

Codificación Digital/Analógico

Julio César Ramirez-Pacheco

Junio 2020

Contents

Codificaciones analógicas	1
Ejemplos usando R	1
Tareas	2
	4

Codificaciones analógicas

Como se mencionó en la práctica anterior, la codificación puede ser de señales digitales a señales digitales, de digitales a analógicas, etc. En el caso de la codificación digital-analógico, nos podemos encontrar con las modulaciones:

- ASK (Amplitude Shift Keying).
- PSK (Phase Shift Keying).
- FSK (Frequency Shift keying)

Este tipo de modulaciones o codificaciones las encontramos en las redes de comunicaciones en los modems (moduladores/demoduladores) que funcionan como acceso a las redes analógicas, o bien para la transmisión inalámbrica de alta velocidad. En la siguiente práctica implementaremos éstas codificaciones usando el lenguaje de programación R.

Ejemplos usando R

Primero generaremos la señal de información:

```
# Información original 00110100010
# Representación gráfica
xt <- c(0,1,1,1,0,1,0,0,0,1,0)
bits <- rep(xt, each=100)
time <- seq(0,2, length=length(bits))
plot(time, bits, type = "l", main="Secuencia de datos de información", xlab="Tiempo", ylab="Valores", y
```

Posteriormente codificamos la anterior señal de comunicación usando las siguientes reglas:

- 0 se codifica con un nivel de amplitud $A_1 \cos(2\pi ft)$.
- 1 se codifica con un nivel de amplitud $A_2 \cos(2\pi ft)$.

Note que las amplitudes sirven para codificar la señal, la amplitud incluso puede ser 0. Ahora programaremos la anterior codificación usando R:

Secuencia de datos de información

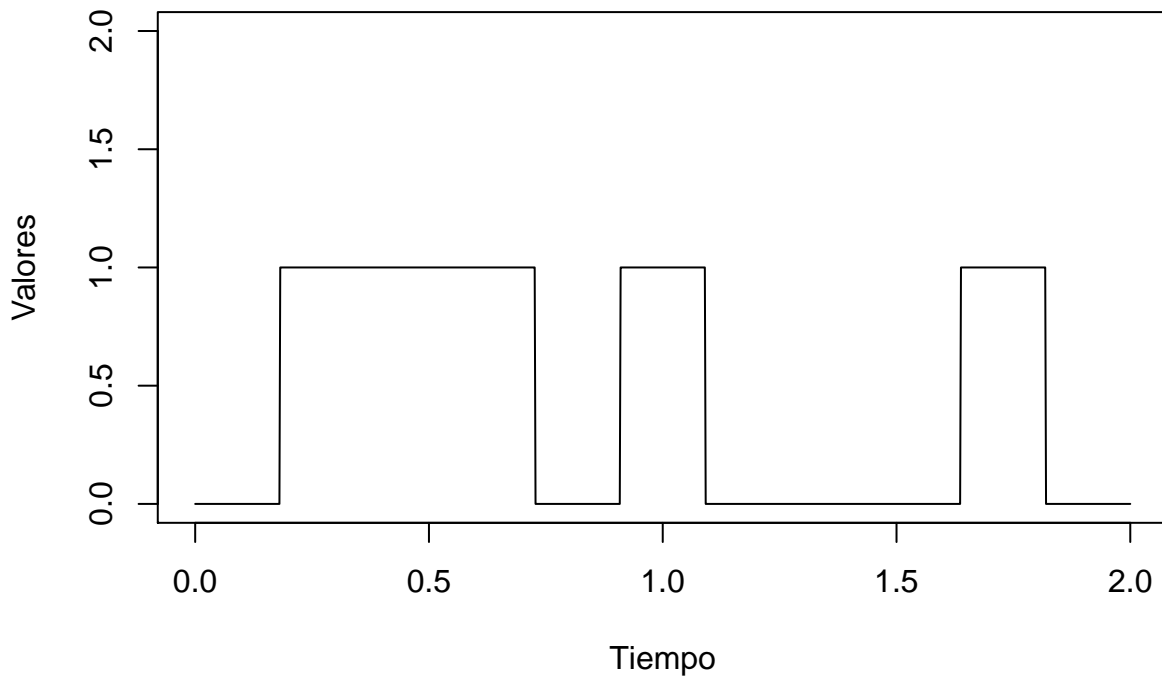


Figure 1: Bits de información. Estos bits serán codificados por una o más señales analógicas.

```
xt <- c(0,1,1,1,0,1,0,0,0,1,0)      # Se generan los códigos
# Ahora se codifica de acuerdo a la regla descrita arriba
bits <- rep(xt, each=100)
time <- seq(0,2, length=length(bits)) # Se genera el tiempo (puede ser cualquiera).
ask <- ifelse(bits==0, 0, cos(20*pi*time)) # Se genera la señal ASK
plot(time,bits+3,type="l",main="ASK e info", xlab="Tiempo", ylab="Valores", ylim=c(-2,5),col="blue")
lines(time, ask)
```

Es importante hacer notar que la regla indica que las amplitudes deben de ser iguales, por lo tanto la siguiente codificación también es válida:

```
xt <- c(0,1,1,1,0,1,0,0,0,1,0)      # Se generan los códigos
# Ahora se codifica de acuerdo a la regla descrita arriba
bits <- rep(xt, each=100)
time <- seq(0,2, length=length(bits)) # Se genera el tiempo (puede ser cualquiera).
ask <- ifelse(bits==0, 0.5*cos(20*pi*time), 1.5*cos(20*pi*time)) # Se genera la señal ASK
plot(time,bits+4,type="l",main="ASK e info", xlab="Tiempo", ylab="Valores", ylim=c(-2,6),col="blue")
lines(time, ask)
```

Tareas

Implementar las siguientes codificaciones de señales digitales a analógicas e investigar su aplicación en las redes de comunicaciones.

- FSK
- PSK

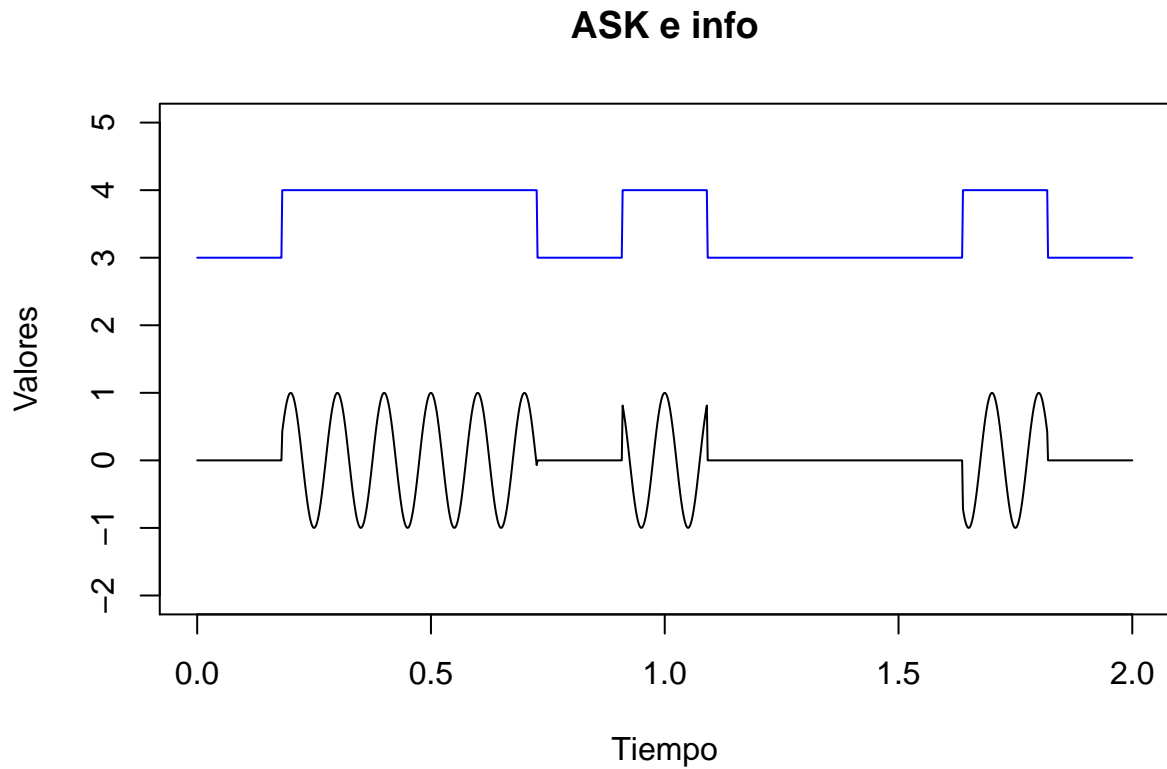


Figure 2: Señal ASK codificada según la regla de arriba y usando amplitudes 0 y A_1 , la señal ASK de este tipo se llama OOK.

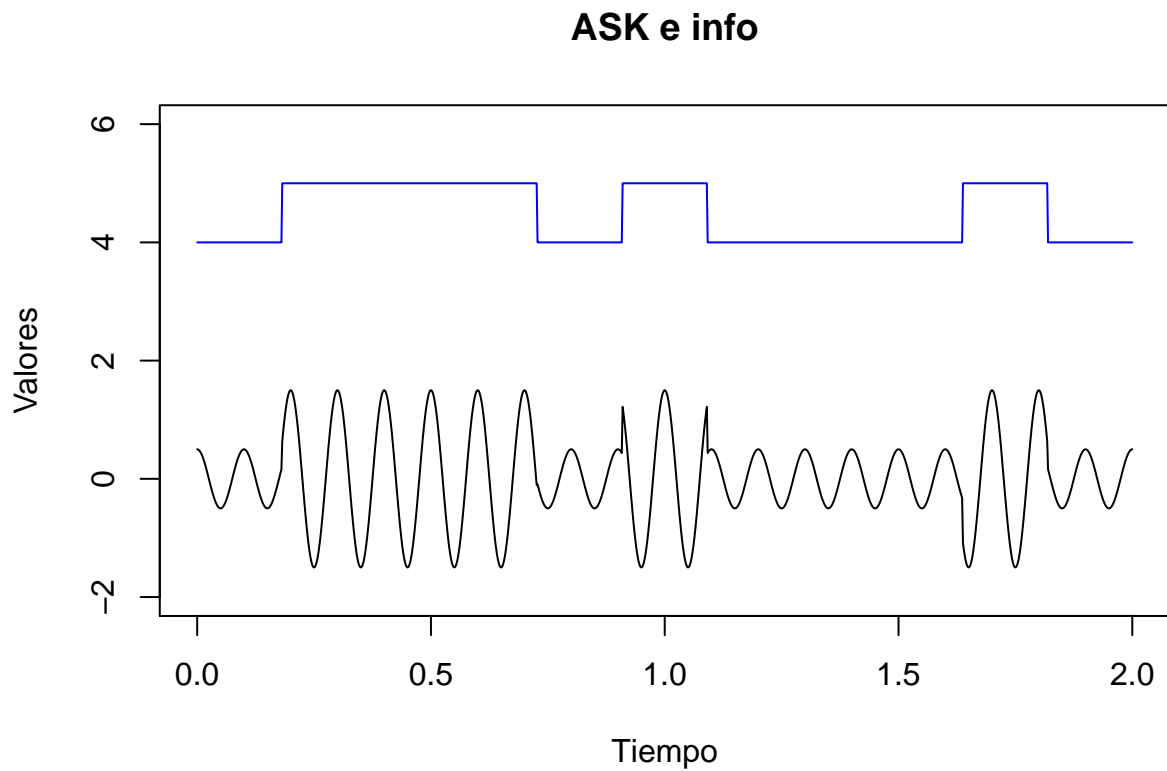


Figure 3: Señal ASK codificada usando dos señales senoidales con amplitudes A_1 y A_2 , donde $A_1 < A_2$.

