Lab Project
CSE 432: Digital Signal Processing Lab
Fall 2020

Al Mehdi Saadat Chowdhury

Assigned Date: 14th November, 2020
Due Date for Task-1: 24th November, 2020; 11.58PM
Due Date for Task-2: 3rd December, 2020; 11.58PM
Due Date for Task-3.1: 13th December, 2020; 11.58PM
Due Date for Task-3.2,3.3: Will be announced later
Date for Viva(all tasks): Will be announced later

Basic Instructions:

In this section, you will find a general overview about the project, along with its grading policy.

- **Exam Format** Each project will have three components:
  1. A single Python code file in .py format
  2. A report describing tasks of your project
  3. A viva-voce based on the project and everything taught in the class before the viva date

  Marks Distribution Total points (different than marks) for each task is given alongside each task question. Marks in a task has 4 units - 1 unit in the submitted code with report, 3 units in the viva-voce. For example, if you answer 10 points, then you will receive 2.5 marks from your submitted code with report and rest of the 7.5 marks will be based on your performance in the viva-voce. Note that, the viva-voce will be based on the task question, along with all contents taught in both theory and lab lectures until the viva date.

  What to submit You will submit a .zip file (not .rar, .7z, or separate files) that contains a single code file (.py) and a report file in PDF. Details about each file is given in the "Project Questions" and the "How to write the Report" sections below.

  Submission Deadline Please look at the due date mentioned above.

- All submissions must follow the honor code mentioned in the class, which in summary is: **Do not copy your task codes from any sources including online resources, your classmates, friends, seniors and so on. You are NOT PERMITTED to do group study for this project.** Of course, you can use the class lecture or the textbook or any other textbooks available to you **(except any solution manual)** for solving your project. If you are using any book, mention the name of the book in the acknowledgement section of the project report. Violating the honor code will incur the following penalties:

  1. The project will not be graded. You will loose the entire marks of this project, which means you will fail this course.
  2. In addition to the above penalty, you loose the chance to answer any other task of the project at a later date. You will also be reported for further disciplinary actions to the head of the department.

- Understand that, even 1% copy will be regarded as violation of the honor code. **All parties involved in the process will be penalized**. Also remember that, any suspected work will only be penalized after a thorough discussion with you. This discussion will start by giving you a chance to accept or reject my suspicion. If you actually were involved in this violation in any way and you accept it, then you will be given 25% marks of your answered questions back. If you reject my suspicion, then further investigation will be conducted and you will also face a viva related to the **TOPICS** of the assignment. I kindly request you to obey the honor code so that we both can be saved from this unwanted hassle.

Project Questions

Project Description

This project has **three tasks** with *variable point scheme*. A student can answer **at most one task**. If a student answers more than one task, then the task *with maximum points* will be graded only and the project will incur a **penalty of 10 marks reduction** for each additional submission after the first submission ever made by the student.

Task 1 ($15 + 15 = 30$ points)

Answer both questions:

1. Write a Python function that can take any samples range ($N_1 \leq n \leq N_2$; where N_1 and N_2 will be custom defined) as parameters and produce an unit step sequence ($u[n]$) as output. Your function should both print sequence values in the console and also draw the signal in a well-defined plot.

2. Write a Python function that can take any samples range ($N_1 \leq n \leq N_2$; where N_1 and N_2 will be custom defined) as parameters and produce an unit ramp sequence ($u_r[n]$) as output. Your function should both print sequence values in the console and also draw the signal in a well-defined plot.

Task 2 ($40 + 10 + 10 = 60$ points)

Answer all three questions:

1. Create Python functions for

   - Unit impulse sequence $\delta[n]$
   - Unit step sequence $u[n]$
   - Unit ramp sequence $u_r[n]$
   - Time shifting operation $x[n-k]$ where positive k value denotes signal delay and negative k value denotes signal advance (for example, $u[n-2]$ means a step function delayed by 2 units, whereas $u_r[n+3]$ denotes a ramp function advanced by 3 units)

   Please note that, all of the above functions/sequences should work on the same samples range ($N_1 \leq n \leq N_2$; where N_1 and N_2 will be custom defined). Now, write a Python function that takes two integer numbers from the user where

   - First number is signal type (1 = impulse, 2 = step, 3 = ramp)
   - Second number is shifting amount (positive number denotes signal delay, negative number denotes signal advances, zero means no shifting)

   After taking inputs, your function should use the appropriate function defined above and draws the final signal along with printing its sequence values in console.

2. Edit the Python program you have created in task-2.1 by adding two new Python functions for

- Mirroring operation $x[-n]$. Remember that, a signal can only be mirrored with respect to the zeroth sample. If the signal has no zeroth sample, then it cannot be mirrored.
- Downsampling operation $x[nT]$, where T is an integer number greater than zero.

Again note that, all of the above functions/sequences would work on the same samples range ($N_1 \leq n \leq N_2$; where N_1 and N_2 will be custom defined). Now, write a Python function that takes two integer numbers (or three integer numbers if operation is shifting or downsampling) from the user where

- First number is signal type (1 = impulse, 2 = step, 3 = ramp)
- Second number is operation type (1 = shifting, 2 = mirroring, 3 = downsampling)
- If the second number is 1 or 3 (shifting or downsampling), then take an additional number denoting shifting or downsampling amount, by maintaining all accepted rules of these operations.

After taking inputs, your function should use the appropriate function and the appropriate operation defined above and draws the final signal along with printing its sequence values in console.

3. Edit the Python program you have created in task-2.2 by adding two new Python functions for

- Amplitude Scaling operation $Ax[n]$. Remember that, the amplitude scaling value of 1 means no scaling.
- A single Python function that can be used to either add $x_1[n] + x_2[n]$, subtract $x_1[n] - x_2[n]$ or multiply $x_1[n] * x_2[n]$ two signals.

Now, your task is to write a Python function that can be used to create any complete signal with several components, each working in the same samples range ($N_1 \leq n \leq N_2$). To make this function simpler, we will assume that all components of the signal takes on the same amplitude axis operator (that is, add or subtract or product). For example, your function should be able to print or draw functions such as $(u_r[n-2] * u[n])$ or $(\delta[n] + u_r[-n+1] + \delta[n+1])$ and so on.

To solve this problem, you will take inputs from the user in the following format:

- First you will take four numbers from the user. First number denotes the lower bound of the samples range N_1, second number denotes the upper bound of the samples range N_2, third number denotes the number of signal components (for example, this signal $(\delta[n] + u_r[-n+1] + \delta[n+1])$ has three components), fourth number denotes the amplitude axis operator (1 = add, 2 = subtract, 3 = product) which will remain fixed from the second component of the signal to the last (first component of the signal has no operator in front of it).
- Now for each of the signal component, you will take several numbers from user as input as described below:
  - First of these numbers is the amplitude scaling value (for example, the value of A in $Ax[nT]$)
  - Second number denotes the type of signal (1 = impulse, 2 = step, 3 = ramp)
  - Third number denotes whether the component has any shifting operation (1 = yes, 2 = no)
  - If third number is 1 (that is, the component has shifting), then take the shifting amount now (positive value denotes delay, negative value denotes advance)
  - Fifth number (or fourth if shifting was 2) denotes whether the component has any mirroring operation (1 = yes, 2 = no)
  - Sixth number denotes whether the component has any downsampling operation (1 = yes, 2 = no)
  - If the above number is 1 (that is, the component has downsampling operation), then take the downsampling amount now (only nonzero positive value)

After taking inputs, your function should use the appropriate function and the appropriate operation defined above and draws the final signal along with printing its sequence values in console.

Task 3 (60 + 15 + 15 = 90 points)

Answer all three questions:

1. This first question has two options. You can choose **any one** of them.

   Option 1 You can transfer the total mark you obtained from task-2 here. However, you will loose 10 marks as penalty as mentioned earlier. For example, if you obtain 50 marks from task-2, then you can get 40 marks if you choose to take this option of task 3.

   Option 2 This is a regular question. The description of this question is as follows.

   Create functions for impulse, step, ramp sequence, time shifting, mirroring, downsampling operations, amplitude scaling, add/subtract/product of two signals as described in task-2.

   Now, your task is to write a python function that can be used to create any complete signal with several parts; each part has multiple components; each component working in the same samples range ($N_1 \leq n \leq N_2$); but each part may have same or different samples range ($M_1 \leq m \leq M_2$). For example, your function should be able to print or draw functions such as $[(\delta[n] + u_r[-n+1] + \delta[n-1]) + (u_r[m-2] * u[m])]$ where $-3 \leq n \leq 3$ and $1 \leq m \leq 7$.

   All your inputs will be given in a file named "input.txt". Input values given in the file will be in the following format:

   - First line contains number of test cases. The following information is given for a single test case.
     - Next line contains a number denoting number of parts in the signal (for example, the above described signal has 2 parts). The following information is given for each part of the signal.
       * If this is not the first part of the signal (that is the part $(u_r[m-2] * u[m])$ or any later part of the signal), then this number denotes how this signal part should be combined with any previous parts (1 = add, 2 = subtract, 3 = multiply). **If this is the first part of the signal, then no number will be taken as input here.**
       * Next two line takes two numbers denoting the signal part's lower bound and upper bound respectively (for example, the first part of the above signal has lower bound -3, upper bound 3. The second part of the above signal has lower bound 1, upper bound 7).
       * Next line denotes number of components that each signal part has (for example, the first signal part has 3 components, the second signal part has 2 components). For each signal component, take several numbers as input. The following information is given for each of these signal components.
         · If this is not the first component of any part of the signal, then this number denotes how this signal component should be combined with any previous component of the same signal part (1 = add, 2 = subtract, 3 = multiply). **If this is the first component of any signal part, then no number will be taken as input here.**
         · First of these numbers is the amplitude scaling value (for example, the value of A in $Ax[nT]$)
         · Second number denotes the type of signal (1 = impulse, 2 = step, 3 = ramp)
         · Third number denotes whether the component has any shifting operation (1 = yes, 2 = no)
         · If third number is 1 (that is, the component has shifting), then take the shifting amount now (positive value denotes delay, negative value denotes advance)
         · Fifth number (or fourth if shifting was 2) denotes whether the component has any mirroring operation (1 = yes, 2 = no)
         · Sixth number denotes whether the component has any downsampling operation (1 = yes, 2 = no)
         · If the above number is 1 (that is, the component has downsampling operation), then take the downsampling amount now (only nonzero positive value)

For your convenience, details of an input file is given below for the signal:

$$(\delta[n] - u_r[n+1]) + (u[m]); -2 \leq n \leq 3; -1 \leq m \leq 2 \quad (1)$$

Input	Explanation
1	Number of test case is 1. That is, we are solving for a single signal
2	The signal has 2 parts. First part: $(\delta[n] - u_r[n+1])$, Second part: $u[m]$
-2	First part's lower bound is -2
3	First part's upper bound is 3
2	First part of the signal has 2 components
1	The first component has no amplitude scaling and hence $A = 1$
1	Type of that component is an impulse sequence
2	The impulse has no shifting
2	The impulse has no mirroring
2	The impulse has no downsampling
2	The second component will be subtracted from the first component (impulse)
1	The second component has no amplitude scaling and hence $A = 1$
3	Type of this second component is a ramp sequence
1	The ramp has shifting
-1	Shifting amount for the ramp is -1, that is, it will be advanced 1 sample
2	The ramp has no mirroring
2	The ramp has no downsampling
1	Now we are in the second part. The second part will be added to the first part
-1	The lower bound of the second part is -1
2	The upper bound of the second part is 2
1	This second part has a single component
1	The component has no amplitude scaling and hence $A = 1$
2	Type of this signal component is step sequence
2	The step has no shifting
2	The step has no mirroring
2	The step has no downsampling

After taking inputs, your function should use the appropriate function and the appropriate operation defined above and draws the final signal along with printing its sequence values in console.

2. This question will be published at a later date
3. This question will also be published at a later date

How to write the Report

Your report will have the following components:

1. Semester (Fall 2020), Course code, Course title, Your ID, Name, Email address, Contact Number.
2. Which tasks have you answered? Which subtasks have you answered?
3. What was your test cases? Mention at least three test cases that you have tried in your program (must be different than all the test signals I have mentioned in this file). Write the signal along with input sequences.
4. Do your program produces correct outputs for all test cases that you have tested with? Which test cases produced errors?