DSP Implementation of an Amplitude Modulation Transmission System: A Capstone Design Approach

Cherian P. Mathews, Adrianne Candia, Waleed Kader
University of West Florida / Gulf Power Company /
University of Florida

Abstract

All students in the Electrical Engineering Program at the University of West Florida (UWF) are required to undertake a capstone design project. This paper presents an overview of the capstone design approach at UWF. The structure of the design course, the use of design milestones, and the role of the faculty mentor are discussed. The paper also presents the capstone project undertaken by a team of two students (Adrianne Candia and Waleed Kader). Their project involved the design of an amplitude modulator to modulate a voice signal onto a carrier, and an amplitude demodulator to demodulate the AM signal. The modulator and the demodulator were implemented using Texas Instruments C3x series floating-point digital signal processors. The project components included design of an anti-aliasing filter and amplifier at the front end, C-code to perform amplitude modulation, C and assembly code for amplitude demodulation (computationally intensive filtering operations were performed by C calls to assembly routines), and the design of a circuit board to support boot-loading of the AM demodulation code into a TMS320C31 processor.

I. Introduction

All students in the Electrical Engineering Program at the University of West Florida are exposed to a capstone design experience. Senior students are required to take the semester long (15 week) course *EEL 4914: Electrical Engineering Design* during which they undertake a capstone design project. This paper provides an overview of the capstone design approach at UWF. It also provides an overview of the design project undertaken by two students, Adrianne Candia and Waleed Kader. Their design project involved digital signal processor (DSP) based implementation of an amplitude modulation (AM) transmission system.

II. The capstone design approach at UWF

The UWF/UF joint program in Electrical Engineering offers a University of Florida (UF) BSEE degree to students at the UWF campus in Pensacola. The joint program is only about 5 years old and presently has an enrollment of about 100 students. The enrollment for the design course EEL 4914 has not yet exceeded 15 students. This small class size has allowed the entire EE program faculty to be involved in the course, as described below:

A faculty member serves as the course coordinator for EEL 4914 and provides overall supervision for the course. Students are encouraged (but not required) to work in teams. Each team has to come up with an idea for a design project. Once a project has been identified, the team selects a faculty mentor who has expertise in the project area. The mentor provides guidance and assistance to the team over the course of the semester. The course coordinator and the faculty mentor serve on the examining committee of each team. The coordinator is familiar with all projects being undertaken and provides input to the faculty mentor that is valuable in maintaining equity in the grading process.

The coordinator meets with the students during the first week of the semester and briefs them on the course organization and requirements. Issues such as the maintenance of a patent-style notebook to document design work, and guidelines for writing final project reports are addressed. Students are presented with the following design milestones: Week 3: Preliminary design review, Week 6: Critical design review, Week 14: Final acceptance test, Week 15: Final acceptance. The requirements at each of the design milestones are detailed below.

Preliminary Design Review: During the third week, each team makes a presentation before the entire faculty detailing the scope and specifications of their project. After hearing all presentations, the faculty confer and decide whether the project proposals are acceptable. The faculty often propose minor changes to project specifications or design approach. On rare occasions the faculty rejects a project idea and asks the team to come up with a new project proposal.

Critical Design Review: During the sixth week, each team makes a presentation in which they outline their progress and present detailed design plans, schematics, flowcharts, etc. showing how they intend to meet their design specifications.

Final Acceptance Test: Each team performs a project demonstration during the fourteenth week. The course coordinator and faculty mentor evaluate whether the design specifications have been met. If the specifications have not been met, the team is given a “punch list” identifying any deficiencies that need to be corrected. Teams also submit a draft of their final project report at this time.

Final Acceptance: During the fifteenth week the team has a final opportunity to demonstrate their project and show that deficiencies listed on the punch list have been satisfied. Final project reports are due the week after final acceptance.

The faculty mentor (in consultation with the course coordinator) assigns grades for the team members. Weights used in determining the final grades are as follows: preliminary design review: 5%, critical design review: 20%, patent-style notebook: 10%, project construction/packaging 10%, project report: 15%, mentor evaluation: 40%.

Students registered for the design course are provided with 24 hour access to the design lab. The lab contains seven stations equipped with computers and other test equipment. The EE program provides supplies of commonly used ICs and other discrete components to the design teams. Specialized components required for projects have to be procured by each individual team.

The capstone design approach presented above has been very successful at UWF. Students teams have drawn on knowledge gained over the entire EE curriculum and designed and built some excellent projects.

III. Project Definition and Specifications

Adrianne Candia and Waleed Kader were students in the EE program who had taken a DSP course/lab taught by the first author. The lecture part of the course covered DSP fundamentals and filter design theory, while the lab component focused on real-time implementation of DSP systems using a Texas Instruments TMS320C30 evaluation module (C30 EVM). The C30 EVM is a circuit board built around the 32 bit TMS320C30 floating point DSP processor. The board (which plugs into a slot in a host computer) contains the DSP processor and supporting circuitry such as an analog interface circuit (AIC) and external memory. After completing the course, the students approached the first author and expressed interest in doing a DSP processor based design project – they proposed designing a simple AM transmission system.

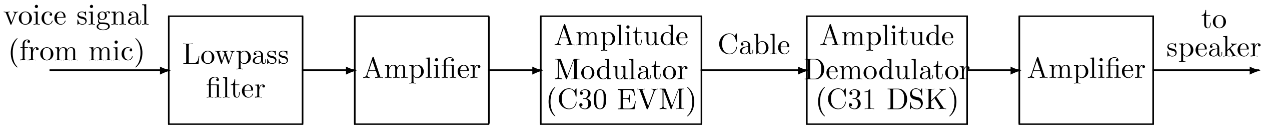

Figure 1: Block diagram of the project

Fig. 1 depicts a block diagram of the project. The project involved taking a voice signal from a microphone (bandlimited to the range 20Hz - 2.5kHz) and amplitude modulating it onto a carrier. The modulation was to be performed by a C30 EVM running C code. The amplitude modulated voice signal then had to be demodulated using a TMS320C31 DSK (DSP starter kit). The code for demodulation was to be written in C, with computationally intensive filter routines being implemented by calls to assembly language routines. The C31 DSK was to operate as a stand-alone unit, with the demodulation code being bootloaded into it from an erasable programmable read only memory (EPROM). The demodulated (voice) signal from the DSK was to drive a speaker.

The students divided the project responsibilities among themselves as follows: Ms. Candia would be responsible for all the algorithm development and coding, while Mr. Kader would construct the required analog filters, amplifiers, and bootloading circuitry.

IV. Amplitude Modulation

The voice signal bandwidth specification $B = 2.5$ kHz led to the selection of an AM carrier frequency $f_c = 3.7$ kHz. The AM spectrum is known to contain no frequency components above $f_c + B = 6.2$ kHz. The Nyquist sampling theorem dictated that a sampling rate of at least $6.2 \times 2 = 12.4$ kHz be employed to digitize the voice signal. A sampling rate $f_s = 16$ kHz was selected (the highest sampling rate possible using the analog interface circuit on board the C30 EVM is 19.2 kHz).

The microphone at the front end of the system was a Panasonic omnidirectional electret microphone. In keeping with the project specifications, the microphone output had to be bandlimited to 2.5 kHz. This was accomplished using an 8^{th} order Chebyshev lowpass filter designed using standard techniques. The filter was comprised of four cascaded second order sections, each built around a 741 operational amplifier. The filter had a halfpower frequency of 2.1kHz and provided an attenuation of about 36dB at a frequency of 3.5 kHz.

The output signal from the lowpass filter was found to lie in the range -70mV to 70mV for normal to loud voice levels at the microphone. The filter output was amplified (using an op-amp based amplifier with a gain of about 10) to bring the signal level close to the input signal range (-0.75V to 0.75V) of the analog interface circuit (AIC).

The AIC contains a 14 bit ADC (analog-to-digital converter) and a 14 bit DAC (digital-to-analog converter). The AIC interfaces to the C30 processor via one of two serial ports present on the processor. The AIC was used in synchronous mode (a digitized voice sample from the ADC is transmitted to the processor at the same time that an output word from the processor, representing a sample of the AM signal, is transmitted to the DAC). The input signal acquisition and output signal generation were performed under interrupt control (using the serial port transmit interrupt). Amplitude modulation of the input signal from the ADC was performed using C code running on the C30 processor.

Let $x(t)$ denote the voice signal to be modulated. The corresponding AM signal is

$$s(t) = A_c[1 + K_a x(t)] \cos(2\pi f_c t) \tag{1}$$

where A_c is the carrier amplitude, f_c is the carrier frequency, and K_a is the amplitude sensitivity coefficient. The processor samples the voice signal $x(t)$ every $T = 1/f_s = 62.5\mu s$ and generates a corresponding AM output sample. The argument of the sinusoid in the above equation increases by the amount $2\pi f_c T = 1.4529$ radians during one sample period. The nth sample of the output AM signal is thus

$$s(n) = A_c[1 + K_a x(n)] \cos(1.4529n) \tag{2}$$

The C code for amplitude modulation is listed in Fig. 2. The carrier amplitude $A_c = 0.042$ and amplitude sensitivity $K_a = 0.695$ were chosen to provide an output signal swing that spans the entire range of the DAC. The main program first initializes the serial port 0 transmit interrupt vector. It then calls the function aicinit_i() which initializes the AIC for interrupt driven acquisition via serial port 0. The program then goes into an infinite loop.

Program control is transferred to the interrupt service routine c_int05() when a serial port 0 transmit interrupt occurs. The function sp0txrx is invoked to transmit the previous AM output sample (outdata) to the DAC and receive a new sample of the input voice signal (indata) from the ADC. The input word from the ADC is next converted to a voltage level X

```
float X, S, k=0;
float pi=3.14159265359;
float Ac=0.04204, Ka=0.6954775;
float con=1.831278e-4, angle=1.45298660229;
int indata, outdata=0;
volatile int *intvec = (volatile int *) 0x00;     /*pointer to start
                                                  of interrupt vector locations */
/********* Main Program *********/
main()
{
  intvec[5] = (volatile int) c_int05;             /* initilize SP0 transmit vector */
  aicinit_i();                                    /* initialize AIC (interrupt driven I/O) */
  for(;;);
}

/*Serial port 0 interrupt service routine */
void c_int05()
{
  indata = sp0txrx(outdata);                      /* receive indata from ADC */
                                                  /* and send outdata to DAC */
  X = indata * con;                               /* X = voltage level of input signal */
  S = Ac * (1 + Ka * X) * cos(k);                 /* voltage level of output AM signal */
  outdata = floor(S / con);                       /* convert to integer value for DAC */
  k = k + angle;                                  /* update argument of cosine */
  if (k > pi) k = k - 2*pi;                       /* ensure range is -pi to pi */
}
```

Figure 2: C code for amplitude modulation

by multiplying by the factor con (which is the voltage level per bit of the ADC). The voltage level S of the AM signal is next computed according to (2). This value is then converted into a corresponding integer word (outdata) for transmission to the DAC. Finally the argument k of the cosine is updated in preparation for the next input sample (the program ensures that k lies in the range $[-\pi, \pi]$).

V. Amplitude Demodulation

Amplitude demodulation was performed using a C31 DSK (a board costing \$99). It is well known that the real envelope of a bandpass signal $s(t)$ is $e(t) = \sqrt{s^2(t) + \hat{s}^2(t)}$, where $\hat{s}(t)$ is the Hilbert transform of $s(t)$. The envelope of the AM signal $s(t)$ of (1) is $e(t) = A_c[1 + K_a x(t)]$. A scaled version of the voice signal $x(t)$ can be recovered from $e(t)$ by removing the DC offset present in it.

Fig. 3 depicts a block diagram of the demodulation algorithm. The incoming AM signal is

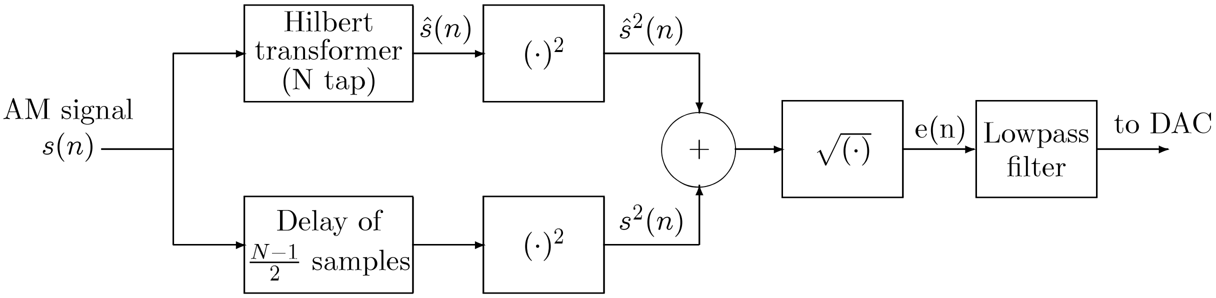

Figure 3: Block diagram of demodulation algorithm

digitized by the ADC (operating at 16 kHz) on the C31 DSK. The digitized AM signal $s(n)$ is passed through a N tap finite-impulse-response (FIR) Hilbert transformer. The resulting output $\hat{s}(n)$ is then squared and added to the square of a delayed version of $s(n)$ (the delay of $(N-1)/2$ units corresponds to the delay introduced by the FIR Hilbert transform filter). The real envelope $e(n)$ is obtained by taking the square root of this sum. The envelope $e(n)$ is next lowpass filtered to remove any high frequency components present in it. The output samples of the digital lowpass filter are then transmitted to the DAC. The DAC output is capacitively coupled to an amplifier whose output is fed to a speaker which plays back the demodulated signal.

All the operations in the block diagram of Fig. 3 (except for the Hilbert transform) were implemented using C code. The $N = 47$ tap Hilbert transformer was implemented in assembly language and called from C. The Hilbert transformer was designed in Matlab using a (Kaiser)window-based method. The filter had a passband that extended from about 615 Hz to 7380 Hz. This passband covers the spectral region 700Hz to 7200 Hz ($f_c \pm B$) occupied by the AM signal $s(t)$. The magnitude and phase responses of the Hilbert transformer are depicted in Fig. 4. Frequency components higher than 2.5 kHz (the known bandwidth of the input voice signal) present in the envelope $e(n)$ were rejected using a digital lowpass filter. The filter (a fourth order elliptic filter) was designed using Matlab to meet the following specifications: passband edge frequency = 2.5 kHz, stopband edge frequency = 3.5 kHz, passband ripple = 1dB, stopband attenuation = 40dB. The difference equation relating the filter input $x(n)$ and the filter output $y(n)$ is $y(n) - 2.2239y(n-1) + 2.5475y(n-2) - 1.4969y(n-3) + 0.4012y(n-4) = 0.0380x(n) + 0.0312x(n-1) + 0.0632x(n-2) + 0.0312x(n-3) + 0.0380x(n-4)$. This lowpass filter was implemented in C using the direct form II structure.

The C31 DSK was to bootload the demodulation code and operate as a standalone unit upon reset. Bootloading was accomplished as follows: The demodulation object code was first produced by compiling, assembling, and linking the source code. The object code was then converted into hexadecimal format (suitable for programming an EPROM) using the hex conversion utility supplied by Texas Instruments. The resulting hex data was then burned into a 16 Kbyte EPROM. A daughter-card to house the EPROM and interface to the DSK board was constructed. The EPROM was mapped into the DSK memory space beginning at address 1000 (hex). Address decoding for the bootload memory was provided

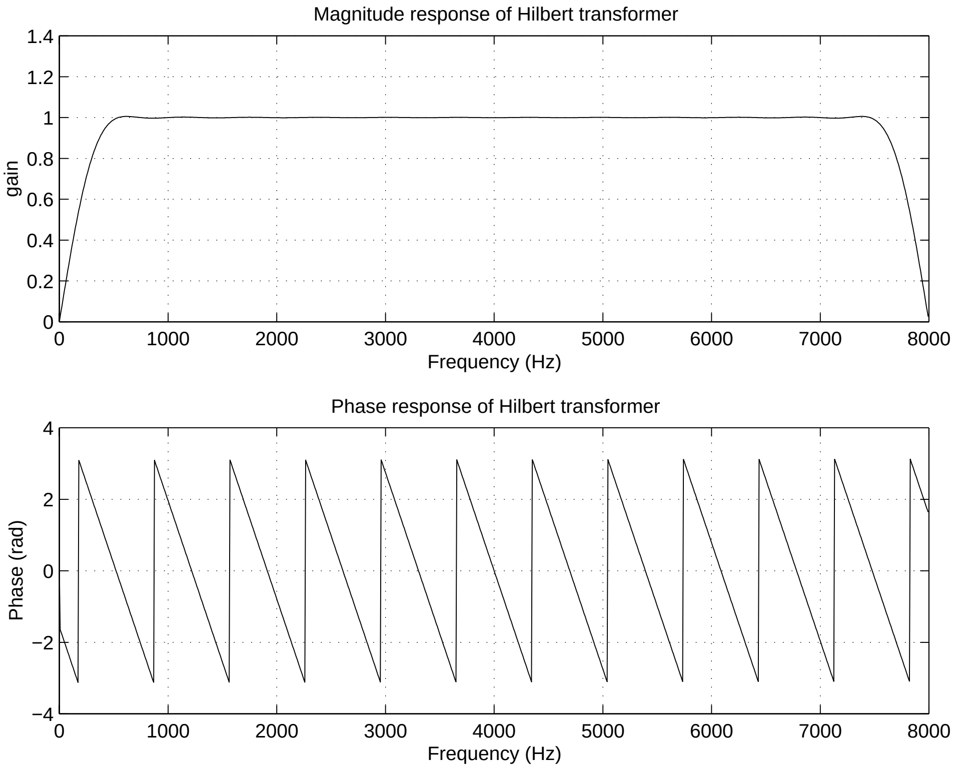

Figure 4: Frequency response of the Hilbert transformer

by the DSK. The daughter-card contained push-button switches to reset the DSK and assert the INT2 interrupt. A reset followed by an INT2 interrupt caused the DSK to bootload the demodulation program from memory and execute it.

VI. Challenges Faced in the Design Process

The students had no experience with the C31 DSK board before they began the project. Learning about all the details specific to the DSK board was a challenge. It was necessary to wade through manuals to learn about the details of C called assembly language functions, usage of the TI hex conversion utility, and the specifics of program bootloading using the C31 DSK. Difficulties in getting the C31 DSK and daughter-card to work together for the bootload were eventually overcome after extensive debugging. The project finally worked according to specifications, much to the joy of all involved.

VII. Conclusions

An overview of the capstone design experience in the Electrical Engineering Program at the University of West Florida has been presented. Details on the organization of the design course were provided. An overview of a DSP based design project undertaken by a team of two students in the EE program was also presented.