

Πολυτεχνείο Κρήτης
Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών



Διπλωματική Εργασία

Βουλγαρέλλης Παναγιώτης

**Γλώσσα Προσομοίωσης Για
Εξυπηρέτηση Αεροπλάνων Σε Κεντρικά
Αεροδρόμια**

Επιβλέποντες Καθηγητές :
Χριστοδούλου Εμμανουήλ
Σταυρακάκης Γεώργιος
Ζερβάκης Μιχάλης

Χανιά 2004

Θα ήθελα να εκφράσω τις ευχαριστίες μου σε όλους όσους βοήθησαν για την εκπόνηση και ολοκλήρωση αυτής της διπλωματικής τόσο σε επίπεδο πρακτικό όσο και σε ψυχολογικό.

Περιεχόμενα

Εισαγωγή	4
Κεφάλαιο 1.Βασικές έννοιες και ορολογία για προσομοίωση συστημάτων	6
1.1 Γενικά	6
1.2 Η έννοια του συστήματος	6
1.3 Μοντέλο Προσομοίωσης	9
1.4 Μηχανισμός Εξέλιξης του Χρόνου στην προσομοίωση διακριτών γεγονότων	11
1.4.1 Εξέλιξη με βάση το Χρόνο του Επομένου Γεγονότος.....	12
1.4.2 Εξέλιξη Σταθερής Αύξησης του Χρόνου.....	14
1.5 Τα Βήματα μιας Μελέτης με Προσομοίωση	15
1.6 Πλεονεκτήματα και Μειονεκτήματα της Προσομοίωσης	15
1.7 Προσομοίωση ενός Συστήματος Αναμονής - Ουρές Αναμονής	17
1.8 Κατανομή Poisson	21
1.9 Ανάπτυξη προγραμμάτων προσομοίωσης	22
Κεφάλαιο 2.Περιγραφή Συστήματος Αεροδρομίου Για Προσομοίωση.....	27
2.1 Γενική περιγραφή συστήματος αεροδρομίου	27
2.2 Παρουσίαση του υπό μελέτη αεροδρομίου	30
Κεφάλαιο 3.Ανάπτυξη γλώσσας προσομοίωσης για κίνηση αεροπλάνων σε χώρους αεροδρομίου και εφαρμογή για το αεροδρόμιο «Κ. Δασκαλογιάννης»	36
3.1 Γενικά	36
3.2 Κυρίως ρουτίνα airport	37
3.3 Υπορουτίνα priority	38
3.4 Υπορουτίνα landing	39
3.5 Υπορουτίνα taxiing.....	40
3.6 Υπορουτίνα service.....	41
3.7 Υπορουτίνα taxiing.....	42
3.8 Υπορουτίνα take_off.....	43
3.9 Περαιτέρω Ανάλυση.....	44
Κεφάλαιο 4.Εφαρμογή Προσομοίωσης - Παραδείγματα	52
4.1 Γενικά	52
4.2 Πρώτο Παράδειγμα – Ρυθμός Αφίξεων 10/100	53
4.3 Δεύτερο Παράδειγμα – Ρυθμός Αφίξεων 40/100	64
4.4 Τρίτο Παράδειγμα – Ρυθμός Αφίξεων 100/1440	71
4.5 Τέταρτο Παράδειγμα – Ρυθμός Αφίξεων 100/100	78
Κεφάλαιο 5.Συμπεράσματα – Παρατηρήσεις.....	82
Παράρτημα - Κώδικας της γλώσσας προσομοίωσης.....	86
Βιβλιογραφία	98

Εισαγωγή

Καθώς η ζήτηση για αερομεταφορές αυξάνεται, η συμφόρηση και οι καθυστερήσεις σε συστήματα εναέριας κυκλοφορίας γίνονται όλο και πιο συχνά. Έτσι η ανάγκη έρευνας για την διαχείριση ενός τέτοιου συστήματος είναι επιτακτική. Οι υπάρχουσες καθυστερήσεις δυσκολεύουν τις αεροπορικές εταιρίες και τις υπηρεσίες εναέριας κυκλοφορίας να διαχειριστούν τους επιβάτες, τις πτήσεις και τα πληρώματα των αεροπλάνων. Επιπρόσθετα, η αυξημένη συμφόρηση σε κομβικά αεροδρόμια οδηγεί σε σημαντικές οικονομικές και χωροταξικές ανεπάρκειες. Το να διαθέτουμε ακριβείς πληροφορίες, σχετικά με την τρέχουσα και της επόμενες θέσεις ενός αεροπλάνου το οποίο βρίσκεται μέσα στο σύστημα ενός αεροδρομίου, είναι πολύτιμο για τις αεροπορικές εταιρίες και τις υπηρεσίες εναέριας κυκλοφορίας.

Πολλές από τις καταγεγραμμένες καθυστερήσεις πτήσεων μπορούν άμεσα ή έμμεσα να αποδοθούν στα ίδια τα αεροδρόμια. Για τον λόγο αυτό γίνονται διάφορες προσπάθειες για βελτίωση της διαχείρισης των αεροδρομίων, και για πρόβλεψη πιθανών καταστάσεων που μπορεί να λάβουν χώρα σε αυτά.. Έχουν αναπτυχθεί λοιπόν λεπτομερή μοντέλα που περιγράφουν τις λειτουργίες ενός αεροδρομίου, κάποια από τα οποία είναι τα TAAM (Total Airspace & Airport Modeller), SIMMOD, The Airport Machine [1]. Τα μοντέλα αυτά είναι ‘μικροσκοπικά’ και απαιτούν την συλλογή πληροφοριών υψηλού επιπέδου για δεδομένα και για προδιαγραφές του αεροδρομίου το οποίο είναι υπό μελέτη. Αυτό έχει ως αποτέλεσμα να είναι ακατάλληλα για έρευνες οι οποίες απαιτούν προσεγγιστικές απαντήσεις καθώς εξετάζουν ένα ευρύ φάσμα σεναρίων και υποθέσεων σχετικά με μελλοντικές καταστάσεις σε ένα αεροδρόμιο.

Πολύ σημαντικές διαδικασίες στη λειτουργία ενός αεροδρομίου είναι οι αφίξεις και οι αναχωρήσεις αεροπλάνων και πολλά μοντέλα επικεντρώνονται στην πρόβλεψη τους. Έτσι πολλά συστήματα εναέριας κυκλοφορίας δεν λαμβάνουν υπόψη τους τις δραστηριότητες ενός αεροπλάνου σε επίπεδο εδάφους, θεωρώντας τον χρόνο παραμονής στο έδαφος σταθερό. Ο χρόνος αυτός περιλαμβάνει όλες τις διεργασίες ενός αεροπλάνου στον χώρο του αεροδρομίου από την προσγείωση ως την απογείωση του. Η παραδοχή αυτού του σταθερού χρόνου αγνοεί πιθανή

εμφάνιση ουρών αναμονής στους χώρους ενός αεροδρομίου, πράγμα που συνεπάγεται ότι οι αεροπορικές εταιρίες αδυνατούν να αντιμετωπίσουν τις καθυστερήσεις όσο χρόνο το αεροπλάνο βρίσκεται στο έδαφος. Ωστόσο στην πραγματικότητα οι αεροπορικές εταιρίες συχνά προσπαθούν με διάφορους τρόπους να περιορίσουν τον χρόνο παραμονής στο έδαφος για ένα αεροπλάνο με καθυστέρηση, σκοπεύοντας να ελέγξουν την αύξηση των καθυστερήσεων.

Τα παραπάνω μας δείχνουν την ανάγκη που υπάρχει για ανάπτυξη μοντέλων που να καθορίζουν επακριβώς το χρόνο των επιμέρους δραστηριοτήτων στο έδαφος ενός αεροπλάνου, καθώς και τους χρόνους προσγείωσης και απογείωσης του. Στην παρούσα διπλωματική αντιμετωπίζουμε το θέμα αυτό με την ανάπτυξη μιας γλώσσας προσομοίωσης βασισμένη σε MATLAB για εξυπηρέτηση αεροπλάνων σε κεντρικά αεροδρόμια .

Η εργασία αποτελείται από πέντε κεφάλαια. Στο πρώτο περιγράφονται σε γενικές γραμμές βασικές αρχές για προσομοιώσεις συστημάτων ,ο διαχωρισμός τους σε διάφορες ομάδες ανάλογα με τα χαρακτηριστικά του καθενός όπως και διάφοροι μηχανισμοί αντιμετώπισης τους. Στο δεύτερο περιγράφεται το σύστημα ενός αεροδρομίου και γίνονται ορισμένες παρατηρήσεις που έχουν να κάνουν με την προσομοίωσή του. Επίσης στο κεφάλαιο αυτό έχουμε και την διεξοδική περιγραφή των χαρακτηριστικών του αεροδρομίου Χανίων "Κ. Δασκαλογιάννης" που επιλέχθηκε για προσομοίωση. Στο τρίτο κεφάλαιο γίνεται η ανάπτυξη της γλώσσας προσομοίωσης με εφαρμογή της για το παραπάνω συγκεκριμένο αεροδρόμιο Το τέταρτο εμφανίζει τα αποτελέσματα της προσομοίωσης του αεροδρομίου Χανίων σε μορφή γραφικών παραστάσεων οι οποίες είναι είτε χρονοδιαγράμματα της κίνησης των αεροπλάνων μέσα στους χώρους του αεροδρομίου ή χρονοδιαγράμματα που δείχνουν πότε χρησιμοποιείτε ο κάθε χώρος του εν λόγω αεροδρομίου. Παρουσιάζονται τέσσερα διαφορετικά παραδείγματα προσομοίωσης. Τέλος στο πέμπτο κεφάλαιο παρουσιάζονται συμπεράσματα και παρατηρήσεις.

Κεφάλαιο 1

Βασικές έννοιες και ορολογία για προσομοίωση συστημάτων

1.1 Γενικά

Η έννοια της μοντελοποίησης, στην οποία βασίζεται η προσομοίωση, χρησιμοποιείται εδώ και πολλά χρόνια . Τα τελευταία όμως χρόνια παράλληλα με την ανάπτυξη των υπολογιστών είχαμε και την χρήση της προσομοίωσης σε πάρα πολλούς τομείς .Σε γενικές γραμμές προσομοίωση είναι η διαδικασία σχεδιασμού του μοντέλου ενός συστήματος και η διεξαγωγή πειραμάτων με το μοντέλο αυτό που σκοπό έχουν είτε την κατανόηση της συμπεριφοράς του συστήματος ή την δοκιμή διάφορων στρατηγικών για την λειτουργία του συστήματος. Προσφέρει στον αναλυτή, χειριστή, σχεδιαστή, ή μελετητή τη δυνατότητα να ελέγχει ιδέες και συλλήψεις σχεδιασμού ή να αλληλεπιδρά με ολόκληρα συστήματα όσον αφορά τη λειτουργία, την απόδοση και τα χαρακτηριστικά τους χωρίς να χρειαστεί να επέμβει στο πραγματικό σύστημα . Παρακάτω γίνεται αναφορά σε βασικές έννοιες και ορολογία για την προσομοίωση συστημάτων βάση και της βιβλιογραφίας [2], [3], [4], [5] και [6].

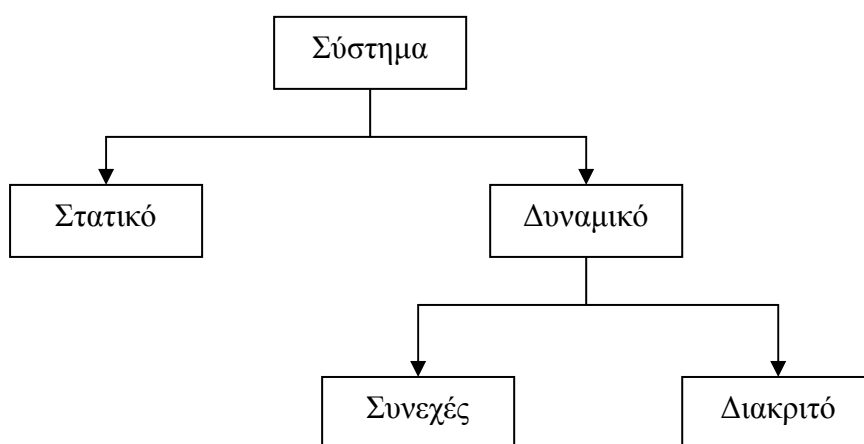
1.2 Η έννοια του συστήματος

Κεντρική ιδέα για κάθε μελέτη προσομοίωσης είναι αυτή του συστήματος. Ένα σύστημα ορίζεται ως μία συλλογή οντοτήτων (π.χ. αεροπλάνα και οι χώροι ενός αεροδρομίου στο οποίο κινούνται) που ενεργούν και αλληλεπιδρούν, με στόχο κάποιο λογικό τερματισμό. Στην πράξη, η έννοια του συστήματος εξαρτάται κάθε φορά από τους στόχους της συγκεκριμένης μελέτης. Η συλλογή οντοτήτων που συγκροτούν ένα σύστημα για μία μελέτη, μπορεί να είναι μόνο ένα υποσύνολο του συνολικού συστήματος για μία άλλη μελέτη.

Ορίζουμε την κατάσταση ενός συστήματος ως τη συλλογή των μεταβλητών που είναι απαραίτητες για την περιγραφή του συστήματος σε μια χρονική στιγμή, αναφορικά με τις απαιτήσεις της μελέτης.

Τα συστήματα μπορούν να ταξινομηθούν σε στατικά και δυναμικά . Στα στατικά δεν έχουμε αλλαγή στην κατάστασή τους με τον χρόνο ενώ στα δυναμικά

ισχύει το αντίθετο. Αν ένα σύστημα είναι δυναμικό μπορεί να ταξινομηθεί ανάλογα με το πώς οι μεταβλητές οι οποίες το περιγράφουν αλλάζουν στην διάρκεια του χρόνου. Έτσι μπορεί να είναι διακριτό ή συνεχές. Διακριτό σύστημα είναι αυτό στο οποίο οι μεταβλητές κατάστασης αλλάζουν σε διακεκριμένες στιγμές του χρόνου, ενώ συνεχές είναι το σύστημα του οποίου οι μεταβλητές κατάστασης αλλάζουν συνεχώς στο χρόνο. Λίγα συστήματα στην πράξη είναι εξ' ολοκλήρου διακριτά ή συνεχή, αλλά επειδή συνήθως μία από τις δύο ιδιότητες κυριαρχεί, μπορούμε τις περισσότερες φορές να χαρακτηρίσουμε το σύστημα ως διακριτό ή συνεχές. Οι διαχωρισμοί των συστημάτων φαίνονται στο Σχήμα 1.1:



Σχήμα 1.1 Διαχωρισμοί συστημάτων

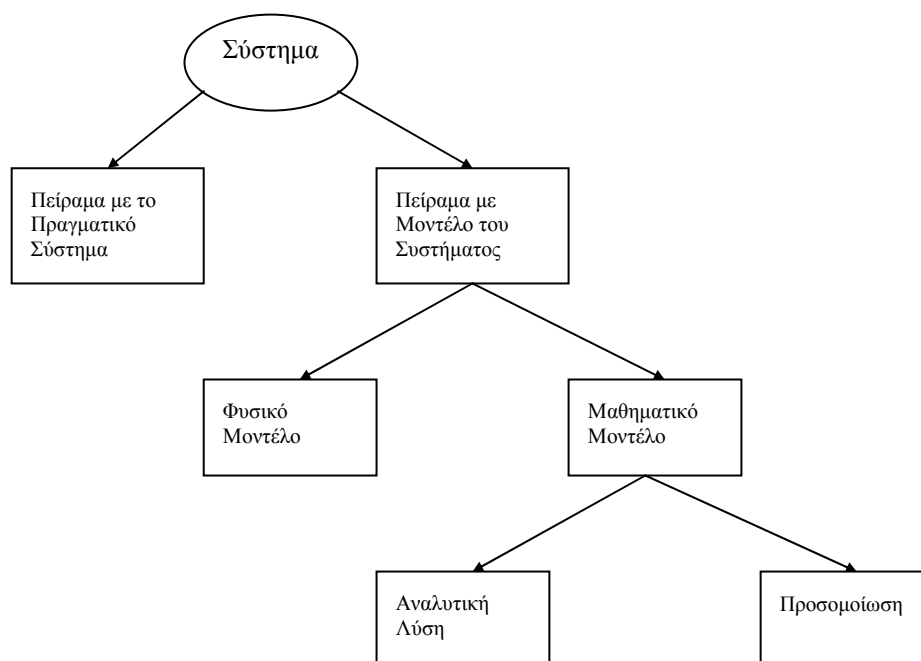
Κατά τη διάρκεια της ζωής ενός συστήματος, παρουσιάζεται η ανάγκη μελέτης του, ώστε να αποκτήσουμε γνώση για τις σχέσεις μεταξύ των διαφόρων τμημάτων του, ή για να προβλέψουμε την συμπεριφορά του κάτω από νέες, πιθανές συνθήκες. Παρακάτω αναφέρονται οι διαφορετικοί τρόποι με τους οποίους μπορεί να μελετηθεί ένα σύστημα οι οποίοι παρουσιάζονται και στο Σχήμα 1.2.

- Πείραμα με το Πραγματικό Σύστημα ή Πείραμα με Μοντέλο του Συστήματος: Αν είναι δυνατόν να έχουμε φυσική πρόσβαση στο σύστημα και να το λειτουργήσουμε κάτω από τις νέες συνθήκες, τότε δεν υπάρχει αμφισβήτηση για την ακρίβεια της μελέτης. Πάντως, σπάνια μας δίνεται αυτή η ευκαιρία, είτε γιατί κοστίζει, ή γιατί είναι πολύ πιθανό να προκαλέσει προβλήματα στην κανονική λειτουργία του συστήματος. Όταν ένα σύστημα

είναι στη φάση σχεδιασμού του, δηλαδή το σύστημα δεν υπάρχει, αν θέλουμε να μελετήσουμε τις διάφορες εναλλακτικές προτάσεις υλοποίησής του είναι αναγκαίο να κατασκευάσουμε ένα μοντέλο, το οποίο θα αναπαριστά το σύστημα και θα το αντικαθιστά κατά τη μελέτη. Όταν χρησιμοποιούμε ένα μοντέλο, η εγκυρότητά του, δηλαδή ο βαθμός ακρίβειας με τον οποίο αναπαριστά το σύστημα, είναι πάντα ένα κρίσιμο ερώτημα .

- Φυσικό Μοντέλο ή Μαθηματικό Μοντέλο: Το φυσικό μοντέλο είναι η φυσική αναπαράσταση του αντικειμένου που αντιπροσωπεύει. Για παράδειγμα φυσικά μοντέλα αποτελούν οι μακέτες. Όμως, η μεγάλη πλειοψηφία των μοντέλων είναι μαθηματικά. Ένα μαθηματικό μοντέλο αναπαριστά το σύστημα με βάση λογικές και ποσοτικές σχέσεις οι οποίες χειριζόμενες και μεταβαλλόμενες κατάλληλα, επιτρέπουν να δούμε πώς αντιδρά το μοντέλο και κατά συνέπεια πώς θα αντιδρούσε το σύστημα, αν το μαθηματικό μοντέλο είναι έγκυρο.
- Αναλυτική Λύση ή Προσομοίωση: Από τη στιγμή που έχει δημιουργηθεί ένα μαθηματικό μοντέλο, θα πρέπει να εξετασθεί ο τρόπος χρήσης του ώστε να μπορεί να απαντήσει στα ερωτήματα που μας ενδιαφέρουν για το σύστημα που αντιπροσωπεύει. Αν το μοντέλο είναι αρκετά απλό, είναι δυνατό να πάρουμε μία ακριβή, αναλυτική λύση μέσω ενός συνόλου εξισώσεων οι οποίες περιγράφουν το σύστημα. Συνήθως αυτό συμβαίνει μόνο με τα στατικά συστήματα. Στις περισσότερες περιπτώσεις η αναλυτική λύση μπορεί να είναι αρκετά πολύπλοκη και να απαιτεί μεγάλη υπολογιστική ισχύ. Πάντως, αν υπάρχει αναλυτική λύση στο μαθηματικό μοντέλο και είναι υπολογιστικά αποδοτική, συνήθως την προτιμούμε από την προσομοίωση. Στην περίπτωση που το εξεταζόμενο σύστημα είναι πολύπλοκο, και είναι πολύ δύσκολο ως αδύνατο να εξευρεθούν μαθηματικές εξισώσεις που να το περιγράφουν, κάθε πιθανότητα αναλυτικής λύσης αποκλείεται. Τότε το μοντέλο πρέπει να μελετηθεί με τη χρήση προσομοίωσης, δηλαδή με την εκτέλεση αριθμητικών πειραμάτων στο μοντέλο για τις εισόδους που μας ενδιαφέρουν, για να δούμε πως αυτά επηρεάζουν τις εξόδους του συστήματος. Η προσομοίωση είναι επίσης πολύ χρήσιμη στις περιπτώσεις που μπορούμε να πάρουμε μόνο προσεγγιστική αναλυτική λύση, οπότε θέλουμε να επιβεβαιώσουμε την

προσέγγιση, να βρούμε το σχετικό λάθος κ.λ.π. Συνήθως κάνουμε χρήση της προσομοίωσης όταν ένα σύστημα είναι δυναμικό.



Σχήμα 1.2 Τρόποι μελέτης ενός συστήματος

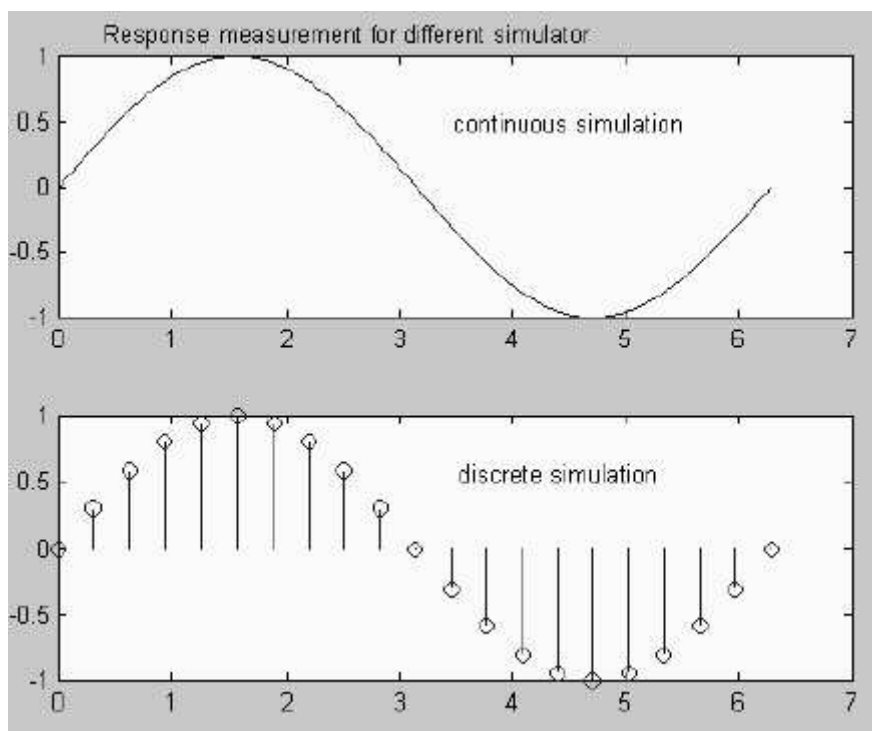
1.3 Μοντέλο Προσομοίωσης

Έχοντας ένα μαθηματικό μοντέλο που πρέπει να μελετήσουμε με προσομοίωση, θα πρέπει να αναζητήσουμε κατάλληλα εργαλεία για το σκοπό αυτό. Στην προσπάθεια αυτή, είναι χρήσιμο να ταξινομήσουμε τα μοντέλα προσομοίωσης με βάση τέσσερις διαφορετικές έννοιες:

1. Στατικά ή Δυναμικά Μοντέλα Προσομοίωσης: Ένα στατικό μοντέλο προσομοίωσης, αναπαριστά ένα σύστημα σε μία συγκεκριμένη χρονική στιγμή, ή αναπαριστά ένα σύστημα στο οποίο ο χρόνος δεν έχει σημασία. Αντίθετα, ένα δυναμικό μοντέλο προσομοίωσης αναπαριστά ένα σύστημα, όπως αυτό εξελίσσεται με την πάροδο του χρόνου.
2. Ντετερμινιστικά ή Στοχαστικά Μοντέλα Προσομοίωσης: Αν ένα μοντέλο προσομοίωσης δεν περιλαμβάνει "τυχαία" τμήματα, ονομάζεται ντετερμινιστικό. Στα ντετερμινιστικά μοντέλα, η έξοδος είναι καθορισμένη, με δεδομένο το σύνολο των ποσοτήτων και σχέσεων εισόδου του μοντέλου. Όμως, πολλά συστήματα

χρησιμοποιούν στοχαστικά μοντέλα προσομοίωσης, δηλαδή μοντέλα που θα έχουν τουλάχιστον ορισμένα τμήματα με "τυχαία" είσοδο. Τα περισσότερα υπολογιστικά συστήματα, που βασίζονται στα συστήματα αναμονής (queueing systems), χρησιμοποιούν στοχαστικά μοντέλα προσομοίωσης.

3. Αυτο-οδηγούμενα ή Ιχνο-οδηγούμενα Μοντέλα Προσομοίωσης: Σε ένα αυτο-οδηγούμενο μοντέλο, υπάρχει μία εσωτερική πηγή τυχαίων αριθμών. Οι τυχαίοι αριθμοί οδηγούν τα τμήματα του μοντέλου, δηλαδή χρησιμοποιούνται για τον προσδιορισμό των στιγμών εμφανίσεων των γεγονότων του συστήματος. Το βασικό χαρακτηριστικό του αυτο-οδηγούμενου μοντέλου είναι ότι αποτελεί ένα αυτόνομο μοντέλο το οποίο δεν χρειάζεται εξωτερικές εισόδους (inputs) για να λειτουργήσει. Αντίθετα, ένα ιχνο-οδηγούμενο (trace-driven) μοντέλο καθοδηγείται από ακολουθίες εισόδου που προέρχονται από δεδομένα που έχουν δημιουργηθεί από τη λειτουργία του πραγματικού συστήματος το οποίο προσομοιώνουμε. Τα ιχνο-οδηγούμενα μοντέλα έχουν ορισμένα πλεονεκτήματα, όπως το γεγονός ότι αποφεύγονται οι δυσκολίες της πιθανοτικής ανάλυσης που χρειάζεται για τη χρήση κατανομών στην περιγραφή των εισόδων του μοντέλου και επίσης το γεγονός ότι τα μοντέλα αυτά είναι εύκολο να επιβεβαιωθούν. Το πρόβλημα με τα ιχνο-οδηγούμενα μοντέλα είναι το μικρό εύρος εφαρμογών που μπορούν να αντιμετωπίσουν. Οι εφαρμογές αυτές πρακτικά περιορίζονται σε υπολογιστικά συστήματα και μάλιστα μόνο για τη μελέτη μετατροπών σε ένα σύστημα που ήδη λειτουργεί.
4. Συνεχή ή Διακριτά Μοντέλα Προσομοίωσης: Οι ορισμοί των συνεχών και διακριτών μοντέλων προσομοίωσης, είναι ανάλογοι με τους ορισμούς των συνεχών και διακριτών συστημάτων που αναφέρθηκαν στην προηγούμενη παράγραφο. Πάντως, πρέπει να σημειωθεί ότι ένα διακριτό μοντέλο δεν χρησιμοποιείται μόνο για την αναπαράσταση ενός διακριτού συστήματος και ένα διακριτό σύστημα δεν αναπαριστάται μόνο από ένα διακριτό μοντέλο προσομοίωσης. Η απόφαση για τη χρήση ενός διακριτού ή ενός συνεχούς μοντέλου για ένα συγκεκριμένο σύστημα, εξαρτάται από τους ιδιαίτερους στόχους της μελέτης. Στο Σχήμα 1.3 έχουμε ένα παράδειγμα συνεχούς και διακριτού μοντέλου προσομοίωσης.



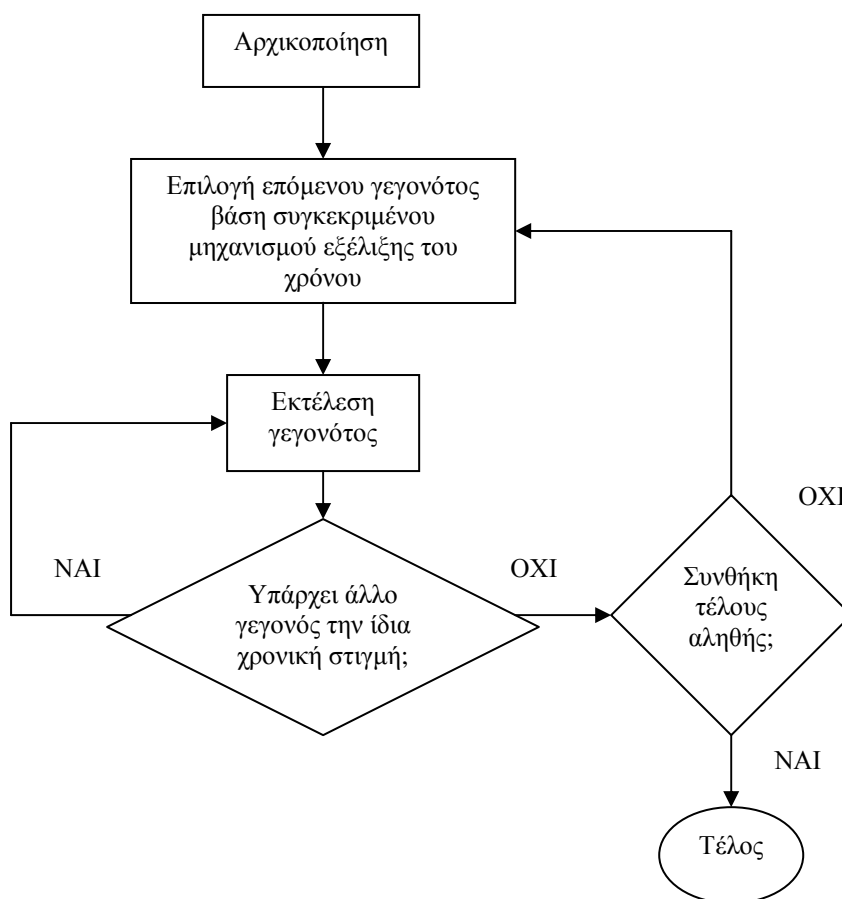
Σχήμα 1.3 Διακριτή και συνεχής προσομοίωση

Το μοντέλο προσομοίωσης που θα μας απασχολήσει στη συνέχεια, θα είναι διακριτό, δυναμικό, στοχαστικό και αυτο-οδηγούμενο δηλαδή είναι ένα μοντέλο προσομοίωσης διακριτών γεγονότων (discrete event simulation model). Μάλιστα, αφού τα ντετερμινιστικά μοντέλα μπορούν να θεωρηθούν ειδικές περιπτώσεις των στοχαστικών μοντέλων, δεν θα έχουμε απώλεια της γενικότητας στη μελέτη των μοντέλων προσομοίωσης.

1.4 Μηχανισμός Εξέλιξης του Χρόνου στην προσομοίωση διακριτών γεγονότων

Λόγω του δυναμικού χαρακτήρα των μοντέλων προσομοίωσης διακριτών γεγονότων, πρέπει να έχουμε τη δυνατότητα αποθήκευσης της τρέχουσας τιμής του προσομοιωμένου χρόνου, ενώ χρειαζόμαστε και ένα μηχανισμό αύξησής του από μία τιμή σε μία άλλη. Η μεταβλητή του μοντέλου προσομοίωσης που μας δίνει την τρέχουσα τιμή του χρόνου, ονομάζεται ρολόι προσομοίωσης (simulation clock). Η μονάδα χρόνου που χρησιμοποιεί το ρολόι είναι συνήθως η ίδια με αυτή που χρησιμοποιούν οι παράμετροι εισόδου, ενώ γενικά δεν υπάρχει σχέση του χρόνου που καταγράφει το ρολόι, με το χρόνο που απαιτείται για την εκτέλεση του προσομοιωτή

στον υπολογιστή. Κατά την προσομοίωση λοιπόν είναι αναγκαία η χρήση κάποιου μηχανισμού καταγραφής του χρόνου για να μπορούμε να ελέγχουμε τις αλλαγές κατάστασης στο σύστημα που μελετάμε. Αυτό το γεγονός φαίνεται και στο Σχήμα 1.4.



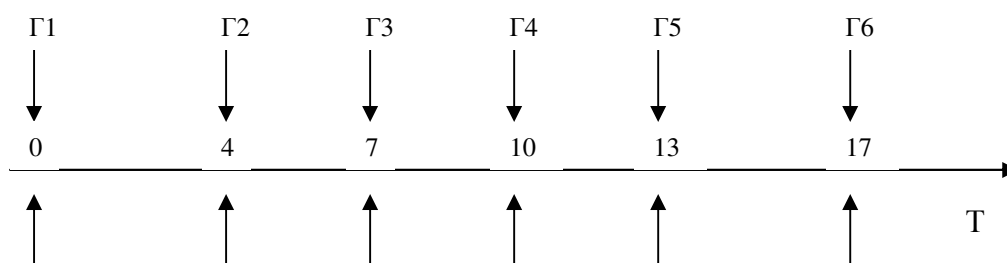
Σχήμα 1.4 Σημασία μηχανισμού εξέλιξης του χρόνου

Έχουν επικρατήσει δύο βασικές μέθοδοι για την εξέλιξη του ρολογιού προσομοίωσης οι οποίοι αναφέρονται παρακάτω.

1.4.1 Εξέλιξη με βάση το Χρόνο του Επομένου Γεγονότος

Η Εξέλιξη με βάση το Χρόνο του Επομένου Γεγονότος ή Ασύγχρονη Προσομοίωση (next-event time advance, asynchronous) είναι η μέθοδος κατά την οποία το ρολόι προσομοίωσης αρχικοποιείται στο μηδέν και καθορίζονται εκ των προτέρων οι στιγμές εμφάνισης των μελλοντικών γεγονότων. Το ρολόι τότε αυξάνει στο χρόνο εμφάνισης του πιο κοντινού στο μέλλον, από τα γεγονότα αυτά. Τη στιγμή αυτή η κατάσταση του συστήματος ενημερώνεται ώστε να πάρει υπόψη της το

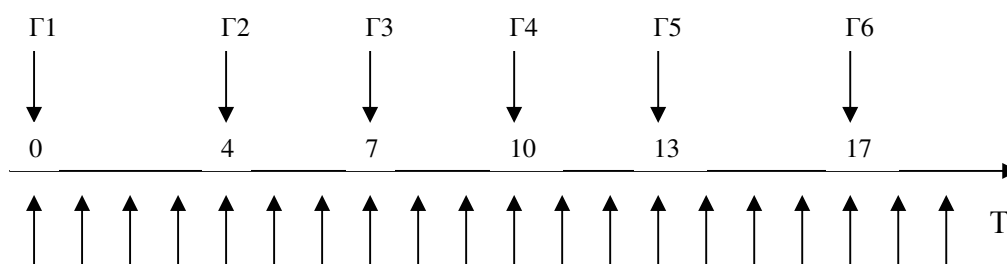
γεγονός που εμφανίστηκε, ενώ ενημερώνεται επίσης η γνώση μας για τις χρονικές στιγμές εμφάνισης των μελλοντικών γεγονότων. Στη συνέχεια, το ρολόι αυξάνει ώστε να δείχνει τη στιγμή εμφάνισης του νέου πιο κοντινού στο μέλλον γεγονότος, η κατάσταση του συστήματος ενημερώνεται, καθορίζονται οι χρονικές στιγμές εμφάνισης των μελλοντικών γεγονότων κ.ο.κ . Η διαδικασία αυτή εξέλιξης του ρολογιού προσομοίωσης από το ένα γεγονός στο άλλο, συνεχίζεται μέχρι να ικανοποιηθεί κάποια προκαθορισμένη συνθήκη τερματισμού της προσομοίωσης. Αφού όλες οι αλλαγές κατάστασης γίνονται μόνο στις χρονικές στιγμές εμφάνισης των γεγονότων, οι ενδιάμεσες ανενεργοί περίοδοι δεν λαμβάνονται υπόψη και το ρολόι μετακινείται αυτόματα στη στιγμή εμφάνισης του επομένου γεγονότος. Στο Σχήμα 1.5 έχουμε εμφάνιση γεγονότων Γ1 ,Γ2, Γ3, Γ4 τις χρονικές στιγμές 0, 4, 7, 10, 13 και 17 αντίστοιχα . Σύμφωνα με την εξέλιξη του χρόνου με βάση το επόμενο γεγονός το ρολόι θα σταματήσει μόνο στις στιγμές εμφάνισης των γεγονότων. Τα βέλη πάνω από τον άξονα του χρόνου T δηλώνουν τις χρονικές στιγμές που συμβαίνουν τα γεγονότα ενώ τα βέλη κάτω από τον άξονα του χρόνου δηλώνουν τις χρονικές στιγμές που το ρολόι σταματά για έλεγχο του συστήματος .



Σχήμα 1.5 Εξέλιξη με βάση το Χρόνο του Επομένου Γεγονότος

1.4.2 Εξέλιξη Σταθερής Αύξησης του Χρόνου

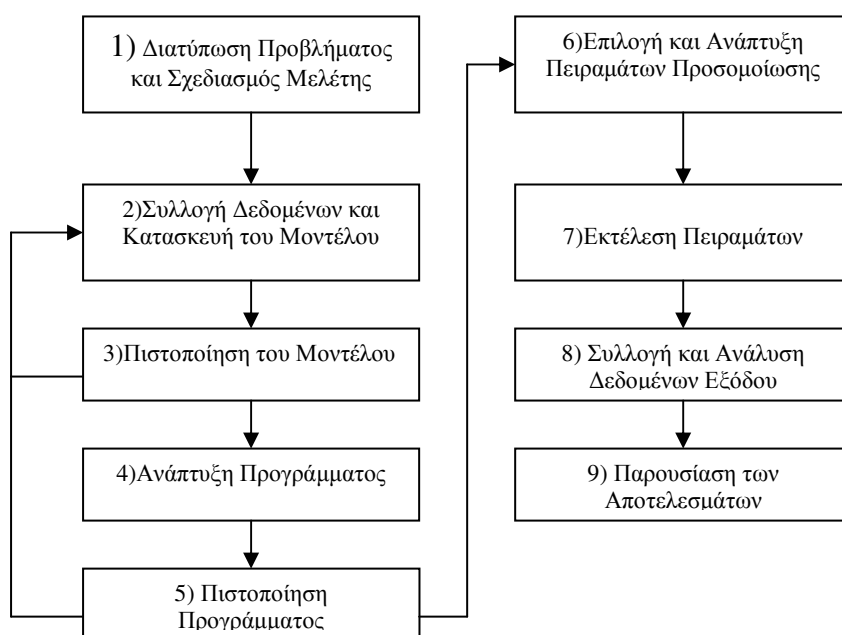
Η Εξέλιξη Σταθερής Αύξησης του Χρόνου ή Σύγχρονη Προσομοίωση (fixed-increment time advance, synchronous) είναι η μέθοδος κατά την οποία το ρολόι προσομοίωσης εξελίσσεται με σταθερές αυξήσεις ακριβώς Δt μονάδων χρόνου κάθε φορά. Μετά από κάθε ενημέρωση του ρολογιού, γίνεται ένας έλεγχος για να εξακριβωθεί εάν θα έπρεπε να έχουν εμφανισθεί κάποια γεγονότα κατά το προηγούμενο χρονικό διάστημα Δt . Αν εμφανίσθηκαν γεγονότα στο διάστημα αυτό, θεωρούμε ότι αυτά εμφανίζονται στο τέλος του χρονικού διαστήματος και η κατάσταση του συστήματος ενημερώνεται κατάλληλα. Η τιμή του Δt θα πρέπει να είναι μικρή για να έχουμε μεγαλύτερη ακρίβεια. Όσο πιο μικρό όμως είναι το Δt τόσο πιο αργή γίνεται η προσομοίωση αφού ελέγχουμε περισσότερα διαστήματα διάρκειας Δt κατά τα οποία πιθανότατα να μην έχουμε κάποια αλλαγή στο σύστημα. Χρήσιμη σε μοντελοποίηση συστημάτων στα οποία έχουμε εμφάνιση γεγονότων κατά διαστήματα σταθερού μεγέθους. Στο Σχήμα 1.6 έχουμε εμφάνιση γεγονότων Γ1, Γ2, Γ3, Γ4 τις χρονικές στιγμές 0, 4, 7, 10, 13 και 17 αντίστοιχα . Σύμφωνα με την εξέλιξη σταθερής αύξησης του χρόνου το ρολόι της προσομοίωσης σταματά σε όλες τις χρονικές στιγμές που καθορίζονται από το μέγεθος του σταθερού διαστήματος (στην περίπτωση αυτή $\Delta t=1$) ανεξάρτητα από το αν συμβαίνει κάποιο γεγονός ή όχι. Τα βέλη πάνω από τον άξονα του χρόνου T δηλώνουν τις χρονικές στιγμές που συμβαίνουν τα γεγονότα ενώ τα βέλη κάτω από τον άξονα του χρόνου δηλώνουν τις χρονικές στιγμές που το ρολόι σταματά για έλεγχο του συστήματος.



Σχήμα 1.6 Εξέλιξη Σταθερής Αύξησης του Χρόνου

1.5 Τα Βήματα μιας Μελέτης με Προσομοίωση

Στο Σχήμα 1.7 φαίνονται τα βήματα που ακολουθεί μία τυπική μελέτη ενός συστήματος με χρήση προσομοίωσης. Τα βήματα 1 έως και 6 αφορούν την κατανόηση και δημιουργία του μοντέλου σε θεωρητικό και μετά σε προγραμματιστικό επίπεδο. Αν αυτό είναι έγκυρο προχωράμε στα βήματα 6 και 7 για την διεξαγωγή των πειραμάτων που μας ενδιαφέρουν. Τέλος τα αποτελέσματα της προσομοίωσης αναλύονται κατά τα βήματα 8 και 9.



Σχήμα 1.7 Βήματα προσομοίωσης

1.6 Πλεονεκτήματα και Μειονεκτήματα της Προσομοίωσης

Τα τελευταία χρόνια η προσομοίωση εφαρμόζεται σχεδόν σε κάθε πεδίο των ανθρώπινων δραστηριοτήτων και το ερώτημα αν όντως είναι τόσο αποτελεσματική για την χρήση που τυγχάνει είναι εύλογο. Είναι γεγονός ότι σε πολλές περιπτώσεις η προσομοίωση δεν είναι εφαρμόσιμη ή ακόμα και αν είναι μπορεί να υπάρχουν άλλοι πιο γρήγοροι και φθηνοί τρόποι επίλυσης του προβλήματος. Μπορεί η προσομοίωση να είναι ένα από τα πιο εύχρηστα εργαλεία για χρήση στην επιστήμη της διαχείρισης συστημάτων αλλά πιθανόν και από τα πιο δύσκολα ως προς την σωστή εφαρμογή τους και ως προς την απεικόνιση εύστοχων συμπερασμάτων. Γενικά θα πρέπει να

αναγνωρίσουμε ορισμένα πλεονεκτήματα και μειονεκτήματα για την όλη διαδικασία μιας προσομοίωσης .

Ορισμένα πλεονεκτήματα είναι τα εξής:

1. Η προσομοίωση επιτρέπει να πραγματοποιούνται ελεγχόμενοι πειραματισμοί πάνω σε ένα σύστημα. Ένα πείραμα προσομοίωσης μπορεί να εκτελεστεί πολλές φορές αλλά με διαφορετικές τιμές εισόδου για να εξεταστεί η συμπεριφορά του υπό εξέταση συστήματος κάτω από διαφορετικές κάθε φορά καταστάσεις και συνθήκες.
2. Ένα άλλο πλεονέκτημα είναι ότι έχουμε συμπίεση του χρόνου. Αυτό σημαίνει ότι η λειτουργία ενός συστήματος η οποία διαρκεί για κάποιο εκτεταμένο χρονικό διάστημα μπορεί να προσομοιωθεί μέσα σε λίγα λεπτά με την χρήση ενός σύγχρονου υπολογιστή.
3. Τα περισσότερα σύνθετα συστήματα του πραγματικού κόσμου με στοχαστικές παραμέτρους, δεν μπορούν να περιγραφούν ικανοποιητικά με κάποιο μαθηματικό μοντέλο που μπορεί να λυθεί αναλυτικά. Έτσι, η προσομοίωση είναι συχνά η μόνη διαθέσιμη μέθοδος μελέτης.
4. Επιτρέπεται ανάλυση της ευαισθησίας του συστήματος η οποία απορρέει από διαφοροποιήσεις των μεταβλητών εισόδου.
5. Δεν έχουμε διατάραξη του πραγματικού συστήματος. Αυτό είναι ένα μεγάλο πλεονέκτημα αφού πολλοί διαχειριστές, αναλυτές δεν είναι πρόθυμοι να εφαρμόσουν πειραματικές στρατηγικές σε ένα πραγματικό σύστημα το οποίο βρίσκεται σε λειτουργία .
6. Η προσομοίωση είναι ένα αποδοτικό εργαλείο εκπαίδευσης.
7. Είναι οικονομική μέθοδος, αφού είναι δυνατό να υλοποιηθεί σε μικρούς υπολογιστές με τη χρήση γλωσσών προγραμματισμού γενικού σκοπού όπως η C, η C++ και η MATLAB.
8. Μπορούν μέσω της προσομοίωσης να συγκριθούν εναλλακτικές προτάσεις τόσο σε επίπεδο σχεδιασμού όσο και σε επίπεδο λειτουργίας του συστήματος, ώστε να προσδιορισθεί η βέλτιστη λύση που ικανοποιεί τις προδιαγραφές που έχουν ορισθεί.

Κάποια από τα μειονεκτήματα είναι τα εξής:

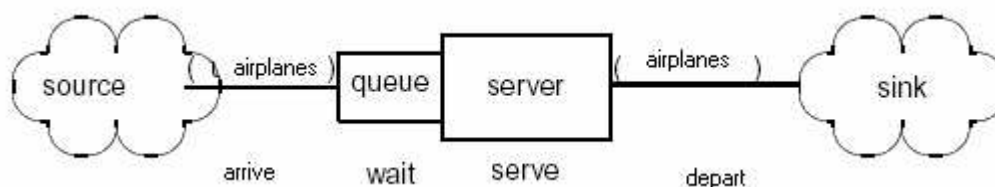
1. Μπορεί να υπάρξουν μη εμφανής κρίσιμες υποθέσεις οι οποίες να παραληφθούν με αποτέλεσμα να διαφοροποιήσουν το πρόβλημα από όπως έχει στην πραγματικότητα. Κάτι τέτοιο θα πρέπει να αποφευχθεί κατά την διαδικασία πιστοποίησης της διαδικασίας προσομοίωσης. Όμως και πάλι η ανίχνευση μιας τέτοιας κατάστασης εξαρτάται από την αυστηρότητα και την επιμέλεια με την οποία πιστοποιούμε το μοντέλο.
2. Οι παράμετροι ενός μοντέλου ίσως είναι δύσκολο να αρχικοποιηθούν. Μπορεί να απαιτούν χρονοβόρες διαδικασίες για την συλλογή, ανάλυση και αποσαφήνιση τους.
3. Τα μοντέλα προσομοίωσης συχνά απαιτούν πολύ χρόνο και πόρους για να αναπτυχθούν. Το κόστος μιας προσομοίωσης μπορεί να ελαττωθεί μέσω μιας εις βάθος κατανόησης του συστήματος που προσομοιώνεται προτού αναπτυχθεί το μοντέλο και μέσω ενός προσεκτικού σχεδιασμού του πειράματος της προσομοίωσης.
4. Ο μεγάλος όγκος αριθμών που παράγονται από μία μελέτη προσομοίωσης ή η εντύπωση που δημιουργούν οι τυχόν γραφικές παραστάσεις των αποτελεσμάτων της, συχνά ενισχύουν μία τάση να δίνεται μεγαλύτερη εμπιστοσύνη στα αποτελέσματα αυτά από όσο πρέπει. Αν το μοντέλο δεν είναι μία αρκετά έγκυρη αναπαράσταση του συστήματος, τα αποτελέσματα της προσομοίωσης, ανεξάρτητα του πόσο εντυπωσιακά είναι, θα προσθέσουν λίγη χρήσιμη πληροφορία για το πραγματικό σύστημα.

1.7 Προσομοίωση ενός Συστήματος Αναμονής - Ουρές Αναμονής

Οι ουρές αναμονής είναι από τα πιο κοινά προβλήματα που εμφανίζονται στην μοντελοποίηση ή προσομοίωση ενός συστήματος και οφείλουν την ύπαρξή τους σε έλλειψη πόρων ενός συστήματος. Η απαίτηση για να αντιμετωπιστούν οι ουρές αναμονής ήταν αυτό που πραγματικά οδήγησε στην ανάπτυξη των γλωσσών προγραμματισμού.

Μία ουρά αναμονής αποτελείται από οντότητες οι οποίες περιμένουν να εξυπηρετηθούν από έναν ή περισσότερους εξυπηρετητές. Στην περίπτωση ενός

αεροδρομίου μπορούμε να θεωρήσουμε ως οντότητες τα αεροπλάνα και ως εξυπηρετητές τις διάφορες εγκαταστάσεις του αεροδρομίου που χρησιμοποιούν τα αεροπλάνα. Έτσι σε ένα αεροδρόμιο έχουμε εμφάνιση συστημάτων αναμονής όταν εμφανίζονται καθυστερήσεις λόγω της απαίτησης χρήσης ενός εξυπηρετητή από περισσότερα του ενός αεροπλάνων. Η όλη διαδικασία έχει ως εξής: έχουμε αφίξεις αεροπλάνων τα οποία ζητούν άμεση εξυπηρέτηση από κάποιον χώρο ενός αεροδρομίου (π.χ. διάδρομο προσγείωσης), όταν ο εξυπηρετητής είναι ελεύθερος δεν έχουμε αναμονή, το αεροπλάνο εξυπηρετείται και μετά αναχωρεί από τον συγκεκριμένο χώρο του αεροδρομίου (εξυπηρετητή). Όταν ο εξυπηρετητής είναι κατειλημμένος από ένα άλλο αεροπλάνο τότε έχουμε αναμονή σε ουρά (queue) μέχρι αυτός να ελευθερωθεί. Όταν ολοκληρωθεί μια εξυπηρέτηση τότε το αεροπλάνο αποχωρεί από το σύστημα αναμονής. Ένα απλό σύστημα αναμονής με έναν εξυπηρετητή παρουσιάζεται στο Σχήμα 1.8:



Σχήμα 1.8 Σύστημα αναμονής ενός εξυπηρετητή

Όπως φαίνεται και από το σχήμα, τα συστήματα αναμονής αποτελούνται από στατικά αντικείμενα τα οποία είναι οι πόροι (server, queue), από δυναμικά αντικείμενα που είναι οι χρήστες (π.χ. αεροπλάνα), και από τα μέρη source όπου δημιουργούνται οι αφίξεις και sink όπου καταλήγουν οι χρήστες όταν εξυπηρετηθούν.

Τα συστήματα αναμονής έχουν ορισμένα χαρακτηριστικά τα οποία και αναφέρουμε παρακάτω:

- 1) Την συνάρτηση πυκνότητας πιθανότητας των χρόνων ανάμεσα στις αφίξεις. Ο χρόνος που μεσολαβεί ανάμεσα σε δύο αφίξεις μπορεί να λαμβάνει μια προκαθορισμένη γνωστή τιμή (ντετερμινιστική προσέγγιση) ή μπορεί να είναι μια τυχαία μεταβλητή που ακολουθεί κάποια κατανομή όπως αυτή της Poisson (στοχαστική προσέγγιση). Εδώ θα πρέπει επίσης να επιλέξουμε το αν θα έχουμε άπειρο ή πεπερασμένο αριθμό αφίξεων.

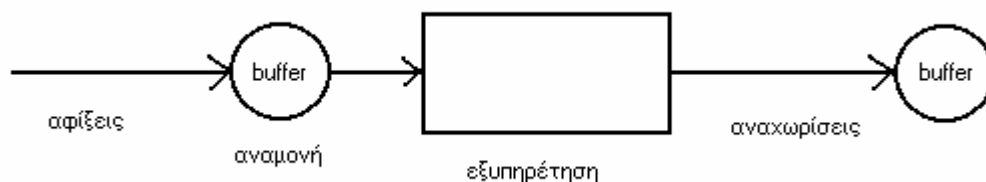
- 2) Την συνάρτηση πυκνότητας πιθανότητας των χρόνων εξυπηρέτησης. Και εδώ, ο χρόνος εξυπηρέτησης μπορεί να λαμβάνει μια προκαθορισμένη γνωστή τιμή (ντετερμινιστική προσέγγιση) ή μπορεί να είναι μια τυχαία μεταβλητή που ακολουθεί κάποια κατανομή όπως αυτή της Poisson (στοχαστική προσέγγιση).
- 3) Τον αριθμό των παράλληλων εξυπηρετητών. Σε μερικά συστήματα αναμονής υπάρχουν περισσότεροι του ενός εξυπηρετητές που εξυπηρετούν ταυτόχρονα τα αεροπλάνα. Οι εξυπηρετητές αυτοί μπορούν να έχουν όλοι την ίδια μία ουρά αναμονής ή να έχει ο καθένας την δικιά του.
- 4) Την πειθαρχία της ουράς. Εδώ καθορίζει ο τρόπος επιλογής του επόμενου αεροπλάνου που πρόκειται να εξυπηρετηθεί. Η πιο συνηθισμένη πειθαρχία είναι η 'Πρώτος ερχόμενος πρώτος εξυπηρετούμενος' (FCFS, first come first served). Υπάρχουν όμως και άλλες πειθαρχίες όπως η 'Τελευταίος ερχόμενος πρώτος εξυπηρετούμενος' (LCFS, last come first served), ή η τυχαία επιλογή του πελάτη προς εξυπηρέτηση, καθώς και αυτή, στην οποία υπάρχουν διαφορετικές προτεραιότητες για κάθε αεροπλάνο (π.χ. αν προέρχονται από διαφορετικές ουρές αναμονής και καταλήγουν στον ίδιο εξυπηρετητή) και το καθένα εξυπηρετείται ανάλογα με την προτεραιότητά του.
- 5) Το μέγιστο επιτρεπόμενο μήκος ουράς. Στις περισσότερες των περιπτώσεων θεωρείται ότι η ουρά αναμονής μπορεί να έχει άπειρο μήκος. Υπάρχουν όμως και συστήματα αναμονής στα οποία ο χώρος που διατίθεται για την ουρά είναι προκαθορισμένος πράγμα που πολλές φορές είναι απαραίτητο για μια ρεαλιστική αντίληψή τους.

Για τη περιγραφή των χαρακτηριστικών κάθε ουράς αναμονής χρησιμοποιείται μια συντομογραφία που αποτελείται από πέντε σύμβολα A/B/m/K/M. Σε αυτή τη συντομογραφία, τα A και B καθορίζουν την κατανομή των χρόνων άφιξης και εξυπηρέτησης αντίστοιχα, το m καθορίζει τον αριθμό των παράλληλων εξυπηρετητών, το K είναι η χωρητικότητα της ουράς αναμονής και τέλος το M είναι ο συνολικός αριθμός των αφίξεων. Τα A, B όταν καθορίζονται από εκθετικές κατανομές (π.χ. Poisson) παίρνουν την τιμή M, ενώ όταν καθορίζονται ντετερμινιστικά παίρνουν την τιμή D. Υποθέτουμε ότι για όλες τις ουρές αναμονής ισχύει η πειθαρχία FCFS.

Θα μας απασχολήσουν τα παρακάτω συγκεκριμένα συστήματα αναμονής:

- $M/D/1/\infty/M$

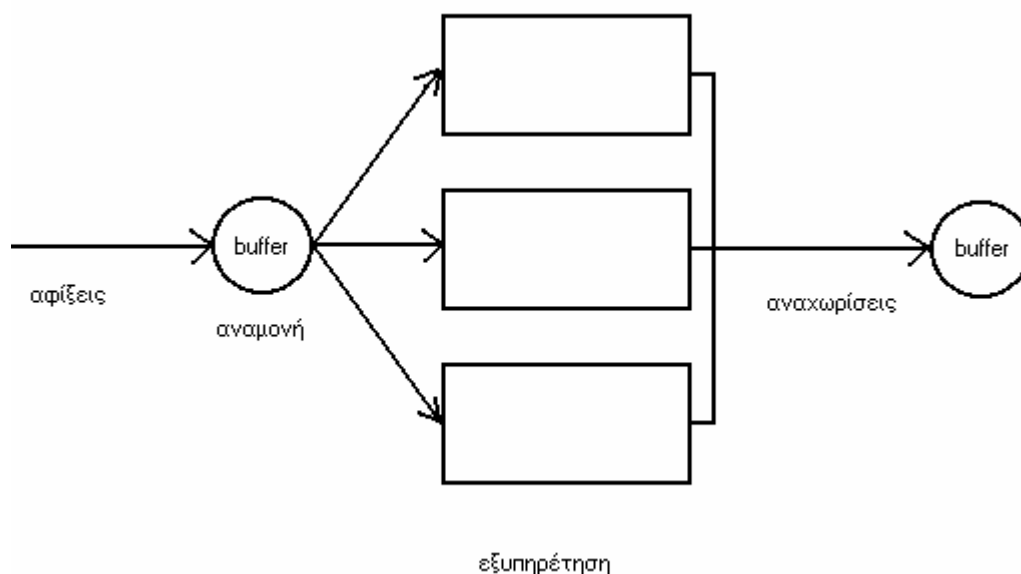
Είναι η περίπτωση όπου έχουμε αφίξεις βάση της κατανομής Poisson ενώ ο χρόνος εξυπηρέτησης είναι σταθερός. Επιπλέον έχουμε έναν εξυπηρετητή άπειρο χώρο για αναμονή και συγκεκριμένο αριθμό αφίξεων. Η περίπτωση αυτή παρουσιάζεται στο Σχήμα 1.9.



Σχήμα 1.9 $M/D/1/\infty/M$

- $M/D/m/\infty/$

Είναι η περίπτωση όπου έχουμε αφίξεις βάση της κατανομής Poisson ενώ ο χρόνος εξυπηρέτησης είναι σταθερός. Επιπλέον έχουμε περισσότερους του ενός εξυπηρετητές με έστω $m=3$ (τρεις εξυπηρετητές), άπειρο χώρο για αναμονή και συγκεκριμένο αριθμό αφίξεων. Η περίπτωση αυτή παρουσιάζεται στο Σχήμα 1.10.



Σχήμα 1.10 $M/D/m$

1.8 Κατανομή Poisson

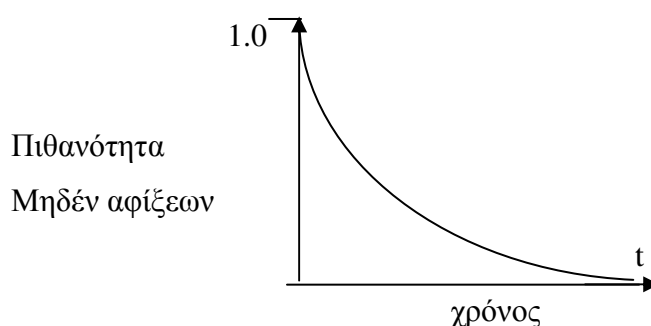
Θα πρέπει να δοθεί ιδιαίτερη σημασία στην κατανομή των τυχαίων αφίξεων η οποία περιγράφει πώς σε οποιοδήποτε δεδομένο χρονικό διάστημα t ο αριθμός των αφίξεων είναι εντελώς τυχαίος. Για την περίπτωση των αφίξεων αεροπλάνων σε ένα αεροδρόμιο η προσέγγιση με μια κατανομή Poisson έχει πολύ καλά αποτελέσματα. Έτσι η πιθανότητα άφιξης x αεροπλάνων την χρονική περίοδο t μπορεί να δοθεί από τον τύπο της κατανομής Poisson, ο οποίος είναι:

$$P(x) = \frac{(\lambda t)^x e^{-\lambda t}}{x!} \quad (1)$$

με λ τον μέσο ρυθμό αφίξεων και την απόκλιση της κατανομής. Έτσι όσο μεγαλώνουν οι αριθμοί σε ένα συγκεκριμένο δείγμα τυχαίων αριθμών Poisson τόσο μεγαλώνει και η διασπορά τους. Από την εξίσωση (1) θέτοντας $x = 0$ έχουμε:

$$P(0) = e^{-\lambda t} \quad (2)$$

Αυτή είναι η πιθανότητα να έχουμε μηδέν αφίξεις το χρονικό διάστημα t . Αυτή είναι μια αρνητική εκθετική συνάρτηση η οποία φαίνεται στο Σχήμα 1.12.



Σχήμα 1.12 Πιθανότητα μηδενικών αφίξεων το χρονικό διάστημα t

Αν αφαιρέσουμε την συνάρτηση (2) από την μονάδα, έχουμε την αθροιστική πιθανότητα (cumulative probability) για μία ή περισσότερες αφίξεις κατά το χρονικό διάστημα t :

$$F(t) = 1 - e^{-\lambda t} \quad (3)$$

Από την θεωρία πιθανοτήτων έχουμε:

$$f(t) = \frac{d}{dt} F(t)$$

Η οποία γίνεται:

$$\underline{f(t)} = \frac{d}{dt} (1 - e^{-\lambda t}) = \lambda e^{-\lambda t} \quad (4)$$

Ο τύπος αυτός καθορίζει την κατανομή του χρόνου μεταξύ των αφίξεων. Είναι η συνάρτηση πυκνότητας της γνωστής εκθετικής κατανομής με μέση τιμή $1/\lambda$. Έτσι μπορούμε να ορίσουμε την κατανομή των αφίξεων από τον χρόνο μεταξύ των αφίξεων ως:

$$a(t) = \lambda e^{-\lambda t} \quad (5)$$

Αυτή είναι μια πολύ σημαντική περίπτωση για τους χρόνους αφίξεων. Η κατανομή Poisson είναι μια διακριτή κατανομή μίας παραμέτρου που λαμβάνει θετικές ακέραιες τιμές, γεγονός που την κάνει απλή σε επίπεδο μαθηματικών.

1.9 Ανάπτυξη προγραμμάτων προσομοίωσης

Η ανάπτυξη προγραμμάτων προσομοίωσης μπορεί να γίνει με δυο διαφορετικούς τρόπους. Ο πρώτος είναι με την χρήση γλωσσών που είναι εξειδικευμένες για προσομοίωση και ο δεύτερος με την χρήση γλωσσών προγραμματισμού γενικής χρήσης όπως C, C++, MATLAB. Στον πρώτο τρόπο έχουμε αυτοματοποίηση των βασικών λειτουργιών της προσομοίωσης όπως του μηχανισμού του χρόνου, της παραγωγής τυχαίων αριθμών, του ελέγχου συνθηκών τερματισμού κλπ. Δεν προσφέρουν όμως την ευεξία που έχουμε από τον δεύτερο τρόπο, όπου μπορεί κανείς να προγραμματίσει κάθε είδους σύστημα όσο πολύπλοκο και να είναι. Από την άλλη μεριά πολλές εξειδικευμένες γλώσσες έχουν την δυνατότητα ενσωμάτωσης σε αυτές ρουτινών και υποπρογραμμάτων γραμμένα σε άλλη γλώσσα πράγμα που μπορούμε να πούμε ότι δίνει και σε αυτές κάποια ευεξία. Όταν όμως ο αριθμός των ρουτινών αυτών είναι μεγάλος, γίνεται δύσκολη η συντήρηση του προγράμματος και η ανίχνευση των λαθών. Τα λάθη αυτά εμφανίζονται συνήθως στην επικοινωνία της εξειδικευμένης γλώσσας με τις ρουτίνες

του χρήστη. Έτσι είναι προτιμότερο να αναπτυχθεί η εφαρμογή εξολοκλήρου σε μια γλώσσα γενικής χρήσεως, αφού έτσι ο προγραμματιστής έχει πλήρη έλεγχο της ροής του προγράμματος και της διασύνδεσης των τμημάτων του. Στην παρούσα διπλωματική έγινε η επιλογή της χρήσης γλώσσας προγραμματισμού γενικής χρήσης και συγκεκριμένα αυτή του MATLAB λόγω του πλεονεκτήματος της ευεξίας.

Παρακάτω θα κάνουμε μια αναφορά για τις εξειδικευμένες γλώσσες προσομοίωσης και θα αναφερθούμε ειδικά σε κάποιες από αυτές. Το ευρύ πεδίο εφαρμογών της προσομοίωσης δημιούργησε την ανάγκη για την ανάπτυξη εξειδικευμένων γλωσσών προσομοίωσης που επιτρέπουν στο χρήστη να προσομοιώνει το μοντέλο του συστήματός του με μερικές απλές εντολές. Το πλεονέκτημα λοιπόν που παρέχουν είναι η δραματική μείωση του χρόνου που απαιτείται για τον προγραμματισμό ενός μοντέλου. Αυτό είναι αρκετά σημαντικό γιατί επιτρέπει στον αναλυτή, που κάνει χρήση της προσομοίωσης στη μελέτη του, να αφιερώσει περισσότερο χρόνο στις άλλες φάσεις της μελέτης του όπως είναι η ανάλυση των αποτελεσμάτων και η επαλήθευση του μοντέλου που χρησιμοποιεί. Εκτός από την βοήθεια στο στάδιο του προγραμματισμού, οι εξειδικευμένες γλώσσες προσομοίωσης προσφέρουν δυνατότητες μοντελοποίησης με παραστατική καθοδήγηση στην ανάπτυξη του μοντέλου και βοήθεια στην τεκμηρίωση και παρουσίαση των αποτελεσμάτων.

Υπάρχουν όμως και μειονεκτήματα όπως το υψηλό κόστος σε χρήματα της απόκτησης και συντήρησης τους. Επίσης οι αναλυτές θα πρέπει να αφιερώσουν αρκετό χρόνο στην εκμάθηση της εξειδικευμένης γλώσσας προσομοίωσης που πρόκειται να χρησιμοποιήσουν η ευελιξία της οποίας μπορεί να είναι μειωμένη και αυτό να δημιουργεί και περαιτέρω προβλήματα. Ακόμα οι εξειδικευμένες γλώσσες προσομοίωσης έχουν υψηλές απαιτήσεις σε υπολογιστική ισχύ και μνήμη του υπολογιστή σε σχέση με γλώσσες προσομοίωσης που αναπτύσσονται πάνω σε γενικές γλώσσες προγραμματισμού.

Οι εξειδικευμένες γλώσσες προσφέρουν έτοιμα αρκετά στοιχεία που χρειάζονται για την μοντελοποίηση και εκτέλεση ενός προγράμματος προσομοίωσης. τα πιο συνηθισμένα από τα οποία είναι:

- 1) Άποψη του κόσμου με προσανατολισμό σε γεγονότα, δραστηριότητες, διεργασίες.
- 2) Μηχανισμό ροής του χρόνου.

- 3) Γεννήτρια τυχαίων που να ακολουθούν την κατανομή που προσδιορίζει ο χρήστης.
- 4) Μηχανισμούς συλλογής δεδομένων του μοντέλου.
- 5) Παροχή στοιχειώδους στατιστικής ανάλυσης των δεδομένων που έχουν συλλέξει.
- 6) Παροχή διαγνωστικών μηχανισμών και ανίχνευσης λαθών.

Κάποιες από τις πιο σημαντικές εξειδικευμένες γλώσσες προσομοίωσης είναι και οι παρακάτω:

- 1) SIMSCRIPT: δημιουργήθηκε από τον Markowitz και τους συνεργάτες του την δεκαετία του 1960 και σήμερα υπάρχουν εκδόσεις σχεδόν για κάθε είδους υπολογιστικό σύστημα. Υλοποιείται στις γλώσσες assembly και C. Οι πλατφόρμες για τις οποίες είναι διαθέσιμη είναι PC, πολλές πλατφόρμες UNIX, και πολλά μεγάλα συστήματα διαμοιρασμού χρόνου. Μέθοδος του προγραμματισμού είναι με εντολές και η διαχείριση μνήμης είναι δυναμική. Έχει άποψη του κόσμου ως γεγονότος ή διεργασίας και το κόστος της είναι υψηλό.
- 2) MODSIM: είναι μια γλώσσα προσομοίωσης νέας γενιάς. Αν και είναι προσανατολισμένη στην προσομοίωση, μπορεί να χρησιμοποιηθεί και ως γλώσσα γενικής χρήσεως. Συνδυάζει τα πλεονεκτήματα των αντικειμενοστραφών γλωσσών με δυνατότητες προσομοίωσης διακριτών γεγονότων. Λόγω της δυνατότητας επαναχρησιμοποίησης κώδικα που προσφέρει, χρησιμοποιείται περισσότερο για την κατασκευή μεγάλων διεργασιακών μοντέλων. Ακόμα έχει ενδιάμεσο μεταγλωττιστή, δηλαδή δεν παράγει κώδικα σε γλώσσα μηχανής αλλά σε πηγαίο κώδικα άλλης γλώσσας η οποία είναι η C++. Είναι διαθέσιμη για πλατφόρμες PC και UNIX, και η μέθοδος προγραμματισμού είναι με εντολές. Διαχειρίζεται την μνήμη δυναμικά και το κόστος είναι υψηλό.
- 3) GPSS: αναπτύχθηκε την δεκαετία του 1960 αλλά ακόμα και σήμερα έχει πολλούς υποστηρικτές και είναι διαθέσιμη για κάθε πλατφόρμα υπολογιστικού συστήματος. Είναι ίσως η μόνη γλώσσα προσομοίωσης που σχεδιάστηκε από την αρχή για αναλυτές οι οποίοι δεν είχαν καμία σχέση με τον προγραμματισμό. Η ανάπτυξη προγραμμάτων στην GPSS γίνεται με την παράθεση μερικών απλών εντολών που καθορίζουν την κίνηση των

οντοτήτων μέσα στο μοντέλο. Οι βασικές της έννοιες είναι η δοσοληψία, η εγκατάσταση και το μπλοκ. Οι δοσοληψίες αντιστοιχούν σε ένα μέρος των οντοτήτων του μοντέλου και έχουν τα ακόλουθα χαρακτηριστικά:

- Είναι δυναμικές οντότητες που κινούνται μέσα στο μοντέλο όπως για παράδειγμα τα αεροπλάνα σε ένα αεροδρόμιο.
- Κατά την διάρκεια της προσομοίωσης, οι δοσοληψίες έρχονται στο μοντέλο και φεύγουν από αυτό.
- Είναι δυνατό να υπάρχουν ταυτόχρονα πολλές παρόμοιες ενεργές δοσοληψίες στο μοντέλο.
- Η κατάσταση των δοσοληψιών, δηλαδή η θέση τους στο μοντέλο και οι αριθμητικές τους ιδιότητες, ενημερώνονται σειριακά από τον επεξεργαστή της GPSS.
- Όταν μια δοσοληψία έχει αρχίσει να κινείται μέσα στο μοντέλο, η κίνησή της από μπλοκ σε μπλοκ συνεχίζεται μέχρι να συμβεί ένα από τρία γεγονότα. Πρώτον όταν η δοσοληψία εισέρχεται σε ένα μπλοκ που έχει ως σκοπό να την κρατήσει εκεί. Δεύτερον, όταν η δοσοληψία εισέρχεται σε ένα μπλοκ που έχει ως σκοπό να την αφαιρέσει από το μοντέλο. Τρίτον όταν το επόμενο μπλοκ στην πορεία της δοσοληψίας αρνείται την είσοδό της σε αυτό.

Τα μπλοκ είναι περίπου αντίστοιχα με τις εντολές σε μια γλώσσα προγραμματισμού με την έννοια ότι ο αναλυτής αναπτύσσει το πρόγραμμα της προσομοίωσης του μοντέλου παραθέτοντας μια σειρά από μπλοκ. Ο σκοπός του κάθε μπλοκ είναι είτε ο έλεγχος της κίνησης των δοσοληψιών, είτε η συλλογή στατιστικών στοιχείων της προσομοίωσης. Εκτός από τα μπλοκ υπάρχουν και εντολές ελέγχου με τις οποίες καθορίζεται η συνολική πορεία της προσομοίωσης. Οι εγκαταστάσεις είναι μια κατηγορία μπλοκ που σκοπό έχουν την παροχή υπηρεσιών προς τις δοσοληψίες. Μπορούμε να τις αντιστοιχίσουμε με τους εξυπηρετητές ενός συστήματος ουράς.

Η διαχείριση του χρόνου γίνεται από τον επεξεργαστή της γλώσσας GPSS χωρίς να απαιτείται καμία προγραμματιστική παρέμβαση από τον αναλυτή και έχει τις παρακάτω ιδιότητες:

- Καταγράφει μόνο ακέραιες τιμές.

- Η μονάδα του χρόνου είναι αυθαίρετη και καθορίζεται από τον αναλυτή ανάλογα με τις απαιτήσεις του μοντέλου. Για τον λόγο αυτό το ότι το ρολόι καταγράφει μόνο ακέραιες τιμές δεν αποτελεί πρόβλημα γιατί ο αναλυτής μπορεί να χρησιμοποιήσει οποιοδήποτε ακέραιο πολλαπλάσιο της χρονικής του μονάδας.
- Αρχικοποιείται στην τιμή μηδέν.
- Καθώς γίνεται η ενημέρωση του συστήματος, το ρολόι παραμένει ακίνητο αν και παρέρχεται πραγματικός χρόνος.
- Αυξάνεται κατά χρονικά διαστήματα που είναι συνάρτηση των συνθηκών του συστήματος. Άρα ο μηχανισμός χρόνου αντιστοιχεί στον μηχανισμό ροής χρόνου επόμενου γεγονότος.

Η συλλογή στατιστικών στοιχείων και η παρουσίαση των αποτελεσμάτων της προσομοίωσης γίνεται αυτόματα από την GPSS. Και ο αναλυτής όμως αν επιθυμεί μπορεί να εισάγει ειδικές εντολές που αφορούν αυτές τις ενέργειες. Τέλος, υλοποιείται σε γλώσσα assembly και το κόστος της είναι μέσο.

Κεφάλαιο 2

Περιγραφή Συστήματος Αεροδρομίου Για Προσομοίωση

2.1 Γενική περιγραφή συστήματος αεροδρομίου

Το είδος του συστήματος που θα μας απασχολήσει είναι αυτού του αεροδρομίου και ειδικότερα το μέρος του που αφορά τις κινήσεις εδάφους των αεροπλάνων. Καταρχήν έχουμε να κάνουμε με ένα δυναμικό σύστημα εφόσον αλλάζει η κατάσταση του κατά την διάρκεια του χρόνου. Αυτό οφείλεται στις κινήσεις των αεροπλάνων μέσα σε αυτό. Λόγο της δυναμικότητάς του αυτής γίνεται η επιλογή της προσομοίωσης ως προσεγγιστικής μεθόδου για την ανάλυσή του. Επιπλέον ένα τέτοιο σύστημα είναι και διακριτό αφού οι μεταβλητές (αεροπλάνα) του συστήματος αλλάζουν με χρονικά ασυνεχή τρόπο. Ο αριθμός των αεροπλάνων μέσα στο σύστημα του αεροδρομίου αλλάζει με διακριτό τρόπο αφού είναι αδύνατο να έχουμε ένα κλάσμα αεροπλάνου στο αεροδρόμιο. Η κατάσταση του συστήματος στην περίπτωση αυτή αλλάζει μόλις αρχίσει ή μόλις ολοκληρωθεί μια δραστηριότητα ενός αεροπλάνου το οποίο βρίσκεται μέσα στο σύστημα του αεροδρομίου. Μια άλλη παρατήρηση που μπορούμε να κάνουμε για ένα σύστημα αεροδρομίου είναι ότι είναι στοχαστικό. Για παράδειγμα ποτέ δεν μπορούμε να είμαστε σίγουροι για την ακριβή ώρα άφιξης ενός αεροπλάνου ή για ενδεχόμενες καθυστερήσεις του αεροπλάνου όσο αυτό βρίσκεται στο έδαφος, γεγονός που εισάγει κάποια τυχαιότητα στο σύστημα και άρα το κάνει στοχαστικό. Βέβαια μπορεί ορισμένες λειτουργίες στο αεροδρόμιο να έχουν και ντετερμινιστικό χαρακτήρα όπως η κίνηση ενός αεροπλάνου σε κάποιο διάδρομο που απαιτεί κάποιο προκαθορισμένο χρόνο. Όμως εφόσον έστω και μια λειτουργία του αεροδρομίου έχει στοχαστικό χαρακτήρα τότε χαρακτηρίζουμε όλο το σύστημα ως στοχαστικό. Στην περίπτωσή μας, η στοχαστικότητα εισέρχεται στο ότι επιλέγουμε μια γεννήτρια τυχαίων αριθμών για δημιουργία των αφίξεων και στο ότι έχουμε εμφάνιση ουρών σε διάφορα μέρη του αεροδρομίου.

Τις δραστηριότητες ενός αεροπλάνου σε ένα αεροδρόμιο μπορούμε να την χωρίσουμε σε ορισμένα βασικά μέρη [7]. Πρώτον έχουμε την κίνησή του στον διάδρομο προσγείωσης. Δεύτερον έχουμε την κίνησή του στον διάδρομο που οδηγεί στον χώρο στάθμευσης και την παραμονή του εκεί για κάποιο χρονικό διάστημα. Τρίτον έχουμε την κίνησή του στον διάδρομο που οδηγεί από τον χώρο στάθμευσης προς τον διάδρομο απογείωσης. Τέλος έχουμε την κίνησή του στον διάδρομο απογείωσης.

Όλες αυτές οι κινήσεις ενός αεροπλάνου μέσα σε ένα αεροδρόμιο συνιστούν την ύπαρξη ορισμένων στοιχειωδών συστημάτων τα οποία συνθέτουν το σύστημα του αεροδρομίου το οποίο είναι υπεύθυνο για την εξυπηρέτηση των αεροπλάνων από την προσγείωση μέχρι την απογείωσή τους. Οπότε ορίζουμε τα παρακάτω πέντε στοιχειώδη συστήματα τα οποία είναι συνδεδεμένα με συγκεκριμένες δραστηριότητες των αεροπλάνων. Οι δραστηριότητες αυτές αποτελούν τα γεγονότα που συμβαίνουν στο σύστημα του αεροδρομίου και έχουν ως αποτέλεσμα τις αλλαγές στην κατάστασή του.

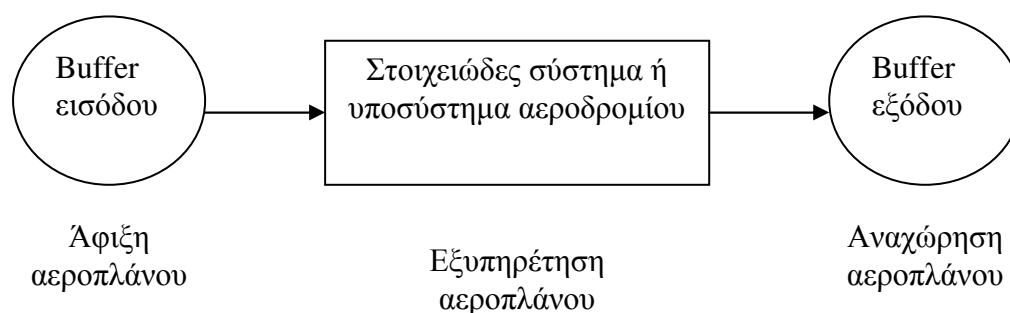
Τα στοιχειώδη συστήματα ενός αεροδρομίου είναι:

1. Ο διάδρομος προσγείωσης (landing runway).
2. Ο διάδρομος τροχοδρόμησης προς θέση στάθμευσης (taxiing in).
3. Ο χώρος στάθμευσης.
4. Ο διάδρομος τροχοδρόμησης προς διάδρομο απογείωσης (taxiing out).
5. Ο διάδρομος απογείωσης (take off runway).

Τα παραπάνω στοιχειώδη συστήματα ενός αεροδρομίου μπορούν να αποτελούνται από δύο ή παραπάνω στοιχειώδη υποσυστήματα τα οποία να έχουν ίδια η παρόμοια μεταξύ τους λειτουργία. Για παράδειγμα μπορεί να υπάρχουν πολλές θέσεις για στάθμευση η κάθε μία από τις οποίες να αποτελεί ένα στοιχειώδες υποσύστημα του συστήματος χώρου στάθμευσης. Επίσης η χρήση κάποιων στοιχειωδών συστημάτων μπορεί να είναι διττή ,οπότε για να διαχωρίσουμε τις διαφορετικές χρήσεις τα χωρίζουμε κατάλληλα σε στοιχειώδη υποσυστήματα θέτοντας ταυτόχρονα και τους αναγκαίους περιορισμούς.

Το κάθε στοιχειώδες σύστημα συνδέεται άμεσα με τις δραστηριότητες των αεροπλάνων αφού χρησιμοποιείται από αυτά . Έτσι έχουμε δύο βασικές δραστηριότητες , τις αφίξεις και τις αναχωρήσεις των αεροπλάνων από αυτά ,οι οποίες συνδετικά οδηγούν από την άφιξη ενός αεροπλάνου στο αεροδρόμιο ως την αναχώρησή του από αυτό. Οι δραστηριότητες αυτές συνδέονται άμεσα με τον χρόνο που απαιτείται για την αποπεράτωση της ενέργειας που επιτελείται μέσα σε κάθε στοιχειώδες σύστημα (χρόνος εξυπηρέτησης) .Ο χρόνος αυτός μπορεί να είναι τυχαίος ή καθορισμένος. Στην παρούσα διπλωματική επιλέχθηκε να είναι καθορισμένος αφού σκοπός είναι η προσομοίωση της κίνησης των αεροπλάνων σε ένα αεροδρόμιο σε επίπεδο δρομολόγησης και όχι βελτιστοποίησης.

Για να είναι δυνατή η επίβλεψη των αεροπλάνων που κινούνται μεταξύ των στοιχειωδών συστημάτων και υποσυστημάτων του αεροδρομίου εισάγουμε κάποιες προσωρινές μονάδες αποθήκευσης με τις οποίες παρακολουθούμε την είσοδο και την έξοδο των αεροπλάνων από αυτά. Την προσωρινή μονάδα αποθήκευσης εισόδου την ονομάζουμε buffer εισόδου ενώ αυτή της εξόδου buffer εξόδου. Την διάταξη αυτή την αναπαριστούμε στο Σχήμα 2.1.



Σχήμα 2.1 Προσωρινές μονάδες αποθήκευσης

Με τον τρόπο αυτό όταν ένα αεροπλάνο θέλει να κάνει χρήση κάποιου στοιχειώδους συστήματος ή υποσυστήματος του αεροδρομίου στο οποία βρίσκεται, αρχικά εισέρχεται στον Buffer εισόδου του συγκεκριμένου στοιχειώδους συστήματος ή υποσυστήματος, αυξάνοντας την τιμή του κατά μία μονάδα, και προτού εισέλθει σε αυτό για να εξυπηρετηθεί γίνεται ένας έλεγχος για το αν αυτό το στοιχειώδες σύστημα ή υποσύστημα είναι διαθέσιμο προς χρήση ή όχι. Αν δεν είναι διαθέσιμο σημαίνει ότι ήδη χρησιμοποιείται από κάποιο άλλο αεροπλάνο και τότε το αεροπλάνο που μόλις ζήτησε εξυπηρέτηση περιμένει στον Buffer εισόδου έως ότου απελευθερωθεί ο εξυπηρετητής του. Αν το στοιχειώδες σύστημα ή υποσύστημα είναι διαθέσιμο για χρήση τότε θεωρούμε ότι το αεροπλάνο εισέρχεται στον συγκεκριμένο χώρο του αεροδρομίου τον οποίο συμβολίζει και εξυπηρετείται. Ο χρόνος εξυπηρέτησης ορίζεται κάθε φορά από το στοιχειώδες σύστημα ή υποσύστημα που έχουμε και είναι προκαθορισμένος π.χ. ο χρόνος παραμονής ενός αεροπλάνου στον διάδρομο προσγείωσης κατά την προσγείωσή του είναι ένα λεπτό. Αφού το αεροπλάνο εξυπηρετηθεί εισέρχεται στον Buffer εξόδου του στοιχειώδους συστήματος ή υποσυστήματος και στο σημείο αυτό είναι που μειώνουμε τον buffer εισόδου κατά μία μονάδα και αυξάνουμε τον Buffer εξόδου κατά μία. Από αυτό το

σημείο έχουμε δύο περιπτώσεις. Να προχωρήσει το αεροπλάνο στο επόμενο στοιχειώδες σύστημα ή υποσύστημα του αεροδρομίου ή αν βρισκόταν στο διάδρομο απογείωσης να τερματιστεί η προσομοίωση του συγκεκριμένου αεροπλάνου.

Στους buffers μπορούμε να έχουμε συσσώρευση άνω του ενός αεροπλάνου πράγμα το οποίο μας οδηγεί στην θεωρία αναμονής ουρών. Ως πειθαρχία της ουράς επιλέγουμε την 'Πρώτος ερχόμενος πρώτος εξυπηρετούμενος' (FCFS, first come first served). Θεωρούμε την χωρητικότητα των buffers, δηλαδή το μέγιστο επιτρεπόμενο μήκος ουράς, ως άπειρη. Οι χρόνοι μεταξύ των αφίξεων αεροπλάνων θεωρούνται τυχαίοι εφόσον και το σύστημά μας του αεροδρομίου είναι στοχαστικό, ενώ οι χρόνοι εξυπηρέτησης επιλέχθηκαν ως σταθεροί και προκαθορισμένοι. Επίσης είναι δυνατόν να έχουμε παραπάνω από έναν εξυπηρετητές σε ένα στοιχειώδες σύστημα, αν και αντιμετώπιση ενός τέτοιου σεναρίου αντιμετωπίζεται συνήθως με το να διαχωρίζουμε τον κάθε εξυπηρετητή θεωρώντας τον ως στοιχειώδες υποσύστημα.

Η Εξέλιξη Σταθερής Αύξησης του Χρόνου επιλέχθηκε ως μηχανισμός για την εξέλιξη του χρόνου στην εφαρμογή μας. Αφού τα αεροπλάνα καταφθάνουν στο αεροδρόμιο με ακρίβεια λεπτού της ώρας και αφού οι χρόνοι λειτουργίας των στοιχειωδών συστημάτων και υποσυστημάτων είναι πολλαπλάσια του ενός λεπτού είναι εφικτή η προσέγγιση με αυτό τον μηχανισμό χωρίς απώλειες στην απόδοση της προσομοιωτικής διάταξης. Ουσιαστικά μπορούμε να προσομοιώσουμε την κίνηση των αεροπλάνων σε ένα αεροδρόμιο για μία μέρα ελέγχοντας κάθε λεπτό (σύνολο $24 \cdot 60 = 1440$ λεπτά) τι δραστηριότητες έχουμε, πράγμα το οποίο για έναν υπολογιστή δεν είναι καθόλου χρονοβόρο και μπορεί να ολοκληρωθεί μέσα σε λίγα δευτερόλεπτα. Με τον τρόπο αυτό αποφεύγουμε την πολυπλοκότητα της εξέλιξης με βάση το χρόνο του επομένου γεγονότος και άρα και πιθανά λάθη στην χρονοδρομολόγηση της κίνησης των αεροπλάνων σε ένα αεροδρόμιο. Επίσης αν θεωρήσουμε ότι έχουμε να κάνουμε με ένα κομβικό αεροδρόμιο τότε είναι επόμενο και οι αφίξεις των αεροπλάνων σε αυτό να είναι αρκετά πυκνές, πιθανότατα ακόμα και σχεδόν κάθε λεπτό. Αυτό μας οδηγεί στην συγκεκριμένη επιλογή του μηχανισμού εξέλιξης του χρόνου βάση σταθερής αύξησης.

2.2 Παρουσίαση του υπό μελέτη αεροδρομίου

Ως παράδειγμα για την παρουσίαση της γλώσσας προσομοίωσης που αναπτύχθηκε επιλέξαμε τον κρατικό αερολιμένα Χανίων, "Κ. Δασκαλογιάννης"[8].

Για τον λόγο αυτό προχωρούμε στην λεπτομερή περιγραφή των εγκαταστάσεών του, που αφορούν κάθε πιθανή κίνηση ενός αεροπλάνου στο έδαφος από την στιγμή της προσγείωσης ως και την στιγμή της απογείωσης.

Όπως και σε κάθε αεροδρόμιο, έτσι και στο "Κ. Δασκαλογιάννης" μπορούμε να αντιστοιχίσουμε τις δραστηριότητες που πραγματοποιεί ένα αεροπλάνο στο έδαφος με συγκεκριμένα στοιχειώδη συστήματα που αποτελούν το υπό μελέτη αεροδρόμιο. Ακολουθώντας την μεθοδολογία αυτή έχουμε τα παρακάτω στοιχειώδη συστήματα:

1. Ένα διάδρομο (runway) ο οποίος εξυπηρετεί τις προσγειώσεις και τις απογειώσεις στο αεροδρόμιο. Χρησιμοποιείται από τα αεροπλάνα είτε για την προσγείωσή τους είτε για την απογείωσή τους και δεν επιτρέπεται η χρήση του και για τις δύο αυτές ενέργειες ταυτόχρονα.
2. Ένα διάδρομο τροχοδρόμησης (taxiing in) των αεροπλάνων από τον διάδρομο προσγείωσης-απογείωσης προς τις θέσεις στάθμευσης. Χρησιμοποιείται μόνο από αεροπλάνα τα οποία μόλις έχουν προσγειωθεί.
3. Τις θέσεις στάθμευσης των αεροπλάνων όπου έχουμε την αποβίβαση-επιβίβαση των επιβατών και τον ανεφοδιασμό των αεροπλάνων(service). Το αεροδρόμιο "Κ. Δασκαλογιάννης" έχει δυναμικό έξι θέσεων στάθμευσης για τα αεροπλάνα.
4. Ένα διάδρομο τροχοδρόμησης (taxiing out) των αεροπλάνων από τους χώρους στάθμευσης προς τον διάδρομο προσγείωσης-απογείωσης. Χρησιμοποιείται μόνο από αεροπλάνα τα οποία κατευθύνονται προς απογείωση.

Για διευκόλυνση της μελέτης μας δίνουμε συγκεκριμένα ονόματα σε κάθε ένα από τα παραπάνω στοιχειώδη συστήματα. Επίσης όπου κρίνουμε αναγκαίο χάριν ευκολίας διαιρούμε κάποια στοιχειώδη συστήματα σε στοιχειώδη υποσυστήματα.

Έτσι ονομάζουμε τον διάδρομο προσγείωσης-απογείωσης στοιχειώδες σύστημα Α και ορίζουμε σε αυτό δύο στοιχειώδη υποσυστήματα. Το Α1 όπου πραγματοποιούνται οι προσγειώσεις και το Α2 όπου πραγματοποιούνται οι απογειώσεις. Όπως προαναφέραμε εδώ έχουμε τον περιορισμό της μη ταυτόχρονης λειτουργίας των δύο αυτών στοιχειωδών υποσυστημάτων Α1, Α2. Τον διάδρομο τροχοδρόμησης taxiing-in τον ονομάζουμε στοιχειώδες σύστημα Β1 ενώ τον διάδρομο τροχοδρόμησης taxiing-out τον ονομάζουμε στοιχειώδες σύστημα D1. Οι

θέσεις στάθμευσης αποτελούν το στοιχειώδες σύστημα C το οποίο μπορούμε να το χωρίσουμε σε τόσα στοιχειώδη υποσυστήματα όσες είναι και οι θέσεις, δηλαδή στην συγκεκριμένη περίπτωση σε έξι. Σε αυτά δίνουμε τις ονομασίες C1, C2, C3, C4, C5 και C6.

Επίσης για να είναι δυνατή η προσομοίωση του αεροδρομίου για διάφορα σενάρια αφίξεων αεροπλάνων σε αυτό ,κάνουμε χρήση μιας γεννήτριας τυχαίων αφίξεων πιθανοτικής κατανομής . Η πιθανοτική κατανομή που επιλέχθηκε είναι η Poisson αφού προσεγγίζει αρκετά ικανοποιητικά τις αφίξεις αεροπλάνων. Έτσι ελλείπει στοιχείων εισόδου για το σύστημά μας μπορούμε ανάλογα με τις απαιτήσεις μας να παράγουμε προσεγγιστικά συγκεκριμένο αριθμό αφίξεων αεροπλάνων κατά την διάρκεια κάποιου χρονικού διαστήματος που επιθυμούμε. Με τον τρόπο αυτό μπορούμε να ελέγχουμε ανά πάσα στιγμή τι όρια έχει ως προς την χρήση του το αεροδρόμιο. Οπότε θεωρούμε και ένα επιπλέον στοιχειώδες σύστημα ,αυτό της γεννήτριας τυχαίων αφίξεων κατανομής Poisson , και το ονομάζουμε ανάλογα.

Με την κατάλληλη συνένωση αυτών των στοιχειωδών συστημάτων και υποσυστημάτων έχουμε την σύνθεση του συστήματος του αεροδρομίου "Κ. Δασκαλογιάννης" και την δυνατότητα δημιουργίας χρονοδιαγράμματος της κίνησης των αεροπλάνων μέσα σε αυτό . Για την επίτευξη του στόχου αυτού είναι αναγκαία η χρήση προσωρινών μονάδων αποθήκευσης εισόδου και εξόδου (input output buffers) για κάθε στοιχειώδες σύστημα ή υποσύστημα του αεροδρομίου .Στον Πίνακα 2.1 καταγράφονται αναλυτικά όλα τα στοιχειώδη συστήματα, υποσυστήματα του αεροδρομίου με τους buffers που τους αντιστοιχούν.

Παρατηρούμε ότι ο buffer b1 ως είσοδος στο στοιχειώδες υποσύστημα A1 αποτελεί την είσοδο στο συνολικό σύστημα του αεροδρομίου και εκεί αποθηκεύονται αρχικά οι αφίξεις των αεροπλάνων σε αυτό. Επίσης ο buffer b6 ως έξοδος από το στοιχειώδες υποσύστημα A2 αποτελεί και την έξοδο του συνολικού συστήματος του αεροδρομίου και εκεί αποθηκεύονται οι αναχωρήσεις των αεροπλάνων από αυτό. Όλοι οι υπόλοιποι buffers (b2, b3, b4, b5) ονομάζονται ενδιάμεσοι και είναι οι συνδετικοί κρίκοι μεταξύ των διαφόρων στοιχειωδών συστημάτων και υποσυστημάτων τα οποία πλαισιώνουν το αεροδρόμιο.

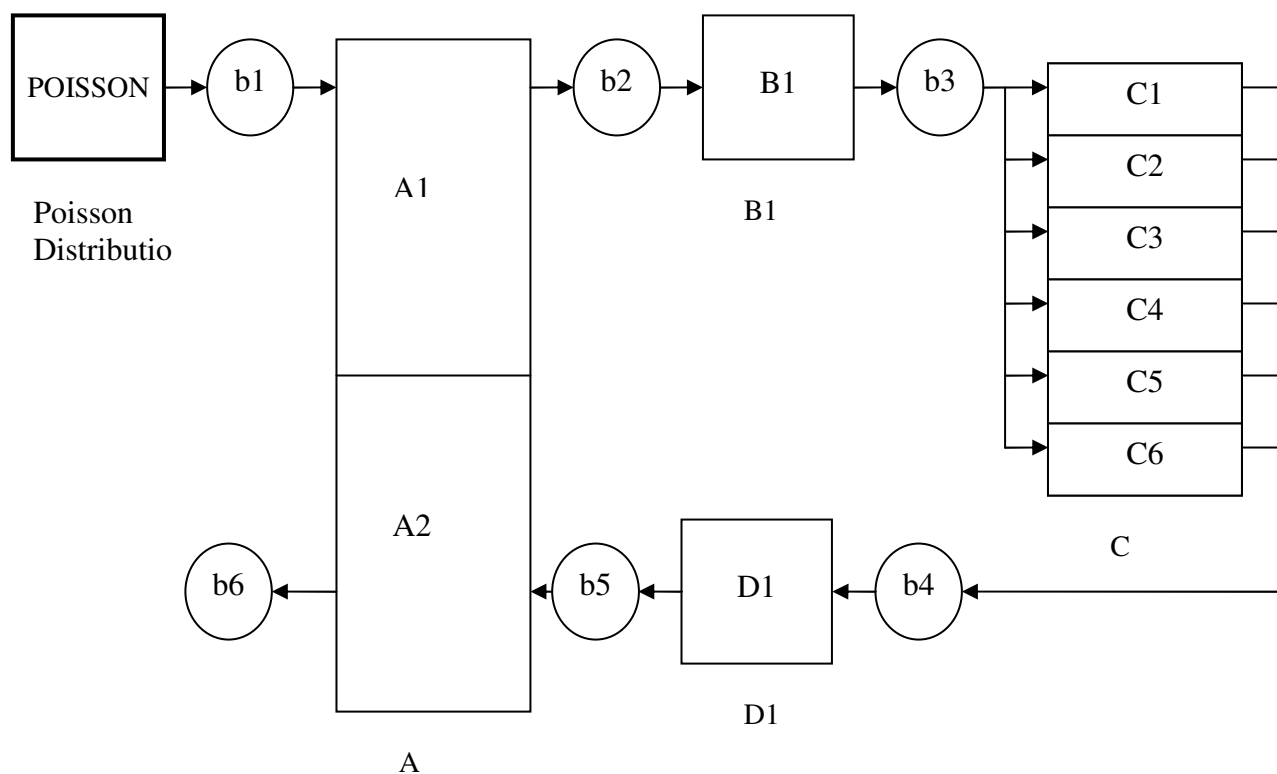
Στοιχειώδες Σύστημα ή Υποσύστημα	Buffer εισόδου	Buffer εξόδου
Poisson	--	b1
A1	b1	b2
A2	b5	b6
B1	b2	b3
C1	b3	b4
C2	b3	b4
C3	b3	b4
C4	b3	b4
C5	b3	b4
C6	b3	b4
D1	b4	b5

Πίνακας 2.1 buffers στοιχειωδών συστημάτων ή υποσυστημάτων

Όταν έχουμε συσσώρευση άνω του ενός αεροπλάνου στους παραπάνω buffers, εκτός βέβαια του b6 ο οποίος δεν μας απασχολεί αφού είναι η τελική έξοδος του αεροδρομίου, τότε ακολουθείται ο μηχανισμός 'Πρώτος ερχόμενος πρώτος εξυπηρετούμενος'.

Στο Σχήμα 2.2 φαίνονται οι εγκαταστάσεις του αεροδρομίου "Κ.

Δασκαλογιάννης", ως σχεδιάγραμμα, με τον τρόπο που περιγράφηκαν παραπάνω .



Σχήμα 2.2 Αεροδρόμιο «Κ. Δασκαλογιάννης»

Στο συγκεκριμένο αεροδρόμιο έχουμε μια ιδιομορφία. Αυτή είναι η ύπαρξη ενός και μόνο διάδρομου όπου γίνονται και οι προσγειώσεις και οι απογειώσεις. Έτσι θα πρέπει να δώσουμε κατάλληλες προτεραιότητες για πώς θα χρησιμοποιηθεί σε περίπτωση που την ίδια χρονική στιγμή έχουμε ένα αεροπλάνο στον αέρα που θέλει να προσγειωθεί και ένα άλλο στο έδαφος που θέλει να απογειωθεί. Γενικός κανόνας είναι να δίνεται προτεραιότητα σε αυτό που είναι στον αέρα αφού στην περίπτωση αυτή, κάθε λεπτό που περνάει χωρίς να εξυπηρετείται κοστίζει πολύ περισσότερο σε χρήματα σε σχέση με ένα αεροπλάνο που βρίσκεται στο έδαφος. Επιπλέον τα καύσιμά του μπορεί να είναι περιορισμένα οπότε η προσγείωσή του να είναι άμεση ανάγκη. Από την άλλη μεριά αν έχουμε πολύ συχνές αφίξεις στο αεροδρόμιο με την λογική να δίνεται η προτεραιότητα στον αέρα είναι βέβαιο ότι κάποια στιγμή θα έχουμε υπερβολική συσσώρευση αεροπλάνων στο έδαφος ,τα οποία περιμένουν για απογείωση .Αυτό είναι φυσικό αφού πολλές αφίξεις σημαίνουν και ανάλογο αριθμό αναχωρήσεων. Έτσι επιλέγουμε μια τακτική κατά την οποία εάν έχουμε συσσώρευση πάνω από έναν συγκεκριμένο αριθμό αεροπλάνων (π.χ. δύο) τα οποία περιμένουν για απογείωση και ταυτόχρονα υπάρχει αεροπλάνο για προσγείωση να δίνεται προτεραιότητα στο έδαφος. Ως ασφάλεια για αποφυγή συσσώρευσης πολλών αεροπλάνων στον αέρα θέτουμε και σε αυτά έναν περιορισμό ο οποίος όταν ξεπεραστεί (γεγονός που σημαίνει ότι ο χρόνος αναμονής για προσγείωση είναι μεγάλος) η προτεραιότητα δίνεται στον αέρα .Με τον τρόπο αυτό λοιπόν διευθετείτε ικανοποιητικά το πρόβλημα ύπαρξης ενός διαδρόμου για απογειώσεις και προσγειώσεις.

Όπως έχει ήδη αναφερθεί το κάθε στοιχειώδες σύστημα ή υποσύστημα του αεροδρομίου απαιτεί κάποιο συγκεκριμένο χρονικό διάστημα για την εξυπηρέτηση ενός αεροπλάνου. Η εξυπηρέτηση αυτή μπορεί να είναι η τροχοδρόμηση ενός αεροπλάνου σε κάποιον διάδρομο ή ακόμα και η παραμονή του στον χώρο στάθμευσης των αεροπλάνων στο αεροδρόμιο. Για το αεροδρόμιο "Κ. Δασκαλογιάννης" έχουν επιλεγεί οι χρόνοι που φαίνονται στον Πίνακα 2.2 .

Στοιχειώδες Σύστημα ή Υποσύστημα	Χρόνος λειτουργίας (min)
A1	1
A2	1
B1	2
C1	45
C2	45
C3	45
C4	45
C5	45
C6	45
D1	2

Πίνακας 2.2 Χρόνοι λειτουργίας στοιχειωδών συστημάτων, υποσυστημάτων

Δηλαδή ένα αεροπλάνο θέλει 1 λεπτό για να προσγειωθεί και άλλα 2 λεπτά για να τροχοδρομήσει ως τη θέση στάθμευσής του. Στην θέση στάθμευσης παραμένει για 45 λεπτά .Αφού ολοκληρωθούν τα 45 λεπτά της στάθμευσης χρειάζεται 2 λεπτά για να ολοκληρώσει την τροχοδρόμηση του προς την θέση της απογείωσης και άλλο ένα λεπτό για να ολοκληρώσει την απογείωσή του. Φυσικά πάντα υποθέτουμε και την ύπαρξη ουρών αναμονής, οπότε υπάρχουν και παραπάνω καθυστερήσεις στην κίνηση ενός αεροπλάνου μέσα στους χώρους ενός αεροδρομίου. Οι παραπάνω χρόνοι δεν αντιπροσωπεύουν τυχόν καθυστερήσεις λόγω έλλειψης πόρων που ως αποτέλεσμα έχουν την παραμονή των αεροπλάνων και συσσώρευση τους στους buffers εισόδου των στοιχειωδών συστημάτων ή υποσυστημάτων του αεροδρομίου.

Κεφάλαιο 3

Ανάπτυξη γλώσσας προσομοίωσης για κίνηση αεροπλάνων σε χώρους αεροδρομίου και εφαρμογή για το αεροδρόμιο «Κ. Δασκαλογιάννης»

3.1 Γενικά

Για την διεξαγωγή της προσομοίωσης της κίνησης των αεροπλάνων στους χώρους ενός αεροδρομίου επιλέχθηκε ο προγραμματισμός σε μια γλώσσα γενικής χρήσης και συγκεκριμένα αυτή του MATLAB. Η επιλογή μιας έτοιμης εξειδικευμένης γλώσσας για προσομοίωση απορρίφθηκε ως μη ευέλικτη. Με την χρήση γλώσσας γενικής χρήσης μπορούμε να αντεπεξέλθουμε σε ιδιομορφίες που παρουσιάζει το υπό μελέτη αεροδρόμιο επιλέγοντας εύκολα τους δικούς μας τρόπους λύσης και όχι κάποιους προκατασκευασμένους οι οποίοι μπορεί να μην συμπίπτουν ακριβώς με την λύση που απαιτείται.

Κατά την δημιουργία του κώδικα που αποτελεί την γλώσσα προσομοίωσης ακολουθήθηκε η μεθοδολογία του κεφαλαίου 2, δηλαδή η θεώρηση ότι το αεροδρόμιο αποτελείται από συγκεκριμένα στοιχειώδη συστήματα και υποσυστήματα. Με τον ίδιο τρόπο χωρίζουμε και τον κώδικα σε υπορουτίνες με την κάθε μία να αντιπροσωπεύει-προσομοιώνει το κάθε στοιχειώδες σύστημα, υποσύστημα του αεροδρομίου καταγράφοντας την κίνηση των αεροπλάνων σε αυτό.

Στο MATLAB οι υπορουτίνες αυτές υλοποιούνται ως χωριστές συναρτήσεις (functions) και όλες μαζί υπακούν σε μια κυρίως ρουτίνα η οποία τις καλεί με κατάλληλο τρόπο για να περιγραφεί το αεροδρόμιο συνολικά. Έχοντας υπορουτίνες που υλοποιούν κάθε δυνατό στοιχειώδες σύστημα ή υποσύστημα ενός αεροδρομίου, τότε με την κατάλληλη κλήση τους από την κυρίως ρουτίνα είναι δυνατόν να προσομοιώσουμε την κίνηση και δρομολόγηση των αεροπλάνων σε οποιοδήποτε αεροδρόμιο. Το τελικό αποτέλεσμα δηλαδή είναι η δημιουργία γλώσσας προσομοίωσης για την εξυπηρέτηση αεροπλάνων σε αεροδρόμια η οποία τελικώς έχει μεγαλύτερη αξία όταν αυτά είναι και κομβικά, όταν δηλαδή έχουμε πολύ συχνές αφίξεις αεροπλάνων. Αυτή την γλώσσα προσομοίωσης την εφαρμόσαμε για το αεροδρόμιο Χανίων «Κ. Δασκαλογιάννης» ως παράδειγμα για την κατανόησή της.

Παρακάτω εξηγούνται αναλυτικά η κυρίως ρουτίνα και οι υπορουτίνες που χρησιμοποιούνται για την προσομοίωση. Ο κώδικας της κάθε ρουτίνας ή υπορουτίνας που αναπτύχθηκε παρατίθεται στο Παράρτημα.

3.2 Κυρίως ρουτίνα airport

Η κυρίως ρουτίνα έχει την ονομασία `airport` και είναι ο σκελετός της γλώσσας προσομοίωσης. Είναι υπεύθυνη πρώτα από όλα για την δημιουργία τυχαίων αφίξεων αεροπλάνων στο αεροδρόμιο εφόσον έχουμε έλλειψη δεδομένων εισόδου στο αεροδρόμιο. Αυτές οι αφίξεις γίνονται με μια γεννήτρια τυχαίων αριθμών με κατανομή Poisson. Το γεγονός να έχουμε πάνω από μία αφίξεις την ίδια χρονική στιγμή έχει ληφθεί υπόψη. Επίσης είναι υπεύθυνη για τον μηχανισμό ροής του χρόνου ο οποίος είναι αυτός της αύξησης σταθερού διαστήματος. Το διάστημα αυτό επιλέχθηκε να έχει διάρκεια ενός λεπτού. Δηλαδή ελέγχουμε τι έχει συμβεί στο αεροδρόμιο κάθε ένα λεπτό. Αυτό είναι δυνατό αφού οι ενέργειες που λαμβάνουν χώρα στο αεροδρόμιο είναι πολλαπλάσια του ενός λεπτού. Μια επιπλέον σημαντική λειτουργία της κύριας ρουτίνας είναι η κλήση όλων των κατάλληλων υπορουτίνων ,οι οποίες αντιστοιχούν η κάθε μία σε κάποιο στοιχειώδες σύστημα ή υποσύστημα του αεροδρομίου, με την σωστή σειρά για να είναι δυνατή η υλοποίηση μιας σωστής και ολοκληρωμένης διάταξης του υπό εξέταση αεροδρομίου. Σύμφωνα με την ανάλυση του αεροδρομίου «Κ. Δασκαλογιάννης» στο κεφάλαιο 2 θα πρέπει να υπάρχει μία υπορουτίνα που να περιγράφει την κίνηση των αεροπλάνων στον διάδρομο προσγείωσης την οποία ονομάζουμε υπορουτίνα **landing** .Άλλη μια υπορουτίνα που να περιγράφει την κίνηση των αεροπλάνων στον διάδρομο τροχοδρόμησης που οδηγεί από τον διάδρομο προσγείωσης προς τις θέσεις στάθμευσης την οποία ονομάζουμε υπορουτίνα **taxiing** και μία που να περιγράφει την παραμονή των αεροπλάνων σε μία από τις έξι διαθέσιμες θέσεις για στάθμευση η οποία ονομάζεται υπορουτίνα **service**. Η κίνηση των αεροπλάνων από τις θέσεις στάθμευσης μέχρι τον διάδρομο απογείωσης περιγράφεται από την υπορουτίνα **taxiing**, η οποία είναι ίδια με πριν, ενώ η χρήση του διαδρόμου απογείωσης από τα αεροπλάνα περιγράφεται την υπορουτίνα **take_off**. Επίσης είναι αναγκαία και μία υπορουτίνα που θα ελέγχει τις προτεραιότητες που δίνονται στα αεροπλάνα που κάνουν χρήση του διαδρόμου προσγείωσης-απογείωσης η οποία ονομάζεται **priority**.

Ο κώδικας της κύριας ρουτίνας ξεκινά αρχικοποιώντας όλες τις σταθερές ή μεταβλητές που θα χρησιμοποιηθούν όπως και τον χρόνο με $t=0$. Επίσης καλεί την υπορουτίνα **Poisson** και έτσι έχουμε την δημιουργία τυχαίων αφίξεων σε τυχαίες χρονικές στιγμές. Μετά ελέγχει αν ο αριθμός των αναχωρήσεων των αεροπλάνων από το αεροδρόμιο είναι ίσος με τον αριθμό των αφίξεων που είχαμε σε αυτό. Αν είναι ίσος τότε η προσομοίωση τερματίζεται. Αν είναι μικρότερος τότε αρχίζει να καλεί τις υπορουτίνες με την κατάλληλη σειρά. Αν θα εκτελεστεί η κάθε μία από τις υπορουτίνες εξαρτάται από το αν ικανοποιούνται οι συνθήκες που έχουν οριστεί στην αρχή της κάθε μίας από αυτές. Όταν προσπελαστούν όλες οι υπορουτίνες ο χρόνος αυξάνει κατά μία μονάδα και επαναλαμβάνεται η ίδια διαδικασία. Το διάγραμμα ροής της ρουτίνας airport φαίνεται στο Σχήμα 3.1 στο τέλος του κεφαλαίου.

3.3 Υπορουτίνα priority

Η υπορουτίνα priority είναι η πρώτη υπορουτίνα που καλείται από την κυρίως ρουτίνα airport. Εκτελείται όταν ικανοποιούνται ορισμένες συνθήκες οι οποίες περιγράφουν την κατάσταση όπου έχουμε κοινό διάδρομο για απογειώσεις-προσγειώσεις, ο buffer εισόδου b5 της υπορουτίνας take_off έχει φτάσει σε κάποιο όριο το οποίο ορίζουμε στην κυρίως ρουτίνα ως b5max και ταυτόχρονα ο buffer εισόδου b1 της υπορουτίνας landing δεν έχει ξεπεράσει κάποιο όριο το οποίο ορίζουμε και αυτό στην κυρίως ρουτίνα ως b1max. Το b5max συμβολίζει ένα επιθυμητό όριο στον αριθμό των αεροπλάνων που συσσωρεύονται περιμένοντας να απογειωθούν ενώ το b1max ένα επιθυμητό όριο στον αριθμό των αεροπλάνων που βρίσκονται στον αέρα περιμένοντας να προσγειωθούν. Για παράδειγμα έχουμε επιλέξει b1max=10 και b5max=2. Αυτό σημαίνει ότι αν αριθμός των αεροπλάνων που περιμένουν για απογείωση φτάσει το δύο και εάν τα αεροπλάνα που βρίσκονται στον αέρα περιμένοντας για προσγείωση δεν ξεπερνούν τα δέκα δίνεται προτεραιότητα για χρήση του διαδρόμου προσγείωσης απογείωσης στα αεροπλάνα στο έδαφος. Σε όποια άλλη διαφορετική περίπτωση η υπορουτίνα δεν εκτελείται. Στην περίπτωση του αεροδρομίου «Κ. Δασκαλογιάννης» είναι πιθανό να εκτελεστεί σε περιόδους υπερβολικού φόρτου αν και κάτι τέτοιο τελικά είναι σπάνιο. Κρίθηκε όμως αναγκαία η υλοποίηση μιας τέτοιας υπορουτίνας για να βρίσκει εφαρμογή η συγκεκριμένη γλώσσα προσομοίωσης και σε άλλα πιο κομβικά αεροδρόμια που το

πρόβλημα της ταυτόχρονης χρήσης ενός διαδρόμου για απογείωση και προσγείωση είναι πιο συχνό. Βεβαίως είναι πιθανό να έχουμε και κάποιο αεροδρόμιο το οποίο να διαθέτει διαφορετικούς διαδρόμους για απογειώσεις και προσγειώσεις αεροπλάνων οπότε και η υπορουτίνα `priority` παραλείπεται. Φυσικά τα όρια `b1max` και `b5max` μπορούμε κάθε φορά να τα προσαρμόζουμε ανάλογα με τις απαιτήσεις μας, όπως και το γεγονός του κοινού διαδρόμου θέτοντας στην κυρίως ρουτίνα `same_runway=1`. Εάν λοιπόν ικανοποιούνται οι παραπάνω έλεγχοι, πραγματοποιείται η απογείωση ενός αεροπλάνου μειώνοντας τον `buffer` εισόδου του στοιχειώδους υποσυστήματος A2 κατά μία μονάδα. Αν υπάρχει κι άλλο αεροπλάνο που περιμένει για απογείωση (δηλαδή ο `buffer` εισόδου του A2 είναι μεγαλύτερος του μηδενός) τότε εκτιμάτε ο μελλοντικός χρόνος εξυπηρέτησης του. Αλλιώς αυξάνεται κατευθείαν ο `buffer` εξόδου του υποσυστήματος A2 που είναι ο `b6` κατά μία μονάδα και έχουμε και την ακύρωση της προγραμματισμένης προσγείωσης, θέτοντας σε αυτήν νέο χρόνο ολοκλήρωσης της. Τότε τερματίζεται η υπορουτίνα για τη συγκεκριμένη χρονική στιγμή για την οποία την ελέγχουμε και περνάμε στην επόμενη υπορουτίνα ξανά για την ίδια χρονική στιγμή. Το διάγραμμα ροής της υπορουτίνας `priority` φαίνεται στο Σχήμα 3.2 στο τέλος του κεφαλαίου.

3.4 Υπορουτίνα landing

Η υπορουτίνα `landing` είναι η δεύτερη υπορουτίνα που καλείται από την κυρίως ρουτίνα `airport`. Αυτή αφορά την διαδικασία προσγείωσης των αεροπλάνων στο αεροδρόμιο δηλαδή το στοιχειώδες υποσύστημα A1. Ο `buffer` εισόδου `b1` της υπορουτίνας `landing` ενημερώνεται για τις αφίξεις αεροπλάνων από την υπορουτίνα `Poisson`. Έτσι στην υπορουτίνα `landing` αρχικά ελέγχουμε αν υπάρχει την συγκεκριμένη χρονική στιγμή που εξετάζουμε κάποια άφιξη. Είναι δυνατόν να έχουμε και πάνω από μία αφίξεις την ίδια χρονική στιγμή. Αν έχουμε άφιξη, αυξάνουμε την τιμή του `b1` ανάλογα με τον αριθμό των αεροπλάνων που καταφθάνουν, και αν επιπλέον ο `buffer` εισόδου της `landing` είναι άδειος, θέτουμε ταυτόχρονα και τον μελλοντικό χρόνο αναχώρησης του από το στοιχειώδες υποσύστημα A1 που συμβολίζει τον διάδρομο προσγείωσης. Αυτός ο χρόνος είναι άμεσα συνδεδεμένος με τον χρόνο λειτουργίας του A1. Έτσι αν υπάρχει άφιξη την χρονική στιγμή t και ο χρόνος λειτουργίας του A1 είναι όπως προαναφέρθηκε ένα

λεπτό, τότε ο χρόνος αναχώρησης θα είναι $t+1$. Όταν έχουμε άφιξη και ο buffer εισόδου δεν είναι άδειος απλώς αυξάνουμε την τιμή του ανάλογα με τον αριθμό των αεροπλάνων που καταφθάνουν. Ένας άλλος έλεγχος που πρέπει να γίνει είναι αν για την χρονική στιγμή που εξετάζεται λαμβάνει χώρα η αναχώρηση κάποιου αεροπλάνου από το υποσύστημα A1 δηλαδή αν ολοκληρώνεται η προσγείωσή του. Αν συμβαίνει κάτι τέτοιο ο buffer εισόδου του landing μειώνεται κατά μία μονάδα και αν ακόμα περιέχει κι άλλο αεροπλάνο, δηλαδή δεν μηδενιστεί, θέτεται ο νέος μελλοντικός χρόνος αναχώρησης που συμβολίζει το τέλος της εξυπηρέτησης του επόμενου αεροπλάνου που βρίσκεται σε ουρά στον buffer b1. Τέλος αφού έχουμε αναχώρηση κάποιου αεροπλάνου από το στοιχειώδες υποσύστημα A1 αυξάνουμε τον buffer εξόδου του b2 κατά μία μονάδα. Αφού ολοκληρωθούν οι παραπάνω έλεγχοι, τερματίζεται η υπορουτίνα landing για τη συγκεκριμένη χρονική στιγμή για την οποία την ελέγχουμε και περνάμε στην επόμενη υπορουτίνα ξανά για την ίδια χρονική στιγμή. Το διάγραμμα ροής της υπορουτίνας landing φαίνεται στο Σχήμα 3.3 στο τέλος του κεφαλαίου.

3.5 Υπορουτίνα taxiing

Η υπορουτίνα taxiing είναι η τρίτη υπορουτίνα που καλείται από την κυρίως ρουτίνα airport. Αυτή αφορά την διαδικασία κατά την οποία ένα αεροπλάνο χρησιμοποιεί τον διάδρομο τροχοδρόμησης (taxiing in) από τον διάδρομο προσγείωσης-απογείωσης προς τις θέσεις στάθμευσης.

Και εδώ έχουμε έναν buffer εισόδου ο οποίος μάλιστα συμπίπτει με τον buffer εξόδου της προηγούμενης υπορουτίνας landing και είναι ο b2 και έναν buffer εξόδου τον b3. Εάν την χρονική στιγμή που ελέγχουμε έχουμε άφιξη αεροπλάνου στον buffer εισόδου της taxiing, σήμα που δίνεται όταν έχουμε αναχώρηση αεροπλάνου από το στοιχειώδες υποσύστημα A2, αυξάνουμε την τιμή του κατά μία μονάδα και αν δεν βρισκόταν ήδη άλλο αεροπλάνο σε αυτόν τότε θέτουμε και τον μελλοντικό χρόνο αναχώρησης του από το συγκεκριμένο στοιχειώδες σύστημα B1. Επιπλέον ελέγχουμε αν την συγκεκριμένη χρονική στιγμή ολοκληρώνεται η εξυπηρέτηση κάποιου αεροπλάνου στο B1. Αν ναι τότε μειώνουμε τον buffer εισόδου b2 και αυξάνουμε τον buffer εξόδου b3 της taxiing κατά μία μονάδα. Αν κατά την μείωση του b2 αυτός δεν μηδενιστεί σημαίνει ότι υπάρχει τουλάχιστον ακόμα ένα

αεροπλάνο που περιμένει για να εξυπηρετηθεί από τον διάδρομο taxiing in .Τότε το θέτουμε υπό εξυπηρέτηση ορίζοντας και τον χρόνο που αυτή θα ολοκληρωθεί . Αφού ολοκληρωθούν οι παραπάνω έλεγχοι, τερματίζεται η υπορουτίνα taxiing για τη συγκεκριμένη χρονική στιγμή για την οποία την ελέγχουμε και περνάμε στην επόμενη υπορουτίνα ξανά για την ίδια χρονική στιγμή. Το διάγραμμα ροής της υπορουτίνας taxiing φαίνεται στο Σχήμα 3.4 στο τέλος του κεφαλαίου.

3.6 Υπορουτίνα service

Η υπορουτίνα service είναι η τρίτη υπορουτίνα που καλείται από την κυρίως ρουτίνα airport. Αυτή αφορά την διαδικασία κατά την οποία ένα αεροπλάνο σταθμεύει σε μία από τις έξι διαθέσιμες θέσεις στάθμευσης του αεροδρομίου «Κ. Δασκαλογιάννης». Εδώ ο buffer εισόδου είναι ο b3 και εξόδου ο b4 και για τις έξι θέσεις. Αρχικά ελέγχουμε τον b3 για το αν έχουμε άφιξη. Αν ο έλεγχος είναι αληθής τότε ελέγχουμε αν υπάρχει ελεύθερη θέση στάθμευσης. Αυτό είναι δυνατό με την χρήση ενός δείκτη που δείχνει κάθε φορά στην επόμενη άδεια θέση ή σε αυτή που θα απελευθερωθεί συντομότερα από όλες τις άλλες. Για κάθε θέση στάθμευσης κρατάμε την πληροφορία που έχει να κάνει με τον μελλοντικό χρόνο απελευθέρωσής της αν αυτή χρησιμοποιείται. Αν δεν χρησιμοποιείται ο χρόνος αυτός έχει μηδενική τιμή. Έτσι αν υπάρχει άδεια θέση την καταλαμβάνει το αεροπλάνο που έφτασε και αντιστοιχούμε σε αυτή τον εκτιμώμενο μελλοντικό χρόνο απελευθέρωσής της. Μετά μετακινούμε τον δείκτη στην επόμενη θέση και αυξάνουμε έναν μετρητή (in_service) που αποθηκεύει τον αριθμό των αεροπλάνων που βρίσκονται στις θέσεις στάθμευσης του αεροδρομίου. Αν υπάρχει μόνο ένα αεροπλάνο στις θέσεις στάθμευσης ορίζουμε τον μελλοντικό χρόνο ολοκλήρωσης της εξυπηρέτησής του σαν την επόμενη χρονική στιγμή που θα εκτελεστεί αναχώρηση αεροπλάνου από τις θέσεις αυτές. Στην περίπτωση που δεν υπάρχει ελεύθερη θέση ,αυξάνουμε τον αριθμό των αεροπλάνων που περιμένουν κατά μία μονάδα. Επίσης έχουμε δεν έχουμε κάποια ελεύθερη θέση αυξάνουμε και τον buffer εισόδου b3 της service που καταμετρά τις αφίξεις. Ένας άλλος βασικός έλεγχος που υλοποιείται στην υπορουτίνα service είναι αυτός του αν έχουμε ολοκλήρωση της στάθμευσης ενός αεροπλάνου την συγκεκριμένη χρονική στιγμή του ελέγχου. Αν κάτι τέτοιο είναι αληθές αρχικά απελευθερώνεται η συγκεκριμένη θέση στάθμευσης και με έναν δείκτη ορίζουμε την επόμενη θέση που πρόκειται να απελευθερωθεί και εξισώνουμε τον χρόνο απελευθέρωσής της με τον

χρόνο ελέγχου για την επόμενη αναχώρηση από τις θέσεις στάθμευσης. Στην περίπτωση αυτή αν δεν υπάρχει αεροπλάνο που περιμένει για εξυπηρέτηση απλώς μειώνουμε τον αριθμό των αεροπλάνων που βρίσκονται στις θέσεις στάθμευσης (in_service) κατά μία μονάδα, αλλιώς αν υπάρχει μπαίνει σε θέση στάθμευσης που ορίζεται από τον δείκτη που δείχνει κάθε φορά στην επόμενη άδεια θέση και ταυτόχρονα ο δείκτης αυτός προχωρά σε επόμενη θέση που είναι άδεια ή πρόκειται να αδειάσει συντομότερα από όλες τις άλλες θέσεις. Στην θέση που μόλις καταλήφθηκε αντιστοιχούμε τον εκτιμώμενο μελλοντικό χρόνο απελευθέρωσής της και μειώνουμε τον αριθμό των αεροπλάνων που περιμένουν εξυπηρέτηση. Αν λοιπόν έχουμε αναχώρηση αεροπλάνου από τις θέσεις στάθμευσης το επόμενο βήμα είναι να μειώσουμε τον buffer εισόδου b3 μία μονάδα και να αυξήσουμε τον buffer εξόδου b4 άλλη μία. Αφού ολοκληρωθούν οι παραπάνω έλεγχοι, τερματίζεται η υπορουτίνα service για τη συγκεκριμένη χρονική στιγμή για την οποία την ελέγχουμε και περνάμε στην επόμενη υπορουτίνα ξανά για την ίδια χρονική στιγμή. Το διάγραμμα ροής της υπορουτίνας service φαίνεται στο Σχήμα 3.5 στο τέλος του κεφαλαίου.

3.7 Υπορουτίνα taxiing

Η υπορουτίνα taxiing είναι η τέταρτη υπορουτίνα που καλείται από την κυρίως ρουτίνα airport, ενώ είναι η δεύτερη φορά που την συναντάμε. Τώρα αφορά την διαδικασία κατά την οποία ένα αεροπλάνο χρησιμοποιεί τον διάδρομο τροχοδρόμησης (taxiing out) από την θέση στάθμευσης που βρισκόταν προς τον διάδρομο προσγείωσης-απογείωσης. Αντιστοιχεί δηλαδή στο στοιχειώδες υποσύστημα D1 . Ενώ οι έλεγχοι που πραγματοποιούνται είναι ίδιοι με πριν αφού έχουμε την ίδια υπορουτίνα αυτό που αλλάζει είναι ο buffer εισόδου, τώρα είναι ο b4, και ο buffer εξόδου που τώρα είναι ο b5. Στον b4 βρίσκονται τα αεροπλάνα που μόλις αποχωρούν από τις θέσεις στάθμευσης ενώ στον b5 αυτά που προορίζονται για απογείωση . Αφού ολοκληρωθούν οι ανάλογοι έλεγχοι , τερματίζεται η υπορουτίνα taxiing για τη συγκεκριμένη χρονική στιγμή για την οποία την ελέγχουμε και περνάμε στην επόμενη υπορουτίνα ξανά για την ίδια χρονική στιγμή. Το διάγραμμα ροής της υπορουτίνας taxiing φαίνεται στο Σχήμα 3.4 στο τέλος του κεφαλαίου.

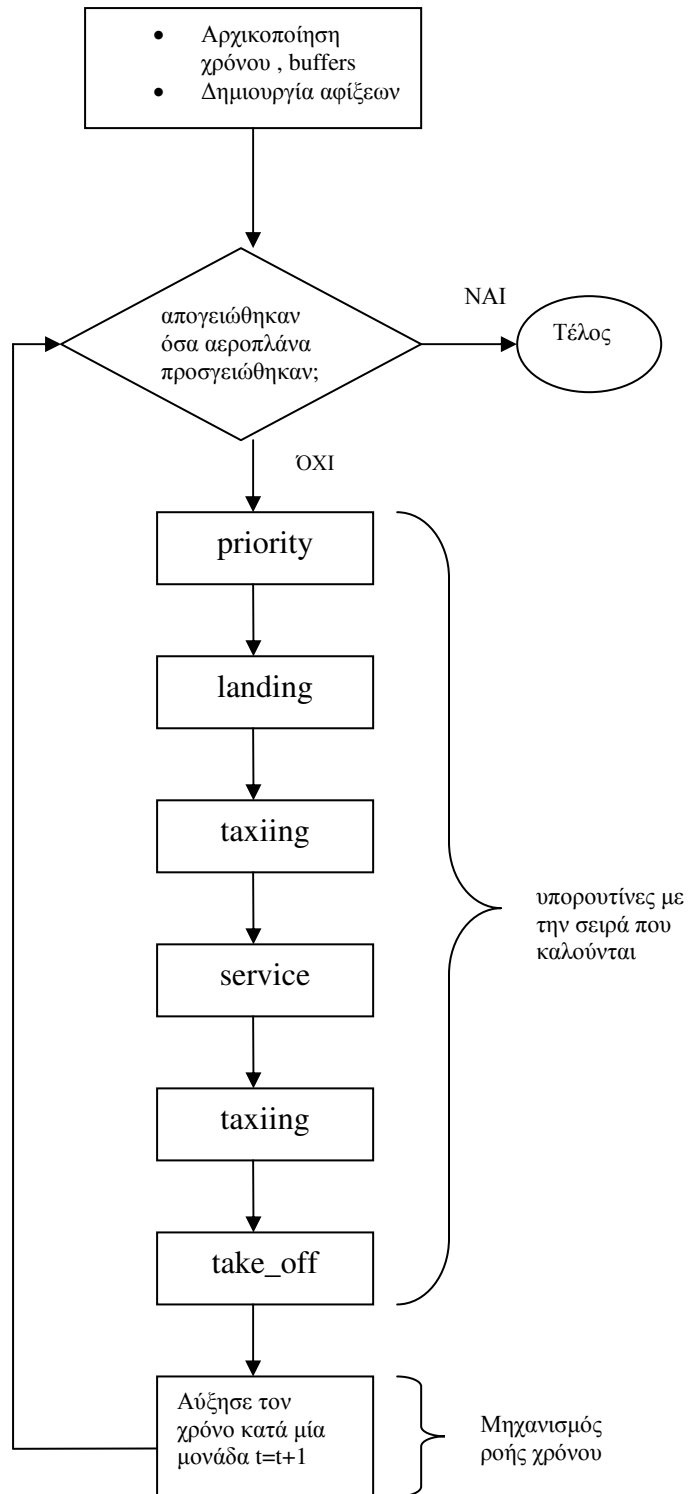
3.8 Υπορουτίνα take_off

Η υπορουτίνα take_off είναι η πέμπτη υπορουτίνα που καλείται από την κυρίως ρουτίνα airport. Αυτή αφορά την διαδικασία απογείωσης των αεροπλάνων από το αεροδρόμιο δηλαδή το στοιχειώδες υποσύστημα A2 με buffer εισόδου τον b5 και buffer εξόδου τον b6. Έτσι όταν έχουμε άφιξη ενός αεροπλάνου στον b5 αυξάνουμε την τιμή του κατά μία μονάδα και στην περίπτωση που αυτός είναι άδειος θέτουμε αμέσως για το αφικνούμενο αεροπλάνο τον εκτιμώμενο μελλοντικό χρόνο ολοκλήρωσης της εξυπηρέτησης του που είναι η απογείωσή του. Αν την συγκεκριμένη χρονική στιγμή ελέγχου υπάρχει κάποια προγραμματισμένη αναχώρηση από το υποσύστημα A2 τότε αφού υπάρχει ένας μόνο διάδρομος για απογειώσεις και προσγειώσεις (same_runway=1) ελέγχουμε αν αυτός είναι ελεύθερος από τις προσγειώσεις. Δεν θα έχουμε προσγειώσεις αν ο buffer εισόδου της landing είναι άδειος δηλαδή b1=0. Αν συμβαίνει αυτό προχωράμε κανονικά στην απογείωση του αεροπλάνου που έχει προγραμματιστεί από κάποια προηγούμενη χρονική στιγμή ,μειώνοντας κατά μία μονάδα τον buffer εισόδου της take_off και αυξάνοντας τον buffer εξόδου της άλλη μία μονάδα. Στην περίπτωση που buffer εισόδου της take_off δεν μηδενιστεί θέτουμε μελλοντικό χρόνο εξυπηρέτησης για το επόμενο αεροπλάνο στην ουρά του b5. Αν τώρα ο buffer εισόδου της landing b1 δεν είναι άδειος αυτό σημαίνει ότι ο διάδρομος προσγείωσης-απογείωσης χρησιμοποιείται για προσγείωση και έτσι η προγραμματισμένη απογείωση ακυρώνεται ,θέτοντας νέα καθυστέρηση για την εξυπηρέτηση του αεροπλάνου που αφορά την απογείωσή του. Αφού ολοκληρωθούν οι παραπάνω έλεγχοι, τερματίζεται η υπορουτίνα take_off για τη συγκεκριμένη χρονική στιγμή για την οποία την ελέγχουμε και αφού αποτελεί και την τελευταία υπορουτίνα της κυρίως ρουτίνας για τους ελέγχους που πραγματοποιούνται σε κάθε χρονική στιγμή, προχωράμε στην επόμενη χρονική στιγμή και όλα τα παραπάνω επαναλαμβάνονται όπως ακριβώς υπαγορεύει η ρουτίνα airport. Το διάγραμμα ροής της υπορουτίνας take_off φαίνεται στο Σχήμα 3.6 στο τέλος του κεφαλαίου.

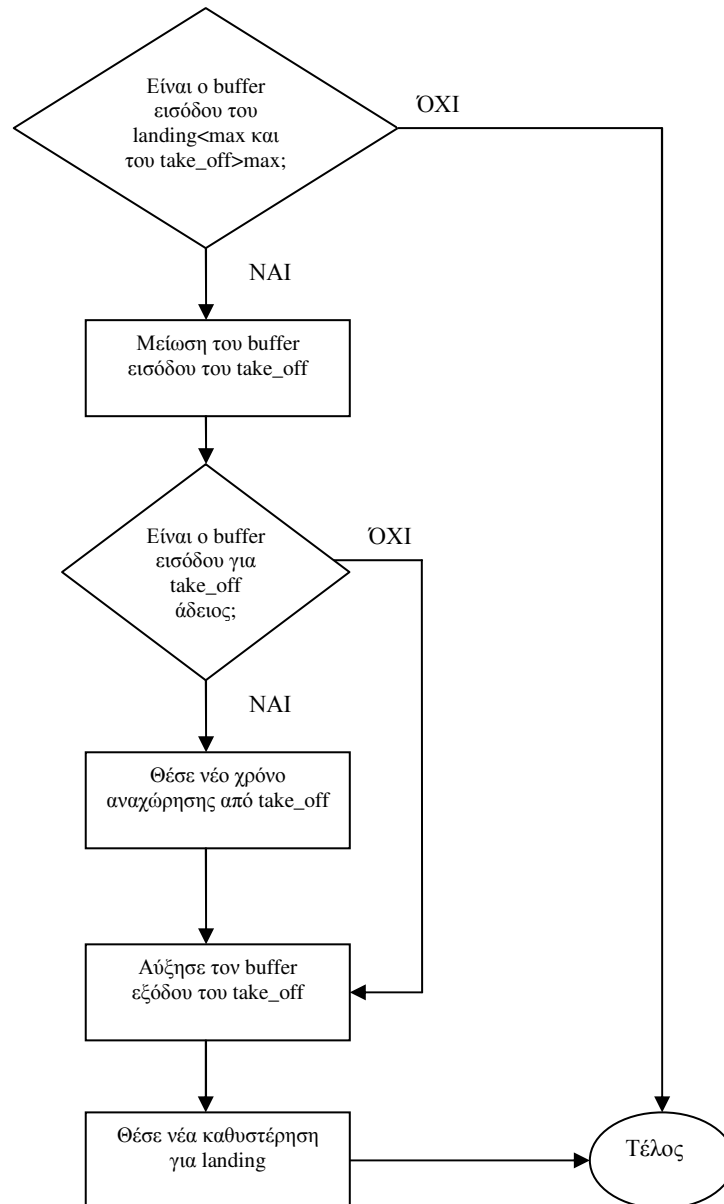
3.9 Περαιτέρω Ανάλυση

Με δεδομένα όλα τα παραπάνω ,δηλαδή τις υπορουτίνες landing, taxiing, priority, service, take_off και μιας κυρίως ρουτίνας που τις ελέγχει, καλύπτουμε τις κύριες κινήσεις αεροπλάνων μέσα σε οποιοδήποτε αεροδρόμιο. Εάν η κυρίως ρουτίνα έχει όπως η παραπάνω τότε η περιγραφή ανήκει στο αεροδρόμιο Χανίων. Φυσικά για κάθε διαφορετικό αεροδρόμιο θα πρέπει να γίνονται και συγκεκριμένες αλλαγές στην κυρίως ρουτίνα.

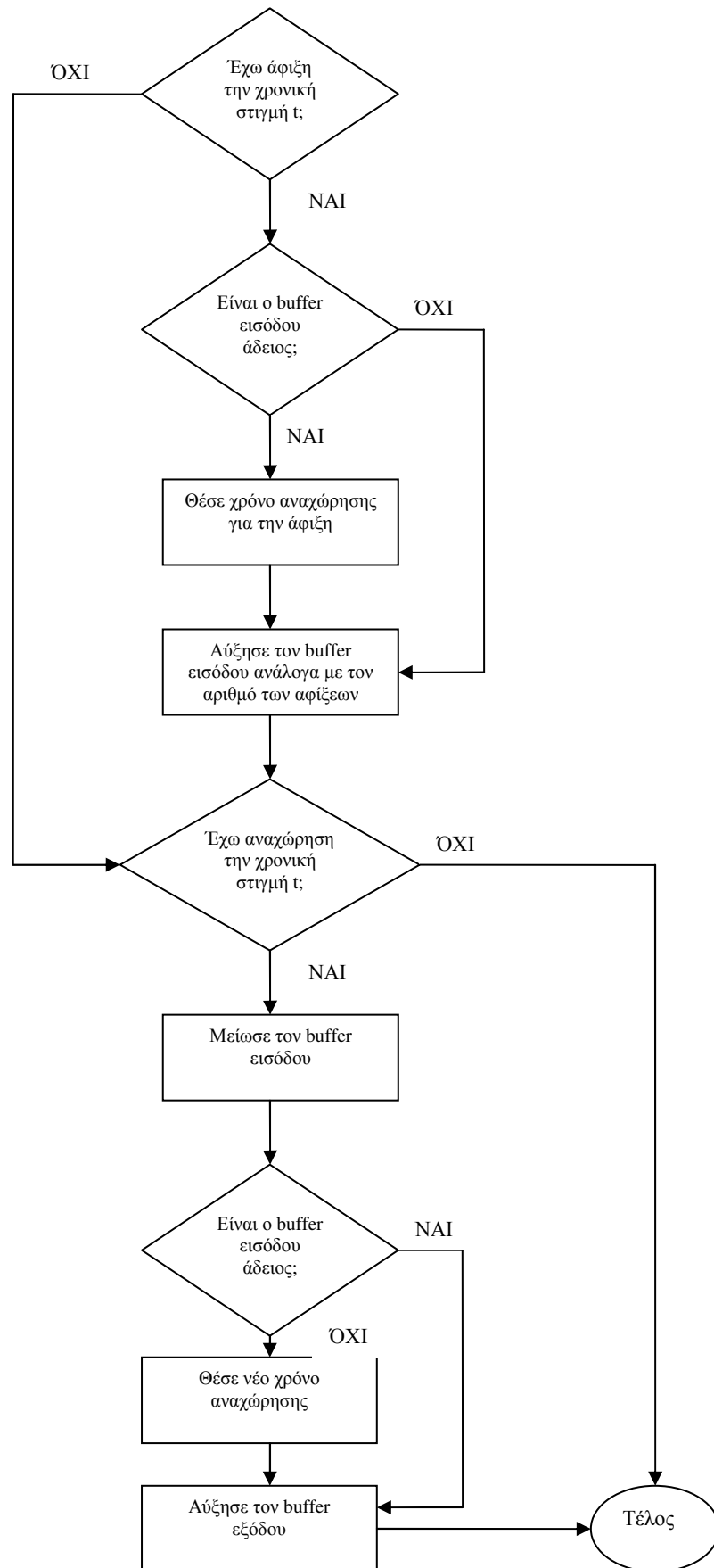
Για παράδειγμα αν μας απασχολεί ένα αεροδρόμιο το οποίο διαθέτει δύο είδη θέσεων στάθμευσης των αεροπλάνων όπως με φυσούνες ή χωρίς αρκεί να εισάγουμε δύο υπορουτίνες service στην κυρίως ρουτίνα ,με την μία υπορουτίνα να αντιπροσωπεύει της φυσούνες και η άλλη τις υπαίθριες θέσεις. Επιπλέον θα πρέπει στην περίπτωση αυτή αντί για μία υπορουτίνα taxiing(που οδηγεί τα αεροπλάνα από τον διάδρομο προσγείωσης στις θέσεις στάθμευσης) να έχουμε δύο με την κάθε μία να οδηγεί τα αεροπλάνα είτε στις θέσεις με φυσούνες είτε στις υπαίθριες. Ένα άλλο παράδειγμα αφορά ένα αεροδρόμιο όπου οι απογειώσεις και οι προσγειώσεις εξυπηρετούνται από διαφορετικούς διαδρόμους οπότε η χρήση της υπορουτίνας priority μπορεί να παραλειφθεί .



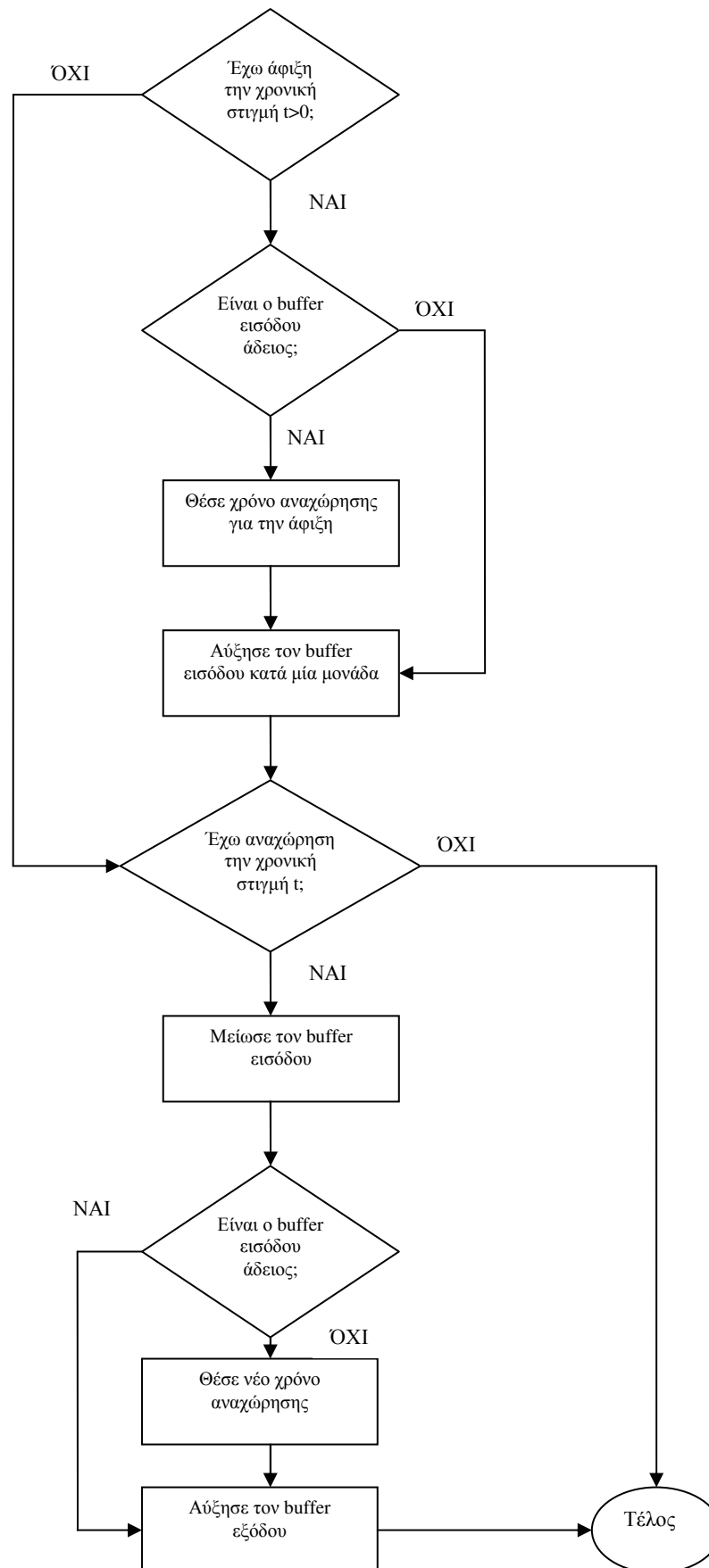
Σχήμα 3.1 Κυρίως ρουτίνα “airport”



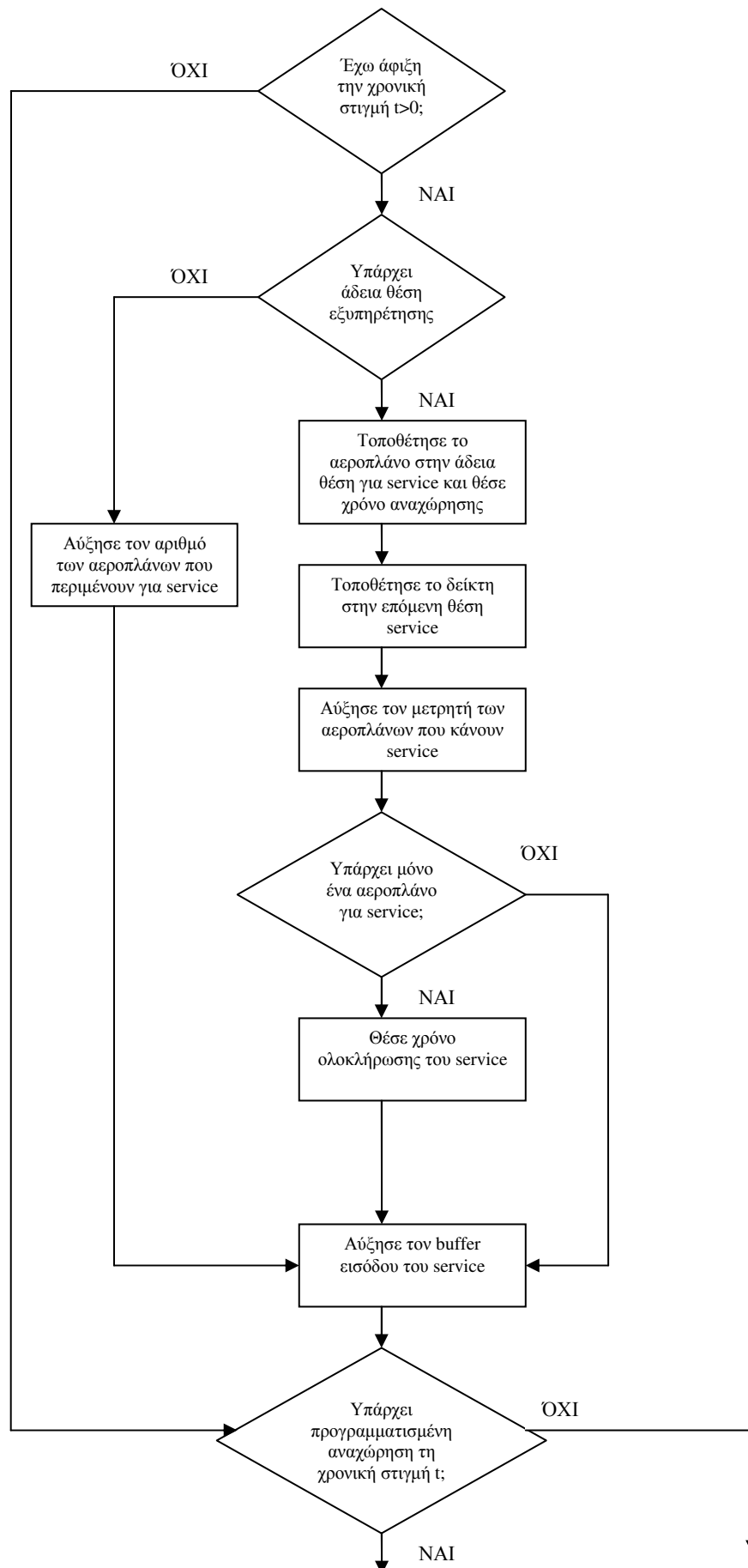
Σχήμα 3.2 Υπορουτίνα “priority”

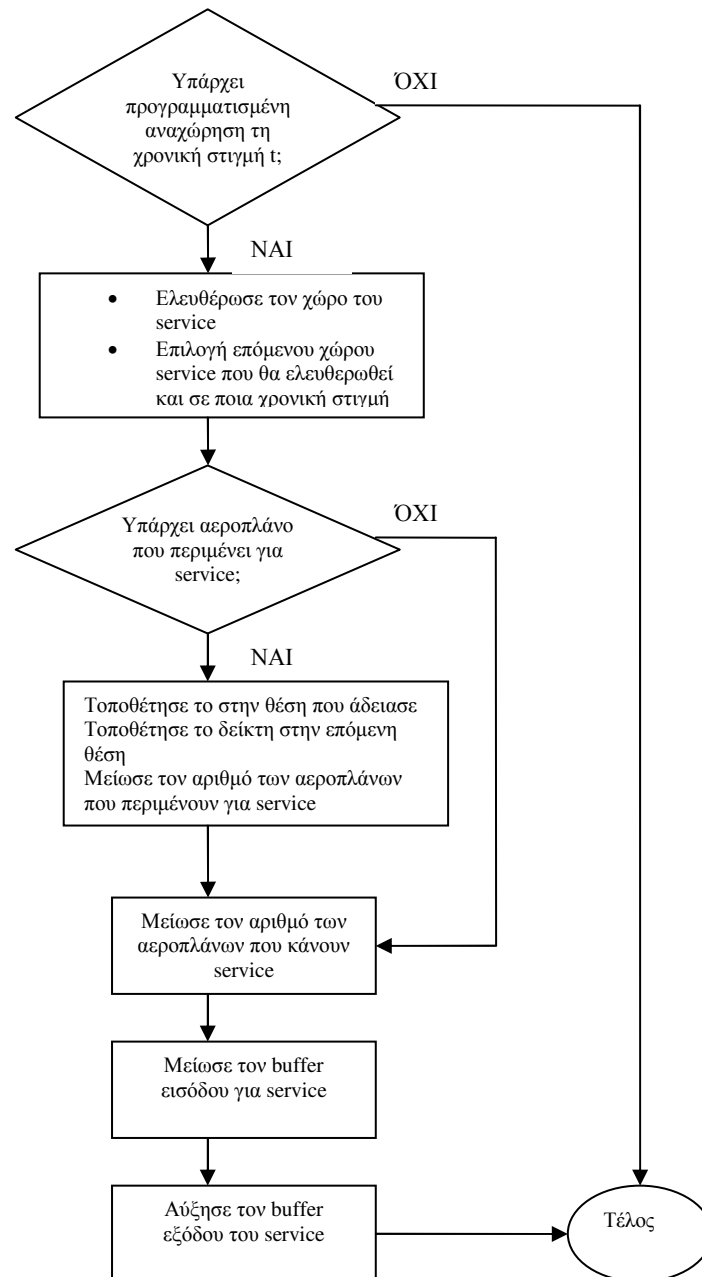


Σχήμα 3.3 Υπορουτίνα “landing”

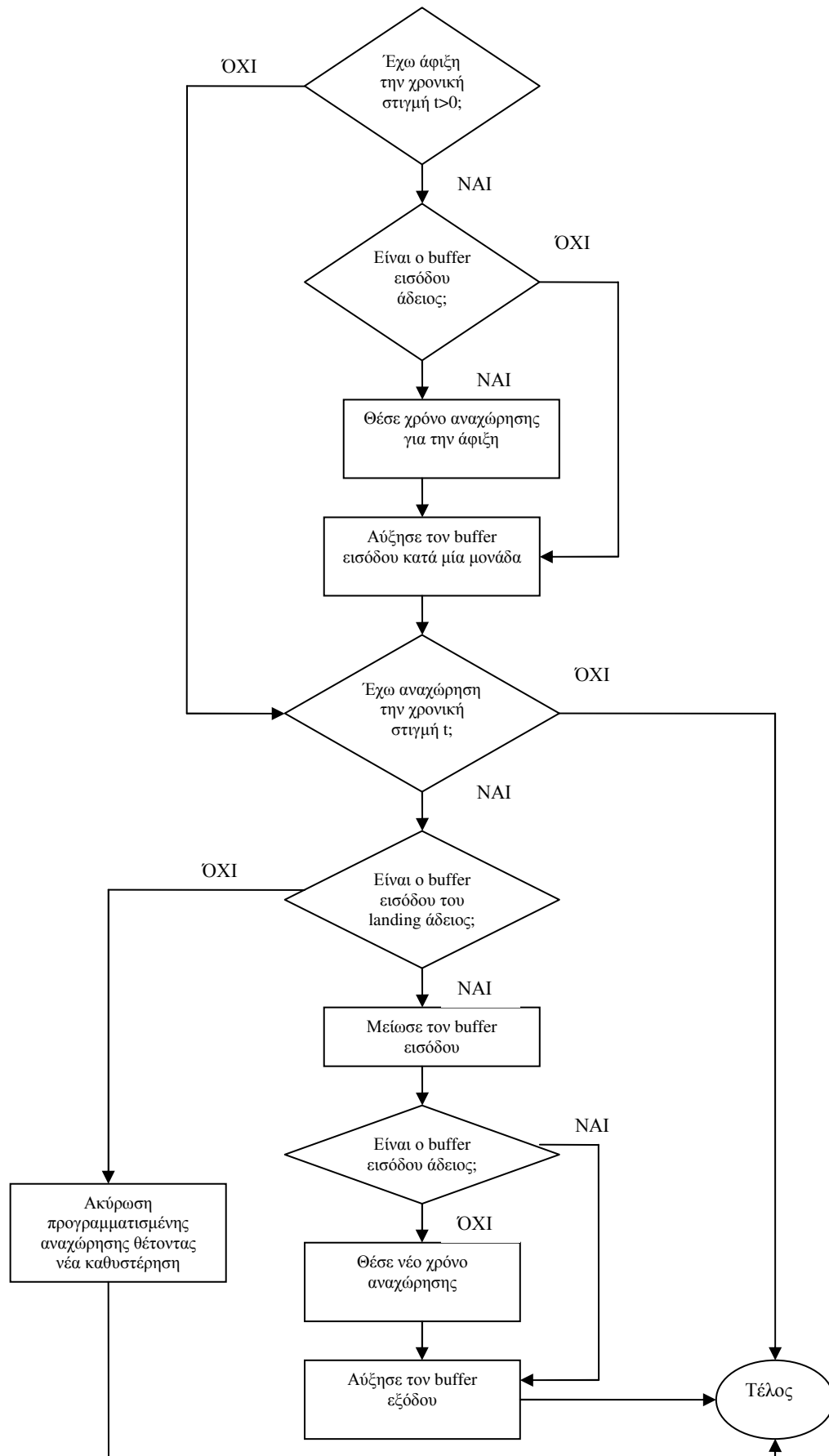


Σχήμα 3.4 Υπορουτίνα "taxiing"





Σχήμα 3.5 Υπορουτίνα "service"



Σχήμα 3.6 Υπορουτίνα “take_off

Κεφάλαιο 4

Εφαρμογή Προσομοίωσης - Παραδείγματα

4.1 Γενικά

Στο κεφάλαιο αυτό γίνεται επίδειξη της γλώσσας προσομοίωσης για εξυπηρέτηση αεροπλάνων στο αεροδρόμιο Χανίων «Κ. Δασκαλογιάννης». Τα αποτελέσματα που λαμβάνουμε είναι σε μορφή χρονοδιαγραμμάτων τα οποία μπορούμε να τα διαχωρίσουμε σε δύο τύπους.

Ο ένας τύπος καλύπτει την περιγραφή της κατάστασης του κάθε buffer τον οποίο έχουμε ορίσει μέσα στο σύστημα του αεροδρομίου. Όπως έχει προαναφερθεί ,χαρακτηριστικό των buffers είναι ο αριθμός των αεροπλάνων που περιέχει ο καθένας τους. Ο αριθμός αυτός είναι ίσος με το πλήθος των αεροπλάνων που περιμένουν για εξυπηρέτηση από κάποιο σύστημα ή υποσύστημα του αεροδρομίου συν το αεροπλάνο που εξυπηρετείται. Τα χρονοδιαγράμματα αυτού του τύπου παρουσιάζονται υπό μορφή σκάλας λόγω του ότι ασχολούμαστε με ένα διακριτό σύστημα. Στην μορφή σκάλας κάθε σκαλοπάτι που προστίθεται μεταφράζεται σε αύξηση κατά ένα των αεροπλάνων που βρίσκονται σε έναν συγκεκριμένο buffer, ενώ κάθε σκαλοπάτι που μειώνεται μεταφράζεται σε μείωση κατά ένα των αεροπλάνων που βρίσκονται στον buffer αυτόν. Οι μεταβολές στον αριθμό των σκαλοπατιών περιγράφονται στον κατακόρυφο άξονα των χρονοδιαγραμμάτων ενώ ο οριζόντιος άξονας αφορά την πάροδο του χρόνου.

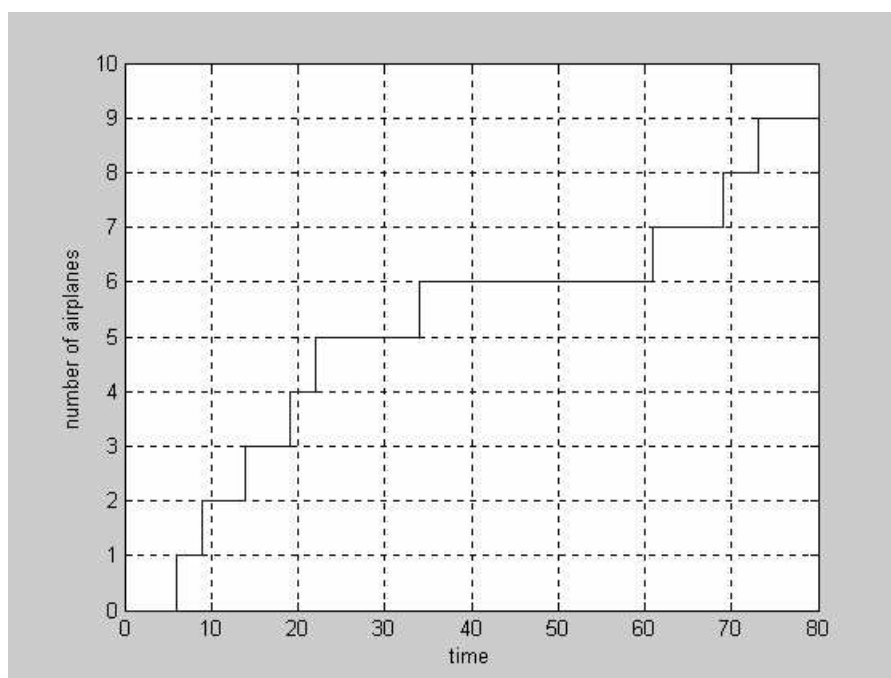
Ο δεύτερος τύπος που διαχωρίζουμε τα χρονοδιαγράμματα καλύπτει την κατάσταση του κάθε στοιχειώδους συστήματος ή υποσυστήματος που αποτελούν το αεροδρόμιο, δηλαδή αν το καθένα από αυτά χρησιμοποιείται (κατειλημμένο) ή δεν χρησιμοποιείται (ελεύθερο) από κάποιο αεροπλάνο. Τα χρονοδιαγράμματα αυτού του τύπου παρουσιάζονται με την μεταβολή της κατάστασης μεταξύ δύο τιμών που η μία δηλώνει χρήση και είναι ο άσος και η άλλη δηλώνει μη χρήση του στοιχειώδους συστήματος ή υποσυστήματος και είναι το μηδέν. Οι μεταβολές της κατάστασης μεταξύ άσσου και μηδέν περιγράφονται από τον κατακόρυφο άξονα των χρονοδιαγραμμάτων ενώ ο οριζόντιος άξονας αφορά την πάροδο του χρόνου.

Με την συγκέντρωση όλων αυτών των πληροφοριών ,μέσω των χρονοδιαγραμμάτων, μπορούμε να παρατηρούμε τις κινήσεις των αεροπλάνων μέσα στους χώρους του αεροδρομίου με γραφικό τρόπο. Όπως έχει προαναφερθεί, λόγο

έλλειψης πραγματικών χρόνων αφίξεων των αεροπλάνων για το αεροδρόμιο που ασχολούμαστε, γίνεται χρήση μιας γεννήτριας τυχαίων αριθμών για την δημιουργία αυτών των αφίξεων. Αυτό μας δίνει το πλεονέκτημα ότι μπορούμε να ελέγχουμε την συμπεριφορά του αεροδρομίου για διαφορετικά σενάρια αφίξεων των αεροπλάνων σε αυτό με απλό και γρήγορο τρόπο. Έτσι κάθε φορά που προσομοιώνουμε το αεροδρόμιο επιλέγουμε έναν επιθυμητό αριθμό αεροπλάνων που θέλουμε να φτάσουν σε αυτό και ένα επιθυμητό χρονικό διάστημα μέσα στο οποίο θα γίνουν αυτές οι αφίξεις. Μετά κατά την προσομοίωση μπορούμε να επιβλέψουμε την λειτουργία του αεροδρομίου από την προσγείωση ως και την απογείωση όλων αυτών των αεροπλάνων. Παρακάτω παρατίθενται διάφορα χαρακτηριστικά παραδείγματα.

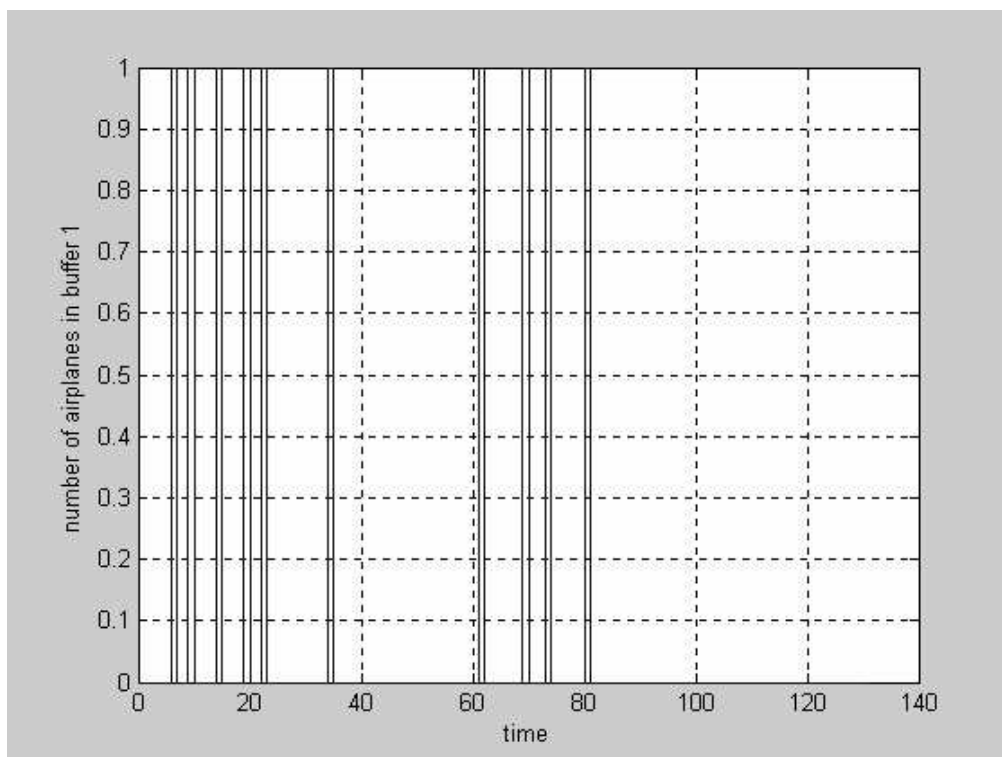
4.2 Πρώτο Παράδειγμα □ Ρυθμός Αφίξεων 10/100

Το πρώτο παράδειγμα ασχολείται με περιορισμένο αριθμό αφίξεων σε κάποιο χρονικό διάστημα για να γίνουν πιο κατανοητά τα χρονοδιαγράμματα κατά την πρώτη αυτή προσέγγιση. Θεωρούμε την περίπτωση όπου έχουμε την άφιξη **10 αεροπλάνων** στο αεροδρόμιο μέσα σε ένα επιθυμητό χρονικό διάστημα που είναι τα **100 λεπτά** της ώρας. Οι αφίξεις ακολουθούν τυχαία κατανομή Poisson με $\lambda=10/100$. Όπως φαίνεται και στο χρονοδιάγραμμα 4.1.1 οι 10 αφίξεις λαμβάνουν χώρα τις χρονικές στιγμές 6, 9, 14, 19, 22, 34, 61, 69, 73 και 80 αντίστοιχα και οι οποίες καλύπτουν το επιθυμητό χρονικό διάστημα των 100 λεπτών.

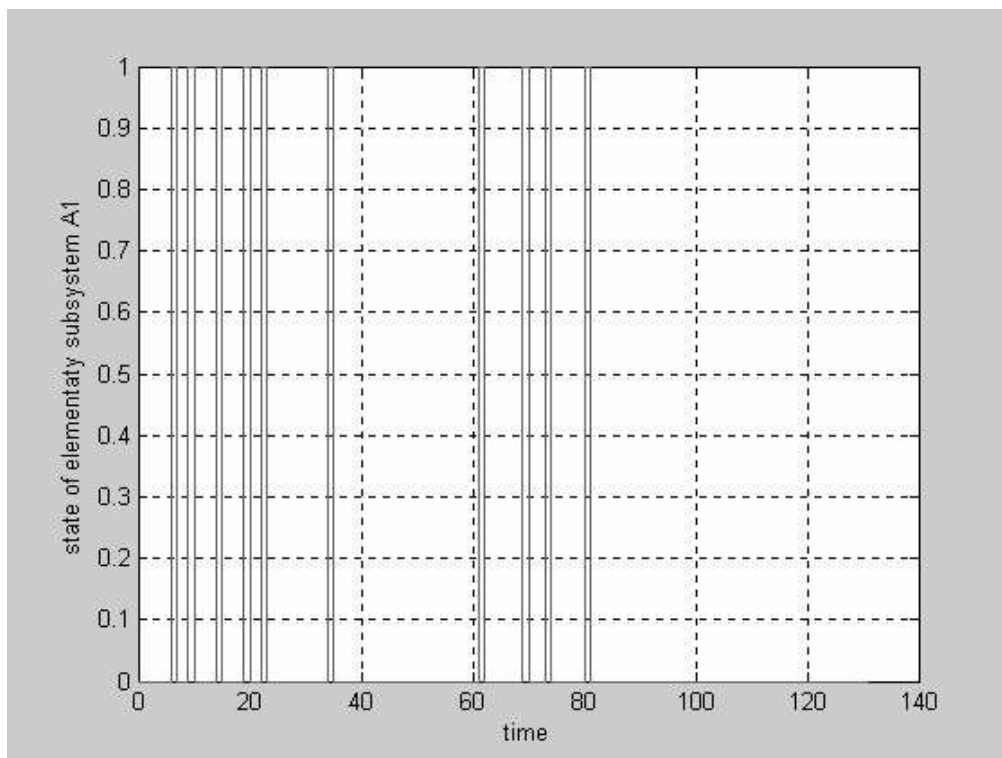


χρονοδιάγραμμα 4.1.1

Τα αεροπλάνα που καταφθάνουν σύμφωνα με το χρονοδιάγραμμα 4.1.1 εισέρχονται αμέσως στον buffer 1 όπως δείχνει το χρονοδιάγραμμα 4.1.2, τον οποίο και αυξάνουν ανάλογα. Παρατηρούμε στο 4.1.2 ότι μόλις έχουμε άφιξη ενός αεροπλάνου, προστίθεται ένα σκαλοπάτι σε αυτό και μετά από κάποιο χρονικό διάστημα που ταυτίζεται με τον χρόνο λειτουργίας του στοιχειώδους υποσυστήματος A1(διάδρομος προσγείωσης landing) που είναι ένα λεπτό αφαιρείται ένα σκαλοπάτι. Η αφαίρεση του σκαλοπατιού σηματοδοτεί και το τέλος της εξυπηρέτησης του αεροπλάνου από το A1. Παρατηρούμε ότι στον buffer 1 για την περίπτωση αυτή δεν έχουμε συσσώρευση δύο ή περισσότερων αεροπλάνων αφού έχει ως ανώτερη τιμή το ένα. Όσα αεροπλάνα εισέρχονται σε αυτόν εξυπηρετούνται αμέσως από το A1. Στο χρονοδιάγραμμα 4.1.3 παρουσιάζονται τα χρονικά διαστήματα κατά τα οποία πραγματοποιούνται οι προσγειώσεις τα οποία αντιστοιχούν με τις χρονικές στιγμές κατά τις οποίες το χρονοδιάγραμμα λαμβάνει τιμή 1. Είναι ίδιο με το 4.1.2 επειδή δεν έχουμε συσσώρευση αεροπλάνων στον buffer 1.



χρονοδιάγραμμα 4.1.2

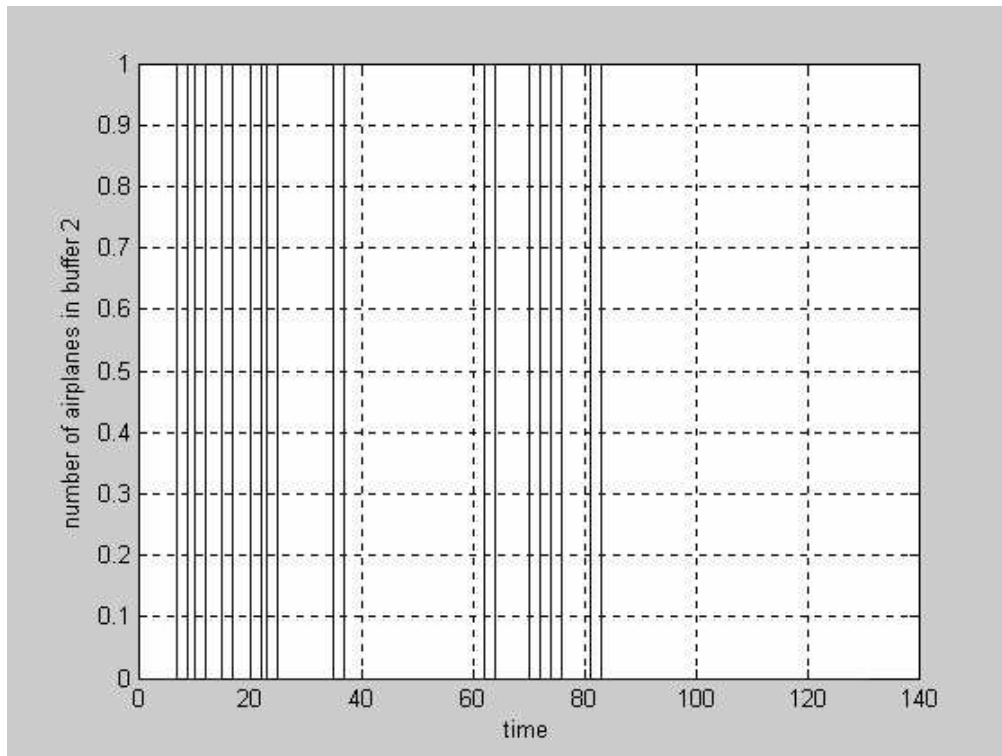


χρονοδιάγραμμα 4.1.3

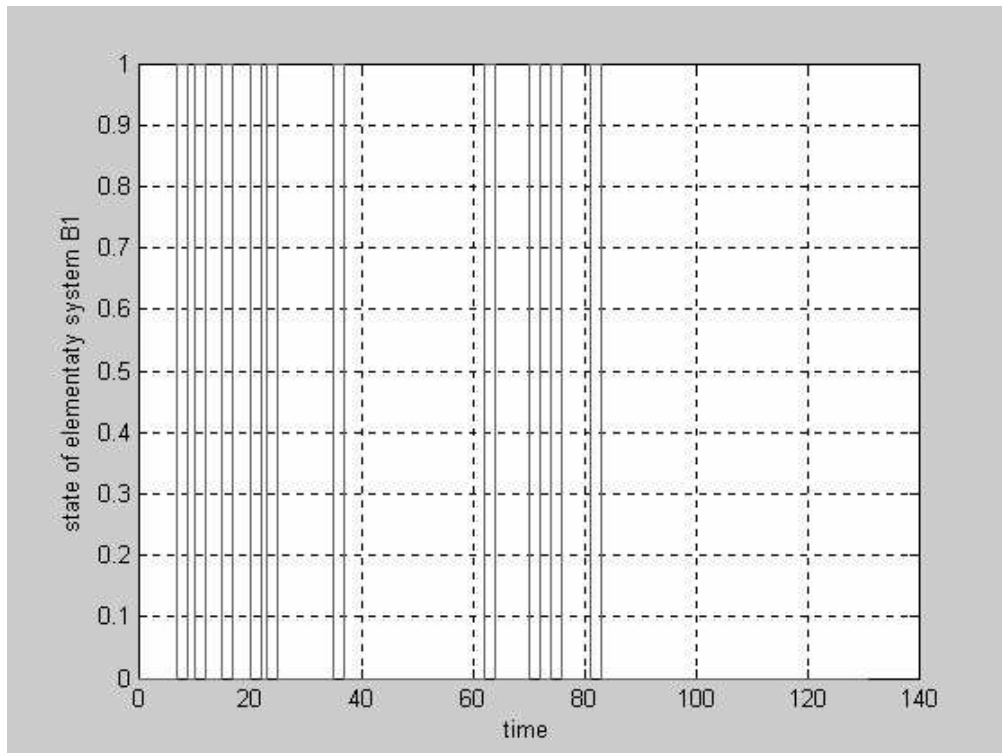
Ως παράδειγμα για τα παραπάνω το πρώτο αεροπλάνο καταφθάνει στο αεροδρόμιο την χρονική στιγμή 6 εισέρχεται στον buffer 1 προσθέτοντας ένα σκαλοπάτι την ίδια χρονική στιγμή στο χρονοδιάγραμμά 4.1.2 και αρχίζει να εξυπηρετείται αμέσως από το A1 θέτοντας τιμή 1 στο χρονοδιάγραμμά 4.1.3. Αφού τελειώσει την εξυπηρέτησή(προσγείωση) του μετά από ένα λεπτό την χρονική στιγμή 7 μειώνεται ένα σκαλοπάτι το 4.1.2 και πέφτει στο μηδέν το 4.1.3.

Τα αεροπλάνα που εξυπηρετούνται από το A1 εισέρχονται αμέσως στον buffer 2 το χρονοδιάγραμμα του οποίου είναι το 4.1.4. Με ίδιο μηχανισμό με αυτόν που περιγράφηκε για τα χρονοδιαγράμματα 4.1.2 και 4.1.3 έχουμε αλλαγές και στα 4.1.4 και 4.1.5 αντίστοιχα. Το 4.1.5 είναι το χρονοδιάγραμμα της κατάστασης του στοιχειώδους συστήματος B1 (taxiing in). Έτσι συνεχίζοντας το παράδειγμα του πρώτου αεροπλάνου, αυτό θα εισέλθει στον buffer 2 την χρονική στιγμή 7 προσθέτοντας ένα σκαλοπάτι στο χρονοδιάγραμμά του. Αφού δεν έχουμε ουρά αναμονής, το αεροπλάνο θα αρχίσει να εξυπηρετείται αμέσως την χρονική στιγμή 7 από το στοιχειώδες σύστημα B1 θέτοντας στο χρονοδιάγραμμά 4.1.5 τιμή 1 η οποία μηδενίζεται ξανά την χρονική στιγμή 9 (αφού ο χρόνος λειτουργίας του B1 είναι δύο λεπτά) όταν η εξυπηρέτηση ολοκληρωθεί. Την χρονική στιγμή 9 θα είναι που θα

έχουμε και την μείωση του 4.1.4 κατά ένα σκαλοπάτι. Όπως και πριν τα 4.1.4 και 4.1.5 εμφανίζονται ίδια επειδή δεν έχουμε συσσώρευση αεροπλάνων στον buffer 2.

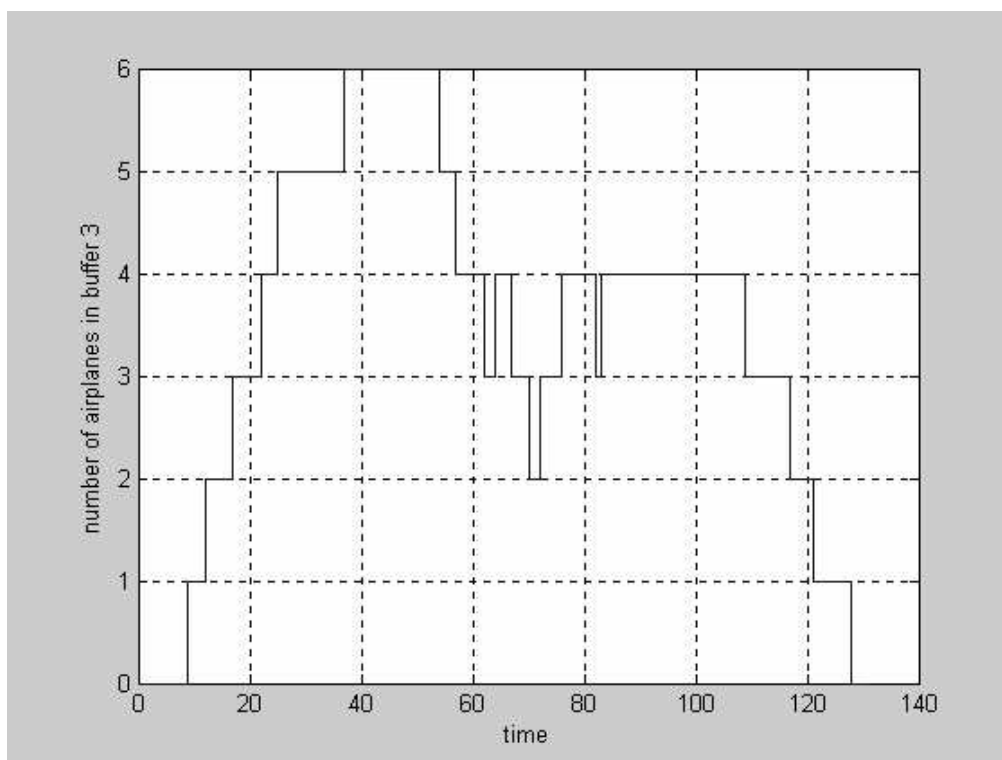


χρονοδιάγραμμα 4.1.4



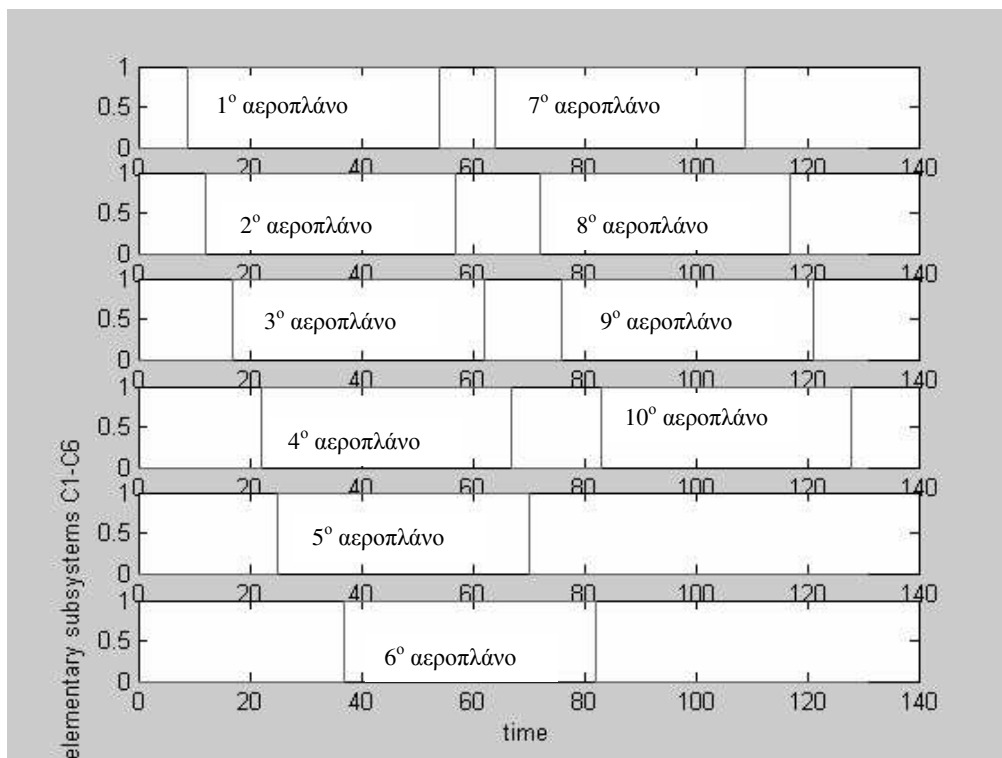
χρονοδιάγραμμα 4.1.5

Τα αεροπλάνα που εξυπηρετούνται από το B1 εισέρχονται αμέσως στον buffer 3 χρονοδιάγραμμα του οποίου είναι το 4.1.6. Ο buffer 3 είναι αυτός που τροφοδοτεί με αεροπλάνα τα στοιχειώδη υποσυστήματα C1-C6 που αποτελούν τις έξι θέσεις στάθμευσης του αεροδρομίου. Ο χρόνος εξυπηρέτησης των υποσυστημάτων αυτών είναι 45 λεπτά. Δηλαδή πολύ μεγαλύτερος από άλλα στοιχειώδη συστήματα ή υποσυστήματα. Αυτό έχει ως αποτέλεσμα να εμφανίζονται ουρές αναμονής από αεροπλάνα στον buffer 3 πολύ πιο συχνά και μεγαλύτερου μήκους από κάθε άλλο buffer του συστήματος. Από την άλλη μεριά όμως τα αεροπλάνα που βρίσκονται στον buffer 3 μπορούν να οδηγηθούν σε έξι εξυπηρετητές ταυτόχρονα. Αυτό έχει ως αποτέλεσμα να μην δημιουργείται ουρά αναμονής εάν ο αριθμός των αεροπλάνων που βρίσκονται στον buffer 3 δεν ξεπερνάει το έξι. Για την συγκεκριμένη περίπτωση που εξετάζουμε παρατηρούμε από το χρονοδιάγραμμα 4.1.6 ότι φθάνουμε το όριο των έξι αεροπλάνων για κάποιο χρονικό διάστημα που σημαίνει κατά την διάρκειά του καταλαμβάνονται από αεροπλάνα και οι έξι διαθέσιμες θέσεις στάθμευσης. Αυτό φαίνεται ακόμα πιο καθαρά από το χρονοδιάγραμμα 4.1.7 όπου περιγράφεται η κατάσταση του κάθε στοιχειώδους υποσυστήματος C1-C6. Συνεχίζοντας το παράδειγμα του πρώτου αεροπλάνου, αυτό θα εισέλθει στον buffer 3 την χρονική στιγμή 9 προσθέτοντας ένα σκαλοπάτι στο χρονοδιάγραμμά 4.1.6. Αφού δεν έχουμε ουρά αναμονής (κάτω από 6 αεροπλάνα στον buffer 3) το αεροπλάνο θα αρχίσει να εξυπηρετείται αμέσως την χρονική στιγμή 9 από το στοιχειώδες υποσύστημα C1 θέτοντας στο πρώτο χρονοδιάγραμμά του 4.1.7 τιμή 1 η οποία μηδενίζεται ξανά την χρονική στιγμή 54 (αφού ο χρόνος λειτουργίας του C1 είναι 45 λεπτά) όταν η εξυπηρέτηση ολοκληρωθεί. Την χρονική στιγμή 54 θα είναι που θα έχουμε και την μείωση του 4.1.6 κατά ένα σκαλοπάτι.



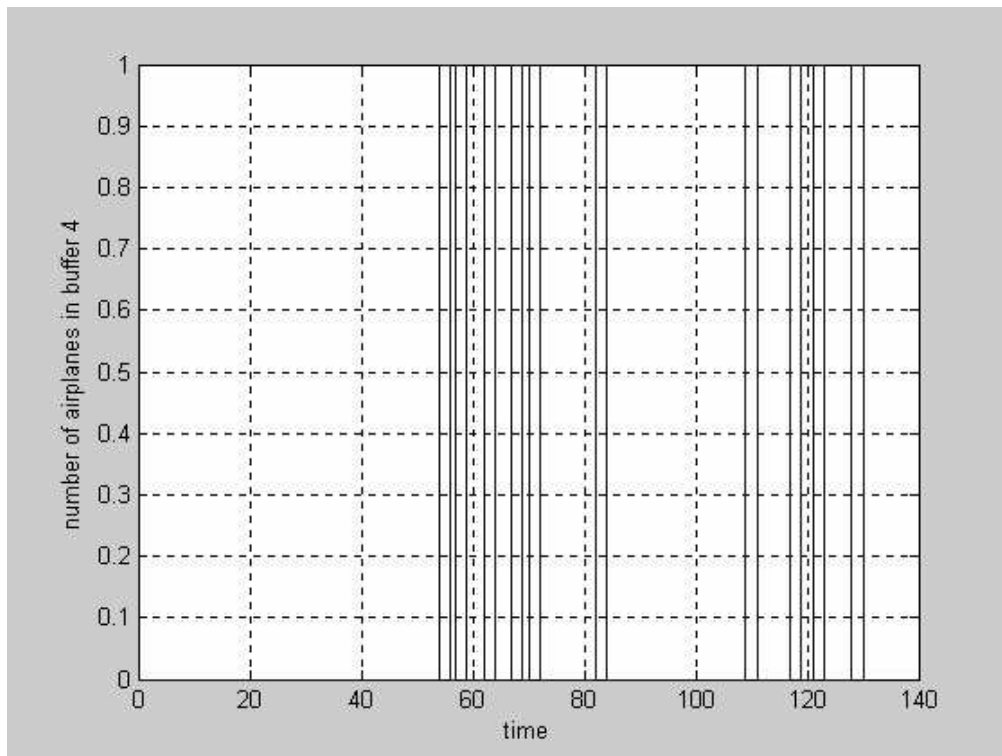
χρονοδιάγραμμα 4.1.6

Στο σημείο αυτό κρίνεται αναγκαία η περαιτέρω επεξήγηση του χρονοδιαγράμματος 4.1.7. Στο χρονοδιάγραμμα αυτό εμφανίζονται έξι χρονοδιαγράμματα ένα για κάθε μία από τις θέσεις στάθμευσης του αεροδρομίου όπου αποτυπώνεται η κατάστασή τους για κάθε χρονική στιγμή που διαρκεί η προσομοίωση. Το πρώτο από πάνω χρονοδιάγραμμα αντιστοιχεί στην πρώτη θέση στάθμευσης το δεύτερο από πάνω στην δεύτερη και ούτω κάθε εξής. Για την συγκεκριμένη περίπτωση που εξετάζουμε το 4.1.7 μας δείχνει ότι το πρώτο και το έβδομο από τα δέκα αεροπλάνα που φθάνουν στο αεροδρόμιο θα εισέλθουν στην 1^η θέση στάθμευσης ,το δεύτερο και το όγδοο στην 2^η θέση ,το τρίτο και το ένατο στην 3^η ,το τέταρτο και δέκατο στην 4^η ,το πέμπτο στην 5^η και το έκτο στην 6^η θέση στάθμευσης. Αν υπήρχαν περισσότερα αεροπλάνα για εξυπηρέτηση από τις θέσεις στάθμευσης ,αυτά θα συνέχιζαν να εισέρχονται σε αυτές με τον ίδιο αυτό τρόπο.

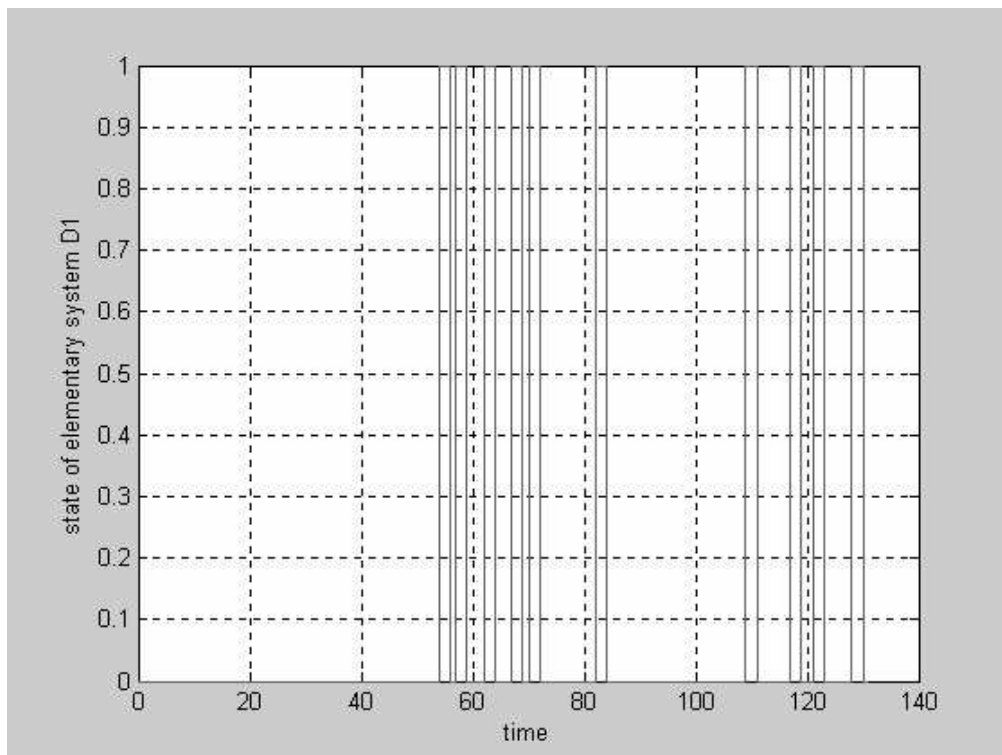


χρονοδιάγραμμα 4.1.7

Τα αεροπλάνα που εξυπηρετούνται από τα στοιχειώδη υποσυστήματα C1-C6 εισέρχονται αμέσως στον buffer 4 το χρονοδιάγραμμα του οποίου είναι το 4.1.8. Κάθε φορά που έχουμε άφιξη προστίθεται και ένα σκαλοπάτι και αν δεν έχουμε ουρά αναμονής το αεροπλάνο αρχίζει να εξυπηρετείτε αμέσως από το στοιχειώδες σύστημα D1 (taxiing out) το οποίου αλλάζει καταστάσεις σύμφωνα με το χρονοδιάγραμμα 4.1.9. Όταν η εξυπηρέτηση ολοκληρωθεί μετά ένα από χρονικό διάστημα ίσο με τον χρόνο λειτουργίας του D1 που έχουμε ορίσει στα δύο λεπτά το 4.1.8 μειώνεται κατά ένα σκαλοπάτι και η κατάσταση στο 4.1.9 μηδενίζεται. Συνεχίζοντας το παράδειγμα με το πρώτο αεροπλάνο παρατηρούμε ότι αυτό εισέρχεται στον buffer 4 την χρονική στιγμή 54 ,απασχολεί το D1 από την ίδια αυτή χρονική στιγμή μέχρι και την 56 οπότε και τελειώνει η εξυπηρέτηση του σε αυτό. Τα 4.1.8 και 4.1.9 είναι ίδια επειδή δεν έχουμε συσσώρευση αεροπλάνων στον buffer 4.

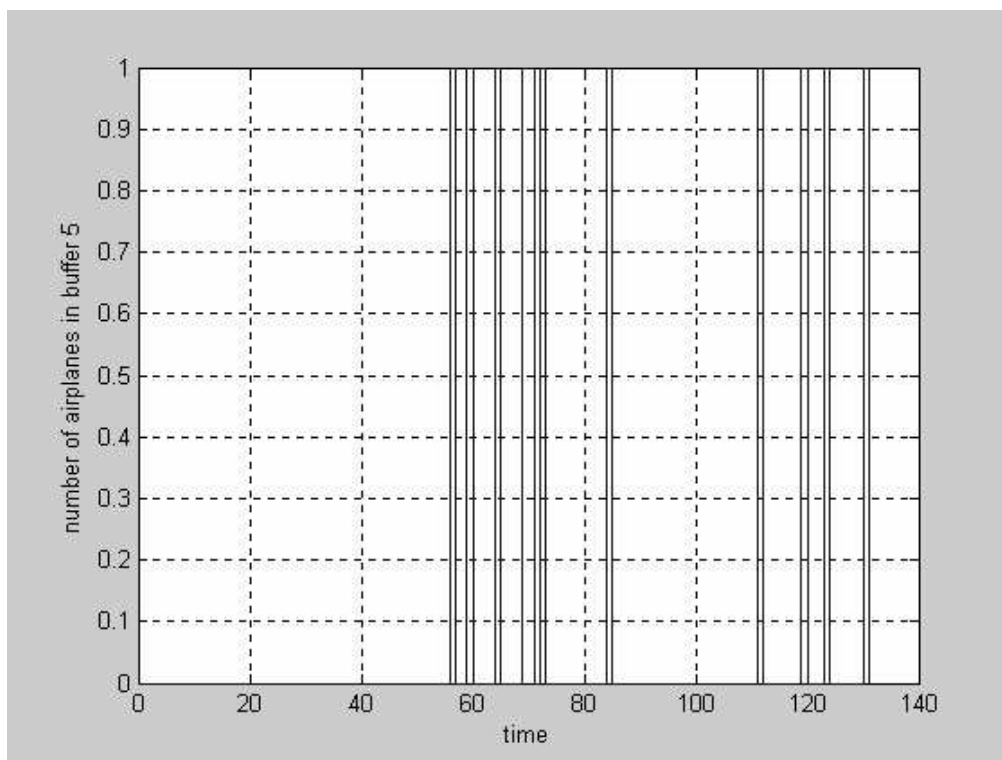


χρονοδιάγραμμα 4.1.8

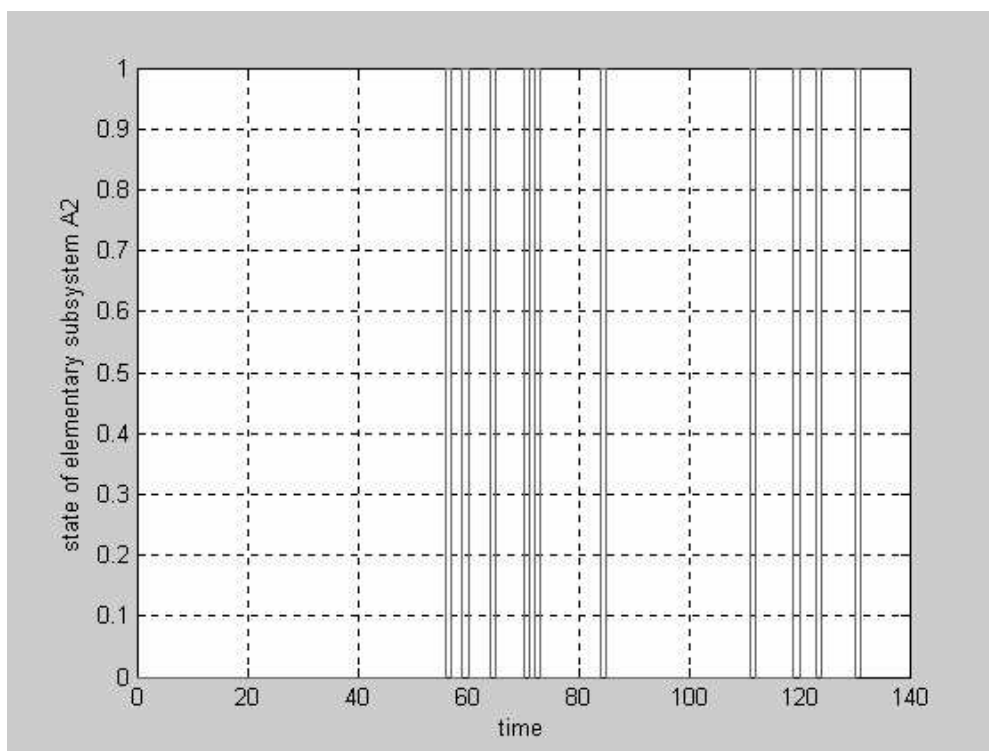


χρονοδιάγραμμα 4.1.9

Τα αεροπλάνα που εξυπηρετούνται από το στοιχειώδες σύστημα D1 εισέρχονται αμέσως στον buffer 5 το χρονοδιάγραμμα του οποίου είναι το 4.1.10. Τα αεροπλάνα που βρίσκονται στον buffer 5 εξυπηρετούνται από το στοιχειώδες σύστημα A2 (take_off) το οποίο αλλάζει καταστάσεις σύμφωνα με το χρονοδιάγραμμα 4.1.11 και έχει χρόνο λειτουργίας ένα λεπτό. Οι αλλαγές στα χρονοδιαγράμματα 4.1.10 και 4.1.11 γίνονται με τρόπο παρόμοιο με αυτόν που έχει ήδη περιγραφεί. Συνεχίζοντας και εδώ το παράδειγμα με το πρώτο αεροπλάνο παρατηρούμε ότι αυτό εισέρχεται στον buffer 5 την χρονική στιγμή 56, απασχολεί το A2 από την ίδια αυτή χρονική στιγμή μέχρι και την 57 οπότε και τελειώνει η εξυπηρέτηση του σε αυτό. Και εδώ τα 4.1.10 και 4.1.11 είναι ίδια επειδή δεν έχουμε συσσώρευση αεροπλάνων στον buffer 5.



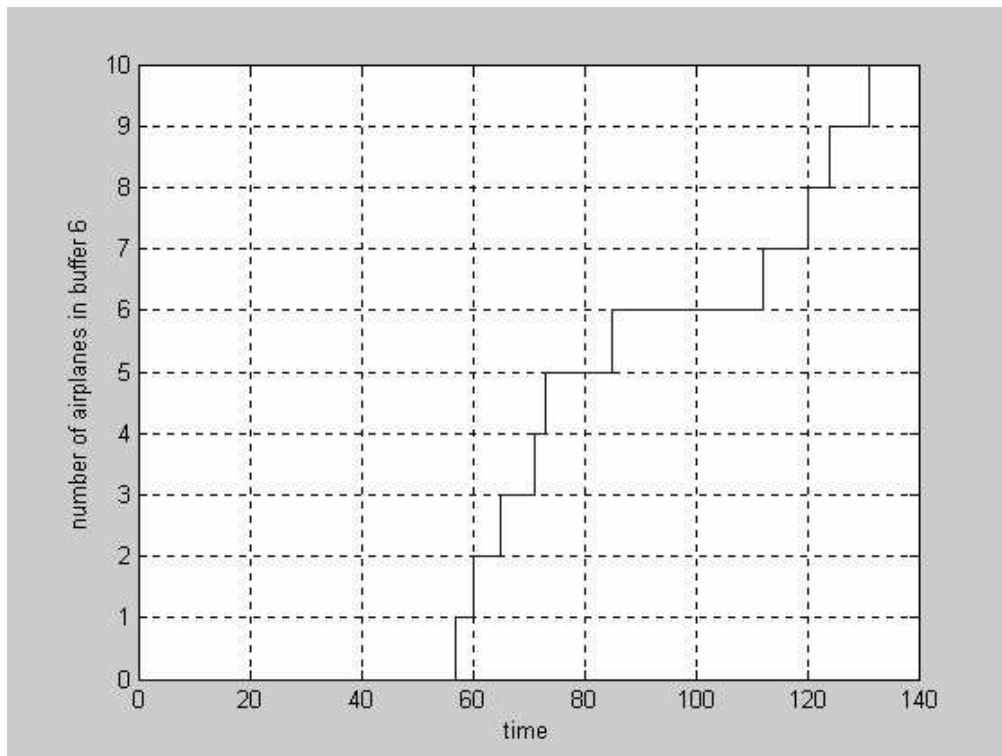
χρονοδιάγραμμα 4.1.10



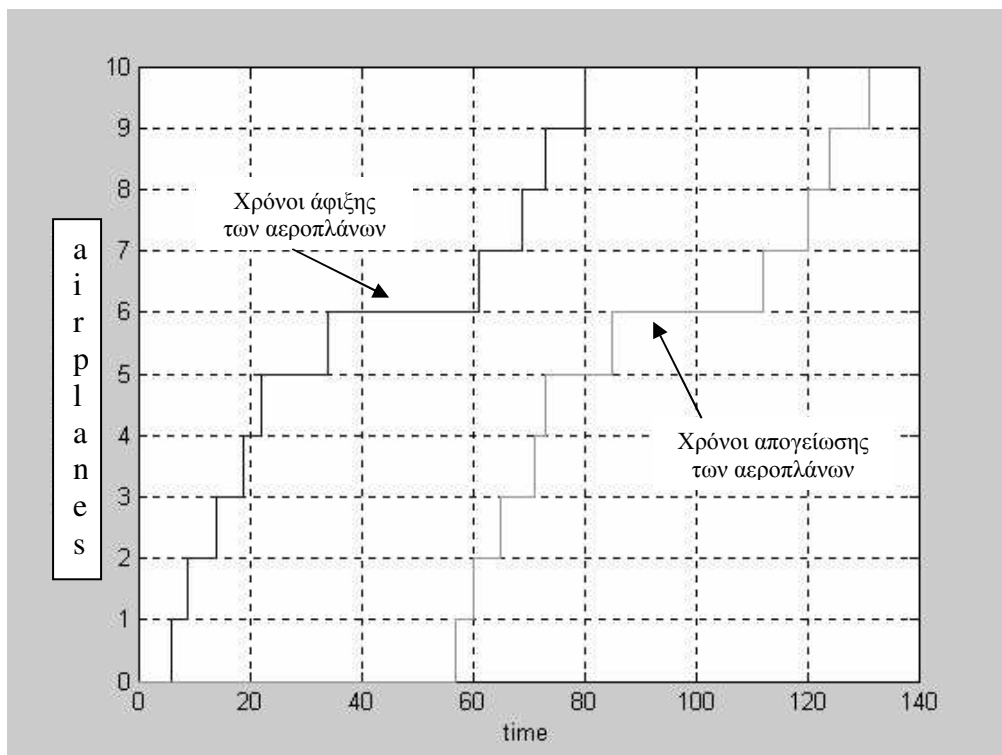
χρονοδιάγραμμα 4.1.11

Τα αεροπλάνα που εξυπηρετούνται από το στοιχειώδες υποσύστημα A2 εισέρχονται αμέσως στον buffer 6 το χρονοδιάγραμμα του οποίου είναι το 4.1.12 και αποτελεί την έξοδο του συνολικού συστήματος του αεροδρομίου. Οι χρονικές στιγμές κατά τις οποίες έχουμε αύξηση ενός σκαλοπατιού στο 1.1.12 είναι οι χρόνοι όπου κάθε αεροπλάνο ολοκληρώνει την απογείωσή του. Έτσι το πρώτο αεροπλάνο που προσγειώθηκε στο αεροδρόμιο θα απογειωθεί από αυτό την χρονική στιγμή 57.

Τέλος στο χρονοδιάγραμμα 4.1.13 παρουσιάζονται μαζί ο τρόπος με τον οποίο κατανέμονται στον χρόνο της προσομοίωσης οι αφίξεις και οι απογειώσεις των αεροπλάνων στο και από το αεροδρόμιο αντίστοιχα. Θέλει προσοχή το γεγονός ότι οι ή χρονική στιγμή άφιξης ενός αεροπλάνου δεν είναι απαραίτητο να ταυτίζεται με την χρονική στιγμή που αρχίζει και η προσγείωσή του. Μπορεί λόγω συμφόρησης των αεροπλάνων στον αέρα ή λόγω προτεραιότητας στην απογείωση κάποιου αεροπλάνου που βρίσκεται στο έδαφος να υπάρξει καθυστέρηση από την στιγμή της άφιξης ενός αεροπλάνου στον εναέριο χώρο του αεροδρομίου μέχρι την έναρξη της προσγείωσής του.



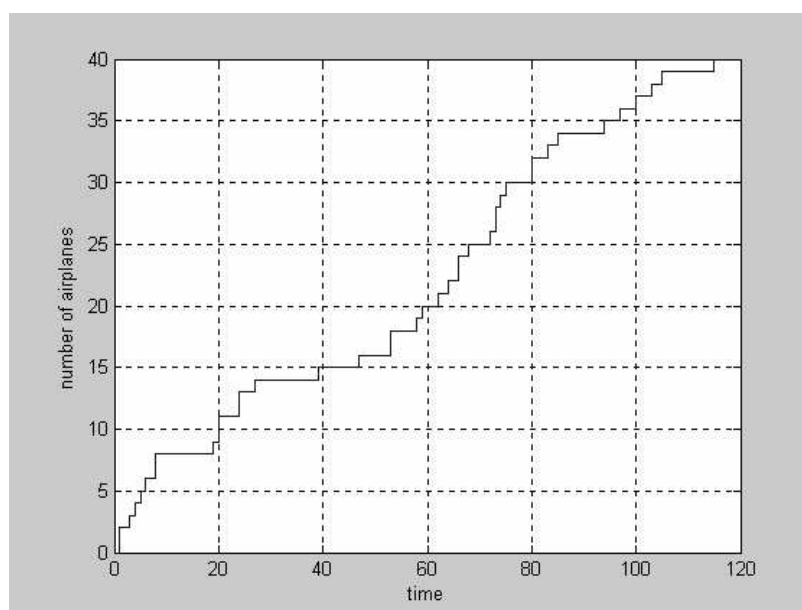
χρονοδιάγραμμα 4.1.12



χρονοδιάγραμμα 4.1.13

4.3 Δεύτερο Παράδειγμα □ Ρυθμός Αφίξεων 40/100

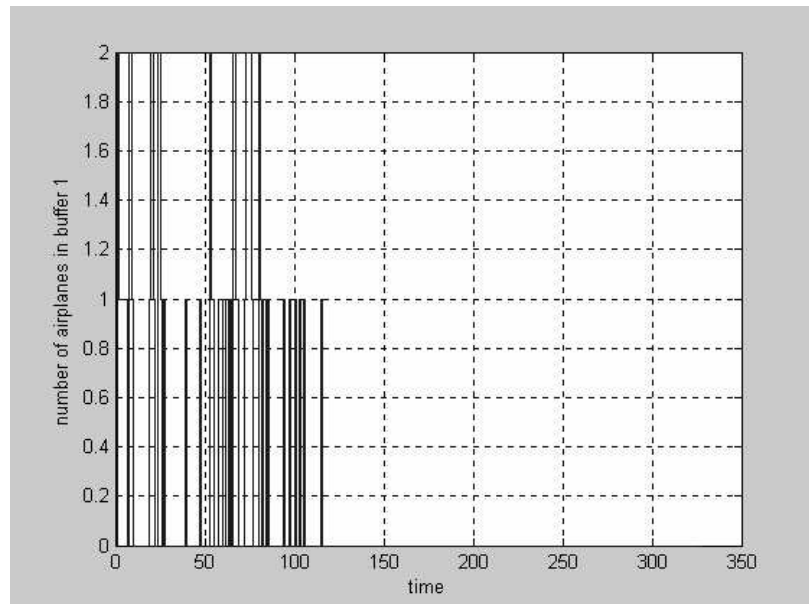
Θεωρούμε την περίπτωση όπου έχουμε την άφιξη 40 αεροπλάνων στο αεροδρόμιο μέσα σε ένα επιθυμητό χρονικό διάστημα που είναι τα 100 λεπτά της ώρας. Δηλαδή $\lambda=40/100$. Αυτή είναι μια περίπτωση υπερβολικής συμφόρησης των αεροπλάνων στο αεροδρόμιο αλλά παρατίθεται καθαρά για λόγους μελέτης του αεροδρομίου και ταυτόχρονα πιστοποίησης της σωστής λειτουργίας της γλώσσας προσομοίωσης που έχει αναπτυχθεί. Παρακάτω παρουσιάζονται τα αποτελέσματα της συγκεκριμένης προσομοίωσης σε μορφή χρονοδιαγραμμάτων των οποίων η σειρά είναι ίδια με την πρώτη περίπτωση. Εκτενή σχόλια παραλείπονται αφού τα χρονοδιαγράμματα θα πρέπει να είναι κατανοητά από τον σχολιασμό τους στην πρώτη περίπτωση. Περιοριζόμαστε μόνο σε παρατηρήσεις καταστάσεων οι οποίες δεν είχαν εμφανιστεί στο προηγούμενο παράδειγμα.



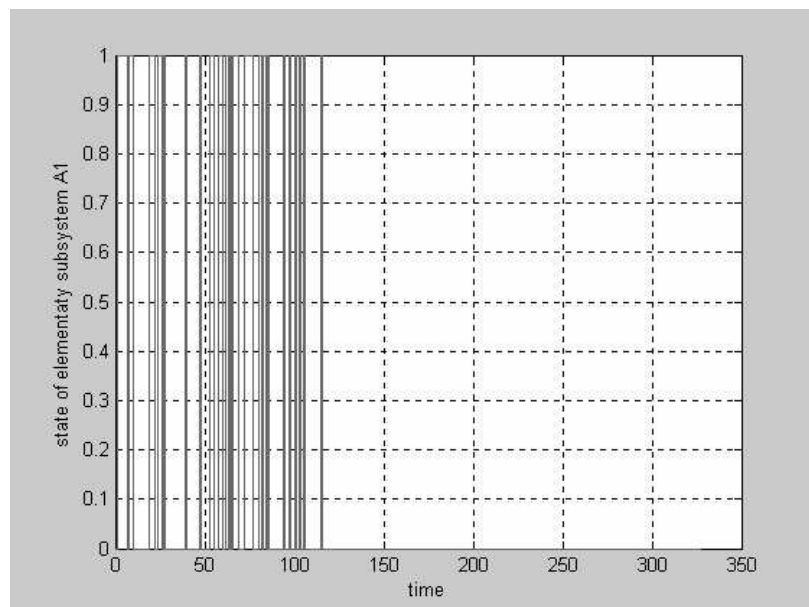
χρονοδιάγραμμα 4.2.1

Στο 4.2.1 φαίνονται οι τυχαίες αφίξεις των αεροπλάνων στον εναέριο χώρο του αεροδρομίου. Λόγω του μεγάλου αριθμού αφίξεων σε μικρό χρονικό διάστημα παρουσιάζονται και αφίξεις δύο αεροπλάνων την ίδια χρονική στιγμή οι οποίες φυσικά εξυπηρετούνται η μία μετά από την άλλη από τον διάδρομο προσγείωσης.

Στο χρονοδιάγραμμα 4.2.2 φαίνεται πότε έχουμε άφιξη δύο αεροπλάνων ταυτόχρονα. Είναι οι χρονικές στιγμές κατά τις οποίες εμφανίζονται σκαλοπάτια με μήκος δύο σκαλοπατιών το ένα. Στο 4.2.3 φαίνονται οι χρόνοι κατά τους οποίους το A1 είναι κατειλημμένο.

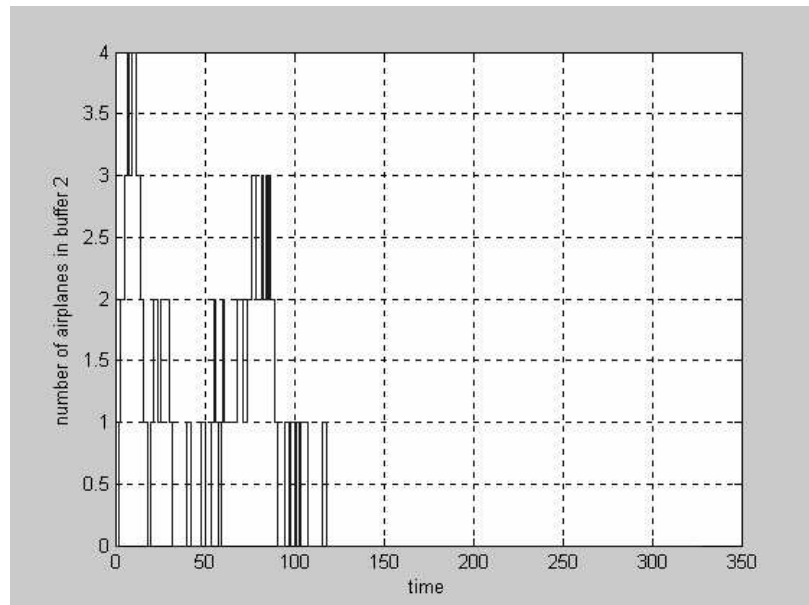


χρονοδιάγραμμα 4.2.2

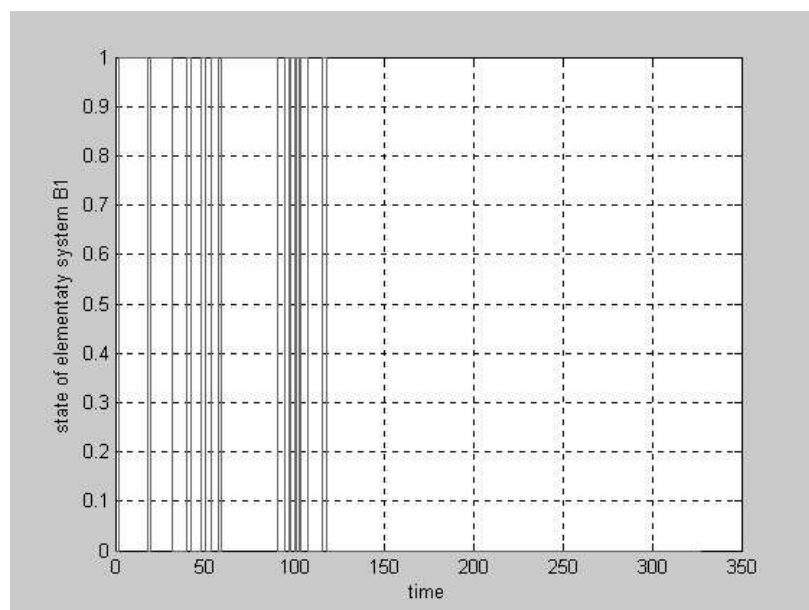


χρονοδιάγραμμα 4.2.3

Το χρονοδιάγραμμα 4.2.4 μας δείχνει ότι ο μέγιστος αριθμός αεροπλάνων που περιμένουν σε ουρά για να εξυπηρετηθούν από το B1 είναι τρία τα οποία προστιθέμενα στο ένα αεροπλάνο που ήδη εξυπηρετείται, μας δίνουν τον μέγιστο αριθμό σκαλοπατιών που έχουμε στο 4.2.4 και είναι τα τέσσερα. Στο 4.2.5 φαίνονται οι χρόνοι κατά τους οποίους το B1 είναι κατειλημμένο εξυπηρετώντας κάποιο από τα αεροπλάνα.

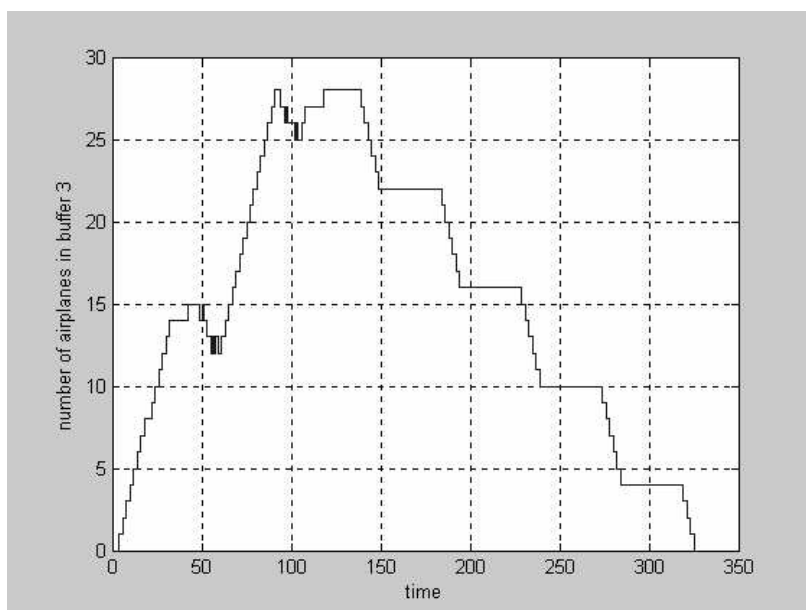


χρονοδιάγραμμα 4.2.4

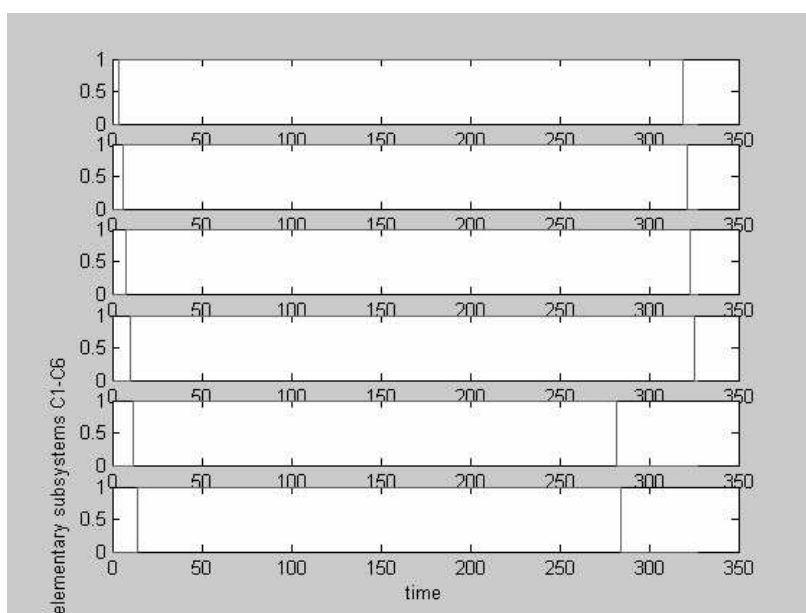


χρονοδιάγραμμα 4.2.5

Το χρονοδιάγραμμα 4.2.6 έχει μέγιστο αριθμό σκαλοπατιών τα 28. Αυτό μας φανερώνει ότι, πέρα από τα έξι αεροπλάνα τα οποία εξυπηρετούνται από τις ανάλογες θέσεις στάθμευσης, για κάποιο χρονικό διάστημα τα αεροπλάνα που περιμένουν για εξυπηρέτηση φθάνουν και τα 22. Κάτι τέτοιο φυσικά δεν είναι ανεκτό από το αεροδρόμιο λόγω περιορισμών στους χώρους του, αλλά στα πλαίσια της προσομοίωσης το εξετάζουμε. Στο χρονοδιάγραμμα 4.2.7 φαίνεται καθαρά η υπερλειτουργία των χώρων στάθμευσης από το γεγονός ότι είναι όλες σχεδόν συνέχεια κατειλημμένες.

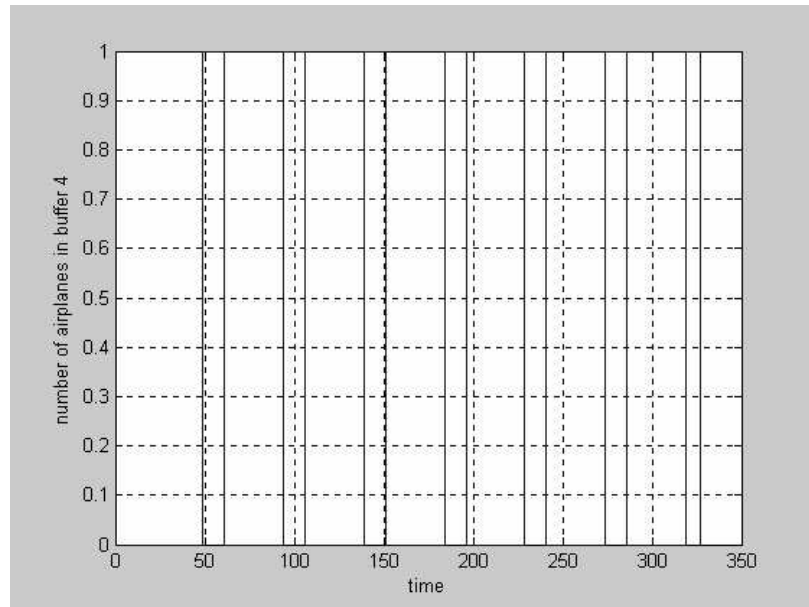


χρονοδιάγραμμα 4.2.6

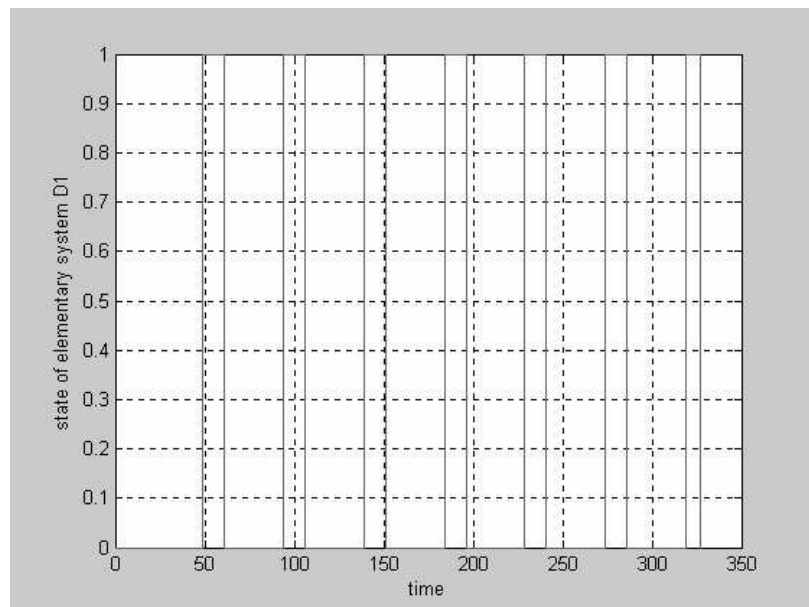


χρονοδιάγραμμα 4.2.7

Τα χρονοδιαγράμματα 4.2.8 και 4.2.9 μας δείχνουν τα αεροπλάνα που βρίσκονται στον buffer 3 και την κατάσταση του D1 αντίστοιχα. Δεν έχουμε ουρά αναμονής και για αυτό παρουσιάζονται ίδια.

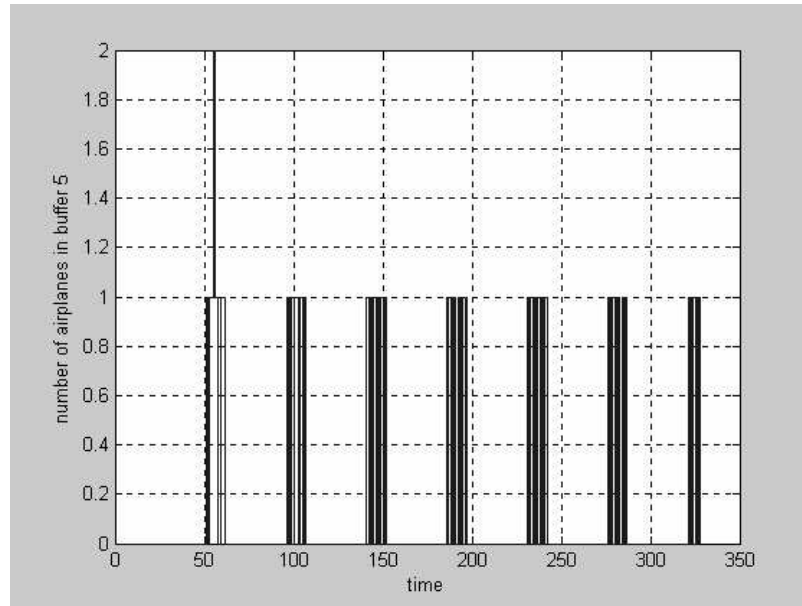


χρονοδιάγραμμα 4.2.8

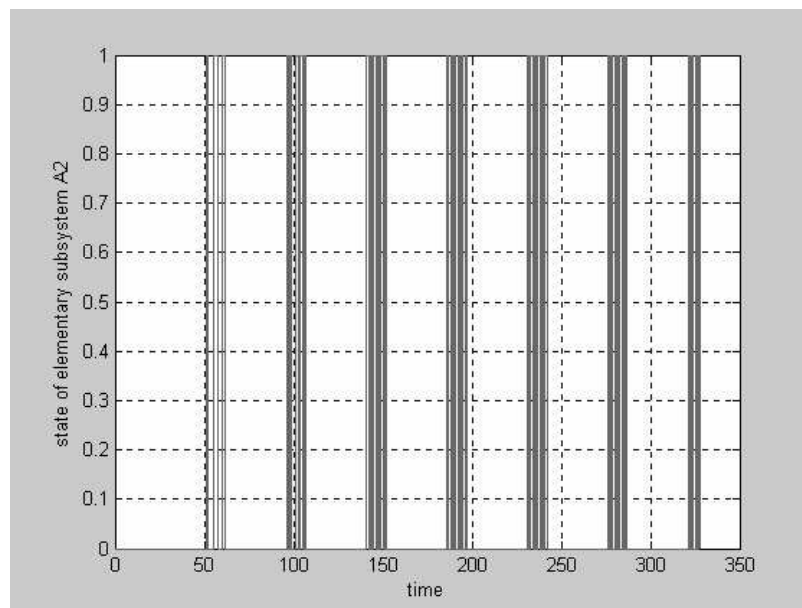


χρονοδιάγραμμα 4.2.9

Στο χρονοδιάγραμμα 4.2.10 φαίνονται τα αεροπλάνα που βρίσκονται στον buffer 5. Παρατηρούμε ότι κάποια στιγμή τα σκαλοπάτια αυξάνονται σε 2 που σημαίνει ότι έχουμε αναμονή κάποιου αεροπλάνου για απογείωση και όχι άμεση εξυπηρέτησή του από το A2 του οποίου οι καταστάσεις λειτουργίας ή μη λειτουργίας παρουσιάζονται στο 4.2.11. Μια τέτοια κατάσταση μπορεί να δημιουργηθεί όταν ο διάδρομος προσγείωσης-απογείωσης χρησιμοποιείται (πάντα με τις προτεραιότητες που έχουν δοθεί) από ένα αεροπλάνο το οποίο προσγειώνεται.

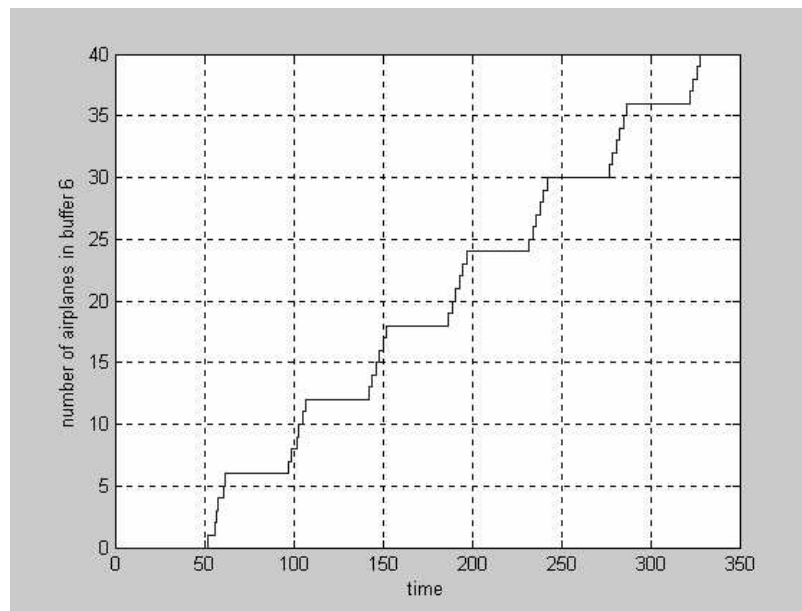


χρονοδιάγραμμα 4.2.10

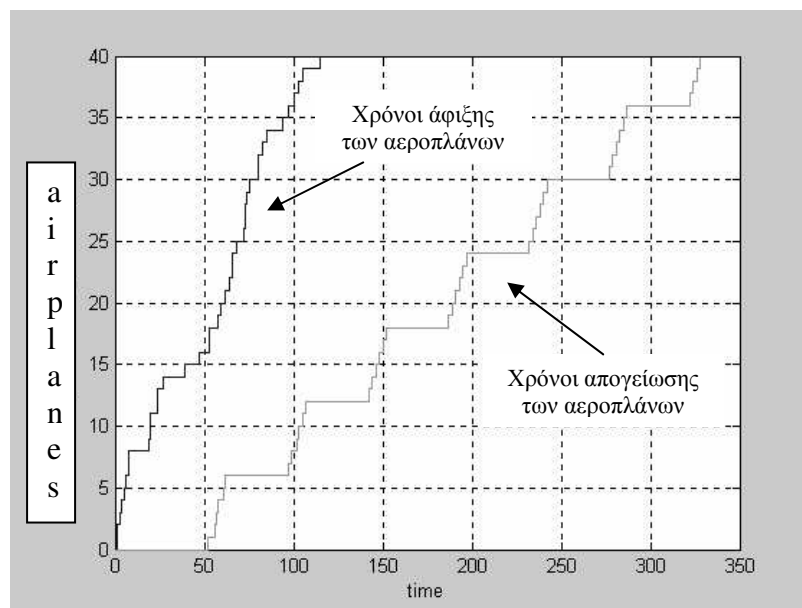


χρονοδιάγραμμα 4.2.11

Το χρονοδιάγραμμα 4.2.12 μας δείχνει τις χρονικές στιγμές κατά τις οποίες ολοκληρώνονται οι απογειώσεις των αεροπλάνων. Παρατηρούμε ότι αυτές γίνονται κατά κύματα των έξι αεροπλάνων (έξι σκαλοπάτια) τα οποία έχουν κάποια περιοδικότητα. Η περιοδικότητα των κυμάτων αυτών είναι άμεσα συνδεδεμένη με τον χρόνο παραμονής των αεροπλάνων στους χώρους στάθμευσης. Τέλος παραθέτουμε και το χρονοδιάγραμμα 4.2.13 όπου φαίνονται μαζί οι αφίξεις των αεροπλάνων και οι απογειώσεις τους από το αεροδρόμιο για την συγκεκριμένη περίπτωση που εξετάσαμε.



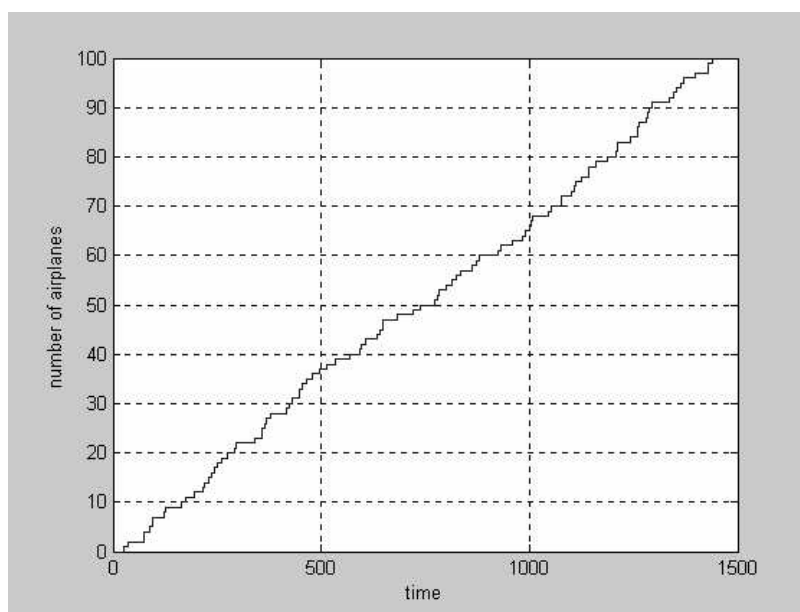
χρονοδιάγραμμα 4.2.12



χρονοδιάγραμμα 4.2.13

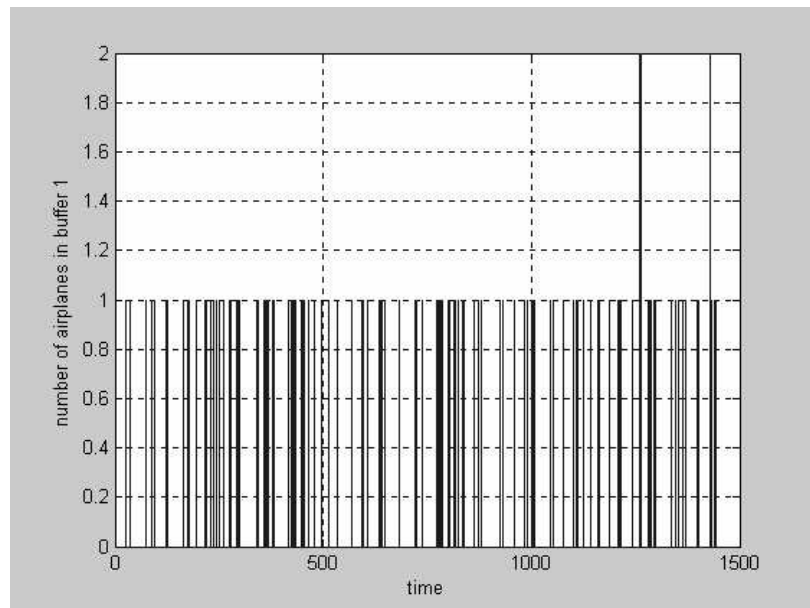
4.4 Τρίτο Παράδειγμα □ Ρυθμός Αφίξεων 100/1440

Ως τρίτο παράδειγμα επιλέχθηκε ένα που καλύπτει την προσομοίωση του αεροδρομίου για μια ολόκληρη μέρα λειτουργίας του. Έτσι επιλέγοντας ως χρονικό διάστημα εμφάνισης των αφίξεων τα 1440 λεπτά, μία μέρα δηλαδή, και θέτοντας τον επιθυμητό αριθμό των αεροπλάνων που καταφθάνουν στο αεροδρόμιο στα 100 προχωράμε στην προσομοίωση αυτής της περίπτωσης. Ο ρυθμός αφίξεων είναι τώρα $\lambda=100/1440$ και η κατανομή τους στο χρόνο φαίνεται στο χρονοδιάγραμμα 4.3.1

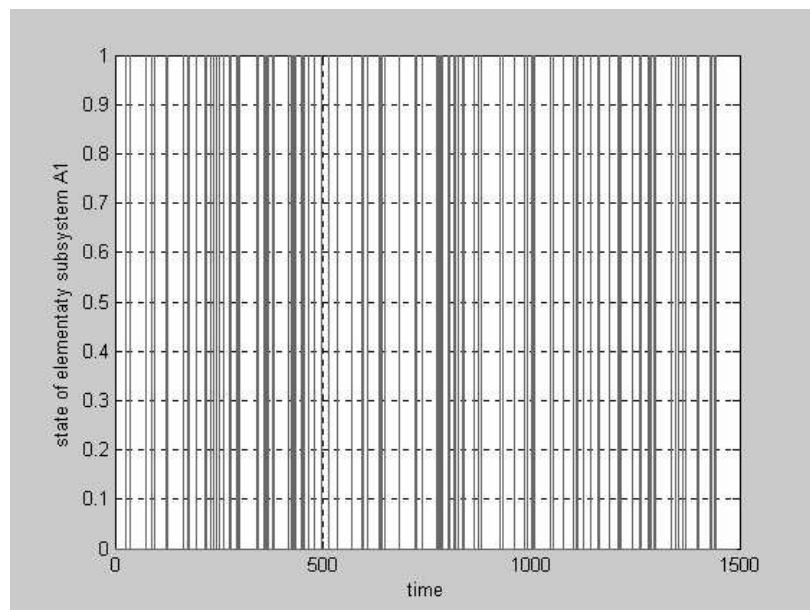


χρονοδιάγραμμα 4.3.1

Και εδώ έχουμε περιπτώσεις ταυτόχρονης άφιξης δύο αεροπλάνων όπως επιδεικνύει και το χρονοδιάγραμμα 4.3.2 καθώς σημειώνεται αύξηση του buffer 1 στην τιμή 2 κάποιες χρονικές στιγμές. Το 4.3.3 μας δείχνει πότε τα 100 αφικνούμενα αεροπλάνα κάνουν χρήση του διαδρόμου προσγείωσης ο οποίος αντιστοιχείται στο στοιχειώδες υποσύστημα A1.

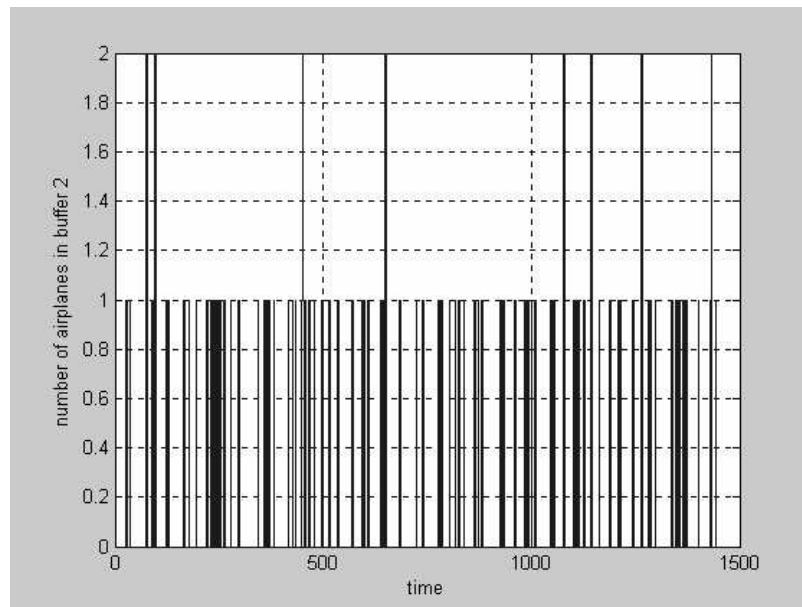


χρονοδιάγραμμα 4.3.2

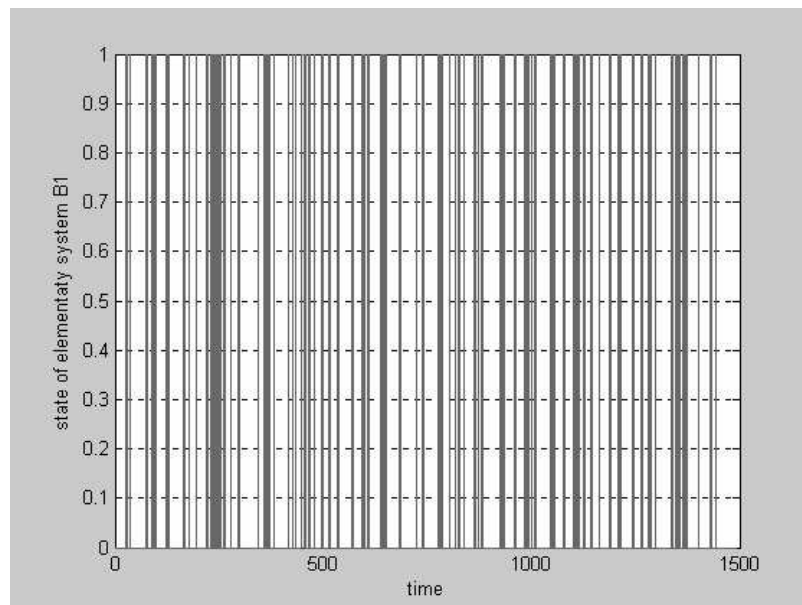


χρονοδιάγραμμα 4.3.3

Στο χρονοδιάγραμμα 4.3.4 περιγράφεται η κίνηση των αεροπλάνων στον buffer 2 όπου και εδώ έχουμε συσσώρευση αεροπλάνων που περιμένουν για εξυπηρέτηση λόγω του ότι το στοιχειώδες σύστημα B1 είναι κατειλημμένο. Η κατάσταση του B1 περιγράφεται στο χρονοδιάγραμμα 4.3.5.

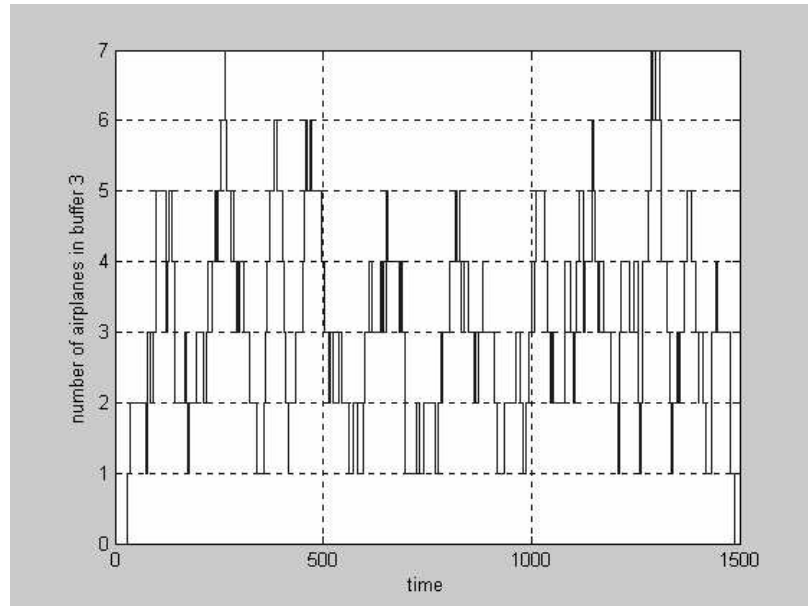


χρονοδιάγραμμα 4.3.4

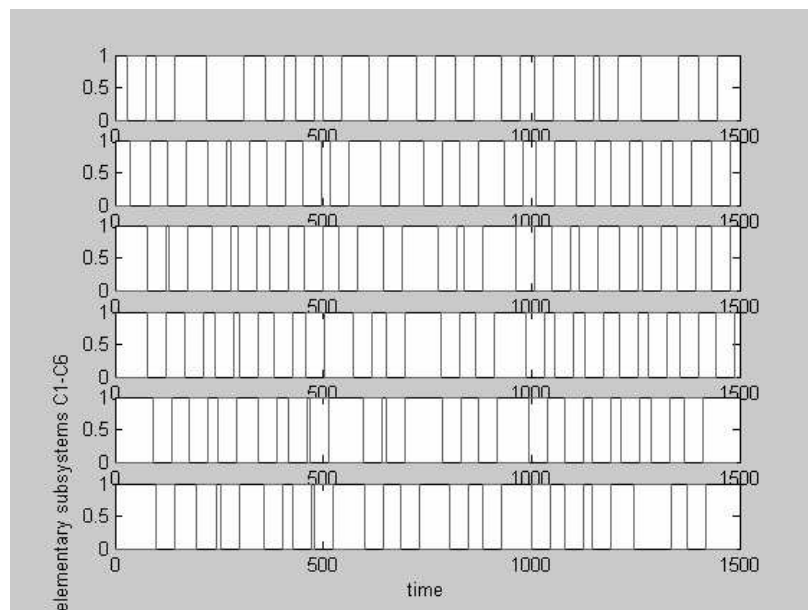


χρονοδιάγραμμα 4.3.5

Το χρονοδιάγραμμα 4.3.6 φανερώνει ότι έχουμε αναμονή μέχρι του ενός αεροπλάνου για εξυπηρέτηση από τις έξι θέσεις στάθμευσης η οποία συμβαίνει κατά τα χρονικά διαστήματα όπου τα σκαλοπάτια φθάνουν τα επτά. Στο 4.3.7 περιγράφεται ο τρόπος με τον οποίο καταλαμβάνονται οι θέσεις στάθμευσης από τα αεροπλάνα.

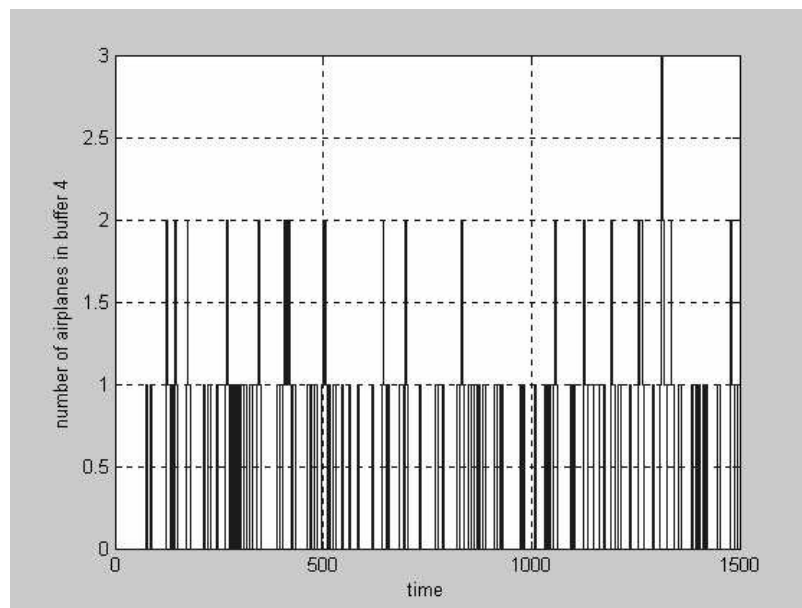


χρονοδιάγραμμα 4.3.6

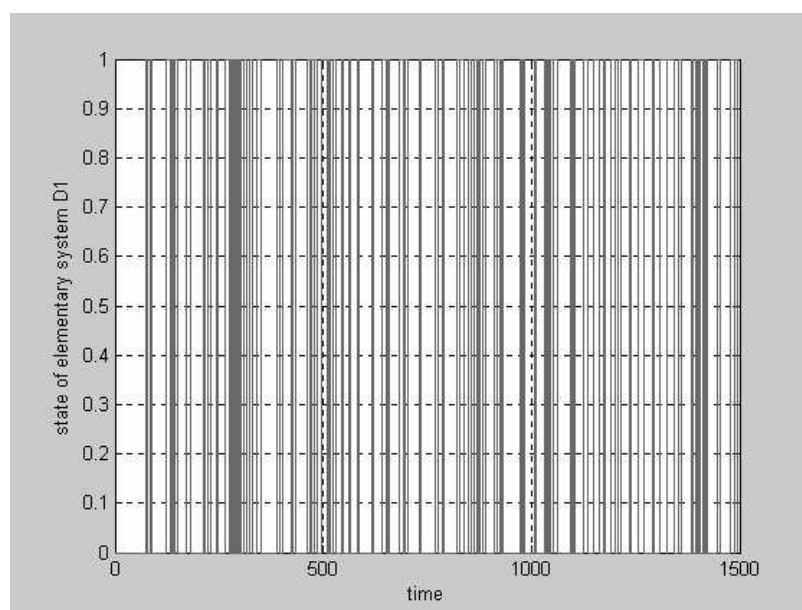


χρονοδιάγραμμα 4.3.7

Τα χρονοδιαγράμματα 4.3.8 και 4.3.9 παρουσιάζουν την κίνηση των αεροπλάνων στον buffer 4 και την κατάσταση του στοιχειώδους συστήματος D1 αντίστοιχα. Παρατηρούμε ότι έχουμε συσσώρευση άνω του ενός αεροπλάνου στον buffer 4 λόγω του ότι μπορεί να αποχωρούν δύο αεροπλάνα από τις θέσεις στάθμευσης σχεδόν την ίδια χρονική στιγμή. Διαφορά ενός λεπτού στις χρονικές στιγμές αποχώρησης τους αρκεί αφού αν ένα αποχωρεί το 120^ο λεπτό, καταλαμβάνει το D1 για 2 λεπτά (χρόνος λειτουργίας του) που σημαίνει ότι όταν ένα άλλο αεροπλάνο αποχωρήσει στο 121^ο λεπτό από κάποια θέση στάθμευσης θα συναντήσει το D1 κατειλημμένο και θα αναγκαστεί να περιμένει για 1 λεπτό.

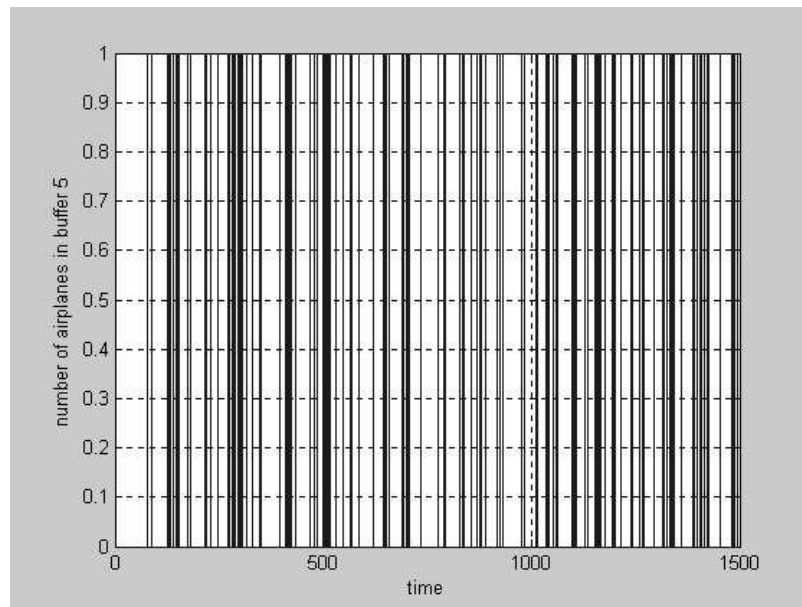


χρονοδιάγραμμα 4.3.8

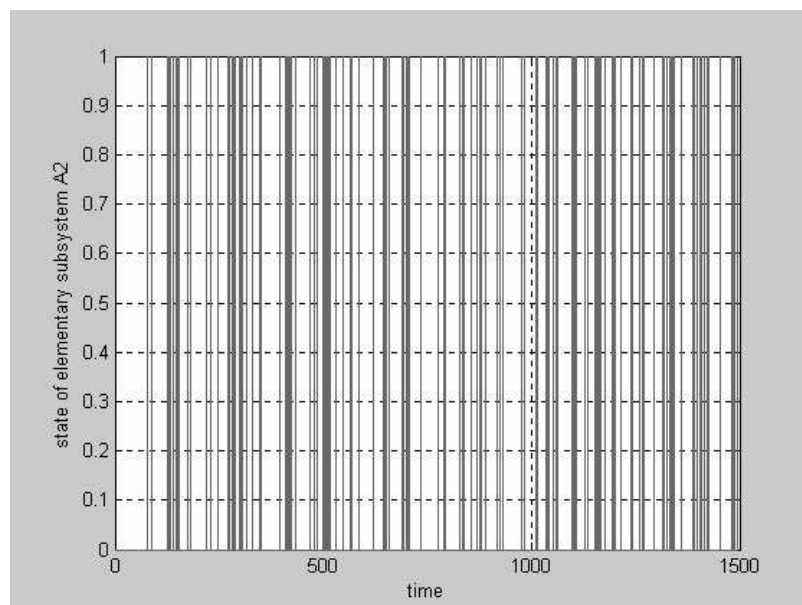


χρονοδιάγραμμα 4.3.9

Τα χρονοδιαγράμματα 4.3.10 και 4.3.11 είναι εντελώς ίδια που σημαίνει ότι δεν έχουμε συμφόρηση αεροπλάνων που περιμένουν να απογειωθούν. Δεν τυχαίνει δηλαδή να έχουμε ταύτιση των χρόνων που ένα αεροπλάνο θέλει να απογειωθεί και ένα άλλο να θέλει να προσγειωθεί. Αν συνέβαινε κάτι τέτοιο, εφόσον δεν περιμένουν πάνω από δύο αεροπλάνα για απογείωση, θα δινόταν προτεραιότητα χρήσης του διαδρόμου προσγείωσης-απογείωσης για προσγείωση και τότε θα είχαμε συσσώρευση αεροπλάνων στον buffer 5.

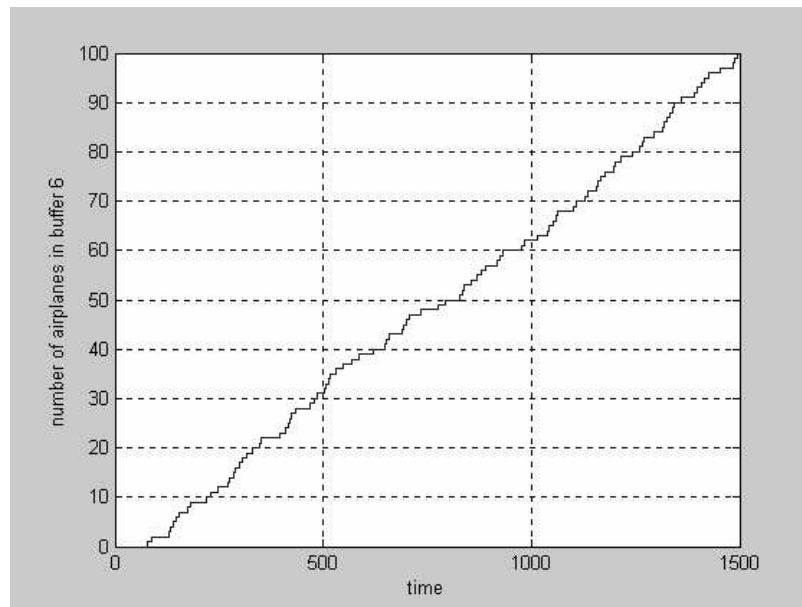


χρονοδιάγραμμα 4.3.10

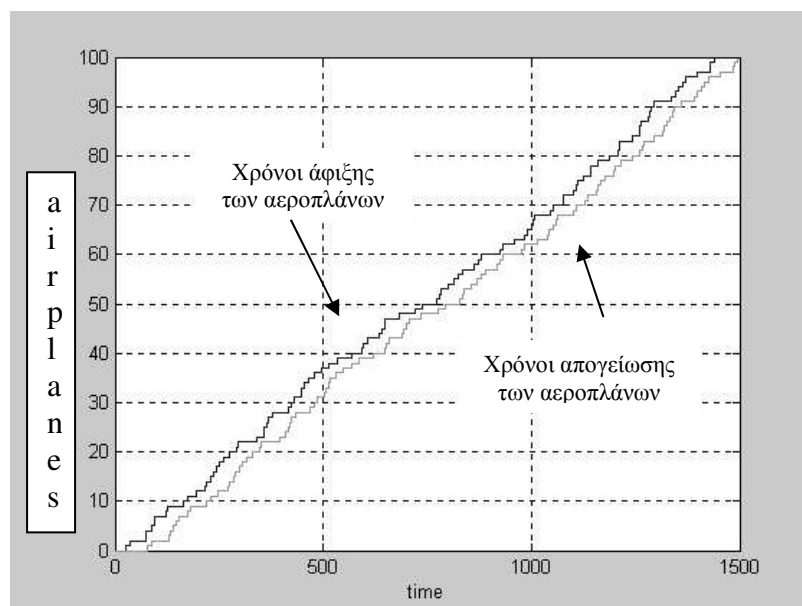


χρονοδιάγραμμα 4.3.11

Το χρονοδιάγραμμα 4.3.12 μας δείχνει τις χρονικές στιγμές κατά τις οποίες ολοκληρώνονται οι απογειώσεις των αεροπλάνων ενώ στο 4.2.13 φαίνονται μαζί οι αφίξεις των αεροπλάνων και οι απογειώσεις τους από το αεροδρόμιο για την συγκεκριμένη περίπτωση που εξετάσαμε.



χρονοδιάγραμμα 4.3.12

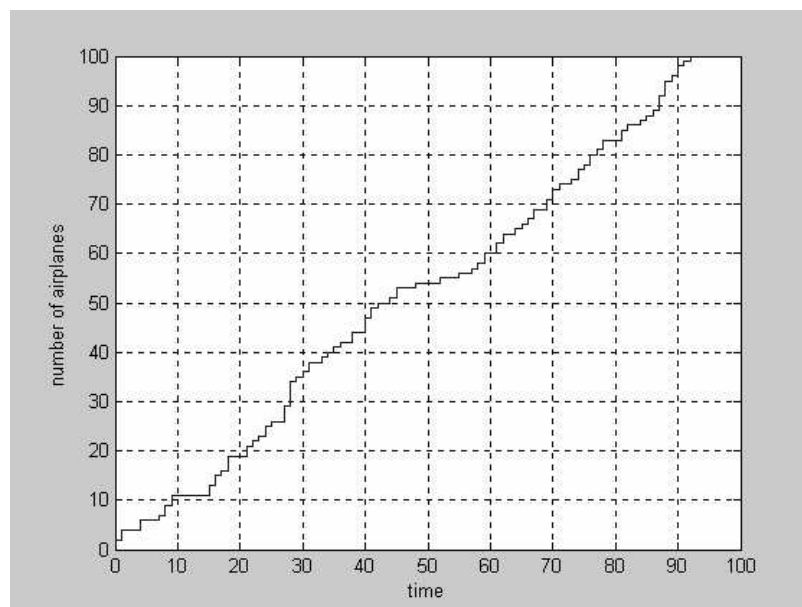


χρονοδιάγραμμα 4.3.13

4.5 Τέταρτο Παράδειγμα □ Ρυθμός Αφίξεων 100/100

Στο παράδειγμα αυτό δίνεται έμφαση στον μηχανισμό των προτεραιοτήτων που δίνονται στα αεροπλάνα που προσγειώνονται και απογειώνονται στον ίδιο διάδρομο. Για να γίνει κάτι τέτοιο απολύτως κατανοητό επιλέγεται μια ακραία περίπτωση που έχουμε αφίξεις 100 αεροπλάνων στο αεροδρόμιο κατά την διάρκεια ενός επιθυμητού χρονικού διαστήματος που είναι τα 100 λεπτά. Δηλαδή $\lambda=100/100$. Παρακάτω παρουσιάζονται τα αναγκαία χρονοδιαγράμματα για μια επαρκή επίβλεψη του μηχανισμού προτεραιοτήτων. Τα υπόλοιπα παραλείπονται αφού έχουν εξηγηθεί επαρκώς στα προηγούμενα παραδείγματα.

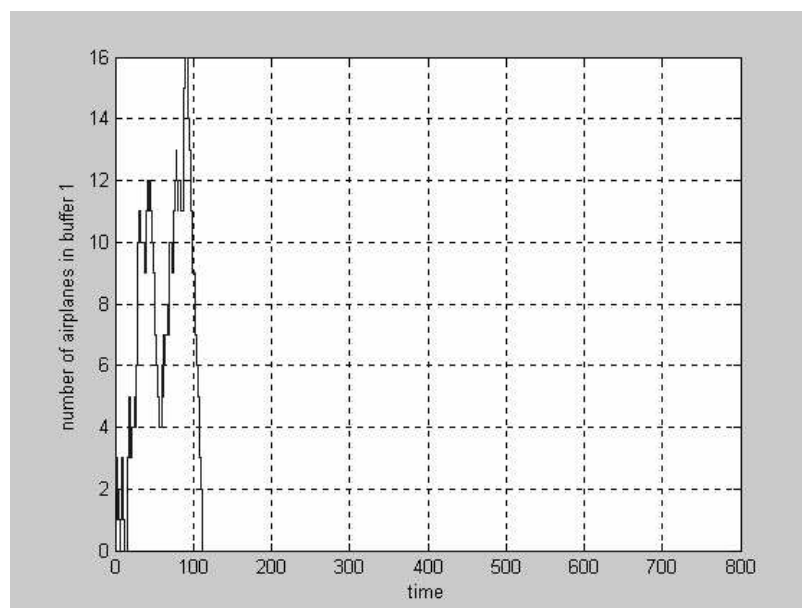
Στο χρονοδιάγραμμα 4.4.1 παρουσιάζονται οι χρονικές στιγμές των αφίξεων των αεροπλάνων.



χρονοδιάγραμμα 4.4.1

Το 4.4.2 περιγράφει την κίνηση των αεροπλάνων στον buffer 1 η οποία χαρακτηρίζεται από μεγάλη συσσώρευση αεροπλάνων που περιμένουν να εξυπηρετηθούν από τον διάδρομο προσγείωσης-απογείωσης. Πρώτον αυτό μπορεί να σημαίνει ότι έχουμε αφίξεις άνω του ενός αεροπλάνου την ίδια χρονική στιγμή γεγονός φυσικό για το συγκεκριμένο παράδειγμα. Δεύτερον μπορεί να σημαίνει ότι δίνεται προτεραιότητα για απογειώσεις αεροπλάνων και όχι για προσγειώσεις. Το δεύτερο ισχύει μόνο όταν ξεπεραστεί το όριο των αεροπλάνων που περιμένουν για απογείωση (στην συγκεκριμένη περίπτωση έχει τεθεί στα 2 αεροπλάνα) με τη προϋπόθεση ότι και τα αεροπλάνα που βρίσκονται στον αέρα και ζητούν άδεια για

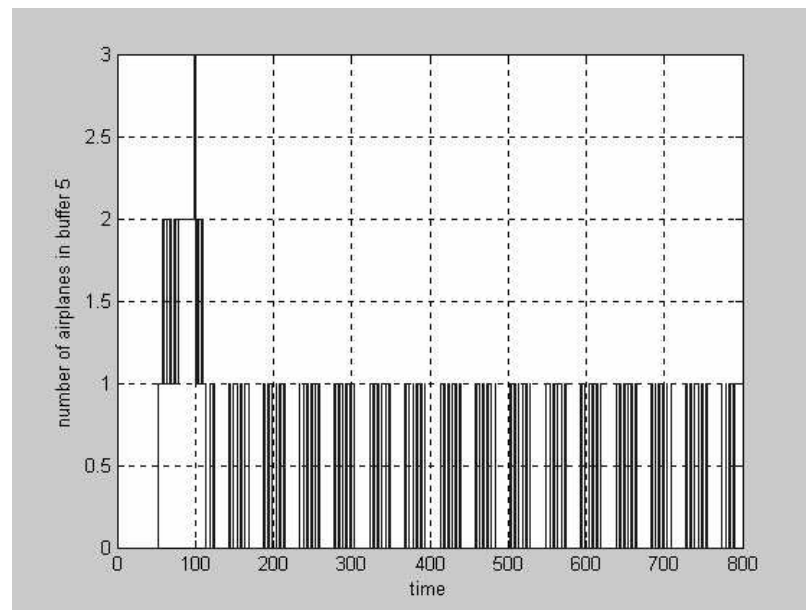
απογείωση δεν έχουν ξεπεράσει κάποιο όριο (στην συγκεκριμένη περίπτωση έχει τεθεί στα 10 αεροπλάνα). Με βάση όλα αυτά όσον αφορά το χρονοδιάγραμμα 4.4.2 μέχρι τα σκαλοπάτια να φθάσουν την τιμή 10, η προτεραιότητα δίνεται σε αεροπλάνα που πιθανόν να θέλουν να απογειωθούν. Από την στιγμή όμως που ξεπερνούν τα 10 (φθάνουν ακόμα και τα 16) η προτεραιότητα δίδεται σε αυτά μέχρι να ξαναφθάσουν την οριακή αυτή τιμή.



χρονοδιάγραμμα 4.4.2

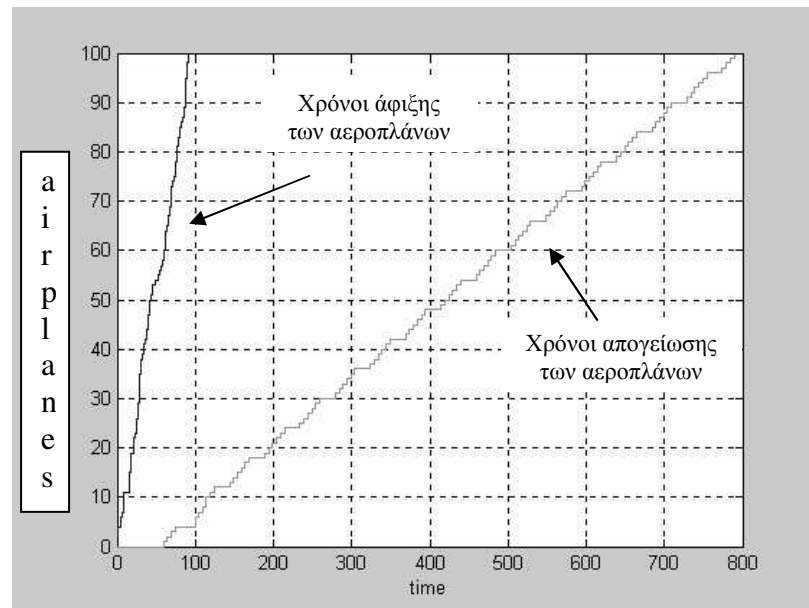
Άλλο ένα χαρακτηριστικό χρονοδιάγραμμα για την περίπτωση που εξετάζουμε είναι το 4.4.3 που περιγράφει την κίνηση των αεροπλάνων στον buffer 5 η οποία χαρακτηρίζεται και αυτή από κάποια συσσώρευση, μικρότερη από αυτή που περιγράφηκε προηγουμένως, αεροπλάνων που περιμένουν να εξυπηρετηθούν από τον διάδρομο προσγείωσης-απογείωσης και πιο συγκεκριμένα για απογείωση. Εδώ η συμφόρηση άνω του ενός αεροπλάνων οφείλεται μόνο στην χρήση του διαδρόμου απογείωσης και για τις προσγειώσεις των αεροπλάνων. Σύμφωνα με τα όρια που περιγράφηκαν, έχουμε αύξηση άνω των δύο αεροπλάνων που περιμένουν για απογείωση μόνο στην περίπτωση όπου τα αεροπλάνα που βρίσκονται στον αέρα ξεπεράσουν τα 10. Κάτι τέτοιο συμβαίνει και στην περίπτωσή μας όταν τα σκαλοπάτια στο 4.4.3 φθάνουν τα τρία. Σε γενικές γραμμές αν συγκρίνουμε τα 4.4.2 και 4.4.3 καταλήγουμε στο ότι με τα συγκεκριμένα όρια που έχουν επιλεχθεί, ευνοούνται τα αεροπλάνα που απογειώνονται. Κάτι τέτοιο έχει κάποια λογική βάση στήριξης αν σκεφτούμε ότι στον εναέριο χώρο του αεροδρομίου δεν έχουμε τις

χωροταξικές πιέσεις λόγω συσσώρευσης αεροπλάνων που υπάρχουν στο αεροδρόμιο σε επίπεδο εδάφους .



χρονοδιάγραμμα 4.4.3

Στο χρονοδιάγραμμα 4.4.4 φαίνονται μαζί οι αφίξεις των αεροπλάνων και οι απογειώσεις τους από το αεροδρόμιο για την συγκεκριμένη περίπτωση που εξετάσαμε. Ξανά όπως και στο δεύτερο παράδειγμα παρατηρούμε ότι οι απογειώσεις γίνονται σε κύματα με κάποια περιοδικότητα. Επίσης υπάρχει μεγάλη διαφορά στον χρόνο προσγείωσης του τελευταίου αεροπλάνου από τα 100 και στον χρόνο που αυτό τελικά απογειώνεται από το αεροδρόμιο. Αυτό μας αποδεικνύει την αδυναμία του συγκεκριμένου αεροδρομίου να αντιμετωπίσει με απόδοση την κατάσταση του παραδείγματος αυτού, γεγονός που έχει ήδη επισημανθεί.



χρονοδιάγραμμα 4.4.4

Κεφάλαιο 5

Συμπεράσματα □ Παρατηρήσεις

Στην διπλωματική αυτή αναπτύχθηκε μία γλώσσα προσομοίωσης για την εξυπηρέτηση αεροπλάνων σε κομβικά αεροδρόμια. Το σύστημα ενός αεροδρομίου θεωρήθηκε ως ένα σύστημα διακριτών συμβάντων που λαμβάνουν χώρα σε συγκεκριμένες χρονικές στιγμές και κατά αυτόν τον τρόπο αντιμετωπίστηκε για την προσομοίωσή του. Η γλώσσα αναπτύχθηκε σε κώδικα MATLAB που σημαίνει ότι είναι ευμετάβλητη και άρα εφαρμόσιμη και για άλλα αεροδρόμια που τυχόν να θέλει κάποιος αναλυτής να προσομοιώσει, κάνοντας πάντα τις κατάλληλες αλλαγές σε αυτήν. Αυτό είναι δυνατό, αφού η γλώσσα προσομοίωσης διαχωρίζει τα βασικά μέρη ενός αεροδρομίου σε στοιχειώδη συστήματα και υποσυστήματα τα οποία μπορούν να περιγράψουν διαφορετικά αεροδρόμια με την ανάλογη συνένωση τους. Τα μέρη αυτά είναι ο διάδρομος προσγείωσης, ο διάδρομος απογείωσης, ο χώρος στάθμευσης των αεροπλάνων στο αεροδρόμιο και οι διάφοροι διάδρομοι τροχοδρόμησης που ενώνουν τα παραπάνω μέρη του αεροδρομίου. Αυτά λοιπόν αποτελούν και τα στοιχειώδη συστήματα ή υποσυστήματα που πλαισιώνουν ένα αεροδρόμιο. Για την περιγραφή ενός αεροδρομίου μπορούμε να έχουμε οποιοδήποτε αριθμό από καθένα από αυτά με οποιαδήποτε μεταξύ τους σύνδεση.

Έχοντας δημιουργήσει μια τέτοια γλώσσα προσομοίωσης για ένα αεροδρόμιο μπορούμε να την χρησιμοποιήσουμε πρώτον για επίβλεψη της πραγματικής λειτουργίας του θέτοντας ως είσοδο πραγματικές αφίξεις αεροπλάνων και με τον τρόπο αυτό να ενημερωνόμαστε έγκαιρα για τυχόν καθυστερήσεις αεροπλάνων που προκαλούνται από χωροταξικές ελλείψεις του συγκεκριμένου αεροδρομίου. Δεύτερον για ένα νέο αεροδρόμιο το οποίο ακόμα δεν έχει κατασκευαστεί μπορούμε να ελέγξουμε αν οι χώροι του, που έχουν σχεδιαστεί προς κατασκευή, καλύπτουν τις ανάγκες εξυπηρέτησης των αεροπλάνων που πρόκειται να δέχεται το εν λόγω αεροδρόμιο. Εδώ θεωρούμε ότι κατά την κατασκευή ενός νέου αεροδρομίου υπάρχει και η γνώση του πόσο κομβικό θα είναι αυτό, δηλαδή ότι γνωρίζουμε την πυκνότητα αφίξεων των αεροπλάνων. Τρίτον μια άλλη εφαρμογή αυτής της γλώσσας προσομοίωσης είναι και ο έλεγχος των επιπτώσεων που θα είχαν διάφορες αλλαγές στον χώρο ενός αεροδρομίου που εξυπηρετούνται τα αεροπλάνα. Αυτές οι αλλαγές έχουν να κάνουν είτε με την χωροταξία του αεροδρομίου ή απλώς

με αλλαγές στον τρόπο εξυπηρέτησης των αεροπλάνων. Για παράδειγμα μια αλλαγή στην χωροταξία θα ήταν η δημιουργία επιπλέον χώρων στάθμευσης των αεροπλάνων και μια αλλαγή στον τρόπο εξυπηρέτησης θα ήταν η αυξομείωση του χρόνου εξυπηρέτησης ενός αεροπλάνου σε κάποιο στοιχειώδες σύστημα ή ακόμα και η μετατροπή διάφορων προτεραιοτήτων που δίνονται στα αεροπλάνα για χρήση κάποιων χώρων του αεροδρομίου. Από τα παραπάνω διαφαίνεται και ο χαρακτήρας της προσομοίωσης ενός συστήματος που είναι το βασικό ερώτημα του τι επιπτώσεις θα έχουμε στο σύστημα του αεροδρομίου που προσομοιώνεται κάτω από μια συγκεκριμένη αλλαγή στην κατάσταση του, «Τι θα συμβεί αν ... ;»

Για επίδειξη της γλώσσας προσομοίωσης για την εξυπηρέτηση αεροπλάνων σε κομβικά αεροδρόμια επιλέχθηκε το αεροδρόμιο Χανίων «Κ. Δασκαλογιάννης». Στην πραγματικότητα το συγκεκριμένο αεροδρόμιο μπορεί να μην είναι κομβικό, αλλά αν κατά την χρήση της γεννήτριας τυχαίων αφίξεων επιλέξουμε να έχουμε μεγάλη πυκνότητα αφίξεων αεροπλάνων μπορούμε αμέσως να το μετατρέψουμε σε κομβικό καλώντας το τελικά να εξυπηρετήσει έναν μεγάλο αριθμό αεροπλάνων. Τα παραδείγματα δύο και τέσσερα του κεφαλαίου τέσσερα καλύπτουν τέτοιες καταστάσεις, κατά τις οποίες έχουμε υψηλό ρυθμό αφίξεων. Χαρακτηριστικό σε αυτά τα παραδείγματα είναι ότι οι απογειώσεις γίνονται κατά κύματα ανάλογου αριθμού αεροπλάνων με αυτόν των θέσεων στάθμευσης. Για την συγκεκριμένη εφαρμογή αυτός ο αριθμός είναι της τάξης του έξι. Τα κύματα αυτά έχουν μια περιοδικότητα 9σχετική με την διάρκεια της εξυπηρέτησης των αεροπλάνων από τις θέσεις στάθμευσης και φαίνονται παραστατικά στα χρονοδιαγράμματα 4.2.12 και 4.4.4.

Μετά από ένα πλήθος δοκιμών για διάφορες αλλαγές στο σύστημα του αεροδρομίου Χανίων σε επίπεδο του τρόπου εξυπηρέτησης των αεροπλάνων σε αυτό βγήκε το συμπέρασμα ότι μια αλλαγή που γινόταν σε κάποιο στοιχειώδες σύστημα με σκοπό την καλύτερη απόδοση του, οδηγούσε ταυτόχρονα σε μείωση της λειτουργικότητας κάποιου άλλου στοιχειώδους συστήματος. Για παράδειγμα, αν με σκοπό την μείωση της πιθανότητας αναμονής αεροπλάνων πριν από τις θέσεις στάθμευσης μειώσουμε τον χρόνο εξυπηρέτησης κατά την στάθμευση ,τότε παράλληλα θα αυξηθεί η συμφόρηση των αεροπλάνων στον μοναδικό διάδρομο προσγείωσης-απογείωσης αφού θα συγκεντρώνονται σε μικρότερο χρονικό διάστημα περισσότερα αεροπλάνα που θα ζητούν να απογειωθούν. Παρόλα αυτά ένα σημείο όπου μπορεί να βρει πρακτική εφαρμογή η βελτιστοποίηση μέσω αλλαγών σε επίπεδο του τρόπου εξυπηρέτησης είναι αυτό όπου ορίζονται οι προτεραιότητα χρήσης του

διάδρομου προσγείωσης-απογείωσης από αεροπλάνα είτε για απογείωση ή για προσγείωση τους. Αυτό συμβαίνει επειδή τα αεροπλάνα που βρίσκονται στον αέρα δεν επιβαρύνουν άμεσα τον χώρο του αεροδρομίου σε επίπεδο εδάφους. Αντίθετα, τα αεροπλάνα που μόλις έχουν προσγειωθεί όπως και αυτά που είναι έτοιμα για απογείωση, τον επιβαρύνουν άμεσα. Έτσι για την προτεραιότητα στον διάδρομο απογείωσης-προσγείωσης έχουν ευνοηθεί τα αεροπλάνα που απογειώνονται ,επειδή ταυτόχρονα σημαίνουν και απελευθέρωση πόρων του αεροδρομίου, εις βάρος των αεροπλάνων που ζητούν να προσγειωθούν πράγμα που σημαίνει επιβάρυνση των πόρων του αεροδρομίου. Φυσικά σε αυτή την επιβάρυνση των αεροπλάνων που είναι έτοιμα για προσγείωση θέτεται και κάποιο όριο που όταν ξεπεραστεί η προτεραιότητα χρήσης του διαδρόμου δίδεται σε αυτά. Για τα παραδείγματα του κεφαλαίου τέσσερα το όριο στον αριθμό αεροπλάνων που περιμένουν για απογείωση είναι τα δυο ενώ το όριο στον αριθμό αεροπλάνων που περιμένουν για προσγείωση είναι τα δέκα.

Τα αποτελέσματα από δοκιμές αλλαγών στο αεροδρόμιο σε επίπεδο χωροταξίας με συνδυασμό αλλαγών και στον τρόπο εξυπηρέτησης των αεροπλάνων σαφώς και δίνει πολύ αποδοτικότερα αποτελέσματα για την λειτουργία του. Θα πρέπει όμως τέτοιες αλλαγές να γίνονται με μεθοδικό τρόπο και κάθε φορά να ελέγχεται η λειτουργία του αεροδρομίου μέσα από τα χρονοδιαγράμματα της προσομοίωσης γιατί υπάρχει ο κίνδυνος να μην έχουμε την αναμενόμενη αποδοτικότητα από τις συγκεκριμένες αλλαγές. Για παράδειγμα αν για το αεροδρόμιο Χανίων, θεωρώντας το ως κομβικό με τον τρόπο που περιγράφηκε παραπάνω, δημιουργηθεί ένας επιπλέον διάδρομος που θα εξυπηρετεί τις απογειώσεις και με τον ήδη υπάρχον διάδρομο να εξυπηρετεί τις προσγειώσεις τότε μέσα από μια προσομοίωση και τα χρονοδιαγράμματά της παρατηρούμε ότι η αποσυμφόρηση των αεροπλάνων που περιμένουν για απογείωση ή προσγείωση είναι δραστική. Όμως παρατηρείται επιπλέον ότι σε βάθος χρόνου οι απογειώσεις των αεροπλάνων τελικά θα συμβαίνουν χωρίς καμία ουσιαστική χρονική διαφορά από το αν είχαμε μόνο ένα διάδρομο προσγείωσης-απογείωσης. Από αυτό καταλήγουμε στο συμπέρασμα ότι για να έχουμε την μέγιστη αποδοτικότητα από ένα επιπλέον διάδρομο για απογειώσεις θα πρέπει να αυξηθούν και οι θέσεις στάθμευσης των αεροπλάνων. Ο κατάλληλος αριθμός της αύξησης των θέσεων αυτών μπορεί να επιλεγεί μέσα από μια σειρά προσομοιώσεων.

Από όλα αυτά φαίνεται και η αξία της προσομοίωσης για ένα σύστημα αεροδρομίου ,πέρα από την απλή επίβλεψη της λειτουργίας του που οδηγεί στην πρόβλεψη της κίνησης ενός αριθμού αφικνούμενων αεροπλάνων μέσα σε αυτό.

Παράρτημα

Κώδικας της γλώσσας προσομοίωσης για το αεροδρόμιο Χανίων

Στο παράρτημα παραθέτουμε τον κώδικα σε γλώσσα προγραμματισμού MATLAB, με αναλυτικά σχόλια ,που χρησιμοποιήθηκε στην ανάπτυξη της γλώσσας προσομοίωσης για εξυπηρέτηση αεροπλάνων ειδικά για το αεροδρόμιο "Κ. Δασκαλογιάννης".

Κυρίως ρουτίνα airport

```
function airport

n=100;           %number of desired airplane arrivals
t_total=100;    %period of time arrivals occur (1440=one day)

%call poisson function
[total_arrivals,A,i,total] = poisson(n,t_total);

%delay time in each elementary-system-subsystem A,B,C,D
T1=1;   %A1   landing
T2=2;   %B1   taxiing to service
T3=45;  %C1-C6 service
T4=5;   %D1   taxiing from service
T5=1;   %A2   take-off

b1max=10; %maximum airplanes queued in b1
b5max=2;  %maximum airplanes queued in b5

same_runway=1; %1 denote one runway for landing-take_off
               %0 denote one runway for landing and one for take_off

delay1=0; %initialize parameters in elementary-subsystem A1
b1=0;
p1=0;
machine1=1;

delay2=0; %initialize parameters in elementary-system B1
b2=0;
p2=0;
m2=0;

delay3=0; %initialize parameters in elementary-system C
b3=0;
p3=0;
b3act=0;
```

```

places=6; %number of elementary-subsystems in elementary-system C
j=0;
service=0;
for j=1:places
    service(j)=0;
end;
counter=1;
free=1;
in_service=0;

delay4=0; %initialize parameters in elementary-system D1
b4=0;
p4=0;
m4=0;

delay5=0; %initialize parameters in elementary-subsystem A2
b5=0;
p5=0;
machine5=0;

buffer6=0; %initialize parameters for airport_output
b6=0;

t=0; %initialize time

while p5<total_arrivals %main while loop calling sub-functions for every time unit

    [delay1,b5,delay5,p5,machine5] =
    priority(b5,p5,delay5,b1,delay1,T5,T1,machine5,b1max,b5max,same_runway,t);

    [b1,delay1,p1,machine1] = landing(A,i,T1,delay1,b1,p1,machine1,t);
    buf1(t+1)=p1;
    buffer1(t+1)=b1;
    machine1(t+1)=0;

    bin=b2;
    delay=delay2;
    T=T2;
    buf=buf1;
    p=p2;
    m=m2;
    [bin,delay,p,m] = taxiing(bin,delay,T,buf,p,t,m);
    b2=bin;
    delay2=delay;
    p2=p;
    m2=m;
    buffer2(t+1)=b2;
    buf2(t+1)=p2;
    machine2(t+1)=m2;

```

```
[b3,b3act,delay3,p3,service,counter,in_service,free] =  
service(b3,b3act,delay3,T3,buf2,p3,t,service,counter,in_service,free,places);  
buffer3(t+1)=b3;  
buffer3act(t+1)=b3act;  
buf3(t+1)=p3;  
if service(1)>0 s1=1;elseif service(1)==0 s1=0;end  
if service(2)>0 s2=1;elseif service(2)==0 s2=0;end  
if service(3)>0 s3=1;elseif service(3)==0 s3=0;end  
if service(4)>0 s4=1;elseif service(4)==0 s4=0;end  
if service(5)>0 s5=1;elseif service(5)==0 s5=0;end  
if service(6)>0 s6=1;elseif service(6)==0 s6=0;end  
serv1(t+1)=s1;serv2(t+1)=s2;serv3(t+1)=s3;  
serv4(t+1)=s4;serv5(t+1)=s5;serv6(t+1)=s6;  
  
bin=b4;  
delay=delay4;  
T=T4;  
buf=buf3;  
p=p4;  
m=m4;  
[bin,delay,p,m] = taxiing(bin,delay,T,buf,p,t,m);  
b4=bin;  
delay4=delay;  
p4=p;  
m4=m;  
buffer4(t+1)=b4;  
buf4(t+1)=p4;  
machine4(t+1)=m4;  
  
[b5,delay5,p5,machine5] =  
take_off(T5,delay5,b5,p5,machine5,buf4,buffer1,same_runway,t);  
buffer5(t+1)=b5;  
buf5(t+1)=p5;  
machine5(t+1)=0;  
  
[buffer6,t] = airport_output(buffer6,buf5,t);  
  
t=t+1; %advance time one unit  
end  
  
%graphical representation of airplane movements in time  
figure;  
stairs(0:i-1,total(1:i),'b');  
xlabel('time');  
ylabel('number of airplanes');  
grid on;
```



```

figure;
stairs(0:t-1,buffer1(1:t),'b');
xlabel('time');
ylabel('number of airplanes in buffer 1');
grid on;
figure;
stairs(0:t-1,machine1(1:t),'m');
xlabel('time');
ylabel('state of elementary subsystem A1');
grid on;

figure;
stairs(0:t-1,buffer2(1:t),'b');
xlabel('time');
ylabel('number of airplanes in buffer 2');
grid on;
figure;
stairs(0:t-1,machine2(1:t),'m');
xlabel('time');
ylabel('state of elementary system B1');
grid on;

figure;
stairs(0:t-1,buffer3act(1:t),'b');
xlabel('time');
ylabel('number of airplanes in buffer 3');
grid on;
figure;
subplot(6,1,1);stairs(0:t-1,serv1(1:t),'m');
subplot(6,1,2);stairs(0:t-1,serv2(1:t),'m');
subplot(6,1,3);stairs(0:t-1,serv3(1:t),'m');
subplot(6,1,4);stairs(0:t-1,serv4(1:t),'m');
subplot(6,1,5);stairs(0:t-1,serv5(1:t),'m');
subplot(6,1,6);stairs(0:t-1,serv6(1:t),'m');
xlabel('time');
ylabel('state of elementary subsystems C1-C6');

figure;
stairs(0:t-1,buffer4(1:t),'b');
xlabel('time');
ylabel('number of airplanes in buffer 4');
grid on;
figure;
stairs(0:t-1,machine4(1:t),'m');
xlabel('time');
ylabel('state of elementary system D1');
grid on;

```

```
figure;  
stairs(0:t-1,buffer5(1:t),'b');  
xlabel('time');  
ylabel('number of airplanes in buffer 5');  
grid on;  
figure;  
stairs(0:t-1,machine5(1:t),'m');  
xlabel('time');  
ylabel('state of elementary subsystem A2');  
grid on;
```

```
figure;  
stairs(0:t-1,buffer6(1:t),'b');  
xlabel('time');  
ylabel('number of airplanes in buffer 6');  
grid on;
```

```
figure;  
stairs(0:i-1,total(1:i),'b');  
hold on;  
stairs(0:t-1,buffer6(1:t),'g');  
xlabel('time');  
ylabel('airplanes landing and take_off');  
grid on;
```

```
figure;  
stairs(0:t-1,machine5(1:t),'b');  
hold on;  
stairs(0:t-1,machine1(1:t),'m');  
grid on;
```

Υπορουτίνα priority

```
% priority: A function to simulate the priority between landing
%           and take_off if same runway
%
% Usage: [delay1,b5,delay5,p5,machine5] =
%         priority(b5,p5,delay5,b1,delay1,T5,T1,machine5,b1max,b5max,t)
%
% Arguments:b5 - input buffer for take_off
%            p5 - output buffer for take_off
%            delay5 - delay in elementary subsystem A2
%            b1 - input buffer for landing
%            delay1 - delay in elementary subsystem A2
%            T5 - accommodation time in A2
%            T1 - accommodation time in A1
%            machine5 - sign of A2 functionality
%            b1max - maximum airplanes queued in b1
%            b5max - maximum airplanes queued in b5
%            t - time unit
%
% Returns:delay1 - new delay in elementary subsystem A1
%          b5      - new input buffer for take_off
%          delay5  - new delay in elementary subsystem A2
%          p5      - new output buffer for take_off
%          machine5 - sign of A2 functionality

function [delay1,b5,delay5,p5,machine5] =
priority(b5,p5,delay5,b1,delay1,T5,T1,machine5,b1max,b5max,same_runway,t);

if same_runway==1
    if b5>=b5max & b1<b1max
        b5=b5-1;
        machine5(t)=1;
        if (b5>0)
            delay5=t+T5;
        end
        p5=p5+1;
        delay1=t+T1;
    end
end
end
```

Υπορουτίνα landing

```
% landing: A function to simulate the landing process of an airplane
%
% Usage: [b1,delay1,p1,machine1] = sub1(A,i,T1,delay1,b1,p1,machine1,t)
%
% Arguments: A,i - outputs from poisson function
%             T1 - accommodation time in A1
%             delay1 - delay in elementary subsystem A1
%             b1 - input buffer for landing
%             p1 - output buffer for landing
%             machine1 - sign of A1 functionality
%             t - time unit
%
% Returns: delay1 - new delay in elementary subsystem A1
%          b1 - new input buffer for landing
%          p1 - new output buffer for landing
%          machine1 - sign of A1 functionality

function [b1,delay1,p1,machine1] = landing(A,i,T1,delay1,b1,p1,machine1,t);

if (t<i) & (A(t+1)>0)
    if (b1==0)
        delay1=t+T1;
    end
    b1=b1+A(t+1);
end
if (t==delay1) & (b1>0)
    machine1(t)=1;
    b1=b1-1;
    if (b1>0)
        delay1=t+T1;
    end
    p1=p1+1;
end
```

Υπορουτίνα taxiing

```
% taxiing: A function to simulate the various taxiing processes of an
%          airplane in the enviroment of an airport
%
% Usage: [bin,delay,p,m] = sub(bin,delay,T,buf,p,t,m);
%
% Arguments:bin - input buffer for elementary system taxiing
%            delay - delay in elementary system taxiing
%            T - accommodation time in taxiing
%            buf - output buffer of preceding elementary system
%            p - output buffer for elementary system taxiing
%            t - time unit
%            m - sign of taxiing functionality
%
% Returns:bin - new input buffer for elementary system taxiing
%          delay - new delay in elementary system taxiing
%          p - new output buffer for elementary system taxiing
%          m - sign of taxiing functionality

function [bin,delay,p,m] = taxiing(bin,delay,T,buf,p,t,m);

if t>0
    if [buf(t+1)]-[buf(t)]>0
        if bin==0
            delay=t+T;
            m=m+1;
        end
        bin=bin+1;
    end
    if (t==delay) & (bin>0)
        bin=bin-1;
        m=m-1;
        if bin>0
            delay=t+T;
            m=m+1;
        end
        p=p+1;
    end
end
end
```

Υπορουτίνα service

```
% service: A function to simulate the service process of airplanes
%         according to the number of service places
%
% Usage: function [b3,b3act,delay3,p3,t,service,counter,in_service,free] =
%         sub3(b3,b3act,delay3,T3,buf2,p3,t,service,counter,in_service,free,places);
%
% Arguments: b3 - input buffer for elementary system C (only airplanes waiting)
%            b3act - input buffer for elementary system C (airplane both waiting
%                  and taking service)
%            delay3 - delay in elementary subsystems of C
%            p3 - output buffer for elementary system service
%            t - time unit
%            service - matrix containing completion times for every occupied place
%            counter - points to the elementary subsystem of interest initialized to 1
%            in_service - number of airplanes taking service initialized to 0
%            free - next place to make free initialized to 0
%            places - number of service places
%
% Returns: updated variables and matrixes of inputs  b3, b3act, delay3, p3, service,
%           counter, in_service, free
```

```
function [b3,b3act,delay3,p3,service,counter,in_service,free] =
service(b3,b3act,delay3,T3,buf2,p3,t,service,counter,in_service,free,places);
```

```
if t>0
    if buf2(t+1)-buf2(t)>0
        if service(counter)==0
            service(counter)=t+T3;
            counter=mod(counter,places)+1;
            in_service=in_service+1;
            if in_service==1
                delay3=t+T3;
            end
        else
            b3=b3+1;
        end
        b3act=b3act+1;
    end

    if (t==delay3) & (service(free)>0)
        service(free)=0;
        free=mod(free,places)+1;
        delay3=service(free);
        if (b3>0)
            service(counter)=t+T3;
            counter=mod(counter,places)+1;
            b3=b3-1;
        end
    end
end
```

```
        else
            in_service=in_service-1;
        end
        b3act=b3act-1;
        p3=p3+1;
    end

    if service(1)>0
        s1=1;
    elseif service(1)==0
        s1=0;
    end
    if service(2)>0
        s2=1;
    elseif service(2)==0
        s2=0;
    end
    if service(3)>0
        s3=1;
    elseif service(3)==0
        s3=0;
    end
    if service(4)>0
        s4=1;
    elseif service(4)==0
        s4=0;
    end
    if service(5)>0
        s5=1;
    elseif service(5)==0
        s5=0;
    end
    if service(6)>0
        s6=1;
    elseif service(6)==0
        s6=0;
    end
end
```

Υπορουτίνα take_off

```
% take_off: A function to simulate the take-off process of an airplane
%
% Usage: [b5,delay5,p5,machine5] =
%         sub5(T5,delay5,b5,p5,machine5,buf4,buffer1,t)
%
% Arguments: T5 - accommodation time in A2
%            delay5 - delay in elementary subsystem A2
%            b5 - input buffer for take_off
%            p5 - output buffer for take_off
%            machine5 - sign of A2 functionality
%            buf4 - output of preceding elementary system
%            buffer1 - input for landing over time
%            t - time unit
%
% Returns: b5 - new input buffer for take_off
%          delay5 - new delay in elementary subsystem A2
%          p5 - new output buffer for take_off
%          machine5 - sign of A2 functionality
```

```
function [b5,delay5,p5,machine5] =
take_off(T5,delay5,b5,p5,machine5,buf4,buffer1,same_runway,t);
```

```
if t>0
    if [buf4(t+1)]-[buf4(t)]>0
        if b5==0
            delay5=t+T5;
        end
        b5=b5+1;
    end
    if (t==delay5) & (b5>0)
        if same_runway==1
            if buffer1(t)==0
                machine5(t)=1;
                b5=b5-1;
                if (b5>0)
                    delay5=t+T5;
                end
                p5=p5+1;
            end
            if buffer1(t)>0
                delay5=t+T5;
            end
        end
    else
        machine5(t)=1;
        b5=b5-1;
        if (b5>0)
            delay5=t+T5;
        end
    end
end
```



```
        end
        p5=p5+1;
    end
end
end
```

%used only for airport system output graphical representation

```
function [buffer6,t] = airport_output(buffer6,buf5,t);
```

Βιβλιογραφία

- [1]<http://web.mit.edu/aeroastro/www/labs/AATT/reviews.html>
- [2]Udo W. Pooch and James A. Wall, “Discrete event simulation”, CPV Press, Inc., 1993
- [3]George S. Fishman, “Concepts and methods in discrete event simulation”, Wiley, Inc., 1994
- [4]Christos G. Cassandras, “Discrete event systems (modeling and performance analysis)”, Richard D. Irwin & Aksen Associates, Inc., 1993
- [5]Behrokh Khoshnevis, “Discrete systems simulation”, McGraw-Hill, Inc., 1994
- [6]Μάνος Ρουμελιώτης, “Τεχνικές προσομοίωσης”, Παρατηρητής, Θεσσαλονίκη, 1998
- [7]Kari Anderson, Francis Carr, Eric Feron and William D. Hall, “Analysis and modeling of ground operations at hub airports”, 3rd USA/Europe Air Traffic Management R&D Seminar, Napoli, 2000
- [8]<http://www.hcaa-eleng.gr/hania.htm>