Documentation for Repository: edlab\_automation

**INTRODUCTION**

The main part of the repository consists of 3 MATLAB scripts*.* Which means that the required functionality is provided by these three scripts:

FullOtomasyon.m

ForLoop.m

scheduler.m

There is also a data processing script named as:

MASTERCODE.m

By now you have probably noticed that the repository is split to 3 branches. The master branch uses the brightest points as measurement references, flicker\_robust branch is forked in order to improve noise robustness and overall reliability with some sacrificed precision. left\_side\_noise\_robust is a special fork, it was created for a special condition and is obsolete by now. This documentation is focused on flicker\_robust branch. **Consequently, you should use and follow the flicker\_robust branch.**

**BRIEF CODE BREAKDOWN**

FullOtomasyon.m is the core script that does the line measurement task. ForLoop.m is responsible of sweeping within 40 Hz range (+-20 Hz) and calling FullOtomasyon.m for measuring the line length. The resonant frequency (the frequency where the line length is at maximum) is also decided by ForLoop.m. scheduler.m is responsible of calling the FullOtomasyon.m hourly. ForLoop.m ensures that the generator frequency is matched with the resonant frequency hourly.

**You should always run scheduler.m for complete functionality**, bear in mind that it asks for an initial estimated resonant frequency and a bad estimation might cause the script to function improperly.

**PREREQUISITES, INPUT AND OUTPUT**

ThorCam application should have a clearly visible line without any noticeable noise in the live image since the code takes snapshots from ThorCam application.

**The scripts do not run with ThorCam application running.**

Output files are created hourly in the script directory with datestamps and resonant frequency as the name property. The script tries to run the codes at XX:05. A text file represents an hourly snapshot with swept frequency. Each line differs from the next by 1 Hz and the largest line distance is at the resonant frequency presented at the name section. Consequently, if we were to plot a single text file, we would get a graph similar to bell-curve. (Two peaks or abrupt changes often indicate errors in experiment setup.)

**THE UNDERLYING MEASUREMENT PRINCIPLE**

First and foremost, the image is filtered by a hard thresholding filter with 10 intensity value. This eliminates most of the weak noise.

The mean value of row numbers of the pixels that went through the previous filter are calculated. This usually gives the line that cuts the scanline into two parts vertically.

The brightest pixels are found, however the weak point of this approach lies here, since the maximum intensity value is 255 (8 bits), usually, there are several brightest points (the condition called saturation), the distance between the farthest two brightest points are taken as the ultimate result.

This measurement is made for a fixed frequency. Therefore, ForLoop.m is used to sweep the frequency and measure the distance for each frequency with 1 Hz sensitivity. (Total 40 different frequencies)

**DATA PROCESSING**

MASTERCODE.m is written to visually represent the experimentally acquired data scanline length, resonant frequency and quality factor are plotted against time for a dataset. Standard deviation and mean values are also calculated and data is also represented in histogram format.

Function prototype is MASTERCODE(myFolder,suppress) , **therefore the user should properly give the file address, suppression speeds up the data processing however omits the printing of some useful data, therefore most of the time you would want to run with suppress=0.**

If the bell curve shape is not present in the dataset, the code might fail to terminate properly due to quality factor.