IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 June 2022, accepted 21 August 2022, date of publication 31 August 2022, date of current version 16 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3203187

RESEARCH ARTICLE

Sensing Frequency Drifts: A Lookup Table Approach

GIUSEPPE AVON[1], MAIDE BUCOLO[1,2], (Senior Member, IEEE), ARTURO BUSCARINO[1,2], (Senior Member, IEEE), AND LUIGI FORTUNA[1,2], (Fellow, IEEE)

[1]Dipartimento di Ingegneria Elettrica Elettronica e Informatica, University of Catania, 95124 Catania, Italy
[2]IASI, Consiglio Nazionale delle Ricerche (CNR), 00185 Rome, Italy

Corresponding author: Luigi Fortuna (luigi.fortuna@unict.it)

This work was supported in part by ENI within the ENI-CNR Joint Research Agreement.

ABSTRACT Real-time frequency analysis is a task commonly implemented in last generation signal processing devices. Even if the reliability of these systems is widely recognized, detecting the presence of drifts in the frequency of analog signals is still an open problem. Besides being simple in principle, it needs a suitably long-term memory in order to determine not only in which direction the frequency drifted but also along which specific path. In this paper, we propose a novel strategy based on the concept of jump resonance in nonlinear forced oscillators to determine a digital signal processing paradigm hinged on a lookup table able to provide reliable and real-time information on complex paths of frequency drifts.

INDEX TERMS Frequency drifts, frequency sensors, jump resonance, lookup table.

I. INTRODUCTION

Real analog signals are often characterized by time-varying frequencies, as the result of complex dynamical processes. Practical examples of analog signals which are subjected to frequency drifts are the power grids mains and the heartbeat rate. In power distribution networks, the nominal main frequency of the carrier is known (50 Hz in Europe, and Asia, 60 Hz in the Americas) but its drifts, which are known to deteriorate the efficiency of the grid [1], and the sequences of increments/decrements are not known, even if some statistical considerations can be drawn [2]. The stability of the main frequency with respect to users interactions is a fundamental property to ensure the highest efficiency for the grid. To this aim, detecting in real-time frequency drifts would allow for suitable control actions, whose efficiency is linked on the drift amplitude and direction [3].

The concept of frequency drift can be further generalized by considering either signals with complex spectral properties in which one or more frequency components are subjected to variations in time, or time-series of different nature characterized by hidden time-varying periodicity, such as economic time-series [4] or pandemic related quantities [5].

The necessity to oversee frequency drifts, however, goes beyond the merely technological field. Let us consider the monitoring of the heartbeat or other biomedical quantities whose frequency is a key indicator to discriminate pathological states. As concerns the human heartbeat rate, in fact, a fundamental diagnostic information to prevent diseases is gained monitoring its variability [6], which may also occur through complex paths of increasing/decreasing drifts.

Detecting frequency drifts is a non-trivial problem from the computational point of view, since it needs to determine the main component of the frequency spectrum of the input signal and to store a significant number of samples to infer the trend. Modern digital signal processing (DSP) units allows to quickly determine the frequency properties of analog signals [7], but this feature alone is not sufficient to efficiently infer frequency drifts and complex paths of frequency drifts.

In order to provide a simple but effective solution to the problem of real-time detection of frequency drifts either increasing or decreasing, a the theoretical concepts adopted to design a device specifically intended to this aim [8] will be coupled to a fast and efficient algorithm based on the construction of a Lookup-Table (LUT).

LUTs are a common way to trade computational power with storage space. LUTs are based on replacing runtime computation by a n-dimensional indexing operation [9], [10].

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

Values are, therefore, taken by addressing stored data instead of running computing-intensive operations. These tables reduce system dynamic behavior to a set of precalculated points, called breakpoints. Differently from the common software-development related interpretation, here the term breakpoint represents the input-output mapping of the system.

From a simple mathematical point of view, LUTs express an enforced relationship between two or more variables, represented as an array or n-th order storage structure which can be implemented as arrays, matrices, or generic n-dimensional structures. Main advantage can be appreciated from a computational point of view, as the LUT cyclomatic complexity [11] can be considered an $O(1)$ (data-size independent complexity). LUTs are generally placed in RAM/ROM and can be hardcoded or computed during a specific phase of the program, which commonly happens during the startup phase. The latter technique, called memoization, is used to shorten the runtime of applications by keeping storage result of previous executions for future reuse [12]. LUT can be also found in specialised hardware, like the FPGAs, often provided as building blocks from common vendors. Their size varies depending on the hardware technical specifications [13].

LUTs and their derivations or specialisations can find applications in almost all fields, ranging from hardware implementations [14] too software algorithms [15]. Their employment is common when processing power is limited or under energy efficiency constraints [16]. As the processing power is only required once at design, compile time or during the memoization phase, this reduction plays a key-role in energy efficiency [17]. LUTs comprise a large percentage of the overall execution of the applications in many automotive applications (10%-20%) and represent a consistent part of the scientific code. Generally speaking, implementing a LUT is a common solution to improve math intensive code.

Main disadvantage of LUTs is memory usage, which is proportional to the number of elements stored. This is due to the fact that the table must be fully held in memory, together with control structures. Therefore, table breakpoints accuracy is determined by a trade-off between the correct choice of the distance between pre-calculated points and the problem of minimizing the LUT size. This must be performed accounting for the error introduced [18]. Moreover, evolution and steady state of the context using the LUT may settle into a subset of the stored values, leading to a waste of a portion of memory for the unused values [19].

The approach described in this paper aims at proposing a novel and efficient strategy to detect and discriminate complex patterns of frequency drifts, either increasing and/or decreasing in a given time-window. The proposed solution is based on a peculiar behavior of a class of driven nonlinear systems, i.e. the so-called jump/multijump resonance systems [20], [21]. These circuits and systems, in fact, when subjected to an input signal with drifting frequency produce an output whose amplitude shows sudden jumps at particular frequencies depending also on the direction of the frequency drift, thus creating an hysteresis in the frequency response. Moreover, the amplitude of the input signal plays a crucial role, since it must be rescaled to a precise value in order to elicit the jump resonance behavior, otherwise, as observed in [21], chaotic oscillations may arise.

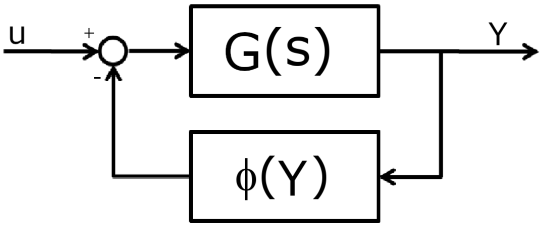


FIGURE 1. **Lur'e form feedback scheme: the linear part is expressed by the transfer function $G(s)$, while the polynomial nonlinearity $\phi(Y)$ acts on the feedback.**

The device described in [8] is based on the digital implementation of a discrete-time non-autonomous nonlinear system with cubic polynomial nonlinearity, whose frequency response displays the jump behavior. It has been shown its efficiency also when the input analog signal contains more harmonics, but the device allows to discriminate drifts increasing or decreasing from the nominal value of the frequency. The approach presented here is intended to improve the efficiency of the frequency drift detection, thus discriminating among more complex patterns of increasing/decreasing drifts and consists in the implementation of a LUT based on the peculiar behavior of jump resonance sensibly reducing both the computational needs and the memory storage. A specifically designed windowed approach allows to detect not only the single frequency drift, but also nested paths of deviations, either increasing or decreasing, thus fully implementing the hysteretic behavior of multijump resonance systems. Moreover, the jump resonance paradigm to design the LUT guarantees the possibility to cope with signals in which direction and amplitude of frequency drifts are not known a-priori.

The paper is organized as follows. In Sect. II the basic elements and the design guidelines of jump/multijump resonance systems are outlined, in Sect. III the windowed approach to design and implement the LUT is presented. The experimental setup obtained implementing the LUT in hardware is discussed in Sect. IV, while the experimental campaign showing the capabilities of the LUT to detect nested complex path of frequency drifting is presented in Sect. V, together with an applicative example aimed at showing the detection of frequency drifts in the mains frequency of Continental Europe power grid. A comparative analysis which allows to determine the main advantages of the proposed approach with respect to different solutions is reported in Sect. VI. Conclusions are drawn in Sect. VII.

II. MULTIJUMP RESONANCE SYSTEMS

Let us consider a nonlinear system in which the linear and nonlinear parts can be separated and organized in a feedback scheme, as shown in Fig. 1 where the linear part is represented

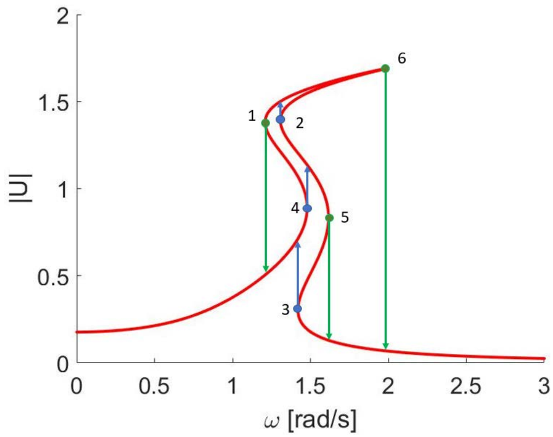


FIGURE 2. **Multijump resonance in the frequency response of a forced nonlinear oscillator with linear part as in Eq. (1) and feedback nonlinear function as in Eq. (2). The curve has been calculated for $K = 1$, $\xi = 0.03$, $\omega_0 = 1$ rad/s, $\bar{A} = \frac{192}{105}$, $B = -\frac{32}{5}$, $C = \frac{19}{4}$, and $R = 0.2$. The points labeled from 1 to 6 identifies the jump frequencies.**

by a transfer function $G(s)$ and the nonlinear block $\phi(Y)$ acts only on the feedback loop. This representation of nonlinear feedback systems is referred to as Lur'e form [22].

We focus now on the case in which the linear part is a generic second-order transfer function of the type

$$G(s) = \frac{K}{\frac{s^2}{\omega_0^2} + \frac{2\xi}{\omega_0}s + 1} \tag{1}$$

characterized by a gain K, a natural frequency ω_0 and a damping factor ξ. Furthermore, let us consider for the nonlinear part a septic odd polynomial nonlinearity of the form

$$\phi(Y) = \bar{A}Y^7 + \bar{B}Y^5 + \bar{C}Y^3 \tag{2}$$

where $\bar{A}$, $\bar{B}$, $\bar{C}$ are suitable gains. The system, thus, can be considered similar to a Duffing oscillator where a higher order odd polynomial nonlinearity has been introduced.

When subjected to an external sinusoidal driving input $u(t) = R\sin(\omega(t)t + \psi)$ with time-varying frequency $\omega(t)$, the output of such system is a nonlinear oscillation whose amplitude at time $\bar{t}$ depends not only on the instantaneous frequency $\omega(\bar{t})$ but also on the values assumed by $\omega(t)$ for $t < \bar{t}$. An example of multijump frequency response is reported in Fig. 2, it can be observed that for a given range of ω, the frequency response is a multi-valued function.

The multijump resonance curve in Fig. 2 contains multiple hysteresis windows nested in the frequency interval $\omega_1 < \omega(t) < \omega_6$, where ω_1 and ω_6 are the frequencies at which the jumps marked with label 1 and 6 occur. In fact, if the frequency of the input signal is swept upwards from values lower than ω_1, the frequency response of the system follows the lower branch up to $\omega(t) = \omega_4$ where a sudden jump up in the amplitude occurs. Now two cases can be discriminated: if the frequency continues to grow, the response will undergo a jump down at $\omega(t) = \omega_5$, conversely if $\omega(t)$ decreases a jump up at $\omega(t) = \omega_2$ can be retrieved in the output amplitude. In this latter case, the upmost branch can be visited either for increasing values of $\omega(t)$, leading to a jump down at $\omega(t) = \omega_6$, or for decreasing values of ω to a jump down at ω_1. Finally, if the frequency of the input signal is swept down from a value larger than ω_6, a single jump up at $\omega(t) = \omega_3$ is observed in the output of the system.

It appears evident that such kind of resonance, and its hysteretic behavior, allows to discriminate among complex paths of increments/decrements of the frequency of the given input signal by simply inspecting the jumps occurring in the amplitude of the output signal. Our approach is based into transferring such behavior, suitably rescaling the frequency ranges, into a simple and effective LUT.

To design the linear and nonlinear parts of the feedback oscillator, so that jumps around a specific value of the frequency, it is useful to adopt an approach based on the describing function method [23]. The method is based on two hypotheses: (i) the system variables can be approximated as the sum of a finite number of sinusoidal terms, and (ii) a low-pass filtering effect occurs in the loop. Provided that these hypotheses are verified, the nonlinear term can be approximated by a function of the amplitude of its input $N(U)$. Notwithstanding the effective approximation introduced by this method, the describing function approach proved to be effective also in presence of higher harmonics, including chaotic oscillations [24].

Considering the odd nonlinearity $\phi(Y)$ as in Eq.(2), the corresponding describing function depends only on the amplitude U of its input signal [25], as

$$N(U) = \bar{A}\frac{105}{192}U^6 + \bar{B}\frac{15}{24}U^4 + \bar{C}\frac{3}{4}U^2 = AU^6 + BU^4 + CU^2 \tag{3}$$

in which we have defined $A = \frac{105}{192}\bar{A}$, $B = \frac{15}{24}\bar{B}$, $C = \frac{3}{4}\bar{C}$. Referring to the feedback scheme reported in Fig. 1, the signal $u(t) = R\sin(\omega t + \psi)$ is a solution for the closed-loop system if it occurs:

$$N(U)U + G^{-1}(j\omega)U = Re^{-j\psi} \tag{4}$$

Assuming $\psi = 0$, and indicating with $\Re(\omega)$ and $\Im(\omega)$ the real and imaginary part of $G^{-1}(j\omega)$, the modulus of Eq. (4) can be written as:

$$(AU^7 + BU^5 + CU^3 + \Re(\omega)U)^2 + (\Im(\omega)U)^2 = R^2 \tag{5}$$

and thus rearranging terms

$$\begin{aligned} A^2U^{14} &+ (2AB)U^{12} + (B^2 + 2AC)U^{10} \\ &+ (2BC + 2A\Re(\omega))U^8 + (C^2 + 2B\Re(\omega))U^6 \\ &+ 2C\Re(\omega)U^4 + (\Im(\omega)^2 + \Re(\omega)^2)U^2 - R^2 = 0 \end{aligned} \tag{6}$$

that, defining now $X = U^2$, it can be written as

$$\begin{aligned} p(X) = A^2X^7 &+ (2AB)X^6 + (B^2 + 2AC)X^5 \\ &+ (2BC + 2A\Re(\omega))X^4 + (C^2 + 2B\Re(\omega)))X^3 \\ &+ 2C\Re(\omega)X^2 + (\Im(\omega)^2 + \Re(\omega)^2)X - R^2 = 0 \end{aligned} \tag{7}$$

Since jump resonance occurs when the frequency response is a multi-valued function of ω, in order to determine the conditions on the parameters to obtain a frequency response shaped as in Fig. 2, it necessary that for a range of ω Eq. (7) admits seven positive real solutions.

Numerical procedures to determine the values of A, B, C, and R, as well as the suitable $G(s)$, leading to this condition can be outlined starting from the principle of polynomial identity. We can, thus, fix the seven positive real solutions to define the corresponding polynomial and solve a system of eight nonlinear equations in six unknowns. A different way is to search for the limit conditions where two pairs of coincident solutions satisfy Eq. (7), thus imposing that the ratios between the polynomial $p(X)$ in (7) and the second order polynomials $p_1(X) = X^2 - 2a_{L,1}X + a_{L,1}^2$, $p_2(X) = X^2 - 2a_{L,2}X + a_{L,2}^2$ have a remainder equal to zero. Therefore it follows:

$$\frac{p(X)}{p_1(X)p_2(X)} = q_1(X) + r_1(X) \tag{8}$$

where the quotient is a third order polynomial on which the Cardano conditions [26] can be applied to guarantee three positive real solutions.

Once retrieved suitable values of the parameters of the nonlinear function, $\Re(\omega)$ and $\Im(\omega)$ can be used to determine the parameters of the linear part ξ and K, while ω_0 is fixed according to the nominal frequency of the signal under investigation. Without lack of generality, we can adopt the following values: $K = 1$, $\xi = 0.03$, $\omega_0 = 16 \cdot 10^4$ rad/s, $A = 1$, $B = -4$, $C = 4.25$, and $R = 0.2$, obtaining the curve reported in Fig. 3, where the jump frequencies are: $f_1 = \frac{\omega_1}{2\pi} = 25$ kHz, $f_2 = \frac{\omega_2}{2\pi} = 28$ kHz, $f_3 = \frac{\omega_3}{2\pi} = 30$ kHz, $f_4 = \frac{\omega_4}{2\pi} = 33$ kHz, $f_5 = \frac{\omega_5}{2\pi} = 37$ kHz, $f_6 = \frac{\omega_6}{2\pi} = 47$ kHz.

The implementation of a frequency drift sensing device based on jump resonance can be obtained by discretizing the nonlinear system on a suitable digital microcontroller unit (MCU), as shown in [8], where a cubic nonlinearity has been considered. However, since the purpose here is to detect complex paths of frequency drifts, either increasing or decreasing or a combination of them, still preserving the efficiency of the method, the higher order nonlinearity scenario outlined above must be implemented following an innovative paradigm based on the construction of a LUT based on the shaping of the frequency response.

III. LOOKUP TABLE DESIGN

In order to implement the LUT and assess its performance on a dedicated hardware platform, a first phase of value extraction was accomplished. The septic response curve, generated using the aforementioned approach has a characteristic behaviour along the axis. As the assessment is implementation-oriented, the septic curve is simplified into a set of piecewise linear functions, where steps happen on the septic jumps.

The curve reported in Fig. 4 shows in green, superimposed to the septic curve, the four piecewise intervals linearized following the original curve values. The choice to linearize the intervals was done to keep the whole LUT essence as simple as possible and highlight, instead, the jump behaviour modelling. The extraction of the values is simply obtained by using a straight line between two points and stepping them by a reasonable amount in order to minimise MCU memory footprint. However, in the proposed approach, an important concept which is referred afterwards as *indexer function* can be used to model more complex scenarios, where a large number of steps is required or a more complex pattern has to be implemented, instead of simple memory arrays.

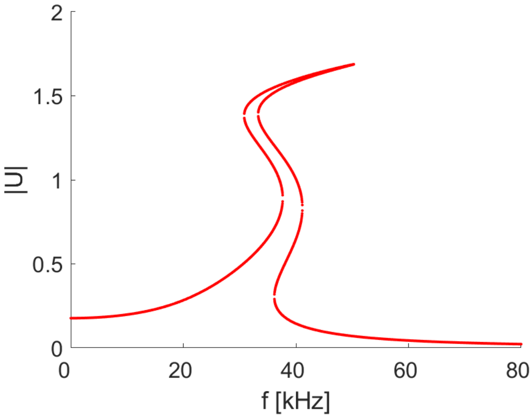


FIGURE 3. Multijump resonance in the frequency response of a forced nonlinear oscillator with linear part as in Eq. (1) and feedback nonlinear function as in Eq. (2). The curve has been calculated for $K = 1$, $\xi = 0.03$, $\omega_0 = 16 \cdot 10^4$ rad/s, $A = 1$, $B = -4$, $C = 4.3$, and $R = 0.2$. The jump frequencies are: $f_1 = \frac{\omega_1}{2\pi} = 25$ kHz, $f_2 = \frac{\omega_2}{2\pi} = 28$ kHz, $f_3 = \frac{\omega_3}{2\pi} = 30$ kHz, $f_4 = \frac{\omega_4}{2\pi} = 33$ kHz, $f_5 = \frac{\omega_5}{2\pi} = 37$ kHz, $f_6 = \frac{\omega_6}{2\pi} = 47$ kHz.

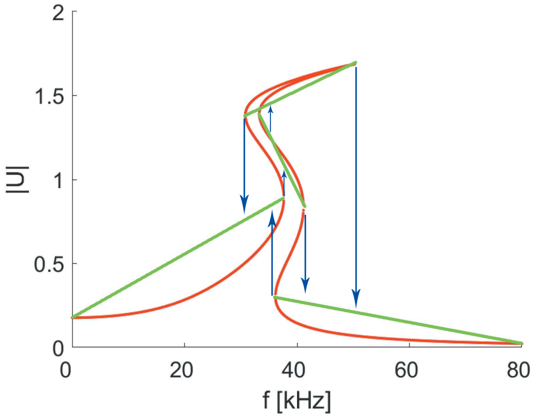


FIGURE 4. Piecewise linearization of the frequency response in Fig. (3) for the definition of the LUT windows.

Each of the piecewise approximation parts is afterwards referred as *frame* which has two boundaries, i.e. the maximum and minimum x-coordinates for the segment.

Notice that the intervals are not strictly contiguous and there are overlapping parts. This is the key to the jump mechanism implementation and to the hysteresis behavior, that is translated into a current frame pointer. The current frame pointer, which models the state of the LUT, is the discriminant for the jump management logic. A query (lookup) to the table is matched on the boundaries first, to discern if a jump is needed. In the occurrence of a jump, the logic is recursively applied to each of the frames, by moving the current frame pointer. Once the queried value falls in between the boundaries, the algorithm falls back to standard LUT

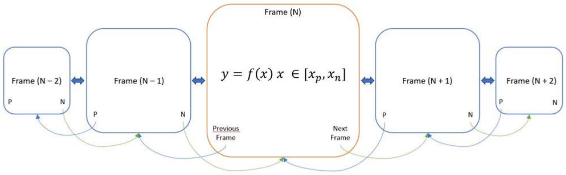


FIGURE 5. Simplified overview of the LUT segmentation.

behaviour. A detailed description of the implementation and logic is presented afterwards.

The proposed LUT is based on a windowed approach. Windowing essentially means splitting the table which, as previously explained, matches the piecewise linearization of the table portions. The whole table is split in N frames, each one having its characteristic values, indexed with an indexing function. Each frame is essentially a LUT by itself, which also carries additional data, i.e. the boundaries, and the jump information metadata, which are the pointers to the neighbouring tables.

A simplified overview of the LUT segmentation is shown in the Fig. 5. Each i-th frame has its domain ($x_p^{(i)}$ and $x_n^{(i)}$)and the corresponding values (shown as a function $f_i(x)$). The whole lookup is managed by a lookup manager which takes the queries to the table, submitting them to the current frame. By checking the current frame k against the two boundaries ($x_p^{(k)}$ and $x_n^{(k)}$), depending on the value the manager can

- take the queried value x_q from the current frame if $x_p^{(k)} \leq x_q \leq x_n^{(k)}$;
- jump to the previous frame and re-run the query if $x_q < x_p^{(k)}$;
- jump to the next frame and re-run the query if $x_q > x_n^{(k)}$.

The query re-run step implicitly means moving the current frame pointer. Using this approach, the hysteresis behaviour is implemented. When a new frequency has been computed, a query is done using the LUT manager entity. The manager has a state where the current frame of the LUT is stored and the frame has the two boundaries (lower and upper). If the queried value is inside the current frame, stored in the manager, the value is directly taken as a regular LUT. Otherwise, the same query is forwarded to the next frame (if the queried value is larger than the upper frame boundary) or to the previous (if the queried values is smaller than the frame lower boundary). The query is iteratively repeated until reaching the frame where the queried value is found. When a query is resolved into a value, the manager state is updated consequently. The hysteretic behaviour is implemented by the action of the manager internal state, as each subsequent query is first matched to the current frame boundaries. The fact that there is no constraint on the previous/next relationship between frames, implements the jump behaviour. The previous/next lack of constraints is stressed across the presented work, as it also allows the detection of complex paths of increasing/decreasing drifts. The schematic representation of the algorithm operations is reported in Fig. 6.

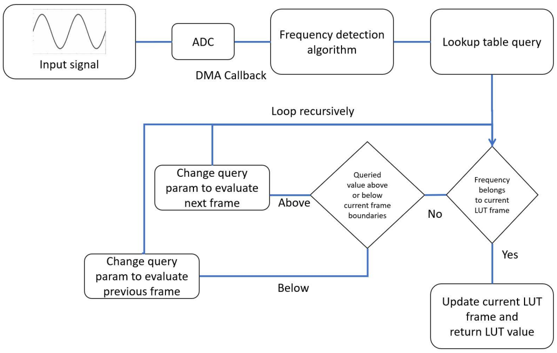

Input signal
ADC
DMA Callback
Frequency detection algorithm
Lookup table query
Loop recursively
Change query param to evaluate next frame
Above
Queried value above or below current frame boundaries
No
Frequency belongs to current LUT frame
Yes
Change query param to evaluate previous frame
Below
Update current LUT frame and return LUT value

FIGURE 6. Schematic representation of the LUT algorithm operations.

In the following sections the aforementioned terms are explained in depth, in order to clarify the implementation logic and strategy.

A. THE FRAME

The approach foundation is the frame, which is a LUT with ancillary data. A frame is composed by:

- **Values**, which are the $f(x)$ storage array. There are no constraints on the size of the values array, except for the hardware (memory) limitations. Moreover, each value array can differ in size from others, there is no constrain about array value size for each frame. This approach allows space saving for lookup zones with linear / low-varying situations. Zones which require a higher count of values can instead be rendered inside a higher count lookup array on another frame.
- **Boundaries**, which are the maximum and minimum x-coordinate value for each frame. This represents the domain of the lookup frame of interest. Boundaries are checked at runtime to discern if a jump must be done.
- **Neighbours**, which are pointers to the previous and next frame. Neighbours are bound to boundaries, as the manager lookup algorithm use them to move to the specific frame. Previous frame is bounded to the lower boundary while next frame to the upper. Note that there is no enforcement on overlapping frame domains, to allow extremely complex jump behaviour like pits or loops.

B. THE MANAGER

An entity, called *manager* handles queries to the LUT while maintaining the last table pointer, which is in turn used to implement hysteresis. Moreover, the manager entity has a pointer to the indexer function. The indexer is the function in charge of finding the index for the value inside a specific frame. The complexity of the indexer allows the implementation of small-count lookup arrays by implementing strategies to obtain intermediate values (e.g., median, linear interpolation, etc).

Key to the understanding of the whole strategy is the *current frame pointer*. This pointer is read when the LUT is accessed, to match search item with the frame boundaries, and written when a lookup refers an off-boundary request and a new frame must be searched into. The search-and-reassign strategy is recursively applied across frames, consolidating into the last, value-resolving call.

C. RUNTIME OPERATION

The lookup strategy needs an initial deployment/configuration step where all the frames are filled up with values and, subsequently, all the relationships (boundaries, previous and next frame pointers) for each frame are set-up. As before mentioned, the choice to avoid enforcing the frame consequentiality relationship allows complex moves inside tables, which in turn allows the creation of hysteresis configurations (e.g., pits or loops).

Once the instantiation process has been fulfilled, the LUT can be queried through the manager entity lookup method. The manager algorithm can be synthesized into the following operating steps:

1) get the current lookup frame;
2) check if the index is inside or outside the current frame values domain. Boundary detection trespassing direction is checked (over upper or under lower). No assumptions are made on the possible position of the searched index inside a specific frame, which becomes key in following the hysteresis path inside the curve;
3) the value is searched with the following logic: (a)
    a) If the index is inside the current frame, the indexer is called to obtain the array index for the current frame value-array
    b) If the index is outside, a recursive search call is done on the neighbour frame, determined by the trespassing direction.
4) Once the value is found, the current frame pointer on the manager is moved accordingly.

Note that, considering the recursive nature of the algorithm, an eye must be kept on the number of the frames and on the input values variability as they are directly affecting the stack pressure. However, considering typical scenarios and value dynamics, the frame count (or the value frame-distance) is typically lower than the manageable stack pressure even on limited resources microcontrollers.

D. FREQUENCY DETECTION

Going back to the frequency response of the closed-loop system with the septic nonlinearity, it becomes clear that an additional processing step is needed for its physical implementation on a MCU. Aside the analog signal acquisition, which is undertaken directly by a common ADC peripheral (which may or may not be integrated on the MCU die), a frequency detection algorithm was implemented. The presented choice favours ease and straightforwardness of implementation over more performing solution based on intensive or DSP offload-able algorithms (i.e., FFT + peak bin detection). The adopted solution is based on a mean value detection plus a zero-crossing strategy. The first one ensures that eventual DC components lead to erratic results while the latter simply counts the times the signal crosses the mean value during an interval slice. To enhance the mean-crossing strategy and reject false crossing flags, signal is low-pass filtered inside the same detection loop.

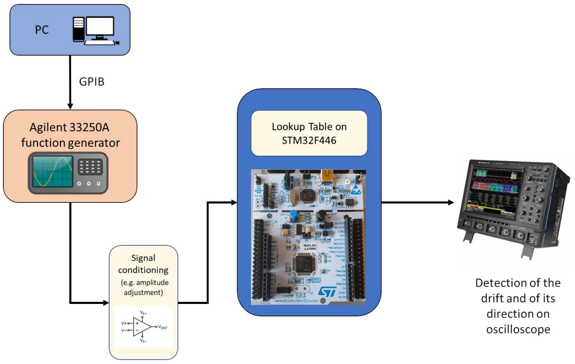

PC
GPIB
Agilent 33250A function generator
Signal conditioning (e.g. amplitude adjustment)
Lookup Table on STM32F446
Detection of the drift and of its direction on oscilloscope

FIGURE 7. Experimental setup: a standard PC equipped with a GPIB interface is used to drive the Agilent 33250A generator imposing a given frequency drift path; the signal is conditioned using standard OP-AMP configurations to fit the MCU amplitude input spefications; the LUT implemented on the STM32F446RET6 microcontroller provides an analog output indicating the occurrence of jumps which is visualized on an oscilloscope.

IV. EXPERIMENTAL SETUP

The previously described system was fully implemented on a hardware platform, to assess its validity and performances, as schematically represented in Fig. 7. A series of choices were done during the implementation transposition, which are discussed below.

A. MCU PLATFORM CHOICE

The solution was tested on an STM32F446RET6 microcontroller, hosted on a NUCLEO-F446RE development board. The choice of this development board was dictated by the general availability and price, NUCLEO boards are off-the-shelves products available at low-cost with ADC and DAC. Moreover, such boards are ready to run, have integrated programmer/debugger, expansion connectors, led and buttons and a free programming environment bundled with the boards. The F4 MCU series has a processing power up to 180 MHz frequency (225 DMIPS or 1.25 DMIPS/MHz) and a dedicated DSP instruction set (ARM DSP on ART accelerator).

B. ANCILLARY DEVICES AND SOFTWARE

To test and evaluate the algorithm implementation correctness and integrity, the MCU board was fed using an Agilent 33250A function/arbitrary waveform generator, an oscilloscope was also connected to watch the DAC (output). To drive a frequency sweep to stimulate all the possible hysteresis paths, the 33250A generator was driven using custom developed software (WinForms C♯ GUI) and GPIB which, in turn used UART debug output from the MCU to watch the current detected frequency and current frame pointer extracted from the running algorithm.

For the system to be able to work on a big frequency span, the whole logic runs inside the ISR of the ADC DMA, which means that a whole block of samples is processed using double buffering strategy to leverage the full power of the MCU and leave behind enough processing power for eventual other applications.

C. EVALUATION METHOD

The complete system was thoroughly tested by using the whole toolchain described. Especially, using the specifically designed GUI and setup, every frequency path was fed to the system, by ensuring the right path on the curve was followed. Recorded data was compared with evidence on the instruments displays to assess the whole setup validity.

V. EXPERIMENTAL RESULTS

Using the previously described setup, a testbench was developed to assess and validate the strategy. Essentially, the evaluation method consists in the ability of the system to track the frequency drift along a provided input path. Moreover, the LUT approach is validated, checking if the output value and current LUT frame is consistent with the expected values.

A. EXPERIMENTAL VALIDATION OF THE WORKING PRINCIPLE

The experimental validation of the LUT is summarized in Fig. 8. We tested the LUT functionality by considering four input sinusoidal signals whose frequency varies according to a specific path, so that all the frames of the LUT are visited. In the left column (Fig. 8(a), Fig. 8(c), Fig. 8(e),and Fig. 8(g)) the four input signals considered to test the setup are schematically represented. The frequency of each signal, thus, varies in time exploring the branches in red of the nonlinear response, invoking the LUT frame corresponding to the specific green slope. In the right column (Fig. 8(b), Fig. 8(d), Fig. 8(f),and Fig. 8(h)) the oscilloscope traces representing the corresponding response as generated by the LUT are reported. By inspecting the output of the LUT and, in particular, the jumps, it is possible to infer the sequence of drifts in the frequency of the input signal.

The first experiment has been performed sweeping from 100 Hz to 100 kHz. The LUT output results in 2 jumps, which are occurring at 33 kHz and 37 kHz. After the first jump, the LUT falls into two windows where values are inversely related to the frequency increase.

The second experiment follows a frequency path that highlights the whole LUT traversal. The input signal reverses its sweep after the first jump at 33 kHz. Notice the changes in slope across the traversal of the second and third LUT windows, with a second jump occurring 28 kHz. A further increase in the frequency of the input signal allows to follow the upper branch up to the jump occurring around 47 kHz. Notice also that as per LUT design, the only allowed path to reach the uppermost values must follow this path.

The third experiment follows a path which highlights the possibility to form loops by falling from the third window back to the first. This happens thanks to the fact that no constraints in the LUT enforce the previous/next window relationships. With a similar mechanism, pits can be created, allowing a state-machine like behaviour. This occurs when the frequency of the input signal increases above the first jump value and then decreases producing a jump up around 28 kHz and a further jump down at 25 kHz.

The last path shows a reverse path following, from the highest to the lowest frequencies values mapped in the table. The reverse sweep is again highlighting the loose relationship between LUT windows. A single jump occurs in this case around 30 kHz.

Clearly, more complex paths of frequency drifts can occur in real cases, however they can be reconstructed simply looking at the series of jumps in the output generated by the LUT.

B. DETECTION OF POWER GRID MAINS FREQUENCY DRIFT

In order to test the LUT approach outlined in this paper, we focused on detecting the occurrence of frequency drifts in an historical dataset reporting the mains frequency of the Continental Europe power grid taken with a sampling time of one minute in the week 9th−15th of September 2012 [27].

The LUT has been realized designing the multivalued frequency response so that the nominal value of the power grid frequency, i.e. $f_n = 50$ Hz, lies within the jump resonance range and the jump frequencies are located at: $f_1 = 49.98$ Hz, $f_2 = 49.99$ Hz, $f_3 = 50.01$ Hz, $f_4 = 50.02$ Hz, $f_5 = 50.04$ Hz, $f_6 = 50.06$ Hz. This is obtained choosing $\omega_0 = 270$ rad/s and $\xi = 0.01$. However, due to the modularity of the LUT implementation, note that the jump frequencies can be fixed by suitably rescaling the frequency ranges in the previously adopted LUT.

The output of the LUT, reported in Fig. 9 together with the trend of the drift of the mains frequency f around the nominal value f_n, allows to determine the specific path of frequency drift by inspecting its jumps and the levels among which they occur. The output of the LUT, in fact, shows a series of jumps which unveils specific path of drifts in the mains frequency. In order to clarify this point, let us refer to the zoom reported in Fig. 10 where two jumps are highlighted. The first jump occurs between sample 7200 and sample 7201, corresponding to the jump down at f_1 from the upper branch of the multivalued frequency response. This indicates a drift from a frequency above the nominal value ($f = 50.02$ Hz) to a frequency below it ($f = 49.98$). Moreover, the second jump still indicates that the frequency is decreasing but, as a jump up is observed at f_3 this indicates that the frequency remains above the nominal values (drifting from $f = 50.02$ Hz to $f = 50.01$ Hz).

VI. COMPARATIVE DISCUSSION

The dynamic behaviour of the LUT approach described in this paper allows the creation of loops and initial/final states, thus allowing the creation of complex state machines which are based on the input values.

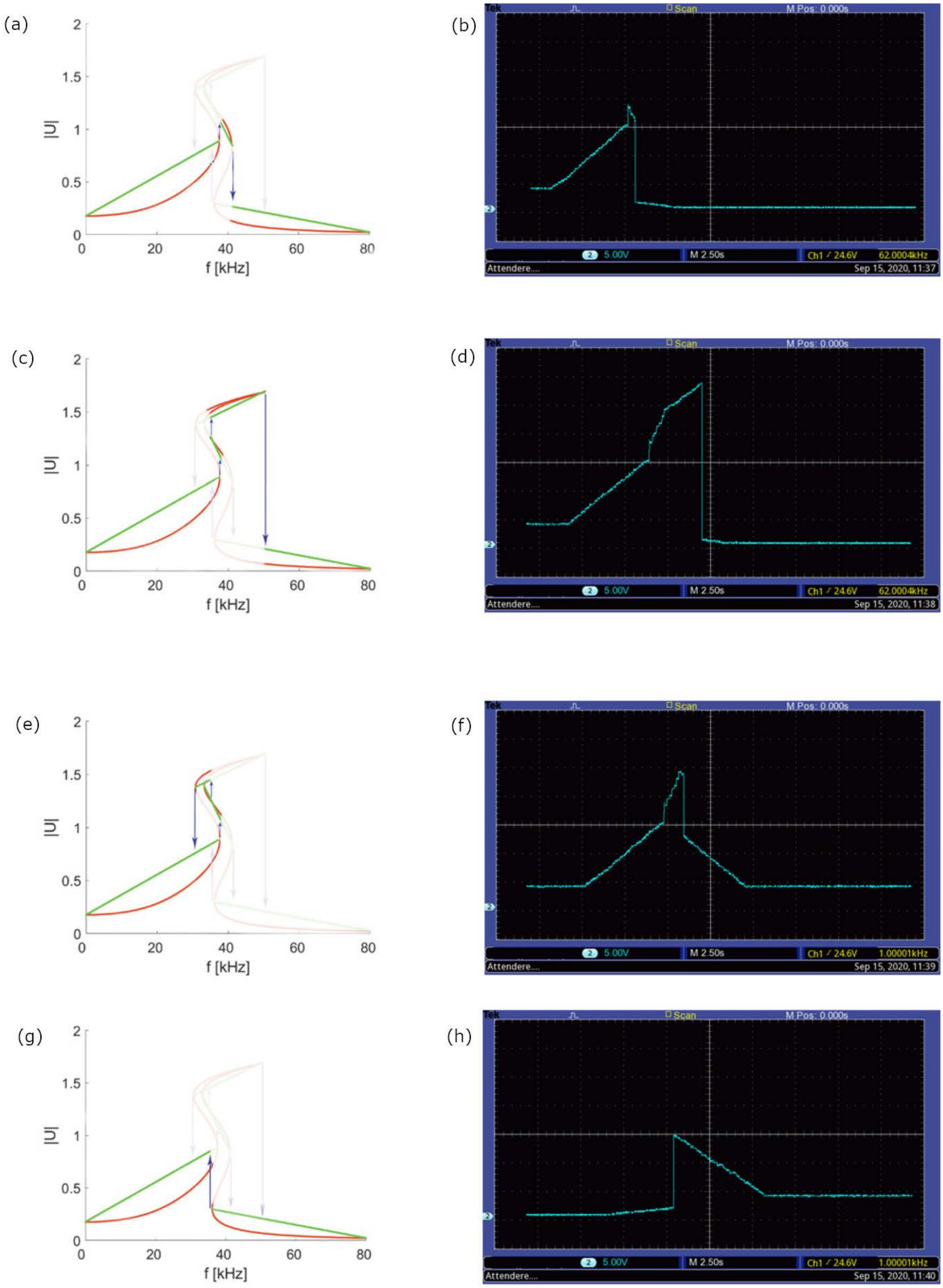


FIGURE 8. Experimental results: Test 1: (a) frequency of the input signal sweeping from 100 Hz to 90 kHz and (b) oscilloscope trace of the LUT output; Test 2: (c) frequency of the input signal sweeping from 100 Hz to 33 kHz, then from 33 kHz to 28 kHz and then from 28 kHz to 90 kHz and (d) oscilloscope trace of the LUT output; Test 3: (e) frequency of the input signal sweeping from 100 Hz to 33 kHz, then from 33 kHz to 100 Hz and (f) oscilloscope trace of the LUT output; Test 4: (g) frequency of the input signal sweeping from 100 kHz to 100 Hz and (h) oscilloscope trace of the LUT output.

The proposed LUT approach has several advantages over other approaches. Leaving out the frequency detection portion of the system, which can be implemented using several strategies, the hysteresis path behaviour can be obtained using one of the below described algorithms.

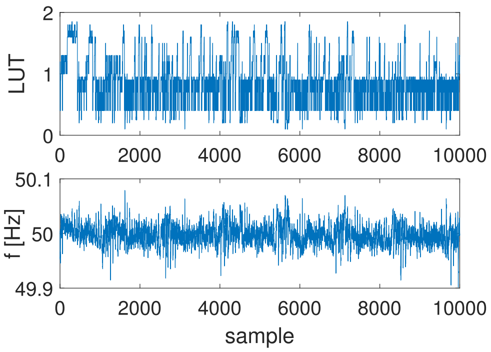


FIGURE 9. Detection of frequency drifts in mains frequency of the Continental Europe power grid. Upper panel: output of the LUT; lower panel: mains frequency trend sampled at one minute intervals.

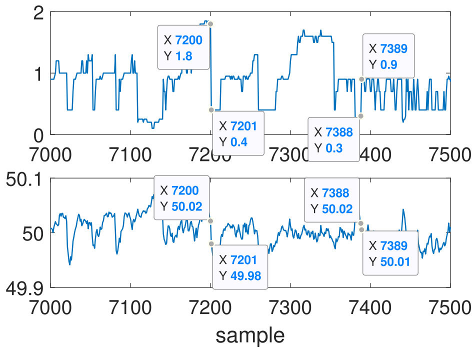


FIGURE 10. Detection of frequency drifts in mains frequency of the Continental Europe power grid, zoom between samples 7000 and 7500. Upper panel: output of the LUT; lower panel: mains frequency trend sampled at one minute intervals. The two highlighted jumps occurs at f_1 and f_3, thus unveiling two different drift paths.

A. JUMP RESONANCE LUT PARADIGM

The windowed LUT offers a straightforward way to assess complex hysteresis paths on an input signal. The proposed solution advantages, summarizing are:

- Less-than O(n) computational complexity, due to the indexing function and direct selection on the window.
- LUT windows (frames) are easy to generate, as easy as standard LUTs.
- Connections between windows (frames) can also be easily automated.
- Connections may also be reorganized on-the-fly (runtime) easily.
- Opportune windowing may help reducing footprint of flat or small-varying table portions.

As concern the computational cost of the proposed solution, as the LUT is evaluated through a single direct memory access, the running time is essentially due by the algorithm estimating the signal frequency. The proposed mean-crossing algorithm is performed within the ADC clock, therefore the running time is up to $\tau_1 \approx 5.5$ ms.

As concerns the bandwidth limits, it should be considered that the detection frequency depends both on the settings of the ADC onboard the MCU and on the DMA buffer size, which is used for the detection. From the ADC point of view, the main parameters directly affecting the sampling frequency are:

1) The clock prescaler (frequency divider): tuning prescaler gives the ability to span from (virtually) 0 to the maximum ADC frequency (2.4Msps in the STM32F443RE). The implemented system is working on a single ADC channel, thus the sample rate also depends on the number and the operating mode of channels that are acquired (e.g. 7.2 Msps in triple interleaved mode).
2) The number of bits of the ADC, which translates in clock cycles to obtain a sample. In the specific setup, the ADC requires 15/13/11/9 cycles for 12/10/8/6 bits, respectively.
3) The DMA buffer size, where the frequency detection routine is taking place. Considering that the frequency detection is essentially a threshold passing count to have a reliable detection, a sufficient number of traversal points must be found in the buffer. When working at higher ADC sample rates, problems may arise with low frequency signals, as the needed buffer size must be increased to become able to see a sufficient number of traversals.
4) The DMA buffer size also directly influences the drift detection speed. Given that the detection algorithm is executed over the whole buffer to compute the frequency, an increase in its size directly translates in an increase of the time between two successive frequency computations.

The choice of these parameters determines, therefore, the effective bandwidth. The clock prescaler puts a limit on the maximum detectable frequency: the lower the prescaler, the higher the sample rate and, consequentially, the higher the maximum detectable frequency. However, having a higher sample rate requires a larger DMA buffer size to contain one or more input waveform cycles, in order to properly detect the frequency. This means that with higher sampling rates, the low frequency detection capability is capped by the DMA buffer size.

Moreover, the varying the number of ADC bits allow a slight clock cycle improvement, thus an higher sample rate, at the cost of the loss of resolution in the signal. This can be considered a mild issue when only the frequency detection algorithm must be executed on the signal.

Summarizing, higher sample rates, obtained by means of lower prescaler values or decreased ADC bits resolution, translate into higher maximum detectable frequency but requires a DMA buffer size increase to allow also low frequency detection; larger DMA buffer sizes allow increased span of detectable frequencies but decreases the detection speed.

Carefully tuning both the parameters, the frequency span and resolution can be adjusted, to suit the specific detection needs. In the experimental analysis presented above, we fixed the ADC clock at 11.250 MHz, with a resolution of 12 bits, and DMA buffer size of 4096 samples. This leads to admissible input signals with frequency in the range between 10 Hz and 300 kHz.

B. STRAIGHT COMPUTATION

Instead of having all the values in a table, each single generating function could be implemented on the obtained frequency instantaneous value and trend. This approach has the advantages of a smaller code and memory requirements (reduced RAM/FLASH footprint), and easier coding, due to simple implementation of the functions, which could become code libraries. On the other hand, disadvantages of this approach are the heavier computation, which may require more power (processing and, in turn, energy), complex behavioural scenarios cannot always be represented by paths described by functions (e.g., pits), dynamics may need added state-machine like control, trend computation may require other numerical filtering to become stable, eventual table adjustment may require full rewriting of equations, generation of complex curves may require entire equations and jump logic rewrite, and difficult automated generation of on-the-fly equation to apply. Adopting a DSP-ARM processor, the frequency content is evaluated on the STM32F446RET6 microcontroller in 1.8 ms, i.e. the time needed to calculate the FFT over a dataset of 4096 samples, consistent to the set used in our approach, once the buffer has been populated in $\tau_1 \approx 5.5$ ms. Even if the FFT can be performed within the same clock cycle, in order to determine the drift, a number N of complete frequency evaluations are necessary to determine the drift, which depends on the drift speed itself. Therefore, the running time is about $\tau_2 = N\tau_1$. The bandwidth is therefore reduced with respect to the LUT case.

C. STATE MACHINES

State machine approach is similar to the proposed one, which means that, instead of describing jumps using an intrinsic strategy, the logic is moved into a switch-case condition where frequency trend is evaluated constantly together with current detected frequency. This approach has a similar memory footprint and code complexity of the proposed solution, while it may be error-prone as jumps logic must be fully specified using a switch case logic, which may result in complex state changes to reach wanted states. Moreover, it presents difficult tuning, and maintainability and difficult automated table generation or on-the-fly changes. The running time is also affected by these nested logic, nevertheless maintaining the need of $\tau_1 = 5.5$ ms to complete the data acquisition. Therefore, the time needed by this approach to detect the drift is $\tau_3 = \tau_1 + \tau_L$, where $\tau_L \approx 1.8$ ms is the average time to process the nested logic. Also in this case, the increased average time leads to a reduced bandwidth with respect to the LUT case.

VII. CONCLUSION

Frequency drifts are common in several areas either technological or scientific. The reliable and efficient detection of frequency drifts, therefore, is a problem with multiple application in industrial informatics and in related fields and not only focusing on electric and electronic scenarios. In fact, it arises also in mechanics and civil engineering, where vibrations and their frequency must suitably monitored. Moreover, a recently emerging area of interest is also in the context of the pandemic monitoring. As a speculative example, in the Covid-19 pandemics, vaccination rate assumed a fundamental role in controlling the dimension of the outbreak. Sensing the variation of the vaccination rate, and the effect that this produces, can be considered as a frequency drift detection problem. By monitoring the vaccination rate, it may be possible to qualitatively infer the impact of the outbreak of new SARS-CoV-2 variants.

The paradigm introduced in this paper offers an easy and MCU friendly way of implementing a device for frequency drift detection based on a nonlinear dynamical feedback system not designed to exclusively monitor electrical quantities. This key feature allows to obtain a device which is highly reconfigurable, as opportunely shaping the frequency response of the nonlinear system, it is possible to implement complex LUTs with hysteresis. Moreover, reconfiguring the nonlinear part of the oscillator, increasing the order of the polynomial nonlinearity easily ensures the possibility to get the desired selectivity for the drift detection, allowing to discriminate over complex paths of increase and decrease of the frequency without the a-priori knowledge of the drift dynamics. It should be noted that the only a-priori knowledge needed to construct the LUT is the nominal frequency of the signal under monitoring. Slightly tuning the parameters of the nonlinear system ensures the capability to adapt the LUT to the specific characteristics of the frequency drifts, thus optimizing its capabilities.

The frequency range over which it is possible to design the desired nested detection hysteresis is limited by several factors which, however, make the solution suitable up to 3 MHz signals for the considered STM32F446RET6 microcontroller. Its ease in terms of algorithm complexity, renders its implementation immediate even on power-limited devices. The possibility to implement the LUT on low-cost devices is a fundamental advantage of the proposed scenario since puts this new strategy in the class of cheap DSP equipment.

REFERENCES

[1] G. Filatrella, A. H. Nielsen, and N. F. Pedersen, "Analysis of a power grid using a Kuramoto-like model," *Eur. Phys. J. B*, vol. 61, no. 4, pp. 485–491, Mar. 2008.

[2] P. T. de Boer. *Accuracy and Stability of the 50 Hz Mains Frequency*. Accessed: Aug. 27, 2022. [Online]. Available: http://wwwhome.cs.utwente.nl/~ptdeboer/misc/mains.html

[3] L. V. Gambuzza, A. Buscarino, L. Fortuna, M. Porfiri, and M. Frasca, "Analysis of dynamical robustness to noise in power grids," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 7, no. 3, pp. 413–421, Sep. 2017.

[4] T. Niu, J. Wang, H. Lu, W. Yang, and P. Du, "A learning system integrating temporal convolution and deep learning for predictive modeling of crude oil price," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4602–4612, Jul. 2020.

[5] T. Odagaki, "Self-organization of oscillation in an epidemic model for COVID-19," *Phys. A, Stat. Mech. Appl.*, vol. 573, Jul. 2021, Art. no. 125925.

[6] G. E. Billman, "Heart rate variability—A historical perspective," *Frontiers Physiol.*, vol. 2, Nov. 2011, Art. no. 86.

[7] M. S. Reza and M. M. Hossain, "Enhanced grid synchronization technique based on frequency detector for three-phase systems," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2180–2191, Apr. 2021.

[8] M. Branciforte, L. Fortuna, A. Buscarino, M. Bucolo, and F. N. Poruthotage, "Electronic device, corresponding apparatus, method and computer program product," U.S. Patent 1 130 059 6 B2, Oct. 9, 2022.

[9] R. C. Agarwal, J. W. Cooley, F. G. Gustavson, J. B. Shearer, G. Slishman, and B. Tuckerman, "New scalar and vector elementary functions for the IBM system/370," *IBM J. Res. Develop.*, vol. 30, no. 2, pp. 126–144, Mar. 1986.

[10] P. T. P. Tang, "Table-lookup algorithms for elementary functions and their error analysis," in *Proc. 10th IEEE Symp. Comput. Arithmetic*, Jun. 1991, pp. 232–236.

[11] T. J. McCabe, "A complexity measure," *IEEE Trans. Softw. Eng.*, vol. SE-2, no. 4, pp. 308–320, Dec. 1976.

[12] A. Suresh, E. Rohou, and A. Seznec, "Compile-time function memoization," in *Proc. 26th Int. Conf. Compiler Construct.*, Feb. 2017, pp. 45–54.

[13] A. Pang and P. Membrey, *Beginning FPGA: Programming Metal*. New York, NY, USA: Apress, 2019.

[14] A. A. Youssef, B. Murmann, and H. Omran, "Analog IC design using precomputed lookup tables: Challenges and solutions," *IEEE Access*, vol. 8, pp. 134640–134652, 2020.

[15] V. Boussard, S. Coulombe, F.-X. Coudoux, and P. Corlay, "CRC-based correction of multiple errors using an optimized lookup table," *IEEE Access*, vol. 10, pp. 23931–23947, 2022.

[16] Q. H. Quadri, S. Nuzzo, M. Rashed, C. Gerada, and M. Galea, "Modeling of classical synchronous generators using size-efficient lookup tables with skewing effect," *IEEE Access*, vol. 7, pp. 174551–174561, 2019.

[17] J. Cong, M. Ercegovac, M. Huang, S. Li, and B. Xiao, "Energy-efficient computing using adaptive table lookup based on nonvolatile memories," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, Sep. 2013, pp. 280–285.

[18] H. Magalháes, F. Marques, B. Liu, J. Pombo, P. Flores, J. Ambrósio, and S. Bruni, "An optimization approach to generate accurate and efficient lookup tables for engineering applications," in *Proc. Int. Conf. Eng. Optim.* Cham, Switzerland: Springer, Sep. 2018, pp. 1446–1457.

[19] M. Pharr and R. Fernando, *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*. Reading, MA, USA: Addison-Wesley, 2005.

[20] A. Buscarino, C. Famoso, L. Fortuna, and M. Frasca, "Multi-jump resonance systems," *Int. J. Control*, vol. 93, no. 2, pp. 282–292, Feb. 2020.

[21] M. Bucolo, A. Buscarino, L. Fortuna, and M. Frasca, "Multiple hysteresis jump resonance in a class of forced nonlinear circuits and systems," *Int. J. Bifurcation Chaos*, vol. 30, no. 15, Dec. 2020, Art. no. 2050258.

[22] A. Buscarino, L. Fortuna, and M. Frasca, *Essentials of Nonlinear Circuit Dynamics With MATLAB and Laboratory Experiments*. Boca Raton, FL, USA: CRC Press, 2017.

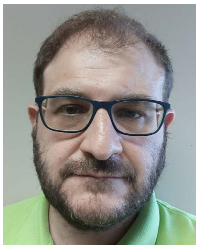
[23] D. P. Atherton, *Nonlinear Control Engineering*. Wokingham, U.K.: Van Nostrand Reinhold, 1982.

[24] R. Genesio and A. Tesi, "Harmonic balance methods for the analysis of chaotic dynamics in nonlinear systems," *Automatica*, vol. 28, no. 3, pp. 531–548, May 1992.

[25] P. A. Cook, *Nonlinear Dynamical Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 1994.

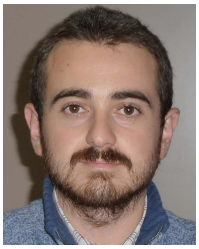
[26] D. Zwillinger, Ed., *CRC Standard Mathematical Tables and Formulae*. Boca Raton, FL, USA: CRC Press, 2011.

[27] *Dataset*. Accessed: Aug. 27, 2022. [Online]. Available: https://www.mainsfrequency.com/services.htm



GIUSEPPE AVON received the M.S. degree in automation engineering and control of complex systems, in 2020. He is currently pursuing the Ph.D. degree in systems, energy, computer and telecommunications engineering with the University of Catania within the framework of the ENI-CNR Joint Research Agreement. He is ending a Traineeship at Fusion for Energy ITER Development, CODAC Department, as a C/C++ Developer for standard and embedded systems, specialized in MARTe2 and MARTe2-components real-time application framework. He has previous work experiences in the software development field, mainly devoted to custom machine supervision and control.

MAIDE BUCOLO (Senior Member, IEEE) received the M.S. degree in computer science engineering and the Ph.D. degree in electronic and control engineering from the University of Catania, in 1997 and 2001, respectively. During the Ph.D., she worked as a Research Scholar at the University of California at San Diego (UCSD) and after that, often, she has been a Visiting Researcher at the Microhemodynamics Laboratory, Department of Bioengineering, UCSD. She is currently an Associate Professor of control system at the Department of Electrical, Electronic and Informatics, University of Catania. In 2010, she established and became responsible of the Bio-Microfluidics Laboratory. She has been the co-ordinator of national projects and international exchange programs. She worked as an expert in the technology innovation demand of regional small and medium enterprisers. She has published more than 100 scientific contributions in peer-reviewed international journals and conferences. Her research interests include methodologies and low-cost technologies for bio-microfluidics systems modeling and control. She serves as an Associate Editor for the IEEE Transactions on Biomedical Circuits and Systems. She is the President of the Master's Course in Automation Engineering at Control of Complex Systems at the University of Catania.



ARTURO BUSCARINO (Senior Member, IEEE) received the degree in computer science engineering and the Ph.D. degree in electronics and automation engineering from the University of Catania, Italy, in 2004 and 2008, respectively. He is currently an Associate Professor at the University of Catania, he teaches modeling and optimization at the Laura Magistrale in management engineering, automatic control, the Laurea in electronics engineering and nonlinear systems control, and the Laurea Magistrale in automation engineering and control of complex systems, University of Catania. He works on nonlinear circuits, control, and synchronization. He published more than 200 papers on refereed international journals and international conference proceedings. His research interests include nonlinear systems and chaos, complex networks, control systems, cellular nonlinear networks, and plasma engineering.

LUIGI FORTUNA (Fellow, IEEE) received the degree *(cum laude)* in electrical engineering from the University of Catania, Italy, in 1977. He was the Co-ordinator of the courses in electronic engineering and the Head of the Dipartimento di Ingegneria Elettrica Elettronica e dei Sistemi. From 2005 to 2012, he was the Dean of the Engineering Faculty. He also teaches automatic control and robust control. He is currently a Full Professor of system theory with the University of Catania. He has published more than 500 technical articles and 12 scientific books. His research interests include robust control, nonlinear science and complexity, chaos, cellular neural networks, soft computing strategies for control, robotics, micronanosensor and smart devices for control, and nanocellular neural networks modeling. He was the IEEE Circuits and Systems (CAS) Chair of the CNN Technical Committee, the IEEE CAS Distinguished Lecturer, from 2001 to 2002, and the IEEE Chair of the IEEE CAS Chapter Central-South Italy.

Open Access funding provided by 'Università degli Studi di Catania' within the CRUI CARE Agreement