

ELT-ESE-3 DSBL

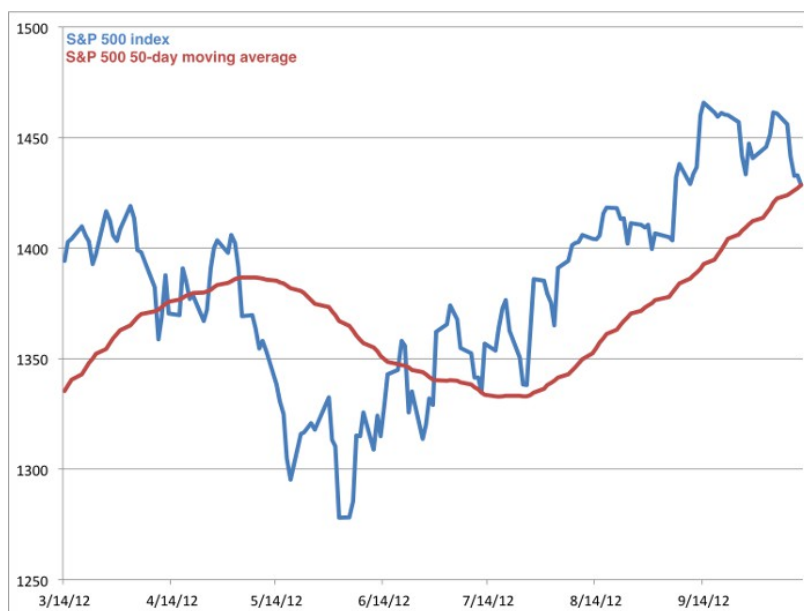


Digitale Signaalbewerking practicum

HAN Elektrotechniek/Embedded Systems

ir drs E.J Boks

Opdracht 2: Implementatie van twee gemiddelde waarden filter methoden in een wxWidgets applicatie



Doel:

Een applicatie te schrijven die het mogelijk maakt om het lopende gemiddelde te berekenen van een reeks wisselkoersen tussen € en US\$. Dit lopend gemiddelde hangt af van een lengte waarover het gemiddelde wordt ingegeven. Het gemiddelde en de originele data dienen te worden afgebeeld in de grafiek.

Tijd:

2 weken.

Benodigde zaken:

- Workstation in B1.29/B1.33 of eigen laptop.
- Theorieboek.
- WxWidgets online handleiding op <http://www.wxwidgets.org>
- Geschreven en geteste klassen uit Opdracht 1.

Beschrijving:

Jullie moeten een grafische applicatie schrijven, waarmee een gebruiker op eenvoudige wijze een reeks data kan inlezen uit een bestand. De data moet dan met twee versies van een lopende gemiddelde filter worden bewerkt, waarbij de gebruiker van te voren kan instellen, hoeveel punten het filter zal bevatten. Zowel het originele signaal als het lopende gemiddelde wordt daarna getoond in een grafisch venster.

De twee lopende gemiddelde methoden zijn:

1. Het lopende gemiddelde zoals geïmplementeerd in opdracht 1.
2. Een recursief lopend gemiddelde bekend als [het exponentieel lopende gemiddelde](#).

Eisen aan het programma zijn :

- Het programma moet worden geschreven in de taal C++.
- Voor de grafische interface wordt de wxWidgets toolkit gebruikt.
- Er moet zelfstandig een functie worden ontworpen en geïmplementeerd die het lopend gemiddelde van een niet van te voren vastgelegde reeks bepaalt.

Gebruik van de wxWidgets toolkit

De applicatie is geschreven met de wxWidgets toolkit. WxWidgets is een open-source cross-platform framework waarmee complexe applicaties kunnen worden geschreven.

Om het practicumwerk goed te kunnen uitvoeren worden door de opleiding ondersteunende klassen in twee bestanden opgevoerd:

- desktopApp.h . Hier staan zaken in om de applicatie te kunnen laten lopen met behulp van wxWidgets. Opslag van configdata, controle op meerdere programma's tegelijkertijd enz worden hier geregeld.
- grafiekVenster.h . Hierin staan routines om het tekenen op het scherm te vereenvoudigen. Je moet deze klasse gebruiken in de hoofdklasse van de applicatie, GemWaardeVenster.

De opdracht

Schrijf als onderdeel van de GemWaardeVenster klasse de volgende zaken:

In de event handler

```
void GemWaardeVenster::drawDataHandler(wxCommandEvent &event)
```

moet je de grafiek tekenen met de punten die je hebt ingelezen, en je moet hier het lopend gemiddelde berekenen en tekenen.

- De data wordt ingelezen met een event handler :

```
(void GemWaardeVenster::dataLadenHandler(wxCommandEvent &event)
```

welke niet verder hoeft te worden uitgewerkt. Let wel op het feit dat de ingelezen data met een offset waarde wordt verlaagd (`static constexpr auto valutaTekeningOffset = 0.6;`)

- De data bevindt zich na het inladen in een `wxArrayDouble` (`wxArrayDouble data;`). Deze array kan worden gebruikt als iedere andere array, maar heeft wel speciale eigenschappen. Lees de online documentatie voor meer info.
- Houd rekening met de hoeveelheid punten die het lopend gemiddelde moet bevatten, de gebruiker kan dit met de `wxSlider* avgValueSlider` ingeven.
- Kijk naar de autoschaal staaf/lijn functies in `grafiekVenster.h` om het tekenen gemakkelijker te maken.
- Implementeer twee methodes zoals hier voor al vermeld. Gebruik de wxWidgets API om de stand van de `wxRadioButton` uit te lezen:

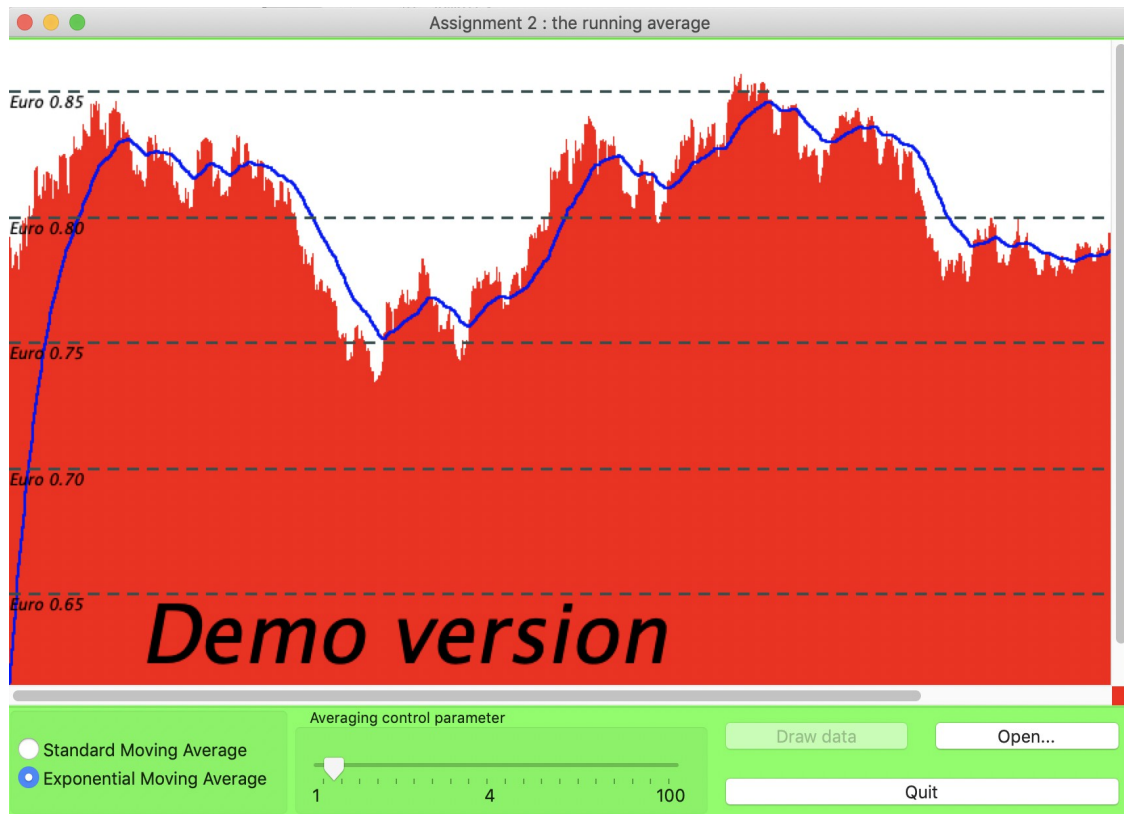
```
const auto keuze = filterSelectionRadioBox->GetSelection();
```

- Het lopend gemiddelde moet je berekenen met behulp van de `RingBuffer` klasse uit opdracht 1. Implementeer een `RingBuffer` uit opdracht 1 in de `drawDataHandler` functie om het gemiddelde te bereken.
- Het exponentieel lopende gemiddelde moet in het bestand `expAverage.cpp` worden geïmplementeerd. Implementeer in de `drawDataHandler` dan een `ExponentialAverageFilter` klasse om dit gemiddelde uit te rekenen.

Compilering en uitvoering van de applicatie

Om de applicatie te bouwen moet je dezelfde compilatiestappen als in opdracht 1 uitvoeren.

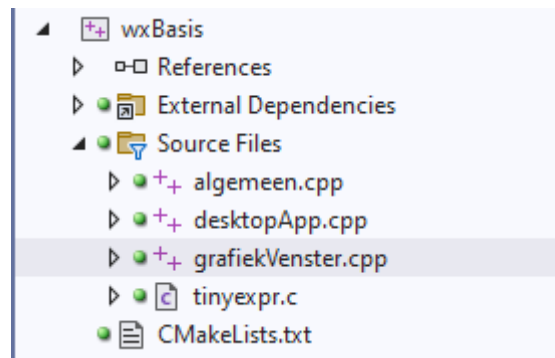
Test de applicatie. De uitvoer moet een soortgelijk beeld opleveren:



Indien gewenst is er een demoversie beschikbaar. Vraag de docent om meer informatie.

Tips

Om te tekenen in het grafiekvenster kun je gebruik maken van de in de klasse GrafiekVenster gedefinieerde functies (zie grafiekVenster.cpp en grafiekVenster.h).



Enkele voorbeelden zijn :

```
void maakSchoon();           /* Maak tekengebied leeg.*/
void zetTekenPen(wxPen &);   /* zet pen om grafiek mee te tekenen */
void tekenStaaf(wxPoint &);  /* 0 --> coord */
void tekenLijn(wxPoint &, wxPoint &); /* begin --> eind */
wxSize geefTekenVeldGrootte(); /* geef de veldgrootte */
```

Deze zijn lid van de GrafiekVenster klasse, dus bijvoorbeeld :

```
grafiek->zetTekenPen( wxPen( wxColour(wxT("RED")), 2, wxSOLID));
```

Kijk in deze header file voor meer functies, bijvoorbeeld om geschaald staven te tekenen in een venster. Dit gebeurt met behulp van de klasse PuntLijst, een dynamische array van wxPoint elementen.

Data opslag vanuit het bestand wordt gedaan met de wxArrayDouble klasse. Dit is een dynamische array voor de opslag van een wisselend aantal double waarden. Zie de [wxWidgets documentatie](#) voor uitleg.

Oplevering

Schrijf een klein verslag, met daarin de volgende zaken:

- broncode applicatie
- screendump van de werkende applicatie

Oplevering geschiedt tezamen met de andere opdrachten. Ga echter niet verder zonder mondelinge goedkeuring van het werk bij de docent.