

Cuaderno de Laboratorio - Tesis de Licenciatura 2025: Estudio experimental de la transferencia de energía en turbulencia de ondas gravito-capilares.

Estudiante: Ignacio Pablo Hernando¹

Director: Pablo Cobelli²

Lugar de trabajo: Laboratorio de Turbulencia Geofísica
DF, FCEN, UBA & INFINA CONICET

¹ignacioph21@gmail.com, LU: 6/21

²cobelli@df.uba.ar

Índice general

Antecedentes Teóricos	2
Semana del 31/03/2025	18
Semana del 07/04/2025	24
Semana del 14/04/2025	25
Semana del 21/04/2025	26
Semana del 28/04/2025	30
Semana del 05/05/2025	33
Semana del 12/05/2025	35
Semana del 19/05/2025	38
Semana del 26/05/2025	44
Semana del 02/06/2025	46
Semana del 16/06/2025.	50
Semana de 16/06/2025.	66

Antecedentes Teóricos

1. Introducción General.

Turbulencia

La turbulencia es un estado de un sistema físico no lineal fuera del equilibrio que cuenta con una distribución de energía sobre muchos grados de libertad. Puede mantenerse mediante un forzado, que inyecta energía, o decaer hasta el equilibrio. [1]

Esto ocurre para números de Reynold suficientemente grandes:

$$\text{Re} \equiv \frac{UL}{\nu} \quad (1)$$

Las perturbaciones producidas para una escala L tienen un bajo efecto disipativo a causa de la viscosidad respecto de los efectos no lineales, que serán los dominantes. Estas no linealidades inducen movimiento en escalas cada vez más chicas, hasta llegar a la escala disipativa donde la viscosidad lidera, en un rango mucho menor a L . En el intermedio existe un amplio rango (llamado inercial) donde la viscosidad es despreciable y domina la no linealidad.

El flujo de energía hacia las escalas más chicas se da en un proceso de tipo *cascada*. Esta idea fue formulada por Richardson (1922):

*Big whorls have little whorls,
Which feed on their velocity,
And little whorls have lesser whorls,
And so on to viscosity.*

Donde por la ruptura de vórtices grandes se forman más y más chicos. Por esta idea también a veces se le llama a esta turbulencia *Eddy Turbulence*. Además asumimos que la cascada es *local*, o sea que se produce una transferencia de energía de forma continua entre escalas.

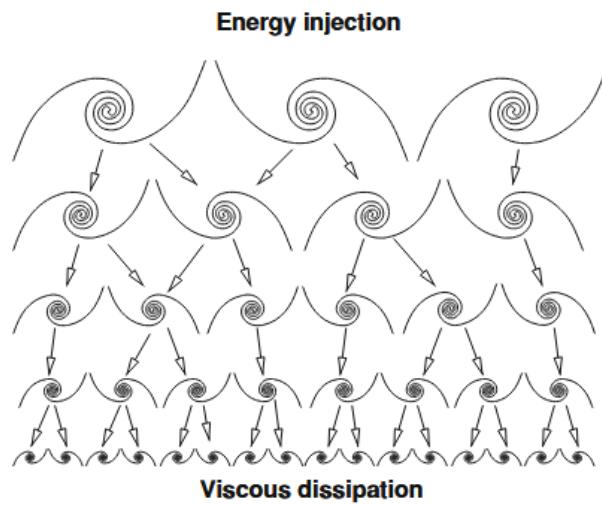


Figura 1: Representación pictórica de la cascada de energía de Richardson. Imagen tomada de [2].

Para poder describir a un sistema dinámico tan complejo es necesario recurrir a herramientas estadísticas.

Kolmogorov-Obukhov plantearon un argumento dimensional donde los únicos parámetros relevantes en el rango inercial son el flujo de energía ε y k . En particular para el caso de turbulencia

homogénea (no depende de la posición) e isótropa (espectro no depende de la dirección de \vec{k}), o HIT, el espectro de energía en Fourier resulta:

$$E(k) = 4\pi k^2 E(\vec{k}) = C\varepsilon^{2/3} k^{-5/3} \quad (2)$$

Que es uno de los resultados más fuertes sobre turbulencia, en acuerdo con muchos resultados experimentales. Sin embargo, para momentos más altos del campo de velocidades empieza a haber diferencias más y más grandes, debido al fenómeno de *intermitencia*, donde se producen fluctuaciones en locales en ε .

Turbulencia 2D

Cuando tenemos un flujo ideal incompresible en 2D tenemos dos cantidades conservadas:

$$E = \frac{1}{2} \langle \vec{u}^2 \rangle = \int E(k) dk \quad \Omega = \frac{1}{2} \langle \vec{\omega}^2 \rangle = \int k^2 E(k) dk \quad (3)$$

Que son la energía y enstrofía por unidad de área. Vamos a tener entonces una *cascada dual*, de energía y enstrofía. Para saber de qué tipo va a ser cada una recurrimos al argumento de Fjørtoft:

Supongamos que inyectamos energía a un ratio ε en la escala k_0 y por tanto enstrofía a $\eta \sim k_0^2 \varepsilon$. Imaginemos que disipamos energía en $k_+ \ll k_0$ como ocurriría en el caso 3D, a un ritmo equivalente al de inyección ε . Entonces la enstrofía se disiparía a un ritmo $k_+^2 \varepsilon \gg k_0^2 \varepsilon \sim \eta$. Esto es una contradicción ya que se estaría disipando enstrofía a un ritmo mayor que el que se inyecta, lo cual no es posible en un sistema estadístico estacionario. Por tanto debe ser que la energía se disipa en $k_- \ll k_0$, y entonces es una *cascada inversa de energía*. Por una lógica análoga concluimos que tenemos una *cascada directa de enstrofía*. De esta forma los vórtices pequeños son estirados por los grandes, y se van uniendo.

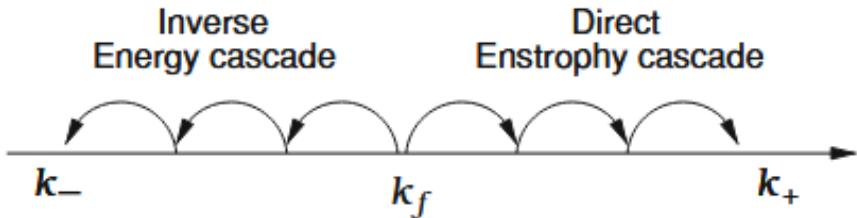


Figura 2: Cascada dual de energía y enstrofía en turbulencia 2D. Imagen de [2].

El espectro de energía en la cascada inversa sigue la misma ley de escala anterior, pero con otra constante, mientras que la ley de escala para la cascada directa resulta de tomar como parámetro a η en lugar de ε :

$$E(k) = C_\eta \eta^{2/3} k^{-3} \quad (4)$$

Turbulencia de Ondas

La turbulencia de ondas es una descripción estadística de un sistema mecánico fuera del equilibrio formado por ondas no lineales aleatorias. La diferencia principal respecto a la Turbulencia clásica es que ahora vamos a tener un sistema de ondas en lugar de un sistema de vórtices.

Una ventaja importante de este tipo de turbulencia es que surge una clausura de forma natural para la jerarquía de momentos estadísticos, que permite calcular más resultados de forma analítica.

2. Formalismo Hamiltoniano para ondas en medios continuos.

Las bases.

En el caso más simple posible podemos describir un medio continuo a partir de un par de variables canónicas $q(\vec{r}, t)$ y $p(\vec{r}, t)$ de forma tal que las ecuaciones de movimiento serán:

$$\frac{\partial q}{\partial t} = \frac{\delta \mathcal{H}}{\delta p} \quad \frac{\partial p}{\partial t} = -\frac{\delta \mathcal{H}}{\delta q} \quad (5)$$

Aquí tenemos la densidad Hamiltoniana \mathcal{H} , y los δ representan la *derivada funcional*, análoga a la derivada parcial en el límite continuo. Vamos a tener las propiedades análogas a la versión para variables finitas:

$$\frac{\delta q(\vec{r})}{\delta q(\vec{r}')} = \delta^3(\vec{r} - \vec{r}') \quad \frac{\delta p(\vec{r})}{\delta p(\vec{r}')} = \delta^3(\vec{r} - \vec{r}') \quad (6)$$

Además, aplica la regla de la cadena:

$$\frac{\delta f(\vec{r})}{\delta q(\vec{r}')} = \frac{\partial f(\vec{r})}{\partial q(\vec{r})} \frac{\delta q(\vec{r})}{\delta q(\vec{r}')} \quad \frac{\delta f(\vec{r})}{\delta p(\vec{r}')} = \frac{\partial f(\vec{r})}{\partial p(\vec{r})} \frac{\delta p(\vec{r})}{\delta p(\vec{r}')} \quad (7)$$

Y conmuta con derivadas parciales de las variables espaciales (e.g. ∂_{r^i}) y con integración en variables especiales (e.g. $\int d^3r$).

Además será importante definir el **corchete de Poisson**. Para un conjunto de variables discretas (q_i, p_i) es (el orden de los dos términos es convención):

$$\{A, B\} = \sum_i \left(\frac{\partial A}{\partial p_i} \frac{\partial B}{\partial q_i} - \frac{\partial A}{\partial q_i} \frac{\partial B}{\partial p_i} \right) \quad (8)$$

De forma que en variables canónicas tenemos:

$$\{q_i, q_j\} = \{p_i, p_j\} = 0 \quad \{q_i, p_j\} = \delta_{ij} \quad (9)$$

Y en el límite continuo esto resulta en:

$$\{A, B\} = \int d^3r \left(\frac{\delta A}{\delta p} \frac{\delta B}{\delta q} - \frac{\delta A}{\delta q} \frac{\delta B}{\delta p} \right) \quad (10)$$

Y la condición de canonicidad queda expresada como:

$$\{q(\vec{r}), q(\vec{r}')\} = \{p(\vec{r}), p(\vec{r}')\} = 0 \quad \{q(\vec{r}), p(\vec{r}')\} = \delta^3(\vec{r} - \vec{r}') \quad (11)$$

Esto es fundamental, ya que las transformaciones canónicas, que preservan la forma de las ecuaciones (5), serán aquellas que preservan estos corchetes de Poisson ($\{A(q', p'), B(q', p')\}_{q,p}$).

Teoremas de Conservación y simetrías.

La derivada temporal de una función (con una posible dependencia explícita en t) será:

$$\frac{df(q, p, t)}{dt} = \{H, f\} + \frac{\partial f}{\partial t} \quad (12)$$

En general no hay dependencias explícitas, con lo cual tenemos que si $\{H, f\} = 0$ entonces la cantidad f es constante de movimiento.

Pasar a variables complejas.

Ahora bien, se puede mostrar que la siguiente transformación es canónica (a menos de un factor multiplicativo):

$$a(\vec{r}, t) = \frac{1}{\sqrt{2}}(Q(\vec{r}, t) + iP(\vec{r}, t)) \quad a^*(\vec{r}, t) = \frac{1}{\sqrt{2}}(Q(\vec{r}, t) - iP(\vec{r}, t)) \quad (13)$$

Con $Q = \lambda q$ y $P = p/\lambda$ tales que Q y P tengan las mismas unidades. La ventaja de pasar a las variables complejas es que se reduce de las dos ecuaciones (5) a una sola (la segunda es la conjugada de la primera):

$$i \frac{\partial a}{\partial t} = \frac{\delta \mathcal{H}}{\delta a^*} \quad (14)$$

Con los corchetes en las coordenadas originales³:

$$\{a(\vec{r}), a(\vec{r}')\}_{Q,P} = \{a^*(\vec{r}), a^*(\vec{r}')\}_{Q,P} = 0 \quad \{a(\vec{r}), a^*(\vec{r}')\}_{Q,P} = i\delta(\vec{r} - \vec{r}') \quad (15)$$

Estas variables complejas son análogas a los operadores de subida y bajada en cuántica.

Las cantidades conservadas

En un primer lugar tendremos la conservación de la energía total (siempre que trabajemos con el modelo más sencillo), pero en el caso que el Hamiltoniano tenga simetría $U(1)$ en las variables a , habrá otra ley de conservación asociada, $N = |a|^2$, que es análogo al número de partículas en el caso cuántico. Ya desarrollaremos esto más adelante.

Paso al espacio de Fourier.

Vamos a imaginar que estamos en una caja de largo L con condiciones de contorno periódicas, que eventualmente podemos hacer tender a infinito, y vamos a expresar a la variable $a(\vec{r})$ como una combinación de las variables $a_{\vec{k}} \equiv a(\vec{k})$ tal que:

$$a_{\vec{k}} = a(\vec{k}) = \frac{1}{V} \int d\vec{r} a(\vec{r}) e^{-i\vec{k} \cdot \vec{r}} \quad a(\vec{r}) = \sum_{\vec{k}} a_{\vec{k}} e^{i\vec{k} \cdot \vec{r}} \quad (16)$$

Es posible mostrar usando los corchetes para la variable a en el espacio real que las nuevas variables en el espacio de momento $a_{\vec{k}}$ satisfacen los mismos corchetes de Poisson, a menos de un factor de volumen. Entonces las ecuaciones de movimiento mantienen su forma:

$$i \frac{\partial a_{\vec{k}}}{\partial t} = \frac{\delta \mathcal{H}}{\delta a_{\vec{k}}^*} \quad (17)$$

3. Desarrollo perturbativo del Hamiltoniano.

Nos va a interesar escribir el Hamiltoniano como una serie de potencias de a y a^* . El término de orden 0, i.e. \mathcal{H}_0 , no es relevante, ya que es una constante que no aparece en la ecuación de movimiento al derivar, y el término de orden 1, \mathcal{H}_1 , podemos eliminarlo asumiendo que el medio está en equilibrio cuando las amplitudes de las ondas son 0, y el mínimo está en $a = a^* = 0$. De esta forma nos queda:

$$\mathcal{H} = \mathcal{H}_2 + \mathcal{H}_3 + \mathcal{H}_4 + \dots \equiv \mathcal{H}_2 + \mathcal{H}_{int} \quad (18)$$

Donde \mathcal{H}_{int} representa las interacciones entre ondas.

³A veces, parece que en las nuevas variables $(\{\cdot, \cdot\}_{a,a^*})$ se escribe el corchete con un factor i adelante.

El orden más bajo.

Se puede mostrar que existe una transformación canónica a unas nuevas variables $b_{\vec{k}}, b_{\vec{k}^*}$ tales que [3]:

$$\mathcal{H}_2 = \int \omega(\vec{k}) b_{\vec{k}} b_{\vec{k}}^* d\vec{k} \quad (19)$$

De forma tal que a primer orden no nulo la ecuación de movimiento resulta trivial:

$$\frac{\partial b_{\vec{k}}}{\partial t} = -i\omega b_{\vec{k}} \quad \Rightarrow \quad b_{\vec{k}} = b(\vec{k}, 0) e^{-i\omega t} \quad (20)$$

En este punto la única diferencia entre distintos medios radica en la relación de dispersión $\omega(\vec{k})$.

El Hamiltoniano de interacción.

Los términos de orden superior se pueden interpretar como términos de interacción entre ondas, en analogía a lo que ocurre en mecánica cuántica. Vamos a usar la notación reducida: $b_{\vec{k}_1} \equiv b_1$.

Para las interacciones de tres ondas tenemos:

$$\begin{aligned} \mathcal{H}_3 &= \frac{1}{2} \int (V_{123} b_1^* b_2 b_3 + \text{c.c.}) \delta(\vec{k}_1 - \vec{k}_2 - \vec{k}_3) d\vec{k}_1 d\vec{k}_2 d\vec{k}_3 \\ &\quad + \frac{1}{6} \int (U_{123} b_1^* b_2^* b_3^* + \text{c.c.}) \delta(\vec{k}_1 + \vec{k}_2 + \vec{k}_3) d\vec{k}_1 d\vec{k}_2 d\vec{k}_3 \end{aligned} \quad (21)$$

Donde tenemos unos coeficientes V_{123} y U_{123} que nos hablan de la magnitud de la interacción. El primer término describe procesos de la forma $1 \rightarrow 2$ y $2 \rightarrow 1$, mientras que el segundo describe procesos del tipo $3 \rightarrow 0$ y $0 \rightarrow 3$, que son aniquilaciones de tres ondas hacia el vacío o creación espontánea de tres ondas por fluctuaciones.

Por otro lado, para las interacciones de cuatro ondas tenemos:

$$\begin{aligned} \mathcal{H}_4 &= \frac{1}{4} \int W_{1234} b_1^* b_2^* b_3 b_4 \delta(\vec{k}_1 + \vec{k}_2 - \vec{k}_3 - \vec{k}_4) d\vec{k}_1 d\vec{k}_2 d\vec{k}_3 d\vec{k}_4 \\ &\quad + \int (G_{1234} b_1 b_2^* b_3^* b_4^* + \text{c.c.}) \delta(\vec{k}_1 - \vec{k}_2 - \vec{k}_3 - \vec{k}_4) d\vec{k}_1 d\vec{k}_2 d\vec{k}_3 d\vec{k}_4 \\ &\quad + \int (R_{1234}^* b_1 b_2 b_3 b_4 + \text{c.c.}) \delta(\vec{k}_1 + \vec{k}_2 + \vec{k}_3 + \vec{k}_4) d\vec{k}_1 d\vec{k}_2 d\vec{k}_3 d\vec{k}_4 \end{aligned} \quad (22)$$

Donde ahora tenemos procesos del tipo $2 \rightarrow 2$ para el primer término, $1 \rightarrow 3$ y $3 \rightarrow 1$ para el segundo, $4 \rightarrow 0$ y $0 \rightarrow 4$ para el tercero.

Los coeficientes de interacción van a tener las simetrías asociadas a la interacción, o sea, se van a poder intercambiar los índices asociados a "partículas" de cada lado de la flecha, por ejemplo, $V_{123} = V_{132}$.

Notemos además que si hay presentes términos con una cantidad distinta de "partículas" de un lado que del otro entonces no habrá simetría $U(1)$ y no se conservará la cantidad de partículas asociadas a ese proceso.

Interacciones Resonantes de N ondas

En general vamos a poder tener interacciones de tanta cantidad de ondas como querramos, a medida que aumentemos el orden del Hamiltoniano de interacción. Las derivadas de estos términos del Hamiltoniano son integrales de colisiones $I_{\vec{k}}^N$ de N ondas, como las de la Ecuación de Boltzmann.

Ahora bien, en general va a ser suficiente con llegar al primer término no nulo, tal que se satisfagan la conservación de lo que en cuántica serían el momento y la energía:

$$\vec{k}_1 \pm \vec{k}_2 \pm \cdots \pm \vec{k}_N = 0 \quad \omega(\vec{k}_1) \pm \omega(\vec{k}_2) \pm \cdots \pm \omega(\vec{k}_N) = 0 \quad (23)$$

Los signos corresponderán a si la "partícula" ingresa o sale.

Si no se satisface la segunda condición de resonancia entonces el término correspondiente del Hamiltoniano puede eliminarse mediante transformaciones canónicas apropiadas, como se verá más adelante [3].

Para el caso en que tenemos una relación de dispersión del tipo $\omega \sim k^\alpha$ la condición de resonancia de tres ondas no será posible en los casos en que $\alpha < 1$, con lo cual el primer término resonante será el de cuatro ondas, para el cual siempre hay solución en el caso de $\omega \sim k^\alpha$. La demostración se puede entender gráficamente, si dibujamos las superficies en 2D para $\omega(\vec{k}, \vec{k}_0)$ y $\omega(\vec{k}) + \omega(\vec{k}_0)$, donde \vec{k}_0 es un parámetro que puede variarse. Si las curvas se intersecan hay solución, sino solo existe la solución trivial en que ambas curvas son iguales, para el caso en que una de las ondas es 0. Esto está ilustrado en la Figura 3.

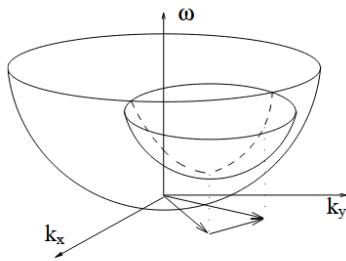


FIG. 1: Graphical solution for the three-wave resonant condition: solutions exist for $\alpha \geq 1$.

(a)

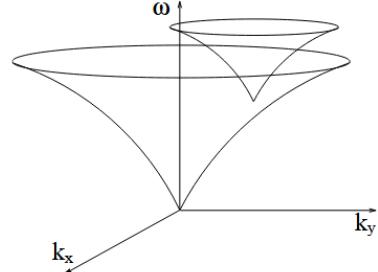


FIG. 2: Graphical solution for the three-wave resonant condition: solutions do not exist for $\alpha < 1$.

(b)

Figura 3: Soluciones gráficas para la condición resonante para dos casos distintos de la relación de dispersión $\omega \sim k^\alpha$.

Para las ondas de gravedad $\alpha = 1/2$ con lo cual las interacciones resonantes de 3 ondas no están permitidas (sino serían ondas con energía negativa). En el caso particular en que todas las ondas son colineales (reducimos el problema efectivamente a 1D, aunque la energía sigue distribuida en 2D) la mínima cantidad de ondas interactuantes resulta ser $N = 5$. [2]

4. Eliminación de los términos no resonantes.

Vamos a querer mostrar que efectivamente en el caso de que un término sea no resonante, o sea, que no cumple la condición (23), existirá una transformación canónica a nuevas variables $c_{\vec{k}}, c_{\vec{k}}^*$ en las cuales ese término no aparece en el Hamiltoniano de interacción.

A modo ilustrativo [3] plantea para una sola variable el Hamiltoniano hasta orden cuatro:

$$\mathcal{H} = \omega b b^* + \frac{V}{2}(b^2 b^* + b^{*2}) + \frac{U}{6}(b^3 + b^{*3}) + \frac{W}{4}(b^2 b^{*2}) + G(b^3 b^* + b b^{*3}) + R(b^4 + b^{*4}) \quad (24)$$

Y pide que la nueva variable sea de la forma (desarrollo perturbativo en b):

$$b = c + A_1 c^2 + A_2 c c^* + A_3 c^{*2} + B_1 c^3 + B_2 c^* c^2 + B_3 c c^{*2} + B_4 c^{*3} + \dots \quad (25)$$

Luego pide que $\{b, b^*\}_{c, c^*} = 1$ para que la transformación sea efectivamente canónica y resuelve para los coeficientes A_i y B_i pidiendo que todos los términos no lineales del Hamiltoniano excepto $c^2 c^{*2}$ sean 0. Entonces llega a que:

$$\mathcal{H} = \omega c c^* + \frac{1}{4} T c^2 c^{*2} \quad T = W - \frac{3V^2}{\omega} - \frac{U^2}{3\omega} \quad (26)$$

O sea, que siempre y cuando $\omega \neq 0$ se pueden eliminar los términos no resonantes del Hamiltoniano. Es posible hacer exactamente lo mismo para el caso en que no hay triadas resonantes, quedando:

$$\mathcal{H} = \frac{1}{4} \int T_{1234} c_1^* c_2^* c_3 c_4 \delta(\vec{k}_1 + \vec{k}_2 - \vec{k}_3 - \vec{k}_4) d\vec{k}_1 d\vec{k}_2 d\vec{k}_3 d\vec{k}_4 \quad (27)$$

Donde el coeficiente de interacción resulta:

$$T_{1234} = W_{1234} - \frac{U_{-1-212} U_{-3-434}}{\omega_3 + \omega_4 + \omega_{3+4}} + \frac{V_{1+212}^* V_{3+434}}{\omega_1 + \omega_2 - \omega_{1+2}} - \frac{V_{131-3}^* V_{424-2}}{\omega_{4-2} + \omega_2 - \omega_4} \\ - \frac{V_{242-4}^* V_{313-1}}{\omega_{3-1} + \omega_1 - \omega_3} - \frac{V_{232-3}^* V_{414-1}}{\omega_{4-1} + \omega_1 - \omega_4} - \frac{V_{141-4}^* V_{323-2}}{\omega_{3-2} + \omega_2 - \omega_3} \quad (28)$$

Notemos que si efectivamente la condición resonante para tres ondas no puede cumplirse (con una relación de dispersión de no decaimiento), entonces los denominadores de T_{1234} no divergen y efectivamente se puede eliminar \mathcal{H}_3 del Hamiltoniano de interacción \mathcal{H}_{int} .

En general la prohibición de interacciones del tipo $2 \rightarrow 1$ y $1 \rightarrow 2$ implican la prohibición de los términos $1 \rightarrow 3$ y $3 \rightarrow 1$. Igualmente se eliminan los términos $4 \rightarrow 0$ y $0 \rightarrow 4$ del Hamiltoniano, resultando en su versión más simplificada.

Notemos que al interactuar la mitad de las ondas de cada lado ($N/2 \rightarrow N/2$) se conserva la cantidad de partículas, asociado a la simetría $U(1)$ que mencionamos antes:

$$N = \int c_{\vec{k}}^* c_{\vec{k}} d\vec{k} \quad (29)$$

Que es lo que se conoce como la integral de *Wave action*.

Los distintos términos de T_{1234} se pueden interpretar gráficamente:

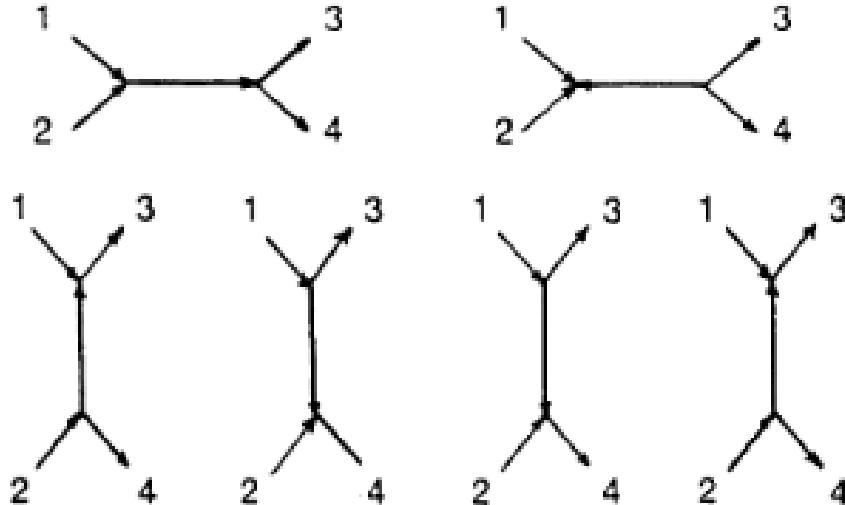


Figura 4: Distintos procesos de cuatro ondas como segundo orden perturbativo a proceso de tres ondas, donde la tercera es virtual y conecta ambos extremos.

Donde se pueden pensar como un segundo orden de perturbación a un proceso de tres ondas, donde aparece una fuerza virtual que no conserva energía y momento.

5. Correspondencia con fluidos con superficie libre.

Hasta ahora hemos dado un visión general del modelado de un sistema continuo no lineal mediante la teoría de perturbaciones, aplicable a una gran cantidad de sistemas físicos. En particular nos va a interesar mostrar que un fluido con superficie libre que se extiende infinitamente y con profundidad finita se puede describir de esta forma.

Vamos a pensar en un fluido irrotacional e incompresible, de esta forma $\vec{u} = \vec{\nabla}\phi$, con ϕ el potencial de velocidad. Éste satisface la Ecuación de Laplace.

$$\nabla^2\phi = 0 \quad -h < z < \eta \quad (30)$$

Donde $-h$ es la posición del fondo y $z = \eta(\vec{x})$ es la altura de la superficie libre. Vamos a tener además la condición de contorno en el fondo:

$$\partial_z\phi = 0 \quad z = -h \quad (31)$$

Y las condiciones de contorno cinemática (que sale por continuidad de la velocidad) y dinámica (que sale de Bernoulli por continuidad de presión) respectivamente:

$$\frac{D\eta}{Dt} = \partial_t\phi + \vec{\nabla}\phi \cdot \vec{\nabla}\eta = \partial_z\phi \quad \partial_t\phi + \frac{1}{2}|\vec{\nabla}\phi|^2 + g\eta = 0 \quad (32)$$

Que tendrá solución única dadas la altura de la superficie libre $\eta(\vec{x}, t)$ y el potencial en la superficie libre $\phi^s = \phi(\vec{x}, z = \eta(\vec{x}, t), t)$, que permiten obtener el potencial para todo z, t . [4]

Definimos:

$$w^s = \left(\frac{\partial\phi}{\partial x} \right)_\eta \quad (33)$$

Y transformamos todo a Fourier. La ecuación de Laplace queda:

$$\frac{\partial^2\phi_k}{\partial z^2} + k^2\phi_k = 0 \quad (34)$$

Que podemos resolver por separación de la variable z para que satisfaga la condición de contorno en el fondo:

$$\phi(\vec{k}, z, t) = \Phi(\vec{k}, t) \cosh(k(z + h)) \quad (35)$$

Luego podemos obtener $\phi^s(\vec{x}, t)$ al antitransformar. Por otro lado si derivamos y antittransformamos podemos obtener $w^s(\vec{x}, t)$.

Ahora debemos tomar la aproximación de que la pendiente de la ola (*steepness*) es muy baja, i.e. $k\eta \ll 1$. Entonces expandimos ϕ^s y w^s en términos del parámetro $k\eta$. Volvemos a transformar y expresamos η en términos de su transformada η_k para obtener ϕ_k^s y w_k^s . Por último invertimos las expresiones iterativamente para hallar Φ_k .

Finalmente podemos definir:

$$b_k = \left(\frac{g}{2\omega} \right)^{1/2} \eta_k + i \left(\frac{\omega}{2g} \right)^{1/2} \phi_k^s \quad (36)$$

Y queda una ecuación de movimiento para la evolución de b_k análoga a las que habíamos encontrado en el caso general. Para los *kernels* de interacción para procesos de tres y cuatro ondas se puede ver el Apéndice de [4].

6. Paso a la estadística y la Ecuación Cinética.

La idea ahora va a ser pasar de la descripción dinámica del sistema a una estadística en función de las funciones de correlación de las amplitudes de ondas.

Resumen Ecuaciones de Movimiento

En las variables apropiadas los Hamiltonianos sin términos no resonantes nos quedan, para el caso de tres ondas:

$$\mathcal{H}_3 = \frac{1}{2} \int (V_{123} c_1^* c_2 c_3 + \text{c.c.}) \delta(\vec{k}_1 - \vec{k}_2 - \vec{k}_3) d\vec{k}_1 d\vec{k}_2 d\vec{k}_3 \quad (37)$$

Con su ecuación de movimiento:

$$i \frac{\partial c_{\vec{k}}}{\partial t} - \omega_k c_{\vec{k}} = \int \left(\frac{1}{2} V_{k12} c_1 c_2 \delta(\vec{k} - \vec{k}_1 - \vec{k}_2) + V_{1k2}^* c_1 c_2^* \delta(\vec{k}_1 - \vec{k} - \vec{k}_2) \right) d\vec{k}_1 d\vec{k}_2 \quad (38)$$

Donde primero tenemos $k \rightarrow 1 + 2$ (que es una sola opción) y luego $k + 2 \rightarrow 1$ (que son dos opciones).

Y para el caso de cuatro ondas:

$$\mathcal{H} = \frac{1}{4} \int T_{1234} c_1^* c_2^* c_3 c_4 \delta(\vec{k}_1 + \vec{k}_2 - \vec{k}_3 - \vec{k}_4) d\vec{k}_1 d\vec{k}_2 d\vec{k}_3 d\vec{k}_4 \quad (39)$$

Con su ecuación de movimiento:

$$i \frac{\partial c_{\vec{k}}}{\partial t} - \omega_k c_{\vec{k}} = \frac{1}{2} \int T_{k123} c_1^* c_2 c_3 \delta(\vec{k} + \vec{k}_1 - \vec{k}_2 - \vec{k}_3) d\vec{k}_1 d\vec{k}_2 d\vec{k}_3 \quad (40)$$

Que son procesos $2 \rightarrow 2$.

Paso a la estadística.

Las ecuaciones de movimiento anteriores son para la evolución temporal de las amplitudes y fases de $c_{\vec{k}} = |c(\vec{k}, t)| e^{i\phi(\vec{k}, t)}$. En el caso en que las nolinealidades son débiles y hay una gran cantidad de modos esta evolución suele ser redundante, ya que incluye la dinámica rápida de $\phi \sim \omega t$ que deja a la evolución lenta de las amplitudes (que es constante en el caso puramente lineal, orden más bajo) virtualmente no afectada. Por esto vamos a querer pasar a una descripción estadística, donde solo trabajaremos con las funciones de correlación para $c_{\vec{k}}$, donde vamos a promediar en un ensamble.

Vamos a asumir una "caotización" de las fases, de forma tal que aunque al inicio estuvieran correlacionadas, luego de un tiempo ya no lo estarán. Podemos pensar que esto se debe a la dispersión del medio o a que inicialmente están en un equilibrio muy inestable del cual los saca cualquier perturbación del medio. De esta forma, bajo la aproximación de fases aleatorias tenemos las correlaciones:

$$\begin{aligned} \langle c_{\vec{k}} \rangle &= \langle |c_{\vec{k}}| e^{i\phi_{\vec{k}}} \rangle = 0 \\ \langle c_{\vec{k}} c_{\vec{k}'} \rangle &= \langle |c_{\vec{k}}| |c_{\vec{k}'}| e^{i(\phi_{\vec{k}} + \phi_{\vec{k}'})} \rangle = 0 \\ \langle c_{\vec{k}} c_{\vec{k}'}^* \rangle &= \langle |c_{\vec{k}}| |c_{\vec{k}'}^*| e^{i(\phi_{\vec{k}} - \phi_{\vec{k}'})} \rangle = n(\vec{k}) \delta(\vec{k} - \vec{k}') \end{aligned} \quad (41)$$

Acá definimos la acción de onda $n(\vec{k}) = c_{\vec{k}} c_{\vec{k}}^*$, que sería la densidad de "partículas" número de onda \vec{k} en la integral (29).

Por último para los correladores de una cantidad impar siempre van a dar 0, y para los de orden 4 y 6 tenemos:

$$\begin{aligned} \langle c_1^* c_2^* c_3 c_4 \rangle &= n_1 n_2 [\delta_{1-3} \delta_{2-4} + \delta_{1-4} \delta_{2-3}] \\ \langle c_1^* c_2^* c_3^* c_4 c_5 c_6 \rangle &= n_1 n_2 n_3 [\delta_{1-4} \delta_{2-5} \delta_{3-6} + \delta_{1-4} \delta_{2-6} \delta_{3-5} + \delta_{1-5} \delta_{2-4} \delta_{3-6} \\ &\quad + \delta_{1-5} \delta_{2-6} \delta_{3-4} + \delta_{1-6} \delta_{2-4} \delta_{3-5} + \delta_{1-6} \delta_{2-5} \delta_{3-6}] \end{aligned} \quad (42)$$

Que básicamente las deltas dan que queden dos o tres n 's. Acá es donde producimos la clausura de la jerarquía, al escribir el cuarto momento como un producto de dos segundos momentos (o tres para el sexto). En parte esta aproximación se puede hacer porque las estadísticas son cercanas a la Gaussiana para turbulencia de ondas débil que es esto. [1]

Clausura Gaussiana.

Ya hemos estado asumiendo que el comportamiento estadístico de la turbulencia de ondas débil es cercano al Gaussiano en el tiempo largo, siendo insensible a las condiciones iniciales. [5]

Esta propiedad, a la cual nos referimos como clausura Gaussiana ya que nos permite expresar momentos superiores en función del segundo momento y escribir de esta forma una expresión cerrada para la ecuación cinética, viene de una de una aplicación del Teorema Central del Límite. [6]

Ecuación cinética de tres ondas.

Para obtener la ecuación de evolución para $n_{\vec{k}}$ hay que multiplicar (38) por $c_{\vec{k}}^*$ y restarle la conjugada por $c_{\vec{k}'}$, para finalmente tomar el promedio y reemplazar $\vec{k} = \vec{k}'$:

$$\frac{\partial n_{\vec{k}}}{\partial t} = \text{Im} \int [V_{k12}\langle c_k^* c_1 c_2 \rangle \delta(\vec{k} - \vec{k}_1 - \vec{k}_2) - 2V_{1k2}\langle c_1^* c_k c_2 \rangle \delta(\vec{k}_1 - \vec{k} - \vec{k}_2)] d\vec{k}_1 d\vec{k}_2 \quad (43)$$

Ahora bien $\langle c_1^+ c_2 c_3 \rangle = 0$ a orden 0. Debemos entonces ir a un orden más en la teoría perturbativa para calcular el correlador de tres puntos. Esto lo hacemos usando la ecuación (38) por otras dos amplitudes y sumando:

$$\left[i \frac{\partial}{\partial t} + (\omega_1 - \omega_2 - \omega_3) \right] \langle c_1^* c_2 c_3 \rangle = \int \left[-\frac{1}{2} V_{145}^* \langle c_4^* c_5 c_2 c_3 \rangle \delta(\vec{k}_1 - \vec{k}_4 - \vec{k}_5) + V_{425}^* \langle c_1^* c_5 c_3 c_4 \rangle \delta(\vec{k}_4 - \vec{k}_2 - \vec{k}_5) + V_{435}^* \langle c_1^* c_5 c_2 c_4 \rangle \delta(\vec{k}_4 - \vec{k}_3 - \vec{k}_5) \right] d\vec{k}_4 d\vec{k}_5 \quad (44)$$

Ahora usamos la expresión a primer orden del correlador de cuatro puntos de antes y llegamos a que:

$$\left[i \frac{\partial}{\partial t} + (\omega_1 - \omega_2 - \omega_3) \right] \langle c_1^* c_2 c_3 \rangle = V_{123}^* [n_1 n_3 + n_1 n_2 - n_2 n_3] \quad (45)$$

Al segundo orden de perturbaciones esta ecuación pedimos que no dependa del tiempo. Integrando nos queda un término constante $A_{123}/\Delta\omega$ y otro que oscila rápidamente a frecuencia $\Delta\omega t$, que puede ser despreciada al integrar el correlador de tres puntos. De esta forma nos queda esencialmente que:

$$\langle c_1^* c_2 c_3 \rangle = \frac{V_{123}^*(n_1 n_2 + n_1 n_3 - n_2 n_3)}{\omega_1 - \omega_2 - \omega_3 + i\delta} \quad (46)$$

Se coloca $+i\delta$ para evitar el polo, que vamos a atravesar ya que se da en la condición resonante. Hacemos tender $\delta \rightarrow 0$ y usamos que $\text{Im}(1/(\omega + i\delta)) \rightarrow -\pi\delta(\omega)$, que es medio residuo al integrar en el eje real y el signo menos sale por la orientación de la curva.⁴

Reemplazando este resultado en la ecuación cinética obtenemos⁵:

$$\begin{aligned} \frac{\partial n_{\vec{k}}}{\partial t} &= \pi \int [|V_{k12}|^2 f_{k12} \delta(\vec{k} - \vec{k}_1 - \vec{k}_2) \delta(\omega - \omega_1 - \omega_2) - 2|V_{1k2}|^2 f_{1k2} \delta(\vec{k}_1 - \vec{k} - \vec{k}_2) \delta(\omega_1 - \omega - \omega_2)] \\ f_{123} &= n_1 n_2 n_3 \left(\frac{1}{n_1} - \frac{1}{n_2} - \frac{1}{n_3} \right) \end{aligned} \quad (47)$$

⁴Queremos la integral de izquierda a derecha en los reales, cerramos por abajo con un semicírculo de forma que encerramos el polo en $\omega = -i\delta$, cuya integral en el infinito da 0. Luego equivale la integral en reales al medio residuo $-2\pi i/2$ y finalmente tomamos la parte imaginaria.

⁵En [3] p. 67 el signo es +, pero por [2] y lo que sigue creo es -.

Ecuación cinética de cuatro ondas.

Seguimos un procedimiento análogo al anterior, nuevamente va a dar a primer orden 0, con lo cual hay que ir a un orden más alto y ahora usar el correlador de seis puntos. Finalmente queda:

$$\frac{\partial n_{\vec{k}}}{\partial t} = \frac{\pi}{2} \int |T_{k123}|^2 f_{k123} \delta(\vec{k} + \vec{k}_1 - \vec{k}_2 - \vec{k}_3) \delta(\omega + \omega_1 - \omega_2 - \omega_3) d\vec{k}_1 d\vec{k}_2 d\vec{k}_3$$

$$f_{1234} = n_1 n_2 n_3 n_4 \left(\frac{1}{n_1} + \frac{1}{n_2} - \frac{1}{n_3} - \frac{1}{n_4} \right) \quad (48)$$

7. Modificaciones a la Ecuación Cinética.

Se pueden agregar términos de disipación y forzado en la ecuación cinética, mediante una modificación de la ecuación de Hamilton [1, 5]:

$$\frac{\partial a_k}{\partial t} = -i \frac{\delta \mathcal{H}}{\delta a_k^*} + f_k(t) - \gamma_k a_k \quad (49)$$

Donde el primer término dará cuenta de la propagación e interacción de ondas, el segundo del forzado y el tercero de la disipación. Si asumimos que tenemos una fuerza aleatoria, que sea homogénea e isotropa en el espacio y blanca en el tiempo entonces,

$$\langle f_k(t) f_{k'}(t') \rangle = F(k) \delta(\vec{k} + \vec{k}') \delta(t' - t) \quad (50)$$

Con $F(k) \neq 0$ en el entorno de k_f donde se produce la inyección de energía. Además asumimos $\gamma_k \ll \omega_k$. El tiempo de disipación se puede estimar como $\tau_{diss} \sim 1/\gamma_k$, que será importante cuando sea del orden de los tiempos no lineales $\tau_{nl} \sim \tau_{diss}$. Para el caso de superficie libre de un fluido tenemos $\gamma_k = 2\nu k^2$. [7]

La ecuación cinética queda:

$$\frac{\partial n_k}{\partial t} = F_k - \gamma_k n_k + I_k^3 \quad (51)$$

Donde I_k^3 es la integral de colisiones para tres ondas en este caso, pero podría ser I_k^4 para otro sistema. En [5] la escriben con una notación alternativa:

$$\frac{\partial n_{\vec{k}}}{\partial t} = C_{\vec{k}} - D_{\vec{k}} + I_{\vec{k}} \quad (52)$$

Donde $C_{\vec{k}}$ es la integral de interacciones entre ondas, $D_{\vec{k}}$ el *damping* y $I_{\vec{k}}$ la *inyección* de energía.

El caso en que $D_k = I_k = 0$ corresponde con el equilibrio termodinámico, o la solución de Rayleigh-Jeans, y está asociada a una equipartición de energía entre todos los modos.

Propiedades de la Ecuación Cinética.

Cantidades conservadas

Como ya se mencionó varias veces el sistema dinámico preserva ciertas cantidades, en particular la energía E y momento \vec{H} en el caso general, y para el caso de interacciones del tipo $2 \rightarrow 2$ también N .

Podemos ver que estas conservaciones se mantienen en la descripción estadística, a partir de la Ecuación Cinética. En el caso de la energía y momento es intuitivo ya que tenemos las deltas que preservan estas cantidades durante las interacciones. Pensemos como [2] en el caso de una magnitud arbitraria Φ cuya densidad espectral es $\phi(\vec{k}) = \rho(\vec{k})n(\vec{k})$, entonces su evolución temporal será:

$$\frac{d\Phi}{dt} = \dot{\Phi} = \frac{d}{dt} \int \phi(\vec{k}) d\vec{k} = \int \rho(\vec{k}) \frac{\partial n(\vec{k})}{\partial t} d\vec{k} \quad (53)$$

Ahora podemos reemplazar la derivada temporal de $n_{\vec{k}}$ por la ecuación cinética. Veamos en particular el caso de tres ondas, el de cuatro es análogo.

$$\frac{\partial n_{\vec{k}}}{\partial t} = \int (\mathcal{R}_{k12} - \mathcal{R}_{2k1} - \mathcal{R}_{12k}) d\vec{k}_1 d\vec{k}_2 \quad (54)$$

$$\mathcal{R}_{123} = \pi |V_{123}|^2 n_1 n_2 n_3 \left(\frac{1}{n_1} - \frac{1}{n_2} - \frac{1}{n_3} \right) \delta(\omega_1 - \omega_2 - \omega_3) \delta(\vec{k}_1 - \vec{k}_2 - \vec{k}_3) \quad (55)$$

Entonces resulta,

$$\dot{\Phi} = \int \rho_{\vec{k}} (\mathcal{R}_{k12} - \mathcal{R}_{2k1} - \mathcal{R}_{12k}) d\vec{k} d\vec{k}_1 d\vec{k}_2 = \int (\rho_k - \rho_1 - \rho_2) \mathcal{R}_{k12} d\vec{k} d\vec{k}_1 d\vec{k}_2 \quad (56)$$

Donde en el último igual solo se renombraron las variables de integración cíclicamente (esto se puede hacer ya que estamos integrando en las tres, con lo cual se vuelven mudas).

Podemos ver entonces que si $\rho_k - \rho_1 - \rho_2 = 0$ ocurre en simultáneo con las condiciones de resonancia, entonces la cantidad Φ se conserva, o sea, $\dot{\Phi} = 0$. Para la energía basta con tomar $\rho_{\vec{k}} = \omega(\vec{k})$, y para la componente i -ésima del momento basta con tomar $\rho(\vec{k}) \vec{k}_i$.

En el caso de cuatro ondas por un método análogo nos va a quedar la condición $\rho_k + \rho_1 = \rho_2 + \rho_3$ y podemos obtener la conservación de la acción de onda tomando simplemente $\rho(\vec{k}) = 1$.

Una cosa importante a mencionar es que en el caso no degenerado las cantidades E , N y Π forman un conjunto completo de integrales de movimiento. Pueden aparecer otras constante sólo en casos degenerados [3] donde se satisfaga la condición $\rho_k = \rho_1 + \rho_2$ para densidades tales que $\rho(\vec{k}) \neq A\omega(\vec{k}) + \vec{B} \cdot \vec{k}$. De ser el caso decimos que $\omega(\vec{k})$ es degenerada.

Un ejemplo de esto es el caso de aguas poco profundas en el modelo de Kadomtsev-Petviashvili, donde se pueden encontrar infinitas integrales de movimiento, propiedad heredada de la ya establecida integrabilidad de la propia ecuación.

Balance de energía.

Ya establecimos que la energía total se conserva, o sea:

$$\frac{dE}{dt} = \int \frac{\partial E_{\vec{k}}}{\partial t} d\vec{k} = \int \omega \frac{\partial n_{\vec{k}}}{\partial t} d\vec{k} \quad (57)$$

Ahora podemos usar nuevamente la ecuación cinética nuevamente, ahora escrita de la forma: $\partial_t n_{\vec{k}} = I_{\vec{k}}^N$, siendo $I_{\vec{k}}^N$ las integrales de colisiones. Esto va a ser cierto en el rango inercial (siempre que la escala de inyección y disipación estén lo suficientemente separadas). Entonces podemos escribir una ecuación de continuidad para la energía

$$\frac{\partial E(\vec{k})}{\partial t} + \vec{\nabla}_{\vec{k}} \cdot \vec{\varepsilon} = 0 \quad \vec{\nabla}_{\vec{k}} \cdot \vec{\varepsilon} \equiv -\omega(\vec{k}) I_{\vec{k}}^N \quad (58)$$

Donde definimos el flujo de energía $\vec{\varepsilon}$. O su versión 1D con la normalización de cilíndricas:

$$\frac{\partial E(k)}{\partial t} + \frac{\partial \varepsilon}{\partial k} = 0 \quad (59)$$

Notamos que ecuaciones estacionarias de la ecuación cinética ($\partial_t n_k = 0$) implican que las integrales de colisiones son 0, y se corresponden con un flujo constante de energía ε entre escalas. Para la relación de dispersión $\omega \sim k^{\alpha}$ son los espectros de KZF, análogos a la Ley de -5/3 (c.f. (2)) de Kolmogorov para turbulencia normal.

Lo mismo se puede hacer para el momento:

$$\frac{\partial \vec{\pi}(\vec{k})}{\partial t} + \vec{\nabla}_{\vec{k}} \cdot \mathbf{R} = 0 \quad \vec{\nabla}_{\vec{k}} \cdot \mathbf{R} \equiv -\vec{k} I_{\vec{k}}^N \quad (60)$$

Donde \mathbf{R} es un tensor de rango 2.

Para el caso en que se conserva la integral de acción de onda $N = \int n_k dk$ vamos a definir el flujo de *wave action* como:

$$\frac{\partial n(\vec{k})}{\partial t} + \vec{\nabla}_{\vec{k}} \cdot \vec{\zeta} = 0 \quad \vec{\nabla}_{\vec{k}} \cdot \vec{\zeta} \equiv - I_{\vec{k}}^N \quad (61)$$

O su versión en 1D:

$$\frac{\partial n_k}{\partial t} + \frac{\partial \zeta}{\partial k} = 0 \quad (62)$$

Cuando hay disipación.

Cuando agregamos el término disipativo en la ecuación cinética $+\Gamma(\vec{k})n_{\vec{k}}$ entonces en el caso estacionario nos va a quedar:

$$I^N(\vec{k}) + \Gamma(\vec{k})n(\vec{k}) = 0 \quad (63)$$

Esto también modela la inyección de ondas para $\Gamma(\vec{k}) > 0$.

Sabemos por lo que mostramos antes que vale por las deltas de la integral de colisiones:

$$\int I^N(\vec{k}) \omega d\vec{k} = \int I^N(\vec{k}) \vec{k} d\vec{k} = 0 \quad (64)$$

Entonces por la conservación de la energía y el momento tenemos, de multiplicar por ω y \vec{k} la ecuación cinética estacionaria e integrar obtenemos:

$$\int \Gamma(\vec{k}) \omega n(\vec{k}) d\vec{k} = 0 \quad \int \Gamma(\vec{k}) \vec{k} n(\vec{k}) d\vec{k} = 0 \quad (65)$$

Finalmente si integramos la ecuación de continuidad la parte de la integral de colisiones va a dar 0, con lo cual efectivamente nos va a quedar que la variación del flujo es:

$$\frac{d\varepsilon}{dk} = \Gamma(\vec{k}) E(\vec{k}) \quad (66)$$

Y si vemos lo que pasa en una esfera se puede mostrar que el flujo de energía por la superficie se corresponde con el disipado y el inyectado. [1]

8. Espectro de energía y PSD.

Nos interesará estudiar la transferencia de energía entre escalas experimentalmente, que estará caracterizada por la densidad espectral de energía E_k tal que la magnitud $E = \int E_k dk$ se conserva.

Para esto vamos a empezar dando las definiciones básicas de lo que estaremos usando.

Como mencioné anteriormente, la energía total va a venir dada por una distribución, que puede escribirse ya sea en el espacio de k o de ω , vinculados ambos a través de la relación de dispersión lineal $\omega(k)$. Entonces:

$$E = \int E(\vec{k}) d\vec{k} = \int k E(\vec{k}) dk d\theta \equiv \int E(k) dk = \int E(\omega) d\omega \quad (67)$$

O sea, estamos definiendo $E(k) = 2\pi E(\vec{k})$ y la relación entre las densidades en espacio de frecuencia y momento está dada por: $E(k)dk = E(\omega)d\omega$.

Esta energía será la de la superficie libre. En principio sabemos que se dará un balance entre la energía cinética y la potencial, como una especie de equipartición [8]. Se suele calcular la potencial (a partir de las mediciones para la superficie libre η) y se multiplica por 2 [7]. Vamos a entonces trabajar con las energías potenciales, cuya expresión depende de si tenemos ondas de gravedad o capilares. Tenemos que:

$$E_p^g = \frac{1}{2} \rho g \int \eta^2 dS \quad E_p^s = \gamma \int \sqrt{1 + (\nabla \eta)^2} - 1 dS \quad (68)$$

Acá la energía por capilaridad es básicamente la integral del elemento de superficie ds . Se puede aproximar a primer orden para una onda armónica en el caso lineal (con $\eta \ll 1$), entonces [7]

$$E_p^s = \frac{1}{2} \gamma \int k^2 \eta^2 dS \quad (69)$$

Usando que $(1 + \varepsilon)^n \approx 1 + n\varepsilon$. Ahora vamos a querer pasar al espacio de Fourier, transformando las energías:

$$E_p^g = \frac{\rho g}{2} \int |\eta_{\vec{k}}|^2 d\vec{k} \quad E_p^s = \frac{\gamma}{2} \int k^2 |\eta_{\vec{k}}|^2 d\vec{k} \quad (70)$$

Donde $\eta_{\vec{k}}$ es la transformada de Fourier de η :

$$\eta(\vec{k}) = \frac{1}{2\pi} \int \eta(\vec{r}) e^{-i\vec{k}\cdot\vec{r}} d\vec{r} \quad (71)$$

Podemos integrar en cilíndricas para definir $\eta_k = 2\pi k \eta_{\vec{k}}$. Las densidades espectrales de energía van a ser los integrandos.

Por último vamos a definir la Power Spectral Density (PSD) que va a ser lo más sencillo para trabajar experimentalmente:

$$S_\eta(\omega) = \frac{1}{T} |\eta_\omega|^2 \quad S_\eta(k) = \frac{1}{L^2} |\eta_k|^2 \quad (72)$$

Acá ahora $\eta(\omega) = \frac{1}{\sqrt{2\pi}} \int \eta(t) e^{i\omega t} dt$ es la transformada de Fourier respecto al tiempo. Además para obtener ya sea $\eta(t)$ o $\eta(\vec{r})$ se puede promediar en el tiempo o espacio, con lo cual en realidad sería $S_\eta \propto \langle |\eta|^2 \rangle$ (y técnicamente el promedio sería en el ensamble, pero consideramos al sistema ergódico). [9]

Podemos relacionar las PSD en ambos espacios como $S(\omega)d\omega = S(k)dk$. De esta forma entonces podemos obtener la densidad de energía a partir de la PSD como:

$$E^g(k) = \frac{\rho g}{2} S_\eta(k) \quad E^s(k) = \frac{\gamma}{2} k^2 S_\eta(k) \quad (73)$$

Una cosa interesante es que si escribimos explícitamente $S_\eta(k) \propto k |\eta_{\vec{k}}|^2$ entonces nos queda en ambos casos:

$$E(k) \propto \omega^2 |\eta_{\vec{k}}|^2 \propto |\dot{\eta}_{\vec{k}}|^2 \quad (74)$$

Que sería la energía cinética efectivamente.

9. Espectro de Kolmogorov-Zakharov-Filonenko.

Experimentalmente es posible observar que el espectro de energía para la elevación de la superficie libre en un determinado rango de números de ondas (de equilibrio) resulta $E_\omega \sim A\omega^s$. En este rango la mayor contribución a la dinámica se debe a los efectos no lineales, ya que los viscosos son menores. Es posible calcular los exponentes de forma analítica a partir de las ecuaciones para la superficie libre [10]

La densidad de energía se puede escribir como:

$$E(\vec{k}) = \omega(\vec{k}) n(\vec{k}) \quad (75)$$

Donde n es la *wave action*, análoga a la cantidad de partículas (o ondas) con ese número de onda.

Ondas de gravedad [10]

Tenemos que $E(\omega) = \omega^4 n(\omega)$ y pueden calcular que para $n(\omega) = A\omega^s$ hay soluciones $s = -1$, que se corresponde con la distribución de Rayleigh-Jeans (equilibrio termodinámico para el caso lineal, con PDF Gaussiana [2]), y $s = -8$ que es la análoga al espectro de Kolmogorov para turbulencia hidrodinámica.

Las equivalencias son [9]:

$$E_k^g \sim P^{1/3} g^{1/2} k^{-5/2} \quad S_k^g \sim P^{1/3} g^{-1/2} k^{-5/2} \quad S_\omega^g \sim P^{1/3} g \omega^{-4} \quad (76)$$

Es posible llegar a estos resultados de forma dimensional, pero para que el exponente sea único hay que asumir a priori que el flujo en el espectro de energía va en el orden $P^{1/(N-1)}$ con N el número de ondas interactuantes más bajo, siendo $N = 4$ para ondas de gravedad.

Para hacer los cambios hay que usar la relación de dispersión:

$$\omega^2 = gk \quad (77)$$

Ondas de capilaridad [11]

Al igual que para las ondas de gravedad calculan $n(\omega)$, llegando esta vez a los resultados de $s = -1$ que es la de Rayleigh-Jeans y $s = -17/6$ que es la que nos interesa. Con esto:

$$E_k^s \sim P^{1/2} \left(\frac{\gamma}{\rho} \right)^{1/4} k^{-7/4} \quad S_k^s \sim P^{1/2} \left(\frac{\gamma}{\rho} \right)^{-3/4} k^{-15/4} \quad S_\omega^s \sim P^{1/2} \left(\frac{\gamma}{\rho} \right)^{1/6} \omega^{-17/6} \quad (78)$$

Acá la relación de dispersión es:

$$\omega^2 = \frac{\gamma}{\rho} k^3 \quad (79)$$

Deducción dimensional.

Para que sea única dijimos que hay que asumir cómo escala el espectro de energía respecto a la cantidad de ondas interactuantes. Esto surge del balance de energía (58):

$$\varepsilon \sim \frac{\partial E_k}{\partial t} dk \sim \omega \frac{\partial n_k}{\partial t} dk \sim I_k^N \sim n_k^{N-1} \quad (80)$$

Por la ecuación cinética. Por lo tanto: [7]

$$E_k \sim n_k \sim \varepsilon^{1/(N-1)} \quad (81)$$

10. Doble cascada para N par

A partir de los resultados para los espectros KZF se puede ver que existe una cascada directa de energía, sin embargo, para el caso en que las interacciones se dan entre un número par de ondas, tenemos otro invariante además de la energía, que es la *wave action*, que tendrá una cascada inversa. [2]

Para el caso de ondas de gravedad, con interacciones de 4 ondas, tenemos el espectro:

$$E_k^g \sim g^{2/3} \zeta^{1/3} k^{-7/3} \quad (82)$$

Con ζ el flujo de *wave action* (ratio de disipación)

11. Turbulencia de Ondas Discreta

Ahora van a estar discretizadas las frecuencias posibles del recinto, de forma tal que entre un número entero de longitudes de onda en el mismo. Por lo tanto la cantidad de interacciones posibles entre ondas se va a ver limitada.

Quasi-resonancias.

Clusters de interacción.

Modelo de pila de arena.

1. El sensor capacitivo de la altura del agua.

Empezaremos por un lado diseñando un sensor capacitivo que nos permita medir los desplazamientos de la superficie libre respecto del valor en equilibrio con una alta resolución temporal y el mínimo error posible. Para esto nos basamos en el modelo propuesto por [12].

Este sensor consiste en un cable de cobre recubierto por un esmalte y rodeado por una determinada cantidad de líquido. El esmalte actúa como medio dieléctrico entre dos conductores de forma tal que el sistema completo actúa a modo de capacitor.

Se puede pensar al cable como un conjunto de segmentos de longitud dl , cada uno de los cuales agrega al circuito una inductancia en serie $L'dl$ y una capacitancia en paralelo $C'dl$, incluyéndose si está sumergido ese segmento. Aquí:

$$C' = 2\pi\varepsilon \ln^{-1} \left(\frac{r_2}{r_1} \right) \quad L' = \frac{\mu}{2\pi} \ln \left(\frac{r_2}{r_1} \right) \quad (83)$$

Son la capacitancia e inductancia por unidad de longitud de un capacitor cilíndrico perfecto.

La inductancia en función de la longitud l sumergida puede calcularse de forma recursiva por lo dicho anteriormente, de forma que $Z(l + dl) = j\omega L'dl + [Z(l)^{-1} + (j\omega C'dl)^{-1}]^{-1}$, con la condición de que $Z(0) = \infty$ ya que el circuito está desconectado.

Esto se puede resolver analíticamente, resultando en que:

$$Z(l) = j\sqrt{\frac{L'}{C'}} \tan \left(\omega\sqrt{L'C'}l - \pi/2 \right) \quad (84)$$

Si suponemos que l es lo suficientemente chica resulta que:

$$Z(l) \approx \frac{1}{j\omega C'l} + \mathcal{O} \left(\frac{1}{L'C'^2\omega^2 l^3} \right) \quad (85)$$

Con lo cual el circuito se comporta principalmente de forma capacitiva. Además, como $r_2 = r_1 + e$ con e el espesor del esmalte, entonces $C' \propto r_1/e$.

2. Simulaciones del sensor y funcionamiento.

El circuito

Para poder medir la altura del agua l vamos a usar un circuito resonante RLC, con el cable de cobre conectado en paralelo a otro capacitor, sujeto a una corriente alterna de frecuencia ω . De esta forma tendremos la siguiente ecuación para el circuito:

$$V \cos(\omega t) = \frac{d^2I}{dt^2} + \frac{R}{L} \frac{dI}{dt} + \frac{1}{C(t)L} I \quad (86)$$

Con $C(t) = C_0 + \Delta C(t)$, tal que $\Delta C(t)/C_0 \ll 1$. Vale aclarar que C_0 es la suma de la capacitancia extra con la capacitancia del cable para el estado de reposo.

Es posible mostrar que siempre y cuando el cociente sea lo suficientemente pequeño, la fase de la corriente será:

$$\tan(\phi) = -\frac{1}{R} \left[\omega L - \frac{1}{\omega C_0} \right] - \frac{1}{R} \frac{\Delta C}{\omega C_0^2} + \mathcal{O} \left[\left(\frac{\Delta C}{C_0} \right)^2 \right] \quad (87)$$

Y si elegimos la frecuencia resonante del circuito $\omega = 1/\sqrt{LC_0}$ se puede traducir en que:

$$\phi = -Q_F \frac{\Delta C}{C_0} = -Q_F \frac{C'}{C_0} l(t) \quad Q_F = \frac{1}{R} \sqrt{\frac{L}{C_0}} \quad (88)$$

Con lo cual la fase es proporcional a la altura de la superficie libre (respecto de la altura de equilibrio).

Extracción de la fase

Para obtener la fase del circuito se puede medir la diferencia de potencial sobre la resistencia (proporcional a la corriente por la Ley de Ohm) y pasarlala por un lock-in. El lock-in nos devolverá la amplitud y fase en función del tiempo respecto de una señal de referencia, que en este caso será la fuente de alterna. Para las simulaciones del circuito se emuló su funcionamiento en *Python* de forma digital:

```
from scipy.signal import butter, filtfilt

def lock_in(signal, omega, fs, cutoff_freq=1):
    ts = np.arange(len(signal)) / fs
    ref_cos = np.cos(omega * ts)
    ref_sin = np.sin(omega * ts)

    in_phase = signal * ref_sin
    quadrature = signal * ref_cos

    b, a = butter_lowpass(cutoff_freq, fs)
    in_phase_filtered = filtfilt(b, a, in_phase)
    quadrature_filtered = filtfilt(b, a, quadrature)

    A_t = np.sqrt(in_phase_filtered**2 + quadrature_filtered**2)
    phi_t_extracted = np.arctan2(quadrature_filtered, in_phase_filtered)

    return A_t, phi_t_extracted

def butter_lowpass(cutoff, fs, order=4):
    nyquist = 0.5 * fs
    normal_cutoff = cutoff / nyquist
    b, a = butter(order, normal_cutoff, btype='low', analog=False)
    return b, a
```

A modo sumario, el funcionamiento radica en multiplicar (o mixear) la señal de interés (que debe estar montada sobre la moduladora) por la señal de referencia y la misma con una fase de $\pi/2$, luego se pasa por un filtro pasa bajos y se reconstruye la amplitud y la fase. En la práctica, el circuito pasa bajos actúa como integrador de forma que al multiplicar por la señal de interés aplicará la ortogonalidad de las funciones sinusoidales con distinta frecuencia, quedando solo la componente con la deseada (también llamada homodina, a diferencia de las no deseadas, que se llaman heterodinas). Al mismo tiempo, al llevarse la señal a frecuencias más altas en el espacio de Fourier (por convolucionar con la señal de referencia) disminuirá el ruido.

Un aspecto clave para el correcto funcionamiento será elegir de forma adecuada la frecuencia de corte del filtro pasabajos, de forma tal que integre por suficiente tiempo como para deshacerse del ruido pero no tanto que se pierda resolución en la señal. Es por esto que debe estar entre la frecuencia moduladora ω_r y la de la señal ω_s , que además debe ser lo suficientemente lenta para que entren varios ciclos de la moduladora en la misma:

$$\omega_{\text{sampleo}} > \omega_r > \omega_{\text{cut-off}} > \omega_s \quad (89)$$

La implementación en código se puso a prueba de la siguiente forma:

```
omega_ref = 100
omega_sig = 1
A = 1

def phi(t):
```

```

    return A * np.sin(omega_sig * t)

ts = np.linspace(0, 10, 10000)
signal = np.sin(omega_ref * ts + phi(ts))

extracted_A, extracted_phi = lock_in(signal, omega_ref, 1/(ts[1]-ts[0]))

```

Y efectivamente funciona de la manera deseada.

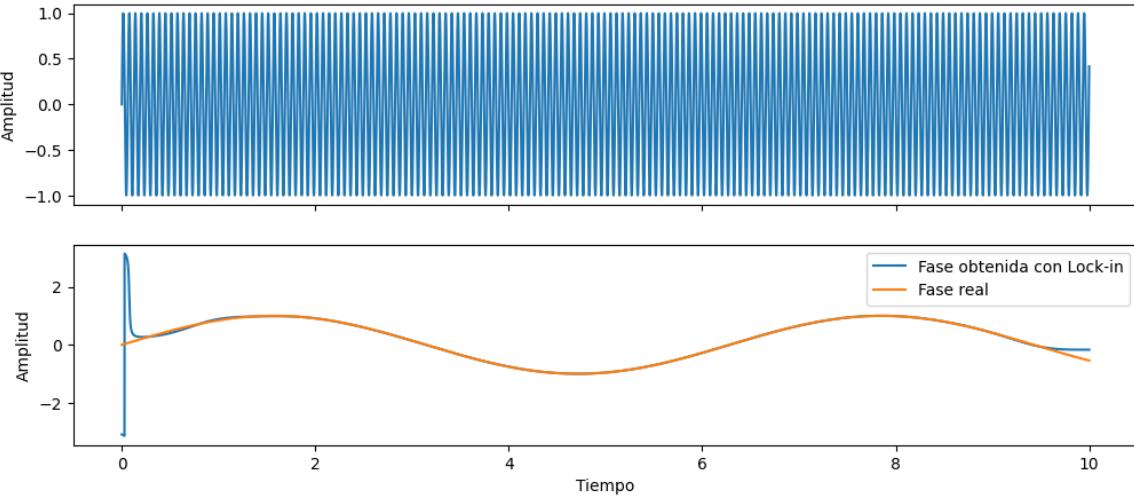


Figura 5: Comparación de fase extraída con lock-in y original para una modulación sinusoidal de la fase.

La simulación

Para estudiar el comportamiento del circuito se simuló mediante *Python*, resolviendo la ecuación diferencial con *Odeint* de *Scipy.integrate* para distintos parámetros y $l(t)$. El código final que toma la lista de parámetros y la función de alturas en función del tiempo es:

```

from scipy.integrate import odeint

def measured_l(ts, l, C0, C, R, L, V=5, cutoff_freq=1):
    omega = 1 / np.sqrt(C0 * L)
    Q = (1 / R) * np.sqrt(L / C0)

    def C_(t):
        return C0 + C * l(t)

    def dXdt(X, t):
        i, didt = X
        didt2 = V * np.cos(omega * t) - R / L * didt - 1 / (C_(t) * L) * i
        return [didt, didt2]

    y0 = [0, 0]
    sol = odeint(dXdt, y0, ts)
    current = sol[:, 0]

    _, phi_t_extracted = lock_in(current, omega, fs = 1 / (ts[1] - ts[0]),
                                   cutoff_freq=cutoff_freq)

    unwrapped = np.unwrap(phi_t_extracted)
    reconstructed_heights = -C0 * unwrapped / (Q * C)

    return reconstructed_heights

```

Devolviendo al final la lista de las alturas reconstruidas. A continuación algunas de las pruebas realizadas.

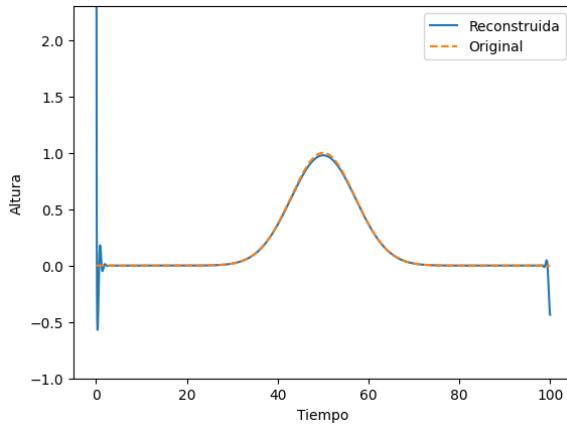


Figura 6: Reconstrucción de alturas para una $l(t)$ Gaussiana.

En primer lugar para una $l(t)$ Gaussiana se reconstruye muy fielmente la altura original con el método propuesto. Al inicio los artefactos son por las condiciones iniciales transitorias del sistema, hasta que alcanza un estacionario.

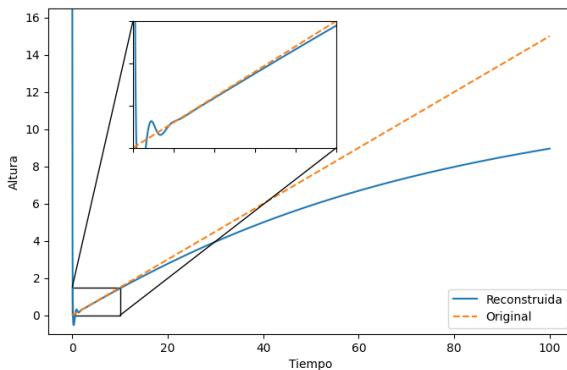


Figura 7: Barrido lineal de alturas.

Luego con una lineal para estudiar sus límites, podemos ver que para l 's bajos se comporta linealmente, pero luego satura. Esto será cuando $C_0 \sim C'l$. Con lo cual si hacemos más pequeño C' mejorará el rango de funcionamiento, aunque disminuirá la sensitividad, ya que es proporcional al error en fase del lock-in (que está fijo) por C' .

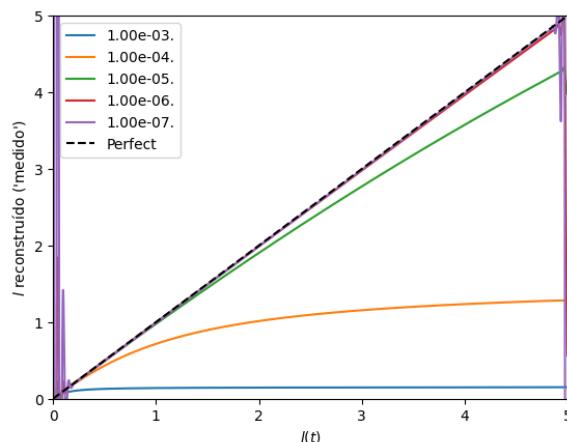


Figura 8: Comparación para el mismo barrido con distintos C' .

Valores reales

Siguiendo los valores usados por [12] se propone usar $R \sim 1 \text{ k}\Omega$, $L = 22 \text{ mH}$, $C_0 = 250 \text{ pF}$ y $C' \sim 10 \text{ pF/mm}$. Estos parámetros dan una frecuencia $\omega \sim 10^5 \text{ Hz}$ o $f \approx 60 \text{ kHz}$. Analizamos el comportamiento para una señal $l(t)$ de frecuencia $\omega = 1 \text{ kHz}$ y amplitud 1 mm.

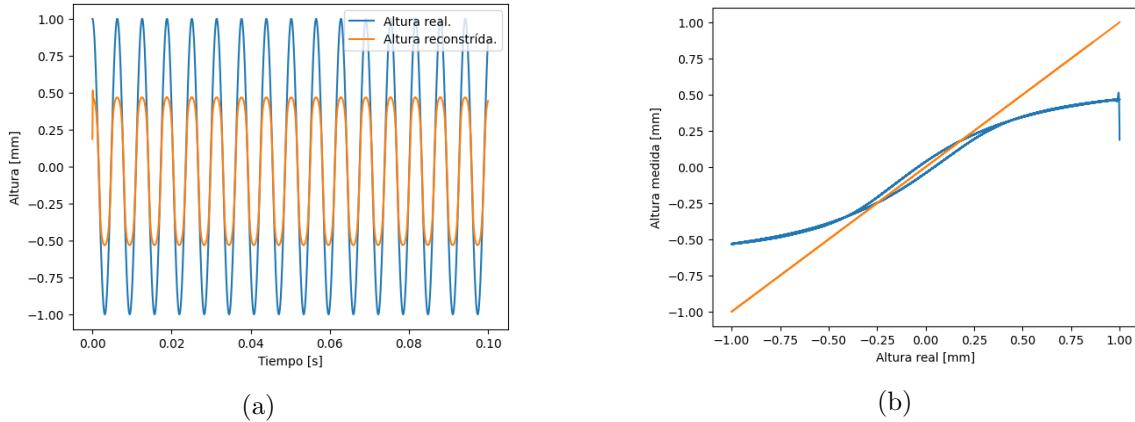


Figura 9: Barido de alturas. A la izquierda la evolución temporal y a la derecha la altura reconstruida contra la original.

Podemos notar que satura cerca de 0.25 mm y que a esta frecuencia de señal hay histéresis. Si aumentamos C_0 , suponiendo C' fija, podemos conseguir aumentar el rango de validez hasta 10 mm y evitar la histéresis a 100 Hz, aunque queríamos frecuencias más altas. Esto se puede lograr también disminuyendo Q_F al disminuir L o aumentar R por ejemplo.

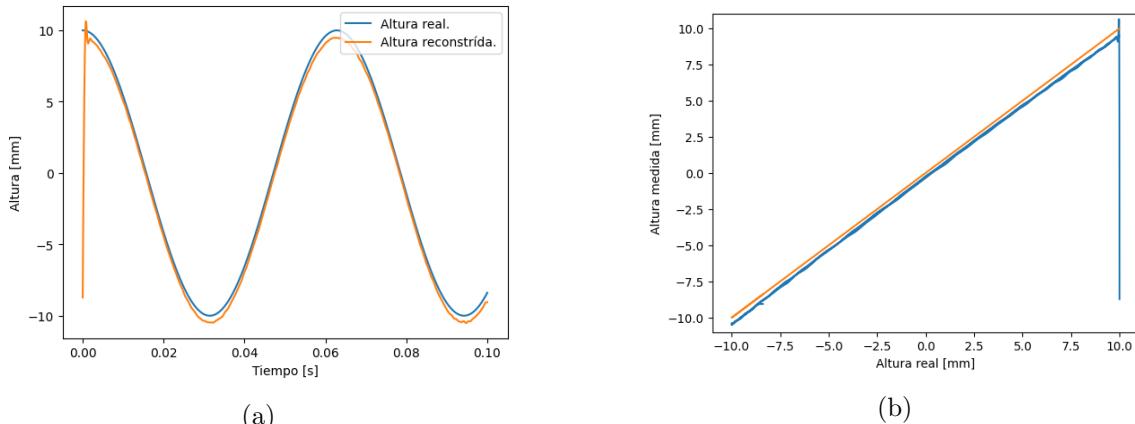


Figura 10: Barido de alturas. A la izquierda la evolución temporal y a la derecha la altura reconstruida contra la original.

Esto último es para $C' = 3 \times 10^{-8} \text{ F/m}$ y $C_0 = 160 \times 10^{-10} \text{ F}$. Habrá que probar experimentalmente qué rango de valores funcionan mejor y si hay más juego antes de que sature.

3. Fabricación del sensor.

Por último se empezó con el diseño del sensor. Éste constará de una carcasa externa para integridad estructural impresa en 3D, diseñada con Blender como se muestra a continuación (primer prototipo).

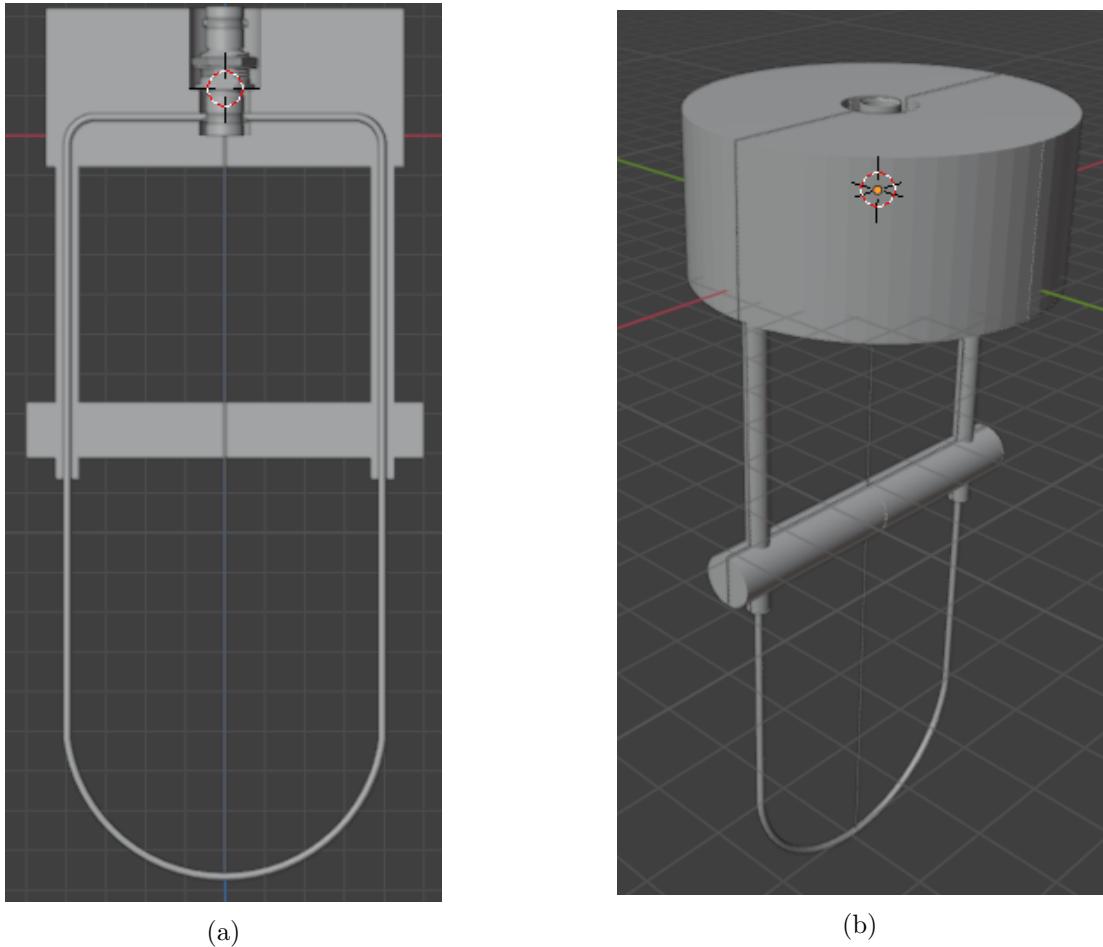


Figura 11: Dos vistas del prototipo inicial. Entero y corte.

Se suelda el cable de cobre a un conector BNC hembra dentro de la carcasa. Además se coloca soldado cable de acero a la carcasa exterior del BNC (a Tierra) que se conectan al alambre por abajo (aislado eléctricamente).

1. Diseño del sensor.

Reducimos el sensor para que no sea tan ancho y más compacto. La nueva versión es la siguiente:

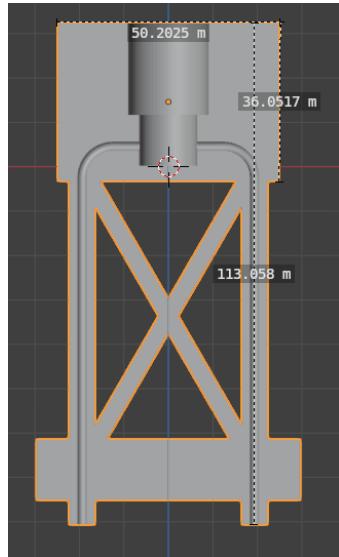


Figura 12: Nuevo modelo, las medidas que dicen metros son en realidad milímetros.

Además le agregamos una cruz para mejorar la integridad estructural, revisar modelo de Truss, tal vez se puede optimizar aún más.

Se mandaron a imprimir dos copias para empezar a probar su funcionamiento.

Semana del 14/04/2025

Ya estaba impreso el prototipo inicial de la carcasa del sensor. Más abajo se muestran un par de fotos de la impresión.

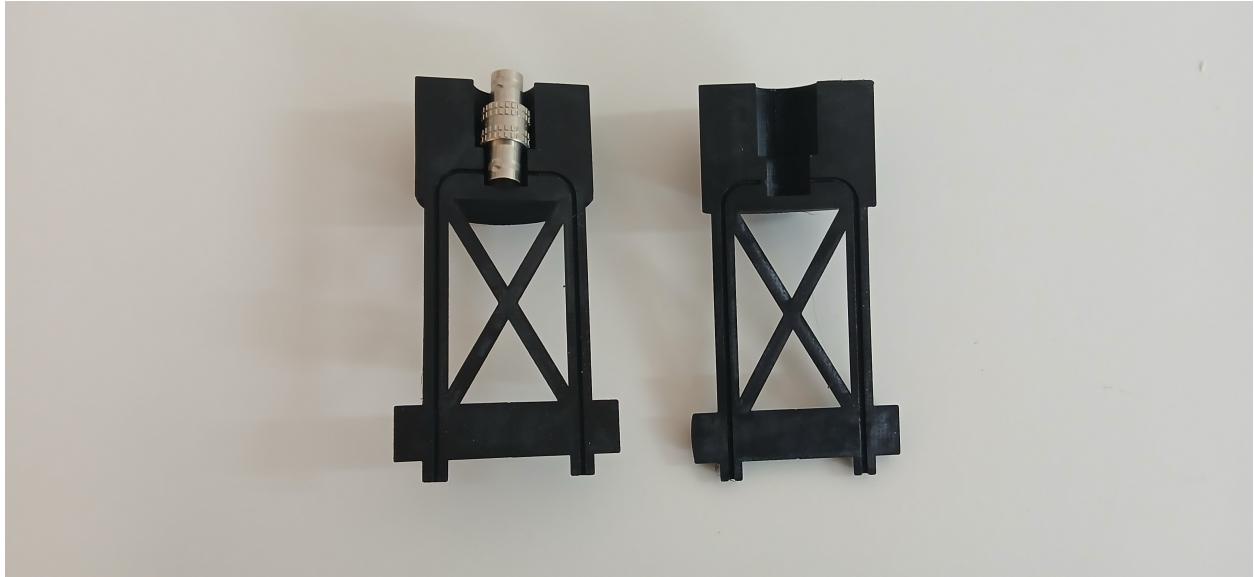


Figura 13: Pieza impresa en 3D.

Un par de consideraciones:

- El agujero para el cable de cobre se cerró, quedó por debajo de la resolución de la impresión.
- El agujero para el cable de acero también se achicó igualmente. Quedó de aproximadamente 1.4 mm de diámetro, entrando un cable tipo jumper de la protoboard.
- La cabeza de BNC entra perfecto, con un poco de juego, pero entra con las dimensiones deseadas y hace tope donde debería.
- Las dos piezas encajan bien entre sí, efectivamente tienen la simetría de reflexión buscada.
- El tiempo de impresión (aunque fue en conjunto con otras piezas) fue de 12 hs aproximadamente. Si queremos hacer una cantidad importante en el futuro se puede hacer más hueco para que tarde menos.
- Para cerrarlos se puede pensar en encastre, que se podría romper al intentar volverlos a separar, o tornillos de plástico con tuerca, dos arriba y uno o dos abajo por ejemplo.

1. Miércoles 23/04

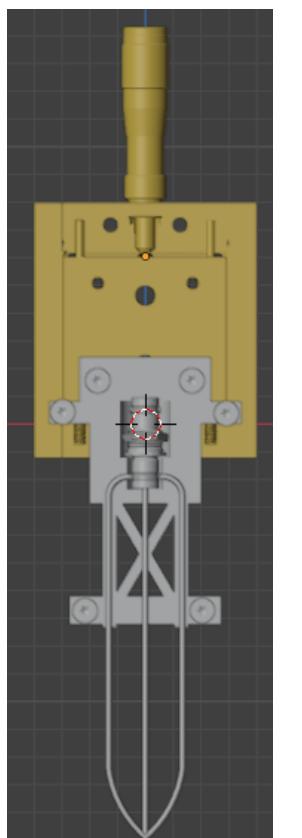
Resumen rápido

- Hicimos los nuevos cambios al diseño de la carcasa del sensor para mandarlo a imprimir.
- Pablo comentó el Lunes de la semana pasada que había comprado los alambres de cobre y que llegaron el Miércoles, sin embargo los diámetros no eran los especificados por MercadoLibre (0.25 y 0.35 mm por 500 g). Realizó la devolución con el reembolso correspondiente (que completaron hoy), y proseguiremos intentando comprarlos presencialmente.
- Respecto al alambre de acero vamos a usar de momento uno de acero que ya había en el Laboratorio bastante rígido, de unos 0.5 mm de diámetro.

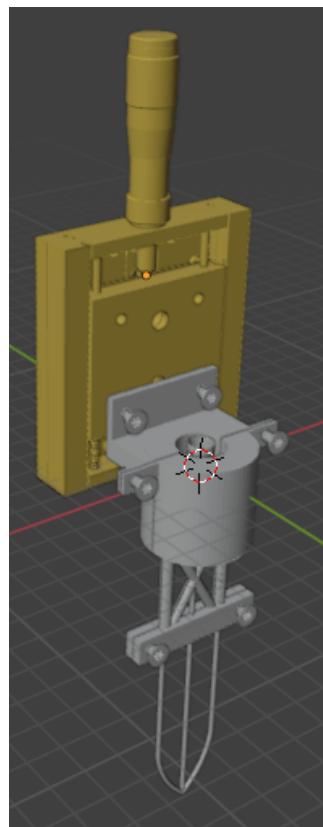
Nuevo diseño de la carcasa

Los cambios que se hicieron fueron principalmente reducir aún más el tamaño de la cabeza del sensor y la altura de la parte de abajo. Además se agregó un adaptador a una de las mitades que permita conectarlo al soporte de un tornillo micrométrico mediante dos tornillos M4. Además se agregaron salientes en la parte de arriba para poder conectar las dos mitades mediante también tornillos M4, y en la sección inferior se cambio el cilindro por un rectángulo con otros dos agujeros que permitan terminar de conectar ambas mitades. Además se aumentó el diámetro del canal para el cable de cobre que se había cerrado en la impresión anterior.

A continuación se muestra el modelo final.



(a) Vista frontal.



(b) Vista 3/4.

Figura 14: Distintas vistas del nuevo modelo de la carcasa del sensor junto con el tornillo micrométrico.

El tornillo y el soporte del mismo son de la marca Newsport, modelos BM17-25 y M-UMR8-25 respectivamente.

Algunos pensamientos para el próximo prototipo:

- Agregar un tope en la parte del BNC (reducir el cilindro) para que una vez cerrado no se pueda salir por arriba tirando del cable.
- Bajar los salientes para los tornillos de arriba que conectan ambas mitades para que queden al ras con el conector al tornillo.

2. Viernes 25/04

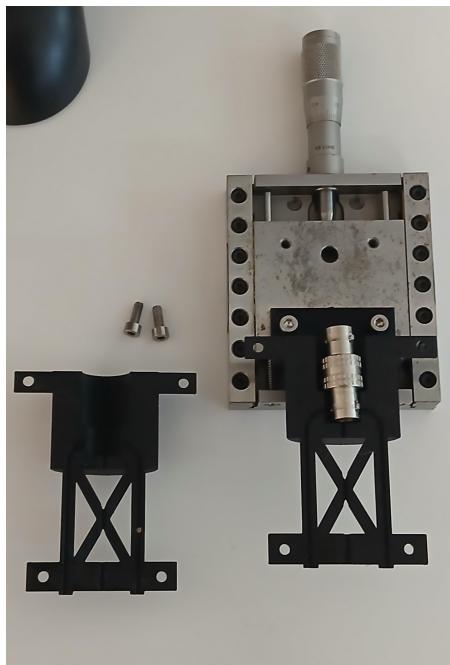
Resumen rápido

- Recibimos las impresiones del nuevo modelo de la carcasa.
- Diseñamos una forma de montar el tornillo micrométrico a un perfil para mandarlo a imprimir también en 3D.
- Se realizó un inventario de los componentes electrónicos presentes en el Laboratorio y se confecionó la lista de compras para los faltantes en ELEMON.

Nuevo modelo impreso.

Recibimos las nuevas impresiones para la carcasa del sensor, que resultaron ser de las medidas especificadas, conectando perfectamente con los agujeros del soporte del tornillo micrométrico. Una cosa a notar es que los demás agujeros para los tornillos M4 se redujeron un poco, aunque igual se pudieron usar, pero casi sin juego.

A continuación algunas fotos:



(a) Vista frontal completa.



(b) Vista frontal.



(c) Vista 3/4.

Figura 15: Fotos del nuevo modelo de carcasa ya impreso y conectado al tornillo micrométrico.

Faltaría conseguir tuercas M4 para terminar de cerrar ambas mitades de la carcasa.

Conecar el tornillo a un perfil.

Pudiendo conectar el sensor al tornillo solo resta poder conectar este a un perfil para poder moverlo entre distintos montajes fácilmente. Se optó por imprimir un adaptador a un perfil de aluminio 40x40 mm con un conector tipo T que encastra y permite deslizarlo sobre el mismo, con dos tornillos M6 que lo atraviesan y permiten fijar la posición con tuercas tipo T en el perfil. Además se conecta el adaptador al tornillo micrométrico mediante 4 tornillos M6.

El posicionamiento pensado para el perfil y el sensor se muestra más abajo, junto con las opciones de perfiles en los que podría montarse. Se eligió trabajar con el de tipo T (el número 2).

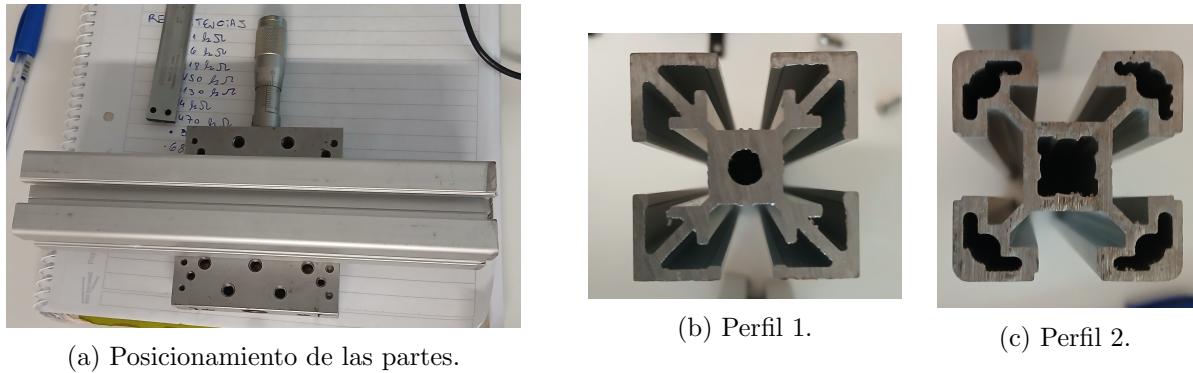


Figura 16: Fotos de las partes relevantes para el conector tornillo micrométrico - perfil.

A continuación algunos renders del diseño en Blender.

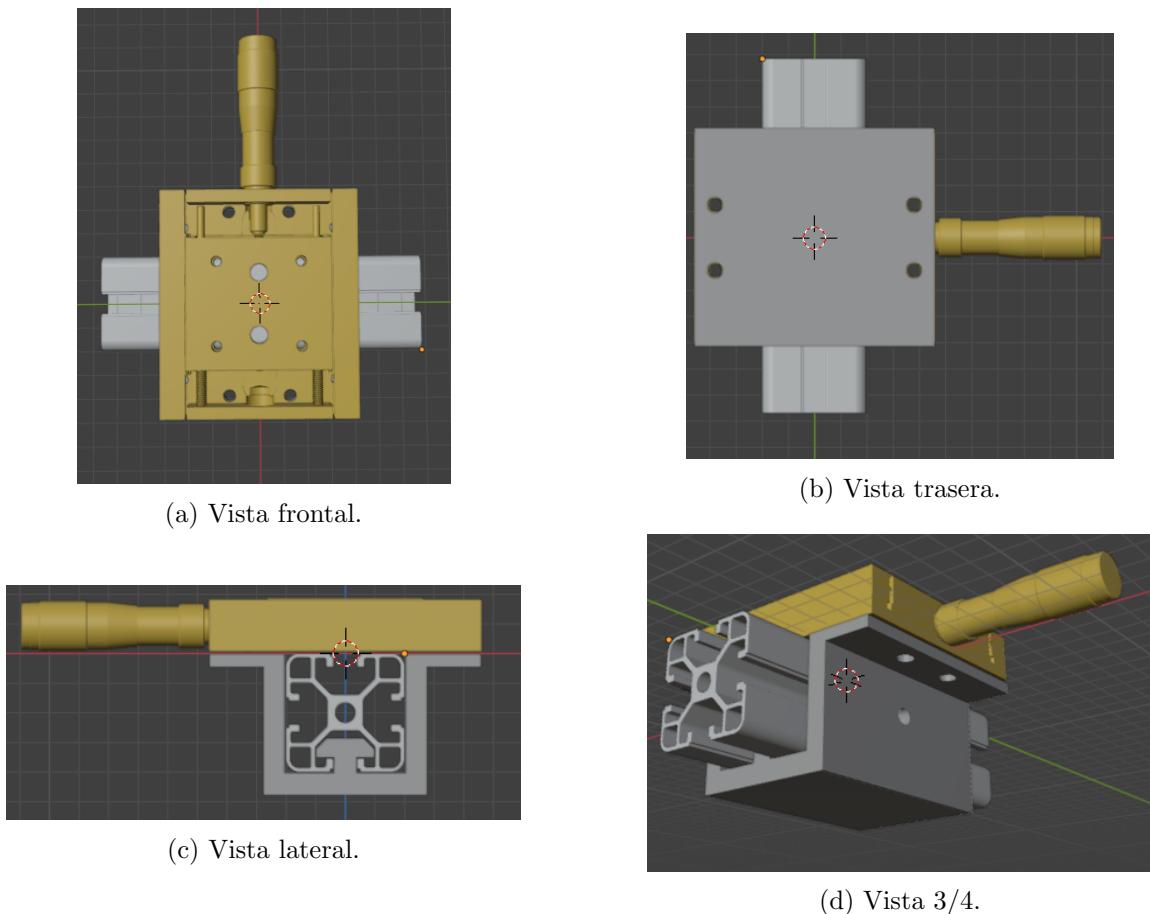


Figura 17: Distintas vistas del modelo del adaptador para conectar el tornillo micrométrico a un perfil tipo T de 40x40 mm.

Componentes electrónicos.

En el laboratorio hay actualmente resistencias en todo el rango desde $1 \text{ M}\Omega$ hasta 10Ω con lo cual no es necesario comprar. Además hay inductancias en el rango de 1 mH a 10 mH , 10 unidades de cada valor aumentando de 2 a 2 mH aproximadamente, con lo cual solo es necesario comprar algunas inductancias más bajas. No hay capacitores.

A continuación la lista de componentes que se deben comprar, en el proveedor ELEMON que fue el elegido por referencia de Pablo.

Producto	Precio x uni	Cantidad solicitada	Stock disponible	Cantidad a comprar	IVA	Subtotal	Guitar
 CA3158901 1PFX500V NPO +/-0,5PF P=5MM ...	USD 0.02905 +IVA	50 uni.	5797 uni. (min. 50 uni.)	50	21%	USD 1.45 +IVA	X
 CA3159081 5.6PFX500V NPO +/-0.5PF P=5MM...	USD 0.02905 +IVA	50 uni.	3840 uni. (min. 50 uni.)	50	21%	USD 1.45 +IVA	X
 CA3159111 10PFX500V NPO +/-10% P=5MM ...	USD 0.0259 +IVA	50 uni.	9790 uni. (min. 50 uni.)	50	21%	USD 1.29 +IVA	X
 CA3159259 100PFX500V SL +/-10% P=5.08M...	USD 0.02905 +IVA	50 uni.	1960 uni. (min. 50 uni.)	50	21%	USD 1.45 +IVA	X
 CA3159391 560PFX500V Y5P +/-10% P=5M...	USD 0.0364 +IVA	50 uni.	6865 uni. (min. 50 uni.)	50	21%	USD 1.82 +IVA	X
 CA3159432 1NFX500V Y5P +/-10% P=5.08M...	USD 0.0322 +IVA	50 uni.	9651 uni. (min. 50 uni.)	50	21%	USD 1.61 +IVA	X
 CA3159570 4.7NFX500V Y5U +/-20 P=5MM C...	USD 0.04655 +IVA	50 uni.	800 uni. (min. 50 uni.)	50	21%	USD 2.32 +IVA	X
 CA3159600 10NFX500V Y5V +/-20% P=5MM ...	USD 0.06265 +IVA	50 uni.	3600 uni. (min. 50 uni.)	50	21%	USD 3.13 +IVA	X
 CA3159220 47PFX500V NPO +/-5% P=5MM ...	USD 0.02345 +IVA	50 uni.	3109 uni. (min. 50 uni.)	50	21%	USD 1.17 +IVA	X
 FI1476440 470UHY 7.7R 125MA AX. 4X11MM	USD 0.18711 +IVA	20 uni.	2608 uni. (min. 10 uni.)	20	21%	USD 3.74 +IVA	X
 FI1473500 100UHY 11E 44MA AX. 2X4MM	USD 0.19206 +IVA	20 uni.	2314 uni. (min. 10 uni.)	20	21%	USD 3.84 +IVA	X

Resumen

Subtotal	USD 23.30+IVA
Impuestos	
IVA al 21%	USD 4.89.-
Percepciones IIBB (6%)	USD 1.40.-
Total (inc. iva)	USD 29.59.-

Los precios no incluyen costos de envío ni percepciones de IIBB. Precios expresados en dólares estadounidenses oficiales al tipo de cambio oficial del Banco Nación Argentina.

[← Continuar comprando](#) [Confirmar pedido](#)

Activar Windows
Ve a Configuración para...

Figura 18: Lista de compras para componenentes electrónicos en ELEMON.

1. Miércoles 30/04/2025

Resumen Rápido

- Estaba lista la impresión del conector entre el tornillo y el perfil. Las dimensiones estaban bien pero la ficha T es muy grande, queda poco espacio por encima y no encastra bien el perfil.
- Estaban las piezas de ELEMON.
- Pablo consiguió el alambre de cobre de 0.35 mm de diámetro.
- También de ELEMON un esmalte aislante .“islamatic”.

Empecé a probar el funcionamiento del circuito RLC y la parte conceptual del funcionamiento del sensor. No fue posible conectar el osciloscopio Siglent a la computadora, sí el generador Tektronix.

Básicamente armé en la protoboard un circuito RLC con $R=1.8 \text{ k}\Omega$ (aunque la cinta decía que era de $1 \text{ k}\Omega$), $H = 10 \text{ mH}$ y $C=100 \text{ pF}$. Luego, con la resistencia conectada en una punta a Tierra levanté con el osciloscopio la caída de potencial sobre la misma (para que no haga corto con la Tierra del generador) y ajusté la frecuencia de resonancia (para que ambas señales estuvieran en fase), que había dado ligeramente distinta a la esperada teóricamente.

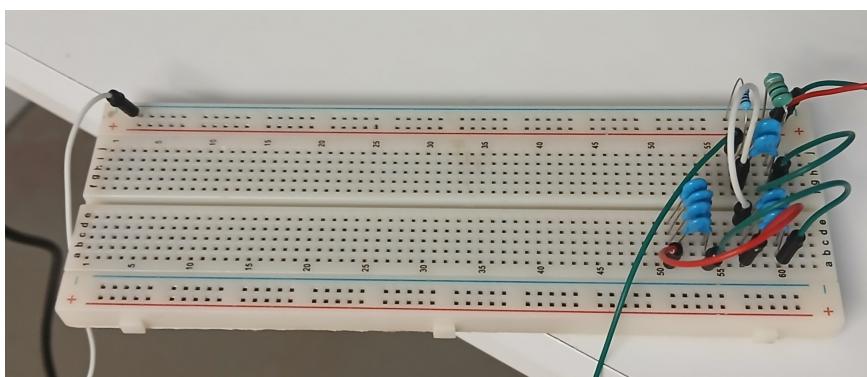
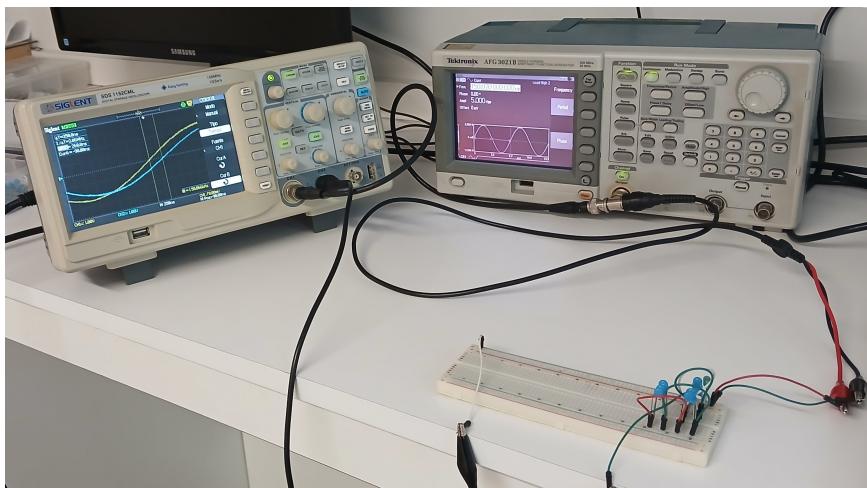


Figura 19: Arriba foto general del setup para el RLC con el osciloscopio y el generador de funciones. Abajo la forma de poner capacitancias en paralelo a la principal C_0 .

Después empecé a poner en paralelo a la capacitancia más grande otras de un par de órdenes de magnitud más chica, para ver si efectivamente esto cambiaba linealmente la fase. Como no pude

conectar el osciloscopio a la computadora para calcular la fase lo hice calculando el ΔT entre los dos cursores de la pantalla, cada uno puesto en el 0 de ambas señales (la de referencia de generador de funciones y la caída de voltaje sobre la resistencia). Luego si se quisiera se puede convertir a fase multiplicando por la frecuencia

$$\phi = \omega \Delta T \quad (90)$$

Abajo están los resultados de esta primera prueba. Al agregar las capacitancias de 5.6 pF se nota que se comporta más o menos lineal hasta unos 15 pF total agregados y después satura como cuando resolvíamos numéricamente la ecuación diferencial del RLC con $C = C(t)$.

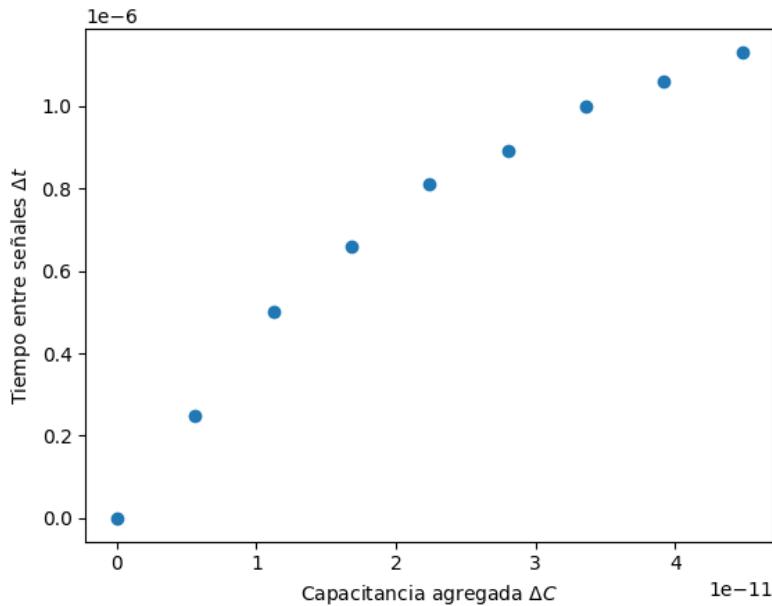


Figura 20: Agregado de capacitancias de 5.6 pF en paralelo a la de 100 pF para ver cómo variaba la fase del circuito.

Después haciendo un barrido mucho más fino se observa que efectivamente el comportamiento para capacitancias chicas es lineal:

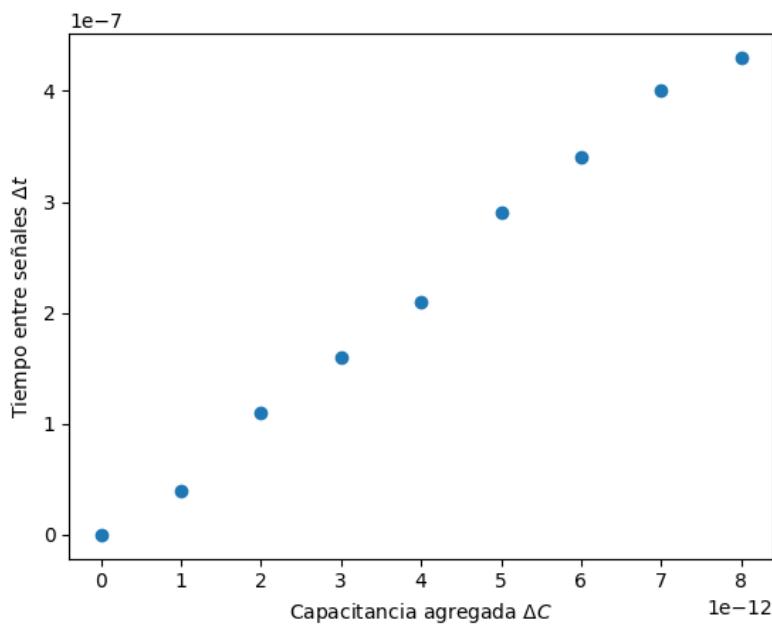


Figura 21: Agregado de capacitacnias de 1 pF en paralelo a la de 100 pF.

Con lo cual si de verdad para el sensor $C = C'\Delta l$ y $C \ll C_0$ entonces podríamos usarlo como queremos. Ya que Δl sería proporcional a ϕ con alguna constante de proporcionalidad que encontraríamos calibrando el sensor, al desplazarlo una distancia conocida Δl en el agua y midiendo el ϕ resultante de esta operación.

1. Lunes 05/05/2025

Continué intentando conectar el Osciloscopio Siglent a la computadora pero no hubo caso. Continúa leyéndolo como dispositivo desconocido.

Después intenté conectarlo a la Compu del Labo que tiene Linux Ubuntu. Acá por lo menos al conectarlo y hacer lsusb lo reconoce como el osciloscopio pero al hacer la lista de recursos con pyvisa no lo lee, mientras que al generador Tektronix sí. Puede que tenga algo que ver con el USBTMC pero no lo tengo del todo claro.

2. Miércoles 07/05/2025

Vamos a intentar dejar todo preparado para el Lunes ir a medir con el Lockin de Laboratorio 4/5.

Voy a seguir trabajando con los elementos base de la semana pasada para el RLC, o sea, $R=1,8\text{ k}\Omega$, $L=10\text{ mH}$ y $C=100\text{ pF}$. Así como está parece que la frecuencia de resonancia real del circuito es de 154.160 kHz .

Ahora estoy viendo el osciloscopio con el canal 1 la referencia del generador de funciones y el canal 2 la medida sobre la resistencia del RLC. Tengo puesto el modo XY para poder ver la figura de Lissajous que se forma entre ambos canales. La frecuencia de resonancia es aquella de máxima amplitud y ambas señales en fase, con lo cual la señal de XY es una línea recta perfecta. Solo agregar una capacitancia extra en serie de 1 pF cambia la fase notablemente en la Figura, y pasa a ser de una línea recta a una elipse rotada 45° .

La resistencia era efectivamente de 1.802 kHz y no 1 kHz como decía la cintita de donde creo la saqué. Comprobado con multímetro Protek 506.

Cable de cobre

Corte un pedazo del cable de cobre de aproximadamente 14 cm. Pelé con el alicate una de las dos puntas para conectarla a la protoboard. Al agregarlo en el aire en una de las dos patas del otro capacitor (la izquierda) no cambia la señal, pero al agregarlo en la derecha sí. Da un salto de 84 ns aproximadamente.

Si agrego de por medio un cable de pin macho-hembra para alargarlo salta bastante más. Unos $180/200\text{ ns}$ aproximadamente.

Lo siguiente que hice fue poner un pin macho-macho de una de las patas del capacitor base al agua y de la otra pata el cable de cobre alargado con el pin macho-hembra también al agua, y al entrar de aire a agua da un salto considerable.

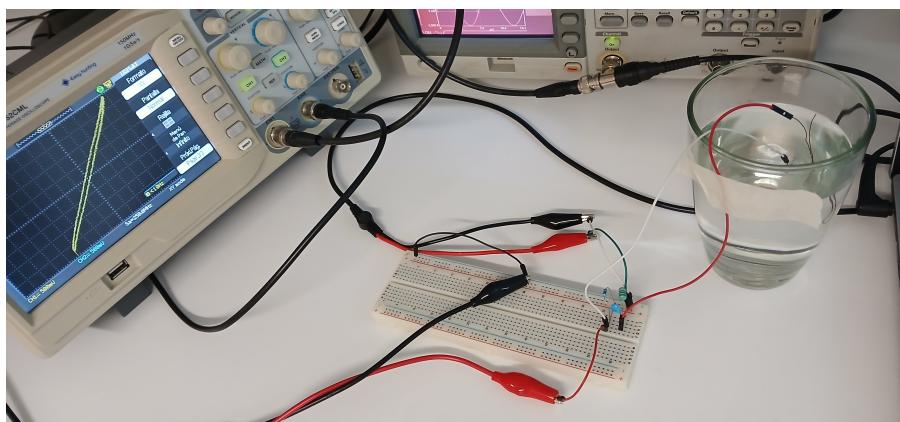


Figura 22: Foto del setup para probar lo que pasa al sumergir el alambre de cobre en agua destilada.

Ahora voy a cambiar la frecuencia para que esté en resonancia en el aire y después lo voy a sumergir un poco a ver cuánto cambia. Para evitar que el extremo no esmaltado (donde corté con el alicate) toque el agua y cierra el circuito directamente lo voy a poner doblado, de forma que quede en el aire, y la distancia sumergida sería el doble de la real.

La nueva frecuencia de resonancia es 151.740 kHz. Una cosa **importante** es que cuando toco el cable con la mano para moverlo cambia la capacitancia y se agrega una fase extra. Con mi mano sobre el cable pin la frecuencia de resonancia da 149.300 kHz. Efectivamente la figura de Lissajeous se abre al tocar el agua (al alambre le cuesta entrar por la capilaridad, pero solo tocarla ya cambia la capacitancia) y pareciera que al sumerjirlo aumenta hasta saturar a cierta profundidad.

Ahora lo pégue con cinta al vaso para no tener que tocarlo. Ajusto la frecuencia de resonancia a 142.270 kHz. Al moverse el agua cambia la fase, pero es muy sensible. Ahora parece que cambió la frecuencia de resonancia a 142.040 kHz.

Bien, ahora cambió a 142.870 kHz al sumergirlo menos al alambre. Quiero probar poniendo una regla y agregando agua para ver el orden de lo que mide. Con la regla también cambia la capacitancia (por lo menos tocándola con la mano), y el vaso es curvo, con lo cual no hay forma fácil de colocarla.

Mover mucho el vaso también cambia la frecuencia de resonancia. Ahora está en 143.620 kHz.

La medición del voltaje

Nosotros queríamos que el agua esté a tierra, entiendo que por ahí esto soluciona el problema de que acercar cualquier cosa a los cables hace que varía mucho la capacitancia. Sin embargo para poder medir con el osciloscopio la caída de voltaje sobre la resistencia ésta debe estar a Tierra. Se puede medir flotante usando dos canales del osciloscopio y la función math -, pero no pude camiarle la escala para probarlo y además así no podríamos medir la referencia ya que hay solo dos canales.

Acabo de pensar que si ponemos la Tierra entre la capacitancia y la resistencia por ahí funcionaría bien.

El sensor

Llevé al taller la pieza impresa que conecta al sensor y tornillo micrométrico al perfil de aluminio para que le corten la T y poder conectarlo al perfil, que Pablo había empezado a sacar el Viernes pasado con un Dremel.

Lo arrancó con un destornillador largo y lijó un poco con una lija de las cilíndricas de metal y entró perfecto en el tornillo.

Después de eso monté todo ya con los tornillos correspondientes y pasé los alambres de cable (pelado en una de las puntas) y acero, que verifiqué con el multímetro que no estuviese recubierto de algún tipo de esmalte. Fue complicado engarzar al alambre de acero por los canales del modelo impreso en 3D y no quedó del todo derecho. Todavía faltan las tuercas M4 para terminar de cerrar más fuerte el sensor y por ahí luego si se pueda tirar más fuerte para enderezarlo sin que se salga de los canales.

Después enrosqué la punta del alambre de cobre al de acero para enderezarlo, corté el excedente del lado enroscado con el alicate y tiré para que quede lo más recto posible. Pero no está demasiado ajustado y se desliza sobre el alambre de acero como si fuese un riel. Quedaría recubrir la punta con el esmalte. Luego si funciona ya quedaría soldar los alambres (que ahora se agarrarían con BNC-pinza-pin a la placa protoboard) al resto de los componentes.

El Arduino

Como no pude conectar el osciloscopio a la computadora intenté utilizar el Arduino UNO del Laboratorio para este fin pero tenía muy mala frecuencia de sampleo que no supe cómo subir, costándole incluso medir bien una señal de unos 20 Hz.

1. Lunes 12/05/2025

Hoy fuimos a tratar de probar el sensor con el Lock-in SR830 de Laboratorio 4/5, que se supone debería ser capaz de medir flotante.

También ya recubrimos la punta del cable de cobre con el esmalte aislante cerca de la ventana y lo montamos en una estructura de perfiles que montó Pablo para sumergirlo en una pecera llena de agua destilada, después de dejarlo secar unos 10 minutos. Apoyarlo solo sobre cajas que fue lo primero que probamos no funcionó ya que el centro de masa estaba muy por delante y hacia que rote y no se quede derecho.



Figura 23: Setup para usar el sensor y el Lockin

Conectamos todo, sin embargo el Lockin estaba teniendo mucho ruido. Pasamos a conectar directamente la salida del generador de funciones a la entrada del lock-in y a la referencia externa, sin embargo seguía fallando ya que no estaban en fase. La amplitud estaba bien, ya que era de 2 Vpp del generador y daba 0.71 que es la RMS, que se relacionan:

$$V_{pp} = 2\sqrt{2}V_{rms} \quad (91)$$

Pasamos a usar la referencia interna y funcionaba bien tanto la amplitud (que ahora era RMS la marcada por el equipo) como la fase de aproximadamente 0°. Para esto usamos un tiempo de integración de aproximadamente 1 ms, entrada A para que no salte el fusible.

Después probamos conectar el RLC con el sensor pero estaba marcando cosas raras. Ahí con el oscilloscopio nos dimos cuenta de que el cable BNC estaba roto y lo cambiamos. Después igualmente siguió dando cosas raras así que desconectamos el sensor y nos centramos en el RLC. Éste ahora tenía C=560 pF con una frecuencia de resonancia teórica de 67 kHz aproximadamente. Conectado Tierra con Tierra en unos 73 kHz daba máximo de amplitud y fase casi de 0° pero al cambiar de lugar la resistencia y el capacitor dejaba de funcionar (siempre midiendo sobre la resistencia). Importante también que no se toquen las cabezas de las pinzas para que no haga corto el circuito.

Después dejamos fijos los componentes y dimos vuelta el orden de los cables que medían sobre la resistencia. Cuando estaba el negro con el negro funcionaba bien pero al rotarlos no, incluso cuando estaba en flotante.

Probando continuidad no parecen estar conectadas las cabezas de la salida de la referencia interna con la de la entrada de voltaje (sí al poner negro con negro).

Cambiamos las posiciones de los componentes en la Protoboard para verificar que no fuese eso pero no pareció cambiar el comportamiento. Al rotar los cables de lugar pasaba de unos 2° a unos 77° que no es el salto en 90° esperado por el cambio del signo al medir el votaje en la otra dirección.

Ya después de eso cerca de las 12:00 decidimos guardar todo y volver al Laboratorio ya que el comportamiento no era el esperado.

Incluso manteniendo el sentido de los cables al cambiar de lugar la capacitancia y la resistencia daban cuatro valores distintos, dos por cada posición (cables normales e invertidos).

Probablemente lo mejor sea usar una placa de adquisición para no preocuparnos por el tema de la Tierra y luego procesar a posteriori usando el filtro pasa bajos.

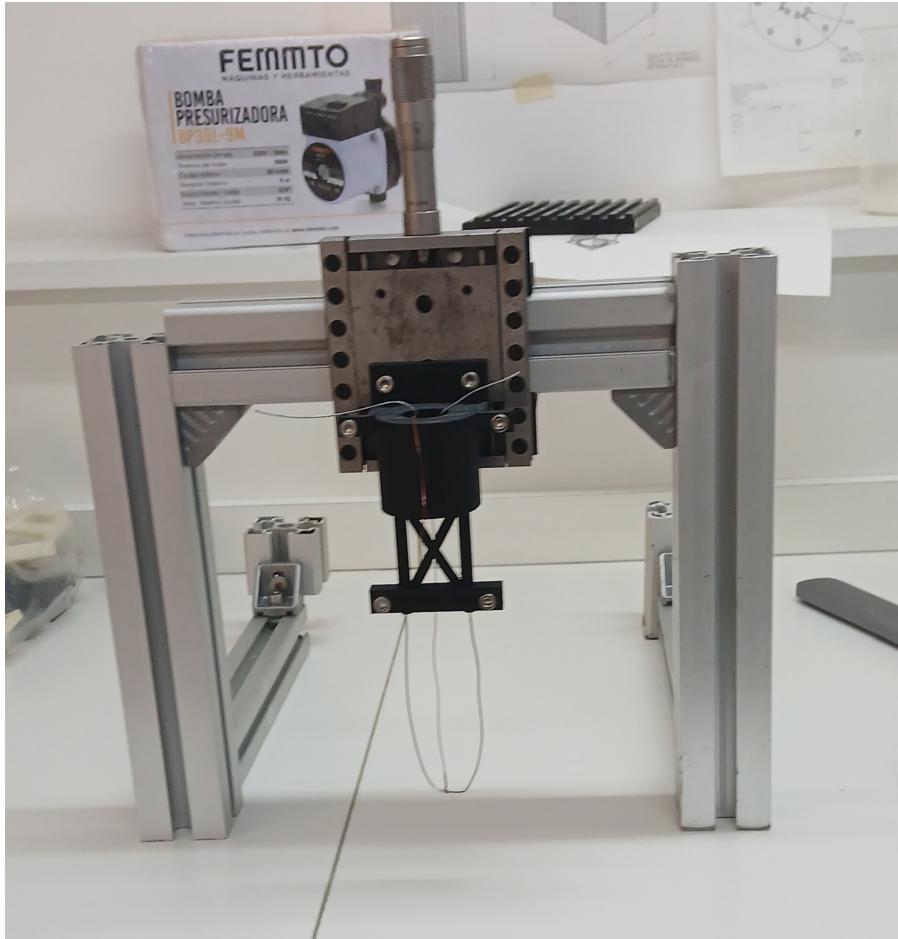


Figura 24: Sensor completo montado ya sobre el soporte.

Algunos links útiles:

- Página para convertir voltaje pico a pico a RMS y vice-versa: <https://www.allaboutcircuits.com/tools/peak-to-peak-voltage-calculator/>
- Página para calcular la frecuencia f y factor de calidad Q de un circuito RLC simple en serie: <https://www.omnicalculator.com/physics/rlc-circuit>

2. Viernes 16/05/2025

Empecé a probar alternativas al Lock-in por las dificultades que dio la vez pasada. De todas formas parece ser que hay uno de los dos lock-ins que da problemas durante los laboratorios según Ale, que fue el que usamos nosotros. Quedaría probar con el otro por si realmente el comportamiento extraño fue cosa del aparato.

Arduino

Estuve investigando la posibilidad de utilizar un Arduino como Osciloscopio o Placa de adquisición como ya mencioné con anterioridad, pero los programas que estuve probando no alcanzaban el nivel de complejidad que necesitaríamos. Usando simplemente el puerto analógico A0 como placa de adquisición logré tomar datos a frecuencia de sampleo de 1 kHz aproximadamente, pudiendo definir bien una señal de 91 Hz. Sin embargo queríamos poder definir señales del orden de 100 kHz, que es mucho más alta. Teóricamente el máximo de adquisición que permite el ADC de la placa es de cerca de 10 kHz, tal vez pudiendo aumentarlo de alguna forma hasta 40 o 50 kHz, pero seguiría sin ser suficiente para lo que queremos usarlo.

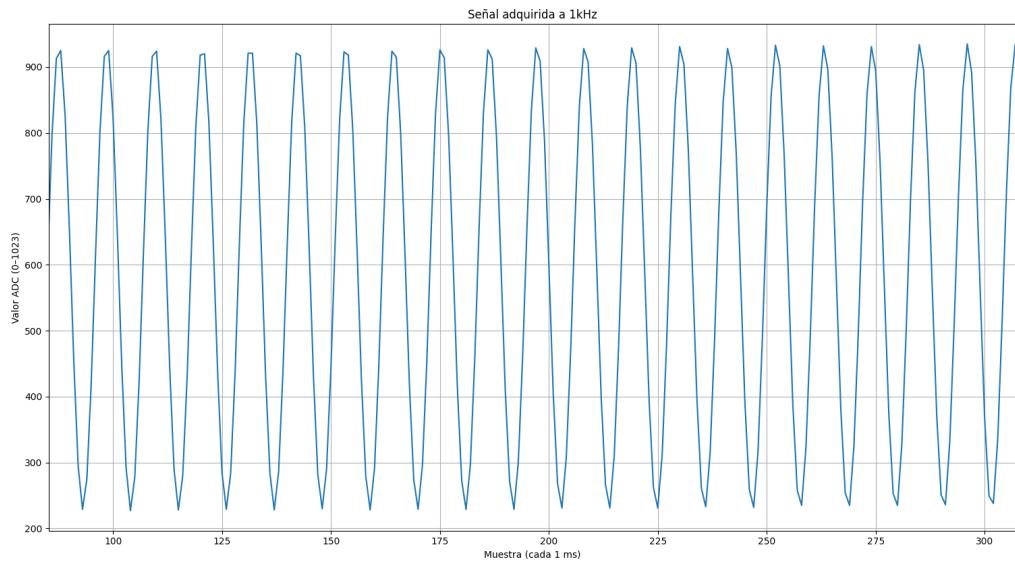


Figura 25: Señal adquirida usando Python + Arduino a 1 kS/s para una señal de 91 Hz.

1. Lunes 19/05/2025

La idea es volver a probar el Lockin el día de hoy. Según dice el manual, el modo A que estuvimos usando es single-ended. Tanto la configuración Ground como float conectan a Tierra, pero la primera con una resistencia de 10Ω y la segunda internamente con una resistencia de $10 k\Omega$. Lo que queremos hacer nosotros debería utilizar dos entradas con el modo diferencial A-B. Se supone que el float es para conectar la Tierra a la de los equipos, mientras que Ground es para usar la propia del Lock-in.

Respecto al AC/DC, se supone que cualquier cosa con más de 136 mHz debería usar el acoplamiento AC, así que eso estaba bien.

Resumen rápido

- Aún en modo diferencial el lock-in de la vez pasada no aprecia funcionar correctamente (o como era esperado). Daba bien intercambiar cables (salto de 180° , no 90°) pero no poner el capacitor a tierra.
- El otro lock-in (con el sticker) pasaba lo mismo.
- Cambiamos los tres componentes y empezó a funcionar (en las cuatro configuraciones daba 90° en la resonancia). Testeamos con el multímetro los anteriores y los valores daban bien así que no sabemos porqué pasaba eso. No volvimos a probar el Lock-in anterior.
- Hicimos pruebas forzando el agua y las calibraciones con y sin Photo-Flo y con sal.
- Calculamos las frecuencias características de la caja con las dimensiones de la misma, para comparar con las que salían de timear el tiempo entre medición y medición del lock-in.

2. Frecuencia de Sampleo.

Para tomar las mediciones de la fase con el Lock-in usamos el código de Diego Shalom, con un loop que tome mediciones, y el tiempo entre sampleos se midió con el módulo *Time* de Python:

```
measures = []
intervalos = []

t0 = time.time()

for i in range(200):
    t1 = time.time()
    phi = lock.get_medicion(isXY=False)[1]
    measures.append(phi)
    t2 = time.time()
    intervalos.append(t2 - t1)
    t0 = t2
```

Y la función que toma la medición en cuestión es:

```
def get_medicion(self, isXY = True):
    orden = "SNAP? "
    if isXY:
        self._lockin.write("DDEF\u1,0") #Canal 1, XY
        orden += "1,\u2" #SNAP? 1,2
    else:
        self._lockin.write("DDEF\u1,1") #Canal 1, RTheta
        orden += "3,\u4" #SNAP? 3, 4
    return self._lockin.query_ascii_values(orden, separator=",")
```

Los resultados fueron los que se ven más abajo, con $T_s = (0,039 \pm 0,004)$ s, o lo que es equivalente, $f = 25,6$ S/s.

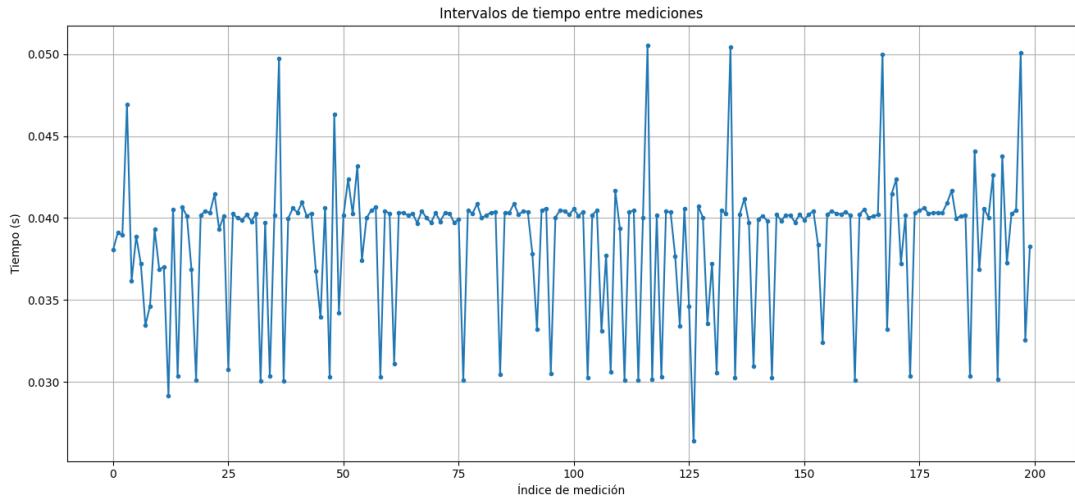


Figura 26: Tiempo entre mediciones consecutivas del Lock-in.

3. Comportamiento inicial - Agua destilada.

Empezamos analizando el funcionamiento del sensor en agua destilada (la del filtro del Labo). Para esto usamos una pecera de vidrio que encontramos arriba en Labo 4/5. Un detalle importante es que tenía brillantina dorada depositada en el fondo, que intentamos limpiar pero estaba pegada y recién se empezó a resuspender en el agua luego de un rato.

Las dimensiones del tanque fueron, medidas desde afuera con un metro:

- Largo: 22.0 cm.
- Ancho: 11.1 cm.
- Alto del agua: 5 cm.
- Grosor de paredes y piso: 3 mm.

Las dimensiones calculadas en consecuencia (restando las paredes y piso):

- Largo: 21.4 cm.
- Ancho: 10.5 cm.
- Alto del agua: 4.7 cm.

La configuración del RLC solo sin sensor fue de 78.55 kHz para fase -89.87° y 2.148 mV, en $\tau = 10$ ms y 6 dB. Siempre usamos los 2 filtros (línea de 50 Hz y 2 x línea de 100 Hz, para un total de 150 Hz).

Lo primero que hicimos fue buscar la frecuencia de resonancia del sistema con el sensor sumergido. Al conectarlo bajó un poco la fase. Ajustamos la frecuencia a 81.68 kHz para que la fase fuese exactamente -90° . La amplitud pasó a ser de 1.823 mV, acá dudo de si tal vez nos paramos en algo que no era la resonancia, aunque terminó funcionando en cualquier caso.

Luego empezamos a sacudir la pecera para ver si veíamos reflejadas las oscilaciones del agua en la fase que media el Lock-in. Acá nos dimos cuenta de un par de cosas:

- Al tocar la pecera para sacudir el agua se daba un salto en la fase, lo que creemos que se debe a un cambio de Tierra del fluido. Haciendo más adelante pruebas de continuidad con el multímetro vimos que el alambre de acero sí estaba a Tierra como queríamos pero al probar el agua no sonaba que fuese continuo (incluso volviendo el agua muy conductora al agregar sal).
- Al tocar el tornillo micrométrico pasaba lo mismo, se daba un salto en la fase.

- Las oscilaciones se veían bien pero con bastante ruido. Acá decidimos empezar a variar el tiempo de integración τ . Ellos ([12] usaban $\tau = 10$ ms).

En la Figura se nota (en amarillo) cómo solo al estar en reposo el agua y tocar y dejar de tocar la pecera se dan saltos en la fase, y mientras se fuerza (en azul) pasa lo mismo.

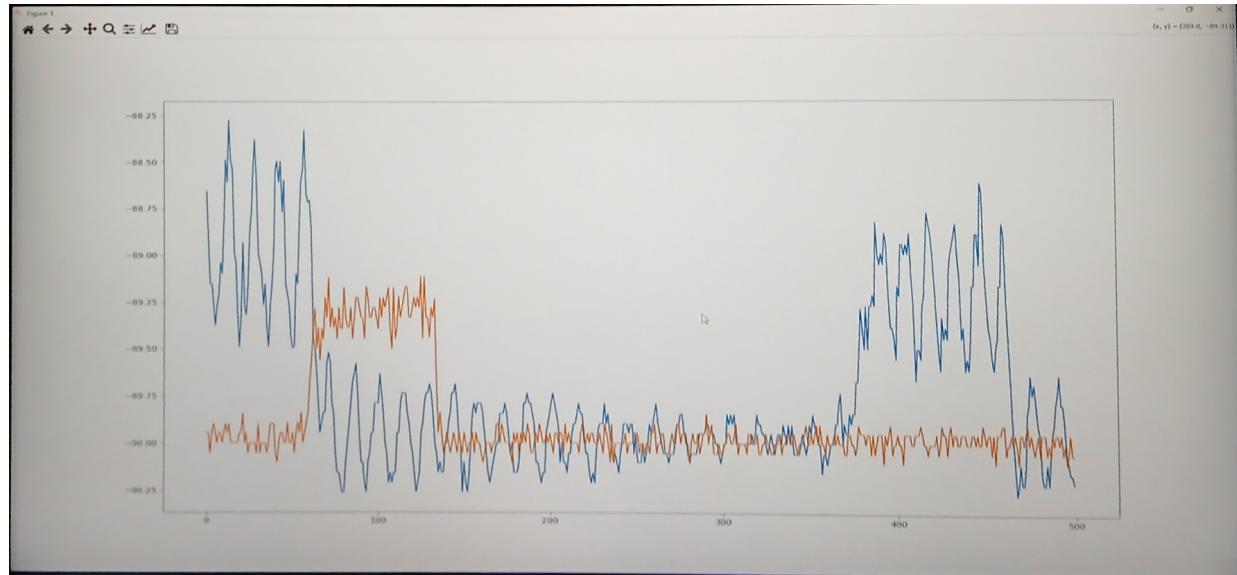


Figura 27: En Azul forzado y después se suelta la pecera y se vuelve a tocar hacia el final. En amarillo el agua en reposo y de repente se toca por un tiempo la pecera sin moverla.

Después tratamos de reducir el ruido variando el tiempo de integración, como se ve en la Figura de abajo.

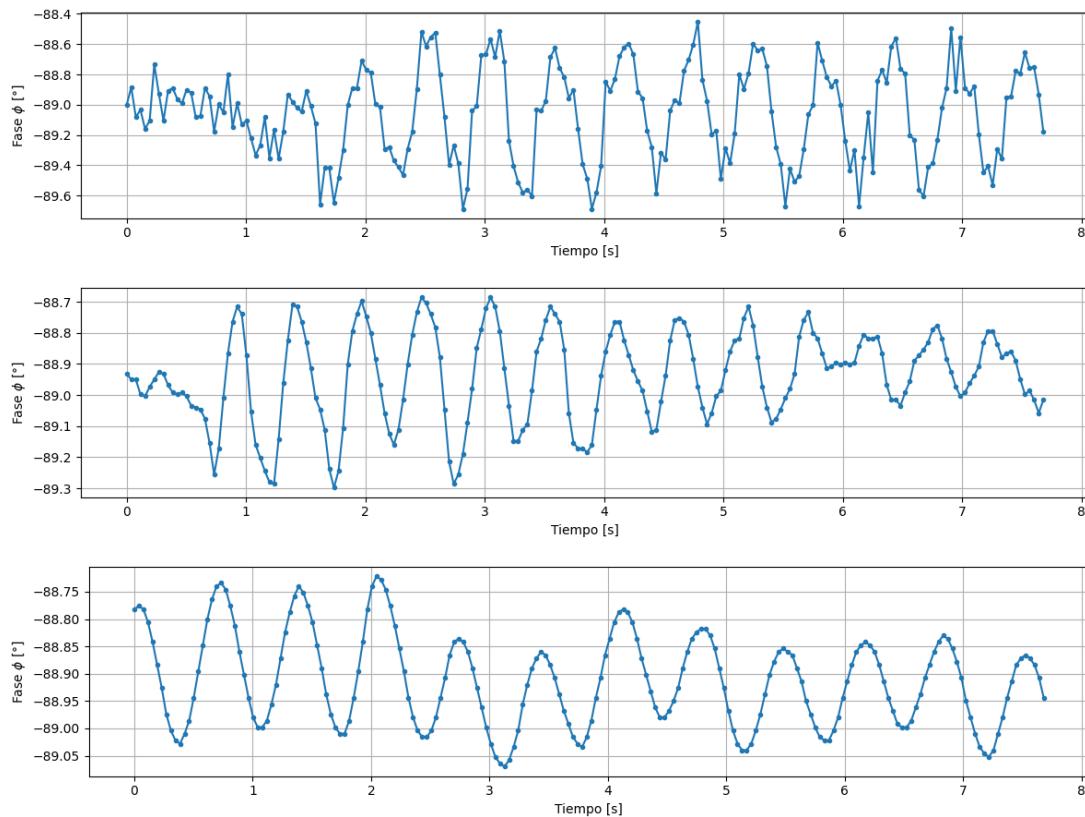


Figura 28: De arriba hacia abajo aumenta la constante de integración en un forzado que deace. Son $\tau = 3$ ms, 30 ms y 100 ms. Notar cómo al aumentarlo disminuye el ruido y se suaviza la señal.

De acá en adelante nos quedamos con $\tau = 100$ ms para el resto de las pruebas.

4. Curva de calibración - Agua destilada.

Algo importante es que las calibraciones se hicieron sin tocar nada cerca del sensor, pero que las mediciones algunas se hicieron tocando la pecera donde se producía un salto en el valor medio de la fase. Por ejemplo al tocar el tornillo pasaba de ser -90° a -89.88° .

Arrancamos con el tornillo en 10 y la frecuencia de resonancia ajustada en 81.25 para que la fase fuese -90° . Fuimos cambiando en ambos sentido el tornillo de a 0.5. Acá paramos cuando tres valores dieron lo mismo ya que parecía que el sensor ya saturaba como mostraban las "simulaciones".

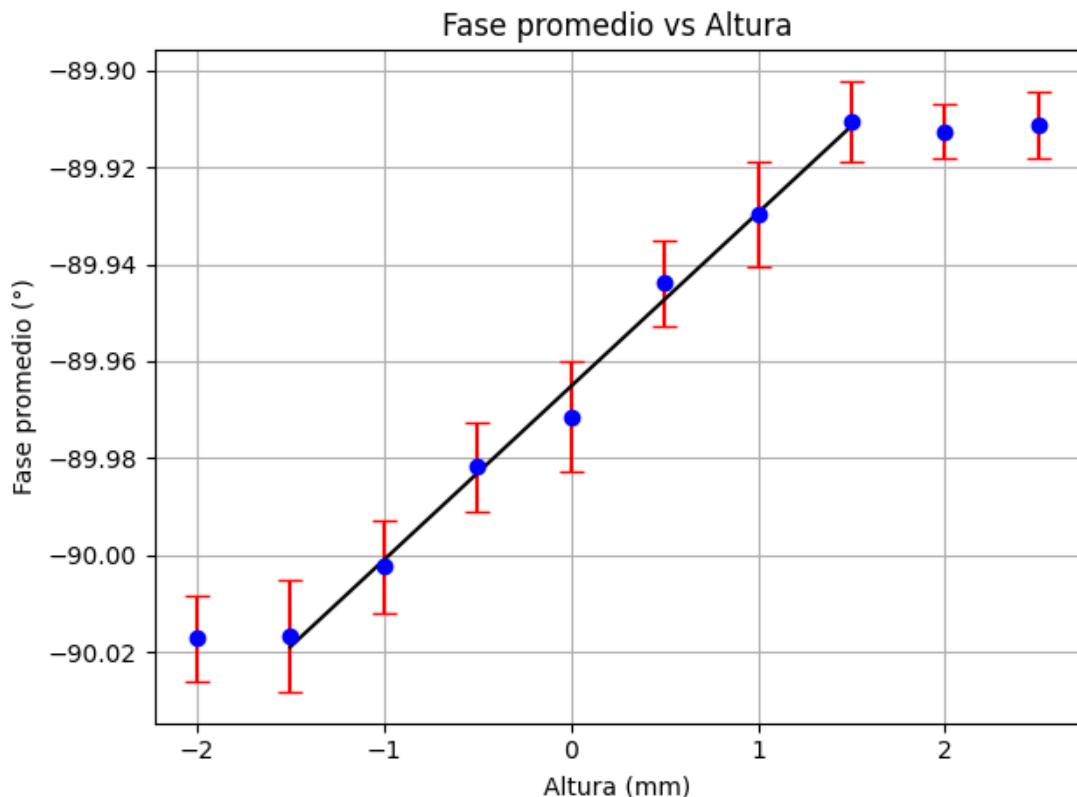


Figura 29: Calibración para agua destilada sola, con el ajuste lineal en la zona central.

Los parámetros que obtuvimos de este ajuste lineal son: $R^2 = 0,99668$, $m = (0,036 \pm 0,001) \text{ mm}/^\circ$, $b = (-89,96 \pm 0,001) \text{ mm}$ para un ajuste lineal de la forma $f(x) = mx + b$.



Figura 30: Zoom del rulo donde toca el alambre de cobre con el de acero, se tuvo cuidado de que no llegase a tocar la superficie libre.

5. Curva de calibración - Agua destilada con PhotoFlo.

Después de agregar una cucharadita blanca con PhotoFlo se repitió la calibración. Ahora nos dimos cuenta de que pasando los dos puntos donde parecía que saturaba seguía moviéndose de forma más o menos lineal, así que seguimos avanzando con el tornillo en ambos sentidos. Además al forzarlo no parecía saturar muy rápido, daban fases por fuera de lo que parecía el rango lineal anterior.

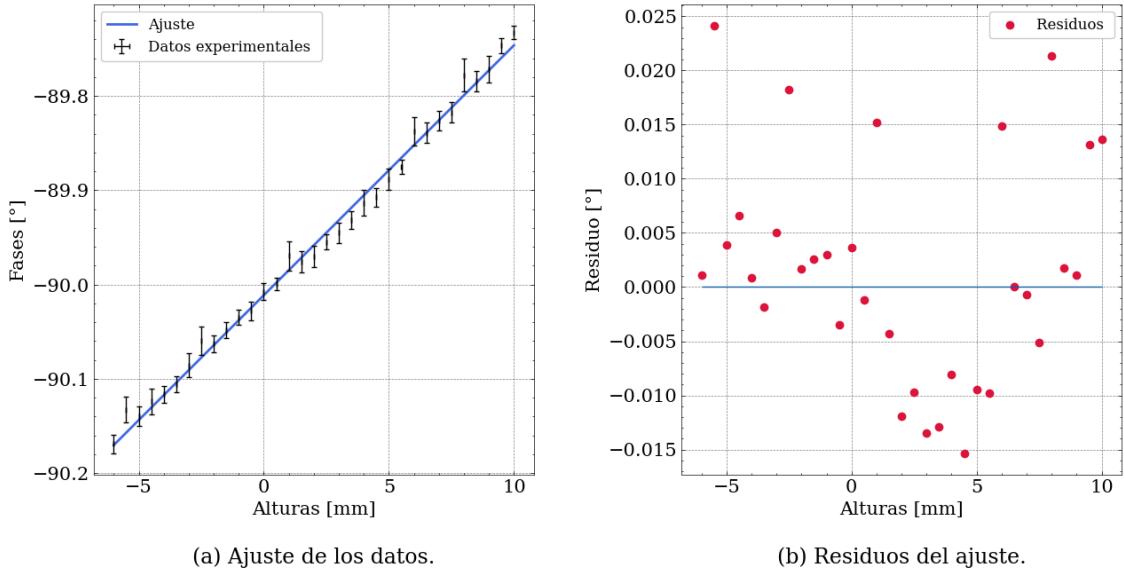


Figura 31: Calibración para agua destilada con PhotoFlo. Se muestra el ajuste lineal y residuos.

Acá los resultados del ajuste fueron: $R^2 = 0,99338$, $\chi^2_\nu = 0,91502$, $m = (0,0265 \pm 0,0003) \text{ mm/}^\circ$, $b = (-90,011 \pm 0,002) \text{ mm}$. Notamos que en ambos casos da muy lineal y con un slope similar, aunque ligeramente mayor en el caso sin PhotoFlo, donde debería mojarse el alambre de cobre.

Importante: en la última altura (con el sensor lo más alto posible) tuvimos cuidado de no llegar a los rulos que tiene donde se conecta con el alambre de acero (que por el esmalte parece que se pegaron), pero inicialmente dio un valor de -89.12256° muy lejos de la tendencia, y resulta que había una brillantina tocando el sensor en ese momento, después de sacarla repetimos la medición y dio coherente con la tendencia, así que pensamos que fue eso.

6. Validación del sensor - Agua destilada con PhotoFlo.

Para saber si lo que estaba midiendo el sensor era coherente o si eran cosas con histéresis/Retraso entre otras cosas tomamos la medición del forzado y calculamos la FFT, para comparar con las frecuencias naturales que surgen de las dimensiones de la pecera:

$$2\pi f_{nm} = \sqrt{\left(gk_{nm} + \frac{\sigma}{\rho} k_{nm}^3\right) \tanh(k_{nm}d)} \quad (92)$$

Con los $k_{nm}^2 = k_{xn}^2 + k_{ym}^2$ y $k_{(x,y)(n,m)} = \pi * (n, m) / (w, h)$ Siendo (w, h) el largo y ancho (acá w es la longitud mayor). Además $\sigma = 0,072 \text{ kg/s}^2$ acá lo tomamos como $0,040 \text{ kg/s}^2$ por haber agregado el PhotoFlo, aunque esto último no cambiaba especialmente las frecuencias naturales en el rango de Gravedad (hasta tipo 10 Hz).

Se nota que la frecuencia más excitada es la primera longitudinal, en la dirección que estábamos forzando al sistema, como era de esperarse. Estaría bueno forzar en la otra dirección o en diagonal o algo más general.

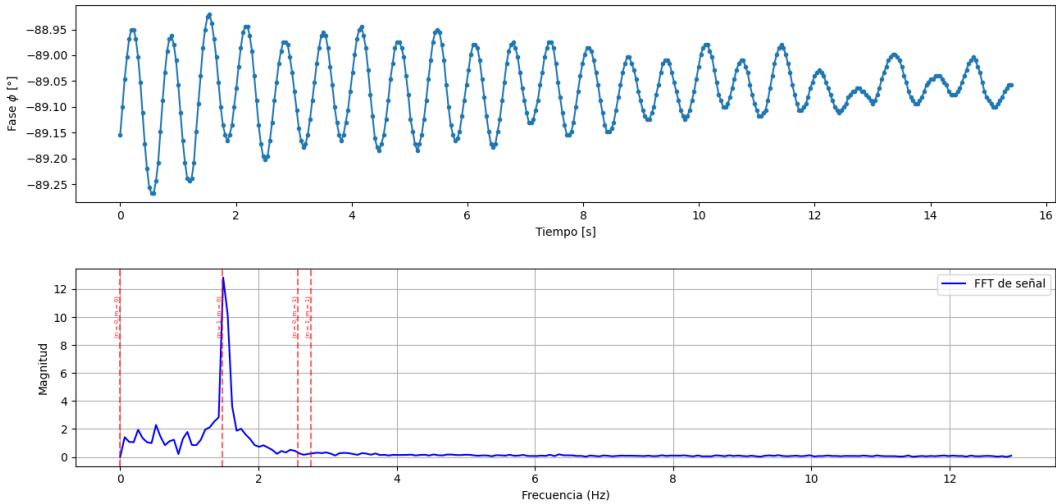


Figura 32: FFT de las oscilaciones de agua destilada con PhotoFlo, y en rojo marcadas verticalmente las primeras frecuencias naturales de la pecera. Oincide bastante bien el primer pico con el primer modo longitudinal (la dirección más larga, que es en la que lo estábamos forzando).

7. Comportamiento - Agua con sal y PhotFlo.

Para terminar agregamos una cucharada de sal al agua con PhotFlo para tratar de aumentar la conductividad. Entonces cambió la fase de equilibrio de -90° a -91.23° , con lo cual ajustamos la frecuencia de resonancia a 78.52 kHz para volver a -90° y medimos el decaimiento del forzado. Sin embargo como se ve abajo no parece funcionar correctamente, casi que es puro ruido. En [12] usaban de 10 a 30 ppm, nosotros una cucharada que supongo sería bastante más. Tal vez habría que replicar la mezcla y medir la conductividad de alguna manera.

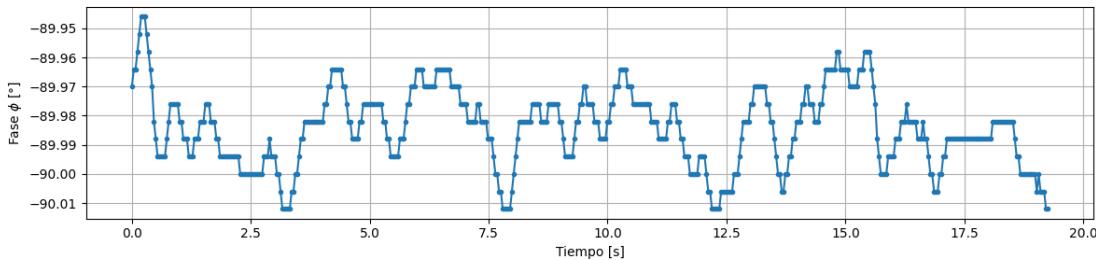


Figura 33: Fases durante la relajación del forzado del agua con sal.

Cosas a discutir.

- Por ahí se planchaba por las irregularidades en el alambre de cobre (no recto).
- Valor medio de 90° , ¿Por qué?. ¿Mezcladas referencias en Lock-in respecto el output?.
- Tierras distintas del circuito y el agua, ¿aislar la pecera? no sería muy práctico en un canal gigante.
- No variamos la resistencia, a mayor R mayor Q_F y más grande puede ser Δl .
- La fase $\phi \propto r/e \Delta l$, a mayor e o menor r deberíamos aumentar el rango.
- Pedir placa de adquisición IO-TECH Personal DAQ 3000 para probar cosas mientras pensamos lo de la Teensy.
- Revisar frecuencia de sampleo máxima Raspberry PI
- Armar placa PCB con KiCAD:
 - Potenciómetro en vez de resistencia fija.
 - 2 x jacks BNC a PCB de 90° , en MercadoLibre pero no la datasheet exacta. Sino comprarlos de antemano para medirlos. Esto para entrada del generador y salida al sensor.
 - 2 x Borneras de 2 pins para lo mismo (más fácil conseguirlo en ELEMON).
 - Borrera para voltaje sobre la resistencia o 1 BNC a 2 con las tierras unidas.
 - Tal vez explorar oscilador 555 a onda sencilla con frecuencia variable mediante potenciómetro.

1. Viernes 30/05/2025

Pedimos prestada la placa, hace falta un transformador AC/DC de 6-16 V y 1A máx para enchufarla, ya que no le alcanza solo con la alimentación de la computadora (no prenden las LEDS), aunque en Ubuntu de la Compu del Labo sí lo reconoce como USB.

- Por ahí se planchaba por las irregularidades en el alambre de cobre (no recto).
Habría que pensar una alternativa, ya sea enderezar el alambre de cobre de alguna forma u otra cosa. Varias ideas serían:
 - Extender la impresión hasta abajo del todo, pero no sabemos como afectaría al agua.
 - Poner varios alambres y promediar, el problema es que todos deberían estar envueltos en agua para que funcione y ya ciertas escalas muy chicas (donde la capilaridad es importante) no se resolverían bien.
- Valor medio de 90° , ¿Por qué?. ¿Mezcladas referencias en Lock-in respecto el output?.
Revisar diagrama de bloques del Lock-in por si agrega la fase, cuál es X y cuál es Y, pero decreció amplitud así que habría que revisarlo.
- Tierras distintas del circuito y el agua, ¿aislar la pecera? no sería muy práctico en un canal gigante.
Poner placas grandes para ponerlo a Tierra. Probar agua de la canilla con/sin agregado de sal.

- No variamos la resistencia, a mayor R mayor Q_F y más grande puede ser Δl .
Hay potenciómetros que varían una década.
- La fase $\phi \propto r/e \Delta l$, a mayor e o menor r deberíamos aumentar el rango.
Casi no cambió respecto de la de [12]
- Pedir placa de adquisición IO-TECH Personal DAQ 3000 para probar cosas mientras pensamos lo de la Teensy.
Ya la fuimos a buscar, falta la potencia, ya que consume más de los 5 V, 0.5 A, que provee la computadora. La compramos, debería llegar el Lunes.
- Revisar frecuencia de sampleo máxima Raspberry PI.
Llega teóricamente hasta 100 kHz, así que tampoco sería ideal para lo que queremos hacer nosotros.
- Armar placa PCB con KiCAD:
 - Potenciómetro en vez de resistencia fija.
 - 2 x jacks BNC a PCB de 90°, en MercadoLibre pero no la datasheet exacta. Sino comprarlos de antemano para medirlos. Esto para entrada del generador y salida al sensor.
 - 2 x Borneras de 2 pins para lo mismo (más fácil conseguirlo en ELEMON).
 - Borrera para voltaje sobre la resistencia o 1 BNC a 2 con las tierras unidas.
 - Tal vez explorar oscilador 555 a onda sinusoidal con frecuencia variable mediante potenciómetro.

Decidimos armar ambas placas, Ale recomendó para los jacks BNC-PCB de 90° los distribuidores iUrbaNet.

2. Placas PCB

Comencé con el diseño de las placas PCB mediante el programa KiCAD.

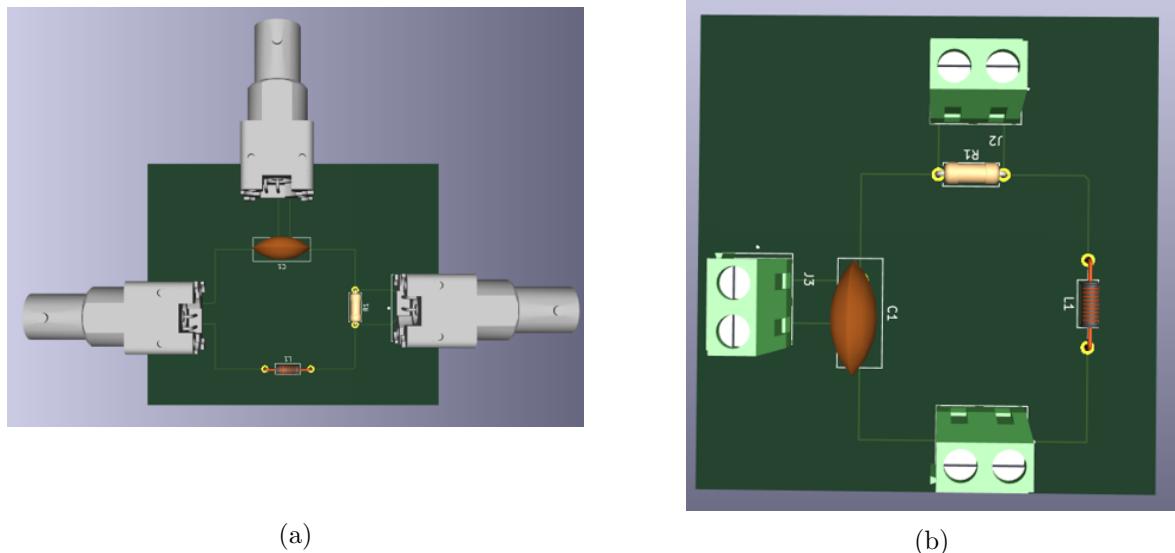


Figura 34: Dos vistas del Modelo 3D de la placa con sus componentes.

Ya están completamente diseñadas las placas, tanto con las borneras de 2 pines a PCB, de ELE-MON, como con los jacks BNC-PCB de 90° de iUrbaNet.

1. PCB con Borneras.

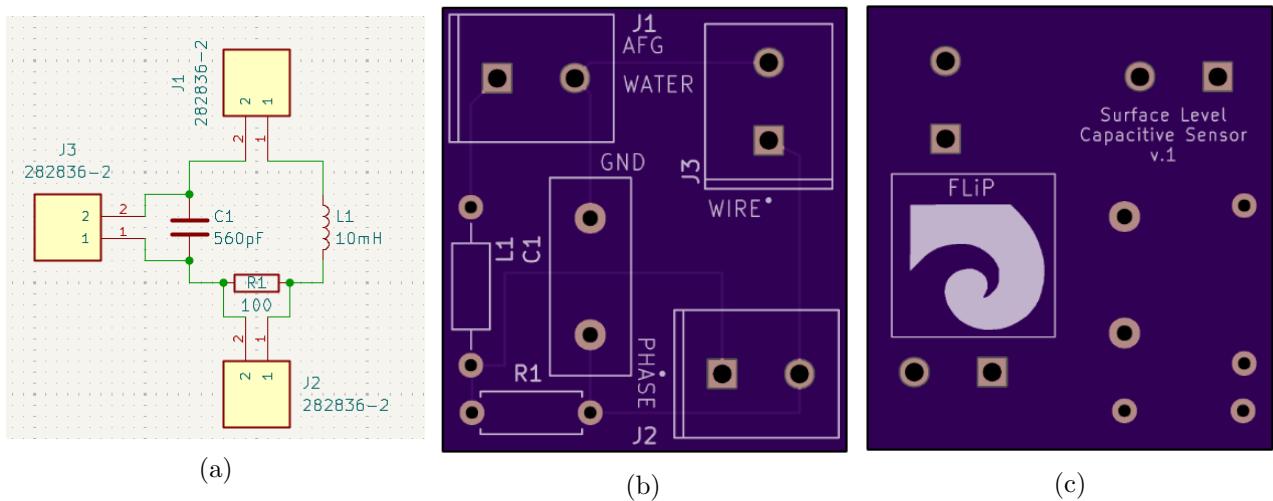


Figura 35: A la izquierda el esquema del circuito eléctrico que corresponde a la placa. A la derecha la vista superior e inferior de la placa terminada.

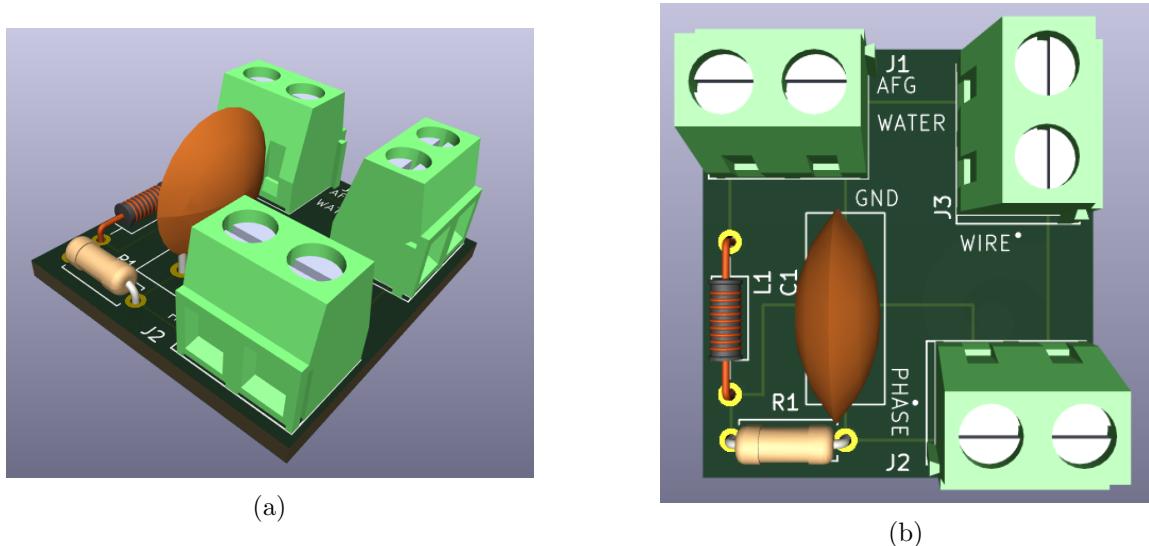


Figura 36: Dos vistas del Modelo 3D de la placa con sus componentes.

La placa terminada tiene dimensiones 26.0 x 28.5mm. El costo estimado de 3 placas debería ser USD 5.70 según OSHPARK.

2. PCB con BNC.

Las dimensiones de la placa son 36.5 x 35.5mm. El costo de 3 placas debería ser USD 10.05 según OSHPARK.

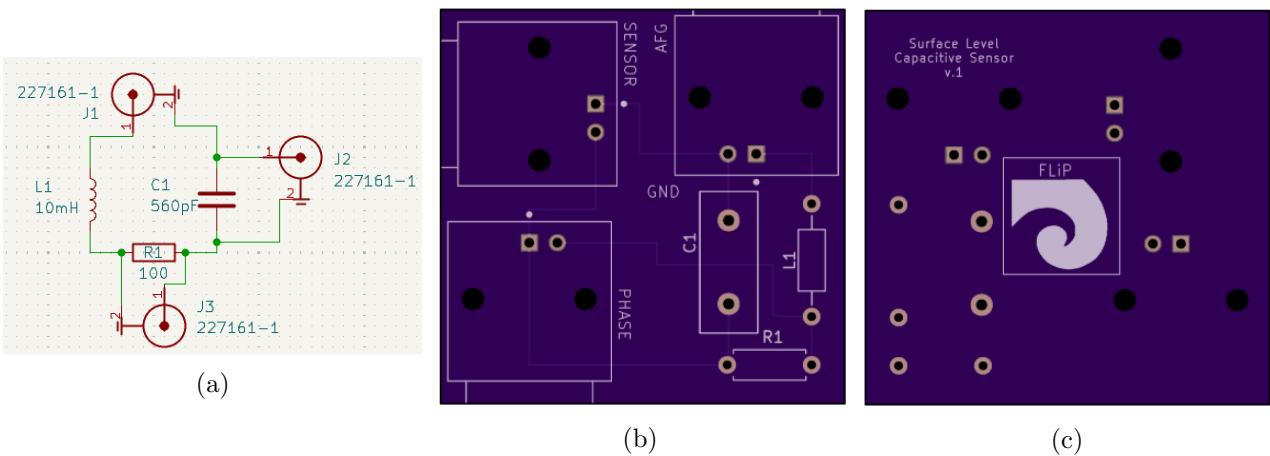


Figura 37: A la izquierda el esquema del circuito eléctrico que corresponde a la placa. A la derecha la vista superior e inferior de la placa terminada.

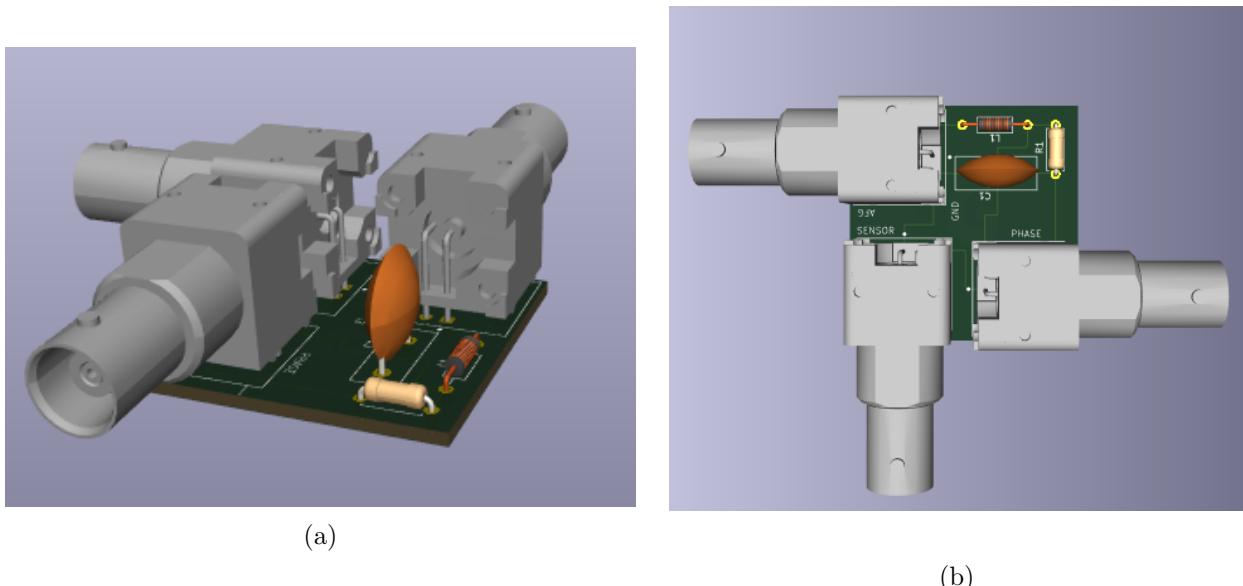


Figura 38: Dos vistas del Modelo 3D de la placa con sus componentes.

3. Componentes.

Los componentes para las placas son las resistencias, capacitancias e inductancias que ya tenemos en el Laboratorio, los agujeros se eligieron según el pin pitch y dimensiones de estos. Después para las conexiones al exterior se utilizarían:

- Bornera de 2 pines: TE Connectivity 282836-2 de ELEMON.
- Jack BNC-PCB 90°: de iUrbaNet, el modelo que declaran no lo encontré, así que usé las plantillas del de TE Connectivity 227161-1.

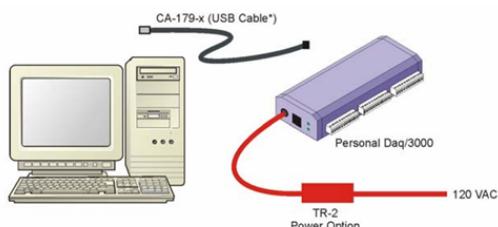
4. Miércoles 04/06/2025.

No llegó el adaptador AC/DC para la placa de adquisición. Fuimos a preguntar arriba a los Labos y a Compu para pedir uno prestado pero no tenían tampoco, ni el jack para soldarlo a un cable y usar una fuente DC regulable de los Labos.

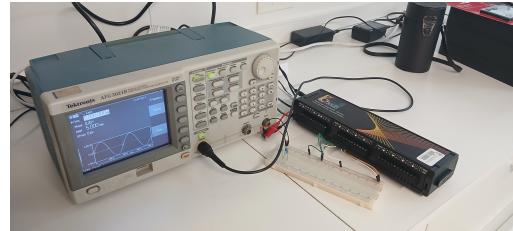
5. Viernes 06/06/2025.

Pablo consiguió la fuente así que podemos empezar a usar la placa. Esta placa (modelo PS 3001) requiere sí o sí potencia externa ya que los 5V a 0.5A máx del USB no le da para operar.

IMPORTANTE: Conectar primero la potencia externa para que luego al conectar el USB pida los requerimientos mínimos de potencia a la computadora.



(a)



(b)

Figura 39: Esquema de la placa y una foto de la misma conectada.

DaqView en Windows 10 no estuvo funcionando, tira algunos errores de cosas no declaradas que no pude arreglar. Intenté también ejecutarlo en modo compatibilidad con Windows XP y 7 pero daba los mismos errores. Principalmente que cuando está activa, si bien parece leer correctamente los valores de los distintos canales, no podía guardar estos valores en disco. Sin embargo sirvió para bajar los drivers de la placa y el .dll necesario para la API. DaqView no funcionó pero pyIOTech, un wrapper de la API en C en Python, con los drivers de la placa sí.

El procedimiento para descargar todo en Windows 10 sería:

- Descargar DaqView para los drivers desde la página de Digilent.
- Ejecutar la instalación.
- Clonar el repositorio de Github de PYIOTech: "<https://github.com/fake-name/PyIOTech>".
- Seguir las instrucciones de instalación, corriendo el "python setup.py install" desde el directorio donde se clonó el repositorio.
- Buscar la ubicación del paquete instalado, en mi caso: "C:/Users/Marcelo/AppData/Local/Programs/Python/Python310/Lib/site-packages/PyIOTech".
- Abrir daq.py y cambiar la ubicación del .dll según donde se haya instalado DaqView. En mi caso: "C:/Program Files (x86)/DaqX/Drivers/USB_x64".
- Cambiar entonces la línea "dll = find_library('daqx')" por "dll = r"C:/Program Files (x86)/DaqX/Drivers/USB_x64/DaqX64.dll". IMPORTANTE: Asegurarse de usar el .dll de 64 bits porque el de 32 bits tira error (no es compatible con Python de 64 bits).
- En DaqView agregar la placa para que instale los drivers correspondientes.
- Correr para encontrar el nombre de tu placa:

```
from PyIOTech import daq

count = daq.GetDeviceCount()
print("Dispositivos encontrados: ", count)
name = daq.GetDeviceList()
```

En mi caso b'PersonalDaq3001374679'.

Después de esto no quedó mucho tiempo pero seguí el ejemplo de PyIOTech para tomar una medición puntual del canal 0 configurado en diferencial.

```

from PyIOTech import daq, daqh

device_name = b'PersonalDaq3001{374679}'
channel = 0
gain = daqh.DgainX1
flags = daqh.DafAnalog | daqh.DafUnsigned | daqh.DafBipolar | daqh.DafDifferential
max_voltage = 10.0
bit_depth = 16

dev = daq.daqDevice(device_name)
data = dev.AdcRd(channel, gain, flags)
data = data*max_voltage*2/(2**bit_depth) - max_voltage
dev.Close()

```

Un par de cosas sobre el funcionamiento de PyIOTech. Está compuesto por dos módulos, *daqh* que contiene las definiciones de todas las flags relevantes y *daq* que contiene la implementación de las funciones con las que es posible comunicarse con el instrumento. Éstas funciones son un wrapper de la API en C que se encuentra en el .dll.

La medición puntual, al igual que las demás, devuelve un entero de 16 bits, y es necesario convertir este valor a voltaje según el rango utilizado. El rango máximo es de -10V a +10V, y luego si se usa una determinada ganancia, por ejemplo 2, se va reduciendo, en el caso de 2 se reduce de -5V a +5V, aumentando la resolución en voltaje ya que los 16 bits son fijos. Unipolar debe usarse cuando queremos que el resultado sea solo positivo, y los 16 bits irían entre 0V y +10V de base.

Si se eligiese Signed entonces el valor devuelto de 16 bits iría entre -32768 y +32768, en Unsigned (que creo está por defecto) viene entre 0 y 65535.

Por último armé un loop que vaya tomando mediciones de a una en el canal 0 de la placa y calculando el tiempo que tardaba en cada una con *time*. El resultado fue conseguir samplear varias de las funciones del generador de 1 Hz de frecuencia, pero a una velocidad de sampleo muy baja.

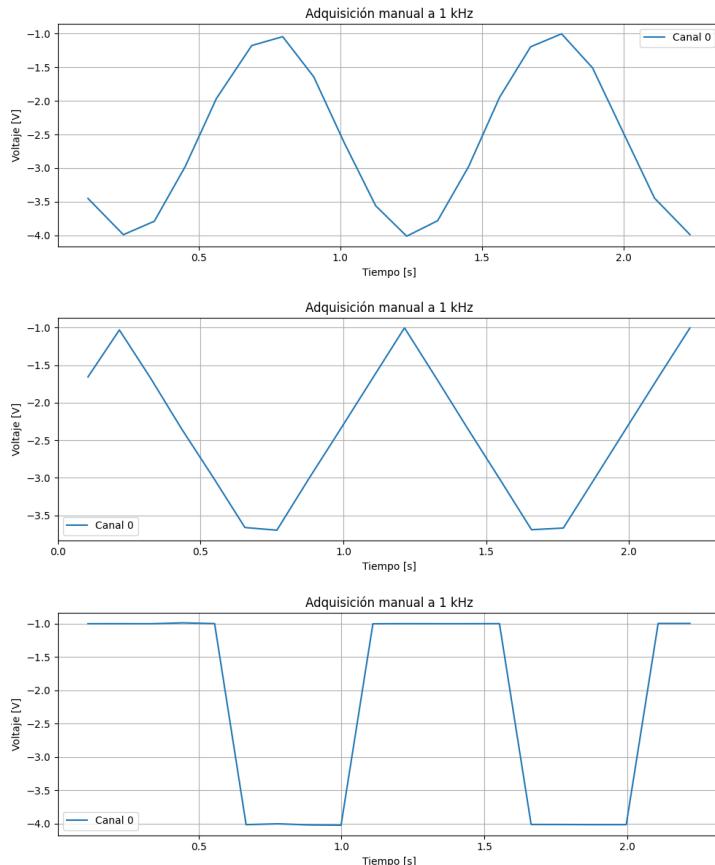


Figura 40: Algunas de las primeras mediciones realizadas con la placa IOTech Personal DAQ 3001 del output del generador Tektronix.

1. Lunes 16/06/2025.

Configurar un Scan.

Lo primero que hice fue pasar de leer un solo canal (donde nos quedamos el Viernes pasado) a leer varios. Para esto se debe configurar los canales que serán medidos durante el Scan. En este caso fueron los canales 0 y 1 con las mismas flags que antes. Luego la función *AdcRdScan* devuelve los valores de ese instante, y lo que hice fue un loop manual donde timeamos el tiempo que tarda cada uno.

```
dev.AdcSetFreq(10000)

volts_data = []
timestamps = []
t_start = time.time()
for i in range(iters):
    vals = dev.AdcRdScan(0, 1, gain, flags)
    data = np.array(vals)*max_voltage*2/(2**bit_depth) - max_voltage

    volts_data.append(data)
    timestamps.append(time.time() - t_start)

dev.Close()
```

Además setee la frecuencia de sampleo de cada canal como 10 kHz, para que la limitación del código fuese el loop (para lo que podemos compensar timeando cada iteración) y no la placa en sí. Nuevamente se deben convertir los valores binarios a voltaje.

IMPORTANTE: En la función *AdcRdScan* de *daq.py* hay que corregir "flags[0]" por "wt.DWORD(flags)" para que funcione correctamente.

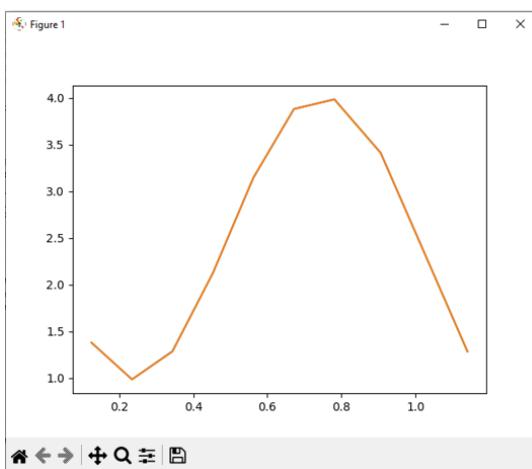


Figura 41: Scaneo manual de los canales 0 y 1 para el mismo output del generador de funciones (un seno de 1 Hz de frecuencia). El canal 0 está en azul por debajo y el 1 en naranja por arriba, pero coinciden casi que perfectamente así que no se nota.

Podemos ver que logramos medir ambos canales pero la frecuencia de sampleo deja bastante que desear, con lo cual el siguiente paso va a ser armar una adquisición continua a una frecuencia de sampleo determinada.

Un par de cosas aprendidas sobre cómo funcionan los Scaneos en la placa.

- La devolución es un array de n valores, con n la cantidad de canales configurados.

- La frecuencia de sampleo configurada es por canal siempre que sea posible, más sobre esto más adelante.

Armar una adquisición de N valores.

Para esto seguí el procedimiento detallado en el Manual del Programador, ya que PyIOTech es solo un wrapper en python de lo que se explica ahí, pero mantiene en general los mismo nombres, argumentos, retornos, entre otras cosas.

El código completo para leer N puntos de la placa es

```
N = 100

dev.AdcSetScan([0, 1], [gain, gain], [flags, flags])
dev.AdcSetFreq(10)
dev.AdcSetAcq(daqh.DaamNShot, 0, N)
dev.AdcSetTrig(daqh.DdtsImmediate, 0, 0, 0, 0)
dev.AdcTransferSetBuffer(daqh.DatmUpdateBlock | daqh.DatmCycleOff, N)
dev.AdcArm()
dev.AdcTransferStart()
dev.WaitForEvent(daqh.DteAdcDone)

vals = np.array(dev.dataBuf)
volts = vals * max_voltage * 2 / (2 ** bit_depth) - max_voltage
```

Este proceso se desglosa a continuación.

Configurar escaneo.

Se establece en *AdcSetScan* los canales que serán leídos con los gains y flags correspondientes a cada uno. Acá usé las mismas flags y ganancia por X1 que veníamos usando desde antes.

Luego se configura la frecuencia de adquisición deseada, esta va a ser la de cada canal siempre que sea posible.

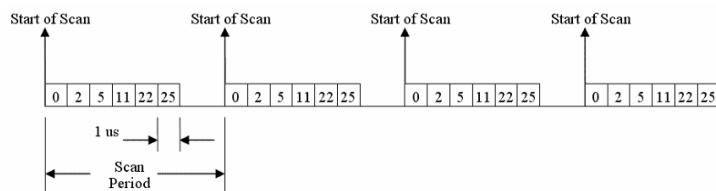


Figura 42: Funcionamiento de la adquisición continua.

El ADC es uno solo en la placa, y las mediciones de todos los canales del Scan se multiplexean para pasar por este, actuando a modo de cuello de botella.

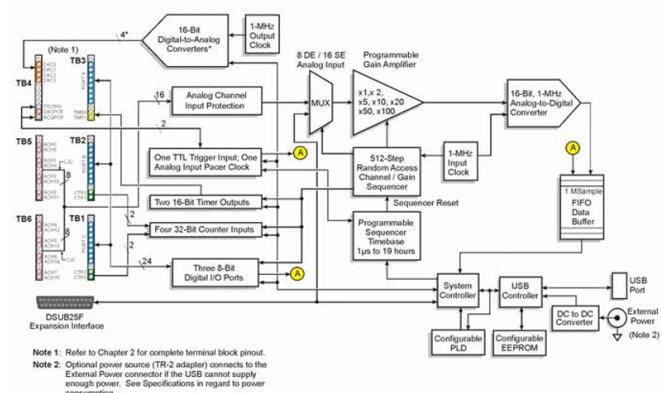


Figura 43: Diagrama de bloques de la PSDAQ 3001.

O sea, que la frecuencia máxima de sampleo es 1MHz aggregated, o sea, **distribuída entre todos los canales** del scaneo, no importan si son single-ended (que mide respecto del common) o differential gastan lo mismo del total (por eso conviene usar modo differential para que dos cuenten como uno solo). O sea, para dos canales, se debería poder samplear a un máximo teórico de 500 kHz cada uno. Además, cada canal necesita x tiempo para medirse, con lo cual esto va a limitar también la frecuencia máxima. En el esquema de arriba está marcado que cada canal tarda 1 μ s en samplearse.

La placa después de *AdcSetFreq* la ajusta para ser la máxima compatible. Se puede saber en cuánto quedó preguntando con *AdcGetFreq*.

Configurar adquisición.

Configuradas las acciones a realizarse en cada escaneo ahora hay que configurar cómo va a funcionar la adquisición de los N escaneos que queremos.

Con la línea:

```
dev.AdcSetAcq(daqh.DaamNShot, 0, N)
```

Estamos diciendo que va a durar N iteraciones (esa es la flag), con 0 disparos pretrigger y N posttrigger, o sea, que todos se toman **después** del trigger.

Después configuraremos el trigger para que sea inmediato al comienzo de la adquisición.

Finalmente debemos configurar el buffer que vamos a usar, donde se van a ir guardando los datos (en la computadora) a medida que los vaya mandando la placa. Como acá sabemos la cantidad de datos que va a mandar la placa, que serán n^*N (canales por iteraciones), vamos a usar un buffer lineal con un tamaño prefijado.

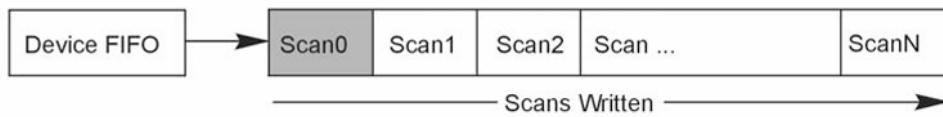


Figura 44: Esquema del funcionamiento del buffer lineal.

Armar la adquisición.

Armamos la adquisición. A partir de este punto la placa va a estar esperando detectar el trigger para empezar a medir, en nuestro caso va a ser inmediato.

La siguiente línea empieza la transferencia de datos, que se va a detener cuando se llene el buffer, aunque este no es el caso, la adquisición podría seguir pero no se escribirían más valores.

Finalmente detenemos la adquisición y cerramos la comunicación con el dispositivo antes de procesar los datos.

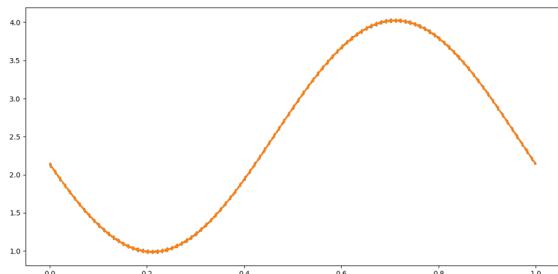
Procesado de los datos.

Como ya habíamos dicho los datos van a estar en binario y va a haber que convertirlos a voltaje. En el buffer además los datos van a estar en el orden que se fueron midiendo, o sea, (canal 0, canal 1, canal 0, ..., canal 0, canal 1), con lo cual se deben separar teniendo esto en cuenta.

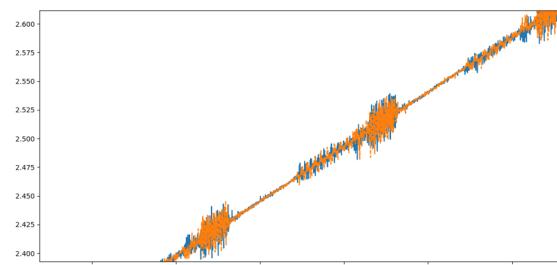
```
max_voltage = 10.0
bit_depth = 16
def convert(vals):
    vals = np.array(vals)
    return vals * max_voltage * 2 / (2 ** bit_depth) - max_voltage

data = np.array(dev.dataBuf)
canal_1 = convert(data[::2])
canal_2 = convert(data[1::2])
tiempos = np.arange(iters)/freq
```

Más abajo una de las primeras adquisiciones que funcionaron.



(a)



(b)

Figura 45: Medición de señal de 1 Hz sampleando a 100 kHz en los canales 0 y 1. A la derecha se muestra un zoom, pero son virtualmente indistinguibles. El ruido que parece ser periódico por ahí son los 50/60 Hz de la pared, los cables no estaban muy proljos.

1.1. Máximo sampleo.

Por último quise caracterizar el RLC con el que venimos trabajando, que tiene una frecuencia de unos 70 kHz. Al intentar samplear con este circuito a esa frecuencia estaba teniendo muy poca resolución de sampleo, y al usar el *AdcGetFreq* resultó que para un canal estaba sampleando a un máximo de 200 kHz y para dos a un máximo de 100 kHz (siempre por canal).

Pensé primero que la placa estaba limitando por algún motivo la frecuencia de sampleo, o bien que los diferenciales contaban como dos canales cada uno, que daría 250 kHz de sampleo máximo para 2, pero seguía siendo bastante más grande de lo que marcaba.

2. Miércoles 10/06/2025.

No pudimos ir al Labo porque tomaron la decisión de tomar la Facultad y cerraron todos los ingresos a Ciudad Universitaria, excepto para tareas esenciales.

3. Resolución en frecuencia del Lock-in digital.

Como parecía que íbamos a estar limitados a samplear a 100 kHz máximo estuve probando si con esa resolución el Lock-in de la semana del 31/03 funcionaría. En principio para reconstruir una señal hay que samplear por lo menos a la frecuencia de Nyquist $f_n = 2 * f_r$. Acá f_r va a ser la de resonancia del RLC, y para lograr que sea menos de 50 kHz habría que aumentar C a 5 nF o más.

Suponiendo que pudiésemos bajarla hasta 10 kHz y sampleando a 100 kHz nos interesa saber hasta qué frecuencia la reconstrucción de la modulación de fase va a ser fidedigna, ya que queremos que el sensor funcione a 1 kHz aproximadamente. Una buena forma de probar esto es generar una fase $\phi(t)$ que tenga contenido en frecuencia constante entre 0 Hz y 1 kHz y ver cómo la reconstruye el Lock-in a partir de samplear $A \sin(2\pi f_r + \phi(t))$.

Para construir ϕ usé:

```
T = 1.0
N = int(T*freq_sample)

freqs = np.fft.rfftfreq(N, 1/freq_sample)

spectrum = np.zeros_like(freqs, dtype=complex)
band = (freqs >= 0) * (freqs < f_max)
spectrum[band] = eps * np.exp(2*np.pi*1j * np.random.rand(np.sum(band)))

phi = np.fft.irfft(spectrum, n=N) * 2 * np.sqrt(N)
ts = np.linspace(0, T, N, endpoint=False)
```

Y al pasar por el lock-in con $f_{cutoff} = 1000$ Hz obtenemos:

Notamos que hasta 750 Hz resuelve bastante bien y luego empieza a bajar. Creo que el decaimiento tiene más que ver con cómo elegir la frecuencia de corte (amplitud decae a la mitad en esta). Si elijo

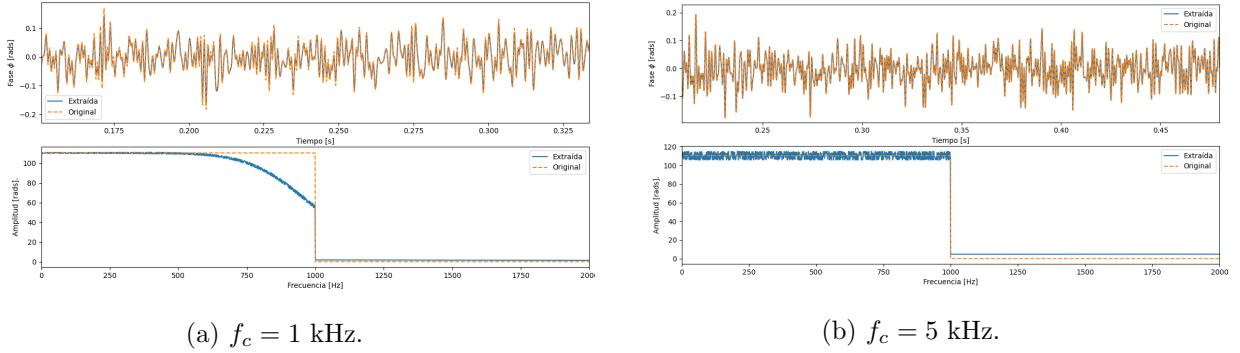


Figura 46: Señal y FFT de la misma, original y reconstruida con el Lock-in para dos frecuencias de corte menores a f_s .

$f_c = 5000$ Hz reconstruye completamente el espectro, así que sería factible usarlo en nuestro sistema bajo esta métrica.

También lo probé con cosenos o senos de una sola frecuencia y funcionaba bien bajo 1 kHz.

4. Viernes 13/06/2025.

Máximo Sampleo.

Finalmente resultó que lo que estaba pasando es que cada canal estaba configurado por defecto para tardar $5\ \mu s$ en ser medido, con lo cual bajaba la frecuencia máxima de sampleo del 1 MHz del ADC a 200 kHz. Esto se puede cambiar con la bandera "daqh.DafSettle1us". También se pueden usar "5us", "10us", "1ms" si se quisiera.

Con esto ya fue posible samplear a 1 MHz máximo en un solo canal, o 500 kHz en dos canales.

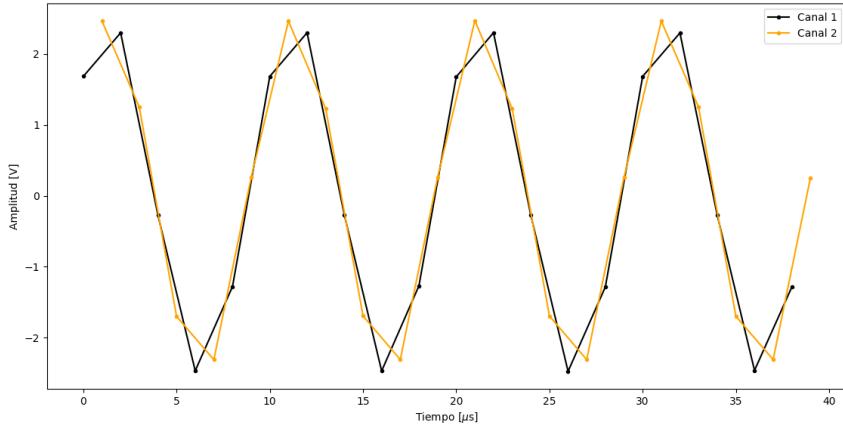


Figura 47: Adquisición en canal 0 y 1 del mismo output del generador de funciones (seno de 100 kHz).

Una cosa importante a considerar acá es que al estar sampleando al máximo posible tenemos que tener cuidado con los tiempos de cada medición, ya que el tiempo de la medición de cada canal es comparable con el tiempo de sampleo entre canales. Ya no podemos decir que ambas mediciones corresponden al mismo tiempo como cuando sampleábamos a frecuencias más bajas. Ahora va a haber $1\ \mu s$ de delay entre el canal 0 y el 1, así que le sumamos ese delay a los tiempos del canal 1.⁶

⁶Antes también estaba este delay pero era muy bajo comparado con el tiempo entre un sampleo y el siguiente.

```

plt.plot(tiempos*1e6, canal_1, ".-", color="black", label = "Canal_1")
plt.plot((tiempos + 1e-6)*1e6, canal_2, ".-", color="orange", label = "Canal_2")

```

Vemos que la señal medida por ambos canales sí es la misma, solo que la samplean en momentos distintos. Si graficamos el buffer crudo sin separar canales es como haber sampleado en un solo canal al doble de la frecuencia de adquisición, confirmando que efectivamente ambos miden la misma señal.

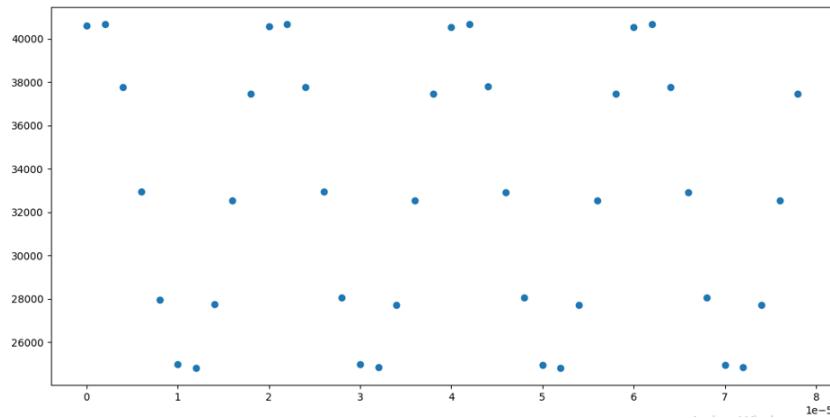


Figura 48: Misma adquisición de la Figura 47 pero sin separar entre canal 1 y 2. Se nota que es un solo seno mejor definido.

Armar una adquisición infinita.

Lo siguiente a programar es la adquisición infinita, a diferencia de cuando tomamos solo N puntos, acá se pude dejar adquiriendo hasta que se desee frenar, o bien tomar una cantidad de puntos fijos pero muy grande, de forma tal que no sería eficiente tener un buffer lineal de ese tamaño en la memoria.

El código final es el siguiente:

```

dev.AdcSetScan([0, 1], [gain, gain], [flags, flags])
dev.AdcSetFreq(freq)
actual_freq = dev.AdcGetFreq()
print(f"Frecuencia de sampleo real por canal: {actual_freq} Hz.")

dev.AdcSetAcq(daqh.DaamInfinitePost, 0, 0)
dev.AdcSetTrig(daqh.DatsSoftware, 0, 0, 0, 0)
dev.AdcTransferSetBuffer(daqh.DatmUpdateBlock | daqh.DatmCycleOn, buf_size)
dev.AdcSetDiskFile(file_name, daqh.DaomCreateFile, 0)
dev.AdcArm()
dev.AdcTransferStart()
dev.AdcSoftTrig()

start_time = time.time()
timeout = 1.0
acqTermination = False
while not acqTermination:
    dev.WaitForEvent(daqh.DteAdcData)
    status = dev.AdcTransferGetStat()
    print(f"Transferidos: {status['retCount']} scans.")
    if time.time() - start_time > timeout:
        acqTermination = True

dev.AdcDisarm()
print(dev.AdcTransferGetStat())

```

Configurar el escaneo.

Esto es igual que cuando tomamos N puntos. Se configuran los canales deseados.

Configurar adquisición.

Ahora se usa "daqh.DaamInfinitePost" para establecer que no sabemos cuántos puntos vamos a tomar, se van a registrar hasta que se detenga en cierta condición. Ahora no importan la cantidad de puntos pos trigger ya que serán indefinidos.

En este caso el trigger no va a ser inmediato sino que va a esperar a que el Software lo trigerá mediante ".^dcSoftTrig".

Ahora el buffer va a ser circular en vez de lineal, ya que los datos se van a ir guardando en el disco se pueden ir reutilizando los espacios anteriores a modo de FIFO (First In - First Out).

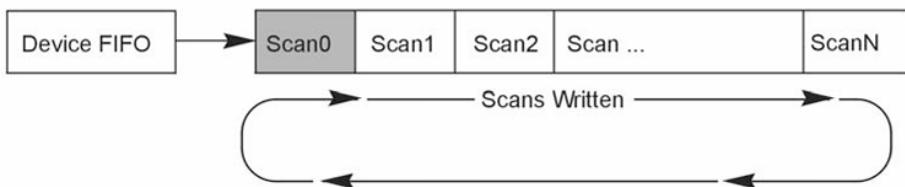


Figura 49: Esquema del buffer circular.

Es importante sin embargo que el buffer sea lo suficientemente grande, ya que el dispositivo acumula datos en su buffer interno y los manda de a tandas hasta donde entiendo, y que le dé tiempo a guardarse en el archivo. Si se recibe el error de "Buffer Overrun" durante la adquisición es bastante probable que sea porque el buffer era demasiado chico. Para samplear a 1 Mhz con un buffer de 50000 (o la función lo transforma en 100000 para 2 canales) parecería alcanzar.

Ahora establecemos en qué archivo se van a guardar los datos binarios (conviene guardarlos así, ya que convertidos serían floats que pesan más a priori). IMPORTANTE: Hay que cambiar en *AdcSetDiskFile* "filename" por "ct.c_char_p(filename.encode('utf-8'))" para que funcione bien el nombramiento de los archivos, sino solo usaba la primera letra del string.

La bandera de "daqh.DaomCreateFile" crea el archivo si no existe, pero si ya existe va a tirar error. Para sobreescribirlo habría que usar otra bandera.

Armar la adquisición.

Se arma igual que antes la adquisición y se inicia la transferencia de datos, y se trigerá inmediatamente después en este caso.

Desarmar la adquisición.

Mientras no haya pasado el tiempo establecido como condición para detener el escaneo se va esperando que haya data disponible en el driver buffer del dispositivo (con la flag "daqh.DteAdcData"), luego se irá transfiriendo al buffer de la computadora y escribiendo en el disco.

Al final con *AdcGetStat* se pregunta cuántos scans se trasnferieron en total.

Lectura de datos binarios.

Después se puede leer el .bin y procesar los datos binarios como antes.

```
binary_data = np.fromfile(filename, dtype=np.uint16)
```

El procesado será sensible a la frecuencia de sampleo y la cantidad de canales, como su configuración. Por esto es conveniente crear un archivo .json de metadata que contenga todo esto, como se hará más adelante.

Error handling.

IMPORTANTE: Para que el printeo correcto de los errores que devuleve la DAQ hay que cambiar en la función "FormatError" de *daq.py* el retorno de "msg.value" por "msg.value.decode(errors='replace')".

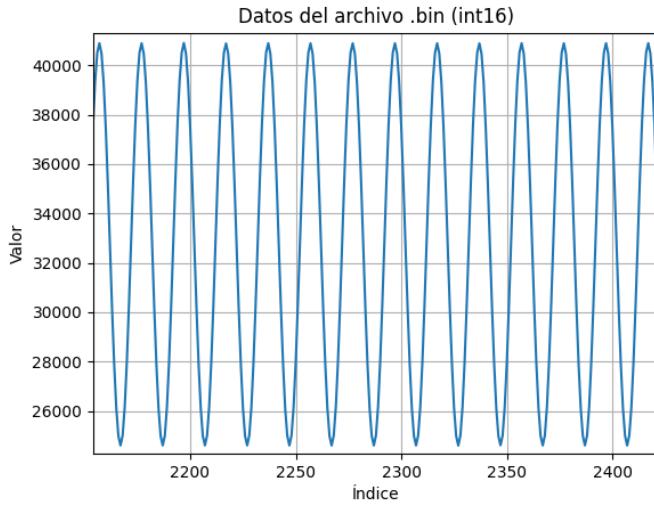


Figura 50: Adquisición directo a disco de muestra.

Interfaz de usuario (GUI).

Con ayuda de ChatGPT armé una mínima GUI:

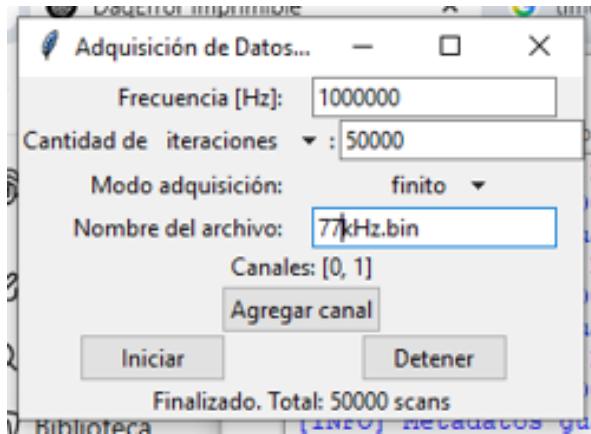


Figura 51: Primera versión de la GUI de adquisición de la placa.

Esta permite tomar mediciones ya sea fijas con un buffer lineal (modo "finito") o indeterminadas (modo "infinito"). También se pueden agregar canales, establecer la frecuencia de sampleo, el nombre del archivo donde se guardarán los datos (que para modo finito guarda el buffer al final del todo cuando se completa la medición) y en el caso de buffer circular determinar su tamaño. Por último en cualquiera de los modos se puede establecer ya sea el tiempo o la cantidad de iteraciones que se quieren tomar (aunque en el modo infinito va a ser un aproximado, y para 1 MHz lo mínimo son unas 8000 iteraciones por el tiempo que tarda en correr el código.)

Alternativamente es posible poner en tiempo infinito un número muy grande y detener la adquisición manualmente cuando se desee con el botón correspondiente.

También genera un archivo "filename_metadata.json" que contiene la configuración de ese escaneo para su posterior lectura.

Esta GUI viene acompañada de un código *Formatter.py* con una función `read(filename)` que lee los archivos .bin y metadata.json y devuelve los tiempos y voltajes correspondientes a todos los canales medidos.

Caracterización RLC.

Haciendo uso de la interfaz se caracterizó el RLC que veníamos usando de $C=561\text{ pF}$, $L=10\text{ mH}$ y $R=1800\text{ k}\Omega$. Fui barriendo con el generador varias frecuencias y tomando scaneos de 50000 puntos a la frecuencia máxima de sampleo. Primero se intentó con solo 5000 puntos pero el lockin no llegaba a resolver bien la amplitud y la fase (un ajuste por sinusoidales sí, pero la idea es usar el Lock-in para este tipo de cosas). El canal 0 es la referencia del generador y el canal 1 la caída de tensión sobre la resistencia, que debería ser proporcional a la corriente circulante por la Ley de Ohm.

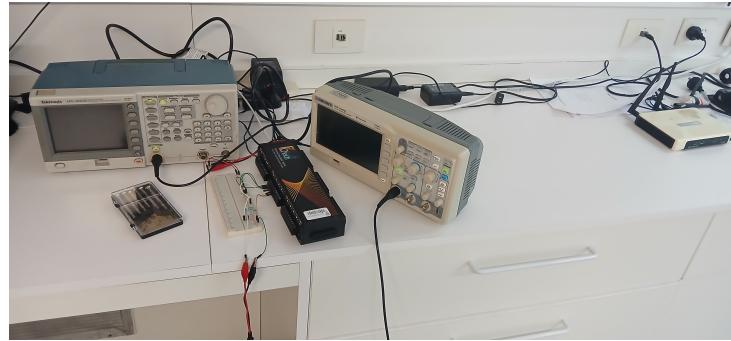


Figura 52: Setup para caracterizar el RLC con la placa.

Los resultados fueron los siguientes:

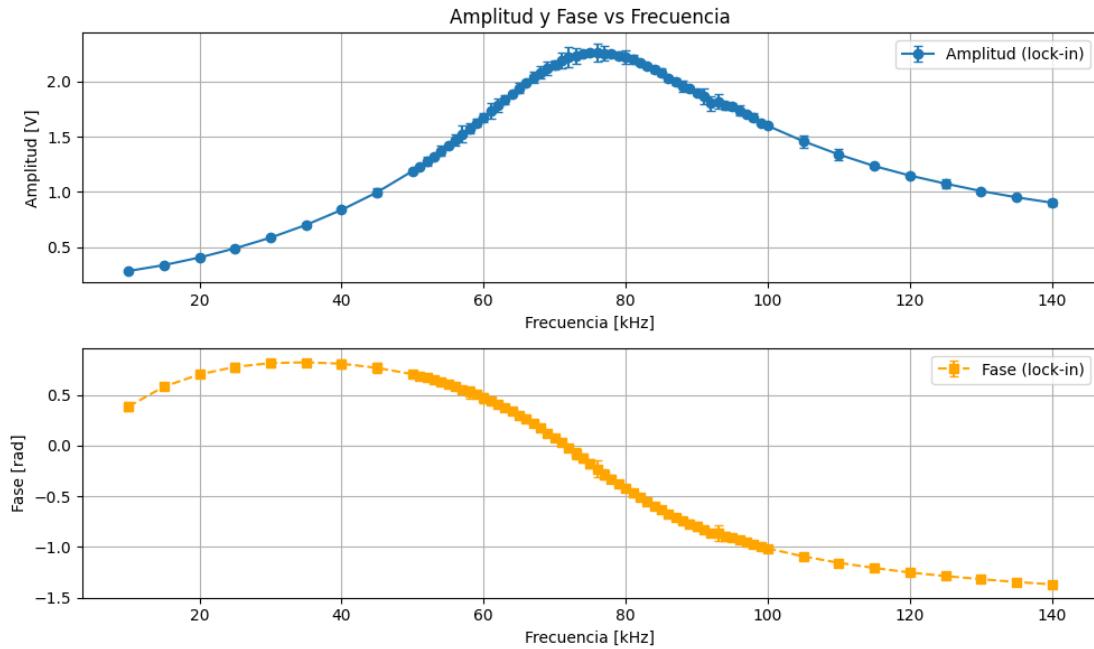


Figura 53: Barrido de frecuencia del RLC.

El cómo se procesaron los datos para llegar a estos resultados se discute más adelante en la sección Lock-In Digital. Lo importante es que vemos la campana de resonancia con una frecuencia de unos 75 kHz donde la amplitud es máxima y la fase se hace 0° .

No parece irse a $+\pi/2$ cuando $f \rightarrow 0$ pero podría ser por todos los cables que agregan capacitancias o algo así.

Mediciones de prueba del sensor.

Para terminar el día con el tiempo que quedaba monté el sensor, conectando los alambres de cobre y acero al RLC ya conectado al generador de funciones y a la placa. Usé la cuba chica de 20x20 cm

internos aproximadamente llena con agua destilada hasta una altura de 3.75 cm interna (medida con la regla de metal).



Figura 54: Setup para medir con el sensor y la placa.

Para poder usar el sensor tenemos que usar la nueva frecuencia de resonancia, que debería ser menor a 75 kHz ya que añadimos una capacitancia extra en paralelo al conectar el sensor sumergido. Estaría bueno medir la frecuencia de resonancia fuera del agua para saber cómo varía al sumergirlo.

Repetí el barrido en frecuencias pero tomando solo 5000 puntos, con lo cual hay mayor error, especialmente en la amplitud.

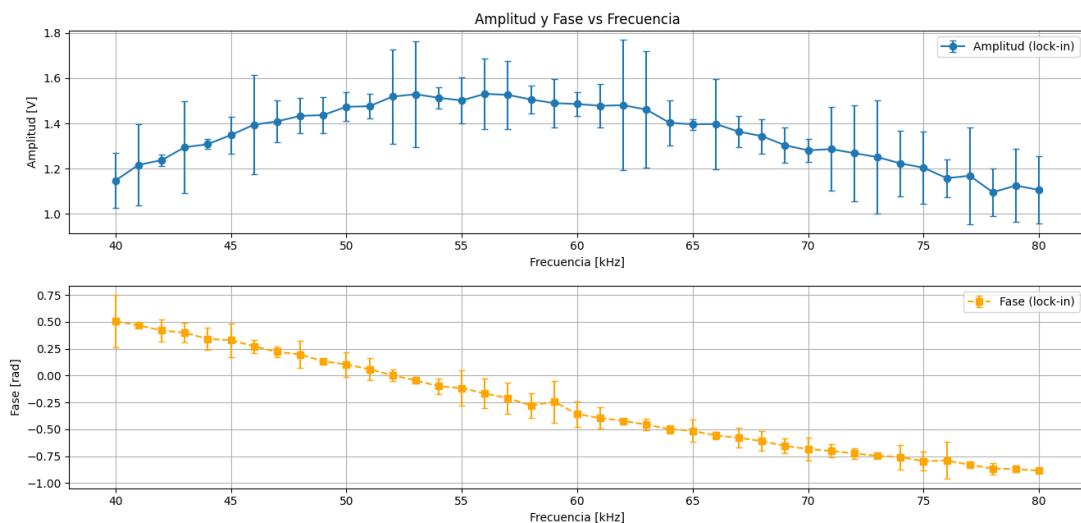


Figura 55: Barrido en frecuencia del RLC conectado con el sensor sumergido.

Por la posición donde la fase se hace 0° estimé que la nueva frecuencia de resonancia es 52 kHz. Así que para las últimas mediciones usé esa. No pude probar si cambiaba la fase al tocar la cuba u otras partes del setup.

Tome varias mediciones de forzado y dejando caer gotas con la regla de metal, pero fueron de muy poca amplitud estas últimas. Un punto importante es que se salió un poco de agua antes de tomar las mediciones finales que voy a discutir más adelante.

5. Lock-In Digital.

Para poder aplicar el Lock-in de forma digital necesitamos los siguientes elementos:

- Señal de referencia.
- Señal de interés con la fase modulada en el tiempo.
- Señal de referencia con shift de 90° .

Todo evaluado en el mismo instante. Nosotros tenemos una pequeña desventaja, por la forma que adquirimos las señales tenemos la señal de referencia sampleada en $t_n=0, 2, 4, \dots \mu s$ y la de interés en $t_m=1, 3, 5, \dots \mu s$.

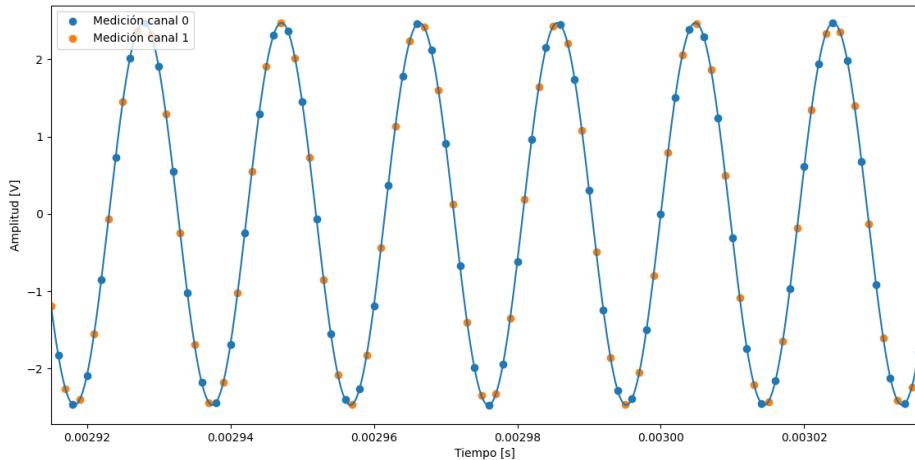


Figura 56: Distintos momentos de sampleo entre canales para una señal de 52 kHz.

O sea, vamos a necesitar una forma de inferir el valor de la señal de referencia en los puntos que se muestrea la señal a analizar, ya sea por interpolación o alguna otra técnica.

Además vamos a necesitar construir el coseno en esos mismos puntos de alguna forma.

Ajuste senoidal.

El primer enfoque para obtener la referencia sampleada en los puntos que necesitamos (t_m) fue ajustar los datos medidos (en t_n) para la referencia por:

$$Q_x(t_n) = A_0 \sin(2\pi f_0 t_n + \phi_0)$$

Con parámetros A_0 , f_0 y ϕ_0 . Luego la referencia resampleada y la señal shifteda 90° van a ser

$$Q_x(t_m) = \text{sgn}(A_0) \sin(2\pi f_0 t_m + \phi_0) \quad y \quad Q_y(t_m) = \text{sgn}(A_0) \cos(2\pi f_0 t_m + \phi_0)$$

El signo es importante ya que sino sería como incorporar una fase de $\pm\pi$ aleatoriamente según si el ajuste da positivo o negativo, y esto daba lugar a tener que hacer unwrappings con periodo π innecesarios.

Para pocos puntos este método funcionaba bastante bien, pero para las medidas de prueba con aproximadamente 13 millones de puntos por canal (para unos 25 segundos) ya no tanto. Los principales problemas de este método son los siguientes:

- Durante mucho tiempo la señal de referencia no es perfectamente estable y va variando lentamente en el tiempo, lo que introduce ruido sino se tienen en cuenta durante el análisis.
- Si el ajuste da una frecuencia levemente distinta a la real la fase va a tener sumando un término de la forma $\Delta\omega t$ que va a tener montada la fase que nos interesa y habría que hacer un detrend lineal.
- El ajuste para muchos puntos lleva mucho tiempo.
- Traté de limitar los parámetros para que A_0 siempre fuese positiva y ϕ_0 vaya entre $[-\pi, +\pi]$ pero no encontraba parámetros óptimos por algún motivo. Y por defecto “curve_fit” de *scipy* daba fases del orden de 50 radianes.

Para evitar estos problemas traté de analizar los datos de a ventanas que se solapen, quedándome con la última mitad de los datos, ya que el lock-in siempre hace cosas raras con los primeros puntos. Pero hacia los últimos datos siguen apareciendo las tendencias lineales en la referencia y no estoy muy seguro porqué (incluso analizando referencia contra referencia). No sé si serán los últimos datos antes de detener la medición o si a partir de 20 segundos de medir pasa algo, como se ve más abajo (para medición de forzado diagonal del tanque y relajación, después de que se salió un poco de agua).

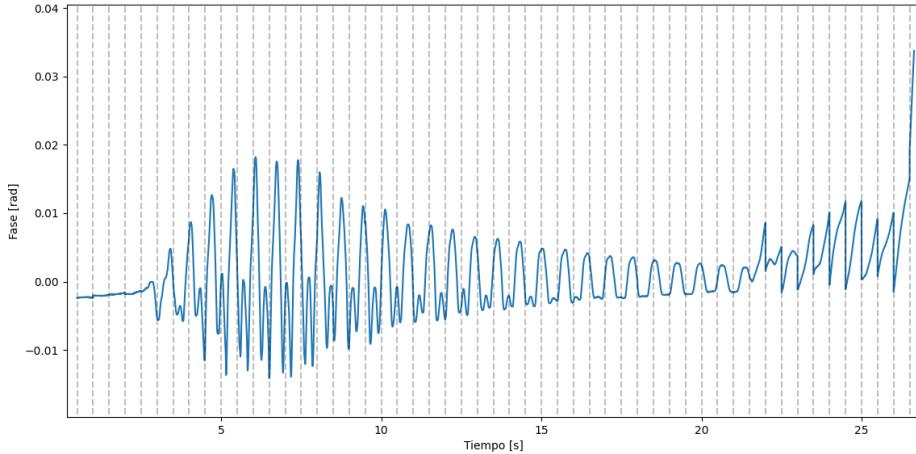


Figura 57: Método de reconstrucción de la señal de referencia por ajuste de cuadrados mínimos en intervalos con overlap de la señal.

Por otro lado haciendo el ajuste en la señal completa imagino que se compensan los errores en la señal de referencia pero igual queda un pequeño ruido que hay que restar a posteriori (por ahí si la amplitud de la fase fuese mayor no se notaría tanto).

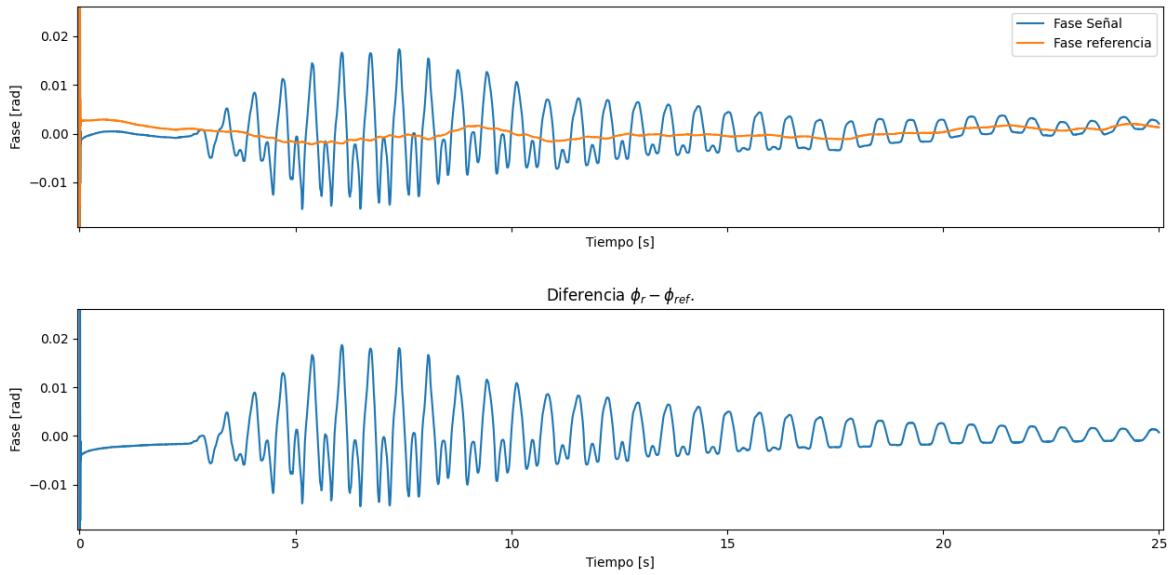


Figura 58: Lock in de la medición completa con reconstrucción de la referencia por Ajuste de cuadrados mínimos.

Con esto queda funcionando el método, pero hay que hacer dos veces el análisis (una para la señal y otro para la referencia) lo que no es del todo muy eficiente.

Fractional delay.

Otra opción para reconstruir la referencia es utilizar lo que se conoce como **Fractional Delay Filters**. Estos nos permiten para una señal con un ancho de banda limitado, tal que esa frecuencia máxima sea sampleada regularmente a la frecuencia de Nyquist. [13]

Esto se debe a la aplicación del Teorema de Whittaker-Shannon que nos permite reconstruir ese tipo de señales completamente. Supongamos que tenemos $Q[n]$ sampleada en los puntos t_n . Entonces la señal completa va a ser:

$$Q(t) = \sum_{n=-\infty}^{+\infty} Q[n] \text{sinc}\left(\frac{t - nT}{T}\right) \quad \text{con} \quad \text{sinc}(t) = \frac{\sin(\pi t)}{\pi t} \quad (93)$$

Con $T = 1/f_s$. Por simplicidad consideremos $T = 1$ ya que vamos a agregar un delay no cambia. Luego si queremos retrasar la señal una fracción $j = \tau/T$, entonces vamos a tener la señal evaluada en $t = t_m - \tau$ o $t/T = m - j$

$$Q[m - j] = \sum_{n=-\infty}^{+\infty} Q[n] \text{sinc}(m - j - n) \quad (94)$$

Y si recordamos la definición de la convolución discreta:

$$Q[m - j] = (Q * h)[m] = \sum_{n=-\infty}^{+\infty} Q[n] h[m - n] \quad \text{con} \quad h[n] = \text{sinc}(n - j) \quad (95)$$

Acá sin embargo tenemos una sumatoria infinita y el $\text{sinc}(x)$ no decae a 0, no tiene soporte finito, y la cantidad de datos que podemos medir son limitados. Así que para solucionar esto usamos una ventana que lo lleve a 0 en cierta cantidad de puntos, cuantos más usemos mejor funcionará ya que más términos tiene la sumatoria.

En python queda:

```
def resample(ts, signal):
    fs = 1/(ts[1] - ts[1-1])
    t0 = -0.5
    h = fractional_delay(t0)
    delayed = convolve(signal, h, mode="same")
    new_ts = ts - t0/fs
    return new_ts, delayed

def shift_90(ts, signal):
    fs = 1/(ts[1] - ts[1-1])

    Ar, fr = fft_parameters(ts, signal)

    Delta = -1/4 * fs/fr
    h_cos = fractional_delay(Delta)
    cos_measured = convolve(signal, h_cos, mode="same")
    return cos_measured
```

Donde obtenemos los coeficientes del FDF con la librería de *pyroomacoustics*:

```
def fractional_delay(t0):
    N = constants.get("frac_delay_length")

    if isinstance(t0, np.ndarray) and t0.ndim == 1:
        t0 = t0[:, None]

    return np.hanning(N) * np.sinc(np.arange(N) - (N - 1) / 2 - t0)
```

El $N=81$ por defecto y usa la ventana Hanning. Parece estar funcionando bastante bien. Para el seno hay que retrasar medio punto de sampleo y a parte correr los tiempos. Para el coseno hay que retrasar un tiempo que de una fase de 90° :

$$2\pi f \Delta t = \pi/2 \Rightarrow \Delta t = j/f_s = 1/(4f) \Rightarrow j = f_s/(4f) \quad (96)$$

Y luego de retrasarla 90° resamplearla para que esté en los mismos puntos que la señal original de interés. Ahora que lo pienso se podría retrasar 90° directamente el seno resampleado.

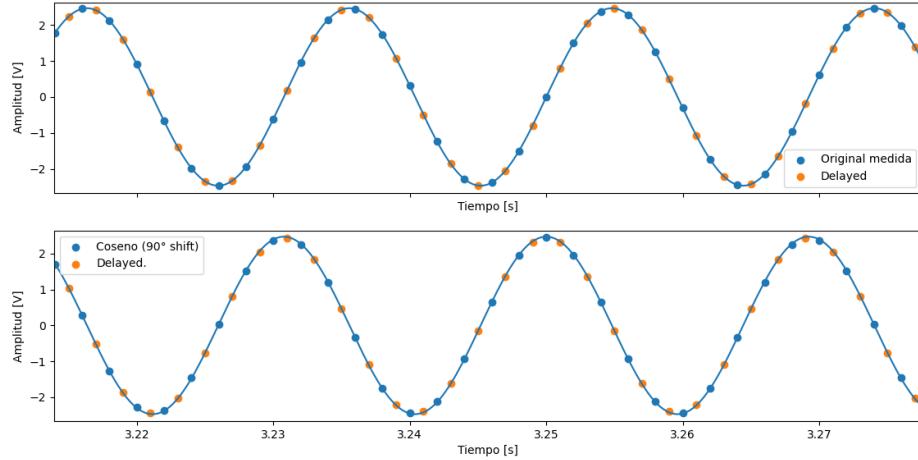


Figura 59: Resampleo de una señal artificial y shifteo de 90° con los FDF.

Con esta reconstrucción de las señales de referencia se puede hacer el Lockin a toda la señal completa (13 millones de puntos) y es bastante más rápido.

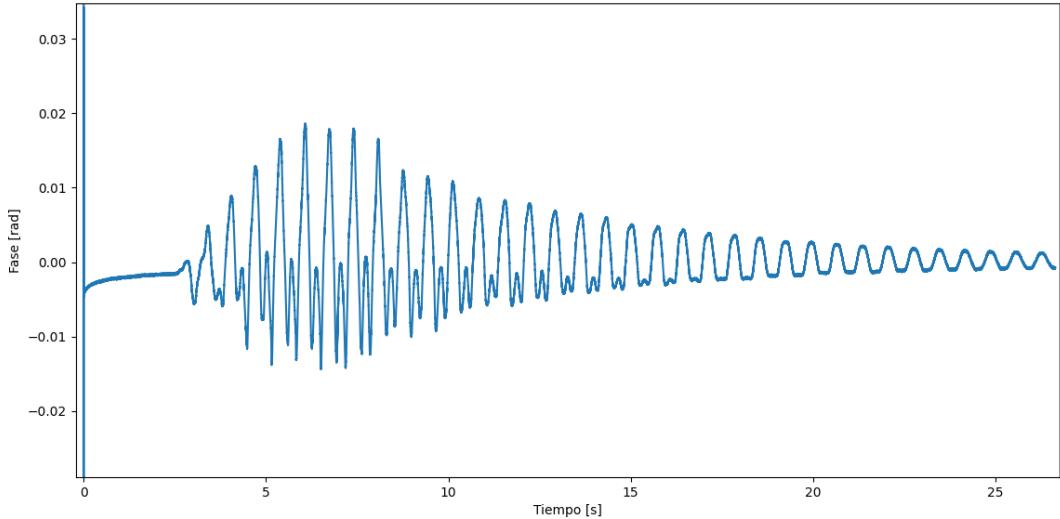


Figura 60: Señal completa analizado con el lock in y FDF.

Y acá además no hace falta analizar la referencia por separado porque se usa la verdadera punto a punto y no una artificial como antes.

Además tarda para analizar 50000 puntos (0.1 s a 500 kHz de sampleo) en 0.015 s con lo que se podría usar a tiempo real sampleando un poco menos, por ejemplo, durante 0.085 s, y tener 10 puntos por segundo.

Filtro exponencial.

Por último, vine usando hasta ahora el butter low pass filter para el lock-in, pero en [14] usan uno que se conoce como filtro exponencial o exponential moving average, tal vez más conveniente para tiempo real, ya que solo depende de los puntos anteriores. Si tenemos nuestros valores en un determinado momento:

$$X_0[n] = A[n]C[n] \quad \text{y} \quad Y_0[n] = A[n]S[n] \quad (97)$$

Con $A[n]$ la señal de interés con la modulación, $S[n]$ la referencia (para esto el seno) y $C[n]$ la shifteada 90° , el coseno. Luego filtramos en dos etapas, usando el parámetro $\alpha = \cos(\gamma) - 1 + \sqrt{\cos^2(\gamma) - 4\cos(\gamma) + 3}$ que se relaciona con la frecuencia de corte o el tiempo de integración τ de la forma: $\gamma = 2\pi f_c/f_s$ con f_s la frecuencia de sampleo. Entonces:

$$X_1[n] = X_1[n-1] + \alpha(X_0[n] - X_1[n-1]) \quad (98)$$

$$Y_1[n] = Y_1[n-1] + \alpha(Y_0[n] - Y_1[n-1]) \quad (99)$$

$$(100)$$

y la segunda etapa:

$$X_2[n] = X_2[n-1] + \alpha(X_1[n] - X_2[n-1]) \quad (101)$$

$$Y_2[n] = Y_2[n-1] + \alpha(Y_1[n] - Y_2[n-1]) \quad (102)$$

$$(103)$$

Y podemos elegir trivialmente $X_0[-1] = Y_0[-1] = 0$. Con este para la señal anterior y el FDF tenemos:

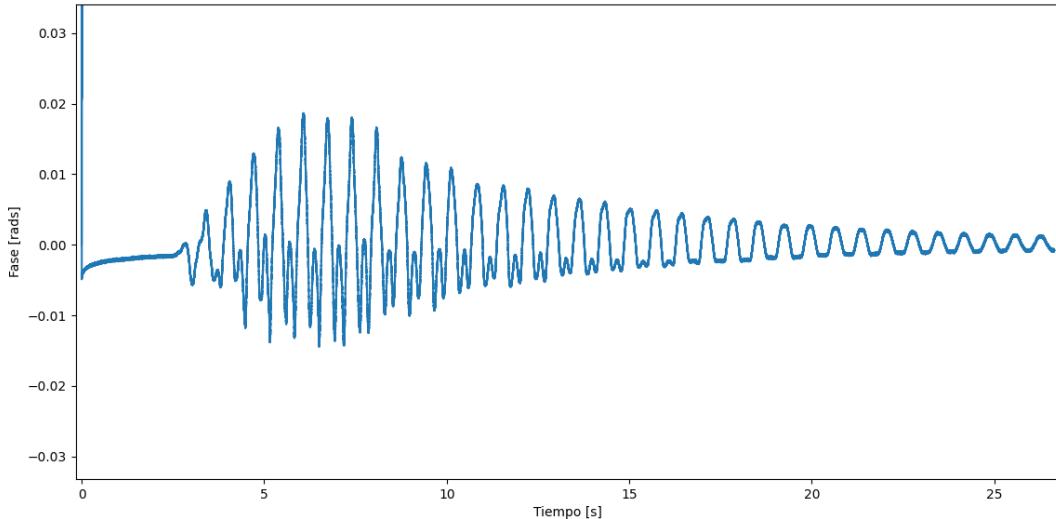


Figura 61: Filtro exponencial + FDF de la señal completa.

También funciona bastante bien, pero para el array ya entero funciona un poco más lento que el butter low pass de antes. Para tiempo real no lo sé todavía.

Con el ajuste de nuevo habría que restar la fase de la referencia original.

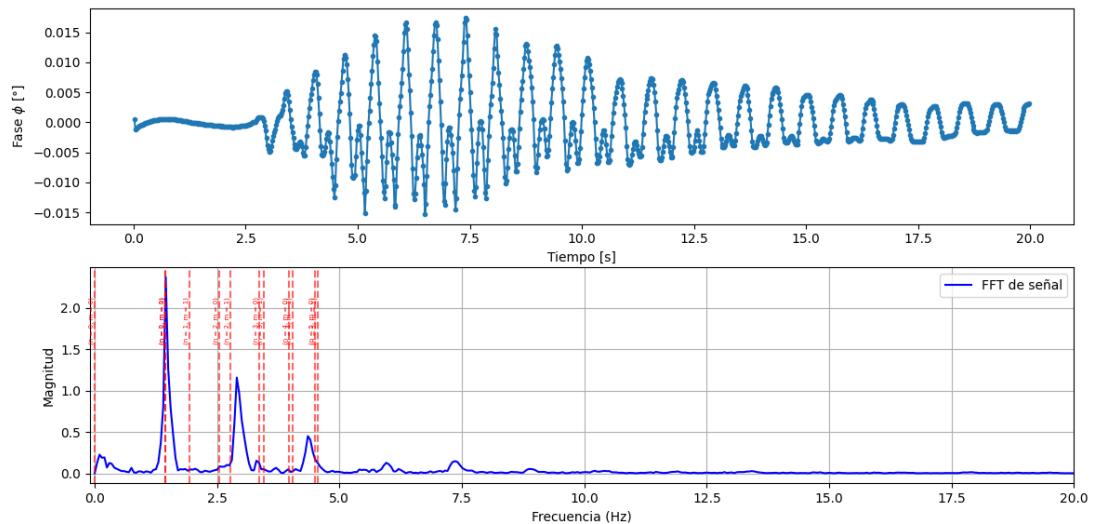


Figura 62: Análisis espectral forzado y relajación.

Análisis espectral.

Por último el mismo análisis espectral que ya hicimos para el SR830.

De nuevo la fundamental es la más excitada pero después aparecen múltiplos de la misma y no sé del todo porqué, por ahí las no linealidades del sistema afectan acá.

Semana de 16/06/2025 .

Bibliografía

- [1] Gregory Falkovich. Turbulence. In Alwyn Scott, editor, *Encyclopedia of Nonlinear Science*. Routledge, New York, May 2006.
- [2] Sergey Nazarenko. *Wave Turbulence*, volume 825 of *Lecture Notes in Physics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [3] V. E. Zakharov, Victor S. L'vov, and Gregory Falkovich. *Kolmogorov Spectra of Turbulence I*. Springer Series in Nonlinear Dynamics. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.
- [4] Chiang C Mei, Michael Aharon Stiassnie, and Dick K-P Yue. *Theory and Applications of Ocean Surface Waves*, volume Volume 42 of *Advanced Series on Ocean Engineering*. World Scientific, June 2016.
- [5] Eric Falcon. Laboratory experiments on wave turbulence. *Discrete & Continuous Dynamical Systems - B*, 13(4):819–840, 2010.
- [6] K. Hasselmann. On the non-linear energy transfer in a gravity-wave spectrum Part 1. General theory. *Journal of Fluid Mechanics*, 12(04):481, April 1962.
- [7] Luc Deike. *Etudes expérimentales et numériques de la turbulence d'ondes de surface*. Doctoral Thesis, Université Paris Diderot - Paris 7, École Doctorale Matière Condensée et Interfaces, Laboratoire Matière et Systèmes Complexe, París, Francia, September 2013.
- [8] Pijush K. Kundu. *Fluid Mechanics*. Elsevier Science & Technology, San Diego, 5th ed edition, 2014.
- [9] Eric Falcon and Nicolas Mordant. Experiments in Surface Gravity–Capillary Wave Turbulence. *Annual Review of Fluid Mechanics*, 54(1):1–25, January 2022.
- [10] V. E. Zakharov and N. N. Filonenko. Energy Spectrum for Stochastic Oscillations of the Surface of a Liquid. *Soviet Physics Doklady*, 11(10):881, April 1967.
- [11] V. E. Zakharov and N. N. Filonenko. Weak turbulence of capillary waves. *Journal of Applied Mechanics and Technical Physics*, 8(5):37–40, 1971.
- [12] Gordillo Zavaleta. *Non-Propagating Hydrodynamic Solitons in a Quasi-One Dimensional Free Surface Subject to Vertical Vibrations*. Doctoral Thesis, Universidad de Chile, Facultad de Ciencias Físicas y Matemáticas, Departamento de Física, Santiago de Chile, Chile, 2012.
- [13] Tom Roelandts. How to Create a Fractional-Delay Filter. <https://tomroelandts.com/articles/how-to-create-a-fractional-delay-filter>, June 2019.
- [14] Andrew J. Harvie and John C. De Mello. OLIA: An open-source digital lock-in amplifier. *Frontiers in Sensors*, 4:1102176, March 2023.