

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica



Filtros y perfilador

Procesamiento Digital de Señales

Integrantes:

Anthony Chaves Achoy

Tobías Fonseca Cruz

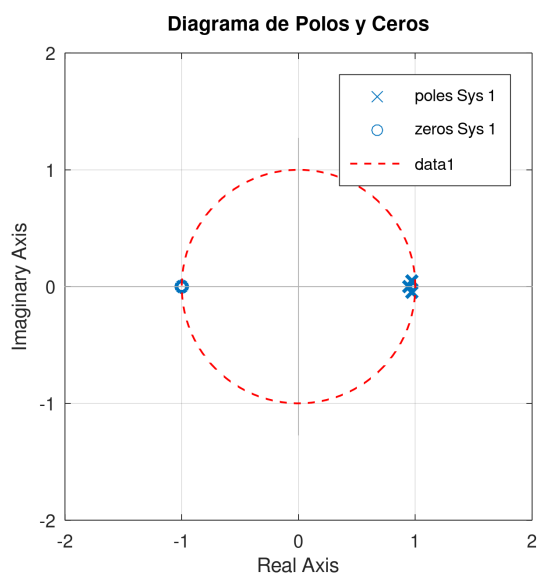
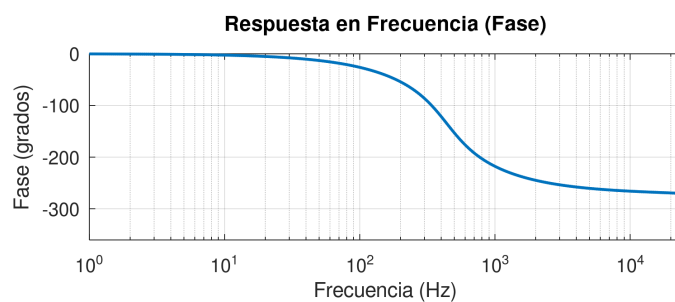
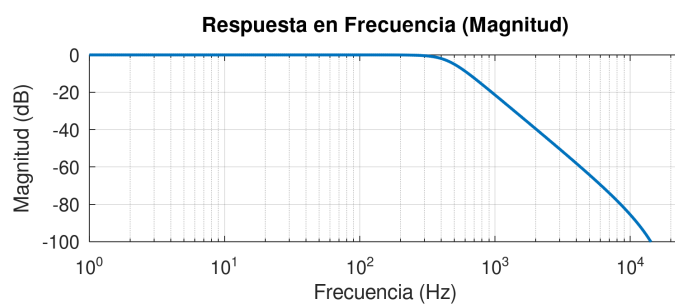
Profesor:

Dr.-Ing. Pablo Alvarado Moya

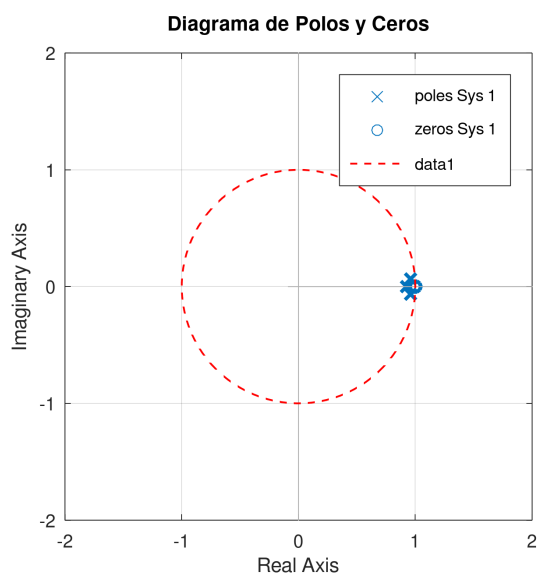
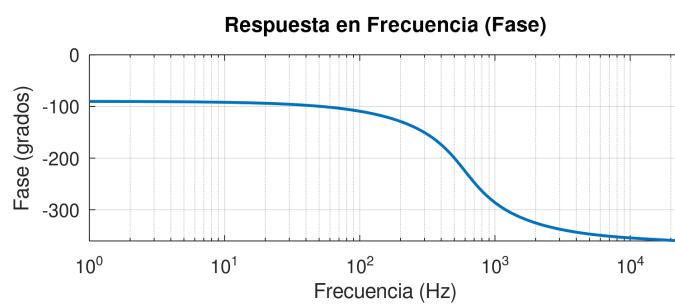
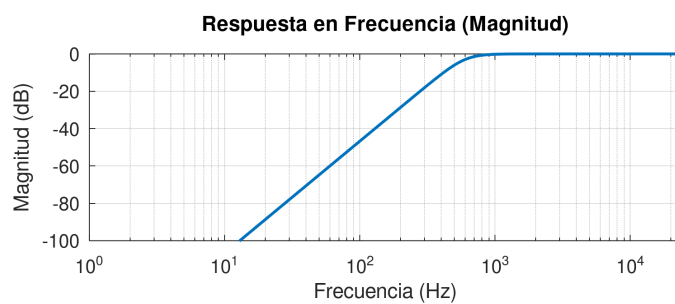
9 de octubre de 2024

Respuesta en frecuencia de los filtros

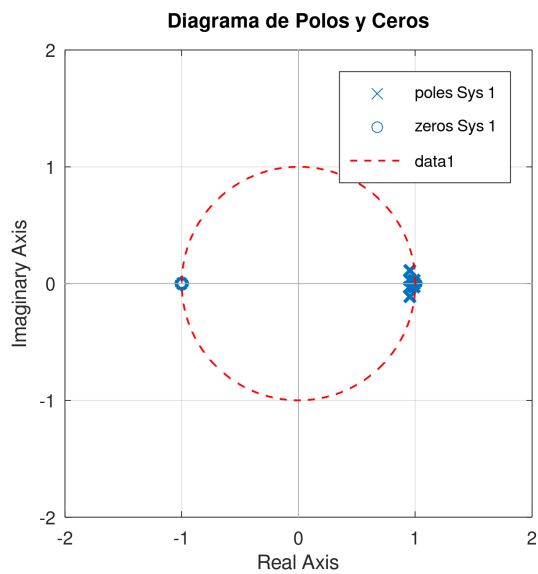
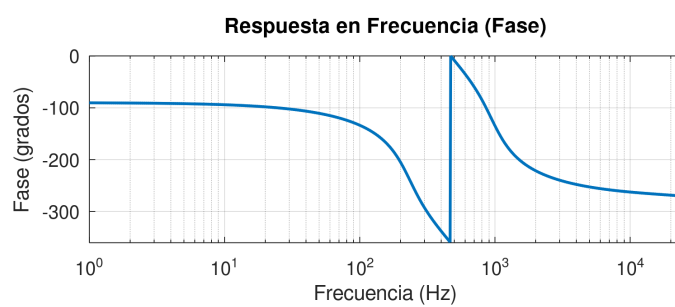
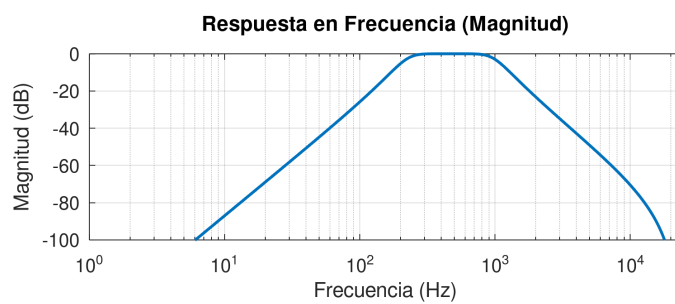
Butterworth LowPass



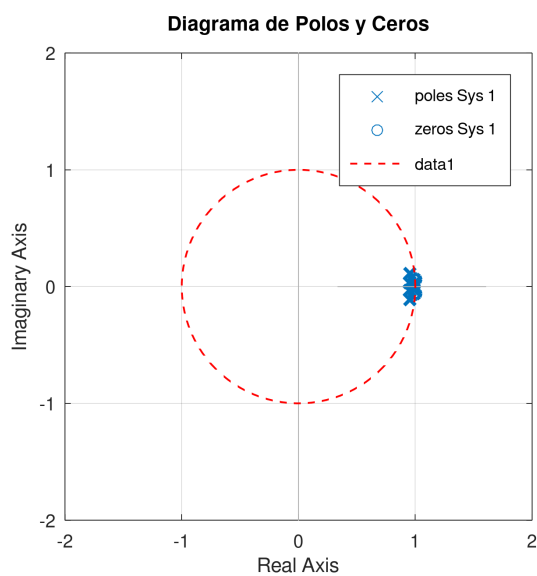
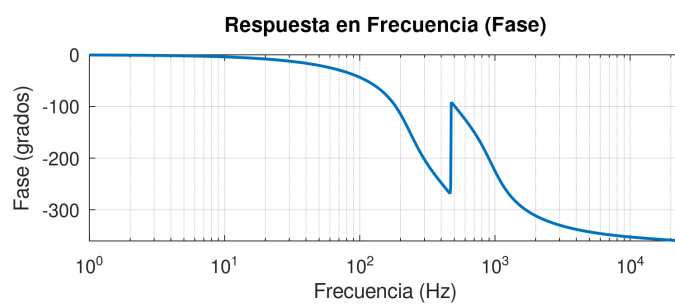
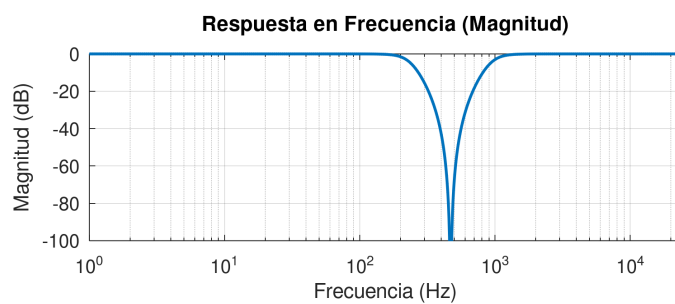
Butterworth HighPass



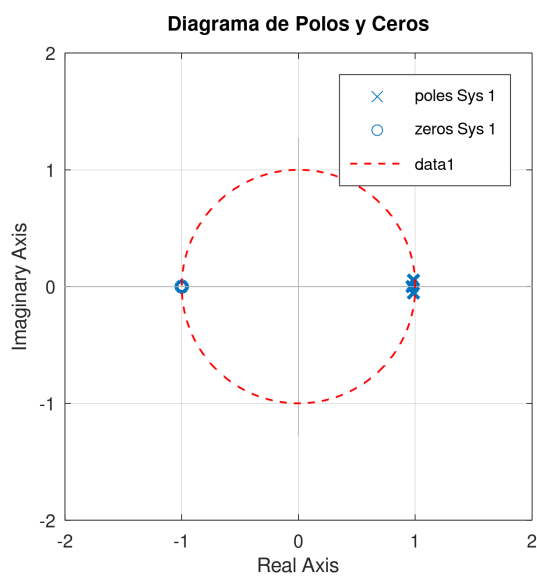
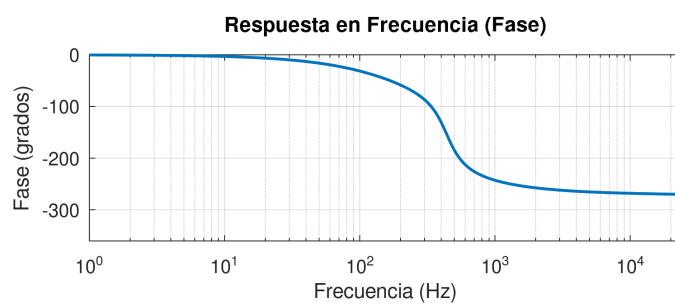
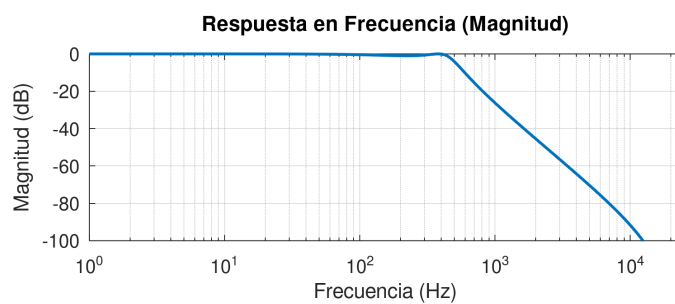
Butterworth BandPass



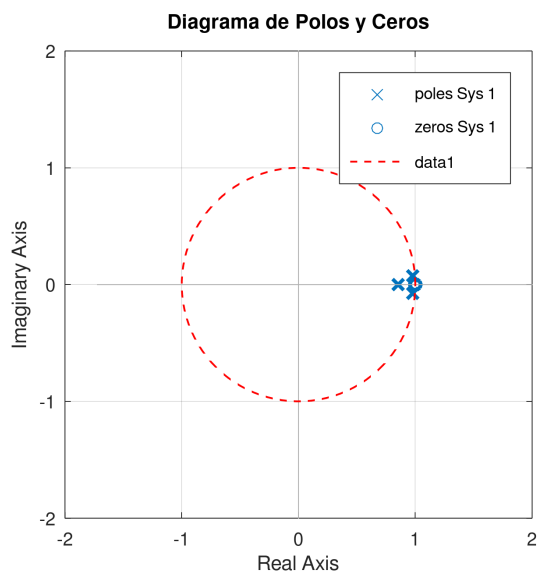
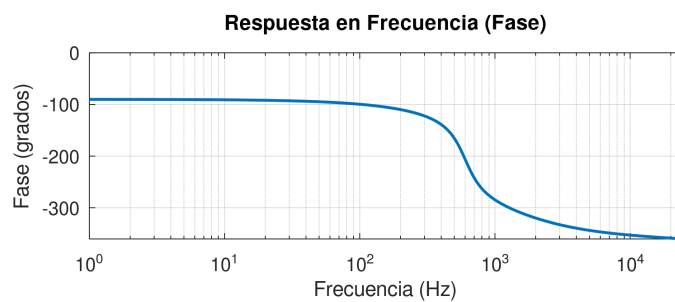
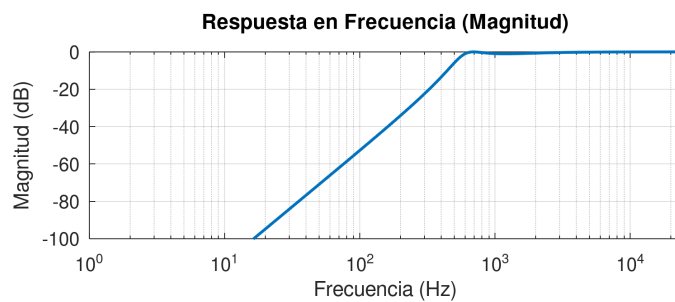
Butterworth StopPass



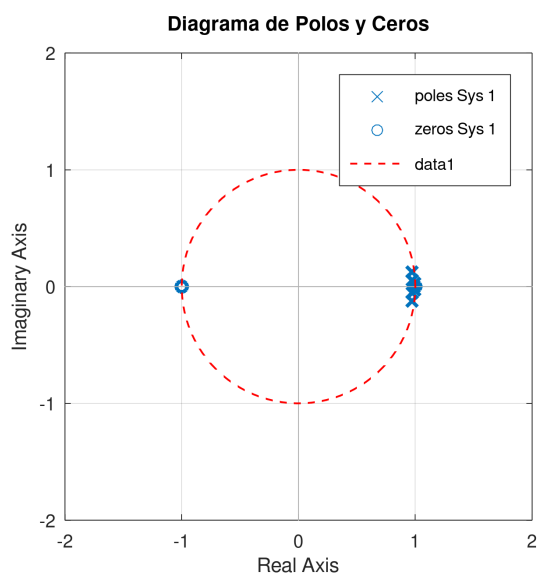
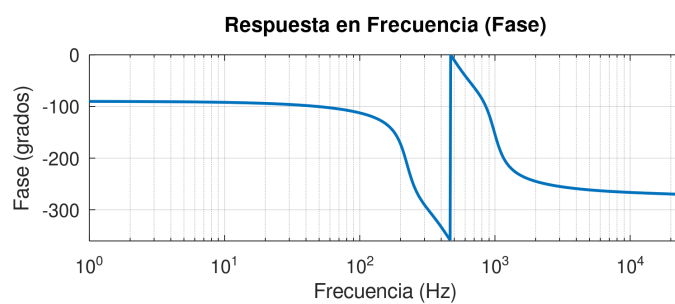
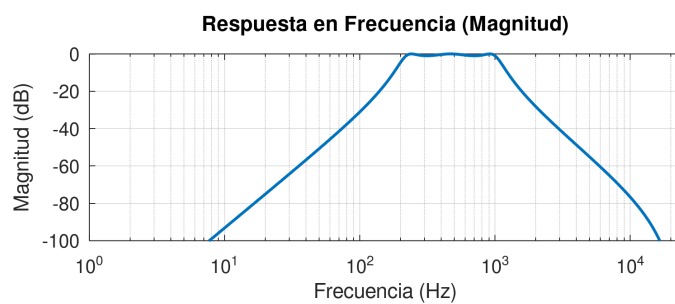
Chebyshev1 LowPass



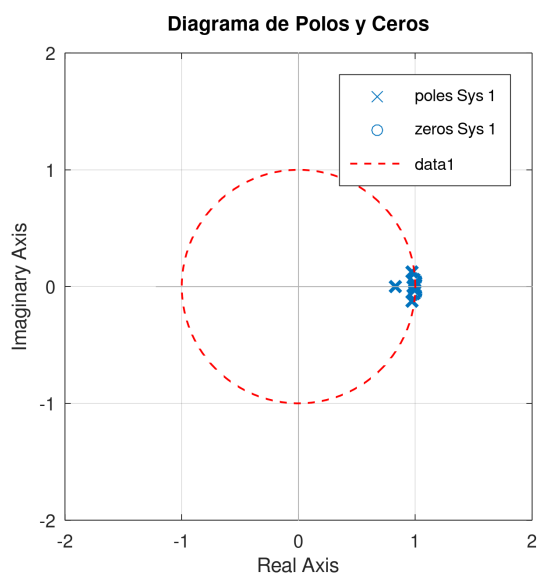
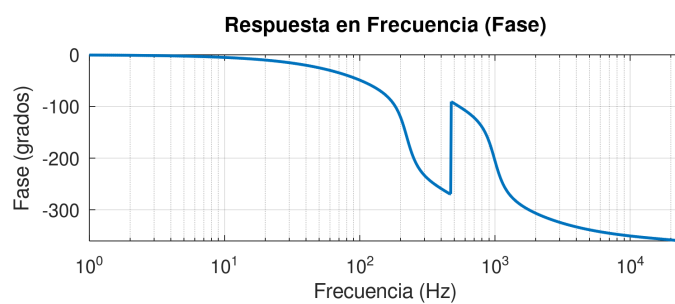
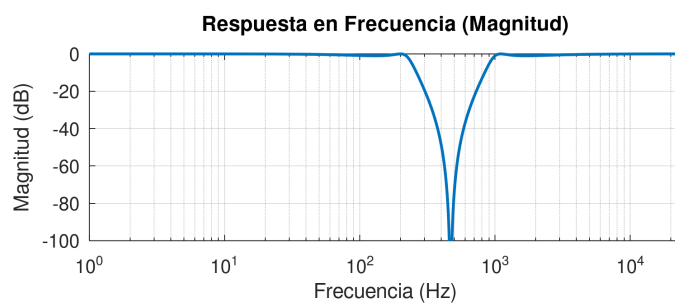
Chebyshev1 HighPass



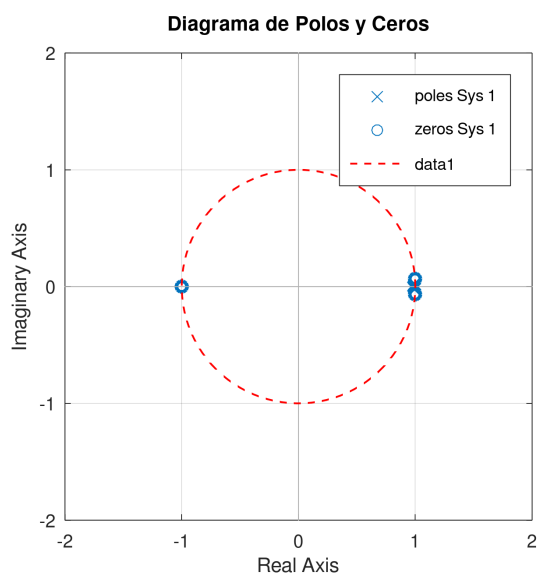
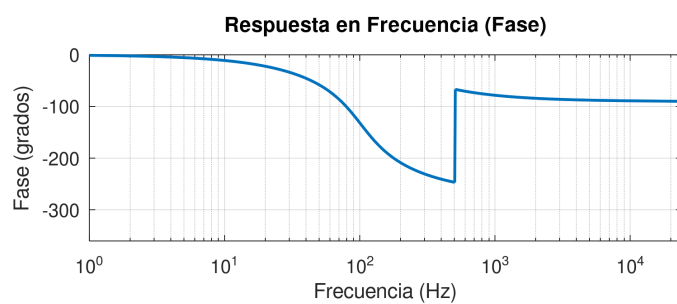
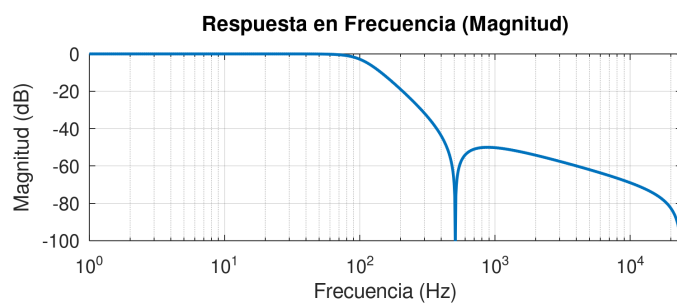
Chebyshev1 BandPass



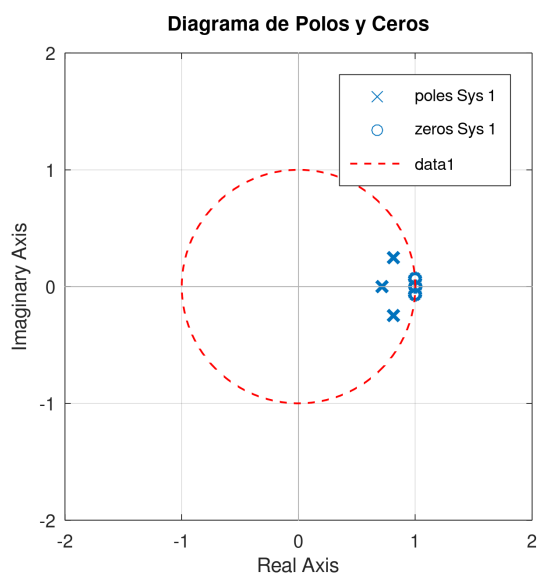
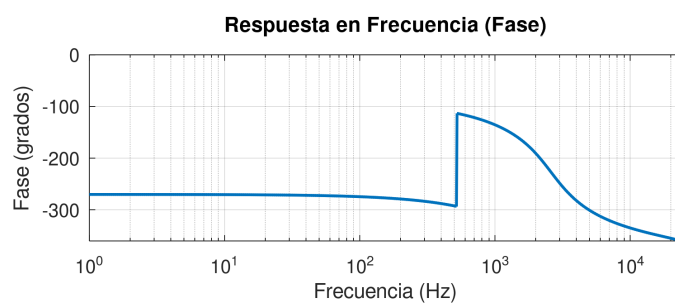
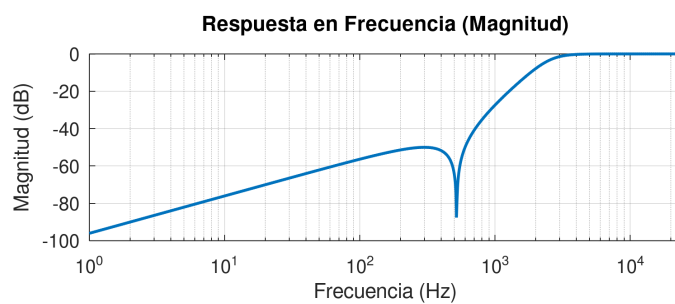
Chebyshev1 StopPass



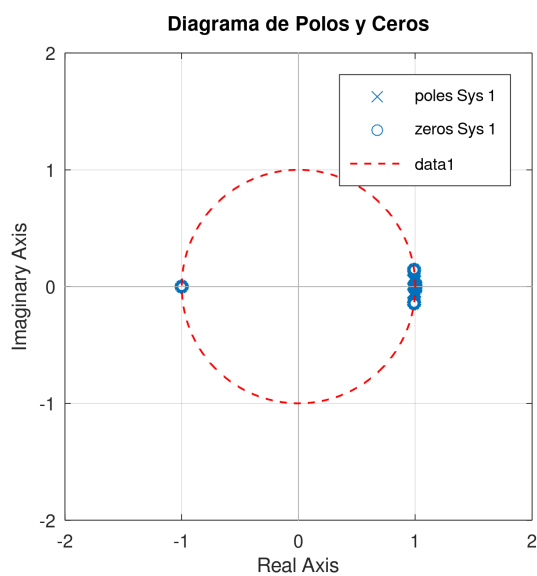
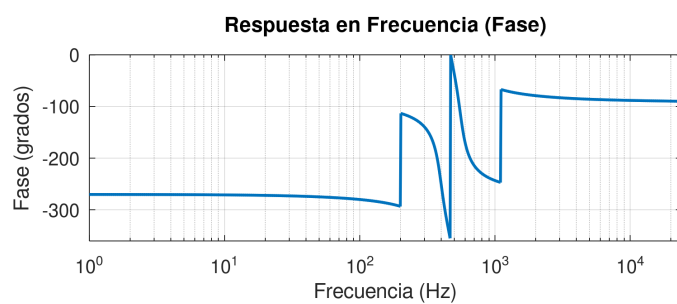
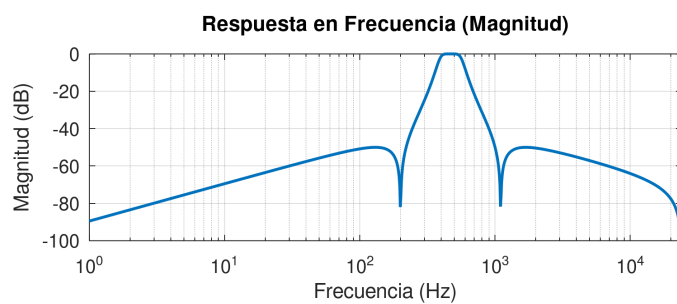
Chebyshev2 LowPass



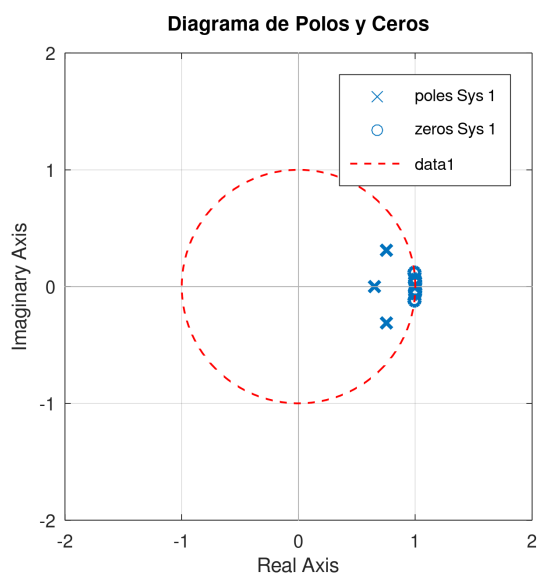
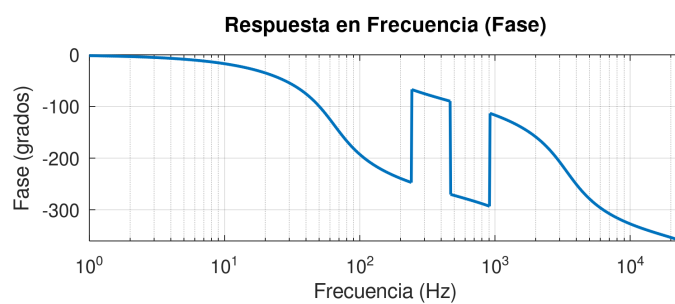
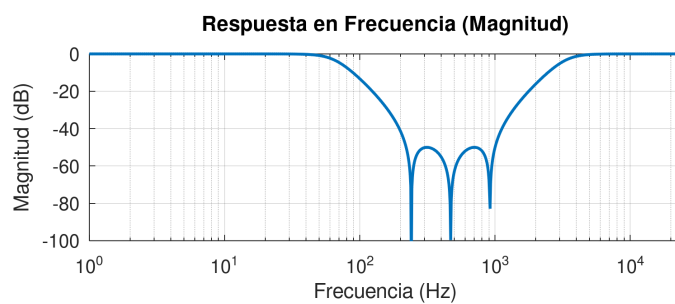
Chebyshev2 HighPass



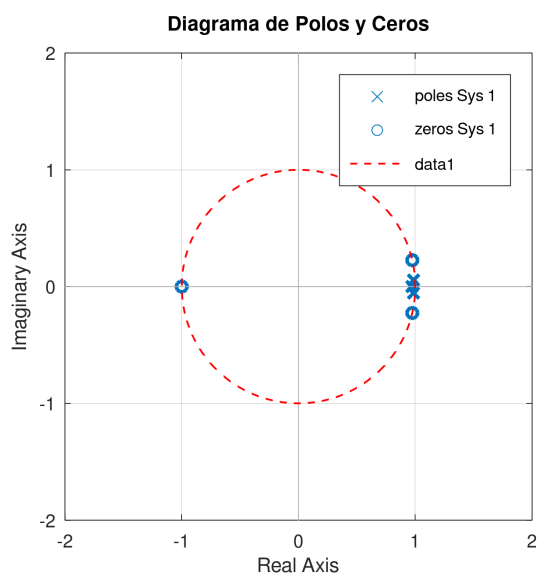
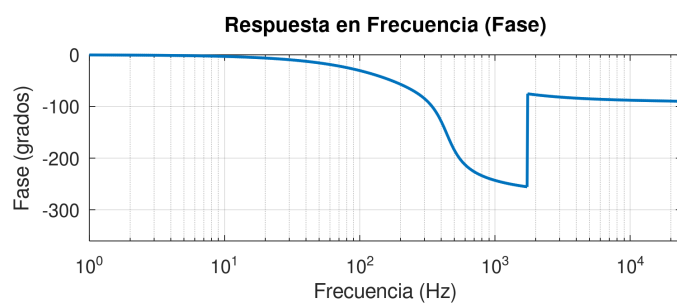
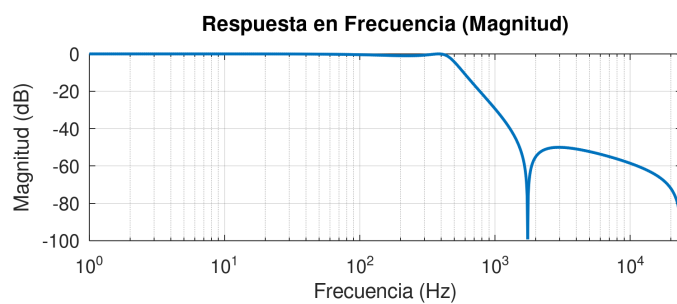
Chebyshev2 BandPass



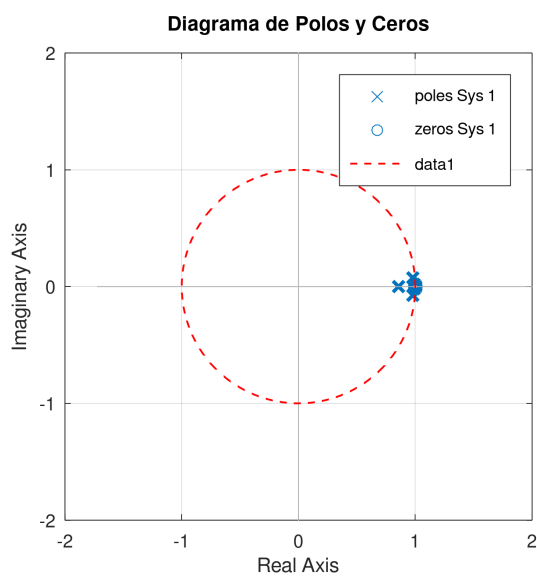
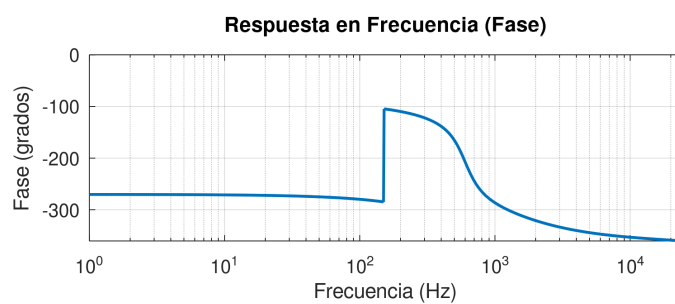
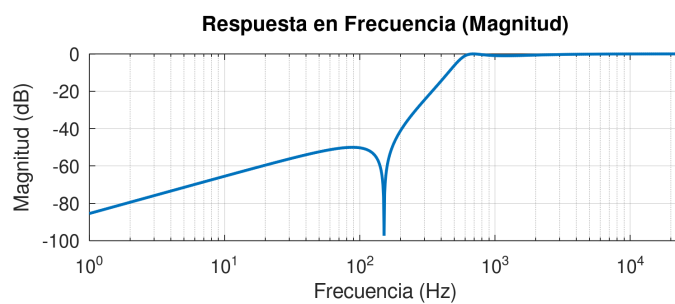
Chebyshev2 StopPass



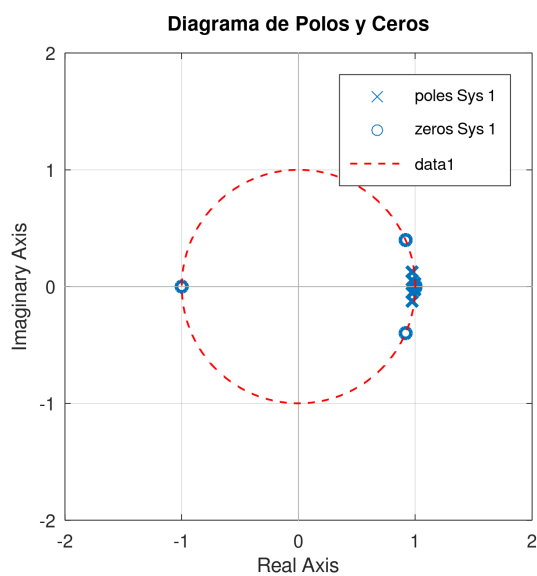
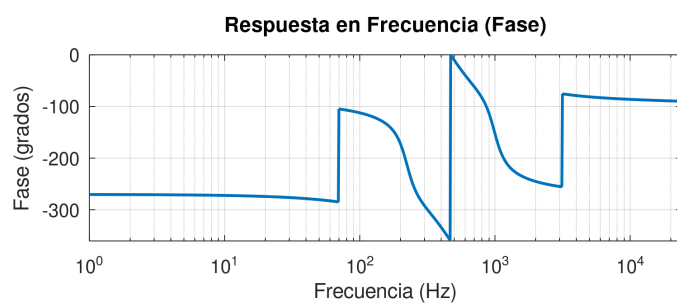
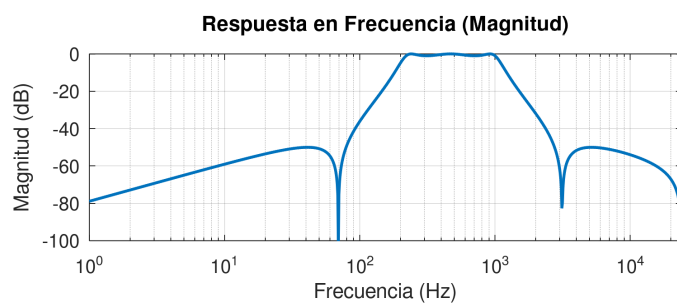
Elliptic LowPass



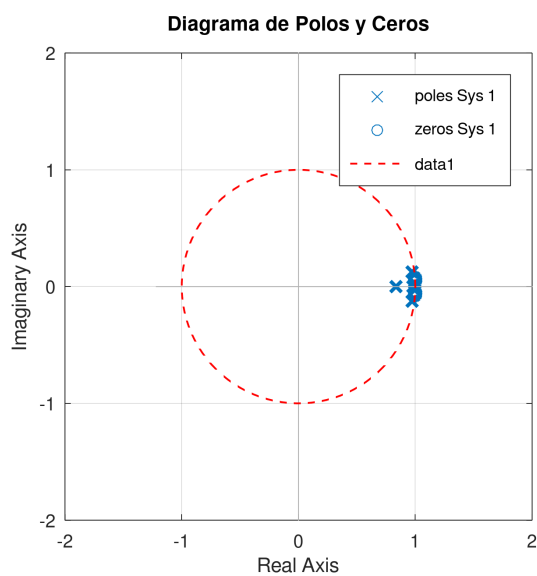
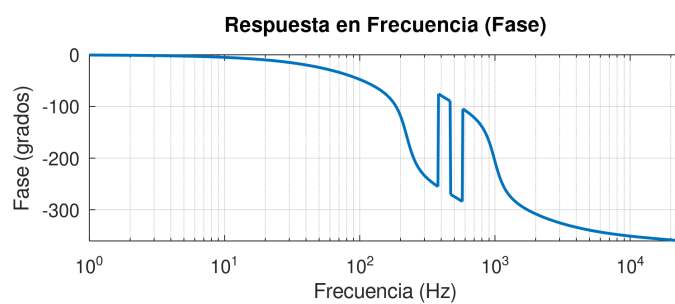
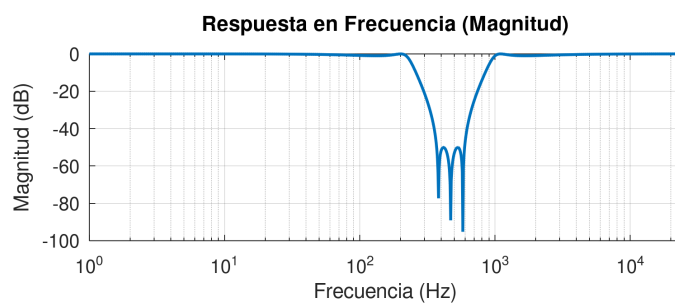
Elliptic HighPass



Elliptic BandPass



Elliptic StopPass



Perfilador

Es muy necesario antes de empezar la sección, identificar que el objetivo primordial es hacer lo más pequeño posible el tiempo de la cascada, ya que el biquad no refleja necesariamente en el process el uso de métodos con lógica avanzada dado que esto implicaría cambiar métodos internos que potencialmente cambiarían la lógica de biquad para algunas pruebas (la lógica vectorial no está integrada directamente con la continuidad de la lógica sin vectores).

En la Figura 1 observamos como un método con la transpuesta, sin loop unrolling y recorriendo totalmente cada buffer de resultados parciales es la solución más lenta.

Benchmark	Time	CPU	Iterations	UserCounters...
BM_Biquad_Process/256	8673 ns	8672 ns	80514	items_per_second=366.632/s
BM_Biquad_Process/512	17677 ns	17676 ns	39596	items_per_second=731.525/s
BM_Biquad_Process/1024	19608 ns	19607 ns	35689	items_per_second=1.46337k/s
BM_Biquad_Process/2048	22062 ns	22061 ns	31727	items_per_second=2.92599k/s
BM_Biquad_Process/4096	26889 ns	26888 ns	26036	items_per_second=5.85097k/s
BM_Biquad_Process/8192	36716 ns	36715 ns	19066	items_per_second=11.7027k/s
BM_Cascade_Process/256	11859 ns	11859 ns	58916	items_per_second=366.416/s
BM_Cascade_Process/512	21682 ns	21682 ns	32269	items_per_second=731.803/s
BM_Cascade_Process/1024	26858 ns	26857 ns	26056	items_per_second=1.46328k/s
BM_Cascade_Process/2048	36198 ns	36197 ns	19474	items_per_second=2.90535k/s
BM_Cascade_Process/4096	54204 ns	54204 ns	12841	items_per_second=5.88479k/s
BM_Cascade_Process/8192	90813 ns	90813 ns	7692	items_per_second=11.7274k/s

Figura 1: Resultados perfilador 1

En la Figura 2, por su parte tenemos la transpuesta pero con una especie loop unrolling en la que se trabaja muestra por muestra, haciendo que los resultados parciales se reflejen directamente en el resultado final, disminuyendo el recorrido de los buffers a sola un ciclo para todas las muestras.

Benchmark	Time	CPU	Iterations	UserCounters...
BM_Biquad_Process/256	8673 ns	8672 ns	80527	items_per_second=366.575/s
BM_Biquad_Process/512	17693 ns	17684 ns	39513	items_per_second=732.757/s
BM_Biquad_Process/1024	20008 ns	20007 ns	35474	items_per_second=1.44281k/s
BM_Biquad_Process/2048	21978 ns	21978 ns	31827	items_per_second=2.92786k/s
BM_Biquad_Process/4096	26900 ns	26900 ns	26022	items_per_second=5.85148k/s
BM_Biquad_Process/8192	36777 ns	36777 ns	19029	items_per_second=11.7059k/s
BM_Cascade_Process/256	11182 ns	11182 ns	62534	items_per_second=366.098/s
BM_Cascade_Process/512	20331 ns	20330 ns	34468	items_per_second=730.649/s
BM_Cascade_Process/1024	23316 ns	23316 ns	30033	items_per_second=1.46235k/s
BM_Cascade_Process/2048	27848 ns	27846 ns	25186	items_per_second=2.92018k/s
BM_Cascade_Process/4096	36971 ns	36971 ns	18934	items_per_second=5.85143k/s
BM_Cascade_Process/8192	55310 ns	55307 ns	12655	items_per_second=11.7044k/s

Figura 2: Resultados perfilador 2

Por último, se realiza una optimización agresiva de la versión anterior, utilizando banderas del compilador para aprovechar la arquitectura y prefetch a L1 de los datos en un rango de 16 posiciones de float.

Benchmark	Time	CPU	Iterations	UserCounters...
BM_Biquad_Process/256	857 ns	857 ns	807580	items_per_second=370.033/s
BM_Biquad_Process/512	1709 ns	1709 ns	408966	items_per_second=732.77/s
BM_Biquad_Process/1024	3416 ns	3416 ns	204455	items_per_second=1.4662k/s
BM_Biquad_Process/2048	6851 ns	6850 ns	101398	items_per_second=2.94851k/s
BM_Biquad_Process/4096	13702 ns	13701 ns	50892	items_per_second=5.87441k/s
BM_Biquad_Process/8192	27352 ns	27351 ns	25631	items_per_second=11.6857k/s
BM_Cascade_Process/256	1207 ns	1207 ns	581577	items_per_second=364.748/s
BM_Cascade_Process/512	2409 ns	2409 ns	290451	items_per_second=731.695/s
BM_Cascade_Process/1024	4816 ns	4816 ns	145258	items_per_second=1.4639k/s
BM_Cascade_Process/2048	9629 ns	9629 ns	72583	items_per_second=2.93039k/s
BM_Cascade_Process/4096	19263 ns	19262 ns	36337	items_per_second=5.85209k/s
BM_Cascade_Process/8192	38516 ns	38515 ns	18153	items_per_second=11.717k/s

Figura 3: Resultados perfilador 3