GUI Design for FIR Filter on Matlab using a TMS320C3X DSK

Abstract: This work takes a step-by-step tutorial approach to building a Matlab Graphical User Interface (GUI) for implementing Finite Impulse Response (FIR) filter design on a TMS320C3X DSK. The GUI is driven by Matlab code, which takes input from various FIR filter parameters, plots the appropriate frequency response, and loads the FIR filter coefficients along with the DSK code to the DSP board. Matlab GUIs that interact with TI DSKs have proven effective for DSP applications in both the classroom and our research program.

Keywords: GUI, FIR, Filter, Matlab, DSP, Window, Frequency, Filter Order.

Introduction and Motivation

One of the difficulties for instructors teaching DSP programs is striking a balance between teaching fundamental theoretical ideas and offering hands-on experience with DSP hardware. The purpose of this study is to explain the usage of Matlab/DSK GUIs that allow students to quickly investigate DSP principles in Matlab and then download their Matlab-based designs to run on a TMS320C3X DSK. This method exposes students to DSP hardware without requiring DSP programming. Having students create their own Matlab/DSK GUI application might serve as a "fun" incentive tool for professors who want to integrate DSP programming into their courses.

Steps to Design Fir Filter Gui in Matlab

We will create a Matlab GUI to construct "window-based" FIR designs on a TMS320C3X DSK. The GUI created here is rather straightforward for better clarity, with only a few filter design choices and no "user input" error-checking capabilities. It is our responsibility to provide a Matlab/DSK GUI that will enable users to select FIR filter design characteristics including the filter order, filter type, and window function. The user can "click on a pushbutton" to load and execute the filter design on the DSK after selecting the required settings.

Step 0. To begin, we will need Matlab with the signal processing toolbox as the prerequisite, along with the 3X DSK software downloaded from TI's website.

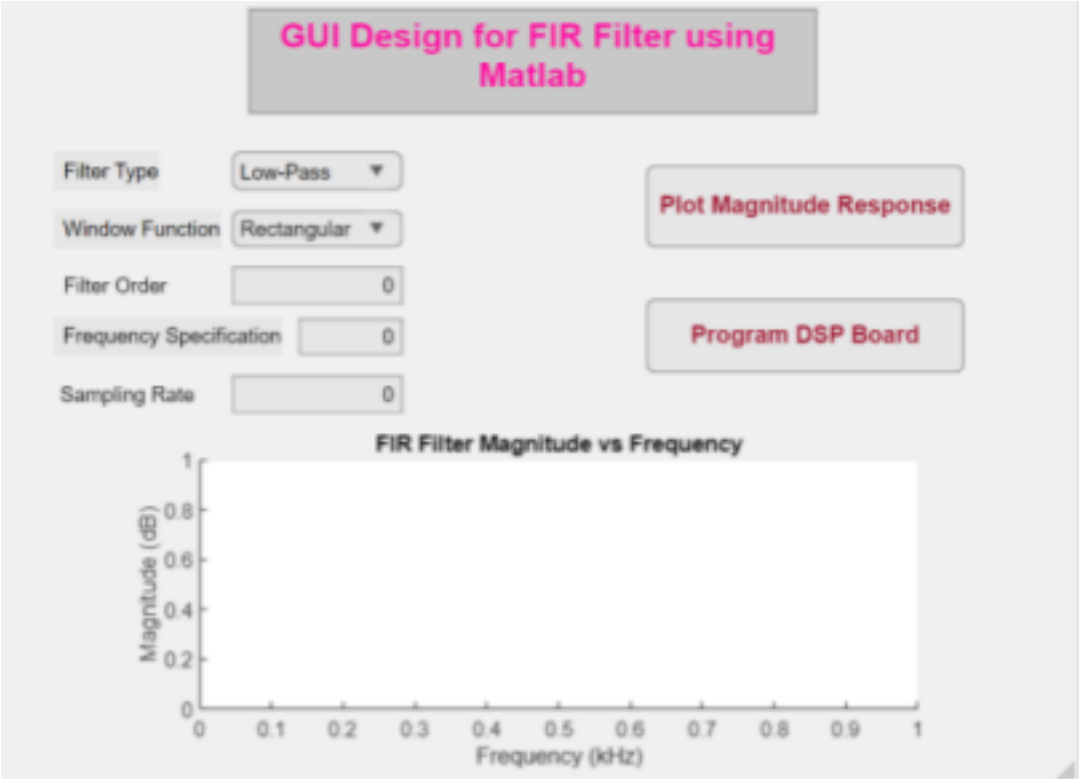
Step 1. Find the assembly language file *'fir.asm'* in the DSK program and copy it to another

file called *'myfir.asm'*. Open *'myfir.asm'* in a text editor (such as notepad) and look for the following block of code: It should be noted that alternative DSK FIR filter implementation code, such as that contained in, might also be used.

```
; size of the coefficeint table must be the same as the data
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; NOTE that the first two locations are skipped.  The first two
locations
; should not be used if the application is to be directly
bootloaded into
; the on-chip RAM.  The first two locations are reserved by the
bootloader
; as a small stack area.
;---------------------------------------------------------------
--------
ADC_recv      .set      $-2
              .loop     N-2
            .float    0.0
              .endloop
;- - - - - - - - - - - - - - - - - - - - - - - -
; FIR filter coefficients
;- - - - - - - - - - - - - - - - - - - - - - - -
            .include "myfir.asm";
;-------------------------------------------------------
; If more coeficients than buffers, generate an error
;-------------------------------------------------------
SZ            .set     END_coef-FIR_coef; Size of filter
               .if      N<SZ
              error -> The buffer length is less than coeficient
length
               .endif
;-------------------------------------
SIZE          .word    SZ                ; Size of filter
ADC_first     .word    ADC_recv          ;
ADC_end       .word    ADC_recv+SZ       ;
ADC_last      .word    ADC_recv          ;
FIR_coefx     .word    FIR_coef          ;
;-------------------------------------
; Define some constant storage data
;-------------------------------------
A_REG         .word  (TA<<9)+(RA<<2)+0 ; A registers
B_REG         .word  (TB<<9)+(RB<<2)+2 ; B registers
```

'myfir.asm' coefficients file exported from Matlab

Step 2. Create a preliminary version of the Matlab/DSK GUI by first creating an x-y plot (Matlab command >plot(0,0)) and then using "App Designer" with the Matlab command >>app designer (Make sure to use App Designer instead of Guide, since the latter would not be supported in the future versions of Matlab) to arrange the text boxes, list boxes, edit boxes, and pushbuttons as shown in the picture below. Change each GUI element's "String" field. If you are unfamiliar with this technique, we recommend that you read the Matlab material article "Building GUIs with Matlab" [2].

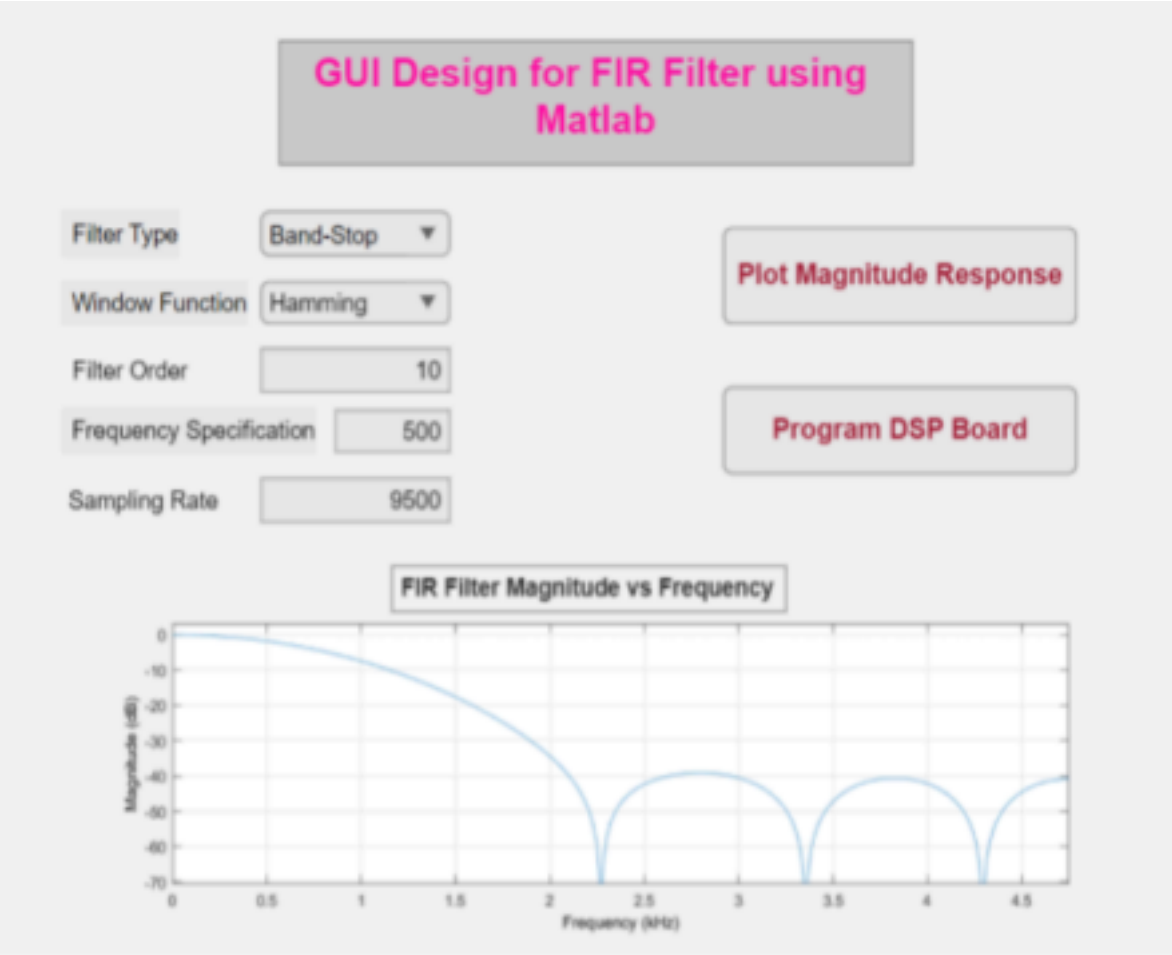


Rough layout of Matlab/DSK GUI

Step 3. Update the "Tag" field that corresponds to the Filter Type and Window Function list boxes to "Type" and "Window," respectively. This step enables you to quickly recognize the two list boxes when it comes time to read the information that the user has chosen. Follow the same for the "Filter Order," "Frequency Specification," and "Sampling Rate" edit boxes, using the "Tag" names "Order," "Specs," and "Sampling," in that order.

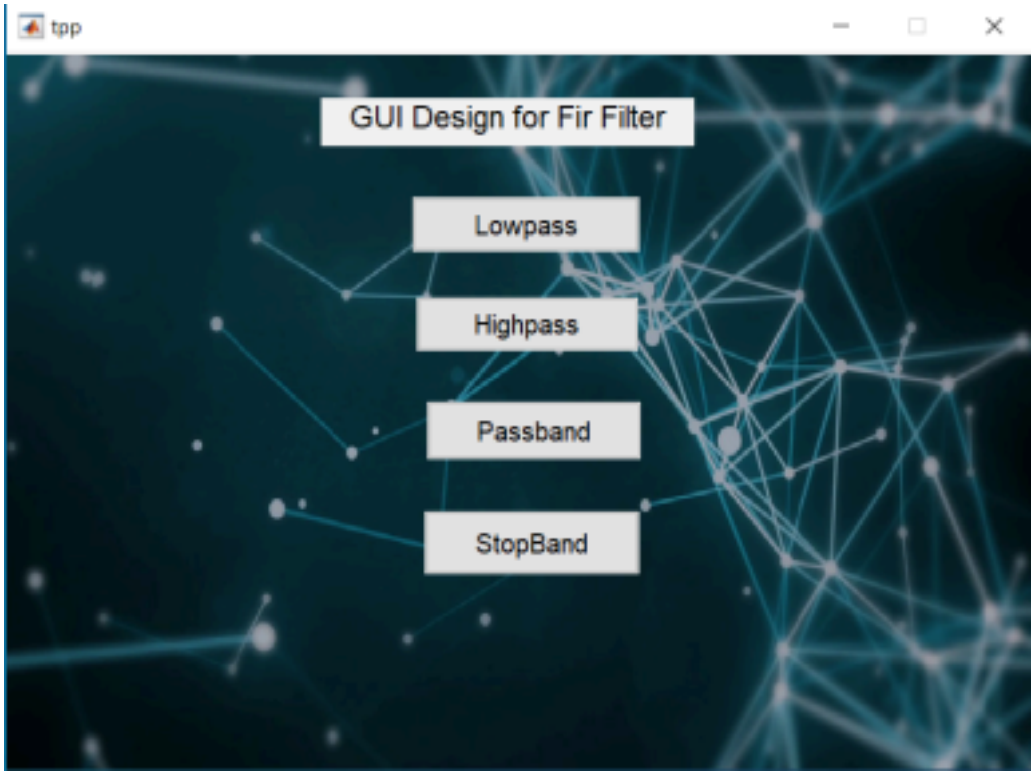
Step 4. Create multi-line strings in Matlab by typing
s1 = { 'Low-Pass'; 'High-Pass'; 'Band-Pass'; 'Band-Stop' } and
s2 = { 'Rectangular'; 'Hamming'; 'Hanning' }
In the Filter Type and Window Function list boxes, set the "String" field to s1 and s2, respectively. The Filter type and Window function options for the Matlab/DSK GUI are generated by this procedure.

Step 5. To match the final version of the Matlab/DSK GUI displayed in Figure 3, edit the String field of the Pushbutton boxes. Also, change the Callback field for the relevant pushbutton box to "fir_gui_action plot" and "fir_gui_action program." The idea is that the magnitude response would be plotted when you press the "Plot Magnitude Response" button and DSK programming would start by clicking the "Program DSP" button, by reading the proper FIR filter design options in Matlab.

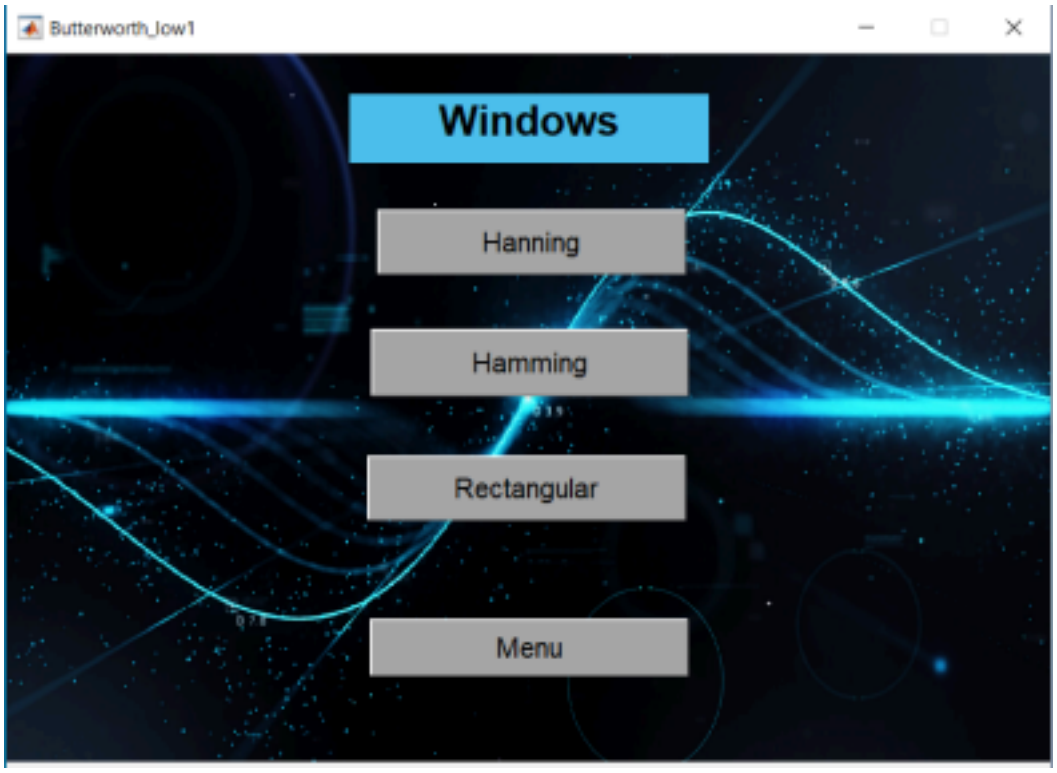


Matlab App GUI Design

Step 6. Matlab Code is used to "drive the GUI" at this stage. The comment lines in the Matlab code describe the procedures required to read the FIR filter parameters, display the proper frequency response, and load the FIR filter coefficients and DSK code to the DSP board.



Choose Filter Type GUI



Choose Window type GUI

FUTURE SCOPE

As making further advancements in our project, the now-made FIR Filter GUI Application can then be implemented on a hardware level. Moving on from Matlab to actual hardware, DSP Board can give in lots of constraints and is not just simply uploading the code into the processor, so taking that into consideration we have already made our application hardware compatible with the 3X DSK tool provided by Texas Instruments (TI). TI's Texas Made Semiconductor or TMS range of DSP starter Kit (DSK) Modules can be used for this purpose, likes of TMS320C3X DSK which comes equipped with all the required features and ports are among the best options available in the market for our use case.

CONCLUSION

In this paper, we have designed a Graphical User interface (GUI) for the Finite Impulse Response (FIR) Filter using the App Designer (previously 'guide') tool on Matlab. Various parameters for filter designing including Filter Type (Low-Pass, High-Pass, Band-Pass & Band-Stop), Windowing type (Rectangular, Hamming & Hanning). Further, Filter Order, Frequency Specification, and Sampling Rate values were also taken as the input parameters for our filter. A Magnitude (dB) vs Frequency (kHz) graph was plotted based on the given filter data. Thus an FIR Filter based on all the above-mentioned parameters was designed using the GUI. Moreover, we have also added provisions for further programming the actual hardware DSP board by flashing the code into it.

REFERENCES

1. Matlab/DSK GUI for FIR Filter Design Using a TMS320C3X DSK https://www.researchgate.net/publication/237751436_MatlabDSK_GUI_for_FIR_Filter_Design_Using_a_TMS320C3X_DSK
2. Rulph Chassaing, Digital Signal Processing: Laboratory Experiments Using C and the TMS320C31 DSK, Wiley-Interscience, 1999.
3. Mathworks, Building Graphical User Interfaces (GUIs) with Matlab, available as of 7/2000 at www.mathworks.com/support/graphics/docs.html.
4. FIR Filter Design https://in.mathworks.com/help/signal/ug/fir-filter-design.html 5. R. P. Narwaria1, Manish Kumar2, "Comparison of FIR Filter Using Different Window Functions"
6. V. Kumar, S. Bangar, S. N. Kumar and S. Jit, "Design of effective window function for FIR filters," 2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), Unnao, 2014, pp. 1-5.
7. M. Shil, H. Rakshit and H. Ullah, "An adjustable window function to design an FIR filter," 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Dhaka, 2017, pp. 1-5.
8. A. A. Eleti and A. R. Zerek, "FIR digital filter design by using windows method with MATLAB," Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2013 14th International Conference on, Sousse, 2013, pp. 282-287.

9. A. Mittal, A. Nandi and D. Yadav, "Comparative study of 16-order FIR filter design using different multiplication techniques," in IET Circuits, Devices & Systems, vol. 11, no. 3, pp. 196-200, 5 2017.
10. Proakis and D. G. Manolakis, Digital Signal Processing fourth edition, Prentice-Hall, 2007. 11. H. Rakshit and M. A. Ullah, "A comparative study on window functions for designing efficient FIR filter," 2014 9th International Forum on Strategic Technology (IFOST), Cox's Bazar, 2014, pp. 91-96.
12. Kaur, Er. Sandeep and Sangeet Pal Kaur. “Design of FIR filter using hanning window, hamming window and modified hamming window.” (2015).
13. M. Mottaghi-Kashtiban and M. G. Shayesteh, "New efficient window function, replacement for the hamming window," in IET Signal Processing, vol. 5, no. 5, pp. 499-505, August 2011.