

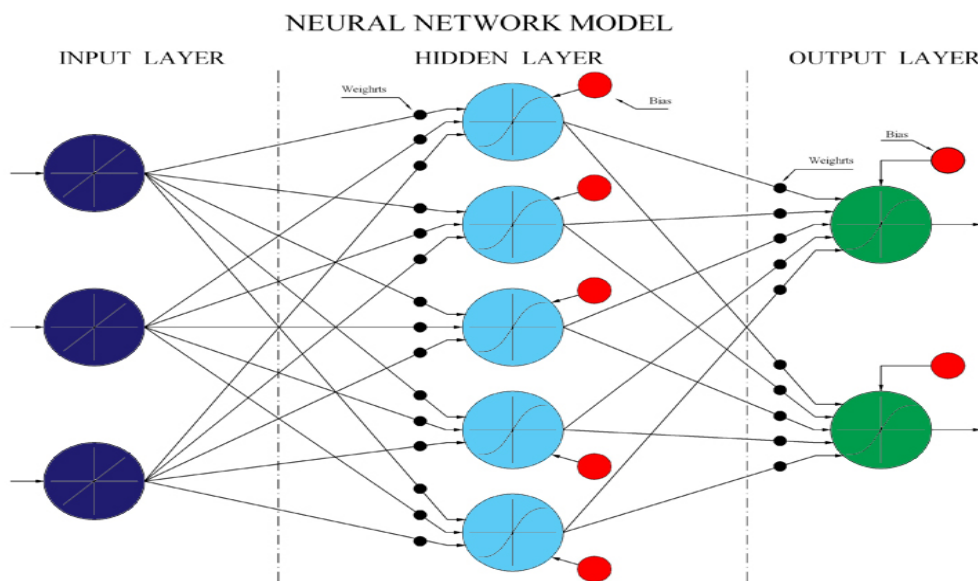


ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΙΑΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΜΕΛΕΤΗ, ΠΡΟΣΟΜΟΙΩΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΕ μC ΜΟΝΤΕΛΟΥ
ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ ΓΙΑ ΤΟΝ ΕΛΕΓΧΟ ΗΛΕΚΤΡΙΚΩΝ
ΜΗΧΑΝΩΝ

ΣΠΟΥΔΑΣΤΗΣ: Πάντος Σωτήρης



ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

Δρ. Μηχ. Ιωάννης Λ. Καρναβάς

ΗΡΑΚΛΕΙΟ 2006

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω:

Τον Επικ. Καθηγητή κ. Ιωάννη Καρναβά, επιβλέπων καθηγητή της παρούσας πτυχιακής εργασίας, για τις υποδείξεις, συμβουλές και την εν γένει καθοδήγησή του κατά τη μελέτη, σύνθεση και παρουσίαση της εργασίας μου. Επίσης, για την επιμέλεια της τελικής μορφής του εντύπου αλλά και γενικά διάθεση στοιχείων και συνεργασία που είχαμε, παρόλο που μεσολαβούσε μεγάλη απόσταση μεταξύ μας.

Τον κ. Τσιάκουλη Πύρρο, Η.Μ & Μ.Η.Υ., υποψήφιος διδάκτορας Ε.Μ.Π., ερευνητής στο ινστιτούτο επεξεργασίας λόγου (ILSP), για την πολύτιμη βοήθεια σε θέματα ανάπτυξης κώδικα και μαθηματικών μοντέλων.

Τέλος θέλω να ευχαριστήσω όλους τους φίλους και δικούς μου ανθρώπους που με βοήθησαν, ο καθένας με τον τρόπο του, καθ' όλη την διάρκεια της παρούσας εργασίας.

Πάντος Σωτήρης

ΑΝΤΙΚΕΙΜΕΝΟ – ΣΥΝΤΟΜΗ ΠΕΡΙΛΗΨΗ

Ο σκοπός της πτυχιακής εργασίας ήταν η ανάπτυξη ευφυούς συστήματος ελέγχου διέγερσης σύγχρονης γεννήτριας. Το σύστημα ελέγχου θα εφαρμοστεί σε σύγχρονη γεννήτρια σε πραγματικές συνθήκες, για να επιτευχθεί αυτό όμως χρησιμοποιήθηκαν δεδομένα από την προσομοίωση λειτουργίας πραγματικού συμβατικού συστήματος ελέγχου σύγχρονης γεννήτριας.

Συγκεκριμένα, έγινε μελέτη και δοκιμές για διάφορα μοντέλα νευρωνικών δικτύων με διαφορετική αρχιτεκτονική και μέγεθος έως ώτου επιλεγεί το καλύτερο για τη συγκεκριμένη εφαρμογή. Μετά από τις δοκιμές μοντέλο που αναδείχθηκε καταλληλότερο για την συγκεκριμένη εφαρμογή είναι μοντέλο T.N.Δ. RBF. η μονάδα επεξεργασίας του συστήματος όπου θα γίνεται η επεξεργασία των δεδομένων είναι ένας μικροελεγκτής AVR ο οποίος θα είναι προγραμματισμένος με το νευρωνικό δίκτυο, γραμμένο σε γλώσσα C.

Το σύστημα αυτό μπορεί να εφαρμοστεί με μικρότερο κόστος και με (καλύτερες) επιδόσεις σε σχέση με ένα συμβατικό σύστημα ελέγχου. Η γενικότητα και η απόδοση του μοντέλου οφείλεται στη χρήση νευρωνικού δικτύου, το οποίο εξαιτίας της εκπαίδευσης του μπορεί να αντιμετωπίζει διαφορετικά προβλήματα και πολλές φορές να επιτυγχάνει καλύτερα αποτελέσματα από τις μεθόδους.

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΓΕΝΙΚΑ ΓΙΑ ΤΗΝ ΗΛΕΚΤΡΙΚΗ ΕΝΕΡΓΕΙΑ	8
1.1	ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ	8
1.2	ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ.	8
1.3	Η ΗΛΕΚΤΡΙΚΗ ΕΝΕΡΓΕΙΑ ΣΤΗ ΣΥΓΧΡΟΝΗ ΕΠΟΧΗ.	9
2	ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΥΓΧΡΟΝΗΣ ΓΕΝΝΗΤΡΙΑΣ ΚΑΙ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΔΙΕΓΕΡΣΗΣ	10
2.1	ΓΕΝΙΚΑ ΓΙΑ ΤΟΝ ΑΥΤΟΜΑΤΟ ΈΛΕΓΧΟ.	10
2.1.1	Το σύστημα ελέγχου	10
2.1.2	Ορισμός της Εισόδου και της Εξόδου	10
2.1.3	Συστήματα Ανοικτού Βρόχου	11
2.1.4	Συστήματα Κλειστού Βρόχου	12
2.2	ΤΟ ΠΡΟΒΛΗΜΑ ΤΟΥ ΕΛΕΓΧΟΥ ΔΙΕΓΕΡΣΗΣ ΣΥΓΧΡΟΝΗΣ ΓΕΝΝΗΤΡΙΑΣ.	13
2.3	ΜΟΝΤΕΛΟ ΣΥΓΧΡΟΝΗΣ ΓΕΝΝΗΤΡΙΑΣ ΣΤΟ ΠΛΑΙΣΙΟ ΑΝΑΦΟΡΑΣ ΤΟΥ ΔΡΟΜΕΑ	14
2.3.1	Υποθέσεις για την ανάπτυξη του μοντέλου.	14
2.3.2	Ανάπτυξη των εξισώσεων του μοντέλου και των ισοδύναμων κυκλωμάτων.	15
2.3.3	Ημιτονοειδής μόνιμη κατάσταση.	19
2.4	ΟΡΙΣΜΟΣ ΠΑΡΑΜΕΤΡΩΝ.	21
2.4.1	Τυπικά Πειραματικά Δεδομένα Πραγματικής Γεννήτριας.	24
2.4.2	Τυπικά Πειραματικά Δεδομένα Διεγέρτριας Μονάδας.	24
3	ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ	25
3.1	ΓΕΝΙΚΑ.	25
3.2	ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ Τ.Ν.Δ.	27
3.2.1	Μοντέλα Τεχνητών Νευρωνικών Δικτύων.	28
3.2.2	Εκπαίδευση.	29
3.2.3	Ικανότητα Γενίκευσης.	31
3.3	ΓΝΩΣΤΑ ΜΟΝΤΕΛΑ Τ.Ν.Δ.	32
3.3.1	Perceptron.	32
3.3.2	Adaline.	34
3.3.3	Brain State in a Box.	35
3.3.4	Radial Basis Function Network (RBF).	36
3.3.5	Άλλα μοντέλα νευρωνικών δικτύων.	36
3.4	ΕΦΑΡΜΟΓΕΣ Τ.Ν.Δ.	37
4	ΜΙΚΡΟΕΛΕΓΚΤΗΣ	38
4.1	Η ΒΑΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ – ΑΝΑΠΤΥΞΙΑΚΗ ΠΛΑΚΕΤΑ.	38
4.2	ΜΙΚΡΟΕΛΕΓΚΤΗΣ Η ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΗΣ ?	38
4.3	ΜΙΚΡΟΕΛΕΓΚΤΕΣ.	39
4.3.1	Μικροελεγκτής AVR ATmega16.	40
4.3.2	Προγραμματισμός Μικροελεγκτή AVR.	41
4.3.3	CodeVision AVR	42
4.3.4	AVR Studio.	42
4.3.5	Παράδειγμα ανάπτυξης ενός προγράμματος για μικροελεγκτή με τη χρήση του CodeVision AVR.	42
4.4	ΕΦΑΡΜΟΓΕΣ ΜΙΚΡΟΕΛΕΓΚΤΩΝ.	48
5	ΠΡΟΣΟΜΟΙΩΣΗ ΤΝΔ ΜΕΣΩ ΜΑΤΛΑΒ	49
5.1	ΕΞΕΛΙΞΗ Η/Υ ΚΑΙ ΛΟΓΙΣΜΙΚΟΥ.	49
5.1.1	Μέθοδοι Υλοποίησης Τ.Ν.Δ.	49
5.1.2	Λογισμικό ανάπτυξης ΤΝΔ – MATLAB.	50
5.1.3	Δημιουργία Μοντέλου Τ.Ν.Δ.	51
5.1.3.1	Δημιουργία Μοντέλου “Feed Forward Back Propagation”	51
5.1.3.2	Συναρτήσεις ενεργοποίησης	52
5.1.3.3	Συναρτήσεις εκπαίδευσης	54
5.1.3.4	Έλεγχος Απόδοσης Μοντέλου	56
5.1.4	Η Γλώσσα Προγραμματισμού C.	56
5.2	ΤΟ ΠΡΟΒΛΗΜΑ ΤΗΣ “XOR”.	57
5.3	ΤΟ ΠΡΟΒΛΗΜΑ “IRIS”.	58
6	ΕΦΑΡΜΟΓΗ Τ.Ν.Δ. ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΣΥΣΤΗΜΑ ΕΛΕΓΧΟΥ ΔΙΕΓΕΡΣΗΣ	67

6.1	ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ ΥΠΟ ΜΕΛΕΤΗ.-----	67
6.2	ΔΙΑΔΙΚΑΣΙΑ ΕΚΠΑΙΔΕΥΣΗΣ Τ.Ν.Δ. -----	68
6.2.1	Κανονικοποίηση Δεδομένων Εισόδου – Εξόδου. -----	68
6.2.2	Δοκιμή Μοντέλου Τ.Ν.Δ. “MLP” .-----	69
6.2.3	Δοκιμή Μοντέλου Τ.Ν.Δ. “RBF” .-----	72
6.2.4	Αποτελέσματα Δοκιμής Μοντέλου Τ.Ν.Δ. RBF.-----	73
6.3	ΕΠΙΛΟΓΗ ΜΟΝΤΕΛΟΥ Τ.Ν.Δ. “RBF” .-----	75
6.4	ΟΛΟΚΛΗΡΩΣΗ ΕΥΦΥΟΥΣ ΕΛΕΓΚΤΗ ΣΕ ΠΡΑΚΤΙΚΟ ΣΥΣΤΗΜΑ ΕΛΕΓΧΟΥ ΔΙΕΓΕΡΣΗΣ.-----	86
6.4.1	Είσοδος Συστήματος-----	86
6.4.2	Συνδεσμολογία Συστήματος .-----	86
6.4.3	Αλγόριθμος Μοντέλου .-----	87
6.4.4	Κώδικας C Μοντέλου RBF.-----	88
7	ΣΥΜΠΕΡΑΣΜΑΤΑ -----	93
	ΠΑΡΑΡΤΗΜΑ Α: ΝΟΜΟΙ ΚΑΙ ΘΕΩΡΗΜΑΤΑ Τ.Ν.Δ. -----	95
8	ΠΑΡΑΡΤΗΜΑ Β: ΚΩΔΙΚΑΣ -----	96
B.1.	ΚΩΔΙΚΑΣ MATLAB (MATLAB SCRIPT) -----	96
B.1.1	Πρόβλημα “IRIS” – Μοντέλο MLP.-----	96
B.1.2	Μοντέλο RBF.-----	105
B.1.3	Κοινά αρχεία – Αποθήκευση τιμών.-----	109
B.2	ΚΩΔΙΚΑΣ C.-----	112
B.2.1	Κώδικας μοντέλου XOR σε C (ANSI).-----	112
B.2.2	Κώδικας μοντέλου IRIS με ένα κρυφό επίπεδο σε C (ANSI).-----	116
B.2.3	Κώδικας μοντέλου IRIS με δύο κρυφά επίπεδα σε C (ANSI).-----	121
B.2.4	Κώδικας μοντέλου RBF σε C (ANSI).-----	127
	ΠΑΡΑΡΤΗΜΑ Γ: ΠΕΡΙΓΡΑΦΗ ΠΡΑΓΜΑΤΙΚΟΥ ΕΜΠΟΡΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΔΙΕΓΕΡΣΗΣ	
	ΣΥΓΧΡΟΝΗΣ ΓΕΝΝΗΤΡΙΑΣ -----	132
	ΕΙΣΑΓΩΓΗ -----	133
	ΕΦΑΡΜΟΓΕΣ ΣΥΣΤΗΜΑΤΟΣ ΔΙΕΓΕΡΣΗΣ-----	133
	ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ-----	133
	ΚΥΡΙΟ ΚΥΚΛΩΜΑ-----	134
	ΈΛΕΓΧΟΣ ΚΑΙ ΔΙΑΓΝΩΣΤΙΚΑ-----	134
	ΜΟΝΑΔΑ ΕΛΕΓΧΟΥ-----	135
	ΔΙΕΠΑΦΗ ΑΝΘΡΩΠΟΥ-ΜΗΧΑΝΗΣ-----	135
	ΠΑΡΑΡΤΗΜΑ Δ: ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΣΥΣΤΗΜΑΤΟΣ Η/Υ ΣΤΟΝ ΟΠΟΙΟ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΑΝ	
	ΟΙ ΔΟΚΙΜΕΣ ΚΑΙ ΟΙ ΠΡΟΣΟΜΟΙΩΣΕΙΣ ΤΗΣ ΠΑΡΟΥΣΙΑΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ -----	139
	ΑΝΑΦΟΡΕΣ -----	140
	ΒΙΒΛΙΟΓΡΑΦΙΑ-----	140
	ΙΣΤΟΣΕΛΙΔΕΣ (URL)-----	141

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

Σχ. 1.1 Το Φωτοηλεκτρικό φαινόμενο	8
Σχ. 1.2 Σύγχρονες γεννήτριες σε σταθμό παραγωγής	9
Σχ. 2.1 Απλοποιημένο διάγραμμα συστήματος ελέγχου	10
Σχ. 2.2 Απόκριση εισόδου – εξόδου ενός συστήματος ελέγχου ανελκυστήρα	11
Σχ. 2.3 Σύστημα ανοικτού βρόχου	11
Σχ. 2.4 Σύστημα κλειστού βρόχου	12
Σχ. 2.5 Ισοδύναμο κύκλωμα σύγχρονης γεννήτριας στο πλαίσιο αναφοράς του δρομέα	18
Σχ. 2.6 Διανυσματικό διάγραμμα στο χώρο για σταθερή ημιτονοειδή κατάσταση	21
Σχ. 3.1 τυπική αναπαράσταση τομής ανθρώπινου εγκέφαλου	25
Σχ. 3.2 Βιολογικός νευρώνας	26
Σχ. 3.3 Τυπικός Τεχνητός Νευρώνας	26
Σχ. 3.4 Τυπικό μοντέλο Τ.Ν.Δ. με ένα κρυφό επίπεδο	28
Σχ. 3.5 Στατικό μοντέλο (“Feed Forward” [2x2x1])	28
Σχ. 3.6 Δομή Μοντέλου “HOPFIELD”	29
Σχ. 3.7 Συνάρτηση μεταβολής των βαρών	30
Σχ. 3.8 Μοντέλο “Kohonen”, ο κόκκινος νευρώνας «επιβιώνει».	31
Σχ. 3.9 Συνάρτηση μεταφοράς ανιχνευτών προτύπων.	33
Σχ. 3.10 Γραμμική συνάρτηση κατωφλίου	33
Σχ. 3.11 Σχηματικό διάγραμμα Τ.Ν.Δ. “Adaline”	34
Σχ. 3.12 Δραστηριοποίηση Τ.Ν.Δ “BSB”	35
Σχ. 3.13 Σχηματική αναπαράσταση δυναμικού μοντέλου στο χώρο	36
Σχ. 4.1 Αναπτυξιακή πλακέτα ATMEL STK – 500 (starter kit)	38
Σχ. 4.2 Μικροεπεξεργαστής – CPU (AMD Atnlon 64bit)	39
Σχ. 4.3 Μικροελεγκτής ATMEL	40
Σχ. 4.4 Σχηματικό διάγραμμα του μικροελεγκτή ATmega16	41
Σχ. 4.5 Περιβάλλον CodeVision AVR	42
Σχ. 4.6 Το περιβάλλον εργασίας CodeVision AVR.	43
Σχ. 4.7 Δημιουργία νέου αρχείου.	43
Σχ. 4.8 Επιλογή τύπου για το αρχείο που θα δημιουργηθεί.	44
Σχ. 4.9 Ορισμός ονόματος για το αρχείο του προγράμματος.	44
Σχ. 4.10 Ορισμός παραμέτρων προγράμματος.	45
Σχ. 4.11 Προσθήκη ενός αρχείου κώδικα.	45
Σχ. 4.12 Περιβάλλον ανάπτυξης και μεταγλωττισμού του κώδικα.	46
Σχ. 4.13 Προγραμματισμός του μικροελεγκτή μέσω του ενσωματωμένου προγραμματιστή.	46
Σχ. 5.1 Περιβάλλον εργασίας “MATLAB”	50
Σχ. 5.2 Συνάρτηση μεταφοράς (ενεργοποίησης) “Purelin”.	52
Σχ. 5.3 Συνάρτηση μεταφοράς (ενεργοποίησης) “LogSig”.	52
Σχ. 5.4 Συνάρτηση μεταφοράς (ενεργοποίησης) “Tansig”	53
Σχ. 5.5 Συνάρτηση μεταφοράς (ενεργοποίησης) “radbas”.	53
Σχ. 5.6 Πύλη “XOR”	57
Σχ. 5.7 Έξοδος μοντέλου συνάρτησης “XOR”	58
Σχ. 5.8 Γραφική παράσταση (MSE) της εκπαίδευσης μοντέλου με ένα κρυφό επίπεδο (4x7x1)	64
Σχ. 5.9 Έξοδος μοντέλου με ένα κρυφό επίπεδο [4x7x1]	65
Σχ. 5.10 Γραφική παράσταση (MSE) της εκπαίδευσης μοντέλου με 2 κρυφά επίπεδα (4x8x8x1)	65
Σχ. 5.11 Έξοδος μοντέλου με 2 κρυφά επίπεδα (4x8x8x1)	66
Σχ. 6.1 Συμβατικό σύστημα ελέγχου διέγερσης σύγχρονης γεννήτριας κλειστού βρόχου	68
Σχ. 6.2 Έξοδος μοντέλου (6x24x1)	70
Σχ. 6.3 Εκπαίδευση μοντέλου (6x24x1)	70
Σχ. 6.4 Έξοδος μοντέλου (6x13x3x1)	71
Σχ. 6.5 Εκπαίδευση μοντέλου (6x13x3x1)	71
Σχ. 6.6 Έξοδος μοντέλου RBF (6x1x1), Σενάριο 2	75
Σχ. 6.7 Έξοδος μοντέλου RBF (6x1x1), Σενάριο 3	75
Σχ. 6.8 Έξοδος μοντέλου RBF (6x2x1), Σενάριο 2	76
Σχ. 6.9 Έξοδος μοντέλου RBF (6x2x1), Σενάριο 3	76
Σχ. 6.10 Έξοδος μοντέλου RBF (6x4x1), Σενάριο 2	77
Σχ. 6.11 Έξοδος μοντέλου RBF (6x4x1), Σενάριο 3	77
Σχ. 6.12 Έξοδος μοντέλου RBF (6x8x1), Σενάριο 2	78
Σχ. 6.13 Έξοδος μοντέλου RBF (6x8x1), Σενάριο 3	78
Σχ. 6.14 Έξοδος μοντέλου RBF (6x16x1), Σενάριο 2	79
Σχ. 6.15 Έξοδος μοντέλου RBF (6x16x1), Σενάριο 3	79
Σχ. 6.16 Έξοδος μοντέλου RBF (6x32x1), Σενάριο 2	80

Σχ. 6.17 Έξοδος μοντέλου RBF (6x32x1), Σενάριο 3	80
Σχ. 6.18 Έξοδος μοντέλου RBF (6x64x1), Σενάριο 2	81
Σχ. 6.19 Έξοδος μοντέλου RBF (6x64x1), Σενάριο 3	81
Σχ. 6.20 Έξοδος μοντέλου RBF (6x128x1), Σενάριο 2	82
Σχ. 6.21 Έξοδος μοντέλου RBF (6x128x1), Σενάριο 3	82
Σχ. 6.22 Έξοδος μοντέλου RBF (6x256x1), Σενάριο 2	83
Σχ. 6.23 Έξοδος μοντέλου RBF (6x256x1), Σενάριο 3	83
Σχ. 6.24 Κριτήριο απόδοσης J_1 του μοντέλου RBF(Σενάρια 1 – 4)	84
Σχ. 6.25 Κριτήριο απόδοσης J_1 του μοντέλου RBF(Σενάρια 2,5,6)	85
Σχ. 6.26 Χρόνος εκπαίδευσης μοντέλου RBF (Σενάρια 1 – 6)	85
Σχ. 6.27 Μοντέλο T.N.Δ RBF προς υλοποίηση	85
Σχ. 6.28 Ψηφιο-Αναλογικός μετατροπέας (A/D analog to digital).	86
Σχ. 6.29 Διάγραμμα Συστήματος Ελέγχου	87

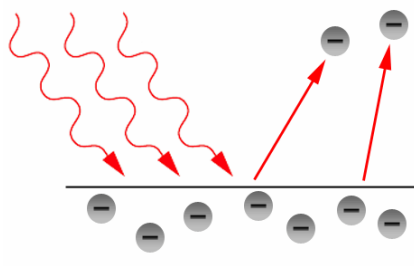
1 ΓΕΝΙΚΑ ΓΙΑ ΤΗΝ ΗΛΕΚΤΡΙΚΗ ΕΝΕΡΓΕΙΑ

1.1 Ιστορική Αναδρομή

Ο ηλεκτρισμός (στατικό ηλεκτρικό φορτίο) είχε ανακαλυφθεί¹ προ χιλιάδων ετών, όμως στους πρόσφατους αιώνες της σύγχρονης ιστορίας του ο άνθρωπος «κατανόησε» και χρησιμοποίησε τις ιδιότητες της ηλεκτρικής ενέργειας(κινούμενα ηλεκτρικά φορτία) με αποτέλεσμα η ηλεκτρική ενέργεια να αποτελέσει αναπόσπαστο τμήμα του σύγχρονου πολιτισμού. Την επανάσταση αυτή έφερε η ανακάλυψη του Faraday², γεγονός το οποίο οδήγησε στην εφεύρεση της ηλεκτρικής μηχανής. Πρωτοπόροι στην διάδοση της ηλεκτρικής ενέργειας και στην βελτίωση της ηλεκτρικής μηχανής ήταν κυρίως δύο (ανταγωνιστές) άνθρωποι, ο Tesla που προωθούσε το εναλλασσόμενο (ac) και ο Edison που υποστήριζε το συνεχές (dc) και όπως όλοι γνωρίζουμε επικράτησε το πρότυπο του Tesla, κυρίως για οικονομικούς λόγους. Έτσι πλέον η ηλεκτρική ενέργεια μπορούσε να παραχθεί από άλλες μορφές ενέργειας να μεταφερθεί και να ξαναμετατραπεί σε μηχανική ενέργεια με λογικό κόστος. Το αποτέλεσμα ήταν η ταχεία εξάπλωση της ηλεκτρικής ενέργειας και η δημιουργία των πρώτων μεγάλων σταθμών παραγωγής και δικτύων ηλεκτρικής ενέργειας στις αρχές του προηγούμενου αιώνα.

1.2 Χαρακτηριστικά Ηλεκτρικής Ενέργειας.

Τα στοιχεία που κάνουν την ηλεκτρική ενέργεια εύκολα εκμεταλλεύσιμη είναι η παραγωγή, η μεταφορά και η χρήσεις της. Για την παραγωγή ηλεκτρικής ενέργειας χρησιμοποιούνται κυρίως τρεις «μέθοδοι», η αλληλεπίδραση ηλεκτρομαγνητικών πεδίων, τα φωτοβολταϊκά στοιχεία – φωτοηλεκτρικό φαινόμενο³ και ηλεκτροχημική αντίδραση (κυψέλες καυσίμου, μπαταρίες κλπ). Η πρώτη μέθοδος είναι αυτή που χρησιμοποιείται για το μεγαλύτερο μέρος (εξ ολοκλήρου σχεδόν) της παραγωγής, καθώς το ηλεκτροπαραγωγό ζεύγος αποτελούμενο από την (σύγχρονη) γεννήτρια και την κινητήρια μηχανή χρησιμοποιείται σε θερμοηλεκτρικούς, πυρηνικούς, υδροηλεκτρικούς σταθμούς παραγωγής, που παράγουν το μεγαλύτερο μέρος της ηλεκτρικής ενέργειας, και γενικά σε κάθε σύστημα που έχουμε μετατροπή μηχανικής ενέργειας σε ηλεκτρική. Η ηλεκτρική ενέργεια μπορεί να μεταφέρεται «εύκολα» μέσω των ηλεκτρικών δικτύων, καθώς το ηλεκτρικό ρεύμα ρέει (σαν υγρό) στα αγωγικά υλικά. Οι χρήσεις της ηλεκτρικής ενέργειας αποτελούν θεμέλιο της σύγχρονης κοινωνίας και τρόπου ζωής, καθώς παντού γύρω μας υπάρχουν ηλεκτρικές εφαρμογές, από τις βαριές βιομηχανικές μέχρι και στον τρόπο διαβίωσης και επικοινωνίας του σύγχρονου ανθρώπου. Το γεγονός αυτό κατά ένα μεγάλο βαθμό οφείλεται στο ότι η ηλεκτρική ενέργεια μετατρέπεται εύκολα σε άλλες μορφές ενέργειας (π.χ. μηχανική, θερμική κλπ).



Σχ. 1.1 Το Φωτοηλεκτρικό φαινόμενο

¹ Ο Θαλής ο Μιλήσιος (περίπου το 600 π.Χ.) παρατήρησε ότι κομμάτι ήλεκτρου που τρίβεται σε ξηρό ύφασμα έλκει μικρά κομμάτια άχυρου. Το 19ο αιώνα αναπτύχθηκαν βαθμιαία διάφορες τεχνικές εφαρμογές του ηλεκτρισμού, αλλά μόλις τον 20ο αιώνα αναγνωρίστηκε η διάχυτη παρουσία των ηλεκτρικών δυνάμεων οι οποίες συγκρατούν όλη την ύλη του περιβάλλοντός μας.

² Ο Faraday παρατήρησε την επίδραση που είχε ένα μεταβαλλόμενο μαγνητικό πεδίο (κινούμενος μαγνήτης) σε ένα ακίνητο πηνίο.

³ Το Φωτοηλεκτρικό φαινόμενο είναι το φαινόμενο κατά το οποίο εξάγονται ηλεκτρόνια από μια μεταλλική επιφάνεια όταν προσπέσει σε αυτή ηλεκτρομαγνητική ακτινοβολία κατάλληλης συχνότητας. Η ερμηνεία του φωτοηλεκτρικού φαινομένου έγινε το 1905 από τον Άλμπερτ Αϊνστάιν που πήρε το βραβείο Νόμπελ για αυτή του την εργασία.

1.3 Η Ηλεκτρική Ενέργεια στη Σύγχρονη Εποχή.

Εύκολα διαπιστώνουμε πως στη σύγχρονη εποχή η ηλεκτρική ενέργεια αποτελεί αναπόσπαστο τμήμα του κοινωνικού οικοδομήματος. Στις αναπτυγμένες χώρες η ποσότητα ηλεκτρικής ενέργειας που καταναλώνουν οι πολίτες αποτελεί και κριτήριο για το βιοτικό επίπεδο⁴ [17], αποδεικνύοντας πόσο καθοριστικό ρόλο έχει στην σύγχρονη τεχνολογική εποχή. Έτσι, όπως και ισχύει και για άλλα βασικά αγαθά που απολαμβάνει ο άνθρωπος, πρέπει να παρέχεται σε ικανοποιητική ποσότητα και στην απαραίτητη ποιότητα. Χαρακτηριστικό είναι ότι σε περιπτώσεις παρατεταμένης διακοπής ηλεκτρικής ενέργειας δημιουργούνται σοβαρά οικονομικά προβλήματα και ταυτόχρονα υπάρχει κίνδυνος για ανθρώπινες ζωές. Η ποσότητα αλλά και η ποιότητα της παρερχόμενης ηλεκτρικής ενέργειας εξαρτάται από την παραγωγή αλλά και τη μεταφορά. Η μεταφορά ηλεκτρικής ενέργειας, εκτός από τη δομή των ηλεκτρικών δικτύων, μπορεί να επηρεαστεί και από αστάθμητους παράγοντες⁵. Βέβαια όπως εύκολα αντιλαμβάνεται κανείς η παραγωγή καθορίζει κατά το μεγαλύτερο βαθμό την παρερχόμενη ηλεκτρική ενέργεια. Όπως ήδη έχουμε αναφέρει η παραγωγή ηλεκτρικής ενέργειας στο μεγαλύτερο ποσοστό γίνεται από ηλεκτροπαραγωγά ζεύγη σύγχρονης γεννήτριας και κινητήριας μηχανής. Ο έλεγχος του ηλεκτροπαραγωγού ζεύγους μπορεί να γίνεται μέσω της κινητήριας μηχανής, της γεννήτριας ή και των δύο. Γνωρίζουμε βέβαια πως σε όλους τους σταθμούς παραγωγής ο έλεγχος πραγματοποιείται μέσω και των δύο τμημάτων του ζεύγους, καθώς αυτός είναι ο πλέον ενδεδειγμένος τρόπος. Ο έλεγχος μέσω της κινητήριας μηχανής υστερεί σημαντικά σε χρόνο, λόγω αδράνειας, σε σχέση με τον έλεγχο της σύγχρονης γεννήτριας ο οποίος επιτυγχάνεται μέσω του ελέγχου της διέγερσης. Έτσι για τις περισσότερες ρυθμίσεις στο σύστημα προτιμάται ο έλεγχος μέσω της διέγερσης, για το λόγο αυτό στην παρούσα πτυχιακή εργασία θα ασχοληθούμε με ένα σύστημα ελέγχου διέγερσης σύγχρονης γεννήτριας.



Σχ. 1.2 Σύγχρονες γεννήτριες σε σταθμό παραγωγής

⁴ Τον τελευταίο καιρό όπου πλέον αρχίζει η ανθρωπότητα να κατανοεί τους κινδύνους που διατρέχει το περιβάλλον (και η ύπαρξη της) από την αλόγιστη χρήση ενέργειας γίνονται προσπάθειες εξοικονόμησης ενέργειας και ταυτόχρονα στροφή προς τις ανανεώσιμες πηγές ενέργειας.

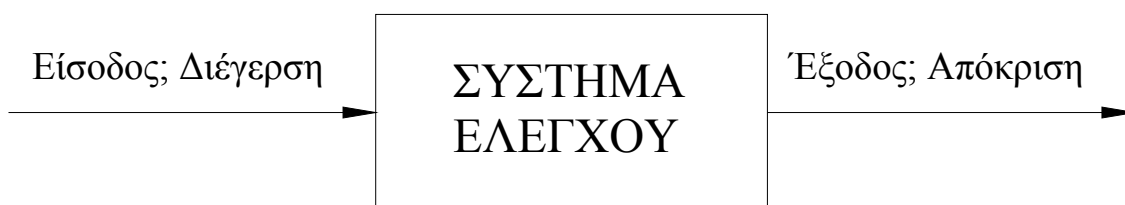
⁵ Η γραμμές μεταφοράς μπορεί να επηρεαστούν από κεραυνούς, ηλιακές καταιγίδες, σεισμούς κλπ.

2 ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΥΓΧΡΟΝΗΣ ΓΕΝΝΗΤΡΙΑΣ ΚΑΙ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΔΙΕΓΕΡΣΗΣ

2.1 Γενικά για τον Αυτόματο Έλεγχο.

2.1.1 Το σύστημα ελέγχου

Ένα σύστημα ελέγχου αποτελείται από επιμέρους διαδικασίες (processes) οι οποίες αντιστοιχούν σε βαθμίδες ελέγχου κατάλληλα συνδεδεμένες μεταξύ τους με σκοπό τον έλεγχο της εξόδου των διαδικασιών. Για παράδειγμα, ένας λέβητας μπορεί να θεωρηθεί ως μία διαδικασία που παράγει θερμότητα ως αποτέλεσμα της ροής καυσίμου. Η διαδικασία αυτή, αποτελούμενη από υποσυστήματα όπως βαλβίδες καυσίμου και ρυθμιστές βαλβίδων καυσίμου, ρυθμίζει την θερμοκρασία ενός δωματίου ελέγχοντας την παραγωγή θερμότητας από τον λέβητα. Άλλα υποσυστήματα όπως θερμοστάτες, που δρουν ως αισθητήρες, μετρούν την θερμοκρασία του δωματίου. Στην πιο απλή μορφή, ένα σύστημα ελέγχου παράγει μια έξοδο (απόκριση) για μία δεδομένη είσοδο (διέγερση) όπως φαίνεται στο Σχ. 2.1.

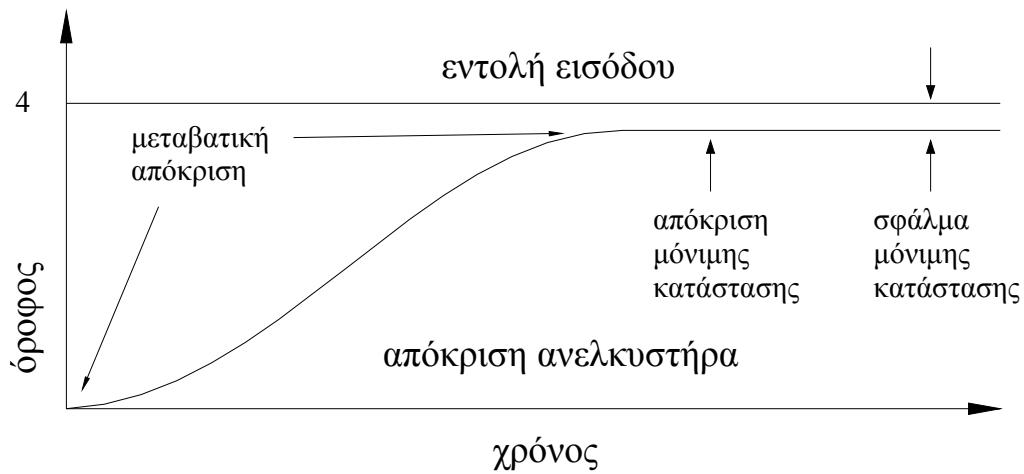


Σχ. 2.1 Απλοποιημένο διάγραμμα συστήματος ελέγχου

2.1.2 Ορισμός της Εισόδου και της Εξόδου

Η είσοδος είναι η επιθυμητή απόκριση ενώ η έξοδος είναι η πραγματική απόκριση. Για παράδειγμα, όταν το κουμπί του 4ου ορόφου ενός ανελκυστήρα πατηθεί από το ισόγειο, ο ανελκυστήρας ανεβαίνει στον 4ο όροφο με ταχύτητα και ακρίβεια ορόφου σχεδιασμένη για την άνεση των επιβατών. Το Σχ. 2.2 δείχνει την είσοδο και την έξοδο για το σύστημα του ανελκυστήρα. Το πάτημα του κουμπιού στον 4ο όροφο είναι η είσοδος και απεικονίζεται με μία βηματική εντολή. Προφανώς λαμβάνοντας υπόψη την άνεση των επιβατών (και θεωρώντας ότι υπάρχει διαθέσιμη ισχύς) δεν είναι επιθυμητό ο ανελκυστήρας να μιμηθεί ένα απότομο (βηματικό) ξεκίνημα. Η είσοδος αντιπροσωπεύει την επιθυμητή έξοδο όταν σταματήσει ο ανελκυστήρας, ο ανελκυστήρας ακολουθεί την μετακίνηση που περιγράφεται από την καμπύλη απόκρισης ανελκυστήρα. Δύο παράγοντες διαφοροποιούν την έξοδο από την είσοδο. Πρώτον, η απότομη αλλαγή στην είσοδο διαφέρει από την σταδιακή αλλαγή στην έξοδο. Οι φυσικές οντότητες δεν μπορούν να αλλάξουν κατάσταση (π.χ. θέση ή ταχύτητα) ακαριαία. Η κατάσταση αλλάζει ακολουθώντας ένα μονοπάτι που σχετίζεται με την φυσική διάταξη και κατά τη διάρκεια που ακολουθεί το μονοπάτι αυτό αποθηκεύεται ή καταναλώνεται ενέργεια. Έτσι, ο ανελκυστήρας αλλάζει κατάσταση σταδιακά καθώς ανεβαίνει από το ισόγειο στον 4ο όροφο. Αυτό το μέρος της απόκρισης καλείται μεταβατική απόκριση. Μετά από την μεταβατική απόκριση, ένα φυσικό σύστημα πλησιάζει στην απόκριση μόνιμης κατάστασης, η οποία είναι μια προσέγγιση της εντολής ή επιθυμητής απόκρισης. Για το παράδειγμα του ανελκυστήρα, αυτή η απόκριση συμβαίνει όταν ο ανελκυστήρας φτάσει στον 4ο όροφο. Η ακρίβεια της τελικής ευθυγράμμισης του ανελκυστήρα είναι ένας δεύτερος παράγοντας που διαφοροποιεί την έξοδο από την είσοδο. Αυτή η διαφορά, όπως φαίνεται στο Σχ. 2.2, καλείται σφάλμα μόνιμης κατάστασης. Η ύπαρξη σφάλματος μόνιμης κατάστασης δεν σημαίνει απαραίτητα ότι το σύστημα ελέγχου είναι

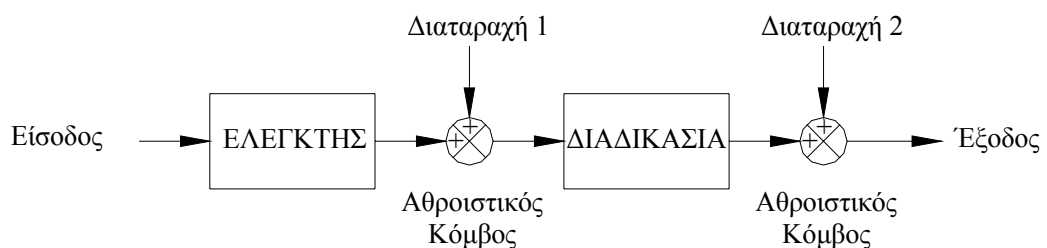
ελαπτωματικό. Πολλές φορές, τα σφάλματα μόνιμης κατάστασης είναι έμφυτα στο σχεδιαζόμενο σύστημα, και ο μηχανικός συστημάτων ελέγχου καθορίζει αν το σφάλμα μπορεί να οδηγήσει σε σημαντική δυσλειτουργία του συστήματος. Για παράδειγμα, σε ένα σύστημα που παρακολουθεί ένα δορυφόρο, ένα σφάλμα μόνιμης κατάστασης είναι ανεκτό αν είναι αρκετά μικρό ώστε ο δορυφόρος να είναι κοντά στο κέντρο του λοβού του ραντάρ.



Σχ. 2.2 Απόκριση εισόδου – εξόδου ενός συστήματος ελέγχου ανελκυστήρα

2.1.3 Συστήματα Ανοικτού Βρόχου

Ένα γενικό σύστημα ανοικτού βρόχου φαίνεται στο Σχ. 2.3. Αποτελείται από ένα υποσύστημα που καλείται ελεγκτής, ο οποίος οδηγεί μια διαδικασία ή εγκατάσταση. Η είσοδος ορισμένες φορές καλείται αναφορά (reference), ενώ η έξοδος μπορεί να καλείται ελεγχόμενη μεταβλητή (control variable). Άλλα σήματα, όπως διαταραχές, εισέρχονται στον ελεγκτή και στην έξοδο της διαδικασίας διαμέσου ενός αθροιστικού κόμβου ο οποίος δίνει το αλγεβρικό άθροισμα των όποιων σημάτων εισόδου του. Για παράδειγμα, η διαδικασία μπορεί να είναι ένας λέβητας ή σύστημα κλιματισμού όπου η μεταβλητή εξόδου είναι η θερμοκρασία. Ο ελεγκτής σε ένα σύστημα θέρμανσης αποτελείται από τις βαλβίδες καυσίμου και το ηλεκτρικό σύστημα που χειρίζεται τις βαλβίδες.



Σχ. 2.3 Σύστημα ανοικτού βρόχου

Ένα χαρακτηριστικό ενός συστήματος ανοικτού βρόχου είναι το ότι δεν μπορεί να αντισταθμίσει διαταραχές που προστίθενται στο σήμα οδήγησης (εξόδου) του ελεγκτή (Διαταραχή 1 στο Σχ. 2.3). Για παράδειγμα, αν ο ελεγκτής είναι ενισχυτής και η Διαταραχή 1 είναι θόρυβος, τότε κάθε προσθετικός θόρυβος ενισχυτή στον πρώτο αθροιστικό κόμβο (σημείο άθροισης) θα εισαχθεί επίσης και στην διαδικασία, με αποτέλεσμα η έξοδος να είναι αλλοιωμένη από την επίδραση του θορύβου. Η έξοδος στα συστήματα ανοικτού βρόχου αλλοιώνεται όχι μόνο από σήματα που προστίθενται στις εντολές του ελεγκτή αλλά και από διαταραχές στην έξοδο (Διαταραχή 2 στο Σχ. 2.3). Το σύστημα δεν μπορεί να διορθώσει ούτε αυτές τις διαταραχές.

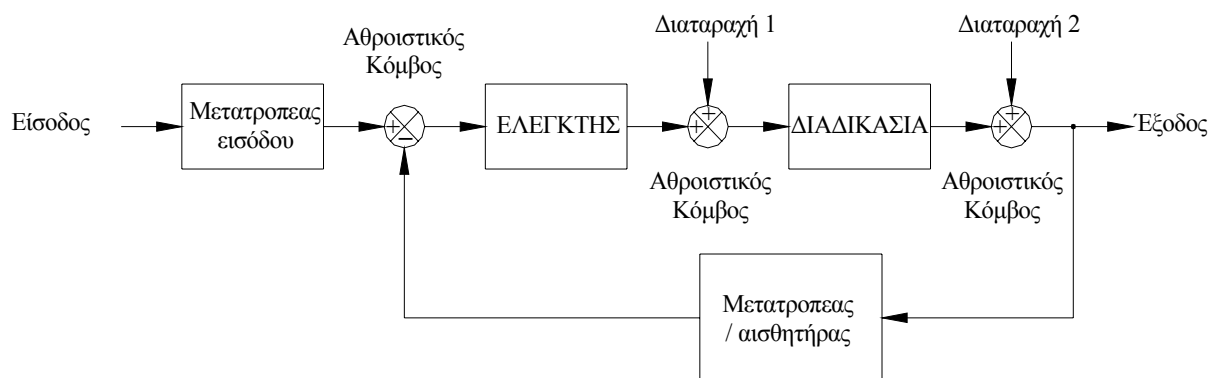
Τα συστήματα ανοικτού βρόχου, επομένως, δεν διορθώνουν διαταραχές αλλά απλώς οδηγούνται από την είσοδο. Για παράδειγμα, μια τοστιέρα είναι σύστημα ανοικτού βρόχου, αφού μπορείς να κάψεις ένα τοστ. Η ελεγχόμενη μεταβλητή (έξοδος) της τοστιέρας είναι το χρώμα του τοστ. Η συσκευή είναι σχεδιασμένη έτσι ώστε το χρώμα να είναι βαθύτερο όσο περισσότερο το τοστ είναι στην τοστιέρα. Η τοστιέρα δεν μετρά το χρώμα του τοστ, δεν διορθώνει το γεγονός ότι το τοστ μπορεί να είναι σικάλεως, λευκό, ή ζυμωτό ούτε το γεγονός ότι το τοστ μπορεί να είναι διαφόρων παχών.

Άλλα αποτελέσματα συστημάτων ανοικτού βρόχου είναι μηχανικά συστήματα αποτελούμενα από μάζα, ελατήριο, και σύστημα απόσβεσης με σταθερή δύναμη στη μάζα. Όσο μεγαλύτερη η δύναμη, τόσο μεγαλύτερη η μετατόπιση. Και πάλι, η θέση του συστήματος θα αλλάξει με μια διαταραχή όπως μια επιπλέον δύναμη, και το σύστημα δεν θα ανιχνεύσει ούτε θα διορθώσει την διαταραχή. Σε ένα άλλο παράδειγμα ας υποθέσουμε ότι υπολογίζεις το χρόνο που απαιτείται να μελετήσεις για μια εξέταση που καλύπτει τρία κεφάλαια με σκοπό να πάρεις άριστα. Αν ο καθηγητής προσθέσει ένα τέταρτο κεφάλαιο (διαταραχή) τότε είσαι ένα σύστημα ανοικτού βρόχου αν δεν εντοπίσεις την διαταραχή (την αύξηση της ύλης) και δεν προσθέσεις χρόνο μελέτης στον προηγούμενο υπολογισμένο χρόνο. Το αποτέλεσμα της παράβλεψης θα είναι ένας βαθμός χαμηλότερος του αναμενόμενου.

Πως μπορεί να βελτιωθεί ένα σύστημα ανοικτού βρόχου;

2.1.4 Συστήματα Κλειστού Βρόχου

Τα μειονεκτήματα των συστημάτων ανοικτού βρόχου, δηλαδή, η ευαισθησία στις διαταραχές και η αδυναμία διόρθωσης των διαταραχών αυτών, μπορούν να ξεπεραστούν στα συστήματα κλειστού βρόχου. Η γενική αρχιτεκτονική ενός συστήματος κλειστού βρόχου φαίνεται στο Σχ. 2.4. Ο μετατροπέας εισόδου, ο οποίος έχει συμπεριληφθεί στον ελεγκτή του συστήματος ανοικτού βρόχου μετατρέπει την μορφή της εισόδου στη μορφή που χρησιμοποιεί ο ελεγκτής. Ένας μετατροπέας εξόδου, ή ένας αισθητήρας, μετράει την απόκριση εξόδου και μετατρέπει επίσης την απόκριση αυτή στη μορφή που χρησιμοποιεί ο ελεγκτής. Για παράδειγμα, αν ο ελεγκτής χρησιμοποιεί ηλεκτρικά σήματα για να χειρίζεται τις βαλβίδες ενός συστήματος ελέγχου θερμοκρασίας, η θέση εισόδου και θερμοκρασία εξόδου μετατρέπονται σε ηλεκτρικά σήματα. Η θέση εισόδου μπορεί να μετατραπεί σε τάση με τη χρήση ενός ποτενσιόμετρου, μιας μεταβλητής αντίστασης, και η θερμοκρασία εξόδου μπορεί να μετατραπεί σε τάση με ένα θερμίστορ, μια συσκευή που αλλάζει την ηλεκτρική της αντίσταση με την θερμοκρασία.



Σχ. 2.4 Σύστημα κλειστού βρόχου

Το πρώτο σημείο άθροισης αθροίζει αλγεβρικά το σήμα από την είσοδο με το σήμα από την έξοδο, το οποίο έρχεται από μία γραμμή ανάδρασης ή γραμμή επιστροφής από την έξοδο στο σημείο άθροισης. Το σήμα εξόδου αφαιρείται από το σήμα εισόδου. Το αποτέλεσμα γενικά καλείται σήμα ενεργοποίησης. Ωστόσο, στα συστήματα όπου οι μετατροπείς εισόδου και εξόδου έχουν μοναδιαίο κέρδος (δηλαδή, οι μετατροπείς ενισχύουν την είσοδο κατά 1), η τιμή του σήματος ενεργοποίησης είναι ίση με τη διαφορά ανάμεσα σε είσοδο και έξοδο. Υπό αυτό τον όρο, το σήμα ενεργοποίησης καλείται

σφάλμα (error). Ένα σύστημα κλειστού βρόχου προσπαθεί να αντισταθμίσει τις διαταραχές μετρώντας την απόκριση εξόδου, και στέλνοντας την μέτρηση πίσω από τη γραμμή ανάδρασης, και συγκρίνοντας την με την είσοδο στο σημείο άθροισης. Αν υπάρχει κάποια διαφορά ανάμεσα στις αποκρίσεις, το σύστημα οδηγεί την διαδικασία, διαμέσου του σήματος ενεργοποίησης, για να κάνει τη διόρθωση. Αν δεν υπάρχει διαφορά, το σύστημα δεν οδηγεί την εγκατάσταση, αφού η απόκριση της εγκατάστασης είναι η επιθυμητή.

Τα συστήματα κλειστού βρόχου, επομένως, έχουν προφανές πλεονέκτημα και μεγαλύτερη ακρίβεια από τα συστήματα ανοικτού βρόχου. Είναι λιγότερο ευαίσθητα στο θόρυβο και γενικά σε διαταραχές. Η μεταβατική κατάσταση και το σφάλμα μόνιμης κατάστασης ελέγχονται ευκολότερα και πιο ευέλικτα στα συστήματα κλειστού βρόχου, συχνά με μια απλή προσαρμογή των κερδών στο βρόχο, και μερικές φορές με την επανασχεδίαση του ελεγκτή. Αναφερόμαστε στην επανασχεδίαση ως «αντιστάθμιση» του συστήματος και στο αντίστοιχο “hardware” ως «το αποτέλεσμα αντισταθμιστή». Από την άλλη πλευρά, τα συστήματα κλειστού βρόχου είναι πιο πολύπλοκα και ακριβότερα από τα συστήματα ανοικτού βρόχου. Μια κοινή τοστιέρα, ανοικτού βρόχου, για παράδειγμα είναι απλή και φτηνή. Μια τοστιέρα - φούρνος κλειστού βρόχου είναι πιο πολύπλοκη και ακριβότερη αφού πρέπει να μετράει το χρώμα (με ανάκλαση φωτός) και την υγρασία μέσα της. Επομένως, ο μηχανικός συστημάτων ελέγχου πρέπει να επιλέξει ανάμεσα στην απλότητα και το χαμηλό κόστος ενός συστήματος ανοικτού βρόχου και την ακρίβεια και το υψηλό κόστος ενός συστήματος κλειστού βρόχου.

Συνοψίζοντας, τα συστήματα που κάνουν την μέτρηση και τη διόρθωση που περιγράψαμε παραπάνω καλούνται κλειστού βρόχου ή συστήματα ελέγχου με ανάδραση. Τα συστήματα που δεν έχουν την ιδιότητα της μέτρησης και διόρθωσης καλούνται συστήματα ανοικτού βρόχου.

2.2 Το πρόβλημα του ελέγχου διέγερσης σύγχρονης γεννήτριας.

Το πρόβλημα της δυναμικής ευστάθειας των ενεργειακών συστημάτων τυγχάνει της ολοένα και αυξανόμενης προσοχής των ερευνητών κατά τη διάρκεια των τελευταίων ετών [1-3]. Οι κύριοι λόγοι γι’ αυτό είναι ο αυξανόμενος αριθμός των ηλεκτροπαραγωγών μονάδων που εγκαθίστανται στους σταθμούς παραγωγής ηλεκτρικής ενέργειας και η χρήση συστημάτων διέγερσης υψηλών ταχυτήτων απόκρισης. Η επίδραση τέτοιων συστημάτων ταχείας απόκρισης στη δυναμική ευστάθεια, συνίσταται στην εφαρμογή αρνητικής απόσβεσης στο υπό έλεγχο σύστημα και κατά συνέπεια στην εμφάνιση ασθενών ταλαντώσεων από αυτό. Η σχεδίαση ενός τέτοιου συστήματος διέγερσης θα πρέπει να εξασφαλίζει ικανοποιητική συμπεριφορά σε μια ευρεία περιοχή λειτουργικών συνθηκών, καθώς επίσης και σε μια ευρεία περιοχή συνθηκών σφαλμάτων.

Η ανασκόπηση στη σχετική διεθνή βιβλιογραφία παρουσιάζει ποικίλες μεθόδους για την αντιμετώπιση του προβλήματος του σχεδιασμού κατάλληλου ελεγκτή διέγερσης. Πρακτικοί μέθοδοι μη γραμμικού ελέγχου περιλαμβάνουν ένα ανάστροφο μοντέλο ανοικτού βρόχου για τη δυναμική του (μη γραμμικού) ενεργειακού συστήματος και τη χρήση βρόχων ανάδρασης προκειμένου να αναστέλλονται αυτές οι μη γραμμικότητες. Η προσέγγιση ενός μη γραμμικού συστήματος με ένα γραμμικοποιημένο μοντέλο, οδηγεί στην εφαρμογή μεθόδων προσαρμοστικού ελέγχου, όπου χρησιμοποιούνται μετρήσεις των μεταβλητών εισόδου του συστήματος σε πραγματικό χρόνο, είτε για την άμεση εξαγωγή του μοντέλου του συστήματος και εν συνεχεία για το σχεδιασμό ενός ελεγκτή βασισμένου στο μοντέλο αυτό (έμμεσος προσαρμοστικός έλεγχος), είτε στην άμεση προσαρμογή των μεταβλητών εξόδου του ελεγκτή (άμεσος προσαρμοστικός έλεγχος). Κατά το παρελθόν, έχουν παρουσιασθεί τέτοιες μελέτες αναφορικά με εφαρμογές αλγεβρικών μεθόδων και μεθόδων βέλτιστου ελέγχου στη σχεδίαση ελεγκτών διέγερσης π.χ.[4-5].

Από την άλλη πλευρά, έχουν εμφανισθεί και εφαρμοστεί μέθοδοι “ευφυούς” ελέγχου κάνοντας χρήση τεχνικών που βασίζονται στην ασαφή λογική, τα νευρωνικά δίκτυα και τους γενετικούς αλγόριθμους. Οι τεχνικές αυτές βρίσκουν έδαφος - και αντιμετωπίζουν ικανοποιητικά- περιπτώσεις όπου κάποιες από τις σχέσεις που περιγράφουν το υπό έλεγχο σύστημα μπορούν να αναπαρασταθούν με ασαφή σύνολα και ασαφείς σχεσιακές εξισώσεις, ή ακόμη δεν μπορούν να περιγραφούν

πλήρως αλλά να μοντελοποιηθούν με ένα κατάλληλα σχεδιασμένο νευρωνικό δίκτυο που να “συλλαμβάνει” τις μη γραμμικότητες που ισχύουν. Ουσιαστικά, οι ασαφείς ελεγκτές αποτελούν μορφή γνωσιολογικών βάσεων δεδομένων οι οποίες περιλαμβάνουν ασαφείς συναρτήσεις συμμετοχής των εμπλεκόμενων μεταβλητών, προκειμένου να “ενσωματωθεί” η ανθρώπινη τεχνολογία στη βάση αυτή. Κατά το παρελθόν, έχουν παρουσιασθεί μελέτες αναφορικά με εφαρμογές τέτοιων τεχνικών και μεθόδων στον έλεγχο διέγερσης σύγχρονων γεννητριών [6-10].

Μία επιπλέον κατηγορία μεθόδων και τεχνικών ελέγχου αποτελούν τα έμπειρα συστήματα και οι ευρετικές τεχνικές (heuristics). Σε συνδυασμό με την εξέλιξη που έχει παρατηρηθεί τα τελευταία χρόνια στη μικροηλεκτρονική και τις αυξημένες δυνατότητες που προσφέρουν -σε επίπεδο υπολογιστικής ισχύος- οι σύγχρονοι μικροεπεξεργαστές, έχει δημιουργηθεί μεγάλο πεδίο εφαρμογών και για τη σχεδίαση συστημάτων ελέγχου διέγερσης [11-14].

Με αφορμή τα παραπάνω, η παρούσα εργασία ασχολείται μία τέτοια «ευφυή» τεχνική, και συγκεκριμένα αυτή των τεχνητών νευρωνικών δικτύων (Τ.Ν.Δ.).

2.3 Μοντέλο σύγχρονης γεννήτριας στο πλαίσιο αναφοράς του δρομέα

Σκοπός της ενότητας αυτής είναι η εισαγωγή ενός λεπτομερούς μοντέλου σύγχρονης μηχανής. Επειδή το μοντέλο είναι ευρέως γνωστό από το 1930, θα παρουσιαστούν μόνο οι κύριες υποθέσεις και τα αποτελέσματά του.

2.3.1 Υποθέσεις για την ανάπτυξη του μοντέλου.

Μια τριφασική σύγχρονη γεννήτρια έχει τρία όμοια τυλίγματα οπλισμού συμμετρικά κατανεμημένα γύρω από το διάκενο αέρος, και ένα τύλιγμα πεδίου διέγερσης. Μπορεί να υπάρχει επίσης ένα ή περισσότερα τυλίγματα απόσβεσης, για λόγους ευκολίας στην παρούσα ενότητα, θα υποθέσουμε ότι υπάρχει ένα τύλιγμα απόσβεσης σε κάθε άξονα (d,q) της μηχανής. Συνήθως τα τυλίγματα οπλισμού τοποθετούνται στο στάτη, και τα τυλίγματα πεδίου και απόσβεσης στον δρομέα. Ωστόσο, υπάρχουν περιπτώσεις που, όπου τα τυλίγματα οπλισμού τοποθετούνται στο δρομέα και τα τυλίγματα πεδίου στο στάτη. Το γεγονός αυτό δεν επηρεάζει καθόλου την προσέγγιση της μοντελοποίησης της μηχανής, αφού μόνο η σχετική κίνηση ανάμεσα στο στάτη και το δρομέα είναι σημαντική. Επομένως, στο παρόν κείμενο, όταν αναφερόμαστε σε «τυλίγματα δρομέα», θα εννοούμε πάντα τα τυλίγματα πεδίου (και απόσβεσης, αν υπάρχουν) τοποθετημένα στην απέναντι πλευρά του διακένου σε σχέση με τα τριφασικά τυλίγματα οπλισμού.

Για να απλοποιηθεί η πραγματική σύγχρονη μηχανή και να καταστεί δυνατή η μοντελοποίησή της με σχετική ευκολία, γίνονται οι ακόλουθες υποθέσεις [16].

1. Υποθέτουμε ότι κάθε τύλιγμα που υπάρχει στη μηχανή παράγει μία ημιτονοειδή μαγνητηγενετική δύναμη (ΜΕΔ) κατά μήκος του διακένου αέρος, η οποία, (για τη φάση a), μπορεί να εκφραστεί ως $\theta_a = \Theta_a \sin\left(\frac{P}{2} \omega_s\right)$, όπου το P αντιπροσωπεύει τον αριθμό των πόλων της μηχανής, και το ω_s συμβολίζει την γωνιακή ταχύτητα του στρεφόμενου μαγνητικού πεδίου του στάτη.

2. Η μαγνητική διαπερατότητα του σιδήρου της μηχανής θεωρείται άπειρη. Αυτό ισοδυναμεί με το να αγνοηθούν όλα τα φαινόμενα μαγνητικού κορεσμού, και της οριακής διαδρομής της μαγνητικής ροής.

3. Η κατασκευή του δρομέα θεωρείται ότι είναι ο μόνος παράγοντας που συμβάλει στην μαγνητική ασυμμετρία της μηχανής. Η επίδραση των εγκοπών του δρομέα ή του στάτη μπορούν να ληφθούν υπόψη με το συντελεστή του Carter. Αυτή η

υπόθεση έχει ως αποτέλεσμα την προσέγγιση της συνάρτησης της μαγνητικής αγωγιμότητας ως $\lambda = \lambda_0 - \lambda_2 \cos(P\omega_s)$, όπου τα λ_0 και λ_2 εξαρτώνται από τη γεωμετρία του διακένου.

4. Η τιμή της μαγνητικής επαγωγής B (ένταση της μαγνητικής ροής), υπολογίζεται με τον πολλαπλασιασμό της μαγνητεγερτικής δύναμης (ΜΕΔ) επί την μαγνητική διαπερατότητα. Η τρίτη αρμονική που προκύπτει από τον πολλαπλασιασμό αυτό αγνοείται, σύμφωνα με την υπόθεση 1.

Τα σφάλματα που εισάγονται από αυτές τις υποθέσεις είναι συνήθως αρκετά μικρά ώστε να αγνοούνται, ειδικά από την σκοπιά της δυναμικής απόδοσης της μηχανής.

2.3.2 Ανάπτυξη των εξισώσεων του μοντέλου και των ισοδύναμων κυκλωμάτων.

Μία σύγχρονη μηχανή μπορεί να περιγραφεί από ένα σύστημα $n + 1$ εξισώσεων, n από τις οποίες είναι ηλεκτρικές και μία μηχανική. Ο αριθμός n των ηλεκτρικών εξισώσεων είναι ίσος με τον αριθμό των ανεξάρτητων ηλεκτρικών μεταβλητών που απαιτούνται για να περιγράψουν τη μηχανή. Οι μεταβλητές αυτές μπορούν να είναι είτε τα ρεύματα είτε οι κλειστές διαδρομές της μαγνητικής ροής. Στην παρούσα εργασία επιλέγονται τα ρεύματα ως ανεξάρτητες μεταβλητές. Οι ηλεκτρικές εξισώσεις προκύπτουν από την εφαρμογή του νόμου τάσεων του Kirchhoff σε κάθε τύλιγμα, δηλαδή εξισώνοντας την τάση τερματισμού του τυλίγματος με το άθροισμα των ωμικών και επαγωγικών πτώσεων τάσης κατά μήκος του τυλίγματος. Έτσι, η συνολική τάση κατά μήκος του ισούται με μηδέν. Για να υπολογιστεί ορθά η επαγωγική πώση τάσης κατά μήκος ενός τυλίγματος, πρέπει να αποτιμηθεί η συνολική συζευγμένη μαγνητική ροή με το τύλιγμα. Αυτό επιτυγχάνεται με τη χρήση ενός πίνακα επαγωγών, οποίος συσχετίζει όλες τις μαγνητικές ροές όλων των τυλιγμάτων με όλα τα ρεύματα όλων των τυλιγμάτων. Στη συνέχεια, για μια σύγχρονη γεννήτρια με δρομέα έκτυπων πόλων, προκύπτει ένας πίνακας που εξαρτάται από τη θέση του δρομέα. Αυτή η εξάρτηση οφείλεται στην μαγνητική ασυμμετρία του δρομέα, γιατί το σχήμα ενός προεξέχοντα πόλου μιας μηχανής ευνοεί μια συγκεκριμένη μαγνητική διεύθυνση. Η διεύθυνση αυτή συμπίπτει με την διεύθυνση της ροής που παράγει το τύλιγμα, και ορίζεται ως ο άξονας της μηχανής " d ". Ο άξονας της μηχανής " q " τοποθετείται στις 90 ηλεκτρικές μοίρες (με αντί-ωρολογιακή φορά) σε σχέση με τον άξονα της μηχανής " d ". Έτσι, η θέση του δρομέα μπορεί να εκφραστεί ως μία γωνία θ , μεταξύ του μαγνητικού άξονα της φάσης a του σπλισμού και του άξονα " q " του δρομέα. Η εξάρτηση του πίνακα επαγωγών από την θέση του δρομέα αντιπροσωπεύει την κύρια δυσκολία στη μοντελοποίηση της σύγχρονης μηχανής. Μια λύση στο πρόβλημα αυτό είναι η αλλαγή συστήματος αναφοράς, (ή πλαίσιο αναφοράς), στο οποίο εκφράζονται οι ηλεκτρικές και μαγνητικές μεταβλητές της μηχανής. Στο παρελθόν, το σύστημα αναφοράς που χρησιμοποιούταν ήταν το ονομαζόμενο στατικό (stationary), ή πλαίσιο αναφοράς του στάτη, ή πλαίσιο αναφοράς " $a - b - c$ ". Σε αυτό, οι μεταβλητές εκφράζονται σαν πραγματικά να μετρούνται στην μηχανή, αλλά οι παράμετροι της μηχανής είναι χρονικά μεταβαλλόμενες (αφού η θ είναι συνάρτηση του χρόνου). Αποδεικνύεται ότι το μόνο πλαίσιο αναφοράς το οποίο παρέχει σταθερές (ανεξάρτητες του χρόνου) παραμέτρους της μηχανής είναι το πλαίσιο αναφοράς " $d - q$ " ή αλλιώς πλαίσιο αναφοράς του δρομέα. Σε αυτό, όλες οι μεταβλητές εκφράζονται σε μια μορφή στην οποία ένας υποθετικός παρατηρητής τοποθετημένος στο δρομέα θα μετρούσε. Ο μετασχηματισμός από το " $a - b - c$ " στο " $d - q$ " σύστημα αναφοράς δίνεται από τον ακόλουθο πίνακα μετασχηματισμού:

$$T = \sqrt{\frac{2}{3}} \begin{bmatrix} \sin \theta & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta + \frac{2\pi}{3}\right) \\ \cos \theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \end{bmatrix} \quad (2.1)$$

Ο αντίστροφος μετασχηματισμός (από το "d - q" στο "a - b - c" σύστημα αναφοράς) δίνεται από

$$T_{inv} = \sqrt{\frac{2}{3}} \begin{bmatrix} \sin \theta & \cos \theta \\ \sin\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) \\ \sin\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \end{bmatrix} \quad (2.2)$$

Στις Εξ. (2.1) και (2.2) το θ υπολογίζεται ως

$$\theta(t) = \int_0^t \omega(\xi) d\xi + \theta_0 \quad (2.3)$$

Όπου το ω αντιπροσωπεύει την (ηλεκτρική) ταχύτητα του δρομέα.

Συνεπώς, οποιαδήποτε ομάδα τριφασικών μεταβλητών f_a , f_b και f_c εκφρασμένες στο "abc" σύστημα αναφοράς μπορούν να μετασχηματιστούν στις μεταβλητές f_d και f_q στο "d - q" σύστημα αναφοράς πολλαπλασιάζοντας τις με T :

$$\begin{bmatrix} f_d \\ f_q \end{bmatrix} = T \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} \quad (2.4)$$

και αντιστρόφως:

$$\begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} = T_{inv} \begin{bmatrix} f_d \\ f_q \end{bmatrix} \quad (2.5)$$

Σημειώνεται ότι ο μετασχηματισμός των μεταβλητών, που ορίζεται από τις Εξ. (2.1) και (2.2), διατηρεί την συνολική ισχύ του συστήματος σε κάθε χρονική στιγμή, δηλ. η ισχύς στο "a - b - c" σύστημα αναφοράς ισούται με την ισχύ στο "d - q" σύστημα αναφοράς.

Όταν οι ηλεκτρικές εξισώσεις της μηχανής μετασχηματίζονται από το "a - b - c" στο "d - q" σύστημα αναφοράς, παίρνουμε τις ακόλουθες σχέσεις:

- Εξισώσεις τάσεων στον οπλισμό

$$v_d = -R_s i_d - \omega \lambda_q - (L_{ls} + L_{md}) \frac{di_d}{dt} + L_{md} \frac{di_{fd}}{dt} + L_{md} \frac{di_{kd}}{dt} \quad (2.6)$$

$$v_q = -R_s i_q - \omega \lambda_d - (L_{ls} + L_{mq}) \frac{di_q}{dt} + L_{mq} \frac{di_{kq}}{dt} \quad (2.7)$$

όπου

$$\lambda_d = -(L_{ls} + L_{md}) i_d + L_{md} (i_{fd} + i_{kd}) \quad (2.8)$$

$$\lambda_q = -(L_{ls} + L_{mq}) i_q + L_{mq} i_{kq} \quad (2.9)$$

- Εξισώσεις πεδίου τάσης διέγερσης

$$v_{fd} = R_{fd} i_{fd} - L_{md} \frac{di_d}{dt} + (L_{lfd} + L_{md}) \frac{di_{fd}}{dt} + L_{md} \frac{di_{kd}}{dt} \quad (2.10)$$

- Εξισώσεις πτώσεων τάσεων στα τυλίγματα απόσβεσης

$$0 = R_{kd} i_{kd} - L_{md} \frac{di_d}{dt} + L_{md} \frac{di_{fd}}{dt} + (L_{lkd} + L_{md}) \frac{di_{kd}}{dt} \quad (2.11)$$

$$0 = R_{kq} i_{kq} - L_{mq} \frac{di_q}{dt} + (L_{lkq} + L_{mq}) \frac{di_{kq}}{dt} \quad (2.12)$$

Όπου:

ω : ταχύτητα δρομέα (γωνιακή)

v_d : τάση εξόδου στον άξονα "d" του οπλισμού

v_q : τάση εξόδου στον άξονα "q" του οπλισμού

i_d : ρεύμα γραμμής στον άξονα "d" του οπλισμού

i_q : ρεύμα γραμμής στον άξονα "q" του οπλισμού

v_{fd} : τάση των τυλιγμάτων πεδίου διέγερσης (ανηγγμένη στο στάτη)

i_{fd} : ρεύμα των τυλιγμάτων πεδίου (ανηγγμένη στο στάτη)

i_{kd} : ρεύμα των τυλιγμάτων απόσβεσης στον άξονα "d" (ανηγγμένη στο στάτη)

i_{kq} : ρεύμα των τυλιγμάτων απόσβεσης στον άξονα "q" (ανηγγμένη στο στάτη)

λ_d : συνολική (μαγνητική) ροή οπλισμού στον άξονα "d"

λ_q : συνολική ροή οπλισμού στον άξονα "q"

R_s : φασική αντίσταση οπλισμού

L_{ls} : φασική επαγωγή σκέδασης οπλισμού

L_{md} : συζευγμένη επαγωγή στον άξονα "d"

R_{fd} : αντίσταση τυλιγμάτων πεδίου (ανηγγμένη στο στάτη)

L_{yfd} : επαγωγή σκέδασης τυλιγμάτων πεδίου (ανηγμένη στο στάτη)

R_{kd} : αντίσταση τυλιγμάτων απόσβεσης στον άξονα (ανηγμένη στο στάτη)

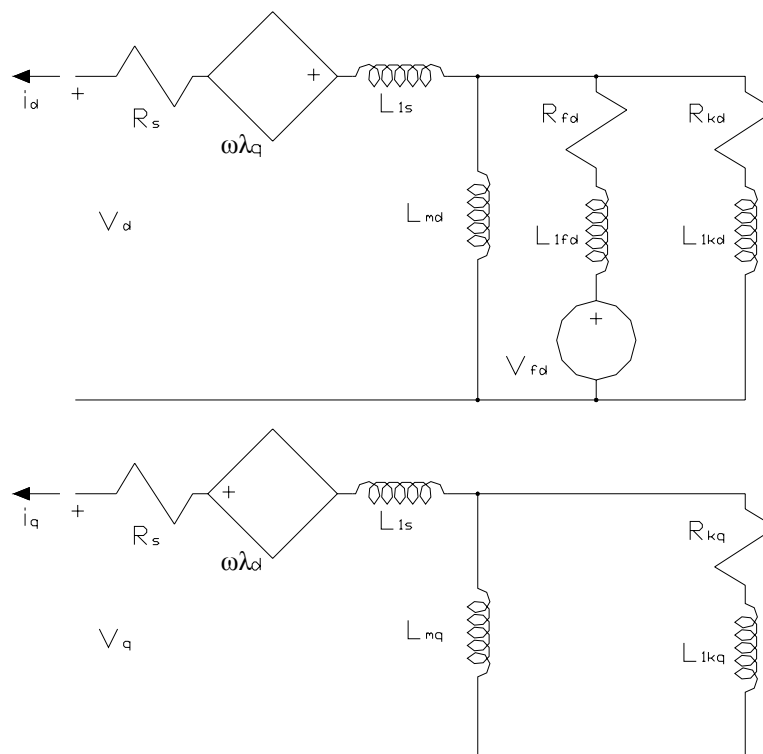
L_{lkd} : επαγωγή διαρροής τυλιγμάτων απόσβεσης στον άξονα (ανηγμένη στο στάτη)

L_{mq} : συζευγμένη επαγωγή στον άξονα

R_{kq} : αντίσταση τυλιγμάτων απόσβεσης στον άξονα (ανηγμένη στο στάτη)

L_{lkq} : επαγωγή σκέδασης τυλιγμάτων απόσβεσης στον άξονα "q" (ανηγμένη στο στάτη)

Οι εξισώσεις (2.6) – (2.16) περιγράφουν το ισοδύναμο κύκλωμα της σύγχρονης γεννήτριας στο πλαίσιο αναφοράς του δρομέα, το οποίο φαίνεται στο Σχ. 2.5.



Σχ. 2.5 Ισοδύναμο κύκλωμα σύγχρονης γεννήτριας στο πλαίσιο αναφοράς του δρομέα

Μπορούν να γίνουν κάποιες παρατηρήσεις σχετικά με το ισοδύναμο κύκλωμα:

- Τα ισοδύναμα κυκλώματα του "d" και "q" άξονα είναι παρόμοια με το ισοδύναμο κύκλωμα ενός μετασχηματιστή: στο καθένα από αυτά, διάφορα τυλίγματα, που το καθένα χαρακτηρίζεται από κάποια αντίσταση και επαγωγή σκέδασης, βρίσκονται συζευγμένα μέσω μιας αμοιβαίας επαγωγής. Η διαφορά, σε σχέση με την περίπτωση του μετασχηματιστή, έγκειται στο ότι, ενώ το ισοδύναμο κύκλωμα του μετασχηματιστή είναι κύκλωμα ac (εναλλασσόμενου ρεύματος), εδώ, όταν η γεννήτρια βρίσκεται σε ημιτονοειδή μόνιμη κατάσταση, όλες οι τάσεις, τα ρεύματα και οι μαγνητικές ροές είναι συνεχείς dc .

- Παρότι τα τυλίγματα οπλισμού εκφράζονται στο πλαίσιο αναφοράς του δρομέα, και ότι δεν υπάρχουν χρονικά μεταβαλλόμενες επαγωγές, το γεγονός ότι τα τυλίγματα οπλισμού είναι μαγνητικά συζευγμένα λαμβάνεται υπόψη με το να υπάρχουν συζευκτικοί όροι στους κλάδους οπλισμού στα ισοδύναμα κυκλώματα των αξόνων " d " και " q ". Για κάθε άξονα, ο όρος αυτός ισούται με το γινόμενο της ταχύτητας του δρομέα και ροής συζευγμένης με τα τυλίγματα οπλισμού του άλλου άξονα.

- Αν μία μηχανή (όπως π.χ. μία διεγέρτρια γεννήτρια) δεν έχει τυλίγματα απόσβεσης, το ισοδύναμο κύκλωμα μπορεί εύκολα να προσαρμοστεί αφαιρώντας από αυτό τους κλάδους που αναφέρονται στα τυλίγματα απόσβεσης. Το υπόλοιπο κύκλωμα παραμένει ως έχει.

- Όλες οι παράμετροι του δρομέα ανάγονται στον οπλισμό. Επομένως, όταν το κύκλωμα χρησιμοποιείται για προσομοίωση, και ενδιαφέρουν οι πραγματικές τιμές των μεταβλητών του δρομέα, πρέπει να ληφθεί υπόψη ο λόγος των σπειρών των τυλιγμάτων ανάμεσα στο δρομέα και τον οπλισμό.

Το ισοδύναμο κύκλωμα περιγράφει ηλεκτρικά μια σύγχρονη γεννήτρια. Η μηχανική μεταβλητή αναπαρίσταται από την (γωνιακή) ταχύτητα του δρομέα ω , και απαιτείται η μηχανική εξίσωση για την ολοκλήρωση του μοντέλου. Η εξίσωση αυτή συσχετίζει την εξωτερική ροπή που εφαρμόζεται στον άξονα της γεννήτριας με την ηλεκτρομαγνητική ροπή που αναπτύσσει εσωτερικά η μηχανή. Ωστόσο, για τις ανάγκες της παρούσας εργασίας, δεν συμπεριλαμβάνεται η μηχανική εξίσωση του συστήματος, δηλαδή η ταχύτητα του δρομέα θεωρείται γνωστή. Ο λόγος για αυτό είναι το γεγονός ότι το ενδιαφέρον μας είναι πρωτίστως η περιγραφή της ηλεκτρικής συμπεριφοράς της γεννήτριας η οποία τροφοδοτεί φορτίο dc δια μέσου ενός ανορθωτή. Έτσι, είναι λογικό να θεωρηθεί σταθερή ταχύτητα, αφού τα ηλεκτρικά μεταβατικά φαινόμενα στη μηχανή μπορούν να θεωρηθούν πολύ ταχύτερα από τα αντίστοιχα μηχανικά (που περιλαμβάνουν τη δυναμική της αδράνειας της κινητήριας μηχανής και της αδράνειας του δρομέα της γεννήτριας).

Υπό τις παραπάνω θεωρήσεις, η ταχύτητα του δρομέα ω δεν είναι μεταβλητή, αλλά παράμετρος του συστήματος. Αυτό καθιστά τις εξισώσεις (2.6 – 2.12) ένα σύστημα από γραμμικές διαφορικές εξισώσεις.

2.3.3 Ημιτονοειδής μόνιμη κατάσταση.

Εάν μια γεννήτρια λειτουργεί σε ημιτονοειδή μόνιμη κατάσταση, οι φασικές τάσεις μπορούν να εκφραστούν στο σύστημα αναφοράς " $a - b - c$ " ως εξής

$$v_a = V_p \cos \theta_v \quad (2.13)$$

$$v_b = V_p \cos \left(\theta_v - \frac{2\pi}{3} \right) \quad (2.14)$$

$$v_c = V_p \cos \left(\theta_v + \frac{2\pi}{3} \right) \quad (2.15)$$

όπου

$$\theta_v = \omega_v t + \theta_{v0} \quad (2.16)$$

και V_p το πλάτος της φασικής τάσης.

Στη μόνιμη κατάσταση, προφανώς, ισχύει $\omega_v = \omega$. Ωστόσο, για να φτάσει στη μόνιμη κατάσταση, η γεννήτρια πρέπει να περάσει από μια μεταβατική κατάσταση, κατά τη διάρκεια της οποίας η ηλεκτρική ταχύτητα του δρομέα μπορεί να είναι διαφορετική από την γωνιακή συχνότητα της τάσης εξόδου. Αυτή η μεταβατική φάση δίνει την δυνατότητα στη μηχανή να φτάσει στη μόνιμη κατάσταση της γωνίας του δρομέα δ , που ορίζεται ως η μετατόπιση του δρομέα με αναφορά στη μέγιστη δυνατή τιμή του θεμελιώδους στοιχείου της τάσης εξόδου της φάσης a . Μπορεί να εκφραστεί σε ακτίνια ως εξής

$$\delta = \theta - \theta_v = \int_0^t (\omega(\xi) - \omega_v(\xi)) d\xi + \theta(0) - \theta_v(0) \quad (2.17)$$

Στη συνέχεια, αφού εφαρμοστεί η Εξ. (2.1) στις (2.13) – (2.15), στο σύστημα αναφοράς " $d - q$ " προκύπτει ότι:

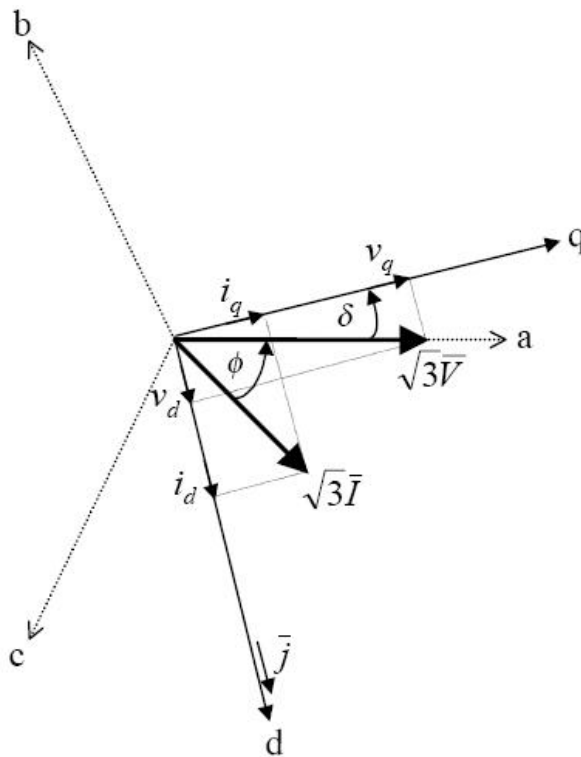
$$v_d = \sqrt{\frac{3}{2}} V_p \sin(\theta - \theta_v) = \sqrt{\frac{3}{2}} V_p \sin \delta \quad (2.18)$$

$$v_q = \sqrt{\frac{3}{2}} V_p \cos(\theta - \theta_v) = \sqrt{\frac{3}{2}} V_p \cos \delta \quad (2.19)$$

Οι Εξ. (2.18) και (2.19) φανερώνουν ότι, για να είναι δυνατή η αναπαράσταση των μεταβλητών του συστήματος αναφοράς " dq " και των διανυσμάτων του χώρου ac στο ίδιο διάγραμμα, το « φυσικό » επίπεδο " dq " της μηχανής να συσχετιστεί με ένα μιγαδικό (Gaussian) επίπεδο. Ο πραγματικός άξονας του μιγαδικού επιπέδου συμπίπτει με τον άξονα " q " της μηχανής, και ο φανταστικός άξονας του μιγαδικού επιπέδου συμπίπτει με τον άξονα " d " της μηχανής. (Σχ 2.6) Έτσι μπορεί να γραφτεί

$$\bar{V} = \frac{1}{\sqrt{3}} (v_q + j\bar{v}_d) \quad (2.20)$$

Στη σχέση (2.20), το \bar{V} αναπαριστά το διάνυσμα χώρου που αντιστοιχεί στις τάσεις v_a , v_b και v_c που ορίζονται από τις (2.13 – 2.15), και το j αντιπροσωπεύει το μοναδιαίο διάνυσμα στον φανταστικό άξονα. Το μέτρο ενός διανύσματος χώρου μιας μεταβλητής ισούται την *rms* τιμή της μεταβλητής. Το Σχ. 2.6 δείχνει την θέση του συστήματος αναφοράς " $d - q$ " και του διανύσματος χώρου $\sqrt{3}\bar{V}$ την χρονική στιγμή $t = 0$. Το διάνυσμα $\sqrt{3}\bar{V}$ και το σύστημα αναφοράς " dq " περιστρέφονται αντιωρολογιακά με σταθερή ταχύτητα ω , και οι στιγμιαίες τιμές των v_a , v_b και v_c προκύπτουν ως οι προβολές του $\sqrt{2}\bar{V}$ στους άξονες a , b και c , οι οποίοι είναι ακίνητοι.



Σχ. 2.6 Διανυσματικό διάγραμμα στο χώρο για σταθερή ημιτονοειδή κατάσταση

Το διάνυσμα $\sqrt{3}\bar{I}$, που αντιπροσωπεύει τα φασικά ρεύματα, είναι μετατοπισμένο κατά τη γωνία φορτίου ϕ σε σχέση με το \bar{V} . Τα στιγμιαία ρεύματα φάσεων στο σύστημα αναφοράς abc προκύπτουν ως οι προβολές του $\sqrt{2}\bar{I}$ στους άξονες a , b και c , και οι (σταθερά) τιμές των φασικών ρευμάτων στο σύστημα αναφοράς του δρομέα προκύπτουν ως οι προβολές του διανύσματος του ρεύματος στους άξονες " d " και " q ", δίνοντας

$$i_d = \sqrt{\frac{3}{2}} I_p \sin(\delta + \phi) \quad (2.21)$$

$$i_q = \sqrt{\frac{3}{2}} I_p \cos(\delta + \phi) \quad (2.22)$$

Όπου το I_p είναι το πλάτος του φασικού ρεύματος.

2.4 Ορισμός Παραμέτρων.

Το ισοδύναμο κύκλωμα της σύγχρονης γεννήτριας του σχήματος απαιτεί να είναι γνωστός μεγάλος αριθμός παραμέτρων. Οι παράμετροι αυτές μπορούν να ληφθούν είτε από τα δεδομένα σχεδιασμού της γεννήτριας είτε από μετρήσεις. Οι δύο πιο συχνόι τρόποι μετρήσεων των παραμέτρων μιας σύγχρονης μηχανής είναι τα χαρακτηριστικά βραχυκυκλώσεως και τα χαρακτηριστικά απόκρισης συχνότητας σε ακινησία. Και οι δύο αυτές μέθοδοι βασίζονται στον προσδιορισμό των παραμέτρων της μηχανής από τις ακόλουθες βασικές επαγωγές και χρονικές σταθερές:

- σύγχρονη επαγωγή L_d στον άξονα "d"

$$L_d = L_{ls} + L_{md} \quad (2.23)$$

- μεταβατική επαγωγή L'_d στον άξονα "d"

$$L'_d = L_{ls} + \frac{L_{md}L_{lfd}}{L_{md} + L_{lfd}} \quad (2.24)$$

- υπομεταβατική επαγωγή L''_d στον άξονα "d"

$$L''_d = L_{ls} + \frac{L_{md}L_{lfd}L_{lkd}}{L_{md}L_{lfd} + L_{md}L_{lkd} + L_{lfd}L_{lkd}} \quad (2.25)$$

- σύγχρονη επαγωγή L_q στον άξονα "q"

$$L_q = L_{ls} + L_{mq} \quad (2.26)$$

- υπομεταβατική επαγωγή L''_q στον άξονα "q"

$$L''_q = L_{ls} + \frac{L_{mq}L_{lfq}}{L_{mq} + L_{lfq}} \quad (2.27)$$

- μεταβατική χρονική σταθερά ανοιχτοκυκλώματος T'_{do} στον άξονα "d"

$$T'_{do} = \frac{L_{lfd} + L_{md}}{R_{fd}} \quad (2.28)$$

- μεταβατική χρονική σταθερά βραχυκυκλώματος T'_d στον άξονα "d"

$$T'_d = \frac{L_{lfd} + \frac{L_{md}L_{ls}}{L_{md} + L_{ls}}}{R_{fd}} \quad (2.29)$$

- υπομεταβατική χρονική σταθερά βραχυκυκλώματος T_{do}'' στον άξονα "d"

$$T_d'' = \frac{L_{lkd} + \frac{L_{md}L_{lfd}}{L_{md} + L_{lfd}}}{R_{kd}} \quad (2.30)$$

- υπομεταβατική χρονική σταθερά βραχυκυκλώματος T_d'' στον άξονα "d"

$$T_d'' = \frac{L_{lkd} + \frac{L_{md}L_{lfd}L_{ls}}{L_{md}L_{lfd} + L_{md}L_{ls} + L_{lfd}L_{ls}}}{R_{kd}} \quad (2.31)$$

- υπομεταβατική χρονική σταθερά βραχυκυκλώματος T_{qo}'' στον άξονα "q"

$$T_{qo}'' = \frac{L_{lkq} + L_{mq}}{R_{kq}} \quad (2.32)$$

- υπομεταβατική χρονική σταθερά βραχυκυκλώματος T_q'' στον άξονα "q"

$$T_q'' = \frac{L_{lkq} + \frac{L_{mq}L_{ls}}{L_{mq} + L_{ls}}}{R_{kq}} \quad (2.33)$$

Οι μετρήσεις από τη χαρακτηριστική βραχυκυκλώματος απαιτεί μέτρηση της κυματομορφής του ρεύματος οπλισμού αμέσως μετά από ένα τριφασικό βραχυκύκλωμα στα άκρα του οπλισμού. Κατά τη διάρκεια της μεταβατικής περιόδου, η μηχανή περιστρέφεται με σταθερή ταχύτητα και η τάση πεδίου διέγερσης διατηρείται σταθερή. Από τη μέτρηση της κυματομορφής του ρεύματος οπλισμού, είναι δυνατόν να εξαχθούν οι τιμές των L_d , L_d'' , T_d και T_d'' , και από αυτές να υπολογιστούν οι παράμετροι του άξονα "d". Η μέθοδος αυτή δεν δίνει την δυνατότητα υπολογισμού των παραμέτρων του άξονα "q".

Η μέτρηση των χαρακτηριστικών απόκρισης συχνότητας απαιτεί την ακινητοποίηση τον δρομέα σε μια θέση όπου ο "d" (ή "q") άξονας του συμπίπτει με την ροή που παράγεται από τα τυλίγματα οπλισμού. Με τα τυλίγματα πεδίου βραχυκυκλωμένα, η απόκριση συχνότητας μετρείται στα άκρα του οπλισμού. Τα διαγράμματα Bode που προκύπτουν επιτρέπουν τον υπολογισμό των L_d , L_d'' , L_q , L_q'' , T_d , T_{do} , T_d' , T_{do}' , T_q , T_{qo} και από αυτά υπολογίζονται όλες παράμετροι των αξόνων d και q. Σε ένα σύστημα που περιλαμβάνει και διεγέρτρια μονάδα, είναι δύσκολο να πραγματοποιηθούν οι παραπάνω μετρήσεις. Αυτό οφείλεται στο ότι τα τυλίγματα οπλισμού της διεγέρτριας και τα άκρα πεδίου του τυλίγματος διέγερσης της κύριας γεννήτριας δεν είναι προσβάσιμα (περιστρέφονται με τον άξονα).

2.4.1 Τυπικά Πειραματικά Δεδομένα Πραγματικής Γεννήτριας.

Σύμφωνα με τα προηγούμενα και για τις περιπτώσεις όπου χρειάζεται να προσομοιωθεί ένα μοντέλο πραγματικής γεννήτριας μέσης ονομαστικής ισχύος, προκύπτουν οι ακόλουθες τιμές για τις παραμέτρους της, με δοκιμές σε θερμοκρασία λειτουργίας 145°C:

- $P = 4$ πόλοι
- $R_s = 0.137\Omega$
- $L_{1s} = 0.897mH$
- $L_{md} = 43.2mH$
- $R_{fd} = 0.0266\Omega$
- $L_{1fd} = 3.37mH$
- $R_{kd} = 0.120\Omega$
- $L_{1kd} = 0.164mH$
- $L_{mq} = 20.8mH$
- $R_{kq} = 0.120\Omega$
- $L_{1kq} = 0.347mH$

2.4.2 Τυπικά Πειραματικά Δεδομένα Διεγέρτριας Μονάδας.

Τα αντίστοιχα δεδομένα σχεδιασμού για μία πραγματική διεγέρτρια είναι :

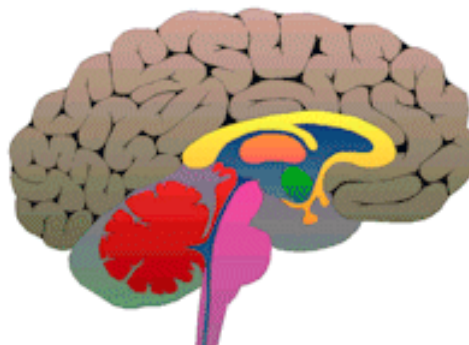
- $P = 8$ πόλοι
- $R_s = 0.218\Omega$
- $L_{1s} = 0.122mH$
- $L_{md} = 2.31mH$
- $R_{fd} = 0.123\Omega$
- $L_{1fd} = 0.845mH$
- $L_{mq} = 2.25mH$

3 ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ

3.1 Γενικά.

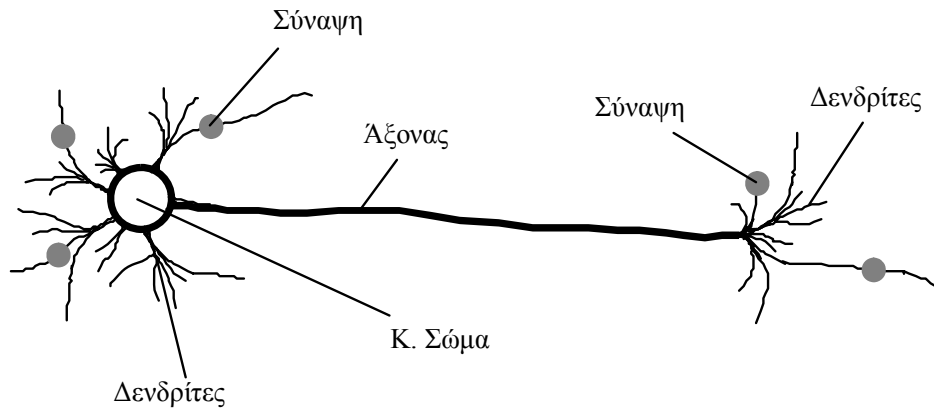
Ένα νευρωνικό δίκτυο είναι ένα σύνολο κόμβων (νευρώνων) που ο καθένας ενώνεται με όλους τους άλλους. Ουσιαστικά πρόκειται για ένα σύστημα επεξεργασίας πληροφορίας. Το σημείο-κλειδί του συστήματος είναι η δικτυακή δομή του συστήματος επεξεργασίας πληροφορίας. Αποτελείται από ένα μεγάλο αριθμό πολλαπλά διασυνδεδεμένων επεξεργαστών (νευρώνες), που δουλεύουν σε πλήρη συμφωνία μεταξύ τους. Τα νευρωνικά δίκτυα έχουν τη δυνατότητα μάθησης μέσα από παραδείγματα. Στα βιολογικά νευρωνικά δίκτυα αυτό επιτυγχάνεται με την τροποποίηση των συναπτικών συνδέσεων των νευρώνων. Η ίδια διαδικασία εφαρμόζεται και στα τεχνητά νευρωνικά δίκτυα (Τ.Ν.Δ.) με τη μεταβολή των τιμών παραμέτρων που σχετίζονται με τις συνάψεις και καλούνται βάρη. Έτσι τα νευρωνικά δίκτυα ανάλογα με το είδος τους ταξινομούνται σε βιολογικά νευρωνικά δίκτυα και τεχνητά νευρωνικά δίκτυα.

Τα βιολογικά νευρωνικά δίκτυα ουσιαστικά είναι το κέντρο ελέγχου των ανώτερων (πολυκύτταρων) οργανισμών. Το πιο γνωστό βιολογικό νευρωνικό δίκτυο είναι ο εγκέφαλος (Σχ. 3.1) του ανθρώπου, αλλά και όλων των ανώτερων οργανισμών. Ο εγκέφαλος θα μπορούσε να χαρακτηριστεί ως ένα υπερσύνολο διασυνδεδεμένων νευρωνικών δικτύων (υποσύνολα, κέντρα) εξειδικευμένο το καθένα σε μία λειτουργία (π.χ. όραση, συναισθήματα, μνήμη, κλπ). Οι νευρώνες είναι ειδικά διαμορφωμένα κύτταρα ώστε να επιτελούν συγκεκριμένη λειτουργία, κατ' αναλογία με τους Η/Υ θα λέγαμε ότι αποτελούν τα στοιχεία μνήμης και το σύνολο των εντολών, πρέπει βέβαια να αναφέρουμε πως ο μηχανισμός των επιμέρους (αλλά και του συνόλου) λειτουργιών του εγκεφάλου κατά ένα μεγάλο μέρος παραμένει άγνωστο. Το βασικό λειτουργικό χαρακτηριστικό ενός νευρώνα είναι η «ερεθιστικότητα», δηλ. η ικανότητα να αντιδρά σε εξωτερικά ερεθίσματα, όποτε αυτό συμβεί. Τότε εκφράζεται με ηλεκτροχημική διαταραχή της μεμβράνης και μέσω της σύναψης διαβιβάζεται στον επόμενο νευρώνα κ.ο.κ. Συνοπτικά είναι ο ελάχιστος χρόνος μεταξύ δύο διεγέρσεων καθώς και το μέγεθος της διέγερσης (τάση).



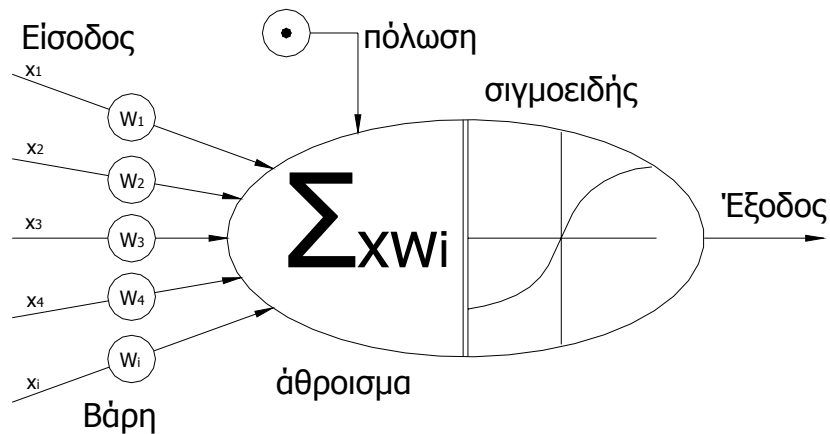
Σχ. 3.1 τυπική αναπαράσταση τομής ανθρώπινου εγκεφάλου

Ο νευρώνας (Σχ. 3.2) έχει τρία χαρακτηριστικά μέρη, τις απολήξεις για να λαμβάνει ερεθίσματα ή αλλιώς δενδρίτες, το κυτταρικό σώμα όπου παράγεται η ηλεκτροχημική αντίδραση και τον άξονα ο οποίος αποτελεί την έξοδο για την μεταφορά του ερεθίσματος σε άλλους νευρώνες (συναπτικές συνδέσεις). Στο σημείο διασύνδεσης δύο νευρώνων δημιουργείται μία σύναψη. Ένας νευρώνας κατά μέσο όρο έχει 10^3 δενδρίτες (εισόδους) και 500 διακλαδώσεις του άξονα (εξόδους, συνάψεις). Ο εγκέφαλος ενός ενήλικου ανθρώπου έχει περίπου 10^{11} νευρώνες (στο φλοιό, όπου και θεωρείται το ενεργό μέρος του εγκεφάλου) και μπορεί να κάνει περίπου 10^{16} συνάψεις το δευτερόλεπτο. Ο εγκέφαλος αποτελείται κατά το μεγαλύτερο μέρος από νευρογλοιακά κύτταρα, αλλά ο ρόλος τους περιορίζεται σε καθαρά εργατικό – προστατευτικό τομέα, δηλαδή βοηθούν και προστατεύουν τους νευρώνες, αλλά δε λαμβάνουν ενεργό ρόλο στην επεξεργασία των σημάτων, έτσι ο εγκέφαλος από πολλούς χαρακτηρίζεται σαν έναν ωκεανό από νευρογλοιακά κύτταρα στον οποίο είναι ποντισμένο ένα δίκτυο νευρώνων.[15]



Σχ. 3.2 Βιολογικός νευρώνας

Τα τεχνητά νευρωνικά δίκτυα (τεχνητή νοημοσύνη, μαθηματικά μοντέλα, κλπ) είναι το αποτέλεσμα της προσπάθειας κατανόησης και μίμησης της λειτουργίας των βιολογικών νευρωνικών δικτύων. Η προσπάθεια αυτή ξεκίνησε πρακτικά από τις εργασίες των McCulloch και Pitts το 1943 και τις ψυχολογικές θεωρίες του Hebb, γνωστή και ως “νόμος του Hebb⁶”. Αν και αρχικά επικράτησε ενθουσιασμός, το κλίμα και οι προσπάθειες ήταν υποτονικές μέχρι τα μέσα σχεδόν της δεκαετίας του ‘80, οπότε το θέμα αναθερμάνθηκε, λόγω του νεότερου γνωστικού πεδίου αλλά και των νέων «ισχυρών» Η/Υ, για να φθάσουμε στις μέρες μας όπου τα Τ.Ν.Δ. μαζί με παρεμφερείς θεωρίες – τεχνικές (ασαφής λογική, γενετικοί αλγόριθμοι, κλπ) αποτελούν έναν ταχύτατα εξελισσόμενο τομέα έρευνας και τεχνολογίας. Στο Σχ. 3.3 απεικονίζεται ένας τεχνητός νευρώνας, κατ’ αντιστοιχία με τον βιολογικό νευρώνα, και είναι περισσότερο από εμφανής οι «ομοιότητες» στη δομή τους.



Σχ. 3.3 Τυπικός Τεχνητός Νευρώνας

Παρόλα αυτά σε καμία περίπτωση δεν μπορούμε να πούμε πως το χάσμα μεταξύ βιολογικών και τεχνητών νευρωνικών δικτύων έχει γεφυρωθεί, καθώς μία ματιά στον Πίνακα 3.1 καταδεικνύει την υπάρχουσα κατάσταση. Παρατηρώντας τα στοιχεία του πίνακα διαπιστώνουμε πως παρά τη ραγδαία εξέλιξη της επιστήμης και τεχνολογίας γύρω από τους υπολογιστές, πάλι το χάσμα μεταξύ εγκεφάλου και Υ/Η παραμένει τεράστιο. Οι ηλεκτρονικοί υπολογιστές είναι ουσιαστικά το μοναδικό μέσο με το οποίο προσομοιώνουμε τη λειτουργία των Τ.Ν.Δ., για το λόγο αυτό γίνεται η συγκεκριμένη σύγκριση.

⁶ Νόμος του Hebb: Αν ένα επεξεργαστικό στοιχείο λαμβάνει στην είσοδο την έξοδο ενός άλλου επεξεργαστικού στοιχείου, και έχουν το ίδιο μαθηματικό πρόσημο, τότε το βάρος μεταξύ των επεξεργαστικών στοιχείων πρέπει να ενισχυθεί.

Πίνακας 3.1 Τυπική σύγκριση στοιχείων μεταξύ ανθρώπινου εγκεφάλου και Η/Υ

Στοιχείο	Ανθρώπινος Εγκέφαλος	Ηλεκτρονικός Υπολογιστής
Συχνότητα (ταχύτητα)	~100Hz	~5GHz
Υπολογιστική λειτουργία	Παράλληλη – Αποκεντρωμένη	Σειριακή – Επικεντρωμένη στον Κ.Ε. (CPU)
Ανοχή Σφάλματος	ΝΑΙ	ΟΧΙ
Ικανότητα Μάθησης	ΝΑΙ	ΝΑΙ ???

Βέβαια σήμερα υπάρχουν συστήματα συνδεδεμένων Η/Υ που μαζί με το σύστημα τροφοδοσίας και ψύξης καταλαμβάνουν ολόκληρα κτίρια, αλλά η δομή και η βάση τους παραμένει η ίδια, αν και έχουν γίνει προσπάθειες για παράλληλη λειτουργία επεξεργαστών, ο αριθμός τους όμως δεν έχει ξεπεράσει (ακόμη) τους 2^{11} (2048CPU), με αποτέλεσμα να απέχουν αρκετά από την ικανότητα του εγκεφάλου, παρόλο που η επεξεργαστική τους ισχύ είναι τεράστια σε σχέση με έναν κοινό προσωπικό Η/Υ. Επίσης γίνονται προσπάθειες για ανάπτυξη κβαντικών υπολογιστών (πχ επεξεργαστές φωτός, όπου το ρόλο τον ηλεκτρονίων τον έχουν τα φωτόνια), όπου αναμένεται να έχουμε νέα έκρηξη υπολογιστικής ισχύος, αλλά και προσπάθειες δημιουργίας υβριδικών επεξεργαστών από ζωντανούς νευρώνες και ολοκληρωμένα κυκλώματα από πυρίτιο (μέχρι στιγμής έχει δημιουργηθεί υβριδικό δίκτυο με περίπου 100 νευρώνες σαλιγκαριού⁷). Φυσικά όλα αυτά είναι ακόμη σε «εμβρυακό» στάδιο και το αποτέλεσμα κανείς δεν το γνωρίζει εκ των προτέρων. Μέχρι νεοτέρας λοιπόν ο (ανθρώπινος) εγκέφαλος παραμένει το «ταχύτερο» και πιο εξελιγμένο (γνωστό) νευρωνικό δίκτυο.

3.2 Χαρακτηριστικά Τ.Ν.Δ.

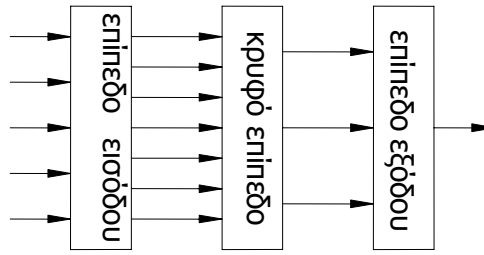
Τα στοιχεία που χαρακτηρίζουν ένα τεχνητό νευρωνικό δίκτυο είναι :

- Η αρχιτεκτονική του μοντέλου Τ.Ν.Δ.
- Ο αλγόριθμος εκπαίδευσης.
- Η ικανότητα γενίκευσης.
- Οι εφαρμογές του.

Η αρχιτεκτονική ενός τυπικού μοντέλου Τ.Ν.Δ. φαίνεται στο Σχ. 3.4, με τον όρο «μοντέλο» εννοούμε το μαθητικό μοντέλο με το οποίο προσεγγίζουμε την λειτουργία του πραγματικού νευρωνικού δικτύου. Ουσιαστικά, πρόκειται για τον αλγόριθμο πολλαπλασιασμού πινάκων και τον τρόπο εισαγωγής των δεδομένων τα οποία είναι διαφορετικά ανάλογα με το πρόβλημα που θέλουμε να επιλύσουμε. Έτσι ένα μοντέλο το οποίο είναι κατάλληλο για προσέγγιση τιμών μιας συνάρτησης είναι τελείως ακατάλληλο για την αναγνώριση φωτογραφιών κλπ. Για τη σωστή εκπαίδευση των διαφόρων μοντέλων, αλλά και ίδιων μοντέλων για διαφορετικές περιπτώσεις, έχουν αναπτυχθεί διαφορετικοί αλγόριθμοι εκπαίδευσης οι οποίοι καθορίζουν κατά ένα βαθμό και την συμπεριφορά του εκπαιδευμένου μοντέλου. Ένας καλός αλγόριθμος εκπαίδευσης πρέπει να συγκλίνει σχετικά γρήγορα, να δίνει μικρό σφάλμα, αλλά και το εκπαιδευμένο μοντέλο να έχει μεγάλη ικανότητα γενίκευσης. Όπως γίνεται αντιληπτό, η ικανότητα γενίκευσης είναι περισσότερο (άμεσα) εξαρτημένη από την εκπαίδευση του μοντέλου και λιγότερο από τον τύπο του νευρωνικού δικτύου.

Η ικανότητα γενίκευσης είναι η ικανότητα που έχει το Τ.Ν.Δ. να αντιμετωπίζει σωστά, «άγνωστα» δεδομένα του προβλήματος που επιλύει, έτσι και ο αλγόριθμος εκπαίδευσης (και κατ' επέκταση η εκπαίδευση ενός Τ.Ν.Δ.) καθίσταται ένα από τα σημαντικότερα σημεία υλοποίησής του. Οι εφαρμογές που μπορεί να έχει ένα νευρωνικό δίκτυο εξαρτώνται από το μοντέλο του νευρωνικού δικτύου και μάλιστα η διαδικασία επιλογής λειτουργεί αντίστροφα, δηλ. το μοντέλο του Τ.Ν.Δ. επιλέγεται (ή αναπτύσσεται) με γνώμονα την εφαρμογή που θέλουμε να έχει [15].

⁷ Οι νευρώνες του σαλιγκαριού είναι μεγαλύτεροι σε μέγεθος από των άλλων ζώων, για αυτό προτιμώνται.



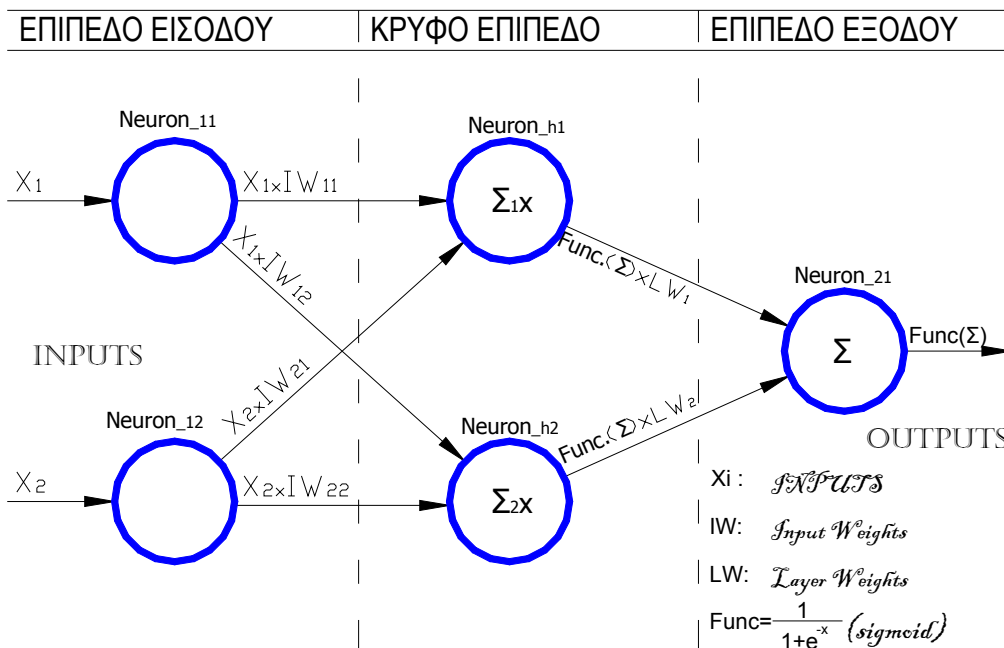
Σχ. 3.4 Τυπικό μοντέλο Τ.Ν.Δ . με ένα κρυφό επίπεδο

3.2.1 Μοντέλα Τεχνητών Νευρωνικών Δικτύων.

Η ανάπτυξη των Τ.Ν.Δ. ήταν ραγδαία από την δεκαετία του 80 και μετά, έτσι έχουν αναπτυχθεί πολλά μοντέλα Τ.Ν.Δ., ανάλογα με την εφαρμογή που προορίζεται το καθένα. Όλα τα γνωστά Τ.Ν.Δ. μπορούν να καταταγούν στις παρακάτω γενικές κατηγορίες, ανάλογα με το μαθηματικό μοντέλο προσομοίωσης που χρησιμοποιείται στο καθένα. Αυτές είναι α) Στατικά Νευρωνικά Δίκτυα, (π.χ. Feed-forward) όπως φαίνεται στο Σχ. 3.5 και β) Δυναμικά Νευρωνικά Δίκτυα (π.χ. Hopfield⁸), όπως φαίνεται στο Σχ. 3.6.

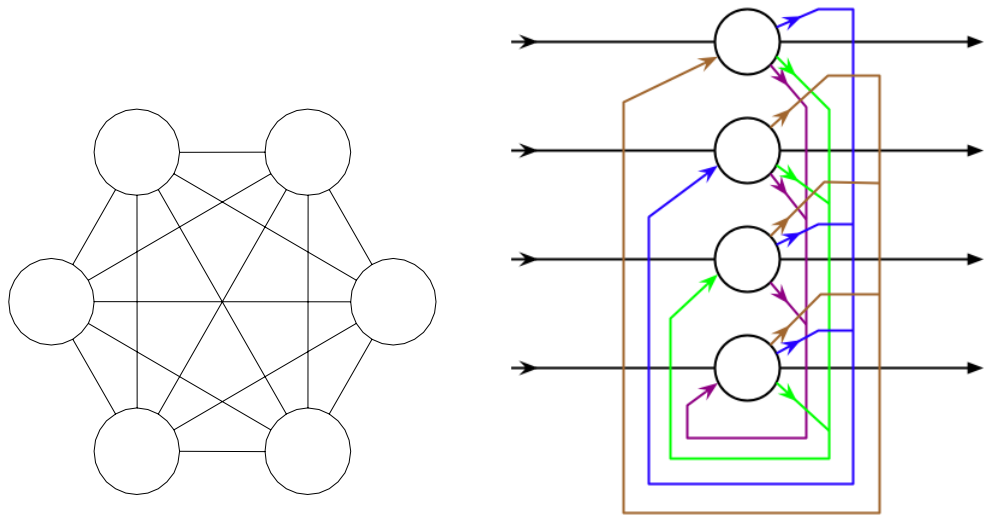
Τα στατικά Τ.Ν.Δ. υλοποιούν συναρτήσεις της μορφής $y = f(x)$, όπου το x είναι πραγματικός αριθμός και το y είναι η τιμή της συνάρτησης. Οι τιμές του y δεν εξαρτώνται μόνο από το x , αλλά και από το είδος της συνάρτησης, έτσι για ορισμένες συναρτήσεις το y περιορίζεται στο πεδίο $[-1, 1]$. Το πεδίο τιμών των x και y είναι πολυδιάστατο και με διαφορετική διάσταση το καθένα. Τα στατικά Τ.Ν.Δ. συνήθως χρησιμοποιούνται σε αναγνώριση χαρακτήρων, προσέγγιση μίας συνάρτησης κλπ.

Στα δυναμικά δίκτυα η είσοδος είναι διαταραγμένη (αλλοιωμένη) και η έξοδος αποτελεί την πλήρη (σωστή) «εικόνα» της εισόδου, έτσι ένα εκπαιδευμένο δίκτυο έχοντας μία ελλιπή ή διαταραγμένη είσοδο μας δίνει στην έξοδο την «ορθά» ταξινομημένη είσοδο. Αυτό έχει ως αποτέλεσμα να χρησιμοποιούνται για την κατάταξη διανυσμάτων, αναγνώριση φωτογραφιών ή επαναφορά κατεστραμμένων εικόνων κλπ.



Σχ. 3.5 Στατικό μοντέλο ("Feed Forward" [2x2x1])

⁸ Νόμος Hopfield: είναι παρόμοιος με τον κανόνα του Hebb με τη διαφορά ότι καθορίζει το πλάτος της ενίσχυσης ή της εξασθένησης. Δηλώνει ότι "αν η επιθυμητή έξοδος και η είσοδος είναι και οι δύο ενεργές ή και οι δύο ανενεργές, αύξησε το βάρος διασύνδεσης κατά (ποσό ίσο με) τον ρυθμό εκμάθησης, αλλιώς μείωσε το βάρος κατά (ποσό ίσο με) τον ρυθμό εκμάθησης.



Σχ. 3.6 Δομή Μοντέλου "HOPFIELD"

Μπορούμε να διαχωρίσουμε τα Τ.Ν.Δ. σε επιπλέον (υπο)κατηγορίες με βάση τον αλγόριθμο εκπαίδευσης, η οποία είναι και η πιο συνήθης διάκριση. Έτσι έχουμε μοντέλα με :

- Εκπαίδευση με επίβλεψη ("supervised", πχ με αλγόριθμο back propagation)
- Εκπαίδευση χωρίς επίβλεψη ("unsupervised", πχ μοντέλα "Kohonen"⁹)
- Εκπαίδευση με ανταγωνισμό. (πχ Τ.Ν.Δ. "Hamming")

3.2.2 Εκπαίδευση.

Ένα από τα κύρια χαρακτηριστικά ενός Τ.Ν.Δ. είναι ο αλγόριθμος εκπαίδευσης, ή αλλιώς εκπαίδευση. Με τον όρο εκπαίδευση εννοούμε τη διαδικασία απόκτησης εμπειρίας (μάθησης) από γνωστά παραδείγματα με διαδοχικές επαναλήψεις και τη διόρθωση του σφάλματος. Αξίζει να δώσουμε εδώ τον ορισμό της μάθησης των Mendel και McClaren:

«Μάθηση είναι μια διαδικασία με την οποία προσαρμόζονται οι ελεύθερες παράμετροι ενός νευρωνικού δικτύου μέσω μίας συνεχούς διαδικασίας διέγερσης από το περιβάλλον στο οποίο βρίσκεται το δίκτυο. Το είδος της μάθησης καθορίζεται από τον τρόπο με τον οποίο πραγματοποιούνται οι αλλαγές των παραμέτρων».

Η εμπειρία και η γνώση που αποκομίζεται από τις παραπάνω επαναλήψεις αποθηκεύονται στα βάρη, συντελεστές του νευρωνικού δικτύου. Ο αλγόριθμος (συνάρτηση) που κάνει τη διόρθωση (βελτίωση – μάθηση) είναι της μορφής¹⁰ :

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (3.1)$$

Όπου w το βάρος και k, j είναι οι δείκτες επιπέδου και νευρώνα αντίστοιχα (Σχ. 3.5), n ο δείκτης επανάληψης και Δ η μεταβολή.

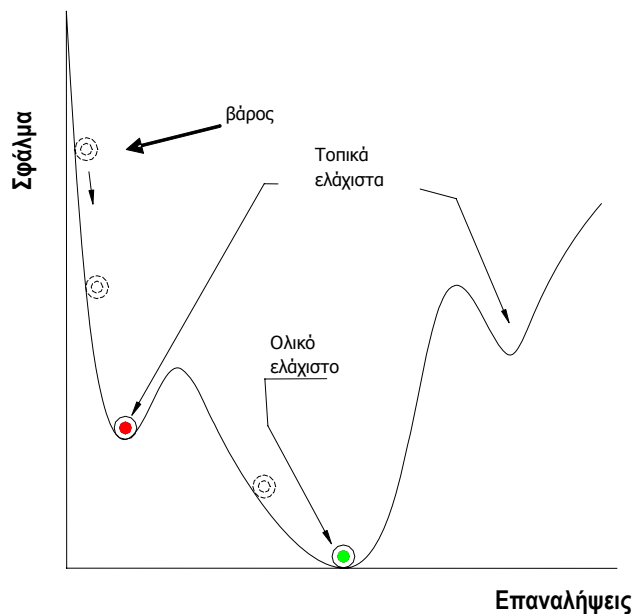
Η εκπαίδευση ενός νευρωνικού δικτύου βέβαια δεν είναι τόσο απλή, όσο ενδεχομένως φαίνεται, αλλά είναι μία αρκετά πολύπλοκη και σύνθετη διαδικασία που διαφέρει όχι μόνο ανάλογα με την αρχιτεκτονική του (μοντέλο) νευρωνικού δικτύου, αλλά και από την εφαρμογή για την οποία προορίζεται, καθώς πάντα υπάρχει ο «κίνδυνος» της υπερεκπαίδευσης (overtraining). Έτσι έχουν αναπτυχθεί διαφορετικοί αλγόριθμοι εκπαίδευσης, οι οποίοι εξάλλου χαρακτηρίζουν και το μοντέλο του Τ.Ν.Δ..

⁹ Αλλιώς γνωστά και ως αυτοοργανωμένοι χάρτες (self organized maps, SOM)
¹⁰ Για εκπαίδευση χωρίς τη χρήση του όρου της ορμής.

Γενικά οι τύποι εκπαίδευσης έχουν τα εξής χαρακτηριστικά :

- Κατά την εκπαίδευση με επίβλεψη, παρουσιάζονται στο Τ.Ν.Δ. ζεύγη διανυσμάτων εισόδου – εξόδου και το Τ.Ν.Δ. προσπαθεί να κατασκευάσει μία σχέση (συνάρτηση) μεταξύ των διανυσμάτων, όσο το δυνατόν πιο πιστή. Ο αλγόριθμος (εξωτερικής επίβλεψης) ακολουθεί μια επαναληπτική διαδικασία βήμα προς βήμα κατά την οποία τροποποιούνται τα βάρη, σε πολλές περιπτώσεις και οι πόλωση (bias) των νευρώνων, που συνδέουν τους νευρώνες. Η μεταβολή των βαρών επηρεάζεται από τους παρακάτω παράγοντες.

- Τις τρέχουσες τιμές των βαρών.
- Τις εισόδους (σταθμισμένο άθροισμα) των νευρώνων.
- Τις εξόδους (δραστηριότητα) των νευρώνων.
- Τις επιθυμητές εξόδους.
- Τον όρο της ορμής (όχι πάντα, εξαρτάται από τον αλγόριθμο).
- Λοιπούς πιθανούς όρους της Εξ. 3.1.



Σχ. 3.7 Συνάρτηση μεταβολής των βαρών

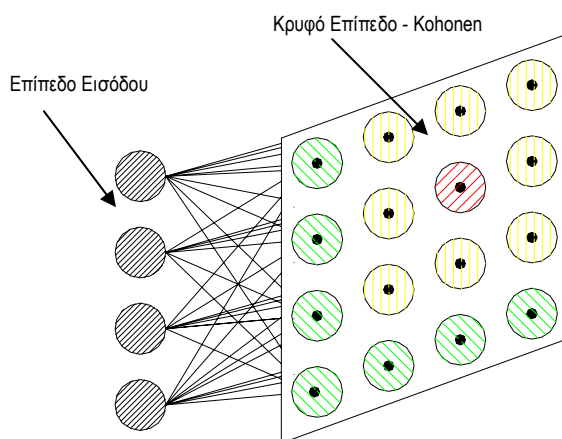
Η διαδικασία σταματάει όταν επιτευχθεί ένα από τα κριτήρια που θέτουμε, συνήθως αυτό είναι ελάχιστο μίας συνάρτησης σφάλματος όπως στο Σχ. 3.7. Το μειονέκτημα αυτής της μεθόδου είναι η πιθανότητα «παγίδευσης» σε τοπικό ελάχιστο όπως και ο χρόνος εκπαίδευσης, το πρόβλημα λύνετε με τη χρήση του παράγοντα της ορμής (Momentum term). Όταν ένας αλγόριθμος χρησιμοποιεί την ορμή, τότε στη μεταβολή των βαρών λαμβάνεται υπόψη και οι προηγούμενη τιμή των βαρών, αν και υπάρχουν μερικές παραλλαγές αλγόριθμων, η γενική ιδέα έχει ως εξής:

$$\Delta w_{kj}(n+1) = -(1 - mc)d \cdot \delta_i \cdot a_j - \Delta w_{kj}(n) \quad (3.2)$$

Όπου w το βάρος, k, j είναι οι δείκτες επιπέδου και νευρώνων αντίστοιχα, n ο δείκτης επανάληψης και Δ η μεταβολή, mc η σταθερά της ορμής (momentum constant), d η μεταβολή του όρου της ορμής, δ_i και a_i η προηγούμενη μεταβολή του βάρους και της πόλωσης (bias).

Έτσι με τη χρήση της ορμής συνήθως αποφεύγεται η παγίδευση σε τοπικό ελάχιστο, και μειώνεται ο χρόνος εκπαίδευσης. Επιπρόσθετα, βελτιώνεται η συνολική διαδικασία της μάθησης και συνήθως επιταχύνεται η σύγκλιση αφού μειώνεται η ταλάντωση γύρω από το ελάχιστο.

Στη εκπαίδευση χωρίς επίβλεψη, παρουσιάζονται στο Τ.Ν.Δ. μόνο τα διανύσματα εισόδου τα οποία ταξινομούνται σε ομάδες ανάλογα με τη σχετική ομοιότητα τους (αυτοοργάνωση του Τ.Ν.Δ.). Ο αλγόριθμος χωρίς εποπτεία πραγματοποιεί (ψευδο)τυχαίες μεταβολές των βαρών. Από τις μεταβολές αυτές διατηρούνται μόνο όσες βελτιώνουν (ελαχιστοποιούν) τη συνάρτηση σφάλματος. Με την κατάλληλη επιλογή του μεγέθους των μεταβολών στις τιμές των βαρών, ο αλγόριθμος οδηγείται τελικά στην εύρεση του ολικού ελάχιστου της συνάρτησης σφάλματος. Το σημαντικότερο μειονέκτημα αυτής της μεθόδου είναι ο μεγάλος χρόνος που απαιτείται για την εύρεση του ολικού ελάχιστου. Χαρακτηριστικό παράδειγμα νευρωνικού δικτύου με εκπαίδευση χωρίς επίβλεψη είναι το μοντέλο τύπου "Kohonen". Ένα τέτοιο μοντέλο ουσιαστικά αποτελείται από δύο επίπεδα το επίπεδο εισόδου και το κρυφό επίπεδο "Kohonen", όπου αν έχουμε N νευρώνες στο επίπεδο εισόδου, τότε στο επίπεδο "Kohonen" έχουμε $N \times N$ νευρώνες. Το επίπεδο Kohonen είναι δύο διαστάσεων, όπως φαίνεται και στο Σχ. 3.8.



Σχ. 3.8 Μοντέλο "Kohonen", ο κόκκινος νευρώνας «επιβιώνει».

Κατά την εκπαίδευση με ανταγωνισμό, οι νευρώνες ανταγωνίζονται για το ποιος θα «επιζήσει». Στην απλούστερη περίπτωση παραμένει ζωντανός μόνο ο νευρώνας που έχει τη μεγαλύτερη δραστηριοποίηση ενώ οι υπόλοιποι καταστέλλονται. Έτσι μετά την ολοκλήρωση της διαδικασίας παραμένουν ενεργοί μόνο οι «σωστοί» νευρώνες, οπότε και έχει ολοκληρωθεί η εκπαίδευση του Τ.Ν.Δ. Είναι δυνατόν ένα νευρωνικό δίκτυο να εκπαιδευτεί με περισσότερους από έναν τρόπο, πχ το κρυφό επίπεδο μπορεί να εκπαιδευτεί με αλγόριθμο με επίβλεψη και το επίπεδο εξόδου με αλγόριθμο χωρίς επίβλεψη. Τέλος να τονίσουμε, ότι σχεδόν σε όλες τις περιπτώσεις η εκπαίδευση γίνεται σε νεκρό χρόνο (off line) πριν τη χρήση του νευρωνικού δικτύου και ελάχιστες είναι οι εφαρμογές όπου έχουμε εκπαίδευση «on line» σε πραγματικό χρόνο.

3.2.3 Ικανότητα Γενίκευσης.

Ως ικανότητα γενίκευσης χαρακτηρίζεται η ικανότητα του τεχνητού νευρωνικού δικτύου να ανταποκρίνεται εξίσου καλά σε άγνωστα διανύσματα εισόδου σε σχέση με τα διανύσματα εισόδου που χρησιμοποιήθηκαν στην εκπαίδευση. Η ικανότητα γενίκευσης είναι από τα σημαντικότερα λειτουργικά χαρακτηριστικά του Τ.Ν.Δ., καθώς στα περισσότερα προβλήματα που αντιμετωπίζονται με μοντέλα Τ.Ν.Δ. δεν είναι δυνατόν να γνωστές εκ των προτέρων οι καταστάσεις στις οποίες ενδεχομένως να περιέλθει αυτό (άλλωστε ένας από τους σημαντικότερους λόγους για τους οποίους καταφεύγουμε στην επίλυση προβλημάτων με μοντέλο Τ.Ν.Δ. είναι αυτός). Οι παράγοντες οι οποίοι επηρεάζουν αρρριότι την ικανότητα γενίκευσης ενός νευρωνικού δικτύου είναι :

- Τα παραδείγματα εκπαίδευσης.
- Η πολυπλοκότητα του προβλήματος.
- Το μέγεθος του νευρωνικού δικτύου.

Αν τα παραδείγματα εκπαίδευσης είναι ελλιπή ή μη αντιπροσωπευτικά της διασποράς των τιμών του προβλήματος μειώνουν σημαντικά την ικανότητα γενίκευσης. Μία εμπειρικά σωστή αναλογία, των γνωστών διανυσμάτων εκπαίδευσης προς τα άγνωστα, είναι περίπου 75% - 25%. Η πολυπλοκότητα του προβλήματος παίζει σημαντικό ρόλο στην ικανότητα γενίκευσης του μοντέλου, γιατί ένα απλό πρόβλημα αντιμετωπίζεται πιο εύκολα από ένα πολύπλοκο, αλλά είναι και αμετάβλητος παράγοντας ο οποίος δεν επιδέχεται επεμβάσεις. Το μέγεθος του νευρωνικού δικτύου είναι μεγάλης σημασίας, γιατί αν επιλέξουμε ένα πολύ μικρό μοντέλο, τότε δεν θα μπορεί να σχηματίσει ένα επαρκές μοντέλο του προβλήματος. Από την άλλη, αν είναι πολύ μεγάλο πάλι θα έχει μειωμένη ικανότητα γενίκευσης και φυσικά μεγαλύτερο χρόνο εκπαίδευσης.

Έχει αποδειχθεί ότι κάποιο απεριόριστου μεγέθους Τ.Ν.Δ. μπορεί να δημιουργήσει οποιαδήποτε μη γραμμική απεικόνιση με απειροστή ακρίβεια, έτσι αν δεν έχουμε κάποια ιδέα για το μέγεθος του μοντέλου από το πρόβλημα καταφεύγουμε σε δοκιμές. Βέβαια ο αριθμός των νευρώνων των κρυφών επιπέδων πρέπει να είναι μικρότερος από τον αριθμό των διανυσμάτων εκπαίδευσης (όχι απλά μικρότερος αλλά σημαντικά μικρότερος) γιατί σε αντίθετη περίπτωση το Τ.Ν.Δ. «αποστηθίζει» τα γνωστά διανύσματα μόνο και δεν γενικεύει.

Η εκπαίδευση γενικότερα επηρεάζει την ικανότητα γενίκευσης του μοντέλου. Ουσιαστικά καθορίζει την ικανότητα γενίκευσης ευρύτερα και όχι μόνο στις περιπτώσεις όπου υπάρχει μεγάλο σφάλμα ή ελλιπής εκπαίδευση. Για παράδειγμα αυτό συμβαίνει όταν σταματήσει η εκπαίδευση χωρίς να επιτευχθεί ο επιθυμητός στόχος, ή τεθούν λάθος κριτήρια, πράγματα που επηρεάζουν αρνητικά την ικανότητα γενίκευσης ενός Τ.Ν.Δ. Πρόβλημα δημιουργείται επίσης και από την υπερβολική εκπαίδευση (υπερεκπαίδευση – overtraining) δηλ. αν η διαδικασία εκπαίδευσης δε σταματήσει στο κρίσιμο σημείο και συνεχιστεί, τότε παρόλο που το σφάλμα στην εκπαίδευση θα συνεχίζει να μειώνεται με τα γνωστά παραδείγματα, αντίστροφη πορεία ακολουθεί η ικανότητα γενίκευσης καθώς θα υπάρχει απόκλιση σε «άγνωστα» για το νευρωνικό δίκτυο παραδείγματα. Στην υπερεκπαίδευση το μοντέλο φαίνεται να μαθαίνει μόνο τα γνωστά παραδείγματα και αδυνατεί να ανταποκριθεί σωστά στα άγνωστα. Για να αποφευχθεί μία τέτοια δυσάρεστη κατάσταση, όπως θα δούμε και στα επόμενα κεφάλαια (Κεφ. 5), χρησιμοποιούμε αλγόριθμους εκπαίδευσης που δεν έχουν μοναδικό κριτήριο το σφάλμα, αλλά χρησιμοποιούν και άλλες παραμέτρους στην εκπαίδευση.

3.3 Γνωστά Μοντέλα Τ.Ν.Δ..

Όπως αναφέρθηκε και προηγουμένως, στην αρχή επικράτησε μεγάλη αισιοδοξία για τις δυνατότητες που παρείχαν τα νευρωνικά δίκτυα, μία αισιοδοξία όμως που σύντομα μετατράπηκε σε απογοήτευση και πολλοί ερευνητές στράφηκαν σε διαφορετικές ή παρεμφερείς θεωρίες. Παρόλα αυτά υπήρξαν και αυτοί που εργάστηκαν σκληρά πάνω στα νευρωνικά δίκτυα τα οποία στη συνέχεια οδήγησαν στη «σύγχρονη εποχή» των νευρωνικών δικτύων. Έτσι δημιουργήθηκαν νευρωνικά δίκτυα ορόσημα στην ιστορία εξέλιξης των Τ.Ν.Δ.. Τα πλέον χαρακτηριστικά μοντέλα είναι: το “Perceptron”, το “Adaline” και το “Brain State in a Box (BSB)”. Τα μοντέλα αυτά αποτέλεσαν την επιστημονική βάση για την μετέπειτα εξέλιξη των Τ.Ν.Δ..

3.3.1 Perceptron.

Το “Perceptron” προτάθηκε από τον Rosenblatt¹¹ το 1960 για να εξηγήσει την λειτουργία αναγνώρισης προτύπων από τον αμφιβληστροειδή. Είναι ένα απλό Τ.Ν.Δ. και η λειτουργία του βασίζεται στην ιδέα ότι ο αμφιβληστροειδής είναι μία

¹¹ Θεώρημα σύγκλισης Perceptron.

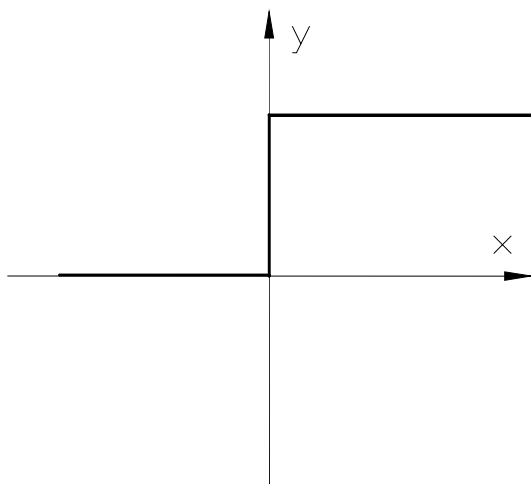
«Αν δοθεί ένα διάνυσμα εισόδου και ένα αντίστοιχο διάνυσμα επιθυμητών εξόδων, τότε υπάρχει κάποιος τρόπος εκπαίδευσης του Τ.Ν.Δ. για να επιτευχθεί η επιθυμητή έξοδος, αν (και μόνο αν) υπάρχουν οι κατάλληλες τιμές για τα βάρη».

δισδιάστατη διάταξη φωτοευαίσθητων αισθητήρων όπου αντιστοιχεί στο επίπεδο εισόδου. Τα αισθητήρια συνδέονται με το επίπεδο των νευρώνων, ανιχνευτές χαρακτηριστικών, μέσω αμετάβλητων βαρών. Το επίπεδο εξόδου αποτελείται από νευρώνες ανιχνευτές προτύπων (perceptrons) και συνδέεται με μεταβλητά βάρη ή με τυχαίο τρόπο με τους ανιχνευτές χαρακτηριστικών. Η συνάρτηση μεταφοράς του "Perceptron" είναι γραμμική συνάρτηση κατωφλίου (Σχ. 3.10). Αυτά τα χαρακτηριστικά προέρχονται από φυσιολογικές παρατηρήσεις, οι νευρώνες του αμφιβληστροειδούς ενεργοποιούνται όταν η δραστηριότητα εισόδου υπερβεί ένα όριο (κατώφλι), στη συνέχεια ο ρυθμός αυξάνει ανάλογα με τη δραστηριότητα της εισόδου. Διαφορετικά μπορούμε να χρησιμοποιήσουμε βηματική συνάρτηση μεταφοράς (Σχ. 3.9) η οποία κβαντίζει την έξοδο του νευρώνα στην τιμή [0] αν το σταθμισμένο άθροισμα είναι $\sum x_i w_i \leq 0$ και στην τιμή [1] αν $\sum x_i w_i > 0$. Η

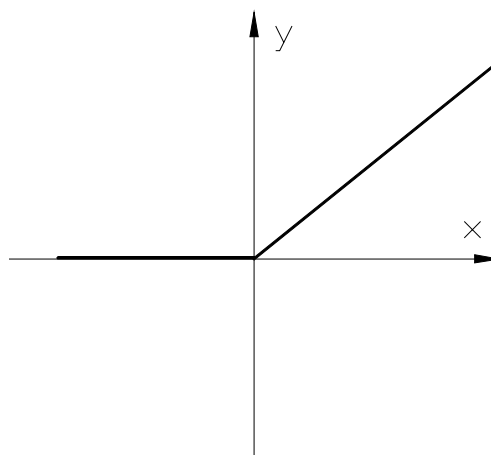
εκπαίδευση γίνεται με επίβλεψη και μεταβολή των βαρών του επιπέδου εξόδου.

Ο αλγόριθμος εκπαίδευσης μπορεί να συνοψιστεί ως εξής : Αν η έξοδος είναι 0 ενώ θα έπρεπε να είναι 1, αυξάνονται τα βάρη που αντιστοιχούν σε ενεργές εξόδους ανιχνευτών χαρακτηριστικών. Αν η έξοδος είναι 1 ενώ θα έπρεπε να είναι 0, μειώνονται τα βάρη που αντιστοιχούν σε ενεργές εξόδους ανιχνευτών χαρακτηριστικών, αν η έξοδος λάβει την επιθυμητή τιμή, τα βάρη δε μεταβάλλονται.

Τα βάρη μπορεί να μεταβάλλονται σταθερά ή με μεταβλητό ρυθμό, βελτιώνοντας έτσι το χρόνο σύγκλισης,, όμως ένας μεγάλος ρυθμός μεταβολής δεν δίνει μόνο γρήγορη σύγκλιση, αλλά και αυξημένη πιθανότητα να παρουσιαστεί αστάθεια. Το βασικό μειονέκτημα του Perceptron είναι η αδυναμία ταξινόμησης μη γραμμικά διαχωρίσιμων εισόδων, σήμερα έχουν αναπτυχθεί πολύ-επίπεδα Perceptron (MLP) τα οποία αντιμετωπίζουν αυτή την αδυναμία.



Σχ. 3.9 Συνάρτηση μεταφοράς ανιχνευτών προτύπων.



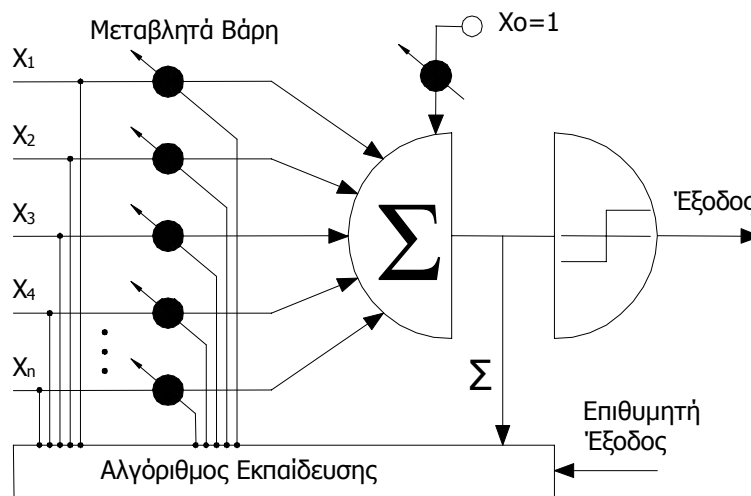
Σχ. 3.10 Γραμμική συνάρτηση κατωφλίου

3.3.2 Adaline.

Το Τ.Ν.Δ. Adaline (adaptive linear neuron, Σχ. 3.11) προτάθηκε από τον Widrow¹² και είναι μία από τις πρώτες πρακτικές εφαρμογές Τ.Ν.Δ. Έχει την ικανότητα να ταξινομεί μόνο ψηφιακά γραμμικά διαχωρίσιμα διανύσματα, όπως και το perceptron. Οι νευρώνες ενός Adaline συνδέονται με μεταβλητά βάρη, εκ των οποίων ένα είναι συνδεδεμένο σε σταθερή μοναδιαία είσοδο, αναλαμβάνοντας έτσι τον ρόλο του κατωφλίου (πόλωση). Τα διανύσματα εισόδου παίρνουν μόνο τις τιμές -1 και 1. Το σταθμισμένο άθροισμα τους επεξεργάζεται από έναν κβαντιστή, ο οποίος είναι το κέντρο λήψης αποφάσεων, η έξοδος του μπορεί να είναι -1 ή 1. Τα βάρη πρέπει να έχουν τυχαίες αρχικές τιμές, για να αποφεύγεται ο εγκλωβισμός σε τοπικά ελάχιστα. Ο αλγόριθμος εκπαίδευσης είναι ο αλγόριθμος «Widrow-Hoff» (κανόνας Δέλτα) και λειτουργεί με εξωτερική επίβλεψη.

Έχει αποδειχθεί πως μετά έναν αριθμό επαναλήψεων (περίπου πενταπλάσιος του αριθμού των βαρών) το Adaline συγκλίνει σε μία σταθερή λύση, αν η εκπαίδευση συνεχιστεί τότε αποκλίνει και πρέπει να μειωθεί ο ρυθμός εκπαίδευσης για να αποφευχθεί η απόκλιση. Όπως και το perceptron, το Adaline έχει ως βασικό μειονέκτημα το γραμμικό διαχωρισμό, και εδώ το πρόβλημα αντιμετωπίζεται κατά τον ίδιο, περίπου, τρόπο με τη δημιουργία του μοντέλων multiple-Adalines ή Madaline.

Το Madaline έχει ένα επίπεδο που αποτελείται από πολλά Adaline τα οποία συνδέονται με σταθερά μοναδιαία βάρη με το επίπεδο εξόδου, το οποίο αποτελείται από έναν μόνο νευρώνα. Ως συναρτήσεις ενεργοποίησης συνήθως χρησιμοποιούνται οι λογικές συναρτήσεις “AND” και “OR” ή κάποια συνάρτηση πλειοψηφίας, όπου η έξοδος παίρνει την τιμή της πλειοψηφίας των εισόδων.



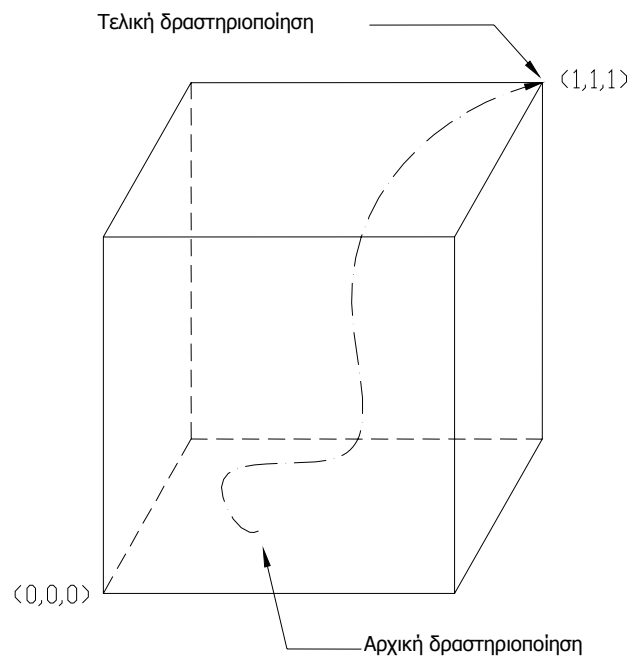
Σχ. 3.11 Σχηματικό διάγραμμα Τ.Ν.Δ. “Adaline”

¹² Αλγόριθμος Widrow-Hoff ή “LMS” (least mean squared) επίσης γνωστός και ως κανόνας Δέλτα (delta rule). $dW = h(t-W \cdot f) f^T$ όπου f είναι η έξοδος και t η είσοδος του Τ.Ν.Δ..

3.3.3 Brain State in a Box.

Το νευρωνικό δίκτυο Brain State in a Box ή “BSB”, είναι μη γραμμικό αυτοσυσχετιζόμενο μοντέλο και δημιουργός του είναι ο Anderson. Έχει ένα επίπεδο νευρώνων οι οποίοι επεξεργάζονται τα διανύσματα εισόδου, τα οποία είναι διπολικά και ανήκουν στο πεδίο $[-1, 1]$. Η έξοδος κάθε νευρώνα ανατροφοδοτείται συνεχώς από όλους τους νευρώνες του επιπέδου μέσω ασύμμετρων βαρών.

Το “BSB” μπορεί να διαχωριστεί σε δύο τμήματα, ένα γραμμικό και ένα μη γραμμικό. Το γραμμικό τμήμα αναλαμβάνει τη (γραμμική) συσχέτιση των διανυσμάτων εισόδου με τα διανύσματα εξόδου. Το μη γραμμικό τμήμα έχει διπλό ρόλο, απορρίπτει λάθος αποτελέσματα του γραμμικού και περιορίζει την έξοδο στο πεδίο $[-1, 1]$. Ο αλγόριθμος εκπαίδευσης χρησιμοποιεί τον κανόνα Δέλτα (Widrow-Hoff) ή τον κανόνα του Hebb, η δραστηριοποίησή του φαίνεται στο Σχ. 3.12. Επειδή είναι αυτοσυσχετιζόμενο δίκτυο έχει εφαρμογή σε βάσεις δεδομένων όπου θέλουμε σχεσιακή ανάκληση πληροφοριών (π.χ. στην ιατρική).



Σχ. 3.12 Δραστηριοποίηση Τ.Ν.Δ “BSB”

3.3.4 Radial Basis Function Network (RBF).

Το νευρωνικό δίκτυο “RBF” είναι μονοκατευθυντικό πλήρως διασυνδεδεμένο Τ.Ν.Δ. που αποτελείται από τρία επίπεδα, εισόδου – κρυφού – εξόδου. Κάθε νευρώνας του κρυφού επιπέδου εξειδικεύεται σε μια μικρή περιοχή του υπερχώρου των διανυσμάτων εισόδου, δηλ. δραστηριοποιείται αν το διάνυσμα εισόδου είναι σε αυτή την περιοχή.

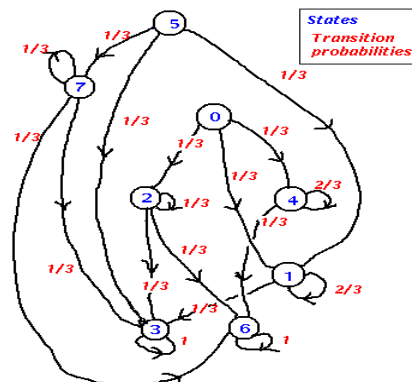
Οι νευρώνες του πρώτου επιπέδου δεν έχουν καμία υπολογιστική ισχύ, ενώ είσοδος στους νευρώνες του κρυφού επιπέδου είναι η ευκλείδεια απόσταση του διανύσματος εισόδου από το αντίστοιχο βάρος του νευρώνα, η οποία στη συνέχεια, μέσω της συνάρτησης μεταφοράς κανονικοποιείται στο πεδίο $[0, 1]$. Έτσι όσο πιο κοντά στο κέντρο της “Gaussian” συνάρτησης βρίσκεται το διάνυσμα εισόδου τόσο πιο μεγάλη θα είναι η έξοδος. Το “RBF” παρουσιάζει ακτινική συμμετρικότητα, λόγω της “Gaussian” συνάρτησης, έτσι ένας νευρώνας δίνει την ίδια έξοδο, όχι μόνο για ίδιες τιμές, αλλά για όλες τις τιμές που ισαπέχουν από το κέντρο του πυρήνα (βάρος).

Η εκπαίδευση ενός δικτύου “RBF”, ουσιαστικά πραγματοποιείται σε τρεις φάσεις, κατά την πρώτη φάση εκπαιδεύονται οι νευρώνες του κρυφού επιπέδου με αλγόριθμο χωρίς επίβλεψη, στην δεύτερη φάση εκπαιδεύονται οι νευρώνες του επιπέδου εξόδου, οι οποίοι επιτελούν κυρίως γραμμικούς μετασχηματισμούς, με αλγόριθμο με επίβλεψη και στην τρίτη φάση, όχι πάντα, χρησιμοποιείται αλγόριθμος με επίβλεψη για την βελτίωση των νευρώνων των δύο επιπέδων (κρυφού – εξόδου). Επίσης κατά τη διάρκεια εκπαίδευσης ενός δικτύου RBF ο αριθμός των νευρώνων του δεν παραμένει πάντα σταθερός, αλλά αυξάνονται οι νευρώνες μέχρι να επιτευχθεί το επιθυμητό αποτέλεσμα εξόδου.

Το μοντέλο “RBF” γενικά παρουσιάζει ικανοποιητική ικανότητα γενίκευσης, οι οποία επηρεάζεται από τον αριθμό εκπαιδευτικών διανυσμάτων. Βελτιώνεται όταν έχουμε μεγάλο αριθμό διανυσμάτων και ειδικά στα προβλήματα ταξινόμησης παρουσιάζει μεγάλη ικανότητα γενίκευσης.

3.3.5 Άλλα μοντέλα νευρωνικών δικτύων.

Από την αρχή της δεκαετίας του 80΄ μέχρι σήμερα έχουν αναπτυχθεί δεκάδες μοντέλα τεχνητών νευρωνικών δικτύων τα οποία μπορούν να δώσουν λύση σε πολλά προβλήματα. Μερικά από αυτά είναι μοντέλα τύπου : Hamming Network “Ha”, Feed Forward “FF”, Self organized maps “SOM”, Hopfield (Σχ. 3.13) κλπ. Ο λόγος που υπάρχουν τόσα πολλά μοντέλα δεν είναι τόσο η ευκολία που παρέχει η εξέλιξη της επιστήμης και της τεχνολογίας, όσο τα αναρίθμητα και δυσεπίλυτα προβλήματα τα οποία ενώ είναι εξαιρετικά δύσκολο να επιλυθούν με παραδοσιακές μεθόδους (ή με λάθος μοντέλο Ν.Δ.) λύνονται πολύ πιο εύκολα με τη χρήση του κατάλληλου μοντέλου νευρωνικού δικτύου. Κάθε μοντέλο έχει το δικό του, περιορισμένο, πεδίο εφαρμογών και δεν «μπορεί» να είναι καλό σε κάθε είδους πρόβλημα, πχ ένα μοντέλο “Feed Forward Back Propagation” είναι ιδανικό για την προσέγγιση ενός μαθηματικού μοντέλου μιας συνάρτησης, αλλά δεν μπορεί να χρησιμοποιηθεί σε πεδία όπου εφαρμόζονται μοντέλα τύπου “SOM” και αντιστρόφως.



Σχ. 3.13 Σχηματική αναπαράσταση δυναμικού μοντέλου στο χώρο

3.4 Εφαρμογές Τ.Ν.Δ.

Οι εφαρμογές και οι χρήσεις νευρωνικών δικτύων είναι ουσιαστικά απεριόριστες. Σήμερα χρησιμοποιούνται σχεδόν σε όλους τους τομείς τις τεχνολογίας και έρευνας και ενδεικτικά αναφέρονται :

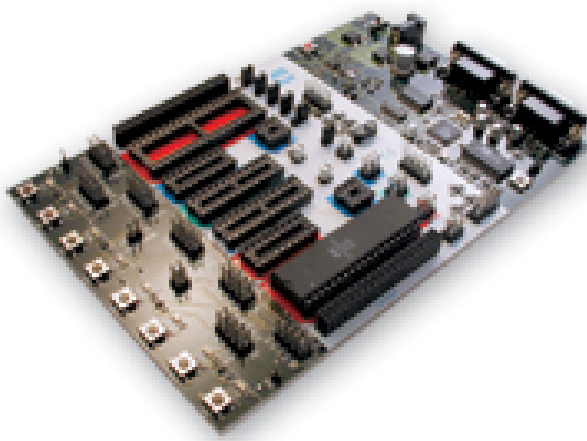
- Πρόβλεψη και προσομοίωση ατμοσφαιρικών φαινομένων, π.χ. καταιγίδες, τυφώνες κλπ.
- Πρόβλεψη άλλων γεγονότων, όπως η πορεία μιας μετοχής ή άλλου χρηματιστηριακού δείκτη, σεισμών κλπ.
- Αυτόνομο έλεγχο οχημάτων, ιπτάμενων, επίγειων, υποβρυχίων κλπ.
- Έλεγχο συστημάτων, π.χ. κατανάλωση καυσίμου σε ένα αεροσκάφος, *έλεγχος μιας ηλεκτρικής μηχανής* κλπ.
- Παρακολούθηση πολλαπλών στόχων με ένα RADAR.
- Αναγνώριση προτύπων, π.χ. χαρακτήρες κειμένου, εικόνας, φωνής, τεχνητή όραση κλπ.
- Ιατρικά μοντέλα για την κατανόηση της λειτουργίας του εγκεφάλου.
- Επίλυση (προσέγγιση) δύσκολων μαθηματικών προβλημάτων (συναρτήσεις).
- Επεξεργασία σήματος.
- Συμπύεση – κρυπτογράφηση δεδομένων.
- Προσαρμοστικά φίλτρα.

4 ΜΙΚΡΟΕΛΕΓΚΤΗΣ

4.1 Η Βάση του Συστήματος – Αναπτυξιακή Πλακέτα.

Στα πλαίσια αυτής της πτυχιακής εργασίας το μοντέλο του νευρωνικού δικτύου θα μεταφερθεί σε κώδικα C όπου στη συνέχεια εύκολα μετατρέπεται σε συμβατή μορφή για τον μικροελεγκτή. Ο μικροελεγκτής θα προγραμματιστεί με το μοντέλο του T.N.Δ., ώστε να χρησιμοποιηθεί για τον έλεγχο ηλεκτρικής μηχανής. Η βάση του συστήματος είναι η αναπτυξιακή πλακέτα “starter kit” ATMEL - STK500 (Σχ. 4.1), έτσι έχουμε τη δυνατότητα να προγραμματίσουμε τον μικροελεγκτή με έναν κοινό Η/Υ μέσω της σειριακής θύρας (RS232), ο οποίος είναι εξοπλισμένος με το απαραίτητο λογισμικό βέβαια. Οι απαιτήσεις του μοντέλου T.N.Δ. δεν εξαντλούν τις δυνατότητες της αναπτυξιακής πλακέτας, καθώς εκτός από τις τέσσερις θύρες (πόρτες) επικοινωνίας με τον μικροελεγκτή μας παρέχει και άλλες δυνατότητες επέκτασης, οι οποίες ενδεχομένως να είναι χρήσιμες σε μελλοντικές εφαρμογές (αλλαγές ή προσθήκες).

Στον έλεγχο μιας ηλεκτρικής μηχανής, ο χρόνος είναι από τους σημαντικότερους παράγοντες. Έτσι για να εξασφαλίσουμε την ικανότητα του μικροελεγκτή να ανταποκριθεί όσο πιο άμεσα γίνεται χρησιμοποιούμε εξωτερικό ταλαντωτή (κρύσταλλο – “oscillator”) συχνότητας (έως)16MHz, όπου είναι η μέγιστη δυνατή συχνότητα που μπορεί να λειτουργήσει ομαλά και χωρίς ενεργό σύστημα ψύξης ο μικροελεγκτής.



Σχ. 4.1 Αναπτυξιακή πλακέτα ATMEL STK – 500 (starter kit)

4.2 Μικροελεγκτής ή Μικροεπεξεργαστής ?

Συχνά συγχέεται ο μικροελεγκτής (Σχ. 4.3) με τον μικροεπεξεργαστή (Σχ. 4.2), παρόλο που έχουν παρόμοια πορεία εξέλιξης, ο μικροελεγκτής διαφέρει από τον μικροεπεξεργαστή. Ο μικροεπεξεργαστής (microprocessor, cpu) αποτελεί τον πυρήνα ενός υπολογιστικού συστήματος (Η/Υ, PLC¹³ κλπ),ακόμη και ένας μικροελεγκτής περιέχει έναν μικροεπεξεργαστή. Όσο δε αφορά την υπολογιστική ισχύ και την πολυπλοκότητα τις δομές, ένας μικροεπεξεργαστής μπορεί να έχει πάνω από 1000 φορές μεγαλύτερη υπολογιστική ισχύ σε σχέση με έναν μικροελεγκτή και πολύ μεγαλύτερη πυκνότητα ολοκλήρωσης.

Αντίθετα ένας μικροελεγκτής επικεντρώνεται σε άλλα χαρακτηριστικά και ουσιαστικά θα μπορούσε να χαρακτηριστεί ως μικροϋπολογιστής (όπως είναι άλλωστε), αφού κύριο προσόν του δεν είναι η επεξεργαστική ισχύς του αλλά η αυτόνομη λειτουργία του. Για να λειτουργήσει, πρακτικά το μόνο που χρειάζεται είναι μία εξωτερική πηγή τροφοδοσίας, καθώς οι σύγχρονοι μικροελεγκτές διαθέτουν ενσωματωμένη επεξεργαστική μονάδα, μνήμη RAM¹⁴ – ROM¹⁵, αλλά και μνήμη

¹³ PLC : προγραμματιζόμενος λογικός ελεγκτής

¹⁴ RAM : (Random Access Memory) Μνήμη τυχαίας προσπέλασης

FLASH¹⁶ για την αποθήκευση του προγράμματος, εσωτερικούς timers – counters, αυτόματη έναρξη – διακοπή λειτουργίας καθώς και (υποτυπώδη, υλοποίηση με λογισμικό) εσωτερικό ταλαντωτή (βέβαια για εφαρμογές ακριβείας ή μεγαλύτερων συχνοτήτων ο εξωτερικός ταλαντωτής κρύσταλλος είναι απαραίτητος).

Βασικό προσόν τους είναι η εξαιρετικά χαμηλή κατανάλωση ενέργειας, πρακτικά η ισχύς τους κυμαίνεται κάτω από ένα watt ($P < 1W$) τη στιγμή που σύγχρονοι μικροεπεξεργαστές ξεπερνούν τα 100W σε κατανάλωση ηλεκτρικής ενέργειας, χωρίς να λαμβάνεται υπόψη η ισχύς του συστήματος ψύξης. Επίσης η απλότητα στον προγραμματισμό είναι ένα σημαντικό πλεονέκτημα, ένας μικροελεγκτής μπορεί να «τρέξει» πρόγραμμα μερικών γραμμών κώδικα ενώ αντίθετα ένας Η/Υ χρειάζεται λειτουργικό Σύστημα εκατομμυρίων γραμμών κώδικα.

Γίνεται φανερό πως τα πλεονεκτήματα ενός μικροελεγκτή έχουν να κάνουν με την αυτόνομη λειτουργία, την ευκολία προγραμματισμού αλλά και την αξιοπιστία (συνήθως δε χρειάζονται ενεργή ψύξη και έχουν ένα όριο διακύμανσης στην τροφοδοσία) καθώς και την επεκτασιμότητα ώστε να καλύπτουν ευρύ φάσμα εφαρμογών.



Σχ. 4.2 Μικροεπεξεργαστής – CPU (AMD Athlon 64bit)

4.3 Μικροελεγκτές.

Οι μικροελεγκτές, όπως ήδη αναφέρθηκε, έχουν ιστορία εξέλιξης ανάλογη με αυτή με τον ηλεκτρονικών υπολογιστών, όλα ξεκίνησαν από το πρώτο ολοκληρωμένο κύκλωμα (CMOS¹⁷) της Intel για να ακολουθήσει ραγδαία εξέλιξη σε οτιδήποτε έχει σχέση με την ηλεκτρονική την πληροφορική και όλες τις σχετικές επιστήμες και τεχνολογίες. Η εξέλιξη των μικροελεγκτών ακολούθησε αυτό το ρεύμα, αλλά η έμφαση δεν δόθηκε στην απόλυτη επεξεργαστική ισχύ, (αυτό το κάνουν οι μικροεπεξεργαστές) αλλά συμπυκνώνουν όσο το δυνατόν περισσότερα και διαφορετικά μεταξύ τους «εξαρτήματα» (περιφερειακά) σε ένα ολοκληρωμένο κύκλωμα. Αυτό έχει ως αποτέλεσμα το ευρύ πεδίο εφαρμογών των μικροελεγκτών, αφού παρέχουν πληθώρα δυνατοτήτων, όπως έλεγχος PWM¹⁸, εξαιρετικά χαμηλή κατανάλωση ενέργειας (ειδικά σε κατάσταση αναμονής) αλλά και ικανότητα διαχείρισης εξειδικευμένων πρωτοκόλλων π.χ. TCP/IP¹⁹. Σήμερα στην αγορά υπάρχουν εκατοντάδες διαφορετικοί μικροελεγκτές, όπου διαφέρουν μεταξύ τους στη δομή, δηλ. στον αριθμό ακροδεκτών, τη συσκευασία κλπ, όσο και στην αρχιτεκτονική λειτουργίας, πχ RISC²⁰, CISC²¹ κλπ. Κατά κύριο λόγο όμως κατατάσσονται σε οικογένειες, ανάλογα τον κατασκευαστή πολλές φορές, όπως AVR, PIC, ARM, Zilog κλπ όπου προορίζονται για γενική αλλά και εξειδικευμένη χρήση. Ειδική κατηγορία αποτελούν οι μικροελεγκτές που προορίζονται για στρατιωτικές εφαρμογές, καθώς το κόστος τους είναι αρκετά υψηλό εξαιτίας των ειδικών προδιαγραφών που έχουν και όχι λόγω των επιδόσεων τους (π.χ. αντέχουν αρκετά χαμηλές ή υψηλές θερμοκρασίες, έχουν μεγαλύτερη αντοχή σε ηλεκτρομαγνητική ακτινοβολία κλπ).

¹⁵ ROM: (Read Only Memory) Μνήμη μόνο για ανάγνωση

¹⁶ Flash Memory: Ειδική μνήμη, εφεύρεση της Intel, η οποία δεν χάνει τα δεδομένα όταν διακοπή η παροχή τάσης.

¹⁷ CMOS : complementary metal-oxide-semiconductor

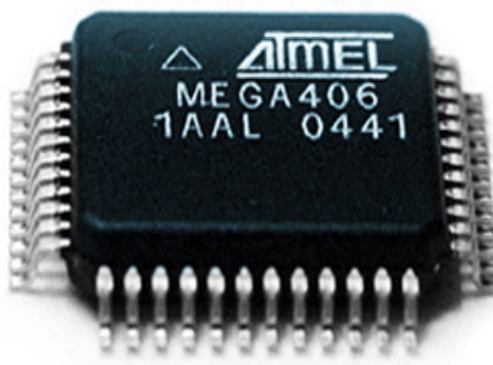
¹⁸ PWM: (Pulse Width modulation) Μεταβλητό εύρος παλμού, τεχνική μεταβολής της τάσης (ισχύος) με χρήση ημιαγώγιμων στοιχείων.

¹⁹ TCP/IP: Πρωτόκολλο επικοινωνίας δικτύων δεδομένων (πολλαπλής διασύνδεσης)

²⁰ RISC : Reduced instruction set Computing

²¹ CISC : Complex Instruction Set Computer

Για τις ανάγκες της παρούσας εργασίας επιλέγεται μικροελεγκτή της οικογένειας AVR καθώς καλύπτει τις απαιτήσεις μας και συγκαταλέγονται στους μικροελεγκτές με την μεγαλύτερη επεξεργαστική ισχύ, σε σχέση με το κόστος του βέβαια, και διαθέτει ικανοποιητικό μέγεθος(16kb) μνήμης Flash και αρκετούς καταχωρητές (32x8 bit registers) όπως και πληθώρα άλλων χαρακτηριστικών (timers, counters, interrupt κλπ).



Σχ. 4.3 Μικροελεγκτής ATMEGA

4.3.1 Μικροελεγκτής AVR ATmega16.

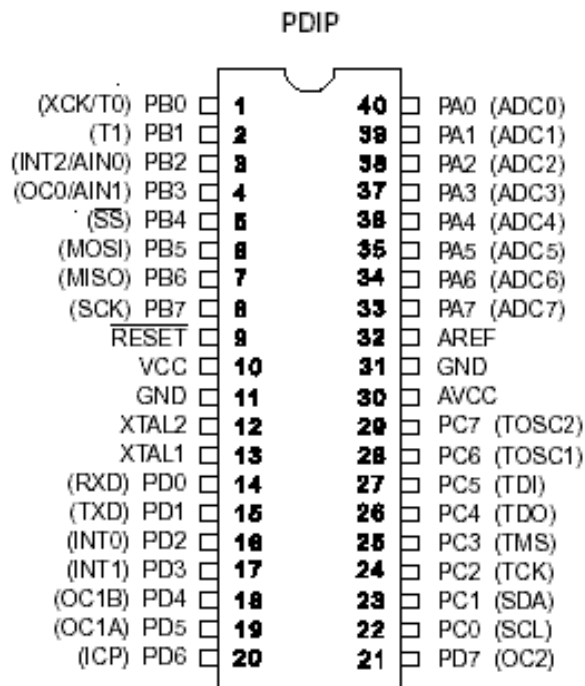
Ο μικροελεγκτής ATmega16²² (Σχ. 4.4) έχει παρόμοια αρχιτεκτονική και επεξεργαστική ισχύ με την ναυαρχίδα της οικογένειας AVR mega (ATmega128), αλλά διαθέτει λιγότερη μνήμη, 16 αντί 128Kbytes. Τα επιμέρους χαρακτηριστικά του είναι :

- Αρχιτεκτονική RISC 8bit.
- 131 εντολές, οι περισσότερες 1 κύκλου.
- 32 καταχωρητές γενικής χρήσης (registers) των 8 bit.
- Επεξεργαστική ισχύς 16 MIPS²³ στα 16 MHz.
- 16Kbytes μνήμη Flash γενικής χρήσης με διάρκεια ζωής 10000 επανεγγραφές.
- 512byte EEPROM²⁴
- 1kbyte μνήμη RAM
- Πλήρως στατική λειτουργία.
- 2 timers/counters των 8bit με ξεχωριστό προγραμματισμό ο καθένας
- 1 timer/counter 16bit.
- 4 κανάλια ελέγχου PWM.
- αναλογικό – ψηφιακό μετατροπέα (ADC) 8 καναλιών 10bit.
- Εξωτερικό και εσωτερικό interrupt καθώς και watchdog timer.
- Ανοχή στην τάση τροφοδοσίας από 4.5 έως 5.5 volt
- Συχνότητα λειτουργίας έως 16MHz
- 6 διαφορετικές καταστάσεις αναμονής για εξοικονόμησης ενέργειας.

²² Τα πλήρη χαρακτηριστικά του μικροελεγκτή παρατίθενται στο παράρτημα.

²³ MIPS: (Millions instructions per second) Εκατομμύρια υπολογισμοί το δευτερόλεπτο , 1MIPS ανά 1MHz

²⁴ Electrically Erasable Programmable Read-Only Memory



Σχ. 4.4 Σχηματικό διάγραμμα του μικροελεγκτή ATmega16

4.3.2 Προγραμματισμός Μικροελεγκτή AVR.

Ο μικροελεγκτής από μόνος του αποτελεί το υλικό μέρος (hardware), όπως και οι Η/Υ, δεν θα είχε καμία ιδιαίτερη αξία αν δεν ήταν εφικτός ο προγραμματισμός του, ώστε να ανταποκρίνεται στην εκάστοτε περίπτωση. Έτσι ο προγραμματισμός του μικροελεγκτή, δηλ. το λογισμικό (software), όχι μόνο έχει ιδιαίτερη σημασία, αλλά σχεδόν πάντα και μεγαλύτερη αξία από το υλικό μέρος αφού αυτό δίνει «ζωή» στον μικροελεγκτή. Όπως είναι λογικό, κάθε μικροελεγκτής μπορεί να προγραμματιστεί με γλώσσα χαμηλού επιπέδου assembly.

Ο προγραμματισμός με assembly προσφέρει σημαντικά μειωμένο μέγεθος κώδικα και αυξημένη ταχύτητα, όμως έχει όλες τις δυσκολίες που αντιμετωπίζει ένας προγραμματιστής με μία γλώσσα χαμηλού επιπέδου και ειδικά στην συντήρηση μεγάλου κώδικα.

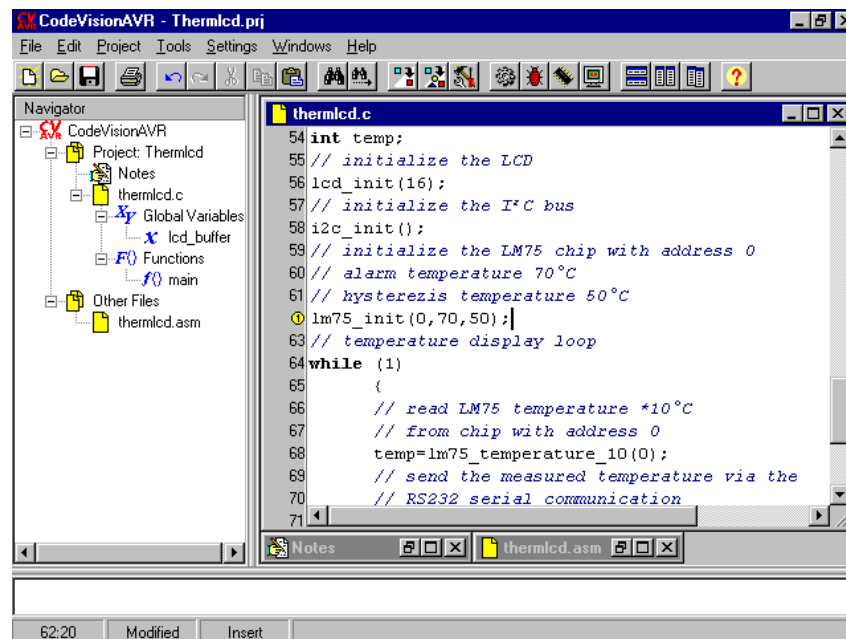
Για να αντιμετωπιστούν αυτά τα προβλήματα και να γίνει ευκολότερος ο προγραμματισμός του, ο μικροελεγκτής μπορεί να προγραμματιστεί και με γλώσσες υψηλότερου επιπέδου όπως C, Basic, Pascal κλπ (τελευταία αρχίζουν να εμφανίζονται compilers ακόμη και για C++), έχοντας βέβαια ως πλεονέκτημα την ευκολότερη ανάπτυξη και συντήρηση του κώδικα. Πολλές φορές μπορεί να συναντήσουμε «υβριδικό» κώδικα όπου μπορεί να είναι γραμμένος σε C αλλά να εμπεριέχει και τμήματα από assembly.

Για να μεταφερθεί ένα πρόγραμμα στον μικροελεγκτή, απαραίτητη προϋπόθεση είναι η μεσολάβηση του μεταγλωττιστή (compiler) ώστε να μετατραπεί σε γλώσσα μηχανής αλλά και ένας «προγραμματιστής» που ουσιαστικά αναλαμβάνει να «φορτώσει» το πρόγραμμα στη μνήμη του μικροελεγκτή. Συνήθως η γλώσσα προγραμματισμού εξαρτάται από την οικογένεια του μικροελεγκτή, πχ οι AVR έχει επικρατήσει να προγραμματίζονται κυρίως με C και κατά δεύτερο λόγο με assembly και ελάχιστα με κάποια άλλη γλώσσα (σε μία τέτοια ερώτηση που έγινε σε σχετικό δικτυακό τόπο²⁵ τα ποσοστά ήταν 62% C – 35% ASM μόλις 3% άλλη), πάντα βέβαια η γλώσσα καθορίζεται από τον “compiler” που θα επιλέξουμε να εργαστούμε. Ο κώδικας για την παρούσα εργασία θα γραφτεί σε γλώσσα C στο περιβάλλον CodeVision AVR, ως “debugger” (εύρεση λαθών – εξομοίωση) χρησιμοποιείται το συνοδευτικό λογισμικό από την ATMEL, AVR Studio 4.

²⁵ <http://www.avrfreaks.net>

4.3.3 CodeVision AVR

Ο μεταγλωττιστής που χρησιμοποιούμε είναι το CodeVision AVR (Σχ. 4.5) και είναι από την εταιρεία HP Info Tech. Δεν είναι ακριβώς ότι ένας κοινός μεταγλωττιστής, αλλά θα μπορούσε να χαρακτηριστεί ως συμβολομεταφραστής (cross-compiler). Υποστηρίζει όλους τους μικροελεγκτές της οικογένειας AVR και δυνατότητα ανάπτυξης κώδικα σε C και βέβαια σε assembly. Επίσης έχει ενσωματωμένο τον «προγραμματιστή» του μικροελεγκτή και δε χρειάζεται άλλο πρόγραμμα για τη μεταφορά του προγράμματος στον μικροελεγκτή. Είναι πλήρως συμβατό με την ANSI έκδοση της C και διαθέτει και τις ανάλογες βιβλιοθήκες όπως επίσης και εργαλεία για την ευκολότερη και ταχύτερη ανάπτυξη των τυποποιημένων τμημάτων του κώδικα.



Σχ. 4.5 Περιβάλλον CodeVision AVR

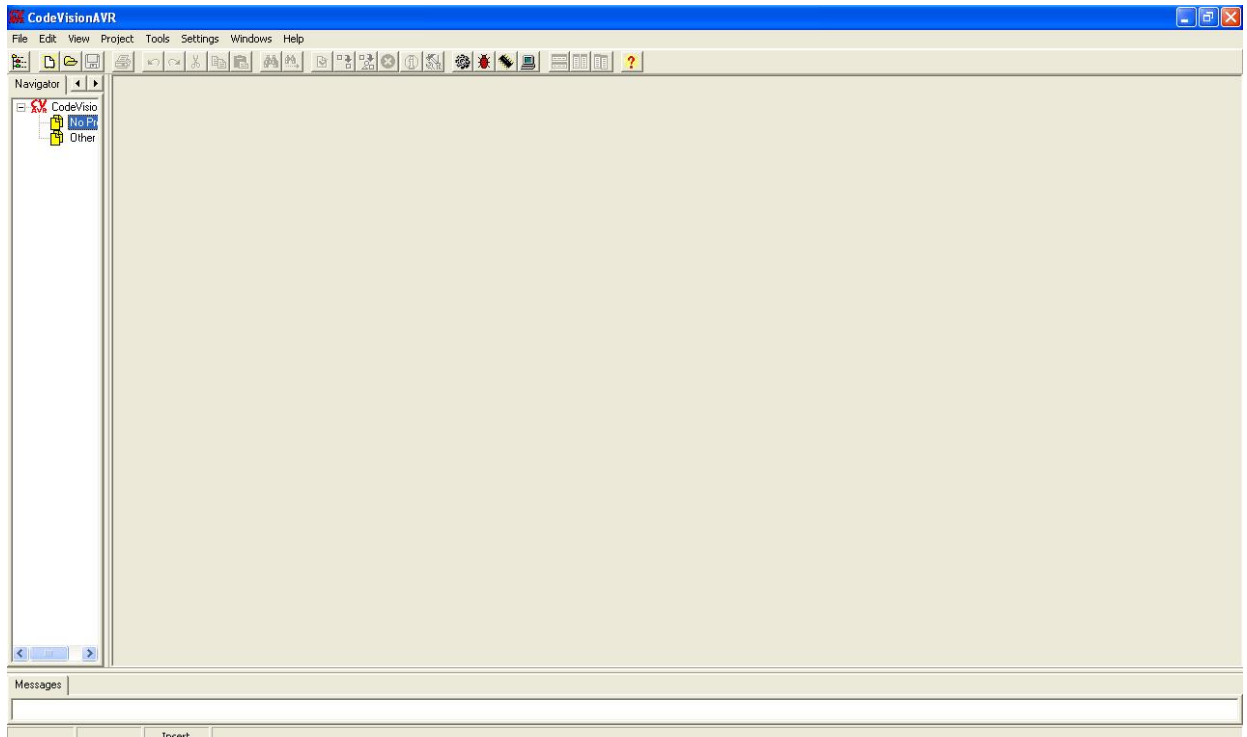
4.3.4 AVR Studio.

Το Atmel AVR Studio, έκδοση 4.09, είναι το βασικό «εργαλείο» ανάπτυξης εφαρμογών για μικροελεγκτές AVR, για το λόγο αυτό άλλωστε διατίθεται δωρεάν από την Atmel η οποία είναι ο κύριος κατασκευαστής AVR μικροελεγκτών. Όπως είναι αναμενόμενο υποστηρίζει όλους μικροελεγκτές της Atmel, αλλά (δυστυχώς) δεν παρέχει δυνατότητα ανάπτυξης κώδικα σε C (αυτό μπορεί να γίνει, αλλά χρειάζεται ειδικός compiler από τρίτο). Παρέχει όμως πολλά εργαλεία διόρθωσης του κώδικα και εντοπισμού σφαλμάτων, όπως assembler – disassembler, εξομοιωτή για τους μικροελεγκτές, γραφικό περιβάλλον εργασίας αλλά και εικονική – γραφική αναπαράσταση του μικροελεγκτή καθώς και όλα τα γνωστά απαραίτητα εργαλεία για “debugging”.

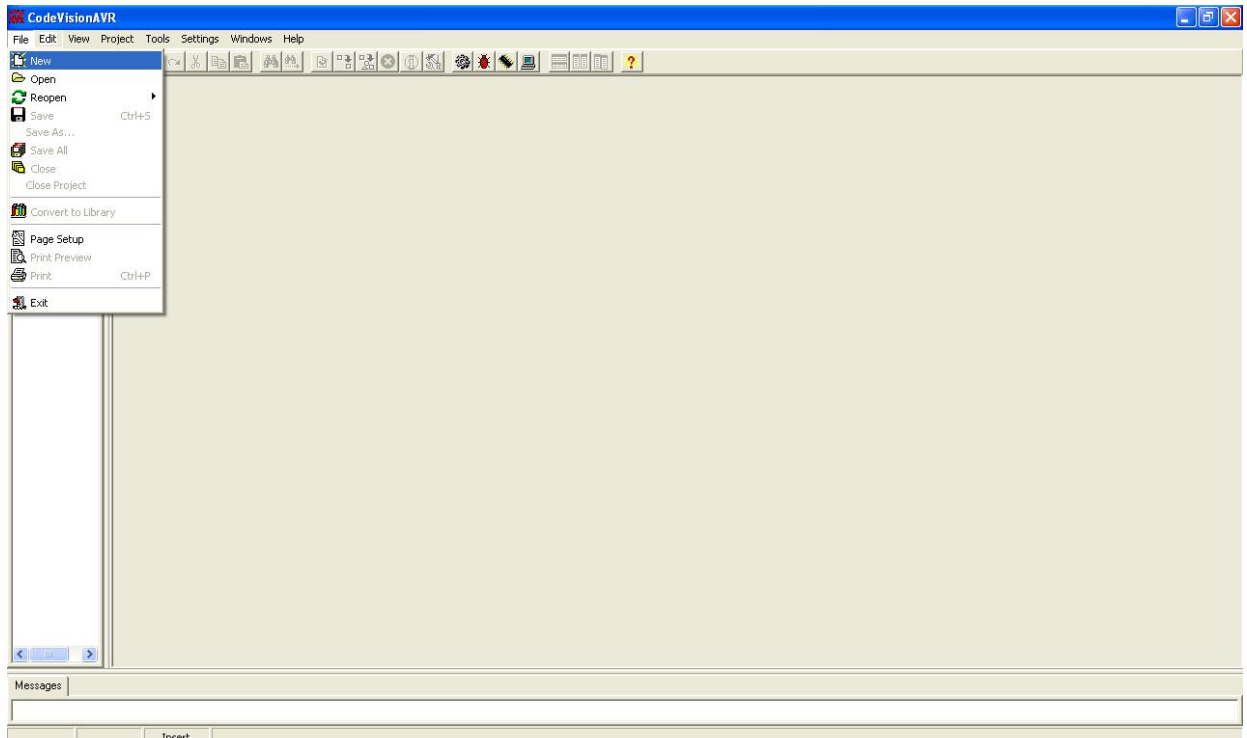
4.3.5 Παράδειγμα ανάπτυξης ενός προγράμματος για μικροελεγκτή με τη χρήση του CodeVision AVR.

Σε αυτή την παράγραφο παρατίθεται ένα απλό παράδειγμα ανάπτυξης κώδικα για τον μικροελεγκτή ATmega 16 και τον προγραμματισμό αυτού. Στο Σχ. 4.6 φαίνεται το περιβάλλον εργασίας στο CodeVision AVR από το οποίο ο χρήστης μπορεί να αρχίσει τη δημιουργία του προγράμματος (Project). Στη συνέχεια όπως φαίνεται στο Σχ. 4.7 ο χρήστης μπορεί να δημιουργήσει το πρόγραμμα , επιλέγοντας να αρχίσει ένα νέο “Project” (Σχ. 4.8) και στη συνέχεια καλείται να το ονομάσει (Σχ. 4.9). Αφού δημιουργηθούν τα αρχεία που απαρτίζουν το “Project” γίνονται οι απαραίτητες ρυθμίσεις και η παραμετροποίηση

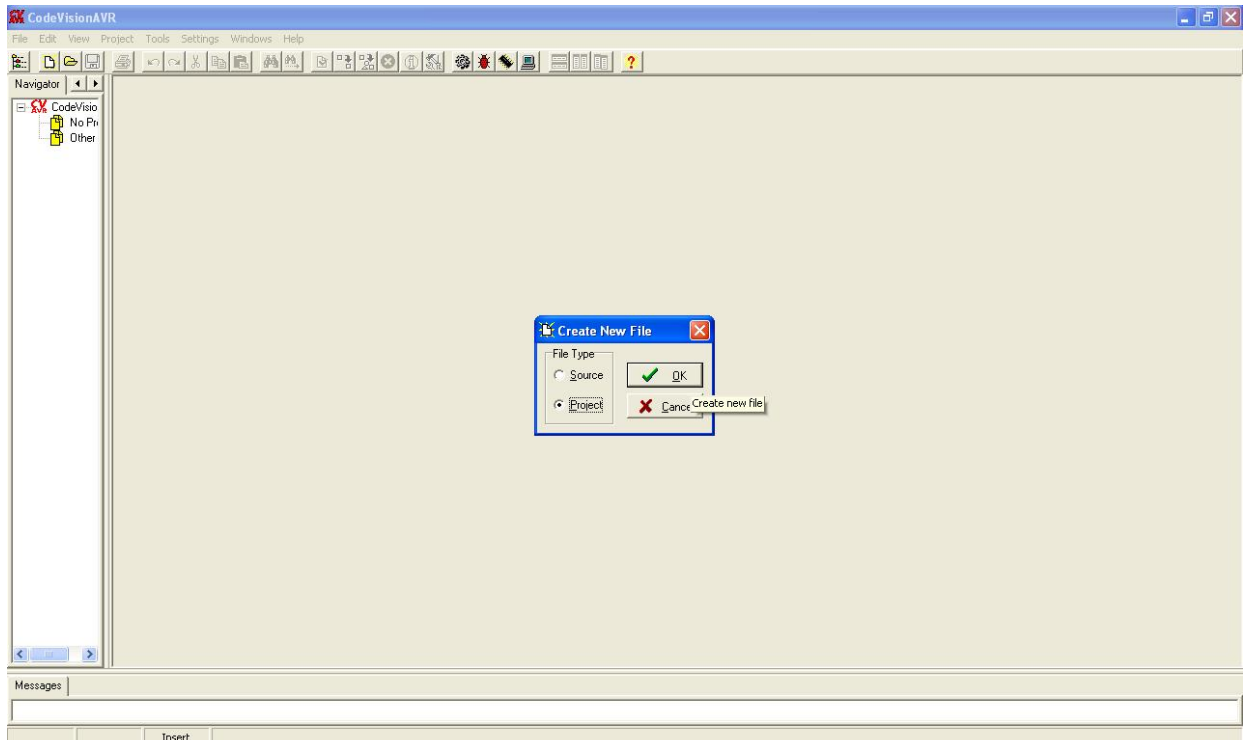
του προγράμματος (Σχ. 4.10). Έχοντας πλέον δημιουργήσει το "Project" ο χρήστης μπορεί να προσθέσει αρχεία κώδικα(source files) (Σχ. 4.11), τα αρχεία κώδικα μπορεί να είναι ήδη έτοιμα ή να αναπτυχθούν ξεκινώντας από κενό αρχείο(Σχ. 4.12). Όταν το πρόγραμμα έχει ολοκληρωθεί και μεταγλωττιστεί μεταφέρεται με τον ενσωματωμένο «προγραμματιστή»(Σχ. 4.13) στο μικροελεγκτή.



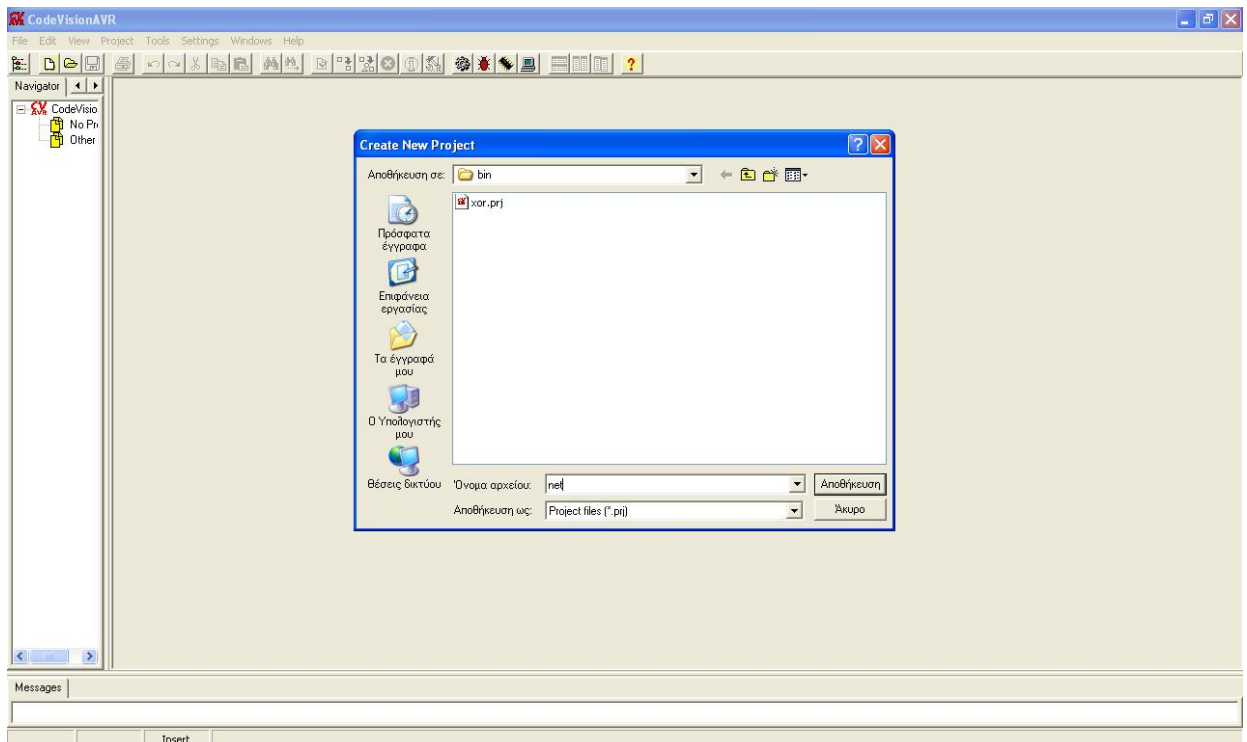
Σχ. 4.6 Το περιβάλλον εργασίας CodeVision AVR.



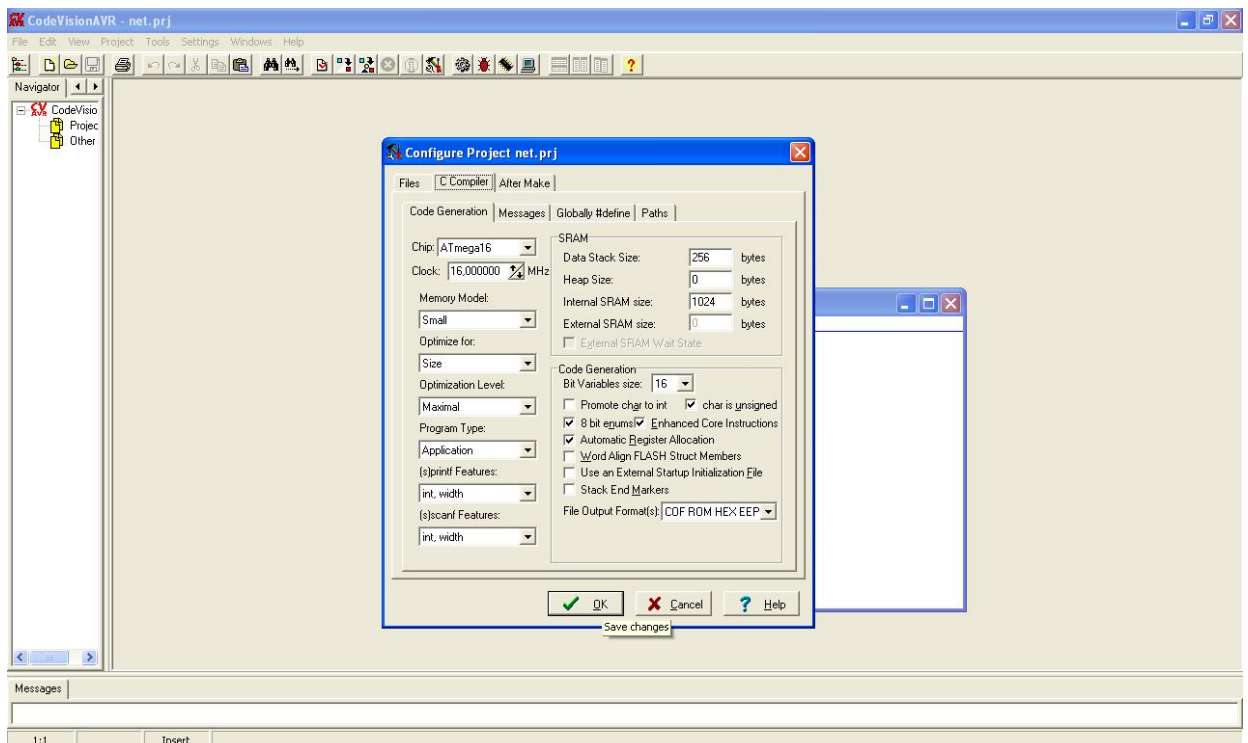
Σχ. 4.7 Δημιουργία νέου αρχείου.



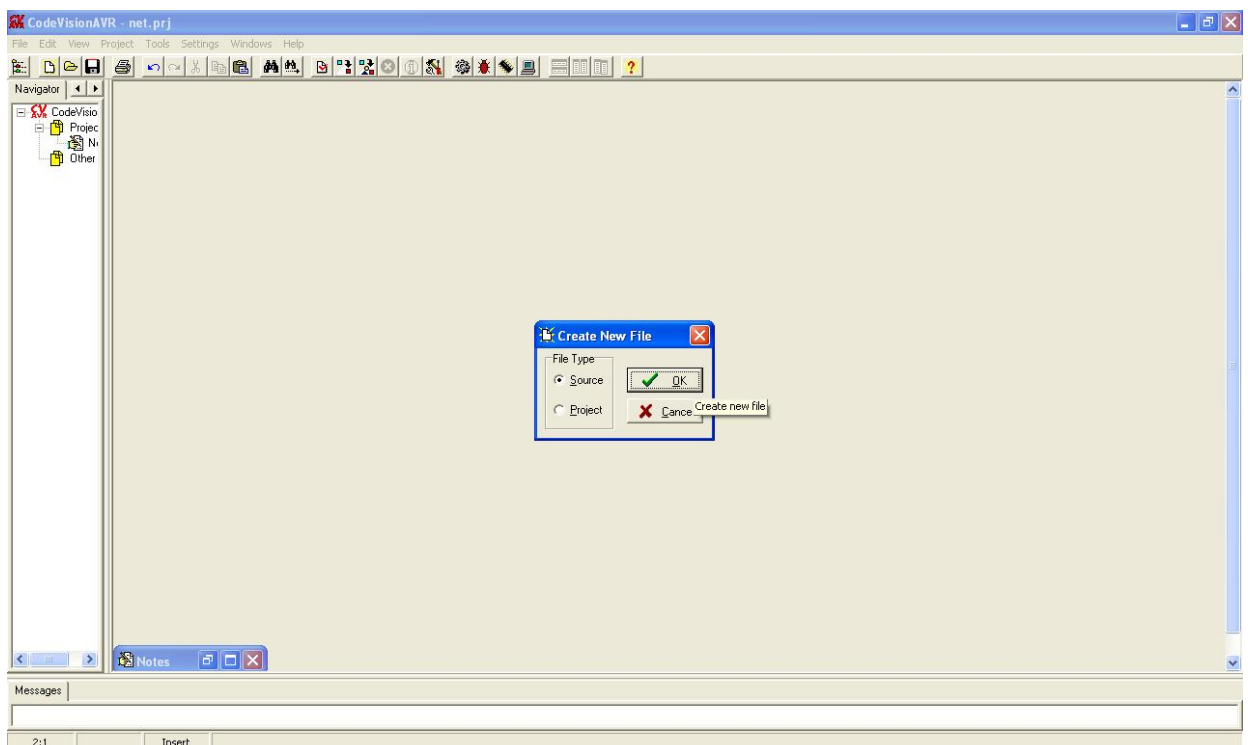
Σχ. 4.8 Επιλογή τύπου για το αρχείο που θα δημιουργηθεί.



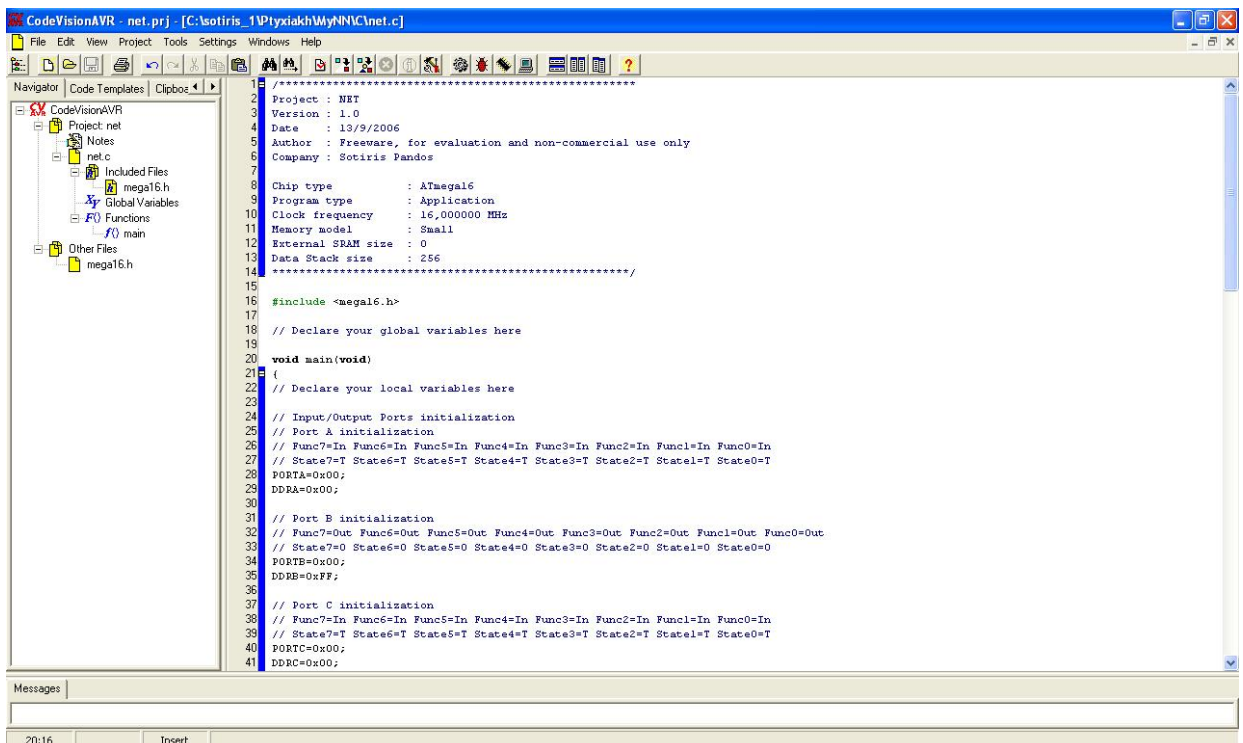
Σχ. 4.9 Ορισμός ονόματος για το αρχείο του προγράμματος.



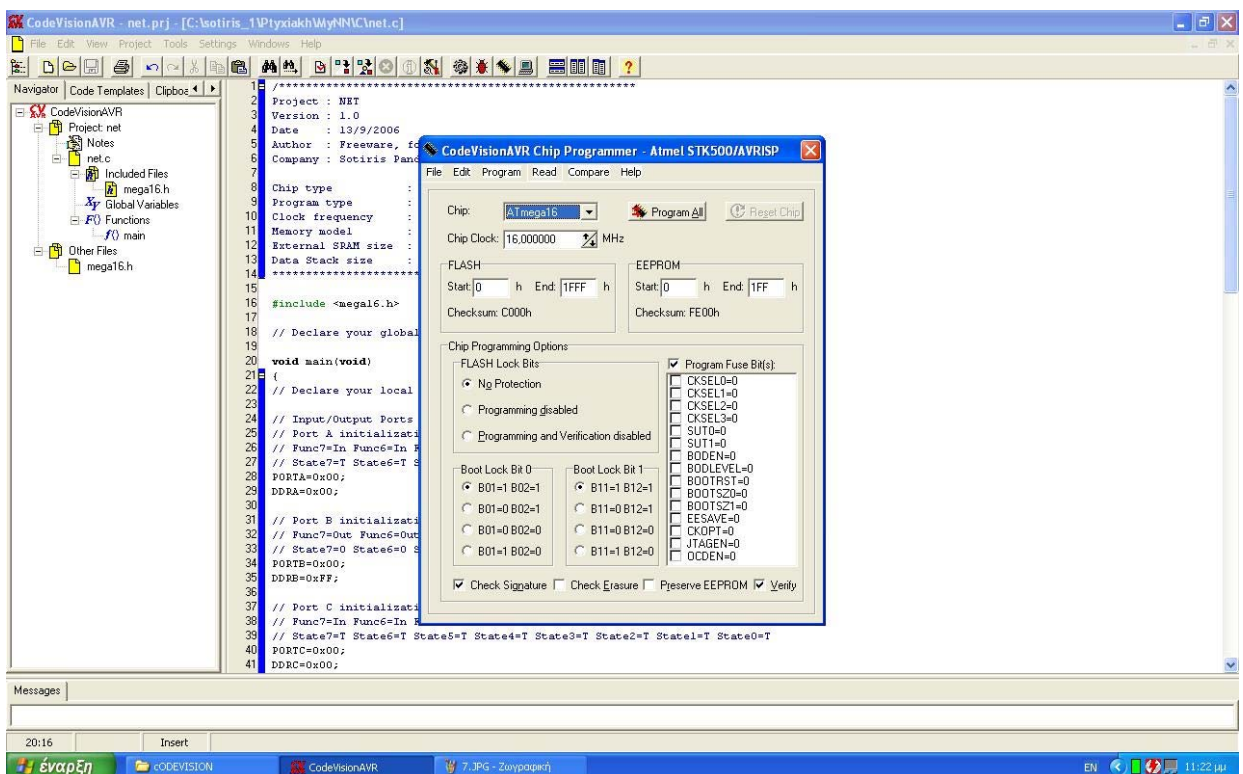
Σχ. 4.10 Ορισμός παραμέτρων προγράμματος.



Σχ. 4.11 Προσθήκη ενός αρχείου κώδικα.



Σχ. 4.12 Περιβάλλον ανάπτυξης και μεταγλωττισμού του κώδικα.



Σχ. 4.13 Προγραμματισμός του μικροελεγκτή μέσω του ενσωματωμένου προγραμματιστή.

Ακολουθεί ο κώδικας ενός απλού προγράμματος το οποίο μπορεί να αναβοσβήνει τα λαμπάκια σε ένα χριστουγεννιάτικο δέντρο. Να σημειώσουμε πως για να γίνει αυτό χρειάζεται και η κατάλληλη συνδεσμολογία με τα λαμπάκια (ή led). Ο συγκεκριμένος κώδικας είναι για μικροελεγκτή ATMEL 8515.

```

/*****
Programma pou anavei Xristougenniatika lampakia
*****/

/* Viliothikes */

#include <90s8515.h>
#include <stdlib.h>

/*****
Prototypa Synarthseon
*****/

void init_hardware(void);
void light_it_up(void);
void led_on_time(void);
void random_delay(void);

void main(void)
{
    init_hardware();

    while(1)
    {
        light_it_up();
        led_on_time();
        random_delay();
    }
}

void init_hardware (void)
{
    PORTA = 0xFF;
    DDRA = 0x00;
    DDRB = 0x00;
}

void light_it_up(void)
{
    unsigned char random_led, led;

    random_led = rand() & 7;

```



```

    led = 1 << random_led;
    DDRB = ~led;
    //DDRB = led;
}

```

```

void led_on_time(void)

```

```

{

    unsigned char i, j, k;
    for(i = 0; i < 255; i++)
        for(j = 0; j < 100; j++)
            k++;
    DDRB = 0xFF;
    //DDRB = 0x00;
}

```

```

void random_delay(void)

```

```

{
    unsigned char wait1, wait2, random_wait, l, random_off_delay;
    random_off_delay = rand() & 9;

    for(wait1 = 0; wait1 < 255; wait1++)
        for(wait2 = 0; wait2 < 255; wait2++)
            for(random_wait = 0; random_wait < random_off_delay;
random_wait++)
                l++;
}

```

4.4 Εφαρμογές Μικροελεγκτών.

Το πεδίο εφαρμογών των μικροελεγκτών είναι πρακτικά απεριόριστο, καθώς και να υπάρξει εφαρμογή στην οποία δεν μπορεί να ανταπεξέρθει ένας μικροελεγκτής, οι κατασκευαστές φροντίζουν ώστε να καλύψουν κάθε πιθανή ανάγκη κυκλοφορώντας νέα μοντέλα αλλά και κατά παραγγελία, σε ανάλογους αριθμούς βέβαια (πχ πάνω από 10000). Έτσι θα συναντήσουμε μικροελεγκτή ξεκινώντας από τις «σκεπτόμενες» οικιακές συσκευές και προγραμματιζόμενους διακόπτες μέχρι τα εξειδικευμένα επιστημονικά όργανα και τα ψηφιακά (διακοπτικά) τροφοδοτικά και γενικά σε οποιαδήποτε εφαρμογή υπάρχει ανάγκη ελέγχου ή καταγραφή και συλλογή δεδομένων υπάρχει τουλάχιστον ένας μικροελεγκτής. Ακόμη και ένας Η/Υ έχει, εκτός από τον κεντρικό επεξεργαστή, περιφερειακά τα οποία βασίζονται σε έναν μικροελεγκτή (πχ ελεγκτές δικτύων, routers, switches, ειδικές κάρτες για λήψη μετρήσεων, εκτυπωτές κλπ), απαλλάσσοντας πολλές φορές τον κεντρικό επεξεργαστή από την επιπλέον δουλειά. Το πεδίο που αφορά τον έλεγχο ηλεκτρικών μηχανών είναι ακόμη ένα στο οποίο οι μικροελεγκτές εφαρμόζονται ολοένα και περισσότερο, καθώς σήμερα τα ημιαγώγιμα στοιχεία τείνουν να αντικαταστήσουν πλήρως τα συμβατικά. Συνεπώς η καρδιά των συστημάτων που θα ελέγχουν όλα αυτά τα στοιχεία στην πλειονότητα των εφαρμογών είναι ο μικροελεγκτής.

5 ΠΡΟΣΟΜΟΙΩΣΗ ΤΝΔ ΜΕΣΩ MATLAB

5.1 Εξέλιξη Η/Υ και Λογισμικού.

Όπως έχουμε αναφέρει η εξέλιξη των νευρωνικών δικτύων την τελευταία 20-ετία είναι αλματώδης, σε μεγάλο βαθμό αυτή οφείλεται στην εξέλιξη των Η/Υ (τεχνολογίας), όπου ο Η/Υ δεν είναι μόνο το υλικό μέρος (hardware), αλλά και το λογισμικό (software). Το υλικό μέρος ενός Η/Υ παραμένει «αμετάβλητο» σε επίπεδο αρχιτεκτονικής από τα τέλη της δεκαετίας του 80, η υπολογιστική ισχύς όμως έχει αυξηθεί κατακόρυφα, κυρίως με βελτίωση της υπάρχουσας αρχιτεκτονικής των Η/Υ (π.χ. συχνότητα επεξεργαστή, μνήμη, μεγαλύτερη πυκνότητα ολοκλήρωσης, μεγαλύτερο εύρος διαύλου κλπ). Αντίθετα το λογισμικό όχι μόνο ακολουθεί ραγδαία πορεία εξέλιξης, αλλά αλλάζει και τον τρόπο με τον οποίο χρησιμοποιούμε τον Η/Υ μετατρέποντας έναν Η/Υ από άψυχη ψυχρή μηχανή σε ένα πλήρες περιβάλλον εργασίας, έτσι έχει αλλάξει ο τρόπος προσέγγισης των προβλημάτων με τη χρήση Η/Υ, προς το καλύτερο βέβαια.

Πλέον πειράματα τα οποία είναι αδύνατο να πραγματοποιηθούν, λόγω της φύσης τους (π.χ. σεισμός, έκρηξη supernova, πυρηνική έκρηξη κλπ) ή απαιτούν χιλιάδες χρόνια για να ολοκληρωθούν, μπορούν να προσομοιωθούν σε ικανοποιητικό βαθμό με τη χρήση υπερυπολογιστών²⁶. Αυτή η δυνατότητα υπήρχε βέβαια από την πρώτη στιγμή της δημιουργίας των Η/Υ, αυτό το οποίο έχει αλλάξει ριζικά χάρη στην εξέλιξη του λογισμικού κυρίως είναι πως πλέον ο επιστήμονας – ερευνητής δεν χρειάζεται να είναι και ειδήμονας στους Η/Υ για να καταφέρει να κάνει την δουλειά του.

5.1.1 Μέθοδοι Υλοποίησης Τ.Ν.Δ..

Για τη δημιουργία - ανάπτυξη ενός Τ.Ν.Δ. θεωρητικά υπάρχουν δύο οδοί. Η πρώτη και πολυδάπανη είναι η ανάπτυξη μοντέλου Τ.Ν.Δ. σε ολοκληρωμένο κύκλωμα (κλίμακας VLSI), δηλ. να δημιουργηθεί πάνω στο ολοκληρωμένο κύκλωμα μία πραγματική αποτύπωση του νευρωνικού δικτύου όπου οι ημιαγωγάτοι διακόπτες (transistor), μεμονωμένα ή ανά ομάδες, αποτελούν τους νευρώνες του μοντέλου. Η μέθοδος αυτή εκτός του ότι είναι εξαιρετικά δαπανηρή απαιτεί ιδιαίτερο τεχνολογικό εξοπλισμό τον οποίο πολύ λίγα ερευνητικά εργαστήρια (κυρίως μεγάλες εταιρείες) διαθέτουν όπως επίσης και εξεζητημένο γνωστικό επίπεδο σχεδιασμού ολοκληρωμένου κυκλώματος. Όπως αντιλαμβανόμαστε ένα τέτοιο μοντέλο είναι εξαιρετικά γρήγορο και αφορά ειδικές εφαρμογές, κυρίως σε πειραματικές διατάξεις.

Η δεύτερη και πιο οικονομική οδός είναι να προσομοιωθεί η λειτουργία ενός μοντέλου Τ.Ν.Δ. με τη χρήση λογισμικού σε έναν ή και περισσότερους επεξεργαστές (CPU). Το σημαντικό μειονέκτημα της δεύτερης μεθόδου είναι η ταχύτητα επεξεργασίας και το ότι ουσιαστικά δεν έχουμε παράλληλη λειτουργία. Το προηγούμενο μειονέκτημα αντισταθμίζεται από το εξαιρετικά χαμηλό κόστος και την ευκολία ανάπτυξης του μοντέλου, καθώς δεν απαιτείται ιδιαίτερα ακριβός εξοπλισμός και ένα (απλό) σύστημα Η/Υ εξοπλισμένο με μία γλώσσα προγραμματισμού 4^{ης} γενιάς (C, Pascal, κλπ) όπου μπορεί να κάνει πράξεις με πίνακες αρκεί. Υπάρχουν επίσης εξειδικευμένα (μαθηματικά) πακέτα τα οποία διευκολύνουν ακόμα περισσότερο το έργο μας, αναλαμβάνοντας πολλές φορές να εκτελούν τις επαναλαμβανόμενες διαδικασίες και παρέχοντας εξειδικευμένα εργαλεία για την ανάπτυξη και προσομοίωση ενός μοντέλου Τ.Ν.Δ. , πάντα κόστος όμως εις βάρος της ταχύτητας. Εύκολα γίνεται αντιληπτό πως στην παρούσα εργασία δεν πρόκειται να ασχοληθούμε με την πρώτη μέθοδο πάρα μόνο με την δεύτερη.

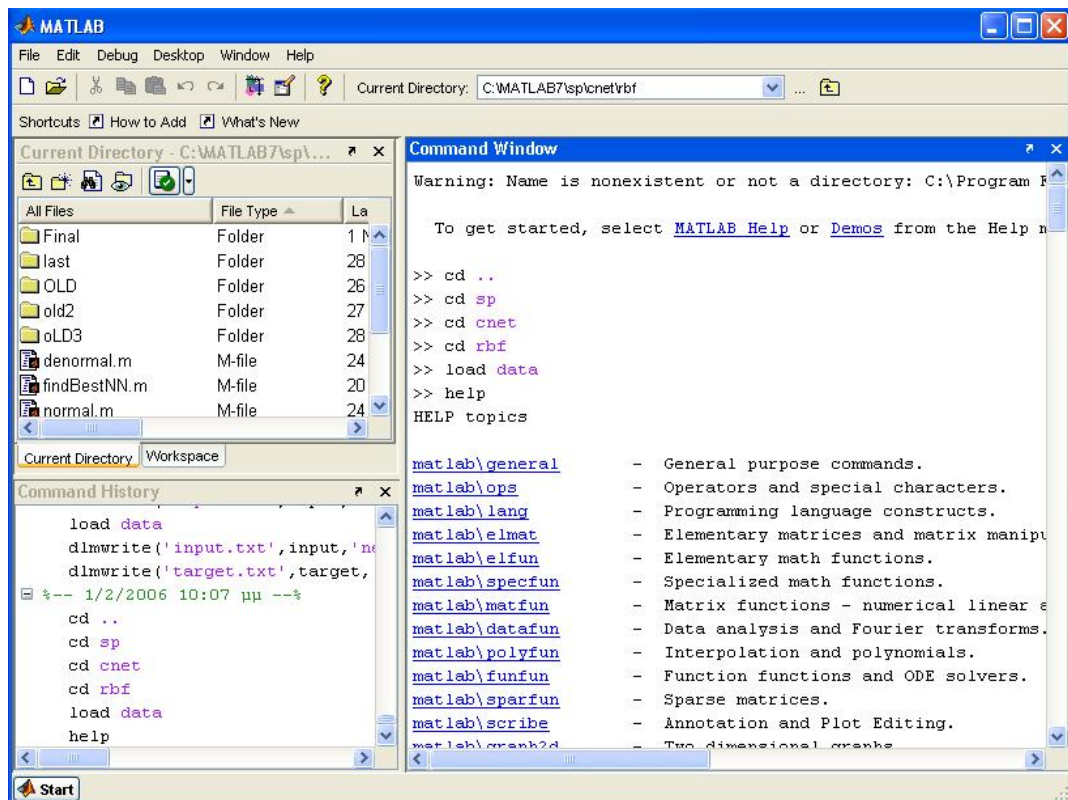
²⁶ Υπερυπολογιστές: είναι ειδικά εξειδικευμένοι Η/Υ με πολύ μεγάλη υπολογιστική ισχύ και χρησιμοποιούνται κυρίως σε ερευνητικά προγράμματα. Η αρχιτεκτονική δεν αλλάζει ιδιαίτερα σε σχέση με τους κοινούς Η/Υ, ακόμη και ένας προσωπικός υπολογιστής μπορεί να κάνει την ίδια δουλειά σε πολύ μεγαλύτερο χρόνο όμως, επίσης χαρακτηριστικό είναι πως οι υπερυπολογιστές τις δεκαετίας του 70, των πολλών εκατομμυρίων δολαρίων είχαν μικρότερη ισχύ από ένα σύγχρονο υπολογιστή παλάμης (calculator).

5.1.2 Λογισμικό ανάπτυξης ΤΝΔ – MATLAB.

Έχουν αναπτυχθεί πολλά προγράμματα (περιβάλλον εργασίας) τα οποία κάνουν ευκολότερη και γρηγορότερη την ανάπτυξη Τ.Ν.Δ., παράλληλα όμως είναι τόσο πολλά και διαφορετικά, που είναι αδύνατο να γνωρίζει κανείς να τα χρησιμοποιεί όλα. Γίνεται αναγκαία η αναφορά και σε αυτό τον τομέα, καθώς για την συγκεκριμένη μελέτη χρησιμοποιήθηκε εκτενώς το πρόγραμμα (περιβάλλον - εφαρμογή) MatLAB. Το MatLAB είναι ένα από τα ευρύτερα διαδομένα παγκοσμίως «μαθηματικό» περιβάλλον εργασίας για νευρωνικά δίκτυα, και όχι μόνο. Παρέχει τη δυνατότητα δημιουργίας, εκπαίδευσης και δοκιμής μοντέλων νευρωνικών δικτύων. Χαρακτηριστικό είναι πως η λέξη MATLAB είναι τα ακρωνύμια των λέξεων “matrix laboratory” «εργαστήριο (μαθηματικών) πινάκων». Στο πακέτο MATLAB περιλαμβάνονται αναρίθμητα επιπλέον εργαλεία όπου μπορούν να καλύψουν κάθε πτυχή οποιουδήποτε προβλήματος. Πιο συγκεκριμένα έχουμε :

- Μαθηματικούς υπολογισμούς.
- Ανάπτυξη αλγορίθμων.
- Δημιουργία δεδομένων.
- Μοντελοποίηση – Προσομοίωση και δημιουργία πρωτότυπων.
- Ανάλυση – επεξεργασία και παρουσίαση δεδομένων.
- Βιβλιοθήκες για την δημιουργία γραφικών παραστάσεων κλπ.

Για να πετύχουμε όλα τα παραπάνω, μπορούμε να χρησιμοποιήσουμε έτοιμες συναρτήσεις από τις βιβλιοθήκες του MATLAB ή να φτιάξουμε νέες. Εκτός από την δική του ενσωματωμένη «γλώσσα προγραμματισμού» (MatLAB Script), όπου μπορούν να φτιάξουμε συναρτήσεις, παρέχεται και η δυνατότητα να κληθούν εξωτερικές συναρτήσεις ή και ολόκληρα τμήματα κώδικα, πχ σε γλώσσα C. Όλα τα παραπάνω γίνονται με «φιλικό» γραφικό (ή παραθυρικό) περιβάλλον (GUI) εργασίας (Σχ. 5.1).



Σχ. 5.1 Περιβάλλον εργασίας “MATLAB”

5.1.3 Δημιουργία Μοντέλου T.N.Δ.

Η δημιουργία ενός T.N.Δ. με το MatLab γίνεται με τη χρήση των κατάλληλων συναρτήσεων και εργαλείων (toolboxes), συγκεκριμένα για τα νευρωνικά δίκτυα χρειάζεται το “NNTool”. Απαραίτητη προϋπόθεση για τη δημιουργία του μοντέλου είναι η ύπαρξη των κατάλληλων δεδομένων, διανύσματα εισόδου, εξόδου, εκπαίδευσης κλπ. Στο MatLab μπορεί να δημιουργηθεί ένα μοντέλο μέσω του γραφικού περιβάλλοντος (GUI – Graphics User Interface) όπου είναι ο πλέον απλοποιημένος τρόπος, με τα αρνητικά που αυτό συνεπάγεται (π.χ. λιγότερη παραμετροποίηση). Εναλλακτικά μπορεί να γίνει μία πιο εξειδικευμένη προσέγγιση, όπου καταχωρούνται σχεδόν όλες τις τιμές και τις εντολές στο παράθυρο εντολών (command window). Ο δεύτερος τρόπος απαιτεί μία σχετικά μεγαλύτερη εξοικείωση με την εφαρμογή, αλλά είναι πιο γρήγορος και οι εντολές ενσωματώνονται πιο εύκολα στον κώδικα, οπότε και αυτός προτιμάται στην παρούσα εργασία.

5.1.3.1 Δημιουργία Μοντέλου “Feed Forward Back Propagation”

Για να γίνουν τα παραπάνω πιο εύκολα κατανοητά στη συνέχεια περιγράφεται η διαδικασία δημιουργίας ενός μοντέλου “FFBPG” (Feed Forward Back Propagation - MLP), η διαδικασία είναι παρόμοια για κάθε μοντέλο, χρησιμοποιώντας τις κατάλληλες συναρτήσεις κάθε φορά. Τα απαραίτητα διανύσματα εισόδου και εκπαίδευσης αποθηκεύονται σε δύο μεταβλητές (πίνακες), η είσοδος στην μεταβλητή “input” και η επιθυμητή έξοδος στην “target”. Η πραγματική έξοδος του μοντέλου αποθηκεύεται στην μεταβλητή “output”, κάθε μεταβλητή είναι πίνακας και ενώ η είσοδος μπορεί να είναι πολυδιάστατος πίνακας, ο στόχος και η έξοδος στην παρούσα εργασία θα είναι μονοδιάστατοι πίνακες. Για ένα μοντέλο “Feed-Forward” χρησιμοποιούμε την συνάρτηση “newff”.

Η συνάρτηση “newff” συντάσσεται ως εξής :

```
Net = newff (PR, [S1 S2...SN1], {TF1 TF2...TFN1}, BTF, BLF, PF)
```

Όπου:

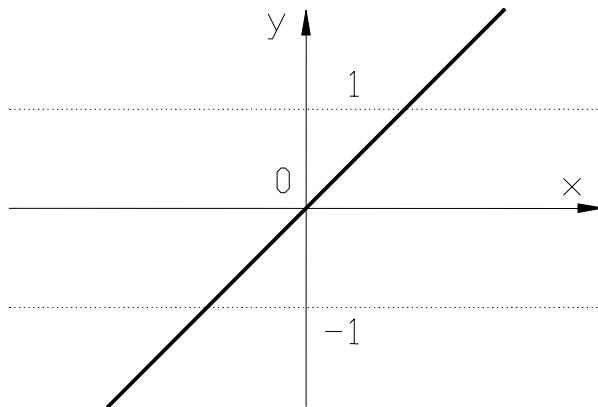
- **Net** : το όνομα του μοντέλου T.N.Δ. που θα δημιουργηθεί.
- **PR** :πίνακας ελάχιστων και μέγιστων τιμών των διανυσμάτων εισόδου.
- **S1 ... SN1** : μέγεθος κάθε επιπέδου (αριθμός νευρώνων ανά επίπεδο).
- **TF1...TFN1** : συνάρτηση ενεργοποίησης (ή μεταφοράς) των νευρώνων του αντίστοιχου επιπέδου.
- **BTF** : συνάρτηση εκπαίδευσης του μοντέλου,(εξ ορισμού χρησιμοποιείται η trainlm)
- **BLF** : συνάρτηση (αλγόριθμος) μεταβολής των βαρών και της πόλωσης των νευρώνων (weight, bias).
- **PF** : συνάρτηση ελέγχου απόδοσης (σφάλματος), συνήθως “MSE”, “SSE” κλπ.

Μετά την εκτέλεση της παραπάνω διαδικασίας το μοντέλο έχει δημιουργηθεί, αλλά ακόμη δεν είναι έτοιμο να επεξεργαστεί σωστά τα δεδομένα που θα του εισάγουμε, πρέπει πρώτα να εκπαιδευτεί. Πριν φθάσουμε στην εκπαίδευση, ακολουθώντας τη λογική σειρά του αλγόριθμου, θα κάνουμε μία αναφορά στις βασικές συναρτήσεις ενεργοποίησης των νευρώνων.

5.1.3.2 Συναρτήσεις ενεργοποίησης

Οι (συνήθεις) συναρτήσεις ενεργοποίησης που θα χρησιμοποιηθούν είναι :

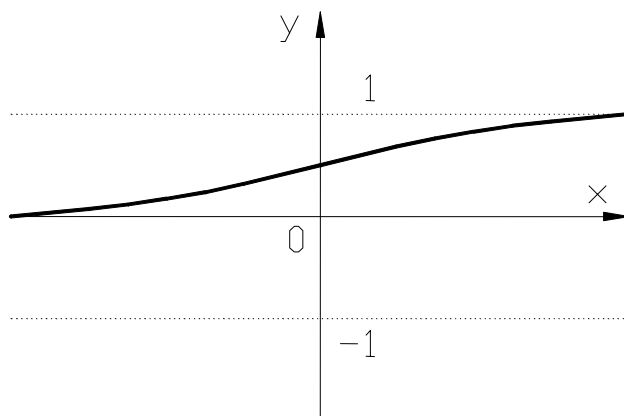
- $purelin(n) = n$, είναι γραμμική συνάρτηση μεταφοράς (Σχ. 5.2), όπου ουσιαστικά η έξοδος παίρνει την τιμή της εισόδου, αυτό βέβαια δεν είναι απαραίτητο να συμβαίνει σε κάθε γραμμική συνάρτηση μεταφοράς. Τη συνάρτηση αυτή συνήθως τη χρησιμοποιούμε για γραμμικούς νευρώνες του πρώτου επιπέδου ενός Τ.Ν.Δ., αλλά και σε ειδικές περιπτώσεις, σε νευρώνες άλλων επιπέδων.



Σχ. 5.2 Συνάρτηση μεταφοράς (ενεργοποίησης) "Purelin".

- $log\ sig(n) = \left(\frac{1}{1 + \exp(-n)} \right)$, (Logarithmic sigmoid transfer function), πρόκειται για λογαριθμική (σιγμοειδής)

συνάρτηση μεταφοράς. Όπως φαίνεται και στο Σχ. 5.3, το πεδίο τιμών της είναι στο [0 1], αυτό το λαμβάνουμε υπόψη όταν θα την χρησιμοποιήσουμε, γιατί αν στο επίπεδο εξόδου οι νευρώνες χρησιμοποιούν την "LogSig", τότε η έξοδος του Τ.Ν.Δ. μπορεί να πάρει τιμές μόνο στο πεδίο τιμών της συνάρτησης [0 1]. Ιδανική για χρήση σε Τ.Ν.Δ. προσομοίωσης λειτουργίας μιας πύλης, πχ XOR, και γενικά σε προβλήματα που έχουμε μόνο δύο καταστάσεις.

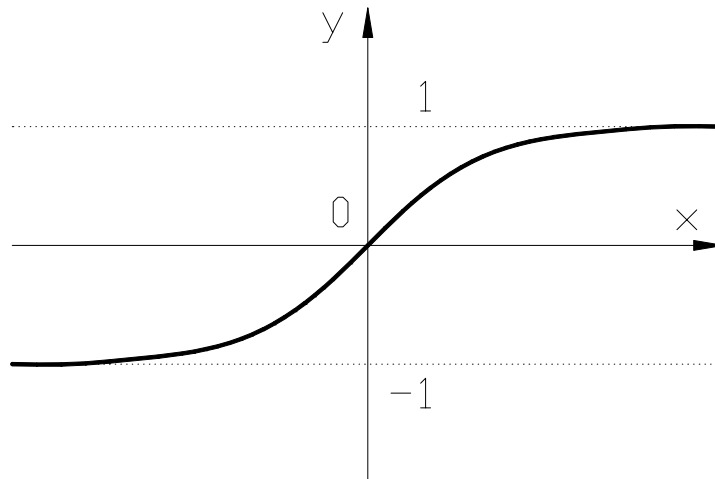


Σχ. 5.3 Συνάρτηση μεταφοράς (ενεργοποίησης) "LogSig".

- $\text{tansig}(n) = \frac{2}{(1 + \exp(-2n)) - 1}$, (Hyperbolic tangent sigmoid transfer function), είναι ίδια με την σιγμοειδή

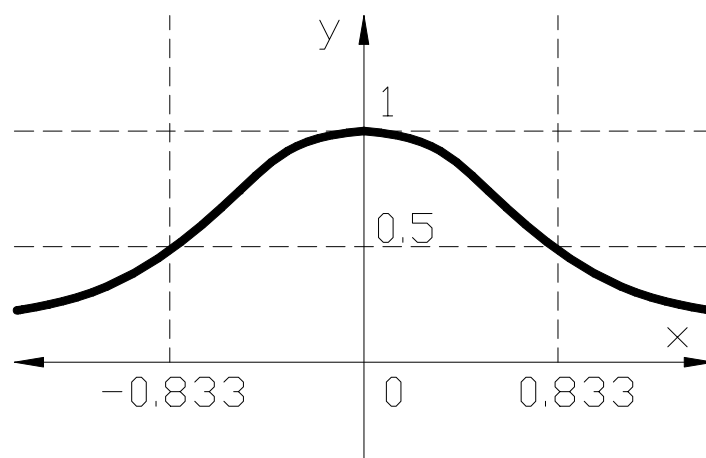
συνάρτηση υπερβολικής εφαπτομένης $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, με τη διαφορά ότι είναι πολύ πιο γρήγορη, αλλά

υπολείπεται σε ακρίβεια σε εφαρμογές όπου απαιτείται μεγαλύτερη ακρίβεια. Στην περίπτωση μας χρησιμοποιούμε την tansig , καθώς η ακρίβεια που παρέχει είναι ικανοποιητική για την εφαρμογή όπως και η ταχύτητα, πράγμα απαραίτητο για ένα νευρωνικό δίκτυο. Όπως φαίνεται και στο Σχ. 5.4 η tansig έχει πεδίο τιμών $[-1, 1]$, οπότε αν στους νευρώνες εξόδου η συνάρτηση μεταφοράς είναι η tansig τότε και η έξοδος του Τ.Ν.Δ. θα είναι στο πεδίο $[-1, 1]$.



Σχ. 5.4 Συνάρτηση μεταφοράς (ενεργοποίησης) "Tansig"

- $\text{radbas}(n) = \exp(-n^2)$, (Radial basis transfer function), η "radbas" είναι συνάρτηση μεταφοράς που χρησιμοποιείται κυρίως σε νευρωνικά δίκτυα τύπου RBF. Από τον αλγόριθμο της συνάρτησης, αλλά και το Σχ. 5.5 γίνεται αντιληπτό πως η τιμές στην έξοδο της συνάρτησης "radbas" δεν υπερβαίνουν τη μονάδα, γίνεται 1 όταν στην είσοδο έχουμε 0.



Σχ. 5.5 Συνάρτηση μεταφοράς (ενεργοποίησης) "radbas".

- Παρέχονται και άλλες συναρτήσεις μεταφοράς, όπως η "softmax", "hardlim", "satlin" κλπ., όπου η κάθε μία έχει αναπτυχθεί για ειδική εφαρμογή και μοντέλο νευρωνικού δικτύου. Περαιτέρω αναφορά την παρούσα μελέτη σε αυτές δεν κρίνεται σκόπιμη, εφόσον δε χρησιμοποιείται κάποια από αυτές στην υλοποίηση του νευρωνικού δικτύου.

5.1.3.3 Συναρτήσεις εκπαίδευσης

Ήδη έχει γίνει μία μικρή αναφορά στην εκπαίδευση του νευρωνικού δικτύου η οποία αποτελεί το σημαντικότερο παράγοντα για την σωστή λειτουργία (ικανότητα γενίκευσης) του μοντέλου. Για το λόγο αυτό είναι ιδιαίτερα σημαντικός ο αλγόριθμος εκπαίδευσης που θα επιλέξουμε. Στην βιβλιοθήκη του MatLAB υπάρχει πλειάδα συναρτήσεων που «αναλαμβάνουν» την εκπαίδευση του T.N.Δ.. Όλες μας παρέχουν τη δυνατότητα μεταβολής των παραμέτρων εκπαίδευσης, όπως ρυθμός μάθησης, ελάχιστο όριο σφάλματος. Για να κληθεί μία συνάρτηση εκπαίδευσης είναι απαραίτητη προϋπόθεση να δοθεί το όρισμα σε κάθε συνάρτηση. Στη συγκεκριμένη περίπτωση τα έχουν αποθηκευτεί στις μεταβλητές “input” και “target” (διανύσματα εκπαίδευσης).

Η σύνταξη της εντολής έχει ως εξής :

```
Net_2 = train(Net,input,target);
```

Όπου `Net_2` είναι το εκπαιδευμένο μοντέλο το οποίο θα προκύψει (ξεκινώντας από τις συνθήκες του `Net`), από τα παραδείγματα που υπάρχουν στους πίνακες “input” και “target”, όπου οι δύο πίνακες πρέπει να έχουν το ίδιο μήκος. Για να μεταβάλουμε τα μεγέθη της συνάρτησης εκπαίδευσης δίνουμε την εντολή `Net.trainParam` και στη συνέχεια επιλέγουμε πιο στοιχείο θα μεταβάλουμε. όπως φαίνεται και στο παρακάτω παράδειγμα :

```
Net.trainParam
ans =
epochs: 100
goal: 0
max_fail: 5
mem_reduc: 1
min_grad: 1.0000e-010
mu: 1.0000e-003
mu_dec: 0.1000
mu_inc: 10
mu_max: 1.0000e+010
show: 25
time: Inf
```

Έτσι αν θέλουμε να ορίσουμε διαφορετικά κάποια τιμή από της προκαθορισμένες, πχ των αριθμό εποχών, έχουμε :

```
Net.trainParam.epochs=155;
```

Όποτε αν τώρα ξαναδώσουμε την εντολή `Net.trainParam`

Έχουμε :

```
>> Net.trainParam
ans =
    epochs: 155
```

```

goal: 0
max_fail: 5
mem_reduc: 1
min_grad: 1.0000e-010
mu: 1.0000e-003
mu_dec: 0.1000
mu_inc: 10
mu_max: 1.0000e+010
show: 25
time: Inf

```

Κατά τον ίδιο τρόπο μπορούμε να μεταβάλουμε οποιαδήποτε άλλη τιμή.

Από τις συναρτήσεις εκπαίδευσης, στις δοκιμές θα χρησιμοποιηθούν κυρίως δύο :

- **“trainlm”**, Levenberg-Marquardt²⁷ backpropagation, η συνάρτηση αυτή χρησιμοποιεί τον αλγόριθμο “Levenberg-Marquardt” για την διόρθωση των βαρών (weights) και την πόλωση των νευρώνων (bias). Όπως περιγράφεται από την Εξ. 5.1 με τη χρήση αυτής της συνάρτησης έχουμε έναν αλγόριθμο για την εύρεση του ελάχιστου της συνάρτησης. Ουσιαστικά έχουμε εκπαίδευση με τη χρήση του όρου “learning rate”. Όταν το σφάλμα γίνει ίσο με το όριο που έχουμε θέσει, τότε θεωρείται ότι η εκπαίδευση έχει ολοκληρωθεί οπότε η διαδικασία σταματάει. Η εκπαίδευση μπορεί να διακοπεί και από άλλους παράγοντες, όπως ο μέγιστος αριθμός εποχών, ο χρόνος εκπαίδευσης, ο ρυθμός μεταβολής της συνάρτησης σφάλματος κλπ. Βέβαια με τη χρήση μόνο του “learning rate” υπάρχει κίνδυνος εγκλωβισμού σε τοπικό ελάχιστο ή ταλάντωσης γύρω από κάποιο ελάχιστο.

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (5.1)$$

- **“traingdx”**, Gradient descent w/momentum & adaptive lr backpropagation, η συνάρτηση “traingdx”, χρησιμοποιεί έναν επιπλέον όρο στην εκπαίδευση, τον όρο της ορμής (momentum term²⁸) και μεταβλητό ρυθμό μάθησης. Όπως περιγράφεται από την Εξ. 5.2²⁹ με τη χρήση του όρου της ορμής αυξάνεται η ταχύτητα εκπαίδευσης και αποφεύγεται ο εγκλωβισμός σε τοπικά ελάχιστα ή η ταλάντωση σε κάποιο ελάχιστο. Επιπλέον στην αύξηση της ταχύτητας συμβάλει και ο μεταβλητός ρυθμός μάθησης. Εκτός των άλλων παραμέτρων τώρα καλούμαστε να ορίσουμε όχι μόνο ρυθμό μάθησης “learning rate”, αλλά και ρυθμό μεταβολής του ρυθμού μάθησης (μέγιστο, ελάχιστο, ρυθμό αύξησης και μείωσης) όπως και τη σταθερά ορμής (Momentum constant, mc). Η εκπαίδευση θεωρείται επιτυχής όταν επιτευχθεί το όριο σφάλματος που έχουμε θέσει, αλλά πάλι μπορεί να διακοπεί από τους ανωτέρω παράγοντες(χρόνος, εποχές, ρυθμός μεταβολής σφάλματος κλπ.).

$$dX(n+1) = mc \cdot dX(n) + lr \cdot mc \frac{dperf}{dX(n)} \quad (5.2)$$

²⁷ Ο αλγόριθμος παρουσιάζεται στο παράρτημα

²⁸ Επίδραση της προηγούμενης μεταβολής βαρών στην επόμενη

²⁹ Στην μεταβλητή X περιέχονται τα βάρη(weights) ή και τα όρια πόλωσης (bias).

5.1.3.4 Έλεγχος Απόδοσης Μοντέλου

Για τον έλεγχο (σφάλματος) της απόδοσης του Τ.Ν.Δ. χρησιμοποιείται η συνάρτηση “MSE” (Εξ. 5.3, mean squared error, μέσος όρος τετραγώνων σφαλμάτων). Το MatLAB μας παρέχει και άλλες συναρτήσεις, όπως η “SSE” (Εξ. 5.4, Sum squared error), “MAE” (Εξ. 5.5, Mean absolute error) κλπ. Ανάλογα με τον τύπο του νευρωνικού δικτύου χρησιμοποιείται η κατάλληλη συνάρτηση, πχ η συνάρτηση “SSE” χρησιμοποιείται κυρίως σε μοντέλα RBF. Ο έλεγχος της απόδοσης ενός μοντέλου είναι το πιο συνηθισμένο κριτήριο επιλογής της εκπαίδευσης, δηλαδή όταν η απόδοση του μοντέλου πάρει την επιθυμητή τιμή, τότε θεωρούμε ότι το μοντέλο εκπαιδεύτηκε και η εκπαίδευση σταματάει. Η επιλογή της κατάλληλης τιμής είναι πολύ σημαντική, γιατί αν το μοντέλο σταματήσει σε μεγάλο σφάλμα δε συγκλίνει όσο θέλουμε, ενώ αν το σφάλμα είναι πολύ μικρό υπάρχει ο κίνδυνος της υπερεκπαίδευσης.

$$MSE = \frac{1}{N} \sum_{i=1}^N (T - O)^2 \quad (5.3)$$

$$SSE = \sum_{i=1}^n (y_i - f_{(x_i)})^2 \quad (5.4)$$

$$MAE_i = \frac{1}{n} \sum_{j=1}^n |P_{(ji)} - T_i| \quad (5.5)$$

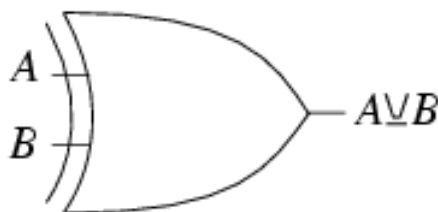
5.1.4 Η Γλώσσα Προγραμματισμού C.

Το MatLAB, όπως είπαμε, αποτελεί ένα από τα δημοφιλέστερα μαθηματικά (και όχι μόνο) πακέτα λογισμικού παρέχοντας μεγάλη ευελιξία και ευκολία στη δημιουργία – επίλυση μαθηματικών μοντέλων, εντούτοις δεν τα καταφέρνει εξίσου καλά στον τομέα της ταχύτητας σε ορισμένες εφαρμογές, για το λόγο αυτό πολλές φορές είναι αναγκαία η μεταφορά του μοντέλου και σε άλλη γλώσσα προγραμματισμού σχετικά χαμηλού επιπέδου (πχ C). Στην περίπτωση μας βέβαια η C αποτελεί αναπόσπαστο εργαλείο της όλης προσπάθειας, γιατί σε C θα γραφτεί το μοντέλο προκειμένου να μπορεί να προγραμματιστεί ο μικροελεγκτής (AVR).

Η C είναι μια γενικής χρήσης γλώσσα προγραμματισμού που χαρακτηρίζεται από οικονομία στην έκφραση, σύγχρονη ροή ελέγχου και δομές δεδομένων, κι ένα μεγάλο σύνολο τελεστών. Αρχικά σχεδιάστηκε για λειτουργικό UNIX με επιρροές από τη γλώσσα BCPL, σύντομα βέβαια επεκτάθηκε σε όλα τα συστήματα και ο ορισμός της στο πρότυπο ANSI (ANSI C) έγινε 1988 από το Αμερικάνικο Εθνικό Ινστιτούτο Προτύπων (ANSI). Δεν χαρακτηρίζεται ως γλώσσα «υψηλού επιπέδου», αντίθετα είναι σχετικά «χαμηλού επιπέδου», χωρίς αυτό να είναι υποτιμητικό. Αυτό σημαίνει ότι η C ασχολείται με χαρακτήρες, αριθμούς και διευθύνσεις, όλα αυτά μπορούν να συνδυάζονται και να μεταφέρονται με τους αριθμητικούς και λογικούς τελεστές που υποστηρίζονται από τα πραγματικά μηχανήματα. Ανήκει στην κατηγορία του δομημένου προγραμματισμού, (πχ η C++ είναι αντικειμενοστραφής) και παρέχει δυνατότητες πράξεων μεταξύ πινάκων. Επίσης παρέχει θεμελιώδεις δομές ελέγχου ροής που απαιτούνται για καλά δομημένα προγράμματα (ομαδοποίηση εντολών, if-else, switch – case, while, for, break κλπ.)

5.2 Το πρόβλημα της “XOR”.

Το πρώτο πρόβλημα το οποίο θα επιλύσουμε με ένα νευρωνικό δίκτυο έχει τη μορφή των πιο συχνών προβλημάτων που υπάρχουν στη φύση, πρόκειται για μία λογική συνάρτηση ,την “XOR” (ή αλλιώς πύλη XOR, Σχ. 5.6), όπου ως γνωστό στις εισόδους τις μπορεί να έχει τιμές “0” ή “1”. Μία συνάρτηση XOR έχει μη μηδενική έξοδο, δηλ. “1”, όταν έχει δύο διαφορετικές εισόδους (όταν πρόκειται για XOR με 2 εισόδους) και έξοδο “0” όταν στις εισόδους έχει ίδιες τιμές.



Σχ. 5.6 Πύλη “XOR”

Η προσομοίωση της “XOR” έγινε με ένα μοντέλο “Feed Forward Back Propagation ” (MLP), και η εκπαίδευση έγινε με επίβλεψη με τη χρήση μόνο τις παραμέτρου “learning rate” (ρυθμός μάθησης). Η συνάρτηση ενεργοποίησης των νευρώνων είναι η “logsig”, σε αυτό βοηθάει το πεδίο τιμών το οποίο είναι μεταξύ [0 1], όπως φαίνεται και από τον Πίνακα 5.1 (πίνακας «αληθείας») της πύλης.

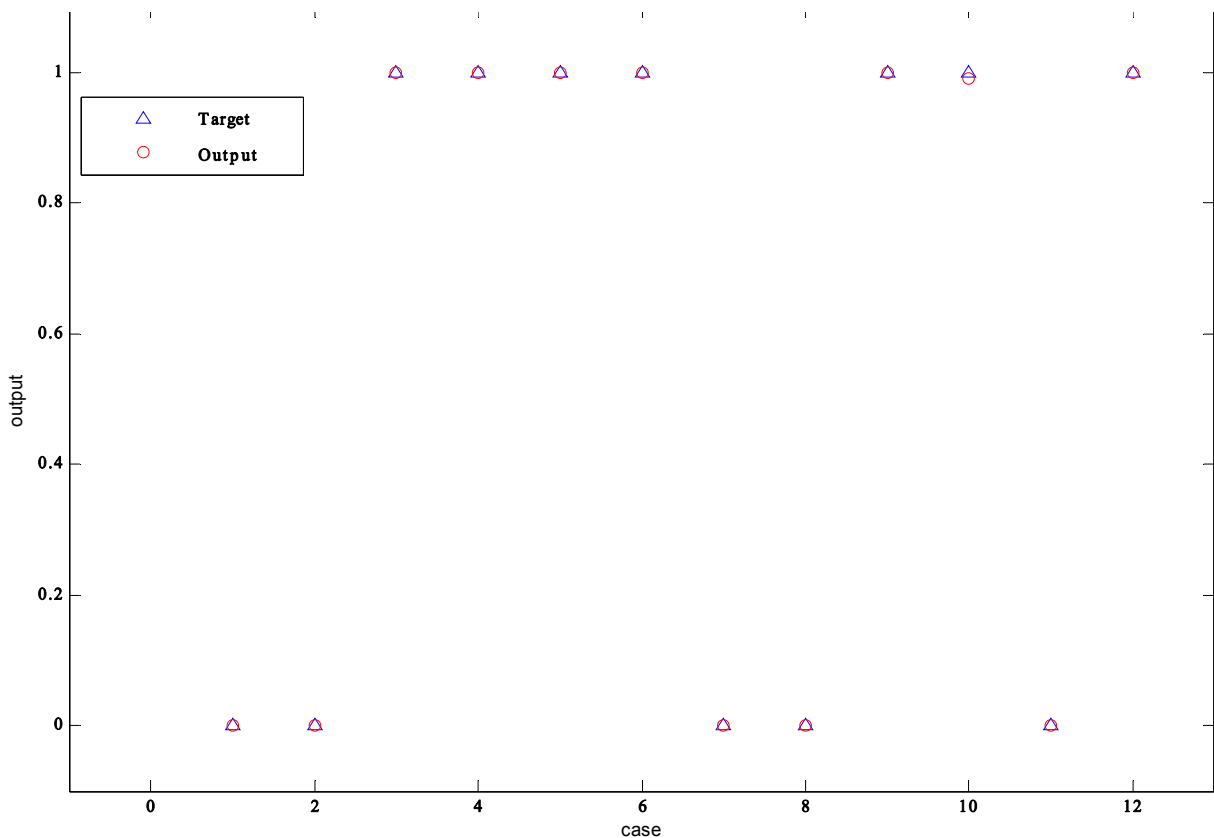
Πίνακας 5.1 Πίνακας αληθείας πύλης XOR

Καταστάσεις Συνάρτησης XOR		
Είσοδος		Έξοδος
1	1	0
1	0	1
0	1	1
0	0	0

Το μοντέλο αποτελείται από τρία επίπεδα, Εισόδου – Κρυφό – Εξόδου, όπου στο πρώτο επίπεδο υπάρχουν 2 νευρώνες, στο κρυφό 3 και στην έξοδο ένας, δηλ. έχουμε μοντέλο [2x3x1]. Η εκπαίδευση ενός τέτοιου μοντέλου είναι πολύ απλή και ολοκληρώνεται σε κλάσματα δευτερολέπτου με ποσοστό επιτυχίας που φτάνει στο 99%. Να τονίσουμε εδώ πως αν οι τιμές τις οποίες δοκιμάζουμε το μοντέλο ήταν οι ιδανικές, δηλαδή “0” ή “1” τότε το νευρωνικό δίκτυο δε θα είχε κανένα νόημα, έτσι καταχωρούμε δεδομένα με δεκαδικούς και ακέραιους αριθμούς. Τα αποτελέσματα φαίνονται στο παρακάτω γράφημα (Σχ. 5.7) και στον Πίνακα 5.2 (πίνακας τιμών) ο οποίος προέκυψε από το εκτελέσιμο αρχείο του μοντέλου σε κώδικα C.

Πίνακας 5.2 Αποτελέσματα μοντέλου T.N.Δ. “XOR” σε κώδικα C

Είσοδος	0	1	1	0	0	0,1	0,9	0,1	0,75	0,2	0	1
	0	1	0	1	1	0,9	0,85	0,25	0,3	1	0,25	0,25
Στόχος	0	0	1	1	1	1	0	0	1	1	0	1
Έξοδος	0	0	1	1	1	1	0	0	1	0,99	0	1



Σχ. 5.7 Έξοδος μοντέλου συνάρτησης "XOR"

5.3 Το πρόβλημα "IRIS".

Το επόμενο πρόβλημα αφορά στην ταξινόμηση λουλουδιών στο ανάλογο είδος (IRIS³⁰). Το πρόβλημα τώρα είναι πιο σύνθετο από το προηγούμενο, καθώς τώρα έχουμε τέσσερα χαρακτηριστικά για κάθε είδος, μήκος και πλάτος στέπαλου και μήκος και πλάτος πέταλου (όλα μετρημένα σε mm, *sepal length*, *sepal width*, *petal length*, *petal width*). Με βάση αυτά τα δεδομένα πρέπει το κάθε λουλούδι να καταταγεί σε ένα από τα τρία είδη (*Iris setosa*, *Iris versicolor*, *Iris virginica*).

Λόγω της πολυπλοκότητας που παρουσιάζει το πρόβλημα αυτό δεν είναι δυνατό να καταλήξουμε στο κατάλληλο μοντέλο με απλές διαδικασίες, αλλά αυτό θα προκύψει μετά από πειραματικές δοκιμές και συγκρίσεις πολλών μοντέλων. Για να επιτευχθεί αυτό θα δοκιμάσουμε μοντέλα με ένα και με δύο κρυφά επίπεδα. Το μοντέλο με ένα κρυφό επίπεδο θα έχει από 2 έως 16 νευρώνες στο κρυφό επίπεδο, ενώ το μοντέλο με δύο κρυφά θα έχει από 2 για κάθε κρυφό και θα φτάσει 16 στο πρώτο και 8 νευρώνες στο δεύτερο κρυφό επίπεδο. Κάθε μοντέλο εκπαιδεύεται με τα μισά δεδομένα επί συνόλου 150 λουλουδιών. Στο κάθε είδος αντιστοιχούν 50 περιπτώσεις από τις 150 και αφού εκπαιδευτεί το μοντέλο με τα 25 στοιχεία από κάθε είδος στη συνέχεια καλείται να ταξινομήσει σωστά τα υπόλοιπα μισά, τα οποία είναι ουσιαστικά άγνωστα για το μοντέλο.

Για να είναι τα αποτελέσματα όσο το δυνατόν πιο αξιόπιστα κάθε δοκιμή επαναλαμβάνεται 10 φορές. Για την αξιολόγηση των μοντέλων χρησιμοποιούνται δύο κριτήρια, α) το μέσο τετραγωνικό σφάλμα (MSE) κατά την εκπαίδευση και β) και μία συνάρτηση παραπλήσιας λειτουργίας (errornet) η οποία συγκρίνει τα αποτελέσματα σε άγνωστα δεδομένα. Το σφάλμα μαζί με τον ορισμό μέγιστου αριθμού εποχών κατά την εκπαίδευση μειώνουν την πιθανότητα επιλογής μοντέλου με υπερεκπαίδευση (overtraining). Στη συνέχεια για να είμαστε σίγουροι ότι δεν θα έχουμε προβληματική εκπαίδευση λόγω της κατανομής των δεδομένων, τροφοδοτούμε το νευρωνικό δίκτυο με δεδομένα τα οποία έχουν τυχαία κατανομή κάθε

³⁰ Fisher (1936) examples in discriminant analysis and cluster analysis.

φορά(RandomValues). Τα αποτελέσματα παρατίθενται στον Πίνακα 5.3 και αφορούν ποσοστά επιτυχίας στην εκπαίδευση (βάσει της "errornet"), ενώ αυτά του Πίνακα 5.4 αφορούν τους αντίστοιχους χρόνους εκπαίδευσης. Τα αντίστοιχα αποτελέσματα για τα μοντέλα με 2 κρυφά επίπεδα φαίνονται στον Πίνακα 5.5 και Πίνακα 5.5.

Πίνακας 5.3 Αποτελέσματα δοκιμών μοντέλων με ένα κρυφό επίπεδο (επιτυχία εκπαίδευσης %)

α/α μοντέλου T.N.Δ.	Νευρώνες εισόδου	Νευρώνες κρυφού	Νευρώνες εξόδου	α/α Δοκιμής									
				1	2	3	4	5	6	7	8	9	10
1	4	2	1	75,3	96,81	42,67	70,89	88,37	42,67	87,93	80,44	81,09	96,71
2	4	3	1	42,67	96,84	95,15	88,1	96,66	93,19	42,67	82,03	91,93	90,61
3	4	4	1	91,79	93,01	42,67	85,36	96,53	82,58	95,18	95,51	96,08	42,67
4	4	5	1	96,26	89,42	95,57	93	86,99	94,79	42,67	74,36	83,05	71,44
5	4	6	1	95,33	83,61	71,59	96,85	96,75	95,38	42,67	94,07	94,31	71,73
6	4	7	1	94,65	95,19	96,48	95,46	96,62	97,07	71,7	95,91	94,48	88,77
7	4	8	1	93,66	95,59	95,23	95,36	96,02	93,36	95,13	96,06	42,67	96,34
8	4	9	1	74,19	94,41	95,18	96,64	95,34	73,99	95,16	90,39	95,16	96,67
9	4	10	1	71,78	96,01	94,32	96,04	93,99	95,44	70,05	73,18	95,35	42,67
10	4	11	1	95,64	94,12	92,7	94,14	95,23	94,39	96,01	95,71	94,84	96,95
11	4	12	1	96,66	72,62	95,63	96,48	92,87	95,69	94,01	95,44	95,51	95,15
12	4	13	1	96,01	70,21	93,45	91,32	94,17	94,2	96,6	96,77	94,24	94,09
13	4	14	1	95,13	96,92	93,71	94,38	92,72	95,03	72,94	93,09	96,53	96,83
14	4	15	1	95,37	95,08	92,34	94,55	94,66	88,5	95,09	95,78	94,35	94,91
15	4	16	1	94,6	42,67	95,36	95,14	95,35	94,97	93,98	95,26	96,47	96,01

Πίνακας 5.4 Αποτελέσματα δοκιμών μοντέλων με ένα κρυφό επίπεδο (χρόνος εκπαίδευσης σε sec)

α/α μοντέλου T.N.Δ.	Νευρώνες εισόδου	Νευρώνες κρυφού	Νευρώνες εξόδου	α/α Δοκιμής									
				1	2	3	4	5	6	7	8	9	10
1	4	2	1	6,844	7,015	7,031	6,953	7,328	6,953	7,031	6,781	7,094	6,797
2	4	3	1	7,016	7	6,843	6,906	6,922	6,844	7,062	7,047	6,844	7,047
3	4	4	1	7,094	6,953	6,922	6,985	7	6,922	6,922	6,922	6,953	6,969
4	4	5	1	7,016	7,157	7,203	6,032	6,859	6,969	6,985	7,109	6,843	7
5	4	6	1	7,031	7,187	7,218	7	7,203	7,187	7,235	7,032	7,125	7,172
6	4	7	1	7,625	7,235	7,219	7,531	7,281	7,204	7,235	7,344	7,156	7,265
7	4	8	1	7,266	7,422	7,172	7,266	7,187	7,266	7,203	7,297	7,11	7,109
8	4	9	1	7,031	7,25	7,422	7,313	7,266	7,313	7,39	7,109	7,359	7,406
9	4	10	1	7,453	7,437	7,328	7,735	7,422	7,485	7,297	7,453	7,453	7,313
10	4	11	1	7,438	7,437	7,391	7,469	7,437	7,484	7,547	7,484	7,484	7,5
11	4	12	1	7,593	7,469	7,437	7,547	7,485	7,438	7,515	7,484	7,625	7,579
12	4	13	1	7,64	7,562	7,297	7,672	7,547	7,687	7,5	7,656	7,672	7,5
13	4	14	1	7,671	7,797	7,703	7,609	7,672	7,781	7,641	7,687	7,609	7,734
14	4	15	1	7,89	7,703	7,781	7,703	7,75	7,688	7,671	7,734	7,797	7,828
15	4	16	1	8,063	7,922	7,891	7,828	7,86	7,86	7,968	7,781	7,656	8,172

Πίνακας 5.5 Αποτελέσματα δοκιμών μοντέλων με 2 κρυφά επίπεδα (επιτυχία %)

περίπτωση	Νευρώνες εισόδου	Νευρώνες 1ου κρυφού	Νευρώνες 2ου κρυφού	Νευρώνες εξόδου	α/α Δοκιμής									
					1	2	3	4	5	6	7	8	9	10
1	4	2	2	1	83,86	42,67	83,37	89,21	87,78	76,51	96,46	83,58	89,53	76,51
2	4	3	2	1	75,42	72,26	95,61	85,68	94,24	42,67	94,6	95,52	78,01	68,02
3	4	4	2	1	83,37	73,65	68,33	42,67	81,3	85,46	87,8	95,12	79,17	96,4
4	4	5	2	1	94,85	94,32	89,57	86,85	91,69	94,07	76,15	94,88	70,18	92,92
5	4	6	2	1	95,75	94,3	72,84	73,93	89,89	92,68	88,6	85,92	76,41	71,35
6	4	7	2	1	96,04	82,99	91,48	90,45	95,1	74,96	95,01	80,98	93,52	94,79
7	4	8	2	1	84,16	91,87	94,98	67,16	94,99	96,02	91,67	91,98	94,14	95,17
8	4	9	2	1	69,29	95,34	95,43	95,43	96,43	94,89	94,84	92,05	96	89,76
9	4	10	2	1	93,79	93,97	93,57	95,15	95,84	84,36	94,9	94,35	88,13	88,55
10	4	11	2	1	95,62	93,65	87,01	96,56	78,76	92,83	95,98	95,03	95,93	93,22
11	4	12	2	1	94,58	94,64	94,05	95,26	95,99	93,62	96,1	92,7	96,29	94,93
12	4	13	2	1	93,79	94,44	95,03	78,67	92,68	94,71	92,94	94,54	90,76	95,99
13	4	14	2	1	95,66	94,72	95,85	96,75	93,23	95,6	94,23	93,08	92,08	96,22
14	4	15	2	1	92,34	95,44	89,91	92,52	95,86	94,64	94,12	94,76	95,28	94,78
15	4	16	2	1	90,59	94,2	93,36	93,61	94,87	95,79	94,17	95,27	92,78	93,94
16	4	2	3	1	42,67	79,65	70,83	72,96	42,67	89,12	85,44	42,67	42,67	74,78
17	4	3	3	1	73,22	42,67	87,82	87,22	42,67	91,54	42,67	77,17	86,16	85,3
18	4	4	3	1	94,84	90,91	42,67	42,67	77,33	87,13	42,67	42,67	85,27	89,47
19	4	5	3	1	87,3	94,69	42,67	85,55	94,36	68,96	94,32	96,39	86,19	42,67
20	4	6	3	1	63,09	90,55	81,47	86,08	88,03	42,67	91,3	96,28	88,04	95,7
21	4	7	3	1	94,73	96,28	86,11	83,92	42,67	89,82	96,41	81,63	76,61	93,28
22	4	8	3	1	94,47	93,97	92,51	92,97	94,97	92,94	84,52	96,49	93,44	92,58
23	4	9	3	1	94,21	94,86	92,56	95,4	95,43	95,65	96,35	89,21	93,3	86,91
24	4	10	3	1	94,48	42,67	93,29	96,88	96,18	95,92	96,13	77,6	95,13	95,8
25	4	11	3	1	88,77	95,22	96,37	90,81	91,33	95,47	93,88	94,77	76,06	84,42
26	4	12	3	1	94,57	96,67	94,64	94,35	94,88	94,67	95,69	95,62	95,87	96,67
27	4	13	3	1	96,23	96,35	96,2	95,04	66,01	96,09	93,37	92,6	95,04	92,73
28	4	14	3	1	87,21	94,76	85,74	92,05	96,13	92,44	94,93	80,86	95,16	42,67
29	4	15	3	1	94,18	94,68	94,84	91,17	94,34	96,01	95,17	92,62	93,31	93,39
30	4	16	3	1	95,16	92,74	95,34	96,41	95,52	95,44	93,02	91,35	68,55	90,28
31	4	2	4	1	42,67	42,67	78,12	77,36	82,29	42,67	42,67	83,24	42,67	42,67
32	4	3	4	1	63,68	42,67	81,15	42,67	42,67	86,35	42,67	42,67	92,6	84,8
33	4	4	4	1	93,6	42,67	86,03	42,67	89,38	80,88	94,87	86,11	78,3	42,67
34	4	5	4	1	42,67	42,67	94,17	42,67	78,75	42,67	76,51	42,67	87,32	42,67
35	4	6	4	1	42,67	95,52	95,35	42,67	92,59	94,84	42,67	95,23	42,67	96,12
36	4	7	4	1	83,46	94,45	75,09	72,2	75,93	95,36	70,66	77,27	94,98	90,04
37	4	8	4	1	88,36	85,95	42,67	93,14	95,51	93,84	42,67	95,12	91,25	67,21
38	4	9	4	1	42,67	75,94	71,24	92,01	94,22	94,56	95,24	74,7	96,47	95,91
39	4	10	4	1	95,04	94,94	94,58	82,67	96,46	93,7	92,76	88,49	83,29	95,25
40	4	11	4	1	72,7	92,66	92,48	95,11	96,13	73,15	96,71	96,54	93,35	92,48
41	4	12	4	1	80,4	96,59	94,06	83,73	95,55	96,05	83,84	96,22	96,6	95,34
42	4	13	4	1	91,42	90,97	95,28	95,85	95,07	81,21	94,94	94,9	70,77	95,96
43	4	14	4	1	92,38	90,94	94,92	74,74	95,07	92,8	92,53	94,89	86,85	96,02
44	4	15	4	1	93,24	95,6	95,5	94,95	96,5	92,61	96,9	95,41	95,08	96,26
45	4	16	4	1	95,51	94,21	95,2	64,21	94,74	95,57	93,33	93,92	95,42	86,86
46	4	2	5	1	42,67	42,67	42,67	42,67	42,67	42,67	42,67	42,67	42,67	42,67
47	4	3	5	1	94	89,67	69,68	95,11	93,48	42,67	86,49	42,67	42,67	87,15
48	4	4	5	1	69,47	82,96	42,67	42,67	42,67	81,73	42,67	42,67	87,42	96,64
49	4	5	5	1	42,67	71,73	94,99	92,03	42,67	80,41	96,15	42,67	92,36	95,7
50	4	6	5	1	76,28	91,33	80,49	79,77	76,44	96,03	90,83	87,41	85,15	92,07
51	4	7	5	1	42,67	96,33	96,32	71,09	94,31	96,21	42,67	92,15	91,07	70,41
52	4	8	5	1	95,46	95,21	93,55	72,21	87,55	95,26	85,82	93,82	95,36	93,77
53	4	9	5	1	69,98	67,5	42,67	70,13	92,53	87,7	94,04	74,28	96,14	95,42
54	4	10	5	1	95,19	92,69	42,67	95,69	94,54	95,76	94,27	95,03	93	42,67

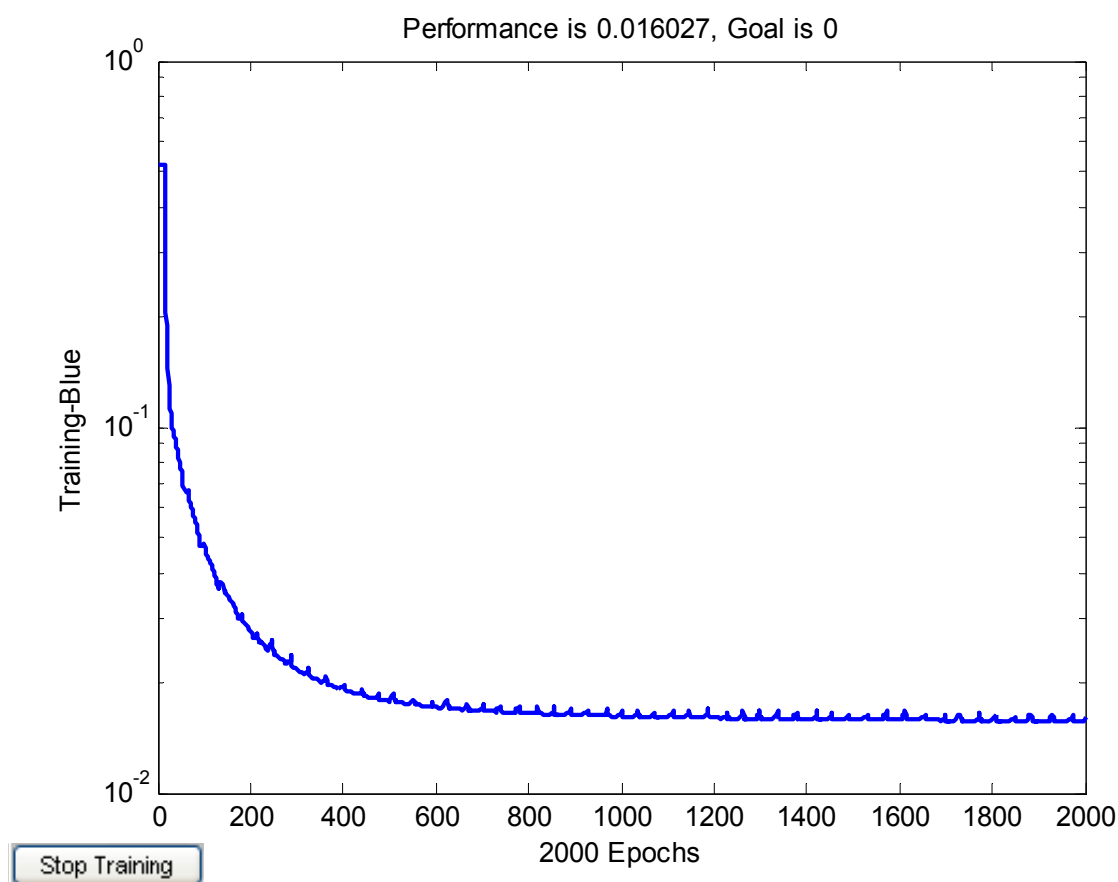
55	4	11	5	1	95,44	42,67	94,66	92,7	92,78	95,34	95,66	68,25	94,22	92,82
56	4	12	5	1	94,98	95,89	94,41	68,73	95,44	93,63	96,92	95,06	95,5	95,34
57	4	13	5	1	92,92	95,46	93,12	95,73	93,24	94,44	81,94	92,83	94,45	94,01
58	4	14	5	1	90,26	92,4	93,05	88,39	94,55	94,78	95,52	94,03	92,6	95,7
59	4	15	5	1	94,88	94,93	91,09	94,91	95,13	93,64	92,61	93,98	92	93,35
60	4	16	5	1	95,24	92,74	65,82	95,52	95,59	95,26	94,58	94,01	93,05	93,54
61	4	2	6	1	79,18	73,69	42,67	42,67	79,46	71,47	72,73	42,67	42,67	42,67
62	4	3	6	1	87,11	42,67	87,57	42,67	77,08	42,67	83,98	71,61	82,06	82,89
63	4	4	6	1	92,54	94,3	68,3	93,77	82,25	42,67	83,34	72,08	42,67	74,1
64	4	5	6	1	94,84	42,67	91,93	85,7	42,67	76,11	79,33	42,67	42,67	93,3
65	4	6	6	1	90,9	42,67	93,66	94,8	91,54	95,57	95,42	96,35	80,56	96,08
66	4	7	6	1	90,97	74,62	79,12	95,45	69,8	90,04	42,67	94,94	42,67	79,18
67	4	8	6	1	42,67	92,16	88,89	69,91	95,11	96,46	95,74	42,67	42,67	95,12
68	4	9	6	1	96,6	74,34	76,89	78,75	92,9	95,94	96,05	42,67	95,13	92,7
69	4	10	6	1	93,79	95,55	77,28	96,57	95,55	94,21	95,15	95,86	92,52	96,37
70	4	11	6	1	72,35	92,8	91,37	94,26	95,14	95,16	95,93	42,67	95,17	94,91
71	4	12	6	1	90,49	94,95	89,85	70,93	81,76	94,56	95,44	85,01	94,06	92,94
72	4	13	6	1	76,6	95,52	95,99	95,62	93,03	95,15	95,34	95,29	95,45	95,23
73	4	14	6	1	94,12	94,91	94,08	95,11	95,31	95,34	96,24	94,21	92,47	93,29
74	4	15	6	1	96,5	95,43	78,92	94,89	93,13	94,65	94,32	94,73	93,57	93,48
75	4	16	6	1	93,19	95,14	95,46	95,91	95,23	87,41	95,56	95,45	91,08	92,83
76	4	2	7	1	42,67	85,84	91,64	66,94	42,67	42,67	67,32	42,67	42,67	42,67
77	4	3	7	1	90,92	78,81	42,67	42,67	90,08	42,67	95,5	94,9	89,93	42,67
78	4	4	7	1	84,6	85,81	74,71	91,52	89,92	87,04	91,96	42,67	42,67	85,39
79	4	5	7	1	42,67	81,53	95,92	74,65	89,96	42,67	96,32	42,67	79,03	88,25
80	4	6	7	1	42,67	95,42	42,67	95,47	92,53	42,67	83,69	94,77	94,51	74,64
81	4	7	7	1	88,24	42,67	71,94	91,95	94,33	94,41	93,21	93,62	42,67	93,55
82	4	8	7	1	90,44	84,8	95,05	91,01	93,6	77,28	42,67	95,04	95,21	90,08
83	4	9	7	1	42,67	72,94	96,35	96,3	95,05	94,67	77,13	94,16	95,51	96,61
84	4	10	7	1	94,71	69,52	94,8	96,29	94,23	96,7	95,27	83,86	95,23	93,79
85	4	11	7	1	88,14	92,22	95,16	95,88	88,35	96,5	92,14	79,87	93,68	67,22
86	4	12	7	1	94,9	95,62	94,39	94,64	94,09	93,34	83,92	93,32	94,88	90,84
87	4	13	7	1	93,14	84,87	92,03	95,76	92,51	96,33	95,19	91,42	92,59	95,86
88	4	14	7	1	70,27	91,09	94,71	92,66	94,21	76,85	94,87	94,84	95,04	83,08
89	4	15	7	1	95,72	92,79	94,98	95,54	96,39	95,9	75,45	95,44	94,21	95,16
90	4	16	7	1	95,17	95,26	95,15	95,26	91,93	95,07	92,6	96,54	95,58	92,53
91	4	2	8	1	78,52	70,39	42,67	42,67	42,67	56,53	66,15	83,25	42,67	77,18
92	4	3	8	1	88,59	42,67	42,67	42,67	83,24	42,67	42,67	92,89	72,72	78,29
93	4	4	8	1	42,67	42,67	86,02	42,67	93,22	71,12	95,07	73,48	88,53	42,67
94	4	5	8	1	88,04	95,85	93,11	94,93	42,67	42,67	69,89	42,67	42,67	42,67
95	4	6	8	1	73,9	42,67	42,67	92,74	89,65	42,67	90,45	96,21	95,3	74,93
96	4	7	8	1	93,27	96,14	95,55	42,67	91,66	84,58	79,23	81,54	94,11	94,93
97	4	8	8	1	79,13	42,67	82,76	70,4	96,29	95,22	79,37	92,45	96,94	96,43
98	4	9	8	1	42,67	42,67	94,59	94,47	75,22	94,77	76,95	92,79	94,84	92,17
99	4	10	8	1	94,38	79,9	76,77	87,77	93,82	94,65	90,79	95,41	93,83	91,69
100	4	11	8	1	95,34	92,95	93,61	86,65	94,31	94,27	93,48	92,99	93,08	96,7
101	4	12	8	1	93,95	92,59	95,32	95,4	96,3	95	95,02	95,08	71,08	95,17
102	4	13	8	1	94,78	78,79	95,6	42,67	92,93	92,64	95,25	92,41	94,17	93,44
103	4	14	8	1	95,12	94,39	93,77	93,42	92,55	95,11	92,73	95,23	96,82	95,63
104	4	15	8	1	93,96	93,1	94,41	95,52	95,58	92,96	94,17	96,13	94,53	94,35
105	4	16	8	1	95,79	94,92	94,47	96,4	93,43	94,23	92,33	94,78	93,02	93,13

Πίνακας 5.6 Αποτελέσματα δοκιμών μοντέλων με 2 κρυφά επίπεδα (χρόνος εκπαίδευσης σε sec)

περίπτωση	Νευρώνες εισόδου	Νευρώνες 1ου κρυφού	Νευρώνες 2ου κρυφού	Νευρώνες εξόδου	α/α Δοκιμής									
					1	2	3	4	5	6	7	8	9	10
1	4	2	2	1	7,844	7,704	8,016	7,641	7,687	7,922	8	7,703	8,078	7,89
2	4	3	2	1	7,781	7,796	7,813	7,719	8,125	7,953	7,844	7,766	7,781	7,969
3	4	4	2	1	8,172	8	7,719	7,86	7,86	7,938	8,094	7,89	8,063	7,89
4	4	5	2	1	7,937	8,312	8,047	7,937	8,172	8,125	8,015	8,031	8,156	8
5	4	6	2	1	8,047	8	8,187	8,094	8,031	8,031	8,062	8,25	8,172	8,047
6	4	7	2	1	8,64	8,11	7,907	8,172	8,109	8,109	8,234	8,375	8,156	8,203
7	4	8	2	1	8,25	8,281	8,39	8,25	8,547	8,188	8,141	8,562	8,187	8,156
8	4	9	2	1	8,235	8,516	8,218	8,203	8,344	8,328	8,282	8,687	8,281	8,187
9	4	10	2	1	8,25	8,297	8,391	8,328	8,407	8,344	8,328	8,297	8,296	8,406
10	4	11	2	1	8,484	8,469	8,422	8,359	8,422	8,485	8,438	8,469	8,484	8,5
11	4	12	2	1	8,438	8,516	8,515	8,578	8,281	8,562	8,39	8,485	8,5	8,485
12	4	13	2	1	8,516	8,515	8,484	8,469	8,407	8,578	8,469	8,484	8,531	8,469
13	4	14	2	1	8,672	8,719	8,703	8,734	8,703	8,657	8,671	8,75	8,656	8,64
14	4	15	2	1	8,672	8,828	8,625	8,875	8,859	8,765	8,547	8,813	8,844	8,859
15	4	16	2	1	8,812	8,781	8,875	8,859	8,937	8,984	8,766	8,734	8,859	8,781
16	4	2	3	1	7,828	7,75	7,516	8	8,031	8,14	8,047	7,985	7,907	7,718
17	4	3	3	1	7,781	7,984	8,078	8,063	7,969	7,984	8,125	7,984	7,781	7,906
18	4	4	3	1	8,156	8,062	8	7,891	7,844	7,937	7,969	7,985	7,938	8,016
19	4	5	3	1	7,953	7,969	8,016	8,078	8,063	8,032	8,11	8,032	8,063	8,094
20	4	6	3	1	8,063	8,235	8,204	8,063	8,093	8,188	8,015	8,047	8,125	8,156
21	4	7	3	1	8,125	8,172	8,344	8,344	8,234	8,172	8,469	8,437	8,203	8,422
22	4	8	3	1	8,219	8,422	8,234	8,406	8,25	8,219	8,484	8,25	8,313	8,391
23	4	9	3	1	8,344	8,407	8,36	8,36	8,422	8,297	8,344	8,39	8,359	8,437
24	4	10	3	1	8,984	8,266	8,547	8,437	8,516	8,484	8,375	8,422	8,437	8,86
25	4	11	3	1	8,5	8,547	8,609	8,484	8,75	8,515	8,5	8,546	8,641	8,453
26	4	12	3	1	8,563	8,578	8,625	8,594	8,531	8,453	8,688	8,579	8,672	8,734
27	4	13	3	1	8,531	8,687	8,703	8,641	8,687	8,703	8,671	8,719	8,641	8,703
28	4	14	3	1	8,782	8,735	8,782	8,766	8,891	8,734	8,89	8,75	8,765	9,015
29	4	15	3	1	8,797	8,812	8,812	8,875	8,75	8,969	8,86	8,844	8,812	8,687
30	4	16	3	1	8,938	9,218	8,906	8,937	8,828	8,859	8,907	9,016	8,843	9
31	4	2	4	1	7,921	7,953	7,844	7,906	8,11	8,094	7,891	8,188	8,062	8,015
32	4	3	4	1	8,063	7,984	7,968	8,078	8,187	7,937	8	8,14	7,969	8
33	4	4	4	1	8	7,86	8,079	8,031	8,203	8,219	8,172	8,078	8,109	7,954
34	4	5	4	1	8,25	8,281	8,031	8,187	8,297	8,015	8,406	8,125	8,312	8,156
35	4	6	4	1	8,172	8,11	8,078	8,281	8,187	8,172	8,14	8,422	8,391	8,141
36	4	7	4	1	8,266	8,25	8,375	8,235	8,313	8,344	8,61	8,328	8,656	8,203
37	4	8	4	1	8,359	8,546	8,219	8,328	7,984	8,328	8,328	8,469	8,547	8,375
38	4	9	4	1	8,546	8,422	8,437	8,453	8,454	8,375	8,453	8,344	8,469	8,39
39	4	10	4	1	8,469	8,453	8,468	8,438	8,563	8,5	8,469	8,5	8,422	8,469
40	4	11	4	1	8,531	8,75	8,61	8,515	8,406	8,609	8,64	8,656	8,562	8,531
41	4	12	4	1	8,562	8,547	8,641	8,719	8,703	8,703	8,516	8,609	8,75	8,688
42	4	13	4	1	8,563	8,735	8,719	8,688	8,641	8,719	8,688	8,438	8,578	8,703
43	4	14	4	1	8,906	9,015	8,859	8,875	8,781	8,875	8,907	8,844	8,906	8,75
44	4	15	4	1	8,922	8,812	8,797	8,937	9,031	8,922	8,813	8,875	8,86	8,813
45	4	16	4	1	8,922	9,031	8,875	8,938	8,969	8,797	9,156	9,109	8,969	8,891
46	4	2	5	1	8,266	8,047	8,031	8,281	8,328	8,172	7,906	8,109	8,25	8
47	4	3	5	1	8,032	8,016	8,266	8,203	8,219	8,094	8,14	8,094	8,063	8,453
48	4	4	5	1	8,094	8,125	8,25	7,875	8	8,14	8,281	8,125	8,046	8,063
49	4	5	5	1	8,453	8,172	8,235	8,297	8,125	8,313	8,094	8,218	8,157	8,157
50	4	6	5	1	8,75	8,344	8,453	8,203	8,343	8,266	8,282	8,266	8,329	8,328
51	4	7	5	1	8,907	8,5	8,375	8,375	8,578	8,375	8,375	8,375	8,359	8,328
52	4	8	5	1	8,454	8,531	8,422	8,468	8,328	8,375	8,578	8,39	8,484	8,531
53	4	9	5	1	8,797	8,563	8,75	8,422	8,531	8,422	8,578	8,188	8,594	8,61
54	4	10	5	1	8,625	8,64	8,828	8,515	8,531	8,688	8,656	8,5	8,593	8,641

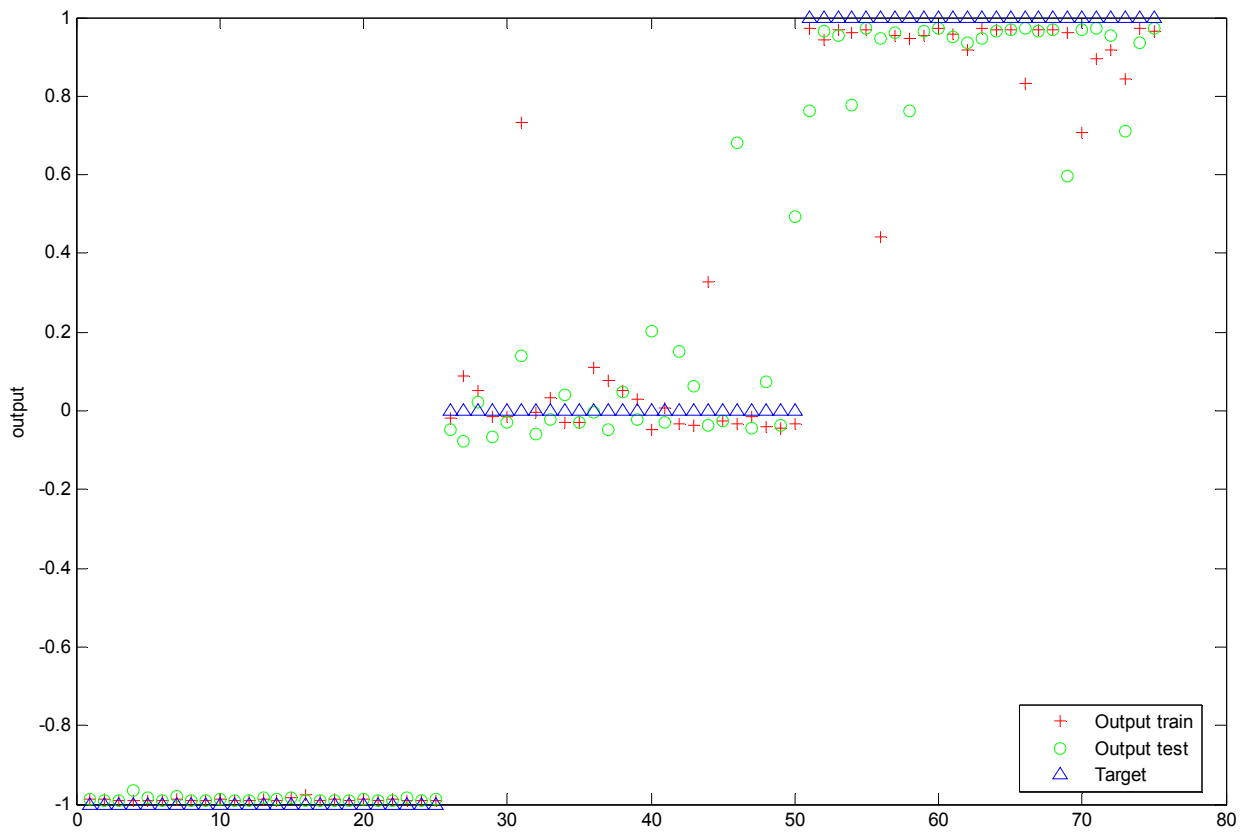
55	4	11	5	1	8,719	8,656	8,687	8,844	8,672	8,656	9,062	8,64	8,578	8,828
56	4	12	5	1	8,812	8,765	8,875	8,921	8,953	8,828	8,75	8,781	8,75	8,532
57	4	13	5	1	8,766	8,766	8,859	8,875	8,75	8,937	9	8,922	8,86	8,86
58	4	14	5	1	9	9	8,985	9,063	8,906	8,86	9,079	9,063	8,985	9,094
59	4	15	5	1	9,109	9,094	9,078	9,094	9,031	9,093	9,016	9,032	9,016	8,75
60	4	16	5	1	9,156	9,062	9,047	9,203	9,109	9,047	9,047	9,047	9,047	9,328
61	4	2	6	1	8,094	8,375	8,063	8,203	8,125	8,109	8,203	8,078	7,954	8,11
62	4	3	6	1	8,047	8,187	8,234	8,187	8,312	8,172	8,187	8,187	8,344	8,141
63	4	4	6	1	8,172	8,235	8,266	8,312	8,187	8,25	8,36	8,422	8,25	8,14
64	4	5	6	1	8,203	8,343	8,25	8,421	8,297	8,578	8,187	8,328	8,281	8,359
65	4	6	6	1	8,297	8,266	8,375	8,5	8,25	8,25	8,469	8,468	8,344	8,313
66	4	7	6	1	8,453	8,516	8,437	8,656	8,765	8,656	8,484	8,343	8,797	8,531
67	4	8	6	1	8,734	8,453	8,75	8,657	9,016	8,531	8,5	8,859	8,937	8,75
68	4	9	6	1	8,891	8,937	8,906	8,734	8,703	8,891	8,969	8,782	8,703	8,734
69	4	10	6	1	9	8,734	8,843	9,359	8,781	9	9,031	8,922	8,922	8,75
70	4	11	6	1	8,829	8,969	8,828	9,078	8,891	8,828	9,203	8,75	8,969	8,672
71	4	12	6	1	8,875	8,891	9,265	8,906	8,906	8,907	9,11	8,875	8,859	8,875
72	4	13	6	1	8,922	9	9,093	9,078	8,859	8,969	8,875	8,969	8,922	9,187
73	4	14	6	1	9,109	9,047	9,062	9,156	9,125	9,094	9,265	9,203	9,062	9,172
74	4	15	6	1	9,079	9,188	9,11	9,188	9,016	9,187	9,062	9,125	8,968	9,188
75	4	16	6	1	9,125	9,156	9,25	9,156	9,188	9,109	9,234	9,375	9,25	9,187
76	4	2	7	1	8,406	8,203	8,562	8,234	8,297	8,531	8,625	8,422	8,453	8,578
77	4	3	7	1	8,375	8,234	8,562	8,625	8,719	8,609	8,312	8,469	8,516	8,421
78	4	4	7	1	8,547	8,609	8,39	8,719	8,375	8,515	8,266	8,703	8,344	8,532
79	4	5	7	1	8,391	8,593	8,5	8,468	8,406	8,469	8,422	8,328	8,813	8,359
80	4	6	7	1	8,609	8,437	8,734	8,718	8,469	8,782	8,563	8,704	8,781	8,672
81	4	7	7	1	8,671	8,813	8,672	8,843	8,688	8,672	8,687	8,687	8,703	8,703
82	4	8	7	1	8,672	8,734	8,812	8,688	8,625	8,657	8,719	8,453	8,734	8,64
83	4	9	7	1	8,859	8,828	8,687	8,797	8,829	8,797	8,922	8,735	8,891	8,766
84	4	10	7	1	8,734	8,922	8,953	8,922	9,344	8,89	9,141	8,922	8,953	8,89
85	4	11	7	1	9,047	9,157	8,969	8,937	8,89	9,047	9	8,984	9,156	9,031
86	4	12	7	1	9,015	9,031	9,297	9	8,891	8,968	9,047	9,14	9,109	9,062
87	4	13	7	1	9,109	9,187	9,094	9,11	8,844	9,078	9,171	9,047	9,187	9,422
88	4	14	7	1	9,25	9,203	9,14	9,312	9,234	9,203	9,234	9,297	9,36	9,156
89	4	15	7	1	9,297	9,329	9,344	9,218	9,25	9,296	9,391	9,172	9,265	9,313
90	4	16	7	1	9,531	9,594	9,657	9,688	9,438	9,515	9,437	9,469	9,5	9,516
91	4	2	8	1	8	8,454	8,36	8,438	8,438	8,375	8,313	8,313	8,344	8,281
92	4	3	8	1	8,468	8,437	8,422	8,531	8,391	8,422	8,75	8,375	8,375	8,563
93	4	4	8	1	8,484	8,313	8,328	8,438	8,422	8,594	8,422	8,532	8,485	8,031
94	4	5	8	1	8,5	8,594	8,609	8,453	8,704	8,531	8,547	8,703	8,609	9,031
95	4	6	8	1	8,594	8,531	8,781	8,688	8,703	8,453	8,578	8,61	8,594	8,734
96	4	7	8	1	8,765	8,703	8,656	8,718	8,813	8,672	8,859	8,718	8,859	8,719
97	4	8	8	1	8,782	8,891	8,875	8,735	8,828	8,844	8,719	8,781	8,765	8,844
98	4	9	8	1	8,828	8,984	8,906	8,954	8,875	8,953	8,937	9,093	8,89	8,953
99	4	10	8	1	9,469	8,891	9,047	8,984	8,922	8,922	9,031	8,938	8,891	9,015
100	4	11	8	1	9,156	9,079	9,125	9,203	9,062	9,078	9,063	9,203	9,141	9,172
101	4	12	8	1	9,078	9,235	9,141	9,25	9,125	9,297	9,125	9,125	9,11	9,282
102	4	13	8	1	9,297	9,328	9,25	9,203	9,25	9,281	9,453	9,25	9,422	9,375
103	4	14	8	1	9,422	9,406	9,469	9,375	9,562	9,578	9,39	9,484	9,25	9,547
104	4	15	8	1	9,437	9,453	9,562	9,75	9,578	9,562	9,531	9,469	9,5	9,641
105	4	16	8	1	9,312	9	9,078	8,938	8,985	8,985	9,031	9,125	8,984	8,969

Από τα παραπάνω δεδομένα διαπιστώνουμε πως από τα μοντέλα με ένα κρυφό επίπεδο, τη μέγιστη επιτυχία (97,07%) είχε το μοντέλο [4x7x1] στην έκτη δοκιμή (περίπτωση 6 – 6) ενώ από τα μοντέλα με δύο κρυφά επίπεδα τη μέγιστη επιτυχία (96,94) είχε το μοντέλο [4x8x8x1] στην ένατη δοκιμή (περίπτωση 97 – 9). Διαπιστώνουμε πως στο συγκεκριμένο πρόβλημα το μοντέλο με ένα κρυφό επίπεδο, αν και με πολύ μικρή διαφορά, είχε μεγαλύτερη επιτυχία. Να τονίσουμε πως σε όλες τις παραπάνω περιπτώσεις η εκπαίδευση ολοκληρώνονταν στις 2000 εποχές για όλα τα μοντέλα. Αξίζει επίσης να σημειωθεί ότι σε δοκιμές που έγιναν για λιγότερες εποχές (300, 500, 1000) το μοντέλο με δύο κρυφά επίπεδα είχε καλύτερα αποτελέσματα από το μοντέλο με ένα κρυφό επίπεδο, αλλά όχι συνολικά τα καλύτερα. Επίσης όταν η εκπαίδευση έγινε σε μεγαλύτερο αριθμό εποχών (2500, inf) τα αποτελέσματα ήταν χειρότερα και για τα δύο μοντέλα, καθώς πλέον είχαμε υπερεκπαίδευση.

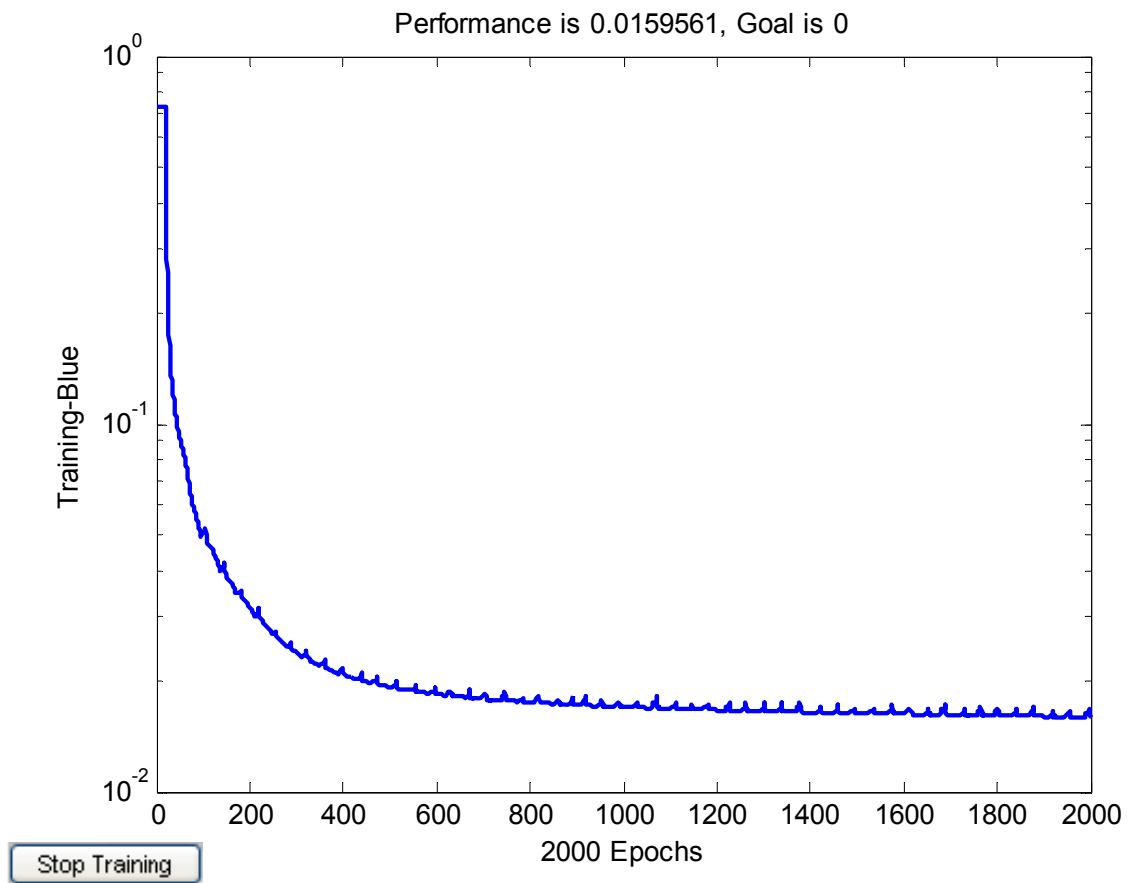


Σχ. 5.8 Γραφική παράσταση (MSE) της εκπαίδευσης μοντέλου με ένα κρυφό επίπεδο (4x7x1)

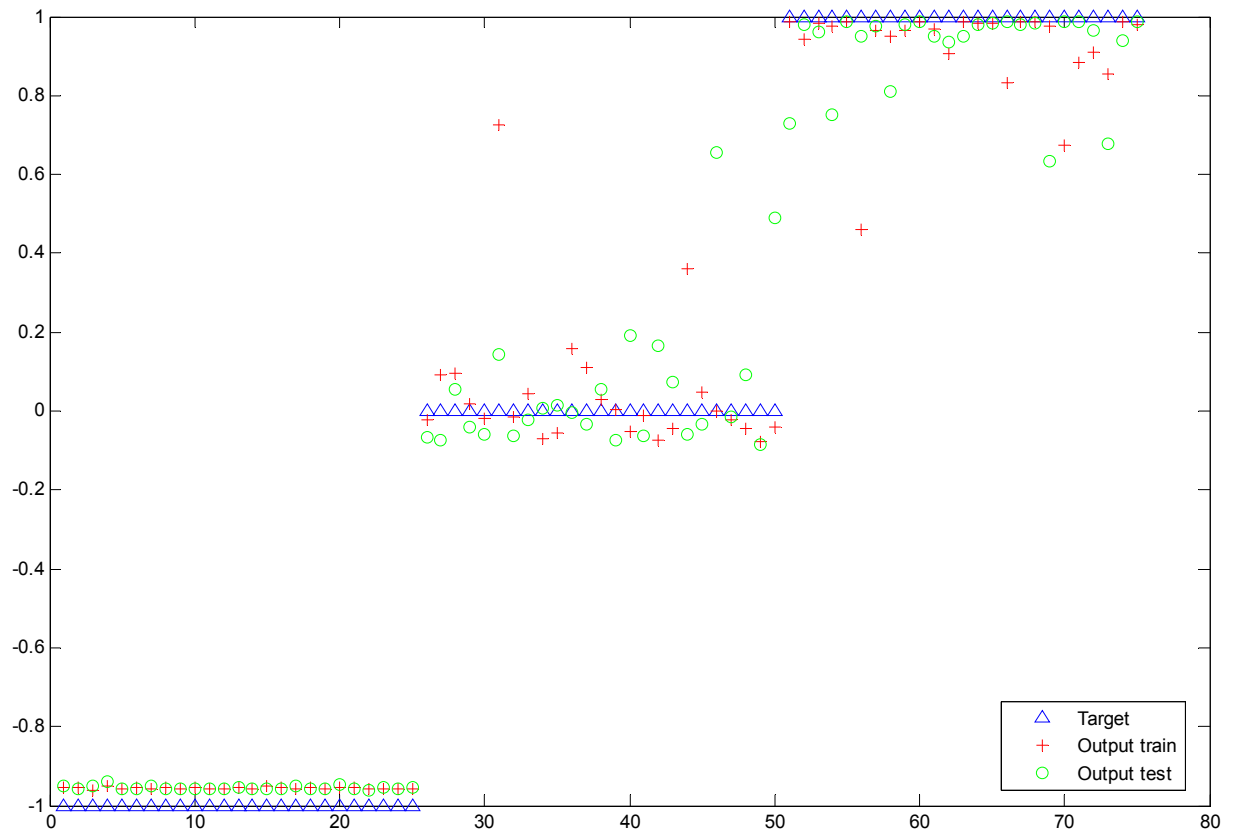
Όπως φαίνεται στο Σχ. 5.8 του νευρωνικού δικτύου, μία «καλή» εκπαίδευση οδηγεί το μοντέλο σε σύγκλιση από τις πρώτες επαναλήψεις εκπαίδευσης και το σφάλμα κυμαίνεται στην τάξη του 10⁻¹ ήδη από τις 250 – 300 εποχές. Αντίθετα παρατηρείται μία πολύ μικρή σύγκλιση έως τις 1800 – 2000 εποχές, όπου από το σημείο αυτό και μετά έχουμε υπερεκπαίδευση. Μετά την ολοκλήρωση κάθε εκπαίδευσης, ακολουθεί η δοκιμή με γνωστά και άγνωστα δεδομένα ώστε να βρεθεί το καλύτερο μοντέλο. Τα αποτελέσματα της δοκιμής αυτής για το μοντέλο (4x7x1) απεικονίζονται στο Σχ. 5.9. Αντίστοιχα, στα Σχ. 5.10 και Σχ. 5.11 απεικονίζονται η μεταβολή του MSE και τα αποτελέσματα δοκιμών για γνωστά και άγνωστα δεδομένα για το (4x8x8x1).



Σχ. 5.9 Έξοδος μοντέλου με ένα κρυφό επίπεδο [4x7x1]



Σχ. 5.10 Γραφική παράσταση (MSE) της εκπαίδευσης μοντέλου με 2 κρυφά επίπεδα (4x8x8x1)



Σχ. 5.11 Έξοδος μοντέλου με 2 κρυφά επίπεδα (4x8x8x1)

Όπως φαίνεται και στα παραπάνω γραφήματα τα μοντέλα έχουν ικανοποιητική επιτυχία σε προβλήματα με άγνωστα δεδομένα. Τα προβλήματα του κλήθηκε να αντιμετωπίσει κάθε μοντέλο είναι σύνθετα, αλλά πολύ απλά σε σχέση με τις δυνατότητες ενός νευρωνικού δικτύου. Το πρόβλημα το οποίο πραγματευόμαστε όμως είναι ο έλεγχος (ηλεκτρικής μηχανής), όπου πλέον η πολυπλοκότητα του συστήματος αυξάνεται κατακόρυφα, έτσι στο επόμενο Κεφάλαιο παρουσιάζουμε την προσπάθεια αυτής της πτυχιακής εργασίας για το συγκεκριμένο πρόβλημα.

6 ΕΦΑΡΜΟΓΗ Τ.Ν.Δ. ΣΕ ΠΡΑΓΜΑΤΙΚΟ ΣΥΣΤΗΜΑ ΕΛΕΓΧΟΥ ΔΙΕΓΕΡΣΗΣ

6.1 Περιγραφή Συστήματος Υπό Μελέτη.

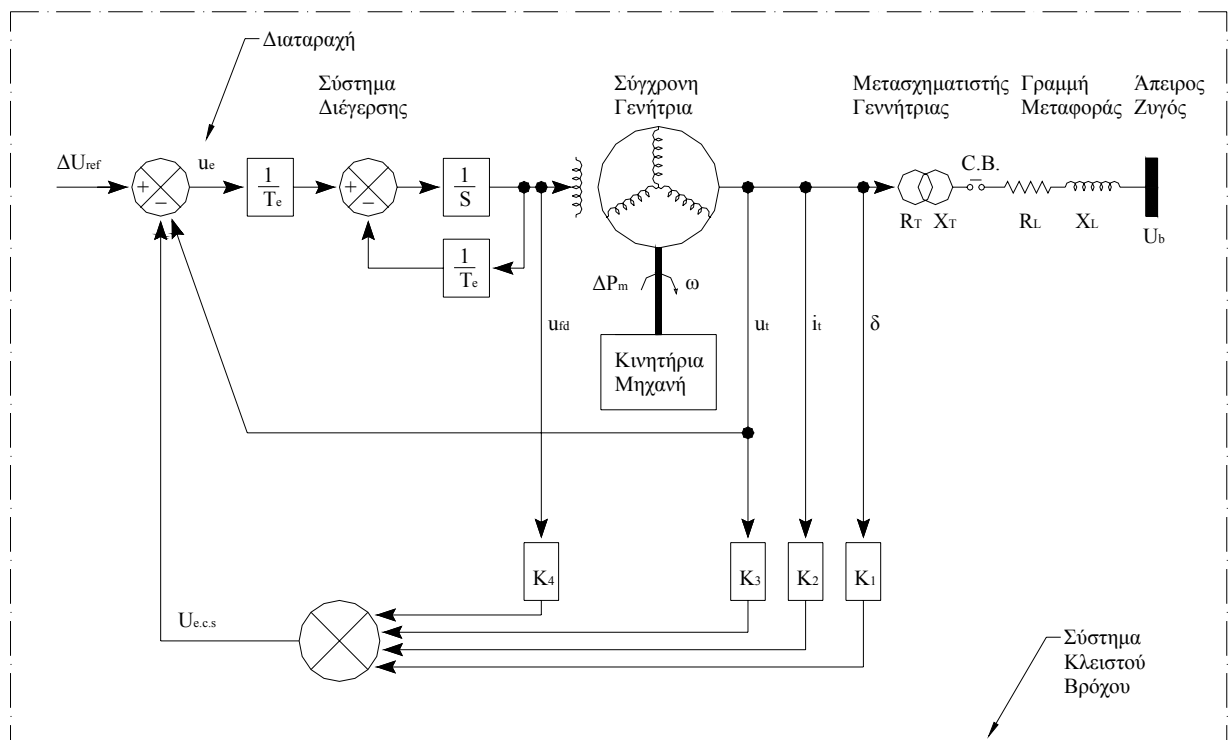
Στο Κεφάλαιο αυτό θα εφαρμοστεί μοντέλο νευρωνικού δικτύου ως ελεγκτής διέγερσης με δεδομένα από πραγματικό σύστημα ελέγχου διέγερσης μιας σύγχρονης γεννήτριας. Το πραγματικό σύστημα, στο Σχ. 6.1, είναι σύστημα ελέγχου κλειστού βρόχου και προσομοιώνεται η λειτουργία του με τη χρήση ηλεκτρονικού υπολογιστή και του κατάλληλου λογισμικού για χρονικό διάστημα πέντε δευτερολέπτων (5sec), όπου έχουμε δειγματοληψία ανά 0,2 χιλιοστά του δευτερολέπτου (0,2ms). Η προσομοίωση γίνεται για δέκα (10) διαφορετικά επίπεδα διαταραχής, όπου αυτή η διαταραχή κυμαίνεται από 0,001pu έως 0,01pu. Από την προσομοίωση προκύπτουν τα επιθυμητά μεγέθη για την διόρθωση του προβλήματος τα οποία θα χρησιμοποιήσουμε για να εκπαιδεύσουμε το νευρωνικό δίκτυο. Με δεδομένο ότι τα αποτελέσματα προκύπτουν από προσομοίωση ενός συμβατικού υπο-βέλτιστου ελεγκτή, κανείς αναμένει ότι το Τ.Ν.Δ. θα έχει επίσης την ίδια υπο-βέλτιστη στην καλύτερη περίπτωση. Σε αυτό το σημείο το νευρωνικό δίκτυο μπορεί να αποδειχθεί πιθανώς καλύτερο, καθώς εξαιτίας της φύσης του και της εκπαίδευσής, είναι δυνατόν να πετύχει καλύτερα αποτελέσματα από τα δεδομένα εκπαίδευσής, αρκεί βέβαια το μέγεθος του μοντέλου, η αρχιτεκτονική και η εκπαίδευση να το επιτρέπουν, δηλ. να έχει τους απαραίτητους νευρώνες και τα επίπεδα, τους κατάλληλους γραμμικούς ή μη μετασχηματισμούς και αντιπροσωπευτικό δείγμα εκπαιδευτικών διανυσμάτων. Οι παράμετροι καθώς και το σημείο λειτουργία της σύγχρονης μηχανής παρατίθενται στον Πίνακα 6.1.

Πίνακας 6.1 Στοιχεία μοντέλου προσομοίωσης

Σύγχρονη Μηχανή 87kW, 415V 4 – πολική	X_d	1.758pu			
	X_q	0.990pu	R_a	0.01450pu	
	X_{fd}	1.761pu	R_{kd}	0.00422pu	
	X_{kd}	1.664pu	R_{kq}	0.01260pu	
	X_{kq}	0.955pu	R_{fd}	0.0268pu	
	X_{md}	1.658pu	H	1.434s	
	X_{mq}	0.899pu			
Συμβατικός διεγέρτης	K_e	1	T_e	0.472s	
Μετασχηματιστής	R_T	0.03630pu	X_T	0.0838pu	
Γραμμή μεταφοράς	R_L	0.03744pu	X_L	0.3903pu	
Σημείο Λειτουργίας της Γεννήτριας					
P_t	Q_t	U_t	U_{fd}	δ	U_b
0.8pu	0.84pu	1.29pu	0.0043pu	0.56rad	0.97pu

6.2 Διαδικασία εκπαίδευσης Τ.Ν.Δ.

Τα δεδομένα που προέκυψαν από την προσομοίωση θα χρησιμοποιηθούν για την εκπαίδευση και τη δοκιμή των νευρωνικών δικτύων, όπου η έξοδος τους θα είναι η επιθυμητή τάση διέγερσης. Ο χρόνος στον οποίο πρέπει το σύστημα να επανέλθει στην μόνιμη κατάσταση λειτουργίας είναι πέντε δευτερόλεπτα. Η δειγματοληψία όπως αναφέρθηκε έγινε ανά 0,2msec (ή 5kHz), έτσι έχουμε 25.000 τιμές για κάθε μέγεθος στη διάρκεια των πέντε δευτερολέπτων. Συνολικά προκύπτουν 250.000 στοιχεία για όλες τις περιπτώσεις, γιατί η προσομοίωση γίνεται για 10 διαφορετικές περιπτώσεις διαταραχής, οπότε τα δεδομένα μας αποτελούν έναν πίνακα 6 στηλών και 250.000 γραμμών. Χωρίς βλάβη της γενικότητας, τα πρότυπα εκπαίδευσης (training patterns) που χρησιμοποιήθηκαν ελήφθησαν ανά 20msec, συνεπώς προέκυψε ένας πίνακας 6 στηλών και 2500 γραμμών. Για τη σωστή εκπαίδευση του μοντέλου δοκιμάζουμε την ικανότητα του ξεκινώντας με γνωστά όλα τα δεδομένα και φτάνοντας να είναι γνωστά μόνο τα μισά (5/10). Έχοντας αναπτύξει διάφορα μοντέλα νευρωνικών δικτύων καλούμαστε τώρα να χρησιμοποιήσουμε ένα μοντέλο νευρωνικού δικτύου για να αντικαταστήσουμε τον «κλασικό» ελεγκτή του Σχ. 6.1 με έναν μικροελεγκτή ο οποίος είναι προγραμματισμένος με ένα μοντέλο νευρωνικού δικτύου. Για να καταλήξουμε στο κατάλληλο μοντέλο θα ακολουθήσουμε για μία ακόμη φορά μία σειρά δοκιμών και θα επιλέξουμε αυτό που θα ικανοποιεί περισσότερο τα κριτήρια που έχουμε θέσει. Η δοκιμές θα ξεκινήσουν με το γνωστό πλέον μοντέλο "MLP" από το προηγούμενο πείραμα.



Σχ. 6.1 Συμβατικό σύστημα ελέγχου διέγερσης σύγχρονης γεννήτριας κλειστού βρόχου

6.2.1 Κανονικοποίηση Δεδομένων Εισόδου – Εξόδου.

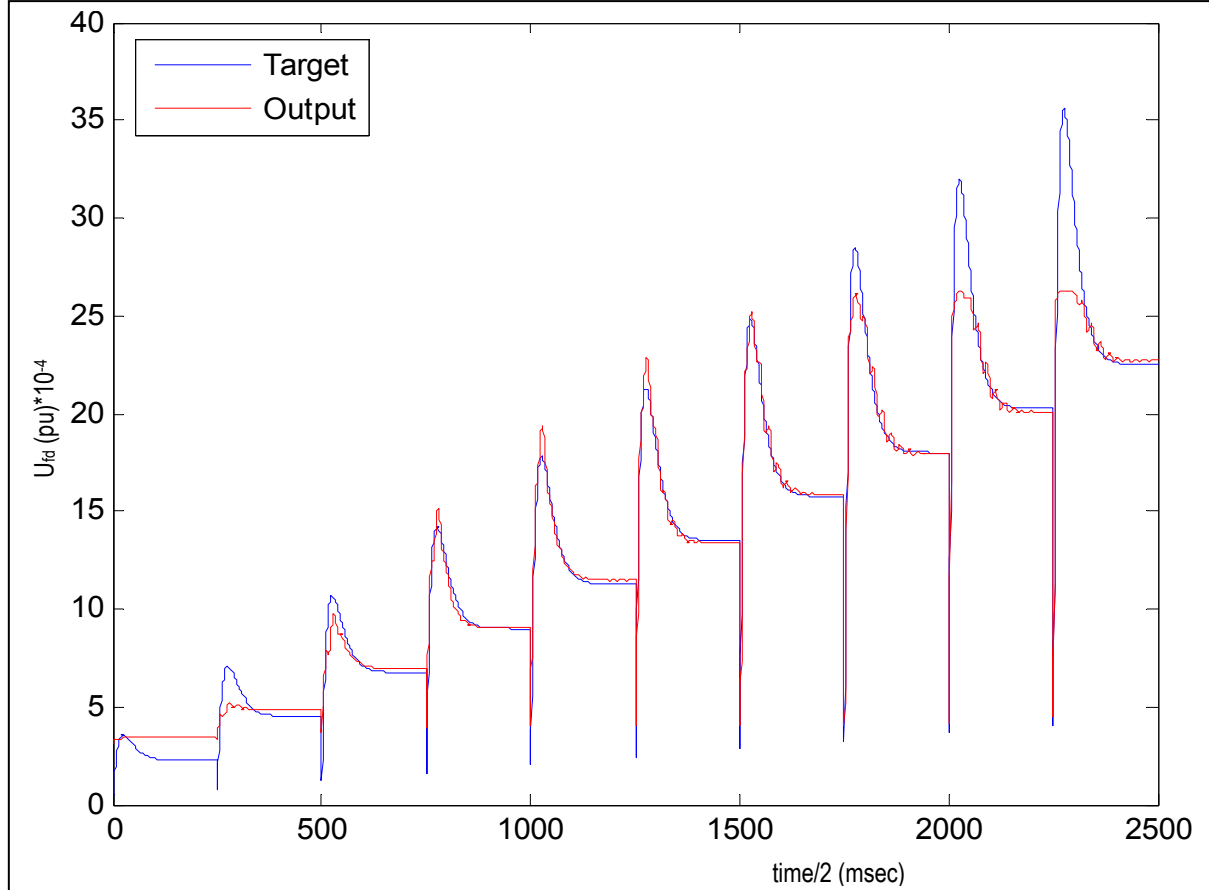
Τα δεδομένα που πρόκειται να χρησιμοποιηθούν για την εκπαίδευση των μοντέλων παρουσιάζουν μεγάλη διασπορά, αφού πρόκειται για διαφορετικά μεταξύ τους μεγέθη, πχ τάση εξόδου, ρεύμα εξόδου, γωνία ισχύος κλπ, έτσι η ανομοιομορφία των δεδομένων τα κάνει απαγορευτικά για χρήση τους σε νευρωνικό δίκτυο, αφού έχουμε διακύμανση σε διαφορετικές τάξεις μεγέθους. Η μεγάλη διαφορά μεταξύ των δεδομένων δημιουργεί σοβαρά προβλήματα, αφού θα έχουμε

μεγάλες ανισότητες μεταξύ των εισόδων, κάτι που ουσιαστικά θα έκανε ανενεργούς τους νευρώνες των εισόδων με τις μικρότερες τιμές, ενώ θα έχουν τεράστια βαρύτητα οι νευρώνες με μεγάλες τιμές, χωρίς αυτό να ανταποκρίνεται στην πραγματικότητα. Για να γίνει αντιληπτό πόσο κρίσιμο είναι αυτό σημείο ας φανταστούμε μία μεταβολή στην τάση από τα 20kV σε 20,1kV (20000V→20100V), δηλαδή έχουμε $\Delta V = 100V$, και η γωνιακή ταχύτητα από 2π rad/sec γίνεται 3π rad/sec, οπότε έχουμε σφάλμα π rad/sec. Αν τώρα συγκρίνουμε το 100 με το 3,14, καταλαβαίνουμε πως το μοντέλο αγνοεί κάθε άλλη ιδιότητα και «βλέπει» μόνο τα απόλυτα μεγέθη, οπότε θεωρεί πολύ πιο σημαντικό το μεγαλύτερο σφάλμα, δηλ. το 100, κατά συνέπεια το μικρότερο σφάλμα σχεδόν αγνοείται, αφού για να έχει ισοδύναμη δράση με το προηγούμενο, πρέπει να έχει βάρη σε κάθε είσοδο περίπου 30 φορές μεγαλύτερα από τα αντίστοιχα τις προηγούμενης τιμής. Ακόμη και στην περίπτωση που οι τιμές των βαρών για τις «μικρές» και τις «μεγάλες» απόλυτες τιμές των μεγεθών «εξισορροπούν» την διαφορά αυτή, δεν θα παρέχουν αντιπροσωπευτική αντιστοιχία αφού για παράδειγμα η μεταβολή στην τάση ανέρχεται στο 0,5% ενώ αυτή της γωνιακής ταχύτητας στο 50%. Ένα πρώτο βήμα προς αποφυγή αυτού του προβλήματος είναι η χρήση των μεγεθών ανηγμένα στο ανά μονάδα σύστημα (pu). Στη συνέχεια είναι απαραίτητο τα δεδομένα να κανονικοποιηθούν, ώστε να εξαλειφθεί αυτή η ανομοιομορφία. Στο σημείο αυτό όμως χρειάζεται ιδιαίτερη προσοχή στα όρια που θα χρησιμοποιηθούν, καθώς υπάρχει κίνδυνος πόλωσης του νευρωνικού δικτύου αν τα όρια δεν είναι σωστά. Έτσι αποφεύγουμε να πάρουμε το πεδίο [0 1], όπου το "0" και το "1" αποτελούν "ιδανικά" σημεία για πόλωση, με αποτέλεσμα το πεδίο κανονικοποίησης να είναι το [0,1 – 0,9]. Η συνάρτηση κανονικοποίησης (Εξ. 6.1) φαίνεται παρακάτω και όπως θα δούμε με επίλυση της εύκολα παίρνουμε την συνάρτηση από-κανονικοποίησης για να επαναφέρουμε τα δεδομένα στην αρχική τους μορφή.

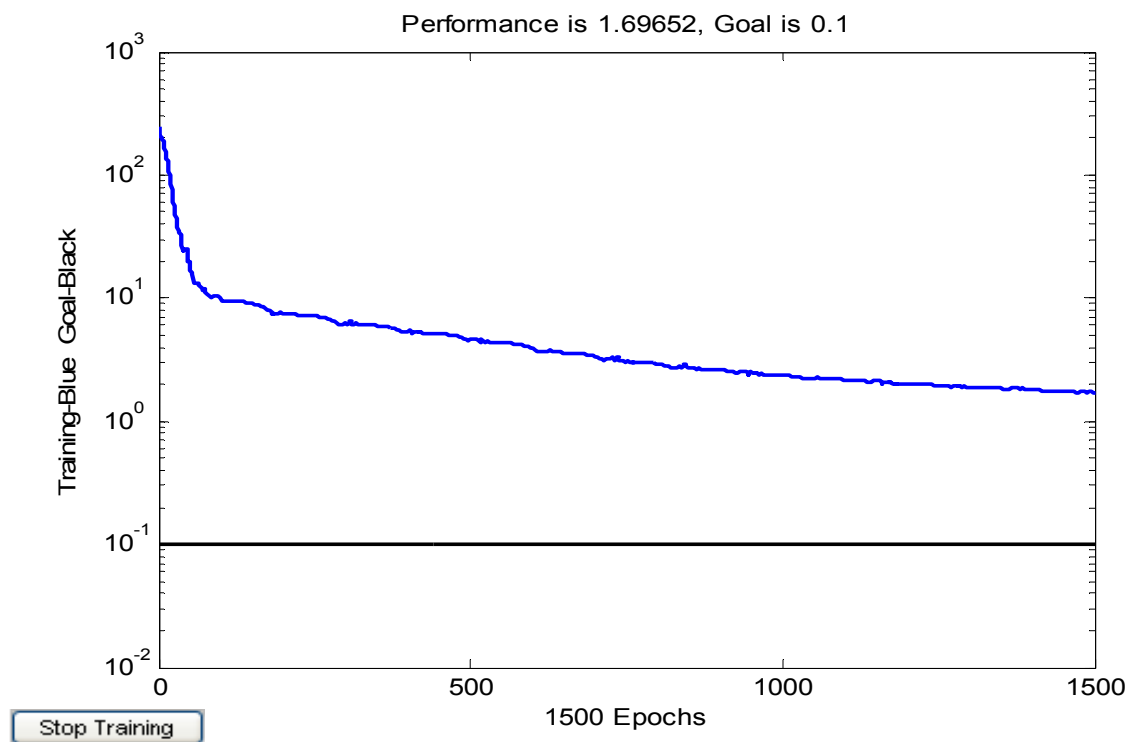
$$x_n = (0.9 - 0.1) \frac{x - x_{\min}}{x_{\max} - x_{\min}} + 0.1 \quad (6.1)$$

6.2.2 Δοκιμή Μοντέλου T.N.Δ. "MLP" .

Το πρώτο μέρος του πειράματος περιλαμβάνει δοκιμή με μοντέλο "MLP", όπως και στην περίπτωση του προβλήματος "IRIS" έτσι και τώρα ακολουθείται η ίδια διαδικασία επαναληπτικών δοκιμών. Τώρα, λόγω τις μεγαλύτερης πολυπλοκότητας που παρουσιάζει το πρόβλημα θα χρησιμοποιήσουμε περισσότερους νευρώνες, έτσι το μοντέλο με ένα κρυφό επίπεδο, θα έχει στην είσοδο 6 νευρώνες και το κρυφό επίπεδο θα φθάσει μέχρι 24 νευρώνες. Το μοντέλο με δύο κρυφά επίπεδα, θα έχει επίσης 6 νευρώνες στην είσοδο και τα κρυφά επίπεδα θα φτάσουν μέχρι 16 και 24 νευρώνες αντίστοιχα. Κάθε δοκιμή επαναλαμβάνεται 10 φορές. Γίνεται αντιληπτό πως ο συνολικός αριθμός δοκιμών για κάθε κατάσταση φθάνει τις 4080 δοκιμές και με χρόνο για κάθε δοκιμή που κυμαίνεται από ένα μέχρι δύο λεπτά ο συνολικός χρόνος τον οποίο χρειάστηκε ένας σύγχρονος Η/Υ (Παράρτημα Δ) για να ολοκληρώσει αυτό το πείραμα είναι περίπου 4 μέρες (συνεχούς λειτουργίας). Αυτό μπορεί να θεωρηθεί ως ένα μειονέκτημα του μοντέλου αυτού. Όπως φαίνεται από τα Σχ. 6.6 έως Σχ. 6.5, παρόλο τον μεγάλο χρόνο εκπαίδευσης και τη χρήση όλων των μέχρι τώρα γνωστών μεθόδων βελτίωσης του μοντέλου τα αποτελέσματα απέχουν αρκετά από το να χαρακτηριστούν ικανοποιητικά, πόσο μάλλον όταν ο σκοπός είναι η αντικατάσταση ενός ήδη πολύ καλού ελεγκτή. Το μέσο τετραγωνικό σφάλμα (MSE) παρέμεινε αρκετά μεγάλο με το ελάχιστο να γίνεται 1,697 για μοντέλο με ένα κρυφό επίπεδο και 1,412 για μοντέλο με 2 κρυφά επίπεδα, τη στιγμή που ο στόχος ήταν να προσεγγίσει μέγεθος της τάξης 10^{-1} , το σφάλμα παρέμεινε μεγαλύτερο της τάξης του 10^0 . Έτσι είναι άσκοπο να συνεχίσουμε τη μελέτη σε αυτό το μοντέλο για το συγκεκριμένο σκοπό, αφού ακόμη και στην καλύτερη περίπτωση το αποτέλεσμα δεν ικανοποιεί.

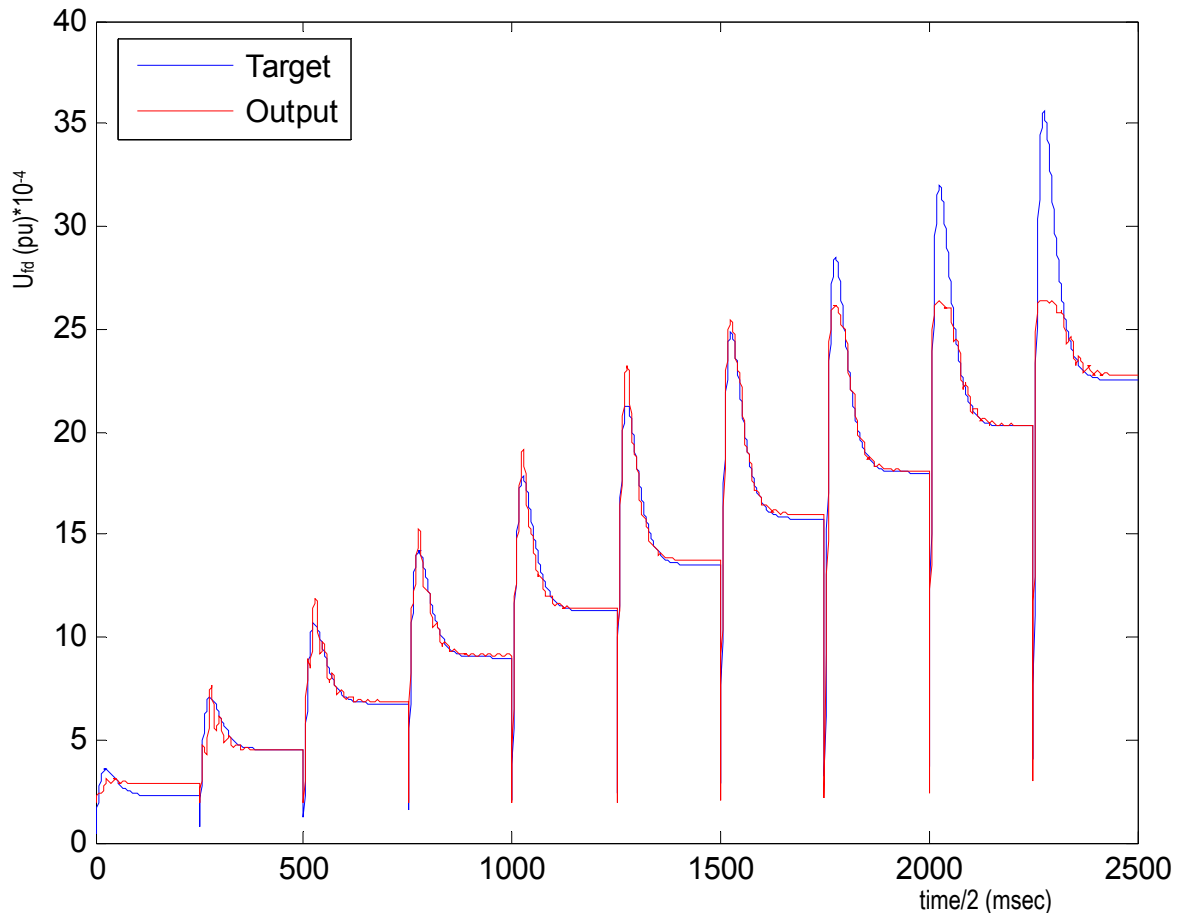


Σχ. 6.2 Έξοδος μοντέλου (6x24x1)

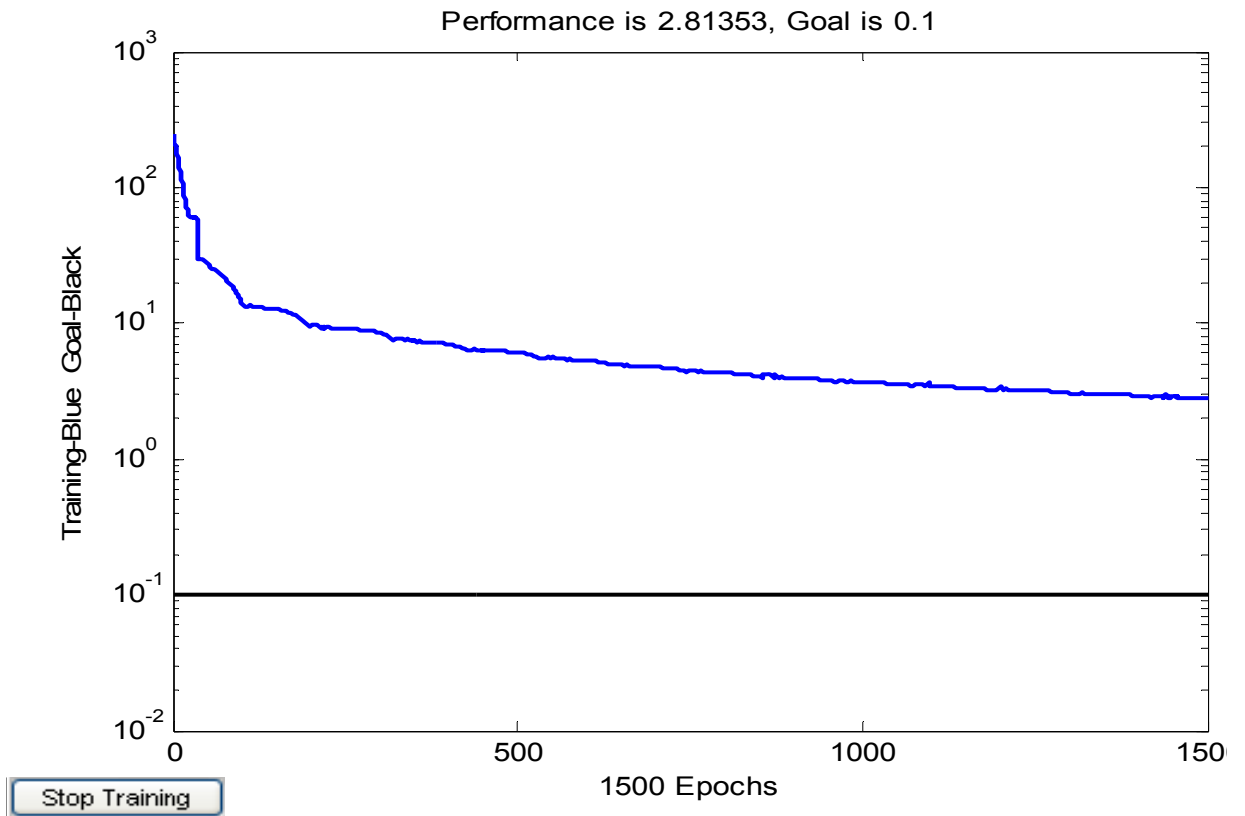


Σχ. 6.3 Εκπαίδευση μοντέλου (6x24x1)

Στο Σχ. 6.2 παρατηρούμε ότι το μοντέλο T.N.Δ δεν συγκλίνει ικανοποιητικά σε όλες τις περιπτώσεις διαταραχών. Ειδικότερα η έξοδος τους αποκλίνει σημαντικά στις μικρές και μεγάλες τιμές διαταραχών, με κύριο χαρακτηριστικό την υπέρβαση της επιθυμητής τιμής, πράγμα που οδηγεί σε ανεπιθύμητες καταστάσεις όταν πρόκειται για έλεγχο διέγερσης. Παρόλο που στο Σχ. 6.2 η απόκλιση στη πρώτη περίπτωση φαίνεται οπτικά μικρή σε απόλυτα μεγέθη έχουμε μεγάλη απόκλιση και σε καμία περίπτωση αυτό δεν μπορεί να θεωρηθεί αποδεκτό. Το ίδιο θα παρατηρήσουμε και στο Σχ. 6.4 όπου έχουμε τη καλύτερη επίδοση μοντέλου με 2 κρυφά επίπεδα (6x13x3x1), όπου είναι εμφανής μία μικρή βελτίωση σε σχέση με το μοντέλο με ένα κρυφό επίπεδο.



Σχ. 6.4 Εξοδος μοντέλου (6x13x3x1)



Σχ. 6.5 Εκπαίδευση μοντέλου (6x13x3x1)

6.2.3 Δοκιμή Μοντέλου T.N.Δ. “RBF” .

Στη συνέχεια οι δοκιμές μας επικεντρώνονται σε ένα διαφορετικό μοντέλο νευρωνικού δικτύου, το οποίο καλείται RBF. Το πείραμα τώρα δεν μπορεί να έχει ακριβώς την ίδια δομή με προηγούμενες, αλλά είναι απαραίτητο να γίνουν τροποποιήσεις του κώδικα και της δομής του πειράματος, λόγω της αρχιτεκτονικής και του τρόπου εκπαίδευσης του T.N.Δ. RBF. Όπως ήδη έχουμε αναφέρει ένα ακτινικό νευρωνικό δίκτυο χρησιμοποιεί Gaussian συναρτήσεις ως συναρτήσεις ενεργοποίησης (π.χ. Radbas, Σχ. 5.5) και συνάρτηση υπολογισμού της Ευκλείδειας απόστασης του βάρους από την είσοδο (Εξ. 6.2), ενώ ο αλγόριθμος εκπαίδευσης μεταβάλλει δυναμικά τον αριθμό των νευρώνων ώστε να επιτευχθεί το επιθυμητό αποτέλεσμα. Σημειώνεται ότι αν δεν περιοριστεί με εξωτερική επίβλεψη ο αριθμός των νευρώνων μπορεί να φθάσει τον αριθμό των διανυσμάτων εκπαίδευσης. Για να αποφύγουμε την ανεξέλεγκτη αύξηση των νευρώνων, πράγμα που θα το καθιστούσε άχρηστο αν πρόκειται να μεταφερθεί σε μικροελεγκτή που εκ των πραγμάτων έχει περιορισμένη μνήμη και υπολογιστική ισχύ, θέτουμε όριο στο μέγιστο αριθμό νευρώνων. Επειδή όμως δεν είναι δυνατόν να γνωρίζουμε εκ των προτέρων τη χρυσή τομή μεταξύ απόδοσης και αριθμού νευρώνων για το συγκεκριμένο πρόβλημα, κάνουμε διερεύνηση σε ένα πεδίο τιμών από το “1” μέχρι το “256” [1 – 256] και η μεταβολή του ορίου των νευρώνων θα ακολουθεί την γεωμετρική πρόοδο του 2 (2ⁿ) ώστε να έχουμε και οικονομία χρόνου.

$$d = \sum \sqrt{(x_{nxi} - y_{nxi})^2} \quad (6.2)$$

Οι νευρώνες του κρυφού επιπέδου έχουν μη γραμμική συνάρτηση ενεργοποίησης (Radbas), ενώ οι νευρώνες του επιπέδου εξόδου έχουν γραμμική συνάρτηση ενεργοποίησης (purelin). Η διαδικασία εκπαίδευσης είναι παρόμοια με την εκπαίδευση του MLP, όμως δεν ορίζεται μέγιστος αριθμός εποχών, αλλά μέγιστος αριθμός νευρώνων, έτσι με την ολοκλήρωση κάθε ελέγχου του σφάλματος προστίθενται νευρώνες μέχρι να επιτευχθεί το ελάχιστο σφάλμα (SSE=10⁻⁴) ή ο αριθμός των νευρώνων φτάσει το όριο. Η εκπαίδευση στη συγκεκριμένη περίπτωση γίνεται αρχικά με όλα τα δεδομένα γνωστά στο νευρωνικό δίκτυο και στη συνέχεια αποκρύπτεται μέρος των δεδομένων ώστε να επιτευχθεί το καλύτερο δυνατό αποτέλεσμα με όσο το δυνατόν λιγότερους νευρώνες και με τα ελάχιστα δυνατά διανύσματα εκπαίδευσης. Μετά την ολοκλήρωση κάθε εκπαίδευσης το μοντέλο δοκιμάζεται σε άγνωστα στοιχεία (εκτός από την περίπτωση όπου όλα τα δεδομένα είναι γνωστά). Για το λόγο αυτό είναι αναγκαίο να χρησιμοποιηθεί διαφορετικό κριτήριο από το (SSE η MSE) ώστε να επιλέξουμε το καλύτερο μοντέλο, έτσι χρησιμοποιούνται δύο νέοι όροι ως κριτήρια σύγκρισης, το IAET (integral of absolute error in time, ολοκλήρωμα του απόλυτου σφάλματος στο χρόνο, Εξ. 6.3) και το ISET (integral of squared error in time, ολοκληρωμένο τετραγωνικό σφάλμα στο χρόνο, Εξ. 6.4). Κατόπιν επιβάλλεται να επαναφέρουμε τη μεταβλητή εξόδου στην αρχική τάξη μεγέθους της. Για να γίνει αυτό απαιτείται μία συνάρτηση αντίστροφη της συνάρτησης κανονικοποίησης (από-κανονικοποίηση, de-normalization) η οποία προκύπτει από την συνάρτηση κανονικοποίησης (normalize) με επίλυση ως προς x (Εξ. 6.5).

$$J_1 = \int_0^T |e_t(t)| dt \quad (6.3)$$

$$J_2 = \int_0^T e_t^2 \cdot dt \quad (6.4)$$

$$\begin{aligned}
x_n &= (0.9 - 0.1) \frac{x - x_{\min}}{x_{\max} - x_{\min}} + 0.1 \\
\Leftrightarrow (0.9 - 0.1)(x - x_{\min}) &= (x_n - 0.1)(x_{\max} - x_{\min}) \\
\Leftrightarrow (x - x_{\min}) &= \frac{(x_n - 0.1)(x_{\max} - x_{\min})}{(0.9 - 0.1)} \\
\Leftrightarrow x &= \frac{(x_n - 0.1)(x_{\max} - x_{\min})}{(0.9 - 0.1)} + x_{\min} \\
\Leftrightarrow x &= \frac{(x_n - 0.1)(x_{\max} - x_{\min})}{0.8} + x_{\min}
\end{aligned} \tag{6.5}$$

6.2.4 Αποτελέσματα Δοκιμής Μοντέλου T.N.Δ. RBF.

Τα μοντέλα τύπου RBF εκπαιδεύονται για 6 διαφορετικά σενάρια που σχετίζονται με τον τρόπο με τον οποίο παρουσιάζονται τα δείγματα εκπαίδευσης στην είσοδο του αναφορικά με τις περιπτώσεις διαταραχών του συστήματος. Συγκεκριμένα έχουμε:

Σενάριο	Δείγματα εκπαίδευσης
1	Γνωστά τα μισά δεδομένα, δηλ. των περιπτώσεων διαταραχής 1, 3, 5, 7, 9
2	Γνωστά τα μισά δεδομένα, δηλ. των περιπτώσεων διαταραχής 2, 4, 6, 8, 10
3	Γνωστά τα μισά δεδομένα, δηλ. των περιπτώσεων διαταραχής 1 – 5
4	Γνωστά τα μισά δεδομένα, δηλ. των περιπτώσεων διαταραχής 6 – 10
5	Γνωστά τα 7/10 των δεδομένων, δηλ. των περιπτώσεων διαταραχής 1 – 2, 4 – 5, 7 – 8, 10
6	Γνωστά όλα τα δεδομένα στο νευρωνικό δίκτυο, δηλ. των περιπτώσεων διαταραχής 1 – 10

Στους Πίνακες 6.2 και 6.3 παρατίθενται τα αποτελέσματα εκπαίδευσης των παραπάνω σεναρίων, ενώ στον Πίνακα 6.4 φαίνονται οι χρόνοι εκπαίδευσης.

Πίνακας 6.2 Απόδοση T.N.Δ. RBF ως προς το J_1 (IAET)

Νευρώνες	1	2	4	8	16	32	64	128	256
Σενάριο	IAET								
1	22,64103	6,087756	6,517798	3,419735	2,656485	0,827513	2,176475	2,308508	3,123646
2	17,44163	6,067907	6,085685	3,089503	1,062143	0,363224	0,210208	0,518977	0,406827
3	66,39777	29,21552	26,79766	18,88395	55,57821	32,44026	60,33609	69,31607	79,16425
4	58,26824	17,6163	9,190208	11,90348	6,443453	10,12999	11,01455	2,395841	2,395841
5	18,20006	6,787028	4,689028	2,752842	2,037555	0,326548	0,225832	0,256569	0,224716
6	18,82227	6,835233	5,362256	2,280766	1,527264	0,350427	0,063175	0,027052	0,027052

Πίνακας 6.3 Απόδοση T.N.Δ. RBF ως προς το J₂ (ISET)

Νευρώνες	1	2	4	8	16	32	64	128	256
Σενάριο	ISET								
1	179,2202	15,49212	19,41277	5,789601	2,96836	0,466202	3,838258	3,70733	7,47425
2	103,9113	12,93001	14,33932	3,987748	0,715818	0,038658	0,01491	0,10253	0,046413
3	1110,776	261,5869	220,1892	154,7108	1114,058	417,8544	1596,117	2241,505	2970,597
4	734,2278	69,64421	31,91211	55,24647	10,45506	49,63216	53,74086	2,26988	2,26988
5	123,0132	16,627	13,14063	4,364475	1,548905	0,049679	0,019522	0,02433	0,020191
6	122,3945	19,28189	12,46949	3,386722	0,959684	0,053963	0,002643	0,000385	0,000385

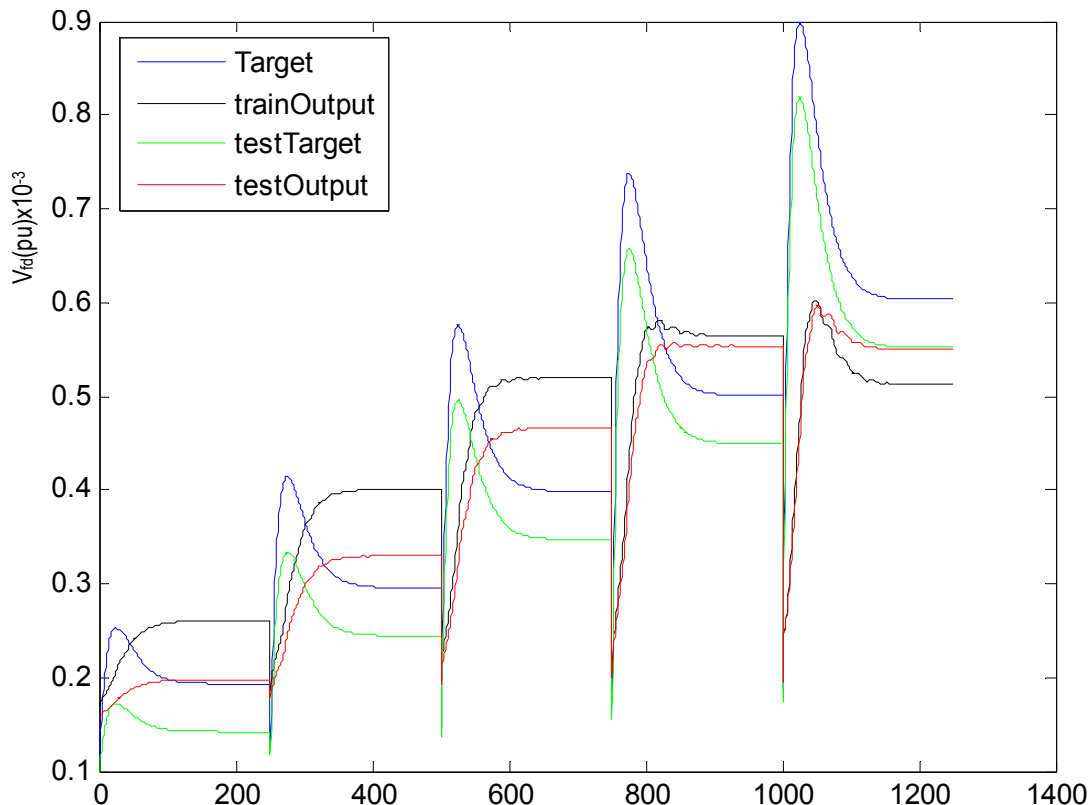
Πίνακας 6.4 Χρόνος εκπαίδευσης μοντέλων RBF.

Νευρώνες	1	2	4	8	16	32	64	128	256
Σενάριο	sec								
1	1,078	1,39	1,266	1,859	3,156	5,984	12,282	29,891	91,75
2	0,922	1,203	1,266	1,891	3,156	5,844	12,172	29,922	94,359
3	0,89	1,078	1,343	1,968	3,297	5,828	12	29,437	92,937
4	0,954	1,125	1,421	2,063	3,328	6,016	12,031	19,797	19,782
5	1,828	2,437	2,765	3,906	6,219	11,032	22,594	53,797	154,938
6	4,141	4,734	5,891	8,094	12,625	22,187	42,828	52,39	52,453

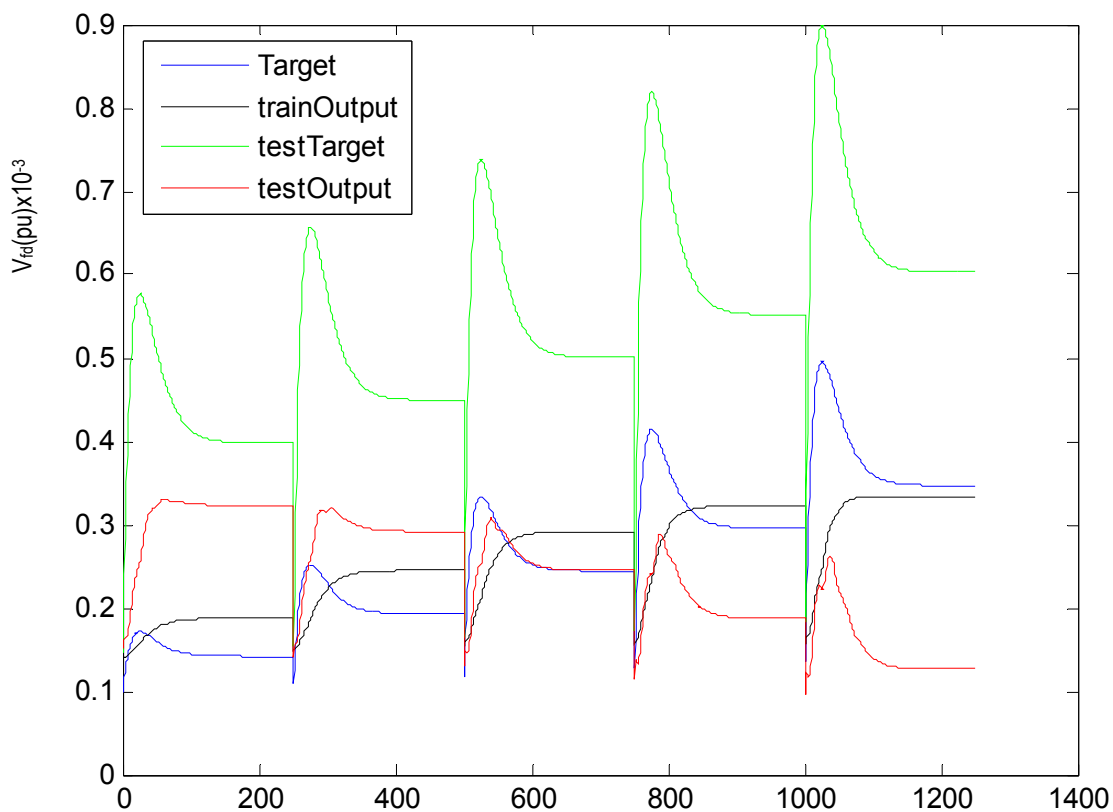
Παρατηρώντας τα αποτελέσματα εύκολα διαπιστώνουμε πως το μικρότερο σφάλμα παρουσιάστηκε από μοντέλο (6x256x1) με όλα τα δεδομένα γνωστά (Σενάριο 6), πράγμα αναμενόμενο βέβαια. Παρουσιάζεται ιδιαίτερο ενδιαφέρον στα Σενάρια 2 και 3 καθώς στο Σενάριο 2 έχουμε σημαντικά καλύτερα αποτελέσματα από κάθε άλλο Σενάριο με τα μισά δεδομένα γνωστά (1,3,4) πράγμα που σημαίνει πως τα διανύσματα εκπαίδευσης στο δεύτερο Σενάριο έχουν τέτοια διασπορά ώστε να παραμένουν αντιπροσωπευτικά του συνόλου. Στο Σενάριο 3 συμβαίνει το ακριβώς αντίθετο, δηλ. πρόκειται για το Σενάριο που έχει μακράν το μεγαλύτερο σφάλμα έστω και αν το ποσοστό γνωστών και αγνώστων στοιχείων είναι ίδιο με τα Σενάρια 1,2 και 4. Αυτό οδηγεί στο συμπέρασμα ότι τα δεδομένα σε αυτό το Σενάριο έχουν τη χειρότερη δυνατή διασπορά. Βέβαια λογικό είναι και στο Σενάριο 5 να υπάρχει μικρό σφάλμα, αφού η αναλογία αγνώστων προς γνωστά δεδομένα είναι 3/7 και μία τέτοια περίπτωση εκπαίδευσης θα μπορούσε να αποτελέσει τη χρυσή τομή ώστε να επιλέξουμε και να εκπαιδεύσουμε το κατάλληλο μοντέλο χωρίς να εκπαιδεύσουμε το μοντέλο με το μέγιστο δυνατό αριθμό διανυσμάτων αφενός μεν για οικονομία χρόνου, αφετέρου δε για να αποφύγουμε την παγίδα της υπερεκπαίδευσης. Μεγάλο ενδιαφέρον παρουσιάζει και ο Πίνακας 6.4 με τους χρόνους εκπαίδευσης, καθώς διαπιστώνουμε πως ο αριθμός των νευρώνων δεν είναι ο μόνος παράγοντας που επηρεάζει το χρόνο εκπαίδευσης, αλλά φαίνεται ότι ο αριθμός διανυσμάτων εκπαίδευσης αποτελεί σημαντικότερο παράγοντα. Σε περιπτώσεις όπου ο χρόνος εκπαίδευσης φαίνεται να παραμένει σταθερός, ενώ το όριο των νευρώνων αυξάνεται, σημαίνει πως η διαδικασία σταματάει λόγω άλλης συνθήκης (ανοχή σφάλματος). Τέλος το κρίσιμο συμπέρασμα το οποίο εξάγεται είναι πως μετά τους 64 νευρώνες στο κρυφό επίπεδο το μοντέλο RBF δεν παρουσιάζει βελτίωση ανάλογη με την αύξηση των νευρώνων (σε ορισμένες περιπτώσεις έχουμε και το ακριβώς αντίθετο αποτέλεσμα) λόγω υπερεκπαίδευσης. Αν συγκρίνουμε την απόδοση συναρτήσει της κατανομής των διανυσμάτων εκπαίδευσης θα δούμε πως το Σενάριο 2 αποτελεί ικανοποιητική λύση, δεδομένου ότι με 64 νευρώνες και την συγκεκριμένη κατανομή έχουμε πολύ μικρό σφάλμα σε άγνωστα δεδομένα. Αυτό γιατί, ένα μοντέλο με 64 νευρώνες στο κρυφό επίπεδο θα έχει ικανοποιητικά αποτελέσματα και σημαντική οικονομία χώρου (μνήμης) και υπολογιστικής ισχύος, αφού απαιτεί τουλάχιστον το 1/4 σε σχέση με ένα μοντέλο με 256 νευρώνες. Για να γίνει αντιληπτή η διαφορά των παραπάνω αποτελεσμάτων πρέπει να δούμε σε αντιπαράθεση την έξοδο του T.N.Δ. που αφορά τα σενάρια 2 και 3 σε κάθε επίπεδο διαταραχής, αλλά και τα υπόλοιπα δεδομένα σε γραφική αναπαράσταση.

6.3 Επιλογή Μοντέλου Τ.Ν.Δ. “RBF” .

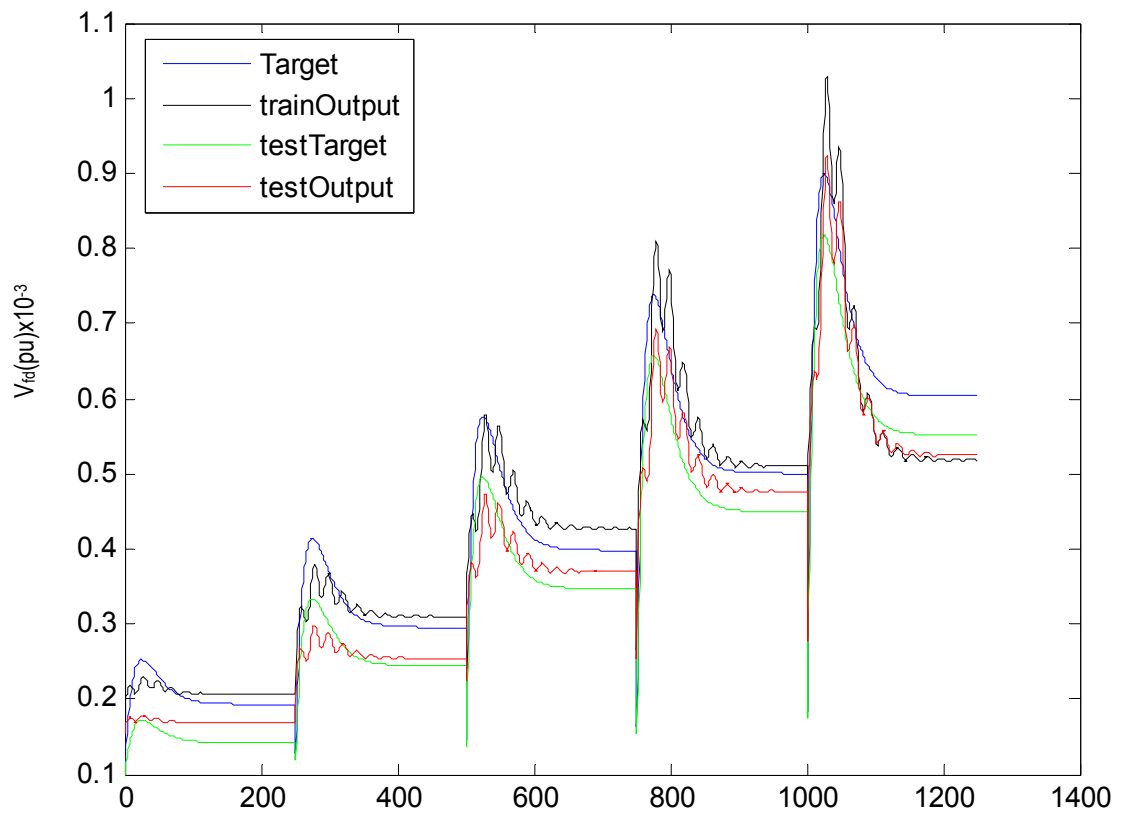
- Στα Σχ. 6.6 έως Σχ. 6.23 απεικονίζονται οι συμπεριφορές των μοντέλων Τ.Ν.Δ. RBF μετά την εκπαίδευση τους για όλους τους συνδυασμούς του αριθμού νευρώνων στο κρυφό επίπεδο (1 έως 256) και τα σενάρια 2 και 3.



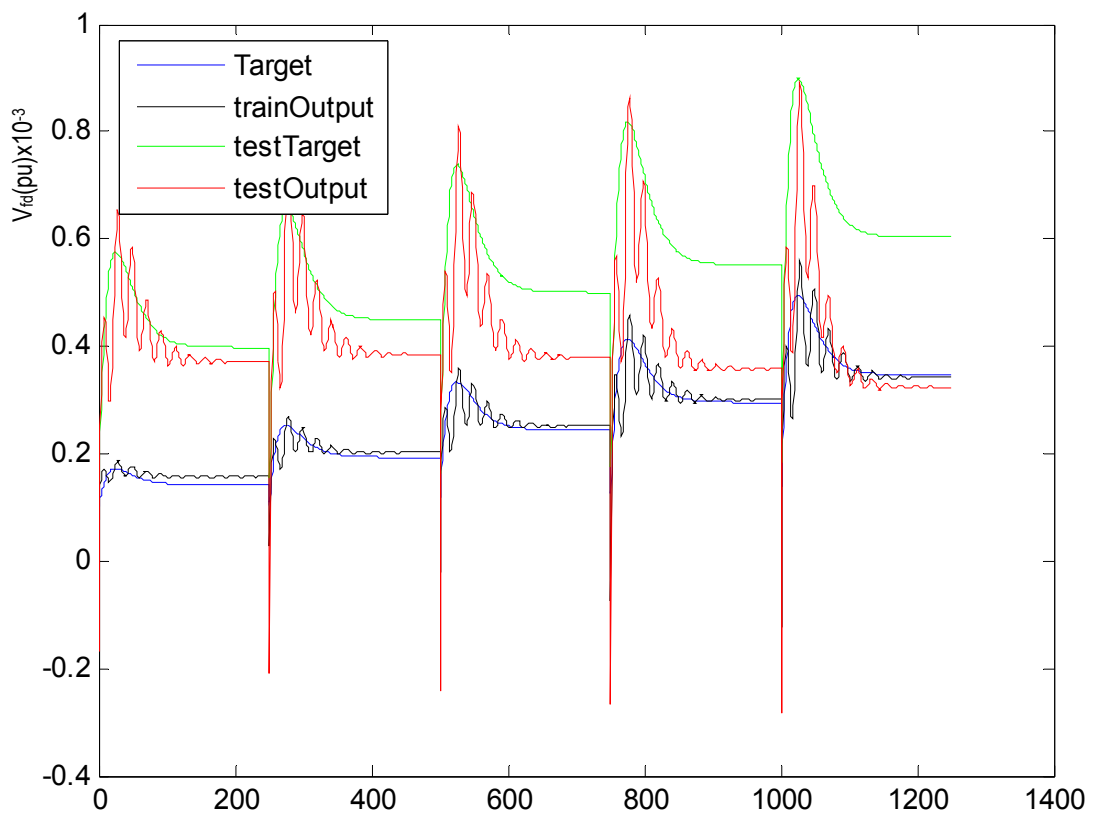
Σχ. 6.6 Έξοδος μοντέλου RBF (6x1x1), Σενάριο 2



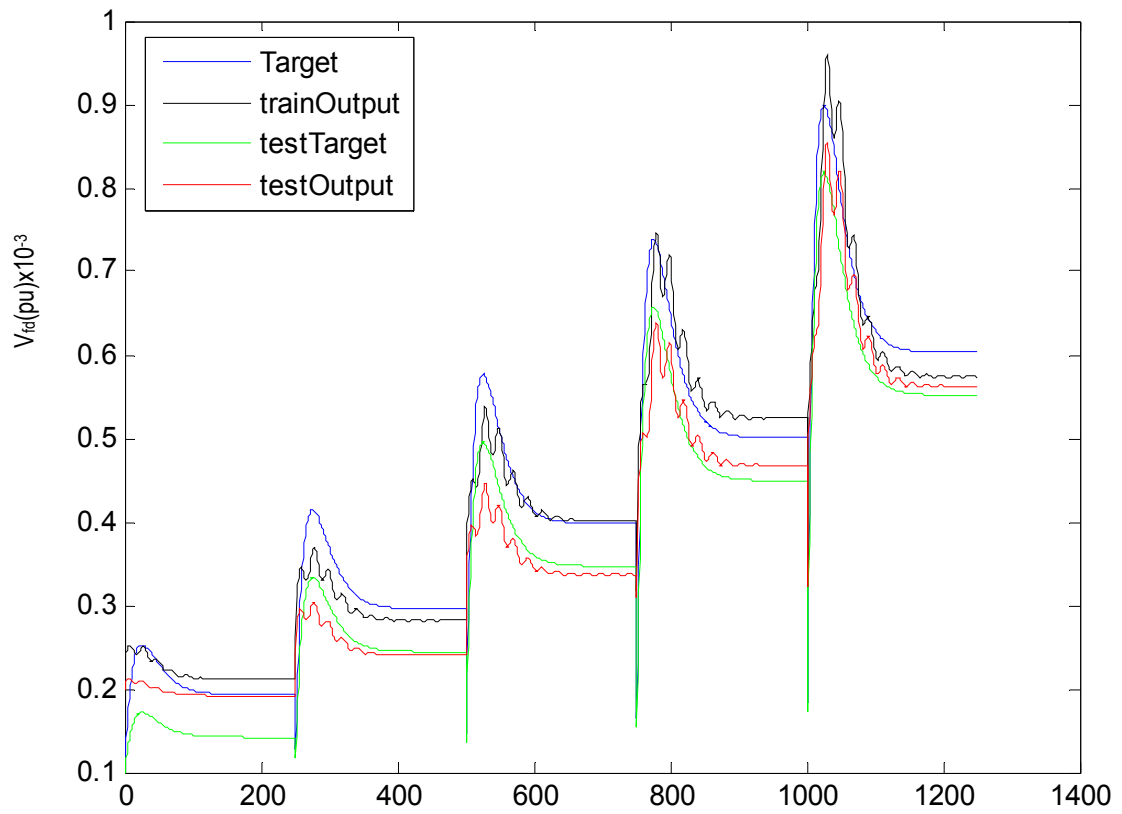
Σχ. 6.7 Έξοδος μοντέλου RBF (6x1x1), Σενάριο 3



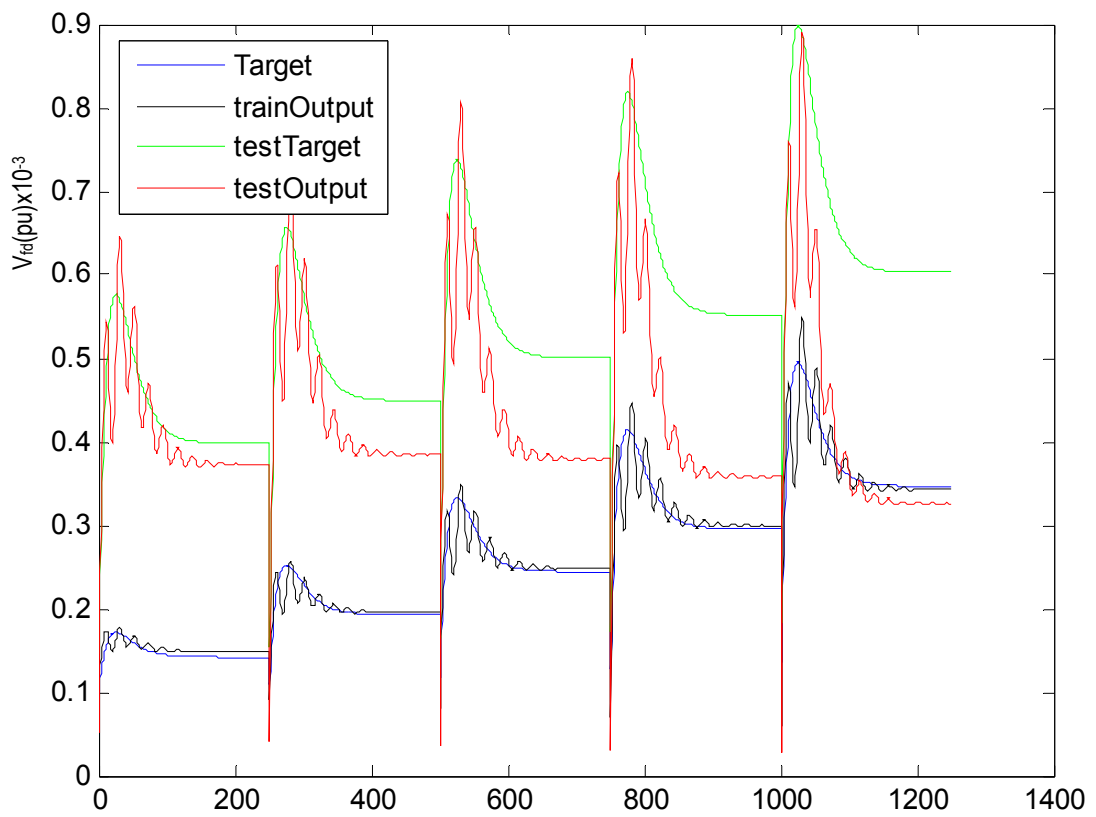
Σχ. 6.8 Εξόδος μοντέλου RBF (6x2x1), Σενάριο 2



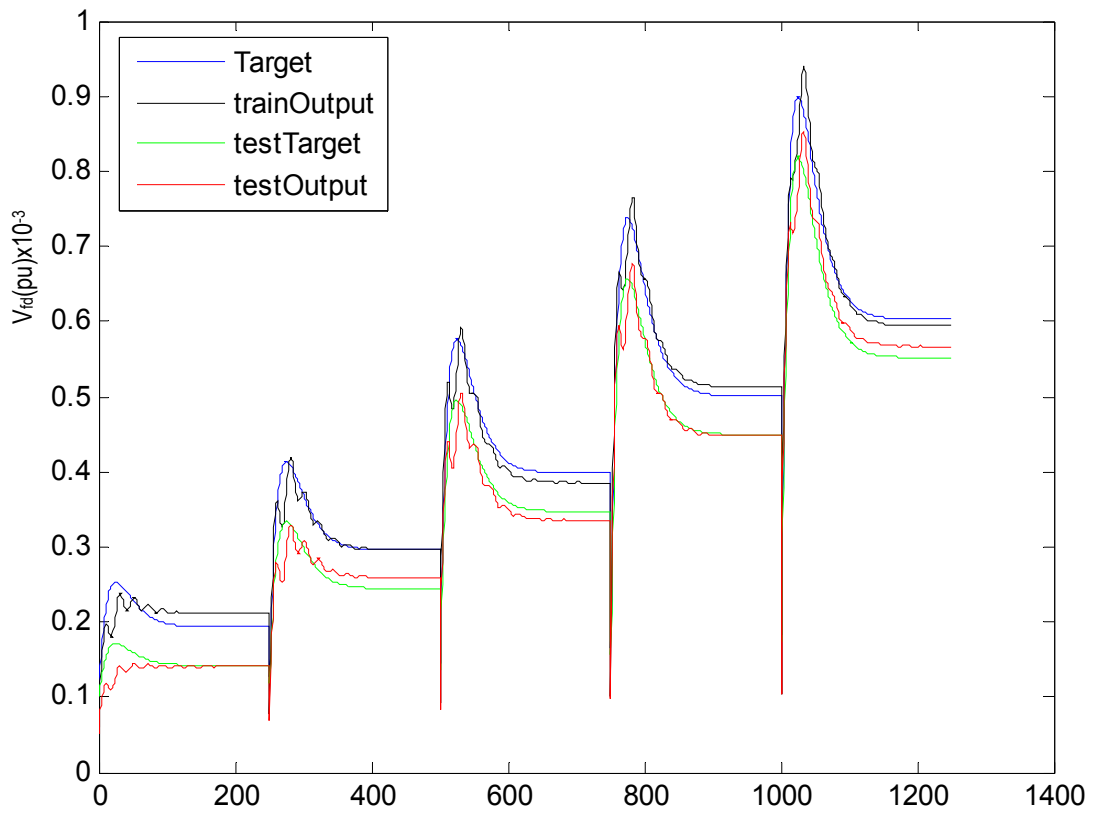
Σχ. 6.9 Εξόδος μοντέλου RBF (6x2x1), Σενάριο 3



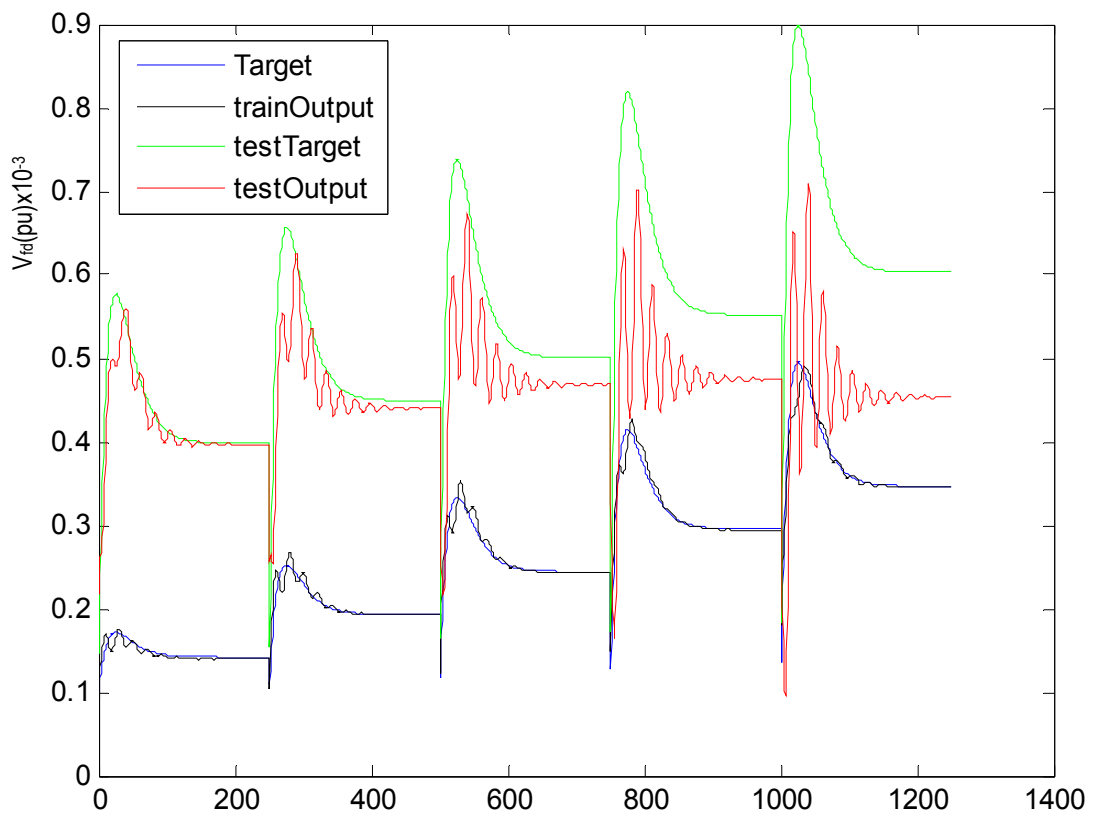
Σχ. 6.10 Έξοδος μοντέλου RBF (6x4x1), Σενάριο 2



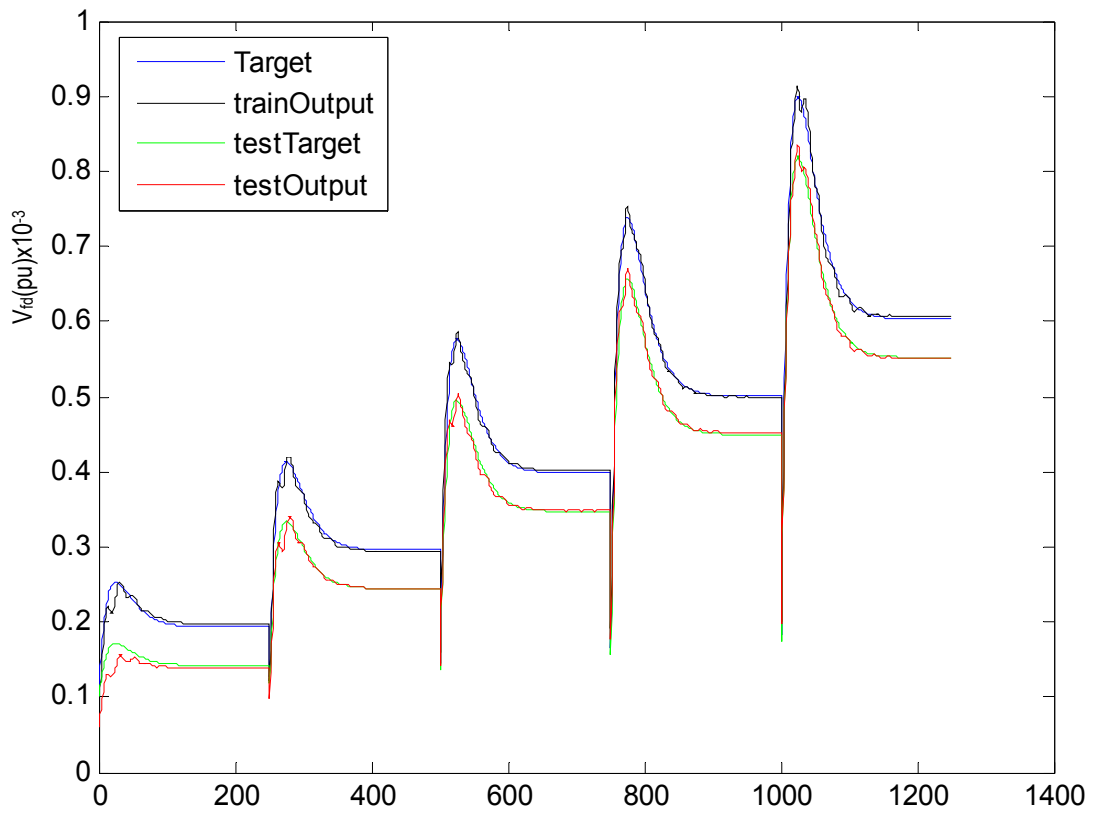
Σχ. 6.11 Έξοδος μοντέλου RBF (6x4x1), Σενάριο 3



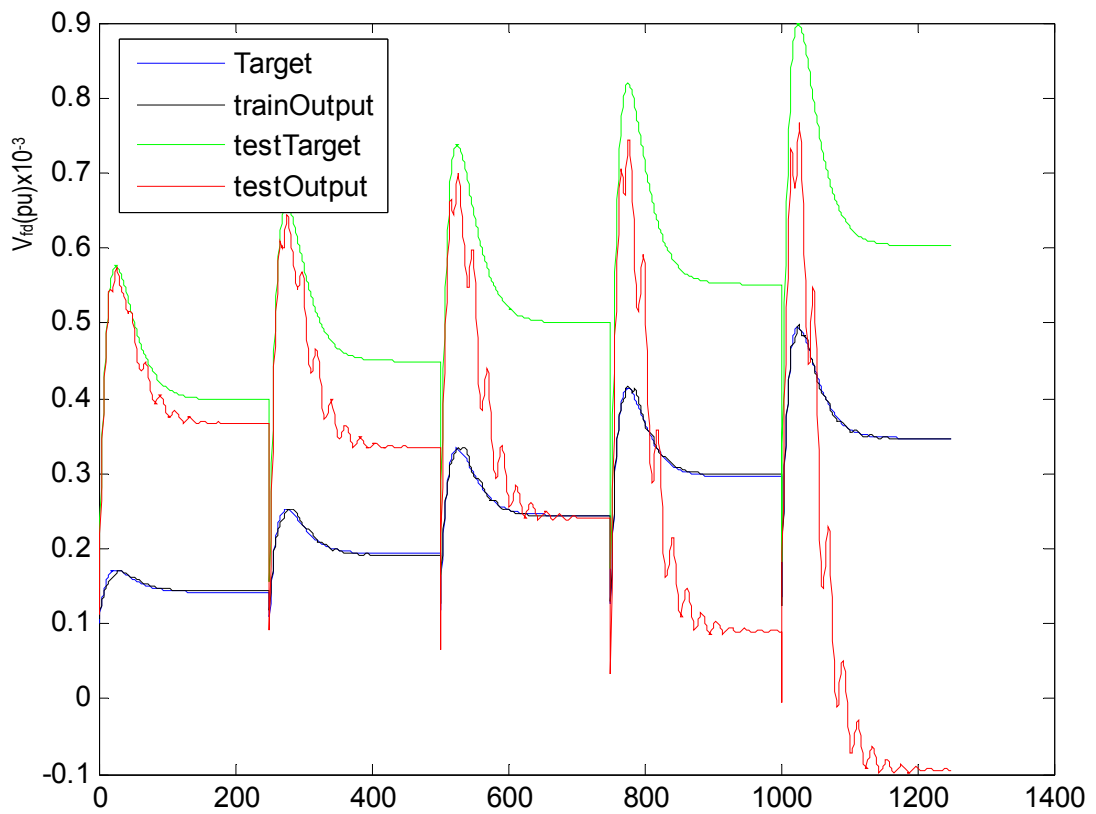
Σχ. 6.12 Έξοδος μοντέλου RBF (6x8x1), Σενάριο 2



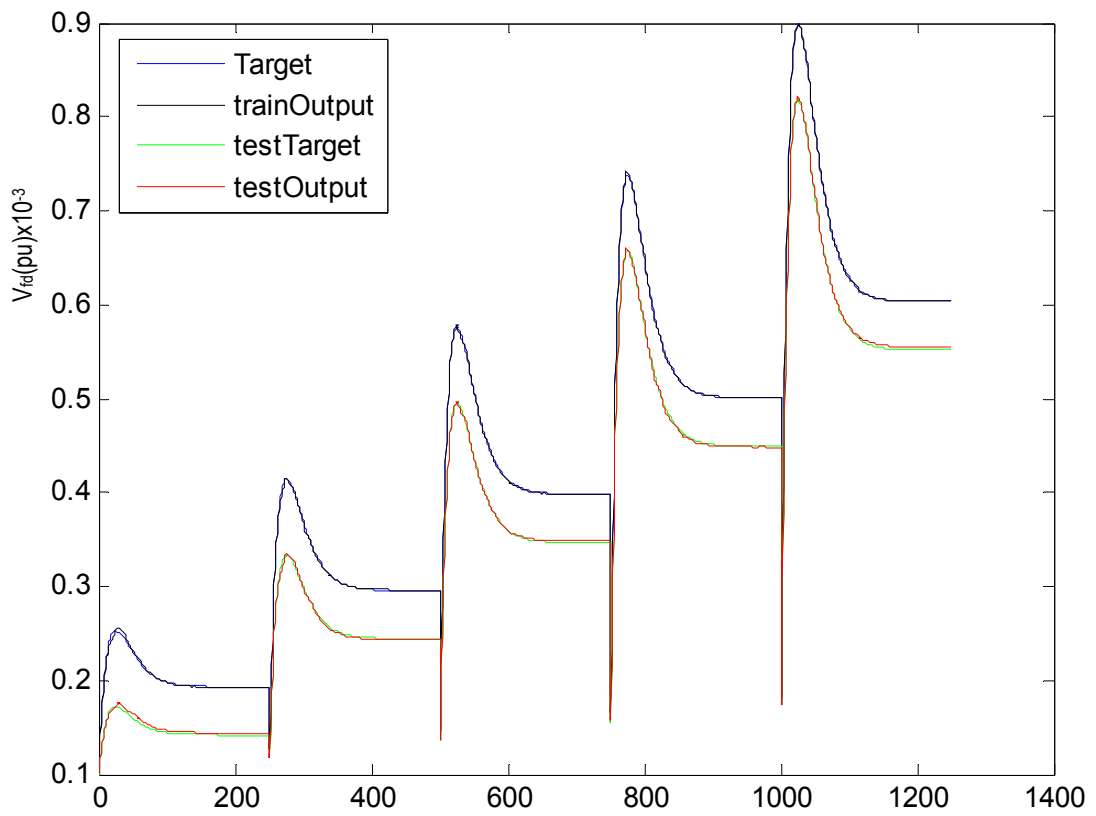
Σχ. 6.13 Έξοδος μοντέλου RBF (6x8x1), Σενάριο 3



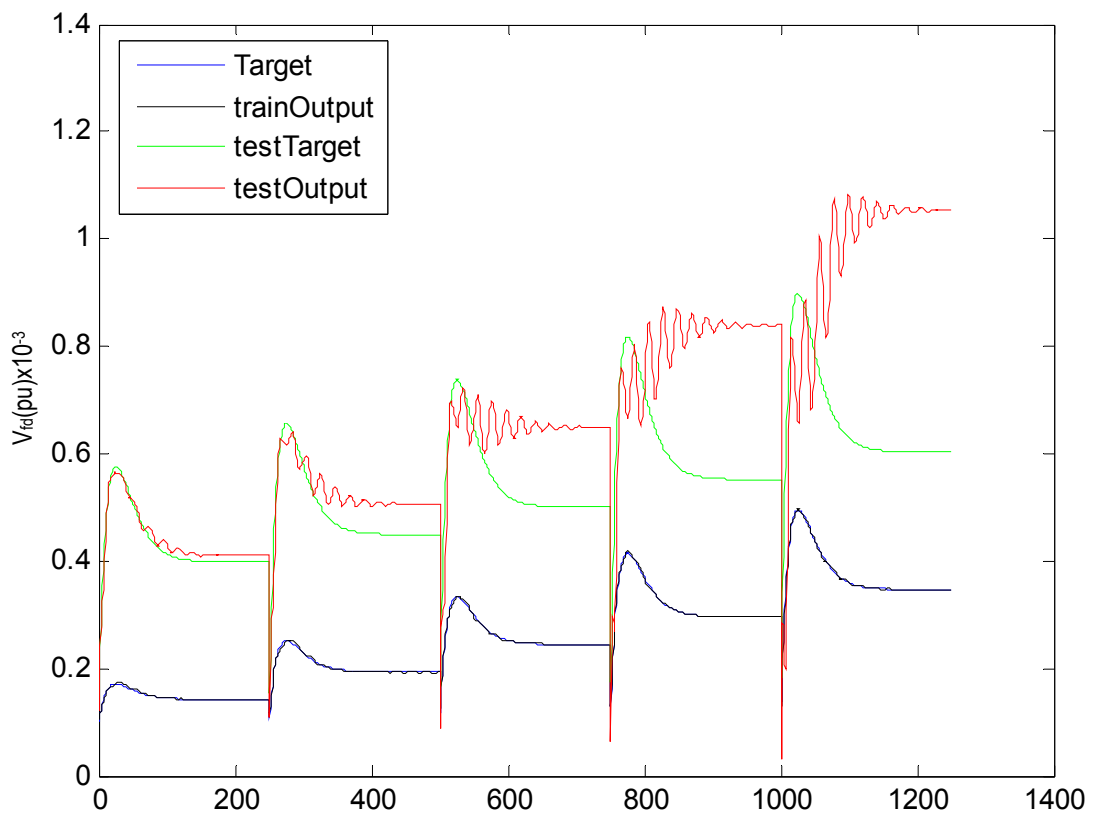
Σχ. 6.14 Έξοδος μοντέλου RBF (6x16x1), Σενάριο 2



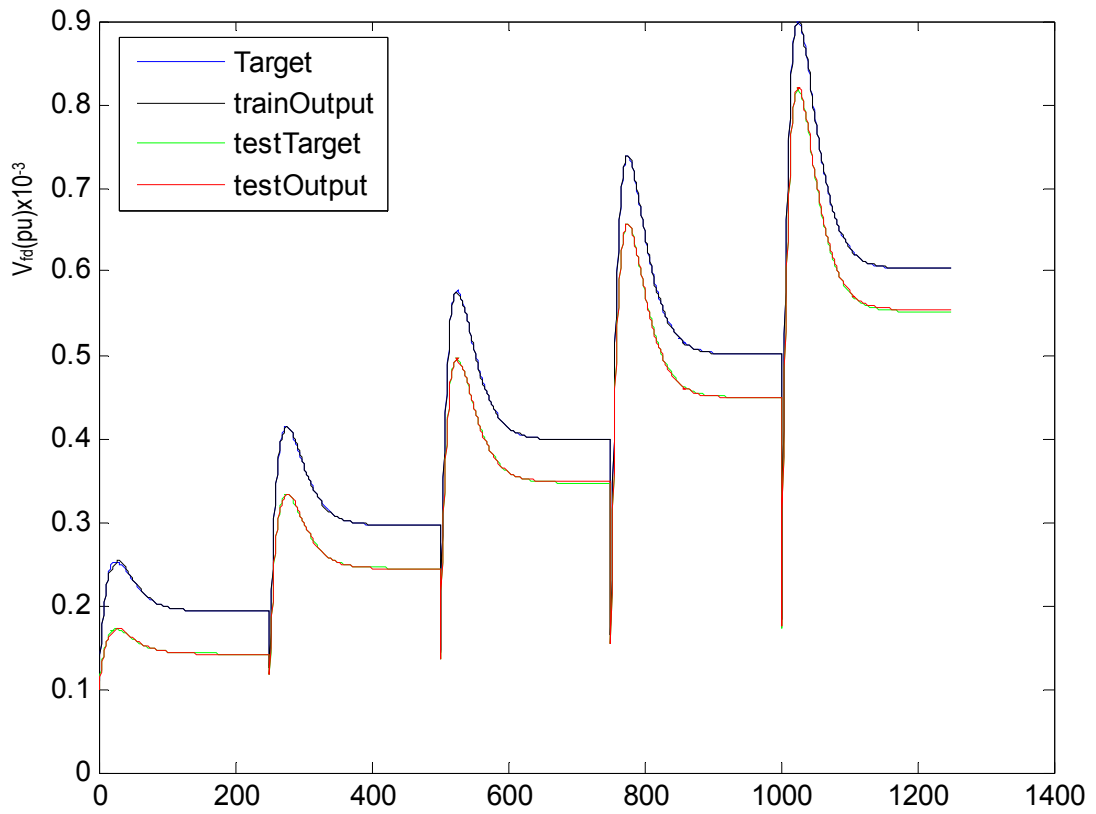
Σχ. 6.15 Έξοδος μοντέλου RBF (6x16x1), Σενάριο 3



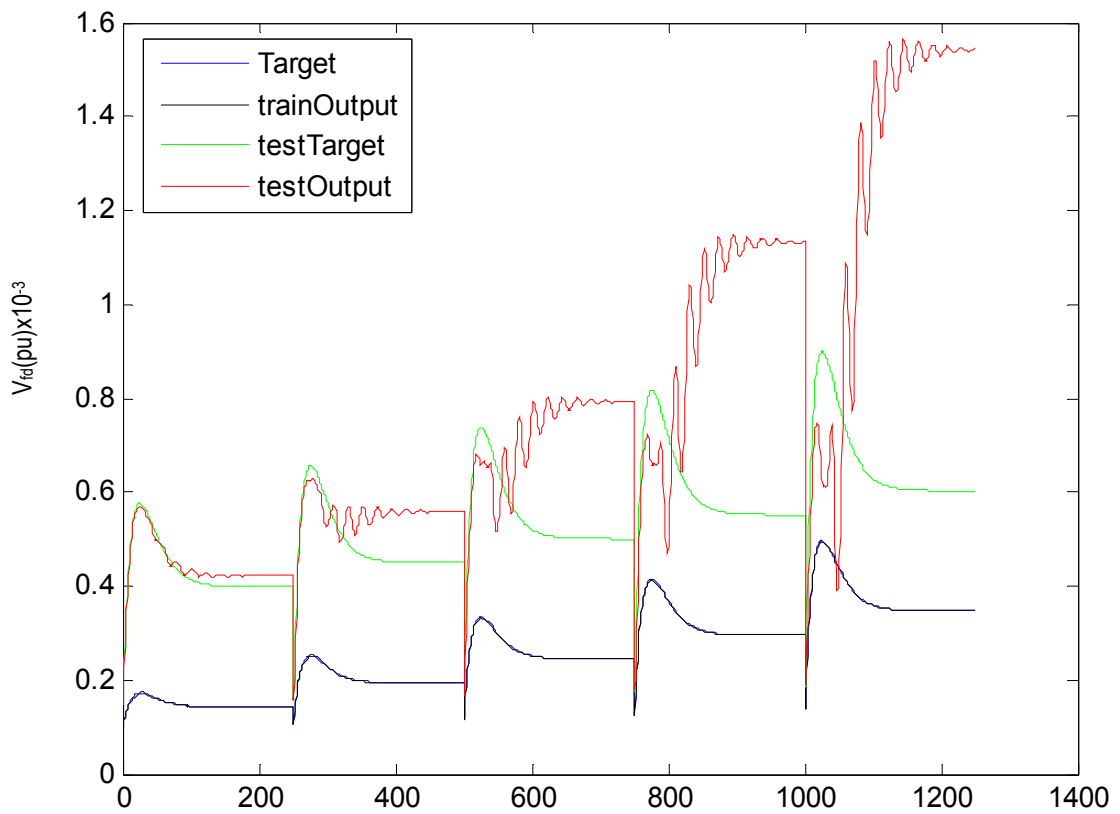
Σχ. 6.16 Έξοδος μοντέλου RBF (6x32x1), Σενάριο 2



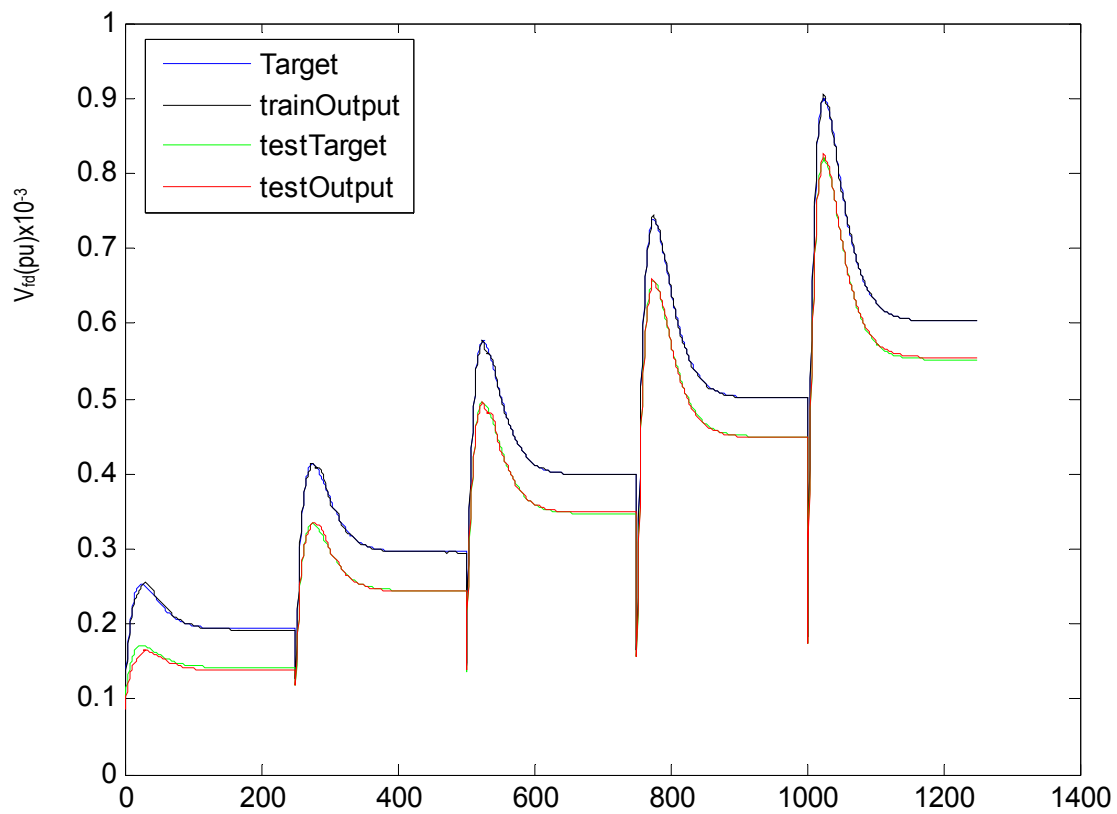
Σχ. 6.17 Έξοδος μοντέλου RBF (6x32x1), Σενάριο 3



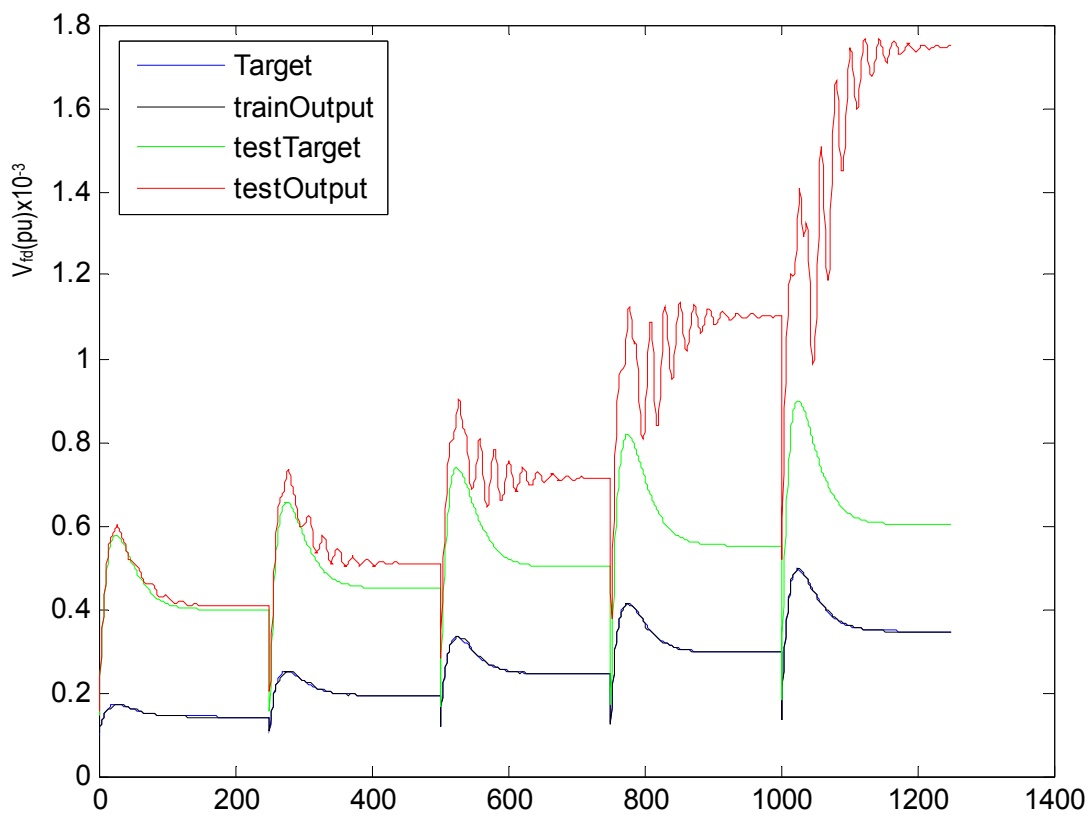
Σχ. 6.18 Εξόδος μοντέλου RBF (6x64x1), Σενάριο 2



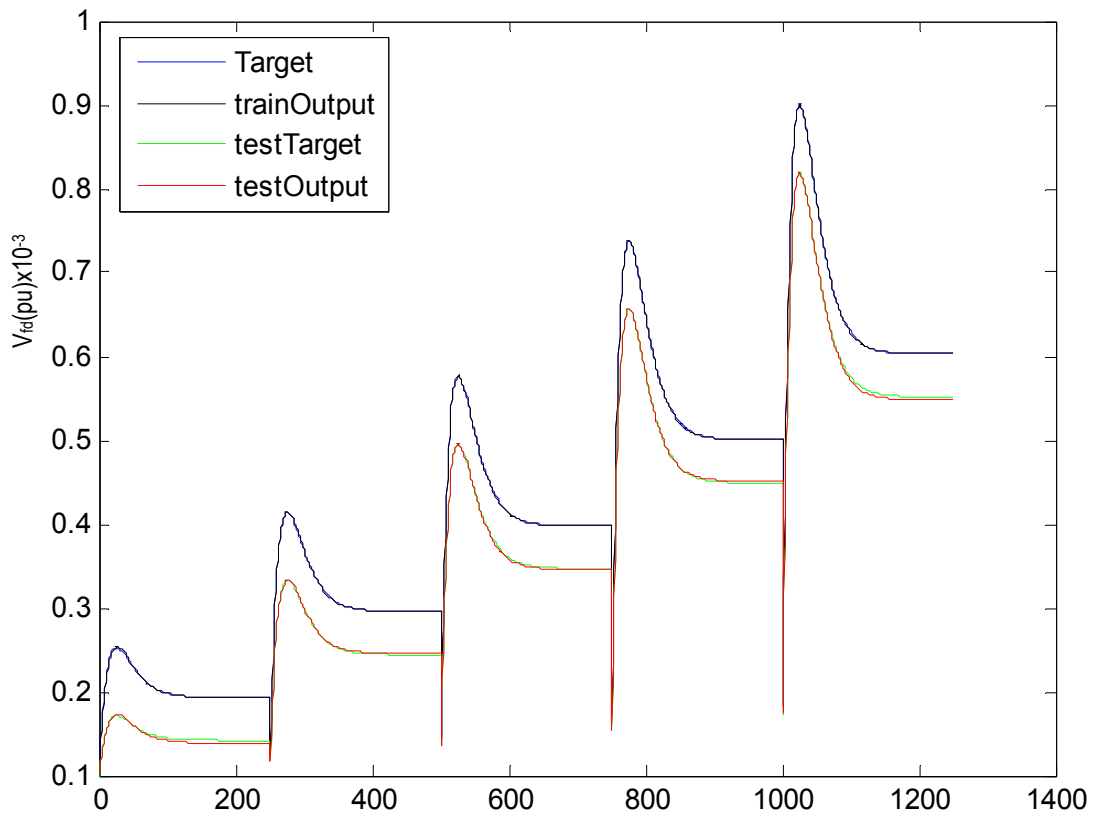
Σχ. 6.19 Εξόδος μοντέλου RBF (6x64x1), Σενάριο 3



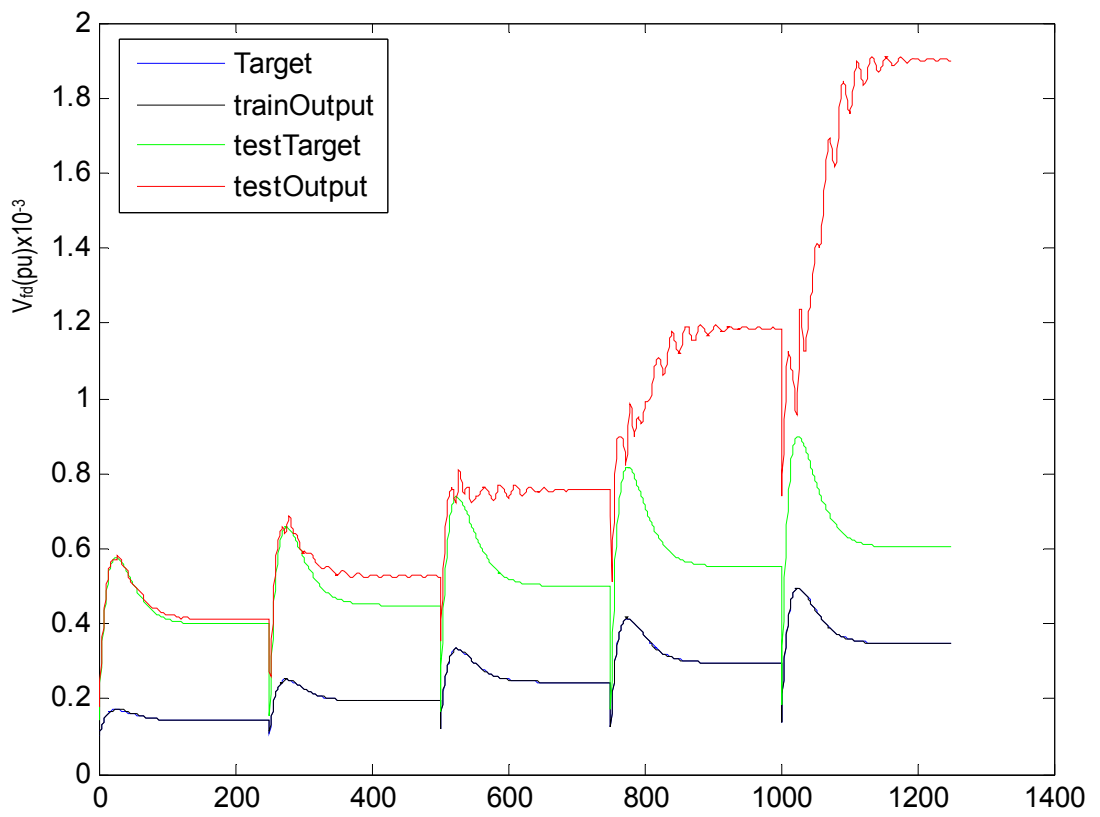
Σχ. 6.20 Έξοδος μοντέλου RBF (6x128x1), Σενάριο 2



Σχ. 6.21 Έξοδος μοντέλου RBF (6x128x1), Σενάριο 3

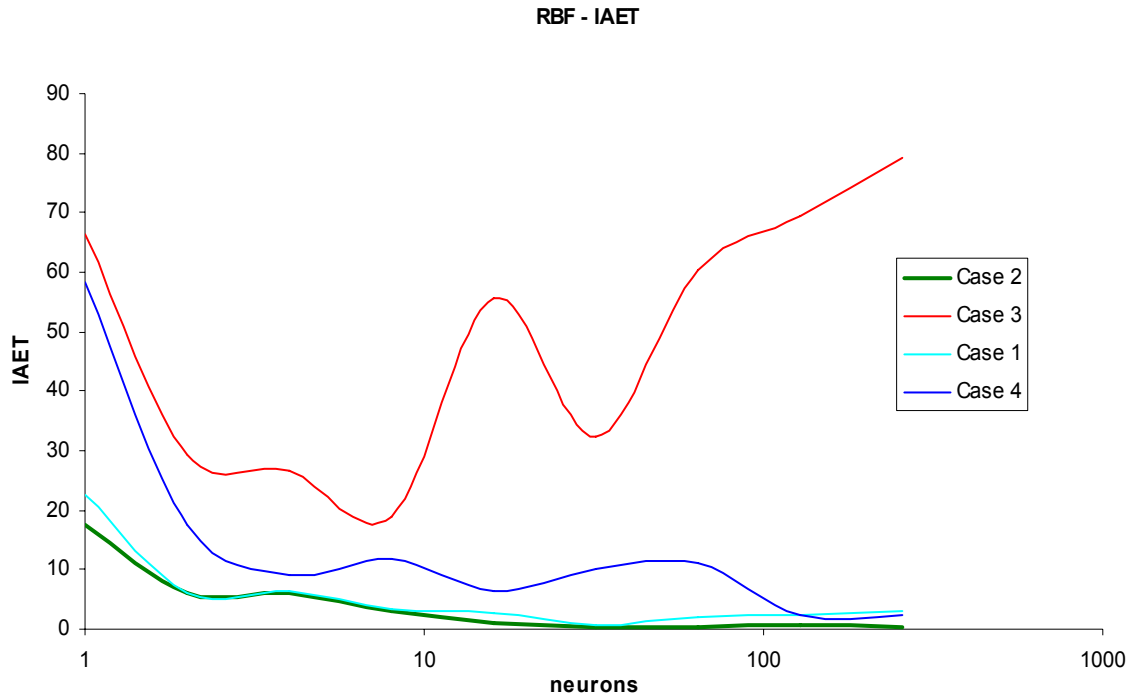


Σχ. 6.22 Έξοδος μοντέλου RBF (6x256x1), Σενάριο 2



Σχ. 6.23 Έξοδος μοντέλου RBF (6x256x1), Σενάριο 3

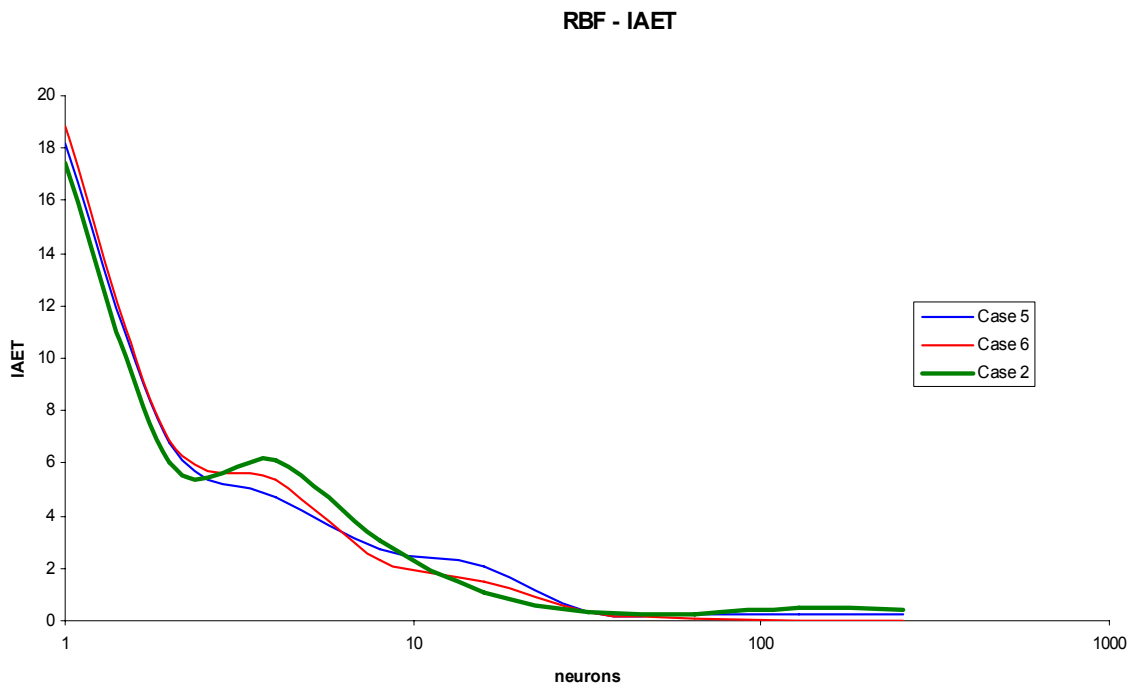
Παρόλο που και στα δύο Σενάρια πρόκειται για ακριβώς το ίδιο νευρωνικό δίκτυο βλέπουμε πόσο σημαντικό ρόλο παίζει η κατανομή των δεδομένων εκπαίδευσης (και επαλήθευσης, test). Στο Σενάριο 2 το μοντέλο παρουσιάζει συνεχή βελτίωση μέχρι να φτάσει τους 64 νευρώνες και μετά βλέπουμε να έχει μία μικρή πτώση στην απόδοση, λόγω υπερεκπαίδευσης. Το ίδιο ισχύει και στο Σενάριο 3, μόνο που σε καμία περίπτωση δεν μπορούμε να μιλήσουμε για ικανότητα γενίκευσης του μοντέλου, αφού σε κάθε περίπτωση το σφάλμα παραμένει υπερβολικά μεγάλο.



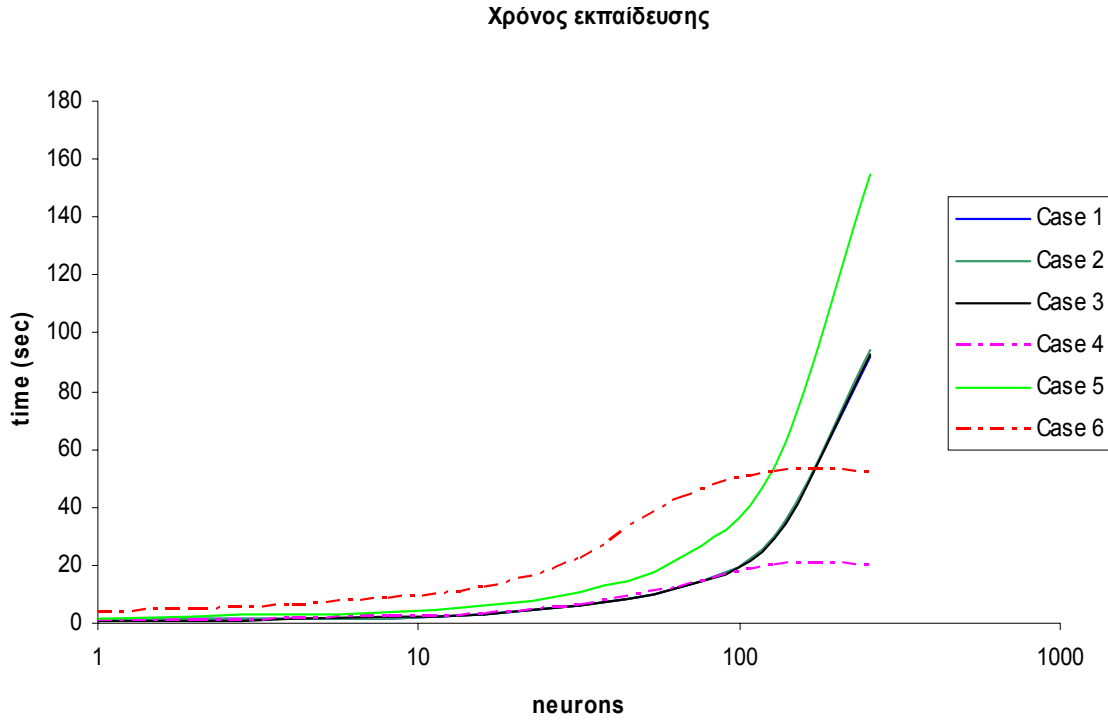
Σχ. 6.24 Κριτήριο απόδοσης J_1 του μοντέλου RBF(Σενάρια 1 – 4)

Όπως φαίνεται και στο

Σχ. 6.24 Το Σενάριο 2 είχε τα καλύτερα αποτελέσματα σε σχέση με τις υπόλοιπες περιπτώσεις με ίδια αναλογία γνωστών και αγνώστων στοιχείων, ενώ στο Σχ. 6.25 είναι φανερό ότι το Σενάριο 2 αποδίδει εξίσου καλά με τα Σενάρια όπου έχουμε γνωστό μεγαλύτερο ή και ολόκληρο το ποσοστό δειγμάτων εκπαίδευσης.



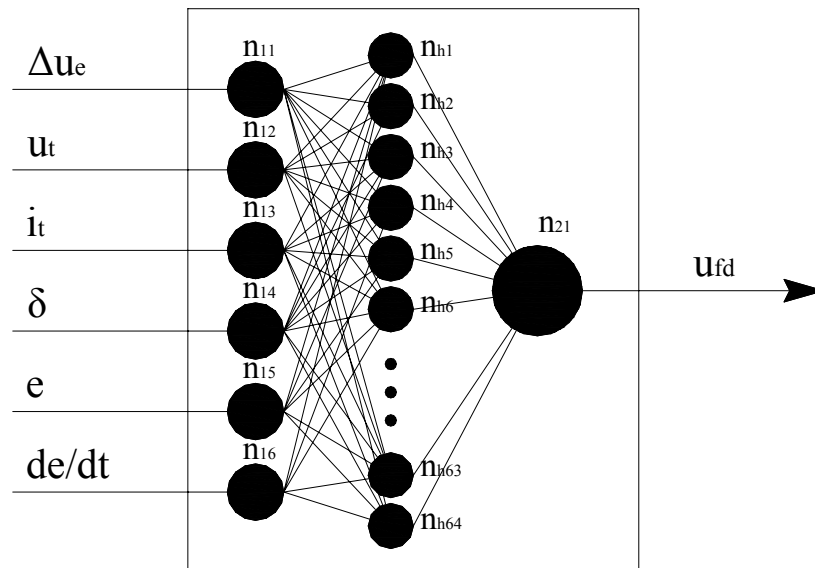
Σχ. 6.25 Κριτήριο απόδοσης J_1 του μοντέλου RBF (Σενάρια 2,5,6)



Σχ. 6.26 Χρόνος εκπαίδευσης μοντέλου RBF (Σενάρια 1 – 6)

Ανάλογη συμπεριφορά ή παραπλήσια παρουσιάζεται αν εξεταστεί το κριτήριο απόδοσης J_2 (δεν φαίνεται εδώ), ενώ ενδιαφέρον παρουσιάζει η μεταβολή του χρόνου εκπαίδευσης συναρτήσει του αριθμού των νευρώνων αλλά και του αριθμού των διανυσμάτων εκπαίδευσης, όπως φαίνεται στο Σχ. 6.26. Βλέπουμε πως στα Σενάρια 4 και 6 ο χρόνος εκπαίδευσης παραμένει σχεδόν σταθερός μετά τους 64 νευρώνες στο κρυφό επίπεδο. Αυτό οφείλεται στο γεγονός πως σε αυτό το σημείο το μοντέλο σταματάει την εκπαίδευση επειδή ικανοποιήθηκε η συνθήκη τερματισμού που σχετίζεται με την ανοχή του σφάλματος (error tolerance).

Έχοντας πλέον καταλήξει στο καλύτερο δυνατό μοντέλο νευρωνικού δικτύου θα μεταφέρουμε το μοντέλο σε κώδικα C και στην συνέχεια σε κώδικα ικανό να εκτελεστεί από μικροελεγκτή AVR. Το μοντέλο αυτό παρουσιάζεται με τις μεταβλητές εισόδου – εξόδου στο Σχ. 3.27 το οποίο διαθέτει συνολικά 448 βάρη.

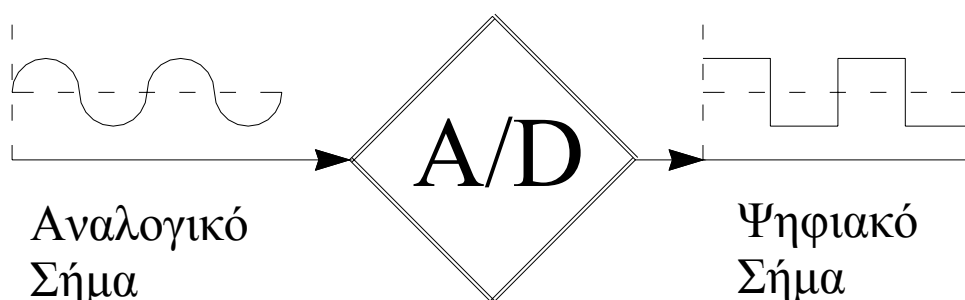


Σχ. 6.27 Μοντέλο T.N.Δ RBF προς υλοποίηση

6.4 Ολοκλήρωση Ευφυούς Ελεγκτή σε Πρακτικό Σύστημα Ελέγχου Διέγερσης.

6.4.1 Είσοδος Συστήματος

Για να έχουμε αλληλεπίδραση του συστήματος με το περιβάλλον χρειάζεται να υπάρχει και ο κατάλληλος τρόπος εισαγωγής του σήματος στον μικροελεγκτή. Ο μικροελεγκτής παρακολουθεί την διαδικασία υπό έλεγχο μέσω των δεδομένων που του μεταφέρονται, όπως τάση, ρεύμα, γωνία ισχύος κλπ. Τα μεγέθη αυτά είναι αναλογικά και παρουσιάζουν συνεχή ροή στο χρόνο, ενώ ο μικροελεγκτής μπορεί να επεξεργαστεί μόνο ψηφιακά δεδομένα. Εκ πρώτης όψεως θα φαινόταν λογικό να εισάγουμε ψηφιακά τα δεδομένα στον μικροελεγκτή, αλλά κάτι τέτοιο είναι αρκετά πολύπλοκο, καθώς απαιτούνται ειδικά ψηφιακά όργανα λήψης μετρήσεων. Επιπρόσθετα είναι δύσκολο να συντονιστεί η συχνότητα του ελεγκτή με την συχνότητα δειγματοληψίας των οργάνων. Από το δύσκολο αυτό έργο μας απαλλάσσει ο ψηφιοαναλογικός μετατροπέας (Σχ. 6.28) (A/D converter³¹) του μικροελεγκτή ο οποίος λαμβάνει το εξωτερικό αναλογικό σήμα και το μετατρέπει σε ψηφιακό ώστε να μπορεί να το επεξεργαστεί ο μικροελεγκτής.



Σχ. 6.28 Ψηφιο-Αναλογικός μετατροπέας (A/D analog to digital).

6.4.2 Συνδεσμολογία Συστήματος .

Όπως έχουμε ήδη αναφέρει η κεντρική μονάδα τους συστήματος μας αποτελείται από την αναπτυξιακή πλακέτα STK 500 και τον μικροελεγκτή AT mega16. Στο σύστημα συνδέονται οι κατάλληλοι αισθητήρες (π.χ. ταχογεννήτρια, κλπ) και κατόπιν επεξεργάζεται τα στοιχεία που λαμβάνει ώστε να διορθώσει την τάση διέγερσης της μηχανής αν αυτό απαιτείται. Ο μικροελεγκτής είναι αρχιτεκτονικής RISC και λειτουργεί σε συχνότητα 16MHz με τη χρήση εξωτερικής πηγής ρολογιού. Το διάγραμμα της συνδεσμολογία φαίνεται στο Σχ. 6.29 στο οποίο έχουμε σύστημα ελέγχου κλειστού βρόχου, δηλαδή η έξοδος ελέγχεται σε κάθε περίπτωση κατά πόσο ακολουθεί την είσοδο (επιθυμητή έξοδος). Στην είσοδο του ελεγκτή εκτός από το σφάλμα μεταφέρονται και άλλα δεδομένα σύμφωνα με το Σχ. 6.27. Ταυτόχρονα στην είσοδο του νευρωνικού δικτύου εισάγεται των παρών σφάλμα αλλά και η διαφορά του από το αμέσως προηγούμενο, έτσι έχουμε ΔV_{error} και με δεδομένο ότι

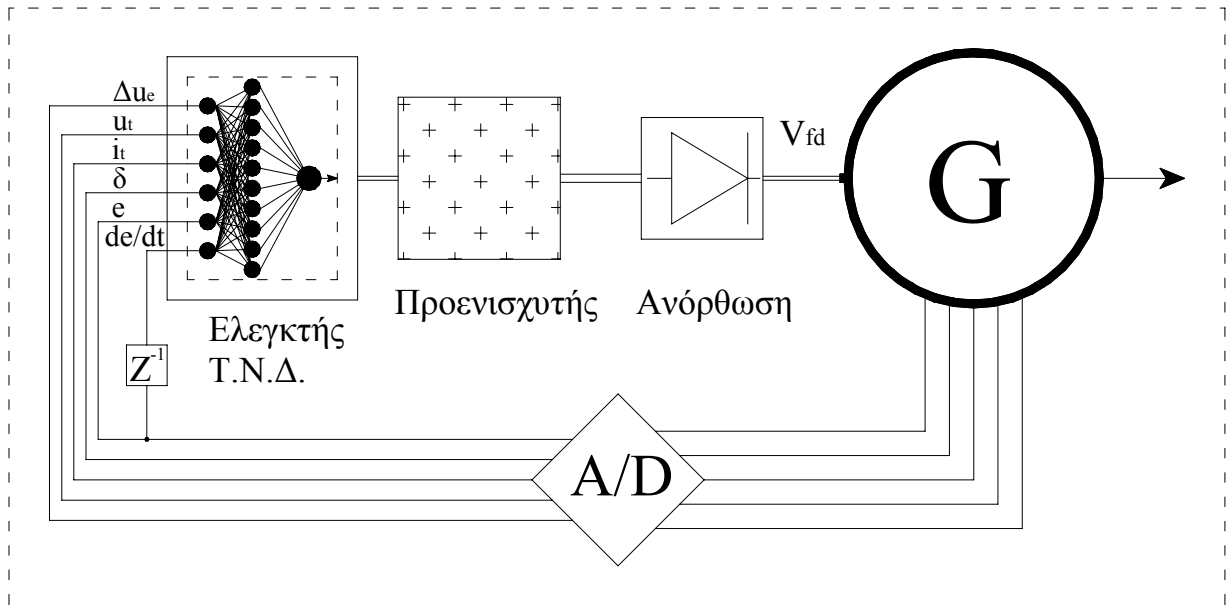
ο επεξεργαστής λειτουργεί σε συχνότητας της τάξης του 10^6 (MHz) ουσιαστικά ο λόγος $\frac{\Delta V_{error}}{\Delta t}$ γίνεται

$\frac{dV_{error}}{dt} \gg \frac{dV_e}{dt}$, οπότε ουσιαστικά έχουμε την παράγωγο του σφάλματος. Η παράγωγος του σφάλματος μπορεί να μας

δώσει πληροφορίες για το ρυθμό που μεταβάλλεται το σφάλμα, επειδή το πρόσημο της παραγώγου μας δείχνει αν έχει ανοδική ή καθοδική πορεία το διάνυσμα του σφάλματος. Ειδικά στη συγκεκριμένη περίπτωση, σχεδόν σε όλα τα ηλεκτρικά σφάλματα υπάρχουν σημεία όπου το απόλυτο σφάλμα είναι το ίδιο αλλά να μπορεί διαφέρει η παράγωγος. Έτσι σε κάθε

³¹ Τα χαρακτηριστικά του αναφέρονται στο κεφάλαιο 4.

περίπτωση η ενέργεια για τη διόρθωση διαφέρει ριζικά. Για να αντιληφθούμε καλύτερα την σημασία της παραγώγου σε ένα ηλεκτρικό μέγεθος πρέπει να λάβουμε υπόψη πως διηλεκτρικά υλικά, και κυρίως τα ημιαγώγιμα στοιχεία, μπορεί να διασπαστούν σε σημαντικά χαμηλότερες τάσης (ή ρεύμα) από την μέγιστη δυνατή όταν η παράγωγος γίνει πολύ μεγάλη (ένας από τους μη επιθυμητούς παράγοντες που μπορεί να αρχίσει να άγει ένα θυρίστορ είναι ο λόγος $\frac{dV}{dt}$ ή $\frac{dI}{dt}$). Στην προκειμένη περίπτωση όπου τα παραπάνω δεδομένα συνδυάζονται με ένα νευρωνικό δίκτυο γίνεται πρόβλεψη για την πορεία που θα έχει το σφάλμα από τα πρώτα στάδια και αντιμετωπίζεται ανάλογα.



Σχ. 6.29 Διάγραμμα Συστήματος Ελέγχου

6.4.3 Αλγόριθμος Μοντέλου .

Ο αλγόριθμος του συστήματος ελέγχου σε γενικές γραμμές θα μπορούσε να χαρακτηριστεί ως αλγόριθμος ελέγχου συστήματος κλειστού βρόχου. Παρακάτω αναφέρονται τα βασικά βήματα στα οποία μπορεί να συνοψιστεί ο αλγόριθμος.

- Βήμα 1. Λήψη μετρήσεων από την σύγχρονη μηχανή.
- Βήμα 2. Διαβίβαση και Ψηφιοποίηση των δεδομένων στον μικροελεγκτή.
- Βήμα 3. Σύγκριση παρούσας κατάστασης με την επιθυμητή.
- Βήμα 4. Σύγκριση σφάλματος με το αμέσως προηγούμενο – αποθήκευση παρόντος.
- Βήμα 5. Κανονικοποίηση δεδομένων.
- Βήμα 6. Εισαγωγή κανονικοποιημένων δεδομένων στο νευρωνικό δίκτυο.
- Βήμα 7. Επεξεργασία δεδομένων από το νευρωνικό δίκτυο
- Βήμα 8. Εξαγωγή δεδομένων – «Λήψη» απόφασης για κατάλληλη ενέργεια.
- Βήμα 9. Αποκανονικοποίηση (επαναφορά) και ενίσχυση(αν κριθεί απαραίτητο) εξόδου νευρωνικού δικτύου.
- Βήμα 10. Μεταφορά σήματος εξόδου στο κύκλωμα ελέγχου της διέγερσης και επιστροφή στο βήμα 1.

6.4.4 Κώδικας C Μοντέλου RBF.

Ο κώδικας του μοντέλου είναι σε γλώσσα C και εκτελείται σε έναν κανονικό Η/Υ. Για να εκτελεστεί το μοντέλο σε μικροελεγκτή χρειάζεται να γίνουν τροποποιήσεις στη δομή του κώδικα, καθώς εκτός του ότι έχουμε διαφορές στις βιβλιοθήκες, είναι δύσκολο να προγραμματίσουμε τον μικροελεγκτή να διαβάζει τιμές από κάποια μονάδα μνήμης, έτσι ο κώδικας τροποποιείται, όπως φαίνεται και στο Παράρτημα Β, όπου έχουμε το συνολικό κώδικα της παρούσας πτυχιακής εργασίας, και οι τιμές των βαρών και των άλλων προκαθορισμένων στοιχείων καταχωρούνται ως σφαιρικές μεταβλητές. Αυτό αποβαίνει εις βάρος του μεγέθους του τελικού αρχείου, αλλά μας διευκολύνει σημαντικά στον προγραμματισμό του μικροελεγκτή.

```
#include<stdio.h>
#include<math.h>
#include<memory.h>

#define neurons_InputLayer 6
/* Neyrones lou epipedou(eisodos)*/

#define neurons_HiddenLayer 64
/* Neyrones kryfou epipedou*/

#define neurons_OutputLayer 1
/* Neyrones epipedou eksodou*/

double InputHidden_Weights[neurons_HiddenLayer][neurons_InputLayer];
/*Barh Eisodou - Kryfou(IW)*/

double HiddenOutput_Weights[neurons_HiddenLayer][neurons_OutputLayer];
/*Barh Kryfou-Eksodou(LW)*/

double biasHiddenLayer[neurons_HiddenLayer];
double biasOutputLayer[neurons_OutputLayer];
/* neyronon epipedou eksodou*/

int i,j;

void read_InputHidden_Weights(const char *file_InputHidden_Weights)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i,j;
    f=fopen(file_InputHidden_Weights,"r");
    for(i=0;i<neurons_HiddenLayer;i++)
```

```

        for(j=0;j<neurons_InputLayer;j++)
            fscanf(f,"%lf",&InputHidden_Weights[i][j]);
    }
void read_HiddenOutput_Weights(const char *file_HiddenOutput_Weights)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i,j;
    f=fopen(file_HiddenOutput_Weights,"r");
    for(i=0;i<neurons_HiddenLayer;i++)
        for(j=0;j<neurons_OutputLayer;j++)
            fscanf(f,"%lf",&HiddenOutput_Weights[i][j]);
}
void read_biasAll(const char *file_bias)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i;
    f=fopen(file_bias,"r");
    for(i=0;i<neurons_HiddenLayer;i++)
        fscanf(f,"%lf",&biasHiddenLayer[i]);

    for(i=0;i<neurons_OutputLayer;i++)
        fscanf(f,"%lf",&biasOutputLayer[i]);
}
double tansig(double x)
{
    double N;
    N = 2/(double)(1+exp(-2*x))-1;
    return N;
}

double radbas(double x)
{
    return (double)exp(-x*x);
}

double purelin(double x)
{
    return x;
}

```

```

main()
{
    int counter=0;
    /*EPIPEDO EISODOU (1o)*/
    FILE *in,*out;
    in = fopen("input.txt","r");
    out = fopen("output.txt","w");

    read_InputHidden_Weights("IW2.txt");
    read_HiddenOutput_Weights("LW2.txt");
    read_biasAll("bias.txt");

    //fprintf(out,"\tEisodos \t\t\t\t\tEksodos \n");
    //fprintf(out,"input1\tinput2\tinput3\tinput4\t\tinput5\t\tinput6\n");
    printf("\tEisodos \t\t\t\t\tEksodos \n");
    printf("input1\tinput2\tinput3\tinput4\t\tinput5\t\tinput6\n");

    while(1)
    {
        int i,j,flag;
        /*dhlosh metavlhton*/
        double input[neurons_InputLayer];
        /*eisodou*/
        double input_HiddenLayer[neurons_HiddenLayer];
        /*eisodou kryfou epipedou*/
        double tempOut_HiddenLayer[neurons_HiddenLayer];
        /*eksodou kryfou epipedou*/
        double input_OutputLayer[neurons_OutputLayer];
        /*eisodou epipedou eksodou*/
        double Output[neurons_OutputLayer];
        /*eksodou*/

        memset(input_HiddenLayer,0,sizeof(input_HiddenLayer));
        memset(input_OutputLayer,0,sizeof(input_OutputLayer));
        /*ton metavlhton-"ekatharish"*/

        flag = 0;
        /*Eisagogh dedomenon (EISODOs) */
        for(i=0;i<neurons_InputLayer;i++)
            if(fscanf(in,"%lf",&input[i])==EOF) flag = 1;
        if(flag) break;
    }
}

```

```

    /***PROTO KRYFO EPIPEDO ***/
    for(i=0;i<neurons_HiddenLayer;i++)
    {
    /********/
    /* Eykleidia apostash eisodoy me ta barh */
        for(j=0;j<neurons_InputLayer;j++)
        {
            double diff = (double) (input[j]-InputHidden_Weights[i][j]);
            input_HiddenLayer[i] += (double)diff*diff;
        }
        input_HiddenLayer[i] = (double) sqrt(input_HiddenLayer[i]);
    /********/

tempOut_HiddenLayer[i]=radbas(input_HiddenLayer[i]+biasHiddenLayer[i]);
/*Klhsh tis synarthshs(activation function) */
    }

    /*** EPIPEDO EKSODOU ***/
    for(i=0;i<neurons_OutputLayer;i++)
    {
    double temp = 0;
        for(j=0;j<neurons_HiddenLayer;j++)
        {
            temp+=(double) (tempOut_HiddenLayer[j]*HiddenOutput_Weights[i][j]);
            /* polapllasiasmos timwn me ta varh kai ypologismos
athroismatos*/

input_OutputLayer[i]+=(double) (tempOut_HiddenLayer[j]*HiddenOutput_Weights[i][j]);
        }
        Output[i]=purelin(input_OutputLayer[i]+biasOutputLayer[i]);
/*Klhsh tis synarthseis (activation function)*/
    }

    for(i=0;i<neurons_InputLayer;i++)
    /*Katagrfh apotelesmaton se arxeio kai sthn Othonh*/
    {
        //fprintf(out,"%2.2f\t",input[i]);
    /* emfanish Dedomenon eisodou*/
printf ("%2.2f\t",input[i]);
    }
    for(j=0;j<neurons_OutputLayer;j++)

```

```
        {
            //fprintf(out, "\t%2.2f", Output[j]);
/*Apotelesmata - Eksosdos*/
            fprintf(out, "%2.2f", Output[j]);
            printf("\t%2.3f", Output[j]);
        }
        fprintf(out, "\n"); printf("\n");
        counter++;
    }

    printf("\nΠιεστε ENTER gia telos.");
    getchar();

return 0;
```

7 ΣΥΜΠΕΡΑΣΜΑΤΑ

Ολοκληρώνοντας την παρούσα πτυχιακή εργασία, οι οποία περιλαμβάνει θεωρητική μελέτη των νευρωνικών δικτύων και των συστημάτων ελέγχου μιας σύγχρονης μηχανής, αλλά και πρακτική εφαρμογή αυτών, καταλήγουμε σε πολύ χρήσιμα συμπεράσματα σε ότι αφορά την χρήση των νευρωνικών δικτύων και την αντικατάσταση των συμβατικών διατάξεων ελέγχου. Γενικά στα συμπεράσματα στα οποία καταλήγουμε είναι τα αναμενόμενα, αλλά αξίζει να σταθούμε σε μερικά από αυτά, τα οποία παρατίθενται με την ίδια σειρά των πειραμάτων που διεξήχθησαν.

Ήδη από το δεύτερο Κεφάλαιο αρχίζουν να διαφαίνονται οι δυσκολίες της μελέτης καθώς καλούμαστε να κάνουμε μαθηματική μοντελοποίηση μιας σύγχρονης ηλεκτρικής μηχανής. Μια πραγματική ηλεκτρική μηχανή είναι εξαιρετικά πολύπλοκη και θα χρειαζόταν μια εργασία αποκλειστικά για την ανάλυση της αν δεν κάναμε κάποιες βασικές παραδοχές, χωρίς ωστόσο να ξεφύγουμε από τα πραγματικά χαρακτηριστικά μιας σύγχρονης ηλεκτρικής μηχανής. Έτσι αποφεύγουμε να αναφερθούμε σε ευρέως κοινά στοιχεία για τις ηλεκτρικές μηχανές όπως επίσης θεωρούμε αμελητέες τις ατέλειες – ασυμμετρίες λόγω υλικών (§ 2.2) κλπ. Παρόλα αυτά όμως χρειάζονται δεκάδες εξισώσεις για να αναπτυχθεί το μαθηματικό μοντέλο με αποτέλεσμα να μην απέχει σημαντικά από το πραγματικό και τα σφάλματα που εισάγονται είναι αμελητέα.

Όπως ήδη φάνηκε από τα προηγούμενα Κεφάλαια η εκπαίδευση ενός νευρωνικού δικτύου αποτελεί ένα από τα σημαντικότερα και χρονοβόρα στάδια ανάπτυξης ενός ευφυούς συστήματος ελέγχου. Η εκπαίδευση είναι ο καταλυτικός παράγοντας για την ικανότητα γενίκευσης του μοντέλου και μπορεί να χρειαστεί να επαναληφθεί πολλές φορές έως ώτου επιτευχθεί το επιθυμητό αποτέλεσμα. Μάλιστα είδαμε πως η υπερβολική εκπαίδευση μόνο θετικά αποτελέσματα δεν έχει, καθώς το μοντέλο υπερεκπαιδεύεται και οδηγείται σε απόκλιση από το στόχο. Για το λόγο αυτό επαναλάβαμε εκατοντάδες φορές τη διαδικασία αυτή μέχρι που επιτευχθεί το επιθυμητό αποτέλεσμα. Επίσης η εκπαίδευση μπορεί να επηρεαστεί από τον αριθμό των διανυσμάτων εκπαίδευσης σε σχέση με τα άγνωστα διανύσματα, αλλά και τους νευρώνες του μοντέλου, καθώς και από την κατανομή των τιμών που έχουμε όσων αφορά το σύνολο του χώρου των διανυσμάτων. Έτσι κατά την εκπαίδευση πρέπει να επιλέγουμε αντιπροσωπευτικό δείγμα διανυσμάτων εκπαίδευσης όσον αφορά τα άγνωστα δεδομένα (η αναλογία πρέπει να είναι περίπου 70% γνωστά και 30% άγνωστα). Εξαιρετικά σημαντικός παράγοντας είναι και η αναλογία των διανυσμάτων εκπαίδευσης προς τον αριθμό των νευρώνων του δικτύου. Σε περίπτωση που ο αριθμός διανυσμάτων δεν είναι σημαντικά μεγαλύτερος από τους νευρώνες του μοντέλου τότε το μοντέλο «μαθαίνει» μόνο το γνωστά παραδείγματα και αδυνατεί να ανταποκριθεί σε άγνωστα δεδομένα. Τέλος δεν πρέπει να ξεχνάμε πως η χρήση της ορμής (momentum term) στην εκπαίδευση ενός νευρωνικού δικτύου μειώνει κατά πολύ τον χρόνο εκπαίδευσης και ταυτόχρονα οδηγεί σε καλύτερα αποτελέσματα (αποφεύγονται τα τοπικά ελάχιστα). Κρίσιμο ρόλο στην εκπαίδευση του μοντέλου έπαιξε και η κανονικοποίηση των τιμών, καθώς είχαν διαφορά τάξης μεγέθους μεταξύ τους χωρίς όμως αυτό να ανταποκρίνεται στο πραγματικό ποσοστό που επηρέαζαν το σύστημα, αλλά οφείλεται στη διαφορετική κλίμακα που μετρούνται τα μεγέθη (το πρόβλημα αυτό βέβαια εξαλείφθηκε με την κανονικοποίηση των τιμών).

Επίσης στην παρούσα μελέτη είδαμε ένα νευρωνικό δίκτυο να λειτουργεί ως μία πύλη XOR, ακόμη και για ενδιάμεσες τιμές, να ταξινομεί τρία διαφορετικά είδη λουλουδιών με μεγάλη επιτυχία, βάση σχετικών ιδιοτήτων, όπου με συμβατικές μεθόδους θα ήταν εξαιρετικά χρονοβόρα. Στο τέλος καταφέραμε να χρησιμοποιήσουμε ένα νευρωνικό δίκτυο για τον έλεγχο μιας σύγχρονης ηλεκτρικής γεννήτριας, όπου ένας μικροελεγκτής προγραμματισμένος με ένα μοντέλο νευρωνικού δικτύου RBF αντικατέστησε ένα (πανάκριβο) συμβατικό σύστημα. Διαπιστώνουμε πως κάθε πρόβλημα μπορεί να επιλυθεί (προσεγγιστεί) με χρήση των νευρωνικών δικτύων, χρησιμοποιώντας το κατάλληλο μοντέλο πάντα βέβαια, και στις περισσότερες περιπτώσεις τα αποτελέσματα είναι καλύτερα από αυτά των συμβατικών μεθόδων. Επίσης πρέπει να τονίσουμε πως η εφαρμογή ελεγκτών με νευρωνικά δίκτυα κατέστη δυνατή χάρη στην εξέλιξη των μικροελεγκτών –

μικροεπεξεργαστών αλλά και των ηλεκτρονικών ισχύος, καθώς ένας μικροελεγκτής όσο κατάλληλα προγραμματισμένος και να είναι δεν μπορεί σε καμία περίπτωση να οδηγήσει το φορτίο ισχύος χωρίς τα απαραίτητα ενδιάμεσα στοιχεία.

Παρόλο που εκ πρώτης όψεως φαινόταν δύσκολο και ίσως περιπτώ να προσπαθήσει κανείς να επιλύσει ένα κοινό πρόβλημα, όπως η λειτουργία μίας πύλης XOR, με τη χρήση νευρωνικού δικτύου, καθώς με συμβατικό κώδικα το πρόβλημα επιλύεται χωρίς ειδικές γνώσεις προγραμματισμού και σε πολύ πιο συμπυκνόμενο κώδικα. Όταν όμως προχωρήσουμε σε πιο σύνθετα πρόβλημα όπου είναι εξαιρετικά δύσκολο να επιλυθούν με κλασικές μεθόδους και πολλές φορές αδύνατο, πχ ταξινόμηση, προσέγγιση μιας συνάρτησης κλπ, διαπιστώνουμε πως μοντέλο ίδιου μεγέθους με αυτό που επιλύει το απλό πρόβλημα της πύλης XOR κάνει εξίσου καλά τη δουλειά και στα σύνθετα προβλήματα χωρίς μεταβολές στη δομή του, παρά μόνο με διαφορετική εκπαίδευση (ουσιαστικά μεταβάλλουμε τα βάρη).

Η ανάπτυξη ενός ευφυούς συστήματος, αν και παραμένει μία αρκετά δύσκολη υπόθεση, έχει απλοποιηθεί σε μεγάλο βαθμό χάρη στην εξέλιξη της τεχνολογίας. Σήμερα με την ισχύ που προσφέρουν οι ηλεκτρονικοί υπολογιστές και εξοπλισμένοι με το κατάλληλο λογισμικό μπορούν να μετατραπούν σε ένα εξαιρετικά χρήσιμο εργαλείο. Όπως είδαμε και στο Κεφάλαιο 5, το σημαντικότερο μέρος της εργασίας έγινε με τη βοήθεια Η/Υ και του MATLAB. Δεν είναι τυχαίο ότι επελέγη το MATLAB, καθώς αν εξαιρέσει κανείς τις ιδιαίτερα υψηλές απαιτήσεις που έχει από τον χρήστη και τον Η/Υ, αποτελεί ένα από τα πλέον πλήρη «πακέτα» στο χώρο. Φυσικά δεν είναι το μόνο περιβάλλον που μπορεί να εργαστεί κανείς στον συγκεκριμένο τομέα, καθώς υπάρχουν και άλλα προγράμματα που παρέχουν παρόμοιες λειτουργίες, π.χ. Neuro solution – Neuro dimension κλπ, επίσης μπορεί να δοκιμάσει να αναπτύξει το μοντέλο απευθείας σε μία γλώσσα προγραμματισμού, πχ C, αλλά σε αυτή την περίπτωση χρειάζονται πού εξειδικευμένες γνώσεις μαθηματικών και προγραμματισμού.

ΠΑΡΑΡΤΗΜΑ Α: ΝΟΜΟΙ ΚΑΙ ΘΕΩΡΗΜΑΤΑ Τ.Ν.Δ.

Νόμος Hebb: Αν ένα επεξεργαστικό στοιχείο λαμβάνει στην είσοδο την έξοδο ενός άλλου επεξεργαστικού στοιχείου, και έχουν το ίδιο μαθηματικό πρόσημο, τότε το βάρος μεταξύ των επεξεργαστικών στοιχείων πρέπει να ενισχυθεί.

Νόμος Hopfield: είναι παρόμοιος με τον νόμο του Hebb με τη διαφορά ότι καθορίζει το πλάτος της ενίσχυσης ή της εξασθένησης. Δηλώνει ότι "αν η επιθυμητή έξοδος και η είσοδος είναι και οι δύο ενεργές ή και οι δύο ανενεργές, αύξησε το βάρος διασύνδεσης κατά (ποσό ίσο με) τον ρυθμό εκμάθησης, αλλιώς μείωσε το βάρος κατά (ποσό ίσο με) τον ρυθμό εκμάθησης.

Θεώρημα σύγκλισης Perceptron ή Θεώρημα Rosenblatt: «Αν δοθεί ένα διάνυσμα εισόδου και ένα αντίστοιχο διάνυσμα επιθυμητών εξόδων, τότε υπάρχει κάποιος τρόπος εκπαίδευσης του Τ.Ν.Δ. για να επιτευχθεί η επιθυμητή έξοδος, αν (και μόνο αν) υπάρχουν οι κατάλληλες τιμές για τα βάρη».

ΠΑΡΑΡΤΗΜΑ Β: ΚΩΔΙΚΑΣ

B.1. Κώδικας Matlab (Matlab script)

B.1.1 Πρόβλημα "IRIS" – Μοντέλο MLP.

```
%Evresh tou kalyterou modelou epilyshs tou provlhmatos katanomhs %"IRIS"

testMyNet1H; %Dokimh modelon me ena kryfo (1H)
             %epipedo
testMyNet2H; %Dokimh modelon me dyo kryfa (2H)
             %epipeda
findBestNN(Succes1H,Succes2H) %Eyresh modelou me to kalytero
                             %apotelesma

%Dokimh gia thn eyresh tou kalyterou neyrokiou diktyou

point=['a':'j']; %deiktes "run"
model=1; %deikths modelou
Succes1H=zeros(15,10); %pinakas
                             %apotelesmaton

for x=2:16
    for run=1:10
        mkdir(['4_' int2str( x) '_1' point(run)]);%dhmiourgia fakelou
                                                %gia kathe dokimh
        w=['4_' int2str( x) '_1' point(run)]; %onoma fakelou -
                                                %modelo
        tic; %arxh xronou
        MyNet1HRdm; %dhmiourgia-dokimh-
                                                %sygkrish diktyon
        time1H(model,run)=toc; %metrhsh xronou
                                                %"ekpaideyshs" tou
                                                %diktyou
        Succes1H(model,run)=epityxia; %kataxorhsh
                                                %apotelesmatons
                                                %ston pinaka
    end
    model=model+1;
end
```

```

xlswrite('Time1H.xls',time1H); %apothhkeysei
                                %xronon
                                %ekpaideyshs
xlswrite('succes1H.xls',Succes1H); %apothhkeysei
                                %apotelesmaton olon
                                %ton dokimon

%Dhmiourgia modelou Neyrnikou diktyou(me 1 kryfo epipedo,x neyronon)-
%ekpaideysh - dokimh

load irisData; %Eisagogh
                %dedomenon(ekpaideysh -
                %dokimhs)
RandomValues; %Tyxaia katanomh
                %dedomenon,hia kalyterh
                %ekpaideysh

                %dhmiourgia modelou(feed-
                %forward backpropagation)
iris=newff([43 79;20 44;10 69;1 25],[x 1],{'tansig','tansig'},'traingdx');
                %x: neyrones protou kryfou
                %epipedou
                %tansig:mh
                %grammikh (sigmoid)
                %synarthsh energopoihseis
                %mono sto proto epipedo
                %exoume gramikous neyrones
                %traingdx: synarthsh
                %ekpaideyshs N.D.
                %traingdx:Gradient descent
                %with momentum and adaptive
                %learning rate
                %backpropagation

                %Parametroi ekpaideyshs,oti
                %den anaferetai exei
                %proepilegmenes times apo
                %to MatLab
                %(iris.trainParam)
iris.trainParam.epochs = 500; %megistos arithos
                                %epoxon(epochs)
iris.trainParam.goal = 0; %stoxos,elaxisto sfalma
                                %MSE(Performance goal)

```

```

iris.trainParam.lr = 0.025; %rythmos ekmathieis
                               %(Learning rate)
                               %to diktyo xrhsimopoiei
                               %metavlhto Learning rate
iris.trainParam.lr_inc = 1.05; %logos aykshshs learning
                               %rate
iris.trainParam.lr_dec = 0.8; %logos meioshs learning
                               %rate
iris.trainParam.max_fail = 5; %Maximum validation
                               %failures
iris.trainParam.max_perf_inc = 1.04; %Maximum performance
                               %increase
iris.trainParam.mc = 0.75; %Momentum constant.
iris.trainParam.min_grad = 1e-30; %Minimum performance
                               %gradient
iris.trainParam.show = 25; %Epoxes gia emfanish
                               %apotelesmaton
iris.trainParam.time = inf; %Megistos xronos
                               %ekpaideyshs

IRIS=train(iris,randomInput,randomTarget);%ekpaideysh tou diktyou
cd(w);
hgsave('train','-v6'); %apothhkeysh grafhmatos
                               %ekpadeyshs

close;
cd ..;

out_train=sim(IRIS,input); %dokimh diktyou se gnosta
                               %stoixeia
out_test=sim(IRIS,test); %dokimh diktyou se agnosta
                               %stoixeia

figure;
plot('v6',target2);hold on; %grafhma stoxou
plot('v6',out_train,'r+'); %apotelesmatos gnoston
                               %stoixeion
plot('v6',out_test,'go'); %apotelesmatps agnostos
                               %stoixeion

h=legend('Target','Output train','Output test',4);%legend
cd(w);
hgsave('output','-v6'); %apothhkeysh grafhmatos
                               %dokimon

close;
cd ..;

```

```

epityxia = ErrorNet(out_test,target2);    %ypologimsos epityxias -
                                           %sfalma

cd(w);
outFiles;                                %apothhkeysh apotelesmaton
                                           %kai stoixeion diktyou

cd ..;

%Katanemei me tyxaio tropo ta stoixeia ton pinakon,oi 2 pinakes %prepei na
exoun to idio mhkos
i=0;
j=1;
flag=1;
randomValues=[0];                        %tyxaies theseis
X=length(input);                         %mhkos pinaka
while(flag == 1)
    i=i+1;
    flag2=1;
    rdm=round(75*rand(1)+0.5);            %synarthsh - gennhtria
                                           %tyxaion akeraion
                                           %arthmon,mh mhdenikoi
    L = length(randomValues);             %arithmos tyxaion theseon
    for k=1:L                             %elegxos theseis pinaka
        if rdm==randomValues(k) flag2 = 0;%an exei hdh
                                           %xrhgsimopoihthei
        end;
    end;
    if flag2 == 1                         %kataxorhsh tyxaias theseis
                                           %ston pinaka
        randomInput(1:4,j)=input(1:4,rdm);%efoson den exei hdh
                                           %xthsimoipoihei
        randomTarget(j)=target2(rdm);
        j=j+1;
        Order(j)=rdm;                    %Telikh tyxaia katanomh
    end;
    randomValues(i)=rdm;                 %kataxorhsh teleytaias
                                           %tyxaias timhs

    if j>X flag = 0;
    end;
end;

%Synarthsh pologismou sfalantos dyo pinakon
%oi pinakes prepei na einai:A = pragmatikes times , B= epithymhtes %times

```

```

function E=ErrorNet (A,B)

x=length (B) ; %mhkos pinaka
rndA=round (A) ; %stroggylopoihs pragmatikon timon
dAB=abs (rndA-B) ; %diafora - mhdenismos mh
%lanthasmenon theseon
errors=find (dAB) ; %epistrofh mh mhdenikon
%heseon(lathos theshs)
y=length (errors) ; %arithmos lathos theseon
dSP=y/x; %sxetiko sfalma theshs

SM=sum (abs (B) ) ; %oliko athroisma epithymhton timon
mo=SM/x; %messh epithymhth timon
dif=abs (A-B) ; %apolyto sfalma ana stoixeio
sm=sum (dif) ; %athroisma sfalmaton
mean=sm/x; %meso sfalma ana apotelesma
dSV = abs (mean/mo) ; %sxetiko sfalma timon
if dSV>1 dSV=1; %megistos vathmos epirohs telikou
%apotelesmatos

end;

n=(dSP)*0.8+(dSV)*0.2; %oliko sxetiko sfalma
E=(1-n)*100; %pososto epityxias
sprintf('epityxia = %2.2f%%',E) %emfanish apotelesmatos

%Dhmiourgia modelou Neyrnikou diktyou(me 2 kryfa epipeda,x,y %neuronon)-
ekpaideysh - dokimh

load irisData; %Eisagogh
%dedomenon(ekpaideysh -
%dokimhs)
RandomValues; %Tyxaia katanomh
%dedomenon,hia kalyterh
%ekpaideysh

%dhmiourgia modelou(feed-
%forward backpropagation)
iris=newff([43 79;20 44;10 69;1 25],[x y
1],{'tansig','tansig','tansig'},'traingdx');
%x: neyrones protou kryfou
%epipedou

```

```

        %y: neyrones deyterou
        %kryfou epipedou
        %tansig:mh
        %grammikh (sigmoooid)
        %synarthsh energopoihseis
        %mono sto proto epipedo
        %exoume gramikous neyrones
        %traingdx: synarthsh
        %ekpaideyshs T.N.D.
        %traingdx:Gradient descent
        %with momentum and adaptive
        %learning rate
        %backpropagation

        %Parametroi ekpaideyshs,oti
        %den anaferetai exei
        %proepilegmenes times apo
        %to MatLab

iris.trainParam.epochs = 500;        %megistos arithos
                                     %epoxon(epochs)
iris.trainParam.goal = 0;            %stoxos,elaxisto sfalma
                                     %MSE(Performance goal)
iris.trainParam.lr = 0.025;         %rythmos ekmathieis
                                     % (Learning rate)
                                     %to diktyo xrhsimopoei
                                     %metavlhto Learning rate
iris.trainParam.lr_inc = 1.05;      %logos aykshshs learning
                                     %rate
iris.trainParam.lr_dec = 0.8;       %logos meioshs learning
                                     %rate
iris.trainParam.max_fail = 5;       %Maximum validation
                                     %failures
iris.trainParam.max_perf_inc = 1.04; %Maximum performance
                                     %increase
iris.trainParam.mc = 0.75;          %Momentum constant.
iris.trainParam.min_grad = 1e-30;   %Minimum performance
                                     %gradient
iris.trainParam.show = 25;          %Epoxes gia emfanish
                                     %apotelesmaton
iris.trainParam.time = inf;         %Megistos xronos
                                     %ekpaideyshs

```

```

IRIS=train(iris,randomInput,randomTarget);%ekpaideysh tou diktyou
cd(w);
hgsave('train','-v6'); %apothhkeysh grafhmatos
                        %ekpadeyshs

close;
cd ..;

out_train=sim(IRIS,input); %dokimh diktyou se gnosta
                           %stoixeia
out_test=sim(IRIS,test); %dokimh diktyou se agnosta
                           %stoixeia

figure;
plot('v6',target2);hold on; %grafhma stoxou
plot('v6',out_train,'r+'); %apotelesmatos gnoston
                           %stoixeion
plot('v6',out_test,'go'); %apotelesmatps agnostos
                           %stoixeion

h=legend('Target','Output train','Output test',4);%legend
cd(w);
hgsave('output','-v6'); %apothhkeysh grafhmatos
                        %dokimon

close;
cd ..;

epityxia = ErrorNet(out_test,target2); %ypologimsos epityxias -
                                        %sfalma

cd(w);
outFiles2HiddenLayers; %apothhkeysh apotelesmaton
                        %kai stoixeion diktyou

cd ..;

%Dokimh gia thn eyresh tou kalyterou neyrokiou diktyou

point=['a':'j']; %deiktes "run"
model=1; %deikths modelou
Succes2H=zeros(105,10); %pinakas apotelesmaton

for y=2:8
    for x=2:16
        for run=1:10 %dhmiourgia fakelou gia
                    %kathe dokimh
            mkdir(['4_' int2str( x) '_' int2str( y) '_1' point(run)]);

```

```

w=['4_' int2str( x) '_' int2str( y) '_1' point(run)];
                                %onoma fakelou - modelo
tic;                                %arxh xronou
MyNet2HRdm;                          %dhmiourgia-dokimh-sygkrish
                                %diktyon
time2H(model,run)=toc;                %metrhsh xronou
                                %"ekpaideyshs" tou diktyou
Succes2H(model,run)=epityxia; %kataxorhsh apotelesmatons
                                %ston pinaka

end
model=model+1;
end
end

xlswrite('Time2H.xls',time2H);        %apothhkeysei xronon
                                %ekpaideyshs
xlswrite('succes2H.xls',Succes2H);    %apothhkeysei apotelesmaton
                                %olon ton dokimon

%Eyresh kalyterou modelou,eisagogh dedomenon: Prota ta apotelesmata %tou
modelo me 1 kryfo
%epipedo.
function FindBestNetwork(OneHiddenLayerRes,TwoHiddenLayerRes)
point=['a':'j'];

%Single Hidden Layer
[l w]=size(OneHiddenLayerRes);
[Cs Ps]=max(OneHiddenLayerRes);
[cs ps]=max(Cs);
maxValueSingle=OneHiddenLayerRes(Ps(ps),ps);
disp(['H megisth epityxia tou modelou me ena kryfo epipedo einai : '
num2str(maxValueSingle)]);
disp(['Einai h periptosh: ' int2str(Ps(ps)) ' run: ' point(ps)]);

%Two Hidden Layer
[Cd Pd]=max(TwoHiddenLayerRes);
[cd pd]=max(Cd);
maxValueDouble=TwoHiddenLayerRes(Pd(pd),pd);
disp(['H megisth epityxia tou modelou me dyo kryfa epipeda einai:'
num2str(maxValueDouble)]);
disp(['Einai h periptosh:' int2str(Pd(pd)) ' run:, ' point(pd) ] );

```



```
%Overall
dif=maxValueSingle-maxValueDouble;
if dif > 0
    disp('To kalytero apotelesma to petyxe to modelo me ena kryfo
epipedo');
else disp('To kalytero apotelesma to petyxe to modelo me dyo kryfa
epipeda');
end;
```

B.1.2. Μοντέλο RBF.

```
load data
[Input In_min In_max]=normal(input);           %kanonikopoihsh eisodou
[Target T_min T_max]=normal(target);          %kanonikopoihsh timon
                                                %stoxou

a=[1:250 501:750 1001:1250 1501:1750 2001:2250];
b=a+250;
c=[1:1250];
d=[1251:2500];
e=[1:500 751:1250 1501:2000 2251:2500];
f=[501:750 1251:1500 2001:2250];

for i=1:6;                                     %Senaria ekpaideyshs
    switch i;
        case (1)
            input=Input(a,:);
            target=Target(a,:);
            TestIn=Input(b,:);
            TestTar=Target(b,:);

        case (2)
            input=Input(b,:);
            target=Target(b,:);
            TestIn=Input(a,:);
            TestTar=Target(a,:);

        case (3)
            input=Input(c,:);
            target=Target(c,:);
            TestIn=Input(d,:);
            TestTar=Target(d,:);

        case (4)
            input=Input(d,:);
            target=Target(d,:);
            TestIn=Input(c,:);
            TestTar=Target(c,:);
```

```

        case (5)
            input=Input (e, :);
            target=Target (e, :);
            TestIn=Input (f, :);
            TestTar=Target (f, :);

        otherwise
            input=Input;
            target=Target;
            TestIn=input;
            TestTar=Target;
    end

    randomValues;                                     %tyxaia katanomh dedomenon

    for x=0:8;                                       %Syndiasmoi Neyronon sto kryfo epipedo
        MaxNeurons=2^x;
        w=['6_' int2str( i) '_' int2str( MaxNeurons) '_1' ];
        mkdir (w);
        tic;
        rbf_Net=newrb (input', target', 1e-4, 1, MaxNeurons, 2);
        time (i, x+1)=toc;
        hgsave ([w '\Train'], '-v6');
        close;
        TrainOut=sim (rbf_Net, input');
        TestOut=sim (rbf_Net, TestIn');
        D_TestTar=denormal (TestTar, T_min, T_max);
        D_TestOut=denormal (TestOut', T_min, T_max);
        error=D_TestTar-D_TestOut;
        Dt=0.02;
        L=length (TestTar);
        ISET (i, x+1)=(sum ((error.^2) *Dt) *250/L);
        IAET (i, x+1)=(sum (abs (error) *Dt) *250/L);

        figure;
        plot ('v6', target); hold on;
        plot ('v6', TrainOut, 'k');
        plot ('v6', TestTar, 'g');
        plot ('v6', TestOut, 'r');
        h=legend ('Target', 'trainOutput', 'testTarget', 'testOutput', 2);
        hgsave ([w '\NetOut'], '-v6');             %Apothikeysh Grafimaton
        RBF_outFiles;                               %Apothikeysh Dedomenon
    end
end

```

```

close;

end
end
findBestNN(ISET, IAET); %Eyresh kalyterou modelou
xlswrite('RBF_time',time);
xlswrite('RBF_IAET', IAET);
xlswrite('RBF_ISET', ISET);

%Katanemei me tyxaio tropo ta stoixeia se 2 pinakes,oi 2 pinakes prepei na
exoun to idio mhkos
i=0;
j=1;
flag=1;

X=size(input); %diastaseis tou pinaka
Y=randperm(X(1));

randomInput=normal(input(Y,1:6));
randomTarget=target(Y);

randomInput=randomInput';
randomTarget=randomTarget';

%Eyresh kalyterou modelou,eisagogh dedomenon: ISET - IAET
function FindBestNetwork(X,Y)
CASE = 1:9;
%ISET
[l w]=size(X);
[Cs Ps]=min(X);
[cs ps]=min(Cs);
minValueX=X(Ps(ps),ps);
disp(['To mikrotero oloklhromeno tetragoniko sfalma sto xrono einai : '
num2str(minValueX)]);
disp(['Einai h periptosh: ' int2str(Ps(ps)) ' case ' int2str(CASE(ps))]);

%IAET
[Cd Pd]=min(Y);
[cd pd]=min(Cd);
minValueY=Y(Pd(pd),pd);

```

```

disp(['To mikrotero oloklhromeno apolyto sfalma sto xrono einai:'
num2str(minValueY)]);
disp(['Einai h periptosh:' int2str(Pd(pd)) ' case ' int2str(CASE(pd)) ] );

%Kanonikopoihsh dedomenon (Normalize data) Xn={(0.9-0.1)*(X-%Xmin)/(Xmax-
Xmin)}+0.1
%metaksy 0.1 kai 0.9
function [X,X_min,X_max]=normal(matrix)

X_max=max(matrix);
X_min=min(matrix);

for i=1:length(X_min)
X(:,i)=((0.9-0.1)*(matrix(:,i)-X_min(i))/(X_max(i)-X_min(i)))+0.1;
end
%Apokanonikopoihsh dedomenon (DeNormalize data) ,prohgoumnos prepei %na exei
%trexei h synarthsh kanonikopoihshs "normal"
function [Y]=denormal(matrix,X_min,X_max)

for i=1:length(X_min)
Y(:,i)=(((matrix(:,i)-0.1)*(X_max(i)-X_min(i)))/0.8)+X_min(i);
end

```

B.1.3. Κοινά αρχεία – Αποθήκευση τιμών.

```
%apothjkeysh dedomenon - apotelesmaton (varh,bias ...) se arxeia %*.txt
```

```
dlmwrite('LW.txt',IRIS.LW,'newline','pc','delimiter','\t');  
dlmwrite('IW.txt',IRIS.IW,'newline','pc','delimiter','\t','precision',6);  
dlmwrite('bias.txt',IRIS.b,'newline','pc','delimiter','\t','precision',  
6);  
dlmwrite('output_test.txt',out_test,'newline','pc','delimiter',  
'\t','precision',6);  
dlmwrite('output_train.txt',out_train,'newline','pc','delimiter',  
'\t','precision',6);  
dlmwrite('epityxia.txt',epityxia,'newline','pc','delimiter','\t','precision',  
4)  
  
save irisNet.mat IRIS -v6;
```

```
%apothjkeysh dedomenon - apotelesmaton (varh,bias ...) se arxeia %*.txt
```

```
dlmwrite('LWHidden1.txt',IRIS.LW{2,1},'newline','pc','delimiter',  
'\t','precision',6);  
dlmwrite('LWHidden2.txt',IRIS.LW{3,2},'newline','pc','delimiter',  
'\t','precision',6);  
dlmwrite('IW.txt',IRIS.IW,'newline','pc','delimiter','\t','precision',6);  
dlmwrite('bias.txt',IRIS.b,'newline','pc','delimiter','\t','precision',  
6);  
dlmwrite('output_test.txt',out_test,'newline','pc','delimiter',  
'\t','precision',6);  
dlmwrite('output_train.txt',out_train,'newline','pc','delimiter',  
'\t','precision',6);  
dlmwrite('epityxia.txt',epityxia,'newline','pc','delimiter','\t','precision',  
4)  
  
save irisNet.mat IRIS -v6;
```

```
%apothjkeysh dedomenon - apotelesmaton (varh,bias ...) se arxeia %*.txt
```

```
dlmwrite('LW.txt',ControlNet.LW,'newline','pc','delimiter','\t');  
dlmwrite('IW.txt',ControlNet.IW,'newline','pc','delimiter',  
'\t','precision',6);
```

```

dlmwrite('bias.txt',ControlNet.b,'newline','pc','delimiter',
'\t','precision',6);
dlmwrite('output_test.txt',out_test,'newline','pc','delimiter',
'\t','precision',6);
dlmwrite('output_train.txt',out_train,'newline','pc','delimiter',
'\t','precision',6);
dlmwrite('epityxia.txt',epityxia,'newline','pc','delimiter','\t','precision'
,4)

save ControlNetNet.mat ControlNet -v6;
%apothjkeysh dedomenon - paotelesmaton (varh,bias ...) se arxeia *.txt

dlmwrite('LWHiden1.txt',ControlNet.LW{2,1},'newline','pc','delimiter',
'\t','precision',6);
%dlmwrite('LWHiden2.txt',ControlNet.LW{3,2},'newline','pc','delimiter',
'\t','precision',6);
dlmwrite('IW.txt',ControlNet.IW,'newline','pc','delimiter',
'\t','precision',6);
dlmwrite('bias.txt',ControlNet.b,'newline','pc','delimiter',
'\t','precision',6);
%dlmwrite('output_test.txt',out_test,'newline','pc','delimiter',
%\t','precision',6);
dlmwrite('output_train.txt',output,'newline','pc','delimiter',
'\t','precision',6);
dlmwrite('MSE.txt',MeanError,'newline','pc','delimiter','\t','precision',4)

save ControlNetNet.mat ControlNet -v6;
%apothjkeysh dedomenon - paotelesmaton (varh,bias ...) se arxeia %*.txt

dlmwrite('LWHiden1.txt',ControlNet.LW{2,1},'newline','pc','delimiter',
'\t','precision',6);
dlmwrite('LWHiden2.txt',ControlNet.LW{3,2},'newline','pc','delimiter',
'\t','precision',6);
%dlmwrite('LWHiden3.txt',ControlNet.LW{4,1},'newline','pc','delimiter',
'\t','precision',6);
dlmwrite('IW.txt',ControlNet.IW,'newline','pc','delimiter',
'\t','precision',6);
dlmwrite('bias.txt',ControlNet.b,'newline','pc','delimiter',
'\t','precision',6);
%dlmwrite('output_test.txt',out_test,'newline','pc','delimiter',
%\t','precision',6);

```

```

dlmwrite('output_train.txt',output,'newline','pc','delimiter',
'\t','precision', 6);
dlmwrite('MSE.txt',MeanError,'newline','pc','delimiter','\t','precision',4)

save ControlNetNet.mat ControlNet -v6;
%apothikeysh dedomenon - apotelesmaton (varh,bias ...) se arxeia %*.txt

dlmwrite([w '\LW.txt'],rbf_Net.LW,'newline','pc','delimiter','\t');
dlmwrite([w '\IW.txt'],rbf_Net.IW,'newline','pc','delimiter',
'\t','precision', 6);
dlmwrite([w '\bias.txt'],rbf_Net.b,'newline','pc','delimiter',
'\t','precision', 6);
dlmwrite([w '\output_test.txt'],TestOut,'newline','pc','delimiter',
'\t','precision', 6);
dlmwrite([w '\output_train.txt'],TrainOut,'newline','pc','delimiter',
'\t','precision', 6);
dlmwrite([w '\ISET.txt'],ISET(i,x+1),'newline','pc','delimiter',
'\t','precision', 6);
dlmwrite([w '\IAET.txt'],IAET(i,x+1),'newline','pc','delimiter',
'\t','precision', 6);
cd(w)
saveas(gcf,'NetOut.emf');
save rbf_Net.mat rbf_Net -v6;
cd ..

```


B.2 Κώδικας C.

B.2.1 Κώδικας μοντέλου XOR σε C (ANSI).

```
#include<stdio.h>
#include<math.h>
#include<memory.h>

#define neurons_InputLayer 2
/* Neyrones lou epipedou(eisodos)*/

#define neurons_HiddenLayer 3
/* Neyrones kryfoy epipedou*/

#define neurons_OutputLayer 1
/* Neyrones epipedou eksodou*/

/* Eisagogh Baron */
float InputHidden_Weights[neurons_InputLayer][neurons_HiddenLayer];
/*Barh Eisodou(input Weights)*/

float LayerWeights[neurons_HiddenLayer][neurons_OutputLayer];
/*Barh Eksodou(Layer Weights)*/

/* Kataxorhsh Orion (Theshold,Bias)*/
float biasHiddenLayer[neurons_HiddenLayer];
/* oria(polosh) neyronon kryfou epipedou*/

float biasOutputLayer[neurons_OutputLayer];
/*oria(polosh) neyronon epipedou eksodou*/

void read_InputHidden_Weights(const char *file_InputHidden_Weights)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i,j;
    f=fopen(file_InputHidden_Weights,"r");
    for(i=0;i<neurons_InputLayer;i++)
        for(j=0;j<neurons_HiddenLayer;j++)
            fscanf(f,"%f",&InputHidden_Weights[i][j]);
}
```

```

}
void read_LayerWeights(const char *file_LayerWeights)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i,j;
    f=fopen(file_LayerWeights,"r");
    for(i=0;i<neurons_HiddenLayer;i++)
        for(j=0;j<neurons_OutputLayer;j++)
            fscanf(f,"%f",&LayerWeights[i][j]);
}

void read_biasHiddenLayer(const char *file_biasHiddenLayer)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i;
    f=fopen(file_biasHiddenLayer,"r");
    for(i=0;i<neurons_HiddenLayer;i++)
        fscanf(f,"%f",&biasHiddenLayer[i]);
}

void read_biasOutputLayer(const char *file_biasOutputLayer)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i;
    f=fopen(file_biasOutputLayer,"r");
    for(i=0;i<neurons_OutputLayer;i++)
        fscanf(f,"%f",&biasOutputLayer[i]);
}

float logsig(float x)
/*Orismos "sigmoidous" synarthshs*/
{
    float M;
    M= 1/(1+(float)exp(-x));
    return M;
}

main()
{
    /*EPIPEDO EISODOU (1o)*/
    FILE *in,*out;

```

```

in = fopen("input.txt", "r");
out = fopen("output.txt", "w");
read_InputHidden_Weights("InputHidden_Weights.txt");
read_LayerWeights("LayerWeights.txt");
read_biasHiddenLayer("biasHiddenLayer.txt");
read_biasOutputLayer("biasOutputLayer.txt");
while(1)
{
    int i,j,flag;
/*dhlosh metavlhton*/
    float input[neurons_InputLayer];
/*eisodou*/
    float input_HiddenLayer[neurons_HiddenLayer] ;
/*eisodou kryfou epipedou*/
    float tempOut_HiddenLayer[neurons_HiddenLayer];
/*eksodou kryfou epipedou*/
    float input_OutputLayer[neurons_OutputLayer] ;
/*eisodou epipedou eksodou*/
    float Output[neurons_OutputLayer];
/*eksodou*/

    memset(input_HiddenLayer,0,sizeof(input_HiddenLayer));
/*orismos arxikhs katastashs*/
    memset(input_OutputLayer,0,sizeof(input_OutputLayer));
/*ton metavlhton-"ekatharish"*/

    flag = 0;
    for(i=0;i<neurons_InputLayer;i++)
        if(fscanf(in,"%f",&input[i])==EOF) flag = 1;
        if(flag) break;

    /*** KRYFO EPIPEDO ***/
    for(i=0;i<neurons_HiddenLayer;i++)
    {
        for(j=0;j<neurons_InputLayer;j++)
        {

/* polapllasiasmos eisodon me ta varh kai ypologismos tou athroismatos */
            input_HiddenLayer[i]
+= (float) (input[j]*InputHidden_Weights[j][i]);
        }
    }
}

```

```

tempOut_HiddenLayer[i]=logsig(input_HiddenLayer[i]+biasHiddenLayer[i]);
/*Klhsh tis synarthshs(activation function) */
    }

    /*** EPIPEDO EKSODOU ***/
    for(i=0;i<neurons_OutputLayer;i++)
    {
        for(j=0;j<neurons_HiddenLayer;j++)
        {

/* polapllasiasmos timwn me ta varh kai ypologismos athroismatos*/

input_OutputLayer[i]+=(float)(tempOut_HiddenLayer[j]*LayerWeights[j][i]);
        }
        Output[i]=logsig(input_OutputLayer[i]+biasOutputLayer[i]);
/*Klhsh tis synarthseis (activation function)*/
    }

    fprintf(out, "\nEisodos :");printf("\nEisodos");
/*Katagrafh apotelesmaton se arxeio*/

for(i=0;i<neurons_InputLayer;i++){fprintf(out, "\t%2.2f",input[i]);printf("\t
%2.4f",input[i]);}
    fprintf(out, "\nEksodos :");printf("\nEksodos");

for(j=0;j<neurons_OutputLayer;j++){fprintf(out, "\t%2.4f\n",Output[j]);printf
("\t%2.4f\n",Output[j]);}

    }
    printf("\nPieste ENTER gia telos.");
    getchar();
    return 0;
}

```

B.2.2 Κώδικας μοντέλου IRIS με ένα κρυφό επίπεδο σε C (ANSI).

```
#include<stdio.h>
#include<math.h>
#include<memory.h>

#define neurons_InputLayer 4
/* Neyrones lou epipedou(eisodos)*/

#define neurons_HiddenLayer 7
/* Neyrones kryfoy epipedou*/

#define neurons_OutputLayer 1
/* Neyrones epipedou eksodou*/

float InputHidden_Weights[neurons_InputLayer][neurons_HiddenLayer];
/*Barh Eisodou - Kryfou(IW)*/

float LayerWeights[neurons_HiddenLayer][neurons_OutputLayer];
/*Barh Kryfou-Eksodou(LW)*/

float biasHiddenLayer[neurons_HiddenLayer];
/* Oria (Theshold,Bias),neyronon kryfou epipedou*/

float biasOutputLayer[neurons_OutputLayer];
/* neyronon epipedou eksodou*/

int i,j;

void read_InputHidden_Weights(const char *file_InputHidden_Weights)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i,j;
    f=fopen(file_InputHidden_Weights,"r");
    for(i=0;i<neurons_HiddenLayer;i++)
        for(j=0;j<neurons_InputLayer;j++)
            fscanf(f,"%f",&InputHidden_Weights[j][i]);
}

void read_LayerWeights(const char *file_LayerWeights)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
```

```

{
    FILE *f;
    int i,j;
    f=fopen(file_LayerWeights,"r");
    for(i=0;i<neurons_HiddenLayer;i++)
        for(j=0;j<neurons_OutputLayer;j++)
            fscanf(f,"%f",&LayerWeights[i][j]);
}

void read_biasHiddenLayer(const char *file_biasHiddenLayer)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i;
    f=fopen(file_biasHiddenLayer,"r");
    for(i=0;i<neurons_HiddenLayer;i++)
        fscanf(f,"%f",&biasHiddenLayer[i]);
}

void read_biasOutputLayer(const char *file_biasOutputLayer)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i;
    f=fopen(file_biasOutputLayer,"r");
    for(i=0;i<neurons_OutputLayer;i++)
        fscanf(f,"%f",&biasOutputLayer[i]);
}

float tansig(float x)
{
    float N;
    N = 2/(float)(1+exp(-2*x))-1;
    return N;
}

main()
{
    int counter=0;
    /*EPIPEDO EISODOU (1o)*/
    FILE *in,*out;
    in = fopen("input.txt","r");

```

```

out = fopen("output.txt", "w");
read_InputHidden_Weights("InputHidden_Weights.txt");
read_LayerWeights("LayerWeights.txt");
read_biasHiddenLayer("biasHiddenLayer.txt");
read_biasOutputLayer("biasOutputLayer.txt");

fprintf(out, "\tEisodos \t\t\tEksodos \n");
fprintf(out, "Sepal\tSepal\tPetal\tPetal\t\tSpecies\n");
fprintf(out, "length\twidth\tlength\twidth\n\n");
printf("\tEisodos \t\t\tEksodos \n");
printf("Sepal\tSepal\tPetal\tPetal\t\tSpecies\n");
printf("length\twidth\tlength\twidth\n\n");

while(1)
{
    int i,j,flag;
/*dhlosh metavlhton*/
    float input[neurons_InputLayer];
/*eisodou*/
    float input_HiddenLayer[neurons_HiddenLayer] ;
/*eisodou kryfou epipedou*/
    float tempOut_HiddenLayer[neurons_HiddenLayer];
/*eksodou kryfou epipedou*/
    float input_OutputLayer[neurons_OutputLayer] ;
/*eisodou epipedou eksodou*/
    float Output[neurons_OutputLayer];
/*eksodou*/

    memset(input_HiddenLayer,0,sizeof(input_HiddenLayer));
/*orismos arxikhs katastashs*/
    memset(input_OutputLayer,0,sizeof(input_OutputLayer));
/*ton metavlhton-"ekkatharish"*/

    flag = 0;
    for(i=0;i<neurons_InputLayer;i++)
        if(fscanf(in, "%f", &input[i])==EOF) flag = 1;
        if(flag) break;

    /*** KRYFO EPIPEDO ***/
    for(i=0;i<neurons_HiddenLayer;i++)
    {

```

```

        for(j=0;j<neurons_InputLayer;j++)
        {

/* polapllasiasmos eisodon me ta varh kai ypologismos tou athroismatos */
            input_HiddenLayer[i]
+= (float) (input[j]*InputHidden_Weights[j][i]);
        }

tempOut_HiddenLayer[i]=tansig (input_HiddenLayer[i]+biasHiddenLayer[i]);
/*Klhsh tis synarthshs (activation function) */
    }

    /*** EPIPEDO EKSODOU ***/
    for(i=0;i<neurons_OutputLayer;i++)
    {
        for(j=0;j<neurons_HiddenLayer;j++)
        {

/* polapllasiasmos timwn me ta varh kai ypologismos athroismatos*/

input_OutputLayer[i]+=(float) (tempOut_HiddenLayer[j]*LayerWeights[i][j]);
        }
        Output[i]=tansig (input_OutputLayer[i]+biasOutputLayer[i]);
/*Klhsh tis synarthseis (activation function)*/
    }

    for(i=0;i<neurons_InputLayer;i++)
/*Katagrfh apotelesmaton se arxeio kai sthn Othonh*/
    {
        fprintf (out, "%2.2f\t", input[i]);
/* emfanish Dedomenon eisodou*/
        printf ("%2.2f\t", input[i]);
    }
    for(j=0;j<neurons_OutputLayer;j++)
    {
        fprintf (out, "\t%2.2f", Output[j]);
/*Apotelesmata - Eksosdos*/
        printf ("\t%2.2f", Output[j]);
    }
    fprintf (out, "\n"); printf ("\n");
    counter++;
}

```



```

/*
printf("biasHiddenLayer\n");
for(i=0;i<neurons_HiddenLayer;i++) printf("%f\n",biasHiddenLayer[i]);
printf("\nbiasOut: %f\n",biasOutputLayer[0]);
printf("\n Input_HiddenWeights \n");
for(i=0;i<neurons_HiddenLayer;i++){for(j=0;j<neurons_InputLayer;j++)
printf("%f\t",InputHidden_Weights[j][i]);printf("\n");}
printf("\nLayer Weights\n");

for(i=0;i<neurons_OutputLayer;i++)for(j=0;j<neurons_HiddenLayer;j++)printf("
%f\t",LayerWeights[j][i]);
*/

printf("\nPieste ENTER gia telos.");
getchar();
return 0;
}

```

B.2.3 Κώδικας μοντέλου IRIS με δύο κρυφά επίπεδα σε C (ANSI).

```
#include<stdio.h>
#include<math.h>
#include<memory.h>

#define neurons_InputLayer 4
/* Neyrones lou epipedou(eisodos)*/

#define neurons_HiddenLayer_1 8
/* Neyrones lou kryfoy epipedou*/

#define neurons_HiddenLayer_2 8
/* Neyrones 2ou kryfoy epipedou*/

#define neurons_OutputLayer 1
/* Neyrones epipedou eksodou*/

float InputHidden_Weights[neurons_InputLayer][neurons_HiddenLayer_1];
/*Barh Eisodou - Kryfou(IW)*/

float Hidden1Hidden2_Weights[neurons_HiddenLayer_1][neurons_HiddenLayer_2];
/*Barh Kryfou_1-Kryfoy2(LW)*/

float Hidden2Output_Weights[neurons_HiddenLayer_2][neurons_OutputLayer];
/*Barh Kryfou-Eksodou(LW)*/

float biasHiddenLayer_1[neurons_HiddenLayer_1];
/* Oria (Theshold,Bias),neyronon lou kryfou epipedou*/

float biasHiddenLayer_2[neurons_HiddenLayer_2];
/* Oria (Theshold,Bias),neyronon 2ou kryfou epipedou*/

float biasOutputLayer[neurons_OutputLayer];
/* neyronon epipedou eksodou*/

int i,j;

void read_InputHidden_Weights(const char *file_InputHidden_Weights)
/*synarthsh eisagoghsh timon apo arxeio text (.txt)*/
{
    FILE *f;
```

```

    int i,j;
    f=fopen(file_InputHidden_Weights,"r");
    for(i=0;i<neurons_HiddenLayer_1;i++)
        for(j=0;j<neurons_InputLayer;j++)
            fscanf(f,"%f",&InputHidden_Weights[j][i]);
}

void read_Hidden1Hidden2_Weights(const char *file_Hidden1Hidden2_Weigths)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i,j;
    f=fopen(file_Hidden1Hidden2_Weigths,"r");
    for(i=0;i<neurons_HiddenLayer_1;i++)
        for(j=0;j<neurons_HiddenLayer_2;j++)
            fscanf(f,"%f",&Hidden1Hidden2_Weights[j][i]);
}

void read_Hidden2Output_Weights(const char *file_Hidden2Output_Weigths)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i,j;
    f=fopen(file_Hidden2Output_Weigths,"r");
    for(i=0;i<neurons_HiddenLayer_2;i++)
        for(j=0;j<neurons_OutputLayer;j++)
            fscanf(f,"%f",&Hidden2Output_Weights[i][j]);
}

void read_biasHiddenLayer_1(const char *file_biasHiddenLayer_1)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i;
    f=fopen(file_biasHiddenLayer_1,"r");
    for(i=0;i<neurons_HiddenLayer_1;i++)
        fscanf(f,"%f",&biasHiddenLayer_1[i]);
}

void read_biasHiddenLayer_2(const char *file_biasHiddenLayer_2)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i;
    f=fopen(file_biasHiddenLayer_2,"r");
    for(i=0;i<neurons_HiddenLayer_2;i++)
        fscanf(f,"%f",&biasHiddenLayer_2[i]);
}

```

```

}
void read_biasOutputLayer(const char *file_biasOutputLayer)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i;
    f=fopen(file_biasOutputLayer, "r");
    for(i=0;i<neurons_OutputLayer;i++)
        fscanf(f, "%f", &biasOutputLayer[i]);
}
float tansig(float x)
{
    float N;
    N = 2/(float) (1+exp(-2*x))-1;
    return N;
}

main()
{
    int counter=0;
    /*EPIPEDO EISODOU (1o)*/
    FILE *in, *out;
    in = fopen("input.txt", "r");
    out = fopen("output.txt", "w");

    read_InputHidden_Weights("InputHidden_Weights.txt");
    read_Hidden1Hidden2_Weights("Hidden1Hidden2_Weights.txt");
    read_Hidden2Output_Weights("Hidden2Output_Weights.txt");

    read_biasHiddenLayer_1("biasHiddenLayer1.txt");
    read_biasHiddenLayer_2("biasHiddenLayer2.txt");
    read_biasOutputLayer("biasOutputLayer.txt");

    fprintf(out, "\tEisodos \t\t\tEksodos \n");
    fprintf(out, "Sepal\tSepal\tPetal\tPetal\t\tSpecies\n");
    fprintf(out, "length\twidth\tlength\twidth\n\n");
    printf("\tEisodos \t\t\tEksodos \n");
    printf("Sepal\tSepal\tPetal\tPetal\t\tSpecies\n");
    printf("length\twidth\tlength\twidth\n\n");

    while(1)
    {

```

```

        int i,j,flag;
/*dhlosh metavlhton*/
        float input[neurons_InputLayer];
/*eisodou*/
        float input_HiddenLayer_1[neurons_HiddenLayer_1];
/*eisodou kryfou epipedou*/
        float tempOut_HiddenLayer_1[neurons_HiddenLayer_1];
/*eksodou kryfou epipedou*/
        float input_HiddenLayer_2[neurons_HiddenLayer_2];
        float tempOut_HiddenLayer_2[neurons_HiddenLayer_2];
        float input_OutputLayer[neurons_OutputLayer];
/*eisodou epipedou eksodou*/
        float Output[neurons_OutputLayer];
/*eksodou*/

        memset(input_HiddenLayer_1,0,sizeof(input_HiddenLayer_1));
/*orismos arxikhs katastashs*/
        memset(input_HiddenLayer_2,0,sizeof(input_HiddenLayer_2));
        memset(input_OutputLayer,0,sizeof(input_OutputLayer));
/*ton metavlhton-"ekatharish"*/

        flag = 0;
/*Eisagogn dedomenon (EISODOs) */
        for(i=0;i<neurons_InputLayer;i++)
            if(fscanf(in,"%f",&input[i])==EOF) flag = 1;
            if(flag) break;

        /***PROTO KRYFO EPIPEDO ***/
        for(i=0;i<neurons_HiddenLayer_1;i++)
        {
            for(j=0;j<neurons_InputLayer;j++)
            {

/* polapllasiasmos eisodon me ta varh kai ypologismos tou athroismatos */
                input_HiddenLayer_1[i]
+=(float) (input[j]*InputHidden_Weights[j][i]);
            }

tempOut_HiddenLayer_1[i]=tansig(input_HiddenLayer_1[i]+biasHiddenLayer_1[i])
;                /*Klhsh tis synarthshs(activation function) */
        }

```

```

    /***DEYTERO KRYFO EPIPEDO ***/
    for(i=0;i<neurons_HiddenLayer_2;i++)
    {
        for(j=0;j<neurons_HiddenLayer_1;j++)
        {

/* polapllasiasmos eisodon me ta varh kai ypologismos tou athroismatos */
            input_HiddenLayer_2[i]
+= (float) (tempOut_HiddenLayer_1[j]*Hidden1Hidden2_Weights[j][i]);
        }

tempOut_HiddenLayer_2[i]=tansig(input_HiddenLayer_2[i]+biasHiddenLayer_2[i])
;
        /*Klhsh tis synarthshs(activation function) */
    }

    /*** EPIPEDO EKSODOU ***/
    for(i=0;i<neurons_OutputLayer;i++)
    {
        for(j=0;j<neurons_HiddenLayer_2;j++)
        {

/* polapllasiasmos timwn me ta varh kai ypologismos athroismatos*/

input_OutputLayer[i]+=(float) (tempOut_HiddenLayer_2[j]*Hidden2Output_Weights
[i][j]);
        }
        Output[i]=tansig(input_OutputLayer[i]+biasOutputLayer[i]);
/*Klhsh tis synarthseis (activation function)*/
    }

    for(i=0;i<neurons_InputLayer;i++)
/*Katagrfh apotelesmaton se arxeio kai sthn Othonh*/
    {
        fprintf(out,"%2.2f\t",input[i]);
/* emfanish Dedomenon eisodou*/
        printf("%2.2f\t",input[i]);
    }
    for(j=0;j<neurons_OutputLayer;j++)
    {
        fprintf(out,"\t%2.2f",Output[j]);
/*Apotelesmata - Eksosdos*/
        printf("\t%2.3f",Output[j]);
    }

```

```

    }
    fprintf(out, "\n"); printf("\n");
    counter++;
}
/*
printf("biasHiddenLayer_1\n");
for(i=0;i<neurons_HiddenLayer_1;i++)
printf("%f\n",biasHiddenLayer_1[i]);
printf("biasHiddenLayer_2\n");
for(i=0;i<neurons_HiddenLayer_2;i++)
printf("%f\n",biasHiddenLayer_2[i]);
printf("\nbiasOut: %f\n",biasOutputLayer[0]);
printf("\n Input_HiddenWeights \n");
for(i=0;i<neurons_HiddenLayer_1;i++){for(j=0;j<neurons_InputLayer;j++)
printf("%f\t",InputHidden_Weights[j][i]);printf("\n");}
printf("\nHidden1Hidden2_Weights\n");

for(i=0;i<neurons_HiddenLayer_2;i++)for(j=0;j<neurons_HiddenLayer_1;j++)prin
tf("%f\t",Hidden1Hidden2_Weights[j][i]);
printf("\nHidde2Output_Weights\n");

for(i=0;i<neurons_OutputLayer;i++)for(j=0;j<neurons_HiddenLayer_2;j++)printf
("%f\t",Hidden2Output_Weights[j][i]);
*/

printf("\nPieste ENTER gia telos.");
getchar();
return 0;
}

```

B.2.4. Κώδικας μοντέλου RBF σε C (ANSI).

```
#include<stdio.h>
#include<math.h>
#include<memory.h>

#define neurons_InputLayer 6
/* Neyrones lou epipedou(eisodos)*/

#define neurons_HiddenLayer 64
/* Neyrones kryfoy epipedou*/

#define neurons_OutputLayer 1
/* Neyrones epipedou eksodou*/

double InputHidden_Weights[neurons_HiddenLayer][neurons_InputLayer];
/*Barh Eisodou - Kryfou(IW)*/

double HiddenOutput_Weights[neurons_HiddenLayer][neurons_OutputLayer];
/*Barh Kryfou-Eksodou(LW)*/

double biasHiddenLayer[neurons_HiddenLayer];
/* neyronon kryfou epipedou*/

double biasOutputLayer[neurons_OutputLayer];
/* neyronon epipedou eksodou*/

int i,j;

void read_InputHidden_Weights(const char *file_InputHidden_Weights)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i,j;
    f=fopen(file_InputHidden_Weights,"r");
    for(i=0;i<neurons_HiddenLayer;i++)
        for(j=0;j<neurons_InputLayer;j++)
            fscanf(f,"%lf",&InputHidden_Weights[i][j]);
}

void read_HiddenOutput_Weights(const char *file_HiddenOutput_Weigths)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
```



```

FILE *f;
int i,j;
f=fopen(file_HiddenOutput_Weigths,"r");
for(i=0;i<neurons_HiddenLayer;i++)
    for(j=0;j<neurons_OutputLayer;j++)
        fscanf(f,"%lf",&HiddenOutput_Weights[i][j]);
}
void read_biasAll(const char *file_bias)
/*synarthsh eisagoghs timon apo arxeio text (.txt)*/
{
    FILE *f;
    int i;
    f=fopen(file_bias,"r");
    for(i=0;i<neurons_HiddenLayer;i++)
        fscanf(f,"%lf",&biasHiddenLayer[i]);

    for(i=0;i<neurons_OutputLayer;i++)
        fscanf(f,"%lf",&biasOutputLayer[i]);
}
double tansig(double x)
{
    double N;
    N = 2/((double) (1+exp(-2*x)))-1;
    return N;
}

double radbas(double x)
{
    return (double) exp(-x*x);
}

double purelin(double x)
{
    return x;
}

main()
{
    int counter=0;
    /*EPIPEDO EISODOU (1o)*/
    FILE *in,*out;
    in = fopen("input.txt","r");

```

```

out = fopen("output.txt", "w");

read_InputHidden_Weights("IW2.txt");
read_HiddenOutput_Weights("LW2.txt");
read_biasAll("bias.txt");

//fprintf(out, "\tEisodos \t\t\t\t\tEksodos \n");
//fprintf(out, "input1\tinput2\tinput3\tinput4\t\tinput5\t\tinput6\n");
printf("\tEisodos \t\t\t\t\tEksodos \n");
printf("input1\tinput2\tinput3\tinput4\t\tinput5\t\tinput6\n");

while(1)
{
    int i, j, flag;
/*dhlosh metavlhton*/
    double input[neurons_InputLayer];
/*eisodou*/
    double input_HiddenLayer[neurons_HiddenLayer];
/*eisodou kryfou epipedou*/
    double tempOut_HiddenLayer[neurons_HiddenLayer];
/*eksodou kryfou epipedou*/
    double input_OutputLayer[neurons_OutputLayer];
/*eisodou epipedou eksodou*/
    double Output[neurons_OutputLayer];
/*eksodou*/

    memset(input_HiddenLayer, 0, sizeof(input_HiddenLayer));
    memset(input_OutputLayer, 0, sizeof(input_OutputLayer));
/*ton metavlhton-"ekatharish"*/

    flag = 0;
/*Eisagogn dedomenon (EISODOs) */
    for(i=0; i<neurons_InputLayer; i++)
        if(fscanf(in, "%lf", &input[i]) == EOF) flag = 1;
    if(flag) break;

    /***PROTO KRYFO EPIPEDO ***/
    for(i=0; i<neurons_HiddenLayer; i++)
    {
/*******/
/* Eykleidia apostash eisodoy me ta barh */
        for(j=0; j<neurons_InputLayer; j++)

```

```

        {
            double diff = (double) (input[j]-InputHidden_Weights[i][j]);
            input_HiddenLayer[i] += (double) diff*diff;
        }
input_HiddenLayer[i] = (double) sqrt(input_HiddenLayer[i]);
/*****/

tempOut_HiddenLayer[i]=radbas(input_HiddenLayer[i]+biasHiddenLayer[i]);
/*Klhsh tis synarthshs (activation function) */
    }

    /*** EPIPEDO EKSODOU ***/
    for(i=0;i<neurons_OutputLayer;i++)
    {
        double temp = 0;
        for(j=0;j<neurons_HiddenLayer;j++)
        {
            temp+=(double) (tempOut_HiddenLayer[j]*HiddenOutput_Weights[i][j]);
            /* polapllasiasmos timwn me ta varh kai ypologismos
athroismatos*/

input_OutputLayer[i]+=(double) (tempOut_HiddenLayer[j]*HiddenOutput_Weights[i
][j]);
        }
        Output[i]=purelin(input_OutputLayer[i]+biasOutputLayer[i]);
/*Klhsh tis synarthseis (activation function)*/
    }

    for(i=0;i<neurons_InputLayer;i++)
/*Katagrph apotelesmaton se arxeio kai sthn Othonh*/
    {
        //fprintf(out,"%2.2f\t",input[i]);
/* emfanish Dedomenon eisodou*/
        printf("%2.2f\t",input[i]);
    }
    for(j=0;j<neurons_OutputLayer;j++)
    {
        //fprintf(out,"\t%2.2f",Output[j]);
/*Apotelesmata - Eksodos*/
        fprintf(out,"%2.2f",Output[j]);
        printf("\t%2.3f",Output[j]);
    }

```

```
        fprintf(out, "\n"); printf("\n");
        counter++;
    }

    printf("\nPieste ENTER gia telos.");
    getchar();
    return 0;
}
```

**ΠΑΡΑΡΤΗΜΑ Γ: ΠΕΡΙΓΡΑΦΗ ΠΡΑΓΜΑΤΙΚΟΥ ΕΜΠΟΡΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ
ΔΙΕΓΕΡΣΗΣ ΣΥΓΧΡΟΝΗΣ ΓΕΝΝΗΤΡΙΑΣ**



Εισαγωγή

Το σύστημα διέγερσης τύπου GENESYS αναπτύχθηκε για να παράγει και να ελέγχει το ρεύμα πεδίου σύγχρονων γεννητριών και μηχανών σε εγκαταστάσεις παραγωγής και σε βιομηχανικές εφαρμογές.

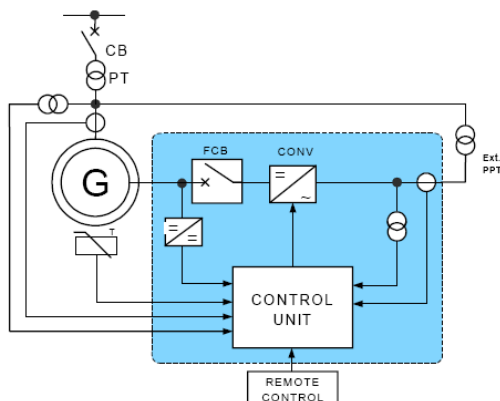
Επιπλέον του ελέγχου ισχύος το σύστημα διέγερσης τύπου GENESYS παρέχει επίσης όλες τις κατάλληλες διορθώσεις, τρόπους λειτουργίας και διατηρεί την μηχανή σε ασφαλή κλίμακα λειτουργίας, αποφεύγοντας υπερθέρμανση και αστάθεια.

Το κύριο χαρακτηριστικό του συστήματος τύπου GENESYS είναι η ευελιξία ως προς το μέγεθος και τις λειτουργίες. Το σύστημα ισχύος αποτελείται από υπομονάδες και το προγραμματιζόμενο σύστημα ελέγχου καταστούν εφικτή την προσαρμογή του συστήματος διέγερσης στις εκάστοτε απαιτήσεις.

Εφαρμογές συστήματος διέγερσης

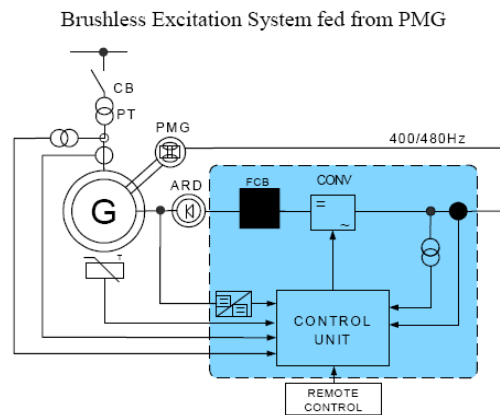
Τα σύγχρονα συστήματα διέγερσης εφαρμόζονται σε στατικές και brushless εκδόσεις.

Static Excitation System fed from the generator terminal



Ο στατικός τύπος συστήματος διέγερσης παράγει ρεύμα πεδίου χρησιμοποιώντας ένα μετατροπέα θυρίστορ Β6C και άγει το ρεύμα στο πεδίο με δακτυλίου επαφής. Ο μετατροπέας ισχύος τροφοδοτείται από εναλλακτικό πλέγμα της

εγκατάστασης παραγωγής ή της τερματικές γεννήτριες διαμέσου του μετασχηματιστή διέγερσης.



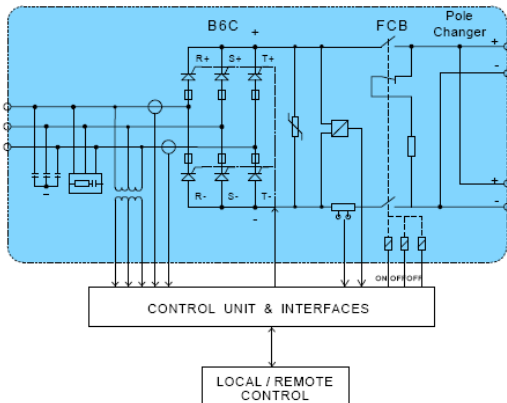
Ο brushless τύπος συστήματος διέγερσης παράγει το ρεύμα πεδίου εφαρμόζοντας ένα περιστρεφόμενο μηχανήμα ενίσχυσης με ένα περιστρεφόμενο ανορθωτή διόδων. Ο μετατροπέας χαμηλής ισχύος ελέγχει το ρεύμα πεδίου του ενισχυτή. Ένα από τα πιο αξιόπιστα είδη μετατροπέων ισχύος είναι η γεννήτρια μόνιμου μαγνήτη (permanent magnet generator – PMG) ενσωματωμένη στον άξονα της μηχανής.

Τεχνικά Χαρακτηριστικά

Ρεύμα λειτουργίας:	max 4000 A
Μέγιστη τάση:	max 1500 V
Μέγιστο ρεύμα για 30 sec	max 6000 A
Συχνότητα λειτουργίας	50 / 60 Hz
Θερμοκρασία λειτουργίας	0...40 °C
Θερμοκρασία αποθήκευσης	-30...70 C
Σχετική υγρασία	0...90%
Υψόμετρο	<1000m
Περιτύλιγμα προστασίας	IP 32
Κατασκευή:	Ατσάλινο πλαίσιο .
Ψύξη:	επιβεβλημένος κλιματισμός άνω των 300 A
Standards:	IEC147, IEEE 421, ANSI C34.2, ANSI C37.18, EN 61000

Επιπλέον είναι διαθέσιμες επιλογές κατά παραγγελία.

Κύριο Κύκλωμα



Το κύρια εξαρτήματα του στατικού συστήματος διέγερσης είναι ο μετατροπέας ισχύος, το σύστημα εκφόρτισης πεδίου και εναλλάκτης πολικότητας.

Ο μετατροπέας ισχύος είναι ένα τριφασικό θυρίστορ, ελεγχόμενο από αερόψυκτη γέφυρα ανόρθωσης (B6C) προσαρμοσμένη στα όρια διέγερσης της σύγχρονης μηχανής. Τα θυρίστορ στον μετατροπέα προστατεύονται από υπέρταση και υπερένταση διαμέσου ημιαγωγίμων ασφαλειών και κυκλώματα προστασίας σε κάθε βραχίονα και καταστολή υπέρτασης σε AC και DC.

Το σύστημα εκφόρτισης πεδίου βασίζεται είτε σε ένα μοχλό στατικού τύπου είτε σε ένα ηλεκτρομηχανικό ασφαλειοδιακόπτη με αντίσταση εκφόρτισης.

Ο εναλλάκτης πολικότητας δίνει την δυνατότητα εξισορρόπησης της τριβής των δακτυλίων επαφής ψηκρών, εναλλάσσοντας την πολικότητα του ρεύματος πεδίου σε κάθε χρόνο.

Το απλοποιημένο και μικρότερο κύριο κύκλωμα ενός brushless συστήματος διέγερσης βασίζεται σε μία γέφυρα θυρίστορ ή σε ένα κόπτη IGBT και δεν περιλαμβάνει εναλλάκτη πολικότητας.

Όταν απαιτείται αυξημένη αξιοπιστία το σύστημα διέγερσης προσφέρει πλεονάζουσες δυνατότητες ρύθμισης, διασφαλίζοντας την λειτουργία χωρίς περιορισμό ισχύος ακόμα και σε ειδικές περιπτώσεις αστοχιών. Ο μετατροπέας γίνεται διπλός ή κάθε βραχίονας του ανορθωτή

περιλαμβάνει $n+1$ θυρίστορ παράλληλα, παρέχοντας δυνατότητα πλήρους φορτίου ακόμα και αν ένα θυρίστορ αστοχήσει.

Έλεγχος και Διαγνωστικά

Το σύστημα διέγερσης GENESYS παρέχει όλες τις σχετικές καταστάσεις λειτουργίες και ρύθμισης τοπικά και τηλεχειριζόμενα.

- Χειροκίνητος Ρυθμιστής Τάσης (MVR)
- Αυτόματος Ρυθμιστής Τάσης (AVR)
- Ρυθμιστής Ενεργούς Ισχύος
- Ρυθμιστής Συντελεστή Ισχύος

Οι ενσωματωμένοι περιοριστές προστατεύουν την γεννήτρια ή μηχανή από υπερθέρμανση και αστάθεια.

- Περιοριστής ρεύματος διέγερσης
- Περιοριστής ρεύματος στάτη
- Περιοριστής υπο-διέγερσης
- Περιοριστής ρεύματος διέγερσης
- Περιοριστής υπερθέρμανσης πυρήνα
- Περιοριστής Τάσης/Συχνότητας

Ειδικές λειτουργίες σε εφαρμογές σύγχρονων μηχανών

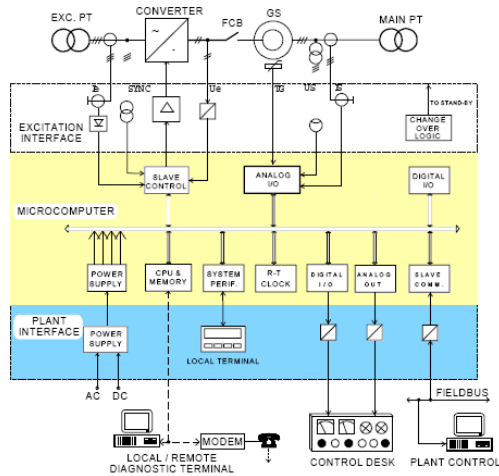
- Έλεγχος ασύγχρονης εκκίνησης

Ο ενσωματωμένος εκτεταμένος έλεγχος σφαλμάτων και προστασίας αποτρέπουν αστοχίες ή καταστροφή εξοπλισμού.

- εντοπισμός βλαβών για παροχή ισχύος, ανεμιστήρες, ασφάλειες και μονάδας ελέγχου,
- έλεγχος σήματος ανάδρασης,
- υπολογισμός θερμοκρασίας δρομέα,
- παρακολούθηση των περιστρεφόμενων δίοδων του brushless διεγέρτη.

Μονάδα ελέγχου

Control Unit



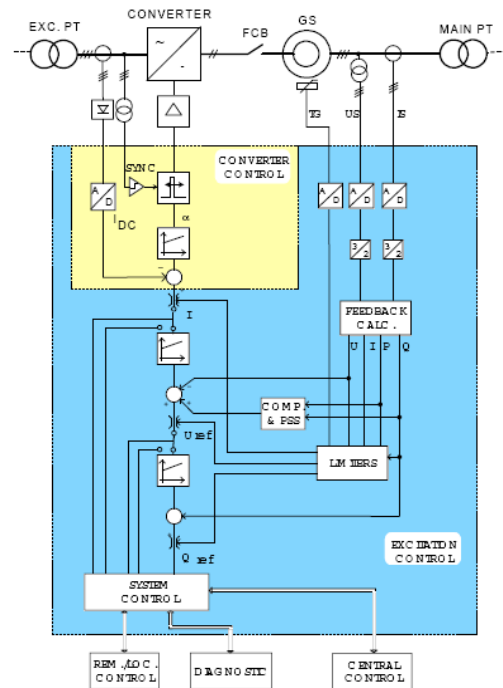
Η μονάδα ελέγχου του συστήματος διέγερσης αποτελείται από έναν μικροϋπολογιστή, με ολοκληρωμένες διεπαφές με τον μετατροπέα ισχύος και το σύστημα ελέγχου.

Οι διεπαφές με τον μετατροπέα περιλαμβάνουν ενισχυτές παλμού για τα θυρίστρος, και κυκλώματα προσαρμογής σήματος και συλλογή δεδομένων.

Ο μικροϋπολογιστής προσφέρει μια δομή καταμεμημένων master-slave πολυ-επεξεργασιών σε μια πλακέτα υψηλής πυκνότητας. Η εύρωστη βιομηχανική κατασκευή και ο ενδεδειγμένος έλεγχος σφαλμάτων υλικού και λογισμικού εγγυάται υψηλή αξιοπιστία στο εργοστάσιο παραγωγής ισχύος.

Όταν απαιτείται αυξημένη αξιοπιστία το σύστημα διέγερσης προσφέρει πλεονάζουσες δυνατότητες ρύθμισης. Όλα τα εξαρτήματα του συστήματος ελέγχου είναι σε πλήρη διπλότυπα σε δύο ανεξάρτητα κανάλια ελέγχου. Τα κανάλια ελέγχου είναι πανομοιότυπα και μπορούν να αντικατασταθούν εν λειτουργία για τη διασφάλιση αδιάλειπτης λειτουργία σε μία αντικατάσταση για συντήρηση ή λόγω βλάβης.

Διεπαφή Ανθρώπου-Μηχανής



Η υψηλού επιπέδου διεπαφή παρέχει όλες τις σχετικές λειτουργίες όπως συμβολική πρόσβαση σε μεταβλητές, ενσωματωμένος παλμογράφος, καταγραφείας δεδομένων 8 καναλιών με οθόνη, καταγραφή συμβάντων, αποθήκευση δεδομένων και εκτύπωση για αρχειοθέτηση.

Το σύστημα διέγερσης είναι προσβάσιμο από τη διεπαφή στο host PC είτε τοπικά (RS232) είτε μέσω modem και γραμμής τηλεφώνου ως απομακρυσμένο τερματικό διαγνωστικών.

Οι λειτουργίες MVR, AVR και ο έλεγχος ενεργού ισχύος είναι διευθετημένος σε μια σειρά τριών επιπέδων ελέγχου.

Ο πιο εσωτερικός βρόχος μπορεί να επιλέγει τον ελεγκτή τάσης ή ρεύματος.

Ο κύριος βρόχος ελέγχου είναι ο έλεγχος τάσης του στάτη ολοκληρωμένος με του περιοριστές προστασίας. Ο υπολογισμό ανάδρασης και ο έλεγχος βασίζονται στις εξισώσεις Park.

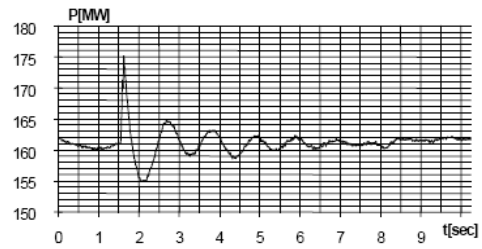
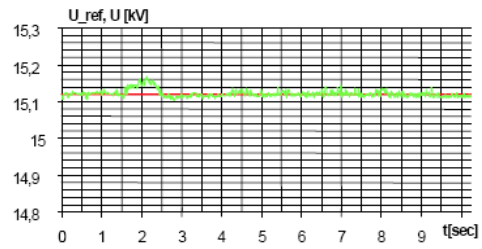
Οι περιοριστές υπερθέρμανσης της γεννήτριας (στάτη, περιοριστές ρεύματος πεδίου και

περιοριστές υπερθέρμανσης πυρήνα) είναι προσδοσμένοι άμεσα με τη θερμοκρασία της εσωτερικής ψήκτρας, για να επιτρέψει μέγιστο φορτίο σε κάθε περίπτωση.

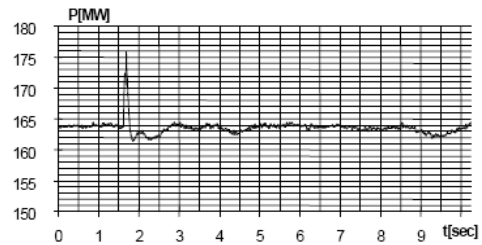
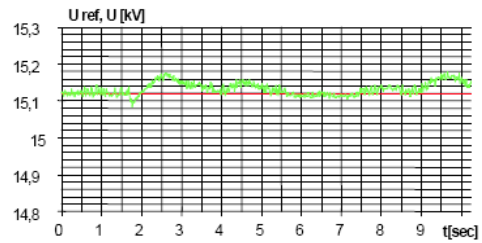
Ο περιοριστής στατικής ευστάθειας (υπό όριο διέγερσης) αποτρέπει την πτώση της γεννήτριας από τη θέση της. Η τιμή του ορίου είναι άμεσα προσαρμοσμένη στα μεγέθη του στάτη και την εμπέδηση του δικτύου.

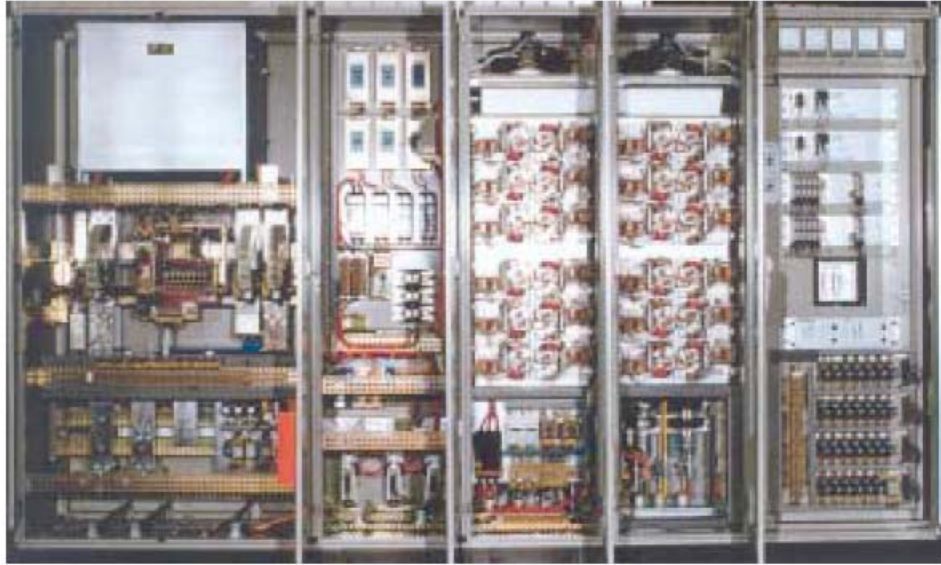
Ο υπερτεθειμένος βρόχος μπορεί να εναλλάσσεται ως ελεγκτής ενεργούς ισχύος ή συντελεστή ισχύος.

Transient on the transmission line with switched off PSS

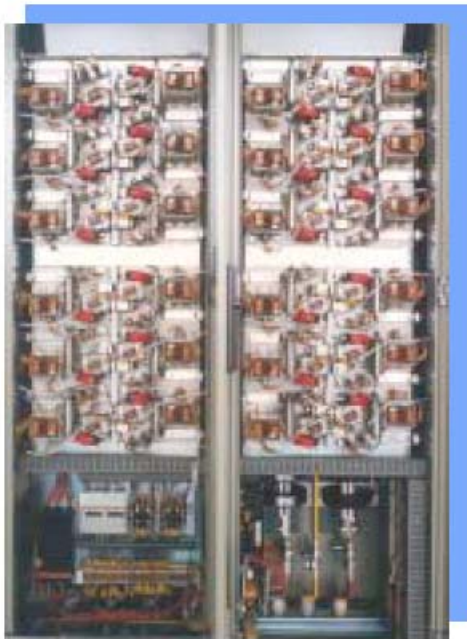


Transient on the transmission line with switched on PSS

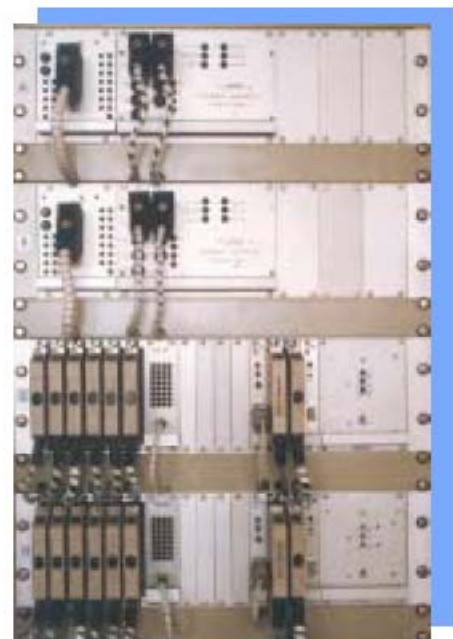




SG type Static Excitation System Cubicle Row



4 Paralell Thyristor Converters (B6C)



Duplicated (one hot standby) Control Unit



GSE type Excitation Regulator (AVR)



GSE type AVR under Test

ΠΑΡΑΡΤΗΜΑ Δ: ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΣΥΣΤΗΜΑΤΟΣ Η/Υ ΣΤΟΝ ΟΠΟΙΟ ΠΡΑΓΜΑΤΟΠΟΙΗΘΗΚΑΝ ΟΙ ΔΟΚΙΜΕΣ ΚΑΙ ΟΙ ΠΡΟΣΟΜΟΙΩΣΕΙΣ ΤΗΣ ΠΑΡΟΥΣΑΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Κεντρικός Επεξεργαστής (CPU):	AMD ATHLON 64bit @ 2GHz
Μνήμη RAM:	1024 MB DDR SDRAM @ 400MHz
Σκληρός Δίσκος (HDD):	160GB SATA RAID 300MB/S
Λειτουργικό Σύστημα (OS):	MS Windows XP Pro
Περιβάλλον Προγραμματισμού:	MS Visual Studio
Περιβάλλον Προσομοίωσης	MATLab 7.0 (R14)

ΑΝΑΦΟΡΕΣ

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Larsen, E.V., Swann, D.A., "Applying Power System Stabilisers I-II-III", *IEEE Trans. on PAS*, vol. PAS-100, no. 6, June 1981, pp.3017-3046.
- [2] IEEE, "Recommended practice for excitation system models for power system stability studies", IEEE Std. 421.5-1992, 1992.
- [3] Herzog, H., Baumberger, H., "Digital control of generators", *ABB Review*, 91/1990, pp. 27-32.
- [4] Mao H; Malik O.P.; Hope G.S.; Fan J., "An adaptive generator excitation controller based on linear optimal control", *IEEE Trans. on Energy Conversion*, vol. 5, no. 4, 1990, pp. 673-678.
- [5] Papadopoulos D.P., "Excitation control of turbogenerators with output feedback", *Int J Electr Power & Energy Systems*, vol. 8, 1986, pp.176-181.
- [6] Hassan M.A.M.; Malik O.P.; Hope G.S., "A fuzzy logic based stabilizer for a synchronous machine", *IEEE Trans Energy Conversion*, vol. 6, no. 3, 1991, pp. 407-413.
- [7] Handschin E.; Hoffmann W., Reyer F., Stephanblome Th., Schlucking U., Westermann D., Ahmed S.S., "A new method of excitation control based on fuzzy set theory", *IEEE Trans. on Power Systems*, vol. 9, 1994, pp. 533-539.
- [8] Djukanovic M.B., Dobrijevic D.M., Calovic M.S., Novicevic M., Sobajic D.J., "Coordinated stabilizing control for the exciter and governor loops using fuzzy set theory and neural nets", *Int. J. Electrical Power & Energy Systems*, vol. 8, 1997, pp. 489-499.
- [9] Karnavas Y.L., Papadopoulos D.P., "Excitation control of a power generating system based on fuzzy logic and neural networks", *European Transactions on Electrical Power (ETEP)*, vol. 10, no. 4, 2000, pp. 233-241.
- [10] Karnavas Y.L., Papadopoulos D.P., "A genetic-fuzzy system for the excitation control of a synchronous machine", in *Proc. of ICEM '02, the 15th International Conference in Electrical Machines*, Bruges, Belgium, 25-28 August, 2002, CD paper Ref. No. 204.
- [11] Corbetta, G., Ottaviani, G. "Digital measurement procedures in a microprocessor-based excitation system", in *Proc. of EPE*, Firenze, vol. 3, 1991, pp.399-402.
- [12] Bojtor, L., Nagy, B., Rakovszky, Gy., "Microcomputer control for electrical drives and synchronous generators", in *Proc. of IFAC Workshop on Motion Control*, Perugia, 1992.
- [13] Nagy, B., Bojtor, L., Rakovszky, Gy., "Microcomputer controlled excitation system for synchronous generators", in *Proc. of ISIE '93*, Budapest, 1993, pp. 183-187.
- [14] Bojtor, L., Nagy, B., Rakovszky, Gy., "A Novel Excitation System for Power Plant Generators", in *Proc. of EPE*, Brighton, 1993.
- [15] Ρίζος Γ., "Τεχνητά Νευρωνικά Δίκτυα", εκδ. Νέων Τεχνολογιών, ΑΘΗΝΑ 1996.
- [16] Krause P., "Analysis Of Electrical Machinery", J. Willey & Sons, 1995.
- [17] Ντοκόπουλος Π., "Εισαγωγή Στα Συστήματα Ηλεκτρικής Ενέργειας", εκδ. ΠΑΡΑΤΗΡΗΤΗΣ, Θεσσαλονίκη 1986.

ΙΣΤΟΣΕΛΙΔΕΣ (URL)

- [18] <http://www.fbk.com/electrical-power-machines/>
- [19] <http://www.mathworks.com/>
- [20] <http://www.avrfreaks.net/>
- [21] http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html
- [22] <http://www.cs.stir.ac.uk/~lss/NNIntro/InvSlides.html>
- [23] <http://www.palisade-europe.com/neuraltools/>
- [24] <http://www.statsoft.com/textbook/stneunet.html>
- [25] International Neural Network Society <http://www.inns.org/>
- [26] IEEE Computational Intelligence Society <http://ieee-cis.org/>
- [27] <http://www.atmel.com/>