

# Práctica: Filtros digitales

Nerea Galera Navarro

Diciembre 2025

## Índice

<b>1. Prelaboratorio</b>	<b>2</b>
<b>2. Laboratorio</b>	<b>3</b>
2.1. Diseño de filtros digitales con ceros y polos de transformada Z . . . . .	3
2.1.1. Análisis del sistema $H_2(z)$ . . . . .	25
2.1.2. Comparación entre $H_1(z)$ y $H_2(z)$ . . . . .	44
2.1.3. Filtrado de una señal de prueba . . . . .	44
2.2. Filtro peine . . . . .	46
2.3. Eliminación del ruido de red . . . . .	57

En esta práctica estudiaremos la respuesta en frecuencia de los sistemas LTI discretos. Veremos cómo podemos utilizar la transformada Z para determinar la salida en el estado estacionario de un sistema dada una entrada. Se analizarán sistemas con una determinada función de transferencia y, por otra parte, se diseñarán sistemas para obtener la respuesta en frecuencia deseada.

## 1. Prelaboratorio

### Relación entre la transformada de Fourier y la transformada Z

La transformada de Fourier es la transformada Z evaluada sobre la circunferencia unidad y permite obtener directamente la respuesta en frecuencia del sistema.

### Funciones que utilizaremos a lo largo de esta práctica

- `zplane`: muestra gráficamente los polos y ceros de un sistema en el plano Z. Es útil para estudiar la estabilidad y el comportamiento del filtro o sistema discreto.
- `impz`: calcula y representa la respuesta al impulso de un sistema definido por sus coeficientes. Sirve para ver cómo responde el sistema a un impulso.
- `freqz`: obtiene y grafica la respuesta en frecuencia del sistema. Nos permite ver cómo se comporta el sistema en distintas frecuencias.

```
suppressWarnings({
  library(signal)
  library(pracma)
  library(R.matlab)
})

##
## Adjuntando el paquete: 'signal'

## The following objects are masked from 'package:stats':
##
##   filter, poly

##
## Adjuntando el paquete: 'pracma'

## The following objects are masked from 'package:signal':
##
##   conv, ifft, interp1, pchip, polyval, roots

## R.matlab v3.7.0 (2022-08-25 21:52:34 UTC) successfully loaded. See ?R.matlab for help.

##
## Adjuntando el paquete: 'R.matlab'

## The following objects are masked from 'package:base':
##
##   getOption, isOpen
```

## 2. Laboratorio

### 2.1. Diseño de filtros digitales con ceros y polos de transformada Z

En este ejercicio vamos a estudiar cómo la posición de los polos y ceros en el plano Z influye en la respuesta en frecuencia de un sistema discreto. Para ello trabajaremos con dos funciones de transferencia de la forma:

$$H_1(z) = 1 - 2r \cos(\omega_0)z^{-1} + r^2z^{-2}$$

$$H_2(z) = \frac{1}{H_1(z)}$$

Donde  $H_1(z)$  y  $H_2(z)$  son sistemas que contienen sólo ceros, o sólo polos, respectivamente, cuya posición viene dada por los parámetros  $r$  y  $w_0$ .

1. Creamos una función que determine la respuesta impulsional (impz), la respuesta en frecuencia (freqz) y la posición de los ceros y polos de la transformada Z para el sistema descrito por  $H_1(z)$  y con parámetros de entrada  $r$  y  $w_0$ .

```
# Función para analizar H1(z)
filtro_H1 <- function(r, w0, N_imp = 50) {

# Coeficientes
num <- c(1, -2 * r * cos(w0), r^2) # numerador
denom <- 1                         # FIR

cat("\n===== \n")
cat("Análisis de H1(z) con r =", r, "y w0 =", w0, "\n")
cat("Numerador (b):", num, "\n")
cat("Denominador (a):", denom, "\n")

# Respuesta impulsional
res_impulso <- impz(num, denom, N_imp)
plot(res_impulso,
     main = bquote("Respuesta al impulso para r =" ~ .(r) * ", w"[0] * " =" ~ .(w0)),
     xlab = "Muestras n",
     ylab = "h[n]")

# Respuesta en frecuencia
res_frecuencia <- freqz(num, denom)
plot(res_frecuencia,
     main = bquote("Respuesta en frecuencia para r =" ~ .(r) * ", w"[0] * " =" ~ .(w0)))

# Polos y ceros en el plano z
ceros <- polyroot(num)
polos <- numeric(0)

cat("Ceros de H1(z): \n")
print(ceros)
cat("Polos de H1(z): \n")
print(polos)

zplane(num, denom,
```

```

    main = bquote("Diagrama de polos y ceros de H"[1] * "(z) para r =" ~ .(r) * ", w"[0] * " =" ~

return(list(
  num = num,
  denom = denom,
  impulso = res_impulso,
  frecuencia = res_frecuencia,
  ceros = ceros,
  polos = polos
))
}

```

Efecto de  $r$  en la respuesta del filtro: Analiza el efecto del parámetro  $r$  en la respuesta en frecuencia del filtro. Para ello, ejecuta la función para diferentes valores de  $r$  (mayores, cercanos y menores que la unidad) fijando  $w_0 = 0$ .

```

# Efecto de r en la respuesta en frecuencia fijando w0
w0_fijo <- 0

for (r in c(0.5, 0.9, 1.0, 1.1)) {
  resultado <- filtro_H1(r = r, w0 = w0_fijo)
}

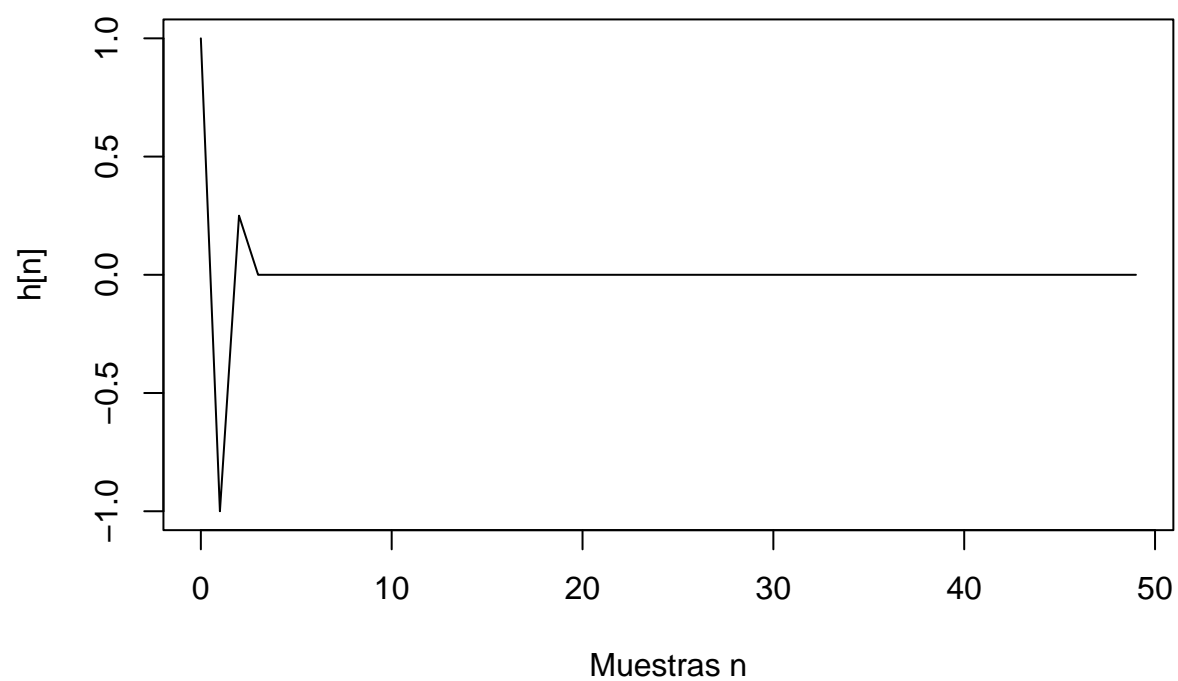
```

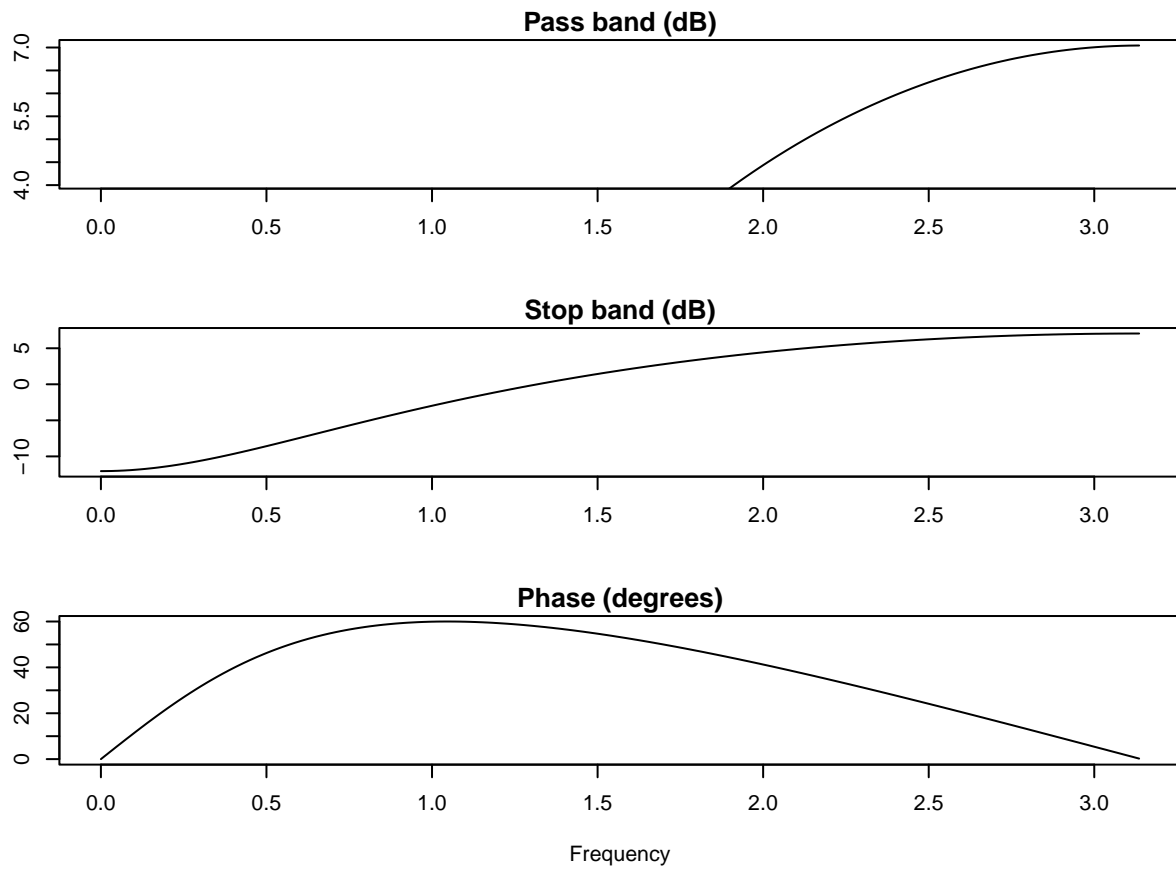
```

##
## =====
## Análisis de H1(z) con r = 0.5 y w0 = 0
## Numerador (b): 1 -1 0.25
## Denominador (a): 1

```

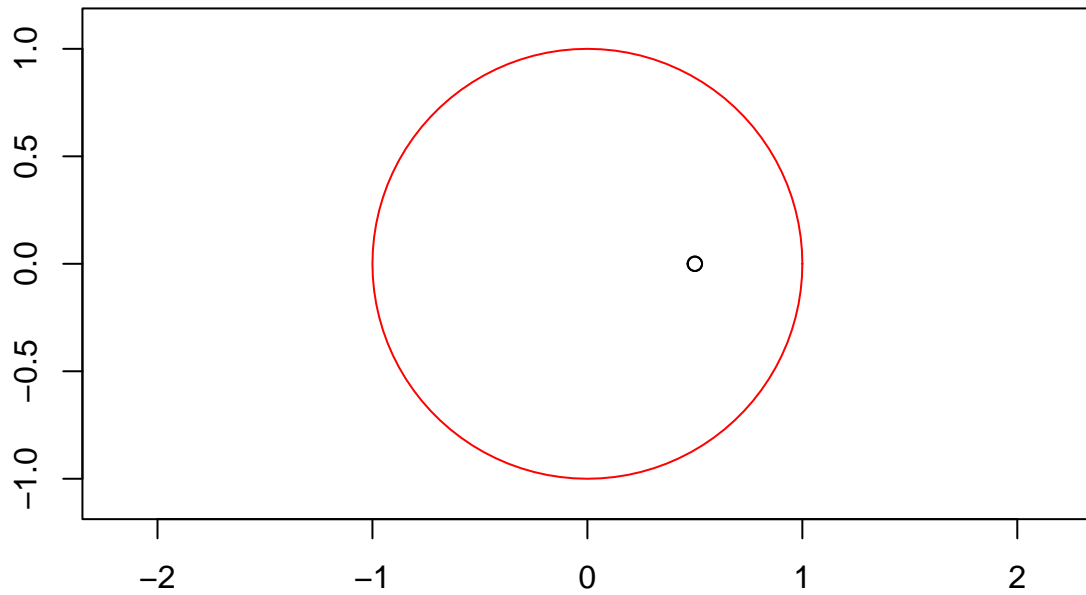
Respuesta al impulso para  $r = 0.5$ ,  $w_0 = 0$





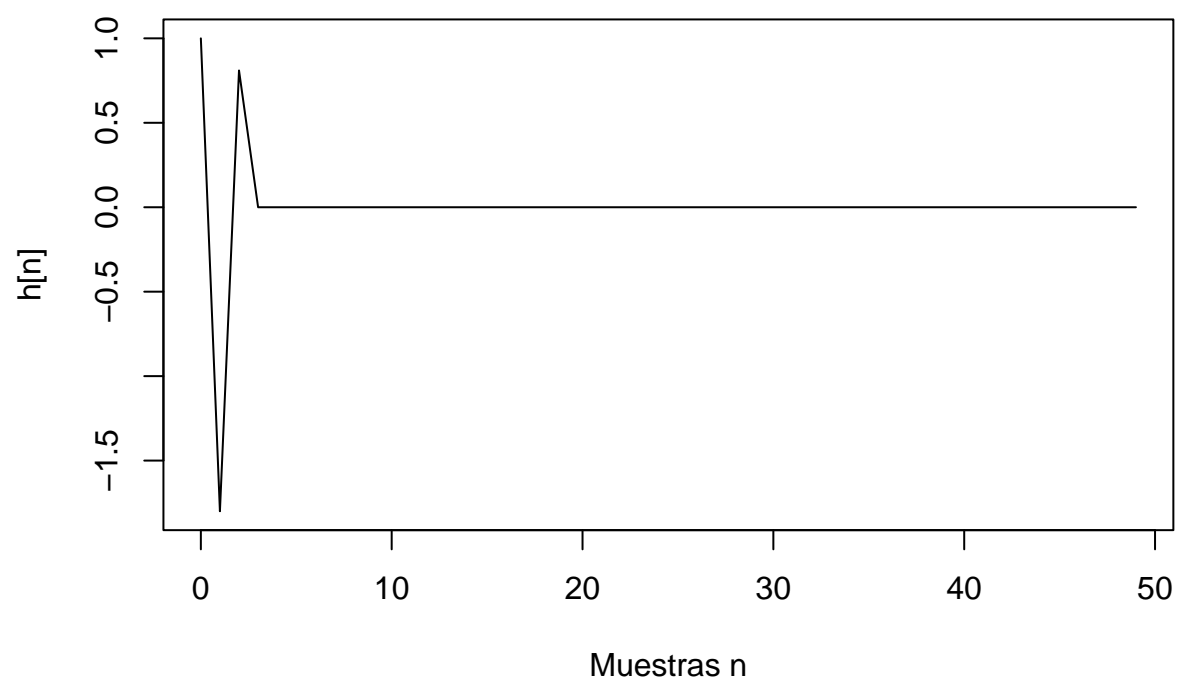
```
## Ceros de H1(z):
## [1] 2+1.110223e-16i 2-1.110223e-16i
## Polos de H1(z):
## numeric(0)
```

Diagrama de polos y ceros de  $H_1(z)$  para  $r = 0.5$ ,  $w_0 = 0$

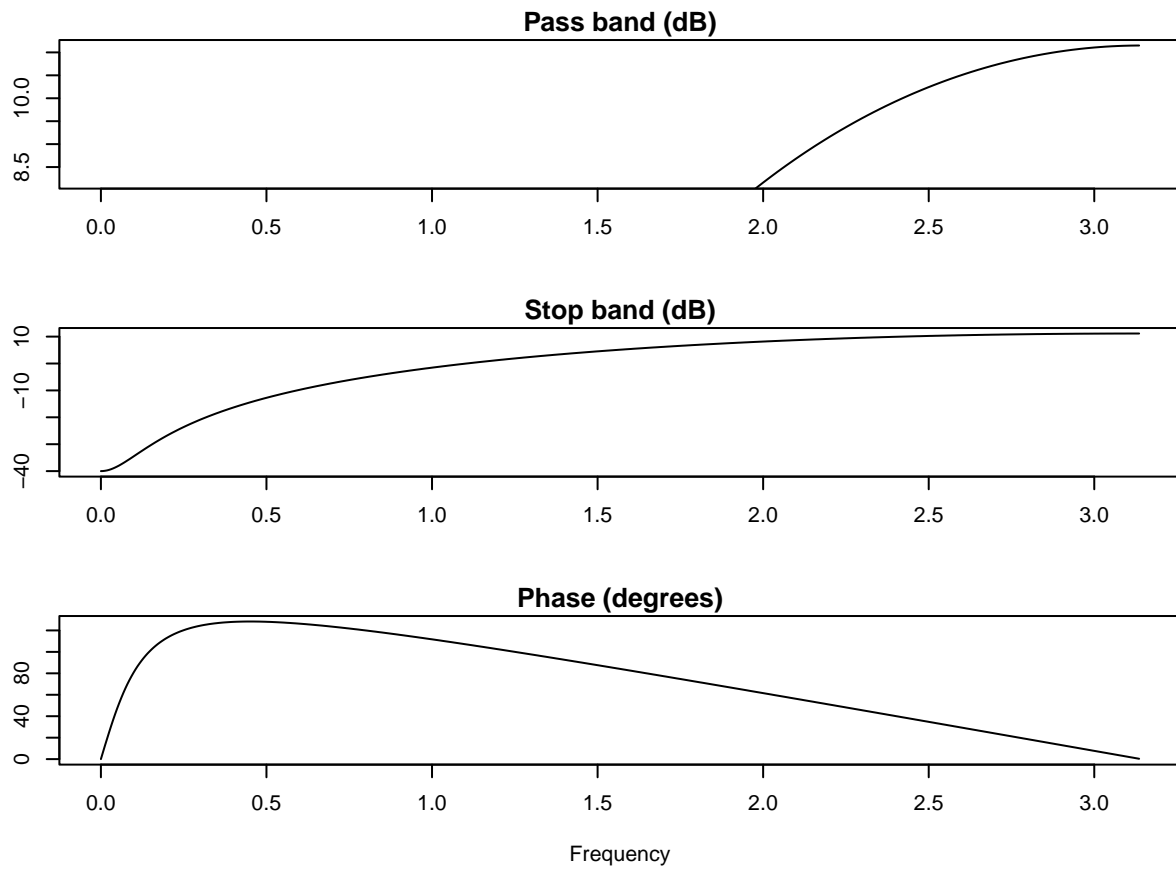


```
##
## =====
## Análisis de H1(z) con r = 0.9 y w0 = 0
## Numerador (b): 1 -1.8 0.81
## Denominador (a): 1
```

Respuesta al impulso para  $r = 0.9$ ,  $w_0 = 0$

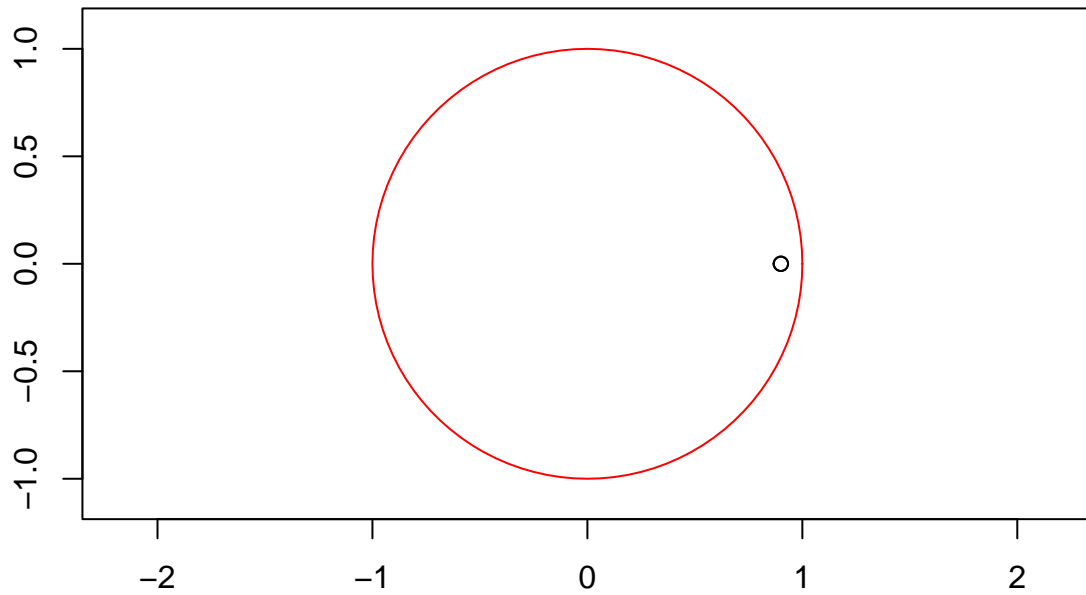






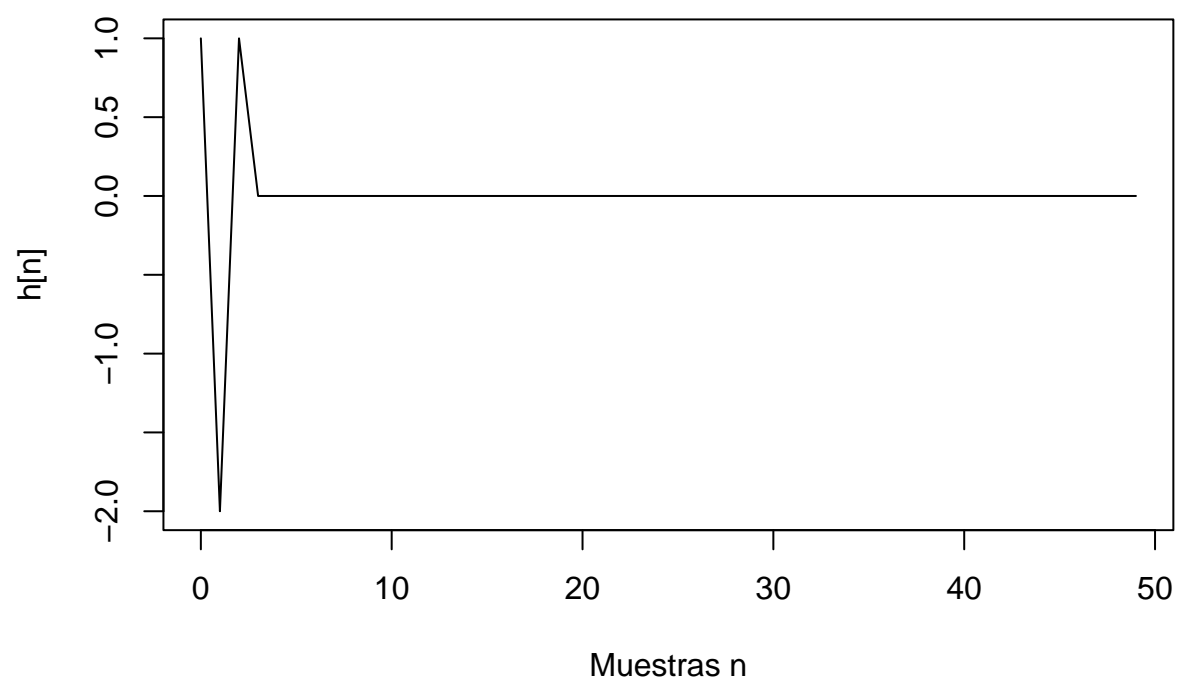
```
## Ceros de H1(z):
## [1] 1.111111-2.775558e-16i 1.111111+2.775558e-16i
## Polos de H1(z):
## numeric(0)
```

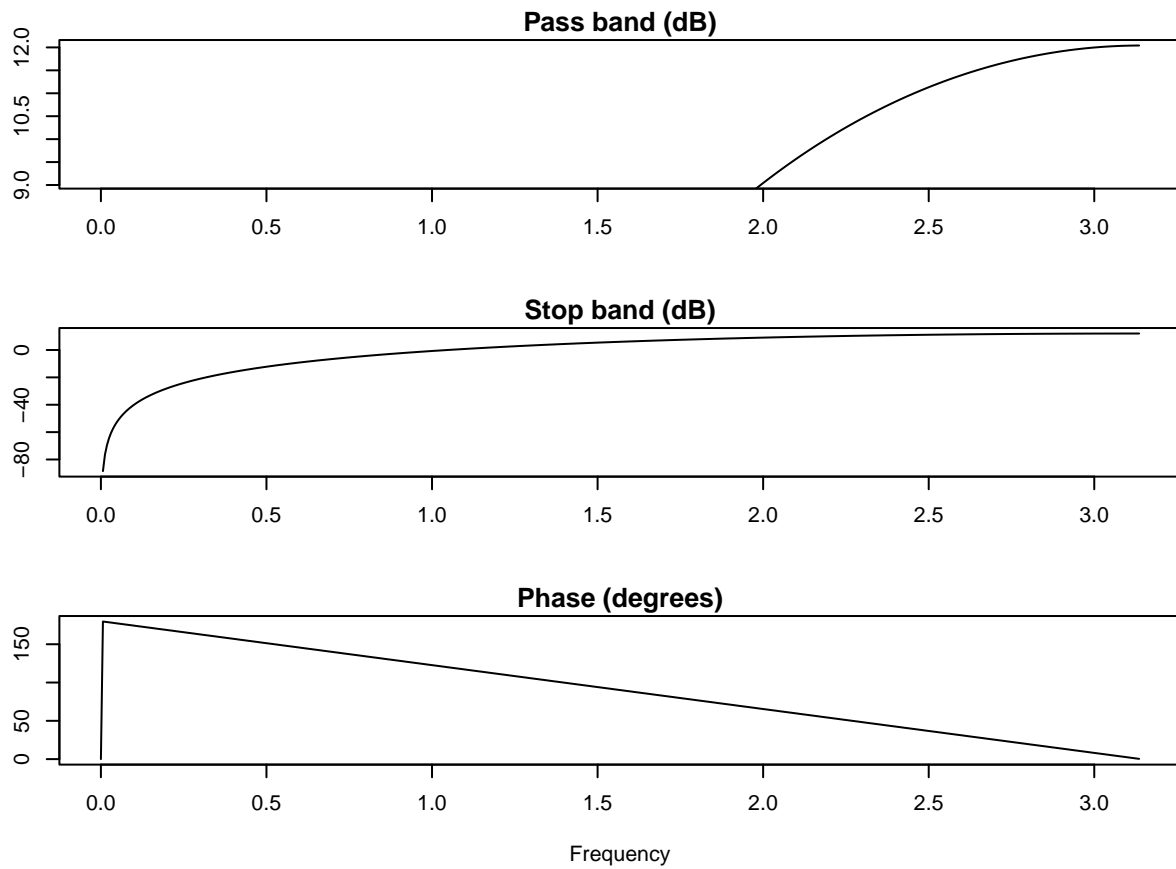
Diagrama de polos y ceros de  $H_1(z)$  para  $r = 0.9$ ,  $w_0 = 0$



```
##
## =====
## Análisis de H1(z) con r = 1 y w0 = 0
## Numerador (b): 1 -2 1
## Denominador (a): 1
```

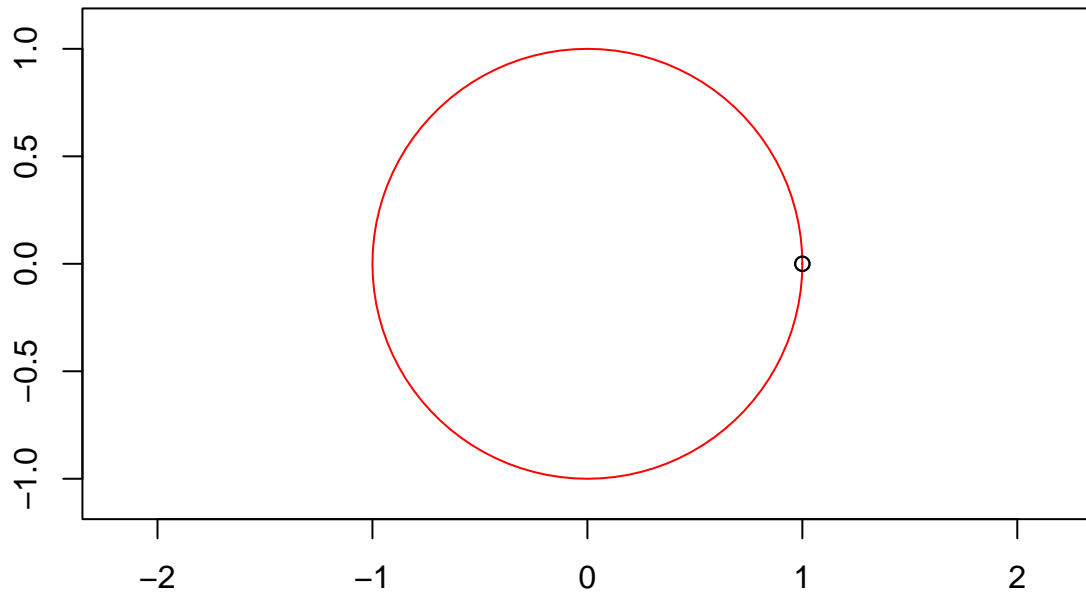
Respuesta al impulso para  $r = 1$ ,  $w_0 = 0$





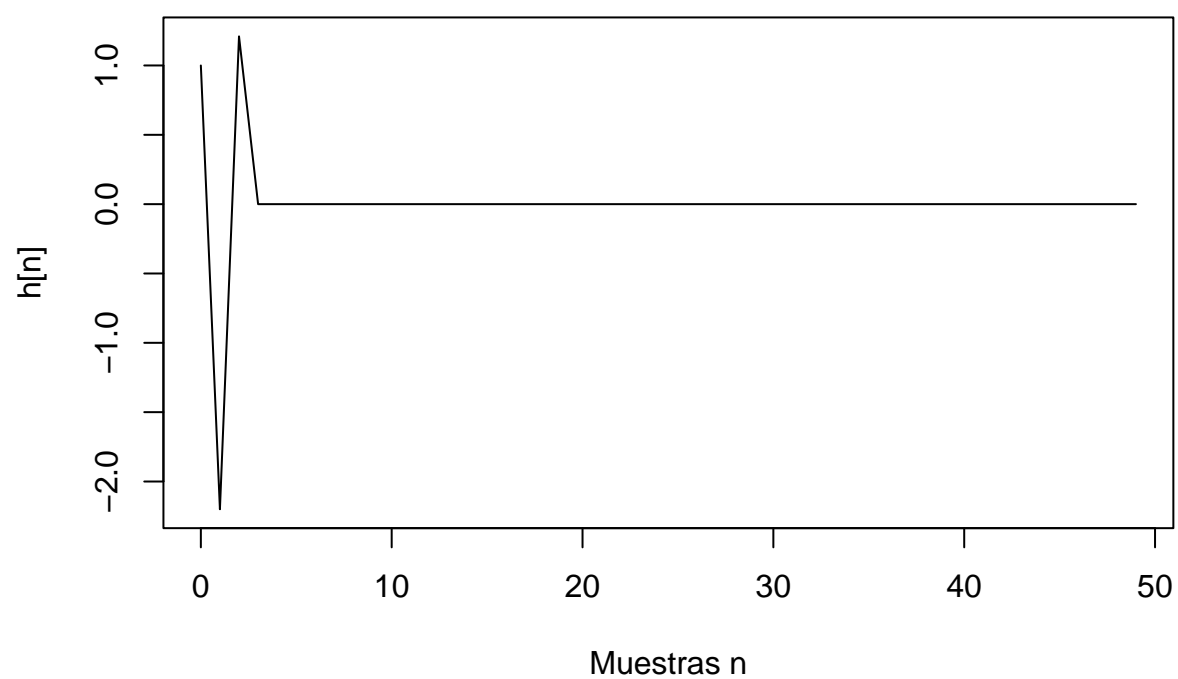
```
## Ceros de H1(z):
## [1] 1+5.551115e-17i 1-5.551115e-17i
## Polos de H1(z):
## numeric(0)
```

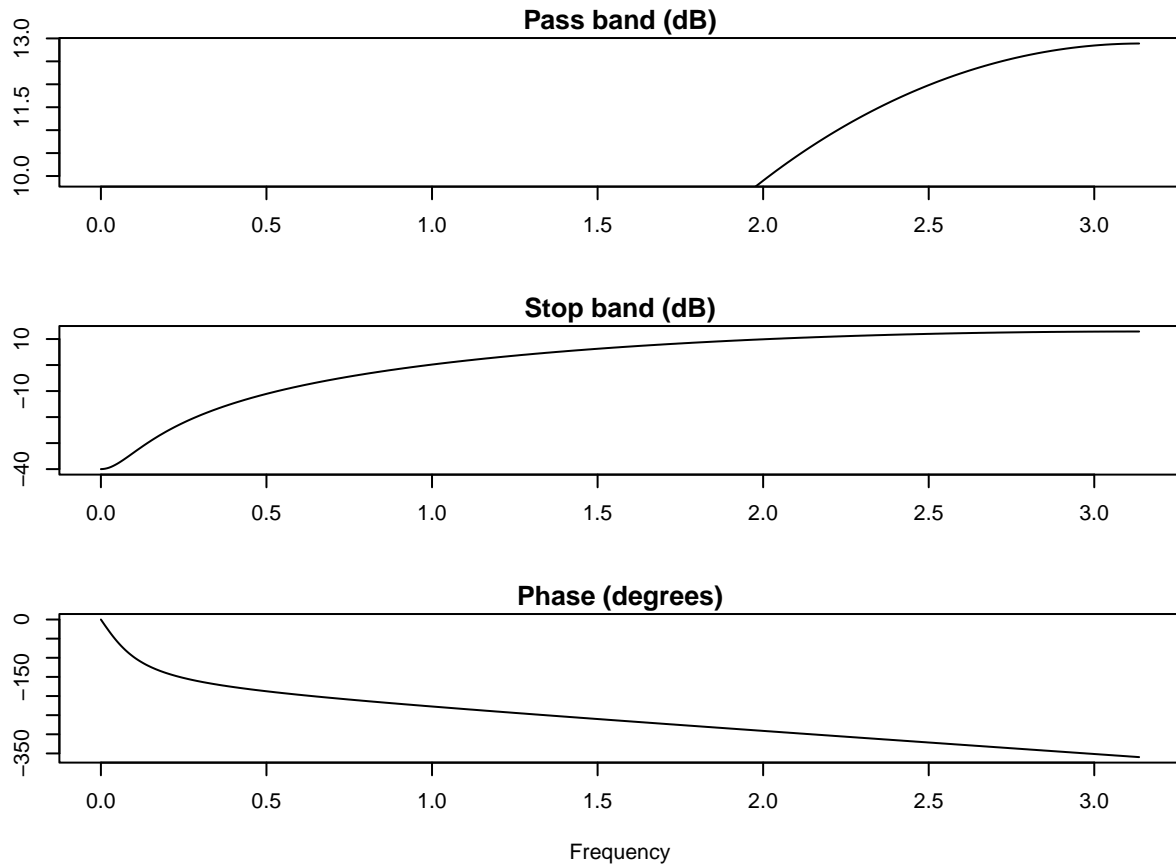
Diagrama de polos y ceros de  $H_1(z)$  para  $r = 1$ ,  $w_0 = 0$



```
##
## =====
## Análisis de H1(z) con r = 1.1 y w0 = 0
## Numerador (b): 1 -2.2 1.21
## Denominador (a): 1
```

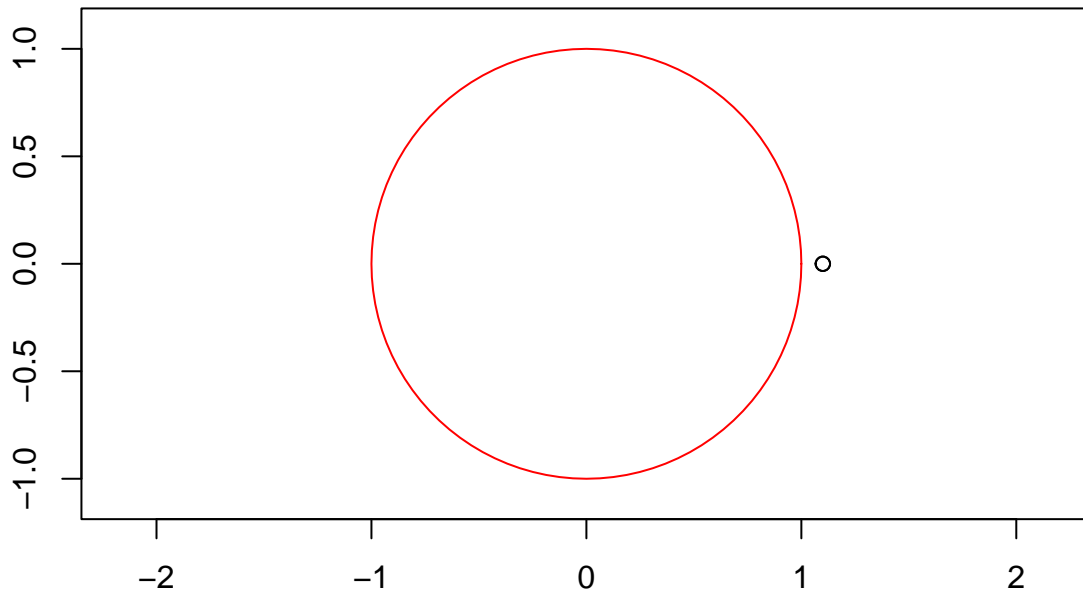
Respuesta al impulso para  $r = 1.1$ ,  $w_0 = 0$





```
## Ceros de H1(z):
## [1] 0.9090909+2.220446e-16i 0.9090909-2.220446e-16i
## Polos de H1(z):
## numeric(0)
```

Diagrama de polos y ceros de  $H_1(z)$  para  $r = 1.1$ ,  $w_0 = 0$



Cuando  $r < 1$ , los ceros quedan dentro de la circunferencia unidad y el notch es más suave. Al acercarse  $r$  a 1, los ceros se sitúan sobre la circunferencia unidad y el filtro presenta un notch más marcado y estrecho. Si  $r > 1$ , los ceros quedan fuera de la circunferencia unidad y el notch pierde claridad. En resumen,  $r$  controla lo fuerte que es el rechazo: cuanto más cerca está de 1, más definido es el mínimo.

**Nota:** el resultado `numeric(0)` indica que  $H_1(z)$  es un filtro FIR y no tiene polos finitos.

**Efecto de  $w_0$  en la respuesta del filtro:** Analiza el efecto del parámetro  $w_0$  en la respuesta del filtro. Para ello, ejecuta la función para diferentes valores de  $w_0$  ( $0, \pi/4, \pi$ ) fijando  $r = 0.9$ .

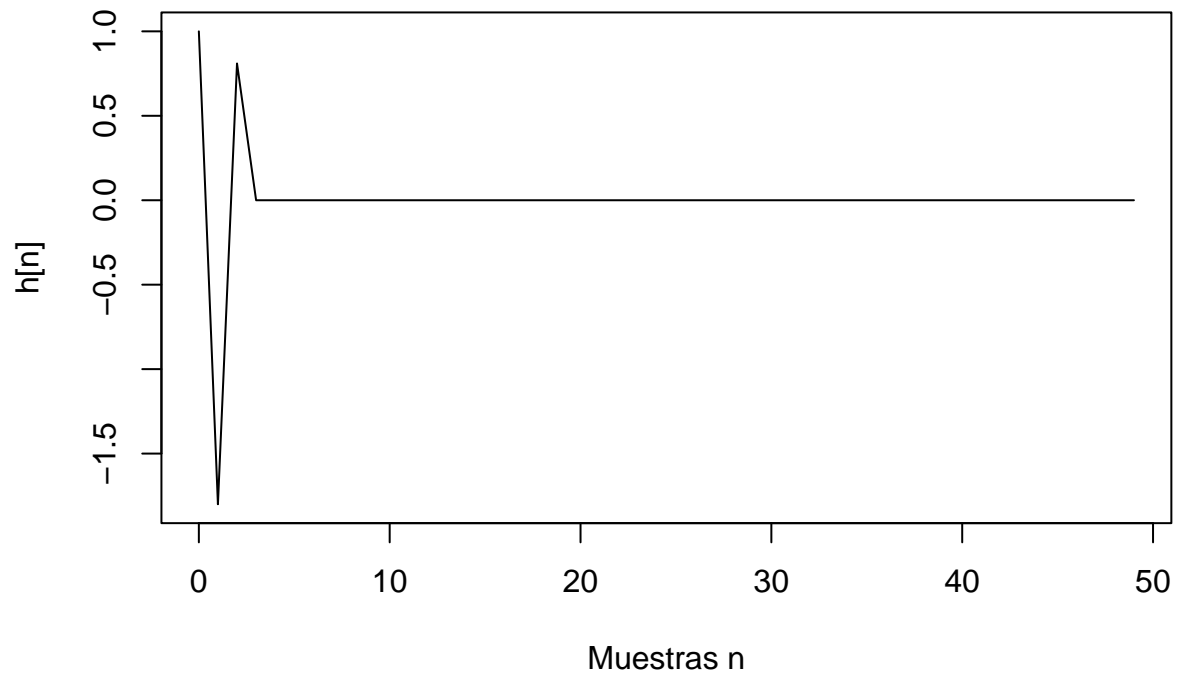
```
# Efecto de w0 en la respuesta del filtro fijando r
r_fijo <- 0.9

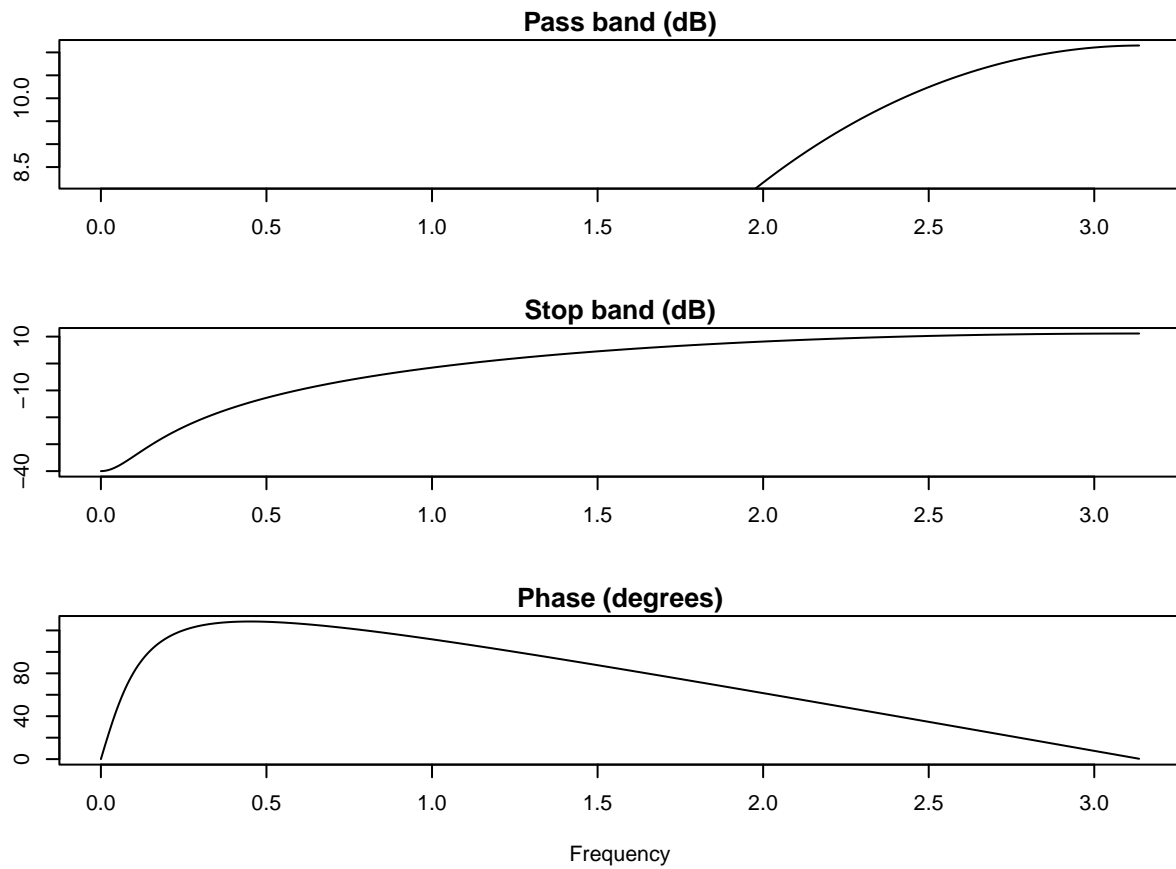
for (w0 in c(0, pi/4, pi)) {
  resultado <- filtro_H1(r = r_fijo, w0 = w0)
}
```

```
##
## =====
## Análisis de H1(z) con r = 0.9 y w0 = 0
## Numerador (b): 1 -1.8 0.81
## Denominador (a): 1
```



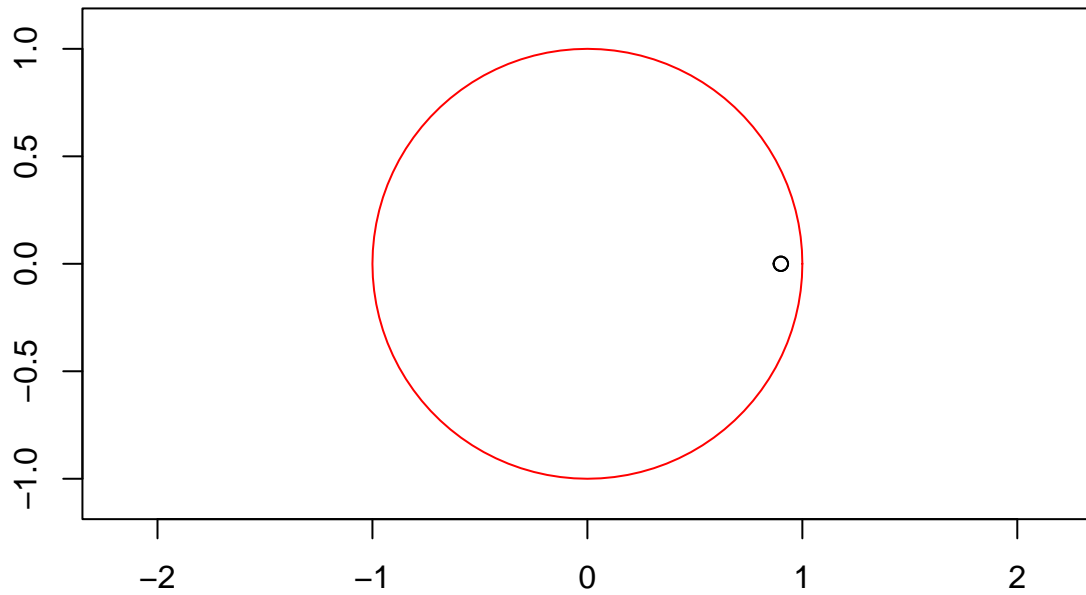
Respuesta al impulso para  $r = 0.9$ ,  $w_0 = 0$





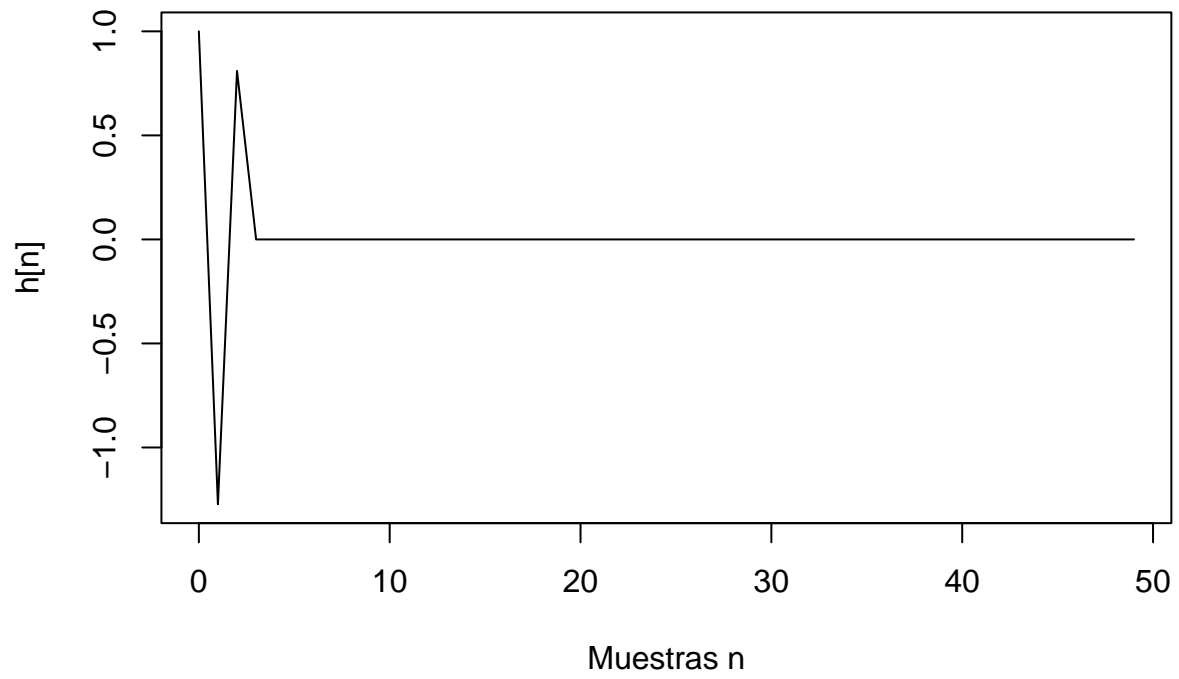
```
## Ceros de H1(z):
## [1] 1.111111-2.775558e-16i 1.111111+2.775558e-16i
## Polos de H1(z):
## numeric(0)
```

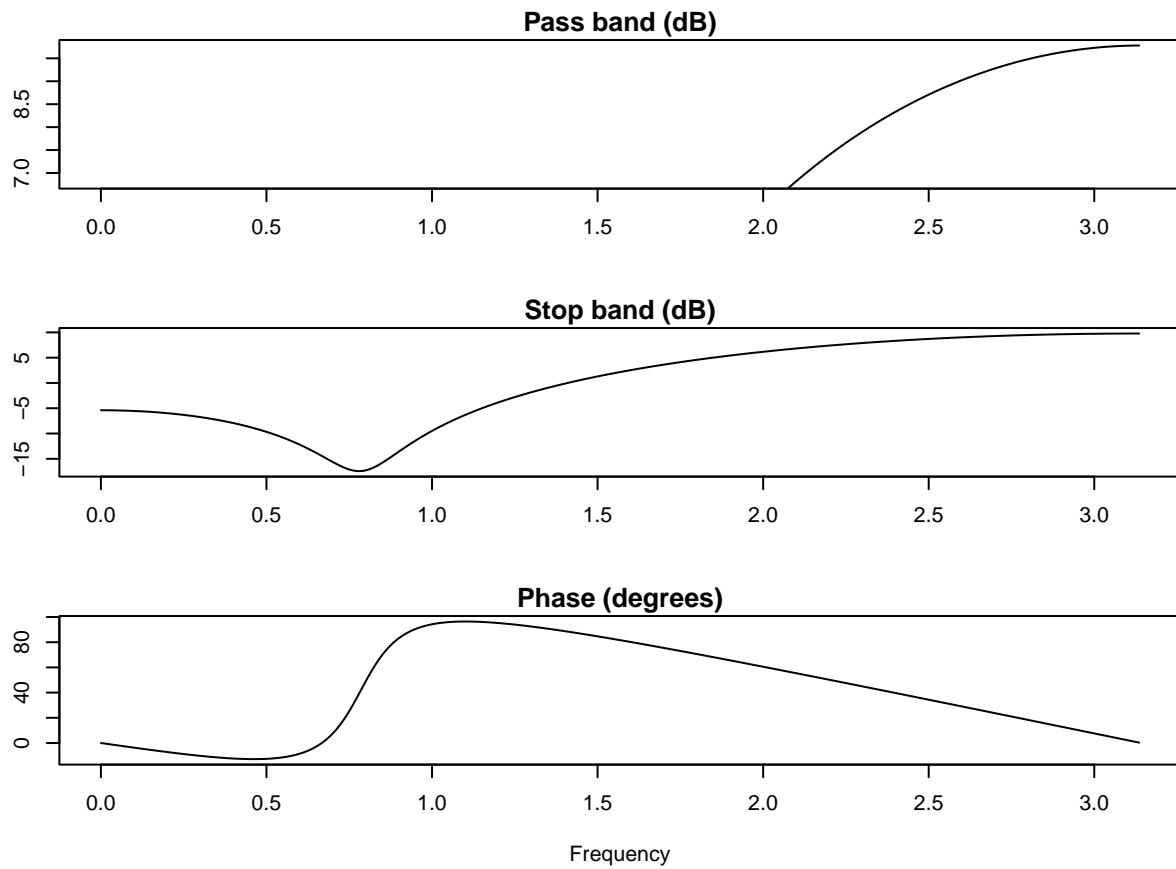
Diagrama de polos y ceros de  $H_1(z)$  para  $r = 0.9$ ,  $w_0 = 0$



```
##
## =====
## Análisis de H1(z) con r = 0.9 y w0 = 0.7853982
## Numerador (b): 1 -1.272792 0.81
## Denominador (a): 1
```

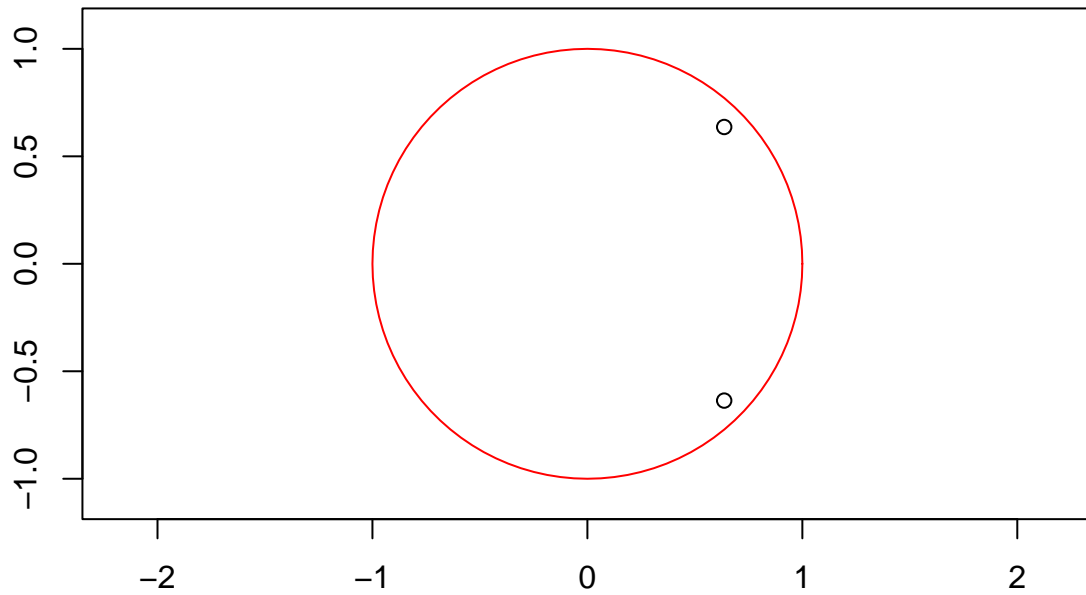
Respuesta al impulso para  $r = 0.9$ ,  $w_0 = 0.7853982$





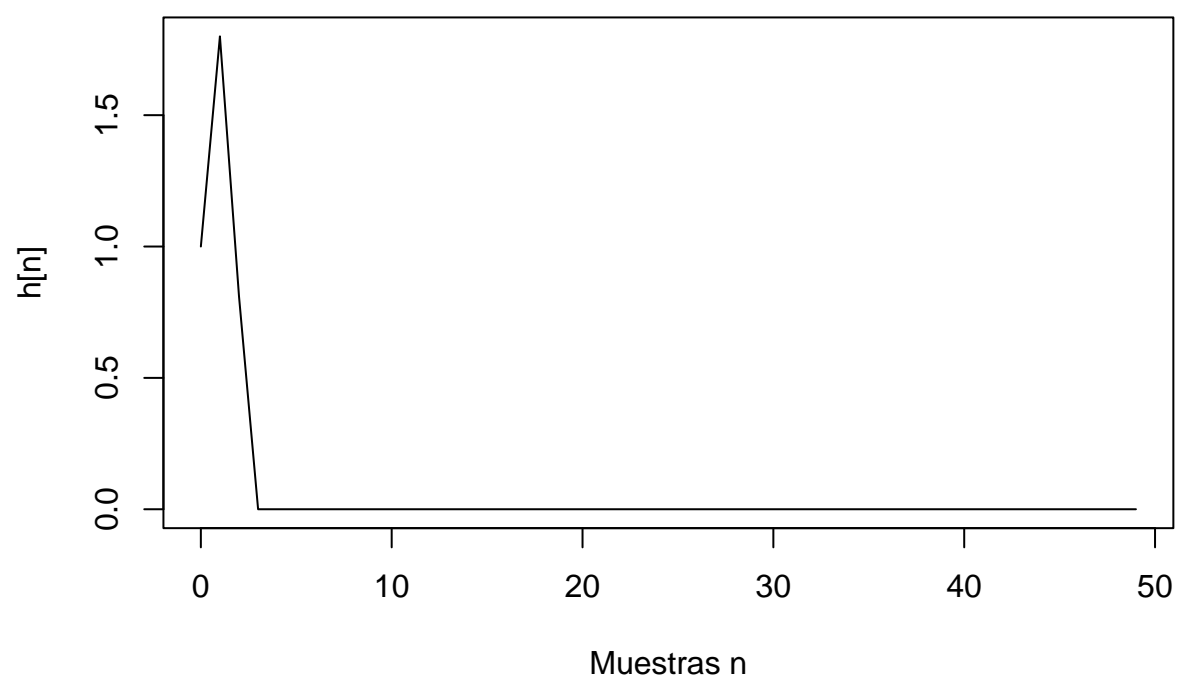
```
## Ceros de H1(z):
## [1] 0.7856742+0.7856742i 0.7856742-0.7856742i
## Polos de H1(z):
## numeric(0)
```

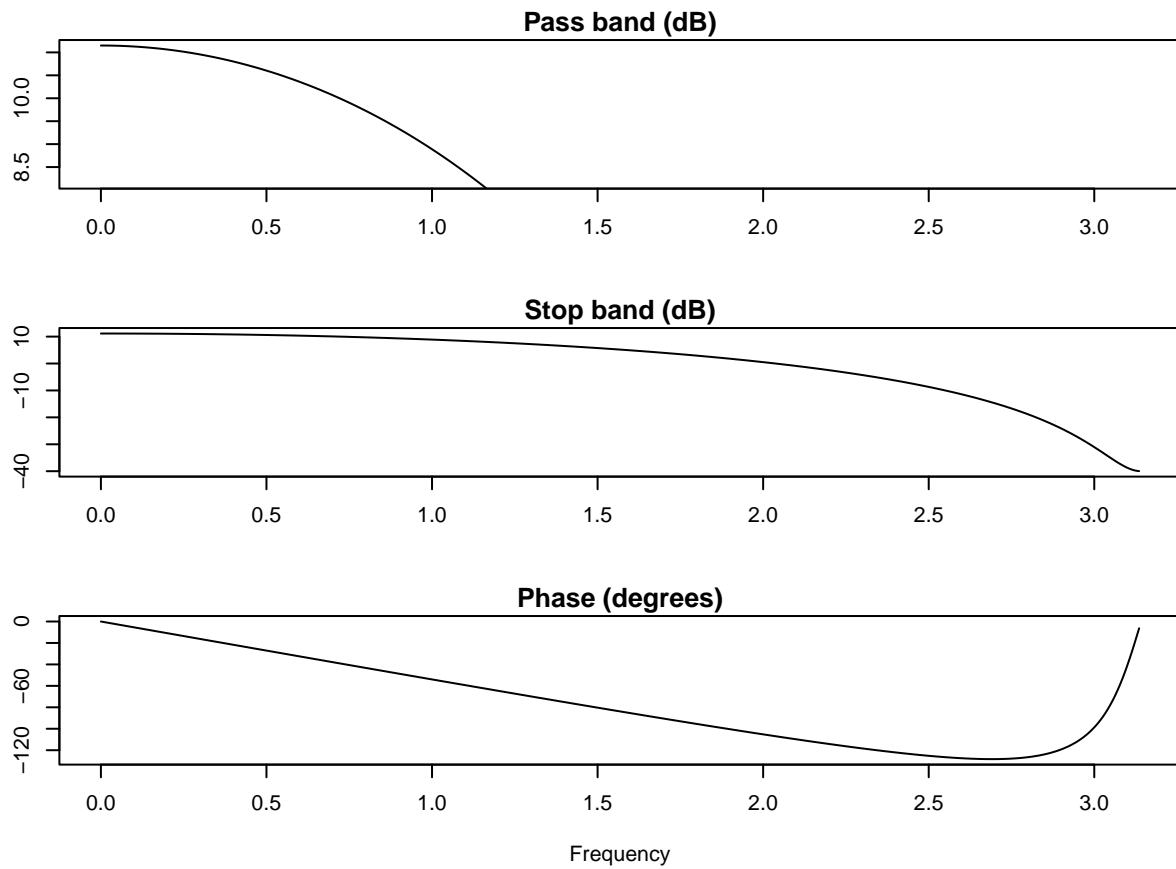
Diagrama de polos y ceros de  $H_1(z)$  para  $r = 0.9$ ,  $w_0 = 0.7853982$



```
##
## =====
## Análisis de H1(z) con r = 0.9 y w0 = 3.141593
## Numerador (b): 1 1.8 0.81
## Denominador (a): 1
```

Respuesta al impulso para  $r = 0.9$ ,  $w_0 = 3.141593$

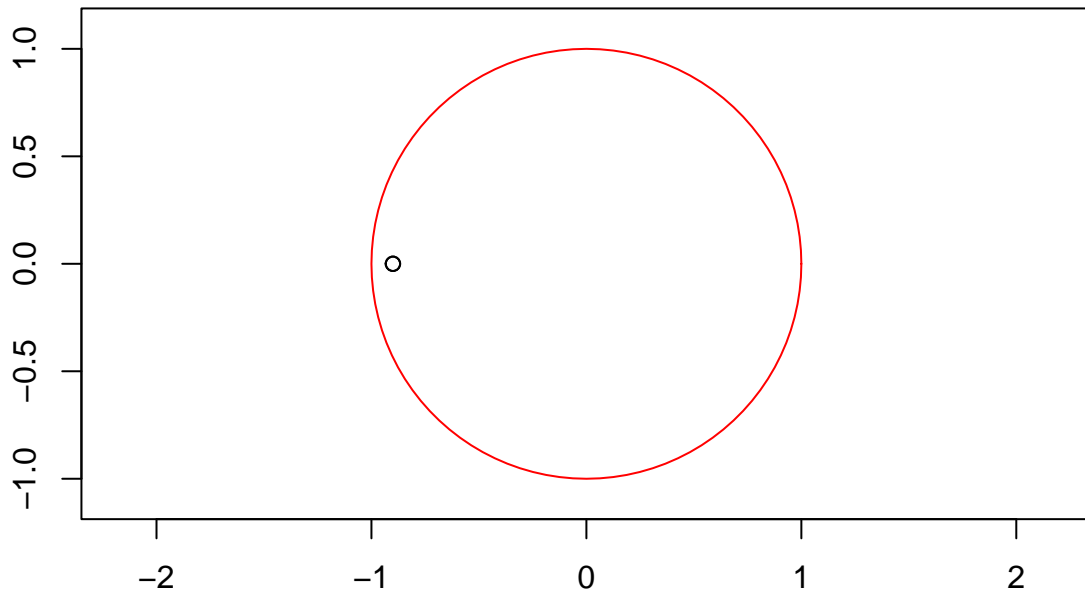




```
## Ceros de H1(z):
## [1] -1.111111-1.665335e-16i -1.111111+1.665335e-16i
## Polos de H1(z):
## numeric(0)
```



Diagrama de polos y ceros de  $H_1(z)$  para  $r = 0.9$ ,  $w_0 = 3.141593$



El resultado es que el notch en la respuesta en frecuencia simplemente se desplaza: con  $\omega_0 = 0$  aparece en baja frecuencia, con  $\pi/4$  en una frecuencia intermedia y con  $\pi$  en frecuencia alta. Así, mientras que  $r$  determina la intensidad del rechazo,  $\omega_0$  fija la frecuencia exacta donde ocurre.

### 2.1.1. Análisis del sistema $H_2(z)$

2. Crea una función para analizar el sistema descrito por  $H_2(z)$  y analiza el efecto de los parámetros  $r$  y  $w_0$  por separado, como en el caso anterior. ¿Qué observas?

```
# Función para analizar H2(z)
analiza_H2 <- function(r, w0) {

  # Coeficientes
  num <- 1
  denom <- c(1, -2 * r * cos(w0), r^2)

  # Respuesta impulsional
  res_impulso <- impz(num, denom, n = 50)

  # Respuesta en frecuencia
  res_frecuencia <- freqz(num, denom)

  # Ceros y polos
  ceros <- polyroot(num)      # no hay ceros = numeric(0)
  polos <- polyroot(denom)    # los polos son los ceros de H1(z)}
```

```

zplane(num, denom,
       main = paste("Polos y ceros de H_2(z)\n r =", r, ", w0 =", round(w0, 3)))

# Resultados
return(list(impulso = res_impulso,
           frecuencia = res_frecuencia,
           ceros = ceros,
           polos = polos))
}

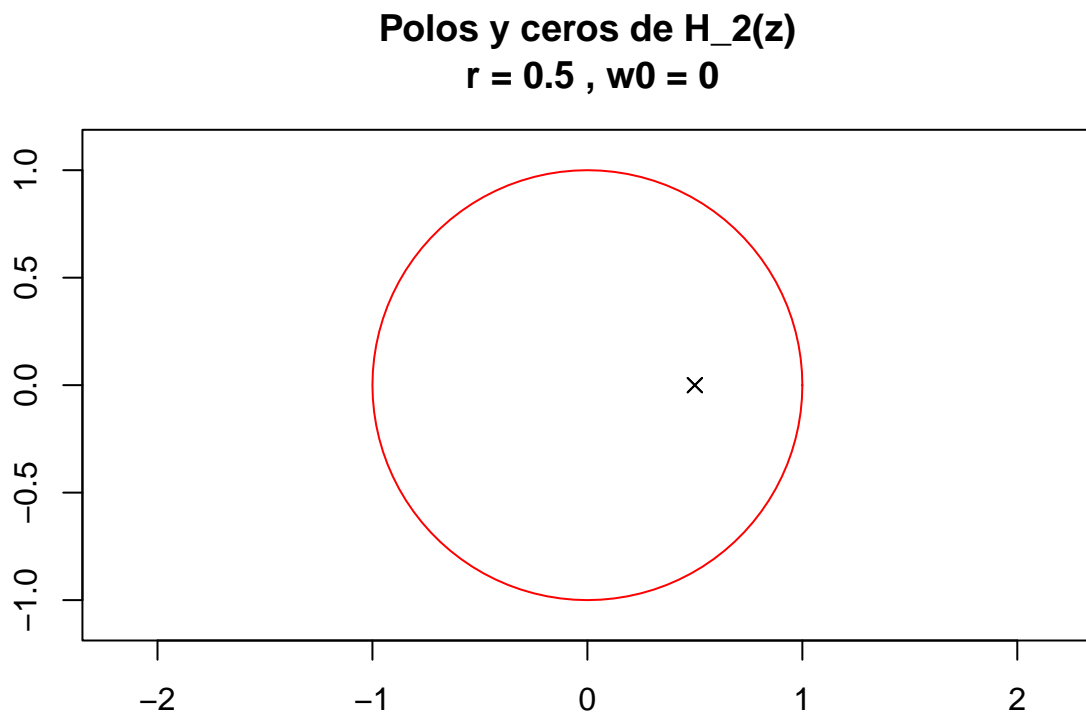
# Análisis del efecto de r con w0 = 0
w0_fijo <- 0
for (r in c(0.5, 0.9, 1.1)) {
  cat("\n===== \n")
  cat("Resultados para H2(z) con r =", r, "y w0 =", w0_fijo, "\n")
  print(analiza_H2(r, w0_fijo))
}

```

```

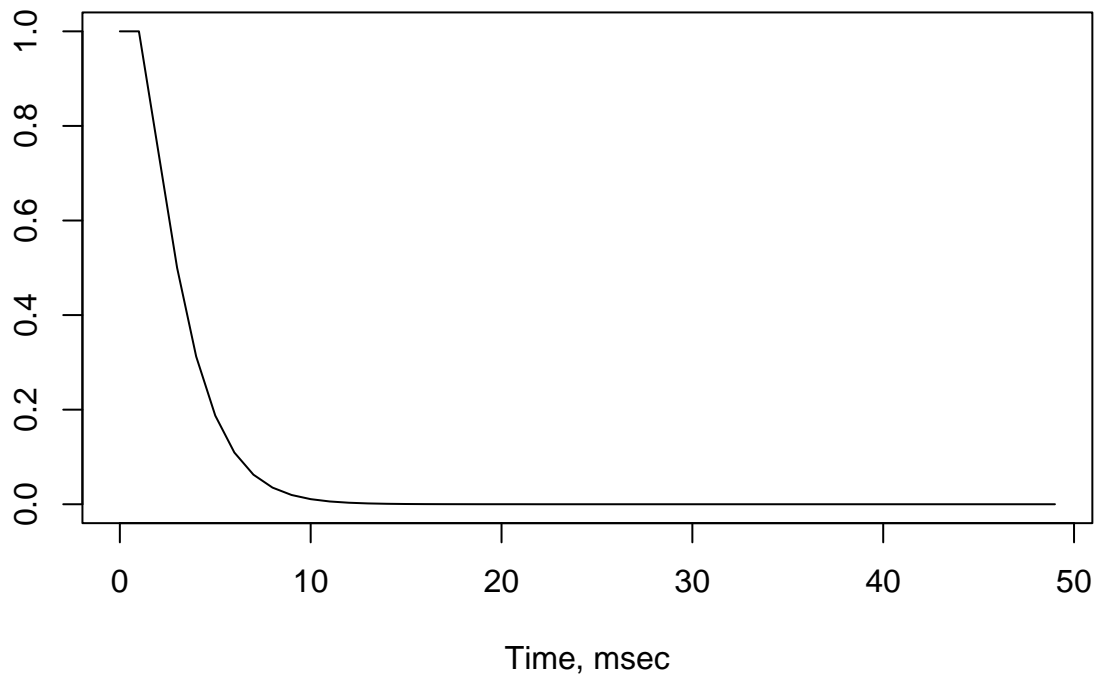
##
## =====
## Resultados para H2(z) con r = 0.5 y w0 = 0

```

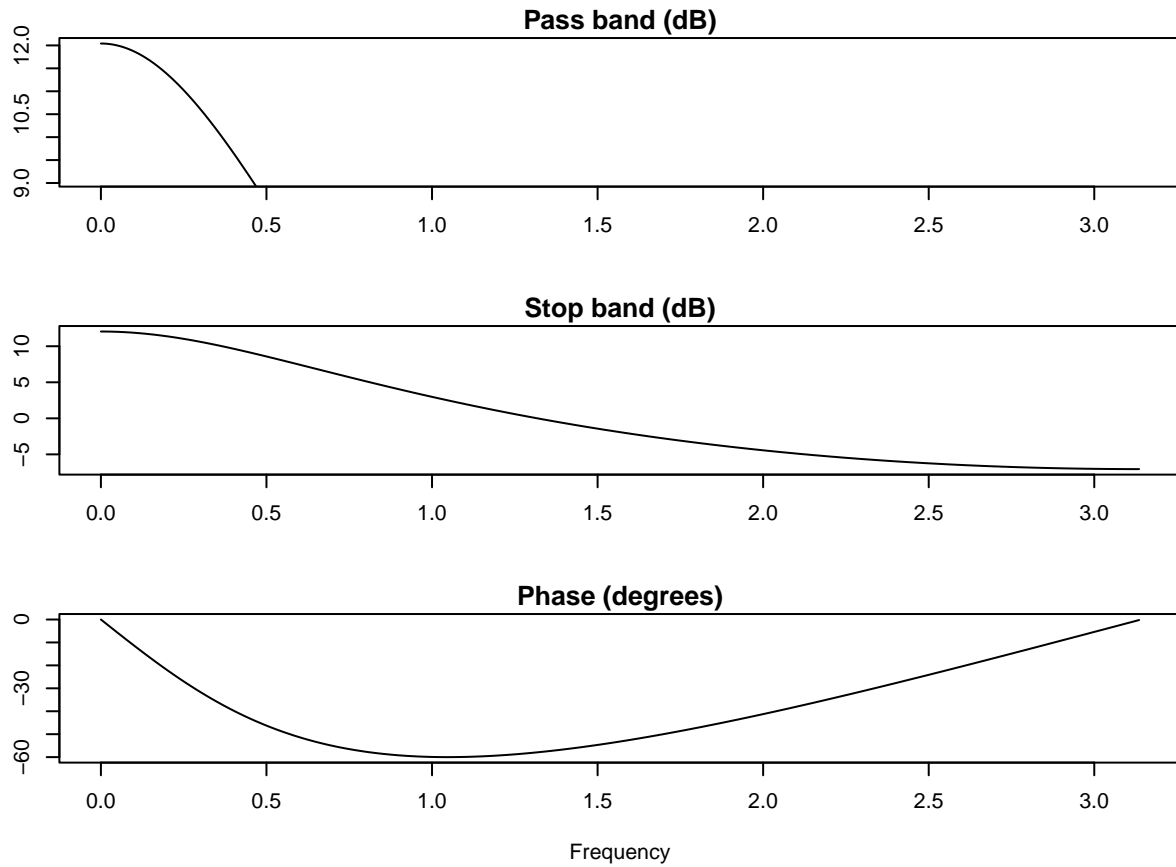


```
## $impulso
```

## Impulse response

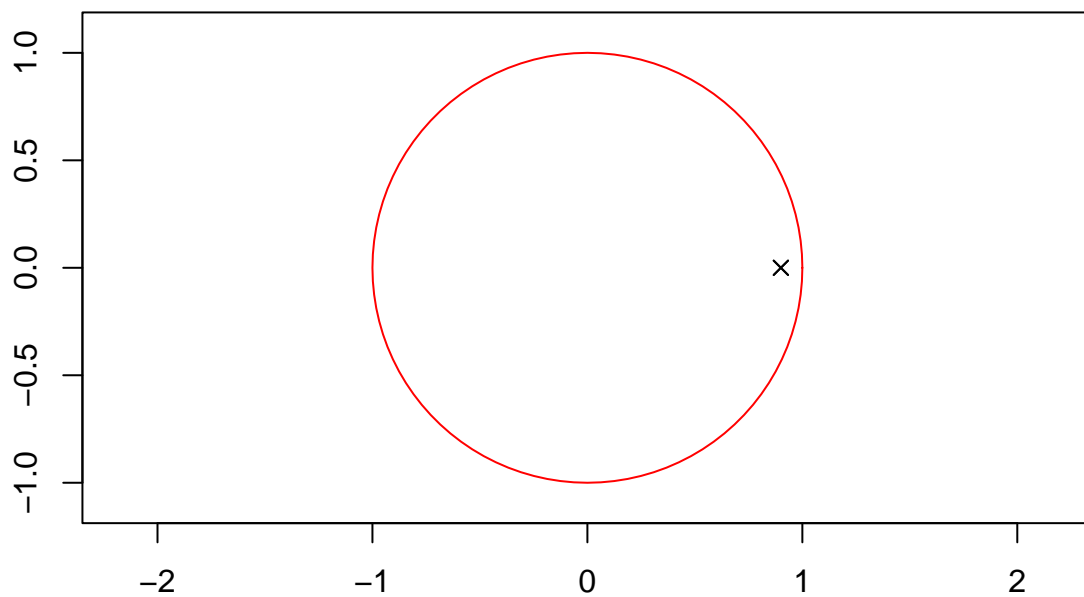


```
##  
## $frecuencia
```



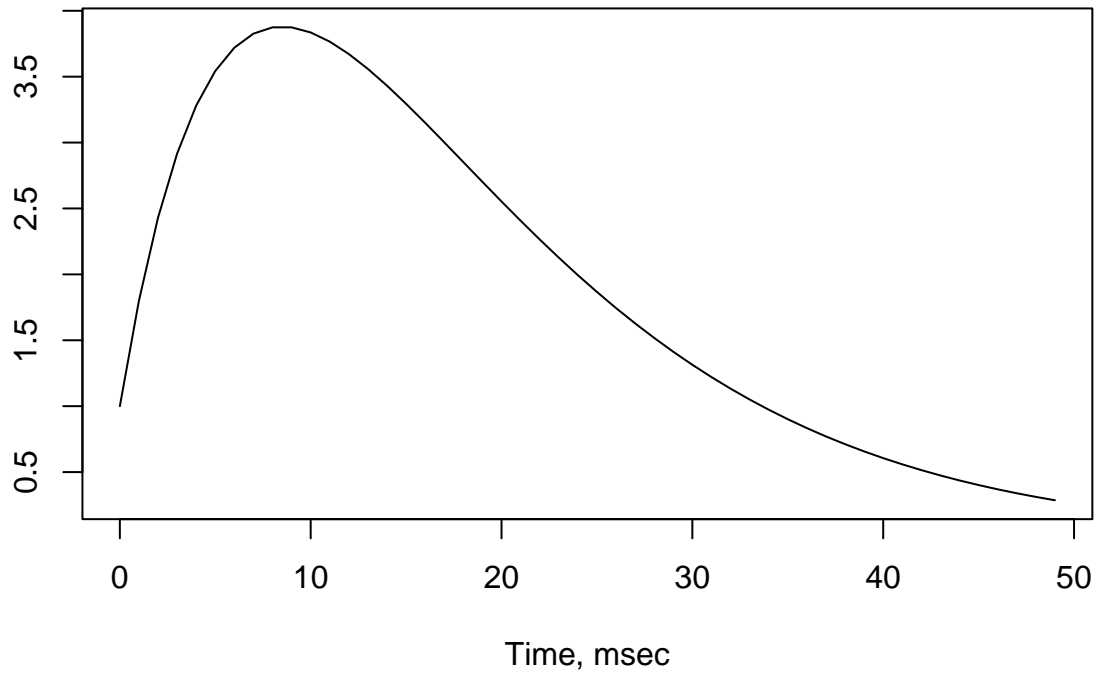
```
##
## $zeros
## complex(0)
##
## $polos
## [1] 2+1.110223e-16i 2-1.110223e-16i
##
##
## =====
## Resultados para H2(z) con r = 0.9 y w0 = 0
```

**Polos y ceros de  $H_2(z)$**   
 **$r = 0.9$  ,  $w_0 = 0$**

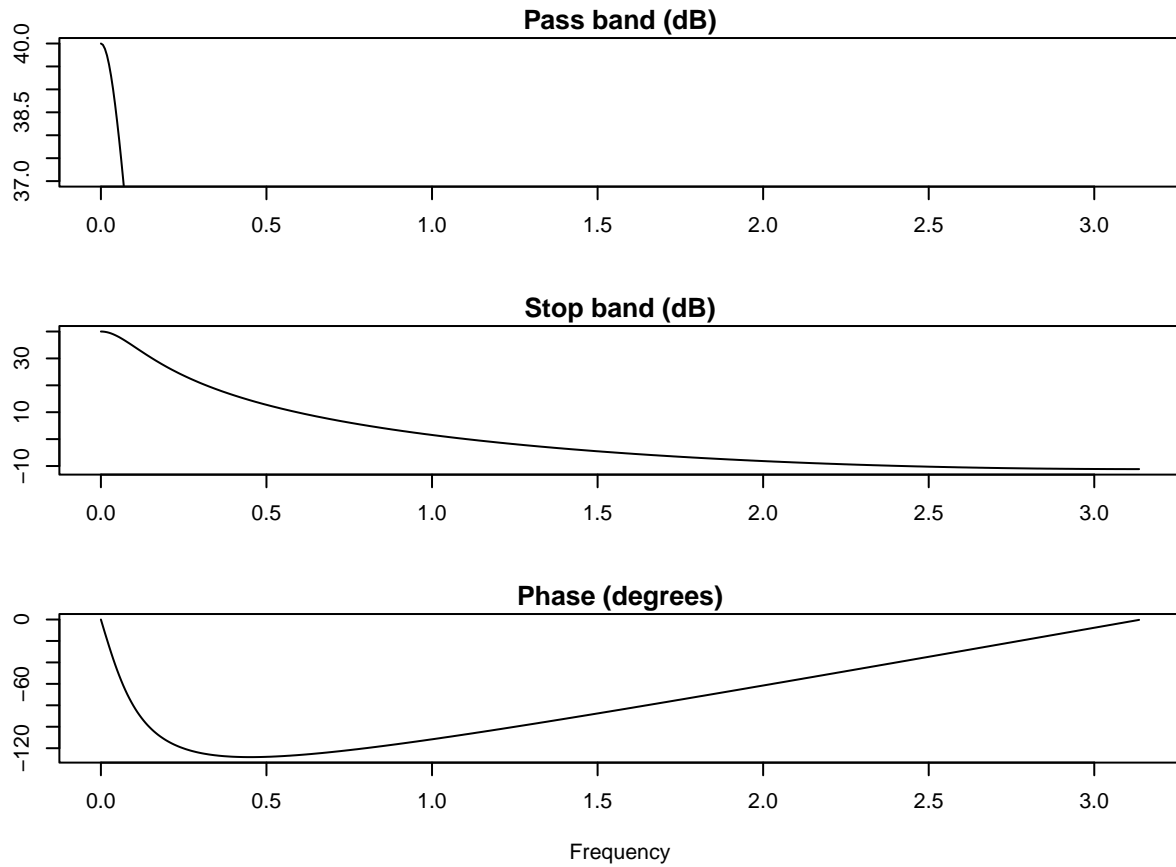


## \$impulso

## Impulse response

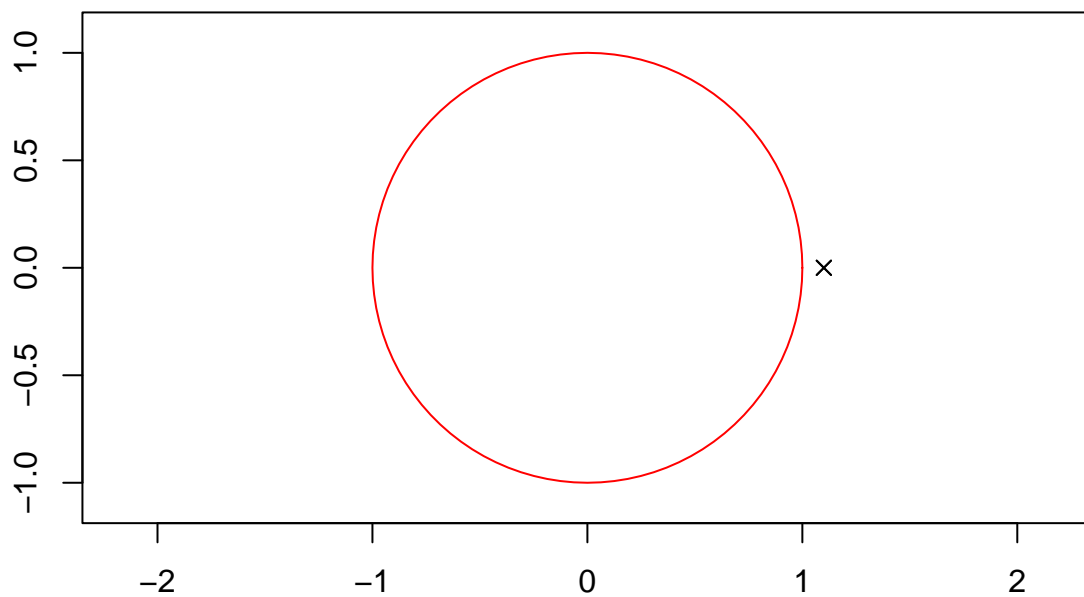


```
##  
## $frecuencia
```



```
##
## $ceros
## complex(0)
##
## $polos
## [1] 1.111111-2.775558e-16i 1.111111+2.775558e-16i
##
##
## =====
## Resultados para H2(z) con r = 1.1 y w0 = 0
```

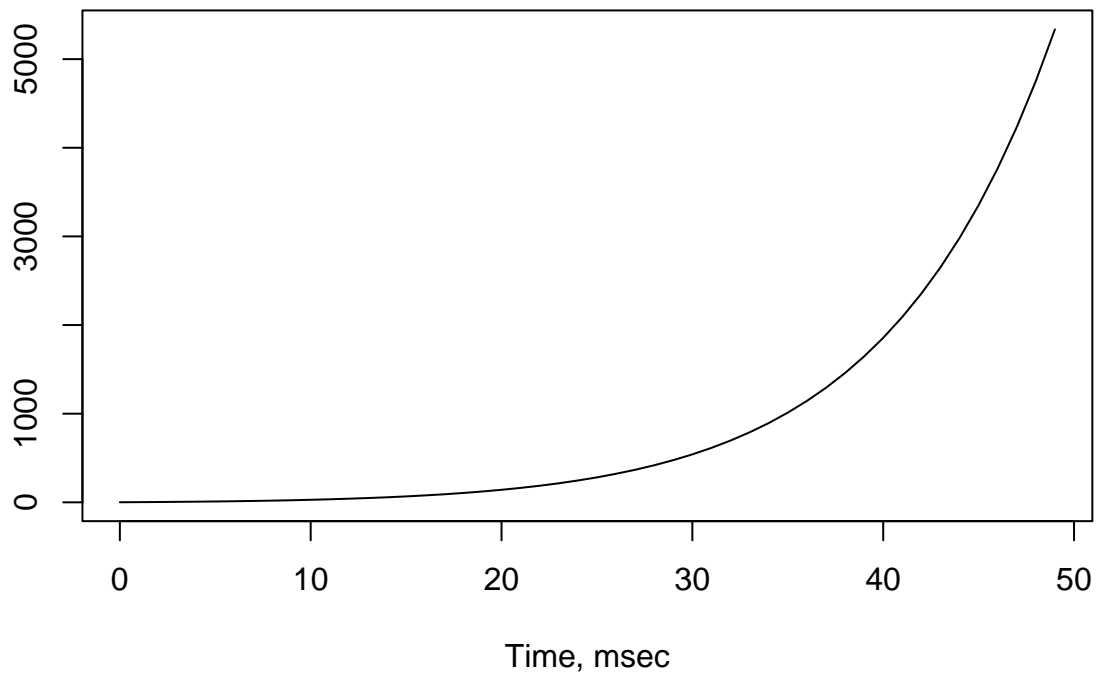
**Polos y ceros de  $H_2(z)$**   
 **$r = 1.1$  ,  $w_0 = 0$**



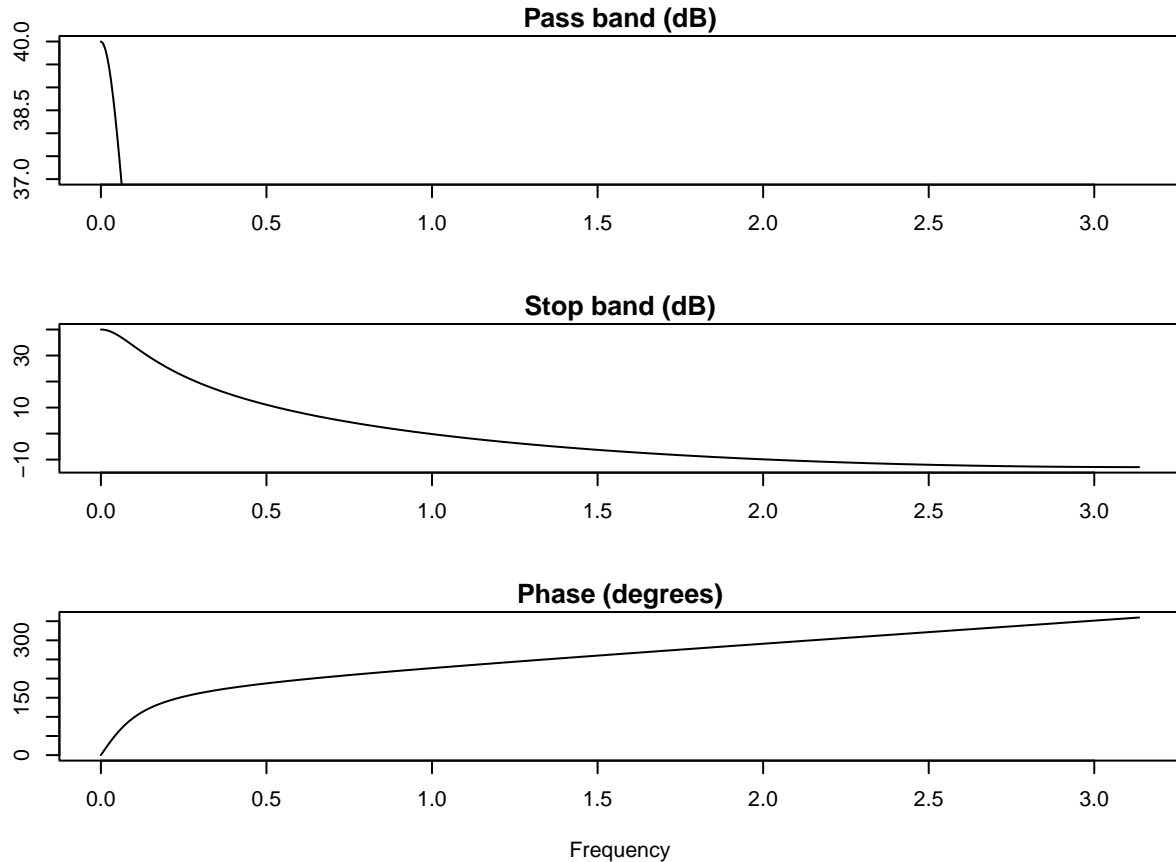
## \$impulso



## Impulse response



```
##  
## $frecuencia
```

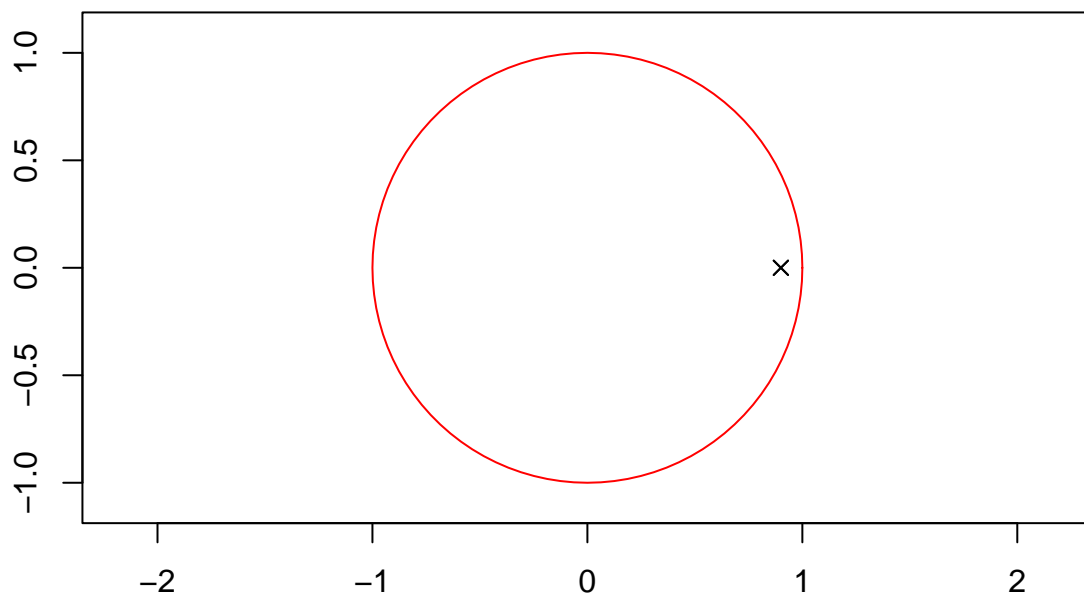


```
##
## $ceros
## complex(0)
##
## $polos
## [1] 0.9090909+2.220446e-16i 0.9090909-2.220446e-16i
```

```
# Análisis del efecto de w0 con r = 0.9
r_fijo <- 0.9
for (w0 in c(0, pi/4, pi)) {
  cat("\n===== \n")
  cat("Resultados para H2(z) con r =", r_fijo, "y w0 =", w0, "\n")
  print(analiza_H2(r_fijo, w0))
}
```

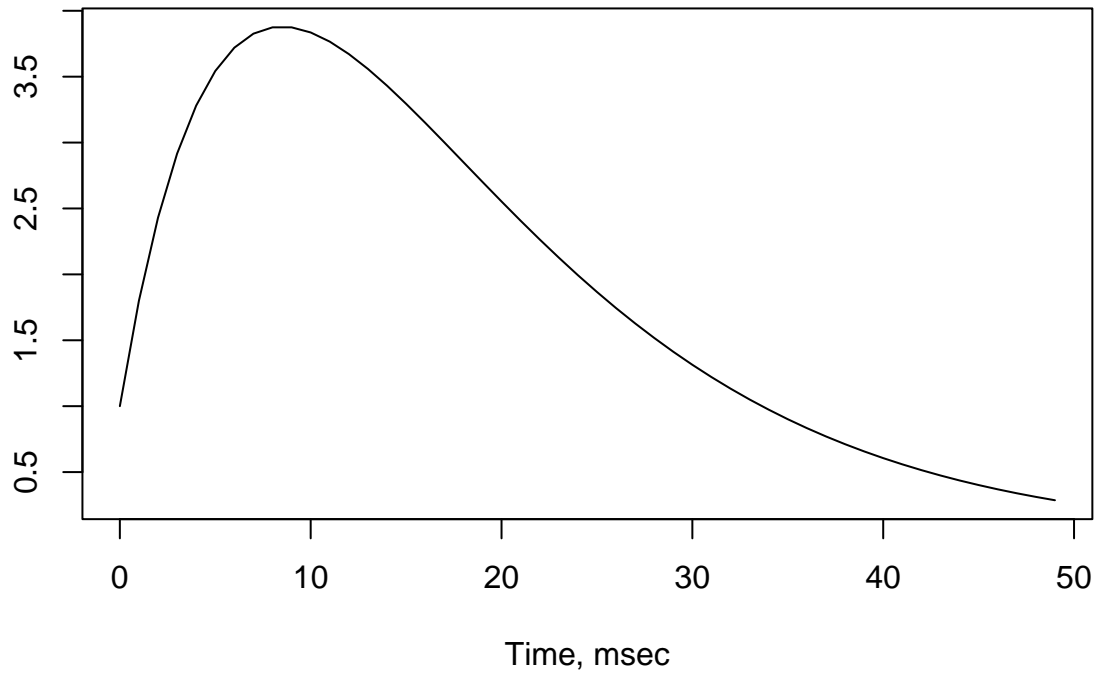
```
##
## =====
## Resultados para H2(z) con r = 0.9 y w0 = 0
```

**Polos y ceros de  $H_2(z)$**   
 **$r = 0.9$  ,  $w_0 = 0$**

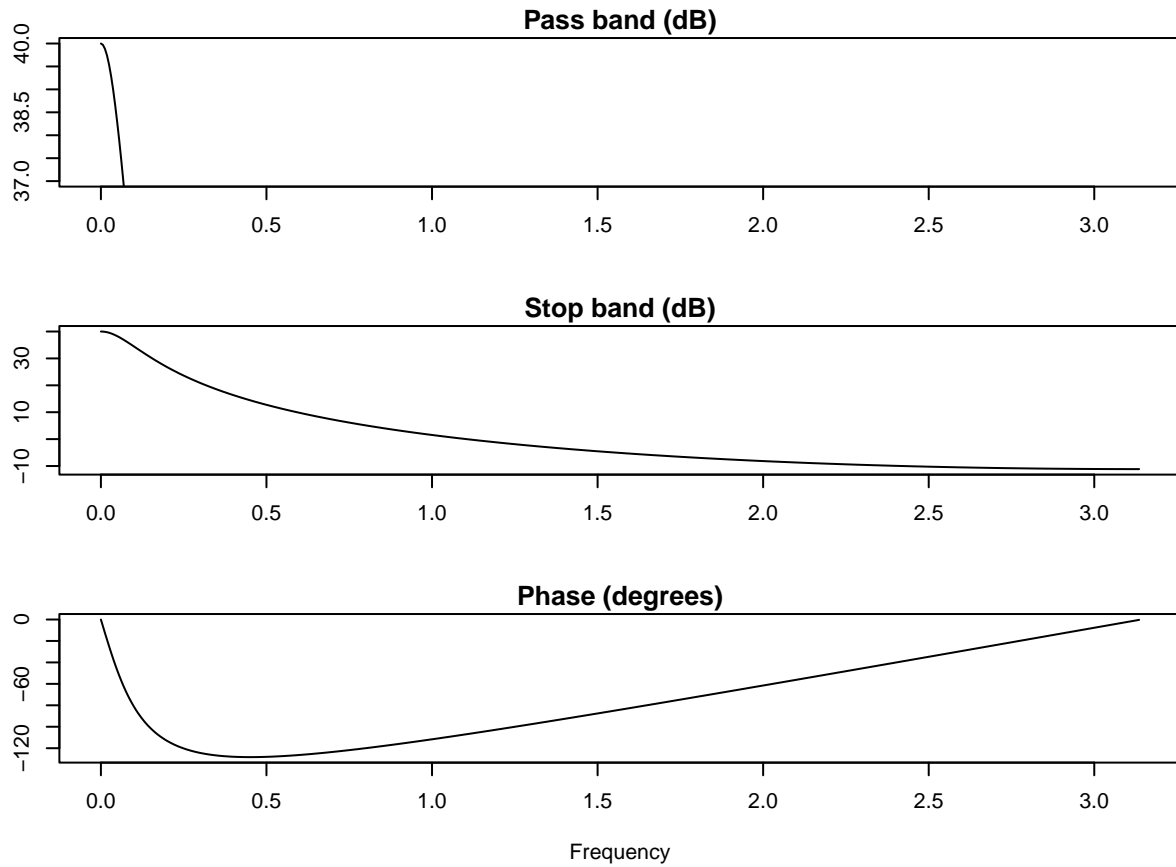


## \$impulso

## Impulse response

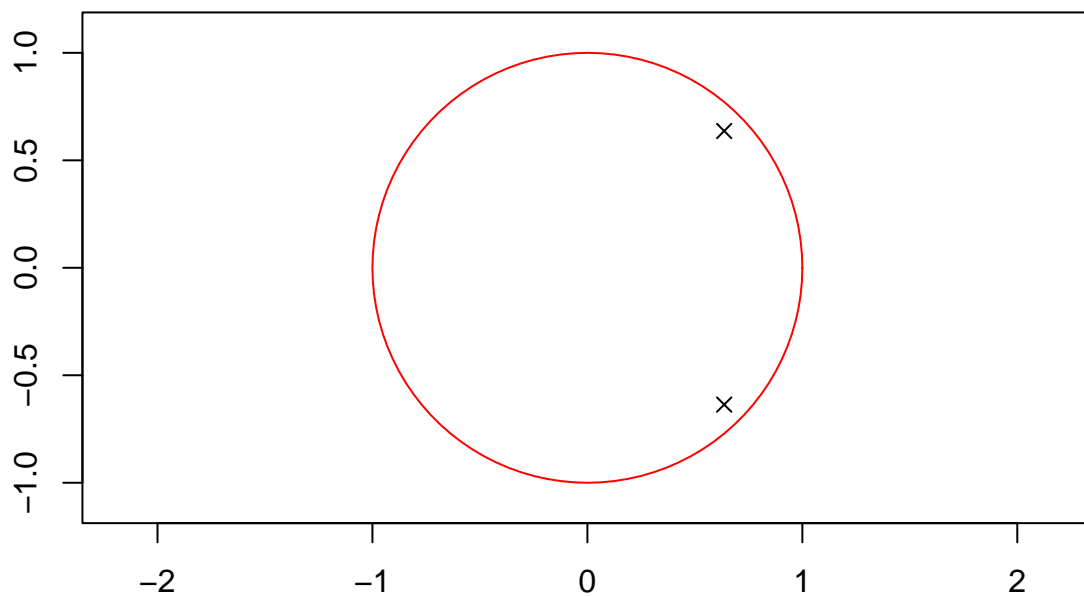


```
##  
## $frecuencia
```



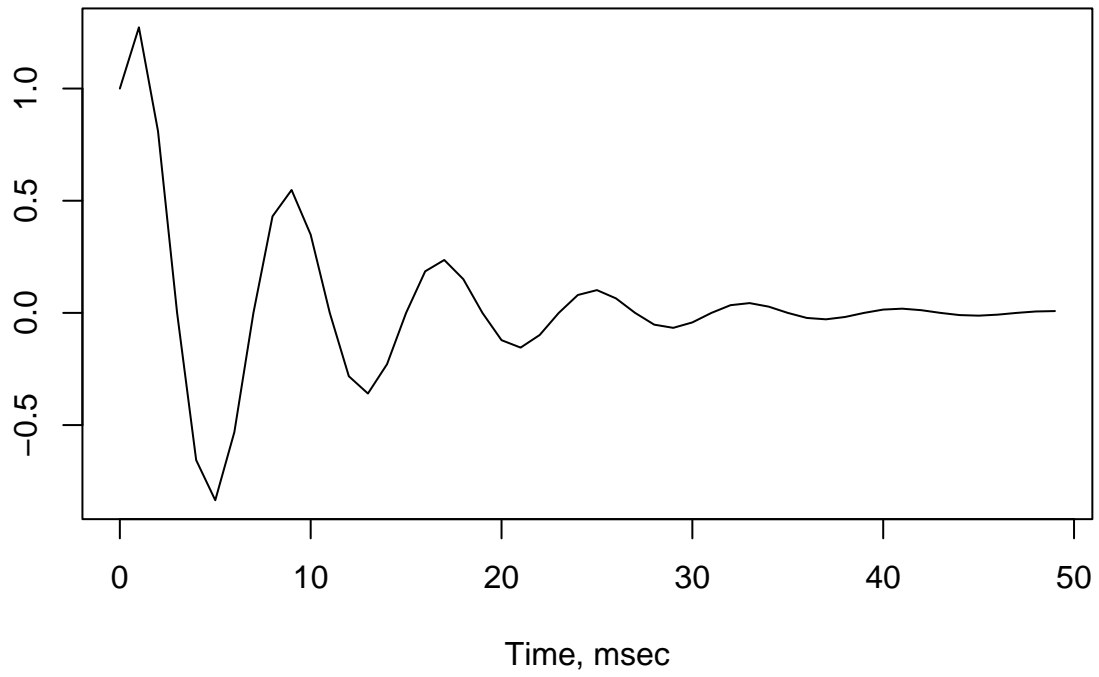
```
##
## $ceros
## complex(0)
##
## $polos
## [1] 1.111111-2.775558e-16i 1.111111+2.775558e-16i
##
##
## =====
## Resultados para H2(z) con r = 0.9 y w0 = 0.7853982
```

**Polos y ceros de  $H_2(z)$**   
 **$r = 0.9$  ,  $w_0 = 0.785$**

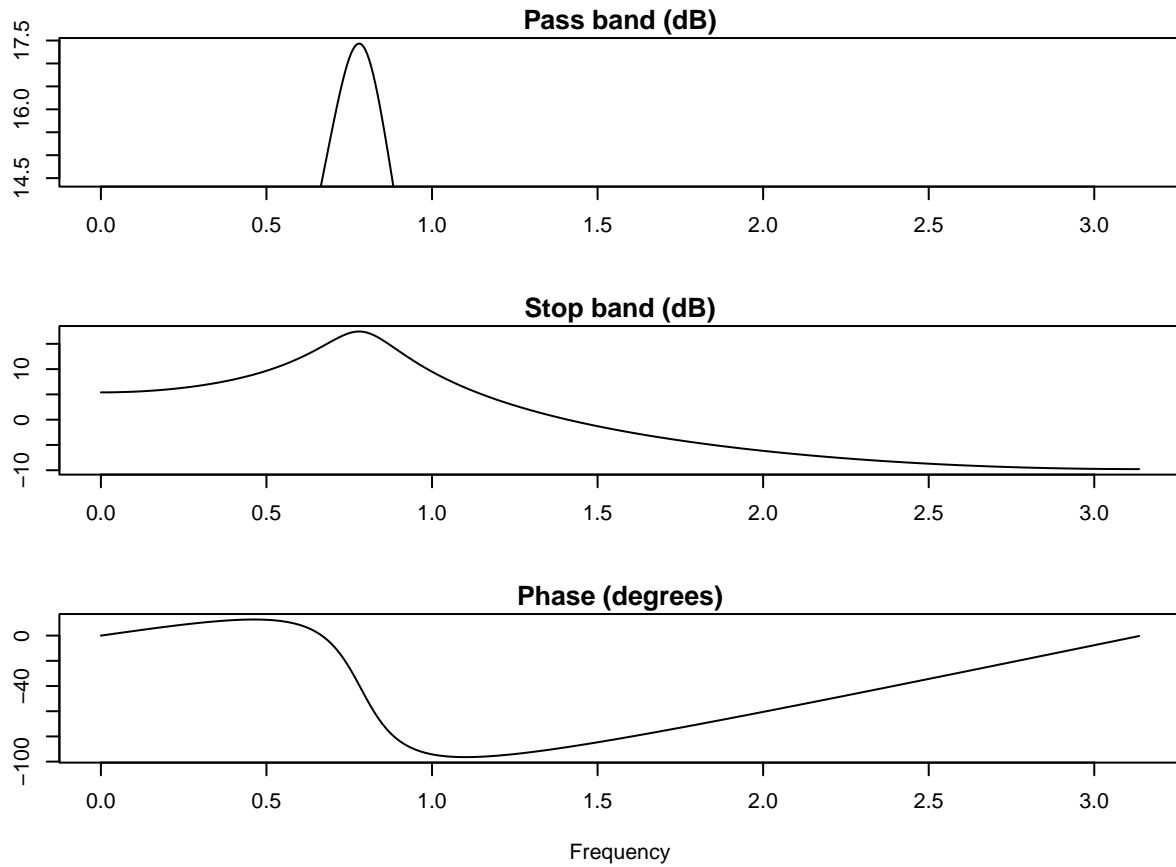


## \$impulso

## Impulse response



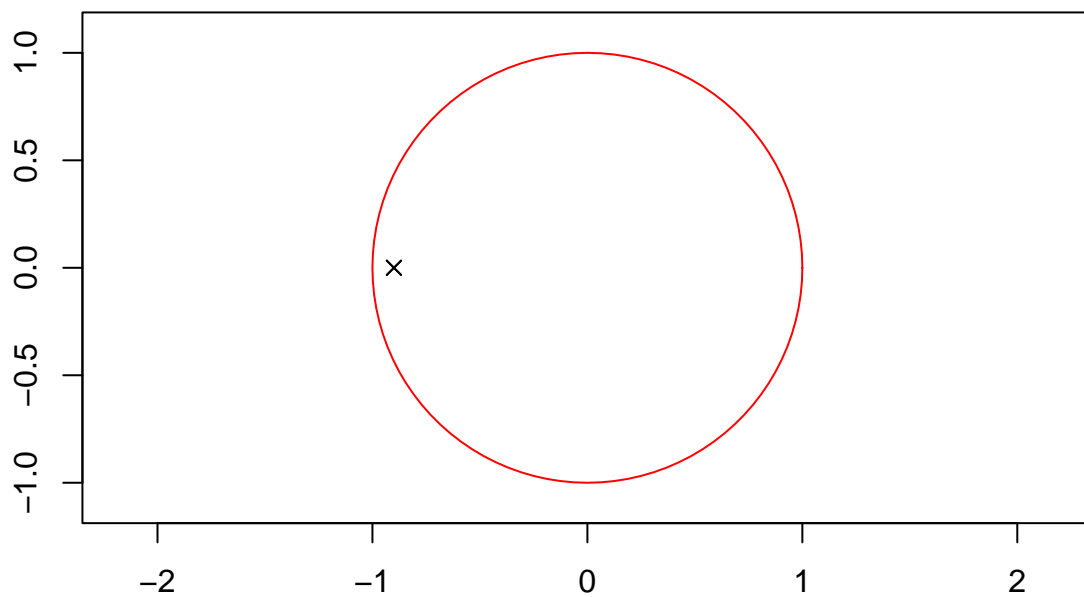
```
##  
## $frecuencia
```



```
##
## $zeros
## complex(0)
##
## $polos
## [1] 0.7856742+0.7856742i 0.7856742-0.7856742i
##
##
## =====
## Resultados para H2(z) con r = 0.9 y w0 = 3.141593
```

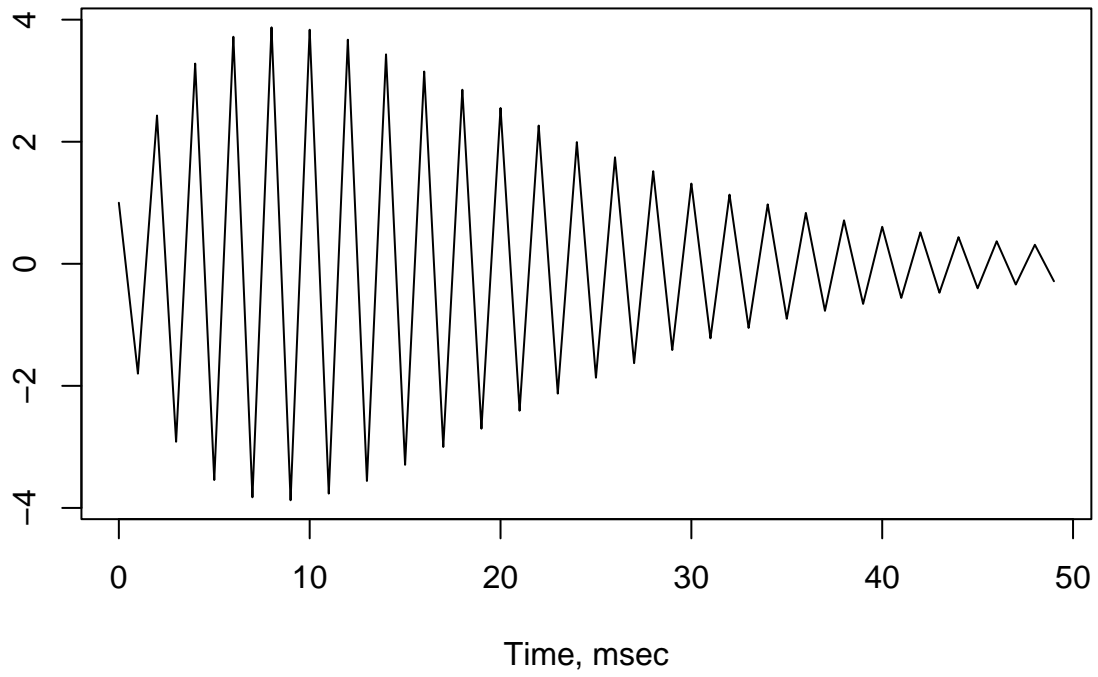


**Polos y ceros de  $H_2(z)$**   
 **$r = 0.9$  ,  $w_0 = 3.142$**

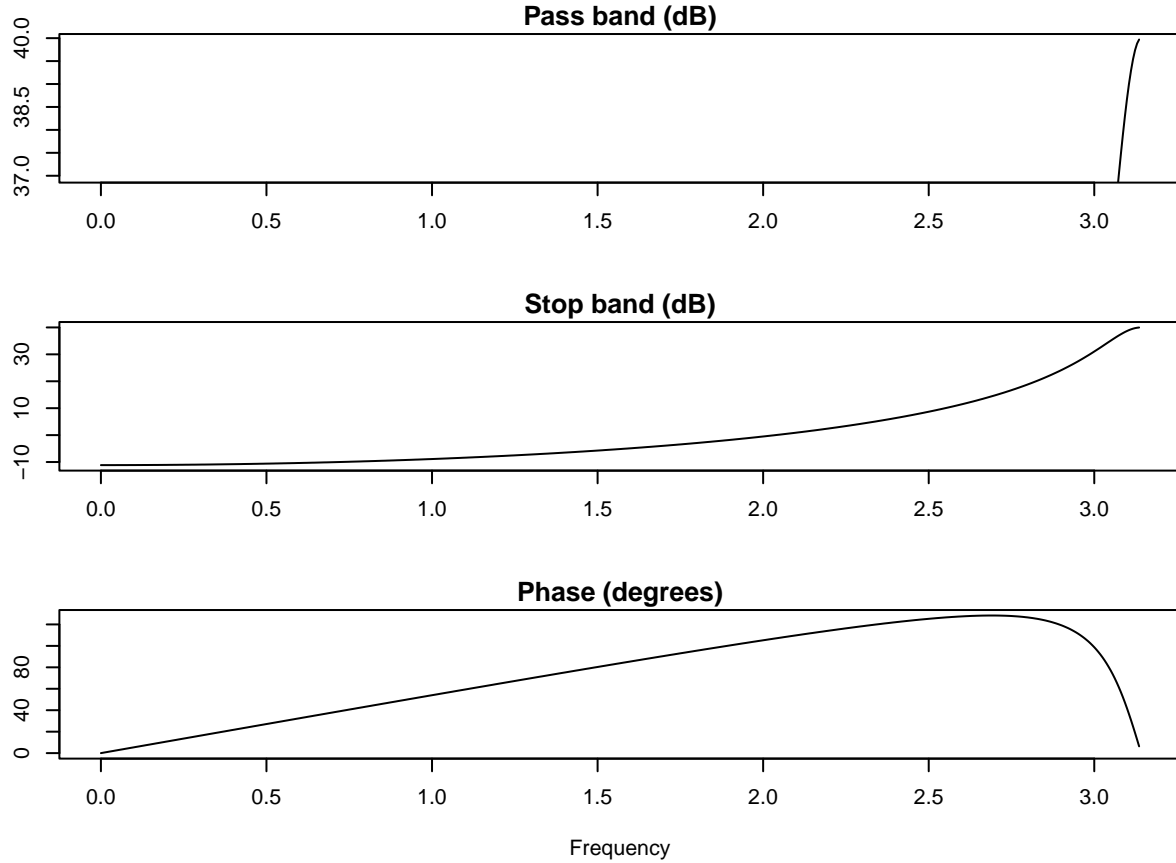


## \$impulso

## Impulse response



```
##  
## $frecuencia
```



```
##
## $ceros
## complex(0)
##
## $polos
## [1] -1.111111-1.665335e-16i -1.111111+1.665335e-16i
```

El sistema  $H_2(z)$  se define como el inverso de  $H_1(z)$ :

$$H_2(z) = \frac{1}{H_1(z)} = \frac{1}{1 - 2r \cos(\omega_0)z^{-1} + r^2 z^{-2}}.$$

En la práctica, esto implica que el numerador se mantiene constante y el denominador es un polinomio de segundo grado. En el plano- $z$  podemos ver que los polos de  $H_2(z)$  coinciden exactamente con los ceros de  $H_1(z)$  y, como era de esperar, el sistema no tiene ceros propios.

#### Efecto de $r$ (con $\omega_0 = 0$ ).

Al cambiar el valor de  $r$ , los polos se alejan o se acercan al origen, lo que modifica la forma de la respuesta en frecuencia. Cuando  $r$  está cerca de 1, los polos se acercan al círculo unidad y aparece un pico muy marcado y estrecho en la respuesta. Para valores más pequeños, el pico es más suave y ancho. Si  $r$  supera 1, los polos quedan fuera del círculo unidad y el sistema deja de ser estable.

#### Efecto de $\omega_0$ (con $r = 0.9$ ).

Manteniendo fijo  $r$  y variando  $\omega_0$ , los polos giran alrededor del origen manteniendo prácticamente el mismo radio. Esto desplaza la frecuencia donde aparece el pico en la respuesta. Así, para  $\omega_0 = 0$  el máximo está en baja frecuencia, para  $\omega_0 = \pi/4$  aparece en una frecuencia intermedia y para  $\omega_0 = \pi$  se sitúa cerca de la frecuencia de Nyquist.

### 2.1.2. Comparación entre $H_1(z)$ y $H_2(z)$

**3. Analiza las respuestas en frecuencia de  $H_1(z)$  y  $H_2(z)$  para  $w_0 = \pi/4$  y  $r = 0.9$ . ¿Qué tipo de filtros podemos diseñar que sólo tengan ceros como  $H_1(z)$  o que sólo tengan polos como  $H_2(z)$ ?**

Con  $\omega_0 = \pi/4$  y  $r = 0.9$  observamos que la respuesta en frecuencia de  $H_1(z)$  (sólo ceros) baja mucho en torno a  $\omega_0$ . Es decir, el filtro atenuará las componentes en esta frecuencia y deja pasar mejor el resto, por lo que se comporta como un filtro de rechazo de banda (notch). Los filtros que sólo usan ceros, como  $H_1(z)$ , suelen ser FIR y permiten hacer filtros notch o filtros peine. Además, siempre son estables y la fase suele ser más suave.

Por otro lado,  $H_2(z)$  sólo tiene polos en las posiciones donde  $H_1(z)$  tenía ceros, y su respuesta en frecuencia muestra un pico muy marcado en  $\omega_0 = \pi/4$ . Esto hace que el sistema funcione como un filtro resonante o pasa-banda estrecho, amplificando las componentes cercanas a esa frecuencia. Los filtros que sólo tienen polos, como  $H_2(z)$ , sirven para hacer filtros IIR muy selectivos (tipo pasa-banda o pasa-bajo), aunque la fase se distorsiona más y la estabilidad depende de que los polos estén dentro de la circunferencia unidad.

Sistema	Polos	Ceros	Tipo	Estabilidad
$H_1(z)$	Ninguno finito	Dependen de $r$ y $\omega_0$	FIR	Siempre estable
$H_2(z) = 1/H_1(z)$	Polos en ceros de $H_1(z)$	No tiene ceros	IIR	Estable si $ r  < 1$

Cuadro 1: Resumen comparativo entre  $H_1(z)$  (solo ceros) y  $H_2(z)$  (solo polos).

### 2.1.3. Filtrado de una señal de prueba

**4. Genera una señal que sea la suma de una componente continua de valor 3 y una senoide de frecuencia  $f = 1/8\text{Hz}$ . Filtra esta señal con el filtro  $H_1(z)$  del apartado anterior y representa la transformada de Fourier de la señal original y la señal filtrada. ¿Qué observas?**

```
# Señal y filtrado con H1(z)

# Definimos la señal
analiza_senal_H1 <- function(r, w0) {
  N <- 150      # Número de muestras
  n <- 0:(N - 1)

  f <- 1/8 # Frecuencia de la senoide

  # Señal original
  x <- 3 + sin(2 * pi * f * n)

  # Coeficientes del filtro H1(z)
  b <- c(1, -2 * r * cos(w0), r^2) # numerador (ceros)
  a <- 1                          # denominador (FIR)

  # Señal filtrada
  y <- filter(b, a, x)

  # Transformada de Fourier
  X <- fft(x)
  Y <- fft(y)

  k <- 0:(N/2)
  omega <- 2 * pi * k / N
```

```

par(mfrow = c(2, 2))

# Señales en el tiempo
plot(n, x, type = "l",
     main = "Señal original x[n]",
     xlab = "n", ylab = "Amplitud")

plot(n, y, type = "l",
     main = "Señal filtrada y[n]",
     xlab = "n", ylab = "Amplitud")

# Espectro de la señal original
plot(omega, Mod(X[k + 1]), type = "l",
     main = "Espectro |X(w)|",
     xlab = "w (rad/muestra)", ylab = "|X(w)|")

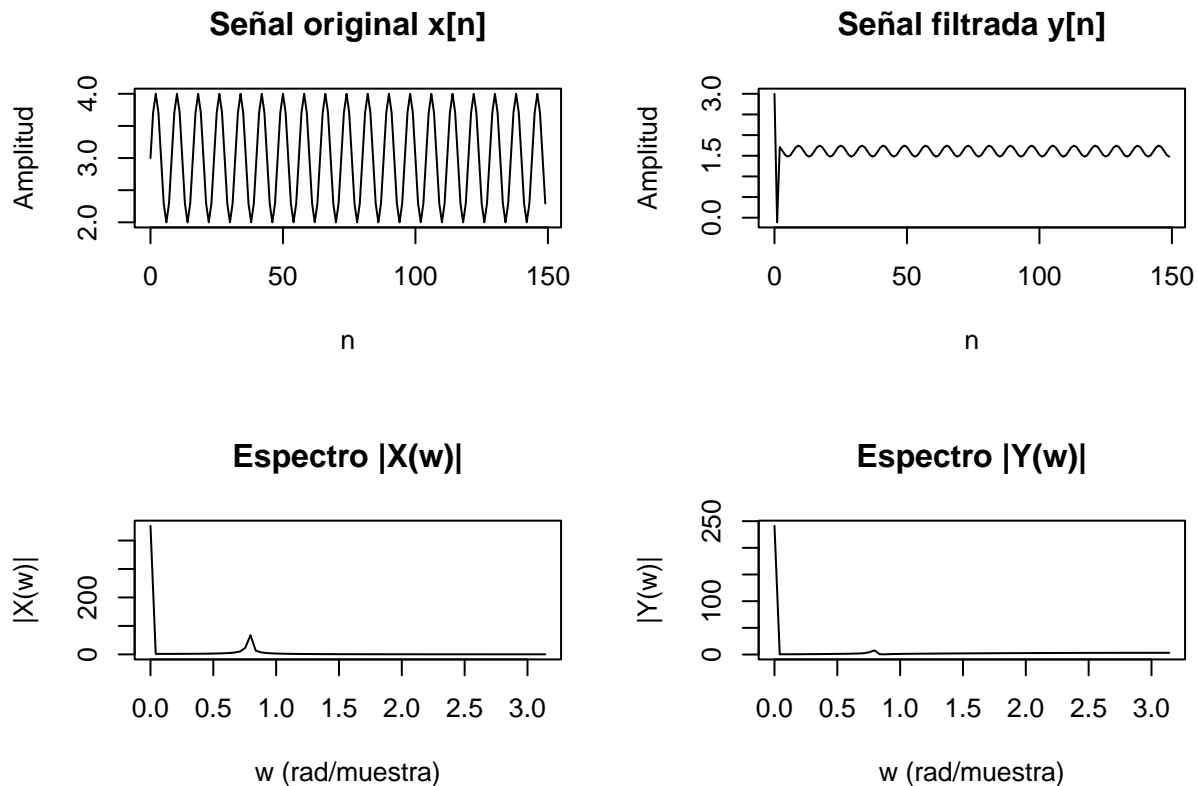
# Espectro de la señal filtrada
plot(omega, Mod(Y[k + 1]), type = "l",
     main = "Espectro |Y(w)|",
     xlab = "w (rad/muestra)", ylab = "|Y(w)|")

(list(x = x, y = y, X = X, Y = Y))
}

# Parámetros del apartado anterior
r <- 0.9
w0 <- pi / 4

resultado_ap4 <- analiza_senal_H1(r, w0)

```



Se observa que la señal original tiene una componente continua y una senoide en  $\omega = \pi/4$ . Después del filtrado con  $H_1(z)$ , la componente continua se mantiene mientras que la senoide prácticamente desaparece. El filtro actúa como un rechazo de banda centrado en  $\pi/4$ , eliminando esa frecuencia y dejando pasar el resto.

## 2.2. Filtro peine

Un sistema digital muy utilizado como filtro paso-bajo es lo que conocemos como promediador. La ecuación en diferencias de este filtro es:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k].$$

Determina la respuesta impulsional y la respuesta en frecuencia del sistema para diferentes  $N$  (2, 4, 9). Calcula los polos y ceros de  $H(z)$ , ¿qué tienen de especial? ¿qué ocurre en los ceros de  $H(z)$ ?

```
# Función para el promediador
funcion_prom <- function(N) {
  num <- rep(1/N, N)
  denom <- 1          # FIR

  # Respuesta impulsional
  res_impulso <- impz(num, denom)
  plot(res_impulso,
       main = paste("Respuesta al impulso h[n] para N =", N),
```

```

        xlab = "Muestras n",
        ylab = "h[n]")

# Respuesta en frecuencia
res_frecuencia <- freqz(num, denom)
plot(res_frecuencia,
      main = paste("Respuesta en frecuencia |H(f)| para N =", N))

# Polos y ceros
polos <- numeric(0)
ceros <- polyroot(num)
zplane(num, denom,
        main = paste("Diagrama de polos y ceros para N =", N))

cat("\nCeros del sistema:\n")
print(ceros) # Mostramos los ceros

# Resultados
return(list(impulso = res_impulso,
            frecuencia = res_frecuencia,
            polos = polos,
            ceros = ceros))
}

for (N in c(2, 4, 9)) {
  cat("\n=====n")
  cat("Resultados para el promediador con N =", N, "n")
  funcion_prom(N)
}

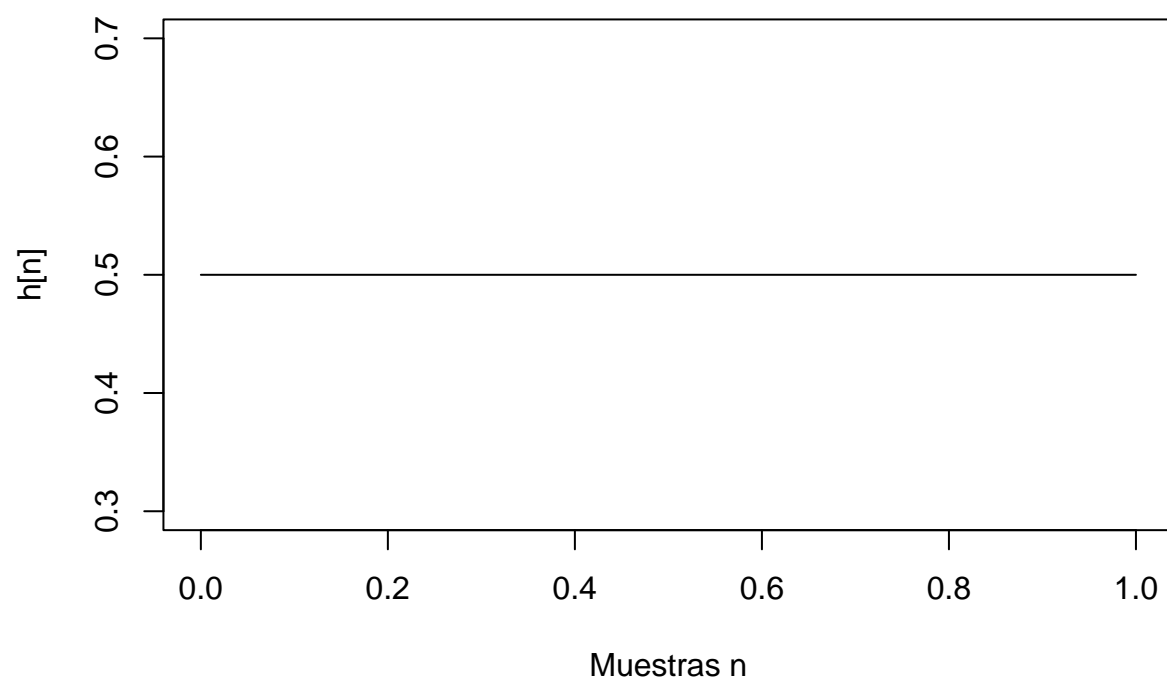
```

```

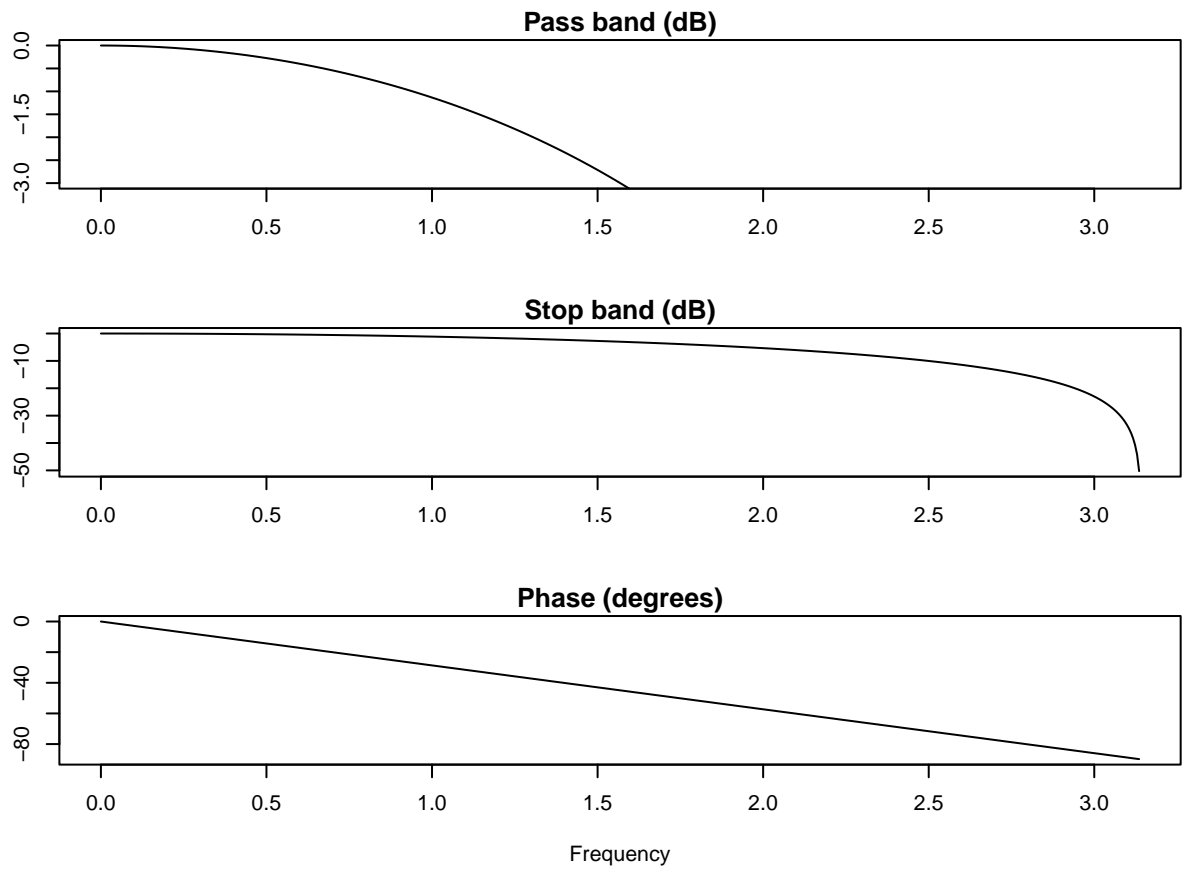
##
## =====
## Resultados para el promediador con N = 2

```

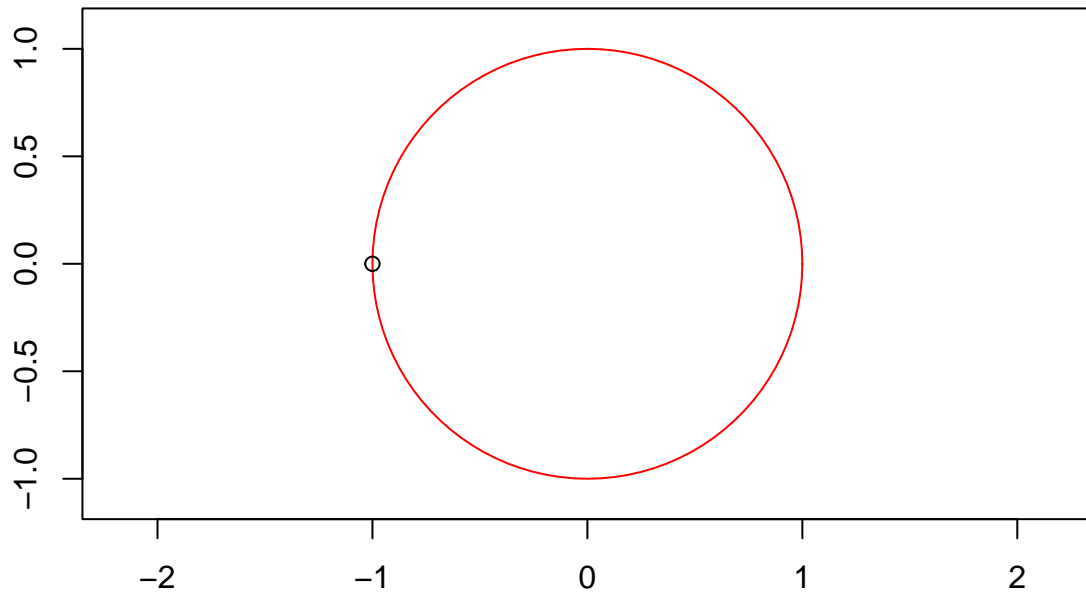
### Respuesta al impulso $h[n]$ para $N = 2$





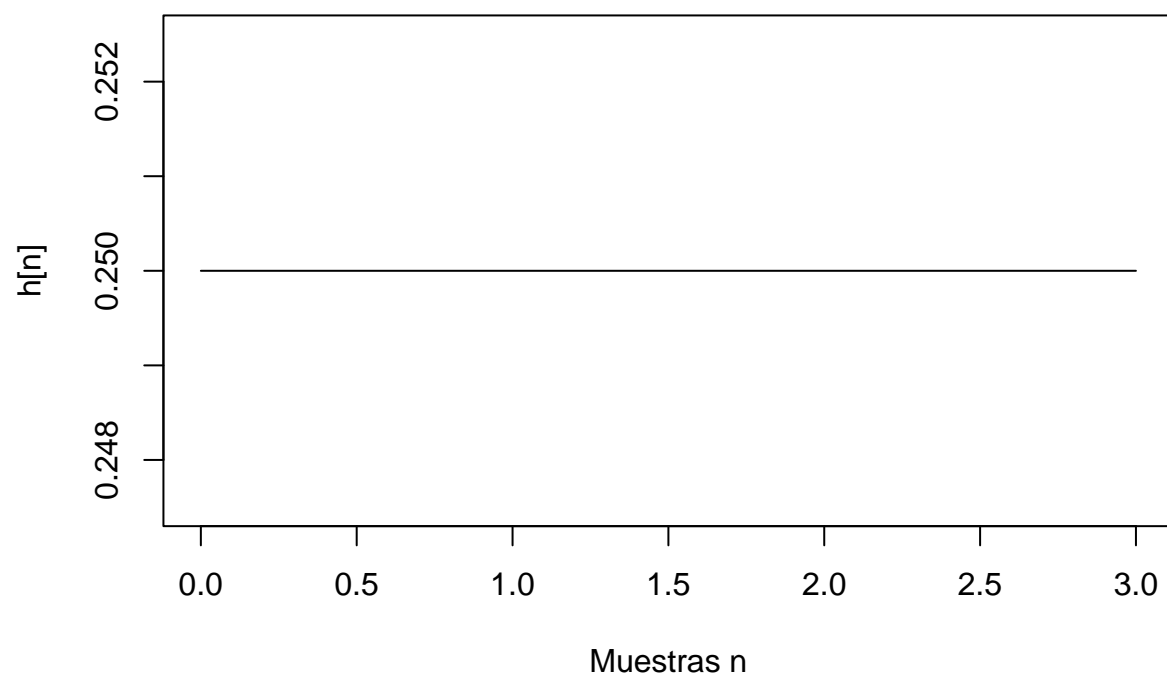


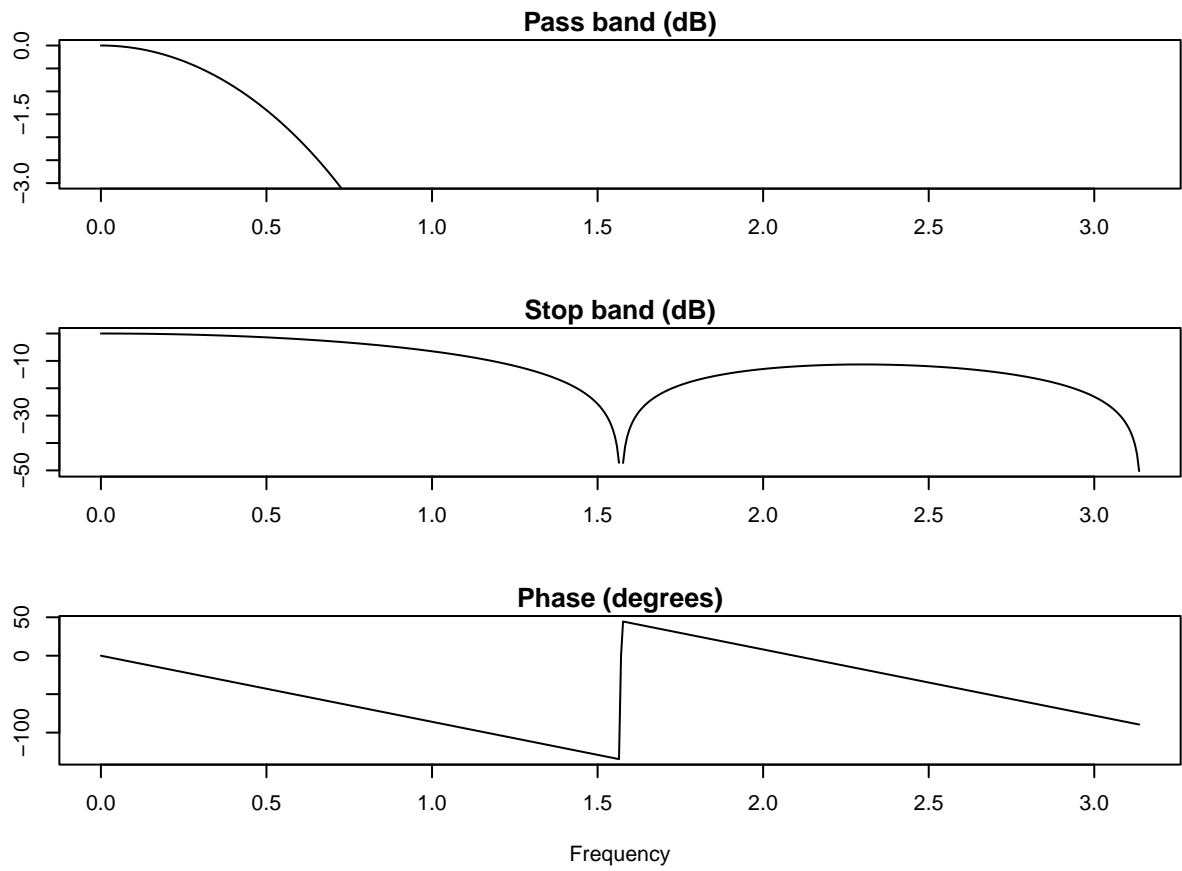
## Diagrama de polos y ceros para N = 2



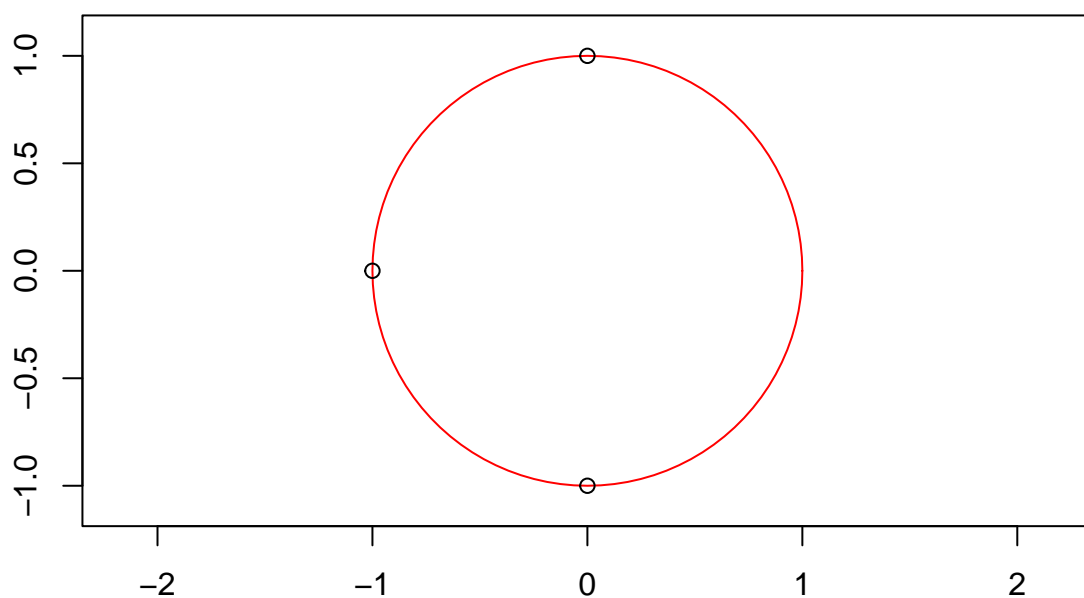
```
##  
## Ceros del sistema:  
## [1] -1+0i  
##  
## =====  
## Resultados para el promediador con N = 4
```

### Respuesta al impulso $h[n]$ para $N = 4$



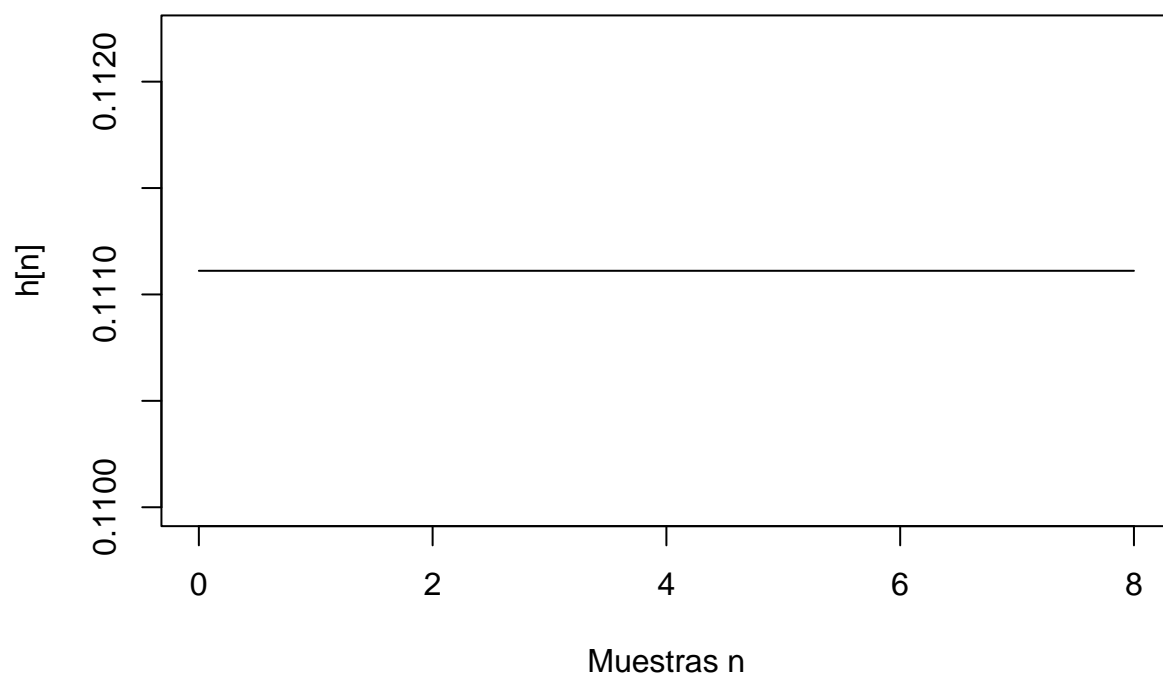


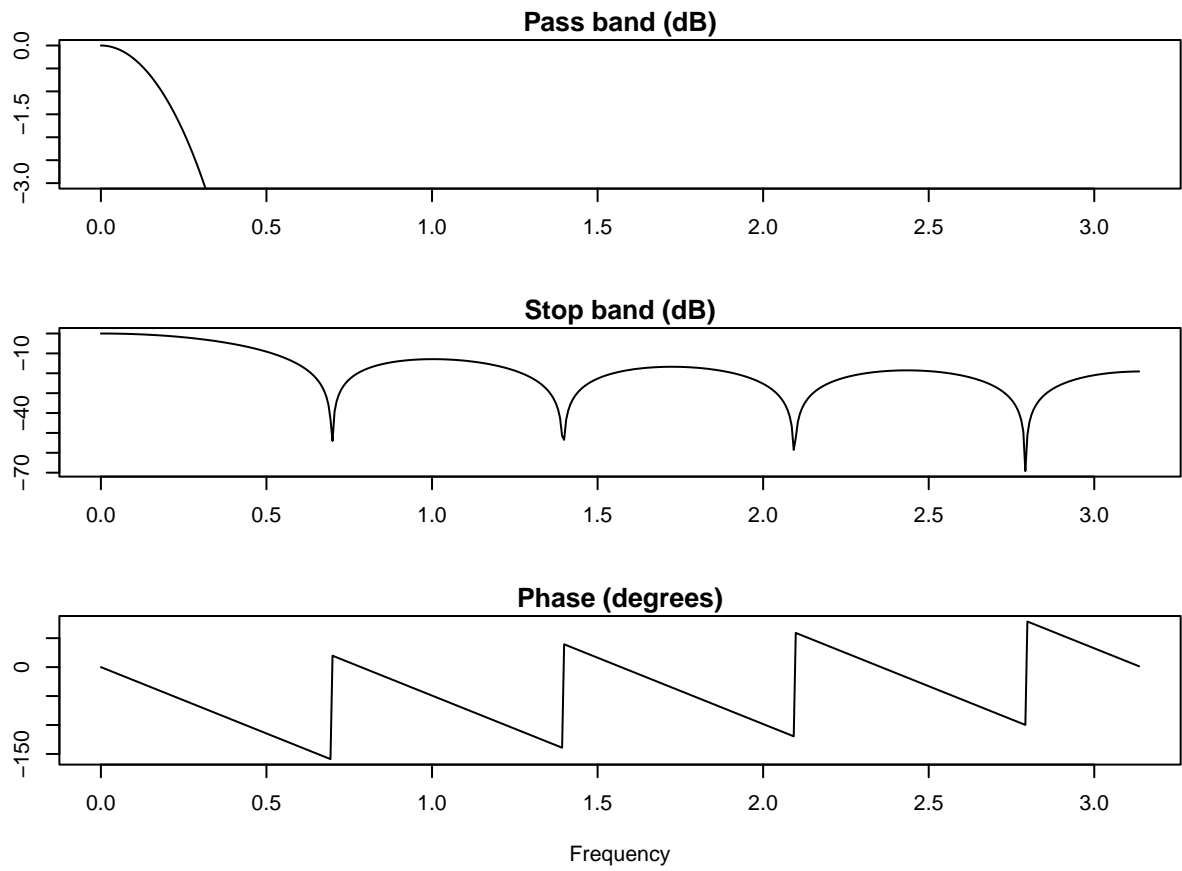
## Diagrama de polos y ceros para N = 4



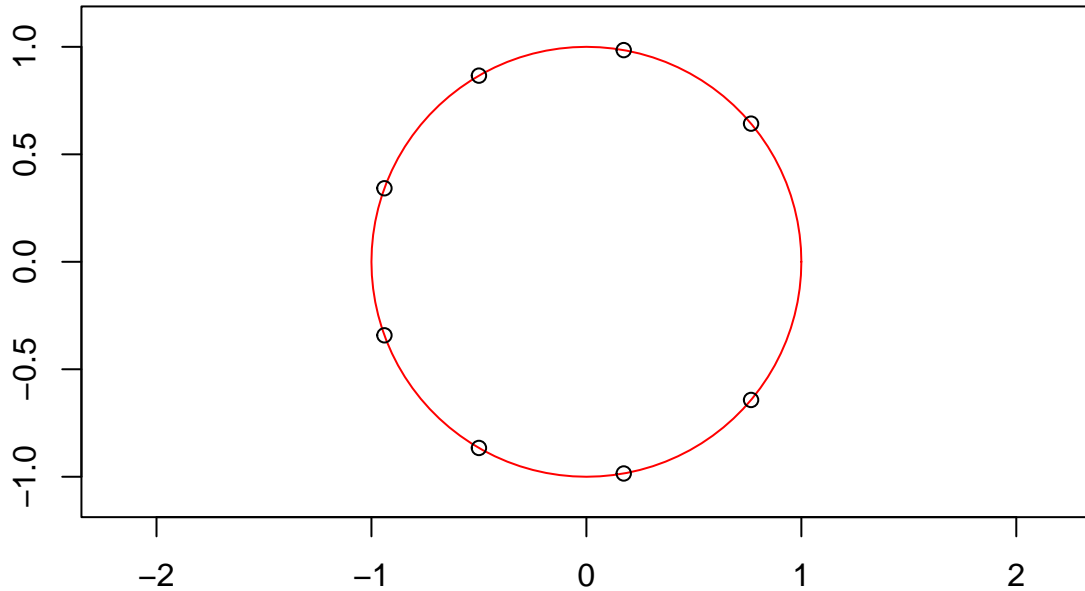
```
##
## Ceros del sistema:
## [1] -2.547508e-17+1.00000e+00i -1.000000e+00+1.27815e-17i
## [3]  0.000000e+00-1.00000e+00i
##
## =====
## Resultados para el promediador con N = 9
```

### Respuesta al impulso $h[n]$ para $N = 9$





## Diagrama de polos y ceros para N = 9



```
##
## Ceros del sistema:
## [1] 0.7660444+0.6427876i -0.9396926+0.3420201i -0.5000000-0.8660254i
## [4] 0.7660444-0.6427876i 0.1736482+0.9848078i -0.5000000+0.8660254i
## [7] 0.1736482-0.9848078i -0.9396926-0.3420201i
```

## Resultados obtenidos

- Para  $N = 2$ : aparece un cero en  $z = -1$ .
- Para  $N = 4$ : aparecen ceros en  $z = j$ ,  $z = -1$  y  $z = -j$ .
- Para  $N = 9$ : aparecen ocho ceros distribuidos uniformemente sobre el círculo unitario, correspondientes a las raíces novenas de la unidad excepto 1.

N	Ceros	Distribución	Consecuencia
2	1 cero ( $z = -1$ )	En el círculo unidad	Rechazo en $f_s/2$
4	3 ceros	Uniformes en el círculo unidad	Respuesta tipo peine
9	8 ceros	Raíces 9ª unidad excepto 1	Atenuación periódica de frecuencias

Cuadro 2: Propiedades del filtro promediador para diferentes valores de  $N$ .

Al ser un filtro FIR, si se expresa  $H(z)$  con potencias positivas de  $z$ , los polos quedan en el origen (en la forma estándar se toma el denominador igual a 1).



## Características especiales de los ceros

Lo que tienen de especial los ceros es que no aparecen en posiciones arbitrarias, sino que se distribuyen uniformemente sobre el círculo unitario. Esto se debe a que el numerador del filtro es un polinomio cuyas raíces son exactamente las raíces de la unidad (excepto 1).

## Comportamiento en los ceros de $H(z)$

En las frecuencias correspondientes a esos ceros, la respuesta en frecuencia se hace exactamente cero. Es decir, el filtro elimina completamente esas componentes de frecuencia. Como los ceros están igualmente espaciados, se eliminan frecuencias también igualmente espaciadas, generando la típica forma de “peine” en el espectro del filtro.

### 2.3. Eliminación del ruido de red

A continuación, implementaremos una aplicación práctica en el ámbito de la Ingeniería Biomédica, centrada en la eliminación del ruido de red en un electrocardiograma (ECG). Para ello, utilizaremos el fichero `ecg.mat`, que contiene un ECG contaminado por ruido de 50 Hz. Diseñaremos un sistema digital para eliminar dicha interferencia mediante un análisis de polos y ceros de la Transformada Z (el ECG ha sido adquirido con una frecuencia de muestreo de 1000 Hz). Además, implementaremos diferentes alternativas para eliminar este ruido.

En este ECG aparece otra interferencia importante: una variación de la línea basal que se manifiesta en bajas frecuencias. Para eliminarla, emplearemos el sistema definido por:

$$H(z) = \frac{1 - z^{-1}}{1 - \mu z^{-1}}$$

1. Comprobaremos su funcionamiento con la instrucción `filter`, considerando  $\mu = 0.9$ .
2. Determinaremos la respuesta en frecuencia del sistema y razonaremos los resultados obtenidos.
3. Analizaremos qué controla el parámetro  $\mu$ .

Finalmente, mediante el uso de la aplicación <http://www.micromodeler.com/dsp/>, visualizaremos los filtros que hemos utilizado en la práctica (ejercicio opcional).

#### Resolución:

Para quitar el ruido de la red eléctrica a 50 Hz, diseñamos un filtro *notch*. La idea del filtro es:

- Colocar dos ceros exactamente en la frecuencia que queremos eliminar:

$$z = e^{\pm j\omega_0}, \quad \omega_0 = 2\pi \frac{50}{1000}$$

- Colocar dos polos cerca de los ceros, multiplicándolos por un factor  $r < 1$  para que el filtro sea estable y solo elimine el ruido alrededor de 50 Hz.

Después de aplicar el filtro con `filter`, la interferencia de 50 Hz prácticamente desaparece y el ECG queda mucho más limpio.

La variación lenta de la línea basal se elimina con un filtro pasa-altos definido como:

$$H(z) = \frac{1 - z^{-1}}{1 - \mu z^{-1}}$$

Para esta práctica usamos  $\mu = 0.9$ .

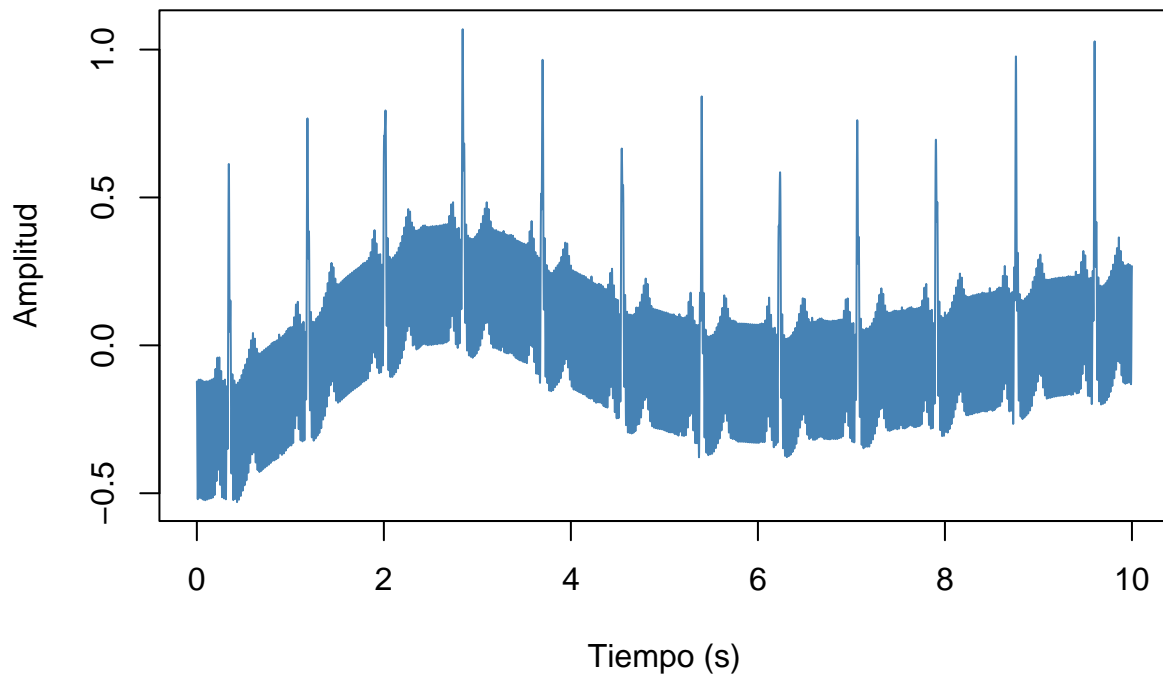
```
# Cargamos los datos y parámetros
datos <- readMat("ecg.mat")
ecg <- as.vector(datos$ecg)

fs <- 1000      # Frecuencia de muestreo (Hz)
ruido <- 50     # Frecuencia del ruido
mu <- 0.9       # Parámetro del filtro de la línea basal

N <- length(ecg)
t <- (0:(N - 1)) / fs # Eje temporal en segundos

# ECG original
plot(t, ecg, type="l", col="steelblue",
     main="ECG contaminado (ruido de red + línea basal)",
     xlab="Tiempo (s)", ylab="Amplitud")
```

### ECG contaminado (ruido de red + línea basal)



```
# 1) Filtro Notch para 50 Hz
omega <- 2 * pi * ruido / fs
```

```

# Ceros
zeros <- c(exp(1i * omega), exp(-1i * omega))

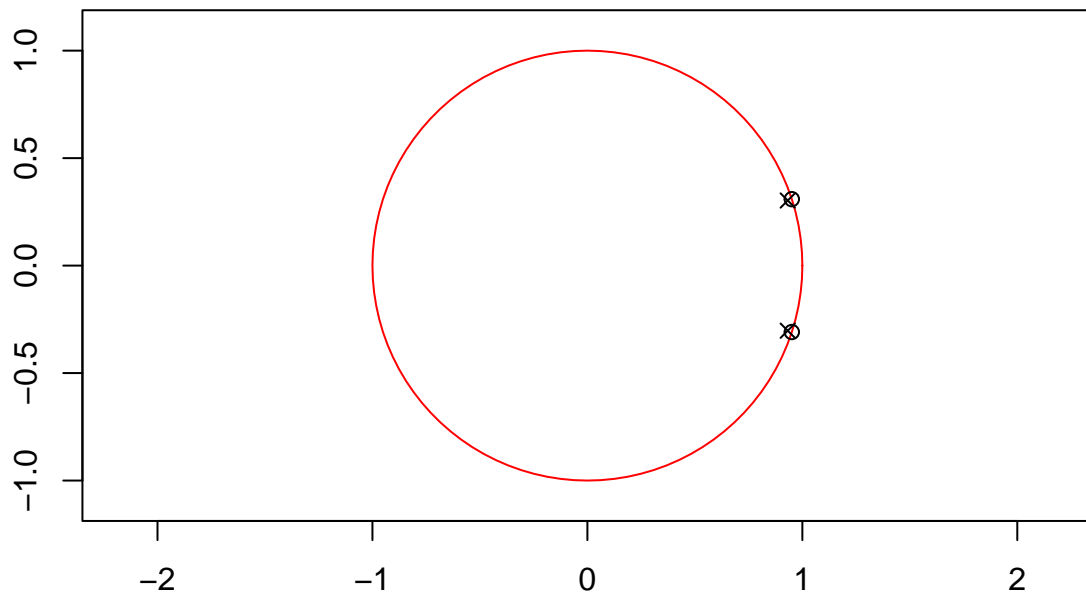
# Polos cerca de los ceros
r <- 0.98
poles <- r * zeros

# Coeficientes del filtro
b <- Re(poly(zeros)) # Numerador
a <- Re(poly(poles)) # Denominador

# Diagrama de polos y ceros del notch
zplane(b, a, main = "Filtro notch para 50 Hz: polos (x) y ceros (o)")

```

### Filtro notch para 50 Hz: polos (x) y ceros (o)

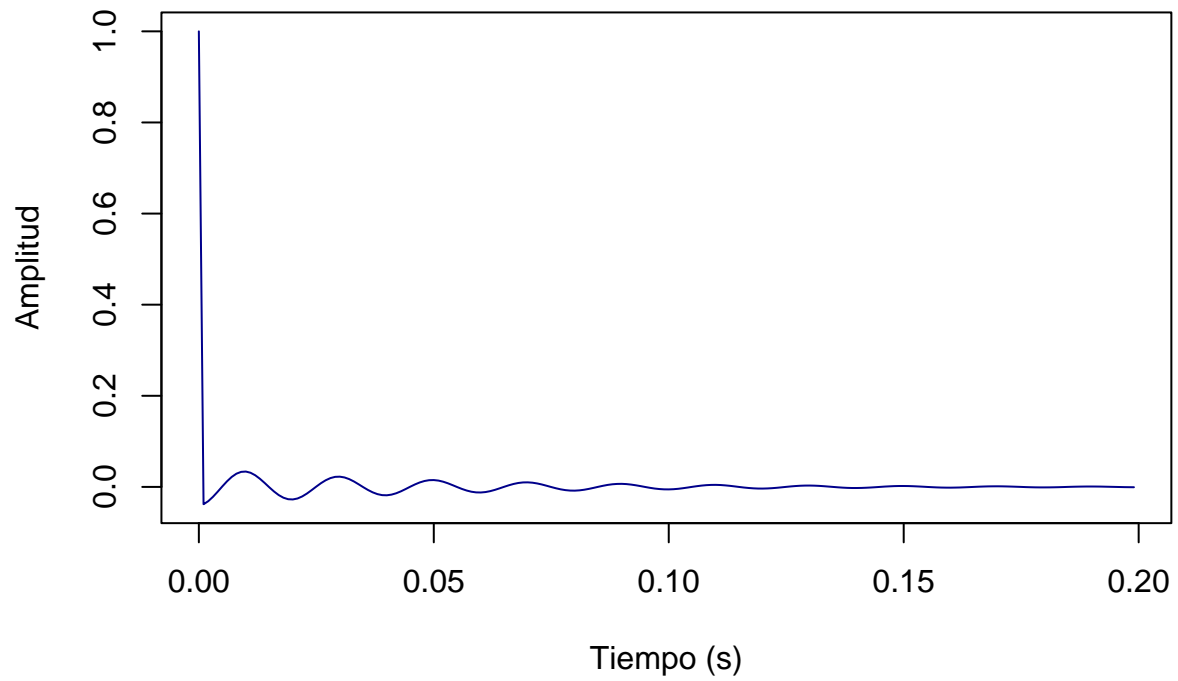


```

# Respuesta al impulso
h <- impz(b, a, n = 200, Fs = fs)
plot(h, type="l",
     col="darkblue",
     main="Respuesta al impulso del filtro notch (50 Hz)",
     xlab="Tiempo (s)", ylab="Amplitud")

```

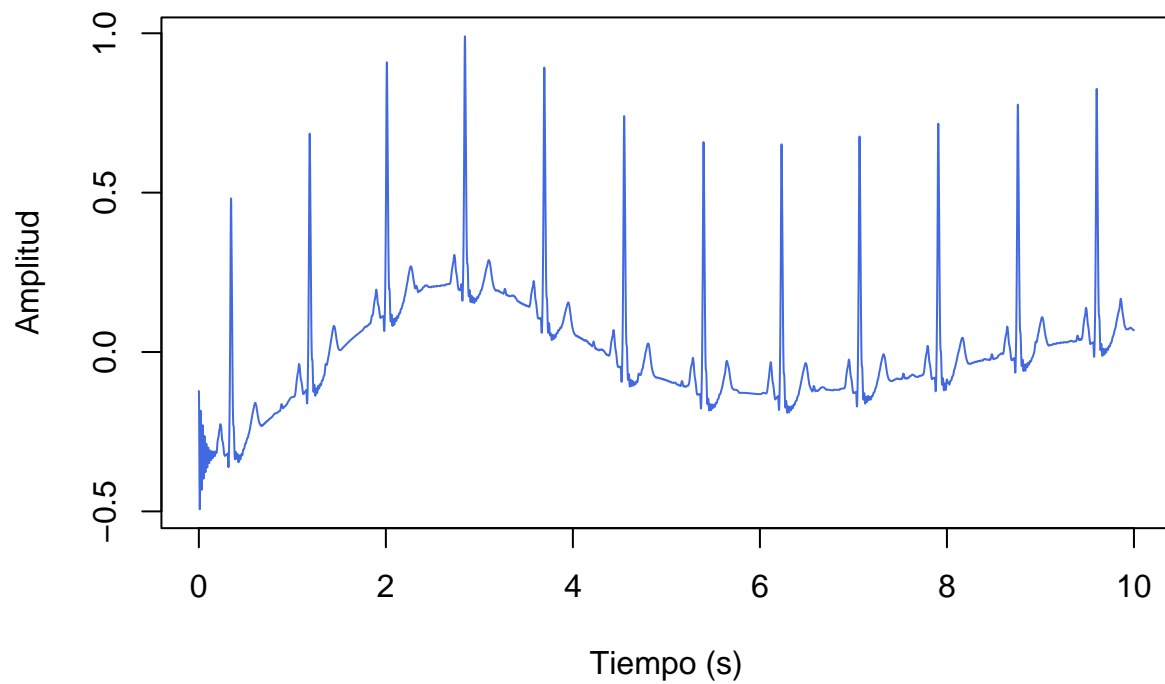
### Respuesta al impulso del filtro notch (50 Hz)



```
# Aplicamos el filtro al ECG
ecg_no50 <- filter(b, a, ecg)

plot(t, ecg_no50, type = "l", col = "royalblue",
     main = "ECG tras eliminar la interferencia de 50 Hz",
     xlab = "Tiempo (s)", ylab = "Amplitud")
```

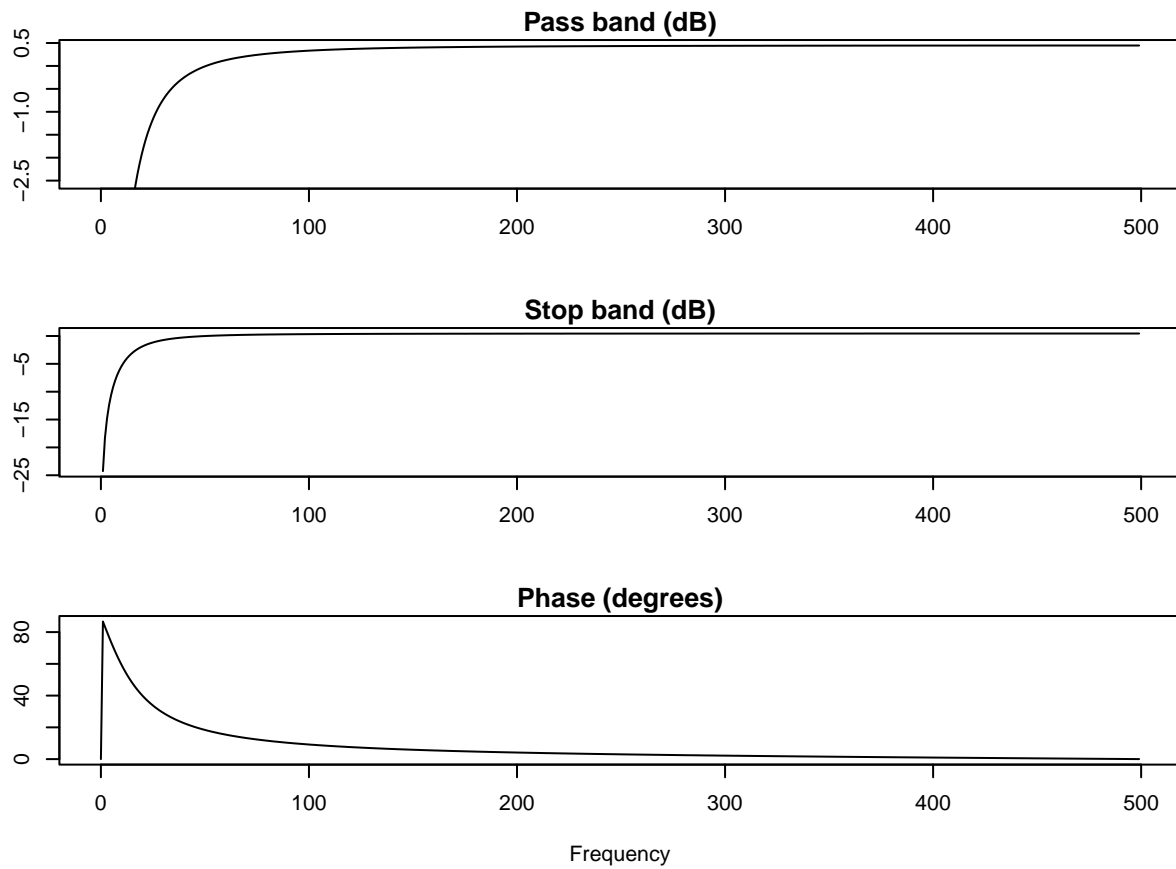
## ECG tras eliminar la interferencia de 50 Hz



```
# 2) Filtro pasa-altos para eliminar línea basal
#  $H(z) = (1 - z^{-1}) / (1 - \mu z^{-1})$ 

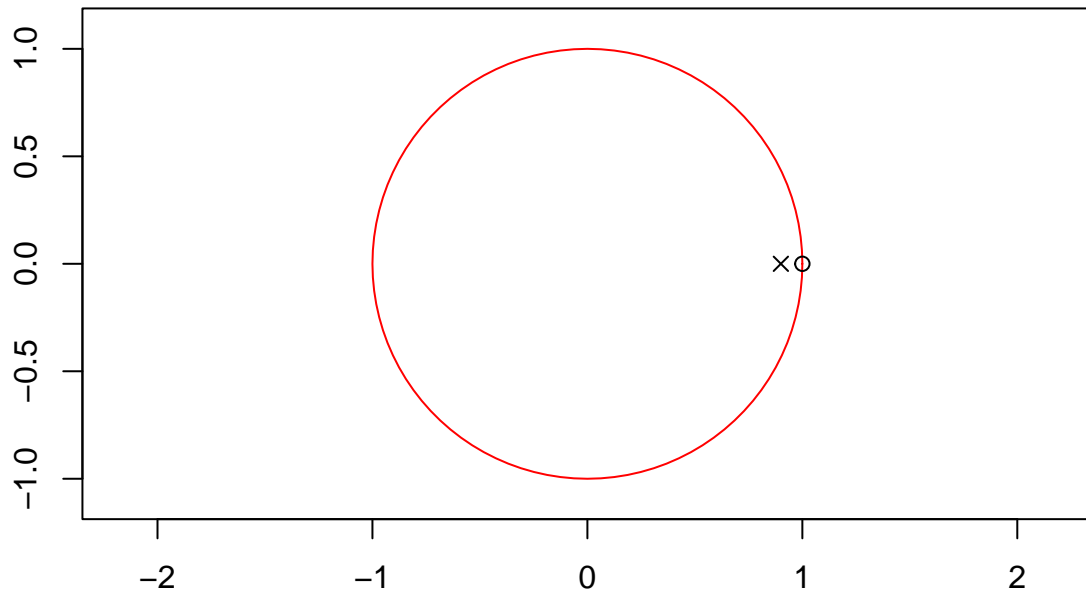
b2 <- c(1, -1) # Numerador
a2 <- c(1, -mu) # Denominador

# Respuesta en frecuencia
freqz(b2, a2, Fs = fs, main = "Respuesta en frecuencia del filtro de línea basal")
```



```
# Diagrama polos y ceros  
zplane(b2, a2, main = "Filtro de línea basal: polos (x) y ceros (o)")
```

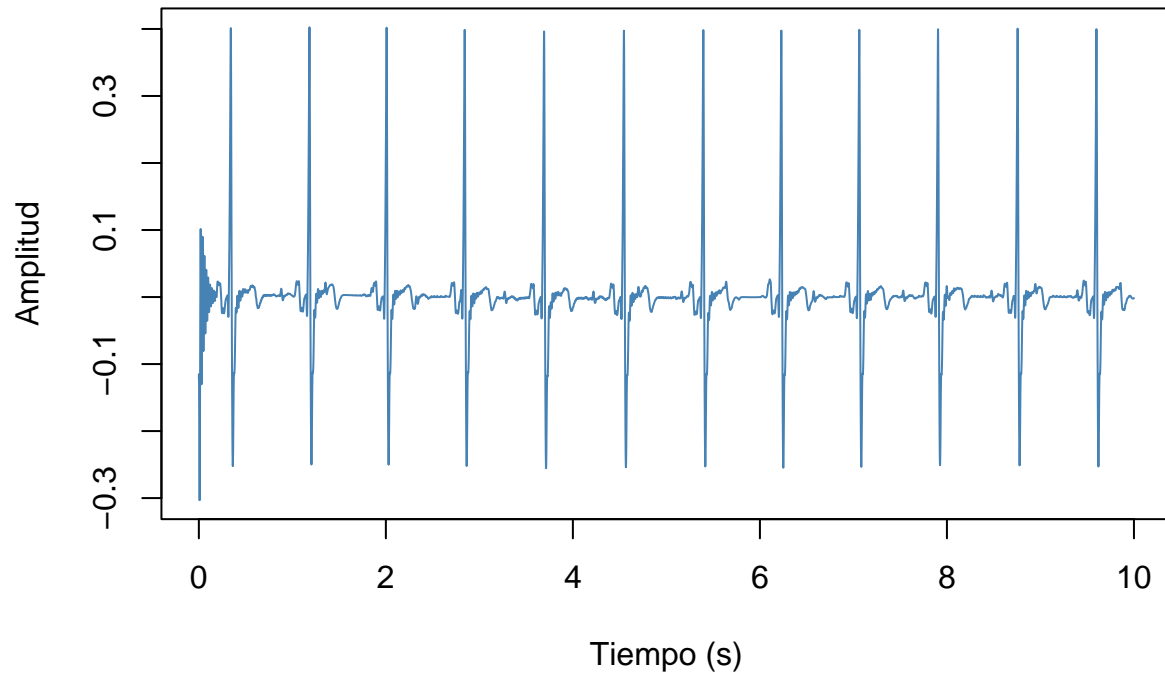
## Filtro de línea basal: polos (x) y ceros (o)



```
# Filtrado de la línea basal
ecg_final <- filter(b2, a2, ecg_no50)

plot(t, ecg_final, type = "l", col = "steelblue",
     main = "ECG final filtrado",
     xlab = "Tiempo (s)", ylab = "Amplitud")
```

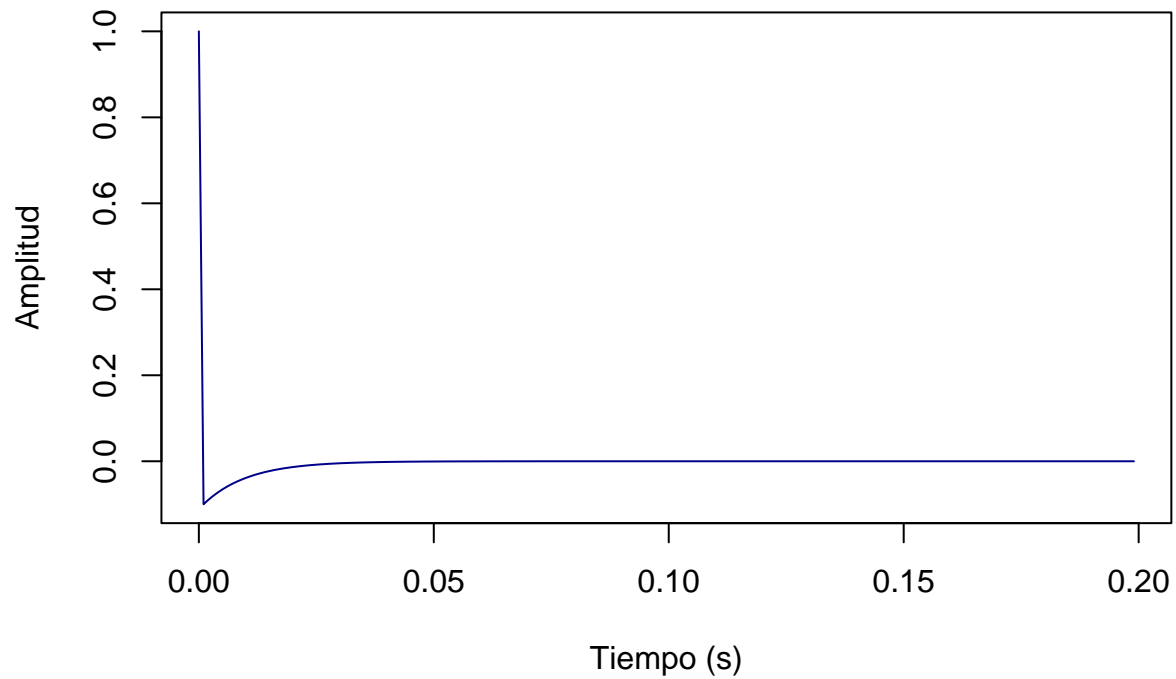
## ECG final filtrado



```
# Respuesta al impulso del filtro de línea basal
h_bas <- impz(b2, a2, n = 200, Fs = fs)
plot(h_bas, type = "l",
     col = "darkblue",
     main = "Respuesta al impulso del filtro pasa-altos (eliminación de línea basal)",
     xlab = "Tiempo (s)", ylab = "Amplitud")
```



## Respuesta al impulso del filtro pasa-altos (eliminación de línea base)

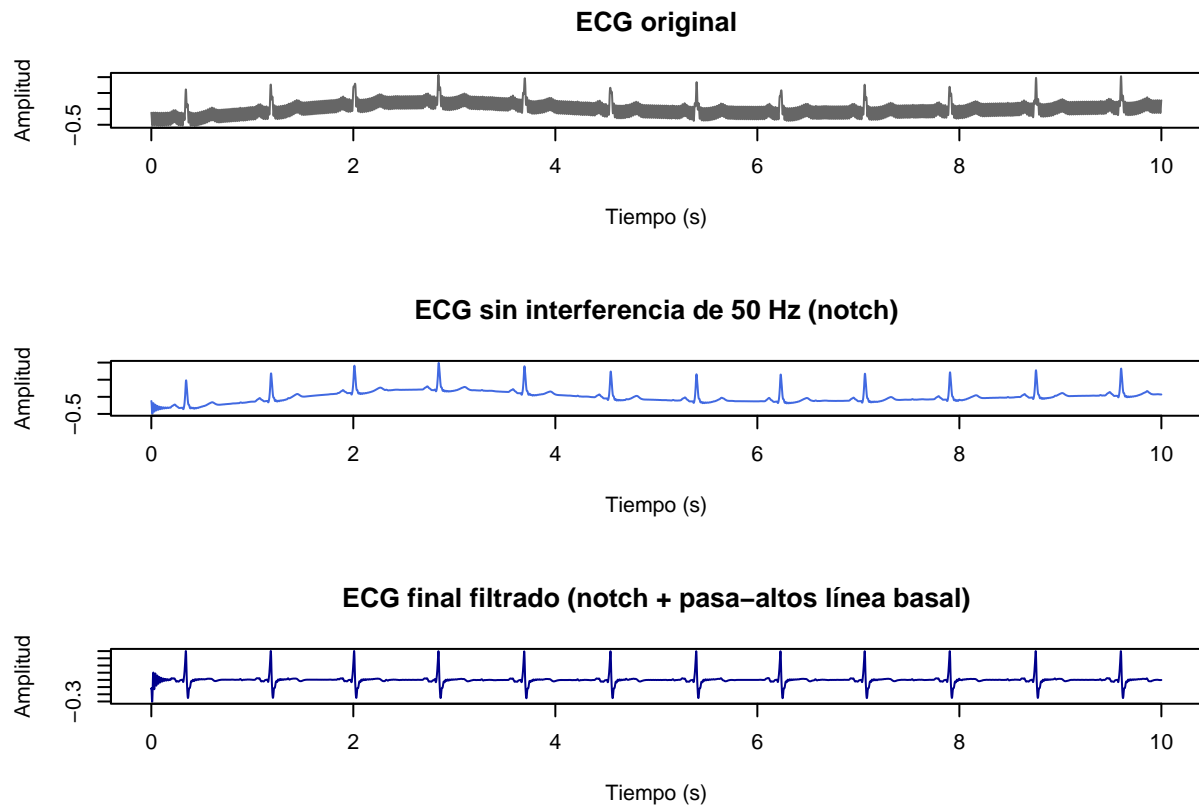


```
# Comparativa
par(mfrow = c(3, 1))

plot(t, ecg, type = "l", col = "grey40",
     main = "ECG original",
     xlab = "Tiempo (s)", ylab = "Amplitud")

plot(t, ecg_no50, type = "l", col = "royalblue",
     main = "ECG sin interferencia de 50 Hz (notch)",
     xlab = "Tiempo (s)", ylab = "Amplitud")

plot(t, ecg_final, type = "l", col = "darkblue",
     main = "ECG final filtrado (notch + pasa-altos línea basal)",
     xlab = "Tiempo (s)", ylab = "Amplitud")
```



```
par(mfrow = c(1, 1))
```

### Comprobación con filter

El filtro se implementa con los coeficientes:

$$b = [1, -1], \quad a = [1, -\mu]$$

Al aplicarlo sobre la señal filtrada del paso anterior, vemos que las variaciones lentas desaparecen y el ECG queda centrado en torno a cero.

### Respuesta en frecuencia

La respuesta en frecuencia muestra que:

- El filtro actúa como un **pasa-altos**, porque el cero en  $z = 1$  elimina las componentes de muy baja frecuencia, incluida la línea basal.
- El polo en  $z = \mu$  controla la pendiente y el efecto temporal del filtro.
- Para frecuencias medias y altas, la magnitud es casi constante, por lo que la forma de las ondas del ECG (P, QRS, T) no se ve afectada.

En resumen, atenúa solo las bajas frecuencias sin distorsionar la señal principal.

### ¿Qué controla el parámetro $\mu$ ?

El parámetro  $\mu$  define cómo de selectivo es el filtro:

- Más cercano a 1  $\rightarrow$  la zona de rechazo es más estrecha y la transición es suave.
- Más pequeño  $\rightarrow$  la transición es más brusca y se eliminan más bajas frecuencias.
- Además, si el polo está cerca del círculo unidad, el filtro “recuerda” más la señal (respuesta más lenta).

Con  $\mu = 0.9$ , conseguimos un buen equilibrio entre eliminar la línea basal y no deformar el ECG.