Laboratory Test

DSP

Explanations

- There are 10 subjects in all, shown below according to the laboratory they were done in.
- The test will last for 60 minutes.

Subjects

Lab 1

1. Define a vector A with 10 zeros, a matrix B with 4×6 elements equal to 1, and a vector C with odd numbers from 1 la 21
   - Change the third element of A to 5
   - Square all the elements of C, and save the result as a new vector D.
   - Compare element-wise the vectors C and D. Compute how many elements of C are larger than the corresponding elements from D.
2. Define a vector t with 1000 elements uniformly spaced between 0 and 10. Compute and plot $cos(2\pi ft)$, where $f = 0.5$.
3. Plot the signal $sin(2\pi ft + \frac{\pi}{4})$, with $f = 0.2$, for a duration of 3 periods.

Lab 3

1. Create a function `mysys1()` that implements the following system H_1:

$$y[n] = H_1\{x[n]\} = n \cdot x[n] + 5$$

   - the function takes 1 input argument `x` and outputs 1 result vector `y`

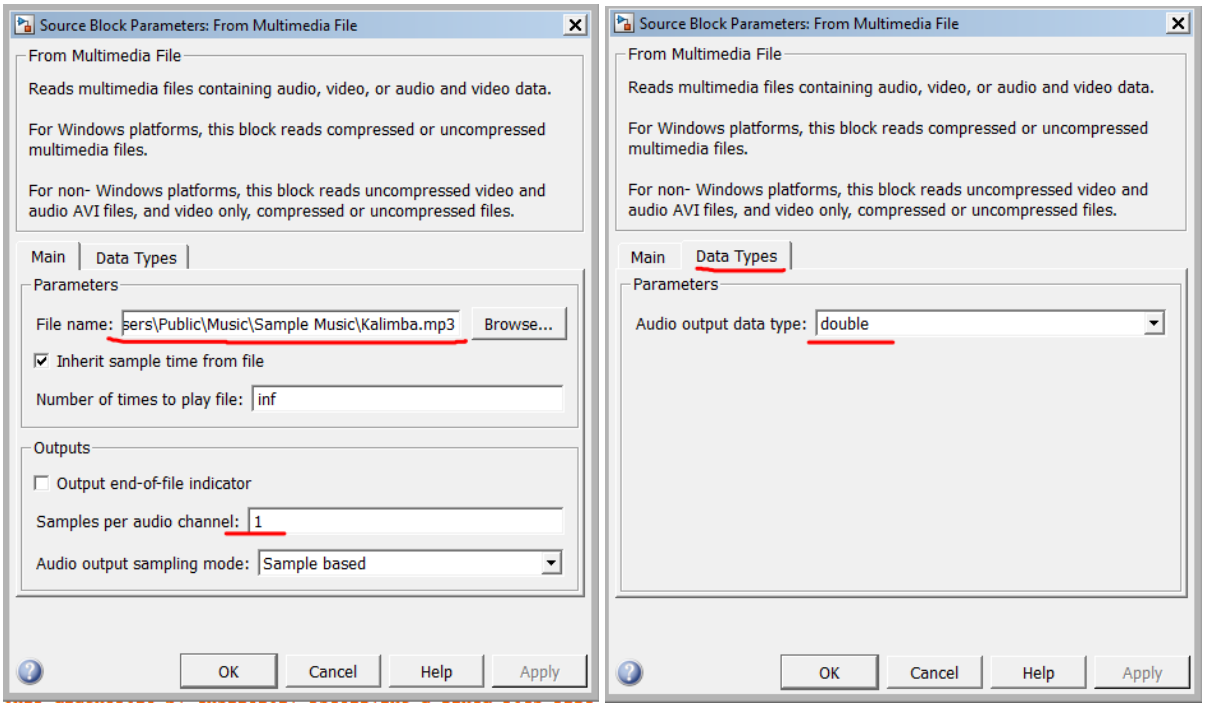
2. Use the function inside a script file
    - generate two random vectors `x` and `y` and two random numbers `a` and `b`
    - apply the function `mysys1()`, separately, to `a*x`, `b*y`, and `a*x + b*y`
    - check if the results verify the linearity equation

Lab 4

1. Create a Simulink model to implement the following system H_1:

$$y[n] = H_1\{x[n]\} = \frac{1}{4}(x[n] + x[n-1] + x[n-2] + x[n-3])$$

    - the system should be implemented as a Subsystem block with one input and one output signal
2. Apply the system to the audio data (mp3 file) loaded with `FromMultimediaFile` block and play the resulting output using `ToAudioSink` block.
    - make sure you set the properties of the `FromMultimediaFile` block as shown below:



Lab 5

1. Create a Simulink model to implement the following system H_1:

$$y[n] = H_1\{x[n]\} = 0.8y[n-1] + 0.25x[n] + 0.1x[n-1]$$

- the system should be implemented as a Subsystem block with one input and one output signal

2. Test linearity of this system as follows:
   - use two random input vectors `x` and `y` (use two *Random* blocks)
   - create three copies of the system inside the model (copy/paste)
   - apply `x`, `y` and `x+y` to the three copies of the system
   - add the outputs of the systems with `x` and `y`, then subtract the output of the copy `x + y`
   - show the resulting signal

Lab 6

1. Implement a Matlab function `y = myconv(x,h)` which implements convolution. The function is given two input vectors and outputs the resulting vector. For two signals $x[n]$ and $h[n]$, the **convolution** operation is defined as

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

2. Use the function to compute the convolution of the sequences `[1 1 1 1 1]` and `[2 2 2]`

Lab 8

1. Generate a 100 samples long signal `x` defined as $x[n] = 0.7\cos(2\pi f_1 n) + 1.2\sin(2\pi f_2 n)$, with $f_1 = 0.05$ and $f_1 = 0.1$.
   a. Plot the signal in the top half of a figure (use `subplot()`).
   b. Compute the Fourier series coefficients and plot their magnitude in the lower half of the figure.

Lab 9

1. Generate a 100 samples long **rectangular** signal `x` defined as $x[n] = [\underbrace{1, 1, ...1}_{50}, \underbrace{0, 0, ...0}_{50}]$.
   a. Plot the signal in the top half of a figure (use `subplot()`).
   b. Compute the Fourier series coefficients with `fft()` and plot their magnitude in the lower half of the figure.

Lab 10

1. Load the `Lena` image (use `imread()`) and display it (use `imshow()`). **Note**: convert the image to a proper grayscale image with the following line of code:

   ```
   I = double(rgb2gray(I));
   ```

2. Apply the system $y[n] = \frac{1}{4}x[n-1] + \frac{2}{4}x[n] + \frac{1}{4}x[n+1]$ to every row and then to every column of the Lena image. (ignore the first and last row/column). Display the resulting image in a new window.

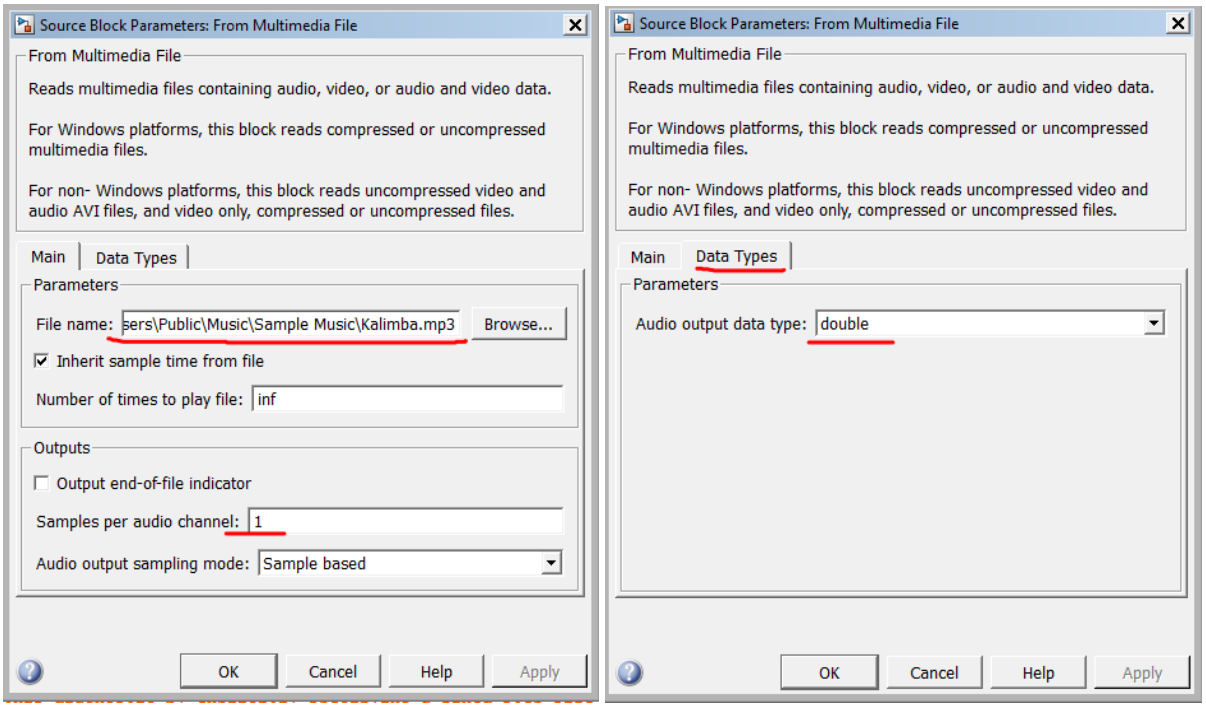
Lab 11

1. Use the Filter Design tool in Matlab (call `fdatool` in command line) to design a Low-Pass filter of order 5, IIR, with cutoff frequency 0.1. Export the coefficients to the Matlab workspace.

2. Generate a signal composed of 30 values of 1 followed by 30 values of 0. Filter the signal with the designed filter (use `filter()`). Plot a figure with 2 subfigures showing the original signal and the filtered signal.

Lab 12

1. Use the Filter Design tool in Matlab (`fdatool`) to design a IIR high-pass filter with order 3, with cutoff frequency 0.1. Implement the filter in Simulink and hear it work on a sample input audio file.
   - use blocks `FromMultimediaFile` and `ToAudioSink`
   - make sure you set the properties of the *From Multimedia File* block as shown below:


Source Block Parameters: From Multimedia File
From Multimedia File
Reads multimedia files containing audio, video, or audio and video data.
For Windows platforms, this block reads compressed or uncompressed multimedia files.
For non- Windows platforms, this block reads uncompressed video and audio AVI files, and video only, compressed or uncompressed files.
Main
Data Types
Parameters
File name: sers\Public\Music\Sample Music\Kalimba.mp3
Browse...
Inherit sample time from file
Number of times to play file: inf
Outputs
Output end-of-file indicator
Samples per audio channel: 1
Audio output sampling mode: Sample based
OK
Cancel
Help
Apply
Source Block Parameters: From Multimedia File
From Multimedia File
Reads multimedia files containing audio, video, or audio and video data.
For Windows platforms, this block reads compressed or uncompressed multimedia files.
For non- Windows platforms, this block reads uncompressed video and audio AVI files, and video only, compressed or uncompressed files.
Main
Data Types
Parameters
Audio output data type: double
OK
Cancel
Help
Apply