LABORATORY 5
DIGITAL FILTERS STRUCTURES

Infinite Impulse Response filters structures

The following transfer function will be considered:

$$H(z)=\frac{\sum_{k=0}^{M} b_k z^{-k}}{1+\sum_{k=1}^{N} a_k z^{-k}}=\frac{B(z)}{A(z)} \tag{0.1}$$

In discrete time domain, the digital filter can be described also by the finite difference equation:

$$y[n]=\sum_{k=0}^{M} b_k x[n-k]-\sum_{k=1}^{N} a_k y[n-k] \tag{0.2}$$

The finite difference equation allows us to compute one output sample ($y[n]$) based on M samples from the input signal and N-1 previous samples from the output signal. The first part of this equation is non-recursive, while the second part introduces a recursive dependency. The entire set of operations: multiplications, additions, and delay units – can be graphically represented as a structure with adders, multipliers and delay cells.

5.1.1. The Direct Form

Starting from the finite difference equation (0.2), the operations that result in the current output sample can be represented as a direct form 1 (figure 5.1) or a direct form 2 (figure 5.2), simply by identifying the coefficients a_k and b_k.

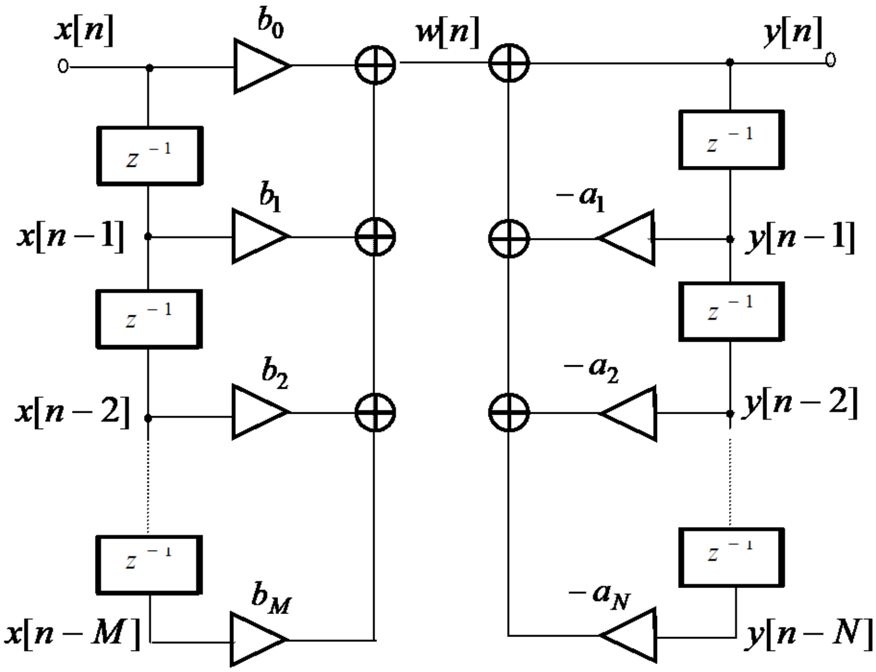

Figure 5.1. Direct form 1

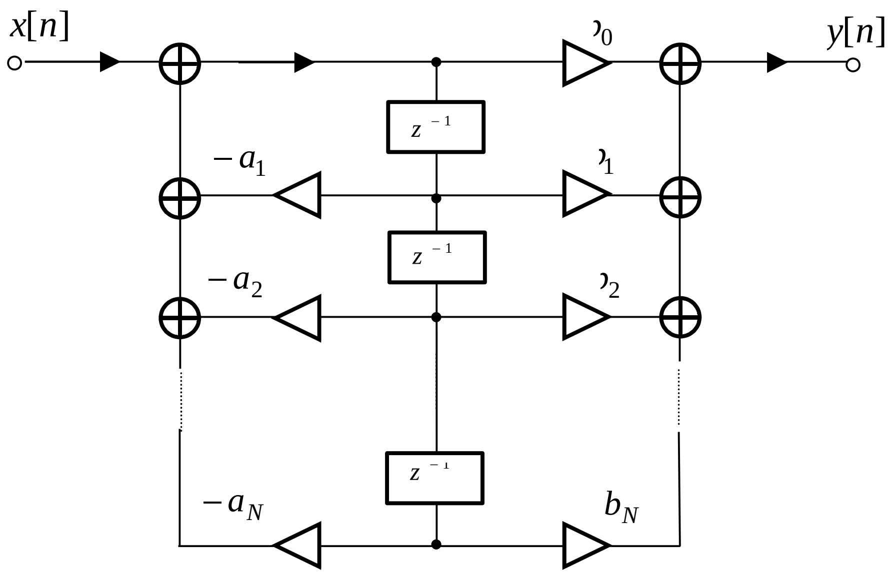

Figure 5.2. Direct form 2

5.1.2. The Cascade Form

The transfer function can be represented as product of several factors:

$$H(z)=\prod_{k=1}^{P}\frac{b_{0,k}(1-z_k z^{-1})(1-z_k^* z^{-1})}{(1-p_k z^{-1})(1-p_k^* z^{-1})}=\prod_{k=1}^{P}\frac{b_{0,k}+b_{1,k}z^{-1}+b_{2,k}z^{-2}}{1+a_{1,k}z^{-1}+a_{2,k}z^{-2}}=\prod_{k=1}^{P}H_k(z) \quad (0.3)$$

where z_k are the filter's zeros, and p_k are the poles of the transfer function, grouped in complex conjugated pairs, in order to attain the transfer functions of order 2 with real coefficients $H_k(z)$. If the transfer function has also real zeros or real poles, then the factorization into partial $H_k(z)$ can be made also with polynomials of order 1. The result is a cascade form represented in figure 5.3.

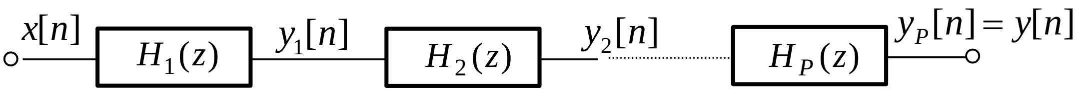

Figure 5.3. The cascade form

Each intermediary filter $H_k(z)$, having a maximum order of 2, can be put in a direct form 1 or in a direct form 2 structure.

The MATLAB functions `zp2sos`, and `tf2sos` allow to determine the 2nd order transfer functions (Second Order Sections) for the cascade form.

***Syntax*:**

`[SOS,G]=tf2sos(B,A)`

- it returns a `SOS` matrix which contains the coefficients of the 2nd order sections for the factorization of the transfer function $H(z)$. `B` represents the vector of the coefficients of the numerator $B(z)$ and `A` represents the vector of the coefficients of the denominator $A(z)$.
- SOS is a matrix of the following form:

```
SOS = [ b01 b11 b21  1 a11 a21
        b02 b12 b22  1 a12 a22
        ...
        b0L b1L b2L  1 a1L a2L ]
```

  where each line of the matrix contains the coefficients of a 2nd order structure.

$$H_k(z)=\frac{b_{0,k}+b_{1,k}z^{-1}+b_{2,k}z^{-2}}{1+a_{1,k}z^{-1}+a_{2,k}z^{-2}}$$

- `G` is a scalar which represents the global gain of the system. If `G` is not specified, it is included in the first section.

`[SOS,G]=zp2sos(Z,P,K)`

- it returns a `SOS` matrix which contains the coefficients of the 2nd order sections for the factorization of the transfer function $H(z)$. `Z` represents the vector of zeros, `P` represents the vector of poles and `K` represents the gain of the pole-zeros decomposition. The poles and zeros must be given in complex conjugated pairs.
- `G` is a scalar which represents the global gain of the system. If `G` is not specified, it is included in the first section.

5.1.3. The Parallel Form

In this case the transfer function $H(z)$ will be decomposed into partial fractions. Assume the transfer function's coefficients, $a_k, b_k \in \mathbf{R}$; then the partial fractions with a denominator of order 1 will generally have complex coefficients. We will group the partial fractions that correspond to complex conjugated poles and so the partial fractions $H_p(z)$ resulted will have order 2 and real coefficients, $a_{k,p}, b_{k,p} \in \mathbf{R}$. The transfer function can be represented as:

$$H(z) = C + \sum_{p=1}^{P} \frac{b_{0,p} + b_{1,p} z^{-1}}{1 + a_{1,p} z^{-1} + a_{2,p} z^{-2}} = C + \sum_{p=1}^{P} H_p(z) \tag{0.4}$$

The parallel form is presented in figure 5.4.

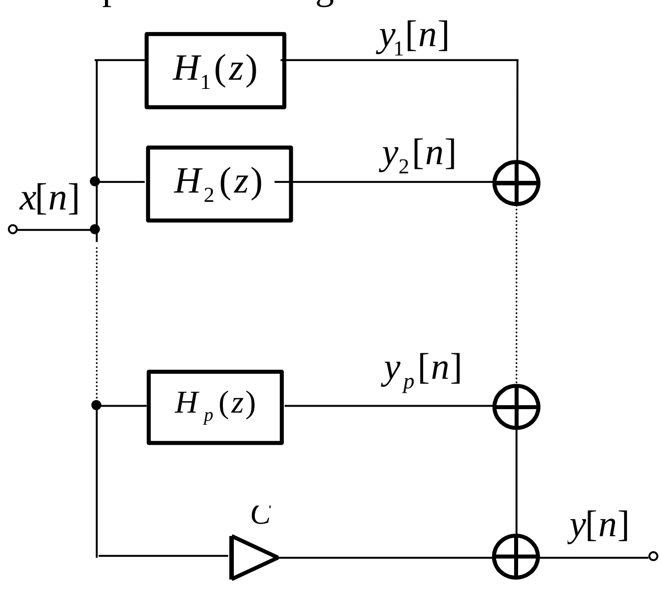

Figure 5.4. The parallel form

The MATLAB function `residuez` achieves the decomposition in simple fractions of the transfer function $H(z)=\dfrac{B(z)}{A(z)}$. It can be applied in 2 ways:

`[R,P,K]=residuez(B,A)`

- it decomposes in simple fractions the function $H(z)=\dfrac{B(z)}{A(z)}$ defined by the coefficients vectors `B` and `A` as follows:

$$\frac{B(z)}{A(z)}=\frac{r_1}{1-p_1z^{-1}}+\ldots+\frac{r_N}{1-p_Nz^{-1}}+k_1+k_2z^{-1}+\ldots$$

- `R` and `P` are column vectors containing the residuez and poles, respectively.
- `K` is a line vector for the free terms (if the order of the numerator is greater than the one of the denominator).
- If p_j is a multiple pole of order m, the decomposition in simple fractions will contain terms of the form:

$$\frac{r_j}{1-p_jz^{-1}}+\frac{r_{j+1}}{\left(1-p_jz^{-1}\right)^2}+\ldots+\frac{r_{j+m-1}}{\left(1-p_jz^{-1}\right)^m}$$

`[B,A]=residuez(R,P,K)`

- it makes the transfer function $H(z)$ from the decomposition in simple fractions.

In this way, in order to obtain the decomposition in the parallel structure with real coefficients, the following method can be applied (only if the poles are ordered in complex conjugated pairs).

```
[r,p,k]=residuez(b,a)
[b1,a1]=residuez(r(1:2),p(1:2),[])
[b2,a2]=residuez(r(3:4),p(3:4),[])
```
etc...

5.1.4. The Lattice-Ladder Form

Applying the synthesis algorithm for an IIR filter transfer function:

$$H(z)=\frac{B(z)}{A(z)}=\frac{Y(z)}{X(z)} \tag{0.5}$$

leads to a set of reflection coefficients corresponding to a lattice structure and a set of ladder coefficients (figure 5.5).

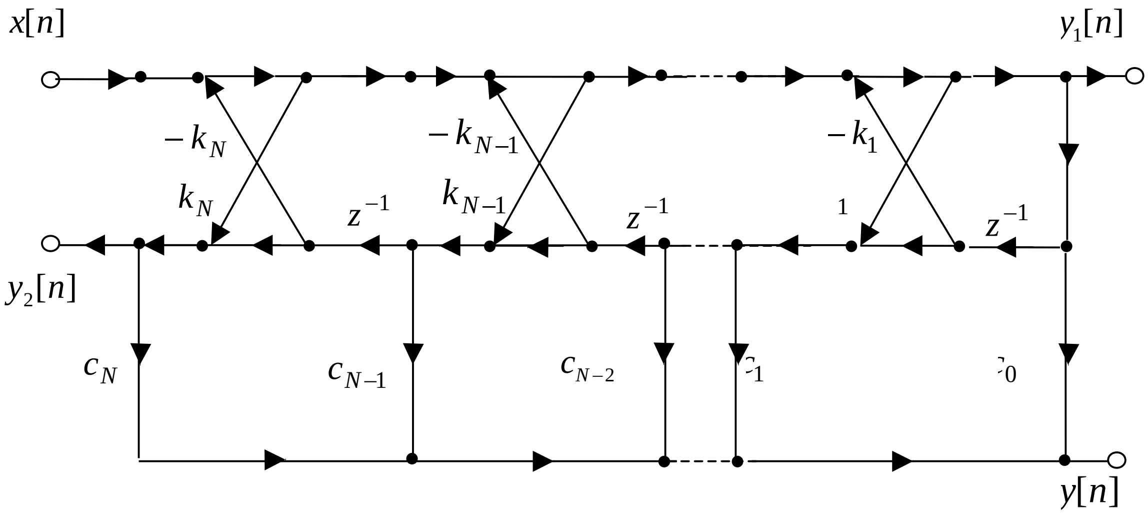

Figure 5.5. Lattice-ladder form

In the figure 5.5 it can be observed that beside the input node $x[n]$ and the output node $y[n]$, there are 2 more nodes: $y_1[n]$ and $y_2[n]$. It can be proved that the partial transfer functions corresponding to those output nodes are:

$$H_1(z) = \frac{Y_1(z)}{X(z)} = \frac{1}{A(z)} \qquad (0.6)$$

$$H_2(z) = \frac{Y_2(z)}{X(z)} = \frac{z^{-N} A(z^{-1})}{A(z)} = \frac{\tilde{A}(z)}{A(z)} \qquad (0.7)$$

From equation (0.6), it is obvious that the partial transfer function (corresponding to the signal path from the input $x[n]$ to the output $y_1[n]$) has only poles (without any ladder).

In equation (0.7), the reciprocal polynomial of the denominator $A(z)$ has been denoted by $\tilde{A}(z)$ (the reciprocal is obtained by taking the coefficients of $A(z)$ in a reversed order).

$$\tilde{A}(z) = a_N + a_{N-1} z^{-1} + \ldots + a_0 z^{-N} \qquad (0.8)$$

Obviously, $H_2(z)$ is an all-pass function and it can be represented only with a lattice, without the ladder structure, the output being $y_2[n]$.

The synthesis algorithm of an IIR transfer function computes the reflection coefficients $k_1, \ldots, k_N$ starting from the denominator ($A(z)$) polynomial coefficients $a_0, a_1, \ldots, a_N$. The algorithm is presented below:

$$\begin{array}{l}
\text{for } j = 0:1:N \\
\quad a_{N,j} = a_j \\
\text{end} \\
\text{for } i = N:\text{-}1:2 \\
\quad k_i = a_{i,i} \\
\quad \text{for } j = 1:1:i\text{-}1 \\
\qquad a_{i-1,j} = \dfrac{a_{i,j} - k_i a_{i,i-j}}{1-k_i^2} \\
\quad \text{end} \\
\text{end} \\
k_1 = a_{1,1}
\end{array}$$

The ladder coefficients $c_0,...,c_N$ are also determined in a recursive manner, using the values $b_0, b_1, ..., b_N$ from the nominator polynomial $B(z)$ of the transfer function.

$$\begin{array}{l}
c_N = b_N \\
\textbf{for } l = N-1, N-2, \cdots, 0 \\
\quad c_l = b_l - \sum_{i=l+1}^{N} c_i a_{i,i-l} \\
\textbf{end}
\end{array}$$

The MATLAB function `tf2latc` computes the coefficients of the lattice structure.

***Syntax*:**

`[K,C]=tf2latc(B,A)`

- it returns the vector `K` which contains the reflection coefficients k_i and the vector `C` which contains the coefficients of the ladder structure c_i for an IIR filter with the coefficients of the numerator in vector `B` and the coefficients of the denominator in vector `A`, normalized to a_0.

`K=tf2latc(B)`

- it returns the vector `K` which contains the reflection coefficients k_i for a FIR filter with the coefficients of the transfer function in vector `B`, normalized to b_0.

`K=tf2latc(1,A)`

- it returns the vector `K` which contains the reflection coefficients k_i for an IIR filter only with poles, with the coefficients of the denominator in vector `A`, normalized to a_0.

Remark:
If from computations one of the reflection coefficients equals 1, then the function `tf2latc` will generate an error (because the term $1-k_i^2$ which appears in the denominator is 0).

The transfer function can also be obtained starting from the reflection coefficients and the coefficients of the ladder structure, using the function `latc2tf`. For details type `help latc2tf`.

Example:
Let an IIR filter have the following transfer function:

$$H(z)=\frac{B(z)}{A(z)}=\frac{(0.5+z^{-1})(1+0.5z^{-1})(1-1.2z^{-1}+1.8z^{-2})}{1+0.5z^{-1}+1.2z^{-3}+0.5z^{-4}}$$

Synthetize the structures for the direct form 2, cascade, parallel and lattice forms (all structures will have real coefficients).

// The polynomial at the denominator is:
```
a=[1 0.5 0 1.2 0.5];
```

// The polynomial at the numerator is obtained from the zeros of the transfer function.
```
z1=[-2; -0.5];
z2=roots([1 -1.2 1.8]);
```

// Because the first free term is 0.5, the polynomial must be multiplied by 0.5.
```
b=0.5*poly([z1; z2])

b =
    0.5000   0.6500  -0.1000   1.6500   0.9000
```

Having the polynomials from the transfer function, we can synthesize the filter in a direct form 2 structure (figure 5.6).

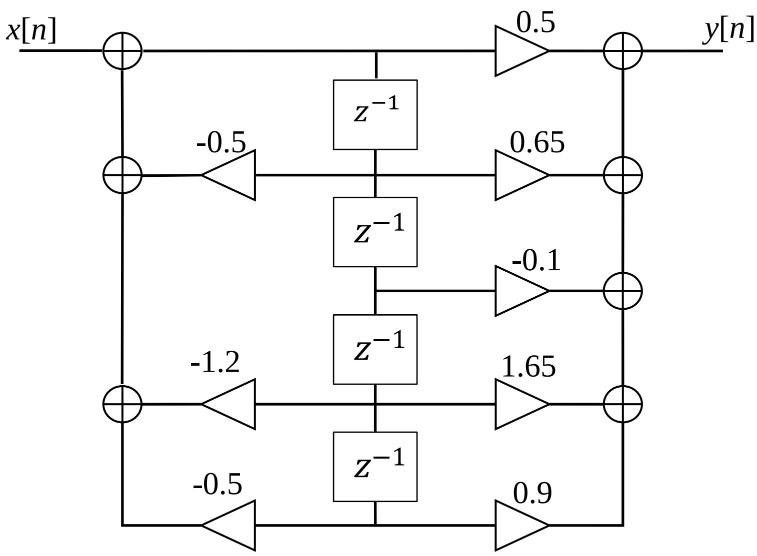

Figure 5.6. Direct form 2 structure

// The cascade form:

```
[SOS,G]=tf2sos(b,a)
SOS =
    1.0000   2.5000   1.0000   1.0000   1.5206   0.4564
    1.0000  -1.2000   1.8000   1.0000  -1.0206   1.0955
G  = 0.5
```

The cascade decomposition (see figure 5.7):

$$H\left(z\right)=G\cdot H_1(z)\cdot H_2(z)=0.5\cdot\frac{1+2.5z^{-1}+z^{-2}}{1+1.52z^{-1}+0.456z^{-2}}\cdot\frac{1-1.2z^{-1}+1.8z^{-2}}{1-1.02z^{-1}+1.095z^{-2}}$$

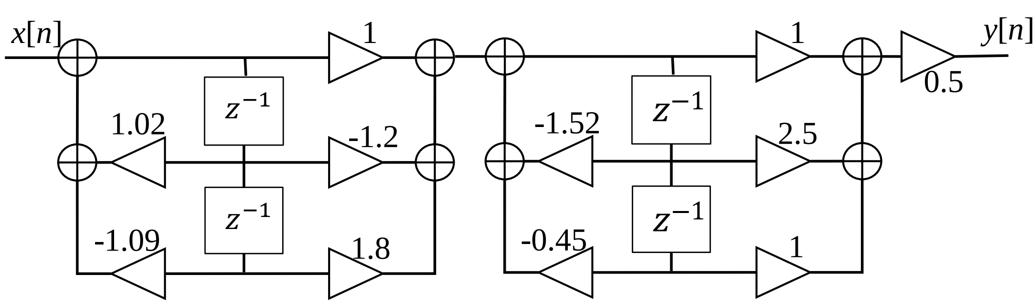

Figure 5.7. Cascade form realization

// The parallel form

```
[r,p,k]=residuez(b,a)

r =
  -0.4424
  -0.2499 - 0.0094i
  -0.2499 + 0.0094i
  -0.3579
p =
  -1.1091
   0.5103 + 0.9138i
   0.5103 - 0.9138i
  -0.4115
k =
    1.8000
```

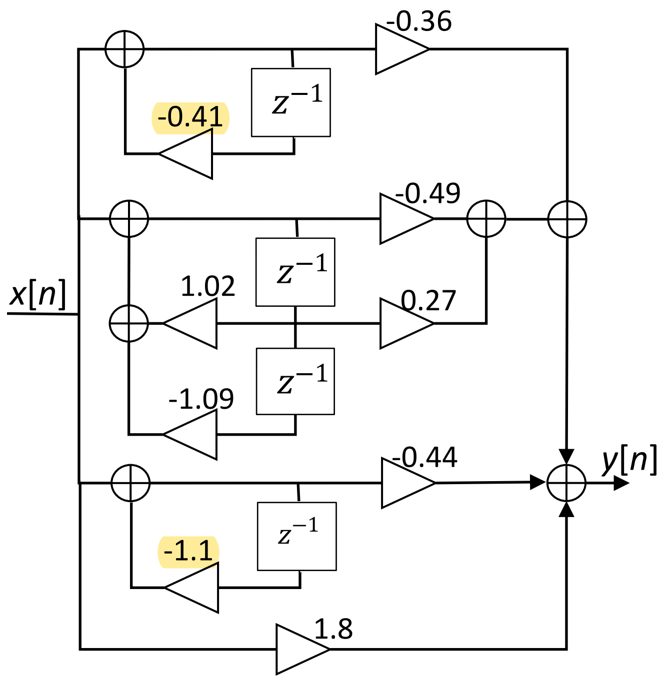

Figure 5.8. Parallel form structure

It can be noticed that there are complex poles and residues. Because we want a structure with real coefficients, we will group the pair of complex conjugated values into a 2nd order transfer function with real coefficients:

```
[b2,a2]=residuez(r(2:3),p(2:3),[])
```

```
b2 =
   -0.4998    0.2722
a2 =
    1.0000   -1.0206    1.0955
```

The parallel decomposition:

$$H(z) = \frac{r_1}{1 - p_1 z^{-1}} + \frac{B_2(z)}{A_2(z)} + \frac{r_4}{1 - p_4 z^{-1}} + k =$$

$$= \frac{-0.44}{1 + 1.1z^{-1}} + \frac{-0.49 + 0.27\, z^{-1}}{1 - 1.02z^{-1} + 1.09z^{-2}} + \frac{-0.36}{1 + 0.41z^{-1}} + 1.8$$

// The coefficients of the lattice form are obtained directly:

```
[k,c]=tf2latc(b,a)

k =
    0.3061
   -0.2794
    1.2667
    0.5000

c =
   -1.3176
   -0.4432
    0.0600
    1.2000
    0.9000
```

Remarks:

- The exact values of the coefficients obtained above have a much greater number of decimal digits than the ones displayed in MATLAB in short format. In practice, because of the representation of numbers on a finite number of bits, differences (truncations) of the coefficients' values with respect to the simulated ones.
- In the previous example it can be seen that the system is unstable, either by noticing that one of the poles has a modulus greater than 1 or by using the Schür-Cohn test for the reflection coefficients k and noticing that the coefficient k_3 is greater than 1.

E1. *Exercises:*

Given the IIR systems with the following transfer functions:

1. $H(z) = \dfrac{2-3.6z^{-1}+1.3z^{-2}+0.4z^{-3}-0.2z^{-4}}{1-2.8z^{-1}+3.15z^{-2}-1.75z^{-3}+0.425z^{-4}}$
2. $H(z) = \dfrac{(1-2z^{-1}+3z^{-2}-4z^{-3})(1-z^{-3})}{1-0.8z^{-2}+0.66z^{-4}+0.35z^{-6}}$
3. $H(z) = \dfrac{1}{1-1.27z^{-1}+1.19z^{-2}+1.18z^{-3}+0.4z^{-4}}$
4. $H(z) = \dfrac{0.4-0.7z^{-1}-0.175z^{-2}+z^{-3}}{1-0.175z^{-1}-0.7z^{-2}+0.4z^{-3}}$
5. $H(z) = \dfrac{(z-0.9e^{j\cdot 0.95\cdot\pi/4})(z-0.9e^{-j\cdot 0.95\cdot\pi/4})(z-0.9e^{j\cdot 1.05\cdot\pi/4})(z-0.9e^{-j\cdot 1.05\cdot\pi/4})}{(z-0.95e^{j\pi/4})(z-0.95e^{-j\pi/4})(z-0.9e^{j\pi/4})(z-0.9e^{-j\pi/4})}$

Synthetize and draw the structures for the direct forms 1 and 2, cascade, parallel, and lattice forms. All structures should have real coefficients.