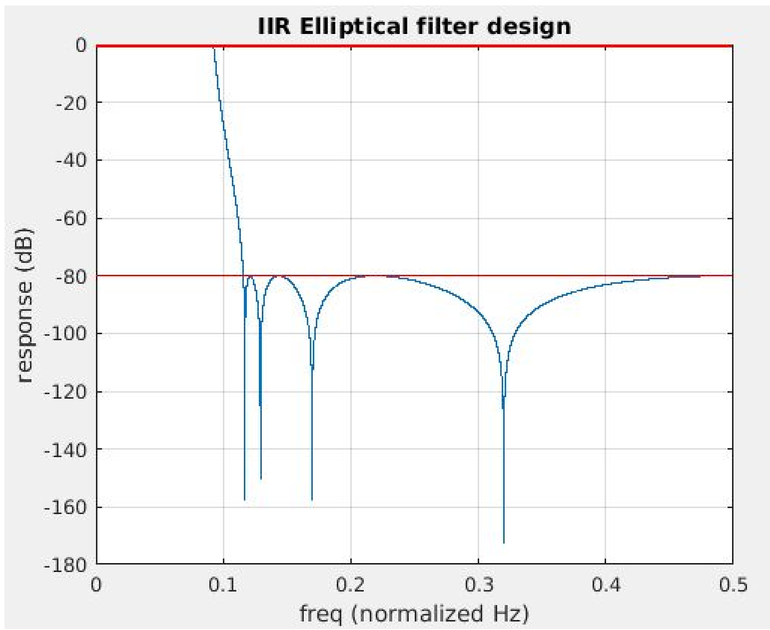
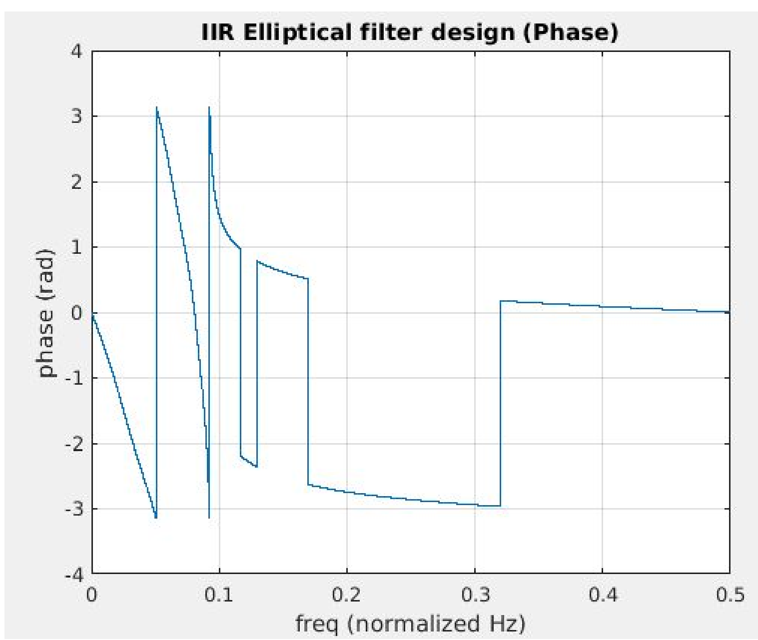
ESE2014: Filter realization exercise

Name: Shreya Mamadapur
Student ID: C0774035
Instructor: Takis Zourntos

Examine the IIR elliptical filter design in the given GitHub script:
https://github.com/takiszourntos/teaching/tree/master/lambton/2020/summer/ese2014/week_11/matlab-workspace

RESULT:





ESE2014: Filter realization exercise

1) What is the transfer function of this filter?

Transfer function of a filter can be found in matlab using the built-in function tf()

Discrete-time transfer function,

H_e =

0.0004022 z^8 - 0.001198 z^7 + 0.002237 z^6 - 0.002749 z^5 + 0.002997 z^4 - 0.002749 z^3 + 0.002237 z^2 - 0.001198 z + 0.0004022

----

z^8 - 6.75 z^7 + 20.6 z^6 - 37.02 z^5 + 42.79 z^4 - 32.53 z^3 + 15.89 z^2 - 4.556 z + 0.5875

Sample time: 5e-06 seconds

2) Can you provide a state-space model for this filter?

The state-space model can be easily found in Matlab used the built-in function ss()

Discrete-time state-space model.

H_e_ss =

A =

	x1	x2	x3	x4	x5	x6	x7	x8
x1	6.75	-5.149	2.314	-1.337	1.017	-0.4965	0.2847	-0.1469
x2	4	0	0	0	0	0	0	0
x3	0	4	0	0	0	0	0	0
x4	0	0	2	0	0	0	0	0
x5	0	0	0	1	0	0	0	0
x6	0	0	0	0	1	0	0	0
x7	0	0	0	0	0	0.5	0	0
x8	0	0	0	0	0	0	0.25	0

ESE2014: Filter realization exercise

B =

	u1
x1	0.0625
x2	0
x3	0
x4	0
x5	0
x6	0
x7	0
x8	0

C =

	x1	x2	x3	x4	x5	x6	x7	x8
y1	0.02426	-0.02418	0.01214	-0.007105	0.005168	-0.002076	0.0006339	0.0006635

D =

	u1
y1	0.0004022

Sample time: 5e-06 seconds

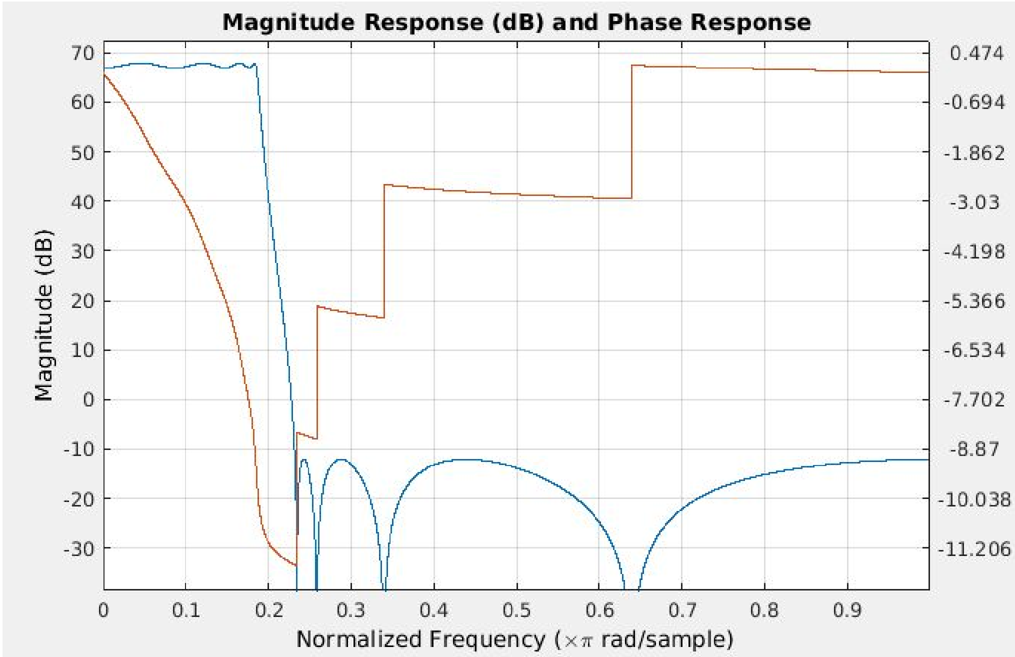
3) Can you provide a realization of this filter based on second-order sections?

Yes, we can realize the filter based on sos(second-order section)

a) Using fvtool()

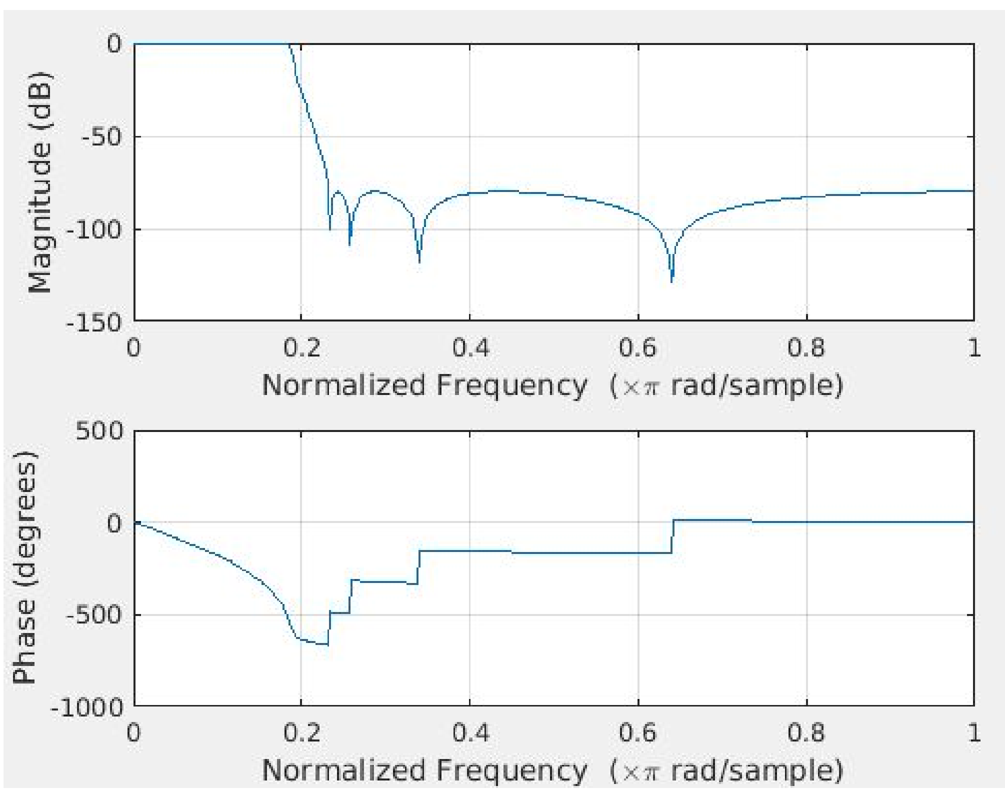
```
[num_e, den_e] = ellip(8, Rp, Rs, 2.0*Fpass/Fs);
[sos,gn] = tf2sos(num_e, den_e)
fvtool(sos)
```

ESE2014: Filter realization exercise



b) Using freqz

```
[z,p,k] = ellip(8, Rp, Rs, 2.0*Fpass/Fs);
soshi = zp2sos(z,p,k);
freqz(soshi)
```

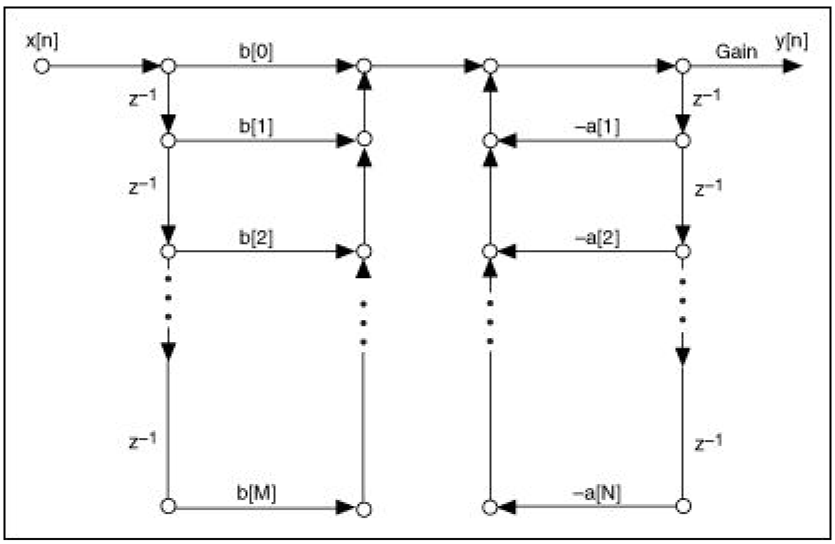


ESE2014: Filter realization exercise

4) Examine the coefficients of your filter realizations. Do they seem "balanced"? Is your filter likely to suffer from the effects of fixed-point arithmetic, such as overflow or rounding errors? Why or why not?

Yes, they seem balanced. The filter realized using second-order sections is less likely to suffer from rounding/truncating and overflow errors.

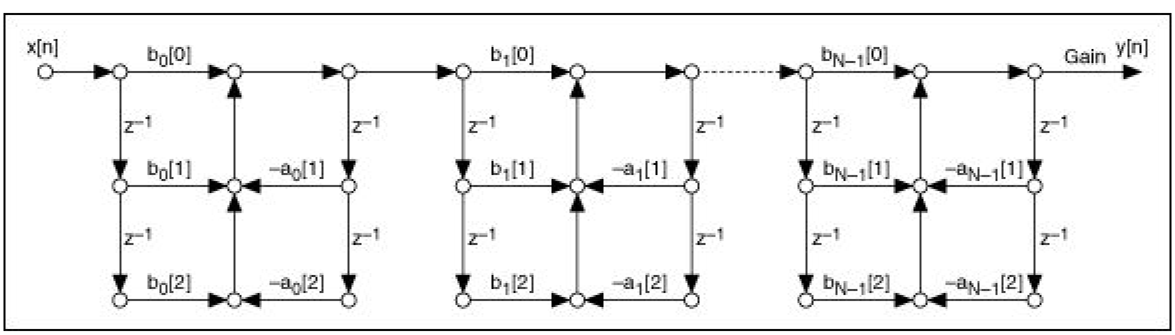
[Direct-form



tf() function returns filter coefficients [b, a]

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_1 + b_2 z^{-1} + \cdots + b_{n+1} z^{-n}}{a_1 + a_2 z^{-1} + \cdots + a_{m+1} z^{-m}}.$$

Cascaded second-order-sections



By calculating, the coefficients using the sos errors such as rounding/truncated error and overflow error are minimized.

ESE2014: Filter realization exercise

[sos, gn] tf2sos() function returns a matrix sos which is in the form

$$\text{sos} = \begin{bmatrix} b_{01} & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ b_{02} & b_{12} & b_{22} & 1 & a_{12} & a_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{0L} & b_{1L} & b_{2L} & 1 & a_{1L} & a_{2L} \end{bmatrix}$$

whose rows contain the numerator and denominator coefficients bik and aik of the second-order sections of H(z)

$$H(z) = g\prod_{k=1}^{L} H_k(z) = g\prod_{k=1}^{L} \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}.$$

]

sos =

1.0000	-1.4832	1.0000	1.0000	-1.6532	0.9770
1.0000	-1.3769	1.0000	1.0000	-1.6597	0.9212
1.0000	-0.9674	1.0000	1.0000	-1.6977	0.8439
1.0000	0.8479	1.0000	1.0000	-1.7390	0.7735