

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 a simular

Se determinó que el valor máximo de amplitud de entrada al sistema es de $5V$ 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 $18V$ de la hoja de datos del CD4066.

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 de variación de las variables. Estos rangos son de $1V$ a $5V$ para A , $21kHz$ (para cumplir con el doble de la frecuencia de corte del filtro recuperador) a $25kHz$ (límite del oscilador) y de 5% (límite del oscilador) a 50% (límite por consigna) para el duty cycle. Finalmente, se utilizó el siguiente script en *Python* para hallar el valor de las tres variables 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)))

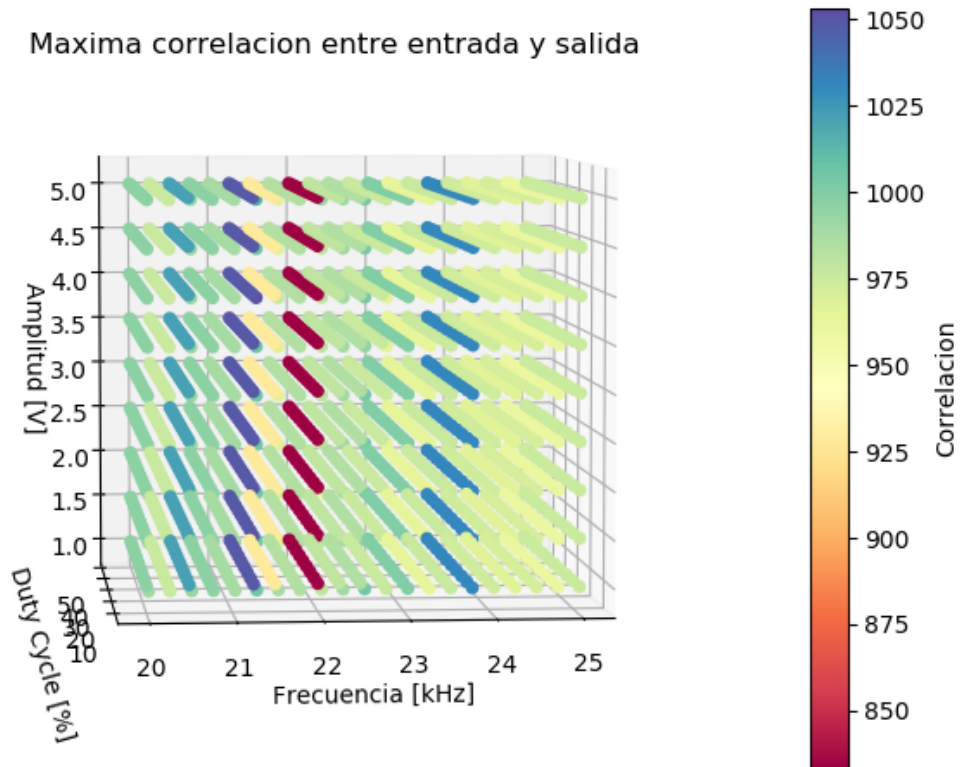
print("Least distorted: ")
for i in range(len(least_distorted_steps)):
    print(LTR.steps[(least_distorted_steps[i][0][0])])

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()
```

Esto dio los siguientes resultados:

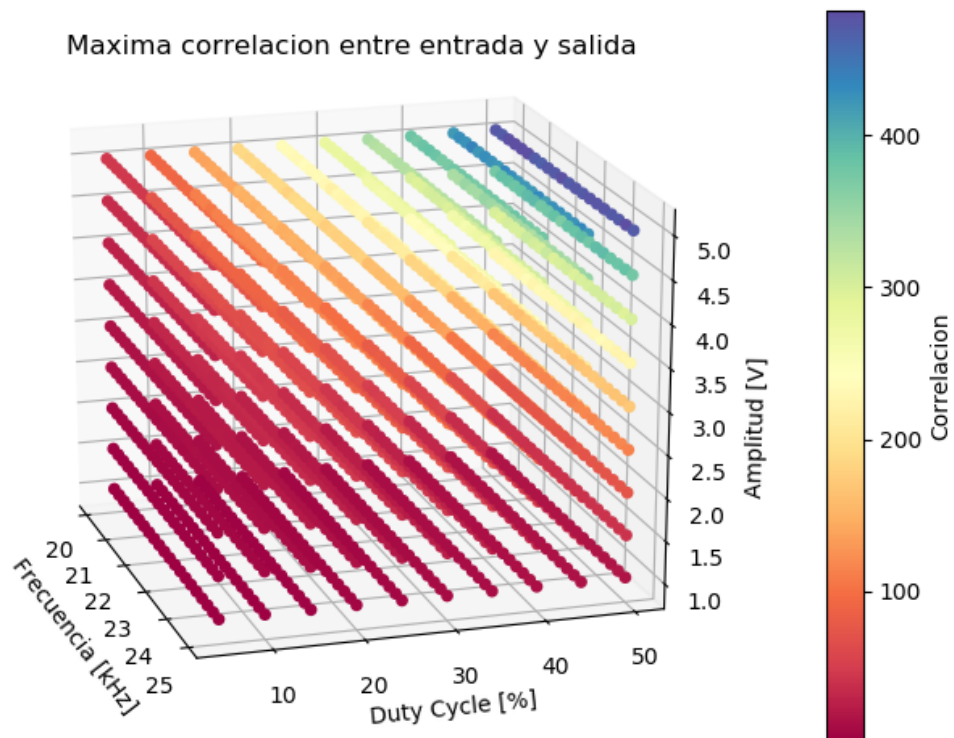


(a) Scatterplot de las simulación.

$$A = 4V \quad F_s = 21250Hz \quad DT = 5\%$$

(b) Parámetros para la menor distorsión.

Figura 1: Resultados experimentales de la simulación para **senoidal con S&H**.

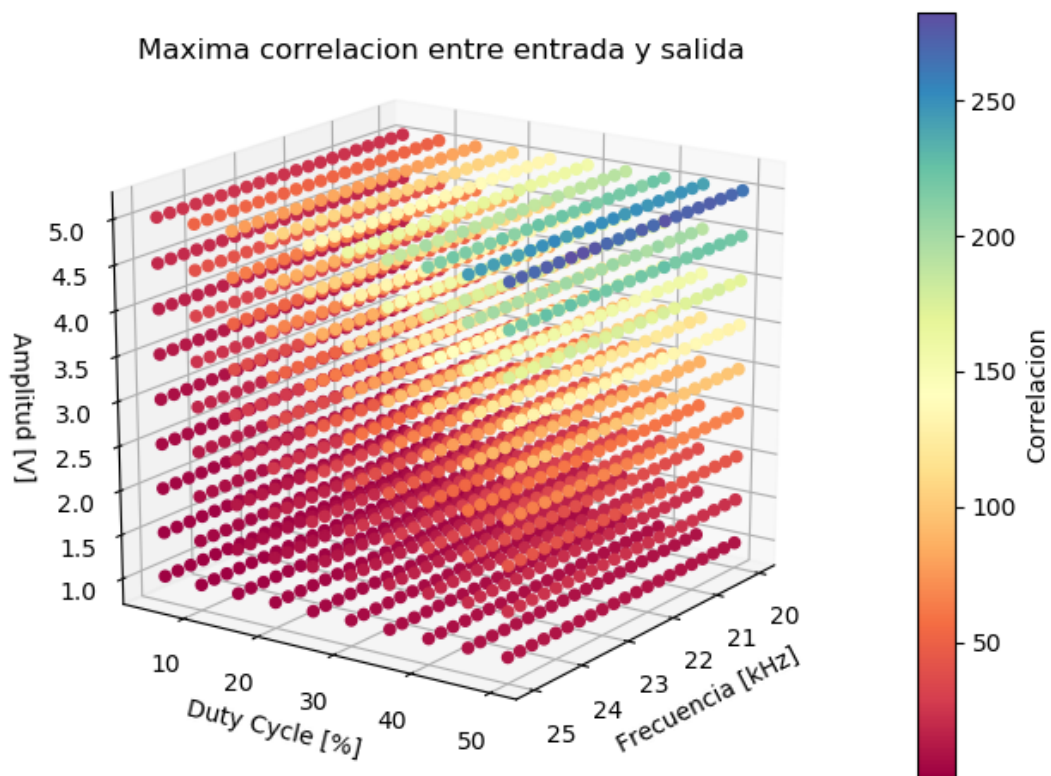


(a) Scatterplot de las simulación.

$$A = 5V \quad F_s = 21250Hz \quad DT = 50\%$$

(b) Parámetros para la menor distorsión.

Figura 2: Resultados experimentales de la simulación para **senoidal con llave analógica**.

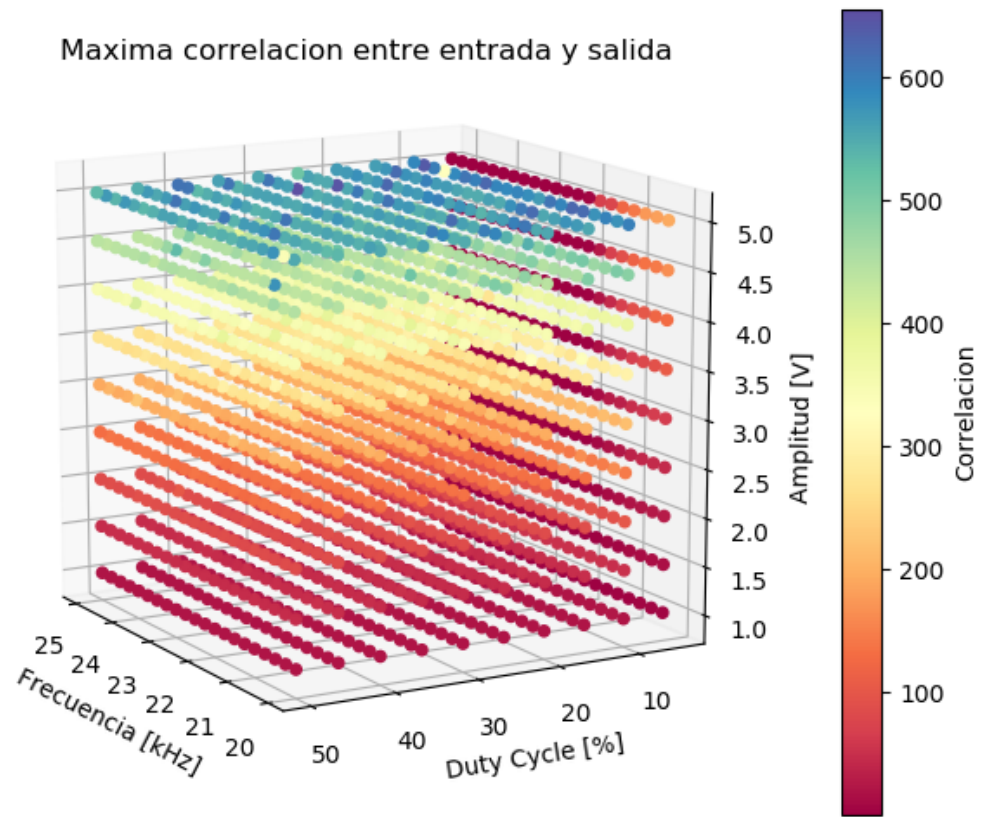


(a) Scatterplot de las simulación.

$$A = 5V \quad F_s = 23750Hz \quad DT = 50\%$$

(b) Parámetros para la menor distorsión.

Figura 3: Resultados experimentales de la simulación para **triangular con llave analógica**.

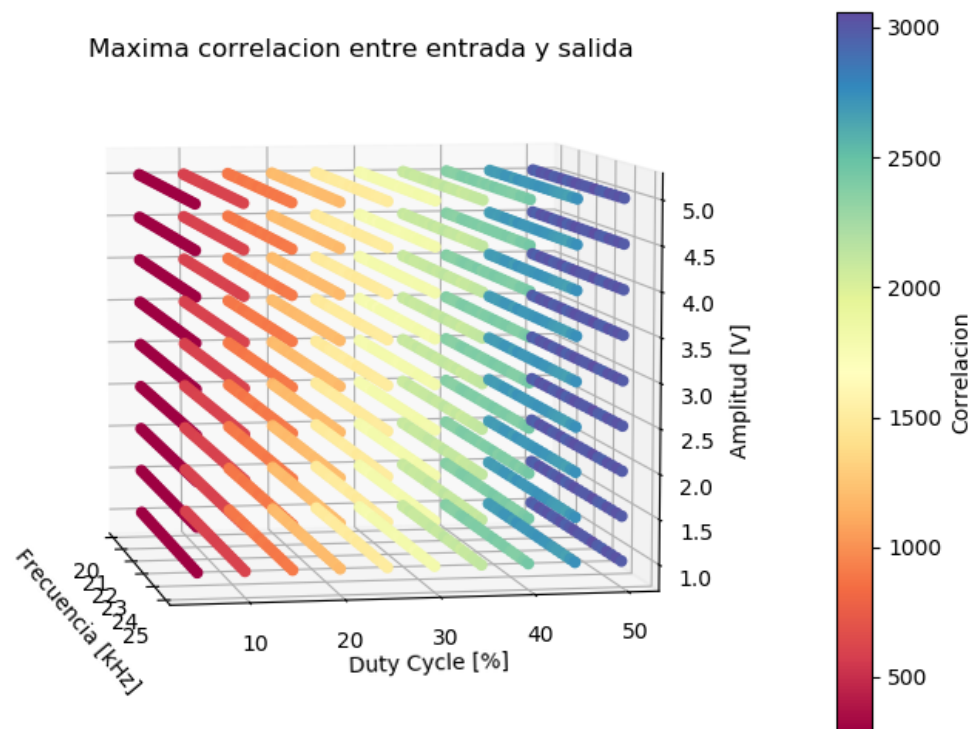


(a) Scatterplot de las simulación.

$$A = 5V \quad F_s = 24000Hz \quad DT = 30\%$$

(b) Parámetros para la menor distorsión.

Figura 4: Resultados experimentales de la simulación para **triangular con S&H**.

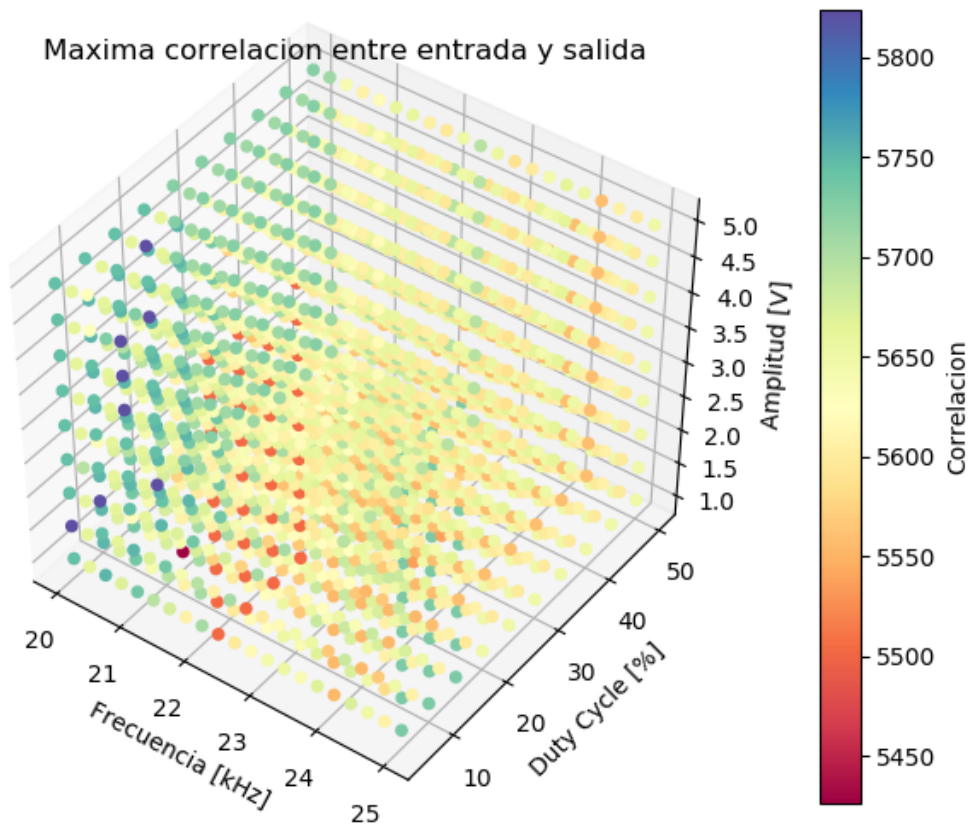


(a) Scatterplot de las simulación.

$$A = 1V \quad F_s = 23500Hz \quad DT = 50\%$$

(b) Parámetros para la menor distorsión.

Figura 5: Resultados experimentales de la simulación para **seno 3/2 con llave analógica**.



(a) Scatterplot de las simulación.

$$A = 1.5V \quad F_s = 20000Hz \quad DT = 5\%$$

(b) Parámetros para la menor distorsión.

Figura 6: Resultados experimentales de la simulación para **seno 3/2 con S&H**.

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 interés.

1.0.2. Simulaciones con LTSpice

Para ambos sistemas, tanto con la llave analógica seleccionada como para el Sample and Hold elegido, se realizaron las simulaciones de las funciones en la Tabla (1) con las excitaciones deseadas con los siguientes esquemas en *LTSPICEXVII*.

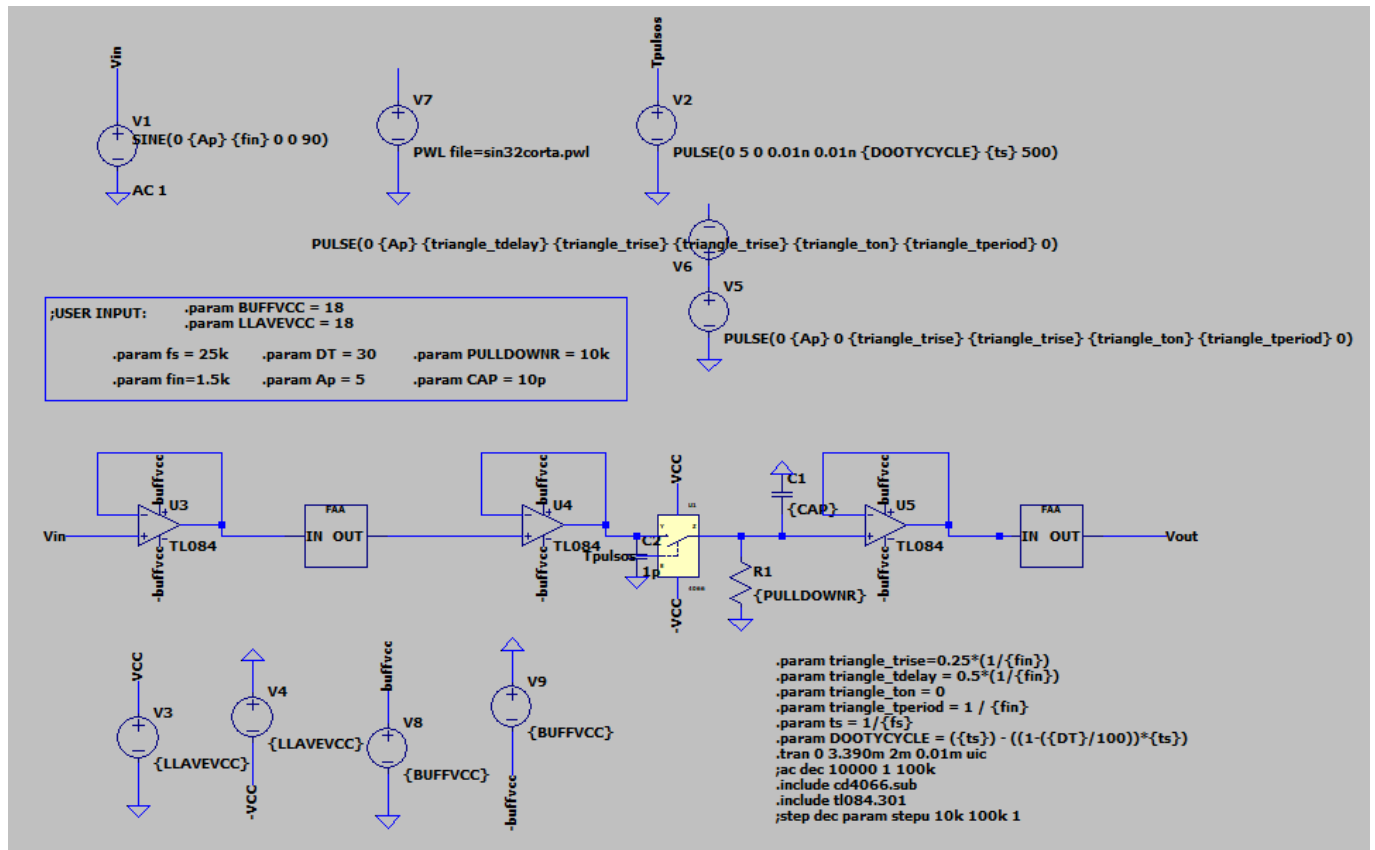


Figura 7: Esquema de simulación para el sistema con llave analógica.

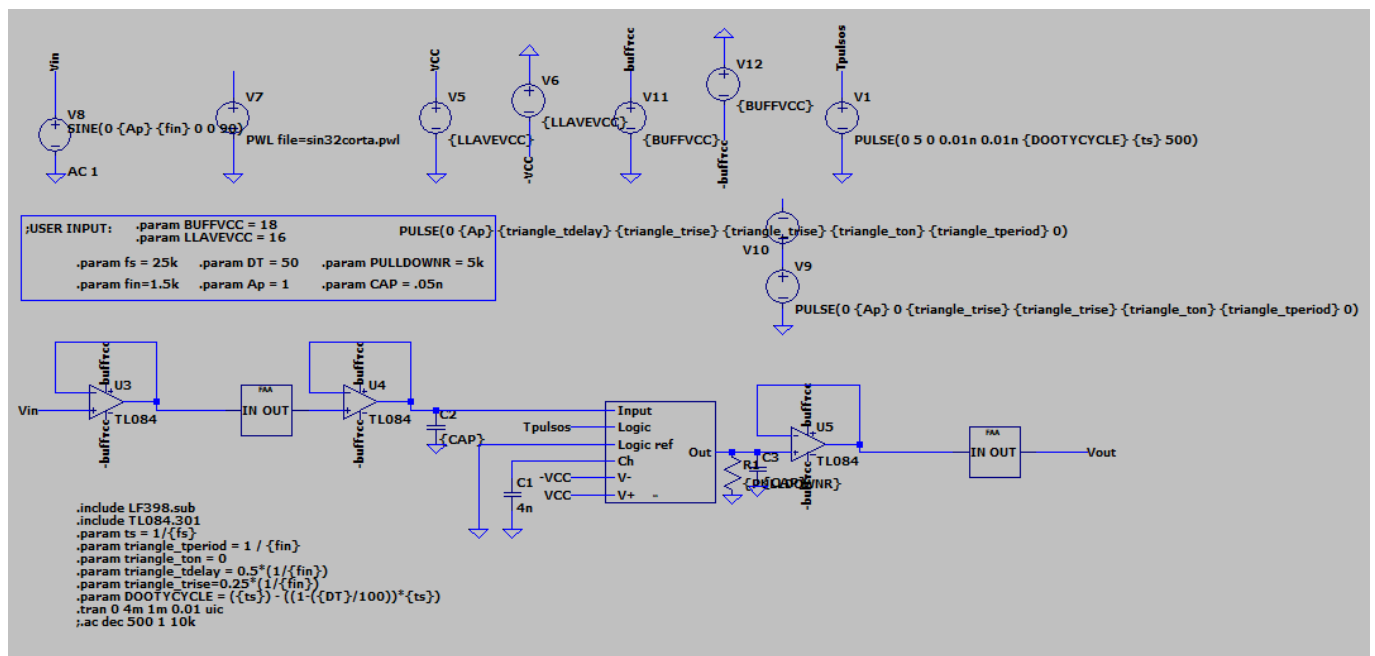


Figura 8: Esquema de simulación para el sistema con Sample and Hold.

Calcular potencia recuperada.

Nuevamente, se realizaron las mismas simulaciones variando...

Aclarar bajo qué condiciones, es el punto 6b.