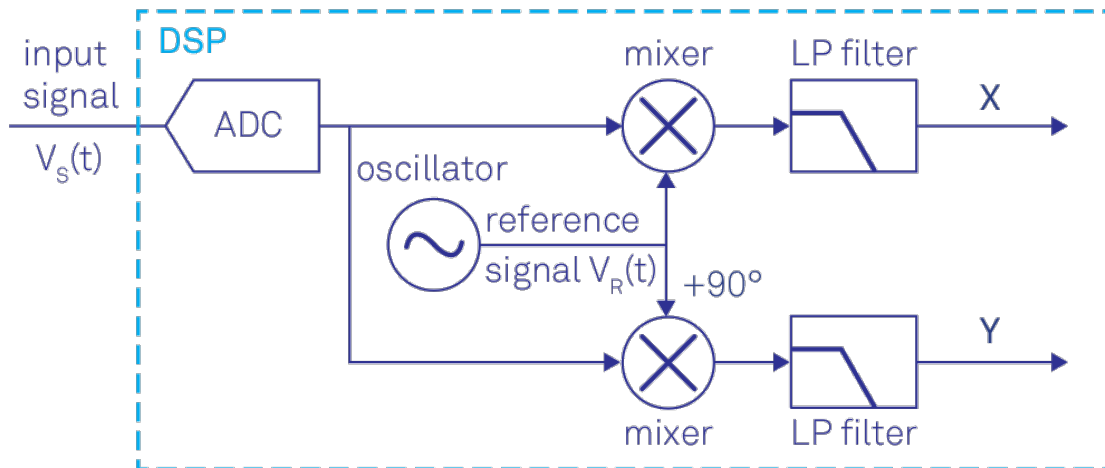


Implementación de un amplificador Lock-in

En esta práctica tienen que implementar un Lock-in Amplifier, simular y caracterizar cada una de sus partes. Para entender cómo funciona el Lock-in Amplifier se puede leer la referencia [1].

Un esquema básico de un Lockin se puede ver en la siguiente figura.



Lo que se pide en esta práctica es que implementen filtros pasa bajos "LP filter" a los cuales se les pueda modificar la frecuencia de corte y el orden del filtro, y que caractericen su comportamiento. Como mínimo tienen que observar el comportamiento de la salida de los mismos ante una señal tipo escalón a la entrada. Además, sería útil para discutir los resultados tener el diagrama tipo Bode de los mismos.

Como primera instancia se pide que la señal de entrada V_s sea la suma de ruido blanco, de 50 Hz y la señal a frecuencia conocida de la cual tenemos que obtener su amplitud y fase. Se tiene que poder cambiar la relación de amplitudes de las tres componentes de la señal y la frecuencia de la señal que estamos midiendo. Como en este punto estamos intentando simular la señal que estamos midiendo en un experimento hay que tener en cuenta que tenemos una frecuencia de medición de datos, es decir que la señal de entrada está muestreada con un cierto espaciado temporal.

En general, un Lockin cuenta con filtros no pasa banda para la señal de línea, en nuestro caso 50 Hz. Esto se utiliza porque el ADC de la figura tiene una cierta resolución, entonces si uno deja pasar los 50 Hz la relación señal ruido no es óptima al digitalizar los datos. En cambio, si uno filtra los 50 Hz es posible mejorar la relación señal ruido al digitalizar los datos. Para simular y probar esto, se pide que implementen este tipo de filtro y comparen las señales que le llegan al ADC con y sin el filtro. En un Lockin esto se relaciona con la ganancia dinámica del mismo.

Por último, tienen que integrar todo para mostrar el comportamiento de un Lockin y analizar sus resultados cambiando las señales de entrada como así también los distintos filtros.

Tienen la libertad de realizar la implementación en el lenguaje y de la forma que les sea más cómoda.

Para los que quieran les dejamos una posibilidad de cómo encarar el problema. Para esto se puede utilizar Python y generar tres clases que luego se integran para formar el lockin.

1: La primera es la clase que genera la señal, la cual tiene la función que la inicializa de acuerdo a los parámetros de trabajo y una segunda función que devuelve un valor de la señal cada vez que es llamada.

```
class genSenial():

    def __init__(self,FrecSampleo, FrecMed, AmpMed, Amp50, AmpBlanco):
.....
....

    def Next(self):

...
...
...

    return tiempo, Vref, Vref90, Vs
```

2: La segunda clase es la que genera los filtros. De forma similar cuenta con la inicialización y la salida del filtro cada vez que se la llama con un nuevo dato.

```
class filtro:

    def __init__(self, frecuenciadecorte,velocidaddesampleo,ordendelfiltro,tipe='low' ):
.....
.....

    def newelement(self,xn): #Es esta la funcion que se llama cuando se ingresa un
nuevo dato, y maneja los vectores para luego llamar a filter. También termina devolviendo la
salida del filtro

...
...
```

```
return Vfiltrada
```

```
def filter(self):  
.....  
return temp
```

Los filtros que pueden implementar son filtros IIR [2], y las constante del filtro se pueden obtener con la librería Scipy que les devuelve los valores de las constantes a,b. En particular pueden utilizar un filtro tipo Butterwoth debido a que los mismos tienen la respuesta más plana en frecuencias [3]

3: La última clase que deberían crear es la Lockin. Y de forma similar a las anteriores puede contener:

```
class Lockin:
```

```
    def __init__(self,  
amplitud,fase,frecuencia,frecuenciadecorte,velocidaddesamplio,ordendelfiltro):  
    ...  
    ....  
    ....
```

```
    def calcular(self,Vref, Vref90, Vs):
```

```
    .....  
    .....  
    .....
```

```
    return Vn,R,Theta #Vn es la salida dell filtro no pasa banda y es solo para luego  
poder comparar y analizar los resultados.
```

Por último tienen que realizar un programa que integre las 3 clases y guarde los archivos con resultados para analizar.

Referencias:

Contacto: Sergio Encina sergioencina@gmail.com

[1] Principles of lock-in detection and the state of the art. Zurich Instruments. 2016. https://www.zhinst.com/sites/default/files/li_primer/zi_whitepaper_principles_of_lock-in_detection.pdf

[2] <https://es.wikipedia.org/wiki/IIR>

[3] <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html>