

# INSTITUTO TECNOLÓGICO DE BUENOS AIRES

22.28 - ELECTRÓNICA DE POTENCIA

PARTE PRÁCTICA EXAMEN FINAL

---

## Converidor Trifásico

---

*Alumnos:*

J. Agustín BARRACHINA 53790  
Alexander BERGNER 55785  
Pedro DANS 55790  
Augusto VIOTTI BOZZINI 55138

*Profesores:*

Miguel Pablo AGUIRRE  
Pablo Martín COSSUTTA  
María Alejandra WEIL  
Matías Damián SALVATI

July 5, 2018

# 1 Introducción

## 1.1 Conversor Monofásico

En la figura 1 se puede observar el circuito de un inversor *Full Bridge* o de puente completo que se utiliza para generar una señal alterna a partir de una tensión continua  $V_d$ .

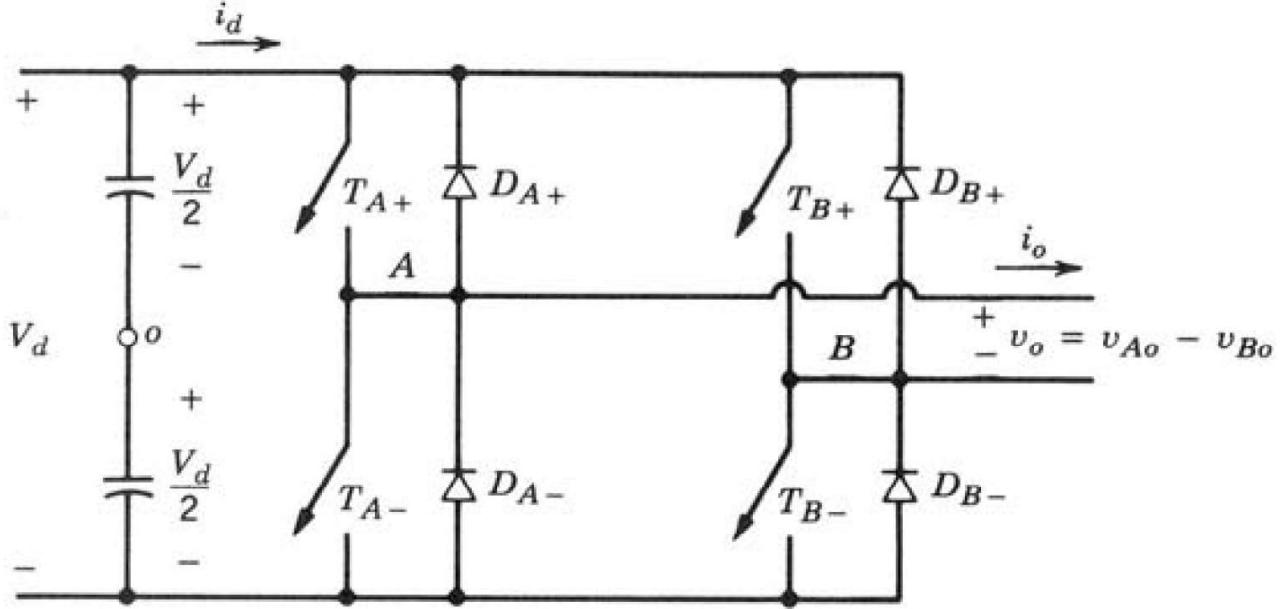


Figure 1: Inversor monofásico de puente completo

En éste circuito, la llave  $T_{A+}$  está sincronizada con la llave  $T_{B-}$  de forma que ambas están siempre en el mismo estado, el cual es opuesto a los estados de las llaves  $T_{B+}$  y  $T_{A-}$ . Si se analiza el circuito puede observarse que mientras se encuentran  $T_{A+}$  y  $T_{B-}$  prendidas al mismo tiempo (y por lo tanto  $T_{B+}$  y  $T_{A-}$  cerradas), la corriente circulará como indica la figura ( $i_d$  e  $i_o$ ) y  $v_o$  será positiva e igual a  $V_d$ . Por el contrario, cuando  $T_{B+}$  y  $T_{A-}$  estén abiertas, la corriente circulará en sentido contrario y la tensión de salida será opuesta a la tensión anterior. Sin embargo, éste método generará una tensión de salida cuadrada lo cual es indeseado, por dicho motivo se suele usar un *método de conmutación modulada por ancho de pulsos* (PWM).

En el mismo se utiliza una señal de control sinusoidal con la frecuencia deseada y una onda triangular de mucho mayor frecuencia como se observa en la figura 2 (a). Se desea que tanto la frecuencia como la tensión de la onda triangular sean lo más estables posible. Se definen  $v_{control}$ ,  $v_{tri}$  y  $f_s$  como están marcados en el gráfico. Además, se define la frecuencia deseada de la señal senoidal como  $f_1$ .

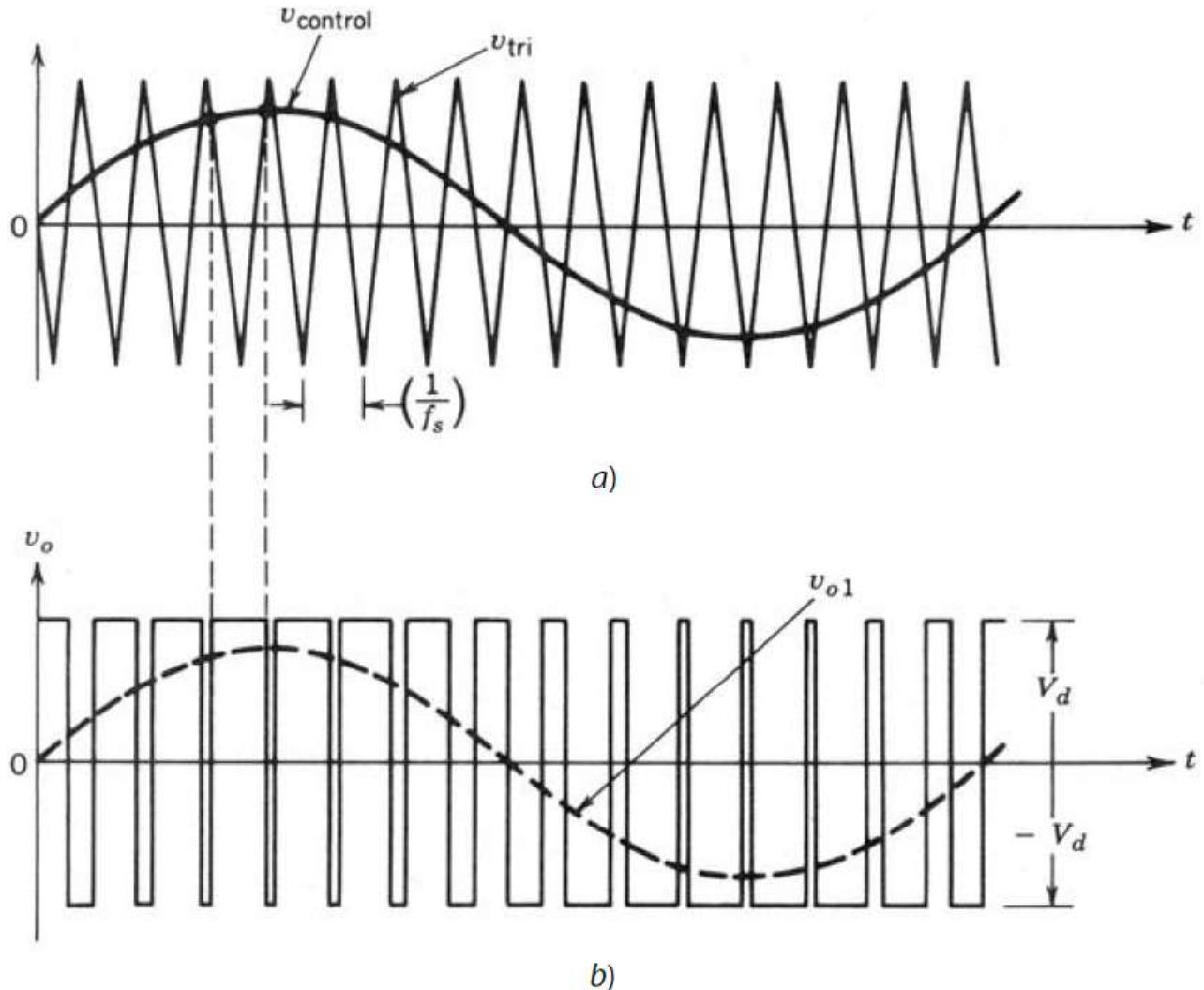


Figure 2: Modulación por ancho de pulsos

Se pueden controlar ahora los *switches* utilizando dos comparadores tal que:

$$f(x) = \begin{cases} ON & \text{si } v_{tri} \leq v_{control} \\ OFF & \text{si } v_{control} \leq v_{tri} \end{cases} \quad (1)$$

A dicha salida se la podría promediar usando un filtro pasa bajos y se obtendría entonces la señal sinusoidal deseada. Existe ciertas condiciones de borde entre las relaciones de las frecuencias y amplitudes de la señal triangular con la señal de control pero que escapan al objetivo de éste informe.

## 2 Sistema de Control

Para generar la señal de control se utilizó un Arduino MEGA. El mismo obtenía de una tabla los valores de salida que debía utilizar para cada fase.

Para generar la tabla que utiliza el Arduino se realizó un código en el lenguaje de programación python. El mismo obtiene tres señales senoidales desfasadas como corresponde para cada fase (figura 3) y una señal triangular que se utilizará para comparar y calcular la salida del PWM.

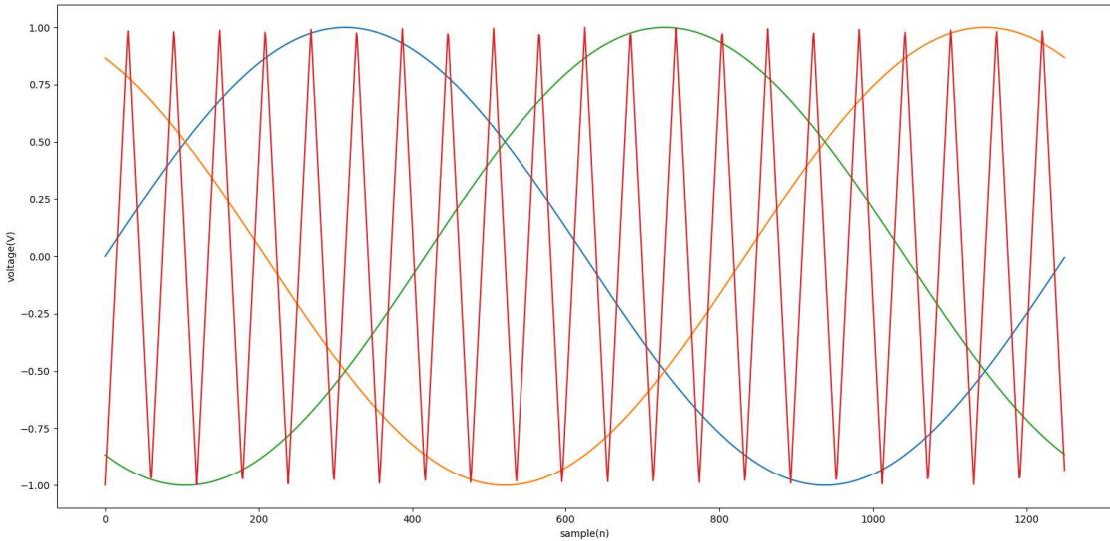


Figure 3: Señales de control y señal trifásica

Luego se implementó una pequeña función que compare las señales senoidales generadas con la señal triangular y genere un vector de 1's o 0's siguiendo la ecuación 1 donde ON es 1 y OFF es 0. Si bien se podría generar un solo vector y luego desde el arduino usar punteros inteligentemente de forma de que se generen las tres señales defasadas. Se realizó de esta manera (generando los 3 vectores por separado) para que accediendo al mismo elemento en cada tabla se realiza las fases correctamente. Si bien este método es menos eficiente en memoria, resulta más sencillo implementar la lógica para el defasaje en un lenguaje de alto nivel como es python en lugar de hacerlo en C.

En la siguiente tabla se muestra el código fuente del programa utilizado.

```

Fs = 1250
f = 1           # Frecuencia de los senos
samples = Fs    # Para que haya solo un ciclo en la tabla
x = np.arange(samples)
ma = 1           # vcontrol / vtriangular
mf = 21          # fs / f1

y1 = ma * np.sin(2 * np.pi * f * x / Fs)
y2 = ma * np.sin(2 * np.pi * f * x / Fs + 2 * np.pi / 3)
y3 = ma * np.sin(2 * np.pi * f * x / Fs + 4 * np.pi / 3)
triangle = signal.sawtooth((2 * np.pi * f * mf * x) / Fs, width=0.5)

k1 = (y1 > triangle).astype(int)
k2 = (y2 > triangle).astype(int)
k3 = (y3 > triangle).astype(int)

# write to txt file
f = open("llave1.txt", "w+")
f.write("const int k1[" + str(samples) + "] = {")
for i in range(len(x)):
    if i == 0:
        f.write(str(k1[i]))
    else:
        f.write(", " + str(k1[i])) f.write("};\n")
f.close()

```

Como se puede ver bien, el código puede dividirse en varias etapas:

- **Declaración de variables:** En la misma se genera todos los valores que puedan ser considerados como variables. De ésta forma es posible modificar fácilmente la cantidad de valores que se desea que tenga el vector o las variables  $m_a$  y  $m_f$ .
- **Generación de las señales senoidales y la triangular:**
- **Obtención del estado de las llaves:** Como Arduino funciona en C, no se podrá usar un “boolean” como tipo de dato, por lo cual se castea el resultado a “int” usando “astype(int)”.
- **Generación del vector en lenguaje C para Arduino:** Se pasó el vector a un archivo de texto para poder copiar y pegar el vector en el código C de Arduino. Por prolíjidad y simplicidad el código que se muestra en el informe sólo corresponde a la primer llave ya que las otras dos llaves se realizaron de forma análoga.

### 3 Mediciones

A continuación se adjuntan las mediciones con las respectivas explicaciones y conclusiones de cada una.

#### 3.1 Comprobación inicial del algoritmo de PWM

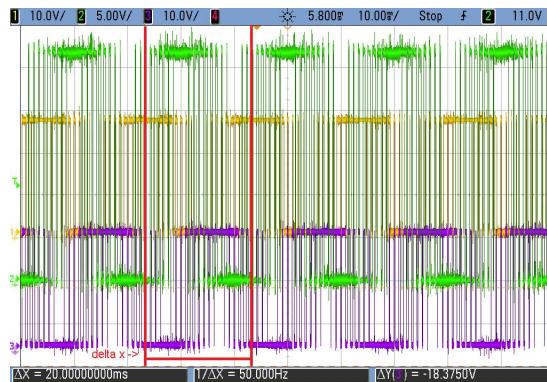


Figure 4: Salida para cada fase, con carga resistiva

Aquí se ignora la amplitud de la salida, ya que la misma es requerida para la potencia necesaria al operar con el motor. Esto es solo una verificación inicial, en la que por un lado se ve el desfasaje de las tres señales en  $120^\circ$  y por otro se ve que la frecuencia de la señal modulada es de 50 Hz. Se ignoran los efectos de *Jitter* ya que, por un lado no son de lo más visibles en el tiempo (salvo al detalle) y por otro se puede analizar mucho mejor al ver el espectro.

#### 3.2 Cálculo del THD

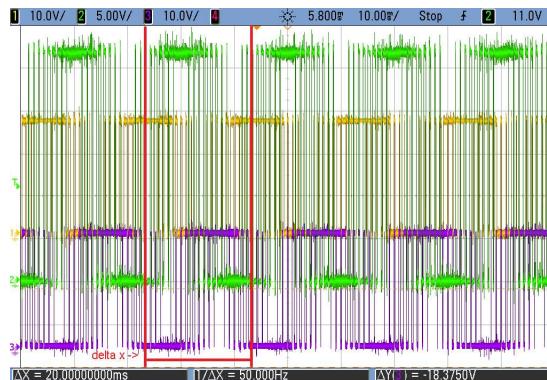


Figure 5: Salida para cada fase, con carga resistiva

Se toma una medición con más períodos (los máximos antes de que visualmente se comiencen a filtrar las commutaciones mínimas por la memoria del osciloscopio), con la cual se genera un CSV que luego se expone a la función THD de MATLAB. Se obtiene:

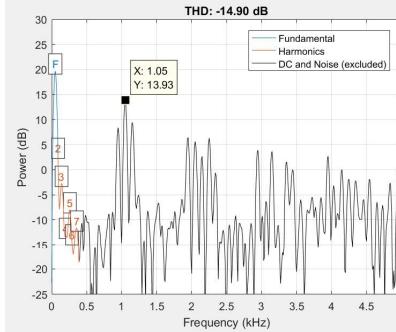


Figure 6: Imagen del THD de una de las salidas de la modulación

Este resultado es significativo ya que, en una aplicación real, se buscaría eliminar los armónicos correspondiente a la frecuencia del PWM (1050 Hz) con algún filtro, o utilizando a la misma carga como filtro. Esto no significa que no haya potencia en los armónicos de alta frecuencia, pero aún así se destaca que el THD respecto a la fundamental es el expuesto en la imagen. A continuación, se adjunta la medición correspondiente en el osciloscopio (sobre la cual NO se calcula el THD, pero se ve que el resultado es similar)

### 3.3 Medición del espectro, sobre la fase

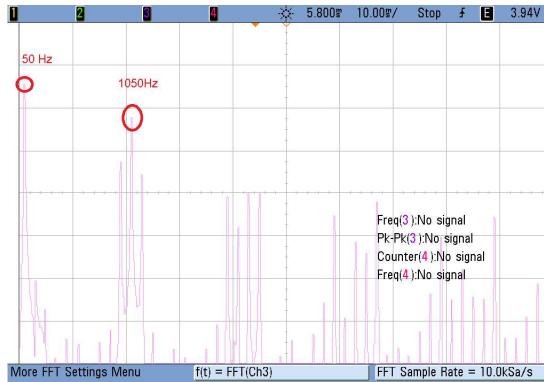


Figure 7: Medición de la FFT utilizando el osciloscopio

Aquí se ve una imagen similar a la anterior, pero medida en el osciloscopio. Se tomo uno de los canales sin pérdida de generalidad, asumiendo a los tres canales identicos en la modulación (salvando el desfasaje de  $120^\circ$ ) Solo se logra ver el primer armónico de la modulación PWM, porque al utilizar un mf impar, se eliminan los armónicos pares. Por otro lado, por ser un mf múltiplo de 3, se eliminan los armónicos múltiplos de 3. Por lo que solo se logra ver el pico en el primero.

A continuación, se adjunta la medición de THD pero esta vez sobre la línea (haciendo la resta entre dos canales). El resultado es interesante ya que se ve que se cancela el armónico correspondiente a la frecuencia del PWM. Dicho resultado se explica luego de la imagen.

### 3.4 Medición del espectro, sobre la línea

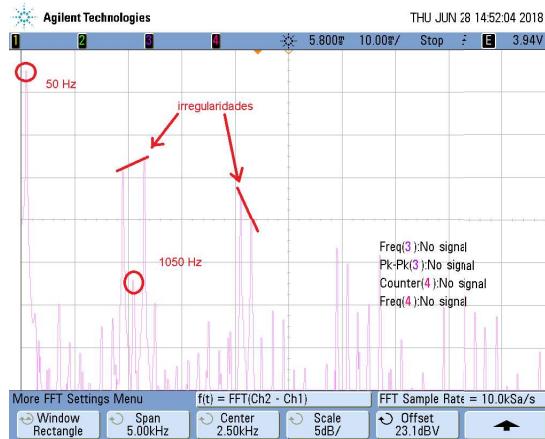


Figure 8: Medición de la FFT en la fase

Aquí se destaca como al medir la FFT de la resta de dos canales se elimina el armónico correspondiente al PWM. En los armónicos marcados como irregulares, uno esperaría encontrar la misma potencia. Esto no es así principalmente por 2 problemas:

- La resolución de la comparación entre la triangular y la senoidal no es infinita. Dado que la triangular tiene 1250 puntos, la resolución es de aproximadamente 11 bits. Esta cuantización lleva a que el espectro correspondiente a la modulación no tenga forma ideal
- Error de muestreo. En menor medida, el *Jitter* afecta al espectro. Aquí, al utilizar un arduino, a pesar de que se buscaron óptimas condiciones de muestreo (dividir el clock de cristal por potencias enteras de 2), aún así el error es relativamente alto.

El último aspecto a destacar es que el armónico correspondiente a la frecuencia de la modulación PWM está atenuado. Esto es así porque ambas fases tienen ese armónico con la misma fase y magnitud, por lo que en la resta se cancelan. La cancelación no es absoluta por haber *jitter*, el cual hace que la fase de el armónico no sea constante y la amplitud fluctúe.

#### 3.4.1 Medición de la corriente de salida, al vacío

Al conectar el motor, se obtiene la corriente de salida al vacío (torque  $\sim 0$ )

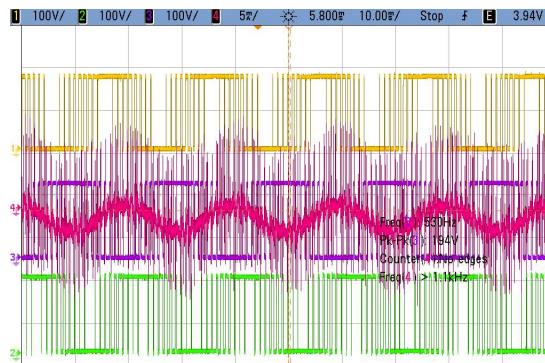


Figure 9: En rojo, la corriente de salida

Se ve la corriente de salida con una forma más senoidal que los resultados de la modulación. Esto es así porque el motor actúa como un pasabajos L-R para la corriente. A continuación, se adjunta la medición de la FFT de dicha corriente en las mismas condiciones.

### 3.5 Espectro de la corriente, al vacío

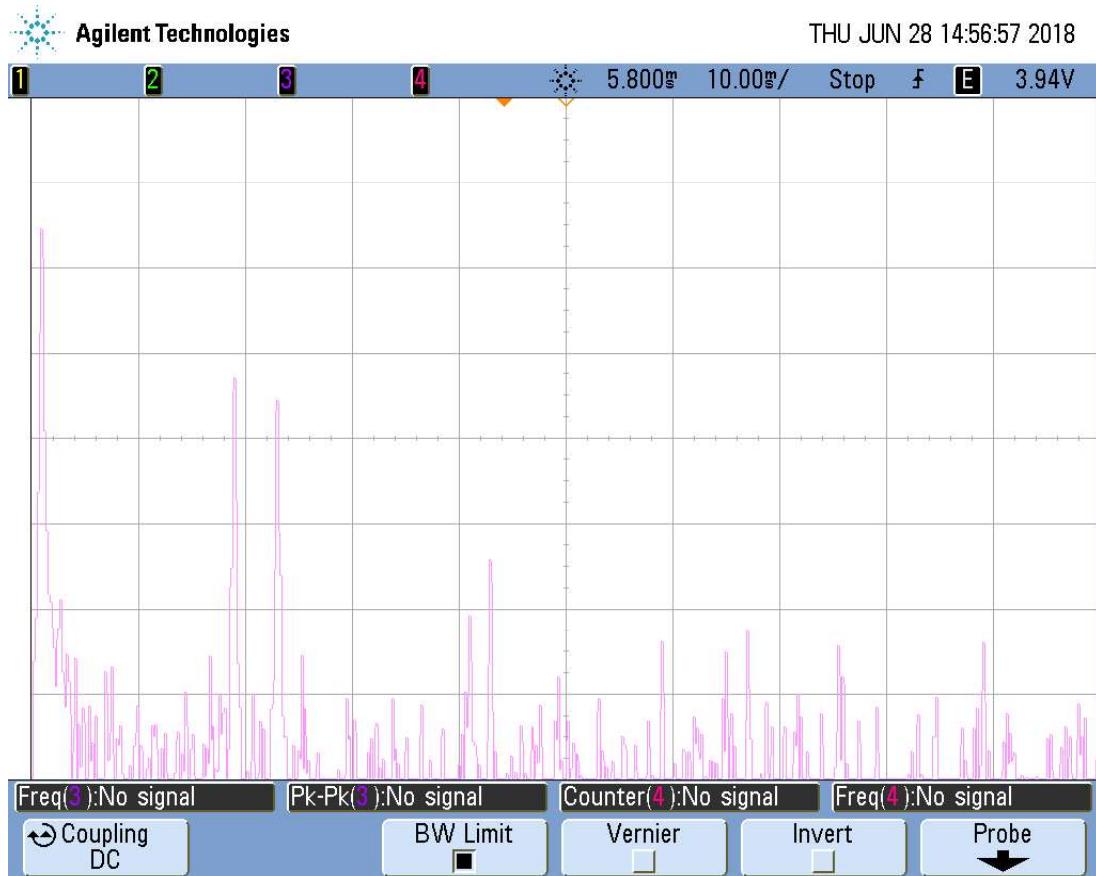


Figure 10: Espectro de la corriente al vacío

Se ve claramente como este espectro tiene mucha menor distorsión armónica que el espectro de la modulación. Esto es por el mencionado efecto de filtro que hace el motor sobre la corriente. A su vez, se ve que el armónico correspondiente a PWM (los 1050 Hz) está atenuado, ya que la corriente es corriente de fase, y como se vió en la tensión de fase, dicho armónico se atenúa de la misma manera.

### 3.6 Medición de la corriente, con torque

A continuación se adjunta la medición de la corriente intentando frenar el motor:

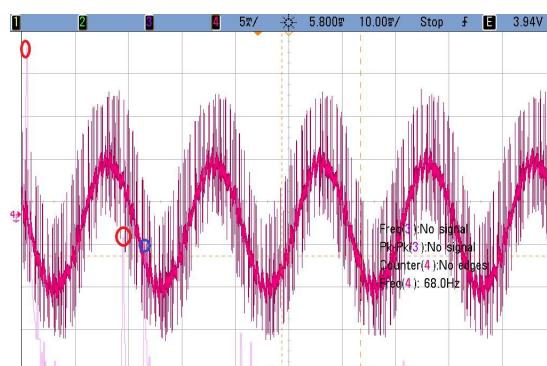


Figure 11: En rojo, la corriente de salida. En rosa, el espectro.

Al aplicar un torque, el modelo equivalente del motor cambia, haciendo mas baja la resistencia. Entonces el polo dominante impuesto por dicho modelo se mueve hacia más bajas frecuencias. En el espectro, se ve que la fundamental crece mientras que los armónicos decrecen.