OnCalismaOrnek7

March 10, 2020

First, we need to import the libraries that will be used in the script. Generally, we will use the following: * numpy : for basic mathematical and array operations. * matplotlib : for various plotting purposes. * scipy : general signal processing functions (specifically "signal")

```
[ ]: # import the necessary libraries
import numpy as np              # for using basic array functions
import matplotlib.pyplot as plt # for this example, it may not be necessary

# the main package for signal processing is called "scipy" and we will use␣
 ↪"signal" sub-package
import scipy.signal as sgnl
# alternative syntax: from scipy import signal as sgnl
```

We define the system that is given in the problem with its zeros and poles, then convert it to the "transfer function" form and obtain its coefficients: The given system is:

$$X(z) = \frac{1}{(1 - \frac{1}{4}z^{-1})(1 - \frac{1}{2}z^{-1})}, |z| > 1/2$$

```
[ ]: z = np.array([0])                # a zero @z=0
p = np.array([1.0/4, 1.0/2])     # poles of the system
b, a = sgnl.zpk2tf(z, p, 1)      # since there is no gain k=1
b,a
```

Now, we obtained the coefficients in the descending order of z, beginning with the constant term, i.e.

$$b(z) = 1$$

and

$$a(z) = 1 - 0.75z^{-1} + 0.125z^{-2}$$

thus,

$$X(z) = \frac{1}{1 - 0.75z^{-1} + 0.125z^{-2}}$$

Alternatively, since we know what the numerator term is, we can expand the product in the denominator by polynomial multiplication, i.e. convolution:

```
[ ]: # alternative way to expand a product:
a = sgnl.convolve(np.array([1, -1/4]),np.array([1, -1/2]))
```

```
a
```

Now that it is in a more convenient format (in the transfer function form), we can compute the coefficients of the partial fraction form as follows:

```
[ ]: # given the coeffs of numerator, i.e. b(z) and the coeffs of denominator a(z),
     # we do the partial fraction expansion by:
     r, p, k = sgnl.residuez(b,a)          #
     r,p,k

     # to check the correctness of the polynomial roots (i.e. p's) we can use
     poless = np.roots(a)      # returns the polynomial coefficients of the denominator
```

the result is interpreted as follows:

$$X(z) = \frac{1}{(1 - \frac{1}{4}z^{-1})(1 - \frac{1}{2}z^{-1})} \quad (1)$$

$$= \frac{r[0]}{1 - p[0]z^{-1}} + ... + \frac{r[-1]}{1 - p[-1]z^{-1}} + k[0] + k[1]z^{-1}... \quad (2)$$

- r: is the numerator of each term,
- p: is the poles of the system,
- k: is the polynomial term (if any).

For the example given above, we get:

$$X(z) = \frac{-1}{1 - 0.25z^{-1}} + \frac{2}{1 - 0.5z^{-1}}, |z| > 0.5$$

Then, we can write the inverse z-transform of $X(z)$ by the inspection method:

$$x[n] = -(\frac{1}{4})^n u[n] + 2(\frac{1}{2})^n u[n]$$

Note the signs of the denominator coeffs