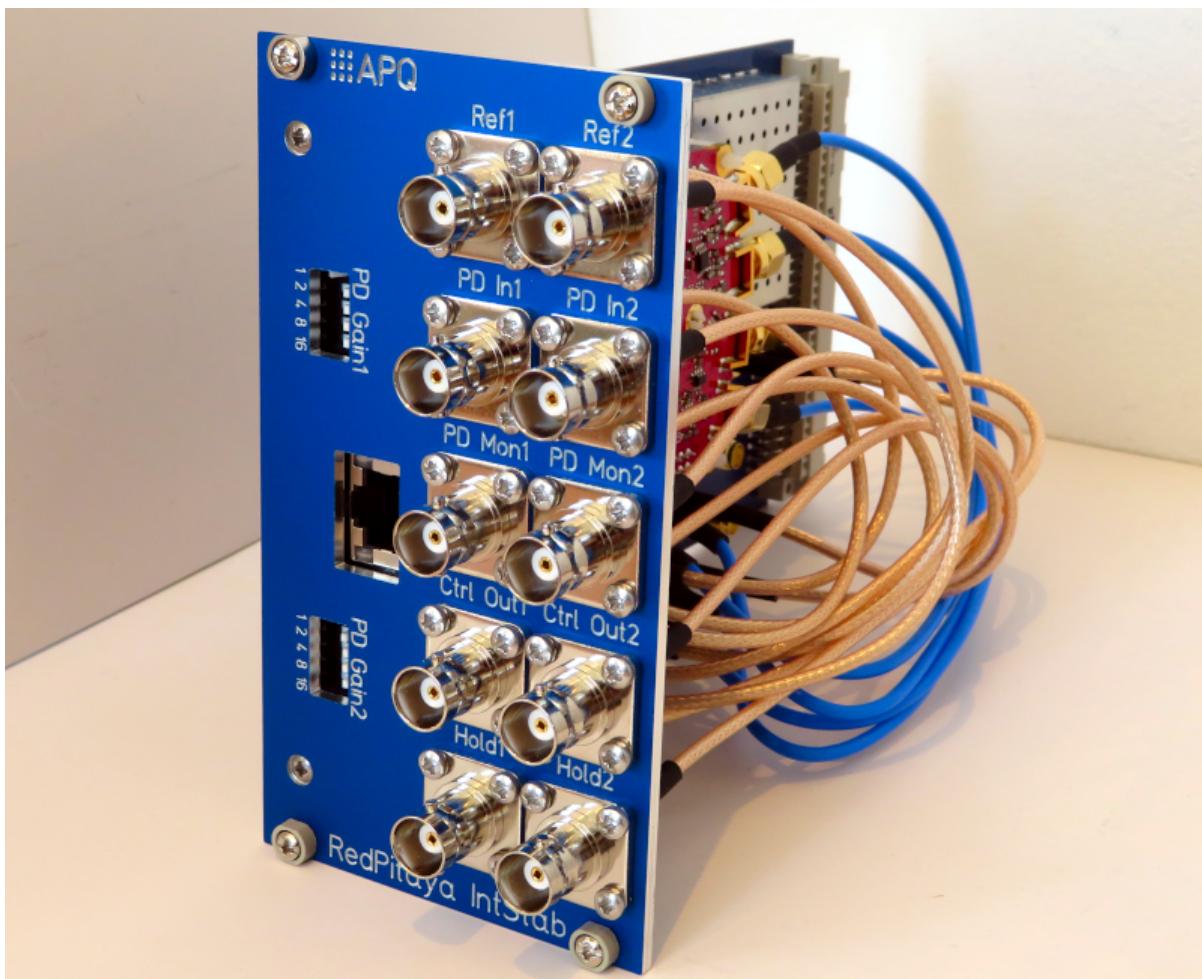

RedPitaya IntStab – Manual

Latest board revision 1.5.1

Tilman Preuschoff

Last modified: 26.09.2025



Abstract

The [RedPitaya IntStab](#) is a fast digital controller for laser intensity stabilization featuring two independent channels. The commercial FPGA based [RedPitaya STEMlab 125-14](#) board is complemented by an analog interface that provides all necessary functionalities – such as an input stage producing an error signal from a reference signal and a photodiode signal, a buffered control output and

additional monitoring outputs. The device is operated with the open-source software package PyRPL.

Contents

1 Setting up the RedPitaya	2
1.1 Mounting the RedPitaya on the IntStab PCB	2
1.2 Power supply and network connection	6
2 Setting up the software	6
3 Connecting the RedPitaya IntStab to the experiment	8
3.1 Description of input and output functionalities	8
3.2 Hardware application notes	9
4 Intensity stabilization with PyRPL	11
4.1 PyRPL configuration	11
4.2 PyRPL settings	11
4.3 Set the PI parameters	12

1 Setting up the RedPitaya

1.1 Mounting the RedPitaya on the IntStab PCB

Install the RedPitaya OS Install the RedPitaya OS on a microSD card (4 GB min.) following the instructions on <https://redpitaya.readthedocs.io/en/latest/quickStart/SDcard/SDcard.html>. PyRPL has been tested with software release 0.94 to 0.99 as well as 1.04. The latest release (2.05-37) does not work with our PyRPL branch so that using release 1.04-9 is recommended. A local copy of this image can be found on \\brain43\public\Labortechnik\Elektronik\RedPitay\Images. Older releases can be found on this website: <http://downloads.redpitaya.com/> downloads/. Insert the card in the SD slot of the RedPitaya after installation.

Remove the DC-offsets Remove the noisy DC-offset for both fast RedPitaya outputs by removing the four resistors marked in Fig. 1. Use a standard soldering iron. This shifts the output voltage range from ± 1 V to 0 V–2 V. Additional information can be found on <https://ln1985blog.wordpress.com/2016/02/07/red-pitaya-dac-performance/>. After removing the resistors, label the board accordingly (e.g. 'offset removed').

Note: The software (PyRPL and STEMlab) will still display and assume a ± 1 V range.

Note: The default output is now 1 V instead of 0 V.

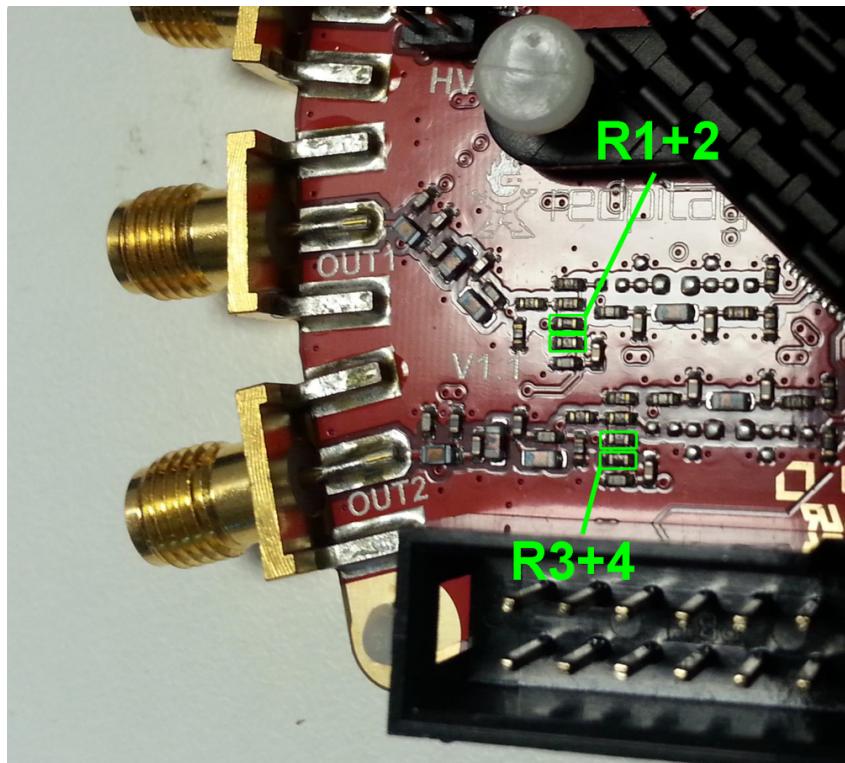


Figure 1: RedPitaya fast output offset resistors. [Image by L. Neuhaus]

Mount the RedPitaya board
on the PCB

Move the four nylon PCB mounts of the RedPitaya to the other side of the board. Attach 17 mm hexnut M3 distance bolts to the PCB mounts using headless M3-screws. Connect the RedPitaya board (upside down) to the two 26-pin IDC-connectors on the PCB. Fix the board using M3 pan head screws and M3 washers.

Note: The RedPitaya's MAC address is denoted on the network connector. Make sure to remember it since it will not be visible after mounting the board.

Note: Make sure all inputs are set to the low voltage range ('LV'). See also https://redpitaya.readthedocs.io/en/latest/developer_Guide/125-14/fastIO.html#analog-inputs.

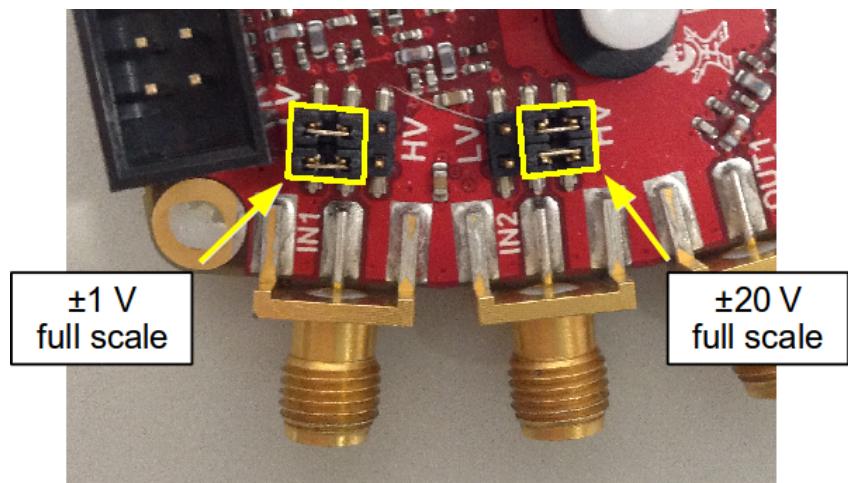


Figure 2: Jumper settings for ± 1 V and ± 20 V input range.

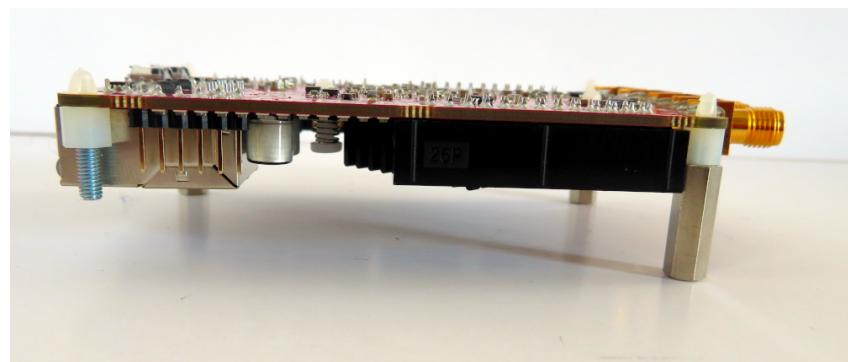


Figure 3: RedPitaya board PCB mounting assembly.



Figure 4: Fully assembled RedPitaya IntStab PCB (Rev1.3.1).

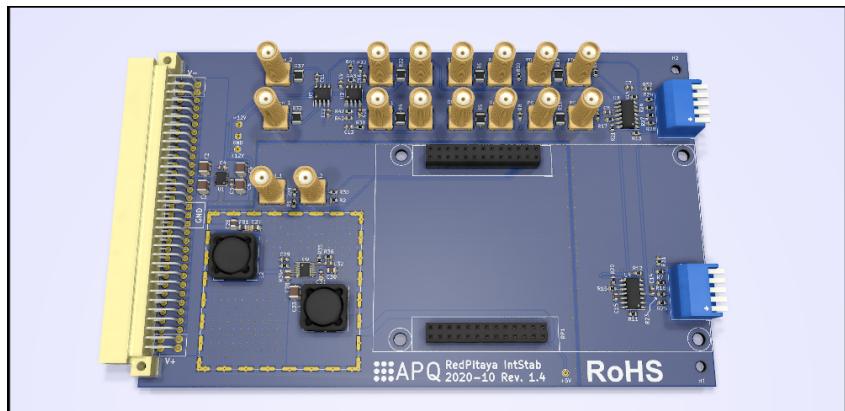


Figure 5: 3D-model of the latest PCB revision 1.4.

Install the frontpanel Use a pre assembled frontpanel and four additional 20 cm SMA cables. Fix the frontpanel on the IntStab PCB using two 10 mm pan head M2.5 screws and M2.5 washers. Connect the cable according the table below.

Front panel	PCB	RedPitaya
Ref1	Ref_In_1	
PD In1	PD_In_1	
PD Mon1	PD_Mon_1	
Ctrl Out1	Ctrl_Out_1	
Hold1	Hold_1	
	Ctrl_In_1	Out1
	Diff_Out_1	In1
Ref2	Ref_In_2	
PD In2	PD_In_2	
PD Mon2	PD_Mon_2	
Ctrl Out2	Ctrl_Out_2	
Hold2	Hold_2	
	Ctrl_In_2	Out2
	Diff_Out_2	In2

Note: The *Diff_Mon* outputs on the PCB are optionally connected to an external frontpanel. Alternatively, they can be connected to the *Hold* inputs on the frontpanel if the sample and hold feature is not used. In this case, label the frontpanel accordingly.

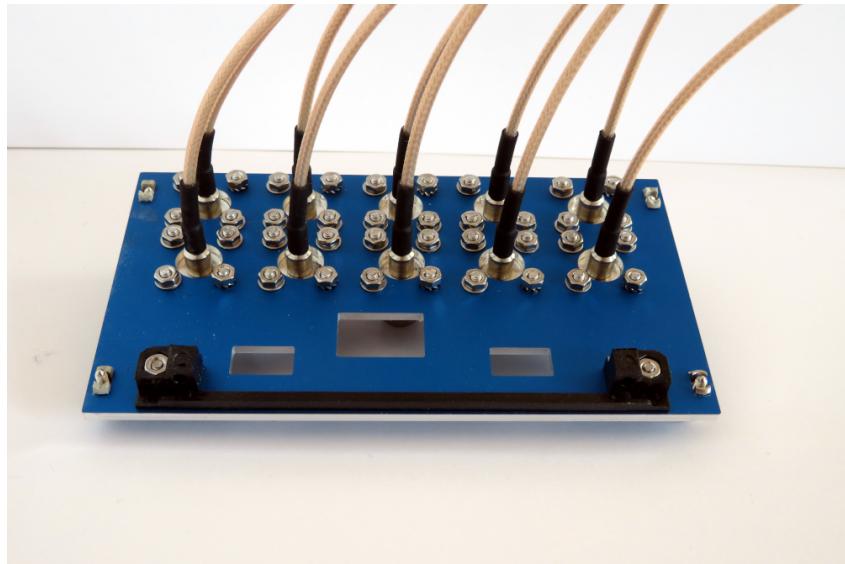


Figure 6: Pre-assembled frontpanel (RedPitaya IntStab).

1.2 Power supply and network connection

Power supply The RedPitaya lockbox is mounted in a 19-inch Fischer rack mount unit with our standard power supply (DIN41612 C64A/C connector). A ± 15 V supply is recommended.

Supply	Voltage range	Current@15 V (typ./max.)	Pin config. (default / Toptica)
V+	12.5 V to 20 V	0.35 A / 1.00 A	1A, 1C, 2A, 2C / 31A, 31C
V-	-20 V to -12.5 V	0.05 A / 0.15 A	31A, 31C, 32A 32C / 30A, 30C
GND	-	-	16A, 16C, 17A, 17C, 18A, 18C / 1A, 1C

Table 1: RedPitaya lockbox power supply rating. For connector pin configuration, see [Harting 09031646921](#).

Note: Always check the power supply polarity and current rating before connecting the controller. Permanent damage to the device may occur by reverse polarity. The RedPitaya requires a larger current while powering up.

Network connection: Connect the RedPitaya to the local network. The color code for RedPitaya network cables is blue. Per default the RedPitaya IP address is acquired via DHCP (see also next section).

2 Setting up the software

Install PyRPL The RedPitaya is operated with the Python software package PyRPL. You need a proper Python installation in order to run the software. For Windows OS, use [Anaconda Python](#). Create a virtual environment for PyRPL. The whole process is outdated, so to begin, an auxiliary environment is created. Some of the packages used require older versions of

Python and Anaconda, starting with Python version 3.11 and Anaconda-Navigator 2.4.0 is recommended. To create the correct environment, run

```
conda create -n pyrpl python=3.9 numpy=1.19
pyqtgraph=0.12 pandas=1.3 scipy paramiko nose pip
pyqt qtpy=1.9.0 pyyaml ipywidgets notebook
conda activate pyrpl
pip uninstall pyopenssl cryptography
pip install pyopenssl==22.0.0 cryptography==36.0.2
pip install quamash
pip install scp
```

Get *pyrpl-env.yaml* from <https://github.com/TU-Darmstadt-APQ/RedPitaya-IntStab/tree/master/pyrpl> and run (in the directory of the file)

```
conda env create -f pyrpl-env.yaml
```

Note: PyRPL is not compatible with the latest python and numpy version.

Activate the environment via

```
conda activate pyrpl-env
```

Clone or download the branch 'external_pid_pause' from https://github.com/lneuhhaus/pyrpl/tree/external_pid_pause. In the directory of the source files, run:

```
python setup.py develop
```

Note: The branch 'external_pid_pause' is not the latest stable version of PyRPL. It will not be updated. If the installation does not work correctly and the sample and hold feature is not used, try installing the 'master' branch in a new environment.

Known issues:

- 'pid'-module: Output limits seem to have a -1V offset resulting in a max. output of 0V. Applying a 1V DC-offset with the 'asg'-module fixes the problem.
- 'lockbox' module: 'sweep' is only activated when the 'pid'-module is active.

Check the network connection and find the IP: The STEMlab web interface can be used for that task. Open this URL with a computer in the RedPitaya's local network:

```
http://rp-*****.local
```

Replace ***** by the last six digits of the RedPitaya's MAC address. Navigate to Settings→Network in order to find the IP.

Note: Label the frontpanel with the device IP, MAC-address and PCB revision.

Set the RedPitaya password The default login for SSH connection to the RedPitaya is

```
User: root    Password: root
```

Connect to the RedPitaya using a SSH terminal (e.g. `putty`). The password can be changed with the command:

```
passwd
```

Note: The password will be saved in plain text in the PyRPL config.-file. Do not reuse another password.

Configure PyRPL Get the PyRPL example configuration file `example_intstab.yml` from <https://github.com/TU-Darmstadt-APQ/RedPitaya-IntStab/tree/master/pyrpl>. Copy `example_intstab.yml` to `\pyrpl_user_dir\config`. Rename the configuration file according to your application, open it with a text editor, and set the RedPitaya's IP (`redpitaya>hostname:`), password (`redpitaya>password:`) and name of the application (`pyrpl>name:`). The default location of the `pyrpl_user_dir` is `C:\Users\USER`.

Run PyRPL After configuration, PyRPL can be started by activating the virtual environment and loading the configuration file (in a python terminal):

```
activate pyrpl-env
python -m pyrpl your_config_file.yml
```

For Windows OS, it is also possible to put the commands in batch-file after adding Anaconda to your path variable:

```
set PATH=%PATH%; C:\Users\USER\Anaconda3\Scripts
```

Make sure the path to the Anaconda directory is correct. At the first start, open the 'lockbox' and 'scope' module and place the tabs at the right position. The changes are saved automatically to your configuration file.

3 Connecting the RedPitaya IntStab to the experiment

3.1 Description of input and output functionalities

The RedPitaya IntStab features two identical channels. All connections are labeled '1' and '2' accordingly.

PD In High-Z input for the photodiode signal. The signal can be amplified internally. Input voltage range: ± 10 V. Gain bandwidth product: 35 MHz.

Gain DIP switches Sets the photodiode gain between discrete values. Push the switch towards the number labels to close it. Open all other switches to set the denoted gain. It is also possible to combine the switches. The resulting gain k from the set gains k_1, k_2, \dots, k_n is

$$k = 1 + \left(\sum_i (k_i - 1)^{-1} \right)^{-1}$$

Note, that a larger gain will limit the bandwidth of the input stage. However, the bandwidth will be larger than the maximal control bandwidth of 1.25 MHz for all applicable gains.

Err Mon	High-Z input for the reference signal. The signal is buffered and subtracted from the amplified photodiode signal. The resulting error signal ('Diff_Out') is fed to the RedPitaya input. Input voltage range: ± 10 V. The RedPitaya input will saturate at ± 1 V. The rated max. input voltage for the RedPitaya is 30 V. Gain bandwidth product: 35 MHz. Note, that the subtracting amplifier stage has gain 2.
Ctrl Out	Buffered and low-pass filtered output of the RedPitaya. Connect this output to the actuator, e.g. the intermediate port of a frequency mixer. The buffer can drive a 50Ω -load at the maximal output voltage. Series resistance: 50Ω . Voltage range: 0–2 V. The default output voltage is 1 V (due to the removed RedPitaya offsets). Max. current: 30 mA. Cut-off frequency: 5 MHz. The cut-off frequency can be altered by hardware modifications (see schematics for details).
PD Mon	Monitoring output for the amplified photodiode voltage. Output voltage range: ± 10 V. Max. current: 30 mA. Series resistance: 50Ω . Do not load the output with $Z < 500\Omega$.
Hold	TTL-level logical input. If this input is pulled high, the P- and I-gains of the controller are set to zero with a 150 ns delay resulting in a constant control voltage. In this way a sample and hold feature is implemented. Input resistance: $3\text{k}\Omega$.
Diff Mon	Monitoring output for the error signal fed to the RedPitaya. Output voltage range: ± 10 V. Max. current: 30 mA. Series resistance: 50Ω . Do not load the output with $Z < 500\Omega$. This output is not connected to the frontpanel per default.

3.2 Hardware application notes

Control signal	For our typical application, the control signal is applied to a diode-ring frequency mixer (e.g. MiniCircuits ZAD-3+) used as a variable RF-attenuator in order to control the RF-signal level of an AOM. Make sure the maximum ratings of the RF signal chain and the AOM are not exceeded when applying the full scale voltage to the intermediate port of the mixer. In order to minimize the influence of DAC quantization noise, it is also preferable to use the full output range for stabilization (i.e. the maximal intensity requires an RedPitaya output of approximately 2 V). This can be achieved by sufficient attenuation (usually 6 dB to 10 dB) of the RedPitaya's output.
----------------	--

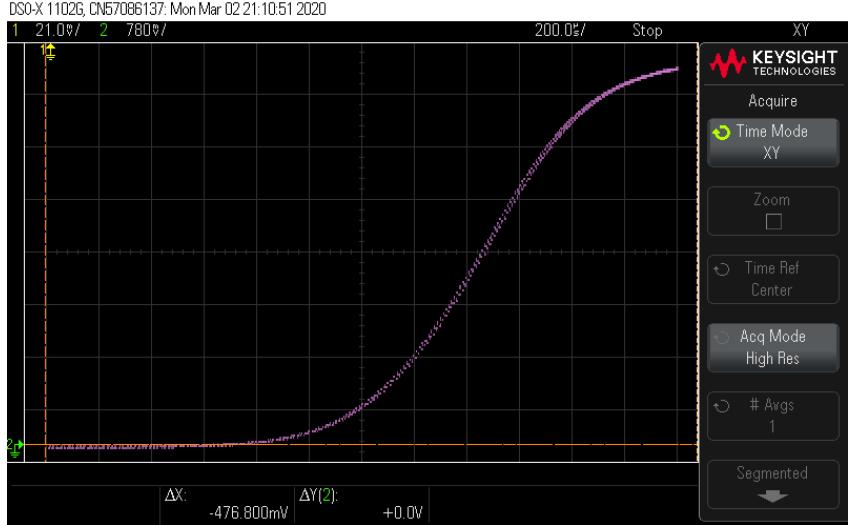


Figure 7: Non-linear frequency mixer transfer function (Mini Circuits ZAD-3+). x (channel 1): Voltage at the intermediate port. y (channel 2): Photodiode signal (deflected intensity).

External loop gain

The external loop gain is given by the ratio between a change in input signal (i.e. photodiode signal) and the change in output signal (Ctrl Out) causing it. It determines the transfer function of the PI controller. Hence, an altered external loop gain requires adapted PI parameters. Note, that the external loop gain is sensitive to multiple changes in the setup, e.g. changes in the optical power applied to the AOM, the peak diffraction efficiency, the maximal RF-power, the fraction of light guided to the photodiode, and the gain of the photodetector (including the gain setting of the RedPitaya IntStab). The external loop gain also depends on the laser intensity setpoint in relation to the maximally deflected intensity due to the non-linear relation between RF-power output of the frequency mixer and control voltage applied to the mixer.

Input noise

Ideally, the obtained intensity noise is limited by photodiode shot noise. In this case, the performance is enhanced by using more light for the stabilization, i.e. on the photodiode. All noise contributions on the reference signal and the photodiode signal (e.g. 50 Hz ground loops) within the control bandwidth are added to the laser intensity noise. In particular, large gain photodiode amplifiers are sensitive instruments. Try minimizing the 'dark' photodiode noise by sufficient shielding. For the reference signal, low-pass filtering might help if fast level transitions are not required.

Control bandwidth

The maximal digital control bandwidth of the RedPitaya is 1.25 MHz limited by a 200 ns propagation delay. The available bandwidth is determined by the phase margin given by the sum of all delays in the system. The phase margin is not necessarily linked to the amplitude transfer function. Several components in the system may induce delays: The AOM has a major share on the overall delay depending on the beam position in the crystal.¹ Also long cables or optical fibers may be relevant. Use a photodetector with a bandwidth well above 1.25 MHz. Since the band-

¹ A typical speed of sound is 4200 m/s resulting in a delay of 240 ns for a distance of 1 mm between the active area and the RF actuator.

width of the RedPitaya's fast output is much larger than the control bandwidth, an additional output filter adapted to effective bandwidth of the system reduces the noise induced by the RedPitaya.

Saturation of the integrator	The digital integrator value has a saturation limit of 4 while the maximal output voltage corresponds to 1. Hence, if the control loop is closed, saturation of the output (e.g. if the setpoint cannot be reached) will only cause an additional delay comparable to the usual delay for transition to a different intensity level. This delay can be ignored if a minimal transition time is not required. In some cases, the control loop is not closed during the experiment. This is the case if the photodiode is blocked by a shutter or if a fast switch is used to control the RF power. In these cases, positive saturation (i.e. a I-value of +4) will lead to significant overshoot when closing the control loop. This can be avoided by setting appropriate reference levels while the control loop is not closed leading to a negative saturation of the output (i.e. a I-value of -4). Note, that most photodetectors have a significant negative offset requiring a negative setpoint. Alternatively, the sample and hold feature can be used.
------------------------------	---

4 Intensity stabilization with PyRPL

4.1 PyRPL configuration

The pre configured example uses the 'pid' module to implement two independent PI controllers and the 'scope' module to observe the error signal (Diff_out).

4.2 PyRPL settings

Input filters	PyRPL allows to set four digital filters as input filters for the PI. Use this feature if the bandwidth of the photodetector is much larger than the control bandwidth.
Sample and hold	In the 'pid' module, set 'pause_gains' to 'pi' if the sample and hold feature is used. The sample and hold feature can be activated manually by activating the 'paused' checkbox. Note, that the features are only available if the branch 'external_pid_pause' is installed.



Figure 8: Illustration of the sample and hold feature without feedback for a P-controller (top) and an I-controller (bottom). Blue: 'Hold' input. Yellow: RedPitaya input. Green: RedPitaya output.

4.3 Set the PI parameters

Calibrate the ADC offset

The RedPitaya ADCs have considerable offsets and gain errors. The STEMlab software features a software calibration that is not used with PyRPL. Gain errors can be neglected since our setpoint is 0 V. Nevertheless, the remaining offset leads to intensity offsets. The offsets have proven to be constant over time. In order to calibrate the input, terminate 'Ref' and 'PD In' with a 50Ω -resistor (or short). Use the 'scope' module to determine the ADC DC-offset and set the 'setpoint' in the 'pid' module accordingly.

Maximal photodiode voltage

Determine the maximal (amplified) photodiode level using the 'PD Mon' output. Deactivate the PI and set the respective output to '1V' (i.e. 2 V) using the 'asg' module. Increase the 'gain' using the DIP switches if the value is below 1 V. If possible, increase the power at the photodiode instead of increasing the gain.

	Note: If the external loop gain is changed the PI parameters must be adapted (see Section 3.2 for details).
I-gain	Choose a setpoint of approximately 50 % of the maximal photodiode voltage. Set the P-gain to zero and increase the I-gain until the system start to oscillate. Use half of this I-gain as a starting point. Note: In order to lock to a positive slope, the set PI-parameters in the 'pid' module have to be negative.
Optimal PI parameters	Apply a short pulse (typically 10 μ s) with a low repetition rate to the reference port and observe the 'PD Mon' output (and the reference pulse) using a scope with a signal generator. Choose the pulse height and offset such that both levels are in the linear region of the actuator (see Fig. 7). For the ZAD-3+, a transition from 25% to 75% of the maximal photodiode voltage is sufficient. Optimize the I-gain for minimal rise and fall times with minimal overshoot. Adding a small P-gain will allow for larger I-gains without oscillations/overshoot. Note: Due to the non-linearity of the mixer, the exact behavior of the controller depends on the setpoint in relation to the maximal photodiode level. E.g., a pulse from 0% to 90% will have larger rise and fall times or a pulse from 0% to 50% will be slightly asymmetric. The suggested optimization strategy yields a robust control performance avoiding oscillatory behavior and featuring decent transit times with minimal overshoot for all pulses. Additionally, these settings provide a maximal control bandwidth resulting in an optimal noise performance for a constant setpoint. Note: The exact performance (rise/fall times and delays) depends strongly on the AOM-type, the beam size, and position in the AOM since these parameters determine the available control bandwidth.



Figure 9: Typical performance for different PI parameters. Top: I-gain too large. Centre: I-gain too small. Bottom: Optimal value.