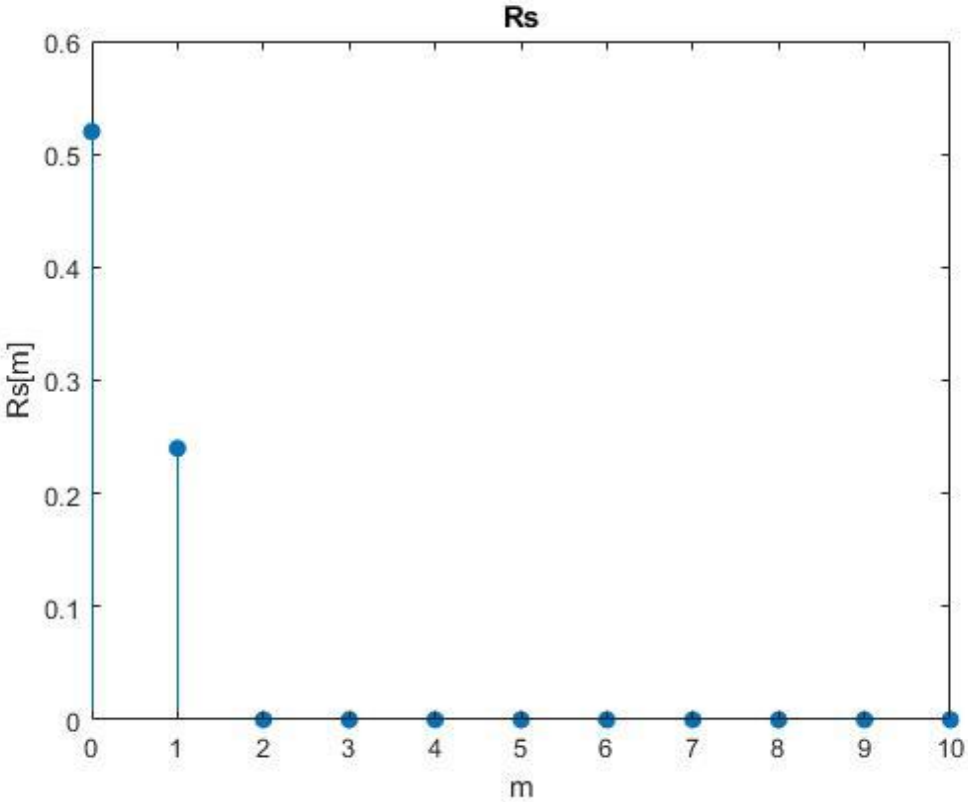
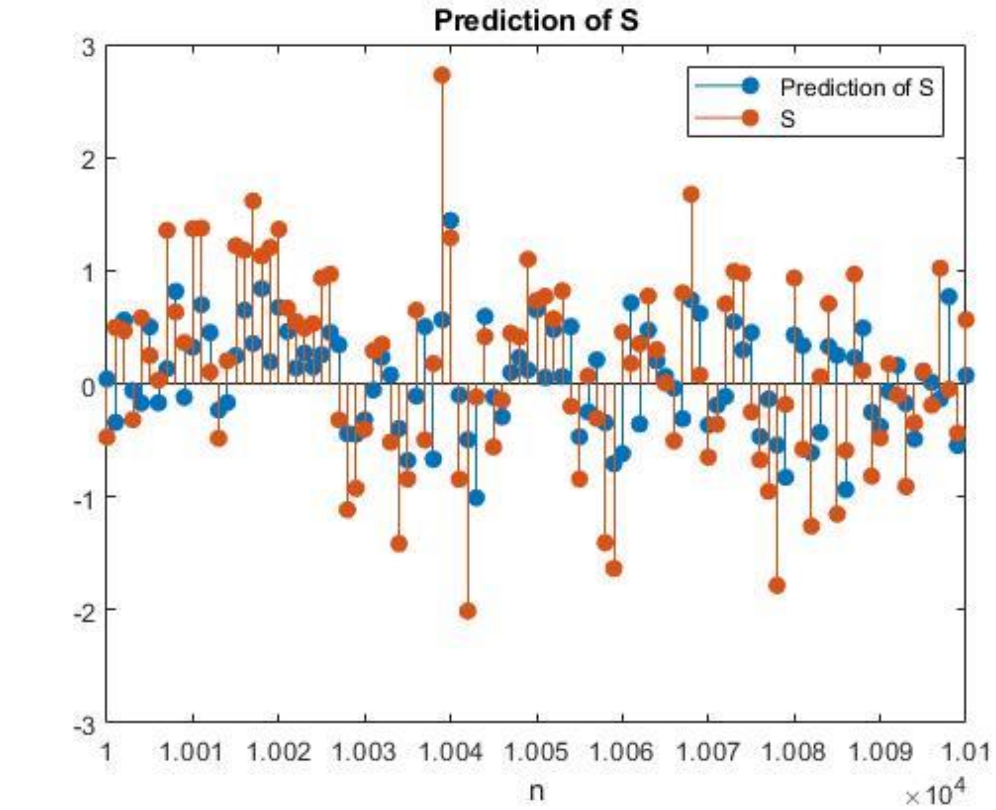
Stochastic Process Programming Homework

309513047 沈衍薰

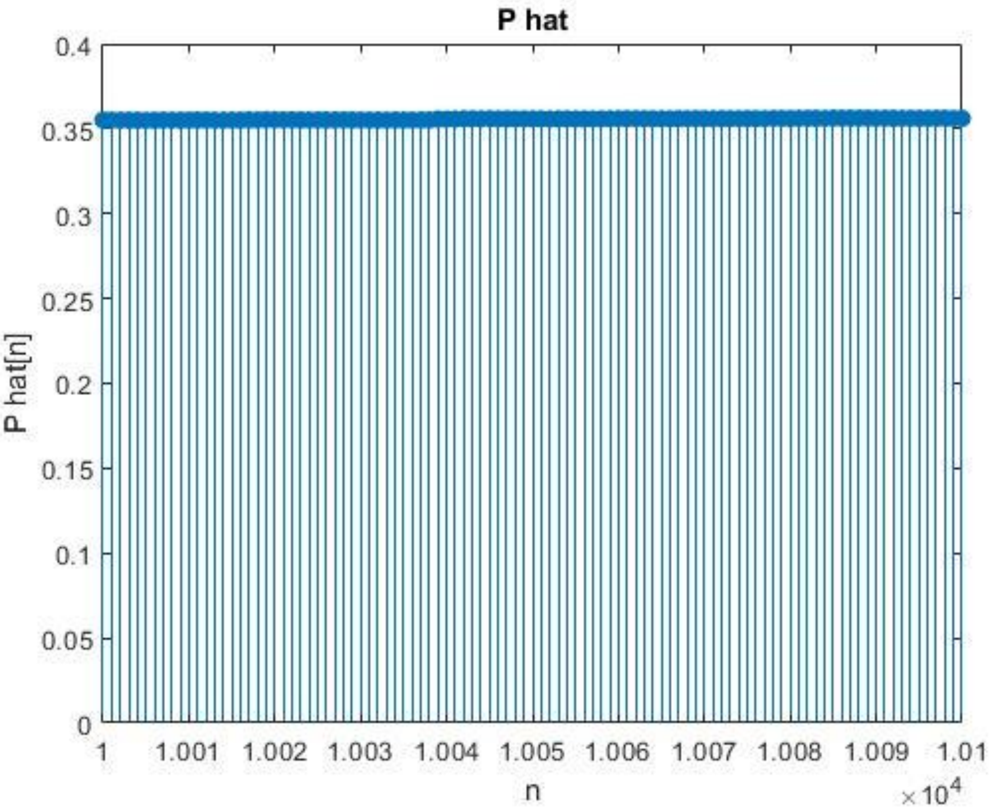
(1)



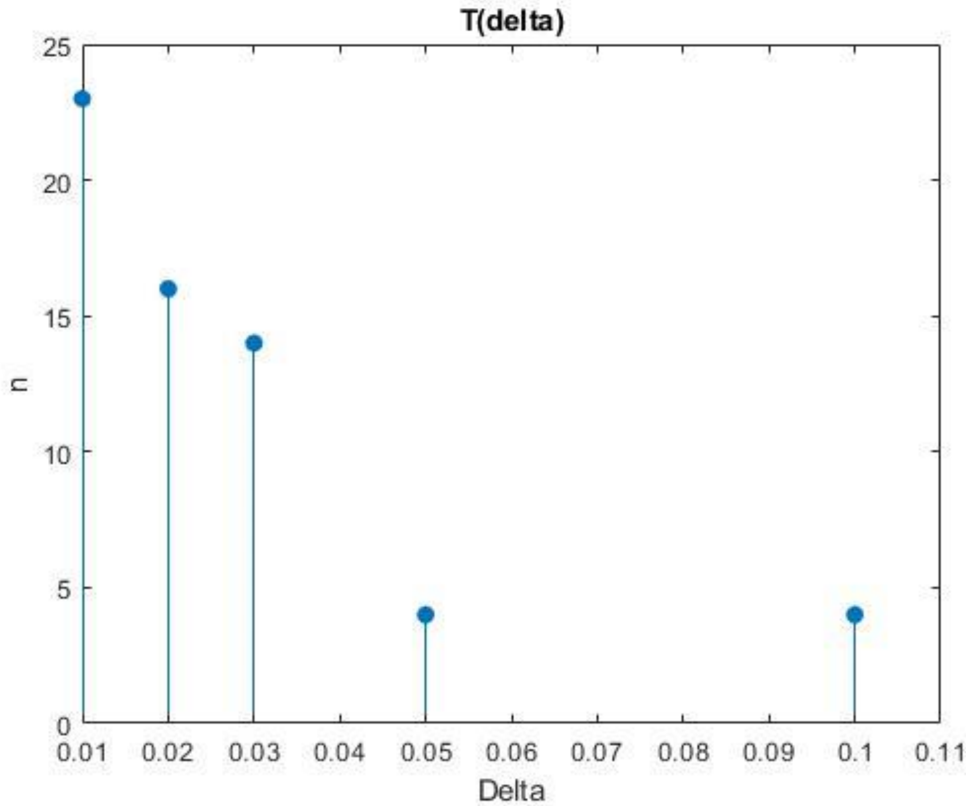
(2)


Prediction of S
Prediction of S
S
3
2
1
0
-1
-2
-3
1
1.001
1.002
1.003
1.004
1.005
1.006
1.007
1.008
1.009
1.01
n
×10^4

(3)


P hat
P hat[n]
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1
1.001
1.002
1.003
1.004
1.005
1.006
1.007
1.008
1.009
1.01
n
×10^4

(4)


T(delta)
25
20
15
10
5
0
n
0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 0.11
Delta

(5)

In my program, I define an array in the domain of 0~100001, then I move the origin from 0 to 50001, so, I can generate S in the domain of -50000~0~50000. The reason is that I think 50000 is large enough to represent infinity.

Source code:

```
%% generate i and s
b0 = 0.6;
b1 = 0.4;
i = wgn(100001, 1, 0);
var_i = 1;

s = zeros(100001, 1);
s(1, 1) = 0;
for k = 2:100001
    s(k, 1) = b0*i(k, 1) + b1*i(k-1, 1);
end

% move origin to 50001
origin = 50001;
```

For question 1, I generate $R_s[m]$ by MATLAB program based on following statement:

(1)

$$R_{ss}[m] = E\{s[n]\, s[n+m]\}$$

$$= E\{(b_0 \cdot i[n] + b_1 \cdot i[n-1])(b_0\, i[n+m] \cdot b_1\, i[n+m-1])\}$$

$$= b_0^2 E\{i[n] \cdot i[n+m]\} + b_0 b_1 E\{i[n]\, i[n+m-1]\}$$

$$+ b_0 b_1 E\{i[n-1]\, i[n+m]\} + b_1^2 E\{i[n-1] \cdot i[n+m-1]\}$$

$$\Rightarrow R_{ss}[0] = b_0^2 E\{(i[n])^2\} + b_1^2 E\{(i[n-1])^2\}$$

$$= b_0^2 + b_1^2 = 0.36 + 0.16 = 0.52$$

$$R_{ss}[1] = b_0 b_1 E\{(i[n])^2\}$$

for any $m > 1$, $R_{ss}[m] = 0$

Source code:

```
%% compute Rs[m]
Rs = [];
n = 100;
for m = 0:10
    if m == 0
        Rs(m+1) = b0*b0*var_i+b1*b1*var_i;

    elseif (m == 1)
        Rs(m+1) = b0*b1*var_i;

    else
        Rs(m+1) = 0;

    end
end
figure, stem(0:1:10,Rs, 'filled'), title('Rs'), xlabel('m'),
ylabel('Rs[m]');
```

For question 2, I generate $\hat{S}$ by MATLAB program based on following statement:

(2) $\hat{s}[n] = \hat{E}\{s[n] \mid s[n-k],\ 1 \le k \le 10\}$

$= \sum_{k=1}^{10} l[k] \cdot i[n-k]$, where $10000 \le n \le 10100$

$s[n] \rightarrow \boxed{\frac{1}{L(z)}} \xrightarrow{i[n]} \boxed{H(z)} \rightarrow \hat{s}[n]$

$\Rightarrow L(z) = \frac{S(z)}{I(z)} = b_0 + b_1 z^{-1} \quad \Rightarrow l[n] = b_0 \delta[n] + b_1 \delta[n-1]$

for $k=1$, $l[1] = b_1 \delta[0]$, and for any $2 \le k \le 10$, $l[k] = 0$

$\Rightarrow \hat{s}[n] = l[1]\, i[n-1] = b_1 i[n-1]$

However, $s[n] = b_0 i[n] + b_1 i[n-1]$, so, there is always the difference between $s[n]$ and $\hat{s}[n]$, this difference equals to $b_0 \cdot i[n]$

Source code:

```
%% compute estimation of s[n]
estimate_s = zeros(100001, 1);
l = zeros(100001, 1);
l(1, 1) = b1;

for n = 0:50000
    for k = 1:10
        estimate_s(n + origin, 1) = estimate_s(n + origin, 1) + l(k,
1)*i(n + origin - 1);
    end
end

figure, stem(10000:1:10100, [estimate_s((origin+10000:origin+10100),
1), s((origin+10000:origin+10100), 1)], 'filled'), legend('Prediction
of S', 'S');
title('Prediction of S'), xlabel('n');
```

For question 3, based on the following equation:

$$\hat{P}[n] = \frac{1}{n}\sum_{k=1}^{n}(\hat{s}[k] - s[k])^2.$$

By using **S** and $\hat{S}$ defined in question 1, we can compute $\hat{P}$ in the domain of 10000~10100.

Source code:

```
%% compute P
P = zeros(100001, 1);
for n = 1:50000
    for k = 1:n
        P(origin + n, 1) = P(origin + n, 1) + (estimate_s(origin + k,
1) - s(origin + k, 1))^2;
    end
    P(origin + n, 1) = P(origin + n, 1)/n;
end
figure, stem(10000:1:10100, P((origin+10000:origin+10100), 1),
'filled'), title('P hat'), xlabel('n'), ylabel('P hat[n]');
```

For question 4, based on the following definition:

$T(\delta): |\hat{p}[n] - \hat{p}[n-1]| \le \delta \cdot \hat{P}[n-1]$

$$\forall \delta \in \{0.1, 0.05, 0.03, 0.02, 0.01\}$$

Combining with $\hat{P}$ we generate in question 4, we can get the minimum n to satisfy this definition of T.

Source code:

```
%% compute T
T = zeros(5, 1);
delta = [0.01 0.02 0.03 0.05 0.1];

for i = 1:5
    for n = 2:50000
        if(abs(P(origin+n, 1)-P(origin+n-1, 1)) <=
delta(i)*P(origin+n-1, 1))
            T(i, 1) = n;
            break;
        end
    end
end

figure, stem(delta, T, 'filled'), title('T(delta)'), xlabel('Delta'),
ylabel('n');
```