A. Implementing Difference Equations

In this part, you will write a function named `diff_eq`, which implements the following difference equation:

\[ y[n] = \sum_{k=0}^{M} b_k x[n - k]. \]

This equation can be used to describe the input-output relationship of a group of LTI systems called FIR filters (we will learn more about FIR filters in future labs). \( x[n] \) and \( y[n] \) denote the input and output of the system respectively.

The individual terms in the above summation are in fact delayed (and scaled) versions of \( x[n] \). To implement the delay operator, let us write a second function named `delay` and include it in the same function M-File as `diff_eq`. Such functions are called local functions or sub_functions in MATLAB; they should be placed after the main function. Note that the main function can be called from the command window or other scripts, while local functions are only visible to other functions in the same file.

(1) delay (Local Function)

1. The function should have the following input and output parameters:
   Inputs:
   - \( X \): vector containing samples of the input signal (row vector)
   - \( nd \): number of unit delays (integer and positive)
   Output:
   - \( DX \): delayed version of \( X \) by \( nd \)

2. Error messages should be generated if the inputs do not meet the specified conditions.

3. Assuming that \( X \) contains samples from the right half of a right-sided signal, and the first element in \( X \) is the value of the signal at zero, find the delayed version and store it in \( DX \). Note that \( DX \) must be of the same length as \( X \). Consider the following example:

\[ X = [2 \ 5 \ 1 \ 9 \ 7 \ 4] \quad \text{and} \quad nd = 2 \quad \Rightarrow \quad DX = [0 \ 0 \ 2 \ 5 \ 1 \ 9]. \]
(II) **diff_eq (Main Function)**

1. The function should have the following input and output parameters:
   Inputs:
   - X: vector containing samples of the input signal (row vector)
   - bk: vector containing coefficients of the difference equation (row vector, real)
   Output:
   - Y: vector containing samples of the output signal

2. Error messages should be generated if the inputs do not meet the conditions.
3. The local function `delay` and the difference equation given above should be used to compute the function output Y.

**B. Impulse Response of LTI Systems**

1. Write a function named `impulse_resp` (in a new M-File) that computes the impulse response of an LTI system. The function should have the following input and output parameters:

   Input:
   - Start: starting point of a discrete-time interval (integer)
   - End: ending point of a discrete-time interval (integer, greater than Start)
   - bk: vector containing coefficients of the difference equation (row vector, real)

   Output:
   - h: vector containing samples of the impulse response

2. Error messages should be generated if the inputs do not meet the specified conditions.
3. The function should generate a linear vector of integer numbers ranging from Start to End and store it in a variable named n.
4. The function should generate samples of the unit impulse function over the range of values in n and store them in a variable named delta. The unit impulse function is defined as follows:

   \[
   \delta[n] = \begin{cases} 
   1 & n = 0 \\ 
   0 & n \neq 0 
   \end{cases}
   \]

5. The function `diff_eq` should be called with two inputs delta and bk in order to compute samples of the impulse response, which will be stored in the output variable h.
6. The function should plot the unit impulse function, delta, and the impulse response, h, versus n, in two separate subplots in the same figure window, (two subplots in a column). Since we are dealing with discrete-time functions, it would be better to use the command `stem` instead of the command `plot`. Title the first subplot Unit Impulse Function and the second Impulse Response. Label x-axes n. Label y-axes \(\delta[n]\) and \(h[n]\) respectively.
7. Call the function in the command window for the following input sets:

i. Start = -2, End = 6, bk=[ 5 4 3 2 1 ]
ii. Start = -5, End = 12, bk=[ 1 1 1 -2 -2 -2 1 1 1 ]
iii. Start = -5, End = 12, bk=[ 3 0 0 -6 0 0 6 0 0 -3 ]
iv. Start = -5, End = 12, bk=[ 0.5 2 3.5 5 6.5 5 3.5 2 0.5 ]

Provide in your report all generated plots, as well as a brief discussion of your observations in each case.

Questions

1. Write a function named `step_resp` that computes the step response of an LTI system. The step response is the response of a system to the unit step function, defined as follows:

\[
    u[n] = \begin{cases} 
    1 & n \geq 0 \\
    0 & n < 0 
    \end{cases}
\]

You should be able to create this function by slightly modifying the `impulse_resp` function. Your function should also make plots of the unit step function and the step response. Call the function for the following input sets:

i. Start = -5, End = 40, bk=[ 0 : 0 : 5 : 8 ]
ii. Start = -5, End = 40, bk=[ 8 : 0 : 5 : 0 ]

Include in your report all generated plots, as well as a brief discussion of your observations in each case.

2. Call your `step_resp` function for Start = -5, End = 20, and bk=[ 1 -1 ]. Does the plot of step response look familiar? What discrete-time function does it look like? Using the input-output relationship of the given system (i.e. the difference equation), try to express the function in question in terms of the unit step function.