1 Introduction

Electrical Flicker is the term used to describe the changes in the illumination intensity of light sources when subject to voltage fluctuation. It is related to the human perception of those changes in the illumination intensity. These changes are reported to cause an irritating effect on the human eye-brain system, referred to as light flicker (Baggini, 2008). It is common to denote flicker as a power quality indicator.

A standard, the IEC 61000-4-15 (IEC, 2010) has been created to indicate the characteristics of the Flickermeter, an instrument designed to measure the flicker level of a given voltage signal. The instrument described on the Standard is apt to detect the flicker perception level for all practical voltage fluctuation waveforms.

In this article, we aim to describe the Flickermeter’s digital implementation using MATLAB. All of the functions used in this apparatus’ digital implementation were designed by the authors, in order to simplify the code translation to other programming languages. To this extent, interesting techniques were used to convert the analog filters characterized in the standard to their digital form.

The need for discretizing the analog filters added a significant degree of complexity to the Flickermeter, and doing so for a variable sampling frequency was one of the main challenges of this work.

The digital Flickermeter was implemented through the common technique of dividing the apparatus in five blocks, each performing a specific task, as is described in the standard. These blocks were conceived as MATLAB functions, and its functioning is exposed in this article. Implemented user functions are cited and detailed in this article.

The main contribution of this article is to show the detailed digital implementation of a Flickermeter while also demonstrating how to perform the digitalization of its analog filters. This discretization is performed for any sampling frequency, which is also a relevant topic of this article. Although we can see some literature dealing with digital implementation of Flickermeters, mostly all of them exhibit an apparatus implemented with analog filters and without many details on the implementation process.

It is our goal with this work to expose a step-by-step view on the digital instrument, showing relevant graphs and outputs. This paper is organized as sections, each describing one of the blocks implemented according to the IEC Standard. There, we cite all the functions performed by each block and the relevant outputs in hope to provide more thorough reference for those who wish to implement a digital Flickermeter in any programming language.

Throughout the article we present the difficulties and contributions of this paper, and at the last section we offer a brief conclusion.
According to the IEC 61000-4-15, the first block of the Flickermeter implementation should be a voltage adapter circuit. Its purpose is to contain the internal voltage of the instrument inside a level which does not compromise the functioning of the rest of the apparatus. By scaling down the voltage to an internal reference level, flicker measurements can be made independently of the actual voltage level of the input, and may be expressed as a percent ratio.

This circuit must maintain the r.m.s. level of the output at a constant reference value while keeping the relative fluctuation. It is specified in the IEC 61000-4-15 that there should be a first order low-pass filter with time constant of 27.3 seconds to process the output of the voltage adapter circuit. The elements described in the Standard are necessary for a physical implementation of the instrument, but for a software implementation the scaling could be done through mathematical procedures (Fregosi et al., 2010). As it was our aim to carry out a generic Flickermeter, these blocks were implemented anyway.

To test this block, a signal with rectangular modulation described as 1 is created. Here we can modify crucial elements such as flicker frequency and percentage of voltage fluctuation, according to the Appendix B in the IEC 61000-4-15 Standard. The results exhibited in this article will be acquired considering a flicker frequency of 8.8Hz and a voltage fluctuation of 0.252%, for a 60Hz, 120V, rectangular modulated input. This input is shown in figure 2:

\[ u(t) = 1 \times \sin(2 \times \pi \times f \times t) \times (1 + v_f ... \times \frac{1}{2} \times \text{sign}[\sin(2 \times \pi \times f_{flicker} \times t)]) \]  

(1)

where

The digital implementation using MATLAB was based on (Macedo Jr. et al., 2011) with some important conceptual changes. The input voltage signal that enters the Flickermeter is analyzed with respect to the amount of zero crossings that it contains. This ensures that we can obtain a correct r.m.s. value even in the event of voltage magnitude change, which is a common circumstance when analyzing flicker.

After detecting each of the zero crossings in the signal, and thus “dividing” the signal in sections between the zero crossings, a r.m.s. value is calculated for each section of the input. This operation allows the block to obtain a reference voltage value that is independent in relation to variations in the input, hence certifying that the circuit operates as expected.

This r.m.s. signal calculated with relation to the input voltage is then passed through a low-pass filter with time constant of 27.3 seconds. For the purpose of implementing a digital filter
without any MATLAB-specific function, the discretization and filtering were both performed by user-created functions, which are described later on in this article. With this operation, we obtained the voltage reference signal.

As we are dealing with r.m.s. values, we must multiply by $\sqrt{2}$ in order to have a proper reference voltage. This reference changes alongside the input voltage, and when we acquire our adapted voltage signal as the output of block 1, it reflects these changes, thus enabling the voltage fluctuation to be measured and assessed in the following blocks of the instrument. It can be seen that the reference voltage takes a long time to stabilize in a value, due to the high time constant of the low-pass filter. While the usage of the filter is required by the Standard, it inserts a difficulty.

![Figure 3: Reference voltage signal with the stabilization delay](image)

We acquire the output of the block 1 through dividing the input by the reference value calculated. This gives us a voltage signal in p.u. which reflects the voltage fluctuation of the input in a more adequate voltage level for the rest of the circuit. As the reference voltage takes long to stabilize, the output from block 1 is also affected by this time. Thus, for the first minute we have very high values in the output as the input voltage is divided by a small reference value. A solution for escaping this issue is to simply ignore the yielded signal until it is stabilized.

Figure 5 shows a simplified take on Block 1’s operation.

![Figure 5: Simple diagram to simplify block 1’s operation](image)

This results in the extraction of the modulating amplitude in the block 2’s input.

![Figure 6: Block 2’s output, the squared output from block 1](image)

4 Block 3

Block 3 consists on three sets of filters. The initial filters are designed to remove both the high and low frequency components, and the third filter provides an output that is related to the human sensitivity to flicker (White and Bhatcharya, 2010).

The first filter implemented is a first order high pass filter, with cutoff frequency of 0.05Hz, as stated in the Standard. The objective of such filter is to remove the DC component from the block.
3’s input. The discretization and filtering were both performed by the already cited user functions “c2d_bilinear” and “difference_equation”. The result of this filtering operation, then, passes through a sixth order low pass Butterworth filter. The purpose of this filter is to remove the 120Hz component inserted in the signal by the squaring performed in block 2. The filter is specified by the Standard with a cutoff frequency of 42Hz. The discretization of such filter was not easily performed, as is described later in this article.

The combination of both high pass and low pass filters previously described yields in a band-pass filter, which was separated in order to ease the understanding and implementation by those following the IEC 61000-4-15. The next filter is the weighting filter, which has a response that reflects the human perception of flicker and is centered around 8.8Hz, where the flicker sensitivity is at its highest. Its transfer function is defined by the Standard, and is shown:

\[ F(s) = \frac{k\omega_1 s}{s^2 + 2\lambda s + \omega_1^2} \times \frac{1 + s/\omega_2}{(1 + s/\omega_3)(1 + s/\omega_4)} \]  

(2)

Its parameters are also cited in the standard, and are available for consultation there.

The output of block 3 is thus a signal that oscillates on the flicker frequency, with the voltage fluctuation being weighted in such manner that only the flicker portion of the voltage signal becomes relevant in the output of block 3.

If we compare the output of block 3 with its input, we can see that the frequency of the signal yielded by block 3 is the same as the flicker frequency. We show this comparison in 10 by adding an offset of 1 to the output and doubling it’s magnitude, just for exhibiting in a more clear way this notable result.

The operation of block 3 is shown in the diagram exposed in Figure 11.

4.1 Implementing the Butterworth 6th Order Low-pass Filter

In order to solve the implementation issues commented before for the Low-pass 6th order Butter-
worth, the filter was designed and realized in a parallel architecture which is described in this section. Initially, the filter was projected in a conventional manner, using the Butterworth filter equations, giving the following transfer function:

\[ G(s) = \frac{K D(s)}{D(s)} \quad (3) \]

where

\[ D(s) = (s^2 + a_{1.1}s + a_{0.1})(s^2 + a_{1.2}s + a_{0.2}) \ldots (s^2 + a_{1.3}s + a_{0.3}) \quad (4) \]

And the coefficients are described in the table:

| \(a_{1.1}\) | 509.80364060344 | \(a_{1.3}\) | 136.60147379814 |
| \(a_{0.1}\) | 696.939.92865408659 | \(a_{0.3}\) | 696.939.92865408650 |
| \(a_{1.2}\) | 373.20216680530 | \(a_{0.2}\) | 337734131319898.1 |
| \(a_{1.3}\) | 107.734131319898.1 |

Table 2: Parameters of the low-pass filter’s transfer function

It can be noticed that the filter has a very high gain, which can yield in numeric order issues. If we implement the filter as the transfer function 3, we find an unstable filter due to these issues, and its output is of no use for the Flickermeter.

The resolution to this problem consists in rewrite the filter in a manner that we have the same transfer function at the end, but involving coefficients of lesser magnitude. This strategy consists in writing the original transfer function in partial fractions:

\[ G(s) = \frac{A_1 s + B_1}{s^2 + a_{1.1}s + a_{0.1}} + \frac{A_2 s + B_2}{s^2 + a_{1.2}s + a_{0.2}} + \ldots \frac{A_3 s + B_3}{s^2 + a_{1.3}s + a_{0.3}} \quad (5) \]

with coefficients shown in the table 3.

| \(A_1\) | 696.4047240061 | \(A_3\) | 107.73418573360 |
| \(B_1\) | 256899.7519725224 | \(B_3\) | -40206.63155478315 |
| \(A_2\) | -804.138697397 | \(B_2\) | -150053.191763562 |

Table 3: Parameters of the low-pass filter’s transfer function in partial fractions

The filtering was performed by applying each of these transfer functions as a filter and adding the three outputs, which resulted in a stable filter and a reasonable output for the low-pass filter. Hence, we obtained a perfectly manageable filter gain with respect to the numeric errors.

5 Block 4

Block 4’s main function is to simulate the “non-linear eye-brain perception”, as stated in the Standard. It also provides a way of simulating the storage effect in the brain, and thus is crucial in determining the flicker sensation.

This block accomplishes its function by squaring the input signal (the output from block 3) and passing the resulting signal in a low-pass filter with time constant of 300ms. This yields one of the main measurements of the apparatus, the instantaneous flicker sensation, \(P_{\text{inst}}\).

The IEC 61000-4-15 Standard provides two sets of tables with specified test points, indicating flicker frequencies and percentage voltage fluctuations which provides a \(P_{\text{inst}}\) value around 1p.u., ±8%. These sets of test points were used for verifying the accuracy of the digitally implemented Flickermeter.

As for each test point the result must be around 1p.u., it can inferred that the low-pass filter must be such that it yields the expected results for all test points. Thus, the gain was calculated by dividing the expected result (1p.u.) by the result obtained with an unitary gain filter. This gives us a filter gain in the order of \(10^6\).

![Figure 12: Block 4's output for \(f_{\text{flicker}} = 8.8Hz\) and \(v_f = 0.252\)](image)

The output at Figure 12 shows a transient stage, which comes from the two long-time-constant filters (White and Bhattacharya, 2010). The measurement of \(P_{\text{inst}}\) values should start at the moment when the signal stabilizes itself, after around 150 seconds.

The signal \(P_{\text{inst}}\) may contain errors when compared to what is actually experienced by humans. Due to the high amount of variables that determine the flicker sensation, it is virtually impossible to achieve an absolutely correct measurement of the flicker stimuli (Wiczynski, 2012).
6 User Created Functions

In this section, we aim to comment about the user created functions built for the digital implementation of the Flickermeter. They were created with the purpose of making it easy to translate the implementation from its original language in MATLAB to other programming languages.

6.1 Discretization Function

The discretization of the analog filters specified in the IEC 61000-4-15 Standard could be done through several different manners. The digital implementation described in this article was performed using MATLAB, there are several functions already present in the environment. Although the Standard doesn’t specify the need to a generic function to be implemented, it was defined by the authors as a condition that the digital implementation should be translated to any other programming language without the loss of usability. Hence, a new function was created, called “c2d_bilinear”, that performed the discretization of the analog filter’s transfer function. Thus, the digital implementation reported in this article is platform independent.

The “c2d_bilinear” function uses coefficient vectors as a representation for the transfer functions, as does the original “c2d” function from MATLAB. It is also able to discretize a transfer function for a generic sampling frequency, which contributes to the versatility of the implementation and is one more user defined variable.

We used the bilinear transform, or Tustin method, for discretizing the transfer functions. In this method, the mapping is done using an approximation for the ideal mapping (ZOH), and it is valid for plant cutoff frequencies smaller than twice the sampling frequency. For a cutoff frequency of 60Hz, for example, a sampling frequency of 1kHz would produce a cutoff frequency deviation of approximately 0.03%, thus reaching a high level of reliability.

6.2 Filtering Function

All the filtering performed in the digital implementation of the Flickermeter was done through the user created function “difference_equation”. While MATLAB counts with several built-in filtering functions, it was already stated that one of the objectives of this implementation was to be translatable to other programming languages.

Detailing the process of filtering a signal through the application of the filter’s difference equation escapes the scope of the article and thus will not be approached. The “difference_equation” function calculates the filtered vector with successive loops for each element of the output.

7 Conclusion

Through the thorough description of each block defined by the IEC Standard 61000-4-15, we supply anyone wishing to implement a digital Flickermeter with a detailed guide to do so. The Flickermeter is a crucial instrument used in power quality analysis.

The differential of this article, its approach to a digital Flickermeter through user created functions for filtering and discretizing the defined filters with a variable sampling frequency, is described through this paper.

With this work we obtained a digitally implemented Flickermeter, able to be translated to any programming language without the loss of usability. Voltage signals can be inserted in this apparatus, regardless of the sampling frequency which were obtained, and an instantaneous flicker sensation, $P_{inst}$, will be acquired, providing an accurate measure of power quality.

References


Fregosi, D., White, L. W., Green, E., Watterson, J. and Bhattacharya, S. (2010). Digital flickermeter design and implementation based on IEC standard.


