**Automated OPO Locking System User Guide**

By Rahul Chawlani, Nonlinear Photonics Group at Caltech

Revisions

08/19/2022: First Draft Created and Published by Rahul Chawlani

Congratulations! You have taken an important step in having an automated locking system for your OPO. Included in this manual is a guide on how to use this software, trouble-shooting tips, and more.

**Table of Contents:**

1. Starting up the Program
2. How to Use
3. Common Troubleshooting Techniques
4. Source Code
5. Links to Helpful References

**Starting up the Program**

When given the source code, immediately divert your attention to the following line:

url = 'http://131.215.138.174/data'

on line 37 of automated\_locking.py. Find the IP address of the Red Pitaya FPGA used in your setup and find the Lockin+PID controller from the Red Pitaya software, and copy the url into this string. Make sure you have /data and not /lockin+pid or /run?.

We also have made several startup options. The first being that the PID settings have been set to PID B and the values are:\

P = 200

I = 200

D = 0

And the Phase offset being = 50

Next, it is important to check what peak your laser is expected to lock at. We do not consider any peaks below 1.20 (a variable known as peak\_cutoff). Thus, change it to a reasonable value based on your experiment. (In a later update, this may be changed to be automated.)

Also important is setting up the oscilloscope. Make sure to connect it to the ethernet (and double check the connection!) Capture the IP address on the oscilloscope (we assume a RIGOL model, for non-RIGOL models, check the user guide and adjust the code in locking\_driver.py as necessary) and record it in OSC\_ADDR in locking\_driver.py. As a test, ping the IP address and wait for a success to occur.

At this point, we can now start connecting the Oscilloscope to the Red Pitaya and OPO. After doing this, we should be ready to start up the GUI. For safe measures, restart the kernel if working in Spyder. Upon running the automated\_locking.py, you should see a GUI start up.

**How to Navigate the Program**

Graphical user interface, application, Teams

Description automatically generated

Upon running the program, you’ll receive the GUI upon start up. You have two options: Lock onto a maximum peak or choose a peak.

To lock onto the maximum peak value, simply press “Lock with Max Peaks” at the top. By doing so, it will scan the oscilloscope data, and choose the highest value. It will then apply an offset and lock on that peak. Then, the program will wait 5 seconds and enter a while loop. If it detects that the power has dipped below 92.5% of the peak power (this variable can be changed in the code), then it will search for the maximum peak and relock. To check this value, it will print a message and display the peak value where it is at. To stop locking, simply press “Stop Locking” to stop locking and stay in the program, or click “Quit/Stop Locking” to stop locking and leave the program. IF YOU EXIT OUT OF THE PROGRAM WITHOUT STOPPING THE LOCK, YOU WILL HAVE TO STOP LOCKING BEFORE YOU RELOCK THE NEXT TIME YOU ENTER OR ERRORS WILL OCCUR!

To lock on a custom peak, press “Show Peaks.” Upon pressing, the program will wait 1-2 seconds while waiting for the oscilloscope to ramp and it will display this updated window:

Graphical user interface, application

Description automatically generated

There are usually between 4-5 peaks. It is possible it is more, but if there are more, it will only save those on a positive time trace. It is then possible to choose one of the peaks by pressing one of the buttons. If you wish to stop locking onto that peak and choose a new one, you MUST hit “Relock User Selected Peaks” and not choose another peak. The program will reload and 5 new peaks will show up. This is because in the time that is has locked on to a specific peak, the values have most likely shifted. Again, like before, press any of the stop locking buttons to stop the lock. If you want to switch to locking with max peaks, stop the lock with “Stop Locking” before pressing Lock with Max Peaks.

Finally, in this program, it is possible to change the PID settings without changing the source code. Go to “PID settings” and go into that window:

Graphical user interface, application

Description automatically generated

Here, you can enter 4 valid float values and update the settings. Note that you must input all 4 values at the same time or it will not record. Thus, if you wish to change P, I, D, but keep the Phase value, you must record all 4 values. When you hit apply, it will change the internal P, I, D, and Phase settings and will be updated the next time you lock.

**Common Troubleshooting Techniques**

1. Not recognizing the device, peak images, or other files: Make sure to have everything in the same path (including the old images! They will be rewritten over).
2. PyVISA Error: This is perhaps the most common and the most troublesome problem. There are two likely culprits: The laser or the oscilloscope.
   1. Check the laser: One good idea is to check the laser is on. Try to turn it off then on again.
   2. If that doesn’t work, the most likely culprit is the Oscilloscope. For further information, check the oscilloscope user manual at the end of this document. But essentially, first try unplugging the ethernet cord and plugging it back in. If you can ping the oscilloscope successfully, but still receive the PyVISA error, try turning off the oscilloscope and on again.
   3. Once you turn it back on, look at the trigger levels, make sure to set them to an appropriate value to make sure we actually trigger the oscilloscope. If we can’t trigger, then we can’t read values.
   4. Also, whenever you’re checking for errors, TURN OFF THE LOCKING
3. Argmax/Value errors: This is most likely because the laser is off. It should catch whenever this happens and purposefully unlock the system, but it may happen. If this does happen, simply follow the steps under 2.
4. Locking Issues: Sometimes, when we lock, we run into issues such as the PyVISA issue. Unlike 2, this is software related since we already connected to the device before. Thus, to fix this, simply stop locking and quit out of the program. Re-enter the program and start locking again. Again, I want to repeat this, before you want to manually relock, you must stop the current lock. If we don’t stop the current lock, you have two threads moving side by side in parallel that are both controlling the OPO.
5. Ramp Issues: Sometimes, if you ramp and start locking, it will fail. If you check the oscilloscope and do not see the full ramp, that will cause issues. At the very least, adjust the voltage and time settings on the oscilloscope to see the bottom of the ramp.
6. Not unlocking: Simply hitting Stop Locking will allow you to stop your current lock and recreate a new lock. However, if you want to completely stop locking, an issue may occur where the power will decrease, but it will still stay in lock. If this happens, this issue occurs on the Red Pitaya side. Simply reloading the Red Pitaya program will fix this issue and throw the OPO out of lock. Rerunning the GUI will lock as normal.
7. Test bench: For a testbench to the code (this helps test the connection to the devices, not the actual locking mechanism), see locking\_tb.py. Running this can help you debug whether an issue is coming from the oscilloscope side or the software.

Graphical user interface

Description automatically generated

1. Sometimes, as we can see in the figure above, not all the peaks are captured correctly. In this case, simply either choose peak 2 if you’d want the first peak or click “Relock User Selected Peaks” to reload the images.

If there are ever issues that cannot be resolved by looking at this guide, feel free to contact me at [rchawlan@caltech.edu](mailto:rchawlan@caltech.edu) or send me a Slack Message.

**Source Code**

**https://github.com/rchawlani/Continuous-OPO-Locking**

**References**

<https://www.batronix.com/pdf/Rigol/ProgrammingGuide/DS1000DE_ProgrammingGuide_EN.pdf>

https://redpitaya.com/