1. Simulaciones Básicas

Se simularon las funciones descritas en la Tabla (1).

Función
$A \cdot Cos(2\pi \cdot 1.5kHz \cdot t)$
Triangular simétrica 1.5kHz de pico V_{MAX}
3/2 Seno amplitud VMAX 1.5kHz

Tabla 1: Funciones simuladas.

Se determinó que el valor máximo de amplitud de entrada al sistema es de 5 V, el cual es el mínimo entre los dos valores limitantes: máxima entrada al CD4066 y límite mínimo de distorsión. Además, los límites de tensión de alimentación recomendados son 18 V de la hoja de datos del Sample and Hold.

Para hallar los valores óptimos de A, F_s y DT se simuló utilizando **LTSpiceXVII** las curvas de entrada y salida del sistema variando simultáneamente los valores de las tres variables, previamente fijando los rangos entre las mismas cambian. Estos rangos son de 1 V a 5 V para A, 21 kHz (para cumplir con el doble de la frecuencia de corte del filtro recuperador) a 25 kHz (límite del oscilador) y de 5 % (límite del oscilador) a 50 % (limite por consigna) para el duty cycle. Finalmente, se utilizó el siguiente script en **Python** para hallar el valor de los tres parámetros tal que la distorsión a la salida respecto a la entrada sea la mínima, computando la correlación entre las dos señales.

```
from PyLTSpice.LTSpice_RawRead import LTSpiceRawRead
from mpl_toolkits.mplot3d import Axes3D
import scipy.signal
import matplotlib.pyplot as plt
import numpy as np

LTR = LTSpiceRawRead("triangsh.raw")

corr = []
corr_maxes = []
step_vars = []
least_distorted_steps = []
time = LTR.get_trace(0)
vin = LTR.get_trace("V(vin)")
vout = LTR.get_trace("V(vout)")

for i in LTR.get_steps():
    corr.append(scipy.signal.correlate(vin.get_wave(i), vout.get_wave(i)))
    step_vars.append(LTR.steps[i])
    corr_maxes.append(np.max(corr[i]))

least_distorted_steps.append(np.where(corr_maxes == np.max(corr_maxes)))
plt.show()
print("Least distorted: ")

for i in range(len(least_distorted_steps)):
    print(LTR.steps[(least_distorted_steps[i][0][0])])

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
x = [(i['freqs']/1000) for i in step_vars]
y = [i['dts'] for i in step_vars]
z = [i['amp'] for i in step_vars]
c = corr_maxes
img = ax.scatter(x, y, z, c=c, cmap='Spectral', alpha=1)
cbar = plt.colorbar(img)
cbar.set_label('Correlacion')
plt.title("Maxima correlacion entre entrada y salida")
ax.set_zlabel('Amplitud [V]', rotation = 0)
ax.set_ylabel('Duty Cycle [%]', rotation = 0)
ax.set_xlabel('Frecuencia [kHz]', rotation = 0)

plt.show()

pow_in = []
pow_out = []
power_restored = []

for i in LTR.get_steps():
    pow_in.append(0)
    pow_out.append(0)
    for j in range(len(time.get_wave(i))):
        pow_in[i] =  pow_in[i] + abs(vin.get_wave(i)[j])**2
        pow_out[i] = pow_out[i] + abs(vout.get_wave(i)[j])**2

for i in LTR.get_steps():
    power_restored.append(pow_out[i]/pow_in[i])

print(round(power_restored[least_distorted_steps[0][0][0]], 4))

plt.plot(power_restored)
plt.plot(least_distorted_steps[0][0][0], power_restored[least_distorted_steps[0][0][0]], "ro")
plt.show()
```

Se obtuvieron los siguientes resultados:

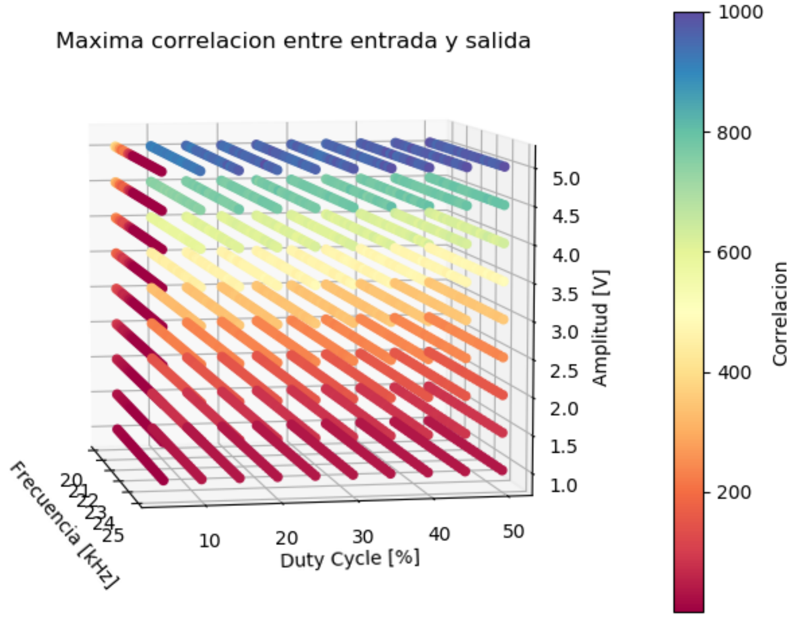


Figura 1: Scatterplot de las simulación para **senoidal con S&H**.

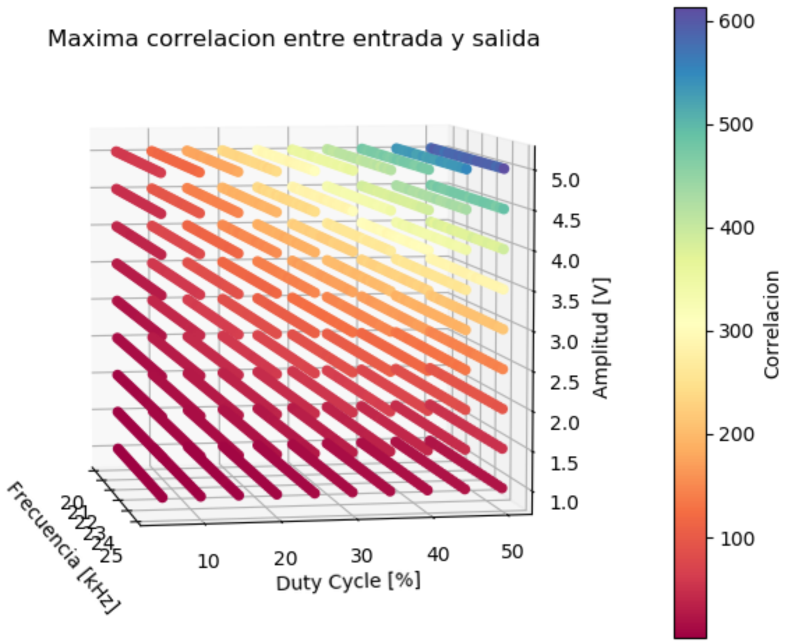


Figura 2: Scatterplot de las simulación para la **senoidal con llave analógica**.

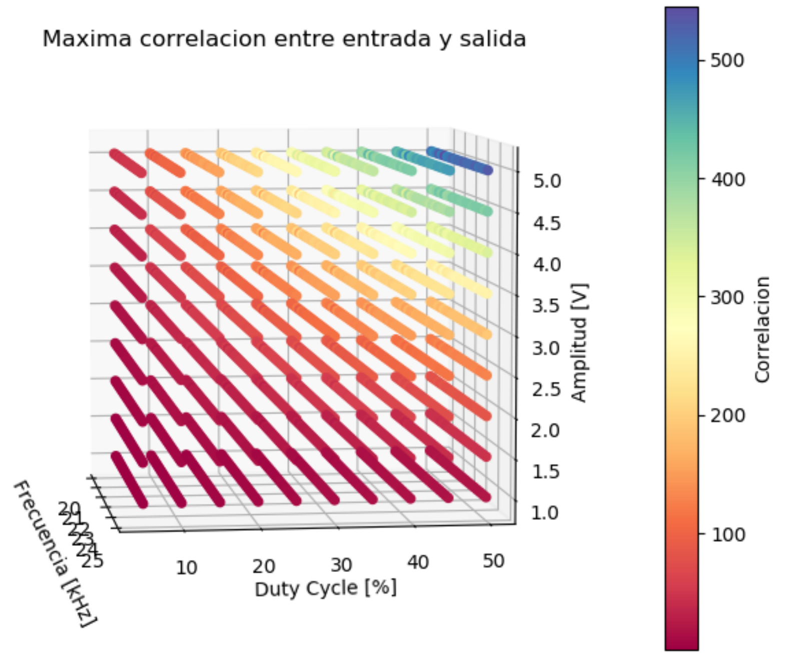


Figura 3: Scatterplot de las simulación para la **triangular con llave analógica**.

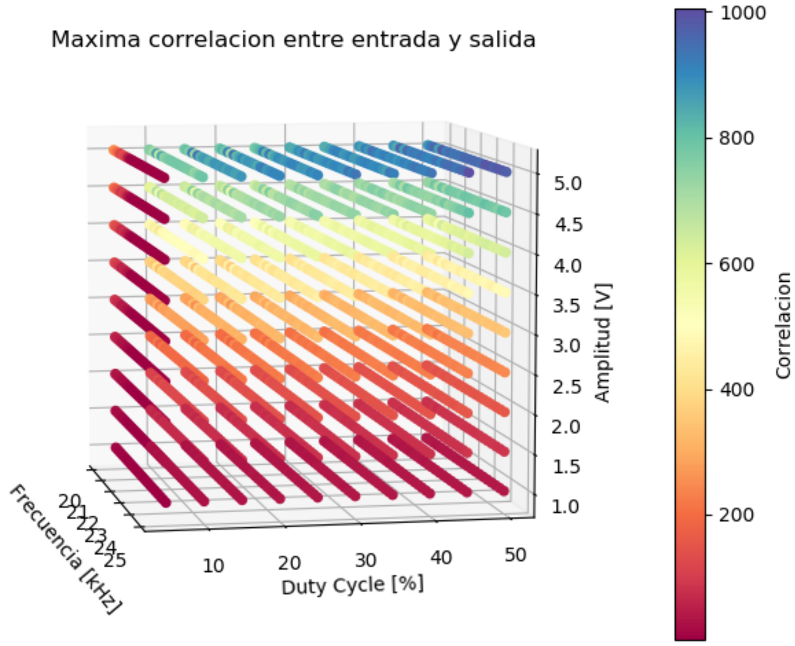


Figura 4: Scatterplot de las simulación para la **triangular con S&H**.

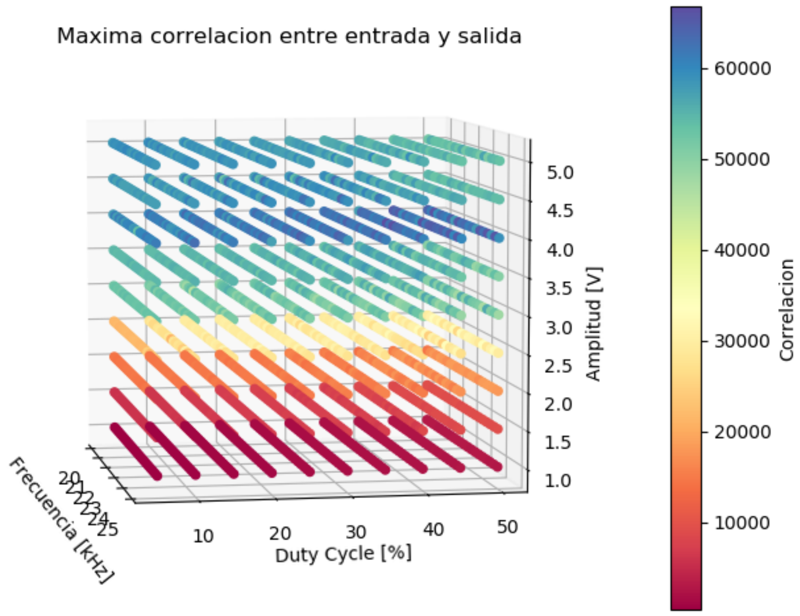


Figura 5: Scatterplot de las simulación para el **seno 3/2 con llave analógica**.

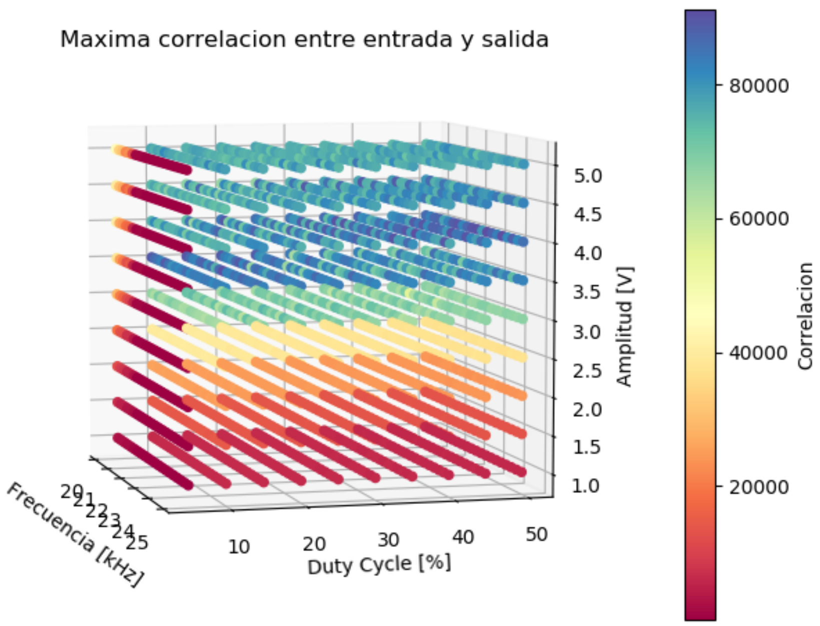


Figura 6: Scatterplot de las simulación para el **seno 3/2 con S&H**.

De esta forma se efectuaron las siguientes tablas con los parámetros para la menor distorsión para cada circuito junto a la relación entre la potencia a la salida versus la salida a la entrada, es decir, $P_r = \frac{P_{out}}{P_{in}}$:

Entrada	**A** [**V**]	$\mathbf{F_s}$ [**Hz**]	**DT** [%]	P_r
Coseno	5	24000	20	1.0196
3/2 Seno	4	23250	40	0.7991
Triangular	5	25000	45	0.6585

Tabla 2: Circuito con Sample and Hold.

Entrada	**A** [**V**]	$\mathbf{F_s}$ [**Hz**]	**DT** [%]	P_r
Coseno	5	25000	50	0.2305
3/2 Seno	4	22250	45	0.5781
Triangular	5	21250	50	0.1489

Tabla 3: Circuito con llave analógica.

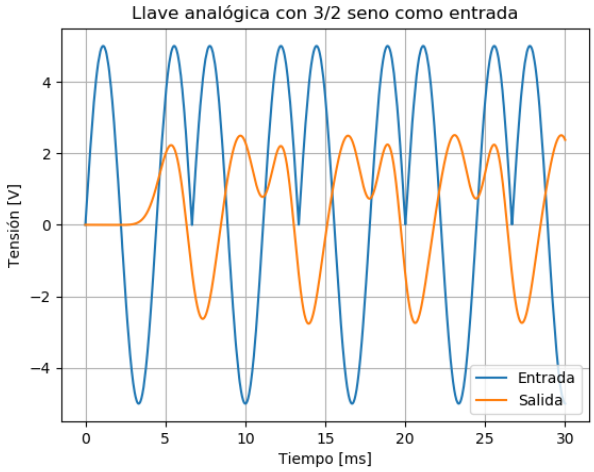
1.0.1. Simulaciones con Python

Se utilizó el framework de **GNURadio** para programar cada módulo del sistema encerrado en una interfaz gráfica, la cual brinda la posibilidad de visualizar tanto la señal en tiempo como su espectro en cada nodo del sistema en el mismo momento. Se puede elegir entre señales sinusoidales, triangulares, 3/2 seno o moduladas AM como entrada, señales cuyo estudio es de interes.

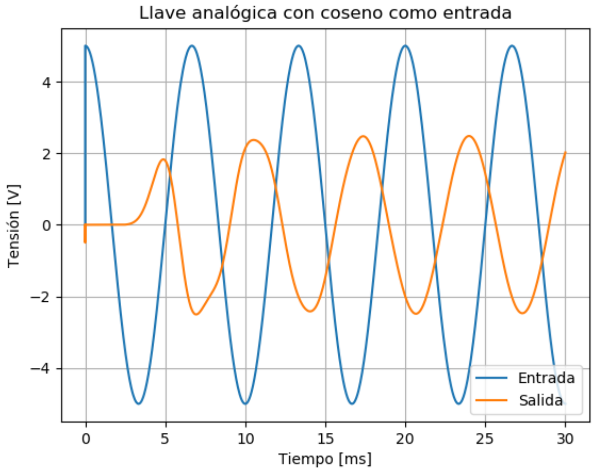
1.0.2. Simulaciones con LTSpice

Para ambos sistemas se realizaron las simulaciones de las funciones en la Tabla (1) con los parámetros que obtenidos y detallados en las Tablas (4) y (3). A continuación se presenta una comparación temporal de la entrada y salida para cada uno de los casos analizados.

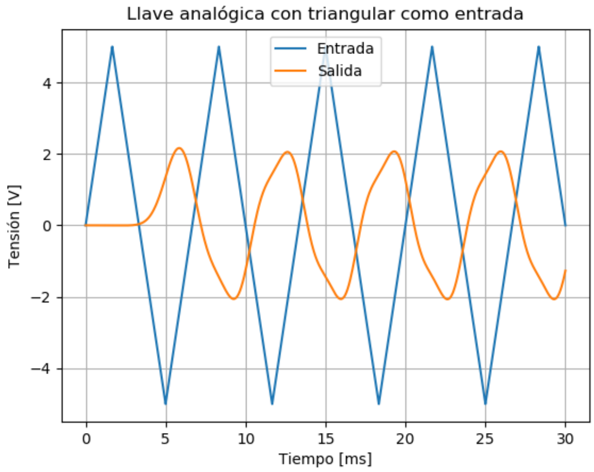
- 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png



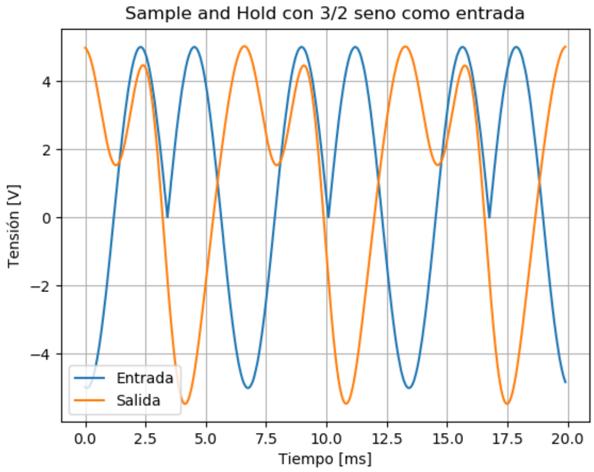
- Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png



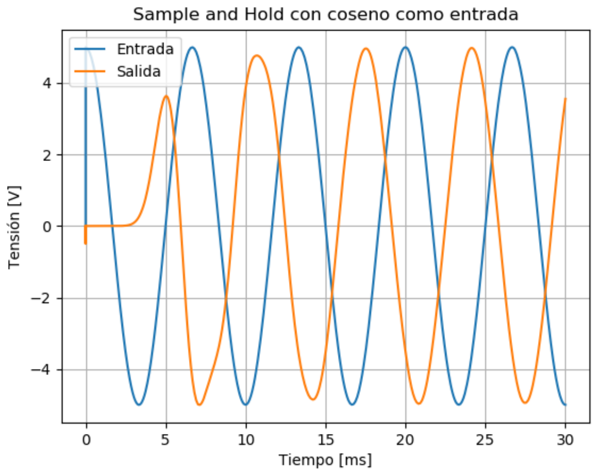
- Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png



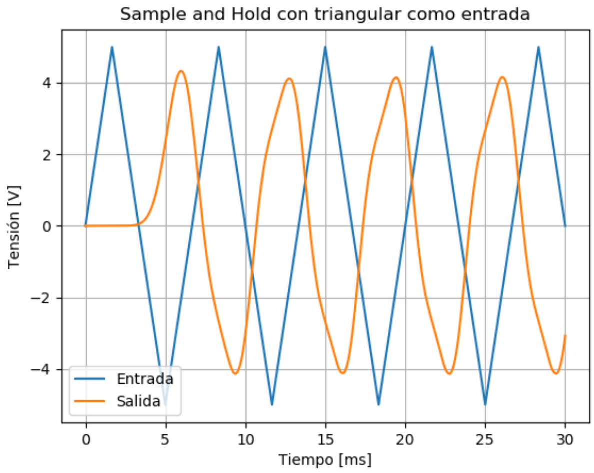
- 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png



- Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png

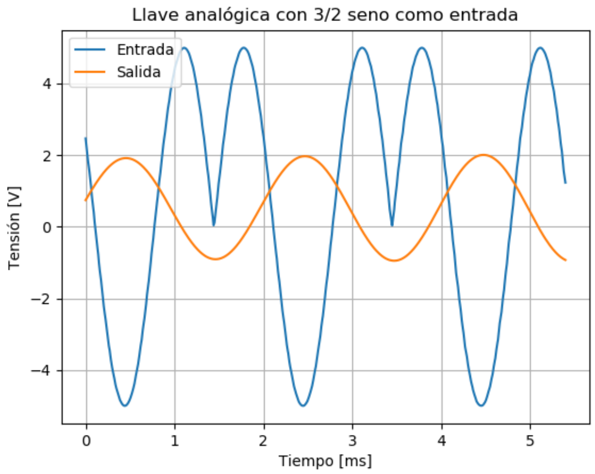


- Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png

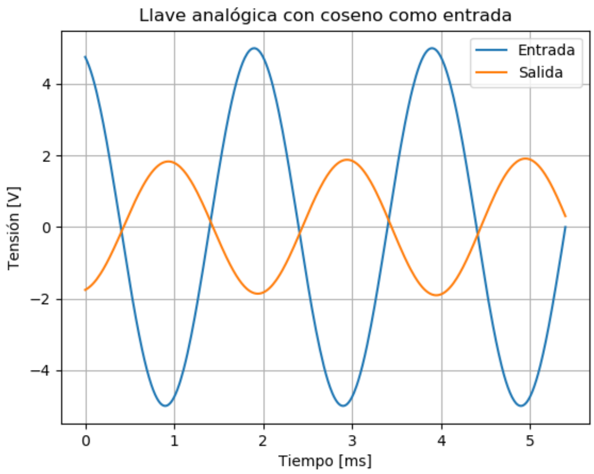


Nuevamente, se realizaron las mismas simulaciones utilizando una frecuencia de entrada de $5kHz$ y una frecuencia de sampleo de $15750Hz$.

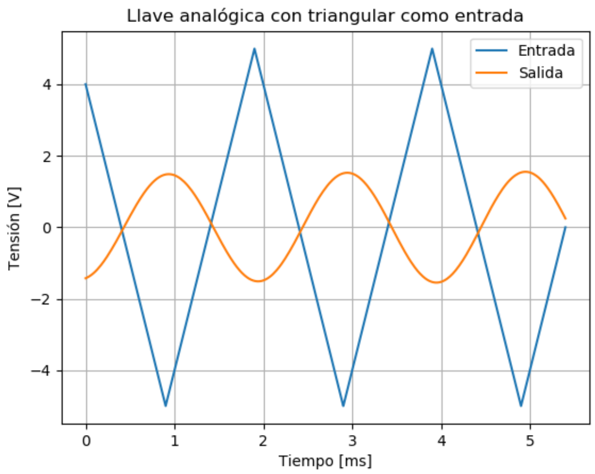
- 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png



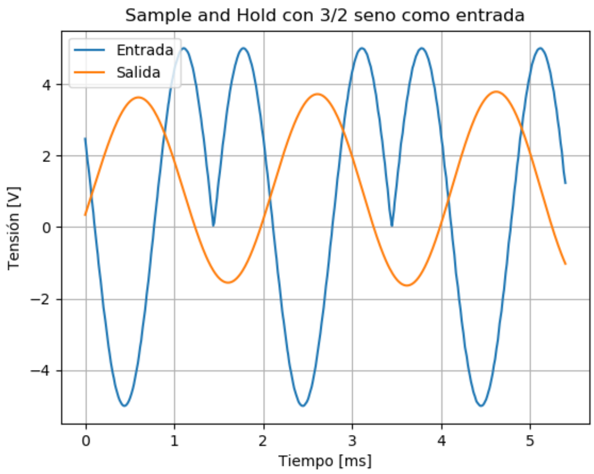
- Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png



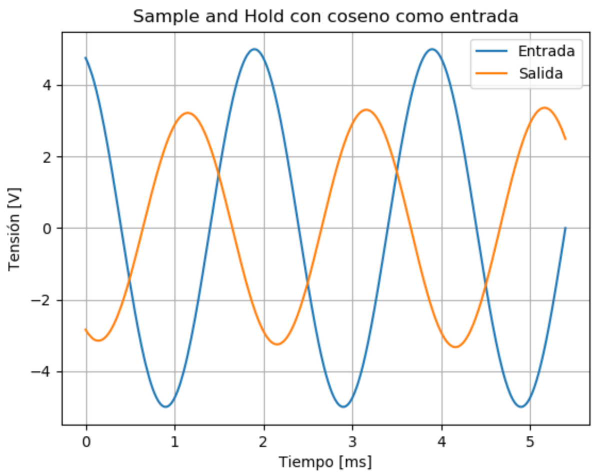
- Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png



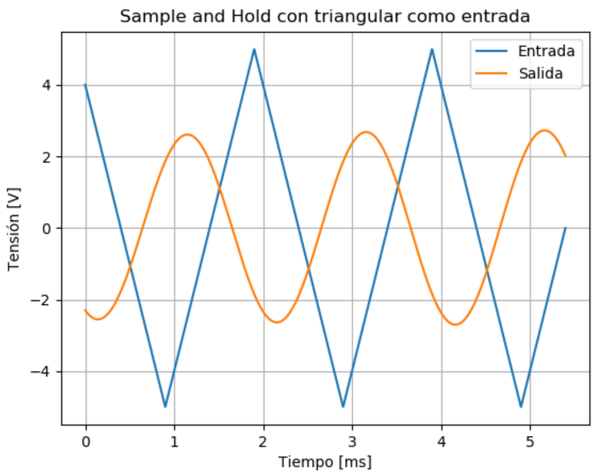
- 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png



- Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png

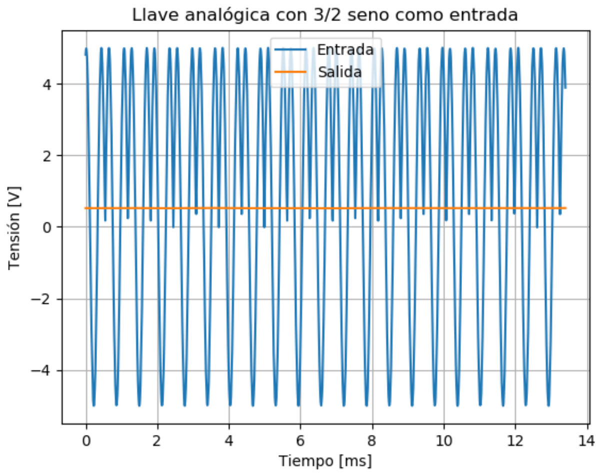


- Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png

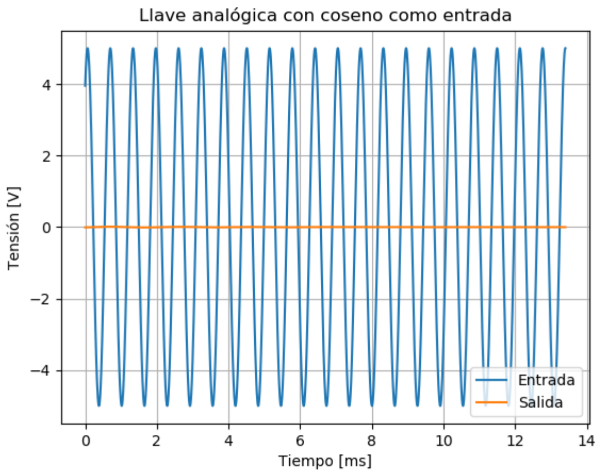


Finalmente, con una frecuencia de entrada de $15750Hz$ y una frecuencia de sampleo de $30kHz$.

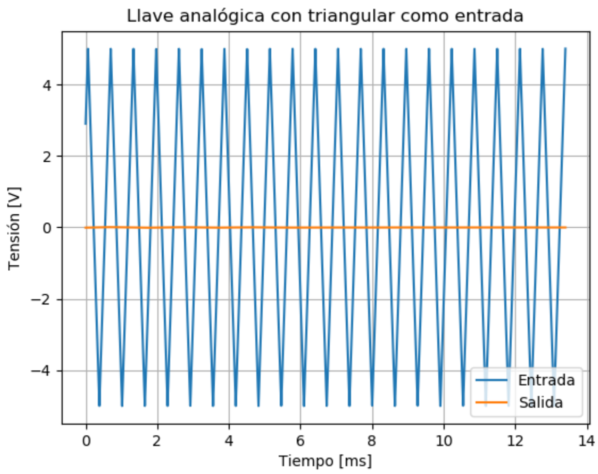
- 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png



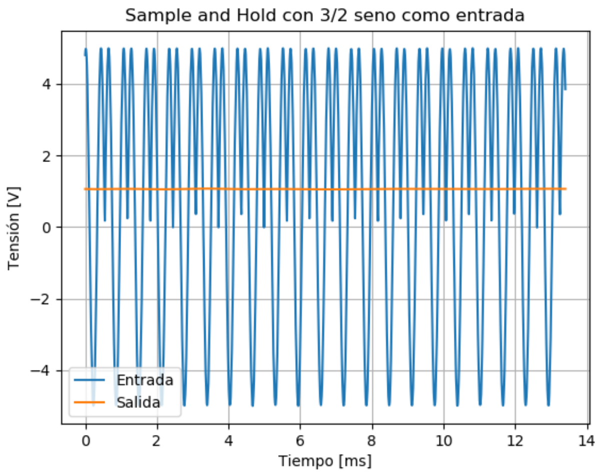
- Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png



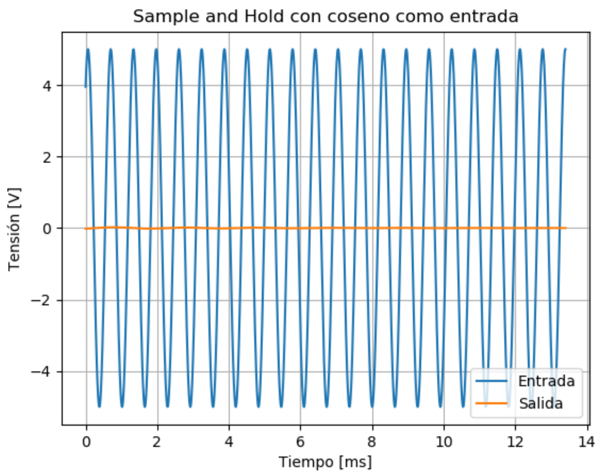
- Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png



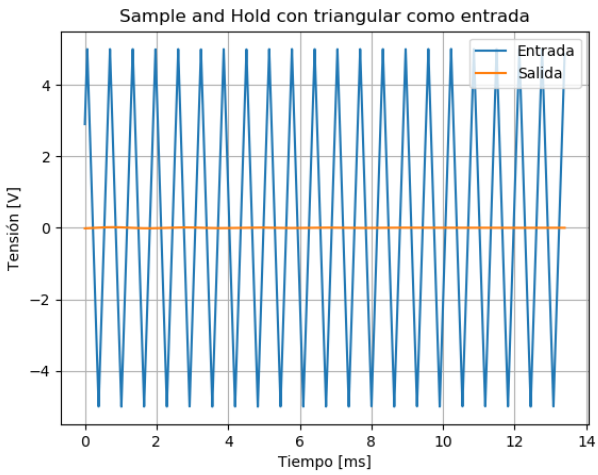
- 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png - 3 2.png



- Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png - Cos.png



- Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png - Tri.png



Se calcularon las potencias recuperadas en cada caso

Sistema + Señal	$\mathbf{Potencia con f_{in} = 5kHz f_s = 15.75kHz}$	$\mathbf{Potencia con f_{in} = 15.75kHz f_s = 30kHz}$
S&H Coseno	0.8282	$2e-06$
S&H Seno 3/2	?	?
S&H Triangular	?	?
Llave Coseno	0.3889	$6e-06$
Llave Seno 3/2	?	?
Llave Triangular	?	?

Tabla 4: Circuito con Sample and Hold.