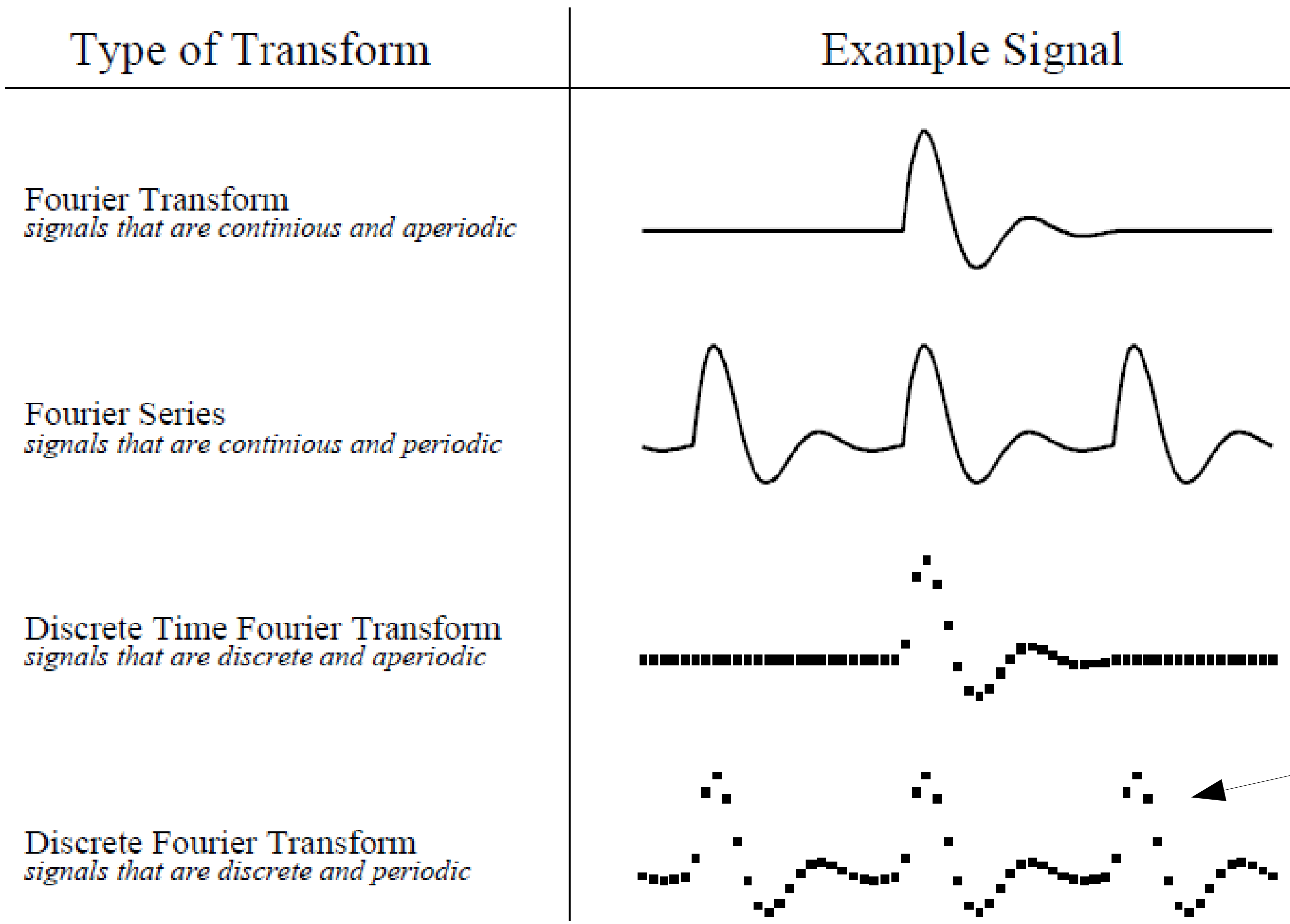
DEC – DSP – SDR 5
Dicrete Fourier Transform

- Familie van Fourier transformaties
  - Fourier Transform
  - Fourier Series
  - Discrete Time Fourier Transform
  - Discrete Fourier Transform

- Berekening van een frequentie spectrum uit een discreet (samples) en periodiek signaal d.m.v. de Discrete Fourier Transform (DFT)

Familie van Fourier transformaties



De Fourier Transformatie kan worden gebruikt om een signaal in het tijdstomein om te zetten naar het frequentie domein.

M.a.w. Een scope-plaatje (signaal zichtbaar in de tijd) wordt opgezet naar een frequentie spectrum (zoals je zou zien op een sprectrum analyser).

Hiernaast zie je verschillende scope plaatjes, met de verschillende transformatie-methoden om ze om te rekenen naar een frequentie spectrum.

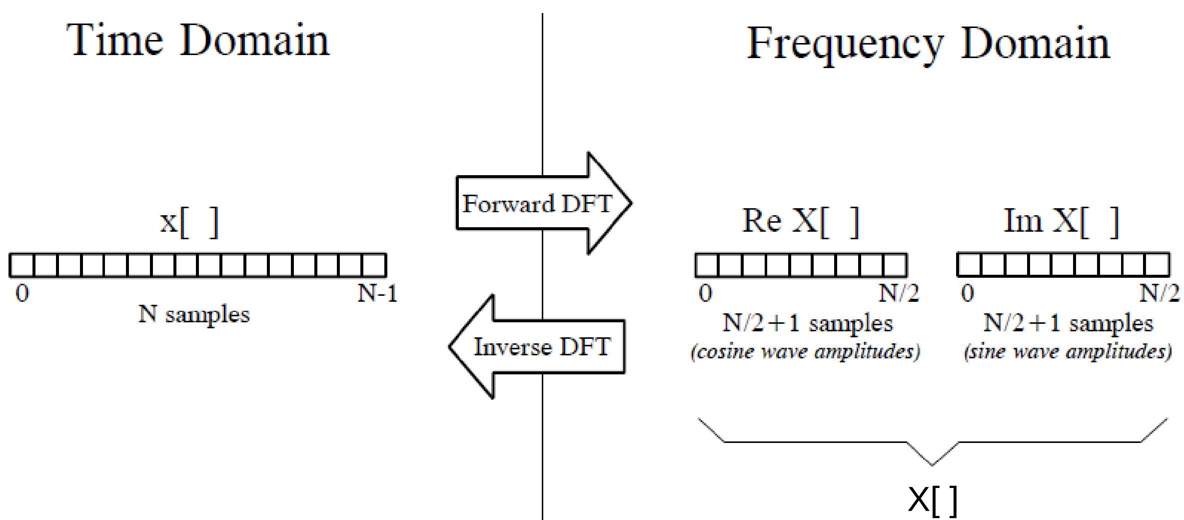
We richten ons op het onderste plaatje, een gesampled periodiek signaal. Hier wordt de DFT toegepast.

Er kan gewerkt worden met complexe getallen (a + jb) maar we werken hier nu alleen met reële getallen. Daarom heet het de ‘Real DFT’.

Wat doet de Real DFT

- Wat je eigenlijk wilt weten is welke frequenties er in een signaal voorkomen en met welke amplitude, die je vervolgens in een frequentie-spectrum kunt weergeven.
- In de berekening komen de volgende zaken naar voren:
  Frequenties, Amplitudes en Faseverschuiving van iedere sinus (dat laatste vinden we niet altijd interessant maar hebben we wel mee te maken)

  De fase verschuiving wordt uitgedrukt in een cosinus en een sinus van de zelfde frequentie. (een I/Q signaal, zie eerdere sheets, ook wel een Reëel (Re) en Imaginair (Im) signaal genoemd)



PI4DEC

Tijdsdomein signaal x[ ] (kleine letter) wordt omgerekend naar frequentiedomein signaal X[ ] (hoofdletter).

Dat laatste valt uiteen in het cosinus-deel Re X[ ] en het sinus-deel Im X[ ]

Als x[ ] N samples heeft, dan kun je N/2 + 1 cosinus- en N/2 + 1 sinus-amplitudes berekenen.

DFT berekenen m.b.v. correlatie

Wat staat hier:

- Er worden N/2 frequentiecomponenten voor sinus (= ImX[ ]) en cosinus (= ReX[ ]) uitgerekend uit N samples x[ ]
- Bijvoorbeeld: de tweede frequentie component ReX[1] (dus k=1) volgt uit :

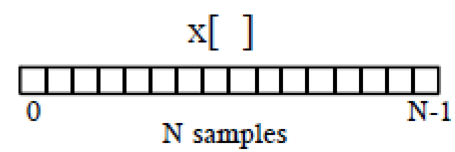
  ReX[1] =
  x[0] * cos(2pi*1* 0/N) +
  x[1] * cos(2pi*1* 1/N) +
  x[2] * cos(2pi*1* 2/N) +
  … +
  x[N-1] * cos(2pi*1* N-1/N)

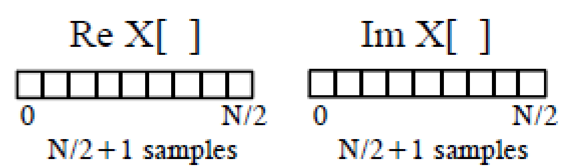
  Als je goed kijkt zie je dat hier één complete enkele cosinus wordt doorlopen: 2pi * 0 t/m 2pi * N-1/N
  Bij berekenen van ReX[2] worden 2 complete cosinussen doorlopen. Bij ReX[3] dus 3 cosinussen.

- En zo dus voor k = 0 t/m k = N/2
  Iedere keer worden er N vermenigvuldigingen gedaan waarvan de uitkomsten worden opgeteld. Dit is vergelijkbaar met de manier van filteren d.m.v. correlatie, zie eerdere sheets.
  Zo worden er bij 16 samples dus 9 frequentievakjes (bins) gevuld waarbij de eerste bin de DC component is
- Iets vergelijkbaars doen we bij de sinus componenten ImX[ ]

$$ReX[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi k i / N)$$

$$ImX[k] = -\sum_{i=0}^{N-1} x[i] \sin(2\pi k i / N)$$

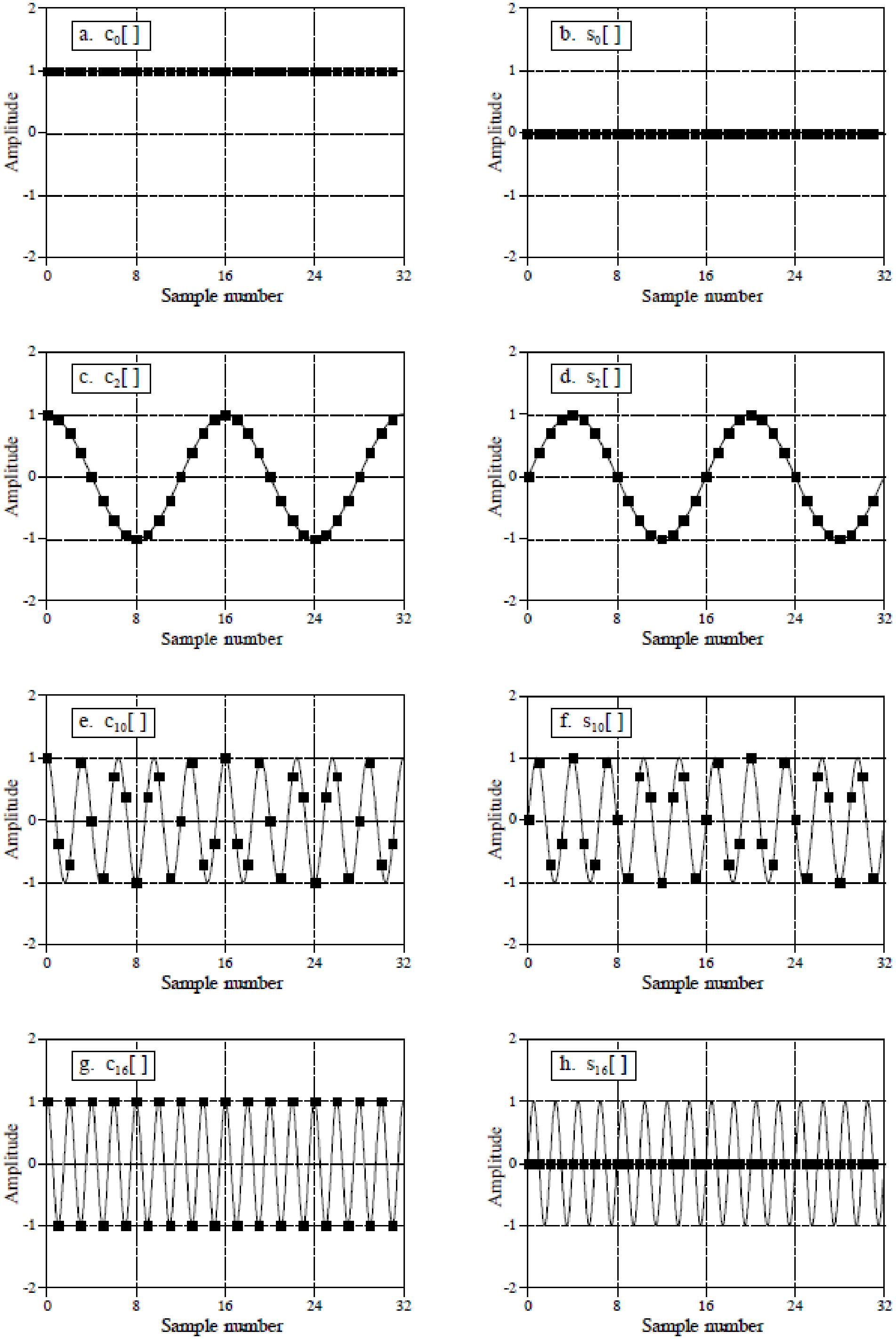




DFT berekening

- Hiernaast zie je een serie sinussen en cosinussen.
- Behalve de DC component (co en so) zie je o.a. c2 en s2, d.w.z. dat er 2 cosinussen of cosinussen in de 32 samples passen en b.v. c16 waarbij 16 cosinussen in de 32 samples passen. (dit is dus het maximum aantal)

- De vorige sheet beschreef de vermenigvuldiging van het oorspronkelijke signaal x[ ] met één enkele sinus die past binnen de samples.
- In totaal berekenen we van alle mogelijke sinussen en cosinussen die passen binnen de 32 samples (0 Hz tot ½ x de samplefrequentie) de correlatie met het gegeven gesampelde signaal
- In nevenstaand voorbeeld van 32 samples moeten we dus 2 x N² vermenigvuldigingen doen. (we nemen s0 en s16 niet mee (dit zijn allemaal nullen)



DFT berekening

- Nu we de sinus- en cosinusbins hebben gevuld kunnen we het vermogen in het spectrum berekenen door de verschillende bins bij elkaar ‘op te tellen’ met het Pythagoras principe:

  Totaal in bin n = Wortel ($ReX[n]^2 + ImX[n]^2$)

- We hebben nu een serie samples omgerekend in een frequentie spectrum d.m.v. de Real DFT. De frequentie van de hoogste bin ligt op de helft van de samplefrequentie (en dat is wel logisch)

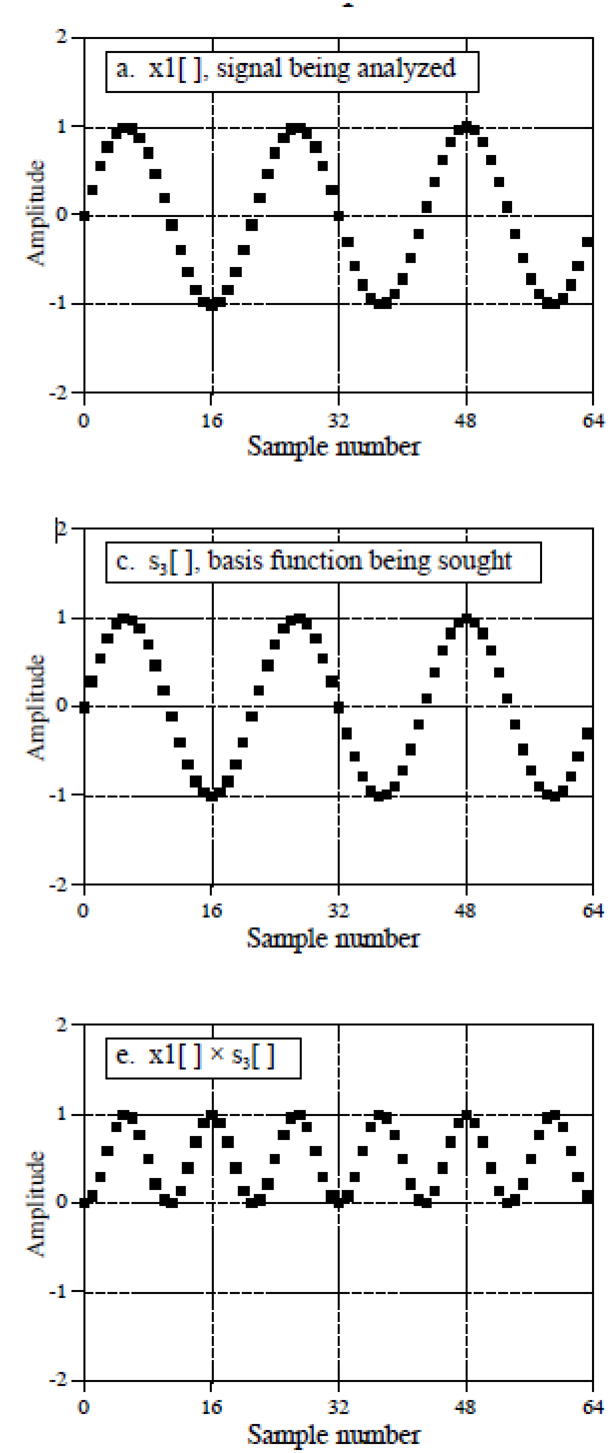
- De illustere FFT (Fast Fourier Transform) werkt met dezelfde uitgangspunten. Het heeft echter een sneller algoritme dan de zojuist uitgelegde DFT, waarbij *het <u>aantal vermenigvuldigingen</u> drastisch afneemt* zodat het gewenste frequentiespectrum snel kan worden berekend.

  B.v.:
  De DFT met 1024 samples vraagt om een orde grootte van 1.000.000 vermenigvuldigingen
  Een FFT met het zelfde aantal samples een ordegrootte van 10.000.
  Winst dus een factor 100!

Voorbeeld van een correlatie (in dit geval met één sinus)

- Een eenvoudig voorbeeld:
  - Het gesampelde signaal past 3 x in de 64 samples (heel toevallig).
  - We testen nu de correlatie met een sinus genaamd S3 die dus ook 3x past (die dus een frequentie heeft van 3/64 x de sample frequentie)
- De vermenigvuldigingen uit de DFT zie je onderaan. Ze komen voort uit:
  x[0] * s3[0], x[1] * s3[1], enz. Dit is een punt voor punt vermenigvuldiging tot aan sample 63.

  Uitkomsten zijn allemaal positieve getallen. Tel je die allemaal bij elkaar op, dan heb je het relatieve vermogen aan sinus in bin 3.



Voorbeeld van een correlatie (geval 2)

- Tweede voorbeeld:
  - Het gesampelde signaal x2[ ] heeft meerdere frequentiecomponenten.
  - We testen nu opnieuw de correlatie met de sinus genaamd S3 die dus ook 3x past in de 64 samples (en die dus een frequentie heeft van 3/64 x de sample frequentie)
- De vermenigvuldigingen uit de DFT zie je onderaan. Deze hebben verschillende waarden, zowel positief als negatief. Tel je die allemaal bij elkaar op, dan zou er in dit geval ongeveer nul uitkomen.

  Er zit nu nauwelijks vermogen in bin 3 voor wat betreft sinus s3[ ]. Cosinus hebben we natuurlijk nog niet getest hier.