FilterStructure

The `FilterStructure` property values are specified as one of the following strings indicating the quantized filter architecture:

'

FilterStructure Property Name	Filter Description
`'antisymmetricfir'`	Antisymmetric finite impulse response (FIR) Even and odd forms.
`'df1'`	Direct form I.
`'df1t'`	Direct form I transposed
`'df2'`	Direct form II.
`'df2t'`	Direct form II transposed. Default filter structure.
`'fir'`	Direct form FIR.
`'firt'`	Direct form FIR transposed
`'latcallpass'`	Lattice allpass
`'latticeca'`	Lattice coupled-allpass
`'latticecapc'`	Lattice coupled-allpass power-complementary
`'latticear'`	Lattice autoregressive (AR)
`'latticema'`	Lattice moving average (MA) minimum phase
`'latcmax'`	Lattice moving average (MA) maximum phase
`'latticearma'`	Lattice ARMA
`'statespace'`	Single-input/single-output state-space
`'symmetricfir'`	Symmetric FIR. Even and odd forms.

Default value: `'df2t'`

Remarks: The syntax for entering values for the `ReferenceCoefficients` property is constrained by the `FilterStructure` property value. See Table 12-4: Syntax for Assigning Reference Filter Coefficients (Single Section), for information on how to enter these coefficients for each filter architecture.

Filter Structure with Quantizers in Place

To help you understand how the quantizers work in filter structures like those provided in the Toolbox, Figure 12-1 presents the structure for a Direct Form 2 filter, including the quantizers that compose the quantized filter. You see that one o more quantizers accompany each filter element, such as a delay, coefficient, or a summation element. The input to or output from each element reflects the result of the associated quantizer. Wherever a particular filter element appears in a structure, recall the quantizers that accompany it as they appear in this figure. For example, a multiplicand quantizer precedes every coefficient element and a produc quantizer follows every coefficient element. Or a sum quantizer follows each sum element.

Notice that in this diagram, the first denominator coefficient in your filter, 1/a(1), appears because a(1) is not equal to 1.

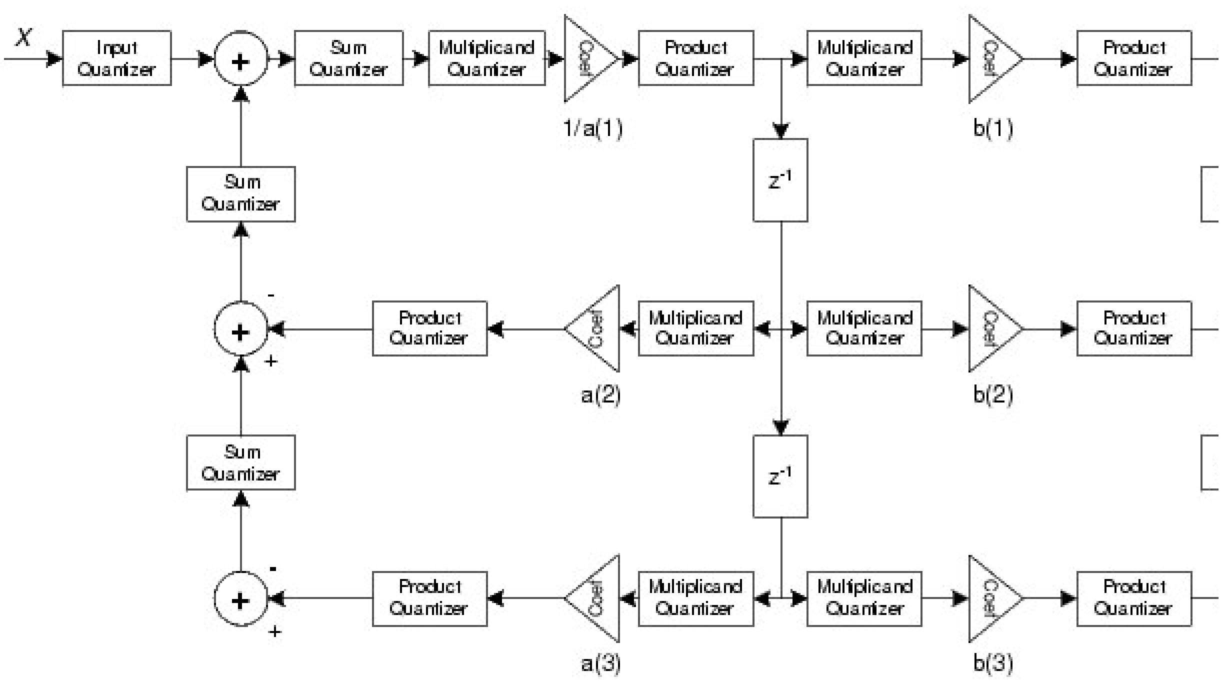


Figure 12-1: df2 Filter Structure Including the Quantizers, with a(1) ≠ 1

When your filter sets a(1) = 1, the df2 structure changes as shown in the next diagram, where the multiplicand and product quantizers for a(1) are not included and are not used when you quantize your filter. Skipping these quantizers removes potential errors that arise when a(1) ends up not quite equal to 1 after quantization, although it should be exactly 1.

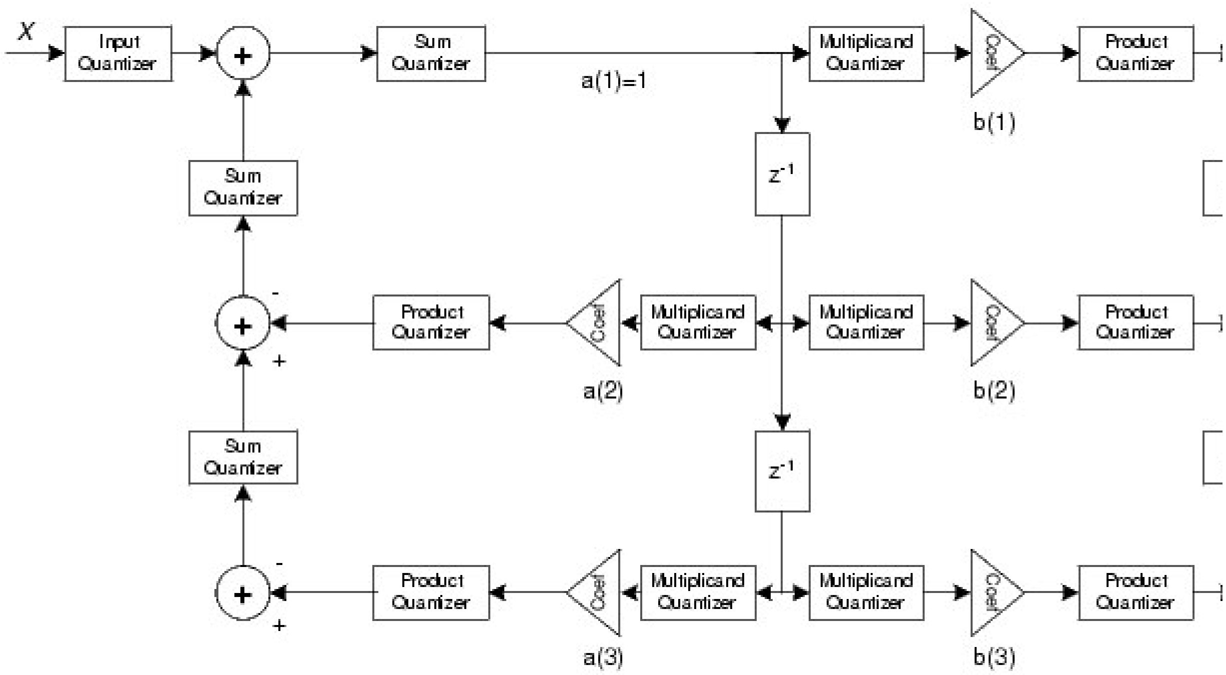


Figure 12-2: df2 Filter Structure Without Input Quantizers, where a(1) = 1

When the leading denominator coefficient a(1) is not 1, choose it to be a power of two so that a shift replaces the multiply that would otherwise be used.

> **Note** The quantized filter structures in the toolbox include the first denominator coefficient a(1) in the feedback loop of direct-form IIR filters (df1, df1t, df2, df2t), although customarily a(1) = 0.
>
> However, when a(1)≠ 1, the coefficient is needed to ensure accurate quantization analysis. For examples of instances where the leading denominator coefficient is not 1, check references[7] and [10] in the Bibliography.

Quantized Filter Structures

You can choose among several different filter structures when you create a quantized filter. You can also specify filters with single or multiple cascaded sections of the same type. Because quantization is a nonlinear process, different filter structures produce different results.

You specify the filter structure by assigning a specific string to the `FilterStructure` property. Refer to the function reference listings for `qfilt` and `set` for information on setting property values.

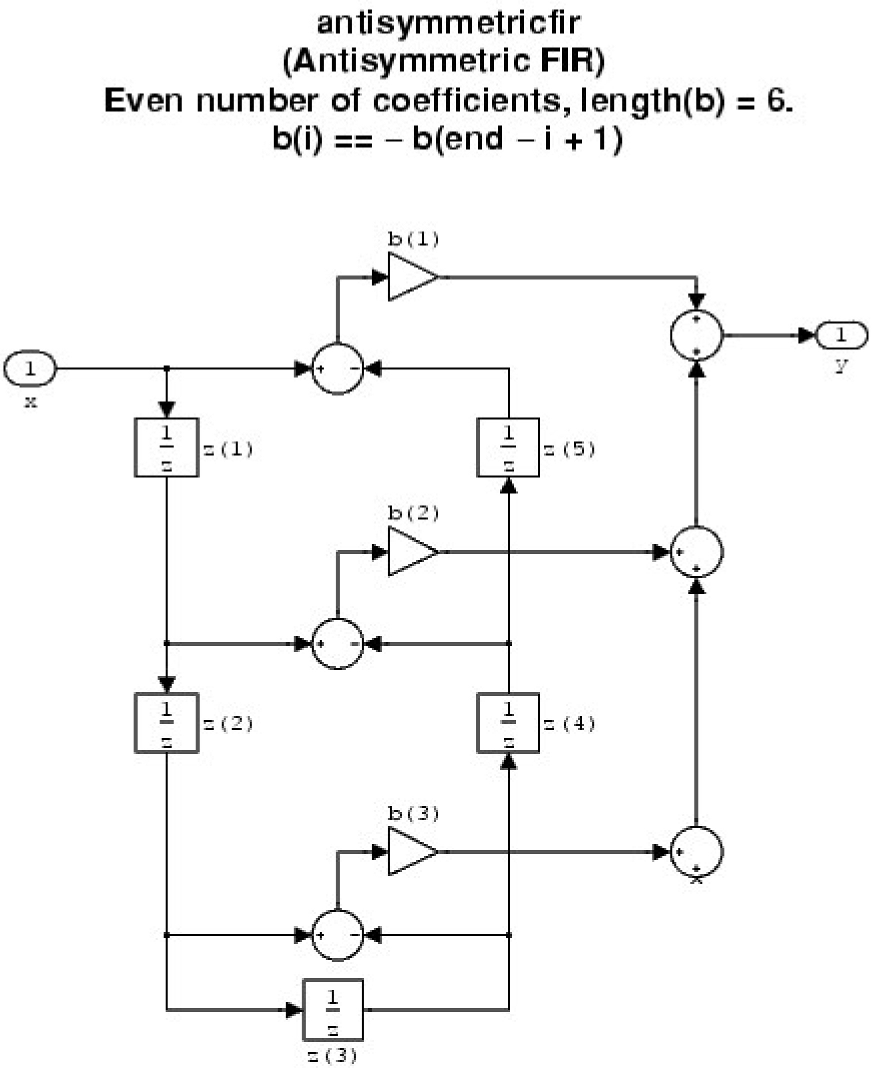
The `FilterStructure` property value constrains the syntax you can use for specifying the filter reference coefficients. For details on the syntax to use for specifying a filter with either a single section, or multiple (L) cascaded sections, see Table 12-4, Syntax for Assigning Reference Filter Coefficients (Single Section), and Table 12-5, Syntax for Assigning Reference Filter Coefficients (L Sections).

The figures in the following subsections of this section serve as visual aids to help you determine how to enter the reference filter coefficients for each filter structure. Each subsection contains a simple example for constructing a filter of a given structure.

Scale factors for the inputs and output for the filters do not appear in the block diagrams. The default filter structures do not include, nor assume, the scale factors

Direct Form Antisymmetric FIR Filter Structure (Odd Order)

The following figure depicts a *direct form antisymmetric FIR* filter structure that directly realizes a fifth-order antisymmetric FIR filter. The filter coefficients are labeled $b(i)$, $i = 1, ..., 6$, and the initial and final state values in filtering are labeled $z(i)$.



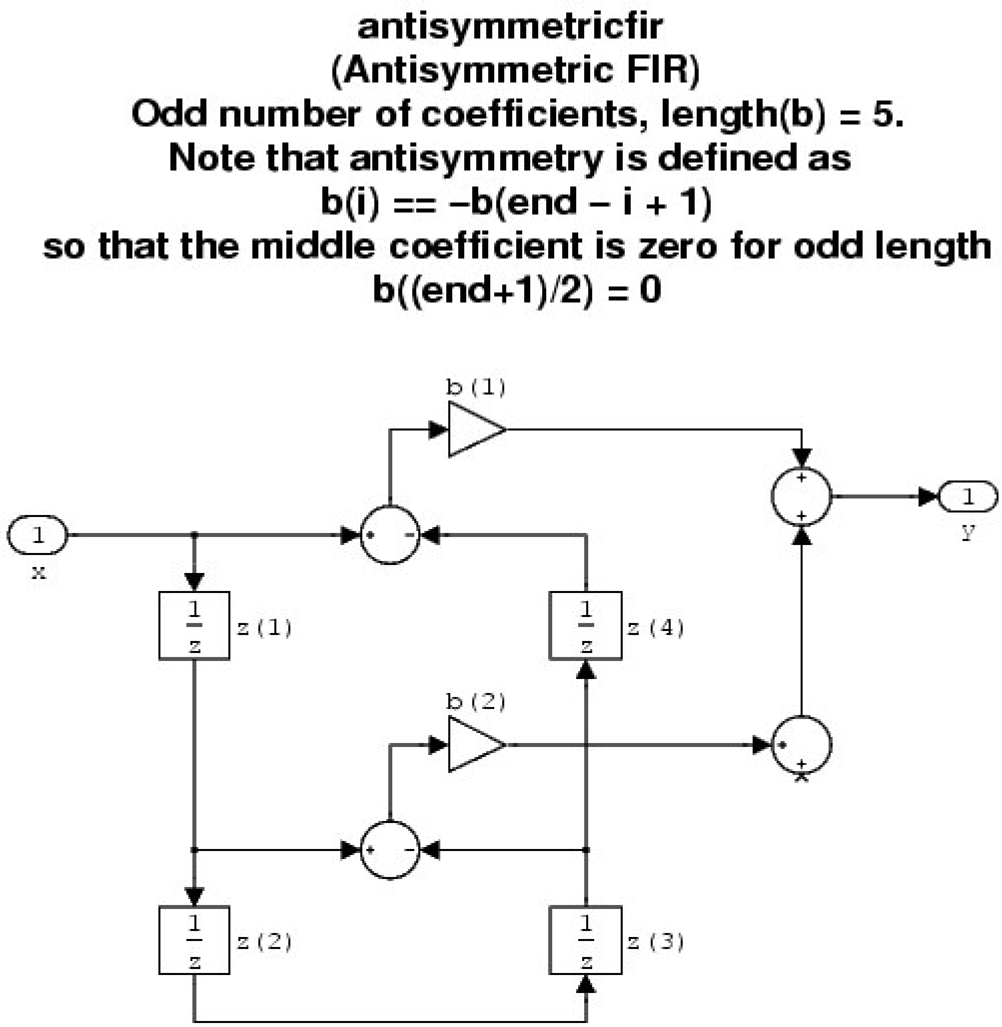
Use the string `'antisymmetricfir'` for the value of the `FilterStructure` property to design a quantized filter with this structure.

Example -- Specifying an Odd-Order Direct Form Antisymmetric FIR Filter Structure. Specify a fifth-order direct form antisymmetric FIR filter structure for a quantized filter `Hq` with the following code.

```
b = [-0.008 0.06 -0.44 0.44 -0.06 0.008];
Hq = qfilt('antisymmetricfir',{b});
```

Antisymmetric FIR Filter Structure (Even Order)

The following figure depicts a *direct form antisymmetric FIR* filter structure that directly realizes a fourth-order antisymmetric FIR filter. The filter coefficients are labeled $b(i)$, $i = 1, ..., 5$, and the states (used for initial and final state values in filtering) are labeled $z(i)$.



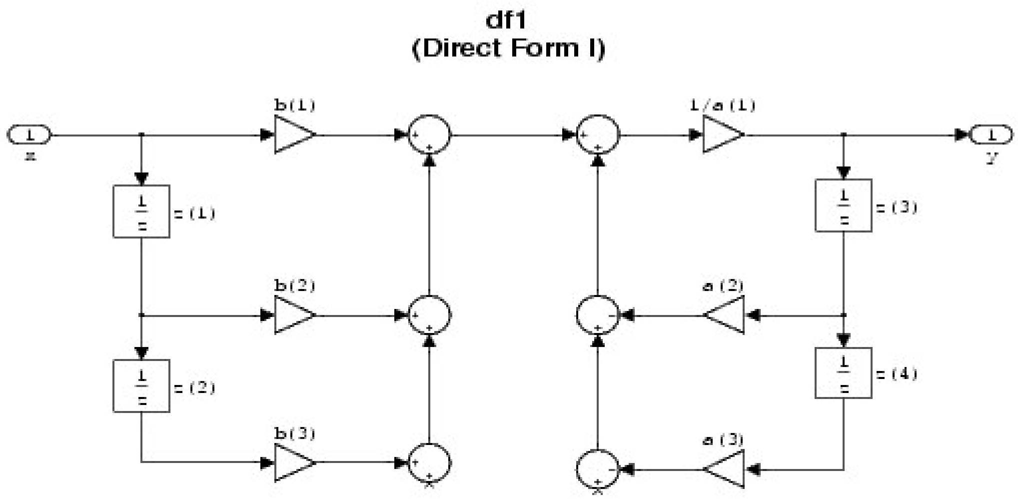
Use the string `'antisymmetricfir'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

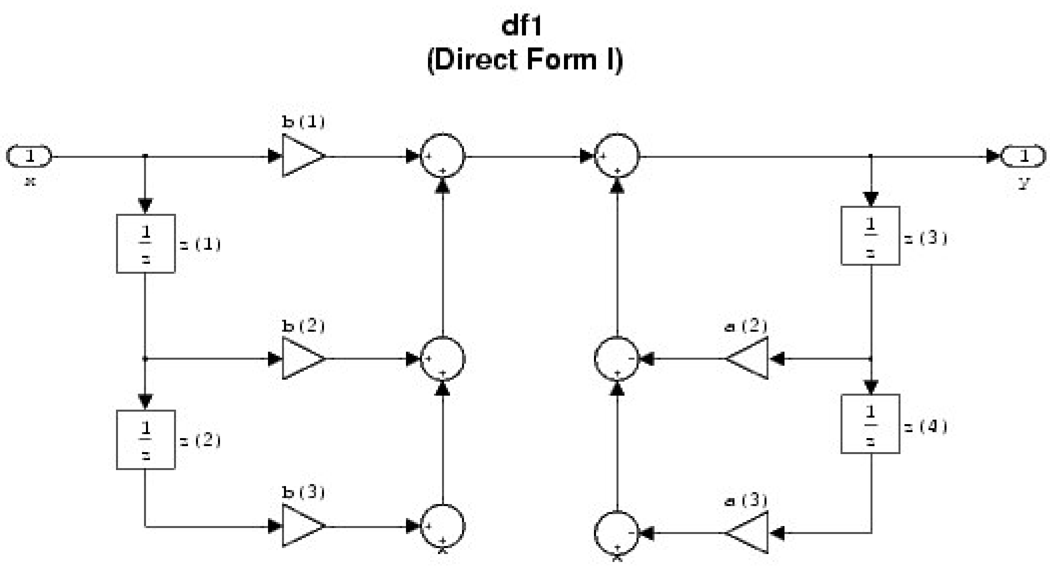
Example -- Specifying an Even-Order Direct Form Antisymmetric FIR Filter Structure. You can specify a fourth-order direct form antisymmetric FIR filter structure for a quantized filter `Hq` with the following code.

```
b = [-0.01 0.1 0.0 -0.1 0.01];
Hq = qfilt('antisymmetricfir',{b});
```

Direct Form I Filter Structure

The following figures depict *direct form I* filter structures that directly realize a transfer function with a second-order numerator and denominator. The numerator coefficients are labeled *b*(*i*), the denominator coefficients are labeled *a*(*i*), *i* = 1, 2, 3, and the states (used for initial and final state values in filtering) are labeled *z*(*i*). In the first figure, *a(1)* is not equal to one and appears in the structure. When *a(1)* is equal to one, the realized structure does not include the coefficient, as you see in the second figure.





Use the string `'df1'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

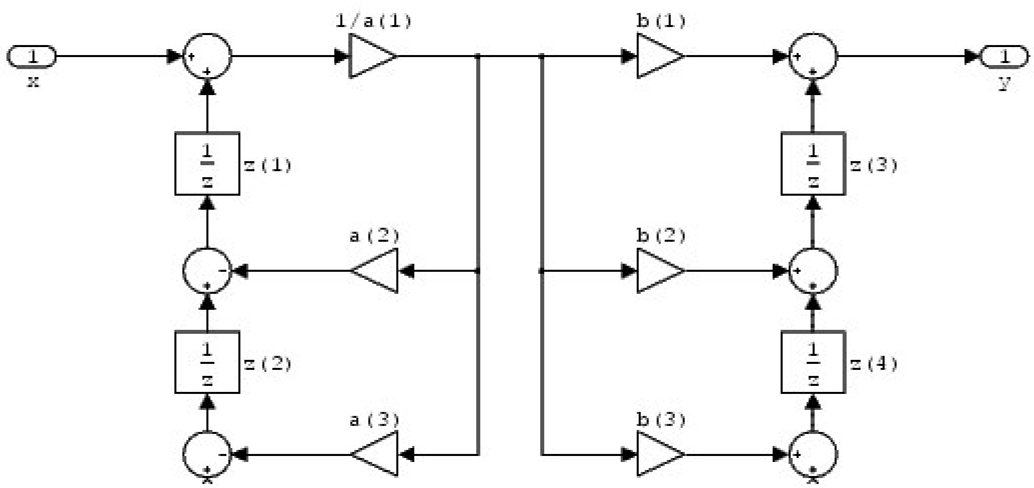
Example -- Specifying a Direct Form I Filter Structure. You can specify a second-order direct form I structure for a quantized filter `Hq` with the following code.

```
b = [0.3 0.6 0.3];
a = [1 0 0.2];
Hq = qfilt('df1',{b,a});
```

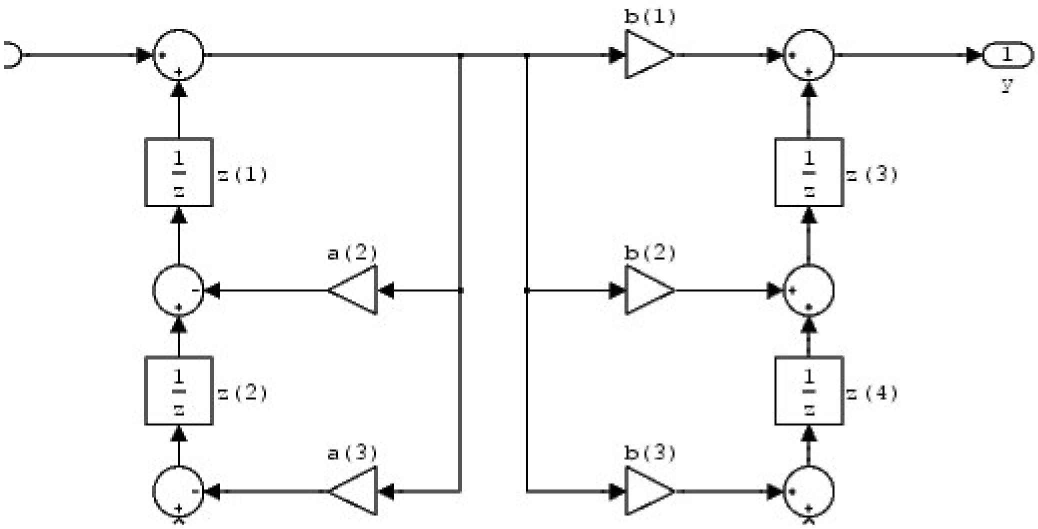
Direct Form I Transposed Filter Structure

The following figures depict *direct form I transposed* filter structures that directly realize a transfer function with a second-order numerator and denominator. The numerator coefficients are labeled *b*(*i*), the denominator coefficients are labeled *a*(*i*), *i* = 1, 2, 3, and the states (used for initial and final state values in filtering) are labeled *z*(*i*). In the first figure, *a(1)* is not equal to one and appears in the structure. When *a(1)* is equal to one, the realized structure does not include the coefficient, as you see in the second figure.

df1t
(Transposed Direct Form I)



df1t
(Transposed Direct Form I)



Use the string `'df1t'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

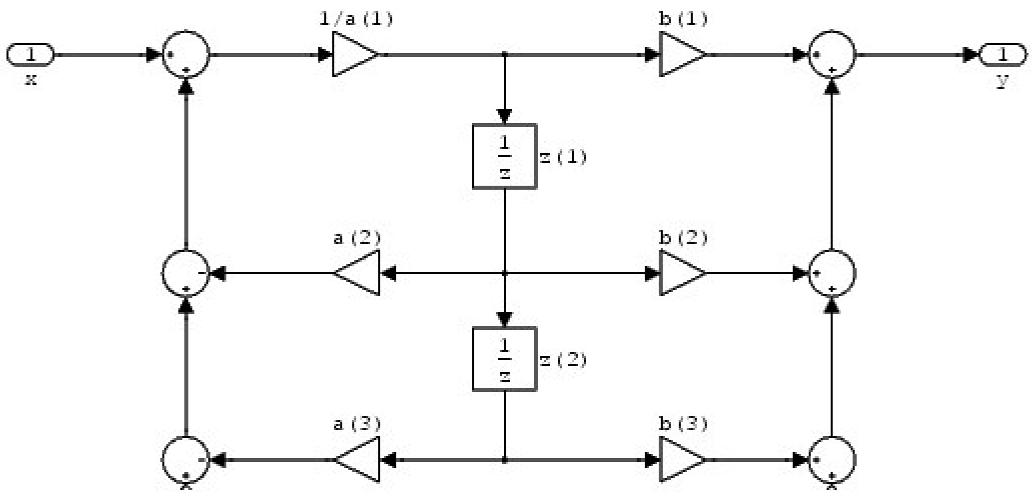
Example -- Specifying a Direct Form I Transposed Filter Structure. You can specify a second-order direct form I transposed filter structure for a quantized filter `Hq` with the following code.

```
b = [0.3 0.6 0.3];
a = [1 0 0.2];
Hq = qfilt('df1t',{b,a});
```

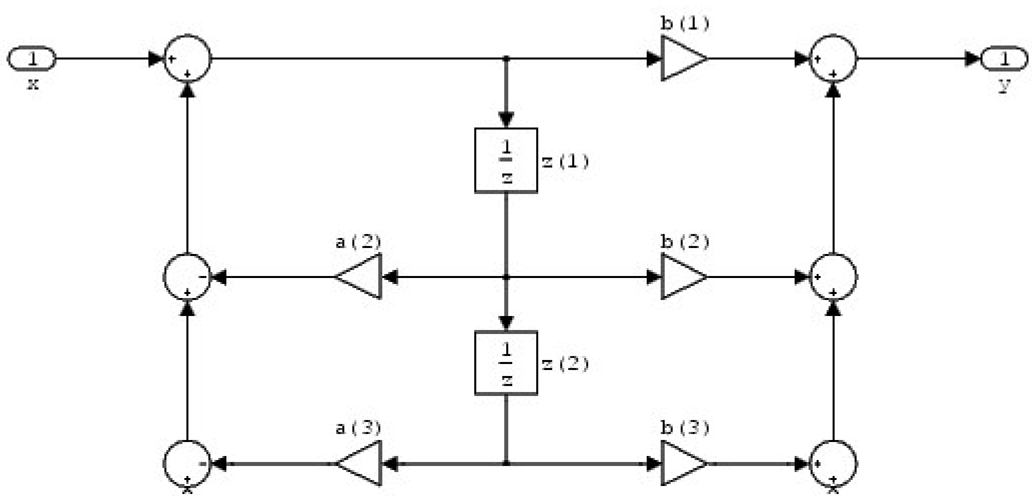
Direct Form II Filter Structure

The following figures depict *direct form II* filter structures that directly realize a transfer function with a second-order numerator and denominator. The numerator coefficients are labeled $b(i)$, the denominator coefficients are labeled $a(i)$, $i = 1, 2, 3$, and the states (used for initial and final state values in filtering) are labeled $z(i)$. In the first figure, *a(1)* is not equal to one and appears in the structure. When *a(1)* is equal to one, the realized structure does not include the coefficient, as you see in the second figure.

df2
(Direct Form II)



df2
(Direct Form II)



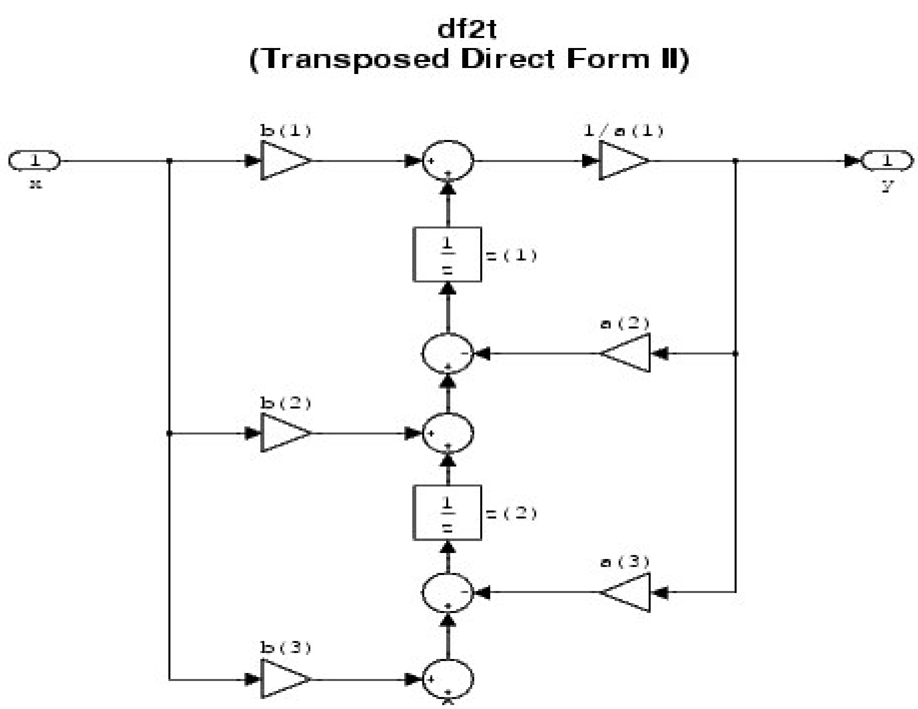
Use the string `'df2'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

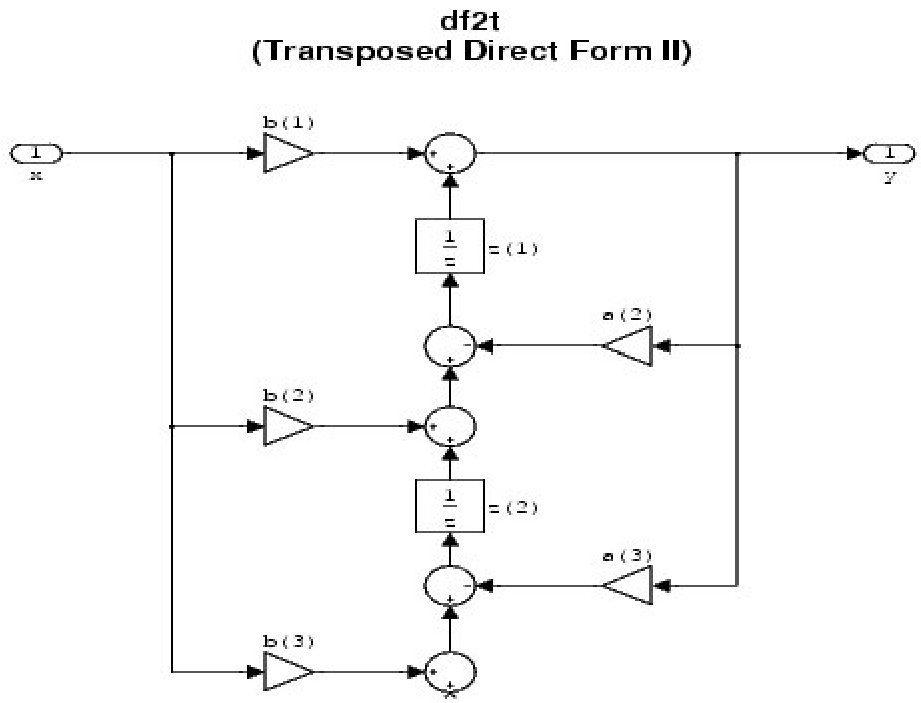
Example -- Specifying a Direct Form II Filter Structure. You can specify a second-order direct form II filter structure for a quantized filter `Hq` with the following code.

```
b = [0.3 0.6 0.3];
a = [1 0 0.2];
Hq = qfilt('df2',{b,a});
```

Direct Form II Transposed Filter Structure

The following figures depict *direct form II transposed* filter structures that directly realize a transfer function with a second-order numerator and denominator. The numerator coefficients are labeled *b*(*i*), the denominator coefficients are labeled *a*(*i*), *i* = 1, 2, 3, and the states (used for initial and final state values in filtering) are labeled *z*(*i*). In the first figure, *a(1)* is not equal to one and appears in the structure. When *a(1)* is equal to one, the realized structure does not include the coefficient, as you see in the second figure.





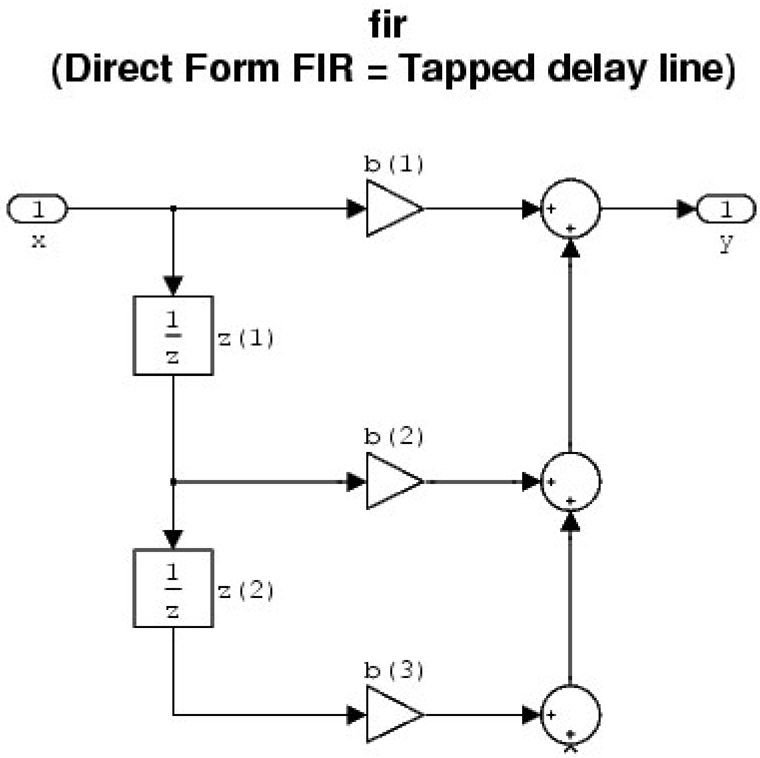
Use the string `'df2t'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

Example -- Specifying a Direct Form II Transposed Filter Structure. You can specify a second-order direct form II transposed filter structure for a quantized filter `Hq` with the following code.

```
b = [0.3 0.6 0.3];
a = [1 0 0.2];
Hq = qfilt('df2t',{b,a});
```

Direct Form Finite Impulse Response (FIR) Filter Structure

The following figure depicts a *direct form finite impulse response (FIR)* filter structure that directly realizes a second-order FIR filter. The filter coefficients are labeled *b*(*i*), *i* = 1, 2, 3, and the states (used for initial and final state values in filtering) are labeled *z*(*i*).



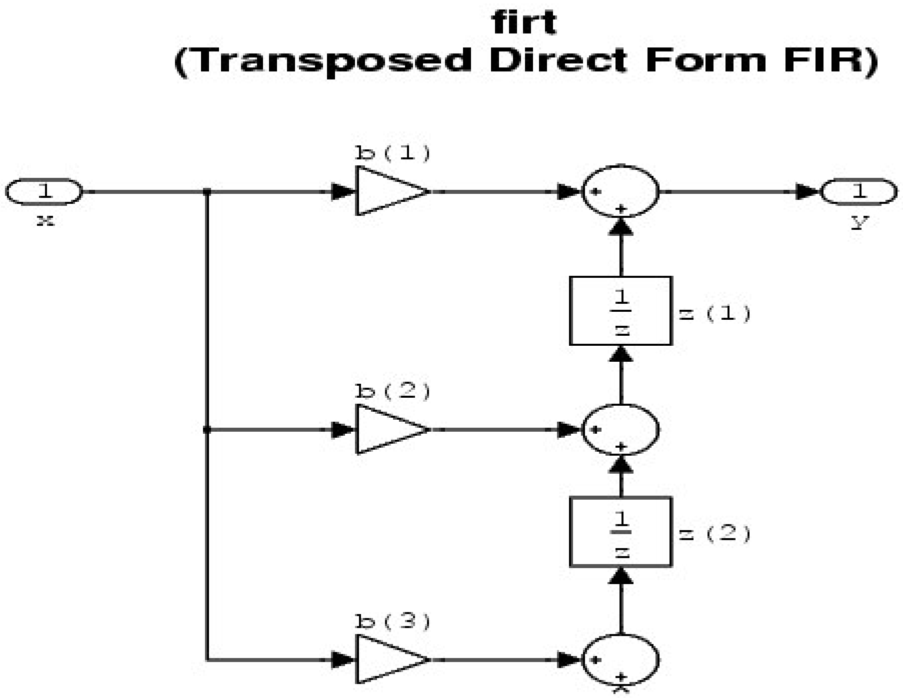
Use the string `'fir'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

Example -- Specifying a Direct Form FIR Filter Structure. You can specify a second-order direct form FIR filter structure for a quantized filter `Hq` with the following code.

```
b = [0.05 0.9 0.05];
Hq = qfilt('fir',{b});
```

Direct Form FIR Transposed Filter Structure

The following figure depicts a *direct form finite impulse response (FIR) transposed* filter structure that directly realizes a second-order FIR filter. The filter coefficients are labeled *b*(*i*), *i* = 1, 2, 3, and the states (used for initial and final state values in filtering) are labeled *z*(*i*).



Use the string `'firt'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

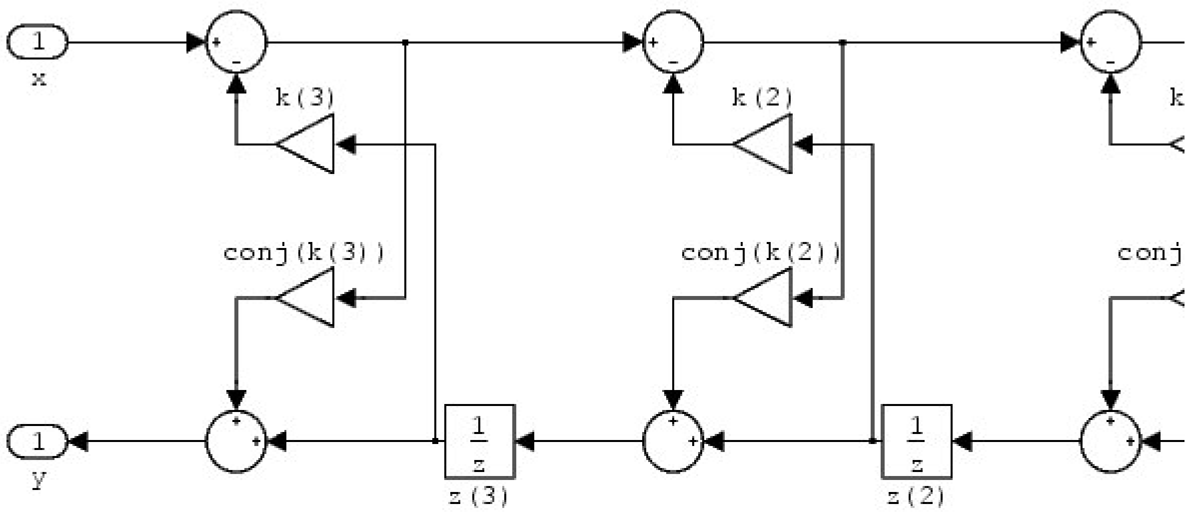
Example -- Specifying a Direct Forn FIR Transposed Filter Structure. You can specify a second-order direct form FIR transposed filter structure for a quantized filter `Hq` with the following code.

```
b = [0.05 0.9 0.05];
Hq = qfilt('firt',{b});
```

Lattice Allpass Filter Structure

The following figure depicts a *lattice allpass* filter structure. The pictured structure directly realizes third-order lattice allpass filters. The filter reflection coefficients are labeled *k1*(*i*), *i* = 1, 2, 3. The states (used for initial and final state values in filtering) are labeled *z*(*i*).

latcallpass
(Lattice AR All−Pass)



Use the string `'latcallpass'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

Example -- Specifying a Lattice Allpass Filter Structure. You can specify a third-order lattice allpass filter structure for a quantized filter `Hq` with the following code.

```
k = [.66 .7 .44];
Hq = qfilt('latcallpass',{k});
```

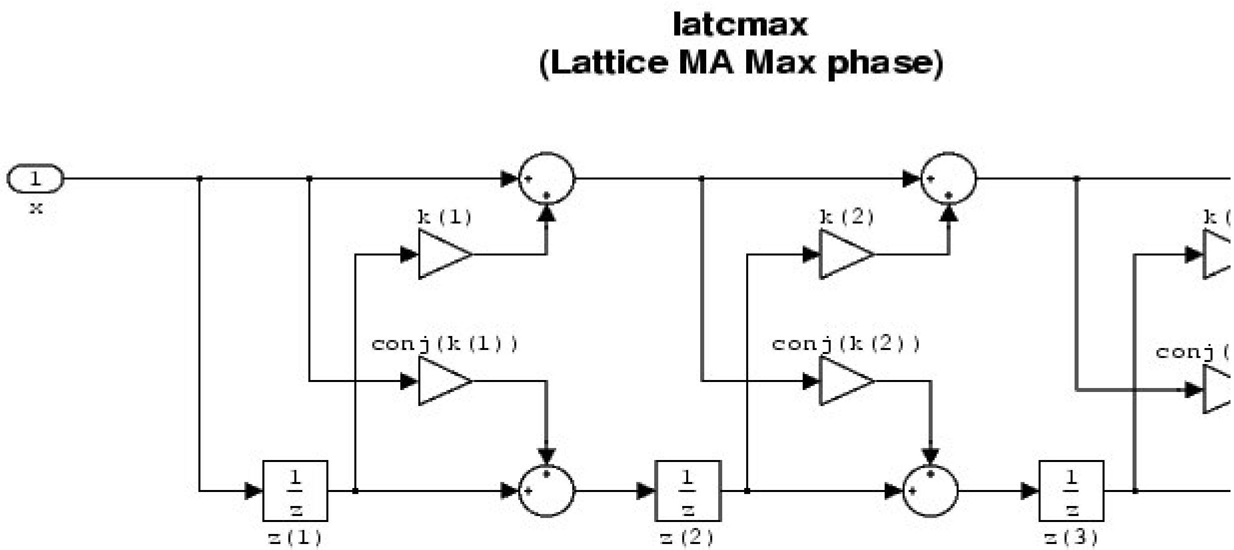
Lattice Moving Average Maximum Phase Filter Structure

The following figure depicts a *lattice moving average maximum phase* filter structure that directly realizes a third-order lattice moving average (MA) filter with the following phase form depending on the initial transfer function:

- When you start with a minimum phase transfer function, the upper branch of the resulting lattice structure returns a minimum phase filter. The lower branch returns a maximum phase filter.
- When your transfer function is neither minimum phase nor maximum phase, the lattice moving average maximum phase structure will not be maximum phase.

- When you start with a maximum phase filter, the resulting lattice filter is maximum phase also.

The filter reflection coefficients are labeled *k*(*i*), *i* = 1, 2, 3. The states (used for initial and final state values in filtering) are labeled *z*(*i*).



Use the string `'latcmax'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

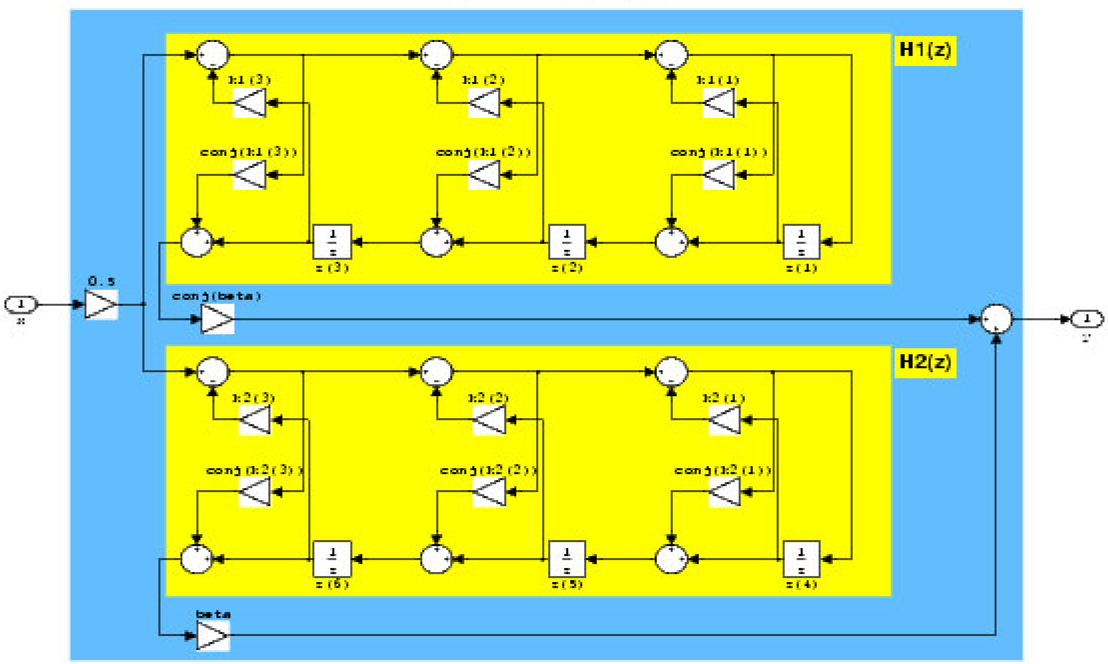
Example--Specifying a Lattice Moving Average Maximum Phase Filter Structure. You can specify a fourth-order lattice MA maximum phase filter structure for a quantized filter `Hq` with the following code.

```
k = [.66 .7 .44 .33];
Hq = qfilt('latcmax',{k});
```

Lattice Coupled-Allpass Filter Structure

The following figure depicts a *lattice coupled-allpass* filter structure. The filter is composed of two third-order allpass lattice filters. The filter reflection coefficients for the first filter are labeled *k1*(*i*), *i* = 1, 2, 3. The filter reflection coefficients for the second filter are labeled *k2*(*i*), *i* = 1, 2, 3. The unity gain complex coupling coefficient is *beta*. The states (used for initial and final state values in filtering) are labeled *z*(*i*).

latticeca
(Coupled Allpass Lattice)



Use the string `'latticeca'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

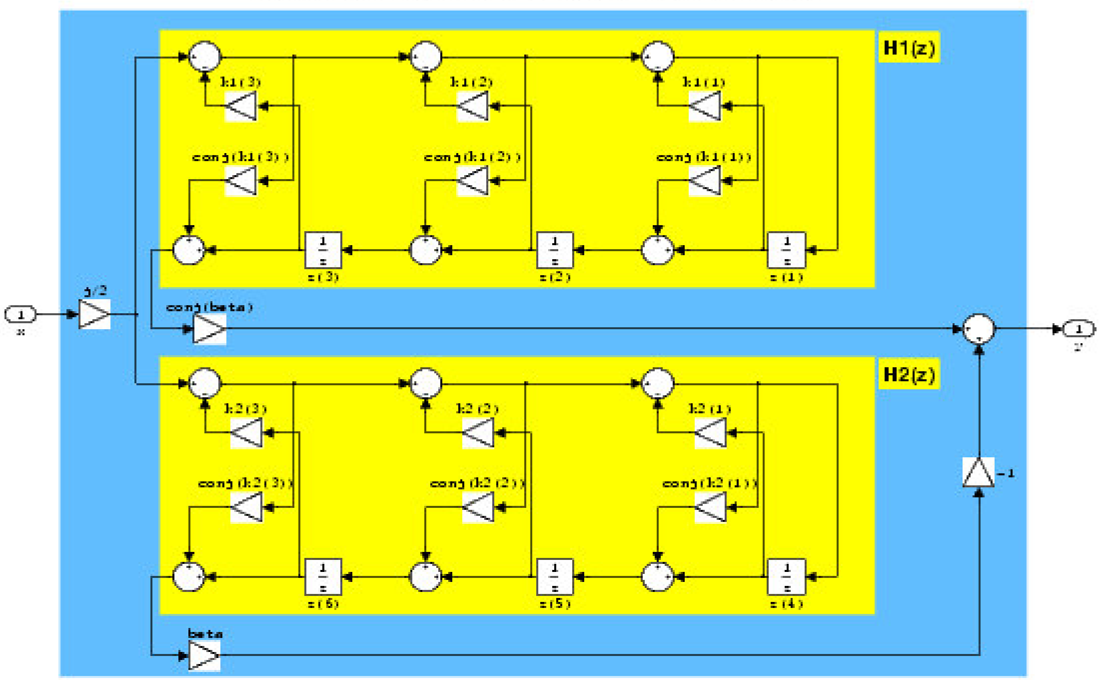
Example -- Specifying a Lattice Coupled-Allpass Filter Structure. You can specify a third-order lattice coupled allpass filter structure for a quantized filter `Hq` with the following code.

```
k1 = [0.9511 + 0.3088i; 0.7511 + 0.1158i]
k2 = 0.7502 - 0.1218i
beta = 0.1385 + 0.9904i
Hq = qfilt('latticeca',{k1,k2,beta});
```

Lattice Coupled-Allpass Power Complementary Filter Structure

The following figure depicts a *lattice coupled-allpass power complementary* filter structure. The filter is composed of two third-order allpass lattice filters. The filter reflection coefficients for the first filter are labeled *k1*(*i*), *i* = 1, 2, 3. The filter reflection coefficients for the second filter are labeled *k2*(*i*), *i* = 1, 2, 3. The unity gain complex coupling coefficient is *beta*. The states used for initial and final state values in filtering are labeled *z*(*i*). The resulting filter transfer function is the power-complementary transfer function of the coupled allpass lattice filter (formed from the same coefficients).

latticecapc
(Coupled Allpass Lattice, Power Complementary output)



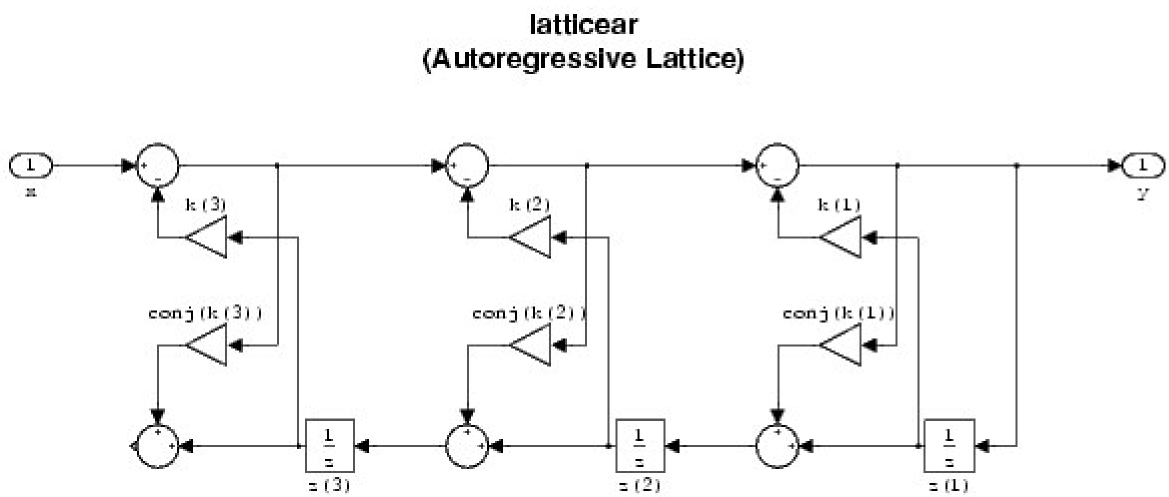
Use the string `'latticecapc'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

Example -- Specifying a Lattice Coupled-Allpass Power Complementary Filter Structure. Specify a third-order lattice coupled-allpass power complementary filter structure for a quantized filter `Hq` with the following code.

```
k1 = [0.9511 + 0.3088i; 0.7511 + 0.1158i]
k2 = 0.7502 - 0.1218i
beta = 0.1385 + 0.9904i
Hq = qfilt('latticecapc',{k1,k2,beta});
```

Lattice Autoregressive (AR) Filter Structure

The following figure depicts a *lattice autoregressive (AR)* filter structure that directly realizes a third-order lattice AR filter. The filter reflection coefficients are labeled *k*(*i*), *i* = 1, 2, 3, and the states (used for initial and final state values in filtering) are labeled *z*(*i*).



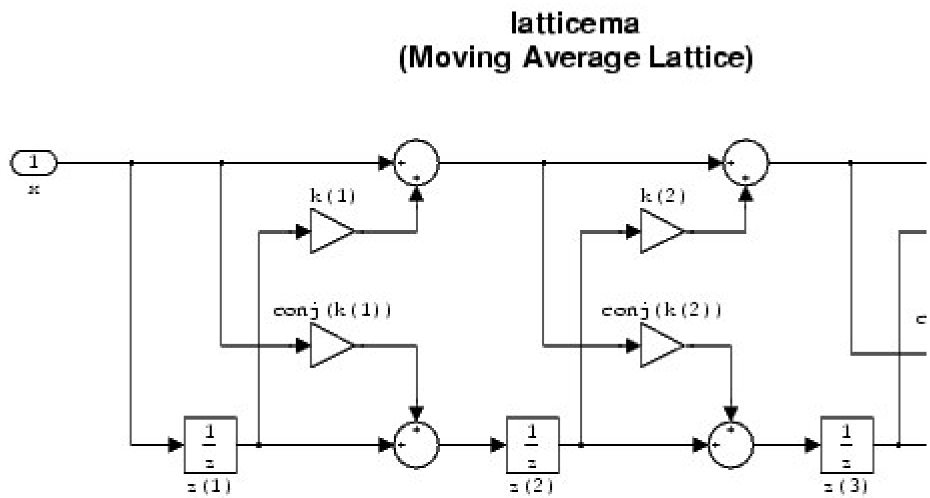
Use the string `'latticear'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

Example -- Specifying an Lattice AR Filter Structure. You can specify a third-order lattice AR filter structure for a quantized filter `Hq` with the following code.

```
k = [.66 .7 .44];
Hq = qfilt('latticear',{k});
```

Lattice Moving Average (MA) Filter Structure

The following figure depicts a *lattice moving average (MA)* filter structure that directly realizes a third-order lattice MA filter. The filter reflection coefficients are labeled *k*(*i*), *i* = 1, 2, 3, and the states (used for initial and final state values in filtering) are labeled *z*(*i*).



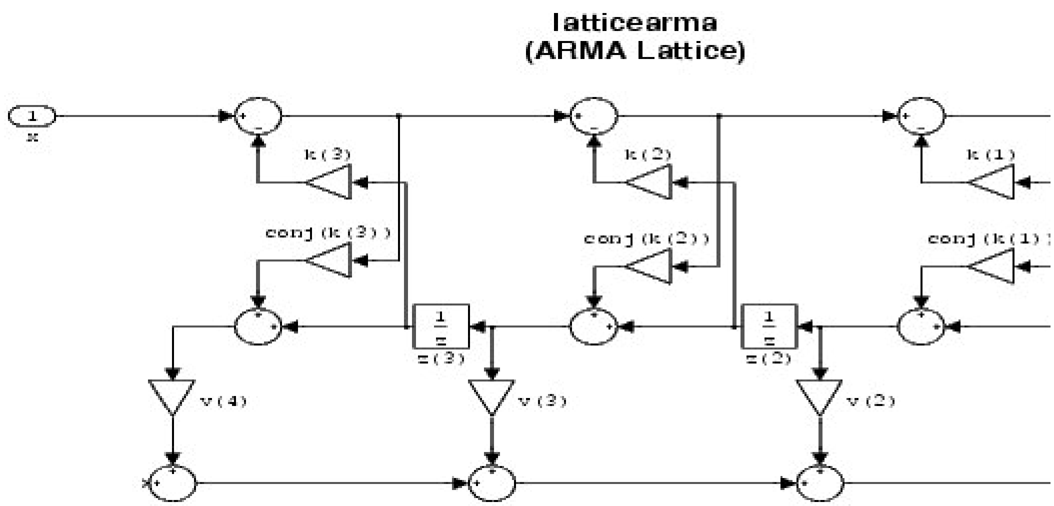
Use the string `'latticema'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

Example -- Specifying an Lattice MA Filter Structure. You can specify a third-order lattice MA filter structure for a quantized filter `Hq` with the following code.

```
k = [.66 .7 .44];
Hq = qfilt('latticema',{k});
```

Lattice Autoregressive Moving Average (ARMA) Filter Structure

The following figure depicts a *lattice autoregressive moving average (ARMA)* filter structure that directly realizes a fourth-order lattice ARMA filter. The filter reflection coefficients are labeled *k*(*i*), *i* = 1, ..., 4, the ladder coefficients are labeled *v*(*i*), *i* = 1, 2, 3, and the states (used for initial and final state values in filtering) are labeled *z*(*i*).



Use the string `'latticearma'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

Example -- Specifying an Lattice ARMA Filter Structure. You can specify a fourth-order lattice ARMA filter structure for a quantized filter `Hq` with the following code.

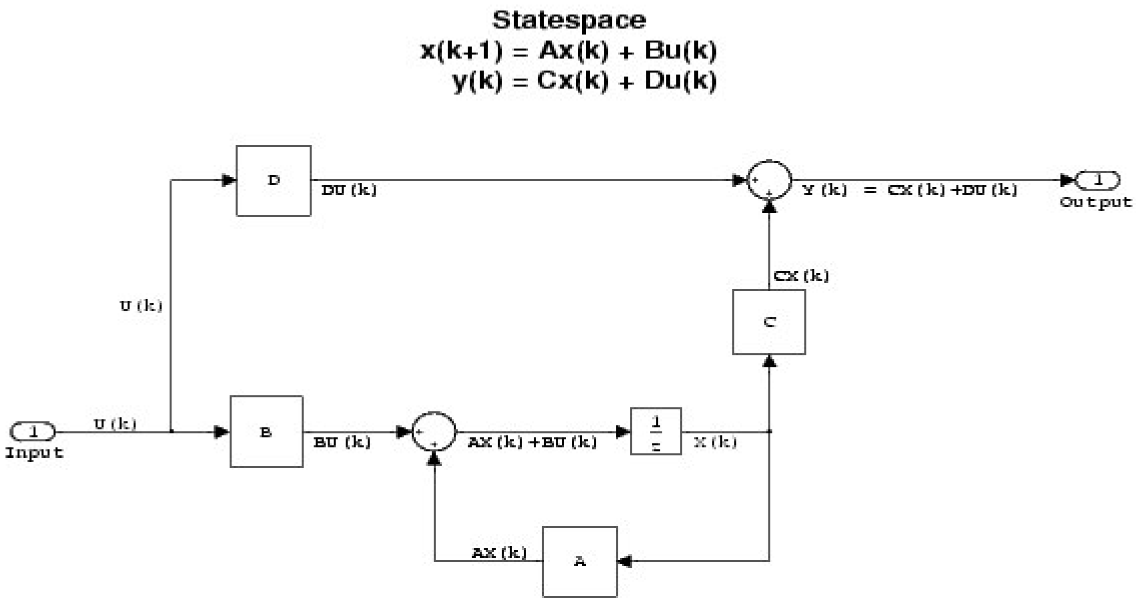
```
k = [.66 .7 .44 .66];
v = [1 0 0];
Hq = qfilt('latticearma',{k,v});
```

State-Space Filter Structure

State-space models with input sequence x_k and output sequence y_k have the following form.

$$z_{k+1} = Az_k + Bx_k$$
$$y_k = Cz_k + Dx_k$$

If the states z_k are vectors of length n, then the matrices A, B, C, and D are n-by-n, n-by-1, 1-by-n, and 1-by-1 respectively.



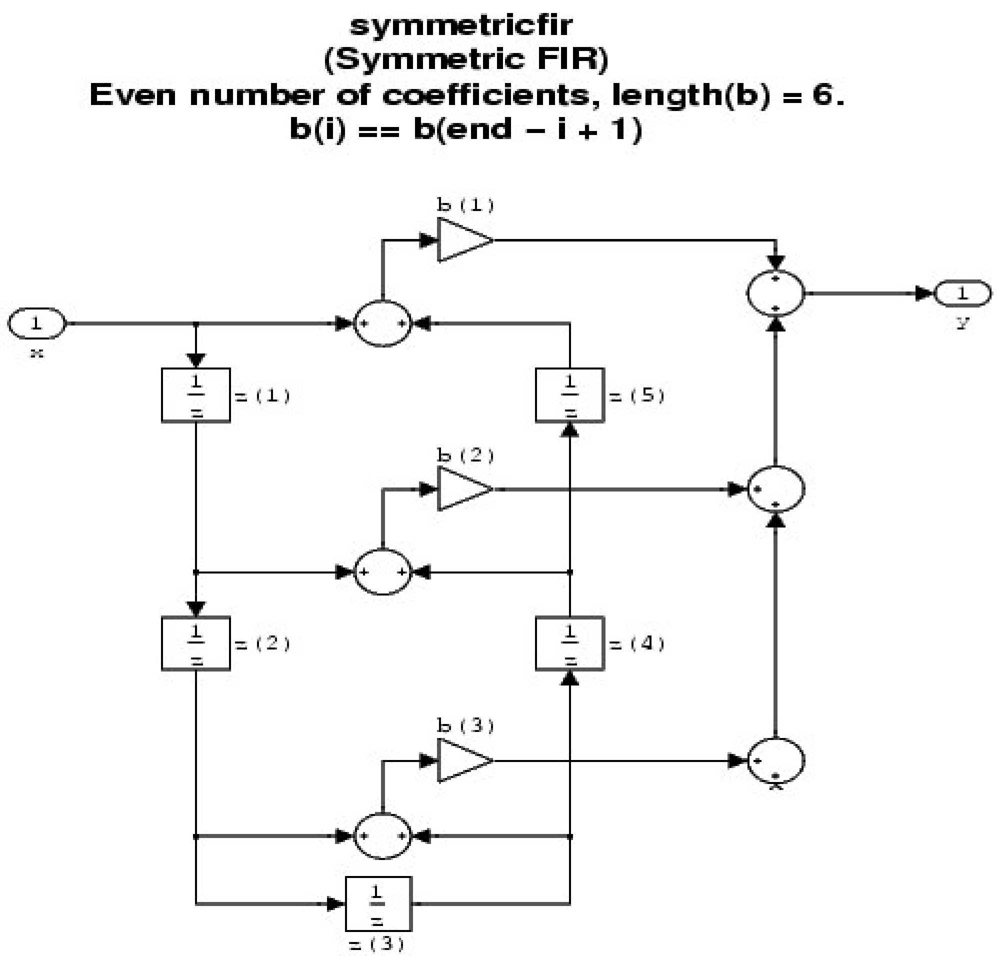
Use the string `'statespace'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

Example -- Specifying a State-Space Filter Structure. You can specify a second-order state-space filter structure for a quantized filter `Hq` with the following code.

```
[A,B,C,D] = butter(2,0.5);
Hq = qfilt('statespace',{A,B,C,D});
```

Direct Form Symmetric FIR Filter Structure (Odd Order)

The following figure depicts a *direct form symmetric FIR* filter structure that directly realizes a fifth-order direct form symmetric FIR filter. The filter coefficients are labeled *b*(*i*), *i* = 1, ..., 6, and the states (used for initial and final state values in filtering) are labeled *z*(*i*).



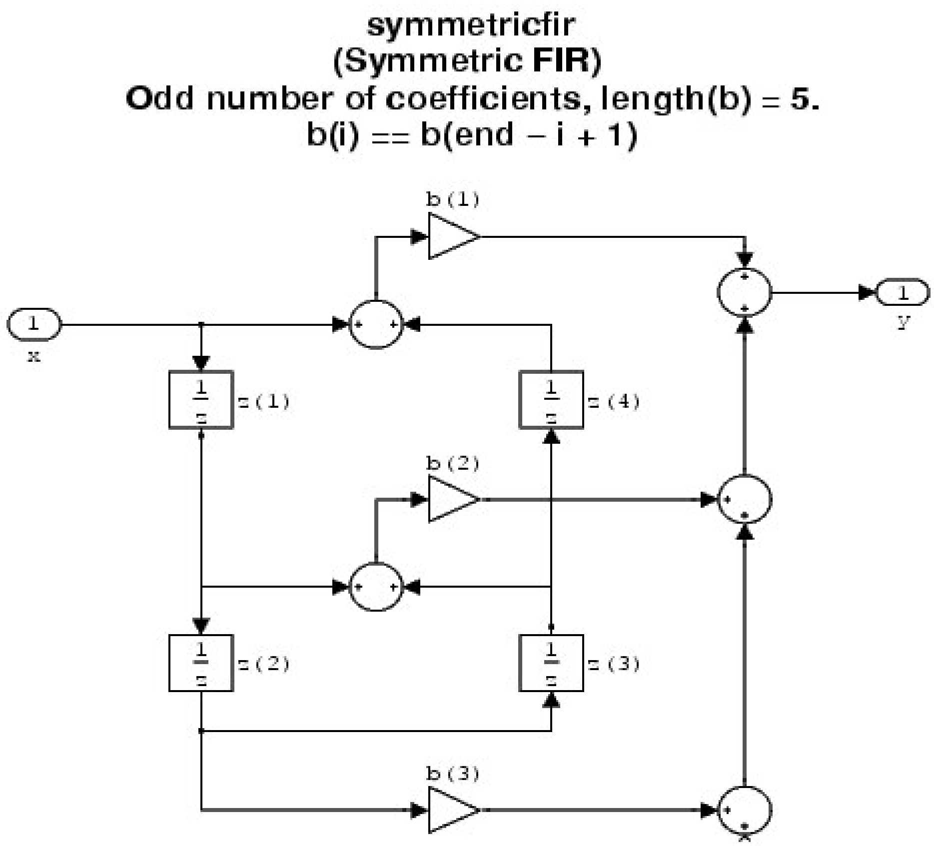
Use the string `'symmetricfir'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

Example -- Specifying an Odd-Order Direct Form Symmetric FIR Filter Structure. You can specify a fifth-order direct form symmetric FIR filter structure for a quantized filter `Hq` with the following code.

```
b = [-0.008 0.06 0.44 0.44 0.06 -0.008];
Hq = qfilt('symmetricfir',{b});
```

Direct Form Symmetric FIR Filter Structure (Even Order)

The following figure depicts a *direct form symmetric FIR* filter structure that directly realizes a fourth-order direct form symmetric FIR filter. The filter coefficients are labeled *b*(*i*), *i* = 1, ..., 5, and the states (used for initial and final state values in filtering) are labeled *z*(*i*).



Use the string `'symmetricfir'` to specify the value of the `FilterStructure` property for a quantized filter with this structure.

Example -- Specifying an Even-Order Direct Form Symmetric FIR Filter Structure. You can specify a fourth-order direct form symmetric FIR filter structure for
a quantized filter `Hq` with the following code.

```
b = [-0.01 0.1 0.8 0.1 -0.01];
Hq = qfilt('symmetricfir',{b});
```