results closely match what we expect and the error bars provide a reasonable estimate of the standard deviation in phase results. We also observe fluctuations in phase where signal is not present since SILIA can only extract noise in those channels (Fig. 3C, D).

5. Software Benchmarking

Methods

To test the speed and effectiveness of the software, we ran a few different benchmarks. Our benchmarking software measured the amount of time it took for SILIA to process different amounts of input data, and also tested the change in the signal to noise ratio (SNR) for different amounts of data.

For our initial benchmarks, we measured the amount of time it took for the software to run for a variable number of input samples, channels and references by varying each of those three parameters individually while holding the other two constant. We used constant values of 4096 samples, 10 channels and 1 reference. This benchmarking allowed us to confirm the linear runtime of SILIA with respect to the data acquisition time, size of each sample and number of signals (Fig. 4).

We also ran benchmarks that quantified the percentage error of the output magnitude while varying the number of simulated input signal cycles and the samples per cycle with constant parameters of 10 samples per cycle and 5000 cycles respectively. We added Gaussian noise to the input with varying standard deviations and the input signal was a sinusoidal wave with a root mean square of 1. Our results were averaged over 100 runs and compared with a theoretical estimate of the results from averaging (Fig.5A, B).

To visualize the response curve of the low-pass filter, we varied the simulated signal frequency to 100 \pm 5Hz but kept the reference frequency constant at 100Hz. The input signal was a sinusoidal wave with an amplitude of 1 and had no additional noise added to it. We plotted the Lock-in output magnitude with respect to the input signal frequency with effective cutoffs of 0.06Hz and 1.02Hz (Fig. 5C).

We additionally quantified the reaction of our lock-in to a class of non-sinusoidal waveforms. Due to their commonality, we chose to use 100Hz square waves with peak to peak height of 1 and varying duty cycles. After generating the signals and running the lock-in, we proceeded to confirm that our results matched the primary Fourier component of the signal (Fig. 5D).

Results

The results from benchmarking help quantify the time complexity of SILIA, as well as its robustness and how it reacts to different sources of error. By quantifying the different aspects of this software implementation, we are able to better highlight its strengths and limitations.

Runtime Benchmarking

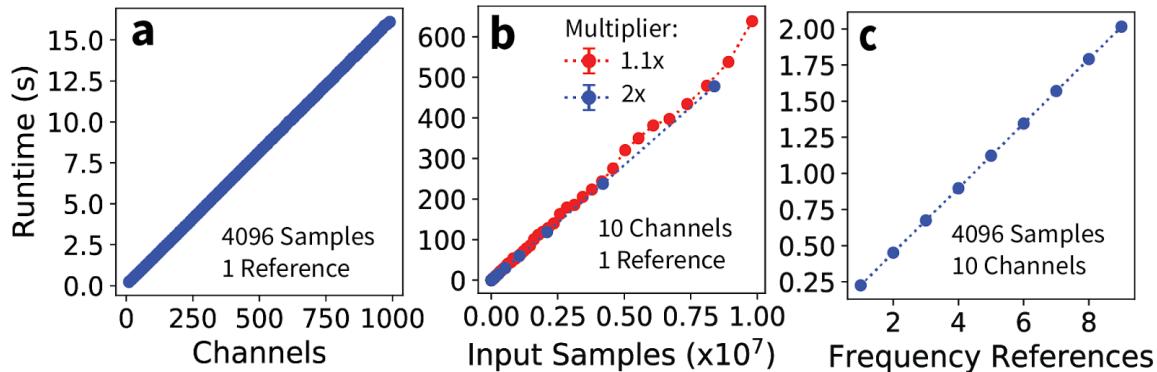


Fig 4: (A) Demonstrates the linear dependence of runtime on the number of input channels. (B) Dependence of runtime with respect to the number of input samples, showing a general linear relationship between the runtime and data acquisition time, and some nonlinearity when the number of input samples is scaled by 1.1 instead of 2. (C) Demonstrates a linear relationship between runtime and the number of frequency references.

As the number of input channels increases, the total runtime of the software rises proportionally (Fig. 4A). We see a similar trend with the number of frequency references, where adding an additional reference frequency to the SILIA input will increase the runtime of the software by a constant amount (Fig. 4C). Both trends can be attributed to the software essentially performing the analysis separately and in sequential order for each channel and frequency reference. We also observe a linear increase in runtime with respect to the number of input samples when each benchmark has 2 \times the number of samples as the previous. However, after reducing that multiplier to 1.1 \times , we see nonlinear fluctuations which result in increased runtimes (Fig. 4B). This is likely due to our use of the radix-2 FFT algorithm, which has a runtime that is optimized for input sample sizes that are a power of 2. (Muqri, Wilson 2015; Amirfattahi, 2013).

It is important to note that these benchmark plots do not necessarily reflect the exact runtime of the software in general, since those runtimes are dependent on the hardware used to run SILIA and potential new optimizations in the code. The observed linear trends imply that the post-processing time will generally be proportional to the data acquisition time.

Error Benchmarking

When benchmarking the effects of experimental errors, we focused on three potential sources of inaccuracies - Gaussian or oscillating noise in the input signal, low-pass filter response and non-sinusoidal signal. A low-pass filter response with low roll-off can introduce unwanted artifacts from frequencies outside the cutoff range into the result of the lock-in amplification, and, as mentioned in section 3, non-sinusoidal input can introduce bias into the result.

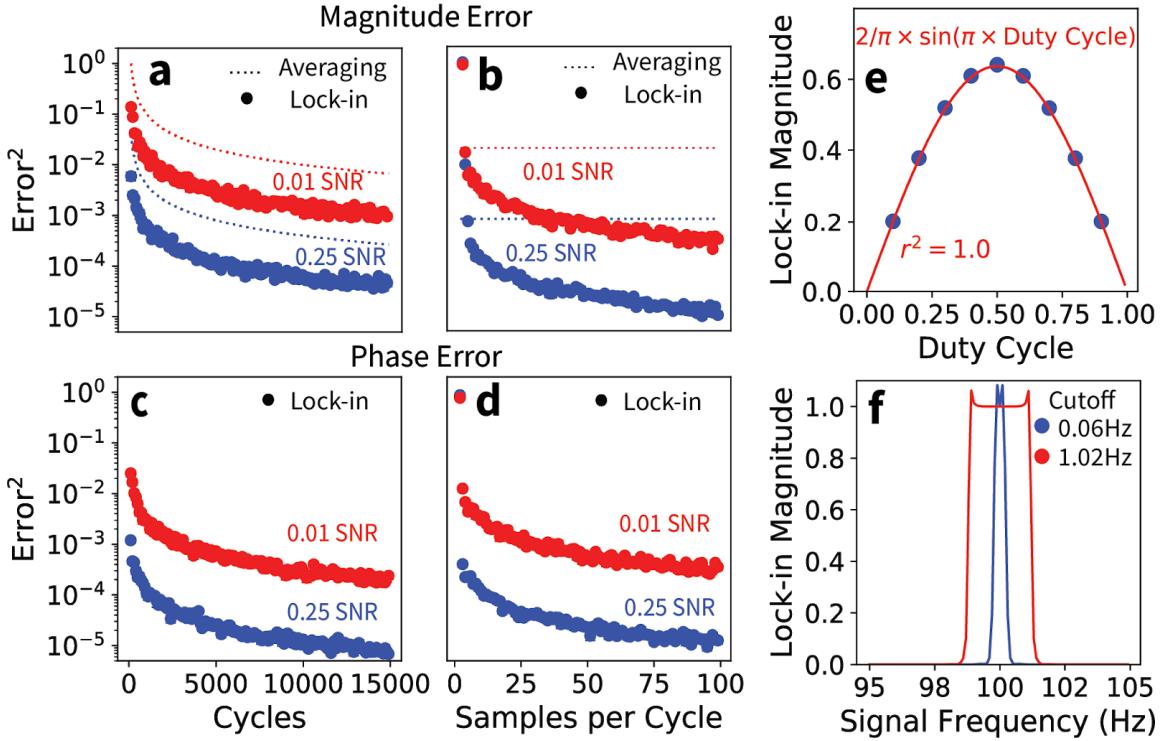


Fig 5: (A, B) Demonstrates a steep decrease in Mean Squared Error of output magnitude with respect to the number of input cycles (Fig. A) and the sampling rate (Fig. B). We show that our method produces superior results to averaging each cycle of data. (C, D) SILIA phase output rapidly increases in accuracy with respect to the number of input cycles (Fig. C) and the sampling rate (Fig. D). (E) Low-pass response curve for a constant reference frequency of 100Hz. (F) Output as a function of the duty cycle of a square wave input signal. SILIA outputs are fit to the red curve, which is the first term in the Fourier series expansion of a square wave with varied duty cycles.

Output error of lock-in amplifiers tends to decrease sharply with respect to data acquisition time and SNR (Bhattacharyya, 2016; Mandelis 1994; Van Baak 2014; Neelakantan, Dattagupta, Rajappan 1979). We see a similar significant improvement in the output error with respect to increased data acquisition time (Fig. 5A, C), and can conclude the necessity of longer data acquisition times for more accurate results, especially from data with low SNR. We see that larger sampling rates yield superior output (Fig. 5B, D) and that SILIA outperforms signal averaging (Fig. 5A, B). The squared error from signal averaging was calculated using the theoretical standard of **Variance = Noise Power/Number of Cycles**. The shape of the input signal waveform affects the output in an easily quantifiable manner. The implementation extracts sinusoidal signal from noise, we expect and observe the output for square waves with different duty cycles closely matches the primary Fourier component of the input signal (Fig. 5E). Furthermore, the significant decrease in the output magnitude when the noise oscillates outside the cutoff frequency range (Fig. 5F) demonstrates the effectiveness of SILIA at filtering out periodic noise.

6. Application 1: Spectral Analysis

Methods

To test SILIA in an experimental setting, we used the Ocean Optics OCEAN-FX-XR1-ES spectrometer to measure amplitude-modulated laser light, and the fluorescence spectrum emitted from a dye that was excited by 532 nm laser pulses from a ThorLabs CPS532 diode-pumped solid state laser with a continuous-wave power of 4.5mW (Fig. 6). To modulate the amplitude of the signal and measure a frequency reference, we used a National Instruments USB6001 data acquisition device connected to a Thorlabs MC2000B chopper.

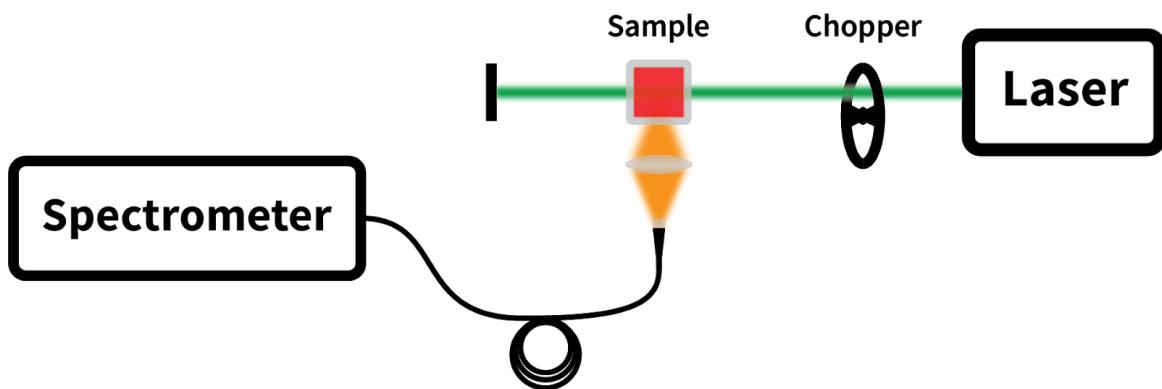


Fig 6: Experimental setup to measure the fluorescence spectrum of a Rhodamine 6G sample and test whether SILIA can suppress noise from our measurement. A chopper wheel is used to modulate the laser which excites the Rhodamine while a lens is used to focus the resulting fluorescence signal into the spectrometer.

During the data acquisition, we used spectrometer readings from a wavelength range from 490nm to 700nm and a wavelength resolution of $0.42 \pm 0.02\text{nm}$. We measured our reference input with a 2000Hz sampling rate from the chopper and chopped the laser at 100Hz for the laser spectrum measurement. The sampling rate of the spectrometer had a mean of 230 Hz and a standard deviation of 49Hz ($230 \pm 49\text{Hz}$), with a two second runtime per data acquisition cycle for three cycles. Using SILIA, we locked into the spectrometer signal in an attempt to suppress any background noise in our measurement (Fig. 7).

For the fluorescence spectrum measurement, we induced fluorescence in a 1.322mM, 95% purity Rhodamine 6G sample in ethanol and measured the output spectrum with the same wavelength range and resolution as mentioned above (Fig. 6b). We used a 250.0 mm convex lens to focus the fluorescence signal into the spectrometer. The laser was chopped at 50Hz and our reference input was measured from the chopper with a 2000Hz sampling rate. For this measurement the sampling rate of the spectrometer was $249 \pm 58\text{Hz}$, with a three second runtime per data acquisition cycle for three cycles. We used SILIA to lock into the fluorescence signal and compared our results to the PhotoChem database (Two Rhodamine, PhotoChem tagged papers) (Fig. 8).

Results

Laser Signal with Noise

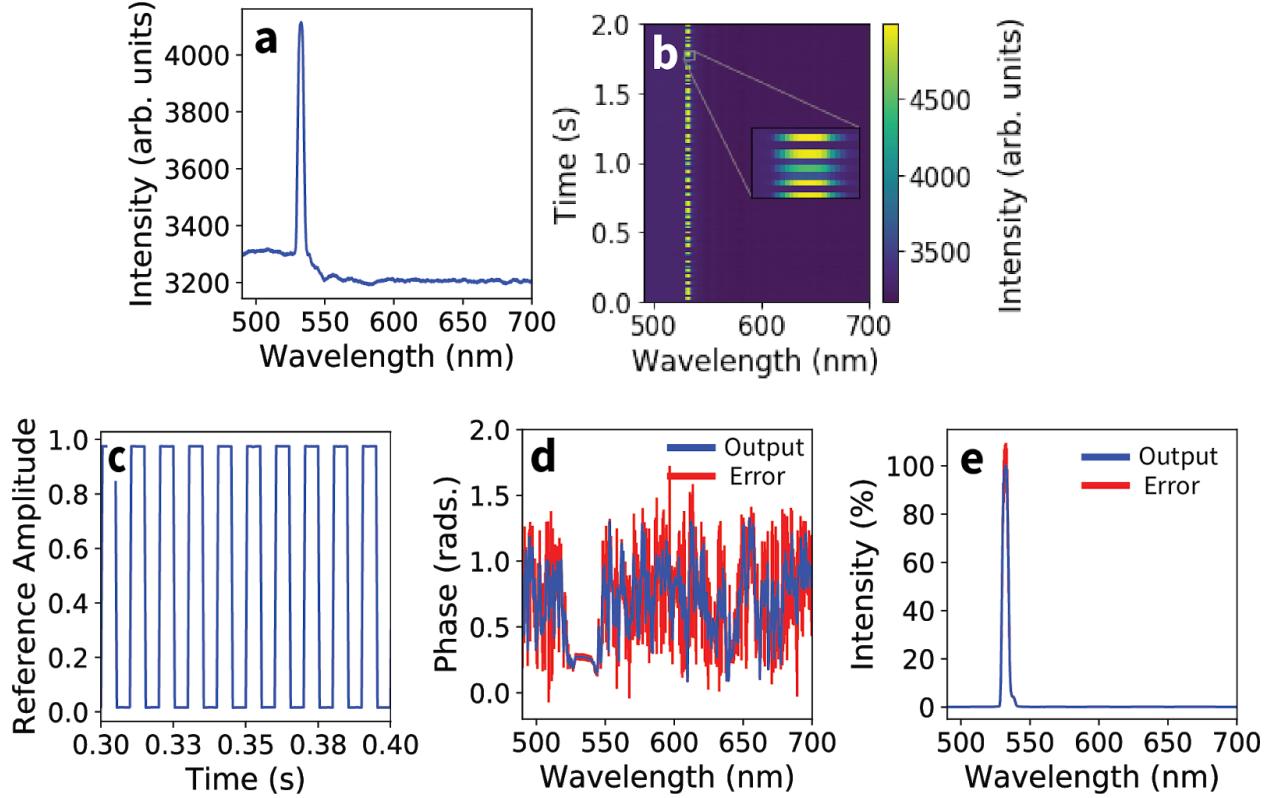


Fig 7: (A) The time-averaged signal of the spectrometer. The peak from the chopped laser signal can be seen at 532nm. (B) A spectrogram of the chopped laser signal from a single data acquisition cycle of the spectrometer, showing the 100Hz oscillation of the signal input. Irregular sampling of light by the spectrometer leads to intensity variations of $\pm 33\%$ in laser peaks. (C) The optical chopper output signal and frequency reference over time, which shows 100 ± 1 Hz oscillation. (D) SILIA output phase averaged over three data acquisition cycles demonstrating phase convergence in the presence of signal. (E)SILIA output magnitude averaged over three data acquisition cycles showcasing a clear laser signal.

SILIA output suppresses background signal (Fig. 7A) and cleanly illustrates the peak of the laser signal at 532nm (Fig. 7E) and suppresses background signal (Fig. 7A). Over the three data acquisition cycles, the peak value of the output fluctuated with a standard deviation of 9% (Fig. 7E) which is likely due to the inconsistencies in the spectrometer measurement (Fig. 7B). We see the expected fluctuations in phase in the presence of no signal as well as a clear convergence in phase near 532nm (Fig. 7D). The phase convergence between wavelengths demonstrates the expected correlation in the output between wavelengths with signal whereas lack of error in the phase convergence region indicates the experimental setup was consistent over the three data acquisition cycles. We observe slight dips in phase on the outskirts of the convergence region which can be explained by noting that the edge of the signal has a lower intensity than the center which could lead to increased output distortion due to noise.

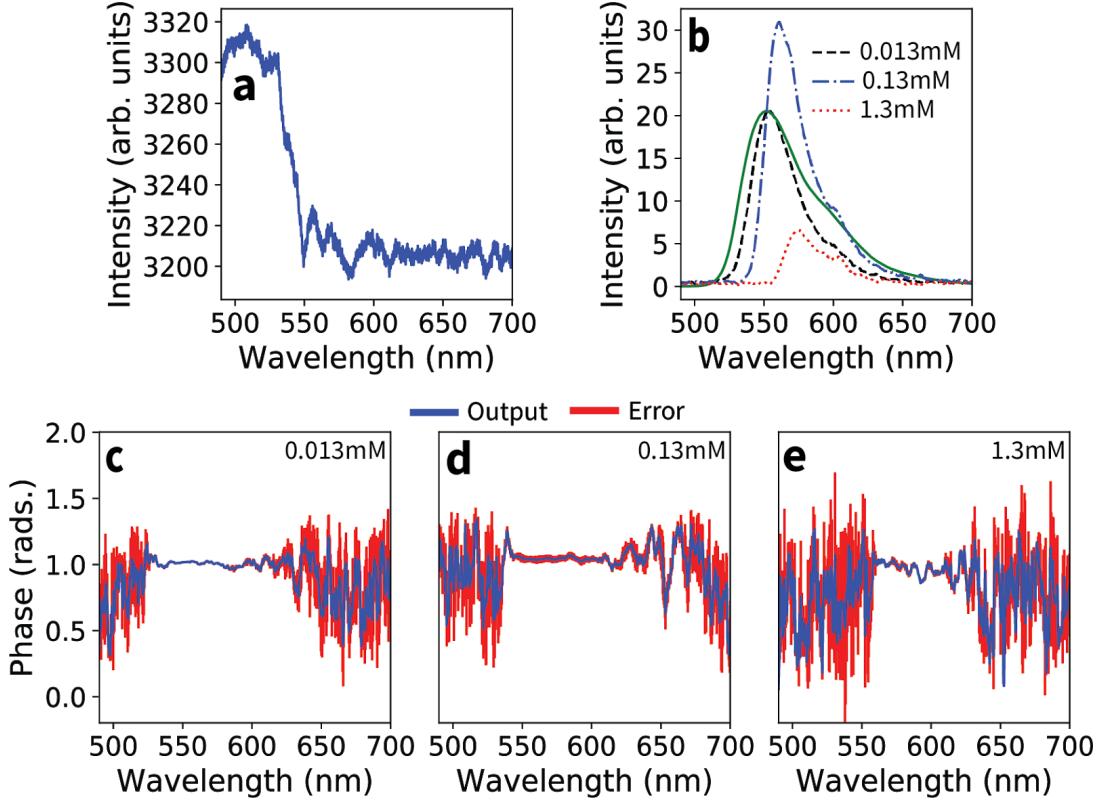


Fig 8: (A) Typical time-averaged signal of the spectrometer. The rhodamine spectrum is invisible in the spectrometer readings. (B) SILIA output for different concentrations of Rhodamine averaged over 3-5 data acquisition cycles. The peaks become more distorted with increasing Rhodamine concentration and have varying heights due to the primary and secondary inner-filter effects. The spectra are plotted against the green curve, a rescaled reference rhodamine 6G fluorescence measurement (Two Rhodamine, PhotoChem tagged papers). Our 0.013mM results closely match the reference despite being taken with inferior equipment. Output peaks shift to higher wavelengths when rhodamine concentration is increased. (C,D,E)) SILIA phase output for the various concentrations of rhodamine. We observe a consistent phase convergence across all three concentrations in the presence of signal.

As another demonstration of the robustness of the SILIA approach, we used its capabilities to lock into a rhodamine fluorescence signal without using a photomultiplier following optical excitation at 532 nm, where the output magnitude illustrates a rhodamine spectrum peak (Fig. 8B) that was initially invisible in the presence of background noise (Fig. 8A). The SILIA output closely matches a rhodamine 6G fluorescence spectrum from the PhotoChem database (Two Rhodamine, PhotoChem tagged papers) and we observe that higher concentrations of rhodamine result in fluorescence peaks at higher wavelengths with increased distortions (Fig. 8B). We can attribute these distortions to a primary inner filter effect which was observed during data acquisition. In addition, we observe a discrepancy in the heights of the peaks (Fig. 8B). The 0.013mM peak was lower than the 0.132mM peak due to an observed secondary inner filter effect and the 1.322mM peak was the smallest due to significant observed primary inner filter effect. As expected, we also see a consistent phase convergence in the presence of signal and note that the convergence is stronger at lower Rhodamine concentrations (Fig. 8C,D,E). This is consistent with our previous observation that the spectra from lower Rhodamine concentrations better agreed to the reference spectrum.

7. Application 2: Imaging Simulation

Methods

SILIA is designed to work with signal inputs of arbitrary dimensionality. To demonstrate the potential application to analyze microscopy data, we used an experimental fluorescence microscopy image of liver cells (**Normal Chang Liver Cells**) (Fig. 9A,B,C). Lock-in amplification can be used on microscopy data to extract a fluorescence signal from noise (**Marriott 2008 live imaging in cell**) and we demonstrate the use of SILIA for such experiments.

To simulate a fluorescence signal, we multiplied the image by a square wave with a peak to peak height of 1 so it would periodically flash on and off. We took 500 total samples of this square wave at 10 samples per cycle. The microscope image was in a RGB format and had a size of 450x508 pixels which resulted in an input signal of 685800 channels. Each RGB value in the image can range from 0 to 255, where the values of (0, 0, 0) represents a black color and (255, 255, 255) represents a white color. We added a significant amount of Gaussian noise with a standard deviation of 75 to each pixel and ran the lock-in amplification on the image signal.

To demonstrate the applicability of this technique on an image with a higher brightness, we repeated the simulation using an image of a lightsaber taken from the Star Wars Wikipedia page. This image had a size of 298x444 pixels and was an RGB image, which resulted in an input signal of 396936 channels. We added the same Gaussian noise and used SILIA to recover the original image (Fig. 9 D,E,F).

Results

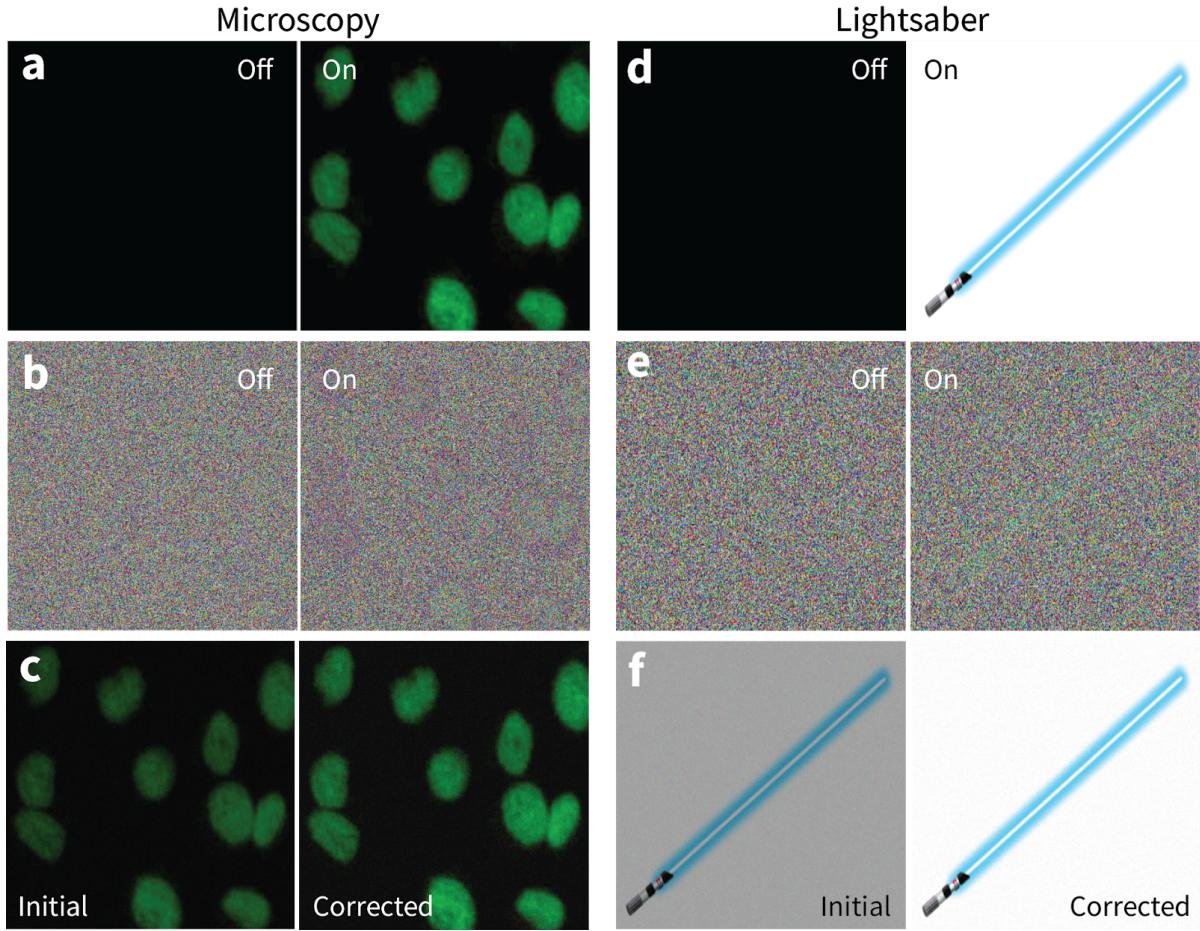


Fig 9: All panels have been scaled to a square aspect ratio. The left half is related to microscopy while the right simulation made use of a lightsaber photo with a white background on Wikipedia. (A) Simulated clean fluorescence signal of the sample using a live cell image ([Normal Chang Liver Cell](#)). The left panel shows the sample when it is not fluorescing and the right panel shows the maximum fluorescence signal. (B) Fluorescence signal with added noise. (C) SILIA recovered the original fluorescence image despite significant noise. The output for the corrected panel is scaled up by $\pi/2$ to compensate for diminished output when locking into a square wave signal. (D,E,F) We repeat the simulation in A, B, C but with an image of a lightsaber with a white background to show the broad applicability of this technique on images with varied color and brightness. The lightsaber is a metaphor for how SILIA “cuts” through noise.

We seek to recover the original microscopy image of the clean signal (Fig. 9A) from the noisy signal (Fig. 9B) by using SILIA. We observe that the cells are barely visible in the noisy signal (Fig. 9B). Using SILIA, we were able to accurately recover the colors, locations and shapes of the cells but with diminished output (Fig. 9C, Left). Since the input signals are square waves with a 50% duty cycle, we expect a SILIA output of $2/\pi$ times the original image (Fig. 5D) which results in a dimmer initial output (Fig. 9C, Left). To correct this error, we scale each RGB value by $\pi/2$ and cap the result so no value is above 255 (Fig. 9C, Right). We recover the live cell image (Fig. 9A, Right). We use a similar process for the lightsaber image, where we flash it at the reference frequency (Fig. 9D) and add significant noise (Fig. 9E). With just 500 samples of the signal at 10 samples/cycle, SILIA is able to recover the shape and contours of the lightsaber (Fig. 9F, Left) and by once again scaling the results up by $\pi/2$, we see an

accurate representation of the original lightsaber image. SILIA took about 8 minutes to compute this output on a laptop computer.

8. Discussion

Our software implementation provides a cheap and general alternative to multi-channel and multi-frequency lock-in amplifiers. The advantages of our implementation include the ability to estimate error in lock-in results and the simplicity and flexibility of SILIA when analyzing periodicity in datasets of arbitrary dimensionality. For example, the code can easily be customized to be conducive to a frequency sweep measurement ([Sonallion and Bonetto](#)) by fitting the reference to a frequency modulated waveform. Our software provides additional flexibility by working under a paradigm where the lock-in procedure is seen as a post-processing step. As a result, measurements can be cleaned and processed prior to performing the lock-in operation, and SILIA can be run multiple times on the same dataset or subsets of a dataset with adjusted input parameters, which is automatically done when computing error bars. For example, by artificially scaling the time axis of the reference prior to lock-in, harmonics of the reference frequency can also be locked into to recreate a Fourier series approximation to the exact signal waveform in each channel.

The limitations of using this implementation include the potentially significant runtime of the software when analyzing large datasets. However, there are methods that can significantly shorten the duration of the lock-in process, which we have not yet implemented for this paper. The software currently has a linear runtime with respect to the number of channels and frequency references (Fig. 4A, C), which can be drastically reduced through parallelization of SILIA with respect to each channel and frequency reference ([Sohal, Kaur 2016; Sah, Vaidya 2012; Parallel FFT docs](#)). However, we chose not to implement parallelization in SILIA since parallelization in Python did not significantly improve runtime for the datasets in this paper and we saw no reason to complicate the software by migrating to C or C++. As it is, SILIA can easily be modified for different specific tasks. In addition, given a prior knowledge of the number of samples being analyzed, faster FFT algorithms ([Parallel FFT docs; Ganapathiraju 1999; Murqi 2015; Amirkhattabi 2013; Pan 2007; Cerna 2000](#)) can be chosen. There are also superior techniques to our use of interpolation which can be implemented to mitigate error from the FFT due to unevenly spaced samples ([Greengard Nonuniform FFT; Ying Sparse Butterfly FFT; Emmanuel Butterfly Fourier integral; Barnett Non Uniform FFT](#)). SILIA is also limited in the size of the dataset it can process since it stores all the data in RAM to perform operations on it. However, this can be circumvented by adding methods to read and write to data storage formats such as csv or even SQL.

We have demonstrated and discussed SILIA's applications in spectroscopy ([ZI Impedance Spectroscopy; Sun Holmes 2007 microfluidic spectroscopy; Auret 1985 Lock-in DLTS;](#)) (Fig. 7, 8) and microscopy ([Fischer, Wilson 2016 Review Pump-Probe Microscopy; Pu Ting Dong Pump Probe Micrscopy Summary; Marriott 2008 live imaging in cell](#)) (Fig. 9), and mentioned its potential use in imaging technologies ([Liu, Shen, Ma 2016 Lock-in Camera; Probst, Jaquier 1993 Lock-in PPM resolution; Foix Lock-in ToF Camera](#)). SILIA may also be used to extract periodic features from video

data without a separate reference input by using a reference pixel ([Alinovi extraction of periodic features from video](#)) .

Conclusion

With our software implementation, we have demonstrated a device that can suppress background from periodic features in datasets with an arbitrary number of channels and frequency references. By adopting a paradigm where the lock-in amplification is seen as a post-processing step, we are able to provide significant amounts of freedom for researchers to apply this technique on various kinds of data while ensuring the lock-in does not hinder or complicate data acquisition. Our implementation is relatively straightforward to edit and we encourage users to improve and customize our code (<https://github.com/AMOS-experiment/SILIA>) to suit their needs.

Acknowledgements

This material is based upon work performed at the Lawrence Berkeley National Laboratory and was supported by the U.S. Department of Energy, Office of Science, Division of Chemical Sciences of the Office of Basic Energy Sciences under Contract DE-AC02-05CH11231.

Citations