



A Stand-alone Application Software of Graph Plot for Digital Signal Processing

Arjuni Budi Pantjawati^{1*}, Budi Mulyanti¹, Agus Heri Setya Budi¹,
Enjang Ahmad Juanda¹ and Rizki Nuzulfikri²

¹Department of Electrical Engineering Education, Universitas Pendidikan Indonesia,
Jalan Dr. Setiabudhi No 229, Bandung 40154, Indonesia

²Faculty of Mathematics and Natural Science, Universitas Islam Bandung, Jalan Tamansari No. 1,
Bandung 40161, Indonesia

ABSTRACT

This paper discusses the application of software design to display graph plots of Digital Signal Processing (DSP) results. This design uses MATLAB programming language by utilising the MATLAB GUI. This application was built using MATLAB R2012a for the 32-bit version and MATLAB R2014a for the 64-bit version. The application can plot graphs of digital signal processing results such as sampling, convolution, deconvolution, Discrete Time Fourier Transform, Z Transform, FIR Digital Filters, IIR Digital Filters and many others. The software can be installed on all windows operating system and function as a stand-alone program with all supporting files integrated in one package installer and therefore, easily distributed. By using this application, we can resolve many cases on DSP simply and interactively.

Keywords: Application software, Digital Signal Processing, Graph Plot, MATLAB, Stand-alone

ARTICLE INFO

Article history:

Received: 12 January 2017

Accepted: 02 October 2017

E-mail addresses:

arjunib@upi.edu (Arjuni Budi Pantjawati),
bmulyanti@upi.edu (Budi Mulyanti),
agusheri@yahoo.com (Agus Heri Setya Budi),
juanda@upi.edu (Enjang Ahmad Juanda),
gururizki@gmail.com (Rizki Nuzulfikri)

*Corresponding Author

INTRODUCTION

Digital technology is used in almost all areas of life that utilise electronic devices and computers. Digital Signal Processing (DSP) in particular, is the most important part of the technology determines the quality of output produced (Tan & Heinzelman, 2003; Abdel-Qader, Bazuin, Mousavinezhad, & Patrick, 2003)

DSP involves of computational mathematics (German-Salloa, 2014; Ownby, & Mahmoud, 2003) that is often difficult to understand without the aid of a visual

presentation (McClellan, Schafer, & Yoder, 1998). For example, in the sampling process, when a sinusoidal signal with the following equation

$$x(t) = A \cos(\Omega_0 t + \phi) \quad (1)$$

is sampled using sampling rate f_T , then the sampled signal

$$x(nT) = A \cos(\Omega_0 nT + \phi) \quad (2)$$

is the value of signal $x(t)$ at the time of $t = nT$ or $(0T)$, $(1T)$, $(2T)$ and so forth with the value of T is (Mitra, 2011)

$$T = \frac{1}{f_T} \quad (3)$$

Without the aid of a visual presentation, the previous sampling process is quite difficult to be understood for beginners. The plot process of signal sampling requires considerable time, and will be even longer for multiple values of T . The visual presentation is also necessary in other cases (Jamehbozorg & Radman, 2013; Ambikairajah & Epps, 2011). Utilising a computer program for visual presentation of the DSP results can be a solution in which the plot image can be displayed in just 1 to 10 seconds.

In this study, MATLAB programming platform was used by utilising the GUI facility to design the program in the form of windows so that users can easily run the application even if they don't have much understanding programming language (Hailan & Wei, 2012). The type of data is also not important. The user only needs to place a value on the variables to be used (Li, Harari, Wong, & Kapila, 2004).

Using MATLAB GUI to develop MASTERS (Multimedia systems in Telecommunication, Medical and Remote Sensing applications) had been done previously in the form of Virtual Laboratory (Alexiadis & Mitianoudis, 2013). Meanwhile, Kulmer, Wurzer, and Geiger, (2016) used MATLAB GUI to develop the MRLT (Magnitude Response Learning Tool) as a learning game to estimate magnitude response of LTI (Linear Time Invariant) systems based on pole/zero chart. The difference between those studies and the current one is the MASTERS Virtual Laboratory was designed for solving and explaining more complex concepts of DSP such as audio, image, and video encoding and compression, video and audio streaming and medical image processing (Alexiadis & Mitianoudis, 2013), while this graph plot application software was designed for explaining basic concepts of DSP. The MRLT was designed with a limited scope that is a magnitude response of LTI system (Kulmer et al., 2016), while this application software covers a wider scope to display graph plots of DSP results. The study covers basic DSP problems such as signal forms, signal operations, system response and system operations. This application software accommodates 32 bit as well as 64-bit computer versions using MATLAB R2012a and MATLAB R2014a respectively. In order to facilitate the distribution, the application was designed to be installed on all windows operating system and stand-alone with all the supporting files integrated in one package of installer (Andreatos & Zagorianos, 2009).

Design of Application Software

The software is designed to display the plot of digital signal processing's results. The input of this application is parameters value required in signal operation. For example, a plot graph of convolution result between discrete signal $x(n) = \delta(n) + \delta(n - 1)$ and system impulse response $h(n) = \delta(n) + \delta(n - 2) + \delta(n - 3)$ can be found by inserting the discrete signal and impulse response's weights to application. As many as 45 types of signals can be displayed using this application, namely several types of analog signals, discrete signals, signal sampled result, convolution, deconvolution, discrete time Fourier transform, transformation Z, until the digital filter. The number of displayed signals can be increased by updating the software program.

The users can easily change the value of some parameters and see the changes on the graph plot. The mathematical equations used can also be displayed in order to provide complete information about the related signal processing.

System Flow Chart

Figure 1 is a flow chart that shows how the application is developed. The first display is a welcome menu window that has a selection of ENTRY and EXIT buttons. The ENTER button selection will take the user to the menu options consisting of main options and additional menus. The main menu selection is a graph plot of some signal processing operations, while the additional menu is in the form of magnifying glass and calculator functions. The graph plot will then appear after the user enters the value of parameters in the provided space.

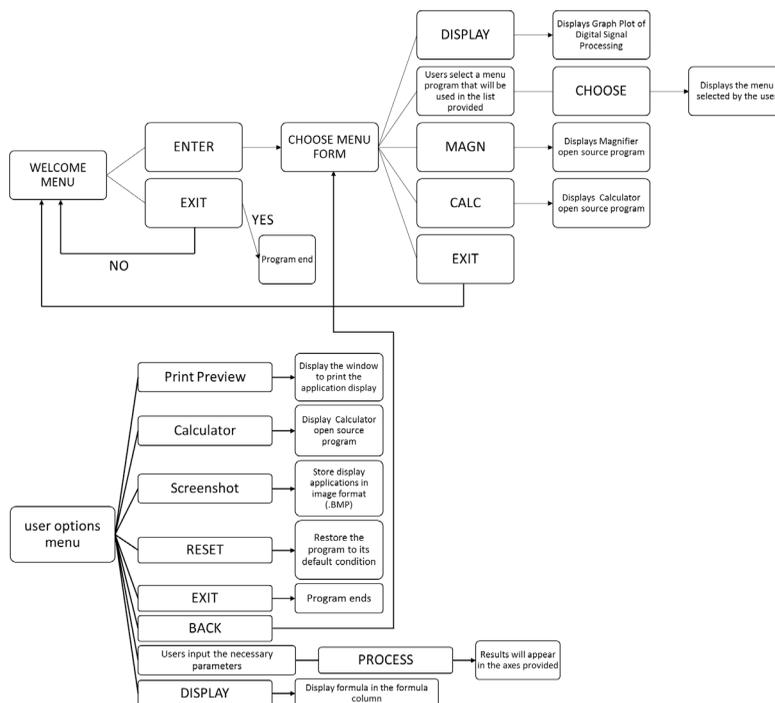


Figure 1. System Flow Chart

Software Programming

The software is combined into a single entity through the facility of Deployment Tool available on MATLAB (Oyetunji & Ale, 2013). This application is designed to accommodate the 32 bit as well as 64-bit versions of windows operating system. These versions were built using MATLAB R2012a and MATLAB R2014a respectively. In order to facilitate the distribution, the applications were designed to be installed on all windows operating system and stand-alone with all the supporting files integrated in one package of installer so it can be run on any computer device without installing MATLAB program (Kumar, Raja, & Selvan, 2012).

Furthermore, the program was tested on three computers with different specification in order to determine their operating performance.

Graph Plot of Digital Signal Processing

This application provides the appearance of graph plots for 45 signal types of Digital Signal Processing. One graph plot that can be displayed is the signal from the sampling process. The parameters entered are analog signal frequency to be sampled and the sampling period. The sampling process has to fulfil the Nyquist condition, namely signal with frequency f_0 has to be sampled with a sampling rate $f_T \geq 2 f_0$ so that signal source can be generated back from the signal sampling results (Mitra, 2011; Mathumisanon & Chayratsami, 2013). By using this application, graph plot of signal of sampling results, whether it fulfils the Nyquist condition or not, can be displayed by changing the parameters.

RESULTS AND DISCUSSIONS

Tests were conducted to determine the general specification of the application software developed and the displays graph plot generated.

The System Hardware and Software Specification

System specifications were obtained from the testing of software installation on three different specification of computers. Table 1 shows the system hardware and software specifications of the application software. From Table 1, it can be seen that the software can be installed on all windows operating system, accommodate the 32 bit as well as 64-bit versions, and can use any Intel processor as well as AMD x86 which supports instruction set of SSE2. RAM minimum specification is 512 MB, and Harddisc minimum capacity is 2 GB. Software resolution is fixed size of 1344 x 634 pixels.

Table 1
The system hardware and software specification

Operating System	Windows XP Service Pack 3, Windows Server 2003 R2 with Service Pack 2, Windows Vista Service Pack 1 or 2, Windows Server 2008 Service Pack 2 or R2, Windows 7 and Windows 8.1
Platform	32 bit or 64 bit
Processor	Any processor Intel as well as AMD x86 which supports instruction set of SSE2
RAM	Minimum of 512 MB (at least 1024 MB Recommended)
Hardisc	2 GB (minimum for installation storage)
Software resolution	1344 x 634

The Display of Graph Plot

In order to clarify the application of graph plot, the following two problem-solving are described.

The test of graph plot display was conducted by solving the problems of the Digital Signal Processing. The result showed that all the buttons worked and all of the graph plot can be displayed on the main axes. To view the graph plot application in problem solving, the following illustration of two problems and solutions as examples with the aid of graph plot.

- Sampling Process: A sinusoidal signal with 5 Hz of frequency and amplitude of 1, yields the following equation:

$$x(t) = \sin(10\pi t) \quad (3)$$

In order to meet Niquist condition, the signal must be sampled with a sampling rate ≥ 10 Hz or sampling period ≤ 0.1 s. In the first case, a sampling rate of 50 Hz or sampling period of 0.02 s was used. In this case, Nyquist condition was fulfilled, a plot of signal of sampling result is shown in Figure 2. In the second case, the parameters of the sampling rate were converted to 8 Hz or sampling period of 0.125 seconds, which does not meet Nyquist condition, and the plot of signal of sampling result can be seen in Figure 3.

Both plots of the sampling result signal show the differences between sampling process that meet the Nyquist condition and vice versa. For the adequate sampling process, signal source can be reproduced from the signal of sampling results (Ludeman, 1987; Oppenheim & Schaffer, 2013).

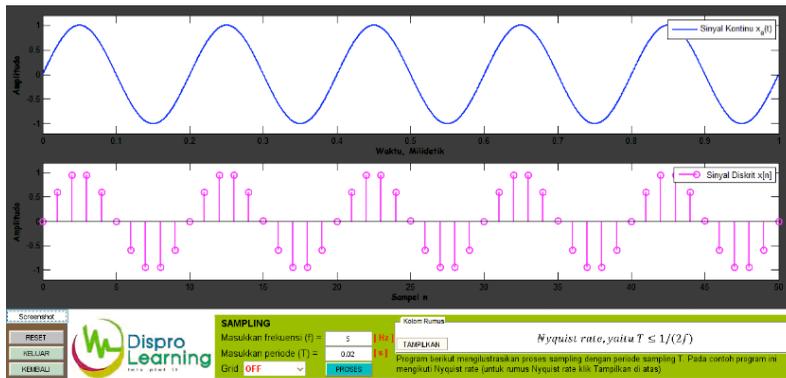


Figure 2. Plot of the sampling result signal at 5 Hz of signal frequency and 50 Hz of sampling rate

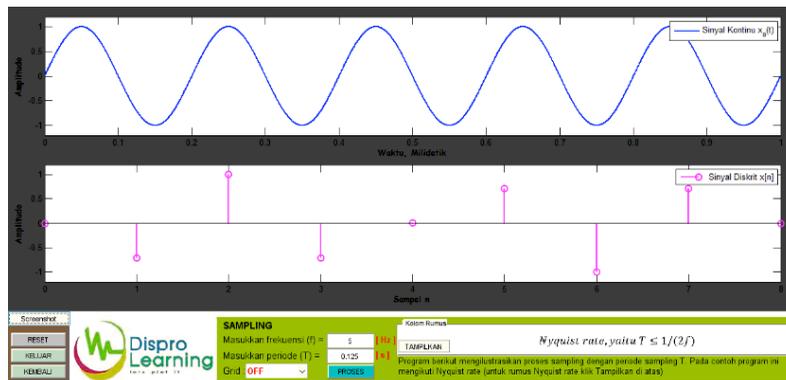


Figure 3. Plot of the sampling result signal at 5 Hz of signal frequency and 8 Hz of sampling rate

- Butterworth Lowpass Filter Order and Frequency Response using BZT Method: BZT method (Bilinear Z Transform) is the IIR (digital filter that has Infinite Impulse Response) digital filter design technique using the transformation of s-domain to z-domain (Theede, 2005)

In this mapping, all $j\Omega$ axis in the s-plane were mapped into the unit circle in the z-plane; all points of the Left Half Plane of s were mapped into a unit circle in the z-field, while all points in Right Half Plane were mapped outside the unit circle (Proakis & Manolakis, 2007). This mapping was carried out based on the following equations:

$$s = \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \tag{7}$$

$$z = re^{j\omega} \tag{8}$$

$$s = \sigma + j\Omega \tag{9}$$

where:

ω : digital frequency

Ω : analog frequency

$$\Omega = \frac{2}{T} \tan \frac{\omega T}{2} \tag{10}$$

Filter order obtained from equation

$$N = \left\lceil \frac{\log_{10} \left(\left(10^{\frac{K_1}{10}} - 1 \right) / \left(10^{\frac{K_2}{10}} - 1 \right) \right)}{2 \log_{10} \left(\frac{\Omega_1}{\Omega_2} \right)} \right\rceil \tag{11}$$

where:

Ω_1 : *cut-off* analog frequency

K_1 : 3 dB of damping

Ω_2 : *stop-band analog frequency*

K_2 : damping at *stopband* frequency

For example, in the case of designing of low pass digital filter with spec: 2 kHz of 3 dB cut-off frequency, 30dB of minimum damping occurs at 4,25 kHz of frequency and 10 kHz of sampling rate. Using eq. (7) to (11) we obtained $\Omega_1 = 14.53 \times 10^3$ rad/s, $\Omega_2 = 83.31 \times 10^3$ rad/s and filter order = 2

Transfer function of low pass filter Butterworth order 2 (Thede, L.D., 2005)

$$H(s) = \frac{\Omega_c^2}{s^2 + s\sqrt{2}\Omega_c + \Omega_c^2} \tag{12}$$

By inserting the value of Ω_1 to Ω_c and using eq. (7), we have

$$H(z) = \frac{0.20657(1 + 2z^{-1} + z^{-2})}{1 - 0.36953z^{-1} + 0.19582z^{-2}}$$

By inserting specification parameter values required for the application software of graph plot, the coefficient of $H(z)$, filter order and the plot of response filter signal will be obtained quickly. The result of response signal plot is shown in Figure 4.

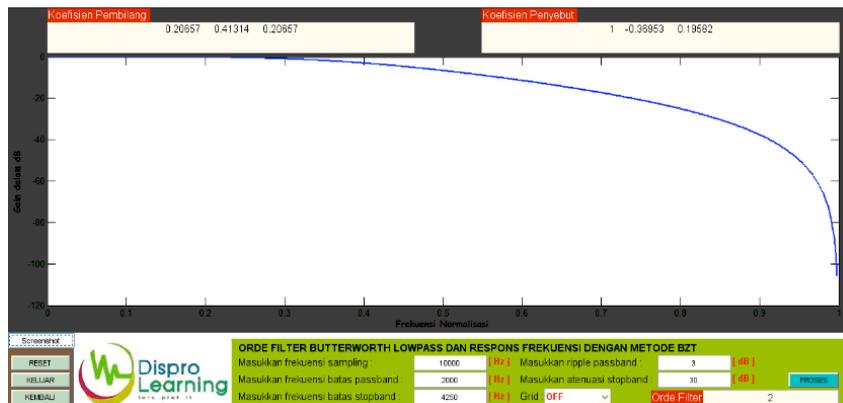


Figure 4. Plot of response signal of low pass filter using BZT method

CONCLUSION

This paper describes the results of design software to plot a graph of Digital Signal Processing using MATLAB programming language. The application can be run on any Windows operating system platform with computer specifications of low end to high end. The application was successful using MATLAB R2012a for the 32-bit version and MATLAB R2014a for the 64-bit version, as stand-alone software with all the supporting files integrated into one package installer. From the test results, the application software developed can properly and quickly solve the problem of Digital Signal Processing such as sampling process and IIR digital Filter design as described above in The Display of Graph Plot Section.

ACKNOWLEDGMENT

This research was financially supported by Dana Non PNPB Universitas Pendidikan Indonesia (UPI) of 2016 Fiscal Year through SK Rektor No. 4573/UN40/PP/2016, 31 Mei 2016. The authors are grateful to the Telecommunication Laboratory of Department of Electrical Engineering Education -UPI.

REFERENCES

- Abdel-Qader, I. M., Bazuin, B. J., Mousavinezhad, H. S., & Patrick, J. K. (2003). Real-time digital signal processing in the undergraduate curriculum. *IEEE Transactions on Education*, 46(1), 95-101.
- Alexiadis, D. S., & Mitianoudis, N. (2013). Masters: A virtual lab on multimedia systems for telecommunications, medical, and remote sensing applications. *IEEE Transactions on Education*, 56(2), 227-234.
- Ambikairajah, E., & Epps, J. (2011, January). Project-based learning in digital signal processing: Development and experiences. In *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011 IEEE* (pp. 506-511). IEEE.

- Andreatos, A. S., & Zagorianos, A. (2009, July). Matlab GUI application for teaching control systems. In *Proceedings of the 6th WSEAS International Conference on ENGINEERING EDUCATION (Vol. 208)*. (pp. 208-211).
- German-Salloa, Z. (2014). Signal processing using FPGA structures. *Procedia Technology*, 12, 112-118.
- Hailan, G., & Wei, Z. (2012, January). An experimental study of college teaching based on MATLAB software. In *2012 Fifth International Conference on Intelligent Computation Technology and Automation (ICICTA)*, (pp. 69-71). IEEE.
- Jamehbozorg, A., & Radman, G. (2013, April). A MATLAB GUI package for studying small signal characteristics of power systems with wind and energy storage units as an education tool. In *2013 Proceedings of IEEE Southeastcon*, (pp. 1-7). IEEE.
- Kulmer, F., Wurzer, C. G., & Geiger, B. C. (2016). The magnitude response learning tool for DSP education: A case study. *IEEE Transactions on Education*, 59(4), 282-289.
- Kumar, S. R., Raja, P., & Selvan, M. P. (2012, December). Virtual laboratory environment using MATLAB-GUI for teaching of induction generators. In *2012 Annual IEEE India Conference (INDICON)* (pp. 676-681). IEEE.
- Li, Y. F., Harari, S., Wong, H., & Kapila, V. (2004, June). MATLAB-based graphical user interface development for basic stamp 2 microcontroller projects. In *Proceedings of the 2004 American Control Conference, 2004*. 4, 3233-3238. IEEE.
- Ludeman, L. C. (1987). *Fundamentals of digital signal processing*. Singapore: John Wiley & Sons, Inc.
- Mathumisanon, T., & Chayratsami, P. (2013, August). MATLAB GUI for digital communication system with tone jamming. In *Teaching, Assessment and Learning for Engineering (TALE), 2013 IEEE International Conference on* (pp. 589-592). IEEE.
- McClellan, J. H., Schafer, R. W., & Yoder, M. A. (1998). A changing role for DSP education. *IEEE Signal Processing Magazine*, 15(3), 16.
- Mitra, S. (2011). *Digital Signal Processing*, New York: McGraw-Hill.
- Oppenheim, A. V., & Schafer, R. W. (2013). *Discrete-time signal processing: Pearson new International Edition*. Pearson Higher Ed.
- Ownby, M., & Mahmoud, W. H. (2003, March). A design methodology for implementing DSP with Xilinx/sup/spl reg//System Generator for Matlab/sup/spl reg. In *System Theory, 2003. Proceedings of the 35th Southeastern Symposium on* (pp. 404-408). IEEE.
- Oyetunji, S. A., & Ale, D. T. (2013). Development of MATLAB-based digital signal processing teaching module with graphical user interface environment for Nigerian university. *International Journal of Engineering and Technology*, 2(3), 230.
- Proakis, J. G., & Manolakis, D. G. (2007) *Digital signal processing*. New Jersey: Prentice Hall.
- Tan, E. J., & Heinzelman, W. B. (2003). DSP architectures: Past, present and futures. *ACM SIGARCH Computer Architecture News*, 31(3), 6-19.
- Thede, L. D. (2005). *Practical analog and digital filter design*. New Jersey: Artech House.

