Discrete-time implementation of second order generalized integrators for grid converters

F.J.Rodríguez, E.Bueno
Department of Electronics.
Alcalá University.
Alcalá de Henares (Madrid) - Spain
emilio@depeca.uah.es

M.Aredes, L.G.B.Rolim
Power Electronics Laboratory
Federal University of Rio de Janeiro
Rio de Janeiro – Brazil
aredes@coe.ufrj.br

F.A.S.Neves, M.C. Cavalcanti
Universidade Federal de Pernambuco -
Departamento de Engenharia Eletrica e
Sistemas de Potencia, Recife – PE, Brazil
fneves@ufpe.br

***Abstract* – Second order generalized integrators (SOGI) have been proposed to obtain zero steady-state error using stationary frames in grid-converters applied to active rectifiers, active filters, uninterruptible power supplies and distributed generation. These integrators have been also included in algorithms of grid synchronization, detection of sequences, harmonic compensation, etc. Due to all these algorithms are implemented over digital processors, it is very important to make a correct discretization to optimize their behaviour. This work is focused in this topic. An exhaustive study about different methods to implement the SOGIs in the discrete-time is achieved, proposing a common structure for the different SOGI applications, including proposals of C program and presenting a comparative analysis about the execution time using different digital structures.**

I. INTRODUCTION

Research on current control for grid converters has been one of the most intensive activities in the last years. In the case of three-phase voltage source converters, one of the most common current controller techniques is based on stationary reference frames ($\alpha\beta$-frames). Since the control variables are sinusoidal waveforms, proportional resonant controllers (PR), which is based on a proportional gain plus a second order generalized integrator (SOGI), are the solution more used, due to its capability of eliminating the steady state error under regulating sinusoidal signals. In a single-phase system, the use of a rotating frame is not possible unless a virtual system is coupled to the real one in order to simulate a two-axis environment. Here, the use of resonant controllers is easier and leads to less computational burden.

Besides using the SOGIs to current control, there are numerous technical works where for grid converters their application has been extended: (1) to eliminate harmonics because this controller acts on a very narrow band around its resonant frequency, ω_O; (2) to generate quadrature-signals (SOGI-QSG) and to detect grid sequences (DSOGI-QSG) [1]; (3) to synchronize the grid converter with the grid [1] [2]; and (4) to detect multiple grid frequencies, which is obtained tuning a DSOGI-QSG [1] to each grid frequency that is desired to detect. This last structure receives the name MSOGI-QSG.

In industrial applications, these algorithms are implemented in digital processors, such as microcontrollers, floating-point DSPs, fixed-point DSPs, FPGAs, etc. In order to the digital implementation response corresponds to the theoretical studies, generally developed in the continuous time-domain, it is very important to make a correct discretization. Other important aspects to obtain a optimize behaviour are the constant algorithm tuning, saturation structure, the implementation, etc.

For it, the paper is organized as follows. Section II presents a brief explanation about the SOGIs and SOGI-QSG in the continuous time-domain. Section III proposes a method of discretization based on transforming the continuous time-domain transfer-functions. Section IV presents other discretization methods based on discretizing individually each integrator of the SOGI. Section V explains how to tune the constants of the proportional resonant controller to control of grid converter connected to the grid through and L-filter when the SOGI proposed in Section IV is used. Finally, execution times of the proposed discrete implementation using different digital structures are presented in Section VI, followed by conclusions in Section VII.

II. SECOND ORDER GENERALIZED INTEGRATORS IN THE CONTINUOUS TIME-DOMAIN

According to the proposal presented in [3] and [4], the transfer function of a second order generalized integrator (SOGI) for a single sinusoidal signal is:

$$G(s) = \frac{2s}{s^2 + \omega_o^2} \tag{1}$$

A typical converter current control system using the stationary-frame generalized integrator can be decomposed into a positive sequence signal system and a negative sequence signal system, as it was indicated in [3] and [4]. Another important application of the SOGIs is to generate quadrature-signals. For this, some works as, for example, [1] and [5] proposed a little modification respect to the original topology. It is represented in Fig. 1, and the new mathematical expression is:

$$G_{SOGI}(s) = \frac{s\omega_o}{s^2 + \omega_o^2} \tag{2}$$

The new transfer function has two poles placed in $\pm j\omega_o$ and a zero in origin, exactly equal that the eq. (1). The only different between the two expressions is the gain. So, this structure is more general and can be used both in current control of grid converters taking account the changes in the gain of eq. (2), and, also, in tasks related to synchronization

with the grid detailed in the previous paragraph.

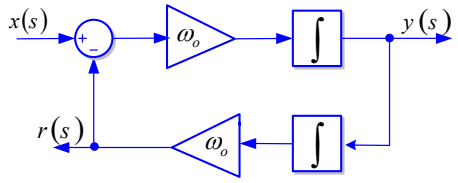

Fig. 1.Proposal of SOGI in continuous time-domain.

The function of generating quadrature-signals is very useful to single-phase PLLs, to calculate positive and negative sequences and to decompose the grid voltage in its harmonics (MSOGI-QSG). The basic structure, denominated SOGI-QSG (SOGI for quadrature-signals generation) is represented in Fig. 2, and as is observed, the basic element is a SOGI [4]. The transfer-functions in the continuous time-domain are (3) and (4) and their Bode diagrams are represented in Fig. 3, for $\omega_o = 2\pi 50\, {}^{rad}\!/_{s}$ and $k_S = 1$.

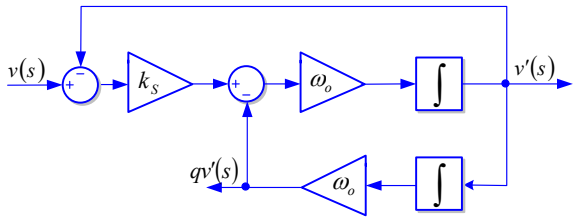

Fig. 2.SOGI-QSG.

$$\frac{v'}{v}(s) = \frac{k_S \omega_o s}{s^2 + k_S \omega_o s + \omega_o{}^2} \tag{3}$$

$$\frac{qv'}{v}(s) = \frac{k_S \omega_o{}^2}{s^2 + k_S \omega_o s + \omega_o{}^2} \tag{4}$$

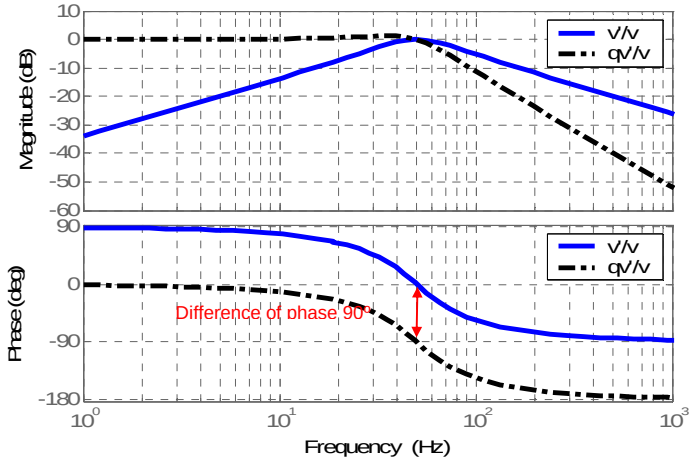

Fig. 3. Bode Diagram of a QSG-SOGI based on the continuous time-domain SOGI.

${}^{v'}\!/_{v}(s)$ behaves as a band-pass filter, and the bandwidth is determined by the value of k_S; whereas ${}^{qv'}\!/_{v}(s)$ behaves as a low-pass filter. For this reason, the SOGI used in current controllers to discriminate frequencies has a very good behavior because the used output is $y(s)$ (Fig. 1). On the other hand, the structure SOGI-QSG is ideal to generate quadrature-signals, and, with two SOGI-QSG structures it is possible to implement a detector of the positive and negative grid sequences. But to discriminate harmonics, even though k_S is very selective, the SOGI-QSG has problems to discriminate the harmonics due to the behavior of low-pass filter of ${}^{qv'}\!/_{v}(s)$, and, so that, in MSOGI-QSGs, it is necessary to include additional elements to make this function.

III. Discrete-Time Implementation of SOGIs based on the Discretization of the Transfer-Functions.

A method to implement in the discrete time-domain consists of developing the continuous time-domain transfer-functions and to discretize them by one of the techniques indicated in Table I. From these and for the SOGI structure (Fig. 1), Table II shows the different transfer functions in z-domain. In the matched pole-zero transformation, k' is adjusted in order to have the same gain for some concrete frequencies in the continuous and discrete time-domains. The implementation in digital processors is achieved with the difference equations obtained from these transfer function. Both for the SOGI and for the SOGI-QSG, the better results are obtained with the Tustin and the ZOH techniques. The main advantage of this method is a very good resolution independently of the sampling time; whereas the main disadvantage is that the difference equations for the SOGI-QSG have to be calculated off-line, especially for ZOH and matched methods, and, then it is not possible to have adaptive response with the grid frequency.

TABLE I
Methods for Discretizing Continuous-Time Transfer Functions

Method	Relation	Method	Relation
Euler Forward	$s = \frac{1 - z^{-1}}{T_S z^{-1}}$	ZOH	$H(z) = (1 - z^{-1}) Z\left[\frac{H(s)}{s}\right]$
Backward Euler	$s = \frac{1 - z^{-1}}{T_S}$	Matched Pole-Zero	$z = e^{T_S s}$
Tustin	$s = \frac{2}{T_S} \frac{1 - z^{-1}}{1 + z^{-1}}$		

IV. Discrete-Time Implementation of SOGIs using Discrete Integrators.

In this section, a method based on discretizing each one of the continuous time-domain integrators of the SOGI of Fig. 1 is presented. The objective is to propose different structures of SOGI in the discrete time-domain that have the same response that the SOGI of Fig. 1. For this, the idea is to compare the output signal ($y(s)$ is the output if the contrary is not indicated), of Fig. 1 SOGI under a step input with amplitude 1 applied in $x(s)$, that is a sinusoidal signal with pulsation ω_o and amplitude 1, with the outputs of the different proposals in the discrete time-domain. Those whose

output coincides will be correct options.

Firstly, Table III shows the mathematical expressions in discrete time-domain and in difference equations of the discrete integrators which will be used in the development of the discrete SOGIs.

TABLE II
TRANSFER FUNCTIONS OF SOGIS

Transfor-mation	z-domain equation
Euler Forward	$H(z)=\dfrac{\omega_o T_S (z-1)}{z^2-2z+\omega_o^2 T_S^2+1}$
Backward Euler	$H(z)=\dfrac{\omega_o T_S}{1+\omega_o^2 T_S^2}\cdot\dfrac{(z-1)z}{z^2-\dfrac{2}{1+\omega_o^2 T_S^2}z+\dfrac{1}{1+\omega_o^2 T_S^2}}$
Tustin	$H(z)=\dfrac{2\omega_o T_S}{4+\omega_o^2 T_S^2}\cdot\dfrac{z^2-1}{z^2+\dfrac{2\omega_o^2 T_S^2-8}{4+\omega_o^2 T_S^2}z+1}$
ZOH	$H(z)=\dfrac{\sin(\omega_o T_S)(z-1)}{z^2-2\cos(\omega_o T_S)z+1}$
Matched Pole-Zero	$H(z)=\dfrac{k'(z-1)}{z^2-2\cos(\omega_o T_S)z+1}$

TABLE III
DISCRETE INTEGRATORS

Type of integrator	z-domain equation	Difference equation
Euler Forward	$T(z)=\dfrac{KT_S z^{-1}}{1-z^{-1}}$	$y(n)=KT_S x(n-1)+y(n-1)$
Backward Euler	$T(z)=\dfrac{KT_S}{1-z^{-1}}$	$y(n)=KT_S x(n)+y(n-1)$
Tustin	$T(z)=\dfrac{T_S}{2}\dfrac{K(1+z^{-1})}{1-z^{-1}}$	$y(n)=\dfrac{KT_S}{2}[x(n)+x(n-1)]+y(n-1)$

From the discrete integrators indicated in Table III, different implementations modifying the discrete integrators used in the direct gain and feedback gain are proposed in Table IV. Moreover, in some realizations a computational delay has been included in the third column in series in series with the feedback gain, which models the inherent delay produced in the programming process. Column 4 of Table IV indicates where is represented in Fig. 4, for each option, the temporal response of the output $y(z)$ faced up a step with amplitude 1 in the input $x(z)$. The used sampling time is $T_S = 200\mu s$.

According to Fig. 4, only, the temporal response of the options a, d and g coincides with the response described for the continuous time-domain SOGI. These options are represented through the z-domain equations in Fig. 5, Fig. 6 and Fig. 7, respectively. Of these three options, the programming of Fig. 5 proposal is not easy because there are algebraic loops, and it is rejected. Regarding to the structures of Fig. 6 and Fig. 7, are very similar, the different is only one delay in the output response. Under these circumstances, the topology of Fig. 6 is chosen because has a structure more similar to a classical PI controller.

TABLE IV
SOGI POSSIBLE IMPLEMENTATIONS

Direct Gain	Feedback Gain	Computat. Delay	Figure
Tustin	Tustin	No	Fig. 4.a
Tustin	Tustin	Yes	Fig. 4.b
Euler Backward	Euler Backward	No	Fig. 4.c
Euler Backward	Euler Backward	Yes	Fig. 4.d
Euler Forward	Euler Forward	No	Fig. 4.e
Euler Forward	Euler Forward	Yes	Fig. 4.f
Euler Forward	Euler Backward	No	Fig. 4.g
Euler Forward	Euler Backward	Yes	Fig. 4.h

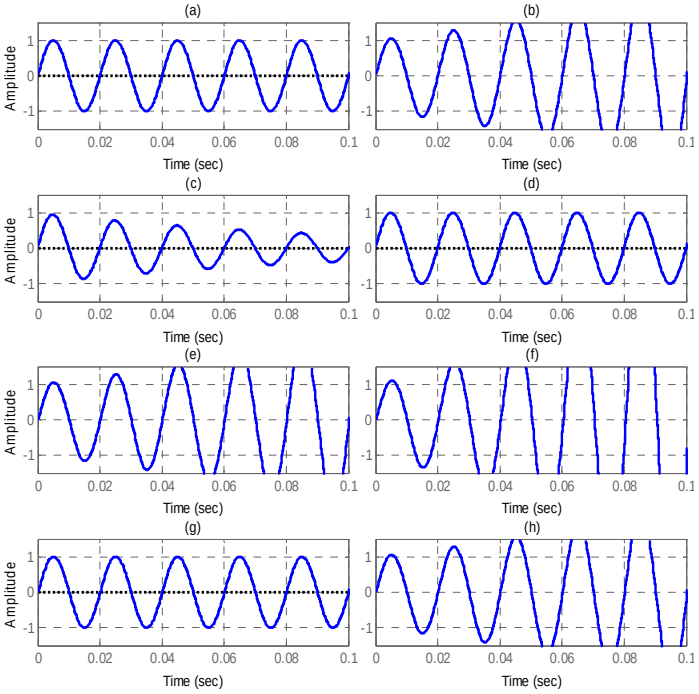

Fig. 4. Temporal response of $y(z)$ under a step in the input $x(z)$ of the proposals of Table IV.

Thus, a correct operation of the SOGI presented in Fig. 6, based on Euler Backward Discrete Integrators, is assured as resonant controller. On the other hand, when the SOGI is working inside of a SOGI-QSG (Fig. 2), in [6] was indicated that the implementation based on Discrete-Time Integrators using Euler method is not correct due to not have an ideal phase of 90-degrees. In Fig. 2 and Fig. 9 are displayed the Bode diagrams of v'/v and qv'/v of the SOGI-QSG in the continuous and discrete time-domain respectively. The objective is that v' and qv' signals be quadrature-signals, which is obtained if the phase shift between them is 90º. As is represented in Fig. 2 and Fig. 9, the phase shift of 90º for $\omega_o = 2\pi 50\, rad/s$ and $T_S = 200\mu s$ is obtained both for the

continuous and for the discrete implementations, so the proposed implementation of Fig. 6 is also useful for SOGI-QSG. About the discrete implementation, it is necessary to take a detail into account. As is shown in Fig. 8, the phase shift is reduced when the frequency is increased. To assure a phase shift of 90º to the syntonization grid frequency in DSOGI-QSG structures ($2k \pm 1$ for single-phase systems or $6k \pm 1$ for three-phase systems) it is necessary that the sampling time is as minimum 50 times higher than the working frequency. This condition, for harmonics above $h = 23$ can suppose sampling times very restrictive for microprocessors or DSPs. So, under these circumstances, to implement the MSOGI-QSG with a parallel processing device as a FPGA it would be a very interesting option.

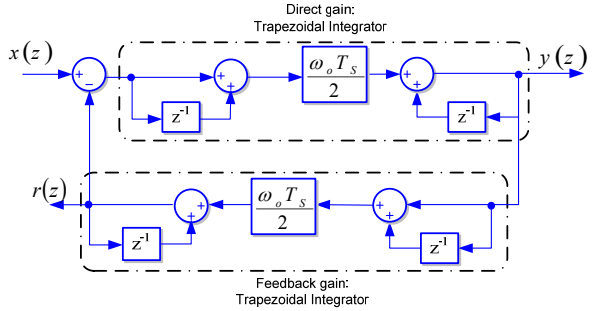

Fig. 5 SOGI based on Tustin integrators.

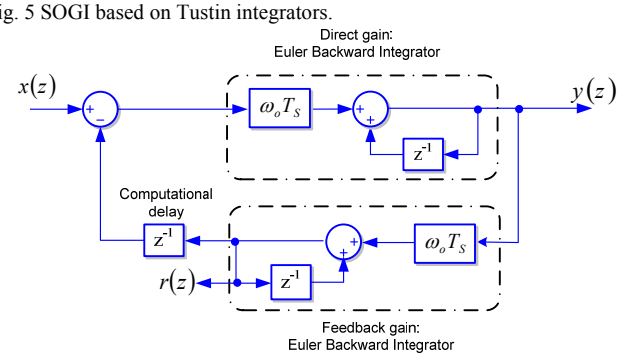

Fig. 6. SOGI based on Euler Backward Integrators and computational delay.

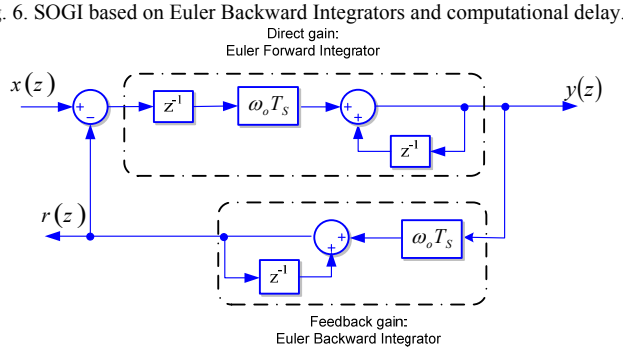

Fig. 7. SOGI based on an Euler Forward Integrator and an Euler Backward Integrator.

To make easy the programming of the SOGI represented in Fig. 6, the modifications indicated in Fig. 9 are applied. Also, the names of the different variables are included in the figure. A source code function implementing the discrete time-domain SOGI, which is executed each sampling period, is described in Fig. 10. It should be self explanatory to anyone with some knowledge of programming. Constants are written in capital letters, whereas declaration of variables, constants, and functions are not shown.

The main advantage of this method is its easy implementation to have an adaptive behaviour with the grid frequency; whereas the main disadvantage is that the resolution depends on the sampling time, and it can require very small sampling times, especially when it is used in DSOGI-QSG and MSOGI-QSG.

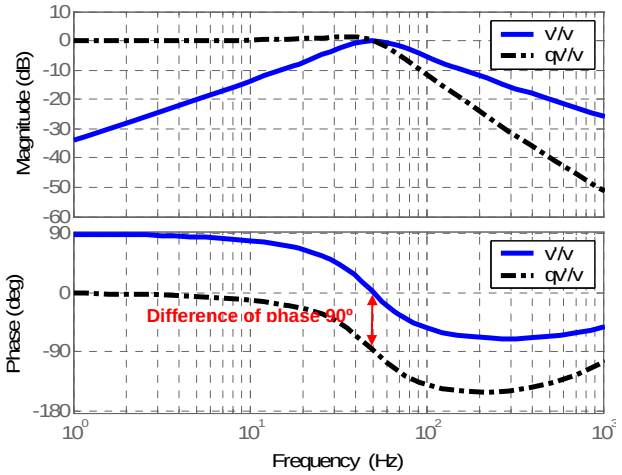

Fig. 8. Bode Diagram of a QSG-SOGI based on the discrete time-domain SOGI of Fig. 6.

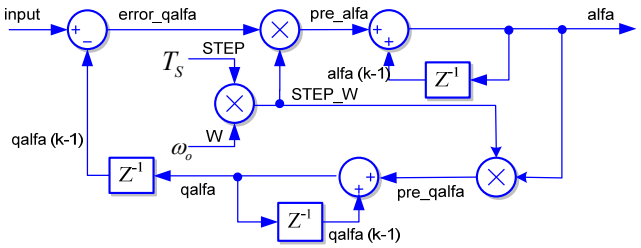

Fig. 9. Proposed SOGI implementation

.

```
static float alfa=0, qalta=0;

for (;;)
{
error_qalfa = input - qalfa;
STEP_W = STEP * W;
pre_alfa = error_qalfa*STEP_W;

alfa += pre_alfa;
pre_qalfa = STEP_W * alfa;
qalfa += pre_qalfa;
}
```

Fig. 10. Source program implementing the SOGI of Fig. 6.

V. TUNING OF THE PROPORTIONAL RESONANT CONTROLLER TO GRID CURRENT CONTROL OF A GRID CONVERTER WITH L-FILTER

In this section, a numerical method to tune the proportional resonant controller based on a discrete time-domain SOGI to control a grid converter with L-filter is proposed. The used SOGI is discretized by the method shown in Section IV. As it was indicated in previous sections, the SOGIs are an ideal solution to current control in stationary reference frames ($\alpha\beta$-frames) of grid converters. In this case, there is not cross coupling between axis and the control

problem can be reduced to one axis α or β giving rise to the control loop for each one of axis represented in Fig. 11.

The transfer function of the SOGI implemented in the previous section (Fig. 6) is:

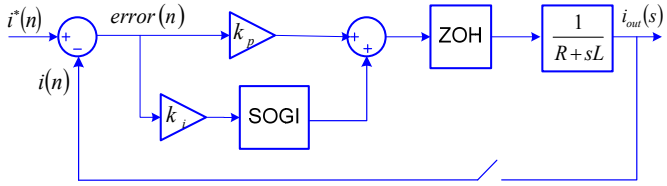

Fig. 11. Current control loop of α or β axis for a grid converter connected to the grid through an L-filter.

$$G_{SOGI}(z)=\frac{k_i\omega_o T_S z(z-1)}{(z-1)^2+\omega_o{}^2 T_S{}^2 z} \tag{5}$$

If $\omega_o{}^2 T_S{}^2 << 2$, which is verified for $T_S < 1.4ms$, if $\omega_o = 2\pi 50\, rad/s$ and supposing a factor much less than 10, the previous equation can be approximated as:

$$G_{SOGI}(z)\approx\frac{k_i\omega_o T_S z}{(z-1)} \tag{6}$$

So, the proportional resonant controller can be expressed as:

$$G_{P+SOGI}(z)=k_p+k_i\frac{\omega_o T_S z}{(z-1)}=K_p\frac{z-\alpha}{z-1} \tag{7}$$

where $K_p=k_p+k_i\omega_o T_S$ and $\alpha=\dfrac{k_p}{k_p+k_i\omega_o T_S}$.

If the proportional resonant controller is used as current controller of a grid converter connected to the grid through an L-filter, then the transfer function of the plant in the discrete time-domain is:

$$G_{RL}(z)=(1-z^{-1})Z\left[\frac{G_{RL}(s)}{s}\right]=\frac{\frac{1}{R_1}\left(1-e^{-R_1T_S/L_1}\right)}{z-e^{-R_1T_S/L_1}}=\frac{b}{z-a} \tag{8}$$

So that, the loop gain has two poles placed in: $p_1=1$ and $p_2=a$, very near to 1; and a zero located in $z=\alpha$. Depending of the controller gain, the closed loop system, $G_t(z)$, can have two conjugate poles as shown. In this case, the general form of the denominator polynomial of $G_t(z)$ is:

$$P(z)=(z-\rho e^{j\vartheta})(z-\rho e^{-j\vartheta})=z^2-(2\rho\cos\vartheta)z+\rho^2 \tag{9}$$

where $\rho=e^{-(\xi\omega_n T_S)}$ and $\vartheta=\omega_n T_S\sqrt{(1-\xi^2)}$. To verify these conditions, K_p and α have to take the next values:

$$K_p=\frac{1+a-2\rho\cos\vartheta}{b};\qquad \alpha=\frac{a-\rho^2}{bK_p} \tag{10}$$

and then $k_p=\alpha K_p$ and $k_i=\dfrac{K_p-k_p}{\omega_o T_S}$.

For a concrete application, the next values are established: $T_S=200\mu s$, $\xi=0.707$ and a settling time $t_S=2ms$. The last one takes the indicated value so that the computational delay, due to that the reference calculated in k is applied to PWM generator in $k+1$, does not affect to the system response. From settling time and damping coefficient values, ω_n can be calculated with the expression $t_s\approx\dfrac{4}{\zeta\omega_n}$ for the second order system. Using the proposed tuning method, Fig. 12 represents the response ($i_{out}(s)$), the reference ($i^*(n)$) and the error ($error(n)$) of the current control loop shown in Fig. 11. As can be verified in Fig. 12, the system responds such as has been configured, because, for example, the error takes null value in 2ms that is the settling time.

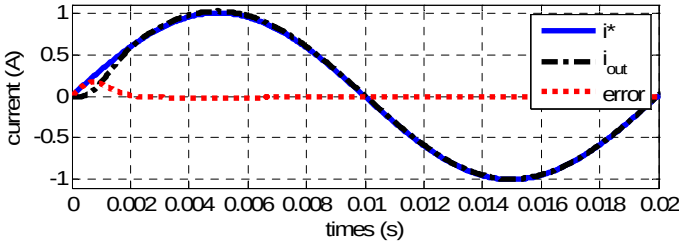

Fig. 12. Reference ($i^*(n)$), output ($i_{out}(s)$) and error ($error(n)$) currents of the control diagram of Fig. 11.

A. Saturation for the Current Controller

To guarantee a correct operation of the PR controllers it is necessary to limit the input references to achievable values for the converter. The PR controllers track perfectly the reference when the signals are sinusoidal, therefore, the reference changes should be carried out in the values close to zero for the α and β signals.

VI. Experimental Results

In this section experimental results are presented from two different points of view. Firstly, the responses of the two techniques of discretization shown in this paper are compared when the SOGI is applied in a MSOGI-QSG. Secondly, it is presented a comparison of the run times and the output signal THD of the SOGI proposed in Fig. 6 when it is implemented in a fixed-point DSP, a floating-point DSP and an FPGA.

The first block of experimental results is presented in Fig. 13, where it is shown how are detected the 5th and 7th harmonics with the two discretization methods presented in this paper. Both harmonics are balanced and the amplitude is 2% and 3% for the 5th and 7th, respectively, with respect to the fundamental amplitude. The algorithms have been programmed in fixed point in the TMS320F2812 DSP, whose internal clock period has been configured as 6.66ns; whereas the sampling time value is 200µs. The results verify the conclusions indicated in previous section. The resolution of second discretization method, especially for the application as DSOGI and MSOGI, depends on sampling period, increasing for less values of sampling time.

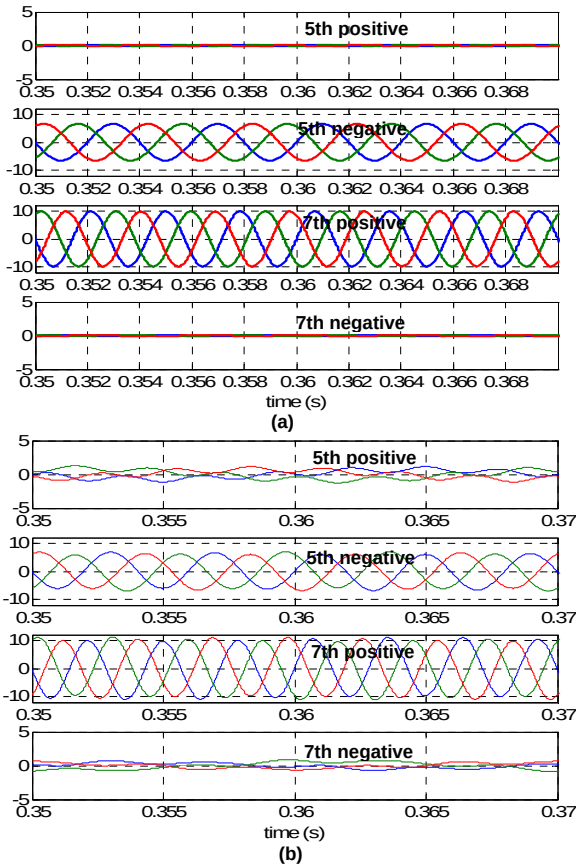

Fig. 13. Detection of 5th and 7th harmonics for the discretization methods presented in: a) Section III, and b) Section IV.

The comparative study about the run time and the output signal THD of the SOGI structure proposed in Fig. 6 is shown in Table V. The presented results are obtained for a $T_S = 200\mu s$, and to measure the THD, a step of amplitude 1 is applied to the input. TMS320F2812 is the chosen fixed-point DSP with an internal clock period of 6.66ns. The algorithms are programmed directly in Code Composer in fixed-point and floating-point, and, besides, the Matlab toolbox "Embedded Target for TI C2000 DSP" is also used to download the algorithms developed in Matlab. Regarding to the floating-point DSP, TMS320C6713 is the chosen DSP and the clock period is 4.44ns. Also, Table V presents the experimental results when the SOGI is programmed directly in Code Composer and with the Matlab toolbox "Embedded Target for the Texas Instruments TMS320C6000". Lastly, the SPARTAN3E with a clock period of 20ns is used for obtaining the experimental results indicated in Table V. The occupation for the SOGI proposed in Fig. 6 is: Number of BUFGMUXs 4%; Number of MULT18X18SIOs 5%; Number of Slices 2%; and Number of SLICEMs 0%.

VII. Conclusions

This paper has been proposed a common discretized SOGI structure for the different applications from two possible methods for discretizing. The first method is based on discretizing the continuous time-domain transfer functions whereas the second one is based on the discretization of each one of two integrators that compose the SOGI. The advantages and disadvantages of each method have been shown in the paper. Regarding to the second one, which is more novel, it has been included a possible implementation in C-language, it has been validated as current controller and as generator of quadrature-signals, and a mathematical method to tune the constants of a proportional resonant controller based on this SOGI to control of grid converter connected to the grid through and L-filter has been presented. Lastly, experimental results of the operation of the two discretization methods when the SOGI is included in a MSOGI-QSG to detect different grid frequency has been presented and the execution times of the novelty discretized structure of the SOGI using different digital structures have been shown.

TABLE V
Run Times of the SOGI proposed in Fig. 6 and THD of the obtained Output Signals.

Kind of digital processor	Run Time	THD
Fixed point DSP		
Code in fixed-point Q12 programmed in Code Composer	253ns	0.15%
Code in fixed-point Q12 generated from Matlab	16µs	0.02%
Code in floating-point programmed in Code Composer	6µs	4.1%
Code in floating-point generated from Matlab	20µs	4.1%
Floating-point DSP		
Code programmed in Code Composer	600ns	0.03%
Code generated from Matlab	8µs	4.1%
FPGA		
Code programmed in Q12	100ns	0.14%

Acknowledgment

This work has been funded by the Spanish Administration (PHB 2006-0074-PC) and Brazil Administration. Authors thank to the students of the departments involved in this international agreement for the collaboration in the implementation of the algorithms.

References

[1] P. Rodriguez, R. Teodorescu, I. Candela, A. V. Timbus, and F. Blaabjerg, "New positive-sequence voltage detector for grid synchronization of power converters under faulty grid conditions," in Proc. of PESC'06, 2006

[2] L.R.Limongi, R.Bojoi, C.Pica, F. Profumo, A.Tenconi. "Analysis and Comparison of Phase Locked Loop Techniques for Grid Utility Applications" Power Conversion Conference - Nagoya, 2007. PCC '07, pp. 674-681.

[3] Yuan, X., Merk, W., Stemmler, H., and Allmeling, J.: "Stationary frame generalized integrators for current control of active power filters with zero steady-state error for current harmonics of concern under unbalanced and distorted operating conditions", IEEE Trans. Ind. Appl., 2002, 38, pp. 523–532.

[4] D. N. Zmood, D. G. Holmes, "Stationary Frame Current Regulation of PWM Inverters with Zero Steady-State Error" IEEE Trans. on Power Electr., vol. 18, no. 3, May 2003, pp. 814 – 822.

[5] B.Burger, A.A.Engler. "Fast signal conditioning in single phase systems". Proc. of European Conference on Power Electronics and Applications, 2001.

[6] M.Cioboratu, R.Teodorescu, F.Blaabjerg. "A New Single-Phase PLL Structure Based on Second Order Generalized Integrator". 37th IEEE Power Electronics Specialists Conference, 2006, PESC 2006.