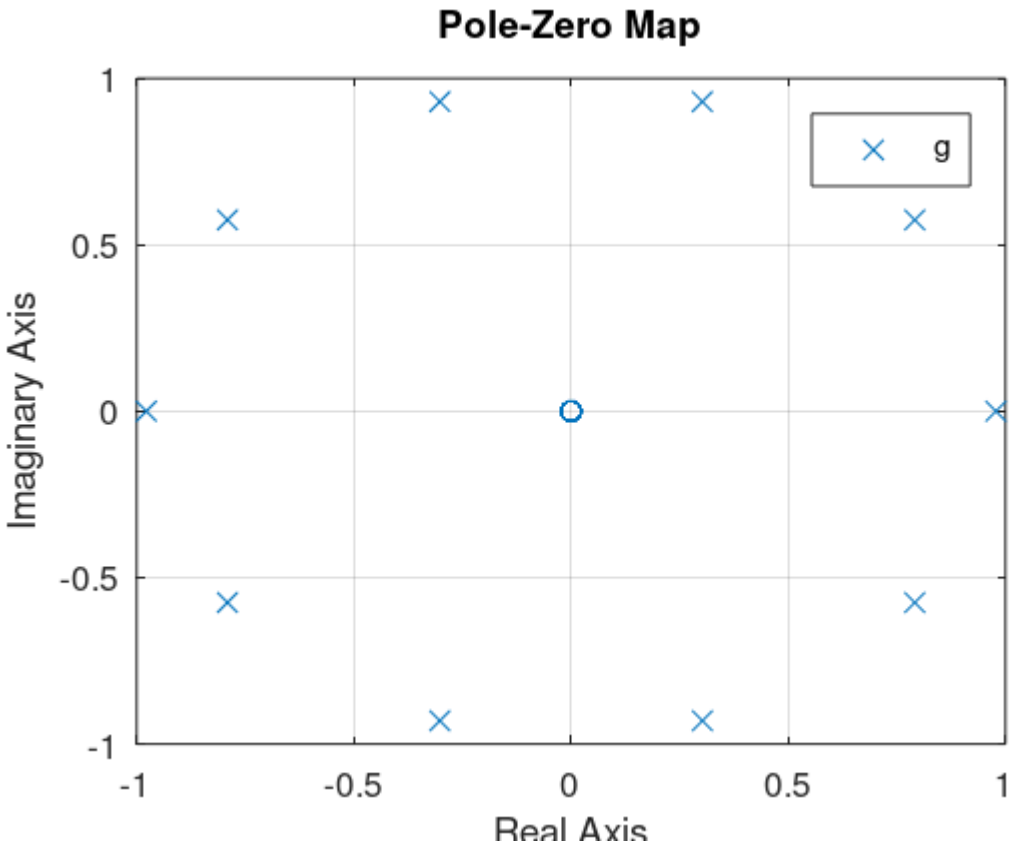
Christopher Ryan Russell Clark
MP6104 PDS
Tarea 4

Sea un sistema digital caracterizado por la función de transferencia:

$$H(z) = \frac{1-\alpha}{1-\alpha z^{-k}}$$

1. Grafique el diagrama de polos y ceros de dicho filtro para $\alpha = 0.8$ y $k = 10$.



2. Encuentre la ecuación de diferencias equivalente para dicho filtro (para todo α y k).

$$H(z) = \frac{1-\alpha}{1-\alpha z^{-k}}$$

$$\left(1-\alpha z^{-k}\right)Y(z) = (1-\alpha)X(z)$$

$$Y(z) - \alpha Y(z) z^{-k} = (1-\alpha)X(z)$$

$$y[n] - \alpha y[n-k] = (1-\alpha)x[n]$$

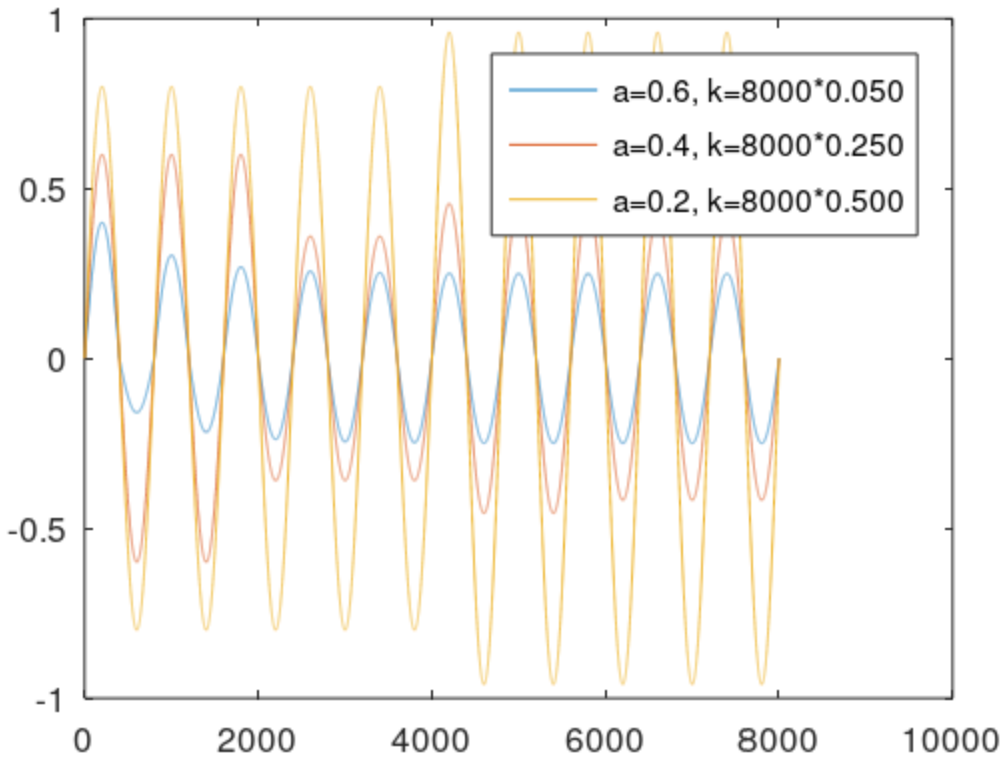
$$y[n] = \alpha y[n-k] + (1-\alpha)x[n]$$

3. Implemente dicho filtro en GNU/Octave (o Python) para los valores:

k	α
$F_s \times 50\ ms$	0.6
$F_s \times 250\ ms$	0.4
$F_s \times 500\ ms$	0.2

y compruebe su implementación con una señal de audio muestreada a la frecuencia de muestreo F_s. Para ello utilice la función filter (o un equivalente en Python). No utilice la convolución directamente.

Prueba con señal de audio sintética de 20 Hz muestreado a 8 kHz.



4. Implemente su filtro en C. Para ello utilice el program de prueba entregado con esta tarea, que debe ser compilado y ejecutado en GNU/Linux. Debe únicamente alterar el archivo tarea04.c (y eventualmente tarea04.h). El programa esta realizado para capturar datos del micrófono si no se entrega ningún argumento de entrada, o del archivo recibido a través del pipe; y pasar a la línea de salida de la tarjeta de sonido los búferes procesados. Todo el proceso de captura y ejecución utiliza Jack; servidor de audio basado en ALSA (Advanced Linux Sound Architecture).

   Junto al código encontrara un README.md con instrucciones sobre como ejecutar la aplicación.

   Todo está implementado en tarea04.c (con una corrección pequeña al nombre del header en jack.c. Incluidos en el push se compilan binarios de los tres ejecutables con las tres configuraciones de filtro solicitadas.