Setup for Adaptive DSP Algorithms Teaching

Zilvinas Nakutis
Kaunas university of technology
Department of electronic and measurement systems
Studentu 50-443, Kaunas LT-51368, Lithuania
nakutis@ktu.lt

Abstract. The paper presents hardware and software setup for the adaptive algorithms teaching in a laboratory. Two popular DSP Starter Kits DSK6713 and widely used modeling package Matlab/Simulink are utilized to compose the working environment. Adaptive digital filters application for unknown system identification and channel equalization are considered. One DSK is used to model an unknown system or a communication channel, while the second DSK is used to implement an adaptive filter. Computer equipped with a sound card is dedicated for DSK processor programs control, excitation signal generation and response signal recording and visualization. Simulink models for all above tasks are discussed and influence of their implementation on some algorithm parameters is explained. An easy student task cases generation together with exercises flow is suggested.

Keywords. Adaptive algorithms, teaching, digital signal processing.

1. Introduction

Adaptive signal processing algorithms are widely used in communication systems for channel equalization, echo cancellation, etc. The theory of adaptive filters is well developed [3,5]. Channel equalization and echo cancellation are mature technologies that usually are taught in the classical courses on digital signal processing, telecommunication systems and similar. For decades adaptive filters (AF) implementation was one of the key applications targeted by digital signal processors (DSP). Implementation of an AF on a DSP requires significant skills in DSP programming. It must be noticed that the background in the optimization theory, real time DSP programming in assembler or C language and DSP hardware design are needed to build and analyze an AF prototype. This makes it virtually impossible to cover all the fields in one coarse. However, only theoretical analysis of AF and avoidance to run and practically test AF applications hardly contribute to the successful student knowledge and skills development [6]. Fortunately, in recent years tools from DSP manufacturers such as Starter Kits and visual application development environments complemented with numerous DSP libraries (Matlab/Simulink, LabView) abstract us from any hard bolts and nuts of the hardware and software. This opens good opportunities to setup test environments for the AF applications teaching and laboratory exercises.

Numerous teaching applications using Texas Instruments C6000 family DSP Starter Kit boards are reported in resent years [4,7-9]. Their authors present and discuss setups for FIR filters and Hilbert transform [8], rate conversion [7], communication modulation [9], image processing [1], audio signal processing [4] modeling. At the PC front end Matlab/Simulink package [1,4,8] or self-developed visualization tools [7,9] are most often utilized.

The goal of this paper is to fill the gap of adaptive filters application implementation for the practical teaching needs. To achieve this the functionality and possibilities offered by the setup composed of two popular DSK6713 Starter Kits are presented. These are often offered via the university program of Texas Instruments. DSK6713 can also be acquired from their manufacturer [2]. In educational institutions widely accepted Matlab/Simulink modeling package is used for the PC software front end. Also, individual student task generation methodology is a part of this presentation.

2. Tools and experimental setup

The considered experimental setup is composed of two DSK6713 Starter Kits interconnected using standard audio cables. The key component of the Starter Kit is Texas Instruments DSP TMS32C6713 featuring performance of up to 400 MMACs. Analog-to-digital and digital-to-analog conversions are
implemented with on-board codecs. The whole setup is shown in the Fig. 1.

The first DSK is connected to the PC using USB interface. This board is dedicated to model an unknown system to be identified or communication channel to be equalized by the second DSK. The second DSK needs to model operation of the adaptive filter. This board is actually the one which program will require more tuning, algorithms testing as well as results and generated data observation. Therefore, to have more advanced data feedback to PC the DSK is connected via JTAG emulator XDS560. Using the RTDX data exchange method sampled signals can be sent for the further processing and visualization to the Matlab/Simulink environment. The Simulink model is loaded to the DSK using just several mouse clicks. The tool chain „Graphical Simulink model->CCS Compiler->DSK6713 hardware“ enables to implement DSP algorithms on the real hardware avoiding the low level processor programming. CCS stands for the Code Composer Studio from Texas Instruments. XDS560 emulator is the most expensive tool in the considered setup. Connection of the DSK boards using USB interfaces (indeed DSK boards have on-board JTAG emulator) was not possible due to USB PC driver (supplied by the manufacturer of DSK6713) inability to distinguish two connected DSK boards [2].

The further details about the setup are discussed presenting two AF applications.

3. Applications

3.1 Unknown system identification

In the Fig. 3 the Simulink model to be loaded to the DSK board No. 2 is shown. Standard components from the Embedded Target for TI C6000 DSP/DSK6713 Board Support toolbox are used to interface analog signals to the DSK: DSK_ADC and DSK_DAC. The LMS filter component is also available from Simulink Signal Processing Blockset. Despite the high speed of the RTDX channels (over 2 MB/sec) data flows generated by the DSP processing data in real time can hardly be managed by the PC. Therefore the model includes the down sampling components that send only each 64-th sample of the LMS Filter Error and each 64-th array of the adaptive filter coefficients to the RTDX channel. The data from the left channel is received from the excitation signal generator and redirected to...
the desired signal port of the adaptive filter, while the signal from the right channel is received from the system to be identified (see also Fig. 1).

**Figure 3. Unknown system identification**

The work flow with the setup can be listed in the following manner:

1. Run the model shown in the Fig.4, on the DSK No.1. The part of the model between DSK_ADC and DSK_DAC implements the system that will be the unknown system for the DSK No.2. The adaptive filter of the DSK No.2 model will seek to identify the unknown system.

2. Compose simple model for excitation signal generation (see Fig. 5) utilizing PC sound blaster. Build executable file of this model and run it from Microsoft Windows command prompt. Execution of the excitation signal generation model from Simulink is not possible because Matlab/Simulink environment will need to run scripts that will communicate with the model loaded to the DSK No.2. Simultaneous execution of several Simulink models is not possible.

**Figure 4. Unknown system for identification**

3. From the Simulink environment compile and load to the DSK No.2 the model shown in the Fig. 3. Load and run actions are implemented in the Matlab script, that initializes Link to CCS toolbox functions to connect to Code Composer Studio, configure real time data exchange (RTDX) channels and visualize data after its reception from the DSK No.2.

**Figure 5. Excitation signal generation**

Changing the type of the adaptive filter (LMS, RLS), tuning parameters of the adaptive filter (filter length, adaptation step size, leakage factor, etc.) as well as the excitation signal parameters can provide a lot of space for adaptive filters investigation in a laboratory.

### 3.2 Channel equalizer

Communication channel equalizer analysis setup can be implemented in the same framework (see Fig.1) without changing any hardware modules or their interconnections. In the Fig. 5 Simulink model of the channel equalizer is shown. Its complexity is expanded compared to the unknown system identification model (see Fig. 3) in order to utilize more functionality offered by the DSK boards. Indeed the goal is to have possibilities to put equalizer to training and direct modes using hardware buttons provided on the DSK.

Both channels from the linear output of the DSK No.2 are connected to the linear input of the PC sound blaster card.

Running the model shown in the Fig.7 and manipulating switches on the DSK No.2 it is possible to visualize real signals from the points of interest of the communication channel equalizer application.

The following work flow with the setup running equalizer application can be suggested:

1. Run the communication channel model similar to the shown in the Fig.4 on the DSK No.1.
2. Determine the delay needed in the equalizer model (see Fig. 6). To do this one can:
   a. Set delay of the Delay component equal to 1 sample in the channel equalizer model (Fig. 6). Then run it on the DSK No.2.
   b. Using DSK No.2 on-board switches force it to route input $u_{input}$ and desired $u_{desired}$ signals to the linear output via DSK_DAC.
c. Launch the model given in the Fig. 7 to observe the delay introduced by the communication channel model. Typical signal records are presented in the Fig. 8 a). Applied communication channel model was the 6-th order FIR filter. However, the delay observed is obviously higher. This is due to the frame based data processing in the DSK_ADC module of the DSK No.1. The frame based processing loads the DSP significantly less than the single sample based processing. In the frame based processing methodology data samples are collected to the buffers of the set length. When the buffer becomes filled the software function to process data or transfer it according to the model is invoked. In the sample based processing each new sample causes call to the processing routine.

Despite the computing power of the DSP the equalizer operation in the sample based processing mode incoming data flow handling in real time was impossible. Therefore, buffer length of the DSK No.1 was set to 32 samples. Using the model shown in the Fig. 7 it was experimentally found that the delay between the input signal and the desired signal is \( \tau = 96 \) samples/\( F_s = 11.4 \) ms. The delay component in the equalizer model then has to be set to 96 samples to ensure input and desired signals enter appropriate inputs of the LMS filter without time shift.

3. Then training of the equalizer is performed. During this stage the LMS filter adapts its coefficients to meet least squares error minimization goal.

4. Finally, in the direct equalization mode the adaptive filter coefficients updating is stopped. Fig. 9 indicates records obtained prior to the LMS filter training (a) and after its training (b). It can be clearly seen that pulses distortion introduced by the channel are reduced at the output of the equalizer. This is the evidence of intersymbol interference impact reduction [3]. Quality of the pulse shape reconstruction can be further investigated analyzing signal records obtained using the considered setup.

The channel equalizer model and the excitation signal generation model only can be loaded directly from Texas Instruments Code Composer Studio if one needs to enter signals of interest using PC sound blaster card. Indeed in such a case model shown in the Fig. 7 is the one...
active in the Simulink environment. Two models can not run simultaneously from the Simulink environment. Alternatively, if the channel equalizer model is loaded from the Simulink then RTDX channels become available and filter coefficients can be downloaded to the Matlab and frequency transfer function or impulse response of the adaptive filter can be plotted.

![Figure 8. Delay introduced by the channel model (a) and its compensation by the additional delay z=96 (b)](image)

**4. Discussion**

The advantage of the described setup is that many different models of the unknown system or communication channel can be generated by the course supervisor and distributed among students in the form of binary executable of the DSP processor. These task files can be simply generated by changing settings of the model given in the Fig. 4 and building them with the Simulink and Code Composer Studio development tools. The process is fast and takes just several mouse clicks. The direct analysis of the contents of generated binary file is meaningless and the only way to identify the system transfer function or equalize the channel is to operate the given executable on one of the DSK boards and apply the second DSK board to solve the task. Presented work flow is given only for the starter and of coarse may be modified to design alternative problem solutions and subject learning techniques.

![Figure 9. Excitation signal distortion (a) and reconstruction (b) at the equalizer output](image)

To name some limitations of the setup I would mention a considerable time of the build-generate-load cycle of the model to be implemented in the DSK. It is due to the large amount of processing performed both by the Matlab/Simulink and the compiler. This condition puts significant requirements to the PC performance increasing the price of the work place. Another expensive part of the setup is the XDS560 emulator. Hopefully, in the future USB drivers supporting multiple DSK boards connected via USB ports will be released by the manufacturer. In that case XDS560 emulator could be removed from the setup.

Software simulation without any doubts is faster and cheaper approach to digital signal processing applications teaching. However, applied solely it does not reveal reality of the real
embedded systems implementation techniques, like frame based processing discussed in the previous section. Therefore, to my opinion software simulation and hardware implementation should be combined to assure teaching quality. In addition, using hardware tools often raises student motivation. Acceptability and relevance of the setup are to be tested in the near future.

5. Conclusions

The proposed experimental setup for the adaptive algorithms teaching is composed from the hardware and software tools that are widely accepted in academic environments and are often supplied by the so called university programs of manufacturers.

The setup structure can be applied for both unknown system identification and communication channel equalization implementation using adaptive filter without any hardware rewiring.

Using the presented setup many different tasks may be conveniently generated by the supervisor and distributed among students in coded binary format hiding internal implementation of the given system.

Frame based and single sample data processing techniques for real time data processing can apparently be faced and their influence on the whole system implementation realized.

6. References


