## C++ Implementation

Pieter F

## Simplifying the Difference Equation

Recall the Simple Moving Average difference equation:

$$y[n] = \frac{1}{N} \sum_{i=0}^{N-1} x[n-i] \tag{1}$$

A naive approach would be to implement the difference equation directly: keeping the last N-1 inputs, and calculate the sum on each iteration, calculating N-1 additions at each time step.

However, we can do much better if we notice how only two terms of the sum change each time:

$$\begin{split} y[n+1] &= \frac{1}{N} \sum_{i=0}^{N-1} x[n+1-i] \\ &= \frac{1}{N} \left( x[n+1] + \sum_{i=1}^{N-1} x[n+1-i] \right) \\ &= \frac{1}{N} \left( x[n+1] + \sum_{i=1}^{N-1} x[n+1-i] + x[n+1-N] - x[n+1-N] \right) \\ &= \frac{1}{N} \left( x[n+1] + \sum_{i=1}^{N} x[n+1-i] - x[n+1-N] \right) \\ &= \frac{1}{N} \left( x[n+1] + \sum_{i=0}^{N-1} x[n-i] - x[n+1-N] \right) \\ &= y[n] + \frac{1}{N} \left( x[n+1] - x[n+1-N] \right) \end{split}$$

We can now define the sum S[n] as follows:

$$egin{aligned} S[n] & riangleq N \cdot y[n] \ &= \sum_{i=0}^{N-1} x[n-i] \ &\Leftrightarrow y[n] = S[n]/N \end{aligned}$$

The difference equation then becomes:

$$S[n+1] = S[n] + x[n+1] - x[n+1-N]$$
(2)

To update the sum, each iteration now requires only one addition and one subtraction, as well as some housekeeping to remember the previous inputs. To get the output y[n], a division by N is needed.

## C++ Implementation

We can now implement Equation 2 directly, and we'll use a rounding division instead of truncating the quotient. Note that this rounding operation is valid for unsigned integer types only.

The previous inputs x[n-i] are kept in a circular buffer.

```
1
      #include <stdint.h>
      template <uint8_t N, class input_t = uint16_t, class sum_t = uint32_t>
 4
5
         public:
           input_t operator()(input_t input) {
                sum -= previousInputs[index];
sum += input;
 7
                 previousInputs[index] = input;
                 if (++index == N)
   index = 0;
return (sum + (N / 2)) / N;
10
11
12
13
14
15
                 static_assert(
                      sum_t(0) < sum_t(-1), // Check that `sum_t` is an unsigned type
"Error: sum data type should be an unsigned integer, otherwise, '
16
17
18
19
20
21
                      "the rounding operation in the return statement is invalid.");
        private:
           uint8 t index
           input_t previousInputs[N]
     };
```