Comb-Assisted Dispersion Measurement: Rees McNally

I hope this document will serve as a reference for understanding, setting up, and performing a frequency comb assisted dispersion measurement. For more detailed instructions and debugging assistance please email me (reeslmcnally@gmail.com) and I will be more than willing to help. This document will include three sections. The first, will be a basic discussion of how the measurement works, and the results already obtained using this setup. The second, will be a quick reference guide, to walk a user through one complete measurement with the system. This will be augmented by comments in the collection and processing code, available for download from my github account ($https://github.com/rm3334/Dispersion\_Code$). The third section, will discuss known issues I have seen while working with the setup, as a first line of debugging assistance. Again, if you have any questions please email me to ask, I am glad to help.

Measurement Principle, and Observed Performance

This is a very basic walk through of how the experiment works, based on the original paper by the Kippenburg group (DOI: 10.1038/NPHOTON.2009.138), with emphasis on how we actually implemented the measurement ourselves.

Measurement Principle

The idea behind this measurement, is to scan a tunable laser across many resonant modes of a micro-resonator, and accurately record the frequency at which each cavity resonance occurs. These frequencies, or specifically the difference between these frequencies, allows use to estimate the Free Spectral Range (FSR) of the cavity, at a variety of laser frequencies. This allows us to estimate the group delay dispersion at a given frequency ($GDD(\omega)$) using the equation

$$GDD(\omega) = \frac{1}{2\pi}\frac{d}{d\nu}\frac{1}{FSR},$$

so if we can obtain the FSR as a function of frequency, the slope of a line fitted to 1/FSR vs. ν directly gives an estimate of the GDD.

What complicates this, is this value is typically very near to zero, and taking a derivative of your data introduces a lot of noise. This means we need a way to very accurately know the frequency of the scanning laser. This is done, by beating the scanning laser against a comb, and counting the number of teeth the laser has swept through. We then interpolate between each comb tooth the scanning lasers crosses, and this gives an estimate for how far (in the frequency domain) the laser has swept. Note, that this gives a relative frequency estimate, and initial knowledge of the laser frequency is required for absolute frequency tracking (because we have no way of knowing at which comb tooth we started, we only know how many we have crossed).

Optical Setup

Our optical setup is rather simple, and schematically identical to that given in the Kippenburg paper (see below). In this measurement thus far, the comb used has been the MENLO comb, and the scanning laser has been the Agilent or CTL.

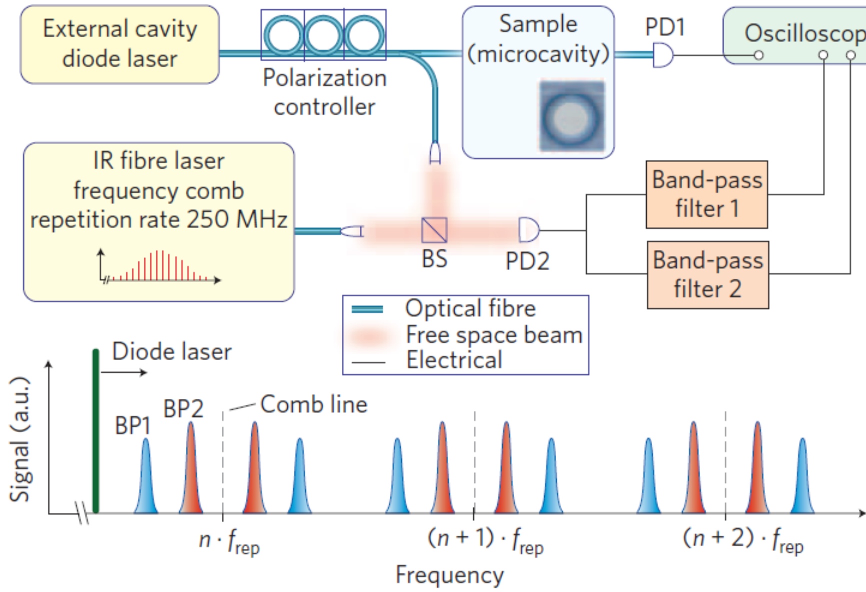

Figure 1: schematic of optical setup needed for the dispersion measurement, taken directly from the Kippenburg paper

RF Setup and Data Collection

For our setup, an ultra-fast lock-in amplifier from Zurich instruments (UHFLI) takes the place of the two bandpass filters and the oscilloscope. For signal from PD1, the auxiliary input for the lock-in is used, acting as a convenient way of recording the signal. For the beat note signal, it is introduced via channel one, and demodulated internally with the fundamental and fourth harmonic of the lock-in's internal oscillator (which we typically set to 25 MHz). This means we have two channels internal to the lock-in, one demodulating the signal from PD2 at 25 MHz, and the other demodulating at 100 MHz. What we save during data collection, is the magnitude of the signal at the 2 demodulation frequencies (so R in typical lock-in language), as a function of time. This means we see non-zero signal when the laser is $\pm$ the demodulation frequency away from a comb tooth, as illustrated in the bottom of Fig. 1. The operating principle, is to use the peaks in the demodulated signal (see bottom of figure one) to find out at which time the scanning laser cross the 4 areas of interest around a given comb tooth. This gives us four times per comb tooth, when we know exactly what the frequency of the laser is. We then take these 4 times and 4 frequencies, and interpolate between them (using a 5th order spline interpolation) to estimate the frequency for all times. We do this for each comb tooth, use the fact that comb spacing is well known, and we can build an accurate conversion between the time of the scan, and the frequency of the laser.

Measurement Performance and Uncertainty Estimation

Using this technique, we have demonstrated frequency uncertainty on the order of 10 MHz, for the entire duration of a 10 THz scan. This estimate is based on repeated measurements of a single chip under similar conditions, and represents the statistical uncertainty of the measurement. For a systematic uncertainty, the estimation is not as straight forward. Given the types of chips we have currently looked at (Pratham's, Alexander's, and Jae's) estimates of the GDD seem to be limited by structure in the FSR. This is to say, a nice linear dependence in the 1/FSR vs. Frequency is not seen, but rather a collection of mode crossings, and little local structure (that appears in each scan we take) obscures the desired signal. This seems to limit the measurement far more than the 10 Mhz standard deviations in the FSR. This means further improvements in the system are not necessary, but rather longer scans (so a more broadly tunable laser) are the best path for improved dispersion measurements. Plots below show the mode crossing and small structure present in the FSR, for two completed measurements, to show what I am talking about. For each of these data sets, data has not been binned and averaged, to help give a better feel for how the spread in peak locations look.

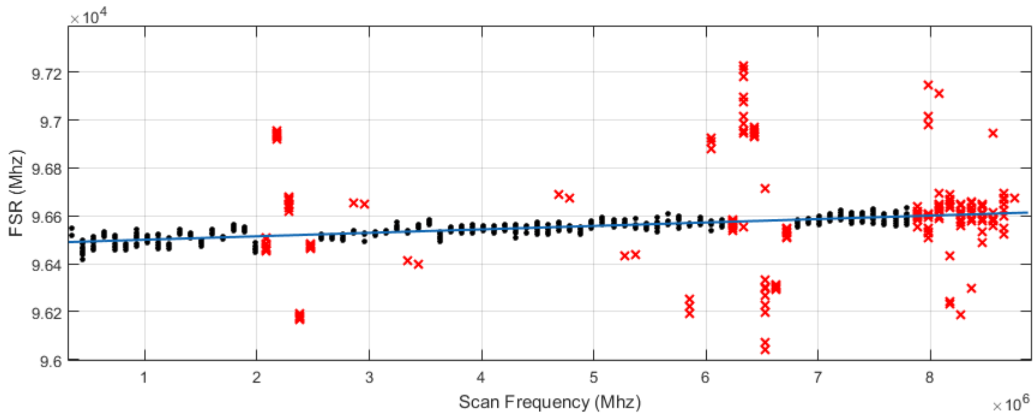

Figure 2: FSR vs. Scan frequency for Pratham's chip. This is the cumulative data, for 10 scans, showing the consistency of the small features, and large mode crossings. Red data was neglected in the linear fit (you must use at least some data exclusion to make the data sensible). In this specific case, it is hard to tell whether the chip has normal or anomalous dispersion

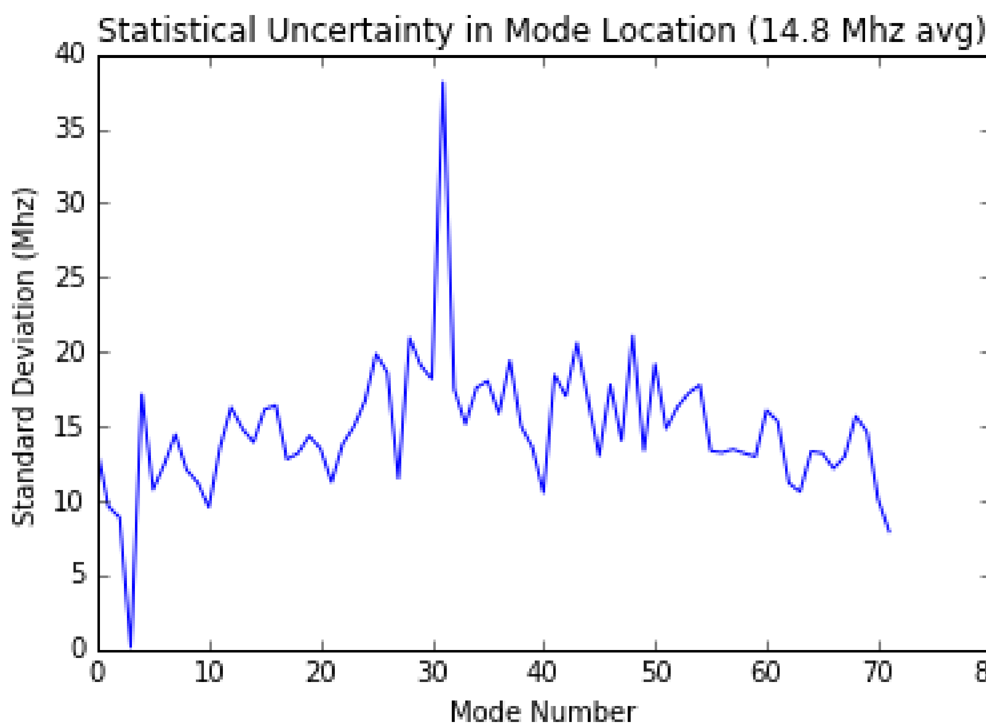

Figure 3: This represents the statistical uncertainty, in the estimation of each peak location for the same data set as above. This was manually processed so that the fourth peak was identically located in each data set, to account for uncertainty in the starting frequency of the scan (because the scan was started manually).

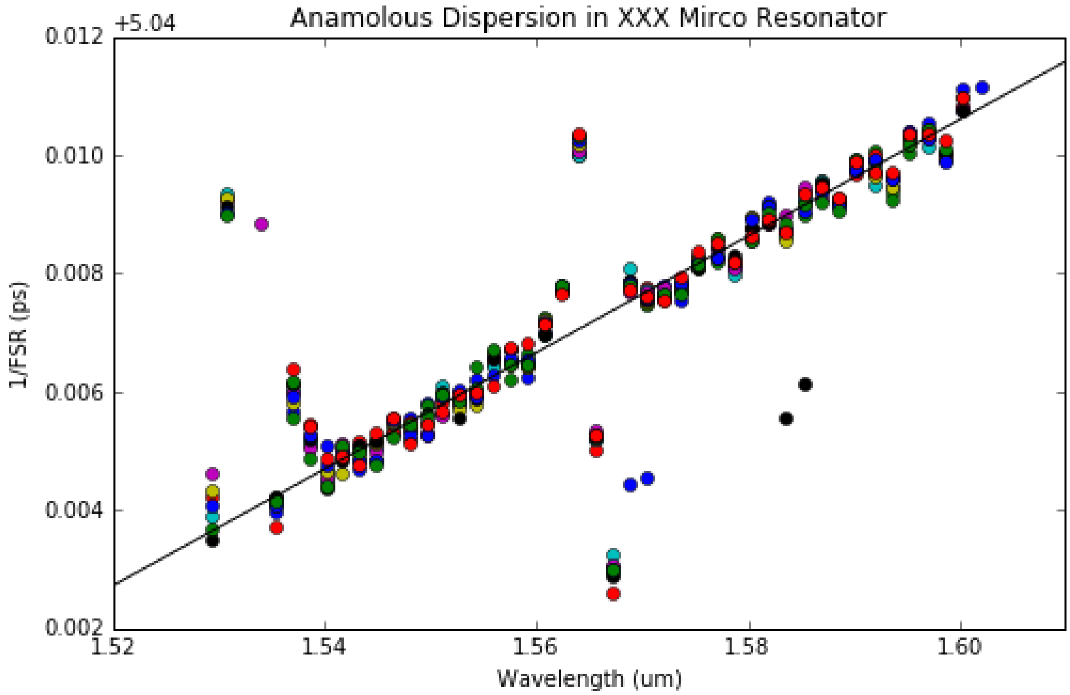

Figure 4: This shows the clearly anomalous dispersion, for one of Alexander's best comb generating chips. While the sign of the slope in this case is quite clear, small structure is still clearly apparent, as well as two mode crossings. This line is fit, with the data around each mode crossing masked.

With a basic feel for the idea behind this measurement, and it's demonstrated capabilities (thus far) I will now walk through an example measurement.

Quick Reference and Walk through

Hardware Set-Up

This is pretty dependent on the chip to be used, but for a general schematic refer to Fig. 1.

- Align the probe laser with your chip, with the laser tuned to the center of the sweep range. Take the transmitted light, align it onto a PD, and connect the output to the auxiliary port on the Lock-In Amplifier.
- Take a portion of the probe-laser, combine it with light from the frequency comb,

and illuminate the PD you will use to track the probe laser. This PD should be high sensitivity, at least 250 MHz, and needs to be connected to channel one of the Lock-In.

- On the control computer, run the Zurich instruments"labOne User Interface" from the start menu. This will open a dialog box, where you can select "open default GUI". With this GUI open, go to the scope option, and select the Freq Domain FFT mode. You should see a very stable beat note at 250 MHz (from neighboring comb teeth interfering) and a smaller beat note from the probe laser and the comb. This beat note should wander slightly, because the probe laser is not stabilized to this level. IN ORDER TO WORK PROPERLY, THIS BEAT NOTE SHOULD BE AT OR ABOVE 10^{-3}.
- Having confirmed the beat note with the comb is large enough, we are now able to setup the data collection.

Software Setup

The originizational idea behind the software in this setup, is for each user to use the code I wrote as a starting point and modify it as needed for their specific application.

- On the lock-in GUI, set the channel one oscillator frequency to 25 Mhz. You need to do this manually, because of a bug I did not have time to work through.
- Download your own set of the control software from my github repository ($https://github.com/rm3334/Dispersion\_Code$) and save to your own folder. This is very important, as each user will make modifications to this code as they see fit, and we don't want to contaminate the original files.
- Run "Jupyter Notebook", from the start menu. This is a browser based python GUI, which functionally feels a lot like Mathematica (individual blocks of code you run with shift+enter).

- After Jupyter starts, a browser tab will open. Use this tab to open all three files you downloaded.

Data Collection: Zurich Talk

This program is used to communicate with the lock0in amplifier, collect data, and initiate measurements.

- In the program, run the first two blocks of code. The first will initialize the program, and call all the libraries it will use (you always need to do this with python). The second block sets the parameters for the Lock-In. Note that the default parameters were optimized for the CTL scanning at 1500 pm/s, and different lasers/scanning speeds may require these number to be tuned.

- Before collecting data, you must tell the program how long it should take data for (scan duration) and the filename the data will be saved under (filename). note, that a new filename is needed each time, to avoid overwriting the data.

- With these values set, the lock-in will begin taking data, and immediately save the data, as soon as the third block of code is run. To take a scan, have some-one manually start the scan at the same time you start the data collection, by giving a short countdown. This has some uncertainty in when the scan starts, but this is not very important, and can be fixed by controlling the laser with the computer.

- This block will run (there will be a * instead of a number on the top left of the block of code) until the scan is completed, and the file has been saved. For long scans the files can be come very large, so this saving step can take several minutes.

Data Processing: Transmission-DashBoard

This program, take the transmission data, and the lock-in signal, to convert from transmission vs. time to transmission versus probe laser frequency.

- Initialize the program, and load the data you want to process.
- The third block of code will show what the transmission, and the 2 demodulation channels look like. If you do not see any resonances in the transmission data, something went wrong in data collection and it must be re-done.
- The next block, shows a zoomed in version of the lock-in signal, to help identify what threshold we should use to determine when the probe laser passes a comb tooth. Tuning this threshold is the most common thing that needs tuned. If the threshold is well below each peak you see, and well above the noise floor, you are fine.
- Run the remainder of this program, and look at the final graph. This shows the detrended scan frequency (so the deviations from a perfectly linear scan). Deviations should be less 1500 Mhz, and the graph should have both green and blue components (blue is the raw data, green is the interpolated data).

Data Analysis: AssistedPeakFit-Dashboard

This program takes the transmission vs. frequency data, filters the data as to focus exclusively on the resonant peaks, and then fits out each peak location. The peak fitting requires manual assistance, in order to uniquely identify separate mode families (if there are more then one present). This algorithm operates by taking your initial locations, then consecutively guessing where the next mode will be, and looking at small window ("fitsize") around the guessed location. If the algorithm is getting lost, try increasing this value. If it is getting confused and jumping between mode families, try decreasing this value.

- Enter the filename for the data you wish to process, and run the program until you reach the ”plots the smoothed data” section. This may take several minutes, as the filter is quite slow.

- At this point the filtered data will also be saved separately, if you wish to fit peak locations with your own algorithm.
- Using the mouse hover over, input the frequency of two consecutive modes in a shared family, from low frequency to high frequency (more details provided in the code).
- Repeat this for each mode family you wish to fit, and input that number into the ”modefamilia” variable.
- Run the remainder of the code, and view the results. If you have tried to fit too many peaks (peaks to fit) the code may freeze, and the last few peaks will be garbage.
- If you see discrete jumps in the FSR of 250 Mhz, this means a comb tooth was missed, and manually adjusting these points are warranted.

De-Bugging

This is an informal collection of errors I have seen, and how I fixed (or avoided) them.

Inconsistent Beat Note Size

For one set of data collection, I noticed that the size of the beat-note (so the PD2 signal after the lock-in) seemed to oscillating. This gave terrible performance, because I could not consistently tell where the beat note was. In the process of finding out why, it was discovered that a bad fiber splitter, was causing small reflections back into the scanning ECDL. This was not causing damage, but the slight feedback seemed to be causing frequency/power instabilities that made the measurement impossible.

Other issues I have seen, are likely attributable to the comb envelope not being flat. I know on my own MENLO comb (in Tanya Zelevinsky’s lab in the physics department) we have a lot of control over the envelope shape. I never saw this inconsistent beat note power to be an issue, but it may be in the future.

Limited Scan Range

The comb (in its current configuration) cuts off at around 1610 nm, so if you see a good beat note that dies towards the end of a scan, make sure that the comb is actually present in that range.

Beat-note Channels Give Different Results

Because one modulation channel is at 25 MHz, and the other is at 100 MHz, if you do note use a fast enough photo-diode selected for PD2, you see decreased response for the second channel. This is an easy fix, but it is good to keep in mind.

Frequently Missed Comb Teeth

This is most likely a symptom of improperly chosen threshold value in the ”transmission-dashboard” re-run the code with a slightly lower threshold value and see if performance improved. You would likely notice this, when the FSR in the final data product seems to have discrete 250 MHz jumps.

code crashes on transmission dashboard

Every once in a while, this program gets very confused about peaks coming to close together, and it will throw an error. This either means there is something wrong with a given scan (i.e. someone banged the table mid scan and the noise floor shot above the threshold), or the threshold value is too low. Try raising the threshold, and re-running the scan. If this doesnt work, and the data still seems fine, email me. This was always the most finicky part of the code.