



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Πειραιά
 Σχολή Τεχνολογικών Εφαρμογών
 Τμήμα Ηλεκτρονικών Υπολογιστικών Συστημάτων

Πτυχιακή Εργασία

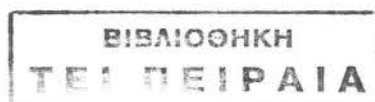
**Αναπτυξη μοντέλου Τεχνητών Νευρωνικών Δικτύων
 για την μελέτη πολύπλοκων φυσικών ή βιολογικών
 διεργασιών**

**Αναξαγόρας Φωτόπουλος
 Α.Μ.: 37331**

Επιβλέπων Καθηγητής

**Δρ. Παναγιώτης Ηρ. Γιαννακόπουλος
 Καθηγητής ΤΕΙ Πειραιά**

Φεβρουάριος 2013





**Technological Education Institute of Piraeus
Faculty of Engineering
Department of Electronic Computer Systems**

Thesis

**Development of a Neural Network Model for the
study of complex natural or biological processes**

**Anaxagoras Fotopoulos
Reg. Number: 37331**

Advisor Professor

**Dr. Panagiotis H. Yannakopoulos
Professor TEI of Piraeus**

February 201

*Στην μνήμη του πατέρα μου,
στον αγώνα της μητέρας μου,
στο φίλο μου Κώστα Φιλιππάκη*

*Ευχαριστώ όλους τους καθηγητές
που συνεργάστηκα στενά
και με βοήθησαν να
αναπτυχθώ ακαδημαϊκά και ερευνητικά.*

Σας θεωρώ Διδάσκαλους, Συνεργάτες, Φίλους

*Σας χρωστάω πολλά,
θα σας ανταποδώσω περισσότερα.*

Α.Φ.

Αναξαγόρας Φωτόπουλος

.....

Copyright 2013[©]

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Πειραιά

Με επιφύλαξη παντός νομίμου δικαιώματος.
All rights reserved

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τους συγγραφείς.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Πειραιά .

Περίληψη

Πρόσφατες εργασίες έδειξαν υψηλό βαθμό συσχέτισης μεταξύ φυσικών και βιολογικών διεργασιών κατά την κρίσιμη περίοδο πριν την εμφάνιση ενός καταστροφικού φαινομένου (Eftaxias, et al., 2006). Η ανάγκη για περαιτέρω διερεύνηση μιας γενικευμένης προσέγγισης οδήγησε στο πλαίσιο της παρούσας πτυχιακής την μελέτη και την ανάπτυξη νέων μεθόδων πολυπλοκότητας για την μελέτη φυσικών και βιολογικών φαινομένων, καθώς και προγραμμάτων διεπαφής χρήστη για την μελέτη χρονοσειρών των ανωτέρω συστημάτων. Ιδιαίτερη βαρύτητα δίδεται σε μεθόδους εντροπίας και Φράκταλ ανάλυσης για τα φυσικά φαινόμενα και στην ανάπτυξη τεχνητών νευρωνικών δικτύων για τα βιολογικά. Επίσης παρατίθεται αρχικά μια βιβλιογραφική ανασκόπηση στην εξέλιξη των νευρωνικών δικτύων.

Λέξεις – κλειδιά

Νευρωνικά δίκτυα, πρόβλεψη πρωτεϊνικής δομής, ασαφής εντροπία, προγραμματισμός GUI

Abstract

Recent work has shown that a high degree of correlation exists between physical and biological processes for the critical period before catastrophic phenomena occur (Eftaxias, et al., 2006) . The need for further investigation of the generalized approach resulted this thesis to the development of new methods for the investigation of complex physical and biological phenomena, along with GUI programs for the time series representation of such systems. Particular attention is given to entropy methods, fractal analysis for the physical phenomena and the development of neural networks models for the biological ones. Moreover a review on the historical evolution of neural networks and their latest trends is given.

Keywords

Neural Networks, Protein structure prediction, Fuzzy Entropy, GUI programming

Περιεχόμενα

1	ΕΙΣΑΓΩΓΗ ΣΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 10 -
1.1	ΒΙΟΛΟΓΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 10 -
1.1.1	ΕΙΣΑΓΩΓΗ	- 10 -
1.1.2	ΟΙ ΝΕΥΡΩΝΕΣ ΤΟΥ ΑΝΘΡΩΠΙΝΟΥ ΕΓΚΕΦΑΛΟΥ	- 11 -
1.1.3	ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΝΕΥΡΩΝΑ	- 13 -
1.1.4	ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ	- 15 -
1.1.5	ΝΕΥΡΟΒΙΟΛΟΓΙΚΗ ΚΑΙ ΝΕΥΡΟΧΗΜΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΠΛΗΡΟΦΟΡΙΩΝ	- 15 -
1.1.6	ΝΕΥΡΟΔΙΑΒΙΒΑΣΤΕΣ	- 15 -
1.1.7	ΥΠΟΔΟΧΕΙΣ ΝΕΥΡΟΔΙΑΒΙΒΑΣΤΩΝ	- 16 -
1.1.8	ΕΝΔΟΝΕΥΡΩΝΙΚΗ ΕΠΙΚΟΙΝΩΝΙΑ	- 17 -
1.1.9	ΔΙΑΝΕΥΡΩΝΙΚΗ ΕΠΙΚΟΙΝΩΝΙΑ	- 20 -
1.1.10	ΒΙΟΛΟΓΙΚΗ ΝΕΥΡΩΝΙΚΗ ΜΝΗΜΗ ΚΑΙ ΜΑΘΗΣΗ	- 23 -
1.1.11	ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΦΥΣΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 23 -
1.2	ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 24 -
1.2.1	ΕΙΣΑΓΩΓΗ	- 24 -
1.2.2	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 26 -
1.2.3	ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 28 -
1.2.4	Η ΙΚΑΝΟΤΗΤΑ ΜΝΗΜΗΣ	- 29 -
1.2.5	Η ΜΕΤΑΔΟΣΗ ΣΗΜΑΤΟΣ ΣΤΟ ΝΕΥΡΩΝΙΚΟ ΔΙΚΤΥΟ – ΣΥΝΑΡΤΗΣΗ ΚΑΤΩΦΛΙΟΥ HEAVYSIDE	- 29 -
1.2.6	ΠΡΟΤΥΠΑ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΕΚΠΑΙΔΕΥΣΗΣ ΤΟΥΣ	- 30 -
1.2.7	ΤΟ ΠΡΩΙΜΟ ΜΟΝΤΕΛΟ ΤΩΝ McCULLOCH ΚΑΙ PITS	- 30 -
1.2.8	Ο ΚΑΝΟΝΑΣ ΜΑΘΗΣΗΣ ΤΟΥ HEBB	- 32 -
1.2.9	ΤΟ ΠΡΟΤΥΠΟ ΤΟΥ ΑΙΣΘΗΤΗΡΑ (PERCEPTRON)	- 33 -
1.2.10	ΠΕΡΙΓΡΑΦΗ ΕΚΠΑΙΔΕΥΣΗΣ ΕΝΟΣ ΤΕΧΝΗΤΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ	- 35 -
1.2.11	Η ΕΚΠΑΙΔΕΥΣΗ ΤΟΥ ΑΙΣΘΗΤΗΡΑ	- 36 -
1.2.12	ΤΟ ΠΡΟΤΥΠΟ ΤΗΣ ΟΠΙΣΘΟΔΙΑΔΟΣΗΣ ΤΟΥ ΛΑΘΟΥΣ (ERROR-BACKPROPAGATION)	- 37 -
1.2.13	ΤΟ ΠΡΟΤΥΠΟ HORFIELD	- 38 -
1.2.14	ΧΡΗΣΙΜΟΤΗΤΕΣ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΜΟΝΤΕΛΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 39 -
1.3	ΑΝΕΠΤΥΓΜΕΝΑ ΠΡΟΤΥΠΑ ΤΕΧΝ. ΝΕΥΡ. ΔΙΚΤ.: ΧΑΟΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 40 -
1.3.1	ΕΙΣΑΓΩΓΗ ΣΤΑ ΧΑΟΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 40 -
1.3.2	Η ΘΕΩΡΙΑ ΤΟΥ ΧΑΟΥΣ	- 42 -
1.3.3	Η ΘΕΩΡΙΑ ΤΟΥ ΧΑΟΥΣ ΣΤΟΝ ΑΝΘΡΩΠΙΝΟ ΕΓΚΕΦΑΛΟ	- 44 -
1.3.4	ΤΟ ΧΑΟΣ ΣΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 45 -
1.3.5	ΤΟ ΥΠΑΡΕΙΑΚΟ ΑΙΝΙΓΜΑ ΤΟΥ ΧΑΟΥΣ ΣΤΑ ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 46 -
1.3.6	ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΧΑΟΤΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 47 -
1.3.7	ΧΡΗΣΙΜΟΤΗΤΑ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΤΑ ΜΟΝΤΕΛΑ ΧΑΟΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ	- 48 -
1.4	ΑΝΕΠΤΥΓΜΕΝΑ ΠΡΟΤΥΠΑ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ: ΚΒΑΝΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 49 -
1.4.1	ΕΙΣΑΓΩΓΗ ΣΤΑ ΚΒΑΝΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 49 -

1.4.2	Η ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ ΟΔΗΓΕΙ ΣΤΑ ΚΒΑΝΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 52 -
1.4.3	ΠΟΛΛΑΠΛΟΤΗΤΑ ΚΑΙ ΚΒΑΝΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 53 -
1.4.4	ΚΒΑΝΤΙΚΗ ΣΥΝΕΙΡΜΙΚΗ ΜΝΗΜΗ ΚΑΙ ΕΦΑΡΜΟΓΗ ΚΒΑΝΤΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 56 -
1.4.5	Ο ΥΠΕΡΚΕΡΑΣΜΟΣ ΤΩΝ ΚΒΑΝΤΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 58 -
1.4.6	Ο ΜΙΜΗΤΙΣΜΟΣ ΤΩΝ ΚΒΑΝΤΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 59 -
1.5	ΕΦΑΡΜΟΓΕΣ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 61 -
1.6	Ο ΔΙΕΠΙΣΤΗΜΟΝΙΚΟΣ ΧΑΡΑΚΤΗΡΑΣ ΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 61 -
1.6.1	ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΚΑΙ ΣΤΑΤΙΣΤΙΚΗ ΦΥΣΙΚΗ	- 61 -
1.6.2	ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΚΑΙ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 62 -
1.6.3	ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΚΑΙ ΙΑΤΡΙΚΗ	- 66 -
1.6.4	ΑΝΑΛΥΣΗ ΚΑΙ ΕΡΜΗΝΕΙΑ ΕΙΚΟΝΑΣ & ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 69 -
1.7	ΑΝΑΓΝΩΡΙΣΗ ΚΑΙ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΠΡΟΤΥΠΩΝ	- 72 -
1.7.1	ΕΠΙΛΟΓΗ ΚΑΤΗΓΟΡΙΟΠΟΙΗΤΗ	- 73 -
1.7.2	ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΣΥΝΑΡΤΗΣΕΩΝ ΑΚΤΙΝΙΚΗΣ ΒΑΣΗΣ	- 74 -
1.7.3	ΕΚΠΑΙΔΕΥΣΗ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ ΣΥΝΑΡΤΗΣΕΩΣ ΑΚΤΙΝΙΚΗΣ ΒΑΣΗΣ	- 75 -
1.7.4	ΜΕΘΟΔΟΣ ΕΚΠΑΙΔΕΥΣΗΣ ΜΕ ΕΝΑ ΠΡΟΤΥΠΟ ΑΝΑ ΚΕΝΤΡΟ	- 76 -
1.7.5	ΜΕΘΟΔΟΣ ΤΥΧΑΙΑΣ ΕΠΙΛΟΓΗΣ ΠΡΟΤΥΠΩΝ ΠΟΥ ΘΑ ΑΠΟΤΕΛΕΣΟΥΝ ΚΕΝΤΡΑ	- 76 -
1.7.6	ΜΕΘΟΔΟΣ ΟΜΑΔΟΠΟΙΗΣΗΣ ΤΩΝ Κ-ΜΕΣΩΝ	- 76 -
1.7.7	ΜΕΘΟΔΟΣ ΕΚΠΑΙΔΕΥΣΕΩΣ ΕΠΙΠΕΔΟΥ ΕΞΟΔΟΥ	- 77 -
1.7.8	ΣΥΓΚΡΙΣΗ Ν. Δ. ΣΥΝΑΡΤΗΣΕΩΣ ΑΚΤΙΝΙΚΗΣ ΒΑΣΗΣ ΚΑΙ Ν.Δ. ΠΟΛΛΩΝ ΕΠΙΠΕΔΩΝ	- 78 -
1.7.9	ΠΙΘΑΝΟΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	- 78 -
1.7.10	ΔΟΜΗ ΠΙΘΑΝΟΤΙΚΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ	- 78 -
2	<u>ΕΞΑΓΩΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΜΕ ΜΕΘΟΔΟΥΣ ΑΝΑΛΥΣΗΣ ΠΟΛΥΠΛΟΚΟΤΗΤΑΣ</u>	- 81 -
2.1	ΜΕΘΟΔΟΙ ΕΝΤΡΟΠΙΑΣ ΣΗΜΑΤΟΣ	- 81 -
2.1.1	ΜΕΘΟΔΟΣ ΤΗΣ ΠΡΟΣΕΓΓΙΣΤΙΚΗΣ ΕΝΤΡΟΠΙΑΣ (APPROXIMATE ENTROPY)	- 82 -
2.1.2	ΜΕΘΟΔΟΣ ΔΕΙΓΜΑΤΟΛΗΠΤΙΚΗΣ ΕΝΤΡΟΠΙΑΣ (SAMPLE ENTROPY)	- 82 -
2.1.3	ΜΕΘΟΔΟΣ ΑΣΑΦΟΥΣ ΕΝΤΡΟΠΙΑΣ (FUZZY ENTROPY)	- 83 -
2.1.4	ΣΥΓΚΡΙΣΗ ΠΡΟΣΕΓΓΙΣΤΙΚΗΣ ΕΝΤΡ., ΑΣΑΦΟΥΣ ΕΝΤΡ. ΚΑΙ ΔΕΙΓΜΑΤΟΛΗΠΤΙΚΗΣ ΕΝΤΡΟΠΙΑΣ	- 84 -
2.1.5	ΑΝΑΛΥΣΗ ΦΡΑΚΤΑΛ	- 85 -
3	<u>ΣΕΙΣΜΙΚΑ ΦΑΙΝΟΜΕΝΑ ΣΤΗΝ ΕΛΛΑΔΑ</u>	- 86 -
3.1	ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΣΕΙΣΜΟΥΣ	- 86 -
3.2	Η ΓΕΩΜΟΡΦΟΛΟΓΙΑ ΤΟΥ ΕΛΛΗΝΙΚΟΥ ΧΩΡΟΥ.	- 86 -
4	<u>ΤΗΛΕΜΑΤΙΚΟ ΔΙΚΤΥΟ ΜΕΤΡΗΣΗΣ ΗΛΕΚΤΡΟΜΑΓΝΗΤΙΚΩΝ ΣΗΜΑΤΩΝ</u>	- 89 -
5	<u>ΕΙΣΑΓΩΓΗ ΣΤΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ GUI ΜΕ ΤΟ ΜΑΤΛΑΒ</u>	- 91 -
6	<u>GUI ΓΙΑ ΤΗΝ ΔΙΑΧ. ΜΕΤΡ. Η/Μ ΠΕΔΙΟΥ ΑΠΟ ΠΟΛΛΑΠΛΟΥΣ ΣΤΑΘΜΟΥΣ ΜΕΤΡ.</u>	- 94 -
6.1	ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ	- 94 -

7	ΟΜΟΙΟΤΗΤΕΣ ΓΕΩΛΟΓΙΚΩΝ ΚΑΙ ΒΙΟΛΟΓΙΚΩΝ ΚΑΤΑΣΤΡΟΦΙΚΩΝ ΦΑΙΝΟΜΕΝΩΝ	- 100 -
8	ΜΕΛΕΤΗ ΠΟΛΥΠΛΟΚΩΝ ΒΙΟΛΟΓΙΚΩΝ ΔΙΕΡΓΑΣΙΩΝ	- 102 -
8.1	ΠΡΟΒΛΕΨΗ ΠΡΩΤΕΪΝΗΣ ΔΕΥΤΕΡΑΣ ΔΟΜΗΣ ΜΕ ΜΟΝΤΕΛΟ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 102 -
8.1.1	ΕΙΣΗΓΗΣΗ ΣΤΟ ΒΙΟΛΟΓΙΚΟ ΜΕΡΟΣ	- 102 -
8.1.2	ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΠΡΩΤΕΪΝΗΣ	- 103 -
8.1.3	ΑΝΑΛΥΣΗ ΤΗΣ ΠΡΟΒΛΕΨΗΣ ΔΕΥΤΕΡΕΥΟΥΣΑΣ ΔΟΜΗΣ	- 104 -
8.1.4	ΧΡΗΣΗ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΓΙΑ ΤΗΝ ΠΡΟΒΛΕΨΗ ΔΟΜΗΣ	- 104 -
8.1.5	ΑΝΑΠΤΥΞΗ ΜΟΝΤΕΛΟΥ ΤΕΧΝΗΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΓΙΑ ΤΗΝ ΠΡΟΒΛΕΨΗ	- 104 -
8.1.6	ΟΡΙΖΟΝΤΑΣ ΤΟ ΝΕΥΡΩΝΙΚΟ ΔΙΚΤΥΟ	- 106 -
8.1.7	ΔΗΜΙΟΥΡΓΩΝΤΑΣ ΤΟ ΝΕΥΡΩΝΙΚΟ ΔΙΚΤΥΟ ΑΝΑΓΝΩΡΙΣΗΣ ΠΡΟΤΥΠΟΥ	- 108 -
8.1.8	ΕΚΠΑΙΔΕΥΟΝΤΑΣ ΤΟ ΝΕΥΡΩΝΙΚΟ ΔΙΚΤΥΟ	- 109 -
8.2	ΑΝΑΛΥΟΝΤΑΣ ΤΗΝ ΑΠΟΚΡΙΣΗ ΤΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ	- 115 -
8.3	ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ ΓΙΑ ΠΙΟ ΑΚΡΙΒΗ ΑΠΟΤΕΛΕΣΜΑΤΑ	- 117 -
8.4	ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΑΠΟΔΟΣΗΣ ΤΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ	- 122 -
8.5	ΣΥΜΠΕΡΑΣΜΑΤΑ	- 124 -
9	ΒΙΒΛΙΟΓΡΑΦΙΑ	- 142 -
10	ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	- 142 -
11	ΚΑΤΑΛΟΓΟΣ ΕΞΙΣΩΣΕΩΝ	- 142 -
12	ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	- 142 -
13	ΕΥΡΕΤΗΡΙΟ ΚΥΡΙΟΤΕΡΩΝ ΟΡΩΝ	- 142 -
14	ΠΑΡΑΡΤΗΜΑ Α – ΙΣΤΟΡΙΚΕΣ ΑΝΑΦΟΡΕΣ ΣΕΙΣΜΩΝ ΤΗΣ ΑΡΧΑΙΟΤΗΤΑΣ	- 142 -
15	ΠΑΡΑΡΤΗΜΑ Β – ΑΝΕΠΤΥΓΜΕΝΟΣ ΚΩΔΙΚΑΣ ΤΗΣ ΠΑΡΟΥΣΑΣ ΠΤΥΧΙΑΚΗΣ	- 147 -
15.1	ΑΝΕΠΤΥΓΜΕΝΟΣ ΚΩΔΙΚΑΣ ΠΡΟΣΕΓΓΙΣΤΙΚΗΣ ΕΝΤΡΟΠΙΑΣ	- 147 -
15.2	ΑΝΕΠΤΥΓΜΕΝΟΣ ΚΩΔΙΚΑΣ ΔΕΙΓΜΑΤΟΛΗΠΤΙΚΗΣ ΕΝΤΡΟΠΙΑΣ	- 149 -
15.3	ΑΝΕΠΤΥΓΜΕΝΟΣ ΚΩΔΙΚΑΣ ΑΣΑΦΟΥΣ ΕΝΤΡΟΠΙΑΣ	- 151 -
15.4	GUI ΓΙΑ ΤΗΝ ΔΙΑΧ. ΚΑΙ ΜΕΛΕΤΗ Η/Μ ΠΕΔΙΟΥ ΑΠΟ ΣΤΑΘΜΟΥΣ ΜΕΤΡΗΣΗΣ	- 154 -
15.5	GUI ΓΙΑ ΓΡΑΦ. Η/Μ ΠΕΔΙΟΥ, ΠΥΚΝΟΤΗΤΑΣ ΦΑΣΜΑΤΟΣ ΙΣΧΥΟΣ ΚΑΙ ΑΣΑΦΟΥΣ ΕΝΤΡΟΠΙΑΣ.	- 380 -
15.6	ΑΝΕΠΤΥΓΜΕΝΟΣ ΚΩΔΙΚΑΣ ΓΙΑ ΤΗΝ ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΕΙΣΜΩΝ ΣΕ ΧΑΡΤΗ	- 450 -
15.7	ΠΑΡΑΘΕΣΗ ΚΩΔ. ΠΥΚΝ. ΦΑΣΜΑΤΟΣ ΙΣΧΥΟΣ ΤΟΥ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΤΟΥ ΚΟΛΟΡΑΝΤΟ	- 461 -
16	ΠΑΡΑΡΤΗΜΑ Γ – ΔΙΑΓΡΑΜΜΑΤΑ ΜΕΤΡΗΣΕΩΝ	- 476 -

1. ΕΙΣΑΓΩΓΗ ΣΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

1.1 Βιολογικά Νευρωνικά Δίκτυα

1.1.1 Εισαγωγή

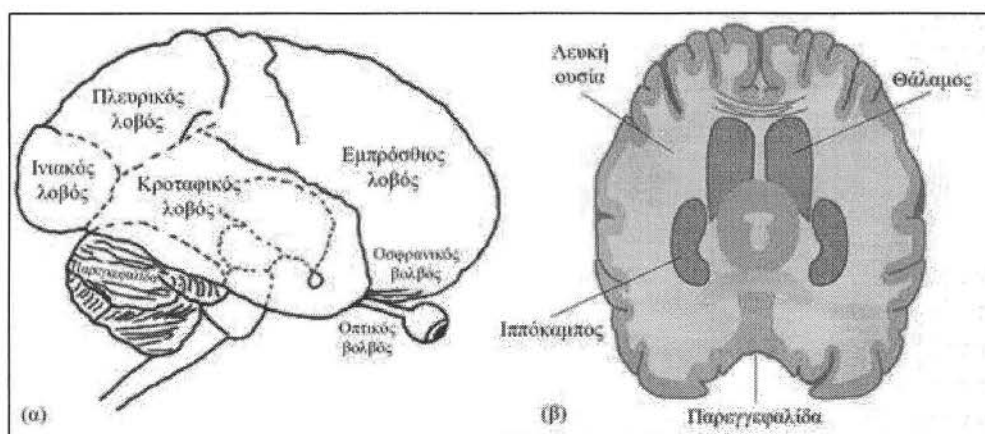
Οι πρώτες συγκρίσεις μεταξύ του ανθρώπου και του ηλεκτρονικού υπολογιστή έγιναν από την στιγμή της εφεύρεσης του. Ο ανθρώπινος εγκέφαλος είναι ουσιαστικά μια τεράστια επεξεργαστική μονάδα που αποτελείται από πολλούς «μικροεπεξεργαστές» παράλληλα συνδεδεμένους. Οι μικροεπεξεργαστές αυτοί είναι γνωστοί στην επιστημονική ορολογία με την ονομασία νευρώνες. Έχει υπολογιστεί ότι ο ανθρώπινος εγκέφαλος έχει κατά μέσο περίπου 100 δισεκατομμύρια νευρώνες. Μικρές ίνες είναι υπεύθυνες για την «παράλληλη» σύνδεση μεταξύ τους. Κάθε νευρώνας έχει περίπου δέκα χιλιάδες ίνες και το σημείο σύνδεσης της κάθε ίνας ονομάζεται σημείο σύναψης. Έτσι στο εσωτερικό του ανθρώπινου εγκεφάλου δημιουργείται ένα τεράστιο δίκτυο επεξεργασίας που ονομάζεται νευρωνικό δίκτυο και αποτελεί το πολυπληθέστερο από πλευράς συνδέσεων δίκτυο. Κάθε νευρώνας λειτουργεί χρονισμένα σε 10 εκατομμύρια κύκλους ανά δευτερόλεπτο. Στην μάχη μεταξύ υπολογιστή και ανθρώπου για την ταχύτητα επεξεργασίας των δεδομένων, την υπολογιστική ισχύ δηλαδή, νικητής είναι και με τεράστια διαφορά ο άνθρωπος. Ο μεγαλύτερος υπερυπολογιστής σήμερα έχει ταχύτητα 12,3 τρισεκατομμύρια πράξεις το δευτερόλεπτο. Ο ανθρώπινος εγκέφαλος αποτελείται από 100 δισεκατομμύρια νευρώνες και υπάρχουν 1000 τρισεκατομμύρια συνδέσεις, αν για κάθε σύνδεση έχουμε 100 λειτουργίες ανά δευτερόλεπτο τότε η υπολογιστική ισχύς του ανθρώπου ανέρχεται σε 100.000 τρισεκατομμύρια πράξεις το δευτερόλεπτο! Ο ανθρώπινος εγκέφαλος είναι ταχύτερος κατά 8130 φορές από τον ταχύτερο υπερυπολογιστή. Τα νευρωνικά δίκτυα έχουν μεγάλη προσαρμοστικότητα που οφείλεται στις εξαιρετικά ευαίσθητογόνες συνδέσεις τους. Για παράδειγμα ο αμφιβληστροειδής χιτώνας του ματιού έχει την δυνατότητα να προσαρμόζεται άμεσα σε κλάσματα δευτερολέπτου σε διαφορετικές καταστάσεις φωτισμού. Επίσης οι νευρώνες επενεργούν επεξεργάζοντας μόνο τα δικά τους «τοπικά δεδομένα». Πολλοί επιστήμονες προσπαθούν να μιμηθούν την τεχνική αυτή διασύνδεσης των νευρώνων φτιάχνοντας τα λεγόμενα τεχνητά νευρωνικά δίκτυα. Σκοπός, είναι η δημιουργία έξυπνων μηχανών, είναι εμφανές ότι οι έννοιες της τεχνητής νοημοσύνης και των τεχνητών νευρωνικών δικτύων συμβαδίζουν.

Πριν από λίγα χρόνια Ρώσοι ερευνητές κατασκεύασαν τον πρώτο τεχνητό εγκέφαλο ενώ στην Καλιφόρνια γίνεται προσπάθεια κατασκευής τεχνητού εγκεφάλου από πραγματικούς νευρώνες εμβρύου ποντικού που συνδέονται με πίνακα ηλεκτροδίων μέσω των οποίων επικοινωνούν. Στο Βέλγιο ομάδα επιστημόνων συνεργάζεται για την κατασκευή ενός μηχανικού ρομπότ με την μορφή γάτας. Ο εγκέφαλος του ρομπότ είναι ένας ηλεκτρονικός υπολογιστής που λειτουργεί ως νευρωνικό δίκτυο με 65.000 τμήματα. Κάθε τμήμα του δικτύου αυτού αποτελείται από

1152 νευρώνες. Το δίκτυο αναπτύσσεται γνωστικά χρησιμοποιώντας γενετικό αλγόριθμο. Η IBM το 2012 στην έκθεση “Supercomputing 2012” παρουσίασε το SyNAPSE (Systems of Neuromorphic Adaptive Plastic Scalable Electronics). Το SyNAPSE ήταν ένα χρηματοδοτούμενο από το DARPA πρόγραμμα το οποίο ξεκίνησε το 2008, με στόχο την ανάπτυξη νευρομορφικής τεχνολογίας (τεχνολογία προσομοίωσης ανθρώπινου εγκεφάλου) με δυνατότητα κλιμάκωσης σε βιολογικά επίπεδα, χρησιμοποιώντας αρχιτεκτονική Γνωσιακής Υπολογιστικής (Cognitive Computing) προσομοιώνονται 1010 νευρώνες (10 δις) και 1014 συνάψεις (100 τρισεκατομμύρια, με βάση τον υπολογισμό του αριθμού των συνάψεων στον ανθρώπινο εγκέφαλο). Κατά την IBM αυτός ο στόχος επιτεύχθηκε με το σύστημα “TrueNorth” το οποίο τρέχει στο Lawrence Livermore National Lab με 1,572,864 επεξεργαστικούς πυρήνες, 1,5 Petabyte μνήμης 6,291,456 threads. Η προσομοίωση ήταν μόλις 1542 φορές πιο αργή σε σχέση με την πραγματικότητα.

1.1.2 Οι νευρώνες του ανθρώπινου εγκεφάλου

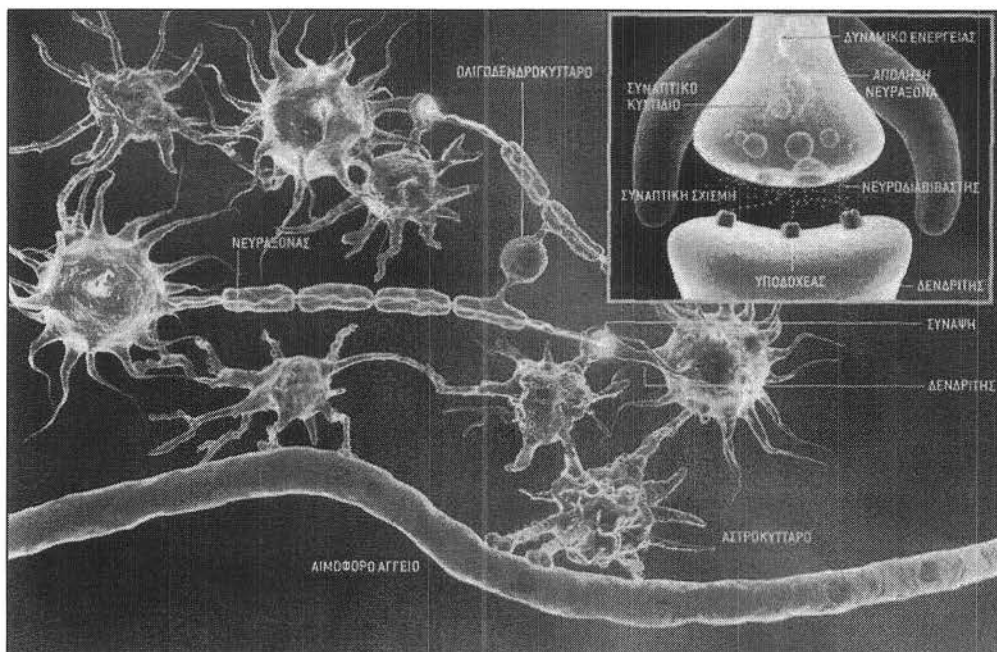
Η μελέτη του ανθρώπινου εγκεφάλου ήταν για την επιστήμη μία πραγματική πρόκληση. Τις γνώσεις μας τις οφείλουμε στις «σύγχρονες» επιστήμες τις νευροανατομίας και νευροφυσιολογίας καθώς και άλλων επιστημών που σχετίζονται με τον ανθρώπινο εγκέφαλο. Ένα μέρος της γνώσης μας οφείλεται και στην εμπειρική κατανόηση πολλών λειτουργιών του. Η πλήρης αποκρυπτογράφηση του είναι ένα ακόμα μεγάλο κάστρο για τις σύγχρονες επιστήμες. Μερικές λειτουργίες του όπως η νόηση, η ευφυΐα και τα συναισθήματα είναι ακόμα ανεφάρμοστες για τις υπολογιστικές μηχανές.



Εικόνα 1. (α) Πλάγια όψη - (β) Γωνιακή τομή

Το νευρικό σύστημα το αποτελούν δυο κατηγορίες κυττάρων τα νευρογλοιακά κύτταρα και τους νευρώνες. Ο ανθρώπινος εγκέφαλος αποτελείται από 100 περίπου δισεκατομμύρια νευρώνες που το μήκος τους είναι από 0,005mm έως

0,1mm περίπου. Τα νευρογλοιακά κύτταρα είναι περίπου δέκα φορές μικρότερα και περισσότερα από τους νευρώνες στον ανθρώπινο εγκέφαλο (Kalat, 1995), (Kandel, 1999).



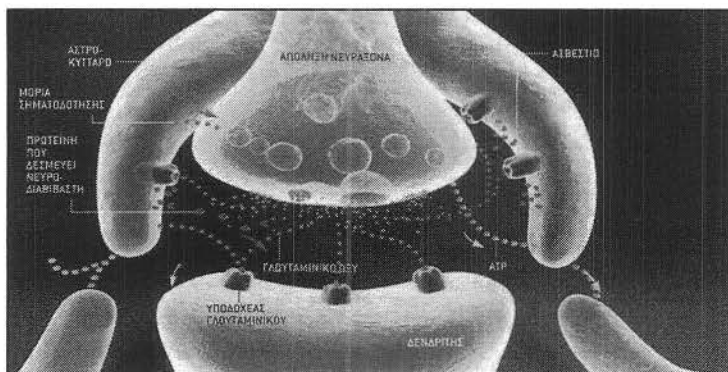
Εικόνα 2. Νευρώνικά κύτταρα. Πηγή: Scientific American
Μετάφραση: http://exeldim.site40.net/tyfli/nevrwnes_a.htm

Τα νευρωνικά κύτταρα λαμβάνουν πληροφορίες και τις μεταδίδουν σε άλλα κύτταρα με την μορφή ηλεκτροχημικών ώσεων. Τα νευρογλοιακά κύτταρα διαχωρίζονται σε τρεις κατηγορίες τα αστροκύτταρα, τους ολιγοδενδρίτες και τα μικρογλοιακά κύτταρα. Τα αστροκύτταρα που πήραν το όνομα τους από το σχήμα τους που μοιάζει σε αστέρι είναι και τα πλέον πολυάριθμα. Η δράση τους έγκειται στο να τροφοδοτούν τους νευρώνες με θρεπτικές ουσίες και να απομακρύνουν τις ουσίες που εκκρίνουν οι νευρώνες. Τέτοιες ουσίες αποτελούν οι νευροδιαβιβαστές οι οποίοι εκκρίνονται από τους νευρώνες μετά την μεταβίβαση των πληροφοριών. Οι ολιγοδενδρίτες παράγουν μυελίνη μια ουσία που προστατεύει τις νευρικές ίνες. Τα μικρογλοιακά κύτταρα έχουν αμυντική δράση και εκκρίνουν κυτταροτοξίνες απέναντι σε μικρόβια που είναι δυνατόν να προσβάλουν νευρικούς ιστούς του ανθρώπινου εγκεφάλου (Kalat, 1995).

Τα νευρογλοιακά κύτταρα συμβάλλουν ενεργά στην μεταβίβαση των πληροφοριών και μάλιστα με μεγάλη πιθανότητα να αποτελούν την ιδιότητα που κάνει τον εγκέφαλο ικανό για υψηλές, νοητικές επιδόσεις στη μαθησιακή και μνημονική διαδικασία.

Ερευνώντας τα επίπεδα ιόντων ασβεστίου και τις διακυμάνσεις τους στους νευρώνες ανακαλύφθηκε ότι τα νευρογλοιακά κύτταρα έχουν την δυνατότητα

επικοινωνίας μεταξύ τους αλλά μπορούν επίσης να παρακολουθούν , να ενημερώνονται για την επικοινωνία των νευρώνων και να επεμβαίνουν (Kettemann & Ransom, 2005) Τα νευρογλοιακά κύτταρα παράγουν μία μη αναγνωρίσιμη ουσία η οποία εντείνει και ρυθμίζει την λειτουργία των σχηματιζόμενων συνάψεων των νευρώνων. Έτσι κατά κάποιο τρόπο ελέγχουν ίσως ορισμένες λειτουργίες των νευρώνων και έχουν την δυνατότητα να ενεργοποιούν και απενεργοποιούν ορσμένα νευρωνικά κυκλώματα και επεμβαίνουν στις διαδικασίες μνήμης και μάθησης (Baer, et al., 2006).



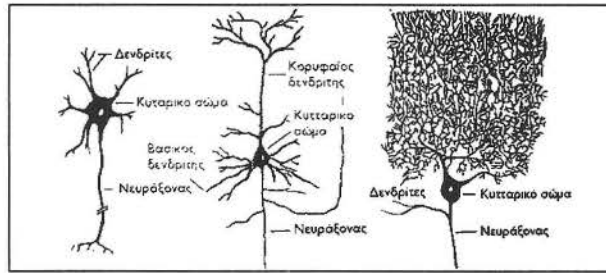
Εικόνα 3. Μεγένθυση Νευρωνικού Κυττάρου. Πηγή: Scientific American
Μετάφραση: http://exeldim.site40.net/tyfli/nevrwnes_a.htm

1.1.3 Δομικά Στοιχεία Νευρώνα

Η δομή των νευρώνων αποτελείται από το κυτταρικό σώμα, τους δενδρίτες, το νευροάξονα και τις συνάψεις (Kandel, 1999), (Kaplan, 1995).

Στο κυτταρικό σώμα περιέχεται ο πυρήνας, τα μιτοχόνδρια, τα ριβοσώματα και άλλα μορφώματα. Η ζωή και η καλή λειτουργία του νευρώνα εξαρτάται από το κυτταρικό σώμα που ο λειτουργικός του σκοπός είναι να ενσωματώνει οι νέες πληροφορίες που εισέρχονται, μπορεί να χαρακτηριστεί και ως «αποθήκη» των γενετικών πληροφοριών. Επίσης στο κυτταρικό σώμα πραγματοποιείται το μεγαλύτερο μέρος της μεταβολικής και συνθετικής δραστηριότητας.

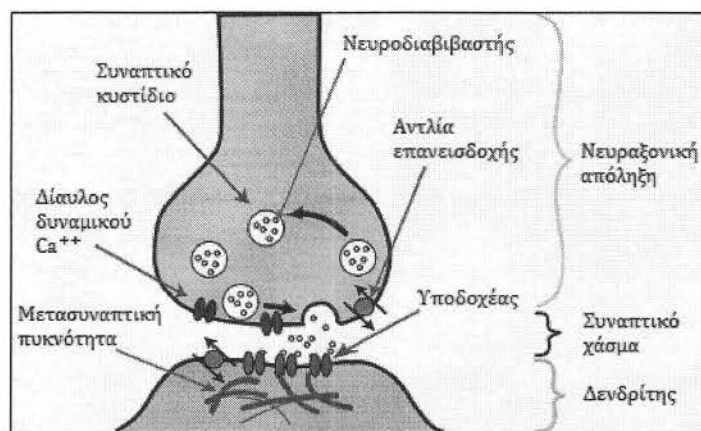
Οι δενδρίτες είναι λεπτές ίνες με δενδρικό διακλαδισμό (εξ' ου και το όνομα) που λεπταίνουν ακόμα περισσότερο καθώς απομακρύνονται από το κυτταρικό σώμα καθ' ομοίωση ενός δέντρου. Οι περισσότεροι νευρώνες έχουν στην επιφάνεια τους πολλαπλούς δενδρίτες οι οποίοι δέχονται ηλεκτροχημικές ώσεις, δηλαδή σήματα που αποτελούνται από διαφορετικές πληροφορίες από άλλους νευρώνες που μεταφέρονται στο κυτταρικό σώμα. Το εύρος της επιφάνειας του δενδρίτη είναι ανάλογο με το πλήθος των πληροφοριών που δέχεται. Ο νευράξονας είναι μια ίνα σωληνοειδούς μορφής με μεγαλύτερο πάχος και μήκος από τους δενδρίτες ο οποίος μέσω ενός δυναμικού ενέργειας μεταφέρει την πληροφορία από το σώμα του κυττάρου προς άλλα κύτταρα στόχους.



Εικόνα 4. Απο αριστερά προς τα δεξιά: α) Κινητικός νευρώνας νωτιαίου μυελού β) Πυραμιδοειδές κύτταρο του ιπποκάμπου γ) Κύτταρο Purkinje της παρεγκεφαλίδας

Οι νευρώνες που παρουσιάζουν έλλειψη νευραξόνων, μπορούν να μεταφέρουν πληροφορίες μόνο σε γειτνιάζοντες νευρώνες. Οι μεγαλύτεροι νευράξονες περιβάλλονται από ένα κολλώδες περίβλημα που ονομάζεται μυελώδες έλυτρο και προστατεύει τον νευράξονα από ηλεκτρικές παρεμβάσεις άλλων γειτονικών νευρώνων. Έτσι διασφαλίζεται η γρήγορη μετάδοση των δυναμικών ενέργειας. Κατά μήκος του νευράξονα το έλυτρο διακόπτεται από τους κόμβους Ranvier, οι οποίοι ανανεώνουν και επιταχύνουν την μεταβίβαση του μηνύματος.

Ο νευράξονας στις άκρες του σχηματίζει ογκίδια που ονομάζονται τελικά κομβία ή κομβικές απολήξεις. Τα κομβία αυτά σχηματίζουν μια σχισμή που ονομάζεται σύναψη. Η σύναψη λειτουργεί ως διεπαφικός και επικοινωνιακός χώρος ανάμεσα στα τελικά κομβία ενός ή περισσότερων νευρώνων και των δενδριτών, ή και μεταξύ του κυτταρικού σώματος και του νευρώνα. Ο ανθρώπινος εγκέφαλος έχει περίπου 10^{11} νευρώνες καθ' ένα από τους οποίους φέρει 10^3 έως 10^{14} συναπτικές συνδέσεις σε σύνδεση με περίπου 10^3 άλλους νευρώνες. Ο αριθμός των συνάψεων στον εγκεφαλικό φλοιό υπολογίζεται σε περίπου 10^{14} .



Εικόνα 5. Διάγραμμα ηλεκτροχημικού σήματος Νευροδιαβιβαστών.
Πηγή: http://www.foundalis.com/dep/cog/N4_gr.htm

1.1.4 Βασικές Λειτουργίες

Οι νευρώνες ανάλογα με το τι λειτουργία επιτελούν μπορούν να διαχωριστούν σε τρεις κατηγορίες:

- Τους αισθητηριακούς που έχουν την ικανότητα της μεταφοράς πληροφοριών από περιφερειακούς αισθητηριακούς υποδοχείς
- Τους κινητικούς που είναι υπεύθυνοι για τις κινητικές εντολές από τον εγκέφαλο στους μυς αλλά και στους αδένες μεταφέροντας δηλαδή ώσεις κινητικής ενεργοποίησης
- Και τους διάμεσους που αποτελούν την μεγαλύτερη ομάδα και έργο τους είναι η μεταφορά πληροφοριών σε τοπικό επίπεδο μεταξύ των γειτονικών νευρώνων αλλά και σε μεγάλες αποστάσεις στο εσωτερικό του εγκεφάλου.

1.1.5 Νευροβιολογική και νευροχημική επεξεργασία πληροφοριών

Με βάση τις τεχνικές απεικόνισης του ανθρώπινου εγκεφάλου μπορούμε να διαπιστώσουμε την ενεργοποίηση ενός μεγάλου δικτύου εγκεφαλονευρώνων κάθε φορά που ο άνθρωπος μεταβάλλει την συμπεριφορά του. Τέτοιες μεταβολές μπορεί να είναι απλές όπως είναι τα αντανακλαστικά αλλά και πιο σύνθετες όπως είναι οι γνωστικές λειτουργίες (σκέψη, γλώσσα, αντίληψη κλπ.). Οι νευρώνες που αποτελούν αυτό το μεγάλο δίκτυο επικοινωνούν μεταξύ τους με ηλεκτρομαγνητικό και χημικό τρόπο απαρτίζοντας έτσι τα λεγόμενα εγκεφαλικά λειτουργικά συστήματα. Η επικοινωνία σε αυτά τα νευρωνικά κυκλώματα γίνεται μέσω των συνάψεων διεγείροντας ή καταστέλλοντας λειτουργίες τους μέσω μιας ηλεκτρομαγνητικής και χημικής διεργασίας που συντελείται (Kalat, 1995), (Kalat, 1995), (Papageorgiou & Wells, 1999), (Anderson, 1995), (Gazzaniga, 1996)

1.1.6 Νευροδιαβιβαστές

Σε συνέχεια με την προηγούμενη παράγραφο οι ουσίες που απελευθερώνονται κατά την χημική διεργασία ονομάζονται νευροδιαβιβαστές και είναι οι αγγελιαφόροι των πληροφοριών. Ενώ υπάρχουν και οι νευροτροποποιητές οι οποίοι τροποποιούν λειτουργίες (Kalat, 1995), (Kandel, 1999), (Lymperakis, 1997), (Purves & Andrews, 1997).

Οι τρεις κύριες κατηγορίες νευροδιαβιβαστών είναι:

- οι βιογενείς αμίνες
- τα αμινοξέα
- και τα πεπτίδια

Οι βασικοί νευροδιαβιβαστές έχουν τα εξής χαρακτηριστικά:

- ημιοργανώνται στο νευρικό κύτταρο

- αποθηκεύονται στα κυστίδια που βρίσκονται στις απολήξεις των νευρώνων
- απελευθερώνουν ορισμένο ποσό χημικών ουσιών

Ουσίες με τέτοιες ιδιότητες είναι οι κατεχολαμίνες. Μέρος της οικογένειας της είναι οι ουσίες σεροτονίνη, ακετυλοχολίνη, ντοπαμίνη, και η νοραδρεναλίνη που θα μελετήσουμε παρακάτω αλλά και άλλες ουσίες όπως η γλουταμίνη, η αδρεναλίνη, το γ-αμινοβουτυρικό οξύ κλπ.

Ας μελετήσουμε όμως τις πιο σημαντικές από τις παραπάνω ουσίες:

- Σεροτονίνη (5-HT): ένας από τους σημαντικότερους νευροδιαβιβαστές του κεντρικού νευρικού συστήματος. Σαν ουσία βρίσκεται στο εσωτερικό ποικίλων ορμονών και η δράση έγκειται στην εκδήλωση της συμπεριφοράς του ανθρώπου, στην εναλλαγή συναισθημάτων, στον επηρεασμό της ορθής λειτουργίας της όρεξης και του ύπνου.
- Ντοπαμίνη (DA): Η συγκεκριμένη ουσία είναι υπεύθυνη για τον επηρεασμό λειτουργιών όπως η προσοχή, η μάθηση και η κίνηση. Ενώ αξίζει να αναφερθεί ότι η υπολειτουργία των νευρώνων που εκλύουν ντοπαμίνη οφείλεται για την νόσο του Πάρκινσον ενώ αντίθετα η υπερλειτουργία ντοπαμίνης στον υποθάλαμο οφείλεται για την σχιζοφρένεια και διάφορες άλλες διαταραχές προσωπικότητας.
- Ακετυλοχολίνη (ACH): Η ουσία αυτή βρίσκεται στον ιππόκαμπο του ανθρώπινου εγκεφάλου και η δράση της μπορεί να είναι διεγερτική ή ανασταλτική. Ανεπάρκεια της συγκεκριμένης ουσίας οφείλεται για την νόσο Αλτσχάιμερ, για την γεροντική άνοια ενώ η ορθή ποσότητα της συγκεκριμένης ουσίας είναι υπεύθυνη για την σωστή λειτουργία του μνημονικού συστήματος και την εκούσια κίνηση των μυών.
- Νοραδρεναλίνη: Η συγκεκριμένη ουσία είναι υπεύθυνη για την ρύθμιση της συμπεριφοράς ενώ η ανεπάρκεια της είναι υπεύθυνη για συναισθηματικές διαταραχές. Δύο παρεμφερής νόσοι η κατάθλιψη και η μανιοκατάθλιψη οφείλονται στα χαμηλά και αντίστοιχα στα υψηλά επίπεδα νοραδρεναλίνης.

1.1.7 Υποδοχείς Νευροδιαβιβαστών

Κάθε νευροδιαβιβαστής έχει περισσότερους από ένα υποδοχείς. Οι υποδοχείς αποτελούνται από πρωτεϊνικά μόρια που βρίσκονται ενσωματωμένα στην κυτταρική

μεμβράνη και ανάλογα με την βιοχημική σύσταση εντοπίζουν τους ειδικούς νευροδιαβιβαστές (Kalat, 1995), (Lymperakis, 1997).

Ανάλογα με την μορφή της ενέργειας των ερεθισμάτων στο εσωτερικό και στο εξωτερικό περιβάλλον του ανθρωπίνου οργανισμού οι υποδοχείς διακρίνονται σε: α) Χημικούς, β) θερμικούς, γ) μηχανικούς, δ) οπτικούς, ε) ηλεκτρικούς, ζ) μαγνητικούς κλπ.

Πρέπει να αναφερθεί ότι ο αριθμός των νευροδιαβιβαστών διαφέρει από άνθρωπο σε άνθρωπο και ο ανάλογος αριθμός των παραπάνω κατηγοριών των νευροδιαβιβαστών καθορίζουν την συμπεριφορά του ατόμου ανάλογα την υπερλειτουργία ή υπολειτουργία κάποιου από τα είδη των συνάψεων. Οι συνάψεις αποτελούν τα σημεία δράσης όλων των ψυχοδραστικών ουσιών. Μία ουσία έχει την δυνατότητα να μιμείται, να ενισχύει ή και να αναστέλλει την δράση ενός συγκεκριμένου νευροδιαβιβαστή. Κάθε ουσία που αναστέλλει την δράση ενός νευροδιαβιβαστή ονομάζεται ανταγωνιστής ενώ αντίθετα οι ουσίες που μιμούνται και ενισχύουν την δράση του κάθε νευροδιαβιβαστή ονομάζονται αγωνιστές. Οι διάφορες ουσίες μπορούν να δράσουν με πολλούς διαφορετικούς τρόπους στο νευρικό σύστημα διευκολύνοντας ή αναστέλλοντας την σύνθεση κάποιου νευροδιαβιβαστή, αυξομειώνοντας την απελευθέρωση ή και την τροποποίηση της δράσης τους έπειτα από την δέσμευση του ίδιου του νευροδιαβιβαστή στον υποδοχέα (Kalat, 1995). Οι ψυχοδραστικές ουσίες επηρεάζουν τον εγκέφαλο και το νευρικό σύστημα με δύο τρόπους: α) θεραπευτικά με ιατρικές χρήσεις συγκεκριμένων φαρμάκων β) βλαπτικά μέσω των διάφορων ναρκωτικών ουσιών.

Οι περισσότερες ψυχοτρόπες ουσίες είτε είναι νευροδιαβιβαστές ή και να λειτουργούν σαν αυτούς. Με ένα σύνολο πολυάριθμων και ποικίλων ενδοκυτταρικών και εξωκυτταρικών μηχανισμών μεταβάλλουν την ισορροπία ανάμεσα στις διεγερτικές και στις κατασταλτικές εγκεφαλικές διεργασίες προκαλώντας πολλές φορές εθισμό (Liappas, 1999).

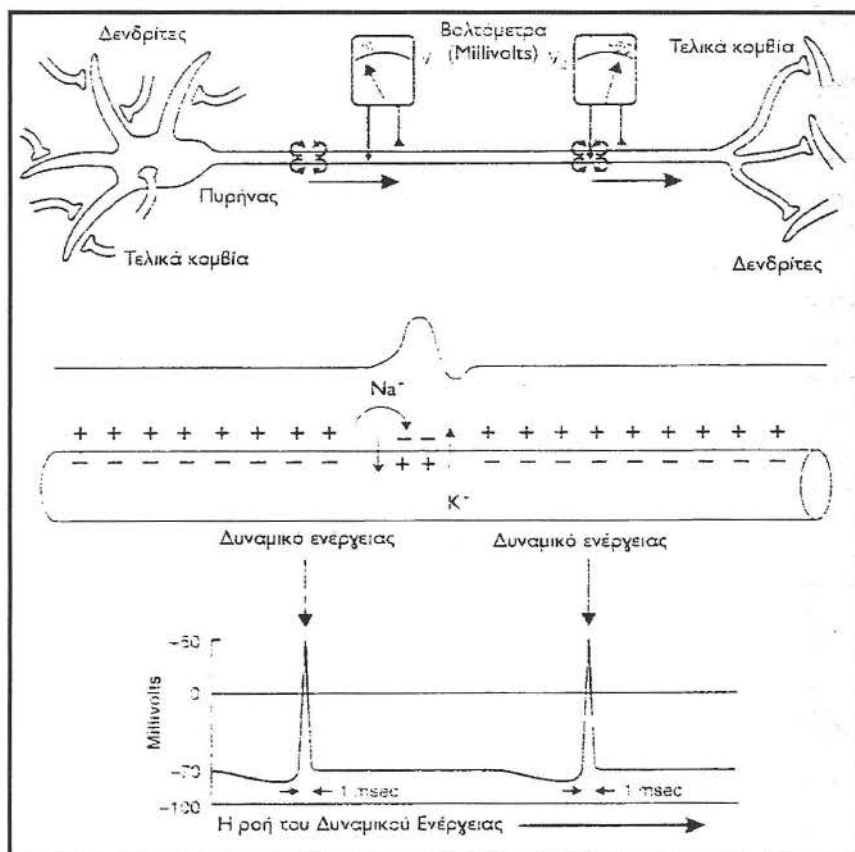
1.1.8 Ενδονευρωνική Επικοινωνία

Ένα εκ των χαρακτηριστικών των νευρώνων είναι η δυνατότητα τους να επικοινωνούν γρήγορα και με ακρίβεια εξ αποστάσεως με άλλους νευρώνες του εγκεφάλου αλλά και νευρώνες που θεωρούνται εκτός νευρικού συστήματος όπως είναι οι αδενικοί νευρώνες και οι μυϊκοί νευρώνες. Η ενδονευρωνική και η διανευρωνική επικοινωνία επιτυγχάνεται με τους δενδρίτες που είναι υπεύθυνοι για την υποδοχή των πληροφοριών από τους άλλους νευρώνες και διαμέσου του νευράξονα που απολήγει στα τελικά κομβία και μεταδίδει την πληροφορία και σε άλλους νευρώνες. Η μετάδοση των πληροφοριών εξαρτάται από μια σειρά ηλεκτρικών και χημικών διεργασιών που γίνονται στο εσωτερικό των νευρώνων. Η διαφορά των νευρώνων με τα άλλα νευρωνικά κύτταρα είναι ότι η μετάδοση των πληροφοριών οφείλεται στην σπάνια ιδιομορφία τους με κύριο χαρακτηριστικό αυτό της διεγερσιμότητας. Σύμφωνα με αυτή οι νευρώνες παρουσιάζουν μικρομεταβολές στο ηλεκτρικό δυναμικό τους σε

πολύ σύντομο χρονικό διάστημα λόγω της ροής ιόντων και προς τις δύο πλευρές της κυτταρικής μεμβράνης τους.

Ορισμένες πρωτεΐνες της κυτταρικής μεμβράνης σχηματίζουν διαύλους οι οποίοι επιτρέπουν σε συγκεκριμένα ανόργανα ιόντα, όπως το Νάτριο, το Κάλιο και το Ασβέστιο, να περάσουν στο εσωτερικό του κυττάρου και να διασφαλίσουν την γρήγορη ροή του ηλεκτρικού ρεύματος. Ο κάθε νευρώνας έχει μια διαφορά δυναμικού της τάξης των -70mV από το εσωτερικό του στο εξωτερικό του. Αυτή η διαφορά δυναμικού ονομάζεται δυναμικό ηρεμίας (Kandel, 1999).

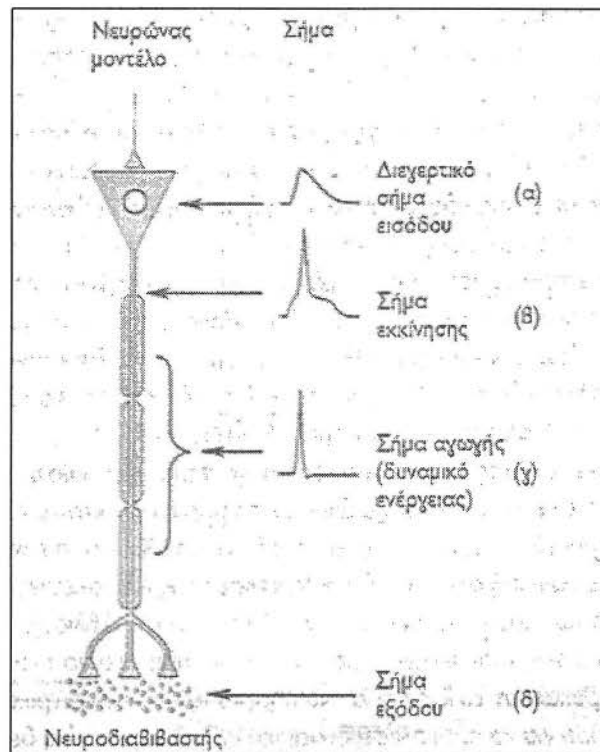
Κατά την διάρκεια της διανευρωνικής επικοινωνίας στους νευροδιαβιβαστές υπάρχει μία μείωση της διαφοράς δυναμικού από -70mV σε -55mV εκείνη την στιγμή έχουμε την έκλυση του δυναμικού ενέργειας ή δράσης. Παρατηρείται ότι κατά την διάρκεια του δυναμικού ενέργειας του νευρώνα η διαφορά δυναμικού αυξάνεται απότομα στα $+50\text{mV}$ και στην συνέχεια έχουμε απότομη μείωση στα -70mV του δυναμικού ηρεμίας. Η συνολική διάρκεια του δυναμικού ενέργειας είναι περίπου 1msec .



Εικόνα 6. Μηχανισμός δημιουργίας και ροής δυναμικού κατά (Walker, 1996)

Το δυναμικό ενέργειας αποτελεί το μεταδιδόμενο σήμα που μεταφέρεται μέσα στον άξονα από το ένα άκρο στο άλλο του νευρώνα με την μορφή ηλεκτρικών ώσεων. Τα ερεθίσματα που δεν φτάνουν στο επίπεδο διέγερσης δεν προκαλούν σήμα ενώ αντίθετα τα ερεθίσματα που το φτάνουν ή και το υπερβαίνουν προκαλούν το ίδιο σήμα. Το ηλεκτρικό φορτίο δεν παίρνει φθίνουσα πορεία ως προς την ένταση τόσο στην περίπτωση που είναι αρκετά ισχυρό ώστε να προκαλέσει δυναμικό ενέργειας τόσο και στην αντίθετη. Το ηλεκτρικό φορτίο παραμένει σταθερό και αναγεννάτε στους κόμβους Ranvier καθώς μεταφέρεται από την αρχή έως το τέλος του αποδέκτη νευρώνα. Όσο μεγαλύτερη είναι η ένταση και η διάρκεια του ερεθίσματος τόσο μεγαλύτερη είναι η διάρκεια και η συχνότητα των δυναμικών ενέργειας, υπάρχει δηλαδή αναλογία.

Η πρόκληση μιας συμπεριφοράς γίνεται από το σύνολο των νευρώνων διαφορετικού τύπου μέσω των ηλεκτρικών σημάτων στα διάφορα σημεία του κυττάρου. Σύμφωνα με το παρακάτω σχήμα ο κάθε νευρώνας έχει τέσσερις λειτουργικές περιοχές διαφορετικής εκπομπής σημάτων.



Εικόνα 7. Αναπαράσταση των τεσσάρων λειτουργιών του νευρώνα κατά την διάρκεια της ηλεκτρικής μεταβίβασης ενός σήματος κατά (Kandel, 1999)

α) Αρχικά τα διάφορα αισθητηριακά ερεθίσματα έχουν επίδραση στην επιφάνεια υποδοχής της κυτταρικής μεμβράνης. Στην συνέχεια πρωτεΐνες του υποδοχέα σχηματίζουν διαύλους ιόντων δηλαδή μικρο-οπές στην μεμβράνη από τις οποίες ρέουν ανόργανα ιόντα (νάτριο και κάλιο). Η ροή ιοντικού ρεύματος μέσω των

οπών μεταβάλλει το δυναμικό ηρεμίας της κυτταρικής μεμβράνης. Η συγκεκριμένη μεταβολή αντιπροσωπεύει τα διεγερτικά τοπικά σήματα εισόδου. Συνολικά σε αυτήν την μικρο-χρονική περίοδο συμπεραίνουμε ότι τα εξωτερικά και εσωτερικά ερεθίσματα μετασχηματίζονται σε ηλεκτρική ενέργεια.

β) Η περιοχή αυτή αποτελεί την ζώνη εκκίνησης όπου το αρχικό σήμα εισόδου μετατρέπεται και το δυναμικό ενέργειας δρομολογείται η συνολική ενεργοποίηση του νευρώνα θα μεταβιβάσει το μήνυμα κατά μήκος του νευράξονα. Η κυτταρική μεμβράνη εκπολώνεται και έχουμε απότομη μείωση ηλεκτρικού φορτίου. Η μεταβολή του δυναμικού στην μεμβράνη ανοίγει τις οπές-διαύλους και αφήνει το νάτριο να μειώσει την διαφορά στο εσωτερικό και στο εξωτερικό περιβάλλον του κύτταρου.

γ) Η περιοχή της αγωγής αποτελεί το μεγαλύτερο μέρος του νευράξονα και εδώ συντελείται η μεταφορά του δυναμικού ενέργειας. Αυτά τα σήματα πληροφορίες έχουν παρόμοιο σχήμα και κυματομορφή. Το σήμα πληροφορία καθορίζεται από την ειδική οδό του εγκεφάλου που πηγαίνει το σήμα. Ειδικά τμήματα του φλοιού ερμηνεύουν και συγχρονίζουν τις πληροφορίες των εισερχόμενων ηλεκτρικών ώσεων με συγκεκριμένο τρόπο ώστε να δημιουργούν την οπτικοακουστική αντίληψη (Kandel, 1999)

δ) Αυτό το τελευταίο κομμάτι αφορά το χημικό μέρος. Την στιγμή που το δυναμικό ενέργειας καταφτάνει στις απολήξεις των νευρώνων χημικές ουσίες απελευθερώνονται για να μεταβιβάσουν την πληροφορία στον επόμενο νευρώνα, οι ουσίες αυτές όπως αναφέραμε και προηγουμένως ονομάζονται νευροδιαβιβαστές.

1.1.9 Διανευρωνική επικοινωνία

Οι πιο σύνθετες γνωστικές λειτουργίες του ανθρώπινου εγκεφάλου όπως η μνήμη, η σκέψη η μάθηση και η αντίληψη οφείλονται στους μηχανισμούς της συναπτικής διαβίβασης των νευρώνων. Η σύνθεση και απελευθέρωση των νευροδιαβιβαστών γίνεται μέσω ενός πολύπλοκου και αρκετά σύνθετου συστήματος βιοχημικών μηχανισμών το οποίο προσαρμόζει άμεσα το νευρικό σύστημα στις γρήγορες περιβαλλοντικές μεταβολές. Το ηλεκτρικής μορφής σήμα από την στιγμή που φτάνει καταληκτικές συνάψεις του νευρώνα πυροδοτεί την απελευθέρωση των νευροδιαβιβαστών στις συνάψεις των νευρώνων. Οι χημικές ουσίες που ονομάζουμε νευροδιαβιβαστές στην συνέχεια θα προχωρήσουν προς την μετασυναπτική μεμβράνη του επόμενου νευρώνα οπότε και θα έχουμε την ηλεκτρική μεταβολή, που έχουμε περιγράψει και προηγουμένως, ικανή να διεγείρει ή να αναστείλει την μετάδοση του σήματος του μηνύματος στον επόμενο νευρώνα (Kandel, 1999), (Purves & Andrews, 1997).

Παρακάτω θα περιγράψουμε την αλληλουχία της ηλεκτρικής μεταβίβασης των σημάτων μεταξύ των νευρωνικών συνάψεων όπως φαίνονται και στο σχήμα.

1. Στον πρώτο νευρώνα έχουμε την σύνθεση και την συγκέντρωση του νευροδιαβιβαστή στα συναπτικά κυστίδια. Από το σύνολο των ουσιών του νευροδιαβιβαστή τα λιπίδια, τις πρωτεΐνες κλπ. Έχουμε

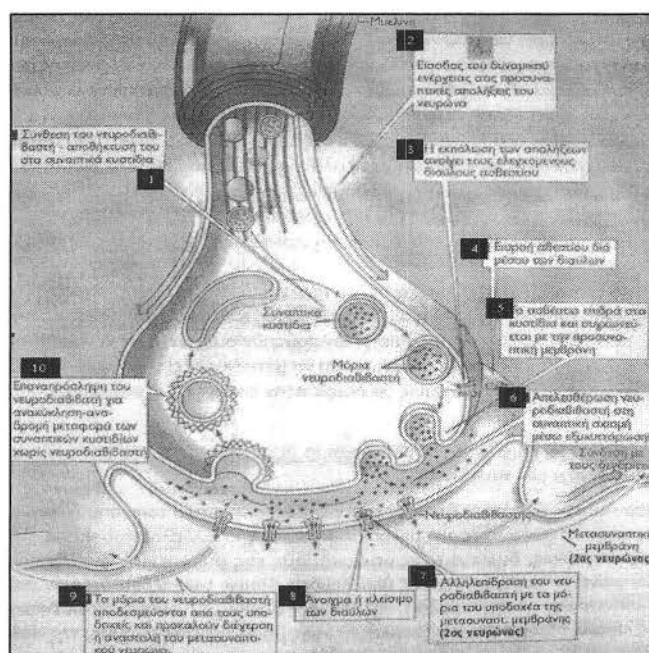
- τον σχηματισμό των οργανιδίων και προώθηση τους στο κυτταρικό σώμα προς τον νευράξονα.
2. Το δυναμικό ενέργειας εισάγεται στις προσυναπτικές απολήξεις του νευρώνα και μεταβιβάζουν τους νευροδιαβιβαστές προς τις απολήξεις του νευράξονα.
 3. Οι απολήξεις εκπολώνονται και ανοίγουν τους ελεγχόμενους διαύλους ασβεστίου.
 4. Το ασβέστιο εισέρχεται μέσω των διαύλων
 5. Τα κυστίδια δέχονται την επίδραση του ασβεστίου το οποίο στην συνέχεια συγχωνεύεται με την προσυναπτική μεμβράνη
 6. Γίνεται η απελευθέρωση του νευροδιαβιβαστή μέσα στην συναπτική σχισμή
 7. Στον δεύτερο νευρώνα τα μόρια του υποδοχέα της μετασυναπτικής μεμβράνης αλληλεπιδρούν με τον νευροδιαβιβαστή
 8. Ανάλογα την περίπτωση έχουμε άνοιγμα ή κλείσιμο των διαύλων.
 9. Γίνεται η αποδέσμευση των μορίων του νευροδιαβιβαστή από τους υποδοχείς, ακολουθεί η δημιουργία ή μη του δυναμικού ενέργειας του μετασυναπτικού νευρώνα.
 10. Ο νευροδιαβιβαστής επαναπροσλαμβάνεται για ανακύκλωση και ταχεία μεταφορά των συναπτικών κυστιδίων ανάδρομα χωρίς την χρησιμοποίησή του.

Οι πιο σύνθετες γνωστικές λειτουργίες του ανθρώπινου εγκεφάλου όπως η μνήμη, η σκέψη η μάθηση και η αντίληψη οφείλονται στους μηχανισμούς της συναπτικής διαβίβασης των νευρώνων. Η σύνθεση και απελευθέρωση των νευροδιαβιβαστών γίνεται μέσω ενός πολύπλοκου και αρκετά σύνθετου συστήματος βιοχημικών μηχανισμών το οποίο προσαρμόζει άμεσα το νευρικό σύστημα στις γρήγορες περιβαλλοντικές μεταβολές. Το ηλεκτρικής μορφής σήμα από την στιγμή που φτάνει καταληκτικές συνάψεις του νευρώνα πυροδοτεί την απελευθέρωση των νευροδιαβιβαστών στις συνάψεις των νευρώνων. Οι χημικές ουσίες που ονομάζουμε νευροδιαβιβαστές στην συνέχεια θα προχωρήσουν προς την μετασυναπτική μεμβράνη του επόμενου νευρώνα οπότε και θα έχουμε την ηλεκτρική μεταβολή, που έχουμε περιγράψει και προηγουμένως, ικανή να διεγείρει ή να αναστείλει την μετάδοση του σήματος του μηνύματος στον επόμενο νευρώνα. (Kandel, 1999), (Purves & Andrews, 1997).

Παρακάτω θα περιγράψουμε την αλληλουχία της ηλεκτρικής μεταβίβασης των σημάτων μεταξύ των νευρωνικών συνάψεων όπως φαίνονται και στο σχήμα.

1. Στον πρώτο νευρώνα έχουμε την σύνθεση και την συγκέντρωση του νευροδιαβιβαστή στα συναπτικά κυστίδια. Από το σύνολο των ουσιών του νευροδιαβιβαστή τα λιπίδια, τις πρωτεΐνες κλπ. Έχουμε

- τον σχηματισμό των οργανιδίων και προώθηση τους στο κυτταρικό σώμα προς τον νευράξονα.
2. Το δυναμικό ενέργειας εισάγεται στις προσυναπτικές απολήξεις του νευρώνα και μεταβιβάζουν τους νευροδιαβιβαστές προς τις απολήξεις του νευράξονα.
 3. Οι απολήξεις εκπολώνονται και ανοίγουν τους ελεγχόμενους διαύλους ασβεστίου.
 4. Το ασβέστιο εισέρχεται μέσω των διαύλων
 5. Τα κυστίδια δέχονται την επίδραση του ασβεστίου το οποίο στην συνέχεια συγχωνεύεται με την προσυναπτική μεμβράνη
 6. Γίνεται η απελευθέρωση του νευροδιαβιβαστή μέσα στην συναπτική σχισμή
 7. Στον δεύτερο νευρώνα τα μόρια του υποδοχέα της μετασυναπτικής μεμβράνης αλληλεπιδρούν με τον νευροδιαβιβαστή
 8. Ανάλογα την περίπτωση έχουμε άνοιγμα ή κλείσιμο των διαύλων.
 9. Γίνεται η αποδέσμευση των μορίων του νευροδιαβιβαστή από τους υποδοχείς, ακολουθεί η δημιουργία ή μη του δυναμικού ενέργειας του μετασυναπτικού νευρώνα.
 10. Ο νευροδιαβιβαστής επαναπροσλαμβάνεται για ανακύκλιση και ταχεία μεταφορά των συναπτικών κυστιδίων ανάδρομα χωρίς την χρησιμοποίησή του.



Εικόνα 8. Οι φάσεις της ηλεκτροχημικής μεταβίβασης των σημάτων ανάμεσα στις νευρωνικές συνάψεις κατά (Purves & Andrews, 1997)

1.1.10 Βιολογική νευρωνική μνήμη και μάθηση

Όπως γνωρίζουμε ο αριθμός των νευρωνικών συνάψεων στο εσωτερικό του ανθρώπινου εγκεφάλου είναι τεράστιος υπολογίζεται σε περίπου σε 1014. Παραπάνω έχουμε προαναφέρει ότι οι συνάψεις για την μεταβίβαση του σήματος-πληροφορίας συντελούν τις λειτουργίες τις ηλεκτρικής και της χημικής μεταβίβασης κατά την διάρκεια της ενδονευρωνικής και διανευρωνικής επικοινωνίας αντίστοιχα (Kandel, 1999). Οι χημικοί νευροδιαβιβαστές έχουν διττή συμπεριφορά είτε δημιουργούν δυναμικό ενέργειας (κατάσταση διέγερσης) είτε λειτουργώντας κατασταλτικά. Σε επίπεδο νευρολογικής φύσεως η διαδικασία της επεξεργασίας πληροφοριών και των γνωστικών λειτουργιών του ανθρώπου δρουν ανάλογα με βάση τα αποτελέσματα διεγέρσεως-καταστολής που καθορίζεται από τον υποδοχέα του νευρώνα. Τέτοιοι υποδοχείς με δυνατότητα καθορισμού της συνοπτικής φύσης είναι: α) Οι διάλυτοι υποδοχείς N-Μέθυλο-D-Ασπαραγινικού οξέως β) Οι διάλυτοι υποδοχείς μη-(κισκαλικουμικαϊκού, Q/K), γλουταμινικού οξέως (μη NMDA). Ο ρόλος των δύο διαύλων που θα αναφέρουμε αναλυτικότερα παρακάτω είναι η κωδικοποίηση και η μακρόχρονη αποθήκευση πληροφοριών. Το γλουταμινικό οξύ αποτελεί υποκατηγορία των αμινοξέων που λειτουργούν ως νευροδιαβιβαστές (Kandel, 1999). Οι υποδοχείς γλουταμινικού οξέως χωρίζονται σε δύο μεγάλες κατηγορίες: α) Τους υποδοχείς άμεσου ελέγχου των διαύλων ιόντων. Τέτοιοι υποδοχείς είναι οι NMDA και οι μη NMDA. β) Τους υποδοχείς έμμεσου ελέγχου των διαύλων ιόντων. Μελετώντας την χημική μεταβίβαση καταλήγουμε στο συμπέρασμα ότι οι χημικές συνάψεις είναι πιο ευέλικτες από τις ηλεκτρικές, δυνατότητα που προκαλεί την παραγωγή πολυσύνθετων μορφών συμπεριφοράς εν συγκρίσει με τις ηλεκτρικές συνάψεις (Bailey & Kandel, 1993).

1.1.11 Χαρακτηριστικά φυσικών νευρωνικών δικτύων

Τα βασικά χαρακτηριστικά των φυσικών νευρωνικών δικτύων είναι τα ακόλουθα:

- Μη γραμμικότητα: Η έξοδος της πληροφορίας ενός φυσικού νευρωνικού δικτύου δεν αποτελεί και δεν απαιτεί τον γραμμικό συνδυασμό των εισόδων του δικτύου.
- Ικανότητα μάθησης: Είναι το χαρακτηριστικό που κάνει τον ανθρώπινο εγκέφαλο να μαθαίνει πράγματα. Τα νευρωνικά δίκτυα μπορούν να μάθουν πράγματα μέσα από εκπαίδευση. Η λειτουργία και η ακριβής δομή δεν έχει περιγραφεί ακόμα επιστημονικά.
- Προσαρμοστικότητα: Ο όρος προσαρμοστικότητα μπορεί να γίνει εύκολα κατανοητός αν μελετήσουμε δομή του αμφιβληστροειδούς χιτών. Ο αμφιβληστροειδής χιτών έχει την ικανότητα να διαστέλλεται και να συστέλλεται ανάλογα με τις διάφορες εναλλαγές του φωτός. Έτσι τα νευρωνικά δίκτυα μπορούν να προσαρμόσουν την συμπεριφορά τους κατάλληλα όταν στις εισόδους τους συμβαίνουν μεγάλης κλίμακας αλλαγές.

- **Παράλληλη επεξεργασία:** Αποτελεί θεμελιώδη κεφάλαιο της πληροφορικής αλλά και των νευρώνων. Είναι γνωστό ότι με τον όρο παράλληλη επεξεργασία θα αναφερθούμε στο πιο γνωστό χαρακτηριστικό της σημερινής κοινωνίας της πληροφορίας την ταχύτητα. Η ταχύτητα των επεξεργασιών αποτελεί το τελευταίο οχυρό που ξέρουμε βάσει ερευνών ότι θα ερευνάται ακόμα μέχρι το 2025. Έτσι λοιπόν η παράλληλη επεξεργασία δεδομένων του νευρωνικού δικτύου επιτυγχάνει απόκριση «δεδομένων εξόδου» σε πραγματικό χρόνο. Ο μέσος χρόνος απόκρισης του φυσικού κυττάρου είναι της τάξης των 10-3 δευτερολέπτων. Πρέπει να σημειωθεί εδώ πως ο χρόνος απόκρισης ενός σύγχρονου (συμβατικού) ολοκληρωμένου κυκλώματος είναι μικρότερος από 10-6 δευτερολέπτα.
- **Αντοχή στις διακοπές μεταξύ των συνδέσεων των νευρώνων:** η μελέτη των νευρώνων γενικότερα γίνεται με βάση την συμπεριφορά τους και τις ομοιότητες ή τις διαφορές που τους διακρίνουν σε σχέση με τα σημερινά υπολογιστικά συστήματα. Έτσι λοιπόν σε ένα πραγματικό δίκτυο η διακοπή κάποιων συνδέσεων μπορεί να προκαλέσει ανάλογα την δομή του δικτύου ολική ή μερική τμηματική καταστροφή. Τα νευρωνικά δίκτυα όμως δεν διαταράσσονται σημαντικά από τυχόν διακοπή συνδέσεων η ακόμα και στις σπάνιες περιπτώσεις αφαίρεσης νευρώνων. Απόδειξη αυτού του γεγονότος αποτελεί η τμηματική διακοπή της εγκεφαλικής λειτουργίας. Η διακοπή δηλαδή λειτουργικών τμημάτων του εγκεφάλου δεν επηρεάζει άλλες εγκεφαλικές λειτουργίες.
- **Ομοιότητα στον τρόπο λειτουργίας των νευρώνων:** Η κατανόηση και αυτού του χαρακτηριστικού γίνεται με ένα παράδειγμα δομικής φύσεως. Τα νευρικά κύτταρα του ίδιου τύπου έχουν πανομοιότυπη φυσιολογία και συμπεριφορά. Κάθε νευρώνας εκτελεί τις βασικές λειτουργίες με τους υπόλοιπους. Το ίδιο συμβαίνει και με τους υπολογιστές, οι υπολογιστές γνωρίζουν την δυαδική γλώσσα και λειτουργίες τους είναι τρεις πρόσθεση, μεταφορά και σύγκριση δεδομένων.

1.2 Τεχνητά Νευρωνικά Δίκτυα

1.2.1 Εισαγωγή

Ο Robert Hecht-Nielsen ήταν ένας από τους πρώτους που κατασκεύασαν νευρωνικούς υπολογιστές (νευρωνικό δίκτυο υπολογιστών) σε πρώιμη μορφή. Για τα τεχνητά νευρωνικά δίκτυα έδωσε έναν αρκετά σαφή ορισμό. «Ένα νευρωνικό δίκτυο είναι ένα υπολογιστικό σύστημα κατασκευασμένο από έναν αριθμό απλών ισχυρά διασυνδεδεμένων επεξεργαστικών στοιχείων, τα οποία επεξεργάζονται την πληροφορία με την δυναμική τους κατάσταση απόκρισης σε εξωτερικές εισόδους.

Κατά το έτος 1943 οι Warren McCulloch και Walter Pitts μελέτησαν την επιστήμη της λογικής και ασχολήθηκαν με τον τομέα της νευροφυσιολογίας. Στην προσπάθεια τους αυτή δημιούργησαν μια κατασκευή με ηλεκτρονικά εξαρτήματα η οποία μπορούσε να μιμηθεί όσα ήταν γνωστά έως τότε για τους νευρώνες του ανθρώπινου εγκεφάλου. Αυτή η έρευνα-κατασκευή αποτελεί την πρώτη σοβαρή ενασχόληση της επιστήμης με τους τεχνητούς νευρώνες. Την δεκαετία του '50 υπήρχε σοβαρή ανάπτυξη των ερευνών η οποία οδήγησε στην ικανότητα μάθησης των τεχνητών νευρωνικών δικτύων σε μικρό βαθμό.

Το perceptron ήταν ένα νευρωνικό δίκτυο που δημιουργήθηκε σε τελική μορφή το 1968 από τον Frank Rosenblatt ενώ είχε αρχίσει να παρουσιάζεται από το 1957 και χρησιμοποιήθηκε για την «αναγνώριση μορφών». Από τα χαρακτηριστικά του προσώπου μπορούσε να δώσει πληροφορία για το γένος του ανθρώπου, να αναγνωρίσει την εθνικότητα κατασκευής ενός ταγκ ή να εκφέρει άποψη για τον βαθμό πληρότητας μιας αποβάθρας τρένου. Λίγο πιο συγκεκριμένα το μοντέλο αυτό είχε δύο επίπεδα αυτό της εισόδου και αυτό της εξόδου. Το σήμα-πληροφορία μεταδιδόταν σε μονή κατεύθυνση από την είσοδο προς την έξοδο (Rosenblatt, 1962). Αρχικά υπήρξε η άποψη ότι το συγκεκριμένο μοντέλο μπορούσε να λύσει άλυτα μέχρι εκείνη την εποχή προβλήματα και έφερε κύμα ενδιαφέροντος και ενθουσιασμού στην επιστημονική κοινότητα. Το 1969 όμως οι Minsky και Papert στο βιβλίο τους Perceptrons παρουσίασαν μια συνολική και ολοκληρωμένη άποψη σχετικά με την χρησιμότητα του συγκεκριμένου προτύπου (Minsky & Papert, 1969). Απέδειξαν με αναλυτικά μαθηματικά ότι το πρότυπο έχει αρκετούς περιορισμούς. Απλά προβλήματα όπως αυτό της λογικής πύλης X-OR δεν μπορούσαν να λυθούν. Η επιστημονική κοινότητα άρχισε να στρέφεται σε μια νέα επιστήμη που άρχισε να εμφανίζεται αυτή της Τεχνητή Νοημοσύνης. Τα νευρωνικά δίκτυα πλέον θα αναπτύσσονταν από τους επιστήμονες αλλά με την πεποίθηση ότι τα προβλήματα που θα έλυαν θα ήταν ειδικά. Παράλληλα με το perceptron το πανεπιστήμιο του Στάνφορντ κατασκεύασε με την χρήση απλών αναλογικών συσκευών το σύστημα ADAPTIVE LINEAR ELEMENT γνωστό και ως ADALINE. Όπως ήταν φυσικό την δεκαετία '80 με την ραγδαία ανάπτυξη των υπολογιστικών συστημάτων και της επιστήμης της πληροφορικής γενικά αλλά και την χρήση ολοκληρωμένων κυκλωμάτων με μεγάλη κλίμακα ολοκλήρωσης και την παράλληλη επεξεργασία πιο συγκεκριμένα αναπτύχθηκε με γοργά βήματα και η έρευνα των τεχνητών νευρωνικών δικτύων. Τα νευρωνικά δίκτυα χρησιμοποιούνται από τότε στην επεξεργασία εικόνας και ομιλίας, στην μηχανική όραση, στην ρομποτική, σε συστήματα αναγνώρισης και ελέγχου κλπ.

Το νευρωνικό δίκτυο είναι μια συστοιχία ενός αρκετά μεγάλου αριθμού υπολογιστικών μονάδων απλουστευμένης λειτουργίας και περιορισμένων σχετικών γνώσεων, τους νευρώνες. Οι νευρώνες συνδέονται μεταξύ τους με κανάλια επικοινωνίας. Η επικοινωνία μεταξύ τους είναι μονοδιάστατη καθώς μεταφέρουν αριθμητικά δεδομένα προς μια μόνο κατεύθυνση. Στον τομέα της επεξεργασίας δεδομένων, τα δεδομένα επεξεργάζονται μόνο σε τοπική βάση.

Τα τεχνητά νευρωνικά δίκτυα και τα φυσικά νευρωνικά δίκτυα παρουσιάζουν κοινές ιδιότητες:

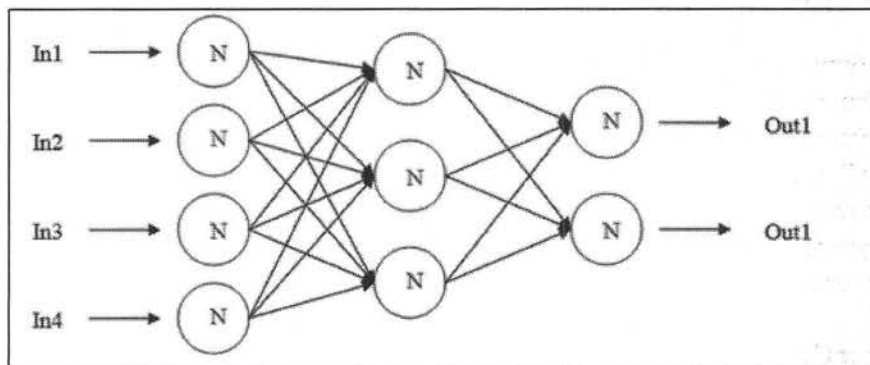
- Για την απόκτηση γνώσης απαιτείται μια διαδικασία μάθησης
- Οι διασυνδέσεις μεταξύ των νερώνουν χρησιμεύουν για την αποθήκευση των πληροφοριών και της γνώσης
- Μπορούν να αξιοποιήσουν την γνώση μνήμης. Δηλαδή μέσω της εμπειρίας.

1.2.2 Αρχιτεκτονική νευρωνικών δικτύων

Τα νευρωνικά δίκτυα έχουν μια πολλή ιδιότυπη αρχιτεκτονική καθώς δεν μπορούμε να τα θεωρήσουμε δίκτυα εξυπηρέτησης πολλών χρηστών με όλα τα πλεονεκτήματα ενός τοπικού δικτύου υπολογιστών. Η ιδιότυπη αυτή αρχιτεκτονική τους παρουσιάζει μειονέκτημα στην εκτέλεση μεγάλων όγκων μαθηματικών υπολογισμών.

Ο ανθρώπινος εγκέφαλος χρησιμοποιεί τον μαζικό παραλληλισμό ή την παράλληλη κατανομημένη επεξεργασία για διαλογή και την επεξεργασία των δεδομένων-πληροφοριών. Τα τεχνητά νευρωνικά δίκτυα ακολουθούν την συγκεκριμένη τάση στο πλαίσιο μιμητισμού του ανθρώπινου εγκεφάλου. Έτσι ένα νευρωνικό δίκτυο αποτελείται από τους νευρώνες οι οποίοι είναι διατεταγμένοι ώστε τμηματικά πλήθος εξ αυτών να μπορούν να επεξεργαστούν τις πληροφορίες σε ταυτόχρονη βάση. Η κατάσταση του κάθε νευρώνα ξεχωριστά εξαρτάται από τα σήματα που δέχεται από τους νευρώνες που βρίσκεται σε διασύνδεση.

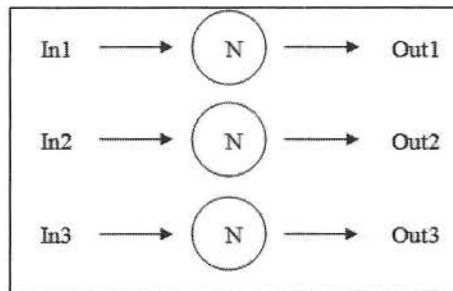
Η υλοποίηση των τεχνητών νευρωνικών δικτύων μπορεί να γίνει με δυο μεθόδους: α) Εικονικά μέσω προσομοίωσης κατάλληλου λογισμικού και β) Υλικοτεχνικά. Η υλικοτεχνική υλοποίηση του δικτύου έχει αρκετά μεγάλο κόστος αλλά έχει μεγαλύτερη ταχύτητα από την εικονική προσομοίωση με λογισμικό. Τα περισσότερα νευρωνικά δίκτυα γίνονται με την προσομοίωση του κατάλληλου λογισμικού. Ο κάθε νευρώνας αποτελεί μια υπορουτίνα ενός κυρίου προγράμματος.



Εικόνα 9. Τεχνητό νευρωνικό δίκτυο {N = νευρώνας, In = είσοδοι, Out = έξοδοι}

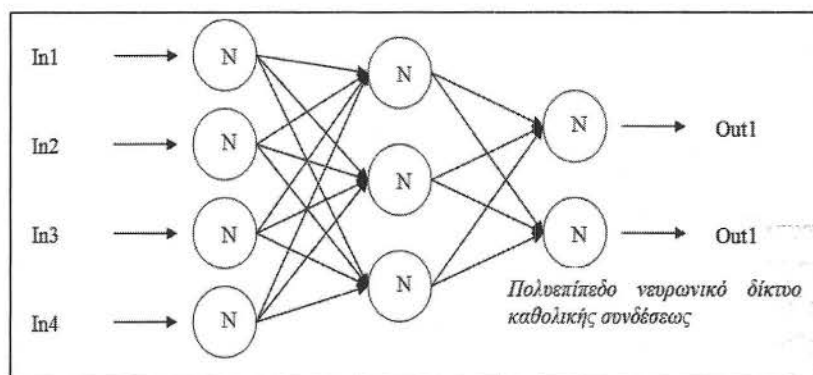
Ο τρόπος διασύνδεσης και οργάνωση των νευρώνων διαχωρίζει τα τεχνητά νευρωνικά δίκτυα σε:

- Νευρωνικά δίκτυα ενός επιπέδου: Είναι η πλέον απλή μορφή οργάνωσης των τεχνητών νευρωνικών δικτύων. Κατά αυτή την μορφή οργάνωσης, οι εισοδοί του κάθε νευρώνα συνδέονται με τις εισόδους του δικτύου και οι έξοδοι κάθε νευρώνα αποτελούν μαζί τις εξόδους του κάθε δικτύου. Είναι δηλαδή μια παράλληλη σύνδεση πολλών νευρώνων μαζί. Η συγκεκριμένη διάταξη χρησιμοποιείται για να επιλύσουμε απλά προβλήματα.



Εικόνα 10. Τεχνητά νευρωνικά δίκτυα ενός επιπέδου

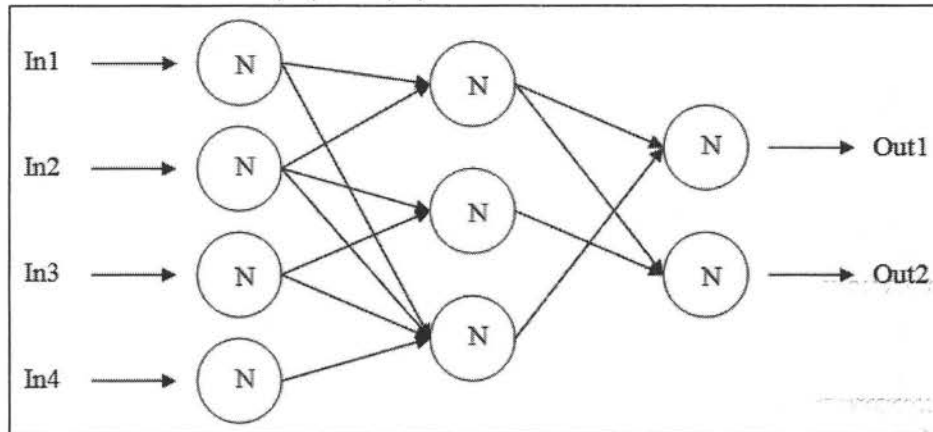
- Πολυεπίπεδα νευρωνικά δίκτυα: Νευρωνικά δίκτυα δύο ή περισσότερων επιπέδων. Με την προσθήκη ενός ή περισσότερων επιπέδων πετυχαίνουμε να αυξήσουμε βαθμούς ελευθερίας του δικτύου. Οι νευρώνες ενός επιπέδου μπορούν να συνδεθούν με τις εξόδους των νευρώνων ενός άλλου επιπέδου. Η διασύνδεση των νευρώνων διαχωρίζονται σε δύο κατηγορίες. Την καθολική σύνδεση όπου οι νευρώνες του ενός επιπέδου συνδέονται με όλους τους νευρώνες ενός άλλου επιπέδου και την μερική σύνδεση όπου



Εικόνα 11. Τεχνητό Νευρωνικό Δίκτυο μερικής συνδέσεως πολλών επιπέδων

ο νευρώνας ενός επιπέδου συνδέεται με μερικούς νευρώνες ενός άλλου επιπέδου.

- Επανατροφοδοτούμενα νευρωνικά δίκτυα: Στην συγκεκριμένη μορφή η διαδρομή διασυνδέσεως ενός νευρώνα μπορεί να ακολουθήσει την φορά ενεργοποίησης ενός δικτύου και να επιστρέψει πάλι σε αυτόν. Η διαδικασία αυτή καλείται επανατροφοδότηση.



Εικόνα 12. Τεχνητό Νευρωνικό Δίκτυο καθολικής συνδέσεως πολλών επιπέδων

1.2.3 Χαρακτηριστικά Τεχνητών Νευρωνικών Δικτύων

Τα τεχνητά νευρωνικά δίκτυα διακρίνονται από τα εξής πλεονεκτήματα:

1. Διεκπεραίωση λειτουργιών σε σύντομο χρόνο
2. Η εξαγωγή γενικών συμπερασμάτων
3. Η πολλαπλότητα των προβλημάτων που μπορούν να δώσουν λύση
4. Η μεγάλη ταχύτητα επεξεργασίας μέσω της λειτουργίας σε παράλληλα υπολογιστικά συστήματα
5. Η λειτουργία χωρίς προγραμματισμό
6. Η ικανότητα μάθησης
7. Η ανεκτικότητα σε σφάλματα

Τα τεχνητά νευρωνικά δίκτυα καθορίζονται σε σημαντικό βαθμό:

- Από τις ιδιότητες των νευρώνων
- Από τις ιδιότητες του δικτύου
- Από την δυναμική του συστήματος

Κύρια χαρακτηριστικά των νευρώνων είναι η κλίμακα δραστηριοποίησης και οι λειτουργίες μεταβίβασης. Ιδιότητες του δικτύου αποτελούν οι τύποι και η διάταξη

των συνδέσεων, η τοπολογία του δικτύου και η κλίμακα επίδρασης. Ενώ στον τομέα της δυναμικής του συστήματος συμπεριλαμβάνονται ο κανόνας μάθησης ο τύπος ενεργοποίησης-υπολογισμών και επίδραση του σχήματος αρχικοποίησης.

1.2.4 Η ικανότητα μνήμης

Αρχικά κατά την ανάπτυξη των νευρωνικών δικτύων διερευνήθηκε η δυνατότητα να αποτελέσουν μονάδες αποθήκευσης. Η ιδέα ήταν να γίνει η αποθήκευση στα βάρη των συνδέσεων w με τέτοιο τρόπο ώστε να περιέχουν συγκεκριμένες τιμές. Το μεγάλο ζήτημα θα ήταν αν ο αριθμός των bits για την αποθήκευση μιας πληροφορίας στα w του αισθητήρα θα ήταν μικρότερος από αυτόν της μνήμης ενός ηλεκτρονικού υπολογιστή. Οι Minsky και Papert έδειξαν με το βιβλίο σταθμό τους (όπου κατεύνασε τον μεγάλο ενθουσιασμό της επιστημονικής κοινότητας για τα νευρωνικά δίκτυα) ότι ο αριθμός των bits αυξάνεται με ρυθμό μεγαλύτερο από την εκθετική αύξηση. Ως αποτέλεσμα είναι η δέσμευση μεγάλης μνήμης (Minsky & Papert, 1969). Το μοντέλο του αισθητήρα απεδείχθη ότι είναι πιο επικερδές για προβλήματα μικρού μεγέθους. Αξίζει να σημειωθεί ότι ποσοτική σχέση μεταξύ των νευρωνικών δικτύων και των παραμέτρων που προαναφέρθηκαν δεν έχει μελετηθεί.

1.2.5 Η μετάδοση σήματος στο νευρωνικό δίκτυο – Συνάρτηση κατωφλίου Heaviside

Η μετάδοση του σήματος γίνεται κατά δυαδικό τρόπο. Έτσι ένας νευρώνας μπορεί να βρεθεί σε δύο δυνατές καταστάσεις: είτε να είναι ενεργός είτε αδρανής. Ο νευρώνας αφού δεχτεί τα σήματα εισόδου από άλλους νευρώνες υπολογίζει το σύνολο τους όπως περιγράψαμε προηγουμένως που συμβολίζεται με X αυτή την τιμή την συγκρίνει με μία τιμή κατωφλίου θ . Η τιμή αυτή είναι μια σταθερή μεταβλητή ορισμένη εξ' αρχής για τον συγκεκριμένο νευρώνα (Widrow & Lehr, 1990), (Rosenblatt, 1958).

Υπάρχουν δύο περιπτώσεις:

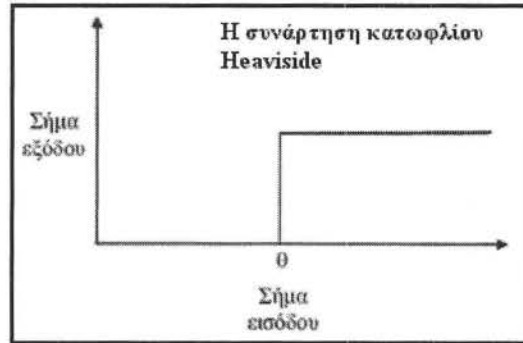
- Η τιμή της μεταβλητής X να είναι μεγαλύτερη ή ίση από την ορισμένη τιμή κατωφλίου. Σε αυτή την περίπτωση έχουμε ενεργοποίηση του νευρώνα.
- Η τιμή της μεταβλητής X να είναι μικρότερη από την τιμή κατωφλίου θ . Τότε ο νευρώνας βρίσκεται σε αδρανή κατάσταση.

Η λειτουργία όπως περιγράφηκε είναι δυαδική οπότε σε κατάσταση ενεργοποίησης η έξοδος f του νευρώνα έχει τιμή 1 ενώ σε κατάσταση αδράνειας έχει τιμή 0.

Ισχύει λοιπόν:

$$f(x) = \begin{cases} 1 & \text{εαν } X > \theta \text{ ή } X = \theta \\ 0 & \text{εαν } X < \theta \end{cases}, \quad \text{Εξίσωση 1}$$

Η παραπάνω συνάρτηση είναι η συνάρτηση εξόδου. Η λογική της παραπάνω συναρτήσεως στηρίζεται στην συνάρτηση κατωφλίου που εισήγαγε ο Heaviside. Το γράφημα της οποίας φαίνεται παρακάτω.



Εικόνα 11. Η συνάρτηση κατά Heaviside

Η ενεργοποίηση ή αδράνεια του νευρώνα-αισθητήρα είναι πλήρως σε εξάρτηση με τις τρεις παραμέτρους που έχουν ήδη αναφερθεί:

- Τα βάρη των συνδέσεων w_i .
- Τις τιμές των εισόδων x_i .
- Την σταθερή τιμή κατωφλίου θ .

Η προσωρινή μάθηση του αισθητήρα αποθηκεύεται στα βάρη w_i των συνδέσεων τα οποία μεταβάλλονται συνεχώς μέχρι ο νευρώνας να «μάθει» μία πληροφορία.

1.2.6 Πρότυπα Νευρωνικών Δικτύων και περιγραφή εκπαίδευσής τους

Στον παρακάτω πίνακα (Πίνακας 1) μπορούμε να δούμε την εξέλιξη των νευρωνικών δικτύων απο την δεκαετία του '50 μέχρι τα μέσα της δεκαετίας του '80 όπου και αναπτύχθηκαν τα βασικά μοντέλα τεχνητών νευρωνικών δικτύων.

Αρκετές παραλλαγές αναπτύχθηκαν στην συνέχεια που όμως ήταν βασισμένες είτε σε κάποιο από αυτά είτε σε συνδυασμούς των ανωτέρω δικτύων. Στην παρούσα εργασία θα γίνει μια εισήγηση στα πιο βασικά πρότυπα.

1.2.7 Το πρώιμο μοντέλο των McCulloch και Pitts

Η πρώτη εισήγηση στα μοντέλα των νευρωνικών δικτύων έγινε το 1943 από τους McCulloch και Pitts. Με την εργασία τους έδειξαν την δομο-λειτουργία ενός

νευρωνικού δικτύου παρουσιάζοντας τους νευρώνες με την διασυνδέσεις τους θεωρώντας ότι το νευρónico δίκτυο είναι ανάλογο πρότυπο ηλεκτρικού κυκλώματος. Το 1947 βελτίωσαν το πρότυπο τους κάνοντας το ικανό για την αναγνώριση σχημάτων. Στοιχείο που θυμίζει αρκετά μεταγενέστερα πρότυπα (McCulloch & Pitts, 1947).

Πίνακας 1. Χρονολογικός πίνακας νευρωνικών δικτύων

ΠΡΟΤΥΠΟ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ	ΕΤΟΣ ΔΗΜΙΟΥΡΓΙΑΣ	ΔΗΜΙΟΥΡΓΟΣ
Αισθητήρας	1957	Rosenblatt
Πρότυπο Adaline–Madaline	1960	Widrow
Πρότυπο χιονοστιβάδας	1967	Grossberg
Παρεγκεφαλιδικό πρότυπο	1969-1982	Marr, Albus, Pellionez
Πρότυπο προσαρμοσμένου συντονισμού	1978-1986	Carpeenter–Grossberg
Πρότυπο εγκεφάλου-σε-κουτί	1977	Anderson
Πρότυπο νευρωνώστη	1978-1984	Fukushima
Πρότυπο Kohonen	1980	Kohonen
Πρότυπο Hopfield	1982	Hopfield
Στατιστικά δίκτυα	1985	Hinton
Πρότυπο συνειρμικής μνήμης διπλής κατευθύνσεως	1985	Kosko
Μοντέλο οπισθοδιάδοσης	1985	Rumelhart
Πρότυπο αντίθετης διάδοσης	1986	Hecht–Nielsen

Τα στοιχεία που διέπουν τα δίκτυα McCulloch και Pitts είναι τα εξής:

- Ένας νευρώνας μπορεί να έχει μόνο δύο καταστάσεις ΑΛΗΘΗΣ και ΨΕΥΔΗΣ, πολλές εισόδους αλλά μόνο μια έξοδο. Με τον όρο αληθής εννοούμε ότι ο νευρώνας πυροδοτεί στέλνοντας ένα παλμό ενώ με τον όρο ψευδής εννοούμε ότι ο νευρώνας βρίσκεται σε ηρεμία.
- Δύο ή περισσότερες εξοδοί δεν μπορούν να ενώνονται αλλά να αποτελούν υποχρεωτικά είσοδο ενός άλλου νευρώνα.
- Οι νευρωνικές απολήξεις μπορεί να είναι είτε διεγερτικές είτε ανασταλτικές.
- Η πληροφορία ρέει μέσα στον νευρώνα μέσω των πυλών όπου και αυτές είναι δύο ειδών διεγερτικές και ανασταλτικές.
- Η κάθε λειτουργία γίνεται σε διακριτό χρόνο ενώ η δράση του συστήματος είναι συγχρονισμένη καθότι θεωρούμε ότι όλοι οι νευρώνες έχουν ταυτόχρονη απόκριση.
- Η επόμενη χρονικά κατάσταση ενός νευρώνα εξαρτάται από την παρούσα κατάσταση σε συνδυασμό με τις εισόδους που εισέρχονται στην χρονική αυτή στιγμή.

Είναι η πρώτη φορά όπου γίνεται προσπάθεια να εξηγηθεί η λειτουργία της μνήμης. Γίνεται αναφορά σε ένα πιθανό μηχανισμό μνήμης ο οποίος θα μπορούσε να λειτουργεί ως εξής, θεωρείται ότι στο δίκτυο υπάρχουν κλειστές διαδρομές ενός σήματος, αν δεν υπάρχει τέτοια διαδρομή αλλά και νέο εξερχόμενο σήμα τότε το δίκτυο θα βρίσκεται μονίμως σε κατάσταση ηρεμίας. Δημιουργείται ένας μηχανισμός ανάδρασης καθώς μία ίνα ενώνει την είσοδο και την έξοδο ενός κυττάρου. Έτσι μόλις ένα τέτοιου είδους κύτταρο αρχίσει και πυροδοτεί θα συνεχίσει μέχρις ότου διακοπεί από την ανασταλτική ίνα. Κατά την διάρκεια της όλης λειτουργίας στην πύλη των κυττάρων αποστέλλονται παλμοί και με αυτό τον τρόπο μεταδίδεται το σήμα και η πληροφορία. Το σύνολο της κυκλικής ροής του σήματος θα μπορούσε να θεωρηθεί ως η πρωταρχική ιδέα που υπήρξε από τους δύο ερευνητές για ένα μηχανισμό μνήμης.

1.2.8 Ο κανόνας μάθησης του Hebb

Το 1949 ο Hebb παρουσιάζει ένα διαχρονικής επιρροής βιβλίο με τίτλο «Ο οργανισμός συμπεριφοράς» όπου και εισάγεται ο ομώνυμος κανόνας μάθησης (Hebb, 1949). Το κατά Hebb μοντέλο δίνει μεγαλύτερη βάση στους νευρώνες και τις συνδέσεις μεταξύ των μονάδων του συστήματος. Κάθε φορά που το δίκτυο χρησιμοποιεί τις νευρωνικές του συνδέσεις αυτές παρουσιάζουν ενίσχυση και το δίκτυο τείνει όλο και περισσότερο να μπορέσει να αφομοιώσει δηλαδή να μάθει το πρότυπο το οποίο παρουσιάζεται. Όταν ένας νευρώνας έστω i διεγείρει συνεχώς έναν άλλο νευρώνα έστω j διαμορφώνεται μια μεταβολική σύνδεση σε ένα εκ των δύο ή

και στους δύο νευρώνες με τέτοιο τρόπο ώστε η απόδοση του φαινομένου να αυξάνεται. Αν w_{i-j} αποτελεί το βάρος της σύνδεσης μεταξύ i και j , και x_i αποτελεί την είσοδο στον νευρώνα j από τον νευρώνα i , ενώ x_j η έξοδος του νευρώνα j τότε ισχύει:

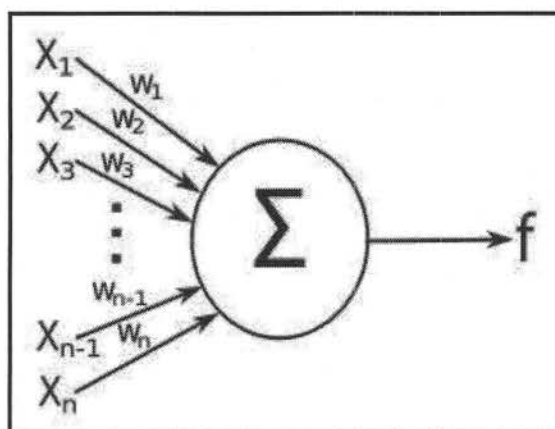
$$w_{ij}(n+1) = w_{ij}(n) + ax_ix_j, \quad \text{Εξίσωση 2}$$

Όπου a θετική σταθερά που ονομάζεται παράμετρος ρυθμού εκπαίδευσης. Το βάρος w_{ij} της νέας κατάστασης θα είναι ίσος με το βάρος της παρούσας κατάστασης προσαυξημένο κατά μία ποσότητα ax_ix_j . Ο κανόνας ισχύει σε τοπικό επίπεδο μεταξύ i και j και όχι για τις υπόλοιπες συνδέσεις του δικτύου.

1.2.9 Το πρότυπο του αισθητήρα (Perceptron)

Στο παρόν κεφάλαιο θα γίνει αναλυτική παρουσίαση του μοντέλου του αισθητήρα. Το μοντέλο εισήχθη από τον Rosenblatt και επήρασε τα επόμενα νευρωνικά δίκτυα που πολλά από αυτά στήριξαν την δομή τους στο συγκεκριμένο πρότυπο (Rosenblatt, 1958), (Rosenblatt, 1962).

Η απλούστερη μορφή είναι αυτή του στοιχειώδους αισθητήρα που αποτελείται ένα μοναδικό νευρώνα και θεωρείται το πιο απλό μη-εξαρτώμενο σύστημα που υπάρχει για να πραγματοποιήσει μια συγκεκριμένη διεργασία. Ο νευρώνας αυτός έχει ως είσοδο ένα σύνολο από συνδέσεις που προέρχονται από άλλους νευρώνες και μία μοναδική έξοδο. Ο νευρώνας αναπαρίσταται με ένα κύκλο και οι συνδέσεις με γραμμές. Οι είσοδοι συμβολίζονται με X_1, X_2, \dots, X_n .



Εικόνα 12. Το μοντέλο του αισθητήρα

Όπως βλέπουμε δεξιά του κύκλου υπάρχει μόνο μία έξοδος, ένας νευρώνας δύναται να έχει πολλές εξόδους που όμως όλες αυτές οι εξοδοί θα έχουν πάντα την ίδια τιμή. Αυτός είναι και ο λόγος που παρόλο που ο νευρώνας μπορεί να έχει πολλές εξόδους λέμε ότι έχει μόνο μία έξοδο. Ένα οποιοδήποτε σήμα εισόδου X_i συνδέεται

με τον νευρώνα με ένα βάρος w_i . Το βάρος w αποτελεί την αλληλεπίδραση μεταξύ των δύο συνδεδεμένων νευρώνων. Τόσο η τιμή του βάρους αλλά και η τιμή του σήματος εισόδου δεν έχουν ανεξάρτητη δράση αλλά συνδέονται και μελετώνται μαζί. Ιδιαίτερα αυτό που μας ενδιαφέρει είναι το γινόμενο $x_i w_i$. Ο νευρώνας αθροίζει το σύνολο των γινομένων εισόδων για n όρους και λαμβάνει ένα συνολικό σήμα που έχει την ακόλουθη τιμή:

$$X = \sum_{i=0}^n x_i w_i, \quad \text{Εξίσωση 3}$$

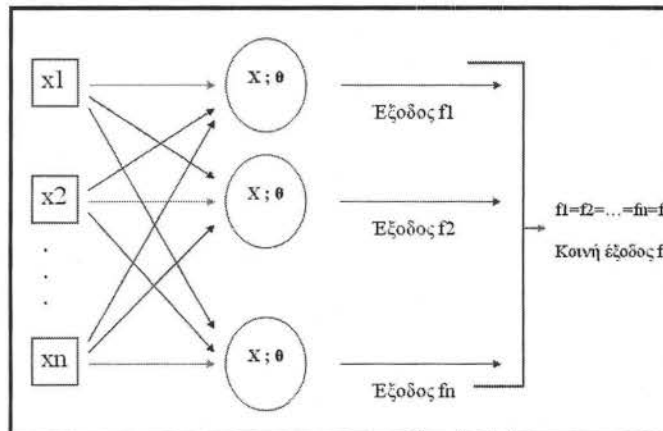
Αυτή δεν είναι η τελική τιμή καθώς ο νευρώνας έχει και ένα δικό του εσωτερικό βάρος το οποίο ονομάζεται bias και συμβολίζεται με b . Αυτό το εσωτερικό βάρος ονομάζεται συχνά και παράγων προδιάθεσης του νευρώνα και είναι ανεξάρτητο από τα άλλα βάρη που όμως η δράση του είναι η ίδια με τα βάρη εισόδου. Η τιμή του σήματος που διαπερνά όλα τα εσωτερικά βάρη είναι ίσο με 1. Συνεπώς οι μονάδες του b θα είναι ίδιες με αυτές του γινομένου ($s \cdot w$).

Η νέα γενική μορφή της εξίσωσης είναι:

$$X = \sum_{i=0}^n x_i w_i + b, \quad \text{Εξίσωση 4}$$

Στην επιστημονική κοινότητα υπάρχουν γνώμες που υποστηρίζουν ότι το bias θεωρείται εξωτερικό ερέθισμα και δεν έχει καμία φυσική σημασία με το εσωτερικό του νευρώνα, μάλιστα ένα μέρος των προβλημάτων λύνεται με την προηγούμενη εξίσωση που δεν περιείχε το b καθώς κρίνεται σκόπιμο να μην λαμβάνεται υπόψη.

Το μοντέλο του αισθητήρα δύναται να αποτελείται από n αριθμό νευρώνων αποτελώντας έτσι μια πιο προχωρημένη μορφή του αρχικού σχήματος του αισθητήρα που παρουσιάστηκε (Haykin, 1999). Μία τέτοια μορφή βλέπουμε παρακάτω:



Εικόνα 13. Πολυεπίπεδο πρότυπο αισθητήρα

Οι είσοδοι που αναπαρίσταται με τετράγωνο είναι γνωστό ότι δεν λαμβάνουν μέρος στην επεξεργασία του σήματος και η χρησιμότητα τους έγκειται στην υποδοχή του σήματος από άλλους νευρώνες. Με κύκλο αναπαρίσταται οι νευρώνες όπου γίνεται η επεξεργασία της πληροφορίας και η διαδικασία ελέγχου για την λογική σχέση μεταξύ χ και θ .

Παραπάνω φαίνεται το n πλήθος των νευρώνων (ο στοιχειώδης αισθητήρας είχε μόνο ένα νευρώνα). Σε κατάσταση πλήρους συνδεσμολογίας κάθε σήμα εισόδου χ παρουσιάζει στους n – νευρώνες διαφορετική τιμή βάρους w_i της συνδέσεως. Το σύστημα της συγκρίσεως των εισόδων με την σταθερή τιμή θ είναι ακριβώς το ίδιο με αυτό του στοιχειώδους αισθητήρα. Η κύρια διαφορά είναι το n - πλήθος των νευρώνων που κάθε ένας από αυτούς μας δίνει έξοδο f_i σύνολο δηλαδή n – εξόδους που όμως όλες θα έχουν την ίδια τιμή. Στην επιστήμη των νευρωνικών δικτύων υπάρχουν διάφορα πιο περίπλοκης μορφής αλλά παρόμοια μοντέλα που η λειτουργία τους είναι η ίδια που περιγράφηκε προηγουμένως.

1.2.10 Περιγραφή εκπαίδευσης ενός τεχνητού νευρωνικού δικτύου

Όπως έχει προαναφερθεί η ικανότητα μάθησης αποτελεί ένα από τα πλεονεκτήματα των τεχνητών νευρωνικών δικτύων και ίσως το σημαντικότερο. Με την έννοια μάθηση στα τεχνητά νευρωνικά δίκτυα εννοούμε τις διαδικασίες μέσα από τις οποίες το δίκτυο προγραμματίζεται μέσω ορισμένων μεταβλητών να ελέγχει τις επιδόσεις του συστήματος, δηλαδή κατά πόσο η έξοδος του δικτύου συμπίπτει με το επιθυμητό αποτέλεσμα.

Υπάρχουν δύο μέθοδοι μάθησης-εκπαίδευσης του δικτύου:

- Η μάθηση με εποπτεία (supervised learning): Ένας ειδικός αξιολογεί την επίδοση τους συστήματος, συγκρίνει δηλαδή το τελικό αποτέλεσμα με το επιθυμητό αποτέλεσμα μέσα από μια επαναληπτική διαδικασία. Μέσω των συγκρίσεων αυτών δημιουργούνται επαναλήψεις, με τροποποιημένα δεδομένα εισόδου ώστε το αποτέλεσμα εξόδου να ταιριάζει με το ηθελημένο αποτέλεσμα. Έτσι πριν μπορέσει το δίκτυο να λειτουργήσει εκπαιδεύεται-προγραμματίζεται.
- Η αυτοαξιολόγηση του δικτύου χωρίς εποπτεία: Ουσιαστικά είναι λίγο αυθαίρετο και ουτοπικό να χρησιμοποιούμε την φράση «χωρίς εποπτεία» καθώς τα πάντα ελέγχονται από τους ανθρώπους. Στην συγκεκριμένη περίπτωση το δίκτυο στήνεται και προγραμματίζεται αρχικά έτσι ώστε να υπάρχουν κάποιοι παράμετροι και κριτήρια που θα χρησιμοποιηθούν για να αυτό-αξιολογήσει το δίκτυο την ορθότητα και την σχέση του τελικού αποτελέσματος με το επιθυμητό. Η αυτοαξιολόγηση έχει την δυνατότητα να στηρίζεται στον ανταγωνισμό μεταξύ συγκεκριμένων ομάδων νευρώνων που έχουν σημείο αναφοράς την εκπαίδευση.

Είναι εύκολα κατανοητό ότι η πολυπλοκότητα του δικτύου και η αρχιτεκτονική επηρεάζουν την ταχύτητα εκπαίδευσης-προγραμματισμού του δικτύου. Όπως επίσης η μέθοδος μάθησης και το ποσοστό ταύτισης του επιθυμητού με το παραγόμενο αποτέλεσμα.

Μερικοί από τους μαθησιακούς κανόνες που χρησιμοποιούνται είναι οι εξής: Kohonen's Learning Law, the Delta Rule, the Last-Mean Square Rule, Hebb's Rule, Gradient Descent Rule και μερικές ακολουθίες πιθανοτήτων και τυχαιών μεταβλητών.

1.2.11 Η εκπαίδευση του αισθητήρα

Προηγουμένως αναφερθήκαμε στο γενικότερο πλαίσιο της εκπαίδευσης των νευρωνικών δικτύων. Παρακάτω θα αναλυθεί πως γίνεται η εκπαίδευση του αισθητήρα. Η ιδιότητα των νευρωνικών δικτύων να μπορούν να επιτελούν μια διαδικασία ξανά και ξανά αν εκπαιδευτούν είναι ίσως το πιο σημαντικό χαρακτηριστικό τους. Σε ομοιότητα με τα βιολογικά νευρωνικά δίκτυα μέσω της γνωστικής εμπειρίας που αποκτούν ύστερα από την εκπαίδευση δύναται να επιλύσουν ένα πρόβλημα βγάζοντας στην έξοδο το επιθυμητό σωστό αποτέλεσμα.

Η εκπαίδευση ενός νευρωνικού δικτύου έγκειται στο να βρούμε τις τιμές των w και θ . Η καταλληλότητα των οποίων θα δώσει την δυνατότητα στο νευρωνικό δίκτυο να αναγνωρίζει τα πρότυπα-μεθόδους που του έχουμε διδάξει. Η εκπαίδευση του αισθητήρα είναι η μέθοδος εκπαίδευσης με εποπτεία. Ο αλγόριθμος της εκπαίδευσης κατασκευάζεται μέσω προγράμματος στον υπολογιστή (Rosenblatt, 1958).

Η διαδικασία που τηρούμε είναι η ακόλουθη: Αρχικά το νευρωνικό δίκτυο δεν έχει γνώση. Έτσι οι τιμές των βαρών πρέπει να είναι τυχαίες. Οι τιμές τους δίνονται μέσω κατανομής ψευδοτυχαίων αριθμών και αριθμητικά κυμαίνονται $0 < w < 1$. Το σύστημα μαθαίνει το πρότυπο που του παρουσιάζουμε κάθε φορά μεταβάλλοντας τα βάρη του προς την σωστή κατεύθυνση. Καθ' ομοίωση με το βιολογικό δίκτυο που εξασκείται για την μάθηση μιας διεργασίας. Αυτή η συνεχής μεταβολή της τιμής των βαρών θα σταματήσει όταν το δίκτυο μάθει το σήμα που του δόθηκε. Τότε οι τιμές των βαρών είναι οι τελικές και αποθηκεύονται. Το δίκτυο πλέον έχει εκπαιδευτεί και έχει μάθει το πρότυπο λύσεως που του διδάξαμε. Η συγκεκριμένη τεχνική είναι σχεδόν η ίδια σε όλα τα νευρωνικά δίκτυα η διαφορά στα άλλα πρότυπα βρίσκεται στην διαδικασία αλλαγής της τιμής w .

Τα πρότυπα-παραδείγματα ονομάζονται patterns και για την εκπαίδευση του νευρωνικού δικτύου απαιτείται ένα σύνολο από παραδείγματα. Επίσης στο σύνολο των σημάτων έχουμε αντιστοιχία μεταξύ των εισόδων και των εξόδων. Κάθε σύνολο αποτελείται από δύο τμήματα:

- Το σύνολο των σημάτων εισόδου s (s_1, s_2, \dots, s_n) που παρουσιάζονται στην είσοδο του νευρωνικού δικτύου
- Το δεύτερο τμήμα είναι τα σήματα εξόδου που έχουν τις επιθυμητές τιμές στόχους που είναι το επιθυμητό αποτέλεσμα μας. Για κάθε ομάδα σημάτων εισόδου αντιστοιχεί μόνο μία σωστή-τιμή στόχος.

1.2.12 Το πρότυπο της οπισθοδιάδοσης του λάθους (Error-Backpropagation)

Η μέθοδος που θα αναπτύξουμε σε αυτό το κεφάλαιο είναι η πιο διαδεδομένη μέθοδος για να εκπαιδύσουμε ένα νευρωνικό δίκτυο. Η εκπαίδευση των νευρωνικών δικτύων πριν από αυτή την τακτική περιορίζονταν σε δίκτυα δύο επιπέδων κατά μέγιστη τιμή. Σκοπός της επιστημονικής κοινότητας ήταν η αναπαράσταση με την χρήση των νευρωνικών δικτύων πιο περίπλοκων προβλημάτων και εφαρμοσμένων διεργασιών. Η τεχνική της οπισθοδιάδοσης του λάθους κατάφερε να λύσει το γνωστό πρόβλημα με τις XOR το οποίο δεν μπορούσε να λυθεί με τις γνωστές έως τότε τεχνικές (Hecht-Nielsen, 1989).

Η μέθοδος οπισθοδιάδοσης του λάθους «άνθισε» την δεκαετία του ογδόντα και η δημοφιλία της την οδήγησε στο να γίνει η πλέον χρησιμοποιούμενη τακτική στα τεχνητά νευρωνικά δίκτυα. (Bryson & Ho, 1969), (Werbos, 1990), (Parker, 1982), (Rumelhart & McClelland, 1986). Η μέθοδος του αισθητήρα ενός και δύο επιπέδων δεν μπορούσε να καλύψει εφαρμοσμένες εφαρμογές έτσι η ανάγκη για ένα νέο μοντέλο δικτύων πολλών επιπέδων έγινε πραγματικότητα. Τα δίκτυα πολλών επιπέδων δεν είχαν τους αναγκαίους τύπους για την περιγραφή της εκπαίδευσής τους. Η τεχνική της οπισθοδιάδοσης κάλυπτε αυτό το κενό και αναπτύχθηκε σε διάφορες πρώιμες μορφές από τους Bryson, Ho, Werbos, Parker. Η εγκαθίδρυση της τεχνικής έγινε από τους Rumelhart και McClelland το 1986.

Η μέθοδος της οπισθοδιάδοσης του λάθους έχει τα ακόλουθα χαρακτηριστικά:

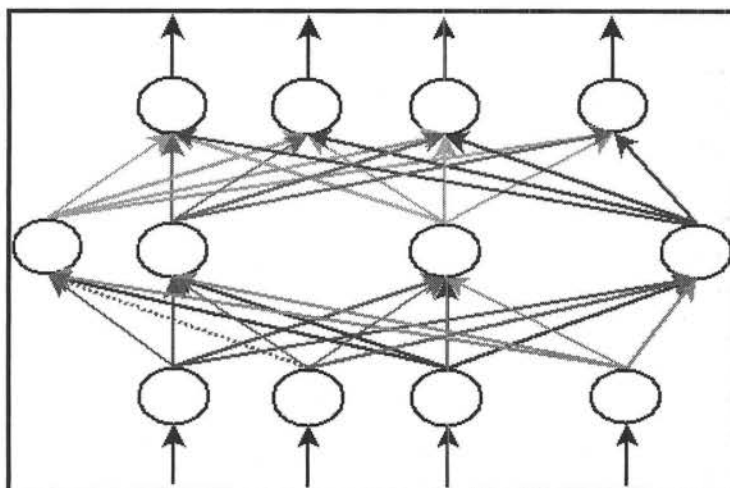
- Η θεώρηση είναι καθαρά μαθηματική με αυστηρά τεκμηριωμένες αποδείξεις.
- Είναι δίκτυο πολλών επιπέδων, όπου το κάθε επίπεδο δύναται να έχει πολλούς νευρώνες.
- Περίπλοκη δομή σε σχέση με τον αισθητήρα.
- Δεν υπάρχει σύνδεση μεταξύ νευρώνων του ίδιου επιπέδου.
- Οι νευρώνες διαφορετικών επιπέδων συνδέονται με αποτέλεσμα να έχουμε πολλές σειρές με βάρη w μεταξύ των επιπέδων.
- Δυνατότητα μεταβολής βαρών των ενδιάμεσων επιπέδων

Ένα δίκτυο που ακολουθεί την μέθοδο της οπισθοδιαδόσεως του λάθους ξεκινά αρχικά την διαδικασία της μάθησης με τυχαίες τιμές στα βάρη των συνάψεων. Εάν δοθεί λάθος απάντηση, κάτι που αρχικά συγκεντρώνει τις περισσότερες πιθανότητες τα βάρη διορθώνονται με τέτοιο τρόπο ώστε το λάθος να μειώνεται συνεχώς μέχρι να γίνει πολύ μικρό και ανεκτό για την επίλυση του εκάστοτε προβλήματος. Ο εξωτερικός παρατηρητής θεωρεί ότι το δίκτυο έχει «καταλάβει» τα παραδείγματα που του δίδαξε με την ακρίβεια που έχει θέσει ο ίδιος στο ελάχιστο αποδεκτό ποσοστό λάθους.

Στο μοντέλο του αισθητήρα το νευρωνικό δίκτυο δεν περιέχει κάποια εσωτερική αναπαράσταση καθώς τα πρότυπα εισόδου και εξόδου είναι του ίδιου τύπου. Αυτός ο περιορισμός δεν επιτρέπει σε αυτά τα δίκτυα να λύσουν πιο γενικά ή πιο περίπλοκα προβλήματα.

Στο πρόβλημα της XOR απαιτείται δύο πρότυπα που είναι τελείως διαφορετικά να δώσουν τελικά την ίδια απάντηση. Αν δίναμε στο δίκτυο μια διαφορετική δομή που θα του έδινε την δυνατότητα να μπορεί να απεικονίσει μια εσωτερική αναπαράσταση των σημάτων εισόδου, τότε θα ξεπερνούσαμε αυτήν την δυσκολία (Touretzky & Pomerlau, 1989).

Η ιδέα ήταν να προσθέσουμε ένα τρίτο επίπεδο μεταξύ του επιπέδου εισόδου και εξόδου που το ονομάζουμε κρυμμένο επίπεδο και το οποίο έχει την δυνατότητα της εσωτερικής αναπαράστασης σημάτων. Το «κρυφό» επίπεδο δεν έχει άμεση είσοδο ούτε άμεση έξοδο αλλά βλέπει μόνον το εσωτερικό του και εσωκλείεται μεταξύ της εισόδου και της εξόδου.



Εικόνα 14. Δίκτυο πολλαπλών επιπέδων με επίπεδο εσωτερικής αναπαράστασης

1.2.13 Το πρότυπο Hopfield

Μια άλλη κατηγορία νευρωνικών δικτύων είναι τα δίκτυα Hopfield τα οποία αναπτύχθηκαν ιδιαίτερα την δεκαετία του 80' (Hopfield, 1982), (Hopfield, 1984).

Το 1982 ο Hopfield αναθέρμανε το στάσιμο ενδιαφέρον στην εξέλιξη των νευρωνικών δικτύων όταν εισηγήθηκε ένα νέο μοντέλο δικτύων που παρουσιάζουν εξαιρετικό ενδιαφέρον σε κύριες κατηγορίες προβλημάτων: Προβλήματα βελτιστοποίησης και προβλήματα συνειρμικής μνήμης.

Η τεχνική του Hopfield δεν πρωτοτύπησε στην επίλυση άλυτων προβλημάτων, ενώ οι λύσεις σε μερικές περιπτώσεις που δίνει το δίκτυο είναι μερικές φορές πιο αργές από άλλες μορφές επίλυσης. Αλλά τα δίκτυα Hopfield παρουσιάζουν μια διορατικότητα ως προς την φύση του προβλήματος που εξετάζουν. Το κύριο

χαρακτηριστικό των δικτύων Hopfield είναι η ανάδραση κάθε έξοδος ενός νευρώνα χρησιμοποιείται ως είσοδος για όλους του άλλους νευρώνες.

1.2.14 Χρησιμότητες και μειονεκτήματα μοντέλων νευρωνικών δικτύων

Κάθε πρότυπο δημιουργήθηκε με σκοπό να καλύψει μια συγκεκριμένη ανάγκη επιλύοντας με συγκεκριμένη μέθοδο ένα ή περισσότερα προβλήματα. Η ειδικότητα αυτή της εφαρμογής του εκάστοτε νευρωνικού δικτύου οδήγησε σε δομικά μειονεκτήματα, τα οποία άλλοτε δημιουργούσαν πρόβλημα και άλλοτε ήταν αδιάφορα για τον συγκεκριμένο τύπο επίλυσης. Παρακάτω παρατίθεται ένας πίνακας με συνοπτική αναφορά στην χρησιμότητα και στα μειονεκτήματα του κάθε νευρωνικού δικτύου.

Πίνακας 2. Πίνακας χρησιμότητων και μειονεκτημάτων νευρωνικών δικτύων

ΠΡΟΤΥΠΟ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ	ΧΡΗΣΙΜΟΤΗΤΑ	ΜΕΙΟΝΕΚΤΗΜΑ
Αισθητήρας	Αναγνώριση χαρακτήρων	Επίλυση γραμμικών προβλημάτων
Πρότυπο Adaline–Madaline	Φίλτρα σε τηλεπικοινωνιακές και εμπορικές εφαρμογές	Γραμμική σχέση εισόδων εξόδων
Πρότυπο χιονοστιβάδας	Αναγνώριση προφορικού λόγου	Πρόβλημα ταχύτητας
Παρεγκεφαλιδικό πρότυπο	Κίνηση βραχιόνων ρομπότ	Περίπλοκοι είσοδοι
Πρότυπο προσαρμοσμένου συντονισμού	Αναγνώριση προτύπων	Συντήρηση, δυσκαμψία μετατροπών
Πρότυπο εγκεφάλου-σε-κουτί	Εκπαίδευση με βάσεις δεδομένων	Εκπαίδευση βήμα προς βήμα
Πρότυπο νευρωνώστη	Αναγνώριση χειρόγραφου κειμένου	Απαίτηση μεγάλων δικτύων και νευρώνων
Πρότυπο Kohonen	Αναπαράσταση γεωμετρικών περιοχών	Μεγάλος χρόνος εκπαίδευσης
Πρότυπο Hopfield	Σύνθεση δεδομένων διάσπαρτων τμημάτων	Μικρή ταχύτητα
Στατιστικά δίκτυα	Αναγνώριση προτύπων ανάλυση δεδομένων ραντάρ	Μεγάλος χρόνος εκπαίδευσης
Πρότ. συνειρμικής μνήμης διπλής κατευθύνσεως	Συνειρμική μνήμη	Κωδικ. δεδομένων, μικρή αποθ. δεδομένων
Μοντέλο οπισθοδιάδοσης	Αναγν. ομιλίας, σύνθεση ομιλίας από κείμενο,	Απαίτηση πολλών παραδειγμάτων
Πρότυπο αντίθετης διάδοσης	Συμπίσση δεδομένων, στατική ανάλυση δεδομένων	Πολύπλοκη δομή, δυσκολία συντήρησης

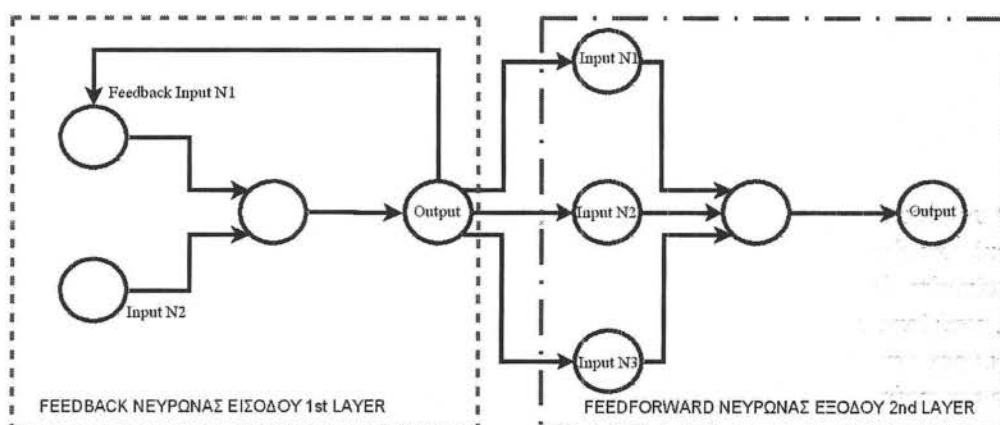
1.3 Ανεπτυγμένα πρότυπα τεχνητών νευρωνικών δικτύων: Χαοτικά Νευρωνικά Δίκτυα

1.3.1 Εισαγωγή στα Χαοτικά Νευρωνικά Δίκτυα

Η θεωρία του χάους είναι μια μαθηματική θεωρία που χρονολογεί τις αρχές της στην δεκαετία του '60 και «άνθισε» από την επιστημονική κοινότητα την δεκαετία του '80. Η διατομή του χάους με την νευροβιολογία χρονολογείται στο 1990. Ενώ η χρήση της θεωρίας του χάους στην ανάπτυξη και στην μελέτη των τεχνητών νευρωνικών δικτύων βρίσκεται ακόμα σε πρώιμο στάδιο και σαν επιστήμη δεν έχει πάνω από 5 χρόνια ζωής. Στο κεφάλαιο αυτό θα γίνει αναφορά στις γενικές αρχές της θεωρίας του χάους, σε έρευνες που συσχετίζουν την θεωρία του χάους και την νευροβιολογία όπως αυτή αναφέρεται στα βιολογικά νευρωνικά δίκτυα. Τελικά θα γίνει επιφανειακή μελέτη για την σημασία του χάους στην ανάπτυξη των τεχνητών νευρωνικών δικτύων. Τα νευρωνικά δίκτυα όπως έχει αναφερθεί προσπαθούν να μιμηθούν ακριβέστερα την δραστηριότητα του εγκεφάλου τόσο μικροσκοπικό επίπεδο (βιολογικοί νευρώνες) όσο μακροσκοπικά (γενική λειτουργία και δραστηριότητα του ανθρωπίνου εγκεφάλου). Οι εξελίξεις αυτές οδήγησαν σε τεχνικές όπως τα Χεμπιανά μοντέλα μάθησης (Hebbian learning) και μη-εποπτευόμενες τεχνικές μάθησης οι οποίες προσέγγιζαν στο μέτρο του δυνατού τα βιολογικά ανάλογα-μοντέλα μάθησης. Πολλές από αυτές τις τεχνικές ανάπτυξης και σχεδιασμού των νευρωνικών δικτύων είχαν απτά και σημαντικά αποτελέσματα όμως αναζητούνταν συνεχώς από την επιστημονική κοινότητα νέοι μέθοδοι ανάπτυξης νευρωνικών δικτύων. Η θεωρία του Χάους ήταν άλλος ένας από τους πολλούς τομείς της ανώτερης επιστήμης που θα συνδεόταν με την ανάπτυξη των νευρωνικών δικτύων (Gross, n.d.).

Ένα τεχνητό νευρωνικό σύστημα δύναται να παράγει μια μη αιτιοκρατική συμπεριφορά. Κάτι τέτοιο θα μπορούσε να γίνει αν φανταστούμε ένα δίκτυο δύο επιπέδων που έχουν feed-forward (δίκτυα με δομές πολλαπλών επιπέδων και στα οποία το σήμα διαδίδεται πάντοτε στην ίδια κατεύθυνση, από το επίπεδο εισόδου προς το επίπεδο εξόδου) και feed-back (Το δίκτυο χρησιμοποιεί έναν εσωτερικό έλεγχο, ψάχνει να βρει κάποιες τάσεις ή κανονικότητα στα σήματα εισόδου και προσπαθεί ώστε οι εξοδοί να έχουν τα ίδια χαρακτηριστικά όπως και οι εισοδοί. Λέμε ότι έτσι έχουμε αυτο-εποπτευόμενη εκπαίδευση διότι το δίκτυο ελέγχει τον εαυτό του και διορθώνει τα σφάλματα στα δεδομένα με ένα μηχανισμό ανάδρασης) έξοδο. Ένα παράδειγμα εισόδου ενός νευρώνα σε ένα τέτοιο σύστημα ανατροφοδοτεί την έξοδο στον εαυτό του με ένα υψηλό βάρος, το ίδιο καλά με το να τροφοδοτεί με το σήμα εξόδου του πρώτου νευρώνα (1st Layer) με τις εισόδους του δεύτερου νευρώνα (2nd Layer) καθεμία από τις οποίες έχει χαμηλό βάρος στην σύνδεση με τον νευρώνα δείγμα(ή διαφορετικά θα μπορούσαμε να πούμε ότι υπάρχει μεγαλύτερη τιμή κατωφλίου –higher threshold-). Φανταστείτε ότι μια αρχική εισαγωγή σε ένα τέτοια σύστημα θα προκαλούσε μια πυροδότηση η οποία δεν θα ήταν αρκετή έτσι ώστε να διεγείρει (trigger) την πυροδότηση οποιοδήποτε νευρώνα εισόδου του δεύτερου

επιπέδου(2nd Layer) αλλά ικανή αρκετά ώστε να ανατροφοδοτήσει τον εαυτό του (ο νευρώνας του πρώτου επιπέδου) ώστε να επαναπροδοτηθεί. Λειτουργώντας έτσι αναδραστικά. Ο νευρώνας ύστερα από το αρχικό ερέθισμα (initial stimulus) αυτοπροδοτείται κυκλικά και συνεχόμενα σε χαμηλό επίπεδο. Τώρα ας υποθέσουμε ότι στο σύστημα δώσουμε την ίδια είσοδο για δεύτερη φορά. Σε αυτήν την χρονική στιγμή ο νευρώνας δεν θα πάρει μόνο το νέο ερέθισμα εισόδου αλλά θα πάρει και το ερέθισμα της προηγούμενης χρονικής στιγμής που προέρχεται από την κυκλική ανατροφοδότηση. Εάν αυτή η προστιθέμενη εισαγωγή καταφέρει να αυξήσει και την έξοδο του νευρώνα του πρώτου επιπέδου ίσως σκανδαλίσει (trigger) την πυροδότηση ενός ή περισσότερων νευρώνων εισόδου του νευρώνα του δεύτερου επιπέδου (επίπεδο



Εικόνα 15. Δυ-επίπεδο συστημα νευρώνων με ανάδραση

εξόδου) παράγοντας τελικά μια ετεροχρονολογημένη απόκριση σε μια δοσμένη είσοδο. Η απόκριση δηλαδή δεν θα παρουσιαστεί την πρώτη φορά που θα παρουσιαστεί είσοδος (Kohonen, 1987), (Gross, n.d.), (Argyarakis, 2001).

Υπάρχουν διάφορες παραλλαγές του παραπάνω παραδείγματος, όπως αντί για ένα νευρώνα ανακύκλησης να έχουμε δύο η περισσότερους οι οποίοι συνεχώς θα ανακυκλώνουν την έξοδο με την είσοδο. Σε αυτή την περίπτωση η έξοδος για δοσμένη είσοδο δεν θα βασίζεται μόνο στο αν η είσοδος έχει προϋπάρξει αλλά και στο ποιος νευρώνας βρίσκεται σε ανάδραση την χρονική στιγμή που η νέα είσοδος παρουσιάζεται στο κύκλωμα.

Την όλη διαδικασία στο επίπεδο του εγκεφάλου θα μπορούσαμε να την παρομοιάσουμε με ένα spin (κύκλιση) ενός βιολογικού νευρωνικού δικτύου. Οι νευροβιολόγοι έχουν διαπιστώσει ότι τέτοιες χαμηλού επιπέδου δραστηριότητες πάντα παρουσιάζονται στον ανθρώπινο εγκέφαλο, αν και για αρκετό καιρό οι επιστήμονες υποστήριζαν ότι απλώς πρόκειται για μη συσχετιζόμενο «ηλεκτρικό θόρυβο». Οι τελευταίες έρευνες ωθούν τους νευροβιολόγους να πιστεύουν ότι τέτοιες δραστηριότητες απέχουν πολύ από το να είναι τυχαίες και μη σχετικές, για την

ακρίβεια πιστεύουν ότι είναι χαοτικές και εξαιρετικά ουσιαστικές για την ορθή λειτουργία του ανθρώπινου εγκεφάλου.

Σε μελέτες που έγιναν το 1991 ερευνητές συγκρίναν την αναγνώριση μορφών – προτύπων (pattern recognition) ενός βιολογικού και ενός τεχνητού νευρωνικού δικτύου και συμπέραναν ότι η διαδικασία της αναγνώρισης προτύπων των συστημάτων που βασίζονται σε τεχνητά νευρωνικά δίκτυα και πιο συγκεκριμένα στο πρότυπο του αισθητήρα (perceptron) λειτουργούν από την «χαλάρωση» μιας εκ των πολλών σταθερών καταστάσεων (Yao, et al., 1991). Ενώ αντίθετα τα βιολογικά νευρωνικά δίκτυα δεν ελαχιστοποιούσαν την συνάρτηση ενέργειας, αλλά αντ' αυτού διατηρούσαν την ταλαντευτική τους δραστηριότητα που κάποιες φορές ήταν περιοδική αλλά κυρίως και πιο συνηθισμένα ήταν χαοτική. Τι εννοούμε όμως με την έννοια χαοτική δραστηριότητα;

1.3.2 Η θεωρία του χάους

Σκοπός της θεωρίας του Χάους είναι η μελέτη της συμπεριφοράς ορισμένων μη γραμμικών δυναμικών συστημάτων τα οποία υπόκεινται στο φαινόμενο που αποκαλείται χάος. Το φαινόμενο αυτό χαρακτηρίζεται από την ευαίσθητη εξάρτηση από τις αρχικές συνθήκες (έχει επικρατήσει να αποκαλείται η συγκεκριμένη εξάρτηση ως φαινόμενο της πεταλούδας) και από μη περιοδικότητα. Η ευαισθησία αυτή προκαλεί την φαινομενική τυχαιότητα της παρατηρούμενης συμπεριφοράς των συστημάτων, χωρίς ιδιότητα αυτή να επηρεάζεται από την αιτιοκρατικότητα του συστήματος θεωρώντας ότι οι νόμοι της εξέλιξης τους είναι ορισμένοι σε ορθή βάση και δεν περιέχουν τυχαίες παραμέτρους. Συστήματα που παρουσιάζουν μαθηματικό χάος είναι αιτιοκρατικά (ντετερμινιστικά) και επομένως εύτακτα. Ο όρος χάος έρχεται σε αντίθεση με την καθομιλουμένη όπου με την λέξη χάος υποδηλώνεται η παντελής έλλειψη τάξης. Ενώ για την μελέτη των ντετερμινιστικών συστημάτων μέσω της θεωρίας του χάους απαιτείται και η ταυτόχρονη μελέτη του συγγενές πεδίου της κβαντικής θεωρίας του Χάους (Devaney, 1989), (Wikipedia, 2012), (Eckmann & Ruelle, 1985), (Stewart, 1989), (Banks, et al., n.d.) .

Μια εκ των συμπεριφορών ενός συστήματος όπως περιγράφηκε προηγουμένως δύναται να εκτελεί χαοτική κίνηση. Κατά τον Devaney για να χαρακτηρίσουμε την συμπεριφορά ενός συστήματος ως χαοτική το σύστημα θα πρέπει έχει τις παρακάτω ιδιότητες:

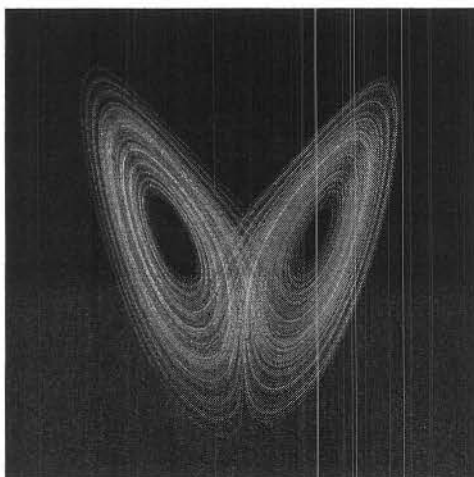
- Να παρουσιάζει ευαίσθητη εξάρτηση από τις αρχικές συνθήκες.
- Να είναι μεταβατικό σε τοπολογικό επίπεδο.
- Να παρουσιάζει πυκνό σύνολο περιοδικών τροχιών.

Η πρώτη ιδιότητα δηλαδή η ευαισθησία στις αρχικές συνθήκες σημαίνει ότι δύο σημεία μπορούν να ακολουθήσουν τελείως διαφορετικές τροχιές στον χώρο ακόμα και αν είναι ελάχιστη η διαφορά από τις αρχικές συνθήκες. Όταν η αρχική διαμόρφωση είναι η ίδια τα συστήματα συμπεριφέρονται με τον ίδιο τρόπο. Δηλαδή οι αρχικές συνθήκες θα πρέπει να προσδιοριστούν με απεριόριστη ακρίβεια έτσι ώστε

να προβλεφθεί η συμπεριφορά του συστήματος για χρόνους εκτός του χρονικού ορίζοντα. Πρακτικά αυτό είναι αδύνατον.

Με την έννοια μεταβατικότητα εννοούμε ότι οποιαδήποτε μετατροπή σε ένα ορισμένο διάστημα θα διατηρείται μέχρι την επικάλυψη οποιοδήποτε άλλου ορισμένου διαστήματος. Οι τρεις παραπάνω ιδιότητες ισχύουν και σε έναν αυθαίρετο μετρικό χώρο.

Η οπτική απεικόνιση της χαοτικής κίνησης γίνεται μέσω της κατασκευής διαγράμματος φάσης της κίνησης. Στο διάγραμμα υπεισέρχεται σιωπηρά ο χρόνος ενώ σε κάθε άξονα αναπαρίσταται μια μεταβλητή της κατάστασης. Ένα τέτοιο παράδειγμα είναι αυτό της θέσης ενός εκκρεμούς σε σχέση με την ταχύτητα του. Το εκκρεμές σε ακινησία θα σχεδιαστεί ως ένα σημείο και το εκκρεμές που βρίσκεται σε κατάσταση περιοδικής κίνησης θα σχεδιαστεί ως μια απλή κλειστή καμπύλη. Όταν το σχέδιο σχηματίζει κλειστή καμπύλη η καμπύλη αυτή λέγεται τροχιά. Το εκκρεμές παρουσιάζει άπειρες τέτοιες τροχιές. Η πλειοψηφία των τροχιών τείνουν σε ένα κοινό όριο. Τελικά το σύστημα εκτελεί την ίδια κίνηση για τις αρχικές καταστάσεις σε μια περιοχή γύρω από την κίνηση σχεδόν σαν να έλκεται το σύστημα σε αυτή την κίνηση. Μια τέτοια ελκυστική κίνηση καλείται ελκυστής του συστήματος.



Εικόνα 16. Ο ελκυστής του Λόρεντζ

Το 1987 οι Skarda και Freeman (Skarda & Freeman, 1987), οι οποίοι ήταν αυτοί που υποστήριξαν την χαοτική δραστηριότητα στον ανθρώπινο εγκέφαλο, δίνουν την εξής περιγραφή για το χάος: «Το χάος είναι ελεγχόμενος θόρυβος με ακριβώς καθορισμένες ιδιότητες». Το 1990 υποστηρίχθηκε ότι: «Σε ένα δυναμικό σύστημα το χάος είναι μια λύση σταθερής κατάστασης αλλά δεν είναι μια λύση σταθερής κατάστασης με αλληλοεξόντωση δυνάμεων (equilibrium state) όπως επίσης το χάος δεν αποτελεί περιοδική λύση (Freeman & Yao, 1990).

Η πραγματικότητα βρίσκεται κάπου ανάμεσα στις παραπάνω θεωρίες το χάος δηλαδή βρίσκεται ανάμεσα στην περιοδική-προβλέψιμη συμπεριφορά και στην τελείως τυχαία συμπεριφορά.

1.3.3 Η θεωρία του χάους στον ανθρώπινο εγκέφαλο

Η ύπαρξη ή μη της θεωρίας του χάους στον ανθρώπινο εγκέφαλο απασχόλησε τους επιστήμονες για περισσότερο από 10 χρόνια. Ύστερα από πολλές έρευνες η χαοτική συμπεριφορά του ανθρώπινου εγκεφάλου ανακαλύφθηκε τόσο μικροσκοπικά όσο και μακροσκοπικά.

Σε μικροσκοπικό επίπεδο δηλαδή στο επίπεδο νευρώνων το 1983 υποστηρίχθηκε η άποψη ότι η ανακάλυψη του χάους σε νευρωνικό επίπεδο θα μπορούσε να είναι υπεύθυνη για διαταραχές του ανθρώπινου εγκεφάλου όπως σχιζοφρένια, αϋπνία επιληψία κλπ. (Guevara, et al., 1983).

Η ανακάλυψη του χάους στον ανθρώπινο εγκέφαλο είναι εξαιρετικά ενδιαφέρουσα σε μακροσκοπικό επίπεδο. Εν αντιθέσει με τις μέχρι πρότινος πεποιθήσεις ότι το χάος αντιπροσώπευε μια πιθανή πηγή επιβλαβούς διαταραχής για τον εγκέφαλο οι επιστήμονες κατέλειξαν στο συμπέρασμα ότι το χάος ήταν ουσιαστικό για την ορθή λειτουργία του ανθρώπινου εγκεφάλου. Ο Dr. Walter Freeman ήταν ο στυλοβάτης της θεωρίας του χάους σε μακροσκοπικό επίπεδο στον εγκέφαλο. Η ανακάλυψη του Freeman για την χαοτική συμπεριφορά στην χάραξη του ηλεκτροεγκεφαλογράφηματος (EEG) των οσφρητικών βολβών στα κουνέλια οδήγησε σε πλήθος επιστημονικών δημοσιεύσεων για τον ρόλο του χάους στον εγκέφαλο και στην διασύνδεση τους με τα τεχνητά νευρωνικά δίκτυα. Ο Freeman παρατήρησε ότι για μερικά γνωστά αλλά σύνθετα ερεθίσματα η αναγνώριση ήταν σχεδόν στιγμιαία. «Ένα άτομο αναγνωρίζει ένα γνωστό πρόσωπο ή την μυρωδιά της σχάρας ή την γεύση μιας σοκολάτας σχεδόν μόλις παρουσιάζεται εκείνο το ερέθισμα... Πως μια τέτοια αναγνώριση συμβαίνει με τόση ακρίβεια και ταχύτητα ακόμα και όταν τα ερεθίσματα είναι σύνθετα και το πλαίσιο στο οποίο προκύπτουν ποικίλλει». Η απάντηση που καλύπτει τον ανωτέρω προβληματισμό του είναι το Χάος (Freeman & Yao, 1990).

Οι Skarda και Freeman το 1987 διαπίστωσαν ότι υπάρχει σταθερή δραστηριότητα στον οσφρητικό φλοιό και αυτή η δραστηριότητα είναι χαοτική (Skarda & Freeman, 1987). Θεωρεί ότι είναι πιθανό το υπόλοιπο μέρος του εγκεφάλου να συμπεριφέρεται με παρόμοιο τρόπο και για την υποστήριξη της θέσης του αυτή εισηγήθηκε ότι το χάος αποτελεί την βασική μορφή συλλογικής νευρωνικής δραστηριότητας για όλες τις αντιληπτικές διαδικασίες και λειτουργεί ως ελεγχόμενη πηγή θορύβου ως μέσο για την εξασφάλιση της συνεχούς πρόσβασης στα προηγούμενα (από πλευράς χρόνου) εκμαθημένα αισθητηριακά πρότυπα. Το χάος παρέχει στο σύστημα μια έτοιμη κατάσταση με τέτοιο τρόπο ώστε να αποφευχθεί η περιττή διαδικασία για το σύστημα με το να «ξυπνήσει» ή να επιστρέψει σε μια «κοιμισμένη» κατάσταση κάθε φορά που δίνεται μια νέα είσοδος (Freeman & Yao, 1990).

Γενικά ένα χαοτικό σύστημα και το χάος του ανθρώπινου εγκεφάλου εναλλάσσονται συχνά με ένα φαινομενικά τυχαίο τρόπο μεταξύ των διαφόρων περιοχών (ή των ομάδων συμπεριφορών) του φασικού διαστήματος (phase-space). Αυτές οι περιοχές είναι όπως αναφέρθηκε προηγουμένως είναι οι χαοτικοί ελκυστές. (βλ. φωτ. Ελκυστής του Λόρεντζ). Οι ελκυστές αυτοί αποκαλούνται και φτερά (βλ.

φαινόμενο της πεταλούδας). Ο τρόπος που ο εγκέφαλος χρησιμοποιεί το χάος για να εξασφαλίσει την συνεχή πρόσβαση στα προηγούμενα (από πλευράς χρόνου) εκμαθημένα αισθητηριακά πρότυπα είναι να «αναπτύξει» αυτά τα φτερά για διαφορετικά εκμαθημένα πρότυπα εισόδου. Σύμφωνα με τους ερευνητές η χαοτική δραστηριότητα που λειτουργεί στο υπόβαθρο επιτρέπει στο σύστημα να μεταπηδήσει ταχύτατα σε ένα από αυτά τα φτερά όταν παρουσιαστεί με την κατάλληλη είσοδο. Η μετάβαση έμπροσθεν και όπισθεν μεταξύ των φτερών ή μεταξύ του κεντρικού μέρους και ενός εκ των φτερών αντιπροσωπεύει την μετάβαση φάσης από φυσικής άποψης αλλά και από την άποψη της αναγνώρισης προτύπων των νευρωνικών δικτύων.

Εάν η είσοδος δεν στέλνει το σύστημα σε ένα από αυτά τα φτερά θεωρείται νέα εισαγωγή (πχ. μια άγνωστη μυρωδιά) και αντί για την παραγωγή ενός από τα προηγούμενα (από πλευράς χρόνου) εκμαθημένα αισθητηριακά πρότυπα το σύστημα τίθεται σε χαοτική κατάσταση υψηλού επιπέδου (high level chaotic state). Αυτή η διαδικασία επιτρέπει στο σύστημα να αποφεύγει προηγούμενα αισθητηριακά πρότυπα και να παράγει καινούργια (Skarda & Freeman, 1987).

Μερικοί ερευνητές πιστεύουν ότι αυτό το είδος χαοτικής συμπεριφοράς με υπόβαθρο είναι στην πραγματικότητα απαραίτητη για την συμμετοχή του εγκέφαλου στην συνεχή μάθηση. Έτσι ο ανθρώπινος εγκέφαλος ακολουθεί μια τακτική κατηγοριοποίησης μιας νέας εισόδου σε μια νέα κατηγορία παρά να προβεί στην διαδικασία να ταιριάζει-καταχωρήσει σε μια ήδη υπάρχουσα. Με την έλλειψη ενός τέτοιου μηχανισμού το σύστημα δεν δύναται να αποφύγει την επαναπαραγωγή προηγούμενων εκμαθημένων αισθητηριακών προτύπων και μπορεί μόνο να συγκλίνει σε ένα πρότυπο συμπεριφοράς που έχει ήδη μάθει.

1.3.4 Το χάος στα νευρωνικά δίκτυα

Ύστερα από την θεωρία - Πρόταση του Freeman (Freeman & Yao, 1990) [ότι «το χάος μπορεί να είναι η κύρια ιδιότητα που καθιστά τον εγκέφαλο διαφορετικό από μία artificial – intelligence μηχανή» ήταν η σειρά των επιστημόνων που ασχολούνταν με τα τεχνητά νευρωνικά δίκτυα να συμπληρώσουν το «κενό» που δημιουργήθηκε από αυτή την θεωρία.

Το 1988 ο Freeman ερευνούσε μια προσομοίωση του οσφρητικού φλοιού στον υπολογιστή με σκοπό να την πιο στενή και πιο συνεχή παρακολούθηση της δραστηριότητας που δυνατή με EEGs πάνω στα βιολογικά πρότυπα. Το συγκεκριμένο μοντέλο κατάφερε να αναπαραστήσει πολλά από τα χαρακτηριστικά των βιολογικών νευρωνικών δικτύων που ερευνούσε ο Freeman πάνω στον οσφρητικό φλοιό. Για το τεχνικό μέρος της επίτευξης του πειράματος χρησιμοποιήθηκαν 8 τεχνητοί νευρώνες (Eisenberg, et al., 1989).

Το 1990 αρκετοί ερευνητές αναπαρέστησαν την χαοτική συμπεριφορά μέσω τεχνητών νευρωνικών δικτύων με σχετικά απλά τεχνητά νευρωνικά δίκτυα. Μακροσκοπικά η χαοτική συμπεριφορά μπορεί να αναπαρασταθεί από πιο παραδοσιακά πρότυπα τεχνητών νευρωνικών δικτύων, με καλύτερο τρόπο σε σχέση

με τα απλοϊκά μοντέλα που αρχικά αναπτύχθηκαν (Aihara, et al., 1990), (Ikeguchi, et al., 1990).

Επόμενες έρευνες στην μακροσκοπική χαοτική συμπεριφορά στα τεχνητά νευρωνικά δίκτυα αναζητούσαν πως θα μπορούσαν να «καλλιεργήσουν» το χάος ως ένα ακούσιο υποπροϊόν ενός μοντέλου εμπροσθοδιάδοσης (feedforward) και οπισθοδιάδοσης (feedback), όπως για παράδειγμα ένα δίκτυο Hopfield. Διαπιστώθηκε ότι αυτού του είδους τα συστήματα ένα έχουν συγχρόνως ενισχυτικές και ανασταλτικές συνδέσεις μεταξύ των νευρώνων μπορούν να επιδείξουν χαοτική συμπεριφορά. Το 1990 οι Fukai και Shiino βρήκαν παρόμοια αποτελέσματα αναθέτοντας σε συγκεκριμένους νευρώνες την λειτουργία είτε της διέγερσης (excitation) είτε της παρεμπόδισης (inhibition). Αυτή η μέθοδος προτιμήθηκε από το να κρατήσουν του νευρώνες ουδέτερους και να αποδώσουν στα βάρη ενισχυτική ή ανασταλτική συμπεριφορά. Η όλη διαδικασία βασίστηκε στην υπόθεση Dale (Dale hypothesis) περιεχόμενο της οποίας ήταν ότι «στον ανθρώπινο εγκέφαλο ο κάθε νευρώνας έχει ενισχυτική ή ανασταλτική συμπεριφορά» (Choi & Huberman, 1983), (Fukai & Shiino, 1990).

Οι προσπάθειες για την εκμετάλλευση του χάους στα τεχνητά νευρωνικά δίκτυα για την παραγωγή αποτελεσμάτων παρόμοια με αυτά του Freeman είχαν αρκετή επιτυχία. Το 1990 ο Sandler ανακάλυψε ότι προσθέτοντας χάος σε ένα δίκτυο τύπου Hopfield ήταν δυνατή η αναγνώριση συγκεκριμένων εισόδων και όχι ολόκληρων προτύπων (form patterns), μια θεωρία που συνέβαλε στην εκλεκτική μάθηση των νευρωνικών δικτύων (Sandler, 1990). Η καλύτερη ένδειξη ότι το χάος μπορεί να χρησιμοποιηθεί ουσιαστικά στα τεχνητά νευρωνικά συστήματα είναι στην εκτέλεση μια διαδικασίας που έχει ήδη αναπτυχθεί. Ένα τέτοιο χαοτικό σύστημα σχεδιάστηκε για την αναγνώριση τεσσάρων διαφορετικών τύπων βιομηχανικών εξαρτημάτων με σκοπό την απόκριση για το αν είναι ελαττωματικό ή όχι το κάθε εξάρτημα. Για την μελέτη της επίδοσης η τεχνική αυτή συγκρίθηκε με μη-χαοτικές τεχνικές. Το αποτέλεσμα ήταν ότι τα χαοτικά τεχνητά νευρωνικά δίκτυα είχαν καλύτερη επίδοση σε σχέση με τα μη-χαοτικά, τόσο στην εύρεση ελαττωματικών και αποδεκτών από τον έλεγχο βιομηχανικών εξαρτημάτων (Yao, et al., 1991).

1.3.5 Το υπαρξιακό αίνιγμα του χάους στα τεχνητά νευρωνικά δίκτυα

Η ύπαρξη χάους στα βιολογικά νευρωνικά δίκτυα του ανθρώπινου εγκεφάλου ήταν μια ανακάλυψη που έγειρε το ενδιαφέρον αρκετών επιστημών που ασχολήθηκαν με τα τεχνητά νευρωνικά δίκτυα. Σε μια άنيση μάχη οι επιστήμονες προσπαθούν να μιμηθούν και να προσομοιώσουν τις δυνατότητες και τις λειτουργίες του ανθρώπινου εγκεφάλου άλλα πάντα θα βρίσκονται ένα βήμα πίσω γιατί η ανακάλυψη 100% της λειτουργίας του ανθρώπινου εγκεφάλου αργεί πολύ ακόμα. Ο Steven Hawking έλεγε ότι ξέρουμε σχεδόν τα πάντα για το σύμπαν εκτός από τα πρώτα τρισεκατομμυριοστά του δευτερολέπτου από την γέννηση του, δυστυχώς για εμάς όλα τα κρυμμένα μυστικά βρίσκονται σε αυτά τα τρισεκατομμυριοστά. Κάπως έτσι είναι και τα πράγματα για

την αποκρυπτογράφηση του ανθρώπινου εγκεφάλου μπορούμε να τον ανακαλύψουμε κατά 99,9% κάποια στιγμή, αλλά εκείνο το 0,1% θα είναι αυτό θα μας κρύβει τα πάντα.

Μερικοί επιστήμονες αναρωτιούνται κατά πόσο είναι χρήσιμη η θεωρία του χάους στα τεχνητά νευρωνικά συστήματα που επιδιώκουν να αναπαράγουν τις δυνατότητες του ανθρώπινου εγκεφάλου και μέχρι ποιο σημείο το χάος μπορεί να χρησιμοποιηθεί για την βελτίωση της απόδοσης των τεχνητών νευρωνικών συστημάτων. Το κατά πόσο είναι απαραίτητο το χάος στα τεχνητά νευρωνικά δίκτυα σαν ερώτημα δεν έχει απαντηθεί ακόμα.

Ο Bower το 1988 είπε ότι «το χάος είναι η ποιότητα που κάνει την διαφορά στην επιβίωση μεταξύ ενός πλάσματος με εγκέφαλο στον πραγματικό κόσμο και ενός ρομπότ που δεν μπορεί να λειτουργήσει έξω από ένα ελεγχόμενο περιβάλλον» (Bower, 1988). Μια θεωρία που από ότι φαίνεται θέλει ακόμα καιρό για να καταργηθεί...

Οι προσπάθειες του Freeman και των υπολοίπων δεν έπεισαν ένα σημαντικό μέρος της επιστημονικής κοινότητας που θεωρεί το χάος ως ένα κατανοητό υποπροϊόν περίπλοκων συστημάτων όπως ο ανθρώπινος εγκέφαλος ή τα τεχνητά νευρωνικά δίκτυα, χωρίς όμως να σημαίνει ότι το χάος αυτό καθαυτό δεν προσφέρει στην αποδοτικότητα του συστήματος. Μερικοί όπως ο Stephen Grossberg και ο Gail Carpenter θεωρούν ότι τα οφέλη που προσφέρονται σε υποθετικό επίπεδο από τα χαοτικά συστήματα μπορούν να επιτευχθούν με άλλους τρόπους στα τεχνητά νευρωνικά συστήματα (Gross, n.d.).

1.3.6 Χαρακτηριστικά χαοτικών νευρωνικών δικτύων

Όπως αναφέρθηκε και προηγουμένως η μοντελοποίηση χαοτικών δυναμικών συστημάτων μέσω των νευρωνικών δικτύων ξεκίνησε με αφορμή την εμφάνιση χαοτικών χαρακτηριστικών στα βιολογικά νευρωνικά δίκτυα τα οποία αποτελούν μη γραμμικά συστήματα. Ο μεγάλος βαθμός πολυπλοκότητας στην μελέτη των βιολογικών νευρωνικών δικτύων μας οδήγησε στην προσομοίωση τους με τεχνητά νευρωνικά δίκτυα, τα οποία μπορούν να μελετηθούν πολύ πιο εύκολα (Margaris, 2003). Έτσι με την πρόοδο των «επιστημών» της θεωρίας του χάους και των νευρωνικών δικτύων αναπτύχθηκαν τεχνητά νευρωνικά δίκτυα για την προσομοίωση των πιο γνωστών χαοτικών ελκυστών, όπως ο ελκυστής του Henon, του Rossler και του Lorenz. Θα πρέπει να σημειώσουμε σε αυτό το σημείο ότι για την πλήρη μελέτη ενός προτεινόμενου νευρωνικού μοντέλου και του χαοτικού δυναμικού συστήματος το οποίο αυτό προσομοιώνει απαιτείται ο υπολογισμός των βέλτιστων τιμών των παραμέτρων για την ανακατασκευή του χώρου των φάσεων.

Γενικά τα νευρωνικά δίκτυα έχουν τρεις τομείς συσχέτισης με τα μη γραμμικά συστήματα:

- Την προσομοίωση μη γραμμικών συστημάτων
- Την ανάλυση και επεξεργασία των εξαγόμενων αποτελεσμάτων των συστημάτων
- Τον εντοπισμό χαοτικών χαρακτηριστικών στα νευρωνικά δίκτυα

Οι δυσκολίες στην κατασκευή νευρωνικών μοντέλων για μη γραμμικά συστήματα οφείλονται κυρίως στο φάσμα ευρείας ζώνης που τα χαρακτηρίζει με κοινά χαρακτηριστικά με τον θόρυβο. Έτσι απαιτείται η κατασκευή φίλτρου συχνοτήτων για την απομάκρυνση του θορύβου από ένα χασοτικό σήμα. Τεχνική που από πολλούς επιστήμονες θεωρείται μη αποτελεσματική. Η αιτιοκρατική φύση των χασοτικών σημάτων δεν επιτρέπει την χρήση κοινών μεθόδων απομάκρυνσης θορύβου όπως φίλτρα τύπου Kalman. Έτσι όταν ένα σήμα έχει θόρυβο υψηλού επιπέδου και ο γεωμετρικός χώρος του μη γραμμικού συστήματος έχει μεγάλο πλήθος διαστάσεων η κατασκευή του νευρωνικού μοντέλου γίνεται αρκετά δύσκολη.

Για να είναι δυνατή η δημιουργία ενός νευρωνικού μοντέλου προσομοίωσης χασοτικής συμπεριφοράς το προς μελέτη μη γραμμικό σύστημα θα πρέπει έχει τις ακόλουθες ιδιότητες:

- Χαμηλό επίπεδο θορύβου
- Μέγιστο εκθέτη Lyapunov $\lambda_1 < 2,50 \text{ bits/sec}$
- Ο γεωμετρικός χώρος του συστήματος να έχει διάσταση εμβύθισης $d < 4$.

1.3.7 Χρησιμότητα νευρωνικών δικτύων στα μοντέλα χασοτικών συστημάτων

Σαν γενικό ερώτημα έχει απαντηθεί σε προηγούμενες παραγράφους του κεφαλαίου αυτού, εδώ θα προσπαθήσουμε να εξηγήσουμε σε επίπεδο μεθόδων – τεχνικών την χρησιμότητα - επιλογή των νευρωνικών δικτύων.

Τα νευρωνικά δίκτυα έχουν την δυνατότητα να περιγράψουν ένα σύστημα με βάση την μέθοδο της πολυωνυμικής προσέγγισης (polynomial approximation), χωρίς ωστόσο να ακολουθούν πλήρως όλα τα χαρακτηριστικά. Με βάση αυτή την τακτική προσεγγίζουμε την οποιαδήποτε συνάρτηση με την βοήθεια ενός πολυωνύμου, του συντελεστές του πολυωνύμου τους υπολογίζουμε μέσω της διαδικασίας ελαχιστοποίησης σφάλματος προσέγγισης. Η μέθοδος αυτή παρουσιάζει ευαισθησία στην επιλογή συντελεστών των μονωνύμων βάσης. Τα νευρωνικά δίκτυα υπολογίζουν τους συντελεστές του πολυωνύμου με δυναμικό τρόπο κατά την διάρκεια της εκπαίδευσης τους. Έτσι τα νευρωνικά δίκτυα είναι αποτελεσματικότερα στην μοντελοποίηση των μη γραμμικών συστημάτων σε σχέση με την τεχνική της πολυωνυμικής προσέγγισης.

Οι Gallant και White απέδειξαν το 1988 ότι ένα νευρωνικό δίκτυο με κρυφό επίπεδο και με συνάρτηση ενεργοποίησης το συνημίτονο (cosine activation function), δύναται να προσομοιώσει ανάπτυγμα Fourier μιας οποιαδήποτε συνάρτησης (Gallant & White, 1988).

Ο Hornic το 1989 απέδειξε πως τα πολυεπίπεδα Perceptron δίκτυα έχοντας οποιαδήποτε συνάρτηση ενεργοποίησης μπορούν να προσεγγίσουν οποιαδήποτε συνάρτηση με συγκεκριμένο βαθμό ακρίβειας (Hornic, et al., 1989).

Τα νευρωνικά δίκτυα αποτελούν δυναμικά συστήματα και κατά την εκπαίδευσή τους η προσέγγιση της κατάστασης ελαχίστου σφάλματος γίνεται και αυτή με δυναμικό τρόπο, αυτός είναι ένας ακόμα λόγος για τον οποίο τα νευρωνικά δίκτυα αποτελούν αξιόπιστη τεχνική προσέγγισης των μη γραμμικών συστημάτων.

Επίσης δίκτυα τέτοιου είδους μπορούν για οποιαδήποτε είσοδο να υπολογίσουν έξοδο [ιδιότητα γενίκευσης]. Η ιδιότητα αυτή λειτουργεί σε συνδυασμό με την τεχνική της παρεμβολής για την επεξεργασία χαοτικών σημάτων πεπερασμένου μήκους με υψηλά επίπεδα θορύβου.

1.4 Ανεπτυγμένα πρότυπα τεχνητών νευρωνικών δικτύων:

Κβαντικά Νευρωνικά Δίκτυα

1.4.1 Εισαγωγή στα κβαντικά νευρωνικά δίκτυα

Σε αυτό το κεφάλαιο θα περιγράψουμε την έννοια του κβαντικού νευρωνικού δικτύου, την έρευνα και την ανάπτυξη που έχει γίνει πάνω σε αυτά καθώς και τις προοπτικές τους (Ezhov & Ventura, 2000). Ο τομέας των κβαντικών νευρωνικών δικτύων είναι ένας τελειώς καινούργιος τομέας της επιστήμης των νευρωνικών δικτύων που ενσωματώνει και συνενώνει θεωρίες των νευρωνικών δικτύων με αυτές των κβαντικών υπολογιστών (quantum computation) (Feynman, 1986). Αυτή η νέα θεωρία πιστεύεται ότι ίσως μας δώσει νέες πληροφορίες για την κατανόηση της λειτουργίας του ανθρωπίνου εγκεφάλου. Η κύρια προσέγγιση τους γίνεται για την ανάπτυξη πρωτοφανών δυνατοτήτων νέων συστημάτων για την επεξεργασία πληροφοριών -συμπεριλαμβανομένων και των συστημάτων για την επίλυση κλασσικών δυσεπίλυτων προβλημάτων- με συνειρμική μνήμη και εκθετικές ικανότητες ταχύτητας που ίσως τελικά να έχουν την δυνατότητα να υπερκεράσουν τους περιορισμούς της πρότασης Church-Turing (Church-Turing thesis).

Ένας πολύ σημαντικός λόγος να ενδιαφερθεί κάποιος για τα κβαντικά νευρωνικά δίκτυα είναι για την περαιτέρω μελέτη της λειτουργίας του ανθρωπίνου εγκεφάλου. Η ουσιαστικότητα των κβαντικών διαδικασιών στον ανθρώπινο εγκέφαλο είναι αντικείμενο της σύγχρονης επιστήμης (Nanopoulos et.al., 2006) και θεωρείται ότι η ανάπτυξη των κβαντικών νευρωνικών δικτύων θα μας φέρει μερικά βήματα πιο κοντά στην προσομοίωση της ανθρώπινης σκέψης (Penrose, 1994). Ο Penrose έχει υποστηρίξει ότι μια νέα φυσική που δεσμεύει τα κβαντικά φαινόμενα με την γενική σχετικότητα μπορεί να εξηγήσει διανοητικές δυνατότητες όπως αυτές της κατανόησης, της συνειδητοποίησης και της συνείδησης.

Αυτή η προσέγγιση υποστηρίζει πιο πολύ την μελέτη των ενδοκυτταρικών δομών -όπως οι μικρο-τουμπουλίνες (microtubules)- και όχι τόσο την αυτονομία των νευρωνικών δικτύων (Hameroff & Rasmussen, 1990).

Οι δυνατότητες που έχει ο τομέας των κλασσικών τεχνητών νευρωνικών δικτύων τεχνητής νοημοσύνης μπορεί να γενικευτεί στην κβαντική περιοχή από τον εκλεκτικό συνδυασμό τους με τον τομέα των κβαντικών υπολογιστών (Brooks, 1999).

Τόσο η μελέτη για την κατανόηση της λειτουργίας του ανθρωπίνου όσο και η μελέτη των νέων δυνατοτήτων για την επεξεργασία πληροφοριών μέσω κβαντικών νευρωνικών δικτύων είναι πολλά υποσχόμενες. Ωστόσο σε αυτό το κεφάλαιο θα γίνει προσπάθεια για την περαιτέρω μελέτη των κβαντικών νευρωνικών δικτύων. Τα κβαντικά νευρωνικά δίκτυα είναι το επόμενο βήμα στην φυσική εξέλιξη των νευρωνικών δικτύων. Η μελέτη της λειτουργίας του ανθρωπίνου εγκεφάλου έχει γίνει στα αρχικά κεφάλαια και η ανάπτυξη της σχέσεως μεταξύ των (πλέον) κβαντικών νευρωνικών και ανθρωπίνου βρίσκεται σε πρώιμο ακόμα στάδιο παρά τις πρόσφατες θεωρίες που έχουν εισαχθεί. Έτσι λοιπόν η θεματική ενότητα θα βασιστεί στα τεχνητά παρά στα βιολογικά νευρωνικά δίκτυα. Διαφορετικές προσεγγίσεις θα αναλυθούν σχετικά με την κβαντική διανεμημένη επεξεργασία του κβαντικού υπολογισμού (Deutsch, 1985), (Everett, 1957). Πιο συγκεκριμένα η ερμηνεία των πολλαπλών συμπάντων κατά τον Everett της θεωρίας της κβαντομηχανικής μπορεί να χρησιμοποιηθεί ως το γενικό πλαίσιο για την παραγωγή των κβαντικών αναλόγων των τεχνητών νευρωνικών δικτύων (Dirac, 1958). Οι απόψεις δίστανται για το τι αποκαλούμε κβαντικό νευρωνικό δίκτυο (Domany, et al., 1992). Πολλοί ερευνητές – επιστήμονες χρησιμοποιούν δικές τους αναλογίες στην εγκατάσταση μιας σύνδεσης μεταξύ της κβαντομηχανικής και των νευρωνικών δικτύων οι κύριες έννοιες των δύο επιστημονικών τομέων που θα μελετήσουμε μπορούν να συνεξεταστούν (αλλά όχι αναλογικά) ως εξής:

Πίνακας 3. Πίνακας αναλογίας κβαντομηχανικής και νευρωνικών δικτύων

ΚΒΑΝΤΟΜΗΧΑΝΙΚΗ	ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ
Συνάρτηση κύματος	Νευρώνας
Υπέρθωση-συνοχή (superposition-coherence)	Βαρυτικές συνδέσεις
Μέτρηση –Μη συνεκτικότητα- (decoherence)	Εξέλιξη προσέλκυσης (attractor evolution)
Εμπλοκή (entanglement)	Κανόνας μάθησης
Μετασχηματισμοί μονάδων (unitary transformations)	Συνάρτηση κέρδους (gain function transformation)

Για την μελέτη των παραπάνω βασικών εννοιών απαιτείται προσοχή καθώς δεν πρέπει να εξεταστούν αναλογικά και η σειρά τους στον πίνακα είναι αυθαίρετη. Στην πραγματικότητα η «εγκαθίδρυση» τέτοιων αναλογιών αποτελεί καίριο στόχο στην ανάπτυξη κβαντικών νευρωνικών δικτύων. Η θεωρία και οι αρχές της κβαντομηχανικής έχουν μεγαλύτερη απήχηση για την απεικόνιση τους μέσω της κλασσικής επιστήμης των υπολογιστών παρά με την νευροπληροφορική. Οι αρχές των κβαντικών υπολογιστών εισηγήθηκαν για πρώτη φορά το 1986 από τον Feynman και της περίφημες ομιλίες του που ενέπνευσαν αρκετούς επιστήμονες. Πιο συγκεκριμένα ο Feynman στις εργασίες του εξέτασε και μελέτησε τον ρόλο που θα έπαιζαν τα κβαντικά φαινόμενα στην μελλοντική ανάπτυξη του υλικού των υπολογιστών (hardware) (Feynman, 1986). Όσο η ταχύτητα των υπολογιστών συνέχιζε να αυξάνει με ταχύτετους ρυθμούς το μέγεθος των υπολογιστών μειωνόταν σταδιακά, ο Feynman υπέθεσε ότι μια μέρα τα καλώδια καθώς και άλλα υπολογιστικά στοιχεία ίσως αποτελούνται από μόνο μερικά ατομικά σωματίδια, σε μια τέτοια εξελικτική «εποχή» τα κβαντικά φαινόμενα θα έπαιζαν βαρυσήμαντο ρόλο στην ανάπτυξη και την σχεδίαση του υλικού των υπολογιστών. Τα τελικά συμπεράσματα της εργασίας του Feynman ήταν ότι οι κβαντικοί υπολογιστές θα είχαν σημαντικά προτερήματα έναντι των κλασσικών υπολογιστών καθώς και διαφορετική προσεγγιστική λογική επίλυσης των προβλημάτων. Το 1985 ο Deutsch συνεχίζοντας το εξαίρετο και πρωτοποριακό για την εποχή του έργο του Feynman καθόρισε τις αρχές της επιστήμης των κβαντικών υπολογιστών (Deutsch, 1985).

Είναι σημαντικό να παρατηρήσουμε ότι ένα μεγάλο μέρος των κλασσικών (μη-κβαντικών) τεχνητών νευρωνικών δικτύων οφείλεται στην παραλληλότητα τους κατά την κατανομημένη μετάδοση της πληροφορίας στο εσωτερικό τους, καθώς και στην μη γραμμικότητα των μετασχηματισμών που γίνονται από τις νευρωνικές συνάψεις. Από την άλλη μεριά η κβαντομηχανική προσφέρει την δυνατότητα ενός ακόμα πιο δυνατού παραλληλισμού αυτού του κβαντικού παραλληλισμού που εκφράζεται από την αρχή της υπέρθεσης. Αυτή η αρχή παρέχει στην επιστήμη των κβαντικών υπολογιστών ένα τεράστιο πλεονέκτημα στην διαχείριση και στην επεξεργασία μεγάλης ποσότητας δεδομένων. Οι κβαντικοί υπολογιστές ενσωματώνουν την παράλληλη επεξεργασία όλων των πιθανών συνδυασμών μιας κατάστασης μιας εγγραφής που αποτελείται από N αριθμό κβαντικών bit (qubit). Μόνο ένα αποτέλεσμα μπορεί να διαβαστεί ύστερα από την ασυνεκτικότητα (decoherence) της κβαντικής υπέρθεσης σε μία εκ των καταστάσεων βάσης. Ωστόσο η εμπλοκή (entanglement) παρέχει την δυνατότητα της μέτρησης των καταστάσεων όλων κβαντικών μπιτ σε ένα καταχωρητή. Οι μαθηματικοί τύποι της κβαντομηχανικής είναι αρκετά αποδεκτοί και κατανοητοί από την επιστημονική κοινότητα ωστόσο η φυσική σημασία για το τι ακριβώς «σημαίνει» η θεωρία της κβαντομηχανικής αποτελεί σημείο διαφορετικών προσεγγίσεων της επιστημονικής κοινότητας (Ezhov & Ventura, 2000).

1.4.2 Η αναγνώριση προτύπων οδηγεί στα κβαντικά νευρωνικά δίκτυα

Μια απλή προσέγγιση για το πώς η επιστήμη της αναγνώρισης προτύπων μπορεί να αποκαλεστεί μέθοδος με χρήση οδηγού(template-based method), όπου παραδείγματα από διαφορετικές τάξεις προτύπων μπορούν να αποθηκευτούν ξεχωριστά ως πολλαπλοί οδηγοί για άλλα πρότυπα. Ένα παρουσιαζόμενο ερέθισμα στο νευρωνικό δίκτυο μπορεί να αναγνωριστεί και να κατηγοριοποιηθεί σύμφωνα με την τάξη του προτύπου-οδηγού που ταιριάζει στον μεγαλύτερο δυνατό βαθμό με το ερέθισμα εισόδου. Για να είναι αποδοτική αυτή η διαδικασία θα πρέπει να εκτελεστεί παράλληλα έτσι ώστε ο αριθμός των αποθηκευμένων προτύπων-οδηγών να είναι αποτρεπτικός στην ακολουθιακή διαδικασία. Γενικά αυτή η μέθοδος χαρακτηρίζεται από χαμηλή απόδοση εξαιτίας της έλλειψης γενίκευσης καθώς και της αναγκαιότητας για την εγγύηση ορθής αναγνώρισης (Ezhov & Ventura, 2000).

Τα νευρωνικά δίκτυα παρέχουν την δυνατότητα για την χρησιμοποίηση μόνο ενός συστήματος για την αποθήκευση πολλαπλών δεδομένων που ανήκουν σε διαφορετικές κλάσεις-τάξεις και την ταξινόμηση του παρουσιαζόμενου ερεθίσματος με παράλληλο και καταναμημένο τρόπο. Έτσι το πρόβλημα του παραλληλισμού μπορεί να λυθεί με φυσικό τρόπο με μία τέτοια προσέγγιση. Επιπλέον η δυνατότητα προσεγγίσεως με αυθαίρετο τρόπο σύνθετων συναρτήσεων, κάνει τα νευρωνικά δίκτυα πολύ αποτελεσματικά στην δημιουργία κατηγοριοποιημένων συστημάτων (Bishop, 1995).

Είναι συχνά ποθητό να χρησιμοποιούνται πολύ-συναρτησιακά συστήματα που να βασίζονται σε νευρωνικά δίκτυα μονής τάξης. Σε έναν τέτοιο σχηματισμό, το νευρωνικό δίκτυο εκπαιδεύεται χρησιμοποιώντας παραδείγματα προτύπων που ανήκουν σε μια συγκεκριμένη τάξη-κλάση, και για κάθε κλάση εκπαιδεύεται για την αναγνώριση των προτύπων ένα διαφορετικό δίκτυο. Η κατηγοριοποίηση πραγματοποιείται με την παρουσίαση του ερεθίσματος εισόδου σε κάθε ένα από τα πρότυπα-οδηγούς της κάθε κλάσης και συγκρίνοντας στην συνέχεια τις εξόδους με βάση κάποιο συγκεκριμένο κριτήριο, μέχρι τελικά να παρουσιασθεί το «πρότυπο νικητής» (Cotrell, et al., 1985), (Schwenk & Milgram, 1994).

Το πρόβλημα του παραλληλισμού υπάρχει και σε αυτή την προσέγγιση (όχι όμως τόσο έντονα όσο στην περίπτωση της μεθόδου με χρήση οδηγού), ωστόσο παρουσιάζει πολλά πλεονεκτήματα σχετιζόμενα με την χρησιμότητα της ψεύτικης μνήμης για την γενίκευση. Διάφοροι τύποι νευρωνικών συστημάτων μπορούν να χρησιμοποιηθούν ως βάση για την πολυ-συναρτησιακή αναγνώριση όπως είναι οι αυτο-συσχετιζόμενοι αισθητήρες αλλά κυρίως τα δίκτυα τύπου Hopfield. Ως γνωστόν οποιαδήποτε κατάσταση του δικτύου Hopfield είτε είναι σταθερός ελκυστής ή εξελίσσεται σε κάποιον ελκυστή. Είναι σύνηθες να ερμηνεύουμε τους ελκυστές ως πρότυπα μνήμης ή μερικές φορές ως ψεύτικη μνήμη, ενώ οι μη-σταθερές καταστάσεις μπορούν να θεωρηθούν ως διεφθαρμένες εκδοχές μνημών που περιέχουν αρκετές επιμέρους πληροφορίες για την ανάκτηση ενός εκ των απομνημονευμένων προτύπων που έχουν αποθηκευτεί ως η κοντινότερη σταθερή κατάσταση (Ezhov & Vvedensky, 1996).

Αρκετές μελέτες έχουν γίνει για να εξερευνηθούν τις ιδιότητες των περιεχομένων των διευθυνσιοδοτημένων μνημών που μπορούν να εφαρμοστούν στα δίκτυα Hopfield και των διάφορων παραγώγων τους. Το κύριο μειονέκτημα τέτοιου είδους μνημών είναι η περιορισμένη χωρητικότητα τους. Ωστόσο, χρησιμοποιώντας πιθανοκρατική ερμηνεία των καταστάσεων ενέργειας του δικτύου μπορεί να υποστηριχθεί ότι το δίκτυο Hopfield είναι το καταλληλότερο για την εξαγωγή, σε τοπικό επίπεδο, των πιο πιθανών εκδοχών ενός προτύπου, για το οποίο όλα τα αποθηκευμένα πρότυπα μπορούν να θεωρηθούν διεφθαρμένες εκδοχές (Müller, et al., 1995), (Ezhov, et al., 1990).

Αυτή η προσέγγιση μπορούμε επίσης να την φανταστούμε ως μια εφαρμογή που θα χρησιμοποιούσε τους οδηγούς-πρότυπα με κατανεμημένο τρόπο έτσι ώστε όλοι οι εκπρόσωποι μιας συγκεκριμένης κλάσης να συγκρίνονται με ένα εξωτερικό ερέθισμα με παράλληλο (και κατανεμημένο) τρόπο. Αυτό που είναι ακόμα πιο ενδιαφέρον είναι ότι αυτή η προσέγγιση ανοίγει την «πόρτα» για την ανάπτυξη των κβαντικών νευρωνικών δικτύων με πρόθεση για την γενίκευση της ιδέας των νευρωνικών δικτύων ειδικών κλάσεων (class specific neural networks). Ονομαστικά μπορούμε να γενικεύσουμε την ιδέα στον υπέρτατο βαθμό, αν σκεφτούμε ένα σύστημα ξεχωριστών νευρωνικών δικτύων, κάθε ένα από τα οποία να έχει εκπαιδευτεί σε μόνο ένα συγκεκριμένο πρότυπο! Αυτό με την σειρά του θα μας έφερνε στην προσέγγιση των πολλαπλών κόσμων (many universes approach) της θεωρίας της κβαντομηχανικής (Ezhov & Ventura, 2000).

1.4.3 Πολλαπλότητα και κβαντικά νευρωνικά δίκτυα

«Γιατί άραγε για την απομνημόνευση ενός αριθμού προτύπων δεν χρησιμοποιούμε ένα αριθμό πολλαπλών δικτύων Hopfield, κάθε ένα από τα οποία αποθηκεύει ένα μόνο πρότυπο...?» αναρωτιούνται οι Ezhov και Ventura. Σε ένα κλασικό μοντέλο Hopfield χρησιμοποιούμε τυπικά μόνο ένα δίκτυο για την αποθήκευση πολλών προτύπων και αθροίζουμε όλους τους βαθμούς συσχέτισης των προτύπων με σκοπό να κατασκευάσουμε της Χεμπιανές διασυνδέσεις του δικτύου ως ακολούθως (Ezhov & Ventura, 2000).

$$T_{ij} = \sum_{S=1}^P \sigma_i^S \sigma_j^S, T_{ij} = 0 \quad i, j = 1, \dots, N, \quad \text{Εξίσωση 5}$$

Το παραπάνω άθροισμα δύναται να προκαλέσει πολλαπλά προβλήματα αν θεωρήσουμε το νευρωνικό δίκτυο ως ένα παθητικό σύστημα απομνημόνευσης. Η παρεμβολή των διαφορετικών προτύπων οδηγεί στην έλλειψη της σταθερότητας για μερικές μνήμες (αντί αυτών παράγονται ψεύτικες μνήμες) και ως αποτέλεσμα υπάρχει περιορισμένη χωρητικότητα, η οποία μεγαλώνει στην καλύτερη περίπτωση μόνο γραμμικά ως προς τον αριθμό των νευρώνων (Müller, et al., 1995).

Εάν από την άλλη πλευρά απλώς παράγουμε πολλαπλά δίκτυα Hopfield, όπου σε κάθε ένα αποθηκεύουμε μόνο ένα πρότυπο, θα χάναμε τον οποιοδήποτε

παραλληλισμό στην επεξεργασία της πληροφορίας. Ισχύει όμως το ίδιο σε μια κβαντική προσέγγιση? Ας φανταστούμε ότι μπορούμε να αποθηκεύσουμε όλα τα πρότυπα με την κβαντική υπέρθεση (quantum superposition).

$$|\psi\rangle = \sum_{s=1}^P |\sigma_1^s \sigma_2^s \dots \sigma_N^s\rangle, \quad \text{Εξίσωση 6}$$

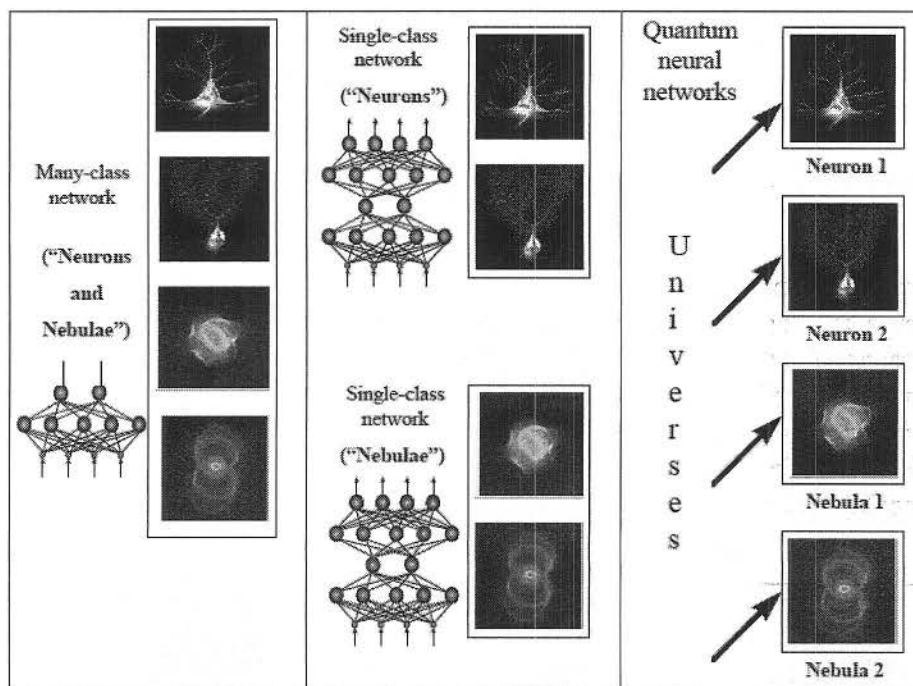
Σε αυτή την περίπτωση κάθε ένα από τα πρότυπα μπορεί να θεωρηθεί ως υπάρχον σε ένα ξεχωριστό κόσμο. Επιπλέον, η αλληλεπίδραση μιας τέτοιας υπέρθεσης (ή αλλιώς της πολλαπλής ύπαρξης σε πολλά μέρη ταυτόχρονα) με το περιβάλλον πραγματοποιείται παράλληλα, και μάλιστα αυτός ο παραλληλισμός είναι κβαντικής φύσεως. Όπως έχει αποδειχτεί, σε ένα δοσμένο αριθμό προτύπων μια τέτοια υπέρθεση είναι δυνατόν να δημιουργηθεί σύμφωνα με την μελέτη των Ventura και Martinez. Κάθε μια από τις συνθήκες βάσης (basis states) στην υπέρθεση θα παίξει τον ρόλο μιας κατάστασης μνήμης, ανεξάρτητα από τον αριθμό τους που υπάρχουν σε υπέρθεση (Ventura & Martinez, 1999).

Σε θεωρητικό επίπεδο έχει αποδειχτεί ότι μία κβαντική συνειρμική μνήμη δύναται να έχει χωρητικότητα μνήμης που αυξάνει με εκθετικό ρυθμό (Ventura & Martinez, 1998).

Κατά τους Ezhon και Ventura οι ψεύτικες καταστάσεις (μνήμης) μπορούν να προκύψουν σε μία τέτοιου είδους κβαντική μνήμη όπως περιγράφηκε προηγουμένως. Αυτές οι ψεύτικες καταστάσεις δεν είναι το αποτέλεσμα της παρεμβολής των μνημών όπως στις κλασσικές περιπτώσεις αλλά εγείρονται για τελείως διαφορετικό λόγο κατά την φάση της ανάκτησης και για αυτό τον λόγο δεν επηρεάζουν απ' ευθείας τα αποθηκευμένα πρότυπα.

Ας φανταστούμε πως στην θέση των διάφορων καταστάσεων μνημών έχουμε την ύπαρξη παράλληλων κόσμων, όπου υπάρχουν νευρωνικά δίκτυα τύπου Hopfield και απλής μνήμης. Στα κλασσικά δίκτυα Hopfield, η συνύπαρξη είναι συμμετρική, οι Χεμπιανές διασυνδέσεις εγγυούνται την σταθερότητα ενός μοναδικού αποθηκευμένου προτύπου. Παρομοίως σε ένα κβαντικό ανάλογο ενός δικτύου Hopfield η ακεραιότητα του αποθηκευμένου προτύπου (κατάσταση βάσης) οφείλεται στην εμπλοκή (ή αλλιώς μπερδεμα-entanglement-). Αυτή η ιδιότητα χαρακτηρίζει πολύ-μοριακά συστήματα (multi-particle systems) και αποτελεί την βάση για όλους τους γνωστούς κβαντικούς αλγόριθμους. Σε αυτό το σημείο, μπορούμε να σκεφτούμε την κβαντική συνειρμική μνήμη ως την υλοποίηση της ακραίας συνθήκης χρησιμοποίησης πολλαπλών δικτύων Hopfield, όπου σε κάθε ένα αποθηκεύεται ένα μόνο πρότυπο σε παράλληλους κβαντικούς κόσμους (Milburn, 1998).

Συνεχίζοντας τον παραπάνω συλλογισμό, μπορούμε να φανταστούμε περαιτέρω ακόμα πιο πολύπλοκες νευρωνικές δομές να υπάρχουν σε τέτοιους παράλληλους κόσμους. Αυτή η άποψη εξερευνήθηκε από τους Menneer και Narayanan οι οποίοι δοκίμασαν ένα πλέγμα από πολύ-επίπεδους αισθητήρες (perceptrons), όπου κάθε ένας εκπαιδεύτηκε σε ένα μόνο πρότυπο, και στο σύνολο τους συνδυάζονταν σε ένα νευρωνικό δίκτυο, του οποίου τα βάρη είναι υπερθέσεις των βαρών όλων των αισθητήρων που υπάρχουν σε παράλληλους κόσμους (Menneer & Narayanan, 1995).



Εικόνα 17. Κβαντικά νευρωνικά δίκτυα κατά Ezhon και Ventura. Σε αυτήν την εικόνα βλέπουμε: α) Αριστερά. Δίκτυα πολλαπλών κλάσεων να εκπαιδεύονται χρησιμοποιώντας παραδείγματα από διαφορετικές κλάσεις [εδώ “νευρώνες” (Neurons) και “νεφελώματα” (Nebulae) μαζί]. β) Κέντρο. Ένα πλέγμα από νευρωνικά δίκτυα μονής κλάσης που χρησιμοποιούνται για την εκπαίδευση μόνο των αντικειμένων που ανήκουν σε μία κλάση (δύο νευρωνικά δίκτυα για δύο κλάσεις: “Νευρώνες” και “Νεφελώματα” βρίσκονται ξεχωριστά). γ) Δεξιά. Κβαντικά νευρωνικά δίκτυα μπορούν να εκπαιδευτούν χρησιμοποιώντας μόνο ένα πρότυπο το καθένα! (τέσσερα δίκτυα για τέσσερα διαφορετικά παραδείγματα σε πολλαπλούς κόσμους)

Η έρευνα αποφαινόταν πως η προσέγγιση των πολλαπλών κόσμων στα κβαντικά νευρωνικά δίκτυα είναι ορθή μεθοδολογικά και νοητικά πιθανή. Η γρήγορη εκμάθηση των δικτύων σε επίπεδο ξεχωριστών κόσμων διαχωρίζεται από την θεώρηση ότι τα κλασσικά νευρωνικά μοντέλα είναι ακριβή αντίγραφα του ανθρώπινου εγκεφάλου καθώς οι πολλαπλές αναπαραστάσεις των προτύπων είναι αδύνατες για την ανθρώπινη μάθηση.

1.4.4 Κβαντική συνειρμική μνήμη και εφαρμογή κβαντικών νευρωνικών δικτύων

Πως μπορούν τα κβαντικά νευρωνικά δίκτυα να εφαρμοστούν ως πραγματικές φυσικές συσκευές? Αρχικά ως αναφέρουμε περιληπτικά μερικές από τις δυσκολίες που πιθανόν να αντιμετωπίσουμε στην ανάπτυξη μιας φυσικής εφαρμογής ενός κβαντικού νευρωνικού δικτύου.

- Συνεκτικότητα. Ένα από τα πιο δύσκολα προβλήματα κατά την ανάπτυξη ενός οποιοδήποτε κβαντικού υπολογιστικού συστήματος είναι η διατήρηση της συνεκτικότητας μέχρι την ολοκλήρωση του υπολογισμού. Η παρουσία μη-συνεκτικότητας οφείλεται στην αντίδραση ενός κβαντικού συστήματος με το περιβάλλον του. Στην κβαντική κρυπτογραφία αυτό το πρόβλημα μπορεί να λυθεί χρησιμοποιώντας κώδικες διόρθωσης σφαλμάτων (Grushka, 1999).

Τι συμβαίνει όμως με τα κβαντικά νευρωνικά δίκτυα? Έχει υποστηριχθεί από την επιστημονική κοινότητα, πως στην πραγματικότητα τέτοιου είδους συστήματα μπορούν να εφαρμοστούν προτού οι κλασσικοί κβαντικοί υπολογιστές γίνουν πραγματικότητα. Αυτό οφείλεται στην σαφώς μικρότερη απαιτούμενη ποσότητα κβαντικών μπιτ που απαιτούνται για την αναπαράσταση των νευρωνικών συνάψεων και στον επίσης σχετικά μικρό αριθμό μετασχηματισμών κατάστασης που απαιτούνται κατά την επεξεργασία των δεδομένων με σκοπό την εκτέλεση χρήσιμων υπολογισμών (Ventura & Martinez, 1998) (Cutting, 1999). Μια διαφορετική προσέγγιση στο πρόβλημα της μη-συνεκτικότητας στην κβαντική παράλληλη καταναμημένη επεξεργασία που προτάθηκε από τον Chrisley αποκλείει την χρήση των καταστάσεων υπέρθεσης (superpositional states) και προτείνει την χρήση κβαντικών συστημάτων για την εφαρμογή κλασσικών νευρωνικών παραδειγμάτων όπως για παράδειγμα ένα πολύ-επίπεδο νευρωνικό σύστημα εκπαιδευμένο με την μέθοδο της οπισθοδιάδοσης. Αυτό το είδος του μοντέλου ωστόσο δεν λαμβάνει υπόψη τα πλεονεκτήματα της χρήσης του κβαντικού παραλληλισμού. Μια πιο υποσχόμενη προσέγγιση στην εφαρμογή της κβαντικής συνειρμικής μνήμης (quantum associative memory) βασίζεται στην χρήση του αλγόριθμου Grover (Chrisley, 1997).

- Επαφές. Η υψηλή πυκνότητα των διασυνδέσεων μεταξύ των εκτελούμενων στοιχείων είναι μια βασική δυσκολία στην εφαρμογή μιας μικρούς κλίμακας ενσωμάτωσης στα υπολογιστικά συστήματα. Γενικά οι νευρο-υπολογιστές αναπαριστούν αυτές τις επαφές μέσω καλωδίων. Στους μη-υπερθεσιακούς (non-superpositional) κβαντικούς νευρο-υπολογιστές αναπαρίστανται μέσω δυνάμεων. Στο μοντέλο της κβαντικής συνειρμικής μνήμης που αναλύεται σε αυτό το κεφάλαιο αυτές οι επαφές οφείλονται στην εμπλοκή των κβαντικών μπιτ.

- Φυσικά Συστήματα. Σε αυτό το σημείο μπορούμε να περιγράψουμε το είδος των φυσικών συστημάτων που μπορούν να χρησιμοποιηθούν για να αναπτύξουν πραγματικά κβαντικά νευρωνικά δίκτυα καθώς και τον τρόπο με τον οποίο μπορούν αυτά τα συστήματα να αντιμετωπίσουν τα προβλήματα που αναφέρθηκαν προηγουμένως.
- Πυρηνική Μαγνητική Αντήχηση (Nuclear Magnetic Resonance. Μια ενδιαφέρουσα προσέγγιση στην εφαρμογή του μοντέλου της κβαντικής συνειρμικής μνήμης που βασίζεται στην χρήση του αλγόριθμου Grover και παρέχεται με μια διογκούμενη περιστροφή υπολογισμού αντήχησης (bulk spin resonance computation). Αυτή η τεχνική μπορεί να πραγματοποιηθεί χρησιμοποιώντας συστήματα Πυρηνικής Μαγνητικής Αντήχησης των οποίων η συνεκτικότητα έχει παρατηρηθεί και διαπιστωθεί από την επιστημονική κοινότητα. Η πειραματική επαλήθευση μιας τέτοιας εφαρμογής έχει πραγματοποιηθεί από τους Gershenfeld και Chuang το 1996, οι οποίοι χρησιμοποίησαν τεχνικές πυρηνικής μαγνητικής αντήχησης και διάλυμα μορίων χλωροφορμίου (CHCl_3) για την εφαρμογή του αλγόριθμου Grover σε ένα σύστημα που βασιζόταν σε δύο κβαντικά μπιτ (Gershenfeld & Chuang, 1996).
 - Το πρώτο κβαντικό μπιτ περιέγραφε την περιστροφή του πυρήνα του ισοτόπου C^{13} .
 - Ενώ το δεύτερο περιέγραφε την περιστροφή του πρωτονίου (πυρήνας υδρογόνου).

Αυτή η προσέγγιση για τον τομέα του κβαντικού υπολογισμού δεν αξιοποιούσε ένα απλό κβαντικό σύστημα αλλά περισσότερο το στατιστικό μέσο όρο πολλών αντιγράφων ενός τέτοιου συστήματος (μια συλλογή μορίων δηλαδή). Είναι ακριβές για τον συγκεκριμένο λόγο, πως η διατήρηση της συνεκτικότητας του συστήματος είναι αξιόλογα μεγαλύτερη από τις πραγματικές κβαντικές εφαρμογές. Επιπρόσθετα αυτή η τεχνολογία είναι σχετικά ώριμη και αποδεδειγμένα ο συνεκτικός υπολογισμός (coherent computation) σε 7 κβαντικά μπιτ χρησιμοποιώντας την μέθοδο της πυρηνικής μαγνητικής αντήχησης έχει παρουσιασθεί στην επιστημονική κοινότητα από τον Knill και τους συνεργάτες του το 1999 (Knill, et al., 1999). Αυτή η τεχνική φαντάζει ως η πιο πολλά υποσχόμενη στο εγγύς μέλλον κατά τους Ezhov και Ventura.

- Κβαντικά σημεία. Αυτού του είδους τα κβαντικά συστήματα αποτελούνται κυρίως από ένα ηλεκτρόνιο παγιδευμένο μέσα σε ένα κλωβό ατόμων (Behrman, et al., 1996), (Behrman, et al., 1999). Αυτά τα ηλεκτρόνια μπορούν να επηρεαστούν από μικρούς

παλμούς laser . Οι περιορισμοί σε αυτού του είδους τα συστήματα που θα πρέπει να ξεπεραστούν συμπεριλαμβάνουν:

- Μικρές περιόδους μη-συνεκτικότητας. Η μικρο-αστάθεια αυτή, οφείλεται στην ύπαρξη του ηλεκτρονίου σε κατάσταση διέγερσης για περίπου ένα microsecond, ενώ η απαιτούμενη διάρκεια ενός παλμού laser είναι περίπου ένα nanosecond.
- Η αναγκαιότητα για την ανάπτυξη εξειδικευμένης τεχνολογίας για την κατασκευή υπολογιστών από κβαντικά σημεία πολύ μικρής κλίμακας (10 ατόμων σε πλάτος).
- Η αναγκαιότητα για την ανάπτυξη ειδικών τύπων laser ικανών για την επιλεκτική επιρροή διαφορετικών ομάδων κβαντικών σημείων, με διαφορετικά μήκη κύματος της δέσμης φωτός.

Η χρήση κβαντικών σημείων ως βάση για την εφαρμογή των κβαντικών νευρωνικών δικτύων ερευνάται από τον Behrman και τους συνεργάτες του από το 1996.

- Άλλα συστήματα. Υπάρχουν μερικά ακόμα φυσικά συστήματα που μπορούν να θεωρηθούν ικανά για την δημιουργία κβαντικών υπολογιστών και συνεπώς κβαντικών νευρο-υπολογιστών (Brooks, 1999). Μερικά από αυτά είναι:
 - Διάφοροι σχηματισμοί κβαντο-ηλεκτροδυναμικών κοιλοτήτων (QED), δηλαδή κβαντική ηλεκτροδυναμική των ατόμων σε οπτικές κοιλότητες.
 - Παγίδες ιόντων.
 - Υπεραγώγιμες συσκευές κβαντικής παρεμβολής (SQUIDS).

Κάθε μια από τις παραπάνω τεχνικές έχει τα δικά της πλεονεκτήματα και τα μειονεκτήματα που δύναται να επηρεάζουν, ανάλογα με το είδος, τις στιγμές μη-συνεκτικότητας, την ταχύτητα, τις δυνατότητες μικροσκοπικοποίησης κλπ. Περισσότερες πληροφορίες για αυτές τις τεχνολογίες μπορούν να βρεθούν σε βιβλία των Milburn και Brooks που υπάρχουν στην βιβλιακή αναφορά της παραγράφου.

1.4.5 Ο υπερκερασμός των κβαντικών νευρωνικών δικτύων

Το κατά πόσο είναι δυνατό τα κβαντικά νευρωνικά δίκτυα να υπερκεράσουν τις δυνατότητες των κλασικών νευρωνικών δικτύων είναι ένα ερώτημα που απασχόλησε τους επιστήμονες του τομέα. Είναι σχεδόν ακριβώς το ίδιο ερώτημα που προέκυψε για την διάκριση των κλασικών και των κβαντικών υπολογιστών. Για παράδειγμα ο αλγόριθμος του Shor δίνει μια πολυωνυμική λύση για το πρόβλημα του prime factorization σε κβαντικό επίπεδο, μια λύση που πιθανολογείται ότι δεν λύνεται

με κλασσικό υπολογιστή (Shor, 1997). Υπάρχουν πολλά παραδείγματα που μπορεί να λύσει ένας κβαντικός υπολογιστής ενώ παράλληλα δεν μπορεί ένας κλασσικός υπολογιστής.

Τι γίνεται όμως στο επίπεδο των κβαντικών νευρωνικών δικτύων? Τα κβαντικά νευρωνικά δίκτυα μας παρέχουν πλεονεκτήματα τόσο έναντι των κλασσικών υπολογιστικών μηχανών που εισήγαγε ο von Neumann, όσο και των κλασσικών τεχνητών νευρωνικών δικτύων. Τα πλεονεκτήματα των κβαντικών νευρωνικών δικτύων δημιουργούν εξαιρετικές προσδοκίες για την ανάπτυξη τους. Οι μελλοντικές δυνατότητες τους ίσως είναι ακόμα πιο εξαιρετικές καθώς η ίδια η επιστήμη των κβαντικών υπολογιστικών μηχανών αναπτύσσεται στις μέρες μας. Συγκεντρωτικά τα πλεονεκτήματα παρατίθενται στον παρακάτω πίνακα:

Πίνακας 4. Πλεονεκτήματα διαφόρων εκδόσεων κβαντικών νευρωνικών δικτύων

ΠΛΕΟΝΕΚΤΗΜΑ	ΕΙΣΗΓΗΣΗ
Απουσία καλωδίων	Behrman, E.C., Steck, J.E., and Skinner, S.R. (1999)
Υψηλότερη σταθερότητα και αξιοπιστία	Cutting, D.(1999)
Υψηλή ταχύτητα επεξεργασίας (10^{10} bits/s)	Behrman, E.C., Steck, J.E., and Skinner, S.R. (1999)
Μικρό μέγεθος(10^{11} νευρώνες/ mm^3)	Behrman, E.C., Steck, J.E., and Skinner, S.R. (1999)
Ταχύτερη εκμάθηση νευρώνων	Menneer, T. and Narayanan, A. (1995)
Ταχύτερη απόδοση για μικρότερο αριθμό «κρυμμένων» νευρώνων	Cutting, D.(1999)
Επίλυση μονοεπίπεδου δικτύου για γραμμικά προβλήματα.	Menneer, T. and Narayanan, A. (1995)
Εξάλειψη της μη-ενθύμησης	Menneer, T. and Narayanan, A. (1995)
Εκθετική χωρητικότητα μνήμης	Ventura, D. and Martinez, T. (1998)

1.4.6 Ο μιμητισμός των κβαντικών νευρωνικών δικτύων

Ως (μέχρι τώρα...) γνωστόν το δεξί ημισφαίριο του ανθρώπινου εγκεφάλου είναι υπεύθυνο για λειτουργίες όπως spatial orientation, intuition, semantics κλπ. Ενώ

το αριστερό ημισφαίριο είναι υπεύθυνο για την προσωρινή επεξεργασία των δεδομένων, την λογική σκέψη και την σύνταξη. Βάση αυτής της οπτικής γωνίας είναι πολύ φυσικό να θεωρήσουμε ότι οι κλασσικοί υπολογιστές μιμούνται την λειτουργία του αριστερού ημισφαιρίου και οι νευροϋπολογιστές μιμούνται την λειτουργία του δεξιού ημισφαιρίου. Ο Penrose αξιοπρόσεκτα χαρακτηρίζει αυτούς τους δύο τύπους υπολογιστών ως «πάνω προς τα κάτω» και «κάτω προ τα πάνω» (Penrose, 1994). Ωστόσο υποστήριξε ότι υψηλότερες εγκεφαλικές λειτουργίες όπως η συνείδηση δεν μπορούν να προσομοιωθούν χρησιμοποιώντας κάποιον από αυτούς τους τύπους. Οι εισηγήσεις που παρουσιάζονται σε αυτό το κεφάλαιο παρουσιάζουν τις δυνατότητες του συνδυασμού των μοναδικών υπολογιστικών δυνατοτήτων των κλασσικών νευρωνικών δικτύων και των κβαντικών υπολογιστών. Το αποτέλεσμα είναι η παραγωγή υπολογιστικών παραδειγμάτων εξέχουσας προοπτικής. Κατά τους Alexandr A. Ezhov και Dan Ventura δεν υπάρχει κάποια συσχέτιση μεταξύ των βιολογικών νευρωνικών δικτύων και των κβαντικών νευρωνικών δικτύων τουλάχιστον στο πεδίο της ανθρώπινης διαίσθησης.

Ωστόσο ο Perus υποστήριξε ότι τα νευρωνικά δίκτυα μπορούν να είναι μια «μακροσκοπική αντιγραφή των κβαντικών επεξεργαστικών δομών» (Perus, 1997). Έτσι τα νευρωνικά δίκτυα θα μπορούσαν να είναι «μια επιφάνεια μεταξύ του μακρόκοσμου του ανθρώπινου περιβάλλοντος και του μικρό-κοσμου της μη τοπικής διαίσθησης του» (“neural networks can be a macroscopic replica of quantum processing structures. Neural networks could be an interface between the macro-world of man’s environment and the micro-world of his non-local consciousness”).

Όπως διαφαίνεται δεν είναι σχετικά απίθανο μελλοντικά μοντέλα κβαντικών νευρωνικών δικτύων να μπορούν να έρθουν ένα ακόμα βήμα πιο κοντά στις πιο «σκοτεινές» και απόμακρες περιοχές της ανθρώπινης σκέψης και του εγκεφάλου, προσομοιώνοντας λειτουργίες που θεωρούνταν απόμακρες για τα κλασσικά πρότυπα υπολογιστών.

Μερικοί επιστήμονες πιστεύουν ότι τα κβαντικά νευρωνικά δίκτυα ίσως αναπτυχθούν σε σημείο έτσι ώστε να υπερκεράσουν τις δυσκολίες που περιγράφονται στην θεωρία Church-Turing (Church-Turing thesis) (Cutting, 1999). Σύμφωνα με την οποία όλοι οι υπάρχοντες υπολογιστές έχουν ίδια υπολογιστική ισχύ με την μηχανή-Turing (Universal Turing Machine). Επίσης όλες οι αλγοριθμικές διαδικασίες που διεξάγονται στον ανθρώπινο εγκέφαλο μπορούν να πραγματοποιηθούν με αυτή την μηχανή και αντίθετα. Κανένας υπάρχων νευρο-υπολογιστής, ούτε κάποιος κβαντικός υπολογιστής θεωρητικά δεν μπορεί να ξεφύγει από την αρχή Church-Turing. Ισχύει το ίδιο όμως και για τα κβαντικά νευρωνικά δίκτυα; Το 1999 ο Dan Cutting αναρωτήθηκε εάν τα κβαντικά νευρωνικά θα αμφισβητήσουν τις αρχές-«σταθερές» των Church-Turing, για τα υπάρχοντα μοντέλα κβαντικών νευρωνικών δικτύων η απάντηση σχεδόν σίγουρα είναι «όχι», αλλά μερικά φυσικά συστήματα όπως οι «σκουληκότρυπες» πιθανολογείται ότι ίσως αποτελέσουν την βάση για να καταφέρουν τα κβαντικά νευρωνικά δίκτυα να υπερκεράσουν τα όρια (Cutting, 1999).

Οι μελλοντικές έρευνες στα κβαντικά νευρωνικά δίκτυα ίσως καταφέρουν να δείξουν ότι κανένα σημείο του ανθρώπινου εγκεφάλου ακόμα και αυτό της διαίσθησης

δεν είναι αδύνατο να προσεγγιστεί. Άραγε θα καταφέρουν τα κβαντικά νευρωνικά δίκτυα να προσθέσουν το τελευταίο λιθαράκι στις συνεχές τάσεις του ανθρώπου για δημιουργία υπολογιστικών μηχανών με τον μέγιστο μιμητισμό προς τον ανθρώπινο εγκέφαλο στο διάβα του χρόνου...;

1.5 Εφαρμογές τεχνητών νευρωνικών δικτύων

Οι εφαρμογές των τεχνητών νευρωνικών κατηγοριοποιούνται στην πρόβλεψη, στην μοντελοποίηση, στον διαγνωστικό έλεγχο και εξόρυξη δεδομένων.

Εφαρμογές διαγνωστικού ελέγχου χρησιμοποιούνται στις ιατρικές διαγνώσεις, στην αναγνώριση φωνής, σε μηχανισμούς ελέγχου-σύγκρισης υπογραφών, αναγνώρισης προσώπου κ.α.

Εφαρμογές πρόβλεψης έχουν χρήση στην πρόγνωση του καιρού, στην επίδραση του κλίματος στις αγροτικές καλλιέργειες, στην πρόβλεψη κίνησης τιμών χρηματιστηρίου, στην πρόβλεψη αναγκών ενέργειας στην πρόβλεψη κινήσεων αποθηκευτικών αποθεμάτων κ.α.

Οι εφαρμογές μοντελοποίησης ίσως η πλέον σημαντική εφαρμογή των τεχνητών νευρωνικών δικτύων χρησιμοποιούνται στην ρομποτική, στον έλεγχο συστημάτων, στις προσομοιώσεις στην αναπαράσταση ελέγχου διαδικασιών κ.α.

Η εξόρυξη δεδομένων αποτελεί την τελευταία εξέλιξη στις εφαρμογές των τεχνητών νευρωνικών δικτύων. Με την έννοια εξόρυξη δεδομένων εννοούμε την προσπάθεια εξαγωγής συμπερασμάτων και γνώσεων γενικότερα μέσα από σειρά απλών πληροφοριών. Χρησιμοποιείται κυρίως για την διεύρυνση σχέσεων μεταξύ μιας βάσης δεδομένων. Για παράδειγμα μια εταιρία που έχει μια βάση δεδομένων με τις αγοραστικές συνήθειες των πελατών θέλει να εξάγει ένα συμπέρασμα για την πορεία ενός προϊόντος στους καταναλωτές που δεν τους ενδιέφερε αρχικά η χρήση του.

1.6 Ο διεπιστημονικός χαρακτήρας των νευρωνικών δικτύων

Τα νευρωνικά δίκτυα μπορούν να αποτελούν υποκεφάλαιο του τομέα της τεχνητής νοημοσύνης και του γενικότερου τομέα των μαθηματικών και της θεωρίας πληροφοριών, ωστόσο έχουν βρει ευρεία εφαρμογή σε πάρα πολλούς τομείς εμπορικούς και μη. Παρακάτω θα γίνει μια προσπάθεια αναφοράς των συσχετίσεων καθώς σχετικών εφαρμογών.

1.6.1 Νευρωνικά δίκτυα και στατιστική φυσική

Ένας από αυτούς τους τομείς είναι και ο τομέας της στατιστικής φυσικής. Στην ουσία πρόκειται για νευρωνικά δίκτυα που έχουμε χρησιμοποιήσει ιδέες – τεχνικές της στατιστικής φυσικής για την εκπαίδευσή τους. Η πιο σύνηθες εφαρμογή τους ήταν τα προβλήματα βελτιστοποίησης. Ένα νέο χαρακτηριστικό διέπει αυτού του

είδους τα δίκτυα, πιο συγκεκριμένα μερικές φορές κατά την εκπαίδευση του δικτύου το σφάλμα μεγαλώνει αντί να ελαττώνεται κάτι το οποίο είναι απαραίτητο χαρακτηριστικό των τεχνικών της στατιστικής φυσικής. Αυτό αν και αρχικά φαίνεται ως σημαντικό μειονέκτημα για την ανάπτυξη και την εκπαίδευση τους αποδεικνύεται σε τελικό βαθμό ως πλεονέκτημα καθώς με αυτή την ιδιότητα τους αποφεύγουν το πρόβλημα των τοπικών ελαχίστων. Τα νευρωνικά δίκτυα που βασίζονται στην στατιστική φυσική μοιάζουν κυρίως στα δίκτυα τύπου Hopfield (Hopfield, 1982), συνεπώς απευθύνονται σε παρόμοια προβλήματα. Η γενική λειτουργία όμως είναι η ίδια, δηλαδή τα βάρη w αλλάζουν τις τιμές τους κατά την διάρκεια της εκπαίδευσης και στο τέλος το νευρωνικό δίκτυο δίνει το σωστό αποτέλεσμα για κάθε καινούργιο πρότυπο που παρουσιάζεται. Η επιστήμη της στατιστικής φυσικής εξετάζει τα φαινόμενα χρησιμοποιώντας την έννοια της τυχαιότητας. Συνήθως για την μελέτη των φαινομένων λαμβάνει μια σειρά μετρήσεων, όπου και έπειτα υπολογίζει την κατανομή, την μέση τιμή καθώς άλλες στατιστικές τιμές. Σκοπός είναι να αποδειχθεί ότι η ιδιότητα που μελετάται, είναι πολύ κοντά στην πραγματικότητα. Δηλαδή το πειραματικό μέρος των μετρήσεων να τείνει στο θεωρητικό. Συνήθως απαιτούνται πολλές πράξεις και για αυτό χρησιμοποιούνται υπολογιστές ειδικά για ορισμένες τεχνικές όπως η Monte Carlo τεχνική.

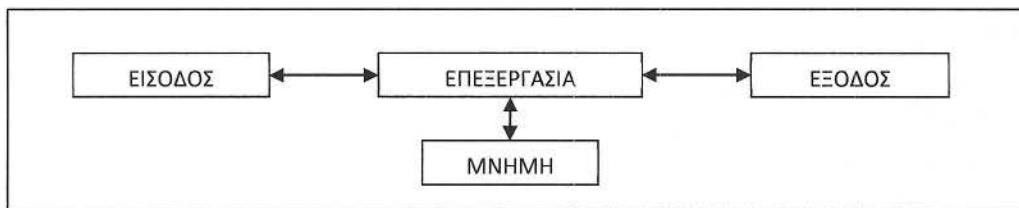
Στα νευρωνικά δίκτυα όπως έχουμε μάθει για τις ήδη υπάρχουσες τεχνικές χρησιμοποιούμε διάφορα μαθηματικά μοντέλα για την αλλαγή της τιμής των βαρών με βάση τα σφάλματα εξόδου. Οι στατιστικές μέθοδοι στα τεχνητά νευρωνικά δίκτυα αλλάζουν τα βάρη με τυχαίο τρόπο επιλέγοντας νούμερα από μια ομοιόμορφη κατανομή τυχαίων αριθμών και η τελική απόφαση για την τιμή των βαρών ακολουθούν τον αλγόριθμο που θα έχουμε δώσει στο νευρωνικό δίκτυο. Οι στατιστικές διαδικασίες για την περιγραφή ενός προβλήματος στην ουσία κάνουν μια προσομοίωση του πραγματικού προβλήματος.

Η μέθοδος που ακολουθούμε για να εκπαιδύσουμε ένα τεχνητό νευρωνικό δίκτυο με τις τεχνικές της στατιστικής φυσικής είναι η ακόλουθη: Αρχικά οι τιμές των βαρών αλλάζουν με τυχαίο τρόπο, έπειτα υπολογίζουμε το σφάλμα. Σε περίπτωση που η επόμενη τυχαία αλλαγή κάνει το σφάλμα μικρότερο το δίκτυο εκπαιδεύεται καλύτερα όποτε και διατηρούμε αυτήν την τυχαία αλλαγή. Ενώ αντίθετα αν το σφάλμα μεγαλώνει τότε η τυχαία αλλαγή απορρίπτεται και λαμβάνουμε νέα τυχαία

1.6.2 Τεχνητή νοημοσύνη και νευρωνικά δίκτυα

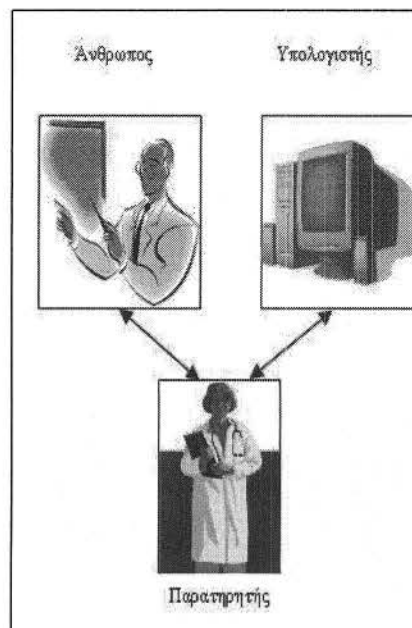
Η πρώτη θεωρητική διασύνδεση υπολογιστικών μηχανών και ανθρώπου προήλθε από τον Τζον Φον Νόμαν. Ο Φον Νόμαν θεώρησε ως πρότυπο για την κατασκευή ενός υπολογιστή τον ανθρώπινο εγκέφαλο. Στον Φον Νόμαν πιστώνεται το ακόλουθο διάγραμμα που αποτελεί την θεμελιώδη δομή του υπολογιστή.

Έτσι μέχρι το 1950 οι ερευνητές είχαν ήδη αρχίσει να σκέφτονται την σχέση μεταξύ της νοημοσύνης και της δομής του ανθρώπινου εγκεφάλου με τους υπολογιστές. Κανένας όμως δεν είχε θίξει το θέμα για το κατά πόσο μια μηχανή είχε την δυνατότητα να σκέφτεται.



Εικόνα 18. Θεμελιώδης διάγραμμα δομής υπολογιστή

Ο Άλαν Τιούρινγκ το 1950 ήταν ο πρώτος που ασχολήθηκε ουσιαστικά, παραδίδοντας μια εργασία σύμφωνα με την οποία έλεγχε μέσα από ένα πολύ απλό τεστ αν μία μηχανή μπορούσε να θεωρηθεί έξυπνη. Έστω τρία δωμάτια, όπου στο πρώτο υπάρχει ένας άνθρωπος, στο δεύτερο ένας υπολογιστής και στο τρίτο ένας άνθρωπος παρατηρητής ο οποίος δεν γνώριζε σε ποιο δωμάτιο βρίσκονταν ο υπολογιστής και σε ποιο ο άνθρωπος. Ο παρατηρητής έχοντας την δυνατότητα επικοινωνίας με τον άνθρωπο και τον υπολογιστή τους υποβάλλει ταυτόχρονα ερωτήσεις. Αν ο παρατηρητής δεν μπορεί να ξεχωρίσει τον συνομιλητή του, τότε ο υπολογιστής μπορεί να θεωρηθεί μία «έξυπνη μηχανή». Ο παρατηρητής υποβάλλει ερωτήσεις που ευνοούν είτε τον υπολογιστή (π.χ. λογικοί υπολογισμοί), είτε τον άνθρωπο (π.χ. άποψη για ένα έργο τέχνης).



Εικόνα 19. Το πείραμα του Τιούρινγκ

Το πείραμα αυτό ονομάστηκε από τον Τιούρινγκ «παιχνίδι μίμησης» και οριοθέτησε ουσιαστικά την νοημοσύνη των μηχανών. Το «παιχνίδι μίμησης»

αποτελέσει την απαρχή για την εξερεύνηση και την ανάπτυξη ενός νέου τομέα της πληροφορικής της τεχνητής νοημοσύνης.

Το έργο του Τιούρινγκ έκανε λόγο για όρια και όπως ήταν φυσικό τέθηκε θέμα αμφισβήτησης αυτών των ορίων από πολλούς επιστήμονες. Εδώ υπεισέρχεται η φιλοσοφία στον τομέα της πληροφορικής. Η αντίθετη με τον Τιούρινγκ άποψη σύνδεε την «σκέψη» με την «συνείδηση» και έθετε το ερώτημα αν ένας υπολογιστής είχε συνείδηση, η «συνείδηση» όμως αποτελεί κομμάτι της «ψυχής» του ανθρώπου. Επίσης η έννοια της σκέψης ενώ είναι κατανοητή είναι δύσκολο να οριοθετηθεί και να περιγραφεί μέσα από κανόνες καθώς παράγοντες όπως ο συνδυασμός των γεγονότων, των πληροφοριών, οι συνειρμοί αυθαίρετοι και μη καθώς και η διαίσθηση αποτελούν μέρος της σκέψης του ανθρώπου. Ο χαρακτηρισμός της «έξυπνης μηχανής» αποτελεί και αυτός σημείο αμφισβήτησης καθώς εφόσον εκτελεί πολυάριθμες αριθμητικές πράξεις σε ελάχιστο χρόνο σε σχέση με τον άνθρωπο τότε είναι έξυπνη.

Προσωπική άποψη του συγγραφέα του παρόντος είναι πως η παραπάνω λογική είναι τελείως εσφαλμένη, καθώς το γεγονός πως οι μηχανές μπορούν να κάνουν αριθμητικές πράξεις σε λιγότερο χρόνο σε σχέση με τον άνθρωπο είναι αυτονόητο άλλωστε αυτός είναι και ο λόγος δημιουργίας τους να μας αντικαταστήσουν στις δύσκολες εφαρμογές και να απλουστεύσουν την ζωή μας. Οι πρώτοι επιστήμονες στον τομέα της πληροφορικής όταν έκαναν λόγο για «έξυπνη» μηχανή οραματίζονταν τον υπολογιστή που θα ξεπερνούσε το πεδίο των υπολογισμών και θα έπαιρνε αποφάσεις. Στην πληροφορική η λέξη απόφαση έχει κατά βάση την άλγεβρα του Μπουλ(Boole). Δηλαδή την απάντηση σε ερωτήσεις λογικού τύπου ΝΑΙ/ΟΧΙ και γενικότερα επιλογής ανάμεσα σε δύο ή περισσότερες περιπτώσεις. Από αυτήν την έννοια οι υπολογιστές μπορούν να θεωρηθούν έξυπνοι καθώς σε ένα ερώτημα που τους τίθεται «σκέφτονται» με την έννοια ότι επιλέγουν την καλύτερη δυνατή και επιθυμητή επιλογή απαντήσεως με βάση κάποια κριτήρια και κάποιους παράγοντες. Η συνείδηση δεν πρέπει να συνδέεται με τον τομέα της τεχνητής νοημοσύνης καθώς στα πλαίσια της λογικής θα πρέπει να αναρωτηθούμε τι ζητάμε από τις μηχανές. Η συνείδηση είναι ένα «συναίσθημα» που έχει ο άνθρωπος πριν την λήψη μιας απόφασης. Το συναίσθημα αυτό επηρεάζει την απόφαση αυτή, επίσης συνείδηση για μία απάντηση έχουμε και για μερικά ερωτήματα που δεν γνωρίζουμε την απάντηση τους απλώς μερικά στοιχεία της μνήμης του ανθρώπινου εγκεφάλου και μερικοί άλλοι παράγοντες μας κάνουν να συγκλίνουμε προς μία συγκεκριμένη απάντηση. Είναι όμως γνωστό ότι η συνείδηση μας έχει οδηγήσει πολλές φορές σε λάθη. Έτσι τι ζητάμε από τον υπολογιστή να έχει «τεχνητή συνείδηση» που ίσως μας οδηγήσει σε λάθος αποτελέσματα; Η απάντηση είναι σαφέστατα όχι γιατί από μία μηχανή αναζητούμε το τέλειο, δεν γίνεται να δημιουργήσουμε κάτι το οποίο σε μια μη γνώριμη κατάσταση θα πάρει ίσως μια λάθος απόφαση, αν η κατάσταση είναι μη γνώριμη απλώς δεν θα πάρει απόφαση. Σε μια μη γνώριμη κατάσταση ο άνθρωπος ίσως πάρει μια απόφαση ίσως και όχι. Όταν μιλούμε για υπολογιστές όμως εδώ είναι το σημείο που εισέρχεται το κεφάλαιο του προγραμματισμού. Σε όσους πιο πολλούς τομείς και καταστάσεις έχουμε

προγραμματίζει τον υπολογιστή να παίρνει λογικές αποφάσεις τόσο πιο νοήμων είναι τεχνητά. Τα προγραμματιστικά όρια που βάζουμε σε μία μηχανή είναι το I.Q. του υπολογιστή. Ένα παράδειγμα είναι τα ρομπότ. Αν σε ένα ρομπότ το πρόγραμμα κίνησης του στον χώρο δεν συμπεριλάμβανε την αποφυγή εμποδίων τότε θα ήταν ένα χαζό ρομπότ, αντίθετα όσο πιο πολλές μεταβλητές του θέσουμε για την αποφυγή εμποδίων τόσο πιο έξυπνο θα είναι. Γίνεται λοιπόν σαφές γιατί θα πρέπει να διαχωρίζουμε την συνείδηση από την τεχνητή νοημοσύνη αν μερικοί επιστήμονες θέλουν να κάνουν μηχανές με συνείδηση ας ορίσουν τις αρχές της «τεχνητής συνείδησης» ως νέας επιστήμης στον τομέα της πληροφορικής και ας μην τις συγχέουν με την τεχνητή νοημοσύνη• οι υπολογιστικές μηχανές θέλουμε να σκέφτονται λογικά και όχι να λειτουργούν κατά συνείδηση.

Το 2000 έρευνες στα εργαστήρια της IBM (τα οποία είναι «υπεύθυνα» για 46.000 δηλωμένες ευρεσιτεχνίες) δημιούργησαν ένα ποντίκι το οποίο μέσα από μία σειρά με αισθητήρες που είχε ανίχνευε τις συσπάσεις των μυών του χεριού, την θερμοκρασία, την αρτηριακή πίεση, τον αριθμό παλμών της καρδιάς και την εφίδρωση. Όλα τα παραπάνω αποτελούσαν δεδομένα τα οποία συγκρίνονταν με προϋπάρχοντα μιας βάσης δεδομένων . Τα αποτελέσματα των συγκρίσεων χρησιμοποιούνταν για να καταλάβει ο υπολογιστής τα συναισθήματα του χρήστη και με την βοήθεια ενός προγράμματος-λογισμικού τεχνητής νοημοσύνης προσάρμοζε τις λειτουργίες του. Η επιτυχία της προσαρμογής της λειτουργίας του υπολογιστή ανάλογα με την συναισθηματική κατάσταση του χρήστη άγγιξε το 75%. Είναι εύκολα κατανοητό μέχρι πιο σημείο έχει φτάσει η «πάντα οριοθετημένη» νοημοσύνη των υπολογιστικών συστημάτων και πόσο κοντά είναι στην γενικότερη νοημοσύνη του ανθρώπου. Επίσης όταν ένα σχεδόν μεγάλο ποσοστό του κόσμου γύρω μας είναι μαθηματικά και μπορούν να περιγραφούν και συγκριθούν μέσα από μαθηματικές σχέσεις είναι πραγματικά πάρα πολύ λίγα αυτά που δεν μπορούμε να προγραμματίσουμε μια υπολογιστική μηχανή να κάνει.

Σκοπός αυτής της θεματικής ενότητας δεν είναι να μελετήσουμε την τεχνητή νοημοσύνη σε όλες τις πτυχές τις αλλά να κατανοήσουμε μερικές βασικές αρχές της και να μελετήσουμε την σχέση της τεχνητής νοημοσύνης με τα τεχνητά νευρωνικά δίκτυα.

Η Ιαπωνία το 1981 χρηματοδότησε με ένα αρκετά μεγάλο πόσο ένα ερευνητικό πλάνο που σκοπό είχε την ανάπτυξη των υπολογιστών 5ης γενιάς δηλαδή μηχανών τεχνητής νοημοσύνης για να καταστήσουν την Ιαπωνία παγκόσμια δύναμη στον τομέα παραγωγής ηλεκτρονικών υπολογιστών.

Η διαφορετική τακτική των Ιαπώνων ήταν ότι έδωσαν βάση στην κατασκευή δυνατού υλικού για την δημιουργία «έξυπνων» μηχανών σε σχέση με τις ερευνητικές ομάδες της δύσης που στόχος τους ήταν η ανάπτυξη όλο και δυνατότερου λογισμικού. Οι Ιάπωνες αυξάνοντας την επεξεργαστική ισχύ σε πολλά MFLOPS και εφαρμόζοντας μη συμβατικές αρχιτεκτονικές αντίθετες με την Φον Νόυμαν θεωρία.

Το σχέδιο των Ιαπώνων, το οποίο περιλάμβανε για την είσοδο δεδομένων την αναγνώριση φωνητικών εντολών, δεν στέφθηκε με επιτυχία αλλά η εξέλιξη και η μελέτη στον τομέα της ανάπτυξης υλικού ήταν απόλυτα θετική.

Οι Ιάπωνες επιστήμονες παρατήρησαν πως η συμβατική αρχιτεκτονική κατά Φον Νόουμαν δεν είχε την δυνατότητα να διαχειριστεί μεγάλο όγκο βάσεων δεδομένων που χρησιμοποιούνταν για την διαδικασία λήψης μιας απόφασης. Οι έρευνες τους κινήθηκαν στην παράλληλη επεξεργασία με την χρήση των μηχανών ροής δεδομένων. Η μηχανή ροής δεδομένων (dataflow machine) ήταν μία παράλληλη αρχιτεκτονική. Οι υπολογιστές ροής δεδομένων είναι σειρές μονάδων επεξεργασίας αποτελούμενες από πολλούς επεξεργαστές μικρότερων δυνατοτήτων, καθένας από τους οποίους είχε μια μικρού μεγέθους μνήμη. Η σύνδεση τους είναι περίπλοκη ενώ ρόλο συντονιστή-διακομιστή αναλαμβάνει ένας κεντρικός υπολογιστής. Όσον αφορά το πλαίσιο της λειτουργίας κάθε επεξεργαστής ανάλογα με την ισχύ και τις δυνατότητες του παίρνει το δικό του μερίδιο διαμοιρασμένης εργασίας. Κατά το αποτέλεσμα ένας επεξεργαστής συνθέτει τα αποτελέσματα του κάθε επεξεργαστή και εξάγει το τελικό αποτέλεσμα.

Η έρευνα του ιαπωνικού σχεδίου εκτός ότι προετοίμασε το έδαφος για την δημιουργία ισχυρών τσιπ εκατομμυρίων τρανζίστορ αλλά και συσκευών αναγνώρισης φωνής έφερε σε πολλή κοντινή απόσταση τα τεχνητά νευρωνικά δίκτυα και την τεχνητή νοημοσύνη.

Τόσα τα τεχνικά νευρωνικά δίκτυα όσο και η τεχνητή νοημοσύνη προσπαθούν να μιμηθούν στον μέγιστο δυνατό βαθμό την λειτουργία του ανθρώπινου εγκεφάλου για την επίλυση προβλημάτων.

Τα τεχνητά νευρωνικά δίκτυα όπως έχουμε αναφέρει προηγουμένως έχουν την δυνατότητα της αυτοεκπαίδευσης και έτσι υπερέχουν των συστημάτων τεχνητής νοημοσύνης. Ένα άλλο σημαντικό πλεονέκτημα των νευρωνικών δικτύων σε σχέση με την χρήση συστημάτων βασισμένων στην τεχνητή νοημοσύνη είναι η επεξεργασία των δεδομένων σε πραγματικό χρόνο. Οι εφαρμογές τεχνητής νοημοσύνης παρουσιάζουν αρκετά μειονεκτήματα όπως η χρονοβόρα διαδικασία για την επίλυση αρκετά πολύπλοκων αλγορίθμων, ο χρόνος επεξεργασίας με το υπάρχον υλικό είναι μεγαλύτερος των συστημάτων παράλληλης επεξεργασίας και ο τεράστιος όγκος διαχείρισης πληροφοριών. Οι εφαρμογές συστημάτων τεχνητής νοημοσύνης αδυνατούν να επιλύσουν προβλήματα γενικού τύπου καθώς παρουσιάζουν εξειδίκευση. Όπως βλέπουμε τα τεχνητά νευρωνικά δίκτυα παρουσιάζουν πολύ περισσότερα πλεονεκτήματα έναντι των εφαρμογών τεχνητής νοημοσύνης αλλά και των παραδοσιακών τύπων εφαρμογών λογισμικού. Ένα από αυτά τα πλεονεκτήματα έναντι του συμβατικού λογισμικού που έχουμε αναφέρει και προηγουμένως είναι η ανεκτικότητα στα σφάλματα (fault tolerant systems) μία βλάβη παραλύει μόνο έναν νευρώνα και όχι ολόκληρο το σύστημα.

1.6.3 Νευρωνικά δίκτυα και ιατρική

Η εξέλιξη στην επιστήμη της πληροφορική είναι σχεδόν παράλληλη με τον τομέα της ιατρικής. Τα τελευταία 50 χρόνια οι δύο επιστήμες αυξάνουν τους ορίζοντες των γνώσεων τους με γεωμετρική πρόοδο, η γενικότερη εξέλιξη της τεχνολογίας φέρνει σήμερα πολύ κοντά τις δύο αυτές επιστήμες δημιουργώντας την νέα επιστήμη

της βιο-ιατρικής. Πολλές εγχειρήσεις έχουν πραγματοποιηθεί με την βοήθεια ειδικών βραχιόνων ρομπότ που φέρουν ιατρικά εργαλεία στα άκρα τους και χειρίζονται εξ αποστάσεως εκπαιδευμένοι γιατροί με την βοήθεια ειδικών των ρομπότ-ιατρών.

Το μέλλον της ιατρικής συσχετίζεται άμεσα με τις τελευταίες εξελίξεις στον τομέα της πληροφορικής. Τέτοια εξέλιξη αποτελούν τα βιονικά ολοκληρωμένα κυκλώματα ή βιοτσιπ. Τα κυκλώματα αυτά δεν υποστηρίζονται από ημιαγωγούς πυριτίου αλλά από βιολογικό υλικό. Αρχικός στόχος κατασκευή τους αποτελεί η χρησιμοποίησή τους για την ταχύτατη διάγνωση ασθενειών που με τα συμβατικά μέσα θα αποτελούσε πολύ περισσότερο χρόνο.

Εννοείται φυσικά ότι μια ολοκληρωμένη και προχωρημένη παρουσίαση ενός τέτοιου τσιπ θα έφερνε επανάσταση στον τομέα της ιατρικής γι' αυτό και μεταξύ των εταιριών όπως η Hyseq, η Incyte, η Affymetrix, και η Vysis επικρατεί άκρα μυστικότητα. Ο τομέας της πληροφορικής που ασχολείται με τα βιοτσιπ ονομάζεται τομέας βιονικής πληροφορικής. Το 1999 στην Ελβετία έγιναν τα πρώτα επιτυχή πειράματα σε πρώιμο στάδιο βέβαια κατασκευής βιονικών ρομπότ και ήταν επιτυχή. Ένα βιονικό ρομπότ αποτελείται από βιονικούς επεξεργαστές, δηλαδή επεξεργαστές φτιαγμένους από βιολογικό υλικό. Το πρόγραμμα λειτουργίας εμπεριέχεται στο γενετικό υλικό και μπορεί να διαμοιραστεί σε όλα τα κύτταρα που βρίσκεται σε επαφή. Οι οδηγίες κάθε κυττάρου καθορίζονται από τον γενετικό κώδικα, με αυτόν τον τρόπο το βιονικό ρομπότ μπορεί να επουλώσει τις πληγές του και να αναπαραχθεί. Η βιονική πληροφορική βρίσκεται ακόμα στα σπάργανα και είναι νωρίς ακόμα και να την κατανοήσουμε λόγω του μυστικισμού που την περιβάλλει πόσο μάλλον να δούμε τις αρνητικές επιπτώσεις που μπορεί να έχει. Είναι όμως σίγουρο ότι αφήνει πολλές ελπίδες για τον άμεσο εντοπισμό των ασθενειών δύσκολων να εντοπιστούν με τα ήδη υπάρχοντα μέσα.

Δεν είναι όμως μόνο τα βιοτσιπ που δίνουν ενθαρρυντικά μηνύματα, σε μια πρόσφατη έρευνα επιστήμονες κατάφεραν με ειδικά εμφυτεύματα σε νευρώνες του αμφιβληστροειδούς να δώσουν μια ασπρόμαυρή όραση σε άτομα που μέχρι πρότινος είχαν χάσει το φως τους. Για την ακρίβεια ήταν μια μαύρο-γκρι απεικόνιση του κόσμου που βλέπουμε εμείς, στους ανθρώπους αλλά και στην φύση δεν μπορούσαν να δουν την ζωή με χρώματα αλλά έβλεπαν ένα γκρι περίγραμμα σε ένα μαύρο φόντο. Δεν μπορεί φυσικά να συγκριθεί με την εικόνα που βλέπουμε εμείς αλλά για κάποιον ο οποίος έχει χάσει το φως θα είναι σαν να ξαναγεννιέται καθώς αποφεύγει ουσιαστικά μια σειρά από κινδύνους όπως το πέρασμα των φαναριών και άλλα. Τα εμφυτεύματα αποτελούν μικροτσιπ που συνδέουν τους νευρώνες με ηλεκτρόδια με αποτέλεσμα να διεγείρουν νεκρές περιοχές νευρώνων. Θεωρητικά μοιάζει με την ανθρώπινη καρδιά που όταν σταματήσει της κάνουμε ηλεκτροσόκ για να αρχίσει να ξαναχτυπάει. Αυτή η έρευνα θεωρώ πως αποτελεί σταθμό στον τομέα της βιο-ιατρικής, καθώς ίσως αυτή αποτελέσει την απαρχή για την ίαση άλλων σοβαρών εγκεφαλικών ασθενειών αλλά και ασθενειών που σχετίζονται γενικότερα με το νευρικό σύστημα. Η εγκεφαλοχειρουργική δεν είχε την ίδια ανάπτυξη με άλλους κλάδους της ιατρικής έχει προχωρήσει με πολύ σταθερά βήματα. Ο λόγος που έμεινε πίσω είναι η λεπτότητα των επεμβάσεων και η τεράστια ευαισθησία του ανθρώπινου εγκεφάλου. Πάντως είναι

σχεδόν βέβαιο ότι επεμβάσεις εγκεφάλου θα γίνονται στο μέλλον μέσω των ειδικών βραχιόνων. Δυστυχώς όμως υπάρχουν μέρη του ανθρωπίνου εγκεφάλου που δεν μπορεί να τα αγγίξει νυστέρι ή άλλο χειρουργικό όργανο. Η λύση μπορεί να βρίσκεται στις σύγχρονες εξελίξεις στους νέους τομείς της πληροφορικής όπως είναι η νανοτεχνολογία, η ρομποτική, η βιονική πληροφορική κλπ. Στο μέλλον ίσως να μπορούμε με τα βιοτσιπ να εντοπίζουμε ταχύτατα τις εγκεφαλικές ασθένειες και μέσω ειδικών απειροστά μικρών ρομπότ (νανορομπότ) να προκαλούμε την ίαση μέσω ηλεκτροδιέγερσης των εγκεφαλικών νευρώνων ή εάν ακολουθήσουμε τις τελευταίες εξελίξεις στον τομέα της ιατρικής που κάνουν λόγω για ανάπτυξη νεκρών τμημάτων μέσω της εισαγωγής βλαστοκυττάρων θα μπορούσαμε να φανταστούμε νανορομπότ να μεταφέρουν βλαστοκύτταρα σε εγκεφαλικές περιοχές που δεν θα μπορούσαμε να αγγίξουμε και να προκαλέσουμε την ίαση των εγκεφαλικών ασθενειών. Είναι δυνατόν να γίνουν όλα αυτά θα μπορούσε να αναρωτηθεί κάποιος, αν δούμε τις εξελίξεις που υπάρχουν και στους δύο χώρους τα τελευταία χρόνια το μέλλον ίσως να μην είναι τόσο μακριά όσο φαίνεται.

Ο χώρος όμως της υγείας ωφελήθηκε πολύ και από την εξέλιξη των νευρωνικών δικτύων. Τα νευρωνικά δίκτυα εκπαιδεύτηκαν για να μπορέσουν να χρησιμοποιηθούν στην διάγνωση σημαντικών ασθενειών όπως εμφράγματος και καρκίνων του μαστού. Ένας άλλος τομέας που χρησιμοποιήθηκαν ήταν για την πρόβλεψη τιμών πιο αναλυτικά σκοπός τους ήταν ο προσδιορισμός τιμών σε διάφορους τομείς όπως την τερατογένεση από την προγεννητική χορήγηση φαρμάκων, την πρόγνωση των ασθενών που έχουν υποστεί υπαρεκτομή, την διάγνωση κινδύνου εμφάνισης κυτταρομεγαλοϊού σε ασθενείς που έχουν υποστεί μεταμόσχευση νεφρού κλπ. Τα νευρωνικά δίκτυα χρησιμοποιούνται ακόμα σε διάφορα συστήματα ανάλυσης και επεξεργασίας βιοσημάτων. Ειδική χρήση γίνεται για την ταξινόμηση των μονήρη πνευμονικών οζιδίων αλλά και των χρωμοσωμάτων ενώ μεγάλη κρίνεται η χρησιμότητα τους στην επεξεργασία εικόνων που μπορούν να προέρχονται από διάφορων ειδών ιατρικές εικόνες όπως τομογραφίες, μαγνητικές, ακτίνες X κλπ. (Miller, 1993), (Miller, et al., 1992), (Baxt, 1995).

Για την ακρίβεια το πλεονέκτημα των νευρωνικών δικτύων έναντι του συμβατικού προγραμματισμού έγκειται στην ικανότητα των νευρωνικών δικτύων να επιλύουν προβλήματα τα οποία δεν παρουσιάζουν αλγοριθμική επίλυση ή η λύση είναι περίπλοκο να βρεθεί.

1.6.4 Ανάλυση και Ερμηνεία σήματος & Νευρωνικά Δίκτυα

Στον τομέα της ιατρικής πληροφορικής μία από τις σημαντικότερες ενότητες είναι αυτή της ανάλυσης και ερμηνείας ιατρικού σήματος. Πολλοί επιστήμονες κατά καιρούς χρησιμοποίησαν τα νευρωνικά δίκτυα την επεξεργασία ανάλυση και τελικώς την ερμηνεία των ιατρικών σημάτων.

Ο Dokur και συνεργάτες χρησιμοποίησαν ένα δίκτυο Kohonen για την ανάλυση τεσσάρων ECG κυματομορφών (Sordo, 2002). Τα δεδομένα για την εκπαίδευση του δικτύου πάρθηκαν από την βάση δεδομένων του τμήματος Αρρυθμίας

του MIT (MIT/BIH Arrhythmia Database). Η βάση δεδομένων περιείχε περίπου 24 ώρες καταγεγραμμένων ECG κυματομορφών.

Το 1994 ο Jervis και συνεργάτες εκπαίδευσαν ένα πολυεπίπεδο δίκτυο αισθητήρα (perceptron) να διαχωρίζει (μέσω Contingent Negative Variation) τις κυματομορφές για ασθένειες Huntington, Parkinson και Σχιζοφρένειας (Jervis, et al., 1994). Χρησιμοποιήθηκαν δεδομένα από 47 ασθενείς (20 πάσχοντες από σχιζοφρένεια, 16 από Πάρκινσον και 11 από Huntington), επίσης 17 διαφορετικά CNV (Contingent Negative Variation) χαρακτηριστικά χρησιμοποιήθηκαν ως είσοδοι για το νευρωνικό δίκτυο. Τα αποτελέσματα ήταν αρκετά ελπιδοφόρα καθότι με επίπεδα ευαισθησίας μεγαλύτερα από 0.9 θεωρούνται κλινικώς χρήσιμα. Ενώ υπήρχε μεγάλη δυνατότητα βελτίωσης της ποιότητας των αποτελεσμάτων με την χρησιμοποίηση περισσότερων δεδομένων για την είσοδο.

Ο Sordo σε έρευνες από το 1999 μέχρι και το 2001 σχεδίασε ένα νευρωνικό δίκτυο για τον διαχωρισμό υπέρηχων του μαγνητικού φάσματος φωσφόρου (31P) μεταξύ υγιών και καρκινογόνων μαστών. Δεδομένα από 26 διαφορετικές περιπτώσεις δόθηκαν ως είσοδοι στο νευρωνικό δίκτυο (Sordo, et al., 2001). Η απριόρι γνώση μεταβολικών χαρακτηριστικών υγιών και καρκινογόνων μαστών ενσωματώθηκε στην κατασκευή του νευρωνικού δικτύου για τον συγκερασμό της έλλειψης δεδομένων. Η έρευνα δοκιμάστηκε σε δύο διαφορετικούς τύπους νευρωνικών δικτύων:

- Στα νευρωνικά δίκτυα γνώσης (Knowledge-Based Neural Networks [KBANN])
- Σε κοινά νευρωνικά δίκτυα χωρίς «γνώση» (Knowledge-free)

Η απόδοση ορθού διαχωρισμού για τα κοινά νευρωνικά δίκτυα χωρίς γνώση ήταν της τάξεως του 62,4% και ενώ για τα νευρωνικά δίκτυα γνώσης ήταν 87,36%. Όπως είναι η προφανής η έρευνα απέδειξε ότι την πολύ καλύτερη απόδοση των νευρωνικών δικτύων με γνώση έναντι αυτών χωρίς για τον υπέρηχο μαγνητικού φάσματος φωσφόρου. Επίσης η έρευνα αυτή απέδειξε ότι ο συνδυασμός εμπειρικών μεθόδων και αυστηρώς επιστημονικά ορισμένων τεχνικών έχει τελικώς πιο εύρωστα αποτελέσματα σε σχέση με την έλλειψη εμπειρικών στοιχείων.

Ο Waltrus και συνεργάτες παρουσίασαν στην επιστημονική κοινότητα αποτελέσματα από την εφαρμογή εργαλείων για την σύνθεση, βελτιστοποίηση και ανάλυση με νευρωνικά δίκτυα μονάδων ηλεκτροκαρδιογραφημάτων (ECG). Είσοδοι αποτελούσαν σειρές κανονικών και μη-κανονικών καρδιοχτύπων. Η ευαισθησία των αποτελεσμάτων απέδειξε ότι μερικά από αυτά τα «εργαλεία» μπορούσαν να μειώσουν το μέγεθος του αρχικού νευρωνικού δικτύου κατά 40% χωρίς την παραμικρή απώλεια στην ακρίβεια των αποτελεσμάτων (Waltrus, et al., 1993).

1.6.5 Ανάλυση και Ερμηνεία εικόνας & Νευρωνικά Δίκτυα

Ο τομέας της εικόνας είναι μια από τις σημαντικότερες επιστημονικές ερευνητικές περιοχές της ανάπτυξης των τεχνητών νευρωνικών δικτύων. Πιο συγκεκριμένα τα νευρωνικά δίκτυα χρησιμοποιούνται για την αναγνώριση προτύπων (Sordo, 2002). Ειδικά για τον τομέα της υγείας η αναγνώριση προτύπων ως τεχνική

χρησιμοποιείται ευρέως για τον προσδιορισμό και την «εξόρυξη» σημαντικών χαρακτηριστικών-δεδομένων, που τις περισσότερες φορές δεν είναι ικανή η παρατήρηση τους δια γυμνού οφθαλμού.

Ο Rajapakse και ο Acharya το 1990 δημιούργησαν ένα αυτο-ταξινομούμενο πολυεπίπεδο νευρωνικό δίκτυο προσαρμοσμένης αντήχησης (self-organizing network multilayer adaptive resonance architecture [MARA]) για την κατηγοριοποίηση CT εικόνων της καρδιάς (Rajapakse & Acharya, 1990). Παρομοίως ο Däschlein δημιούργησε ένα δυεπίπεδο νευρωνικό δίκτυο για την κατηγοριοποίηση CT εικόνων της κοιλιάς. Η μέθοδος απαιτούσε την διάκριση μεταξύ του νεφρού, του συκωτιού, των οστών αλλά και άλλων παθολογικών ανεπαρκειών όπως η πέτρα και όγκοι στα νεφρά (Däschlein, et al., 1994).

Ο Özkan και συνεργάτες το 1990 σε μία έρευνα τους εκπαίδευσαν νευρωνικά δίκτυα με την μέθοδο της οπισθοδιάδοσης (backpropagation) για τον κατακερματισμό και τον διαχωρισμό πολυφασματικών εικόνων μαγνητικών υπερήχων (Magnetic Resonance Imaging [MRI]) από υγιείς και νοσηρούς ανθρώπινους εγκέφαλους. Τα αποτελέσματα έδειξαν ότι ο οξύς (sharp segmentation) και συμπαγής (compact segmentation) κατακερματισμός μπορούσε να πραγματοποιηθεί με νευρωνικό δίκτυο μικρής αρχιτεκτονικής (Özkan, et al., 1990).

Ο Miller και συνεργάτες το 1992 εκπαίδευσαν διαφόρων ειδών νευρωνικά δίκτυα για την αναγνώριση «περιοχών ενδιαφέροντος» (regions of interest [ROIs]) αντίστοιχων συγκεκριμένων οργάνων με τομογραφίες ηλεκτρικής εμπέδησης (electrical impedance tomography images [EIT]) θώρακα. Το νευρωνικό δίκτυο μπορούσε αυτόματα να επιλέγει τα βέλτιστα pixel με βάση έναν αριθμό εικόνων μια περιόδου δειγματοληψίας, όπου κάθε pixel κατηγοριοποιείται ανάλογα το όργανο στο οποίο ανήκει. Τα πρώτα αποτελέσματα μιας τόσο δύσκολης για την εποχή έρευνας απέδειξαν ότι τα νευρωνικά δίκτυα μπορούν να χρησιμοποιηθούν για την επεξεργασία ιατρικών εικόνων (Miller, et al., 1992).

Ο Hall συνέκρινε την χρήση των νευρωνικών δικτύων με αυτή ασαφούς λογικής (fuzzy logic) σε εικόνες μαγνητικών υπερήχων του ανθρώπινου εγκεφάλου (Magnetic Resonance Imaging [MRI]). Και οι δύο προσεγγίσεις εφαρμόστηκαν για την «νοήμων διάγνωση», την διάγνωση σύγχρονες επιστημονικές τεχνικές χωρίς την συμμετοχή του ανθρώπου. Τα αποτελέσματα όπως πιστοποιήσαν πεπειραμένοι ακτινολόγοι παρείχαν καλή «διορατικότητα» για την αυτόματη κατηγοριοποίηση των εικόνων (υγιείς/μη-υγιείς). Και οι δύο τεχνικές παρουσίασαν αυξημένη καταλληλότητα για τον συγκεκριμένο επιστημονικό τομέα αλλά έδωσαν και το έναυσμα για την ευρύτερη διάδοση τους στον τομέα ευφυούς ιατρική διάγνωσης (Hall, et al., 1992).

Ο Ossien και συνεργάτες το 1994 δημιούργησαν επιτυχώς ένα τεχνητό νευρωνικό δίκτυο για την βελτίωση του χαμηλού επιπέδου κατακερματισμού εικόνων για τον έλεγχο του οφθαλμού στην πάθηση Grave. Το όλο σύστημα βασίστηκε σε ένα ήδη υπάρχον ιατρικό σύστημα ανάλυσης εικόνας. Σε τελική μορφή το σύστημα είχε την δυνατότητα αλληλεπίδρασης με τον χρήστη για την επιλογή συγκεκριμένων

εικόνων προς ανάλυση, διαφορετικών τεχνικών νευρωνικών δικτύων, διαφορετικών αλγορίθμων εκπαίδευσης και ειδικών δεδομένων (Ossen, et al., 1994).

Ο Anthony και συνεργάτες το 1994 αξιολόγησαν τις επιδόσεις των νευρωνικών δικτύων στην συμπίεση εικόνας σπινθηρογραφήματος πνεύμονα. Ανέλυσαν την καταλληλότητα των νευρωνικών δικτύων και παρουσίασαν τους περιορισμούς και τις επιστημονικές συστάσεις τους με ειδική αναφορά για την ανάλυση της ιατρικής εικόνας (Anthony, et al., 1990).

Ο DeDomicis και συνεργάτες το 1994 παρουσίασαν ένα πολυσυναρτησιακό σύστημα για τον εστιασμό, τον κερματισμό και την κατηγοριοποίηση στις ραδιογραφίες στήθους πνευμονικών παθήσεων. Εφαρμόστηκε ένα Λαπλασιανό - Γκαουσιανού πυρήνα - φίλτρο (Laplacian-of-Gaussian kernel filter) στις εικόνες ακτίνων X για την αφαίρεση τμημάτων χαμηλών συχνοτήτων παράλληλα με την διατήρηση της υψηλής ευκρίνειας. Για τον σχεδιασμό του τεχνητού νευρωνικού δικτύου χρησιμοποιήθηκε η μέθοδος της εμπροσθοδιάδοσης. Η έξοδος του δικτύου εντόπιζε τις περιοχές ενδιαφέροντος στην εικόνα (Regions Of Interest [ROIs]) που αργότερα αναλύονταν από άλλα εργαλεία του συστήματος (Sordo, 2002).

[Houston, A., Kemp, P., Macleod, M., Norsworthy, J., et al. (1994). Analysis of a Nuclear Medicine Database for Colorectal Cancer Using Expert System Rule Induction and Neural Networks. In Ifeachor, E., & Rosen, K. (Eds.), International Conference on Neural Networks and Expert Systems in Medicine and Healthcare, pp. 303--309.] Ο Houston και συνεργάτες το 1994 συνέκριναν ένα εμπειροσύστημα και ένα νευρωνικό δίκτυο για να καθορίσουν την μέθοδο οπτικής διάγνωσης για τον εντοπισμό καρκίνου και άλλων όγκων σε MRI εικόνες. Χρησιμοποιήθηκαν δεδομένα από 39 ασθενείς για την εξακρίβωση της καταλληλότητας των συγκεκριμένων μεθόδων. Τελικώς η έρευνα αποφάνθηκε πως και για τις δύο μεθόδους θα πρέπει να χρησιμοποιηθεί ικανός αριθμός δειγμάτων.

Ο Karakas και συνεργάτες το 1994 χρησιμοποίησαν τεχνητά νευρωνικά δίκτυα για την αυτόματη κατηγοριοποίηση ιστών του αίματος από εικόνες μικροσκοπίων. Συνολικά 82 αντικείμενα εξάχθηκαν από 133 ψηφιοποιημένες εικόνες που είχαν κατακερματιστεί με κλασσικούς αλγορίθμους βελτίωσης εικόνας. Ως αποτέλεσμα εξαγόταν μια δυαδική έξοδος αληθής ή ψευδής ανάλογα αν ο ιστός είχε προσβληθεί από κάποια ασθένεια ή όχι. Τελικά 65 από τα 82 αντικείμενα κατηγοριοποιήθηκαν ορθά (Karakas, et al., 1994).

Οι Xing και Zheng το 1994 μαζί με τις επιστημονικές τους ομάδες αποτέλεσαν δυο από τα πολλά παραδείγματα ενασχόλησης των νευρωνικών δικτύων με την αναγνώριση προτύπων στις μαστογραφίες. Ο Xing και οι συνεργάτες χρησιμοποίησαν 14 συγκεκριμένα χαρακτηριστικά για την ανάλυση των εικόνων ύστερα από διαβουλεύσεις με έμπειρους ακτινολόγους. Ένα πυραμιδοειδές νευρωνικό δίκτυο αναγνώριζε κακοήθεις όγκους σημεία ασβεστοποίησης σε προ-επεξεργασμένες μαστογραφίες. Τα αποτελέσματα απέδειξαν ότι τα πρότυπα που παρουσίαζαν ανωμαλίες μπορούσαν να διαχωριστούν. Παρομοίως ο Zheng και οι συνεργάτες του χρησιμοποίησαν ένα πολύ-επίπεδο νευρωνικό δίκτυο για τον εντοπισμό σημείων ασβεστοποίησης και την κατηγοριοποίηση (υγιών και μη-υγιών) σε ψηφιακές

μαστογραφίες. Το νευρωνικό δίκτυο εκπαιδεύτηκε χρησιμοποιώντας την μέθοδο την οπισθοδιάδοσης και φίλτρο τύπου Kalman. Τα πειραματικά αποτελέσματα έδειξαν 100% προβλεπτικότητα για ποσοστό λάθους μικρότερο από 1 σημείο μικρο-ασβεστοποίησης ανά εικόνα (Xing & Feltham, 1994), (Zheng, et al., 1994).

Ο Aizenberg και συνεργάτες το 2001 παρουσίασαν δείγματα τεχνικών για το φιλτράρισμα, τον καταμερισμό και την αναγνώριση αιχμών χρησιμοποιώντας κυτταρικά νευρωνικά δίκτυα για την βελτίωση της ανάλυσης στις εγκεφαλοτομογραφίες και την διόρθωση της συχνότητας για την εύρεση ασβεστοποίησης στα μαστογραφήματα (Aizenberg, et al., 2001).

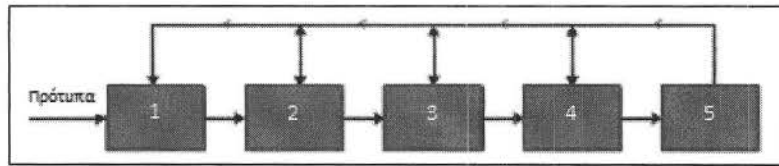
Ο Weinstein και συνεργάτες στο Εθνικό Ινστιτούτο Καρκίνου των ΗΠΑ σχεδίασαν ένα νευρωνικό δίκτυο για την ανάπτυξη φαρμακευτικών ουσιών. Το νευρωνικό δίκτυο μπορούσε να προβλέψει τον μηχανισμό δράσης ενός φαρμάκου από το πρότυπο της δραστηριότητας του. Το δίκτυο κατηγοριοποίησε επιτυχώς σε ποσοστό 91,5% τους παρουσιαζόμενους αντικαρκινικούς «πράκτορες» (anticancer agents) σύμφωνα με τον μηχανισμό δράσης τους. Εν συγκρίσει με το 85,8% του βαθμού κατηγοριοποίησης της γραμμικής ανάλυσης και άλλων στατιστικών μεθόδων, τα νευρωνικά δίκτυα ξεκάθαρα απέδειξαν την ικανότητα τους για την κατηγοριοποίηση σύνθετων μορφών δεδομένων σε έναν ακόμα τομέα της επιστήμης (Weinstein, et al., 1992).

1.7 Αναγνώριση και κατηγοριοποίηση προτύπων

Η αναγνώριση προτύπων έχει ως σκοπό την κατηγοριοποίηση «αντικειμένων» σε ένα πλήθος κατηγοριών ή κλάσεων. Ανάλογα με την χρήση, αντικείμενα προς μελέτη μπορεί να είναι εικόνες ή χρονοσειρές ή άλλου είδους μετρήσεις οι οποίες χρειάζονται κατηγοριοποίηση. Με την ανάπτυξη των υπολογιστικών συστημάτων από το 1960 μέχρι σήμερα αναπτύχθηκε και η αναγνώριση προτύπων με πληθώρα εφαρμογών στον βιομηχανικό εξοπλισμό. Πιο πολύ θα μπορούσαμε να πούμε ότι η αναγνώριση προτύπων ανήκει στην κατηγορία της μηχανικής νοημοσύνης και των συστημάτων υποστήριξης αποφάσεων.

Στην παρακάτω εικόνα μπορούμε να δούμε τα διάφορα στάδια για την σχεδίαση ενός συστήματος κατηγοριοποίησης. Όπως γίνεται αντιληπτό, λόγω των αναδράσεων, τα διάφορα στάδια δεν είναι ανεξάρτητα μεταξύ τους. Αντιθέτως η επαναφορά στα πρώτα στάδια έχει ως σκοπό την βελτίωση της αποδοτικότητας του συστήματος καθώς και της επανασχεδίασης των σταδίων.

Οι κατηγοριοποιητές χωρίζονται σε διάφορες κατηγορίες ανάλογα το είδος των δεδομένων. Έτσι μπορεί τα δεδομένα να είναι γραμμικά ή μη – γραμμικά και να εφαρμοστεί το ανάλογο είδος κατηγοριοποιητή



Εικόνα 20. Διάγραμμα Βασικών σταδίων σχεδιασμού ενός συστήματος κατηγοριοποίησης. Όπου 1) Αισθητήρας 2) Παραγωγή χαρακτηριστικών 3) Διαλογή χαρακτηριστικών 4) Σχεδιασμός κατηγοριοποιητή 5) Αξιολόγηση συστήματος

. Τα γραμμικά μοντέλα χαρακτηρίζονται από απλότητα και υπολογιστική ελκυστικότητα. Ενώ χωρίζονται σε διάφορες υποκατηγορίες ανάλογα με την εξάρτηση (ή ανεξαρτησία) από την κατανομή περιγραφής των δεδομένων εκπαίδευσης. Το πρότυπο του αισθητήρα perceptron, η μέθοδος των ελαχίστων τετραγώνων, οι μηχανές υποστήριξης διανυσμάτων αποτελούν τεχνικές κατηγοριοποίησης ανεξάρτητες από τις κατανομές περιγραφής των δεδομένων εκπαίδευσης. Αρχικά για μη γραμμικά προβλήματα διακριτής κλάσης γίνεται προσπάθεια οι γραμμικοί κατηγοριοποιητές να σχεδιαστούν με βέλτιστο τρόπο, για παράδειγμα ελαχιστοποιώντας το λάθος στην μέθοδο των ελαχίστων τετραγώνων. Στην φύση αλλά και γενικότερα υπάρχουν προβλήματα τα οποία ακόμα και η βελτιστοποίηση των γραμμικών κατηγοριοποιητών δεν οδηγεί σε ικανοποιητικά αποτελέσματα. Έτσι θα μπορούσαμε να πούμε πως η επιλογή των μη – γραμμικών κατηγοριοποιητών αποτελεί αναπόδραστη αναγκαιότητα (Theodoridis & Koutroumbas, 2009). Μέθοδοι μη-γραμμικών κατηγοριοποιητών είναι τα νευρωνικά δίκτυα πολλών επιπέδων, ο αλγόριθμος της οπισθοδιάδοσης, πολυωνυμικοί κατηγοριοποιητές, συναρτήσεων ακτινικής βάσης (Radial Basis Function Networks), πιθανοτικά νευρωνικά δίκτυα, δέντρων απόφασης, κλπ.

1.7.1 Επιλογή κατηγοριοποιητή

Οι σεισμοί αποτελούν συστήματα μη-γραμμικά με μεγάλη πολυπλοκότητα και για αυτό τον σκοπό θα πρέπει να επιλέξουμε μη – γραμμικούς κατηγοριοποιητές. Η δυνατότητα που έχουν τα τεχνητά νευρωνικά δίκτυα να μαθαίνουν καθώς η δυνατότητα βελτιστοποίησης των αποτελεσμάτων μέσα από την χρήση υβριδικών μοντέλων και συνδυαστικών τεχνικών μας οδηγεί στην επιλογή τους. Τα πολυ-επίπεδα νευρωνικά δίκτυα, τα πιθανοτικά νευρωνικά δίκτυα, και τα δίκτυα συναρτήσεων ακτινικής βάσης θα είναι αυτά τα οποία θα μελετήσουμε για την τελική επιλογή για την κατασκευή του μοντέλου της παρούσας εργασίας. Καθώς τα νευρωνικά δίκτυα πολλαπλών επιπέδων έχουν προαναφερθεί θα γίνει μια προσπάθεια εμβάθυνσης στα νευρωνικά δίκτυα συνάρτησης ακτινικής βάσης και στα πιθανοτικά νευρωνικά δίκτυα.

1.7.2 Νευρωνικά δίκτυα συναρτήσεων ακτινικής βάσης

Τα νευρωνικά δίκτυα συναρτήσεων ακτινικής βάσης (Radial Basis Function – RBF neural networks) έχουν ένα κρυφό επίπεδο και ένα επίπεδο εξόδου. Πήραν την ονομασία τους από την ιδιότητα τους, να αντικαθίσταται ο γραμμικός συντελεστής των νευρώνων του κρυφού επιπέδου.

Συνάρτηση ακτινικού τύπου $f(x)$ είναι η συνάρτηση η οποία η τιμή της εξαρτάται από την απόσταση X από ένα διάνυσμα c το οποίο ονομάζεται κέντρο. Ισχύει δηλαδή:

$$f(X) = f(\|X - c\|), \quad \text{Εξίσωση 7}$$

Για κύκλο με κέντρο c και ακτίνα

$$r = \|X - c\|, \quad \text{Εξίσωση 8}$$

η τιμή της f είναι σταθερή. Ενά κλασσικό παράδειγμα ακτινικής συνάρτησης είναι η συνάρτηση Gauss. Η συνάρτηση που δείχνει την σχέση εισόδου εξόδου και συντελεστών βαρύτητας των συνάψεων, είναι η ακόλουθη:

$$O = \sum_{i=1}^N w_i G(x, q_i), \quad \text{Εξίσωση 9}$$

Όπου: x είναι η είσοδος του δικτύου και w_i με $i = 1, 2, \dots, N$ οι συντελεστές βαρύτητας του νευρώνα εξόδου, το G αποτελεί έναν πίνακα σταθερών τιμών.

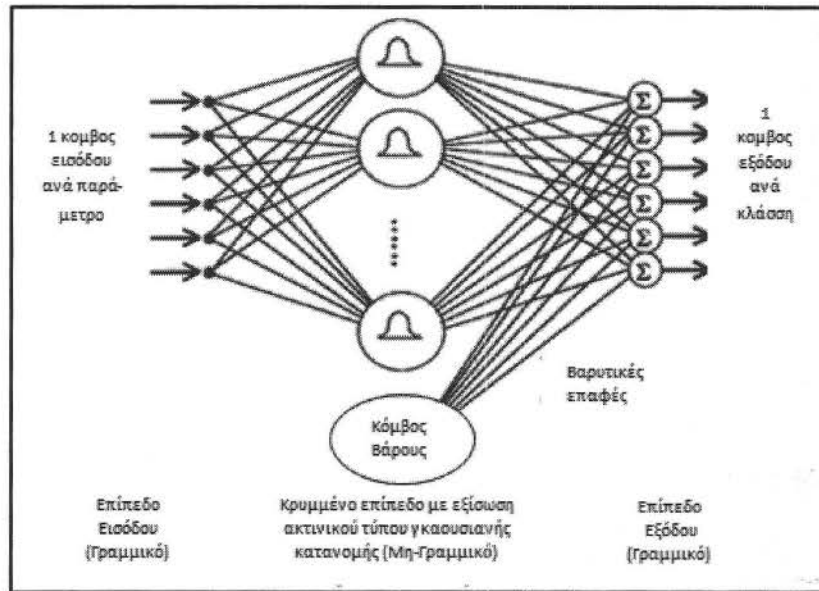
$$G = \begin{pmatrix} G(x_1, q_1) & G(x_1, q_2) & K & G(x_1, q_N) \\ G(x_2, q_1) & G(x_2, q_2) & K & G(x_2, q_N) \\ K & K & K & K \\ G(x_M, q_1) & G(x_M, q_2) & K & G(x_M, q_N) \end{pmatrix}, \quad \text{Εξίσωση 10}$$

Το μέσο σφάλμα για μια επιθυμούμενη έξοδο b , μιας προσομοίωσης M παραδειγμάτων είναι:

$$\text{Error} = (b - Gw)^T (b - Gw) = \|b - Gw\|^2 = \sum_{j=1}^M (b_j - \sum_{i=1}^N w_i G(x, q_i))^2, \quad \text{Εξίσωση 11}$$

Με $b = (b_1, b_2, \dots, b_M)^T$ να είναι το διάνυσμα των επιθυμούμενων εξόδων και $w = (w_1, w_2, \dots, w_N)^T$ οι συντελεστές βαρύτητας του νευρώνα.

Η μορφολογία ενός νευρωνικού δικτύου συναρτήσεως ακτινικής βάσης είναι η ακόλουθη, εδώ θεωρούμε ότι η εξίσωση βάσης είναι η γκαουσιανή.



Εικόνα 21. Αρχιτεκτονική νευρωνικού δικτύου συναρτήσεως ακτινικής βάσης

Βασικό χαρακτηριστικό της αρχιτεκτονικής του παραπάνω νευρωνικού δικτύου είναι η έλλειψη πολλών επιπέδων, έτσι έχουμε 3 επίπεδα 1 επίπεδο εισόδου, 1 κρυμμένο επίπεδο της ακτινικής εξίσωσης και 1 επίπεδο εξόδου. Αυτό συμβαίνει λόγω της έλλειψης ικανοποιητικού αλγορίθμου εκπαίδευσης για 4 και περισσότερα στρώματα.

1.7.3 Εκπαίδευση νευρωνικού δικτύου συναρτήσεως ακτινικής βάσης

Για να εκπαιδεύσουμε το νευρωνικό δίκτυο ακτινικής βάσης θα πρέπει τα κέντρα c_i και τα εύρη σ_i των νευρώνων του κρυφού επιπέδου ($i = 1, 2, \dots, K$) να εκπαιδευτούν, καθώς επίσης και οι τιμές των συναπτικών βαρών w_{ij} του επιπέδου εξόδου αντίστοιχα. Θα πρέπει να σημειωθεί ότι το επίπεδο εξόδου εκπαιδεύεται με διαφορετικό σε σχέση με το κρυμμένο επίπεδο έτσι λέμε ότι τα δίκτυα ακτινικής βάσης παρουσιάζουν ανομοιογένεια. Παρακάτω θα γίνει αναφορά στα πιο σημαντικά πρότυπα εκπαίδευσης.

1.7.4 Μέθοδος εκπαίδευσης με ένα πρότυπο ανά κέντρο

Σύμφωνα με αυτή την μέθοδο κάθε διάνυσμα εισόδου $X^i, i = 1, 2, \dots, P$ αποτελεί και ένα κέντρο, έτσι έχουμε $C_i = X^i, K = P, i = 1, 2, \dots, K$. Το αρνητικό που έχει αυτή η μέθοδος είναι ότι για πολλά πρότυπα εισόδου θα δημιουργηθούν πάρα πολλοί νευρώνες στο κρυφό επίπεδο, όμως η μη χρησιμοποίηση επαναληπτικού αλγόριθμου δίνει ταχύτητα στην μάθηση του νευρωνικού δικτύου. Κατά την εκπαίδευση του νευρωνικού δικτύου θα πρέπει να προσέχουμε όταν χρησιμοποιούμε ως ακτινική συνάρτηση την γκαουσιανή κατανομή, οι καμπάνες θα πρέπει να μην είναι ούτε πολύ μυτερές ούτε πολύ επίπεδες. Για να αποφευχθεί το συγκεκριμένο φαινόμενο χρησιμοποιούμε την ακόλουθη ισότητα

$$\sigma_1 = \dots = \sigma_K = \sigma = \frac{D}{\sqrt{2P}} \quad \text{Εξίσωση 12}$$

Το σ_i αποτελεί το εύρος του κάθε i -οστού κρυφού νευρώνα $f(\|X - c_i\|, \sigma_i)$ και D η απόσταση των πιο απομακρυσμένων κέντρων, δίνουμε δηλαδή σε όλους του νευρώνες το ίδιο εύρος.

1.7.5 Μέθοδος τυχαίας επιλογής προτύπων που θα αποτελέσουν κέντρα

Μια άλλη μέθοδος η οποία θα είχε χαμηλότερο κόστος ως προς την δημιουργία των κέντρων, είναι να κάνουμε με τυχαίο τρόπο την επιλογή των προτύπων εισόδου που θα γίνουν κέντρα με βάση κάποιο ποσοστό του συνόλου, δηλαδή έστω $K=25\%P$. Ως πλεονέκτημα μπορούμε να θεωρήσουμε την μείωση του πλήθους κέντρων. Το αρνητικό στοιχείο είναι ότι για περιοχές με πυκνά πρότυπα θα υπάρξουν αρκετά κέντρα από αυτή την περιοχή. Έτσι περιγράφεται μια περιοχή με παραπάνω από ένα κέντρα. Η τεχνική αυτή όπως και η προηγούμενη θεωρούνται χαμηλής πολυπλοκότητας και το μεγάλο πλήθος κέντρων επιτυγχάνει ικανοποιητική περιγραφή του προβλήματος.

1.7.6 Μέθοδος ομαδοποίησης των K -μέσων

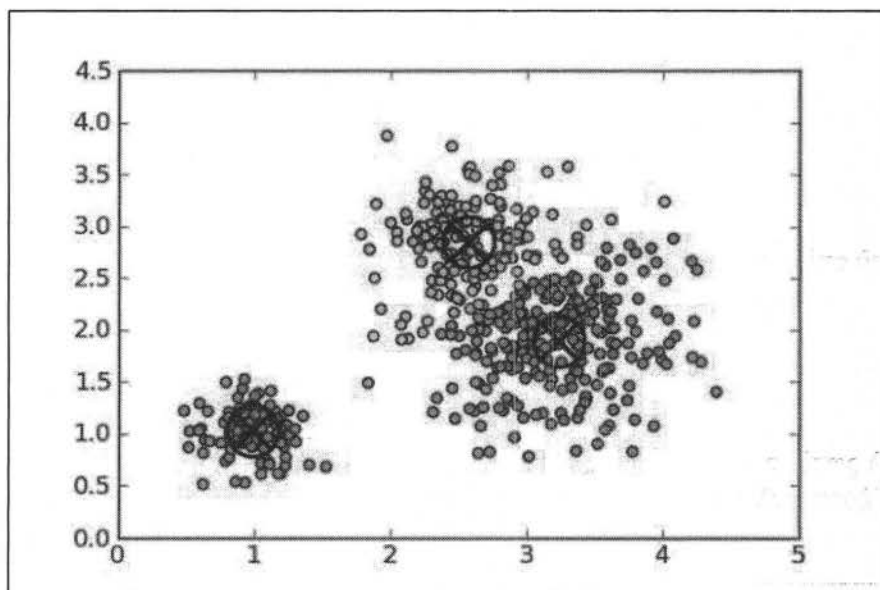
Αποτελεί την πιο γνωστή μέθοδο και λειτουργεί ως εξής: Για ένα πλήθος κλάσεων K, X_1, X_2, \dots, X_K και ένα πλήθος προτύπων P, x_1, x_2, \dots, x_P που ανήκουν σε αυτές, αναζητείται ένα αντιπροσωπευτικό διάνυσμα για την κάθε κλάση. Για να βρεθούν τελικώς τα κέντρα των κλάσεων θα πρέπει τηρούνται οι δύο ακόλουθες συνθήκες:

- I. Για κάθε $i = 1, 2, \dots, K$, η κλάση X_i αποτελείται από τα πρότυπα x_p τα οποία βρίσκονται πιο κοντά στο διάνυσμα c_i σε σχέση με όλα τα άλλα διανύσματα c_j όπου $j \neq i$, ισχύει δηλαδή $\|x_p - c_i\| = \min_j \|x_p - c_j\|$

II. Για κάθε $i = 1, 2, \dots, K$, το διάνυσμα c_i αποτελεί τον μέσο όρο των προτύπων που ανήκουν στην κλάση X_i και ισχύει

$$c_i = \frac{1}{|X_i|} \sum_{p \in X_i} x_p, \quad \text{Εξίσωση 13}$$

Όπου $|X_i|$ είναι το πλήθος προτύπων της κλάσης και το c_i όπως έχουμε αναφέρει είναι το κέντρο της κλάσης. Εδώ μπορούμε να δούμε ένα παράδειγμα υλοποίησης:



Εικόνα 22. Διάγραμμα K -μέσων με ανεύρεση των κέντρων των περιοχών. Στην εικόνα μπορούμε να διακρίνουμε τρεις κύριες κλάσεις

1.7.7 Μέθοδος εκπαίδευσης επιπέδου εξόδου

Για να εκπαιδέσουμε το επίπεδο εξόδου όπως έχουμε προαναφέρει θα ακολουθήσουμε τελείως διαφορετική διαδικασία έτσι σύμφωνα με αυτή, μετά το τέλος της διαδικασίας εκπαίδευσης των νευρώνων του κρυμμένου επιπέδου και γνωρίζοντας πλέον τις τιμές ενεργοποίησης τους a_1, a_2, \dots, a_K , για κάθε πρότυπο εξόδου $x = [x_1, x_2, \dots, x_n]^T$, έχουμε:

$$y_i = \sum_{j=1}^K w_{ij} a_j + w_{i0}, \quad \text{Εξίσωση 14}$$

1.7.8 Σύγκριση νευρωνικών δικτύων συναρτήσεως ακτινικής βάσης και νευρωνικών δικτύων πολλών επιπέδων

Στα πολυεπίπεδα νευρωνικά δίκτυα οι εισοδοί των συναρτήσεων ενεργοποίησης του πρώτου επιπέδου αποτελούν γραμμικούς συνδυασμούς των παραμέτρων εισόδου, έτσι η έξοδος του κάθε νευρώνα είναι η ίδια για όλα τα x_i . Οπότε η έξοδος είναι ίδια για όλα τα σημεία ενός υπερεπιπέδου. Σε αντίθεση, στα νευρωνικά δίκτυα συναρτήσεως ακτινικής βάσης η έξοδος κάθε κόμβου του δικτύου είναι η ίδια για όλα τα σημεία που έχουν την ίδια ευκλειδική απόσταση από το κέντρο c_i , και μειώνονται εκθετικά για τις γκαουσιανές κατανομές. Γενικά δηλαδή οι αποκρίσεις ενεργοποίησης των κόμβων είναι τοπικής φύσης στα νευρωνικά δίκτυα συναρτήσεως ακτινικής βάσης και γενικής φύσης στα πολυεπίπεδα νευρωνικά δίκτυα (Theodoridis & Koutroumbas, 2009). Τα πολυεπίπεδα νευρωνικά δίκτυα μαθαίνουν με πιο αργό τρόπο σε σχέση με τα δίκτυα ακτινικής βάσης και εμφανίζουν γενικευμένες ιδιότητες για περιοχές που δεν έχουν παρουσιαστεί επαρκώς στα δεδομένα εκπαίδευσης (Lane, et al., 1991). Ο Hartman κ. α. έδειξε πως για να επιτευχθεί επίδοση παρόμοια με τα πολυεπίπεδα νευρωνικά δίκτυα, τα δίκτυα ακτινικής βάσης θα πρέπει να είναι αρκετά υψηλότερου βαθμού. Αυτό οφείλεται στην τοπικότητα των συναρτήσεων ενεργοποίησης στα δίκτυα ακτινικής βάσης, το οποίο κάνει αναγκαία την χρησιμοποίηση μεγάλου αριθμού κέντρων (Hartman, et al., 1990).

1.7.9 Πιθανοτικά νευρωνικά δίκτυα

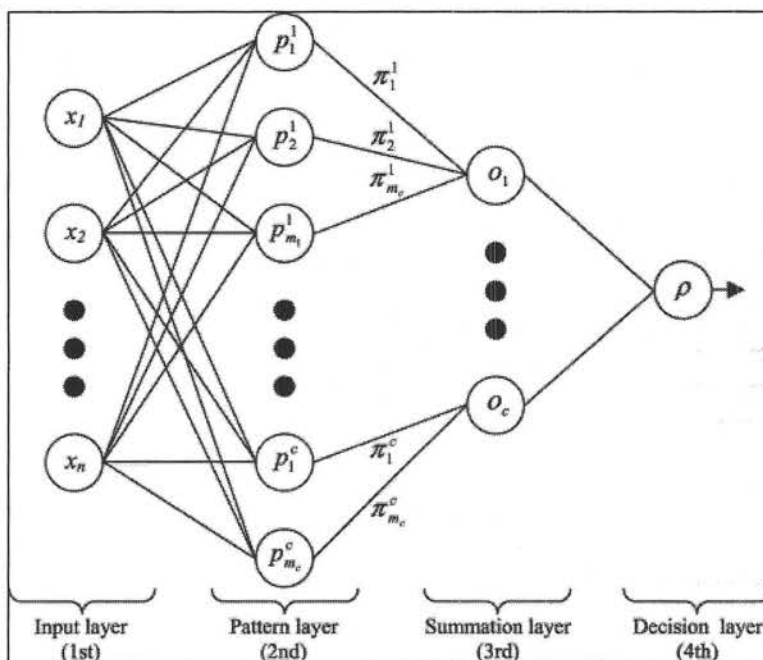
Τα πιθανοτικά νευρωνικά δίκτυα (Probabilistic Neural Networks) εισήχθησαν από τον Specht το 1990 και συνδυάζουν στοιχεία από την στατιστική αναγνώριση προτύπων και των τεχνητών νευρωνικών δικτύων εμπροσθοδιάδοσης (Specht, 1990). Τα πιθανοτικά νευρωνικά δίκτυα χρησιμοποιούν τον κανόνα ταξινόμησης του Bayes για την ταξινόμηση των προτύπων (Raghu & Yegnanarayana, 1998), (Romero, et al., 1997) καθώς επίσης και την συνάρτηση πυκνότητας πιθανότητας του Parzen (Parzen, 1962). Γενικά τα πιθανοτικά νευρωνικά δίκτυα απαιτούν μικρό χρόνο εκπαίδευσης καθώς δεν απαιτούν κάποια εκτίμηση για τους συντελεστές βαρύτητας του δικτύου παρά μόνο απλή προσπέλαση των κανόνων εκπαίδευσης, όμως απαιτούν αρκετά ποσά μνήμης και έχουν αργό χρόνο εκτέλεσης μιας ταξινόμησης για την καταχώρηση νέων στοιχείων στις προκαθορισμένες κλάσσες [Specht, 1990].

1.7.10 Δομή πιθανοτικού νευρωνικού δικτύου

Το πιθανοτικό νευρωνικό δίκτυο έχει μέχρι τέσσερα επίπεδα τα οποία είναι α) το επίπεδο εισόδου β) το επίπεδο προτύπων γ) το επίπεδο άθροισης και δ) το επίπεδο εξόδου. Στο παρακάτω σχήμα (Εικόνα 25) μπορούμε να δούμε την απεικόνισή του.

Αν T_{tr} είναι ένα σύνολο εκπαιδεύσεως του πιθανοτικού νευρωνικού δικτύου και N_{tr} ο αριθμός των διανυσμάτων – χαρακτηριστικών που ανήκουν στο T_{tr} , τότε ορίζουμε ως N_k τον αριθμό διανυσμάτων εκπαίδευσης για κλάσεις $K, k = 1, 2, \dots, K$.

$$N_{tr} = \sum_{k=1}^K N_k, \quad \text{Εξίσωση 15}$$



Εικόνα 23. Διάγραμμα Πιθανοτικού Νευρωνικού δικτύου, (Folland, et al., 2004)

Το επίπεδο εισόδου αποτελείται από p νευρώνες, οι οποίοι μεταφέρουν στους νευρώνες του επιπέδου προτύπων το διάνυσμα εισόδου του πιθανοτικού δικτύου. Κάθε συνιστώσα του διανύσματος εισόδου $x = [x_1, x_2, \dots, x_p]^T$ εφαρμόζεται στους νευρώνες εισόδου. Οι N_{tr} νευρώνες του επιπέδου προτύπων χωρίζονται σε K ομάδες. Ο i -οστός νευρώνας προτύπων της k ομάδας ($k = 1, 2, \dots, K$) χρησιμοποιεί την ακόλουθη συνάρτηση ενεργοποίησης:

$$f_{ik}(x) = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - x_{ik})^T \Sigma^{-1}(x - x_{ik})\right), \quad \text{Εξίσωση 16}$$

όπου ο πίνακας Σ ονομάζεται πίνακας παραμέτρων λείανσης και καθορίζει το μέγεθος και το σχήμα του πεδίου δράσης της συνάρτησης. Ο Specht περιόρισε τον Σ σε μια ολική παράμετρο λείανσης σ^2 , έτσι αντικαθιστώντας το $\Sigma = \sigma^2 I_p$, με I_p να

είναι ο μοναδιαίος πίνακας της p τάξης. Έτσι η προηγούμενη εξίσωση λαμβάνει την ακόλουθη μορφή:

$$f_{ik}(x) = \frac{1}{(2\pi\sigma^2)^{\frac{p}{2}}} \exp\left(-\frac{\|x-x_{ik}\|^2}{2\sigma^2}\right), \quad \text{Εξίσωση 17}$$

Στην συνέχεια κάθε νευρώνας του επιπέδου άθροισης, υπολογίζει το άθροισμα των εξόδων των νευρώνων του επιπέδου προτύπων της αντίστοιχης κλάσης, πολλαπλασιαζόμενο με έναν συντελεστή βαρύτητας.

$$G_k(x) = w_k \sum_{i=1}^{M_k} f_{i,k}(x), \quad \text{με } k = 1, 2, \dots, K, \quad \text{Εξίσωση 18}$$

Ο θετικός συντελεστής βαρύτητας w_k ισούται με την πιθανότητα π_k της κλάσης k και ικανοποιεί την ακόλουθη ισότητα

$$\sum_{k=1}^K w_k = 1, \quad \text{Εξίσωση 19}$$

Το επίπεδο άθροισης του πιθανοτικού νευρωνικού δικτύου εκτιμά εκ των υστέρων την πιθανότητα της κάθε κλάσης, για ένα διάνυσμα χαρακτηριστικών x , διαμέσου του αθροίσματος των συναρτήσεων πυρήνων των νευρώνων προτύπων της κάθε κλάσης.

Ένα διάνυσμα εισόδου x ταξινομείται στην κλάση που αντιστοιχεί στον νευρώνα αθροίσματος που είχε την μεγαλύτερη τιμή εξόδου

$$C(x) = \arg \max(G_k)_{1 \leq k \leq K}, \quad \text{Εξίσωση 20}$$

Τα πιθανοτικά νευρωνικά δίκτυα που χρησιμοποιούν μια κοινή παράμετρο λείανσης ονομάζονται ομοσκεδαστικά (homoscedastic), ενώ όταν τα στοιχεία της κύριας διαγωνίου του πίνακα Σ δεν είναι ίσα μεταξύ τους, ονομάζονται ετεροσκεδαστικά (heteroscedastic) (Spect & Romsdahl, 1994).

Το αρνητικό στοιχείο των πιθανοτικών νευρωνικών δικτύων είναι ότι πάσχει από την λεγόμενη από την επιστημονική κοινότητα κατάρα της διάστασης (curse of dimensionality), αυτο σημαίνει ότι όταν η διάσταση ενός συνόλου δεδομένων είναι μεγάλη, με μικρό αριθμό δεδομένων, παρουσιάζεται αποτυχία επίτευξης ικανοποιητικών αποτελεσμάτων.

Θα πρέπει εδώ να σημειωθεί πως απαιτείται αρκετή προσοχή ως προς την επιλογή του πίνακα παραμέτρων λείανσης καθώς είναι το σημαντικότερο στοιχείο επηρεασμού της απόδοσης των πιθανοτικών νευρωνικών δικτύων.

2. Εξαγωγή χαρακτηριστικών με μεθόδους ανάλυσης πολυπλοκότητας

Στόχος της παρούσας εργασίας είναι η χρησιμοποίηση των νευρωνικών δικτύων ως μηχανισμού κατηγοριοποίησης του βαθμού επικινδυνότητας ενός καταστροφικού φαινομένου όπως αυτό του σεισμού. Για να γίνει αυτό θα πρέπει ως είσοδοι στο νευρωνικό δίκτυο να δοθούν κάποια χαρακτηριστικά γνωρίσματα – δείκτες που μας δείχνουν σε ποια σημεία των προς μελέτη χρονοσειρών παρουσιάζονται ανωμαλίες. Παρακάτω θα γίνει μια ανάλυση των επιλεγμένων μεθόδων ανάλυσης της πολυπλοκότητας για την εξαγωγή αυτών των χαρακτηριστικών.

2.1 Μέθοδοι εντροπίας σήματος

Τα τελευταία χρόνια υπήρξε ταχεία ανάπτυξη των διάφορων μεθόδων μέτρησης πολυπλοκότητας όπως ο εκθέτης του Lyapunov και οι εντροπίες. Οι υπολογισμοί όμως ήταν συχνά αντιμέτωποι με το πρόβλημα του μη επαρκούς αριθμού δεδομένων. Επίσης οι περισσότεροι ορισμοί εντροπίας παρουσιάζουν περιορισμούς, καθώς πολλές φορές παρουσιάζουν θόρυβο κατά τον υπολογισμό τους, όταν θόρυβος επιπέδου 2% είναι ικανός να αποτρέψει τον ακριβή υπολογισμό (Yu, et al., 2000).

Με σκοπό την επίλυση τέτοιου είδους προβλημάτων ο Pincus παρουσίασε την προσεγγιστικής εντροπίας (Approximate Entropy) ως μέθοδο μέτρησης της πολυπλοκότητας για θορυβώδη, μεσαίου μεγέθους σύνολα δεδομένων (Pincus, 1991). Η συγκεκριμένη στατιστική μέθοδος βρήκε ευρεία αποδοχή στην ιατρική στατιστική ανάλυση όπως είναι η ανάλυση γενετικών ακολουθιών, η μεταβλητότητα του καρδιακού ρυθμού, τα ηλεκτροεγκεφαλογράφημα, τα ηλεκτροκαρδιογραφήματα και τα ηλεκτρομυογραφήματα (Bruhn, et al., 2000), (Burioka, et al., 2005), (Pincus & Keefe, 1992), (Storella, et al., 1998). Η μέθοδος αυτή παρουσίασε σημαντικά προτερήματα έναντι άλλων μεθόδων όπως η φρακταλική διάσταση (fractal dimension), η εντροπία Kolmogorov – Sinai και η φασματική εντροπία (Grouven, et al., 2005). Το αρνητικό στοιχείο της προσεγγιστικής εντροπίας είναι η εξάρτηση των ορθών αποτελεσμάτων με το μήκος των δεδομένων. Οι Richman και Moorman βασιζόμενοι στην προσεγγιστική εντροπία πρότειναν μια παρόμοια στατιστική μέθοδο την δειγματοληπτική εντροπία (Sample Entropy), η οποία παρουσιάζει μεγαλύτερη συνεκτικότητα και μικρότερη εξάρτηση από το μήκος των δεδομένων (Richman & Moorman, 2000). Και οι δύο εντροπίες παρουσιάζουν ομοιότητα στον ορισμό των διανυσμάτων βασιζόμενες στην συνάρτηση του Heaviside. Λόγω των εγγενών αδυναμιών της συνάρτησης του Heaviside, υπάρχουν ακόμη προβλήματα στην εγκυρότητα της εντροπίας, ειδικά όταν οι παράμετροι είναι μικρού μεγέθους. Ο Chen κ. α. το 2007 εισηγήθηκαν την μέθοδο της ασαφούς εντροπίας (fuzzy entropy) και όπως απέδειξαν αποτελεί πιο ακριβή μέθοδο μέτρησης της πολυπλοκότητας δείχνοντας μεγαλύτερη συνοχή, μικρότερη εξάρτηση από το μήκος των δεδομένων,

μεγαλύτερη ελευθερία στην επιλογή των παραμέτρων, καθώς και μεγαλύτερη ανθεκτικότητα στον θόρυβο (Chen, et al., 2007).

2.1.1 Μέθοδος της προσεγγιστικής εντροπίας (approximate entropy)

Για ένα δείγμα N χρονικής σειράς $\{u(i): 1 \leq i \leq N\}$, δοσμένου m μήκους συγκρινόμενου παραθύρου και ακολουθία διανύσματος:

$$X_i^m = \{u(i), u(i+1), \dots, u(i+m-1)\}, \text{ με } i = 1, 2, \dots, N-m+1, \quad \text{Εξίσωση 21}$$

Για κάθε $i \leq N-m+1$, έστω ότι $C_i^m(r)$ είναι οι $(N-m+1)^{-1}$ οι φορές, όπου ο αριθμός των διανυσμάτων X_j^m υπάρχουν στο r του X_i^m . Ορίζουμε:

$$\varphi^m(r) = (N-m+1)^{-1} \sum_{i=1}^{N-m+1} \ln C_i^m(r), \quad \text{Εξίσωση 22}$$

Όπου \ln είναι ο φυσικός λογάριθμος. Τελικώς ο Pincus όρισε την προσεγγιστική εντροπία ως εξής (Pincus, 1991), (Chen, et al., 2009):

$$ApEn(m, r) = \lim_{N \rightarrow \infty} [\varphi^m(r) - \varphi^{m+1}(r)], \quad \text{Εξίσωση 23}$$

2.1.2 Μέθοδος δειγματοληπτικής εντροπίας (Sample Entropy)

Η δειγματοληπτική εντροπία διαφέρει σε σχέση με την προσεγγιστική εντροπία σε δύο κύρια μέρη: α) Η δειγματοληπτική εντροπία δεν μετράει τα ιδιοταιριάσματα (self-matches) και β) δεν χρησιμοποιεί μια προτυπο-κεντρική προσέγγιση.

Ορίζουμε ως $B_i^m(r)$ τις $(N-m-1)^{-1}$ φορές, όπου ο αριθμός των διανυσμάτων X_j^m υπάρχουν στο r του X_i^m , με το j να κινείται μεταξύ 1 και $N-m$ με $j \neq i$ έτσι ώστε να αποκλείονται τα αυτο-ταιριάσματα. Στην συνέχεια ορίζεται:

$$B^m(r) = (N-m)^{-1} \sum_{i=1}^{N-m} B_i^m(r), \quad \text{Εξίσωση 24}$$

Ομοίως ορίζουμε ως $A_i^m(r)$ ως τις $(N-m-1)^{-1}$ τις φορές, όπου ο αριθμός των διανυσμάτων X_j^{m+1} υπάρχουν στο r του X_i^{m+1} , με το j να κινείται μεταξύ 1 και $N-m$ (με $j \neq i$). Έτσι έχουμε:

$$A^m(r) = (N-m)^{-1} \sum_{i=1}^{N-m} A_i^m(r), \quad \text{Εξίσωση 25}$$

Η παράμετρος της δειγματοληπτικής εντροπίας μπορεί να υπολογιστεί στην συνέχεια από την παρακάτω ισότητα:

$$SampEn(m,r,N) = -\ln \left[\frac{A^m(r)}{B^m(r)} \right], \quad \text{Εξίσωση 26}$$

2.1.3 Μέθοδος ασαφούς εντροπίας (Fuzzy Entropy)

Η μέθοδος της ασαφούς εντροπίας, όπως και δειγματοληπτική εντροπία εξαιρεί τα ιδιοσυνιστώματα και ελέγχει μόνο τα πρώτα $N-m$ διανύσματα μήκους m έτσι ώστε να εξασφαλιστεί ότι X_i^m και X_{i+m}^m ορίζονται για όλα τα i τέτοια ώστε $1 \leq i \leq N - m$. Παρακάτω ορίζεται:

$$X_i^m = \{u(i), u(i+1), \dots, u(i+m-1)\} - u_0(i), \text{ με } i = 1, 2, \dots, N - m + 1, \quad \text{Εξίσωση 27}$$

Όπου το X_i^m αντιπροσωπεύει τις m διαδοχικές u τιμές, αρχής γενομένης με το i -οστό σημείο και γενικευμένο με την αφαίρεση του $u_0(i)$ το οποίο απεικονίζεται με την παρακάτω ισότητα:

$$u_0(i) = m^{-1} \sum_{j=0}^{m-1} u(i+j), \quad \text{Εξίσωση 28}$$

Δοθέντος διανύσματος X_i^m , υπολογίζεται ο βαθμός ομοιότητας D_{ij}^m για το γειτνιαζόμενο διάνυσμα X_j^m οριζόμενο από ασαφή συνάρτηση ως εξής:

$$D_{ij}^m = \mu(d_{ij}^m, r), \quad \text{Εξίσωση 29}$$

Όπου d_{ij}^m είναι η μέγιστη απόλυτη διαφορά των αντίστοιχων βαθμωτών συνιστώσεων του X_i^m και του X_j^m . Για κάθε διάνυσμα X_i^m ($i = 1, 2, \dots, N - m + 1$), βρίσκοντας των μέσο όρο του βαθμού ομοιότητας των γειτνιαζόντων διανυσμάτων X_j^m ($j = 1, 2, \dots, N - m + 1, j \neq i$) έχουμε:

$$\varphi_i^m(r) = (N - m - 1)^{-1} \sum_{j=1, j \neq i}^{N-m} D_{ij}^m, \quad \text{Εξίσωση 30}$$

Ορίζουμε:

$$\varphi^m(r) = (N - m)^{-1} \sum_{i=1}^{N-m} \varphi_i^m(r), \quad \text{Εξίσωση 31}$$

και

$$\varphi^{m+1}(r) = (N - m)^{-1} \sum_{i=1}^{N-m} \varphi_i^{m+1}(r), \quad \text{Εξίσωση 32}$$

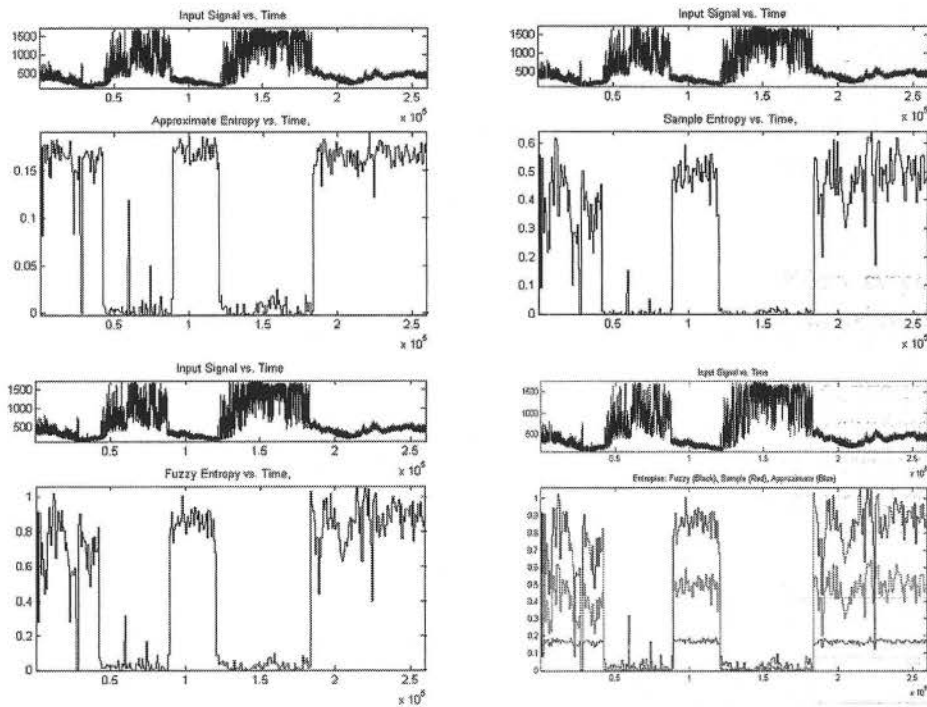
Η παράμετρος της ασαφούς εντροπίας μπορεί να υπολογιστεί στην συνέχεια από την παρακάτω ισότητα για περατό αριθμό συνόλου δεδομένων:

$$\text{FuzzyEn}(m, r) = \ln \varphi^m(r) - \ln \varphi^{m+1}(r), \quad \text{Εξίσωση 33}$$

2.1.4 Σύγκριση μεταξύ της Προσεγγιστικής Εντροπίας, της Ασαφούς Εντροπίας και της δειγματοληπτικής εντροπίας

Με σκοπό την επιλογή της καλύτερης εκ των τριών μεθόδων εντροπίας παρατίθεται παρακάτω ένας πεπερασμένος αριθμός πειραματικών μετρήσεων σε σεισμικές χρονοσειρές.

Όπως βλέπουμε παρακάτω οι 3 εντροπίες φαίνεται να μοιάζουν οι αποκρίσεις τους όταν όμως τεθούν σε παράθεση, όπως φαίνεται στο τέταρτο τεταρτημόριο, θα παρατηρήσουμε την διαφορά ως προς άξονα του Y. Η μεγαλύτερη διαφορά που παρουσιάζουν οι υψηλότερες τιμές της ασαφούς εντροπίας σε σχέση με τις άλλες δυο μας κάνει να την επιλέξουμε ως πιο κατάλληλη μέθοδο για την μελέτη των σεισμικών



Εικόνα 24. Διάγραμμα σύγκρισης εντροπιών σε προ-σεισμικό σήμα Κανάλι 5 σεισμός Αθήνας 1999

σημάτων καθώς η διαφορά των υψηλότερων τιμών εντροπίας με τις μικρότερες για τις εντροπίες της δειγματοληπτικής εντροπίας και της Προσεγγιστικής εντροπίας είναι πολύ μικρή και υπάρχει κίνδυνος να θεωρηθεί ως θόρυβος για την μετέπειτα επεξεργασία από το νευρωνικό δίκτυο.

Ο Chen κ. α. το 2009 υποστήριξε πως η χρησιμοποίηση των ασαφών συναρτήσεων συμμετοχής (fuzzy membership functions) αντί της συνάρτησης του Heaviside από την προσεγγιστική εντροπία και την δειγματοληπτική εντροπία, διασφαλίζουν στην ασαφή εντροπία τον καλό ορισμό της στις μικρές παραμέτρους και την συνεχή αλλαγή της τιμής της. Επίσης η ασαφής εντροπία παρουσιάζει πιο ακριβή ορισμό εντροπίας σε σχέση με τις άλλες δύο εντροπίες και αποφασίζει αν δύο διανύσματα είναι παρόμοια με βάση το σχήμα τους και όχι τις απόλυτες συντεταγμένες τους. Ένα άλλο θετικό στοιχείο της ασαφούς εντροπίας είναι το μεγαλύτερο άθροισμα του βαθμού ομοιότητας που σε συνδυασμό με τον αποκλεισμό των αυτο-ταιριασμάτων εξασφαλίζει την ισχυρότερη σχετική συνεκτικότητα (relative consistency) και την μικρή εξάρτηση με το μήκος των δεδομένων. Τέλος καταλήγει πως τόσο η θεωρητική ανάλυση όσο και τα πειραματικά αποτελέσματα δείχνουν ότι η ασαφής εντροπία παρέχει μια βελτιωμένη αξιολόγηση της πολυπλοκότητας του συστήματος και μπορεί να χρησιμοποιηθεί και σε μικρής χρονικής περιόδου χρονοσειρές υψηλού θορύβου (Chen, et al., 2009).

2.1.5 Ανάλυση Φράκταλ

Όπως έχει βρεθεί κατά την διαδικασία εξέλιξης ενός φαινομένου αυξημένης πολυπλοκότητας όπως είναι ένας σεισμός, οι συσχετίσεις μεταξύ χώρου και χρόνου παράγουν χαρακτηριστικές φρακταλικές δομές (Eftaxias, 2010), (Smirnova, et al., 2004), (Smirnova & Hayakawa, 2007). Στην διεθνή βιβλιογραφία είναι κοινώς αποδεκτό ότι αυτές οι φρακταλικές δομές επηρεάζουν σήματα προερχόμενα από την ευρύτερη διαδικασία παραγωγής ενός σεισμικού φαινομένου. Η μέθοδος της πυκνότητας φάσματος ισχύος (Power Spectral Density), $S(f)$ αποτελεί την πιο κοινά αποδεκτή χρησιμοποιημένη τεχνική για την ανάδειξη στοιχείων σχετικών με την εγγενή μνήμη του συστήματος. Παρ' όλο που η ισχύς φάσματος είναι το χαμηλότερης σειράς στατιστικό μέτρο των αποκλίσεων της τυχαίας πυκνότητας από την ομοιογένεια, απεικονίζει ευθέως τις φυσικές διακυμάνσεις των διαδικασιών σχηματισμού της γεωλογικής δομής (Eftaxias, et al., 2009).

Εάν μια καταγραφόμενη χρονοσειρά $A(t_i)$, αποτελεί προσωρινό φράκταλ (temporal fractal), τότε αναμένεται να ισχύει ο τύπος της πυκνότητας φάσματος ισχύος

$$S(f) = \alpha \cdot f^{-b}, \quad \text{Εξίσωση 34}$$

όπου f είναι η συχνότητα των μετασχηματισμών. Σε μια λογαριθμική αναπαράσταση $S(f) - \log(f)$, η ισχύς φάσματος είναι μια ευθεία γραμμή, με γραμμική φασματική κλίση (linear spectral slope) b . Η φασματική ενίσχυση

ποσοτικοποιεί την ισχύ των φασματικών συνιστωσών που ακολουθούν τον νόμο της πυκνότητας φάσματος ισχύος. Ο φασματικός εκθέτης αποτελεί ένα μέτρο για την ισχύ των χροικών συσχετίσεων. (Eftaxias, et al., 2009).

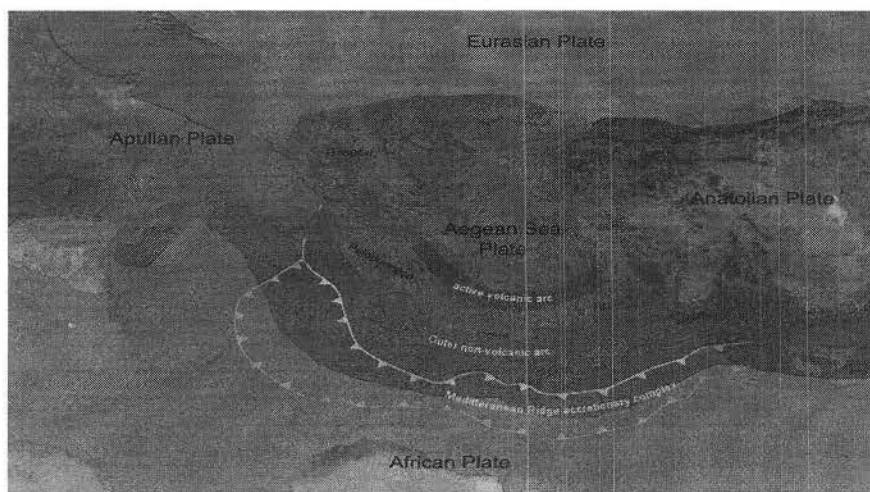
3. Σεισμικά φαινόμενα στην Ελλάδα

3.1 Εισαγωγή στους σεισμούς

Η σύγκρουση των λιθοσφαιρικών πλακών λόγω της αντίθετης κίνησης τους προκαλεί αργή παραμόρφωση στα πετρώματα τους. Τα πετρώματα τα οποία η θέση τους βρίσκεται στα όρια των πλακών, αποκτούν μεγάλη δυναμική ενέργεια οι οποίες αυξάνουν συνεχώς ως προς το χρόνο μέχρις ότου ξεπεράσουν την αντοχή του πετρώματος. Σε εκείνη την φάση τα πετρώματα διαρυνγνείονται σε άλλα μικρότερα και κατακρημνίζονται και ολισθαίνουν μέχρις ότου να αποκτήσουν μια σχετική ισορροπία. Η ανωμαλία που παρουσιάζεται κατά την σύγκρουση δύο η περισσότερων λιθοσφαιρικών πλακών έγκειται στα μεγάλα μεγέθη δυνάμεων τριβής και αντίστασης, οι οποίες προκαλούν μια ταλάντωση. Σε αυτή την φάση η δυναμική ενέργεια μετατρέπεται σε κινητική και μεταδίδεται σε γειτονικά σημεία εξασθενώντας καθώς απομακρυνόμαστε από το ρήγμα. Η μετάδοση αυτή των ταλαντώσεων αποκαλείται σεισμικό κύμα (Papazachos & Papazachou, 1989).

3.2 Η γεωμορφολογία του Ελληνικού χώρου.

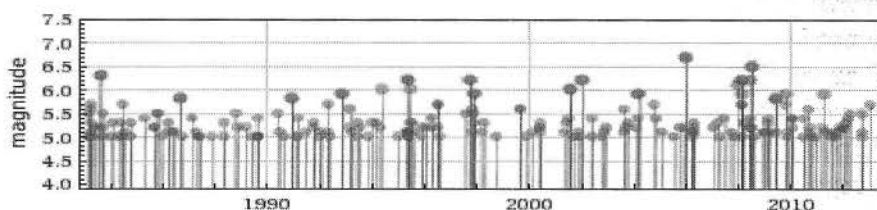
Ο Ελληνικός χώρος χωριοθετείται στα όρια επαφής της Αφρικανικής λιθοσφαιρικής πλάκας και της Ευρω-Ασιατικής λιθοσφαιρικής πλάκας. Η λιθοσφαιρική πλάκα της Απούλιας (Ανδριατική Λιθοσφαιρική πλάκα) αποτελεί το



Εικόνα 25. Γεωτεκτονικός χάρτης της Ελλάδας (Fotopoulos, et al., 2012)

βορειοδυτικό τμήμα της βαλκανικής χερσονήσου. Η λιθόσφαιρα της ανατολικής μεσογείου (πρόσθιο μέρος αφρικανικής λιθοσφαιρικής πλάκας) και η λιθόσφαιρα του Αιγαίου (πρόσθιο μέρος Ευρασιατικής λιθοσφαιρικής πλάκας) συγκλίνουν με βόρεια – βορειοδυτική κατεύθυνση. Πιο συγκεκριμένα η λιθόσφαιρα της ανατολικής μεσογείου βυθίζεται υπό γωνία 36ο περίπου ως προς το οριζόντιο επίπεδο κάτω από την λιθόσφαιρα του Αιγαίου (Parazachos & Parazachou, 1989). Υπάρχουν τρία χαρακτηριστικά γνωρίσματα της γεωμορφολογικής εικόνας της χώρας: α) Το ελληνικό τόξο β) Η λεκάνη του Βορείου Αιγαίου γ) Το Ελληνικό τόξο. Στην παρακάτω εικόνα διαφαίνεται και η λιθοσφαιρική πλάκα της Ανατολίας (ή αλλιώς Τουρκική πλάκα). Η συγκεκριμένη πλάκα έχει δυτική κίνηση, αντίθετη με την ανατολική κίνηση της λιθόσφαιρας της ανατολίας. Συνεπώς η λιθοσφαιρική πλάκα του αιγαίου δέχεται πειστικές δυνάμεις από τα σύνορα της.

Η θέση αυτή, είχε ως αποτέλεσμα για την Ελλάδα την παραγωγή αρκετών σεισμικών φαινομένων από την αρχαιότητα ως σήμερα. Στην παρακάτω εικόνα μπορούμε να δούμε για τα τελευταία 30 χρόνια τα κυριότερα σεισμικά φαινόμενα ($M_L > 5$).



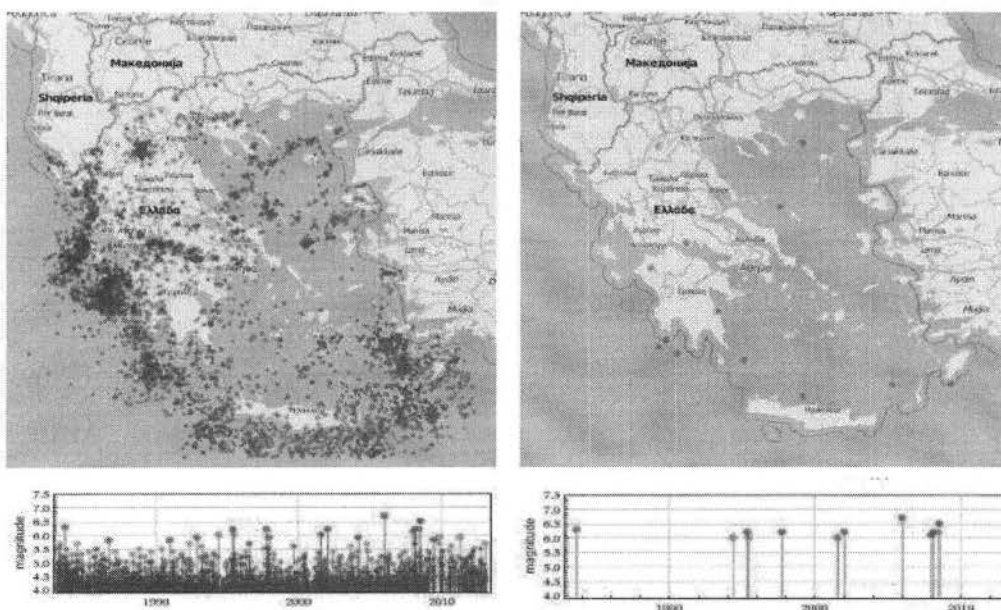
Εικόνα 26. Διάγραμμα μεγεθών σεισμών για $M_L > 5$ στην Ελλάδα τα τελευταία 30 χρόνια. Πηγή: Wolfram Alpha

Η αυξημένη πολυπλοκότητα που παρουσιάζουν τα σεισμικά φαινόμενα στην Ελλάδα προσέλκυσε το ερευνητικό ενδιαφέρον αρκετών ερευνητών. Καθώς και την χρησιμοποίηση αυξημένης δυσκολίας στατιστικών μοντέλων για την εξόρυξη πληροφοριών και την επιβεβαίωση η μη, ως πρόδρομου μαθηματικού μοντέλου συσχετιζόμενο με το επερχόμενο καταστροφικό γεγονός του σεισμού.

Πρόδρομο φαινόμενο ενός μεγάλου επιφανειακού σεισμού αποτελεί και ο προσεισμός. Πλήθος μελετών έχουν δείξει πως οι προσεισμοί παρουσιάζουν υψηλού επιστημονικού ενδιαφέροντος ιδιότητες, οι οποίες παρουσιάζουν σημαντικές διαφορές σε σχέση με τους μετασεισμούς. Ο μικρός όμως αριθμός προσεισμών κάνει δύσκολη στατιστικά την εξεύρεση τους



Εικόνα 27. Χάρτης απεικόνισης σεισμών μεγάλους $M_L > 5$ στην Ελλάδα τα τελευταία 30 χρόνια Πηγή: Wolfram Alpha

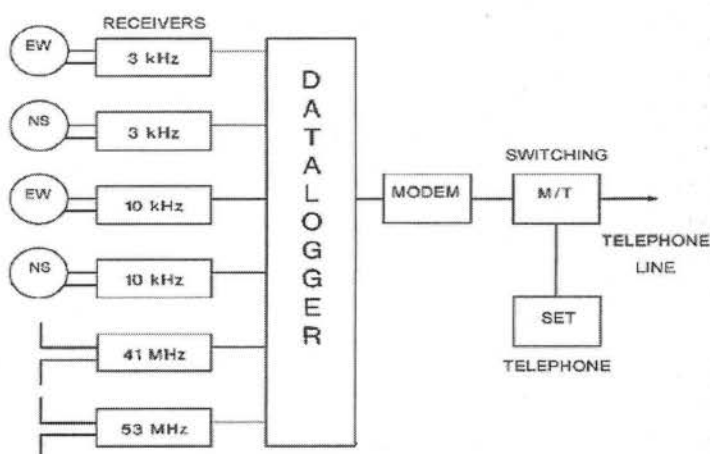


Εικόνα 28. Χάρτες απεικόνισεων σεισμών μεγάλους $M_L > 4$ και $M_L > 6$ στην Ελλάδα. Πηγή Wolfram Alpha

4. Τηλεματικό Δίκτυο Μέτρησης Ηλεκτρομαγνητικών Σημάτων

Με σκοπό την μέτρηση και παρακολούθηση ηλεκτρομαγνητικών διαφοροποιήσεων κατά την κρίσιμη περίοδο ενός σεισμικού φαινομένου δημιουργήθηκε ένας τηλεματικός σταθμός παρακολούθησης ηλεκτρομαγνητικών σημάτων MHz στην Κρήτη το 1998 (Nomikos & Valianatos, 1998). Οι μετρήσεις γίνονται σε κανάλια των 3 και 10 KHz και 41 και 53 MHz.

Οι διαφοροποιήσεις που συμβαίνουν στο γήινο μαγνητικό πεδίο θα μπορούσαν να θεωρηθούν ως ένα ενδιαφέρουσες για την μελέτη προσεισμικών σημάτων (Shapiro, et al., 1994). Διαφόρων επιπέδων συχνότητες έχουν συσχετιστεί με σεισμικές διαταραχές στην βιβλιογραφία. Πιο συγκεκριμένα στις εξαιρετικά χαμηλές συχνότητες (ULF) 0,01-10 Hz (Frazer - Smith, et al., 1990), σε συχνότητες 0,1-100 KHz (ELF-VLF) (Nomikos, et al., 1994) και σε συχνότητες υψηλών συχνοτήτων με $f > 1$ MHz (Nomikos, et al., 1997). Το τηλεματικό δίκτυο είχε αρχικά εγκατασταθεί στην περιοχή της Κρήτης. Η συγκεκριμένη περιοχή επιλέχθηκε ως κεντρικό μέρος του ελληνικού γεωλογικού τόξου, με πλήθος σεισμικών φαινομένων μεσαίου βάθους να έχουν συμβεί τα τελευταία 100 χρόνια. Το πρώτο δίκτυο είχε 4 σταθμούς κατά μήκος της Κρήτης. Κάθε σταθμός μέτρησης πεδίου χρησιμοποιεί συντονισμένες κεραίες βρόχου (tuned loop antenna) για την μέτρηση χαμηλών συχνοτήτων σε 3 και σε 10 KHz. Με δίπολα ημι-μήκους κύματος (half wavelength dipoles) μετράμε υψηλές συχνότητες πχ. Σε 41 και 53 MHz. Ο κεντρικός σταθμός επικοινωνούσε με έναν καταγραφέα δεδομένων (data-logger) και μετέδιδε τα δεδομένα μέσω τηλεφωνικής γραμμής (Nomikos & Valianatos, 1998).



Εικόνα 29. Σχηματικό Τηλεματικού Σταθμού Μέτρησης Η/Μ διαταρχών (Nomikos & Valianatos, 1998)

Κάθε δέκτης μέτρησης διαφοροποιήσεων ηλεκτρομαγνητικού πεδίου 3 και 10 kHz είχε κατεύθυνση Ανατολής-Δύσης και Νότου-Βορά. Οι δέκτες είναι κατασκευασμένοι χρησιμοποιώντας ευρείας ζώνης και χαμηλού θορύβου ενισχυτές και ζωνοπερατά φίλτρα (band pass) συντονισμένα με κρυσταλλικούς ταλαντωτές. Καθώς και μετατροπείς RMS σε DC.

Επίσης υπάρχουν δύο δέκτες για την μέτρηση του διαφοροποιήσεων ηλεκτρομαγνητικού πεδίου σε 41 και 53 MHz. Οι δέκτες είναι κατασκευασμένοι χρησιμοποιώντας τεχνολογία διπλής υπερ-ηχητικής ετεροδύνησης (double super heterodyne technology). Μέρος του συστήματος αποτελεί και ο καταγραφέας δεδομένων, ο οποίος αποτελεί το κύριο όργανο για το διάβασμα της αναλογικής πληροφορίας των ηλεκτρομαγνητικών δεκτών. Ο ρυθμός δειγματοληψίας ορίστηκε στο 1 sec. και μια μέση τιμή περίπου 60 δειγμάτων από κάθε κανάλι αποθηκεύεται στην τελική μνήμη. Ένα από τα πιο σημαντικά προβλήματα που αντιμετωπίστηκαν ήταν η εύρεση συχνοτήτων οι οποίες δεν θα περιείχαν θόρυβο από άλλες πηγές. Για αυτόν τον σκοπό λήφθηκαν μετρήσεις σε διάφορες συχνότητες για αρκετούς μήνες, πριν γίνει η κατάληξη στις προ-αναφερθείς συχνότητες. (Nomikos & Valianatos, 1998).

Από την στιγμή της ύπαρξης του μέχρι σήμερα το τηλεματικό δίκτυο ηλεκτρομαγνητικών μετρήσεων υπέστη αρκετές αλλαγές και αναβαθμίσεις. Στην σημερινή του μορφή αποτελείται από 12 σταθμούς διασκορπισμένους σε όλη την ελληνική επικράτεια για την καλύτερη δυνατή μελέτη και επίβλεψη των ελληνικών σεισμικών φαινομένων.



*Εικόνα 30. Τηλεματικό Δίκτυο
Μετρήσεων MHz/KHz συχνοτήτων*

Ιδιαίτερη αναβάθμιση υπήρξε στο τομέα της καταγραφής των δεδομένων καθώς χρησιμοποιήθηκε ένα νέου είδους σύστημα αποθήκευσης τύπου compact flash

με χαμηλότερο κόστος και ενεργειακή κατανάλωση σε σχέση με άλλα συστήματα αποθήκευσης (Koulouras, et al., 2005).

Αναλυτικά οι σταθμοί με το κωδικό μονόγραμμα ανά περιοχή είναι οι ακόλουθοι:

Πίνακας 5. Κωδικοί Σταθμών Μετρήσεων

ΟΝΟΜΑ ΣΤΑΘΜΟΥ	ΚΩΔ. ΣΤΑΘΜΟΥ	ΟΝΟΜΑ ΣΤΑΘΜΟΥ	ΚΩΔ. ΣΤΑΘΜΟΥ
Καλαμάτα	O	Νεάπολις	E
Κεφαλονιά	F	Κέρκυρα	P
Κομοτηνή	T	Κοζάνη	K
Ιωάννινα	J	Αταλάντη	H
Μυτιλήνη	M	Ρόδος	A
Ζάκυνθος	Z	Βάμος	V

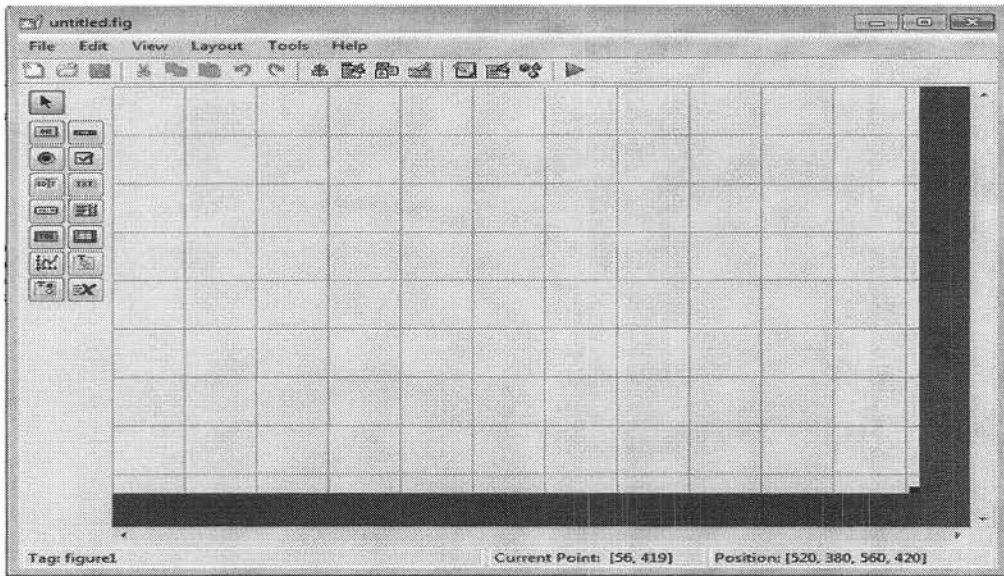
5. Εισαγωγή στο προγραμματισμό GUI με το MATLAB

Το λογισμικό Γραφικής Διεπαφής Χρήστη, είναι μια γραφική αναπαράσταση ενός ή περισσότερων παραθύρων που περιέχουν εργαλεία/κουμπιά που επιτρέπουν στον χρήστη να εκτελέσει διαδραστικές εργασίες. Εργαλεία (Components) του GUI μπορούν να συμπεριλαμβάνουν μενού, μπάρες, κουμπιά, listboxes κ.α. Τα GUI που δημιουργούνται με το MATLAB παρέχουν την δυνατότητα εκτύπωσης διαγραμμάτων.

Όπως και στα άλλα GUI, έτσι και τα MATLAB GUI περιμένουν από τον χρήστη να αλληλοεπιδρά με κάποιο από τα εργαλεία του GUI. Κάθε εργαλείο, καθώς και το ίδιο το GUI έχει μια ή περισσότερες ρουτίνες γνωστές ως “callback functions”, οι οποίες ξανά-καλούν το MATLAB για να πραγματοποιήσει κάποια διεργασία.

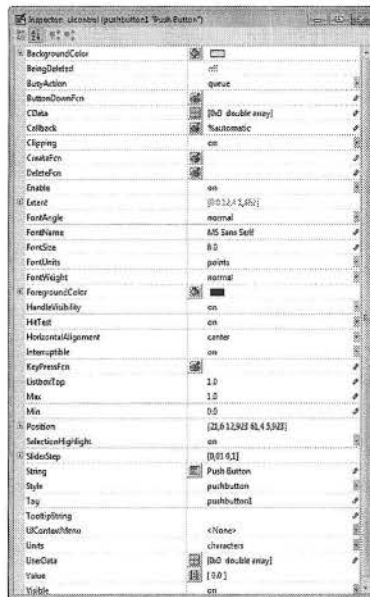
Ο συγκεκριμένος προγραμματισμός είναι γνωστός και ως ακολουθούμενος-από-το-γεγονός. Και η εκτέλεση του προγράμματος είναι ασύγχρονη καθώς ένα γεγονός όπως π.χ. Το πάτημα ενός κουμπιού οδηγεί στην εκτέλεση γεγονότων εξωτερικών της κύριας λειτουργίας του προγράμματος (Mathworks, 2012).

Το MATLAB έχει ως βασικό εργαλείο ανάπτυξης γραφικού περιβάλλοντος το GUIDE (Graphical User Interface Development Environment), το οποίο διαθέτει εργαλειοθήκη για την μεταφορά και δημιουργία εργαλείων στο προς ανάπτυξη project. Τα αρχεία τα οποία παράγει το GUIDE είναι τύπου .fig.



Εικόνα 31. Το περιβάλλον ανάπτυξης του GUIDE του MATLAB

Στην παραπάνω εικόνα διαφαίνεται το περιβάλλον ανάπτυξης του GUIDE. Ενώ κάτω μπορούμε να δούμε το παράθυρο των ιδιοτήτων για ένα οποιοδήποτε εργαλείο.



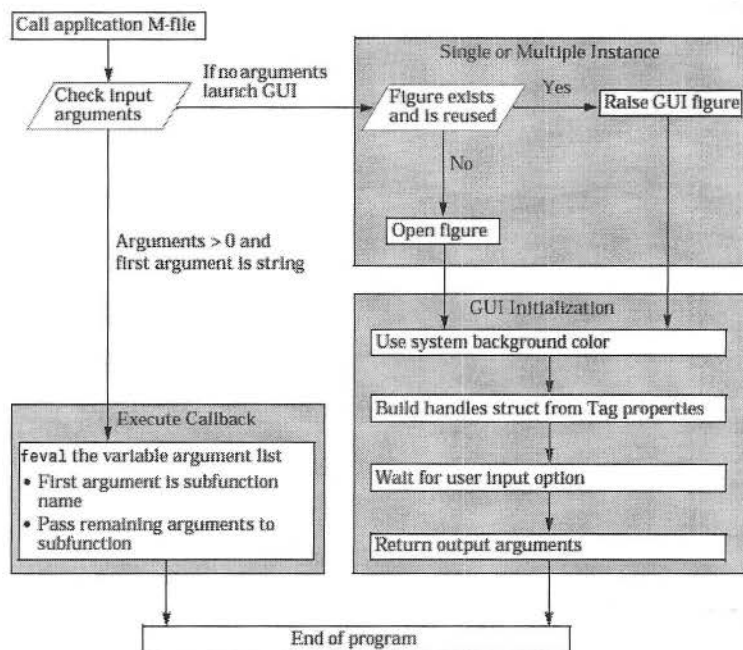
Εικόνα 32. Παράθυρο ιδιοτήτων ενός αντικειμένου στο GUIDE

Έτσι το πιο σημαντικό στοιχείο στην συγγραφή κώδικα GUI, είναι η δομή handles. Αυτή η δομή έχει δυο σκοπούς: α) Αφ' ενός την αποθήκευση όλων των αντικειμένων ελέγχου, μενού, αξόνων και άλλων εργαλείων που χρησιμοποιούνται στο GUI. β) Αφ' ετέρου την αποθήκευση global μεταβλητών. Το δεύτερο γίνεται καθώς οι μεταβλητές σε μια call back function έχουν τοπική ισχύ. Έτσι για να μπορέσουμε να αλλάξουμε την τιμή μιας μεταβλητής από την μια στην άλλη function θα πρέπει να την έχουμε αποθηκεύσει στην δομή handles.

Για να αποθηκεύσουμε μια global μεταβλητή αρκεί να χρησιμοποιήσουμε τις παρακάτω εντολές:

```
handles.x_data=X;
guidata(h, handles)
```

όπου h είναι ο χειριστής (handler) της callback function. Η πρώτη εντολή αποθηκεύει την μεταβλητή X στην μεταβλητή x_data η οποία είναι προσβάσιμη από την δομή handles. Η δεύτερη εντολή αναθεωρεί σε νέα έκδοση έχοντας αποθηκευμένη την νέα μεταβλητή. Την δεύτερη εντολή θα πρέπει να την βάζουμε σχεδόν πάντα στο τέλος κάθε function έτσι ώστε να αναθεωρούνται οι όποιες αλλαγές.



Εικόνα 33. Μονοπάτι εκτέλεσης ενός GUI m-file Πηγή: Mathworks

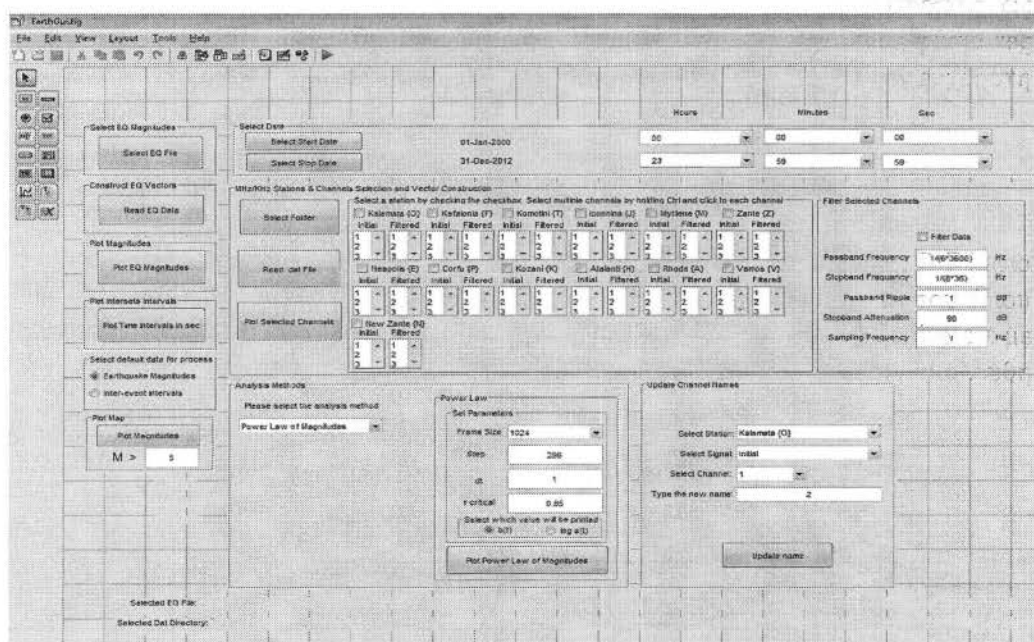
6. Ανάπτυξη λογισμικού Γραφικής Διεπαφής Χρήστη (GUI) για την διαχείριση μετρήσεων ηλεκτρομαγνητικού πεδίου από πολλαπλούς σταθμούς μέτρησης

Με σκοπό την παρατήρηση και αναπαράσταση των σεισμικών χρονοσειρών για την καταγραφή του μεγέθους ισχύος τους M_L , της ασαφούς εντροπίας ως επιλεγμένη εκ των τριών παρουσιασμένων εντροπιών και της πυκνότητας φάσματος ισχύος, αναπτύχθηκε σχετικό λογισμικό γραφικής διεπαφής χρήστη για την εύχρηστη διαχείριση του.

6.1 Περιγραφή Συστήματος

Ο κώδικας για την ανάπτυξη του GUI βρίσκεται στο Παράρτημα Β Εδάφιο 4. Το σύστημα έχει την δυνατότητα να ανοίγει πολλαπλά στιγμιότυπα του ίδιου προγράμματος μέσω του μενού (File-> New Instance). Δέχεται ως είσοδο ένα txt με τα σεισμικά μεγέθη των σεισμών. Τα δεδομένα στο .txt αρχείο έχουν την ακόλουθη μορφή:

2008 DEC 15	06 04 13.1	36.53	21.48	21	3.0
2008 DEC 15	06 25 32.4	36.54	21.62	20	3.4
2008 DEC 15	06 28 03.3	38.73	23.60	24	2.8



Εικόνα 34. Ανάπτυξη GUI μοντέλου στο GUIDE

Αναφέρουν δηλαδή της εξής πληροφορίες: α) Ημερομηνία και ώρα του καταγεγραμμένου σεισμού, β) το γεωμετρικό μήκος και το γεωμετρικό πλάτος του σεισμού γ) το εστιακό βάθος σε χιλιόμετρα και δ) το μέγεθος του καταγεγραμμένου σεισμού. Στην συνέχεια τα δεδομένα αυτά διαβάζονται γίνονται διανύσματα για περαιτέρω χρήση και επεξεργασία με μαθηματικές/στατιστικές μεθόδους.

Το σύστημα έχει την δυνατότητα εκτός από την εκτύπωση των σεισμών να υπολογίσει και να τυπώσει τις χρονικές διαφοροποιήσεις μεταξύ των σεισμικών φαινομένων. Για την επιλογή του προς μελέτη χρονικού διαστήματος το σύστημα έχει την δυνατότητα επιλογής της κατάλληλης ημερομηνίας έναρξης και λήξης του γεγονότος μέσω νέο συστήματος GUI με ενσωματωμένο calendar. Η διαλογή του χρόνου μπορεί να γίνει με ακρίβεια δευτερολέπτου.

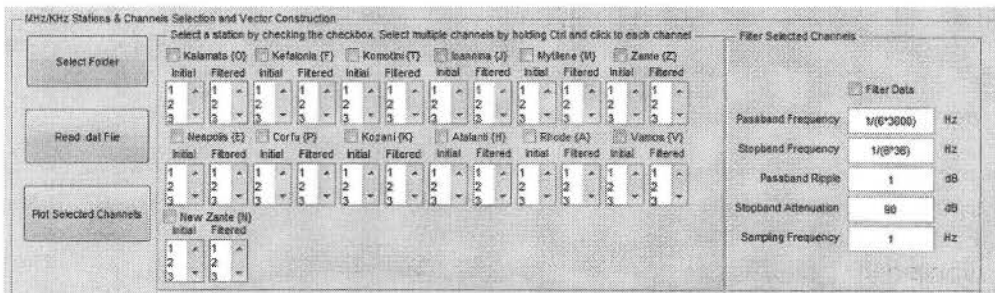


Εικόνα 35. Παράθυρο διεπαφής για την επιλογή ημερομηνίας

Η δεύτερη είσοδος για το σύστημα είναι τα MHz/KHz 'σήματα τα οποία αποκαλούνται κανάλια. Τα αρχεία που δέχεται το σύστημα ως είσοδο είναι τύπου .dat και είναι ένα ανά ημέρα. Η ονομασία του κάθε αρχείου ακολουθεί ένα συγκεκριμένο πρότυπο δηλαδή το πρώτο γράμμα είναι η κωδική ονομασία του σταθμού, στην συνέχεια χωρίς κενά ακολουθεί το ημερολογιακό έτος που έγινε η μέτρηση και τελικώς υπάρχει και ένα τριψήφιο νούμερο το οποίο καταδुकνείει την ημέρα που έγινε η μέτρηση (κινείται δηλαδή στο διάστημα 001-365). Κάθε .dat περιέχει όλα τα κανάλια μετρήσεων και τα δεδομένα του είναι της μορφής:

```
102,0013.,0000.,0.000,2061.,1662.,1086.,10.06,256.8,20.12,0928.,1107.,13.51
102,0013.,0000.,1.000,2060.,1661.,1086.,10.06,238.1,19.45,0901.,1111.,13.51
102,0013.,0000.,2.000,2059.,1664.,1088.,10.06,237.4,19.78,0887.,1111.,13.51
102,0013.,0000.,3.000,2057.,1666.,1087.,10.06,227.3,20.12,0890.,1112.,13.51
102,0013.,0000.,4.000,2057.,1664.,1088.,10.06,223.3,20.12,0891.,1113.,13.52
102,0013.,0000.,5.000,2058.,1664.,1088.,10.06,220.6,20.12,0894.,1107.,13.51
102,0013.,0000.,6.000,2059.,1664.,1087.,10.06,232.7,20.12,0897.,1100.,13.51
102,0013.,0000.,07.00,2059.,1662.,1086.,10.06,230.7,20.12,0891.,1092.,13.51
102,0013.,0000.,08.13,2059.,1662.,1086.,10.06,232.7,20.12,0919.,1047.,13.52
```

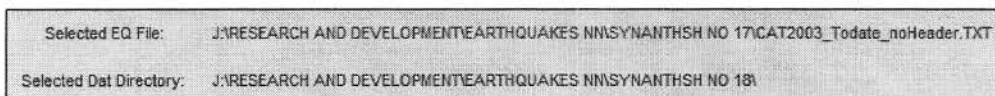
Εξαιρουμένων κάποιων σταθερών τιμών οι οποίες αποτελούν θερμοκρασία ή τάση του συστήματος μέτρησης ή κάποιο κωδικό καταγραφής, όλα τα υπόλοιπα αποτελούν κανάλια μέτρησης. Στην παρακάτω εικόνα φαίνονται όλες οι επιλογές που έχει ο χρήστης. Πιο συγκεκριμένα με το κουμπί Select Folder ο χρήστης διαλέγει τον φάκελο στον οποίο βρίσκονται οι μετρήσεις του. Το σύστημα αναμένει σε εκείνο τον φάκελο να υπάρχουν υποφάκελοι, με καθένα από αυτούς να φέρει ως όνομα ένα ημερολογιακό έτος. Στην συνέχεια ο χρήστης επιλέγει κάνοντας check τον προς μελέτη σταθμό.



Εικόνα 36. Επιλογή καναλιών απο το GUI και φιλτράρισμα τους

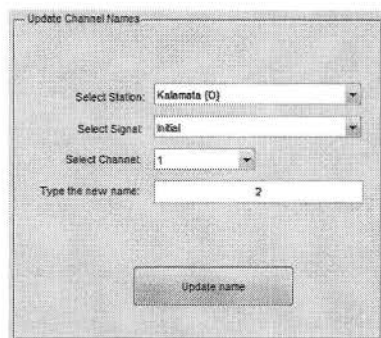
Σε κάθε φάκελο ημερολογιακού έτους υπάρχουν υποφάκελοι με την κωδική ονομασία του κάθε σταθμού. Αφού πατηθεί το κουμπί Read .dat File γίνεται το διάβασμα η δημιουργία vector από το MATLAB. Στην συνέχεια έχουμε την δυνατότητα να επιλέξουμε συγκεκριμένα ποια κανάλια θέλουμε και να τα φιλτράρουμε. Τα φιλτραρισμένα κανάλια μαζί με τα αρχικά μπορούμε να τα εκτυπώσουμε καλώντας το δεύτερο GUI το οποίο θα παρουσιάσουμε παρακάτω. Το όλο σύστημα μπορεί να θεωρηθεί δυναμικό καθώς υπάρχει η δυνατότητα μελέτη διαφορετικών καναλιών από διαφορετικούς σταθμού.

Στο κάτω μέρος του GUI μπορεί να φανεί το επιλεγμένο αρχείο .txt για τα μεγέθη σεισμικότητας και ο επιλεγμένος φάκελος όπου βρίσκονται τα .dat αρχεία.



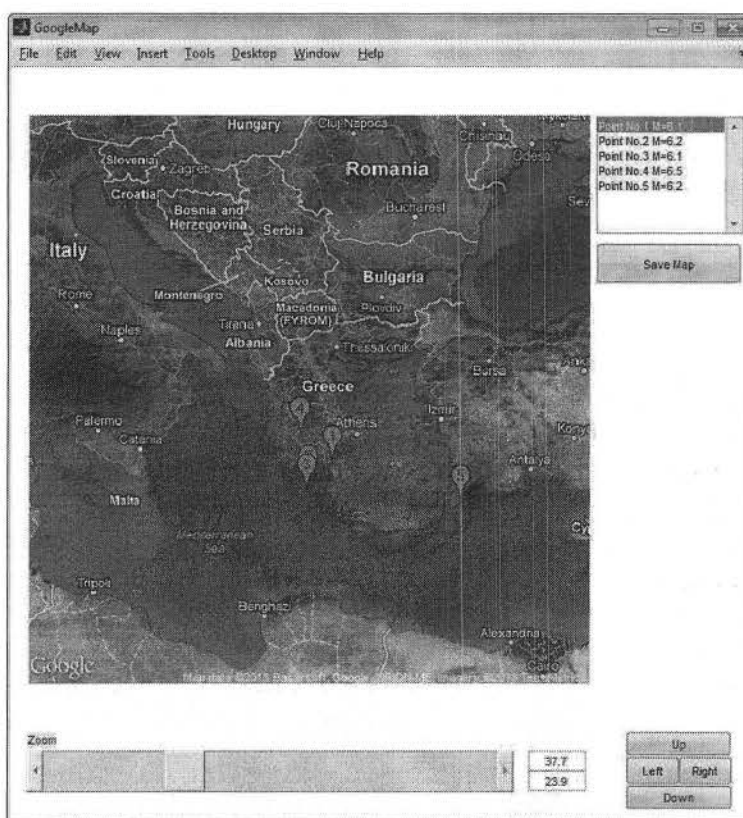
Εικόνα 37. Προβολή διαδρομών επιλεγόμενων αρχείων και φακέλων προς μελέτη

Ο χρήστης σε περίπτωση που το επιθυμεί μπορεί να αλλάξει την ονομασία οποιoδήποτε καναλιού. Η αρχική ονομασία είναι ένας αριθμός από το 1 έως 7.



Εικόνα 38. Παράθυρο αλλαγής ονομασίας καναλιού

Με το κουμπί Plot Magnitudes στα αριστερά και τον ορισμό της σεισμικότητας μπορούμε να τυπώσουμε τον ελληνικό χάρτη με όλους τους σεισμούς. Παρακάτω παρατίθεται ένα παράδειγμα για το ημερολογιακό έτος 2008 παρουσιάζονται όσοι σεισμοί είναι μεγαλύτεροι από $M_L=6$



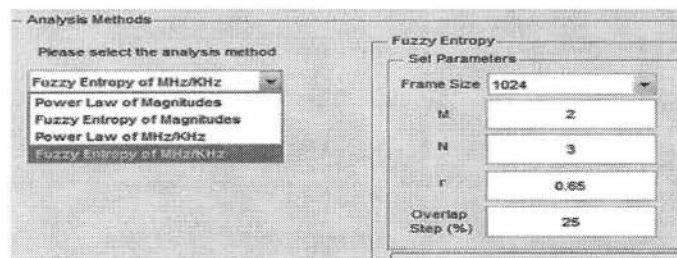
Εικόνα 39. Χρήση Google Map για την αποτύπωση σεισμών με $M_L > 6$ για το 2008.

Στον χάρτη υπάρχει η δυνατότητα για zoom και μετακίνηση. Όλα τα σημεία έχουν ένα κωδικό αριθμό όπου στα δεξιά υπάρχει μια λίστα με τα σεισμικά μεγέθη τους. Έχει χρησιμοποιηθεί το standard menu για την επεξεργασία των figures του Matlab. Ενώ μπορούμε να αποθηκεύσουμε την εικόνα. Οι χάρτες οι οποίοι χρησιμοποιούνται είναι τύπου Google Maps, ενώ για τον προγραμματισμό και την ανάπτυξη του συγκεκριμένου έγινε χρήση του Google API των static χαρτών. Προς επαλήθευση των ανωτέρω αποτελεσμάτων παρατίθεται ο αντίστοιχος χάρτης του Wolfram Alpha. Όπως βλέπουμε από την σύγκριση επιβεβαιώνεται η καλή λειτουργία.



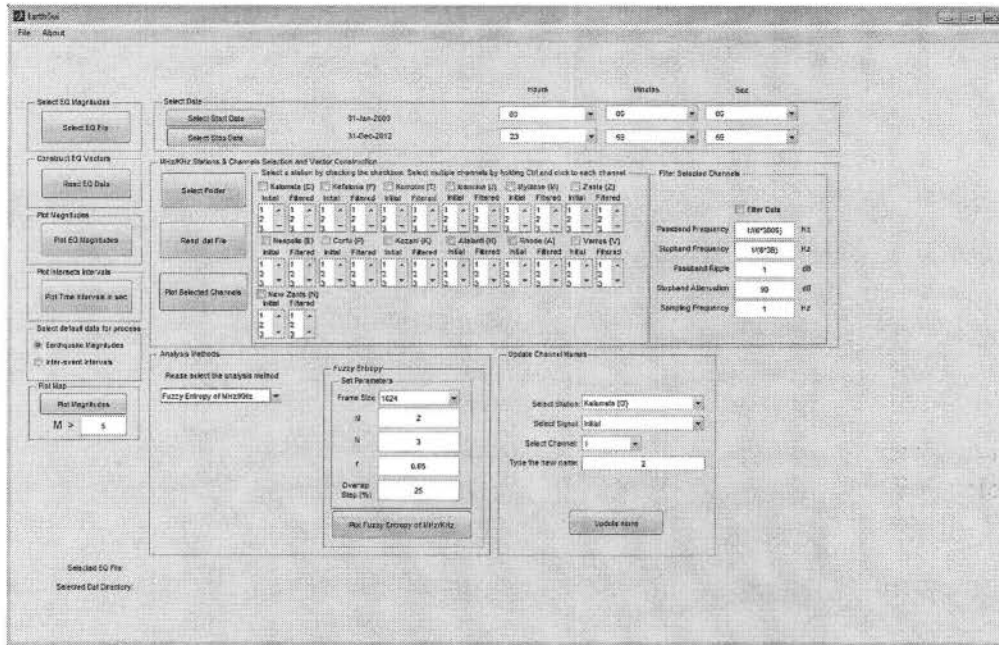
Εικόνα 40. Οι σεισμοί στον Ελληνικό χώρο κατά το 2008 με $ML > 6$ Πηγή: Wolfram Alpha

Στην παρακάτω εικόνα μπορούμε να δούμε την δυνατότητα μελέτης και παραμετροποίησης των μεθόδων Ασαφούς εντροπίας και Πυκνότητας Φάσματος Ισχύος.



Εικόνα 41. Επιλογή μεθόδων για ανάλυση σημάτων απο το GUI

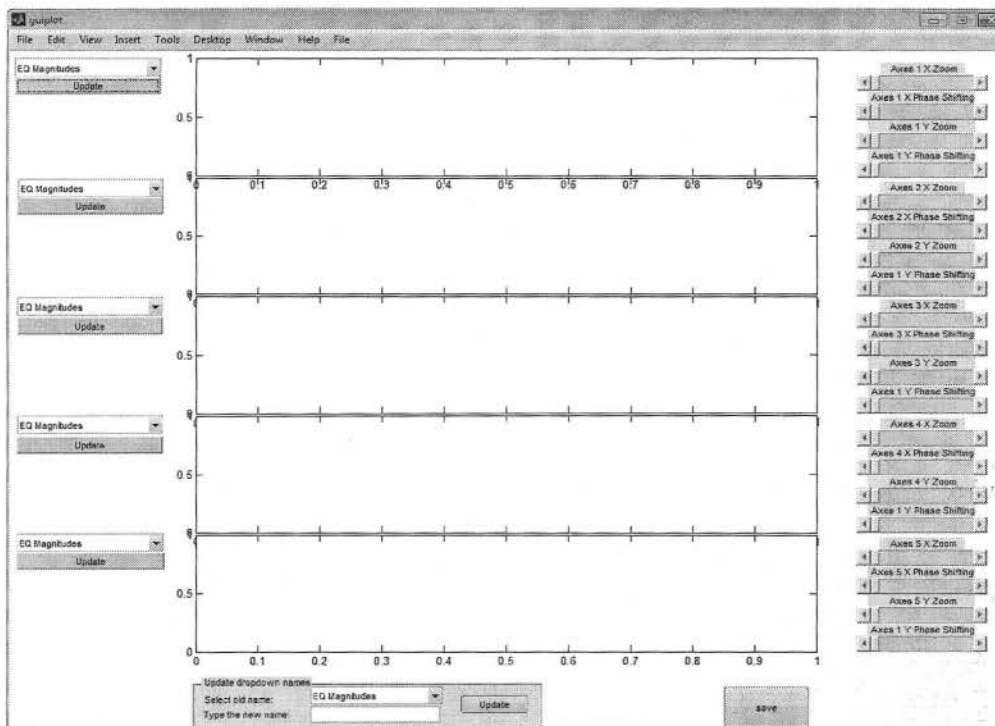
Ολοκληρωμένο το σύστημα GUI για την διαχείριση μετρήσεων ηλεκτρομαγνητικού πεδίου από πολλαπλούς σταθμούς μέτρησης βρίσκεται παρακάτω:



Εικόνα 42. Σύστημα GUI για την διαχείριση μετρήσεων ηλεκτρομαγνητικού πεδίου απο πολλαπλούς σταθμούς μέτρησης

Για να αποτυπωθούν οι πολλαπλές εκτυπώσεις καναλιών, σεισμικών διαταραχών και μεθόδων σε αυτά δημιουργήθηκε ένα GUI την εικόνα του οποίου μπορούμε να δούμε στην παρακάτω εικόνα. Μεταξύ των δύο αναπτυχθέντων GUI υπάρχει επικοινωνία. Έτσι σε κάθε κλήση του δεύτερου GUI μια σειρά μεταβλητών τίθενται ως είσοδοι από το πρώτο GUI. Και σε αυτό το GUI υπάρχει η δυνατότητα για αλλαγή των ονομάτων που βρίσκονται στο dropdown μενού. Ενώ υπάρχει η δυνατότητα για κάθε γράφημα να ζουμάρω σε μήκος και σε ύψος καθώς και να μετακινώ την φάση του με τις αλλαγές να φαίνονται σε ζωντανό χρόνο χωρίς να απαιτείται κάποια ανανέωση. Το κουμπί update υπάρχει κάτω από κάθε επιλογή του dropdown μενού, για την ανανέωση του διπλανού γραφήματος. Τελικά το σύστημα έχει την δυνατότητα αποθήκευσης της εικόνας των γραφημάτων. Θα πρέπει εδώ να αναφέρουμε πως ένα GUI σχεδιάζεται πάνω σε MATLAB figures, έτσι όταν θέλουμε να αποθηκεύσουμε τα γραφήματα θα αποθηκεύονται μαζί και οι αναπαραστάσεις των κουμπιών κλπ. Για τον λόγο αυτό την στιγμή που ο χρήστης πατήσει save όλα τα κουμπιά εξαφανίζονται στιγμιαία, γίνεται η αποθήκευση και στην συνέχεια επανεμφανίζονται. Υπάρχει το standard μενού των figures το οποίο μπορεί να χρησιμοποιηθεί για την επεξεργασία καθ' ενός εκ των γραφημάτων. Κατά την διάρκεια της αποθήκευσης η ονομασία των γραφημάτων πηγαίνει εναλλάξ, για την πιο

εύχρηστη απεικόνιση. Ενώ σε περίπτωση που το επιθυμούσαμε θα ήταν δυνατό να μην φαίνονται και τα κουτάκια που περιέχουν τα γραφήματα, κάνοντας τα μοιάζουν με ένα ενιαίο γράφημα.

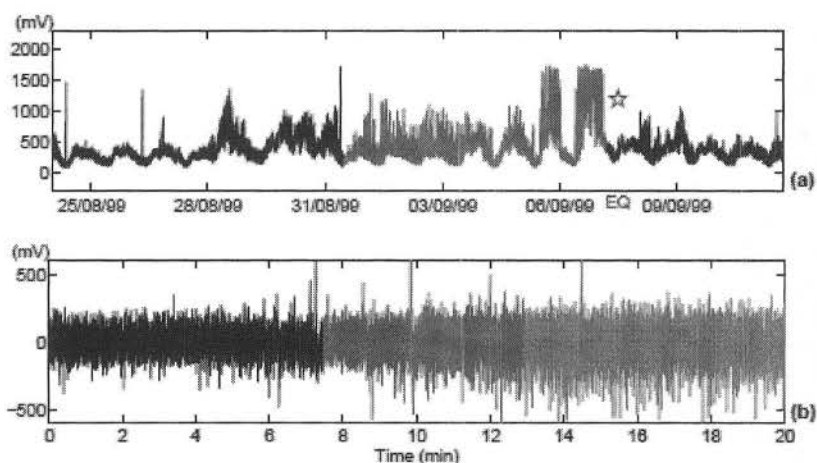


Εικόνα 43. Λογισμικό GUI για την αποτύπωση MHz/KHz καναλιών, σεισμικών διαταραχών και μεθόδων

7. Ομοιότητες στην εξέλιξη των γεωλογικών και βιολογικών καταστροφικών φαινομένων

Με την έννοια καταστροφικό φαινόμενο θεωρούμε το φαινόμενο ενός σεισμού από γεωφυσικής απόψεως ή το φαινόμενο μιας επιληπτικής κρίσης από βιολογικής απόψεως. Οι προσεισμικές ηλεκτρομαγνητικές εκπομπές παρουσιάζουν αρκετό ενδιαφέρον για την μελέτη τους ως προπομπού ενός επερχόμενου σεισμού. Από πλευράς πολυπλοκότητας, είναι εντυπωσιακή η ομοιότητα στην συμπεριφορά καθώς βαίνουμε σε μια μη-αναστρέψιμη φάση καθώς οδεύουμε προς ένα καταστροφικό φαινόμενο όπως αυτό του σεισμού με άλλα συστήματα τελείως διαφορετικού χαρακτήρα (Eftaxias, et al., 2006). Σε θεωρητικές έρευνες της προηγούμενης δεκαετίας έχει υποστηριχθεί πως η δυναμική των σεισμών και των νευρικών σπασμών κατά το τελικό τους στάδιο παρουσιάζουν ομοιότητες στα

χαρακτηριστικά τους (Hertz & Hopfield, 1995), (Rundle, et al., 2002). Τις θεωρίες αυτές ενίσχυσε ο Eftaxias το 2006, με μια σειρά μαθηματικών μεθόδων παρουσίασε



Εικόνα 44. Διαφορετικές φάσεις χρονοσειρών σεισμού Αθήνας 1999 και επιληψίας σε ποντίκι (Eftaxias, et al., 2006)

τις ομοιότητες μεταξύ δύο καταστροφικών φαινομένων α) του σεισμού της Αθήνας του 1999 και β) της επιληπτική κρίσης σε ποντίκι. Παρακάτω φαίνονται οι δυο προς μελέτη χρονοσειρές.

Στην πρώτη χρονοσειρά βλέπουμε για τα 10 KHz τον σεισμό (ο οποίος αναπαρίσταται με το αστέρι) της Αθήνας του 1999 3 φάσεις, την φάση ηρεμίας τους συστήματος, την προσεισμική φάση και την μετασεισμική φάση. Στην δεύτερη χρονοσειρά βλέπουμε το ηλεκτροεγκεφαλογράφημα (EEG - Electroencephalogram) ενός ποντικού. Η πράσινη περίοδος δείχνει την φάση της ηρεμίας, η πορτοκαλί απεικονίζει την προ-επιληπτική φάση και στην τελευταία περίοδο φαίνεται η φάση της επιληψίας. Οι μαθηματικές μέθοδοι οι οποίοι εφαρμόστηκαν είναι οι ακόλουθες: α) πυκνότητας φάσματος ισχύος β) συντελεστή Hurst γ) Persistence/Anti-persistence δ) T-πολυπλοκότητα ε) φρακταλική διάστασης ζ) Διάστασης συσχέτισης (Correlation dimension) η) Προσεγγιστικής εντροπίας. Όπως παρατηρήθηκε από την εκτενή μελέτη των παραπάνω μεθόδων και τα δύο είδη καταστροφικών φαινομένων ακολουθούν μια κοινή συμπεριφορά κατά την διάρκεια της προ-καταστροφικής φάσης. Τα ευρήματα των μεθόδων αυτών ενισχύουν την υπόθεση ότι η ευρεθείσα ηλεκτρομαγνητική ανωμαλία προ-έρχεται κατά την διάρκεια των μικρο-ρωγμών της εστιακής περιοχής του σεισμού. Πιο συγκεκριμένα στο πεδίο της μη-γραμμικής ανάλυσης (βλ. T-πολυπλοκότητα, διάσταση συσχέτισης, προσεγγιστική εντροπία) δυσκνείται μια μετάβαση από ένα καθεστώς υψηλής πολυπλοκότητας σε ένα καθεστώς χαμηλής πολυπλοκότητας καθώς προσεγγίζουμε ένα καταστροφικό γεωλογικό ή βιολογικό φαινόμενο. Στο πεδίο της ανεξάρτητης γραμμικής ανάλυσης (βλ. Πυκνότητα φάσματος ισχύος) η μετάβαση από μία anti-persistent (στα ελληνικά το

persistent μεταφράζεται ως επίμονη) κατάσταση σε μια persistent κατάσταση δυκνώνει επίσης ότι προσεγγίζεται το καταστροφικό φαινόμενο. Τονίζεται δε, ότι η μετάβαση από την υψηλή πολυπλοκότητα στην χαμηλή καθώς και η μετάβαση από την non-persistent κατάσταση στην persistent κατάσταση συμπίπτουν χρονικά. Οι δυναμικές παράμετροι T- πολυπλοκότητας, Διάσταση συσχέτισης, προσεγγιστικής εντροπίας και συντελεστή Hurst φαίνεται να αντιπροσωπεύουν ένα μέτρο της απόστασης του συστήματος από την γενική αστάθεια. Όσο χαμηλότερες είναι οι τιμές της T- πολυπλοκότητας, της διάστασης συσχέτισης, της προσεγγιστικής εντροπίας και υψηλότερες τιμές για τον συντελεστή Hurst, τόσο μικρότερη είναι και η απόσταση του συστήματος από το κρίσιμο σημείο της γενικής αστάθειας (Eftaxias, et al., 2006). Τα παραπάνω ευρήματα κρίνουν σκόπιμη την μελέτη πολύπλοκων φυσικών και βιολογικών διεργασιών κάτω από ένα κοινό φάσμα μεθόδων.

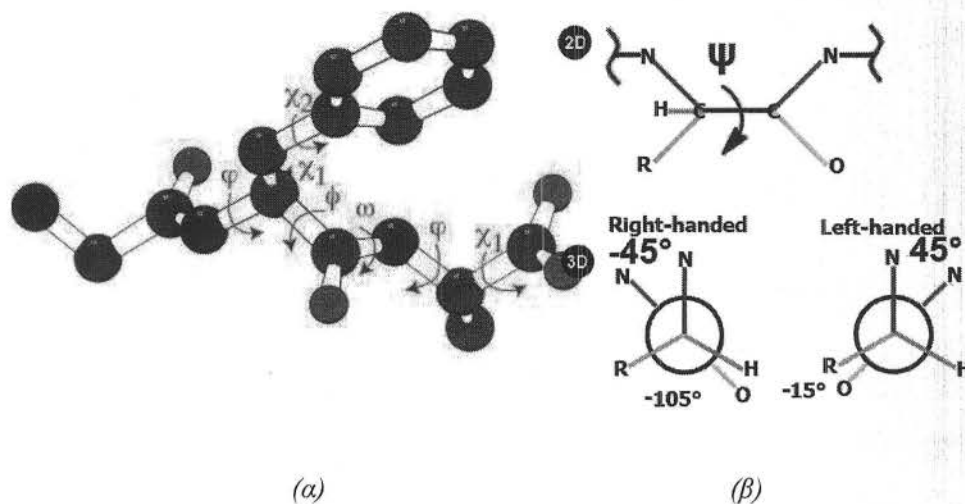
8. Μελέτη Πολύπλοκων Βιολογικών διεργασιών

8.1 Πρόβλεψη Πρωτεΐνης Δευτέρας Δομής με μοντέλο Νευρωνικών Δικτύων

8.1.1 Εισήγηση στο Βιολογικό μέρος

Πρόκειται για την πρόβλεψη της τρισδιάστατης δομής μια πρωτεΐνης από την αλληλουχία των αμινοξέων της. Η πρόβλεψη δύναται να είναι α) δευτεροβάθμια β) τριτοβάθμια και γ) τεταρτοταγής σε σχέση με την πρωτογενή δομή του (Zhang, 2008). Η πρόβλεψη της πρωτεϊνικής δομής είναι ένας από του σημαντικότερους στόχους που επιδιώκονται από την πληροφορική και την θεωρητική χημεία. Η πρακτική εφαρμογή έγκειται στον σχεδιασμό φαρμάκων (επιστήμη φαρμακευτικής) και στον σχεδιασμό νέων ενζύμων (επιστήμη βιοτεχνολογίας). Κάθε δύο χρόνια οι μέθοδοι εκτιμούνται με το πείραμα CASP (Critical Assessment of Techniques for Protein Structure Prediction). Η δομή των πρωτεϊνών μπορεί να θεωρηθεί ως μια ακολουθία δευτερευόντων στοιχείων δομής, όπως οι α-έλικες και τα β-φύλλα, τα οποία συνιστούν την τρισδιάστατη αναπαράσταση της πρωτεϊνικής αλυσίδας. Στις δευτερεύουσες δομές πρότυπα Η-δεσμών σχηματίζονται μεταξύ των γειτονικών αμινοξέων, με τα αμινοξέα να διατηρούν παρόμοιες Φ και Ψ γωνίες.

Ο σχηματισμός των δομών αυτών εξουδετερώνει τις πολικές ομάδες σε κάθε αμινοξύ. Οι δευτερεύουσες δομές είναι στενά συνδεδεμένες στον πρωτεϊνικό πυρήνα σε ένα υδροφοβικό περιβάλλον. Κάθε ομάδα αμινοξέων έχει πεπερασμένο αριθμό πιθανών αλληλεπιδράσεων με γειτονικές αλυσίδες, κατάσταση η οποία θα πρέπει να ληφθεί υπόψη στην μοριακή μοντελοποίηση.



Εικόνα 45. (α) Αναπαράσταση διέδρης γωνίας της δομής της πρωτεΐνης. Υπάρχουν 3 γωνίες έκφρασης: Φ , Ψ , ω . Το ω είναι η γωνία στρέψης, του οποίου ο άξονας είναι ο πεπτιδιακός δεσμός και συνήθως είναι ορισμένος στις 180° . Οι πλευρικές αλυσίδες αναπαρίστανται από μια ή περισσότερες γωνίες χ . (β) Αναπαράσταση γωνιών σε 2 και 3 διαστάσεις (Vasquez, 1996)

8.1.2 Κατηγοριοποίηση Πρωτεΐνης

Οι πρωτεΐνες μπορούν να ταξινομηθούν με βάση τις ομοιότητες τόσο των δομών όσο και των ακολουθιών. Η κατηγοριοποίηση με βάση την ακολουθία ήταν βιβλιογραφικά η πρώτη η οποία χρησιμοποιήθηκε. Αρχικά, οι ομοιότητες βρίσκονταν με βάση την ευθυγράμμιση του συνόλου των ακολουθιών. Μετέπειτα, οι πρωτεΐνες κατηγοριοποιούνταν με βάση την εκδήλωση προτύπων των διατηρημένων αμινοξέων. Στην διεθνή βιβλιογραφία υπάρχουν βάσεις δεδομένων που κατηγοριοποιούν τις πρωτεΐνες με βάση ένα ή περισσότερα από τα παραπάνω συστήματα. Κατά την μελέτη των συστημάτων, είναι σημαντικό να κρατήσουμε τα εξής: α) Δύο εντελώς διαφορετικές πρωτεϊνικές ακολουθίες, από διαφορετικές εξελικτικές ρίζες μπορούν να αναδιπλωθούν σε παρόμοια δομή β) Αντίθετα η ακολουθία ενός πρότερου χρονικά γονιδίου, μιας δοσμένης ακολουθίας, δύναται να αποκλίνει σημαντικά σε διαφορετικά είδη ενώ ταυτόχρονα διατηρούνται τα ίδια βασικά χαρακτηριστικά. Η αναγνώριση οποιασδήποτε εναπομένουσας ακολουθίας σε τέτοιες περιπτώσεις μπορεί να είναι αρκετά δύσκολο έργο. Επίσης δύο πρωτεΐνες που μοιράζονται έναν σημαντικό βαθμό ομοιότητας στην ακολουθία τους, είτε μεταξύ τους είτε με μια τρίτη ακολουθία μοιράζονται και μια εξελικτική προέλευση, ενώ θα πρέπει να μοιράζονται και κάποια δομικά χαρακτηριστικά. Ωστόσο, η γενετική αλληλοεπικάλυψη και οι γενετικές ανακατατάξεις κατά την διάρκεια της εξέλιξης μπορούν να οδηγήσουν σε νέα γενετικά αντίγραφα, τα οποία μπορούν να εξελιχθούν σε πρωτεΐνες με νέα δομή και λειτουργία (Mount, 2004), (Guzzo, 1965).

8.1.3 Ανάλυση της Πρόβλεψης Δευτερεύουσας Δομής

Η πρόβλεψη δευτερεύουσας δομής είναι μια σειρά τεχνικών στην πληροφορική που στοχεύει στην πρόβλεψη των δευτεροβάθμιων δομών των πρωτεϊνών και των RNA ακολουθιών βασιζόμενη μόνο στην γνώση των της πρωτογενούς δομής τους, των αμινοξέων ή της ακολουθίας των νουκλεοτιδίων αντίστοιχα. Για τις πρωτεΐνες, μια πρόβλεψη αποτελείται από την ανάθεση των περιοχών της ακολουθίας των αμινοξέων σε α -έλικες και β -φύλλα ή αναδιπλώσεων. Η επιτυχία μιας πρόβλεψης προσδιορίζεται μέσω σύγκρισης των αποτελεσμάτων με αλγόριθμους που εφαρμόζονται στην κρυσταλλική δομή της πρωτεΐνης. Προχωρημένοι αλγόριθμοι μεγάλου βαθμού πολυπλοκότητας έχουν αναπτυχθεί για την ανίχνευση συγκεκριμένων προτύπων όπως διαμεμβρανικές έλικες στις πρωτεΐνες. Ο υψηλός βαθμός πρόβλεψης επιτρέπει την χρησιμοποίηση των προβλέψεων αυτών για την αναγνώριση πτυχών, την κατηγοριοποίηση δομικών μοτίβων και την βελτιστοποίηση της στοιχίσεως μια ακολουθίας (Zhang, 2008), (Constantini, et al., 2006), (Lyngso & Pedersen, 2000).

8.1.4 Χρήση Νευρωνικών Δικτύων για την πρόβλεψη δομής

Διάφορα μοντέλα τεχνητών νευρωνικών δικτύων έχουν αναπτυχθεί κατά καιρούς από την επιστημονική κοινότητα για την πρόβλεψη πρωτεϊνικών δομών. Τα νευρωνικά δίκτυα χρησιμοποιούσαν σειρά επιλυμένων δομών για την εκπαίδευση τους που προσδιόριζαν κοινά μοτίβα ακολουθιών, που σχετίζονταν με ειδικές ρυθμίσεις των δευτεροβάθμιων δομών. Αυτές οι μέθοδοι είναι σε ποσοστό άνω του 70% ακριβείς στις προβλέψεις τους, παρ' όλο που τα β -φύλλα είναι ακόμα συχνά απρόβλεπτα λόγω της ελλείψεως τρισδιάστατων δομών πληροφοριών που θα επέτρεπαν την αξιολόγηση των προτύπων των δεσμών υδρογόνου.

Οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines) έχουν αποδειχθεί ιδιαίτερα χρήσιμες στην πρόβλεψη των θέσεων των στροφών (locations of turns), οι οποίες είναι δύσκολο να εντοπιστούν με στατιστικές μεθόδους. Η απαίτηση σχετικά μικρών σε αριθμό δεδομένων εκπαίδευσης έχει επίσης αναφερθεί ως ένα πλεονέκτημα για την αποφυγή του overfitting σε υπάρχουσα δεδομένα δομών (Pham, et al., 2005), (Chen, et al., 2006).

Ως επέκταση των ανωτέρω υβριδικά μοντέλα που χρησιμοποιούν και νευρωνικά και SVMs για την πρόβλεψη τοπικών ιδιοτήτων των πρωτεϊνών όπως διεδρικές γωνίες σε μη-καταχωρημένες περιοχές (Zimmermann & Hansmann, 2006).

8.1.5 Ανάπτυξη Μοντέλου Τεχνητών Νευρωνικών Δικτύων για την πρόβλεψη

Σε αυτό το κεφάλαιο θα περιγράψουμε τα στάδια ανάπτυξης, εκπαίδευσης, ανάλυσης, και βελτιστοποίησης του τεχνητού νευρωνικού δικτύου για την πρόβλεψη. Ο τύπος του νευρωνικού δικτύου που θα χρησιμοποιηθεί είναι αυτός του feedforward.

Χρησιμοποιούμε τα δεδομένα των Rost-Sander τα οποία αποτελούνται από πρωτεΐνες των οποίων οι δομές καλύπτουν ένα σχετικά ευρύ φάσμα τύπων, συνθέσεων και μήκους (αλληλουχιών) (Rost & Sander, 1993). Στο αρχείο RostSanderDataset.mat περιέχονται υποσύνολα αυτών των δομών δεδομένων, όπου η δομική ανάθεση (structural assignment) αναφέρεται για κάθε μια ακολουθία πρωτεϊνών.

```
(
    load RostSanderDataset.mat
    N=numel(allSeq);
    id = allSeq(7).Header
)

id =
1CSE-ICOMPLEX(SERINEPROTEINASE-INHIBITOR)03-JU

(seq = int2aa(allSeq(7).Sequence)

seq =
KSFPEVVGKTVDQAREYFTLHYPQYNVYFLPEGSPVTLDLRYN
RVRV
FYNPGTNVNVNHVPHVG

)

str = allSeq(7).Structure

str =
CCCHHHCCCCHHHHHHHHHHHHCCCCEEEEEECCCCECCCCCEEEEE
EEEECCCCCECCCCCEEC
```

Όπου:

- **id** είναι το πρωτεύον κλειδί μια δοθείσης ακολουθίας πρωτεΐνης
- **seg** είναι η πρωτεϊνική ακολουθία
- **str** είναι η δομική ανάθεση

Λόγω της τυχαιότητας της συγκεκριμένης προσέγγισης των Rost-Sander, τα αριθμητικά αποτελέσματα μπορεί να είναι ελαφρώς διαφοροποιημένα κάθε φορά που το νευρωνικό δίκτυο εκπαιδεύεται ή κάποια πρόβλεψη προσομοιώνεται. Για την εξασφάλιση της επανα-παραγωγικότητας των αποτελεσμάτων, επαναφέρουμε την γεννήτρια τυχαίων αριθμών σε μία σωσμένη κατάσταση όπως φαίνεται παρακάτω.

```
(
    rng(savedState);
)
```

Όπου: **rng** είναι η γεννήτρια τυχαίων αριθμών (random number generator)

8.1.6 Ορίζοντας το Νευρωνικό Δίκτυο

Για τις συνθήκες του τρέχοντος προβλήματος ορίζουμε για το νευρωνικό δίκτυο τα παρακάτω:

- 1 επίπεδο εισόδου (input layer)
- 1 κρυφό επίπεδο (hidden layer)
- 1 επίπεδο εξόδου (output layer)

Το επίπεδο εισόδου κωδικοποιεί ένα συρόμενο παράθυρο (sliding window) σε κάθε είσοδο της αλληλουχίας του αμινοξέως και μια πρόβλεψη γίνεται στην δομική κατάσταση του κεντρικού υπολείμματος αμινοξέως στο παράθυρο. Κάθε παράθυρο έχει μέγεθος 17, σύμφωνα με την στατιστική συσχέτιση που βρέθηκε μεταξύ της δευτερεύουσας δομής και των οκτώ υπολειμμάτων στις πλευρές του σημείου πρόβλεψης (Holley & Karplus, 1988).

Κάθε θέση παραθύρου κωδικοποιείται με βάση έναν δυαδικό πίνακα μεγέθους 20, έχοντας ένα στοιχείο για κάθε ένα τύπο αμινοξέως. Σε κάθε γκρουπ των 20 εισόδων, το στοιχείο που αντιστοιχεί στον τύπο του αμινοξέως στην δοθείσα είσοδο έχει οριστεί σε 1. Ενώ όλες οι υπόλοιπες εισοδοί ορίζονται σε 0. Έτσι το επίπεδο εισόδου αποτελείται από $R=17 \times 20$ μονάδες εισόδου.

Στον κώδικα που ακολουθεί, καθορίζουμε αρχικά για κάθε ακολουθία πρωτεΐνης όλες τις πιθανές υπ-ακολουθίες (subsequences) που αντιστοιχούν σε ένα συρόμενο παράθυρο μεγέθους W με την δημιουργία ενός Hankel πίνακα. Κάθε i -στήλη αναπαριστά την υπ-ακολουθία ξεκινώντας από την i -οστή θέση της αρχικής ακολουθίας. Στην συνέχεια για κάθε θέση στο παράθυρο δημιουργούμε έναν πίνακα μήκους 20, και θέτουμε το j -οστό στοιχείο σε 1 εάν το υπόλειμμα αμινοξέως στην συγκεκριμένη θέση έχει αριθμητική αναπαράσταση ίση με j .

```
W = 17; % Μέγεθος συρόμενου παραθύρου
for i = 1:N %binarization των εισόδων
    seq = double(allSeq(i).Sequence); % τρέχουσα ακολουθία
    win = hankel(seq(1:W), seq(W:end)); % όλα τα πιθανά συρόμενα
                                        % παράθυρα
    myP = zeros(20*W, size(win,2)); % πίν. Εις.για την τρέχουσα
    ακολουθία
    for k = 1:size(win, 2)
        index = 20*(0:W-1)' + win(:,k); % πίνακας εισόδου για κάθε
    θέση k
        myP(index,k) = 1;
    end
    allSeq(i).P = myP;
end
```

Το επίπεδο εξόδου του νευρωνικού δικτύου αποτελείται από τρεις μονάδες, μια για κάθε δομική κατάσταση, που κωδικοποιούνται με βάση ένα σχήμα τύπου binary. Για να δημιουργήσουμε τον πίνακα στόχο για το νευρωνικό δίκτυο, αρχικά από τα δεδομένα λαμβάνουμε τις δομικές αναθέσεις όλων των πιθανών υπο-ακολουθιών

που αντιστοιχούν με συρόμενο παράθυρο. Στην συνέχεια θεωρούμε την κεντρική θέση σε κάθε παράθυρο και μετατρέπουμε την αντίστοιχη δομική ανάθεση ως ακολούθως:

- Η σπείρα αμυνοξέως (coil) αναπαρίσταται με 1 0 0
- Τα φύλλα αμυνοξέως (sheets) αναπαρίσταται με 0 1 0
- Ο έλικας αμυνοξέως (helix) αναπαρίσταται με 0 0 1

```

cr = ceil(W/2); % θέση κεντρικού υπολείμματος αμυνοξέως

%binarization των στόχων
for i = 1:N
    str = double(allSeq(i).Structure); % τρέχουσα δομική ανάθεση
    win = hankel(str(1:W),str(W:end)); % όλα τα πιθανά συρόμενα
                                     % παράθυρα

    myT = false(3,size(win,2));
    myT(1,:) = win(cr,:) == double('C');
    myT(2,:) = win(cr,:) == double('E');
    myT(3,:) = win(cr,:) == double('H');
    allSeq(i).T = myT;
end

```

Εναλλακτικά αντί για τα δύο τελευταία τμήματα κώδικα θα μπορούσαμε να εκτελέσουμε το ακόλουθο ισοδύναμο:

```

%binarization εισόδων και στόχων
for i = 1:N
    seq = double(allSeq(i).Sequence);
    win = hankel(seq(1:W),seq(W:end)); % concurrent inputs (sliding
    windows)

    %=== binarization of the input matrix
    allSeq(i).P = kron(win,ones(20,1)) ==
    kron(ones(size(win)),(1:20)');

    %=== binarization of the target matrix
    allSeq(i).T = allSeq(i).Structure(repmat((W+1)/2:end-(W-
    1)/2,3,1)) == ...
    repmat('CEH',1,length(allSeq(i).Structure)-W+1);
end

```

Μόλις ορίσουμε τους πίνακες εισόδου και στόχου για κάθε ακολουθία, δημιουργούμε έναν πίνακα εισόδου P και ένα πίνακα στόχου T, κωδικοποιώντας έτσι όλες τις ακολουθίες που διοχετεύονται στο δίκτυο.

```

P = double([allSeq.P]);
T = double([allSeq.T]);

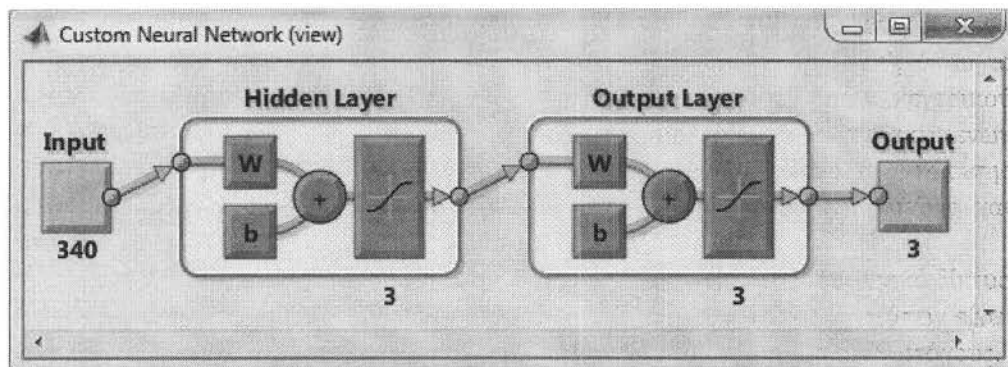
```

8.1.7 Δημιουργώντας το Νευρωνικό Δίκτυο Αναγνώρισης Προτύπου

Το πρόβλημα της προβλέψεως της δευτεροβάθμιας δομής θα μπορούσε να θεωρηθεί ως ένα πρόβλημα αναγνώρισης προτύπων (pattern recognition), όπου το δίκτυο εκπαιδεύεται να αναγνωρίζει την δομική κατάσταση του κεντρικού υπολείμματος αμινοξέως με βάση την παρατηρησιμότητα των παρακείμενων υπολειμμάτων. Έτσι δημιουργούμε ένα τεχνητό νευρωνικό δίκτυο αναγνώρισης προτύπων με τους πίνακες εισόδου και στόχων να έχουν οριστεί παραπάνω και προσδιορίζουμε το κρυφό επίπεδο μεγέθους 3.

Η εντολή `newpr` δημιουργεί μέσω της γραμμής εντολών το νευρωνικό δίκτυο αναγνώρισης προτύπων. Τα τεχνητά νευρωνικά δίκτυα αναγνώρισης προτύπων είναι feed-forward δίκτυα τα οποία επιλύουν προβλήματα με Boolean ή 1-από-N στόχους.

```
hsize = 3;  
net = newpr(P,T,hsize);  
view(net)  
net.layers{1} % κρυμμένο επίπεδο  
net.layers{2} % επίπεδο εξόδου
```



Εικόνα 46. Διάγραμμα τύπου Block για την αναπαράσταση του Νευρωνικού δικτύου από το Matlab

```
ans =  
  
Neural Network Layer  
  
name: 'Hidden Layer'  
dimensions: 3  
distanceFcn: (none)  
distanceParam: (none)  
distances: []  
initFcn: 'initnw'  
netInputFcn: 'netsum'  
netInputParam: (none)
```

```

positions: []
  range: [3x2 double]
  size: 3
topologyFcn: (none)
transferFcn: 'tansig'
transferParam: (none)
userdata: (your custom info)

ans =

Neural Network Layer

  name: 'Output Layer'
  dimensions: 3
  distanceFcn: (none)
  distanceParam: (none)
  distances: []
  initFcn: 'initnw'
  netInputFcn: 'netsum'
  netInputParam: (none)
  positions: []
  range: [3x2 double]
  size: 3
  topologyFcn: (none)
  transferFcn: 'tansig'
  transferParam: (none)
  userdata: (your custom info)

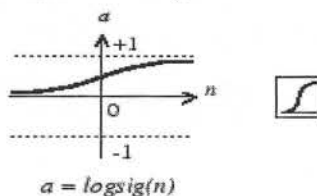
```

8.1.8 Εκπαιδεύοντας το Νευρωνικό Δίκτυο

Το τεχνητό νευρωνικό δίκτυο αναγνώρισης προτύπων χρησιμοποιεί ως προεπιλογή τον αλγόριθμο Κλιμακωτής Συζευκτικής Κλίσης (Scaled Conjugate Gradient algorithm) για την εκπαίδευση (Hestenes, et al., 1952), (Atkinson & Kendell, 1988), (Mordecai, 2003).

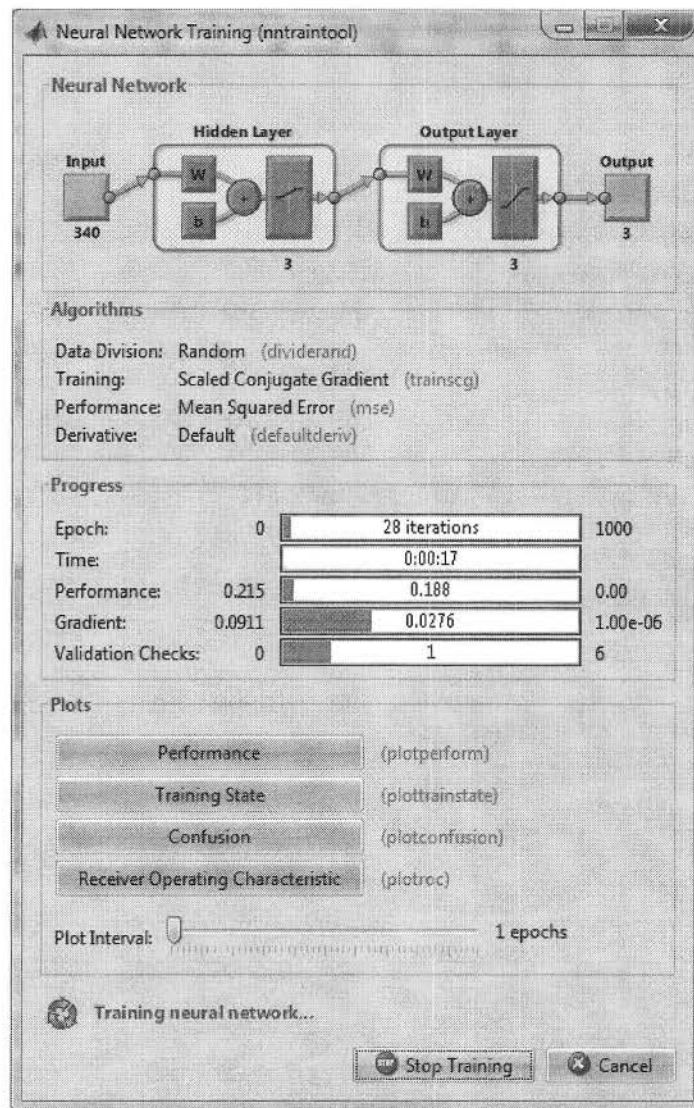
Log-sigmoid transfer function

Graph and Symbol



Εικόνα 47. Λογαριθμο-σιγμοειδής Συνάρτηση Μεταφοράς

Σε κάθε κύκλο εκπαίδευσης, οι ακολουθίες εκπαίδευσης που παρουσιάζονται στο δίκτυο μέσω του συρόμενου παραθύρου, αφορούν ένα κατάλοιπο αμινοξέως την φορά. Κάθε κρυφή μονάδα μετατρέπει τα σήματα που λαμβάνει από επίπεδο εισόδου χρησιμοποιώντας ως συνάρτηση μεταφοράς την $\log\text{sig}$ για να παράγει ένα σήμα εξόδου που είναι κοντά είτε στο 0 είτε στο 1, προσομοιώνοντας έτσι την πυροδότηση ενός νευρώνα. (Holley & Karplus, 1988). Τα βάρη προσαρμόζονται έτσι ώστε το σφάλμα μεταξύ παρατηρούμενης εξόδου από κάθε μονάδα και την επιθυμητή έξοδο που βρίσκεται τον πίνακα στόχων, να ελαχιστοποιείται. Κατά την διάρκεια της εκπαίδευσης, το GUI του Matlab ανοίγει και εμφανίζει την πρόοδο.



Εικόνα 48. Στιγμιότυπο εκπαίδευσης Νευρωνικού Δικτύου Αναγνώρισης Προτύπων

Ένα από προβλήματα που μπορούν να συμβούν κατά την διάρκεια της εκπαίδευσης των νευρωνικών δικτύων είναι το data overfitting, όπου το νευρωνικό δίκτυο τείνει να απομνημονεύει τα εκπαιδευόμενα παραδείγματα χωρίς να επιλύει γενικά προβλήματα και συνεπώς γίνεται αρκετά ειδικό στο συγκεκριμένο τύπο προβλήματος. Η προεπιλεγμένη μέθοδος για την βελτίωση της γενίκευσης ονομάζεται πρόωρη διακοπή (early stopping) και βασίζεται στην κατανομή των διαθέσιμων δεδομένων εκπαίδευσης σε τρεις υποκατηγορίες:

- (a) Το σύνολο εκπαίδευσης (training set), που χρησιμοποιείται για τον υπολογισμό της κλίσης και την ανανέωση των τιμών των βαρών
- (b) Το σύνολο επικύρωσης (validation set), όπου γίνεται καταγραφή των λαθών κατά την διάρκεια της εκπαίδευσης γιατί τείνει να μεγαλώνει όταν έχουμε overfitting δεδομένων.
- (c) Το σύνολο δοκιμής (test set), όπου το σφάλμα μπορεί να χρησιμοποιηθεί για την αξιολόγηση της ποιότητας της κατανομής δεδομένων.

Όταν χρησιμοποιείται η συνάρτηση εκπαίδευσης, εκ προεπιλογής τα δεδομένα χωρίζονται έτσι ώστε το 60% εξ' αυτών να ανατεθούν στο σύνολο εκπαίδευσης, το 20% στο σύνολο επικύρωσης και το υπόλοιπο 20% στο σύνολο δοκιμής. Κάτι το οποίο μπορεί να παραμετροποιηθεί με την εντολή `net.divideFcn` (default dividerand). Η δομική σύνθεση (structural composition) των υπολειμμάτων αμινοξέων σε αυτά τα τρία υποσύνολα μπορούν να συγκριθούν όπως φαίνεται στα ακόλουθα σχήματα.

(Εδώ φαίνεται ο κώδικας για την παραγωγή των σχημάτων)

```
[i,j] = find(T(:,tr.trainInd));
Ctrain = sum(i == 1)/length(i);
Etrain = sum(i == 2)/length(i);
Htrain = sum(i == 3)/length(i);

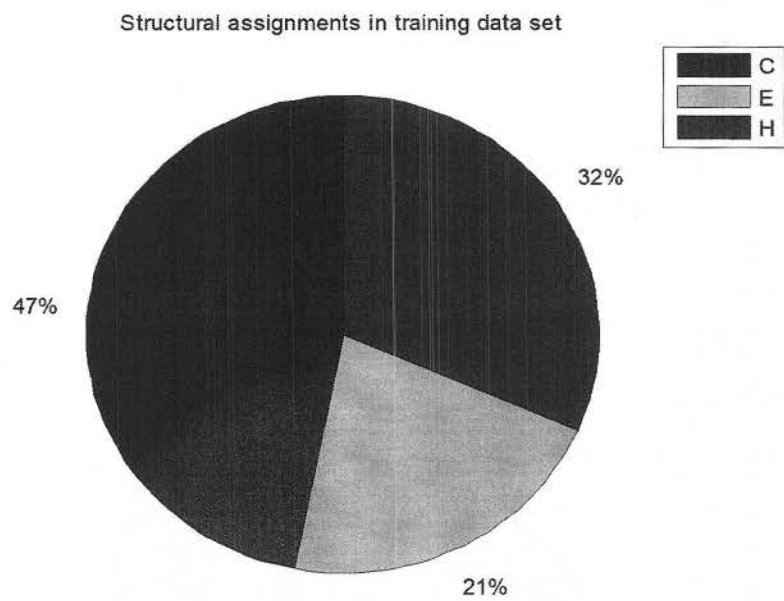
[i,j] = find(T(:,tr.valInd));
Cval = sum(i == 1)/length(i);
Eval = sum(i == 2)/length(i);
Hval = sum(i == 3)/length(i);

[i,j] = find(T(:,tr.testInd));
Ctest = sum(i == 1)/length(i);
Etest = sum(i == 2)/length(i);
Htest = sum(i == 3)/length(i);

figure(); pie([Ctrain; Etrain; Htrain]);
title('Structural assignments in training data set');
legend('C', 'E', 'H')
```

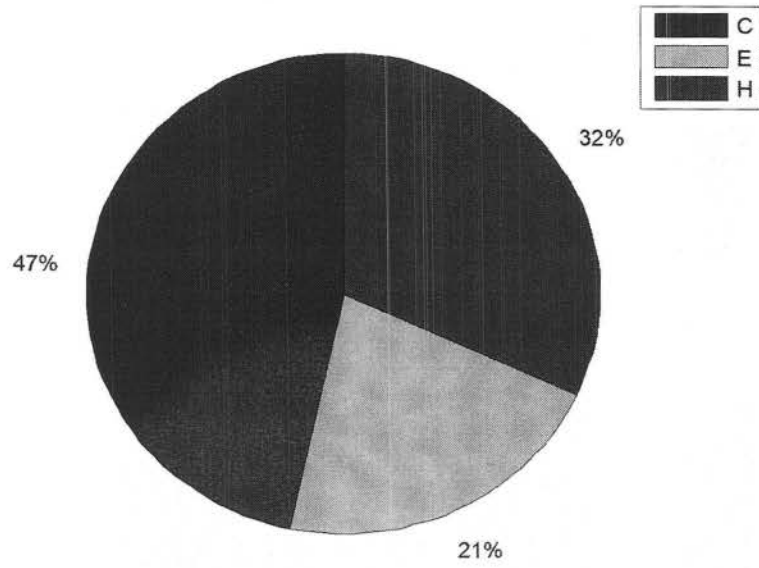
```
figure(); pie([Cval; Eval; Hval]);  
title('Structural assignments in validation data set');  
legend('C', 'E', 'H')
```

```
figure(); pie([Ctest; Etest; Htest]);  
title('Structural assignments in testing data set ');  
legend('C', 'E', 'H')
```



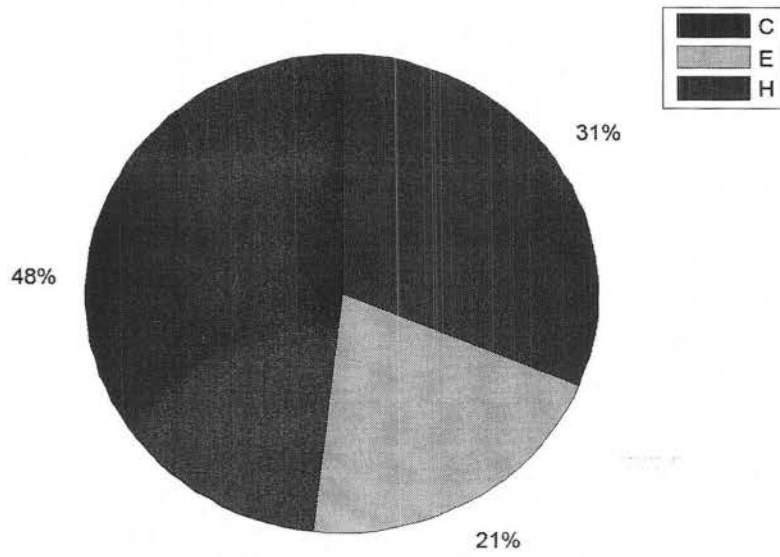
Εικόνα 49. Διαρθρωτικές αναθέσεις σε δεδομένα εκπαίδευσης

Structural assignments in validation data set



Εικόνα 50. Διαρθρωτικές αναθέσεις σε δεδομένα επαλήθευσης

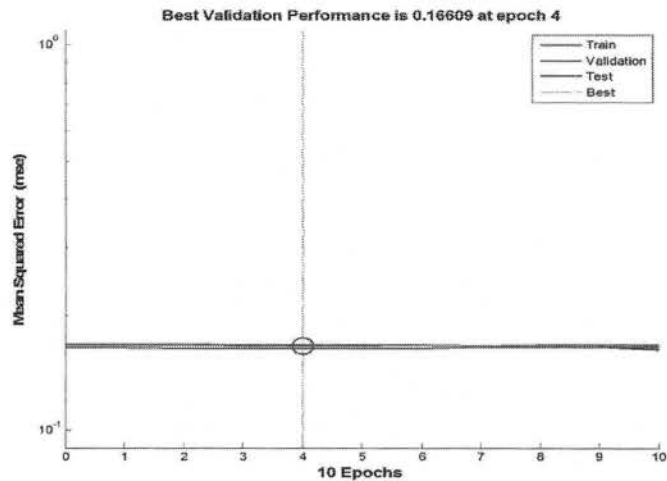
Structural assignments in testing data set



Εικόνα 51. Διαρθρωτικές αναθέσεις σε δεδομένα testing

Επίσης με την συνάρτηση `plotperform` μπορούμε να δούμε τις τάσεις απόδοσης της εκπαίδευσης, της επικύρωσης και της δοκιμής.

```
figure();plotperform(tr)
```



Εικόνα 52. Απόδοση εκπαίδευσης, ελέγχου και δοκιμής

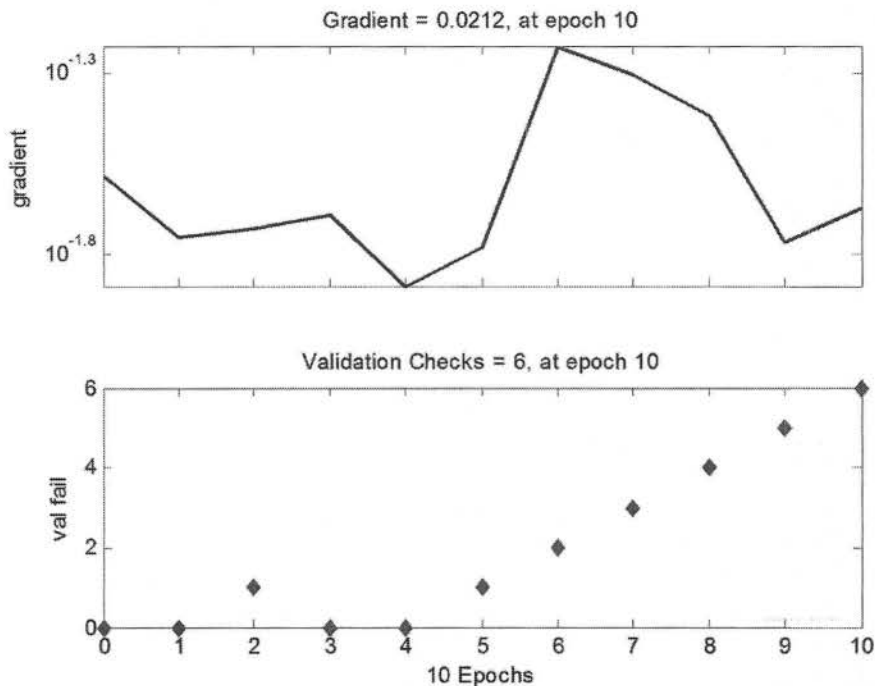
Η εκπαιδευτική διαδικασία σταματά όταν μια από τις συνθήκες εκπληρώνεται (βλ. `net.trainParam`). Για παράδειγμα, στο πρόβλημα μας, η εκπαιδευτική διαδικασία σταματά όταν το σφάλμα επικύρωσης (`validation error`) μεγαλώσει για συγκεκριμένο αριθμό επαναλήψεων (6) ή ο `maximum` αριθμός των επιτρεπόμενων επαναλήψεων φτάσει το 1000.

```
net.trainParam
%Εκτύπωση επικύρωσης και κλίσης
figure();plottrainstate(tr)
```

```
ans =

Function Parameters for 'trainseg'

Show Training Window Feedback      showWindow: true
Show Command Line Feedback        showCommandLine: false
Command Line Frequency              show: 25
Maximum Epochs                     : 1000
Maximum Training Time               : Inf
Performance Goal                    goal: 0
Minimum Gradient                    min_grad: 1e-06
Maximum Validation Checks            max_fail: 6
Sigma                                sigma: 5e-05
Lambda                              lambda: 5e-07
```



Εικόνα 53. Απόδοση Επικύρωσης δεδομένων

8.2 Αναλύοντας την απόκριση του Νευρωνικού Δικτύου

Για να αναλύσουμε την απόκριση του δικτύου θα εξετάσουμε τον πίνακα συγχύσεως (confusion matrix) συγκρίνοντας τις εξόδους του νευρωνικού και τα επιθυμητές εξόδους – στόχους.

```

(
  O = sim(net,P);
  figure();plotconfusion(T,O);
)

```

Τα διαγώνια κελιά δείχνουν τον αριθμό των θέσεων των υπολειμμάτων αμινοξέως τα οποία κατηγοριοποιήθηκαν σωστά για κάθε δομική τάξη. Τα εκτός διαγωνίου κελιά δείχνουν τον αριθμό των θέσεων των λανθασμένων κατηγοριοποιήσεων (π.χ. οι θέσεις για τους έλικες να προβλέφθηκαν ως θέσεις για τις σπείρες). Το μπλε κελί εμφανίζει το συνολικό ποσοστό των σωστών προβλέψεων (με πράσινο χρώμα) και τον τελικό αριθμό των λανθασμένων προβλέψεων.

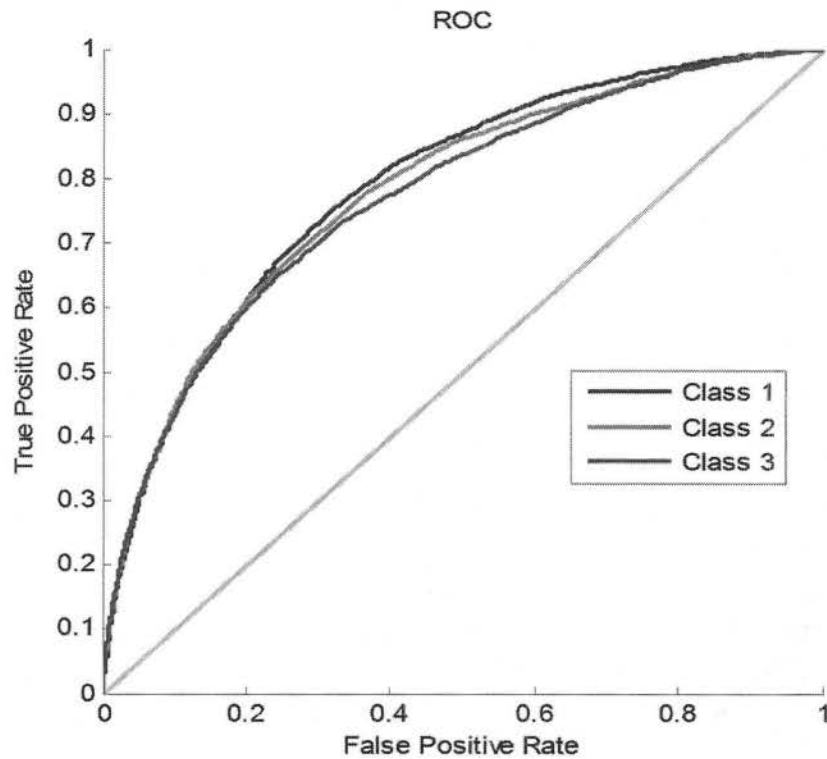
Confusion Matrix

Output Class	1	5700 36.8%	1290 8.3%	1684 10.9%	65.7% 34.3%
	2	573 3.7%	1309 8.5%	477 3.1%	53.5% 44.5%
	3	1017 6.6%	702 4.5%	2719 17.6%	61.3% 38.7%
		78.2% 21.8%	39.7% 60.3%	55.7% 44.3%	62.9% 37.1%
		1	2	3	
		Target Class			

Εικόνα 54. Πίνακας Συγγήσεως Νευρωνικού δικτύου

Μπορούμε επίσης να εξετάσουμε την καμπύλη του Χαρακτηριστικού Λειτουργίας Δέκτη (Receiver Operating Characteristic - ROC), που είναι ένα γράφημα του πραγματικού θετικού συντελεστή (ευαισθησία) σε σχέση με τον αρνητικό συντελεστή.

```
( figure(); plotroc(T,O); )
```



Εικόνα 55. Γράφημα Χαρακτηριστικού λειτουργίας δέκτη

8.3 Βελτιστοποίηση του νευρωνικού δικτύου για πιο ακριβή αποτελέσματα

Το νευρωνικό δίκτυο το οποίο πραγματευόμαστε είναι ένα σχετικά απλό νευρωνικό δίκτυο. Για να πετύχουμε κάποιες βελτιώσεις στην ακρίβεια πρόβλεψης θα μπορούσαμε να δοκιμάσουμε ένα εκ των παρακάτω:

- Αύξηση του αριθμού των διανυσμάτων εκπαίδευσης. Η αύξηση του αριθμού των ακολουθιών που αφορούν την εκπαίδευση του δικτύου απαιτεί μια μεγαλύτερη και πιο ακριβής βάση δεδομένων των πρωτεϊνικών δομών, με πιο ορθές κατανομές σπειρών, ελίκων και φύλλων.
- Αύξηση του αριθμού των εισόδων. Αυξάνοντας το μέγεθος του παραθύρου ή προσθέτοντας πιο σχετικές πληροφορίες όπως βιοχημικές ιδιότητες των αμινοξέων θα ήταν προς την σωστή κατεύθυνση.
- Χρησιμοποίηση διαφορετικού αλγόριθμου εκπαίδευσης. Υπάρχουν αρκετοί αλγόριθμοι οι οποίοι διαφέρουν στην μνήμη και στις απαιτήσεις ταχύτητας. Για παράδειγμα, η Κλίση Κλιμακωτής

Σύζευξης (Scaled Conjugate Gradient) είναι σχετικά αργή αλλά αποτελεσματική από πλευράς μνήμης, ενώ η τεχνική Levenberg-Marquardt είναι γρηγορότερη αλλά πιο απαιτητική όσον αφορά την μνήμη.

- Αύξηση του αριθμού των νευρώνων του κρυμμένου επιπέδου. Με τη προσθήκη περισσότερων μονάδων στο κρυφό επίπεδο, έχουμε ένα γενικά πιο πολύπλοκο δίκτυο με δυνατότητες για καλύτερες επιδόσεις αλλά θα πρέπει να είμαστε αρκετά προσεκτικοί να μην έχουμε overfitting δεδομένων.

Μπορούμε να καθορίσουμε περισσότερα κρυφά επίπεδα ή να αυξήσουμε το μέγεθος των κρυφών επιπέδων όταν το τεχνητό νευρωνικό δίκτυο αναγνώρισης προτύπων δημιουργηθεί όπως φαίνεται παρακάτω:

```

hsize = [3 4 2];
net3 = newpr(P,T,hsize);

hsize = 20;
net20 = newpr(P,T,hsize);

```

Μπορούμε επίσης να εκχωρήσουμε στο δίκτυο αρχικά βάρη από τυχαίες τιμές στο $[-0.1,0.1]$ όπως προτείνεται από την ερευνητική εργασία των Holley και Karplus (Holley & Karplus, 1988). Καθορίζουμε τις παραμέτρους net20.IW και net20.LW ως ακολούθως.

```

net20.IW{1} = -.1 + (.1 + .1) .* rand(size(net20.IW{1}));
net20.LW{2} = -.1 + (.1 + .1) .* rand(size(net20.LW{2}));

```

Σε γενικές γραμμές, τα μεγαλύτερα δίκτυα (με 20 ή περισσότερες κρυφές μονάδες) πετυχαίνουν μεγαλύτερη ακρίβεια στο σύνολο εκπαίδευσης της πρωτεύουσας, αλλά χειρότερη ακρίβεια στην πρόβλεψη. Επειδή ο αριθμός των 20 κρυμμένων μονάδων να περιλαμβάνουν περίπου 7000 βάρη, το δίκτυο είναι σε θέση να εφαρμόσει το σύνολο εκπαίδευσης αλλά χάνει την ικανότητα γενίκευσης. Ο συμβιβασμός μεταξύ της εντατικής εκπαίδευσης και της ακρίβειας πρόβλεψης είναι ένας από τους βασικούς περιορισμούς των νευρωνικών δικτύων.

```

net20 = train(net20,P,T);

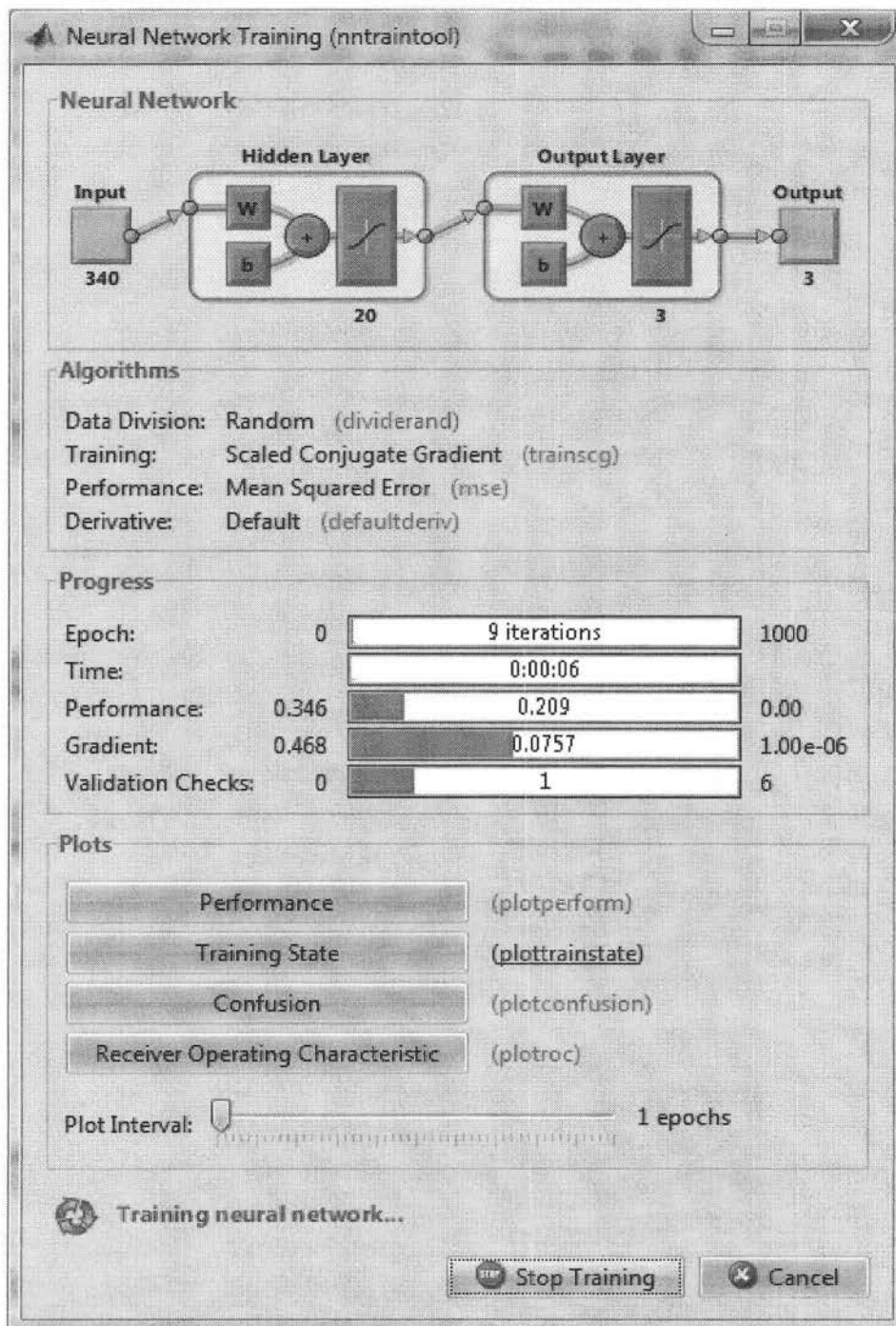
O20 = sim(net20,P);
numWeightsAndBiases = length(getx(net20))

```

```

numWeightsAndBiases =
6883

```



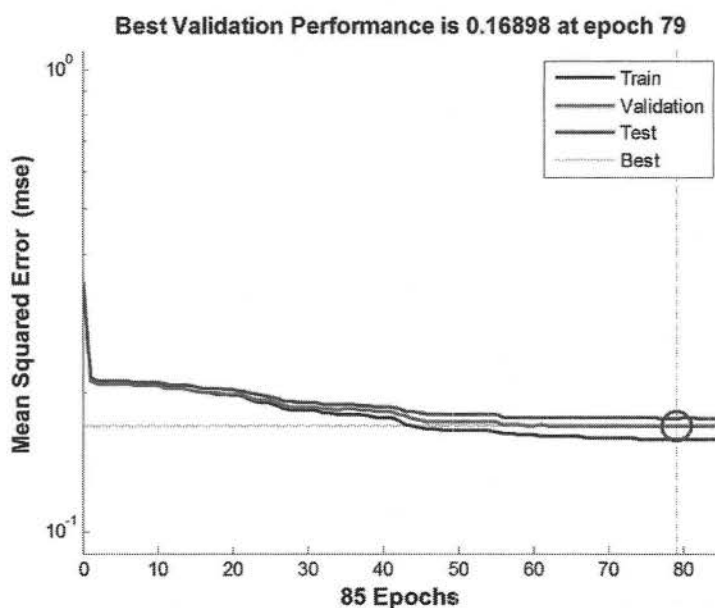
Εικόνα 56. Εκ νέου εκπαίδευση νευρωνικού δικτύου

Μπορούμε να εμφανίσουμε τους πίνακες σύγχυσης (confusion matrices) για την εκπαίδευση, επικύρωση και την δοκιμή των ακολουθιών.



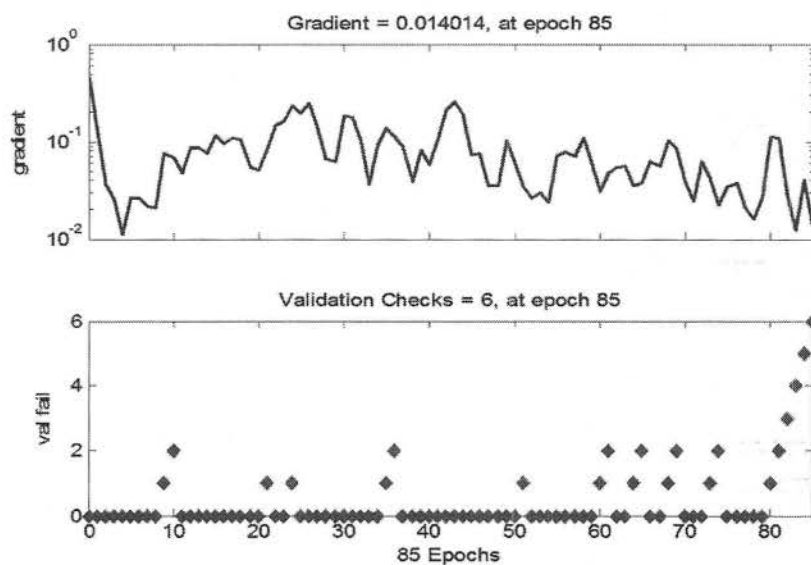
Εικόνα 57. Πίνακες Συγχύσεως για την εκπαίδευση, επικύρωση και την δοκιμή των ακολουθιών

Στο παρακάτω σχήμα βλέπουμε την Επίδοση

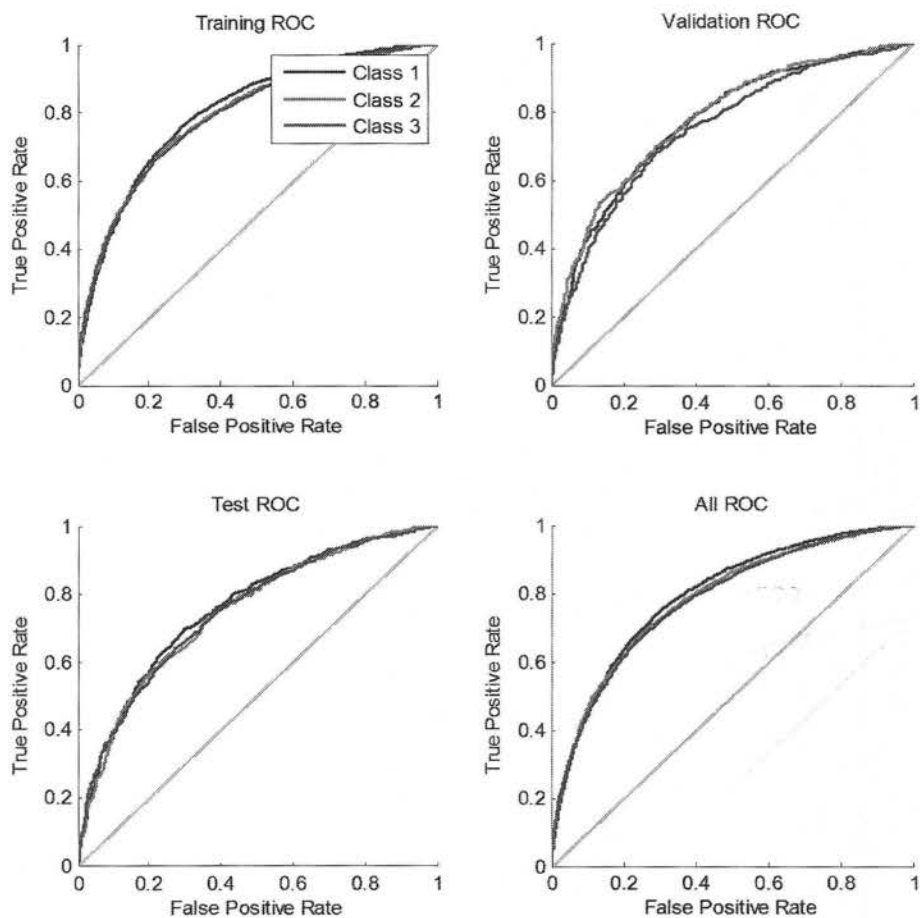


Εικόνα 58. Επίδοση Νευρωνικού δικτύου

Στα δύο σχήματα που ακολουθούν βλέπουμε τα καινούργια γραφήματα της Κατάστασης εκπαίδευσης και του Χαρακτηριστικού Λειτουργικού Δέκτη (Receiver Operating Characteristic) αντίστοιχα.



Εικόνα 59. Νέο γράφημα κατάστασης επαλήθευσης



Εικόνα 60. Γραφήματα Χαρακτηριστικού Λειτουργικού Δέκτη για τις φάσεις της εκπαίδευσης, της επικύρωσης του testing

8.4 Αξιολόγηση της Απόδοσης του Νευρωνικού Δικτύου

Μπορούμε να αξιολογήσουμε τις προβλέψεις των δομών με ακρίβεια με βάση τον υπολογισμό των πινάκων πρόβλεψης ποιότητας που πρότειναν οι Kabsch και Sander (Kabsch & Sander, 1983). Με την προτεινόμενη μέθοδο μπορούμε να παρατηρήσουμε πόσο καλά μια συγκεκριμένη κατάσταση έχει προβλεφθεί και που έχουμε διαφορές. Ορίζουμε τον δείκτη $pcObs(S)$ ως τον αριθμό των υπολειμμάτων αμινοξέως που έχουν προβλεφθεί σωστά για την κατάσταση S ($S=\{C,E,H\}$), διαιρούμενο τον αριθμό των υπολειμμάτων αμινοξέων που παρατηρήθηκαν στην κατάσταση S . Ομοίως, ορίζουμε τον δείκτη $pcPred(S)$.

```

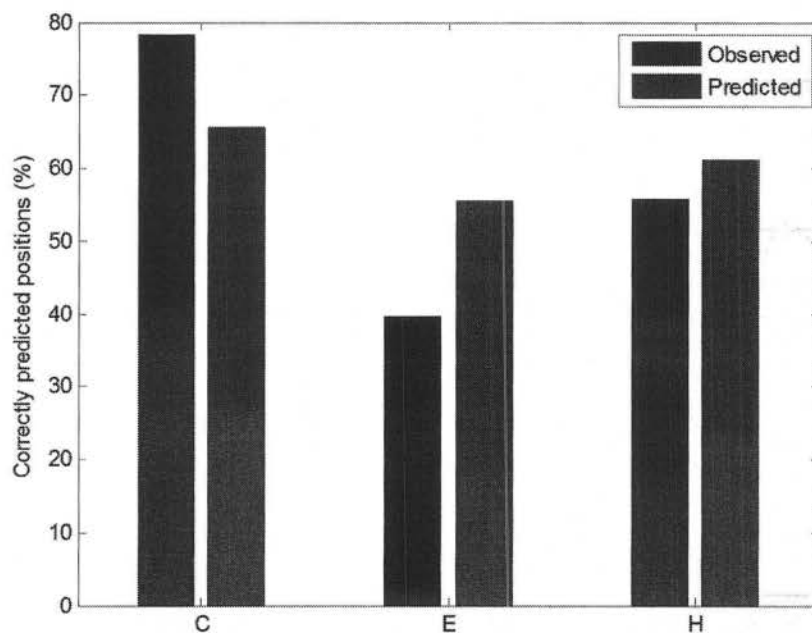
[i,j] = find(compet(0));
[u,v] = find(T);

%Υπολογισμός των σωστών προβλέψεων δεδομένης παρατηρούμενης κατάστασης
pcObs(1) = sum(i == 1 & u == 1)/sum (u == 1); % state C
pcObs(2) = sum(i == 2 & u == 2)/sum (u == 2); % state E
pcObs(3) = sum(i == 3 & u == 3)/sum (u == 3); % state H

% Υπολογισμός των σωστών προβλέψεων δεδομένης προβλεφθείσας κατάστασης
pcPred(1) = sum(i == 1 & u == 1)/sum (i == 1); % state C
pcPred(2) = sum(i == 2 & u == 2)/sum (i == 2); % state E
pcPred(3) = sum(i == 3 & u == 3)/sum (i == 3); % state H

%Σύγκριση ποιοτικών πινάκων προβλέψεων
figure(); bar([pcObs' pcPred'] * 100);
ylabel('Correctly predicted positions (%)');
set(gca, 'XTickLabel', {'C'; 'E'; 'H'});
legend({'Observed', 'Predicted'});

```



Εικόνα 61. Τελικά αποτελέσματα προβλέψεων νευρωνικού δικτύου

Οι συγκεκριμένοι δείκτες ποιότητας είναι χρήσιμοι για την ερμηνεία της ακρίβειας πρόβλεψης. Στην πραγματικότητα, σε περιπτώσεις όπου η τεχνική πρόβλεψης τείνει να υπερπροβλέπει/υποπροβλέπει για μια δοσμένη κατάσταση, μια

υψηλή/χαμηλή ακρίβεια πρόβλεψης θα μπορούσε να είναι φαινομενική και δεν παρέχεται ένα μέτρο ποιότητας για την τεχνική καθ' αυτή.

8.5 Συμπεράσματα

Η μέθοδος που παρουσιάστηκε, προβλέπει τη δομική κατάσταση ενός δοσμένου υπολείμματος πρωτεΐνης με βάση την δομική κατάσταση των γειτνιαζουσών υπολειμμάτων. Ωστόσο, υπάρχουν ακόμη περιορισμοί κατά την πρόβλεψη του περιεχομένου δομικών στοιχείων μιας πρωτεΐνης, όπως το ελάχιστο μήκος κάθε δομικού στοιχείου. Συγκεκριμένα, ένας έλικας εκχωρείται σε κάθε γκρουπ τεσσάρων ή περισσότερων συνεχόμενων υπολειμμάτων, και ένα φύλο ορίζεται σε κάθε γκρουπ δύο ή περισσότερων συνεχόμενων υπολειμμάτων. Για διαχειριστούμε αυτό το είδος της πληροφορίας, δημιουργούμε ένα επιπλέον δίκτυο έτσι ώστε το πρώτο δίκτυο να προβλέπει την δομική κατάσταση από την αλληλουχία του αμινοξέως και το δεύτερο να προβλέπει τα δομικά στοιχεία από την δομική κατάσταση.

9. Βιβλιογραφία

- Aihara, K., Takabe, T. & Toyoda, M., 1990. Chaotic Neural Networks. *Physics Letters A*, Τόμος 144, pp. 333-340.
- Aizenberg, I., Aizenberg & N. Hiltnerb, J., 2001. Cellular neural networks and computational intelligence in medical image processing. *Image and Vision Computing*, 19(4), pp. 177-183.
- Anderson, P., 1995. Mutagenesis. *Methods Cell Biol*, pp. 31-58.
- Anthony, D., Hines, E., Barham, J. & Taylor, D., 1990. A comparison of image compression by Neural Networks and Principal Component Analysis. *International Joint Conference on Neural Networks*, Τόμος 1, pp. 339-344.
- Argyarakis, 2001. *Neural networks and applications*. Patras: s.n.
- Atkinson & Kendell, 1988. *An introduction to numerical analysis*. 2nd επιμ. s.l.:John Wiley and Sons.
- Baer, Connors & Paradiso, 2006. *Neuroscience: Exploring the brain*. 3rd επιμ. s.l.:Lippincott Williams & Wilkins.
- Bailey, C. & Kandel, E., 1993. Structural changes accompanying memory storage. *Annu. Rev. Physiol.*, Τόμος 55, pp. 397-426.
- Banks, J. και συν., n.d. On Devaney's definition of chaos, s.l.: Dep. of Mathematics, La Trobe University, Melbourne Australia.
- Baxt, W., 1995. Application of Artificial Neural Networks to Clinical Medicine. *Lancet*, Τόμος 346, pp. 1135-1138.
- Behrman, E., Niemel, J., Steck, J. & Skinner, S., 1996. A quantum dot neural network. *Proceedings of the 4th Workshop on Physics of Computation in Boston*, pp. 22-24.
- Behrman, E., Steck, J. & Skinner, S., 1999. A spatial quantum neural computer. *Proceedings of the International Joint Conference on Neural Networks*.
- Bishop, C., 1995. *Neural networks for pattern recognition*. Oxford: Clarendon Press.
- Bower, B., 1988. Chaotic Connections. *Science News*, Τόμος 133, pp. 58-59.
- Brooks, M., 1999. *Quantum computing and communications*. Heidelberg: Springer - Verlag.
- Bruhn, J., Ropcke, H. & Hoefft, A., 2000. Approximate entropy as an electroencephalographic measure of anesthetic drug effect during desflurane anesthesia. *Anesthesiology*, 92(3), pp. 715-726.

- Bryson, A. & Ho, Y., 1969. Applied Optimal Control. New York: Blaisdell Publishing.
- Burioka, N. και συν., 2005. Approximate entropy of the electroencephalogram in healthy awake subjects and absence epilepsy patients. *Clinical EEG and Neuroscience*, 36(3), pp. 188-193.
- Chen, C. και συν., 2006. Predicting protein structural class with pseudo - amino acid composition and support vector machine fusion network. *Anal. Biochem.*, 357(1), pp. 116-121.
- Chen, W., Wang, Z., Xie, H. & Yu, W., 2007. Characterization of surface EMG signal based on fuzzy entropy. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(2), pp. 266-272.
- Chen, W., Zhuang, J., Yu, W. & Wang, Z., 2009. Measuring Complexity using FuzzyEn, ApEn and SampEn. *Medical Engineering & Physics*, Τόμος 31, pp. 61-68.
- Choi, M. & Huberman, B., 1983. Dynamic behavior of nonlinear networks. *Physical Review A*, Τόμος 28, pp. 1204-1206.
- Chrisley, R., 1997. Learning in Non-superpositional Quantum Neurocomputers. *Brain, Mind and Physics*, pp. 126-139.
- Constantini, S., Colonna, G. & Facchiano, A., 2006. Amino acid propensities for secondary structures are influenced by the protein structural class. *Biochem Biophys. Res. Commun.*, 342(2), pp. 441-451.
- Cotrell, G., Munro, P. & Zipser, D., 1985. Learning internal representation from gray-scale images: An example of extensional programming. *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*.
- Cutting, D., 1999. Would quantum neural networks be subject to the decidability constraints of the Church - Turing thesis?. *Neural Network World*, Τόμος N.1-2, pp. 163-168.
- Däschlein, R., Waschulzik, T. & Brauer, W., 1994. Segmentation of Computer tomographies with Neural Networks Based on Local Features. *International Conference on Neural Networks and Expert Systems in Medicine and Healthcare*, pp. 240-247.
- Deutsch, D., 1985. Quantum theory, the Church - Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London*, Τόμος A400, pp. 97-117.
- Devaney, R., 1989. *An Introduction to Chaotic Dynamical Systems*. s.l.:Addison - Wesley.
- Dirac, P., 1958. *The principles of Quantum Mechanics*. Oxford: Oxford Clarendon Press.
- Domany, E., van Hemmen, J. & Schulten, K., 1992. *Models of neural networks*. Heidelberg: Springer - Verlag.

- Eckmann, J. & Ruelle, D., 1985. Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.*, Τόμος 57, pp. 617-656.
- Eftaxias, K., 2010. Footprints of non - extensive Tsallis statistics, self - affinity and universality in the preparation of the L'Aquila earthquake hidden in a pre - seismic EM emission. *Phys. A*, Τόμος 389, pp. 133-140.
- Eftaxias, K. και συν., 2009. Unfolding the procedure of characterizing recorded ULF, KHz and MHz electromagnetic anomalies prior to the L'Aquila earthquake as pre-seismic ones. part 1, *Nat. Hazard Earth Sys.*, Τόμος 9, pp. 1953-1971.
- Eftaxias, K. και συν., 2006. Unified approach to catastrophic events : from the normal state to geological or biological shock in terms of spectral fractal and nonlinear analysis. *Nat. Hazards Earth Syst. Sci.*, Τόμος 6, pp. 205-228.
- Eisenberg, J., Freeman, W. & Burke, B., 1989. Hardware Architecture of a Neural Network Model Simulating Pattern Recognition. *Olfactory Bulb Neural Networks*, Τόμος 2, pp. 315-325.
- Everett, H., 1957. Relative state formulation of quantum mechanics. *Review of Modern Physics*, Τόμος 29, pp. 454-462.
- Ezhov, A., Kalambet, Y. & Knizhnikova, L., 1990. *Neural networks: general properties and particular applications*. Manchester: Neural Networks: Theory and Architectures. Manchester University Press.
- Ezhov, A. & Ventura, D., 2000. *Quantum Neural Networks. Future Directions for Intelligent Systems and Information Science*, Physica-Verlag.
- Ezhov, A. & Vvedensky, V., 1996. Object generation with neural networks (when spurious memories are useful). *Neural Networks*, Τόμος 9, pp. 1491-1495.
- Feynman, R., 1986. Quantum mechanical computers. *Foundations of Physics*, Τόμος 16, pp. 507-531.
- Folland, R. και συν., 2004. Comparison of neural network predictors in the classification of tracheal - bronchial breath sounds by respiratory auscultation. *Artificial Intelligence in Medicine*, 31(3), pp. 211-220.
- Fotopoulos, A. και συν., 2012. Environmental monitoring of soil radon in a very tectonic area in South West Greece. *Piraeus, eRA-7 Int. Sci. Conf.*.
- Frazer - Smith, A. και συν., 1990. Low - frequency magnetic field measurements near the epicenter of the 7.1 Loma Prieta earthquake. *Geophy. Res. Lett.*, Τόμος 17, pp. 1465-1467.

- Freeman, W. & Yao, Y., 1990. Model of Biological Pattern Recognition with Spatially Chaotic Dynamics Neural Networks. Τόμος 3, pp. 153-170.
- Fukai, T. & Shiino, M., 1990. Asymmetric Neural Networks Incorporating the Dale Hypothesis and Noise-Driven Chaos. *Physical Review Letters*, Τόμος 64, pp. 1465-1468.
- Gallant, A. & White, H., 1988. There Exists a Neural Network That Does Not Make Avoidable Mistakes. *Proceedings of the Second Annual IEEE Conference on Neural Networks*, Τόμος 1, pp. 657-664.
- Gazzaniga, M., 1996. *Cognitive Neuroscience and the future of psychiatry*. San Francisco, American Association of Directors of Psychiatric Residency.
- Gerschenfeld, N. & Chuang, I., 1996. Bulk Spin Resonance Quantum Computation. *Science*, Τόμος 257, p. 350.
- Gross, D., n.d. *The importance of Chaos Theory in the Development of Artificial Neural Systems*, s.l.: s.n.
- Grouven, U., Beger, F., Schultz, B. & Schultz, A., 2005. Correlation of Narcotrend Index, entropy measures, and spectral parameters with calculated propofol effect - site concentrations during induction of propofol-remifentanyl anaesthesia. *Journal of Clinical Monitoring and Computing*, 18(4), pp. 231-240.
- Grushka, J., 1999. *Quantum Computing*. s.l.: McGraw - Hill.
- Guevara, M., Glass, L., Mackey, M. & Shrier, A., 1983. Chaos in Neurobiology. *IEEE Transactions on Systems, Man, and Cybernetics*, Τόμος SMC-13, pp. 790-798.
- Guzzo, A., 1965. Influence of Amino-Acid Sequence on Protein Structure. *Biophys. J.*, 5(6), pp. 809-822.
- Hall, L. και συν., 1992. A comparizon of Neural Network and Fuzzy Clustering Techniques in Segmenting MRI of the brain. *IEEE Transactions on Neural Networks*, 3(5), pp. 672-682.
- Hameroff, S. & Rasmussen, S., 1990. Hameroff, S. and Rasmussen, S. (1990) Microtubule Automata: Sub-Neural Information Processing in Biological Neural Networks. In: *Theoretical Microtubule Automata: Sub-Neural Information Processing in Biological Neural Networks*. Theoretical Aspects of Neurocomputing, World Scientific, pp. 3-12.
- Hartman, E., Keeler, J. & Kowalski, J., 1990. Layered neural networks with Gaussian Hidden units as universal approximations. *Neural Computations*, 2(2), pp. 210-215.
- Haykin, S., 1999. *Neural Networks: A Comprehensive Foundation*. 2nd εκμ. Upper Saddle Point: Prentice Hall.

- Hebb, D., 1949. The organisation of behavior. s.l.:s.n.
- Hecht-Nielsen, R., 1989. Theory of Backpropagation Neural Networks. s.l., Proceedings of the International Joint Conference on Neural Networks.
- Hertz, A. & Hopfield, J., 1995. Earthquake cycles and neural reverberations: collective oscillations in systems with pulse - coupled threshold elements. Phys. Rev. Lett., Τόμος 75, pp. 1222-1225.
- Hestenes, M. & Stiefel, E., 1952. Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards, 49(6).
- Holley, L. & Karplus, M., 1988. Protein secondary structure prediction with a neural network. Proc. Natl. Acad. Sci. USA, Τόμος 86, pp. 152-156.
- Hopfield, J., 1982. Neural Networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci., Τόμος 79, p. 2554.
- Hopfield, J., 1984. Neurons with graded response have collective computational properties like those of two state neurons. Proc. Natl. Acad. Sci., Τόμος 81, p. 3088.
- Hornik, K., Stinchcombe, G. & White, H., 1989. Multilayer Feedforward Networks are Universal Approximators. Neural Networks, Τόμος 2, pp. 359-366.
- Ikeguchi, T., Itoh, S., Utsunomiya, T. & Aihara, K., 1990. A dimensional analysis on chaotic neural networks. Electronics and Communications in Japan, 73(3), pp. 89-97.
- Jervis, B., Yu, S., Saatchi, M. & Allen, E., 1994. The sensitivity of Multilayer Perceptrons for Differentiating between CNV Waveforms of Schizophrenic, Parkinson's Disease and Huntington's Disease Subjects. Iteachor, International Conference on Neural Networks and Expert Systems in Medicine and Healthcare, pp. 275-282.
- Kabsch, W. & Sander, C., 1983. How good are predictions of protein secondary structure?. FEBS Letters, 155(2), pp. 179-182.
- Kalat, J. W., 1995. Biological psychology. 5th επιμ. s.l.:Brooks/Cole Pub. Co..
- Kandel, 1999. Biology and the future of psychoanalysis: a new intellectual framework for psychiatry revisited.. Am J Psychiatry, 156(4), pp. 505-24..
- Kaplan, H., 1995. Comprehensive Textbook of Psychiatry. Baltimore: MD: Williams & Wilkins..
- Karakas, M., Duru, H. & Tipi, V., 1994. A system for an automated classification of blood cells from microscope images. International Conference on Neural Networks and Expert Systems in Medicine and Healthcare, pp. 248-256.
- Kettemann & Ransom, 2005. The Concept of Neuroglia: A Historical Perspective. Oxford: Oxford University Press.

- Knill, E., Laflamme, R., Martinez, R. & Tseng, C., 1999. A Cat-State Benchmark on a Seven Bit Quantum Computer. Los Alamos pre-print archive.
- Kohonen, T., 1987. Adaptive, associative and self-organizing functions in neural computins. *Applied Optics*, Τόμος 26, p. 4910.
- Koulouras, G. και συν., 2005. Embedded Compact Flash. *Circ. And Devices Magazine, IEEE*, 21(4).
- Lane, S., Flax, M., Handelman, D. & Gelfand, J., 1991. Multilayer perceptrons with B-spline receptive field functions. *Advances in Neural Information Processing Systems*, pp. 684-692.
- Liappas, 1999. *Drugs, Addiction, Clinical Problems and Treatment*. s.l.:Patakis Publications.
- Lymperakis, S., 1997. *Brain and Psychology: Introduction to Neuropsychology*. 2nd επιμ. s.l.:Ellinika Grammata.
- Lyngso, R. & Pedersen, C., 2000. RNA pseudoknot prediction in energy - based models. *Journal Comput. Biol.*, 7(3-4), pp. 409-427.
- Margaris, A., 2003. *Neural Networks and Chaos: Study and simulation of chaotic attractors through the use of neural networks*. Thessaloniki: Ph.D. Thesis, Univ. of Mac..
- Mathworks, 2012. *Creating Graphical User Interfaces*. s.l.:s.n.
- McCulloch, W. & Pitts, W., 1947. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, Τόμος 9, p. 127.
- Menneer, T. & Narayanan, A., 1995. *Quantum - inspired neural networks*, Exeter: Dept. of Computer Science, University of Exeter, UK.
- Milburn, G., 1998. *The Feynman Processor*. Reading: Perseus Books.
- Miller, A., 1993. *The applicationn of Neural Networks to Imaging and Signal Processing in Astronomy and Medicine*. Ph.D. thesis, Faculty of Science, Department of Physics, University of Southampton.
- Miller, A., Blott, B. & Hames, T., 1992. Review of Neural Network Applications in Medical Imaging and Signal Processing. *Medical and Biological Engineering and Computing*, 30(5), pp. 449-464.
- Miller, A., Blott, B. & Hames, T., 1992. Review of Neural Network Applications in Medical Imaging and Signal Processing. *Medical and Biological Engineering and Computing*, 30(5), pp. 449-464.
- Minsky, M. & Papert, S., 1969. *Perceptrons*. s.l.:MIT Press.

- Mordeciai, A., 2003. Nonlinear Programming: Analysis and Methods. 2nd επιμ. σ.Ι.:Dover Publishing.
- Mount, D., 2004. Bioinformatics: Sequence and Genome Analysis. Cold Spring Harbor: Laboratory Press.
- Müller, B., Reinhardt, J. & Strickland, M., 1995. Neural Networks. Heidelberg: Springer-Verlag.
- Nomikos, C. & Valianatos, 1998. Electromagnetic variations associated with the seismicity of the frontal hellenic arc. *Geologica Carpathica*.
- Nomikos, K. και συν., 1994. Development of a telemetric system for observation of radioemission associated with earthquakes in Crete island. *Proceedings of the XXIV General Assembly, European Seismological Commission,, Τόμος II*, pp. 1112-1120.
- Nomikos, K. και συν., 1997. Latest aspects of electromagnetic variations associated with shallow and intermediate depth earthquakes in South Aegean. *Annali di Geophysica*.
- Ossen, A., Zamzow, T., Oswald, H. & Fleck, E., 1994. Segmentation of Medical Images using Neural Network Classifiers. *Ifeachor, International Conference on Neural Networks and Expert Systems in Medicine and Healthcare*, pp. 427-432.
- Özkan, M., Sprenkels, H. & Dawant, B., 1990. Multi Spectral Magnetic Resonance Image Segmentation Using Neural Networks. *International Joint Conference on Neural Networks, Τόμος 1*, pp. 429-434.
- Papageorgiou, C. & Wells, A., 1999. Process and metacognitive dimensions of depressive and anxious thoughts and relationships with emotional intensity. *Clinical Psychology and Psychotherapy, Τόμος 6*, pp. 156-162.
- Papazachos, B. & Papazachou, K., 1989. Earthquakes of Greece. Thessaloniki: ZETE Publications.
- Parker, D., 1982. Learning Logic, California: Office of Technology Licencing, Stanford.
- Parzen, E., 1962. On the estimation of a probability density function and mode. *Annals of Mathematical Statistics, Τόμος 3*, pp. 1065-1076.
- Penrose, R., 1994. *Shadows of the Mind. A search for the missing science of consciousness*. New York, Oxford: Oxford University Press.
- Perus, M., 1997. Neural networks, quantum systems and consciousness. *Science Tribune, May*.

- Pham, T., Satou, K. & Ho, T., 2005. Support vector machines for prediction and analysis of beta and gamma turns in proteins. *Journal Bioinform. Comput. Biol.* , 3(2), pp. 343-358.
- Pincus, S., 1991. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences of the United States of America*, 88(6), pp. 2297-2301.
- Pincus, S. & Keefe, D., 1992. Quantification of hormone pulsatility via an approximate entropy algorithm. *American Journal of Physiology, Endocrinology and Metabolism*, 262(5 25-5), pp. 741-754.
- Purves, D. & Andrews, T., 1997. Correlated size variations in human visual cortex, lateral geniculate nucleus and optic tract. *J Neurosci* , Τόμος 17, pp. 2859-2868.
- Raghu, P. & Yegnanarayana, B., 1998. Supervised texture classification using a probabilistic neural network and constraint satisfaction model. *IEEE Transactions on Neural Networks*, 9(3), pp. 516-522.
- Rajapakse, J. & Acharya, R., 1990. Medical Image Segmentation with MARA. *International Joint Conference on Neural Networks*, pp. 965-9722.
- Richman, J. & Moorman, J., 2000. Physiological time series analysis using approximate and sample entropy. *American Journal of Physiological: Heart and Circulatory Physiology*, 278(6 74-6), pp. 2039-2049.
- Romero, R., Touretzky, D. & Thibadeu, R., 1997. Optical chinese character recognition using probabilistic neural networks. *Pattern Recognition*, 30(7), pp. 1279-1292.
- Rosenblatt, 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, Τόμος 65, p. 386.
- Rosenblatt, F., 1962. *Principles of Neurodynamics*. Washington DC: Spartan Books.
- Rost, B. & Sander, C., 1993. Prediction of protein secondary structure at better than 70% accuracy. *Journal Mol. Biol.*, Τόμος 232, pp. 584-599.
- Rumelhart, D. & McClelland, J., 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. 2nd επιμ. Cambridge, Mass.: MIT Press.
- Rundle, J., Tiampo, K., Klein, W. & Sa Martins, J., 2002. Self-organization in leaky threshold systems: the influence of near mean field dynamics and its implications for EQs, neurology, and forecasting. *PNAS*, Τόμος 99, pp. 2514-2521.
- Sandler, Y. M., 1990. Model of neural networks with selective memorization and chaotic behavior. *Physics Letters A*, Τόμος 144 , pp. 462-466].

- Schwenk, H. & Milgram, M., 1994. Structured diablo-networks for hand-written character recognition. International Conference on Artificial Neural Networks, Τόμος 2, pp. 985-988.
- Shapiro, V., Muminov, V. & Abtullabecov, K., 1994. High precision magnetometry for earthquake prediction in Uzbekistan: Ninety one forecasts between 1982. Electromagnetic Phenomena related to Earthquake prediction, Terra Pub..
- Shor, P., 1997. Polynomial-time algorithm for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing, Τόμος 26, pp. 1484-1509.
- Skarda, C. & Freeman, W., 1987. How brains make chaos in order to make sense of the world behavioral and Brain Sciences. Open Peer Commentary, Τόμος 10, pp. 161-195.
- Smirnova, N. & Hayakawa, M., 2007. Fractal Characteristics of the ground observed ULF emissions in relation to geomagnetic and seismic activities. J. Atmos. Sol. Terr. Phys. , Τόμος 69, pp. 1833-1841.
- Smirnova, N., Hayakawa, M. & Gotoh, K., 2004. Precursory behavior of fractal characteristics of the ULF electromagnetic fields in seismic active zones before strong earthquakes. Phys. Chem. Earth, Τόμος 29, pp. 445-451.
- Sordo, M., 2002. Introduction to Neural Networks in Healthcare. OpenClinical.
- Sordo, M., Buxton, H. & Watson, D., 2001. A hybrid approach to breast cancer diagnosis. Practical Applications of Computational Intelligence Techniques.
- Specht, D., 1990. Probabilistic Neural Networks. Neural Networks, 1(3), pp. 109-118.
- Spect, D. & Romsdahl, H., 1994. Experience with adaptive probabilistic neural network and adaptive general regression neural network. Proceedings of the IEEE International Conference on Neural Networks, Τόμος 2, pp. 1203-1208.
- Stewart, I., 1989. Does God Play Dice? Mathematics of Chaos. s.l.:Blackwell.
- Storella, R. και συν., 1998. Approximate entropy and point correlation dimension of heart rate variability in healthy subjects. Integrative Physiological and Behavioral Science, 33(4), pp. 315-320.
- Theodoridis & Koutroumbas, 2009. Pattern Recognition. 4th επιμ. s.l.:s.n.
- Touretzky, D. & Pomerlau, D., 1989. What is hidden in the hidden layers?. s.l.:Byte.
- Vasquez, M., 1996. Modeling side-chain conformation. Curr. Opin. Struct. Biology, 6(2), pp. 217-221.
- Ventura, D. & Martinez, T., 1998. Quantum associative memory with exponential capacity. s.l., Proceedings of the International Joint Conference on Neural Networks.

- Ventura, D. & Martinez, T., 1999. Initializing the amplitude distribution of a quantum state. Foundations of Physics Letters.
- Walker, C., 1996. Analysis of Multispectral Microscope Images Using Neural Networks. Surface and Interface Analysis, 24(3), pp. 173-180.
- Waltrus, R., Towell, G. & Glassman, M., 1993. Application of Neural Networks Tools to ECG Patient Monitoring. Synthesize, Optimize, Analyze, Repeat - SOAR.
- Weinstein, J., Kohn, K. & Grever, M., 1992. Neural Computing in Cancer Drug Development: Predicting Mechanism. Action. Science, Τόμος 258, pp. 447-451.
- Werbos, P., 1990. Backpropagation through time: what it does and how to do it. Proceedings of the IEEE, Τόμος 78, p. 1550.
- Widrow, B. & Lehr, M., 1990. 30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation. Proceedings of the IEEE, 78(9), pp. 1415-1420.
- Wikipedia, 2012. Wikipedia. [Ηλεκτρονικό] Available at: http://el.wikipedia.org/wiki/Θεωρία_του_χάους [Πρόσβαση 20 November 2012].
- Xing, G. & Feltham, R., 1994. Pyramidal Neural Networking for Mammogram Tumour Pattern Recognition. IEEE International Conference on Neural Networks, Τόμος VI, pp. 3546-3551.
- Yao, Y., Freeman, W., Burke, B. & Yang, Q., 1991. Pattern Recognition by a distributed neural network an industrial application.
- Yu, D., Small, M., Harrison, R. & Diks, C., 2000. Efficient implementation of the Gaussian kernel algorithm in estimating invariants and noise level from noisy time series data. Physical Review E: Statistical Physics, Plasmas, Fluids and Related Interdisciplinary Topics, 61(4A), pp. 3750-3756.
- Zhang, Y., 2008. Progress and challenges in protein structure prediction. Curr. Opin. Struct. Biol., 18(3), pp. 342-348.
- Zheng, B., Qian, W. & Clarke, L., 1994. Multistage Neural Network for Pattern Recognition in Mammogram Screening. IEEE International Conference on Neural Networks, Τόμος VI, pp. 3537-3441.
- Zimmermann, O. & Hansmann, U., 2006. Support vector machines for prediction of dihedral angle regions. Bioinformatics, 22(24), pp. 3009-3015.

10. Κατάλογος Εικόνων

ΕΙΚΟΝΑ 1. (Α) ΠΛΑΓΙΑ ΟΨΗ - (Β) ΓΩΝΙΑΚΗ ΤΟΜΗ	- 11 -
ΕΙΚΟΝΑ 2. ΝΕΥΡΩΝΙΚΑ ΚΥΤΤΑΡΑ. ΠΗΓΗ: SCIENTIFIC AMERICAN	- 12 -
ΕΙΚΟΝΑ 3. ΜΕΓΕΘΥΣΗ ΝΕΥΡΩΝΙΚΟΥ ΚΥΤΤΑΡΟΥ. ΠΗΓΗ: SCIENTIFIC AMERICAN	- 13 -
ΕΙΚΟΝΑ 4. Α) ΝΕΥΡ. ΝΩΤ. ΜΥΕΛΟΥ Β)ΚΥΤΤΑΡΟ ΙΠΠΟΚΑΜΠΟΥ Γ) ΚΥΤΤΑΡΟ ΠΑΡΕΓΚΕΦΑΛΙΔΑΣ	- 14 -
ΕΙΚΟΝΑ 5. ΔΙΑΓΡΑΜΜΑ ΗΛΕΚΤΡΟΧΗΜΙΚΟΙΥ ΣΗΜΑΤΟΣ ΝΕΥΡΟΔΙΑΒΙΒΑΣΤΩΝ.	- 14 -
ΕΙΚΟΝΑ 6.ΜΗΧΑΝΙΣΜΟΣ ΔΗΜΙΟΥΡΓΙΑΣ ΚΑΙ ΡΟΗΣ ΔΥΝΑΜΙΚΟΥ ΚΑΤΑ (WALKER, 1996)	- 18 -
ΕΙΚΟΝΑ 7. ΑΝΑΠΑΡΑΣΤΑΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΝΕΥΡΩΝΑ ΚΑΤΑ ΤΗΝ ΜΕΤΑΒΙΒΑΣΗ ΣΗΜΑΤΟΣ	- 19 -
ΕΙΚΟΝΑ 8. ΟΙ ΦΑΣΕΙΣ ΤΗΣ ΗΛΕΚΤΡΟΧΗΜΙΚΗΣ ΜΕΤΑΒΙΒΑΣΗΣ ΤΩΝ ΣΗΜΑΤΩΝ ΣΤΙΣ ΣΥΝΑΨΕΙΣ	- 22 -
ΕΙΚΟΝΑ 9. ΤΕΧΝΗΤΟ ΝΕΥΡΩΝΙΚΟ ΔΙΚΤΥΟ	- 26 -
ΕΙΚΟΝΑ 10. ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΕΝΟΣ ΕΠΙΠΕΔΟΥ	- 27 -
ΕΙΚΟΝΑ 11. Η ΣΥΝΑΡΤΗΣΗ ΚΑΤΑ HEAVISIDE	- 28 -
ΕΙΚΟΝΑ 12. ΤΕΧΝΗΤΟ ΝΕΥΡΩΝΙΚΟ ΔΙΚΤΥΟ ΚΑΘΟΛΙΚΗΣ ΣΥΝΔΕΣΕΩΣ ΠΟΛΛΩΝ ΕΠΙΠΕΔΩΝ	- 29 -
ΕΙΚΟΝΑ 13. Η ΣΥΝΑΡΤΗΣΗ ΚΑΤΑ HEAVISIDE	- 30 -
ΕΙΚΟΝΑ 14. ΤΟ ΜΟΝΤΕΛΟ ΤΟΥ ΑΙΣΘΗΤΗΡΑ	- 33 -
ΕΙΚΟΝΑ 15. ΠΟΛΥΕΠΙΠΕΔΟ ΠΡΟΤΥΠΟ ΑΙΣΘΗΤΗΡΑ	- 34 -
ΕΙΚΟΝΑ 16. ΔΙΚΤΥΟ ΠΟΛΛΑΠΛΩΝ ΕΠΙΠΕΔΩΝ ΜΕ ΕΠΙΠΕΔΟ ΕΣΩΤΕΡΙΚΗΣ ΑΝΑΠΑΡΑΣΤΑΣΗΣ	- 38 -
ΕΙΚΟΝΑ 17. ΔΥ-ΕΠΙΠΕΔΟ ΣΥΣΤΗΜΑ ΝΕΥΡΩΝΩΝ ΜΕ ΑΝΑΔΡΑΣΗ	- 41 -
ΕΙΚΟΝΑ 18. Ο ΕΛΚΥΣΤΗΣ ΤΟΥ ΛΟΡΕΝΖ	- 43 -
ΕΙΚΟΝΑ 19. ΚΒΑΝΤΙΚΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΚΑΤΑ ΕΖΗΟΝ ΚΑΙ VENTURA.	- 55 -
ΕΙΚΟΝΑ 20. ΘΕΜΕΛΙΩΔΗΣ ΔΙΑΓΡΑΜΜΑ ΔΟΜΗΣ ΥΠΟΛΟΓΙΣΤΗ	- 63 -
ΕΙΚΟΝΑ 21.ΤΟ ΠΕΙΡΑΜΑ ΤΟΥ ΤΙΟΥΡΙΝΓΚ	- 63 -
ΕΙΚΟΝΑ 22. ΔΙΑΓΡΑΜΜΑ ΒΑΣΙΚΩΝ ΣΤΑΔΙΩΝ ΣΧΕΔΙΑΣΜΟΥ ΣΥΣΤΗΜΑΤΟΣ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ.	- 73 -
ΕΙΚΟΝΑ 23. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ ΣΥΝΑΡΤΗΣΕΩΣ ΑΚΤΙΝΙΚΗΣ ΒΑΣΗΣ	- 75 -
ΕΙΚΟΝΑ 24. ΔΙΑΓΡ.ΝΚ-ΜΕΣΩΝ ΜΕ ΑΝΕΥΡΕΣΗ ΤΩΝ ΚΕΝΤΡΩΝ ΤΩΝ ΠΕΡΙΟΧΩΝ.	- 77 -
ΕΙΚΟΝΑ 25. ΔΙΑΓΡΑΜΜΑ ΠΙΘΑΝΟΤΙΚΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ, (FOLLAND, ET AL., 2004)	- 79 -
ΕΙΚΟΝΑ 26. ΔΙΑΓΡ. ΣΥΓΚΡΙΣΗΣ ΕΝΤΡΟΠΙΩΝ ΣΕ ΠΡΟ-ΣΕΙΣΜΙΚΟ ΣΗΜΑ ΣΕΙΣΜΟΣ ΑΘΗΝΑΣ 1999	- 84 -
ΕΙΚΟΝΑ 27. ΧΑΡΤΗΣ ΑΠΕΙΚΟΝΙΣΗΣ ΣΕΙΣΜΩΝ ΣΤΗΝ ΕΛΛΑΔΑ ΤΑ ΤΕΛΕΥΤΑΙΑ 30 ΧΡΟΝΙΑ	- 88 -
ΕΙΚΟΝΑ 28. ΓΕΩΤΕΚΤΟΝΙΚΟΣ ΧΑΡΤΗΣ ΤΗΣ ΕΛΛΑΔΑΣ (ΦΟΤΟΡΟΥΛΟΣ, ET AL., 2012)	- 86 -
ΕΙΚΟΝΑ 29. ΔΙΑΓΡΑΜΜΑ ΜΕΓ. ΣΕΙΣΜΩΝ ΓΙΑ $M_L > 5$ ΣΤΗΝ ΕΛΛΑΔΑ ΤΑ ΤΕΛΕΥΤΑΙΑ 30 ΧΡΟΝΙΑ.	- 87 -
ΕΙΚΟΝΑ 30. ΧΑΡΤΕΣ ΑΠΕΙΚΟΝΙΣΕΩΝ ΣΕΙΣΜΩΝ ΜΕΓΑΘΟΥΣ $M_L > 4$ ΚΑΙ $M_L > 6$ ΣΤΗΝ ΕΛΛΑΔΑ.	- 88 -
ΕΙΚΟΝΑ 31. ΣΧΗΜΑΤΙΚΟ ΤΗΛΕΜΑΤΙΚΟΥ ΣΤΑΘΜΟΥ ΜΕΤΡ. Η/Μ ΔΙΑΤΑΡΧΩΝ	- 89 -
ΕΙΚΟΝΑ 32. ΤΗΛΕΜΑΤΙΚΟ ΔΙΚΤΥΟ ΜΕΤΡΗΣΕΩΝ ΜΗΖ/ΚΗΖ ΣΥΧΝΟΤΗΤΩΝ	- 90 -
ΕΙΚΟΝΑ 33. ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ΤΟΥ GUIDE ΤΟΥ MATLAB	- 92 -
ΕΙΚΟΝΑ 34. ΠΑΡΑΘΥΡΟ ΙΔΙΟΤΗΤΩΝ ΕΝΟΣ ΑΝΤΙΚΕΙΜΕΝΟΥ ΣΤΟ GUIDE	- 92 -
ΕΙΚΟΝΑ 35. ΜΟΝΟΠΑΤΙ ΕΚΤΕΛΕΣΗΣ ΕΝΟΣ GUI M-FILE ΠΗΓΗ: MATHWORKS	- 93 -
ΕΙΚΟΝΑ 36. ΑΝΑΠΤΥΞΗ GUI ΜΟΝΤΕΛΟΥ ΣΤΟ GUIDE	- 94 -
ΕΙΚΟΝΑ 37. ΠΑΡΑΘΥΡΟ ΔΙΕΠΑΦΗΣ ΓΙΑ ΤΗΝ ΕΠΙΛΟΓΗ ΗΜΕΡΟΜΗΝΙΑΣ	- 95 -
ΕΙΚΟΝΑ 38. ΠΡΟΒΟΛΗ ΔΙΑΔΡΟΜΩΝ ΕΠΙΛΕΓΟΜΕΝΩΝ ΑΡΧΕΙΩΝ ΚΑΙ ΦΑΚΕΛΩΝ ΠΡΟΣ ΜΕΛΕΤΗ	- 96 -
ΕΙΚΟΝΑ 39. ΕΠΙΛΟΓΗ ΚΑΝΑΛΙΩΝ ΑΠΟ ΤΟ GUI ΚΑΙ ΦΙΛΤΡΑΡΙΣΜΑ ΤΟΥΣ	- 96 -
ΕΙΚΟΝΑ 40. ΠΑΡΑΘΥΡΟ ΑΛΛΑΓΗΣ ΟΝΟΜΑΣΙΑΣ ΚΑΝΑΛΙΟΥ	- 97 -
ΕΙΚΟΝΑ 41. ΧΡΗΣΗ GOOGLE MAP ΓΙΑ ΤΗΝ ΑΠΟΤΥΠΩΣΗ ΣΕΙΣΜΩΝ ΜΕ $M_L > 6$ ΓΙΑ ΤΟ ΕΤΟΣ 2008.	- 97 -
ΕΙΚΟΝΑ 42. ΟΙ ΣΕΙΣΜΟΙ ΣΤΟΝ ΕΛΛΗΝΙΚΟ ΧΩΡΟ ΚΑΤΑ ΤΟ 2008 ΜΕ $M_L > 6$	- 98 -
ΕΙΚΟΝΑ 43. ΕΠΙΛΟΓΗ ΜΕΘΟΔΩΝ ΓΙΑ ΑΝΑΛΥΣΗ ΣΗΜΑΤΩΝ ΑΠΟ ΤΟ GUI	- 98 -

ΕΙΚΟΝΑ 44. GUI ΓΙΑ ΤΗΝ ΔΙΑΧ. ΜΕΤΡΗΣΕΩΝ Η/Μ ΠΕΔΙΟΥ ΑΠΟ ΠΟΛ. ΣΤΑΘΜΟΥΣ ΜΕΤΡΗΣΗΣ	- 99 -
ΕΙΚΟΝΑ 45. GUI ΓΙΑ ΤΗΝ ΑΠΟΤΥΠΩΣΗ ΜΗΖ/ΚΗΖ ΚΑΝ/ΩΝ, ΣΕΙΣΜ.ΔΙΑΤΑΡΑΧΩΝ ΚΑΙ ΜΕΘΟΔΩΝ	- 100 -
ΕΙΚΟΝΑ 46. ΔΙΑΦΟΡ. ΦΑΣΕΙΣ Χ/ΣΕΙΡΩΝ ΣΕΙΣΜΟΥ ΑΘΗΝΑΣ 1999 ΚΑΙ ΕΠΙΛΗΨΙΑΣ ΣΕ ΠΟΝΤΙΚΙ	- 101 -
ΕΙΚΟΝΑ 47. (Α) ΑΝΑΠΑΡΑΣΤΑΣΗ ΔΟΜΗΣ ΤΗΣ ΠΡΩΤΕΪΝΗΣ. (Β) ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΩΝΙΩΝ	- 103 -
ΕΙΚΟΝΑ 48. ΔΙΑΓΡΑΜΜΑ BLOCK ΓΙΑ ΤΗΝ ΑΝΑΠ. ΤΟΥ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ ΑΠΟ ΤΟ MATLAB	- 108 -
ΕΙΚΟΝΑ 49. ΛΟΓΑΡΙΘΜΟ-ΣΙΓΜΟΕΙΔΗΣ ΣΥΝΑΡΤΗΣΗ ΜΕΤΑΦΟΡΑΣ	- 109 -
ΕΙΚΟΝΑ 50. ΣΤΙΓΜΙΟΤΥΠΟ ΕΚΠΑΙΔΕΥΣΕΩΣ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ ΑΝΑΓΝΩΡΙΣΗΣ ΠΡΟΤΥΠΩΝ	- 110 -
ΕΙΚΟΝΑ 51. ΔΙΑΡΘΡΩΤΙΚΕΣ ΑΝΑΘΕΣΕΙΣ ΣΕ ΔΕΔΟΜΕΝΑ ΕΚΠΑΙΔΕΥΣΗΣ	- 112 -
ΕΙΚΟΝΑ 52. ΔΙΑΡΘΡΩΤΙΚΕΣ ΑΝΑΘΕΣΕΙΣ ΣΕ ΔΕΔΟΜΕΝΑ ΕΠΑΛΛΗΘΕΥΣΗΣ	- 113 -
ΕΙΚΟΝΑ 53. ΔΙΑΡΘΡΩΤΙΚΕΣ ΑΝΑΘΕΣΕΙΣ ΣΕ ΔΕΔΟΜΕΝΑ TESTING	- 113 -
ΕΙΚΟΝΑ 54. ΑΠΟΔΟΣΗ ΕΚΠΑΙΔΕΥΣΗΣ, ΕΛΕΓΧΟΥ ΚΑΙ ΔΟΚΙΜΗΣ	- 114 -
ΕΙΚΟΝΑ 55. ΑΠΟΔΟΣΗ ΕΠΙΚΥΡΩΣΗΣ ΔΕΔΟΜΕΝΩΝ	- 115 -
ΕΙΚΟΝΑ 56. ΠΙΝΑΚΑΣ ΣΥΓΧΗΣΕΩΣ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ	- 116 -
ΕΙΚΟΝΑ 57. ΓΡΑΦΗΜΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΥ ΛΕΙΤΟΥΡΓΙΑΣ ΔΕΚΤΗ	- 117 -
ΕΙΚΟΝΑ 58. ΕΚ ΝΕΟΥ ΕΚΠΑΙΔΕΥΣΗ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ	- 119 -
ΕΙΚΟΝΑ 59. ΠΙΝ. ΣΥΓΧΗΣΕΩΣ ΓΙΑ ΤΗΝ ΕΚΠΑΙΔΕΥΣΗ, ΕΠΙΚΥΡΩΣΗ ΚΑΙ ΤΗΝ ΔΟΚΙΜΗ ΑΚΟΛΟΥΘΙΩΝ	- 120 -
ΕΙΚΟΝΑ 60. ΕΠΙΔΟΣΗ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ	- 121 -
ΕΙΚΟΝΑ 61. ΝΕΟ ΓΡΑΦΗΜΑ ΚΑΤΑΣΤΑΣΗΣ ΕΠΑΛΛΗΘΕΥΣΗΣ	- 121 -
ΕΙΚΟΝΑ 62. ΓΡΑΦΗΜΑΤΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΥ ΛΕΙΤΟΥΡΓΙΚΟΥ ΔΕΚΤΗ	- 122 -
ΕΙΚΟΝΑ 63. ΤΕΛΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΡΟΒΛΕΨΕΩΝ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ	- 123 -

11. Κατάλογος Εξισώσεων

$$f(x) = \begin{cases} 1 & \text{εαν } X > \theta \text{ ή } X = \theta \\ 0 & \text{εαν } X < \theta \end{cases},$$

ΕΞΙΣΩΣΗ 1 - 30 -

$$W_{i|N} + 1 = W_{i|N} + AX_{i|N},$$

ΕΞΙΣΩΣΗ 2 - 33 -

$$X = \sum_{i=0}^n x_i w_i,$$

ΕΞΙΣΩΣΗ 3 - 34 -

$$X = \sum_{i=0}^n x_i w_i + b,$$

ΕΞΙΣΩΣΗ 4 - 34 -

$$T_{ij} = \sum_{s=1}^P \sigma_i^s \sigma_j^s, T_{ij} = 0 \text{ } i, j = 1, \dots, N,$$

ΕΞΙΣΩΣΗ 5 - 53 -

$$|\psi\rangle = \sum_{s=1}^P |\sigma_1^s \sigma_2^s \dots \sigma_N^s\rangle,$$

ΕΞΙΣΩΣΗ 6 - 54 -

$$FX = F(|X-C|),$$

ΕΞΙΣΩΣΗ 7 - 74 -

$$R = |X-C|,$$

ΕΞΙΣΩΣΗ 8 - 74 -

$$O = I = 1NWIG(X, QI),$$

ΕΞΙΣΩΣΗ 9 - 74 -

$$G = G(X1, Q1)G(X1, Q2)KG(X1, QN) \dots KKKG(XM, QN),$$

ΕΞΙΣΩΣΗ 10 - 74 -

$$\text{ERROR} = B-GWTB-GW = |B-GW|^2 = J = 1M(BJ-I = 1NWIG(X, QI))^2,$$

ΕΞΙΣΩΣΗ 11 - 74 -

$$\Sigma 1 = \dots = \Sigma K = \Sigma = D2P,$$

ΕΞΙΣΩΣΗ 12 - 76 -

$CI = 1|XI|P \in XIXP,$ *ΕΞΙΣΩΣΗ 13 - 77 -*

$YI = J = 1KWIIAJ + WIO,$ *ΕΞΙΣΩΣΗ 14 - 77 -*

$NTR = K = 1KNK,$ *ΕΞΙΣΩΣΗ 15 - 79 -*

$FIKX = 1(2Π)P2|Σ|12EXP(-12(X-XIK)TΣ-1(X-XIK)),$ *ΕΞΙΣΩΣΗ 16 - 79 -*

$FIKX = 1(2ΠΣ2)P2EXP(-X-XIK22Σ2),$ *ΕΞΙΣΩΣΗ 17 - 80 -*

$GKX = WKI = 1MKFI, KX, ME K = 1, 2, \dots, K,$ *ΕΞΙΣΩΣΗ 18 - 80 -*

$K = 1KWK = 1,$ *ΕΞΙΣΩΣΗ 19 - 80 -*

$CX = ARGMAX(GK)1 \leq K \leq K,$ *ΕΞΙΣΩΣΗ 20 - 80 -*

$XIM = UI, UI + 1, \dots, UI + M-1, ME I = 1, 2, \dots, N-M + 1,$ *ΕΞΙΣΩΣΗ 21 - 82 -*

$\Phi MR = (N-M + 1)-1I = 1N-M + 1LNCIM(R),$ *ΕΞΙΣΩΣΗ 22 - 82 -*

$APEN(M, R) = LIMN \rightarrow \infty[\Phi MR - \Phi M + 1(R)],$ *ΕΞΙΣΩΣΗ 23 - 82 -*

$BMR = (N-M)-1I = 1N-MBIM(R),$ *ΕΞΙΣΩΣΗ 24 - 82 -*

$AMR = (N-M)-1I = 1N-MAIM(R),$ *ΕΞΙΣΩΣΗ 25 - 82 -*

$SAMPEN(M, R, N) = -LNAM(R)BM(R),$ *ΕΞΙΣΩΣΗ 26 - 83 -*

$XIM = UI, UI + 1, \dots, UI + M-1-U0(I), ME I = 1, 2, \dots, N-M + 1,$ *ΕΞΙΣΩΣΗ 27 - 83 -*

$U0I = M-1J = 0M-1U(I + J),$ *ΕΞΙΣΩΣΗ 28 - 83 -*

$$DIJM = M(DIJM, R),$$

ΕΞΙΣΩΣΗ 29 - 83 -

$$\Phi IMR = (N-M-1)-1J = 1, J \neq IN-MDIJM,$$

ΕΞΙΣΩΣΗ 30 - 83 -

$$\Phi MR = (N-M)-1I = 1N-M\Phi IM(R),$$

ΕΞΙΣΩΣΗ 31 - 83 -

$$\Phi M + 1R = (N-M)-1I = 1N-M\Phi IM + 1(R),$$

ΕΞΙΣΩΣΗ 32 - 84 -

$$FUZZYEN(M, R) = LN\Phi MR-LN\Phi M + 1(R),$$

ΕΞΙΣΩΣΗ 33 - 84 -

$$S(F) = A \cdot F-B,$$

ΕΞΙΣΩΣΗ 34 - 85 -

12. Κατάλογος Πινάκων

ΠΙΝΑΚΑΣ 1. ΧΡΟΝΟΛΟΓΙΚΟΣ ΠΙΝΑΚΑΣ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 31 -
ΠΙΝΑΚΑΣ 2. ΠΙΝΑΚΑΣ ΧΡΗΣΙΜΟΤΗΤΩΝ ΚΑΙ ΜΕΙΟΝΕΚΤΗΜΑΤΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 39 -
ΠΙΝΑΚΑΣ 3. ΠΙΝΑΚΑΣ ΑΝΑΛΟΓΙΑΣ ΚΒΑΝΤΟΜΗΧΑΝΙΚΗΣ ΚΑΙ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 50 -
ΠΙΝΑΚΑΣ 4. ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΔΙΑΦΟΡΩΝ ΕΚΔΟΣΕΩΝ ΚΒΑΝΤΙΚΩΝ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ	- 59 -
ΠΙΝΑΚΑΣ 5. ΚΩΔΙΚΟΙ ΣΤΑΘΜΩΝ ΜΕΤΡΗΣΕΩΝ	- 91 -

13. Ευρετήριο Κυριότερων Όρων

BACKPROPAGATION:

8,34,116,121

FEYNMAN: 46,115,118

GOOGLE MAP:123

GUI:9,80,81,83,84,85,88,99,123,124,142,346,347

HEBB:8,30,33,116

HOPFIELD:8,29,35,37,42,43,48,49,55,116,117

HURST: 90

KHZ:78,84,90,115,123,124

PERCEPTRON:8,30,45,121

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ:
8,47,61,63,68

ΑΣΑΦΗΣ ΕΝΤΡΟΠΙΑ:
9,71,72,73,83

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ:
10,105

ΔΕΙΓΜΑΤΟΛΗΠΤΙΚΗ

ΕΝΤΡΟΠΙΑ: 9,72,73

ΔΙΚΤΥΑ ΣΥΝΑΡΤΗΣΕΩΝ

ΑΚΤΙΝΙΚΗΣ ΒΑΣΗΣ: 9,64,65

ΚΒΑΝΤΙΚΑ ΝΕΥΡΩΝΙΚΑ

ΔΙΚΤΥΑ:
8,45,46,47,48,49,50,51,52,53,54

ΜΑΘΗΣΗ ΜΕ ΕΠΟΠΤΕΙΑ: 33

ΜΗΖ: 78,124

ΝΕΥΡΟΔΙΑΒΙΒΑΣΤΕΣ: 8,15

ΟΜΑΔΟΠΟΙΗΣΗΣ ΤΩΝ Κ-ΜΕΣΩΝ: 9,67

ΠΙΘΑΝΟΤΙΚΑ ΝΕΥΡΩΝΙΚΑ

ΔΙΚΤΥΑ: 9,68

ΠΟΛΛΑΠΛΟΤΗΤΑ: 8,48

ΠΟΛΥΠΛΟΚΟΤΗΤΑ:
9,44,67,70,71,74,90,91,93

ΠΡΟΒΛΕΨΗ:

10,54,60,92,93,94,95,106,112

ΠΡΟΣΕΓΓΙΣΤΙΚΗ ΕΝΤΡΟΠΙΑ:
9,71, 91

ΠΡΩΤΕΪΝΗ: 10,92,93

ΠΥΚΝΟΤΗΤΑ ΦΑΣΜΑΤΟΣ

ΙΣΧΥΟΣ: 74,75,83,90,346,422

ΥΠΕΡΘΕΣΗ: 46

ΦΡΑΚΤΑΛ: 9,74

ΧΑΟΣ: 8,42

ΧΑΟΤΙΚΑ ΝΕΥΡΩΝΙΚΑ

ΔΙΚΤΥΑ: 8,38

ΧΑΡΑΚΤΗΡΙΣΤΙΚΟΣ

ΛΕΙΤΟΥΡΓΙΚΟΣ ΔΕΚΤΗΣ:
109,110,124

14. Παράρτημα Α – Ιστορικές Αναφορές Σεισμών της αρχαιότητας

Η καταγραφή των παρακάτω ιστορικών δεδομένων προέρχεται από: α) Ηρόδοτος, Ιστορικά Βιβλία (Μετάφραση Διεθύνσεως Στρατιωτικών Εκδόσεων ΑΣ), Αθήνα 1970. β) Στράβων, Γεωγραφικά (Εισαγωγή – Μετάφραση – Σχόλια: Π. Βαλάκης), Αθήνα γ) Το αλφαβητάρι των σεισμών, Ελένη Ι. Ιωαννίδου, Εκδόσεις Δίαυλος, Αθήνα, 1997.

1. Ηρόδοτος – ΣΤ', 27. Σεισμός του 494 π. Χ. στην Χίο, με 119 μαθητές νεκρούς
«...Κατά τον αυτόν χρόνον και ολίγον προ της ναυμαχίας, κατέπεσεν η στέγη σχολείου της πόλεως, και εξ εκατόν είκοσι παιδίων τα οποία ήσαν εκεί και κατεπλακώθησαν, εν μόνον εσώθη. Αυτά τα σημεία έδειξεν ο Θεός προηγουμένως εις αυτούς, και μετά από αυτά επηκολούθησεν η ναυμαχία και κατελήφθη η πόλις. Μετά δε την ναυμαχίαν επήλθεν ο Ιστιαίος μετά των Λεσβίων και ευρών αυτούς εξαντλημένους, ευκόλως τους υπέταξαν.»
2. Ηρόδοτος – ΣΤ', 44. Σεισμός και τσουνάμι έτους 479 π.Χ. με 20,000 νεκρούς
«...Τα λοιπά έθνη τα εντός της Μακεδονίας είχαν ήδη υποταχθή εις αυτούς (τους Πέρσες). Από την Θάσον διαπεράσαντες πλησίον της αντικειμένης ακτής, έπλεον μέχρι Ακάνθου, από την οποίαν ορμώμενοι προσεπάθουν να περιπλεύσουν τον Άθω. Παραπλεόντες όμως τούτον, σφοδρός και ενάντιος βόρειος άνεμος επιπεσών επέφερεν εις αυτούς μεγάλας καταστροφάς, ρίνας έξω εις Άθωνα πολλά πλοία. Λέγεται ότι τα καταστρεφέντα πλοία ανήρχοντο περίπου εις τριακόσια, και οι απολεσθέντες άνθρωποι πλέον των είκοσι χιλιάδων. Επειδή δε η περί τον Άθωνα θάλασσα βρίθει τεράτων, άλλοι εξ αυτών ηρπάζοντο και κατεβροχθίζοντο υπό τούτων, άλλοι συνετρίβοντο επί των βράχων, άλλοι επνίγοντο μη γνωρίζοντες να κολυμβούν, και άλλοι τέλος απέθνησκον εκ τους ψύχους. Αυτά λοιπόν έπαθεν ο στόλος.»
3. Ηρόδοτος – ΣΤ', 98 Σεισμός του 481 π.Χ. στην Δήλο
«...Αφού έπραξαν αυτά ο Δάτις, έχων μεθ' εαυτού τους Ίωνας και τους Αιολείς, έπλευσεν με τον στρατόν του το πρώτον εναντίον της Ερετρίας. Μετά την αναχώρησίν του από εκεί εγένετο σεισμός εις την Δήλον, δια πρώτην και τελευταίαν φοράν μέχρι σήμερον, ως λέγουν οι Δήλιοι. Τούτο βεβαίως ήτο σημείον διά του οποίου ο Θεός έδειξεν εις τους ανθρώπους τα μέλλοντα να συμβώσι κακά. Διότι επί Δαρείου του Υστάσπου, Ξέρξου του υιού του Ξέρξου, επί τρεις δηλαδή διαδοχικάς γενεάς εις την Ελλάδα εγένοντο υπό των Περσών, άλλα δε υπ' αυτών τούτων

των κορυγαίων εκ των Ελλήνων, οίτινες ηγωνίζοντο διά την αρχήν. Ουδέν λοιπόν το παράδοξον το ότι εκινήθη η Δήλος, η οποία κατά το παρελθόν ήτο ακίνητος (Και εις χρησμόν τινα είχε γραφή δι' αυτήν το εξής: Θα κινήσω και την Δήλον, μολονότι ακίνητον). Τα ονόματα Δαρείος, Ξέρξης, Αρταξέρξης μεταφραζόμενα εις την Ελληνικήν σημαίνουν, το μεν Δαρείος, δραστήριος, το δε Ξέρξης, πολεμικός και το Αρταξέρξης πολεμικώτατος. Δικαίως λοιπόν οι Έλληνες δύνανται να ονομάζουν τους βασιλείς αυτούς εις την γλώσσαν των έτσι.»

4. Θουκυδίδης – Ιστοριών Δ', 52 Σεισμός του 424 π. Χ.
«...Κατά το επόμενον θέρος και αμέσως με την νέαν σελήνην έγινε έκλεισις ηλίου και σεισμός κατά τας αρχάς του ίδιου μηνός.»
5. Θουκυδίδης – Ιστοριών Ε', 45 Σεισμός του 420 π. Χ.
«...Όταν δηλαδή οι πρέσβεις ενεφανίσθησαν ενώπιον της εκκλησίας του δήμου και ηρωτήθησαν, ηρνήθησαν ότι έχουν έλθει με απόλυτον εξουσιοδότησιν, όπως είχαν καταθέσει ενώπιον της βουλής, οι δε Αθηναίοι δεν συνεκρατούντο πλέον, αλλά με προσοχήν ήκουον την εισήγησιν του Αλκιβιάδου, ο οποίος κατεφέρετο πολύ σφοδρότερον από πρωύτερα κατά των Λακεδαιμονίων, και ήσαν έτοιμοι την στιγμήν εκείνην να εισαγάγουν εις την συνεδρίασιν του Αργείους πρέσβεις και τους συντρόφους των και να συνάψουν συμμαχίαν με αυτούς, επειδή όμως έγινε σεισμός πριν ληφθή καμμία οριστική απόφασις, ανεβλήθη η εκκλησία εκείνη.»
6. Θουκυδίδης – Ιστοριών Ε', 50. Σεισμός του 420 π.Χ.
«...Μετά την εορτήν των Ολυμπίων ήλθον εις την Κόρινθον οι Αργείοι και οι σύμμαχοί των διά να παρακαλέσουν τους Κορινθίους να ενωθούν μαί των. Κατά τύχην παρευρίσκοντο εκεί και πρέσβεις των Λακεδαιμονίων. Μετά πολλάς συζητήσεις δεν έγινε τελικώς τίποτε, επειδή ένεκα σεισμού διελύθησαν και οι διάφοροι πρέσβεις επέστρεψαν εις την πατρίδα των. Με τα γεγονότα αυτά τελειώνει το θέρος.»
7. Θουκυδίδης – Ιστοριών ΣΤ', 95. Σεισμός του 414 π.Χ.
«...Κατά την αυτήν άνοιξιν εκστρατεύσαντες οι Λακεδαιμόνιοι εναντίον του Άργους έφθασαν μεν μέχρι των Κλεωνών, αλλά επειδή έγινε σεισμός επανήλθον εις τα ίδια. Μετά ταύτα οι Αργείοι εισβάλλοντες εις την γειτονική Θυρεάτιν ήρπασαν πολλά λάφυρα από τους Λακεδαιμονίους, τα οποία επώλησαν αντί είκοσι πέντε περίπου ταλάντων.»
8. Θουκυδίδης – Ιστοριών Η', 41. Σεισμός του 414 π. Χ.
«...Κατά τον διάπλουν απεβιβάσθη εις την Μεροπίδαν Κων, η οποία ήτο ατείχιστος και είχε καταστροφή υπό σεισμού, του μεγίστου από όσους ενθυμούμεθα, και την ελεηλάτησεν, ενώ οι κάτοικοι είχαν καταφύγει εις τα όρη, ενεργών δε επιδρομάς εις την χώραν την ελεηλάτει, και μόνον τους ελευθέρους άνδρας άφινεν ελευθέρους.»

9. Στράβων – Γεωγραφικά Α', 3

«...Αφού δε είπε και αυτός (ο Ερατοσθένης) πόσον πολύ προήχθησαν αι περί της Οικουμένης γνώσεις των ζησάντων μετά τον Αλέξανδρον και των συγχρόνων αυτού, φέρει κατόπιν τον λόγον εις το περί του σχήματος ζήτημα, ουχί μόνον της Οικουμένης, όπερ είναι σχετικώτερον προς την (γεωγραφικήν) ταύτης εξέτασιν, αλλά περί του σχήματος συμπάσης της γης, διότι πρέπει μεν να μνημονεύση τις και τούτο, αλλ' ουχί κατά τρόπον ακατάστατον. Αφού δε είπεν ότι η σύμπασα είναι σφαιροειδής, ουχί όμως ως εκ τόρνου, αλλ' έχει ανωμαλίας τινάς, προσθέτει το πλήθος των κατά σειράν γενομένων μεταβολών του σχήματος αυτής, αίτινες προέρχονται και εκ του ύδατος και πυρός και σεισμών, και αναφυσήσεως ανέμων προς τα άνω (ηφαιστειών) και άλλων προμοίων, χωρίς να τηρή και ενταύθα την τάξιν. Διότι το μεν σφαιροειδές όλης της γής προέρχεται εκ της διαρκούς καταστάσεως του όλου, αι δε τοιαύται μεταβολαί σχήματος την μεν όλην γην ουδόλως αλλοιώνουσι, διότι μεταξύ των μεγάλων αφανίζονται τα ούτω μικρά της δε οικουμένης διατάξεις άλλας και άλλας μεταβάλλουσι και τας προσεχείς αιτίας άλλας και άλλας έχουσι.»

10. Στράβων – Γεωγραφικά, Α', 10. Σεισμός του 373 π. Χ.

«...Την τοιαύτην λοιπόν αιτιολογίαν ολιγώτερον ήθελε τις αποδεχθή, μάλλον δε εκ των φανερωτέρων και των καθημερινώς τρόπον τινά βλεπομένων πρέπει να αναζητήσωμεν την αιτίαν, διότι και κατακλυσμοί και σεισμοί και αναφυσήσεις ανέμων και εξογκώσεις των υπό το ύδωρ της θαλάσσης μερών ανυψώνουσι και την θάλασσαν, αι δε καταπτώσεις χαμηλώνουσιν αυτήν, διότι ουχί μεν μύδροι δύνανται να αναφανώσι και μικραί νήσοι, μεγάλοι δε ουχί, ούτε νήσοι μεν, ήπειροι δε ουχί. Ομοίως δε και καταπτώσεις και μικραί και μεγάλοι δύναται να γίνωσιν, αν ακόμη και χάσματα και καταπόσεις χωρίων και κατοικιών γίνωνται, ως εις την Βούραν και την Βιζώνην και άλλους περισσοτέρους τόπους, εγένετο, ως λέγουσιν, υπό σεισμού, και την Σικελίαν δύναται, τις ομοίως να εικάση ως αποκεκομμένην εκ της Ιταλίας ή ότι αφού ανεφάνη εκ του βυθού της θαλάσσης ένεκα του πυρός της Αίτνης παρέμεινε ομοίως δε και αι Λιπάροι νήσοι και αι Πιθηκούσαι.»

11. Στράβων – Γεωγραφικά, Α', 16

«...Τώρα όμως το ασυνήθιστον συνταράσσει τον νουν και δεικνύει απειρίαν των κατά φύσιν συμβαινόντων και όλου του βίου, όπως εάν τις λέγη τα συμβάντα περί τας νήσους Θήραν και Θηρασίαν, κειμένας μεταξύ Κρήτης και Κυρηναϊκής πορθμόν, εξ ων η Θήρα είναι μητρόπολις της Κυρήνης, και τα περί την Αίγυπτον και πολλά τοιαύτα μέρη της Ελλάδος. Εν τω μεταξύ δηλαδή Θήρας και Θηρασίας διαστήματι εξήρχοντο εκ του πελάγους φλόγες επί τέσσαρας ημέρας, εις τρόπον ώστε να βράζη και να φλέγεται ολόκληρος η θάλασσα, ανεφάνη δε ολίγον κατ'

ολίγον ανηψουμένη, ως εάν συνετελείτο οργανική σύνθεσις εκ πευρακτωμένων ογκολίθων, νήσος έχουσα περίμετρον δώδεκα σταδίων. Μετά δε την κατάπαυσιν του φαινομένου πρώτοι οι Ρόδιοι, οι οποίοι εκυριάρχουν της θαλάσσης, ανεθάρρησαν και προσέπλευσαν προς τον τόπον και ίδρυσαν επί της νήσου ιερὸν Ποσειδώνος Ασφαλίου. Εἰς δε την Φοινίκην λέγει ο Ποσειδώνιος ὅτι γενομένου σεισμοῦ κατεπόθη πόλις ἣτις ἔκειτο ἀνωθεν της Σιδόνος, και αὐτῆς δε της Σιδόνος σχεδόν τα δύο τρίτα κατέπεσαν, ἀλλ' οὐχὶ ὅλα ομοῦ, ὥστε να μη γίνη μεγάλη φθορά ἀνθρώπων, το ἴδιον δε φαινόμενον ἐφθασε και εἰς ολόκληρον την Συρίαν, ἀλλὰ κάπως μετρίως. Διέβη δε και εἰς τινας νήσους, εἰς τα Κυκλάδας και εἰς την Εὐβοίαν, ὥστε της Αρεθούσης (ἣτις εἶναι κρήνη εν Χαλκίδι) αἱ πηγαὶ ἐπαυσαν να ρέωσι, κατόπιν δε ἀπὸ πολλὰς ἡμέρας ἐξεχύθη μεθ' ὀρμῆς το ὕδωρ ἀπὸ ἄλλο στόμιον, δεν ἐπαυσε δε να σείεται κατὰ τόπους η νήσος παρά ἀφοῦ ἠνοιχθῆ εἰς την πεδιάδα του Λήλαντος χάσμα γῆς, το ὁποῖον ἐξῆμесе ποταμὸν πηλοῦ φλογισμένου.»

12. Στράβων – Γεωγραφικά Α', 17.

«...Μνημονεύει δε σχετικῶς (ο Δημήτριος του Σκηψίου) προς ταῦτα τα λεγόμενα ὑπὸ του Δημοκλέους, ὅστις ἀναφέρει κάποιους μεγάλους σεισμούς οι ὁποῖοι ἐγίναν κατὰ τους παλαιούς χρόνους περί την Λυδίαν και την Ἰωνίαν μέχρι της Τρωάδος, ὑπὸ των ὁποίων και κωμοπόλεις ἐξηφανίσθησαν και η Σίπυλος κατεστράφη ὅτε ἦτο βασιλεὺς ο Τάνταλος... και ἀπὸ ἑλῆ ἐγίναν λίμναι, την δε Τροίαν κατέκλυσε το κύμα. Η δε νήσος Φάρος, η κατὰ την Αἴγυπτον, ἦτο πτε εν μέσω του πελάγους, νυν δε κατέστη τρόπον τινά χερσόνησος, ὁμοίως δε και η Τύρος και αἱ Κλαζομεναί. Ὅτε δε ἡμεῖς ἐπεδημούμεν εν Αλεξανδρείᾳ της Αιγύπτου, ὑψωθέν το πέλαγος περί το Πηλούσιον και το Κάσιον ὄρος κατέκλυσε την ξηρὰν και κατέστησε νήσον το ὄρος, ὥστε να γίνη πλωτὴ η προς την Φοινίκη οδός...»

13. Στράβων – Γεωγραφικά Δ', Α', 7

«...Και το μεν εν παράδοξον πράγμα που ἔχουν τα προλεχθέντα παράλια, εἶναι τα δι' ἀνασκαφῆς συλλαμβανόμενα ψάρια. Ἢδη θα προσθέσω ἄλλο που εἶναι ἀκόμη σχεδόν παραδοξότερον ἀπὸ αὐτό: δηλαδή μεταξύ της Μασσαλίας και των ἐκβολῶν του Ροδανού εἶναι πεδιάς ευρισκόμενη μακρὰν της θαλάσσης ἑκατὸν στάδια και ἔχουσα ἄλλην τόσῃν την διάμετρον, το δε σχῆμα κυκλικόν. Αὐτὴ ὀνομάζεται Λιθώδης, διότι πράγματι εἶναι τοιαύτη, ἦτοι εἶναι γεμάτη ἀπὸ πέτρας μεγάλας, ὥστε μόλις δύναται κάθε μια να χωρέση εἰς την χεῖρα μας, κάτωθεν των πετρῶν αὐτῶν φυτρῶνουν ἄγρια χόρτα παραπολλά, κατὰλληλα δια την διατροφήν ποιμνίων, ἐνὼ εἰς το μέσον της πεδιάδος εἶναι νερά και ἀλμυραὶ πηγαὶ και ἀλυκαί. Και ὅλη η κειμένη ὑψηλότερα χώρα εἶναι ἀνεμώδης ἀλλὰ ἰδίως εἰς την πεδιάδα αὐτὴν φυσὰ με ὀρμὴν σκοτεινιάζων βόρειος ἀνεμος, σφοδρὸς και φρικώδης

μάλιστα λέγουν ότι και λίθοι σύρονται και κυλίνονται και ότι
ρίπτονται με το φύσημά του και από τα όπλα και από τα ενδύματα.
Και ο μιν Αριστοτέλης λέγει ότι από τους σεισμούς τους
ανασείοντας την γην εκ των κάτω προς τα άνω, οι λίθοι
εκσφενδονισθέντες κάτωθεν της γης εις την επιφάνειαν
κατωλίσθησαν εις τα κοιλότητας του εδάφους. Ο Ποσειδώνιος δε
λέγει ότι υπήρξεν εδώ λίμνη, η οποία εξηράνθη, συνοδευομένη
από δυνατόν κυματισμόν και διά τούτο ο πυθμήν της ετεμαχίσθη
εις περισσότερους λίθους ομοιάζοντας με τα χαλίκια των
παταμών και τα στρογγυλά πετραδάκια, τα οποία είναι όμοια το
εν με το άλλο και λεία και εκτός της ομοιότητος ίσα κατά το
μέγεθος. Και οι δύο βεβαίως έδωκαν επαρκή εξήγησιν της αιτίας,
αφού οι λόγοι των είναι πιθανοί, διότι εξ ανάγκης οι έχοντες
τοιαύτην την σύστασιν λίθοι δεν εσχηματίσθησαν μόνοι των,
αλλά διά της επιδράσεως υγρού τινός έλαβον την μορφήν αυτήν,
ή απεχωρίσθησαν από βράχους μεγάλους, οι οποίοι υπέστησαν
αλλεπάλληλα ρήγματα...»

15. Παράρτημα Β – Ανεπτυγμένος κώδικας της παρούσας πτυχιακής

15.1 Ανεπτυγμένος κώδικας Προσεγγιστικής Εντροπίας

```
function ApEn = ApEntr(u,m,r)

%This is a function to calculate Approximate Entropy
%
% INPUTS:
% u      is a Time-Series Fragment to be analyzed
% m      is the length of compared vectors of the Time-Series
Fragment
%       Pincus suggests m=2 (increase of m reduces matches)
% r      is the critical distance between vectors (increase of
r increases
%       matches). 2004 Nicolopoulos (Natural Hazards and Earth
System
%       Sciences (2004) 4: 615-631) suggests that for seismic
EM the
%       best choice is  $r=0.65*STD$ , where STD is the standard
deviation
%       of the time-series. 2000 Richman (Am J Physiol Heart
Circ Physiol
%       278: H2039-H2049, 2000) suggests that "It is convenient
to set
%       the tolerance as  $r*STD$ , the standard deviation
of the data set,
%       allowing measurements on data sets with different
%       amplitudes to be compared"
%
% Before the analysis of a signal, an optimal selection of m,
r should be
% done, by checking multiple (m,r) combinations that result to
clear
% detection of a critical event, already detected by other
means.
%
% OUTPUT:
% ApEn   is the Approximate Entropy of the Time-Series Fragment
%
% References:
```

```

% "Measuring complexity using FuzzyEn, ApEn, and SampEn"
Chen et. al.
% Medical Engineering & Physics 31 (2009) 61-68
% &
% "Approximate entropy as a measure of system complexity"
Pincus SM
% Proceedings of the National Academy of Sciences of the
United States
% of America 1991;88(6):2297-301
%

```

```

%% Approximate Entropy [ApEn] Calculation
N = length(u);
vs_max = N-m+1;%vector sequences max number

```

```

%% Calculations for m
phi_m = 0;
X = zeros(vs_max,m);
parfor i = 1:vs_max %multicore execution
    X(i,:) = u(i:(i+m-1));
end
C =ones(vs_max,1);
for i = 1:vs_max
    for j = i+1:vs_max
        d = max(abs(X(i,:)-X(j,:)));
        if d<=r
            C(i) = C(i)+1;
            C(j) = C(j)+1;
        end
    end
end
end

```

```

C = C/(vs_max);
phi_m = sum(log(C))/(vs_max);

```

```

%% Calculations for m+1
phi_mplus1 = 0;
m = m+1;
vs_max = N-m+1;%vector sequences max number
X = zeros(vs_max,m);
parfor i = 1:vs_max
    X(i,:) = u(i:(i+m-1));
end
C = ones(vs_max,1);
for i = 1:vs_max
    for j = i+1:vs_max

```

```

        d = max(abs(X(i,:)-X(j,:)));
        if d<=r
            C(i) = C(i)+1;
            C(j) = C(j)+1;
        end
    end
end
C = C/(vs_max);
phi_mplus1 = sum(log(C))/(vs_max);

ApEn = (phi_m-phi_mplus1);

```

15.2 Ανεπτυγμένος κώδικας Δειγματοληπτικής Εντροπίας

```

function SampEn = SampEntr(u,m,r)

%This is a function to calculate Sample Entropy
%
% INPUTS:
% u      is a Time-Series Fragment to be analyzed
% m      is the length of compared vectors of the Time-Series
Fragment
%       Pincus suggests m=2 (increase of m reduces matches)
% r      is the critical distance between vectors (increase of
r increases
%       matches). 2004 Nicolopoulos (Natural Hazards and Earth
System
%       Sciences (2004) 4: 615-631) suggests that for seismic
EM the
%       best choice is  $r=0.65*STD$ , where STD is the standard
deviation
%       of the time-series. 2000 Richman (Am J Physiol Heart
Circ Physiol
%       278: H2039-H2049, 2000) suggests that "It is convenient
to set
%       the tolerance as  $r*STD$ , the standard deviation
of the data set,
%       allowing measurements on data sets with different
%       amplitudes to be compared"
%
% Before the analysis of a signal, an optimal selection of m,
r should be
% done, by checking multiple (m,r) combinations that result to

```

```

clear
% detection of a critical event, already detected by other
means.
%
% OUTPUT:
% SampEn is the Sample Entropy of the Time-Series Fragment
%
% References:
% "Measuring complexity using FuzzyEn, ApEn, and SampEn"
Chen et. al.
% Medical Engineering & Physics 31 (2009) 61-68
% &
% "Physiological time-series analysis using approximate
entropy
and sample entropy" J.S. Richman & J.R. Moorman
% American Journal of Physiology: Heart and Circulatory
Physiology
% 2000;278(6 47-6):H2039-49
%
%

%% Sample Entropy [SampEn] Calculation
N = length(u);
vs_max = N-m+1;%vector sequences max number

%% Calculation of B
X = zeros(vs_max,m);
parfor i = 1:vs_max %multicore execution
    X(i,:) = u(i:(i+m-1));
end
B = zeros(vs_max,1);

for i = 1:vs_max-1
    for j = i+1:vs_max-1
        d = max(abs(X(i,:)-X(j,:)));
        if d<=r
            B(i) = B(i)+1;
            B(j) = B(j)+1;
        end
    end
end

end

B = B/(vs_max-2);
Bm = 0;
Bm = sum(B)/(vs_max-1);

```

```

%% Calculation of A
m = m+1;
vs_max = N-m+1;%vector sequences max number
X = zeros(vs_max,m);
parfor i = 1:vs_max
    X(i,:) = u(i:(i+m-1));
end
A = zeros(vs_max,1);

for i = 1:vs_max-1
    for j = i+1:vs_max-1
        d = max(abs(X(i,:)-X(j,:)));
        if d<=r
            A(i) = A(i)+1;
            A(j) = A(j)+1;
        end
    end
end

A = A/(vs_max-2);
Am = 0;
Am = sum(A)/(vs_max-1);

%% Calculation of SampEn
SampEn = -log(Am/Bm);

```

15.3 Ανεπτυγμένος κώδικας Ασαφούς Εντροπίας

```

function FuzzyEn = FuzzyEntr(u,m,r,n)

%This is a function to calculate Fuzzy Entropy
%
% INPUTS:
% u      is a Time-Series Fragment to be analyzed
% m      is the length of compared vectors of the Time-Series
Fragment
%       Pincus suggests m=2 (increase of m reduces matches)
% r      is the critical distance between vectors (increase of
r increases
%       matches). 2004 Nicolopoulos (Natural Hazards and Earth
System
%       Sciences (2004) 4: 615-631) suggests that for seismic
EM the
%       best choice is  $r=0.65*STD$ , where STD is the standard

```

```

deviation
%       of the time-series. 2000 Richman (Am J Physiol Heart
Circ Physiol
%       278: H2039-H2049, 2000) suggests that "It is convenient
to set
%       the tolerance as  $r \cdot \text{STD}$ , the standard deviation
of the data set,
%       allowing measurements on data sets with different
%       amplitudes to be compared"
% n     power of the the family of exponential membership
functions  $\exp( ? (dij/r)^n )$ 
%       The gradient n acts as the weight of vectors'
similarity.
%       A value of  $n > 1$  weights the similarity degree of
the close
%       vectors and unweights that of the far ones, while a
value
%       of  $n < 1$  functions contrarily. The larger n is, the
more the
%       closer vectors and the less the further vectors are
weighted.
%
% Before the analysis of a signal, an optimal selection of m,
r, n should be
% done, by checking multiple (m,r,n) combinations that result
to clear
% detection of a critical event, already detected by other
means.
%
% OUTPUT:
% FEn   is the Fuzzy Entropy of the Time-Series Fragment
%
% References:
%       "Measuring complexity using FuzzyEn, ApEn, and SampEn"
Chen et. al.
%       Medical Engineering & Physics 31 (2009) 61-68
%

```

```

%% Fuzzy Entropy [FuzzyEn] Calculation
N = length(u);
vs_max = N-m+1; %vector sequences max number

%% Calculations for m
phi_m = 0;
X = zeros(vs_max,m);

```



```

parfor i = 1:vs_max
    X(i,:) = u(i:(i+m-1));
    u0(i) = sum(u(i:i+m-1))/m;
end
U0=u0'*ones(1,m);
X = X-U0;
clear u0

D = zeros(vs_max,vs_max);

for i = 1:vs_max-1
    for j = i+1:vs_max-1
        d = max(abs(X(i,:)-X(j,:)));
        DD = exp(-(d/r)^n);
        D(i,j) = DD;
        D(j,i) = DD;
    end
end

phi_i_m = sum(D)/(vs_max-2); % there is no point to exclude
i=j, because the D(i,i) is zero, eitherway
phi_m = sum(phi_i_m)/(vs_max-1);

%% Calculations for m+1
phi_mplus1 = 0;
m = m+1;
vs_max = N-m+1;%vector sequences max number
X = zeros(vs_max,m);

parfor i = 1:vs_max
    X(i,:) = u(i:(i+m-1));
    u0(i) = sum(u(i:i+m-1))/m;
end
U0=u0'*ones(1,m);
X = X-U0;
clear u0

D = zeros(vs_max,vs_max);

for i = 1:vs_max-1
    for j = i+1:vs_max-1
        d = max(abs(X(i,:)-X(j,:)));
        DD = exp(-(d/r)^n);
        D(i,j) = DD;
    end
end

```

```

        D(j,i) = DD;
    end
end

phi_i_mplus1 = sum(D)/(vs_max-2);
phi_mplus1 = sum(phi_i_mplus1)/(vs_max-1);

FuzzyEn = log(phi_m)-log(phi_mplus1);

```

15.4 Ανεπτυγμένος κώδικας γραφικής διεπαφής χρήστη για την διαχείριση και μελέτη μετρήσεων του ηλεκτομαγνητικού πεδίου απο πολλαπλούς σταθμούς μέτρησης.

```

function varargout = EarthGui(varargin)
% EARTHGUI MATLAB code for EarthGui.fig
%     EARTHGUI, by itself, creates a new EARTHGUI or raises
the existing
%     singleton*.
%
%     H = EARTHGUI returns the handle to a new EARTHGUI or
the handle to
%     the existing singleton*.
%
%     EARTHGUI('CALLBACK',hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in EARTHGUI.M with the given
input arguments.
%
%     EARTHGUI('Property','Value',...) creates a new EARTHGUI
or raises the
%     existing singleton*. Starting from the left, property
value pairs are
%     applied to the GUI before EarthGui_OpeningFcn gets
called. An
%     unrecognized property name or invalid value makes
property application
%     stop. All inputs are passed to EarthGui_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI
allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

```

```

% Edit the above text to modify the response to help EarthGui

% Last Modified by GUIDE v2.5 14-Dec-2012 03:35:13

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @EarthGui_OpeningFcn, ...
                  'gui_OutputFcn',  @EarthGui_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
    varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before EarthGui is made visible.
function EarthGui_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to EarthGui (see VARARGIN)

% Choose default command line output for EarthGui
handles.output = hObject;

%Set default values for time
handles.timecheck1=[0,0,0];
handles.timecheck2=[23,59,59];

%Set default fsize
handles.fsize=1024;

```

```

handles.fsize2=1024;
handles.fsize3=1024;
handles.fsize4=1024;

set(handles radiobutton2,'value',1);
set(handles radiobutton3,'value',0);
set(handles radiobutton6,'value',1);
set(handles radiobutton7,'value',0);
set(handles radiobutton8, 'value',1);
set(handles radiobutton9, 'value', 0);
handles.radio_choice='0';
set(handles.text39,'string','1');
set(handles.text38,'string','1');
handles.plot_dropdown_names='Init. Line';
%handles.plot_fig_commands='plot(rand(15));axis tight;';

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes EarthGui wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command
line.
function varargout = EarthGui_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see
VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject     handle to pushbutton1 (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)
[filename, pathname] = uigetfile( ...
{ '*.txt;', 'Text (*.txt)'; ...

```

```

'*.m;*.fig;*.mat;*.slx;*.mdl','MATLAB
(*.m,*.fig,*.mat,*.slx,*.mdl)';
    '*.m', 'Code files (*.m)'; ...
    '*.fig','Figures (*.fig)'; ...
    '*.mat','MAT-files (*.mat)'; ...
    '*.mdl;*.slx','Models (*.slx, *.mdl)'; ...
    '*.*', 'All Files (*.*)'}, ...
    'Select a file with magnitudes');

handles.path=pathname;
handles.file=filename;
datain=[pathname,filename];
handles.datain=datain;
set(handles.text1,'String',datain);
guidata(hObject,handles);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, ~, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

EQ_data = importdata(handles.datain);
EQ_data.textdata;
EQ_data.data;
% EQ_M = EQ_data.data(:,end);
% EQ_LAT = EQ_data.data(:,5);
% EQ_LONG = EQ_data.data(:,6);
% EQ_Date_Time = zeros(length(EQ_data.data(:,1)),30);

%Init counter
i=1;

%Get low limit
tmp1=datevec(get(handles.text3,'String'));
datecheck1=tmp1(:,[1:3]);

%Get up limit
tmp2=datevec(get(handles.text4,'String'));
datecheck2=tmp2(:,[1:3]);
cur_dates=zeros(1,6);

```

```

%
%           startdate=datetime(eval([num2str(tmp1(2)),'-
',num2str(tmp1(3)),'-',num2str(tmp1(1))]));
%           stopdate=datetime(eval([num2str(tmp2(2)),'-
',num2str(tmp2(3)),'-',num2str(tmp2(1))]));

startdate=datetime(sprintf([num2str(tmp1(2)),'-
',num2str(tmp1(3)),'-',num2str(tmp1(1))]));
stopdate=datetime(sprintf([num2str(tmp2(2)),'-
',num2str(tmp2(3)),'-',num2str(tmp2(1))]));

assignin('base','startdate',startdate);
assignin('base','stopdate',stopdate);

for n = 1:length(EQ_data.data(:,1))

    %Get date to check in date vector
    EQ_Date_Time = cell2mat([num2str(EQ_data.data(n,1)),'-
',EQ_data.textdata(n,2),'-',...
    EQ_data.textdata(n,1),'
',
num2str(EQ_data.data(n,2)),':',...
    num2str(EQ_data.data(n,3)),':',
num2str(fix(EQ_data.data(n,4)))]);
    curlinedate=datevec(EQ_Date_Time);

    %Check if current date exceeds the upper limit
    if (etime([datecheck2,handles.timecheck2],curlinedate)
<0)
        break;
    end

    %Check if current time is above the lower limit
    if
(etime(curlinedate,[datecheck1,handles.timecheck1]) >0)
        EQ_M(i) = EQ_data.data(n,end);
        EQ_LAT(i) = EQ_data.data(n,5);
        EQ_LONG(i) = EQ_data.data(n,6);
        cur_dates =[cur_dates;curlinedate];
        i=i+1;
    end
end
end
%trim the first zero line
cur_dates=cur_dates(2:length(cur_dates),:);

```

```

% computation of intersets intervals in seconds
interv=[];
for i=1:length(cur_dates)-1
    interv(i)=etime(cur_dates((i+1,:),:),cur_dates(i,:));
end
% figure; plot(interv)

msg=msgbox('EQ Magnitudes were readed');
handles.EQ_M=EQ_M;
handles.EQ_LAT=EQ_LAT;
handles.EQ_LONG=EQ_LONG;
handles.interv=interv;

guidata(hObject, handles);

%figure; plot(EQ_M);axis tight;

%
+++++
+++++
%               Plot Fuzzy Entropy of Magnitudes
%
+++++
+++++

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

EQ_Data=handles.EQ_M;

fsize = handles.fsize;
ovl = str2double(get(handles.edit4,'String')); % percentage of
overlapping
inc=fsize*(1-ovl/100);%increment in samples

NoF=floor((length(EQ_Data)-fsize)/inc)+1;
EQ_DataFrame=EQ_Data(1:fsize);

M = str2double(get(handles.edit1,'String'));
N=str2double(get(handles.edit2,'String'));

```

```

r_var=str2double(get(handles.edit3,'String'));

FuzzyEn(1:fsize) = zeros(fsize,1);
for nFrame=2:NoF
    %sprintf('Remaing execution times: %d\n*',NoF-nFrame)
    indx1=(nFrame-1)*inc+1;
    indx2=(nFrame-1)*inc+fsize;

    r =r_var*std(EQ_Data(indx1:indx2));

    % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
end
FuzzyEn(1:fsize) = FuzzyEn(inc+1:fsize+inc);

% figure;
%
% subplot(3,1,1);
% plot(EQ_Data,'k')
% axis tight;
% title('Input Signal vs. Time');
%
% subplot(3,1,[2:3]);
% plot(FuzzyEn,'k')
% axis tight;
% title_text=['Fuzzy Entropy (W=',num2str(fsize),'',
Overlap=',num2str(ovl),'%) vs. Time'];
% title(title_text);

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('Fuzzy Entropy Magn')];

%handles.plot_fig_commands=[handles.plot_fig_commands;cellstr
('plot(handles.EQ_M)')];
guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
guidata(hObject,handles);
%print -f2 -dtiff 'Fuzzy Entropy Optimized'
%
% function FuzzyEn = FuzzyEntr_v3(u,m,r,n)
%
% %This is a function to calculate Fuzzy Entropy
% %
% % INPUTS:
% % u is a Time-Series Fragment to be analyzed

```



```

%% m is the length of compared vectors of the Time-Series
Fragment
%% Pincus suggests m=2 (increase of m reduces matches)
%% r is the critical distance between vectors (increase
of r increases
%% matches). 2004 Nicolopoulos (Natural Hazards and
Earth System
%% Sciences (2004) 4: 615-631) suggests that for seismic
EM the
%% best choice is  $r=0.65*STD$ , where STD is the standard
deviation
%% of the time-series. 2000 Richman (Am J Physiol Heart
Circ Physiol
%% 278: H2039-H2049, 2000) suggests that "It is
convenient to set
%% the tolerance as  $r*STD$ , the standard deviation
of the data set,
%% allowing measurements on data sets with
different
%% amplitudes to be compared"
%% n power of the the family of exponential membership
functions  $\exp(-d_{ij}/r)^n$ 
%% The gradient n acts as the weight of vectors'
similarity.
%% A value of  $n > 1$  weights the similarity degree of
the close
%% vectors and unweights that of the far ones, while
a value
%% of  $n < 1$  functions contrarily. The larger n is,
the more the
%% closer vectors and the less the further vectors
are weighted.
%%
%% Before the analysis of a signal, an optimal selection of
m, r, n should be
%% done, by checking multiple (m,r,n) combinations that result
to clear
%% detection of a critical event, already detected by other
means.
%%
%% OUTPUT:
%% FEn is the Fuzzy Entropy of the Time-Series Fragment
%%
%% References:
%% "Measuring complexity using FuzzyEn, ApEn, and SampEn"
Chen et. al.

```

```

%% Medical Engineering & Physics 31 (2009) 61-68
%%
%% Developed by Stelios Potirakis, Maria Barbarosou, Anax
Fotopoulos 2011
%
%% Fuzzy Entropy [FuzzyEn] Calculation
%
% N = length(u);
% %vector sequences max number
%
%% Calculations for m
%% phi_m = 0;
%% phi_mplus1 = 0;
%
% vs_max = N-m+1;
% X = zeros(vs_max,m);
% X2 = zeros(vs_max-1,m+1);
%
%
% parfor i = 1:vs_max
%     X(i,:) = u(i:(i+m-1));
%     u0(i) = sum(u(i:i+m-1))/m;
%     while (i~=vs_max)
%         X2(i,:) = u(i:(i+m));
%         u02(i) = sum(u(i:i+m))/m+1;
%         break;
%     end
% end
% U0=u0'*ones(1,m);
% U02=u02'*ones(1,m+1);
% X = X-U0;
% X2 = X2-U02;
% clear u0
% clear u02
%
% D = zeros(vs_max,vs_max);
% D2 = zeros(vs_max-1,vs_max-1);
%
% for i = 1:vs_max-1
%     if (i+1~=vs_max)
%         %for j = i+1:vs_max-1
%             d = max(abs(X(i,:)-X(i+1,:)));
%             DD = exp(-(d/r)^n);
%             D(i,i+1) = DD;
%             D(i+1,i) = DD;
%             while (i~=vs_max)

```

```

%         d2 = max(abs(X2(i,:)-X2(i+1,:)));
%         DD2 = exp(-(d2/r)^n);
%         D2(i,i+1) = DD2;
%         D2(i+1,i) = DD2;
%         break;
%     end
% %end
% end
% end
%
% phi_i_m = sum(D)/(vs_max-2); % there is no point to exclude
i=j, because the D(i,i) is zero, eitherway
% phi_i_mplus1 = sum(D2)/(vs_max-3);
% phi_m = sum(phi_i_m)/(vs_max-1);
% phi_mplus1 = sum(phi_i_mplus1)/(vs_max-2);
%
% FuzzyEn = log(phi_m)-log(phi_mplus1);
% %sprintf('Fuzzy Entropy Calculated at: %d',FuzzyEn)
%
%
```

```

%=====Plot EQ Magnitudes Data=====
% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject     handle to pushbutton7 (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)
%figure; plot(handles.EQ_M1);axis tight;

%plotgui(handles.EQ_M)

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('EQ Magnitudes')];

%handles.plot_fig_commands=[handles.plot_fig_commands;cellstr
('plot(handles.EQ_M)')];
guiplot(handles.plot_dropdown_names,handles.EQ_M,0,0,0,0,0,4)
```

;

```
guidata(hObject,handles);
```

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Create a UICALENDAR with the following properties:
% 1) Highlight weekend dates.
% 2) Only allow a single date to be selected at a time.
% 3) Send the selected date to the edit box uicontrol.
uicalendar('SelectionType', 1, ...
'DestinationUI', handles.text3);
% tmp1=datevec(get(handles.text3,'String'));
% handles.datecheck1=tmp1(:,[1:3]);
guidata(hObject,handles);
```

```
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Create a UICALENDAR with the following properties:
% 1) Highlight weekend dates.
% 2) Only allow a single date to be selected at a time.
% 3) Send the selected date to the edit box uicontrol.
uicalendar('SelectionType', 1, ...
'DestinationUI', handles.text4);
% tmp2=datevec(get(handles.text4,'String'));
% handles.datecheck2=tmp2(:,[1:3]);
guidata(hObject,handles);
```

```

% --- Executes on selection change in hours_start.
function hours_start_Callback(hObject, eventdata, handles)
% hObject    handle to hours_start (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
hours_start contents as cell array
%         contents{get(hObject,'Value')} returns selected item
from hours_start
row=get(handles.hours_start,'Value');
string=get(handles.hours_start,'String');
handles.timecheck1(1)=str2num(strtrim(string{row}));
guidata(hObject,handles);

% --- Executes on selection change in min_start.
function min_start_Callback(hObject, eventdata, handles)
% hObject    handle to min_start (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
min_start contents as cell array
%         contents{get(hObject,'Value')} returns selected item
from min_start
row=get(handles.min_start,'Value');
string=get(handles.min_start,'String');
handles.timecheck1(2)=str2num(strtrim(string{row}));
guidata(hObject,handles);

% --- Executes on selection change in sec_start.
function sec_start_Callback(hObject, eventdata, handles)
% hObject    handle to sec_start (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
sec_start contents as cell array
%         contents{get(hObject,'Value')} returns selected item
from sec_start

```

```

row=get(handles.sec_start,'Value');
string=get(handles.sec_start,'String');
handles.timecheck1(3)=str2num(strtrim(string{row}));
guidata(hObject,handles);

```

```

% --- Executes on selection change in hours_stop.
function hours_stop_Callback(hObject, eventdata, handles)
% hObject    handle to hours_stop (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject,'String')) returns
hours_stop contents as cell array
%          contents{get(hObject,'Value')} returns selected item
from hours_stop
row=get(handles.hours_stop,'Value');
string=get(handles.hours_stop,'String');
handles.timecheck2(1)=str2num(strtrim(string{row}));
guidata(hObject,handles);

```

```

% --- Executes on selection change in min_stop.
function min_stop_Callback(hObject, eventdata, handles)
% hObject    handle to min_stop (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: contents = cellstr(get(hObject,'String')) returns
min_stop contents as cell array
%          contents{get(hObject,'Value')} returns selected item
from min_stop
row=get(handles.min_stop,'Value');
string=get(handles.min_stop,'String');
handles.timecheck2(2)=str2num(strtrim(string{row}));
guidata(hObject,handles);

```

```

% --- Executes on selection change in sec_stop.
function sec_stop_Callback(hObject, eventdata, handles)
% hObject    handle to sec_stop (see GCBO)
% eventdata  reserved - to be defined in a future version of

```

```

MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
sec_stop contents as cell array
%           contents{get(hObject,'Value')} returns selected item
from sec_stop
row=get(handles.sec_stop,'Value');
string=get(handles.sec_stop,'String');
handles.timecheck2(3)=str2num(strtrim(string{row}));
guidata(hObject,handles);

% --- Executes during object creation, after setting all
properties.
function hours_start_CreateFcn(hObject, eventdata, handles)
% hObject      handle to hours_start (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%           See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes during object creation, after setting all
properties.
function min_start_CreateFcn(hObject, eventdata, handles)
% hObject      handle to min_start (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%           See ISPC and COMPUTER.
if ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes during object creation, after setting all

```

```

properties.
function sec_start_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sec_start (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc    &&    isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes during object creation, after setting all
properties.
function hours_stop_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hours_stop (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc    &&    isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes during object creation, after setting all
properties.
function min_stop_CreateFcn(hObject, eventdata, handles)
% hObject    handle to min_stop (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc    &&    isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');

```



```

end
% --- Executes during object creation, after setting all
properties.
function sec_stop_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sec_stop (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function Menu_File_Callback(hObject, eventdata, handles)
% hObject    handle to Menu_File (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function Menu_instance_2_Callback(hObject, eventdata, handles)
% hObject    handle to Menu_instance_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
EarthGui

% -----
function Menu_About_3_Callback(hObject, eventdata, handles)
% hObject    handle to Menu_About_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% -----
-----
function submenu_application_1_Callback(hObject, eventdata,
handles)
% hObject    handle to submenu_application_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
Message='This application was developed from Stelios Potirakis
& Anaxagoras Fotopoulos. All rights reserved.For any problems
you may have please contact anax.fotopoulos@gmail.com';
Title='About';
h=msgbox(Message,Title);

% --- Executes on selection change in popupmenu7.
function popupmenu7_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
row=get(handles.popupmenu7,'Value');
string=get(handles.popupmenu7,'String');
handles.fsize=str2num(strtrim(string{row}));
% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu7 contents as cell array
%          contents{get(hObject,'Value')} returns selected item
from popupmenu7

% --- Executes during object creation, after setting all
properties.
function popupmenu7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: popupmenu controls usually have a white background on
Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as
text
%         str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as
text
%         str2double(get(hObject,'String')) returns contents of
edit2 as a double

```

```

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles     empty - handles not created until after all
CreateFcns called

```

```

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit3 as
text
%         str2double(get(hObject,'String')) returns contents of
edit3 as a double

```

```

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles     empty - handles not created until after all
CreateFcns called

```

```

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

end

```
function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as
text
%         str2double(get(hObject,'String')) returns contents of
edit4 as a double
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
```

```
% Hint: edit controls usually have a white background on
Windows.
```

```
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function day = datevec2doy(mydate)
```

```
% Takes a date vector and returns the day of year, i.e.
12/31/2005
```

```
% is returned as day 365, day 06/22/2010 is returned as 173,
etc... The
```

```
% function is vectorized. This function needs etime.m (R2009a
and later).
```

```
%
```

```
% USAGES
```

```
% julday = datevec2doy(mydate)
```

```
%
```

```
% INPUT
```

```
% mydate:   Either a 6xN or Nx6 array of date vectors, as
```

```

output by
%           functions like datevec.
%
% OUTPUT
% julday:   An Nx1 array of julian days.
%
% -----
% EXAMPLE
% %Take the current day and add normally distributed random
% days to the
% %date.
%
% tadd      = randn(1,12);
% mydate    = datevec(now)';
% mydate    = repmat(mydate,1,12);
% mydate(2,:) = mydate(2,:) + tadd;
% day       = datevec2doy(mydate);

[M,N] = size(mydate);
ind    = [M,N]==[6,6];

if( nnz(ind) < 1 )

    error('MATLAB:datevec2doy', 'Input is not in date vector
format');

end;

if(ind(1))

    mydate = mydate';

end;

doy      = mydate;
doy(:,2:3) = 1;
doy(:,4:6) = 0;

if(length(mydate) <= 6)

    day      = 1+floor(etime(mydate,doy)./(3600*24));

elseif(length(mydate) > 6)

    A        = mat2cell(doy,ones(size(doy,1),1),6);

```

```

    B      = mat2cell(mydate,ones(size(doy,1),1),6);
    dt     = cellfun(@etime,B,A);
    day    = 1+floor((dt)./(3600*24));

else

    error('Matlab:etime','Not a date format');

end;

return

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
folder_name = uigetdir();
set(handles.text19,'String',[folder_name,'\']);

% --- Executes on button press in checkbox14.
function checkbox14_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox14 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox14

% --- Executes on button press in checkbox15.
function checkbox15_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox15 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox15

% --- Executes on button press in checkbox2.
function checkbox2_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox2 (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of
MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of checkbox2
```

```
% --- Executes on button press in checkbox3.
```

```
function checkbox3_Callback(hObject, eventdata, handles)
```

```
% hObject handle to checkbox3 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of
MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of checkbox3
```

```
% --- Executes on button press in checkbox4.
```

```
function checkbox4_Callback(hObject, eventdata, handles)
```

```
% hObject handle to checkbox4 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of
MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of checkbox4
```

```
% --- Executes on button press in checkbox5.
```

```
function checkbox5_Callback(hObject, eventdata, handles)
```

```
% hObject handle to checkbox5 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of
MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of checkbox5
```

```
% --- Executes on button press in checkbox6.
```

```
function checkbox6_Callback(hObject, eventdata, handles)
```

```
% hObject handle to checkbox6 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of
MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of checkbox6
```



```
% --- Executes on button press in checkbox7.
function checkbox7_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox7
```

```
% --- Executes on button press in checkbox8.
function checkbox8_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox8
```

```
% --- Executes on button press in checkbox9.
function checkbox9_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox9
```

```
% --- Executes on button press in checkbox10.
function checkbox10_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox10
```

```
% --- Executes on button press in checkbox11.
function checkbox11_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox11 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of checkbox11
```

```
% --- Executes on button press in checkbox12.
```

```
function checkbox12_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to checkbox12 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of  
MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of checkbox12
```

```
% --- Executes on button press in checkbox13.
```

```
function checkbox13_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to checkbox13 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of  
MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of checkbox13
```

```
%
```

```
+++++  
+++++
```

```
%                Plot Power Law of Magnitudes
```

```
%
```

```
+++++  
+++++
```

```
% --- Executes on button press in pushbutton10.
```

```
function pushbutton10_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to pushbutton10 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of  
MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
```

```
radio_choice=str2double(get(handles.text39, 'String'));
```

```
%plotgui(handles.EQ_M)
```

```
r_critical =str2double(get(handles.edit16, 'String'));
```

```
Step = str2double(get(handles.edit5, 'String'));
```

```
dt =str2double(get(handles.edit7, 'String'));
```

```
fsize2 = handles.fsize2;
```

```
% the octave flag has been set by default to 0
```

```

% radio_1=get(handles radiobutton2, 'value');
%
% if radio_1==1
%     %b(t) has been choosed
%     set(handles radiobutton3, 'value', 0)
%     radio_choice=1;
% else
%     %log a(t) has been choosed
%     set(handles radiobutton3, 'value', 1)
%     radio_choice=2;
% end
% if radio_2==1
%     %b(t) has been choosed
%     set(handles radiobutton2, 'value', 0)
%     radio_choice=2;
% else
%     %log a(t) has been choosed
%     set(handles radiobutton2, 'value', 1)
%     radio_choice=1;
% end
radio_choice=str2double(get(handles.text38, 'String'));
% radio_choice2=str2double(get(handles.text39, 'String'));
%
test_powerlaw_PAOSwt_m1_ex_Fig3(handles.EQ_M, dt, Step, fsize2, 0
, r_critical, radio_choice2)
radio_choice3=str2double(get(handles.text50, 'String'));
if radio_choice3==2

test_powerlaw_PAOSwt_m1_ex_Fig3(handles.interv, dt, Step, fsize2
, 0, r_critical, radio_choice)
else

test_powerlaw_PAOSwt_m1_ex_Fig3(handles.EQ_M, dt, Step, fsize2, 0
, r_critical, radio_choice)
end

% tt=evalin('base', 'tt');

if radio_choice ==1
    tt=evalin('base', 'tt');
    b=evalin('base', 'b');

tb_not_accepted=evalin('base', 'tb_not_accepted');

b_not_accepted=evalin('base', 'b_not_accepted');

```

```

        tb_accepted=evalin('base','tb_accepted');
        b_accepted=evalin('base','b_accepted');

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('b(t)')];
        eval(evalin('base','send_to_gui_PL'));
else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
        ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('log a(t)')];
        eval(evalin('base','send_to_gui_PL') );

end
rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('r^2(t)')];
eval(evalin('base','send_to_gui_PL2'));

guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');

```

```

evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');

% if radio_choice==2
%
%   log_a=evalin('base','log_a');
%   ta_not_accepted=evalin('base','ta_not_accepted');
%   log_a_accepted=evalin('base','log_a_accepted');
%   ta_accepted=evalin('base','ta_accepted');
%   log_a_accepted=evalin('base','log_a_accepted');
%
%
%
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('log a(t)')];
%   eval(evalin('base','send_to_gui_PL') );
%
% else
%
%   b=evalin('base','b');
%   tb_not_accepted=evalin('base','tb_not_accepted');
%   b_not_accepted=evalin('base','b_not_accepted');
%   tb_accepted=evalin('base','tb_accepted');
%   b_accepted=evalin('base','b_accepted');
%
%
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('b(t)')];
%   eval(evalin('base','send_to_gui_PL'));
% end
%
%
% rr=evalin('base','rr');
%
%
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('r^2(t)')];
% eval(evalin('base','send_to_gui_PL2'));

%assignin('base','radio_choice',radio_choice);

% %=====log a(t)=====

```

```

%% --- Executes on button press in pushbutton23.
% function pushbutton23_Callback(hObject, eventdata, handles)
%% hObject    handle to pushbutton23 (see GCBO)
%% eventdata  reserved - to be defined in a future version of
MATLAB
%% handles     structure with handles and user data (see
GUIDATA)
%
%
% try
%     tt=evalin('base','tt');
%     log_a=evalin('base','log_a');
%     ta_not_accepted=evalin('base','ta_not_accepted');
%     log_a_accepted=evalin('base','log_a_accepted');
%     ta_accepted=evalin('base','ta_accepted');
%     log_a_accepted=evalin('base','log_a_accepted');
%
%
% handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('log a(t)')];
%     eval(evalin('base','send_to_gui_PL') );
%
%     %Clear data
%     %evalin('base','clear tt');
%     evalin('base','clear log_a');
%     evalin('base','clear ta_not_accepted');
%     evalin('base','clear log_a_accepted');
%     evalin('base','clear ta_accepted');
%     evalin('base','clear log_a_accepted');
%
%
% catch exception
%     msg10=msgbox('Please compute before pressing the
button');
% end
% guidata(hObject,handles);
%
% %=====b(t)=====
%% --- Executes on button press in pushbutton26.
% function pushbutton26_Callback(hObject, eventdata, handles)
%% hObject    handle to pushbutton26 (see GCBO)
%% eventdata  reserved - to be defined in a future version of
MATLAB
%% handles     structure with handles and user data (see
GUIDATA)

```

```

% try
%     tt=evalin('base','tt');
%     b=evalin('base','b');
%     tb_not_accepted=evalin('base','tb_not_accepted');
%     b_not_accepted=evalin('base','b_not_accepted');
%     tb_accepted=evalin('base','tb_accepted');
%     b_accepted=evalin('base','b_accepted');
%
%
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('b(t)')];
%     eval(evalin('base','send_to_gui_PL'));
%
%     %clear Data
%     %evalin('base','clear tt');
%     evalin('base','clear b');
%     evalin('base','clear tb_not_accepted');
%     evalin('base','clear b_not_accepted');
%     evalin('base','clear tb_accepted');
%     evalin('base','clear b_accepted');
% catch exception
%     msg10=msgbox('Please compute before pressing the
button');
% end
% guidata(hObject,handles);
%
% %=====r^2=====
% % --- Executes on button press in pushbutton27.
% function pushbutton27_Callback(hObject, eventdata, handles)
% % hObject    handle to pushbutton27 (see GCBO)
% % eventdata  reserved - to be defined in a future version of
MATLAB
% % handles    structure with handles and user data (see
GUIDATA)
% try
% rr=evalin('base','rr');
% tt=evalin('base','tt');
% r_critical=evalin('base','r_critical');
%
%
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('r^2(t)')];
% eval(evalin('base','send_to_gui_PL2'));
% catch exception
%     msg10=msgbox('Please compute before pressing the
button');

```

```

% end
%
% guidata(hObject,handles);
%
% %=====b(t)=====
% % --- Executes on button press in pushbutton28.
% function pushbutton28_Callback(hObject, eventdata, handles)
% % hObject    handle to pushbutton28 (see GCBO)
% % eventdata  reserved - to be defined in a future version of
MATLAB
% % handles    structure with handles and user data (see
GUIDATA)
%
% % b=evalin('base','b');
% % tb_not_accepted=evalin('base','tb_not_accepted');
% % b_not_accepted=evalin('base','b_not_accepted');
% % tb_accepted=evalin('base','tb_accepted');
% % b_accepted=evalin('base','b_accepted');
% %
%
%
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('b(t)')];
% % eval(evalin('base','send_to_gui_PL'));
%
%
% %=====r^2=====
% % --- Executes on button press in pushbutton29.
% function pushbutton29_Callback(hObject, eventdata, handles)
% % hObject    handle to pushbutton29 (see GCBO)
% % eventdata  reserved - to be defined in a future version of
MATLAB
% % handles    structure with handles and user data (see
GUIDATA)
%
% %=====log a(t)=====
% % --- Executes on button press in pushbutton30.
% function pushbutton30_Callback(hObject, eventdata, handles)
% % hObject    handle to pushbutton30 (see GCBO)
% % eventdata  reserved - to be defined in a future version of
MATLAB
% % handles    structure with handles and user data (see
GUIDATA)
% % log_a=evalin('base','log_a');
% % ta_not_accepted=evalin('base','ta_not_accepted');
% % log_a_accepted=evalin('base','log_a_accepted');
% % ta_accepted=evalin('base','ta_accepted');

```



```

%% log_a_accepted=evalin('base','log_a_accepted');
%%
%%
%%
%
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str('log a(t)')];
%% eval(evalin('base','send_to_gui_PL') );

```

```

function myRadio(RadioH,EventData)
handles = guidata(RadioH);
otherRadio = handles.radio(handles.radio ~= RadioH);
set(otherRadio, 'Value', 0);

```

```

function
test_powerlaw_PAOSwt_m1_ex_Fig3(Input_Signal,dt,step>window_s
ize,octave_flag,r_critical,radio_choice)
%
%
% This function is based on the Power Law fitting function
% powerlawPAOSwt_m1.m.
% It is designed to be flexible in use, providing different
possibilities
% of application and can be used either on GNU-Octave or on
% Matlab
% It is property of the ?????? team.
%
% INPUT ARGUMENTS:
%
% Input_Signal = measured time-series [vector of real numbers]
% dt = sampling period [in seconds]
% step = rolling step for the estimation of power law fitting
[in samples]
% window_size = window_size for wavelet transform calculation
[in samples]
% octave_flag = flag to check Octave run. If equals 1 then run
is in Octave
%
% If equals 0 then run
is in MATLAB

Ni = length(Input_Signal); % extract # of data-series points
sp_ty = 'log';

```

```

nr_sc = 25;%25;
sc_sp = 0.25;%0.25;
mother = 'MORLET';
t = (1:Ni)*dt/3600; % construct time vector in hours
plot_flag = 0;

print_shift = 50;
fast_flag = 1;
octave_legend=1; % if 1 plot legend, if 0 do not plot

% if (octave_flag == 0),
%   figure;
%   plot(t,Input_Signal,'-rs','LineWidth',1,...
%       'MarkerEdgeColor','b',...
%       'MarkerFaceColor','b',...
%       'MarkerSize',2);axis tight;
%   xlabel('Time (h)');
%   ylabel('Signal Ampl.(t)');
%   title('Input Signal Variation with Time');
%   print -f1 -dtiff 'Figure1'
% end

%
%           [f,log_a,b,rr,tt]           =
powerlawPAOSwt_m1(Input_Signal,dt,step,...
%
window_size,sp_ty,nr_sc,sc_sp,mother,r_critical,plot_flag,...
%   print_shift,octave_flag,fast_flag);

[f,log_a,b,rr,tt] = powerlawPAOS(Input_Signal,dt,step,...
window_size,sp_ty,nr_sc,sc_sp,mother,r_critical,plot_flag,...
    print_shift,octave_flag,fast_flag);

% disp([sprintf(['If you wish to construct fig 103 of
discriminated \n'...
% 'by r_critical parameters, please run OctaveFig3.exe,\n'...
% 'or MatlabFig3.exe'])]))

%% Plot Power Law

```

```

%Initialization

load powerlawPAOSwt_m1_temp f log_a b rr tt window_size step y
r_critical

Input_Signal = y;
yy = Input_Signal(window_size+step:step:end);
clear y

N_accepted=0;
test_ind = 1;

ta_accepted=zeros(1,length(tt));
ta_not_accepted=zeros(1, length(tt));
tb_accepted=zeros(1, length(tt));
tb_not_accepted=zeros(1, length(tt));
log_a_accepted=zeros(1, length(tt));
log_a_not_accepted=zeros(1, length(tt));
b_accepted=zeros(1, length(tt));
b_not_accepted=zeros(1, length(tt));

log_a_accepted_for_printout=0;
b_accepted_for_printout=0;
r_sq_accepted_for_printout=0;

%% Find critical values

for i=1:length(tt)
    ta_accepted(i)=tt(i);
    ta_not_accepted(i)=tt(i);
    tb_accepted(i)=tt(i);
    tb_not_accepted(i)=tt(i);

    log_a_accepted(i)=-100; %arbitrary low values to cut-off
the plot
    log_a_not_accepted(i)=-100;
    b_accepted(i)=100; %arbitrary high values to cut-off
the plot
    b_not_accepted(i)=100;
    if (rr(i) >= r_critical)
        log_a_accepted(i)=log_a(i);
        b_accepted(i)=b(i);
        N_accepted=N_accepted+1;
        log_a_accepted_for_printout(N_accepted)=log_a(i);
        b_accepted_for_printout(N_accepted)=b(i);
    end
end

```

```

        r_sq_accepted_for_printout(N_accepted)=rr(i);
    else
        log_a_not_accepted(i)=log_a(i);
        b_not_accepted(i)=b(i);
    end
end

%% Plots

assignin('base','tt',tt);

if radio_choice==2
    %plot log a(t)
    % figure;
    % subplot(3,1,1);
    %
    plot(ta_not_accepted,log_a_not_accepted,'ro','markersize',4);
    % hold on
    % plot(ta_accepted,log_a_accepted,'bo','markersize',4);
    % set(gca,'xlim',[tt(1) tt(end)]);
    % set(gca,'ylim',[min(log_a) max(log_a)]);
    % ylabel('log a(t)')
    % grid;
    % hold off

    assignin('base','log_a',log_a);
    assignin('base','ta_not_accepted',ta_not_accepted);
    assignin('base','log_a_accepted',log_a_accepted);
    assignin('base','ta_accepted',ta_accepted);
    assignin('base','log_a_accepted',log_a_accepted);
    %handles.plot_dropdown_names=[handles.plot_dropdown_names;cellstr('log a(t)')];
    %guiplot(handles.plot_dropdown_names,ta_not_accepted,log_a_accepted,ta_accepted,log_a_accepted);

    %assignin('base','send_to_gui_PL','guiplot(handles.plot_dropdown_names,ta_not_accepted,log_a_accepted,ta_accepted,log_a_accepted);');

assignin('base','send_to_gui_PL','guiplot(handles.plot_dropdown_names,ta_not_accepted,log_a_accepted,ta_accepted,log_a_accepted,tt,log_a,2);');
else
    %plot b(t)

```

```

% figure;
% subplot(3,1,1);
%
plot(tb_not_accepted,b_not_accepted,'ro','MarkerSize',4);
% hold on;
% plot(tb_accepted,b_accepted,'bo','MarkerSize',4);
% set(gca,'xlim',[tt(1) tt(end)]);
% set(gca,'ylim',[min(b) 0]);
% ylabel('b(t)')
% grid;
% hold off

assignin('base','b',b);
assignin('base','tb_not_accepted',tb_not_accepted);
assignin('base','b_not_accepted',b_not_accepted);
assignin('base','tb_accepted',tb_accepted);
assignin('base','b_accepted',b_accepted);
%handles.plot_dropdown_names=[handles.plot_dropdown_names;cellstr('b(t)')];
%guiplot(handles.plot_dropdown_names,tb_not_accepted,b_not_accepted,tb_accepted,b_accepted);

%assignin('base','send_to_gui_PL','guiplot(handles.plot_dropdown_names,tb_not_accepted,b_not_accepted,tb_accepted,b_accepted);');

assignin('base','send_to_gui_PL','guiplot(handles.plot_dropdown_names,tb_not_accepted,b_not_accepted,tb_accepted,b_accepted,tt,b,1);');
end

%plot r^2
% subplot(3,1,2);
% plot (tt,rr,'r');
% set(gca,'xlim',[tt(1) tt(end)]);
% set(gca,'ylim',[r_critical 1.01]);
% ylabel('r^2(t)')
% grid;

assignin('base','r_critical',r_critical);

assignin('base','rr',rr);
%handles.plot_dropdown_names=[handles.plot_dropdown_names;cellstr('r^2(t)')];
%guiplot(handles.plot_dropdown_names,tt,rr);
%assignin('base','send_to_gui_PL2',

```

```
'guiplot(handles.plot_dropdown_names,tt,rr);');
assignin('base','send_to_gui_PL2',
'guiplot(handles.plot_dropdown_names,tt,rr,r_critical,0,0,0,3
);');
```

```
% %plot magnitudes
% subplot(3,1,3);
% plot (tt,yy,'b');
% set(gca,'xlim',[tt(1) tt(end)]);
% xlabel('Time (h)');
% ylabel('Amplitude')
% grid;
% axis tight;
```

```
% %% SAVE
% hgsave('PowerLaw')
% saveas(gcf,'PowerLaw.fig')
% print -dtiff 'PowerLaw'
% save plot_Power_Law_Workspace
```

```
%
%
% function [f,log_a,b,rr,tt,yy,wave_j,lin_fit] =
powerlawPAOSwt_m1(y,dt,step,...
%
window_size,sp_ty,nr_sc,sc_sp,mother,r_critical,plot_flag,...
% print_shift,octave_flag,fast_flag)
%
% %function [f,log_a,b,rr,tt,yy,wave_j,lin_fit] =
powerlawPAOSwt_m1(y,dt,...
% % step>window_size,sp_ty,nr_sc,sc_sp,mother,r_critical,...
% % plot_flag,print_shift,octave_flag,fast_flag)
% %
% % Power Law fitting function, using the wavelet.m function
found at
% % http://paos.colorado.edu/research/wavelets/ for wavelet
transform.
% % It is designed to be flexible in use, providing different
possibilities
% % of application and can be used either on GNU-Octave or on
```

```

%% Matlab
%% It is property of the ????? team.
%%
%% INPUT ARGUMENTS:
%%
%% y = measured time-series [vector of real numbers]
%% dt = sampling period [in seconds]
%% step = rolling step for the estimation of power law fitting
[in samples]
%% window_size = window_size for wavelet transform calculation
[in samples]
%% sp_ty = spacing type, 'log', or 'linear'
%% nr_sc = number of scales, default = 25
%% sc_sp = scale spacing, default = 0.25 for 'log' spacing,
use integer for
%%         linear spacing
%% mother = wavelet name, use any of the wave_bases.m valid
wavelet names,
%%         like 'MORLET' or 'PAUL' or 'DOG'
%% plot_flag = flag to check plotting inside the function.
%%         If = 0 do not plot
%%         If >1 plot (plot_flag) number of figures (if
available)
%% print_shift = how many iterations to shift before checking
to print %
%% octave_flag = flag to check Octave run. If equals 1 then
run is in Octave
%%         If equals 0 then
run is in MATLAB
%% fast_flag = flag to indicate fast version of the routine,
where memory
%%         consuming computations (wave_j, lin_fit and
yy) are ommited
%%
%% OUTPUT ARGUMENTS:
%%
%% f = frequencies corresponding to central frequencies of
scales
%% log_a = vector of the log10 of factor "a" of the fitted
power law
%%         "S(f)= a * f^b" through the time-series scanning
with "step"
%%         and "window_size"
%% b = vector of the exponents "b" of the fitted power law
"S(f)= a * f^b"
%%         through the time-series scanning with "step" and

```

```

"window_size"
%% rr = square (r^2) of the Spearman correlation coefficients
(showing the
%% quality of the linear -on log-log scale- curve fitting)
%% tt = table of all time vectors corresponding to the
processed
%% time-series windows [hours]
%% yy = table of all signal vectors corresponding to the
processed
%% time-series windows
%% wave_j = table of the wavelet transform coefficients for
all iterations
%% lin_fit = linear fit parameters P for each iteration
(contains log_a and
%% b, but useful for applying "polyfit.m")
%
%
% if (nargin < 13), fast_flag = -1; end
% if (nargin < 12), octave_flag = -1; end
% if (nargin < 11), print_shift = -1; end
% if (nargin < 10), plot_flag = -1; end
% if (nargin < 9), r_critical = -1; end
% if (nargin < 8), mother = -1; end
% if (nargin < 7), sc_sp = -1; end
% if (nargin < 6), nr_sc = -1; end
% if (nargin < 5), sp_ty = 'log'; end
% if (nargin < 4), window_size = -1; end
% if (nargin < 3), step = -1; end
% if (nargin < 2)
% error('At least an input vector y and sampling time dt
MUST be provided')
% end
%
% if (fast_flag == -1), fast_flag = 1; end
% if (octave_flag == -1), octave_flag = 1; end
% if (print_shift == -1), print_shift = 50; end
% if (plot_flag == -1), plot_flag = 0; end
% if (r_critical == -1), r_critical = 0.95; end
% if (mother == -1), mother = 'MORLET'; end
% if (sc_sp == -1), sc_sp = 0.25; end
% if (nr_sc == -1), nr_sc = 25; end
% if (window_size == -1), window_size = 128; end
% if (step == -1), step = 1; end
%
% pad = 0;
% segment_size=length(y);

```



```

% t = [1:segment_size];
% iters = (segment_size-window_size)/step;
% iterations = iters
% range = 60;
%
% N_of_passes = 0;
% fs=1/dt;
% if (octave_flag == 0)
%     wvname = 'morl';
% end
% colormap = pink; %jet;
% %t = (1:Nr)*dt/3600; % construct time vector in hours
% range = 50;
% bw = 10;
%
% tic;
%
% for j=1:iters
%
%
%
%     if (fast_flag == 0)
%
%         yy(j,:) = y((j-1)*step+[1:window_size]);
%                                     [WAVE,PERIOD,SCALE,COI] =
wavelet2(yy(j,:),dt,pad,sc_sp,2*dt,nr_sc-1,mother,sp_ty);
%         wave_j(j,,:) = WAVE;
%         LOG_Power_Spectrum=log10((sum(abs(WAVE').^2)))';
%         f = 1./PERIOD;
%         log_f=log10(1./PERIOD');
%         [P]=polyfit(log_f,LOG_Power_Spectrum,1);
%         lin_fit(j,:) = P;
%
%     else
%
%                                     [WAVE,PERIOD,SCALE,COI] = wavelet2(y((j-
1)*step+[1:window_size]),dt,pad,sc_sp,2*dt,nr_sc-
1,mother,sp_ty);
%         LOG_Power_Spectrum=log10((sum(abs(WAVE').^2)))';
% %         f = 1./PERIOD(1:end-0);
%                                     f = 1./PERIOD(1:end-
8);
%                                     %Rn
% %         log_f=log10(1./PERIOD(1:end-0)');
%                                     log_f=log10(1./PERIOD(1:end-
8)');
%                                     %Rn
% %         [P]=polyfit(log_f,LOG_Power_Spectrum(1:end-0),1);

```

```

%           [P]=polyfit(log_f,LOG_Power_Spectrum(1:end-
8),1);           %Rn
%
%           papap=1;
%
%       end
%
%       if (octave_flag == 1)
%
%           r=spearman(log_f,LOG_Power_Spectrum(1:end-0));
%           r=spearman(log_f,LOG_Power_Spectrum(1:end-
8));           %Rn
%       else
%
%           r=corr(log_f,LOG_Power_Spectrum(1:end-
0),'type','Spearman');
%           r=corr(log_f,LOG_Power_Spectrum(1:end-
8),'type','Spearman');           %Rn
%       end
%
%       b(j) = P(1);
%       log_a(j) = P(2);
%       rr(j) = r.^2;
%       tt(j)=t(window_size+j*step)*dt/3600;
%
%
%       %%
%       figure
%       fit = polyval(P,log_f);
%       plot(log_f,LOG_Power_Spectrum(1:end-25),'r*','...
log_f,fit,'b-');
%       xlabel('Log(Frequency)');
%       ylabel('Log(S(f))');
%       grid;
%
%       %%
%
%       %       construct (plot_flag) WAVE vectors for plotting
purposes
%       if (plot_flag >= 1 && N_of_passes <= plot_flag &&
j>1+print_shift)
%           if (abs(b(j-print_shift)) > 1 && rr(j-
print_shift)>=r_critical)
%               N_of_passes=N_of_passes+1;

```

```

%           initial_j(N_of_passes)=j;
%           polynomial_in_window(N_of_passes,:)=P;
%           beta_in_window(N_of_passes,:)=b(j);
%           log_a_in_window(N_of_passes,:)=log_a(j);
%           r_square_in_window(N_of_passes,:)=rr(j);
%
log_ps_in_window(N_of_passes,:)=LOG_Power_Spectrum;
%           log_f_in_window(N_of_passes,:)=log_f;
%           wavelet_transform(N_of_passes,,:)= WAVE;
%       end
%   end
%   %memory
%
% end
% % plot (plot_iflag) examples of powerlaw fit and scalograms
if plot_flag >=1
%
% if (plot_flag >= 1 && N_of_passes ~= 0)
%
%     for i=1:plot_flag
%         % for plotting purpose
%         for k = 1:N_of_passes
%             wave_tran = squeeze(wavelet_transform(k,,:));
%         end
%         for l = 1:window_size
%             t_in_window(l)=t(initial_j(N_of_passes)+(l-
1))*dt/3600; % in hours
%             y_in_window(l)=y(initial_j(N_of_passes)+(l-1));
%         end
%         % end for plotting purpose
%
%         figure(i+1000);
%         subplot(3,1,1);
%
%             %           fit           =
polyval(polynomial_in_window(i,:),log_f_in_window(i,:));
%
%           plot(log_f_in_window(i,:),log_ps_in_window(i,:), 'r*',...
% %           log_f_in_window(i,:),fit, 'b-');
%
%           %           fit           =
polyval(polynomial_in_window(i,:),log_f_in_window(i,:));
%           Rn
%           %           fit           =
plot(log_f_in_window(i,:),log_ps_in_window(i,:), 'r*',...
%           % Rn
log_f_in_window(i,:),fit, 'b-

```

```

');
Rn
%
%       xlabel('Log(Frequency)');
%       ylabel('Log(S(f))');
%       grid;
%       if (octave_flag == 0)% Create textbox in Matlab
%           annotation('textbox',[0.6635 0.7857 0.2044
0.09762],...
%                               'String',{'log(a)=',
num2str(log_a_in_window(i,:),3),...
%
sprintf('\n'),'b=',num2str(abs(beta_in_window(i,:)),3),...
%           '\pm ',num2str(std(beta_in_window),2),...
%
sprintf('\n'),'r^2=',num2str(r_square_in_window(i,:),2),...
%           sprintf('\n'),'t=',num2str(t_in_window(1),2),'
- ',...
%                               num2str(t_in_window(end),2),
'h']},'FitBoxToText','on');
%       else % Create legent in Octave
%           legend(strcat('log_a=',
num2str(log_a_in_window(i,:)),...
%           ', b=',num2str(beta_in_window(i,:)),...
%           ', r^2=',num2str(r_square_in_window(i,:))));
%       end
%       title(strcat('Power Law Fit and Scalogram,
',num2str(window_size),...
%           ' Window size, ', num2str(step),' Step'));
%
%       subplot(3,1,2);
%       plot (t_in_window,y_in_window,'r');
%       set(gca,'xlim',[t_in_window(1) t_in_window(end)]);
%       xlabel('Time (h)');
%       ylabel('Amplitude')
%       grid;
%
%       S=wave_tran.*wave_tran;
%       maxS = max(max(abs(S))); minS = 10^-8*maxS;
%       dBS = 10*log10(abs(S));
%       %dBS = 20*log10(max(abs(S), minS));
%       maxdBS = max(max(dBS));
%       mindBS = min(min(dBS));
%       %dBS = min(maxdBS, max(maxdBS-range, dBS));
%       %dBS = max(mindBS, min(mindBS+range, dBS));
%

```

```

% subplot(3,1,3)
% yscale='log';
% mesh(t_in_window,f,dBS);
% set(gca,'YScale',yscale);
% colormap(colormap);
% ylim([min(f) max(f)]);
% ylabel('f(Hz)');
% xlabel('Time (h)');
% set(gca,'TickDir','out')
% view(2);
% axis tight;
%
% if (octave_flag == 0)
%     title(['Signal ' wvname ' Scalogram (',...
%           num2str(range),'dB, ',yscale,' scale, bw
parameter=',...
%           num2str(bw),')',sprintf('\n'),...
%           'dB of Absolute Value of Ca,b coefficients']);
% end
%
% set(gca,'TickDir','out')
% axis tight;
%
% if (octave_flag == 1),
%     colorbar('EastOutside');
%     print(num2str(i+1000),'-dgif')
% else
%     colorbar([0.93 0.11 0.02 0.7742],'FontSize',8);
%     figure_name_1000i = ['Figure',num2str(1000+i)];
%     print('-dtiff', figure_name_1000i);
% end
%
% figure(i+2000);
% yscale='log';
% mesh(t_in_window,f,dBS);
% set(gca,'YScale',yscale);
% colormap=pink;
% colormap(colormap);
% ylim([min(f) fs/2]);
% ylabel('f(Hz)');
% xlabel('Time (h)');
% set(gca,'TickDir','out')
% axis tight;
% title ('3D Wavelet Tranform');
%
% if (octave_flag == 1),

```

```

%             print(num2str(i+2000),'-dgif')
%         else
%             figure_name_2000i = ['Figure',num2str(2000+i)];
%             print('-dtiff', figure_name_2000i);
%
%             figure(i+3000)
%
%
%                                     COEFS      =
cwtext(y_in_window,(1:1:25),'morl','PlotMode','glbabs');
%             figure_name_3000i = ['Figure',num2str(3000+i)];
%             print('-dtiff', figure_name_3000i);
%         end
%     end
% end
%
% save powerlawPAOSwt_m1_temp

```

```

function [daughter,fourier_factor,coi,dofmin] = ...
    wave_bases(mother,k,scale,param)
%WAVE_BASES 1D Wavelet functions Morlet, Paul, or DOG
%
% [DAUGHTER,FOURIER_FACTOR,COI,DOFMIN] = ...
%     wave_bases(MOTHER,K,SCALE,PARAM);
%
% Computes the wavelet function as a function of Fourier
frequency,
% used for the wavelet transform in Fourier space.
% (This program is called automatically by WAVELET)
%
% INPUTS:
%
% MOTHER = a string, equal to 'MORLET' or 'PAUL' or 'DOG'
% K = a vector, the Fourier frequencies at which to calculate
the wavelet
% SCALE = a number, the wavelet scale
% PARAM = the nondimensional parameter for the wavelet
function
%
% OUTPUTS:
%
% DAUGHTER = a vector, the wavelet function
% FOURIER_FACTOR = the ratio of Fourier period to scale
% COI = a number, the cone-of-influence size at the scale
% DOFMIN = a number, degrees of freedom for each point in

```

```

the wavelet power
%           (either 2 for Morlet and Paul, or 1 for the DOG)
%
%-----
%
% Copyright (C) 1995-1998, Christopher Torrence and Gilbert
% P. Compo
% University of Colorado, Program in Atmospheric and Oceanic
% Sciences.
% This software may be used, copied, or redistributed as long
% as it is not
% sold and this copyright notice is reproduced on each copy
% made. This
% routine is provided as is without any express or implied
% warranties
% whatsoever.
%-----
%
mother = upper(mother);
n = length(k);

if (strcmp(mother,'MORLET')) %-----
---- Morlet
    if (param == -1), param = 6.;; end
    k0 = param;
    expnt = -(scale.*k - k0).^2/2.*(k > 0.);
    norm = sqrt(scale*k(2))*(pi^(-0.25))*sqrt(n); % total
energy=N [Eqn(7)]
    daughter = norm*exp(expnt);
    daughter = daughter.*(k > 0.); % Heaviside step
function
    fourier_factor = (4*pi)/(k0 + sqrt(2 + k0^2)); % Scale-
->Fourier [Sec.3h]
    coi = fourier_factor/sqrt(2); % Cone-of-
influence [Sec.3g]
    dofmin = 2; % Degrees
of freedom
elseif (strcmp(mother,'PAUL')) %-----
---- Paul
    if (param == -1), param = 4.;; end
    m = param;
    expnt = -(scale.*k).*(k > 0.);
    norm = sqrt(scale*k(2))*(2^m/sqrt(m*prod(2:(2*m-
1))))*sqrt(n);
    daughter = norm*((scale.*k).^m).*exp(expnt);
    daughter = daughter.*(k > 0.); % Heaviside step

```

```

function
    fourier_factor = 4*pi/(2*m+1);
    coi = fourier_factor*sqrt(2);
    dofmin = 2;
elseif (strcmp(mother,'DOG')) %-----
--- DOG
    if (param == -1), param = 2.;; end
    m = param;
    expnt = -(scale.*k).^2 ./ 2.0;
    norm = sqrt(scale*k(2)/gamma(m+0.5))*sqrt(n);
    daughter = -norm*(i^m)*((scale.*k).^m).*exp(expnt);
    fourier_factor = 2*pi*sqrt(2./(2*m+1));
    coi = fourier_factor/sqrt(2);
    dofmin = 1;
else
    error('Mother must be one of MORLET,PAUL,DOG')
end

return

```

```

function [wave,period,scale,coi] = ...
    wavelet2(Y,dt,pad,dj,s0,J1,mother,sp_ty,param)

%WAVELET 1D Wavelet transform with optional singificance
testing
%
% [WAVE,PERIOD,SCALE,COI] =
wavelet2(Y,DT,PAD,DJ,S0,J1,mother,sp_ty,param)
%
%
% Custom version of wavelet.m that permits either 'log' or
'linear' scales
% spacing.
%
% In case of "inear" scales spacing, it is proposed to use
DJ=1,
% S0=irrelevant % (e.g. = 1) and J1 = number of scales
%
%
% Computes the wavelet transform of the vector Y (length N),
% with sampling rate DT.
%
% By default, the Morlet wavelet (k0=6) is used.
% The wavelet basis is normalized to have total energy=1 at

```



```

all scales.
%
%
% INPUTS:
%
%   Y = the time series of length N.
%   DT = amount of time between each Y value, i.e. the sampling
time.
%
% OUTPUTS:
%
%   WAVE is the WAVELET transform of Y. This is a complex
array
%   of dimensions (N,J1+1). FLOAT(WAVE) gives the WAVELET
amplitude,
%   ATAN(IMAGINARY(WAVE),FLOAT(WAVE)) gives the WAVELET phase.
%   The WAVELET power spectrum is ABS(WAVE)^2.
%   Its units are sigma^2 (the time series variance).
%
%
% OPTIONAL INPUTS:
%
% *** Note *** setting any of the following to -1 will cause
the default
%           value to be used.
%
%   PAD = if set to 1 (default is 0), pad time series with
enough zeroes to get
%           N up to the next higher power of 2. This prevents
wraparound
%           from the end of the time series to the beginning,
and also
%           speeds up the FFT's used to do the wavelet transform.
%           This will not eliminate all edge effects (see COI
below).
%
%   DJ = the spacing between discrete scales. Default is 0.25.
%           A smaller # will give better scale resolution, but
be slower to plot.
%
%   S0 = the smallest scale of the wavelet. Default is 2*DT.
%
%   J1 = the # of scales minus one. Scales range from S0 up
to S0*2^(J1*DJ),
%           to give a total of (J1+1) scales. Default is J1 =
(LOG2(N DT/S0))/DJ.

```

```

%
%
% OPTIONAL OUTPUTS:
%
%   PERIOD = the vector of "Fourier" periods (in time units)
%   that corresponds
%           to the SCALES.
%
%   SCALE = the vector of scale indices, given by  $S_0 * 2^{(j * DJ)}$ ,
%   j=0...J1
%           where J1+1 is the total # of scales.
%
%   COI = if specified, then return the Cone-of-Influence,
%   which is a vector
%         of N points that contains the maximum period of useful
%   information
%         at that particular time.
%         Periods greater than this are subject to edge effects.
%         This can be used to plot COI lines on a contour plot
%   by doing:
%           IDL> CONTOUR,wavelet,time,period
%           IDL> PLOTS,time,coi,NOCLIP=0
%
%-----
%
% Copyright (C) 1995-1998, Christopher Torrence and Gilbert
% P. Compo
% University of Colorado, Program in Atmospheric and Oceanic
% Sciences.
% This software may be used, copied, or redistributed as long
% as it is not
% sold and this copyright notice is reproduced on each copy
% made. This
% routine is provided as is without any express or implied
% warranties
% whatsoever.
%
% Notice: Please acknowledge the use of this program in any
% publications:
%   ``Wavelet software was provided by C. Torrence and G.
%   Compo,
%           and is available at URL:
%   http://paos.colorado.edu/research/wavelets/''
%
% Reference: Torrence, C. and G. P. Compo, 1998: A Practical

```

```

Guide to
%           Wavelet Analysis. <I>Bull. Amer. Meteor. Soc.</I>,
79, 61-78.
%
% Please send a copy of such publications to either C. Torrence
or G. Compo:
% Dr. Christopher Torrence           Dr. Gilbert P. Compo
% Advanced Study Program             NOAA/CIRES Climate
Diagnostics Center
% National Center for Atmos. Research   Campus Box 449
% P.O. Box 3000                       University of Colorado
at Boulder
% Boulder CO 80307--3000, USA.         Boulder CO 80309-
0449, USA.
% E-mail: torrence@ucar.edu           E-mail:
gpc@cdc.noaa.gov
%-----
-----

```

```

if (nargin < 9), param = -1;, end
if (nargin < 8), sp_ty = -1;, end
if (nargin < 7), mother = -1;, end
if (nargin < 6), J1 = -1;, end
if (nargin < 5), s0 = -1;, end
if (nargin < 4), dj = -1;, end
if (nargin < 3), pad = 0;, end
if (nargin < 2)
    error('Must input a vector Y and sampling time DT')
end

n1 = length(Y);

if (s0 == -1), s0=2*dt;, end
if (dj == -1), dj = 1./4.;;, end
if (J1 == -1), J1=fix((log(n1*dt/s0)/log(2))/dj);, end
if (mother == -1), mother = 'MORLET';, end
if (mother == -1), sp_ty = 'log';, end

%...construct time series to analyze, pad if necessary
x(1:n1) = Y - mean(Y);
if (pad == 1)
    base2 = fix(log(n1)/log(2) + 0.4999); % power of 2
nearest to N
    x = [x,zeros(1,2^(base2+1)-n1)];

```

```

end
n = length(x);

%...construct wavenumber array used in transform [Eqn(5)]
k = [1:fix(n/2)];
k = k.*((2.*pi)/(n*dt));
k = [0., k, -k(fix((n-1)/2):-1:1)];

%...compute FFT of the (padded) time series
f = fft(x); % [Eqn(3)]

%...construct SCALE array & empty PERIOD & WAVE arrays

if strcmp(sp_ty,'log')==1
    scale = s0*2.^((0:J1)*dj);
else
    scale = (1:dj:J1+1)*dt;
end

period = scale;
wave = zeros(J1+1,n); % define the wavelet array
wave = wave + i*wave; % make it complex

% loop through all scales and compute transform
for a1 = 1:J1+1
    [daughter,fourier_factor,coi,dofmin]=wave_bases(mother,k
, scale(a1),param);
    wave(a1,:) = ifft(f.*daughter); % wavelet
transform[Eqn(4)]
end

period = fourier_factor*scale;
coi = coi*dt*[1E-5,1:((n1+1)/2-1),fliplr((1:(n1/2-1))),1E-
5]; % COI [Sec.3g]
wave = wave(:,1:n1); % get rid of padding before returning

return

%
++++
++++
%           End of Plot Power Law of Magnitudes
%

```

```
+++++
+++++
```

```
% --- Executes on selection change in popupmenu8.
function popupmenu8_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
popupmenu8 contents as cell array
%         contents{get(hObject,'Value')} returns selected item
from popupmenu8
row=get(handles.popupmenu8,'Value');
string=get(handles.popupmenu8,'String');
handles.fsize2=str2num(strtrim(string{row}));
```

```
% --- Executes during object creation, after setting all
properties.
function popupmenu8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called
```

```
% Hint: popupmenu controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```

% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as
text
%          str2double(get(hObject,'String')) returns contents of
edit5 as a double

% --- Executes during object creation, after setting all
properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as
text
%          str2double(get(hObject,'String')) returns contents of
edit7 as a double

% --- Executes during object creation, after setting all
properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all

```

CreateFcns called

```
% Hint: edit controls usually have a white background on  
Windows.
```

```
% See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
% --- Executes on button press in pushbutton12.
```

```
function pushbutton12_Callback(hObject, eventdata, handles)  
% hObject handle to pushbutton12 (see GCBO)  
% eventdata reserved - to be defined in a future version of  
MATLAB  
% handles structure with handles and user data (see GUIDATA)  
bksls = ['\'];  
tmp1=datevec(get(handles.text3,'String'));  
tmp2=datevec(get(handles.text4,'String'));
```

```
startdate=datenum(sprintf([num2str(tmp1(2)),'-'  
,num2str(tmp1(3)),'-',num2str(tmp1(1))]]));  
stopdate=datenum(sprintf([num2str(tmp2(2)),'-'  
,num2str(tmp2(3)),'-',num2str(tmp2(1))]]));
```

```
assignin('base','startdate',startdate);  
assignin('base','stopdate',stopdate);
```

```
data_dir = get(handles.text19,'String');  
year1 =tmp1(1);  
data_dir = [data_dir,num2str(year1)];  
station_code='';  
%Kalamata  
if ( get(handles.checkbox2,'Value') == 1 )  
    %station_code=[station_code,'A'];  
    station_code=[station_code,'0'];  
end;  
%Kefalonia  
if ( get(handles.checkbox3,'Value') == 1 )  
    %station_code=[station_code,'B'];  
    station_code=[station_code,'F'];  
end;  
%Ioannina
```

```

if ( get(handles.checkbox4,'Value') == 1 )
    %station_code=[station_code,'C'];
    station_code=[station_code,'J'];
end;
%Kozani
if ( get(handles.checkbox5,'Value') == 1 )
    %station_code=[station_code,'D'];
    station_code=[station_code,'K'];
end;
%Komotini
if ( get(handles.checkbox6,'Value') == 1 )
    %station_code=[station_code,'E'];
    station_code=[station_code,'T'];
end;
%Mytilene
if ( get(handles.checkbox7,'Value') == 1 )
    %station_code=[station_code,'F'];
    station_code=[station_code,'M'];
end;
%Rhode
if ( get(handles.checkbox8,'Value') == 1 )
    %station_code=[station_code,'G'];
    station_code=[station_code,'A'];
end;
%Neapolis
if ( get(handles.checkbox9,'Value') == 1 )
    %station_code=[station_code,'H'];
    station_code=[station_code,'E'];
end;
%Vamos
if ( get(handles.checkbox10,'Value') == 1 )
    %station_code=[station_code,'J'];
    station_code=[station_code,'V'];
end;
%Corfu
if ( get(handles.checkbox11,'Value') == 1 )
    %station_code=[station_code,'L'];
    station_code=[station_code,'P'];
end;
%Zante
if ( get(handles.checkbox12,'Value') == 1 )
    station_code=[station_code,'Z'];
end;
%Atalanti
if ( get(handles.checkbox13,'Value') == 1 )
    %station_code=[station_code,'N'];

```



```

        station_code=[station_code,'H'];
    end;
    %New Zante
    if ( get(handles.checkbox30,'Value') == 1 )
        station_code=[station_code,'N'];
    end;

    days = [datevec2doy(tmp1):1:datevec2doy(tmp2)];
    % Filter Data Checkbox
    data_filt = get(handles.checkbox14,'Value');
    % if data_filt==1
    %     set(handles.checkbox23,'Enable','on');
    %     set(handles.checkbox24,'Enable','on');
    %     set(handles.checkbox25,'Enable','on');
    %     set(handles.checkbox26,'Enable','on');
    %     set(handles.checkbox27,'Enable','on');
    %     set(handles.checkbox28,'Enable','on');
    %     set(handles.checkbox29,'Enable','on');
    % end

    % % Use EQ Data Checkbox
    % EQ_swt = get(handles.checkbox15,'Value');
    % if EQ_swt ==1
    %     EQ_file = get(handles.text1,'String');
    % end

    guidata(hObject,handles);
    %=====
    %     construction
    %=====

%% Acquire data from .dat files

for j=1:length(station_code)
    data_dir2 = [data_dir,'\',station_code(j)];
    data_name = ['RF_',station_code(j)];
    init_str = [data_name,'= [];'];
    eval(init_str);

    for i=days(1):1:days(end)

```

```

        if i<10
            try
                vname_i =
[station_code(j),num2str(year1),'00',num2str(i)];
                fname_i = [vname_i,'.dat'];
                %load_str = ['load ' fname_i,']; temp = ', vname_i,
'; clear ',vname_i];
                %eval(load_str);
                importfile_em([data_dir2,'\',fname_i]);
                %load_str = ['temp = ', vname_i, '; clear
',vname_i];
                %eval(load_str);
                data_str =
[data_name,[' ',data_name, ';vname_i'];'];
                eval(data_str);
                %clear temp
                clc
                disp([station_code(j), sprintf('\n')])
                % memory
                perc_done = floor(100*(i-days(1))/(days(end)-
days(1)));
                disp([sprintf('%3.0f', perc_done),'% of reading
done'])
            catch exception
                end
        elseif i<100 && i>=10
            try
                vname_i =
[station_code(j),num2str(year1),'0',num2str(i)];
                fname_i = [vname_i,'.dat'];
                %load_str = ['load ' fname_i,']; temp = ', vname_i,
'; clear ',vname_i];
                %eval(load_str);
                importfile_em([data_dir2,'\', fname_i]);

                %h=evalin('base',vname_i);

                %load_str = ['temp = ', vname_i, '; clear
',vname_i];
                %eval(load_str);
                data_str =
[data_name,[' ',data_name, ';vname_i'];'];
                eval(data_str);
                %clear temp
                clc

```

```

disp([station_code(j), sprintf('\n')])
%      memory
perc_done = floor(100*(i-days(1))/(days(end)-
days(1)));
disp([sprintf('%3.0f', perc_done), '% of reading
done'])
catch exception

end
else
try
vname_i =
[station_code(j), num2str(year1), num2str(i)];
fname_i = [vname_i, '.dat'];
%load_str = ['load ' fname_i, '; temp = ', vname_i,
'; clear ', vname_i];
%eval(load_str);
importfile_em([data_dir2, '\', fname_i]);
%load_str = ['temp = ', vname_i, '; clear
', vname_i];
%eval(load_str);
data_str =
[data_name, '=[', data_name, '; vname_i];'];
eval(data_str);
clear temp
disp([station_code(j), sprintf('\n')])
%      memory
perc_done = floor(100*(i-days(1))/(days(end)-
days(1)));
disp([sprintf('%3.0f', perc_done), '% of data
reading done, for station', num2str(j), '
', num2str(length(station_code))])
catch exception

end

end

end
% %%
% for i=1:length(RF(1,:))
%     figure;
%     plot(RF(:,i));
% end
% %%

end
clear j

```

```

%% Construct Signal Vectors

chan_nr = 7;
chan_list='';
try
    %data_name_file2 =
    ['RF_data_',station_code,'_',num2str(year1),'_d',num2str(days
(1)),'_d',num2str(days(end)),'_signal_vect'];
    for j=1:length(station_code)

        clear data_name
        data_name = ['RF_',station_code(j)];
        clear tmp
        tmp=eval(data_name);

        for i =1:length(tmp(:,1))%find the number of lines
(days) in RF_
            clear cur_file_data
            cur_file_data = evalin('base',tmp(i,:));

            for jj=1:chan_nr
                tic;

                meas_name = [data_name,'_ch', num2str(jj)];

                if exist(meas_name,'var')==0
                    eval([meas_name, '=[]']);
                    chan_list=[chan_list;meas_name];
                end
                kk = [meas_name, '=[]',meas_name, ';
cur_file_data(:,', num2str(jj+4), ')]';];
                eval(kk)
                % show construction progress
                clc
                disp([station_code(j), sprintf('\n')])
                % memory
                perc_done = floor(100*jj/chan_nr);
                disp([sprintf('%3.0f', perc_done),'% of signal
vector construction done, for station ',num2str(j),' /
',num2str(length(station_code))]);

            end
            evalin('base',['clear ',tmp(i,:)]) %Clears base
workspace fro

```

```

        end

    end
    clc
    catch exception
        msgbox('Warning! It appears that some data for the selected
time are missing...')
    end
    clear j
    clear i
    clear jj
    for i=1:length(chan_list(:,1))
        eval(['handles.',chan_list(i,:),'=',chan_list(i,:),';']);
    end
    handles.chan_list=chan_list;
    guidata(hObject,handles);

```

```

%=====
%           end of construction
%=====

```

```

#####
#####
#####
#####
% %% rest of the program
if data_filt == 1

```

```

% All frequency values are in Hz.
% Fpass = 1/(6*3600); % Passband Frequency
% Fstop = 1/(6*36); % Stopband Frequency
% Apass = 1; % Passband Ripple (dB)
% Astop = 90; % Stopband Attenuation (dB)
% Fs = 1; % Sampling Frequency
Fpass=eval(get(handles.edit17, 'string'))
Fstop=eval(get(handles.edit18, 'string'))

```

```

Apass=eval(get(handles.edit19, 'string'))
Astop=eval(get(handles.edit20, 'string'))
Fs=eval(get(handles.edit21, 'string'))

h = fdesign.lowpass('fp,fst,ap,ast', Fpass, Fstop, Apass,
Astop, Fs);
Hd = design(h, 'equiripple', 'MinOrder', 'any', ...
'StopbandShape', 'flat');

% usage: y = filter(Hd,x); , where "x" is input and "y" the
filtered data

for j=1:length(station_code)
clear data_name
data_name = ['RF_',station_code(j)];

for jj=1:chan_nr
meas_name = [data_name,'_ch', num2str(jj)];
kk = ['x = handles.',meas_name,'];
eval(kk);
y = filter(Hd,x);
y(1:800)=y(801);
ll = ['handles.',meas_name,'_fil=y;'];
eval(ll)

% show construction progres
clc
disp([station_code(j), sprintf('\n')])
% memory
perc_done = floor(100*jj/chan_nr);
disp([sprintf('%3.0f', perc_done),'% of signal
filtering done, for station ',num2str(j),' /
',num2str(length(station_code))])

clear(meas_name)

end

end
clear j
clear jj
guidata(hObject,handles);

```

```

msg4=msgbox('Filter is finished. ');
end
msg3=msgbox('Read .dat is finished. ');
clc

%
% %% Import EQ Data
%
% if EQ_swt ==1
%
%     EQ_data = importdata(EQ_file);
%     EQ_data.textdata;
%     EQ_data.data;
%
%     EQ_Date_Time = zeros(length(EQ_data.data(:,1)),30);
%
%     for n = 1:length(EQ_data.data(:,1))
%
%         EQ_Date_Time = cell2mat([num2str(EQ_data.data(n,1)),'-
%',EQ_data.textdata(n,2),'-',...
%                               EQ_data.textdata(n,1),' ',
%                               num2str(EQ_data.data(n,2)),':', ...
%                               num2str(EQ_data.data(n,3)),':',
%                               num2str(EQ_data.data(n,4))]);
%
%         EQ_Date_Time_num(n) = datenum(datevec(EQ_Date_Time));
%         EQ_M(n) = EQ_data.data(n,end);
%
%         if EQ_M(n)<=6.5 && EQ_M(n)>6
%             EQ_M_c(n) = 'r';
%         elseif EQ_M(n)<=6 && EQ_M(n)>5.5
%             EQ_M_c(n) = 'm';
%         elseif EQ_M(n)<=5.5 && EQ_M(n)>5
%             EQ_M_c(n) = 'g';
%         elseif EQ_M(n)<=5 && EQ_M(n)>4.5
%             EQ_M_c(n) = 'c';
%         else
%             EQ_M_c(n) = 'k';
%         end
%
%     end
%
%
%     EQ_Name = 'EQ, M>5R';
%     tElapsed = toc/60;
%     clc

```

```

% disp( sprintf('\n'))
% disp(['EQ data acquired, elapsed time =
',num2str(tElapsed),'min'])
% memory
%
% end
%
%
%% Construct Time Vectors Matrices
%
% load(data_name_file1,data_name)
% year_str = ['year =
ones(size('data_name','(:,3)))*year1;']; % note that in
datenum notation 733407
% eval(year_str);
% %is 2007 12 31, and adding day number gives the date,
% %e.g. 152th day of 2008 is 31/05, full
% %datenum=733407+152 --> 2008 5 31 0 0 0
% day_str = ['day = datevec('data_name','(:,2));'];
% eval(day_str);
% date = [year day(:,[2:3])];
%
% hour_min_str = ['hour_min = ',data_name,'(:,3);'];
% eval(hour_min_str);
% second_str = ['second = ',data_name,'(:,4);'];
% eval(second_str);
%
% data_name_file4 =
['RF_data_',station_code,'_',num2str(year1),'_d',num2str(days
(1)),'_d',num2str(days(end)),'_time_and_more'];
%
% save(data_name_file4)
%
save('temp','data_name_file4','data_name_file3','data_name_fi
le2','data_name_file1')
%
save('temp','data_name_file4','data_name_file3','data_name_fi
le2','data_name_file1')
%
% clear all
%
%
%% Reconstruct Hour & Minute Vectors
%
% load temp
%

```



```

%
load(data_name_file4,'EQ*','year','day','date','data_name','h
our_min',...
%
'second','station_code','year1','days','chan_nr','print_sw','
EQ_swt','print_slct')
%
% tic
%
% hour_init = floor(hour_min(1)/100);
% minute_init = abs(100*hour_init-hour_min(1));
%
% hour = zeros(1,length(hour_min));
% minute = zeros(1,length(hour_min));
% hour(1) = hour_init;
% minute(1) = minute_init;
%
% for m = 2:length(hour_min)
%     if second(m)~=0
%         full_min_ind = 0;
%     else
%         full_min_ind = 1;
%     end
%
%     if full_min_ind == 1
%         if minute(m-1) ~= 59
%             minute(m) = minute(m-1)+1;
%         else
%             minute(m) = 0;
%         end
%     else
%         minute(m) = minute(m-1);
%     end
%
%     if minute(m)== 0 && second(m)== 0
%         full_hour_ind = 1;
%     else
%         full_hour_ind = 0;
%     end
%
%     if full_hour_ind == 1
%         if hour(m-1)~=23
%             hour(m) = hour(m-1)+1;
%         else
%             hour(m) = 0;
%         end
%     end

```

```

%     else
%         hour(m) = hour(m-1);
%     end
% end
% clear m
%
% time = [hour',minute',second];
% toc
% date_time = [date time];
% toc
% date_time_num = datenum(date_time);
% toc
%     date_period      =      [datestr(date_time(1,:)), '      /
% ,datestr(date_time(end,:))];%dd-mm-yyyy
% toc
% dt = 1; % Sampling period in sec
%
%
% t = 0:(length(hour_min)-1)*dt; % in sec
% t_m = t/60; % in min
% t_h = t/3600; %in hours
%
% tElapsed = toc/60;
% clc
% disp( sprintf('\n'))
% disp(['Date-time vectors formed, elapsed time =
% ',num2str(tElapsed),'min'])
% % memory
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%
%
% %% Plotting Results
% if print_sw ==1
%
%     if print_slct(1) == 1
%
%         for m=1:length(station_code)
%
%             for j=1:chan_nr
%                 clear data_name
%                 data_name = ['RF_',station_code(m)];
%                 meas_name = [data_name,'_ch', num2str(j)];
%                 load(data_name_file2,meas_name)
%
%                 legend_meas_name = ['RF ',station_code(m),'

```

```

ch', num2str(j)];
%           title_1 = ['station ',station_code(m),'
channel # ',num2str(j), ', RF E.M. Variation with Time on ',
date_period];
%
%
%
%           figure(100+10*m+j)
%
%
%           hold on
%
%           kk3 = ['plot(date_time_num,', meas_name,
')'];
%           H1 = eval(kk3);
%
set(gca,'XTick',[date_time_num(1):(date_time_num(end)-
date_time_num(1))/10:date_time_num(end)]);
%           datetick('x','mmm-dd','keepticks');
%           set(gca,'xlim',[date_time_num(1)
date_time_num(end)]);
%           set(H1,{'DisplayName'},{legend_meas_name}')
%           temp_ylim = get(gca,'ylim');
%
%           hold on
%
%           if EQ_swt ==1
%
%               for i=1:length(EQ_data.data(:,1))
%
%                   kk4 =
['stem(EQ_Date_Time_num(',num2str(i),'),max(', meas_name,
'))'];
%                   H2 = eval(kk4);
%
set(H2,'Marker','*','MarkerSize',20,'Color',EQ_M_c(i),'LineWi
dth',1.5);
%                   set(gca,'ylim',temp_ylim);
%                   set(H2,{'DisplayName'},{EQ_Name}')
%                   legend ('show', 'Location','Best')
%
%               end
%
%           end
%
%           hold off

```

```

%
%           xlabel(num2str(year1));
%           ylabel('V(t) (mV)');
%           title(title_1);
%
%           figure_name_100j =
['Figure',num2str(100+10*m+j)];
%           print ('-dtiff', figure_name_100j);
%           %hgsave(figure_name_100j)
%           saveas(gcf,figure_name_100j,'pdf')
%
%           clear(data_name)
%           clear(meas_name)
%           close all
%       end
%
%   end
%   clear m
% end
%
%   if print_slct(2) == 1
%
%       for m=1:length(station_code)
%
%           for j=1:chan_nr
%               clear data_name
%               data_name = ['RF_',station_code(m)];
%               meas_name = [data_name,'_ch', num2str(j)];
%               load(data_name_file3,meas_name)
%
%               legend_meas_name = ['RF ',station_code(m),'
ch', num2str(j)];
%               title_1 = ['station ',station_code(m),'
channel # ',num2str(j), ', Filt. RF E.M. Variation with Time
on ', date_period];
%
%
%
%           figure(200+10*m+j)
%
%
%           hold on
%
%               kk3 = ['plot(date_time_num,', meas_name,
')'];
%               H1 = eval(kk3);
%

```

```

set(gca,'XTick',[date_time_num(1):(date_time_num(end)-
date_time_num(1))/10:date_time_num(end)]);
%           datetick('x','mmm-dd','keepticks');
%           set(gca,'xlim',[date_time_num(1)
date_time_num(end)]);
%           set(H1,{'DisplayName'},{legend_meas_name}')
%           temp_ylim = get(gca,'ylim');
%
%           hold on
%
%           if EQ_swt ==1
%
%               for i=1:length(EQ_data.data(:,1))
%
%                   kk4 =
['stem(EQ_Date_Time_num(' ,num2str(i), '),max(' ,      meas_name,
')')'];
%                   H2 = eval(kk4);
%
%                   set(H2,'Marker','*','MarkerSize',20,'Color',EQ_M_c(i),'LineWi
dth',1.5);
%                   set(gca,'ylim',temp_ylim);
%                   set(H2,{'DisplayName'},{EQ_Name}')
%                   legend ('show', 'Location','Best')
%
%               end
%
%           end
%
%           hold off
%
%           xlabel(num2str(year1));
%           ylabel('V(t) (mV)');
%           title(title_1);
%
%                   figure_name_200j =
['Figure',num2str(200+10*m+j)];
%           print ('-dtiff', figure_name_200j);
%           %hgsave(figure_name_100j)
%           saveas(gcf,figure_name_200j,'pdf')
%
%           clear(data_name)
%           clear(meas_name)
%           close all
%
%       end
%
%   end

```

```

%         clear m
%     end
%
%
%
% %% NON-FILTERED DATA COMPARATIVE PLOTS
%
%
%
% %% Case all channels of a station non-filtered
%
% if print_slct(3) == 1
%
%     load_str = ''; Ylabel_str = ''; axShift(1) = 0;
%     for p=1:length(station_code)
%         data_name = ['RF_',station_code(p)];
%         for pp=1:chan_nr-1
%             meas_name = [data_name,'_ch', num2str(pp)];
%             load_str = ['','','meas_name,'];
%             eval(['load('',data_name_file2,',',
% ,load_str,')'])
%
%             data_stct_str =
% ['M.data',num2str(pp),]=[date_time_num','',',';meas_name,'];
% ];
%
%             eval(data_stct_str)
%             clear data_stct_str RF_*
%
%             if EQ_swt==1
%                 scFact(pp) = 1/chan_nr;
%             else
%                 scFact(pp) = 1/(chan_nr-1);
%             end
%
%             YLabelRotation(pp) = 45;
%
%             if pp>1 && EQ_swt==1
%                 axShift(pp) = axShift(pp-1)+1/(chan_nr);
%             elseif pp>1 && EQ_swt~=1
%                 axShift(pp) = axShift(pp-1)+1/(chan_nr-
1);
%
%             end
%
%         end
%
%     end
%
%     if EQ_swt==1
%
%         data_stct_str =
% ['M.data',num2str(pp+1),]=[EQ_Date_Time_num;EQ_M];'];

```

```

%           eval(data_stct_str)
%           scFact(pp+1) = 1/(chan_nr);
%           YLabelRotation(pp+1) = 45;
%           axShift(pp+1) = axShift(pp)+1/(chan_nr);
%       end
%
%
%       figure(300+p)
%       [ax,hLine,hText] = myplotyy(M,...
%           'Xlabel','time (h)',...
%           'YLabelRotation',YLabelRotation,...
%           'ScaleFactor',scFact,...
%           'AxesShift',axShift,...
%           'boolLeftRight',1,...
%           'BoolLineLabel',0,...
%           'boolGGInt',1,...
%           'LineColors', zeros(chan_nr,3));%,lines(5));
%
%       set(ax(1),'XTick',[date_time_num(1):(date_time_num(end)-
%           date_time_num(1))/10:date_time_num(end)]);
%
%       set(ax(1),'XTick',[date_time_num(1):(date_time_num(end)-
%           date_time_num(1))/10:date_time_num(end)]);
%           datetick(ax(1),'x','mmm-dd','keepticks');
%
%       set(ax(1),'XLim',[date_time_num(1);date_time_num(end)]);
%           set(get(ax(1),'XLabel'),'String',['time period:
%           ',datestr(date_time_num(1)),',',datestr(date_time_num(end))]
%           to
%           ',datestr(date_time_num(end))]');
%           set(ax(1),'TickDir','out');
%
%           set(hLine(pp+1),'LineStyle','none')
%           set(hLine(pp+1),'Marker','*')
%       set(ax(pp+1),'Ylim',[(min(EQ_M)-1) (max(EQ_M)+1)])
%
%       for pp=1:chan_nr-1
%           disp_name = ['RF ',station_code(p),' ',
num2str(pp)];
%
%           Ylabel_str =
['set(get(ax(' ,num2str(pp),'),' ,''YLabel''),' ,''String'',' ,'' ,disp
_name, ''')''];
%           eval(Ylabel_str)
%       end
%       set(get(ax(pp+1),'YLabel'),'String','M')
%       title('all channels (non-filtered) of a station')

```

```

%         figure_name = ['Figure',num2str(300+p)];
%         print ('-dtiff', figure_name);
%         %hgsave(figure_name)
%         saveas(gcf,figure_name,'pdf')
%     end
%
%     close all
%     clear p pp load_str Ylabel_str scFact axShift M
% end
%
% %% Case a single channel of all stations non-filtered
%
% if    print_slct(4) == 1
%
%     load_str = ''; Ylabel_str = ''; axShift(1) = 0;
%     for pp=1:chan_nr-1
%         for p=1:length(station_code)
%             data_name = ['RF_',station_code(p)];
%             meas_name = [data_name,'_ch', num2str(pp)];
%             load_str = [''',meas_name,'''];
%                 eval(['load(''',data_name_file2,''',
% ',load_str,')'])
%                 data_stct_str =
% ['M.data',num2str(p),'=[date_time_num''',',';',meas_name,'''];'
% ];
%             eval(data_stct_str)
%             clear data_stct_str RF_*
%
%             if EQ_swt==1
%                 scFact(p) = 1/(1+length(station_code));
%             else
%                 scFact(p) = 1/(length(station_code));
%             end
%
%             YLabelRotation(p) = 45;
%
%             if p>1 && EQ_swt==1
%                 axShift(p) = axShift(p-
% 1)+1/(1+length(station_code));
%             elseif p>1 && EQ_swt~=1
%                 axShift(p) = axShift(p-
% 1)+1/(length(station_code));
%             end
%
%         end
%     end
%
% end

```



```

%           if EQ_swt==1
%
%                               data_stct_str =
['M.data', num2str(p+1), '= [EQ_Date_Time_num; EQ_M];'];
%                               eval(data_stct_str)
%                               scFact(p+1) = 1/(1+length(station_code));
%                               YLabelRotation(p+1) = 45;
%                               axShift(p+1) =
axShift(p)+1/(1+length(station_code));
%                               end
%
%
%
%           figure(400+pp)
%           [ax, hLine, hText] = myplotyy(M, ...
%           'Xlabel', 'time (h)', ...
%           'YLabelRotation', YLabelRotation, ...
%           'ScaleFactor', scFact, ...
%           'AxesShift', axShift, ...
%           'boolLeftRight', 1, ...
%           'BoolLineLabel', 0, ...
%           'boolGGInt', 1, ...
%           'LineColors', zeros(chan_nr, 3)); % , lines(5));
%
%           set(ax(1), 'XTick', [date_time_num(1):(date_time_num(end)-
date_time_num(1))/10:date_time_num(end)]);
%
%           set(ax(1), 'XTick', [date_time_num(1):(date_time_num(end)-
date_time_num(1))/10:date_time_num(end)]);
%           datetick(ax(1), 'x', 'mmm-dd', 'keepticks');
%
%           set(ax(1), 'XLim', [date_time_num(1); date_time_num(end)]);
%           set(get(ax(1), 'XLabel'), 'String', ['time period:
', datestr(date_time_num(1)), ' to
', datestr(date_time_num(end))]);
%           set(ax(1), 'TickDir', 'out');
%
%           set(hLine(p+1), 'LineStyle', 'none')
%           set(hLine(p+1), 'Marker', '*')
%           set(ax(p+1), 'Ylim', [(min(EQ_M)-1) (max(EQ_M)+1)])
%
%           for p=1:length(station_code)
%               disp_name = ['RF ', station_code(p), ' ',
num2str(pp)];
%                               Ylabel_str =
['set(get(ax(' , num2str(p), '), 'YLabel'), 'String', '', disp_
name, ''')'];

```

```

%           eval(Ylabel_str)
%       end
%       set(get(ax(p+1), 'YLabel'), 'String', 'M')
%           title('one channel (non-filtered) for all
stations')
%       figure_name = ['Figure', num2str(400+pp)];
%       print ('-dtiff', figure_name);
%       %hgsave(figure_name)
%       saveas(gcf, figure_name, 'pdf')
%   end
%
%   close all
%   clear p pp load_str Ylabel_str scFact axShift M
% end
%
%% FILTERED DATA COMPARATIVE PLOTS
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%% Case all channels of a station filtered
%
% if print_slct(5) == 1
%
%     load_str = ''; Ylabel_str = ''; axShift(1) = 0;
%     for p=1:length(station_code)
%         data_name = ['RF_', station_code(p)];
%         for pp=1:chan_nr-1
%             meas_name = [data_name, '_ch', num2str(pp)];
%             load_str = ['', meas_name, ''];
%             eval(['load(''', data_name_file3, '',
', load_str, ')'])
%
%                 data_stct_str =
['M.data', num2str(pp), '= [date_time_num'', '; ', meas_name, ''];
'];
%
%             eval(data_stct_str)
%             clear data_stct_str RF_*
%
%             if EQ_swt==1
%                 scFact(pp) = 1/chan_nr;
%             else
%                 scFact(pp) = 1/(chan_nr-1);
%             end
%
%             YLabelRotation(pp) = 45;
%
%             if pp>1 && EQ_swt==1

```

```

%             axShift(pp) = axShift(pp-1)+1/(chan_nr);
%         elseif pp>1 && EQ_swt~=1
%             axShift(pp) = axShift(pp-1)+1/(chan_nr-
1);
%         end
%     end
%
%     if EQ_swt==1
%
%             data_stct_str =
['M.data',num2str(pp+1),'=[EQ_Date_Time_num;EQ_M];'];
%             eval(data_stct_str)
%             scFact(pp+1) = 1/(chan_nr);
%             YLabelRotation(pp+1) = 45;
%             axShift(pp+1) = axShift(pp)+1/(chan_nr);
%         end
%
%
%         figure(500+p)
%         [ax,hLine,hText] = myplotyy(M,...
%             'XLabel','time (h)',...
%             'YLabelRotation',YLabelRotation,...
%             'ScaleFactor',scFact,...
%             'AxesShift',axShift,...
%             'boolLeftRight',1,...
%             'BoolLineLabel',0,...
%             'boolGGInt',1,...
%             'LineColors', zeros(chan_nr,3));% ,lines(5));
%
%         set(ax(1),'XTick',[date_time_num(1):(date_time_num(end)-
date_time_num(1))/10:date_time_num(end)];
%
%         set(ax(1),'XTick',[date_time_num(1):(date_time_num(end)-
date_time_num(1))/10:date_time_num(end)];
%             datetick(ax(1),'x','mmm-dd','keepticks');
%
%         set(ax(1),'XLim',[date_time_num(1);date_time_num(end)];
%             set(get(ax(1),'XLabel'),'String',['time period:
',datestr(date_time_num(1)),'
to
',datestr(date_time_num(end))])
%             set(ax(1),'TickDir','out');
%
%             set(hLine(pp+1),'LineStyle','none')
%             set(hLine(pp+1),'Marker','*')
%         set(ax(pp+1),'Ylim',[(min(EQ_M)-1) (max(EQ_M)+1)])

```

```

%
%         for pp=1:chan_nr-1
%             disp_name = ['RF ',station_code(p),' ',
num2str(pp)];
%
%                                     Ylabel_str =
['set(get(ax(' ,num2str(pp),' ),'YLabel'),'String',' ',disp
_name, ''))'];
%
%             eval(Ylabel_str)
%         end
%         set(get(ax(pp+1),'YLabel'),'String','M')
%         title('all channels (filtered) of a station')
%         figure_name = ['Figure',num2str(500+p)];
%         print ('-dtiff', figure_name);
%         %hgsave(figure_name)
%         saveas(gcf,figure_name,'pdf')
%     end
%
%     close all
%     clear p pp load_str Ylabel_str scFact axShift M
% end
%
% %% Case a single channel of all stationsfiltered
%
% if     print_slct(6) == 1
%
%     load_str = ''; Ylabel_str = ''; axShift(1) = 0;
%     for pp=1:chan_nr-1
%         for p=1:length(station_code)
%             data_name = ['RF_',station_code(p)];
%             meas_name = [data_name,'_ch', num2str(pp)];
%             load_str = ['','','meas_name, '''];
%             eval(['load(''',data_name_file3, ''',
',load_str,')'])
%
%                                     data_stct_str =
['M.data',num2str(p),'=[date_time_num''',';','meas_name, '''];'
];
%
%             eval(data_stct_str)
%             clear data_stct_str RF_*
%
%         if EQ_swt==1
%             scFact(p) = 1/(1+length(station_code));
%         else
%             scFact(p) = 1/(length(station_code));
%         end
%
%         YLabelRotation(p) = 45;

```

```

%
%           if p>1 && EQ_swt==1
%                               axShift(p) = axShift(p-
%                               1)+1/(1+length(station_code));
%           elseif p>1 && EQ_swt~=1
%                               axShift(p) = axShift(p-
%                               1)+1/(length(station_code));
%           end
%
%       end
%
%       if EQ_swt==1
%                               data_stct_str =
% ['M.data',num2str(p+1),'[EQ_Date_Time_num;EQ_M];'];
%                               eval(data_stct_str)
%                               scFact(p+1) = 1/(1+length(station_code));
%                               YLabelRotation(p+1) = 45;
%                               axShift(p+1) =
axShift(p)+1/(1+length(station_code));
%                               end
%
%
%       figure(600+pp)
%       [ax,hLine,hText] = myplotyy(M,...
%       'Xlabel','time (h)',...
%       'YLabelRotation',YLabelRotation,...
%       'ScaleFactor',scFact,...
%       'AxesShift',axShift,...
%       'boolLeftRight',1,...
%       'BoolLineLabel',0,...
%       'boolGGInt',1,...
%       'LineColors', zeros(chan_nr,3));%,lines(5));
%
%       set(ax(1),'XTick',[date_time_num(1):(date_time_num(end)-
date_time_num(1))/10:date_time_num(end)]);
%
%       set(ax(1),'XTick',[date_time_num(1):(date_time_num(end)-
date_time_num(1))/10:date_time_num(end)]);
%       datetick(ax(1),'x','mmm-dd','keepticks');
%
%       set(ax(1),'XLim',[date_time_num(1);date_time_num(end)]);
%       set(get(ax(1),'XLabel'),'String',['time period:
',datestr(date_time_num(1)),
',datestr(date_time_num(end))]);
%       set(ax(1),'TickDir','out');

```

```

%
%         set(hLine(p+1),'LineStyle','none')
%         set(hLine(p+1),'Marker','*')
%         set(ax(p+1),'Ylim',[(min(EQ_M)-1) (max(EQ_M)+1)])
%
%         for p=1:length(station_code)
%             disp_name = ['RF ',station_code(p),' ',
num2str(pp)];
%
%             Ylabel_str =
['set(get(ax(' ,num2str(p),' ),'YLabel'),'String',''),disp_
name, '')'];
%             eval(Ylabel_str)
%         end
%         set(get(ax(p+1),'YLabel'),'String','M')
%         title('one channel (filtered) for all stations')
%         figure_name = ['Figure',num2str(600+pp)];
%         print ('-dtiff', figure_name);
%         %hgsave(figure_name)
%         saveas(gcf,figure_name,'pdf')
%
%
%     end
%
%     close all
%     clear p pp load_str Ylabel_str scFact axShift M
% end
% end

```

```

function importfile_em(fileToRead1)
%IMPORTFILE(FILETOREAD1)
% Imports data from the specified file
% FILETOREAD1: file to read

% Auto-generated by MATLAB on 10-Aug-2010 11:55:49

% Import the file
rawData1 = importdata(fileToRead1);

% For some simple files (such as a CSV or JPEG files),
IMPORTDATA might
% return a simple array. If so, generate a structure so that
the output
% matches that from the Import Wizard.
[unused,name] = fileparts(fileToRead1); %#ok

```

```

assignin('base',name, rawData1);

%=====
====
% Start of Power Law Computation Per Channel of Each Station
%=====
====
% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton14 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
r_critical =str2double(get(handles.edit14,'String'));
radio_choice2=str2double(get(handles.text39, 'String'));
%% Zante Channels
if ( get(handles.checkbox12,'Value') == 1 ) %Then Zante is
checked
    %if ( get(handles.checkbox16,'Value') == 1 )
    %handles.sl_ch_Z=index_of_selected;
    index_of_selected=get(handles.listbox4,'value');
    for i=1:length(index_of_selected)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_Z_ch',num2str(index_of_selected(i))
];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

```

```

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {Z} b(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
        ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {Z} log(a) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

    end

    rr=evalin('base','rr');
    tt=evalin('base','tt');
    r_critical=evalin('base','r_critical');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {Z} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

        guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');

```



```

        evalin('base','clear ta_not_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear ta_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear b');
        evalin('base','clear tb_not_accepted');
        evalin('base','clear b_not_accepted');
        evalin('base','clear tb_accepted');
        evalin('base','clear b_accepted');
    end;

    data_filt = get(handles.checkbox14,'Value');
    if data_filt==1
        index_of_selected_filtered=get(handles.listbox5,
'Value');
        for i=1:length(index_of_selected_filtered)
            clear Step;
            clear dt;
            clear fsize2;

            Step = str2double(get(handles.edit12,'String'));
            dt =str2double(get(handles.edit13,'String'));

            fsize3 = handles.fsize3;

            data_to_plot=['handles.RF_Z_ch',num2str(index_of_selected_fil
tered(i)),'_fil'];

            test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
                if radio_choice2 ==1
                    tt=evalin('base','tt');
                    b=evalin('base','b');

            tb_not_accepted=evalin('base','tb_not_accepted');

            b_not_accepted=evalin('base','b_not_accepted');
                tb_accepted=evalin('base','tb_accepted');
                b_accepted=evalin('base','b_accepted');

                text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {Z} filtered b(t)
'];

            handles.plot_dropdown_names=[handles.plot_dropdown_names;cell

```

```

str(text_to_display)];
    eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {Z} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {Z} filtered r^2(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL2'));

guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');

```

```

        evalin('base','clear log_a_accepted');
        evalin('base','clear b');
        evalin('base','clear tb_not_accepted');
        evalin('base','clear b_not_accepted');
        evalin('base','clear tb_accepted');
        evalin('base','clear b_accepted');
    end
end
end;
%% Kalamata Channels
if ( get(handles.checkbox2,'Value') == 1 ) %Then Kalamata is
checked
    index_of_selected=get(handles.listbox14,'value');
    for i=1:length(index_of_selected)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_O_ch',num2str(index_of_selected(i))
];
test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
        if radio_choice2 ==1
            tt=evalin('base','tt');
            b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
        tb_accepted=evalin('base','tb_accepted');
        b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)),' {0} b(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');

```

```

        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
        ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {0} log(a) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {0} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');

```

```

end;

data_filt = get(handles.checkbox14, 'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox15,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12, 'String'));
        dt =str2double(get(handles.edit13, 'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_0_ch', num2str(index_of_selected_fil
tered(i)), '_fil'];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot), dt, Step, fs
ize3, 0, r_critical, radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base', 'tt');
        b=evalin('base', 'b');

tb_not_accepted=evalin('base', 'tb_not_accepted');

b_not_accepted=evalin('base', 'b_not_accepted');
    tb_accepted=evalin('base', 'tb_accepted');
    b_accepted=evalin('base', 'b_accepted');

        text_to_display=['Channel
', num2str(index_of_selected_filtered(i)), ' {0} filtered b(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names; cell
str(text_to_display)];
        eval(evalin('base', 'send_to_gui_PL'));
    else
        tt=evalin('base', 'tt');
        log_a=evalin('base', 'log_a');

ta_not_accepted=evalin('base', 'ta_not_accepted');

log_a_accepted=evalin('base', 'log_a_accepted');

```

```

        ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {0} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

end

        rr=evalin('base','rr');
        tt=evalin('base','tt');
        r_critical=evalin('base','r_critical');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {0} filtered r^2(t)
'];
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

        guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');

end
end
end;

```

```

%% Kefalonia Channels
if ( get(handles.checkbox3,'Value') == 1 ) %Then Kefalonia is
checked
    index_of_selected=get(handles.listbox12,'value');
    for i=1:length(index_of_selected)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_F_ch',num2str(index_of_selected(i))
];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {F} b(t) '];
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
        ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

```

```

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {F} log(a) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

text_to_display=['Channel
',num2str(index_of_selected(i)), ' {F} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');

end;

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

        index_of_selected_filtered=get(handles.listbox13,
'Value');
        for i=1:length(index_of_selected_filtered)
                clear Step;

```



```

clear dt;
clear fsize2;

Step = str2double(get(handles.edit12,'String'));
dt =str2double(get(handles.edit13,'String'));

fsize3 = handles.fsize3;

data_to_plot=['handles.RF_F_ch',num2str(index_of_selected_filtered(i)),'_fil'];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fsize3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {F} filtered b(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cellstr(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {F} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell

```

```

str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

        end

        rr=evalin('base','rr');
        tt=evalin('base','tt');
        r_critical=evalin('base','r_critical');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {F} filtered r^2(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

        guidata(hObject,handles);

        %clear Data
        %evalin('base','clear tt');
        %evalin('base','clear tt');
        evalin('base','clear log_a');
        evalin('base','clear ta_not_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear ta_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear b');
        evalin('base','clear tb_not_accepted');
        evalin('base','clear b_not_accepted');
        evalin('base','clear tb_accepted');
        evalin('base','clear b_accepted');
    end
end
end;

%% Ioannina Channels
if ( get(handles.checkbox4,'Value') == 1 ) %Then Ioannina is
checked
    index_of_selected=get(handles.listbox8,'value');
    for i=1:length(index_of_selected)
        clear Step;
        clear dt;
        clear fsize2;
    end
end

```

```

Step = str2double(get(handles.edit12,'String'));
dt =str2double(get(handles.edit13,'String'));

fsize3 = handles.fsize3;

data_to_plot=['handles.RF_J_ch',num2str(index_of_selected(i))
];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fsize3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

    text_to_display=['Channel
',num2str(index_of_selected(i)), ' {J} b(t) '];
handles.plot_dropdown_names=[handles.plot_dropdown_names;cellstr(text_to_display)];
    eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

    text_to_display=['Channel
',num2str(index_of_selected(i)), ' {J} log(a) '];
handles.plot_dropdown_names=[handles.plot_dropdown_names;cellstr(text_to_display)];
    eval(evalin('base','send_to_gui_PL') );

```

```

end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

text_to_display=['Channel
',num2str(index_of_selected(i)), ' {J} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
eval(evalin('base','send_to_gui_PL2'));

guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');

end;

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox9,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

```

```

data_to_plot=['handles.RF_J_ch',num2str(index_of_selected_filtered(i)),'_fil'];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
size3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
tb_accepted=evalin('base','tb_accepted');
b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {J} filtered b(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {J} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

end

rr=evalin('base','rr');
tt=evalin('base','tt');

```

```

        r_critical=evalin('base','r_critical');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {J} filtered r^2(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

        guidata(hObject,handles);

        %clear Data
        %evalin('base','clear tt');
        %evalin('base','clear tt');
        evalin('base','clear log_a');
        evalin('base','clear ta_not_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear ta_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear b');
        evalin('base','clear tb_not_accepted');
        evalin('base','clear b_not_accepted');
        evalin('base','clear tb_accepted');
        evalin('base','clear b_accepted');
    end
end
end;
%% Kozani Channels
if ( get(handles.checkbox5,'Value') == 1 ) %Then Kozani is
checked
index_of_selected=get(handles.listbox22,'value');
    for i=1:length(index_of_selected)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_K_ch',num2str(index_of_selected(i))
];

```

```

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {K} b(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {K} log(a) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

    end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

text_to_display=['Channel
',num2str(index_of_selected(i)), ' {K} r^2(t) '];

```

```

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL2'));

    guidata(hObject,handles);

    %clear Data
    %evalin('base','clear tt');
    %evalin('base','clear tt');
    evalin('base','clear log_a');
    evalin('base','clear ta_not_accepted');
    evalin('base','clear log_a_accepted');
    evalin('base','clear ta_accepted');
    evalin('base','clear log_a_accepted');
    evalin('base','clear b');
    evalin('base','clear tb_not_accepted');
    evalin('base','clear b_not_accepted');
    evalin('base','clear tb_accepted');
    evalin('base','clear b_accepted');
end;

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox23,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_K_ch',num2str(index_of_selected_fil
tered(i)),'_fil'];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
        if radio_choice2 ==1
            tt=evalin('base','tt');
            b=evalin('base','b');

```



```

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {K} filtered b(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL'));
    else
    tt=evalin('base','tt');
    log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {K} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL') );

    end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {K} filtered r^2(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL2'));

```

```

guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');
    end
end
end;
%% Komotini Channels
if ( get(handles.checkbox6,'Value') == 1 ) %Then Komotini is
checked
    index_of_selected=get(handles.listbox10,'value');
    for i=1:length(index_of_selected)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_T_ch',num2str(index_of_selected(i))
];
test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
        if radio_choice2 ==1
            tt=evalin('base','tt');
            b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');

```

```

        tb_accepted=evalin('base','tb_accepted');
        b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)),' {T} b(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
        ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)),' {T} log(a) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

    end

    rr=evalin('base','rr');
    tt=evalin('base','tt');
    r_critical=evalin('base','r_critical');

    text_to_display=['Channel
',num2str(index_of_selected(i)),' {T} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

    guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');

```

```

%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');

end;

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox11,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_T_ch',num2str(index_of_selected_fil
tered(i)),'_fil'];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
        if radio_choice2 ==1
            tt=evalin('base','tt');
            b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
        tb_accepted=evalin('base','tb_accepted');
        b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {T} filtered b(t)
'];

```

```

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {T} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL') );

    end

    rr=evalin('base','rr');
    tt=evalin('base','tt');
    r_critical=evalin('base','r_critical');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {T} filtered r^2(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL2'));

    guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');

```

```

        evalin('base','clear log_a_accepted');
        evalin('base','clear ta_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear b');
        evalin('base','clear tb_not_accepted');
        evalin('base','clear b_not_accepted');
        evalin('base','clear tb_accepted');
        evalin('base','clear b_accepted');
    end
end
end;
%% Mytilene Channels
if ( get(handles.checkbox7,'Value') == 1 ) %Then Mytilene is
checked

    index_of_selected=get(handles.listbox6,'value');
    for i=1:length(index_of_selected)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_M_ch',num2str(index_of_selected(i))
];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
        if radio_choice2 ==1
            tt=evalin('base','tt');
            b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
        tb_accepted=evalin('base','tb_accepted');
        b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {M} b(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

```

```

        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

    ta_not_accepted=evalin('base','ta_not_accepted');

    log_a_accepted=evalin('base','log_a_accepted');
        ta_accepted=evalin('base','ta_accepted');

    log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)),' {M} log(a) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

    end

    rr=evalin('base','rr');
    tt=evalin('base','tt');
    r_critical=evalin('base','r_critical');

    text_to_display=['Channel
',num2str(index_of_selected(i)),' {M} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

    guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');

```

```

        evalin('base','clear b_not_accepted');
        evalin('base','clear tb_accepted');
        evalin('base','clear b_accepted');
    end;

    data_filt = get(handles.checkbox14,'Value');
    if data_filt==1

        index_of_selected_filtered=get(handles.listbox7,
'Value');
        for i=1:length(index_of_selected_filtered)
            clear Step;
            clear dt;
            clear fsize2;

            Step = str2double(get(handles.edit12,'String'));
            dt =str2double(get(handles.edit13,'String'));

            fsize3 = handles.fsize3;

data_to_plot=['handles.RF_M_ch',num2str(index_of_selected_fil
tered(i)),'_fil'];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
        if radio_choice2 ==1
            tt=evalin('base','tt');
            b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
            tb_accepted=evalin('base','tb_accepted');
            b_accepted=evalin('base','b_accepted');

            text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {M} filtered b(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
            eval(evalin('base','send_to_gui_PL'));
        else
            tt=evalin('base','tt');
            log_a=evalin('base','log_a');

```



```

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {M} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL') );

end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {M} filtered r^2(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL2'));

guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');

```

```

        end
    end
end;
%% Rhode Channels
if ( get(handles.checkbox8,'Value') == 1 ) %Then Rhode is
checked

    index_of_selected=get(handles.listbox18,'value');
    for i=1:length(index_of_selected)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_A_ch',num2str(index_of_selected(i))
];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {A} b(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');

```

```

        ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)),' {A} log(a) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

end

        rr=evalin('base','rr');
        tt=evalin('base','tt');
        r_critical=evalin('base','r_critical');

        text_to_display=['Channel
',num2str(index_of_selected(i)),' {A} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

        guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');

end;

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

```

```

        index_of_selected_filtered=get(handles.listbox19,
'Value');
        for i=1:length(index_of_selected_filtered)
            clear Step;
            clear dt;
            clear fsize2;

            Step = str2double(get(handles.edit12,'String'));
            dt =str2double(get(handles.edit13,'String'));

            fsize3 = handles.fsize3;

data_to_plot=['handles.RF_A_ch',num2str(index_of_selected_fil
tered(i)),'_fil'];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
        if radio_choice2 ==1
            tt=evalin('base','tt');
            b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
        tb_accepted=evalin('base','tb_accepted');
        b_accepted=evalin('base','b_accepted');

            text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {A} filtered b(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
            eval(evalin('base','send_to_gui_PL'));
        else
            tt=evalin('base','tt');
            log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
            ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

            text_to_display=['Channel

```

```

',num2str(index_of_selected_filtered(i)), ' {A} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL') );

    end

    rr=evalin('base','rr');
    tt=evalin('base','tt');
    r_critical=evalin('base','r_critical');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {A} filtered r^2(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL2'));

    guidata(hObject,handles);

    %clear Data
    %evalin('base','clear tt');
    %evalin('base','clear tt');
    evalin('base','clear log_a');
    evalin('base','clear ta_not_accepted');
    evalin('base','clear log_a_accepted');
    evalin('base','clear ta_accepted');
    evalin('base','clear log_a_accepted');
    evalin('base','clear b');
    evalin('base','clear tb_not_accepted');
    evalin('base','clear b_not_accepted');
    evalin('base','clear tb_accepted');
    evalin('base','clear b_accepted');
    end
end
end

%% Neapolis Channels
if ( get(handles.checkbox9,'Value') == 1 ) %Then Neapolis is
checked

```

```

index_of_selected=get(handles.listbox26,'value');
for i=1:length(index_of_selected)
    clear Step;
    clear dt;
    clear fsize2;

    Step = str2double(get(handles.edit12,'String'));
    dt =str2double(get(handles.edit13,'String'));

    fsize3 = handles.fsize3;

data_to_plot=['handles.RF_E_ch',num2str(index_of_selected(i))
];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {E} b(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {E} log(a) '];

```

```

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

        end

        rr=evalin('base','rr');
        tt=evalin('base','tt');
        r_critical=evalin('base','r_critical');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {E} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

        guidata(hObject,handles);

        %clear Data
        %evalin('base','clear tt');
        %evalin('base','clear tt');
        evalin('base','clear log_a');
        evalin('base','clear ta_not_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear ta_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear b');
        evalin('base','clear tb_not_accepted');
        evalin('base','clear b_not_accepted');
        evalin('base','clear tb_accepted');
        evalin('base','clear b_accepted');

end;

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

        index_of_selected_filtered=get(handles.listbox27,
'Value');
        for i=1:length(index_of_selected_filtered)
                clear Step;
                clear dt;
                clear fsize2;

```

```

Step = str2double(get(handles.edit12,'String'));
dt =str2double(get(handles.edit13,'String'));

fsize3 = handles.fsize3;

data_to_plot=['handles.RF_E_ch',num2str(index_of_selected_fil
tered(i)),'_fil'];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {E} filtered b(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {E} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL') );

```



```

end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {E} filtered r^2(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
eval(evalin('base','send_to_gui_PL2'));

guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');
end
end;
end;
%% Vamos Channels
if ( get(handles.checkbox10,'Value') == 1 ) %Then Vamos is
checked

index_of_selected=get(handles.listbox16,'value');
for i=1:length(index_of_selected)
clear Step;
clear dt;
clear fsize2;

Step = str2double(get(handles.edit12,'String'));
dt =str2double(get(handles.edit13,'String'));

```

```

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_V_ch',num2str(index_of_selected(i))
];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fsize3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {V} b(t) '];
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {V} log(a) '];
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

    end

rr=evalin('base','rr');

```

```

        tt=evalin('base','tt');
        r_critical=evalin('base','r_critical');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {V} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

        guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');

end;

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox17,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_V_ch',num2str(index_of_selected_fil
tered(i)),'_fil'];

```

```

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {V} filtered b(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {V} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

text_to_display=['Channel

```

```

',num2str(index_of_selected_filtered(i)), ' {V} filtered r^2(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL2'));

    guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');
end
end
end
%% Corfu Channels
if ( get(handles.checkbox11,'Value') == 1 ) %Then Corfu is
checked
    selected_indx_Cor=get(handles.listbox24,'value');
    index_of_selected=get(handles.listbox24,'value');
    for i=1:length(index_of_selected)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_P_ch',num2str(index_of_selected(i))
];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)

```

```

        if radio_choice2 ==1
            tt=evalin('base','tt');
            b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
            tb_accepted=evalin('base','tb_accepted');
            b_accepted=evalin('base','b_accepted');

            text_to_display=['Channel
',num2str(index_of_selected(i)),' {P} b(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
            eval(evalin('base','send_to_gui_PL'));
        else
            tt=evalin('base','tt');
            log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
            ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

            text_to_display=['Channel
',num2str(index_of_selected(i)),' {P} log(a) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
            eval(evalin('base','send_to_gui_PL') );

        end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

            text_to_display=['Channel
',num2str(index_of_selected(i)),' {P} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

```

```

eval(evalin('base','send_to_gui_PL2'));

guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');

end;

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox25,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_P_ch',num2str(index_of_selected_fil
tered(i)),'_fil'];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
        if radio_choice2 ==1
            tt=evalin('base','tt');
            b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

```

```

b_not_accepted=evalin('base','b_not_accepted');
    tb_accepted=evalin('base','tb_accepted');
    b_accepted=evalin('base','b_accepted');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {P} filtered b(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL'));
else
    tt=evalin('base','tt');
    log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {P} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL') );

end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {P} filtered r^2(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL2'));

guidata(hObject,handles);

```



```

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');
    end
end
end;
%% Atalanti Channels
if ( get(handles.checkbox13,'Value') == 1 ) %Then Atalanti is
checked

    index_of_selected=get(handles.listbox20,'value');
    for i=1:length(index_of_selected)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_H_ch',num2str(index_of_selected(i))
];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
        if radio_choice2 ==1
            tt=evalin('base','tt');
            b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
        tb_accepted=evalin('base','tb_accepted');
    end
end
end;

```

```

        b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)),' {H} b(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
        ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)),' {H} log(a) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

    end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

        text_to_display=['Channel
',num2str(index_of_selected(i)),' {H} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

        guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');

```

```

        evalin('base','clear log_a');
        evalin('base','clear ta_not_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear ta_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear b');
        evalin('base','clear tb_not_accepted');
        evalin('base','clear b_not_accepted');
        evalin('base','clear tb_accepted');
        evalin('base','clear b_accepted');
    end;

    data_filt = get(handles.checkbox14,'Value');
    if data_filt==1

        index_of_selected_filtered=get(handles.listbox21,
'Value');
        for i=1:length(index_of_selected_filtered)
            clear Step;
            clear dt;
            clear fsize2;

            Step = str2double(get(handles.edit12,'String'));
            dt =str2double(get(handles.edit13,'String'));

            fsize3 = handles.fsize3;

            data_to_plot=['handles.RF_H_ch',num2str(index_of_selected_fil
tered(i)),'_fil'];

            test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
            if radio_choice2 ==1
                tt=evalin('base','tt');
                b=evalin('base','b');

            tb_not_accepted=evalin('base','tb_not_accepted');

            b_not_accepted=evalin('base','b_not_accepted');
            tb_accepted=evalin('base','tb_accepted');
            b_accepted=evalin('base','b_accepted');

            text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {H} filtered b(t)
'];

```

```

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL'));
else
    tt=evalin('base','tt');
    log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {H} filtered log(a)
'];
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL') );

end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {H} filtered r^2(t)
'];
handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL2'));

guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');

```

```

        evalin('base','clear ta_accepted');
        evalin('base','clear log_a_accepted');
        evalin('base','clear b');
        evalin('base','clear tb_not_accepted');
        evalin('base','clear b_not_accepted');
        evalin('base','clear tb_accepted');
        evalin('base','clear b_accepted');
    end
end
end;

%% New Zante Channels
if ( get(handles.checkbox30,'Value') == 1 ) %Then Atalanti is
checked

    index_of_selected=get(handles.listbox28,'value');
    for i=1:length(index_of_selected)
        clear Step;
        clear dt;
        clear fsize2;

        Step = str2double(get(handles.edit12,'String'));
        dt =str2double(get(handles.edit13,'String'));

        fsize3 = handles.fsize3;

data_to_plot=['handles.RF_N_ch',num2str(index_of_selected(i))
];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fsize3,0, r_critical,radio_choice2)
    if radio_choice2 ==1
        tt=evalin('base','tt');
        b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
        tb_accepted=evalin('base','tb_accepted');
        b_accepted=evalin('base','b_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {N} b(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

```

```

        eval(evalin('base','send_to_gui_PL'));
    else
        tt=evalin('base','tt');
        log_a=evalin('base','log_a');

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
        ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

        text_to_display=['Channel
',num2str(index_of_selected(i)),' {N} log(a) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL') );

    end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

text_to_display=['Channel
',num2str(index_of_selected(i)),' {N} r^2(t) '];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
        eval(evalin('base','send_to_gui_PL2'));

        guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');

```

```

        evalin('base','clear b_not_accepted');
        evalin('base','clear tb_accepted');
        evalin('base','clear b_accepted');
    end;

    data_filt = get(handles.checkbox14,'Value');
    if data_filt==1

        index_of_selected_filtered=get(handles.listbox29,
'Value');
        for i=1:length(index_of_selected_filtered)
            clear Step;
            clear dt;
            clear fsize2;

            Step = str2double(get(handles.edit12,'String'));
            dt =str2double(get(handles.edit13,'String'));

            fsize3 = handles.fsize3;

data_to_plot=['handles.RF_N_ch',num2str(index_of_selected_fil
tered(i)),'_fil'];

test_powerlaw_PAOSwt_m1_ex_Fig3(eval(data_to_plot),dt,Step,fs
ize3,0, r_critical,radio_choice2)
        if radio_choice2 ==1
            tt=evalin('base','tt');
            b=evalin('base','b');

tb_not_accepted=evalin('base','tb_not_accepted');

b_not_accepted=evalin('base','b_not_accepted');
            tb_accepted=evalin('base','tb_accepted');
            b_accepted=evalin('base','b_accepted');

            text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {N} filtered b(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
            eval(evalin('base','send_to_gui_PL'));
        else
            tt=evalin('base','tt');
            log_a=evalin('base','log_a');

```

```

ta_not_accepted=evalin('base','ta_not_accepted');

log_a_accepted=evalin('base','log_a_accepted');
    ta_accepted=evalin('base','ta_accepted');

log_a_accepted=evalin('base','log_a_accepted');

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {N} filtered log(a)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL') );

end

rr=evalin('base','rr');
tt=evalin('base','tt');
r_critical=evalin('base','r_critical');

text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {N} filtered r^2(t)
'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];
    eval(evalin('base','send_to_gui_PL2'));

guidata(hObject,handles);

%clear Data
%evalin('base','clear tt');
%evalin('base','clear tt');
evalin('base','clear log_a');
evalin('base','clear ta_not_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear ta_accepted');
evalin('base','clear log_a_accepted');
evalin('base','clear b');
evalin('base','clear tb_not_accepted');
evalin('base','clear b_not_accepted');
evalin('base','clear tb_accepted');
evalin('base','clear b_accepted');

```



```

        end
    end
end;
msg2=msgbox('Power Law Computation was finished');
clear selected_indx_NZa_fil;
clear selected_indx_NZa;
clear selected_indx_Ata_fil;
clear selected_indx_Ata;
clear selected_indx_Cor_fil;
clear selected_indx_Cor;
clear selected_indx_Vam_fil;
clear selected_indx_Vam;
clear selected_indx_Nea_fil;
clear selected_indx_Nea;
clear selected_indx_Rod_fil;
clear selected_indx_Rod;
clear selected_indx_Myt_fil;
clear selected_indx_Myt;
clear selected_indx_Kom_fil;
clear selected_indx_Kom;
clear selected_indx_Koz_fil;
clear selected_indx_Koz;
clear selected_indx_Ioa_fil;
clear selected_indx_Ioa;
clear selected_indx_kef_fil;
clear selected_indx_kef;
clear selected_indx_kal_fil;
clear selected_indx_kal;
clear selected_indx_fil;
clear selected_indx;

%=====
====
% End of Power Law Computation Per Channel of Each Station
%=====
====

%=====
====
% Start of Fuzzy Entropy Computation Per Channel of Each Station

```

```

%=====
====

% --- Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton13 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

%% Zante Channels
if ( get(handles.checkbox12,'Value') == 1 ) %Then Zante is
checked
    index_of_selected=get(handles.listbox4, 'Value');

    for i=1:length(index_of_selected)

        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

        data_to_plot=['handles.RF_Z_ch',num2str(index_of_selected(i))
];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;

```

```

%EQ_DataFrame=EQ_Data(1:fsize4);

M = str2double(get(handles.edit11,'String'));
N=str2double(get(handles.edit10,'String'));
r_var=str2double(get(handles.edit8,'String'));

FuzzyEn(1:fsize4) = zeros(fsize4,1);
for nFrame=2:NoF
    sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
    indx1=(nFrame-1)*inc+1;
    indx2=(nFrame-1)*inc+fsize4;

    r =r_var*std(EQ_Data(indx1:indx2));

    % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
    end
    FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

    text_to_display=['Channel
',num2str(index_of_selected(i)),' {Z} Fuzzy En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
guidata(hObject,handles);
% plot(EQ_Data,'k')
% axis tight;
% title('Input Signal vs. Time');
%
% subplot(3,1,[2:3]);
% plot(FuzzyEn,'k')
% axis tight;
% title_text=['Fuzzy Entropy (W=',num2str(fsize4),'
Overlap=',num2str(ovl),'%) vs. Time'];
% title(title_text);

end

```

```

data_filt = get(handles.checkbox14, 'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox5,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_Z_ch', num2str(index_of_selected(i))
, '_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9, 'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11, 'String'));
        N=str2double(get(handles.edit10, 'String'));
        r_var=str2double(get(handles.edit8, 'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            %sprintf('Remaing execution times: %d\n*', NoF-
nFrame)

            indx1=(nFrame-1)*inc+1;
            indx2=(nFrame-1)*inc+fsize4;

            r =r_var*std(EQ_Data(indx1:indx2));

            % Computation of FuzzyEn per frame

```

```

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
    end
    FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {Z} Filtered Fuzzy
En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
    guidata(hObject,handles);
    end
end

end
%% Kalamata Channels Fuzzy Computation
if ( get(handles.checkbox2,'Value') == 1 ) %Then Kalamata is
checked

    index_of_selected=get(handles.listbox14, 'Value');
    for i=1:length(index_of_selected)

        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_0_ch',num2str(index_of_selected(i))
];
    EQ_Data=eval(data_to_plot);

    fsize4 = handles.fsize4;
    ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
    inc=fsize4*(1-ovl/100);%increment in samples

```

```

NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
%EQ_DataFrame=EQ_Data(1:fsize4);

M = str2double(get(handles.edit11,'String'));
N=str2double(get(handles.edit10,'String'));
r_var=str2double(get(handles.edit8,'String'));

FuzzyEn(1:fsize4) = zeros(fsize4,1);
for nFrame=2:NoF
    sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
    indx1=(nFrame-1)*inc+1;
    indx2=(nFrame-1)*inc+fsize4;

    r =r_var*std(EQ_Data(indx1:indx2));

    % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
    end
    FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

    text_to_display=['Channel
',num2str(index_of_selected(i)), ' {0} Fuzzy En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
    guidata(hObject,handles);
%     plot(EQ_Data,'k')
%     axis tight;
%     title('Input Signal vs. Time');
%
%     subplot(3,1,[2:3]);
%     plot(FuzzyEn,'k')
%     axis tight;
%     title_text=['Fuzzy Entropy (W=',num2str(fsize4),'
Overlap=',num2str(ovl),'%) vs. Time'];
%     title(title_text);

```

```

end

data_filt = get(handles.checkbox14, 'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox15,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_O_ch', num2str(index_of_selected(i))
, '_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9, 'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11, 'String'));
        N=str2double(get(handles.edit10, 'String'));
        r_var=str2double(get(handles.edit8, 'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            %sprintf('Remaing execution times: %d\n*', NoF-
nFrame)

            indx1=(nFrame-1)*inc+1;
            indx2=(nFrame-1)*inc+fsize4;

            r =r_var*std(EQ_Data(indx1:indx2));

```

```

        % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
        end
        FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {0} Filtered Fuzzy
En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
        end
    end

end;
%% Kefalonia Channels Fuzzy Computation
if ( get(handles.checkbox3,'Value') == 1 ) %Then Kefalonia -
Valsamata is checked

    index_of_selected=get(handles.listbox12, 'Value');
    for i=1:length(index_of_selected)

        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_F_ch',num2str(index_of_selected(i))
];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

```



```

NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
%EQ_DataFrame=EQ_Data(1:fsize4);

M = str2double(get(handles.edit11,'String'));
N=str2double(get(handles.edit10,'String'));
r_var=str2double(get(handles.edit8,'String'));

FuzzyEn(1:fsize4) = zeros(fsize4,1);
for nFrame=2:NoF
    sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
    indx1=(nFrame-1)*inc+1;
    indx2=(nFrame-1)*inc+fsize4;

    r =r_var*std(EQ_Data(indx1:indx2));

    % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
    end
    FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

    text_to_display=['Channel
',num2str(index_of_selected(i)),' {F} Fuzzy En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
guidata(hObject,handles);
% plot(EQ_Data,'k')
% axis tight;
% title('Input Signal vs. Time');
%
% subplot(3,1,[2:3]);
% plot(FuzzyEn,'k')
% axis tight;
% title_text=['Fuzzy Entropy (W=',num2str(fsize4),'
Overlap=',num2str(ovl),'%) vs. Time'];
% title(title_text);

```

end;cell

2,411 - 289 -

```

end

data_filt = get(handles.checkbox14, 'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox13,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_F_ch', num2str(index_of_selected(i))
, '_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9, 'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11, 'String'));
        N=str2double(get(handles.edit10, 'String'));
        r_var=str2double(get(handles.edit8, 'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            %sprintf('Remaing execution times: %d\n*', NoF-
nFrame)

            indx1=(nFrame-1)*inc+1;
            indx2=(nFrame-1)*inc+fsize4;

            r = r_var*std(EQ_Data(indx1:indx2));

```

```

        % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
        end
        FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {F} Filtered Fuzzy
En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
        end
    end

end;
%% Ioannina Channels Fuzzy Computation
if ( get(handles.checkbox4,'Value') == 1 ) %Then Ioannina is
checked

    index_of_selected=get(handles.listbox8, 'Value');
    for i=1:length(index_of_selected)

        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_J_ch',num2str(index_of_selected(i))
];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping

```

```

inc=fsize4*(1-ovl/100);%increment in samples

NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
%EQ_DataFrame=EQ_Data(1:fsize4);

M = str2double(get(handles.edit11,'String'));
N=str2double(get(handles.edit10,'String'));
r_var=str2double(get(handles.edit8,'String'));

FuzzyEn(1:fsize4) = zeros(fsize4,1);
for nFrame=2:NoF
    sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
    indx1=(nFrame-1)*inc+1;
    indx2=(nFrame-1)*inc+fsize4;

    r = r_var*std(EQ_Data(indx1:indx2));

    % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
    end
    FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

    text_to_display=['Channel
',num2str(index_of_selected(i)),' {J} Fuzzy En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
guidata(hObject,handles);
% plot(EQ_Data,'k')
% axis tight;
% title('Input Signal vs. Time');
%
% subplot(3,1,[2:3]);
% plot(FuzzyEn,'k')
% axis tight;
% title_text=['Fuzzy Entropy (W=',num2str(fsize4),'
Overlap=',num2str(ovl),'%) vs. Time'];
% title(title_text);

```

```

handles;cell

```

```

handles;

```

```

end

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox9,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_J_ch',num2str(index_of_selected(i))
,'_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            %sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
            indx1=(nFrame-1)*inc+1;
            indx2=(nFrame-1)*inc+fsize4;

```

```

        r = r_var*std(EQ_Data(indx1:indx2));

        % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
        end
        FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)),' {} Filtered Fuzzy
En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
        end
    end

end;
%% Kozani Channels Fuzzy Computation
if ( get(handles.checkbox5,'Value') == 1 ) %Then Kozani is
checked

    index_of_selected=get(handles.listbox22, 'Value');
    for i=1:length(index_of_selected)

        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_K_ch',num2str(index_of_selected(i))
];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %

```

```

percentage of overlapping
    inc=fsize4*(1-ovl/100);%increment in samples

    NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
    %EQ_DataFrame=EQ_Data(1:fsize4);

    M = str2double(get(handles.edit11,'String'));
    N=str2double(get(handles.edit10,'String'));
    r_var=str2double(get(handles.edit8,'String'));

    FuzzyEn(1:fsize4) = zeros(fsize4,1);
    for nFrame=2:NoF
        sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
            indx1=(nFrame-1)*inc+1;
            indx2=(nFrame-1)*inc+fsize4;

            r =r_var*std(EQ_Data(indx1:indx2));

            % Computation of FuzzyEn per frame

    FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
    ;
        end
        FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

        text_to_display=['Channel
',num2str(index_of_selected(i)), ' {K} Fuzzy En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
    guidata(hObject,handles);
%     plot(EQ_Data,'k')
%     axis tight;
%     title('Input Signal vs. Time');
%
%     subplot(3,1,[2:3]);
%     plot(FuzzyEn,'k')
%     axis tight;
%     title_text=['Fuzzy Entropy (W=',num2str(fsize4),'
Overlap=',num2str(ovl),'%) vs. Time'];
%     title(title_text);

```

```

end

data_filt = get(handles.checkbox14, 'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox5,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_K_ch', num2str(index_of_selected(i))
, '_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9, 'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11, 'String'));
        N=str2double(get(handles.edit10, 'String'));
        r_var=str2double(get(handles.edit8, 'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            %sprintf('Remaining execution times: %d\n*', NoF-
nFrame)
            indx1=(nFrame-1)*inc+1;
            indx2=(nFrame-1)*inc+fsize4;

```



```

        r = r_var*std(EQ_Data(indx1:indx2));

        % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
        end
        FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {K} Filtered Fuzzy
En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
        end
    end

end;
%% Komotini Channels Fuzzy
if ( get(handles.checkbox6,'Value') == 1 ) %Then Komotini is
checked

    index_of_selected=get(handles.listbox10, 'Value');
    for i=1:length(index_of_selected)

        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_T_ch',num2str(index_of_selected(i))
];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;

```

```

        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
                indx1=(nFrame-1)*inc+1;
                indx2=(nFrame-1)*inc+fsize4;

                r = r_var*std(EQ_Data(indx1:indx2));

                % Computation of FuzzyEn per frame

        FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
                end
                FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

                text_to_display=['Channel
',num2str(index_of_selected(i)), ' {T} Fuzzy En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
        %
        plot(EQ_Data,'k')
        %
        axis tight;
        %
        title('Input Signal vs. Time');
        %
        subplot(3,1,[2:3]);
        %
        plot(FuzzyEn,'k')
        %
        axis tight;
        %
        title_text=['Fuzzy Entropy (W=',num2str(fsize4),',
Overlap=',num2str(ovl),'%') vs. Time'];

```

```

%           title(title_text);

end

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox11,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_T_ch',num2str(index_of_selected(i))
,'_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            %sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
            indx1=(nFrame-1)*inc+1;

```

```

        indx2=(nFrame-1)*inc+fsize4;

        r =r_var*std(EQ_Data(indx1:indx2));

        % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
        end
        FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {T} Filtered Fuzzy
En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
        end
    end
end;
%% Mytilene Channels
if ( get(handles.checkbox7,'Value') == 1 ) %Then Mytilene is
checked

        index_of_selected=get(handles.listbox6, 'Value');

        for i=1:length(index_of_selected)

                clear EQ_Data;
                clear M;
                clear N;
                clear r_var;
                clear ovl;
                clear inc;
                clear NoF;
                clear r;
                clear FuzzyEn;

data_to_plot=['handles.RF_M_ch',num2str(index_of_selected(i))
];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;

```

```

        ov1 = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ov1/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
                indx1=(nFrame-1)*inc+1;
                indx2=(nFrame-1)*inc+fsize4;

                r =r_var*std(EQ_Data(indx1:indx2));

                % Computation of FuzzyEn per frame

        FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
                end
                FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

                text_to_display=['Channel
',num2str(index_of_selected(i)), ' {M} Fuzzy En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
%         plot(EQ_Data,'k')
%         axis tight;
%         title('Input Signal vs. Time');
%
%         subplot(3,1,[2:3]);
%         plot(FuzzyEn,'k')
%         axis tight;
%         title_text=['Fuzzy Entropy (W=',num2str(fsize4),',
Overlap=',num2str(ov1),'%') vs. Time'];

```

```

%         title(title_text);

end

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox7,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_M_ch',num2str(index_of_selected(i))
,'_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            %sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
            indx1=(nFrame-1)*inc+1;

```

```

        indx2=(nFrame-1)*inc+fsize4;

        r =r_var*std(EQ_Data(indx1:indx2));

        % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
        end
        FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {M} Filtered Fuzzy
En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
        end
    end

end;
%% Rhode Channels Fuzzy Computation
if ( get(handles.checkbox8,'Value') == 1 ) %Then Rhode is
checked
    index_of_selected=get(handles.listbox18, 'Value');

    for i=1:length(index_of_selected)

        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_A_ch',num2str(index_of_selected(i))
];
        EQ_Data=eval(data_to_plot);

```

```

        fsize4 = handles.fsize4;
        ov1 = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ov1/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
                indx1=(nFrame-1)*inc+1;
                indx2=(nFrame-1)*inc+fsize4;

                r =r_var*std(EQ_Data(indx1:indx2));

                % Computation of FuzzyEn per frame

        FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
        ;
            end
            FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

            text_to_display=['Channel
',num2str(index_of_selected(i)), ' {A} Fuzzy En.'];

        handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

        guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
            guidata(hObject,handles);
        %         plot(EQ_Data,'k')
        %         axis tight;
        %         title('Input Signal vs. Time');
        %
        %         subplot(3,1,[2:3]);
        %         plot(FuzzyEn,'k')
        %         axis tight;
        %         title_text=['Fuzzy Entropy (W=',num2str(fsize4),',

```



```

Overlap=',num2str(ovl),'% vs. Time'];
%         title(title_text);

end

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox19,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_A_ch',num2str(index_of_selected(i))
,'_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            %sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
            %));
            %);

```

```

        indx1=(nFrame-1)*inc+1;
        indx2=(nFrame-1)*inc+fsize4;

        r =r_var*std(EQ_Data(indx1:indx2));

        % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
        end
        FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

        text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {A} Filtered Fuzzy
En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
        end
    end

end;
%% Neapolis Channels Fuzzy Computation
if ( get(handles.checkbox9,'Value') == 1 ) %Then Neapolis is
checked

    index_of_selected=get(handles.listbox26, 'Value');

    for i=1:length(index_of_selected)

        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_E_ch',num2str(index_of_selected(i))
];
        EQ_Data=eval(data_to_plot);

```

```

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
                indx1=(nFrame-1)*inc+1;
                indx2=(nFrame-1)*inc+fsize4;

                r =r_var*std(EQ_Data(indx1:indx2));

                % Computation of FuzzyEn per frame

        FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
                end
                FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

                text_to_display=['Channel
',num2str(index_of_selected(i)), ' {E} Fuzzy En.'];

        handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

        guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
        % plot(EQ_Data,'k')
        % axis tight;
        % title('Input Signal vs. Time');
        %
        % subplot(3,1,[2:3]);
        % plot(FuzzyEn,'k')
        % axis tight;

```

```

%         title_text=['Fuzzy Entropy (W=',num2str(fsize4),'
Overlap=',num2str(ovl),'%) vs. Time'];
%         title(title_text);

end

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox27,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_E_ch',num2str(index_of_selected(i))
,'_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            %sprintf('Remaing execution times: %d\n*',NoF-

```

```

nFrame)
    indx1=(nFrame-1)*inc+1;
    indx2=(nFrame-1)*inc+fsize4;

    r =r_var*std(EQ_Data(indx1:indx2));

    % Computation of FuzzyEn per frame

    FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
    end
    FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

    text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {E} Filtered Fuzzy
En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
    guidata(hObject,handles);
    end
end

end;
%% Vamos Channels
if ( get(handles.checkbox10,'Value') == 1 ) %Then Vamos is
checked

    index_of_selected=get(handles.listbox16, 'Value');

    for i=1:length(index_of_selected)

        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

        data_to_plot=['handles.RF_V_ch',num2str(index_of_selected(i))

```

```

];
EQ_Data=eval(data_to_plot);

fsize4 = handles.fsize4;
ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
inc=fsize4*(1-ovl/100);%increment in samples

NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
%EQ_DataFrame=EQ_Data(1:fsize4);

M = str2double(get(handles.edit11,'String'));
N=str2double(get(handles.edit10,'String'));
r_var=str2double(get(handles.edit8,'String'));

FuzzyEn(1:fsize4) = zeros(fsize4,1);
for nFrame=2:NoF
    sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
    indx1=(nFrame-1)*inc+1;
    indx2=(nFrame-1)*inc+fsize4;

    r =r_var*std(EQ_Data(indx1:indx2));

    % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
    end
    FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

    text_to_display=['Channel
',num2str(index_of_selected(i)), ' {v} Fuzzy En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
guidata(hObject,handles);
% plot(EQ_Data,'k')
% axis tight;
% title('Input Signal vs. Time');
%
% subplot(3,1,[2:3]);

```

```

%      plot(FuzzyEn,'k')
%      axis tight;
%      title_text=['Fuzzy Entropy (W=',num2str(fsize4),'',
Overlap=',num2str(ovl),'%) vs. Time'];
%      title(title_text);

end

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox17,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_V_ch',num2str(index_of_selected(i))
,'_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

        FuzzyEn(1:fsize4) = zeros(fsize4,1);

```

```

        for nFrame=2:NoF
            %sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
            indx1=(nFrame-1)*inc+1;
            indx2=(nFrame-1)*inc+fsize4;

            r =r_var*std(EQ_Data(indx1:indx2));

            % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
            end
            FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

            text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {v} Filtered Fuzzy
En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
            end
        end

end;
%% Corfu Channels
if ( get(handles.checkbox11,'Value') == 1 ) %Then Corfu is
checked

        index_of_selected=get(handles.listbox24, 'Value');

        for i=1:length(index_of_selected)

            clear EQ_Data;
            clear M;
            clear N;
            clear r_var;
            clear ovl;
            clear inc;
            clear NoF;
            clear r;
            clear FuzzyEn;

```



```

data_to_plot=['handles.RF_P_ch',num2str(index_of_selected(i))
];
EQ_Data=eval(data_to_plot);

fsize4 = handles.fsize4;
ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
inc=fsize4*(1-ovl/100);%increment in samples

NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
%EQ_DataFrame=EQ_Data(1:fsize4);

M = str2double(get(handles.edit11,'String'));
N=str2double(get(handles.edit10,'String'));
r_var=str2double(get(handles.edit8,'String'));

FuzzyEn(1:fsize4) = zeros(fsize4,1);
for nFrame=2:NoF
    sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
    indx1=(nFrame-1)*inc+1;
    indx2=(nFrame-1)*inc+fsize4;

    r =r_var*std(EQ_Data(indx1:indx2));

    % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
    end
    FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

    text_to_display=['Channel
',num2str(index_of_selected(i)),' {P} Fuzzy En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
guidata(hObject,handles);
% plot(EQ_Data,'k')
% axis tight;
% title('Input Signal vs. Time');
%

```

```

%      subplot(3,1,[2:3]);
%      plot(FuzzyEn,'k')
%      axis tight;
%      title_text=['Fuzzy Entropy (W=',num2str(fsize4),'
Overlap=',num2str(ovl),'%) vs. Time'];
%      title(title_text);

end

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox25,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_P_ch',num2str(index_of_selected(i))
,'_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

```

```

        FuzzyEn(1:fsize4) = zeros(fsize4,1);
        for nFrame=2:NoF
            nFrame)
                %sprintf('Remaing execution times: %d\n*',NoF-
                    indx1=(nFrame-1)*inc+1;
                    indx2=(nFrame-1)*inc+fsize4;

                    r =r_var*std(EQ_Data(indx1:indx2));

                    % Computation of FuzzyEn per frame

FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
                end
                FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

                text_to_display=['Channel
',num2str(index_of_selected_filtered(i)), ' {P} Filtered Fuzzy
En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
        guidata(hObject,handles);
        end
        end

end;
%% Atalanti Channels Fuzzy Computation
if ( get(handles.checkbox13,'Value') == 1 ) %Then Atalanti is
checked

        index_of_selected=get(handles.listbox20, 'Value');

        for i=1:length(index_of_selected)

                clear EQ_Data;
                clear M;
                clear N;
                clear r_var;
                clear ovl;
                clear inc;
                clear NoF;
                clear r;
                clear FuzzyEn;

```

```

data_to_plot=['handles.RF_H_ch',num2str(index_of_selected(i))
];
    EQ_Data=eval(data_to_plot);

    fsize4 = handles.fsize4;
    ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
    inc=fsize4*(1-ovl/100);%increment in samples

    NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
    %EQ_DataFrame=EQ_Data(1:fsize4);

    M = str2double(get(handles.edit11,'String'));
    N=str2double(get(handles.edit10,'String'));
    r_var=str2double(get(handles.edit8,'String'));

    FuzzyEn(1:fsize4) = zeros(fsize4,1);
    for nFrame=2:NoF
        sprintf('Remaing execution times: %d\n*',NoF-
nFrame)
        indx1=(nFrame-1)*inc+1;
        indx2=(nFrame-1)*inc+fsize4;

        r =r_var*std(EQ_Data(indx1:indx2));

        % Computation of FuzzyEn per frame

    FuzzyEn(indx1:indx2)=FuzzyEntr_v3(EQ_Data(indx1:indx2),M,r,N)
;
        end
        FuzzyEn(1:fsize4) = FuzzyEn(inc+1:fsize4+inc);

        text_to_display=['Channel
',num2str(index_of_selected(i)),' {H} Fuzzy En.'];

handles.plot_dropdown_names=[handles.plot_dropdown_names;cell
str(text_to_display)];

guiplot(handles.plot_dropdown_names,FuzzyEn,0,0,0,0,0,6);
    guidata(hObject,handles);
%     plot(EQ_Data,'k')
%     axis tight;
%     title('Input Signal vs. Time');

```

```

%
%     subplot(3,1,[2:3]);
%     plot(FuzzyEn,'k')
%     axis tight;
%     title_text=['Fuzzy Entropy (W=',num2str(fsize4),',
Overlap=',num2str(ovl),'% vs. Time'];
%     title(title_text);

end

data_filt = get(handles.checkbox14,'Value');
if data_filt==1

    index_of_selected_filtered=get(handles.listbox21,
'Value');
    for i=1:length(index_of_selected_filtered)
        clear EQ_Data;
        clear M;
        clear N;
        clear r_var;
        clear ovl;
        clear inc;
        clear NoF;
        clear r;
        clear FuzzyEn;

data_to_plot=['handles.RF_H_ch',num2str(index_of_selected(i))
,'_fil'];
        EQ_Data=eval(data_to_plot);

        fsize4 = handles.fsize4;
        ovl = str2double(get(handles.edit9,'String')); %
percentage of overlapping
        inc=fsize4*(1-ovl/100);%increment in samples

        NoF=floor((length(EQ_Data)-fsize4)/inc)+1;
        %EQ_DataFrame=EQ_Data(1:fsize4);

        M = str2double(get(handles.edit11,'String'));
        N=str2double(get(handles.edit10,'String'));
        r_var=str2double(get(handles.edit8,'String'));

```