

# 50 Hz-brom verwijderen

© Rutger Teunissen

30 aug 2014

t.b.v. NLT-module **Sound Design H4**

Sound Design tekst: <http://betavak-nlt.nl/downloads/...>

Sound Design site: <http://muziekexact.nl/NLT/NLT.htm>

software versie: [WaveWizard okt 2014](#)

voorkennis: Sound Design H2 t/m H4.3.

trefwoorden: *periodiciteit, echo toevoegen.*

## 1 Inleiding

Hoe 50 Hz-brom kan klinken en hoe je die aan een geluid toevoegt, heb je in Opgaven 33 en 34 (pag 74-75) al gezien. Veel nuttiger en interessanter is het natuurlijk om te weten hoe je die brom er weer uit haalt!

Het eerste wat misschien bij je opkomt is om de bewerking simpelweg om te keren: in Opgave 33 maakte je brom door een blokgolf met een frequentie van 50 Hz bij het raaf-geluid *op te tellen*. Dus waarom niet gewoon diezelfde blokgolf weer van het resultaat *afrekken*? Inderdaad, dan hou je het oorspronkelijke, onvervormde raaf-geluid weer over. Maar dat heeft natuurlijk alleen zin als je een blokgolf aftrekt die *identiek* is met de blokgolf die je er eerst bij optelde; dus niet alleen met dezelfde frequentie, maar ook hetzelfde startpunt (fase) en dezelfde amplitude. En als je een of ander onbekend geluidsbestand opent (bijv. een video lecture van YouTube) en er blijkt een brom in te zitten, dan ken je noch de fase noch de amplitude...

Dat is helaas niet het enige probleem, want de brom is niet perse altijd een blokgolf. Bijvoorbeeld, als er zich tijdens de opname een lichtdimmer in de buurt van de audio-apparatuur bevond, dan is de brom een *pulsgolf* (pag 92). In weer andere gevallen is het een sinus- of een driehoekgolf. Ook komt het voor dat de brom afwisselend harder en weer zachter wordt, bijvoorbeeld als een spreker met een defecte das-speld-microfoon heen en weer loopt.

Als je luistert naar een geluidssignaal dat met een brom is verstoord, dan beseft je meestal onmiddellijk dat het gaat om twee *verschillende, onafhankelijke* geluidsbronnen. Want we kunnen ons gemakkelijk indrukken als gescheiden *voorstellingen* die onze zintuigen bereiken als één geheel, als we die indrukken eerder al eens *afzonderlijk* hebben waargenomen. Maar als we een wiskundige bewerking willen uitvoeren die tot doel heeft om de brom en het ongestoorde geluid *fysiek* van elkaar te scheiden (zodat we ze bijvoorbeeld op twee verschillende geluidssporen kunnen zetten), dan staan we voor een uitermate lastige, zo niet onmogelijke klus. De meer geavanceerde, statistische signaalverwerking biedt methodes waarmee verschillende geluiden in een mix, tot op zekere hoogte en onder allerlei voorwaarden, kunnen worden geïdentificeerd en van elkaar gescheiden. Maar die methodes liggen ver buiten ons bereik. Wat we wél kunnen doen is een wat eenvoudiger vraag beantwoorden:

*Is het mogelijk om van een signaal alle tonen met frequentie  $f$  onhoorbaar te maken?*

Als dat mogelijk is, dan kunnen we elke storende brom- of pieptoon buiten gevecht stellen, ongeacht de golfvorm of schommelingen in het volume. Maar dan moeten we wel voor lief nemen dat we misschien ook iets weghalen van het oorspronkelijke, onverstoord signaal, nl. elke toon van dezelfde frequentie  $f$ .

## 2 echotijd: periode $N$

We staan dus voor de vraag om uit een gegeven geluidsopname  $G[n]$  elke toon waarvan de grondtoonfrequentie gelijk is aan  $f$  te verwijderen. Daarbij is het handig als we ons  $G[n]$  voorstellen als een mix van twee signalen:  $P_N[n]$  en  $Q[n]$ . Die mix kunnen we noteren als:

$$G[n] = P_N[n] + Q[n] \quad (1)$$

We definiëren  $P_N[n]$  en  $Q[n]$  als volgt:

- $P_N[n]$  is de verzameling van alle *periodieke* signalen met een periode van  $N$  samples (vandaar dat subscript  $N$ ). Als de frequentie  $f$  is gegeven, dan is het aantal samples  $N = F_s/f$  (ga dat na! zie pag 54).  
 $P_N[n]$  kan van alles zijn: een blokgolf, een sinusgolf, een pulsgolf, een zangtoon, een bromtoon, maar wel altijd een *toon* en altijd met een periode van  $N$  samples.
- $Q[n]$  noemen we de verzameling van alle *andere* signaalcomponenten van  $G[n]$ ; componenten die dus *niet* tot  $P_N[n]$  behoren en waarvan de periode *ongelijk* is aan  $N$ .

Gegeven is dus dat signaal  $P_N[n]$  periodiek is met periode  $N$  en we zijn op zoek naar een manier om  $P_N[n]$  te verwijderen uit  $G[n]$ .

Herinner je (pag. 35) dat volgens de definitie van een periodieke functie geldt dat:

$$P_N[n] = P_N[n \pm N] = P_N[n \pm 2N] = P_N[n \pm 3N] = \dots \quad (2)$$

Dit brengt ons op de gedachte om op het geluid van  $G[n]$  een echo te laten volgen die precies  $N$  samples later komt dan  $G[n]$ . Deze echo kunnen we in formulevorm uitdrukken als  $G[n - N]$ . Ook die echo is natuurlijk een mix van  $P_N$  en  $Q$  en we vinden volgens (1):

$$G[n - N] = P_N[n - N] + Q[n - N]$$

Maar volgens (2) mogen we ook noteren:

$$G[n - N] = P_N[n] + Q[n - N]$$

Als we nu deze echo  $G[n - N]$  *afrekken* van  $G[n]$ , en het verschilsignaal  $V[n]$  noemen, krijgen we:

$$\begin{aligned} V[n] &= G[n] - G[n - N] \\ &= (P_N[n] + Q[n]) - (P_N[n] + Q[n - N]) \\ &= Q[n] - Q[n - N] \end{aligned} \quad (3)$$

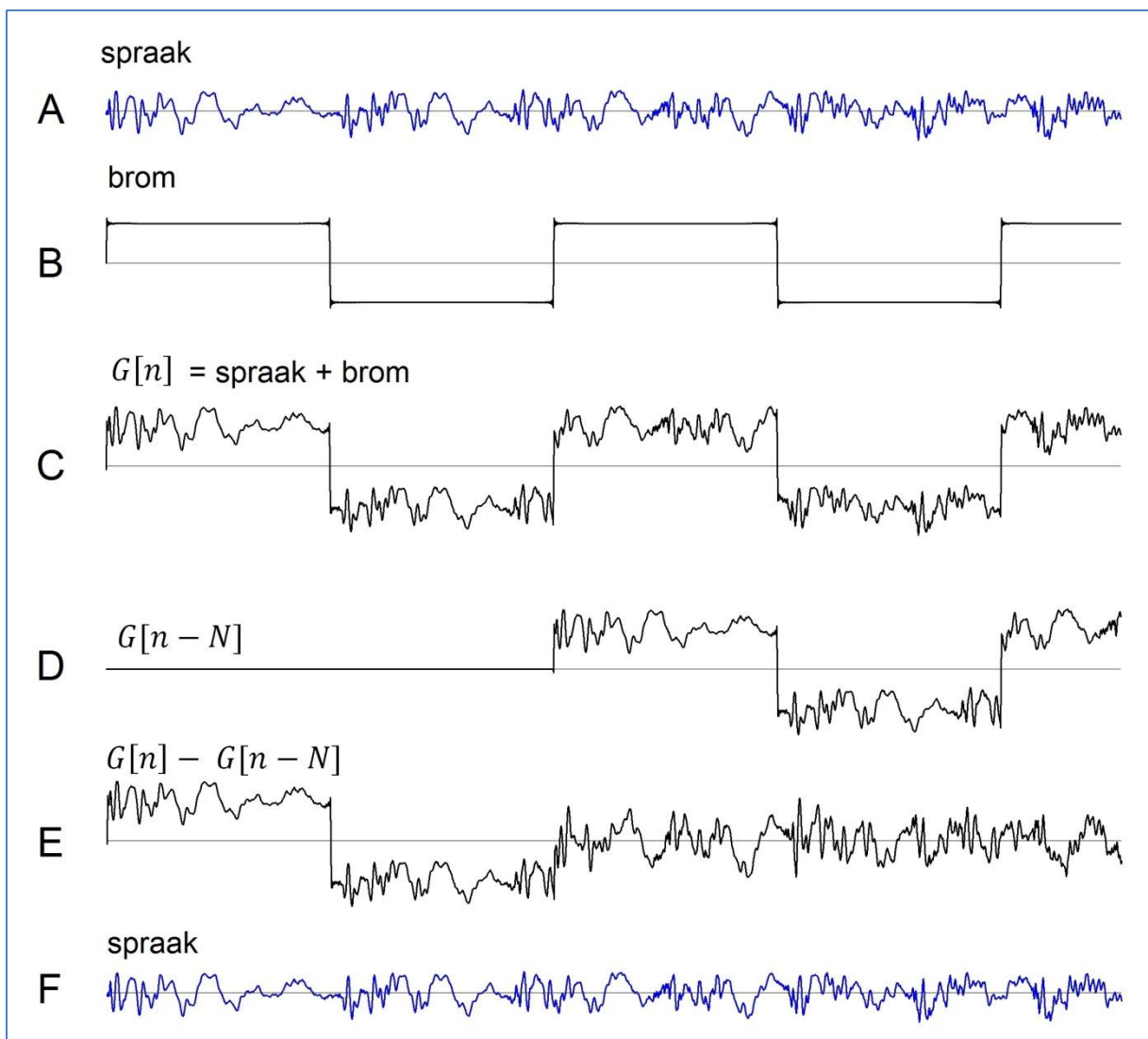
In die laatste uitdrukking komt  $P_N[n]$  niet meer voor, dus  $V[n]$  bevat geen tonen meer met periode  $N$ . Dus als  $G[n]$  een brom bevat met periode  $N$ , dan zal die in  $V[n]$  ontbreken!

### Conclusie

We kunnen tonen waarvan de periode  $N$  samples duurt verwijderen uit een signaal  $G[n]$ , *zonder* dat we iets weten over bijv. de golfvorm of het volume-verloop van die tonen! Aan formule (3) kun je dat ook goed zien: daarin komt alleen  $Q$  voor, d.w.z. een signaal *zonder* tonen met een periode van  $N$  samples.

In de figuur hieronder een voorbeeld.

- A. Spraaksignaal (blauw).
- B. Bromtoon (blokgolf) waarvan de periode uit  $N$  samples bestaat. (Let wel: ook het spraaksignaal kan zelf natuurlijk tonen bevatten waarvan de periode  $N$  samples is.)
- C. De mix van spraak + brom noemen we  $G[n]$ .
- D. Echo  $G[n - N]$ ; dit is een kopie van het directe geluid  $G[n]$ . De echo komt *later* dan het directe geluid en daarom is de kopie *naar rechts verschoven* over precies één periode van  $N$  samples.
- E. Trekken we deze echo af van het directe geluid  $G[n]$ , dan krijgen we  $V[n] = G[n] - G[n - N]$ , waarin geen tonen met periode  $N$  meer voorkomen.
- F. Nogmaals het oorspronkelijke spraaksignaal. Let op de verschillen met  $G[n] - G[n - N]$ !



### Opdracht Brom 1 $P_N[n]$ en $Q[n]$

Noteer welke van onderstaande signalen  $S[n]$  behoren tot  $P_N[n]$ .

- (a)  $S[n]$  is het geluid van ruisende bomen.
- (b)  $S[n]$  is een toon met een constante frequentie en een volume dat veranderlijk is, bijv. een stemvorktoon.
- (c)  $S[n]$  is een sinustoon met constant volume en een constante periode  $N$ . Maar de toon *begint* op tijdstip 0. (dus vóór die tijd is het stil, d.w.z.  $S[n] = 0$  voor elke  $n < 0$ ).
- (d)  $S[n]$  is een periodieke toon met frequentie  $F_s/(2N)$  Hz
- (e)  $S[n]$  is een periodieke toon met frequentie  $2F_s/N$  Hz
- (f)  $S[n]$  is een periodieke toon met frequentie  $kF_s/N$  Hz, voor  $k = 1, 2, 3, \dots$
- (g)  $S[n] = \sin \pi n/N$
- (h)  $S[n] = \sin 2\pi n/N$
- (i)  $S[n] = \cos 2\pi kn/N$ , voor  $k = 1, 2, 3, \dots$

### Opdracht Brom 2 *echo* $G[n - N]$

Open [Preset Brom Opdracht 2 echo  \$G\[n-N\]\$](#) .

Er wordt een zaagtandgolf aangemaakt op spoor S1, waarvan de frequentie **bromfreq** heet. Bovenaan de code wordt **bromfreq** ingesteld op 50 Hz.

- (a) Wat is de periode van de zaagtandgolf, uitgedrukt in seconden?
- (b) Wat is de periode van de zaagtandgolf, uitgedrukt in aantal samples? Noem dat aantal  $N$ .
- (c) Vervang de vraagtekens achter de code  $\mathbf{N} = ???$  door een formule waarin **bromfreq** voorkomt.
- (d) Waarom is dat handig?
- (e) Wat moet je invullen bij codeblok 2e om op spoor S2 de zaagtandgolf van spoor S1 te krijgen met een tijdverschuiving van  $N$  samples (een echo dus)?
- (f) Vul codeblok 2f in dat ervoor moet zorgen dat op spoor S3 het verschilsignaal  $V[n] = G[n] - G[n - N]$  komt te staan. Geef twee oplossingen.

### Opdracht Brom 3 *Sinus met random frequentie verwijderen uit spraak*

- (a) Open [Preset Brom Opdracht 3 Deel 1](#). Zorg ervoor dat weerbericht.wav in dezelfde map staat als WaveWizard, vóórdát je op Start klikt. Dit wav- bestand wordt dan door een code-instructie automatisch geopend (zie kader *de instructie Open WAV*). Er wordt een sinustoon bij het weerbericht gemixt, maar de frequentie daarvan is een random getal tussen 50 en 1000 en het kan dus zowel een brom- als een pieptoon zijn. Meet de frequentie op met de techniek uit H2.6 Opdracht 20.
- (b) Als  $f$  de frequentie is die je bij (a) vond, dan is  $T = \dots$  sec. En dan is de periode uitgedrukt in aantal samples  $N$  gelijk aan ... Open [Preset Brom Opdracht 3 Deel 2](#) en vul je antwoord in bij de code.
- (c) Vul bij codeblok 3c een formule in waarmee je op spoor S2 de brom- of piepvijve versie van spoor S1 krijgt.

### Opdracht Brom 4 *twee bijzondere effecten*

Je maakt een zaagtandgolf  $Z[n]$  met een frequentie  $f = 100$  Hz.

- (a) Hoe groot is de periode, uitgedrukt in aantal samples  $N$ ? Schrijf zelf code om de antwoorden op vragen (b) en (c) hieronder te vinden met behulp van WaveWizard. (Hint: kopieer codeblokken, bijv. uit vorige opdracht, en breng daar veranderingen in aan.)
- (b) Welke bekende golfvorm krijg je als je van  $Z[n]$  een echo  $Z[n - N/2]$  **afrekt**?
- (c) Wat krijg je als je bij  $Z[n]$  een echo  $Z[n - N/2]$  **optelt**?

## Samenvatting en vooruitblik

Een bromtoon met een constante frequentie  $f$  kun je uit een geluidsoptname verwijderen door van dat geluid een echo af te trekken, waarvan de echotijd gelijk is aan  $T = 1/f$ , dus een tijd die precies overeenkomt met 1 periode van de bromtoon.

De echotijd  $T$  kun je uitdrukken in aantal samples  $N = T \cdot F_s$ .

Dit echo-effect verwijdert niet alleen de frequentie  $f$ , maar ook  $2f, 3f, 4f, \dots < 0,5F_s$ , want al die tonen zijn óók periodiek op  $T$ .

Het echo-effect heeft als "bijwerking" dat ook uit het *oorspronkelijke* geluid al de tonen die periodiek zijn op  $T$  worden weggehaald. Maar dat laatste is meestal veel minder storend dan de brom.

Door een echo van  $N/2$  samples *af te trekken* van een zaagtandtoon met periode  $N$  kun je een toon een octaaf hoger maken; door een echo van  $N/2$  samples *op te tellen* wordt de klank "neuziger" (blokgolf-achtiger). Geldt dat ook voor andere golfvormen dan de zaagtand? Misschien heb je die vraag zelf al beantwoord door te experimenteren met WaveWizard! Maar welk "systeem" zit daar achter?

Verder hebben we ook geen antwoord gegeven op de belangrijke vraag wat er nu precies is gebeurd met het bewerkte spraaksignaal. Dat het resultaat  $V[n]$  *anders* klinkt dan het onbewerkte en ongestoorde signaal is wel duidelijk, maar de vraag is of *alleen maar* tonen met periode  $N$  worden weggehaald en of daarnaast niet ook *alle andere* frequenties, die we hierboven  $Q$  noemden, enigszins worden "aangetast"? Als je vermoedt dat dat laatste het geval is, zit je op het goede spoor! Maar wat is de verklaring daarvoor? En hoe kunnen we die wiskundig formuleren?

Zo gaat het meestal in de wetenschap! We hebben een prima oplossing gevonden voor het "brom-"probleem, maar zijn daarmee tevens op een verschillende nieuwe, diepere vragen gestuit!

Een antwoord daarop vind je in *Een echo-effect is een kamfilter*, dat aansluit bij H5.4.

## Appendix

Hieronder iets over de nieuwe instructie **Open WAV** die in de opdrachten hierboven wordt gebruikt en die je aantreft in WaveWizard vanaf versie oktober 2014.

### de instructie **Open WAV** de variabele **WAV\_Lengte**

Als je een wav-bestand in dezelfde map plaatst als WaveWizard, dan hoef je het bestand niet handmatig te openen via de muis en het bekende Windows-selectieformulier, maar kun je het openen overlaten aan WaveWizard. Dat heeft een aantal grote voordelen. De instructie luidt:

```
Open WAV
WAV naam   Bach_vioolconcert.wav
vanaf      240*Fs
duur       ?
buffer     S1[20*Fs]
```

Je kunt hiermee het bestand inladen vanaf en plaatsen op elk gewenst tijdstip. Je kunt ook slechts een fragment inladen.

**WAV naam** de naam van het WAV-bestand dat je wilt openen.  
**vanaf** zeker voor lange geluidsbestanden is het handig als je het kunt inladen vanaf een zeker tijdstip. Druk dat tijdstip uit in aantal samples. Dat kan ook in formule-vorm, bijv.  $240*Fs$  : inladen vanaf tijdstip 4 minuten ( $4 \times 60 = 240$  sec; 1 sec =  $Fs$  samples).  
**duur** uitgedrukt in aantal samples, zoals **vanaf**. Je kunt dus ook slechts een deel van een bestand laden. Als je het *hele* bestand wilt inladen, of als je niet weet hoe lang het duurt, kun je invullen: '?'  
**buffer** de buffer en de index op van de eerste sample van het WAV-bestand. Bijv.  $S1[20*Fs]$ .  
Als het WAV-bestand stereo is, dan wordt het tweede spoor geplaatst in de eerstvolgende buffer; bijv. als je opgeeft  $S1[0]$ , dan wordt het tweede spoor op  $S2[0]$  gezet.

#### de variabele **WAV\_lengte**

Het aantal samples van het geopende bestand slaat WaveWizard op in de variabele **WAV\_lengte**. Gebruik **WAV\_lengte** om bijvoorbeeld het aantal bewerkingen op te geven dat je wilt uitvoeren op het geopende WAV-bestand.

Als je bij **duur** geen vraagteken hebt gezet, zal **WAV\_lengte** gelijk zijn aan wat je bij **duur** hebt ingevuld.