Convolution

EECS 4462 - Digital Audio

October 28, 2020

Definition

- Convolution is a mathematical operation between two functions that produces a new function. It is defined both in the continuous and the discrete domain (formulas from Wikipedia)
- Continuous

$$(f * g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\,d\tau$$

- Discrete

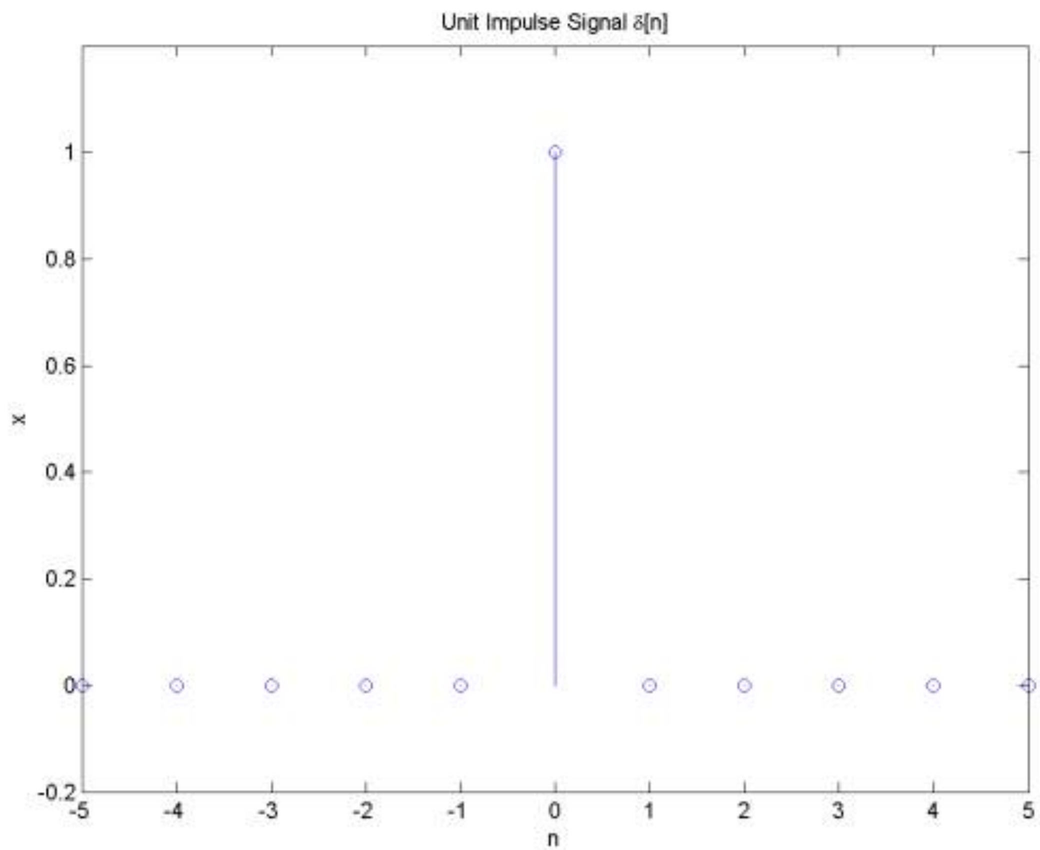
$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]$$

Convolution in digital audio

- In our case, function f is the incoming audio signal to our plugin, and function g describes the operation that our plugin performs
- The convolution of f and g gives us the output signal of our plugin
- Deriving function g is not trivial. When we deal with linear functions, the behaviour of our plugin can be described by its impulse response

Impulse signal

- In the digital world, an impulse signal is a signal that has a value of 1 at time 0, and a value of 0 at all other times



https://zrae.global/2019/01/18/is-the-universe-digital-or-analog/

Impulse response

- The output we obtain when presenting our plugin with an impulse signal is called the plugin's **impulse response**
- For a linear system, we can compute the output of our plugin by adding together the impulse responses for each individual sample of the input signal (multiplied by the magnitude of each sample of course)
- Even if the system is not exactly linear (as in the case of reverb), this process can give acceptable results

Reverb impulse responses

- The reverberation impulse response of a particular room can be captured by recording the response of an impulse, i.e. a loud clap, or the popping of a balloon
- While this is only an approximation of the real impulse response of the room, it can produce very good results when convolved with an input signal
- Impulse responses can be constructed programmatically also

Convolution in JUCE

- JUCE provides a DSP processor that performs the convolution of the input signal with any impulse response
- It can be used in the same way as the processors in the starter code of A3
- Your code will need to load the impulse response from an audio file
- JUCE provides facilities to help with this…

Loading impulse responses

- Check out the **`FileChooser`** class in the JUCE API for a way to create a **`File`** object that points to a file in the file system
- This **`File`** can be used as an argument in the **`loadImpulseResponse`** method of the **`Convolution`** processor

Loading impulse responses

- For pre-loaded impulse responses, your code must find impulse responses on the desktop in a directory called Resources.
- JUCE provides a handy way to access the desktop

```
juce::File::getSpecialLocation(
    juce::File::userDesktopDirectory);
```

- This gives you a **`File`** that corresponds to the user desktop. You can use the **`getChildFile`** method to access the Resources directory and its contents

Additional hints for A3 – part 1

- In the starter code, I created a **`juce::dsp::ProcessContextReplacing`** object to hold the context for the processor
- This means that the input signal is replaced by the output of the processor
- If we need to add the output of the processor to the input signal, use **`juce::dsp::ProcessContextNonReplacing`**
- This will return a separate **`AudioBlock`** that you can add to the input one

Additional hints for A3 – part 2

- For the third reverb, you will need to create an **`AudioBuffer`** that holds your impulse response
- The **`loadImpulseResponse`** method that takes an **`AudioBuffer`** as an argument requires what is called an rvalue in C++ (something that cannot be assigned to)
- To achieve this, provide your buffer as follows

```
std::move (yourBuffer)
```


YORK UNIVERSITÉ UNIVERSITY

Additional hints for A3 – part 3

- Remember never to do anything time-consuming in **`processBlock`**, it gets called many times a second
- In particular, do not call methods that do input/output, or methods that allocate or resize memory
- You can use the DBG macro for debugging, but you have to remove it before you test the audio output. You will hear glitches/artifacts or even no output if you keep it for testing