

# Measuring Allan Deviation

## A guide to Allan Deviation with Moku:Lab's Phasemeter

Moku:Lab's Phasemeter measures phase with up to 6  $\mu$ radian precision for input signals oscillating between 1 kHz and 200 MHz. Using a digitally implemented phase-locked loop architecture, it provides exceptional dynamic range and precision far exceeding the capabilities of conventional lock-in amplifiers and frequency counters. Moku:Lab's Phasemeter calculates and plots Allan Deviation, which is a unitless measure of stability, typically used to quantify the stability of clocks and other oscillators. In this guide, we will cover the math and provide an example Allan Deviation calculation with the overlapped variable  $\tau$  estimator.



## Allan Deviation with overlapped variable $\tau$ estimators

Allan Deviation is unitless measure of stability, typically used to quantify the stability of clocks and other oscillators. By definition, Allan deviation  $\sigma_y$  is calculated as a function of observation time  $\tau$  by the following equation:

$$\sigma_y^2(\tau) = \frac{1}{2\tau^2} \langle (x_{n+2} - 2x_{n+1} + x_n) \rangle \quad (1)$$

where  $x$  is the phase measured over time. In practice, it's not possible to calculate the expected value over infinite time. Several estimators are used to calculate Allan deviation. Moku:Lab uses the overlapped variable  $\tau$  estimator to calculate the Allan deviation, which is defined as:

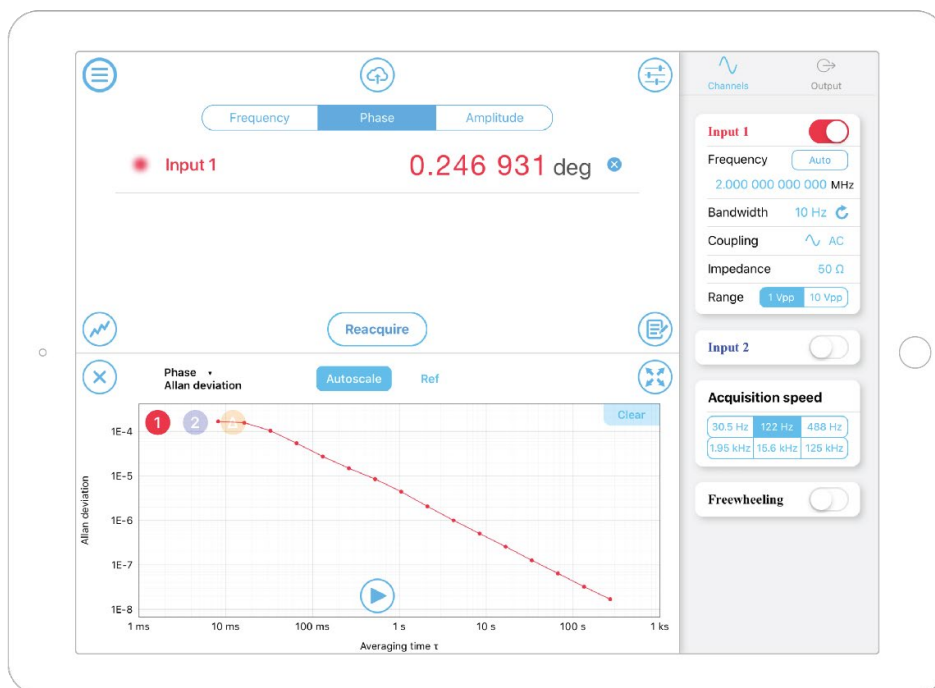
$$\sigma_y^2(n\tau_0, N) = \frac{1}{2n^2\tau_0^2(N-2n)} \sum_{i=0}^{N-2n-1} (x_{i+2n} - 2x_{i+n} + x_i)^2 \quad (2)$$

where  $\tau_0$  is the sampling period of the Phasemeter and  $N$  is the number of data points acquired for the input time series.  $n$  is an integer multiplier of the sampling period that best estimates the desired  $\tau \cong n\tau_0$ .  $x_i$  represents the  $i$ th element in the time series.

## Allan Deviation with Moku:Lab's Phasemeter

In this example, we measured the Allan Deviation of a 2 MHz signal with Moku:Lab Phasemeter. The frequency and phase of the signal were measured at 122 Hz for 10 minutes. To display the Allan Deviation, select the "Phase" tab on the top of the Moku display and "Allan deviation" in the plot area.

Please note if "Frequency" or "Amplitude" tabs are selected, Moku:Lab will use frequency or amplitude time series as the input for Equation (2).



## Python implementation

The function `cal_oadev` takes the time series, sampling rate and an array of observation time  $\tau$  as the input. It calculates the overlapped Allan Deviation using equation (2) and returns the  $n\tau_0$  and  $\sigma_y(n\tau_0)$  as arrays.

Required Libraries: NumPy, math

```
#Import libraries
import numpy as np
import math

def cal_oadev(data,rate,tauArray):
    tau0 = 1/rate #Calculate the sampling period
    dataLength = data.size #Calculate N
    dev = np.array([]) #Create empty array to store the output.
    actualTau = np.array([])
    for i in tauArray:
        n = math.floor(i/tau0) #Calculate n given a tau value.
        if n == 0:
            n = 1 #Use minimal n if tau is less than the sampling period.
        currentSum = 0 #Initialize the sum
        for j in range(0,dataLength-2*n):
            currentSum = (data[j+2*n]-2*data[j+n]+data[j])**2+currentSum #Cumulate the sum squared
        devAtThisTau = currentSum/(2*n**2*tau0**2*(dataLength-2*n)) #Divide by the coefficient
        dev = np.append(dev,np.sqrt(devAtThisTau))
        actualTau = np.append(actualTau,n*tau0)
    return actualTau, dev #Return the actual tau and overlapped Allan deviation
```

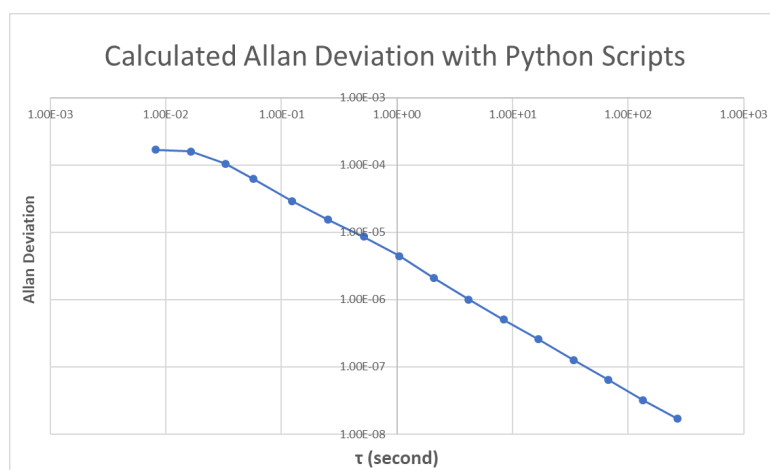
## Example calculation

In this example, we imported the time series we acquired previously on the Moku:Lab and calculated the Allan Deviation with Python scripts. You can find all the raw data and scripts at:

[https://liquidinstruments.com/s/19-1112\\_FileForAppNote-AllanDeviation-DataAndScripts.zip](https://liquidinstruments.com/s/19-1112_FileForAppNote-AllanDeviation-DataAndScripts.zip)

The time series can be found in the “data.xlsx” file. The first column “A” is the measured frequency in Hz, and the second column “B” is the measured phase in cycles. The script “CalculateAllanDeviation.py” reads the excel file and calls the `cal_oadev` function in the “AllanFunc.py”. The Allan Deviation is plotted as a function of  $\tau$  in log scales. The Allan Deviation plots matched well between the Moku:Lab and Python scripts.

Required Libraries: NumPy, math, pandas, Matplotlib



## Reference

- [1] Land, D. V., A. P. Levick, and J. W. Hand. "The use of the Allan deviation for the measurement of the noise and drift performance of microwave radiometers." *Measurement Science and Technology* 18, no. 7 (2007): 1917.
- [2] Allan, David W. "Statistics of atomic frequency standards." *Proceedings of the IEEE* 54.2 (1966): 221-230.
- [3] Howe, D.A., Allan, D.W., and Barnes, J.A. "Properties of signal sources and measurement methods." *Proceedings of the 35th Annual Symposium on Frequency Control* (1981): TN-14 – TN-60.

---

## Questions or comments?

Please contact us at [support@liquidinstruments.com](mailto:support@liquidinstruments.com)