

# Control y Sistemas

## Trabajo práctico: Filtrado digital IIR

### 1) Filtro Leaking Integrator (LI) en MATLAB:

- a) Genere una señal senoidal con frecuencia fundamental de 100Hz.
- b) Agregue ruido a la señal senoidal tal que la relación señal-ruido entre la señal senoidal y la señal con ruido sea de 15 dB.
- c) Aplique filtrado del tipo *leaking integrator* a la señal con ruido para filtros con dimensión para  $\lambda$  iguales a 0.8, 0.9 y 0.98. Utilice la función `filter()`.
- d) Grafique la respuesta en el tiempo de las señales original y filtradas y compare.
- e) Grafique la respuesta en frecuencia y fase de los 3 filtros LI. Use la función `freqz`.

### 2) El objetivo de este ejercicio es el de comparar las dimensiones de filtros FIR e IIR para un mismo tipo de filtrado.

- a) Ejecute la función `iir_vs_fir.m`.
- b) Analice las funciones `fir_kaiser_3400_44100.m` e `iir_elliptic_3400_44100.m` ¿Qué tipos de filtros implementan ambas funciones?.
- c) Observe ambas respuestas en frecuencia. ¿Qué diferencias hay entre ambas respuestas?
- d) Grafique las respuestas en fase y compárelas. ¿Qué diferencias hay entre ambas respuestas?
- e) ¿Cuál es la dimensión del numerador del filtro FIR y cuántos coeficientes presenta la matriz SOS del filtro IIR? ¿A qué conclusión puede abordar?

### 3) Discretización de un sistema en el dominio del tiempo usando diferentes transformaciones.

- a) Discretice el siguiente sistema  $s = 10 * (s/2 + 1) / (s/10 + 1)$  aplicando las transformadas ZOH y bilineal.

- b) Use como tiempo de muestreo  $T=0.1$  segundos.
- c) Utilice la función `c2d` de MATLAB.

d) Grafique la respuesta al escalón del sistema continuo y los sistemas discretos. ¿A qué conclusión puede llegar?

e) Cambie tiempo de muestreo a  $T=0.001$  segundos y grafique. ¿A qué conclusión puede llegar?

**4)** El objetivo de este ejercicio es el de analizar el desempeño de filtros IIR de 2do orden tipo Direct I y Direct II descritos en lenguaje C bajo el entorno de MATLAB.

MATLAB permite ejecutar funciones desarrolladas en lenguaje C usando una función *wrapper* cuya finalidad es la actuar como interfaz entre MATLAB y C. Este wrapper debe adaptar las variables de entrada y salida entre ambos lenguajes.

a) Abra la función wrapper `iir_matlab_wrapper.c` y analice las diferentes secciones. Al final de la misma verá que se invoca a la función `iir_filter_I_float()`.

b) Las funciones en C que implementan diferentes filtros IIR de 2do orden se encuentran en el archivo `iir_filters.c`, donde hay funciones para filtros IIR tipo Direct I y Direct II, en punto flotante y en punto fijo. Ejecute en consola el comando:

```
>> mex iir_matlab_wrapper.c iir_filters.c
```

La función `mex` es la encargada de compilar los archivos en C bajo MATLAB.

c) Abra en el editor de MATLAB el script `iir_matlab_to_C.m` y ejecútelo. Este se encarga de invocar la función `iir_matlab_wrapper` y de analizar su salida. Concéntrese en comparar las señales de salida para las funciones implementadas en MATLAB y en C.

d) Comente la línea de la función `iir_filter_I_float()`, descomente la línea de la función `iir_filter_II_float()` y vuelva a compilar. Ejecute nuevamente `iir_matlab_to_C.m`. Observe la señal filtrada de salida, ¿detecta alguna diferencia respecto a la salida para el filtrado tipo Direct I?

e) Abra el archivo `iir_filters.c` y analice las funciones que implementan los filtros IIR tipo Direct I y Direct II. Observe cómo la estructura de cada función representa los esquemas de los filtros IIR tipo Direct I y Direct II, respectivamente ¿Qué diferencias presentan estas funciones?

**5)** Utilizando las funciones del ejercicio anterior, escriba las funciones en C para filtrado IIR tipo Direct I y Direct II en precisión punto fijo Q15. Verifique su correcto funcionamiento en MATLAB.

**6)** Escriba funciones en C para implementar filtrado IIR tipo Direct I y Direct II de orden N. Verifique su correcto funcionamiento en MATLAB.