Review of Scientific Instruments ARTICLE scitation.org/journal/rsi

Compact embedded device for lock-in measurements and experiment active control

Cite as: Rev. Sci. Instrum. 90, 023106 (2019); doi: 10.1063/1.5080345
Submitted: 7 November 2018 • Accepted: 1 February 2019 •
Published Online: 19 February 2019

Marcelo Alejandro Luda,[1,a)] Martin Drechsler,[2] Christian Tomás Schmiegelow,[2] and Jorge Codnia[1]

AFFILIATIONS

[1]CEILAP, CITEDEF, J. B. de La Salle 4397, 1603 Villa Martelli, Buenos Aires, Argentina
[2]Departamento de Física and IFIBA, FCEyN, UBA, Pabellón 1, Ciudad Universitaria, 1428 Buenos Aires, Argentina

[a)]mluda@citedef.gob.ar

ABSTRACT

We present a multi-purpose toolkit for digital processing, acquisition, and feedback control designed for physics labs. The kit provides in a compact device the functionalities of several instruments: function generator, oscilloscope, lock-in amplifier, proportional-integral-derivative filters, ramp scan generator, and a lock-control. The design combines field-programmable-gate-array processing and microprocessor programing to get precision, ease of use, and versatility. It can be remotely operated through the network with different levels of control: from simple off-the-shelf Web graphical user interface to remote script control or in-device programmed operation. Three example applications are presented in this work on laser spectroscopy and laser locking experiments. The examples include side-fringe locking, peak locking through lock-in demodulation, and complete in-device Pound–Drever–Hall modulation and demodulation at 31.25 MHz and advanced acquisition examples like real-time data streaming for remote storage.

I. INTRODUCTION

The demand for stabilization systems in optics, atomic, and molecular physics experiments is increasing more and more with time. The realization of high accuracy measurements relies often on the fine control of several variables at a time in the same experiment: light intensities, emission wavelength, temperature, modulation depth, mechanical stress, or position of elements are just some examples of variables that need to be controlled for that purpose.

The standard approach in control theory for a stabilization and control procedure is a feedback scheme.[1] This is accomplished by measuring certain variables of the system-under-control and acting on it with a set of control-parameters. The most common procedure is to measure outcomes, compare them to some preset desired values, and then act on the system with a chosen strategy to compensate the deviations. The realization of this strategy is the core of the control system, and its instrumentation has evolved from welded electronics analog circuits to more sophisticated and versatile digital processing systems. Probably, the most popular strategy used for control is the proportional-integral-derivative (PID) filter. Though it is known not to be the optimal strategy for all systems, its ease of use and understanding, as well as its satisfactory results makes it the usual workhorse in most practical setups and the one here chosen. Moreover, one can combine various PID filters to control multiple inputs and multiple outputs (MIMO) and add filters and signal processing to any stage, making the uses of PID control a versatile and intuitive platform for control.

The advent of fast microprocessor devices and field programmable gate arrays (FPGA) has paved the way to the implementation of the digital processing feedback schemes in experiment control setups. For example, PID filters were implemented on pure-FPGA boards.[2] Also pre-processing information like phase detectors was also accomplished with FPGAs.[3] A pure microprocessor implementation of the feedback controller has been showed to be more versatile and less complex in some simple or slow tasks.[4] Most of the experiences show the need to get a compromise between the fast and robust FPGA processing and the versatile and easily reprogrammable microprocessor global functions and control.

This way, the FPGA can handle time sensitive functions often also in parallel, while the microprocessor provides the user higher level functions and control in an on-site configurable manner.

Embedded devices with FPGAs and microprocessors were successfully used to achieve similar tasks as we show here.[5,6] However they were developed in the framework of proprietary, platform-dependent software that limits the available tools for developing custom solutions and modifications. An open-source system and an open-hardware solution have also been presented with a servo and loop-back control system in pure FPGA for an atomic, molecular, and optical (AMO) laboratory.[7] Finally, we note that another toolkit with open-source software and based on the same hardware we use has been presented in a recent study.[8] The project provides similar hardware features to the ones present in our work. The main difference is in the control environment graphical user interface (GUI) which is Python based instead of a web server as we use here.

In this work, we present a versatile general purpose toolkit for system control instrumentation built on a FPGA and microprocessor embedded device that is ready for simple off-the-shelf usage and also for high complex integrated schemes of acquisition and control. The hardware is based on a Red Pitaya STEMLab 125-14[9] board, which makes it a compact solution with a credit card size that can be acquired on web store. The FPGA and software design is all open-source and can be accessed through a web repository.[10] The toolkit includes an oscilloscope, a ramp/scan controller, two lock-in amplifiers, modulation generators, PID filters, and an overall control module which includes automatic locking, lock-watch, and re-lock. A big difference with previous implementations is the availability of a lock-in module which works with frequencies up to 31 MHz. All stages of the high frequency lock-in are implemented in the Red Pitaya module without the use of any extra components. We show an example of this feature for frequency stabilization of a laser using the Pound-Drever-Hall technique.

This device works as a stand-alone device that can be remotely controlled through Ethernet-TCP/IP network connection via a web-based interface and is therefore platform independent. Moreover, one can log into the embedded operating system via SSH protocol and set operation parameters through a communication bus to the FPGA, enabling different types of usage going from predefined easy configuration to low level user-defined programed control.

The present report is divided in two parts. In the first part, we describe the toolkit hardware, software, and usage. We describe the user strategy options and the toolkit design in Sec. II A, the architecture organized in layers in Sec. II B, and the usage logic organized in "instruments" in Sec. II C.

In the second part, the toolkit is tested in several example applications in an AMO physics laboratory. We show a rubidium absorption spectroscopy scheme in Sec. III A, with a Vertical-Cavity Surface-Emitting Laser (VCSEL), using the toolkit for acquisition of transmission signal and for side-fringe locking stabilization. Then we use an External Cavity Diode Laser (ECDL) to implement a saturated absorption spectroscopy scheme, in Sec. III B, using lock-in phase sensitive acquisition to lock laser emission frequency to spectral peak maximum. Finally, we present a Pound–Drever–Hall locking scheme for another ECDL, in Sec. III C. One of the goals of this proposal is the lock-in operation at 31 MHz, enabling the implementation of the complete lock-in modulation and demodulation scheme in only one device, something which has never been reported so far.

II. SYSTEM DESIGN

The toolkit is based on a commercial system on chip (SOC) device known as Red Pitaya v1.1 (STEMLab 125-14) with a Xilinx Zynq 7010 integrated circuit core, which includes an FPGA and a dual core ARM Cortex-A9 processor, integrated to peripherals suitable for signal acquisition and generation (fast ADCs, DACs, DIOs, and communication ports). The device operation follows a client-server architecture that can be thought as a layer structure. The client side is the user front-end running on a client's own device. The server side is the STEMLab device, including the operating system, programmable electronics, and hardware peripherals. The layers can be controlled on different levels depending on the user's requirements. In the following, we describe possible application strategies and the layer architecture, both depicted in Fig. 1.

A. User strategy

We present here three different user strategies with growing complexity and versatility. Depending on the needs, a user can choose from simple pre-defined working modes to modifying instrument cabling or programming complex measurement and feedback routines.

The "GUI predefined operation" option allows us to perform most of the tasks described here from a web browser in the user device. The front-end is part of the client-side and is built over the HTML+JS page and presents a friendly interface for several instruments' operation: two lock-in amplifiers, two PID loops with different locking and re-locking routines, an on-screen oscilloscope, and cabling between them. These allow us to perform a wide variety of measurement and control tasks which we describe in detail in Sec. III, with several examples which include low and high frequency lock-in measurements and active stabilization of lasers.

An even greater versatility can be achieved via the "Programed and remote acquisition and control" option. All the instruments can be remote controlled and data can be acquired through user commands, enabling the possibility of incorporating them to algorithms designed by the user in various programming languages. An example code can be found in the project repository[10] for Python and Matlab, including two channel oscilloscope acquisition, on-demand multichannel reading of several signal values and instruments' operation by writing control register values.

The user commands are Python scripts executed in the Red Pitaya shell that implement the basic procedures for reading and writing RAM addresses' values linked to FPGA registers. The remote execution can be performed by any

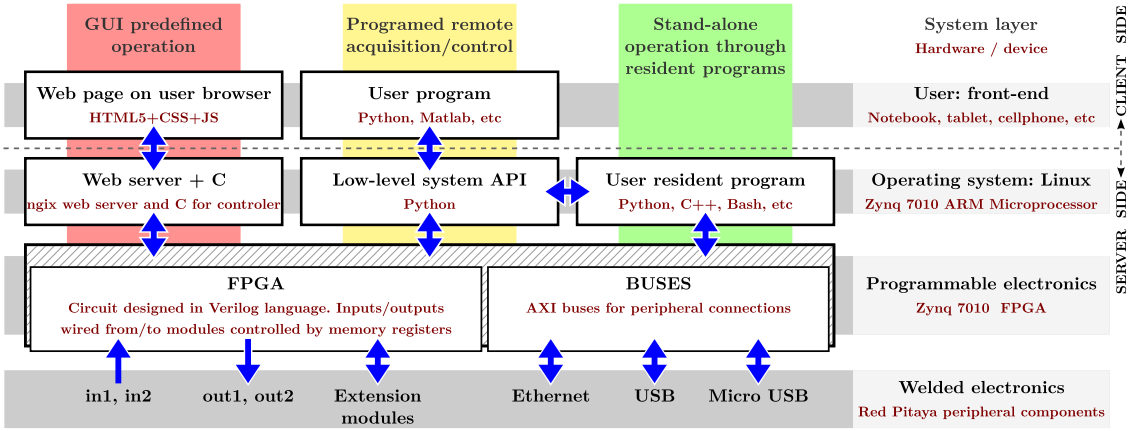

FIG. 1. Design of the client-server architecture. The toolkit is organized in a layer structured way. The elements are grouped horizontally by site and type. As indicated in the right-hand-side column, system layers are organized in increasing levels of complexity: welded electronics, programmable electronics, operating system, and front end. The dashed line separates the client part (end-user device) and the server part (STEMLab device). The horizontal and vertical blue arrows show communication between elements. There are three possible user strategies indicated by colored columns (red, yellow, and green). The first (red) column follows the Red Pitaya STEMLab original design philosophy. Here operation is controlled from the graphical user interface (GUI). The next two columns show more advanced operation where the system is remotely controlled (yellow) or where resident programs do automated acquisition and signal processing (green).

remote-shell tool, using serial communication (microUSB) or Ethernet (RJ45 port). The examples provided online[10] use SSH protocol over a TCP/IP network, being a secure and versatile option that can be incorporated to almost any already working computer network. Moreover, open-source implementations of SSH protocol are available for all possible operating systems of the client, enabling the operation from any device or programing language.

The on-demand read and write procedures are only limited by network communication delays and microprocessor process priority. The provided examples include combined operation of instruments for tasks like ramp/scan configuration, lock-in acquisition of system response, PID configuration, and launching a triggered controlled stabilization scheme.

We also provide an API (Application Program Interface) for Python to simplify code writing of user designed "resident programs," running in the server side. This enables the operation of instruments with lower latency and enables the design of algorithms for fast decision making, faster multichannel acquisition, and stand-alone working without the client side. In this way, data acquisition and instrument commands can be made with a latency of ~ms order.

The "resident programs" advantages can be exploited to implement different integration strategies of the toolkit in the lab. For example, acquired data can be pre-processed on the server-side and stored in the STEMLab device for later user retrieval. On the other hand, the raw or the pre-processed data can be streamed to the client side for online monitoring and storage. The user can choose these or other options, depending on whatever suits the experimental scheme better. In Subsection III C 1, we report an example of long accumulated measurements streamed to the client-side, used for Allan-deviation calculation.

B. Layer architecture

The system design can be understood in a 4 layer scheme (see Fig. 1). The lower three layers are part of the server side and take place in different parts of STEMLab board hardware.

The lowest one is the "Welded electronics layer," which provides the connection to digital and analog input/output ports and the access to peripherals, including some analog electronics for signal conditioning. Signals in this layer are converted from continuous analog voltage signals on wires to digital clocked buses of data, suitable for FPGA reading. For this purpose, the board includes two fast Analog to Digital Converters (ADCs), associated with in1 and in2 FPGA buses, and two fast Digital to Analog Converters (DACs), associated with out1 and out2 FPGA buses, working at 125 MSa/s with 14 bits of resolution, on the range of ±1 V. ADC inputs can work in the range of ±20 V with hardware jumper configuration. Also, 16 input/output digital pins are included, with a maximum refresh rate of 125 MSa/s, on a LVCMOS 3.3 V logic. It also includes four slow outputs of 0–1.8 V, 1 MSa/s built from low pass filtered PWM (Pulse-Width Modulation) signals and four slow inputs with the same sample rate.

The "Programmable electronics layer" stores the digital electronic FPGA circuits for real-time processing, like filters, mixers, and adders, which build up the core of each instrument implementation. In this layer, the signal data flow through digital buses that represent signed integer values of 14 bit resolution. The circuits were designed in the Verilog language, synthesized using Xilinx Vivado 2005.2 software and implemented on the Zynq-7010 FPGA chip on the Red Pitaya board. The Xilinx development tools for this chip are available for free, so no additional costs are incorporated on development licenses if the design should be modified. This layer also provides connectivity between the

microprocessor and "Welded electronics layer" through specialized data buses.

The FPGA design was made with simplicity in mind, to ease the user understanding of the electronic logic. Also, the implementation was made trying to prioritize direct wire processing, reducing the usage of registers in the middle of input-output signal flow. This decision was taken because each register adds a clock period delay (8 ns) to the data flow. Closed-loop control schemes are bandwidth limited by this delay value.[1] The achieved delay for device input-output using one PID filter was 130 ns, which imposes a theoretical limit to the feedback bandwidth of 3.8 MHz.

The "Operating system layer" stores the back-end logic that controls the operation of the digital circuits. It runs a GNU/Linux operating system with a set of RAM memory addresses mapped to FPGA registers that are used by the instruments' circuits to set their configurations or store data. The back-end logic reads and writes these registers and makes some data conditioning, like conversion from FPGA integer-raw data to end-user float-voltage data values. This logic is implemented in the API (programmed in Python) and also in the Nginx web server (programmed in a C extension module). The web server provides the client access and control from the user web browser, exchanging data using the JSON standard format and HTTP/POST protocol. The API is used by the user commands introduced in Sec. II A.

The "User Front-end" layer is only provided for the "GUI predefined operation" strategy, and it consists of a dynamic web page loaded in the client device. The data acquired in the FPGA layer, and conditioned in the back-end layer, are shown here in an intuitive way, as can be seen in Fig. 6.

C. The instruments

The functionalities of the toolkit have been organized in logical units called "instruments," which allow the user to interact with it. Each instrument provides some of the functionalities of typical laboratory equipment used for acquisition and control. The instruments implemented are the following:

1. Two lock-in amplifiers.
2. Two Proportional-Integral-Derivative (PID) controllers.
3. A ramp/scan function generator.
4. A general lock-control helper, with an event monitor and re-lock routine.
5. An oscilloscope.

Each instrument is composed by a set of HTML controls in the Web GUI, grouped in panels, and a set of Verilog modules for the FPGA layer. Instruments 2 and 5 are modified versions of the open-source applications developed by Red Pitaya community and released with a BSD license.[11] Instruments 1, 3, and 4 were developed for this work.

In the FPGA layer, the instruments are composed by logical modules whose behavior is controlled and monitored by registers. The modules are independent interconnected circuits which handle signal acquisition, processing, and generation. They also include internal memory which allows implementing filters and control loops. The base modules include function generators, ramp generators, multipliers, demodulators, low-pass filters, multiplexers, etc.

The control registers can be set and read from the Red Pitaya local shell, from remote software, or from the application Web GUI. The last one includes user friendly options for operation, described in more detail in Sec. II D.

A set of buses and multiplexers allow the interconnection of the different instruments. The buses transport the ADC input signals and the instruments' output signals. The multiplexers are controlled by the user through register values to choose the instruments' inputs or the DAC output signals. In this way, several instruments can be combined into a system with dedicated purpose, such as lock-in demodulation and PID control for stabilization schemes.

The core design of the instruments is described in this section, and usage information is documented in the project web page.[12]

1. The lock-in amplifiers

Two lock-in amplifiers were implemented to cover different kinds of applications: a "standard" harmonic lock-in, with a frequency range that goes from 3 Hz to 49.6 kHz, and a square wave lock-in with a wider frequency range, from 30 mHz to 31 MHz.

Figure 2 shows the scheme of the FPGA implementation logic of a lock-in demodulation path. The input signal signal_i is multiplied by a local oscillator ref_i signal that settles the frequency and phase reference and then is filtered by using a low pass filter (LPF) with a cut-off frequency of fcut_i. The LPF output out27_i is a 27 bit signed int register which is available for recoding measurements and which can be monitored via the Web GUI. A bus-trim applied to the register reduces it to a 14 bit signal out14_i. The selection of the trimmed bits has the net effect of an amplifier by powers of 2, controlled by the amp_i parameter. This 14 bit signal can then be connected to either the PIDs, the output DACs, or the oscilloscope input channels.

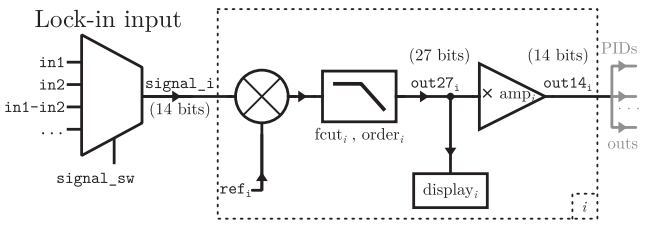

FIG. 2. Lock-in schematic design. Lock-in demodulation, filtering, and amplification are performed digitally at the FPGA layer. The 14-bit input is digitally selected by using a multiplexer with the control parameter signal_sw. The input is then multiplied with a reference signal ref_i and filtered with a low pass filter. The order of the filter and the cutoff frequency are selected with the parameters order_i and fcut_i. After the filter, the signal has 27 bits and is accessible through the variable out27_i. Finally the signal is amplified by the factor 2^{amp_i} and trimmed to 14 bits to generate the output signal out14_i. This process occurs in parallel for 7 different reference signals indicated by the letter i and detailed in Table I.

TABLE I. Register name reference for lock-in demodulation paths. The columns represent the control parameters depicted in Fig. 2 with generic names. Each row shows the actual register names for each demodulation path. The 1–5 paths are part of the harmonic lock-in, and the 6–8 paths are part of the square wave lock-in. Some registers are shared between several demodulation paths.

i	ref_i	Signal equation	$fcut_i$ name	$order_i$ name	amp_i name	$out27_i$ name	$out14_i$ name
1	cos_ref	$\cos(2\pi ft)$	lpf_F1_tau	lpf_F1_order	sg_amp1	X	Xo
2	sin_ref	$\sin(2\pi ft)$	lpf_F1_tau	lpf_F1_order	sg_amp1	Y	Yo
3	cos1f	$\cos(2\pi ft - \phi)$	lpf_F1_tau	lpf_F1_order	sg_amp1	F1	F1o
4	cos2f	$\cos(2\pi 2ft - 2\phi)$	lpf_F2_tau	lpf_F2_order	sg_amp2	F2	F2o
5	cos3f	$\cos(2\pi 3ft - 3\phi)$	lpf_F3_tau	lpf_F3_order	sg_amp3	F3	F3o
6	sq_ref	$\mathrm{sgn}(\cos(2\pi f_{sq}t))$	lpf_sq_tau	lpf_sq_order	sg_amp_sq	sqX	sqXo
7	sq_quad	$\mathrm{sgn}(\sin(2\pi f_{sq}t))$	lpf_sq_tau	lpf_sq_order	sg_amp_sq	sqY	sqYo
8	sq_phas	$\mathrm{sgn}(\cos(2\pi f_{sq}t - \varphi))$	lpf_sq_tau	lpf_sq_order	sg_amp_sq	sqF	sqFo

The harmonic lock-in is composed of five demodulation paths like that depicted in Fig. 2. Each path has its own ref_i signal and bus names, as shown in Table I, which can be used in the application for DAC outputs or oscilloscope visualization. The cos_ref and sin_ref paths have their reference in quadrature, so the X and Y outputs provide the full phase and amplitude information of the signal_i filtered frequency component. The other three paths allow setting a fixed phase relation with respect to cos_ref and also obtaining information about the first ($2f$) and second ($3f$) harmonics. The ϕ parameter, which can be configured through the Web GUI phase control, sets a phase displacement of $\phi = 2\pi \frac{\mathrm{phase}}{2520}$.

Each local oscillator signal is made from 2520 signed integer values proportional to a sine period. They are stored in a memory module implemented as various lookup tables. The reading addresses for the memory modules are set by counter modules, driven by a configurable clock divider that allows us to change the reading sample rate and, with it, the resulting ref_i working frequency. The cos_ref, sin_ref, cos2f, and cos3f signals were designed to satisfy the discrete Fourier orthogonality relations of Eq. (1), which avoid offsets generated by digital rounding, and are important not only for measurement precision but also useful for reducing instrumentation induced instabilities in a feedback stabilization scheme. This condition is not satisfied if we use signals built from real valued cos functions discretized with global criteria applied to all the values (like using standard integer conversion functions: floor, ceil, or round),

$$\sum_{i=0}^{2519} \mathtt{f[i]} \cdot \mathtt{g[i]} = 0 \quad \text{for } \mathtt{f} \neq \mathtt{g} \quad \text{and } \mathtt{f}, \mathtt{g} = \begin{bmatrix} \mathtt{cos} \\ \mathtt{sin} \\ \mathtt{cos2f} \\ \mathtt{cos3f} \end{bmatrix}. \tag{1}$$

The square wave lock-in is composed of three demodulation paths: 6-8 from Table I. Two local oscillators in quadrature (sq_ref and sq_quad) allow the acquisition of the complete quadrature information in the sqX and sqY registers. Also, another oscillator path, sq_phas, with a configurable phase relation is available, with the output value stored in sqF.

The square wave signals are built on run-time, switching a bit that represents the ±1 values. The working frequency f_{sq} is set by defining the semi-period time length, counting steps of the base FPGA clock. In this way, the period can be set with a resolution of 16 ns, starting from 32 ns (31.25 MHz) and expanding the possible values up to 68 s. The φ phase relation is set as an integer factor of 8 ns. Both parameters, period and phase, are set by 32 bit unsigned integer registers. The maximum frequency is ultimately limited by the clock period of 8 ns. The lower limit could be increased, if needed, by expanding the size of the registers. The multipliers of these paths are modified to map the binary input from 0, 1 values to ±1. The sq_ref binary signal is available in one of the fast binary outputs of the extension pins of the STEMLab device, and all the signals can be used on the DAC outputs, where they are mapped from the binary values to ±0.5 V.

The out27 register enables lock-in measurements with a full resolution of 27 bits and a sensitivity of 59.6 μV. The sensitivity can be enhanced with pre-amplification, as is shown in the experience described in Sec. III B.

2. PID controllers

Two PID modules were implemented for use in feedback stabilization schemes. The circuit design is based on the PID included in the free software applications of the Red Pitaya community.[11] They were modified to extend the working parameter range over several orders of magnitude, by incorporating scale registers. The output control was enhanced by adding features like output value freezing and integrator memory freezing, useful for re-lock routines (see Sec. II C 4).

Both modules have a common error signal as default input, which can be selected from different input buses, as is shown in Fig. 3. The input can be shifted by a user-controlled error_offset value that can be interpreted as a working setpoint. Also, independent input signals for each module can be chosen (not shown in the figure).

Each module implements three filters, proportional, integral, and derivative, which sums up to build the output signal pidX_out (X = A, B), as it is shown in Fig. 3. The summed result is stored in a register which allows us to freeze the value at command.

The behavior of the three filters depends on the value of three parameters: k_p, $k_i = {}^1/_{\tau_i}$, and $k_d = \tau_d$. Two registers allow the user to control each of them: one 14 bit signed int value changes the parameter linearly, while the second value allows

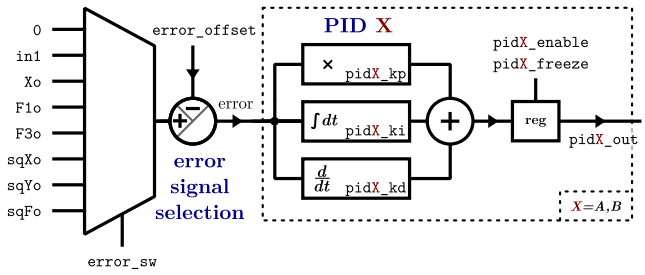

FIG. 3. Proportional, integral, derivative (PID) module. This module is used for feedback/control loops. It produces an output which is a functional of a selectable input and configurable parameters. The desired input is selected with a multiplexer via the control signal error_sw. Next, the value error_offset (sometimes also called the set-point) is subtracted from the input signal to produce the error signal. The action of the PID filter is controlled by three submodules as described in the main text. At the output stage, two flow control parameters pidX_enable and pidX_freeze allow enabling/disabling (zeroing) or freezing the output. There are two independent PID modules running in parallel labeled A and B. Alternatively different inputs can be selected for each PID but is not included in this graphic (see the main text).

us to change the order of magnitude of the parameter, working as a scale control. For example, the k_p parameter is controlled by the pidX_kp register value, and by the scale factor set by the register pidX_PSR, which allows us to choose one of the predefined values for n_p in Eq. (2a). The same logic is applied to control k_i and k_d using pidX_ki, pidX_kd and pidX_ISR, pidX_DSR, respectively, as is shown in Eqs. (2b) and (2c),

$$k_p = \frac{\texttt{pidX\_kp}}{2^{n_p}} \qquad \text{with } n_p = \{0,3,6,10,12\}, \tag{2a}$$

$$\tau_i = \frac{2^{n_i}\cdot 8\,\text{ns}}{\texttt{pidX\_ki}} \qquad \text{with } n_i = \{0,3,6,10,13,16,20,23,26,30\}, \tag{2b}$$

$$\tau_d = \frac{2^{n_d}\cdot 8\,\text{ns}}{\texttt{pidX\_kd}}\cdot\frac{60}{64} \qquad \text{with } n_d = \{0,3,6,10,13,16\}. \tag{2c}$$

The linear coefficient register lets the user have fine control over the PID parameters, while the power of two factor, implemented efficiently through shift registers, expand the parameter scope over several orders of magnitude. This dual scale allows controlling systems with time constants ranging from a few μs to many seconds.

The derivative module of the PIDs was completely rewritten to avoid high frequency noise amplification. The new design ensures that spikes and frequency components whose characteristic times are below the order of magnitude settled by the n_d value will not induce undesired perturbations that can make the feedback scheme unstable. The implemented design includes a down-sampling procedure and a slope calculation sub-module, called slope9. The PID input signal is down-sampled taking the mean value of 2^{n_d} consecutive data samples, so the processed signal that feeds the slope9 submodule has a ladder shape with a $T_d = 9\cdot 2^{n_d}\cdot 8$ns step time length. The net effect of this procedure is similar to the application of a low pass filter with a time constant of T_d before the derivative calculation. The slope9 submodule calculates the signal derivative by taking the slope of a linear square least regression of the last nine step values of the down-sampled signal.

The parameters k_p, τ_i, and τ_d can be used for prediction of an ideal PID response to an $e(t)$ input signal using Eq. (3a). Equation (3b) provides a more accurate prediction of the implemented PID response in clock steps of 8 ns. The "slope9out" function represents the slope9 submodule output signal that can be simulated by an algorithm following the procedure described above,

$$\texttt{pid}_{\text{out}}(t) = k_p\cdot e(t) + \frac{1}{\tau_i}\cdot\int_0^t e(t')\,dt' + \tau_d\cdot\frac{d}{dt}e(t), \tag{3a}$$

$$\texttt{pid}_{\text{out}}[n+1] = k_p\cdot e[n] + \frac{8\,\text{ns}}{\tau_i}\cdot\sum_{i=0}^{n} e[i] + \frac{\tau_d}{8\,\text{ns}}\cdot\text{slope9out}(e[n,\ldots,n-9\cdot 2^{n_d}+1]). \tag{3b}$$

3. Ramp function generator for scanning

A triangle wave-shape generator was implemented which is useful to control two parameters at the same time. This module outputs two synchronized signals called ramp_A and ramp_B.

The behavior of ramp_A is controlled by setting the range and the time base. The triangle wave-shape is generated by increasing/decreasing the output by 1 int at each time step T until it reaches one of the limits and then changing the slope sign (see Fig. 4). The registers ramp_hig_lim and ramp_low_lim control the top and bottom limits. The register ramp_step sets the time step with the rule T = ramp_step · 8 ns. The behavior of ramp_B is linked to the first one: it is generated by multiplying ramp_A by the factor ramp_B_factor/4096.

The chosen control parameters allow the user to set the ramp by its slope and limits. This is important since an acquired signal waveform may change with sweeping speed. These changes are a natural consequence of bandwidth limits present in all physical systems. By maintaining a constant slope, changing the range does not affect the measured

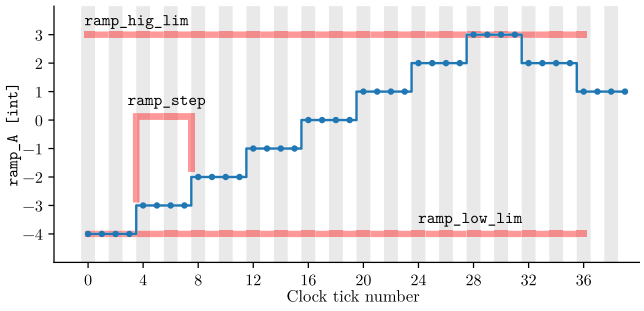

FIG. 4. Output of the ramp function generator as a function of clock tick. The triangle-wave output ramp_A signal is controlled by the following parameters: ramp_hig_lim, ramp_low_lim, ramp_step, as shown in the figure and described in the main text. The clock runs with a period of 8 ns.

waveforms. This design has been proven to be helpful when using the ramp instrument for optical spectroscopy.

The ramp generator has extra functionality: it can be started and stopped at any time with the ramp_enable register, it can be reset to the ramp_A = 0 value with the ramp_reset register, and its starting slope sign can be set with the ramp_direction register. Also, as will be discussed later, the lock-control module can take control of the start/stop command and the ramp limits for locking and re-locking procedures.

4. Lock-control

A device meant for locking to a given error signal requires switching between different behaviors: scanning, locking, re-locking on events, etc. Changing from one state to the other requires timed interaction between the previous modules. This interaction is handled by the lock-control module. Figure 5 shows the schematic inter-cabling of the modules.

A typical example of this behavior is what we call *Trigger Lock*. Here the system switches from a scan-mode, where one can see the error signal by sweeping the control parameter, to a lock-mode where the scan is turned off and the PIDs are turned on to stabilize the error signal. That is, on a trigger event, the ramp's outputs are frozen and the PID's outputs are enabled. The trigger can be set to a given value of the ramp period (*time trigger*), a given value of the signal (*level trigger*), or the fulfillment of both conditions (*level + time trigger*). The values for these triggers can be selected graphically on the oscilloscope screen or manually. Alternatively one could set lock-control parameters using user defined algorithms to automatically scan a signal and lock to given peak.

The module also provides a *Re-lock* routine which tries to automatically re-lock on eligible events. This submodule will consider that the system is out-of-lock when the absolute value of the error signal exceeds a set value or when one of the inputs (in1, in2, in1 – in2, or any lock-in demodulated signals) drops below a set threshold. When any of these conditions is met the re-locking routine is started. The re-locking procedure was inspired on a recent study[7] and consists on freezing the PIDs and starting a ramp scan increasing the scan limits on a two factor in each half period. If during scanning the system reaches a lock condition, the submodule stops the scan and turns on the PIDs again. If the lock condition is never met, the submodule will continue increasing the scan amplitude until it makes a semi-period sweep with the longest width (1 V ≡ 8192 int) and then will stop ramp.

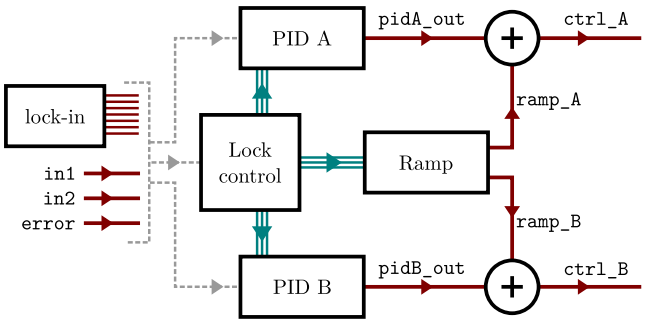

FIG. 5. Lock-control data flow. The lock-control runs synchronized actions on the PID and Ramp instruments. The red lines are signal data buses, and the turquoise lines represent sets of buses used to control timing and operation of the modules. Gray dashed lines are selectable data buses. These include all the lock-in outputs and the ADC inputs. Any of them can be routed to the PIDs for signal processing or the lock-control for event identification and triggering.

Finally, a submodule which carries out a *Step Response Measurement* allows the user to evaluate the system response to an abrupt change in the control value. This information can then be used to calibrate the parameters of the PID modules.

5. Oscilloscope

The oscilloscope instrument is a modified version of Scope Application from Free Software Red Pitaya developers community.[11] It is composed of two memory arrays with a length of 16 384 for storing 14 bit signed int values, modules for triggering, decimation control, and anti-aliasing filters, and it allows us to make acquisitions on the FPGA clock sample rate (125 MSa/s). Here we extended the functionality of the basic oscilloscope to allow the display of various internal and external signals. These include the ADC inputs (in1, in2), instrument outputs (PIDs and ramp), important link buses (error, ctrl_A, ctrl_B), and all the lock-in path outputs and local oscillators. With this extension, the oscilloscope is useful not only for external data acquisition but also for internal data flow and processing inspection, essential for debugging and tuning the parameters of each module.

Also, some acquisition options were added. We included the possibility to disable the anti-aliasing FPGA module filter and to incorporate more triggers from useful signals. The external trigger is extended for the user selectable event list, including out-of-lock, triggered lock-start, ramp_A at limit reach, and lock-in oscillator period start. Also, some web GUI options were added, like an R-ϕ (absolute and phase) runtime calculation option for lock-in X, Y and sqX, sqY visualization and switch between volts and int units. By default, the acquired curve values are expressed in volts, using the ADC/DAC conversion factor (signed 14 bits int resolution): 8192 int ≡ 1 V. This can be switched to raw int values.

D. The Web GUI

The Web GUI frontend design is also based on free software Scope application.[11] It provides the off-the-shelf functionality shown in the first column of Fig. 1. It is structured in columns with dropdown panels that group configuration settings by functionality. We improved the interface adding several useful functions, like data save/load, configuration save/load, and stop button.

A left column was added for placing the instrument panels designed for this work, and two bottom panels for the lock-in amplifier output visualization (Fig. 6). The instrument controls were designed with the philosophy of keeping simplicity without compromising the low-level accuracy. Most of the controls use integer numbers for input data type, which allows the user to define the precise value of the FPGA registers that

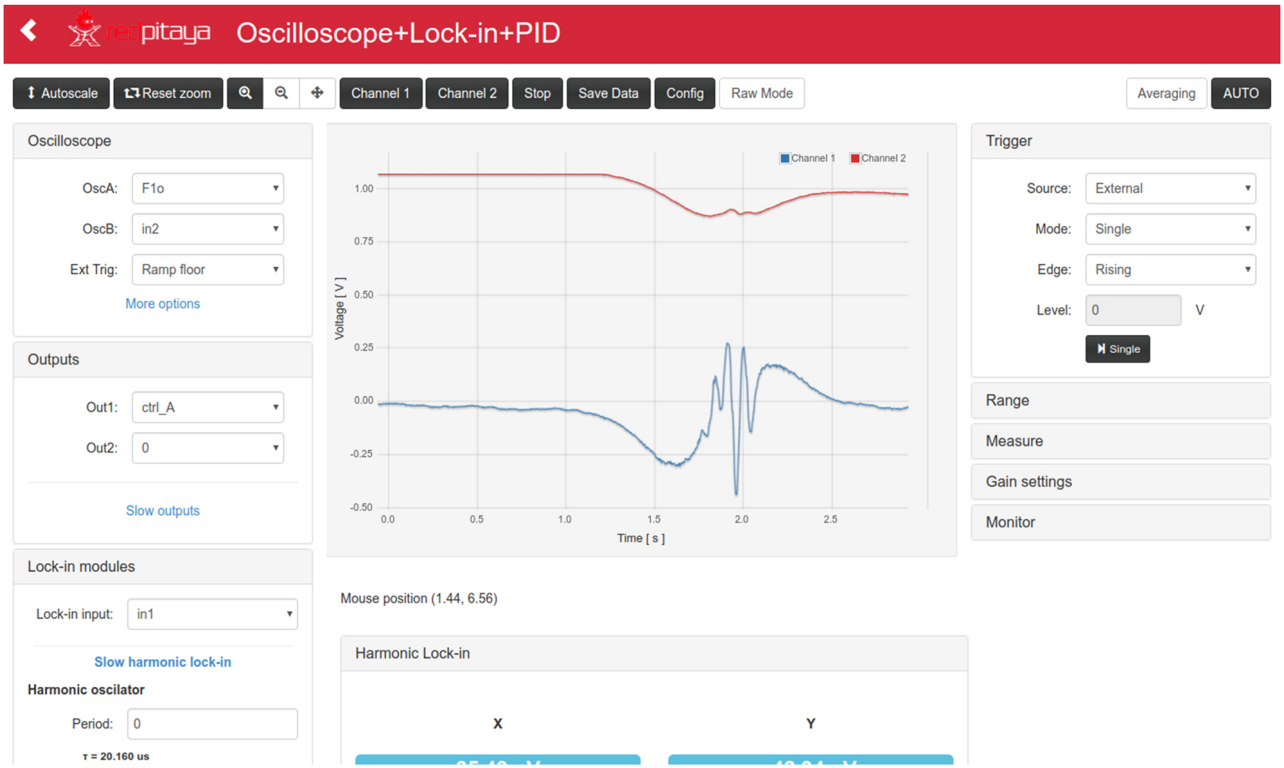

FIG. 6. Web page GUI front-end showing a rubidium absorption and saturated absorption signal. The display is organized in three columns with sub-panels. The left one contains one panel per instrument containing their configuration options. The right one contains the oscilloscope visualization options. The middle one contains the oscilloscope screen and a bottom panel showing the lock-in output values.

the circuits will use and includes text displays that translates these values to the physical magnitudes involved (i.e., seconds and volts).

The oscilloscope screen is in the middle of the page. It displays two channels for plotting user selected signals. In the sample applications discussed below, in Sec. III, one channel shows the spectral response of the system, while the other displays the error signal. This provides a fast way to view and evaluate the performance of the system on lock, entering a lock, and re-locking.

III. EXAMPLE APPLICATIONS

We report a set of experimental applications of the toolkit presented in order of complexity to show the instruments' usability and potential. In the first one (Sec. III A), we introduce the usage of the ramp and the oscilloscope to take the absorption spectrum of an atomic vapor, with a simple scheme of one control parameter and one measured variable. Also, we implement a side-of-fringe locking scheme using a PID controller and the lock control instrument. In the second application (Sec. III B), we add the harmonic lock-in amplifier to the measure and stabilization scheme, in a saturated absorption experiment. We used two output ports and measure two input variables, in a multi-input-output control system but with only one control parameter. In the third one (Sec. III C), we show the square lock-in amplifier working at 31 MHz for a Pound-Drever-Hall stabilization application. Some advanced capabilities are shown here, like multiple controlled variables, in-device programing for special measurements, and different hardware implementations of the same stabilization scheme. Finally, we discuss some potential applications of the toolkit.

A. Absorption spectroscopy: Ramp, oscilloscope, and PIDs

We tested the off-the-shelf capabilities of the toolkit in an atomic vapor absorption spectroscopy experiment with rubidium. A tunable laser that goes across an atomic vapor cell is used to make an optic-frequency scan around one of the atomic electronic transitions. A photodiode is used to measure the intensity decay at the cell output (Fig. 7), as a result of the absorption on transition resonance. We used a VCSEL, what ensures mode-hop-free scanning using only one control parameter: the diode current.[13] The laser wavelength is centered in 795 nm, suitable for rubidium D_1 line ($5^2S_{1/2} \rightarrow 5^2P_{1/2}$) spectroscopy.

The ramp instrument was used to make the frequency scan, by performing a voltage-controlled sweep of the current

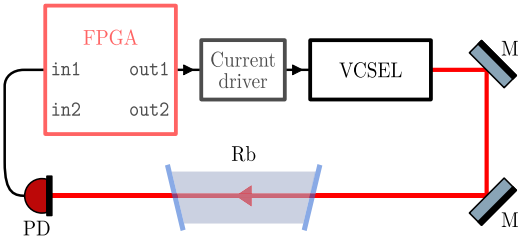

FIG. 7. Scheme of the rubidium absorption spectroscopy experiment. A vertical-cavity surface-emitting laser (VCSEL) is steered with two mirrors (M) through a rubidium gas cell (Rb), and the transmitted signal is detected in a photodiode (PD). The signal from the photodiode is read by the ADC on the Red Pitaya board. The laser's frequency is controlled with output DAC out1 by changing the set-point on a current driver.

driver. The oscilloscope instrument was used for photodiode signal acquisition.

The ramp was configured to make a scan of ~1 V through out1 DAC at 7 Hz. In Fig. 8, the transmitted signal measured by using the photodiode is shown with a blue line. The base slope of the curve is related to the laser power increment along the scan because of the current variation. The four dips in the curve are the absorption lines for each of the hyperfine-split transitions.

This experiment was used to test the most simple stabilization scheme: side-fringe locking. The variable to stabilize is the transmitted signal, measured from the in1 port. The PID instrument was used to make the feedback response, and the lock control instrument was used to control the loop-close event.

The locking set-point was configured on error_offset = 2620 int ≅ 0.32 V to lock the side of the last spectrum dip in the ramp scan, making error = in1 – 2620. To prevent the lock on the first crossover of error_offset, the lock-control instrument was configured to close the loop on *level + time* trigger. The loop-close procedures were configured to stop the ramp scan and enable the pidA_out. The PID parameters were set to τ_i[ki = 600] ≅ 895 μs and k_p[kp = 1024] = 1.

The result after reaching the lock-control *level + time* trigger condition is shown in Fig. 8 (orange line). The transmitted signal remained in the configured set-point level with a standard error of σ = 1.85(3) mV.

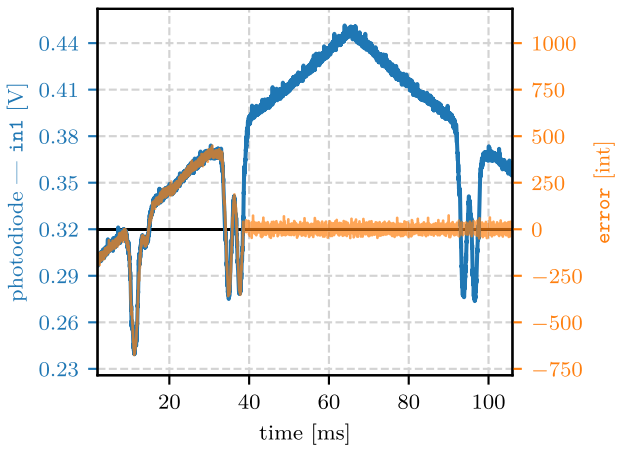

FIG. 8. Absorption spectroscopy of rubidium and laser lock as obtained with the experimental setup shown in Fig. 7. The transmitted intensity across the rubidium cell is plotted for a ramp scan (blue line) and for a lock-start event (orange line). The curve dips correspond mainly to ^{87}Rb hyperfine absorption lines. The triangular shape of the measured signal is due to the frequency-intensity correlation which is typical of VCSEL diodes.

B. Saturated absorption spectroscopy: Harmonic lock-in

In this experience, we extended the previous setup to make a saturation absorption spectroscopy[14] scheme, which allows us to measure transitions with resolution exceeding the limit posed by Doppler broadening.[15] This scheme uses a pump and probe configuration, implemented here with the same beam reflected in a mirror, as shown in Fig. 9. The pump saturates the transition and induces transparency, sensed by the probe as a transmission peak. This technique enables the possibility of laser frequency locking to absolute references with higher accuracy.

We used an ECDL at 780 nm wavelength, suitable for rubidium D_2 line ($5^2S_{1/2} \rightarrow 5^2P_{3/2}$) spectroscopy.

In this kind of laser, the diode current and the position of the diffraction grating are controlled to tune the laser. The position of the grating is controlled with a piezoelectric (PZT) element. If both parameters are controlled simultaneously, a mode-hop-free tuning range of several GHz can be achieved.[16] Figure 10 shows an example laser frequency scan (blue line) around the transitions $F_g = 3 \rightarrow F_e = 2, 3, 4$ for an ^{85}Rb isotope. In the figure, the optical frequency increases

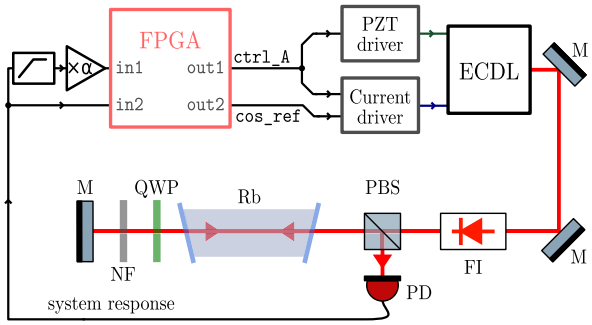

FIG. 9. Scheme of saturated absorption spectroscopy of rubidium. The emission wavelength of an external cavity diode laser (ECDL) is controlled with two parameters: a voltage on a piezoelectric element (PZT) and the diode current. Moreover, the laser current can be controlled through two independent signals. One is used for tuning, and the other is used to generate a modulation in the frequency. The laser beam is steered with two mirrors (M) through a Faraday isolator (FI), a polarizing beam splitter (PBS), and a rubidium cell. Following a quarter wave plate (QWP), a neutral density filter (NF) and a mirror reflect the beam back but with opposite polarization. The beam now reflects on the PBS and is detected on a photodiode (PD). The measured signal is read through the two ADCs on the Red Pitaya board. One of them is previously DC-decoupled and amplified with an analog circuit.

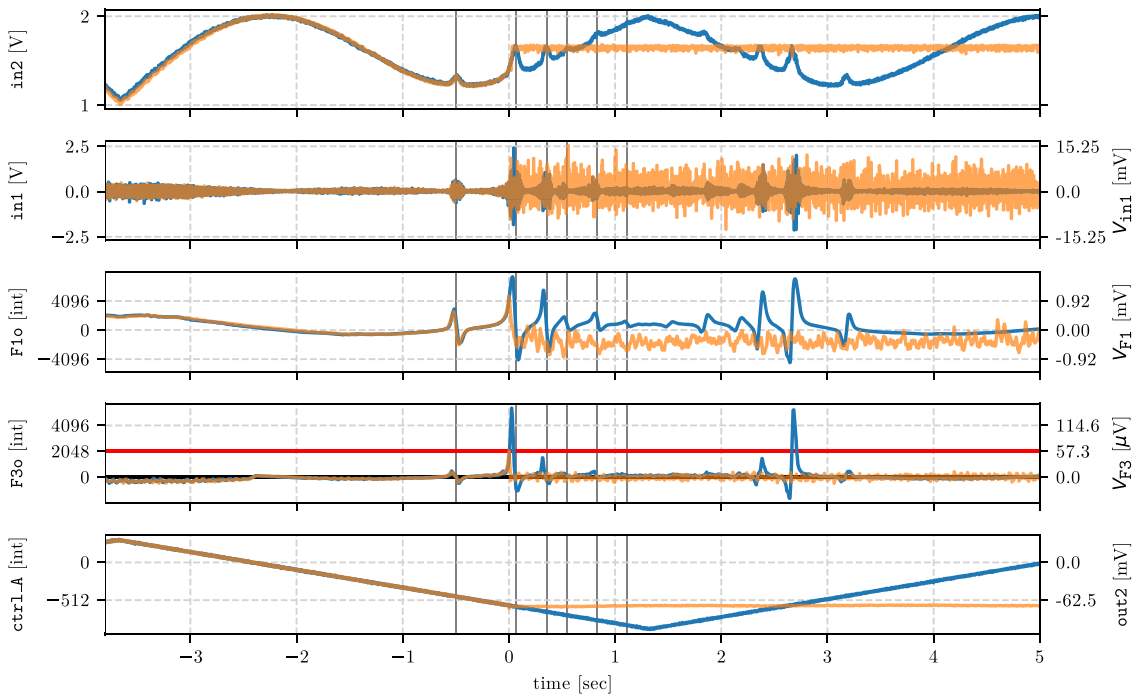

FIG. 10. Relevant signals for a saturated absorption spectroscopy open-loop scan (blue) and a triggered closed-loop stabilization event (orange). in2 is the saturated absorption signal sensed by the photodiode. The laser is scanned across a Doppler-broadened absorption line. At times −3.6 s and 1.2 s, the scan changes the direction. The positive peaks seen are the expected saturated absorption spectral lines. in1 is similar to in2 but has been DC-decoupled and amplified (see Fig. 7). This is the signal used for lock-in demodulation. F1o and F3o are the demodulated signals at 1f and 3f, respectively. ctrl_A is the control signal acting on the laser wavelength. The demodulated signals show the first and third derivative-like behavior, with zero-crossing points on the peaks seen in in2. For locking the laser to a line, the F3o was used as the error signal. The red line shows the level threshold where the lock-start triggers. The right hand side axis indicates a voltage scale of each signal as expected on the photodiode output before amplification and on the Red Pitaya's output.

with ctrl_A. The sub-Doppler hyperfine transition peaks and the crossover[14] peaks were marked with vertical gray lines.

For the stabilization of the emission frequency, a lock-in scheme was implemented, to get an error signal suitable for peak maximum locking, and demodulated from the signal sensed by the photodiode. A harmonic modulation cos_ref was introduced through the current driver.

Two fast inputs and two fast outputs were used in this scheme (Fig. 9). The in2 port was used for direct photodiode signal digitalization, to measure the transmitted laser intensity. The in1 port was used to measure AC components of the photodiode signal, using an analog high pass filter and amplifier, what allows us to improve the sensitivity of lock-in demodulation. The acquisition was made using the oscilloscope instrument, registering the transmitted intensity and the demodulated signal. The ctrl_A signal on the out1 port was used to control the laser frequency. The ramp instrument was used for frequency scanning and PID for frequency stabilization, through feedback loop. The current and PZT sweep amplitudes were tuned through driver hardware. The out2 port was used to produce the modulation of the laser current.

The harmonic lock-in was configured to demodulate the in1 signal. Demodulated signals Xo, Yo, F1o, F2o, and F3o were available for visualization and acquisition. The first three of them are proportional to the first derivative of the transmitted signal in2 (at the first order of modulation depth[17]), and the last two are proportional to the second and third derivatives, respectively. The odd derivatives expose in2 minimum and maximum as zero crossing points. This characteristic makes them suitable as an error signal for min/max peak locking.

In Fig. 10, the F1o and F3o signals are shown. The F3o is not sensible to first derivative offsets, like the linear power increment of the current sweep, and less sensible to peak base line contributions that shift the minimum/maximum positions of the transmitted signal with respect to ideal ones, which is why it was selected as the error signal for the PID input. The filter was configured with a proportional component with constant $k_p = 1.56 \times 10^{-2}$, for fast corrections, and an integral component with constant $k_i = 5.59\ s^{-1}$. The lock-control instrument was configured to trigger the loop-close event whenever F3o gets over 2048 int (red line), with the "level trigger." An example of lock start is presented in Fig. 10, with an orange line superimposed to the normal scan signal. In this example, a stability of 226(13) kHz of the optical frequency was achieved after lock, on a 10 min measurement. The frequency deviation was estimated from F3o standard deviation and the knowledge of the F3o slope on the locked peak position.[18] The

CTRL_A was used to measure the corrections made to the system, as an estimation of the frequency deviation that the laser would have had if it had worked in open-loop configuration: 54(4) MHz RMS over 10 min.

C. Pound-Drever-Hall technique: Square Lock-in

In this experience, an ECDL (centered at 854 nm wavelength) was stabilized to a reflection dip of a high finesse Fabry-Pérot (FP) cavity using the Pound-Drever-Hall (PDH) technique.[19] This stabilization scheme uses lock-in demodulation of the laser intensity reflected from the interferometer to produce the error signal. The modulation frequency must be greater than the cavity's bandwidth, which are both typically chosen to be in the few MHz range. The laser is modulated producing sidebands that lay out of the transmission peak bandwidth of the interferometer, so their reflection produces a beating that can be measured by using a fast photodiode. This technique is often used for laser stabilization in AMO Labs.[19]

The experimental setup is depicted in Fig. 11. Two control signals were used for laser control: ctrl_A for the current driver and ctrl_B for the PZT driver. The radio frequency modulation was incorporated directly into the laser diode using a bias-T configuration and passive electronic components. The reflected signal was measured by using a fast photodiode with DC-decoupled output and digitalized, using port in1, for lock-in demodulation. Another photodiode was used to measure the transmitted signal through the in2 port for system state reference. An example measurement for demodulated signal error and in2 input is shown in Fig. 12.

Two configurations of hardware ports were used in the implementation. Both of them dedicated one of the 14 bit fast outputs (out1) to the current driver. In the first configuration, the other 14 bit fast output (out2) was used for sq_ref modulation signal and one of the 12 bit slow outputs of the extension bus (slow_out1) was used for the PZT driver. The

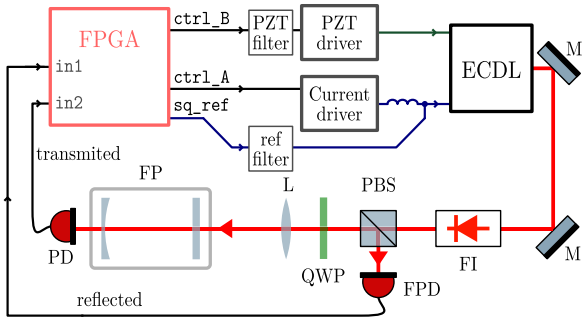

FIG. 11. Scheme of the Pound-Drever-Hall frequency stabilization. An external cavity diode laser (ECDL) is controlled with three signals: a slow control of a piezoelectric element (PZT), a slow control of the current, and a fast modulation of the current (sq_ref) coupled directly to the laser diode with a bias-T. The laser beam is coupled to a Fabry-Pérot interferometer (FP). The reflected and transmitted signals are detected with two photodiodes and are read by the ADCs on the Red Pitaya board.

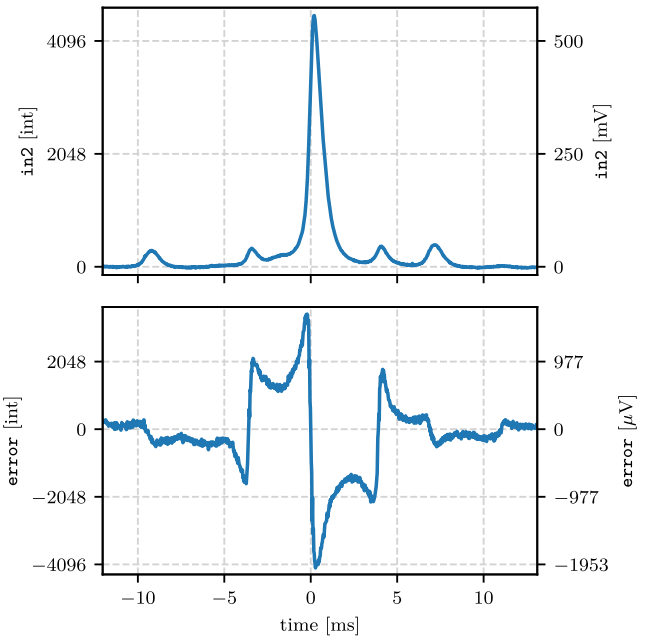

FIG. 12. Transmitted beam intensity (top) and the lock-in demodulation of the reflected intensity (bottom) for a frequency scan with the Pound-Drever-Hall scheme, as shown in Fig. 11. The error signal shows a steep zero-crossing point suitable for frequency stabilization via a PID lock.

slow output update speed is enough for a PZT driver, whose bandwidth limit is in the order of tens of kHz but with the downside that the output signal has a high frequency ripple, which comes from the original PWM signal. To reduce side-effects of the ripple, we included a second order passive low-pass-filter before the PZT driver. The second configuration option was to use the out2 port for the ctrl_B signal of PZT, without any filter. In this case, the sq_ref signal was supplied through one of the extension digital pins (3.3 V at 125 MSa/s) as a square function. A DC-decoupling capacitor and a second order passive low-pass-filter were used to build a band-pass filter, for signal conditioning: suppression of high frequency components, zero-centered signal, and power attenuation.

The sq_ref modulation signal was configured on 31.25 MHz, the higher available frequency for square Lock-in. A second order filter with a frequency cut of f_c = 38.856 kHz was used for demodulation, which produces a time lag of $\Delta t \sim 8.2\ \mu s$ over the error signal. This imposes an upper bound to the feedback gain and/or to the feedback bandwidth[1] but with no fundamental limits to achieve stabilization up to $\frac{1}{2\Delta t} \sim 61\,\text{kHz}$, resilient to acoustic and mechanical perturbations.

The ramp instrument was used for PDH spectrum acquisition, configuring the triangle functions ramp_A and ramp_B with a proportional relation ramp_B_factor tuned for optimal mode-hop-free scanning. The proportional constant was selected considering the hardware configuration option selected. The feedback scheme was built using one error

signal and two PIDs, one for each ctrl signal. The PID for the current setup used proportional and integral terms, while PZT PID only used the integral term, avoiding undesired fast corrections over the piezoelectric line. The sample transmitted and error signal are presented in Fig. 12 for one FP transmission peak, showing the characteristic PDH pattern.[19]

Next, we will present two advanced usage cases of the toolkit for continuous data acquisition and re-lock procedures.

1. Allan deviation: In-device measurements

A measurement of the Allan deviation was made to make a detailed analysis of the stabilization performance. The Allan deviation $\sigma_y(\tau)$[20] provides a detailed description of the stability of a system in several orders of magnitude of time. The value $\sigma_y(\tau)$ is the standard deviation of the differences of successive mean values of y, presented in Eq. (4), where y is the fractional frequency of an oscillator under study,

$$\sigma_y(\tau) = \sqrt{\frac{1}{2}\langle(\bar{y}_{n+1} - \bar{y}_n)^2\rangle}. \tag{4}$$

The measurement of $\sigma_y(\tau)$ requires the continuous acquisition of data channels at a high sample rate for a long time range. This kind of acquisition cannot be performed by using the oscilloscope instrument, limited on a total time range, nor by a remote acquisition procedure, with the sample rate limited because of high communication latency.

To make this acquisition, we implemented an in-device program, running from a Python script in the Red Pitaya operating system. The sample script is published in the online documentation.[12] The program consists in a large loop that reads values directly from RAM memory mapped registers that correspond to the signals that should be saved and prints them on the standard output. This simple approach allows one to redirect the output for local storage or network streaming and remote storage, using the operating system tools. The RAM memory access and the identification of memory addresses are simplified by the usage of the local API. A set of FPGA registers can be frozen for each read procedure, so all the acquired values keep coherence (in the sense that they correspond to the same clock time bin). A 64 bit counter running in the FPGA layer is used to register the accurate internal clock time value of each acquisition.

This implementation allowed to take large measurements of error, ctrl_A, and ctrl_B signals with a sample rate of at least 50 Sa/s along several hours, which were stored in binary files of hundreds of Mbytes in a remote computer. With this information, the Allan deviation of the fractional frequency of the stabilized laser (calculated from error signal) was measured, shown in Fig. 13. Also, the ctrl_A and ctrl_B signals allow us to estimate the open-loop behavior that the laser would have had, by taking into account the corrected deviations during the stabilization time. The fractional frequency Allan deviation derived from these signals was also plotted. The vertical dashed lines mark the PID integrator time constants associated with each ctrl signal, as a reference. For τ_i larger than these references, the curves derived from ctrl

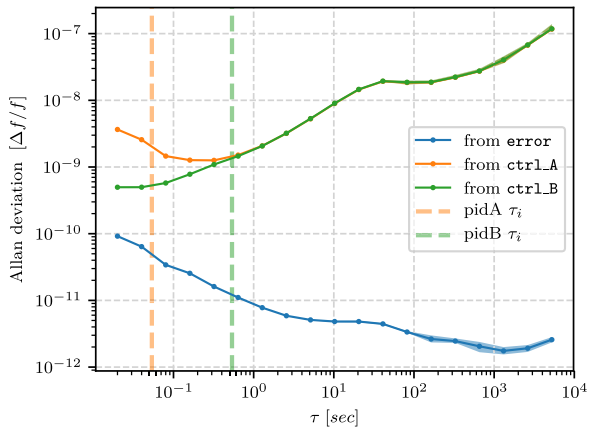

FIG. 13. Allan deviation (4) of a frequency stabilized ECDL. The laser is stabilized to a Fabry-Pérot cavity with the Pound-Drever-Hall technique. The fractional frequency of the laser is derived from the error signal (blue) and from control signals (orange and green). The blue curve reflects the performance of the closed-loop stabilization system on different time scales. The Allan deviations of control signals for time ranges $\tau \gg \tau_i$ represent the corrections made to the system to keep it locked. They can be used as a rough estimation of the open-loop behavior that the system would have had without stabilization. Dashed lines mark the time constants τ_i for each one of the PID integrators. At 10 s, the stabilization scheme achieves an improvement of 3 orders of magnitude with respect to the estimated open-loop behavior.

signals can be interpreted as frequency corrections. The increment on ctrl_A derived values on short times is related with the proportional term of the PID used for current control and tends to reflect the behavior of the error signal. An improvement of three orders of magnitude in the stability at 1 s time range can be seen from the plot.

2. Re-lock system

The lock-control instrument includes the feature to identify locking events and actuate on them, already described in Subsection II C 4. The PDH example provides a case study to test it. With the system locked to a transmission peak, the re-lock tool was configured to trigger when error > 1000 int or when transmitted signal error < 3000 int ≅ 366 mV. The "Out of lock" external trigger allowed to capture the re-lock system response under a stabilization induced fail, by hard hitting the optic table, which is shown in Fig. 14. Three cycles of re-locking can be seen, while the mechanical vibrations are still affecting the system.

3. Parameter optimization and automated tuning

The platform presented allows the user to program scripts which can improve the performance of a locked system in several ways. In particular, the remote programed operation of the toolkit enables automated operation and auto-tuning of parameters. We outline some ideas which can be implemented in this direction.

A simple example we tried is as follows. The fine tuning of several parameters was automated by a systematic procedure of trial and evaluation. By scanning several parameters and evaluating the error signal for a locked system, optimal

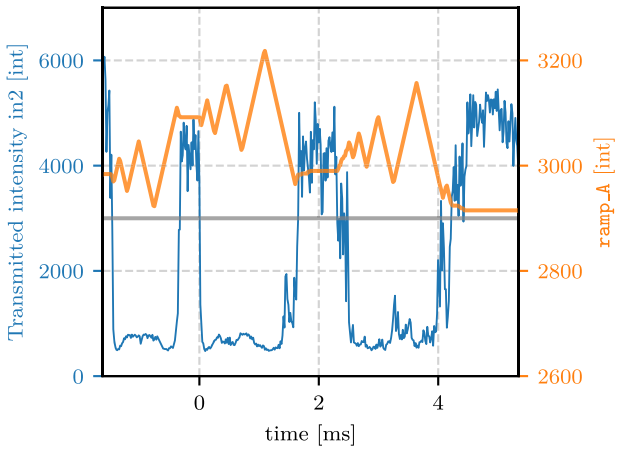

FIG. 14. Relocking behavior example. An ECLD locked to a Fabry-Pérot cavity is pushed out of lock by an external perturbation. When this happens, the transmitted signal (blue) drops below a set threshold (gray line) and the re-lock mechanism is activated. To re-lock, the control parameter is scanned with increasing amplitudes until a new lock condition is found. The perturbation introduced was a mechanical shock, which takes some ms to dissipate. In the process, the laser gets kicked out of lock a few times until it finally stabilizes. In the figure, three re-locking attempts are seen. After the third try, the laser stays in lock. This feature is described in Subsection II C 4.

values were chosen in a semi-automatic fashion. This scan and search procedure can be updated to automatically choose the best parameters.

Another more complex example which outlines a direction in which to work in the future is as follows. We found that the optimal values for reducing the standard deviation of the error signal in a locked system are not always the best for starting a lock or re-locking after a perturbation. A more stable system is sometimes easier to kick out of lock. This points into the direction of adaptive control parameters depending on the state of the system.

In this way, the toolkit could be used as a dedicated device that detects a laser is on, starts a ramp sweep, finds and optimizes an error signal, chooses the adequate peak to lock, configures the lock control trigger, and starts a lock on its own, without user intervention. Also, the re-lock system could be complemented to restart parameter tuning when the lock condition cannot be found again and send alerts to the user when intervention is needed.

IV. CONCLUSION

We presented an embedded toolkit for digital processing, acquisition, and feedback control through MIMO control design. The combination of the FPGA fast deterministic-timing for signal processing and a microprocessor with an operating system for overall monitoring and control provides a balance between programmable electronic precision and algorithm versatility. This allowed the implementation of several usage strategies going from simple off-the-shelf gross-control to in-device programed fine-control, as shown in the experimental examples.

The key features include the following: an in-device operating system which provides portability and multi-platform GUI access through a web browser, PIDs which were designed so that their parameters can be set over several orders of magnitude, enabling the usage on different control applications, even beyond the ones belonging to AMO labs presented in this work, two lock-in instruments for precision measurements with a large working frequency range, including the possibility to implement a complete Pound-Drever-Hall lock scheme at 31.25 MHz in one device.

The selection of an economical commercial board[9] may ease the acquisition and fast implementation of this toolkit by a third party in new experiments. Also, the compact design and remote programmable feature make it useful for mass implementation in experiments with several control systems with the centralized monitoring and operation. The FPGA design and software of the presented toolkit are public domain,[10] and the board's operating system and application framework are open-source.

ACKNOWLEDGMENTS

We thank Ferdinand Schmidt-Kaler for opening the doors of the Cold ions and quantum information laboratory (Institut für Physik, Mainz), where the first laser stabilization tests with FPGA were conducted. The financial support for this research was provided by the Ministry of Defense of Argentina (PIDDEF 23/14) and the National Agency of Promotion of Science and Technology (No. PICT 2014-3711.11).

REFERENCES

[1] J. Bechhoefer, Rev. Mod. Phys. **77**, 783 (2005).
[2] A. Schwettmann, J. Sedlacek, and J. P. Shaffer, Rev. Sci. Instrum. **82**, 103103 (2011).
[3] Z. Xu, X. Zhang, K. Huang, and X. Lu, Rev. Sci. Instrum. **83**, 093104 (2012).
[4] K. Huang, H. L. Jeannic, J. Ruaudel, O. Morin, and J. Laurat, Rev. Sci. Instrum. **85**, 123112 (2014).
[5] N. B. Jørgensen, D. Birkmose, K. Trelborg, L. Wacker, N. Winter, A. J. Hilliard, M. G. Bason, and J. J. Arlt, Rev. Sci. Instrum. **87**, 073106 (2016).
[6] R. A. R. Picone, S. Davis, C. Devine, J. L. Garbini, and J. A. Sidles, Rev. Sci. Instrum. **88**, 045108 (2017).
[7] D. R. Leibrandt and J. Heidecker, Rev. Sci. Instrum. **86**, 123115 (2015).
[8] L. Neuhaus, R. Metzdorff, S. Chua, T. Jacqmin, T. Briant, A. Heidmann, P.-F. Cohadon, and S. Deleglise, in *2017 Conference on Lasers and Electro-Optics Europe and European Quantum Electronics Conference (CLEO/Europe-EQEC)* (IEEE, 2017), Vol. 2016, p. 1.
[9] See https://www.redpitaya.com/f130/STEMlab-board for Red Pitaya, "Stemlab."
[10] M. Luda, "Lock-in+pid" (2018), https://github.com/marceluda/rp_lock-in_pid.
[11] See https://github.com/RedPitaya/RedPitaya/tree/release-v0.95/apps-free for Red Pitaya, "Free applications," 2015.
[12] M. Luda, "Lock-in+pid project site" (2018), https://marceluda.github.io/rp_lock-in_pid/.
[13] C. J. Chang-Hasnain, IEEE J. Sel. Top. Quantum Electron. **6**, 978 (2000).
[14] D. Bhattacharyya, B. K. Dutta, B. Ray, and P. N. Ghosh, Chem. Phys. Lett. **389**, 113 (2004).
[15] D. W. Preston, Am. J. Phys. **64**, 1432 (1996).

[16]T. Führer and T. Walther, Opt. Lett. **33**, 372 (2008).
[17]W. Demtröder, in *Laser Spectroscopy*, 3rd ed. (Springer Berlin Heidelberg, Berlin, Heidelberg, 2008), Chap. 6, pp. 7–10.
[18]M. Luda and J. Codnia, Opt. Pura Apl. **51**, 49031:1 (2018).
[19]R. W. P. Drever, J. L. Hall, F. V. Kowalski, J. Hough, G. M. Ford, A. J. Munley, and H. Ward, Appl. Phys. B: Photophys. Laser Chem. **31**, 97 (1983); e-print arXiv:1602.03504.
[20]J. R. V. (Chairman), IEEE Standard 1139, 1999, p. 36.