DSP

Labo-opdracht 3

Moving Average Filter

Ing Patrick Van Houtven
s]

ARTESIS PLANTIJN
HOGESCHOOL ANTWERPEN

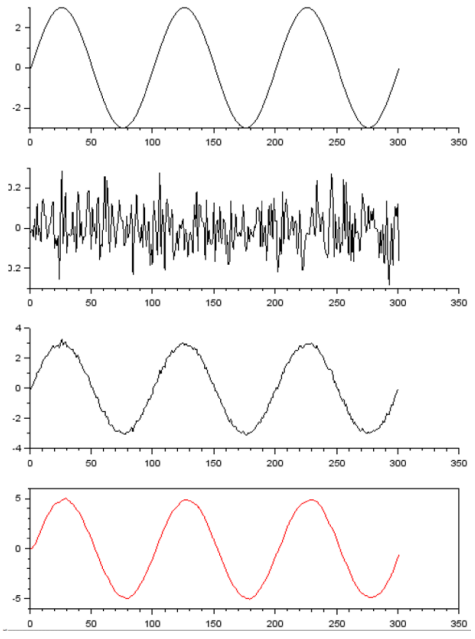
Opdracht 2 : Moving avarage filter

1 Opgave 1

Onderzoek de invloed van het aantal factoren in een moving average filter om ruis van een signaal zoveel mogelijk weg te filteren.

Hoe ga je te werk?

a. Je creëert in scilab een sinussignaal met amplitude 3V en frequentie 100 Hz Je samplet deze sinus met een frequentie van 8 kHz en slaat 3 perioden in samples op in een vector.
b. Je creëert in scilab een ruissignaal met een amplitude rond de 100 mV (pieken kunnen hogere waarden bekomen)
c. Je voegt beide signalen samen en gebruikt dit als ingangssignaal voor de filter
d. Je ontwerpt een moving average filter met 10 factoren en een moving averagefilter met 5 factoren
e. Je filtert het samengevoegde signaal (sinus 100 Hz en ruis) met beide filters en geef het resultaat weer in 1 grafiek. Links de resultaten van de filter met 10 factoren, rechts de resultaten van de filter met 5 factoren. In een voorstelling zoals hieronder is weergegeven voor één filter: (volgorde: sinus, ruis, samengesteld signaal , gefilterd signaal

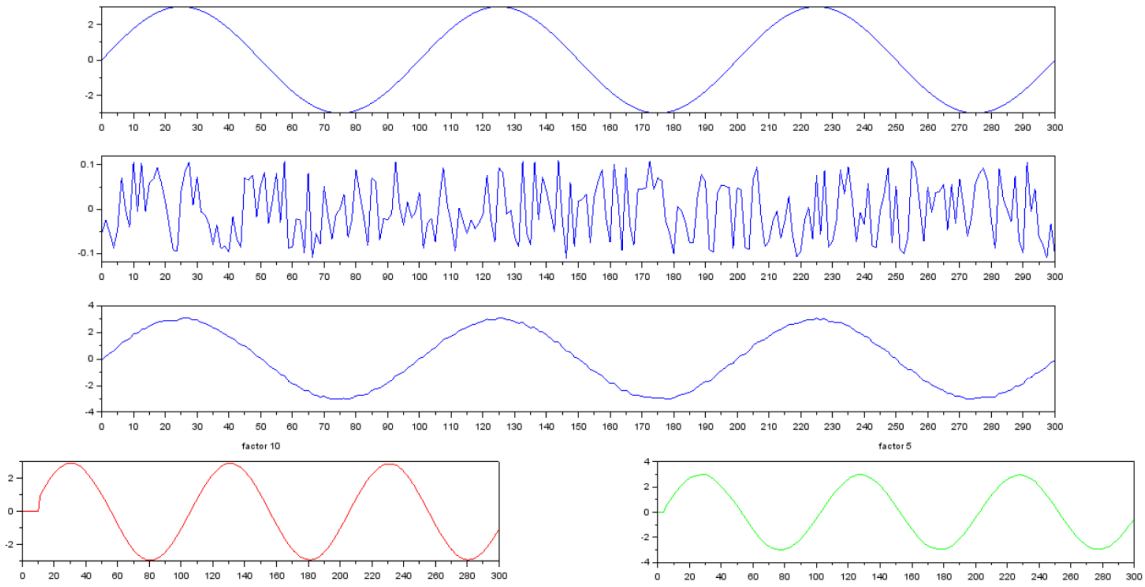


f. Van beide filters maak je de stapresponse zicht baar in 1 grafiek, tezamen met de stap. Creëer een aantal stappen met een gemaakte blokstpanning van 100 Hz.
Vraag: welk van beide filters heeft de beste stapresponse en hoe zie je dat?
g. Geef van beide filters het bodedigram weer in één grafiek met legende.
Vraag: welk van beide filters verdient jouw voorkeur en waarom?

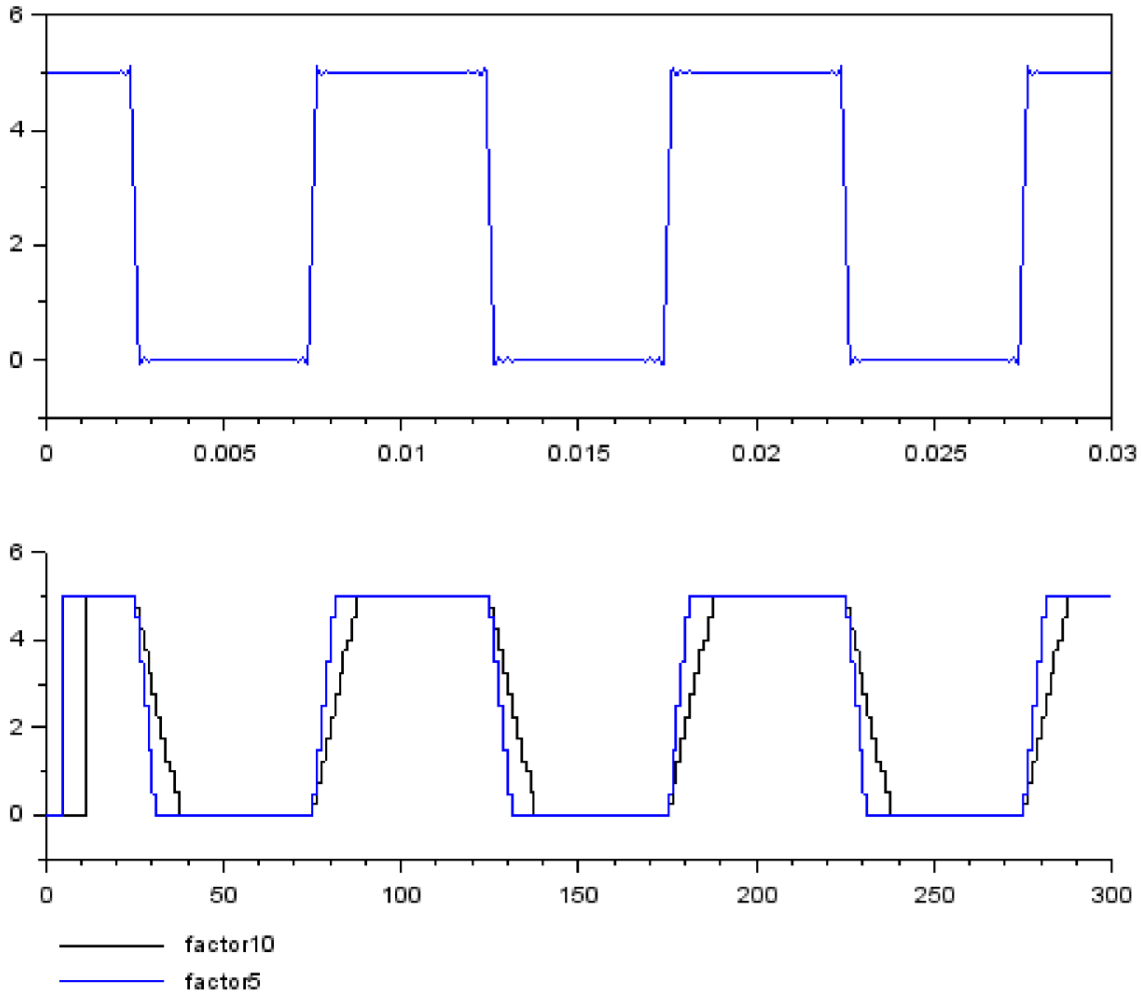
Oplossing:

```
clf
scf(3)
//a)
t8kHz=0.0:(1/8000):(1/100)*3
sig100Hz=sin(2*%pi*100*t8kHz)*3
subplot(511)
plot(t8kHz*10000,sig100Hz)
title("100Hz")
//---------------------------------------------------------------
//b)
sigRan=grand(1,(((1/100)*8000)*3)+1,"uin",0,220)-110
sigRan=sigRan/1000
subplot(512)
plot(t8kHz*10000,sigRan)
title("ruis")
//---------------------------------------------------------------
//c)
siging=sigRan+sig100Hz
//---------------------------------------------------------------
//d)
filter10=zeros (length(siging),1)
subplot(513)
plot(t8kHz*10000,siging)
title("100Hz + ruis")

for i=10:length(siging)
   filter10(i)= ...
   0.1*siging(i-9)+...
   0.1*siging(i-9)+...
   0.1*siging(i-7)+...
   0.1*siging(i-6)+...
   0.1*siging(i-5)+...
   0.1*siging(i-4)+...
   0.1*siging(i-3)+...
   0.1*siging(i-2)+...
   0.1*siging(i-1)+...
   0.1*siging(i)
end
filter5=zeros (length(siging),1)
subplot(513)
plot(t8kHz*10000,siging)

for i=5:length(siging)
   filter5(i)= ...
   0.2*siging(i-4)+...
   0.2*siging(i-3)+...
   0.2*siging(i-2)+...
   0.2*siging(i-1)+...
   0.2*siging(i)
end
```

```
//-----------------------------------------------------------------
//e)
subplot(527)
plot(t8kHz*10000,filter10,'r')
title("factor 10")

subplot(528)
plot(t8kHz*10000,filter5,'g')
title("factor 5")
//-----------------------------------------------------------------
//f)
scf(1)
clf
dc=0.50
t=0.0:(1/8000):(1/100)*3
f=100
Umax=5
w=2*%pi*f*t
harm = 0
for i=1:200
  harm=harm +((sin(i*%pi*dc))/(i*%pi*dc))*cos(i*w)
end
y=(Umax*dc+2*Umax*dc*harm)
subplot(211)
plot(t,y)

for i=10:length(y)
  filter10(i)= ...
  0.1*y(i-9)+...
  0.1*y(i-9)+...
  0.1*y(i-7)+...
  0.1*y(i-6)+...
  0.1*y(i-5)+...
  0.1*y(i-4)+...
  0.1*y(i-3)+...
  0.1*y(i-2)+...
  0.1*y(i-1)+...
  0.1*y(i)
end
for i=5:length(y)
  filter5(i)= ...
  0.2*y(i-4)+...
  0.2*y(i-3)+...
  0.2*y(i-2)+...
  0.2*y(i-1)+...
  0.2*y(i)
end

subplot(212)
plot2d2(t*10000,[filter10 filter5],[1,2],leg="factor10@factor5")

//-----------------------------------------------------------------
//g)
scf(2)

param=[1/10,1/10,1/10,1/10,1/10,1/10,1/10,1/10,1/10,1/10]
transp10=poly(param,'z','coeff')
transh10=horner(transp10,(1/%z))
transsys10=syslin('d',transh10)

param=[1/5,1/5,1/5,1/5,1/5]
transp5=poly(param,'z','coeff')
transh5=horner(transp5,(1/%z))
transsys5=syslin('d',transh5)

bode([transsys10; transsys5],["10";"5"])
```

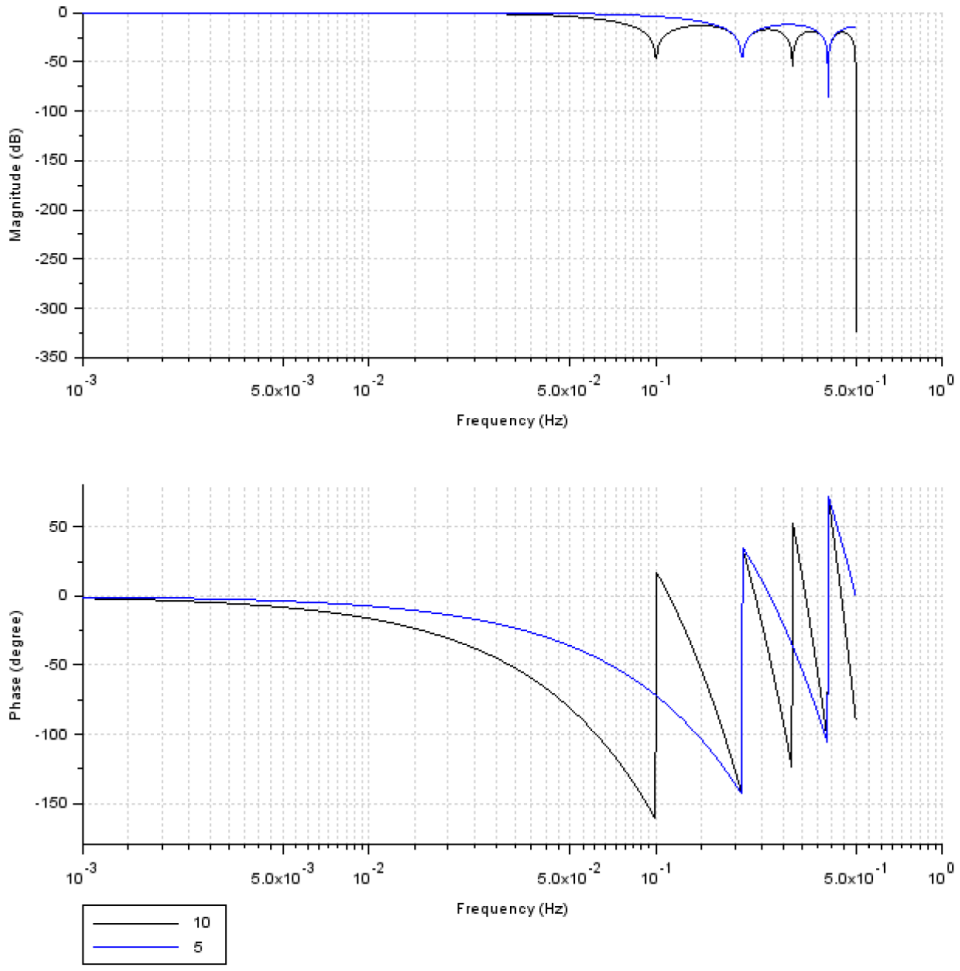


Antwoord deelvraag f:



De moving avarage filter met 5 factoren volgt het best de blokspanning. De filter met 10 factoren vervormt het signaal sterker en heeft een grotere tijdsvertraging

Antwoord deelvraag g:



Als je naar de faseverschuiving kijkt, dan heeft de filter met 5 factoren een stabieler verloop dan deze met 10 factoren. In het doorlaatgebied heeft de filter met 5 factoren een betere lineariteit dan deze met 10 factoren.

2 Opgave 2

Maak met een arduino nano en een IR-afstandssensor een afstandsmeter welke de afstand van een voorwerp kan bepalen tussen dit voorwerp en de sensor. De minimum detecteerbare afstand is 5 cm en de maximaal detecteerbare afstand is 30 cm. (opmerking door gebruik van een grafiek spanning IR-sensor in functie van 1/(afstand sensor ten opzichte voorwerp) bekom je een min of meer lineair verband tussen spanning van sensor en afstand. Gebruik een 5-punts moving avarage filter om stoorinvloeden (lees verkeerde afstandswaarden die door de sensor worden doorgegeven ten gevolge van IR-stoornissen) zoveel mogelijk te verwijderen. Om een meer betrouwbaar afstandcijfer te bekomen neem je het gemiddelde van 20 metingen na elkaar.

Breng de afstanden zichtbaar naar de console toe (zowel de ongefilterde afstandswaarden als de gefilterde zo kan je een idee vormen wat de filter doet met de meetresultaten). Als ongefilterde waarde neem je de waarde telkens aan het begin van de 20 metingen

Ter controle van de (goede) werking van de afstandsmeter kies je vier verschillende afstanden waar je het voorwerp plaatst om de afstand te detecteren. De precieze afstand waar het voorwerp zich bevindt moet gekend zijn zodat je een oordeel kan vormen over de nauwkeurigheid van je

schakeling. Kies deze afstanden over het meetbereik van je sensor: 10 cm, 18,5 cm, 20 cm, 25,7 cm.

Gevraagd:

- Maak een verslag over je realisatie in pdf-vorm. Hierin zijn volgende zaken inbegrepen:
  - De opgave
  - De code voor de afstandssensor (uitgebreid gedocumenteerd)
  - Uitleg via een flow/stroom-diagram hoe je code werkt
  - De hardwareschakeling (bv met fritzing)
  - Meetresultaten (zowel gefilterde als ongefilterde resultaten)
  - Je besluit aangaande de afstandsmeter:
    - Is de schakeling bruikbaar als afstandsmeter? (ja/neen + verklaring van je besluit)
    - Tussen welke minimum afstand en maximum afstand is jou afstandsmeter bruikbaar?
    - Meet je afstandsmeter de afstand tussen eender welk voorwerp goed of moeten de voorwerpen aan bepaalde eisen voldoen om een goede meting te waarborgen?
    - Is de afstandsmeter precies (meerdere metingen na elkaar geven dezelfde waarde weer) (ja/neen + waarom)
    - Is de afstandsmeter nauwkeurig? (afwijking meetresultaat (door sensor gemeten afstand) ten opzichte van de werkelijke afstand (zowel vergelijk met filteresultaat als zonder filterresultaat)?
    - Heeft de filter invloed op de nauwkeurigheid van je meetresultaat? Verklaar je antwoord

Oplossing (dank aan Fernando Declerq)

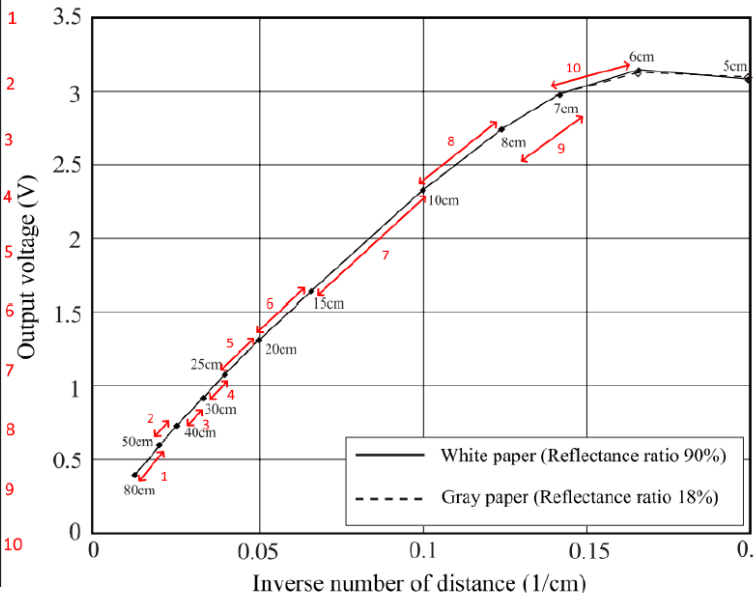
```
String result = "";

void setup() {
  Serial.begin(9600);
}

void loop() {
  mvgAvgFilter(20, A0, "cm");
  //mvgAvgFilter(double aantal_punten, int inSignal, String format)
  //aantal_punten: bepaalt de factor van de filter. Dit mag 1 tot de gewenste factor zijn.
  //inSignal: is een van de analoge ingangen van de Arduino
  //format: mag "cm", "volts" of "raw" zijn. Als de parameter iets anders is dan deze,
  //wordt het resultaat per defect de "raw" resultaat van de ADC.
  Serial.println(result);

  delay(100);
}

void mvgAvgFilter(double aantalPunten, int inSignal, String format) {
  double bufferResult = 0;
  int tempResult = 0;
  format.toLowerCase(); //in het geval dat de gebruiker een rare string zou meegeven voor "format"

  if (aantalPunten <= 0) { // nachecken of "aantalPunten" 0 of kleiner dan 0 is
    aantalPunten = 1;
  }

  for (int i = 0; i < aantalPunten; i++) {
    bufferResult += (analogRead(inSignal)) * (1 / aantalPunten);
    //het aantal punten stelt de maximale keren die deze for-loop zal werken.
    //in de variabele "bufferResult" bewaar ik telkens de som van het product van...
    //... de ingelezen signaal(via ADC) en de factor van de filter.
  }
```

```
if (format == "cm") {
    // ADC-resultaeat omvormen in "cm".
    // Om de juiste waarden te bepalen,
    // heb ik gebruik gemaakt van de karateristiekscurve die
    //in de datasheet van de IR-sensor staat.
    if (bufferResult >= 83 && bufferResult <= 125) {
      tempResult = map(bufferResult, 83, 125, 80, 50);
    }
    else if (bufferResult >= 126 && bufferResult <= 156) {
      tempResult = map(bufferResult, 126, 156, 50, 40);
    }
    else if (bufferResult >= 157 && bufferResult <= 188) {
      tempResult = map(bufferResult, 157, 188, 40, 30);
    }
    else if (bufferResult >= 189 && bufferResult <= 229) {
      tempResult = map(bufferResult, 189, 229, 30, 25);
    }
    else if (bufferResult >= 230 && bufferResult <= 271) {
      tempResult = map(bufferResult, 230, 271, 25, 20);
    }
    else if (bufferResult >= 272 && bufferResult <= 333) {
      tempResult = map(bufferResult, 272, 333, 20, 15);
    }
    else if (bufferResult >= 334 && bufferResult <= 479) {
      tempResult = map(bufferResult, 334, 479, 15, 10);
    }
    else if (bufferResult >= 480 && bufferResult <= 573) {
      tempResult = map(bufferResult, 480, 573, 10, 8);
    }
    else if (bufferResult >= 574 && bufferResult <= 625) {
      tempResult = map(bufferResult, 574, 625, 8, 7);
    }
    else if (bufferResult >= 626 && bufferResult <= 646) {
      tempResult = map(bufferResult, 626, 646, 7, 6);
    }
    //toevoegen van "cm" tekst aan het resultaat.
    result = String(tempResult) + " cm";
  }
```

```
else if (format == "volts") {
  result = String(bufferResult * 0.0048F) + " volts";
  //Als de "format"-variabele gelijk is aan "volts",
  // dan wordt de waarde in de bufferResult omgevormd in een spanningswaarde.
  //De ADC heeft een resolutie van 10-bits,
  // m.a.w. zijn er 1024 stappen van 0 tot 5V(Vcc).
  //Dus: 5V / 1024 = 0.0048mV.
}
else {
  result = bufferResult;
  //als de "format"-variabele geen van de bovenstaande waarden heeft,
  // krijgen wij het "raw"-resultaat van de ADC.
}
```

```
if (bufferResult >= 83 && bufferResult <= 125) {
  tempResult = map(bufferResult, 83, 125, 80, 50);
}
else if (bufferResult >= 126 && bufferResult <= 156) {
  tempResult = map(bufferResult, 126, 156, 50, 40);
}
else if (bufferResult >= 157 && bufferResult <= 188) {
  tempResult = map(bufferResult, 157, 188, 40, 30);
}
else if (bufferResult >= 189 && bufferResult <= 229) {
  tempResult = map(bufferResult, 189, 229, 30, 25);
}
else if (bufferResult >= 230 && bufferResult <= 271) {
  tempResult = map(bufferResult, 230, 271, 25, 20);
}
else if (bufferResult >= 272 && bufferResult <= 333) {
  tempResult = map(bufferResult, 272, 333, 20, 15);
}
else if (bufferResult >= 334 && bufferResult <= 479) {
  tempResult = map(bufferResult, 334, 479, 15, 10);
}
else if (bufferResult >= 480 && bufferResult <= 573) {
  tempResult = map(bufferResult, 480, 573, 10, 8);
}
else if (bufferResult >= 574 && bufferResult <= 625) {
  tempResult = map(bufferResult, 574, 625, 8, 7);
}
else if (bufferResult >= 626 && bufferResult <= 646) {
  tempResult = map(bufferResult, 626, 646, 7, 6);
}
```

1 2 3 4 5 6 7 8 9 10

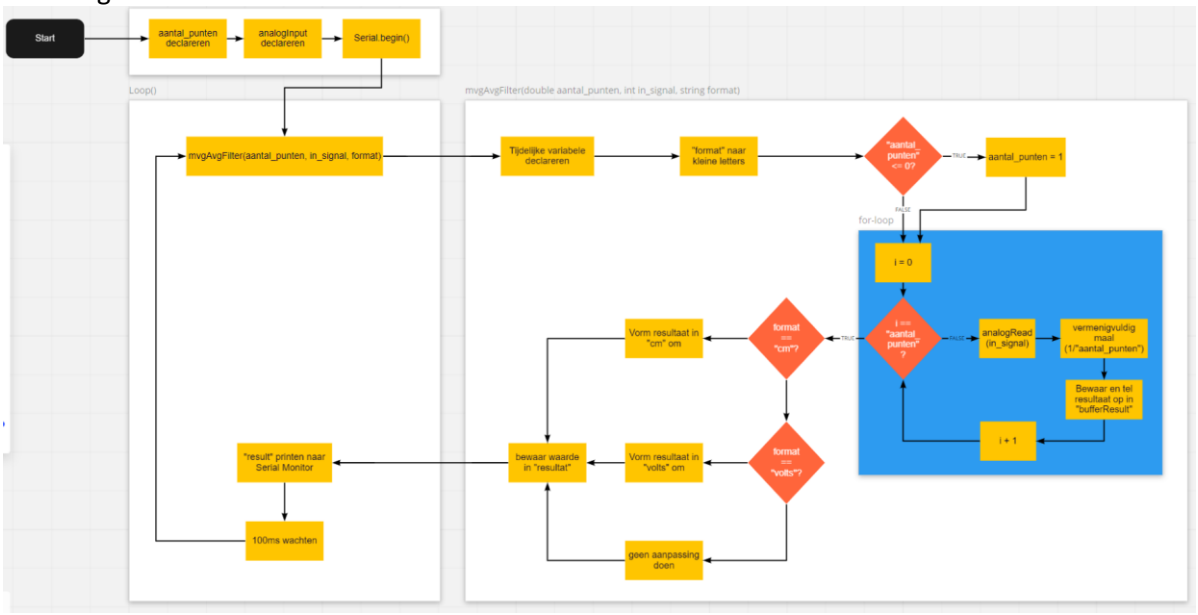


Om al deze if-statements te bekomen zijn eerst handmatig al deze waardes van bovenstaande grafiek genoteerd in een tabel:

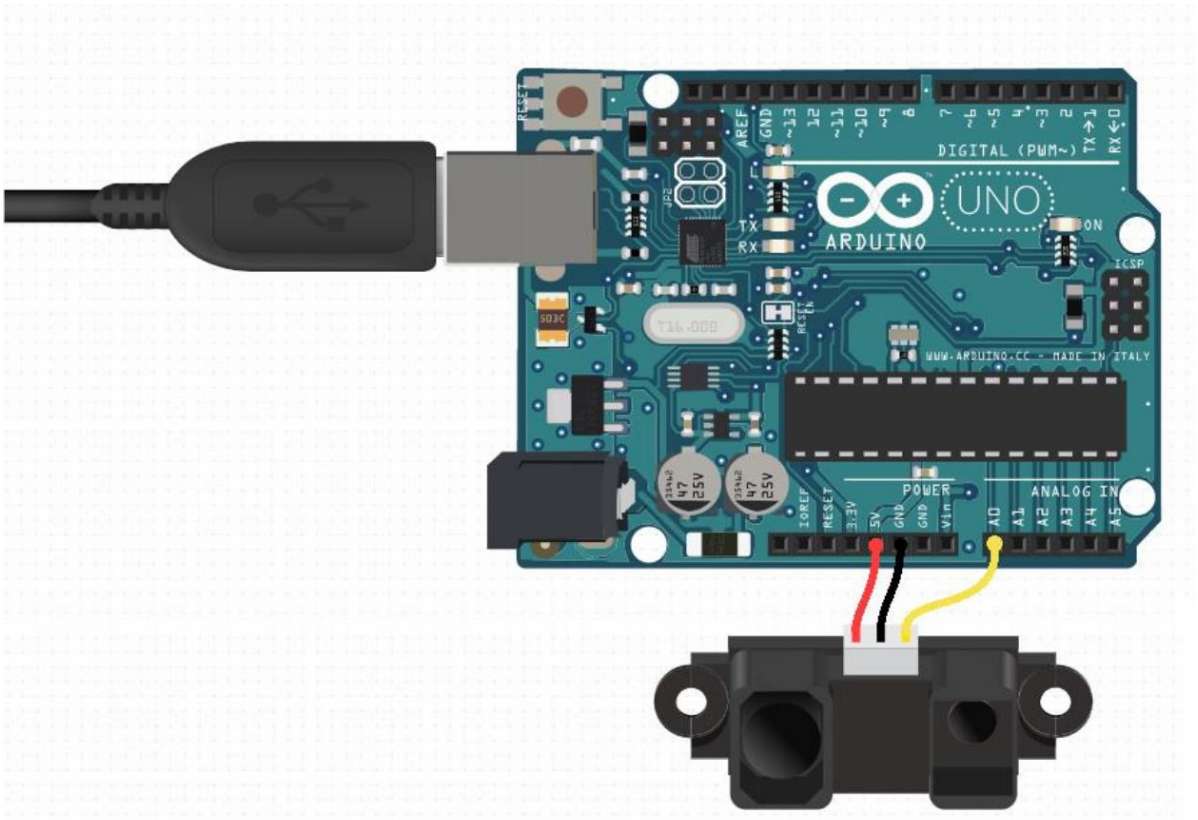
Afstand	Spanning(Vout)	ADC-waarde(Vref=5V)
80cm – 50cm	0.4V – 0.6V	83 – 125
50cm – 40cm	06V – 0.75V	125 – 156
40cm – 30cm	0.75V – 0.9V	156 – 188
30cm – 25cm	0.9V – 1.1V	188 – 229
25cm – 20cm	1.1V – 1.3V	229 – 271
20cm – 15cm	1.3V – 1.6V	271 – 333
15cm – 10cm	1.6V – 2.3V	333 – 479
10cm – 8cm	2.3V – 2.75V	479 – 573
8cm – 7cm	2.75V – 3V	573 – 625
7cm – 6cm	3V – 3.1V	625 - 646

De spanningswaarde tussen 2 punten in de grafiek geven min of meer een lineair verhoging in spanning naargelang de afstand van de ene punt naar de andere een transitie maakt.
Vandaar dat er gekozen is om de functie “map()” in de code te gebruiken. Alhoewel “map()” een integer als return-waarde heeft, is deze waarde bruikbaar voor deze applicatie.
Zowel in de tabel als in de code ontbreken de waarden 6cm en 5cm. De reden hiervoor is dat de spanningswaarde terug begint te zakken waardoor de sketch in een oneindig lus zou terechtkomen.

Flowdiagram:

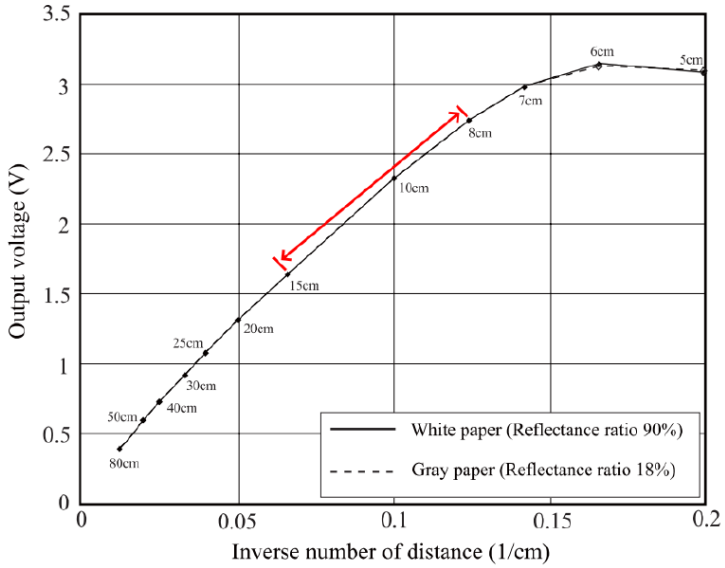


Hardwareschakeling:



De hardwareschakeling is met de figuren uit circuit.io opgebouwd (enkel afbeeldingen van Arduino Uno en de Sharp-sensor zijn hiervan gebruikt). Vervolgens is met behulp van paint.net de afzonderlijke figuren samengebracht en de nodige verbindingen getekend..
Op de datasheet van de sensor wordt aanbevolen om een ontkoppelingscondensator van 10µF of meer te gebruiken tussen Vcc en massa van de sensor. Deze is rechtstreeks gesoldeerd over deze 2 pinnen.

Gekozen afstanden:



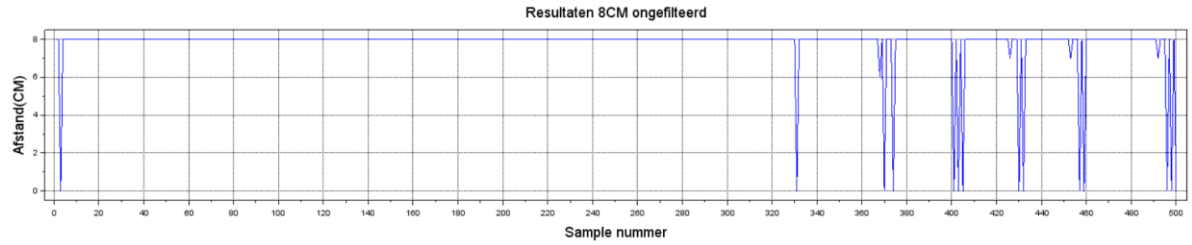
Er is voor de volgende vaste afstanden gekozen:

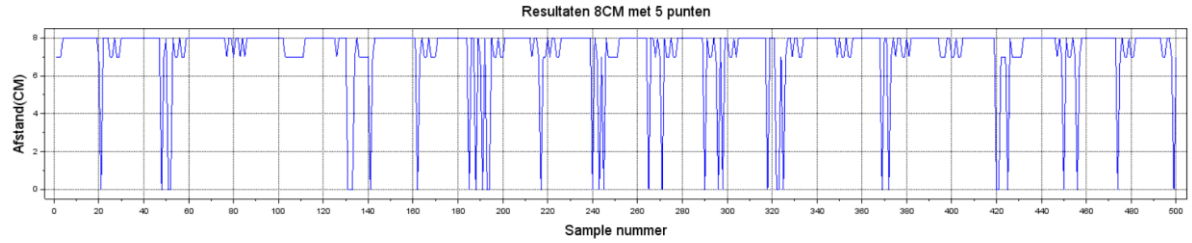
- 8cm
- 10cm
- 13cm
- 15cm

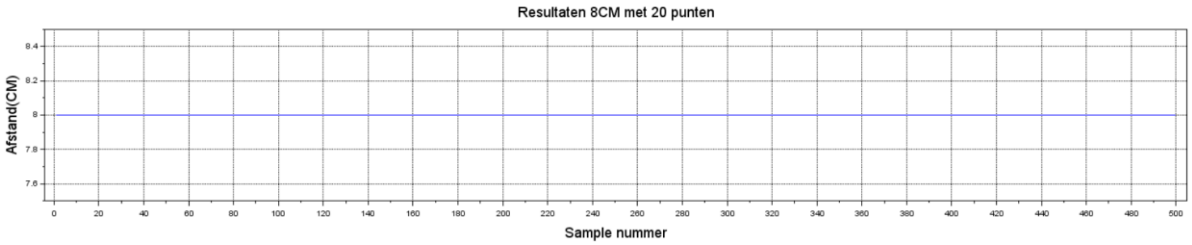
Deze afstanden zijn gekozen om volgende redenen:

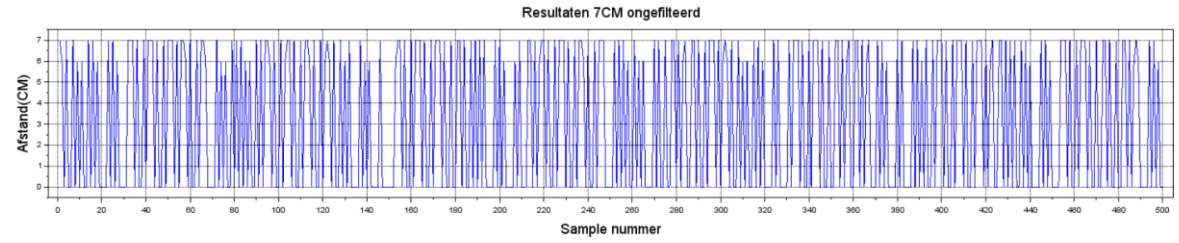
- De spanning verandert vrij lineair in functie van de afstand.
- Tussen bv. een afstand tussen 15cm en 10cm (een verandering van 5cm), is er een verandering van bijna 0.7V . Dit betekent dat er meer tussenstapjes (146 stapjes van de ADC) zijn t.o.v. de tussenstapjes die mogelijk zijn als de afstandswaarden helemaal aan de linkse kant van de karakteristiek zouden gemeten worden. Als de ADC-waarden tussen 6cm en 5cm meet gebruikt worden, dan wordt de code om dit te realiseren heel wat complexer vermits de spanning daar terug begint te dalen.
-

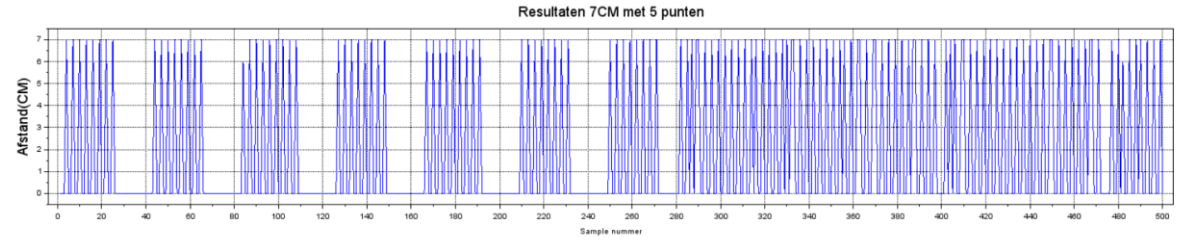
Meetresultaten:

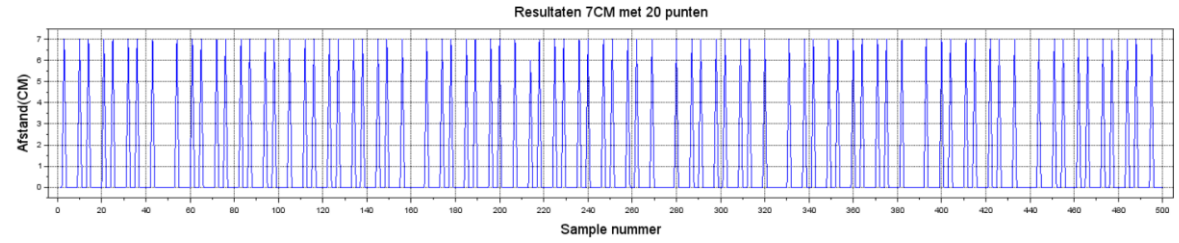












Besluit

Na het uitvoeren van al de metingen voor verschillende afstanden, kan geconcludeerd worden dat het mogelijk is om deze sensor als een goedkope afstandsmeter te gebruiken.
Afhankelijk van de toepassing waarin deze als z'n afstandsmeter gaat fungeren, kan deze sensor deftige meetresultaten opleveren.
Afstanden vermeld op de datasheet van de sensor (van 5cm tot 80cm) zijn volgens echter niet meetbaar in alle situaties van de omgevingsruimte waarin de schakeling zich bevindt.

Afstanden van 7 cm tot 31 cm kunnen min of meer accuraat worden gemeten als de sensor vóór een wit object staat. Het object moet ook loodrecht staan op de sensor.
De sensor kan precieze resultaten opleveren binnen afstanden van 8cm tot 15cm omdat het aantal stappen nodig tussen elke centimeter groter is in dit gedeelte van de karakteristiekscurve van de sensor.
Deze sensor is ook heel nauwkeurig binnen de bovenvermelde afstanden omwille van dezelfde redenen. Zonder filter zijn er uiteraard samples die niet overeenkomen met de reële afstand ten gevolge door storingen (licht van andere bronnen). Als men een filter gebruikt met 10 factoren of hoger, zijn 90% van alle metingen heel nauwkeurig.
Een filter met een veel punten kan dus de nauwkeurigheid van deze sensor sterk beïnvloeden vermits er meerdere samples nodig zijn voor een gemiddeld resultaat. Als een van deze resultaten een grote afwijking heeft t.o.v. de andere meetresultaten, zal deze door de filter afgevlakt worden waardoor de invloed van de foutieve waarde wordt verminderd op het uiteindelijke resultaat. Afhankelijk ook van hoe groot het gewicht is van die bepaalde sample. Voor de schakeling is voor een filter gekozen met gelijke gewichten.