

Opgave reeks 1 Scilab DSP Daan Dekoning Krekels

[Oefening 1](#)

[Oefening 2](#)

[Oefening 3](#)

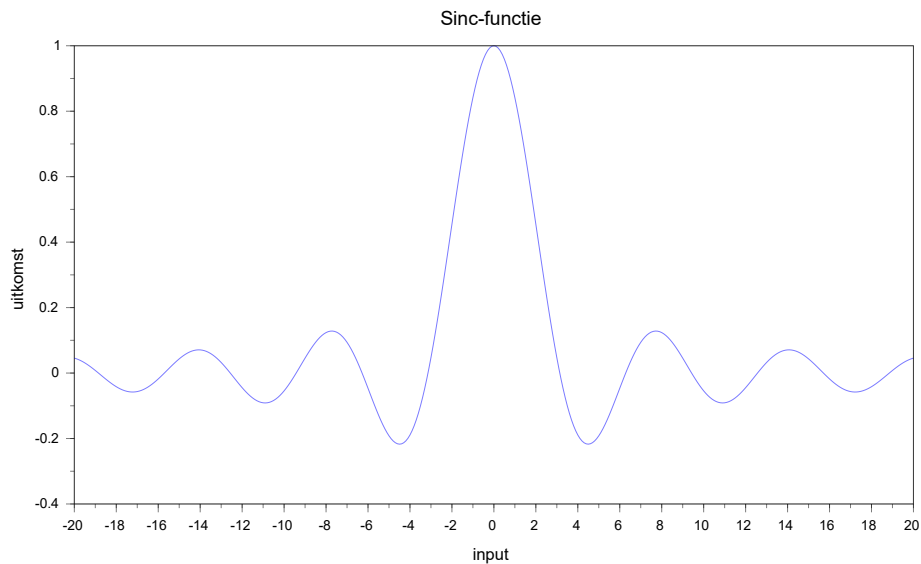
[Oefening 4](#)

Voor iedere opgave geef je de code en grafische weergave weer in een pdf-bestand. Je stuurt ook voor iedere opgave het codebestand mee. Code kan je schrijven in SciNotes en bewaren. Een scilab codebestand heeft de extentie ".sce".

Oefening 1

Teken in scilab de functie $y=\sin(\pi x)/\pi x$ in voor x van -20 tot 20. Dit is een zogenaamde sinc-functie. Varianten op deze functie vind je terug bij signaalverwerkingsmethode. Bij het maken van deze opdracht hou je best rekening met het delen van 0. Als dit voorkomt vervang je de uitkomst met de waarde 1.

```
1 // Daan Dekoning - Feb 12, 2022
2 // Labo 1 oefening 1
3 // Sinc-functie
4
5 clf;
6 clear;
7
8 x = [-20:0.1:20];
9
10 function [y]=f(x)
11     if x <> 0 then
12         y=sin(x)/ x;
13     else
14         y = 1;
15     end
16 endfunction
17
18 xtitle("Sinc-functie", "input", "uitkomst")
19 plot(x,f)
```



Oefening 2

Ontwerp een functie die een vergelijking van de tweede graad kan inlezen, de nulpunten bepalen en deze functie kan tekenen? Vang hierbij het eventueel delen door 0 op door een melding te geven “opgelet deling door 0” Zorg ervoor dat je code in alle omstandigheden werkt (tip hiervoor maak je best een analyse)

Een vergelijking van de tweede graad heeft volgende vorm: $y=ax^2+bx+c$. Met a , b en c reële getallen.

✓ Nulpunten bepalen $D = b^2 - 4 * a * c$

- $D > 0 \Rightarrow n1 = \frac{-b-\sqrt{D}}{2a}; n2 = \frac{-b+\sqrt{D}}{2a}$

```

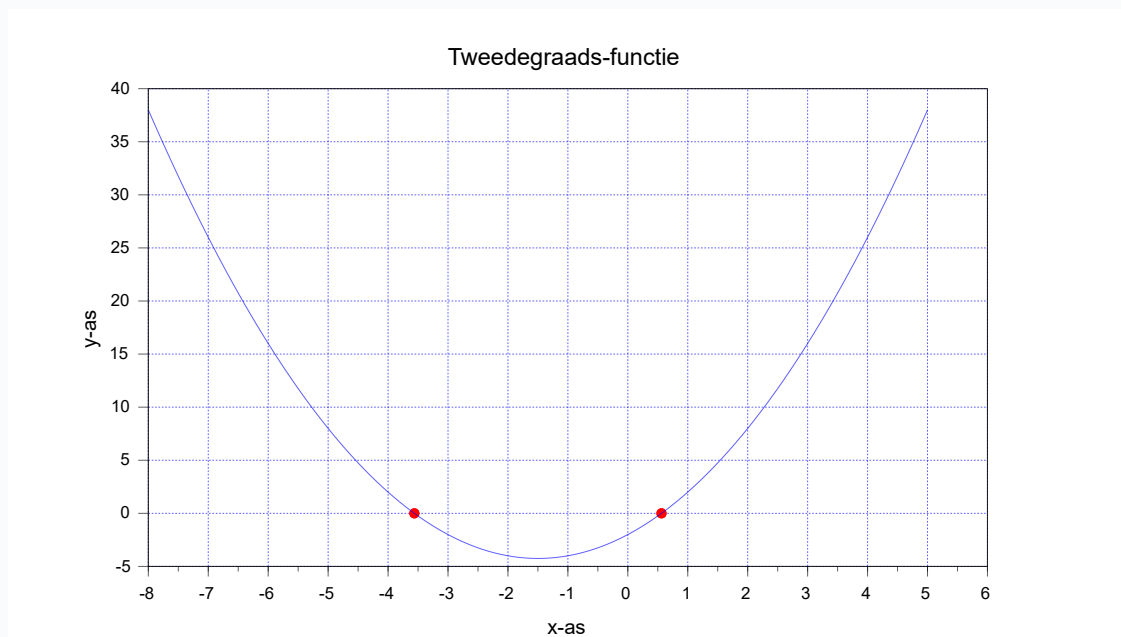
1 // Daan Dekoning - Feb 12, 2022
2 // Labo 1 Oefening 2
3 // Tweedegraadsvergelijking oplossen
4
5 // y=ax^2+bx+c
6
7 clf;
8 clear;
9
10 x = [-8:0.1:5];
11
12 function [y]=f(x)
13     a=1;
14     b=3;
15     c=-2;
16     if a == 0 & b == 0 & c == 0 & x == 0 then
17         disp("Opgelet, deling door 0")
18         y=0;
19     else
20         y= (a * x)^2 + (b * x) + c;
21     end

```

```

22     D = b^2 - 4 * a * c;
23     if D > 0 then
24         n1 = (-b-sqrt(D))/2;
25         n2 = (-b+sqrt(D))/2;
26         plot(n1, 0, '.r');
27         plot(n2, 0, '.r')
28     end
29
30 endfunction
31
32 //disp(f(1,3,7,5))
33 xtitle("Tweedegraads-functie", "x-as", "y-as")
34 plot(x,f)
35 xgrid(2)

```



Oefening 3

In een digitaal systeem wordt een sinus met amplitude 2 V ingelezen met een 10-bit ADC en een samplefrequentie van 8 kHz. De sinus is gesupponeerd op een DC-spanning van 2,5 V.

Stel dat er vier perioden van het signaal worden ingelezen.

Gevraagd:

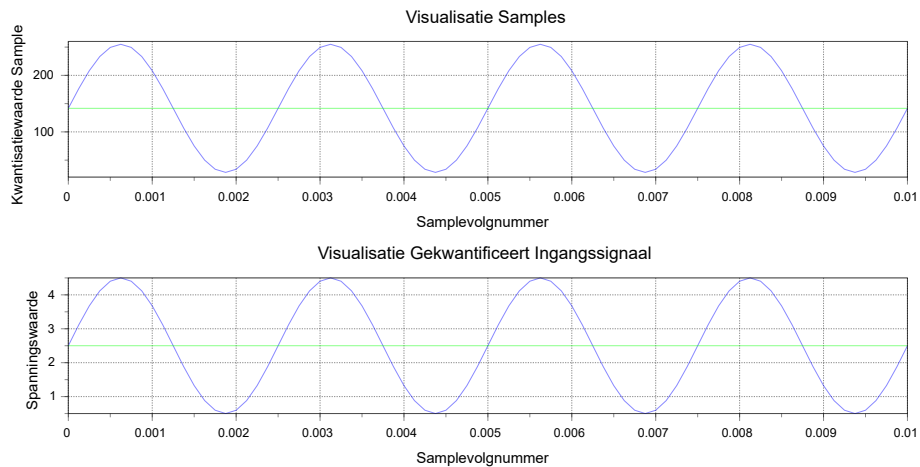
- Geef de ingevoerde digitalisatie grafisch weer in een grafiek (samplewaarde in functie van de samplenummer) (bv code 255 op moment dat vierde sample wordt ingelezen). Pas de benaming van de assen van de grafiek aan als volgt: titel grafiek : visualisatie samples; Y-as : kwantisatiewaarde sample; X-as : samplevolgnummer
- Zet de grafiek van a) om in een amplitude in functie van de tijd grafiek. Pas de benaming van de assen van de grafiek aan als volgt: titel grafiek : visualisatie gekwantificeert ingangssignaal; Y-as : spanningswaarde
- Bepaal de gemiddelde waarde van het digitaal ingelezen signaal
- Bepaal de amplitude variatie van het ingelezen signaal

1/2 pi f t

☒ Digitale weergave

- ✓ Amplitude weergave
- ✓ Gemiddelde waarde
- ✓ Amplitude variatie

```
1 // Daan Dekoning - Feb 13, 2022
2 // Labo 1 Oefening 3
3 // Analoge spanning
4
5 clf;
6 clear;
7
8
9 t = 0.0 : 0.000125 : 0.01; // 0.01 seconden signaal, fs=8kHz
10
11 sin_1Hz = 2*sin(2*pi*400*t); // 2V amplitude 400Hz signaal
12
13 // DC spanning 2.5V
14 DC = 2.5;
15
16 // Het testsignaal bestaat uit de som van de 3 signalen
17 signal = DC + sin_1Hz;
18
19 function gemiddelde(t, sig)
20     // Bepalen van gemiddelde waarde van signaal
21     gemid = mean(sig)
22     disp (gemid, "gemiddelde waarde = ")
23     // Maken van rijvector met waarde van gemid
24     gemiddelde = gemid*(ones(t));
25     // Weergeven van gemiddelde waarde als een lijn op de grafiek in het
    groen
26     plot (t, [gemiddelde], 'g.')
27 endfunction
28
29 subplot(2, 1, 1)
30 kwantis = (signal/max(signal))*255;
31 xtitle("Visualisatie Samples", "Samplevolgnummer", "Kwantisatiewaarde
    Sample")
32 xgrid(1)
33 plot(t,kwantis); // (signaal / 4.5)*255
34 gemiddelde(t, kwantis)
35
36 subplot(2, 1, 2)
37 xtitle("Visualisatie Gekwantificeert Ingangssignaal", "Samplevolgnummer",
    "Spanningswaarde")
38 xgrid(1)
39 plot(t,signal);
40 gemiddelde(t, signal)
41
42 disp("Amplitude variatie= ", stdev(signal))
```



"Amplitude variatie= " 1.4142136

Oefening 4

Maak een grafiek waarin een uniform randomsignaal (in het groen) met 200 randomwaarden gelegen tussen de waarde 0 en 4. Iedere randomwaarde stelt een bepaalde amplitude voor. Gevraagd:

- Bepaal het maximum en minimum van dit signaal en geef weer in de grafiek waar deze zich bevinden aan de hand van een asterix (*) in het rood.
- Bepaal de "DC-component" van dit signaal en maak deze zichtbaar in een grafiek via een blauwe lijn. De DC-component kan gevonden worden door het gemiddelde te bepalen van dit signaal (kijk of hiervoor geen instructie is)
- Bepaal de deviatie van het signaal tot deze "DC-component". Welk is de gemiddelde amplitudegrootte van dit signaal. Kijk of je voor deviatie geen instructie vindt. Deviatie geeft de afwijking weer van het randomgetal ten opzichte van het gemiddelde. Op deze wijze geeft de deviatie de "gemiddelde" amplitude weer van het random signaal

- ☒ Max en min
- ☒ DC-component bepalen
- ☒ Deviatie bepalen

```

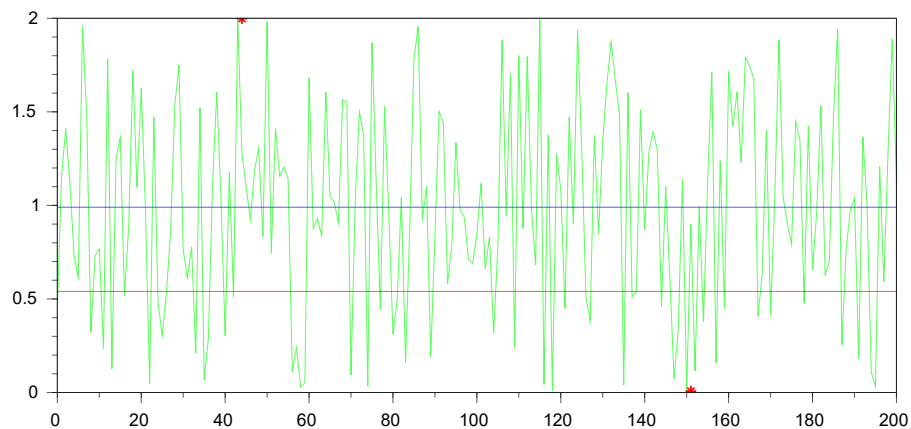
1 // Daan Dekoning - Feb 13, 2022
2 // Labo 1 Oefening 4
3 // Random Signaal
4
5 clf;
6 clear;
7
8 t = [0:1:200];
9
10 r=(rand(1,201,'uniform')/1)*2 // Random tussen 0 en 2
11
12 plot(t, r,'g')
13 [y, x] = min(r)
14 plot(x, y, '*r')
15

```

```

16 [y, x] = max(r)
17 plot(x, y, '*r')
18
19 // Bepalen van gemiddelde waarde van signaal
20 gemid = mean(r)
21 //disp (gemid, "gemiddelde waarde = ")
22 // Maken van rijvector met waarde van gemid
23 gemiddelde = gemid*(ones(t));
24 // Weergeven van gemiddelde waarde als een lijn op de grafiek in het groen
25 plot (t, [gemiddelde], 'b')
26
27 st = stdev(r) // Standaard deviatie bepalen
28 st_line = st * (ones(t)) // Lijn maken om deze stdev weer te geven
29 plot(t, st_line, 'r') // Lijn plotten
30 disp("Standaard deviatie=", st)
31

```



Gemiddelde = blauw; Standaard deviatie = rood

"Standaard deviatie=" 0.5952740