

Application of LabVIEW in Digital Signal Processing Bilingual Teaching

MA Lun^a, LUO Xianglong^b, MING Yang^c and LIU Ruochen^d

School of Information Engineering, Chang'an University, Xi'an, Shaanxi, China, 710064

^alunma@126.com, ^bxlluo@chd.edu.cn, ^cyangming@chd.edu.cn, ^drc Liu@chd.edu.cn

Keywords: Bilingual teaching; Digital signal processing (DSP); LabVIEW.

Abstract. It has been 5 years that our department set DSP as a bilingual model course. However, both our teaching team and the students majoring the course encountered many difficulties during this period. To motivate students to overcome language and obscure concept problem, we adopted a graphical programming approach instead of the traditional and commonly used text-based programming approach in our DSP teaching process, and achieved good feedback from students. In this paper, some considerations and strategies for the combining LabVIEW with bilingual teaching are proposed. Moreover, two interesting combination examples are presented to make description understandable.

Introduction

With the rapid development of digital signal processor, more and more digital signal processing technologies and methods can be applied to practice based on DSP devices. Thus DSP plays an ever-increasing role in electronic systems. Given this, nearly all electrical and communication engineering curricula include DSP courses [1]. DSP courses which are usually set for junior or sophomore students in most of universities aims to give them a general understanding of the manipulation of digital signals in order to extract useful information from them.

On the other hand, bilingual education [2], as an advanced teaching method, which means employing two different languages during teaching process, has been adopted in more and more universities in China. And most of the bilingual courses use English as the second language of communication besides Chinese due to its universal acknowledgement and wide adoption in academic material. For most of the majors with regard to information engineering, DSP is set for the follow-up course of "Signals and Systems". Students already have a general understanding of digital signals and systems and have been laid a certain foundation of English when taking DSP courses. In view of this, many universities set DSP as bilingual model course during education reform.

However, opportunities and challenges often exist side by side. According to 5-year teaching experience of DSP bilingual courses, both our teaching team and the students majoring these courses encountered many difficulties. Although many of them have passed College English Test-4 (CET-4) even CET-6, they have difficulties in reading English textbook. They said they knew every word, but they could not understand the sentences. Therefore those students spent so much of their studying time on English itself that their learning enthusiasm greatly declined.

To improve students' studying motivation for DSP, many universities including us try to introduce computer simulation into teaching process. Simulation results can give them a live straight view of each concept, theorem or algorithm. To this end, a number of textbooks, such as [3-5], have been written to provide better understanding with the help of computer simulation. The programming language used in these textbooks consists of C, MATLAB, or Assembly. Our teaching team assigns "Digital Signal Processing A Computer-Based Approach" written by Sanjit K. Mitra as the textbook. There are so many MATLAB simulations with source codes in this book. New problem arises when students try to combine complicated theory to practice with the help of MATLAB codes. Besides spending so much time on language understanding, they have to spend additional plenty of time to understand MATLAB codes after class due to the fact that MATLAB is still text-based programming.

Based on the limited teaching hours and the difficulties above, an easier-to-use and easier-to-understand simulation tool needs to be introduced to relax students from understanding lengthy codes thus improving their studying motivation. LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) is a graphical programming environment developed by National Instruments (NI), which allows high-level or system-level designs. It uses a graphical programming language to create so-called Virtual Instruments (VI) blocks in an intuitive flowchart-like manner. A design is achieved by integrating different components or subsystems within a graphical framework. Due to its usability and visualization, LabVIEW receives better comments from our students compared to MATLAB and C. In the next section, we will give LabVIEW a brief introduction.

Brief Introduction to LabVIEW

As is described above, LabVIEW constitutes a graphical programming environment that allows students to design and analyze a DSP system in a shorter time as compared to text-based programming environments such as MATLAB and C. LabVIEW graphical programs are called virtual instruments (VIs). VIs run based on the concept of data flow programming. This means that execution of a block or a graphical component is dependent on the flow of data, or more specifically a block executes when data is made available at all of its inputs. Output data of the block are then sent to all other connected blocks. Data flow programming allows multiple operations to be performed in parallel since its execution is determined by the flow of data and not by sequential lines of code.

A VI consists of two major components; a front panel (FP) and a block diagram (BD). An FP provides the user-interface of a program, while a BD incorporates its graphical code. Take, for example, an adder, which realized in LabVIEW, is shown in Fig.1. The right window in Fig. 1 is the BD, which contains three terminal icons represent output $x+y$ and inputs x and y , a node “+” performs adding operation and three wires connecting inputs, outputs and nodes. It can be seen that terminal icons are interfaces through which data are exchanged between FP, the left window in Fig. 1, and BD. Whenever we input x and y in FP and press running button, adding operation will be run in the background and output $x+y$ will be also visually displayed in FP.

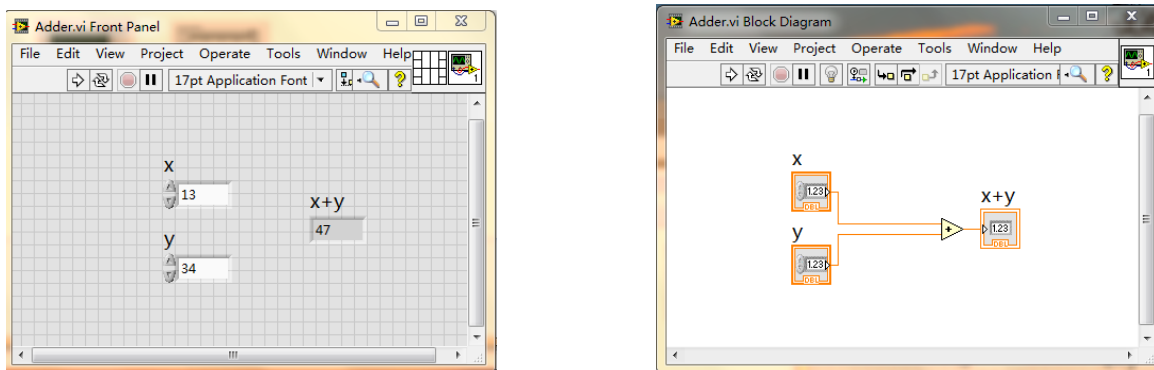


Fig.1 LabVIEW windows: FP and BD

Examples of LabVIEW aided DSP bilingual teaching

Picket fence effect

During our 5-year teaching process, we found that it is difficult for students especially those taking bilingual course to exactly tell the difference and relationship between CTFT (Continuous Time Fourier Transform), DTFT (Discrete Time Fourier Transform) and DFT (Discrete Fourier Transform). Picket fence effect is exactly a good example to understand frequency-domain sampling. In view of

this, we designed an experiment of picket fence effect by using LabVIEW as shown in Fig.2. In the experiment, it is easily to observe that the height of spectral peak corresponding to the sinusoidal signal with frequency 200.7Hz located at the middle of two discrete sampled frequencies is almost half of that corresponding to the sinusoidal signal with frequency 200Hz exactly located at one sampled frequency. With this experiment, students get a better understanding of frequency-domain sampling and frequency resolution.

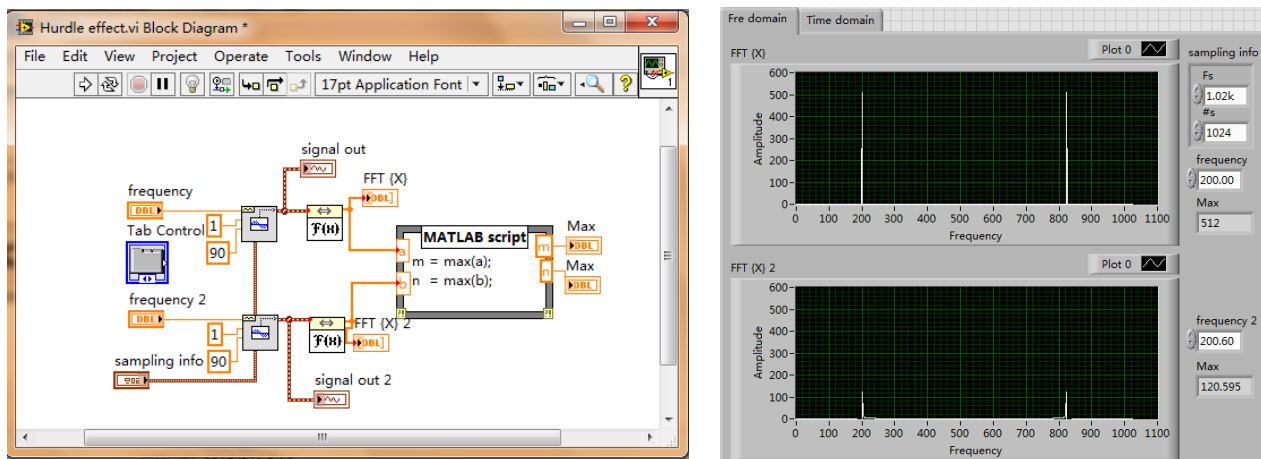


Fig. 2 Experiment of picket fence effect

Sound signal real-time collection

Analog signal are everywhere in our daily life. There are many reasons why one might wish to process an analog signal in a digital fashion by converting it into a digital signal. The main reason is that digital processing allows programmability. However, students may not realize the great advantages of digitalization when they are new to DSP. To motivate them from visualization, we implement “Acquire.Sound.VI” in LabVIEW in the first class. Through simple design, sound signal can be visually collected by microphone and sound card of PC, and then dynamically shown on the screen as shown in Fig.3. Moreover, students can easily copy the whole experiment all by themselves after class as long as teachers give them a simple introduction to LabVIEW. According to 2-year observation by our teachers, this simple experiment in the first class greatly motivates students to studying DSP.

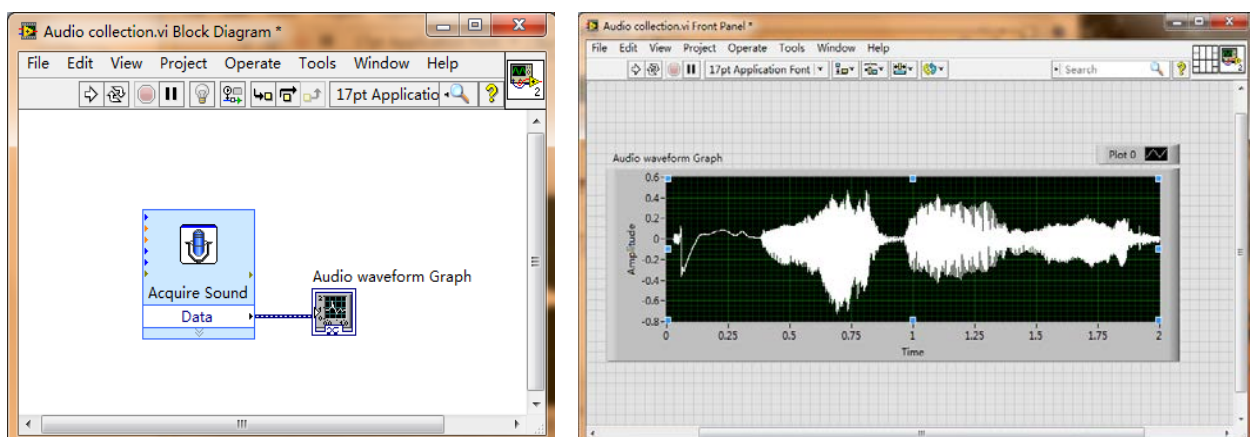


Fig. 3 Experiment of sound signal real-time collection

Strategy of combining LabVIEW with DSP bilingual teaching

According to the description above, it can be seen that graphical programming environment LabVIEW can not only motivate students but also make the obscure and theoretical DSP definitions and algorithms be visible and easy to understand. In the 5-year bilingual DSP teaching process, our teaching team has been exploring the way to more effectively combine LabVIEW to the course. In this section, some preliminary considerations and strategies for the combination are proposed.

Firstly, the time arrangements of DSP bilingual course are as follows: 46 class hours for lectures, 40 class hours for curriculum design and 8 class hours for experimentation. During lecture hours, only 40~50% of the time is used in the explanation of the concepts, 20~30% of the time is used for the solution of the examples and problems. The last 30~40% of the time is used for LabVIEW simulation demonstration with regard to the given concepts. As to curriculum design, students are required to design and implement a practical DSP systems using LabVIEW. In the past 3~4 semesters, students have designed many interesting systems with the learnt knowledge, such as voice changer, speech recognition, Function signal generator and so on. Besides, in the experiment hours, students are to use real electronic instruments, like signal source, Oscilloscope and DSP experiment board. During the experiment, they need to implement the specified experiment content in LabVIEW, and then realize that content by real electronic instruments. Mutual verification and comparison can give students a better understanding of the whole experiment and the related basic concepts.

Summary

This paper presents LabVIEW as a Graphical Development System environment and explores how to combine it to DSP bilingual teaching. According to 5-year experience of bilingual DSP teaching, we found that LabVIEW can not only motivate students but also make the obscure and theoretical DSP definitions and algorithms be visible and easy to understand. Two combination examples are given and some preliminary considerations and strategies are also discussed.

Acknowledgements

This work was financially supported by the Undergraduate teaching project of Chang'an University and Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2015JM6278).

References

- [1] Qian Xiong and Yangjun Pei, in: *International Conference on Control, Automation and Systems Engineering (CASE), 2011*, edited by IEEE eXpress Conference Publishing, Singapore, (2011).
- [2] Alan V. Oppenheim and Ronald W. Schaffer: *Discrete-Time Signal Processing (3rd Edition)* (Prentice Hall, USA 2009).
- [3] Sanjit K. Mitra: *Digital Signal Processing A Computer-Based Approach (4th Edition)* (McGraw-Hill Education, USA 2010)
- [4] John G. Proakis: *Digital Signal Processing (4th Edition)* (Prentice Hall, USA 2006).
- [5] Richard G. Lyons: *Understanding Digital Signal Processing (3rd Edition) (3rd Edition)* (Prentice Hall, USA 2010).