1. **The Difference Equation and the Digital Filtering**

**Consider a DSP system in the form of an LTI system as shown in the figure below**

**An LTI System**

**Digital Input**

**Digital Output**

**The relationship between its input and output can be expressed in the form of a *difference equation* as,**

|  |  |
| --- | --- |
|  | **(1)** |

**where and represent the coefficients of the system and is the time index. This equation can also be written as**

|  |  |
| --- | --- |
|  | **(2)** |

**The equation shows that the current value of the output depends on the current and past values of the input as well as the past values of the output.**

**We have already seen that a system expressed in this form of difference equation fulfills the conditions of linearity, time-invariance and causality.**

**If the initial conditions are given, the system output (i.e. time response), , can be obtained recursively (illustrated below by the examples). This process is called *digital filtering*.**

**Example 1**

**Compute the system output**

**for the first four samples using the following initial conditions**

1. **Initial conditions:, and input .**
2. **Zero initial conditions:, and input .**

**Solution**

1. **Setting, and using the initial conditions, we obtain the input and output as**

**Setting, and using the initial conditions, we obtain the input and output as**

**Setting, and using the past values of the input and output,**

**Setting, and using the past values of the input and output,**

**Clearly, it can be seen that the further value of the output can be obtained recursively.**

1. **Setting, and using the initial conditions, we obtain the input and output as**

**Setting, and using the initial conditions, we obtain the input and output as**

**Setting, and using the past values of the input and output,**

**Setting, and using the past values of the input and output,**

**Clearly, it can be seen that the further value of the output can be obtained recursively.**

**Example 2**

**Compute the DSP system output**

**with the initial conditions, and input .**

1. **Compute the system response for 20 samples using MATLAB.**

**Solution**

**A MATLAB program to compute the system response for 20 samples is given below along with the corresponding output shown in graphical form.**

**% Example 2**

**%**

**% Compute the response y(n) of a DSP system expressed by**

**% y(n)=2x(n)-4x(n-1)-0.5y(n-1)-y(n-2)**

**% for the first 20 samples. Initial conditions are**

**% y(-2)=1, y(-1)=0, x(-1)=-1 and the system input is**

**% x(n)=(0.8)^n\*u(n).**

**%**

**% Initialize the input and output vectors**

**xi = [0 -1]; % for n=-2 and n=-1**

**yi = [1 0]; % for n=-2 and n=-1**

**% Compute time indices**

**n = 0:1:19;**

**% Compute the input samples x(n) for these time instants n**

**x = (0.8).^n;**

**% Include the initial values of input into this vector**

**x = [xi x];**

**% Now compute the system response**

**y = []; % an empty vector**

**y = [yi y]; % after including the initial conditions**

**% compute y(n) recursively**

**for k = 3:1:22**

**r = 2\*x(k-2)-4\*x(k-1)-0.5\*y(k-1)-0.5\*y(k-2);**

**y = [y r];**

**end**

**subplot(2,1,1), stem(n,x(3:22),'filled','LineWidth',2), grid on**

**xlabel('Sample number'); ylabel('Input x(n)');**

**subplot(2,1,2), stem(n,y(3:22),'filled','LineWidth',2), grid on**

**xlabel('Sample number'); ylabel('Output x(n)');**



**Figure 2: Plots of the input and system output for Example 2.**

**There are two MATLAB functions (syntax given below), that can used to perform this filtering process:**

**Zi = filtic(B, A, Yi, Xi)**

**y = filter(B, A, x, Zi)**

**where B and A are vectors for the coefficients given as**

**and**

**Xi and Yi are the vectors containing the initial conditions. Also x, y are the input and system output vectors.**

**The function filtic is used to obtain the initial states required by the second function filter. The function filter is based on the *direct-form II realization* to implement a digital filter from its difference equation form. This will be studied in a coming lecture.**

**The following MATLAB code, illustrates how to solve Example 1, using filtic and filter MATLAB functions.**

**>> B = [0 1];**

**>> A = [1 0 -0.5];**

**>> Xi = [-1 0];**

**>> Yi = [0 1];**

**>> Zi = filtic(B, A, Yi, Xi);**

**>> n = 0:3;**

**>> x = (0.5).^n;**

**>> y = filter(B, A, x, Zi)**

**y =**

**-0.5000 1.0000 0.2500 0.7500**

**These are the same results as obtained in Example 1.**

1. **The Difference Equation and the Transfer Function**

**From Equation 1, we have**

|  |  |
| --- | --- |
|  |  |

**Assuming that all initial conditions for this system are zero, we take the z-transform of both sides to get**

|  |  |
| --- | --- |
|  | **(3)** |

**We have made use of the shift-theorem in the above equation. Rearranging, we obtain**

|  |  |
| --- | --- |
|  | **(4)** |

**where is defined as the z-transfer function with its numerator and denominator polynomials given by**

|  |  |
| --- | --- |
|  | **(5)** |
|  | **(6)** |

**It can clearly be notices that the z-tranfer function is the ratio of the z-transform of the output with the z-transform of the input. This can diagrammatically be shown as**

**The z-transfer function can be used to determine the stability and frequency response of the digital filter.**

**Example 3**

**A DSP system is described by the following difference equation**

**Find the z-transfer function, the denominator polynomial, and the numerator polynomial.**

**Solution**

**Taking the z-transform of both sides of the given difference equation, and using the shift-theorem, we get**

**It can also be written as**

**The transfer function, is therefore, given by**

**The denominator and numerator polynomials are**

**Example 4**

**A digital system is described by the following difference equation**

**Find the z-transfer function, the denominator polynomial, and the numerator polynomial.**

**Solution**

**Taking the z-transform of both sides of the given difference equation, and using the shift-theorem, we get**

**It can also be written as**

**The transfer function, is therefore, given by**

**The denominator and numerator polynomials are**

**In some DSP applications, the given transfer function of a digital system can be converted into a difference equation for DSP implementation. The following example illustrates this procedure.**

**Example 5**

**Convert each of the following transfer functions into its difference equation**

**Solution**

**Part (a): We first divide the numerator and denominator by to obtain the transfer function whose numerator and the denominator polynomials have the negative powers of, it follows that**

**According to the definition of the transfer function**

**Therefore, in this case,**

**Cross multiplication gives**

**Applying the inverse z-transform and applying the shift-theorem**

**This equation can be re-arranged to give the required difference equation for the DSP system, as**

**Part (b): In this case also, we first divide the numerator and denominator by to obtain the transfer function whose numerator and the denominator polynomials have the negative powers of, it follows that**

**According to the definition of the transfer function**

**Therefore, in this case,**

**It can be written as**

**Applying the inverse z-transform and applying the shift-theorem**

**This is the required difference equation for the DSP system.**

**Transfer Function in Pole-Zero Form**

**From Equation 4, we know that the transfer function for a digital filter can be written as**

**The numerator and the denominator polynomials of the transfer function can be factorized. The transfer function can therefore, be written in its pole-zero form as**

|  |  |
| --- | --- |
|  | **(7)** |

**where the zeros and poles can be found by solving (finding the roots of) the polynomial equations**

**This is explained with the following example.**

**Example 6**

**Given the following transfer function**

**Convert it into its pole-zero form.**

**Solution**

**We first multiply the numerator and denominator by to obtain the transfer function whose numerator and the denominator polynomials have the positive powers of, as follows**

**Putting the numerator polynomial equal to zero and then finding the roots, gives us the zeros of the transfer function,**

**Therefore, we get and as the roots.**

**Now, setting the denominator polynomial equal to zero and find the roots, gives us the poles of the transfer function,**

**Therefore, the poles are and. The transfer function can now be written in the pole-zero form as**

**Impusle Response, Step Response and System Response**

**Example 6.7**

**Given a transfer function depicting a DSP system**

**Determine**

1. **The impulse response**
2. **The step response, and**
3. **The system response, if the input is given as.**

**Solution**

**Part (a): In this case, thus. As**

**Therefore, in this case, the z-transform of the output is equal to the transfer function:**

**By taking the inverse z-transform of the transfer function we can find out the unit impulse response of the system. The transfer function can be written as**

**This can further be written in the form of partial fractions as**

**where**

**Thus we have**

**Or**

**Taking inverse z-transform of both sides (and using Table 5.1), we get**

**which is the required impulse response of the system.**

**Part (b): In this case, thus. As**

**Therefore, in this case,**

**It can be written as**

**This can further be written in the form of partial fractions as**

**where**

**Thus we have**

**Or**

**Taking inverse z-transform of both sides (and using Table 5.1), we get**

**which is the required step response of the system.**

**Part (c): In this case, thus from Table 5.1,. As**

**Therefore, in this case,**

**It can be written as**

**This can further be written in the form of partial fractions as**

**where**

**Thus we have**

**Or**

**Taking inverse z-transform of both sides (and using Table 5.1), we get**

**which is the required system response.**

**Table 5.1 Table of z-transform pairs (for causal sequences)**

|  |  |  |  |
| --- | --- | --- | --- |
| Line No. | Signal | z-Transform | Region of Convergence |
| 1 |  |  |  |
| 2 |  | **1** | **Entire z-plane** |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| 11 |  |  |  |
| 12 |  |  |  |
| 13 |  |  |  |
| 14 |  |  |  |
| 15 | **where and are complex constants defined by**  **,** |  |  |

**Shift Theorem:**