Exercício 4

Athos Damiani (NUSP 6796736)

05-03-2021

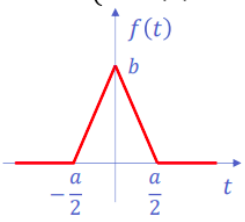
EXERCÍCIO 4

Utilizando a função fft do Matlab determine a transformada de Fourier, na forma polar, dos seguintes sinais:

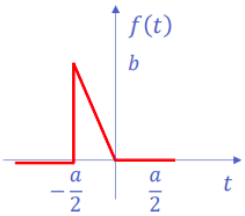
(a) $f(t) = \sin(\omega t)$

(b) $f(t) = \begin{cases} 0 & \text{se } |t| \leq 1 \\ 1 & \text{se } |t| > 1 \end{cases}$

(c)



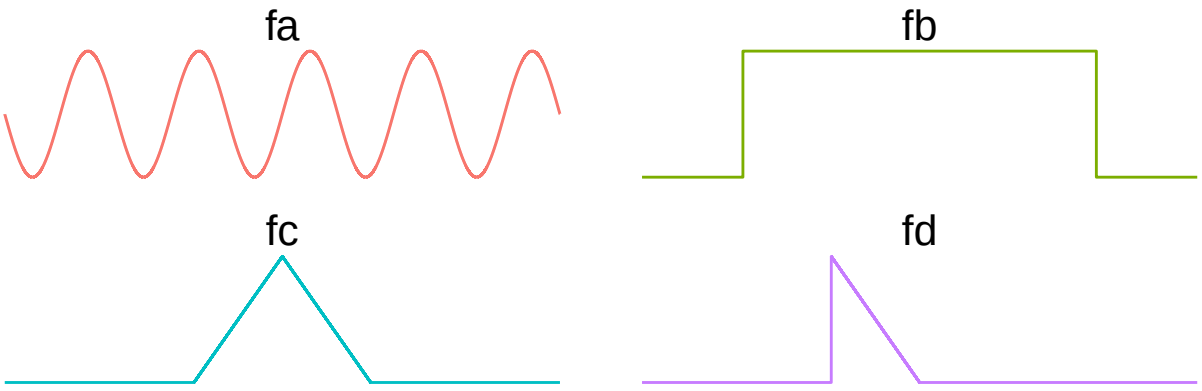
(d)



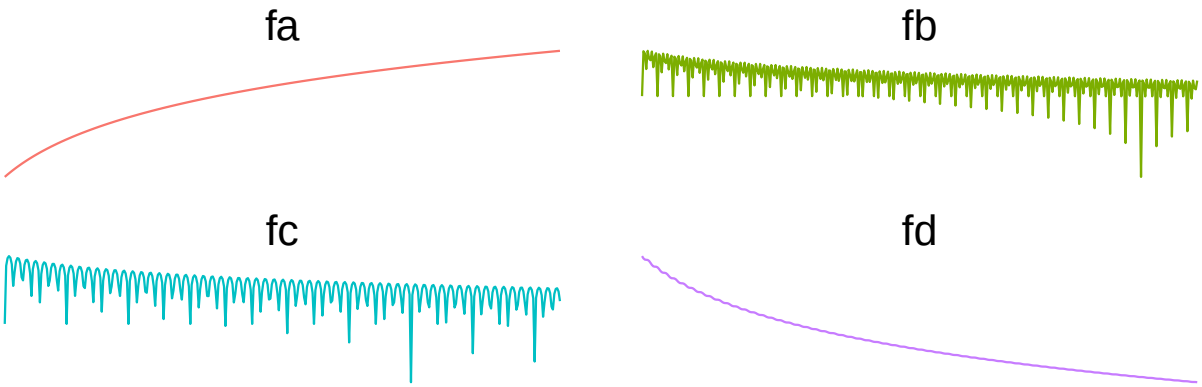
(e) Aplicar aos sinais dos itens b-d os seguintes filtros: 1) Passa-baixa ideal; 2) Butterworth de 4ª ordem;3) Chebyshev tipo 1; 4) Chebyshev tipo 2. Em seguida, reconstruir os sinais filtrados.

```
library(tidyverse)
fa <- function(t, w = 10) sin(t*w)
fb <- function(t) as.numeric(abs(t) < 1)
fc <- function(t, a=1, b=1) (abs(t) < a/2)*ifelse(t < 0, b + t*(2*b/a),  b + t*(-2*b/a))
fd <- function(t, a=1, b = 1) (t < 0)*(t > -a/2)*(-2*b/a)*t
t <- seq(-pi/2, pi/2, length = 44000)
```

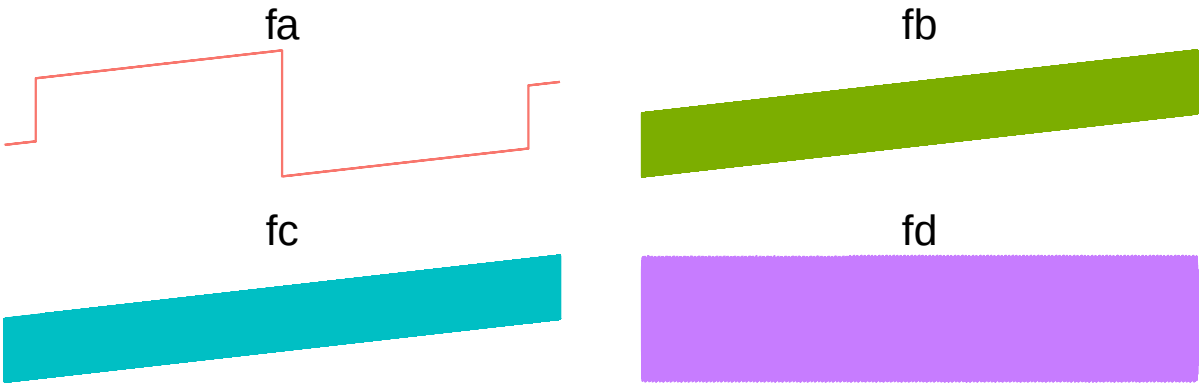
Sinais



Espectros



Fases



Filtros

```
# Passa Baixa Ideal
h1 <- function(n = 44000, wc = 500)
  ifelse(abs(seq_len(n) - n/2) < n/2 - wc, 0, 1)
```

```r
# Butterworth
h2 <- function(n = 44000, wc = 500, nn = 4) {
  w <- abs(seq_len(n) - n/2)
  1/(1 + (w/wc)^nn)
}

# Auxiliar para Chebychev
Tn <- function(w, wc = 500, nn = 1) {
  wwc <- w/wc
  ifelse(w < wc,
         cos(nn*acos(wwc)),
         ((wwc*sqrt(wwc^2 - 1))^nn + (wwc*sqrt(wwc^2 - 1))^(-nn))/2)
}

# Chebychev Tipo 1
h3 <- function(n = 44000, wc = 500, nn = 4, e = 1) {
  w <- seq_len(n/2)
  w <- c(w, w[length(w):1])
  f <- 1/sqrt(1 + e^2 * Tn(w, wc, nn)^2)

  ifelse(is.nan(f), 0, f)
}

# Chebychev Tipo 2
h4 <- function(n = 44000, wc = 500, nn = 4, e = 1) {
  w <- seq_len(n/2)
  w <- c(w, w[length(w):1])
  f <- 1/sqrt(1 + 1/(e^2 * Tn(w, wc, nn)^2))

  ifelse(is.nan(f), 0, f)
}
```

```r
n <- length(t)
fs_nested <- list(
  fa = fa(t)/2 + 0.5,
  fb = fb(t),
  fc = fc(t, a = 1),
  fd = fd(t, a = 1)
) %>%
  enframe(name = "f", value = "f_t") %>%
  mutate(
    fft_w = map(f_t, fft),
    passa_baixa = map(fft_w, ~.x*h1(n)),
    butterworth_4a_ordem = map(fft_w, ~.x*h2(n)),
    chebchev_tipo_1 = map(fft_w, ~.x*h3(n)),
    chebchev_tipo_2 = map(fft_w, ~.x*h4(n))
  )

fs <- fs_nested %>%
  select(-fft_w, -f_t) %>%
  pivot_longer(-f, names_to = "filtro", values_to = "fft_w")
```

Sinais reconstruídos