# Autocorrelation method

M.Tognolini HEIG-VD 2022  MA-StatDig

## Exemple  with a sinus  signal

We  want to modelize  the  signal  with  a  all pole second order model :

$$H(z) = \frac{b(0)}{1 + a(1)z^{-1} + a(2)z^{-2}} \quad (1)$$

The normal equation are :

$$\begin{bmatrix} r_x(0) & r_x(1) \\ r_x(1) & r_x(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \end{bmatrix} = - \begin{bmatrix} r_x(1) \\ r_x(2) \end{bmatrix} \quad (2)$$

$$R_x \overline{a}_p = -r_x$$

### Theoric  value of  the autocorrelation  and  Prony  computed all pole  filter

The theoric autocorrelation of x(n)  are given by (see exemple 3.3.1 in chapter 3) :

$$r_x(k) = E\{x(n)x(n-k)\} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=k}^{N} x(n)x(n-k) = \frac{1}{2} A^2 cos(k\omega_0) \quad (3)$$

Try  several  number od  samples N ,  firs  a multiple of  the period length (ex 40)  and  alse  38  or  42  or 55 .... the  result  wil be  very  different to  the  théory.

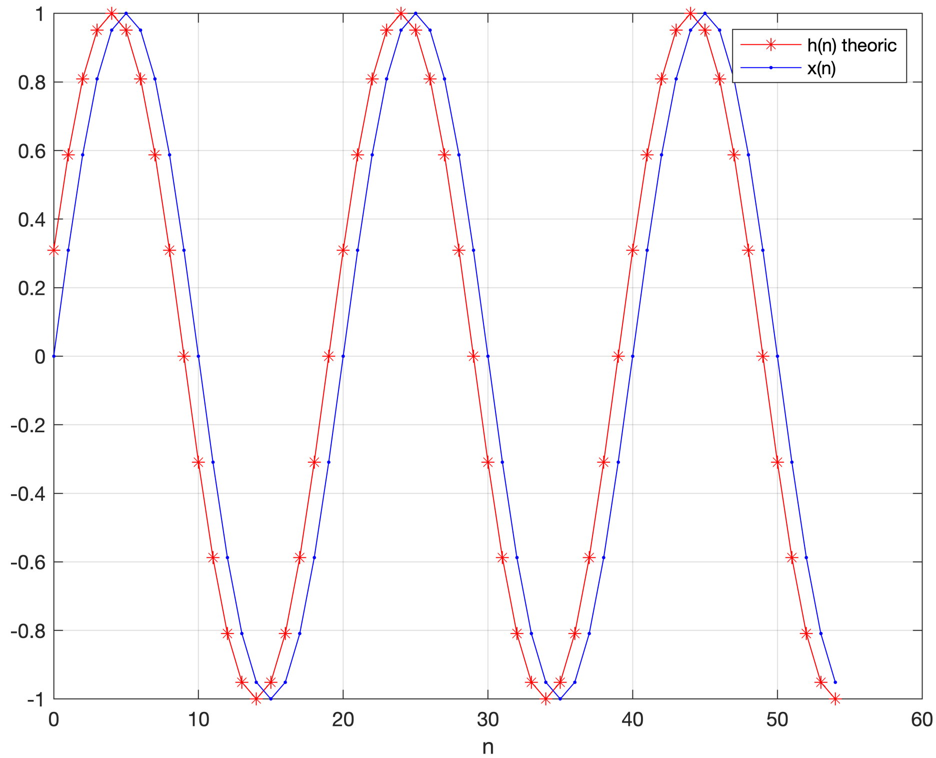
```
N = 55
N0 = 20
A = 1
x = 55×1
         0
    0.3090
    0.5878
    0.8090
    0.9511
    1.0000
    0.9511
    0.8090
    0.5878
    0.3090
      ⋮

p = 2
```

If we compare with the theoric values of the autocorrelation (3) we have the values $r_x(k)$ for k = 0,1,2 :

```
rxth = 1×3
    0.5000    0.4755    0.4045

Rxth = 2×2
    0.5000    0.4755
    0.4755    0.5000

apth = 3×1
    1.0000
   -1.9021
    1.0000

Ep = 5.5511e-17

pth = 2×1 complex
   0.9511 + 0.3090i
   0.9511 - 0.3090i
```

As expected the norm of the poles is = 1, that produces an impulse response h(n) = x(n).

```
Norm_pth = 2×1
    1.0000
    1.0000

H_th =

        0.309
  -----------------
  z^2 - 1.902 z + 1

Sample time: 1 seconds
Discrete-time transfer function.

h_th = 1×55
    0.3090    0.5878    0.8090    0.9511    1.0000    0.9511    0.8090    0.5878 ···
```




## Autocorrelation Method to estimate the all pole filter

The estimation of the autocorrelation as the Prony Autocorrelation method is different:

$$\hat{r}_x(k) = \frac{1}{N}\sum_{n=k}^{N} x(n)x(n-k) \quad ; \; k \geq 0 \quad (4)$$

To estimate the autocorrelation with a finite record of N points we use the cnvolution matrix:

```
X = convmtx(x,p+1);
Xp = X(1:N+p-1, 1:p)
Rx = Xp' * Xp *1/N
rx = Xp'*X(2:N+p,1) / N
```

```
Xp = 56×2
        0         0
   0.3090         0
   0.5878    0.3090
   0.8090    0.5878
   0.9511    0.8090
   1.0000    0.9511
   0.9511    1.0000
   0.8090    0.9511
   0.5878    0.8090
   0.3090    0.5878
```

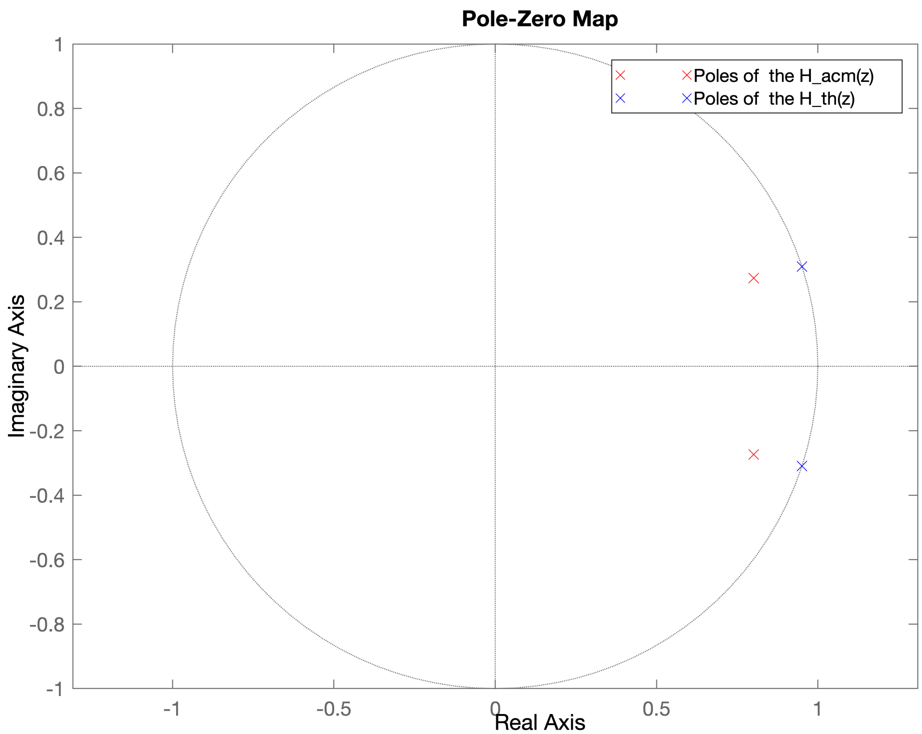
```
      ⋮

Rx = 2×2
    0.4909    0.4582
    0.4582    0.4909

rx = 2×1
    0.4582
    0.3824

ap = 3×1
    1.0000
   -1.6027
    0.7170

err = 1.6882

bq0 = 1.2993
```

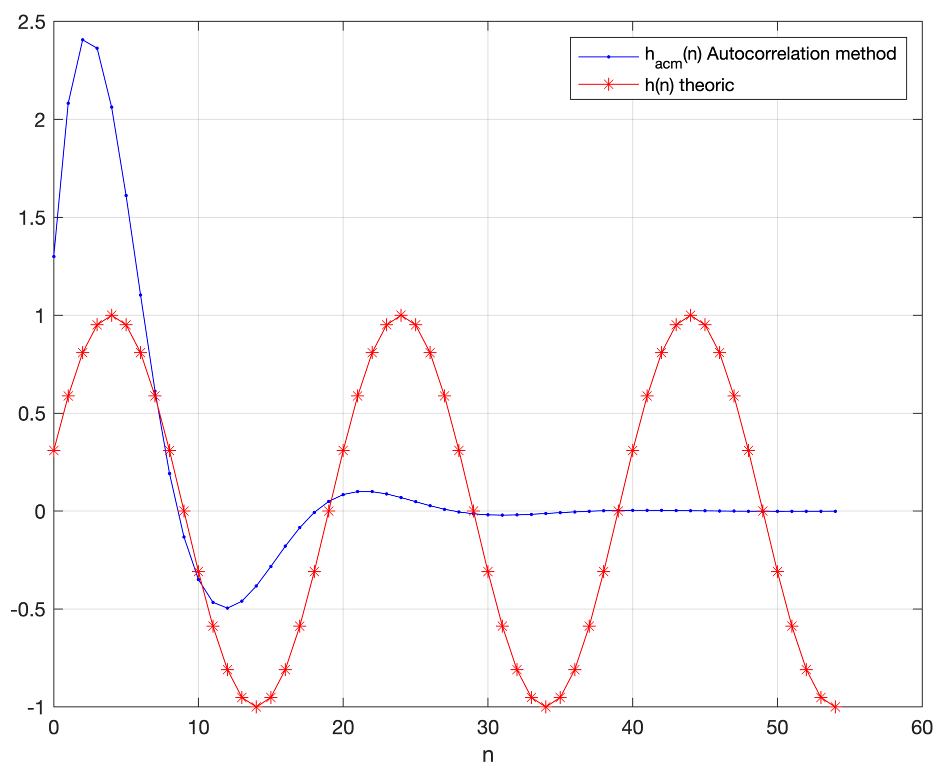
The pole of the transfer function are :

```
poles = 2×1 complex
   0.8014 + 0.2735i
   0.8014 - 0.2735i
```

We see the norm of the poles are < 1 , filter is stable but the impulse response h(n) is the damped sinus ....

```
Norm_p = 2×1
    0.8468
    0.8468

H_acm =

          1.299
  ---------------------
  z^2 - 1.603 z + 0.717

Sample time: 1 seconds
Discrete-time transfer function.

h_acm = 1×55
    1.2993    2.0825    2.4060    2.3630    2.0622    1.6108    1.1031    0.6130 ···
```




## Covariance Method to estimate the all pole filter

Now try with the Covariance method:

$$\hat{r}_{xcov}(k,l) = \frac{1}{N}\sum_{n=p}^{N} x(n-l)x(n-k)$$

we compute the autocorrelation matrix with the convm :

```
Xp_cov = X(p:N-1, 1:p)
Rxcov = Xp_cov' * Xp_cov *1/N
```

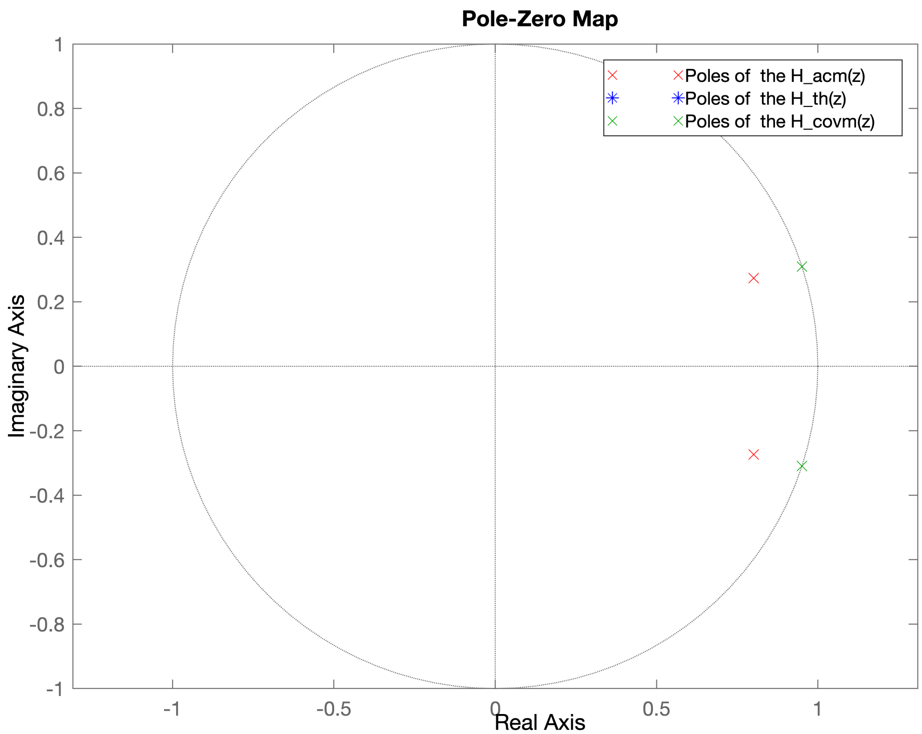
```
Xp_cov = 53×2
    0.3090         0
    0.5878    0.3090
    0.8090    0.5878
    0.9511    0.8090
    1.0000    0.9511
    0.9511    1.0000
    0.8090    0.9511
    0.5878    0.8090
    0.3090    0.5878
    0.0000    0.3090
      ⋮

Rxcov = 2×2
    0.4745    0.4442
    0.4442    0.4626

rx_cov = 2×1
    0.4582
    0.3824

ap_cov = 3×1
    1.0000
   -1.9021
    1.0000

err_cov = 7.1054e-15

bq0_cov = 8.4294e-08

bq0_th = 0.3090
```

The pole of the transfer function are :

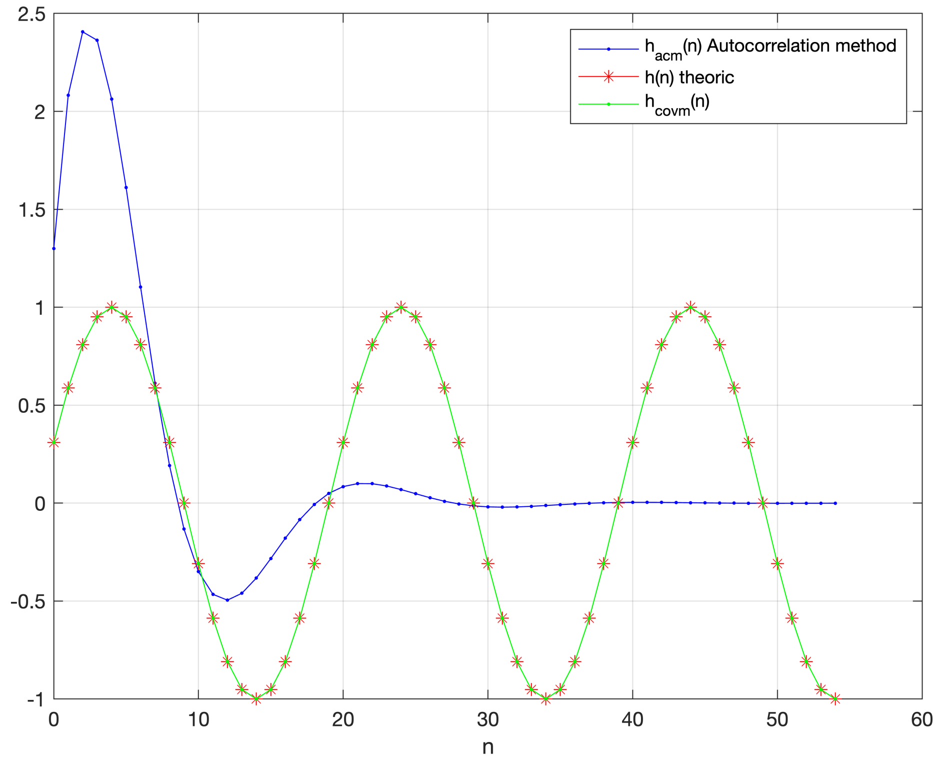
```
poles_cov = 2×1 complex
   0.9511 + 0.3090i
   0.9511 - 0.3090i

Norm_p_cov = 2×1
    1.0000
    1.0000

H_cov =

        0.309
  -----------------
  z^2 - 1.902 z + 1
```

```
Sample time: 1 seconds
Discrete-time transfer function.

h_cov = 1×55
    0.3090    0.5878    0.8090    0.9511    1.0000    0.9511    0.8090    0.5878 ···
```

## Models compare: Theory, Autocorrelation method and Covariance method




If the estimation of the autocorrelation rx is not accurate ( because N is not bigger enough or is not a multiple of the periode length) the poles given by the autocorrelation method is not well estimated giving a damped sinus impulse response :




## Write and Test AutoCorrelation function with a simple example

Write the Autocorrelation method function for all pole model using Prony with finite data records that suppose that outside the recording the samples are equal to 0.

```
acm compute the all pole model using Prony  with the autocorrelation
method:
  [ap,bq0,ep] = acm(x,p) :x is  the finite length  record  with N samples, p is  the
  desired  number of poles : ap is  the  denominator  coefficients ap = [1,a(1), a(2), ..., a(p)]
  bq0 is  the  numerator unique coefficient bq(0) ,ep  is  the approximation norm of the error.
  M.Tognolini (c) HEIG-VD 2022
```

Test the fuction with a finite data record and find an all pole 2nd order model :

$$H(z) = \frac{b(0)}{1 + a(1)z^{-1} + a(2)z^{-2}}$$

With the signal x(n) defined as:

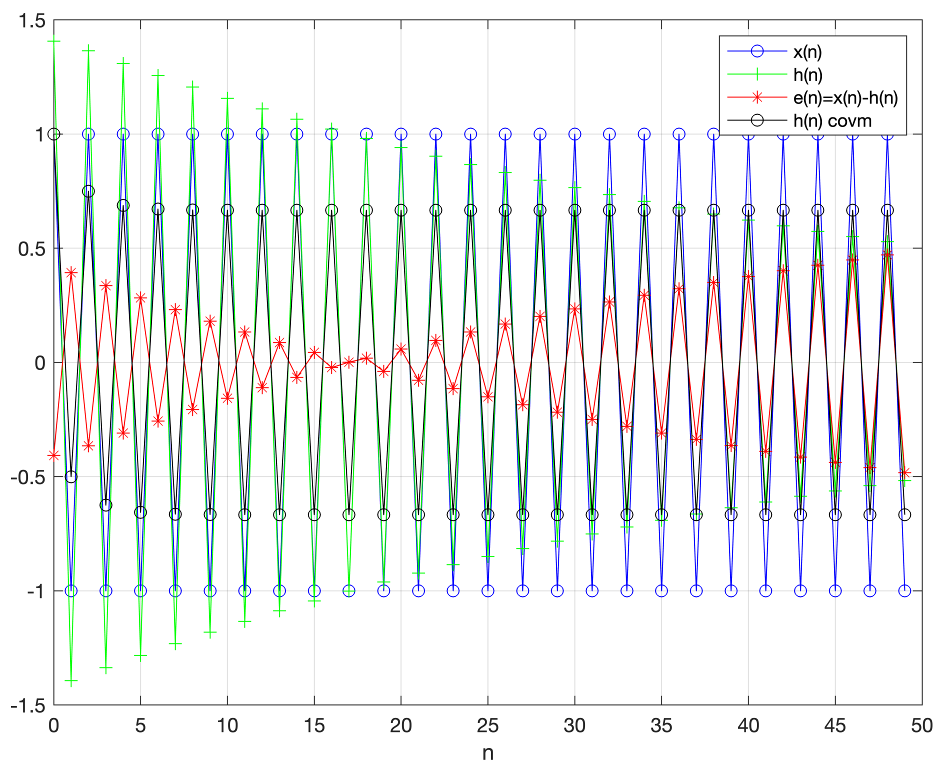
```
N = 20
n = 0:N-1;
x = (-1).^n.*ones(1,N)
```

```
N = 50
```

```
x = 1×50
     1    -1     1    -1     1    -1     1    -1     1    -1     1    -1     1···

p = 2

Calculating the  squared error of  the approximation
ap_acm = 3×1
    1.0000
    0.9899
    0.0101
bq0_acm = 1.4071
err_acm = 1.9798

Calculating the  squared error of  the approximation
ap_com = 3×1
    1.0000
    0.5000
   -0.5000
bq0_com = 2.0648e-07
err_com = 4.2633e-14
```

Test the model with the impulse response h(n) and compare it to the x(n) signal:

```
h_acm = 1×50
    1.4071   -1.3928    1.3646   -1.3367    1.3094   -1.2827    1.2565   -1.2309···

h_com = 1×50
    1.0000   -0.5000    0.7500   -0.6250    0.6875   -0.6563    0.6719   -0.6641···

e_LS_acm = 2.0052
```




## Exemple with a noisy sinus signal

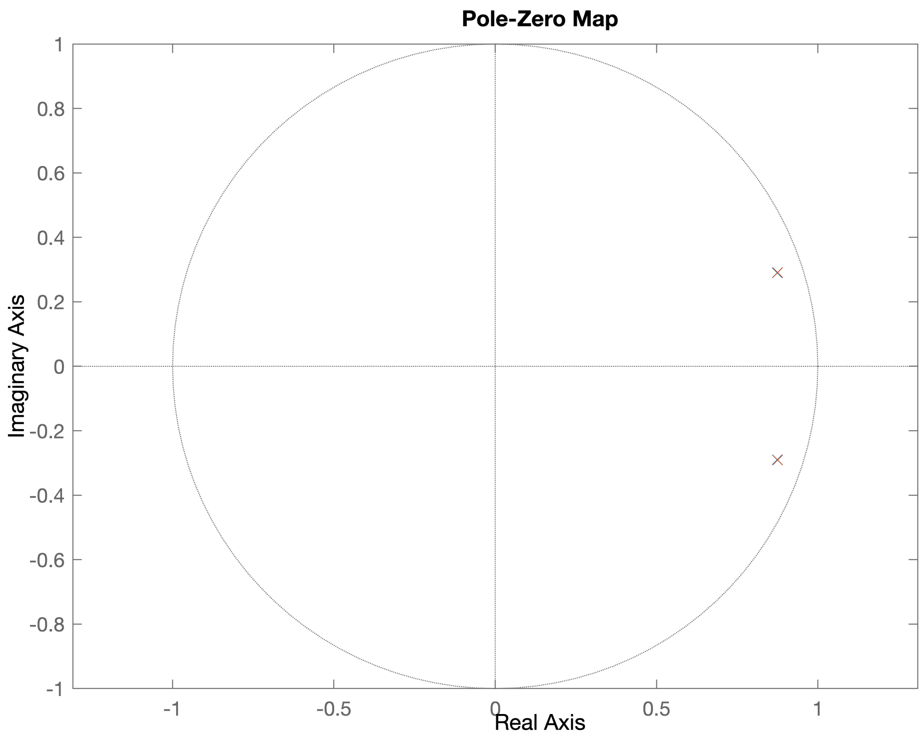
```
N = 400

N0 = 20

A = 1

x = 1×400
         0    0.3090    0.5878    0.8090    0.9511    1.0000    0.9511    0.8090 ···

Sigman = 0.0500

xn = 1×400
    0.0269    0.4007    0.4748    0.8521    0.9670    0.9346    0.9294    0.8261 ···
```

We want to find a second order model (2 poles) with acm:

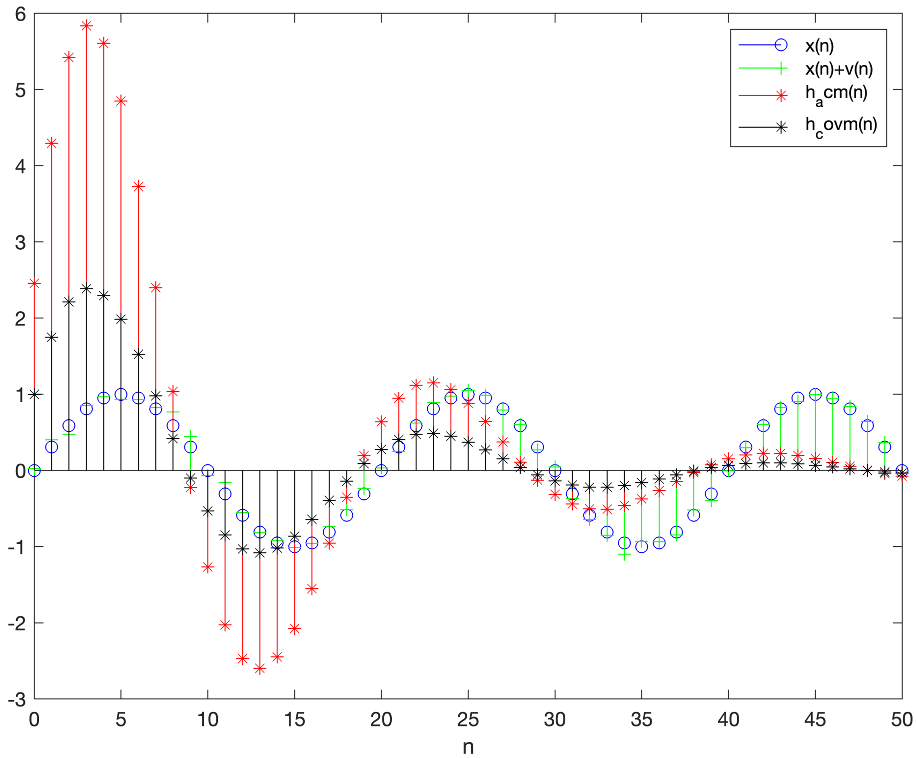
```
p = 2

Calculating the  squared error of  the approximation
ap_acm2 = 3×1
    1.0000
   -1.7479
    0.8477
bq0_acm2 = 2.4565
err_acm2 = 6.0346

Calculating the  squared error of  the approximation
ap_com2 = 3×1
    1.0000
   -1.7506
    0.8507
bq0_com2 = 2.4195
err_com2 = 5.8540
```

The pole zero map is :

```
H_acm =

          2.457
  ----------------------
  z^2 - 1.748 z + 0.8477

Sample time: 1 seconds
Discrete-time transfer function.

H_com =

            1
  ----------------------
  z^2 - 1.751 z + 0.8507

Sample time: 1 seconds
Discrete-time transfer function.
```




```
h_acm2 = 1×400
    2.4565    4.2937    5.4223    5.8376    5.6068    4.8512    3.7262    2.4005 ···

h_com2 = 1×400
    1.0000    1.7506    2.2138    2.3863    2.2942    1.9862    1.5253    0.9806 ···
```



x(n)
x(n)+v(n)
h_acm(n)
h_covm(n)
n