ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Μάθημα: Συστήματα Αναμονής

Ονοματεπώνυμο: Ειρήνη Δόντη

Α.Μ: 03119839

2η Ομάδα Ασκήσεων

Αθήνα 2022

Θεωρητική μελέτη της ουράς M/M/1

(α)

Η ουρά M/M/1 είναι εργοδική, όταν ο λόγος $\frac{\lambda}{\mu}$ είναι μικρότερος της μονάδας. Με άλλα λόγια, πρέπει η συνθήκη Erlang να ικανοποιεί τη σχέση ρ = $\frac{\lambda}{\mu}$ < 1 όπου η ποσότητα ρ εκφράζει τον βαθμό χρησιμοποίησης (ή ισοδύναμα την ένταση κυκλοφορίας), δηλαδή την πιθανότητα να μην είναι άδειο το σύστημα ή να είναι απασχολημένη η μονάδα εξυπηρέτησης.

Οι εργοδικές οριακές πιθανότητες προκύπτουν από τις γραμμικά ανεξάρτητες εξισώσεις ισορροπίας:

$$(\lambda_k + \mu_k)P_k = \lambda_{k-1}P_{k-1} + \mu_{k+1}P_{k+1}, k > 1 \text{ (1)}$$

$$\lambda_0 P_0 \ = \ \mu_1 P_1 \text{ (2)}$$

$$P_0 + P_1 + \ldots + P_k + \ldots = \ 1 \text{ ή } \sum_{k=0}^{\infty} P_k \ = \ 1 \text{ (3)}$$

Οι παράμετροι λ_k είναι ο ρυθμός αφίξεων, ενώ οι παράμετροι μ_k είναι ο ρυθμός εξυπηρετήσεων όταν η διαδικασία βρίσκεται στην κατάσταση k (είναι ανεξάρτητες του χρόνου).

Από τη σχέση (2), η γενική λύση του συστήματος εξισώσεων είναι:

$$P_k \ = \ P_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}, \text{ k = 1,2,… (4)}$$

Από τη σχέση (3) και (4) ισχύει: $\sum_{k=0}^{\infty} P_k \ = \ 1$ ή $\displaystyle\sum_{k=0}^{\infty} P_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \ = \ 1$ (5)

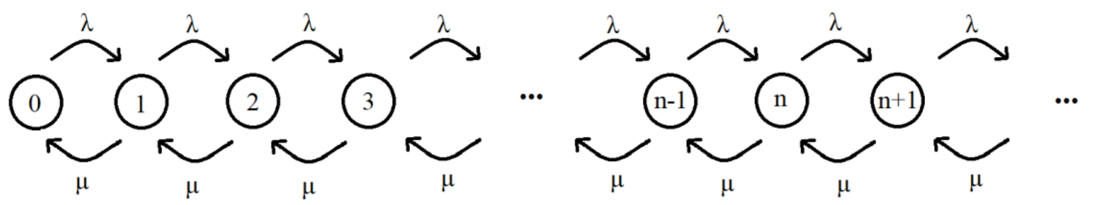
Η εφαρμογή σε απλή ουρά Μ/Μ/1, αποτελεί απλό σύστημα αναμονής, καθώς οι ρυθμοί αφίξεων-αναχωρήσεων είναι ανεξάρτητοι από την κατάσταση του συστήματος, δηλαδή $\lambda_k = \lambda$ & $\mu_k = \mu$ για κάθε k = 1,2,3,…. Επίσης, ισχύει η συνθήκη Erlang για την εργοδικότητα συστήματος ρ = $\frac{\lambda}{\mu}$ < 1. Έπειτα, από τη σχέση (4) ισχύει ότι $P_k = P_0 \left(\frac{\lambda}{\mu}\right)^k = P_0\rho^k$ (6). Από τη σχέση (5): $P_0 \sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k = 1$ (7)

Βάσει της συνθήκης Erlang για την εργοδικότητα συστήματος ρ = $\frac{\lambda}{\mu}$ < 1, ισχύει ότι:

$\sum_{k=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^k = \frac{1}{1-\frac{\lambda}{\mu}} = \frac{1}{1-\rho}$ (8) και συνεπώς από τη σχέση (6): $P_0 = 1\text{-}\rho$ (9).

Οπότε, από τη σχέση (6): οι εργοδικές πιθανότητες των καταστάσεων του συστήματος είναι: $P_k = P_0 \left(\frac{\lambda}{\mu}\right)^k = (1-\rho)\rho^k$, k=0,1,2,…

Σχεδιάζουμε το διάγραμμα ρυθμού μεταβάσεων της ουράς Μ/Μ/1 μεταξύ εργοδικών καταστάσεων:



(β)

Δίνεται ότι, για τις ουρές Μ/Μ/1 που βρίσκονται σε κατάσταση ισορροπίας, ισχύει η ακόλουθη ισότητα:

E[n(t)] = $\sum_{k=1}^{\infty} k\, P_k = \frac{\rho}{1-\rho}$, όπου ρ = $\frac{\lambda}{\mu}$. Ο μέσος χρόνος καθυστέρησης από τον τύπο του Little είναι: E[T] = $\frac{E[n(t)]}{\lambda} = \frac{1/\mu}{1-\rho}$. Οπότε, ο μέσος χρόνος αναμονής Μ/Μ/1 είναι: E[W] = E[T] – 1/μ = $\frac{1/\mu}{1-\rho} - 1/\mu = \frac{\rho}{\mu(1-\rho)}$.

(γ)

Θεωρούμε ότι το σύστημα πρέπει να έχει k = 57 πελάτες. Επιλέγουμε τρεις τιμές του ρ οι οποίες, εξ ορισμού, πρέπει να είναι μικρότερες της μονάδας, οπότε βρίσκουμε την παρακάτω πιθανότητα P_{57} για τις διάφορες τιμές του ρ:

Για ρ = 0.3:

$P_k = (1-\rho)\rho^k = P_{57} = (1-0.3)0.3^{57} = 1{,}01\text{x}10^{-30}$

Για ρ = 0.6:

$P_k = (1-\rho)\rho^k = P_{57} = (1-0.6)0.6^{57} = 9.05\ \text{x}10^{-14}$

Για ρ = 0.9:

$P_k = (1-\rho)\rho^k = P_{57} = (1-0.9)0.9^{57} = 2.465\text{x}10^{-4}$

Παρατηρούμε ότι μπορεί να βρεθεί το σύστημα μας με 57 πελάτες. Όσο το ρ είναι πολύ μικρότερο από τη μονάδα, η πιθανότητα P_{57} είναι αμελητέα.

Ανάλυση ουράς M/M/1 με Octave

(α)

Όπως αναλύσαμε παραπάνω, ένα σύστημα είναι εργοδικό όταν ρ = $\frac{\lambda}{\mu}$ < 1. Επειδή οι αφίξεις στην ουρά ακολουθούν κατανομή Poisson με μέσο ρυθμό λ=5 πελάτες/min, τότε οι ρυθμοί εξυπηρέτησης πρέπει να ικανοποιούν την ανισότητα μ > λ ή μ > 5 πελάτες/min. Επίσης, έχουμε από την εκφώνηση τον περιορισμό ότι 0 ≤ μ ≤10 πελάτες/min.

Οπότε, οι ρυθμοί εξυπηρέτησης που είναι αποδεκτοί για το σύστημά μας είναι:
5<μ≤10 πελάτες/min.

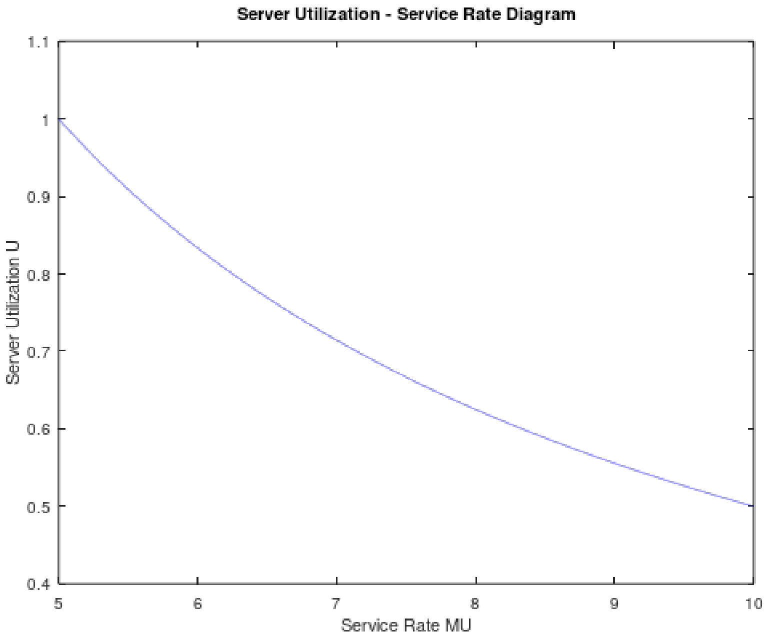
(β)

Χρησιμοποιούμε την εντολή qsmm1 του πακέτου queueing, ώστε να υπολογίσει τις παραμέτρους που χρειαζόμαστε για τη συνέχεια της άσκησης (π.χ. Q, R, U, X κ.λ.π.).

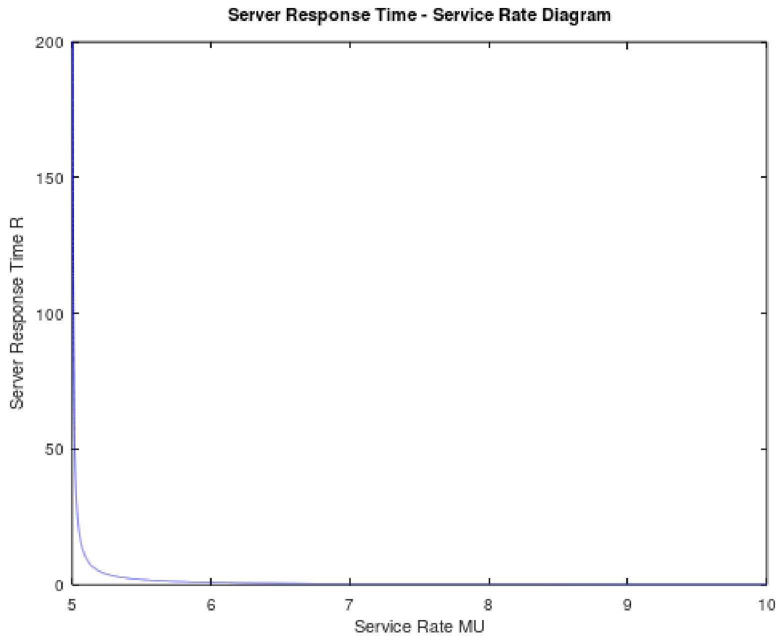
Ορίζουμε τη μεταβλητή mu που αναπαριστά τους διάφορους ρυθμούς εξυπηρέτησης. Οπότε, από το παραπάνω ερώτημα πρέπει 5< mu ≤10, δηλαδή ορίζουμε στον κώδικα: mu = 5.0001:0.0001:10;.

Παρακάτω, απεικονίζονται τα ζητούμενα διαγράμματα για τους επιτρεπτούς ρυθμούς εξυπηρέτησης:

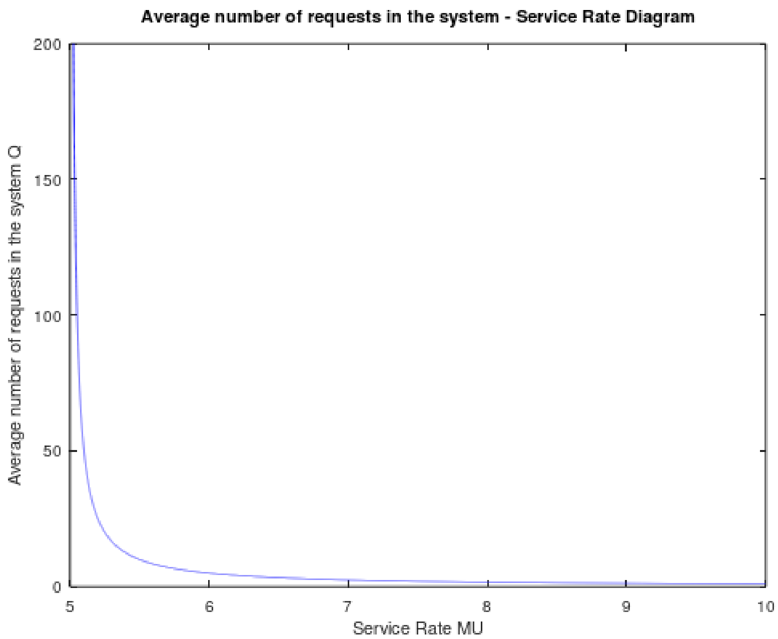
1ο Διάγραμμα: Βαθμός χρησιμοποίησης ρ ως προς το ρυθμό εξυπηρέτησης μ:



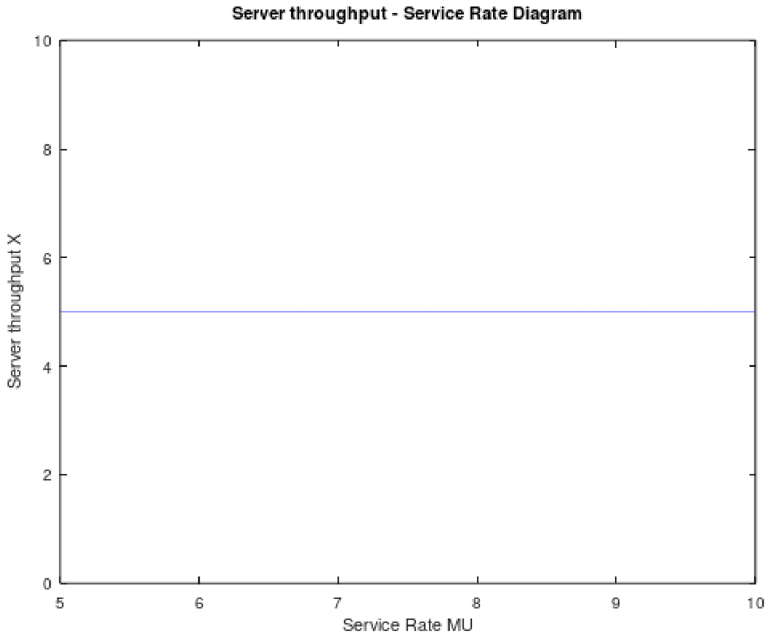
2º Διάγραμμα: Μέσος χρόνος καθυστέρησης του συστήματος E[T] ως προς το ρυθμό εξυπηρέτησης μ:



3º Διάγραμμα: Μέσος αριθμός πελατών στο σύστημα E[n(t)] ως προς ρυθμό εξυπηρέτησης μ:



4º Διάγραμμα: Ρυθμαπόδοση πελατών γ ως προς το ρυθμό εξυπηρέτησης μ:



Ο κώδικας που χρησιμοποιήθηκε για τις παραπάνω γραφικές, παρατίθεται παρακάτω:

```
# B Task

#(β)
lambda = 5 # clients/sec
mu = 5.0001:0.0001:10;
[U, R, Q, X] = qsmm1(lambda,mu);

# First Diagram
#subplot(4,1,1);
figure(1);
plot(mu,U, "b");
title("Server Utilization - Service Rate Diagram");
xlabel("Service Rate MU");        #Service rate MU or μ
ylabel("Server Utilization U");   #utilization rate U or ρ

# Second Diagram
#subplot(4,1,2)
figure(2);
plot(mu,R, "b");
axis([5 10 0 200]);
title("Server Response Time - Service Rate Diagram");
xlabel("Service Rate MU");        #Service rate MU or μ
ylabel("Server Response Time R"); #Server Response Time R or E[T]
```

```
# Third Diagram
#subplot(4,1,3)
figure(3);
plot(mu,Q, "b");
axis([5 10 0 200]);
title("Average number of requests in the system - Service Rate Diagram");
xlabel("Service Rate MU");        # Service rate MU or μ
ylabel("Average number of requests in the system Q");
                                  #Average number of requests
                                  #in the system Q or E[n(t)]
# Fourth Diagram
#subplot(4,1,4)
figure(4);
plot(mu,X, "b");
axis([5 10 0 10]);
title("Server throughput - Service Rate Diagram");
xlabel("Service Rate MU");        #Service rate MU or μ
ylabel("Server throughput X"); #Server throughput X or γ
# the system is ergodic, so we always have X = lambda
```

(γ)

Από το δεύτερο διάγραμμα του μέσου χρόνου καθυστέρησης του συστήματος E[T] ως προς το ρυθμό εξυπηρέτησης μ, παρατηρούμε ότι ο μέσος χρόνος καθυστέρησης του συστήματος E[T] με το ρυθμό εξυπηρέτησης μ είναι ποσά αντιστρόφως ανάλογα. Το γεγονός αυτό διακρίνεται από τη υπερβολική μορφή της γραφικής παράστασης, η οποία φανερώνει ότι αυξάνοντας το ρυθμό εξυπηρέτησης μ, τότε θα μειωθεί δραματικά ο χρόνος καθυστέρησης του συστήματος E[T] και αντίστροφα.

Αν επιλέξουμε τη μέγιστη τιμή του ρυθμού εξυπηρέτησης μ (από περιορισμούς πρέπει μ=10), τότε θα ήταν αρκετά δαπανηρή η επένδυση σε αυτό το σύστημα. Οπότε, στην περίπτωση που δεν διαθέτουμε μεγάλο κεφάλαιο για εξοπλισμό, η καταλληλότερη επιλογή ρυθμού εξυπηρέτησης μ είναι εκείνη που συνδυάζει το κόστος και τον μικρότερο δυνατό μέσο χρόνο καθυστέρησης, δηλαδή μία προτεινόμενη επιλογή θα ήταν μ = 7.

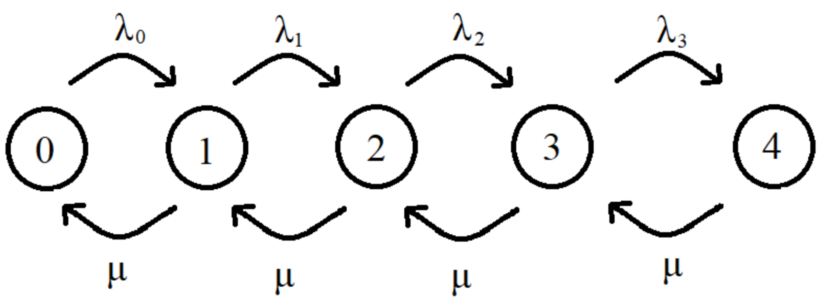
(δ)

Εφόσον το σύστημα είναι εργοδικό, τότε η ρυθμαπόδοση πελατών θα ταυτίζεται πάντα με τη τιμή λ, ανεξαρτήτως τιμής του ρυθμού εξυπηρέτησης μ, κάτι το οποίο επιβεβαιώνεται με την παραπάνω μορφή της γραφικής, η οποία είναι μία οριζόντια γραμμή που διέρχεται από την τιμή 5 του άξονα y'y. Το γεγονός αυτό επιβεβαιώνεται με τον ορισμό της ρυθμαπόδοσης. Ισχύει ότι η ρυθμαπόδοση είναι, σύμφωνα με τη σχέση, ίση με γ = $\lambda(1-P_{blocking})$, όπου Pblocking είναι η πιθανότητα απώλειας. Στην περίπτωσή μας, εφόσον το σύστημα είναι εργοδικό και το σύστημα M/M/1 έχει άπειρη χωρητικότητα χωρίς απόρριψη πελατών, τότε $P_{blocking} = 0$. Οπότε, η ρυθμαπόδοση γ = λ = 5 (πελάτες/ sec) ανεξαρτήτως ρυθμού εξυπηρέτησης μ.

<u>Διαδικασία γεννήσεων θανάτων (birth-death process): εφαρμογή σε σύστημα M/M/1/K</u>

(α)

Η μοντελοποίηση του συστήματος αναμονής M/M/1/4, μπορεί να περιγραφεί με το εξής διάγραμμα γεννήσεων-θανάτων:



Για την άσκηση, δίνεται ότι $\lambda_i = \frac{\lambda}{i+1}$ με λ = 5 πελάτες/sec και μ=10 πελάτες/sec, οπότε ισχύουν οι παρακάτω εξισώσεις ισορροπίας:

$$\lambda_i P_i = \mu P_{i+1}, i = 0,1,2,3 \ (*)$$

$$P_0 + P_1 + P_2 + P_3 + P_4 = 1 \ (**)$$

Από τη σχέση (**) και (*), ισχύει ότι:

$P_0 + P_0 \frac{\lambda_0}{\mu} + P_0 \frac{\lambda_1 \lambda_0}{\mu^2} + P_0 \frac{\lambda_2 \lambda_1 \lambda_0}{\mu^3} + P_0 \frac{\lambda_3 \lambda_2 \lambda_1 \lambda_0}{\mu^4} = 1$ ή $P_0 = 0.607$

Από τη σχέση (*):

$$P_1 = P_0 \frac{\lambda_0}{\mu} = 0.3035$$

$$P_2 = P_1 \frac{\lambda_1}{\mu} = 0.0759$$

$$P_3 = P_2 \frac{\lambda_2}{\mu} = 0.0126$$

$$P_4 = P_3 \frac{\lambda_3}{\mu} = 0.00158$$

Από τη θεωρία, η πιθανότητα της τελευταίας κατάστασης του συστήματος είναι ίση με την πιθανότητα απώλειας πελάτη $P_{blocking}$, καθώς το πλήρες σύστημα δεν μπορεί να δεχτεί επιπρόσθετους πελάτες και γι'αυτό τους απορρίπτει. Οπότε,

$$Pblocking = P_4 = P_3 \frac{\lambda_3}{\mu} = 0.00158$$

(β)

Με τη βοήθεια της εντολής ctmcbd του πακέτου queueing του Octave, μοντελοποιούμε το παραπάνω σύστημα ως μία διαδικασία γεννήσεων-θανάτων συνεχούς χρόνου.

i. Ο πίνακας που περιγράφει τους ρυθμούς ανάμεσα στις καταστάσεις του συστήματος, απεικονίζεται παρακάτω:

```
(i)
transition_rhythm_matrix =

   -5.00000    5.00000    0.00000    0.00000    0.00000
   10.00000  -12.50000    2.50000    0.00000    0.00000
    0.00000   10.00000  -11.66667    1.66667    0.00000
    0.00000    0.00000   10.00000  -11.25000    1.25000
    0.00000    0.00000    0.00000   10.00000  -10.00000
```

Παρατηρούμε ότι, το άθροισμα κάθε γραμμής, του πίνακα που περιγράφει τους ρυθμούς ανάμεσα στις καταστάσεις του συστήματος, είναι μηδέν. Επίσης, παρατηρούμε ότι τα διαγώνια στοιχεία του πίνακα δεν προσδίδουν κάποιο φυσικό νόημα. Τέλος, τα στοιχεία που βρίσκονται εκτός της διαγωνίου και έχουν μηδενική τιμή, προκύπτουν από το γεγονός ότι οι μεταβάσεις πραγματοποιούνται αποκλειστικά μεταξύ διαδοχικών καταστάσεων.

ii. Με την εντολή ctmc του πακέτου queueing του Octave, υπολογίζουμε τις εργοδικές πιθανότητες των καταστάσεων του συστήματος. Παρακάτω, απεικονίζεται το command window αφότου εκτελέσουμε το πρόγραμμα:

```
(ii)
Pr(i) =
ans =  0.60664
ans =  0.30332
ans =  0.075829
ans =  0.012638
ans =  0.0015798
```

Από τα παραπάνω, επιβεβαιώνεται ότι οι θεωρητικές τιμές των εργοδικών πιθανοτήτων ταυτίζονται με τις τιμές που υπολογίστηκαν μέσω της εντολής ctmc.

iii. Ο μέσος αριθμός πελατών στο σύστημα αναμονής M/M/1/4, όταν εκείνο βρίσκεται σε κατάσταση ισορροπίας υπολογίζεται, θεωρητικά, με τον τύπο: E[n(t)] = $\sum_{k=1}^{4} k\, P_k$. Παρακάτω, απεικονίζεται το command window αφότου εκτελέσουμε το πρόγραμμα:

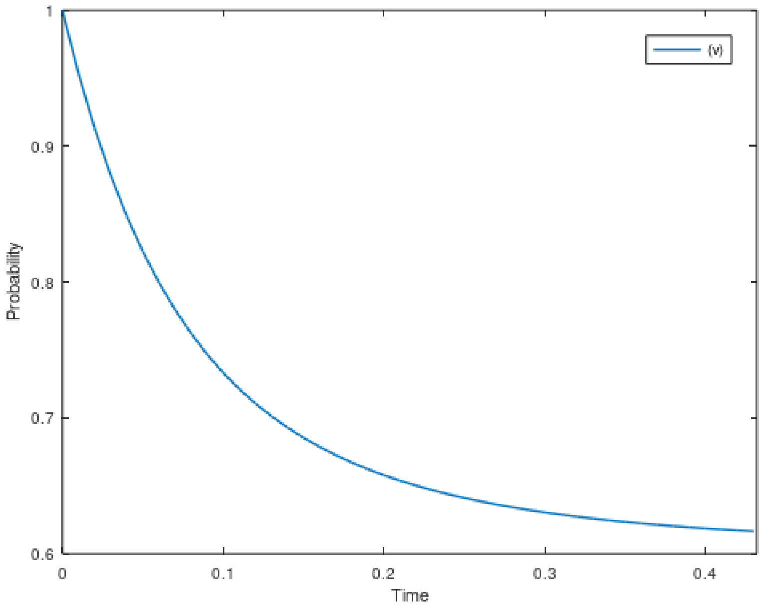
```
(iii)
Mean number of clients in system = mean_number_of_clients =  0.49921
```

iv. Η πιθανότητα της τελευταίας κατάστασης του συστήματος, από τη θεωρία, είναι ίση με την πιθανότητα απώλειας πελάτη Pblocking, καθώς το πλήρες σύστημα δεν μπορεί να δεχτεί επιπρόσθετους πελάτες και γι'αυτό τους απορρίπτει. Οπότε, $Pblocking\ = P_4$. Παρακάτω, απεικονίζεται το command window αφότου εκτελέσουμε το πρόγραμμα:

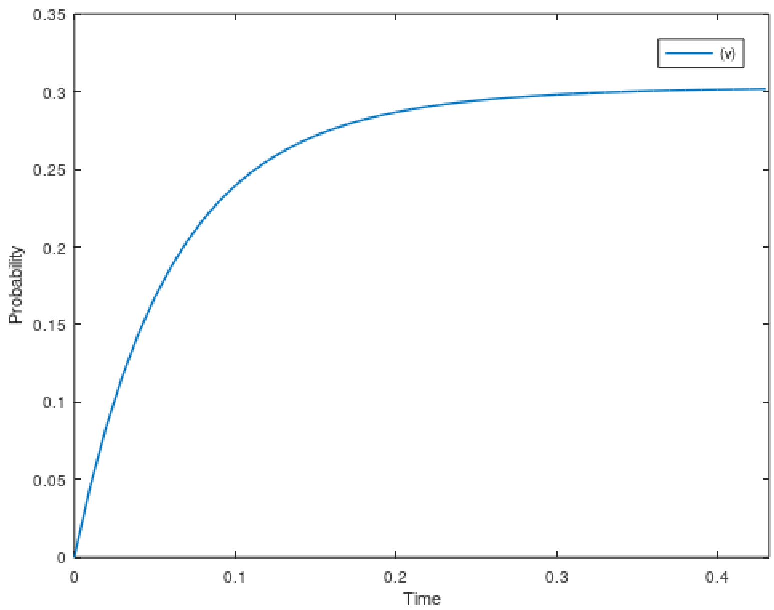
```
(iv)
Blocking Probability = ans =  0.0015798
```

v. Για κάθε μία από τις καταστάσεις του συστήματος, σχεδιάζουμε τα διαγράμματα των πιθανοτήτων των καταστάσεων του συστήματος σαν συναρτήσεις του χρόνου από την αρχική κατάσταση μέχρι οι πιθανότητες να έχουν απόσταση μικρότερη του 1% από τις εργοδικές πιθανότητες του ερωτήματος (2). Παρακάτω, απεικονίζονται με τη σειρά τα ζητούμενα διαγράμματα:

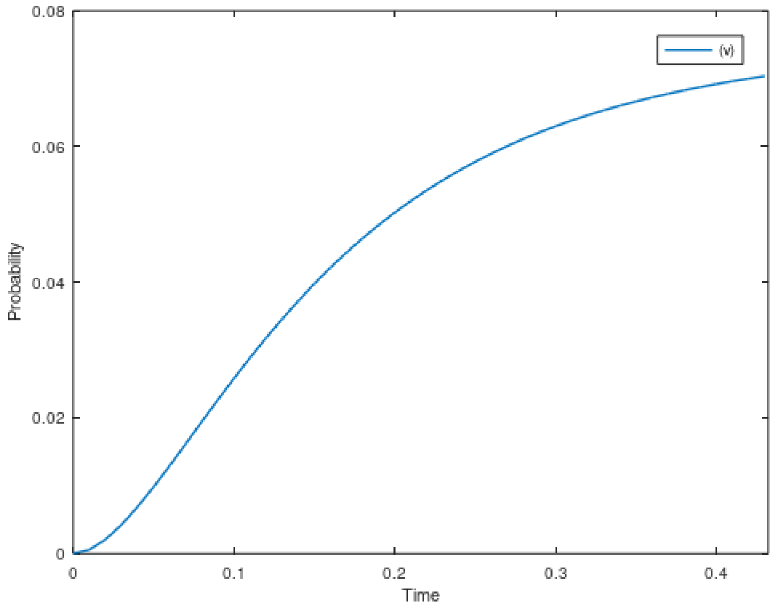
Παρακάτω, απεικονίζεται το διάγραμμα των πιθανοτήτων της μηδενικής κατάστασης του συστήματος σαν συνάρτηση του χρόνου:



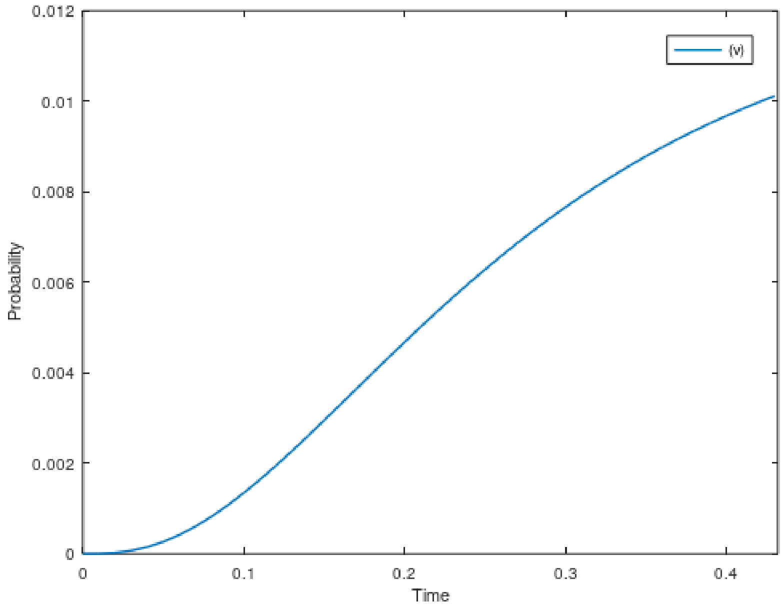
Παρακάτω, απεικονίζεται το διάγραμμα των πιθανοτήτων της πρώτης κατάστασης του συστήματος σαν συνάρτηση του χρόνου:



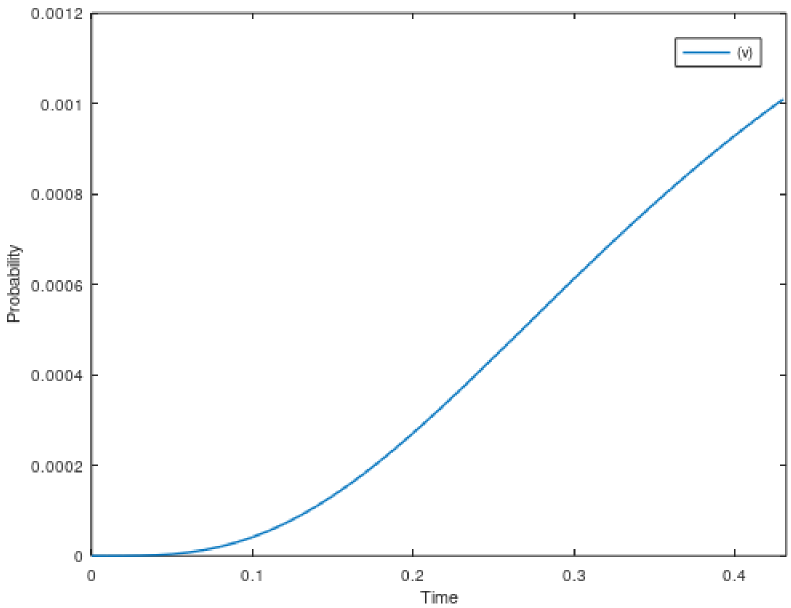
Παρακάτω, απεικονίζεται το διάγραμμα των πιθανοτήτων της δεύτερης κατάστασης του συστήματος σαν συνάρτηση του χρόνου:



Παρακάτω, απεικονίζεται το διάγραμμα των πιθανοτήτων της τρίτης κατάστασης του συστήματος σαν συνάρτηση του χρόνου:



Παρακάτω, απεικονίζεται το διάγραμμα των πιθανοτήτων της τέταρτης κατάστασης του συστήματος σαν συνάρτηση του χρόνου:



Ο κώδικας που χρησιμοποιήθηκε για τα παραπάνω ερωτήματα, παρατίθεται παρακάτω:

```
clc;
clear all;
close all;

#(i)
lambda = 5; # λ = 5
mu = 10;    # μ = 10
arrivals_lambda_i = [lambda/1, lambda/2, lambda/3, lambda/4];
departure_mu_i = [mu, mu, mu, mu];
printf("(i)\n");
transition_rhythm_matrix = ctmcbd(arrivals_lambda_i,departure_mu_i)

#(ii)
Pr = ctmc(transition_rhythm_matrix);#Compute stationary or transient
#state occupancy probabilities for a continuous-time Markov chain
printf("(ii)\n");
printf("Pr(i) =\n")
for i = 1:5
   Pr(i)
endfor
#(iii)
mean_number_of_clients = 0;

for i = 1:1:5
  mean_number_of_clients = mean_number_of_clients + Pr(i)*(i-1);
endfor
printf("(iii)\n");
printf("Mean number of clients in system = "); mean_number_of_clients
#(iv)
printf("(iv)\n");
printf("Blocking Probability = "); Pr(5)

#(v)
for i = 1:1:5
  cnt =0;
  for k = 0:0.01:50
    cnt = cnt + 1;
    Prob = ctmc(transition_rhythm_matrix, k, [1, 0, 0, 0, 0]);
    Probability(cnt) = Prob(i);
    if Prob - Pr < 0.01
      break;
    endif
  endfor
  t = 0:0.01:k;
  figure;
  plot(t, Probability, "linewidth",1.4);
  xlim([0 0.432]);
  xlabel("Time");
  ylabel("Probability");
  legend("(v)");
endfor
```