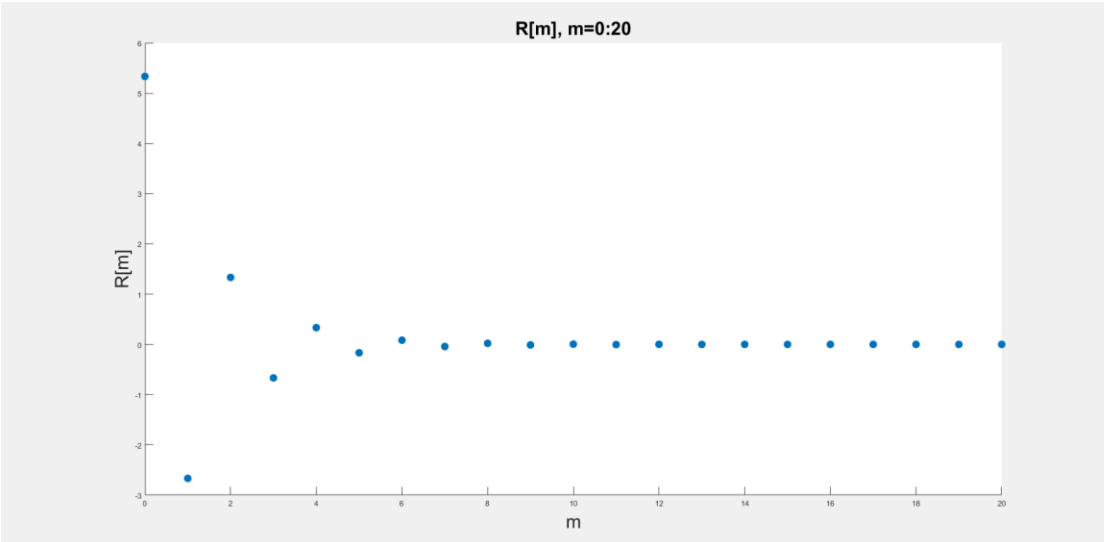
(1) $\hat{F}_s(t = 0) = 0.5100$

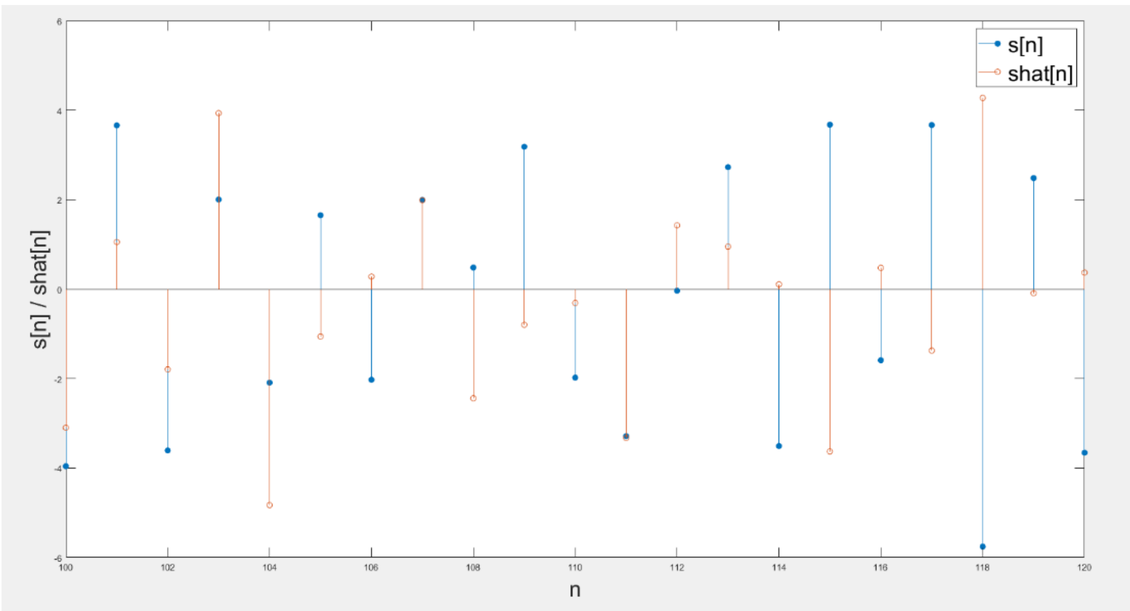
$\hat{F}_s(t = 0.5) = 0.5810$

Random Process hw1
A071547
鄭澄遠

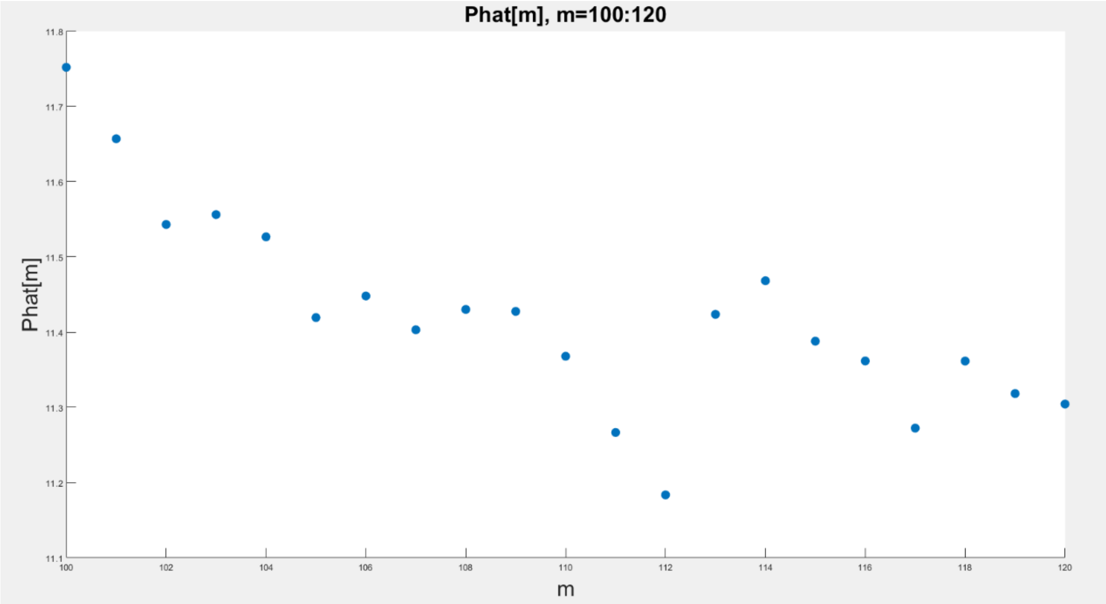
(2)



(3)



(4)


Phat[m], m=100:120
Phat[m]
m
11.8
11.7
11.6
11.5
11.4
11.3
11.2
11.1
100
102
104
106
108
110
112
114
116
118
120

```
Q = 10;
alpha = 0.5;
b = 2;
var = 1;
n = 1000;
```

```
%%
Mdl = arima('AR',{-alpha},'Constant',0,'Variance',var*(b^2));
s = simulate(Mdl,n);

ToEstMdl = arima(Q,0,0);
EstMdl = estimate(ToEstMdl,s);
s_hat = simulate(EstMdl,n);
```

Initially set all parameters. I use "arima" function which is built in Matlab, and use this function to generate sequence s[m]. After generating sequence s[m], I define another AR model which has lag 10, and take sequence s[m] as input. Finally use the input to estimate s_hat[m] sequence, then numbers I need are all computed.

```
%%
t = 0.5;
count = 0;
for i = 1:1000
    if s(i) <= t
        count = count + 1;
    end
end
Fs_hat = count/1000;
```

```
%%
p_hat = zeros(n,1);
for i = 1:n
    p_hat(i) = sum((s_hat(1:i)-s(1:i)).^2)/i;
end
```

This part computes problem (1), (4) base on the sequences I get from previous part, is trivial.

```
function y = Get_R(n)
    A = [1 0.5; 0.5 1];
    b = [4 0];
    R_01 = A\b';
    if n<1
        if n == 0
            y = R_01(1);
        elseif n == 1
            y = R_01(2);
        end
    else
        y = -0.5*Get_R(n-1);
    end
end
```

This part I use Yule-Walker equation to computed R[m]'s. R[0] and R[1] particularly computed by matrix multiplication. For R[m], m>1, I use Yule-Walker equation and recursive call to compute.