Microcontroller Implementation of a Harmonic Control Arrays System

Mehmet F. Celebi and Murat Dogruel

***Abstract*— Harmonic Control Arrays (HCA) is a technique that is based on adjusting the harmonic components of the control signal for systems with periodic references and disturbances. In this study, a microcontroller implementation of the HCA method, the algorithm, and the results has been represented. A first order system, consisting of a resistor and a capacitor has chosen as the plant, and a periodic reference signal have been applied on it.**

I. Introduction

The Harmonic Control Arrays (HCA) is a method that is recently asserted for control systems with periodic references and disturbances [1-3]. The technique is proposed and applied for inverter systems before. It provided zero steady state error for fundamental frequency, and the harmonics controlled. The principle of the method is, applying the desired control process to each harmonic component of the error signal one after another.

The discrete-time Fourier transform of a function X to a sequence $x[n]$ is defined by

$$X\left(e^{j\omega}\right)=\sum_{n=-\infty}^{\infty}x\left[n\right]e^{-j\omega n} \quad (1)$$

The cost of lesser settling times, without increasing the steady state errors, of the HCA technique is more processing speed. In addition, a single-tasking microcontroller cannot handle complex variables and the sequence is needed to be defined in rectangular form as

$$X\left(e^{j\omega}\right) = X_{re}\left(e^{j\omega}\right) + jX_{im}\left(e^{j\omega}\right) \quad (2)$$

where $X_{re}(e^{j\omega})$ and $X_{im}(e^{j\omega})$ are the real and imaginary parts of X, respectively, and are real functions of ω [5]. Applying the control method to all of the real and imaginary components of the Fourier series expansion one by one, rather than applying to the variable at the left hand side of Eq. 1 requires extra processing power for each component computed.

In this paper, a study of a microcontroller-based control with HCA technique has been represented, the algorithm has been discussed, the application has been shown and the results have been given.

M. F. Celebi is with the Mechatronics Engineering Department, Faculty of Technology, Marmara University, Kadikoy, Istanbul 34722 Turkey (corresponding author to provide phone: +90-216-336-5770 / 1674; fax: +90-216-337-8987; e-mail: fatih.celebi@marmara.edu.tr).

M. Dogruel., is with the Electrical and Electronics Engineering Department, Faculty of Engineering, Marmara University, Kadikoy, Istanbul 34722 Turkey. (e-mail: mdogruel@marmara.edu.tr).

Proportional-integral (PI) controller has been chosen as the controller type, and applied as in the dynamical phasor representation [6-9] of the error variable with different proportional and integral parameters (K_p and K_i). A sinusoidal periodic reference has been chosen, and the control signal have been applied to an L-type low-pass filter shown in Fig. 1 with a STM32F4Discovery kit featuring a 32-bit ARM® Cortex® - M4 with Floating Point Unit (FPU) core microcontroller unit (MCU) as the control unit [10].

The FPU of the MCU empowers the 168 MHz processing frequency and makes the system capable, even for more complex applications with more processing speed cost.

The control signal applied to the plant via analog-to-digital converter (ADC), and the feedback is taken from the plant using digital-to-analog converter (DAC) included with the MCU.

The results have been taken using MATLAB via a Universal Serial Bus to Transistor-Transistor Logic (USB-to-TTL) converter using serial port protocol [11].

II. Discrete Time Description of HCA Method

The HCA structure can be shown as Fig. 2 where r is the reference input, y is the system output, e is the error, and u is the control signal. Special operations of the method can be shown consisted by three blocks, consisting of the block shown with “< >” symbol is called Harmonic Disperser, the one shown with “> <” symbol is called Harmonic Assembler, and the main HCA block where the controller applied to the error signal for each harmonic. The arrows between the Harmonic Disperser and the Harmonic Assembler are corresponding for each harmonic component and the number of components are limited by the designer and processing capability of the control unit.

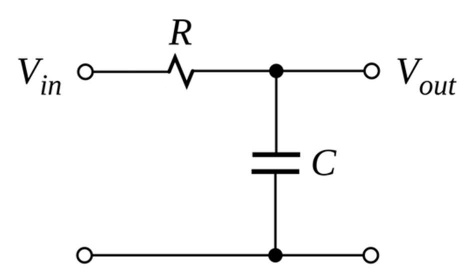

Figure 1. The plant.

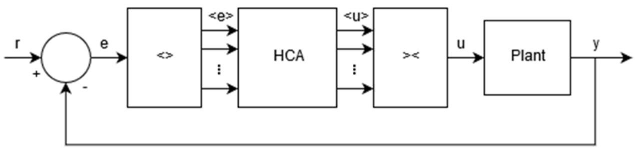

Figure 2. The HCA block diagram.

A. Harmonic Disperser

The harmonic disperser transforms the harmonic components of a signal $x(t)$ as a function of time. For any signal, any h^{th} order harmonic of a signal $x[n]$ in discrete-time domain can be obtained as

$$\langle x\rangle_h[n]=\frac{1}{N}\sum_{k=n-N+1}^{n}x[k]e^{-jh\omega kT_s}=\frac{1}{N}\sum_{k=n-N+1}^{n}x[k]e^{-j2\pi hk/N} \quad (3)$$

where h is an integer, T is the fundamental period, $\omega = 2\pi / T$ is the angular frequency, T_s is the sampling period.

$x[k] = x(kT_s)$, and $<x>_h[n]$ approximately represents $<x>_h(hT_s)$. Note that the exponential term in (3) is a periodic function in time, that is

$$e^{-j2\pi hk/N}=e^{-j2\pi h(k+N)/N} \quad (4)$$

For each $k \epsilon Z$, because of periodic nature of the Eq. 4, it is enough to calculate for only one period, for all of N samples. After these values are calculated for $k = 0, 1, \ldots, N-1$, instead of recalculating each time, these values can be stored at the memory, and be recalled whenever they are to be used. One other important processing time saving can be earned by not making extra identical calculations. Thus, Eq. 3 can alternatively be determined as

$$\langle x\rangle_h[n]=\langle x\rangle_h[n-1]+\frac{1}{N}(x[n]-x[n-N])e^{-\frac{j2\pi hn}{N}} \quad (5)$$

After the equation (5) is formed, because it depends on complex terms, and MCUs cannot process with complex variables, it is needed to transform to the rectangular form as in the Eq. 2. The exponential equation can be transformed to its sinusoidal equivalent with Euler's formula as

$$e^{jx}=\cos(x)+j\sin(x) \quad (6)$$

These separated real and complex terms can be stored as real variables in the memory. To implement the harmonic disperser of (5), it have to be transformed too, by the Euler's formula as

$$\langle x\rangle_h[n]=\langle x\rangle_h[n-1]+\frac{1}{N}(x[n]-x[n-N])\cos\left(\frac{2\pi hn}{N}\right)$$
$$-j\frac{1}{N}(x[n]-x[n-N])\sin\left(\frac{2\pi hn}{N}\right) \quad (7)$$

After this transformation, the separated real and complex parts of $<x>_h$ can be stored for the further processes.

B. The HCA Controller Block

In this block, desired controller can be applied to each harmonic component individually, with different parameters and different controllers.

In this study, a PI controller with different proportional and integral parameters has applied as shown in Fig. 3 in z-domain. The microcontroller can calculate the summation at the integral as

$$E_h[n]=E_h[n-1]+T_s<e>_h[n] \quad (8)$$

where E_h stands for the integral of $<e>_h$ in discrete frequency domain, as adding on each time the loop. The control signal after the PI block, therefore, can be written as

$$<u>_h[n]=K_p<e>_h[n]+K_iE_h[n] \quad (9)$$

C. Harmonic Assembler

The control signal must be reconstructed (i.e. assembled) before it is applied to the plant. Thinking the harmonic disperser as a kind of Fourier transform and the harmonic assembler as an inverse transform is conceivable. The control signal can be inverse transformed to discrete-time domain as

$$u[n]=\langle u\rangle_0[n]+2\,Re\sum_{h=1}^{H}\langle u\rangle_h[n]e^{\frac{j2\pi hn}{N}} \quad (10)$$

After the harmonic assembler, convenient mapping, and transformation, control signal will become suitable in continuous-time domain.

III. Microcontroller Implementation Of HCA On A First-Order System

The resistance (R), capacitance (C) and signal frequency (f) values of the plant, and the sampling period (T_S) (i.e. timer period) of the MCU is given in Table 1. K_p and K_i for each harmonic component has chosen as in Table 2.

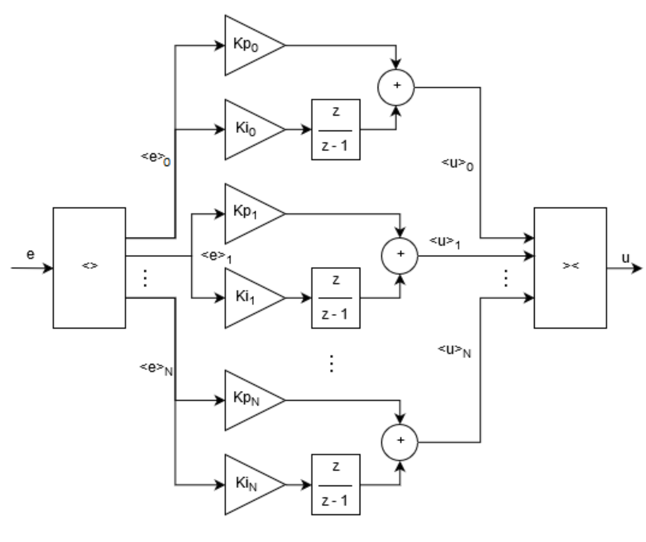

Figure 3. The controller inside the HCA.

TABLE I. SYSTEM VALUES

Variable	*Value*	*Unit*
R	2.17	*kΩ*
C	1	µ*F*
f	100	*Hz*
T_S	100	µ*s*

TABLE II. PI PARAMETERS

Harmonic number	K_p	K_i
0	1.8	100
1	2.7	135
2	3	150
3	2.7	135
4	2.7	135

The reference signal is sent as in the Fig. 4 where the x-axis is the sample number and y-axis is the corresponding integer value. It is enough to send the reference once via serial-port, and the program uses it in the algorithm repetitively.

We used a periodic reference to study with five harmonic components, so that the method can be represented as revealing as possible in this study. Choosing the harmonic number higher may improve the results especially for nonlinear loads.

A flowchart of the general algorithm used to program the microprocessor has given in Fig. 5, and the subprogram, the infinite timer loop where the control process happens used in the algorithm has given in Fig. 6.

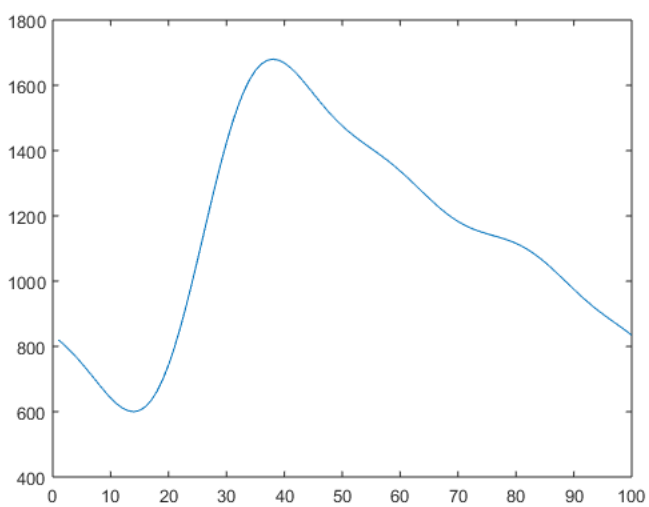

Figure 4. The reference signal

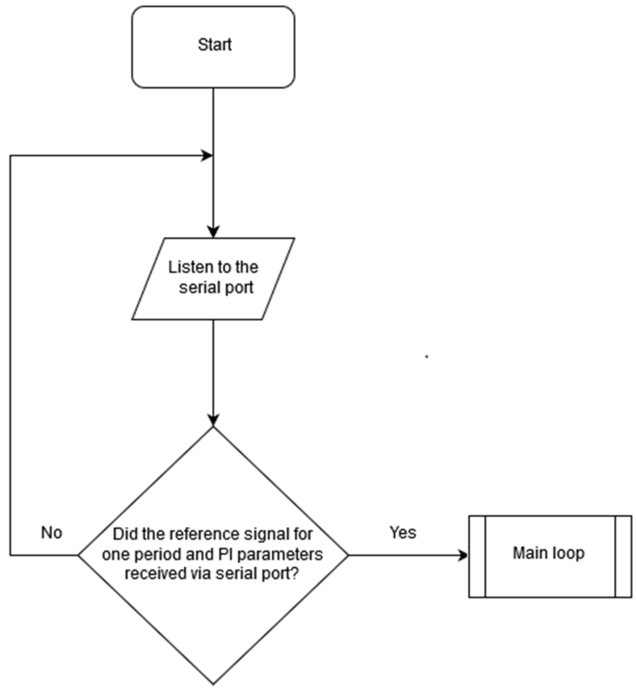

Figure 5. Flowchart of setup algorithm.

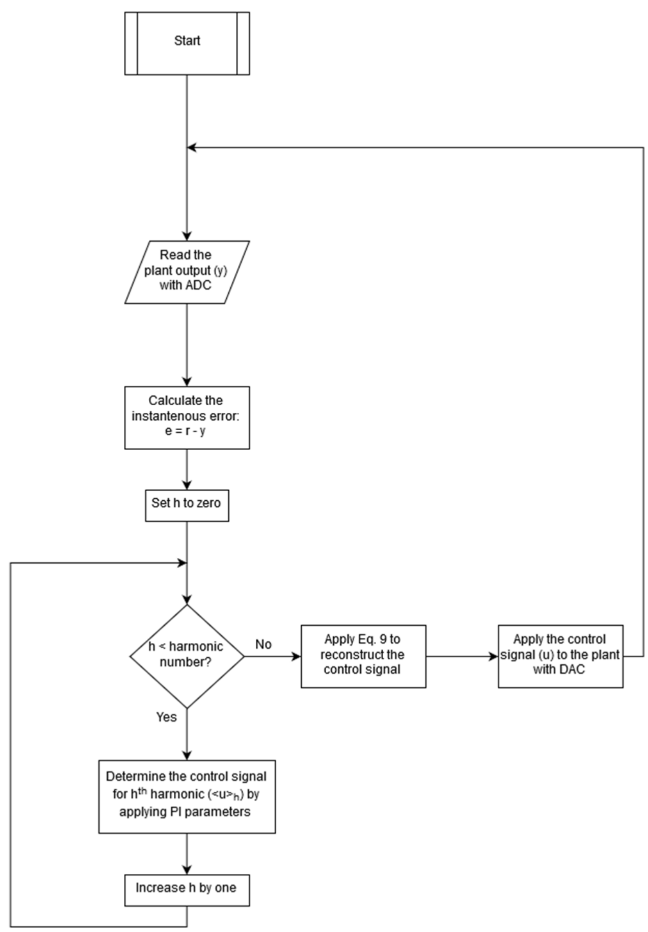

Figure 6. Flowchart of the main loop (timer interrupt).

IV. RESULTS

The reference signal (*r*), the output signal (*y*), the error signal (*e*), and the control signal (*u*) are given with Fig. 7. There, and on the reference signal given with Fig. 4, the numbers of the y-axis on the plot can be converted to real-time voltage values by

$$V_{real} = \frac{Digital\ value}{Digital\ max} V_{ref} \qquad (11)$$

where V_{real} is the real time voltage, and V_{ref} is the reference voltage of the DAC [12]. In this study, V_{ref} of the DAC is 3.3 *Volts*, and maximum digital value of it is 4095 since it supports and works with 12-bits.

The corresponding output voltage of the steady state output signal in Fig. 7 and the reference signal sent as given in Fig. 4 can be seen as equal by applying the values on the y-axis in Fig. 7 to Eq. 11.

The amplitude is (1000 / 4095) * 3.3 = 0.8 Volts, and the offset is (2000 / 4095) * 3.3 = 1.6 Volts.

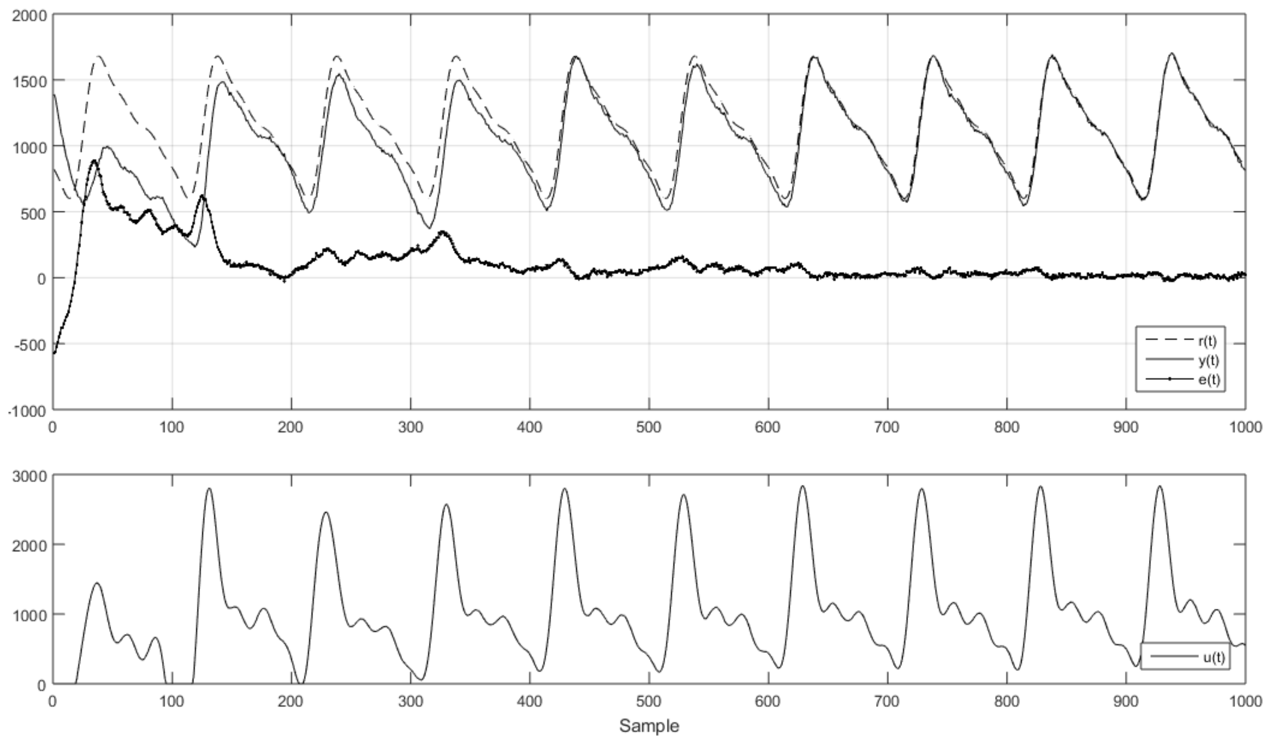

Figure 7. The reference, output, error, and control signals.

An oscilloscope screen image of the steady state output voltage of the plant has given with Fig. 8 to crosscheck the results and the image of the prepared graphical user interface (GUI) in MATLAB used to send the parameters via serial port has given in Fig. 9.

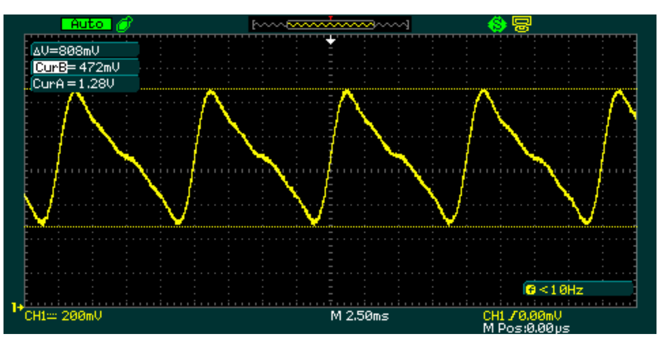

Figure 8. Screen image of the oscilloscope.

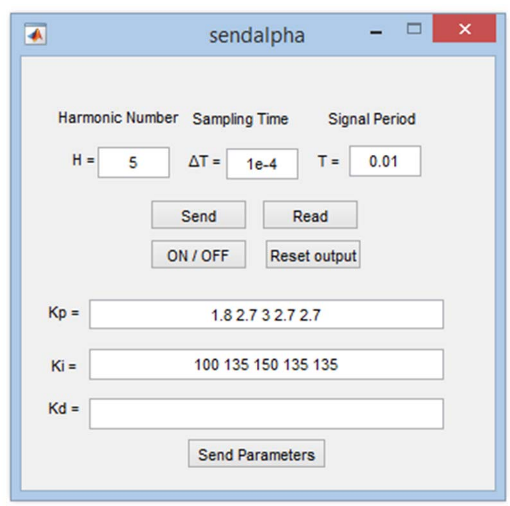

Figure 9. GUI.

In addition, sample correlative code blocks of the setup and main loop is given in Fig. 10 in C language, and a photograph of the control unit and the plant is given in Fig 11.

```c
...
/* Setup */
...
// Calculating the complex values applying Eq. 3 and Eq. 5 for Eq. 4
for(h = 0; h <= _H; h++) {
        for(k = 0; k < _N; k++) {
                exp_cos[h][k] = cos(2*pi*h*k/_N);
                exp_sin[h][k] = -sin(2*pi*h*k/_N);
        }
}
...
/* End of setup */
...
/* Main loop */
...
//calculate error
x = r[k] - y;
k++;
if (k >= _N) k = 0;
for(h = 0; h <= _H; h++) {
         //apply Eq. 7 for each harmonic component
         X_real[h] += (x - x_buff[k]) * exp_cos[h][k] / _N;
         X_imag[h] += (x - x_buff[k]) * exp_sin[h][k] / _N;
         //apply Eq. 8 for each harmonic component
         E_real[h] += T_s * X_real[h];
         E_imag[h] += T_s * X_imag[h];
         //apply Eq. 9 for each harmonic component
         U_real[h] = Kp_real[h] * X_real[h] - Kp_imag[h] * X_imag[h]
                   + Ki_real[h] * E_real[h] - Ki_imag[h] * E_imag[h];
         U_imag[h] = Kp_real[h] * X_imag[h] + Kp_imag[h] * X_real[h]
                   + Ki_real[h] * E_imag[h] + Ki_imag[h] * E_real[h];
}
x_buff[k] = x;
ubuff = 0;
//Apply Eq. 10
for(h = 1; h <= _H; h++) {
        ubuff +=  (U_real[h] * exp_cos[h][k] + U_imag[h] * exp_sin[h][k]);
}
u = U_real[0] + 2 * ubuff;
...
/* End of main loop */
...
```

Figure 10. Sample C-codes.

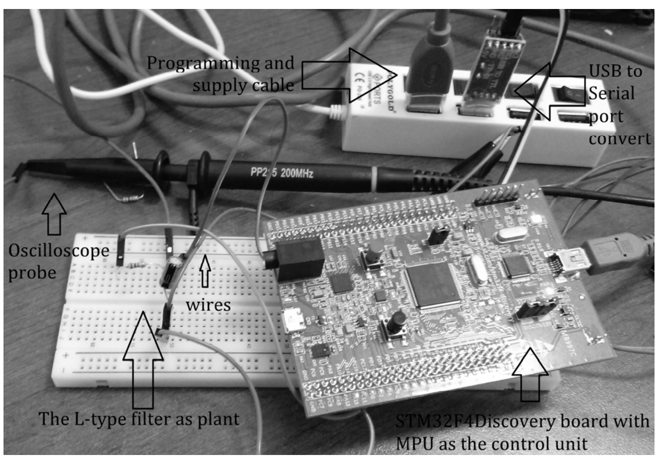

Figure 11. Photograph of the system.

V. Conclusion

The method has been implemented on a simple first order electronic system and the error has been reached to steady state for its fundamental frequency and first four harmonics for a periodic reference with four harmonics. Working with more harmonics will provide better results, especially for nonlinear systems and for this example, the harmonic ripples on the error signal could be compensated. But this will require more processing speed and power.

The PI parameters are chosen constant in this experiment, while an estimation based control can reach better results and also robustify the system.

The method is also suitable for a system with FPGA as control unit if the technique will be applied in a way that one harmonic component information will effect the other. Parallel programming with an FPGA will meet the processing speed requirement for a situation like that.

References

[1] M. Dogruel, “Harmonik kontrol dizileri,” presented at the Turkish Nat. Symp. Autom. Control, Istanbul, Turkey, Jun. 2005.

[2] M. Dogruel, “Harmonic control arrays,” presented at the Syst. Control Theory Workshop, Gebze, Turkey, Sep. 2005.

[3] M. Dogruel, H. H. Çelik, “Harmonic control arrays method with a real time application to periodic position control”, IEEE Trans. Control Syst. Technol., vol. 19, no. 3, pp. 521–530, May. 2011

[4] M. S. Karbasforooshan, M. Monfared, M. Dogruel, “Application of the harmonic control arrays technique to single-phase stand-alone inverters”, IET Power Electron., vol. 9, iss. 7, pp. 1445–1453, Jan. 2016

[5] S. K. Mitra, *Digital Signal Processing*, 2nd ed., McGraw-Hill, 2001, pp. 118.

[6] W. Levine, *The Control Handbook*, Boca Raton, FL: CRC Press, 1996, pp. 1422.

[7] S. R. Sanders, J. M. Noworolski, X. Z. Liu, and G. C. Verghese, “Generalized averaging method for power conversion circuits,” IEEE Trans. Power Electron., vol. 6, no. 2, pp. 251–259, Apr. 1991.

[8] P. Mattavelli, G. C. Verghese, and A. M. Stankovic, “Phasor Dynamics of thyristor-controlled series capacitor systems,” *IEEE Trans. Power Syst.*, vol. 12, no. 3, pp. 1259–1267, Aug. 1997.

[9] A. M. Stankovic, S. R. Sanders, and T. Aydin, “Dynamic phasors in modeling and analysis of unbalanced polyphase AC machines,” *IEEE Trans. Energy Conv.*, vol. 17, no. 1, pp. 107–113, Mar. 2002.

[10] STMicroelectronics, “UM1472 User manual - Discovery kit with STM32F407VG MCU”, www.st.com, DocID022256 Rev 5, 2016.

[11] Future Tech. Dev. Int. Ltd., “Application Note AN_243 FT312D USB Host to UART Cable Application”, Doc. Ref. No. FT_000839, May. 2013.

[12] D. A. Neamen, *Microelectronics Circuit Analysis and Design,* 4th ed., New York, McGraw-Hill, 2010, pp. 1227.