ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ

6η Ομάδα Ασκήσεων

Νικόλαος Δημητριάδης
03114016
ΗΜΜΥ 8ο

1 Μηχανισμός ελέγχου ροής παραθύρου

ερώτημα 1

Για τις τιμές των παραμέτρων της εκφώνησης, τα ζητούμενα διαγράμματα είναι τα ακόλουθα:

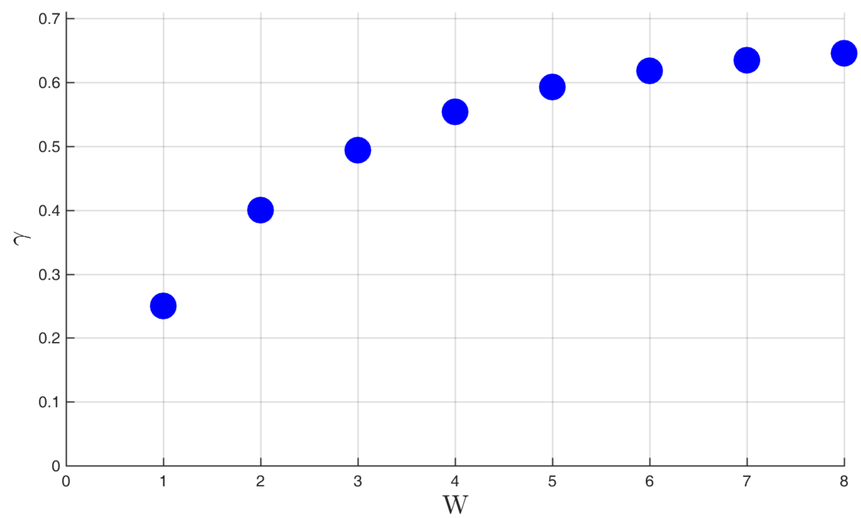

Figure 1: ρυθμαπόδοση (throughput) του συστήματος ως συνάρτηση του αριθμού των πακέτων στο σύστημα (μέγεθος παραθύρου W)

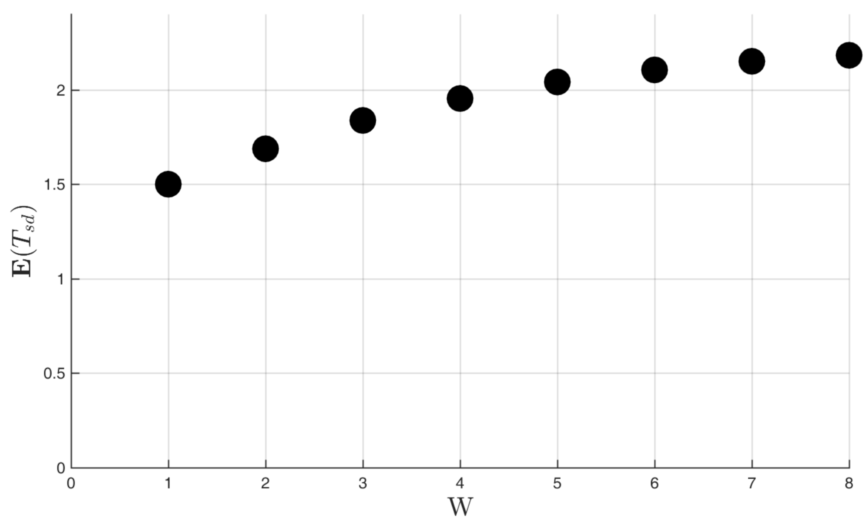

Figure 2: Μέσος χρόνος καθυστέρησης του συστήματος από το S μέχρι το D ως συνάρτηση του αριθμού των πακέτων στο σύστημα

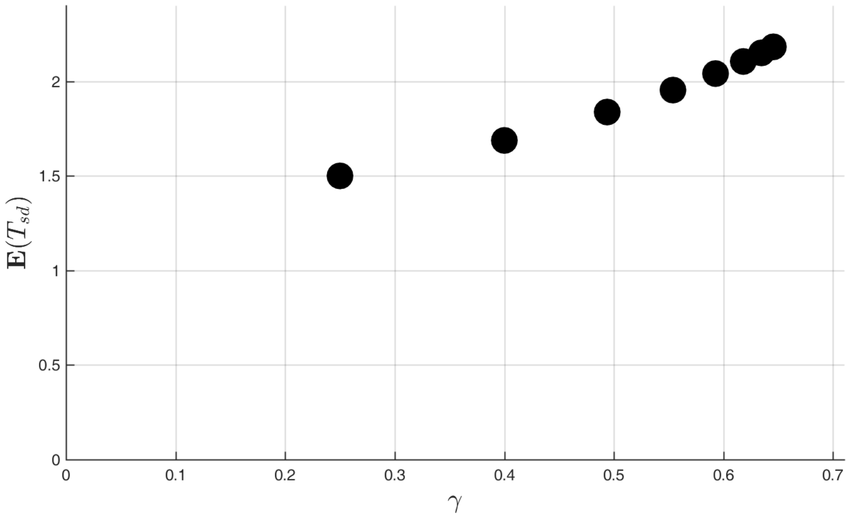

Figure 3: Μέσος χρόνος καθυστέρησης του συστήματος από το S μέχρι το D ως συνάρτηση της ρυθμαπόδοσης του συστήματος

Παρατηρούμε ότι με την αύξηση του παραθύρου W, ο μέσος χρόνος καθυστέρησης του συστήματος από το S μέχρι το D και η ρυθμαπόδοση γ του συστήματος τείνουν να σταθεροποιηθούν.

ερώτημα 2

Σε αυτό το σημείο θεωρούμε ότι οι παράμετροι μεταβάλλονται σύμφωνα με το k με τον ακόλουθο τρόπο:

$$\begin{bmatrix} k\lambda \\ k\mu_1 \\ k\mu_2 \\ k\mu_3 \\ k\mu_r \end{bmatrix} \qquad k = 1, 2, 3, 4, 5 \tag{1}$$

Επιπλέον, οι τιμές των παραμέτρων διέπονται από την ακόλουθη σχέση:

$$\lambda X_0 = \mu_1 X_1 = \mu_2 X_2 = \mu_3 X_3 = \mu_r X_r \qquad \text{με } X_0 = 1 \tag{2}$$

Από τη σχέση αυτή συμπεραίνουμε ότι οι παράμετροι X_i, με $i = 1, 2, 3, 4, r$ αυξάνονται αναλογικά. Με άλλα λόγια, για οποιοδήποτε k παραμένουν σταθεροί. Από τον ορισμό της σταθεράς κατακερματισμού $G(N)$ βρίσκουμε ότι αυτή εξαρτάται μόνο από τις παραμέτρους X_i και, συνεπώς, παραέμενει αμετάβλητη. Αυτό σημαίνει ότι:

- ο βαθμός χρησιμοποίησης U παραμένει σταθερός $\forall k$, καθώς εξαρτάται από X_i και $G(N)$,
- ο μέσος αριθμός πελατών Q παραμένει σταθερός $\forall k$, καθώς εξαρτάται από X_i και $G(N)$,
- η ρυθμαπόδοση γ μεταβάλλεται, αφού εξαρτάται από τους ρυθμούς λ, μ_i. Αυτό φαίνεται και στο αντίστοιχο διάγραμμα παρακάτω.

- ο μέσος χρόνος καθυστέρησης $\mathbb{E}[T]$ μεταβάλλεται, αφού εξαρτάται από τους ρυθμούς λ, μ_i.

Τα αποτελέσματα παρουσιάζονται σε μορφή διαγράμματος για τα ζητούμενα μεγέθη. Επιλέγεται η παρουσίαση των ίδιων διαγραμμάτων με το προηγούμενο ερώτημα:

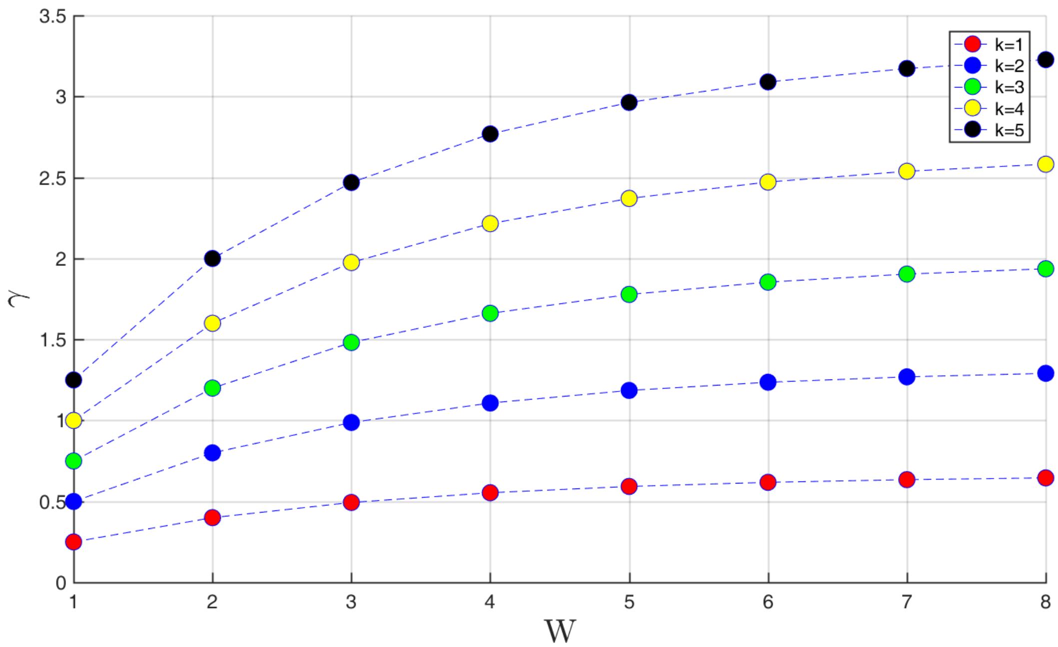

Figure 4: ρυθμαπόδοση (throughput) του συστήματος
ως συνάρτηση του αριθμού των πακέτων στο σύστημα (μέγεθος παραθύρου W)

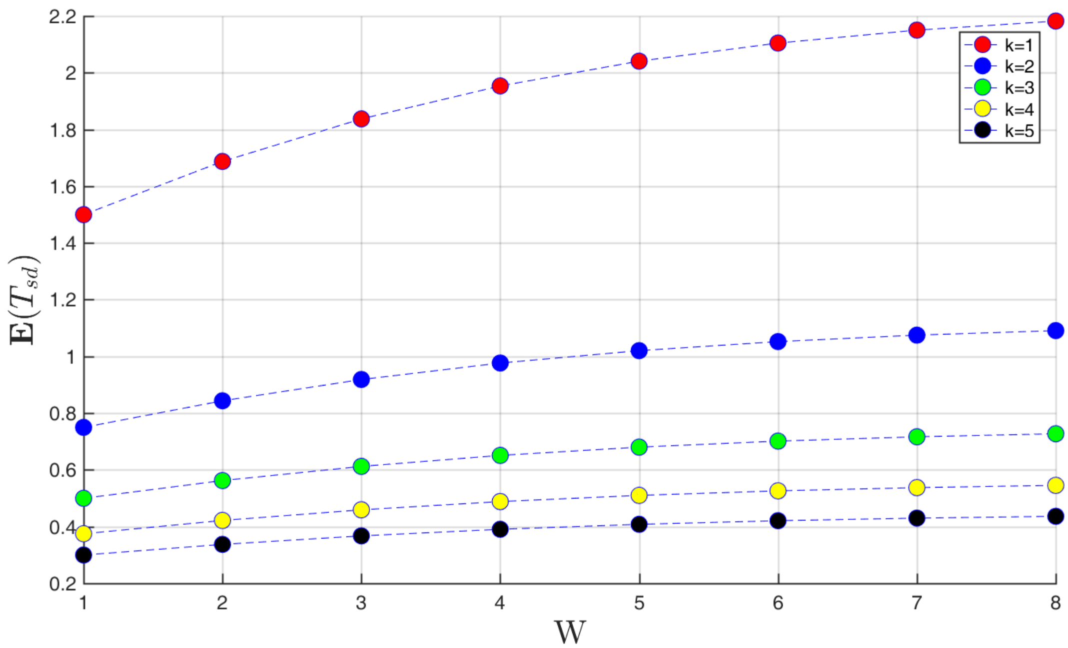

Figure 5: Μέσος χρόνος καθυστέρησης του συστήματος από το S μέχρι το D ως συνάρτηση του αριθμού των πακέτων στο σύστημα

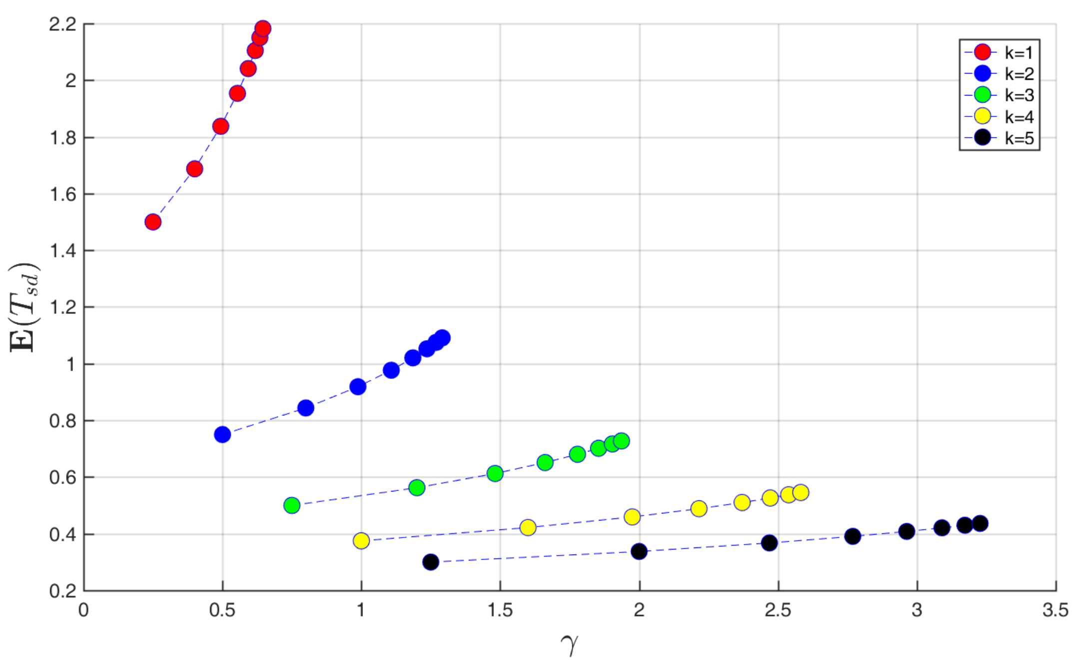

Figure 6: Μέσος χρόνος καθυστέρησης του συστήματος από το S μέχρι το D ως συνάρτηση της ρυθμαπόδοσης του συστήματος

Πράγματι, ρυθμαπόδοση γ μεταβάλλεται για τις διάφορες τιμές του k

2 Ο αλγόριθμος του Buzen

ερώτημα 1

Από το κλειστό δίκτυο της εκφώνησης προκύπτουν οι ακόλουθες εξισώσεις

$$\mu_1 X_1 = (1-p)\mu_1 X_1 + \mu_2 X_2 \tag{3}$$

$$\mu_2 X_2 = p\mu_1 X_1 \tag{4}$$

$$(4) \overset{X_1=1}{\Longrightarrow} X_2 = \frac{p\mu_1 \cdot 1}{\mu_2} = 0.6 \tag{5}$$

Άρα

$$\boxed{X_1 = 1} \text{ και } \boxed{X_2 = 0.6} \tag{6}$$

ερώτημα 2

Η συνάρτηση **buzen** προυσιάζεται παρακάτω:

Listing 1: buzen

```
function G_N = buzen( N, M , X)

        for n = 1:N
            g(n,1) = 1;
        end

        for m = 1:M
            g(1,m) = 1;
        end

        for m = 2:M
            for n = 2:N
                g(n,m) = g(n,m-1) + X(m)*g(n-1,m);
            end
        end

        G_N = g(N,M);

end
```

ερώτημα 3

Σύμφωνα με τα προηγούμενα προκύπτουν τα ακόλουθα διαγράμματα:

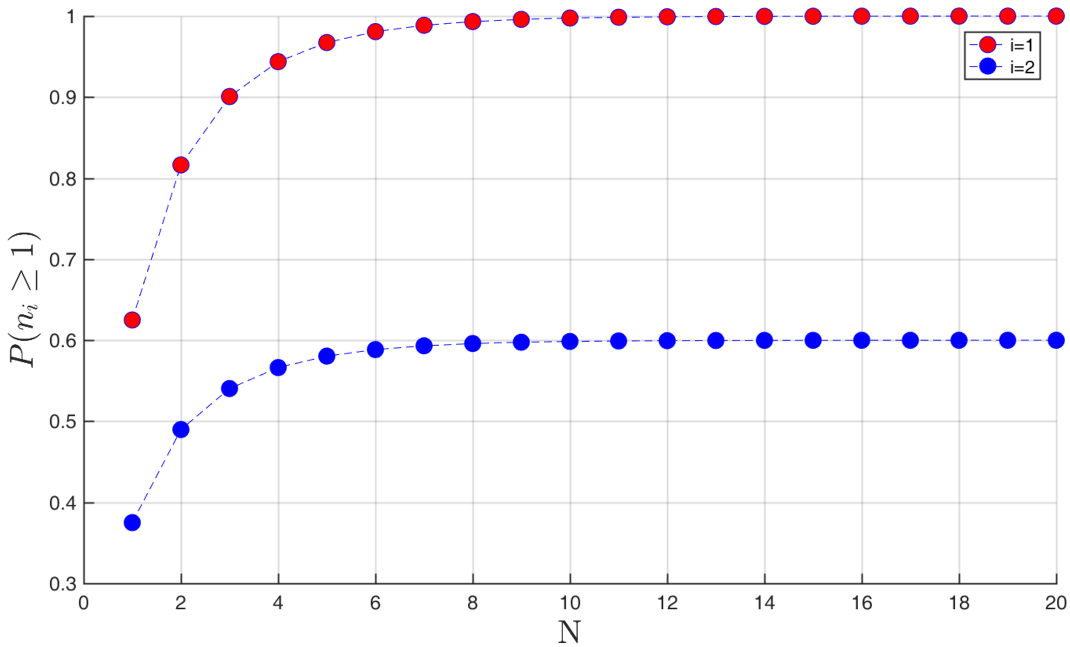

Figure 7: Βαθμός χρησιμοποίησης (utilization) των δύο ουρών ως συνάρτηση του αριθμού πελατών N σε κοινό διάγραμμα αξόνων

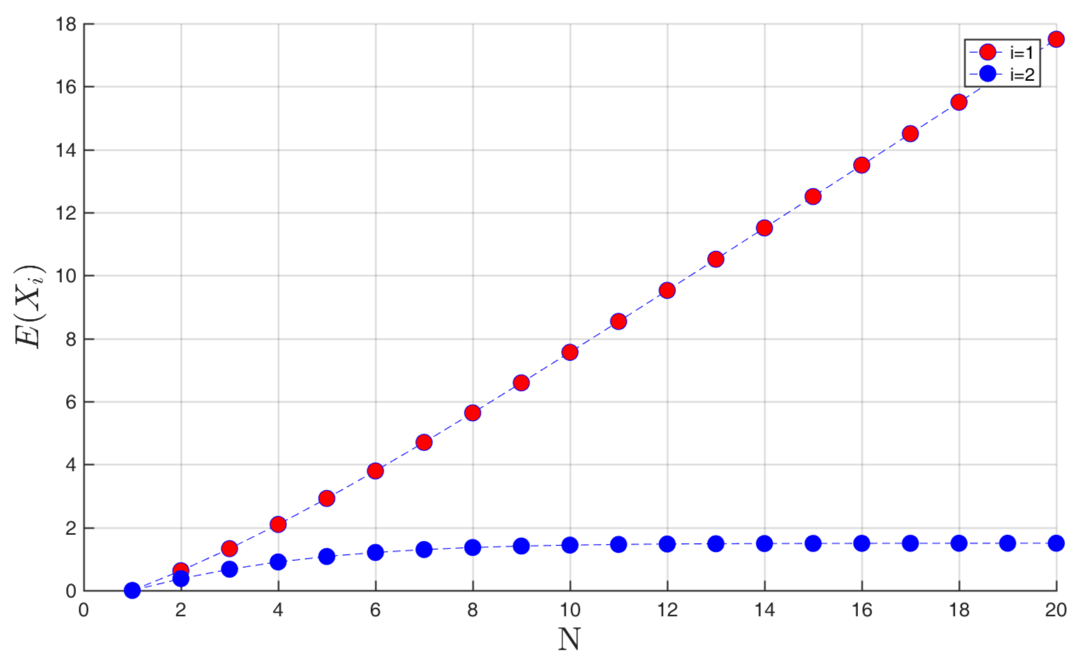

Figure 8: Το μέσο αριθμό πελατών στις δύο ουρές ως συνάρτηση του αριθμού πελατών N σε κοινό διάγραμμα αξόνων

Παρατηρούμε ότι με την αύξηση των πελατών στο κλειστό σύστημα, οι βαθμοί χρησιμοποίησης του συστήματος τείνουν να λάβουν τις τιμές των παραμέτρων X_i. Τέλος, εφόσον έχουμε κλειστό σύστημα, το πλήθος των πελατών παραμένει σταθερό. Αυτό είναι εμφανές και στο παραπάνω σχήμα.

3 Προσομοίωση σε κλειστό δίκτυο εκθετικών ουρών αναμονής

Ως διάνυσμα κατάστασης επιλέγεται το $\mathbf{n} = (n_1, n_2, n_3)$.
Το διάγραμμα ρυθμού μεταβάσεων είναι:

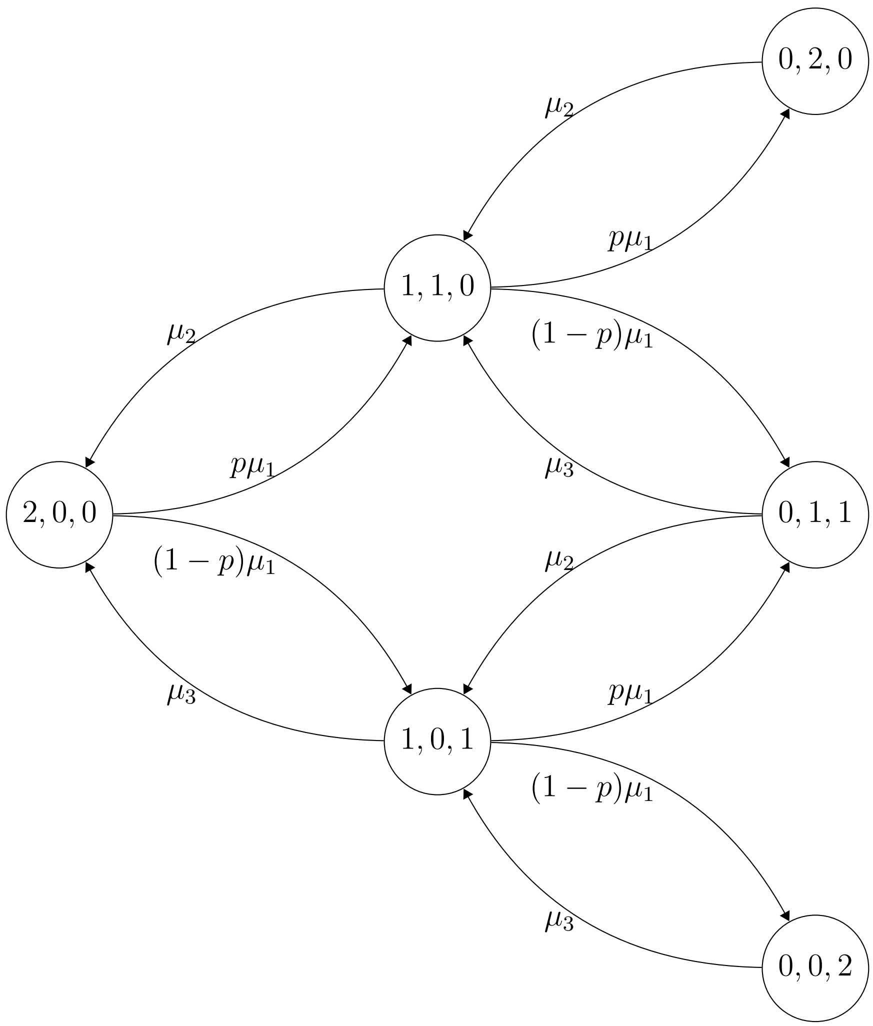

Figure 9: Διάγραμμα ρυθμού μεταβάσεων συστήματος

ερώτημα 1

Μέσω της προσομοίωσης βρίσκουμε τις ακόλουθες εργοδικές πιθανότητες:

$$\begin{bmatrix} P_{(2,0,0)} \\ P_{(1,0,1)} \\ P_{(1,1,0)} \\ P_{(0,1,1)} \\ P_{(0,2,0)} \\ P_{(0,0,2)} \end{bmatrix} = \begin{bmatrix} 0.5520 \\ 0.1664 \\ 0.1475 \\ 0.0448 \\ 0.0389 \\ 0.0504 \end{bmatrix} \tag{7}$$

ερώτημα 2

Ο μέσος αριθμός πελατών σε κάθε ουρά του συστήματος, όπως αυτός εξελίσσεται κατά της διάρκεια της προσομοίωσης παρουσιάζεται παρακάτω:

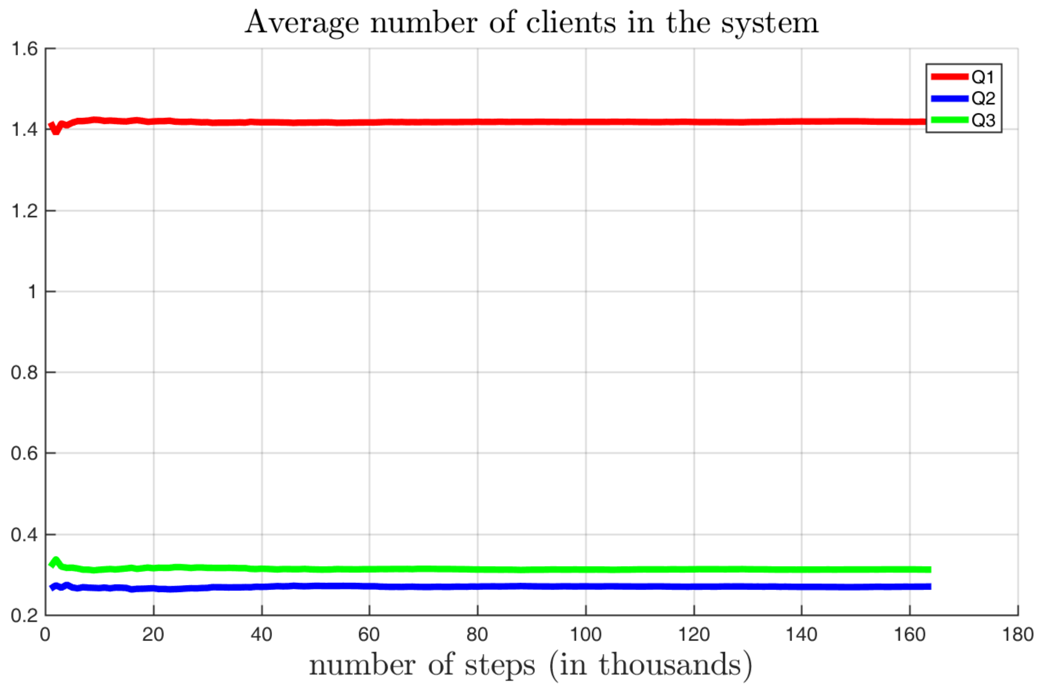

Figure 10: Μέσος αριθμός πελατών σε κάθε ουρά του συστήματος, όπως αυτός εξελίσσεται κατά της διάρκεια της προσομοίωσης

Παρατηρούμε ότι το άθροισμα των τριών μέσων όρων όταν το σύστημά έχει φτάσει στη σύγκλιση ισούται με $N = 2$.

4 Appendix

Listing 2: `OCTAVE` initialization

```
W = 8;
S = 1./[1, 2, 2, 2, 2/3];
P = [ 0 1 0 0 0 ; 0 0 1 0 0 ; 0 0 0 1 0 ; 0 0 0 0 1 ; 1 0 0 0 0 ];
V = qncsvisits(P);


for i = 1:W
  % X = throughput
  [U R Q X] = qnclosed( i , S , V );
  gamma(i) = X(1);
  E_T_sd(i) = 0;
  for j = 2:4
    E_T_sd(i) = E_T_sd(i) + R(j) ;
  endfor
%  disp(E_T_sd)
endfor

% PLOTS



%% erwthma 2


for i = 1:W
  for k = 1:5
    S_now = S / k;
  % X = throughput
    [U R Q X] = qnclosed( i , S_now , V );
    U_analytics(i,k,:) = U;
    Q_analytics(i,k,:) = Q;
    gamma2(i,k) = X(1);
    E_T_sd2(i,k) = 0;
    for j = 2:4
      E_T_sd2(i,k) = E_T_sd2(i,k) + R(j) ;
    endfor
  endfor
endfor
```

Listing 3: `part 1`

```
close all;
```

```
clear all;
clc;
%%
% TAKEN gamma, E_T_sd from Octave
gamma = [  0.25000,  0.40000,  0.49383,  0.55385,  0.59271,  0.61803,
    0.63461,  0.64549 ];
E_T_sd = [  1.5000,  1.6875,  1.8375,  1.9537,  2.0410,  2.1049,  2.1504,
    2.1823 ];

fontsize = 12;
lineWidth = 1;
plotLineWidth = 4;
width=1024;
height=568;

fig1 = figure();
fig1.Color = 'w';
set(gcf,'units','pixels','position',[0,0,width,height]) ;
hold on
plot(gamma,'bo','MarkerSize',20,'MarkerFaceColor','b');
temp = gca;
temp.Color = 'w';
temp.LineWidth = lineWidth;
temp.GridColor = 'k';
% temp.GridAlpha = 0.5;
temp.FontSize = fontsize;
xlim([0 8]);
ylim([0 1.1*max(gamma)]);
hold off


ylabel(['$\gamma$'],'FontSize',20,'Interpreter','latex');
xlabel(['W'],'FontSize',20, 'Interpreter','latex');
grid on;
saveas(gcf,'figure1.png');

%%
fig2 = figure();
fig2.Color = 'w';
set(gcf,'units','pixels','position',[0,0,width,height]) ;
hold on
plot(E_T_sd,'ko','MarkerSize',20,'MarkerFaceColor','k');
temp = gca;
temp.Color = 'w';
temp.LineWidth = lineWidth;
```

```
temp.GridColor = 'k';
% temp.GridAlpha = 0.5;
temp.FontSize = fontsize;
xlim([0 8]);
ylim([0 1.1*max(E_T_sd)]);
hold off


ylabel(['$\mathbf{E}(T_{sd})$'],'FontSize',20,'Interpreter','latex');
xlabel(['W'],'FontSize',20, 'Interpreter','latex');
grid on;
saveas(gcf,'figure2.png');

%%
fig3 = figure();
fig3.Color = 'w';
set(gcf,'units','pixels','position',[0,0,width,height]) ;
hold on
plot(gamma, E_T_sd,'ko','MarkerSize',20,'MarkerFaceColor','k');
temp = gca;
temp.Color = 'w';
temp.LineWidth = lineWidth;
temp.GridColor = 'k';
% temp.GridAlpha = 0.5;
temp.FontSize = fontsize;
xlim([0 1.1*max(gamma)]);
ylim([0 1.1*max(E_T_sd)]);
hold off


ylabel(['$\mathbf{E}(T_{sd})$'],'FontSize',20,'Interpreter','latex');
xlabel(['$\gamma$'],'FontSize',20, 'Interpreter','latex');
grid on;
saveas(gcf,'figure3.png');
```

Listing 4: part 2

```
close all;
clear all;
clc;
%%
M = 2;
X = [1,0.6];
for N = 1:21
%     arithmhths = buzen(N,M,X);
```

```
    G_N(N) = buzen(N,M,X);       % N+1 giati h arithmhsh twn grammwn ston Buzen xekinaei apo to 0
end

for N = 1:20
    for i = 1:M
        E_n(i,N) = 0;
        for k = 1:(N-1)
           E_n(i,N) = E_n(i,N) + X(i)^k * G_N(N-k) / G_N(N);
        end
        P(i,N) = X(i) * G_N(N) / G_N(N+1);
    end
    pop(N) = E_n(1,N)+E_n(2,N);
end

%%
fontsize = 12;
lineWidth = 1;
plotLineWidth = 4;
width=1024;
height=568;
colors = 'rbgyk';
fig1 = figure();
fig1.Color = 'w';
set(gcf,'units','pixels','position',[0,0,width,height]) ;
hold on
for k = 1:M
    plot(P(k,:),'b--o','MarkerSize',10,'MarkerFaceColor',colors(k));
end
temp = gca;
temp.Color = 'w';
temp.LineWidth = lineWidth;
temp.GridColor = 'k';
% temp.GridAlpha = 0.5;
temp.FontSize = fontsize;
hold off


ylabel(['$P(n_{i}\geq 1)$'],'FontSize',20,'Interpreter','latex');
xlabel(['N'],'FontSize',20, 'Interpreter','latex');
legend('i=1','i=2');
grid on;
saveas(gcf,'figure_2a.png');
```

```
%%
colors = 'rbgyk';
fig1 = figure();
fig1.Color = 'w';
set(gcf,'units','pixels','position',[0,0,width,height]) ;
hold on
for k = 1:M
    plot(E_n(k,:),'b--o','MarkerSize',10,'MarkerFaceColor',colors(k));
end
temp = gca;
temp.Color = 'w';
temp.LineWidth = lineWidth;
temp.GridColor = 'k';
% temp.GridAlpha = 0.5;
temp.FontSize = fontsize;
hold off


ylabel(['$E(X_i)$'],'FontSize',20,'Interpreter','latex');
xlabel(['N'],'FontSize',20, 'Interpreter','latex');
legend('i=1','i=2');
grid on;
saveas(gcf,'figure_2b.png');
```

Listing 5: part 3

```
% Example: two queues in a row. The output of the first queue is the input of the second.
% The output of the second queue is the input of the first queue.

clc;
clear all;
close all;
N = 3;
mu1 = 2; % queue 1
mu2 = 3; % queue 2
mu3 = 4; % queue 3
p = 0.4;
rng(1)
arrivals(200) = 0;
arrivals(110) = 0;
arrivals(011) = 0;
arrivals(020) = 0;
arrivals(002) = 0;
arrivals(101) = 0;
```

```
total_arrivals = 0;
%%
% threshold definition
threshold = mu1/(mu1 + mu2);
% system starts at state 3
current_state = 200;
% count the time steps of the simulation
steps = 0;

previous_mean1 = 0;
previous_mean2 = 0;
previous_mean3 = 0;
% times checked for convergence
times = 0;

while steps < 300000
  steps = steps + 1;
  % every 1000 steps check for convergence
  if mod(steps,1000) == 0
    times = times + 1;


    %
       %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % total time in every state
    T200 = 1/mu1            * arrivals(200);
    T002 = 1/mu3            * arrivals(002);
    T020 = 1/mu2            * arrivals(020);
    T101 = 1/(mu1 + mu3)    * arrivals(101);
    T110 = 1/(mu1 + mu2)    * arrivals(110);
    T011 = 1/(mu2 + mu3)    * arrivals(011);

    % total time in all states
    total_time = T200 + T002 + T020 + T101 + T110 + T011;
    % Probability of every state
    P(200) = T200/total_time;
    P(002) = T002/total_time;
    P(020) = T020/total_time;
    P(110) = T110/total_time;
    P(101) = T101/total_time;
    P(011) = T011/total_time;

    % mean number of clients in queues 1 and 2
    current_mean1 = P(101) + P(110) + 2*P(200);
```

```
  current_mean2 = P(011) + P(110) + 2*P(020);
  current_mean3 = P(101) + P(011) + 2*P(002);

  clients_1(times) = current_mean1;
  clients_2(times) = current_mean2;
  clients_3(times) = current_mean3;

  % check both queues for convergence
  if abs(current_mean1 - previous_mean1)<0.00001 && abs(current_mean2 - previous_mean2) && abs(current_mean3 - previous_mean3) < 0.00001
    break;
  end

  previous_mean1 = current_mean1;
  previous_mean2 = current_mean2;
  previous_mean3 = current_mean3;

end

arrivals(current_state) = arrivals(current_state) + 1;
total_arrivals = total_arrivals + 1;

% get a random number from uniform distribution
random_number = rand(1);
if current_state == 200
    if (random_number < p)
        current_state = 110;
    else
        current_state = 101;
    end
elseif current_state == 110
  if random_number < mu2 / (mu1 + mu2)
    current_state = 200;
  else
      if random_number < (mu2 + p * mu1) / (mu1 + mu2)
          current_state = 020;
      else
          current_state = 011;
      end
  end
elseif current_state == 101
  if random_number < mu3 / (mu1 + mu3)
    current_state = 200;
  else
      if random_number < (mu3 + p * mu1) / (mu1 + mu3)
```

```
            current_state = 011;
        else
            current_state = 002;
        end
    end
  elseif current_state == 020
    current_state = 110;
  elseif current_state == 002
    current_state = 101;
  else
    if random_number < mu2 / (mu2 + mu3)
      current_state = 101;
    else
      current_state = 110;
    end
  end

end


pop = [P(200)  P(101) P(110)  P(011)  P(020) P(002) ];
P(200)
P(020)
P(002)
P(110)
P(101)
P(011)
clients_1(times)
clients_2(times)
clients_3(times)




%%
fontsize = 12;
lineWidth = 1;
plotLineWidth = 4;
width=1024;
height=568;
colors = 'rbgyk';
fig1 = figure();
fig1.Color = 'w';
set(gcf,'units','pixels','position',[0,0,width,height]) ;
hold on
```

```
plot(clients_1,'r','linewidth',plotLineWidth);
plot(clients_2,'b','linewidth',plotLineWidth);
plot(clients_3,'g','linewidth',plotLineWidth);
temp = gca;
temp.Color = 'w';
temp.LineWidth = lineWidth;
temp.GridColor = 'k';
% temp.GridAlpha = 0.5;
temp.FontSize = fontsize;
hold off

legend('Q1','Q2','Q3');
title(['Average number of clients in the system'],'FontSize',20,'Interpreter'
    ,'latex');
xlabel(['number of steps (in thousands)'],'FontSize',20, 'Interpreter','latex
    ');
grid on;
saveas(gcf,'figure3_1.png');



%%

for i = 1:times
    t(i) = clients_1(i)+clients_2(i)+clients_3(i);
end

t(times)
```