

IIR FILTERING: INTRODUCTION*

Douglas L. Jones
Swaroop Appadwedula
Matthew Berry
Mark Haun
Jake Janovetz
Michael Kramer
Dima Moussa
Daniel Sachs
Brian Wade

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 1.0[†]

Abstract

Infinite impulse response (IIR) filters are an alternative to finite impulse response (FIR) filters. Often, an IIR implementation can meet a given filter specification with less computation than an FIR implementation, but IIR filters induce nonlinear phase and are more sensitive to numerical problems.

1 Introduction

Like finite impulse-response (FIR) filters, **infinite impulse-response (IIR)** filters are **linear time-invariant (LTI)** systems that can recreate a large range of different frequency responses. Compared to FIR filters, IIR filters have both advantages and disadvantages. On one hand, implementing an IIR filter with certain stopband-attenuation and transition-band requirements typically requires far fewer filter taps than an FIR filter meeting the same specifications. This leads to a significant reduction in the computational complexity required to achieve a given frequency response. However, the poles in the transfer function require feedback to implement an IIR system. In addition to inducing nonlinear phase in the filter (delaying different frequency input signals by different amounts), the feedback introduces complications in implementing IIR filters on a fixed-point processor. Some of these complications are explored in IIR Filtering: Filter-Coefficient Quantization Exercise in MATLAB¹.

Later, in the processor exercise, you will explore the advantages and disadvantages of IIR filters by implementing and examining a fourth-order IIR system on a fixed-point DSP. The IIR filter should be implemented as a cascade of two second-order, Direct Form II sections. The data flow for a second-order,

*Version 2.22: Jun 3, 2009 3:27 pm +0000

[†]<http://creativecommons.org/licenses/by/1.0>

¹"IIR Filtering: Filter-Coefficient Quantization Exercise in MATLAB" <<http://cnx.org/content/m10813/latest/>>

Direct-Form II section, or **bi-quad**, is shown in Figure 1. Note that in Direct Form II, the states (delayed samples) are neither the input nor the output samples, but are instead the intermediate values $w[n]$.

Image not finished

Figure 1: Second-order, Direct Form II section
