EE 123: Digital Signal Processing	Spring 2007
Lecture 23 — April 10	
Lecturer: Anant Sahai	*Scribe: Chuo J. Liu*

23.1 Outline

In this lecture, we talked about implementation issue for rational transfer function.

- **Realization**
  - a) Direct Form
  - b) Cascade / parallel Form
- **Finite Precision Overview**

23.2 Rational Transfer Function Realization

There are 4 ways to build the filter with specific rational transfer function: Direct Form 1, Direct Form 2, Cascade Form, and Parallel Form. Let's say we have a filte with transfer function:

$$H(Z) = \frac{\sum_{k=0}^{M} b_k Z^{-k}}{1 - \sum_{k=1}^{N} a_k Z^{-k}} \tag{23.1}$$

With this transfer function, we are able to implemet directly from coefficient b_k and a_k using Direct Form 1.

23.2.1 Direct Form 1

With transfer function given in equation (23.1), we are able to realize the implementation using Linear Constant-Coefficient Difference Equation (LCCDE). We may re-write the transfer function in time domain:

$$y[n] = \sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k] \tag{23.2}$$

With this transfer function, we are able to implement the filter using Driect Form 1:

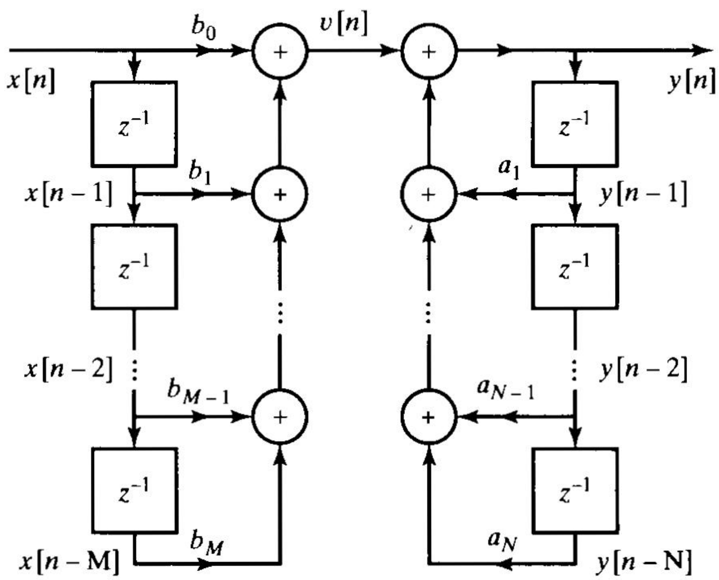


Figure 23.1 Filter design using Direct From 1.

23.2.2 Direct Form 2

Direct form 1 is a easy way to implement a filter. However, we figure out that it use too many delays, and we don't like that. We also find out that we may view direct form 1 as a transfer function with 2 separate transfer functions, one contains all the pole a_i, the other contains all the zero b_i, cascading together. Since we are implementing an LTI system, we are able to rearrange the poles and zeros transfer function without changing the property of the filter. Therefore, by rearranging the poles and zeros transfer functions, we can get our filter in Direct Form 2 (see figure 23.2).

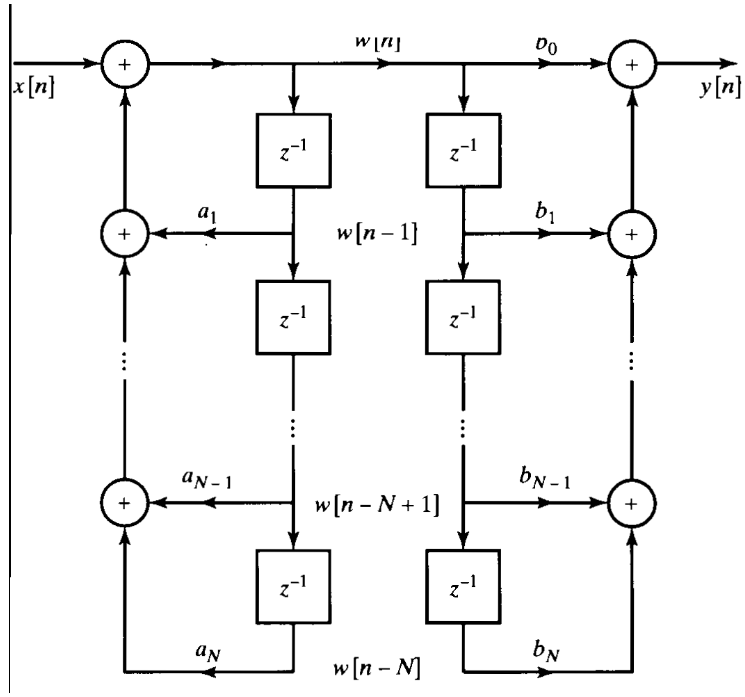


Figure 23.2 Rearranged DF1. Since the number of poles and zeros might not be the same, some of the coefficient may be zero, and N is MAX(M, N).

Then, we see that the delays are grouped in the middle, and we are able to reduce the number of delay used by almost half.

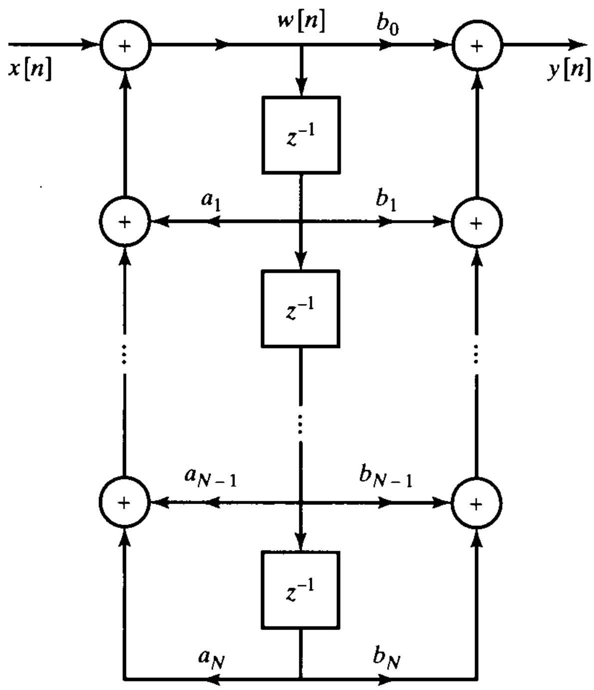


Figure 23.3 Direct Form 2 design by combining delays in figure 23.2

In direct form 2, we reduced the number of delays used to MAX(M, N).

23.2.3 Cascade Form

Another approach to implement rational transfer function is cascade form. As it is named, the way to implement it is to break the entire rational transfer function into couple small transfer function with less or equal to 2 poles / zeros in each of the small transfer function. It may has 2 poles / zeros if the roots of the poles / zeros are in complex conjugate pairs. Let's say that the transfer function we have only has real roots, then we may re-write the transfer function into this form:

$$H(Z) = \alpha \frac{\Pi_{k=0} 1 - b_k Z^{-1}}{\Pi_{k=0} 1 - a_k Z^{-1}} = \alpha \Pi_{k=0} \frac{1 - b_k Z^{-1}}{1 - a_k Z^{-1}} \tag{23.3}$$

Then we are able to build the filter using Cascade form.

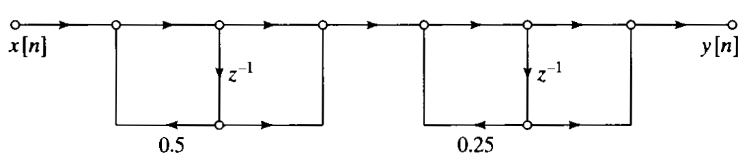


Figure 23.4 Parallel form illustration with each stage implemented using Direct Form 2.

Questions arise when we have multi-stage cascade system:

1. Choice of ordering stages.
2. Choice of paqiring pole and zero.

As we will discuss this issue when talking about Finite Precision and Quantization Effect later.

23.2.4 Parallel Form

The idea of Parallel Form comes from Partial Fraction Expansion. We may write our transfer function in the following form:

$$H(Z) = \sum_{k=1} \frac{C_k}{1 - Z_k^p Z^{-1}} \tag{23.4}$$

And we are able to implement the filter in terms of several transfer function in parallel.

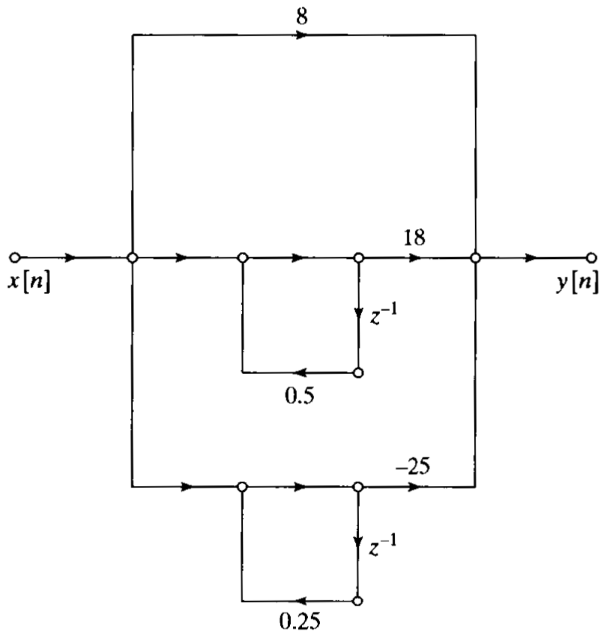
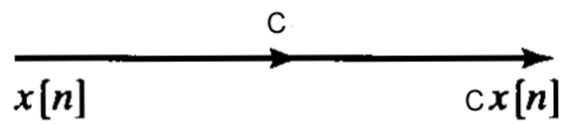


Figure 23.5 Parallel form illustration with several first order system in parallel.

23.3 Finite Precision Overview

When designing the system we want on paper, we can always have each component we desired. However, in reality, the precision is limited by hardware implementation or some other factor. Therefore, we need to take quantization effect into account.

For example, we have a system that has constant gain of C. We may model the quantization effect as a ***white noise*** e[n] added to the system:



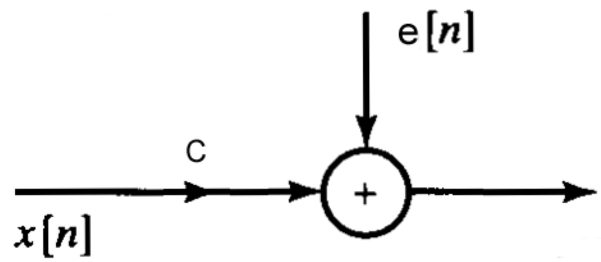


Figure 23.6 Top: System with constant gain C
Buttom: Same system with quantization effect

Where e[n] has the following properties:
1) $|e[n]| \leq 2^{-b}$
2) e[n] is "white", which means it has constant
power spectrum density (PSD) thoughout all frequency.
3) Each e[n] is independent to each other.

23.4 Next Lecture

In the next lecture, we are going to talk about Quantization effect in detail, and how error effects the system in different implementation (DF1, DF2, Cascade, Parallel).