

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ  
τμήμα  
Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΕ  
ΜΙΚΡΟΠΕΞΕΡΓΑΣΤΗ ΤΗΛΕΧΕΙΡΙΖΟΜΕΝΟΥ ΕΛΕΓΚΤΗ  
ΑΣΑΦΟΥΣ ΛΟΓΙΚΗΣ ΓΙΑ ΤΡΟΦΟΔΟΣΙΑ  
ΙΧΘΥΟΚΑΛΛΙΕΡΓΕΙΑΣ**

ΜΩΡΑΪΤΗΣ ΑΝΑΣΤΑΣΙΟΣ  
2006

## Ευχαριστίες

Ευχαριστώ πολύ τον επιβλέποντα καθηγητή μου κ. Γεώργιο Σταυρακάκη για την εμπιστοσύνη που έδειξε στο πρόσωπο μου και για την βοήθεια του καθ' όλη τη διάρκεια της διπλωματικής εργασίας μου. Ευχαριστώ πολύ τον αναπληρωτή καθηγητή κ. Εμμανουήλ Αντωνιδάκη για την καθοδήγησή που μου παρείχε κατά τη διάρκεια της συνεργασίας μας, που χωρίς αυτή δεν θα είχα ποτέ καταφέρει να εντρυφήσω στο θέμα με το οποίο ασχολήθηκα. Ευχαριστώ τον καθηγητή κ. Ματτία Μπούχερ για την ανάγνωση της διπλωματικής μου και τα σχόλιά του. Ακόμα να ευχαριστήσω τον Δρ. Ερευνητή Νικόλαο Παπανδρουλάκη για την ουσιαστική καθοδήγηση του και την εμπιστοσύνη του. Επίσης τον Ηρακλή Ρηγάκη που ήταν πάντα διαθέσιμος και πρόθυμος να βοηθήσει σε όποιο μικρό ή μεγάλο πρόβλημα παρουσιαζόταν.

Τέλος θέλω να ευχαριστήσω τους φίλους μου για την συμπαράσταση τους και για τα έξι απίστευτα χρόνια που περάσαμε μαζί :) , και ασφαλώς την οικογένεια μου που ήταν πάντα δίπλα μου, χωρίς αυτήν δε θα έγγραφα αυτές τις γραμμές...

*Μωραΐτης Αναστάσιος*

## ΠΕΡΙΛΗΨΗ

Η Ιχθυοκαλλιέργεια έχει αναδειχθεί σε μία από τις πιο ραγδαία εξελισσόμενες μορφές επιχειρηματικότητας στον ελληνικό χώρο. Είναι γεγονός ότι αποτελεί τον τρίτο μεγαλύτερο τομέα εξαγωγής και έχει ισχυρές προοπτικές. Η ενσωμάτωση της τεχνολογίας στις πρακτικές της έχει ήδη εξαιρετικά αποτελέσματα. Βασικός παράγοντας στην μεγιστοποίηση της απόδοσης των ιχθυοκαλλιεργειών είναι η σωστή διαχείριση της παρεχόμενης τροφής και αποτελεί ακόμα ανοιχτό πρόβλημα. Η εργασία αυτή διερευνά την εφαρμογή της ασαφούς λογικής στην εκτίμηση της ποσότητας της παρεχόμενης τροφής. Υλοποιούνται τρεις αλγόριθμοι πάνω σε σύστημα, βασισμένο σε μικροεπεξεργαστή, το οποίο είναι σχεδιασμένο να λειτουργεί αυτόνομα. Το υλοποιηθέν σύστημα αξιοποιεί GSM λειτουργικότητα για επιπρόσθετη ευελιξία και ανταλλαγή δεδομένων με ένα απομακρυσμένο τερματικό.

## ABSTRACT

Fish farming has emerged as one of the fastest growing technology driven Greek industries. It is in fact the country's third largest export sector and has a great potential for further development. The incorporation of technology has already brought exceptional results. Major performance factor in the aquaculture industry is the efficient management of food supplies, still an open project. This thesis explores the possible application of fuzzy logic in the food quantity estimation. Three algorithms are implemented on a microprocessor based system, which is designed for autonomous operation. The implemented system develops GSM functionality for additional flexibility and data exchange with a remote terminal.

## Περιεχόμενα:

	Σελίδα
1. Εισαγωγή.....	4
1.α Παροχή τροφής σύμφωνα με τη ζήτηση.....	6
1.β Περιγραφή των συστημάτων τροφοδοσίας σύμφωνα με τη ζήτηση.....	7
1.γ.Σχετικά με την συγκεκριμένη εργασία.....	8
2. Θεωρητικό και Πρακτικό Υπόβαθρο.....	10
3. Εισαγωγή στην Ασαφή Λογική (Fuzzy Logic).....	12
3.α Βασικές έννοιες.....	12
4. Περιγραφή του Συστήματος.....	21
4.α Περιγραφή Λειτουργίας.....	21
4.β Αλγόριθμοι Τροφοδοσίας.....	22
4.γ Επικοινωνία.....	29
5. Υλικό.....	33
5.α Το αναπτυξιακό .....	33
5.β Ο Μικροεπεξεργαστής M68HC912B32 .....	34
5.γ Το Κινητό Τηλέφωνο.....	40
5.δ Αυτοχειριζόμενος Μοχλός.....	42
6. Λογισμικό.....	45
6.α Εργαλεία.....	45
6.β Περιγραφή του Λογισμικού .....	46
7. Επίλογος.....	60
7.α Αξιολόγηση Συστήματος & Αποτελέσματα.....	60

7.α.1 Μοντέλο με κατανομή Gauss .....	61
7.α.2 Τυχαία κατανομή.....	64
7.α.3 Ένα πιο ρεαλιστικό σενάριο.....	69
7.β Περαιτέρω Εργασία.....	71

#### -Βιβλιογραφία

## 1.ΕΙΣΑΓΩΓΗ

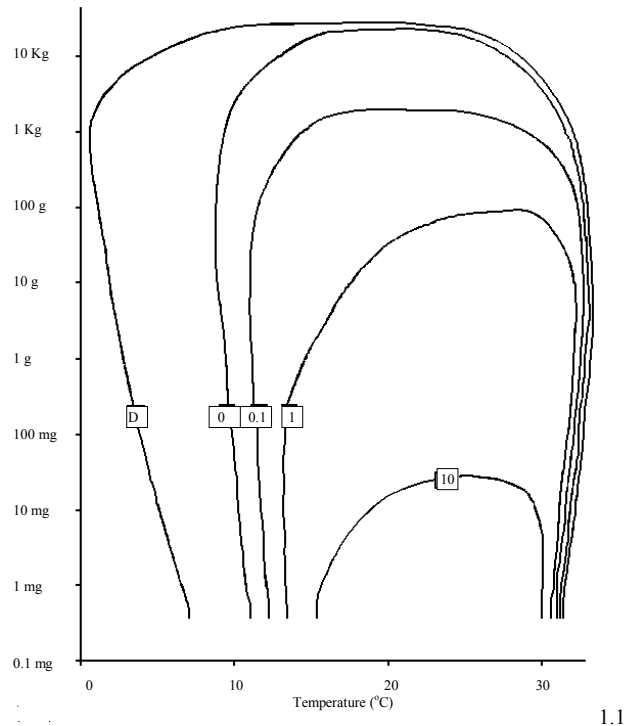
Η παροχή τροφής σε μια ιχθυοκαλλιέργεια αποτελεί το βασικό κόστος. Οι μέθοδοι που έχουν αναπτυχθεί για την ανατροφή ιχθύων είναι 3 και ξεχωρίζουν από τον τρόπο που γίνεται η παροχή τροφής. Ο πρώτος τρόπος είναι διανομή με το χέρι, μια παλιά μέθοδος αρκετά δαπανηρή και σε κόστος εργασίας αλλά και σε υλικό. Η δεύτερη και πιο δημοφιλής μέθοδος σήμερα είναι η χρησιμοποίηση αυτόματων ταϊστών που παρέχουν τροφή σύμφωνα με κάποιο προγραμματισμό(πχ. κάθε 2 ώρες). Όμως εδώ συναντάμε το μειονέκτημα της υπερτροφοδοσίας ή υποτροφοδοσίας. Η τελευταία μέθοδος που έχει αναπτυχθεί τα τελευταία χρόνια έχει εφαρμοσθεί με επιτυχία σε ερευνητικό επίπεδο αλλά όχι σε βιομηχανικό, παρότι τα αποτελέσματα είναι αισιόδοξα. Αυτή είναι η παροχή τροφής ανάλογα με την ζήτηση.

Οι δύο πρώτες μέθοδοι βασίζονται σε μοντέλα αύξησης που εξαρτώνται από την θερμοκρασία και το ατομικό βάρος των ατόμων. Όμως ο προσδιορισμός της ποσότητας αλλά και η συχνότητα χορήγησης της τροφής σε μονάδες παραγωγής αποτελούν ένα πάγιο πρόβλημα των καλλιεργητών διότι επηρεάζονται ταυτόχρονα από πολλούς παράγοντες (βιοτικούς και αβιοτικούς) που ρυθμίζουν την πρόσληψη τροφής από τα ψάρια. Τέτοιοι παράγοντες είναι η θερμοκρασία, το οξυγόνο, η ηλικία και το ατομικό βάρος των ψαριών, η δομή του πληθυσμού, η παρουσία ή όχι ασθενειών ή και στρες κλπ.

Η ημερησίως χορηγούμενη ποσότητα συνήθως εκφράζεται σαν ποσοστό του ατομικού βάρους, η εκτίμηση του οποίου γίνεται από μοντέλα ανάπτυξης που έχουν αναπτυχθεί είτε από τις εταιρείες παραγωγής ιχθυοτροφών είτε από τους παραγωγούς με βάση την εμπειρία διαχείρισης της μονάδας τους.

Παρά το γεγονός ότι το σύστημα είναι πολυπαραμετρικό, είναι δυνατόν να θεωρηθεί ότι οι κύριες μεταβλητές είναι η θερμοκρασία ο χρόνος και το ατομικό βάρος. Έτσι τα εφαρμοζόμενα μοντέλα είναι συνήθως της μορφής  $W_f = W_i \cdot \exp^{SGR \cdot t}$  όπου  $W_f$  το τελικό

αναμενόμενο βάρος,  $W_i$  το αρχικό βάρος, SGR ο ρυθμός αύξησης και  $t$  ο χρόνος. Ο ρυθμός αύξησης εξαρτάται από την θερμοκρασία εκτροφής και το ατομικό βάρος των ατόμων. Η μορφή της σχέσης τους για το λαβράκι φαίνεται στο παρακάτω διάγραμμα όπου παρουσιάζονται ισο-γραμμές σταθερού ρυθμού αύξησης στο χώρο που ορίζεται από την θερμοκρασία και το ατομικό βάρος σε λογαριθμική κλίμακα.



Η μορφή αυτή βασίστηκε στη συλλογή δεδομένων από πειραματικές εκτροφές και εμπειρική προσαρμογή της καμπύλης (Divanach.....). Η αδυναμία να παρουσιαστεί η σχέση αυτή υπό μορφή κλειστής μαθηματικής έκφρασης οδήγησε στον προσδιορισμό πινάκων διατροφής βάση των οποίων καθορίζεται η ημερήσια ποσότητα σαν συνάρτηση της θερμοκρασίας και του ατομικού βάρους. Η έως σήμερα ερευνητική εμπειρία για την εξέλιξη συστημάτων αυτορύθμισης της παρεχόμενης τροφής (Adron *et al*, 1973; Juell, 1991; Blyth *et al*, 1993; Juell *et al*, 1993) έχει εστιαστεί κυρίως στον προσδιορισμό της τροφής που χάνεται και εκ των υστέρων περιορισμό της τροφής που απελευθερώνεται.

Παρά το γεγονός ότι διάφορες μελέτες έχουν δείξει την ύπαρξη συγκεκριμένου προτύπου (pattern) που διαφέρει ανάλογα με το είδος ψαριού που εκτρέφεται και την

ηλικία των ατόμων κανείς δεν το εφαρμόζει τουλάχιστον σε βιομηχανική κλίμακα στην πράξη. Οι απαιτήσεις σε τροφή καθορίζονται εκτός των άλλων και από τους ημερήσιους εσωτερικούς ρυθμούς των ψαριών. Η χορήγηση της τροφής με βάση αυτό το πρότυπο βελτιώνει την ανάπτυξη των ατόμων και παράλληλα μειώνει την ποσότητα τροφής που καταναλώνεται. (Papandroulakis et al 1999. “Automated management system in vertical integrated farms of intensively reared euryaline fish”).

### **1.α Παροχή τροφής σύμφωνα με τη ζήτηση.**

Σε μικρή κλίμακα, και κυρίως σε χερσαίες εγκαταστάσεις η χορήγηση της τροφής πραγματοποιείται με την χρήση ταϊστών αυτοχειρισμού όπου η ρύθμιση της ποσότητας και η συχνότητα χορήγησης καθορίζεται από το ίδιο το ψάρι με την ενεργοποίηση ενός μοχλού που απελευθερώνει την τροφή. Σε βιομηχανική κλίμακα η χρήση των ταϊστών αυτοχειρισμού παρουσιάζει δυσκολίες καθώς εμπλέκονται προβλήματα διανομής τροφής (πρέπει να γίνεται σε αρκετά μεγάλη έκταση) για να ικανοποιήσει τις απαιτήσεις ενός μεγάλου αριθμού ατόμων και συμπεριφοράς του ιχθυοπληθυσμού. Διάφορες συγκριτικές μελέτες έχουν αποδείξει την υψηλή απόδοση που έχει ένα τέτοιο σύστημα όσον αφορά το κέρδος βάρους των ιχθύων, την αξιοποίηση της τροφής και την μείωση της ανθρώπινης εργασίας (Thorpe et al., 1990; Kentouri et al., 1993, Papandroulakis et al. 2001). Ωστόσο προκύπτουν κάποιες δυσκολίες. Ορισμένοι πληθυσμοί όπως οι πέστροφες δεν προσαρμόζονται σε αυτό το σύστημα παροχής τροφής με αποτέλεσμα να μην τρέφονται ή προκαλούν υπερτροφοδοσία “παίζοντας” με τον μοχλό-αισθητήρα (για όλα τα μεσογειακά θαλάσσια έχει δειχθεί το αντίθετο). Η ανάπτυξη των συστημάτων αυτών έχει εστιαστεί στην βελτίωση των αισθητήρων, των μηχανισμών τροφοδοσίας και στην αντοχή των υλικών.

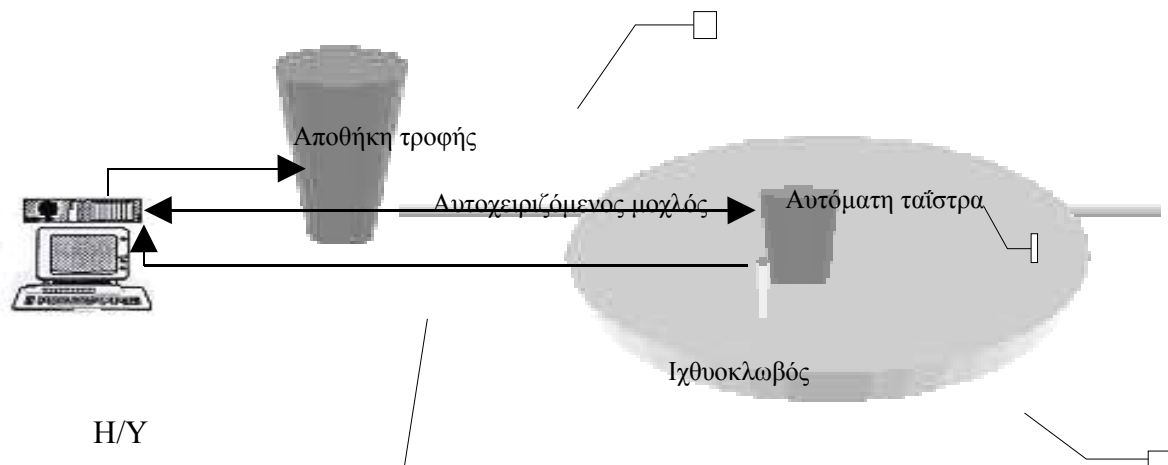
Τελευταίες έρευνες δείχνουν ότι για τα περισσότερα είδη ιχθύων, τα οποία ζουν σε ζεστά νερά, ο χρόνος προσαρμογής τους, στη μέθοδο, είναι 2-3 εβδομάδες. Μάλιστα έχει παρατηρηθεί ότι μέρος των πληθυσμών που “έμαθε” την μέθοδο ταΐσματος φροντίζει και για τον υπόλοιπο πληθυσμό. Η μεσόγειος κατακλύζεται από αυτά τα είδη και αυτό κάνει



την εφαρμογή της παροχής τροφοδοσίας σύμφωνα με τη ζήτηση ιδιαίτερα ελκυστική. (M.Paspatis 1998)

### 1.β Περιγραφή των συστημάτων τροφοδοσίας σύμφωνα με τη ζήτηση

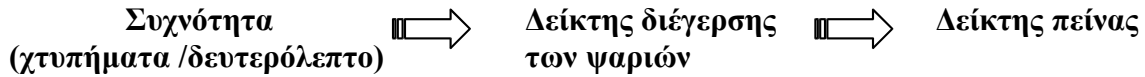
Η λογική των συστημάτων αυτοχειρισμού στηρίζεται στη κίνηση ενός μοχλού από το ψάρι και στην άμεση παροχή της τροφής. Το συγκεκριμένο σύστημα, που φαίνεται σχηματικά στο επόμενο διάγραμμα, σχεδιάστηκε στο Ινστιτούτο Υδατοκαλλιεργειών του ΕΛΚΕΘΕ στα πλαίσια υλοποίησης ερευνητικού έργου (Papandroulakis et al 1999)



1.2 Σύστημα τροφοδοσίας σύμφωνα με τη ζήτηση

Σε κάθε ιχθυοκλωβό τοποθετείται ένας ή περισσότεροι αυτοχειριζόμενοι μοχλοί ανάλογα με το μέγεθος του πληθυσμού. Η ενεργοποίηση του μοχλού καταγράφεται σε ένα ηλεκτρονικό υπολογιστή ο οποίος στην συνέχεια ενεργοποιεί μία μηχανική ταΐστρα για συγκεκριμένο χρόνο. Η ποσότητα τροφής και η συχνότητα χορήγησης καθορίζονται από λογισμικό. Ο υπολογισμός της ποσότητας και συχνότητας της χορηγούμενης τροφής βασίζεται στην παρακολούθηση και καταγραφή της δραστηριότητας - χτυπήματα των ψαριών σε ένα αυτοχειριζόμενο μοχλό (όπως περιγράφηκε παραπάνω) και στην διανομή της τροφής, με βάση την συχνότητα που παρουσιάζει αυτή η δραστηριότητα, σε χρονικά διαστήματα μισού λεπτού. Η μεγάλη συχνότητα χορήγησης (30 δευτερόλεπτα) είναι

απαραίτητη ώστε τα ψάρια να μπορούν να συνδέσουν την ενεργοποίηση του μοχλού με την παροχή τροφής. Η βιολογική βάση της μεθοδολογίας παρουσιάζεται σχηματικά στην συνέχεια.



### 1.γ.Σχετικά με την συγκεκριμένη εργασία

Όλες οι μελέτες υποδεικνύουν ότι η μέθοδος της παροχής τροφής σύμφωνα με τη ζήτηση, ειδικά, στη μεσόγειο έχει να προσφέρει πολλά. Ταυτόχρονα όμως πρέπει να επιλυθούν αρκετά σημαντικά προβλήματα. Η πρακτική δυσκολία της λειτουργίας ενός υπολογιστικού συστήματος κοντά στους ιχθυοκλωβούς το υψηλό κόστος ύπαρξης ενός PLC για τον έλεγχο των μηχανικών ταϊστών. Ακόμα υπάρχει το πρόβλημα των ιχθυοπληθυσμών να προσαρμοστούν στο σύστημα και ασφαλώς το ζήτημα της ευελιξίας ενός τέτοιου συστήματος στον χειρισμό του αλλά και στην διαχείριση των δεδομένων.

Για να ξεπεραστούν αυτά τα προβλήματα αποφασίστηκε η δημιουργία ενός συστήματος βασισμένο σε κάποιον μικροεπεξεργαστή. Έτσι το κόστος μειώνεται κατά πολύ, επίσης ένα τέτοιο σύστημα καταλαμβάνοντας μικρό χώρο και έχοντας μικρή κατανάλωση γίνεται ιδανικό για αυτόνομη λειτουργία σε “σκληρές περιβαλλοντικές συνθήκες” αφού μπορεί να συνδυαστεί και με ένα σύστημα παροχής ενέργειας από ανανεώσιμες πηγές. Επιπλέον υπάρχει η δυνατότητα να μην λειτουργεί σαν μια απλή ταΐστρα αυτοχειρισμού (με κάθε αίτηση να δίνει μια μονάδα τροφής) ή να ακολουθεί κάποιους πίνακες διατροφής αλλά να τρέχει αλγορίθμους που θα μοντελοποιούν καλύτερα την συμπεριφορά των ιχθύων και θα έχουν καλύτερη απόδοση σε ακανόνιστες συμπεριφορές, ακόμη και να αυτοπροσαρμόζονται. Ακόμα με τη βοήθεια ενός GSM/GPRS μόντεμ έχουμε πρόσθετες λειτουργικότητες, ευελιξία στον χειρισμό του και διαχείριση δεδομένων αφού έχουμε την δυνατότητα επικοινωνίας με έναν σταθμό βάσης (όχι απαραίτητα σταθερό). Στην συνέχεια παρουσιάζεται η δουλειά που έχει γίνει στον συγκεκριμένο τομέα και ακριβώς το σύστημα που αναπτύχθηκε κατά την διάρκεια αυτής

της διπλωματικής εργασίας.

## 2. ΘΕΩΡΗΤΙΚΟ ΚΑΙ ΠΡΑΚΤΙΚΟ ΥΠΟΒΑΘΡΟ

Όπως και σε κάθε πραγματική εφαρμογή χρησιμοποιήθηκαν τεχνικές από ένα ευρύ γνωστικό φάσμα. Αν θα ήθελε κανείς να αναλύσει την συγκεκριμένη εργασία θα έβρισκε πέντε σημεία. Το ένα σημείο είναι μελέτη του τρόπου λειτουργίας μιας ιχθυοκαλλιέργειας (συμπεριφορά ψαριών, τεχνικές λεπτομέρειες, μέθοδοι τροφοδοσίας). Ένα άλλο είναι η υλοποίηση του συστήματος, διαδικασία που απαιτεί γνώσεις σχεδίασης κυκλωμάτων και προγραμματισμού του υλικού. Από άποψη διαχείρισης και συντονισμού του υλικού χρειάστηκε η ανάπτυξη κλασσικής και ασαφούς λογικής. Το πρόβλημα επεκτείνεται και σε θέματα τηλεπικοινωνιών με την εμφάνιση της ανάγκης για απομακρυσμένη επικοινωνία. Τέλος κατά το στάδιο της επαλήθευσης και της προσομοίωσης αναπτύχθηκαν προγραμματιστικά μοντέλα και έγινε χρήση στατιστικών μεθόδων για την αναπαράσταση της συμπεριφοράς των ψαριών και την ανάλυση των αποτελεσμάτων.

Ξεκινώντας κανείς ένα τέτοιο έργο, χωρίς προηγούμενη πείρα σε μεγάλης κλίμακας εργασίες, καταλαβαίνει αμέσως την αξία τεχνικών ανάλυσης απαιτήσεων, οργάνωσης και χρονοπρογραμματισμού. Για την υλοποίηση της συγκεκριμένης εργασίας χρησιμοποιήθηκε το μοντέλο του καταρράκτη ενώ για τον χρονοπρογραμματισμό της δημιουργήθηκε κατάλληλο PERT chart.

Όσον αφορά την ανάλυση των απαιτήσεων του συστήματος χρειάστηκε εκτός των άλλων και η πληροφόρηση γύρω από θέματα ιχθυοκαλλιεργειών. Πάνω σε θέματα τροφοδοσίας και ειδικότερα στην τροφοδοσία σύμφωνα με τη ζήτηση, το άρθρο του Μ. Πασπάτη που παρουσιάστηκε στο 3ο συμπόσιο του MAST (Maritimes Systems and Technology) με θέμα *”Demand feeding systems in aquaculture. A profitable practice for warm water marine species rearing”* βοήθησε στην κατανόηση αυτών των θεμάτων και στην ανάδειξη των προβλημάτων που έπρεπε να αντιμετωπιστούν.

Όμως οι μελέτες στις οποίες βασίστηκε η παρούσα εργασία είναι κατά αρχήν των Papndroulakis et al 2001 “*An automated feeding system for intensive hatceries*” η οποία πραγματεύεται τον σχεδιασμό και την υλοποίηση ενός αυτόματου συστήματος τροφοδοσίας βασισμένο σε διατροφικούς πίνακες. Οι πίνακες αυτοί έχουν πληροφορίες για την ποσότητα και συχνότητα χορήγησης της τροφής. Στο σύστημα αυτό απαιτείται η εισαγωγή αυτών των πινάκων κάθε εικοσιτέσσερις ώρες και προσφέρει ευελιξία στην προσαρμογή τους σε πραγματικό χρόνο. Και κατά δεύτερον η μελέτη των Papndroulakis et al 1999. “*Automated management system in vertical integrated farms of intensively reared euryaline fish*”. Από την οποία εκτός από τα στοιχεία που αναφέρθηκαν στην εισαγωγή προέκυψε και ο crisp αλγόριθμος.

Με την τροφοδοσία σύμφωνα με τη ζήτηση επίσης έχει ασχοληθεί και το Department of Fisheries της κυβέρνησης της δυτικής Αυστραλίας, για την ιχθυοκαλλιέργεια των μαρονιών. (<http://www.fish.wa.gov.au/docs/aq/aq012/index.php?0404>)

Μετά την ανάλυση των απαιτήσεων και τον σχεδιασμό του συστήματος έρχεται η υλοποίηση του. Κατά αρχήν χρειάστηκε η μελέτη όλων των εγχειριδίων (manuals, datasheets) για τον μικροεπεξεργαστή, το αναπτυξιακό και το μόντεμ. Για τον προγραμματισμό του υλικού χρειάστηκε η μελέτη της assembly του μικροεπεξεργαστή και προγραμματισμός σε embedded C. Για την embedded C το βιβλίο “Developing Embedded Software in C Using ICC11/ICC12/Metrowerks” του Jonathan W. Valvano χρησιμοποιήθηκε σαν αναφορά. Όσον αφορά την ανάπτυξη της ασαφούς λογικής στηρίχθηκε η εργασία σε άρθρα του Lofti A. Zadeh όσον αφορά τη θεωρία των ασαφών συνόλων και το εγχειρίδιο του Matlab σε θέματα ασαφούς λογικής.

Για το κομμάτι της τηλεπικοινωνίας χρειάστηκε η μελέτη για τον χειρισμό GSM/GPRS μόντεμ, συγκεκριμένα ο χειρισμός γινόταν με εντολές AT. Ενώ τέλος για την προσομοίωση και την ανάλυση δεδομένων χρησιμοποιήθηκαν εργαλεία όπως το Matlab και στατιστικές μέθοδοι.

### 3. ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΑΣΑΦΗ ΛΟΓΙΚΗ (FUZZY LOGIC)

“Η Ασαφής Λογική έχει δύο διαφορετικές σημασίες. Σε μια στενότερη έννοια, η Ασαφής Λογική είναι ένα λογικό σύστημα, το οποίο είναι μια προέκταση της λογικής πολλών τιμών. Αλλά σε μια πιο ευρεία έννοια, που είναι και η κυρίαρχη σε χρήση σήμερα, η Ασαφής Λογική είναι σχεδόν συνώνυμη με τη θεωρία των ασαφών συνόλων, μιας θεωρίας που σχετίζεται με κλάσεις αντικειμένων με μη διακριτά όρια, στα οποία η συγγένεια είναι θέμα κάποιου βαθμού. Κατά αυτή την προοπτική, η Ασαφής Λογική στην στενή της έννοια είναι ένα παρακλάδι αυτής της ευρείας. Αυτό που είναι σημαντικό να αναγνωρίσουμε είναι ότι, ακόμα και με την στενή έννοια, η ατζέντα της Α.Λ. είναι πολύ διαφορετική σε πνεύμα και ουσία από τις ατζέντες των παραδοσιακών συστημάτων λογικής πολλών τιμών.” Lotfi A. Zadeh Berkeley, CA Ιανουάριος 1995.

Ένα κλασσικό παράδειγμα της θεωρίας των ασαφών συνόλων είναι αυτό με το ύψος. Αν ένας άνθρωπος θεωρείται ψηλός όταν ξεπερνάει τα 185 εκατοστά, είναι απόλυτα σωστό να πούμε ότι κάποιος με ύψος 184 εκ. δεν είναι ψηλός; Εδώ το πρόβλημα δεν οφείλεται τόσο στις έννοιες που χρησιμοποιούνται όσο στην αντίληψη που έχει ο καθένας για λεκτικούς προσδιορισμούς ποσοτικών μεγεθών. Έτσι λοιπόν η *ασαφής λογική (fuzzy logic)* είναι ένα υπερσύνολο της κλασσικής λογικής, η οποία έχει επεκταθεί ώστε να μπορεί να χειριστεί τιμές αληθείας μεταξύ του "απολύτως αληθούς" και του "απολύτως ψευδούς". Επίσης θα μπορούσε να πει κανείς ότι αποτελεί μια πολύ καλή λύση για την εύρεση της χρυσής τομής μεταξύ ακρίβειας και σημαντικότητας.

(ΕΙΚΟΝΑ)

#### 3.α Βασικές έννοιες

*Ασαφές Σύνολο (fuzzy set) A*: ένα σύνολο διατεταγμένων ζευγών  $(x, \mu_A(x))$  όπου  $x \in X$

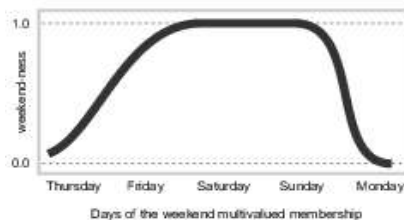
και  $u_A(x) \in [0,1]$ ).

1. Το σύνολο  $X$  αποτελεί ένα ευρύτερο σύνολο αναφοράς (*universe of discourse*) που περιλαμβάνει όλα τα αντικείμενα στα οποία μπορεί να γίνει αναφορά.
2. Η τιμή  $u_A(x)$  λέγεται *βαθμός αληθείας* (*degree of truth*), συμβολίζει το βαθμό της συγγένειας του  $x$  στο  $A$  και παίρνει τιμές στο διάστημα  $[0,1]$ .
3. Η συνάρτηση  $u_A$  ονομάζεται *συνάρτηση συγγένειας* (*membership function*) και στην πράξη μπορεί να προέρχεται από:

- Υποκειμενικές εκτιμήσεις
- Προκαθορισμένες (ad hoc) και απλοποιημένες μορφές
- Συχνότητες εμφανίσεων και πιθανότητες
- Φυσικές μετρήσεις
- Διαδικασίες μάθησης και προσαρμογής (συνήθως με νευρωνικά δίκτυα)

Σημείωση: Η ασαφής θεωρία συνόλων μεταπίπτει στην αντίστοιχη κλασική, όταν οι δυνατές τιμές της συνάρτησης συγγένειας είναι μόνο 0 και 1.

Ένα παράδειγμα ενός τέτοιου συνόλου είναι οι ημέρες του σαββατοκύριακου. Το σύνολο αναφοράς περιλαμβάνει όλες τις ημέρες, από αυτές το Σάββατο και η Κυριακή ανήκουν ασφαλώς στο σύνολο. Η Παρασκευή κατά κάποιο τρόπο ανήκει και δεν ανήκει στο Σ/Κ. Αυτό εκφράζεται και στην συνάρτηση συγγένειας για το συγκεκριμένο ασαφές σύνολο όπως φαίνεται παρακάτω.

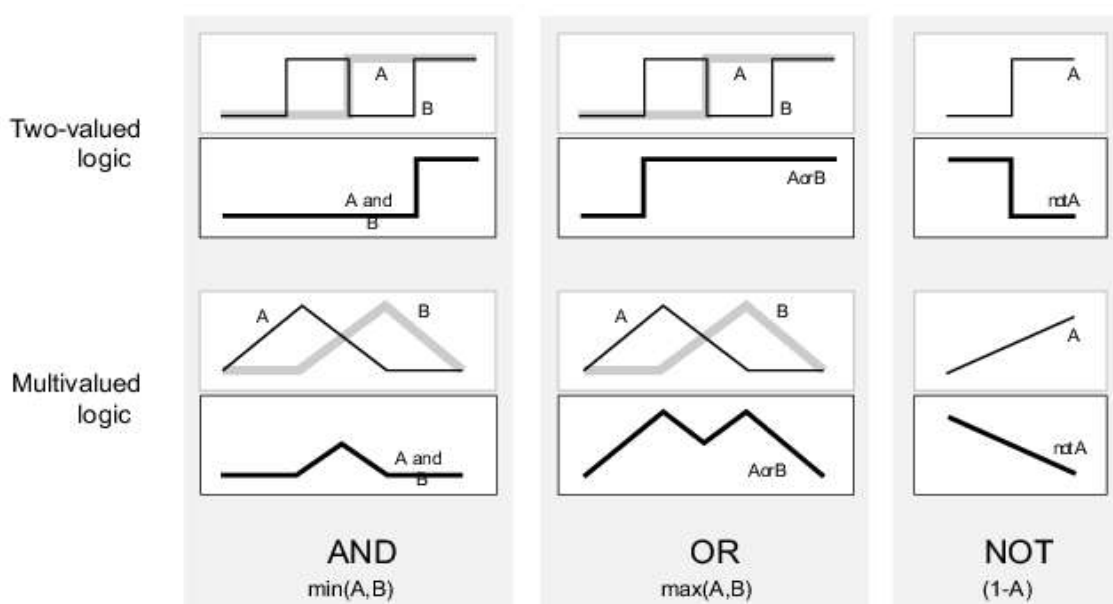


3.1 συνάρτηση συγγένειας των ημερών της εβδομάδας

Αυτά όσον αφορά τα ασαφή σύνολα, τώρα θα μελετήσουμε την λογική. Η ασαφής λογική είναι υπερσύνολο της Boolean λογικής. Αυτό σημαίνει ότι αν εφαρμόσουμε τους τελεστές της Α.Λ. με τιμές 0 και 1 θα έχουμε πλήρη ταύτιση με τους πίνακες αληθείας της διττής λογικής.

### Λεκτικοί Τελεστές:

Στην Α.Λ. Το πόσο αληθής είναι μια δήλωση αποτελεί μέρος κάποιου βαθμού οπότε πρέπει να βρούμε μια άλλη συνάρτηση που να περιγράφει τους τελεστές. Για τον τελεστή **Ή** χρησιμοποιούμε την συνάρτηση του μεγίστου των δύο τιμών, για τον τελεστή **ΚΑΙ** χρησιμοποιούμε την συνάρτηση της ελάχιστης τιμής ενώ για την **ΑΝΤΙΣΤΡΟΦΗ** το  $1-A$  όπου  $A$  η μεταβλητή μας. Το σχήμα παρακάτω εξηγεί ότι έχει λεχθεί ως τώρα.



3.2 Λεκτικοί τελεστές

Εκτός από τις παραπάνω συναρτήσεις για την εξαγωγή της Ασαφούς Τομής(ΚΑΙ) και της Ασαφούς Ένωσης(Η) από τον Zadeh, υπάρχουν και η συνάρτηση του γινομένου για την τομή  $u_A(x) * u_B(x)$  ενώ για την ένωση έχουμε το άθροισμα μείον το γινόμενο  $u_A(x) + u_B(x) - u_A(x) u_B(x)$ .

### Ασαφείς Μεταβλητές(Fuzzy Variables):

Στην ασαφή λογική ασφαλώς υπάρχουν μεταβλητές, η *Ασαφής Μεταβλητή* ορίζεται



ως μια μεταβλητή της οποίας οι τιμές ορίζονται με ασαφή σύνολα.

Π.χ. τα ασαφή σύνολα {κοντός, μεσαίος, ψηλός} θα μπορούσαν να είναι το πεδίο τιμών της ασαφούς μεταβλητής "ύψος".

Η μεταβλητή "ύψος" χαρακτηρίζεται και ως *λεκτική (linguistic)* μεταβλητή. Επίσης από ένα μικρό αρχικό αριθμό πρωταρχικών λεκτικών τιμών, να προκύψει ένας πολύ μεγαλύτερος αριθμός *σύνθετων λεκτικών τιμών* με τη χρήση των *λεκτικών τελεστών* ΚΑΙ, Ή, ΑΝΤΣΤΡΟΦΗ.

#### Ασαφείς Κανόνες(Fuzzy Rules):

Το τελευταίο κομμάτι στην ασαφή λογική συμπληρώνουν οι ασαφείς κανόνες(α.κ.) *Εάν Τότε*. Είναι αυτοί που διέπουν τη λειτουργία του συστήματος, θα έλεγε κανείς ότι τα ασαφή σύνολα και οι τελεστές είναι τα υποκείμενα και τα αντικείμενα στην γλώσσα της ασαφούς λογικής όπου οι ασαφείς κανόνες *Εάν Τότε* είναι οι προτάσεις.

Οι κανόνες είναι της μορφής Εάν  $\chi$  είναι A Τότε  $\phi$  είναι B.

Όπου A,B είναι λεκτικές μεταβλητές ορισμένες από ασαφή σύνολα στο σύνολο αναφοράς του X και  $\Phi$  αντίστοιχα. Το πρώτο μέρος του κανόνα αποτελεί την πρόταση ενώ το δεύτερο το συμπέρασμα.

Να σημειωθεί ότι το A και το B αντιπροσωπεύουν έναν αριθμό από το 0 έως το 1, έτσι η πρόταση είναι μια παράσταση που επιστρέφει έναν αριθμό από το 0 έως το 1. Αντιθέτως το B αναπαριστά ένα ασαφές σύνολο. Έτσι το συμπέρασμα είναι μια ανάθεση του ασαφούς συνόλου B στη μεταβλητή εξόδου  $\phi$ .

πχ. Εάν η ταχύτητα είναι μέτρια τότε η πίεση στα φρένα να είναι μέτρια

Η ταχύτητα και η πίεση είναι ασαφείς μεταβλητές το μέτρια για την ταχύτητα είναι μια τιμή από 0 έως 1 ενώ το μέτρια στο συμπέρασμα είναι ένα ασαφές σύνολο.

Η ανάλυση ενός α.κ. αποτελείται από διακριτά μέρη και είναι μία ασαφής σχέση  $R(x,y)$  που ονομάζεται *σχέση συνεπαγωγής (implication relation)*. Πρώτα υπολογίζεται η πρόταση (το οποίο περιλαμβάνει ασαφοποίηση ,fuzzifying, των εισόδων και εφαρμογή των αναγκαίων ασαφών τελεστών) και έπειτα εφαρμόζεται το αποτέλεσμα της πρώτης αυτής διαδικασίας στο συμπέρασμα(*συνεπαγωγή*). Γενική μορφή της σχέσης (συνάρτησης) συνεπαγωγής:  $R(x,y) \equiv u(x,y) = \varphi(uA(x), uB(y))$

Η συνάρτηση  $\varphi$  ονομάζεται *τελεστής συνεπαγωγής (implication operator)* και υποδεικνύει τον ακριβή τρόπο συνδυασμού των συναρτήσεων συγγένειας του *εάν* και του *τότε* τμήματος ενός ασαφούς κανόνα, ώστε να προκύψει η αναλυτική του έκφραση. Οι πιο συνηθισμένοι τελεστές είναι ο Mamdani Min και ο Larsen Product, αναλυτικά στον πίνακα παρακάτω.

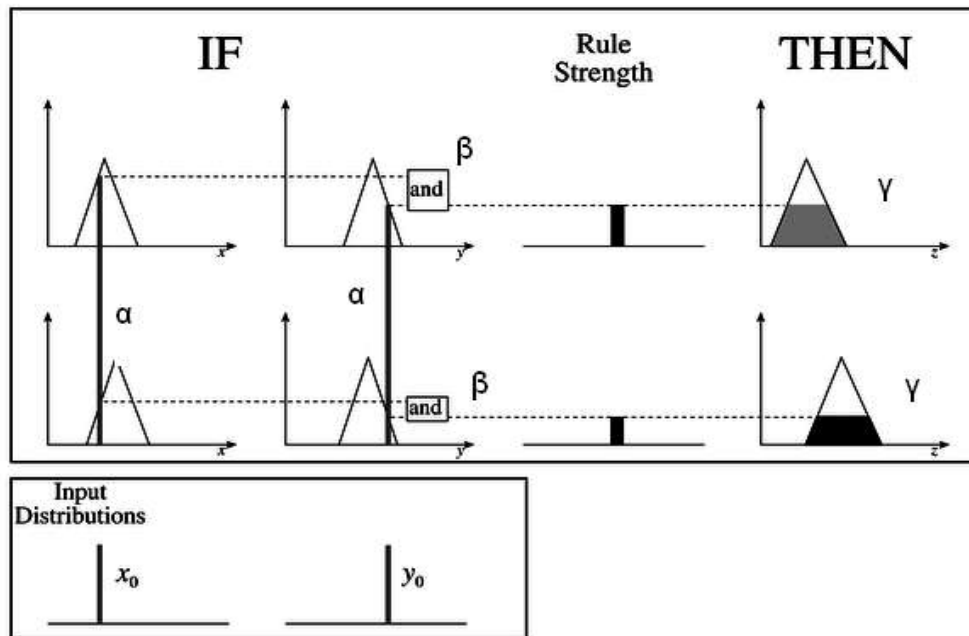
Ονομασία Τελεστή	Αναλυτική Έκφραση του $\varphi[uA(x),uB(y)]$
$\varphi_m$ : Zadeh Max-Min	$(uA(x) \wedge uB(y)) \vee (1 - uA(x))$
$\varphi_c$ : Mamdani Min	$uA(x) \wedge uB(y)$
$\varphi_p$ : Larsen Product	$uA(x) \cdot uB(y)$
$\varphi_a$ : Arithmetic	$1 \wedge (1 - uA(x) + uB(y))$
$\varphi_b$ : Boolean	$(1 - uA(x)) \vee uB(y)$

Πίνακας 3.1 τελεστές συνεπαγωγής

Συνοπτικά η ανάλυση των α.κ. έχει τρία βήματα:

- α) Μετατροπή των σαφών τιμών εισόδων σε ασαφείς
- β) Εφαρμογή των τελεστών στην πρόταση του κανόνα
- γ) Εφαρμογή της μεθόδου συνεπαγωγής

Όπως φαίνονται στο παρακάτω σχήμα



### 3.3 Ανάλυση ασαφών κανόνων

#### Ασαφή Συλλογιστικά Συστήματα(Fuzzy Inference Systems) ΑΣΣ

Ασαφή Συλλογιστική είναι η διαδικασία με την οποία αντιστοιχίζουμε εισόδους με εξόδους χρησιμοποιώντας ασαφή λογική. Η διαδικασία αυτή περιλαμβάνει όλα τα κομμάτια που περιγράφηκαν παραπάνω :ασαφή σύνολα, λεκτικοί τελεστές, κανόνες και συναρτήσεις συγγένειας. Θα αναφερθούν δύο τέτοια συστήματα τύπου Mamdani και τύπου Sugeno. Τα δύο αυτά συστήματα διαφέρουν στο πως ορίζεται η έξοδος. Αξίζει να αναφερθεί ότι τα ΑΣΣ έχουν εφαρμοσθεί με επιτυχία σε τομείς όπως ο αυτόματος έλεγχος, ταξινόμηση δεδομένων, ανάλυση αποφάσεων και μηχανική όραση.

#### Mamdani

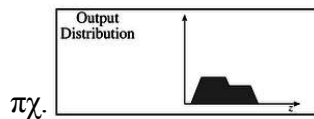
Τα συστήματα τύπου Mamdani είναι τα πιο κοινά. Η μέθοδος αυτή προτάθηκε το 1975 από τον Ebrahim Mamdani σαν μια προσπάθεια να ελέγξει μια μηχανή ατμού σε συνδυασμό με έναν βραστήρα συνθέτοντας ένα σύνολο λεκτικών ελεγκτικών κανόνων από ειδικούς και ήταν ο πρώτος ελεγκτής που δημιουργήθηκε με την θεωρία των ασαφών συνόλων. Το σύστημα του βασίστηκε σε μια διατριβή του Lotfi Zadeh's, το 1973, σε

ασαφείς αλγορίθμους για πολύπλοκα συστήματα και διαδικασίες αποφάσεων.

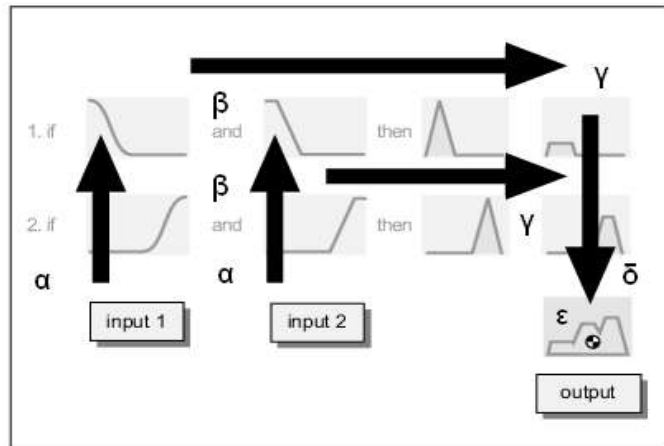
Τα βήματα λοιπόν για τη δημιουργία ενός τύπου Mamdani συστήματος είναι:

1. Ο ορισμός ενός συνόλου ασαφών κανόνων
2. Ασαφοποίηση των σαφών(crisp) εισόδων χρησιμοποιώντας τις συναρτήσεις συγγένειας
3. Εφαρμογή των τελεστών στην πρόταση του κανόνα
4. Εφαρμογή του τελεστή συνεπαγωγής στον κανόνα
5. Άθροιση όλων των συνεπαγωγών για την δημιουργία του ασαφούς σετ εξόδου και
6. Αποσαφήνιση της εξόδου

Όλα τα παραπάνω έχουν καλυφθεί εκτός από τα 5,6. Το Mamdani τύπου σύστημα θεωρεί ότι η έξοδος εκφράζεται με συναρτήσεις συγγένειας από ασαφή σύνολα. Μετά από την εφαρμογή του τελεστή συνεπαγωγής σε κάθε κανόνα έχουμε μια ασαφή τιμή για την μεταβλητή της εξόδου. Αυτές οι τιμές αθροίζονται ώστε να προκύψει η έξοδος σαν ένα ασαφές σύνολο.



Στην συνέχεια έχουμε την διαδικασία της αποσαφήνισης (defuzzyfication), η διαδικασία αυτή μας επιστρέφει ένα συγκεκριμένο αποτέλεσμα από την ασαφή έξοδο. Υπάρχουν πολλοί μέθοδοι για αποσαφήνιση, οι γνωστότεροι είναι ο κεντροειδής (υπολογίζεται το κέντρο μάζας), ο μέσος όρος των μεγίστων, του μεγαλύτερου από τα μέγιστα και του ελαχίστου από τα μέγιστα. Στο παρακάτω σχήμα βλέπουμε τη περιγραφή του συστήματος τύπου Mamdani.



3.4

- α. ασαφοποίηση εισόδων
- β. εφαρμογή τελεστών
- γ. συνεπαγωγή
- δ. άθροιση εξόδων
- ε. αποσαφήνιση

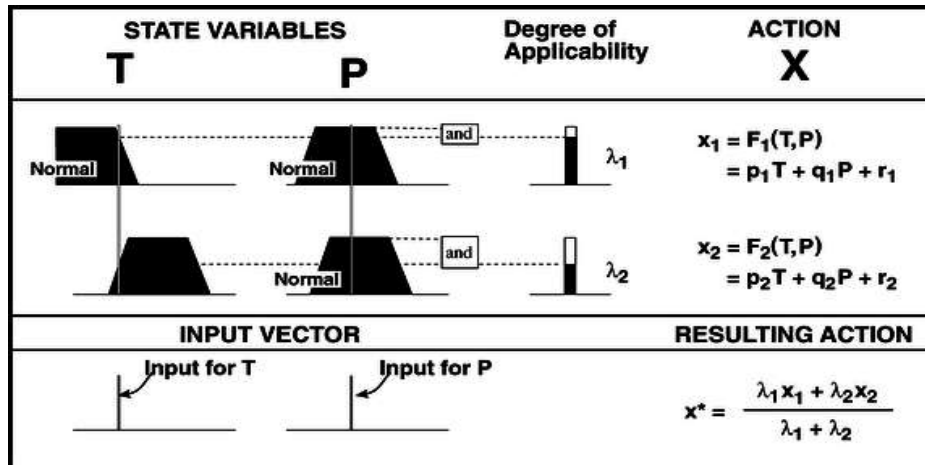
### Sugeno

Το ΑΣΣ τύπου Sugeno πρωτοεμφανίστηκε το 1985 και έχει πολλές ομοιότητες με το Mamdani. Τα δύο πρώτα μέρη της συλλογιστικής διαδικασίας δηλαδή η ασαφοποίηση και η εφαρμογή των τελεστών είναι ταυτόσημες με αυτά που περιγράφηκαν παραπάνω. Η κύρια διαφορά είναι ότι οι συναρτήσεις συγγένειας της εξόδου είναι σταθερές η γραμμικές. Ένας τυπικός κανόνας σε αυτό το μοντέλο έχει την μορφή:

Εάν Είσοδος1 =τ και Είσοδος2 =ρ τότε η έξοδος είναι  $\chi = p\tau + q\rho + \sigma$

Για ένα μοντέλο μηδενικού βαθμού έχουμε  $p=q=0$  οπότε  $\chi=\sigma$ . Το επίπεδο της εξόδου κάθε κανόνα  $\chi_i$  έχει συγκεκριμένη βαρύτητα  $\lambda_i$  πχ. Για την πρόταση με τελεστή ΚΑΙ “Εάν Είσοδος1 =τ και Είσοδος2 =ρ” η βαρύτητα  $\lambda_i$  είναι  $\lambda_i = KAI(\Sigma 1(\tau), \Sigma 2(\rho))$ . Όπου  $\Sigma 1,2(.)$  είναι οι συναρτήσεις συγγένειας των εισόδων. Τελικά η έξοδος του συστήματος είναι το άθροισμα των γινομένων εξόδων,βαρύτητας προς το άθροισμα όλων των βαρών(

Μέσος Όρος με βάρη). Τελική Έξοδος= 
$$\frac{\sum \chi_i * \lambda_i}{\sum \lambda_i}$$
 το πως δουλεύει ένα τέτοιο σύστημα φαίνεται στο διάγραμμα.



3.5 Το σύστημα Sugeno

Ένα από τα μεγαλύτερα προβλήματα με τα ΑΣΣ τύπου Sugeno είναι ότι δεν υπάρχει διαισθητικός τρόπος για να ορίσει κανείς τους συντελεστές  $p, q$  και  $r$ . Επίσης έχουμε μόνο σαφείς εξόδους που σε μερικές περιπτώσεις δεν είναι επιθυμητό. Όμως είναι μια μέθοδος πολύ πιο εύκολη στον υπολογισμό της, επίσης μπορούν να εφαρμοσθούν προσαρμοστικοί αλγόριθμοι για την ρύθμιση των συναρτήσεων συγγένειας έτσι ώστε το Ασαφές Σύστημα να μοντελοποιεί με τον καλύτερο τρόπο τα δεδομένα.

### Επίλογος

Η μέθοδος Mamdani είναι διαισθητική, δημοφιλής και κοντά στα ανθρώπινα δεδομένα από την άλλη πλευρά η μέθοδος Sugeno είναι υπολογιστικά προσιτή, συνεργάζεται πολύ καλά με γραμμικές τεχνικές (πχ. PID) και προσφέρεται για μαθηματική ανάλυση.

## 4. ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ

### 4.α Περιγραφή Λειτουργίας

Το σύστημα αποτελείται από τον μικροεπεξεργαστή πάνω σε ένα αναπτυξιακό, το κινητό τηλέφωνο, το αισθητήριο και το μηχανισμό τροφοδοσίας. Στο αναπτυξιακό του μικροεπεξεργαστή υπάρχει μια σειριακή θύρα που επιτρέπει την επικοινωνία του με το κινητό τηλέφωνο. Το αισθητήριο συνδέεται στην ακίδα εξωτερικών διακοπών και ο μηχανισμός τροφοδοσίας σε μια θύρα του μικροεπεξεργαστή. Το τελικό σύστημα περιέχει όλα τα επιμέρους (εκτός του αναπτυξιακού) σε μια πλακέτα.

Κατά τη λειτουργία του συστήματος δέχεται ο μικροεπεξεργαστής μέσω του αισθητηρίου παλμούς που δηλώνουν αίτηση για τροφή. Τις αιτήσεις αυτές επεξεργάζεται κάθε μισό λεπτό και δίνει την απαραίτητη ποσότητα τροφής σύμφωνα με τον αλγόριθμο που τρέχει εκείνη την περίοδο. Η ποσότητα που δίνεται κάθε μισόλεπτο κρατείται και αθροίζεται, έτσι ώστε να έχουμε μια αναφορά για αυτήν κάθε μισάωρο. Όταν η πληροφορία, από τα διαδοχικά μισάωρα, είναι αρκετά μεγάλη ώστε να ξεπερνάει το μέγεθος χωρητικότητας ενός μηνύματος κινητού τηλεφώνου τότε δημιουργείται ένα πακέτο που στέλνεται στον σταθμό βάσης.

Η επικοινωνία με τον σταθμό βάσης( ο οποίος δεν είναι κατά ανάγκη σταθερός ) είναι αμφίδρομη, δηλαδή δεν εξυπηρετεί στην αποστολή μόνο των δεδομένων. Μέσα στα χαρακτηριστικά λειτουργίας του συστήματος είναι να δέχεται εντολές με την χρήση μηνυμάτων κινητού τηλεφώνου (GSM). Με τις εντολές αυτές γίνονται διάφορες λειτουργίες, ίσως η σημαντικότερη είναι η δυνατότητα επιλογής του αλγόριθμου τροφοδοσίας που περιγράφονται αμέσως παρακάτω.

#### 4.β ΑΛΓΟΡΙΘΜΟΙ ΤΡΟΦΟΔΟΣΙΑΣ

Δημιουργήθηκαν τρεις αλγόριθμοι τροφοδοσίας ο ένας είναι crisp αλγόριθμος και οι άλλοι δύο χρησιμοποιούν ασαφή λογική. Η καταγραφή των χτυπημάτων γίνεται κάθε μισό λεπτό και ο αριθμός τους αποθηκεύεται (σημειώνεται σαν  $N_i$ ). Υπάρχει μια αρχική ημερήσια ποσότητα  $Q_D$  η ποσότητα αυτή διαιρείται σε ίσα μέρη που αντιπροσωπεύουν την ποσότητα που θα έπρεπε να διανεμηθεί κάθε μισό λεπτό αν η κατανάλωση τροφής από τα ψάρια ήταν συνεχής στο χρόνο που τρέφονται τα ψάρια. Η ποσότητα που προκύπτει σημειώνεται με  $Q_p$ . Αυτή είναι η ποσότητα που θα διανεμηθεί το πρώτο και δεύτερο μισό λεπτό της ημέρας που θα καταγραφεί ενεργοποίηση των αισθητηρίων. Τα υπόλοιπα μισόλεπτα της ημέρας θα δίνεται τροφή διαφοροποιημένη κατά κάποιο ποσοστό την κάθε φορά από το προηγούμενο μισόλεπτο η έξοδος αυτή συμβολίζεται  $Q_i$ .

Όμως είναι δυνατό να μην δοθεί και καθόλου τροφή σε κάποια μισόλεπτα, τι γίνεται όταν η τροφή μισολέπτου έχει μειωθεί τόσο ώστε να έχει φτάσει στο μηδέν; Λογικά με ότι και να την πολλαπλασιάσουμε θα παραμείνει μηδέν. Εδώ έχει πάλι ρόλο το  $Q_p$ , όταν έχουμε έξοδο κάτω από ένα όριο και μετά από μια περίοδο αίτηση για τροφή τότε το ποσοστό διαφοροποίησης θα εφαρμοσθεί πάνω στο  $Q_p$ . Με λίγα λόγια το  $Q_p$  είναι η βασική μονάδα τροφής ανά μισόλεπτο και δεν μπορεί να μένει σταθερή όσο αυξάνεται ο πληθυσμός, η ανάγκη για τροφή γενικότερα. Η ύπαρξη του κάτω ορίου εξυπηρετεί επίσης στην περίπτωση που έχουμε έστω και μικρή ζήτηση αυτή να εξυπηρετηθεί.

πχ. Έστω  $Q_p = 10$ , κάτω όριο 3 και  $Q_i = 2$ . Στην επόμενη μονάδα χρόνου το  $Q_i$  θα έχει ως αναφορά σε προηγούμενη είσοδο το  $Q_p$ .

Τα παραπάνω είναι τα κοινά στοιχεία των αλγορίθμων αυτό στο οποίο διαφέρουν είναι ο τρόπος με τον οποίο διαχειρίζονται την τροφοδότηση μετά το δεύτερο μισόλεπτο.

##### **i)Σαφής(Crisp):**

Η λογική εδώ είναι η εξής, όσες πιο πολλές αιτήσεις είχαμε στο προηγούμενο



μισόλεπτο από το προ-προηγούμενο τόση περισσότερη τροφή θα δώσουμε. Και το αντίστροφο. Για παράδειγμα έστω  $N_2$  οι αιτήσεις το δεύτερο μισόλεπτο και  $N_3$  του τρίτου, τότε η τροφή που θα δοθεί το τρίτο μισόλεπτο  $Q_3$  είναι:

$$\text{Αν } N_2 > N_1 \text{ τότε } Q_3 > Q_p \text{ με } Q_3 = Q_p * N_2 / N_1$$

$$\text{Αν } N_2 < N_1 \text{ τότε } Q_3 < Q_p \text{ με } Q_3 = Q_p * N_2 / N_1$$

Γενικά η ποσότητα τροφής ( $Q_i$ ) που θα χορηγείται το χρονικό διάστημα  $i$ , θα υπολογίζεται όπως παρακάτω:

$$\text{Αν } N_{i-1} > N_{i-2} \text{ τότε } Q_i > Q_{i-1} \text{ με } Q_i = Q_{i-1} * N_{i-1} / N_{i-2}$$

$$\text{Αν } N_{i-1} < N_{i-2} \text{ τότε } Q_i < Q_{i-1} \text{ με } Q_i = Q_{i-1} * N_{i-1} / N_{i-2}$$

ή αλλιώς

$$Q_i = Q_{i-1} * N_{i-1} / N_{i-2}$$

Όπου  $N_{i-1}$ ,  $N_{i-2}$  τα χτυπήματα που καταγράφηκαν στα διαστήματα  $(i-1)$  και  $(i-2)$ .

## ii) Αλγόριθμοι Ασαφούς Λογικής (Α.Λ.):

Κατά αρχήν πρέπει να αναφερθεί γιατί επιλέχθηκε η δημιουργία δύο επιπλέον αλγορίθμων και μάλιστα Α.Λ. . Εκ πρώτης όψεως ο παραπάνω αλγόριθμος φαίνεται να δουλεύει καλά και ίσως να το κάνει αλλά εμείς ζητάμε την μέγιστη απόδοση. Η μέγιστη απόδοση για το σύστημα μας ορίζεται ως την *χωρίς πρόβλημα ανάπτυξη των ψαριών με τη λιγότερη δυνατή τροφή*. Ασφαλώς ο κάθε πληθυσμός χρειάζεται συγκεκριμένη ποσότητα καθημερινά για την υγιεινή του ανάπτυξη, όμως σε όλες τις ιχθυοκαλλιέργειες ως τώρα δινόταν πολλαπλάσιο της τροφής αυτής στους ιχθυοκλωβούς. Αυτό συμβαίνει γιατί μέρος της ποσότητας που δίνεται χάνεται στη θάλασσα (δεν προλαβαίνει να καταναλωθεί) και για άλλους λόγους.

Με την χρησιμοποίηση λοιπόν της Α.Λ. περιμένει κανείς να πλησιάσει την συμπεριφορά του πληθυσμού και έτσι να αποφύγουμε απότομες αυξομειώσεις στην

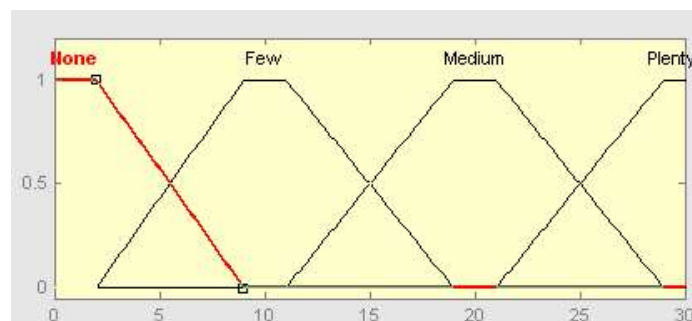
ποσότητα που δίνεται κάθε μισόλεπτο που οδηγούν σε απώλεια τροφής, να απορροφήσουμε τυχόν εσφαλμένες αιτήσεις και να αποδίδουμε τη σωστή ποσότητα κάθε φορά ανεξάρτητα από ένα αυστηρό κλασικό μοντέλο.

π.χ. Έστω ότι στο μισόλεπτο 3 έχουμε “πολλές”(28) αιτήσεις και στο 4 πάλι “πολλές”(30) τότε δεν θα έπρεπε να δώσουμε περισσότερη τροφή ή τουλάχιστο την ίδια με πριν; Ο crisp όμως αλγόριθμος δεν θα συμπεριφερθεί έτσι και θα δώσει λιγότερη τροφή κατά 7% περίπου.

Οι δύο αλγόριθμοι έχουν κοινό σύνολο τιμών και συναρτήσεις συγγένειας καθώς και κανόνες. Όμως το ένα είναι σύστημα κατά Mamdani και το άλλο κατά Sugeno. Το σύστημα και στις δύο περιπτώσεις είναι δύο εισόδων μιας εξόδου.

Οι μεταβλητές εισόδου ονομάζονται  $N_{i-1}$ ,  $N_{i-2}$  δηλαδή τα χτυπήματα που καταγράφηκαν στα διαστήματα (i-1) και (i-2). Ενώ η έξοδος είναι το ποσοστό κατά το οποίο πρέπει να αλλάξει η προηγούμενη ποσότητα τροφής.

Τα ασαφή σύνολα στην περίπτωση των εισόδων είναι η ένταση ή αριθμός των αιτήσεων και το σύνολο αναφοράς είναι ο αριθμός των χτυπημάτων από 0 έως 30. Αυτά είναι *Καθόλου*, *Μικρή*, *Μεσαία* και *Μεγάλη*. Οι συναρτήσεις συγγένειας έχουν τραπεζοειδή μορφή και φαίνονται στο σχήμα παρακάτω.



4.1 Οι συναρτήσεις συγγένειας

Η έξοδος έχει σχέση με το ποσοστό κατά το οποίο πρέπει να μεταβληθεί η

προηγούμενη ποσότητα τροφής και κυμαίνεται από 0-200%. Λεκτικά προσδιορίζεται ως *Μέγιστη Μείωση*, *Μείωση*, *Διατήρηση*, *Αύξηση* και *Μέγιστη Αύξηση*, ενώ ανάλογα με την υλοποίηση αποτελούν οι προσδιορισμοί ασαφή σύνολα ή όχι.

Στην συνέχεια γίνεται αναφορά στους κανόνες *Εάν Τότε* των δύο Συστημάτων Πορίσματος με Α.Λ. . Η αναφορά θα γίνει με την βοήθεια του παρακάτω πίνακα.

Εάν	N <sub>i-1</sub>	Τελεστής	N <sub>i-2</sub>	Τότε	Ποσοστό
Εάν	K	KAI	K	Τότε	MM
Εάν	K	KAI	MI	Τότε	Δ
Εάν	K	KAI	ME	Τότε	A
Εάν	K	KAI	MΓ	Τότε	A
Εάν	MI	KAI	K	Τότε	M
Εάν	MI	KAI	MI	Τότε	Δ
Εάν	MI	KAI	ME	Τότε	A
Εάν	MI	KAI	MΓ	Τότε	A
Εάν	ME	KAI	K	Τότε	M
Εάν	ME	KAI	MI	Τότε	M
Εάν	ME	KAI	ME	Τότε	Δ
Εάν	ME	KAI	MΓ	Τότε	Δ
Εάν	MΓ	KAI	K	Τότε	MM
Εάν	MΓ	KAI	MI	Τότε	M
Εάν	MΓ	KAI	ME	Τότε	Δ
Εάν	MΓ	KAI	MΓ	Τότε	A

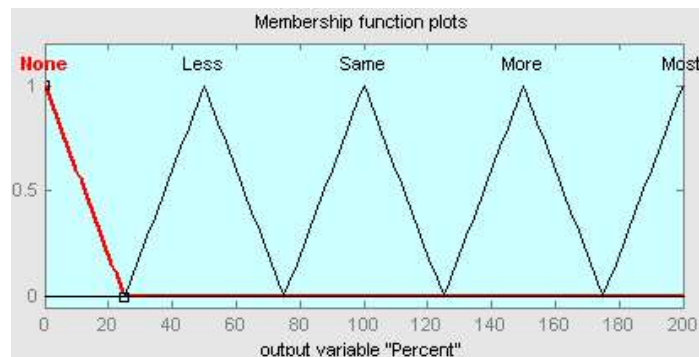
Πίνακας 4.1. Όπου K=Καθόλου, MI=Μικρή, ME=Μεσαία, MΓ=Μεγάλη και MM=Μέγιστη Μείωση  
M=Μείωση, Δ=Διατήρηση, A=Αύξηση, MA=Μέγιστη Αύξηση.

Ο παραπάνω πίνακας ακολουθεί θα έλεγε κανείς την απλή λογική. Αυτή η λογική είναι η εξής όταν υπάρχει μεγάλη αύξηση ή μείωση από το ένα χρονικό διάστημα στο άλλο υπάρχει πιο άμεση ανταπόκριση ενώ σε μικρότερες τέτοιες διαφορές έχουμε πιο ήπια προσαρμογή στην έξοδο. Ενδιαφέρον έχει η τελευταία περίπτωση όπου όταν έχουμε *Μεγάλη* ένταση και στις δύο προηγούμενες χρονικές περιόδους αυξάνουμε την ποσότητα

τροφής. Είναι μια λογική κίνηση αν σκεφτεί κανείς ότι για να διατηρείται μια τέτοια ένταση στην αίτηση τροφής σημαίνει ότι υπάρχει όλο και περισσότερη ανάγκη για τροφή.

### Mamdani

Στο κατά Mamdani σύστημα έχουμε συναρτήσεις συγγένειας σε αντίθεση με το κατά Sugeno σύστημα. Οπότε τα ασαφή σύνολα έχουν σχέση με το ποσοστό κατά το οποίο πρέπει να μεταβληθεί η προηγούμενη ποσότητα τροφής και το σύνολο αναφοράς είναι από 0-200%. Αυτά είναι *Μέγιστη Μείωση*, *Μείωση*, *Διατήρηση*, *Αύξηση* και *Μέγιστη Αύξηση*. Οι συναρτήσεις συγγένειας είναι τριγωνικής μορφής και φαίνονται παρακάτω.

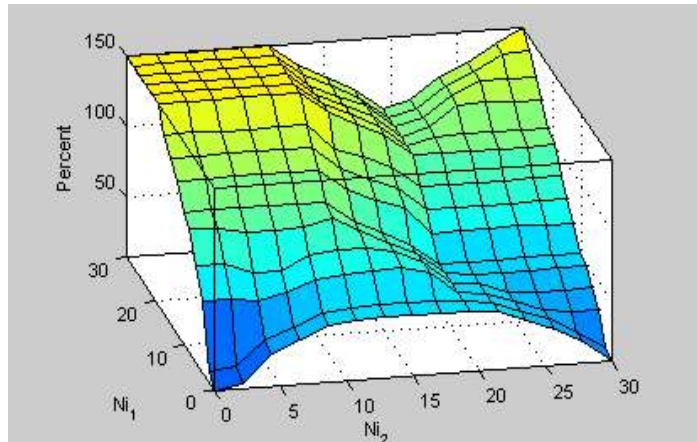


4.2 Συναρτήσεις συγγένειας των εξόδων στο Mamdani σύστημα

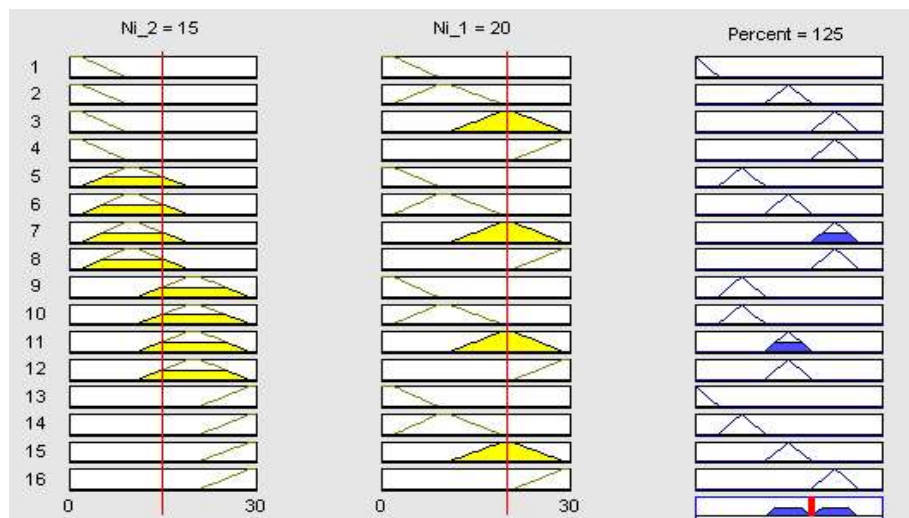
Η μέθοδος υπολογισμού του τελεστή **ΚΑΙ** είναι η μέθοδος του *ελαχίστου* ενώ για τον τελεστή **Ή** είναι η μέθοδος του *μεγίστου*. Ακόμα η μέθοδος **συνεπαγωγής** των αποτελεσμάτων για τους κανόνες είναι η μέθοδος του *ελαχίστου* ενώ για την **άθροιση** των αποτελεσμάτων κάθε κανόνα χρησιμοποιείται η μέθοδος του *μεγίστου*.

Η μέθοδος που χρησιμοποιήθηκε για την αποσαφήνιση είναι ο κεντροειδής υπολογισμός όπου υπολογίζεται το κέντρο μάζας της καμπύλης εξόδου. Στην συνέχεια παρουσιάζεται η τρισδιάστατη αναπαράσταση της επιφάνειας του συστήματος όπως προέκυψε από το εργαλείο MATLAB. Βλέπουμε τη συμπεριφορά του ΑΣΣ τύπου Mamdani.

#### 4.3 Το ΑΣΣ τύπου Mamdani



Παρατηρούμε ότι οι μεγαλύτερες αυξήσεις και μειώσεις γίνονται σε “ακραίες” μεταβολές των ώστε να επιτρέπεται η γρήγορη και ομαλή προσαρμογή. Ακολουθεί και τον τρόπο λειτουργίας του αλγορίθμου για συγκεκριμένες εισόδους στο επόμενο διάγραμμα :

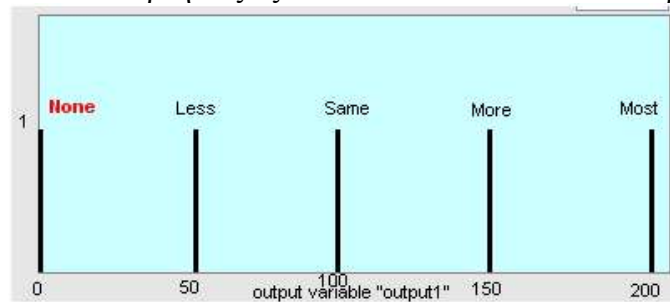


4.4 Παράδειγμα λειτουργίας του Mamdani

#### Sugeno

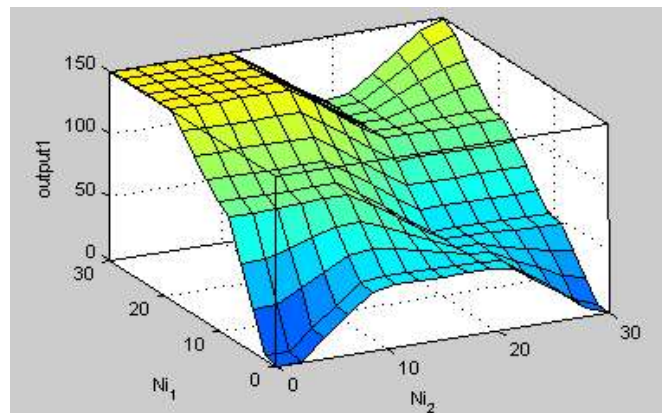
Το ΑΣΣ τύπου Sugeno που δημιουργήθηκε είναι μηδενικού βαθμού. Οι ασαφείς κανόνες είναι αυτοί που χρησιμοποιούνται και στο τύπου Mamdani και τους έχουμε ήδη

αναφέρει. Όμως εδώ οι συναρτήσεις εξόδου αποτελούνται από σταθερές.



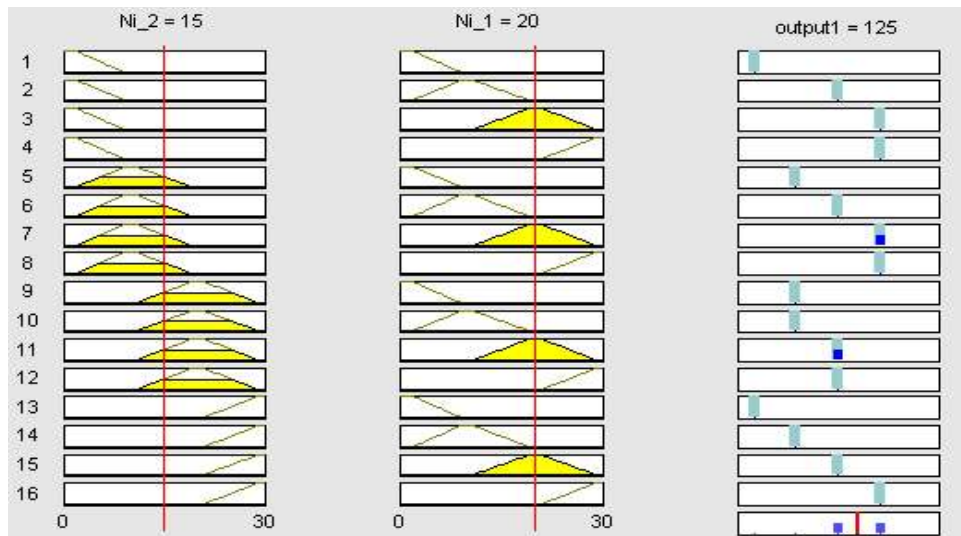
4.5 Στο κατά Sugeno έχουμε singleton εξόδους

Ασφαλώς η μεταβλητή εξόδου παραμένει η ίδια, είναι το ποσοστό κατά το οποίο θα μεταβληθεί η επόμενη ποσότητα τροφής. Αυτό που αλλάζει είναι ο τρόπος υπολογισμού της. Αφού υπολογιστεί η βαρύτητα κάθε κανόνα τότε η έξοδος είναι ο μέσος όρος(με βάρη) όλων των κανόνων. Ας δούμε και σε αυτήν την περίπτωση την επιφάνεια συμπεριφοράς του συστήματος.



4.6 Το κατά Sugeno σύστημα

Είναι φανερό πόσο πιο γραμμική είναι η έξοδος αλλά η συμπεριφορά παραμένει στην ίδια “φιλοσοφία”. Ας δούμε και ένα παράδειγμα με συγκεκριμένες εξόδους.



4.7 Παράδειγμα λειτουργίας του Sugeno

#### Αυτοπροσαρμογή της ημερήσιας ποσότητας

Η διαδικασία υπολογισμού και παροχής τροφής συνεχίζεται όλη την διάρκεια της ημέρας οπότε και υπολογίζεται συνολικά η ποσότητα τροφής που χορηγήθηκε (σημειώνεται σαν  $Q_{D-A}$ ). Στο τέλος της ημέρας επαναξιολογείται η αρχική ημερήσια ποσότητα τροφής του συστήματος. Αν η ποσότητα που δόθηκε ξεπερνάει κατά 15% την ποσότητα που είχε υπολογιστεί για τη συγκεκριμένη μέρα τότε αναπροσαρμόζουμε αυτήν την τιμή κατά 10%. Με αυτόν τον τρόπο αναπροσδιορίζεται και το  $Q_p$ , η βασική τιμή τροφής για το μισόλεπτο.

#### 4.γ ΕΠΙΚΟΙΝΩΝΙΑ

Η επικοινωνία του μικροεπεξεργαστή με κάποιο υπολογιστικό σύστημα (σταθερό ή όχι) κρίθηκε απαραίτητη. Για τον σκοπό αυτό έπρεπε να βρεθεί ένας τρόπος επικοινωνίας καθώς και ένα πρωτόκολλο. Το σύστημα επικοινωνίας έπρεπε να πληρεί τα εξής κριτήρια:

- Ικανότητα Επικοινωνίας σε απομακρυσμένη περιοχή

- Μικρή κατανάλωση Ισχύος
- Απλότητα με πολλές Δυνατότητες
- Μικρό Κόστος

Τα κριτήρια αυτά ικανοποιεί πλήρως μια μονάδα μόντεμ GSM/GPRS (κινητό τηλέφωνο). Τα χαρακτηριστικά του περιγράφονται σε άλλο κεφάλαιο, αλλά σαν μια γρήγορη αναφορά μπορεί κάποιος να πει ότι ένα τέτοιο σύστημα μπορεί να καλύψει σχεδόν όλες τις περιοχές(ανάλογα με τη κάλυψη του δικτύου), δεν έχει μεγάλη κατανάλωση, το χειρίζεσαι με απλές εντολές και έχει σχετικά μικρό κόστος.

#### Αποστολή Δεδομένων

Η λειτουργία αυτή είναι από τις πιο σημαντικές. Τα δεδομένα που αποστέλλονται είναι ο αριθμός του μισάωρου, η τροφή σε μονάδες που δόθηκε σε αυτό το μισάωρο και πόσες αιτήσεις για τροφή είχαμε. Η αποστολή γίνεται μέσω γραπτού μηνύματος. Το τελευταίο αποστέλλεται όταν γεμίσει με δεδομένα και στο τέλος κάθε ημέρας. Η μορφή των δεδομένων είναι η εξής:

<αύξων αριθμός μισάωρου> A <ποσότητα> B <Αιτήσεις> C

πχ. 14A18B150C που σημαίνει ότι στο 14ο μισάωρο είχαμε 150 αιτήσεις και δόθηκαν 18 μονάδες τροφής. Τα A,B,C λειτουργούν σαν διαχωριστικά.

4.8 Το πακέτο δεδομένων

<Halfhour>	A	<Quantity>	B	Requests	C
------------	---	------------	---	----------	---

Τα δεδομένα στέλνονται σε συγκεκριμένους αριθμούς και μέχρι την αποστολή τους αποθηκεύονται στην μνήμη.

#### Εντολές Διαχείρισης και Επιβεβαίωσης Λειτουργίας

Οι εντολές αυτές στέλλονται στον μικροεπεξεργαστή από κινητό τηλέφωνο με ένα απλό μήνυμα είτε ξεχωριστά είτε ομαδικά. Ακολουθεί η περιγραφή τους.



**Επανεκκίνηση(reboot):** Ενεργοποιείται με την αποστολή ενός **r** και έχει σαν αποτέλεσμα την επανεκκίνηση του συστήματος με τις προκαθορισμένες ρυθμίσεις. Αυτά που επηρεάζονται είναι ο αύξων αριθμός της μέρας, του μισάωρου, η αρχική ποσότητα τροφής ημέρας και η βασικά μονάδα μισάωρου, ο αριθμός αποστολής των δεδομένων, εκκαθάριση της μνήμης των δεδομένων και τέλος ορίζεται σαν αλγόριθμος τροφοδοσίας ο Crisp.

**Αλλαγή αριθμού κινητού(change):** Η εντολή είναι της μορφής **c<x><a>**, όπου **x** ποιο καταχωρημένο νούμερο θα αλλάξει(υπάρχουν δύο) και δυνατές τιμές είναι 1 ή 2. Όπου **a** ο 10ψήφιος αριθμός κινητού τηλεφώνου. πχ. Η αποστολή του “c16987654321” θα αλλάξει τον αριθμό του πρώτου καταχωρημένου τηλεφώνου. *Προσοχή* δεν μπορούμε να αλλάζουμε στέλνοντας ένα μήνυμα και τους δύο αριθμούς.

**Αλλαγή αλγορίθμου τροφοδοσίας(algorithm change):** Η εντολή είναι της μορφής **a<x>** όπου **x** ο αριθμός του αλγορίθμου που θέλουμε. Δυνατές τιμές 0 για τον Σαφή Crisp αλγόριθμο, 1 για το Ασαφές Συλλογιστικό Σύστημα τύπου Sugeno και 2 για το ΑΣΣ τύπου Mamdani.

**Ορισμός του χρόνου που θέλει ο μηχανισμός τροφοδοσίας για να προσφέρει μια μονάδα τροφής(time):** Η εντολή διατυπώνεται ως **t<x>** όπου **x** ο αριθμός των δευτερολέπτων που χρειάζεται η αυτόματη ταΐστρα για να δώσει μια μονάδα τροφής. Δυνατές τιμές 0-9. Οι τιμές είναι σε δευτερόλεπτα πχ. στέλνοντας t5 για να δοθεί μια μονάδα τροφής θα ανοίγει ο μηχανισμός για 5 δευτερόλεπτα.

**Επιλογή πλήθους κινητών για επικοινωνία(cell number):** Η μορφή της εντολής είναι **l<x>** με **x** το πλήθος των κινητών στα οποία θέλουμε να στέλνονται τα δεδομένα. Δυνατές τιμές είναι 1,2 ανάλογα αν θέλουμε να στέλνονται τα δεδομένα μας στο πρώτο κινητό ή στο δεύτερο και **b** για να στέλνονται τα μηνύματα και στα δύο(πχ. για λόγους ασφάλειας των δεδομένων).

**Αλλαγή της ποσότητας τροφής στην αρχική μονάδα χρόνου(Qp):** Αρκεί να στείλουμε το λατινικό **p** ακολουθούμενο από έναν αριθμό 0-9 για να ορίσουμε πόσες βασικές μονάδες τροφής αντιστοιχούν στην αρχική μονάδα χρόνου.

**Αποστολής Κατάστασης Συστήματος(status):** Με την αποστολή ενός **s** ζητάμε την αποστολή ενός μηνύματος με την κατάσταση του συστήματος. Το μήνυμα που θα μας σταλεί θα έχει πληροφορία για το ποιος αλγόριθμος τροφοδοσίας τρέχει, πόσες μέρες έχουν περάσει από την αρχή της λειτουργίας του ή από την τελευταία επανεκκίνηση, σε ποιο μισάωρο της ημέρας είμαστε και πόσες μονάδες τροφής έχει δώσει τη συγκεκριμένη μέρα. Πχ. Ένα πιθανό τέτοιο μήνυμα είναι “Alg=0,Day=0,HlfHour=0, Todays Quantity until now=26” .

Οι παραπάνω εντολές μας δίνουν τεράστια ευελιξία στον χειρισμό του συστήματος, δεν πρέπει κανείς να ξεχνάει ότι το σύστημα θα δουλεύει σε πολλές περιπτώσεις κάτω από δύσκολες συνθήκες(πχ. Μέσα στη θάλασσα) και το να μπορεί κάποιος να το ελέγχει από μακριά είναι κάτι παραπάνω από επιθυμητό. Ακόμα το γεγονός ότι έχουμε στη διάθεση μας τα δεδομένα δίνει την δυνατότητα για επεξεργασία, ακόμα και σε πραγματικό χρόνο, για ερευνητικούς ή όχι σκοπούς(πχ. Έλεγχος για το πότε το απόθεμα στην αποθήκη πλησιάζει στο τέλος ή για την καταγραφή των τροφικών συνηθειών των ψαριών).

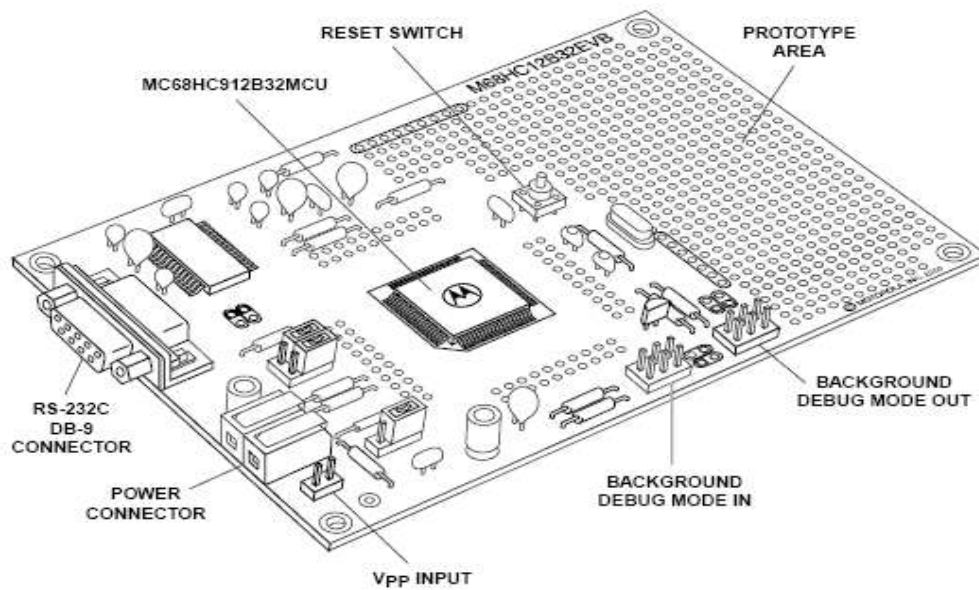
## 5. ΥΛΙΚΟ

Στο κεφάλαιο αυτό περιγράφονται όλα τα υλικά μέρη του συστήματος. Περισσότερη έμφαση δίνεται στον μικροεπεξεργαστή αφού είναι και η καρδιά αυτού του συστήματος γιατί “τρέχει” τους αλγορίθμους και αποφασίζει την ποσότητα της παρεχόμενης τροφής. Επίσης αναφέρεται σχηματικά και η συνδεσμολογία όλων των μερών αυτής της εργασίας. Η περιγραφή όλων των μερών αναλυτικά θα γίνει μόνο στα σημεία όπου πραγματικά χρειάζεται, θα αναφερθούν δηλαδή τα βασικότερα στοιχεία κάθε υλικού μέρους αλλά κυρίως τα χαρακτηριστικά που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος.

### 5.α Το αναπτυξιακό-πλακέτα επεξεργαστή

Θα ξεκινήσουμε με την περιγραφή του αναπτυξιακού. Η πλακέτα που χρησιμοποιήθηκε για τον προγραμματισμό και αποσφαλμάτωση του μικροεπεξεργαστή M68HC912B32, για την ανάπτυξη του συστήματος είναι η M68HC12B32. Το αναπτυξιακό αυτό προσφέρει τον απαραίτητο χρονισμό και τις βασικές εισόδους εξόδους. Εκτός από αυτά έχει και άλλα χαρακτηριστικά τα οποία είναι:

- Είσοδος τροφοδοσίας
- Διασύνδεση RS232C (σειριακή θύρα)
- 16 Mhz Κρύσταλλος δηλαδή λειτουργία διαύλου στα 8 Mhz
- Εισόδους/εξόδους για αποσφαλμάτωση απευθείας πάνω στον μικροεπεξεργαστή



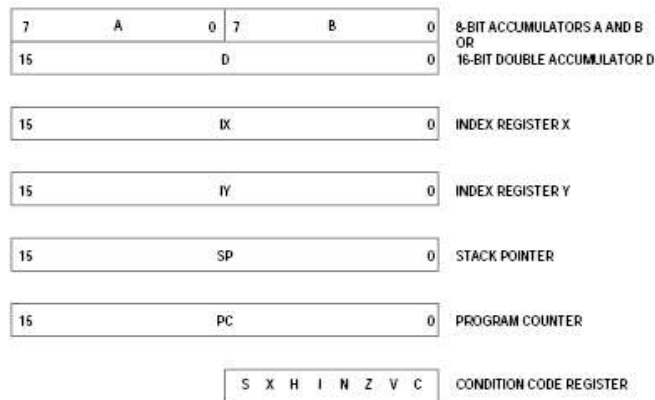
5.1 Το αναπτυξιακό

## 5.β Ο Μικροεπεξεργαστής M68HC912B32

Στη συνέχεια θα αναπτύξουμε κάποιες πληροφορίες για τον μικροεπεξεργαστή, μια μονάδα επεξεργασίας 16 bit.

### Προγραμματιστικό Μοντέλο και οργάνωση της μνήμης

Η Κεντρική Μονάδα Επεξεργασίας(ΚΜΕ) έχει δύο 8 bit καταχωρητές γενικής χρήσης οι οποίοι μπορούν να συγχωνευθούν σε έναν 16 bit καταχωρητή για συγκεκριμένες εντολές. Επίσης έχει δύο καταχωρητές διεύθυνσης(16 bit), έναν καταχωρητή για δείκτη στοίβας (stack pointer 16 bit ), έναν καταχωρητή για την αποθήκευση της επόμενης εντολής (program counter 16 bit) και έναν 8 bit για condition code register. Το προγραμματιστικό μοντέλο φαίνεται και στο σχήμα παρακάτω.



5.2 Το προγραμματιστικό μοντέλο του μικροεπεξεργαστή

Ο μικροεπεξεργαστής υποστηρίζει διάφορους τρόπους διευθυνσιοδότησης όπως άμεσης, έμμεσης, με δείκτες κ.α. . Οι εντολές έρχονται προς εκτέλεση μια προς μία και εκτελούνται μια την φορά, δηλαδή εδώ δεν έχουμε το κλασικό σύστημα της σωλήνωσης (pipeline) αλλά μια ουρά εντολών.

Η ΚΜΕ έχει ένα στάνταρ ποσό μνήμης των 64 Kbytes. Επίσης υποστηρίζει επέκταση της μνήμης με την χρήση τεχνικών σελιδοποίησης έχοντας ειδικές εντολές. Ακολουθεί ένας πίνακας με τον τρόπο που διατάσσεται η μνήμη.

Περιοχή Μνήμης	Χρήση	Περιγραφή
\$0000-\$01FF	Καταχωρητές επεξεργαστή	Καταχωρητές Εισόδου/Εξόδου,ελέγχου κτλ
\$0800-\$0BFF	Κώδικας Χρήστη/ Δεδομένα	1 Kbyte RAM
\$0D00-\$0FFF	Κώδικας Χρήστη/ Δεδομένα	768 bytes EEPROM
\$8000-\$F77F	Κώδικας Χρήστη/ Δεδομένα	32 Kbytes Flash EEPROM
\$F780-\$F7FF	Καθρέφτης του πίνακα διακοπών	
\$F800-\$FBFF	Θέση για επέκταση του bootloader	
\$FC00-\$FBFF	EEPROM bootloader	
\$FFC0-\$FFFF	Πίνακας των διακοπών	

Πίνακας 5.1 χάρτης της μνήμης

Στην αρχιτεκτονική του M68HC12 όλη η μνήμη και οι είσοδοι/έξοδοι αντιστοιχίζονται στον κοινό χώρο των 64Kbyte. Αυτό επιτρέπει το ίδιο σύνολο εντολών να χρησιμοποιείται για να προσπελάσει μνήμη, εισόδους εξόδους και καταχωρητές ελέγχου. Το σύνολο εντολών αποτελείται από γενικού σκοπού εντολών αποθήκευσης, μεταφοράς φορτώματος και ανταλλαγής που διευκολύνουν την μεταφορά δεδομένων από και προς την μνήμη ή των περιφερειακών. Επίσης έχει ένα πλήρες σύνολο μαθηματικών εντολών (8 και 16 bit), για πράξεις με προσημασμένους και μη προσημασμένους αριθμούς. Ακόμα διαθέτει ειδικές αριθμητικές και λογικές εντολές για χειρισμό της στοίβας, για διευθυνσιοδότηση, υπολογισμό BCD και για χειρισμό του CCR. Επιπλέον υπάρχουν, και αυτός ήταν ένας από τους λόγους που χρησιμοποιήθηκε ο συγκεκριμένος μικροεπεξεργαστής, αφοσιωμένες εντολές για πολλαπλασιασμό και συσσώρευση, αντιστοίχιση πινάκων και για εφαρμογές ασαφούς λογικής.

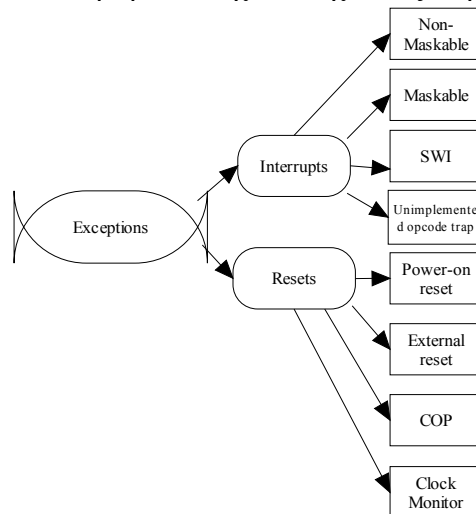
### **Εξαιρέσεις**

Ένα πολύ σημαντικό χαρακτηριστικό του μικροεπεξεργαστή είναι η δυνατότητα του να χειρίζεται εξαιρέσεις. Με τον όρο εξαιρέσεις εννοούνται τα συμβάντα τα οποία απαιτούν εξυπηρέτηση έξω από τη συνηθισμένη ροή εκτέλεσης των εντολών. Στις εξαιρέσεις περιέχονται οι:

- Οι επανεκκινήσεις: Από το ρεύμα, από την ακίδα επανεκκίνησης, από το ρολόι και από τον έλεγχο σωστής λειτουργίας. Ο έλεγχος σωστής λειτουργίας δεν είναι παρά η περιοδική χρήση συγκεκριμένων εντολών. Όταν το πρόγραμμα μας αποτύχει να ακολουθεί αυτήν την περίοδο τότε το σύστημα επανεκκινεί. Η επανεκκίνηση από το ρολόι συμβαίνει όταν η συχνότητα του πέσει κάτω από κάποια συγκεκριμένα όρια.
- Μη υλοποιημένα interrupt: Ο M68HC12 έχει 256 interrupt από τα οποία μόνο τα 54 είναι υλοποιημένα. Δηλαδή μόνο για αυτά υπάρχει αντιστοιχία για κάθε ένα ξεχωριστά στον πίνακα διακοπών και συγκεκριμένη λειτουργικότητα. Για τα υπόλοιπα 202 υπάρχει μια καταχώρηση. Έτσι αν η ΚΜΕ πάει να εκτελέσει ένα από αυτά τα 202 θα προκαλέσει μία μη υλοποιημένη παγίδα.

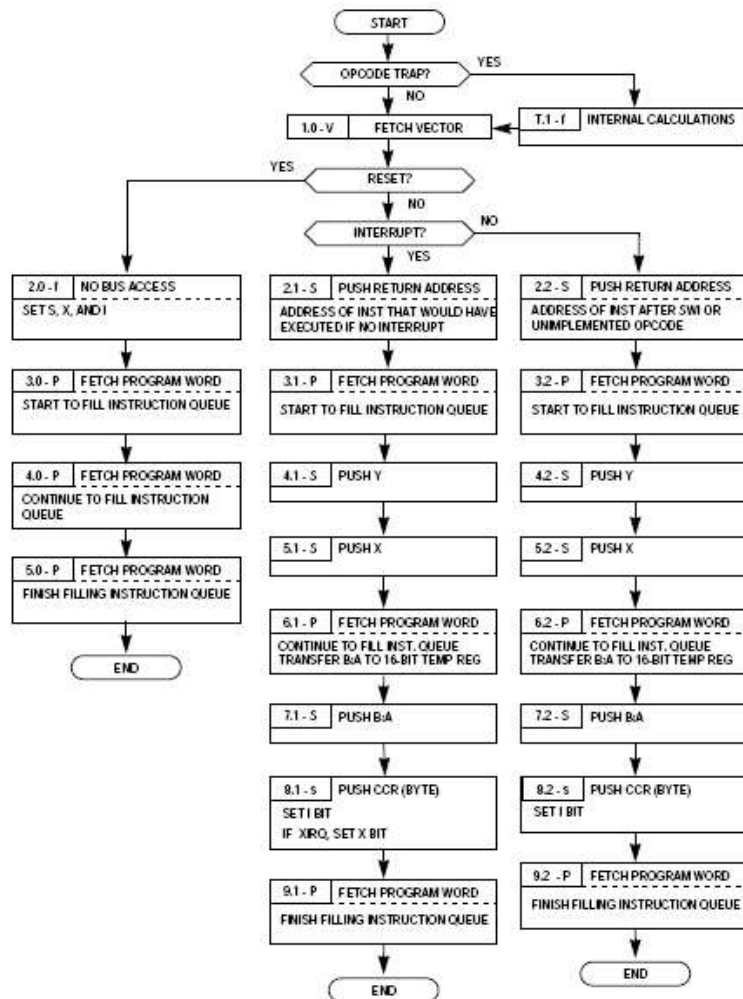
- Interrupt μέσω εντολής προγράμματος.
- Καλυπτόμενα interrupt: Πρόκειται για interrupt τα οποία μπορούν να προκύψουν αλλά δεν εξυπηρετούνται αν δεν έχει τεθεί το I bit στον CCR.
- Μη καλυπτόμενα interrupt: Από τη στιγμή που θα ενεργοποιηθούν δεν βρίσκονται πίσω από καμία κάλυψη-μάσκα

Πως λειτουργούν όμως οι εξαιρέσεις; Η διαδικασία είναι η ίδια αλλά ανάλογα την εξαίρεση αλλάζει ο τρόπος. Η σειρά είναι η εξής γίνεται αίτηση για εξαίρεση, ένα ενσωματωμένο σύστημα καταλαβαίνει για τι τύπου πρόκειται και κάνει βασικές ρυθμίσεις, στην συνέχεια αναλαμβάνει η ΚΜΕ να μεταπηδήσει τον έλεγχο του προγράμματος στην θέση που υποδεικνύει η θέση της εξαίρεσης στον πίνακα εξαιρέσεων. Εδώ πρέπει να γίνει ξεκάθαρο ότι άλλο εξαίρεση και άλλο interrupt. Το interrupt είναι ένας τύπος εξαιρέσεων (ένας άλλος είναι οι επανεκκινήσεις). Το σχήμα παρακάτω βοηθάει στην κατανόηση όσων έχουν λεχθεί ως τώρα.



5.3 Διάγραμμα των εξαιρέσεων

Επίσης στο επόμενο διάγραμμα φαίνεται ο τρόπος με τον οποίο εξυπηρετούνται οι εξαιρέσεις.



5.4 Διάγραμμα ροής χειρισμού των εξαιρέσεων

Και τέλος παρουσιάζονται όλα τα interrupt που έχουμε στη διάθεση μας



Vector Address	CPU Interrupt
\$FFC0–FFCF	Reserved
\$FFD0	BDLC (J1850)
\$FFD2	ATD
\$FFD4	Reserved
\$FFD6	SCI0
\$FFD8	SPI
\$FFDA	Pulse Acc. Input Edge
\$FFDC	Pulse Acc. Overflow
\$FFDE	Timer Overflow
\$FFE0	Timer Channel 7
\$FFE2	Timer Channel 6
\$FFE4	Timer Channel 5
\$FFE6	Timer Channel 4
\$FFE8	Timer Channel 3
\$FFEA	Timer Channel 2
\$FFEC	Timer Channel 1
\$FFEE	Timer Channel 0
\$FFF0	Real-Time Interrupt
\$FFF2	$\overline{\text{IRQ}}$
\$FFF4	$\overline{\text{XIRQ}}$
\$FFF6	$\overline{\text{SWI}}$
\$FFF8	Illegal Opcode Trap
\$FFFA	COP Failure Reset
\$FFFC	Clock Mon. Fail Reset
\$FFFE	$\overline{\text{RESET}}$

5.4.β Ο πίνακας των Interrupt

### Υποστήριξη Ασαφούς Λογικής

Ο συγκεκριμένος μικροεπεξεργαστής έχει και εντολές αποκλειστικά για τον χειρισμό ασαφούς λογικής. Επίσης διαθέτει εντολές που είναι πολύ χρήσιμες σε τέτοιες εφαρμογές. Οι αποκλειστικές εντολές είναι:

- Η εντολή MEM όπου καθορίζει το βαθμό που ανήκει μια μεταβλητή σε μια συνάρτηση συγγένειας.
- Η εντολή REVW και REV όπου αξιολογεί τους ασαφείς κανόνες με βάρη ή χωρίς.
- Η εντολή WAV που εφαρμόζει αποσαφήνιση τύπου Sugeno.

### 5.γ Το Κινητό Τηλέφωνο

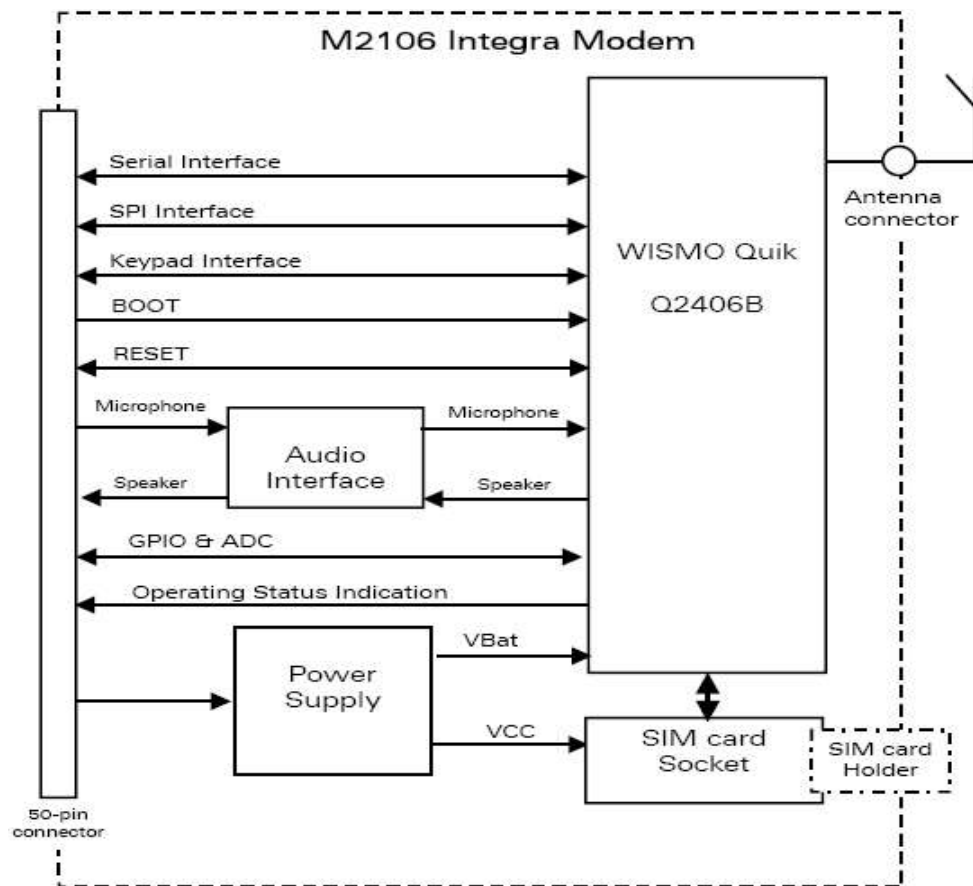
Το κινητό τηλέφωνο που χρησιμοποιήσαμε είναι της εταιρείας WAVECOM. Πρόκειται για ένα E-GSM/GSM-GPRS 900/1800 μόντεμ(GPRS κλάσης 10). Διαθέτει μία θύρα 50 ακίδων σαν θύρα γενικού σκοπού, θύρα για την SIM, θύρα για την κεραία και για τροφοδοσία.



Οι υπηρεσίες που παρέχει σε GSM/GPRS περιγράφονται στον παρακάτω πίνακα:

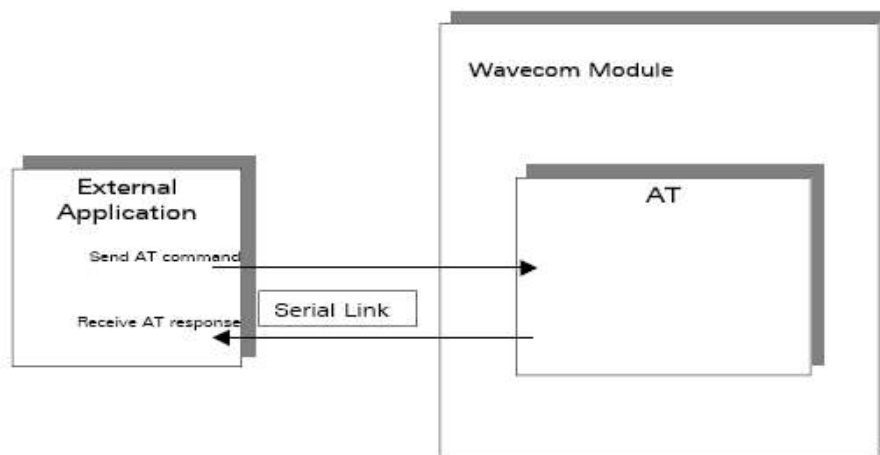
Standard	900 MHz, E-GSM compliant ; Class 4 (2W). 1800 MHz ; Class 1 (1W). GSM phase 2. GPRS Class 10.
Interface	Serial interface V.24/CMOS AT command set based on V.25ter and GSM 07.05 & 07.07. Auto-bauding function between 2400 bits/s and 19200 bits/s No auto-framing available
Audio	Half rate / Full rate / Enhanced Full rate operation. Accessories (options): Handset, Car Kit.
SMS	Mobile Originated (MO) and Mobile Terminated (MT). Mode Text & PDU point to point. Cell broad cast. In accordance with GSM 07.05
Data	Asynchronous 2400, 4800, 9600, 14400 bits/s. Transparent and Non Transparent mode In Non Transparent Mode: 300, 1200, 1200/75 bauds. Mode 3.1 KHz (PSTN) and V110 (ISDN)
Fax	2400/4800/7200/9600 bits/s, GSM teleservice 62 in Transparent Mode.. Class 1 & Class 2. Group 3 compatible.
GPRS	Class 10. Coding schemes: CS1 to CS4 Compliant with SMG31bis
SIM	Plug-In SIM 3V only Full compatible with GSM 11.11 SIM Toolkit Release 97

Πίνακας 5.2 Χαρακτηριστικά του GSM/GPRS μόντεμ



5.5 Η αρχιτεκτονική του μόντεμ

Ενώ στο παραπάνω διάγραμμα φαίνεται η αρχιτεκτονική του. Η επικοινωνία με το GSM/GPRS μόντεμ γίνεται δια μέσου σειριακής επικοινωνίας όσον αφορά τη σύνδεση και με μηνύματα βασισμένα σε ATtention COMMANDS(εντολές προσοχής) όσον αφορά την χρησιμοποίηση των υπηρεσιών GSM. Οι εντολές αυτές πάντα ξεκινάνε με AT και τελειώνουν με <cr>(carriage return). Οι απαντήσεις από τη μεριά του μόντεμ πάντα αρχίζουν και τελειώνουν με <cr><lf>. Παρακάτω φαίνεται ένα σχεδιάγραμμα της επικοινωνίας αυτής.



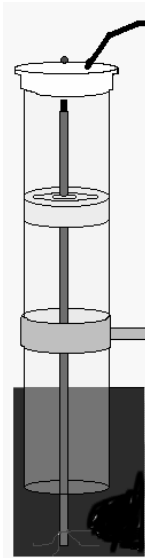
5.6 Αναπαράσταση της επικοινωνίας μεταξύ της εφαρμογής μας και του κινητού τηλεφώνου- μόντεμ

### 5.8 Αυτοχειριζόμενος μοχλός

Πρόκειται για έναν εξελιγμένο μηχανισμό που δέχεται "μηνύματα" από τα ψάρια και στη συνέχεια τα μετατρέπει σε κατάλληλη μορφή προς χρήση από ηλεκτρονικό υπολογιστή ο οποίος θα συνεκτιμά παράλληλα και άλλες παραμέτρους πριν τη χορήγηση της ποσότητας τροφής που θα πρέπει να διανεμηθεί. Ο παραπάνω μηχανισμός κατασκευάστηκε λαμβάνοντας υπόψη τα εξής:

1. α) στεγανότητα στην υψηλή υγρασία,
2. β) ανθεκτικότητα στις καιρικές συνθήκες (ήλιος, άνεμος, βροχή και κυματισμό),
3. γ) θωράκιση σε χτυπήματα,
4. δ) απλότητα ηλεκτρικών κυκλωμάτων,
5. ε) ευκολία επιδιόρθωσης.

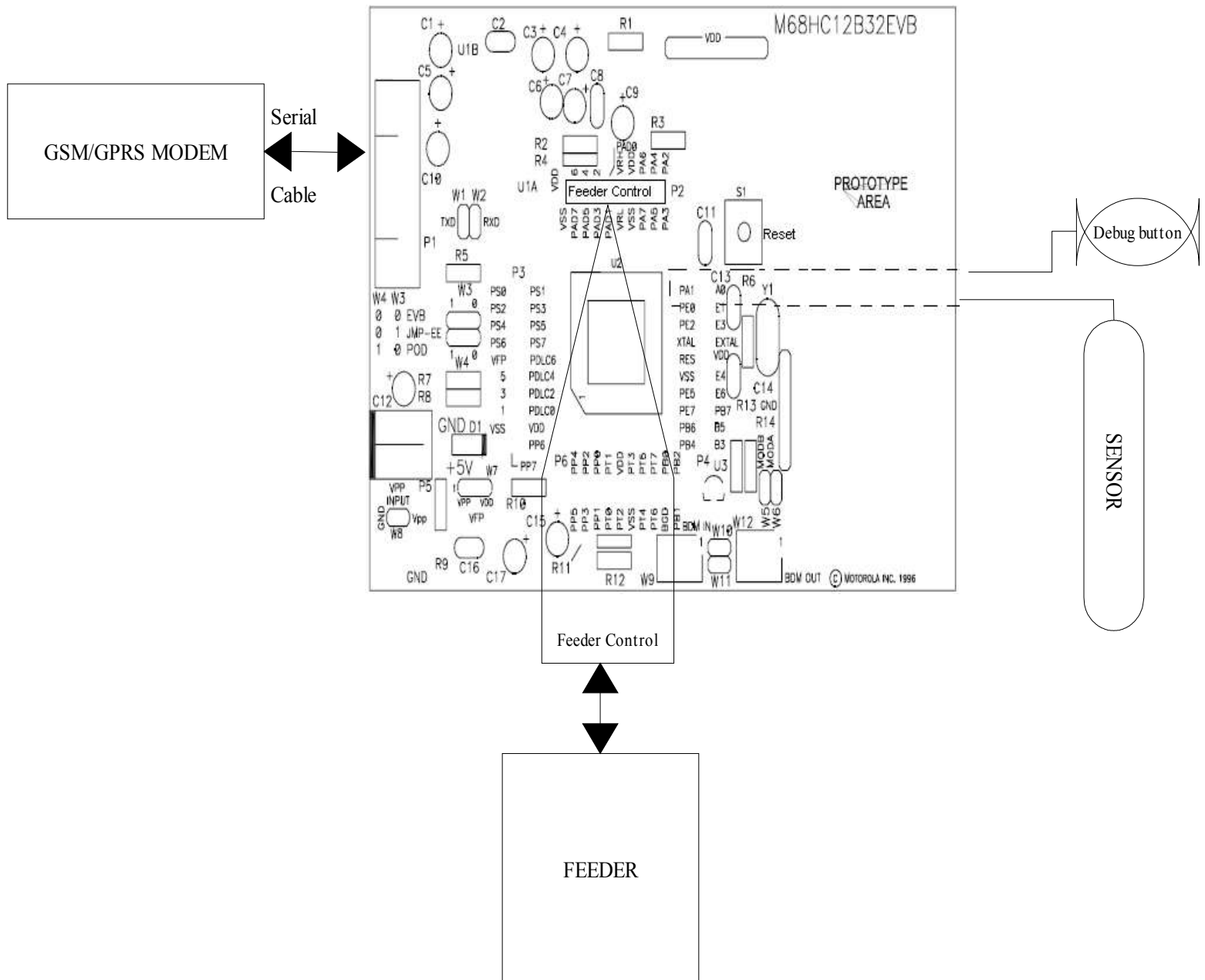
Η όλη κατασκευή θωρακίζεται (σχηματική αναπαράσταση στην Εικόνα 5.7) μέσα σε έναν σωλήνα που στο ένα του άκρο προεξέχει μια ράβδος που κινείται από τα ψάρια. Το άλλο άκρο της ράβδου φέρει μαγνήτη, ο οποίος βρίσκεται στο πεδίο ενός ειδικού ηλεκτρομαγνητικού διακόπτη.



5.7 Το αισθητήριο – μοχλός

Η οποιαδήποτε μετακίνηση της ράβδου μεταφράζεται σε ηλεκτρικό παλμό στα άκρα του διακόπτη. Με βάση ειδικό αντίβαρο η κίνηση της ράβδου ρυθμίζεται σύμφωνα με τις απαιτήσεις των ψαριών. Ο συγκεκριμένος μηχανισμός εφαρμόζεται με επιτυχία σε συνθήκες καλλιέργειας στις εγκαταστάσεις του ΕΛ.ΚΕ.ΘΕ.

Το τελευταίο σχεδιάγραμμα μας δείχνει την τελική συνδεσμολογία.



5.8 Η τελική συνδεσμολογία

Το παραπάνω αποτελεί πρωτότυπο. Σκοπός είναι να αναπτυχθεί μια ολοκληρωμένη πλακέτα αμέσως μετά την αξιολόγηση του συστήματος από το ΕΛ.ΚΕ.ΘΕ. . Επίσης να αναφερθεί ότι η όλη κατασκευή συσκευάστηκε σε ηλεκτρολογικό στεγανό κουτί για προστασία από την υγρασία, τον ήλιο και άλλους παράγοντες που θα μπορούσαν να

προκαλέσουν φθορά.

## 6. ΛΟΓΙΣΜΙΚΟ

Στην ενότητα αυτή περιγράφονται τα εργαλεία που χρησιμοποιήθηκαν για την δημιουργία της προσομοίωσης του συστήματος, για την ανάπτυξης του κώδικα για τον μικροεπεξεργαστή, για την αποσφαλμάτωση και για τον προγραμματισμό του μικροεπεξεργαστή. Επίσης περιγράφεται το λογισμικό όσον αφορά τη φιλοσοφία του και τις κύριες ρουτίνες του. Δεν θα γίνει ανάλυση σε βάθος κώδικα αφού δεν είναι αυτός ο σκοπός του κειμένου και ασφαλώς θα γίνει αναφορά στο λογισμικό της προσομοίωσης.

### 6.α Εργαλεία

**Matlab:** Το γνωστό εργαλείο της Mathworks χρησιμοποιήθηκε για την δημιουργία της προσομοίωσης. Έχει ειδικές ρουτίνες για τον χειρισμό της ασαφούς λογικής και την δυνατότητα χειρισμού από γραφικό περιβάλλον. Ένα πολύ γνωστό εργαλείο δεν θα αναφερθεί κάτι παραπάνω για αυτό, χρησιμοποιήθηκε για να δημιουργηθεί ένα εικονικό μοντέλο του συστήματος.

**CodeWarrior:** Το συγκεκριμένο εργαλείο προέρχεται από τη Metrowerks. Πρόκειται για ένα περιβάλλον ανάπτυξης εφαρμογών για τον επεξεργαστή HC12 σε γλώσσα Assembly, C και C++ . Έχει ενσωματωμένο επεξεργαστή κειμένου και μεταγλωττιστές για τις γλώσσες που προαναφέρθηκαν. Με αυτό δημιουργήθηκε ο κώδικας για τον μικροεπεξεργαστή.

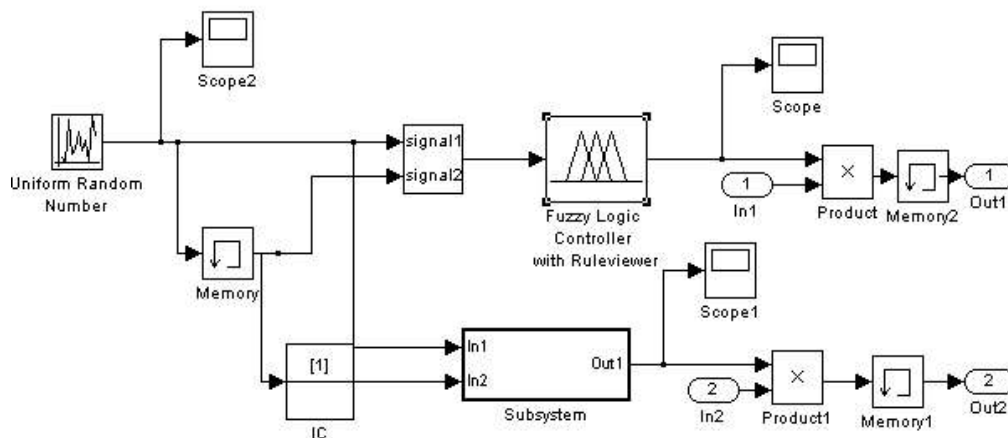
**MiniIDE:** Άλλο ένα περιβάλλον ανάπτυξης εφαρμογών με την γλώσσα Assembly. Το συγκεκριμένο εργαλείο έχει και ενσωματωμένο τερματικό σειριακής επικοινωνίας και χρησιμοποιήθηκε για το “κατέβασμα” του κώδικα στον μικροεπεξεργαστή καθώς και για αποσφαλμάτωση. Πρόκειται για ελεύθερα διανεμημένη εφαρμογή.

**Sim68HC12:** Άλλη μια ελεύθερη εφαρμογή. Πρόκειται για προσομοιωτή συμπεριφοράς του μικροεπεξεργαστή. Δέχεται .s19 αρχεία και χρησιμοποιήθηκε μαζί με τον προσομοιωτή της Metrowerks για αποσφαλμάτωση και προσομοίωση.

## 6.β Περιγραφή Λογισμικού

### Προσομοίωση Συστήματος

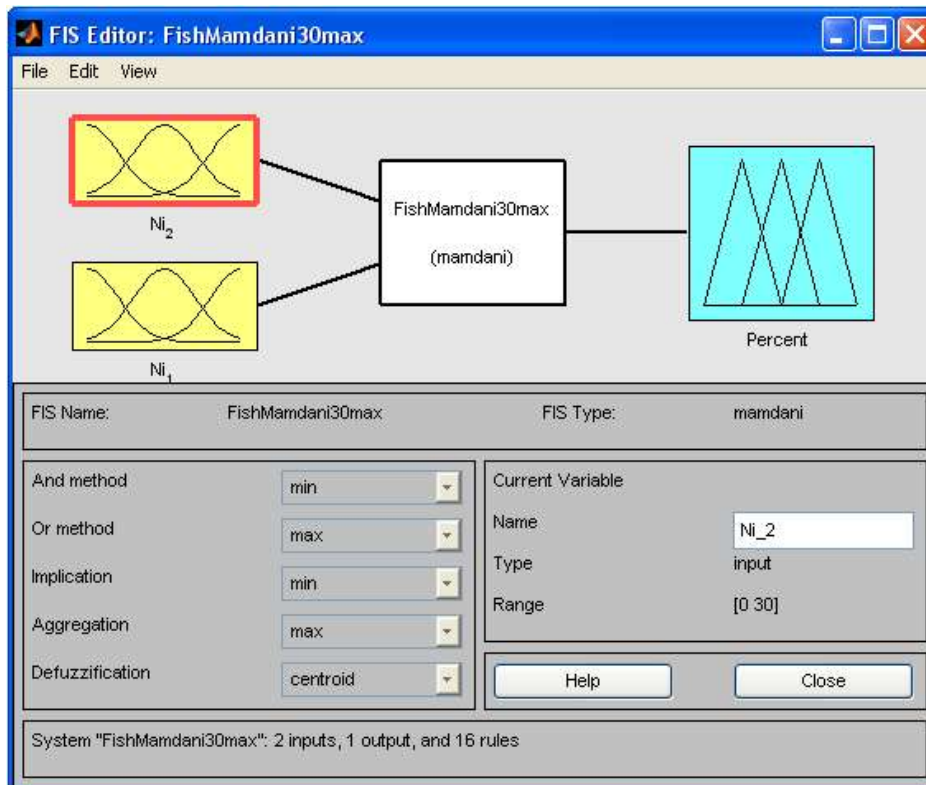
Η προσομοίωση έγινε με το εργαλείο Matlab και συγκεκριμένα με τα πακέτα εργασίας Simulink και Fuzzy Logic. Φτιάχτηκε με αυτόν τον τρόπο ένα μοντέλο. Σε αυτό το μοντέλο έχουμε δύο εισόδους δύο εξόδους. Η μία έξοδος είναι η έξοδος του ασαφούς ελεγκτή και οι άλλη του σαφούς. Οι εισοδοί παράγονται από μια μηχανή τυχαίων αριθμών με ομοιόμορφη κατανομή. Δημιουργώντας αυτό το μοντέλο περιγράφηκε η επιθυμητή συμπεριφορά που θέλαμε να έχει το σύστημα μας. Βέβαια το μοντέλο δεν είναι ακριβής αναπαράσταση του τελικού(πχ. το τελικό έχει ασφαλώς μόνο μια έξοδο) αλλά η προσομοίωση των κύριων χαρακτηριστικών του (αλγόριθμοι τροφοδοσίας). Το μοντέλο δίνεται ακριβώς παρακάτω.



6.1. Προσομοίωση με Simulink



Το μπλοκ στο οποίο υλοποιείται ο σαφής αλγόριθμος είναι το *subsystem* ενώ ο ασαφής στο *Fuzzy Logic Controller*. Τα  $in1,2$  είναι οι τιμές των εξόδων των προηγούμενων μισόλεπτων για τον ασαφή και τον σαφή αλγόριθμο αντίστοιχα. Η δημιουργία του σαφούς αλγορίθμου με το Simulink ήταν απλή διαδικασία. Για τον ασαφή όμως χρειάστηκε η χρησιμοποίηση του εξειδικευμένου πακέτου εργασίας της Matlab Fuzzy Logic Toolbox. Για την δημιουργία των συναρτήσεων συγγένειας εισόδων και εξόδων διαθέτει γραφικό περιβάλλον όπως επίσης και για την εισαγωγή των κανόνων η τελική συνεπτυγμένη μορφή αυτού του γραφικού βοηθήματος έχει την παρακάτω μορφή. Η είσοδος σε ακριβείς λεπτομέρειες, χειρισμού του πακέτου εργασίας, θεωρείται τετριμμένη.

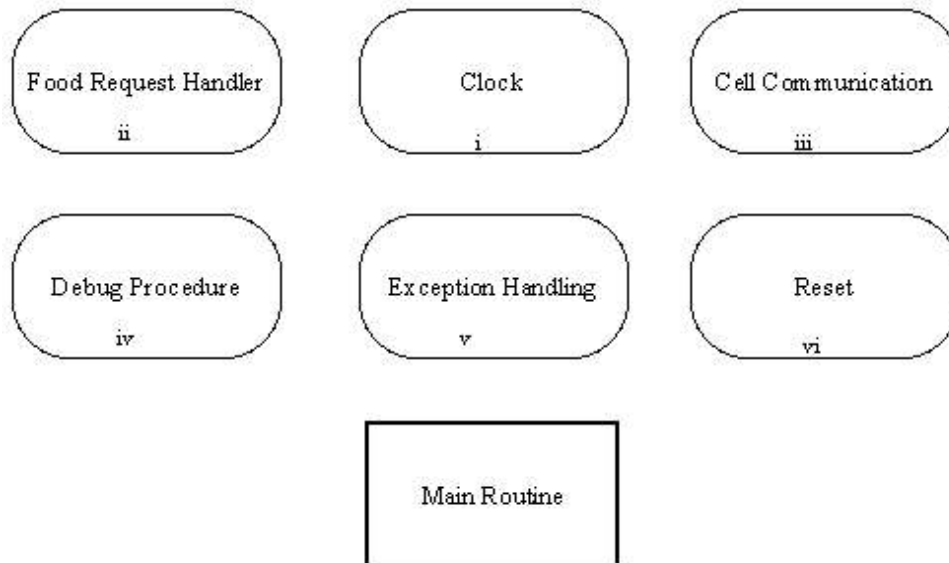


6.2  
Το γραφικό  
περιβάλλον  
δημιουργίας  
ασαφών  
αλγορίθμων

### Το Λογισμικό του συστήματος

Εδώ θα αναφερθεί πλήρως όλο το λογισμικό του συστήματος, αξίζει να επαναληφθεί ότι δεν θα γίνει καμία αναφορά σε αδιάφορες αναλύσεις κώδικα. Ξεκινώντας θα

παρατεθεί το διάγραμμα ροής και θα γίνει μια αρχική αναφορά, στην συνέχεια θα αναλυθούν τα κομμάτια που παρουσιάζουν το περισσότερο ενδιαφέρον.



6.3 ΤΟ ΔΙΑΓΡΑΜΜΑ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ(ΑΝΑΛΥΕΤΑΙ ΣΤΗ ΣΥΝΕΧΕΙΑ)

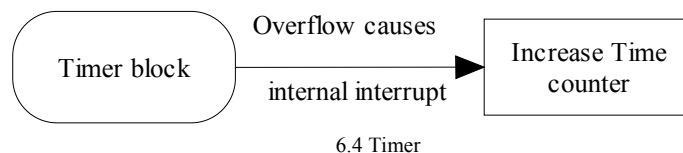
Κατά αρχήν θα περιγραφούν οι ρουτίνες των interrupt. Οι ρουτίνες αυτές, που στο διάγραμμα ροής σημειώνονται σαν ξεχωριστοί κύκλοι, δεν ανήκουν στην συνηθισμένοι σειριακή λειτουργία του λογισμικού. Πρόκειται για ρουτίνες οι οποίες ενεργοποιούνται από το υλικό είτε εσωτερικά(πχ από το ρολόι του μικροεπεξεργαστή) είτε εξωτερικά (πχ. από την διακύμανση της τάσης σε κάποια από τις ακίδες του μικροεπεξεργαστή). Όταν αυτές ενεργοποιούνται σταματάει η κανονική ροή του προγράμματος και συνεχίζει από το σημείο που υποδεικνύει το interrupt, όταν αυτή εξυπηρετηθεί τότε συνεχίζεται κανονικά το πρόγραμμα από εκεί που σταμάτησε. Οι ρουτίνες των interrupts χρησιμοποιήθηκαν για τους εξής λόγους i) **μέτρηση πραγματικού χρόνου** ii) **χειρισμός των αιτήσεων τροφής** iii) **επικοινωνία με το κινητό τηλέφωνο** iv) **αναφορά παραμέτρων μέσω σειριακής( για αποσφαλμάτωση )** v) **χειρισμός εξαιρέσεων** vi) **Επαναφορά ( Reset).**

#### Μέτρηση Πραγματικού Χρόνου

Πρόκειται για ένα εσωτερική interrupt που ενεργοποιείται από τον χρονομετρητή (timer) του μικροεπεξεργαστή. Ο χρονομετρητής είναι ένας 16 bit καταχωρητής ο οποίος αυξάνεται, όταν ενεργοποιηθεί, κατά ένα σύμφωνα με το διπλάσιο της περιόδου του κρυστάλλου χρονισμού του αναπτυξιακού μας προς τον διαιρέτη που έχουμε επιλέξει. Συγκεκριμένα ο κρύσταλλος μας λειτουργεί στα 16 Mhz, ο διαιρέτης που επιλέξαμε είναι 32 οπότε έχουμε σε ένα δευτερόλεπτο  $16\text{Mhz}/2*32 = 250000$  αυξήσεις της τιμής του χρονομετρητή. Ή αλλιώς 1 αύξηση κάθε 4μs.

Η ρουτίνα του interrupt μπορεί να ενεργοποιηθεί με διάφορους τρόπους πχ. όταν φτάσει σε μια συγκεκριμένη μέτρηση ή όταν φτάσει ο χρονομετρητής στα όρια της μέτρησης του και μηδενιστεί. Έχει επιλεγθεί ο δεύτερος τρόπος. Αυτό σημαίνει ότι, ο χρονομετρητής ως 16 bit μπορεί να μετράει από το 0 έως και το 65535 άρα εφόσον έχουμε μια αύξηση κάθε 4μs η ρουτίνα θα εξυπηρετείται κάθε 0,26124 sec.

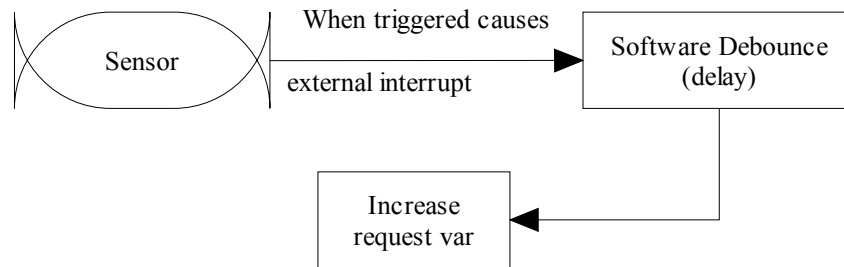
Ποια είναι όμως αυτή η ρουτίνα που εξυπηρετείται; Η ρουτίνα δεν κάνει τίποτα άλλο από το να αυξάνει την τιμή μιας μεταβλητής. Πως βοηθάει αυτό; Με αυτό τον τρόπο έχουμε φτιάξει ένα ψηφιακό ρολόι μεγάλης ακρίβειας για να μετράμε τα μισόλεπτα. Με τις δεδομένες στιγμές ένα μισόλεπτο έχει περάσει όταν η τιμή της μεταβλητής μας γίνει 114 αφού  $114*0,26124=29,88396$  sec. Με την διαδικασία αυτή γνωρίζουμε και συντονίζουμε το σύστημα μας.



### Χειρισμός των Αιτήσεων Τροφής

Ο χειρισμός των αιτήσεων τροφής γίνεται από εξωτερικό interrupt. Όπως έχουμε δει, στην συνδεσμολογία του συστήματος, στην ακίδα εξωτερικών interrupt είναι συνδεδεμένο το αισθητήριο. Όταν δώσει λοιπόν παλμό (στην πραγματικότητα στην αρνητική ακμή του παλμού) ενεργοποιείται η ρουτίνα. Η λειτουργία της είναι και εδώ απλή απλά αυξάνει μια μεταβλητή-μετρητή και αλλάζει την κατάσταση στο led ελέγχου

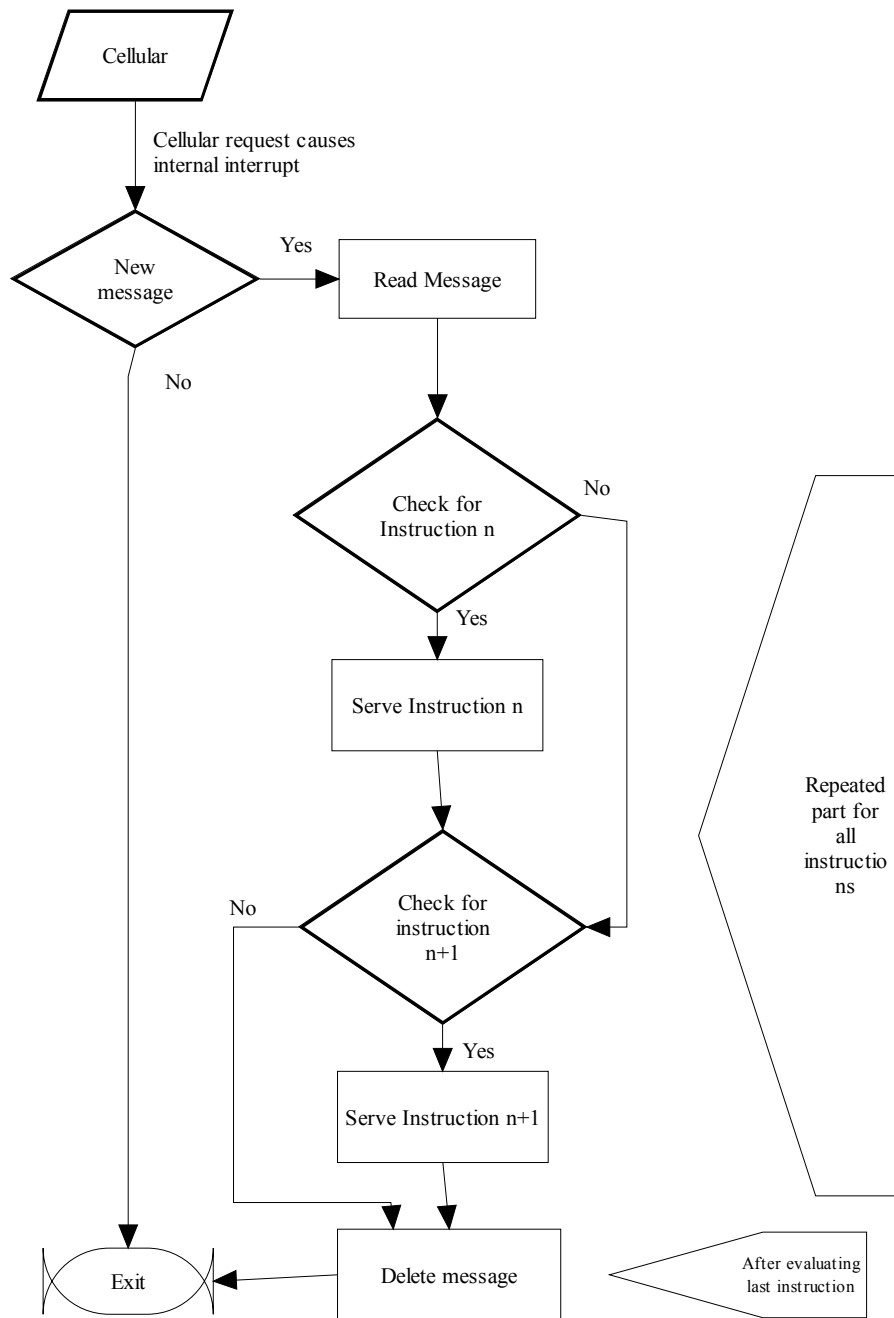
της λειτουργίας. Κάνει όμως και κάτι άλλο σημαντικό και αυτό είναι το software debounce δηλαδή η απορρόφηση(μέσω λογισμικού) μη καθαρού σήματος παλμού δημιουργώντας μια χρονοκαθυστέρηση. Με τον τρόπο αυτό δεν κινδυνεύουμε να μετρήσουμε παραπάνω αιτήσεις από όσες πραγματικά πρέπει αλλά και μειώνουμε το ποσοστό σφάλματος αποτυχίας του υλικού.



6.5 IRQ

### Επικοινωνία με το Κινητό Τηλέφωνο

Κατά αρχήν να λυθεί ίσως η απορία γιατί χρειάζεται μια ρουτίνα εκτός κανονικής ροής για την επικοινωνία. Η απάντηση είναι απλή είναι γνωστό πότε χρειάζεται η επικοινωνία του επεξεργαστή με τον έξω κόσμος αλλά όχι πότε χρειάζεται ο έξω κόσμος να επικοινωνήσει με τον επεξεργαστή. Η ρουτίνα αυτή ενεργοποιείται όταν υπάρχει κίνηση στην σειριακή θύρα που σημαίνει ότι το κινητό θέλει να επικοινωνήσει. Καταγράφεται αυτό που στέλνει και αξιολογείται. Αν η μονάδα του τηλεφώνου δηλώσει ότι ήρθε νέο μήνυμα τότε ζητάει ο επεξεργαστής το περιεχόμενο του μηνύματος για να το ελέγξει για τυχόν εντολές. Ο έλεγχος γίνεται σειριακά και εφόσον υπάρχουν τις εκτελεί( σειριακά πάλι), στο τέλος διαγράφει το μήνυμα για να μην έχει πρόβλημα μνήμης η μονάδα του τηλεφώνου.

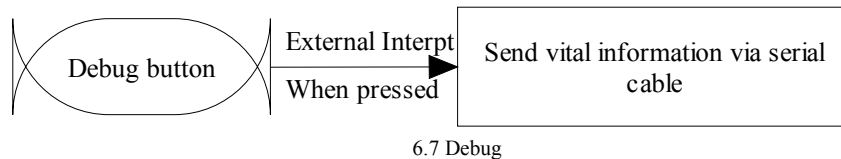


6.6 Χειρισμός της επικοινωνίας με το κινητό

### Αναφορά Παραμέτρων μέσω Σειριακής( για αποσφαλμάτωση )

Άλλη μια ρουτίνα εξωτερικού interrupt αυτή μάλιστα δεν έχει μάσκα( εκτελείται πάντα και έχει μεγάλη προτεραιότητα). Όπως φαίνεται στην συνδεσμολογία, η ακίδα του μικροεπεξεργαστή που αντιστοιχεί σε αυτό το interrupt συνδέεται με ένα κουμπί. Ο

λόγος ύπαρξης της είναι ο εξής, όταν κάποιος θέλει να ελέγξει το σύστημα κατά πόσο λειτουργεί κανονικά και βρίσκεται κοντά τότε ενώνει ένα φορητό τερματικό( πχ. έναν υπολογιστή παλάμης) στην σειριακή και με το πάτημα του κουμπιού στέλνονται από τον μικροεπεξεργαστή δεδομένα στο τερματικό. Αυτά είναι: τα δεδομένα που έχει μαζέψει και δεν έχει στείλει ακόμα, το κείμενο του τελευταίου μηνύματος που παρέλαβε, ο καταχωρητής προσωρινής απάντησης (προσωρινός καταχωρητής για την ενδοσυνεννόηση μικροεπεξεργαστή και κινητού), ο αλγόριθμος που “τρέχει”, η μεταβλητή ελέγχου της επανεκκίνησης και το νούμερο του βασικού τηλεφώνου επικοινωνίας.

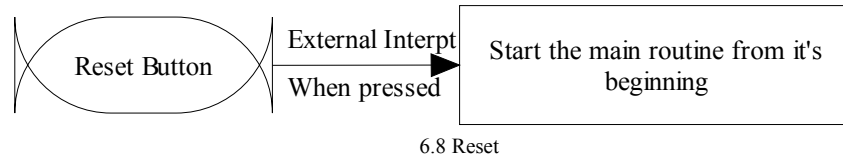


### Χειρισμός Εξαιρέσεων

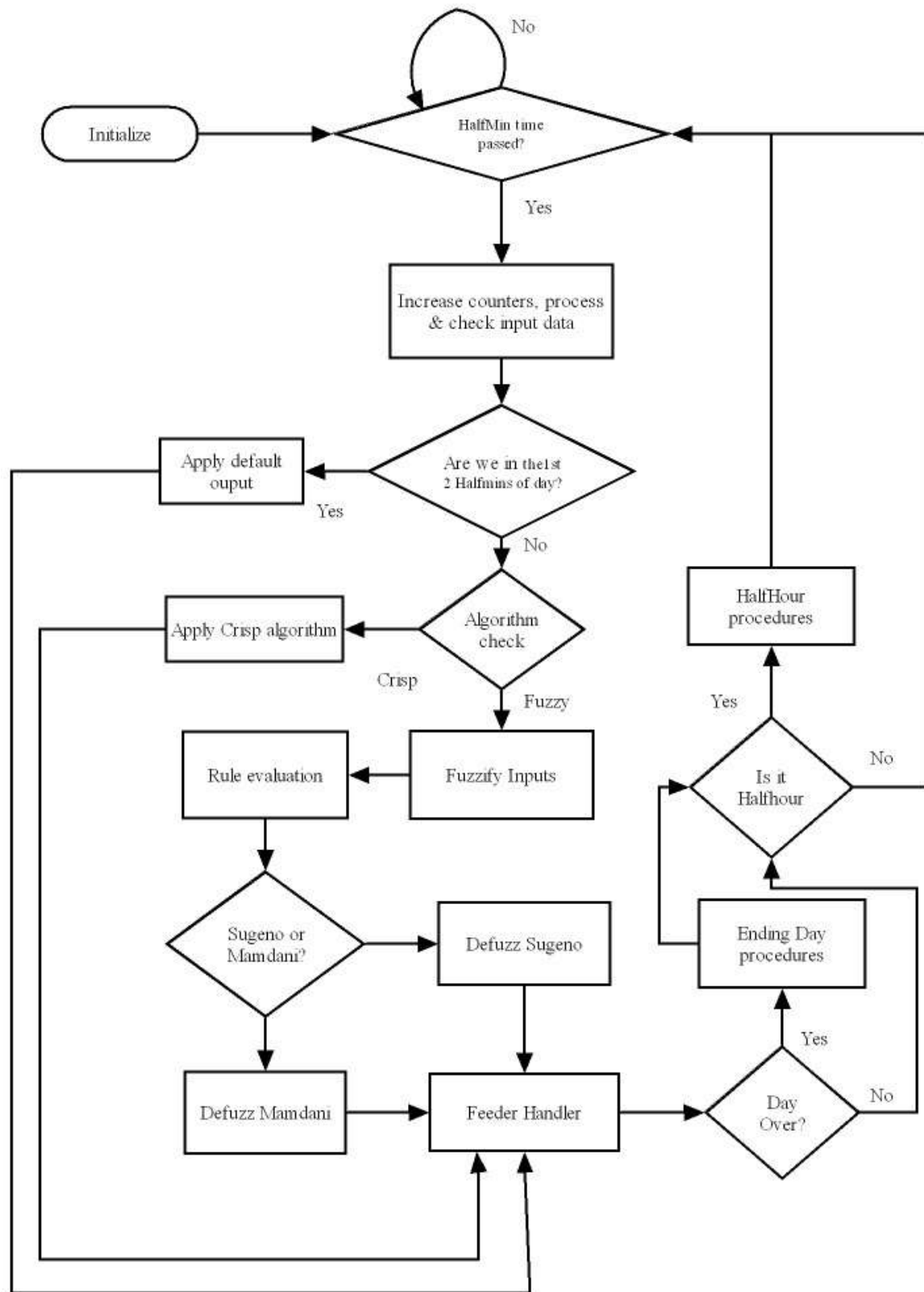
Κατά αρχήν εδώ με τον όρο εξαίρεση εννοείται μια μη λογική συμπεριφορά του μικροεπεξεργαστή που θα μπορούσε να προκαλέσει την ενεργοποίηση ενός μη υλοποιημένου interrupt και όχι ο κανονικός όρος για τον οποίο έχει γίνει λόγος σε παραπάνω κείμενο.. Για να καλυφθεί αυτή η πιθανότητα, για τα μη υλοποιημένα interrupt ορίζεται να γίνεται επαναφορά του συστήματος.

### Εξωτερική Επαναφορά ( Reset)

Ο μικροεπεξεργαστής ξεχωρίζει την εσωτερική και εξωτερική επαναφορά από την ταχύτητα με την οποία στην ακίδα επαναφοράς επανέρχεται το λογικό 1. Με το που αντιληφθεί εξωτερικό interrupt επαναφοράς το σύστημα ξεκινάει την λειτουργία του από την αρχή. Το σύστημα είναι έτσι υλοποιημένο που δεν χάνονται οι βασικές ρυθμίσεις ή τα δεδομένα από την επαναφορά.



6.8 Reset





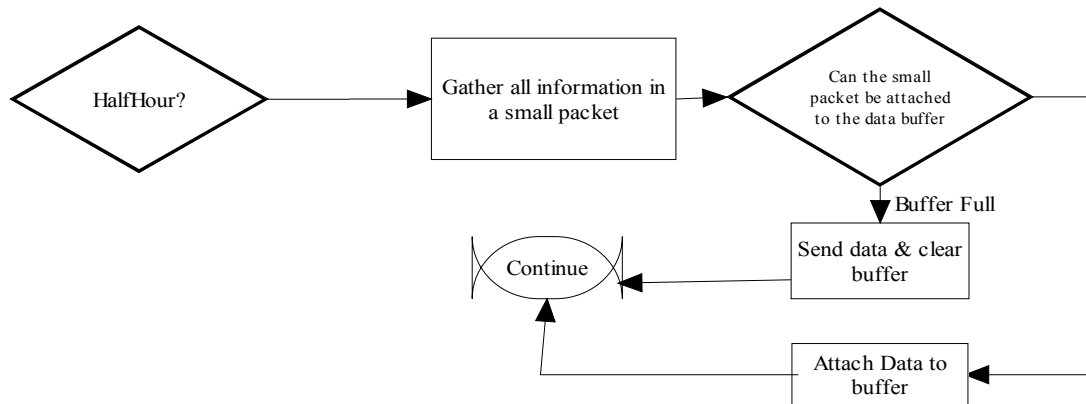
## KYPIA ROYTINA

Η ρουτίνα ξεκινάει με την αρχικοποίηση των ρυθμίσεων του μικροεπεξεργαστή. Εκκινείται ο χρονομετρητής, ενεργοποιούνται τα interrupt (εσωτερικά και εξωτερικά), γίνεται η παραμετροποίηση για την σειριακή επικοινωνία και οι κατάλληλες ρυθμίσεις για την συνεννόηση μικροεπεξεργαστή και κινητού. Διαφαίνεται από το διάγραμμα ότι η αρχικοποίηση δεν θα επαναληφθεί κατά την κανονική ροή της λειτουργίας του συστήματος. Και αυτό γιατί από εκεί και έπειτα μπαίνουμε στην βασική λειτουργία της κύριας ρουτίνας που είναι ο έλεγχος της τροφοδοσίας ανά μονάδα χρόνου(μισόλεπτο).

Για να ξεκινήσει η διαδικασία αξιολόγησης πρέπει να γνωρίζει ο μικροεπεξεργαστής πότε πέρασε ένα μισόλεπτο. Τον χρόνο τον αντιλαμβάνεται με την διαδικασία που περιγράψαμε παραπάνω. Στην συνέχεια ενημερώνονται όλες εκείνες οι μεταβλητές και οι παράμετροι που αποτελούν είσοδο για τους αλγόριθμους και επίσης γίνονται κάποιοι έλεγχοι πάνω σε αυτά τα δεδομένα( πχ. δεν μπορούμε να έχουμε παραπάνω από συγκεκριμένο αριθμό αιτήσεων ανά μισόλεπτο). Ακολουθεί η επεξεργασία των δεδομένων από τον αλγόριθμο που έχει επιλεγθεί κάθε φορά, εκτός από τα πρώτα δύο, χρονικά, μισόλεπτα κάθε ημέρας όπου έχουμε σταθερή έξοδο. Αφού επεξεργαστούν τα δεδομένα προκύπτει η ποσότητα τροφής σε μονάδες που θα δοθεί. Αυτή η ποσότητα πρέπει να μεταφραστεί σε χρόνο που θα είναι ανοιχτό το μηχάνημα τροφοδοσίας. Την εργασία αυτή έχει αναλάβει μια άλλη ρουτίνα, η οποία κρατάει ανοιχτό το μηχάνημα τροφοδοσίας για όσο χρειάζεται. Τι γίνεται όμως όταν χρειαστεί να δώσει τροφή για παραπάνω από ένα μισόλεπτο; Δεν υπάρχει πρόβλημα η διαδικασία τροφοδότησης σταματάει προσωρινά και συνεχίζεται μετά την επεξεργασία των νέων δεδομένων, ασφαλώς όση ποσότητα δεν δόθηκε πριν θα δοθεί τώρα.

Αμέσως έπειτα ελέγχει αν έχουν περάσει 24 ώρες, δηλαδή μια μέρα, και αν αυτό

συμβαίνει στέλνει τα τελικά δεδομένα της ημέρας( συνολική ποσότητα τροφής, αριθμό αιτήσεων) στην βάση και αναπροσαρμόζει αν χρειάζεται την βασική συνολική ποσότητα ημέρας( όπως περιγράφηκε στο κεφάλαιο των αλγόριθμων, Τελευταία πράξη πριν γυρίσει σε κατάσταση αναμονής είναι να ελέγξει αν βρισκόμαστε σε κάποιο μισάωρο ώστε να αποθηκεύσει τα δεδομένα μισάωρου και αν χρειαστεί( δηλαδή έχει γεμίσει ο προσωρινός καταχωρητής) να τα αποστείλει στη βάση.



6.10 Το διάγραμμα ροής για το μισάωρο

Καλό είναι να σημειωθούν επίσης κάποια χαρακτηριστικά που είτε δεν έχουν φανεί ως τώρα είτε δεν έχει τονιστεί η σημασία τους. Ένα είναι η μονιμότητα των δεδομένων.

Τα σημαντικά για την ομαλή λειτουργία του συστήματος δεδομένα τα οποία πρέπει να διατηρηθούν ακόμα και σε απώλεια της ηλεκτρικής τροφοδοσίας δεν υπάρχει κίνδυνος να χαθούν. Αυτά είναι το μισόλεπτο και η μέρα που βρισκόμαστε η ποσότητα τροφής που έχουμε δώσει έως τώρα, ποιος αλγόριθμος λειτουργεί, ο αριθμός του κινητού κ.α. . Κάτι τέτοιο διασφαλίζεται με την εγγραφή τους στην EEPROM λόγω των χαρακτηριστικών που έχουν ήδη αναφερθεί.

Επίσης πρέπει να τονιστεί η ανεξαρτησία και ο γενικότερος ρόλος των διακοπών. Προσφέροντας λειτουργικότητα, εκτός της συνηθισμένης ροής, κάνουν δυνατή την λειτουργία του συστήματος. Για παράδειγμα αν κάποιος δει το διάγραμμα ροής παρατηρεί στην κύρια ρουτίνα μια αναμονή για την πάροδο ενός μισολέπτου, στην πραγματικότητα δεν είναι παρά ο έλεγχος που γίνεται στην μεταβλητή που μεταβάλλει το interrupt μέτρησης του χρόνου(βλ. παραπάνω).

Ασφαλώς για να γίνουν δυνατά όλα αυτά χρειάστηκε η δημιουργία κάποιων βιβλιοθηκών. Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι η χειρισμού της EEPROM, της ασαφούς λογικής και της σειριακής θύρας. Επίσης δημιουργήθηκαν συναρτήσεις για διάφορες ανάγκες του συστήματος(πχ. δημιουργία πακέτου δεδομένων για αποστολή).

#### Χάρτης Μνήμης και Δομές Δεδομένων

Ο κώδικας που αναπτύχθηκε δεν πρέπει να ξεχνάει κανείς ότι είχε συγκεκριμένο προορισμό ένα μικροεπεξεργαστή. Αξίζει λοιπόν να παρουσιαστεί ο τρόπος με τον οποίο χρησιμοποιήθηκε η μνήμη του. Επίσης είναι σωστό να αναφερθούν και οι ιδιαίτερες δομές που αναπτύχθηκαν για τα δεδομένα.

Περιοχή Μνήμης	Χρήση	Περιγραφή
\$0000-\$01FF	Καταχωρητές επεξεργαστή	Καταχωρητές Εισόδου/Εξόδου,ελέγχου κτλ
\$0800-\$09FF	Προσωρινά Δεδομένα(μεταβλ.)	1 Kbyte RAM
\$0A00-\$0BFF	Στοιβα	
\$0D00-\$0FFF	Κρίσιμα Δεδομένα(μεταβλητές)	768 bytes EEPROM
\$8000-\$8059	Συναρτήσεις βασικής βιβλιοθήκης	32 Kbytes Flash EEPROM
\$805A-\$8C73	Προγράμματος	
\$913C-\$91D6	Συναρτήσεις βασικής βιβλιοθήκης	
\$9221-\$9362	Σταθερά Δεδομένα	
\$936D-\$94F0	Συναρτήσεις βασικής βιβλιοθήκης	
\$F780-\$F7FF	Καθρέφτης του πίνακα διακοπών	
\$F800-\$FBFF	Θέση για επέκταση του bootloader	
\$FC00-\$FBFF	EEPROM bootloader	
\$FFC0-\$FFFF	Πίνακας των διακοπών	
		Πίνακας 6.1 χάρτης της μνήμης

Δομές χρησιμοποιήθηκαν στην υλοποίηση της ασαφής λογικής. Δημιουργήθηκαν ξεχωριστές δομές

- i) για την αναπαράσταση της τραπεζοειδούς συνάρτησης συγγένειας μία δομή που περιέχει πληροφορία όσον αφορά την κλίση των μη παράλληλων ευθύγραμμων τμημάτων και σε ποια σημεία τέμνουν τις παράλληλες γραμμές.
- ii) Για την αναπαράσταση της τριγωνικής συνάρτησης συγγένειας μια δομή που περιέχει τρία σημεία τα δύο της βάσης και το ένα της κορυφής
- iii)Ακόμα χρειάστηκε η δημιουργία ξεχωριστής δομής για τους κανόνες ώστε κάθε κανόνας να συγκρατεί τις δύο εισόδους του και την έξοδο του καθώς και απαραίτητα flags(διαχωριστικό εισόδων εξόδων και αν ακολουθεί άλλος κανόνας)
- iv)Δομή για τις εισόδους
- v) Και για τις εξόδους
- vi)Χρειάστηκαν επίσης κάποιες δομές σφαλμάτων πχ. σφαλμάτων εγγραφής στην EEPROM ή σφαλμάτων στην σειριακή

### Χειρισμός του μόντεμ

Όπως έχει ήδη αναφερθεί το μόντεμ δέχεται εντολές μέσω της σειριακής του θύρας. Αυτές είναι τύπου AT (εντολές που ξεκινάνε πάντα έτσι). Οι εντολές που χρησιμοποιήσαμε εμείς είναι:

AT+cmgr=<n>: ανάγνωση του <n> μηνύματος

AT+cmgd=<n>: διαγραφή<n> μηνύματος

AT+cmgs=<t><text>: αποστολή γραπτού μηνύματος

AT+cfun=1: software reset

## 7. ΕΠΙΛΟΓΟΣ

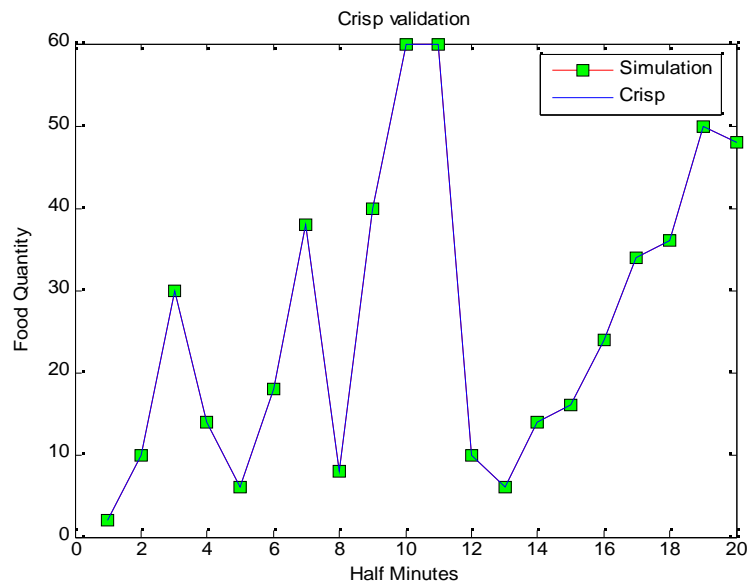
### 7.α Αξιολόγηση συστήματος και αποτελέσματα

Το σύστημα έχει σχεδιαστεί και υλοποιηθεί, πρέπει όμως να αξιολογηθεί και να δοκιμαστεί η σωστή λειτουργία του. Για την επαλήθευση της σωστής λειτουργίας του όσον αφορά τους αλγορίθμους βοήθησε το μοντέλο που δημιουργήθηκε στο Matlab. Το μοντέλο “έτρεξε” όλους τους αλγορίθμους με συγκεκριμένη είσοδο, στην συνέχεια δόθηκε η ίδια αλληλουχία στο σύστημα και καταγράφηκε η έξοδος του. Οι δύο έξοδοι τοποθετήθηκαν μετά στο ίδιο γράφημα για να δούμε αν έχουμε σωστή απόκριση, παρακάτω φαίνεται η διαδικασία.

Για τον crisp αλγόριθμο βλέπει κανείς ότι έχουμε ίδια αποτελέσματα σε προσομοίωση και πραγματικότητα. Βλέπουμε το αποτέλεσμα στην γραφική αναπαράσταση αμέσως μετά.

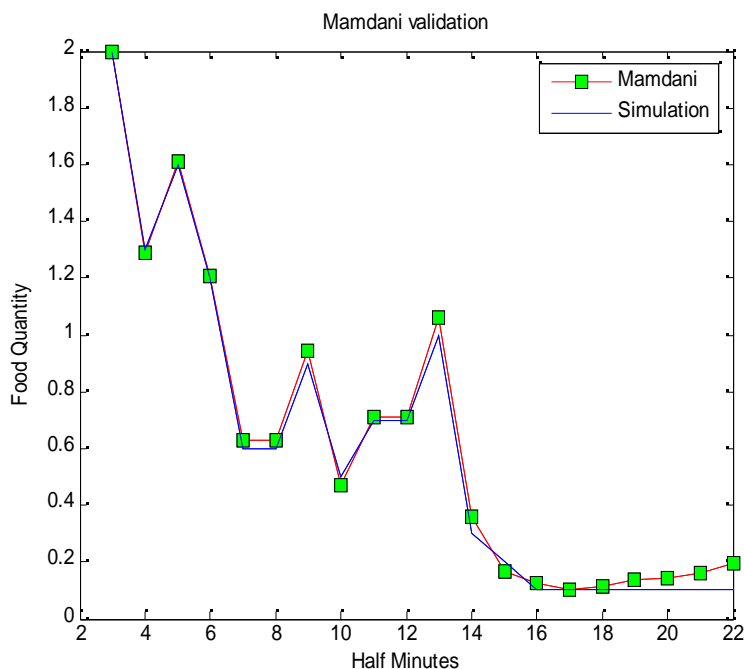
Πίνακας 7.α.1 οι δοκιμαστικές είσοδοι

1	5	15	7	3	9	19	4	20	30	30	5	3	7	8	12	17	18	25	24
---	---	----	---	---	---	----	---	----	----	----	---	---	---	---	----	----	----	----	----



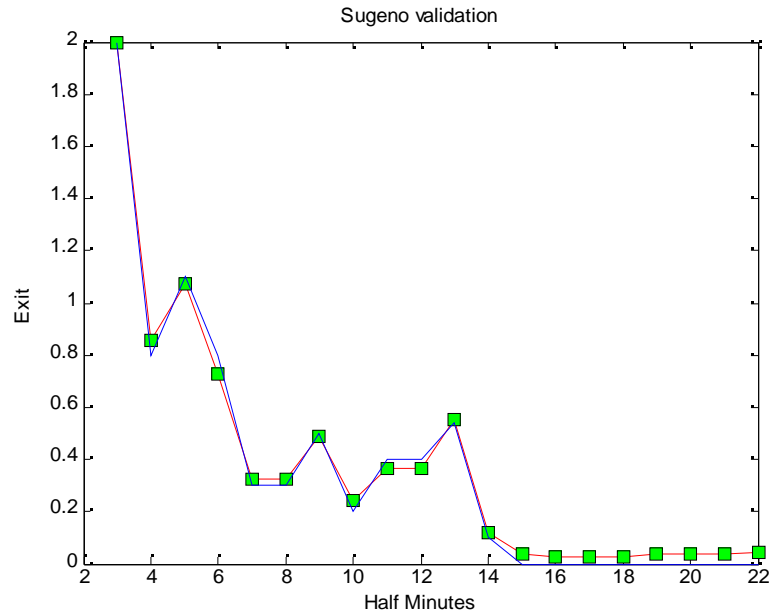
#### 7.α.1 Ταύτιση στον crisp αλγόριθμο.

Στην συνέχεια βλέπουμε τις εξόδους για το Mamdani σύστημα, για τις ίδιες εισόδους και πάλι με τον ίδιο χρωματικό κώδικα. Παρατηρούμε ότι δεν υπάρχει πλήρης ταύτιση, αυτό οφείλεται στο ότι δεν έχουμε ακρίβεια πάνω από ένα δεκαδικά ψηφία στον μικροεπεξεργαστή.



#### 7.α.2 Επαλήθευση για το Mamdani

Ακολουθεί το Sugeno σύστημα. Εδώ επίσης μπορούμε να κάνουμε την ίδια παρατήρηση που είχαμε κάνει για το Mamdani ότι δηλαδή λόγω έλλειψης δεύτερου δεκαδικού σημείου ακρίβειας έχουμε μια μικρή παρέκκλιση.



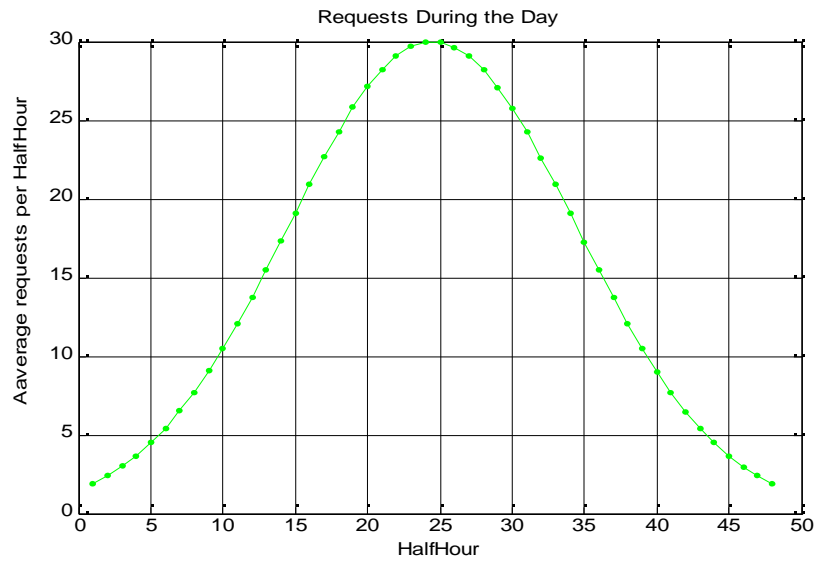
7.α.3 Η επαλήθευση του Sugeno συστήματος

Έχουν γίνει πολλές περισσότερες μετρήσεις για να αποδειχθεί η σωστή λειτουργία του συστήματος. Από αυτές τις μετρήσεις βγήκε το συμπέρασμα ότι το σύστημα λειτουργεί σωστά και σύμφωνα με τις θεωρητικές προβλέψεις. Ακολουθούν κάποια απλά μοντέλα συμπεριφορών.

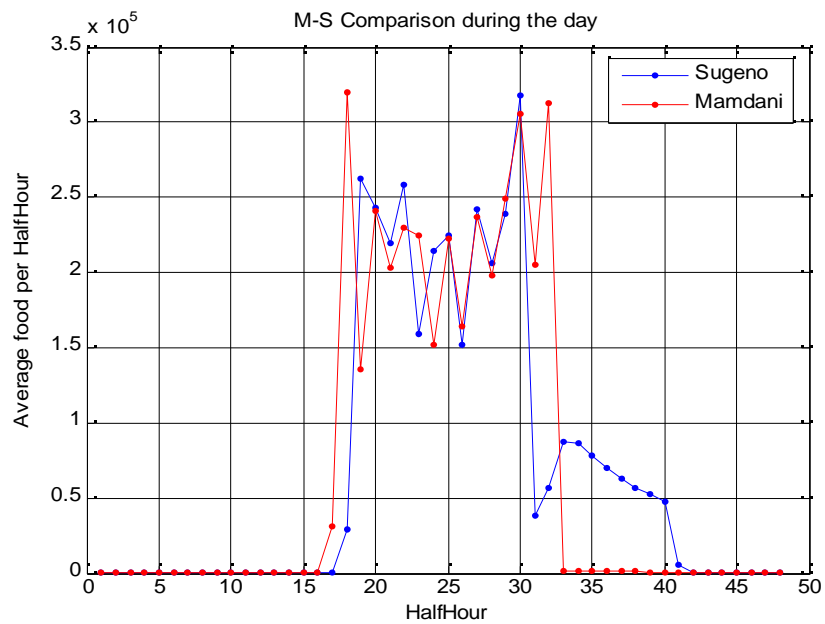
#### 7.α.1 Μοντέλο με κατανομή Gauss

Παρατηρούμε ότι με την κατανομή αυτή το σύστημα αποκρίνεται όσον αφορά τον crisp αλγόριθμο με πολύ ομαλές αυξήσεις. Είναι αναμενόμενο αν σκεφτεί κανείς ότι ο crisp λειτουργεί σαν ένας γραμμικός ενισχυτής. Από την άλλη πλευρά οι Α.Λ. αλγόριθμοι αποκρίνονται στην συνεχή πολύ υψηλή δραστηριότητα με αύξηση της ποσότητας τροφής.

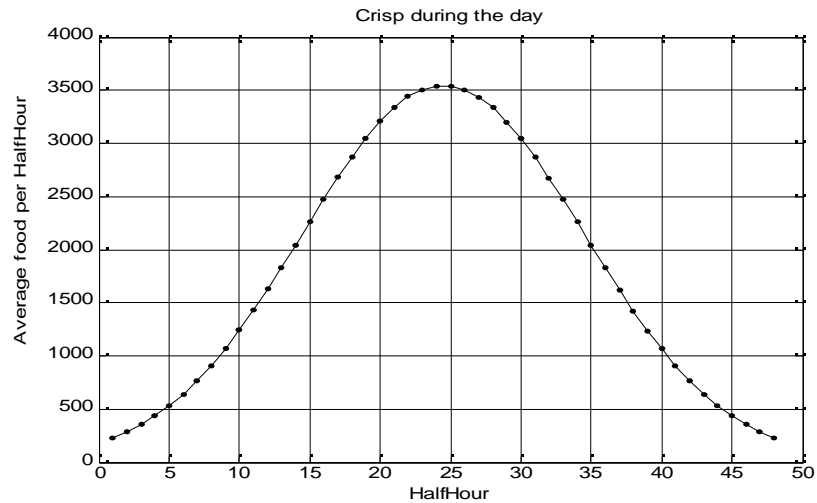




7.α.1 Οι αιτήσεις εκφρασμένες κατά μέσο όρο ανα μισάωρο



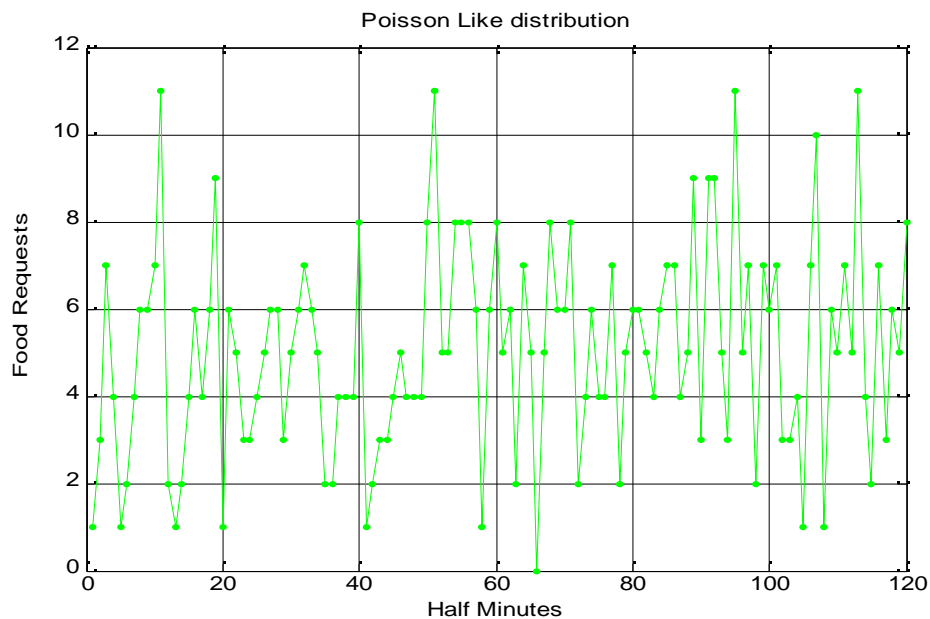
7.α.1.2 Οι δύο αλγόριθμοι Α.Λ. βλέπουμε ότι πιάνουν πολύ υψηλές τιμές



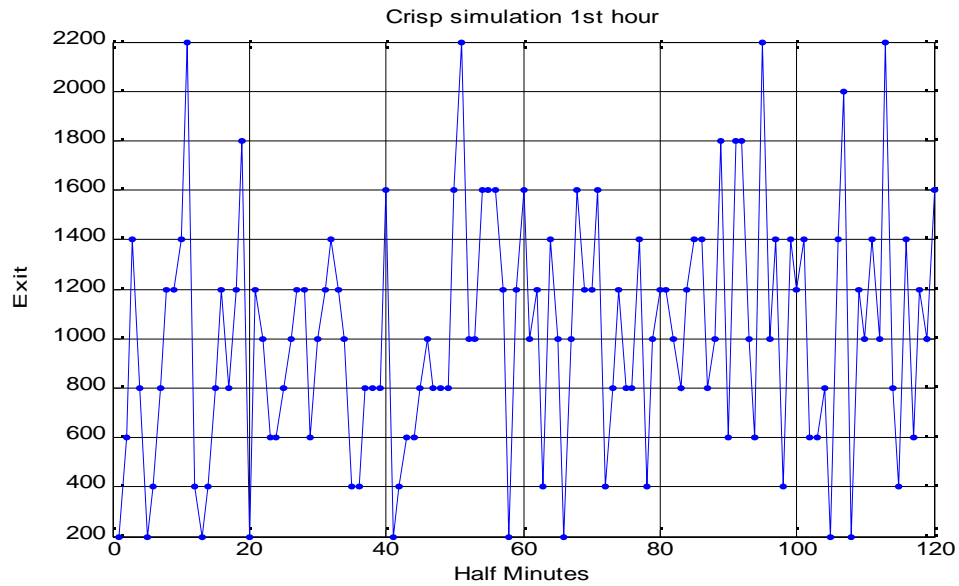
7.α.1.3 Βλέπουμε ότι εδώ ο crisp ανταποκρίνεται πολύ καλύτερα, η ερώτηση όμως είναι αν αυτό είναι ένα ρεαλιστικό μοντέλο.

### 7.α.2 Τυχαία κατανομή

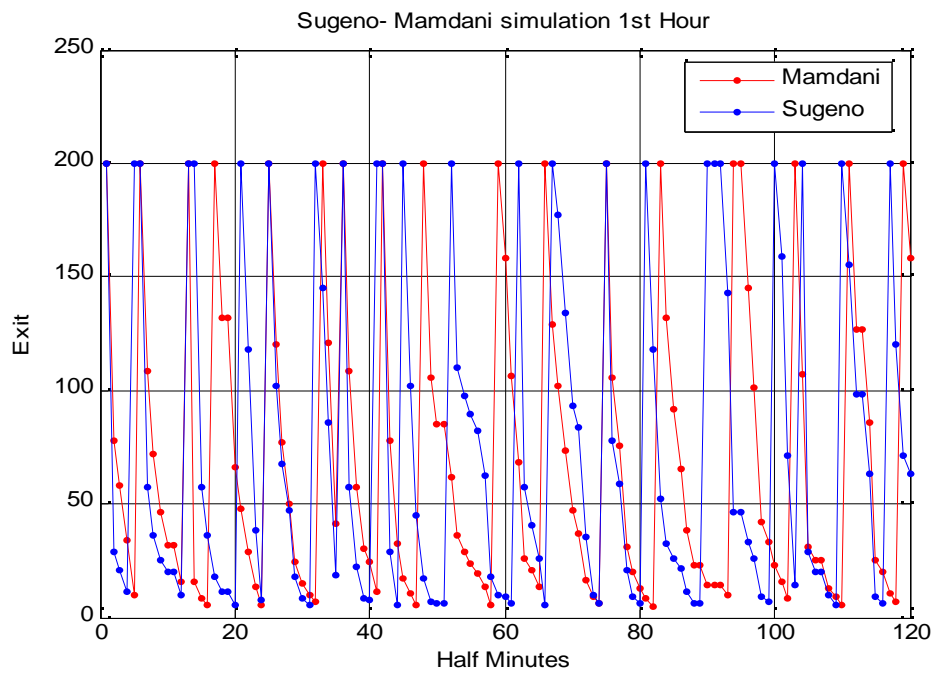
Η κατανομή αυτή δημιουργήθηκε στηριζόμενη πάνω στην κατανομή Poisson. Θεωρείται ότι για κάποιες ώρες της ημέρας τα ψάρια ακολουθούν μια Poisson με  $\lambda=5$  (ώρες μη αιχμής) και κάποιες άλλες με  $\lambda=15$  (ώρες αιχμής). Ας δούμε τώρα την συμπεριφορά των τριών αλγορίθμων.



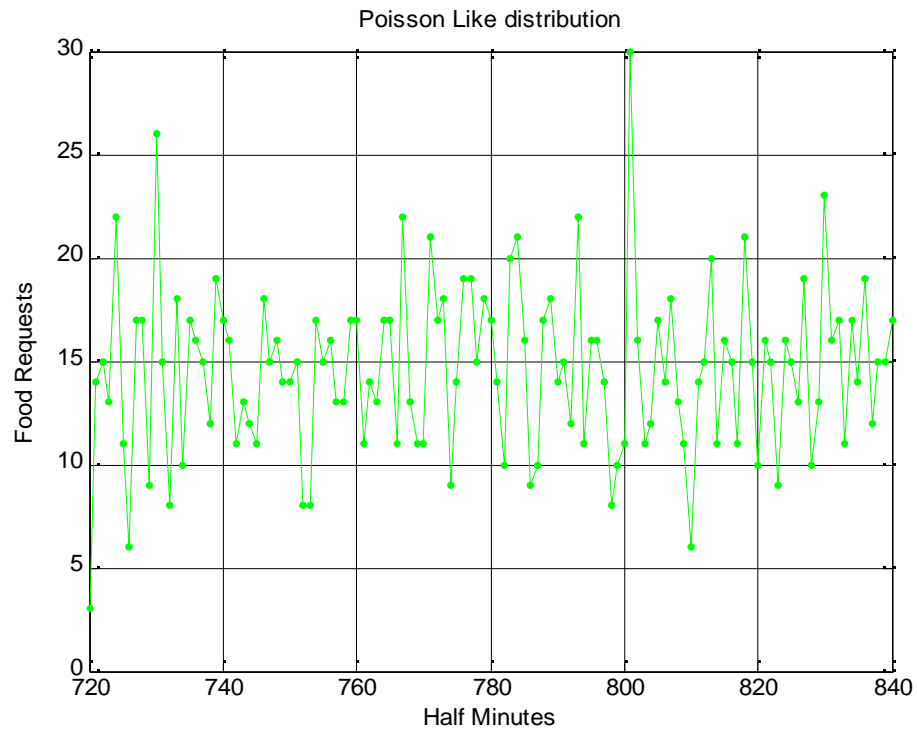
7.α.2.1 Τα requests σε ώρα μη αιχμής



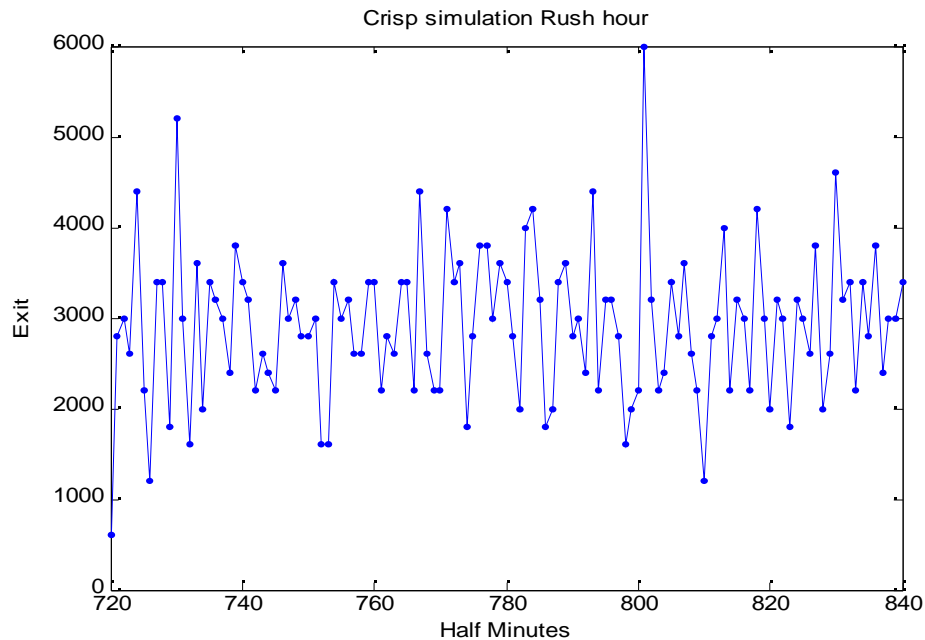
7.α.2.1 Η συμπεριφορά του Crisp κατά την πρώτη ώρα



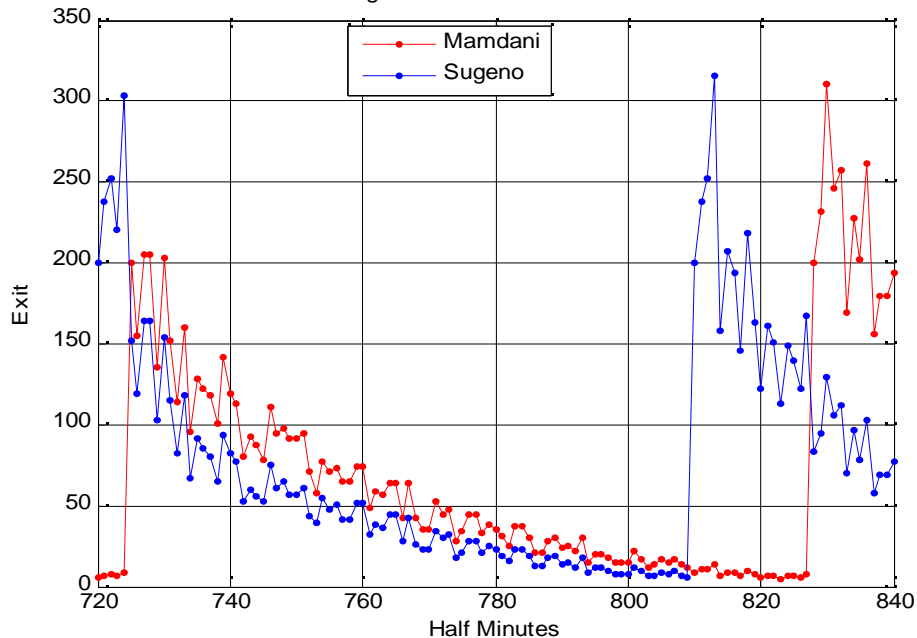
7.α.2.3 Mamdani-Sugeno σε ώρα μη αιχμής βλέπουμε ότι κινούνται στα όρια της βασική μονάδας τροφής Qp.



7.α.2.3 Οι αιτήσεις σε ώρα αιχμής

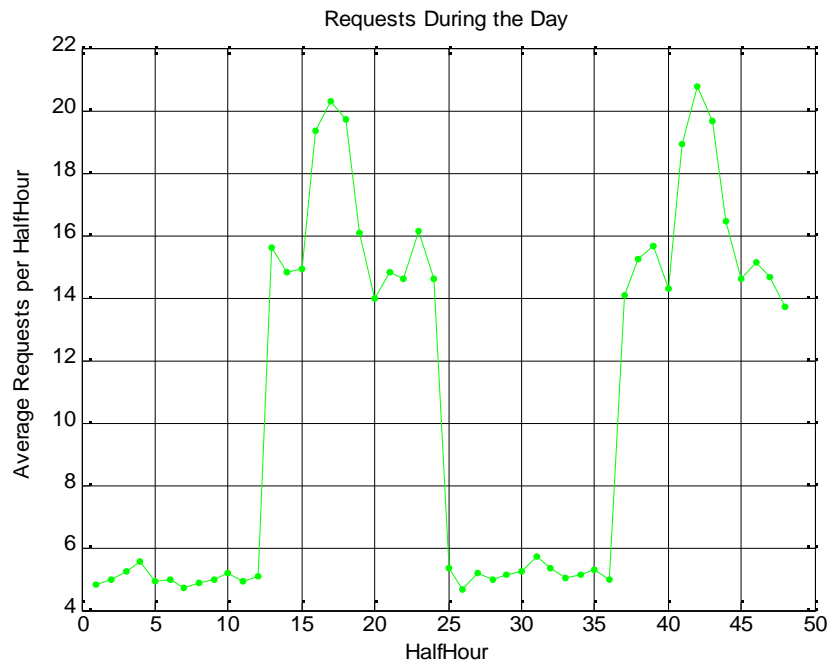


7.α.2.4 Ο Crisp την ώρα αιχμής  
Sugeno simulation Rush Hour

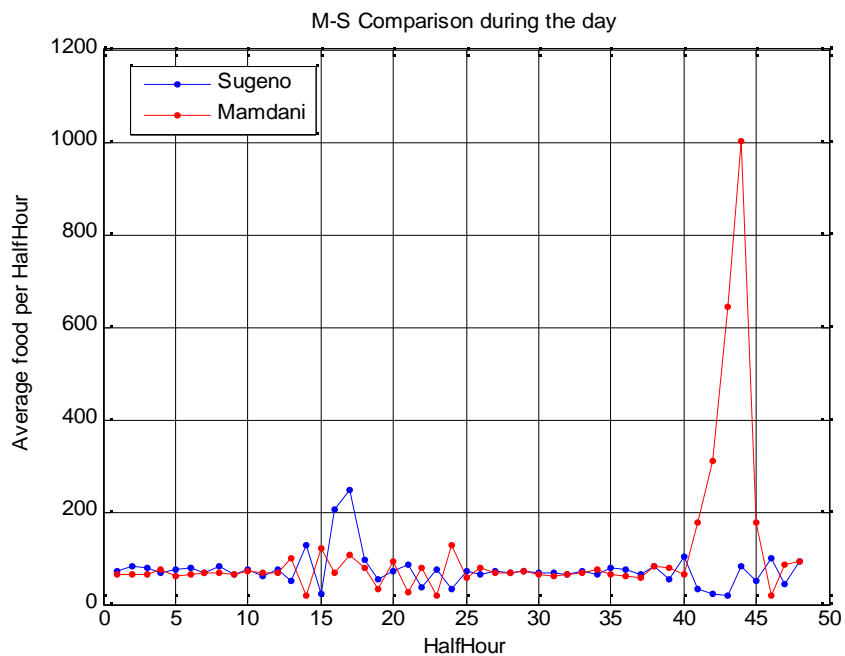


7.α.2.5 Σύγκριση Mamdani και Sugeno, ο πρώτος βλέπουμε ότι έχει μια πιο ήπια προσαρμογή

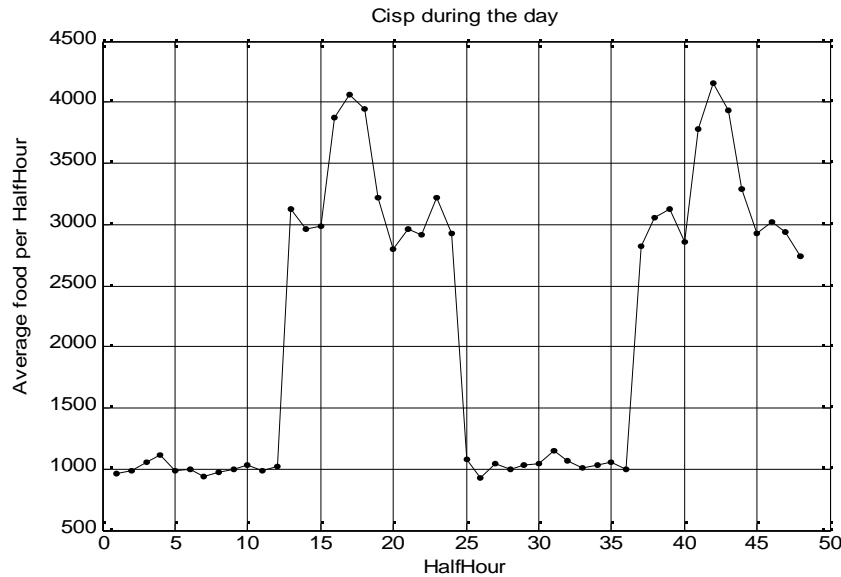
Ας παρατηρήσουμε τώρα και συνολικά την συμπεριφορά όλων των αλγορίθμων κατά τη διάρκεια όλης της ημέρας. Όπως θα δούμε ο crisp έχει πολύ μεγάλη έξοδο κινούμενη μακριά από την βασική μονάδα τροφής, σε αντίθεση με τους άλλους δύο.



7.α.2.6 Οι αιτήσεις κατά την διάρκεια της ημέρας (ο μέσος όρος ανά μισάωρο)



7.α.2.7 Οι δύο αλγόριθμοι Α.Λ. σε 48 μισάωρα (μέσος όρος)

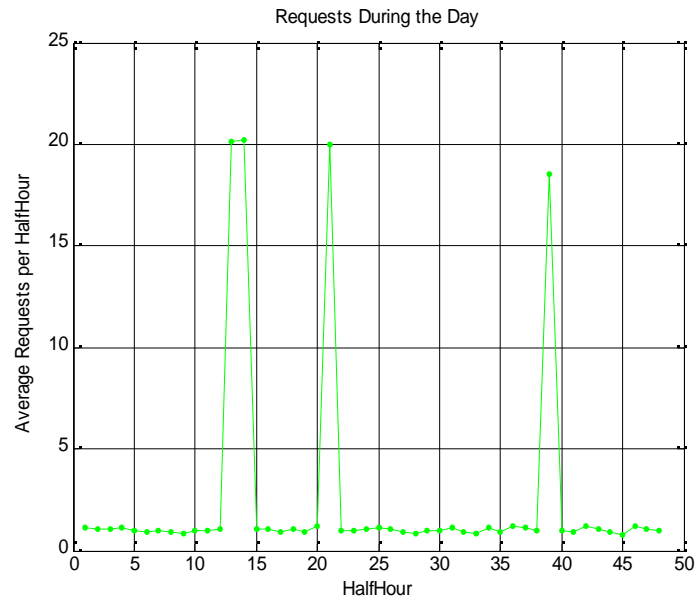


7.α.2.8 Ο crisp καθ' όλη την ημέρα

Από τα δύο αυτά σενάρια προέκυψαν μερικά χρήσιμα συμπεράσματα. Παρατηρήθηκε ότι οι Α.Λ. αλγόριθμοι σε καταστάσεις όπου έχουμε μέγιστη δραστηριότητα για αρκετή ώρα (πχ.15 λεπτά) θα συναντήσουν πρόβλημα(υπερβολική αύξηση στην έξοδο) ενώ ο crisp όχι. Αντίθετα αποκρίνονται πολύ καλά σε καταστάσεις με ποικιλία εισόδων χωρίς να έχουν υπερβολικές αντιδράσεις.

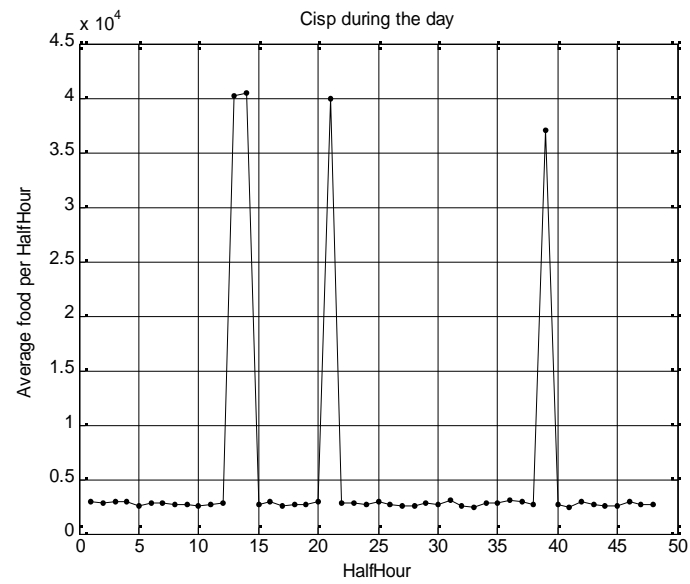
### 7.α.3 Ένα πιο ρεαλιστικό σενάριο

Τα ψάρια τρέφονται κυρίως τις ώρες 06:00-07:00, 10:00-10:30, 18:30-19:00 όπου έχουμε μεγάλη ζήτηση. Τις υπόλοιπες ώρες υπάρχει μικρή κινητικότητα (ελάχιστα χτυπήματα). Ακόμα η επιθυμητή ποσότητα που επιθυμεί ο ιχθυοκαλλιεργητής να δώσει είναι περίπου 60 Kg. Το Qr λοιπόν είναι 20gr ή όπως φαίνεται στα διαγράμματα 2000 μονάδες. Για αρχή ας παρατηρήσουμε τις αιτήσεις κατά τη διάρκεια της ημέρας.



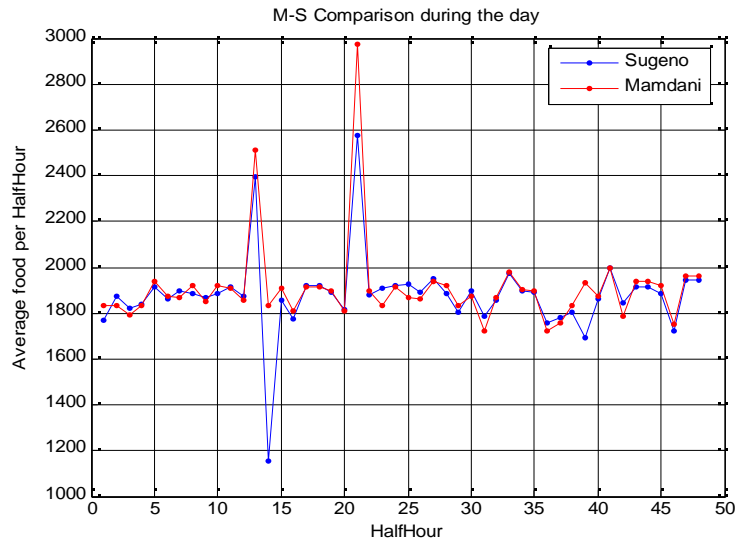
#### 7.α.3.1 Μέσος όρος των αιτήσεων ανά μισάωρα

Και ακολουθούν τα διαγράμματα των τριών αλγορίθμων



#### 7.α.3.2 Ο Crisp αλγόριθμος





7.α.3.3 Mamdani-Sugeno στο ρεαλιστικό σενάριο

Ποσότητα που δόθηκε κατά την διάρκεια εφαρμογής του crisp 167.060gr, Sugeno 54.140gr και Mamdani 55.018gr. Ο crisp κινήθηκε εντελώς εκτός ορίων με μια απόκλιση της τάξης του 280%. Όμως και οι δύο ασαφείς αλγόριθμοι συναντούν προβλήματα. Από το 37 έως το 38 μισάωρο έχουμε αυξημένη ζήτηση αλλά αποτυγχάνουν να ακολουθήσουν. Αυτό συμβαίνει γιατί τυχαίνει η τελευταία έξοδος πριν από την έξαρση έτυχε να είναι μικρή αλλά όχι μικρότερη του κάτω ορίου που έχουμε θέσει έτσι ενώ ακολουθούν αυξητική πορεία κατά μέσο όρο δίνουν λιγότερη τροφή.

## 7.β Περαιτέρω Εργασία

Το σύστημα μπορεί όμως να επεκταθεί και να γίνει πολύ καλύτερο και ακόμα πιο λειτουργικό και χρήσιμο.

Κατά αρχήν πρέπει να δημιουργηθεί μια εφαρμογή στον σταθμό βάσης, φιλική προς τον χρήστη (γραφικό περιβάλλον), για την αξιοποίηση των δεδομένων και καταγραφή τους. Η εφαρμογή θα χειρίζεται μια βάση στην οποία θα ταξινομεί τα στοιχεία που στέλνονται μέσω του GSM/GPRS μόντεμ. Στην συνέχεια θα μπορεί να προσφέρει διάφορα στατιστικά για ανάλυση. Επίσης η εφαρμογή θα προσφέρει ένα εύχρηστο

περιβάλλον επικοινωνίας με το σύστημα. Προχωρημένη υπηρεσία σε συνδυασμό με κατάλληλο προγραμματισμό στον επεξεργαστή θα είναι ο επαναπρογραμματισμός του μικροεπεξεργαστή μέσω του μόντεμ.

Ένας άλλος τομέας στον οποίο θα μπορούσε να γίνει μια σημαντική βελτίωση είναι αξιοποιώντας τα πρώτα δεδομένα να δημιουργηθεί ένας neuro-fuzzy αλγόριθμος για τον χειρισμό των μηχανημάτων τροφοδοσίας αντί των ασαφών που χρησιμοποιούνται τώρα. Ένας neuro-fuzzy αλγόριθμος έχει την ικανότητα να εκπαιδεύεται από τα υπάρχοντα δεδομένα και έτσι να είναι δυναμικός.

Ένα σύστημα μπορεί να εξυπηρετήσει μέχρι κάποιο συγκεκριμένο αριθμό ιχθυοκλωβών (2-3). Και αυτός ο αριθμός μπορεί να μικρύνει κι άλλο από την τοπολογία των κλωβών. Οπότε μπορεί σε μια εγκατάσταση να δημιουργηθεί η ανάγκη για περισσότερα του ενός συστήματα. Θα ήταν λοιπόν επιθυμητό να υπάρχει επικοινωνία μεταξύ των συστημάτων, πχ. για λόγους οικονομίας να μην χρησιμοποιούνται παραπάνω από ένα μόντεμ. Θα μπορούσε λοιπόν να υλοποιηθεί ένα δίκτυο ομοτίμων (Peer2Peer) για αυτό τον σκοπό.

Επιπλέον το σύστημα τροφοδοτείται από μια μπαταρία 12 Volt. Ο τρόπος αυτός όμως αποτελεί τροχοπέδη στην αυτονομία του, ένας ηλιακός συλλέκτης ή μια ανεμογεννήτρια θα ήταν μια πολύ καλή λύση όσον αφορά το θέμα της ενέργειας.

Στην αυτονομία επίσης θα προσέφερε ίσως στην περίπτωση πολλών ιχθυοκλωβών η δημιουργία συστημάτων αποκλειστικά για συλλογή πληροφοριών, λήψης αποφάσεων και χειρισμού της τροφοδοσίας. Επίσης σε μια τόσο μεγάλη κλίμακα όπου θα έχουμε μεγαλύτερο όγκο δεδομένων η επικοινωνία θα ήταν καλύτερο να γινόταν μέσω GPRS.

## BIBΛΙΟΓΡΑΦΙΑ

1. M. Paspatis “Demand feeding systems in aquaculture. A profitable practice for warm water marine species rearing” 3<sup>rd</sup> MAST Conference 1998
2. Papandroulakis N., Papaioannou D., Divanach P. “An automated feeding system for intensive hatceries” Aquacultural Engineering, 26 (1): 13-26 2001
3. Papandroulakis N., Divanach P., Paspatis M., Papaioannou D., Spyropoulou E., Gdisis S., Veneris I. 1999. “Automated management system in vertical integrated farms of intensively reared euryaline fish” RCID 555. Final report 129 p.
4. Paul Rozin and Jean Mayer ”Regulation of food intake in the goldfish” Am J Physiol 201: 968-974, 1961 june 1961
5. Perry R. Cook “Human-Computer Interface Technology course notes” Associate Professor Princeton University
6. Seattle Sobotics Society (<http://www.seattlerobotics.org/>)
7. 68HC12 yahoo forum
8. Jonathan W. Valvano Developing Embedded Software in C Using ICC11 /ICC12 / Metrowerks
9. Automotive Control and Electronics Laboratory HC12 Hardware Abstraction Layer ([www.acelab.com.org/HAL/index.html](http://www.acelab.com.org/HAL/index.html))
10. Motorola Freescale “M68HC12B32 EVB Manual” October 1999
11. Motorola Freescale “CPU12 Reference Manual” 2002
12. Motorola Freescale “M68HC12B Microcontrollers Family Datasheet” June 2004
13. Motorola Freescale “HC12 Compiler” August 2003
14. Motorola Freescale “Development Using Metrowerk's Code warrior” AN 2216/D Rev 0/, December 2001
15. Motorola Freescale “Flash Programming fo Motorola M68HC912 Microcontrollers” AN 1836/D 2000
16. Motorola Freescale official 68HC12 Forum
17. Metrowerks Official Forum
18. Matlab Help “Tutorial for Fuzzy Logic Toolbox”
19. Kevin Self "Designing With Fuzzy Logic" , IEEE SPECTRUM, November 1990, 42:44,105
20. Miller B. “Fuzzy logic does real time on the DSP” October 2004 from embedded.com

21. Sellis T. "Artificial Intelligence course notes (chap4 fuzzy logic) " Professor National Technical University of Athens 2003
22. Yixin Chen, James Z. Wang, "Support Vector Learning for Fuzzy Rule-Based Classification Systems" IEEE Transactions on Fuzzy Systems, vol. 11, no. 6, pp. 716-728, 2003. December 2003
23. Lofti Zadeh "Fuzzy sets, Inf. Control" 8, 338-353, 1965
24. Wavecom "AT commands interface Guide for Atx41" September 2004
25. Wavecom "Integra modem M2106 Product Specification" August 2003
26. D. Kolokotsa, D. Tsiavos, G. S. Stavrakakis, K. Kalaitzakis, E. Antonidakis "Advanced fuzzy logic controllers design and evaluation for buildings' occupants thermal-visual comfort and indoor air quality satisfaction" Energy and Buildings, Vol. 33, pp. 531-543, 2001.