# 4. Digital filters

In the previous chapter we have discussed linear, shift invariant discrete systems. The most important members of this class of systems are the digital filters, and in this chapter we will discuss a number of different schemes for digital filtering.

There are two major classifications that can be made for digital filters

I.1 non recursive digital filters (NRDF)
I.2 recursive digital filters (RDF)

II.1 finite impulse response digital filters (FIR)
II.2 infinite impulse response digital filters (IIR)

As will be shown, the first classification concentration on the structure of the filter, whereas the second scheme only deals with the impulse response, which is a more global characteristic of the filter and does not uniquely specify the structure. In the sequel the relation between the two clasification schemes will be discussed, and some examples will be given.

## 4.1. Non recursive digital filters

In section 3.2 it was indicated that a linear shift invariant discrete system, constructed of adders, delays and multipliers can be described by one - or a set of - difference equations. An example of such a set of difference equations was given in eq(3.4). In that equation x(n) was the input and y(n) the output signal, and v(n) an internal variable of the system.
Such a set of difference equations will be called non recursive, if none of the internal variables or the output depends on previous values of itself. This means that the set of difference equations can be put into the form:

$$u_1(n) = \sum_{k=0}^{K_1} a_{1k}\, x(n-k)$$

$$u_2(n) = \sum_{k=0}^{K_2} a_{2k}\, x(n-k) + \sum_{l=0}^{L_{2,1}} b_{2l}^{(1)}\, u_1(n-l)$$

$$u_m(n) = \sum_{k=0}^{K_m} a_{mk}\, x(n-k) + \sum_{l=0}^{L_{m,1}} b_{ml}^{(1)}\, u_1(n-l) + - - - -$$

$$- - - - + \sum_{l=0}^{L_{m,m-1}} b_{ml}^{(m-1)}\, u_{m-1}(n-l)$$

$$y(n) = \sum_{i=0}^{I} c_i\, x(n-i) + \sum_{k=1}^{m} \sum_{i=0}^{I_m} d_{ki}\, u_k(n-i) \qquad (4.1).$$

From (4.1) it follows that $u_1$ only depends on x, $u_2$ only on x and $u_1$ etc, whereas y(n) may depend on all of the internal variables but not on previous values of itself
As a simple example consider:

$$u(n) = a_1\, x(n) + a_2\, x(n-1)$$
$$y(n) = b_1\, x(n) + c_1\, u(n) + c_2\, u(n-1) \qquad (4.2)$$

The corresponding structure is shown in fig.4.1
From (4 1) it can be seen that a non recursive structure does not have any closed loops.

Now we have the following proposition:

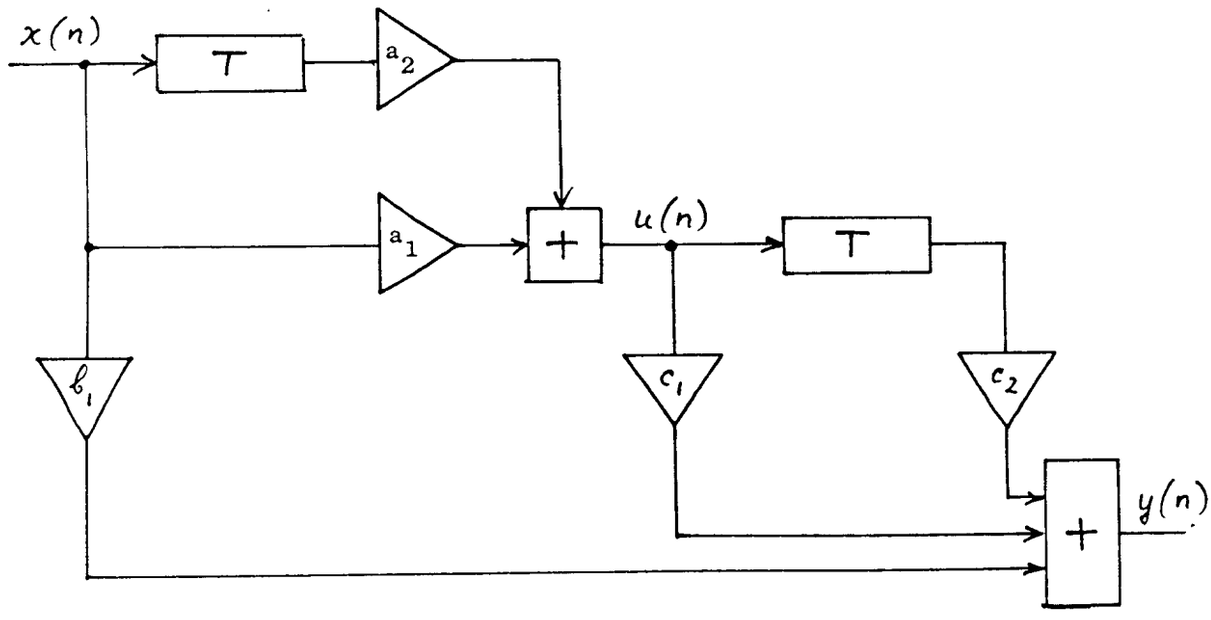


Fig.4.1

Proposition 4. A non recursive digital filter (with a finite number of delays) has a finite impulse response.

This theorem thus states that any realizable non recursive filter is an FIR filter. (We will see in section 4.2 that the converse is not true !)
The proof is very simple. Insert in eq.(4.1) $x(n) = \delta(n)$.
Then we find that

$$u_1(n) = 0 \qquad n > N_1 = K_1$$

$$u_2(n) = 0 \qquad n > N_2 = \max\,(K_2,\ N_1 + L_{2,1})$$

$$\vdots$$

$$u_m(n) = 0 \qquad n > N_m = \max\,(K_m,\ N_1 + L_{m,1},\ N_2 + L_{m,2},\ \text{---}$$

$$\text{---},\ N_{m-1} + L_{m,m-1})$$

$$y(n) = h(n) = 0 \qquad n > N_{m+1} = \max\,(I, N_1 + I_1,\ \text{- - -}$$

$$\text{---},\ N_m + I_m)$$

and clearly $N_{m+1}$ is a finite number, which completes the proof.
In the example of eq (4.2) the impulse response has a length ( duration) of 3, which means that three values of the impulse response are different from zero.

A special form of non recursive digital filter is the quite popular transversal filter. This filter is described by one equation:

$$y(n) = \sum_{i=0}^{M} a_i\, x(n-i) \tag{4.3}$$

For this filter the impulse response is:

$$h(n) = \begin{cases} 0 & n < 0,\ n > M \\ a_n & 0 \leq n \leq M \end{cases} \tag{4.4}$$

and has length M+1.

Since a non recursive filter is a specific type of FIR filter we will defer a further discussion of the properties to the corresponding section 4.3.

## 4.2. Recursive digital filters

Any filter in which at least one of the variables depends on previous values of itself is called a recursive digital filter (RDF), and thus every RDF must contain at least one closed loop. In eq.(3.4) we have given an example of a recursive set of equations, and indeed the corresponding structure contained a closed loop.

For realizability it is required that each closed loop contains at least one delay element since in a delay-free loop the adders or multipliers must operate on values that are not yet determined.

A recursive filter can have either a finite or an infinite impulse response. To show this consider the two systems in fig.4.2.

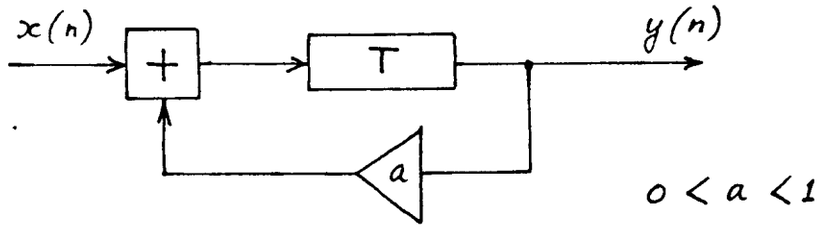


Fig. 4.2.a.

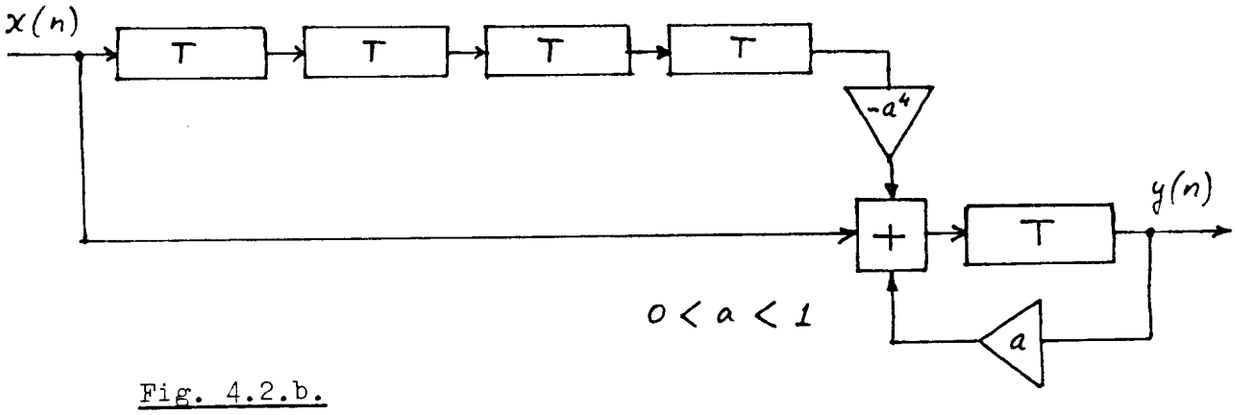


Fig. 4.2.b.

The system in fig. 4.2a is described by the simple relation:

$$y(n) = x(n-1) + a\, y(n-1) \tag{4.5}$$

and has the impulse response:

$$h_1(n) = \begin{cases} 0 & n \leq 0 \\ a^{n-1} & n > 0 \end{cases} \tag{4.6}$$

See fig 4.3.

The second system is described by:

$$y(n) = x(n-1) - a^4\, x(n-5) + a\, y(n-1) \tag{4.7}$$

Inserting $x(n) = \delta(n)$ we find (see fig.4.3)

$$h_2(n) = \begin{cases} 0 & n \leq 0 \\ a^{n-1} & 1 \leq n \leq 4 \\ 0 & n \geq 5 \end{cases} \tag{4.8}$$

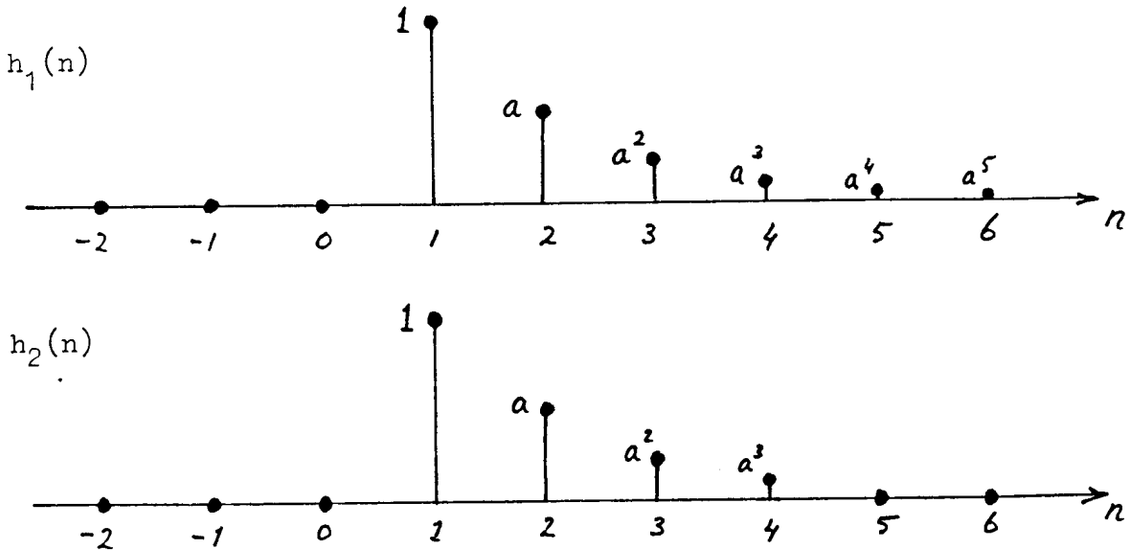


fig.4.3

The fact that $h_2(n) = 0$ for $n \geq 5$ results because for n=4 the contribution from the "forward path" $(-a^4\, x(n-4))$ and the "feedback path" $(a.y(n))$ cancelled exactly. This exact cancellation requires that the arithmetical operations are performed with infinite precision If an error was made in either of the two paths, and the cancellation therefore was not exact, the residue would circulate in the feedback path and the output would differ from zero also for $n > 4$.

From this it can be remarked that recursive structures that realize a finite impulse response are very sensitive unless only multiplications with integer factors have to be performed.

A point that we have ignored hitherto are the initial conditions. There does not exist a digital filter that operates already since the beginning of time, and consequently every digital filter has been started at some time. Until now we have always assumed that the contents of the registers (delay elements) were zero until some value was put into it by the normal signal flow. This assumption requires that in the actual realization precautions are taken that assure that the filter always starts with empty registers.
There is in this respect an important difference between recursive and non recursive digital filters. If a non recursive filter with an impulse response of length N is started with non zero contents in the registers, then after at most N sampling periods the influence of these initial conditions has disappeared and the filter behaves just as if it were started with zero registers. Therefore special precautions to reset the registers in general are not required in these filters.

In a recursive filter the situation is quite different. If such a filter is started with any of the registers having a non zero content, then this may result in an additional response that can propagate in the feedback loop(s) of the filter. If the filter is stable such a transient response will eventually decay and converge to zero It should be noted, however, that even if the filter has a finite impulse response, the transients may have infinite duration As an example consider the transient response resulting from a non zero value in the x(n-4) register of the system in fig 4 2 b Assume the filter to be started at n=0 with:

$$x(-4) = 1$$

$$x(-3) = x(-2) = x(-1) = y(o) = 0$$

and without input, ($x(n) \equiv 0$). Then from eq (4.7) it follows:

$$y(n) = \begin{cases} 0 & n \leq 0 \\ -a^{n-5} & n \geq 1. \end{cases}$$

When determining the response of a recursive d.f. it is therefore very important to specify also the initial conditions The total response will consist of the sum of the response to the input signal assuming zero initial conditions and the transient response resulting from the initial conditions assuming no input signal.

## 4.3. Finite impulse response filters

Any digital filter that has an impulse response of finite duration is called a finite impulse response filter. Denoting the "length" of the impulse response by N+1 we thus have (assuming causality)

$$h(n) = 0 \qquad n<0, \quad n>N \tag{4.9}$$

Therefore the system function of such a filter is given by:

$$\tilde{H}(z) = \sum_{n=0}^{N} h(n) \quad z^{-n} \tag{4.10}$$

which may be factored according to:

$$\tilde{H}(z) = h(0) \prod_{n=1}^{N} (z - z_n) / z^N \tag{4.11}$$

From eq.(4.11) it can be concluded that H(z) has all its poles at z = 0.

We have already shown that an important subclass of FIR filters are the non recursive filters. And more precisely every finite impulse response can be realized by a non recursive filter. From eq.(4.10) two non recursive structures can be derived. The first realization is a transversal filter, and is designated as direct form structure (see fig.4.4)

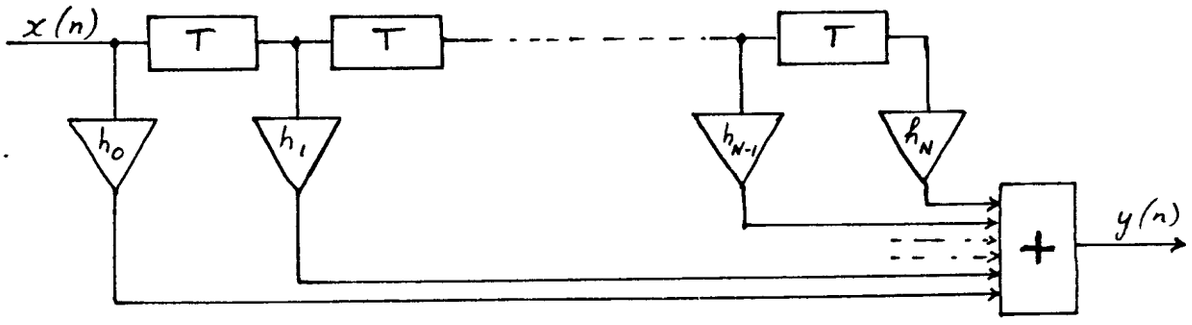


Fig. 4.4.

The second realization can be derived from eq.(4.10) by defining the interval variables:

$$u_0(n) = h_N\, x(n)$$

$$u_k(n) = h_{N-k}\, x(n) + u_{k-1}(n-1). \qquad k=1,---,N.$$

$$y(n) = u_N(n)$$

It is not difficult to show that the corresponding system function satisfies (4.10) (Do it !)
The corresponding filter realization is called the transpose direct form because it can be obtained from the direct form by reversing the signal flow graph. It is shown in fig.4.5.

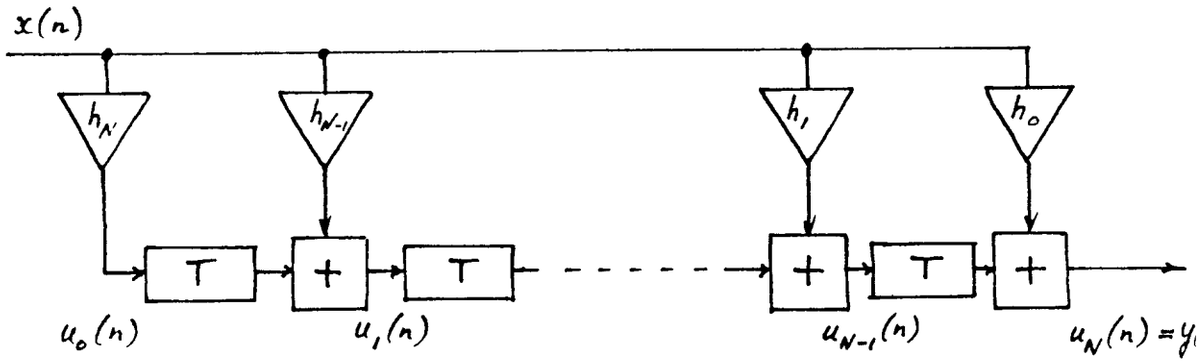


Fig.4.5

Both direct realizations utilize exactly N registers and therefore they are called canonic realizations. Both realizations require N+1 multiplications and N additions to compute a new output sample, and thus they are computationally equivalent.

Now let us reconsider the impulse response of eq.(4.8). As stated above this impulse response can be realized by a transversal FIR filter, and the circuit of this filter is shown in fig.4.6.

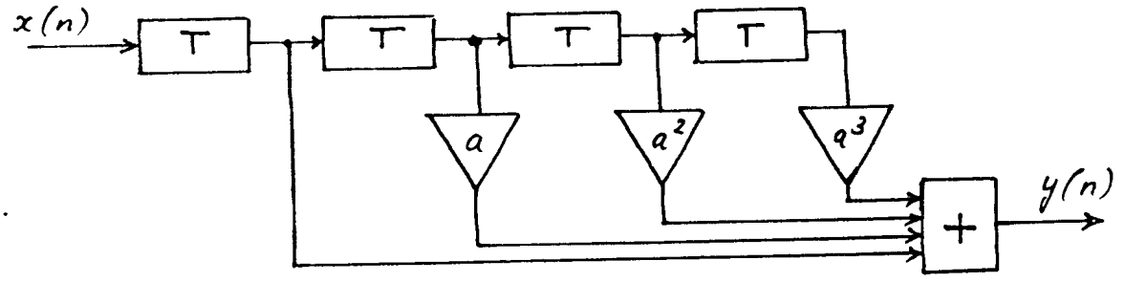


Fig.4.6

As can be seen it requires 4 delay elements, and 3 multiplications and 3 additions to compute an output sample. It realizes the same impulse response, and consequently the same transmission function as the structure of fig.4.2.6 which has 5 delay elements but this latter structure only requires 2 multiplications and 2 additions to compute an output sample.
The difference between the two structures is that the recursive filter utilizes the peculiar properties that are present in the form of the impulse response i.e. that consecutive samples are closely related; in fact the impulse response is determined by only 2 parameters: the coefficient a and its length N.

In many applications, and especially in data transmission, it is desirable to use filters that have a linear phase which means a constant group delay. In this way, signals in the passband of the filter are reproduced exactly at the filter output except for a delay corresponding to the slope of the phase. It can be shown that a linear phase is obtained if h(n) satisfies the symmetry relation:

$$h(n) = \pm h\,(K-n) \qquad n=0,1,2,\text{---} \qquad (4.12)$$

for some integer K.
Since for a causal system $h(n) = o \quad n<o$, eq.(4.12) requires $h(n) = o \quad n>K$ and thus a linear phase can only be obtained by an FIR filter which is either symmetrical or anti-symmetrical around the half of its length.
The linear phase property is quite important so that almost all FIR filters that are used to date are symmetrical. We will come back to this point in section 6 when discussing approximation procedures for FIR filters.

## 4.4. Infinite impulse response filters

The impulse response of a stable filter will converge to zero (see proposition 3). This means that any desired impulse response h(n) can be approximated arbitrarily good by a FIR filter with response $h_N(n)$ such that

$$h_N(n) = \begin{cases} h(n) & 0 \leq n \leq N \\ 0 & n > N \end{cases} \qquad (4.13)$$

by taking N sufficiently large. But such a filter will require N+1 delay elements and N+1 multiplications and N additions to compute an output sample.
For large N the complexity of such a solution may become prohibitive.
Especially filters with high Q-factors (which means that they have a narrow transition band and large stopband attenuation) have impulse responses that require a large value of N to be sufficiently approximated, and such filters can more economically be realized by a recursive structure.

It will be clear that not every infinite impulse response can be realized. A digital system constructed from delays, multipliers and adders as explained in section 3 can realize system functions of the form:

$$\tilde{H}(z) = \frac{T(z)}{N(z)} = \frac{\sum_{k=0}^{M} a_k z^{-k}}{\sum_{k=0}^{N} b_k z^{-k}} \qquad (4.14)$$

Now for an arbitrary impulse response h(n) the corresponding system function is

$$\tilde{H}(z) = \sum_{n=0}^{\infty} h(n) z^{-n} \qquad (4.15)$$

Therefore the impulse response can only be realized if the infinite sum in (4.15) can be brought into the form of (4.14). This requires certain relations to exist between successive values of the impulse response, just as was the case with the recursive realization of the FIR discussed in section 4.2.
From the discussion in section 3.7 it follows that responses of the form:

$$\sum_{k=1}^{N} A_k p_k^n u(n) \qquad (4.16)$$

can be realized (where $p_k$ may be real or complex but if it is complex, then a $p_l$ must exist such that $p_l = p_k^*$, $A_l = A_k^*$).
The filter will be stable if $|p_k| < 1$ for all k.

As for the FIR case the impulse response does not uniquely specify the structure. We will now briefly discuss some possible structures that may be used to implement IIR of the form (4.16). We start in this case from the system function.
Noting that

$$\tilde{H}(z) = \tilde{Y}(z)/\tilde{X}(z)$$

a direct realization follows from eq. (4.14):

$$\tilde{Y}(z) . \sum_{k=0}^{N} b_k z^{-k} = \tilde{X}(z) . \sum_{k=0}^{M} a_k z^{-k}$$

or

$$\sum_{k=0}^{N} b_k y(n-k) = \sum_{k=0}^{M} a_k x(n-k)$$

thus

$$y(n) = - \sum_{k=1}^{N} \frac{b_k}{b_o} . y(n-k) + \sum_{k=0}^{M} \frac{a_k}{b_o} x(n-k) \qquad (4.17)$$

The corresponding structure is shown in fig.4.7.

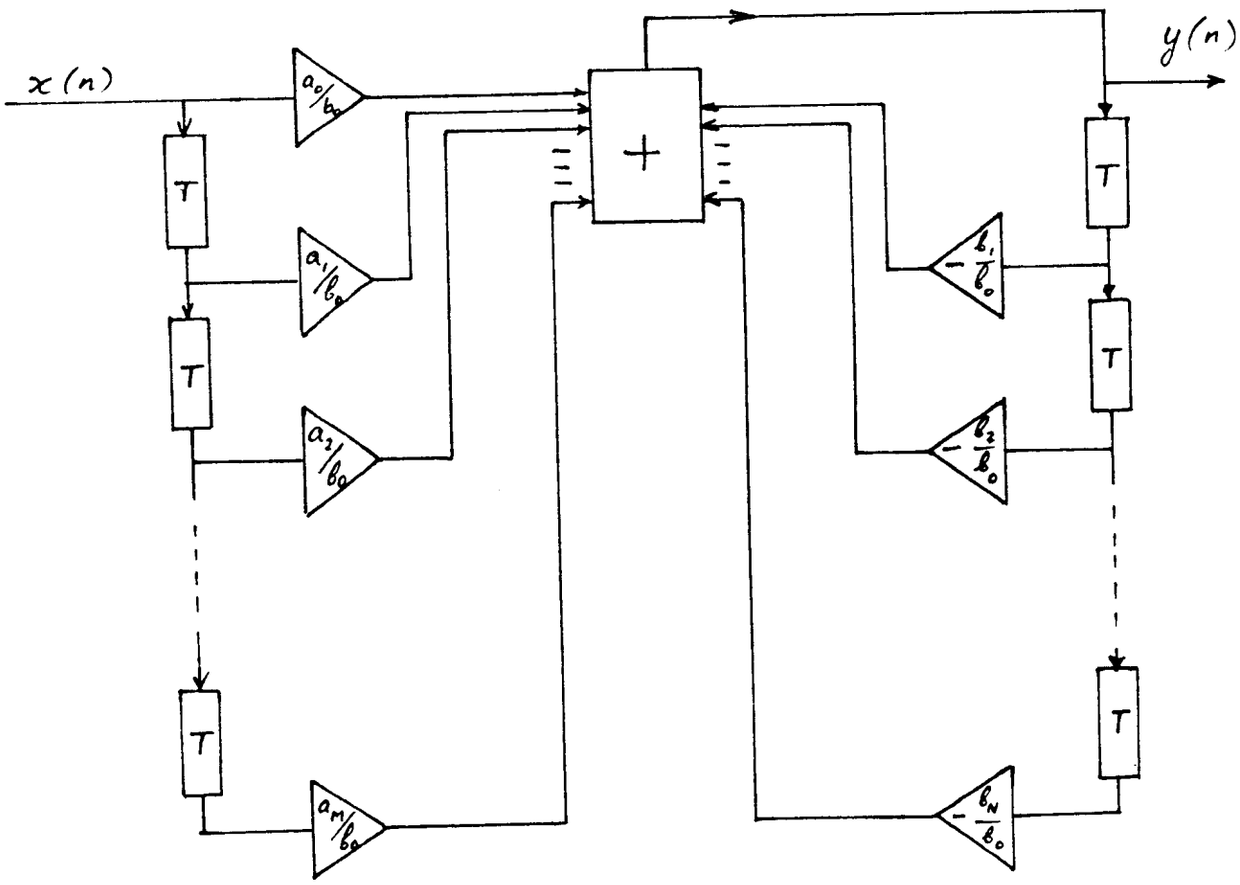


fig.4.7.

It is called the direct form 1; and requires N+M delay elements and N+M+1 multipliers and N+M adders to compute an output sample. We see that this system consists of a cascade of two parts. The first is a non recursive part which attributes the zeros of the system function. The second a purely recursive part that forms the poles. Since we are dealing with a cascade of linear systems we may reverse the order of the two subsystems resulting in the system of fig.4.7a, where it is now assumed that M=N.

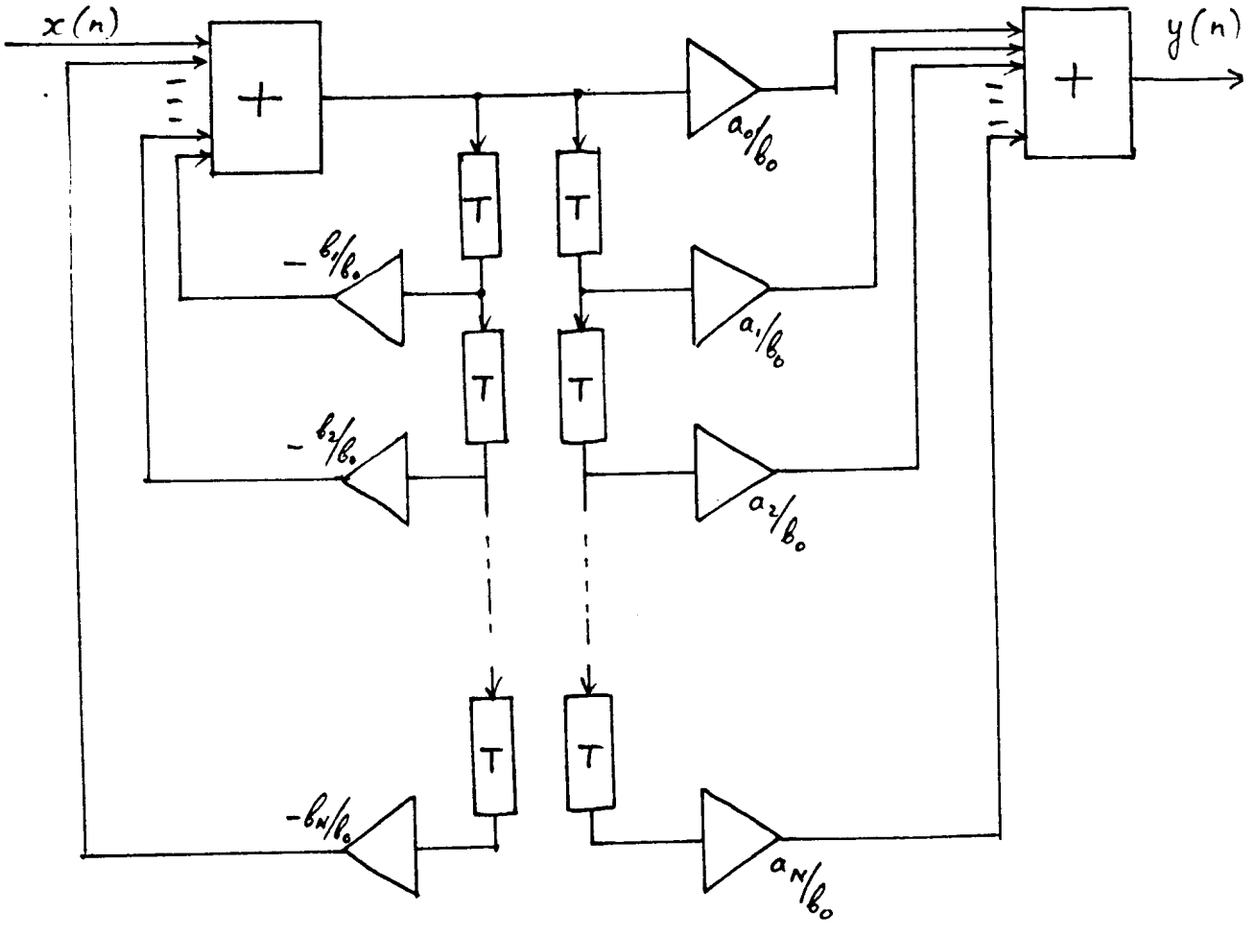


fig.4.7a

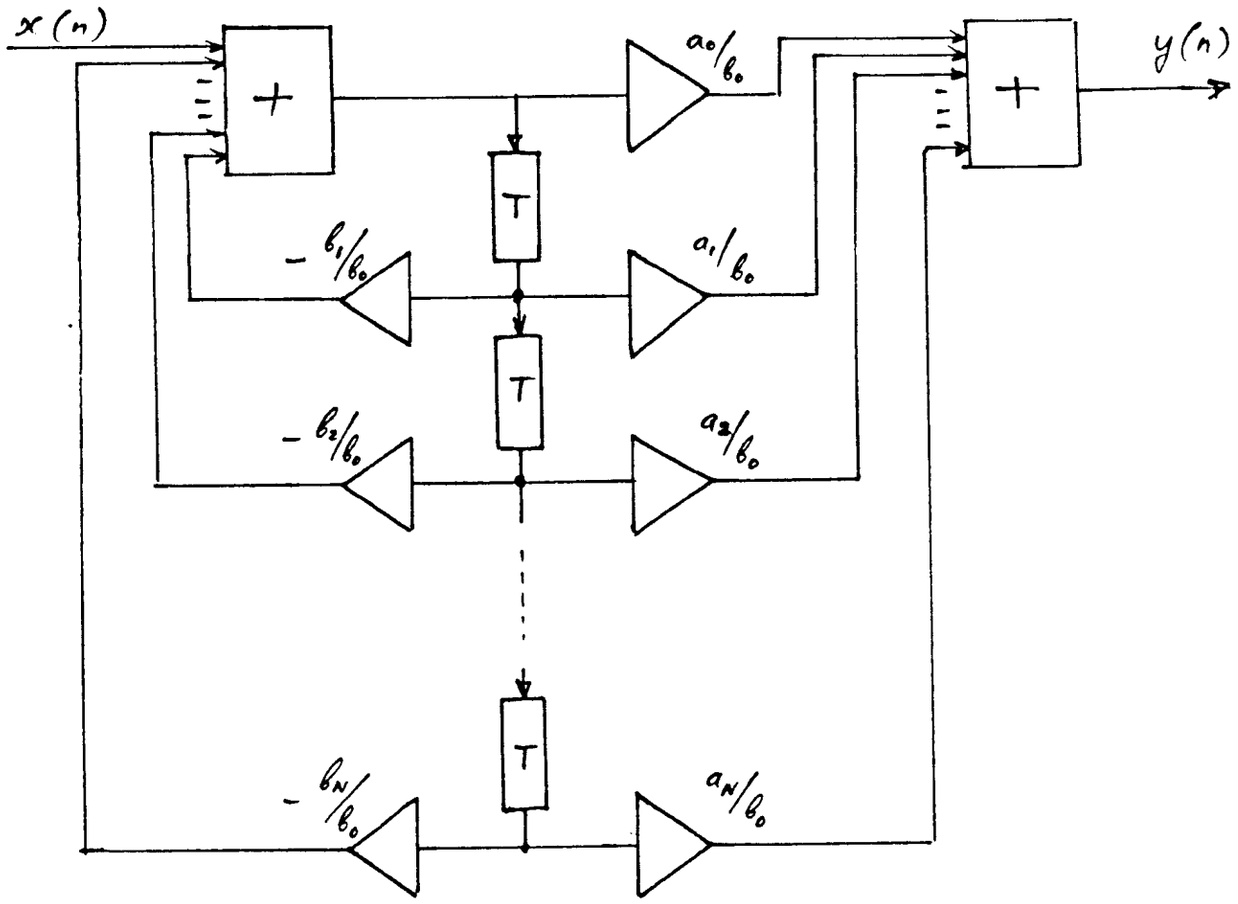


fig.4.7b

It will be clear that the two chains of registers contain exactly the same information and can therefore be combined as shown in fig.4.7b. This structure is referred to as the direct form II and is canonic as concerns the number of delays.

Both the direct form I and II have a transpose which is a structure obtained by reversing the signal flow (output becomes input and vice versa, adders become nodes and vice versa). There exists a theorem stating that a structure obtained by transposition of a structure of a digital system has the same system function as the original system.It may be instructive to try to derive the transpose direct form I and II structures.

Two more structures for realizing recursive filters are of importance. The first is the cascade realization and is derived by rewriting the system function as indicated in eq. (3.12) (assuming again N=M)

$$H(z) = \alpha . \frac{\prod_{k=1}^{N} (z-z_k)}{\prod_{k=1}^{N} (z-p_k)} = \alpha . \frac{\prod_{k=1}^{N} (1-z_k z^{-1})}{\prod_{k=1}^{N} (1-p_k z^{-1})}$$

As stated before $z_k$ and $p_k$ are either real or there exist $z_l$ and $p_l$ such that $z_l = z_k^*$, $p_l = p_l^*$, Taking together these complex conjugate poles and zeros we obtain

$$\tilde{H}(z) = \alpha \prod_{\substack{\text{real}\\ \text{poles,}\\ \text{zeros}}} \left( \frac{1 - z_k z^{-1}}{1 - p_k z^{-1}} \right) \cdot \prod_{\substack{\text{complex}\\ \text{conjugate}\\ \text{poles,zeros}}} \left( \frac{1 + a_{1k} z^{-1} + a_{2k} z^{-2}}{1 - b_{1k} z^{-1} - b_{2k} z^{-2}} \right) \quad (4.18)$$

$\tilde{H}(z)$ in this form can be realized by a cascade of first and second order sections. (see fig.4.8).

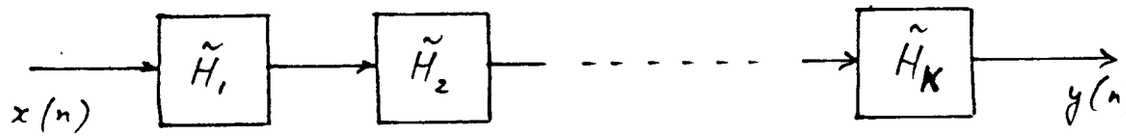


fig.4.8.

Each of these sections realizes one real or two complex conjugate poles and one real or two complex conjugate zeros. They may be realized in any of the direct forms previously discussed.
There is a certain amount of arbitrariness in the decomposition according to eq.(4.18) and thus in the structure of fig.4.8.
First in the combination of the poles and zeros,
secondly in the ordering of the various sections. Whatever combination and ordering is chosen, the resulting filter will realize $\tilde{H}(z)$, i.e. the linear behaviour of all such filters will be identical.
However, as concerns the effects
of the finite representation of the coefficients and the signals different orderings and pole-zero combinations may result in entirely different behaviour and it pays in general to look for an optimum.

Still a different structure results from a partial fraction expansion of $\tilde{H}(z)$ as in eq. (3.13). Again combining complex conjugate pole pairs, such a decomposition results in

$$\tilde{H}(z) = a_o + \sum_{\substack{\text{real}\\ \text{poles}}} \frac{a'_k}{1 - p_k z^{-1}} + \sum_{\substack{\text{complex}\\ \text{poles}}} \frac{a'_{ok} + a'_{1k} z^{-1}}{1 - b_{1k} z^{-1} - b_{2k} z^{-2}}$$

In this form H(z) can be realized by a parallel connection of first and second order sections as shown in fig.4.9.

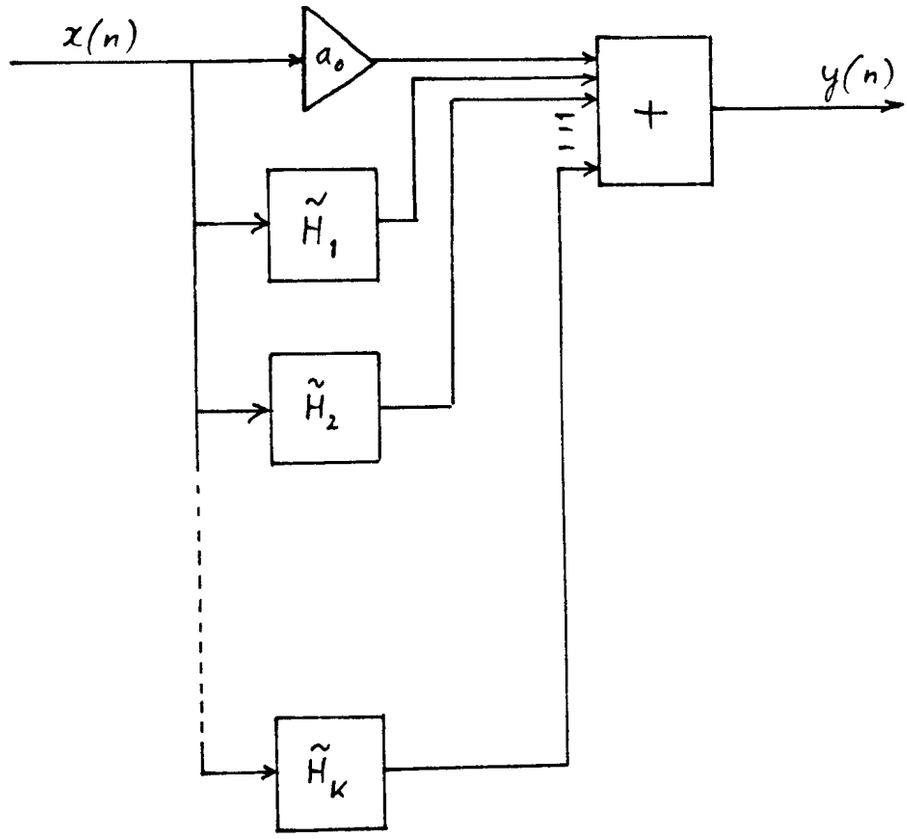


fig.4.9.

Each of the sections $\tilde{H}_k$ realizes one real or two complex conjugat poles. The zeros are obtained by combining the output signals of the various sections, thus by a compensation.

There is a countless number of other structures that has been proposed for some reason or the other to realize recursive digital filters, but they will not be discussed here.

## 4.5. Digital oscillators.

In section 1 it was mentioned that a digital sine oscillator can be used for spectral analysis of digital systems. Such an oscillator can be made with the structure of a 2nd order digital filter by inserting appropriate coefficients.

To see this we can use the well-known goniometric formulas

$$\cos\alpha + \cos\beta = 2\cos\left(\frac{\alpha+\beta}{2}\right)\cdot\cos\left(\frac{\alpha-\beta}{2}\right)$$

$$\sin\alpha + \sin\beta = 2\sin\left(\frac{\alpha+\beta}{2}\right)\cdot\cos\left(\frac{\alpha-\beta}{2}\right)$$

Inserting $\alpha = (n+1)\,\theta$, $\beta = (n-1)\theta$ we get

$$\cos (n+1)\theta = 2 \cos n\theta . \cos\theta - \cos (n-1)\,\theta$$

$$\sin (n+1)\theta = 2 \sin n\theta \cos\theta - \sin (n-1)\,\theta$$

Therefore if we want to obtain $y(n) = A \cos n\theta$ we can get it from a circuit that satisfies:

$$y(n+1) = 2 \cos\theta . y(n) - y(n-1)$$

starting it with initial conditions

$$y(o) = A$$

$$y(-1) = A \cos \theta$$

The structure is shown in fig.4.10 and is equal to the recursive part of a 2nd order section as discussed in section 4.4 of which $\tilde{H}(z)$ has denominator coefficients $b_1 = 2 \cos \theta$, $b_2 = -1$.

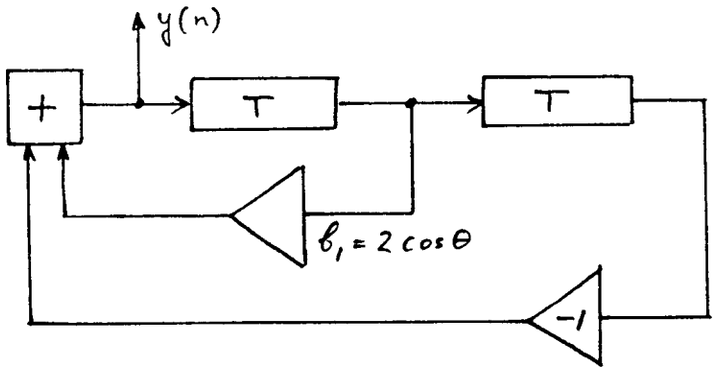


fig.4.10

The sine wave can be obtained by starting the system with other initial conditions. (Which ?)
The frequency can be "tuned" by changing the coefficient $b_1$:

$$\theta = \text{arc cos} \left( \frac{b_1}{2} \right).$$

(A different way to derive this structure is to start from eq.(1.18) with $\rho = 1$).

Different structures can be obtained by using different goniometric relations. A structure that delivers a cosine and sine simultaneously results from:

$$\cos(n+1)\theta = \cos n\theta . \cos\theta - \sin n\theta . \sin\theta$$

$$\sin(n+1)\theta = \cos n\theta . \sin\theta + \sin n\theta . \cos\theta$$

with the structure shown in fig.4.11.

Sine oscillators can be used to deliver the carrier in modulators. Frequently in modulation schemes both the cosine and the sine are needed to obtain in-phase and quadrature components. In that case the structure of fig.4.11 can be used. It has the disadvantage that 4 multipliers are required which makes it a rather complex circuit.

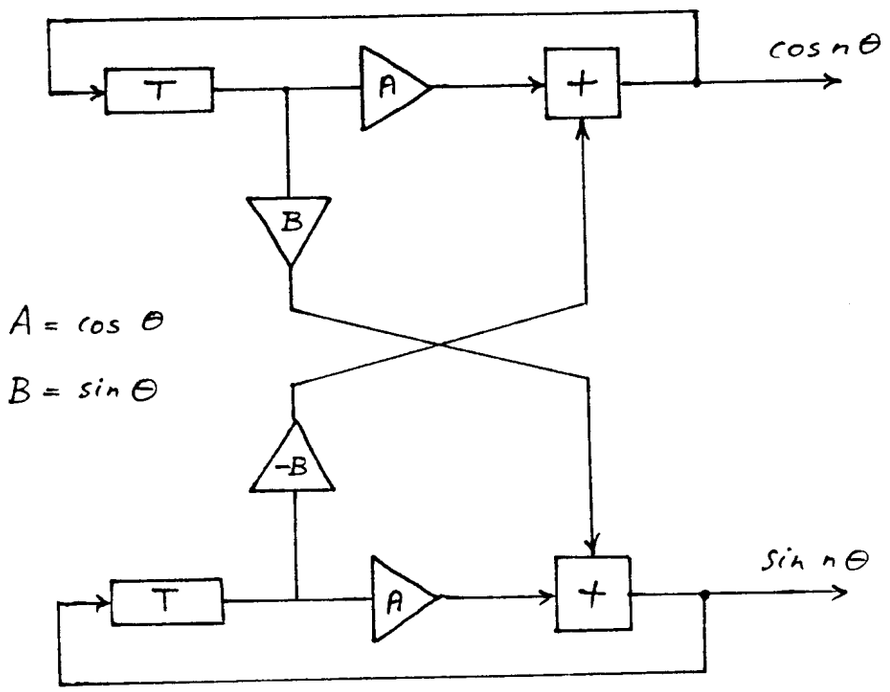


fig.4.11

Sometimes it is possible to choose the sampling frequency $f_s$ and the carrier frequency $f_c$ such that

$$f_c/f_s = \frac{K}{N}$$

with some integers N and K. In that case the carrier $\cos(n\theta_c) = \cos(n.2\pi\ f_c/f_s) = \cos(n.\frac{K}{N}.2\pi)$ is periodic with period N (see section 1.).
It is then possible to store the N values (or less if certain symmetry relations exist) in a memory, for example a ROM. For moderate values of N this requires substantially less hardware than the oscillators discussed above.