Padé approximation

M.Tognolini HEIG-VD 2022 MA-StatDig

Example 4.3.2

Test Padé approximation with the signal x(n) defined below.

Test with AR(2) model, AM(2) and ARMA(1,1).

```
N = 12;
n = 0:N-1;
xn = [1, 3* (0.5).^n(2:end)]
dn = [1,zeros(1,N-1)];
```

```
xn = 1×12
    1.0000    1.5000    0.7500    0.3750    0.1875    0.0938    0.0469    0.0234 ···
```

a) Test with AR(2) : p=2 and q = 0

```
p=2
q=0
xn = xn(:)
```

```
p = 2

q = 0

xn = 12×1
    1.0000
    1.5000
    0.7500
    0.3750
    0.1875
    0.0938
    0.0469
    0.0234
    0.0117
    0.0059
      ⋮
```

Compute the Xq matrix using the Matlab X= convmtx(xn,...) function and chose the right index to extract the Xq as in theory eq (4.13)

```
X = convmtx(xn,p+1)
```

```
X = 14×3
    1.0000         0         0
    1.5000    1.0000         0
    0.7500    1.5000    1.0000
    0.3750    0.7500    1.5000
    0.1875    0.3750    0.7500
```

```
     0.0938    0.1875    0.3750
     0.0469    0.0938    0.1875
     0.0234    0.0469    0.0938
     0.0117    0.0234    0.0469
     0.0059    0.0117    0.0234
       ⋮

  Xq = 2×2
     1.0000         0
     1.5000    1.0000
```

Compute the vector x_{q+1} as well:

```
  xq_1 = 2×1
     1.5000
     0.7500
```

And now compute the ap coefficients :

```
  ap = 3×1
     1.0000
    -1.5000
     1.5000
```

Compute the numerator coefficienzs here is obvous that bq(0) = xn(0)

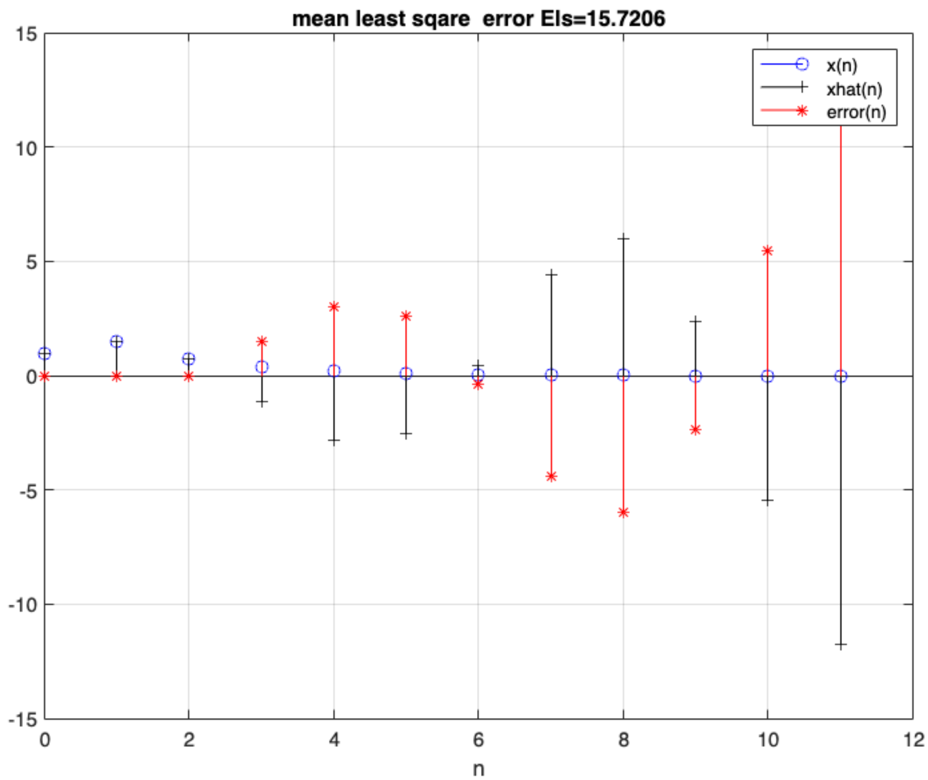
```
  X0 = 1×3
      1     0     0

  bq = 1
```

Now compute the impulse response that is $\widehat{x}(n) = h(n)$

```
  xhat = 12×1
     1.0000
     1.5000
     0.7500
    -1.1250
    -2.8125
    -2.5312
     0.4219
     4.4297
     6.0117
     2.3730
       ⋮
```

Compute the error of t he approximation and the least square error:

```
  Els = 15.7206
```



b) Test with MA(2) : p=0 and q = 2

In this case the solution for bq(n) is obvous because   h(n) = bq(n) so bq(n) = x(n) :

```
q = 2

p = 0

bqma = 3×1
    1.0000
    1.5000
    0.7500

apma = 1

xhatma = 12×1
    1.0000
    1.5000
    0.7500
         0
         0
         0
         0
         0
         0
         0
    ⋮

Elsma = 0.4330
```

c) Test with ARMA(1,1) : p=1 and q = 1

In this case we need to solve with 2 step algorithm first we compute the ap coeff as in point a)

```
q = 1

p = 1

Xq = 1.5000
```

Compute the vector x_{q+1} as well:

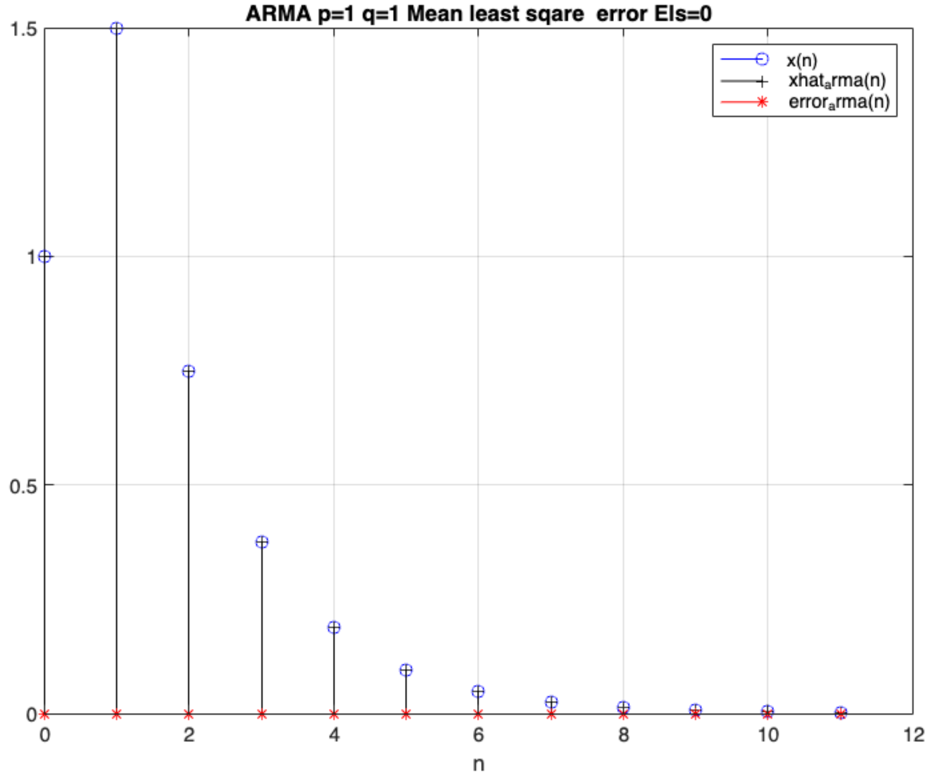
```
xq_1 = 0.7500

ap_arma = 2×1
    1.0000
   -0.5000
```

Next we apply the step 2 and compute the bq coefficients:

```
X0 = 2×2
    1.0000         0
    1.5000    1.0000

bq_arma = 2×1
     1
     1
```

The transfer function H(z) is:

```
num/den =

    z + 1
   -------
   z - 0.5

xhat_arma = 12×1
    1.0000
    1.5000
    0.7500
    0.3750
    0.1875
    0.0938
    0.0469
    0.0234
    0.0117
    0.0059
      ⋮

Els_arma = 0
```



Build a functionto compute the coefficients ap and bq by Padé approximation

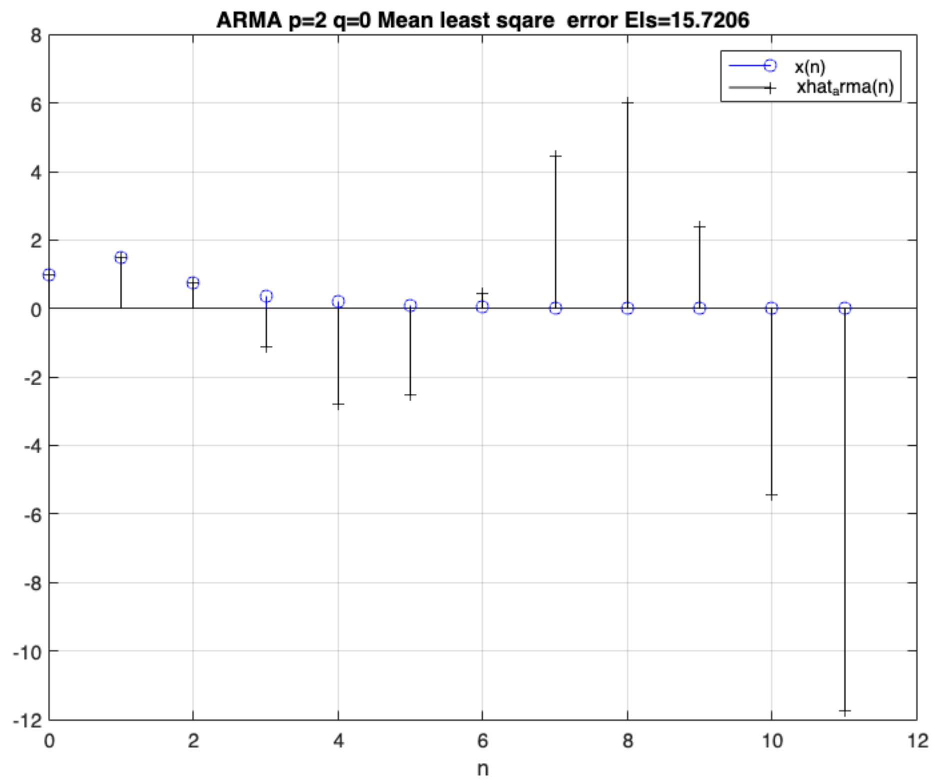
The function heade would be :

```
pade compute the pade approximation of the coeff of  ARMA  system with p
poles  and q zeros usage: [ap,bq,Els,xhat] = pade(x,p,q)
Els is optional and is the L2 norm of  the approx error x-xhat
xhat is optional  and it is the  approximated signal.
(c) M.Tognolini HEIG-VD 2022 r1.0

  Other functions named pade
```

test it with the prevous example.

```
p = 2

q = 0

Calculating the  squared error of  the approximation
and the approximation signal xhat
ap_test = 3×1
    1.0000
   -1.5000
    1.5000
bq_test = 1
Els_test = 15.7206
xhat_test = 12×1
```

```
 1.0000
 1.5000
 0.7500
-1.1250
-2.8125
-2.5312
 0.4219
 4.4297
 6.0117
 2.3730
  :
```



Example 4.3.3 : Singular exemple

Digital filter approximation

```
x2 = 1×5
     1     4     2     1     3

p = 2

q = 2
```

Compute the pade approximation with the pade function:

```
Error using pade
Xq is  a singular matrix !
```

The Xq matrix is singular in this case !

That means the assumption that ap(0) = 1 is incorrect for this model so we put it to 0.

Example 4.3.4: Filter design using the Padé approximation

The proble of the filter approximation is to find the impulse response h(n) of the filter from a frequency constraints.

For this exemple to be as simple as possible we want to synthetize a filter with a frequency response $|H(jf)| = 1$ for f < Fp and 0 for Fp < f < 0.5,

aditionally we want a linear phase response then $H(jf) = e^{-jn_d 2\pi f}$ for f < Fp and 0 for Fp < f < 0.5 , the constant nd is the time delay of the filter to assure the causality of the filter.

$$H(jf) = \begin{cases} e^{jn_d 2\pi f} & ; \ |f| < 0.25 \\ 0 & ; \ \text{otherwise} \end{cases}$$

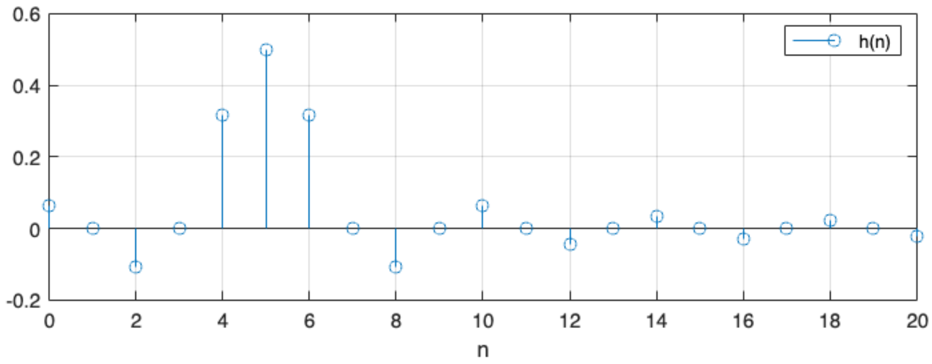
Since the frequency response is a rectangular function the inverse transform is a sinc function of the variable n, the the impulse response is :

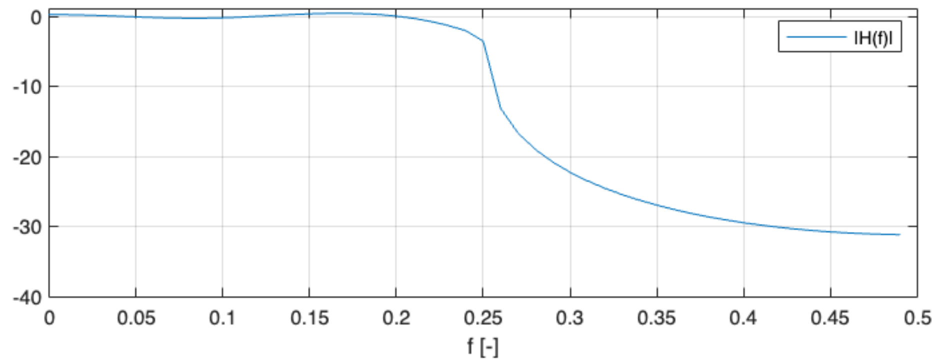
$$h(n) = \frac{sin[(n-n_d)\pi/2]}{(n-n_d)\pi}$$

With the function sinc we write it as: hn = 1/2 * sinc((n-nd)*2*Fp)

```
N = 100
Fp = 0.25
n= 0:N-1;
nd = 5

hn = 1/2 * sinc((n-nd)*2*Fp)

H = fftshift(fft(hn));
f = -0.5:1/N:0.5-0.5/N;


figure;
subplot(2,1,1)
stem(n,hn)
axis([0,20,-0.2,0.6])
grid;
xlabel('n')
legend('h(n)')
subplot(2,1,2)
plot(f,20*log10(abs(H)))
axis([0,0.5,-40,1])
grid;
xlabel('f [-]')
legend('|H(f)|')
```

```
N = 100

Fp = 0.2500

nd = 5

hn = 1×100
    0.0637   -0.0000   -0.1061    0.0000    0.3183    0.5000    0.3183    0.0000···
```



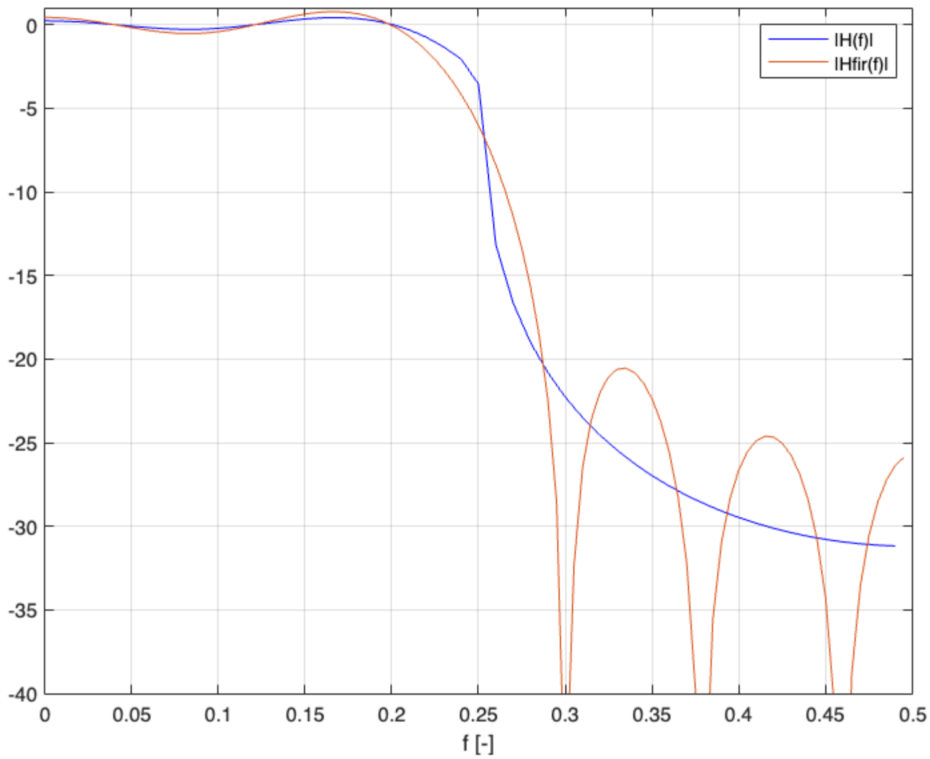


If we design a FIR filter with q= 10 and p=0 (MA) the Padé approximation wil give the 11 first values of the h(n). In this case bq(n) = h(n) for n = 0 ..10.

```
p = 0

q = 10

bqFIR = 1×11
    0.0637   -0.0000   -0.1061    0.0000    0.3183    0.5000    0.3183    0.0000···

apFIR = 1×11
     1     0     0     0     0     0     0     0     0     0     0

Hfir =

  0.06366 z^10 - 1.949e-17 z^9 - 0.1061 z^8 + 1.949e-17 z^7 + 0.3183 z^6 + 0.5 z^5 + 0.3183 z^4 + 1.949e-
                                                                                                      - 1.9
  ----------------------------------------------------------------------------------------------------------
                                                  z^10
```

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

We could plot the frequency response of this filter with the function freqz():

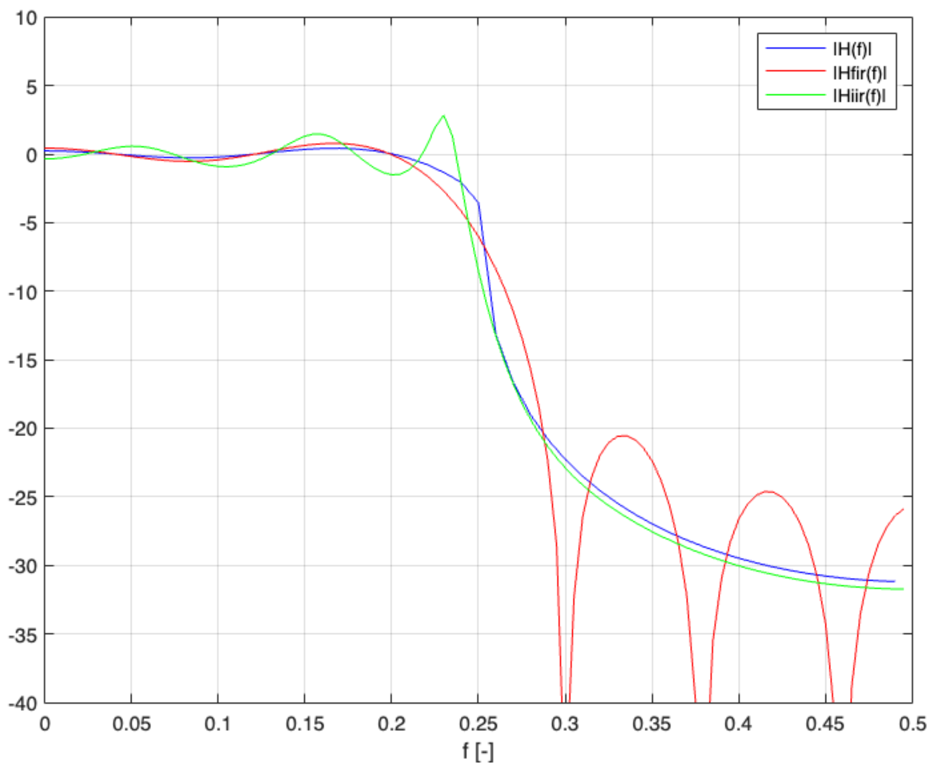


We want to find the transfer fuction of an ARMA system with p=6 and q=6

```
p = 6

q = 6

Calculating the  squared error of  the approximation
ap_IIR = 7×1
    1.0000
   -2.5171
    3.9055
   -4.0872
    2.9647
   -1.4088
    0.3500
bq_IIR = 7×1
    0.0637
   -0.1602
    0.1425
    0.0069
    0.0927
    0.0428
    0.0106
Els_IIR = 4.3922e-14
```

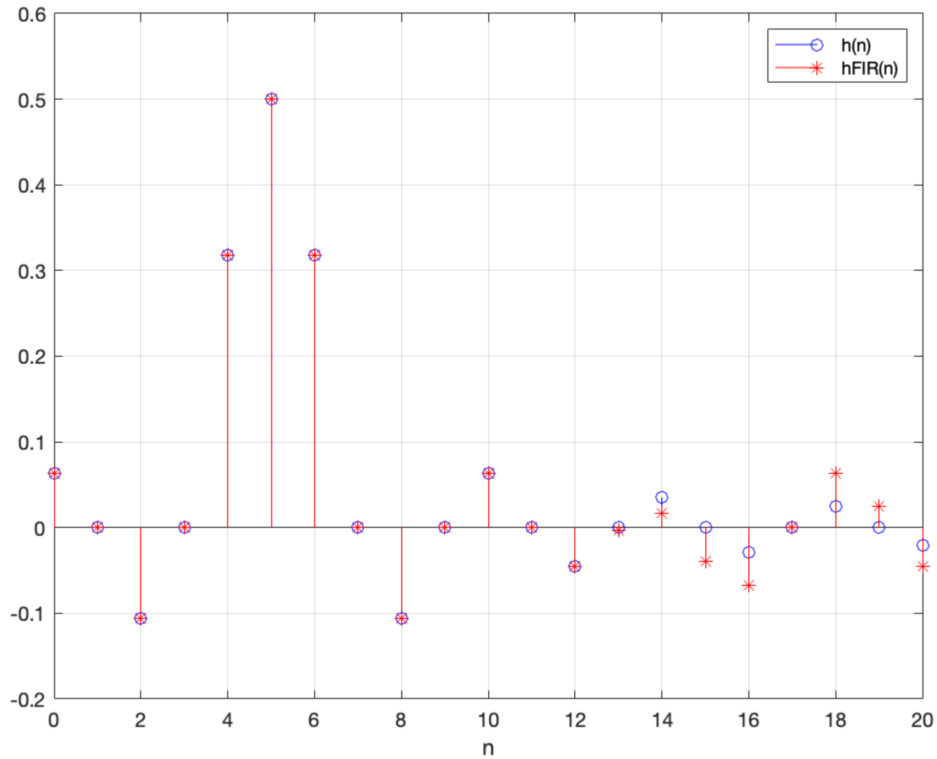
```
Hiir =

  0.06366 z^6 - 0.1602 z^5 + 0.1425 z^4 + 0.006869 z^3 + 0.09267 z^2 + 0.04277 z + 0.01064
  -----------------------------------------------------------------------------------------
            z^6 - 2.517 z^5 + 3.905 z^4 - 4.087 z^3 + 2.965 z^2 - 1.409 z + 0.35

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

As before we will plot the frequency response compared to the desired frequency response H(f):



Compute the impulse response of this filter over the N samples and compare to the original impulse response h(n):

```
hiir = 1×100
    0.0637   -0.0000   -0.1061         0    0.3183    0.5000    0.3183   -0.0000 ···
```



h(n)
hFIR(n)
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0
2
4
6
8
10
12
14
16
18
20
n