Basic filtering in Simulink and Matlab

DSP Lab 07

Table of contents

1 Objective

Students should implement direct filtering techniques in Simulink and Matlab.

2 Theoretical aspects

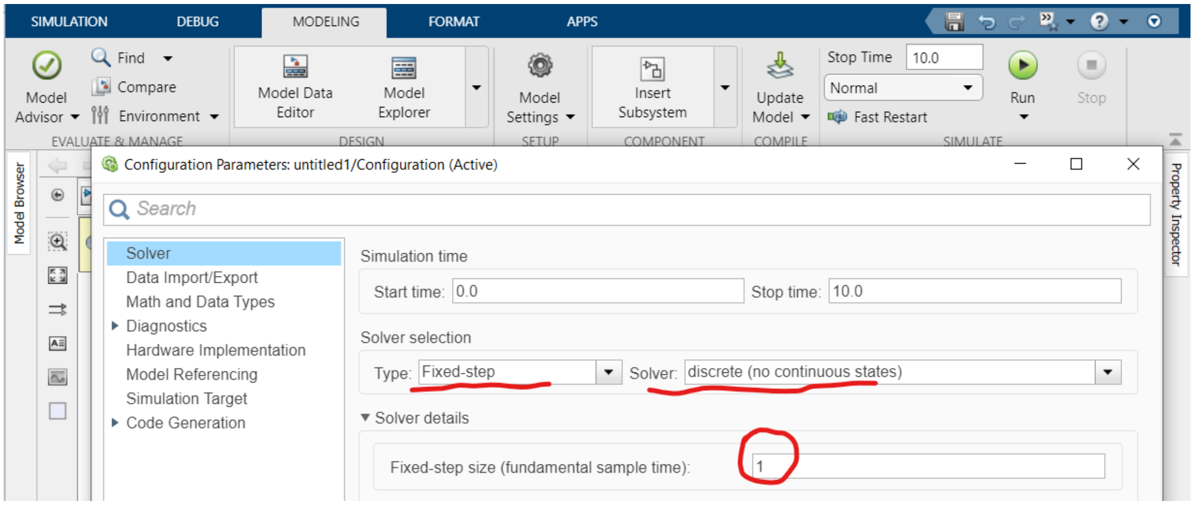
TBD

2.1 Basic Simulink blocks for audio signal processing

Advanced Multimedia blocks from the DSP Toolbox: FromMultimediaFile, AudioDeviceWriter, Buffer

2.2 Settings needed for our models

Running a discrete model requires configuring some settings, as depicted in Figure 1.



Configuration Parameters: untitled1/Configuration (Active)

Simulation time

Start time: 0.0 Stop time: 10.0

Solver selection

Type: Fixed-step Solver: discrete (no continuous states)

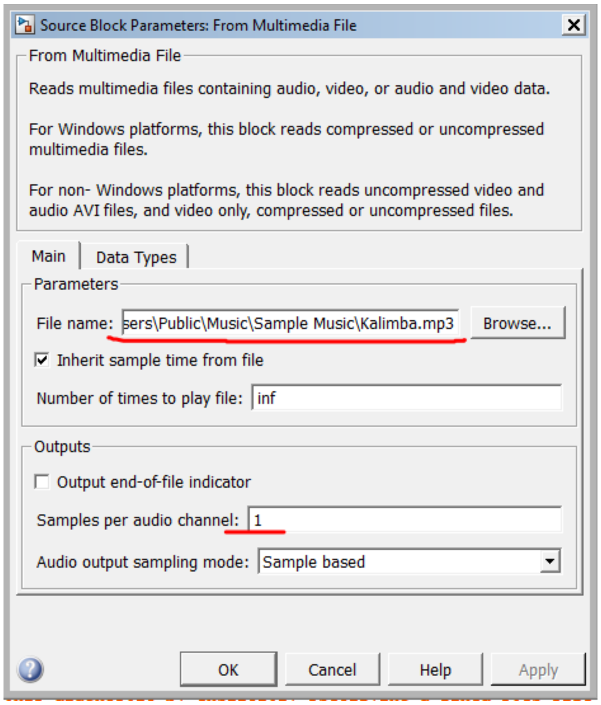
Solver details

Fixed-step size (fundamental sample time): 1

Figure 1: Model settings for discrete models

2.3 Setting needed for the *From Multimedia Device* block

In our work, using the `From Multimedia File` block requires special settings as well:



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.

Parameters

File name: sers\Public\Music\Sample Music\Kalimba.mp3

☑ Inherit sample time from file

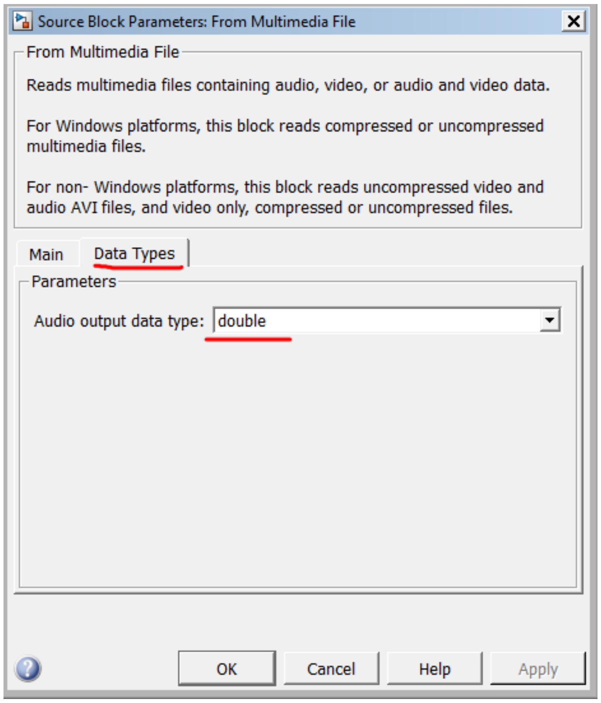
Number of times to play file: inf

Outputs

□ Output end-of-file indicator

Samples per audio channel: 1

Audio output sampling mode: Sample based



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.

Parameters

Audio output data type: double

2.4 Setting needed for the *Buffer* block

The `Buffer` block also needs changing the buffer size to a value like 512 or 1024.

3 Exercises

1. Implement a Simulink model for the following filter. Use it to filter the file `Kalimba.mp3` (use FromMultimediaFile) and play the resulting output (Buffer + AudioDeviceWriter).

$$y[n] = 0.8y[n-1] + \frac{1}{9} \cdot (x[n] + 0.8x[n-1])$$

Make sure you set the properties of the *From Multimedia File* block as shown above.

a. Listen to the original sound and the filtered sound. Is there an audible difference?

b. Plot the input and the output signals

c. What is the system function $H(z)$ of this system?

d. Change the filter to implement the system function $H(z) + 1$. What is this filter doing?

e. Repeat the exercise with the following filter:

$$y[n] = -0.8y[n-1] + \frac{1}{9} \cdot (x[n] - 0.8x[n-1])$$

2. Do the same filtering with Matlab code. Load the same audio file with `audioread()` and use a `for` loop to implement the system equation at every time moment `n`.

   a. Listen to the original sound and the filtered sound (`sound()`)

   b. Plot the input and the output signals

3. In Simulink, check the linearity of these systems by checking if the linearity equation holds:

   - create multiple copies of the system inside the model (copy/paste)
   - use two randomly generated input vectors `x` and `y` (use one of the *Random* blocks), and some two constants `a` and `b`
   - check that the output of the system when the input is `a*x + b*y` is exactly equal to the weighted sum of the outputs applied separately to `x` and `y`

4. Test time-invariance in a similar way

   - the system will be applied to an input vector `x`, and to `x` prepended with a variable number of zeros (i.e. time delayed)
   - the outputs shall be checked if they verify the time invariance equation

5. Find an input signal $x[n]$ to show that the system $y[n] = y[n-1] + x[n]$ is unstable. Show it by simulating the model and displaying the output.

4 Final questions

1. TBD