Second-order sections – supplement to Chapter 4

Objective: To help understand the relationship between DF-I and DF-II second-order sections (SOS) to implement infinite impulse response (IIR) digital filters.

Background: In chapter 4 of our book, *Real-Time Digital Signal Processing: from MATLAB to C with theTMS320C6x DSPs*, 2nd edition, by Welch, Wright, and Morrow (CRC Press 2012) we discuss a number of forms for implementing IIR digital filters. However, we only develop and implement the code associated with direct form I (DF-I).

Discussion: As a convenient starting point we have implemented a MATLAB simulation of a second order low pass digital IIR filter. This simulation may be found in **SOSfilter.m**. This script file will plot both the frequency response of the filter (Figure 1) and the impulse response (IR) of the filter (Figure 2). In the second figure, a comparison between the impulse response calculated using MATLAB's **impz** command is made to our algorithm's results. This type of comparison is routinely performed to gain confidence in an algorithm. Calculating one of the mathematical norms of the two resulting signals would provide even higher levels of confidence in the algorithm, if you wish.

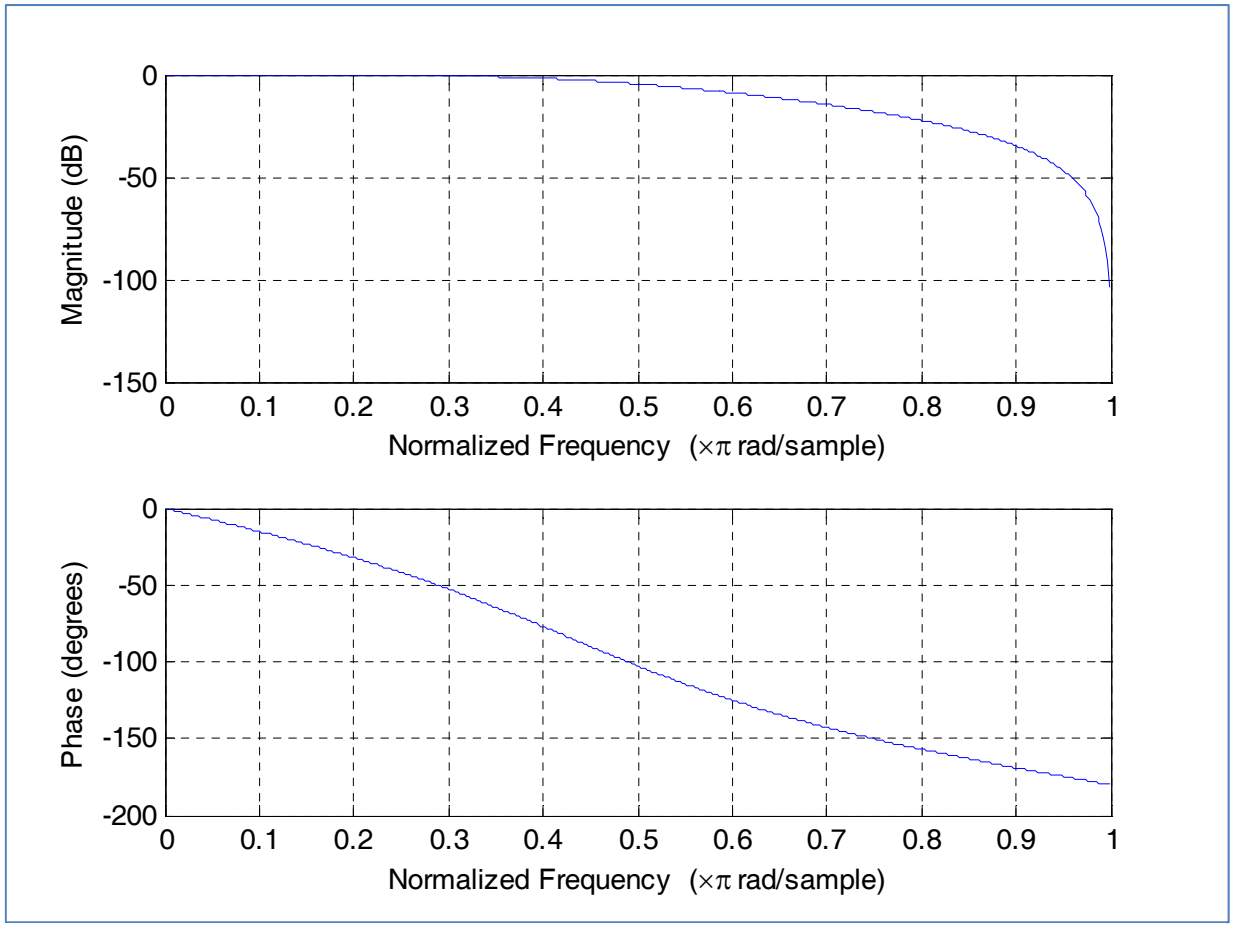

Figure 1

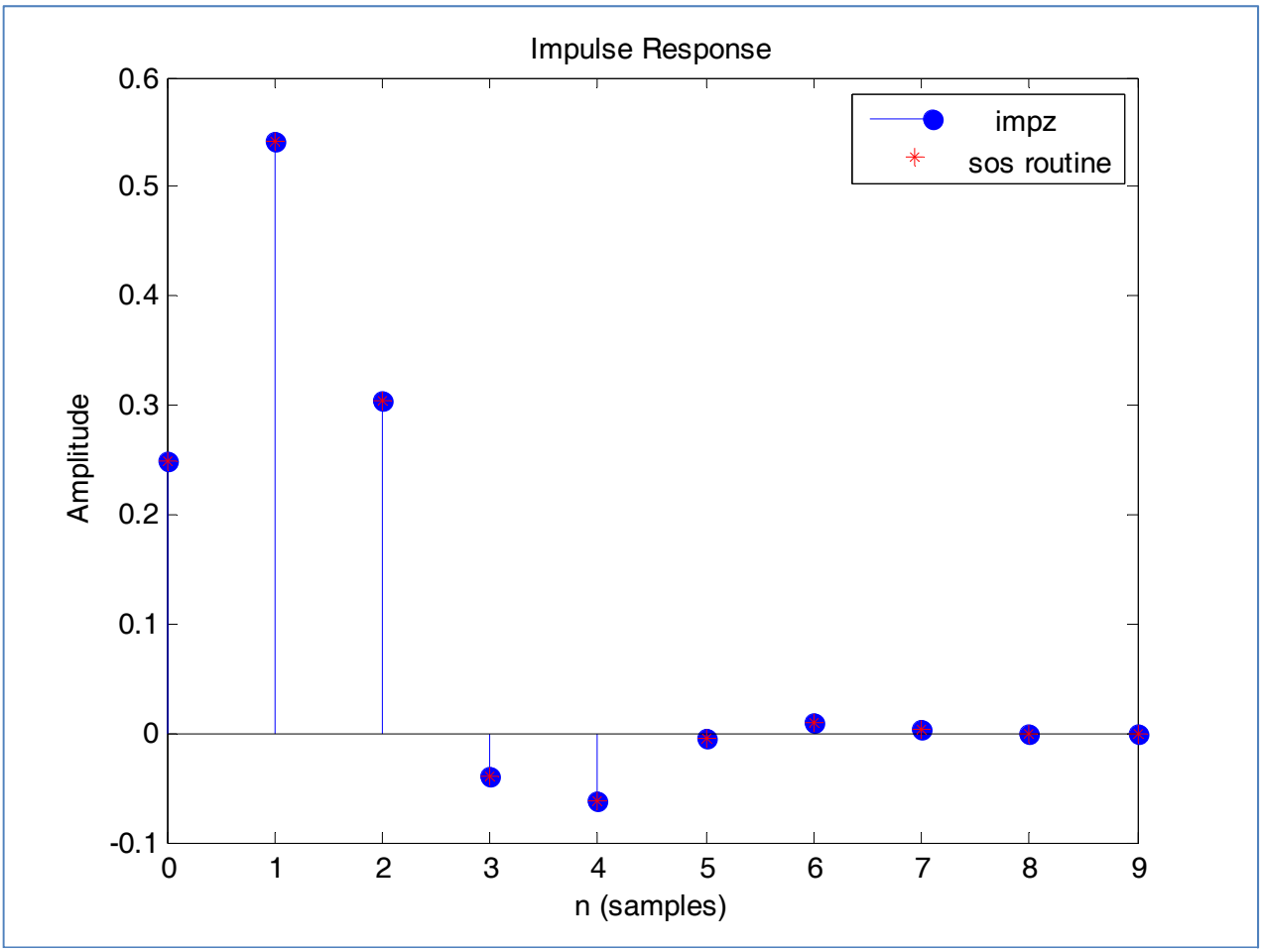

Figure 2

The next file, **SOSfilterRevA.m**, modifies the **SOSfilter.m** by turning the DF-I filtering operation into a function call. The function called by **SOSfilterRevA.m** is named **sosFiltFunDFI.m**. Everything else associated with this filter simulation remains the same. For consistency, this script file **SOSfilterRevA.m** generates the same two output plots.

The next file, **SOSfilterRevB.m**, modifies the **SOSfilterRevA.m** by turning the DF-I filtering operation into a DF-II filtering operation. The DF-II filter function called by **SOSfilterRevB.m** is named **sosFiltFunDFII.m**. Note that, unlike a DF-I function call, a DF-II function call uses w, where w is the current state of the filter. But DF-II is slightly more efficient than DF-I, as any DSP theory text will tell you. Everything else associated with this filter simulation remains the same. For consistency, this script file **SOSfilterRevB.m** generates the same two output plots.

The next file, **sosIIRmono_ISR.c**, implements the DF-II IIR filter as an interrupt service routine (ISR) written in C that runs in real-time using CCS. This will require one of the floating-point DSP boards and a working knowledge of real-time IIR filters that we developed in chapter 4. This ISR *does not* use a filter function. Don't forget to add the **StartUp.c** file to your project!

The final file, **sosIIRmonoFun_ISR.c**, implements the DF-II IIR filter as an interrupt service routine (ISR) written in C that runs in real-time, but using a filter function. Again, don't forget to add the **StartUp.c** file to your project!

File summary:

DF-I (direct form - I) simulation

SOSfilter.m – simulates the ISR associated with a single SOS

SOSfilterRevA.m – simulates the ISR associated with a functionalized single SOS

sosFiltFunDFI.m – is the SOS DF-I function called by **SOSfilterRevA.m**

DF-II (direct form – II) simulation/implementation

SOSfilterRevB.m – simulates the ISR associated with a functionalized single SOS

sosFiltFunDFII.m – is the SOS DF-II function called by **SOSfilterRevB.m**

sosIIRmono_ISR.c – is the real-time implementation of DF-II (without a filter function)

sosIIRmonoFun_ISRs.c – is the real-time implementation of DF-II (with a filter function)

Follow-on challenges: This update results in the implementation of a single SOS. Expand on what you have learned by accomplishing one or all of the following tasks.

- Implement a higher-order, but even-ordered, filter using a cascade of multiple SOS using multiple function calls.
- Write a new function that will implement the cascade of multiple SOS for you.
- Write a new function that will implement an odd ordered filter using cascaded SOS and one additional filter stage.
- How would you filter a stereo signal (signals from both the left and right channels)?