This article has been accepted for publication in IEEE Open Journal of Industry Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJIA.2024.3366415

Analytical Delay Evaluation for FPGA based Repetitive Controller in AC Variable Frequency Applications

A. Faro, A. Lidozzi, *Member, IEEE,* M. di Benedetto, *Member, IEEE* and L. Solero, *Senior Member, IEEE*

Abstract Repetitive Controller provides a very low THD in the quantities under control. It exhibits an inherent issue when operated to track variable frequency references. The paper deals with the analysis of the operating conditions when the controller is executed at variable frequency without any resynchronization with respect to the PWM carrier, which is the most common mode of operation. The delays introduced are then evaluated analytically concerning the sampling and output frequency that must be tracked. The proposed analysis allows obtaining the maximum delay affecting the control chain, which introduced by the Repetitive Control desynched operation. The knowledge of the delay introduced in the control loops is at the basis of any control tuning procedure and gains selection, even when adaptive control strategies are used.

Index Terms— repetitive control, digital control, variable frequency, FPGA

NOMENCLATURE

T_{SW}	Switching and sampling period.
n	Depth of the Repetitive Control delay line.
ΔT_{ACQ}	Time required by the sampling and digital conversion, which is related to the Analog-to-Digital input interface.
ΔT_{MODg}	Control Algorithm execution time, at the end of that the updated modulating signals are available.
n_{sh}	Initial shift between the PWM synch signal (triggering the sampling of measurements, etc…) and the Control Algorithm call.
$MCTM$	Maximum Control Trigger Mismatch (Absolute maximum value).
$MCTM_{MS}$	Maximum Control Trigger Mismatch of the Modulating Signals (Maximum delay related to the modulating signal update).
$ATM_{critical}$	Acquire Triggers Mismatch Critical (It represents the critical value of the *ATM*: it will be linked to the *MCTM* when $MCTM_{MS}$ does not exist; there are no issues in updating the modulating signals).
MST_{SW}	Mismatch immediately Smaller than T_{SW} (Relative Maximum delay within T_{SW}).
$MMAL$	Minimum Mismatch Acquiring Lag (Minimum time mismatch related to the acquisition of measurements, related to the T_{SW} and ΔT_{ACQ} quantities).
ATM_{MIN}	Acquire Trigger Mismatch Minimum (Complementary with respect T_{SW} to *MMAL*).
$CTM_{ATM_{MIN}}$	Control Trigger Mismatch ATM (It is linked to T_{sw} and ATM_{MIN}).
$MaxTotDelay$	Maximum delay to be considered for the Control Algorithm.
MTM	Minimum Triggers Mismatch.

I. INTRODUCTION

Repetitive Controller has been widely used in several applications where a sinusoidal reference must be tracked [1] - [2], and when the main controllers must be supported for harmonics compensation [3]. In grid-connected [4] - [5], stand-alone [6] - [7] or electrical drives [8] - [9] applications, where the fundamental frequency changes or even in stand-alone mode when the output fundamental harmonic must be modified, a conventional Repetitive Controller requires an interpolation method or additional memory to achieve the required frequency adaptability. It results in very complicated control structures, which in many cases tend to favor alternative control strategies [10]. On the other hand, Repetitive Control (RC) could offer much better performance combined with implementation simplicity [11], even including the adoption of digital filter solutions to avoid instabilities [12]. A very straightforward approach is to de-link the execution of the RC from the main PWM Modulator (i.e. the main scheduler), which represents the main timing. The results are a straightforward control structure that keeps the well-known and reliable architecture typical of the RC; however, it is executed at variable frequency. Discrete-time variable-structure repetitive control operating in quasi-sliding mode is presented in [13], verified by simulation only, and resulting in a very complicated implementation. In contrast, a preliminary approach to an adaptive repetitive control for discrete systems is proposed in [14]. In [15], a robust RC based on a causal IIR compensator with time-varying sampling periods is described. However, variable delays are not considered in the implementation. A frequency adaptive proportional-RC for grid-connected inverters is proposed in [16], being based on the Thiran all-pass IIR filter. In order to resist to limited grid frequency variation, a multi-bandwidth RC is proposed in [17] where each of the resonant bandwidth is set individually by an internal model filter. Variable delay estimation when the RC is operated desynchronized from the main scheduler is reported in [18], with the limitation of assuming that the RC loops will be synched to the main scheduler at any frequency variation. The control method proposed in [19] could compensate for the harmonics more accurately with an integer order RC thanks to the high switching and sampling frequency of the adopted SiC devices. Similarly, in [20] - [21] the virtual-variable sampling is proposed to

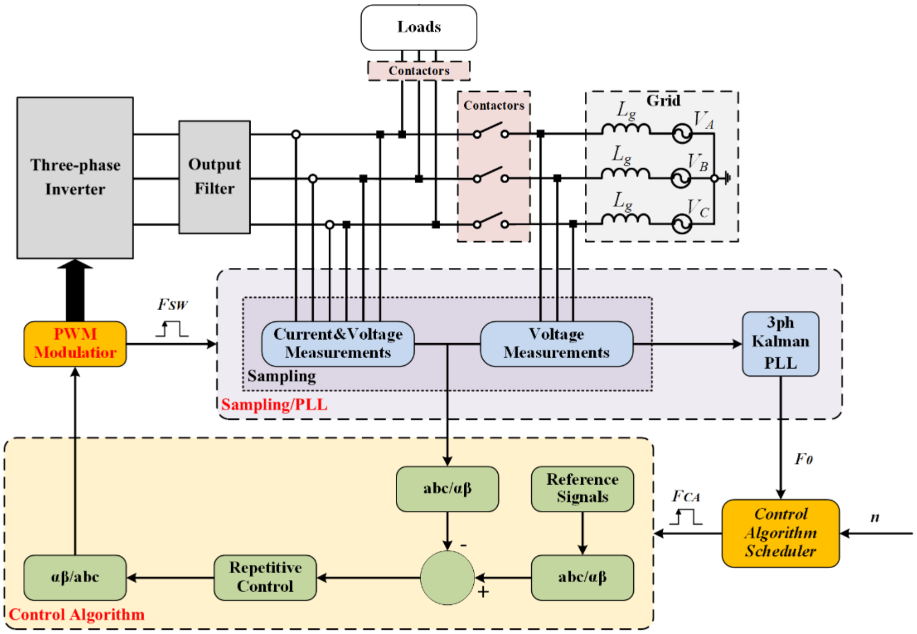


Figure 1. Block scheme of the Frequency Adaptive Repetitive Control.

overcome the gain degradation of the RC when operated at a frequency that is different from the rated one. However, in many industrial or cost-effective applications, the usage of wide bandgap devices is not suitable and hence, alternative control solutions must be adopted to take advantage of the RC structure. A specific manipulation of the delay line to overcome the performance degradation of the RC due to the frequency variations of the signal to be tracked is illustrated in [22]. In the presented cases, the RC is executed with a variable period to keep the delay-line at a constant depth, hence, the introduced variable delays must be accurately considered. The desynched operation of the RC brings into the control chain a certain amount of variable delay that must be accurately determined before any tuning procedure can be applied.

The proposed analytical approach allows the real-time evaluation of the delay introduced by the use of repetitive controllers with variable frequency and constant delay line. Furthermore, the analytical treatment will allow estimating the instantaneous delay and its maximum value before it is present in the system. That is completely different from a direct measurement with embedded digital counters, as they provide the amount of delay after the delay has already produced its effects on the system.

II. System Description and Analytical Approach

In both grid-tied and intentional islanding operations, the output frequency of the AC signals cannot be considered a constant, non-varying signal. Hence, the advantages of using the Repetitive Control are then lost. Figure 1 illustrates the block scheme of the variable frequency Repetitive Controller, with a desynched operation concerning the main PWM scheduler [23]. Hence, important considerations must be taken to achieve a comprehensive analysis and then obtain suitable RC parameters tuning. Due to a time shift between the Control Algorithm (i.e., the Repetitive Control) and the PWM synch signal, the delay evaluation is a mandatory task before any tuning considerations. In this method three loops are realized to execute the control algorithm with a variable frequency, which is generally different from the switching and sampling main frequency.

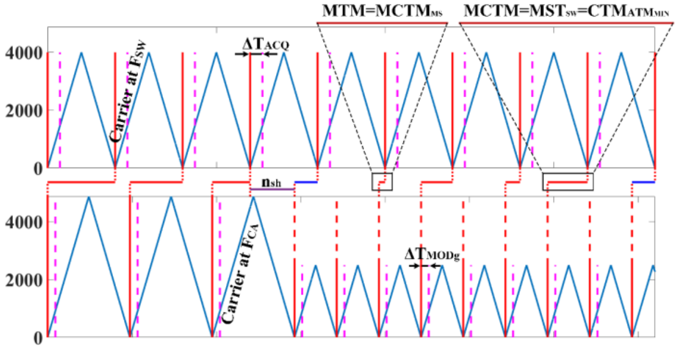


Figure 2. F_0 step variation from *41 Hz* to *80 Hz* when the PWM and control carriers are not aligned.

The first loop contains two decoupled and totally independent structures; the PWM Modulator (PWMM) is used to provide opening/closing signals to the inverter's switches and to send the trigger signal to the Sampling/PLL block according to the switching frequency.

This guarantees that measure acquisition task samples the instantaneous average value. The Control Algorithm Scheduler has been specifically intended to provide the trigger signal for the Control Algorithm (CA) according to the PLL estimated output fundamental frequency.

As shown in Figure 1 , the CA is tight to the synch signal, which exhibits the same frequency of the PWM Modulator only when the output frequency is the rated. In fact, that loop runs at a variable frequency (F_{CA}), calculated from the delay line length n (i.e. with constant depth) and the output fundamental frequency F_0 usually achieved by a PLL in grid-tied mode of operation (or the desired output frequency when operating in stand-alone mode): $F_{CA}=n\cdot F_0$.

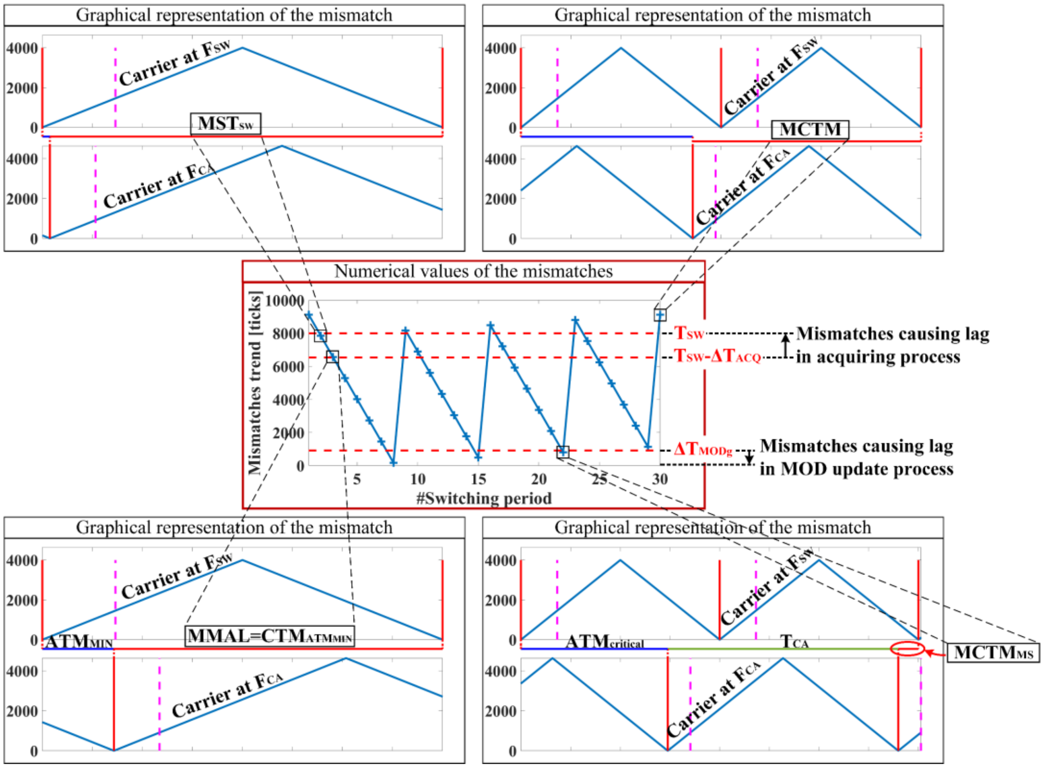
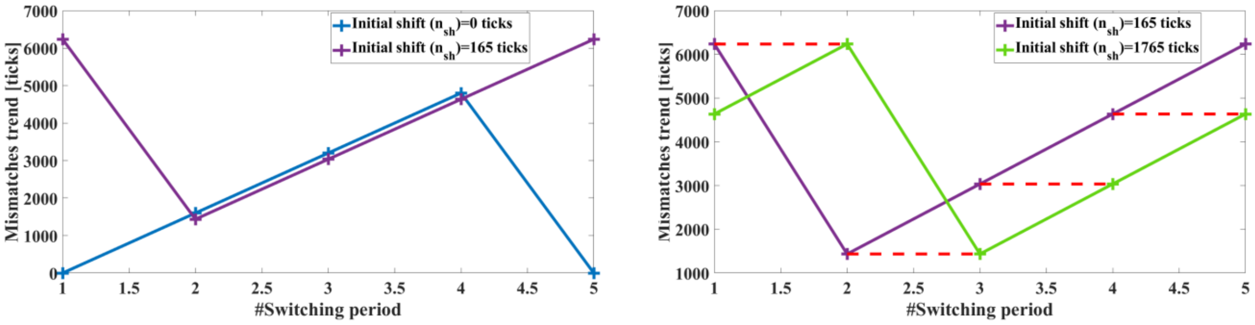


Figure 3. Newly introduced parameters for the most comprehensive *MaxTotDelay* evaluation. F_0=*43.1 Hz*, n_{sh}=*150 ticks*.

When the control algorithm is running and the fundamental frequency changes, since this variation most probably will occur when the two carrier waveforms are not aligned, the resulting synchronous event could not appear anymore for the new F_0 as it can be noticed from Figure 2.

The quantity n_{sh} is considered as the initial shift, equal to an integer number of ticks ranging from *1* to T_{SW} *(ticks)*, between the two carrier waveforms (i.e. PWM and sampling) when F_{CA} is changed due to a F_0 variation. The initial shift is totally random, being related to the history of operation. The proposed analysis has been performed considering an initial shift different from zero, avoiding the ideal case of aligned carriers. The algorithm can only know the range of the shift that goes from 0 to the number of ticks relative to the switching frequency. This will result in a possible alteration of the maximum delay achieved with the two methods presented in [23]. Therefore, a new technique has been developed to provide the maximum delay whatever it is the starting shift (n_{sh}) between the two triggers. As it can be seen from Figure 4a, for $n_{sh}\neq 0$ the mismatches between the PWM modulator trigger and the Control Algorithm call will assume different values from the case in which synchronism would be present ($n_{sh}=0$).

However, the typical period of the specific mismatch is maintained after a certain number of switching periods. Moreover, the initial shift n_{sh} to be considered could be equal to an integer number ranging from *1* to T_{SW} *(ticks)*. However, as shown in Figure 4b, for a n_{sh} greater than the *MTM* at $n_{sh}=0$, the mismatches will assume the same values as for a case of a certain shift lower than the *MTM* at $n_{sh}=0$. In the reported case the *MTM* at $n_{sh}=0$ is equal to *1600* ticks, whereas its general expression is as in (1). Therefore, only n_{sh} smaller than *MTM* at



a)

b)

Figure 4. a) Mismatches trend comparison for F_0=*62.5 Hz* when initial shift changes from *0* ticks to *165* ticks. b) Mismatches trend comparison for F_0=*62.5 Hz* when initial shift changes from *165* ticks to *1765* ticks.

n_{sh}=0 can be considered, reducing the possible scenarios that could arise when the synchronism is not present.

The proposed analysis is based on evaluating the *MTM* quantity, which would be possible by introducing the *n_{sh}* parameter and the variables defined in the nomenclature section. Main quantities have been graphically displayed in Figure 3 and their closed form analytical expressions will be shown in the proposed analysis.

$$\begin{cases} \begin{bmatrix} Mismatch_1 \\ \dots \\ Mismatch_N \end{bmatrix}_{n_{sh_1} > MTM|_{n_{sh}=0}} = \begin{bmatrix} Mismatch_1 \\ \dots \\ Mismatch_N \end{bmatrix}_{n_{sh_2} < MTM|_{n_{sh}=0}} \\ n_{sh_2} = n_{sh_1} - \left\lfloor \dfrac{n_{sh_1}}{MTM|_{n_{sh}=0}} \right\rfloor \cdot MTM|_{n_{sh}=0} \end{cases} \tag{1}$$

The image in the center represents the value assumed by the mismatch between the triggers of two adjacent control cycles as the switching period varies. The images on the outside are specific zooms taken from the main plot. As can be seen, the trend is periodic, therefore the initial mismatch is found after *N* switching periods, with *N* depending on *F_{sw}* and *F_0*. To calculate the *MaxTotDelay*, additional parameters have been introduced, such as the *MCTM*, *MSTSW*, etc.

For each of these values a zoom has been made to view them through the two carriers. Therefore, the crosses on the central image represent the numerical value of the given parameter which is then zoomed and graphically represented by the distance between the carrier triggers, as reported in the outer plots of Figure 3. The first parameter to evaluate is the new *MTM* that will be present when the initial *shift* between the two carrier waveforms is different from 0. This value can be rated from (2):

$$MTM|_{n_{sh}} = \left(\left\lfloor \frac{n_{sh}}{MTM|_{n_{sh}=0}} \right\rfloor + 1 \right) \cdot MTM|_{n_{sh}=0} - n_{sh} \tag{2}$$

The *Floor* operator ⌊ ⌋ returns the integer part without rounding.

The second parameter to be updated is the *MaxTM*. To take into account the possible initial *shift*, the formula that can be used is the (3):

$$\begin{aligned} MaxTM|_{n_{sh}} &= \overbrace{(N-1) \cdot MTM|_{n_{sh}=0}}^{MaxTM|_{n_{sh}=0}} + \left\lfloor \frac{n_{sh}}{MTM|_{n_{sh}=0}} \right\rfloor \cdot \\ &\cdot MTM|_{n_{sh}=0} - n_{sh} + \left\lceil \frac{\mathrm{mod}\left(n_{sh}, MTM|_{n_{sh}=0}\right)}{T_{SW}} \right\rceil \cdot MTM|_{n_{sh}=0} \end{aligned} \tag{3}$$

Ceil operator ⌈ ⌉ will provide the nearest greater integer value. From the updated value of the *MaxTM*, related to the current *n_{sh}*, it is possible to rate the maximum value that the mismatch will assume (MCTM):

$$MCTM = MaxTM|_{n_{sh}} + \left\lfloor \frac{MTM|_{n_{sh}}}{MTM|_{n_{sh}=0}} \right\rfloor \cdot MTM|_{n_{sh}=0} \tag{4}$$

After updating the parameters previously introduced in [18], where the case with the two carrier waveforms starting aligned was analyzed, additional quantities must be evaluated to give the most general formula for computing the *MaxTotDelay*. First, it is necessary to understand if a mismatch immediately smaller than the *Δt_{MODg}* arises, and its value. That misalignment, defined as *$MCTM_{MS}$*, will be the greater mismatch for which the modulating signals update will lag. The value can be computed using (5):

$$\begin{aligned} MCTM_{MS} &= \left\lfloor \frac{\Delta t_{MOD_g}}{MTM|_{n_{sh}=0}} \right\rfloor MTM|_{n_{sh}=0} + MTM|_{n_{sh}} - \\ &+ \left(1 - \left\lceil \frac{\left\lfloor \dfrac{\Delta t_{MOD_g} - \left\lfloor \dfrac{\Delta t_{MOD_g}}{MTM|_{n_{sh}=0}} \right\rfloor \cdot MTM|_{n_{sh}=0}}{MTM|_{n_{sh}}} \right\rfloor}{s_3 \cdot T_{SW} + (1 - s_3) \cdot T_{CA}} \right\rceil \right) MTM|_{n_{sh}=0} \end{aligned} \tag{5}$$

When there are no time mismatches in the current observation window that are smaller than or equal to *Δt_{MODg}*, it can be read that the delay in the modulating signals update will not affect any of the switching periods: (5) will return a negative or a 0 value. On the contrary, when the time discrepancies are smaller than or equal to *Δt_{MODg}*, the related switching periods will be affected by the delay of the modulating signals update, which do not receive the last computed value: (5) will return the greater of these mismatches. It is then possible to directly identify the previously shown cases by introducing a selector value, which can be expressed as in (6):

$$s_1 = \left\lceil \frac{\dfrac{|MCTM_{MS}| + MCTM_{MS}}{2}}{\Delta t_{MOD_g}} \right\rceil \tag{6}$$

The next step is to evaluate the *$ATM_{critical}$* parameter. This value is the *MCTM* paired part when *s_1* is 0. In fact, in this case all the switching periods will receive the updated modulating signals without additional delays. On the contrary, when *s_1* is equal to *1* it points out that *$ATM_{critical}$* will be the complementary part of *$MCTM_{MS}$* + *T_{CA}*. In fact, the switching period related to the *$MCTM_{MS}$* will receive the modulating signals from the previous CA iteration, instead of the current one. It follows that the largest delay related to the modulating signals update process will be equal to the *$MCTM_{MS}$* plus a CA period. The *$ATM_{critical}$* can be evaluated as in (7):

$$\begin{aligned} ATM_{critical} &= \left\lfloor \frac{s_1 \left(MCTM_{MS} + T_{CA} \right) + (1 - s_1) MCTM}{T_{SW}} \right\rfloor T_{SW} - \\ &+ \left[s_1 \left(MCTM_{MS} + T_{CA} \right) + (1 - s_1) MCTM \right] \end{aligned} \tag{7}$$

To later distinguish the case where the *$ATM_{critical}$* is smaller or equal than *Δt_{ACQ}* (i.e. the measurement acquisition is lagging the current CA iteration) from the case where the *$ATM_{critical}$* is greater than *Δt_{ACQ}* (i.e. the measurement acquisition is temporally consistent with the current CA iteration), an additional index could be introduced as shown in (8):

$$s_2 = \left\lceil \frac{\left\lfloor \dfrac{\Delta t_{ACQ}}{ATM_{critical} + 1 - \left\lceil \dfrac{ATM_{critical}}{T_{SW}} \right\rceil} \right\rfloor}{\Delta t_{ACQ}} \right\rceil \qquad (8)$$

The coefficient s_2 will be equal to *0* when $ATM_{critical}$ will be greater than Δt_{ACQ} (i.e. the current CA iteration will receive the measurements from the last sampling period), while s_2 will be equal to 1 when $ATM_{critical}$ will be smaller than or equal to Δt_{ACQ} (i.e. the current CA iteration will receive the measurements from the previous sampling, resulting in a lag of the acquiring process).

Concerning the quantities Δt_{ACQ} e Δt_{MODg}, totally random values were used, since they depend on how the control algorithm was implemented and on its computational burden. The constraint is that Δt_{ACQ} must not be greater than the sampling time ($1/F_{sw}$), otherwise a suitable under sampling would be necessary, keeping valid the proposed theory. Now, since F_{SW} is theoretically fixed, the upper limit of Δt_{ACQ} is limited by the sampling time.

Concerning Δt_{MODg} however, the procedure takes place with a completely opposite approach. In fact, it depends on how the control is implemented and its value will impose a higher limit on the F_{CA} frequency that can be used. At this point, four scenarios could arise.

A. s_1=1 and s_2=1

Since $s_1=1$, a mismatch called $MCTM_{MS}$, and smaller than Δt_{MODg}, exists, as shown in Figure 5. The control algorithm (CA) will not be able to provide the modulating signal for the next switching period, which will apply the previous values. Therefore, the resulting control actions will be loaded and then applied with a delay: formally, a time equal to T_{ca} is lost.

Moreover, since $s_2=1$, the $ATM_{critical}$ is lower than Δt_{ACQ}, as highlighted in Figure 5. In this case, the $ATM_{critical}$ is related to the current CA iteration that will provide the updated modulating signals to the switching period that will start after a T_{CA}+$MCTM_{MS}$ ticks. Moreover, the acquired measurements will be supplied to the Control Algorithm after the scheduler has called it. Therefore, the Control Algorithm will use the samples available from the previous iteration: a T_{SW} is then lost. Therefore, the updated control actions are loaded with a delay of a CA period, jointly with a delay due to non-coherent measurements that exhibit a time lag of a sampling period.

B. s_1=0 and s_2=0

Since $s_1=0$ there are no mismatches lower than Δt_{MODg}. Hence, all switching periods will receive the updated and coherent values of the modulating signals. Moreover, since $s_2=0$, the $ATM_{critical}$ is not smaller than Δt_{ACQ}. In this case $ATM_{critical}$ is directly linked to the *MCTM*, being the minimum *ATM*. All the CA calls will be executed with the updated values of the measurements, as highlighted in Figure 6. There will be no additional delays in the control structure, neither in measurements acquisition nor when loading the updated modulating signals.

C. s_1=0 and s_2=1

Since $s_1=0$ there are no mismatches smaller than Δt_{MODg}. Each CA iteration will end within the current switching cycle, as graphically shown in Figure 7.

However, since $s_2=1$, the resulting $ATM_{critical}$ is lower than Δt_{ACQ}. In this case the $ATM_{critical}$ is directly related to the *MCTM*, being the minimum *ATM*). During one or more cycles of the Control Algorithm (CA) the modulating signals will be calculated with measurements that are not consistent with the current control period: the previous cycle quantities will be used, as illustrated in Figure 7. The effect can be explained by considering that the measures acquisition process ends after the trigger of the CA, even if started before the Control Algorithm call. The delay in this case is equal to T_{sw}.

D. s_1=1 and s_2=0

Since $s_1=1$, there is an $\mathrm{MCTM_{MS}}$ mismatch lower than $\Delta \mathrm{t_{MODg}}$. The control algorithm, whose end is $\mathrm{MCTM_{MS}}$ ticks before the start of the next switching period and does not close before the next switching period. Therefore, the provided control actions are not related to the last CA call, coming from the previous iteration: a time of T_{ca} is then lost.

Moreover, since $s_2=0$, the $ATM_{critical}$ is not lower than Δt_{ACQ}, the $ATM_{critical}$ is related to the CA iteration that provides the modulating signals to the switching period that will start after a time equal to T_{CA} + $MCTM_{MS}$ ticks. The CA related to the mismatch equal to the $MCTM_{MS}$ will be executed with the consistent measurements' values.

Due to that behavior, it becomes necessary to evaluate the maximum delay, which is related to the switching period affected by the loss of one T_{CA} or to the switching iteration which exhibits a delay larger to one T_{SW} or more. Accordingly, it requires the definition of new parameters as follows.

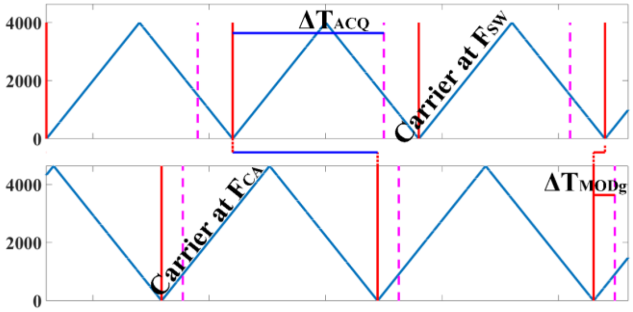


Figure 5. Graphical illustration of Case A.

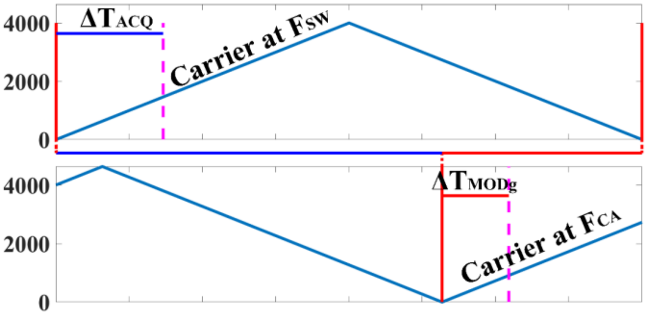


Figure 6. Graphical illustration of Case B.

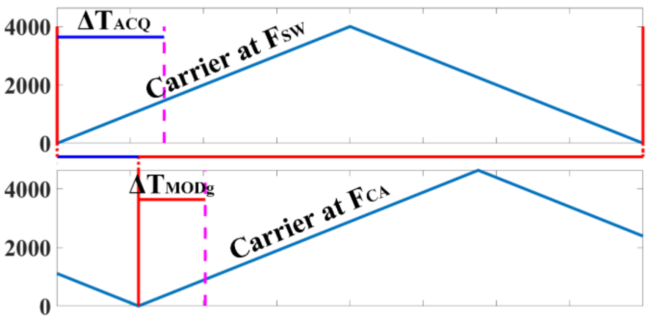


Figure 7. Graphical illustration of Case C.

The idea behind is to recognize if there are *ATM* smaller than Δt_{ACQ}. Therefore, the first step is to calculate the smallest *ATM*, that will be related to the largest mismatch still lower than the switching period. Hence, the Mismatch immediately Smaller than T_{SW} (MST_{SW}) must be introduced, and achievable as in (9):

$$MST_{SW}=\left(\left\lfloor\frac{(1-s_3)\cdot T_{SW}+s_3\cdot T_{CA}}{MTM\big|_{n_{sh}=0}}\right\rfloor-1\right)\cdot MTM\big|_{n_{sh}=0}+ \\ +MTM\big|_{n_{sh}} \tag{9}$$

Where the selector s_3 can be defined as

$$s_3=\frac{|M-N|+(M-N)}{2(M-N)+\left(1-\left\lceil\frac{|M-N|}{M-N}\right\rceil\right)} \tag{10}$$

The s_3 parameter is equal to *1* when *M* is greater than *N* (i.e. T_{SW} results greater than T_{CA}). On the contrary, s_3 is equal to *0* when M is smaller than or equal to *N* (i.e. T_{SW} is lower than or equal to T_{CA}). The evaluation of the MST_{SW} parameter in necessary to understand if the complementary value of MST_{SW} with respect to T_{SW} is greater, equal, or lower than the measures acquisition time (Δt_{ACQ}).
Therefore, the *y* parameter is introduced as in (11):

$$y=\Delta t_{ACQ}-(T_{SW}-MST_{SW}) \tag{11}$$

When the quantity *y* is negative, all the CA iterations will receive samples from the current sampling iteration. On the contrary, when *y* is positive or equal to 0, at least a CA iteration will receive delayed measurements from the previous call. There is at least a CA iteration lagging the acquiring process. However, after defining the MST_{SW} and evaluating *y*, the next step is to evaluate the minimum mismatch for which the delay in the acquisition process will occur (*Minimum Mismatch Acquiring Lag, MMAL*). *MMAL* can be directly achieved by (12):

$$MMAL=MST_{SW}-\left\lfloor\frac{\frac{|y|+y}{2}}{MTM\big|_{n_{sh}=0}}\right\rfloor MTM\big|_{n_{sh}=0} \tag{12}$$

Notice that:

$$\begin{cases}\left.\dfrac{|y|+y}{2}=0\right|_{\Delta t_{ACQ}\le T_{SW}-MST_{SW}} \\ \left.\dfrac{|y|+y}{2}>0\right|_{\Delta t_{ACQ}>T_{SW}-MST_{SW}}\end{cases} \tag{13}$$

Therefore, it will result:

$$\begin{cases}MMAL=MST_{SW}\big|_{\Delta t_{ACQ}-(T_{SW}-MST_{SW})\le MTM|_{n_{sh}=0}} \\ MMAL<MST_{SW}\big|_{\Delta t_{ACQ}-(T_{SW}-MST_{SW})>MTM|_{n_{sh}=0}}\end{cases} \tag{14}$$

Hence, the ATM_{MIN}, which is the complementary part of the *MMAL* with respect to T_{SW}, can be defined as in

$$ATM_{MIN}=T_{SW}-MMAL \tag{15}$$

Evaluated the ATM_{MIN}, it is required the introduction of an additional selector s_4, which will be compared to the Δt_{ACQ} time:

$$s_4=\left\lceil\frac{\left\lfloor\dfrac{\Delta t_{ACQ}}{ATM_{MIN}+1-\left\lceil\dfrac{ATM_{MIN}}{T_{SW}}\right\rceil}\right\rfloor}{\Delta t_{ACQ}}\right\rceil \tag{16}$$

When s_4 is equal to *0*, ATM_{MIN} is greater than Δt_{ACQ} that is, all CA iterations will receive the samples coming from the last sampling period started. When s_4 is equal to *1*, the ATM_{MIN} will be lower than or equal to the Δt_{ACQ} time. Hence, the CA iteration will receive the samples not from the last sampling iteration started but from the previous one with a consequent delay in the acquisition process.
It is also necessary to understand whether there is, and if so, its value, a mismatch greater than T_{SW}, whose complementary part is equal to the ATM_{MIN} just calculated. This mismatch will be introduced as $CTM_{ATM_{MIN}}$ and it can be evaluated as in (17):

$$CTM_{ATM_{MIN}}=\left\lfloor\frac{MaxTM\big|_{n_{sh}}-MMAL}{T_{SW}}\right\rfloor\cdot T_{SW}+MMAL \tag{17}$$

Finally, the *MaxTotDelay* can be obtained using (18):

$$\begin{aligned}&MaxTotDelay=(s_2+1)T_{SW}+ \\ &+\left\lfloor\frac{s_1(MCTM_{MS})+(1-s_1)MCTM}{T_{SW}}\right\rfloor T_{SW}+ \\ &+\left[(s_4+1)T_{SW}+\left\lfloor\frac{CTM_{ATM_{MIN}}}{T_{SW}}\right\rfloor T_{SW}\right]s_3(1-s_2)s_1s_4+ \\ &-\left[(s_2+1)T_{SW}+\left\lfloor\frac{s_1(MCTM_{MS})+(1-s_1)MCTM}{T_{SW}}\right\rfloor T_{SW}\right]\cdot \\ &\cdot s_3(1-s_2)s_1s_4\end{aligned} \tag{18}$$

This article has been accepted for publication in IEEE Open Journal of Industry Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJIA.2024.3366415

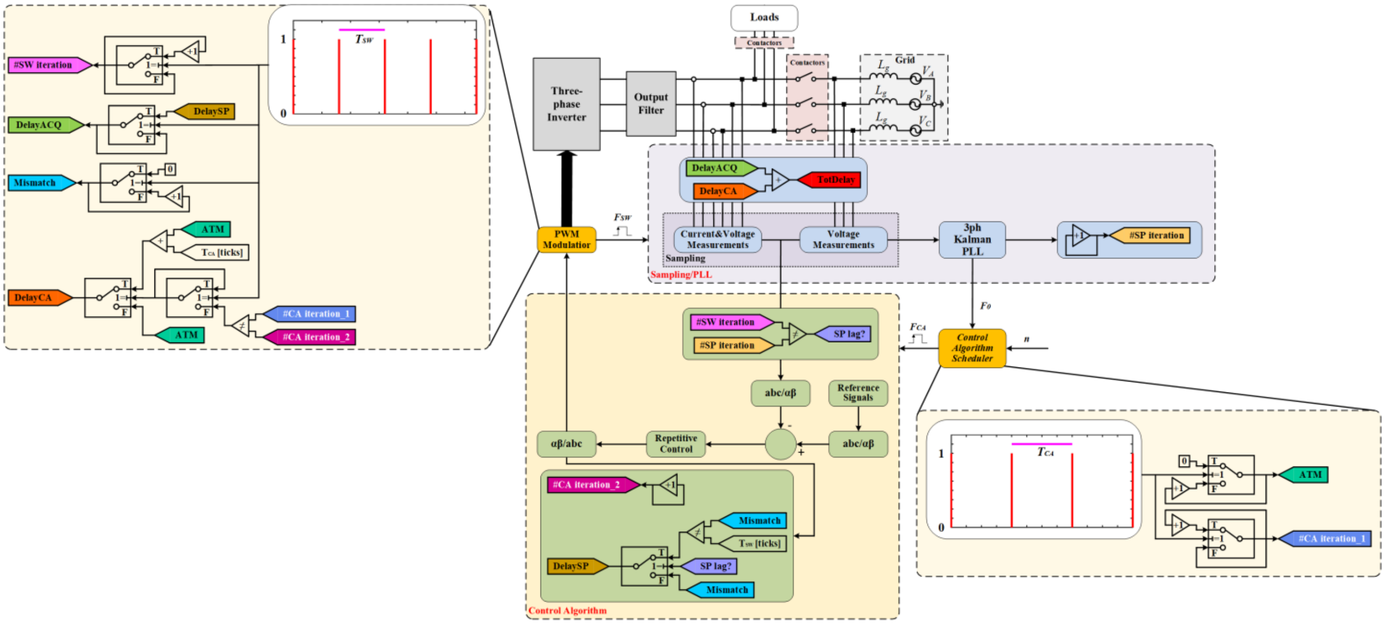
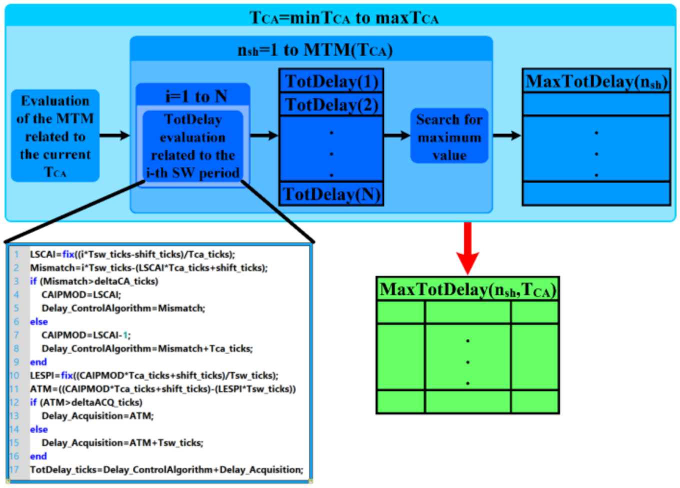


Figure 8. Graphical illustration of the subsystems for control algorithm real-time delay evaluation.

The equation (18) can be considered as the general formula disregarding the specific case under investigation to obtain the maximum delay, which must be considered in the complete repetitive control action. The value can be computed online, in real-time when the algorithm is running, allowing the controller to adapt its gains appropriately. The following steps will be related to the demonstration of the correctness of the proposed analysis by performing a comparison with the direct evaluation of the delays from the FPGA.

III. Comparison with the 2nd Method

To verify (18), a possible calculation method for the *MaxTotDelay* has been presented in [18], which has been updated in the present work to take into account the n_{sh} value, providing a full general expression.



```
LSCAI=fix((i*Tsw_ticks-shift_ticks)/Tca_ticks);
Mismatch=i*Tsw_ticks-(LSCAI*Tca_ticks+shift_ticks);
if (Mismatch>deltaCA_ticks)
   CAIPMOD=LSCAI;
   Delay_ControlAlgorithm=Mismatch;
else
   CAIPMOD=LSCAI-1;
   Delay_ControlAlgorithm=Mismatch+Tca_ticks;
end
LESPI=fix((CAIPMOD*Tca_ticks+shift_ticks)/Tsw_ticks);
ATM=((CAIPMOD*Tca_ticks+shift_ticks)-(LESPI*Tsw_ticks))
if (ATM>deltaACQ_ticks)
   Delay_Acquisition=ATM;
else
   Delay_Acquisition=ATM+Tsw_ticks;
end
TotDelay_ticks=Delay_ControlAlgorithm+Delay_Acquisition;
```

Figure 9. *MaxTotDelay* alternative calculation.

The workflow is reported in Figure 9. It can be noted that in the shown alternative method, the *MaxTotDelay* correlated to a specific T_{CA} and a certain n_{sh}, is evaluated by choosing the maximum value among the *TotDelay* relative to all the *N* switching periods. By varying the T_{CA}, and for each T_{CA} by varying the value of n_{sh} from *0* to T_{SW} [ticks], this alternative technique provides a matrix where one column contains all possible *MaxTotDelay* values which, for a specific T_{CA}, will occur as a function of the current n_{sh}.

Therefore, by comparing the data contained in the table obtained with the values provided by (18), it can be demonstrated that the proposed approach provides absolutely consistent results, regardless of the chosen T_{CA} and n_{sh} values. In particular, the T_{CA} varied from *20 Hz* to *80 Hz* with a step of *0.01 Hz*. Furthermore, as already mentioned, for each T_{CA} also n_{sh} was varied consistently from *0* to $MTM\big|_{n_{sh}=0}$.

IV. Experimental Results

The proposed analytical approach has been verified by a specific Hardware-In-the-Loop real-time testbed based on the OP4610XG high-fidelity solver and a suitable experimental test ring. Hence, the control structure has been implemented in an industrial grade control platform as it would be in a real testing rig. In fact, there are no differences in the control structure between HIL and the experimental setup, which is the basis of the proposed analysis. The variable frequency repetitive controller has been implemented on a Xilinx Artix7 FPGA, which was made available by the PED-Board® control board, allowing it to be fully programmable by the LabVIEW development environment. Thanks to the FPGA reconfigurable capabilities, which allows to create specific counters that can be placed in the exact code position, the quantities already defined in the analytical derivation have been directly evaluated by an accurate delay measurement and analysis directly implemented in the FPGA. Hence, the above expressions were not directly used for delay evaluation. As proof, the FPGA results have been compared with the results achieved by the reported analytical derivation. The proposed validation methodology allows to use in a future implementation, the illustrated closed form expressions, implemented in a DSP or µProcessor, relieving the

This article has been accepted for publication in IEEE Open Journal of Industry Applications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJIA.2024.3366415

FPGA space which was dedicated to the measurement of the delays (i.e. about 30% of the total required space). The results have been obtained by adding specific FPGA digital counters having the purpose to evaluate all the delays as shown in Figure 8. The same code structure is present in both HIL and experimental test-rig. It can be imagined that those parts will be removed in a final implementation, taking advantage of the proposed analytical expressions, which can also be implemented on a μProcessor, saving the FPGA space.

The setup used for both HIL verifications and the experimental campaign are shown respectively in Figure 10 and Figure 11. In the testbed the variable frequency AC-side has been emulated by a suitable 3-phase inverter with output voltage control.

The analytical method has been illustrated and initially compared with the results presented in [18]. The full validation of the proposed closed-form expressions requires the direct comparison with the results achieved when control software runs on an industrial grade control platform. In the experimental setup, the fundamental frequency has been forced to be *49.91265 Hz* and imposing n_{sh} equal to *0*. Then, the trend for the *Mismatches* and for the normalized *TotDelay* have been directly achieved as reported in Figure 12.

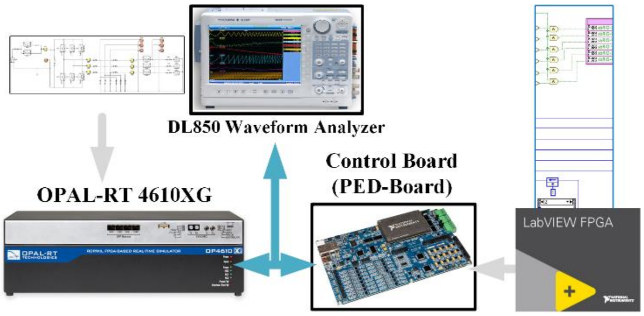


Figure 10. HIL arrangement.

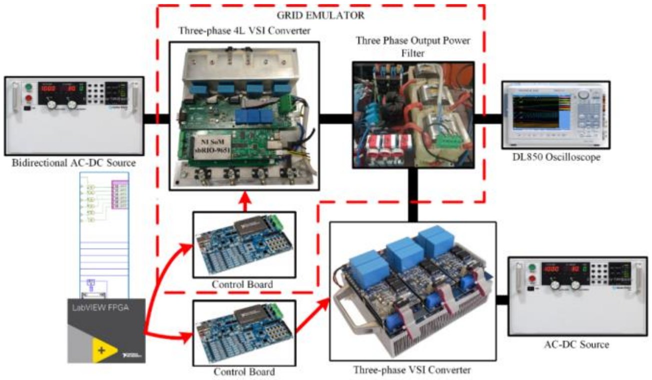

GRID EMULATOR
Three-phase 4L VSI Converter
Three Phase Output Power Filter
NI SoM sbRIO-9651
Bidirectional AC-DC Source
DL850 Oscilloscope
LabVIEW FPGA
Control Board
Control Board
Three-phase VSI Converter
AC-DC Source

Figure 11. Experimental setup.

After extracting the maximum value from the trend related to the *TotDelay*, the value has been compared with the result from (18). By varying n_{sh} and F_0, it is then possible to verify the presented method, having the possibility to evaluate the *MaxTotDelay* before it appears in the upcoming iterations. In the next, there will be no distinction between the HIL and the pure experimental results, being the control platform the same, with the same control software. Hence, from the FPGA code updated according to Figure 8, it was possible to obtain the maximum delay for a given fundamental frequency F_0 and for a specific n_{sh} coefficient. The reported maximum delay is the quantity seen from the control point of view, directly affecting the system stability and performance.

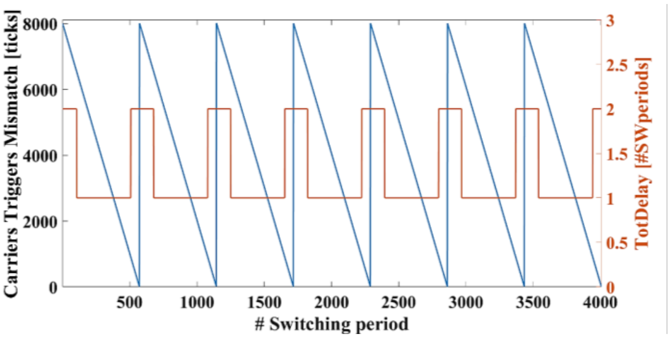


Figure 12. Mismatches and *TotDelay(#)* trend analytically evaluated

In the following figures, the validity of the proposed closed-form expressions and analytical derivation will be proved by comparing the results with the FPGA direct measurements. The fundamental frequency F_0 was varied between *20 Hz* and *80 Hz* with a step of *0.01 Hz* to include non-integer values. Moreover, starting from zero, the n_{sh} quantity has been varied from 0 to 8000 (with step of 1 ticks), to test all the possible scenarios that could arise, randomly changed, whereas only some results have been reported. Initially, as shown from Figure 13 the n_{sh} quantity has been set to zero and the delays have been calculated by the proposed expressions and from the FPGA counters, for different fundamental frequencies. It can be noticed that the results are perfectly overlapped, which proves the validity of the proposed mathematical approach.

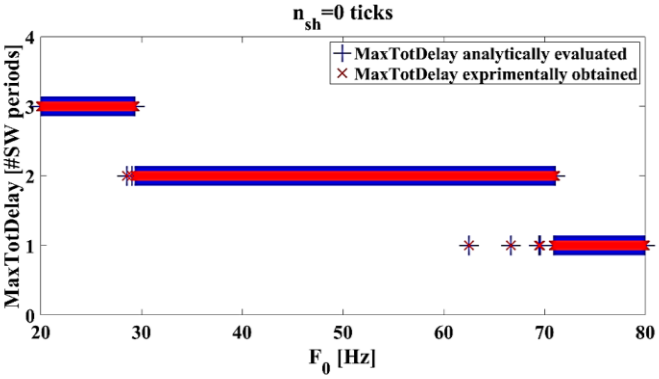


Figure 13. FPGA and analytical *MaxTotDelay* evaluation for $n_{sh}=0$ with a frequency resolution of *0.01 Hz*.

For a more general validation, the results obtained using mathematical expressions or evaluation via the FPGA counters are also shown, for two random values of n_{sh}. The results are shown respectively in Figure 14 for n_{sh}*=1756 ticks* and in Figure 15 for n_{sh}*=3795 ticks*. In both cases, the results are perfectly coherent, confirming the correctness of the proposed solution. As can been noticed, varying the n_{sh} the changing of the *MaxTotDelay*, for a certain F_0, can be realized. However, the analytic proposed method effectiveness has been proved, leading, as already mentioned, to the possibility to evaluate the *MaxTotDelay* before it appears while the software is running.That is a very fundamental feature. In fact, the control algorithm will be able to adjust its control parameters by evaluating a few simple equations, as soon as the new fundamental frequency occurs, without the need to wait for the switching periods. Hence, with a specific prediction of the future delay due to frequency variation.

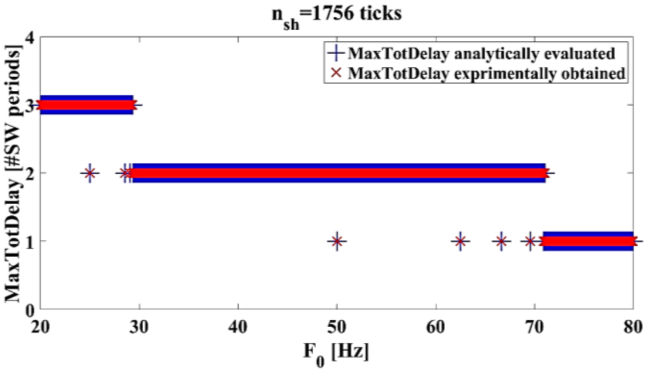


Figure 14. FPGA and analytical *MaxTotDelay* evaluation for n_{sh}*=1756 ticks* with a frequency resolution of *0.01 Hz*.

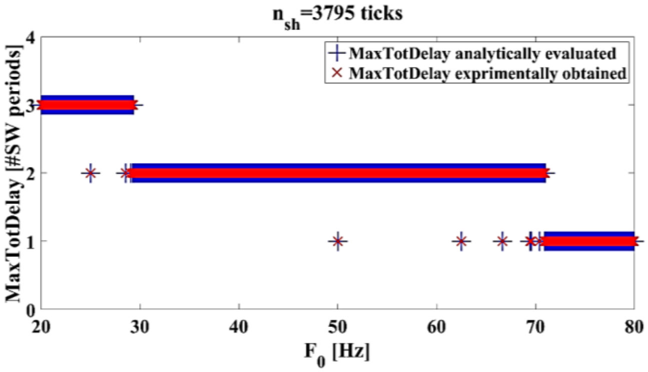


Figure 15. FPGA and analytical *MaxTotDelay* evaluation for n_{sh}*=3795 ticks* with a frequency resolution of *0.01 Hz*.

Figure 16(a) shows the phase voltages emulated via the 4-leg inverter shown in Figure 11. The fundamental frequency has been set at *48 Hz*. The carriers are represented below where the one at the top is related to the switching frequency, which is imposed and fixed, while the one at the bottom is related to the control frequency proportional to the delay line and the frequency fundamental. These information are used to calculate the *MaxTotDelay*. Coherently, Figure 16(b) shows the same variables where the grid frequency has been moved to *52 Hz*.

Results for a wider and non-integer grid frequency variation have been reported respectively in Figure 17a and Figure 17b, where the frequency of the emulated grid has been changed from *46.3Hz* to *60.4Hz*.

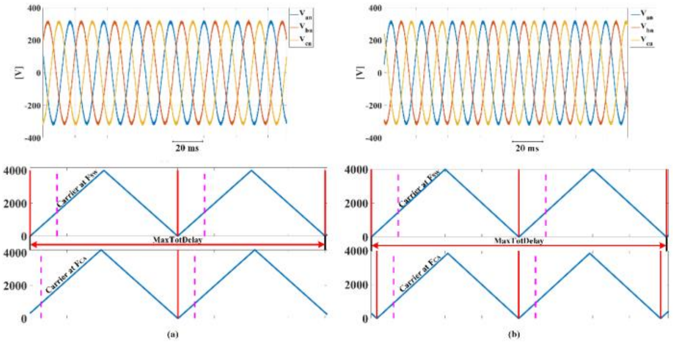


Figure 16. Phase voltages, switching and control carriers signals for a narrow frequency variation.

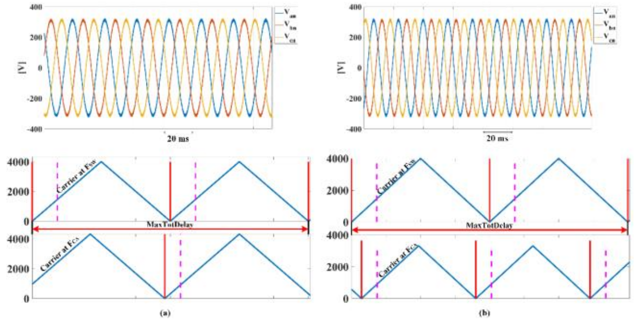


Figure 17. Grid voltages and carriers for both switching and control loops.

CONCLUSIONS

The Repetitive Control structure operated at variable frequency grants to take advantage of excellent tracking of sinusoidal references. This permits to go beyond RC's weakness related to variable frequency operations, without any noticeable effects in terms of increased computational efforts or memory usage, and still operating with a constant delay line (i.e. straightforward implementation and suitable for FPGA systems). However, unlinking the Repetitive Control operation to the main PWM scheduler, which represents the main scheduler of most common control structures, leads to a time mismatch between the control algorithm and PWM modulation, which needs to be evaluated to later move on a suitable RC's gain tuning. A deeper investigation has been reported with comprehensive experimental validation. The illustrated complete analytical derivation provides the closed form expressions for real-time delay evaluation applicable to any general operating case. The analytical formulas take the advantage to determine the amount of delay as soon as the PLL (i.e. or any frequency estimation algorithm or internal reference for AC voltage generators) provides the updated frequency, without the need to wait for the counters to be updated, which will introduce an additional delay in updating the control parameters.

ACKNOWLEDGEMENT

This work was supported by the Regione Lazio of Italy under the Grant "Progetti di Gruppi di Ricerca 2020, N. 36774".

REFERENCES

[1] K. Chang, I. Shim e G. Park, «Adaptive Repetitive Control for an Eccentricity Compensation of Optical Disk Drivers,» *IEEE Transactions on Consumer Electronics,* vol. 52, p. 445–450, May 2006.

[2] S. T. Navalkar, E. van Solingen e J.-W. van Wingerden, «Wind Tunnel Testing of Subspace Predictive Repetitive Control for Variable Pitch Wind Turbines,» *IEEE Transactions on Control Systems Technology,* vol. 23, p. 2101–2116, November 2015.

[3] A. Lunardi, E. Conde, J. Assis, L. Meegahapola, D. A. Fernandes e A. J. Sguarezi Filho, «Repetitive Predictive Control for Current Control of Grid-Connected Inverter Under Distorted Voltage Conditions,» *IEEE Access,* vol. 10, p. 16931–16941, 2022.

[4] M. Tang, M. di Benedetto, S. Bifaretti, A. Lidozzi and P. Zanchetta, "State of the Art of Repetitive Control in Power Electronics and Drive Applications," *IEEE Open Journal of Industry Applications*, vol. 3, p. 13–29, 2022.

[5] S. Sahoo, S. Prakash e S. Mishra, «Power Quality Improvement of Grid-Connected DC Microgrids Using Repetitive Learning-

Based PLL Under Abnormal Grid Conditions,» *IEEE Transactions on Industry Applications,* vol. 54, p. 82–90, January 2018.

[6] C. Blanco, F. Tardelli, D. Reigosa, P. Zanchetta e F. Briz, «Design of a Cooperative Voltage Harmonic Compensation Strategy for Islanded Microgrids Combining Virtual Admittance and Repetitive Controller,» *IEEE Transactions on Industry Applications,* vol. 55, p. 680–688, January 2019.

[7] G. A. Ramos, R. I. Ruget e R. Costa-Castello, «Robust Repetitive Control of Power Inverters for Standalone Operation in DG Systems,» *IEEE Transactions on Energy Conversion,* vol. 35, p. 237–247, March 2020.

[8] M. Tang, A. Gaeta, A. Formentini e P. Zanchetta, «A Fractional Delay Variable Frequency Repetitive Control for Torque Ripple Reduction in PMSMs,» *IEEE Transactions on Industry Applications,* vol. 53, p. 5553–5562, November 2017.

[9] Y. Liu, S. Cheng, B. Ning e Y. Li, «Robust Model Predictive Control With Simplified Repetitive Control for Electrical Machine Drives,» *IEEE Transactions on Power Electronics,* vol. 34, p. 4524–4535, May 2019.

[10] A. L. Julian, G. Oriti, C. Ji and P. Zanchetta, "Single-Phase Energy Management System Operating in Islanding Mode With Repetitive Control and Active Damping," *IEEE Transactions on Industry Applications,* vol. 54, p. 5163–5172, September 2018.

[11] H. Geng, Z. Zheng, T. Zou, B. Chu and A. Chandra, "Fast Repetitive Control With Harmonic Correction Loops for Shunt Active Power Filter Applied in Weak Grid," *IEEE Transactions on Industry Applications,* vol. 55, p. 3198–3206, May 2019.

[12] G. Escobar, P. Mattavelli e M. F. Martinez-Montejano, «Modifications to repetitive-based controllers using FIR filters for practical implementation,» in *2009 35th Annual Conference of IEEE Industrial Electronics*, 2009.

[13] M. Sun, Y. Wang e D. Wang, «Variable-Structure Repetitive Control: A Discrete-Time Strategy,» *IEEE Transactions on Industrial Electronics,* vol. 52, p. 610–616, April 2005.

[14] Z. Cao e G. F. Ledwich, «Adaptive repetitive control to track variable periodic signals with fixed sampling rate,» *IEEE/ASME Transactions on Mechatronics,* vol. 7, p. 378–384, September 2002.

[15] E. Kurniawan, Z. Cao and Z. Man, "Design of Robust Repetitive Control With Time-Varying Sampling Periods," *IEEE Transactions on Industrial Electronics,* vol. 61, p. 2834–2841, June 2014.

[16] J. Ye, L. Liu, J. Xu and A. Shen, "Frequency Adaptive Proportional-Repetitive Control for Grid-Connected Inverters," *IEEE Transactions on Industrial Electronics,* vol. 68, p. 7965–7974, September 2021.

[17] M. Zhu, Y. Ye, Y. Xiong e Q. Zhao, «Multibandwidth Repetitive Control Resisting Frequency Variation in Grid-Tied Inverters,» *IEEE Journal of Emerging and Selected Topics in Power Electronics,* vol. 10, p. 446–454, February 2022.

[18] A. Faro, M. d. Benedetto, A. Lidozzi and L. Solero, "Constant Delay-Line Repetitive Control Analysis for Variable Frequency Operation," *IEEE Open Journal of Power Electronics, vol. 3, pp. 481-491, 2022.*

[19] H. Lin, X. Guo, D. Chen, S. Wu and G. Chen, "A Frequency Adaptive Repetitive Control for Active Power Filter With 380V/75A SiC-Inverter," *IEEE Transactions on Industry Applications,* vol. 58, p. 5469–5479, July 2022.

[20] Z. Liu, K. Zhou, Y. Yang, J. Wang e B. Zhang, «Frequency-Adaptive Virtual Variable Sampling-Based Selective Harmonic Repetitive Control of Power Inverters,» *IEEE Transactions on Industrial Electronics,* vol. 68, p. 11339–11347, November 2021.

[21] Z. Liu, B. Zhang, K. Zhou e J. Wang, «Virtual Variable Sampling Discrete Fourier Transform Based Selective Odd-Order Harmonic Repetitive Control of DC/AC Converters,» *IEEE Transactions on Power Electronics,* vol. 33, p. 6444–6452, July 2018.

[22] S. Baek, Y. Cho and J.-S. Lai, "Average Periodic Delay-Based Frequency Adaptable Repetitive Control With a Fixed Sampling Rate and Memory of Single-Phase PFC Converters," *IEEE Transactions on Power Electronics,* vol. 36, p. 6572–6585, June 2021.

[23] M. di Benedetto, A. Faro, L. Bigarelli, A. Lidozzi and L. Solero, "Constant Delay-Line Frequency Adaptive Repetitive-Resonant Control for Grid-Tied and Intentional Islanding Operations," *IEEE Transactions on Industry Applications,* p. 1–10, 2022.

Alessandro Faro received his master’s degree in mechanical and industrial Engineering, in 2020, at the ROMA TRE University, Rome, Italy. Since 2020 he has been working for the PhD in Mechanical and Industrial Engineering at the Department of Engineering, of the ROMA TRE University, C-PED. His research activity concerns the control aspect for medium voltage power converters in railway applications. This activity involves the use of hardware-in-the-loop real-time simulators and FPGA-based control platforms.

Alessandro Lidozzi received the Electronic Engineering degree and the Ph.D. degree from the Roma Tre University, Rome, Italy, in 2003 and 2007, respectively.

From 2010 to 2017, he was a Researcher with the Department of Engineering, ROMA TRE University; where, since 2017, he has been Associate Professor. His research interests are mainly focused on power converters and electrical drives real-time modeling and time critical control strategies, medium voltage power conversion systems and development of high-performance control platforms based on combined DSP-FPGA architectures. He is the co-inventor of the PED-Board ecosystem. He serves as Associate Editor of IEEE Transaction on Industry Applications.

Marco di Benedetto received the M.Eng. degree in Electronic Engineering from the University of Roma Tor Vergata, Rome, Italy, in 2014 and the Ph.D. in Mechanical and Industrial Engineering from the Roma Tre University, Rome, Italy, in 2018. In October 2022, he was appointed assistant professor at Roma Tre University's Department of Industrial, Electronic, and Mechanical Engineering. His research interests mainly focus on hardware and FPGA control design for power converter topologies. He serves as the IEEE IAS Industrial Power Converter Committee (IPCC) secretary. He serves as an Associate Editor of the IEEE Open Journal on Industry Applications and Electrical Systems in Transportation (IET).

Luca Solero (Senior Member, IEEE) received the M. Eng degree in electrical engineering degree from the University of Rome “La Sapienza,” Italy, in 1994. Since 1996 he has been with the Department of Engineering, University ROMA TRE where he currently is a Full Professor in charge of teaching courses in the fields of Power Electronics and Electric Drives. He has authored or co-authored more than 200 peer-reviewed papers. His current research interests include design and control of power electronic converters and drives, multi-port and multi-level converters. Prof. Solero is the Past Chair of the IEEE-IAS Industrial Power Converter Committee IPCC (2020-2021). He serves as an Associate Editor for the IEEE Transactions on Industry Applications and for the IEEE Journal of Emerging and Selected Topics in Industrial Electronics. He is the Deputy Editor in Chief for the IEEE Open Journal of Industry Applications. Since 2022 he is also the Vice-Chair of the IEEE-IAS Industrial Power Conversion Systems Department IPCSD. Prof. Solero was the recipient of the IAS-IPCC 2019 First Prize Paper Award and of the IAS-IPCC 2022 Third Prize Paper Award.