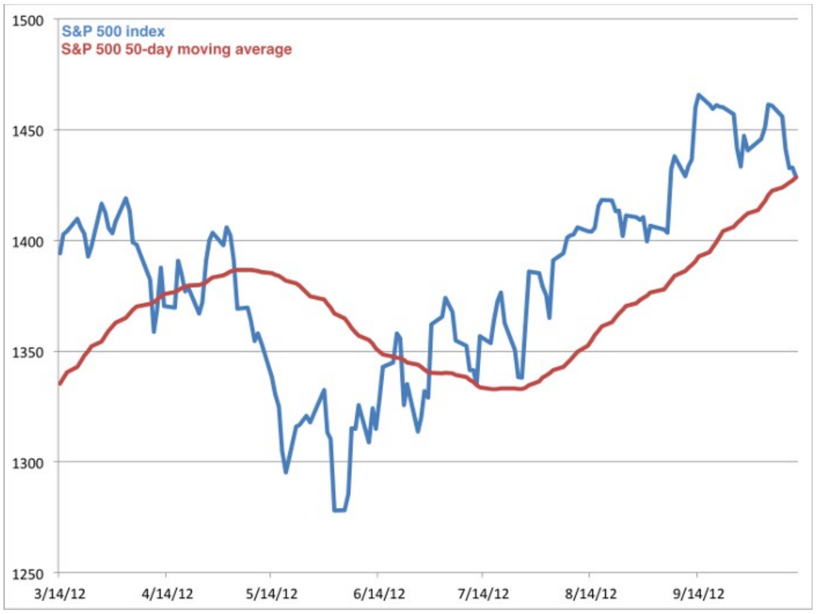
ELT-ESE-3 DSBL

Digital Signal processing practical work

HAN Electrical Engineering/Embedded Systems

E.J Boks MSEE MA

Assignment 2: Implementation of two average values filter methods in a wxWidgets application



Goal:

Write an application that makes it possible to calculate the current average of a series of exchange rates between € and US $. This running average depends on a length over which the average is entered. The average and the original data should be shown in the graph.

Time:

2 weeks.

Requirements:

- Workstation in B1.29/B1.33
- “Lynn & Fürst” theory book.
- WxWidgets online manual at http://www.wxwidgets.org
- A working assignment 1 implementation.

Description:

You must write a graphical application with which a user can easily read a series of data from a file. The data must then be edited with two versions of a running average filter, whereby the user can set in advance how many points the filter will contain. Both the original signal and the running average are then shown in a graphical window.

The two current average methods are:

1. The current average as implemented in assignment 1.
2. A recursive moving average known as the exponentially running average.

Requirements for the program are:

- The program must be written in the C ++ language.
- The wxWidgets toolkit is used as a C++ framework for the application..
- A function must be designed and implemented independently to determine the running average of a non-predefined sequence.

Usage of the wxWidgets toolkit

The application is written with the wxWidgets toolkit. WxWidgets is an open-source cross-platform framework with which complex applications can be written.

In order to be able to carry out the practical work well, the training support classes are presented in two files:

- desktopApp.h. Here are things in order to run the application using wxWidgets. Storage of configuration data, control of multiple programs at the same time etc. are regulated here.
- grafiekVenster.h. It contains routines to simplify drawing on the screen. You must use this class in the main class of the application, GemWaardeVenster.

The assignment

Write the following items as part of the GemWaardeVenster class:

In the event handler

void GemWaardeVenster :: drawDataHandler (wxCommandEvent & event)

you have to draw the graph with the points you have read, and you must calculate and draw the running average here.

- Take into account the amount of points that the running average must contain, the user can enter this via the textctrl.
- Consider applying a fixed offset to facilitate drawing. Also look at the auto scale bar / line functions in graph window.h to make the sign more convenient.
- Implement two methods as mentioned above. Use the wxWidgets API to read out the state of the wxRadioButton:

  const auto choice = filterSelectionRadioBox-> GetSelection ();

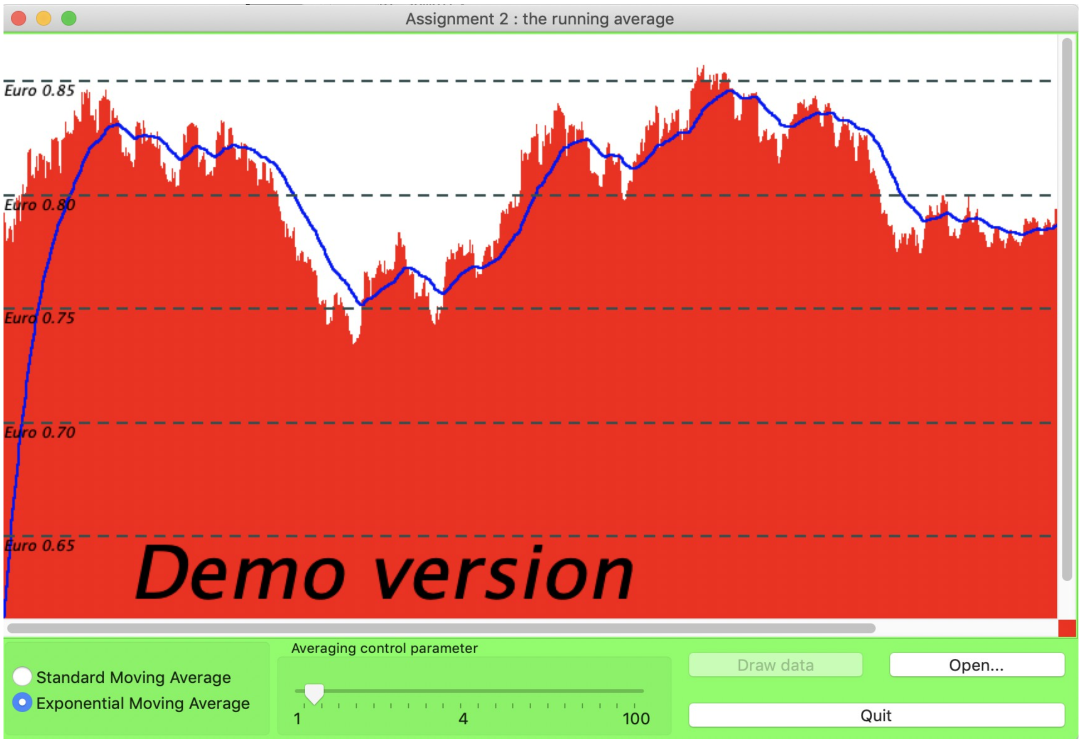
  You should calculate the running average using the RingBuffer class from assignment 1. Implement a RingBuffer from assignment 1 in the drawDataHandler function to calculate the average.

  The exponentially running average must be implemented in the expAverage.cpp file. Then implement an ExponentialAverageFilter class in the drawDataHandler to calculate this average.

Compilation and implementation of the application

To build the application you have to perform the same compilation steps as in assignment 1.

Test the application. The output must provide a similar image:



Assignment 2 : the running average

○ Standard Moving Average
● Exponential Moving Average

When desired, a demo version is available for clarification of the assignment. Ask the teacher for more information.

Delivery:

Show the teacher the working application. Write a small report as indicated in the Practical Work Information document.

Tips

To draw in the graph window, you can use the functions defined in the Graph window, see Graph window.h. Some examples are :

```
void Clean (); / * screen clean * /
void setSignsPen (wxPen &); / * puts pen to draw chart * /
void drawBar (wxPoint &); / * 0 -> coord * /
void drawLine (wxPoint &, wxPoint &); / * start -> end * /
```

These are members of the GraphicsWindow class, so for example:

```
graph->setPen(wxPen (wxColour (wxT ("RED")), 2, wxSOLID));
```

Look in this header file for more functions, for example to draw scaled bars in a window. This is done using the PointList class, a dynamic array of wxPoint elements.

Data storage from the file is done with the wxArrayDouble class. This is a dynamic array with a PointList. See the wxWidgets documentation for an explanation.