Automatic Signal Transformer and Custom Filter Designer

Sushant Mahadik
EXTC
ACE, Malad (W)
Mumbai, India

Karan Mali
EXTC
ACE, Malad (W)
Mumbai, India

Sameer Mohadikar
EXTC
ACE, Malad (W)
Mumbai, India

Tejas Mulye
EXTC
ACE, Malad (W)
Mumbai, India

Prof. Jyoti Kolap
EXTC
ACE, Malad (W)
Mumbai, India

***Abstract*— A user is able to draw a transfer function on a touchscreen which is connected to the PIC. A custom interpolation/interpretation algorithm is used to understand the essential passband, stop bands of the user drawn transfer function. These bands are used as the input to either a Parks-McClellan or Fourier Transform algorithm. This creates the discrete impulse responses in terms of coefficients corresponding to the drawn frequency response in time domain. The discrete time finite impulse response (FIR) filter coefficients are stored in memory. Then convolution of this coefficients is taken with an analog to digital converter sampled input signal. The generated output is given to a 12-bit digital to analog converter to get the digital representation of input signal. In short user can analyze the drawn characteristics of the filter in real sense.**

***Keywords*— PIC; FIR; Fourier; Parks- McClellan; Interpolation**

I. INTRODUCTION

This paper is based on digital signal processing. A user draws a transfer function on a touchscreen provided with the PIC. A custom interpolation/interpretation algorithm is used to understand the essential pass band-stop bands of the function on the screen. These bands are then saved. Then these bands are input to either a Parks- McClellan or Fourier Transform[1]. The selected algorithm from either of two which creates the discrete impulse responses corresponding to the drawn frequency response in terms of coefficients. The discrete time finite impulse response (FIR) filter coefficients are stored and convolution is taken with an analog to digital converter sampled input signal. Then these are given as input to a 12-bit digital to analog converter to create final output . Thus, user can understand the different types of filter characteristics.

II. LOGIC STRUCTURE

The General block diagram that describes the logic of the paper is shown below. Basically the flow of the program algorithm is as follows: First, the user decides to use the Parks-McClellan algorithm or the Frequency Sampling Algorithm to implement FIR filter[1]. Second step would be that user can set the sampling rate, maximum amplitude of the transfer function plot that will be displayed on the touch screen. Next, user draws the desired transfer function on the touch screen. Then the chosen algorithm will begin to run. This will generate the filter coefficients related to the drawn frequency response. Once the coefficients have been generated, these are stored for further processing. At last input signal will be convolved in to produce the output in digital terms.

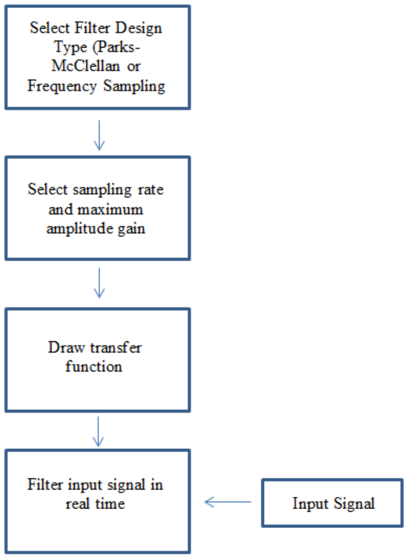
Select Filter Design Type (Parks-McClellan or Frequency Sampling
Select sampling rate and maximum amplitude gain
Draw transfer function
Filter input signal in real time
Input Signal

Fig.1

III. HARDWARE AND SOFTWARE TREADOFFS

The main tradeoff in this paper between hardware as well as software. That is the debate between digital and analog filtering of signals. Theoretically, all signals can be filtered in hardware. However, user must have knowledge frequencies needed to be filtered. Selection of filtering in software, is more computationally intensive, allows for greater flexibility for the user. So as to determine exactly how user want to filter their input. Thus, in terms of hardware and software tradeoffs, not only computational speed but also execution time is sacrificed for greater modularity.

IV. HARDWARE DESIGN

As will all the labs in this course, the PIC32 acts as the logical backbone for the entire system. A signal is input to the PIC through a 10-bit analog to digital converter (ADC) internal to the PIC32. The filtered signal acts as output consisting of a 12-bits to the MCP4822[6] digital to analog converter (DAC) over a serial peripheral interface (SPI). The DAC generates an output which is an analog signal representation of filtered input. During the creation of this paper a sinusoidal input from a signal generator is used as the input to the system. An oscilloscope is used to measure the DAC output. A schematic of the whole system is in the Appendix following the report.

A. ADC & DAC

There are three ADC channels namely, AN5 (port pin RB3) and AN11 (port pin RB13) to read the Y and X axis of the touchscreen respectively, as well as AN0 (port pin RA0) for the signal input. AN0 is not at the ADC channel until it is time for convolution to take place inside the state machine. The input of ADC and the output of DAC occur at the Nyquist rate of whatever input frequency the user selects from a list. They are contained within the interrupts of timer 2 and timer 3 match respectively. These timers are turned on and the ADC channel is set to AN0 at the same place. In this way the cycle time is only taken away by the interrupts from the executing program when it is absolutely necessary.

B. Touchscreen

The user drawn transfer function on a resistive touchscreen. The touchscreen used is a standard 320x240 resolution resistive touchscreen. Simple SPI interface is used to write data to the screen. Two resistive sheets are located on top of TFT display. On touching the sheet , the sheets touch and voltage divider is created which is measured through the PIC32's ADC. To determine touch coordinate, power is applied to one sheet and voltage is read to the other, the location of the touch results in larger or smaller division of the voltage divider and results in varying output voltages. This process is then flipped. The other sheet is powered and the ADC is used to measure the divided voltage off the other.

A significant delay on the order of 10 ms is used in between switching which sheet is powered for outputs to stabilize. This reduces the touch points per time resolution to below 50 times per second. This however is of little concern because a human will never be touching the screen that fast.The actual coordinates of the touch is represented as two numbers of 10-bits, following directly from their ADC measurements in either direction. Some scaling is necessary to get back to a pixel location, however to speed up the computation values relating to touch points to the screen positions are often hard coded as their ADC values instead of pixel value. Due to manufacturing defects and nature of two-resistive sheet geometry of the touchscreen, high and low ends of the 10-bit spectrum are impossible to obtain while determining the touch location. For this reason, an additional check and consideration is made to make sure a touch point is in the physically viable range of the screen.

C. External Interrupt

An external button is used to initialize an external rising edge interrupt on pin 16 of the PIC32[7]. This interrupt is used to return program execution to its start state and stop to convolution. As will be analyzed more, convolving the ADC input with an FIR filter in a real-time and therefore it is best interfaced with an interrupt service routine. The interrupt is triggered by using an external button because of its ease of implementation is software and reaction time with respect to the touch-screen display. The button is tied to the ground via a capacitor in order to confirm that the input to the PIC32 does not float and to reject noise or the sinusoidal input, located close in proximity to the button on the bread board.

V. SOFTWARE DESIGN

The paper is majorly concerned with a software intensive digital signal processing exercise It is divided into the three major design components; the touch-screen and drawing the filter transfer function on touch-screen; Parks McClellan and Fourier Transform algorithms and determining coefficients of FIR filter; and the real-time convolution with signal output. First it is worth investigating in the software structure before each component is addressed individually. The executing program is scheduled per the Protothreads cooperative scheduler. This paper uses two threads, first one to handle touch-screen reading and data parsing, and another to facilitate a state machine.

State Machine

The linear operation of state machine goes accordingly; first collecting user's data, like what kind of transform and particular parameters the user would like to use, and then moves towards drawing and processing the transfer function by interpolation and interpretation of graph drawn by the user, and final computing filter coefficients and outputting the signal[2]. State machine provided below.

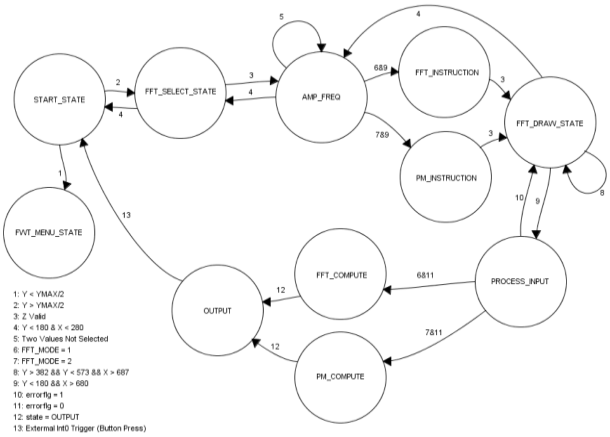

Fig.2

As shown in the above image, most of the transitions from state to state occur according to the touch input given to the system. Further, the error and transformation mode flags also play a role in surveying the machine.

VI. CONCLUSION

A theoretical introduction to digital filters has been presented. Implementation of this filters deals with a number of design objectives which include required area, consumed power, speed or latency of the designed filter. In the wake of today's digital world wherein signal processing has become an integral part of the technology. Huge transformation has taken place in the field of Signals and this paper perfectly showcases the modern development and also the potential for future development in this field. People who are not tech savvy will have an easier time in using the devices as all they have to do this just by appropriate and basic knowledge about filters & their characteristics.

VII. FUTURE SCOPE

There are little variations in terms of time and power consumed, when the processing speed increases. By solving these variations can further enhance the system performance. This work can be extended by increasing the order of filter for accurate outputs. The more advancement can be done with efficient use of algorithm/code with less error expectation. Efficient designs with minimized errors and interferences are still main research area in the field of filter banks.

REFERENCE

[1] McClellan, James and Parks, Thomas (2005), "A Personal History of the Parks–McClellan Algorithm", IEEE Xplore
[2] Meijering, Erik (2002), "A chronology of interpolation: from ancient astronomy to modern signal and image processing", Proceedings of the IEEE
[3] Kilgore, T. A. (1978). "A characterization of the Lagrange interpolating paperion with minimal Tchebycheff norm". J. Approx
[4] Ben Bolker Local Maxima and Minima
[5] Larson, Ron; Edwards, Bruce H. (2009).