



**ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΕΠΙΚΟΙΝΩΝΙΩΝ**

Τίτλος Πτυχιακής Εργασίας :

**Υλοποίηση σε παραθυρικό περιβάλλον, με τη χρήση του προγράμματος
MATLAB, του δυναμικού νευρωνικού δικτύου Block-Diagonal Recurrent
Neural Network (BDRNN) και του αλγόριθμου εκμάθησης Recurrent
Neural Network Constrained Optimization Method (RENNCOM)**

Σπουδαστής:

Χειρδάρης Σταμάτης

Επιβλέπων:

Δρ. Πάρις Μαστοροκώστας

Αναπληρωτής Καθηγητής

Σέρρες 2006



**ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΕΠΙΚΟΙΝΩΝΙΩΝ**

Τίτλος Πτυχιακής Εργασίας :

**Υλοποίηση σε παραθυρικό περιβάλλον, με τη χρήση του προγράμματος
MATLAB, του δυναμικού νευρωνικού δικτύου Block-Diagonal Recurrent
Neural Network (BDRNN) και του αλγόριθμου εκμάθησης Recurrent
Neural Network Constrained Optimization Method (RENNCOM)**

Σπουδαστής:

Χειρδάρης Σταμάτης

Επιβλέπων:

Δρ. Πάρις Μαστοροκώστας

Αναπληρωτής Καθηγητής

Το παρόν CD περιλαμβάνει δυο καταλόγους

1. Κατάλογος DOCUMENTATION: περιλαμβάνει την έντυπη μορφή της εργασίας στο αρχείο bdrnn_renncom.pdf και το εξώφυλλο στο αρχείο cover.pdf.
2. Κατάλογος CODE: περιλαμβάνει τα αρχεία που συντελούν στην υλοποίηση της εφαρμογής.

Σέρρες 2006

Περιεχόμενα

Κεφάλαιο 1

Περιγραφή του μοντέλου

1.1 Το αναδρομικό νευρωνικό δίκτυο με πίνακα βαρών <i>Ο αλγόριθμος εκπαίδευσης RENNCOM</i>	4
1.2 Ορισμοί και διατύπωση του προβλήματος	6
1.2.1 Ενημέρωση των παραμέτρων	8

Κεφάλαιο 2

2.1 Περιγραφή προγράμματος <i>Ανάλυση Κώδικα</i>	10
2.2 Περιγραφή κώδικα για την περίπτωση test case	15
2.3 Περιγραφή κώδικα για την περίπτωση των πνευμονικών ήχων CC, FC, SQ	27

Κεφάλαιο 3

<i>Αξιολόγηση του μοντέλου με εφαρμογή σε πρόβλημα αναγνώρισης μη γραμμικού δυναμικού συστήματος</i>	
3.1 Αναγνώριση ενός μη γραμμικού δυναμικού συστήματος	38

Κεφάλαιο 4

Παράδειγμα πνευμονικών ήχων

4.1 Εφαρμογή του BDRNN στην ανάλυση των πνευμονικών ήχων	58
4.1.1 Βασικά χαρακτηριστικά των αναπνευστικών ήχων και της σχετιζόμενης παθολογίας	58
4.1.2 Το φίλτρο BDRNN	60

Πίνακες

Πίνακας 1	38
Πίνακας 1.1	56
Πίνακας 2	61
Πίνακας 2.1	62
Πίνακας 3	171
Πίνακας 3.1	172
Πίνακας 4	281
Πίνακας 4.1	282

Παράρτημα	310
------------------	-----

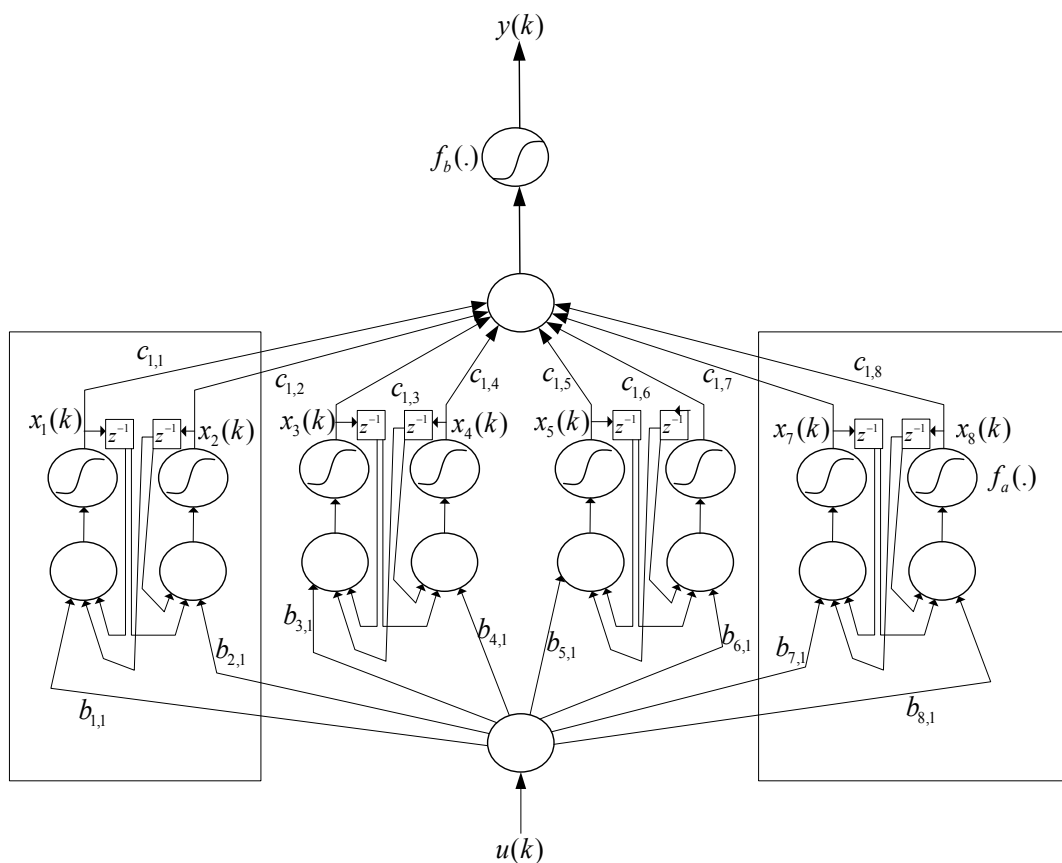
Βιβλιογραφία	313
---------------------	-----

ΚΕΦΑΛΑΙΟ 1

Περιγραφή του μοντέλου

1.1 Το αναδρομικό νευρωνικό δίκτυο με διαγώνιο πίνακα βαρών

Το αναδρομικό νευρωνικό δίκτυο με διαγώνιο πίνακα βαρών (block-diagonal recurrent neural network, BDRNN) είναι ένα δίκτυο δύο στρωμάτων, με το στρώμα εξόδου να είναι στατικό και το κρυμμένο στρώμα δυναμικό. Το κρυφό στρώμα αποτελείται από ζεύγη νευρώνων (μπλοκ), τα οποία έχουν αναδρομικές συνάψεις. Με τον τρόπο αυτό εισάγεται δυναμική στο σύστημα. Η δομή του BDRNN παρουσιάζεται στο σχήμα 1, όπου χάριν ευκολίας εμφανίζεται η περίπτωση BDRNN με μία είσοδο, μία έξοδο και τέσσερα ζεύγη νευρώνων στο κρυμμένο στρώμα.



Η λειτουργία του BDRNN με m εισόδους, r εξόδους και N νευρώνια στο κρυμμένο στρώμα περιγράφεται από το ακόλουθο σύνολο εξισώσεων κατάστασης:

$$\mathbf{x}(k) = \underline{f}_a(W \cdot \mathbf{x}(k-1) + B \cdot \mathbf{u}(k)) \quad (1\alpha)$$

$$\mathbf{y}(k) = \underline{f}_b(C \cdot \mathbf{x}(k)) \quad (1\beta)$$

όπου

• $\underline{f}_a, \underline{f}_b$ είναι διανύσματα N στοιχείων, τα οποία περιλαμβάνουν τις συναρτήσεις ενεργοποίησης των νευρώνων του κρυμμένου στρώματος και του στρώματος εξόδου, αντίστοιχα. Στη συνέχεια της εργασίας ως συναρτήσεις \underline{f}_a και \underline{f}_b χρησιμοποιούνται σιγμοειδείς συναρτήσεις $f(z) = \frac{1 - e^{-a_n \cdot z}}{1 + e^{-a_n \cdot z}}$.

• $\mathbf{u}(k) = [u_i(k)]$ είναι ένα διάνυσμα m στοιχείων, αποτελούμενο από τις εισόδους του δικτύου κατά τη χρονική στιγμή k .

• $\mathbf{x}(k) = [x_i(k)]$ είναι ένα διάνυσμα N στοιχείων, αποτελούμενο από τις εξόδους του κρυμμένου στρώματος. Ειδικότερα, $x_i(k)$ είναι η έξοδος του i -στού κρυμμένου νευρονίου κατά τη χρονική στιγμή k .

• $\mathbf{y}(k) = [y_i(k)]$ είναι ένα διάνυσμα r στοιχείων, αποτελούμενο από τις εξόδους του δικτύου κατά τη χρονική στιγμή k .

• $B = [b_{i,j}]$ και $C = [c_{l,j}]$ είναι οι πίνακες βαρών εισόδου και εξόδου, με διαστάσεις $N \times m$ και $r \times N$ αντίστοιχα.

• $W = [w_{i,j}]$ είναι ο διαγώνιος πίνακας των αναδράσεων, διαστάσεων $N \times N$. Ειδικότερα,

$$w_{i,j} = \begin{cases} \neq 0 & \text{ΕΑΝ } i = j \\ \neq 0 & \text{ΕΑΝ } i \neq j \text{ ΚΑΙ } i = j - 1 \text{ ΚΑΙ ΤΟ } i \text{ ΕΙΝΑΙ ΠΕΡΙΤΤΟ} \\ \neq 0 & \text{ΕΑΝ } i \neq j \text{ ΚΑΙ } i = j + 1 \text{ ΚΑΙ ΤΟ } i \text{ ΕΙΝΑΙ ΑΡΤΙΟ} \\ 0 & \text{ΣΕ ΑΝΤΙΘΕΤΗ ΠΕΡΙΠΤΩΣΗ} \end{cases}$$

Ο πίνακας των αναδράσεων, W , είναι διαγώνιος κατά ζεύγη (block diagonal): $W = \text{diag} \left[W^{(1)}, \dots, W^{(N/2)} \right]$.

Κάθε διαγώνιο στοιχείο, που αντιστοιχεί σε ένα ζεύγος αναδρομικών νευρωνίων, έχει ένα διαγώνιο υποπίνακα της μορφής

$$W^{(i)} = \begin{bmatrix} w_{2i,2i} & w_{2i,2i+1} \\ w_{2i+1,2i} & w_{2i+1,2i+1} \end{bmatrix} \quad i = 1, 2, \dots, \frac{N}{2} \quad (2)$$

Οι ανωτέρω εξισώσεις περιγράφουν τη λειτουργία της γενικής μορφής των BDRNN, η οποία καλούνται *BDRNN με υποπίνακας ελεύθερης (γενικής) μορφής*. Μία ειδική κατηγορία BDRNN αποτελούν οι *κλιμακωτά ορθογώνιοι υποπίνακες* (scaled orthogonal submatrices), οι οποίοι λαμβάνουν την ακόλουθη μορφή

$$W^{(i)} = \begin{bmatrix} w_{2i,2i} & w_{2i,2i+1} \\ -w_{2i,2i+1} & w_{2i,2i} \end{bmatrix} = \begin{bmatrix} w_i^{(1)} & w_i^{(2)} \\ -w_i^{(2)} & w_i^{(1)} \end{bmatrix} \quad i = 1, 2, \dots, \frac{N}{2} \quad (3)$$

Από τις (2) και (3) καθίσταται φανερό ότι τα BDRNN ελεύθερης μορφής αποτελούνται από υποπίνακες ανάδρασης με τέσσερα διακριτά στοιχεία, παρέχοντας με αυτόν τον τρόπο μεγαλύτερο βαθμό ελευθερίας συγκριτικά με τα κλιμακωτά ορθογώνια BDRNN, τα οποία έχουν δύο στοιχεία (βάρη) σε κάθε υποπίνακα ανάδρασης. Ωστόσο, όπως αποδεικνύεται στη βιβλιογραφία, τα τελευταία δίκτυα εμφανίζουν ανώτερες δυνατότητες μοντελοποίησης σε σύγκριση με τα BDRNN ελεύθερης μορφής. Έτσι, θα χρησιμοποιηθεί αυτή η ιδιαίτερη κατηγορία, επί της οποίας θα αναπτυχθεί η μέθοδος εκπαίδευσης. Παρά ταύτα ο αλγόριθμος είναι γενικός και άμεσα εφαρμόσιμος σε BDRNN ελεύθερης μορφής, μετά από ελάχιστες τροποποιήσεις.

Με βάση τα παραπάνω οι εξισώσεις κατάστασης (1) μπορούν να λάβουν την ακόλουθη μορφή για το κλιμακωτά ορθογώνιο BDRNN:

$$x_{2i-1}(k) = f_a \left(\sum_{j=1}^m b_{2i-1,j} \cdot u_j(k) + w_i^{(1)} \cdot x_{2i-1}(k-1) + w_i^{(2)} \cdot x_{2i}(k-1) \right) \quad i = 1, \dots, \frac{N}{2} \quad (4\alpha)$$

$$x_{2i}(k) = f_a \left(\sum_{j=1}^m b_{2i,j} \cdot u_j(k) - w_i^{(2)} \cdot x_{2i-1}(k-1) + w_i^{(1)} \cdot x_{2i}(k-1) \right) \quad i = 1, \dots, \frac{N}{2} \quad (4\beta)$$

$$y_l(k) = f_b \left(\sum_{j=1}^N c_{l,j} \cdot x_j(k) \right) \quad l = 1, \dots, r \quad (4\gamma)$$

όπου $w_i^{(1)}$, $w_i^{(2)}$ είναι τα βάρη ανάδρασης του κρυμμένου στρώματος.

Ο αλγόριθμος εκπαίδευσης RENNCOM

1.2 Ορισμοί και διατύπωση του προβλήματος

Η μέθοδος RENNCOM στοχεύει στο μετασχηματισμό της διαδικασίας εκπαίδευσης σε ένα πρόβλημα βελτιστοποίησης με περιορισμούς, το οποίο επιλύθηκε στο παρελθόν από τους Bryson και Denham, βασιζόμενοι σε μεθόδους της θεωρίας βέλτιστου ελέγχου και του λογισμού των μεταβολών. Με βάση αυτές τις αρχές, επινοήθηκε από τους Μαστοροκόστα και Θεοχάρη ένας αλγόριθμος εκπαίδευσης για αναδρομικά νευρωνικά μοντέλα, καλούμενος μέθοδος D-FUNCOM (Dynamic Fuzzy Neural Constrained Optimization Method). Το προτεινόμενο σχήμα εκμάθησης μοιράζεται την ίδια αναλυτική σημειογραφία με τη μέθοδο D-FUNCOM, με την κύρια διαφορά να έγκειται στο γεγονός ότι, πέραν από το ζήτημα της αποδοτικής εκμάθησης, ο αλγόριθμος RENNCOM δύναται να λάβει υπόψη του και το ζήτημα της ευσταθούς διαδικασίας εκμάθησης.

Ας θεωρήσουμε ένα BDRNN με m εισόδους, r εξόδους και κλιμακωτά ορθογώνιους υποπίνακες ανάδρασης. Η διαδικασία εκπαίδευσης αποσκοπεί στην προσαρμογή των παραμέτρων (βαρών) του δικτύου κατά τρόπο ώστε να επιτευχθεί ακριβής προσέγγιση της πραγματικής απεικόνισης εισόδου-εξόδου. Χρησιμοποιείται η ιδιαίτερα απαιτητική παράλληλη εκμάθηση (learning in parallel mode), δηλαδή σε κάθε χρονική στιγμή πληροφορία λαμβάνεται μόνο από τις εισόδους και όχι από τις πραγματικές εξόδους προηγούμενων χρονικών στιγμών. Το σύνολο των δεδομένων εκπαίδευσης λαμβάνει τη μορφή $\{(\mathbf{u}(k), \hat{\mathbf{y}}(k)), k = 1, \dots, k_f\}$, αποτελούμενο από k_f ζεύγη δεδομένων εισόδου-εξόδου. Αρχικά εισάγουμε:

a) το διάνυσμα κατάστασης $\mathbf{st}(t)$, οριζόμενο ως:

$$\mathbf{st}(k) = [x_1(k), \dots, x_N(k), y_1(k), \dots, y_r(k)]^T \quad (12)$$

το οποίο αποτελείται από τις εξόδους του κρυμμένου στρώματος και του στρώματος εξόδου.

b) το διάνυσμα ελέγχου $\boldsymbol{\theta}$ αποτελούμενο από τα βάρη των συνάψεων και των αναδράσεων.

$$\boldsymbol{\theta} = \left[b_{1,1}, \dots, b_{1,m}, \dots, b_{N,1}, \dots, b_{N,m}, w_1^{(1)}, w_N^{(1)}, w_1^{(2)}, w_N^{(2)}, c_{1,1}, \dots, c_{r,N} \right]^T \quad (N \times (m+r+1) \text{ weights}) \quad (13)$$

Για σύνολο δεδομένων αποτελούμενο από k_f ζεύγη, οι εξισώσεις κατάστασης γράφονται

$$\mathbf{f}(\mathbf{st}(k), \boldsymbol{\theta}(k)) = \mathbf{0}, \quad k = 1, \dots, k_f \quad (14)$$

$$f_{2i-1}^{(1)}(k) \quad i = 1, \dots, \frac{N}{2}: \quad f_a \left(\sum_{j=1}^m b_{2i-1,j} u_j(k) + w_i^{(1)} x_{2i-1}(k-1) + w_i^{(2)} x_{2i}(k-1) \right) - x_{2i-1}(k) = 0$$

($k_f \times \frac{N}{2}$ εξισώσεις) (15α)

$$f_{2i}^{(1)}(k) \quad i = 1, \dots, \frac{N}{2}: \quad f_a \left(\sum_{j=1}^m b_{2i,j} u_j(k) - w_i^{(2)} x_{2i-1}(k-1) + w_i^{(1)} x_{2i}(k-1) \right) - x_{2i}(k) = 0$$

($k_f \times \frac{N}{2}$ εξισώσεις) (15β)

$$f_l^{(2)}(k) \quad l = 1, \dots, r: \quad f_b \left(\sum_{j=1}^N c_{l,j} x_j(k) \right) - y_l(k) = 0 \quad (k_f \times r \text{ εξισώσεις}) \quad (15\gamma)$$

Ο αλγόριθμος είναι μία επαναληπτική διαδικασία, η οποία αποσκοπεί στην ταυτόχρονη επίτευξη των ακόλουθων δύο στόχων:

- Επιδιώκεται η ελαχιστοποίηση μίας συνάρτησης σφάλματος E , έτσι ώστε να επιτευχθεί η ζητούμενη απεικόνιση εισόδου-εξόδου. Ως μέτρο σφάλματος επιλέγεται το μέσο τετραγωνικό σφάλμα, οριζόμενο ως

$$E = \frac{1}{k_f} \sum_{k=1}^{k_f} \sum_{l=1}^r [y_l(k) - \hat{y}_l(k)]^2 \quad (16)$$

όπου $y_l(k)$ είναι η l -στή έξοδος του μοντέλου και $\hat{y}_l(k)$ είναι η l -στή επιθυμητή (πραγματική) έξοδος του συστήματος κατά τη χρονική στιγμή k .

- Επιδιώκεται η βελτιστοποίηση μίας μη γραμμικής συνάρτησης πολλών μεταβλητών Φ , η οποία καλείται συνάρτηση προσθήκης. Στην παρούσα περίπτωση ο κύριος στόχος είναι η διατήρηση της ευστάθειας καθόλη τη διάρκεια της διαδικασίας εκμάθησης. Έτσι, η συνάρτηση προσθήκης θα πρέπει να ενσωματώνει τους περιορισμούς που τίθενται από τις (11) and (12). Μία κατάλληλη συνάρτηση προσθήκης είναι

$$\Phi_s = \frac{1}{2} \sum_{i=1}^{\frac{N}{2}} (-1 - g_i^{(s)})^2 \quad (17\alpha)$$

η οποία θα πρέπει να ελαχιστοποιείται σε κάθε επανάληψη. Η ελαχιστοποίηση της Φ_s σημαίνει ότι τα βάρη του δικτύου θα ενημερώνονται έτσι ώστε οι ιδιοτιμές των υποπινάκων να κείνται εντός κύκλου ακτίνας $\rho_1 < 1$. Επιπρόσθετα, εφόσον οι ιδιοτιμές κινηθούν σε μία ευσταθή περιοχή και ο ανωτέρω περιορισμός πληρούται, η Φ_s δεν είναι πλέον απαραίτητη. Σε αυτήν την περίπτωση μπορεί να αντικατασταθεί με μία άλλη συνάρτηση,

στόχος της οποίας να είναι η επιτάχυνση της διαδικασίας εκμάθησης. Επιλέγεται η ακόλουθη συνάρτηση προσθήκης

$$\Phi_a = (\theta - \theta_{cur})^T (\theta_{cur} - \theta_{prev}) \quad (17\beta)$$

όπου θ_{cur} , θ_{prev} είναι τα διανύσματα των παραμέτρων ελέγχου κατά την τρέχουσα και την προηγούμενη επανάληψη. Μεγιστοποίηση αυτής της συνάρτησης προσθήκης σημαίνει ότι σε κάθε επανάληψη το τρέχον και το προηγούμενο διάνυσμα ελέγχου θα είναι κατά το δυνατόν ευθυγραμμισμένα, αποφεύγοντας έτσι ταλαντώσεις στο χώρο των παραμέτρων ελέγχου και βελτιώνοντας την ταχύτητα σύγκλισης, όπως αναφέρεται στη βιβλιογραφία.

Με βάση τα παραπάνω, η συνολική συνάρτηση προσθήκης περιγράφεται ως εξής:

$$\Phi = \begin{cases} \Phi_a & \text{ΕΑΝ ΟΙ ΙΔΙΟΤΗΤΕΣ ΚΕΙΝΤΑΙ ΕΝΤΟΣ ΤΗΣ ΕΥΣΤΑΘΟΥΣ ΠΕΡΙΟΧΗΣ} \\ \Phi_s & \text{ΣΤΗΝ ΑΝΤΙΘΕΤΗ ΠΕΡΙΠΤΩΣΗ} \end{cases} \quad (18)$$

Η βελτιστοποίηση των συναρτήσεων (16) και (18) διεξάγεται επαναληπτικά, ως προς τα βάρη θ και τις εξισώσεις κατάστασης st , υπό τους αρχιτεκτονικούς περιορισμούς που θέτουν οι εξισώσεις κατάστασης (15). Υπό αυτή την έννοια η διαδικασία εκπαίδευσης μπορεί να θεωρηθεί ως ένα πρόβλημα βελτιστοποίησης υπό περιορισμούς.

1.2.1 Ενημέρωση των παραμέτρων

Σε κάθε επανάληψη οι μεταβλητές ελέγχου μεταβάλλονται κατά μικρές ποσότητες $d\theta$, έτσι ώστε να ισχύει η ακόλουθη τετραγωνική μορφή:

$$\sum_{i=1}^n \frac{(d\theta_i)^2}{\Delta_i^2} = 1 \quad \text{ή} \quad d\theta^T (\Delta^2)^{-1} d\theta = 1 \quad (19)$$

όπου Δ_i καλείται η μέγιστη μεταβολή παραμέτρου (maximum parameter change, MPC) για τη θ_i και $\Delta = \text{diag} (\Delta_1, \Delta_2, \dots, \Delta_{N \times (m+r+1)})$ είναι ο πίνακας MPCx. Η εξίσωση (19) περιγράφει ένα υπερελλειψοειδές, με το κέντρο του να βρίσκεται στο τρέχον διάνυσμα ελέγχου και οι άξονές του να είναι οι Δ_i . Οι μεταβολές του διανύσματος ελέγχου προκαλούν αλλαγές στις μεταβλητές κατάστασης st και στις συναρτήσεις E και Φ . Θεωρώντας ότι οι MPC είναι αρκετά μικρές, οι αλλαγές αυτές μπορούν να προσεγγισθούν από τα διαφορικά dE και $d\Phi$. Η μέθοδος επιδιώκει να βρει τις βέλτιστες μεταβολές των παραμέτρων $d\theta$, οι οποίες να μεγιστοποιούν το $d\Phi$, για δεδομένη τιμή του dE . Οι αλλαγές αυτές δίνονται από τις ακόλουθες σχέσεις, ενώ πιο λεπτομερής περιγραφή και εξαγωγή των εξισώσεων φιλοξενείται στο Παράρτημα:

$$d\theta = -\Delta^2 \cdot \sqrt{\frac{I_{EE} - (\delta E)^2}{I_{\Phi\Phi} \cdot I_{EE} - I_{E\Phi}^2}} \cdot \left[\mathbf{A}_{\Phi}^T - (\mathbf{A}_E)^T \cdot \frac{I_{E\Phi}}{I_{EE}} \right] + \Delta^2 \cdot (\mathbf{A}_E)^T \cdot \frac{\delta E}{I_{EE}} \quad (20)$$

όπου

$$\mathbf{A}_{\Phi} = \left(\frac{\partial \Phi}{\partial \theta} \right)^T, \quad \mathbf{A}_E = \left(\lambda^T \cdot \frac{\partial f}{\partial \theta} \right)^T \quad (21)$$

$$I_{EE} = \mathbf{A}_E \Delta^2 \mathbf{A}_E^T, \quad I_{\Phi\Phi} = \mathbf{A}_\Phi \Delta^2 \mathbf{A}_\Phi^T, \quad I_{E\Phi} = \mathbf{A}_\Phi \Delta^2 \mathbf{A}_E^T, \quad (22)$$

$$\text{και } \delta E = -\xi \cdot \sqrt{I_{EE}} \quad (23)$$

με τα λ να είναι οι πολλαπλασιαστές Lagrange, οριζόμενοι στο Παράρτημα, και ξ σταθερά στο διάστημα $[0,1]$. Έχοντας υπολογίσει τον όρο διόρθωσης $d\theta$, οι παράμετροι ελέγχου (βάρη) ενημερώνονται ως εξής:

$$\theta(t+1) = \theta(t) + d\theta \quad (24)$$

όπου t είναι ο δείκτης της επανάληψης.

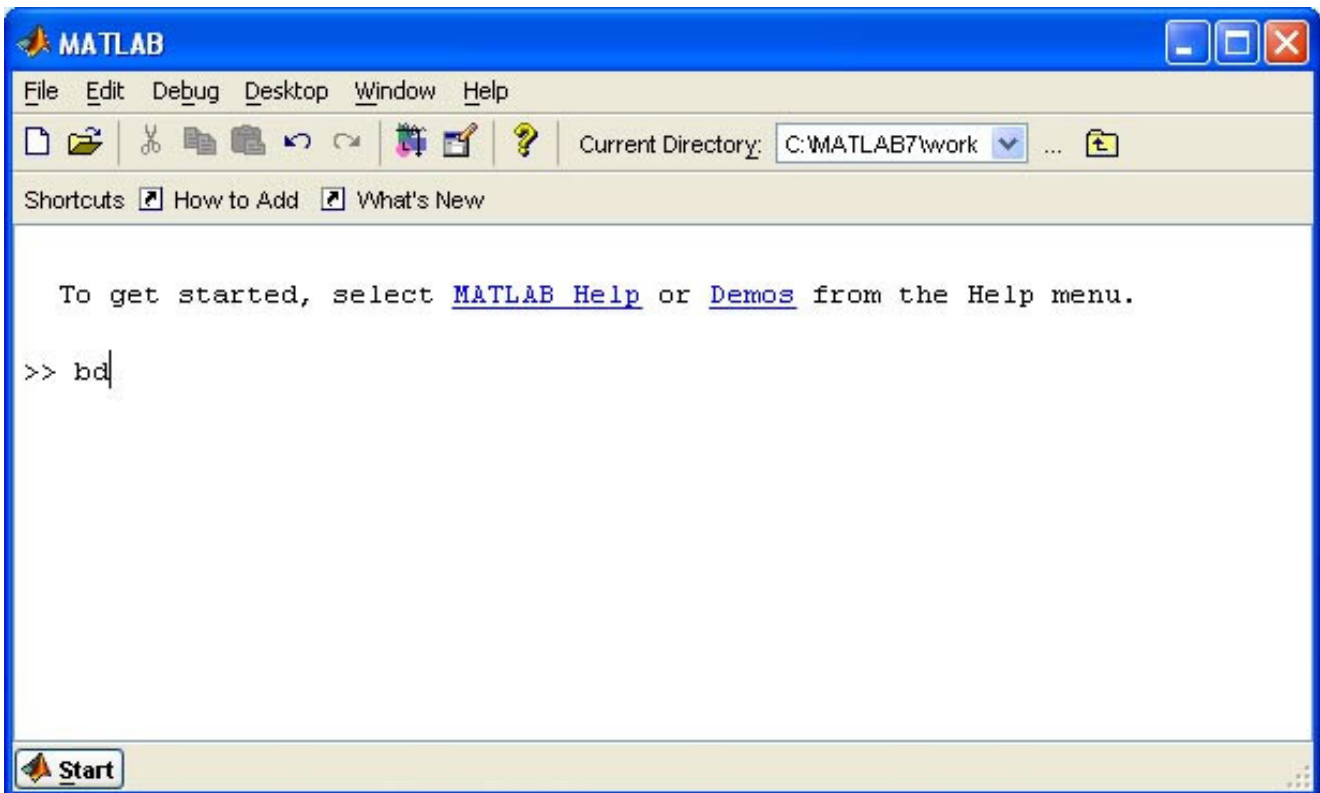
Όπως σημειώνεται παραπάνω, σε κάθε επανάληψη τα βάρη μεταβάλλονται ανεξάρτητα το ένα από το άλλο, κάτω από τους περιορισμούς $|d\theta_i| \leq \Delta_i$. Εφόσον ο χώρος αναζήτησης καθορίζεται από τις Δ_i , οι MPC διαδραματίζουν ένα σημαντικό ρόλο στη διαδικασία εκμάθησης και επιδιώκεται να μεταβάλλονται καθόλη τη διάρκεια της διαδικασίας. Έτσι υιοθετείται ο μηχανισμός προσαρμογής των βαρών που περιγράφεται στην αναφορά [A9] για τον καθορισμό των Δ_i σε κάθε επανάληψη.

Σε κάθε επανάληψη $t=1,2,\dots$ ο αλγόριθμος RENNCOM λειτουργεί ως εξής: Πρώτα υπολογίζονται οι ιδιοτιμές. Στην περίπτωση που υπάρχει αστάθεια ενεργοποιείται η Φ_s , αλλιώς η συνάρτηση προσθήκης είναι η $\Phi = \Phi_a^{-1}$. Ακολούθως εξάγονται οι τρέχουσες τιμές των Δ_i και μετά γίνεται η ενημέρωση των βαρών με χρήση των (19)-(24). Εάν το σφάλμα καταστεί μικρότερο του δοθέντος ορίου η διαδικασία τερματίζεται αλλιώς προχωράμε στην επόμενη επανάληψη.

ΚΕΦΑΛΑΙΟ 2

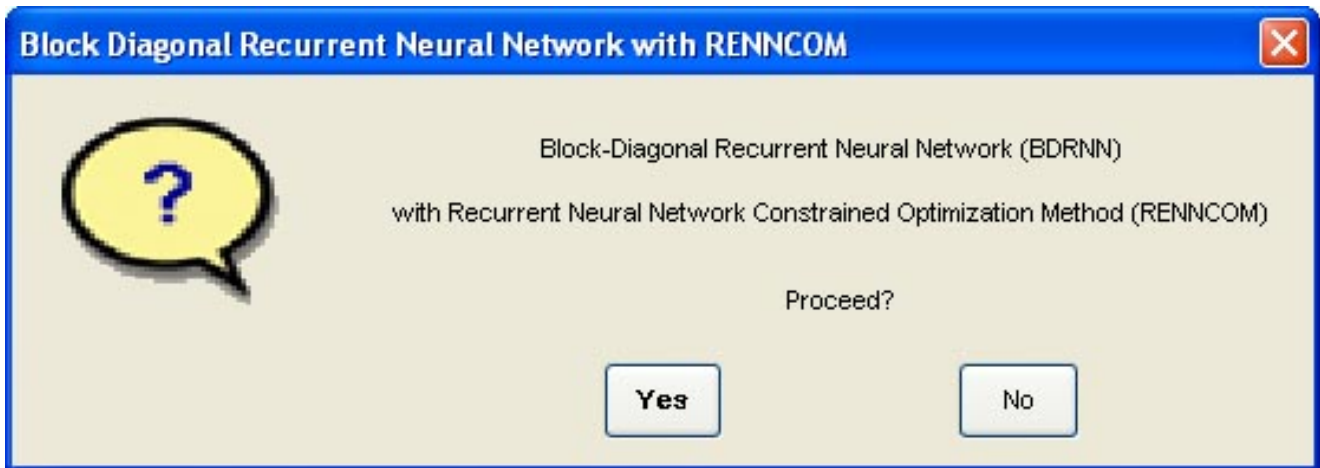
2.1 ΠΕΡΙΓΡΑΦΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

Για την εκτέλεση του προγράμματος, πληκτρολογούμε στη γραμμή εντολών του Matlab την εντολή `bd`.

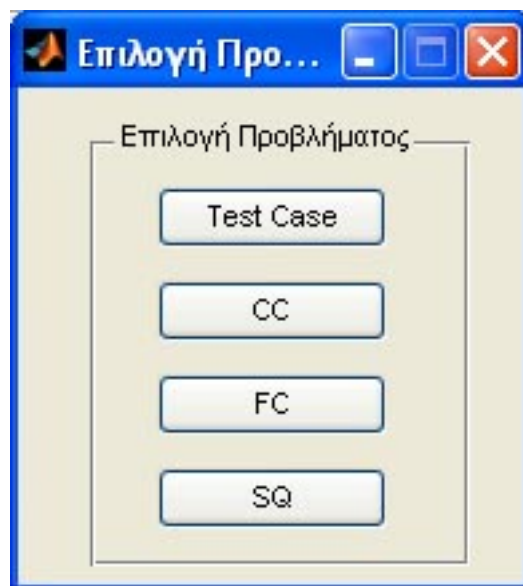


Στην οθόνη εμφανίζεται η αρχική φόρμα του προγράμματος, που μας ενημερώνει για το είδος του νευρωνικού δικτύου και του αλγορίθμου εκμάθησης.

¹ Στην παρούσα εργασία υλοποιείται μόνο το δεύτερο τμήμα της συνάρτησης προσθήκης, καθώς το ζήτημα της ευσταθούς εκμάθησης δεν αποτελεί αντικείμενο της μελέτης.



Στην περίπτωση που επιλέξουμε να μην προχωρήσουμε και πατήσουμε No τότε το πρόγραμμα τερματίζεται, αλλιώς ακολουθεί η παρακάτω φόρμα.



Σε αυτήν τη φόρμα ο χρήστης καλείται να επιλέξει το είδος του προβλήματος που θέλει να επιλυθεί. Αν επιλέξει την πρώτη περίπτωση (Test Case) θα εμφανιστεί η ακόλουθη φόρμα.

BDRNN for Test Case

Structural Features

in_num	1
out_num	1
n_hid	12
n2	6

Data Set

samples	4096
tst_samples	3072
max_iter	1500
ep_sz	270
max_trials	25

Learning Parameters

ksi	0.9
n_plus_w	1.05
n_minus_w	0.5
dmin_w	0.0001
dmax_w	0.5
d0_w	0.02

Sigmoid's Parameters

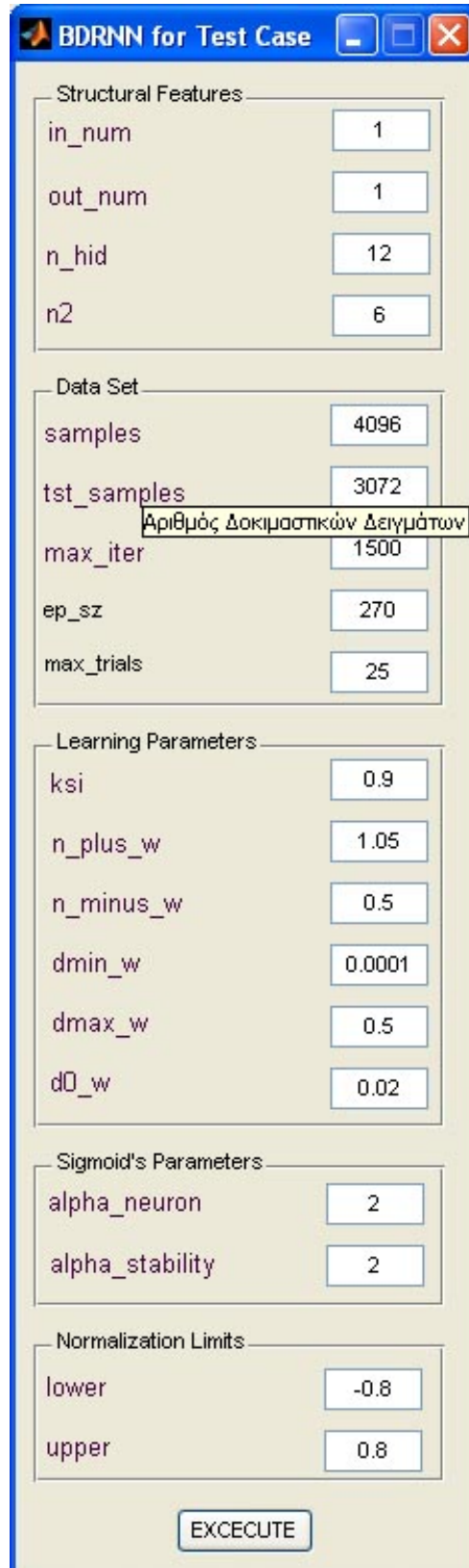
alpha_neuron	2
alpha_stability	2

Normalization Limits

lower	-0.8
upper	0.8

EXCECUTE

Σε αυτήν τη φόρμα ο χρήστης καλείται να συμπληρώσει ή να αλλάξει τις προεπιλεγμένες τιμές των παραμέτρων που έχουν οριστεί και καθορίζουν τη συμπεριφορά του νευρωνικού δικτύου. Αν ο χρήστης δε γνωρίζει τη σημασία της κάθε παραμέτρου, αφήνοντας το δείκτη του ποντικιού πάνω από το πεδίο εισαγωγής εμφανίζεται η επεξήγησή του, όπως φαίνεται στην παρακάτω εικόνα.



BDRNN for Test Case

Structural Features

in_num	1
out_num	1
n_hid	12
n2	6

Data Set

samples	4096
tst_samples	3072
max_iter	1500
ep_sz	270
max_trials	25

Learning Parameters

ksi	0.9
n_plus_w	1.05
n_minus_w	0.5
dmin_w	0.0001
dmax_w	0.5
d0_w	0.02

Sigmoid's Parameters

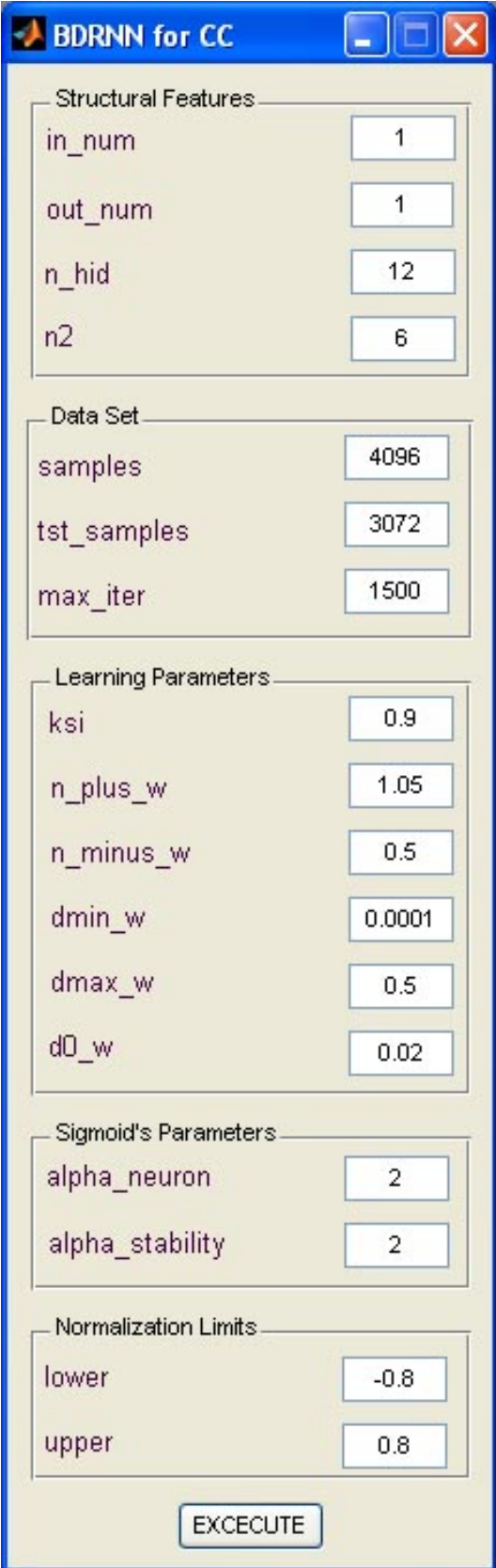
alpha_neuron	2
alpha_stability	2

Normalization Limits

lower	-0.8
upper	0.8

EXCECUTE

Αφού συμπληρωθούν όλα τα πεδία με τις απαραίτητες τιμές, ο χρήστης καλείται να πατήσει το κουμπί EXCECUTE ώστε να αρχίσει η εκτέλεση του προγράμματος. Στη συνέχεια παρατίθεται η φόρμα για την περίπτωση CC, η οποία είναι ίδια και για τις υπόλοιπες δυο περιπτώσεις των FC και SQ αντίστοιχα.



The image shows a software window titled "BDRNN for CC". The window contains several sections of configuration parameters, each with a label and a corresponding input field:

- Structural Features**
 - in_num: 1
 - out_num: 1
 - n_hid: 12
 - n2: 6
- Data Set**
 - samples: 4096
 - tst_samples: 3072
 - max_iter: 1500
- Learning Parameters**
 - ksi: 0.9
 - n_plus_w: 1.05
 - n_minus_w: 0.5
 - dmin_w: 0.0001
 - dmax_w: 0.5
 - d0_w: 0.02
- Sigmoid's Parameters**
 - alpha_neuron: 2
 - alpha_stability: 2
- Normalization Limits**
 - lower: -0.8
 - upper: 0.8

At the bottom of the window, there is a large button labeled "EXCECUTE".

Ανάλογα με το πρόβλημα που θα επιλέξει ο χρήστης, θα πρέπει να γνωρίζει ότι εκτός από τη συμπλήρωση των τιμών θα πρέπει στο τοπικό δίσκο *C:* να υπάρχουν συγκεκριμένοι φάκελοι και αρχεία, που είναι απαραίτητα για τη σωστή εκτέλεση του προγράμματος πριν πατηθεί το κουμπί EXCECUTE. Για την περίπτωση Test Case οι φάκελοι είναι: *C:\mat_inI*, ο οποίος χρησιμοποιείται ως φάκελος για τα αρχεία εισόδου του προγράμματος και στον οποίο θα πρέπει να υπάρχουν τα εξής αρχεία: *dtrn.dat* και *dtst.dat*, όπως επίσης θα πρέπει να υπάρχει και ο φάκελος *C:\mat_tempI*, στον οποίο θα αποθηκεύονται τα αρχεία εξόδου του προγράμματος, ενώ για τις τρεις άλλες περιπτώσεις η διαφορά έγκειται στα αρχεία που θα βρίσκονται στο φάκελο *C:\mat_inI*, τα οποία θα είναι *mat_ccnst_trn.dat*, *mat_ccnst_tst.dat*, *mat_dparams.ini* για την περίπτωση του CC, *mat_fcnst_trn.dat*, *mat_fcnst_tst.dat*, *mat_dparams.ini* για την περίπτωση του FC και *mat_sqnst_trn.dat*, *mat_sqnst_tst.dat*, *mat_dparams.ini* για την περίπτωση του SQ.

Ανάλυση κώδικα

2.2 Περιγραφή κώδικα για την περίπτωση test case

```
function
x=statistical_scale(tmp2,tmp3,tmp4,tmp5,tmp6,tmp7,tmp9,tmp10,tmp11,tmp12,tmp13,tmp16,tmp17,tmp18,tmp21,tmp22,tmp23,tmp24,
tmp25)
clc;
% BLOCK-DIAGONAL RECURRENT NEURAL NETWORK (BDRNN),
% according to A172, trained with RENNCOM.
% Test Case

global eigen;global p1;global p2;global B;global W1;global W2;global C;global B_prev;
global W1_prev;global W2_prev;global C_prev;global state;global output_trn;
global output_tst;global outpat_trn;global outpat_tst;global u_trn;global u_tst;
global Lw;global Ly;global state_init;global mini_in_trn;global mini_in_tst;
global maxi_in_trn;global maxi_in_tst;global mini_out_trn;global mini_out_tst;
global maxi_out_trn;global maxi_out_tst;global gradprev_B;global gradprev_W1;
global gradprev_W2;global gradprev_C;global stabprev_W1;global stabprev_W2;
global delprev_B;global delprev_W1;global delprev_W2;global delprev_C;
global dwprev_B;global dwprev_W1;global dwprev_W2;global dwprev_C;global delta_B;
global delta_W1;global delta_W2;global delta_C;global dw_B;global dw_W1;
global dw_W2;global dw_C;global kf_meg;global ssas_choice;global trial_index;
global in_num;global out_num;global n_hid;global n2;global alpha_neuron;
global alpha_stability;global eigen_limit;global ksi;global kf_trn;global kf_tst;
global ep_sz;global max_iter;global limit_B_C;global limit_W;global max_trials;
global lower;global upper;global pi;global ZERO;global n_plus_w;global n_minus_w;
global dmin_w;global dmax_w;global d0_w;global stam_counter;global gstam1;global gstam2;global gstam3;global gstam4;
global gstam5;global gstam6;global gstam7;global tot_error;global ga1;
global ga2;global ga3;global ga4;global ga5;global ga6;global ga7;
global a1;global a2;global a3;global a4;global a5;global a6;global a7;
global stam1;global stam2;global stam3;global stam4;
global stam5;global stam6;global stam7;

%---Μεταβλητές από gui---
in_num=tmp2;
out_num=tmp3;
n_hid=tmp4;
n2=tmp5;
alpha_neuron=tmp6;
alpha_stability=tmp7;
ksi=tmp9;
kf_trn=tmp10;
kf_tst=tmp11;
ep_sz=tmp12;
max_iter=tmp13;
```

```

max_trials=tmp16;
lower=tmp17;
upper=tmp18;
n_plus_w=tmp21;
n_minus_w=tmp22;
dmin_w=tmp23;
dmax_w=tmp24;
d0_w=tmp25;
%-----

which_p2=1.0; % 1.0// -1.0

p1=0;
which_p2=tmp1;
p2=which_p2;
kf_meg=maximum(ep_sz,kf_tst);
normalize_data();
for trial_index=1:max_trials,
    initialization();
    training_process();
end

end
%-----

function x=maximum(x1,x2)

if x1>x2
    x=x1;
else x=x2;
end
end

function x=minimum(x1,x2)

if x1<x2
    x=x1;
else x=x2;
end
end

function x=my_exp(y)
if y>200
    fprintf('Argument to exp() =%f\n',y);
elseif y < -20
    x=0;
else x=exp(y);
end
end

function x=sqr(x1)
x=x1*x1;
end

function x=sgn(x1)
if x1<0
    x=-1;
else
    x=1;
end
end

function x=sigmoid(x1,alpha)
x=((1.0-my_exp(-alpha*x1))/(1.0+my_exp(-alpha*x1)));
end

%Παράγωγος σιγμοειδούς ως προς όρισμα (έξοδος αθροιστή)

```



```

function x=der_sigmoid(x1,alpha)
x=((2*alpha*my_exp(-alpha*x1))/sqrt(1.0+my_exp(-alpha*x1)));
end

%Παράγωγος σιγμοειδούς ως προς την είσοδο στο κρυμμένο στρώμα
function x=der_fa(kk,ii)
global B;global W1;global W2;global state;global u_trn;global state_init;
global in_num;global alpha_neuron;

j=0;l=0;deikt=0;x1=0;

if mod(ii,2)==0
    deikt=ii/2;

    for j=1:in_num,
        x1=x1+B(ii,j)*u_trn(kk,j);
    end
    if (kk==1)
        x1=x1-W2(deikt)*state_init(ii-1)+W1(deikt)*state_init(ii);
    else
        x1=x1-W2(deikt)*state(kk-1,ii-1)+W1(deikt)*state(kk-1,ii);
    end
else
    deikt=(ii+1)/2;
    for j=1:in_num,
        x1=x1+B(ii,j)*u_trn(kk,j);
    end
    if kk==1
        x1=x1+W1(deikt)*state_init(ii)+W2(deikt)*state_init(ii+1);
    else
        x1=x1+W1(deikt)*state(kk-1,ii)+W2(deikt)*state(kk-1,ii+1);
    end
end
x=der_sigmoid(x1,alpha_neuron);
end

%Παράγωγος σιγμοειδούς ως προς την είσοδο στο στρώμα εξόδου
function x=der_fb(kk,ll)
global C;global state;global n2;global alpha_neuron;

x1=0;

for j=1:n2
    x1=x1+C(ll,2*j-1)*state(kk,2*j-1)+C(ll,2*j)*state(kk,2*j);
    x=der_sigmoid(x1,alpha_neuron);
end
end

function normalize_data()

global output_trn;global output_tst;global u_trn;global u_tst;
global Lw;global Ly;global state_init;global mini_in_trn;global mini_in_tst;
global maxi_in_trn;global maxi_in_tst;global mini_out_trn;global mini_out_tst;
global maxi_out_trn;global maxi_out_tst;global gradprev_B;global gradprev_W1;
global gradprev_W2;global gradprev_C;global stabprev_W1;global stabprev_W2;
global delprev_B;global delprev_W1;global delprev_W2;global delprev_C;
global dwprev_B;global dwprev_W1;global dwprev_W2;global dwprev_C;global delta_B;
global delta_W1;global delta_W2;global delta_C;global dw_B;global dw_W1;
global dw_W2;global dw_C;global kf_meg;global ssas_choice;global trial_index;
global in_num;global out_num;global n_hid;global n2;global alpha_neuron;
global alpha_stability;global eigen_limit;global ksi;global kf_trn;global kf_tst;
global ep_sz;global max_iter;global limit_B_C;global limit_W;global max_trials;
global lower;global upper;global pi;global ZERO;global n_plus_w;global n_minus_w;
global dmin_w;global dmax_w;global d0_w;global cc_dtrn;

u_tst=0;
output_tst=0;

```

```

%----Άνοιγμα αρχείου dtrn.dat-----
load c:\mat_in1\dtrn.dat;
for i=2:in_num+2,
    sthles_dtrn(:,i-1)=cc_dtrn(:,i);
end

%----Εύρεση max,min εισόδου-----
for i=1:in_num,
    max_in(:,i)=max(sthles_dtrn(:,i));
    min_in(:,i)=min(sthles_dtrn(:,i));
end

%---- Εύρεση max,min εξόδου-----
for i=in_num+1:out_num+in_num,
    max_out(:,i)=max(sthles_dtrn(:,i));
    min_out(:,i)=min(sthles_dtrn(:,i));
end
maxi_in_trn=max_in;
maxi_out_trn=max_out(:,in_num+1:in_num+out_num);
mini_in_trn=min_in;
mini_out_trn=min_out(:,in_num+1:in_num+out_num);

%----Κανονικοποίηση εισόδου-----
for j=1:in_num,
    for i=1:kf_trn,
        inp(i,j)=((lower-upper)/(min_in(:,j)-max_in(:,j))*sthles_dtrn(i,j))+((min_in(:,j)*upper-max_in(:,j)*lower)/(min_in(:,j)-max_in(:,j))));
    end
end

%----Κανονικοποίηση εξόδου -----
for j=in_num+1:out_num+in_num,
    for i=1:kf_trn,
        out(i,j)=(((lower-upper)/(min_out(:,j)-max_out(:,j))*sthles_dtrn(i,j))+((min_out(:,j)*upper-max_out(:,j)*lower)/(min_out(:,j)-max_out(:,j))));
    end
end

temp=out(:,in_num+1:in_num+out_num);
dnorm_trn=[inp temp];
save c:\mat_in1\dnorm_trn.dat dnorm_trn -ASCII;

%----Ανάγνωση και κανονικοποίηση των δοκιμαστικών δεδομένων -----
load c:\mat_in1\dtst.dat;
for i=2:in_num+2,
    sthles_dtst(:,i-1)=cc_dtst(:,i);
end

%---- Εύρεση max,min εισόδου -----
for i=1:in_num,
    max_in_tst(:,i)=max(sthles_dtst(:,i));
    min_in_tst(:,i)=min(sthles_dtst(:,i));
end

%---- Εύρεση max,min εξόδου ----
for i=in_num+1:out_num+in_num,
    max_out_tst(:,i)=max(sthles_dtst(:,i));
    min_out_tst(:,i)=min(sthles_dtst(:,i));
end

%----Δοκιμή των δεδομένων εισόδου-----
for j=1:in_num,
    for i=1:kf_tst,
        u_tst(i,j)=((upper-lower)*sthles_dtst(i,j)+max_in(:,j)*lower-min_in(:,j)*upper)/(max_in(:,j)-min_in(:,j));
    end
end

```

```
%----Δοκιμή των δεδομένων εξόδου----
for j=in_num+1:out_num+in_num,
    for i=1:kf_tst,
        temp1(i,j)=(upper-lower)*sthles_dtst(i,j)+max_out(:,j)*lower-min_out(:,j)*upper)/(max_out(:,j)-min_out(:,j));
    end
end
outpat_tst=temp1(:,in_num+1:in_num+out_num);
end
```

```
%Αρχικοποίηση των B,C,W1,W2 με τυχαίες τιμές
function x=initialization(n_hid,in_num,out_num,limit_W,limit_B_C,n2)
global eigen;global p1;global p2;global B;global W1;global W2;global C;global B_prev;
global W1_prev;global W2_prev;global C_prev;global state;global output_trn;
global output_tst;global outpat_trn;global outpat_tst;global u_trn;global u_tst;
global Lw;global Ly;global state_init;global mini_in_trn;global mini_in_tst;
global maxi_in_trn;global maxi_in_tst;global mini_out_trn;global mini_out_tst;
global maxi_out_trn;global maxi_out_tst;global gradprev_B;global gradprev_W1;
global gradprev_W2;global gradprev_C;global stabprev_W1;global stabprev_W2;
global delprev_B;global delprev_W1;global delprev_W2;global delprev_C;
global dwprev_B;global dwprev_W1;global dwprev_W2;global dwprev_C;global delta_B;
global delta_W1;global delta_W2;global delta_C;global dw_B;global dw_W1;
global dw_W2;global dw_C;global kf_meg;global ssas_choice;global trial_index;
global in_num;global out_num;global n_hid;global n2;global alpha_neuron;
global alpha_stability;global eigen_limit;global ksi;global kf_trn;global kf_tst;
global ep_sz;global max_iter;global limit_B_C;global limit_W;global max_trials;
global lower;global upper;global pi;global ZERO;global n_plus_w;global n_minus_w;
global dmin_w;global dmax_w;global d0_w;
```

```
sB=0;
sC=0;
sW=0;
```

```
for i=1:n_hid,
    state_init(i)=2*limit_W*rand()-limit_W;
end
```

```
for i=1:n_hid,
    for j=1:in_num,
        B(i,j)=2*limit_B_C*rand()-limit_B_C;
        sB=sB+sqr(B(i,j))/(n_hid*in_num);
    end
end
```

```
for i=1:out_num,
    for j=1:n_hid,
        C(i,j)=2*limit_B_C*rand()-limit_B_C;
        sC=sC+sqr(C(i,j))/(n_hid*out_num);
    end
end
```

```
fid=fopen('c:\mat_temp1\mat_dweights.dat','a+');
fprintf(fid,'0\t 0\t');
for i=1:n2,
    W1(i)=2*limit_W*rand()-limit_W;
    W2(i)=2*limit_W*rand()-limit_W;
    fprintf(fid,'%2.8ft %2.8ft %2.8ft %2.8ft',W1(i),W2(i));
    sW=sW+(sqr(W1(i))+sqr(W2(i)))/n_hid;
end
fclose(fid);
```

```
fid=fopen('c:\mat_temp1\mat_dall_wgh.dat','a+');
fprintf(fid,'0\t 0\t %2.8ft %2.8ft %2.8ft %2.8ft',sqrt(sB),sqrt(sC),sqrt(sW),sqrt(sC+sB+sW));
fclose(fid);
```

```
%Αρχικοποίηση των προηγούμενων βαρών
for i=1:n_hid,
```

```

    for j=1:in_num,
        B_prev(i,j)=0;
    end
end

for i=1:out_num,
    for j=1:n_hid,
        C_prev(i,j)=0;
    end
end

for i=1:n2,
    W1_prev(i)=0;
    W2_prev(i)=0;
end
end

function x=read_epoch_data(epoch_index)

global outpat_trn;global u_trn;global in_num;global out_num;global ep_sz;

j=0;k=0;

load c:\mat_in1\dnorm_trn.dat;
dnorm_trn_ep=cc_dnorm_trn;
if epoch_index==1
    u_trn=dnorm_trn_ep(1:ep_sz,in_num);
    outpat_trn=dnorm_trn_ep(1:ep_sz,in_num+1:in_num+out_num);
else
    u_trn=dnorm_trn_ep((epoch_index-1)*ep_sz+1:epoch_index*ep_sz,in_num);
    outpat_trn=dnorm_trn_ep((epoch_index-1)*ep_sz+1:epoch_index*ep_sz,in_num+1:in_num+out_num);
end
end

%Λειτουργία του RENNCOM
function x=model_operation()

global W1;global W2;global C;global state;global output_trn;global output_tst;
global u_trn;global u_tst;global state_init;global in_num;global out_num;
global n_hid;global n2;global alpha_neuron;global kf_tst;global ep_sz;global B;

for k=1:ep_sz,
    for i=1:n2,
        sum1=0;
        sum2=0;
        for j=1:in_num,
            sum1=sum1+B(2*i-1,j)*u_trn(k,j);
            sum2=sum2+B(2*i,j)*u_trn(k,j);
        end
        if k==1
            sum1=sum1+W1(i)*state_init(2*i-1)+W2(i)*state_init(2*i);
            sum2=sum2-W2(i)*state_init(2*i-1)+W1(i)*state_init(2*i);
        else
            sum1=sum1+W1(i)*state(k-1,2*i-1)+W2(i)*state(k-1,2*i);
            sum2=sum2-W2(i)*state(k-1,2*i-1)+W1(i)*state(k-1,2*i);
        end
        state(k,2*i-1)=sigmoid(sum1,alpha_neuron);
        state(k,2*i)=sigmoid(sum2,alpha_neuron);
    end
    for l=1:out_num,
        sum1=0;
        for j=1:n2,
            sum1=sum1+C(1,2*j-1)*state(k,2*j-1)+C(1,2*j)*state(k,2*j);
        end
        output_trn(k,l)=sigmoid(sum1,alpha_neuron);
    end
end

```

```

end
% Αρχικοποίηση για τη δοκιμαστική φάση και για την επόμενη εποχή
for j=1:n_hid,
state_init(j)=state(ep_sz,j);
end

for k=1:kf_tst,
for i=1:n2,
sum1=0;
sum2=0;
for j=1:in_num,
sum1=sum1+B(2*i-1,j)*u_tst(k,j);
sum2=sum2+B(2*i,j)*u_tst(k,j);
end
if k==1
sum1=sum1+W1(i)*state_init(2*i-1)+W2(i)*state_init(2*i);
sum2=sum2-W2(i)*state_init(2*i-1)+W1(i)*state_init(2*i);
else
sum1=sum1+W1(i)*state(k-1,2*i-1)+W2(i)*state(k-1,2*i);
sum2=sum2-W2(i)*state(k-1,2*i-1)+W1(i)*state(k-1,2*i);
end
state(k,2*i-1)=sigmoid(sum1,alpha_neuron);
state(k,2*i)=sigmoid(sum2,alpha_neuron);
end
for l=1:out_num,
sum1=0;
for j=1:n2,
sum1=sum1+C(1,2*j-1)*state(k,2*j-1)+C(1,2*j)*state(k,2*j);
end
output_tst(k,l)=sigmoid(sum1,alpha_neuron);
end
end
end

%Υπολογισμός των πολλαπλασιαστών Lagrange στη μέθοδο RENNCOM
function x=make_Lf(x1)

global W1;global W2;global C;global output_trn;global outpat_trn;global Lw;
global Ly;global out_num;global n_hid;global n2;global ep_sz;
% ---- k=kf ----
for l=1:out_num,
Ly(ep_sz,l)=(output_trn(ep_sz,l)-outpat_trn(ep_sz,l))/(ep_sz);
end
for i=1:n_hid,
x=0;
for l=1:out_num,
x=x+(C(1,i)*der_fb(ep_sz,l)*(output_trn(ep_sz,l)-outpat_trn(ep_sz,l)))/(ep_sz);
x=x+Ly(ep_sz,l)*C(1,i)*der_fb(ep_sz,l);
end
Lw(ep_sz,i)=x;
end
% ---- k=kf-1,...,1 ----
for k=(ep_sz-1):-1:1,
for l=1:out_num,
Ly(k,l)=(output_trn(k,l)-outpat_trn(k,l))/(ep_sz);
end
for i=1:n2,
x=0;
for l=1:out_num,
x=x+(C(1,2*i-1)*der_fb(k,l)*(output_trn(k,l)-outpat_trn(k,l)))/(ep_sz);
x=x+Ly(k,l)*C(1,2*i-1)*der_fb(k,l);
end
Lw(k,2*i-1)=x+Lw(k+1,2*i-1)*W1(i)*der_fa((k+1),(2*i-1))-Lw(k+1,2*i)*W2(i)*der_fa((k+1),(2*i));
x=0;
for l=1:out_num,
x=x+(C(1,2*i)*der_fb(k,l)*(output_trn(k,l)-outpat_trn(k,l)))/(ep_sz);
x=x+Ly(k,l)*C(1,2*i)*der_fb(k,l);

```

```

    end
    Lw(k,2*i)=x+Lw(k+1,2*i-1)*W2(i)*der_fa((k+1),(2*i-1))+Lw(k+1,2*i)*W1(i)*der_fa((k+1),(2*i));
end
end

end

%Εφαρμογή της μεθόδου RENNCOM
function x=RENNCOM(iterat,epoch_index,end_signal)

global eigen;global p1;global p2;global B;global W1;global W2;global C;global B_prev;
global W1_prev;global W2_prev;global C_prev;global state;global output_trn;
global output_tst;global outpat_trn;global outpat_tst;global u_trn;global u_tst;
global Lw;global Ly;global state_init;global mini_in_trn;global mini_in_tst;
global maxi_in_trn;global maxi_in_tst;global mini_out_trn;global mini_out_tst;
global maxi_out_trn;global maxi_out_tst;global gradprev_B;global gradprev_W1;
global gradprev_W2;global gradprev_C;global stabprev_W1;global stabprev_W2;
global delprev_B;global delprev_W1;global delprev_W2;global delprev_C;
global dwprev_B;global dwprev_W1;global dwprev_W2;global dwprev_C;global delta_B;
global delta_W1;global delta_W2;global delta_C;global dw_B;global dw_W1;
global dw_W2;global dw_C;global kf_meg;global ssas_choice;global trial_index;
global in_num;global out_num;global n_hid;global n2;global alpha_neuron;
global alpha_stability;global eigen_limit;global ksi;global kf_trn;global kf_tst;
global ep_sz;global max_iter;global limit_B_C;global limit_W;global max_trials;
global lower;global upper;global pi;global ZERO;global n_plus_w;global n_minus_w;
global dmin_w;global dmax_w;global d0_w;

% ---- Lf, Le ----
%Υπολογισμός μερικών παραγώγων των συναρτήσεων F k E
for i=1:n2,
    x1=sqr(W1(i))+sqr(W2(i))-1;
    Lf_w1(i)=p1*(1+sigmoid(x1,alpha_stability))*2*W1(i)*der_sigmoid(x1,alpha_stability));
    Lf_w1(i)=Lf_w1(i)+p2*(2*(W1(i)-W1_prev(i)));
    Lf_w2(i)=p1*(1+sigmoid(x1,alpha_stability))*2*W2(i)*der_sigmoid(x1,alpha_stability));
    Lf_w2(i)=Lf_w2(i)+p2*(2*(W2(i)-W2_prev(i)));
    Le_w1(i)=Lw(1,2*i-1)*state_init(2*i-1)*der_fa(1,(2*i-1))+Lw(1,2*i)*state_init(2*i)*der_fa(1,(2*i));
    for k=2:ep_sz,
        Le_w1(i)=Le_w1(i)+Lw(k,2*i-1)*state(k-1,2*i-1)*der_fa(k,(2*i-1))+Lw(k,2*i)*state(k,2*i)*der_fa(k,(2*i));
    end
    Le_w2(i)=Lw(1,2*i-1)*state_init(2*i)*der_fa(1,(2*i-1))-Lw(1,2*i)*state_init(2*i-1)*der_fa(1,(2*i));
    for k=2:ep_sz,
        Le_w2(i)=Le_w2(i)+Lw(k,2*i-1)*state(k-1,2*i)*der_fa(k,(2*i-1))-Lw(k,2*i)*state(k,2*i-1)*der_fa(k,(2*i));
    end
end
for i=1:n2,
    for j=1:in_num,
        Le_b(2*i-1,j)=0.0;
        for k=1:ep_sz,
            Le_b(2*i-1,j)=Le_b(2*i-1,j)+Lw(k,2*i-1)*u_trn(k,j)*der_fa(k,(2*i-1));
        end
        Le_b(2*i,j)=0;
        for k=1:ep_sz,
            Le_b(2*i,j)=Le_b(2*i,j)+Lw(k,2*i)*u_trn(k,j)*der_fa(k,(2*i));
        end
        Lf_b(2*i-1,j)=p2*(2.0*(B(2*i-1,j)-B_prev(2*i-1,j)));
        Lf_b(2*i,j)=p2*(2.0*(B(2*i,j)-B_prev(2*i,j)));
    end
end
for l=1:out_num,
    for i=1:n2,
        Le_c(l,2*i-1)=0;
        for k=1:ep_sz,
            Le_c(l,2*i-1)=Le_c(l,2*i-1)+Ly(k,l)*state(k,2*i-1)*der_fb(k,l);
        end
        Le_c(l,2*i)=0;
        for k=1:ep_sz,
            Le_c(l,2*i)=Le_c(l,2*i)+Ly(k,l)*state(k,2*i)*der_fb(k,l);
        end
    end
end

```

```

        end
        Lf_c(1,2*i-1)=p2*(2.0*(C(1,2*i-1)-C_prev(1,2*i-1)));
        Lf_c(1,2*i)=p2*(2.0*(C(1,2*i)-C_prev(1,2*i)));
    end
end
%-----
% ---- Delta updates ----
%Ανανέωση των μεγεθών βήματος με τα οποία θα ενημερωθούν τα βάρη
if ((epoch_index==1) && (iterat==1))
    for i=1:n_hid,
        for j=1:in_num,
            delta_B(i,j)=d0_w;
        end
    end
    for i=1:out_num,
        for j=1:n_hid,
            delta_C(i,j)=d0_w;
        end
    end
    for i=1:n2,
        delta_W1(i)=d0_w;
        delta_W2(i)=d0_w;
    end
else
    for i=1:n_hid,
        for j=1:in_num,
            if (gradprev_B(i,j)*Le_b(i,j)>0)
                delta_B(i,j)=minimum(delta_B(i,j)*n_plus_w,dmax_w);
            elseif (gradprev_B(i,j)*Le_b(i,j)<0)
                delta_B(i,j)=maximum(delta_B(i,j)*n_minus_w,dmin_w);
            end
        end
    end
    for i=1:out_num,
        for j=1:n_hid,
            if (gradprev_C(i,j)*Le_c(i,j)>0)
                delta_C(i,j)=minimum(delta_C(i,j)*n_plus_w,dmax_w);
            elseif (gradprev_C(i,j)*Le_c(i,j)<0)
                delta_C(i,j)=maximum(delta_C(i,j)*n_minus_w,dmin_w);
            end
        end
    end
    for i=1:n2,
        if (gradprev_W1(i)*Le_w1(i)>0.0)
            delta_W1(i)=minimum(delta_W1(i)*n_plus_w,dmax_w);
        elseif (gradprev_W1(i)*Le_w1(i)<0)
            delta_W1(i)=maximum(delta_W1(i)*n_minus_w,dmin_w);
        end
        if (gradprev_W2(i)*Le_w2(i)>0)
            delta_W2(i)=minimum(delta_W2(i)*n_plus_w,dmax_w);
        elseif (gradprev_W2(i)*Le_w2(i)<0)
            delta_W2(i)=maximum(delta_W2(i)*n_minus_w,dmin_w);
        end
    end
end
%-----
%---- Iee, Iff, Ief ----
%Υπολογισμός των βοηθητικών μεταβλητών Iee, Iff, Ief
Iee=0; Iff=0; Ief=0;
for i=1:n2,
    Iff=Iff+sqr(Lf_w1(i)*delta_W1(i))+sqr(Lf_w2(i)*delta_W2(i));
    Ief=Ief+Lf_w1(i)*Le_w1(i)*sqr(delta_W1(i))+Lf_w2(i)*Le_w2(i)*sqr(delta_W2(i));
    Iee=Iee+sqr(Le_w1(i)*delta_W1(i))+sqr(Le_w2(i)*delta_W2(i));
end

```



```

if (abs(Iff*Iee-sqr(Ief))>=0)
    for i=1:n_hid,
        for j=1:in_num,
            gradprev_B(i,j)=Le_b(i,j);
            dwprev_B(i,j)=dw_B(i,j);
        end
    end
    for i=1:out_num,
        for j=1:n_hid,
            gradprev_C(i,j)=Le_c(i,j);
            dwprev_C(i,j)=dw_C(i,j);
            C_prev(i,j)=C(i,j);
        end
    end
    for i=1:n2,
        gradprev_W1(i)=Le_w1(i);
        gradprev_W2(i)=Le_w2(i);
        stabprev_W1(i)=Lf_w1(i);
        stabprev_W2(i)=Lf_w2(i);
        dwprev_W1(i)=dw_W1(i);
        dwprev_W2(i)=dw_W2(i);
    end
end
fclose('all');
end

%Συνάρτηση υπολογισμού σφάλματος και εγγραφής τελικών αποτελεσμάτων σε αρχεία
function x=give_E(ep,iterat,end_index)

global output_trn;global output_tst;global outpat_trn;global outpat_tst;
global mini_out_trn;global mini_out_tst;global maxi_out_trn;global trial_index;
global out_num;global kf_tst;global ep_sz;global lower;global upper;
j=0;k=0;stab=0;out=0;outp=0;MSE=0;NRMSE=0;mesi_timh=0;paron=0;gs=0;ge=0;

if end_index==0
%Ανάκτηση σφάλματος εκπαίδευσης και σφάλματος δοκιμής της εποχής
%Σφάλμα εκπαίδευσης μετά την αποκανονικοποίηση
MSE=0;
for k=1:ep_sz,
    for j=1:out_num,
        out(j)=((maxi_out_trn(j)-mini_out_trn(j))*outpat_trn(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j)))/(upper-lower);
        outp(j)=((maxi_out_trn(j)-mini_out_trn(j))*outpat_tst(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j)))/(upper-lower);
        MSE=MSE+sqr(out(j)-outp(j));
    end
end
MSE=MSE/ep_sz;
fprintf('MSE_trn=%2.8f\n',MSE );
fid = fopen('c:\mat_temp1\mat_derr.dat','a+');
fprintf(fid,'%d\t %d\t %2.8f\t',iterat,ep,MSE);
%Σφάλμα δοκιμής μετά την αποκανονικοποίηση
MSE=0;
for k=1:kf_tst,
    for j=1:out_num,
        out(j)=((maxi_out_trn(j)-mini_out_trn(j))*outpat_tst(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j)))/(upper-lower);
        outp(j)=((maxi_out_trn(j)-mini_out_trn(j))*outpat_tst(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j)))/(upper-lower);
        MSE=MSE+sqr(out(j)-outp(j));
    end
end
MSE=MSE/kf_tst;
fprintf(fid,'%2.8f\n',MSE);
fclose(fid);
else
%Ανάκτηση των τελικών δοκιμαστικών δεδομένων και δεδομένων εκπαίδευσης της τελευταίας εποχής
fid = fopen('c:\mat_temp1\mat_dfin.dat','a+');
MSE=0.0;
for k=1:kf_tst,
    fprintf(fid,'%d\t',k);

```

```

    for j=1:out_num, %ta alla3a
        out(j)=(maxi_out_trn(j)-mini_out_trn(j))*outpat_tst(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j))/(upper-lower);
        outp(j)=(maxi_out_trn(j)-mini_out_trn(j))*output_tst(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j))/(upper-lower);
    if j<=(out_num-1)
        fprintf(fid,'%2.8f\t %2.8f',out(j),outp(j));
    else
        fprintf(fid,'%2.8f\t %2.8f\n',out(j),outp(j));
    end
    MSE=MSE+sqr(out(j)-outp(j));
    end
end
fclose(fid);
MSE=MSE/kf_tst;
fid=fopen('c:\mat_temp1\mat_dtot_err.dat','a+');
fprintf(fid,'%d\t %2.6f\n',trial_index,MSE );
fprintf('Try=%d\t MSEchk=%2.6f\n',trial_index,MSE );
fclose(fid);
fid=fopen('c:\mat_temp1\mat_dfn.tst','a+');
for k=1:ep_sz,
    fprintf(fid,'%d\t',k);
    for j=1:out_num,
        out(j)=(maxi_out_trn(j)-mini_out_trn(j))*outpat_trn(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j))/(upper-lower);
        outp(j)=(maxi_out_trn(j)-mini_out_trn(j))*output_trn(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j))/(upper-lower);
    if (j<=(out_num-1))
        fprintf(fid,'%2.8f\t %2.8f',out(j),outp(j));
    else
        fprintf(fid,'%2.8f\t %2.8f\n',out(j),outp(j));
    end
    end
end
end
fclose(fid);
end
end

%Συνάρτηση που φιλοξενεί τη διαδικασία εκπαίδευσης
function x=training_process()

global state;global state_init;global kf_trn;global ep_sz;global max_iter;
global max_trials;

iterat=1;epoch_index=1;end_signal=1;

while ((iterat<=max_iter) && (end_signal))
    for epoch_index=1:(kf_trn/ep_sz),
        read_epoch_data(epoch_index);
        model_operation();
        give_E(epoch_index,iterat,0);
        make_Lf();
        RENNCOM(iterat,epoch_index,end_signal);
    end
    iterat=iterat+1;
end
model_operation();
give_E(epoch_index,iterat,1);
end

```

2.3 Περιγραφή κώδικα για την περίπτωση των πνευμονικών ήχων CC, FC, SQ

```

function
x=statistical_scale(tmp2,tmp3,tmp4,tmp5,tmp6,tmp7,tmp9,tmp10,tmp11,tmp13,tmp17,tmp18,tmp21,tmp22,tmp23,tmp24,tmp25)
clc;
% BLOCK-DIAGONAL RECURRENT NEURAL NETWORK (BDRNN),
% σύμφωνα με το A172, εκπαιδευμένο με RENNCOM.
% - W έχει SCALED ORTHOGONAL υποπίνακες.
% - E = MSE
% - E = p2*{ ALIGNMENT FUMCTION }. p2={-1,0}
% - Όλα τα βάρη είναι ενημερωμένα.
% CC

global eigen;global p1;global p2;global B;global W1;global W2;global C;global B_prev;
global W1_prev;global W2_prev;global C_prev;global state;global output_trn;
global output_tst;global output_trn;global output_tst;global u_trn;global u_tst;
global Lw;global Ly;global state_init;global mini_in_trn;global mini_in_tst;
global maxi_in_trn;global maxi_in_tst;global mini_out_trn;global mini_out_tst;
global maxi_out_trn;global maxi_out_tst;global gradprev_B;global gradprev_W1;
global gradprev_W2;global gradprev_C;global stabprev_W1;global stabprev_W2;
global delprev_B;global delprev_W1;global delprev_W2;global delprev_C;
global dwprev_B;global dwprev_W1;global dwprev_W2;global dwprev_C;global delta_B;
global delta_W1;global delta_W2;global delta_C;global dw_B;global dw_W1;
global dw_W2;global dw_C;global kf_meg;global ssas_choice;global trial_index;
global in_num;global out_num;global n_hid;global n2;global alpha_neuron;
global alpha_stability;global eigen_limit;global ksi;global samples;global tst_samples;
global samples;global max_iter;global limit_B_C;global limit_W;global max_trials;
global lower;global upper;global pi;global ZERO;global n_plus_w;global n_minus_w;
global dmin_w;global dmax_w;global d0_w;global which_p2;

%----Μεταβλητές από gui----
in_num=tmp2;
out_num=tmp3;
n_hid=tmp4;
n2=tmp5;
alpha_neuron=tmp6;
alpha_stability=tmp7;
eigen_limit=1;
ksi=tmp9;
samples=tmp10;
tst_samples=tmp11;
max_iter=tmp13;
lower=tmp17;
upper=tmp18;
pi=3.141592654;
ZERO=0;
n_plus_w=tmp21;
n_minus_w=tmp22;
dmin_w=tmp23;
dmax_w=tmp24;
d0_w=tmp25;
%-----

which_p2=1; % 1.0// -1.0

%-----

%-----main-----
normalize_data();
initialization();
training_process();
get_parameters(1);
end
%-----Τέλος main-----

function x=maximum(x1,x2)

```

```

if x1>x2
    x=x1;
else x=x2;
end
end

function x=minimum(x1,x2)

if x1<x2
    x=x1;
else x=x2;
end
end

function x=my_exp(y)
if y>200
    fprintf('Argument to exp() =%f\n',y);
elseif y < -20
    x=0;
else x=exp(y);
end
end

function x=sqr(x1)
x=x1*x1;
end

function x=sgn(x1)
if x1<0
    x=-1;
else
    x=1;
end
end

function x=sigmoid(x1,alpha)
x=((1.0-my_exp(-alpha*x1))/(1.0+my_exp(-alpha*x1)));
end

%Παράγωγος σιγμοειδούς ως προς όρισμα (έξοδος αθροιστή)
function x=der_sigmoid(x1,alpha)
x=((2*alpha*my_exp(-alpha*x1))/sqr(1.0+my_exp(-alpha*x1)));
end

%Παράγωγος σιγμοειδούς ως προς την είσοδο στο κρυμμένο στρώμα
function x=der_fa(kk,ii)
global B;global W1;global W2;global state;global u_trn;global state_init;
global in_num;global alpha_neuron;

j=0;l=0;deikt=0;x1=0;

if mod(ii,2)==0
    deikt=ii/2;

    for j=1:in_num,
        x1=x1+B(ii,j)*u_trn(kk,j);
    end
    if (kk==1)
        x1=x1-W2(deikt)*state_init(ii-1)+W1(deikt)*state_init(ii);
    else
        x1=x1-W2(deikt)*state(kk-1,ii-1)+W1(deikt)*state(kk-1,ii);
    end
end
else
    deikt=(ii+1)/2;
    for j=1:in_num,
        x1=x1+B(ii,j)*u_trn(kk,j);
    end
end

```

```

        if kk==1
            x1=x1+W1(deikt)*state_init(ii)+W2(deikt)*state_init(ii+1);
        else
            x1=x1+W1(deikt)*state(kk-1,ii)+W2(deikt)*state(kk-1,ii+1);
        end
    end
    x=der_sigmoid(x1,alpha_neuron);
end

%Παράγωγος σιγμοειδούς ως προς την είσοδο στο στρώμα εξόδου
function x=der_fb(kk,ll)
global C;global state;global n2;global alpha_neuron;

x1=0;

for j=1:n2
    x1=x1+C(ll,2*j-1)*state(kk,2*j-1)+C(ll,2*j)*state(kk,2*j);
    x=der_sigmoid(x1,alpha_neuron);
end
end

function normalize_data()

global outpat_trn;global outpat_tst;global u_trn;global u_tst;
global Lw;global Ly;global state_init;global mini_in_trn;global mini_in_tst;
global maxi_in_trn;global maxi_in_tst;global mini_out_trn;global mini_out_tst;
global maxi_out_trn;global maxi_out_tst;global gradprev_B;global gradprev_W1;
global gradprev_W2;global gradprev_C;global stabprev_W1;global stabprev_W2;
global delprev_B;global delprev_W1;global delprev_W2;global delprev_C;
global dwprev_B;global dwprev_W1;global dwprev_W2;global dwprev_C;global delta_B;
global delta_W1;global delta_W2;global delta_C;global dw_B;global dw_W1;
global dw_W2;global dw_C;global kf_meg;global ssas_choice;global trial_index;
global in_num;global out_num;global n_hid;global n2;global alpha_neuron;
global alpha_stability;global eigen_limit;global ksi;global samples;global tst_samples;
global samples;global max_iter;global limit_B_C;global limit_W;global max_trials;
global lower;global upper;global pi;global ZERO;global n_plus_w;global n_minus_w;
global dmin_w;global dmax_w;global d0_w;global cc_dtrn;

u_tst=0;
outpat_tst=0;

%----Ανοίγμα αρχείου dtrn.dat-----
load c:\mat_in1\mat_ccnst_trn.dat;
for i=1:in_num+1,
    sthles_dtrn(:,i)=mat_ccnst_trn(:,i);
end

%----Εύρεση max,min εισόδου-----
for i=1:in_num,
    max_in(:,i)=max(sthles_dtrn(:,i));
    min_in(:,i)=min(sthles_dtrn(:,i));
end

%----Εύρεση max,min εξόδου-----
for i=in_num+1:out_num+in_num,
    max_out(:,i)=max(sthles_dtrn(:,i));
    min_out(:,i)=min(sthles_dtrn(:,i));
end
maxi_in_trn=max_in;
maxi_out_trn=max_out(:,in_num+1:in_num+out_num);
mini_in_trn=min_in;
mini_out_trn=min_out(:,in_num+1:in_num+out_num);
%----Κανονικοποίηση εισόδου-----
for j=1:in_num,
    for i=1:samples,
        inp(i,j)=((lower-upper)/(min_in(:,j)-max_in(:,j))*sthles_dtrn(i,j))+((min_in(:,j)*upper-max_in(:,j)*lower)/(min_in(:,j)-max_in(:,j))));
    end
end

```

```

    end
end
u_trn=inp;
%----Κανονικοποίηση εξόδου-----
for j=in_num+1:out_num+in_num,
    for i=1:samples,
        out(i,j)=(((lower-upper)/(min_out(:,j)-max_out(:,j))*sthles_dtrn(i,j))+((min_out(:,j)*upper-max_out(:,j)*lower))/(min_out(:,j)-
max_out(:,j))));
    end
end
output_trn=out;
temp=out(:,in_num+1:in_num+out_num);
dnorm_trn=[inp temp];
save c:\mat_in1\mat_dnorm_trn.dat dnorm_trn -ASCII;

%----Ανάγνωση και κανονικοποίηση των δοκιμαστικών δεδομένων-----
load c:\mat_in1\mat_ccnst_tst.dat;
for i=1:in_num+1,
    sthles_dtst(:,i)=mat_ccnst_tst(:,i);
end

%----Εύρεση max,min εισόδου-----
for i=1:in_num,
    max_in_tst(:,i)=max(sthles_dtst(:,i));
    min_in_tst(:,i)=min(sthles_dtst(:,i));
end

%----Εύρεση max,min εξόδου----
for i=in_num+1:out_num+in_num,
    max_out_tst(:,i)=max(sthles_dtst(:,i));
    min_out_tst(:,i)=min(sthles_dtst(:,i));
end

%----Δοκιμή δεδομένων εισόδου-----
for j=1:in_num,
    for i=1:tst_samples,
        u_tst(i,j)=((upper-lower)*sthles_dtst(i,j)+max_in(:,j)*lower-min_in(:,j)*upper)/(max_in(:,j)-min_in(:,j));
    end
end

%----Δοκιμή δεδομένων εξόδου----
for j=in_num+1:out_num+in_num,
    for i=1:tst_samples,
        temp1(i,j)=((upper-lower)*sthles_dtst(i,j)+max_out(:,j)*lower-min_out(:,j)*upper)/(max_out(:,j)-min_out(:,j));
    end
end
output_tst=temp1(:,in_num+1:in_num+out_num);
end

%Αρχικοποίηση των B,C,W1,W2 με τυχαίες τιμές
function x=initialization()
global eigen;global p1;global p2;global B;global W1;global W2;global C;global B_prev;
global W1_prev;global W2_prev;global C_prev;global state;global output_trn;
global output_tst;global output_trn;global output_tst;global u_trn;global u_tst;
global Lw;global Ly;global state_init;global mini_in_trn;global mini_in_tst;
global maxi_in_trn;global maxi_in_tst;global mini_out_trn;global mini_out_tst;
global maxi_out_trn;global maxi_out_tst;global gradprev_B;global gradprev_W1;
global gradprev_W2;global gradprev_C;global stabprev_W1;global stabprev_W2;
global dltaprev_B;global dltaprev_W1;global dltaprev_W2;global dltaprev_C;
global dwprev_B;global dwprev_W1;global dwprev_W2;global dwprev_C;global delta_B;
global delta_W1;global delta_W2;global delta_C;global dw_B;global dw_W1;
global dw_W2;global dw_C;global kf_meg;global ssas_choice;global trial_index;
global in_num;global out_num;global n_hid;global n2;global alpha_neuron;
global alpha_stability;global eigen_limit;global ksi;global samples;global tst_samples;
global samples;global max_iter;global limit_B_C;global limit_W;global max_trials;
global lower;global upper;global pi;global ZERO;global n_plus_w;global n_minus_w;

```

```

global dmin_w;global dmax_w;global d0_w;

sB=0;
sC=0;
sW=0;

load c:\mat_in1\mat_dparams.ini;
state_init=mat_dparams(1:n_hid);

B(:,1)=mat_dparams(n_hid+1:2*n_hid+1);
for i=1:n_hid,
    for j=1:in_num
        sB=sB+sqr(B(i,j))/(n_hid*in_num);
    end
end

C(1,:)=mat_dparams(2*n_hid+2:3*n_hid+2);
for i=1:out_num
    for j=1:n_hid,
        sC=sC+sqr(C(i,j))/(n_hid*out_num);
    end
end

W1(:,1)=mat_dparams(3*n_hid+3:3*n_hid+3+n2);
W2(:,1)=mat_dparams(3*n_hid+4+n2:3*n_hid+4+2*n2);
fid=fopen('c:\mat_temp1\mat_dweights.dat','a+');
for i=1:n2,
    fprintf(fid,'%2.8ft %2.8ft',W1(i),W2(i));
    sW=sW+(sqr(W1(i))+sqr(W2(i)))/n_hid;
end
fclose(fid);

fid=fopen('c:\mat_temp1\mat_dall_wgh.dat','a+');
fprintf(fid,'0\t 0\t %2.8ft %2.8ft %2.8ft %2.8ft',sqrt(sB),sqrt(sC),sqrt(sW),sqrt(sC+sB+sW) );
fclose(fid);

% Αρχικοποίηση προηγούμενων βαρών
for i=1:n_hid,
    for j=1:in_num,
        B_prev(i,j)=0;
    end
end

for i=1:out_num,
    for j=1:n_hid,
        C_prev(i,j)=0;
    end
end

for i=1:n2,
    W1_prev(i)=0;
    W2_prev(i)=0;
end
end
%-----

%Λειτουργία του RENNCOM
function x=model_operation(deiktis)

global W1;global W2;global C;global state;global output_trn;global output_tst;
global u_trn;global u_tst;global state_init;global in_num;global out_num;
global n_hid;global n2;global alpha_neuron;global tst_samples;global samples;global B;

if (deiktis==0)
for k=1:samples,
    for i=1:n2,
        sum1=0;
        sum2=0;

```

```

for j=1:in_num,
    sum1=sum1+B(2*i-1,j)*u_trn(k,j);
    sum2=sum2+B(2*i,j)*u_trn(k,j);
end
if k==1
    sum1=sum1+W1(i)*state_init(2*i-1)+W2(i)*state_init(2*i);
    sum2=sum2-W2(i)*state_init(2*i-1)+W1(i)*state_init(2*i);
else
    sum1=sum1+W1(i)*state(k-1,2*i-1)+W2(i)*state(k-1,2*i);
    sum2=sum2-W2(i)*state(k-1,2*i-1)+W1(i)*state(k-1,2*i);
end
state(k,2*i-1)=sigmoid(sum1,alpha_neuron);
state(k,2*i)=sigmoid(sum2,alpha_neuron);
end
for l=1:out_num,
    sum1=0;
    for j=1:n2,
        sum1=sum1+C(1,2*j-1)*state(k,2*j-1)+C(1,2*j)*state(k,2*j);
    end
    output_trn(k,l)=sigmoid(sum1,alpha_neuron);
end
end
else
% Αρχικοποίηση για τη δοκιμαστική φάση και για την επόμενη εποχή
for k=1:tst_samples,
    for i=1:n2,
        sum1=0;
        sum2=0;
        for j=1:in_num,
            sum1=sum1+B(2*i-1,j)*u_tst(k,j);
            sum2=sum2+B(2*i,j)*u_tst(k,j);
        end
        if k==1
            sum1=sum1+W1(i)*state_init(2*i-1)+W2(i)*state_init(2*i);
            sum2=sum2-W2(i)*state_init(2*i-1)+W1(i)*state_init(2*i);
        else
            sum1=sum1+W1(i)*state(k-1,2*i-1)+W2(i)*state(k-1,2*i);
            sum2=sum2-W2(i)*state(k-1,2*i-1)+W1(i)*state(k-1,2*i);
        end
        state(k,2*i-1)=sigmoid(sum1,alpha_neuron);
        state(k,2*i)=sigmoid(sum2,alpha_neuron);
    end
    for l=1:out_num,
        sum1=0;
        for j=1:n2,
            sum1=sum1+C(1,2*j-1)*state(k,2*j-1)+C(1,2*j)*state(k,2*j);
        end
        output_tst(k,l)=sigmoid(sum1,alpha_neuron);
    end
end
end
end

%Υπολογισμός των πολλαπλασιαστών Lagrange στη μέθοδο RENNCOM
function x=make_Lf(x1)

global W1;global W2;global C;global output_trn;global output_trn;global Lw;
global Ly;global out_num;global n_hid;global n2;global samples;
% ---- k=kf ----
for l=1:out_num,
    Ly(samples,l)=(output_trn(samples,l)-output_trn(samples,l))/(samples);
end
for i=1:n_hid,
    x=0;
    for l=1:out_num,
        x=x+(C(l,i)*der_fb(samples,l)*(output_trn(samples,l)-output_trn(samples,l)))/(samples);
        x=x+Ly(samples,l)*C(l,i)*der_fb(samples,l);
    end
end

```



```

end
Lw(samples,i)=x;
end
% ---- k=kf-1,...,1 ----
for k=(samples-1):-1:1,
    for l=1:out_num,
        Ly(k,l)=(output_trn(k,l)-output_trn(k,l))/(samples);
    end
    for i=1:n2,
        x=0;
        for l=1:out_num,
            x=x+(C(1,2*i-1)*der_fb(k,l)*(output_trn(k,l)-output_trn(k,l)))/(samples);
            x=x+Ly(k,l)*C(1,2*i-1)*der_fb(k,l);
        end
        Lw(k,2*i-1)=x+Lw(k+1,2*i-1)*W1(i)*der_fa((k+1),(2*i-1))-Lw(k+1,2*i)*W2(i)*der_fa((k+1),(2*i));
        x=0;
        for l=1:out_num,
            x=x+(C(1,2*i)*der_fb(k,l)*(output_trn(k,l)-output_trn(k,l)))/(samples);
            x=x+Ly(k,l)*C(1,2*i)*der_fb(k,l);
        end
        Lw(k,2*i)=x+Lw(k+1,2*i-1)*W2(i)*der_fa((k+1),(2*i-1))+Lw(k+1,2*i)*W1(i)*der_fa((k+1),(2*i));
    end
end
end

%Εφαρμογή της μεθόδου RENNCOM
function x=RENNCOM(iterat,epoch_index,end_signal)
global eigen;global p1;global p2;global B;global W1;global W2;global C;global B_prev;
global W1_prev;global W2_prev;global C_prev;global state;global output_trn;
global output_tst;global output_trn;global output_tst;global u_trn;global u_tst;
global Lw;global Ly;global state_init;global mini_in_trn;global mini_in_tst;
global maxi_in_trn;global maxi_in_tst;global mini_out_trn;global mini_out_tst;
global maxi_out_trn;global maxi_out_tst;global gradprev_B;global gradprev_W1;
global gradprev_W2;global gradprev_C;global stabprev_W1;global stabprev_W2;
global delprev_B;global delprev_W1;global delprev_W2;global delprev_C;
global dwprev_B;global dwprev_W1;global dwprev_W2;global dwprev_C;global delta_B;
global delta_W1;global delta_W2;global delta_C;global dw_B;global dw_W1;
global dw_W2;global dw_C;global kf_meg;global ssas_choice;global trial_index;
global in_num;global out_num;global n_hid;global n2;global alpha_neuron;
global alpha_stability;global eigen_limit;global ksi;global samples;global tst_samples;
global samples;global max_iter;global limit_B_C;global limit_W;global max_trials;
global lower;global upper;global pi;global ZERO;global n_plus_w;global n_minus_w;
global dmin_w;global dmax_w;global d0_w;

% ---- Lf, Le ----
%Υπολογισμός μερικών παραγώγων των συναρτήσεων F k E
for i=1:n2,
    x1=sqrt(W1(i))+sqrt(W2(i))-1;
    Lf_w1(i)=p1*((1+sigmoid(x1,alpha_stability))*2*W1(i)*der_sigmoid(x1,alpha_stability));
    Lf_w1(i)=Lf_w1(i)+p2*(2*(W1(i)-W1_prev(i)));
    Lf_w2(i)=p1*((1+sigmoid(x1,alpha_stability))*2*W2(i)*der_sigmoid(x1,alpha_stability));
    Lf_w2(i)=Lf_w2(i)+p2*(2*(W2(i)-W2_prev(i)));
    Le_w1(i)=Lw(1,2*i-1)*state_init(2*i-1)*der_fa(1,(2*i-1))+Lw(1,2*i)*state_init(2*i)*der_fa(1,(2*i));
    for k=2:samples,
        Le_w1(i)=Le_w1(i)+Lw(k,2*i-1)*state(k-1,2*i-1)*der_fa(k,(2*i-1))+Lw(k,2*i)*state(k,2*i)*der_fa(k,(2*i));
    end
    Le_w2(i)=Lw(1,2*i-1)*state_init(2*i)*der_fa(1,(2*i-1))-Lw(1,2*i)*state_init(2*i-1)*der_fa(1,(2*i));
    for k=2:samples,
        Le_w2(i)=Le_w2(i)+Lw(k,2*i-1)*state(k-1,2*i)*der_fa(k,(2*i-1))-Lw(k,2*i)*state(k,2*i-1)*der_fa(k,(2*i));
    end
end
for i=1:n2,
    for j=1:in_num,
        Le_b(2*i-1,j)=0.0;
        for k=1:samples,
            Le_b(2*i-1,j)=Le_b(2*i-1,j)+Lw(k,2*i-1)*u_trn(k,j)*der_fa(k,(2*i-1));

```

```

    end
    Le_b(2*i,j)=0;
    for k=1:samples,
        Le_b(2*i,j)=Le_b(2*i,j)+Lw(k,2*i)*u_trn(k,j)*der_fa(k,(2*i));
    end
    Lf_b(2*i-1,j)=p2*(2.0*(B(2*i-1,j)-B_prev(2*i-1,j)));
    Lf_b(2*i,j)=p2*(2.0*(B(2*i,j)-B_prev(2*i,j)));
end
end
for l=1:out_num,
    for i=1:n2,
        Le_c(1,2*i-1)=0;
        for k=1:samples,
            Le_c(1,2*i-1)=Le_c(1,2*i-1)+Ly(k,l)*state(k,2*i-1)*der_fb(k,l);
        end
        Le_c(1,2*i)=0;
        for k=1:samples,
            Le_c(1,2*i)=Le_c(1,2*i)+Ly(k,l)*state(k,2*i)*der_fb(k,l);
        end
        Lf_c(1,2*i-1)=p2*(2.0*(C(1,2*i-1)-C_prev(1,2*i-1)));
        Lf_c(1,2*i)=p2*(2.0*(C(1,2*i)-C_prev(1,2*i)));
    end
end
end
% -----
% ---- Delta updates ----
%Ανανέωση των μεγεθών βήματος με τα οποία θα ενημερωθούν τα βάρη
if ((epoch_index==1) && (iterat==1))
    for i=1:n_hid,
        for j=1:in_num,
            delta_B(i,j)=d0_w;
        end
    end
    for i=1:out_num,
        for j=1:n_hid,
            delta_C(i,j)=d0_w;
        end
    end
    for i=1:n2,
        delta_W1(i)=d0_w;
        delta_W2(i)=d0_w;
    end
end
else
    for i=1:n_hid,
        for j=1:in_num,
            if (gradprev_B(i,j)*Le_b(i,j)>0)
                delta_B(i,j)=minimum(delta_B(i,j)*n_plus_w,dmax_w);
            elseif (gradprev_B(i,j)*Le_b(i,j)<0)
                delta_B(i,j)=maximum(delta_B(i,j)*n_minus_w,dmin_w);
            end
        end
    end
end
for i=1:out_num,
    for j=1:n_hid,
        if (gradprev_C(i,j)*Le_c(i,j)>0)
            delta_C(i,j)=minimum(delta_C(i,j)*n_plus_w,dmax_w);
        elseif (gradprev_C(i,j)*Le_c(i,j)<0)
            delta_C(i,j)=maximum(delta_C(i,j)*n_minus_w,dmin_w);
        end
    end
end
end
for i=1:n2,
    if (gradprev_W1(i)*Le_w1(i)>0.0)
        delta_W1(i)=minimum(delta_W1(i)*n_plus_w,dmax_w);
    elseif (gradprev_W1(i)*Le_w1(i)<0)

```

```

    delta_W1(i)=maximum(delta_W1(i)*n_minus_w,dmin_w);
end

if (gradprev_W2(i)*Le_w2(i)>0)
    delta_W2(i)=minimum(delta_W2(i)*n_plus_w,dmax_w);
elseif (gradprev_W2(i)*Le_w2(i)<0)
    delta_W2(i)=maximum(delta_W2(i)*n_minus_w,dmin_w);
end
end
end
%-----
%---- Iee, Iff, Ief ----
%Υπολογισμός των βοηθητικών μεταβλητών Iee, Iff, Ief
Iee=0; Iff=0; Ief=0;
for i=1:n2,
    Iff=Iff+sqr(Lf_w1(i)*delta_W1(i)+sqr(Lf_w2(i)*delta_W2(i));
    Ief=Ief+Lf_w1(i)*Le_w1(i)*sqr(delta_W1(i))+Lf_w2(i)*Le_w2(i)*sqr(delta_W2(i));
    Iee=Iee+sqr(Le_w1(i)*delta_W1(i)+sqr(Le_w2(i)*delta_W2(i));
end

for i=1:n_hid,
    for j=1:in_num,
        Iff=Iff+sqr(Lf_b(i,j)*delta_B(i,j));
        Ief=Ief+Le_b(i,j)*sqr(delta_B(i,j))*Lf_b(i,j);
        Iee=Iee+sqr(Le_b(i,j)*delta_B(i,j));
    end
end
for l=1:out_num,
    for i=1:n_hid,
        Iff=Iff+sqr(Lf_c(l,i)*delta_C(l,i));
        Ief=Ief+Le_c(l,i)*sqr(delta_C(l,i))*Lf_c(l,i);
        Iee=Iee+sqr(Le_c(l,i)*delta_C(l,i));
    end
end
%-----
dE=-ksi*sqr(Iee);
sqr_root=(Iee-sqr(dE))/(Iff*Iee-sqr(Ief));
if (sqr_root<0)
    fprintf('\n\n\t\t\tERROR!!! Yporrizh posothta < 0\n\n');
    fprintf('Paron=%2.8f Iff=%2.8f Iee=%2.8f Ief=%2.8f\n\n',Iff*Iee-sqr(Ief),Iff,Iee,Ief);
else
    sqr_root=sqr(sqr_root);
end
%Ενημέρωση των βαρών
% ---- weight updates ----
fid = fopen('c:\mat_temp1\mat_dall_wgh.dat','a+');
fprintf(fid,'%d\t %d\t',iterat,epoch_index);
sB=0; sC=0; sW=0;
dsB=0; dsC=0; dsW=0;
fid1 = fopen('c:\mat_temp1\mat_dweights.dat','a+');
fprintf(fid1,'\n%d\t %d\t',iterat,epoch_index);
if (abs(Iff*Iee-sqr(Ief))>=0)
    for i=1:n_hid,
        for j=1:in_num,
            dw_B(i,j)=-sqr(delta_B(i,j))*sqr_root*(Lf_b(i,j)-(Le_b(i,j)*Ief/Iee))+sqr(delta_B(i,j))*Le_b(i,j)*dE/Iee;
            B_prev(i,j)=B(i,j); % update of previous weights
            B(i,j)=B(i,j)+dw_B(i,j);
            sB=sB+sqr(B(i,j))/(n_hid*in_num);
            dsB=dsB+sqr(dw_B(i,j))/(n_hid*in_num);
        end
    end
    for i=1:out_num,
        for j=1:n_hid,
            dw_C(i,j)=-sqr(delta_C(i,j))*sqr_root*(Lf_c(i,j)-(Le_c(i,j)*Ief/Iee))+sqr(delta_C(i,j))*Le_c(i,j)*dE/Iee;
            C_prev(i,j)=C(i,j); % update of previous weights
            C(i,j)=C(i,j)+dw_C(i,j);
            sC=sC+sqr(C(i,j))/(n_hid*out_num);
        end
    end
end

```

```

        dsC=dsC+sqr(dw_C(i,j))/(n_hid*out_num);
    end
end
for i=1:n2,
    dw_W1(i)=-sqr(delta_W1(i))*sqr_root*(Lf_w1(i)-(Le_w1(i)*Ief/Iee))+sqr(delta_W1(i))*Le_w1(i)*dE/Iee;
    W1_prev(i)=W1(i); % update of previous weights
    W1(i)=W1(i)+dw_W1(i);
    dw_W2(i)=-sqr(delta_W2(i))*sqr_root*(Lf_w2(i)-(Le_w2(i)*Ief/Iee))+sqr(delta_W2(i))*Le_w2(i)*dE/Iee;;
    W2_prev(i)=W2(i); % update of previous weights
    W2(i)=W2(i)+dw_W2(i);
    sW=sW+(sqr(W1(i))+sqr(W2(i)))/n_hid;
    dsW=dsW+(sqr(dw_W1(i))+sqr(dw_W2(i)))/n_hid;
end
end
for i=1:n2,
    fprintf(fid1,'%2.8f\t %2.8f\t %2.8f\t %2.8f\n',W1(i),W2(i) );
end
fprintf(fid,'%2.8f %2.8f %2.8f %2.8f\n',sqrt(sB),sqrt(sC),sqrt(sW),sqrt(sC+sB+sW) );
%Ενημέρωση προηγούμενων βαρών
if (abs(Iff*Iee-sqr(Ief))>=0)
    for i=1:n_hid,
        for j=1:in_num,
            gradprev_B(i,j)=Le_b(i,j);
            dwprev_B(i,j)=dw_B(i,j);
        end
    end
    for i=1:out_num,
        for j=1:n_hid,
            gradprev_C(i,j)=Le_c(i,j);
            dwprev_C(i,j)=dw_C(i,j);
            C_prev(i,j)=C(i,j);
        end
    end
    for i=1:n2,
        gradprev_W1(i)=Le_w1(i);
        gradprev_W2(i)=Le_w2(i);
        stabprev_W1(i)=Lf_w1(i);
        stabprev_W2(i)=Lf_w2(i);
        dwprev_W1(i)=dw_W1(i);
        dwprev_W2(i)=dw_W2(i);
    end
end
fclose('all');
end

%Συνάρτηση υπολογισμού σφαλμάτων και εγγραφής τελικών αποτελεσμάτων σε αρχεία
function x=give_E(ep,iterat,end_index)

global output_trn;global output_tst;global output_trn;global output_tst;
global mini_out_trn;global mini_out_tst;global maxi_out_trn;global trial_index;
global out_num;global tst_samples;global samples;global lower;global upper;
j=0;k=0;stab=0;out=0;outp=0;MSE=0;NRMSE=0;mesi_timh=0;paron=0;gs=0;ge=0;

if end_index==0 %Ανάκτηση σφάλματος εκπαίδευσης και σφάλματος δοκιμής της εποχής
%Σφάλμα εκπαίδευσης μετά την αποκατονικοποίηση
    RMSE=0;
    for k=1:samples,
        for j=1:out_num,
            out(j)=((maxi_out_trn(j)-mini_out_trn(j))*output_trn(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j)))/(upper-lower);
            outp(j)=((maxi_out_trn(j)-mini_out_trn(j))*output_trn(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j)))/(upper-lower);
            RMSE=RMSE+sqr(out(j)-outp(j));
        end
    end
    RMSE=sqrt(RMSE/samples);
    fprintf('RMSE_trn=%2.8f\n',RMSE );
    fid = fopen('c:\mat_temp1\mat_derr.dat','a+');
    fprintf(fid,'%d\t %d\t %2.8f\t',iterat,ep,MSE);

```

```

fclose(fid);
else
%Ανάκτηση τελικών δοκιμαστικών δεδομένων και δεδομένων εκπαίδευσης της τελευταίας εποχής
fid = fopen('c:\mat_temp1\mat_ccnst_trn.fin','a+');
for k=1:tst_samples,
    fprintf(fid,'%d\t',k);
    for j=1:out_num, %ta alla3a
        out(j)=((maxi_out_trn(j)-mini_out_trn(j))*outpat_tst(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j)))/(upper-lower);
        outp(j)=((maxi_out_trn(j)-mini_out_trn(j))*output_tst(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j)))/(upper-lower);
        if j<=(out_num-1)
            fprintf(fid,'%2.8f\t %2.8f',out(j),outp(j));
        else
            fprintf(fid,'%2.8f\t %2.8f\n',out(j),outp(j));
        end
    end
end
end
fclose(fid);
fid1=fopen('c:\mat_temp1\mat_derr_tst.dat','a+');
fid2=fopen('c:\mat_temp1\mat_ccnst_tst.fin','a+');
RMSE=0;
for k=1:samples,
    fprintf(fid2,'%d\t',k);
    for j=1:out_num,
        out(j)=((maxi_out_trn(j)-mini_out_trn(j))*outpat_trn(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j)))/(upper-lower);
        outp(j)=((maxi_out_trn(j)-mini_out_trn(j))*output_trn(k,j)+(upper*mini_out_trn(j)-lower*maxi_out_trn(j)))/(upper-lower);
        RMSE=RMSE+sqr(out(j)-outp(j));
        if (j<=(out_num-1))
            fprintf(fid2,'%2.8f\t %2.8f\t',out(j),outp(j));
        else
            fprintf(fid2,'%2.8f\t %2.8f\n',out(j),outp(j));
        end
    end
end
end
RMSE=sqrt(RMSE/(tst_samples));
fprintf(fid1,'RMSE_tst=%2.6f\n',RMSE);
fprintf('RMSE_tst=%2.6f\n',RMSE);
fclose(fid1);
fclose(fid2);
end
end

%Συνάρτηση που φιλοξενεί τη διαδικασία εκπαίδευσης
function x=training_process()

global state;global state_init;global samples;global samples;global max_iter;
global max_trials;

iterat=1;epoch_index=1;end_signal=1;
while ((iterat<=max_iter) && (end_signal))
    epoch_index=1
    monitor_eig(iterat,epoch_index);
    select_p1_p2(iterat,epoch_index);
    model_operation(0);
    give_E(epoch_index,iterat,0);
    make_Lf();
    RENNCOM(iterat,epoch_index,end_signal);
    iterat=iterat+1;
end
model_operation(0);
model_operation(1);
give_E(epoch_index,iterat,1);

end

```

ΚΕΦΑΛΑΙΟ 3^ο

Αξιολόγηση του μοντέλου με εφαρμογή σε πρόβλημα αναγνώρισης μη γραμμικού δυναμικού συστήματος

3.1 Αναγνώριση ενός μη γραμμικού δυναμικού συστήματος

Η έξοδος στο παράδειγμα αυτό λαμβάνεται από την ακόλουθη εξίσωση διαφορών:

$$y_p(k) = \frac{y_p(k-1) \cdot y_p(k-2) \cdot y_p(k-3) \cdot u(k-1) \cdot [y_p(k-3) - 1] + u(k)}{1 + y_p^2(k-2) + y_p^2(k-3)} \quad (25)$$

Από την ανωτέρω εξίσωση καθίσταται φανερό ότι η έξοδος εξαρτάται από τρεις παρελθούσες εξόδους, την τρέχουσα είσοδο και την είσοδο την προηγούμενη χρονική στιγμή. Χρησιμοποιείται παράλληλη μορφή εκπαίδευσης, όπου η μοναδική είσοδος είναι η $u(k)$ και δεν χρησιμοποιείται οποιαδήποτε άλλη πληροφορία από το σύνολο δεδομένων εκπαίδευσης.

Ακολουθεί ο πίνακας στον οποίο περιγράφονται οι συνδυασμοί των παραμέτρων εκπαίδευσης για την περίπτωση του προβλήματος Test Case

Πίνακας 1

Αριθμός Επανάληψης	n_hid	n2	alpha stability	ep_sz	n+	n-	max_ite r	MSE (error)
0	6	3	2	90	1,05	0,5	3	0,040217
1	6	3	2	90	1,05	0,5	7	0,036642
2	6	3	2	90	1,05	0,5	12	0,030163
3	6	3	2	90	1,05	0,7	3	0,027566
4	6	3	2	90	1,05	0,7	7	0,027093
5	6	3	2	90	1,05	0,7	12	0,027544
6	6	3	2	90	1,05	0,9	3	0,028127
7	6	3	2	90	1,05	0,9	7	0,031669
8	6	3	2	90	1,05	0,9	12	0,027071
9	6	3	2	90	1,15	0,5	3	0,032435
10	6	3	2	90	1,15	0,5	7	0,027441
11	6	3	2	90	1,15	0,5	12	0,033819
12	6	3	2	90	1,15	0,7	3	0,028505
13	6	3	2	90	1,15	0,7	7	0,028233
14	6	3	2	90	1,15	0,7	12	0,044038
15	6	3	2	90	1,15	0,9	3	0,045524
16	6	3	2	90	1,15	0,9	7	0,288894
17	6	3	2	90	1,15	0,9	12	1,165765
18	6	3	2	90	1,25	0,5	3	0,050707
19	6	3	2	90	1,25	0,5	7	0,034109
20	6	3	2	90	1,25	0,5	12	0,159311
21	6	3	2	90	1,25	0,7	3	1,089141
22	6	3	2	90	1,25	0,7	7	5,82183
23	6	3	2	90	1,25	0,7	12	0,269548
24	6	3	2	90	1,25	0,9	3	4,428352
25	6	3	2	90	1,25	0,9	7	0,269927
26	6	3	2	90	1,25	0,9	12	6,491691

27	6	3	2	270	1,05	0,5	3	0,085015
28	6	3	2	270	1,05	0,5	7	0,254716
29	6	3	2	270	1,05	0,5	12	0,48585
30	6	3	2	270	1,05	0,7	3	0,180003
31	6	3	2	270	1,05	0,7	7	0,050918
32	6	3	2	270	1,05	0,7	12	0,054985
33	6	3	2	270	1,05	0,9	3	0,03084
34	6	3	2	270	1,05	0,9	7	0,02724
35	6	3	2	270	1,05	0,9	12	0,227625
36	6	3	2	270	1,15	0,5	3	0,028249
37	6	3	2	270	1,15	0,5	7	0,027932
38	6	3	2	270	1,15	0,5	12	0,028375
39	6	3	2	270	1,15	0,7	3	0,027107
40	6	3	2	270	1,15	0,7	7	0,0273
41	6	3	2	270	1,15	0,7	12	0,028081
42	6	3	2	270	1,15	0,9	3	0,030981
43	6	3	2	270	1,15	0,9	7	0,027599
44	6	3	2	270	1,15	0,9	12	0,02736
45	6	3	2	270	1,25	0,5	3	0,028664
46	6	3	2	270	1,25	0,5	7	0,027447
47	6	3	2	270	1,25	0,5	12	0,027178
48	6	3	2	270	1,25	0,7	3	0,027196
49	6	3	2	270	1,25	0,7	7	0,027078
50	6	3	2	270	1,25	0,7	12	0,02764
51	6	3	2	270	1,25	0,9	3	0,47967
52	6	3	2	270	1,25	0,9	7	0,030958
53	6	3	2	270	1,25	0,9	12	0,16193
54	6	3	2	900	1,05	0,5	3	0,3298
55	6	3	2	900	1,05	0,5	7	0,318445
56	6	3	2	900	1,05	0,5	12	0,487715
57	6	3	2	900	1,05	0,7	3	0,216078
58	6	3	2	900	1,05	0,7	7	0,284856
59	6	3	2	900	1,05	0,7	12	0,381325
60	6	3	2	900	1,05	0,9	3	0,13403
61	6	3	2	900	1,05	0,9	7	0,701726
62	6	3	2	900	1,05	0,9	12	0,220945
63	6	3	2	900	1,15	0,5	3	0,475594
64	6	3	2	900	1,15	0,5	7	0,291338
65	6	3	2	900	1,15	0,5	12	0,465988
66	6	3	2	900	1,15	0,7	3	0,523434
67	6	3	2	900	1,15	0,7	7	0,541772
68	6	3	2	900	1,15	0,7	12	0,111424
69	6	3	2	900	1,15	0,9	3	0,138536
70	6	3	2	900	1,15	0,9	7	0,212683
71	6	3	2	900	1,15	0,9	12	0,479515
72	6	3	2	900	1,25	0,5	3	0,278
73	6	3	2	900	1,25	0,5	7	0,335456
74	6	3	2	900	1,25	0,5	12	0,324293
75	6	3	2	900	1,25	0,7	3	0,681548
76	6	3	2	900	1,25	0,7	7	0,265491
77	6	3	2	900	1,25	0,7	12	0,156921
78	6	3	2	900	1,25	0,9	3	0,248289
79	6	3	2	900	1,25	0,9	7	0,044058
80	6	3	2	900	1,25	0,9	12	0,133348

81	6	3	6	90	1,05	0,5	3	0,030889
82	6	3	6	90	1,05	0,5	7	0,028732
83	6	3	6	90	1,05	0,5	12	0,02809
84	6	3	6	90	1,05	0,7	3	0,02742
85	6	3	6	90	1,05	0,7	7	0,027222
86	6	3	6	90	1,05	0,7	12	0,027439
87	6	3	6	90	1,05	0,9	3	0,029857
88	6	3	6	90	1,05	0,9	7	0,029371
89	6	3	6	90	1,05	0,9	12	0,028222
90	6	3	6	90	1,15	0,5	3	0,033091
91	6	3	6	90	1,15	0,5	7	0,027352
92	6	3	6	90	1,15	0,5	12	0,030856
93	6	3	6	90	1,15	0,7	3	0,027769
94	6	3	6	90	1,15	0,7	7	0,028457
95	6	3	6	90	1,15	0,7	12	0,028402
96	6	3	6	90	1,15	0,9	3	0,239244
97	6	3	6	90	1,15	0,9	7	1,076438
98	6	3	6	90	1,15	0,9	12	0,297587
99	6	3	6	90	1,25	0,5	3	1,443363
100	6	3	6	90	1,25	0,5	7	0,876457
101	6	3	6	90	1,25	0,5	12	1,453275
102	6	3	6	90	1,25	0,7	3	0,031849
103	6	3	6	90	1,25	0,7	7	0,33583
104	6	3	6	90	1,25	0,7	12	0,052072
105	6	3	6	90	1,25	0,9	3	6,506938
106	6	3	6	90	1,25	0,9	7	0,822814
107	6	3	6	90	1,25	0,9	12	0,103005
108	6	3	6	270	1,05	0,5	3	0,15646
109	6	3	6	270	1,05	0,5	7	0,031355
110	6	3	6	270	1,05	0,5	12	0,132395
111	6	3	6	270	1,05	0,7	3	0,239136
112	6	3	6	270	1,05	0,7	7	0,150202
113	6	3	6	270	1,05	0,7	12	0,17526
114	6	3	6	270	1,05	0,9	3	0,060774
115	6	3	6	270	1,05	0,9	7	0,332598
116	6	3	6	270	1,05	0,9	12	0,144942
117	6	3	6	270	1,15	0,5	3	0,02733
118	6	3	6	270	1,15	0,5	7	0,026981
119	6	3	6	270	1,15	0,5	12	0,027169
120	6	3	6	270	1,15	0,7	3	0,028325
121	6	3	6	270	1,15	0,7	7	0,027893
122	6	3	6	270	1,15	0,7	12	0,027249
123	6	3	6	270	1,15	0,9	3	0,027064
124	6	3	6	270	1,15	0,9	7	0,027751
125	6	3	6	270	1,15	0,9	12	0,027489
126	6	3	6	270	1,25	0,5	3	0,027281
127	6	3	6	270	1,25	0,5	7	0,027497
128	6	3	6	270	1,25	0,5	12	0,027179
129	6	3	6	270	1,25	0,7	3	0,032191
130	6	3	6	270	1,25	0,7	7	0,027401
131	6	3	6	270	1,25	0,7	12	0,02777
132	6	3	6	270	1,25	0,9	3	0,030574
133	6	3	6	270	1,25	0,9	7	0,040297

134	6	3	6	270	1,25	0,9	12	0,035435
135	6	3	6	900	1,05	0,5	3	0,498484
136	6	3	6	900	1,05	0,5	7	0,359749
137	6	3	6	900	1,05	0,5	12	0,23237
138	6	3	6	900	1,05	0,7	3	0,683415
139	6	3	6	900	1,05	0,7	7	0,467353
140	6	3	6	900	1,05	0,7	12	0,041124
141	6	3	6	900	1,05	0,9	3	0,286663
142	6	3	6	900	1,05	0,9	7	0,116349
143	6	3	6	900	1,05	0,9	12	0,276418
144	6	3	6	900	1,15	0,5	3	0,265431
145	6	3	6	900	1,15	0,5	7	0,354203
146	6	3	6	900	1,15	0,5	12	0,402792
147	6	3	6	900	1,15	0,7	3	0,223957
148	6	3	6	900	1,15	0,7	7	0,371252
149	6	3	6	900	1,15	0,7	12	0,220534
150	6	3	6	900	1,15	0,9	3	0,273573
151	6	3	6	900	1,15	0,9	7	0,315134
152	6	3	6	900	1,15	0,9	12	0,491772
153	6	3	6	900	1,25	0,5	3	0,277219
154	6	3	6	900	1,25	0,5	7	0,47317
155	6	3	6	900	1,25	0,5	12	0,306153
156	6	3	6	900	1,25	0,7	3	0,242006
157	6	3	6	900	1,25	0,7	7	0,157831
158	6	3	6	900	1,25	0,7	12	0,484333
159	6	3	6	900	1,25	0,9	3	0,039563
160	6	3	6	900	1,25	0,9	7	0,437989
161	6	3	6	900	1,25	0,9	12	0,079477
162	6	3	8	90	1,05	0,5	3	0,02745
163	6	3	8	90	1,05	0,5	7	0,027554
164	6	3	8	90	1,05	0,5	12	0,057872
165	6	3	8	90	1,05	0,7	3	0,027336
166	6	3	8	90	1,05	0,7	7	0,027259
167	6	3	8	90	1,05	0,7	12	0,027319
168	6	3	8	90	1,05	0,9	3	0,029021
169	6	3	8	90	1,05	0,9	7	0,027444
170	6	3	8	90	1,05	0,9	12	0,03166
171	6	3	8	90	1,15	0,5	3	0,028379
172	6	3	8	90	1,15	0,5	7	0,028194
173	6	3	8	90	1,15	0,5	12	0,02753
174	6	3	8	90	1,15	0,7	3	0,030015
175	6	3	8	90	1,15	0,7	7	0,029561
176	6	3	8	90	1,15	0,7	12	0,028632
177	6	3	8	90	1,15	0,9	3	3,791597
178	6	3	8	90	1,15	0,9	7	0,158955
179	6	3	8	90	1,15	0,9	12	4,265654
180	6	3	8	90	1,25	0,5	3	0,295548
181	6	3	8	90	1,25	0,5	7	0,351522
182	6	3	8	90	1,25	0,5	12	0,20086
183	6	3	8	90	1,25	0,7	3	0,052984
184	6	3	8	90	1,25	0,7	7	0,38007
185	6	3	8	90	1,25	0,7	12	0,088706
186	6	3	8	90	1,25	0,9	3	6,501205
187	6	3	8	90	1,25	0,9	7	6,519698

188	6	3	8	90	1,25	0,9	12	6,5197
189	6	3	8	270	1,05	0,5	3	0,100626
190	6	3	8	270	1,05	0,5	7	0,136841
191	6	3	8	270	1,05	0,5	12	0,306797
192	6	3	8	270	1,05	0,7	3	0,02862
193	6	3	8	270	1,05	0,7	7	0,328329
194	6	3	8	270	1,05	0,7	12	0,20194
195	6	3	8	270	1,05	0,9	3	0,223671
196	6	3	8	270	1,05	0,9	7	0,11362
197	6	3	8	270	1,05	0,9	12	0,030639
198	6	3	8	270	1,15	0,5	3	0,027727
199	6	3	8	270	1,15	0,5	7	0,027218
200	6	3	8	270	1,15	0,5	12	0,030481
201	6	3	8	270	1,15	0,7	3	0,027067
202	6	3	8	270	1,15	0,7	7	0,02721
203	6	3	8	270	1,15	0,7	12	0,027306
204	6	3	8	270	1,15	0,9	3	0,027222
205	6	3	8	270	1,15	0,9	7	0,031596
206	6	3	8	270	1,15	0,9	12	0,029945
207	6	3	8	270	1,25	0,5	3	0,027195
208	6	3	8	270	1,25	0,5	7	0,029106
209	6	3	8	270	1,25	0,5	12	0,027202
210	6	3	8	270	1,25	0,7	3	0,027235
211	6	3	8	270	1,25	0,7	7	0,027079
212	6	3	8	270	1,25	0,7	12	0,027397
213	6	3	8	270	1,25	0,9	3	0,04145
214	6	3	8	270	1,25	0,9	7	0,038092
215	6	3	8	270	1,25	0,9	12	0,059831
216	6	3	8	900	1,05	0,5	3	1,042587
217	6	3	8	900	1,05	0,5	7	0,628248
218	6	3	8	900	1,05	0,5	12	0,554801
219	6	3	8	900	1,05	0,7	3	0,345927
220	6	3	8	900	1,05	0,7	7	0,190531
221	6	3	8	900	1,05	0,7	12	0,14776
222	6	3	8	900	1,05	0,9	3	0,314028
223	6	3	8	900	1,05	0,9	7	0,064552
224	6	3	8	900	1,05	0,9	12	0,37319
225	6	3	8	900	1,15	0,5	3	0,447265
226	6	3	8	900	1,15	0,5	7	0,209467
227	6	3	8	900	1,15	0,5	12	0,929933
228	6	3	8	900	1,15	0,7	3	0,175058
229	6	3	8	900	1,15	0,7	7	0,510488
230	6	3	8	900	1,15	0,7	12	0,591826
231	6	3	8	900	1,15	0,9	3	0,342574
232	6	3	8	900	1,15	0,9	7	0,430559
233	6	3	8	900	1,15	0,9	12	0,199716
234	6	3	8	900	1,25	0,5	3	0,418967
235	6	3	8	900	1,25	0,5	7	0,37993
236	6	3	8	900	1,25	0,5	12	0,173752
237	6	3	8	900	1,25	0,7	3	0,028617
238	6	3	8	900	1,25	0,7	7	0,035689
239	6	3	8	900	1,25	0,7	12	0,553566
240	6	3	8	900	1,25	0,9	3	0,235605

241	6	3	8	900	1,25	0,9	7	0,51104
242	6	3	8	900	1,25	0,9	12	0,253204
243	12	6	2	90	1,05	0,5	3	0,033519
244	12	6	2	90	1,05	0,5	7	0,060593
245	12	6	2	90	1,05	0,5	12	0,036356
246	12	6	2	90	1,05	0,7	3	0,027323
247	12	6	2	90	1,05	0,7	7	0,027392
248	12	6	2	90	1,05	0,7	12	0,027217
249	12	6	2	90	1,05	0,9	3	0,035751
250	12	6	2	90	1,05	0,9	7	0,027435
251	12	6	2	90	1,05	0,9	12	0,031851
252	12	6	2	90	1,15	0,5	3	0,027121
253	12	6	2	90	1,15	0,5	7	0,027797
254	12	6	2	90	1,15	0,5	12	0,029357
255	12	6	2	90	1,15	0,7	3	0,030127
256	12	6	2	90	1,15	0,7	7	0,033093
257	12	6	2	90	1,15	0,7	12	0,029849
258	12	6	2	90	1,15	0,9	3	0,164884
259	12	6	2	90	1,15	0,9	7	4,971294
260	12	6	2	90	1,15	0,9	12	0,553177
261	12	6	2	90	1,25	0,5	3	0,04233
262	12	6	2	90	1,25	0,5	7	0,092022
263	12	6	2	90	1,25	0,5	12	0,291463
264	12	6	2	90	1,25	0,7	3	0,883506
265	12	6	2	90	1,25	0,7	7	4,6573
266	12	6	2	90	1,25	0,7	12	0,176958
267	12	6	2	90	1,25	0,9	3	6,502847
268	12	6	2	90	1,25	0,9	7	6,519698
269	12	6	2	90	1,25	0,9	12	6,504453
270	12	6	2	270	1,05	0,5	3	0,032851
271	12	6	2	270	1,05	0,5	7	0,139803
272	12	6	2	270	1,05	0,5	12	0,027421
273	12	6	2	270	1,05	0,7	3	0,028733
274	12	6	2	270	1,05	0,7	7	0,329966
275	12	6	2	270	1,05	0,7	12	0,214436
276	12	6	2	270	1,05	0,9	3	0,030305
277	12	6	2	270	1,05	0,9	7	0,027805
278	12	6	2	270	1,05	0,9	12	0,027387
279	12	6	2	270	1,15	0,5	3	0,028746
280	12	6	2	270	1,15	0,5	7	0,027528
281	12	6	2	270	1,15	0,5	12	0,027926
282	12	6	2	270	1,15	0,7	3	0,02725
283	12	6	2	270	1,15	0,7	7	0,027037
284	12	6	2	270	1,15	0,7	12	0,027249
285	12	6	2	270	1,15	0,9	3	0,029338
286	12	6	2	270	1,15	0,9	7	0,027478
287	12	6	2	270	1,15	0,9	12	0,030091
288	12	6	2	270	1,25	0,5	3	0,027187
289	12	6	2	270	1,25	0,5	7	0,02738
290	12	6	2	270	1,25	0,5	12	0,027177
291	12	6	2	270	1,25	0,7	3	0,027407
292	12	6	2	270	1,25	0,7	7	0,032959
293	12	6	2	270	1,25	0,7	12	0,028096
294	12	6	2	270	1,25	0,9	3	0,425136

295	12	6	2	270	1,25	0,9	7	0,027801
296	12	6	2	270	1,25	0,9	12	0,032149
297	12	6	2	900	1,05	0,5	3	0,784046
298	12	6	2	900	1,05	0,5	7	0,637365
299	12	6	2	900	1,05	0,5	12	0,291615
300	12	6	2	900	1,05	0,7	3	0,120068
301	12	6	2	900	1,05	0,7	7	0,153812
302	12	6	2	900	1,05	0,7	12	0,2914
303	12	6	2	900	1,05	0,9	3	0,292178
304	12	6	2	900	1,05	0,9	7	0,413103
305	12	6	2	900	1,05	0,9	12	0,545875
306	12	6	2	900	1,15	0,5	3	0,710065
307	12	6	2	900	1,15	0,5	7	0,205304
308	12	6	2	900	1,15	0,5	12	0,289121
309	12	6	2	900	1,15	0,7	3	0,032382
310	12	6	2	900	1,15	0,7	7	0,47853
311	12	6	2	900	1,15	0,7	12	0,183004
312	12	6	2	900	1,15	0,9	3	0,134196
313	12	6	2	900	1,15	0,9	7	0,183525
314	12	6	2	900	1,15	0,9	12	0,374781
315	12	6	2	900	1,25	0,5	3	0,567133
316	12	6	2	900	1,25	0,5	7	0,651363
317	12	6	2	900	1,25	0,5	12	0,088291
318	12	6	2	900	1,25	0,7	3	0,321483
319	12	6	2	900	1,25	0,7	7	0,075155
320	12	6	2	900	1,25	0,7	12	0,925532
321	12	6	2	900	1,25	0,9	3	0,165092
322	12	6	2	900	1,25	0,9	7	0,028599
323	12	6	2	900	1,25	0,9	12	0,670651
324	12	6	6	90	1,05	0,5	3	0,027606
325	12	6	6	90	1,05	0,5	7	0,034689
326	12	6	6	90	1,05	0,5	12	0,027288
327	12	6	6	90	1,05	0,7	3	0,027494
328	12	6	6	90	1,05	0,7	7	0,028613
329	12	6	6	90	1,05	0,7	12	0,02718
330	12	6	6	90	1,05	0,9	3	0,032477
331	12	6	6	90	1,05	0,9	7	0,030859
332	12	6	6	90	1,05	0,9	12	0,033021
333	12	6	6	90	1,15	0,5	3	0,027761
334	12	6	6	90	1,15	0,5	7	0,02751
335	12	6	6	90	1,15	0,5	12	0,030605
336	12	6	6	90	1,15	0,7	3	0,0321
337	12	6	6	90	1,15	0,7	7	0,029817
338	12	6	6	90	1,15	0,7	12	0,030281
339	12	6	6	90	1,15	0,9	3	0,038871
340	12	6	6	90	1,15	0,9	7	1,522632
341	12	6	6	90	1,15	0,9	12	6,506913
342	12	6	6	90	1,25	0,5	3	0,521902
343	12	6	6	90	1,25	0,5	7	0,610842
344	12	6	6	90	1,25	0,5	12	0,203049
345	12	6	6	90	1,25	0,7	3	0,183945
346	12	6	6	90	1,25	0,7	7	3,863373
347	12	6	6	90	1,25	0,7	12	6,504117

348	12	6	6	90	1,25	0,9	3	0,176375
349	12	6	6	90	1,25	0,9	7	6,428496
350	12	6	6	90	1,25	0,9	12	6,504453
351	12	6	6	270	1,05	0,5	3	0,040071
352	12	6	6	270	1,05	0,5	7	0,246075
353	12	6	6	270	1,05	0,5	12	0,028551
354	12	6	6	270	1,05	0,7	3	0,029624
355	12	6	6	270	1,05	0,7	7	0,176636
356	12	6	6	270	1,05	0,7	12	0,027847
357	12	6	6	270	1,05	0,9	3	0,095467
358	12	6	6	270	1,05	0,9	7	0,033802
359	12	6	6	270	1,05	0,9	12	0,065277
360	12	6	6	270	1,15	0,5	3	0,028124
361	12	6	6	270	1,15	0,5	7	0,027195
362	12	6	6	270	1,15	0,5	12	0,027246
363	12	6	6	270	1,15	0,7	3	0,027745
364	12	6	6	270	1,15	0,7	7	0,027513
365	12	6	6	270	1,15	0,7	12	0,027734
366	12	6	6	270	1,15	0,9	3	0,029131
367	12	6	6	270	1,15	0,9	7	0,027945
368	12	6	6	270	1,15	0,9	12	0,028732
369	12	6	6	270	1,25	0,5	3	0,027354
370	12	6	6	270	1,25	0,5	7	0,028433
371	12	6	6	270	1,25	0,5	12	0,029057
372	12	6	6	270	1,25	0,7	3	0,027701
373	12	6	6	270	1,25	0,7	7	0,028595
374	12	6	6	270	1,25	0,7	12	0,027418
375	12	6	6	270	1,25	0,9	3	3,072963
376	12	6	6	270	1,25	0,9	7	0,028298
377	12	6	6	270	1,25	0,9	12	0,395523
378	12	6	6	900	1,05	0,5	3	0,641359
379	12	6	6	900	1,05	0,5	7	0,420552
380	12	6	6	900	1,05	0,5	12	0,277156
381	12	6	6	900	1,05	0,7	3	0,039411
382	12	6	6	900	1,05	0,7	7	0,437748
383	12	6	6	900	1,05	0,7	12	0,321687
384	12	6	6	900	1,05	0,9	3	0,185143
385	12	6	6	900	1,05	0,9	7	0,080164
386	12	6	6	900	1,05	0,9	12	0,166723
387	12	6	6	900	1,15	0,5	3	0,327209
388	12	6	6	900	1,15	0,5	7	0,365398
389	12	6	6	900	1,15	0,5	12	0,966358
390	12	6	6	900	1,15	0,7	3	0,149955
391	12	6	6	900	1,15	0,7	7	0,154115
392	12	6	6	900	1,15	0,7	12	0,156122
393	12	6	6	900	1,15	0,9	3	0,495126
394	12	6	6	900	1,15	0,9	7	0,090311
395	12	6	6	900	1,15	0,9	12	0,600243
396	12	6	6	900	1,25	0,5	3	0,095921
397	12	6	6	900	1,25	0,5	7	0,083257
398	12	6	6	900	1,25	0,5	12	0,118269
399	12	6	6	900	1,25	0,7	3	0,198938
400	12	6	6	900	1,25	0,7	7	0,374158
401	12	6	6	900	1,25	0,7	12	0,470205

402	12	6	6	900	1,25	0,9	3	0,028119
403	12	6	6	900	1,25	0,9	7	0,476353
404	12	6	6	900	1,25	0,9	12	0,48688
405	12	6	8	90	1,05	0,5	3	0,027883
406	12	6	8	90	1,05	0,5	7	0,027271
407	12	6	8	90	1,05	0,5	12	0,027231
408	12	6	8	90	1,05	0,7	3	0,031339
409	12	6	8	90	1,05	0,7	7	0,027166
410	12	6	8	90	1,05	0,7	12	0,027322
411	12	6	8	90	1,05	0,9	3	0,028928
412	12	6	8	90	1,05	0,9	7	0,030654
413	12	6	8	90	1,05	0,9	12	0,029661
414	12	6	8	90	1,15	0,5	3	0,030657
415	12	6	8	90	1,15	0,5	7	0,029704
416	12	6	8	90	1,15	0,5	12	0,030678
417	12	6	8	90	1,15	0,7	3	0,033592
418	12	6	8	90	1,15	0,7	7	0,065195
419	12	6	8	90	1,15	0,7	12	0,028162
420	12	6	8	90	1,15	0,9	3	0,183458
421	12	6	8	90	1,15	0,9	7	0,229752
422	12	6	8	90	1,15	0,9	12	2,981444
423	12	6	8	90	1,25	0,5	3	0,393061
424	12	6	8	90	1,25	0,5	7	0,276679
425	12	6	8	90	1,25	0,5	12	2,010811
426	12	6	8	90	1,25	0,7	3	0,871758
427	12	6	8	90	1,25	0,7	7	0,041858
428	12	6	8	90	1,25	0,7	12	0,844247
429	12	6	8	90	1,25	0,9	3	0,37458
430	12	6	8	90	1,25	0,9	7	0,585051
431	12	6	8	90	1,25	0,9	12	0,279241
432	12	6	8	270	1,05	0,5	3	0,10287
433	12	6	8	270	1,05	0,5	7	0,040436
434	12	6	8	270	1,05	0,5	12	0,081605
435	12	6	8	270	1,05	0,7	3	0,03162
436	12	6	8	270	1,05	0,7	7	0,02727
437	12	6	8	270	1,05	0,7	12	0,031187
438	12	6	8	270	1,05	0,9	3	0,027803
439	12	6	8	270	1,05	0,9	7	0,074653
440	12	6	8	270	1,05	0,9	12	0,079174
441	12	6	8	270	1,15	0,5	3	0,028458
442	12	6	8	270	1,15	0,5	7	0,029021
443	12	6	8	270	1,15	0,5	12	0,029779
444	12	6	8	270	1,15	0,7	3	0,028335
445	12	6	8	270	1,15	0,7	7	0,027292
446	12	6	8	270	1,15	0,7	12	0,027051
447	12	6	8	270	1,15	0,9	3	0,027589
448	12	6	8	270	1,15	0,9	7	0,027129
449	12	6	8	270	1,15	0,9	12	0,027551
450	12	6	8	270	1,25	0,5	3	0,027481
451	12	6	8	270	1,25	0,5	7	0,027258
452	12	6	8	270	1,25	0,5	12	0,027037
453	12	6	8	270	1,25	0,7	3	0,029721
454	12	6	8	270	1,25	0,7	7	0,02719

455	12	6	8	270	1,25	0,7	12	0,031479
456	12	6	8	270	1,25	0,9	3	1,331014
457	12	6	8	270	1,25	0,9	7	0,212468
458	12	6	8	270	1,25	0,9	12	0,506288
459	12	6	8	900	1,05	0,5	3	0,433733
460	12	6	8	900	1,05	0,5	7	0,239982
461	12	6	8	900	1,05	0,5	12	0,404504
462	12	6	8	900	1,05	0,7	3	0,160264
463	12	6	8	900	1,05	0,7	7	0,168989
464	12	6	8	900	1,05	0,7	12	0,562596
465	12	6	8	900	1,05	0,9	3	0,132123
466	12	6	8	900	1,05	0,9	7	0,221755
467	12	6	8	900	1,05	0,9	12	0,468526
468	12	6	8	900	1,15	0,5	3	0,077899
469	12	6	8	900	1,15	0,5	7	0,337884
470	12	6	8	900	1,15	0,5	12	0,280279
471	12	6	8	900	1,15	0,7	3	0,371325
472	12	6	8	900	1,15	0,7	7	0,395129
473	12	6	8	900	1,15	0,7	12	0,559249
474	12	6	8	900	1,15	0,9	3	0,410336
475	12	6	8	900	1,15	0,9	7	0,422329
476	12	6	8	900	1,15	0,9	12	0,219841
477	12	6	8	900	1,25	0,5	3	0,137572
478	12	6	8	900	1,25	0,5	7	0,117947
479	12	6	8	900	1,25	0,5	12	0,579841
480	12	6	8	900	1,25	0,7	3	0,121094
481	12	6	8	900	1,25	0,7	7	0,031296
482	12	6	8	900	1,25	0,7	12	0,051581
483	12	6	8	900	1,25	0,9	3	0,167041
484	12	6	8	900	1,25	0,9	7	0,418438
485	12	6	8	900	1,25	0,9	12	0,586901
486	18	9	2	90	1,05	0,5	3	0,037606
487	18	9	2	90	1,05	0,5	7	0,028454
488	18	9	2	90	1,05	0,5	12	0,038225
489	18	9	2	90	1,05	0,7	3	0,028
490	18	9	2	90	1,05	0,7	7	0,028658
491	18	9	2	90	1,05	0,7	12	0,031354
492	18	9	2	90	1,05	0,9	3	0,356445
493	18	9	2	90	1,05	0,9	7	6,25422
494	18	9	2	90	1,05	0,9	12	5,780275
495	18	9	2	90	1,15	0,5	3	0,034641
496	18	9	2	90	1,15	0,5	7	0,04821
497	18	9	2	90	1,15	0,5	12	5,756718
498	18	9	2	90	1,15	0,7	3	2,920576
499	18	9	2	90	1,15	0,7	7	4,434448
500	18	9	2	90	1,15	0,7	12	0,309584
501	18	9	2	90	1,15	0,9	3	2,153897
502	18	9	2	90	1,15	0,9	7	6,519697
503	18	9	2	90	1,15	0,9	12	0,972157
504	18	9	2	90	1,25	0,5	3	0,3459
505	18	9	2	90	1,25	0,5	7	0,174295
506	18	9	2	90	1,25	0,5	12	0,715406
507	18	9	2	90	1,25	0,7	3	0,040394
508	18	9	2	90	1,25	0,7	7	0,677711

509	18	9	2	90	1,25	0,7	12	0,063126
510	18	9	2	90	1,25	0,9	3	6,491691
511	18	9	2	90	1,25	0,9	7	6,369779
512	18	9	2	90	1,25	0,9	12	6,503875
513	18	9	2	270	1,05	0,5	3	0,029519
514	18	9	2	270	1,05	0,5	7	0,668752
515	18	9	2	270	1,05	0,5	12	0,027474
516	18	9	2	270	1,05	0,7	3	0,027912
517	18	9	2	270	1,05	0,7	7	0,206863
518	18	9	2	270	1,05	0,7	12	0,224553
519	18	9	2	270	1,05	0,9	3	0,029909
520	18	9	2	270	1,05	0,9	7	0,027318
521	18	9	2	270	1,05	0,9	12	0,038726
522	18	9	2	270	1,15	0,5	3	0,027545
523	18	9	2	270	1,15	0,5	7	0,027794
524	18	9	2	270	1,15	0,5	12	0,027617
525	18	9	2	270	1,15	0,7	3	0,02721
526	18	9	2	270	1,15	0,7	7	0,027528
527	18	9	2	270	1,15	0,7	12	0,027531
528	18	9	2	270	1,15	0,9	3	0,029794
529	18	9	2	270	1,15	0,9	7	0,02772
530	18	9	2	270	1,15	0,9	12	0,031973
531	18	9	2	270	1,25	0,5	3	0,026912
532	18	9	2	270	1,25	0,5	7	0,027199
533	18	9	2	270	1,25	0,5	12	0,032389
534	18	9	2	270	1,25	0,7	3	0,039566
535	18	9	2	270	1,25	0,7	7	0,30035
536	18	9	2	270	1,25	0,7	12	0,027998
537	18	9	2	270	1,25	0,9	3	0,028238
538	18	9	2	270	1,25	0,9	7	1,138322
539	18	9	2	270	1,25	0,9	12	0,043501
540	18	9	2	900	1,05	0,5	3	0,56059
541	18	9	2	900	1,05	0,5	7	0,216737
542	18	9	2	900	1,05	0,5	12	0,349071
543	18	9	2	900	1,05	0,7	3	0,350763
544	18	9	2	900	1,05	0,7	7	0,35455
545	18	9	2	900	1,05	0,7	12	0,430667
546	18	9	2	900	1,05	0,9	3	0,329433
547	18	9	2	900	1,05	0,9	7	0,048442
548	18	9	2	900	1,05	0,9	12	0,027614
549	18	9	2	900	1,15	0,5	3	0,244083
550	18	9	2	900	1,15	0,5	7	0,321591
551	18	9	2	900	1,15	0,5	12	0,028345
552	18	9	2	900	1,15	0,7	3	0,029933
553	18	9	2	900	1,15	0,7	7	0,32224
554	18	9	2	900	1,15	0,7	12	0,876972
555	18	9	2	900	1,15	0,9	3	0,027268
556	18	9	2	900	1,15	0,9	7	0,255101
557	18	9	2	900	1,15	0,9	12	0,102762
558	18	9	2	900	1,25	0,5	3	0,28581
559	18	9	2	900	1,25	0,5	7	0,094725
560	18	9	2	900	1,25	0,5	12	0,347241
561	18	9	2	900	1,25	0,7	3	0,421024

562	18	9	2	900	1,25	0,7	7	0,027824
563	18	9	2	900	1,25	0,7	12	0,130942
564	18	9	2	900	1,25	0,9	3	0,255861
565	18	9	2	900	1,25	0,9	7	0,462368
566	18	9	2	900	1,25	0,9	12	0,193165
567	18	9	6	90	1,05	0,5	3	0,045423
568	18	9	6	90	1,05	0,5	7	0,032384
569	18	9	6	90	1,05	0,5	12	0,027211
570	18	9	6	90	1,05	0,7	3	0,031468
571	18	9	6	90	1,05	0,7	7	0,030635
572	18	9	6	90	1,05	0,7	12	0,027892
573	18	9	6	90	1,05	0,9	3	0,031046
574	18	9	6	90	1,05	0,9	7	0,028811
575	18	9	6	90	1,05	0,9	12	0,037836
576	18	9	6	90	1,15	0,5	3	0,027636
577	18	9	6	90	1,15	0,5	7	0,028116
578	18	9	6	90	1,15	0,5	12	0,03366
579	18	9	6	90	1,15	0,7	3	0,031574
580	18	9	6	90	1,15	0,7	7	0,028663
581	18	9	6	90	1,15	0,7	12	0,113472
582	18	9	6	90	1,15	0,9	3	0,060777
583	18	9	6	90	1,15	0,9	7	0,929078
584	18	9	6	90	1,15	0,9	12	0,937644
585	18	9	6	90	1,25	0,5	3	1,148368
586	18	9	6	90	1,25	0,5	7	0,173973
587	18	9	6	90	1,25	0,5	12	0,127065
588	18	9	6	90	1,25	0,7	3	0,864241
589	18	9	6	90	1,25	0,7	7	2,564306
590	18	9	6	90	1,25	0,7	12	0,089006
591	18	9	6	90	1,25	0,9	3	6,504453
592	18	9	6	90	1,25	0,9	7	0,420356
593	18	9	6	90	1,25	0,9	12	0,319701
594	18	9	6	270	1,05	0,5	3	0,02784
595	18	9	6	270	1,05	0,5	7	0,028212
596	18	9	6	270	1,05	0,5	12	0,0381
597	18	9	6	270	1,05	0,7	3	0,032077
598	18	9	6	270	1,05	0,7	7	0,027906
599	18	9	6	270	1,05	0,7	12	0,135693
600	18	9	6	270	1,05	0,9	3	0,039957
601	18	9	6	270	1,05	0,9	7	0,027219
602	18	9	6	270	1,05	0,9	12	0,213725
603	18	9	6	270	1,15	0,5	3	0,027897
604	18	9	6	270	1,15	0,5	7	0,028921
605	18	9	6	270	1,15	0,5	12	0,027711
606	18	9	6	270	1,15	0,7	3	0,030824
607	18	9	6	270	1,15	0,7	7	0,027669
608	18	9	6	270	1,15	0,7	12	0,027379
609	18	9	6	270	1,15	0,9	3	0,028395
610	18	9	6	270	1,15	0,9	7	0,034117
611	18	9	6	270	1,15	0,9	12	0,029721
612	18	9	6	270	1,25	0,5	3	0,028145
613	18	9	6	270	1,25	0,5	7	0,027981
614	18	9	6	270	1,25	0,5	12	0,190206
615	18	9	6	270	1,25	0,7	3	0,041873

616	18	9	6	270	1,25	0,7	7	0,027145
617	18	9	6	270	1,25	0,7	12	0,027486
618	18	9	6	270	1,25	0,9	3	0,051407
619	18	9	6	270	1,25	0,9	7	0,03255
620	18	9	6	270	1,25	0,9	12	0,079696
621	18	9	6	900	1,05	0,5	3	0,633691
622	18	9	6	900	1,05	0,5	7	0,561256
623	18	9	6	900	1,05	0,5	12	0,028972
624	18	9	6	900	1,05	0,7	3	0,187893
625	18	9	6	900	1,05	0,7	7	0,102869
626	18	9	6	900	1,05	0,7	12	1,26221
627	18	9	6	900	1,05	0,9	3	1,075388
628	18	9	6	900	1,05	0,9	7	0,094016
629	18	9	6	900	1,05	0,9	12	0,810222
630	18	9	6	900	1,15	0,5	3	0,255564
631	18	9	6	900	1,15	0,5	7	0,676879
632	18	9	6	900	1,15	0,5	12	0,174145
633	18	9	6	900	1,15	0,7	3	0,093975
634	18	9	6	900	1,15	0,7	7	0,842408
635	18	9	6	900	1,15	0,7	12	0,042067
636	18	9	6	900	1,15	0,9	3	0,167269
637	18	9	6	900	1,15	0,9	7	0,041111
638	18	9	6	900	1,15	0,9	12	0,336038
639	18	9	6	900	1,25	0,5	3	0,106164
640	18	9	6	900	1,25	0,5	7	0,30883
641	18	9	6	900	1,25	0,5	12	0,062477
642	18	9	6	900	1,25	0,7	3	0,028493
643	18	9	6	900	1,25	0,7	7	0,62887
644	18	9	6	900	1,25	0,7	12	0,09898
645	18	9	6	900	1,25	0,9	3	0,028199
646	18	9	6	900	1,25	0,9	7	0,458902
647	18	9	6	900	1,25	0,9	12	0,066682
648	18	9	8	90	1,05	0,5	3	0,028795
649	18	9	8	90	1,05	0,5	7	0,034503
650	18	9	8	90	1,05	0,5	12	0,02821
651	18	9	8	90	1,05	0,7	3	0,027316
652	18	9	8	90	1,05	0,7	7	0,027089
653	18	9	8	90	1,05	0,7	12	0,045208
654	18	9	8	90	1,05	0,9	3	0,034372
655	18	9	8	90	1,05	0,9	7	0,030351
656	18	9	8	90	1,05	0,9	12	0,029578
657	18	9	8	90	1,15	0,5	3	0,027953
658	18	9	8	90	1,15	0,5	7	0,027264
659	18	9	8	90	1,15	0,5	12	0,027405
660	18	9	8	90	1,15	0,7	3	0,028269
661	18	9	8	90	1,15	0,7	7	0,029079
662	18	9	8	90	1,15	0,7	12	0,030648
663	18	9	8	90	1,15	0,9	3	0,523311
664	18	9	8	90	1,15	0,9	7	0,192335
665	18	9	8	90	1,15	0,9	12	0,076854
666	18	9	8	90	1,25	0,5	3	0,445511
667	18	9	8	90	1,25	0,5	7	0,635824
668	18	9	8	90	1,25	0,5	12	0,176296

669	18	9	8	90	1,25	0,7	3	0,358339
670	18	9	8	90	1,25	0,7	7	0,051637
671	18	9	8	90	1,25	0,7	12	1,28561
672	18	9	8	90	1,25	0,9	3	6,504451
673	18	9	8	90	1,25	0,9	7	0,345245
674	18	9	8	90	1,25	0,9	12	1,011538
675	18	9	8	270	1,05	0,5	3	0,028366
676	18	9	8	270	1,05	0,5	7	0,146477
677	18	9	8	270	1,05	0,5	12	0,125195
678	18	9	8	270	1,05	0,7	3	0,02771
679	18	9	8	270	1,05	0,7	7	0,028226
680	18	9	8	270	1,05	0,7	12	0,135347
681	18	9	8	270	1,05	0,9	3	0,028133
682	18	9	8	270	1,05	0,9	7	0,029653
683	18	9	8	270	1,05	0,9	12	0,166496
684	18	9	8	270	1,15	0,5	3	0,02724
685	18	9	8	270	1,15	0,5	7	0,026987
686	18	9	8	270	1,15	0,5	12	0,028766
687	18	9	8	270	1,15	0,7	3	0,027411
688	18	9	8	270	1,15	0,7	7	0,027525
689	18	9	8	270	1,15	0,7	12	0,027314
690	18	9	8	270	1,15	0,9	3	0,027628
691	18	9	8	270	1,15	0,9	7	0,027954
692	18	9	8	270	1,15	0,9	12	0,028256
693	18	9	8	270	1,25	0,5	3	0,029195
694	18	9	8	270	1,25	0,5	7	0,02717
695	18	9	8	270	1,25	0,5	12	0,027892
696	18	9	8	270	1,25	0,7	3	0,027441
697	18	9	8	270	1,25	0,7	7	0,035194
698	18	9	8	270	1,25	0,7	12	0,029397
699	18	9	8	270	1,25	0,9	3	0,061401
700	18	9	8	270	1,25	0,9	7	0,263757
701	18	9	8	270	1,25	0,9	12	0,030301
702	18	9	8	900	1,05	0,5	3	0,917522
703	18	9	8	900	1,05	0,5	7	0,4336
704	18	9	8	900	1,05	0,5	12	0,531475
705	18	9	8	900	1,05	0,7	3	0,302063
706	18	9	8	900	1,05	0,7	7	0,490618
707	18	9	8	900	1,05	0,7	12	0,144362
708	18	9	8	900	1,05	0,9	3	0,16982
709	18	9	8	900	1,05	0,9	7	0,30985
710	18	9	8	900	1,05	0,9	12	0,029385
711	18	9	8	900	1,15	0,5	3	0,266394
712	18	9	8	900	1,15	0,5	7	0,780733
713	18	9	8	900	1,15	0,5	12	0,515762
714	18	9	8	900	1,15	0,7	3	1,443653
715	18	9	8	900	1,15	0,7	7	0,143045
716	18	9	8	900	1,15	0,7	12	0,414835
717	18	9	8	900	1,15	0,9	3	0,439565
718	18	9	8	900	1,15	0,9	7	0,058511
719	18	9	8	900	1,15	0,9	12	0,255757
720	18	9	8	900	1,25	0,5	3	0,24148
721	18	9	8	900	1,25	0,5	7	0,035203
722	18	9	8	900	1,25	0,5	12	0,027736

723	18	9	8	900	1,25	0,7	3	0,192427
724	18	9	8	900	1,25	0,7	7	0,581618
725	18	9	8	900	1,25	0,7	12	0,0273
726	18	9	8	900	1,25	0,9	3	0,027678
727	18	9	8	900	1,25	0,9	7	0,653822
728	18	9	8	900	1,25	0,9	12	0,778049
729	24	12	2	90	1,05	0,5	3	0,042183
730	24	12	2	90	1,05	0,5	7	0,028387
731	24	12	2	90	1,05	0,5	12	0,029325
732	24	12	2	90	1,05	0,7	3	0,028438
733	24	12	2	90	1,05	0,7	7	0,027238
734	24	12	2	90	1,05	0,7	12	0,027694
735	24	12	2	90	1,05	0,9	3	0,035627
736	24	12	2	90	1,05	0,9	7	0,029462
737	24	12	2	90	1,05	0,9	12	0,027593
738	24	12	2	90	1,15	0,5	3	0,027691
739	24	12	2	90	1,15	0,5	7	0,030411
740	24	12	2	90	1,15	0,5	12	0,030768
741	24	12	2	90	1,15	0,7	3	0,031079
742	24	12	2	90	1,15	0,7	7	0,031465
743	24	12	2	90	1,15	0,7	12	0,029054
744	24	12	2	90	1,15	0,9	3	6,503409
745	24	12	2	90	1,15	0,9	7	1,085146
746	24	12	2	90	1,15	0,9	12	0,571788
747	24	12	2	90	1,25	0,5	3	1,322305
748	24	12	2	90	1,25	0,5	7	0,223675
749	24	12	2	90	1,25	0,5	12	1,38527
750	24	12	2	90	1,25	0,7	3	0,053719
751	24	12	2	90	1,25	0,7	7	0,275659
752	24	12	2	90	1,25	0,7	12	0,050409
753	24	12	2	90	1,25	0,9	3	0,103108
754	24	12	2	90	1,25	0,9	7	6,506938
755	24	12	2	90	1,25	0,9	12	6,502216
756	24	12	2	270	1,05	0,5	3	0,217212
757	24	12	2	270	1,05	0,5	7	0,168311
758	24	12	2	270	1,05	0,5	12	0,031362
759	24	12	2	270	1,05	0,7	3	0,027495
760	24	12	2	270	1,05	0,7	7	0,029249
761	24	12	2	270	1,05	0,7	12	0,037352
762	24	12	2	270	1,05	0,9	3	0,030523
763	24	12	2	270	1,05	0,9	7	0,027472
764	24	12	2	270	1,05	0,9	12	0,027129
765	24	12	2	270	1,15	0,5	3	0,028941
766	24	12	2	270	1,15	0,5	7	0,027649
767	24	12	2	270	1,15	0,5	12	0,027527
768	24	12	2	270	1,15	0,7	3	0,027245
769	24	12	2	270	1,15	0,7	7	0,027783
770	24	12	2	270	1,15	0,7	12	0,027666
771	24	12	2	270	1,15	0,9	3	0,032027
772	24	12	2	270	1,15	0,9	7	0,027367
773	24	12	2	270	1,15	0,9	12	0,029307
774	24	12	2	270	1,25	0,5	3	0,027249
775	24	12	2	270	1,25	0,5	7	0,027663

776	24	12	2	270	1,25	0,5	12	0,02737
777	24	12	2	270	1,25	0,7	3	0,027418
778	24	12	2	270	1,25	0,7	7	0,028532
779	24	12	2	270	1,25	0,7	12	0,027979
780	24	12	2	270	1,25	0,9	3	0,044336
781	24	12	2	270	1,25	0,9	7	0,146665
782	24	12	2	270	1,25	0,9	12	0,038112
783	24	12	2	900	1,05	0,5	3	0,044407
784	24	12	2	900	1,05	0,5	7	0,870972
785	24	12	2	900	1,05	0,5	12	0,034625
786	24	12	2	900	1,05	0,7	3	0,12214
787	24	12	2	900	1,05	0,7	7	0,343167
788	24	12	2	900	1,05	0,7	12	0,597929
789	24	12	2	900	1,05	0,9	3	0,15025
790	24	12	2	900	1,05	0,9	7	0,333373
791	24	12	2	900	1,05	0,9	12	0,748523
792	24	12	2	900	1,15	0,5	3	0,034276
793	24	12	2	900	1,15	0,5	7	0,079005
794	24	12	2	900	1,15	0,5	12	0,100539
795	24	12	2	900	1,15	0,7	3	0,180471
796	24	12	2	900	1,15	0,7	7	0,708326
797	24	12	2	900	1,15	0,7	12	0,43267
798	24	12	2	900	1,15	0,9	3	0,254716
799	24	12	2	900	1,15	0,9	7	1,157493
800	24	12	2	900	1,15	0,9	12	0,208139
801	24	12	2	900	1,25	0,5	3	0,129469
802	24	12	2	900	1,25	0,5	7	0,055592
803	24	12	2	900	1,25	0,5	12	0,040657
804	24	12	2	900	1,25	0,7	3	0,120197
805	24	12	2	900	1,25	0,7	7	0,096798
806	24	12	2	900	1,25	0,7	12	1,27377
807	24	12	2	900	1,25	0,9	3	0,031236
808	24	12	2	900	1,25	0,9	7	0,596525
809	24	12	2	900	1,25	0,9	12	0,435749
810	24	12	6	90	1,05	0,5	3	0,027691
811	24	12	6	90	1,05	0,5	7	0,032969
812	24	12	6	90	1,05	0,5	12	0,045213
813	24	12	6	90	1,05	0,7	3	0,027241
814	24	12	6	90	1,05	0,7	7	0,028296
815	24	12	6	90	1,05	0,7	12	0,027729
816	24	12	6	90	1,05	0,9	3	0,0305
817	24	12	6	90	1,05	0,9	7	0,034093
818	24	12	6	90	1,05	0,9	12	0,037644
819	24	12	6	90	1,15	0,5	3	0,028274
820	24	12	6	90	1,15	0,5	7	0,027082
821	24	12	6	90	1,15	0,5	12	0,033764
822	24	12	6	90	1,15	0,7	3	0,028077
823	24	12	6	90	1,15	0,7	7	0,031258
824	24	12	6	90	1,15	0,7	12	0,040012
825	24	12	6	90	1,15	0,9	3	0,023811
826	24	12	6	90	1,15	0,9	7	0,127354
827	24	12	6	90	1,15	0,9	12	6,519666
828	24	12	6	90	1,25	0,5	3	0,046622
829	24	12	6	90	1,25	0,5	7	0,061525

830	24	12	6	90	1,25	0,5	12	0,685293
831	24	12	6	90	1,25	0,7	3	6,514775
832	24	12	6	90	1,25	0,7	7	2,103146
833	24	12	6	90	1,25	0,7	12	0,14629
834	24	12	6	90	1,25	0,9	3	6,519699
835	24	12	6	90	1,25	0,9	7	6,491691
836	24	12	6	90	1,25	0,9	12	2,134231
837	24	12	6	270	1,05	0,5	3	0,264348
838	24	12	6	270	1,05	0,5	7	0,027896
839	24	12	6	270	1,05	0,5	12	0,030224
840	24	12	6	270	1,05	0,7	3	0,027413
841	24	12	6	270	1,05	0,7	7	0,046576
842	24	12	6	270	1,05	0,7	12	0,027463
843	24	12	6	270	1,05	0,9	3	0,069891
844	24	12	6	270	1,05	0,9	7	0,02894
845	24	12	6	270	1,05	0,9	12	0,119217
846	24	12	6	270	1,15	0,5	3	0,027428
847	24	12	6	270	1,15	0,5	7	0,027086
848	24	12	6	270	1,15	0,5	12	0,027917
849	24	12	6	270	1,15	0,7	3	0,027889
850	24	12	6	270	1,15	0,7	7	0,028241
851	24	12	6	270	1,15	0,7	12	0,027099
852	24	12	6	270	1,15	0,9	3	0,030433
853	24	12	6	270	1,15	0,9	7	0,030455
854	24	12	6	270	1,15	0,9	12	0,032485
855	24	12	6	270	1,25	0,5	3	0,027177
856	24	12	6	270	1,25	0,5	7	0,030419
857	24	12	6	270	1,25	0,5	12	0,029306
858	24	12	6	270	1,25	0,7	3	0,028824
859	24	12	6	270	1,25	0,7	7	0,028133
860	24	12	6	270	1,25	0,7	12	0,027952
861	24	12	6	270	1,25	0,9	3	0,039678
862	24	12	6	270	1,25	0,9	7	0,366274
863	24	12	6	270	1,25	0,9	12	0,038872
864	24	12	6	900	1,05	0,5	3	1,051992
865	24	12	6	900	1,05	0,5	7	0,33315
866	24	12	6	900	1,05	0,5	12	0,704329
867	24	12	6	900	1,05	0,7	3	0,032729
868	24	12	6	900	1,05	0,7	7	0,379043
869	24	12	6	900	1,05	0,7	12	1,084883
870	24	12	6	900	1,05	0,9	3	0,035831
871	24	12	6	900	1,05	0,9	7	0,320783
872	24	12	6	900	1,05	0,9	12	0,404636
873	24	12	6	900	1,15	0,5	3	0,30449
874	24	12	6	900	1,15	0,5	7	0,402205
875	24	12	6	900	1,15	0,5	12	0,065206
876	24	12	6	900	1,15	0,7	3	0,857649
877	24	12	6	900	1,15	0,7	7	0,252333
878	24	12	6	900	1,15	0,7	12	0,979369
879	24	12	6	900	1,15	0,9	3	0,080773
880	24	12	6	900	1,15	0,9	7	0,492877
881	24	12	6	900	1,15	0,9	12	0,136824
882	24	12	6	900	1,25	0,5	3	0,028124

883	24	12	6	900	1,25	0,5	7	0,028602
884	24	12	6	900	1,25	0,5	12	0,458098
885	24	12	6	900	1,25	0,7	3	0,027575
886	24	12	6	900	1,25	0,7	7	0,73942
887	24	12	6	900	1,25	0,7	12	0,075428
888	24	12	6	900	1,25	0,9	3	0,155443
889	24	12	6	900	1,25	0,9	7	0,028598
890	24	12	6	900	1,25	0,9	12	0,029936
891	24	12	8	90	1,05	0,5	3	0,047877
892	24	12	8	90	1,05	0,5	7	0,029286
893	24	12	8	90	1,05	0,5	12	0,039462
894	24	12	8	90	1,05	0,7	3	0,032015
895	24	12	8	90	1,05	0,7	7	0,028556
896	24	12	8	90	1,05	0,7	12	0,028003
897	24	12	8	90	1,05	0,9	3	0,028142
898	24	12	8	90	1,05	0,9	7	0,029568
899	24	12	8	90	1,05	0,9	12	0,028217
900	24	12	8	90	1,15	0,5	3	0,032591
901	24	12	8	90	1,15	0,5	7	0,028973
902	24	12	8	90	1,15	0,5	12	0,035118
903	24	12	8	90	1,15	0,7	3	0,032859
904	24	12	8	90	1,15	0,7	7	0,031647
905	24	12	8	90	1,15	0,7	12	0,054441
906	24	12	8	90	1,15	0,9	3	1,118434
907	24	12	8	90	1,15	0,9	7	6,50334
908	24	12	8	90	1,15	0,9	12	6,506938
909	24	12	8	90	1,25	0,5	3	0,039739
910	24	12	8	90	1,25	0,5	7	2,49878
911	24	12	8	90	1,25	0,5	12	0,131161
912	24	12	8	90	1,25	0,7	3	1,281001
913	24	12	8	90	1,25	0,7	7	0,519218
914	24	12	8	90	1,25	0,7	12	1,340442
915	24	12	8	90	1,25	0,9	3	6,23699
916	24	12	8	90	1,25	0,9	7	2,403954
917	24	12	8	90	1,25	0,9	12	6,5197
918	24	12	8	270	1,05	0,5	3	0,02954
919	24	12	8	270	1,05	0,5	7	0,027543
920	24	12	8	270	1,05	0,5	12	0,028155
921	24	12	8	270	1,05	0,7	3	0,028419
922	24	12	8	270	1,05	0,7	7	0,027089
923	24	12	8	270	1,05	0,7	12	0,027363
924	24	12	8	270	1,05	0,9	3	0,161282
925	24	12	8	270	1,05	0,9	7	0,060982
926	24	12	8	270	1,05	0,9	12	0,027278
927	24	12	8	270	1,15	0,5	3	0,027172
928	24	12	8	270	1,15	0,5	7	0,030128
929	24	12	8	270	1,15	0,5	12	0,027295
930	24	12	8	270	1,15	0,7	3	0,027296
931	24	12	8	270	1,15	0,7	7	0,029305
932	24	12	8	270	1,15	0,7	12	0,02721
933	24	12	8	270	1,15	0,9	3	0,380004
934	24	12	8	270	1,15	0,9	7	0,039889
935	24	12	8	270	1,15	0,9	12	0,028025
936	24	12	8	270	1,25	0,5	3	0,027553

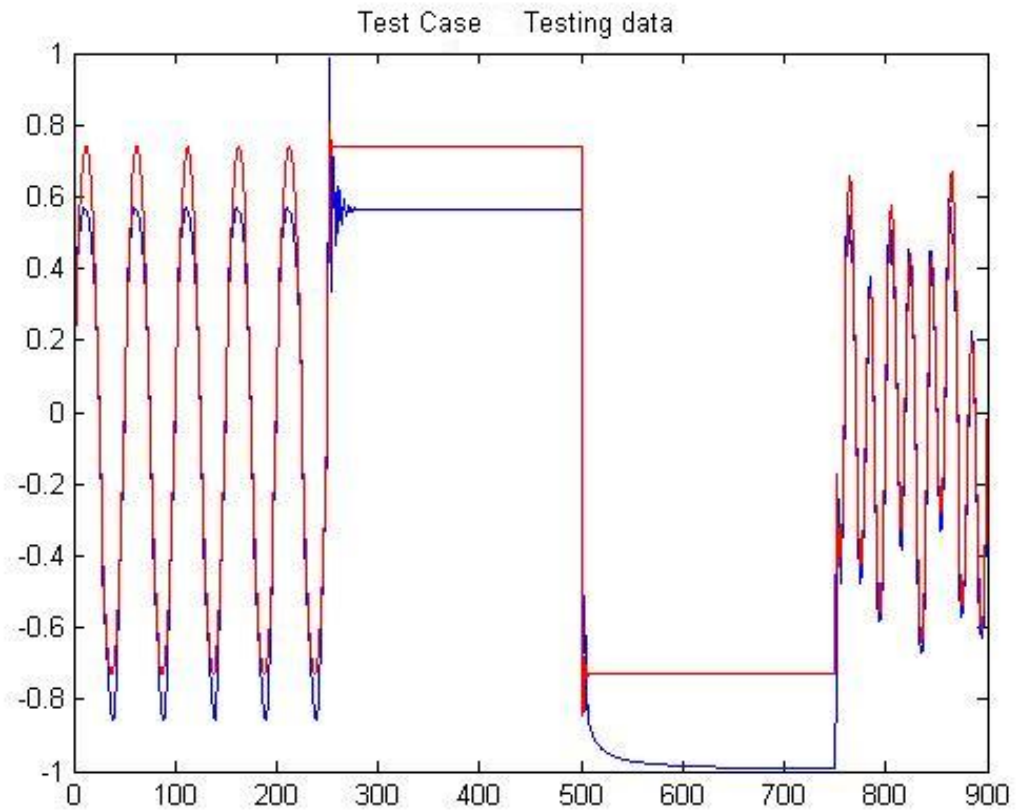
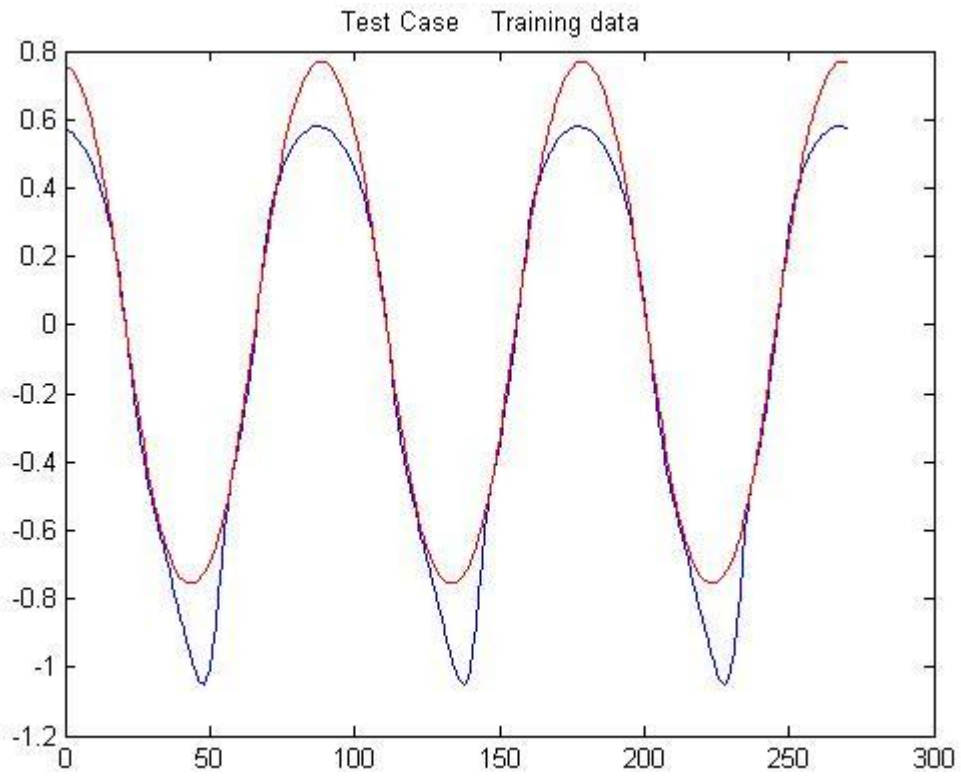
937	24	12	8	270	1,25	0,5	7	0,027475
938	24	12	8	270	1,25	0,5	12	0,027609
939	24	12	8	270	1,25	0,7	3	0,027459
940	24	12	8	270	1,25	0,7	7	0,027269
941	24	12	8	270	1,25	0,7	12	0,045941
942	24	12	8	270	1,25	0,9	3	0,028775
943	24	12	8	270	1,25	0,9	7	0,066437
944	24	12	8	270	1,25	0,9	12	0,085169
945	24	12	8	900	1,05	0,5	3	0,155872
946	24	12	8	900	1,05	0,5	7	0,368605
947	24	12	8	900	1,05	0,5	12	0,064769
948	24	12	8	900	1,05	0,7	3	0,061169
949	24	12	8	900	1,05	0,7	7	0,27781
950	24	12	8	900	1,05	0,7	12	0,670278
951	24	12	8	900	1,05	0,9	3	0,07995
952	24	12	8	900	1,05	0,9	7	0,34297
953	24	12	8	900	1,05	0,9	12	0,508623
954	24	12	8	900	1,15	0,5	3	0,104994
955	24	12	8	900	1,15	0,5	7	0,366371
956	24	12	8	900	1,15	0,5	12	0,027642
957	24	12	8	900	1,15	0,7	3	0,274319
958	24	12	8	900	1,15	0,7	7	0,52711
959	24	12	8	900	1,15	0,7	12	0,886441
960	24	12	8	900	1,15	0,9	3	0,578245
961	24	12	8	900	1,15	0,9	7	0,043949
962	24	12	8	900	1,15	0,9	12	1,187216
963	24	12	8	900	1,25	0,5	3	0,774454
964	24	12	8	900	1,25	0,5	7	0,027689
965	24	12	8	900	1,25	0,5	12	0,045653
966	24	12	8	900	1,25	0,7	3	0,170245
967	24	12	8	900	1,25	0,7	7	0,097549
968	24	12	8	900	1,25	0,7	12	0,440494
969	24	12	8	900	1,25	0,9	3	0,556113
970	24	12	8	900	1,25	0,9	7	0,092554
971	24	12	8	900	1,25	0,9	12	0,777646

Στον παρακάτω πίνακα βλέπουμε ορισμένα στατιστικά στοιχεία για τα παραπάνω αποτελέσματα :

Πίνακας 1.1

Minimum MSE	Maximum MSE	Average	Standard Deviation	(Standard Deviation / Average)*100
0,027387	6,050060	0,01779383 4	0,034556963	194,2075191 %

Ακολουθεί η γραφική απεικόνιση των αποτελεσμάτων της εκτέλεσης του προγράμματος στην οποία είχαμε το μικρότερο σφάλμα:



ΚΕΦΑΛΑΙΟ 4

Παράδειγμα πνευμονικών ήχων

4.1 Εφαρμογή του BDRNN στην ανάλυση των πνευμονικών ήχων

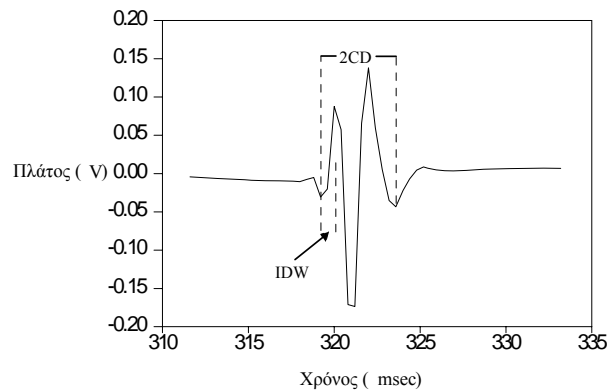
Στο παρόν κεφάλαιο αυτό θα παρουσιασθεί μία ακόμη εφαρμογή του μοντέλου BDRNN σε ένα πραγματικό πρόβλημα, το διαχωρισμό των παθολογικών διακοπτόμενων αναπνευστικών ήχων (**D**iscontinuous **A**dventitious **S**ounds, DAS) από τους κυψελιδικούς ήχους (**V**esicular **S**ounds, VS), που αποτελεί βασικό ζήτημα στην ανάλυση των πνευμονικών ήχων. Το μοντέλο εφαρμόζεται για τον αυτοματοποιημένο διαχωρισμό των DAS από τους VS, το οποίο βασίζεται στη μη στάσιμη φύση των πρώτων. Το φίλτρο τροφοδοτείται με ηχογραφημένα αναπνευστικά σήματα και στην έξοδο παράγεται η εκτίμηση του ήχου DAS.

4.1.1 Βασικά χαρακτηριστικά των αναπνευστικών ήχων και της σχετιζόμενης παθολογίας

Τα ακουστικά αναπνευστικά σήματα σχετίζονται με ανατομικές μεταβολές ή μεταβολές στη φυσιολογία του αναπνευστικού συστήματος, γεγονός που αντανακλάται στις ακουστικές ιδιότητες των τμημάτων ή των οργάνων, που επηρεάζονται από τις μεταβολές αυτές [A1]. Έτσι, όταν π.χ. εμφανίζεται μία στένωση του τραχειοβρογχικού δένδρου του ανθρώπου, παράγεται μία στροβιλώδης ροή του αέρα, που οδηγεί στην γένεση ενός συγκεκριμένου ακουστικού θορύβου, καλούμενου ως επιπρόσθετος ήχος (adventitious sound), καθώς προστίθεται στο φυσιολογικό αναπνευστικό ήχο, ο οποίος καλείται κυψελιδικός ήχος.

Σύμφωνα με τη βιβλιογραφία, οι επιπρόσθετοι ήχοι, οι οποίοι ακούγονται μόνο σε παθολογικές περιπτώσεις, διακρίνονται σε δύο μείζονες κατηγορίες: τους συνεχείς και τους διακοπτόμενους. Η πρώτη κατηγορία αποτελείται από τους συριγμούς (wheezes) και τους ρόγχους (rhonchi), ηχητικά σήματα που χαρακτηρίζονται από σχετικά μικρή διάρκεια (250 msec) και μία οξεία αιχμή στο φάσμα ισχύος, στην περιοχή των 400 Hz για τους συριγμούς και των 200 Hz ή λιγότερο για τους ρόγχους. Η δεύτερη κατηγορία περιλαμβάνει τους τρίζοντες (crackles) και τα «κακαρίσματα»² (squawks, SQ), ήχοι διακρινόμενοι κυρίως από τα χαρακτηριστικά τους στο πεδίο του χρόνου (βλ. Σχ. 4.1): α) τη μικρή διάρκεια (20 msec), β) το αρχικό πλάτος βύθισης (Initial Deflection Width, IDW), δηλαδή το χρονικό διάστημα που διαρκεί η πρώτη βύθιση του τρίζοντος αναπνευστικού ήχου, και γ) τη διάρκεια δύο περιόδων (two Cycle Duration, 2CD), που είναι το χρονικό διάστημα των δύο πρώτων περιόδων του τρίζοντος αναπνευστικού ήχου.

² Η ονομασία «κακάρισμα», παρά την ιδιομορφία της αποδίδει κατά τον πλέον ακριβή τρόπο τη φύση του συγκεκριμένου αναπνευστικού ήχου.



Σχήμα 4.1

Ορισμός των IDW και 2CD των τριζόντων επιπρόσθετων αναπνευστικών ήχων.

Παράδειγμα ενός λεπτού τρίζοντος.

Οι τρίζοντες είναι διακριτοί, μη μουσικοί ήχοι, οι οποίοι συμπεριφέρονται ως μη στάσιμος εκρηκτικός θόρυβος, επικαθήμενος στον αναπνευστικό ήχο. Ταξινομούνται στους λεπτούς (fine crackles, FC) και τους τραχείς τρίζοντες (coarse crackles, CC). Οι πρώτοι παρουσιάζουν IDW=0.90 msec και 2CD=6.0 msec ενώ οι αντίστοιχες τιμές για τους δεύτερους είναι IDW=1.25 msec; 2CD=9.50 msec. Οι λεπτοί τρίζοντες είναι αποκλειστικά ήχοι της εισπνοής. Εμφανίζονται κοντά στη μέση και προς το τέλος της εισπνοής, και επαναλαμβάνονται με τα ίδια ακουστικά χαρακτηριστικά σε συνεχόμενες αναπνοές. Συνδέονται τόσο με πνευμονικές συμφορήσεις λόγω καρδιακής ανεπάρκειας όσο και με πνευμονικές ινώσεις όπως η αμυάντωση (asbestosis) και η διάμεση ίνωση (interstitial fibrosis). Οι τραχείς τρίζοντες εκδηλώνονται από την αρχή της αναπνοής και περιστασιακά στην εκπνοή. Είναι λιγότερο επαναλαμβανόμενοι από αναπνοή σε αναπνοή, σε σχέση με τους λεπτούς τρίζοντες, γιατί η παραγωγή τους στηρίζεται σε συχνά μεταβαλλόμενους μηχανισμούς [A2]. Σχετίζονται με χρόνιες βρογχίτιδες.

Ένα πρόσθετο στοιχείο, που διαφοροποιεί τους λεπτούς από τους τραχείς τρίζοντες είναι οι περιοχές συχνοτήτων, στις οποίες εμφανίζεται αύξηση του φασματικού τους περιεχομένου. Έτσι, οι λεπτοί τρίζοντες εμφανίζουν μία κορυφή στην περιοχή των 650 Hz ενώ οι τραχείς τρίζοντες σε χαμηλότερη ζώνη συχνοτήτων, γύρω στα 250 Hz [A5].

Το «κακάρισμα» είναι ένας συνδυασμός συριγμού και τρίζοντος αναπνευστικού ήχου, όπου το πλάτος του αποσβαίνει πολύ γρήγορα. Αν και εμφανίζεται ως μικρός εισπνευστικός συριγμός, ακούγεται σε συνδυασμό με λεπτούς τρίζοντες, ως ήχος του τέλους της αναπνοής. Οι ήχοι αυτοί σχετίζονται με την αλλεργική κυψελίτιδα (allergic alveolitis) και τη διάμεση ίνωση.

4.1.2 Το φίλτρο BDRNN

Το BDRNN τροφοδοτείται με το σήμα εισόδου $u(k)$, το οποίο είναι ο πνευμονικός ήχος, κανονικοποιημένος και με μηδενική μέση τιμή. Η έξοδος του δικτύου είναι ο προσεγγιστικός ήχος DAS.

Το φίλτρο εφαρμόζεται σε σήματα που ελήφθησαν από διεθνείς βάσεις δεδομένων αναπνευστικών ήχων. Αναλύονται ήχοι FC από ασθενείς με πνευμονική και διάμεση ίνωση, ήχοι CC από ασθενείς με χρόνια βρογχίτιδα και ήχοι SQ από ασθενείς με διάμεση ίνωση και αλλεργική κυψελίτιδα. Συγκεκριμένα χρησιμοποιούνται δέκα περιπτώσεις από κάθε κατηγορία (FC, CC και SQ) για το σχηματισμό των συνόλων δεδομένων εκμάθησης, ενώ για τα σύνολα δεδομένων ελέγχου είναι διαθέσιμες έξι περιπτώσεις ασθενών με FC, πέντε με CC και πέντε με SQ, αντίστοιχα. Ως επιθυμητές έξοδοι θεωρούνται οι έξοδοι του φίλτρου WTST-NST ([A3]).

Η διαδικασία εκπαίδευσης διεξάγεται ανεξάρτητα για κάθε κατηγορία σήματος, οδηγώντας σε τρία BDRNN. Για κάθε δίκτυο και κατηγορία σήματος εξετάζεται ένας μεγάλος αριθμός συνδυασμών παραμέτρων εκπαίδευσης, και τα αποτελέσματα παρατίθενται στους πίνακες που ακολουθούν.

ΔΟΚΙΜΕΣ

Πίνακας 2

Συνδυασμός παραμέτρων και αποτελέσματα για την περίπτωση του πνευμονικού ήχου CC

Αριθμός Επανάληψης	n_hid	alpha stability	n2	ep_size	n+	n-	max_iter	RMSE (error)
1	12	6	2	4096	1,05	0,5	4000	0,024011
2	12	6	2	4096	1,05	0,7	4000	0,016783
3	12	6	2	4096	1,05	0,9	4000	0,007772
4	12	6	2	4096	1,15	0,5	4000	0,008559
5	12	6	2	4096	1,15	0,7	4000	0,004925
6	12	6	2	4096	1,15	0,9	4000	0,005056
7	12	6	2	4096	1,25	0,5	4000	0,031529
8	12	6	2	4096	1,25	0,7	4000	0,006423
9	12	6	2	4096	1,25	0,9	4000	0,005312
10	12	6	6	4096	1,05	0,5	4000	0,010835
11	12	6	6	4096	1,05	0,7	4000	0,033348
12	12	6	6	4096	1,05	0,9	4000	0,067793
13	12	6	6	4096	1,15	0,5	4000	0,047044
14	12	6	6	4096	1,15	0,7	4000	0,014615
15	12	6	6	4096	1,15	0,9	4000	0,008118
16	12	6	6	4096	1,25	0,5	4000	0,010157
17	12	6	6	4096	1,25	0,7	4000	0,025747
18	12	6	6	4096	1,25	0,9	4000	0,025467
19	18	9	2	4096	1,05	0,5	4000	0,008908
20	18	9	2	4096	1,05	0,7	4000	0,016994
21	18	9	2	4096	1,05	0,9	4000	0,023601
22	18	9	2	4096	1,15	0,5	4000	0,010386
23	18	9	2	4096	1,15	0,7	4000	0,006211
24	18	9	2	4096	1,15	0,9	4000	0,012316
25	18	9	2	4096	1,25	0,5	4000	0,042614
26	18	9	2	4096	1,25	0,7	4000	0,011439
27	18	9	2	4096	1,25	0,9	4000	0,045924
28	18	9	6	4096	1,05	0,5	4000	0,004999
29	18	9	6	4096	1,05	0,7	4000	0,020116
30	18	9	6	4096	1,05	0,9	4000	0,006553
31	18	9	6	4096	1,15	0,5	4000	0,019307
32	18	9	6	4096	1,15	0,7	4000	0,010528
33	18	9	6	4096	1,15	0,9	4000	0,011902
34	18	9	6	4096	1,25	0,5	4000	0,015421
35	18	9	6	4096	1,25	0,7	4000	0,01401
36	18	9	6	4096	1,25	0,9	4000	0,041839
37	24	12	2	4096	1,05	0,5	4000	0,039648
38	24	12	2	4096	1,05	0,7	4000	0,033219
39	24	12	2	4096	1,05	0,9	4000	0,015429
40	24	12	2	4096	1,15	0,5	4000	0,053026
41	24	12	2	4096	1,15	0,7	4000	0,024571
42	24	12	2	4096	1,15	0,9	4000	0,04205
43	24	12	2	4096	1,25	0,5	4000	0,02701
44	24	12	2	4096	1,25	0,7	4000	0,009045
45	24	12	2	4096	1,25	0,9	4000	0,024411
46	24	12	6	4096	1,05	0,5	4000	0,035346

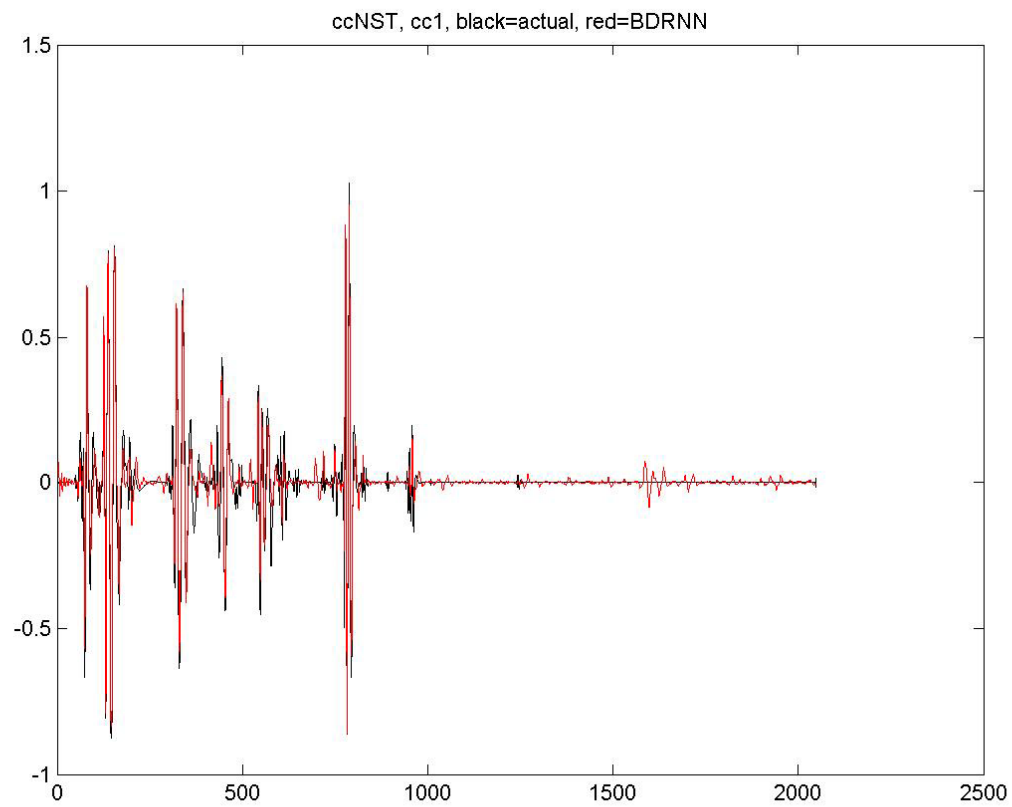
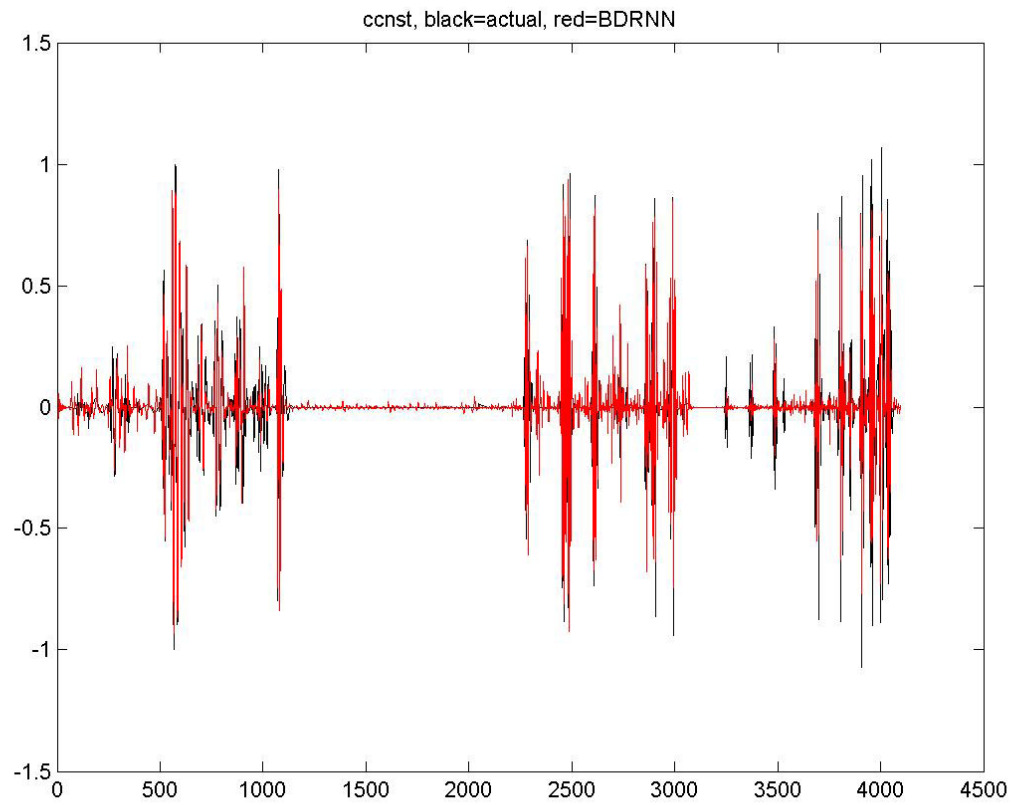
47	24	12	6	4096	1,05	0,7	4000	0,005709
48	24	12	6	4096	1,05	0,9	4000	0,006832
49	24	12	6	4096	1,15	0,5	4000	0,006256
50	24	12	6	4096	1,15	0,7	4000	0,00513
51	24	12	6	4096	1,15	0,9	4000	0,029521
52	24	12	6	4096	1,25	0,5	4000	0,015183
53	24	12	6	4096	1,25	0,7	4000	0,018102
54	24	12	6	4096	1,25	0,9	4000	0,027772

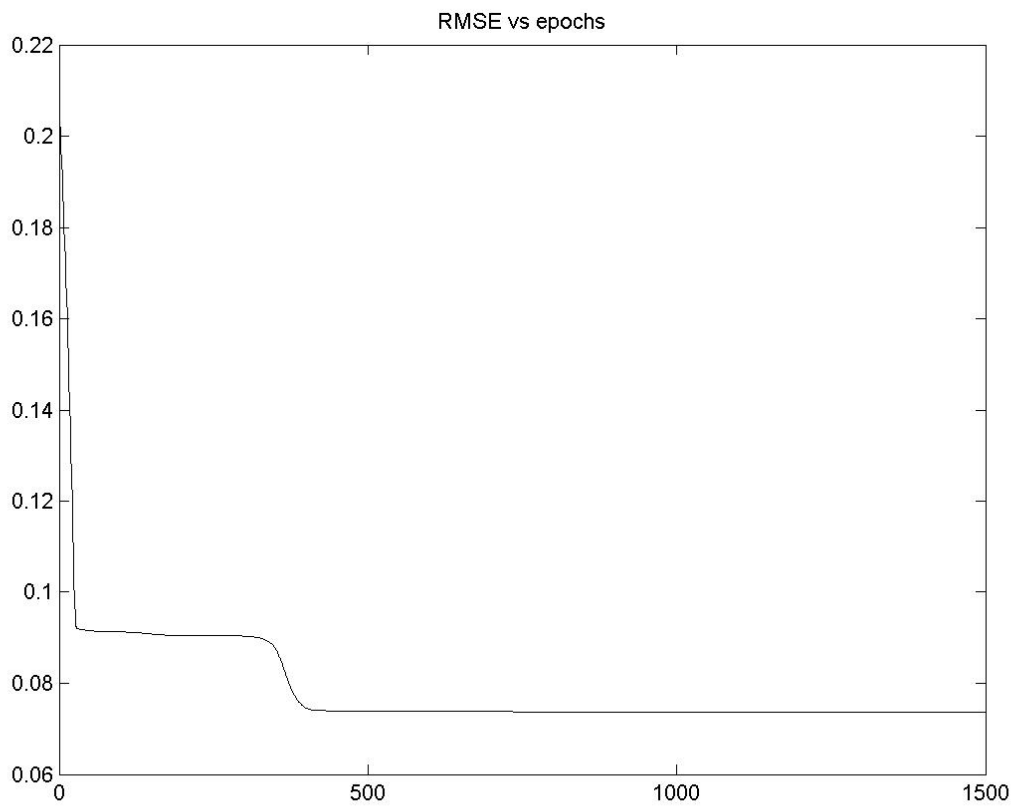
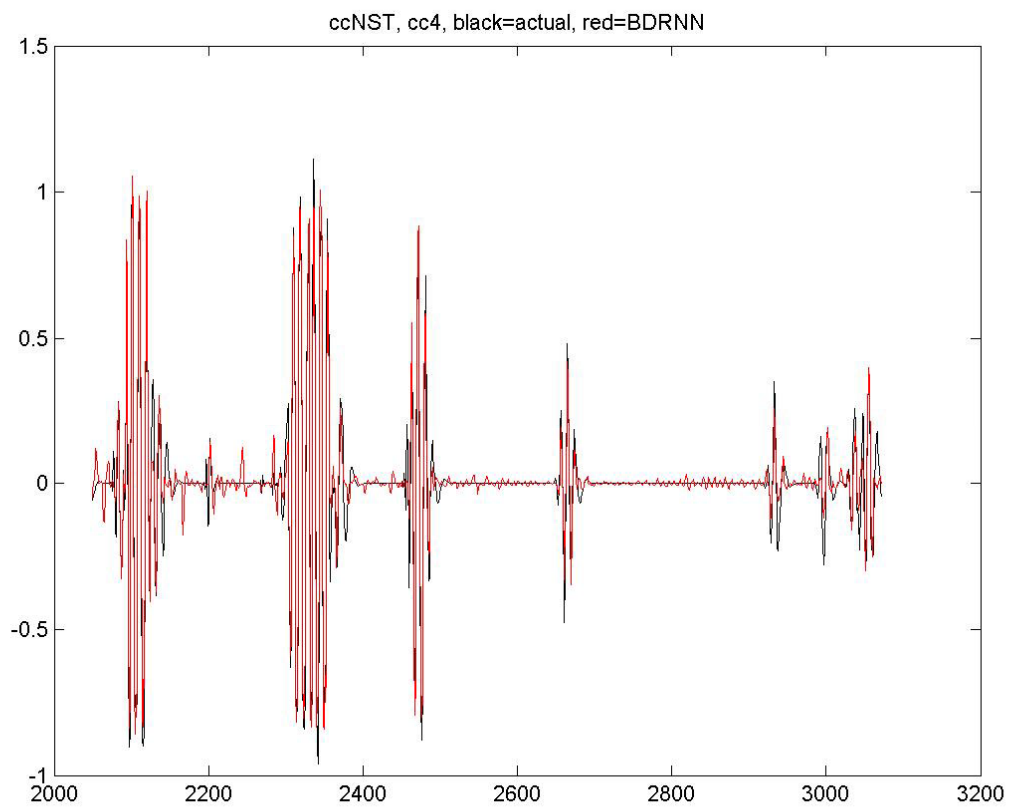
Πίνακας 2.1

Ελάχιστο MSE	Μέγιστο MSE	Μέσος Όρος	Τυπική Απόκλιση	(Τυπική απόκλιση / Μέσο Όρος)*100
0,004925	0,067793	0,020274	0,014603	72,02602 %

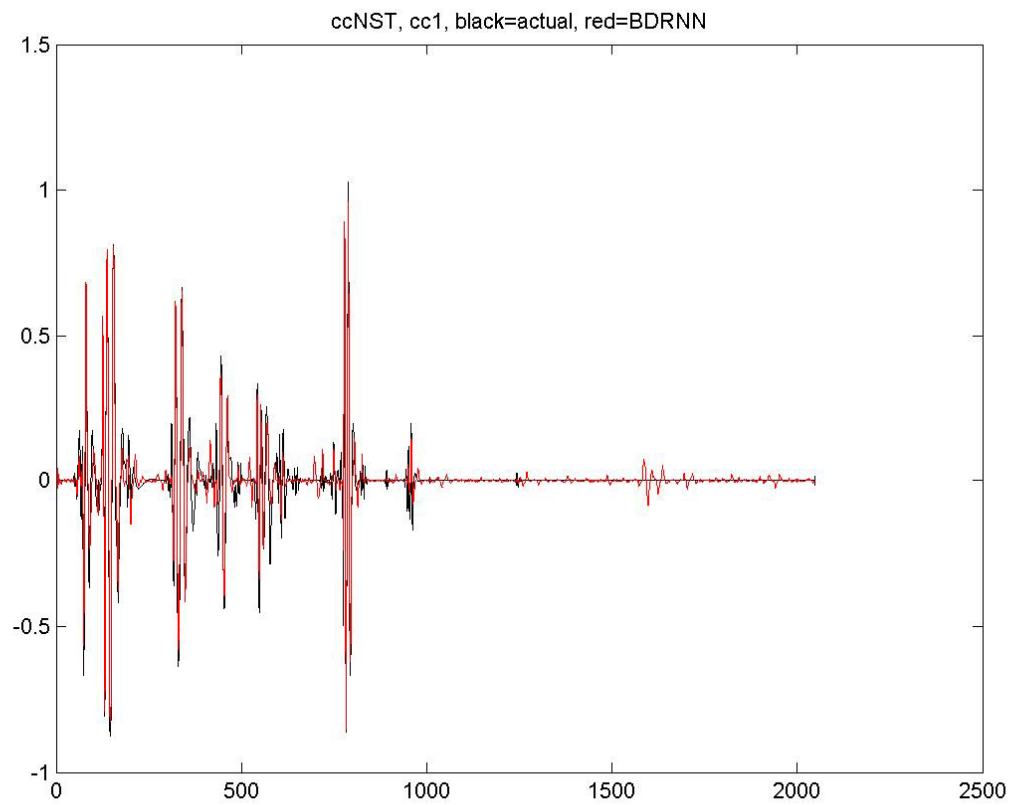
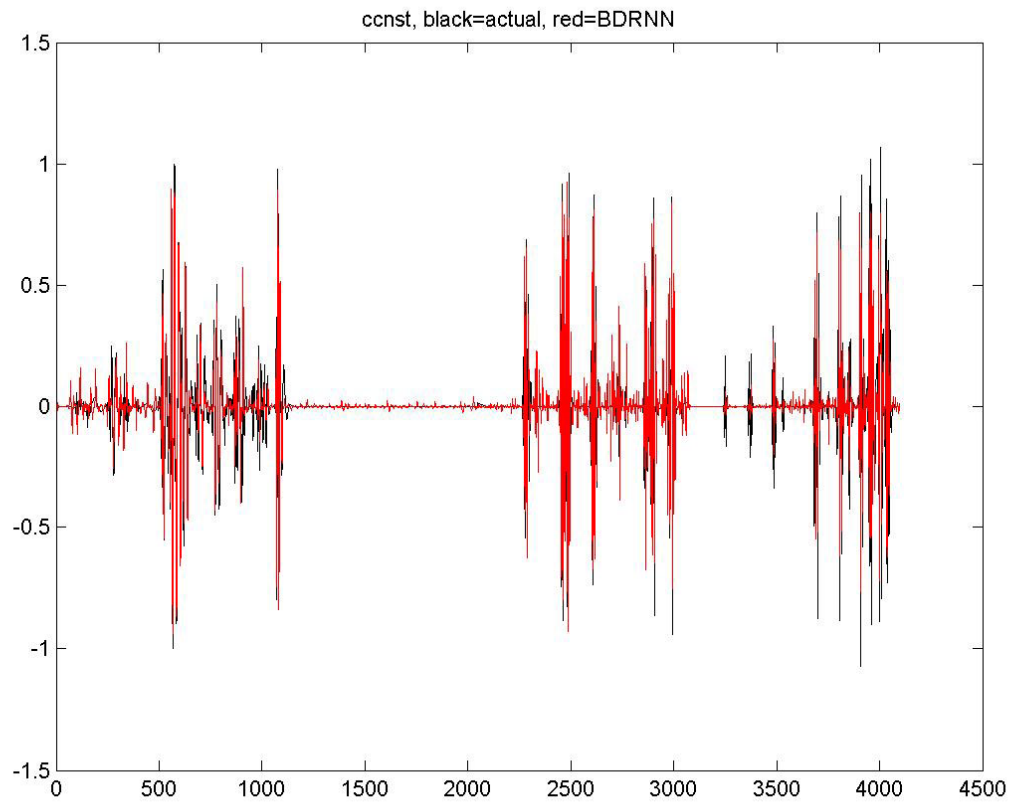
Στις επόμενες σελίδες ακολουθούν οι γραφικές απεικονίσεις των αποτελεσμάτων για κάθε ένα από τους 54 συνδυασμούς παραμέτρων εκπαίδευσης που εκτελέστηκαν για την περίπτωση του πνευμονικού ήχου CC.

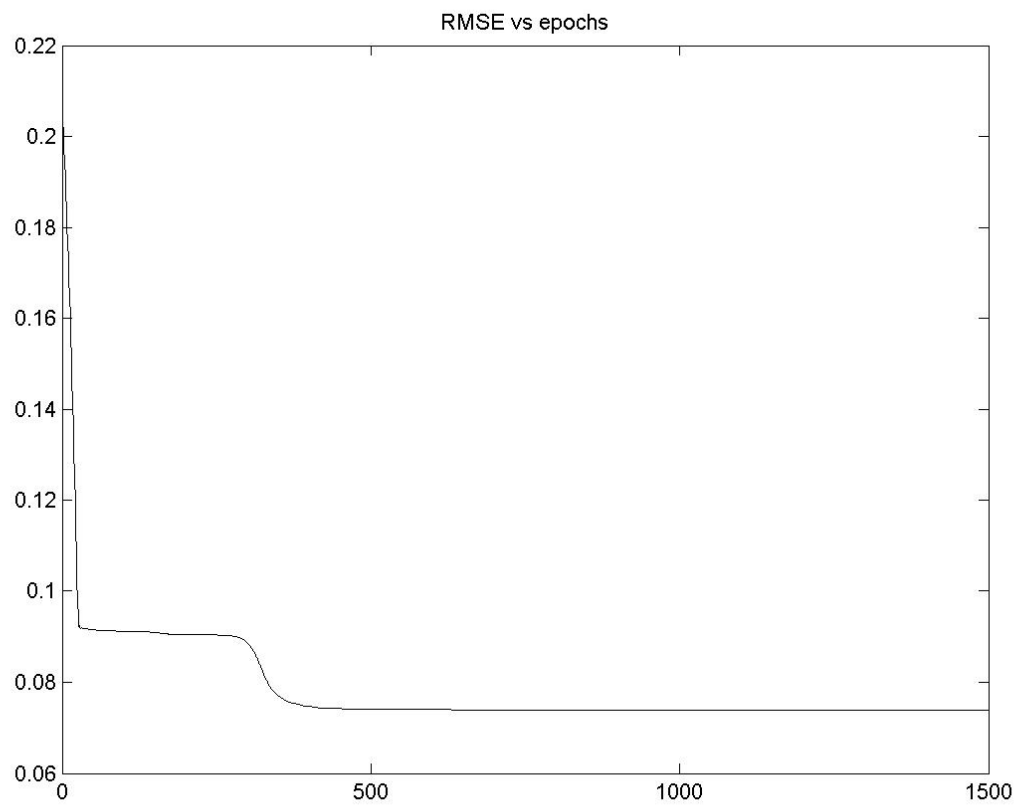
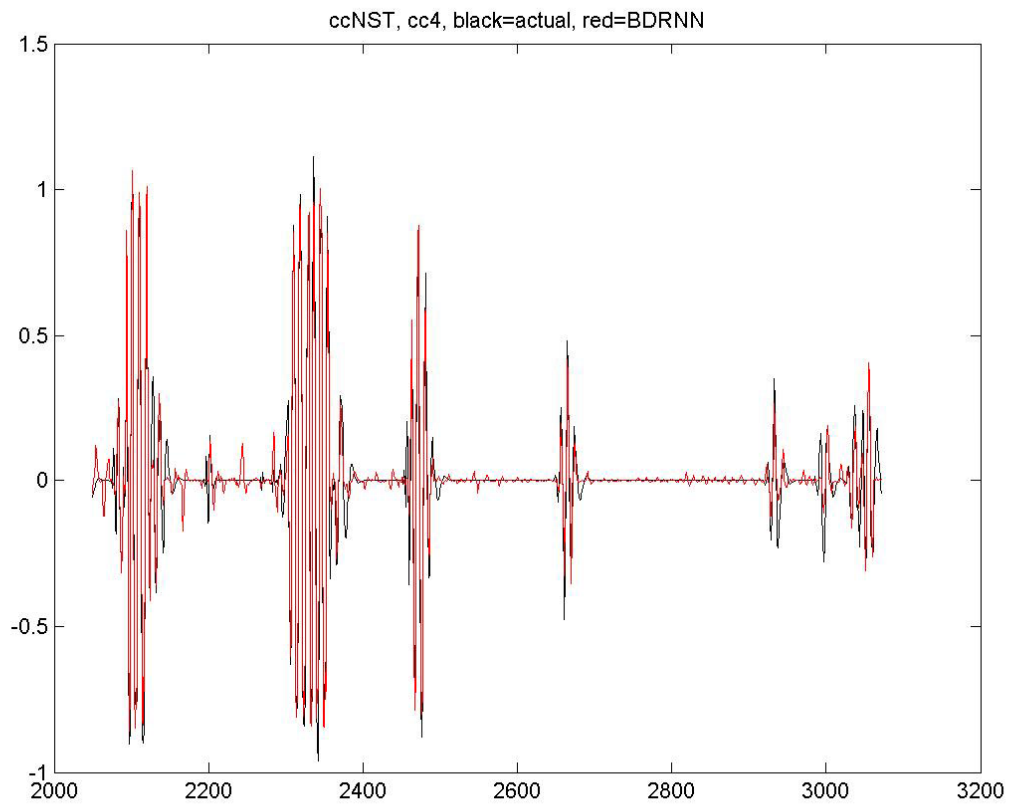
1^η Επανάληψη

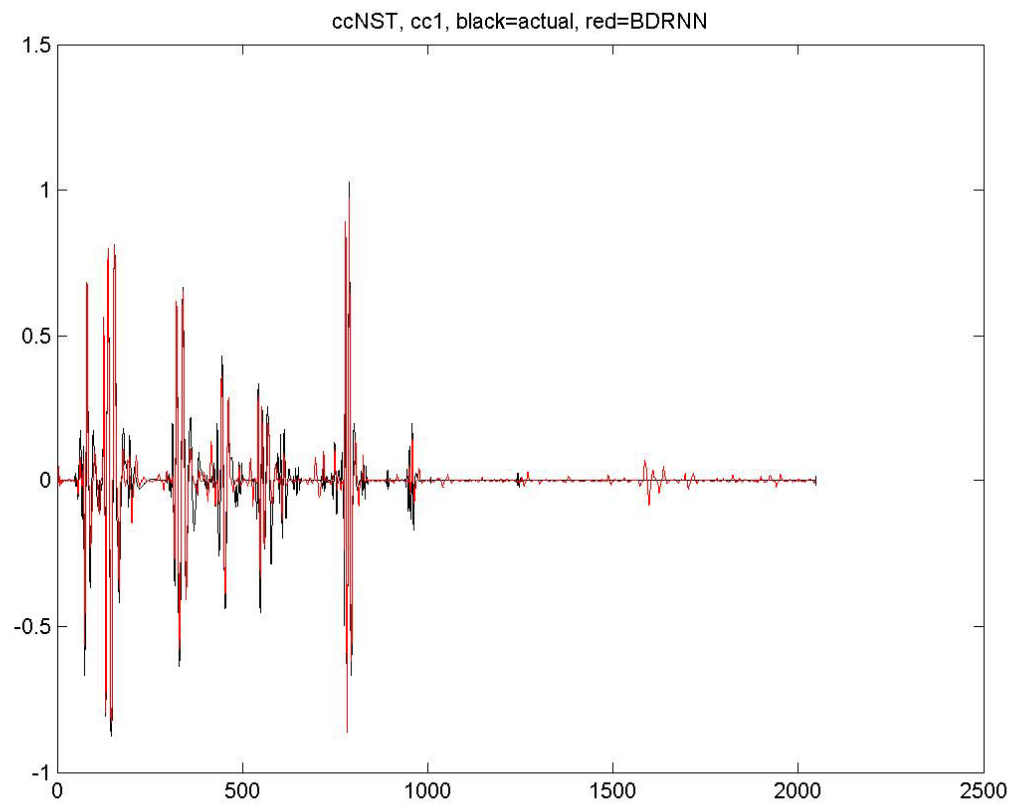
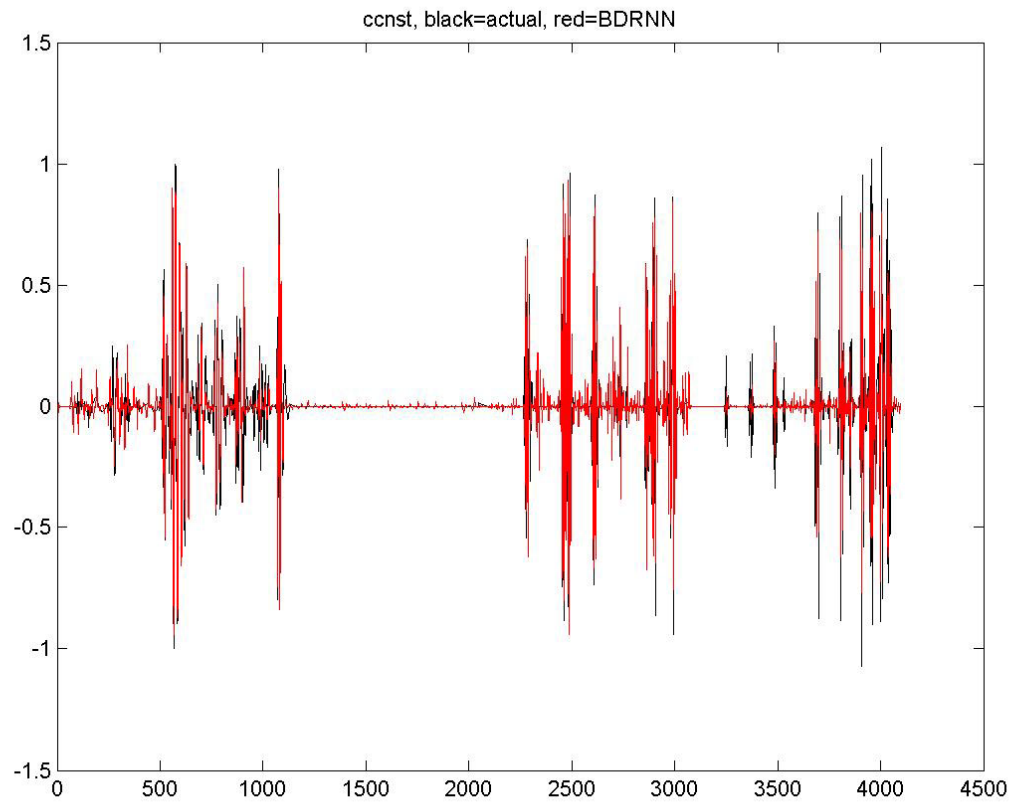


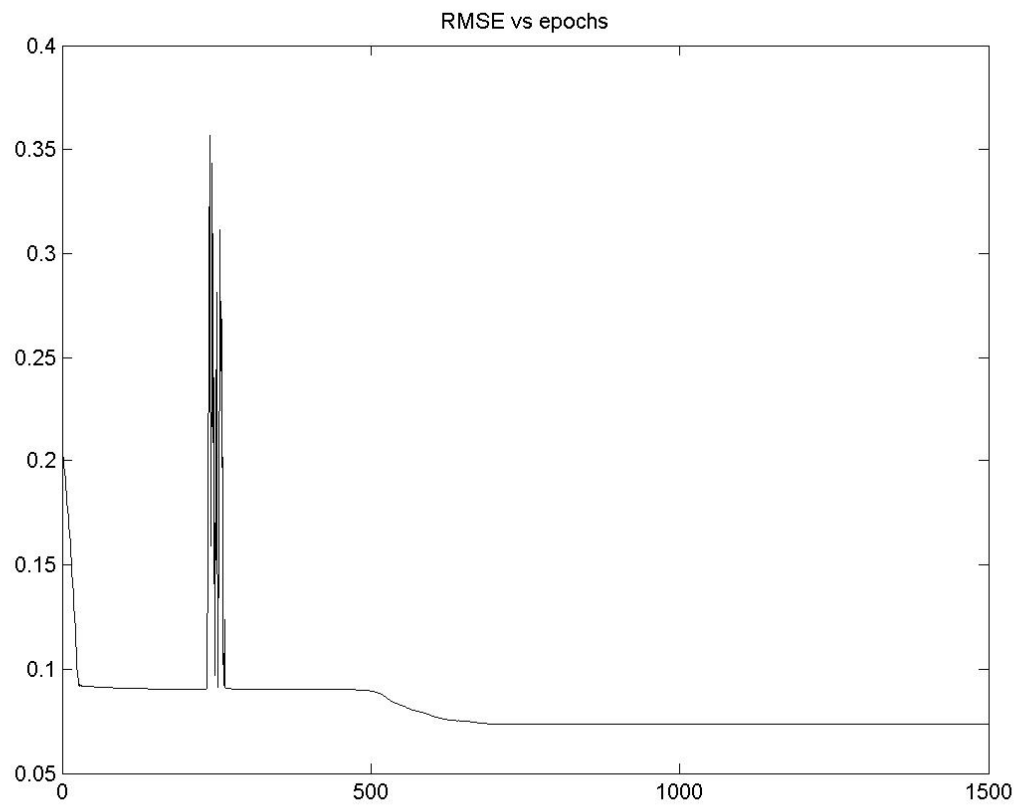
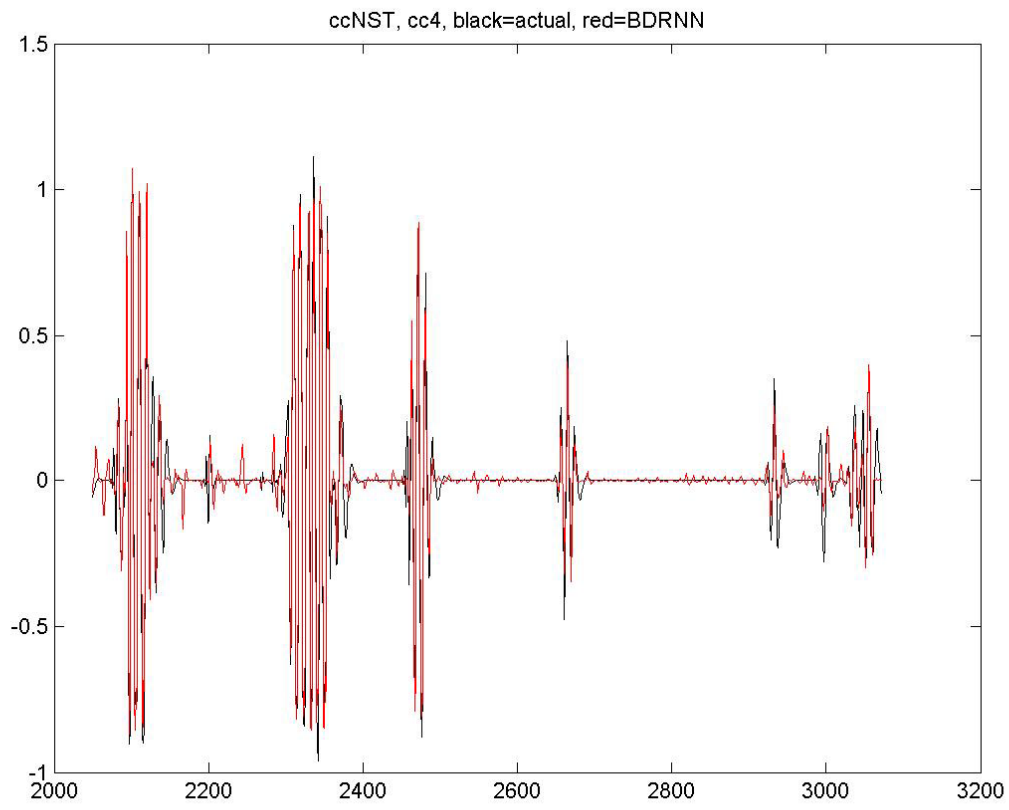


2^η Επανάληψη

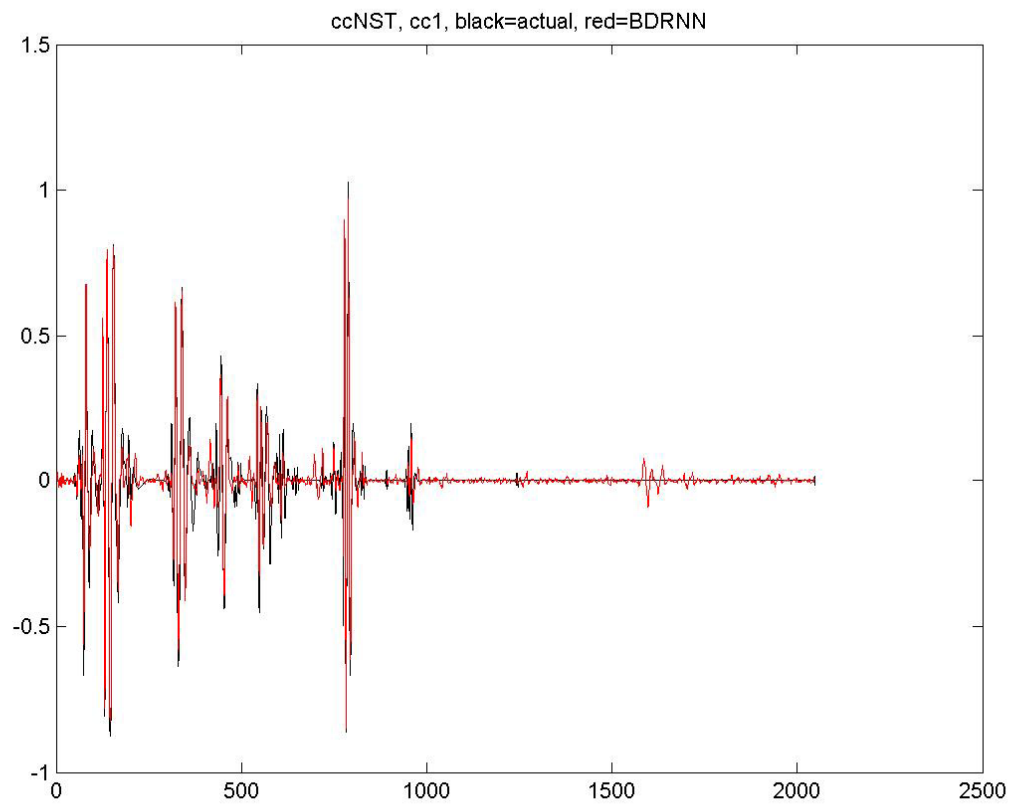
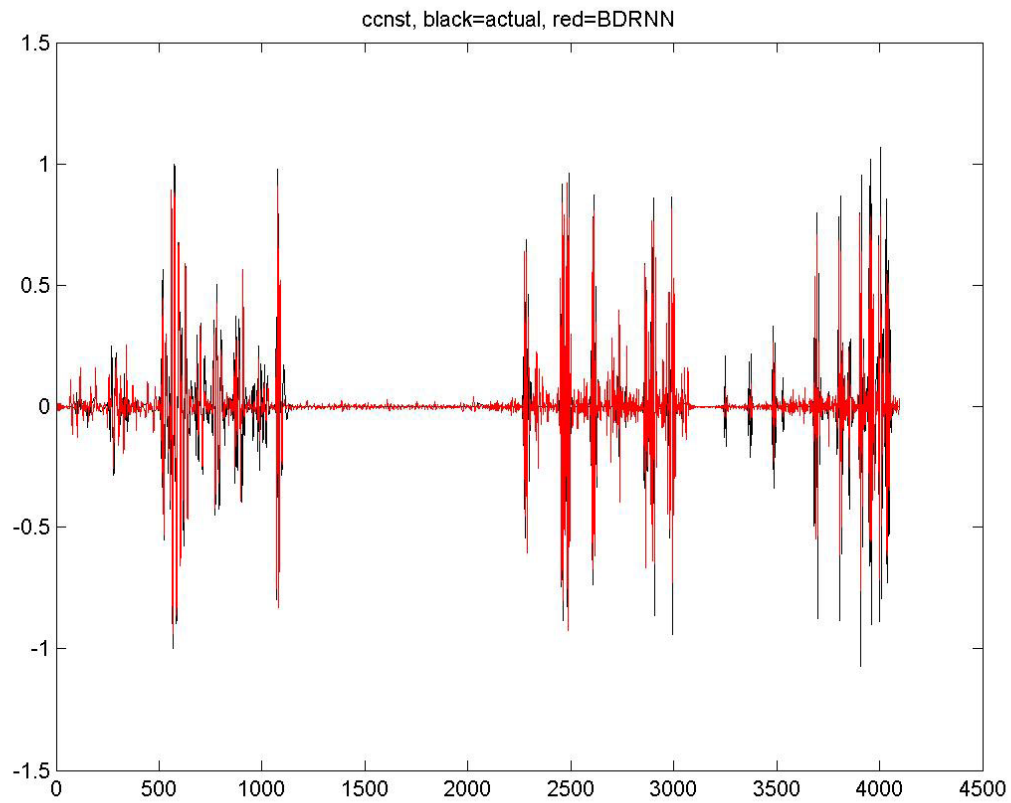


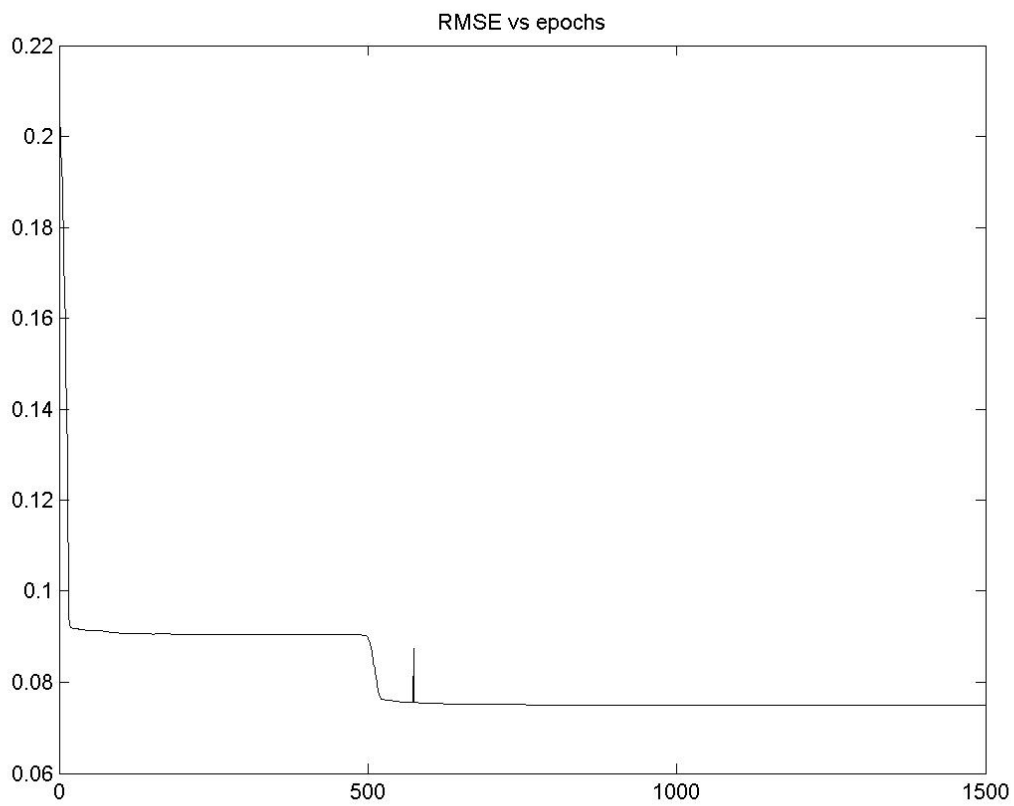
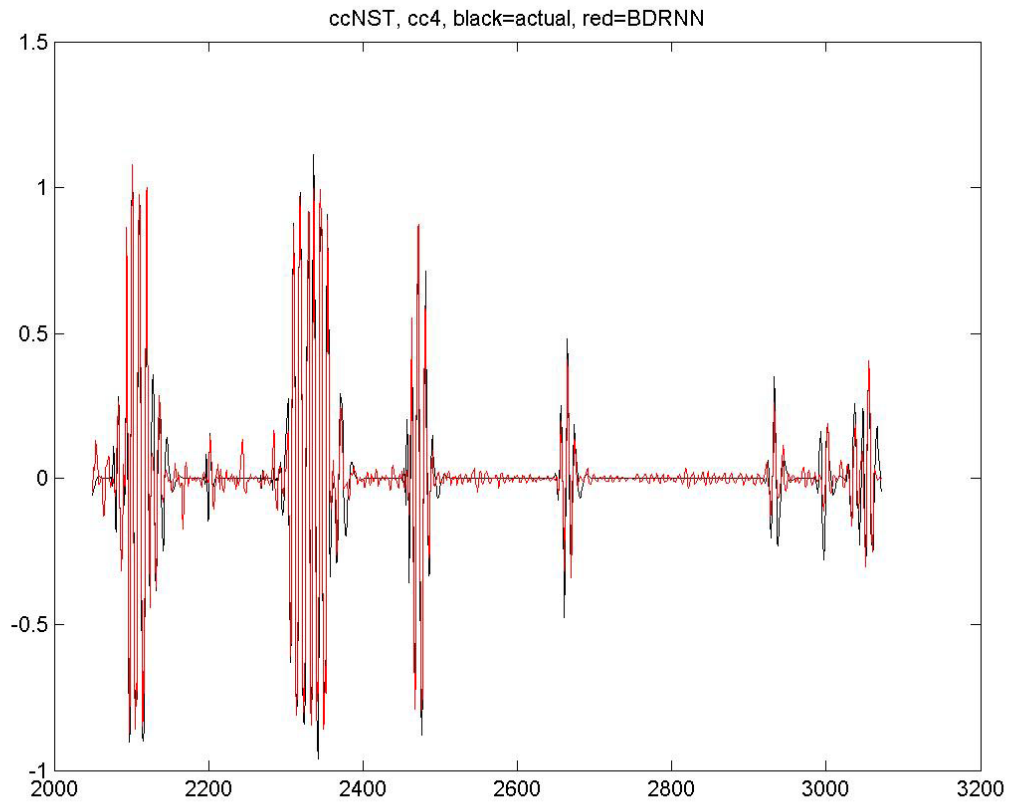


3^η Επανάληψη

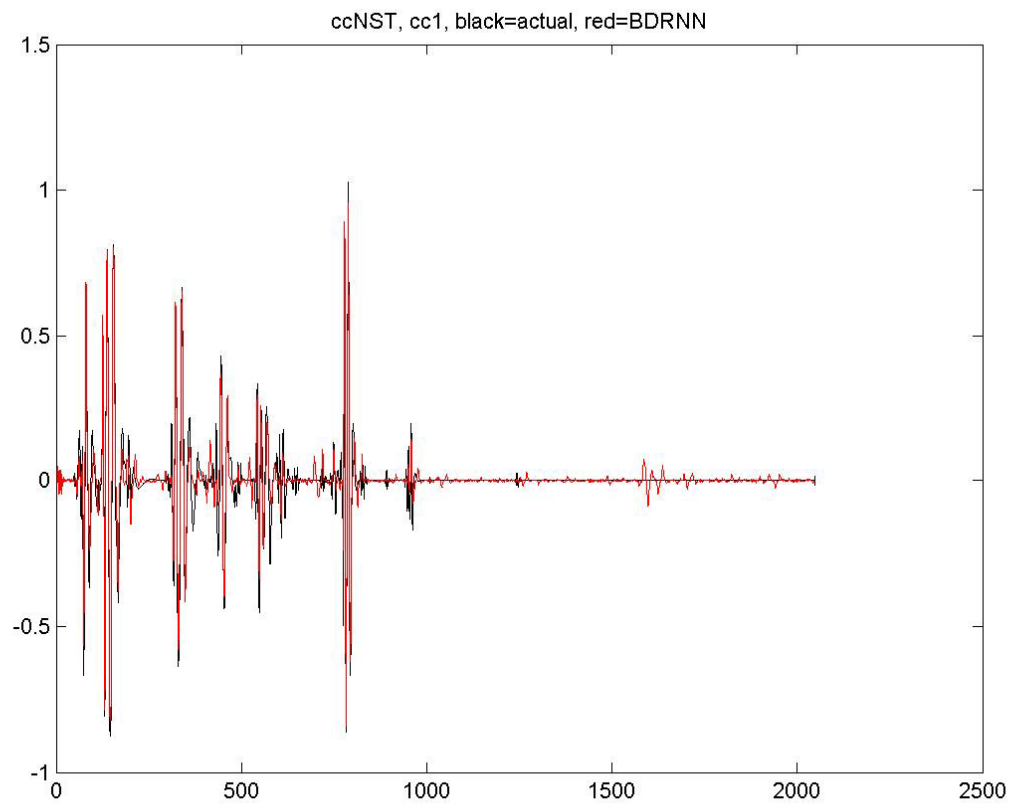
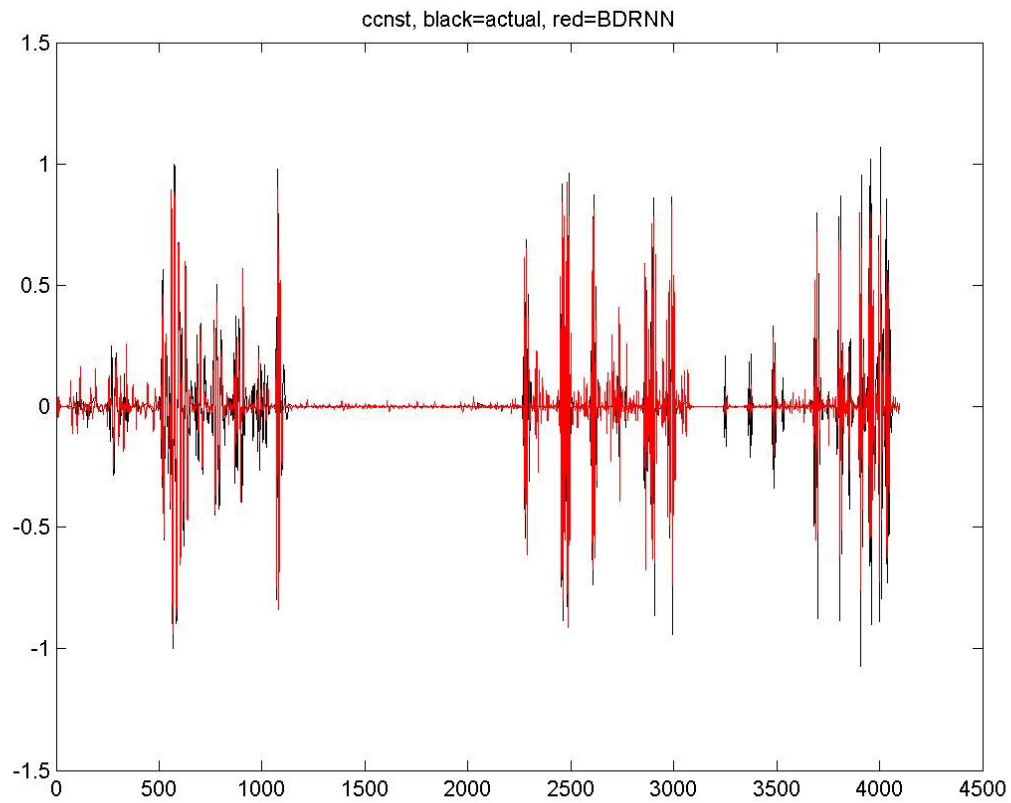


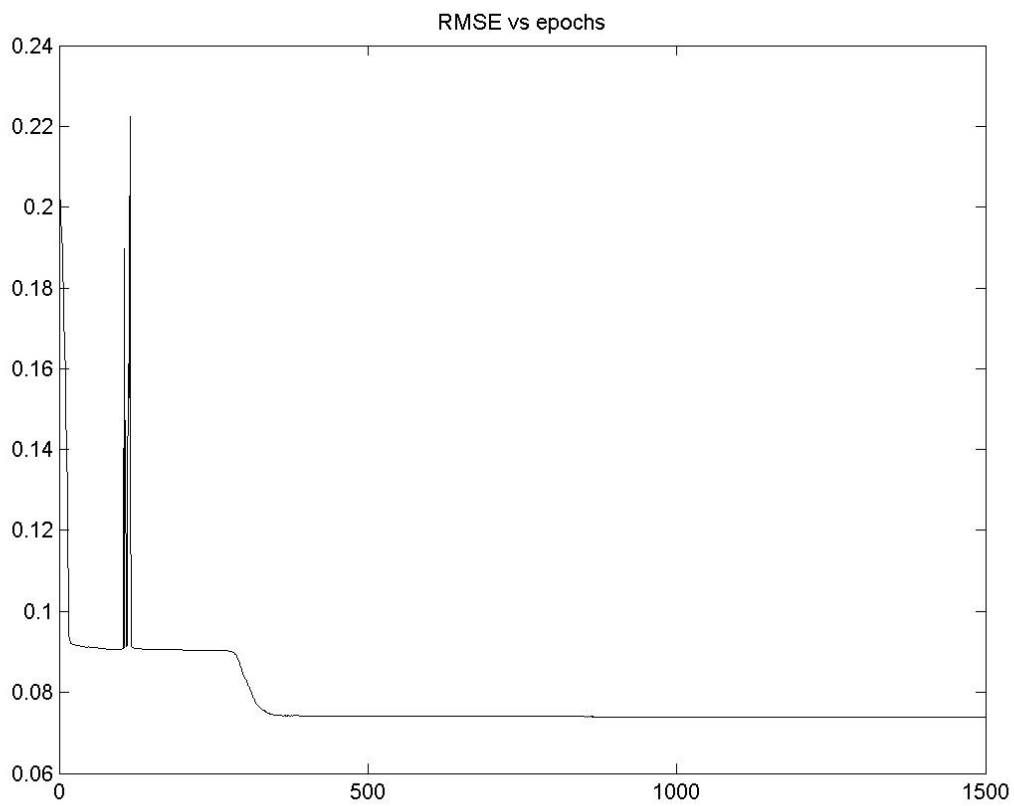
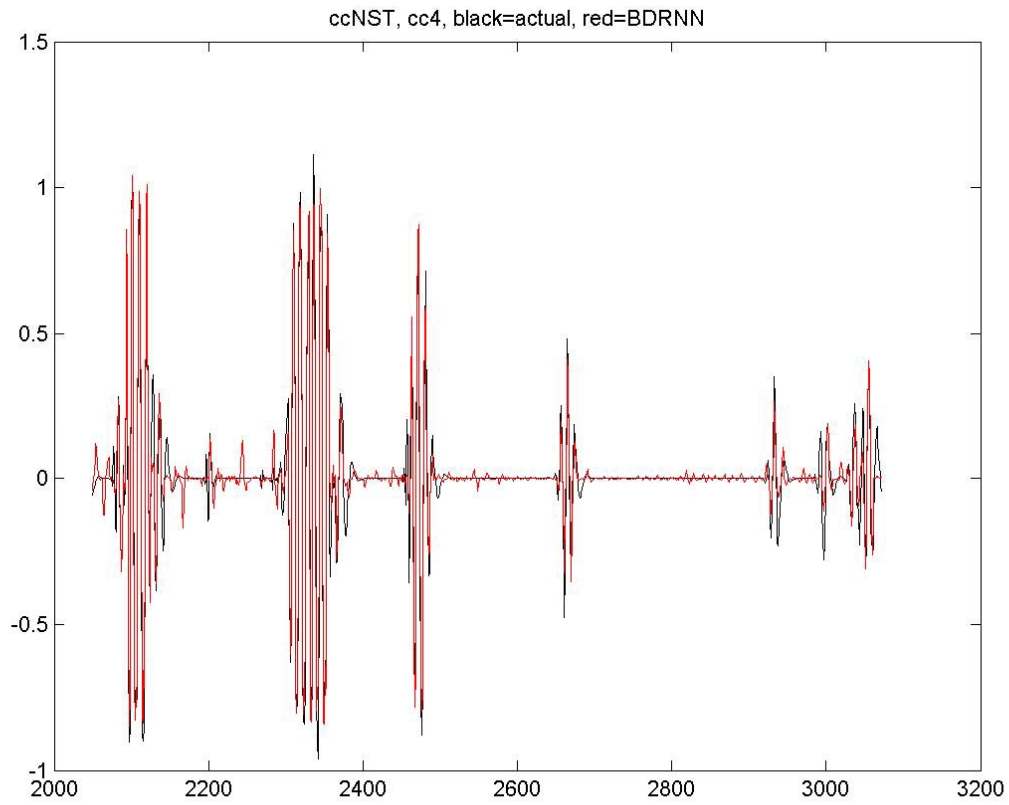
4^η Επανάληψη



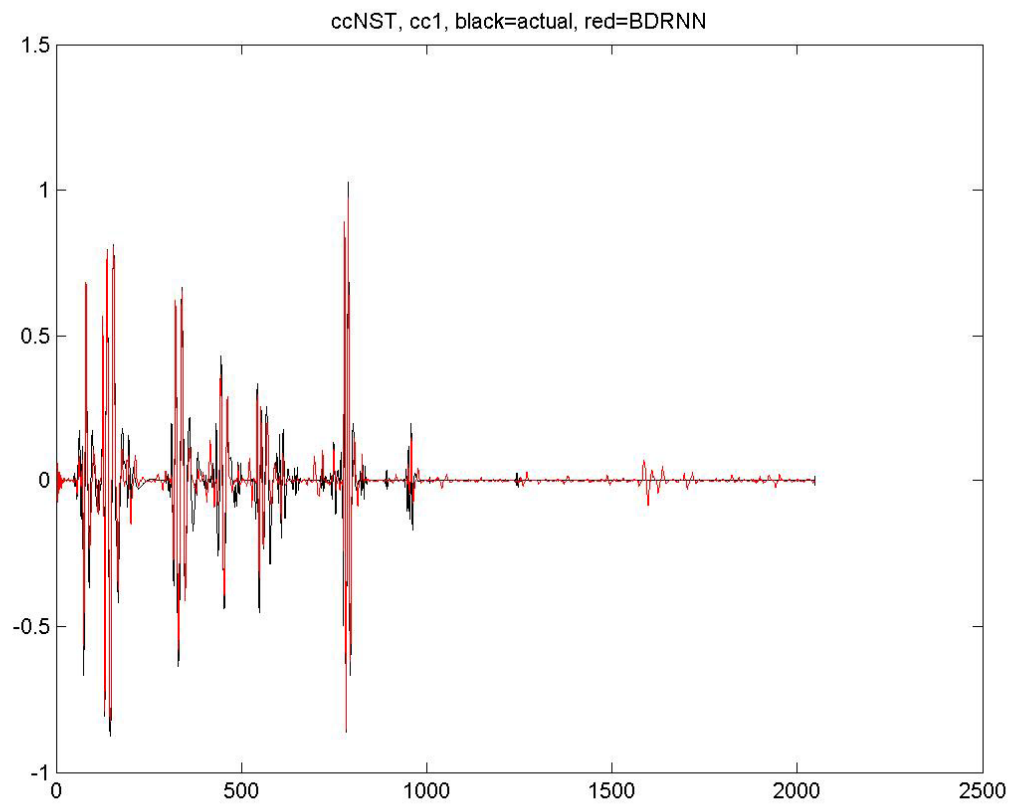
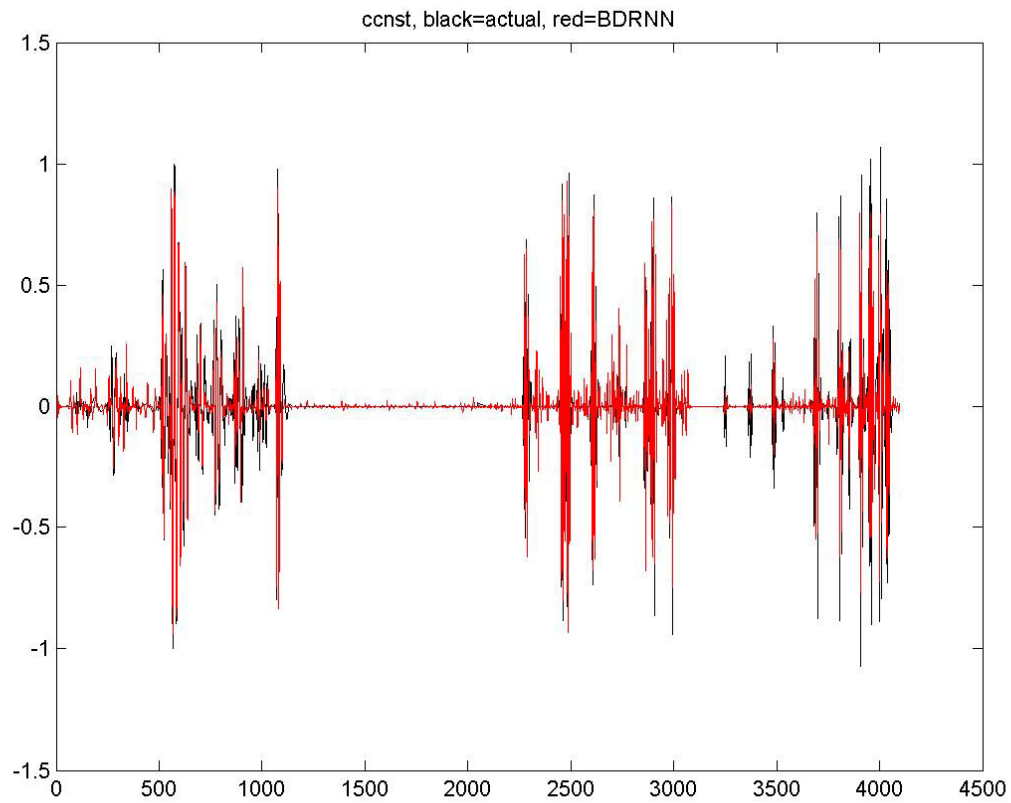


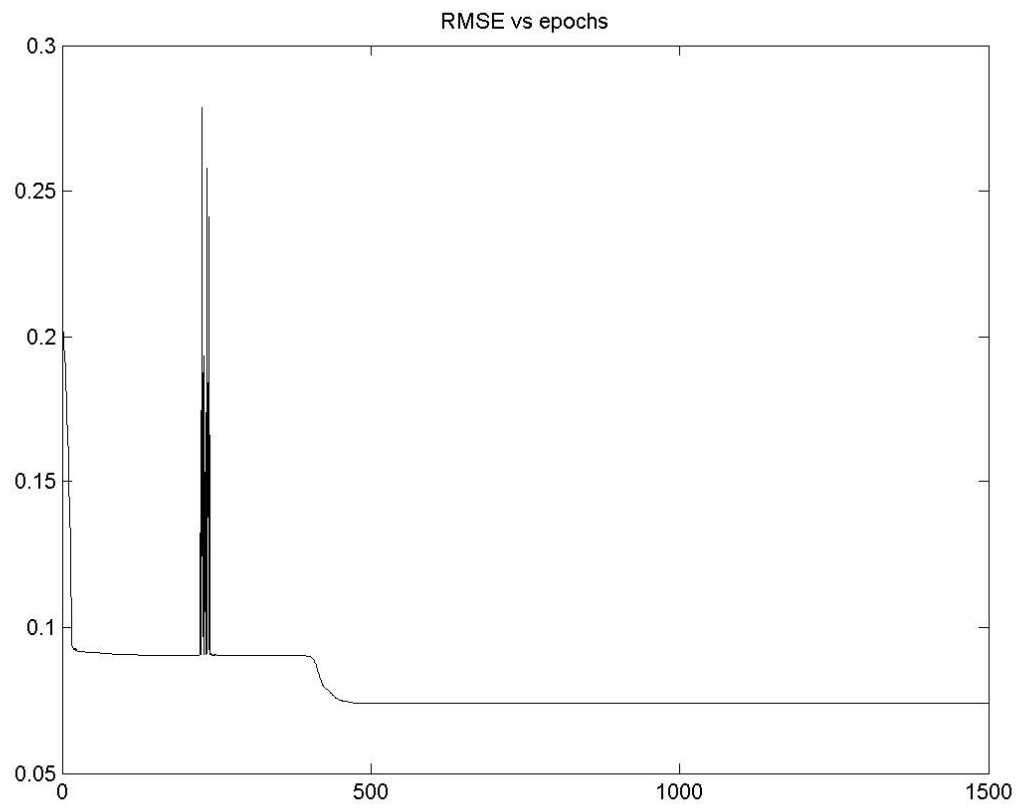
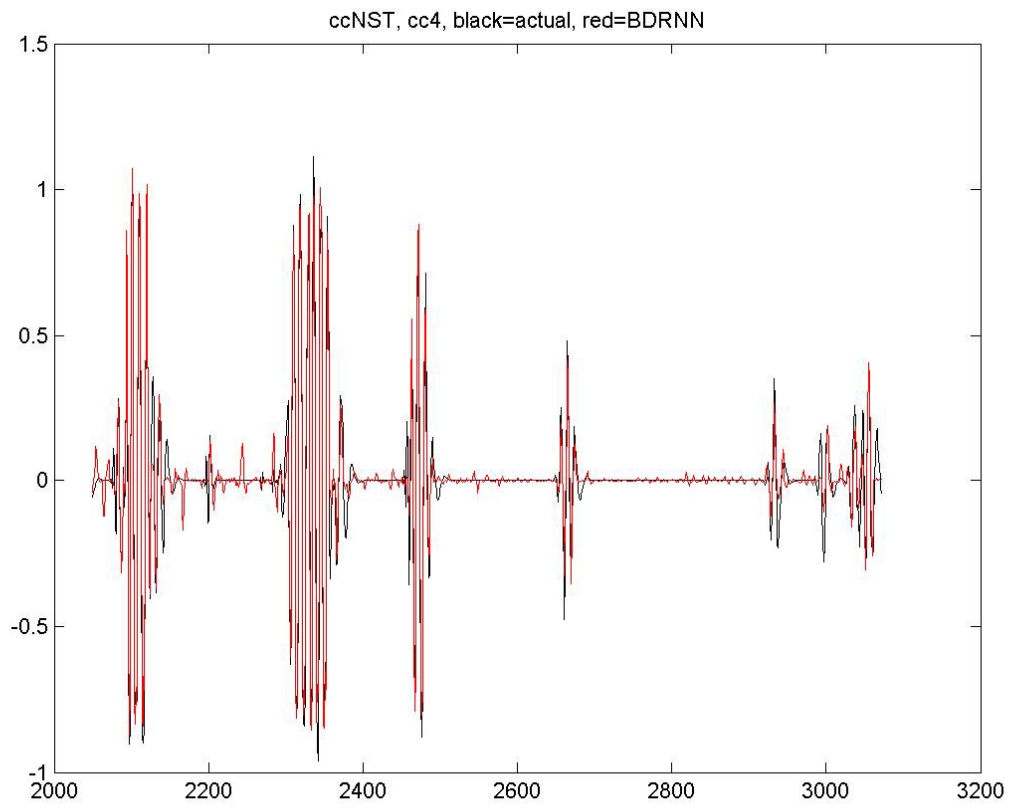
5^η Επανάληψη



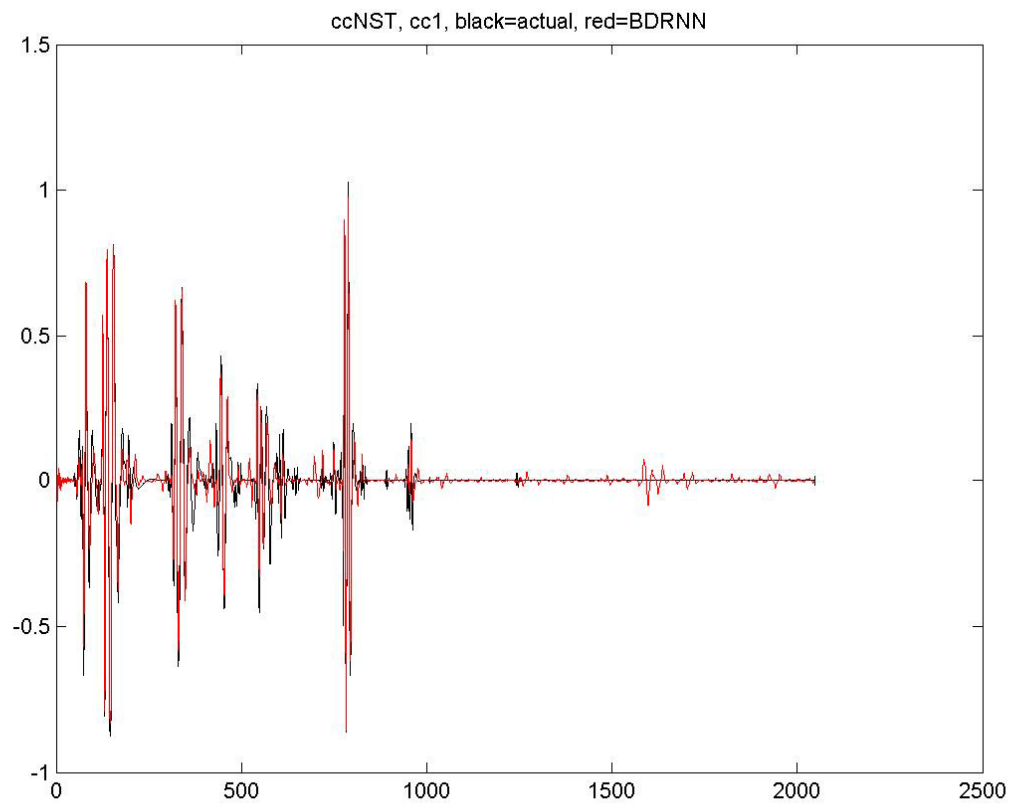
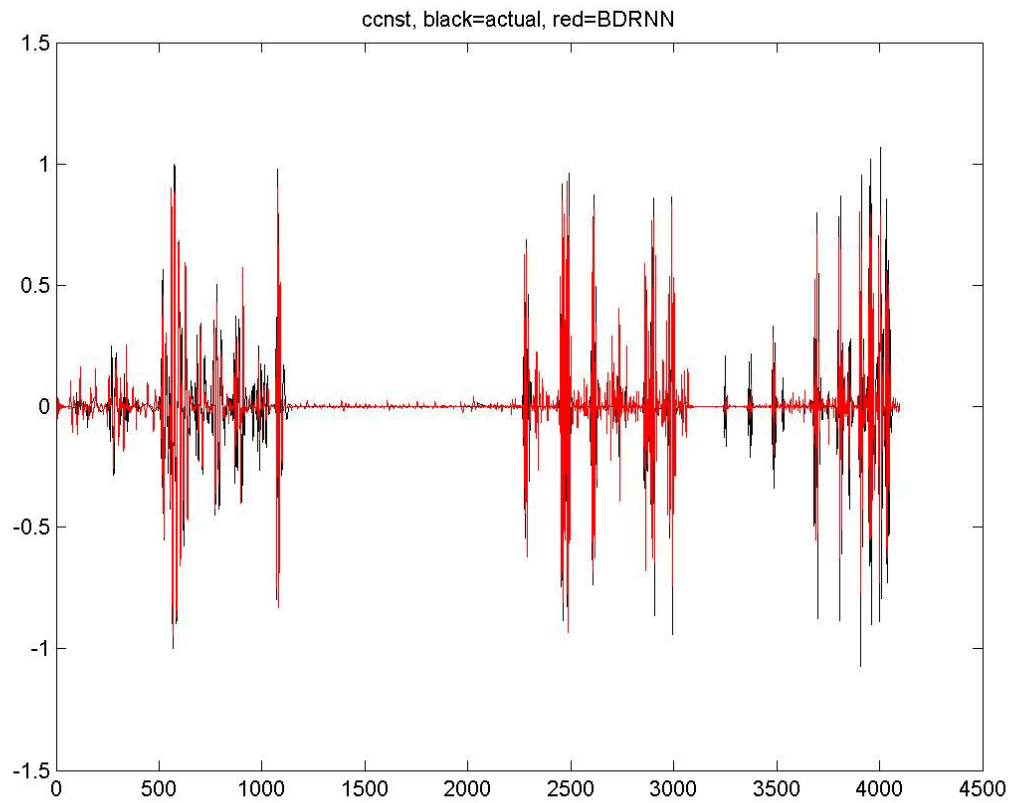


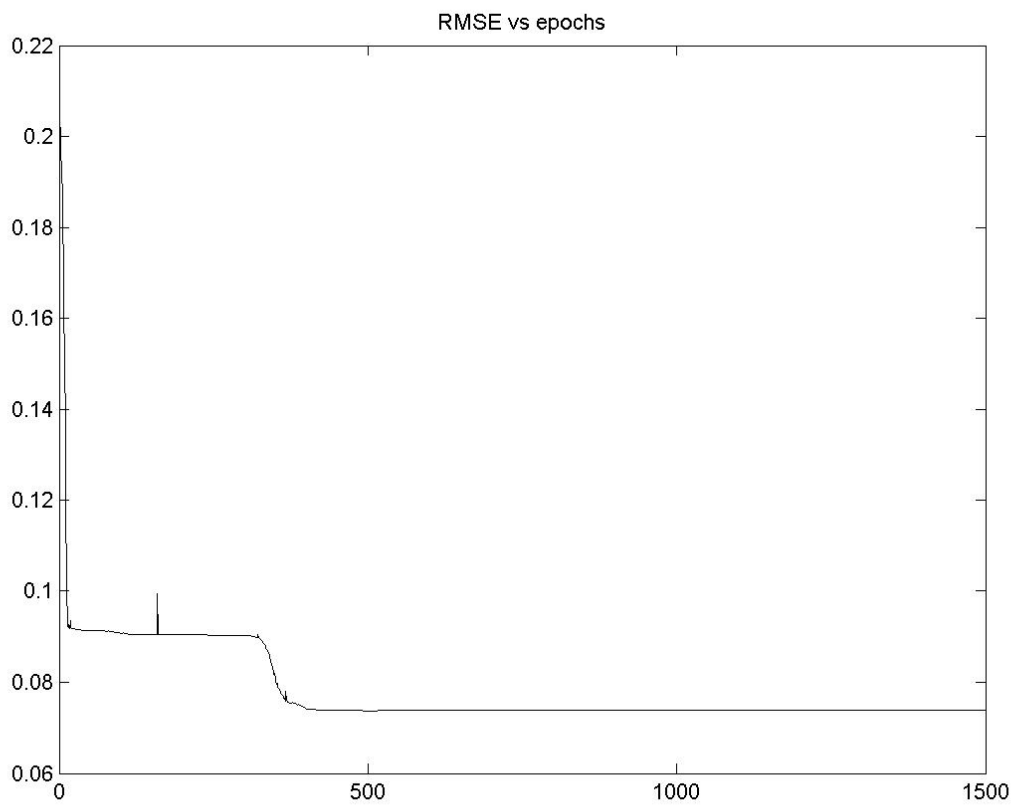
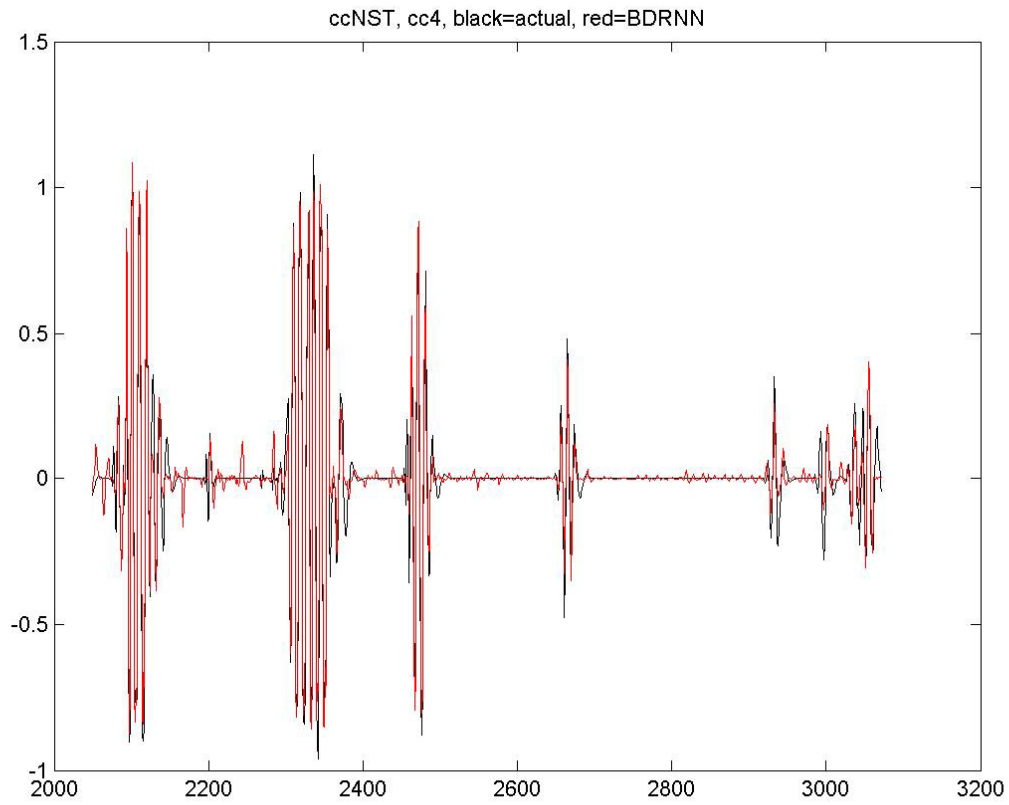
6^η Επανάληψη



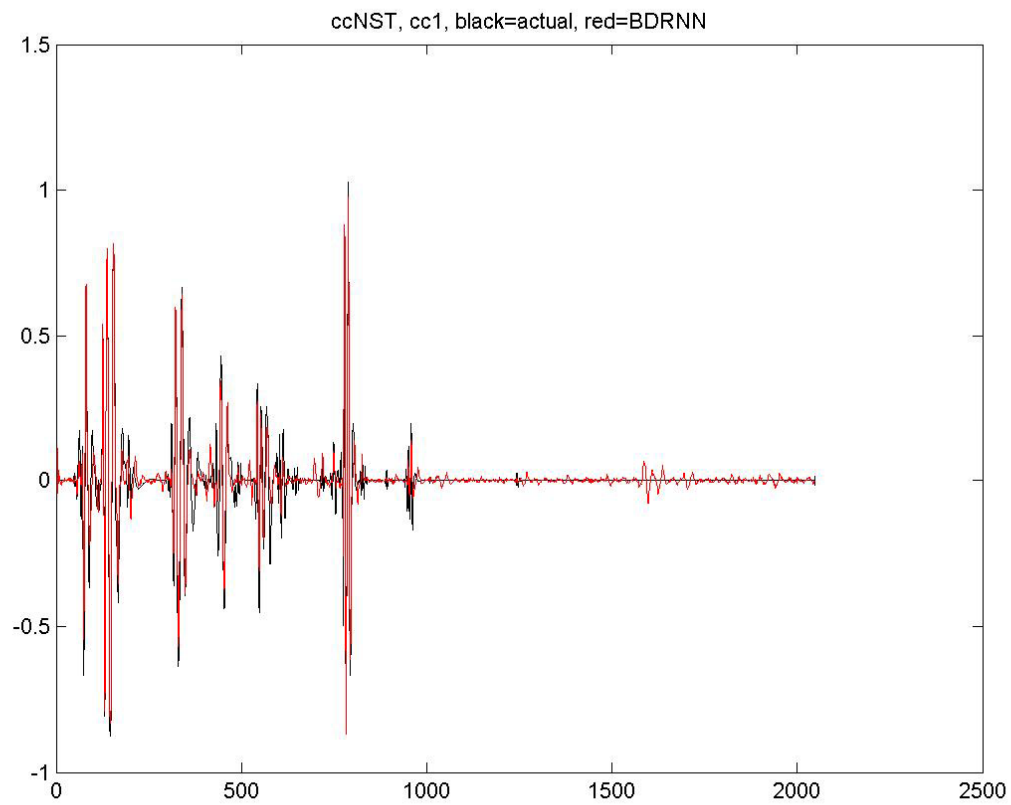
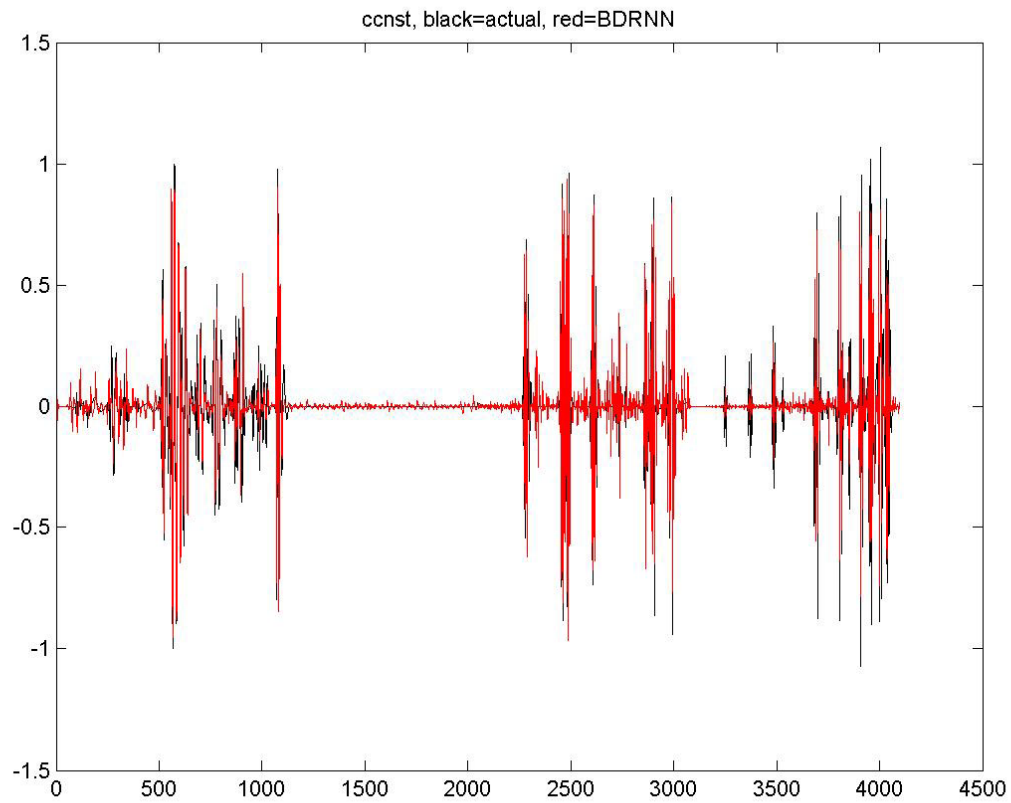


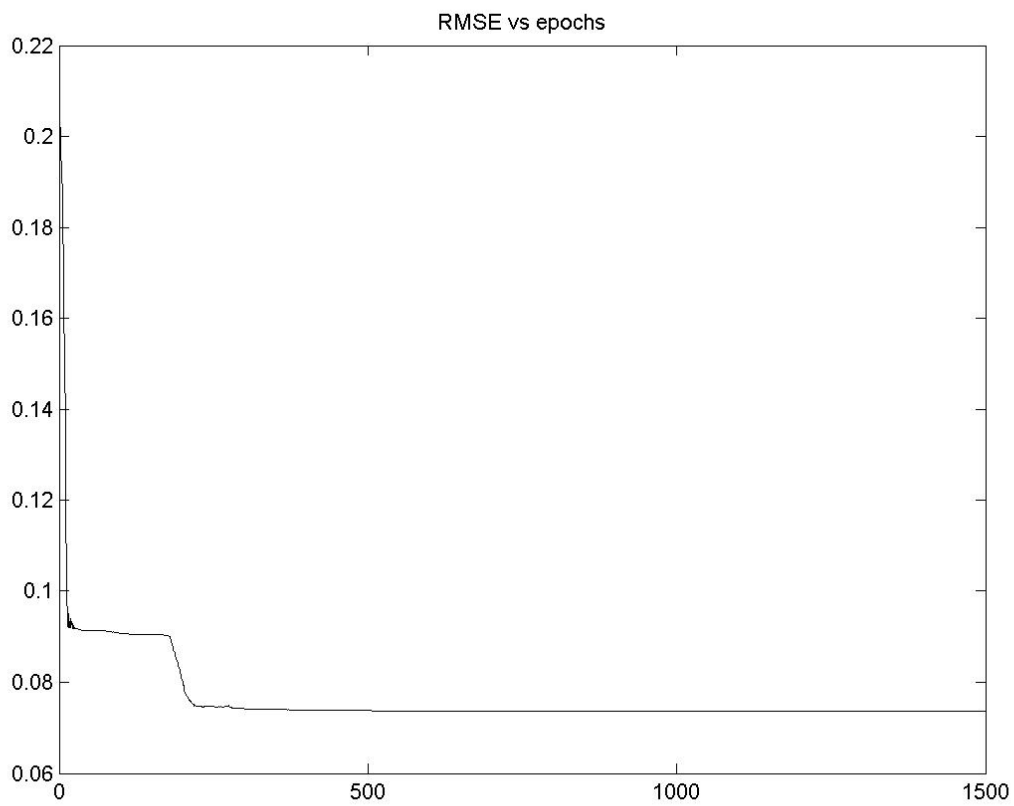
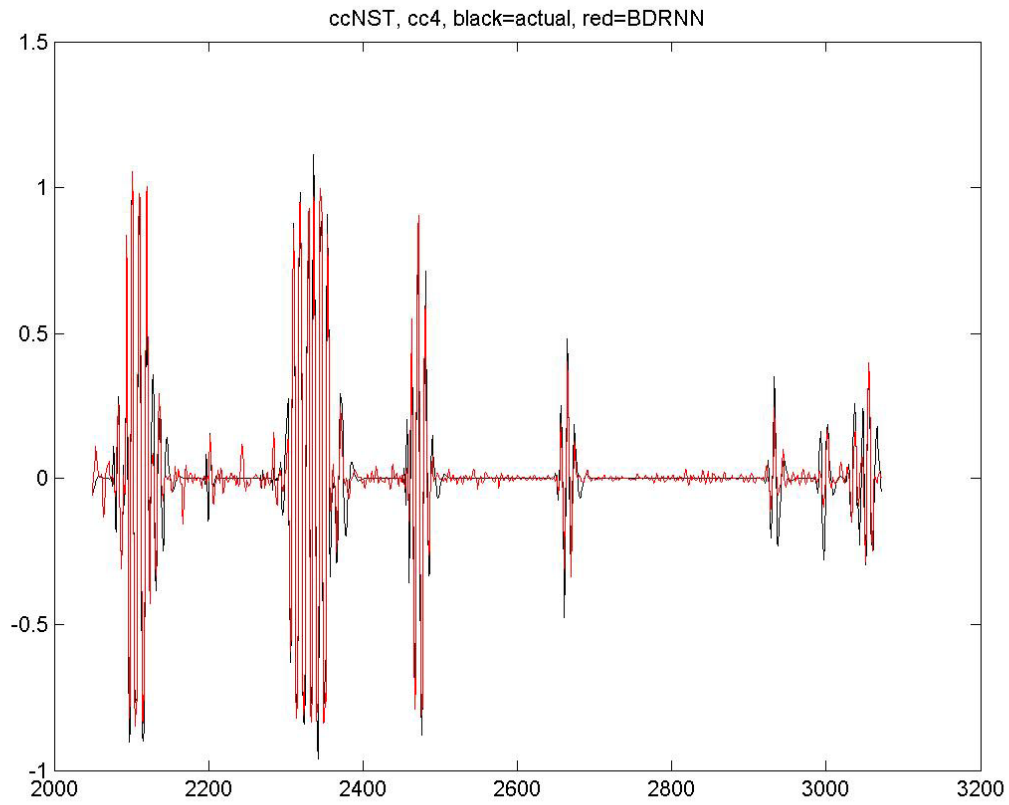
7^η Επανάληψη



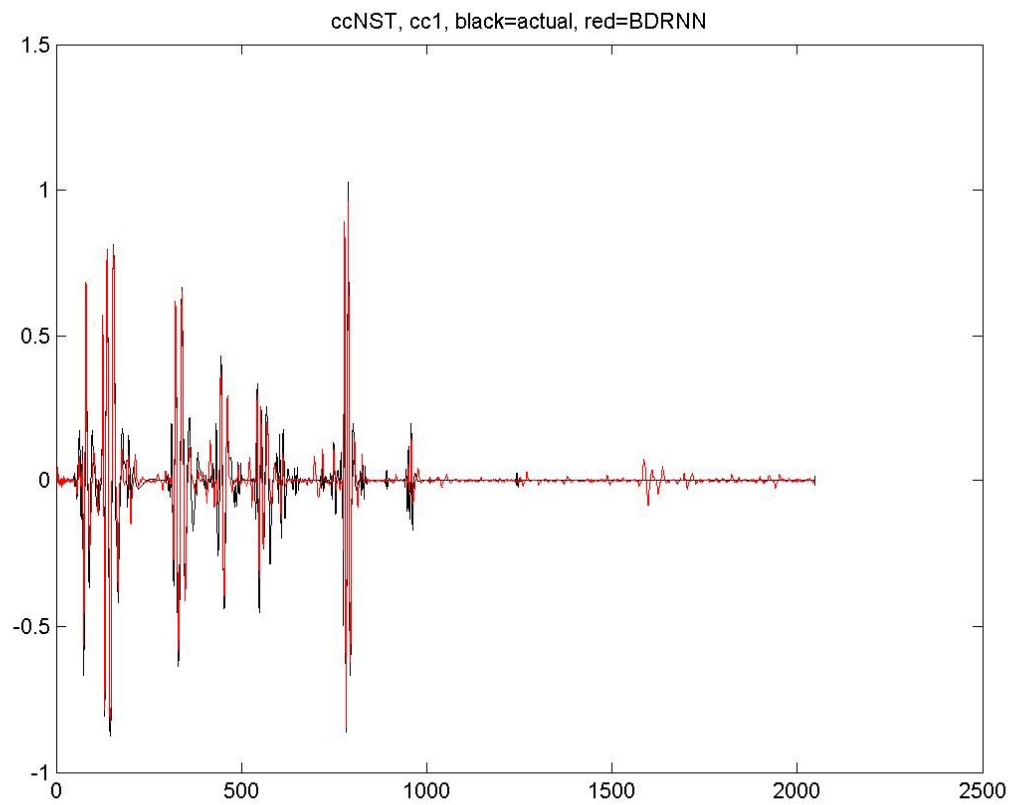
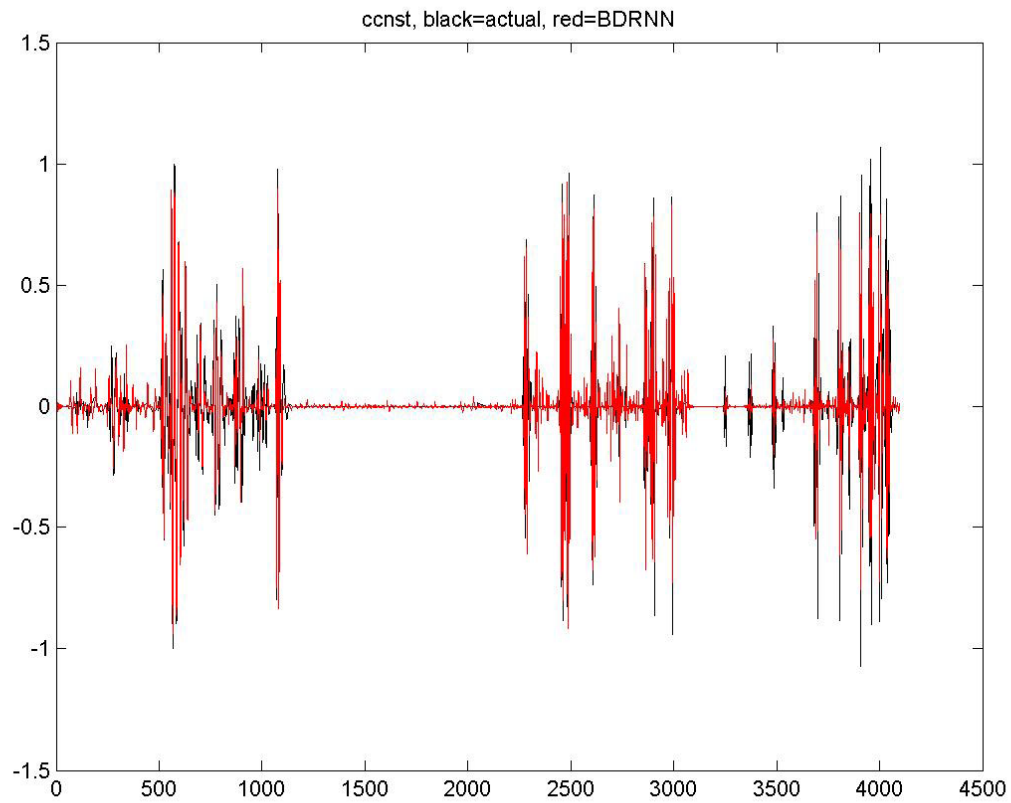


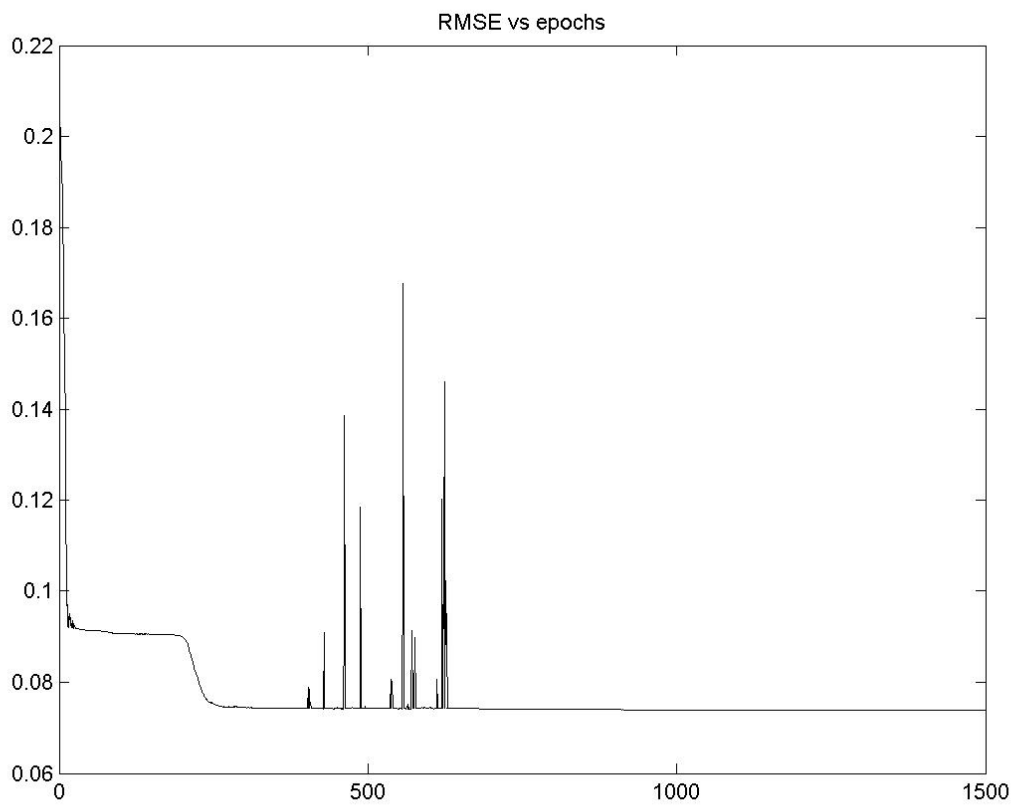
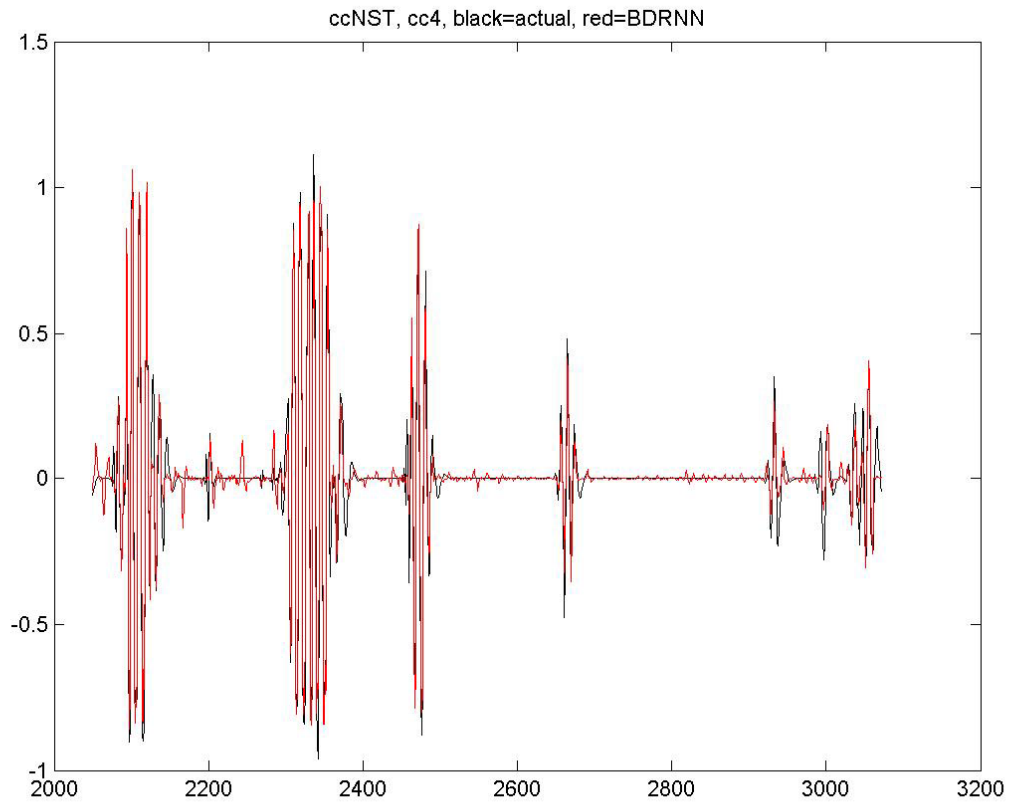
8^η Επανάληψη

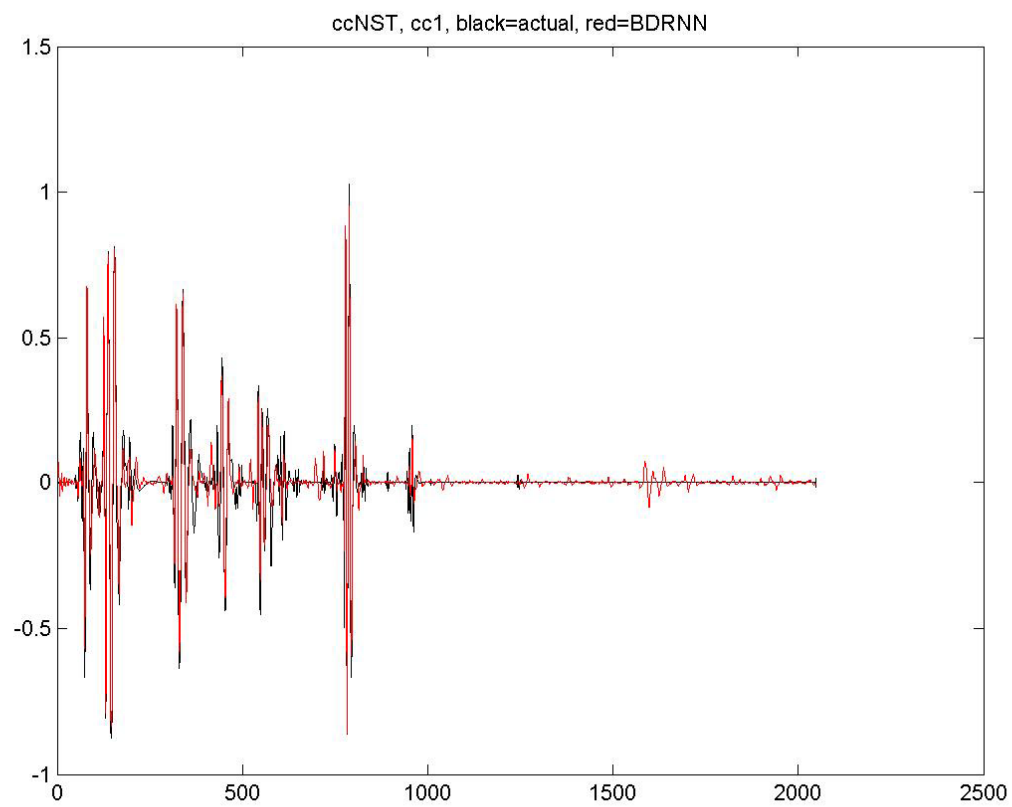
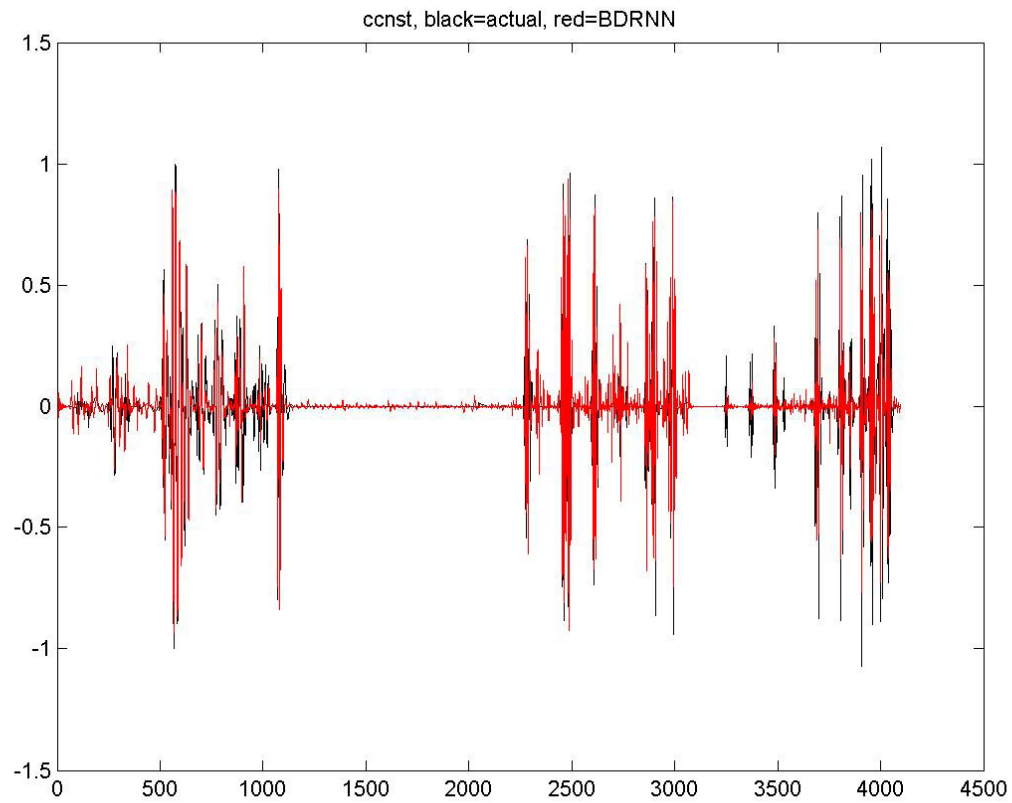


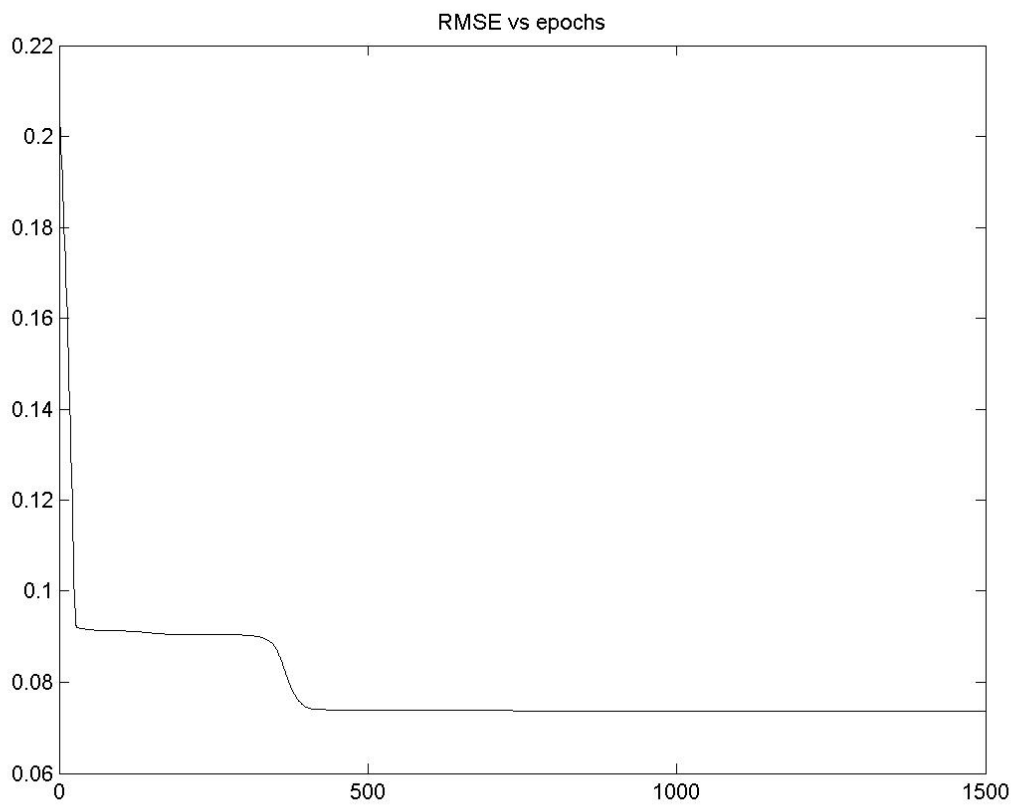
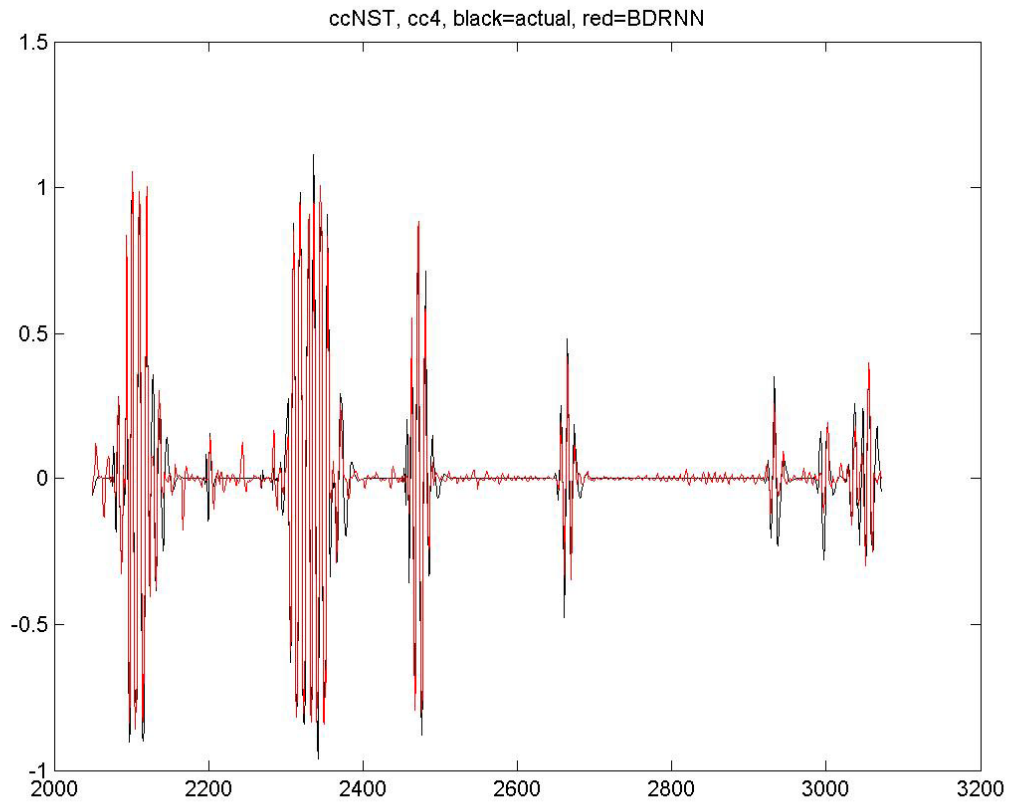


9^η Επανάληψη

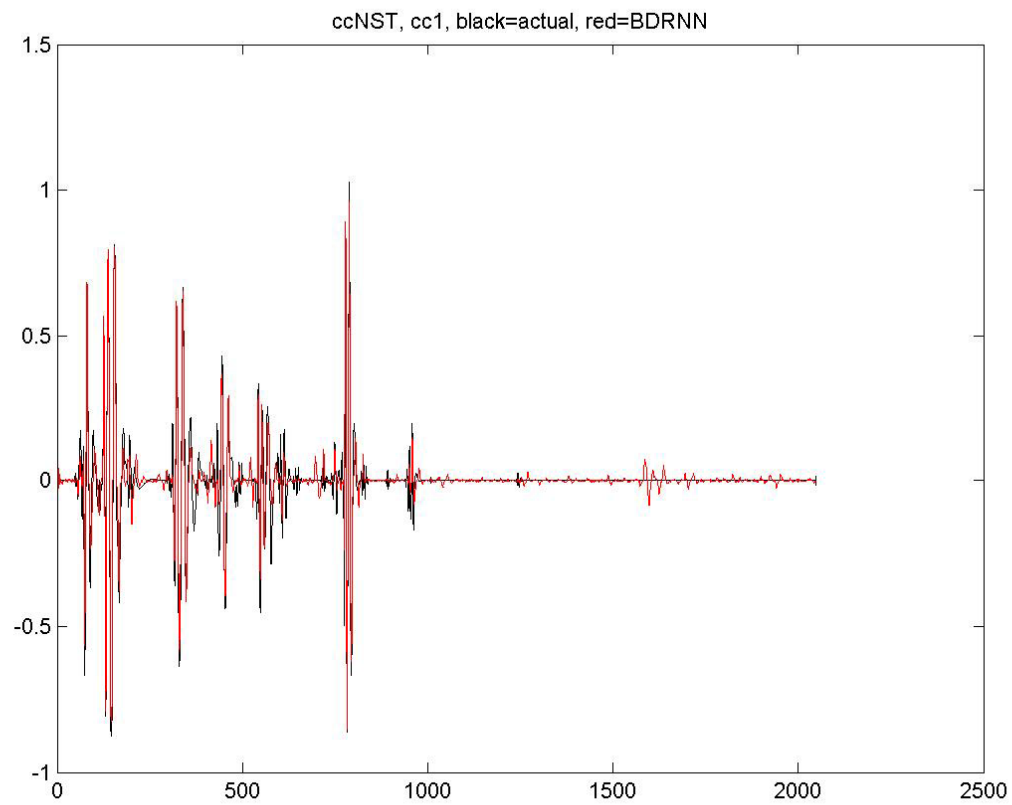
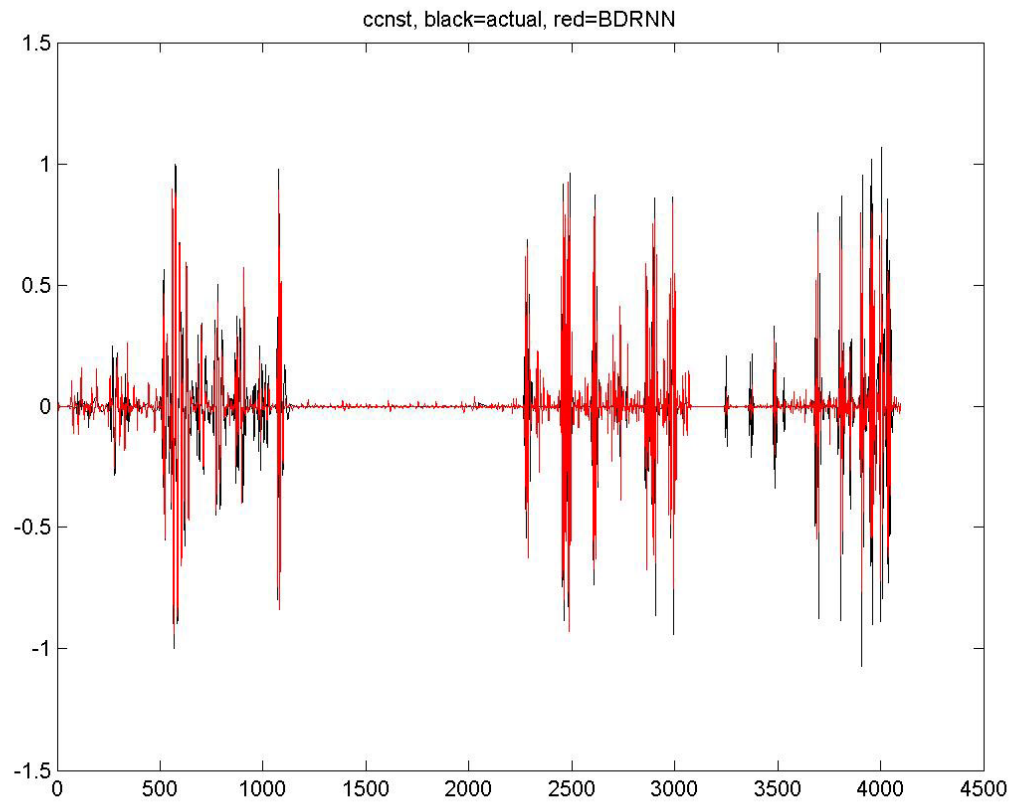


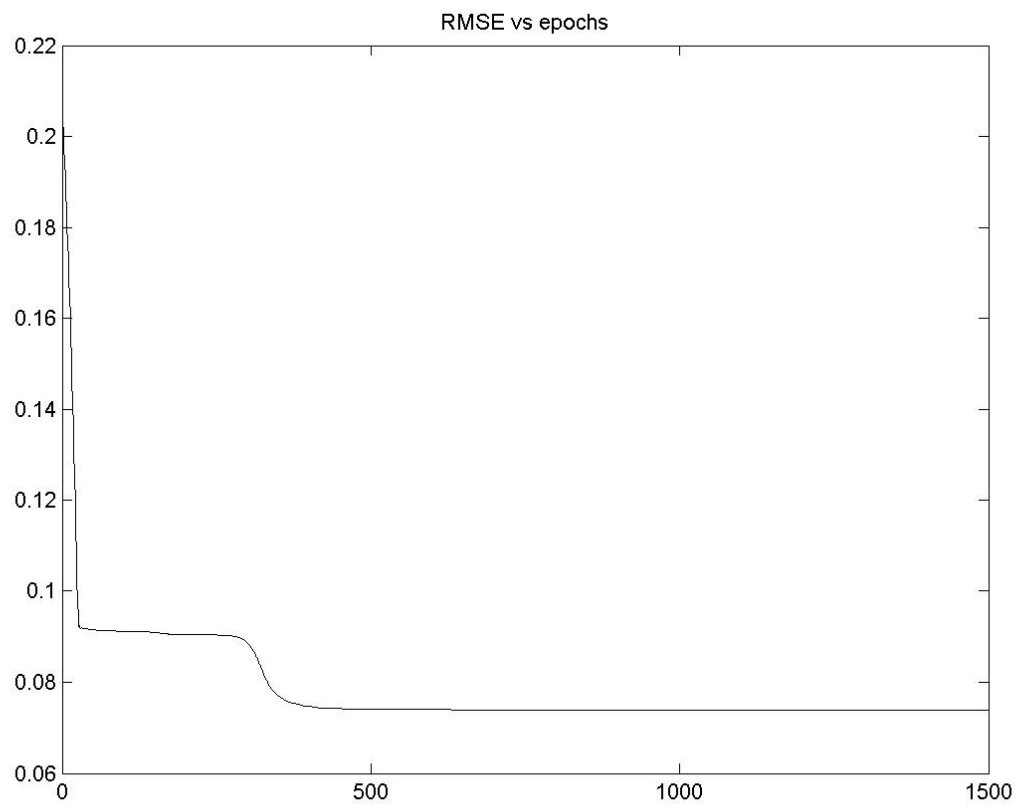
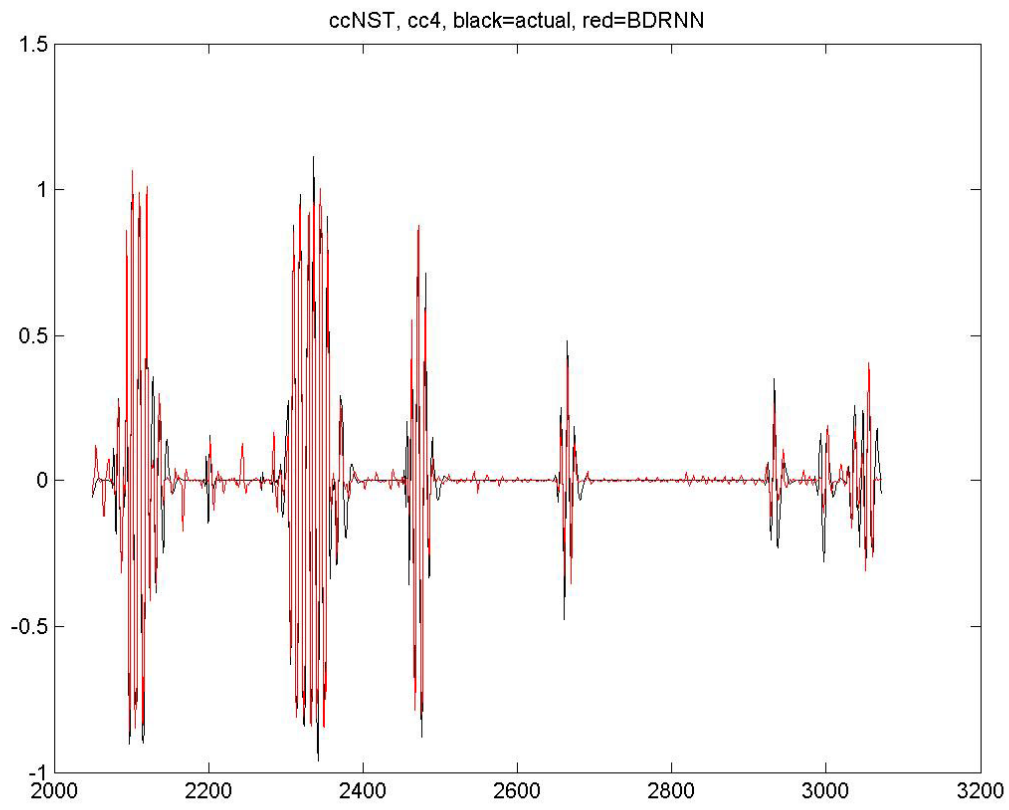


10¹ Επανάληψη

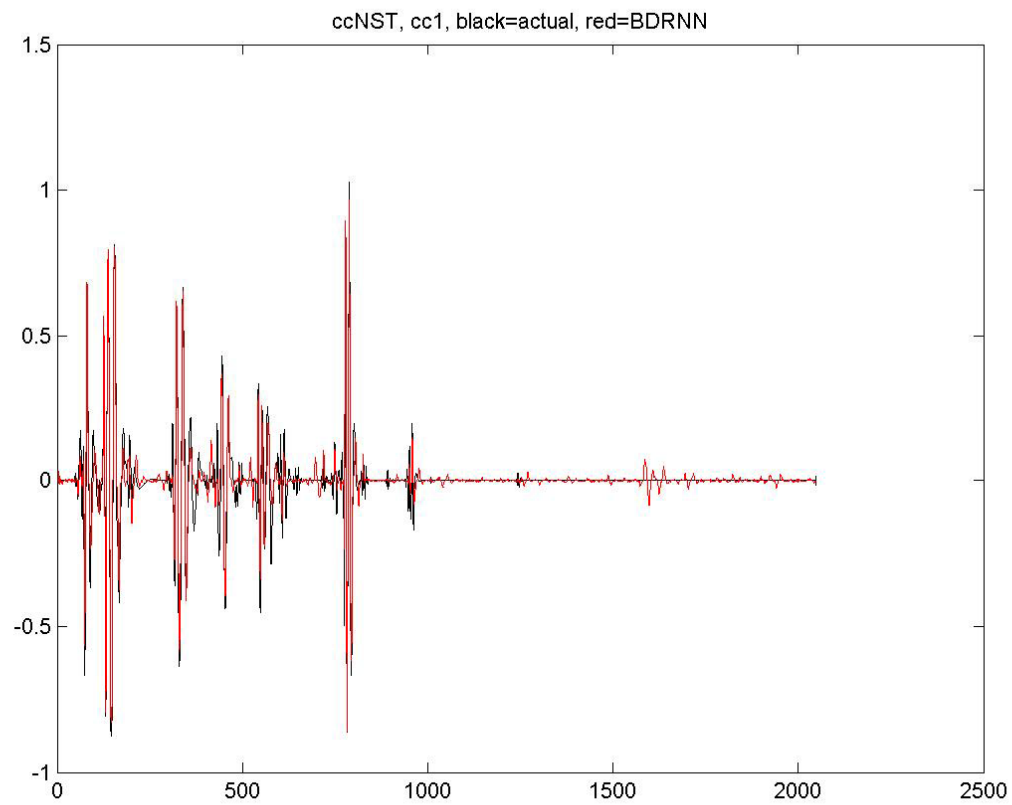
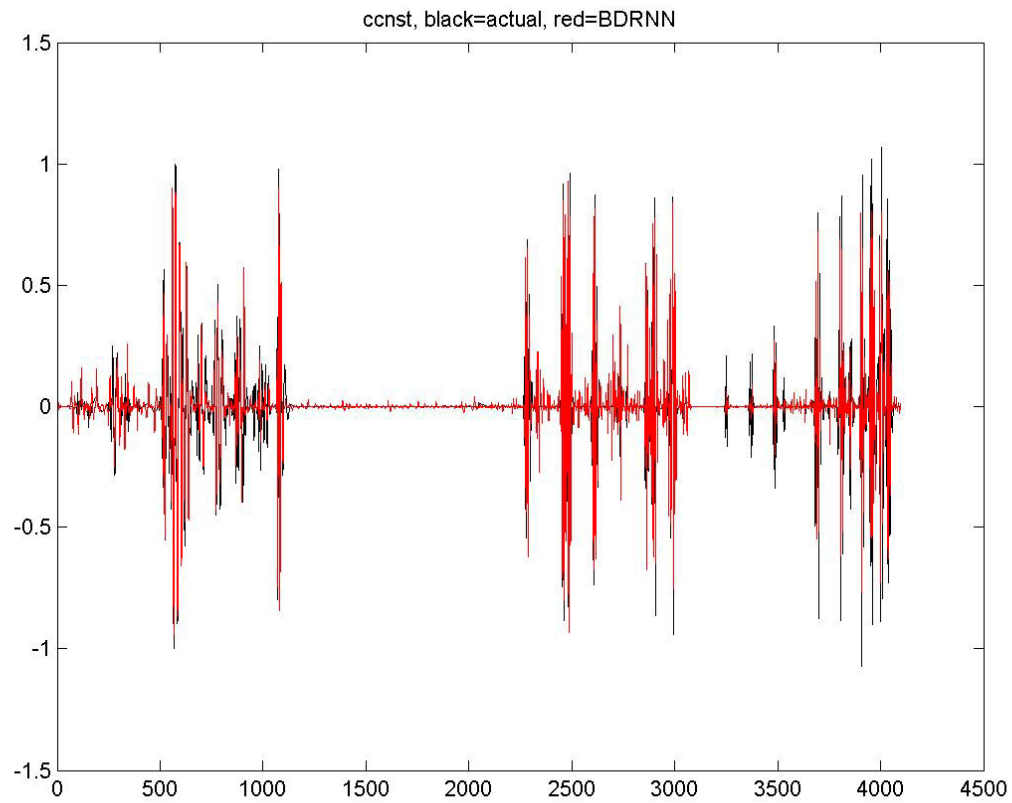


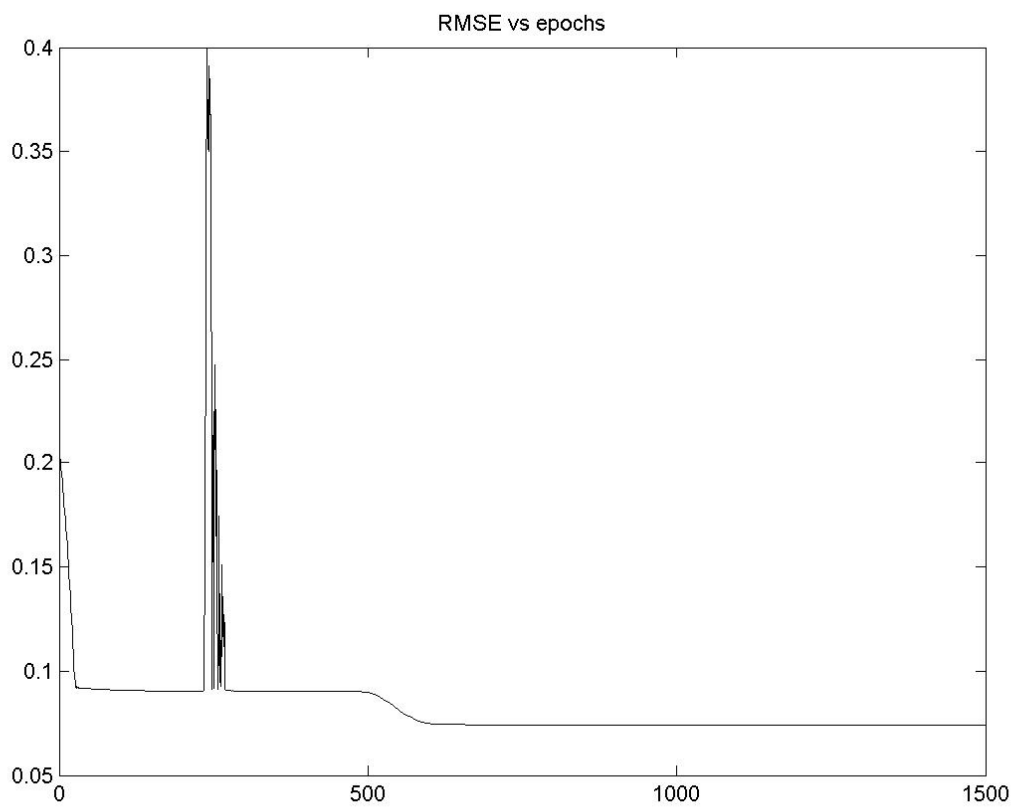
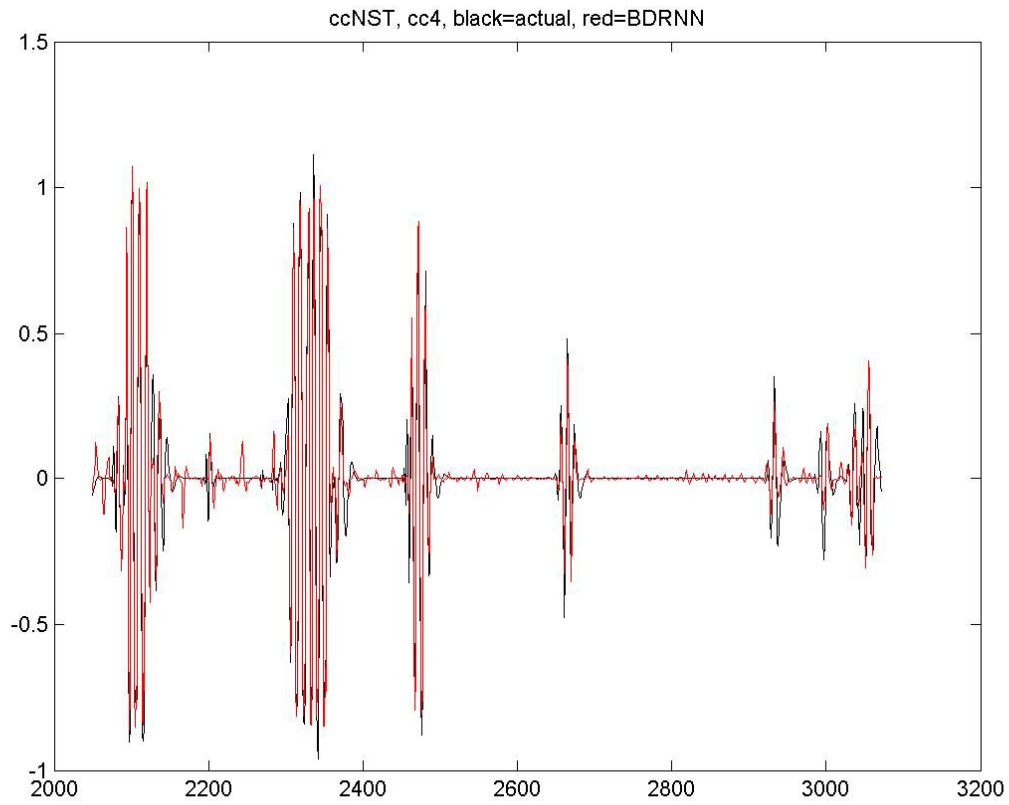
11^η Επανάληψη

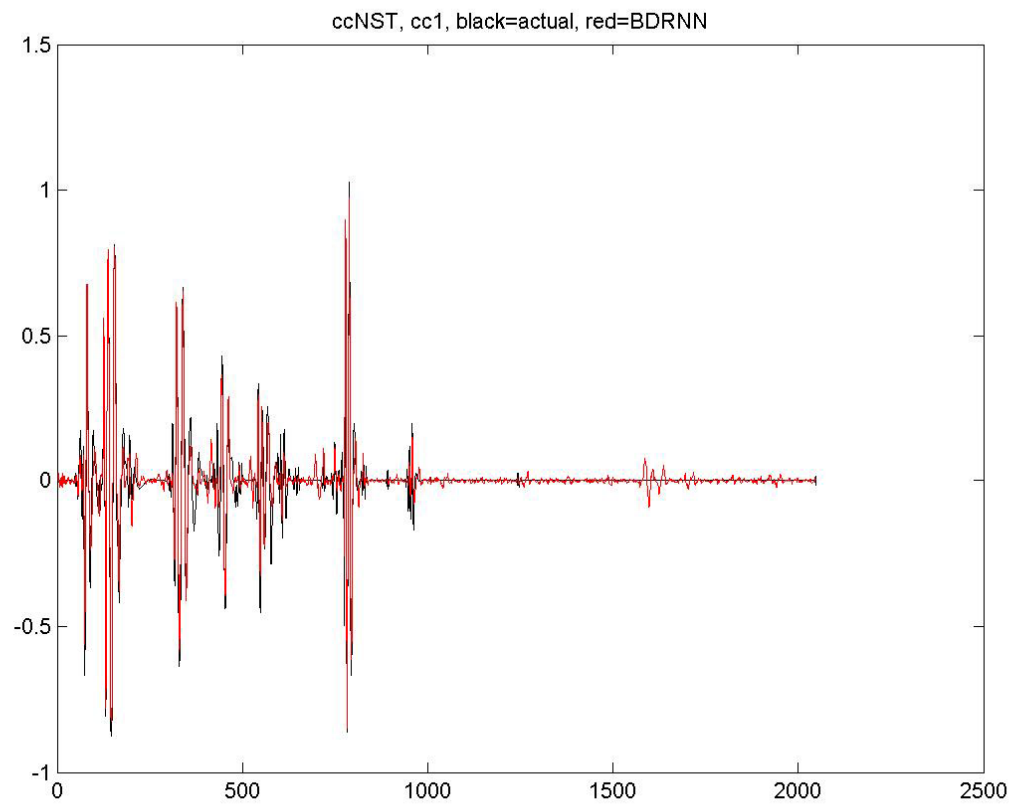
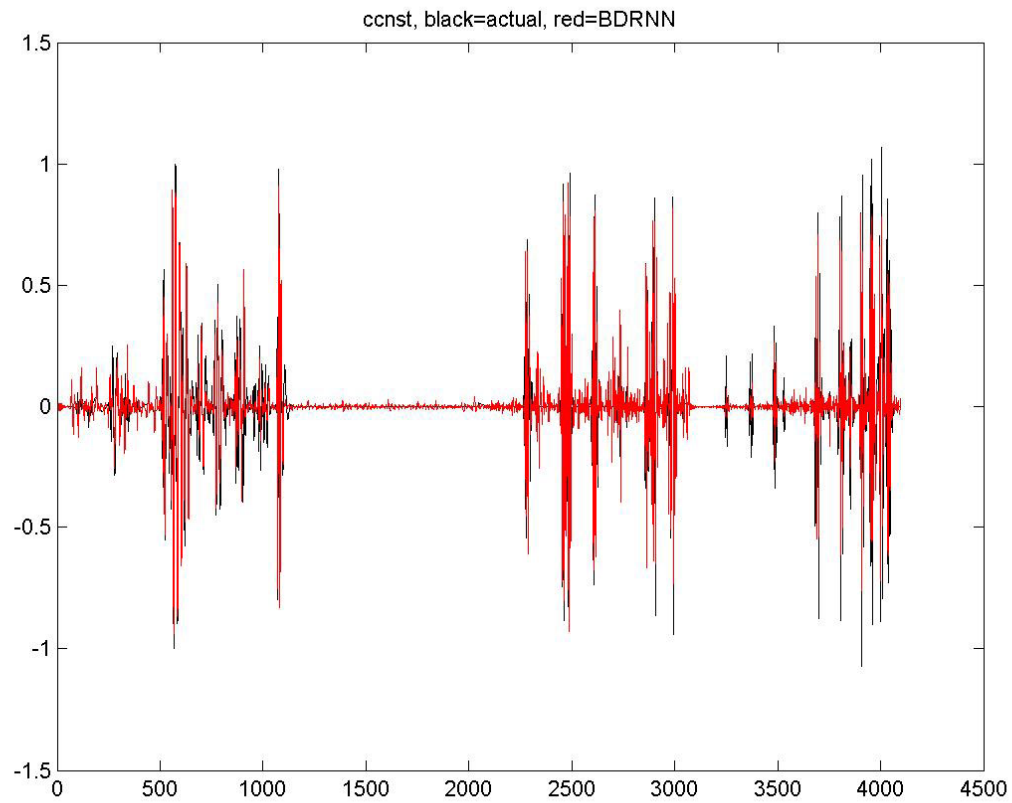


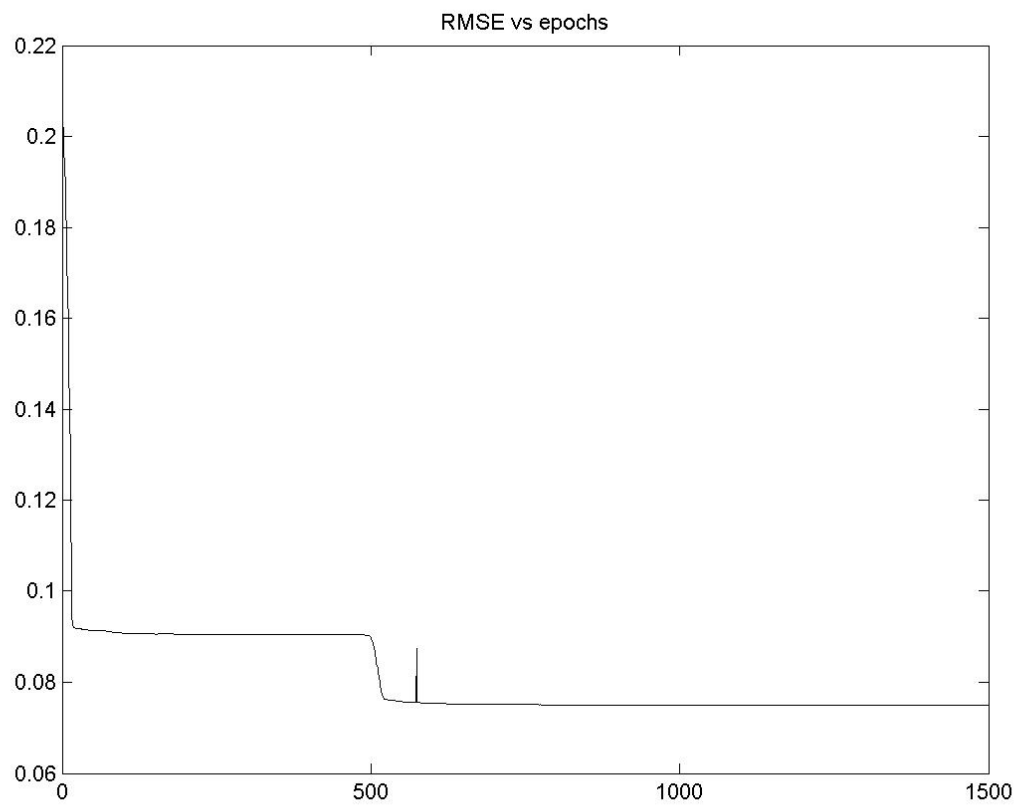
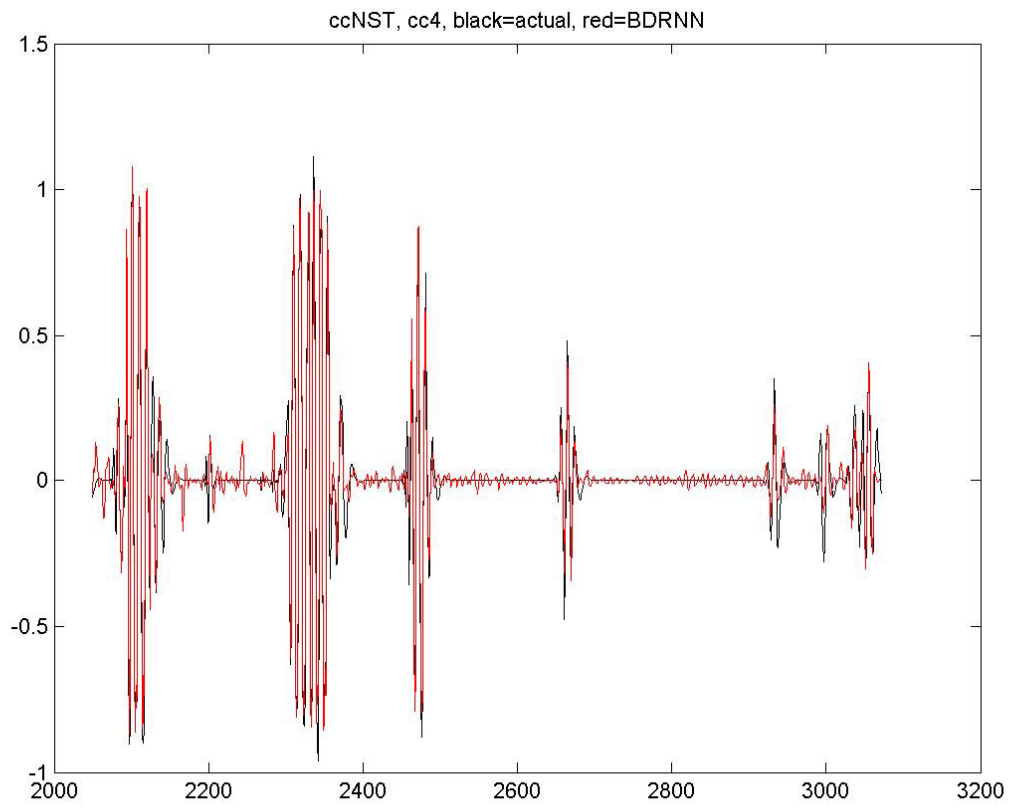


12^η Επανάληψη

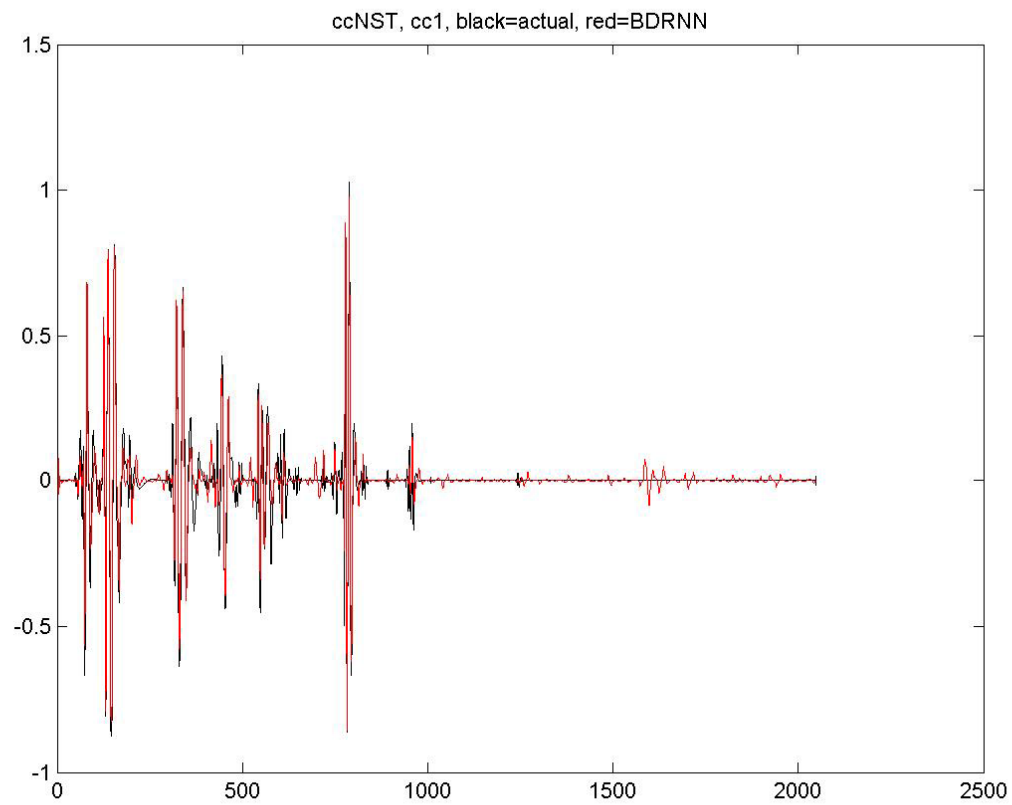
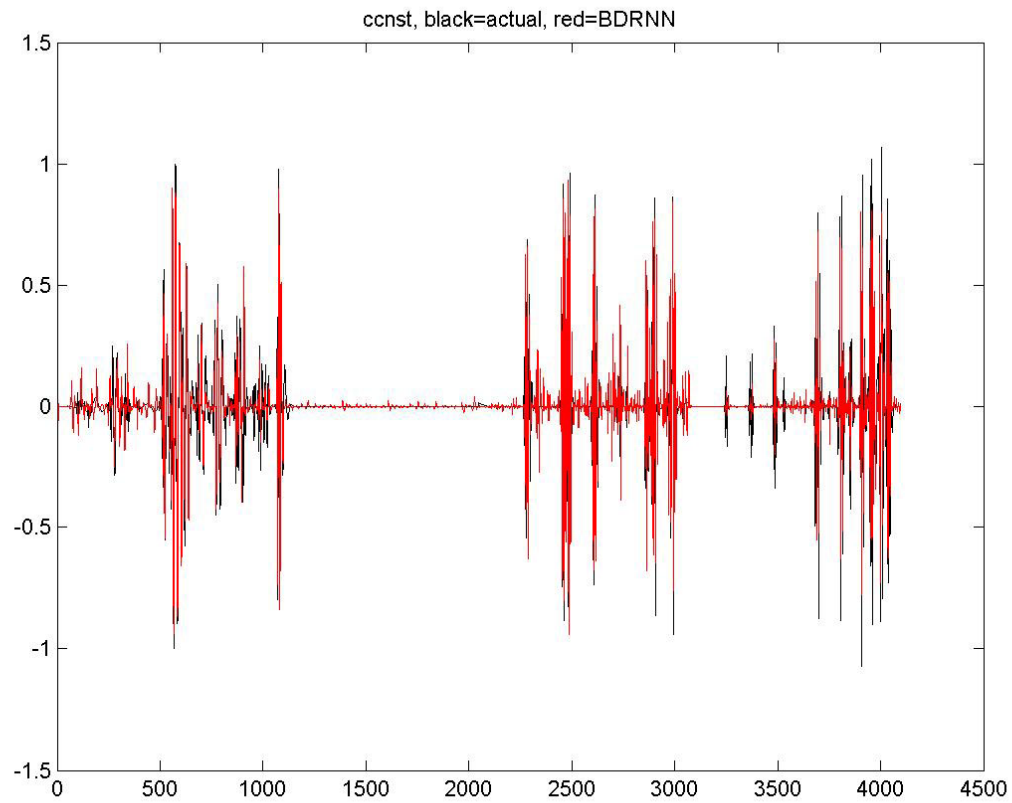


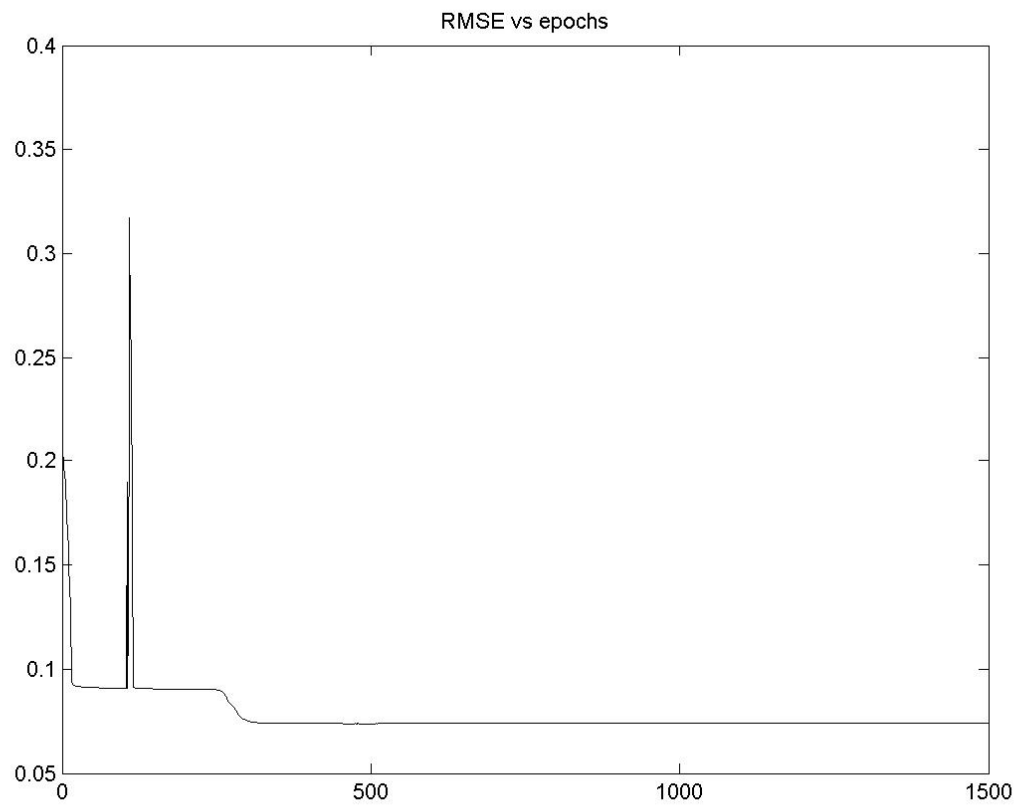
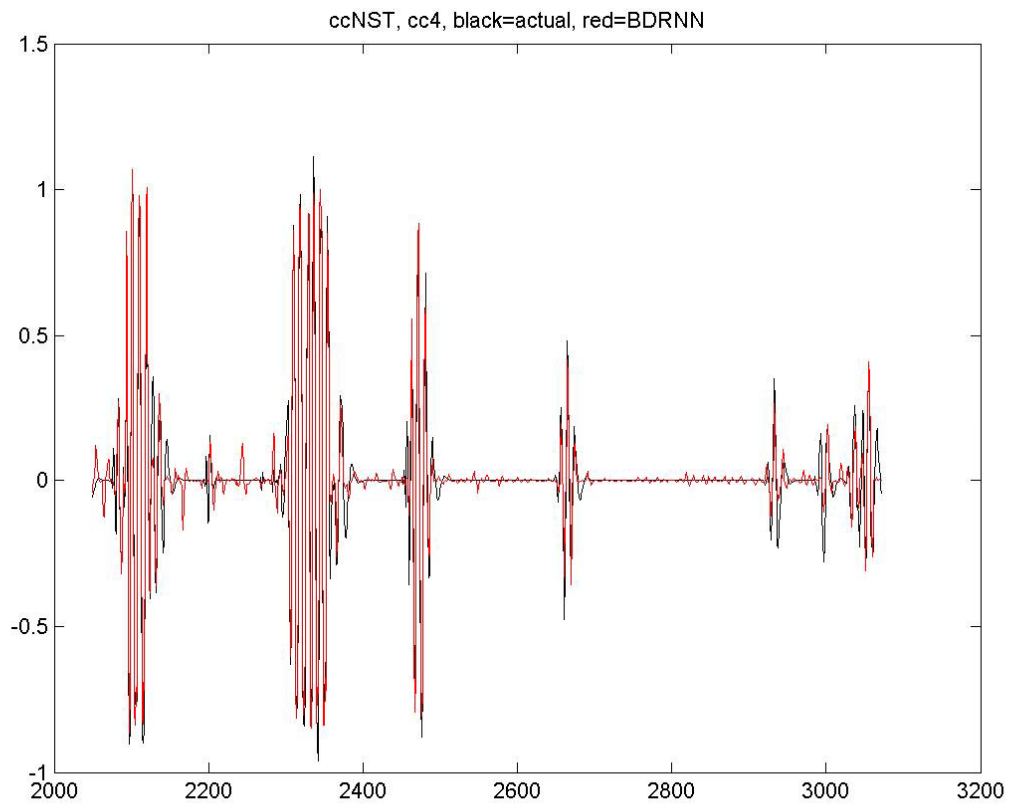


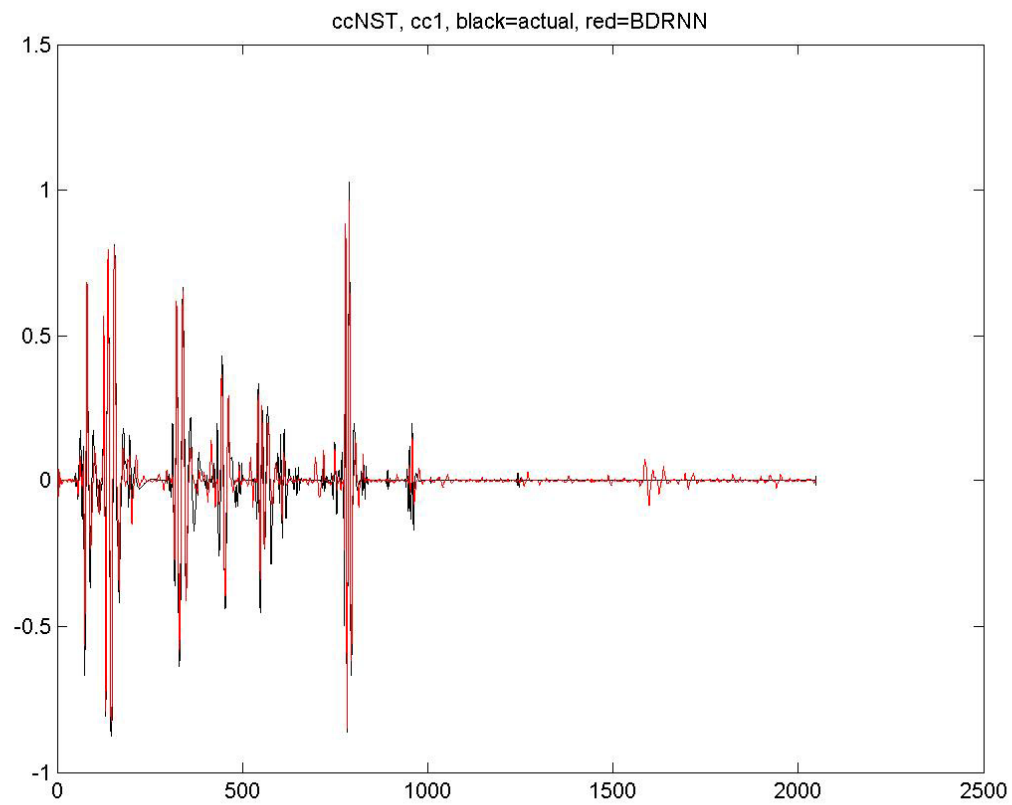
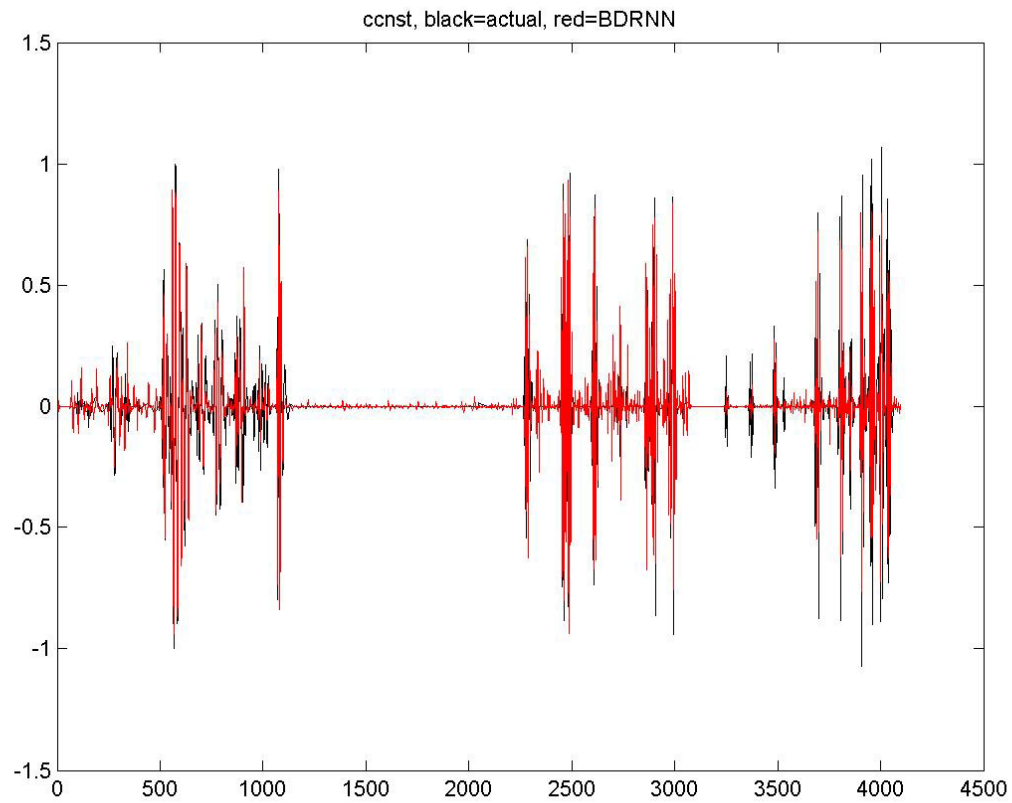
13^η Επανάληψη

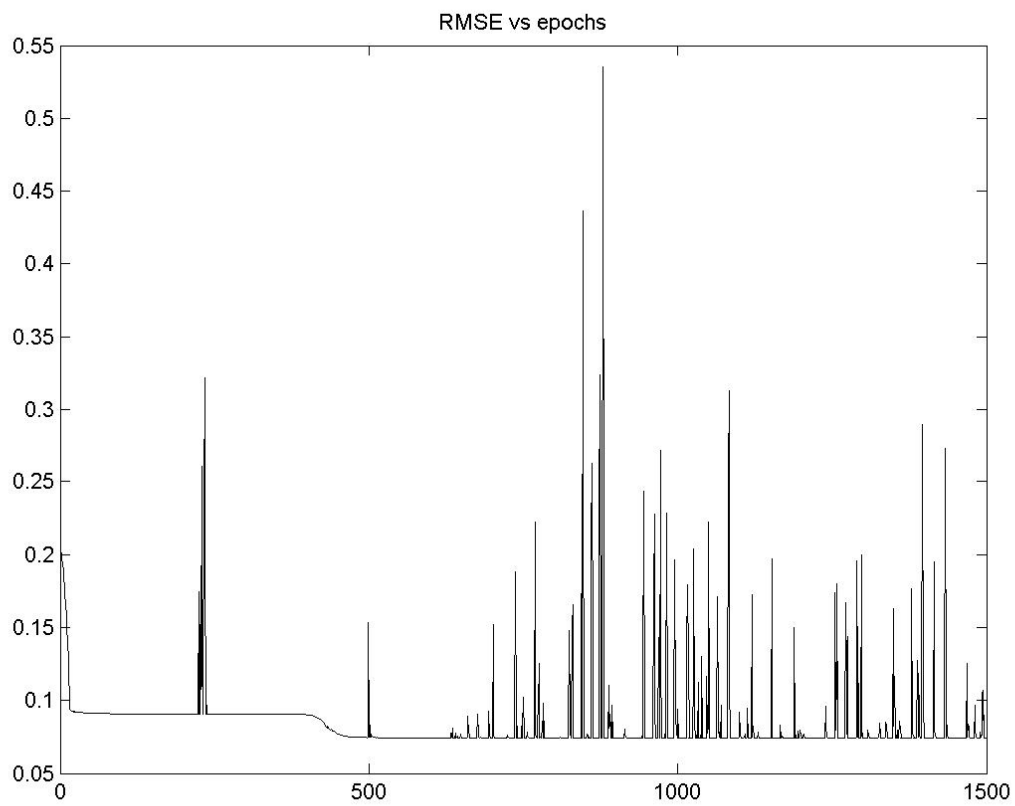
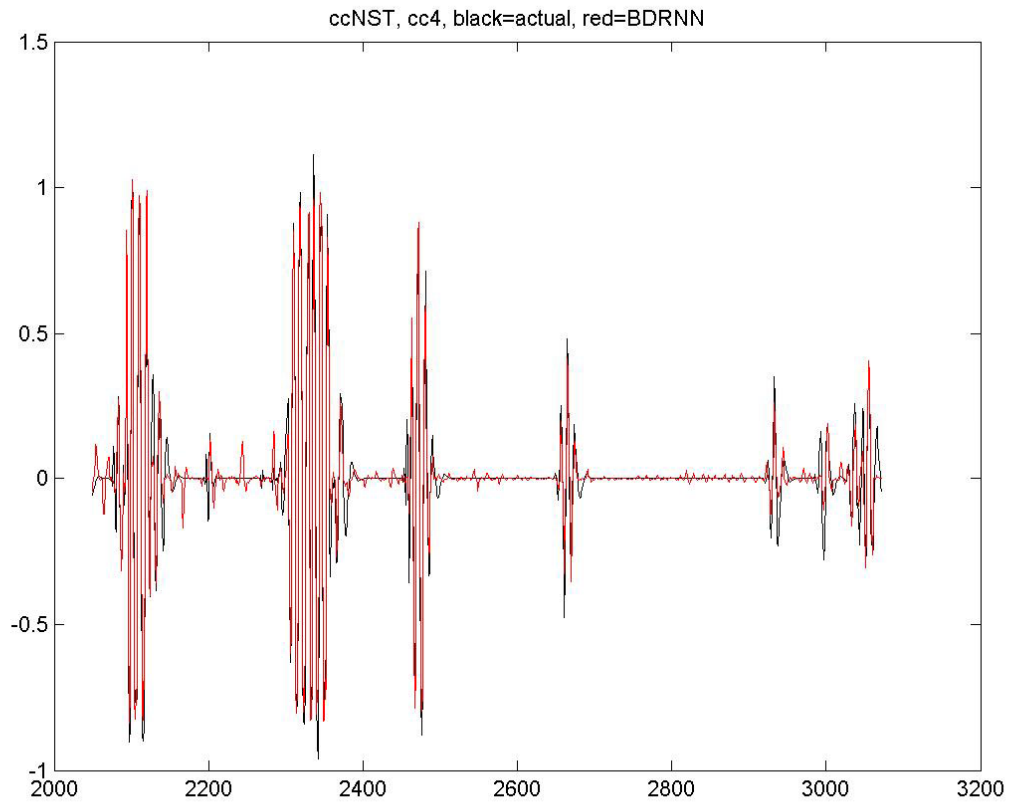


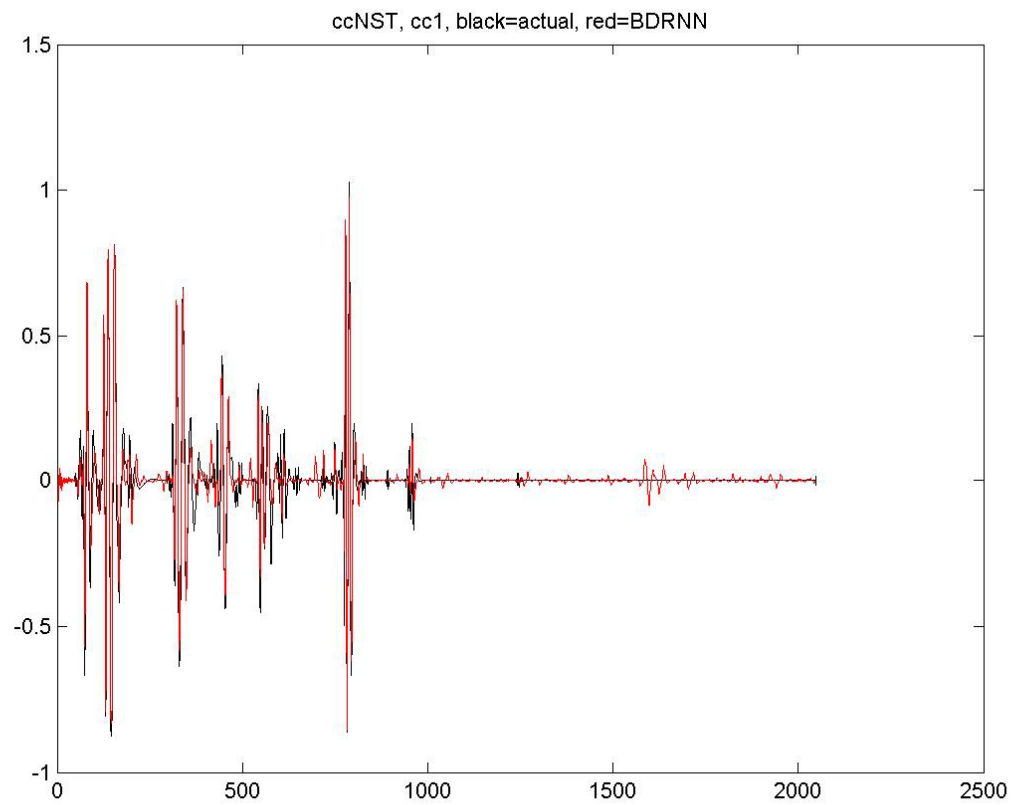
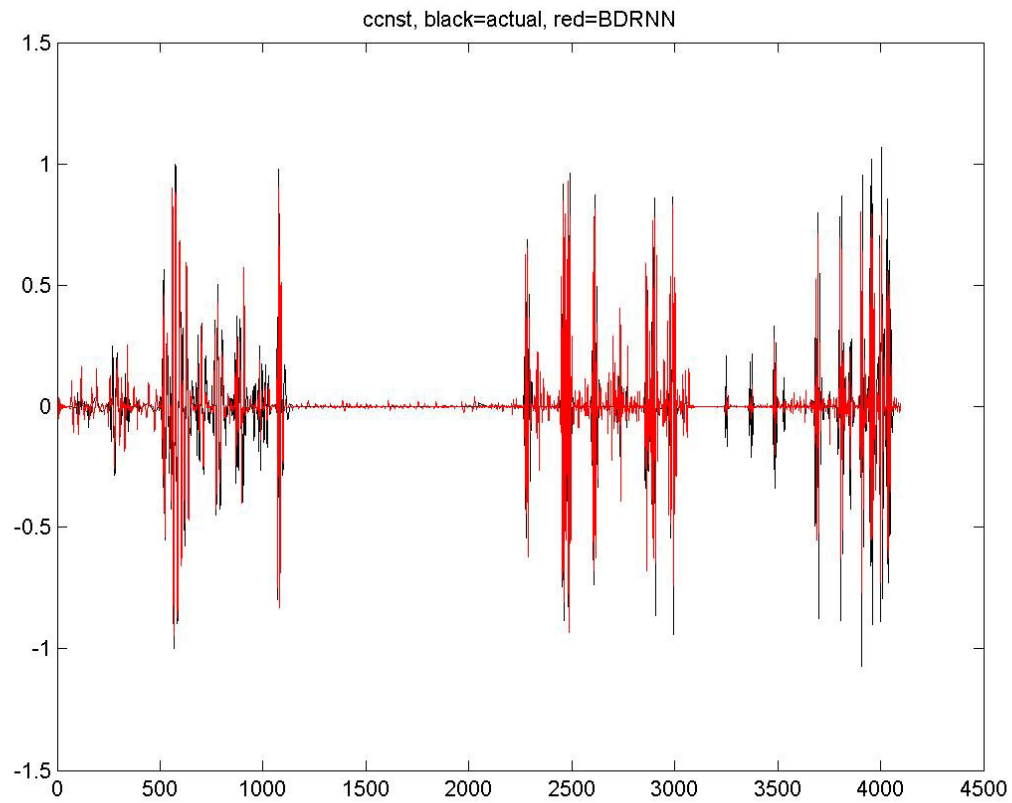
14^η Επανάληψη

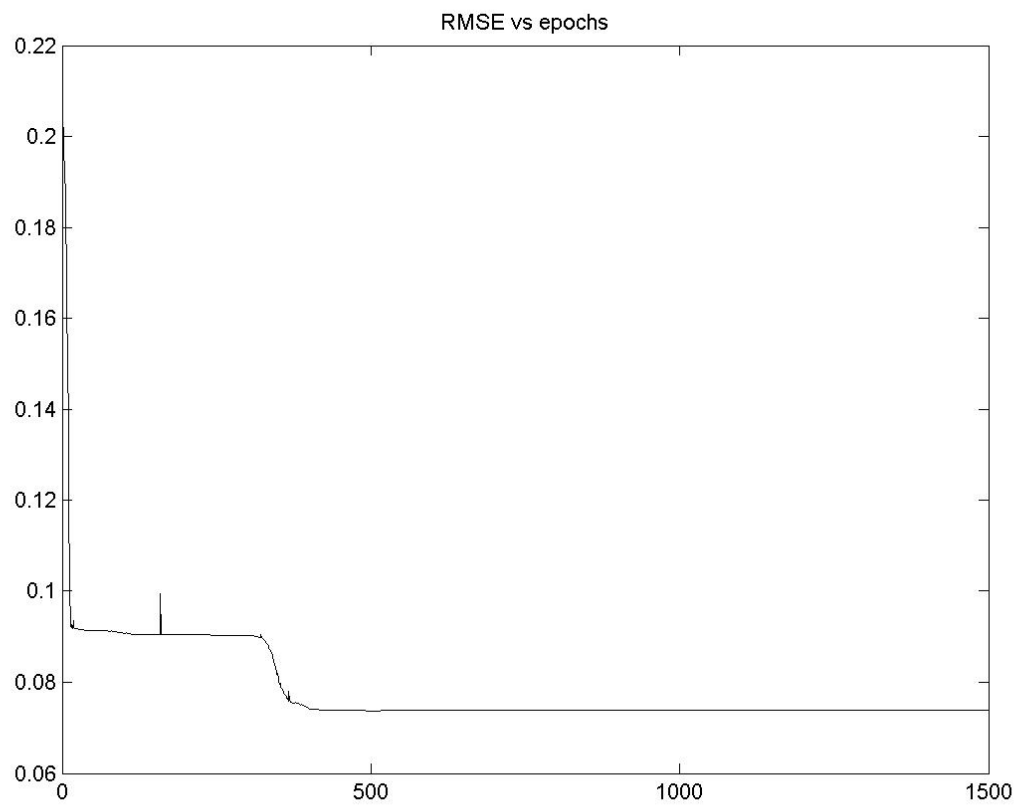
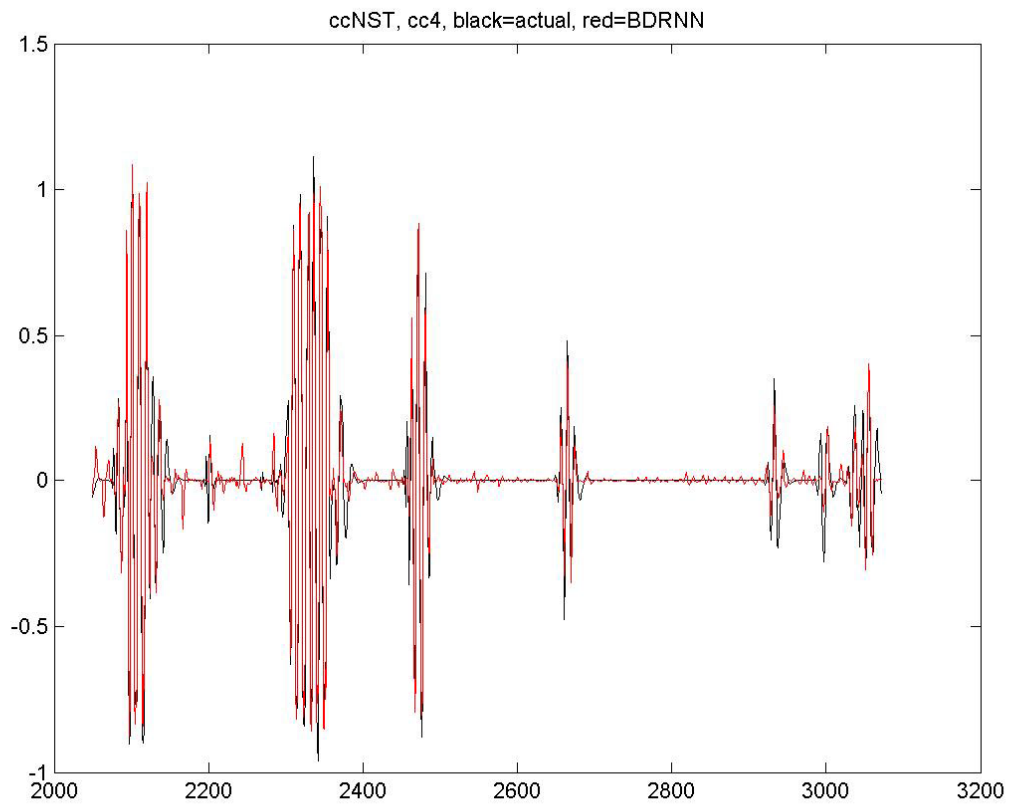




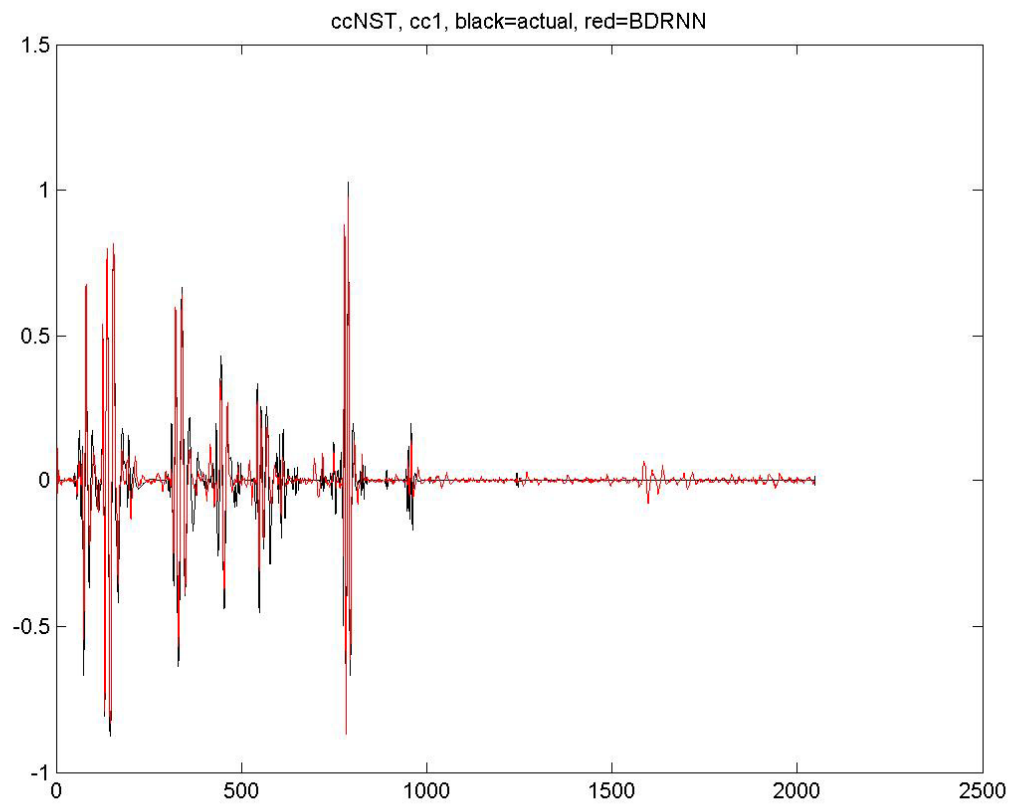
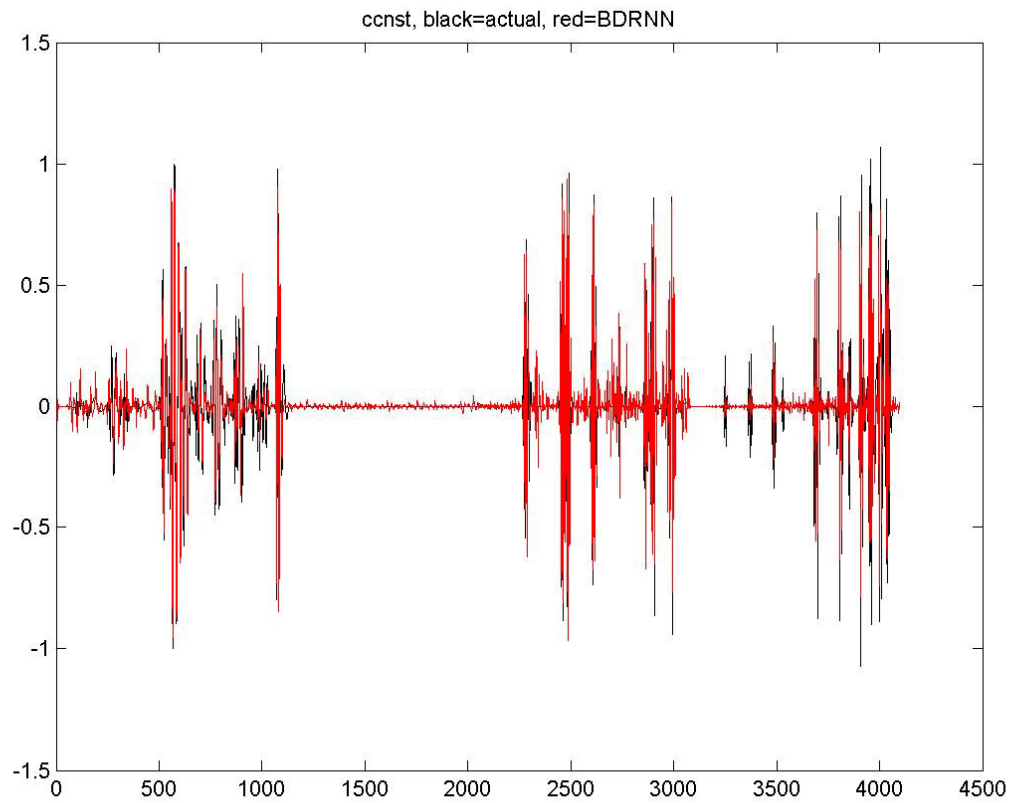
15^η Επανάληψη

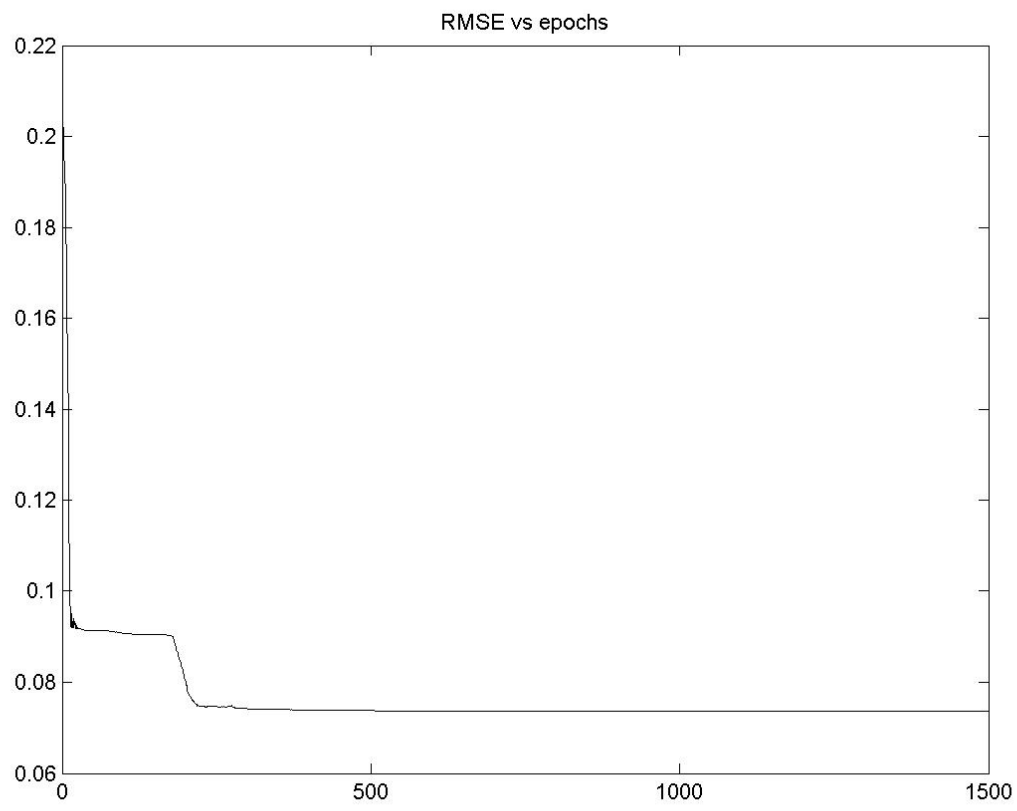
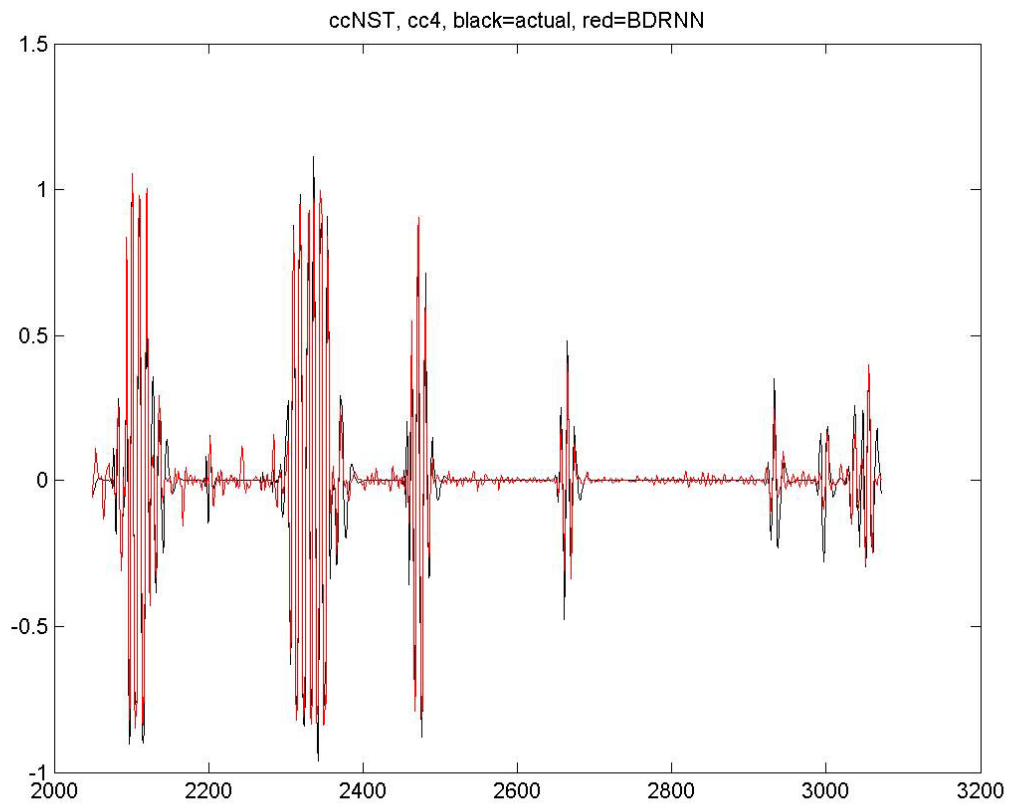


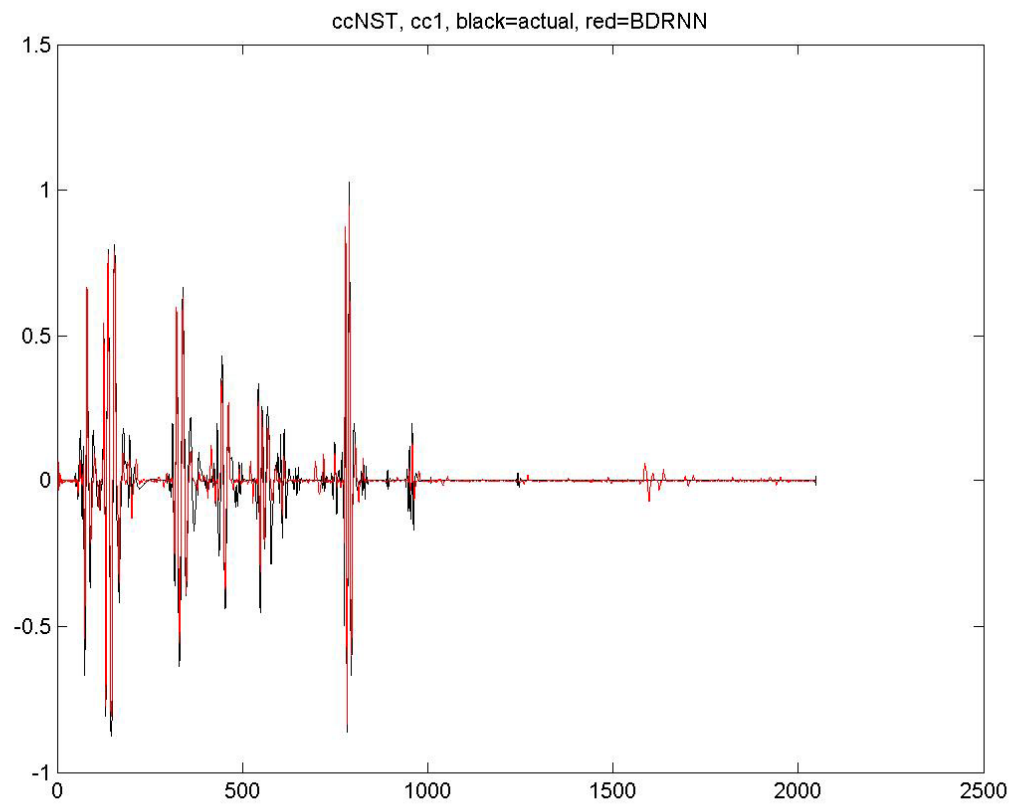
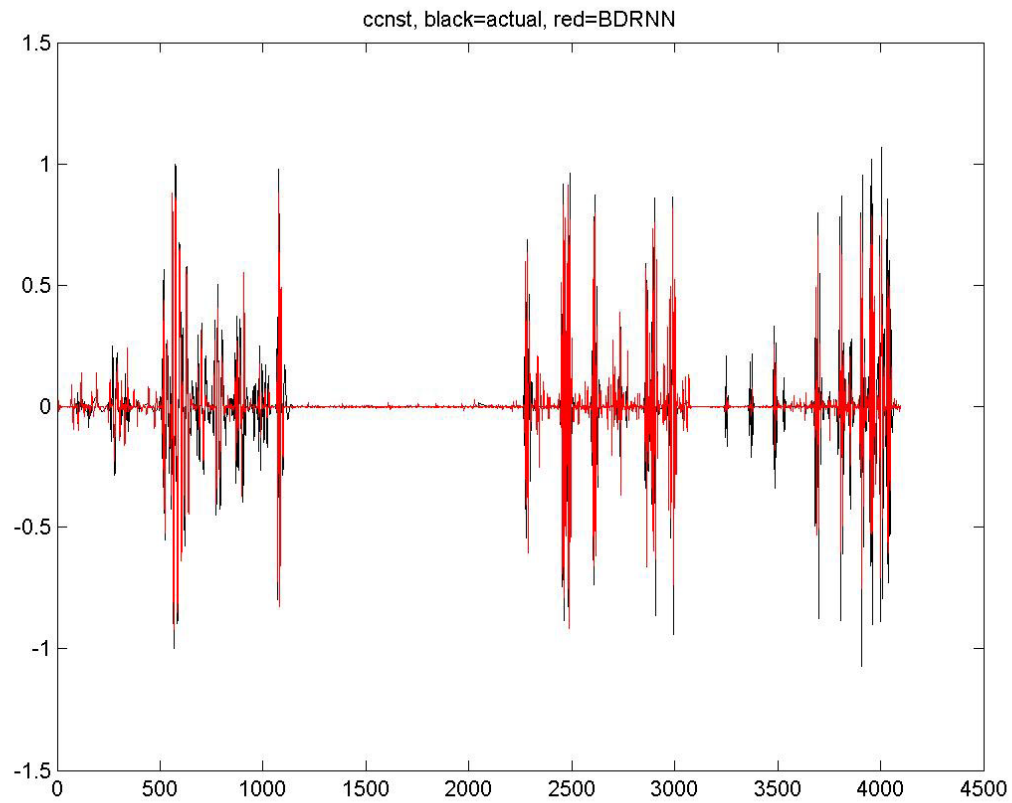
16^η Επανάληψη

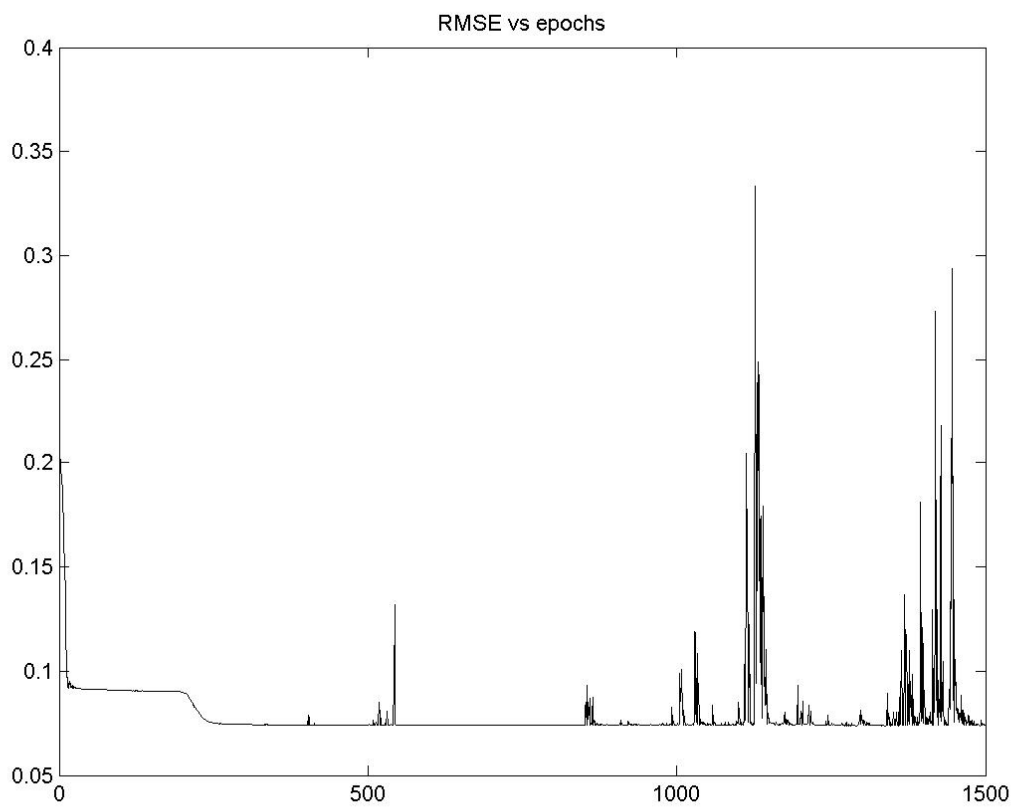
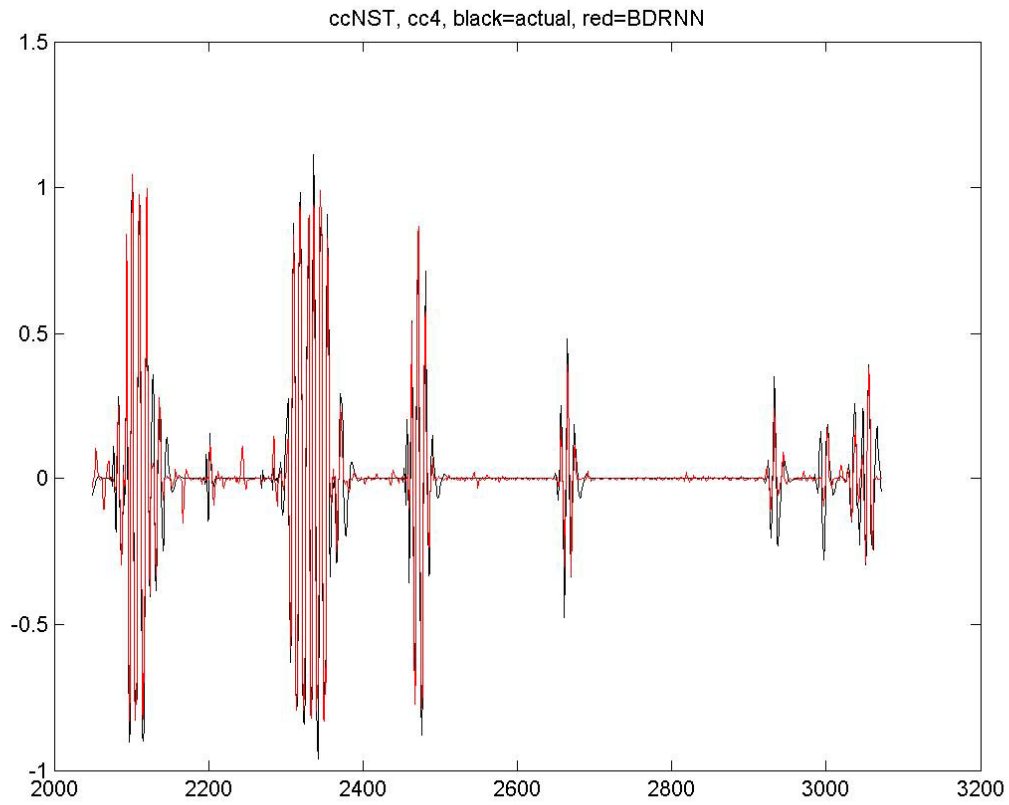


17^η Επανάληψη

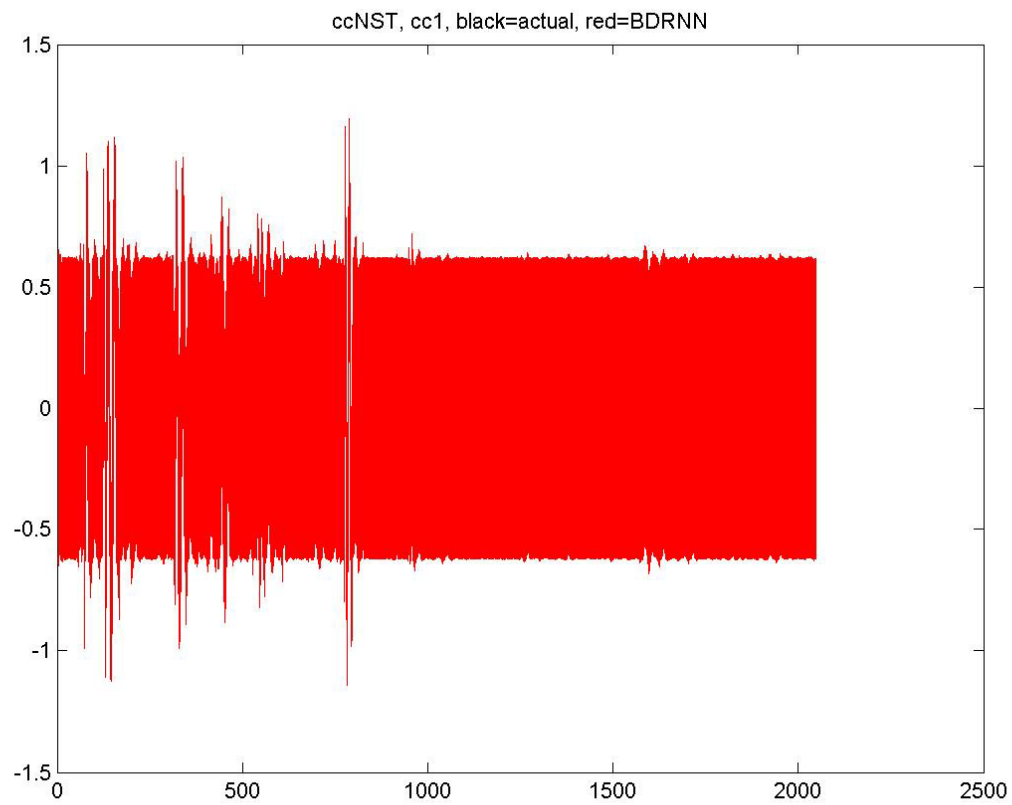
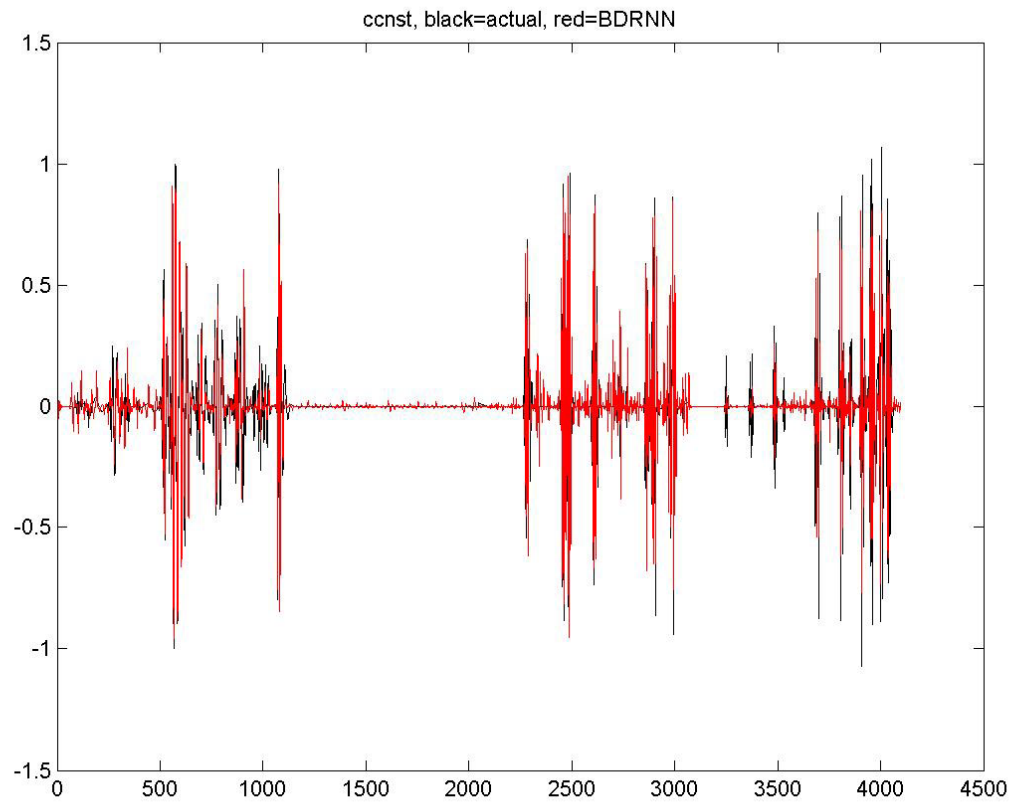


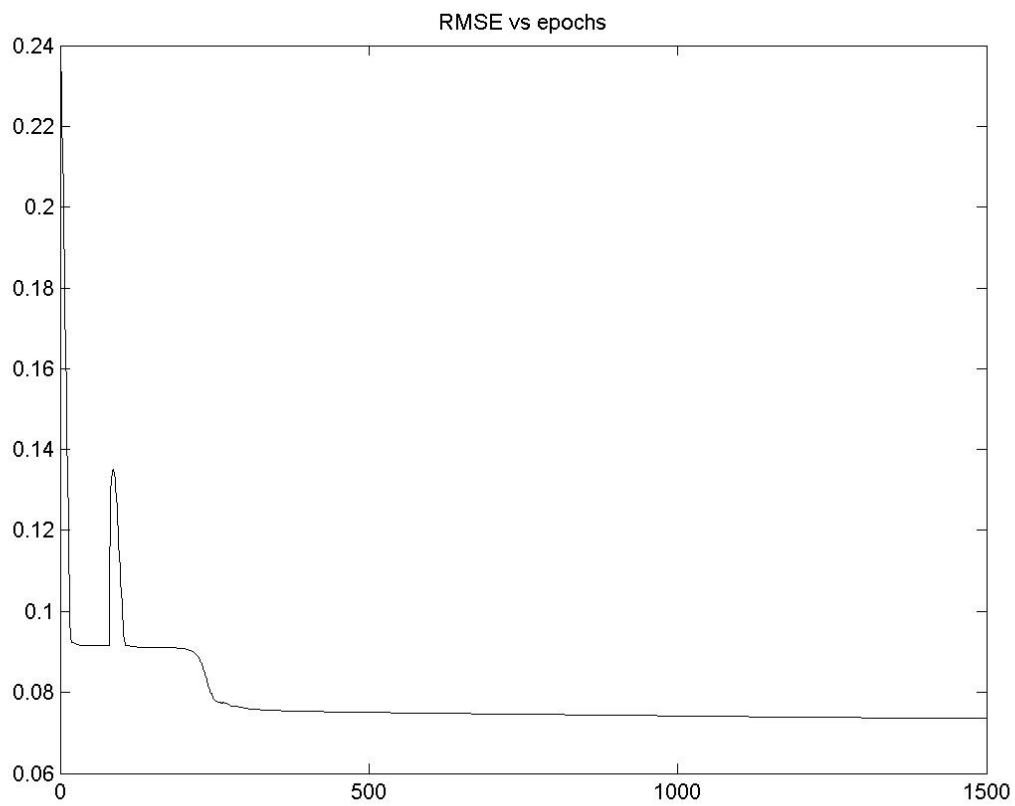
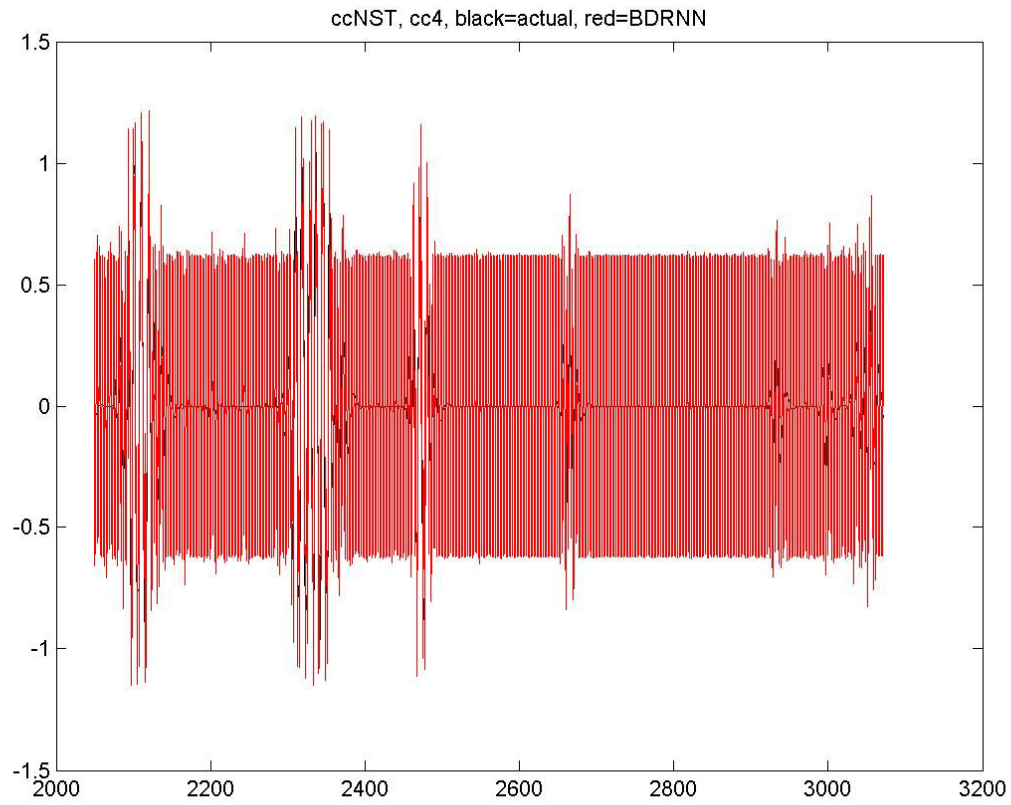


18^η Επανάληψη

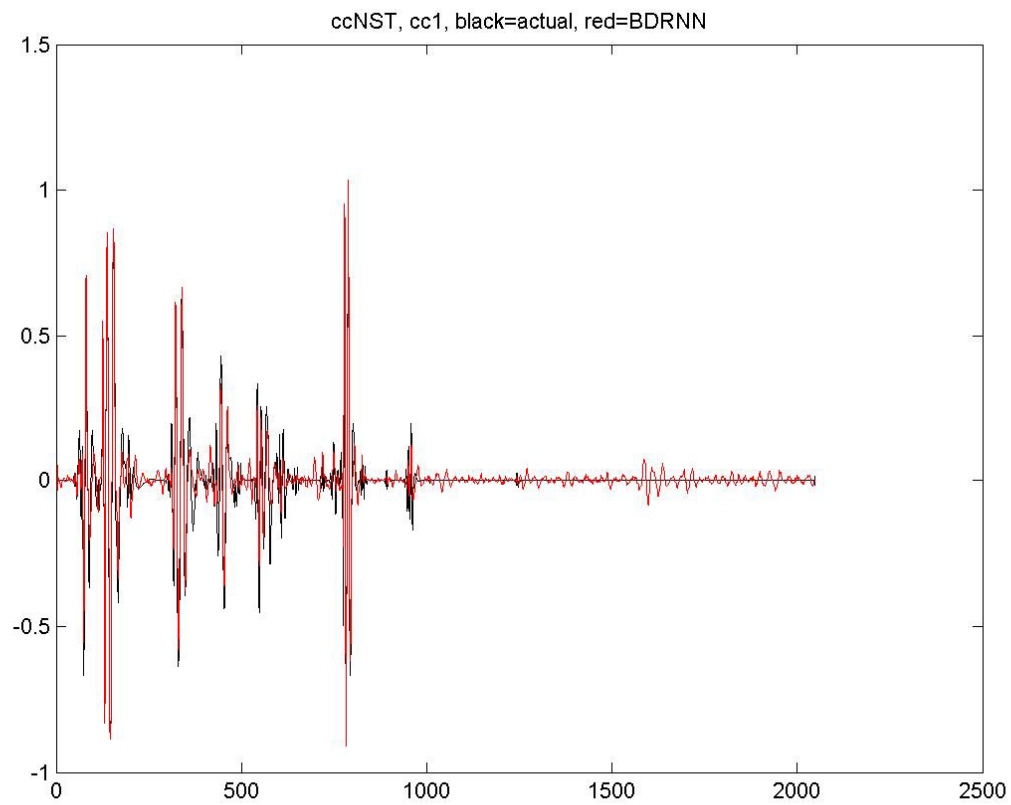
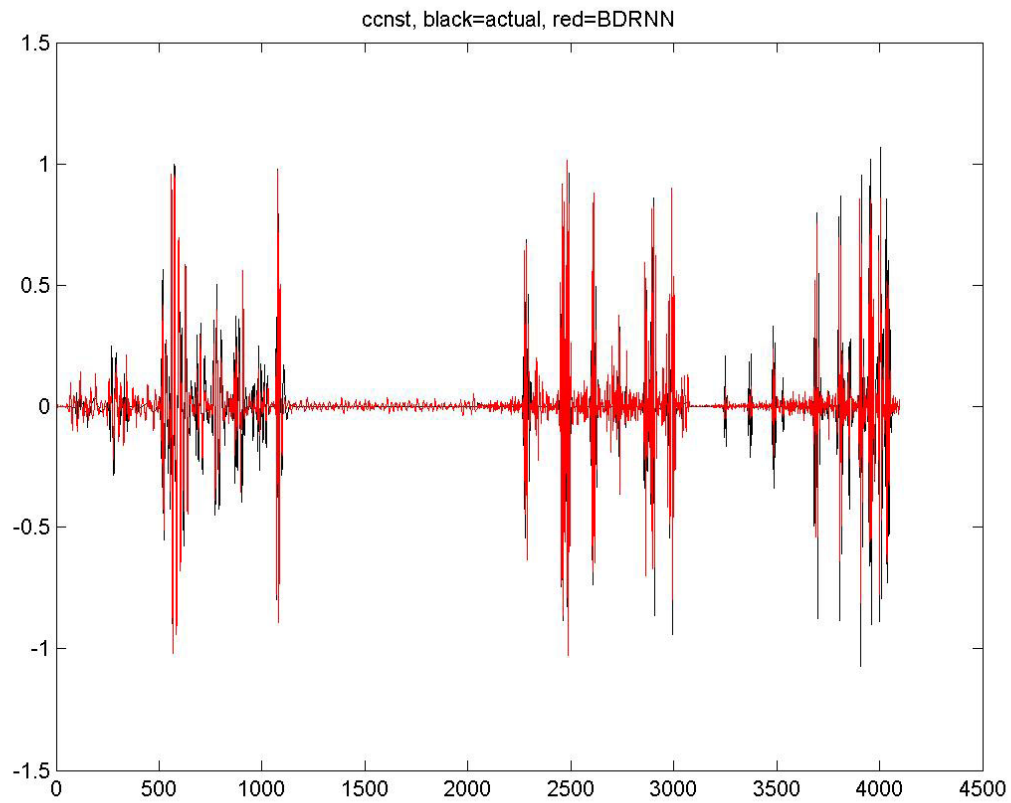


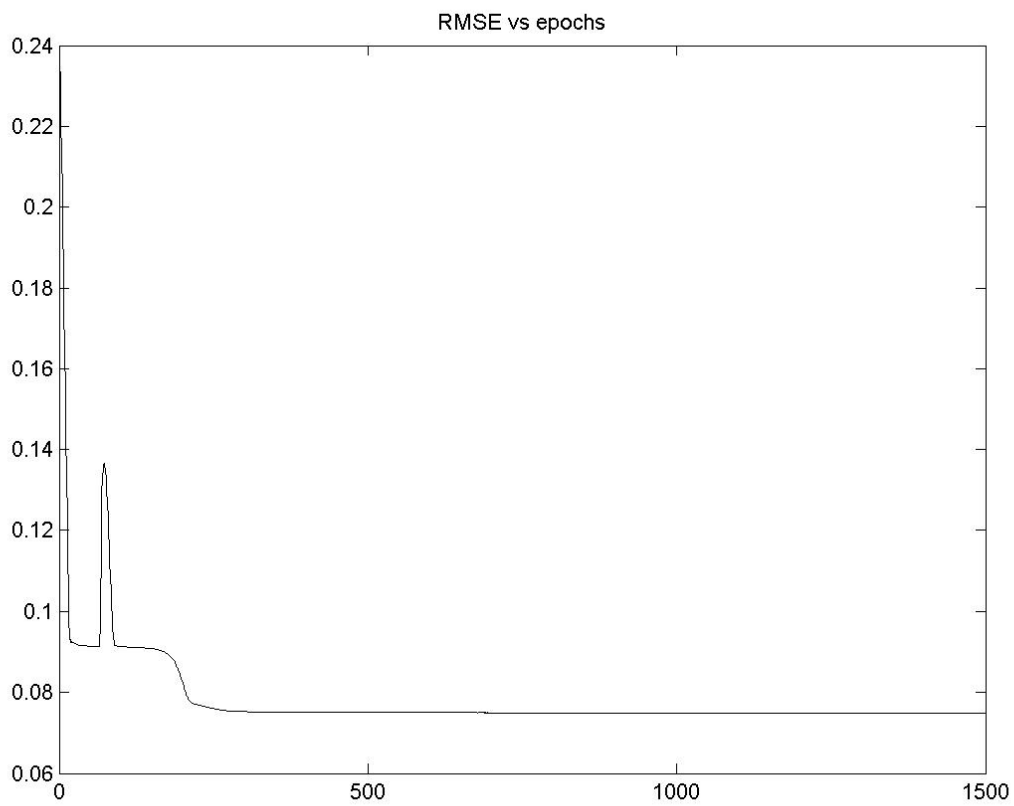
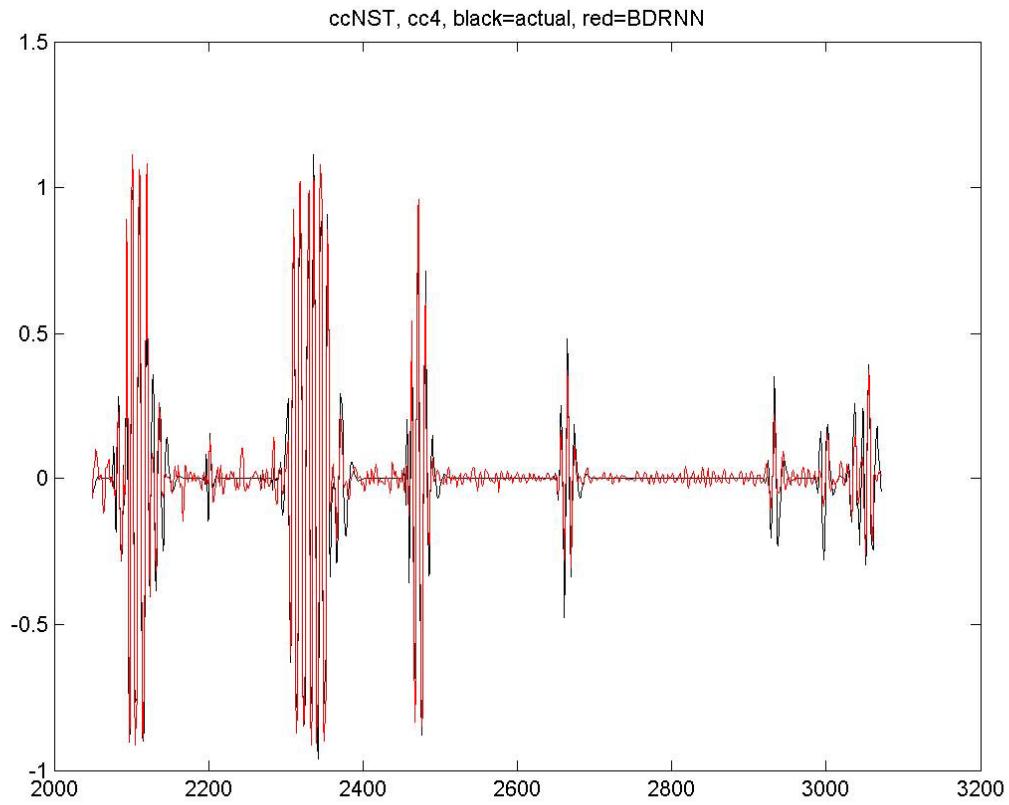
19^η Επανάληψη

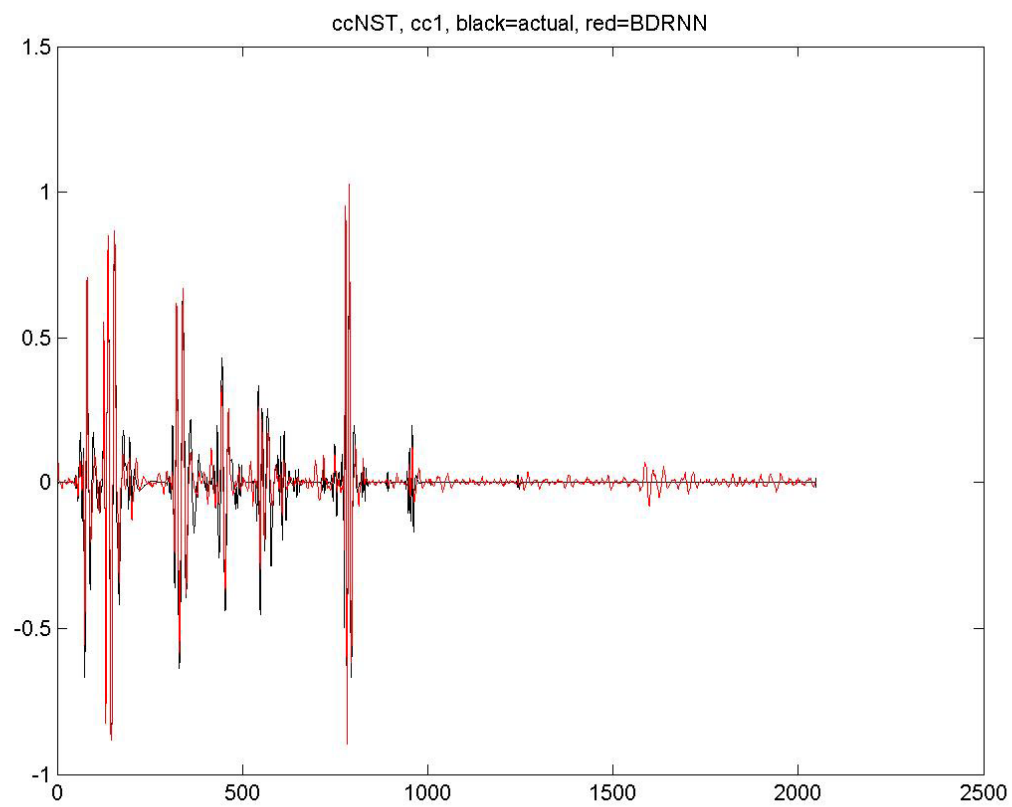
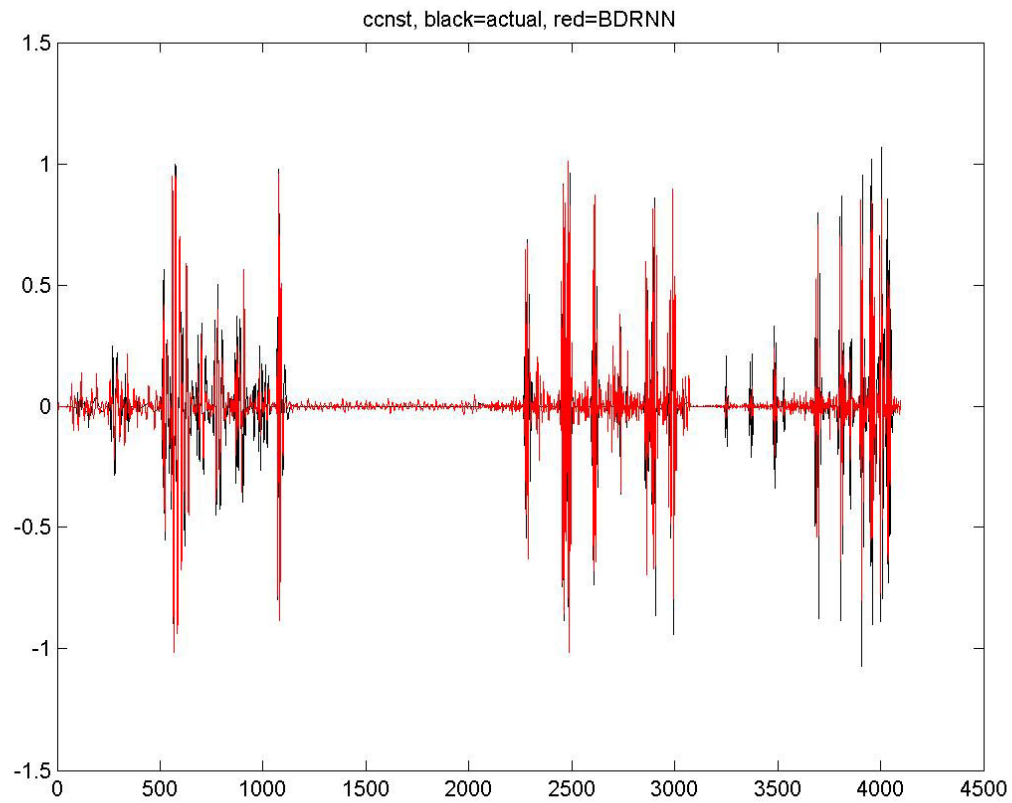


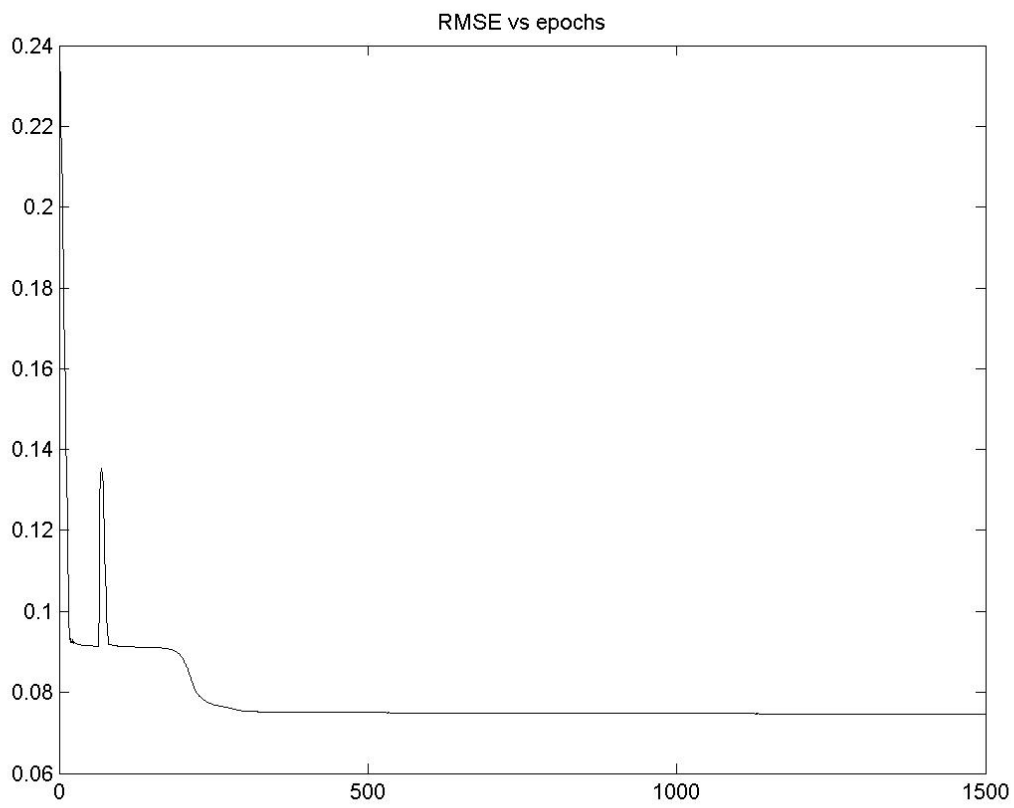
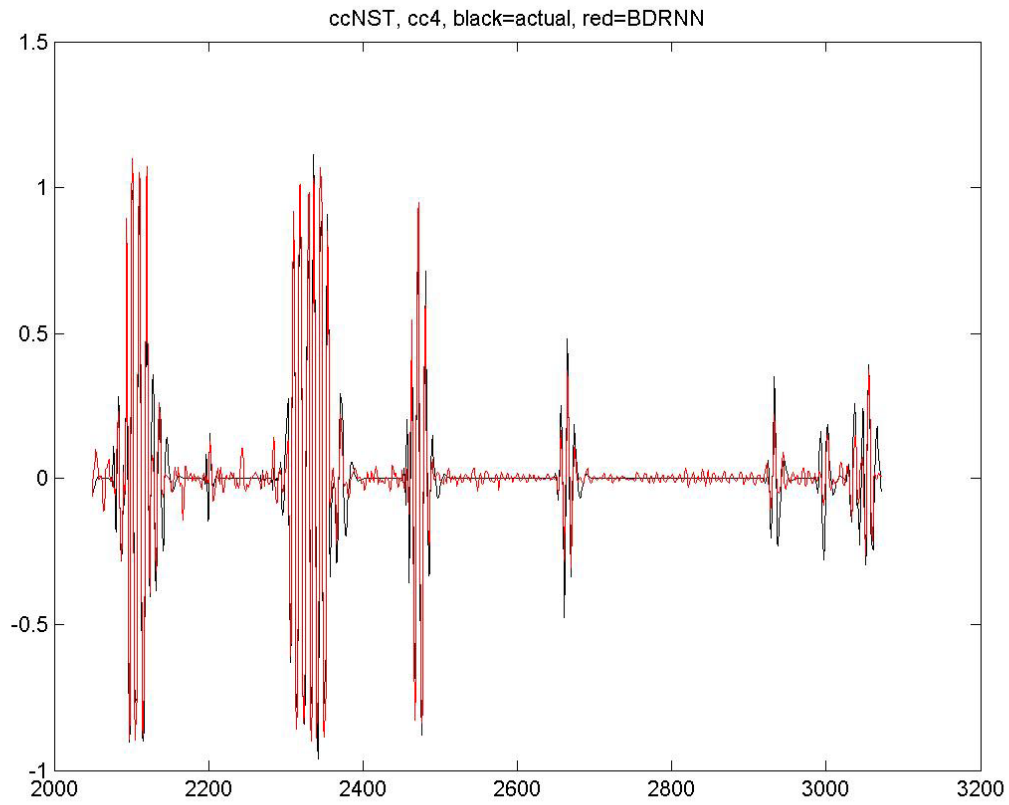


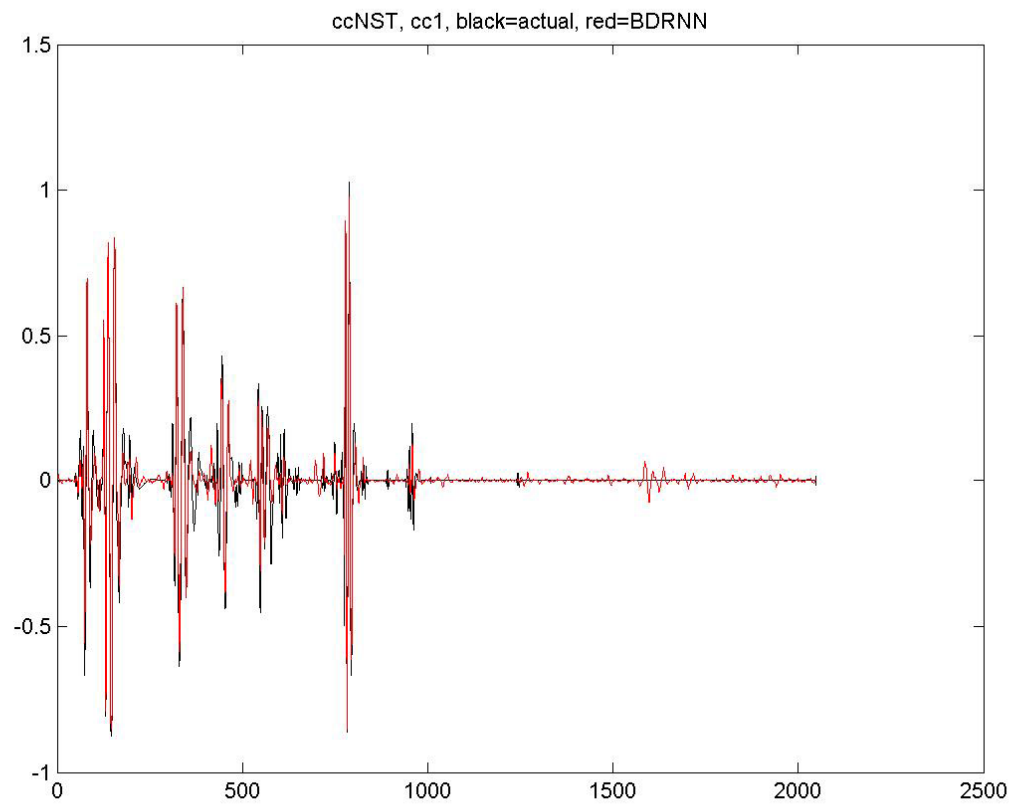
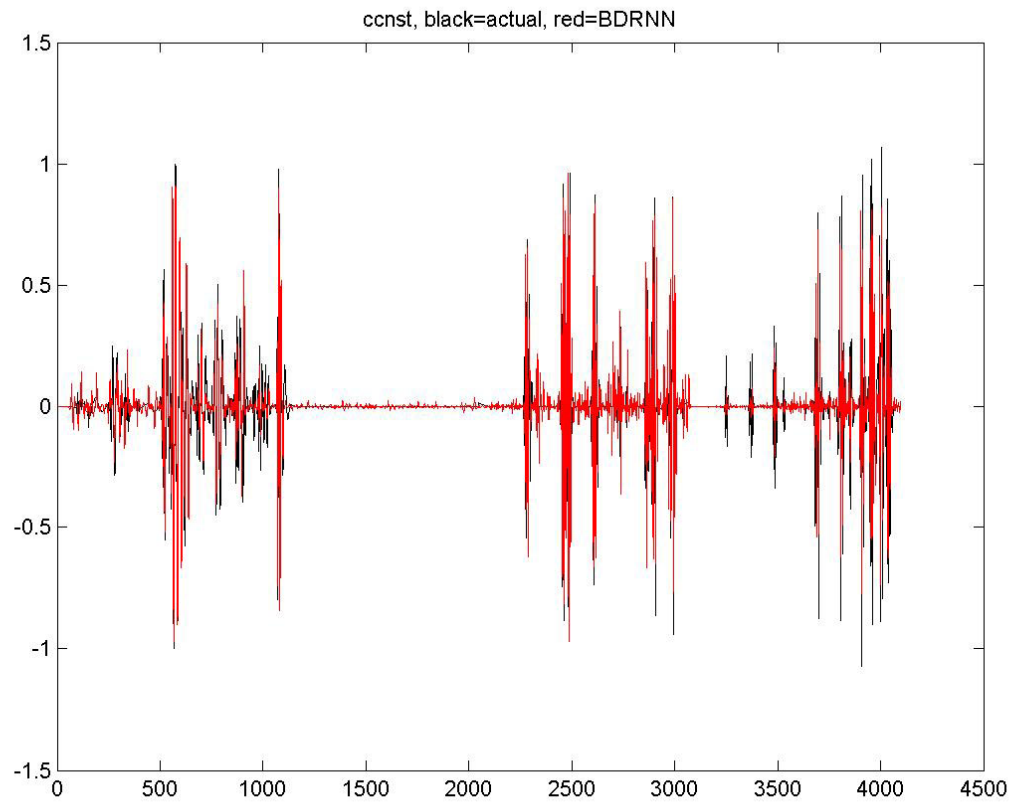
20^η Επανάληψη

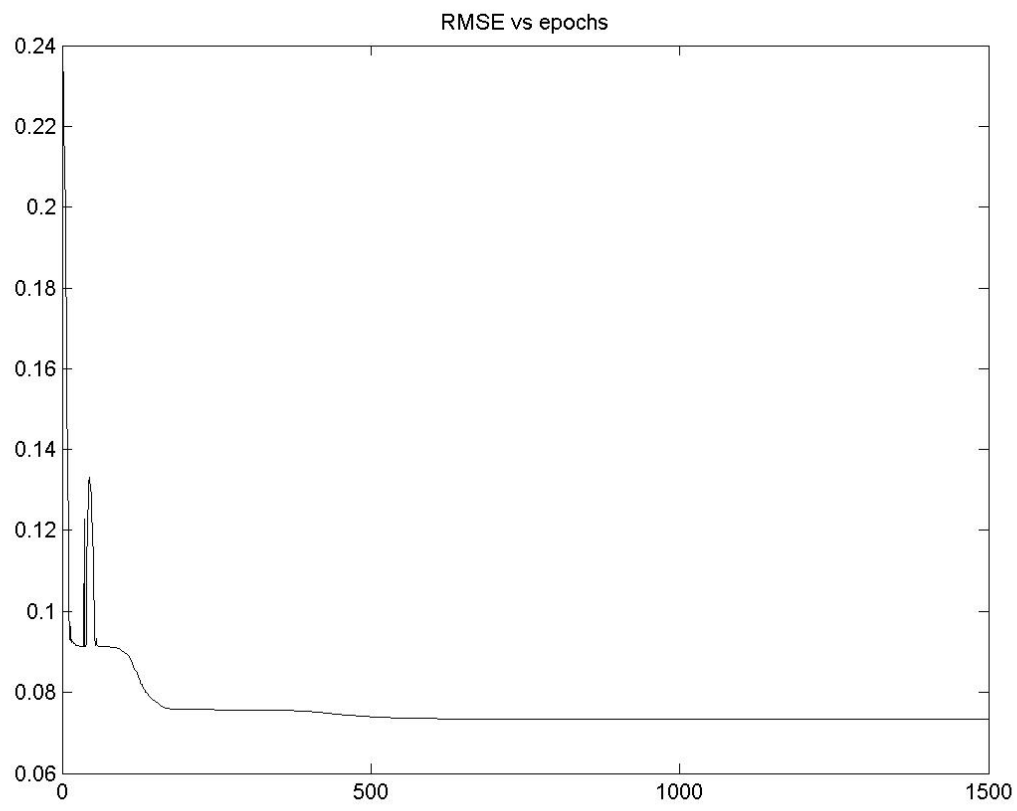
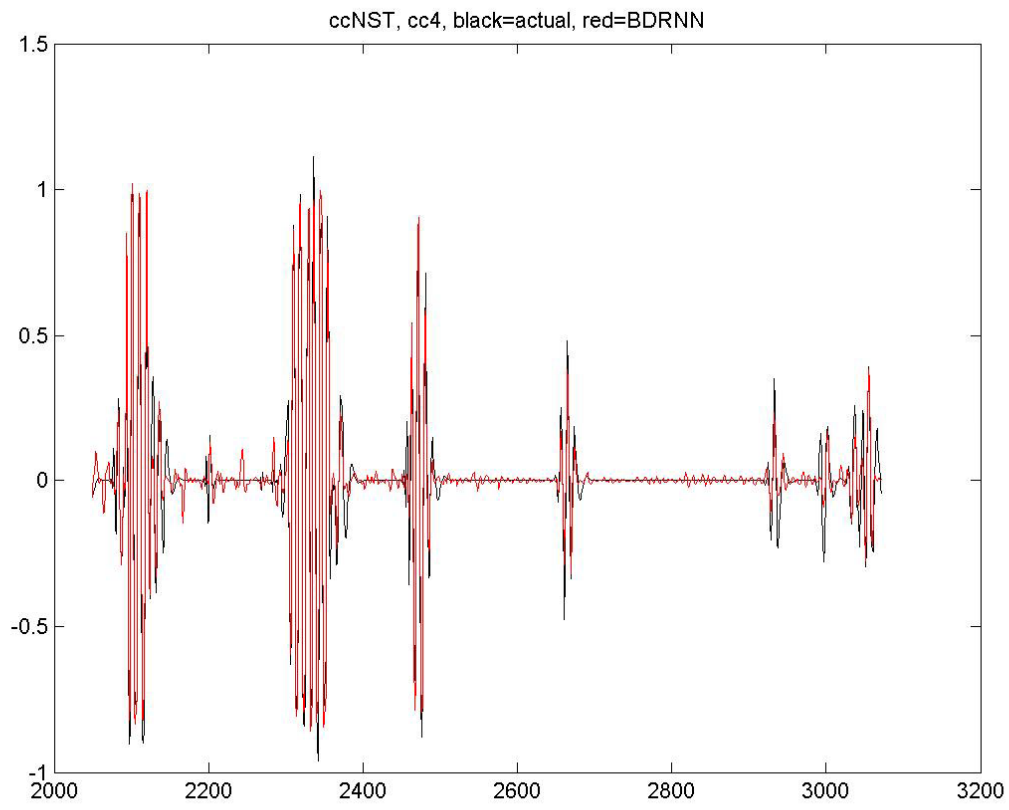


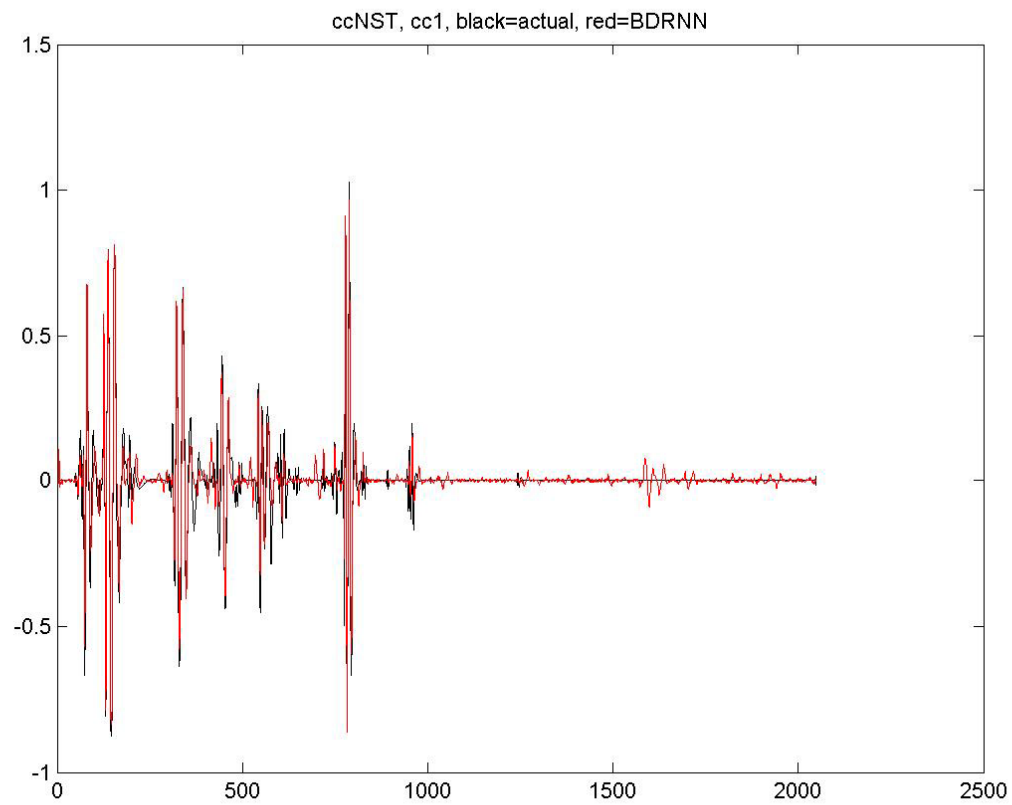
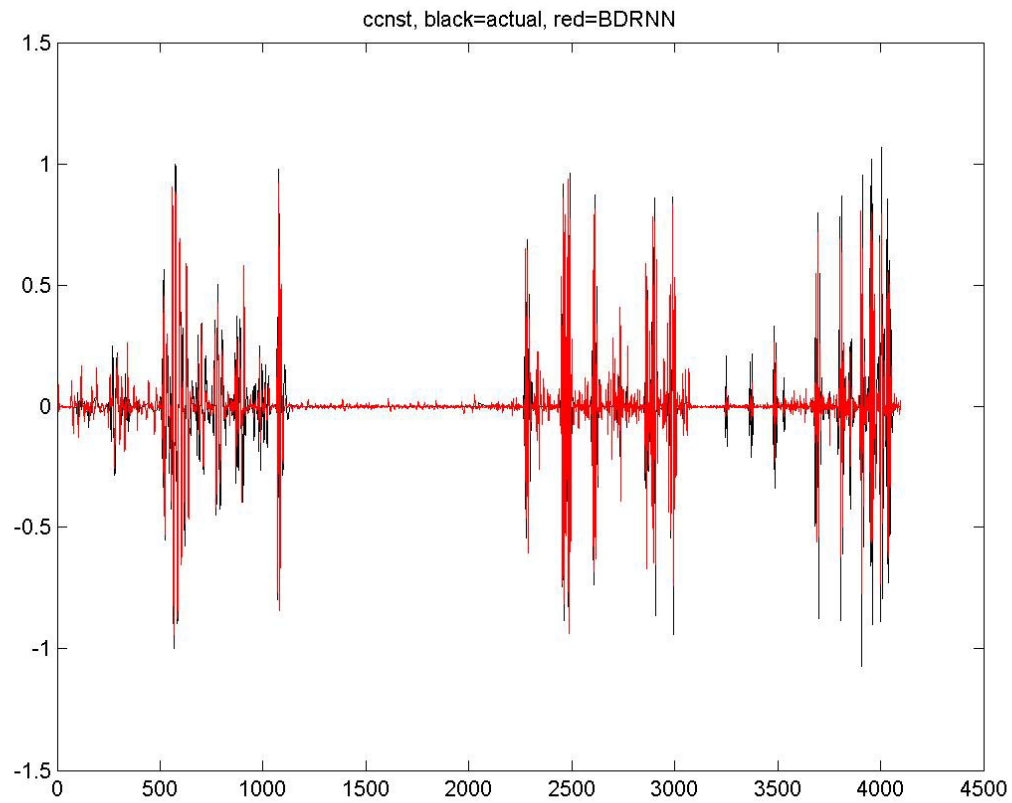


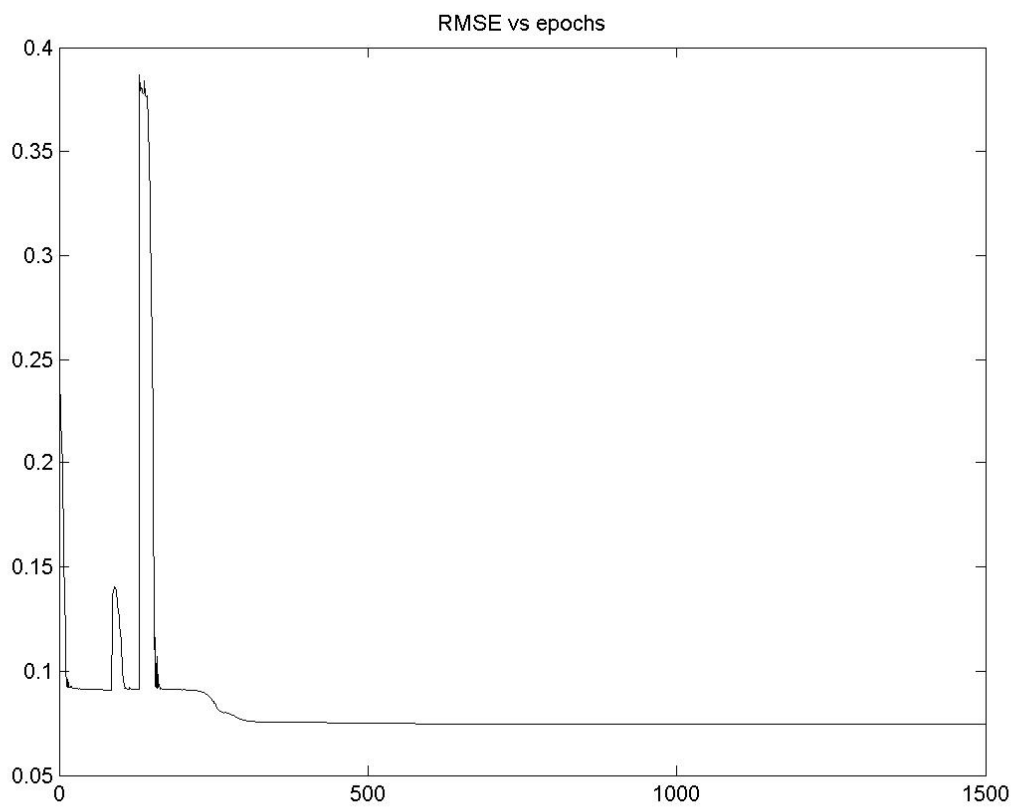
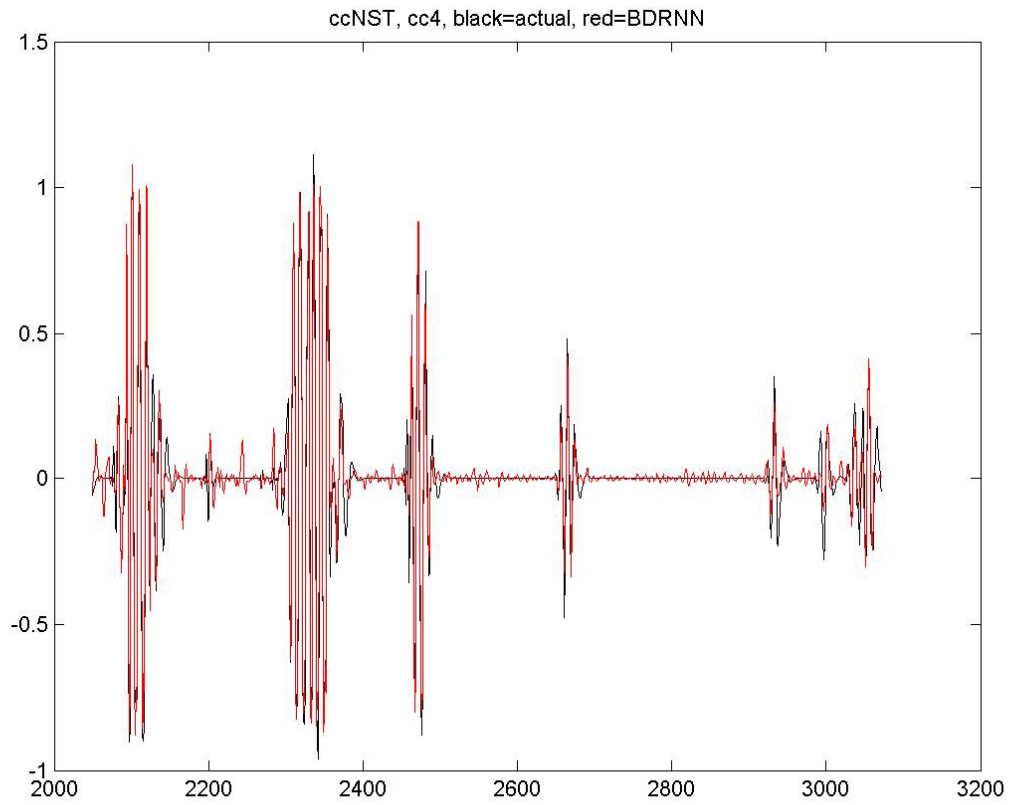
21^η Επανάληψη



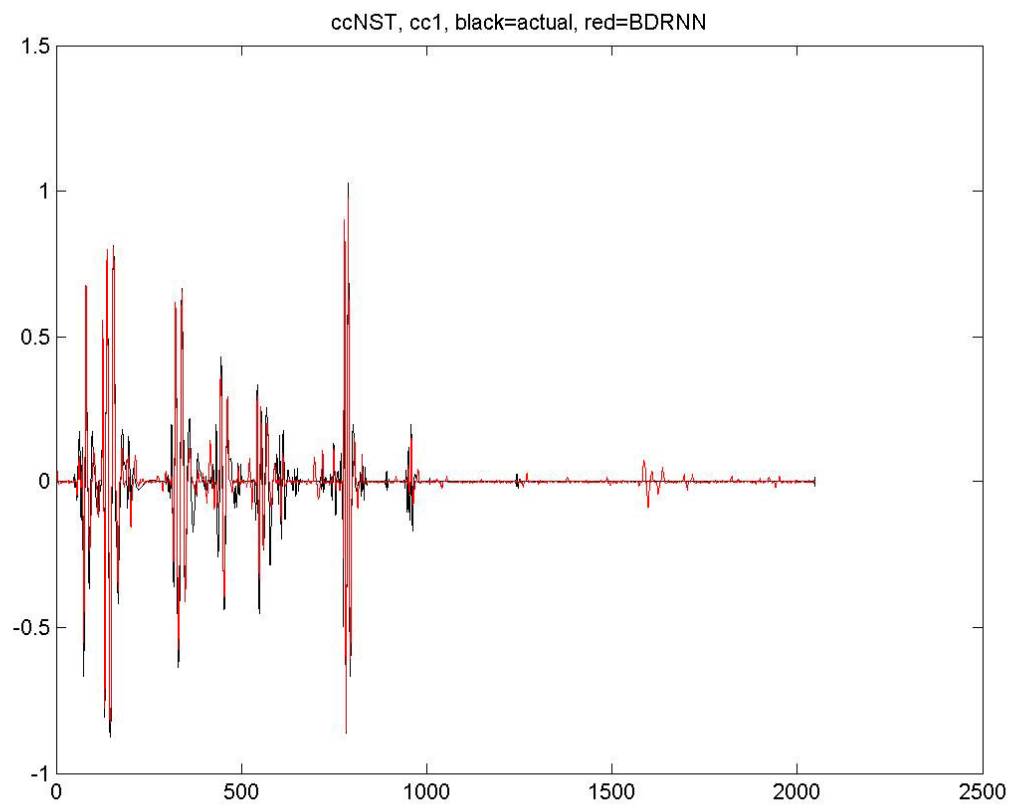
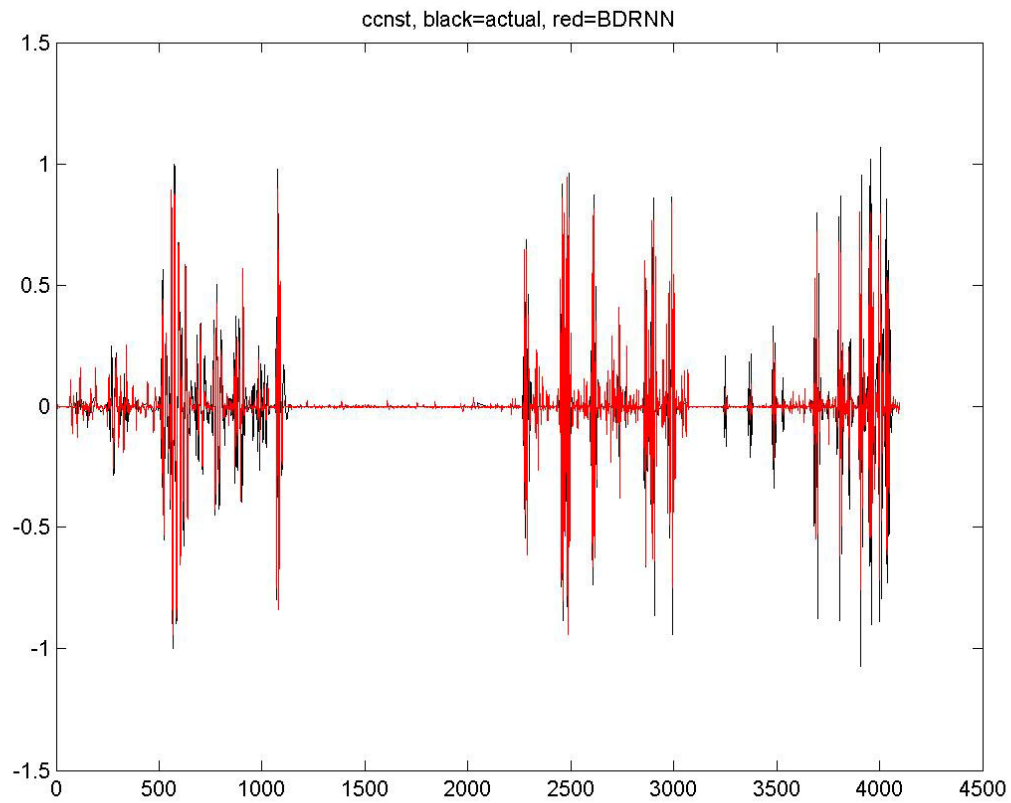
22^η Επανάληψη

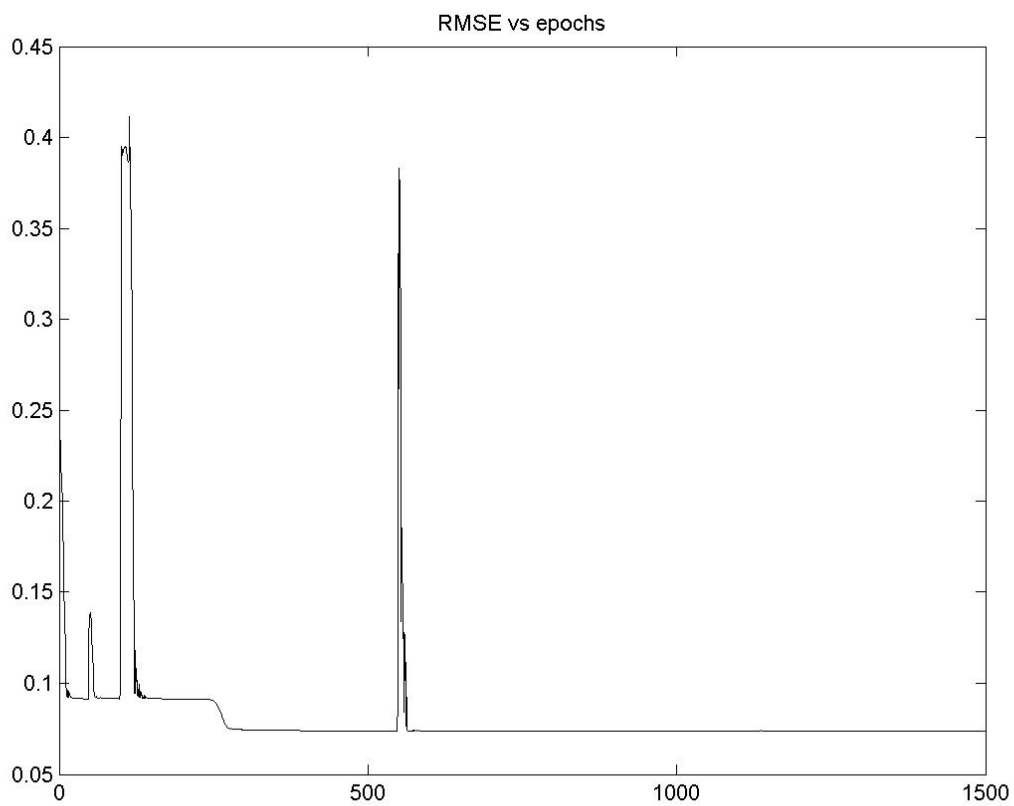
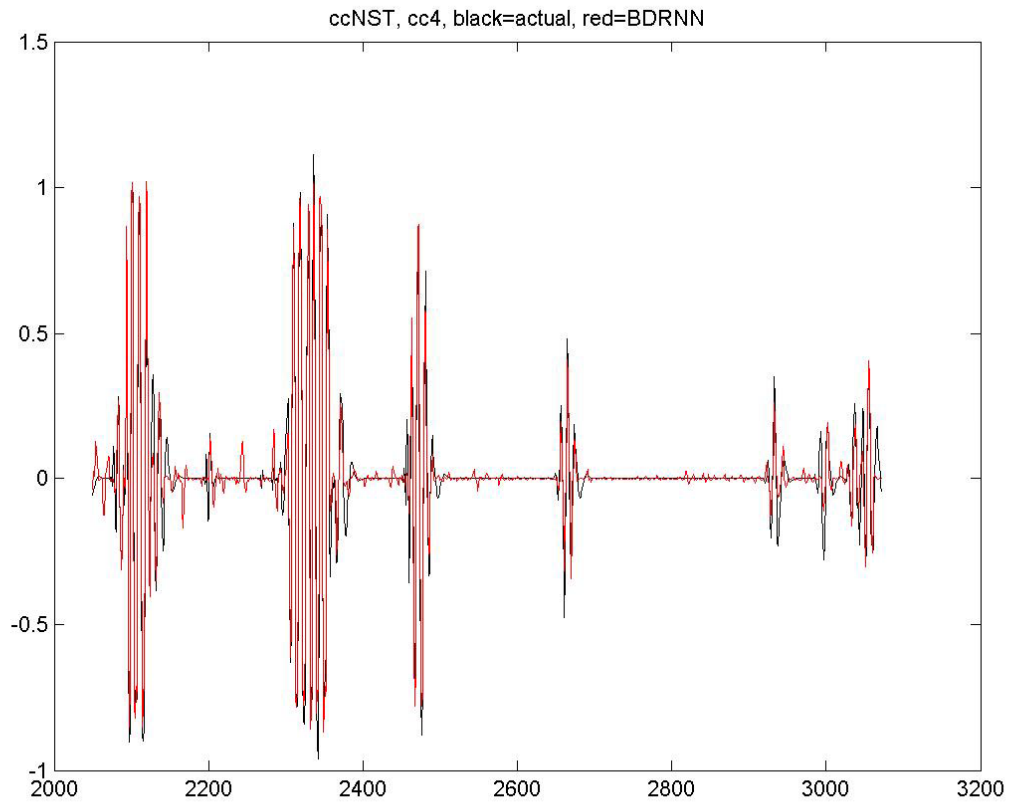


23^η Επανάληψη

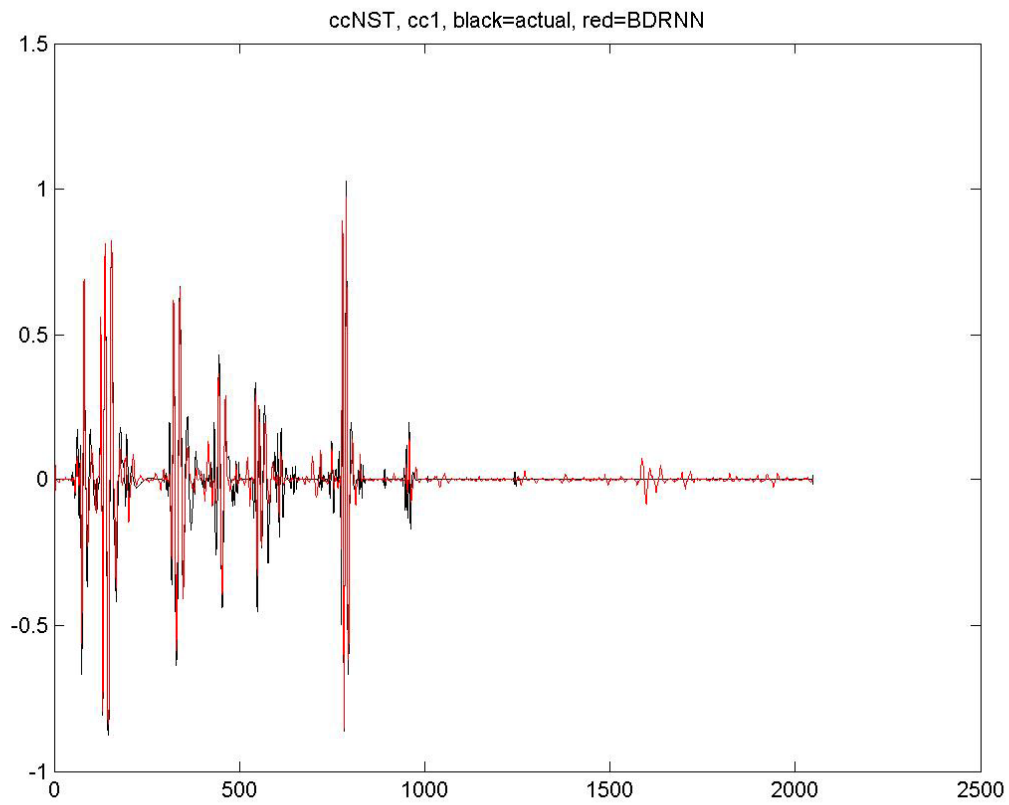
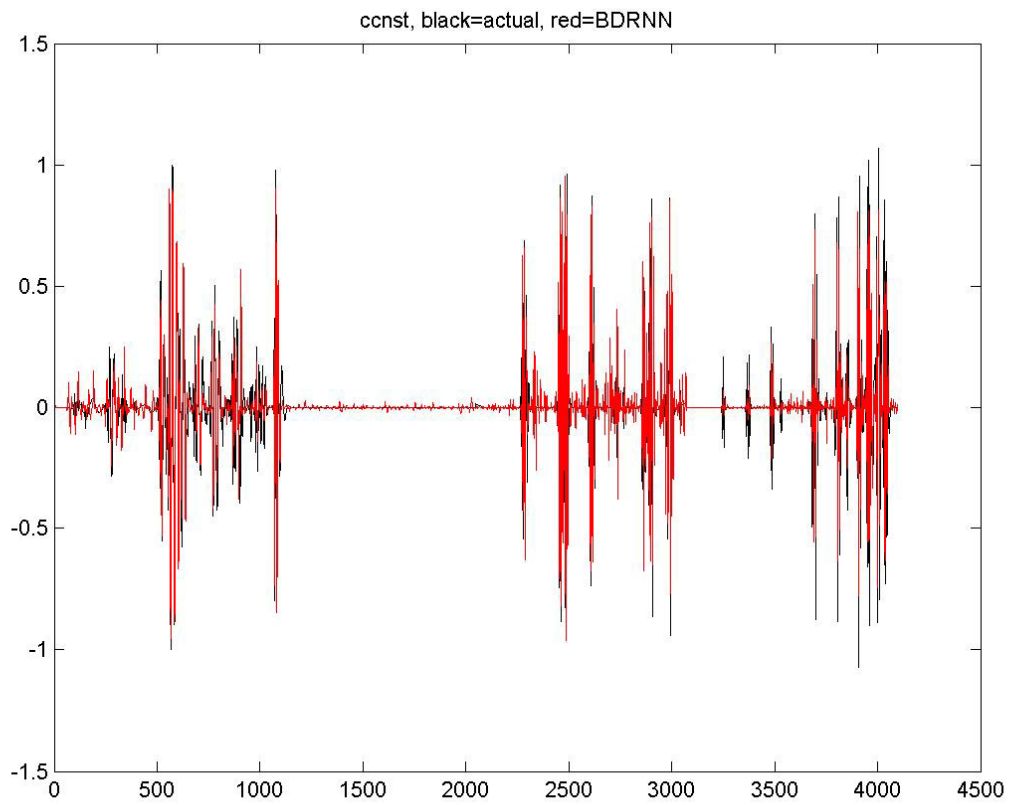


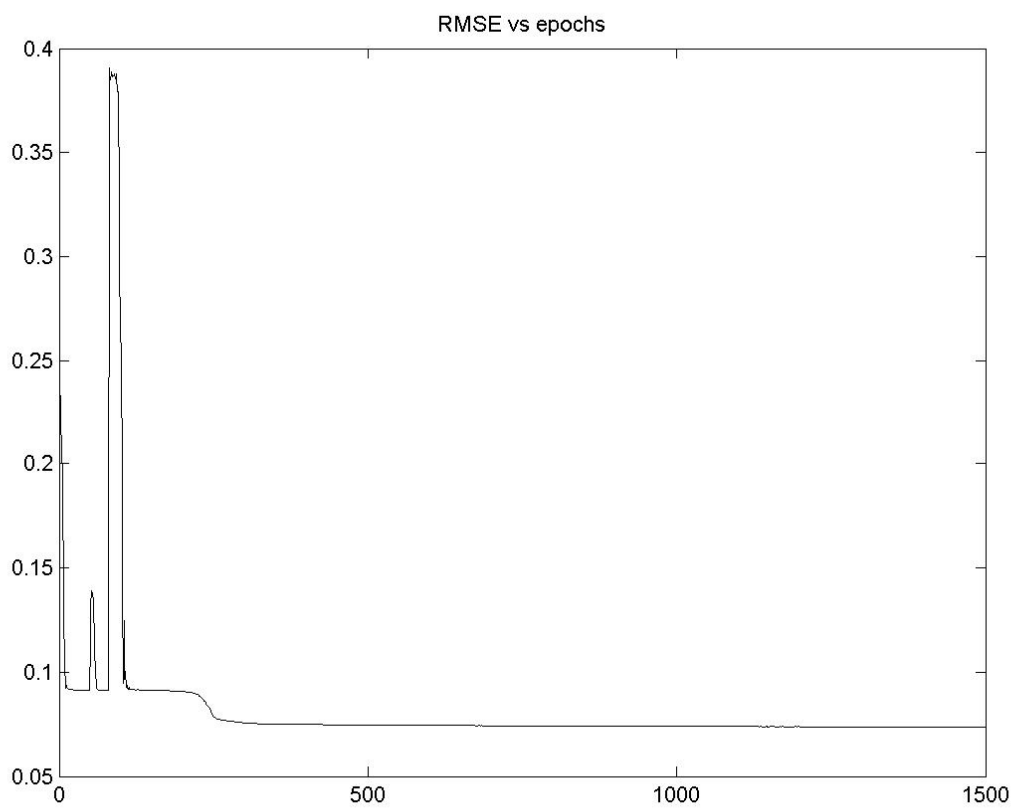
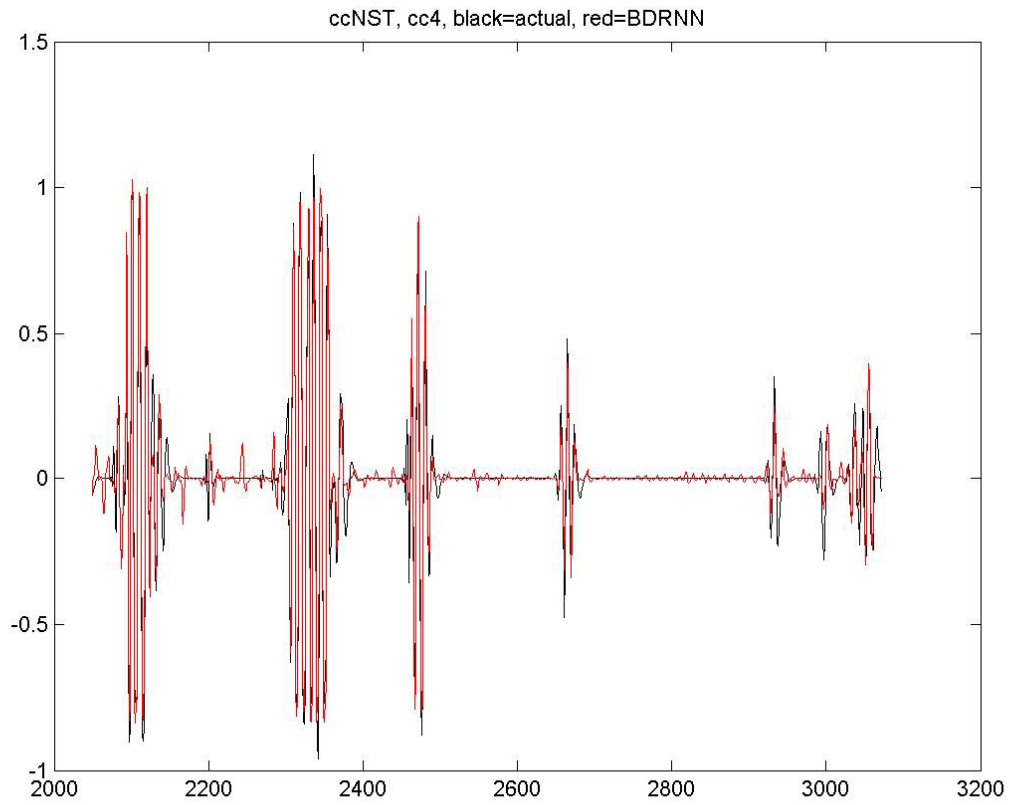
24^η Επανάληψη



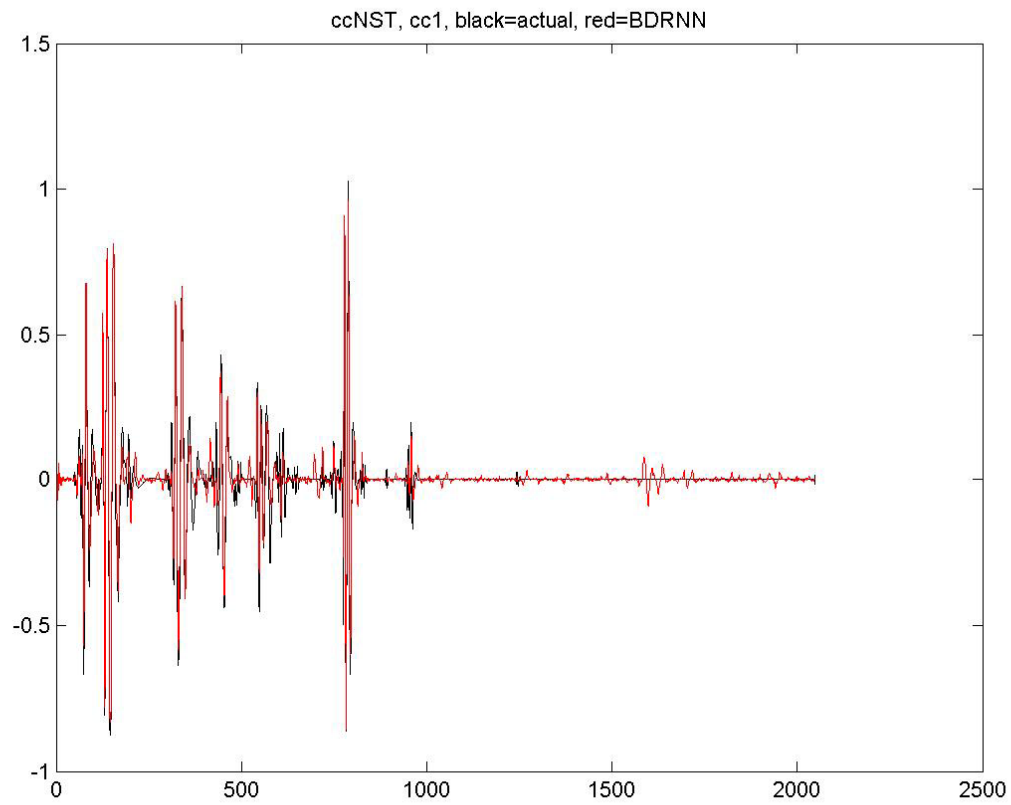
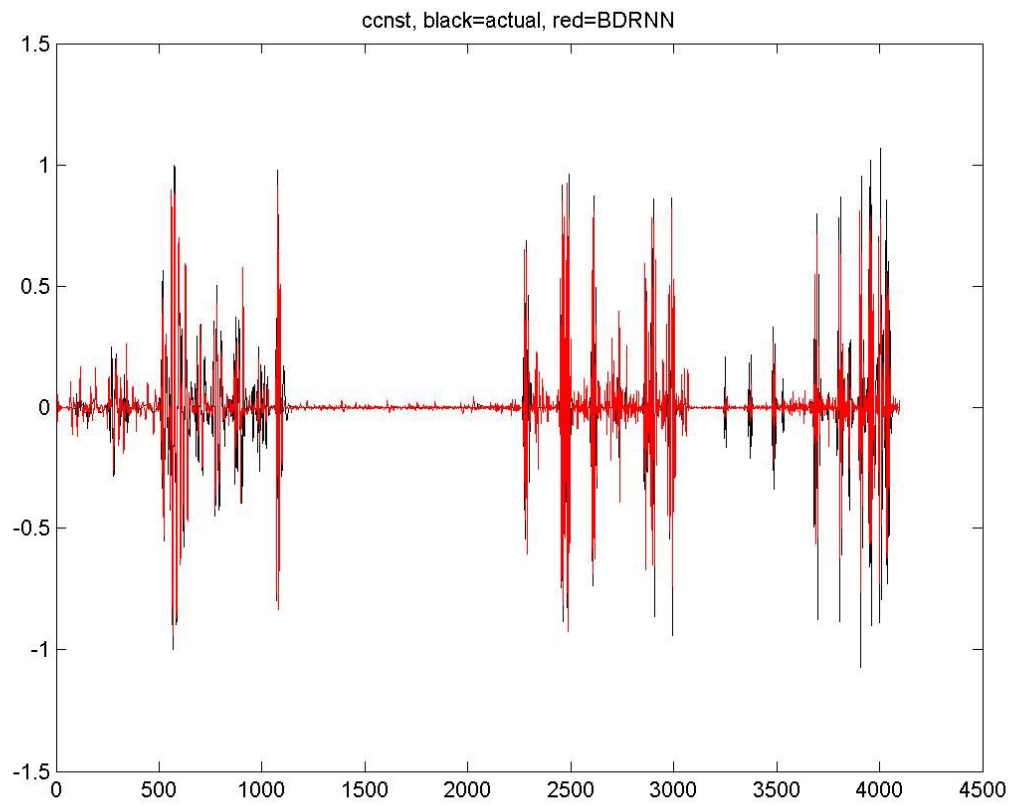


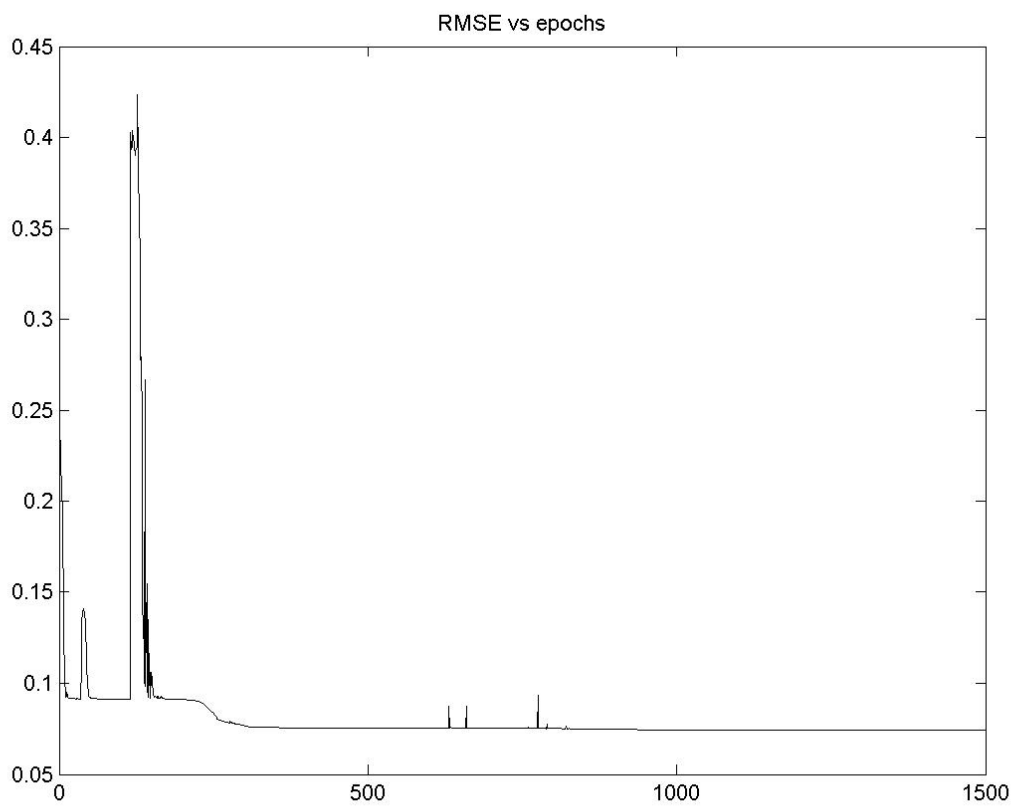
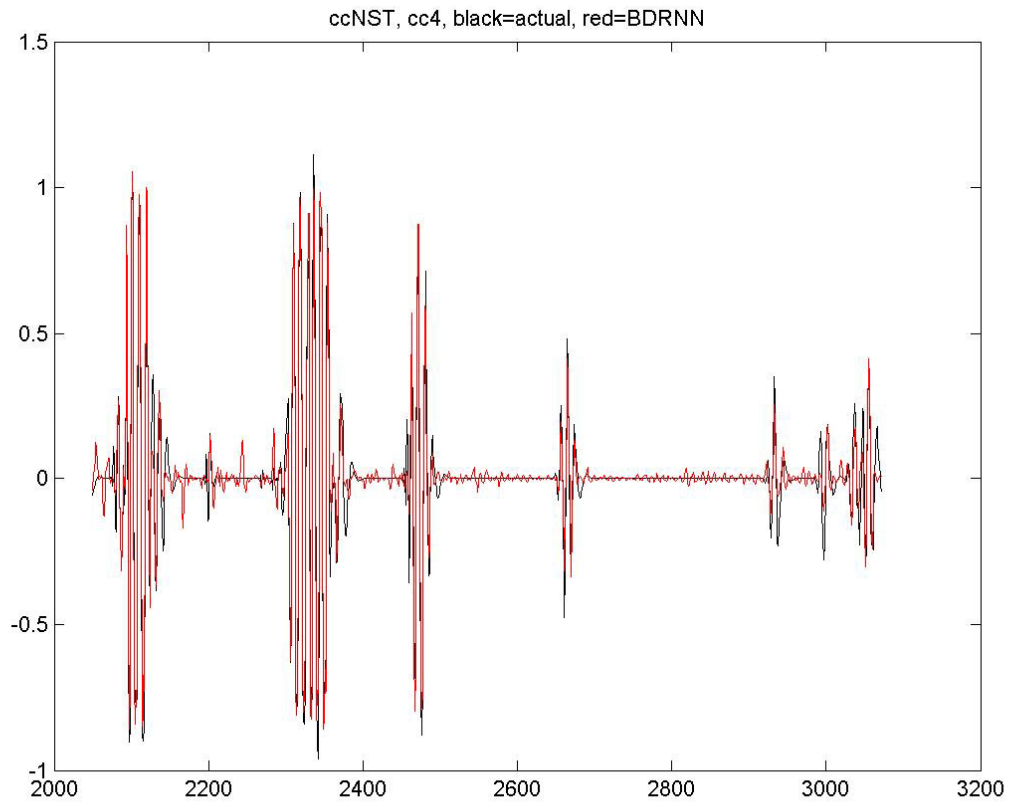
25^η Επανάληψη



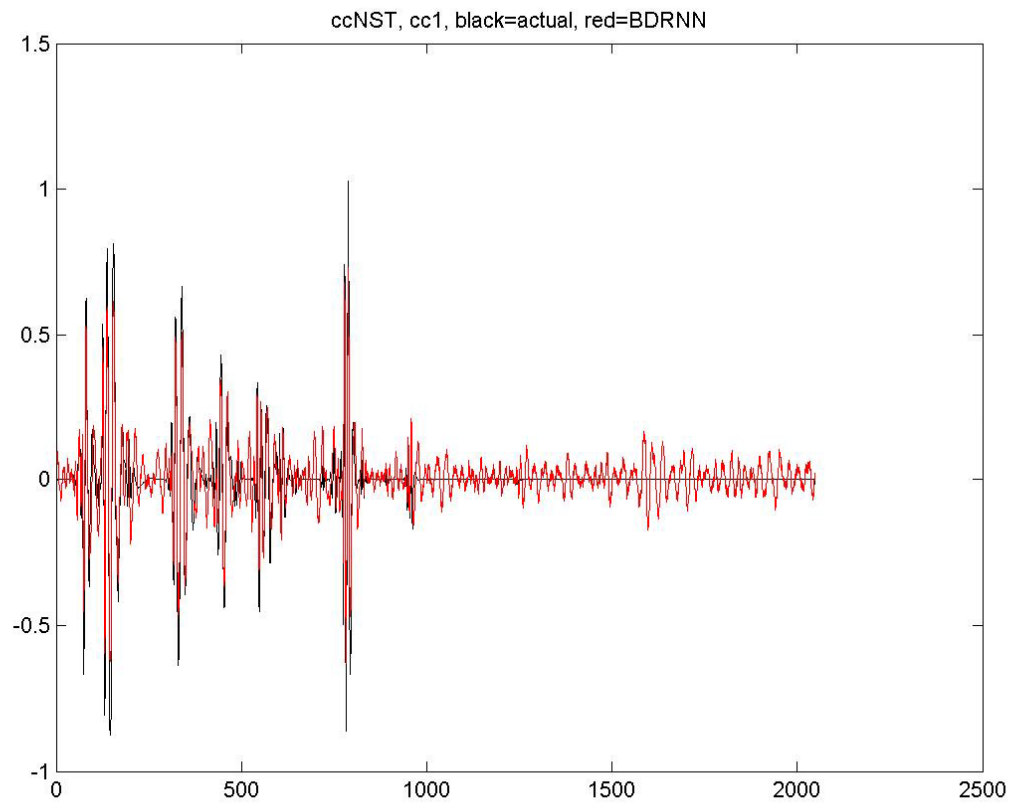
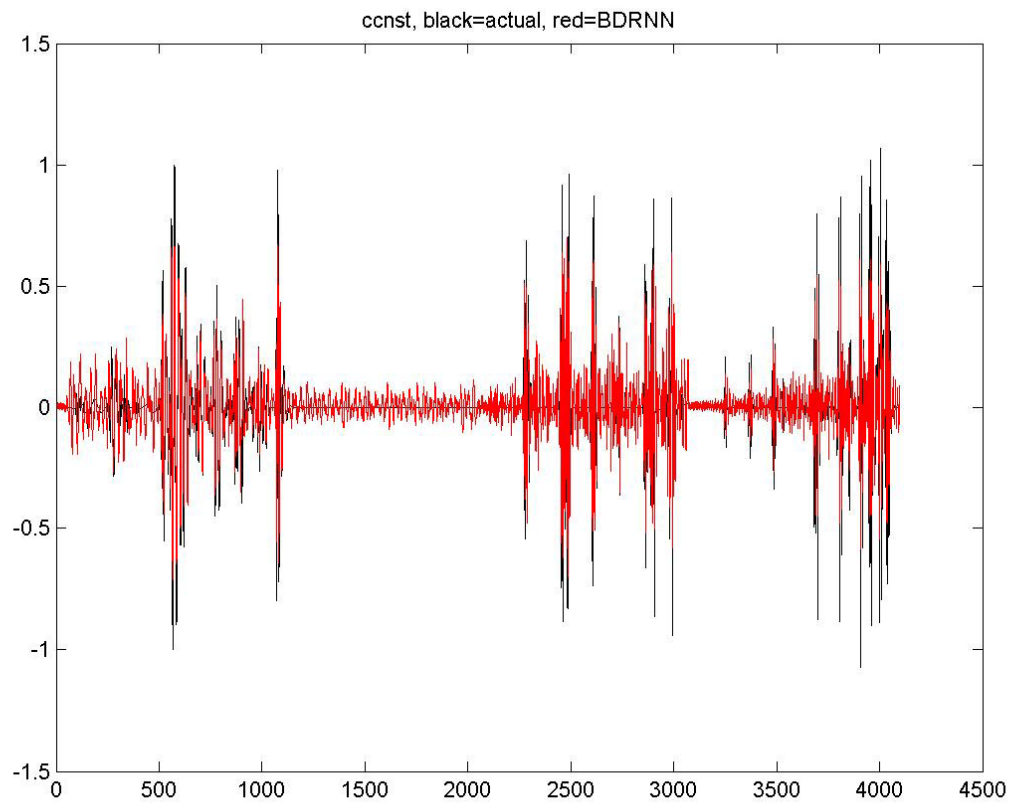


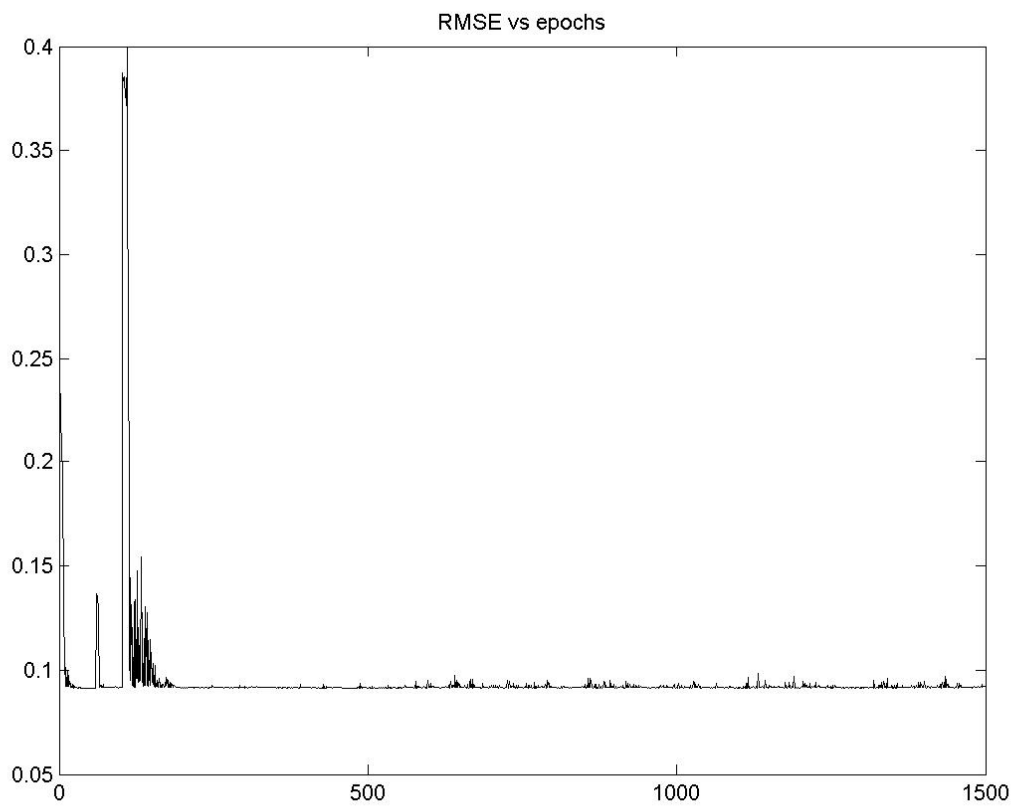
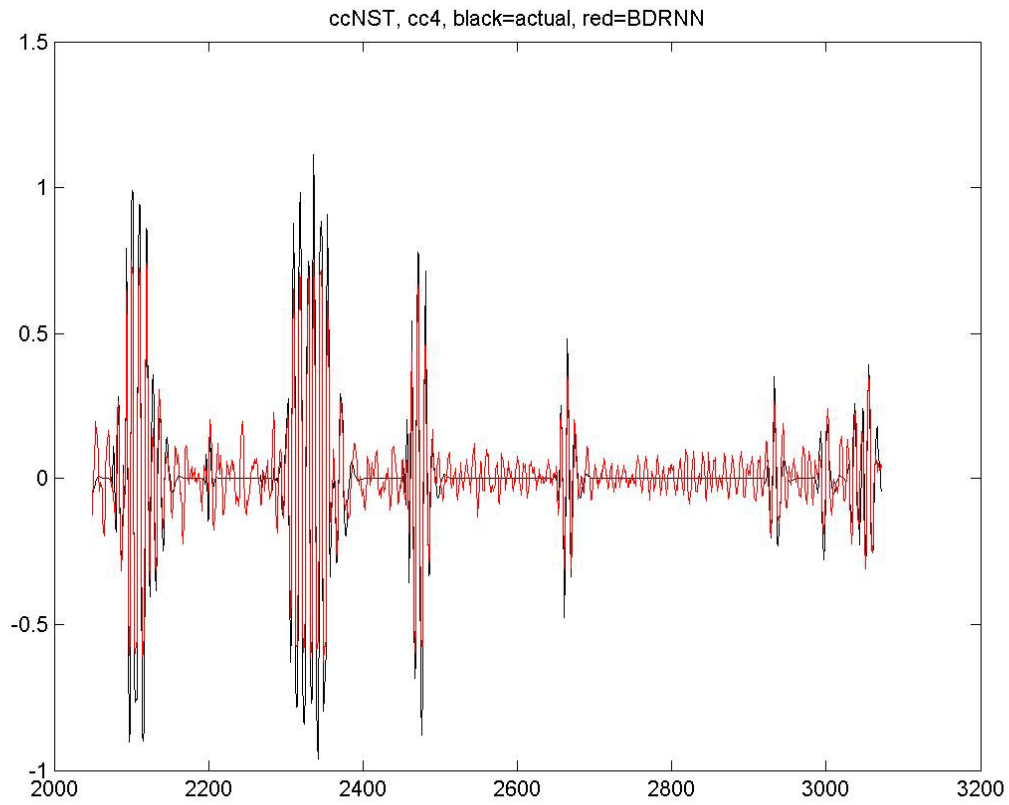
26^η Επανάληψη



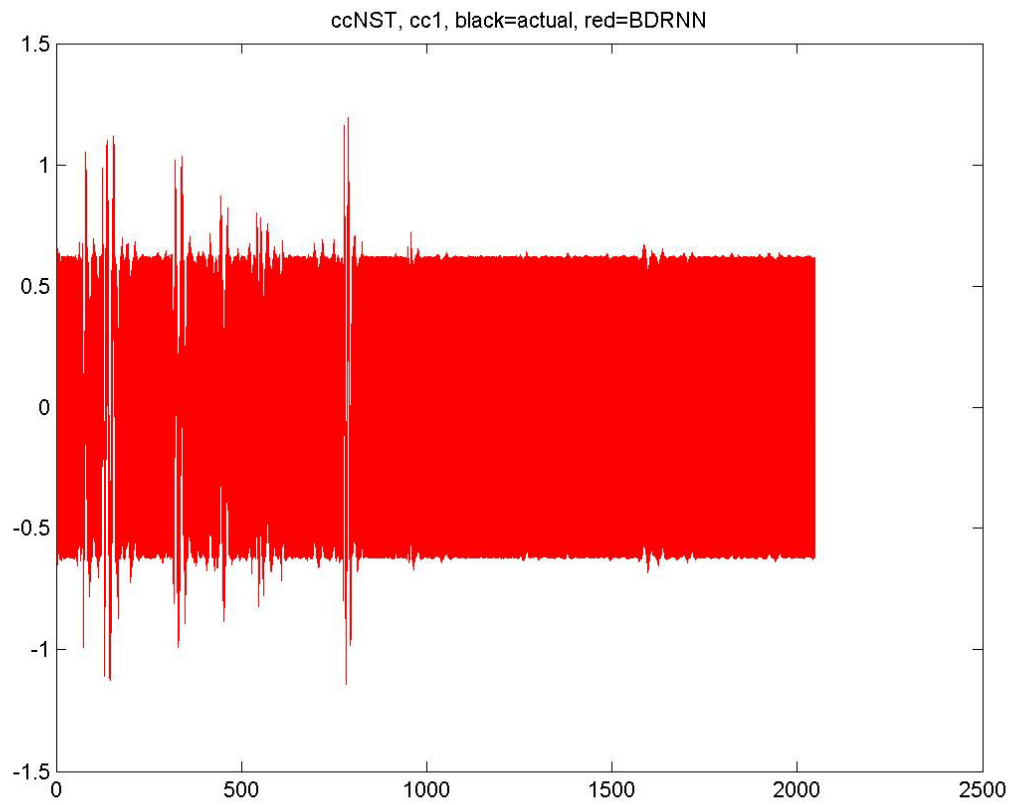
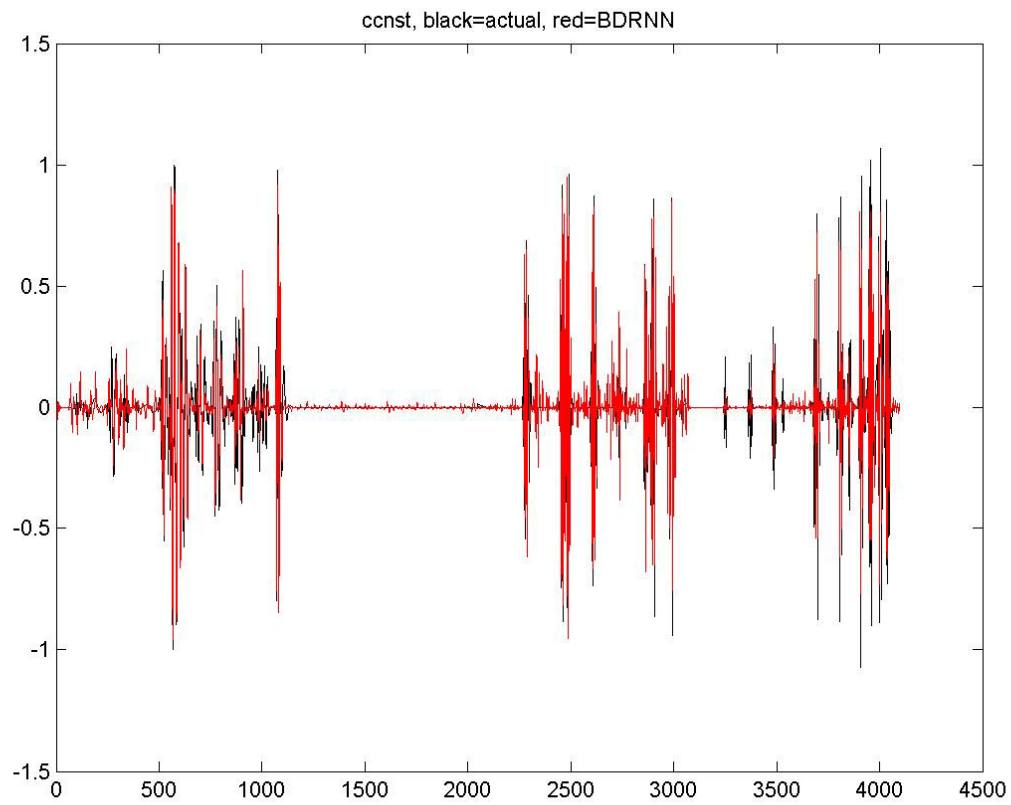


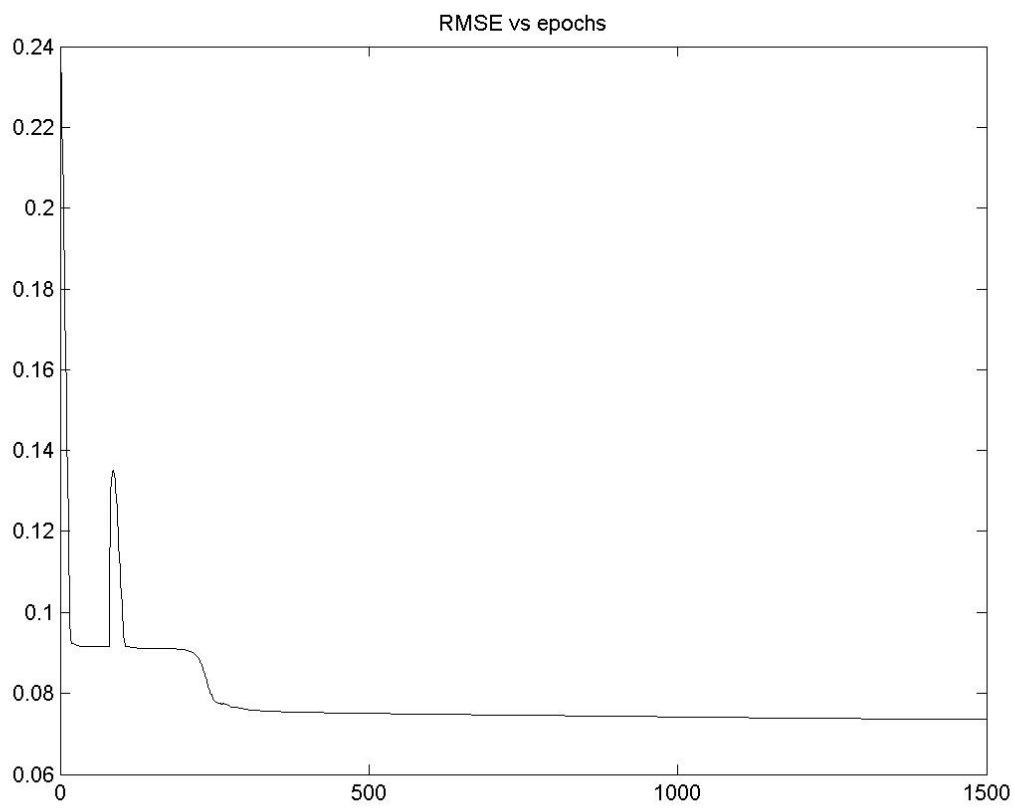
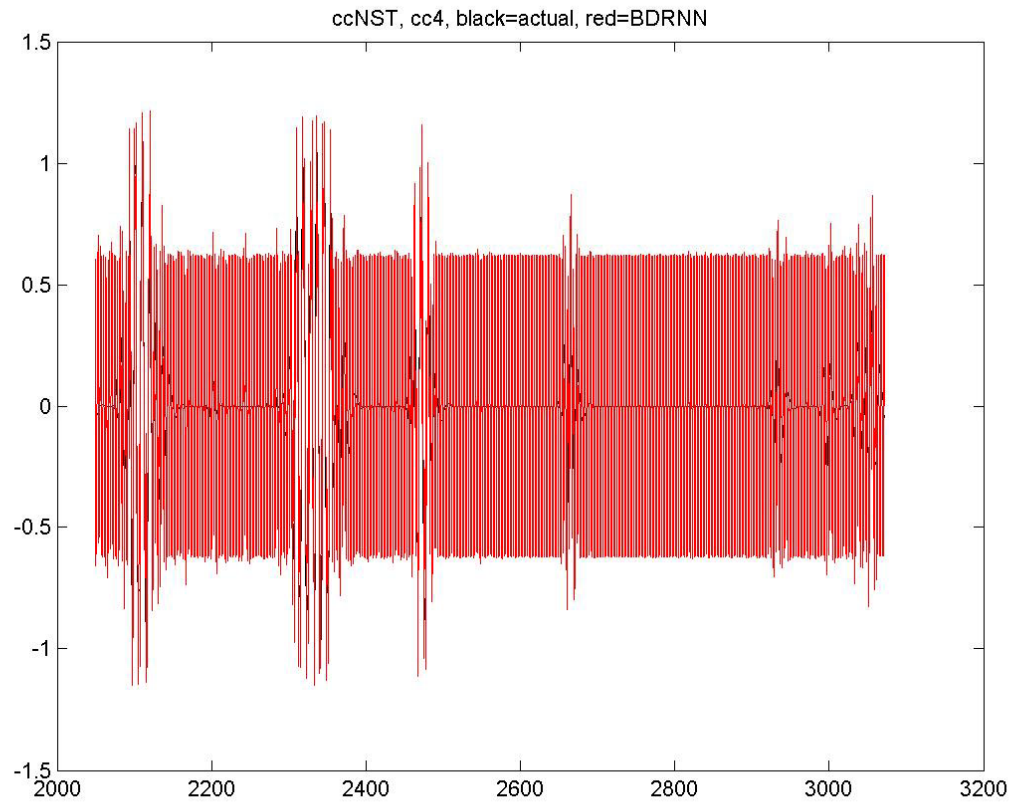
27^η Επανάληψη



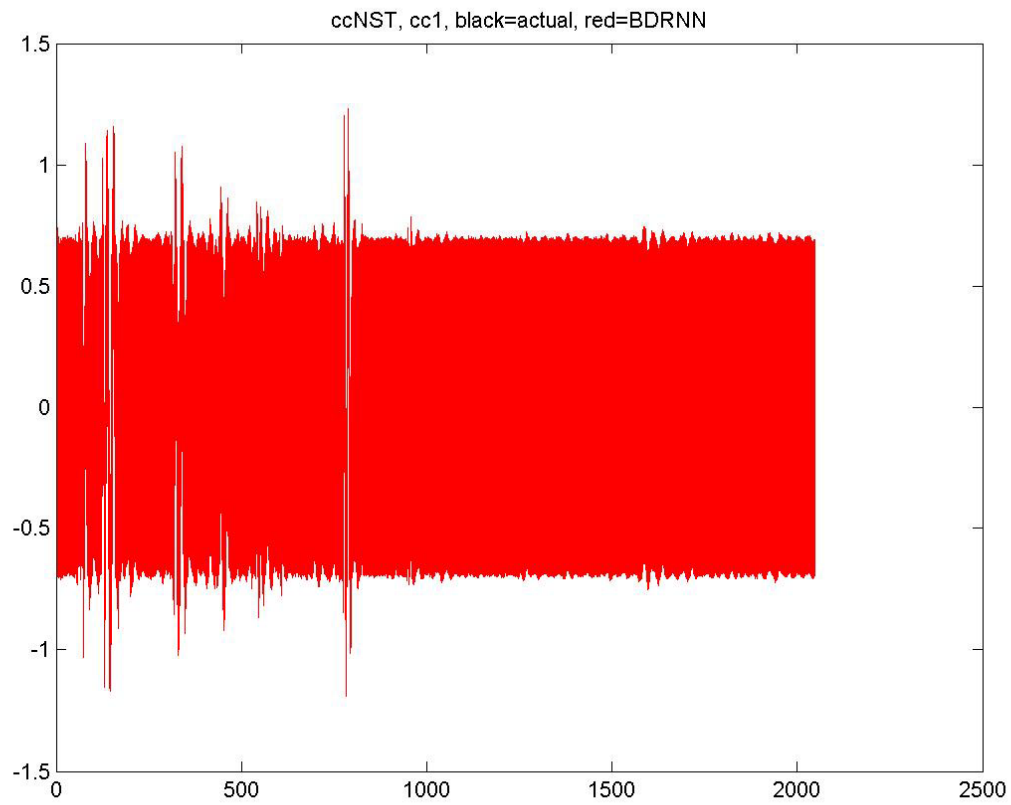
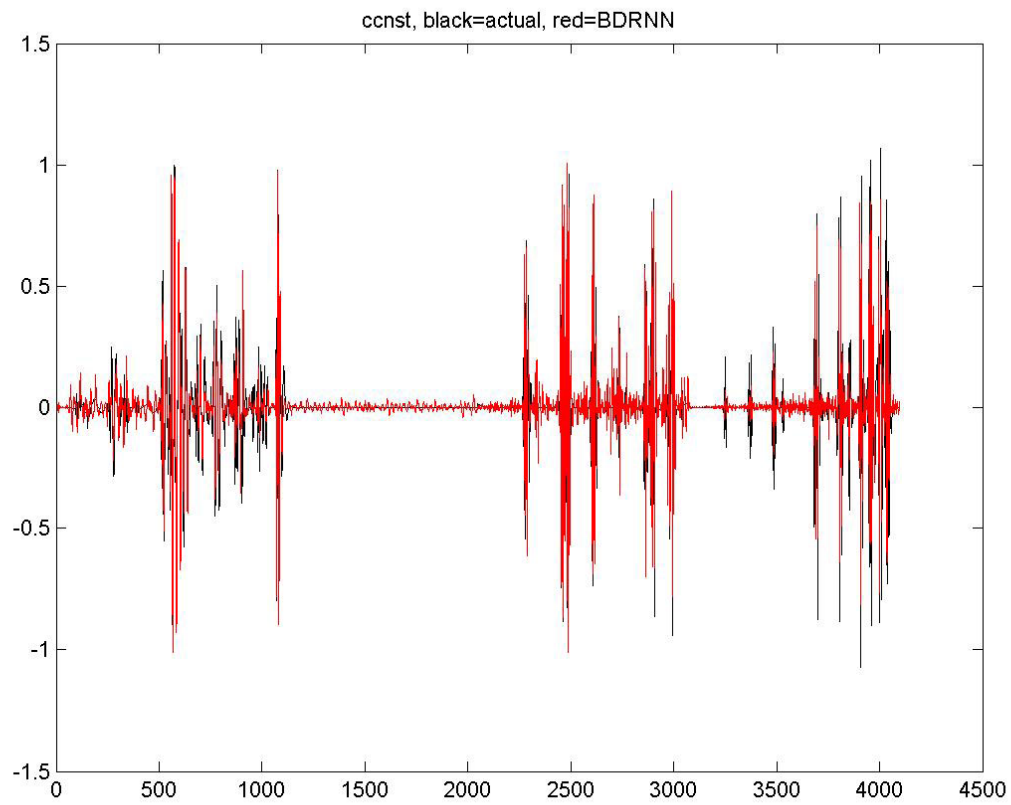


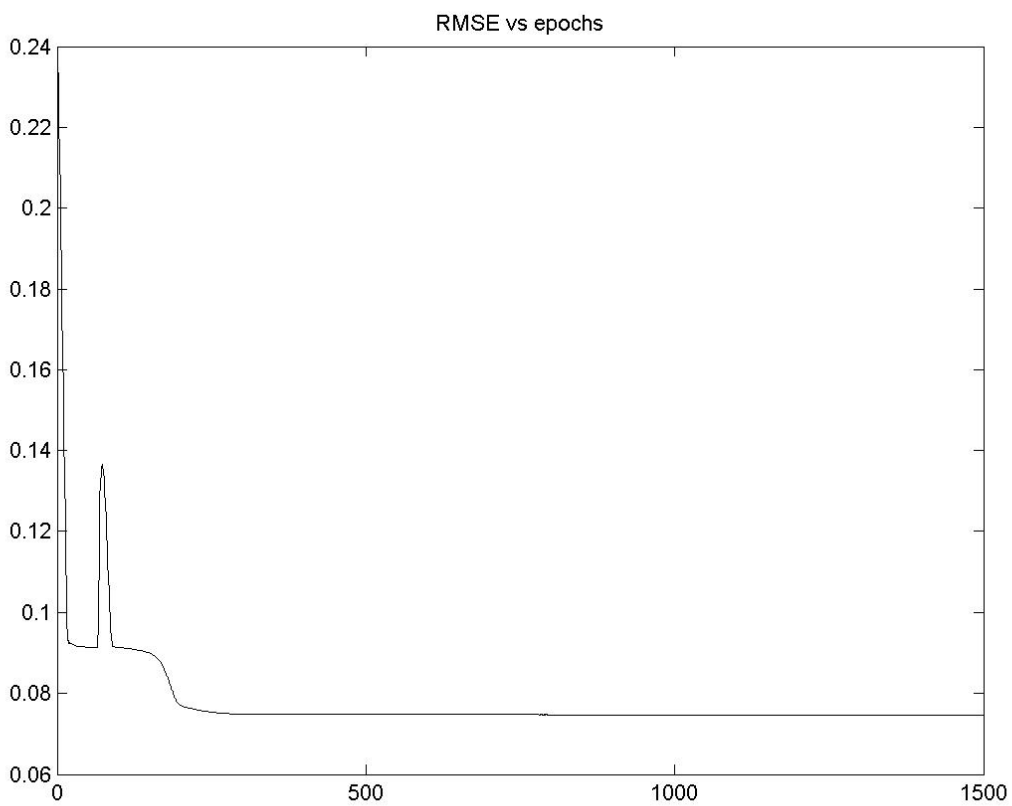
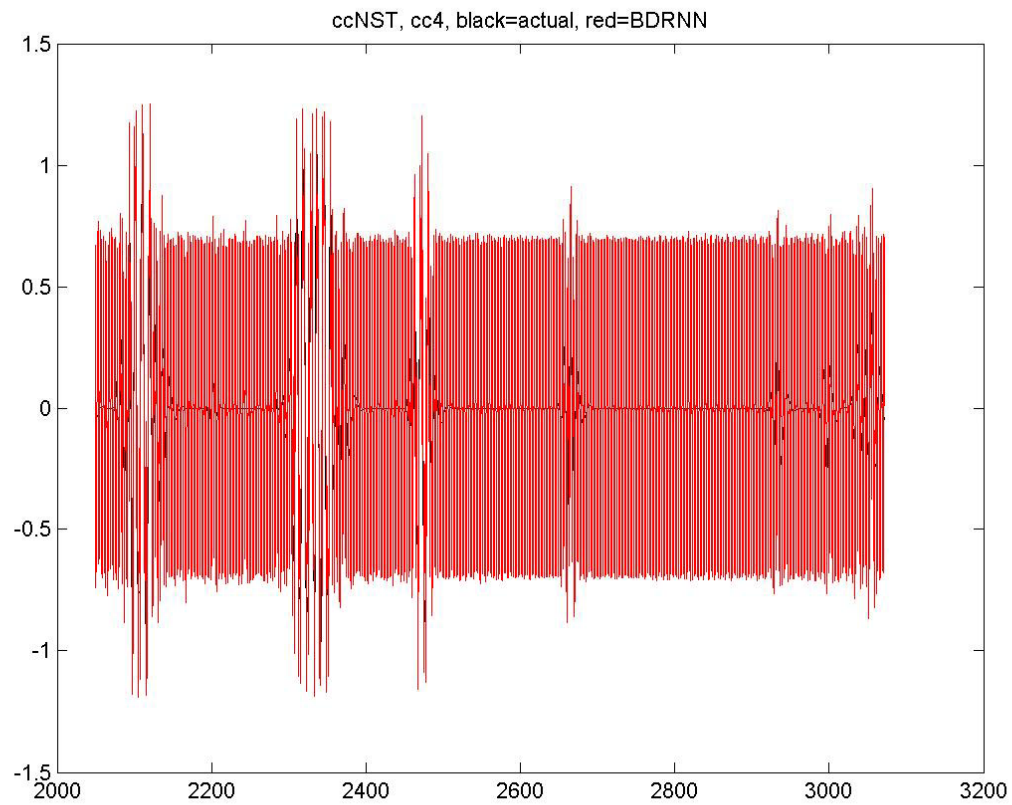
28^η Επανάληψη



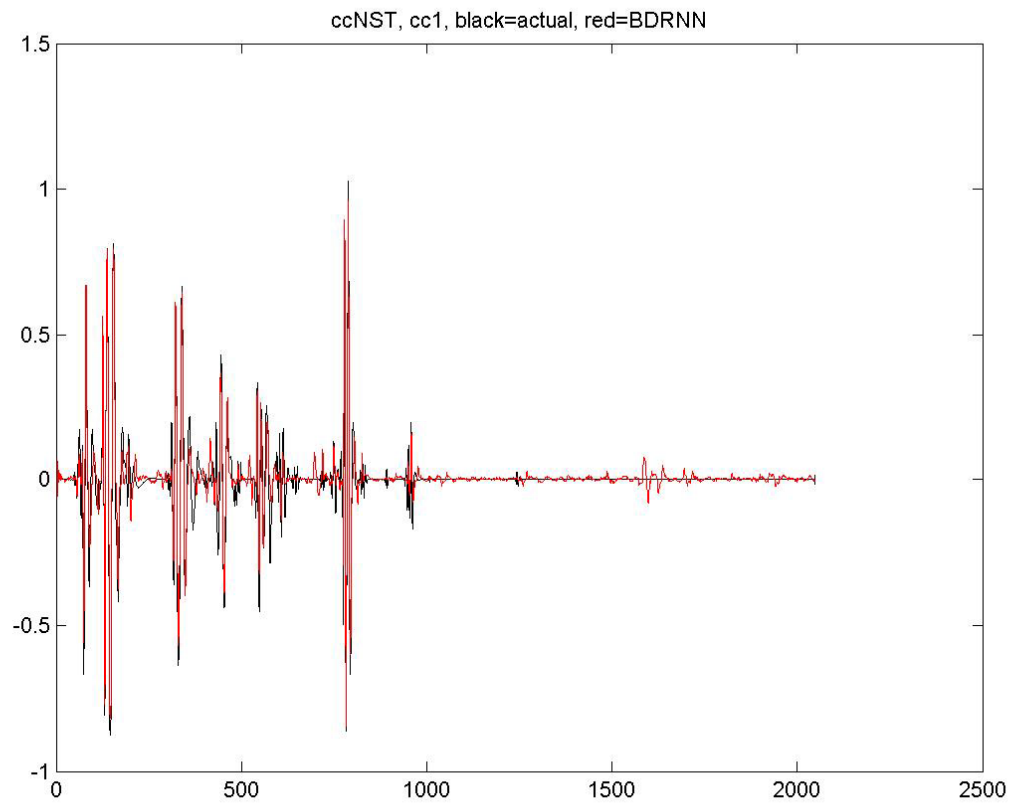
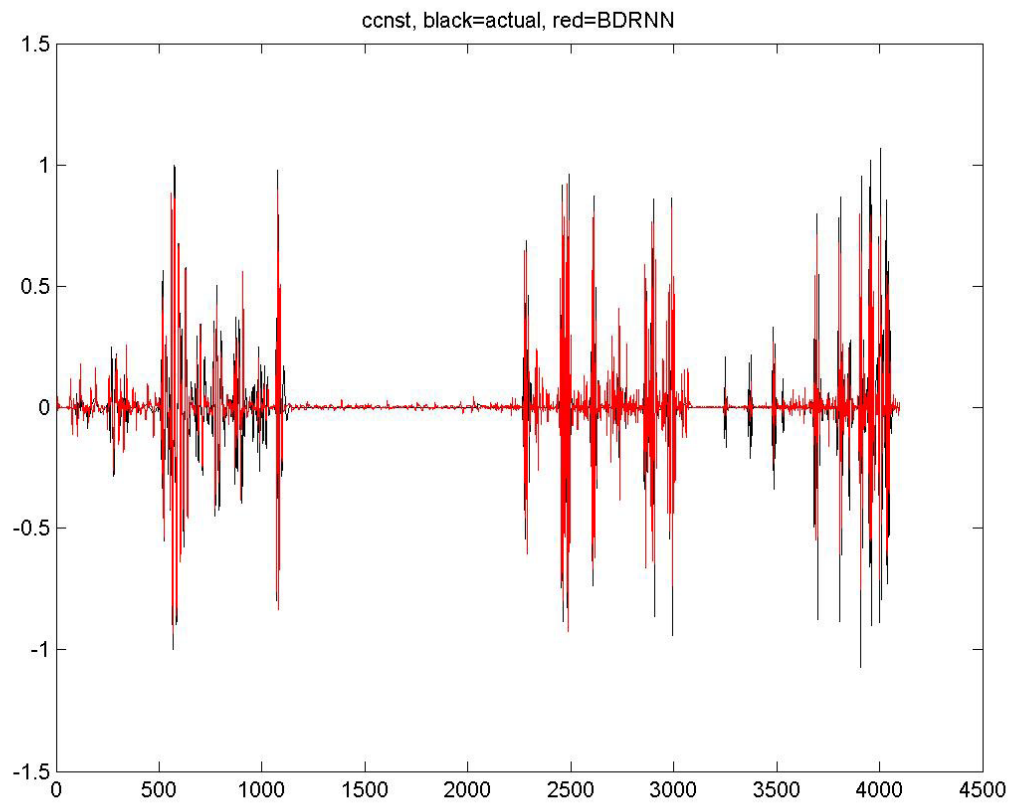


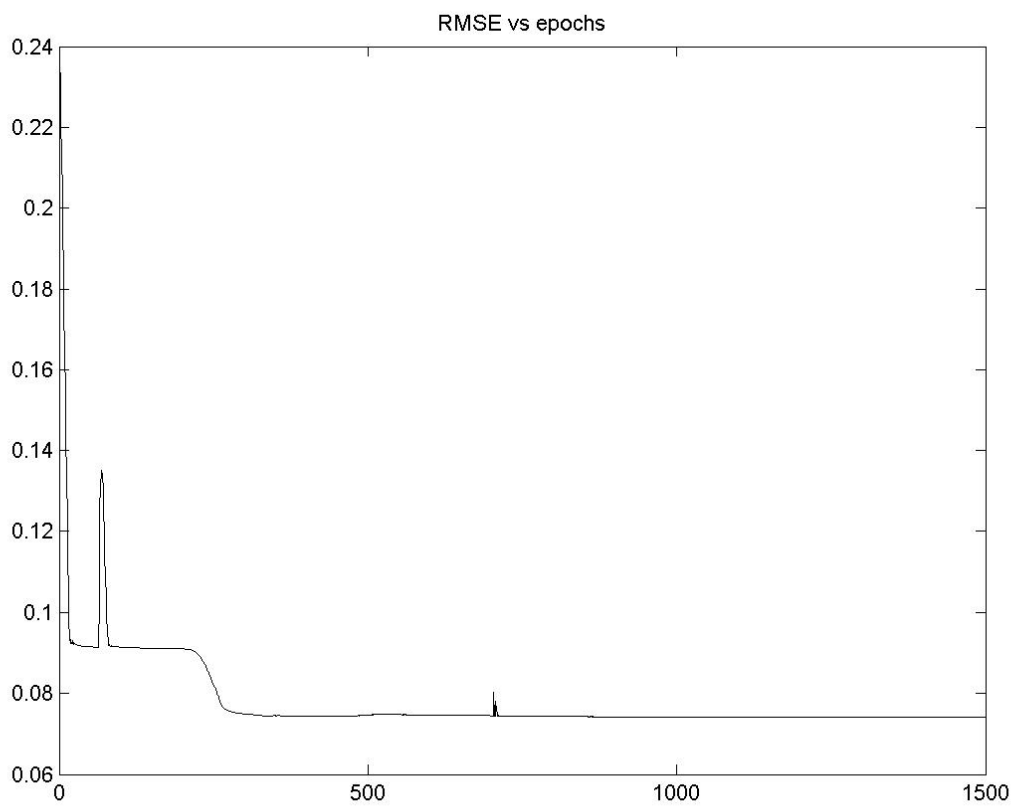
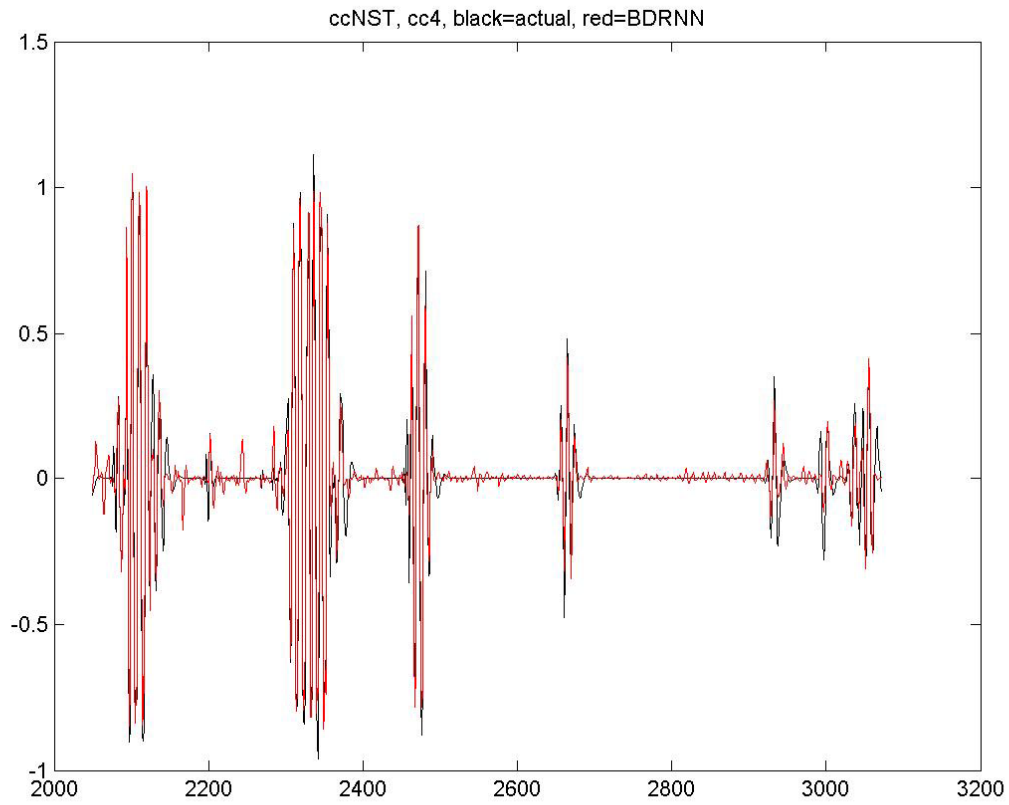
29^η Επανάληψη



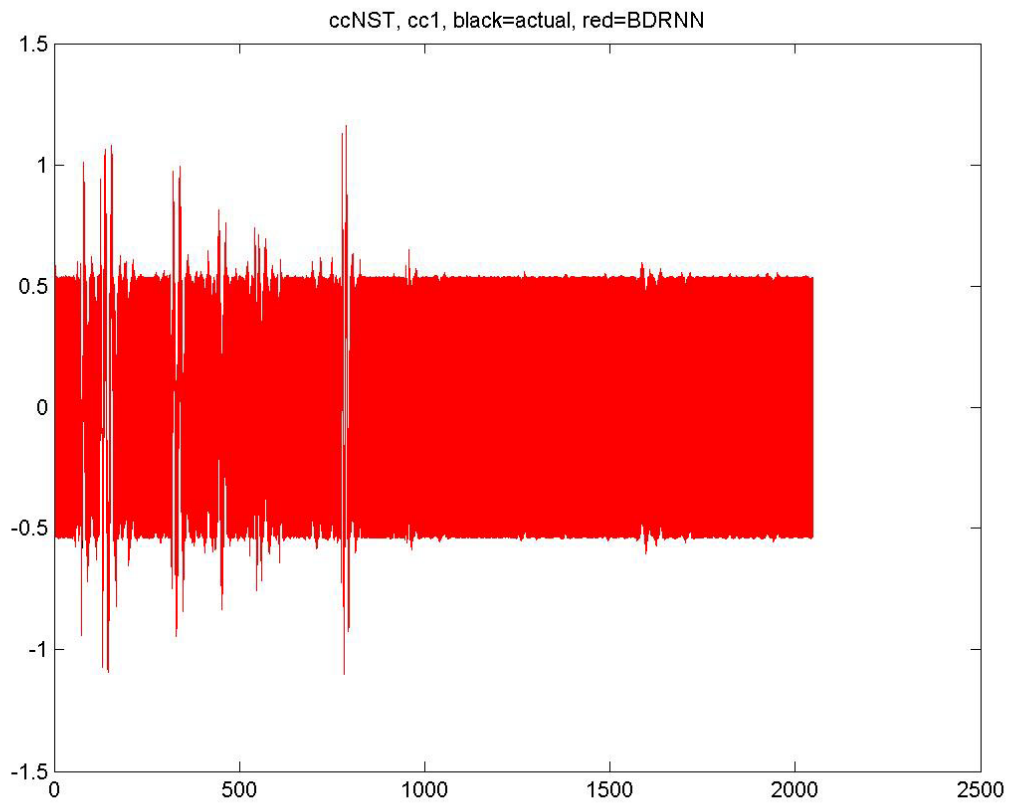
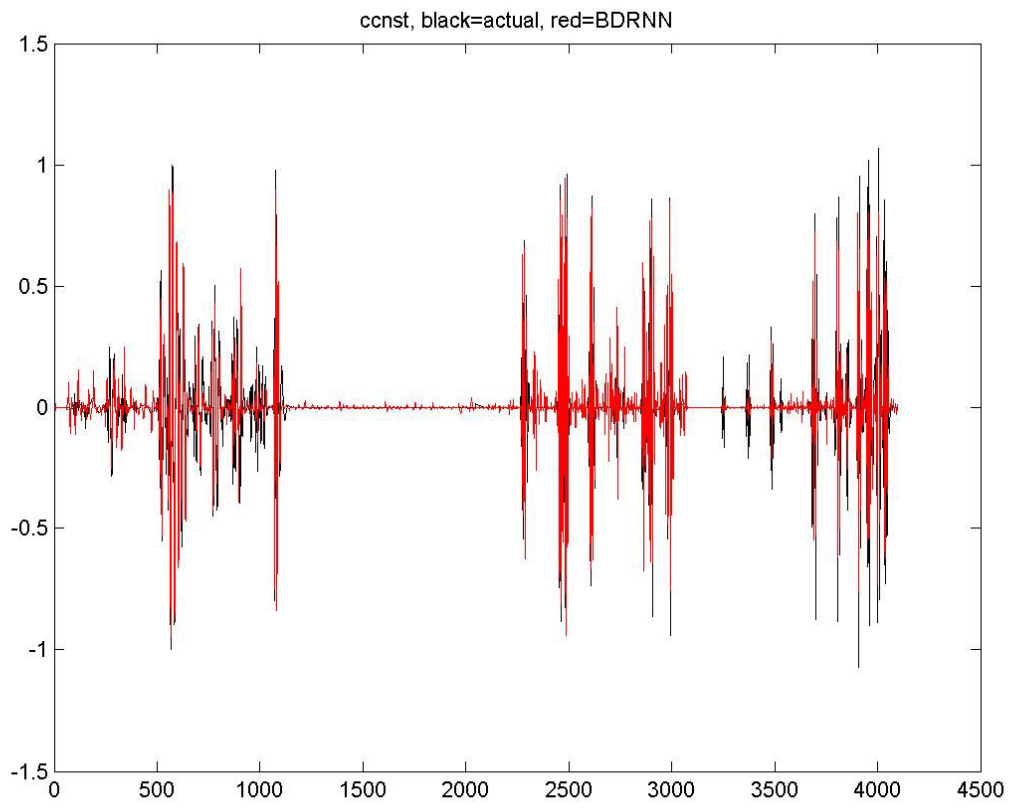


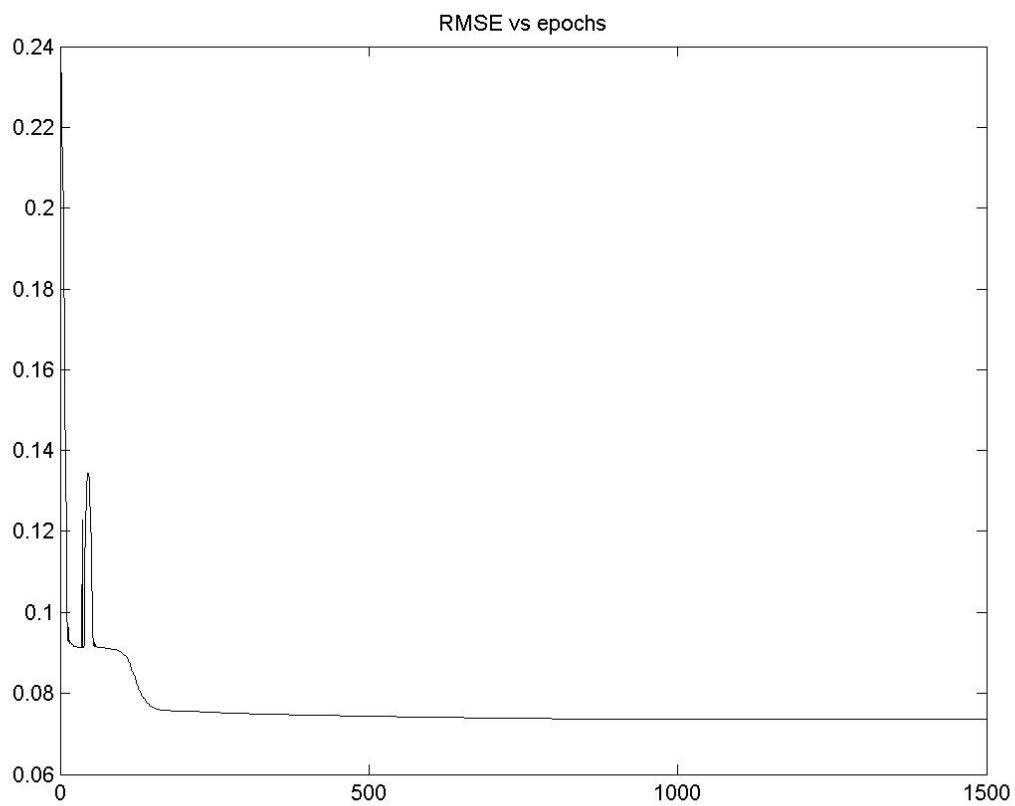
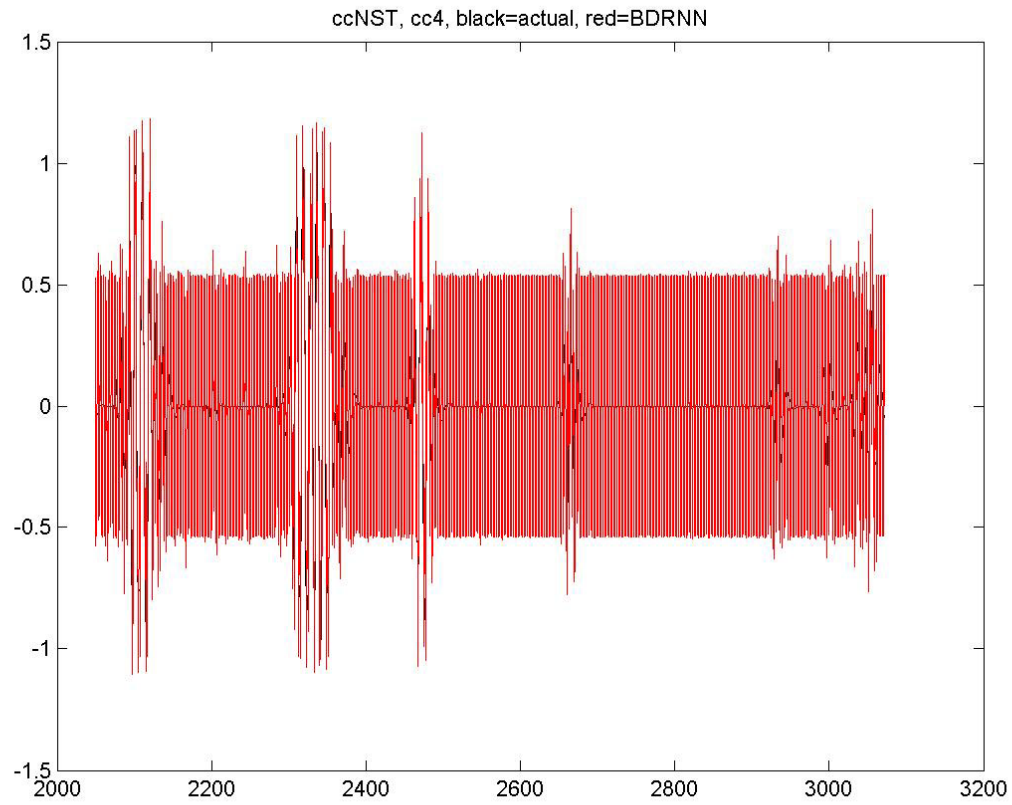
30^η Επανάληψη



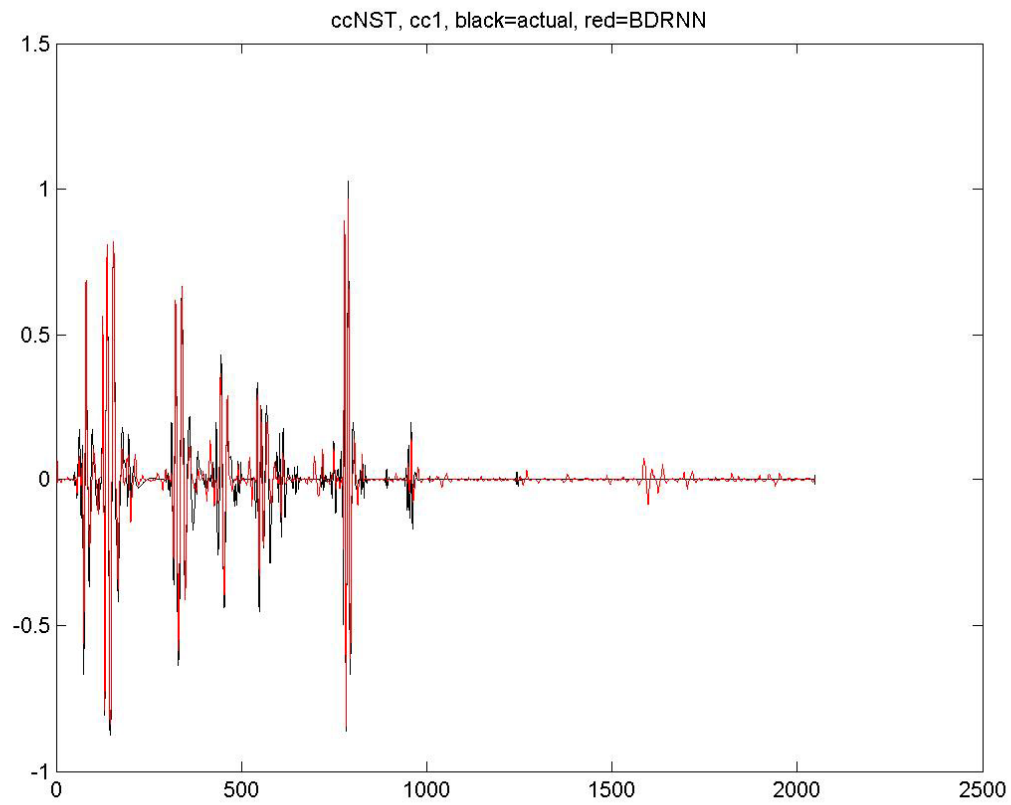
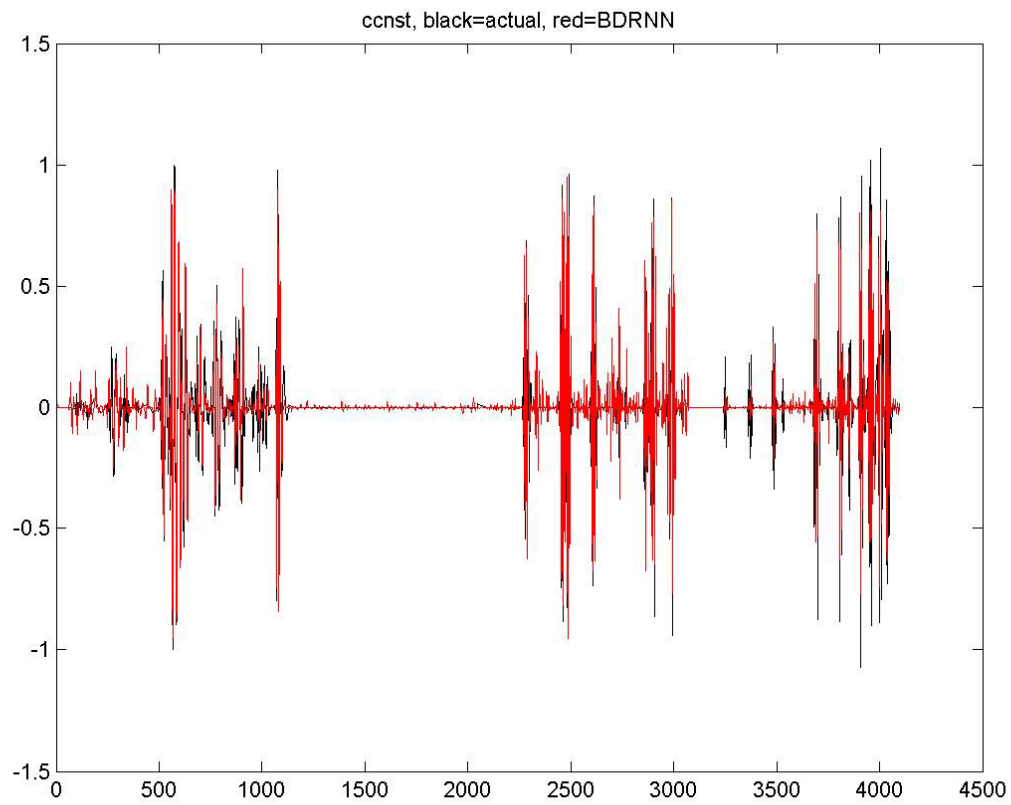


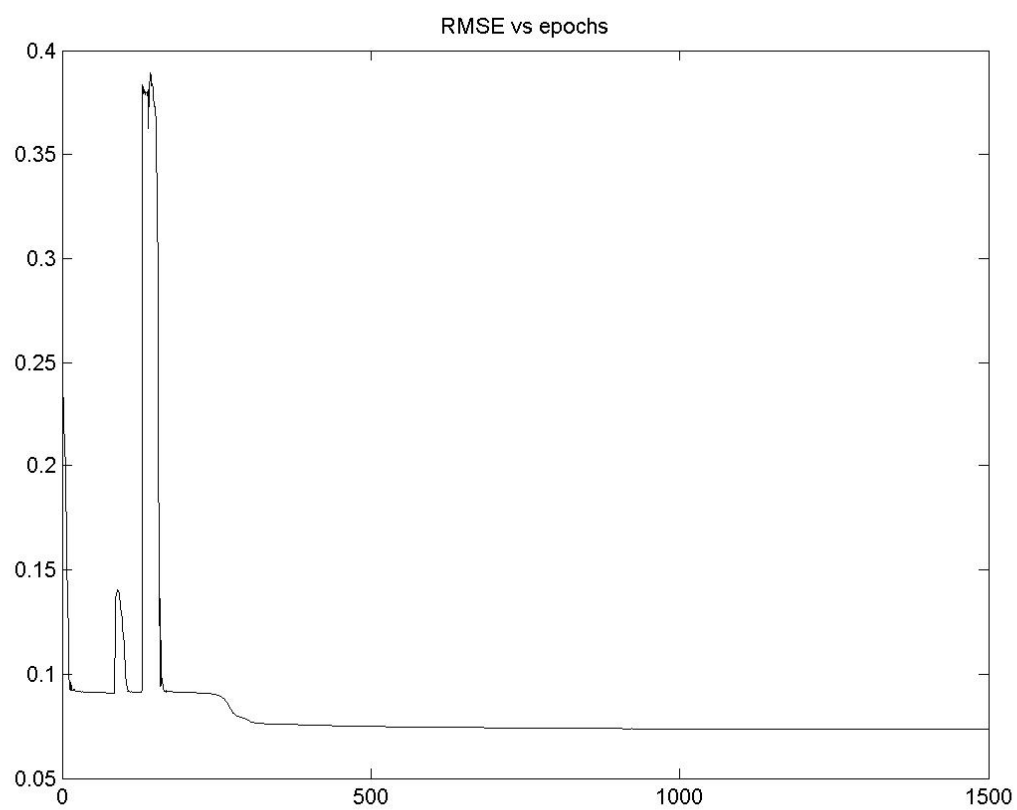
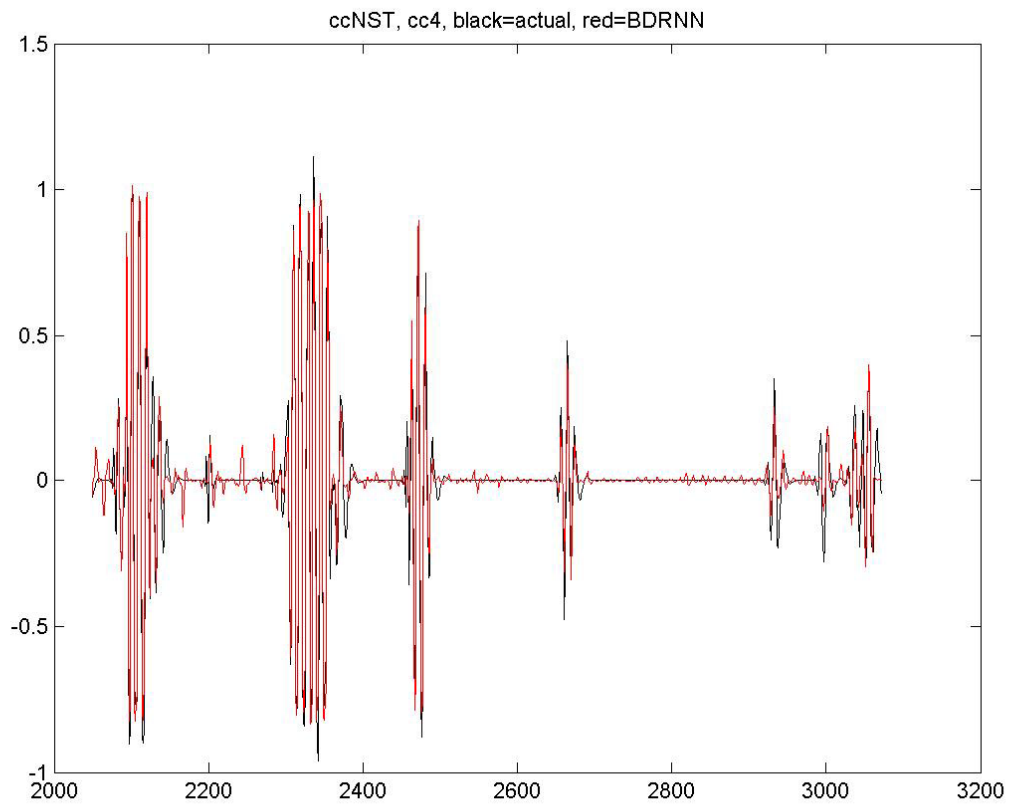
31^η Επανάληψη



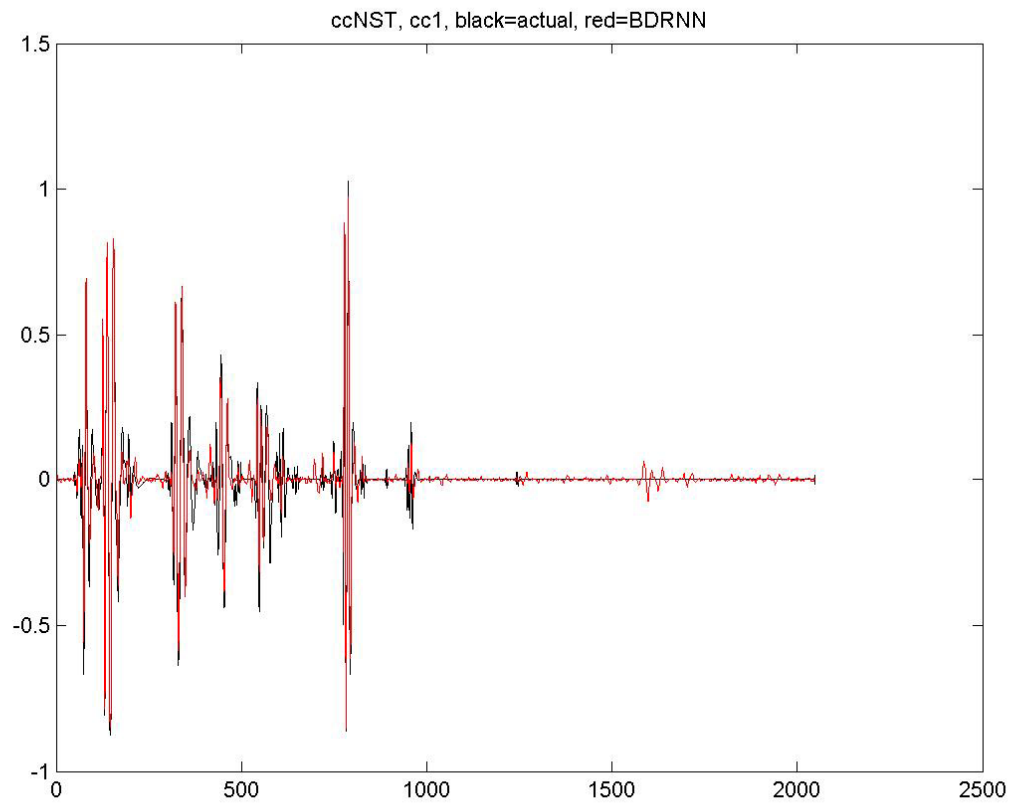
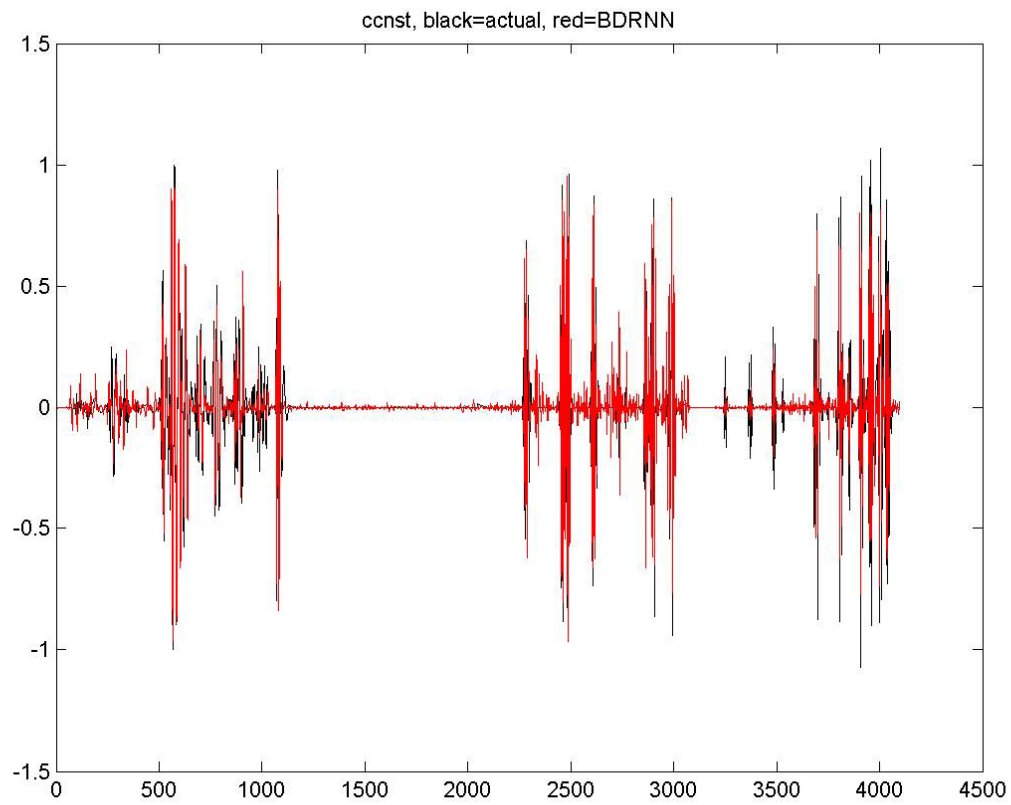


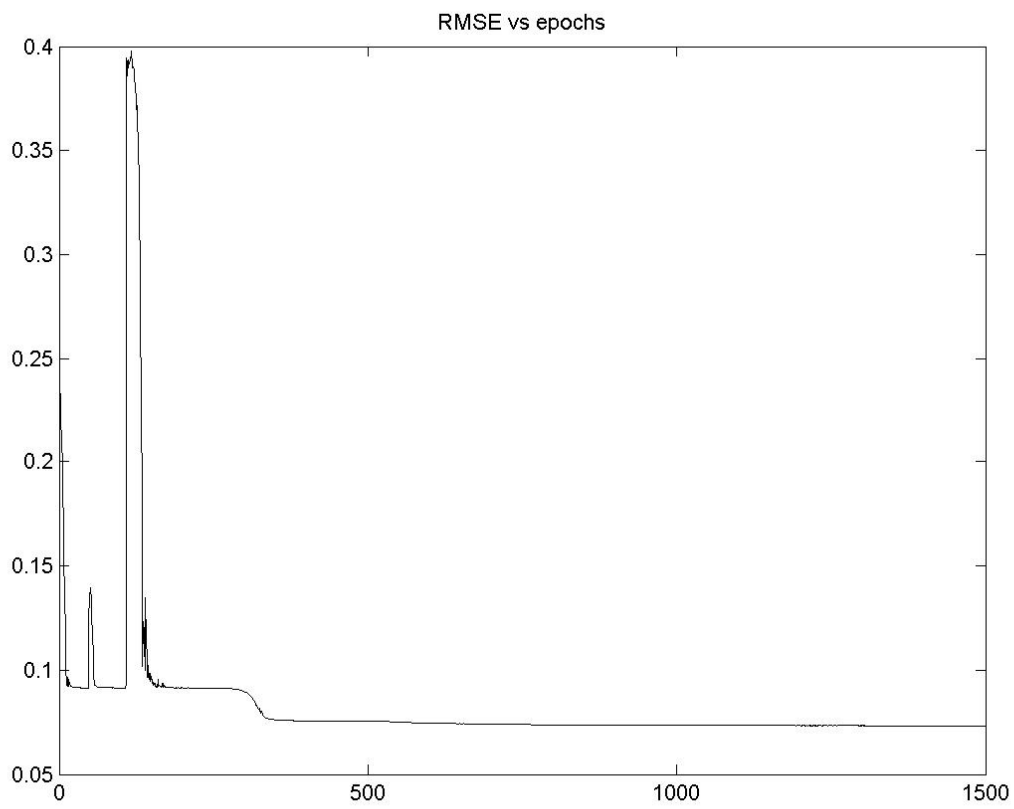
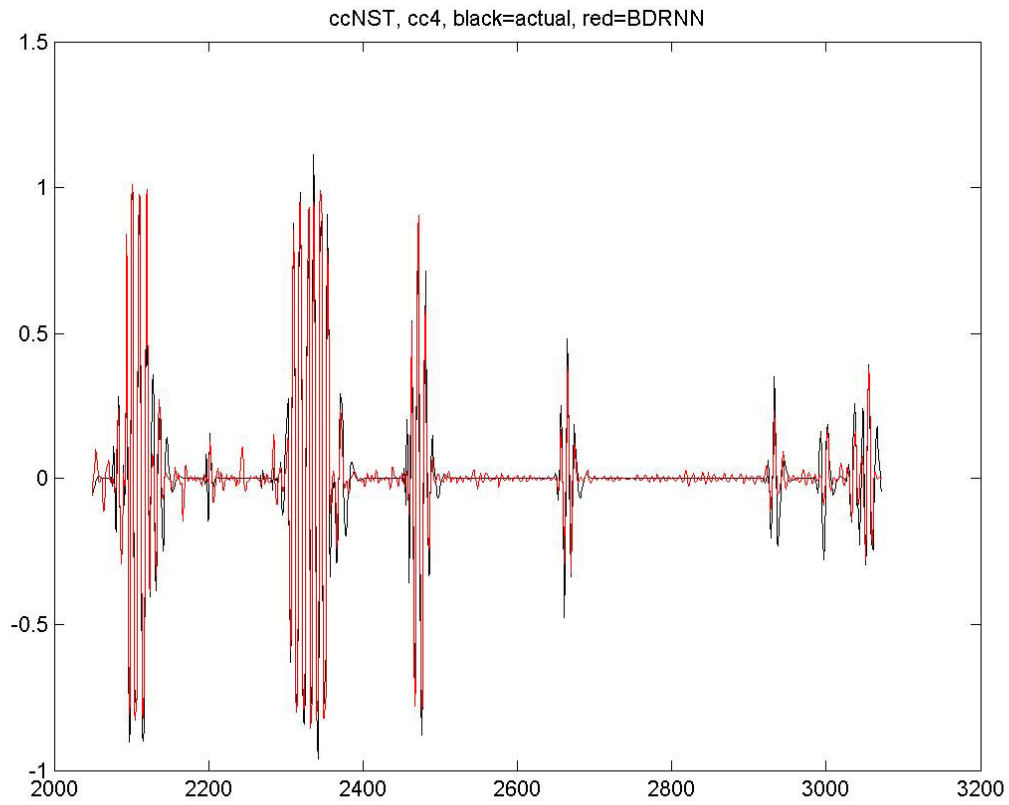
32^η Επανάληψη



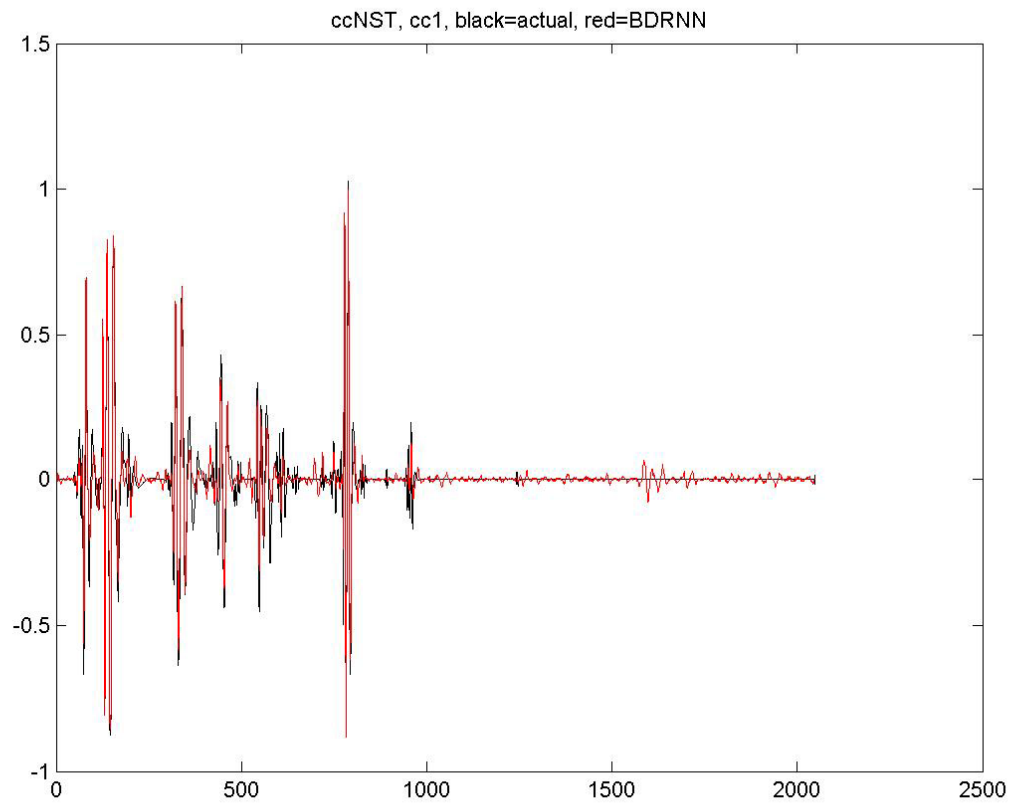
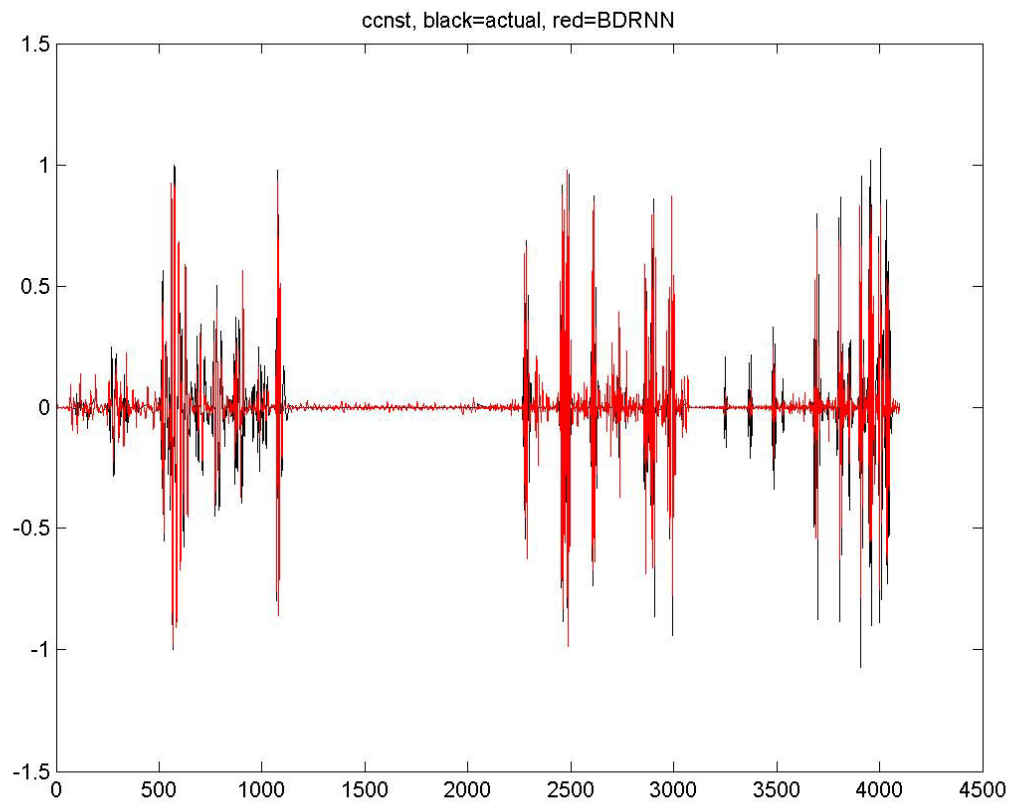


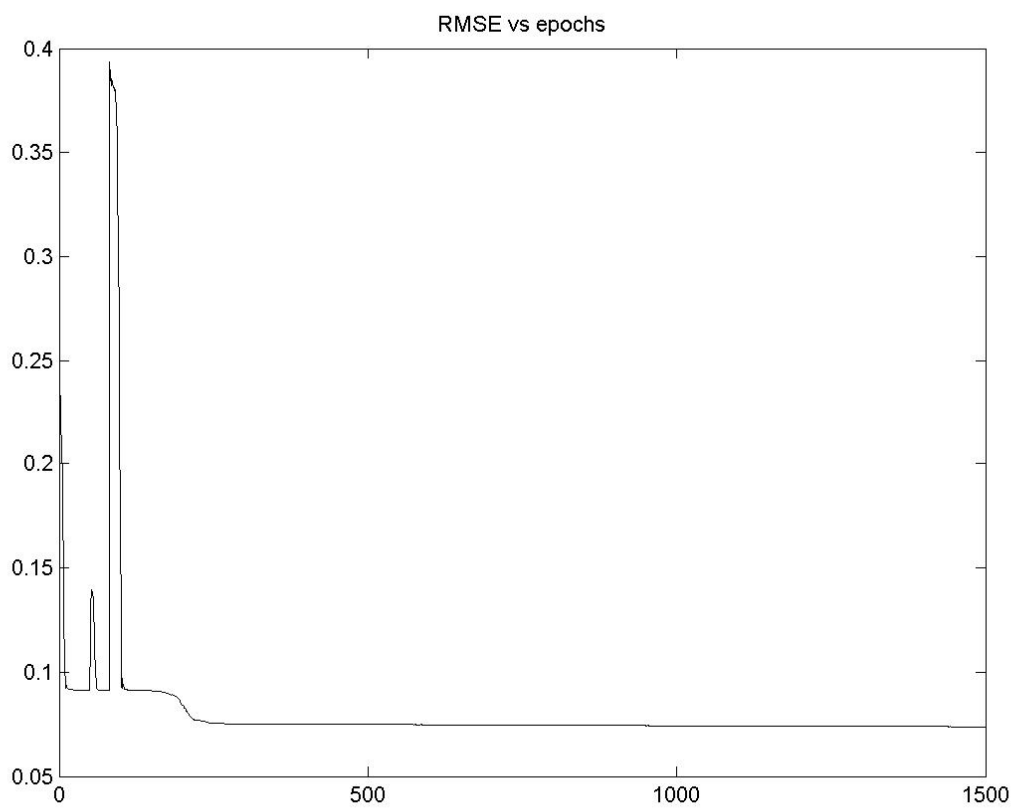
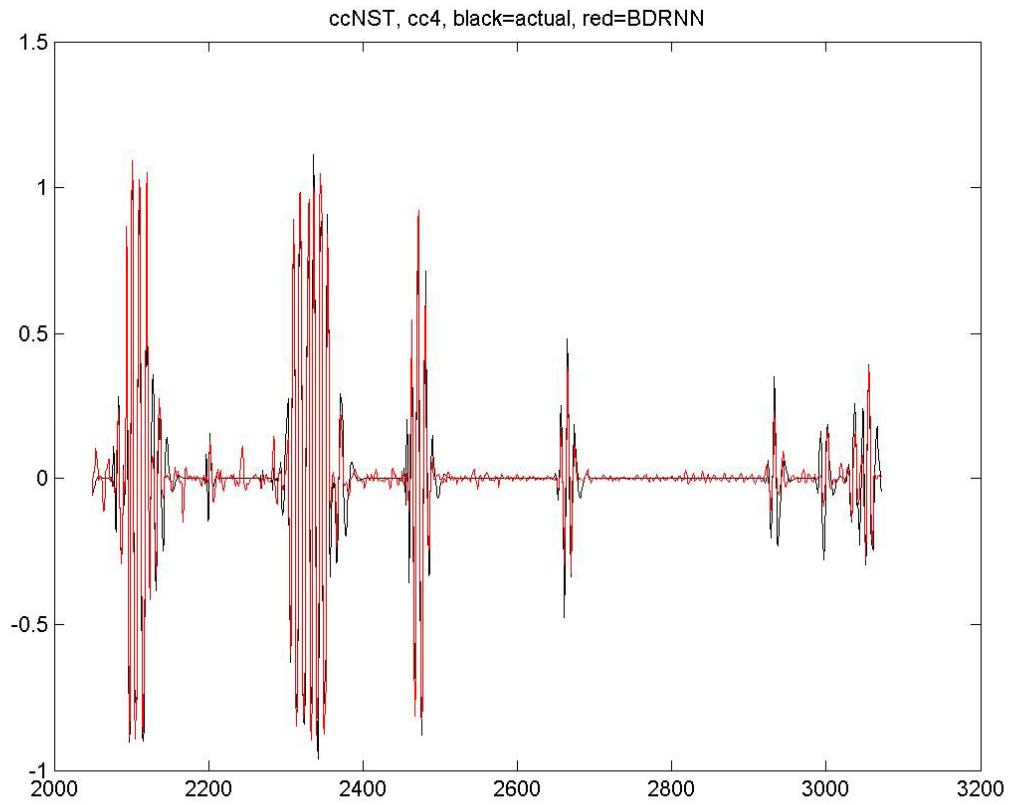
33^η Επανάληψη



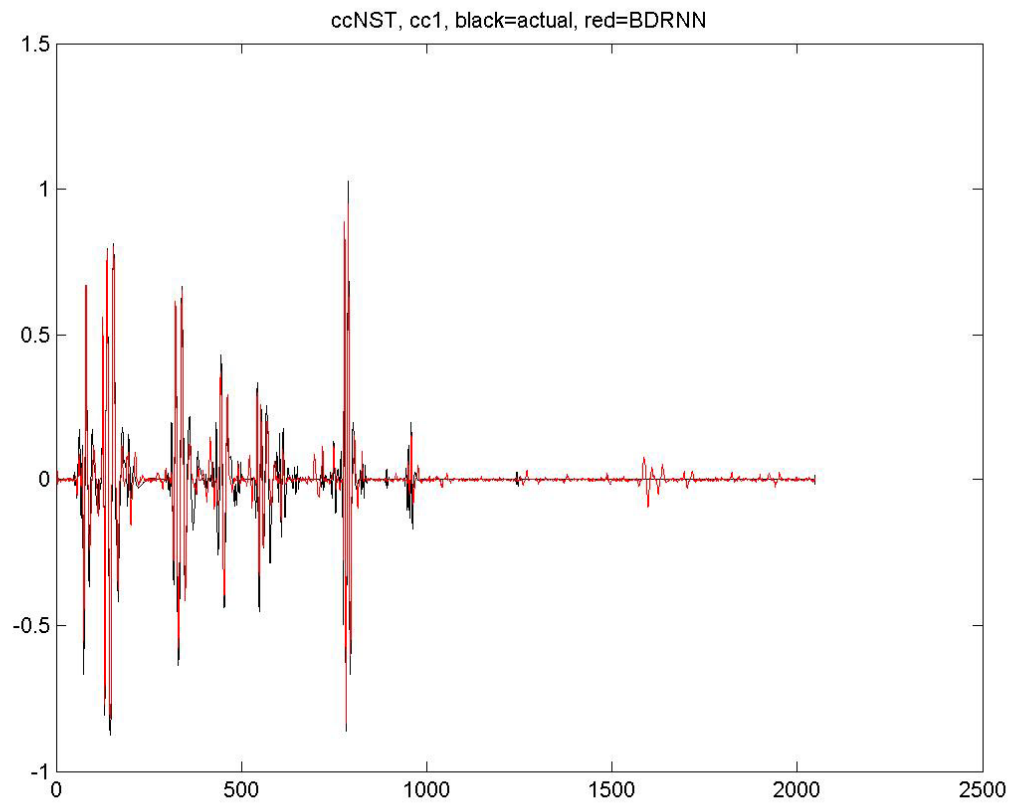
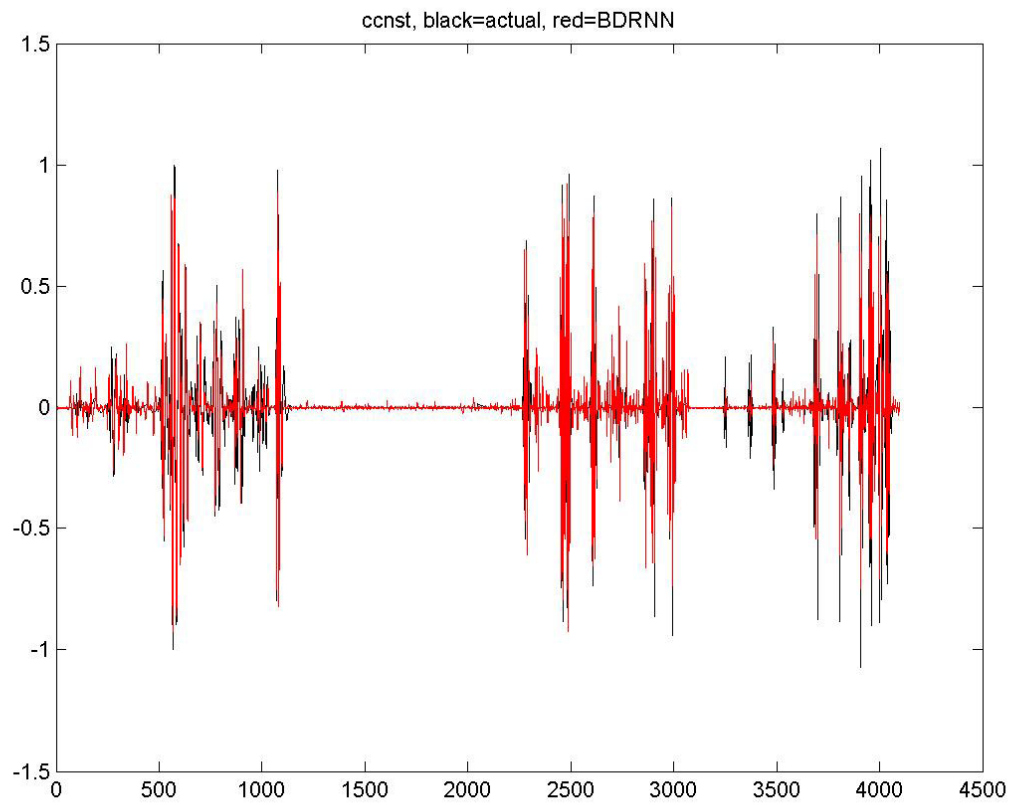


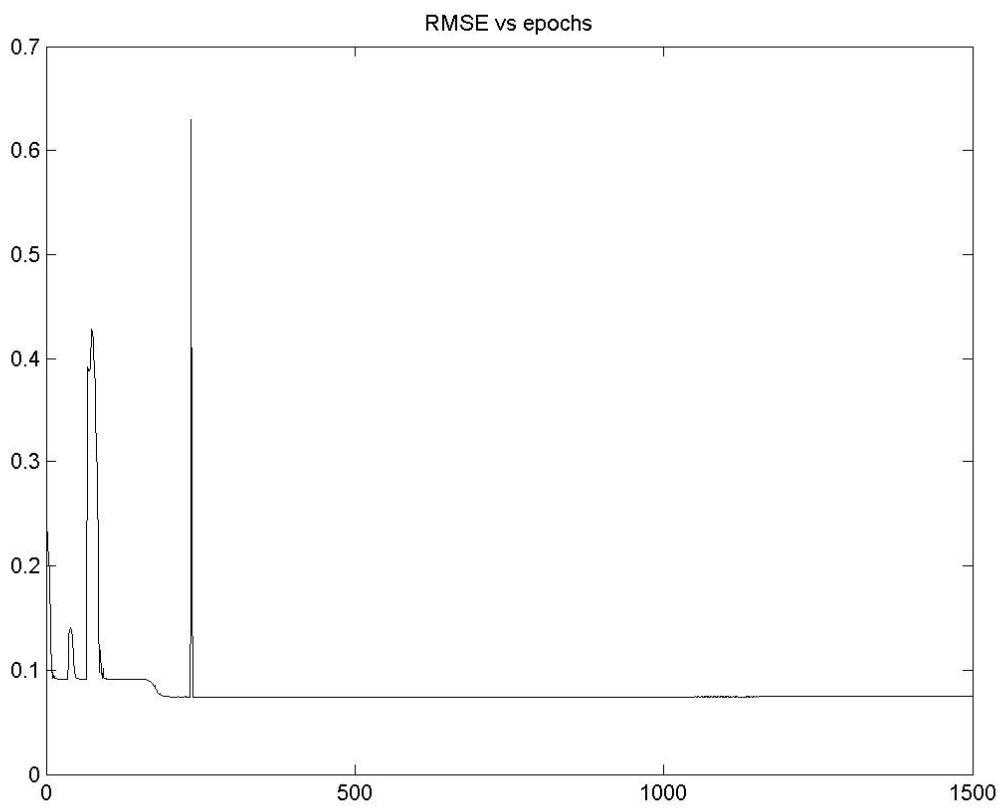
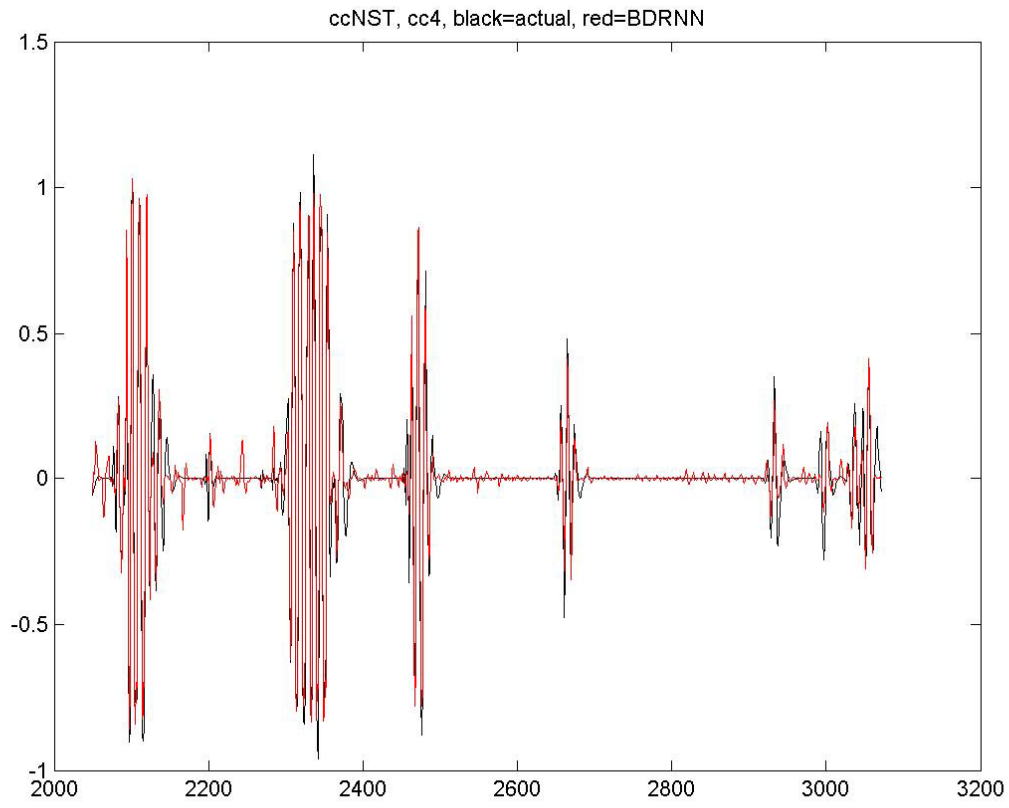
34^η Επανάληψη



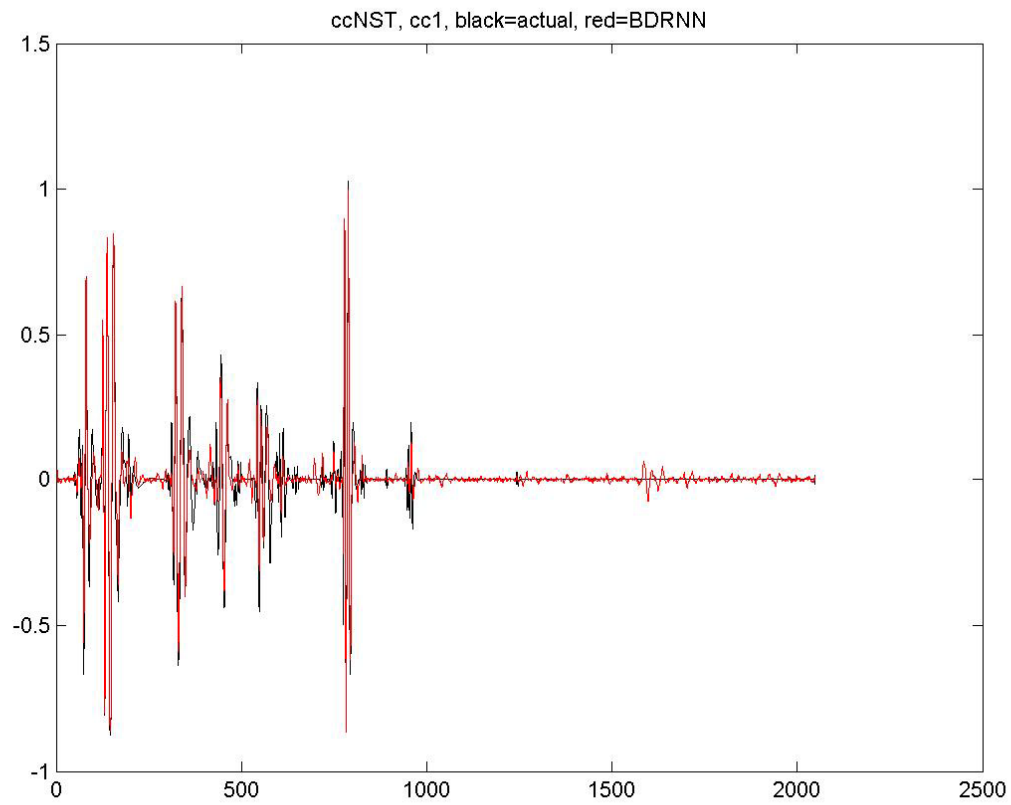
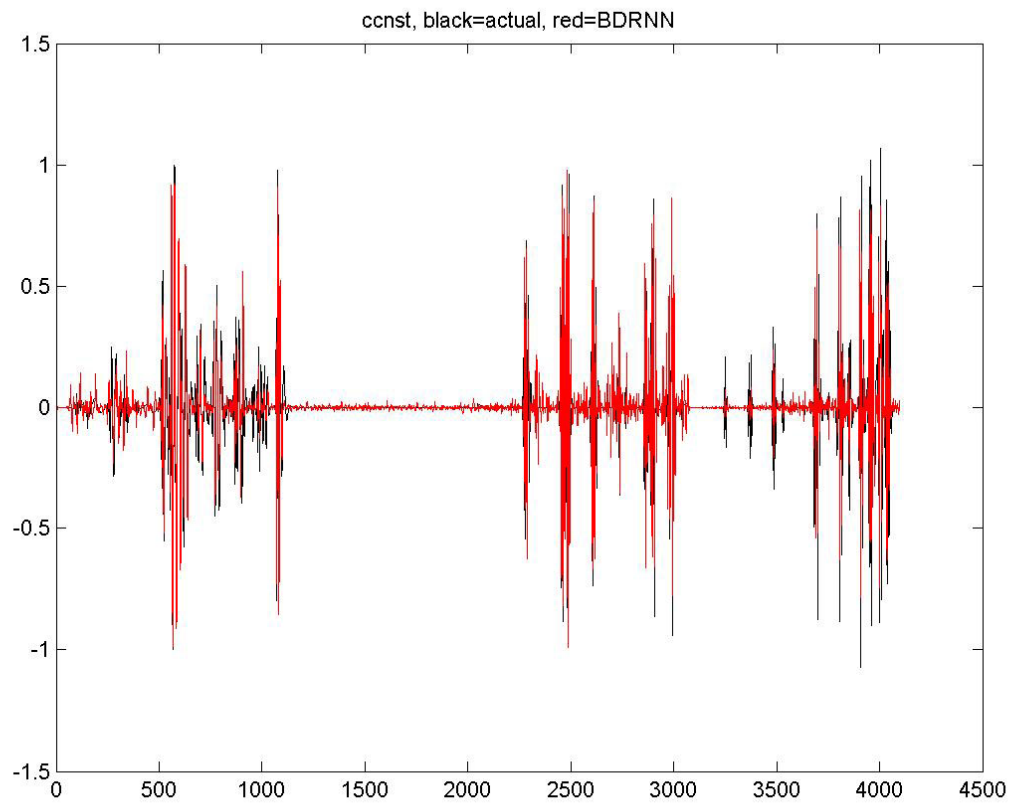


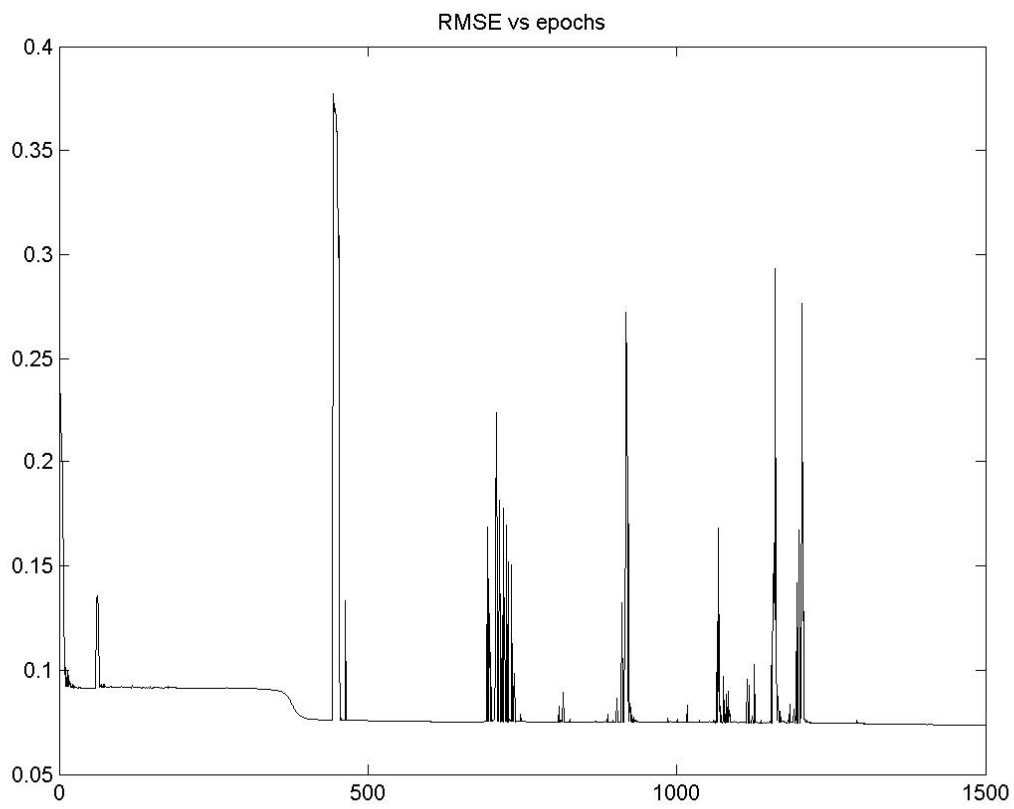
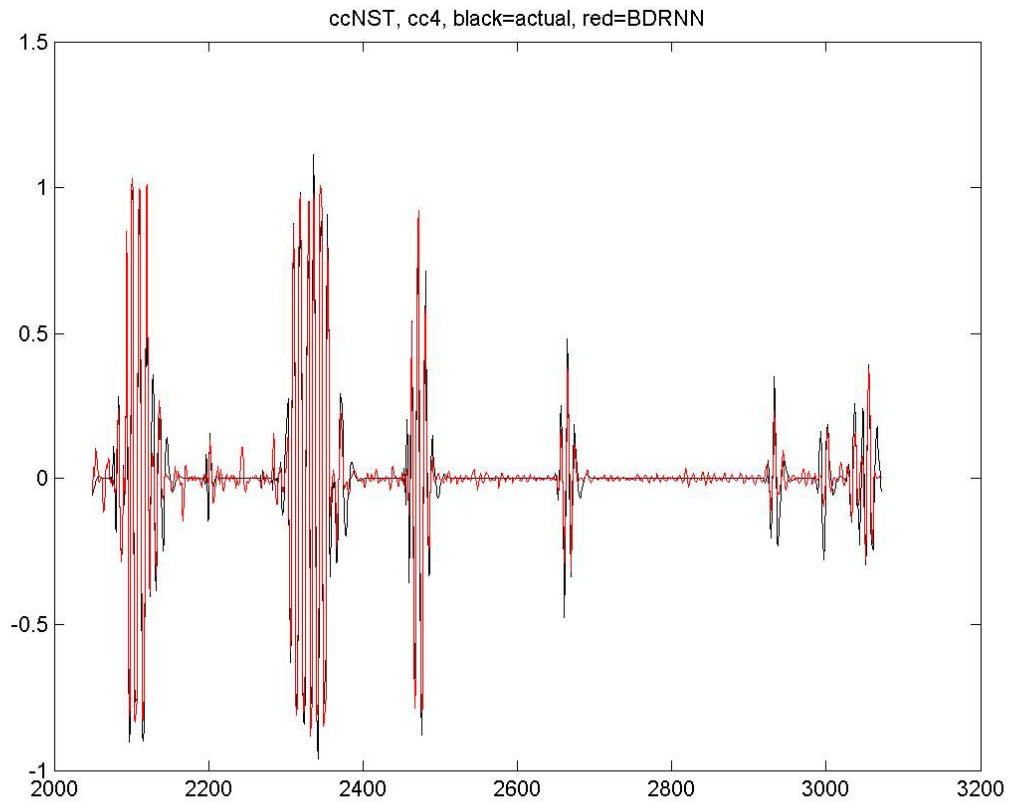
35^η Επανάληψη



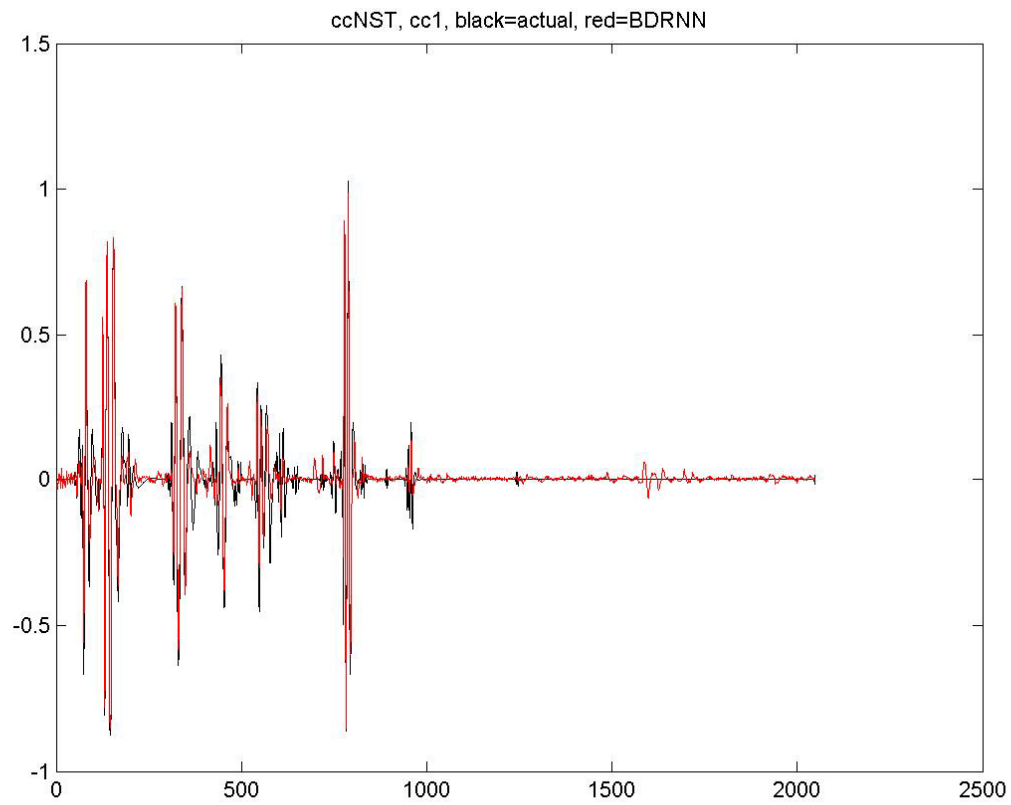
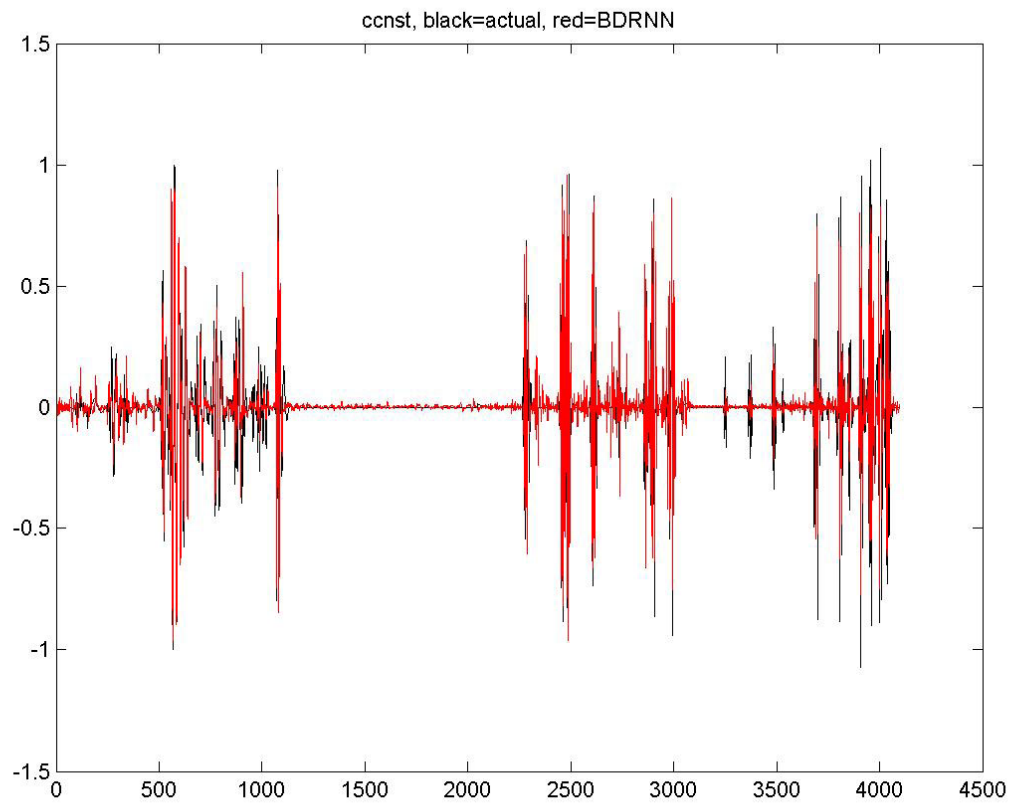


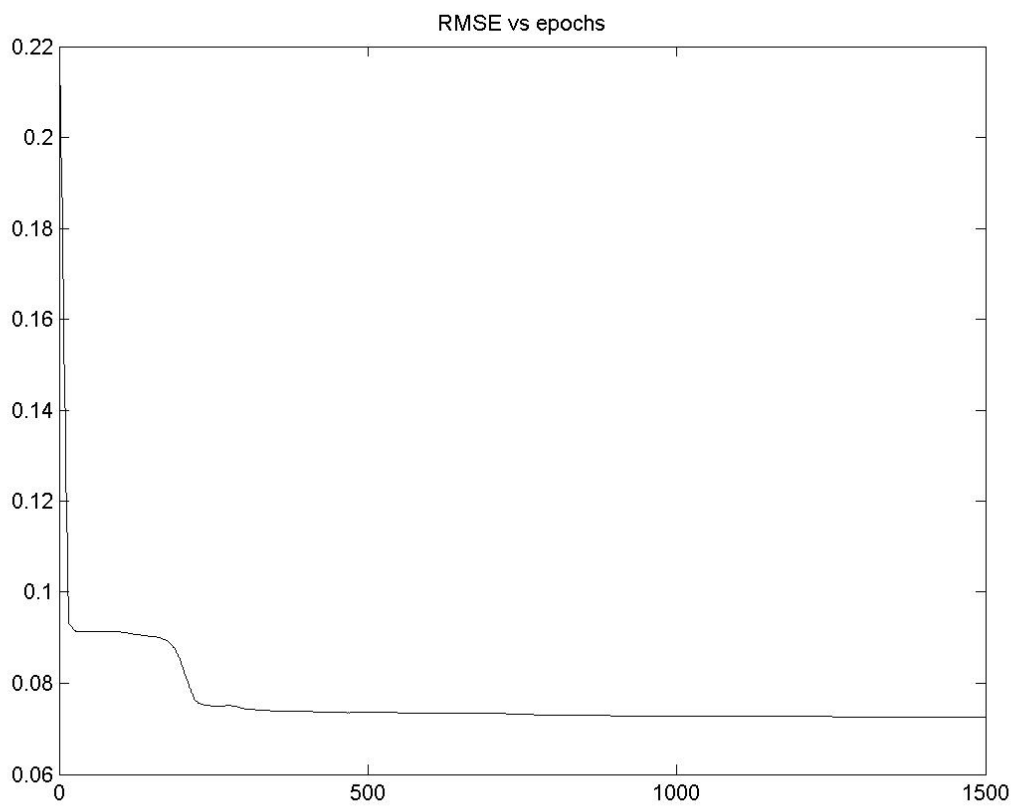
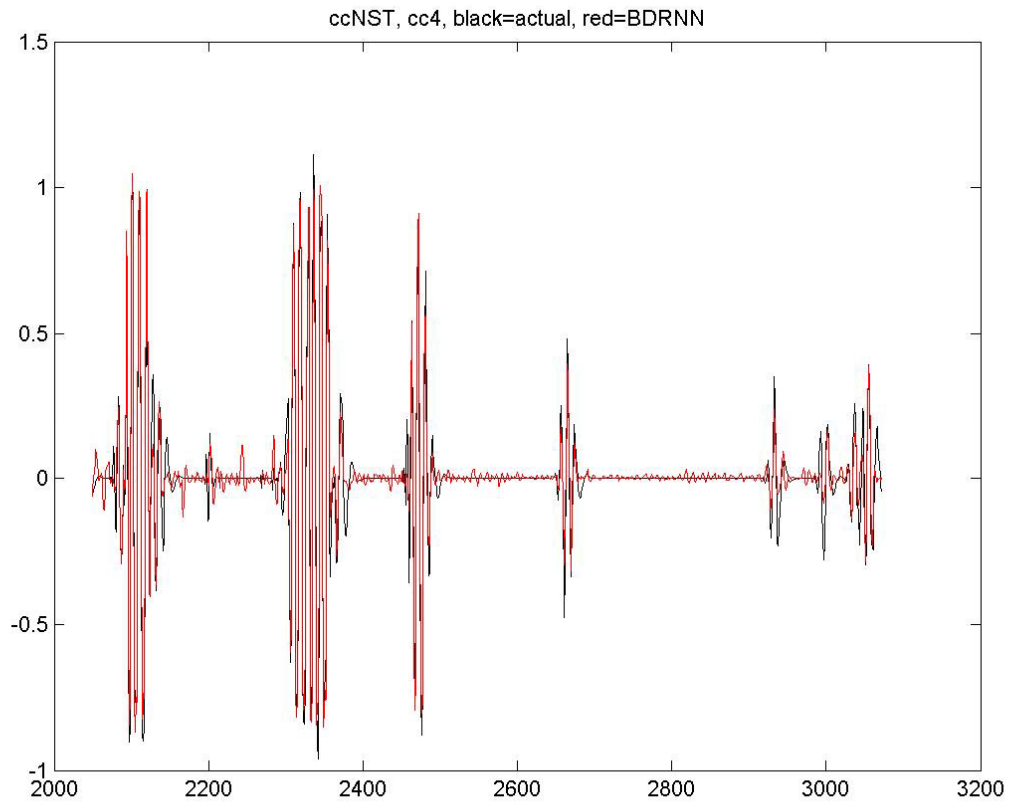
36^η Επανάληψη



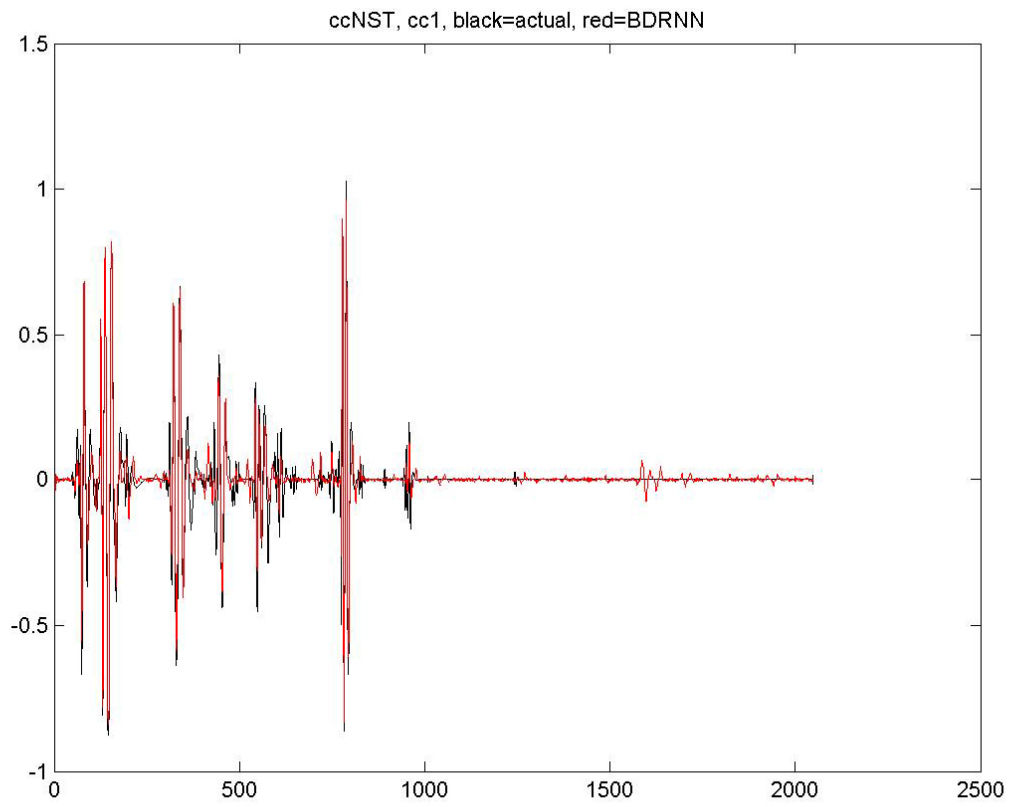
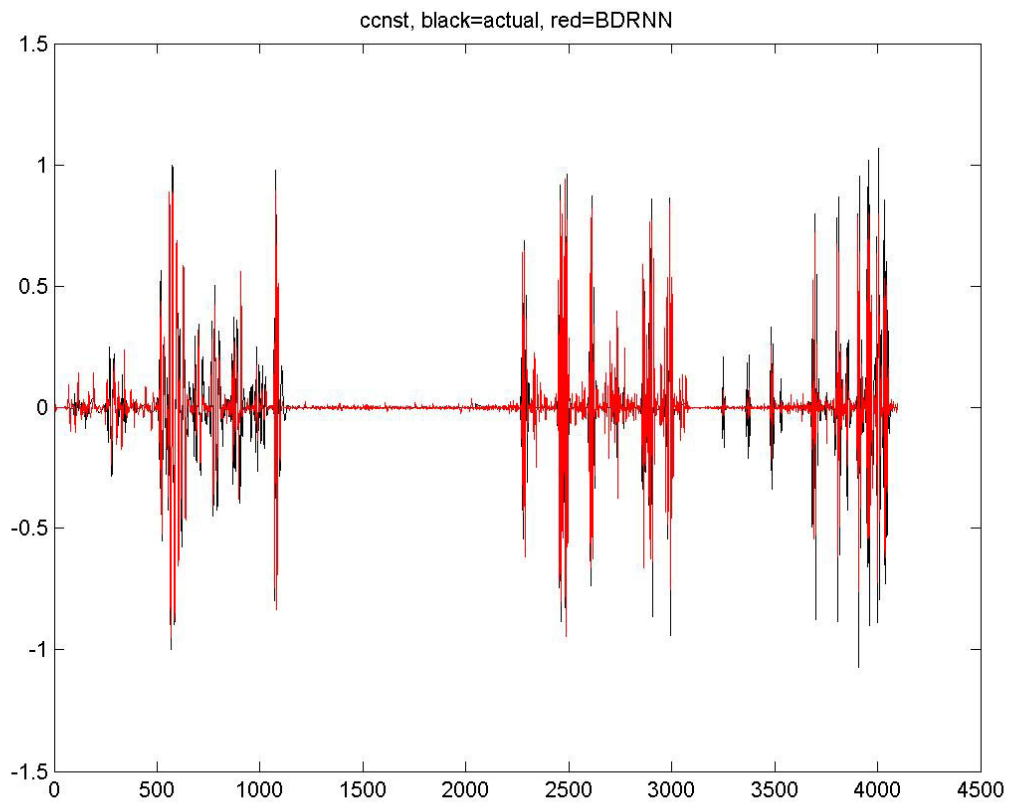


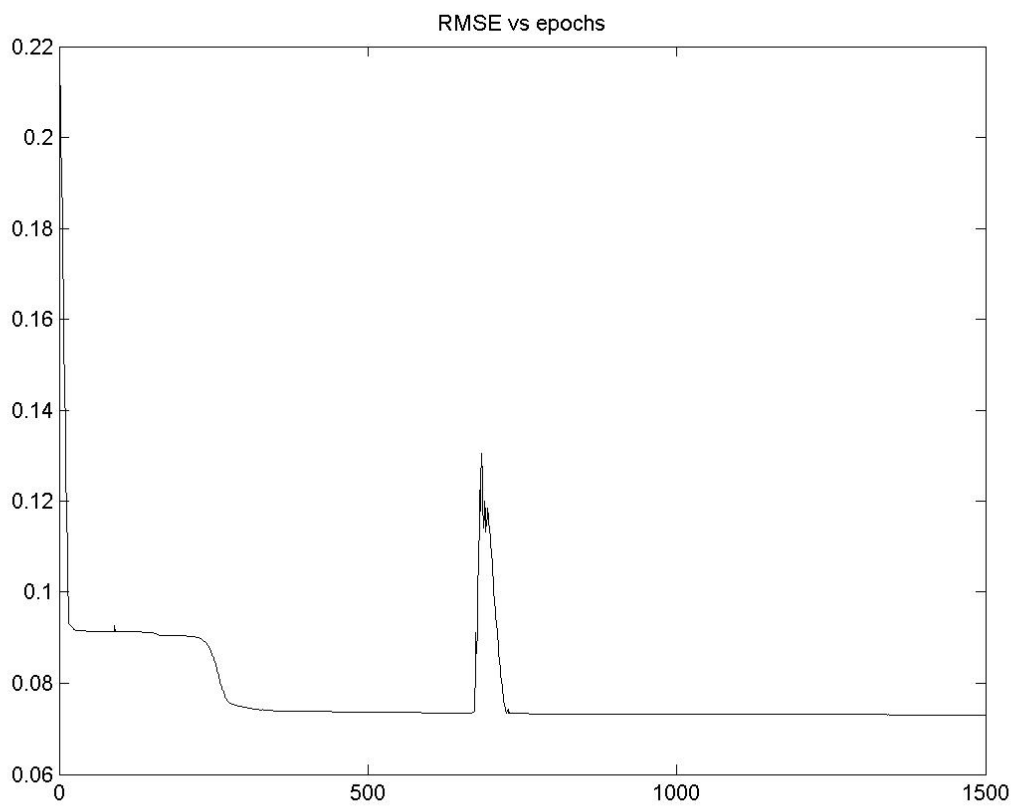
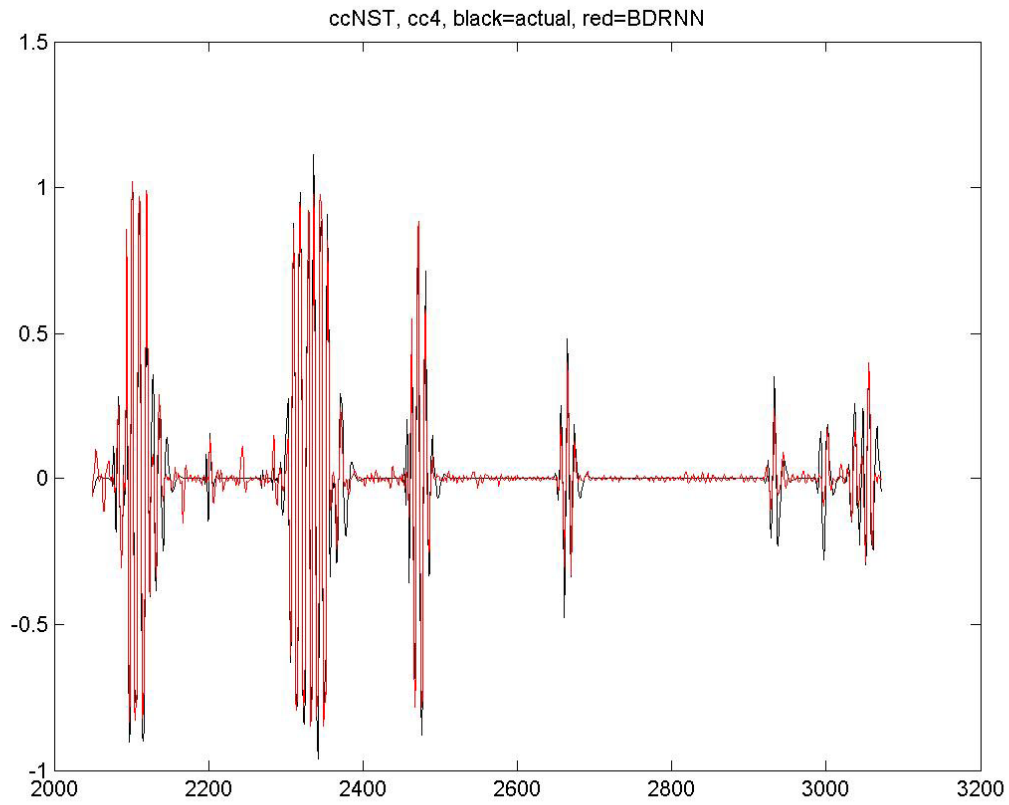
37^η Επανάληψη



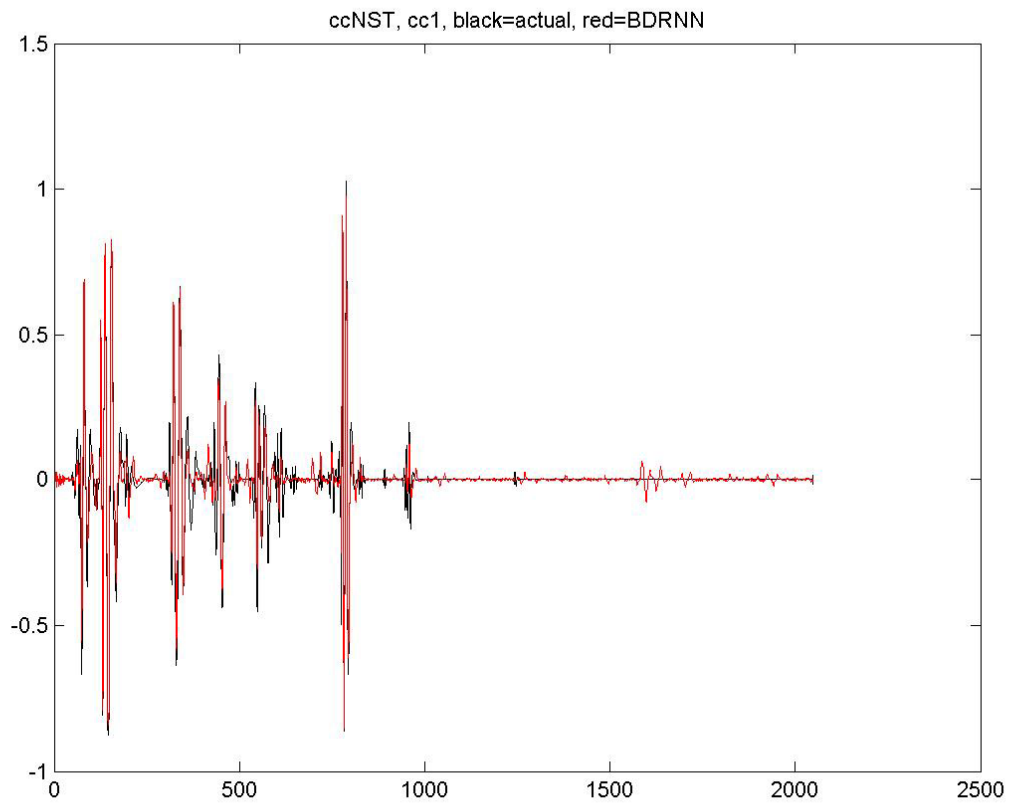
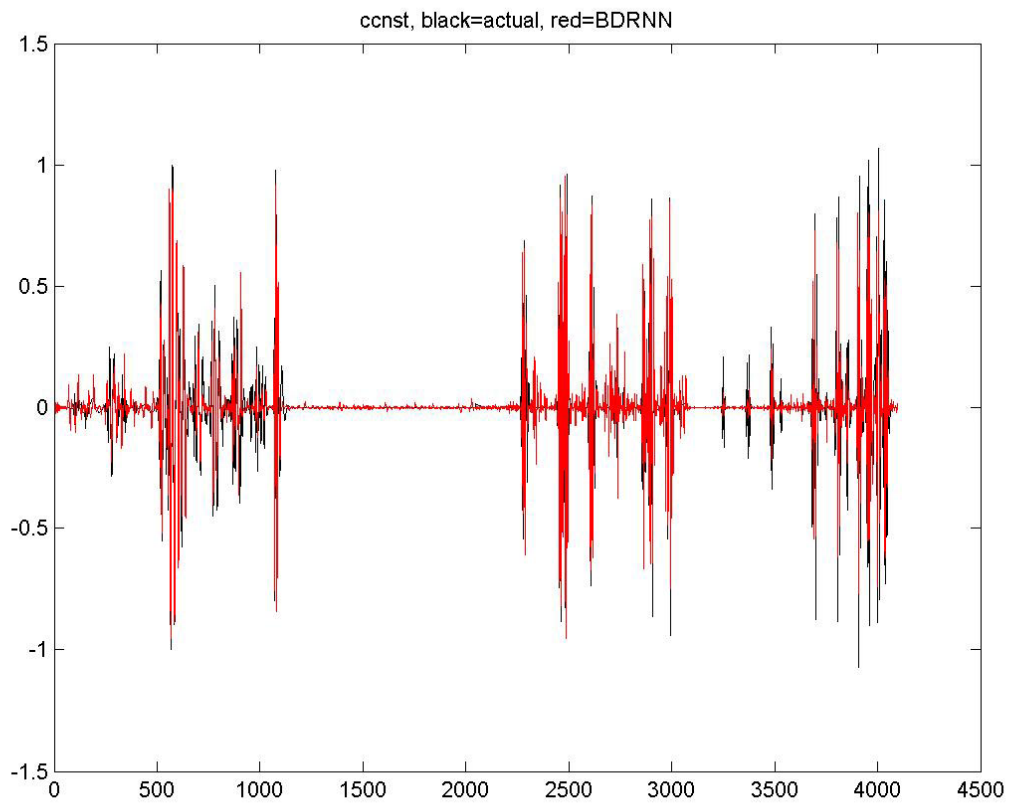


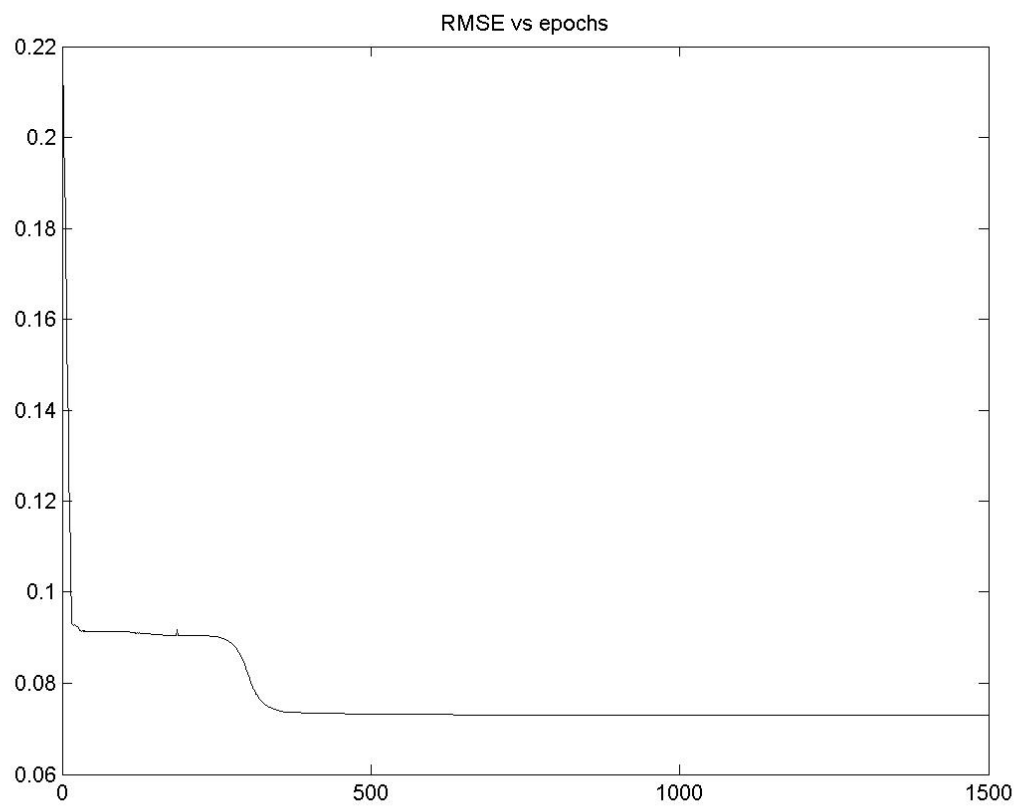
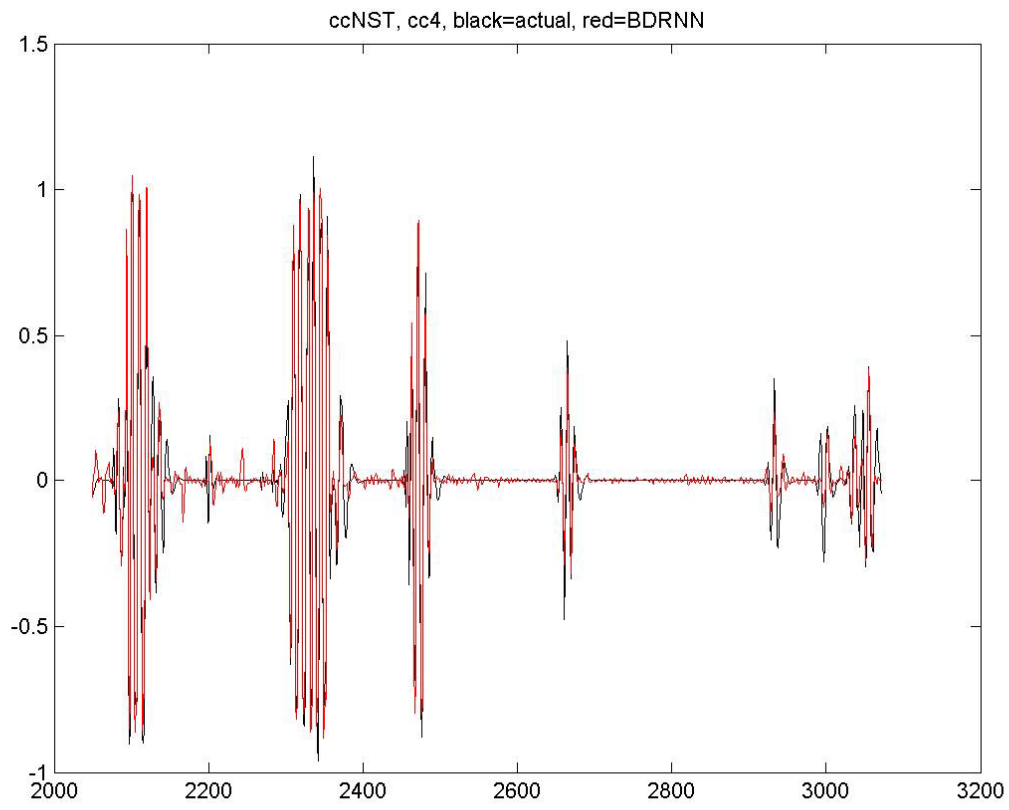
38^η Επανάληψη



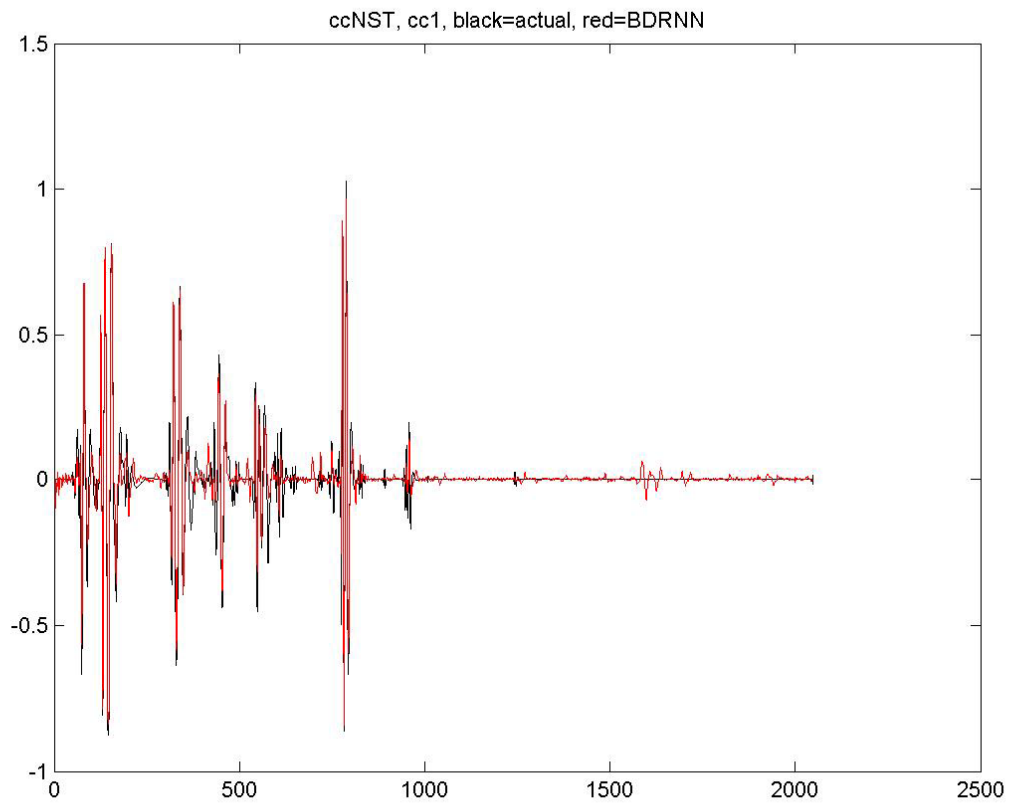
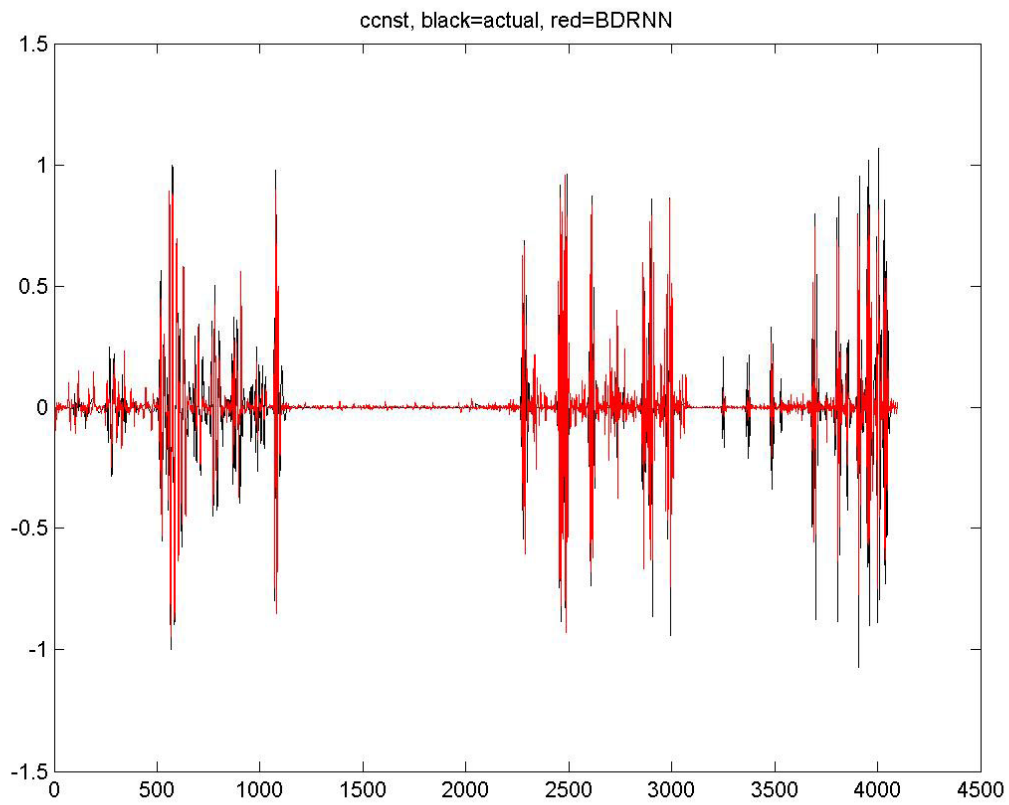


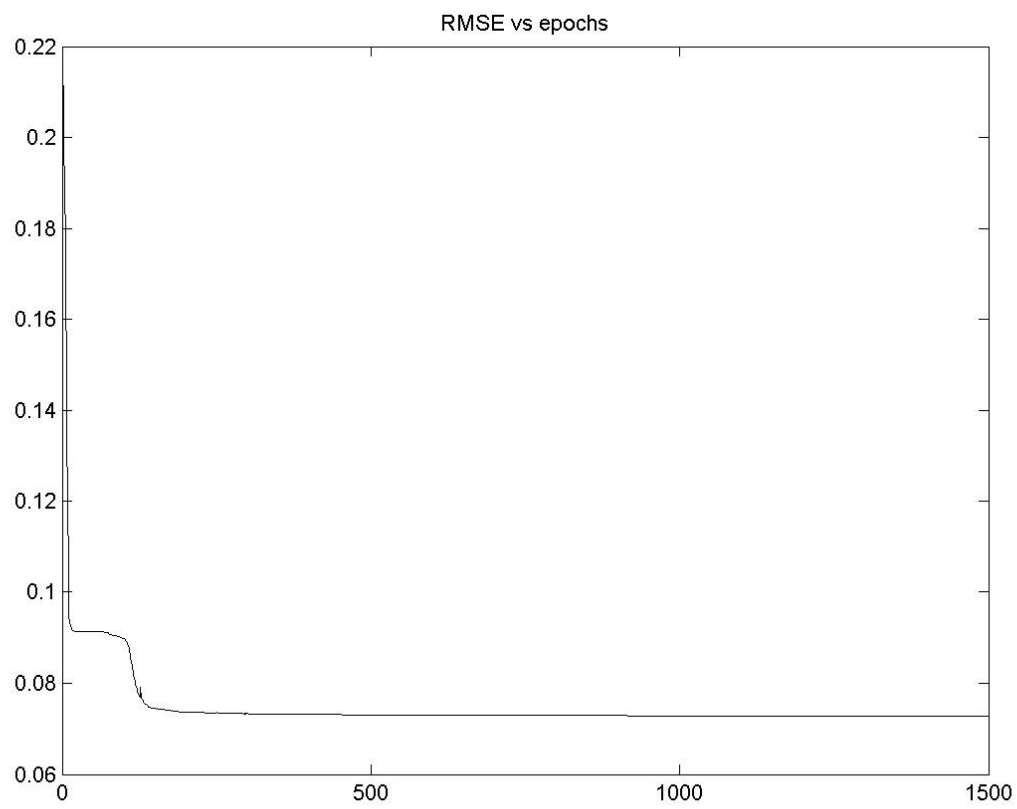
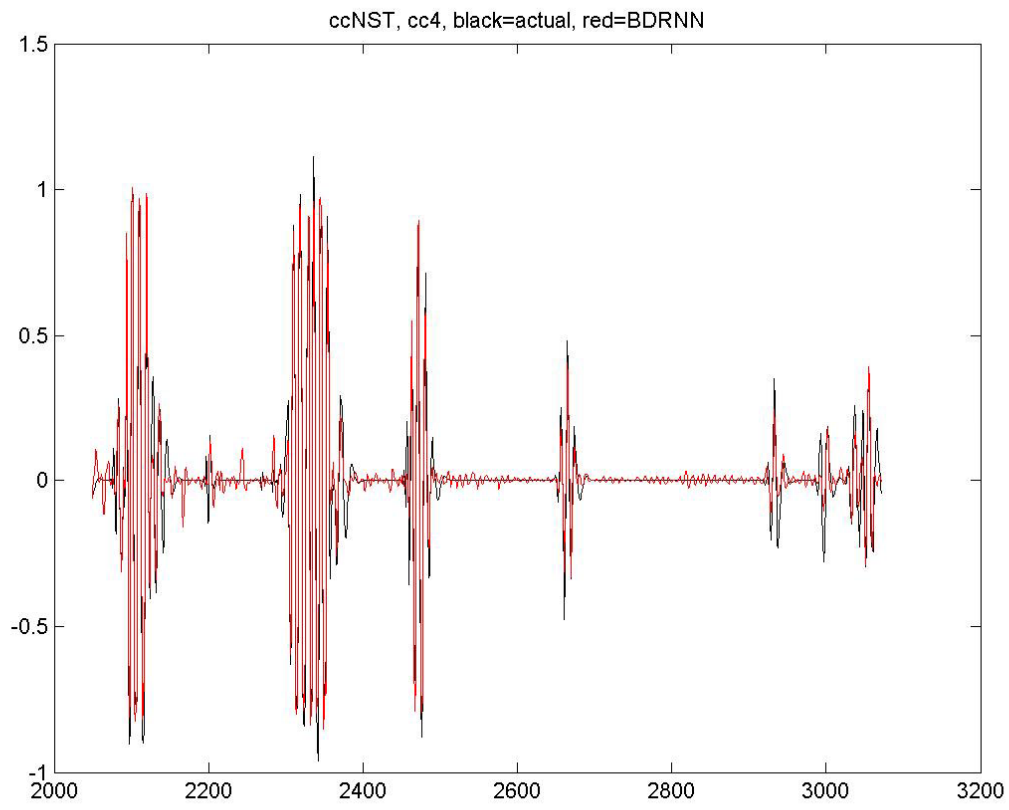
39^η Επανάληψη



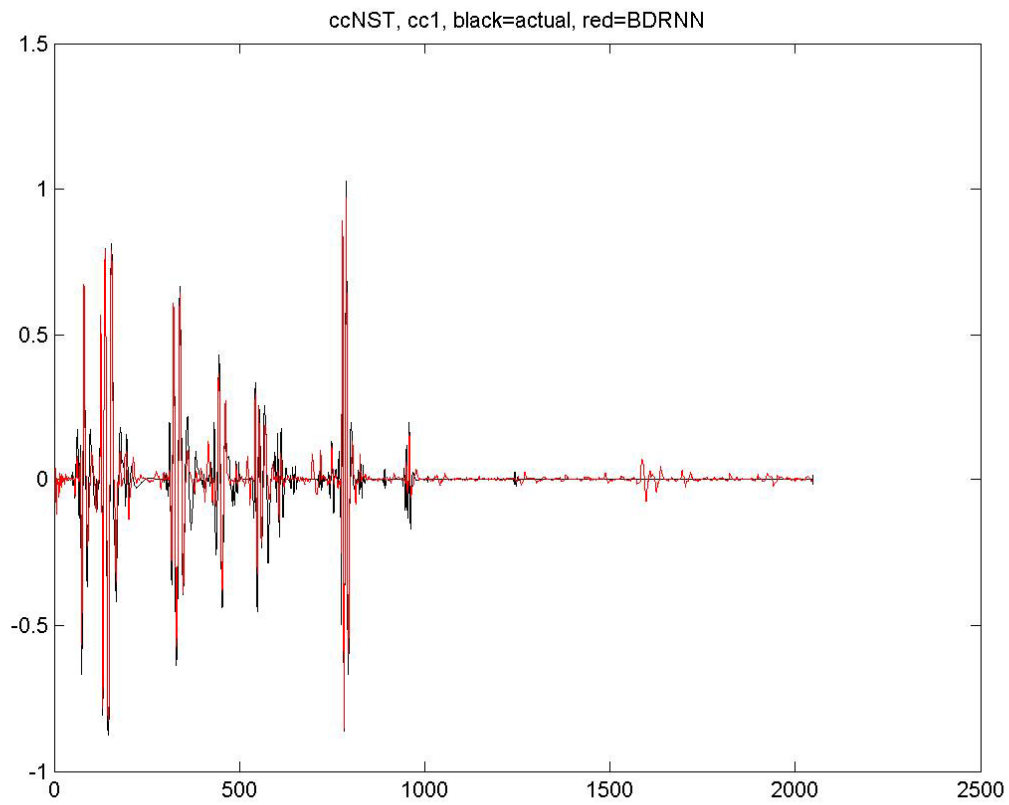
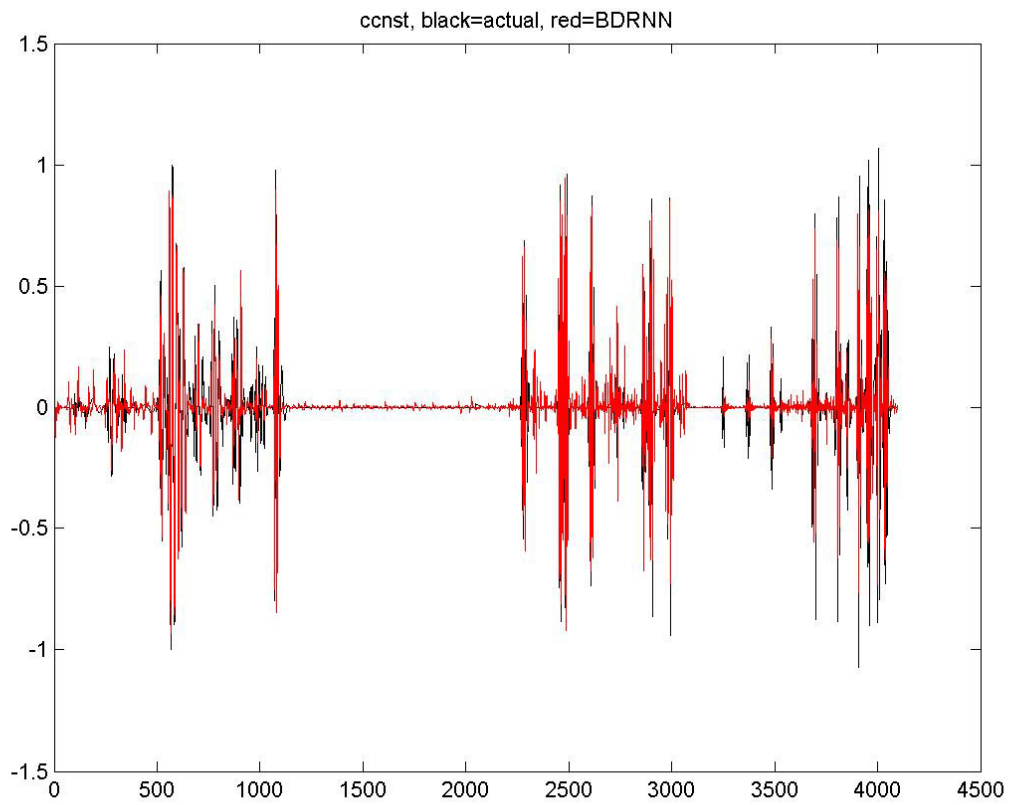


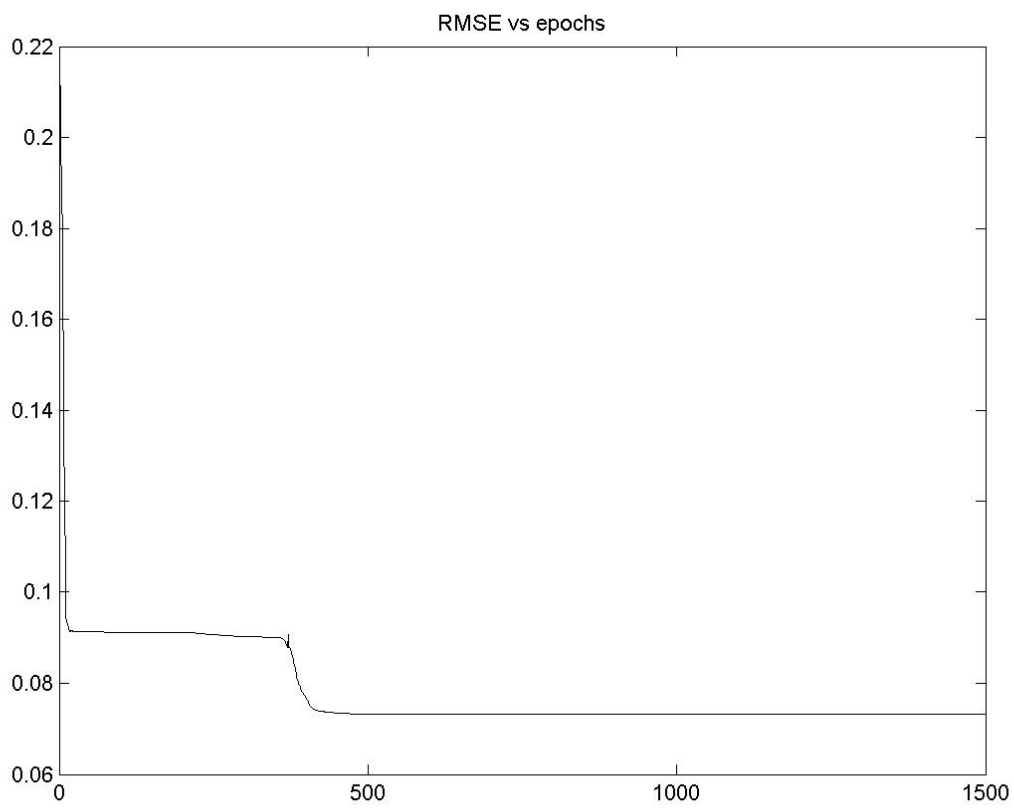
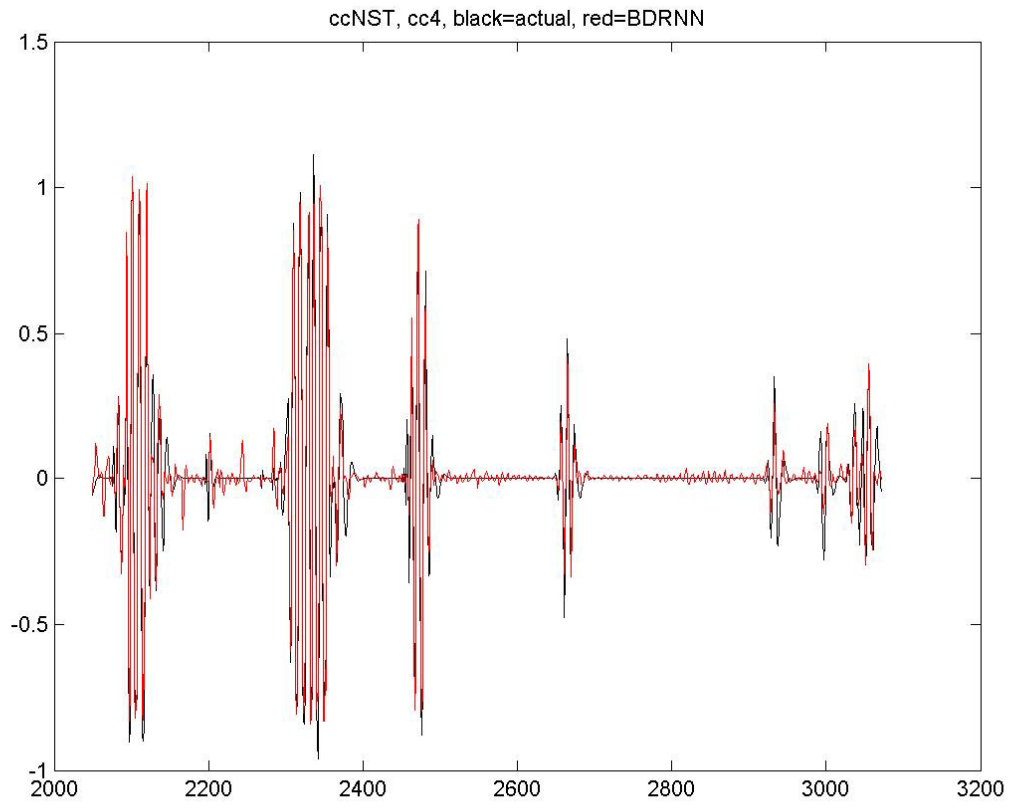
40^η Επανάληψη



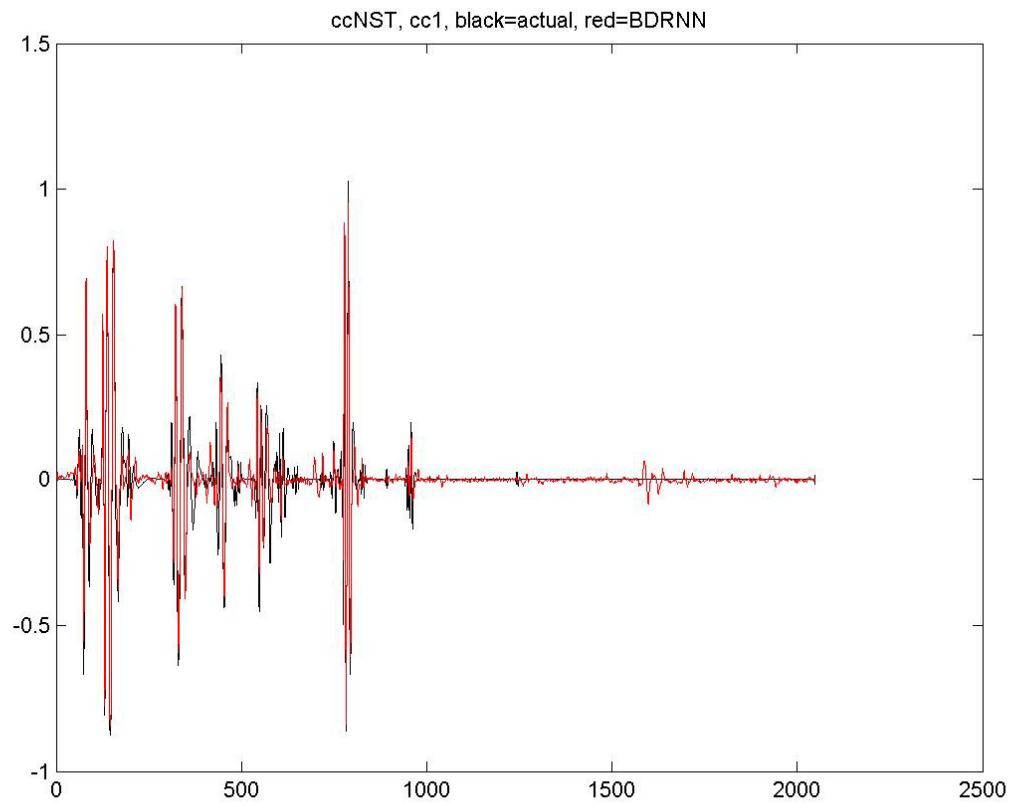
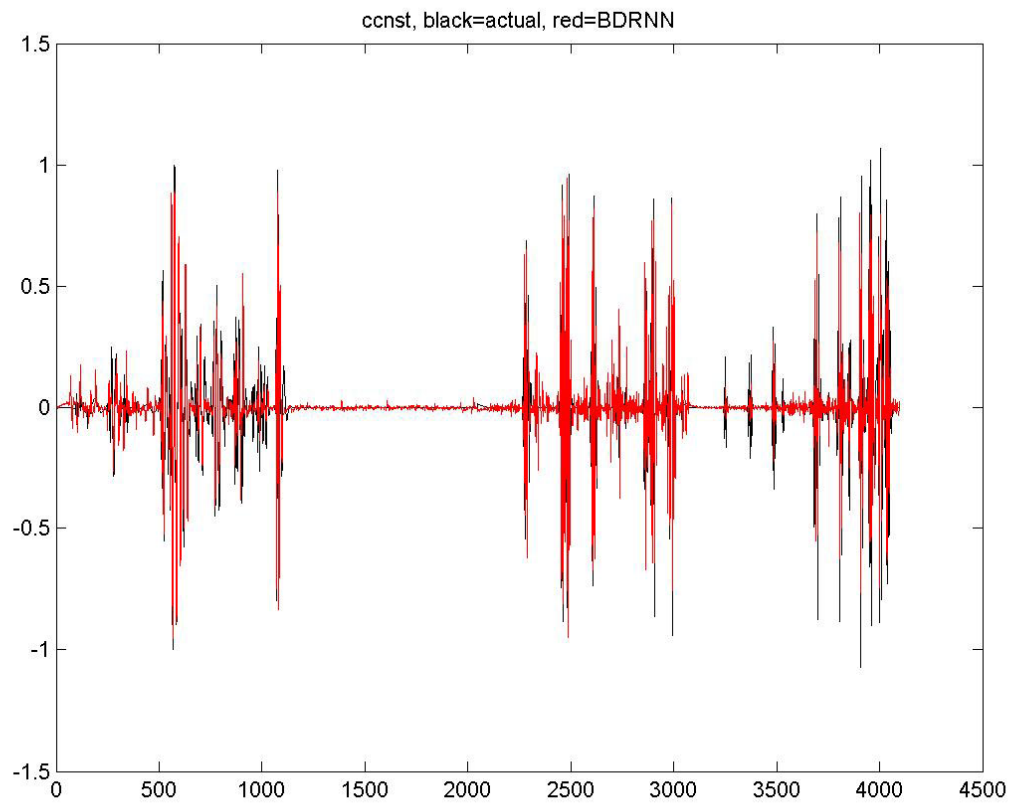


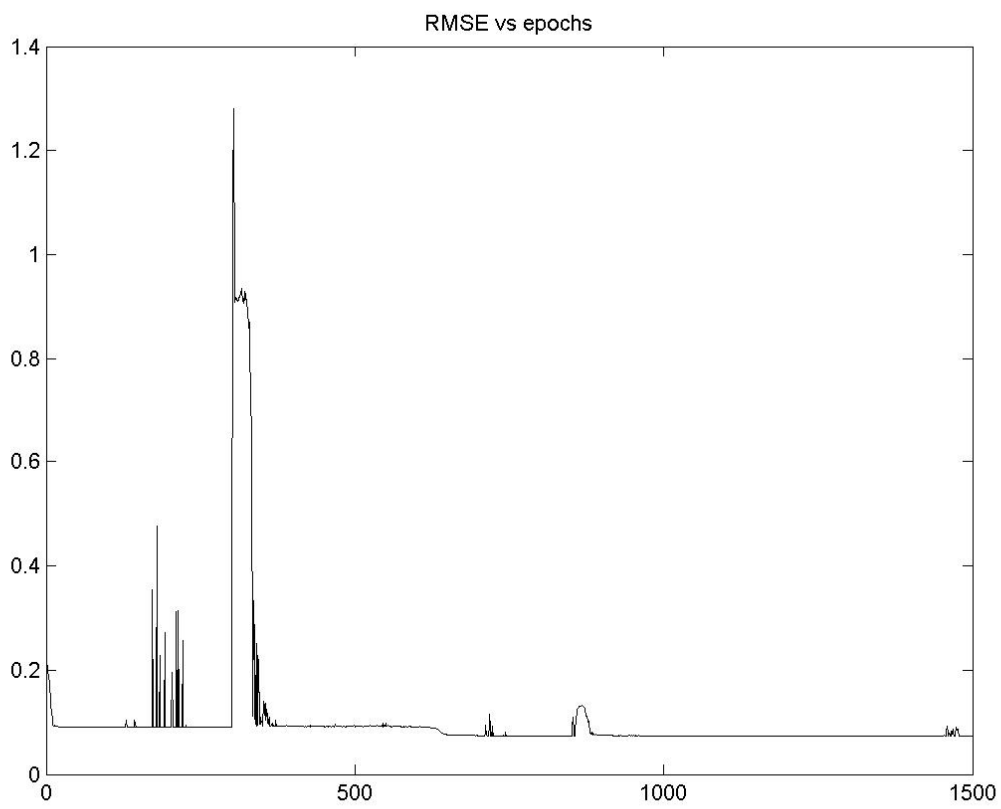
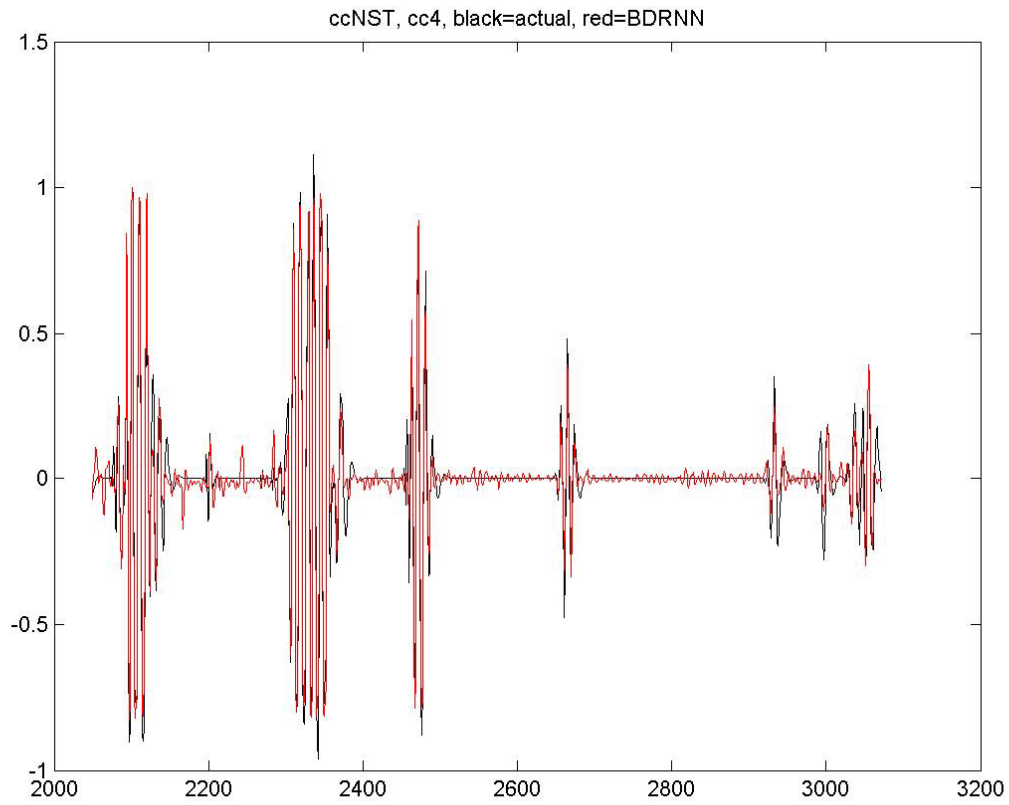
41^η Επανάληψη



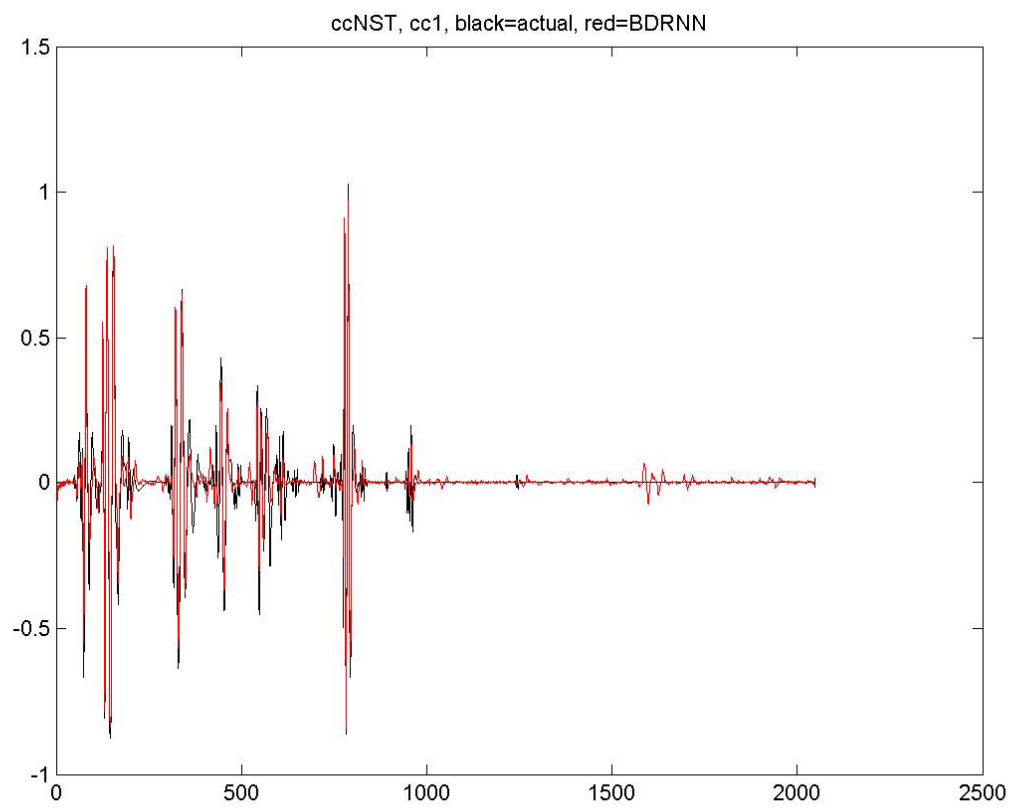
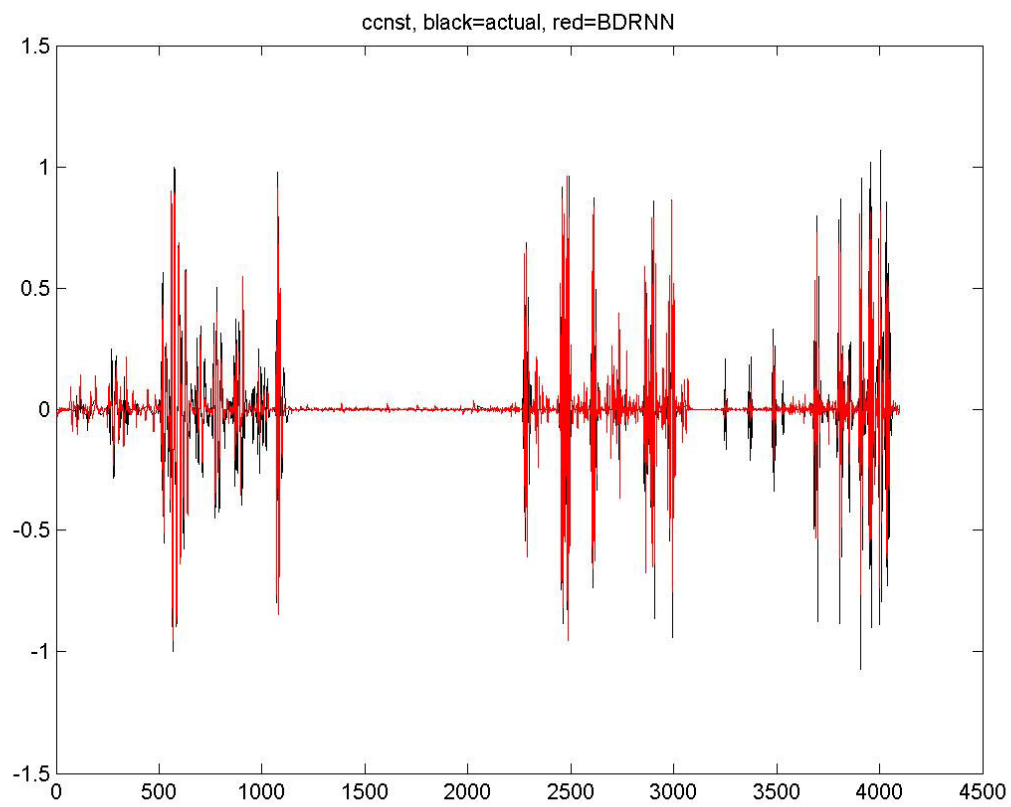


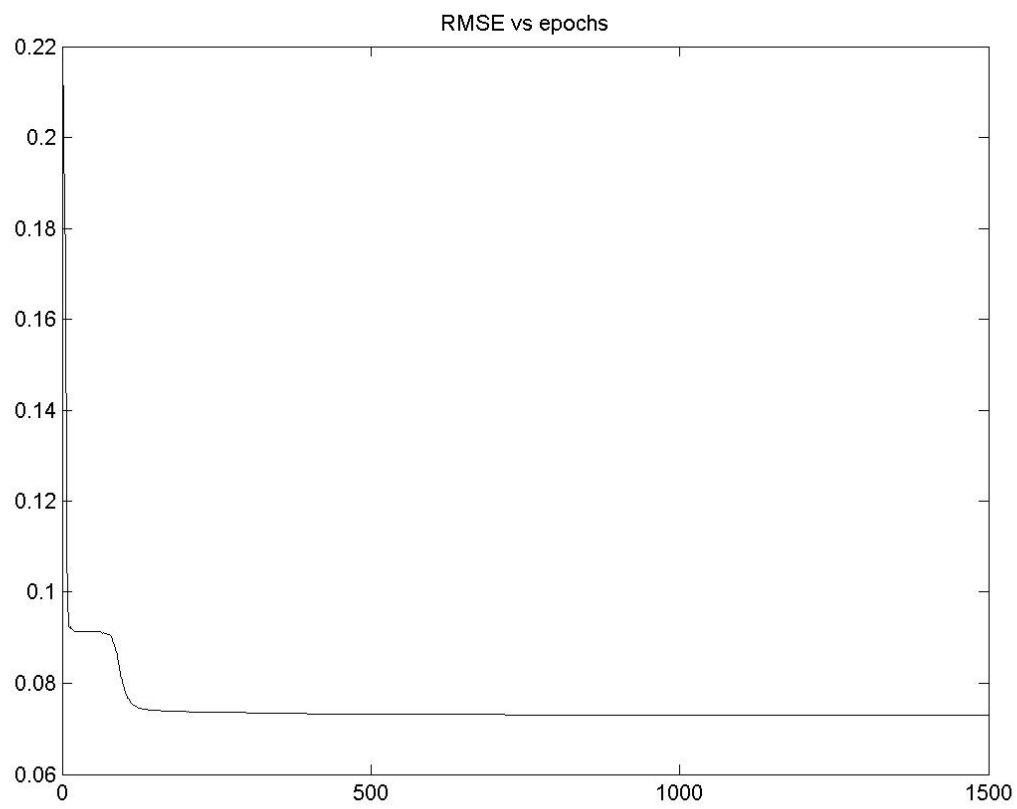
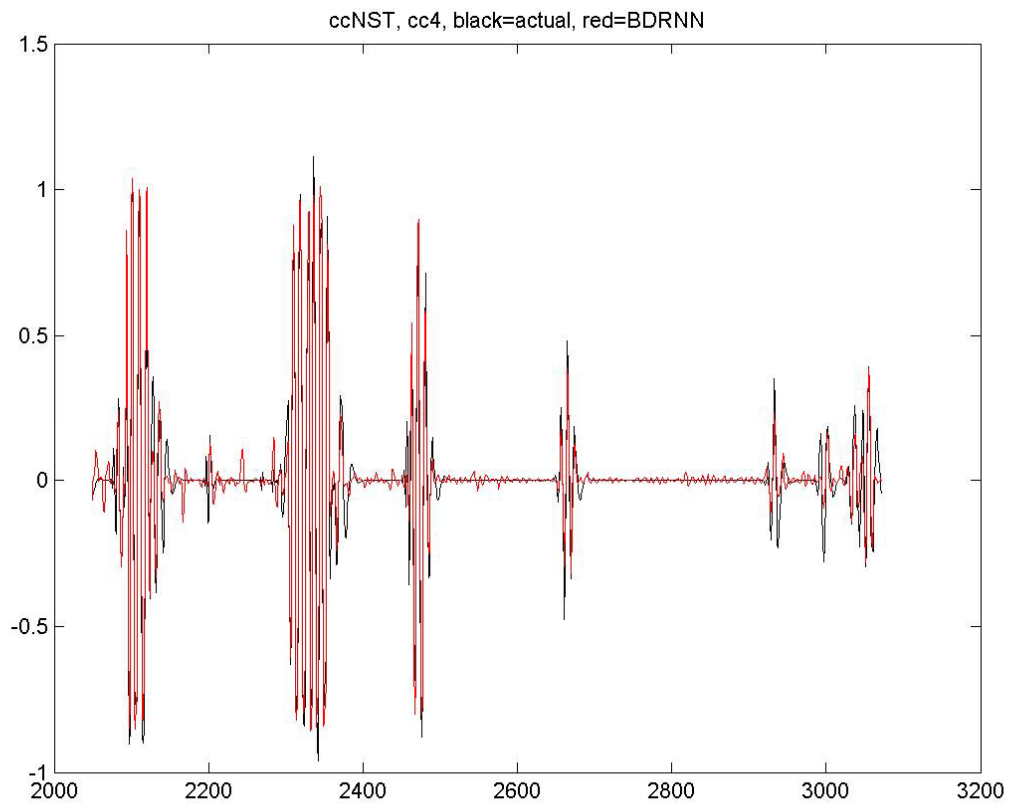
42^η Επανάληψη



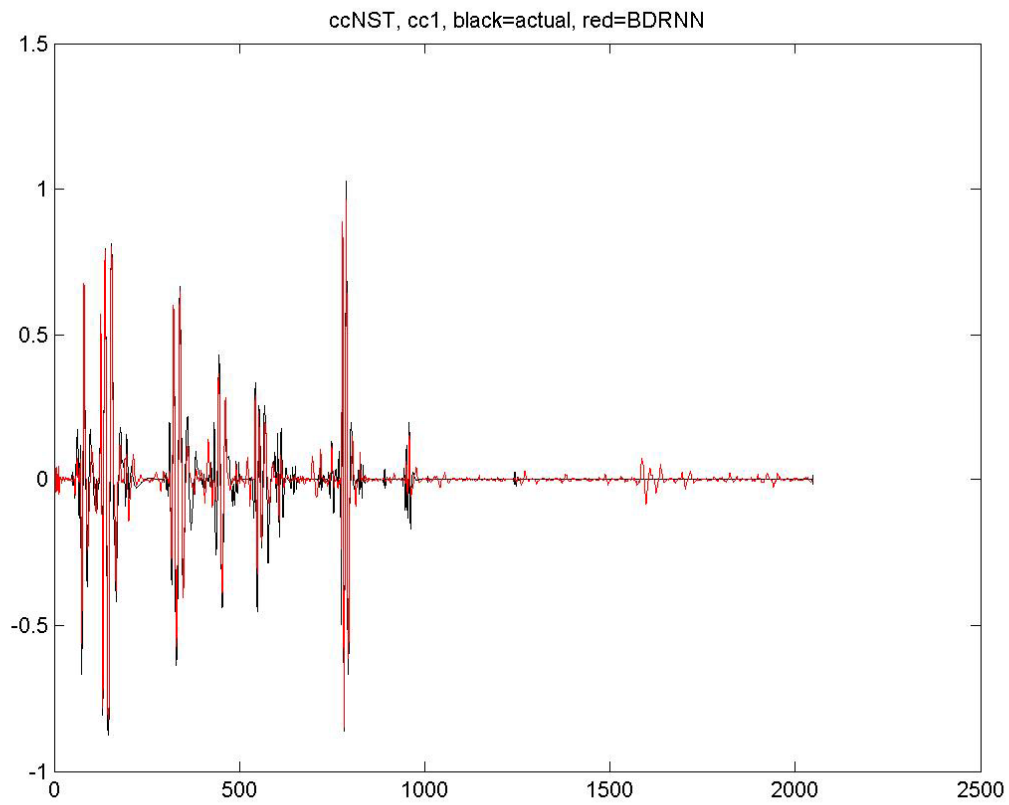
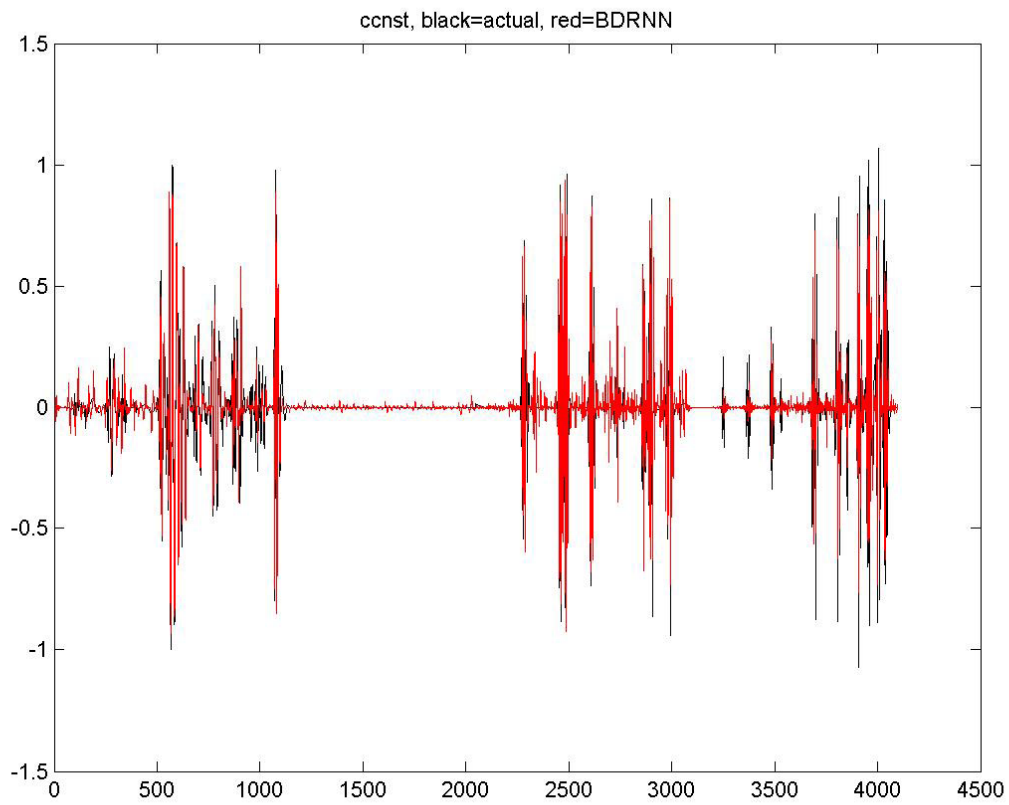


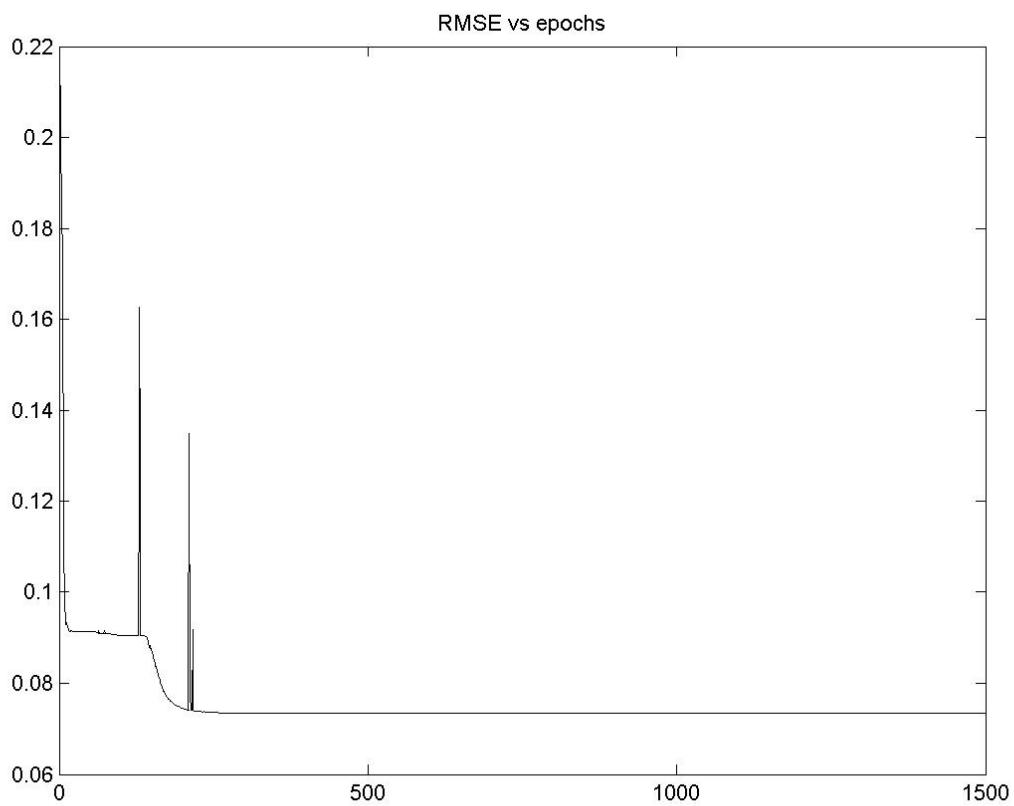
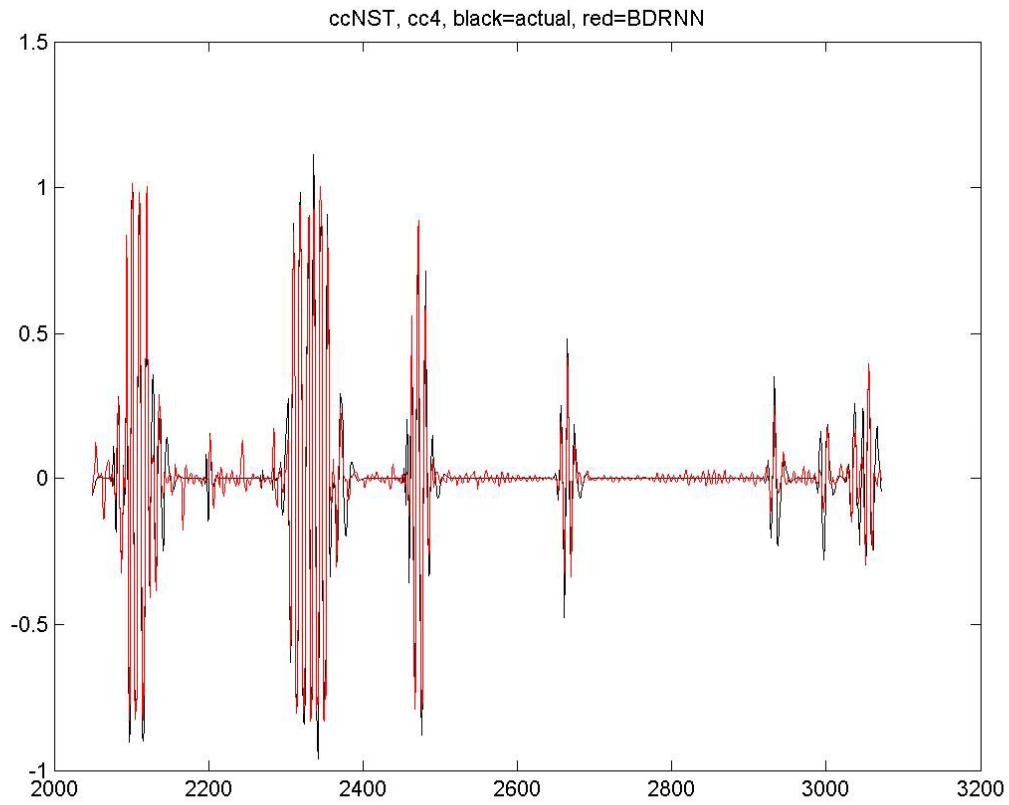
43^η Επανάληψη



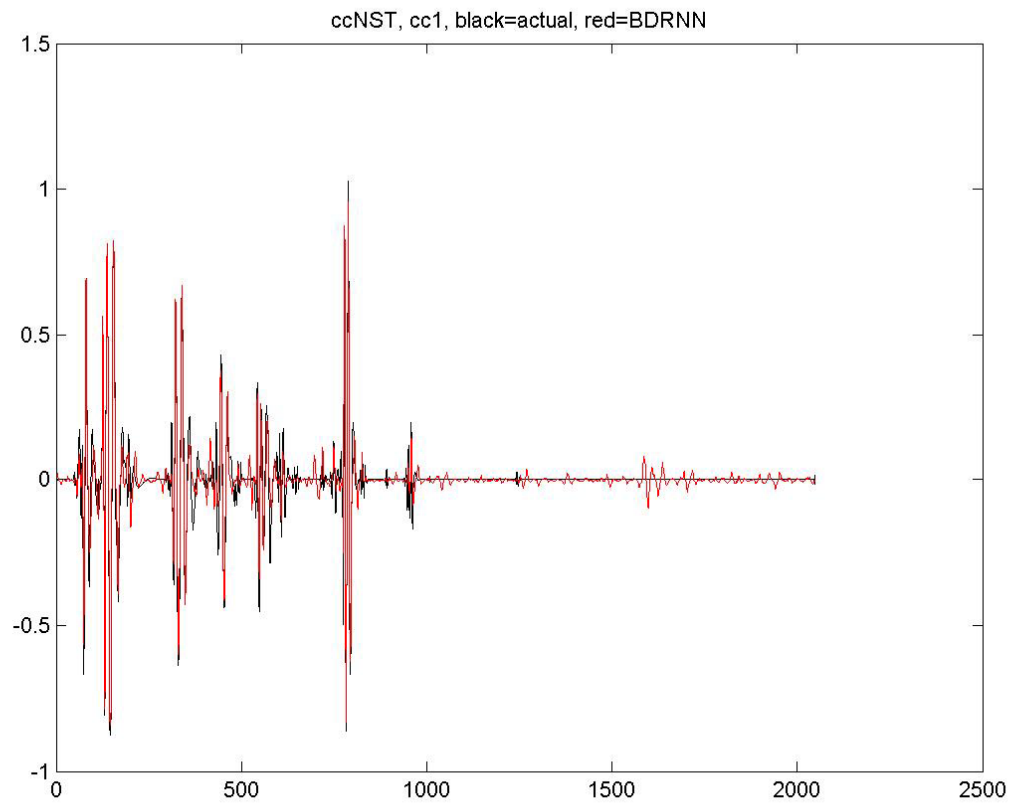
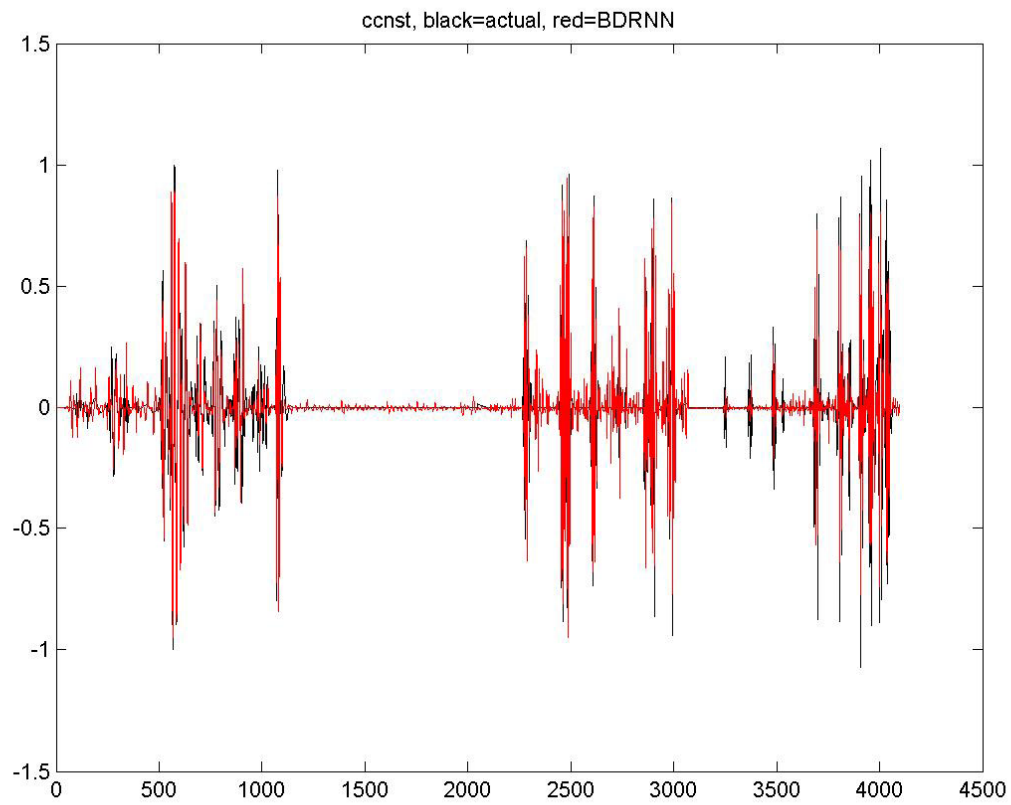


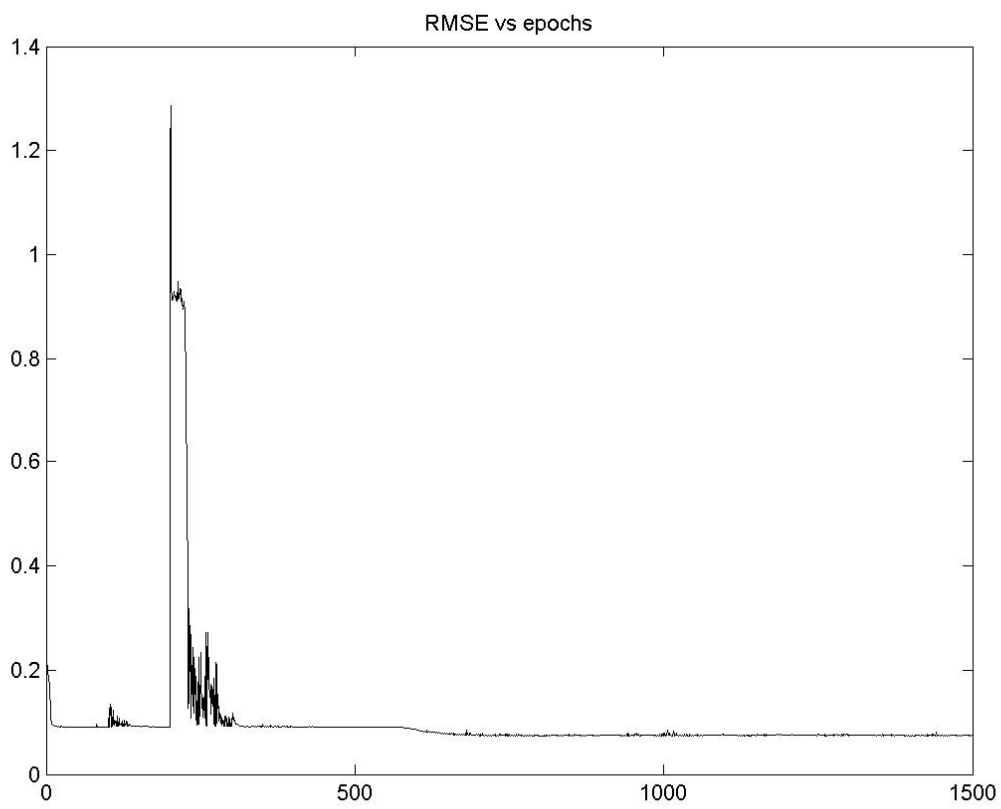
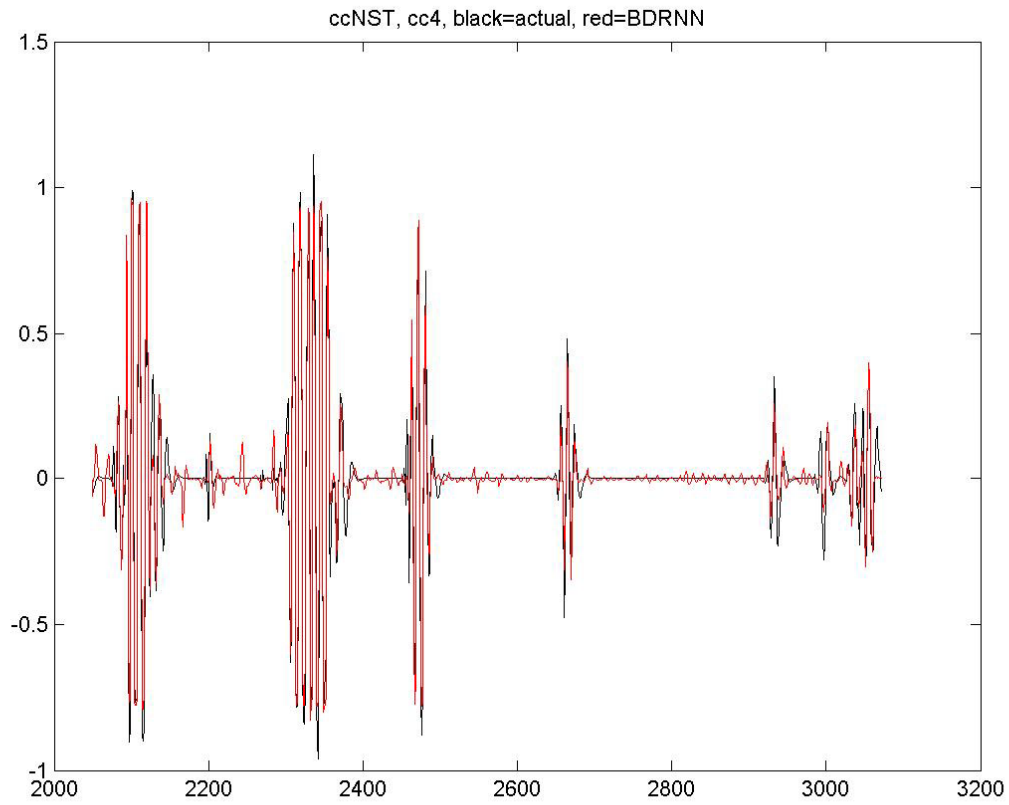
44^η Επανάληψη



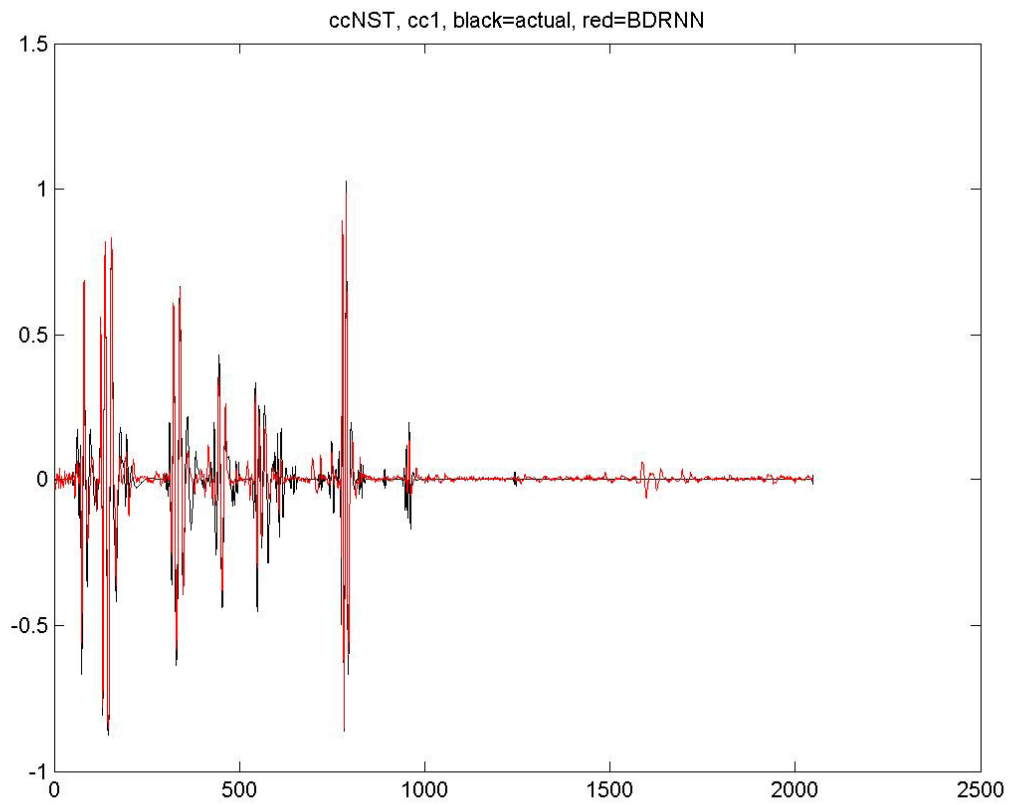
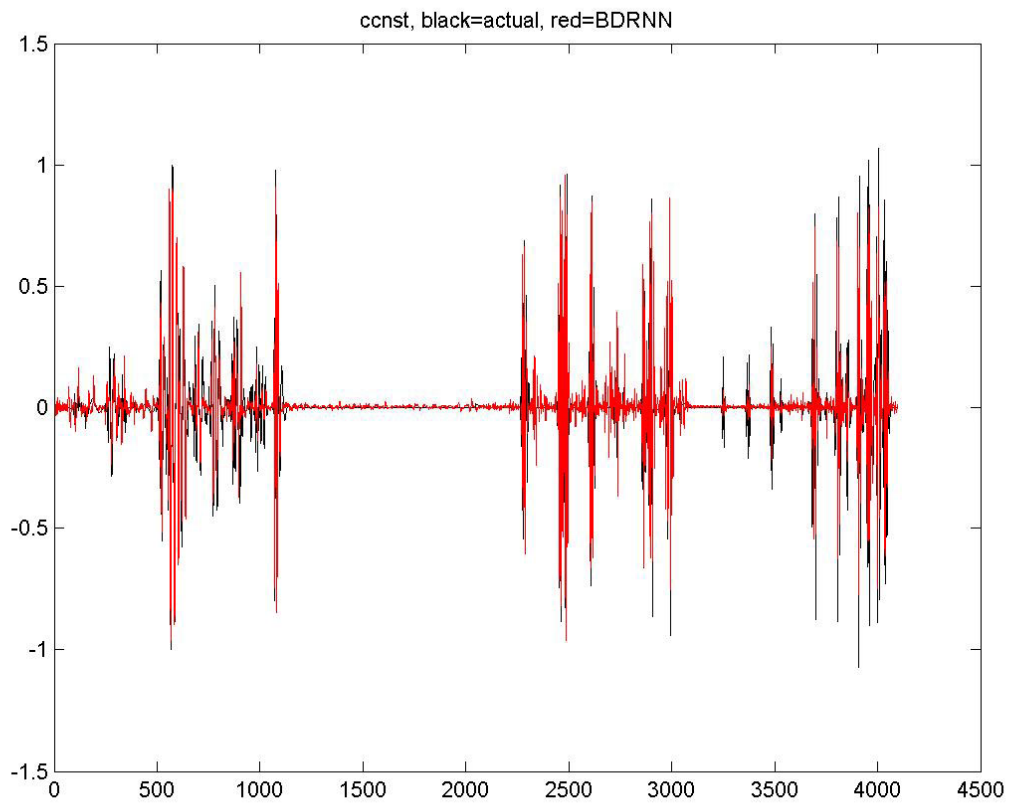


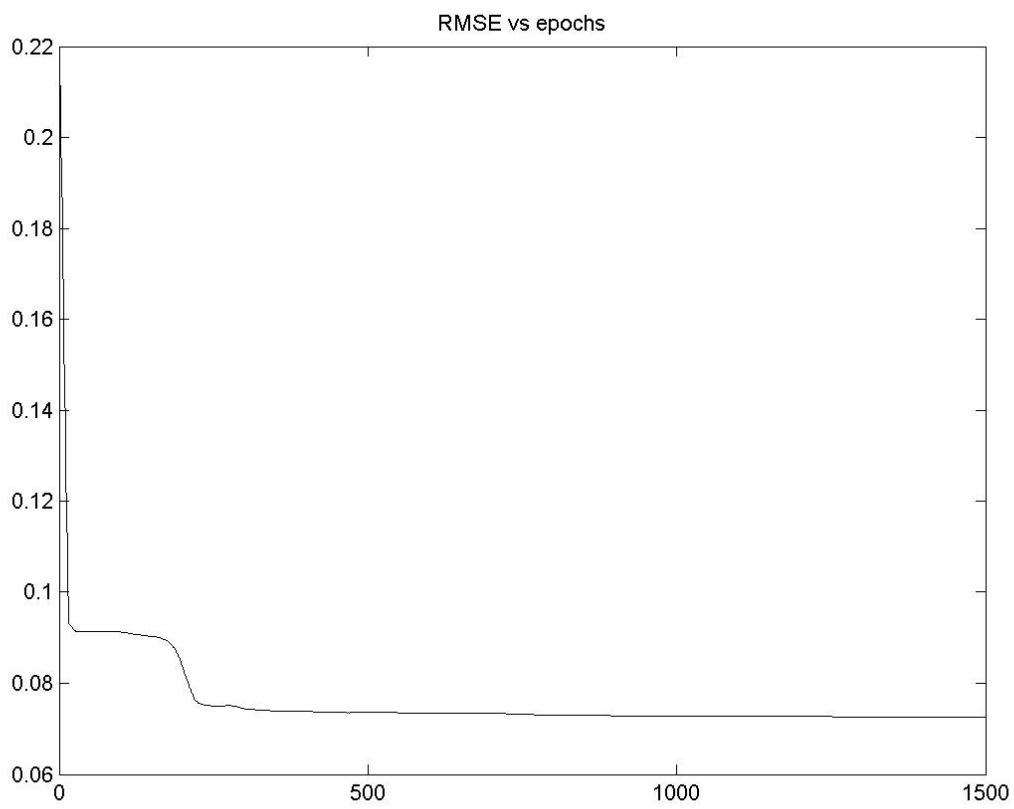
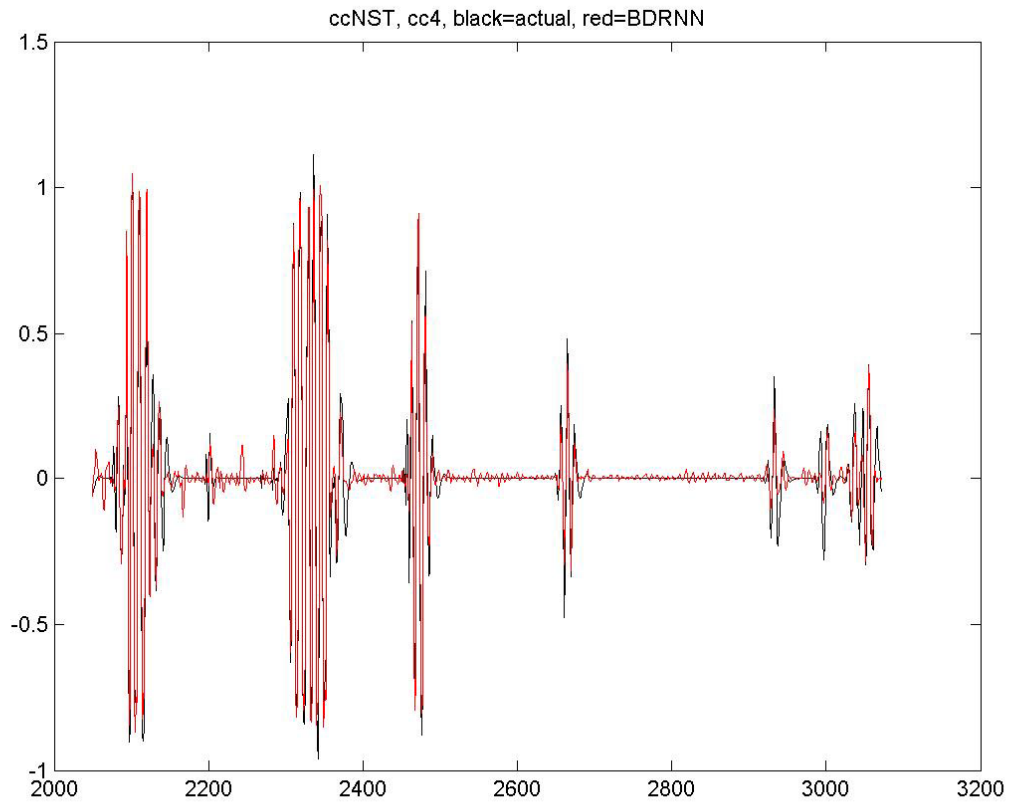
45^η Επανάληψη



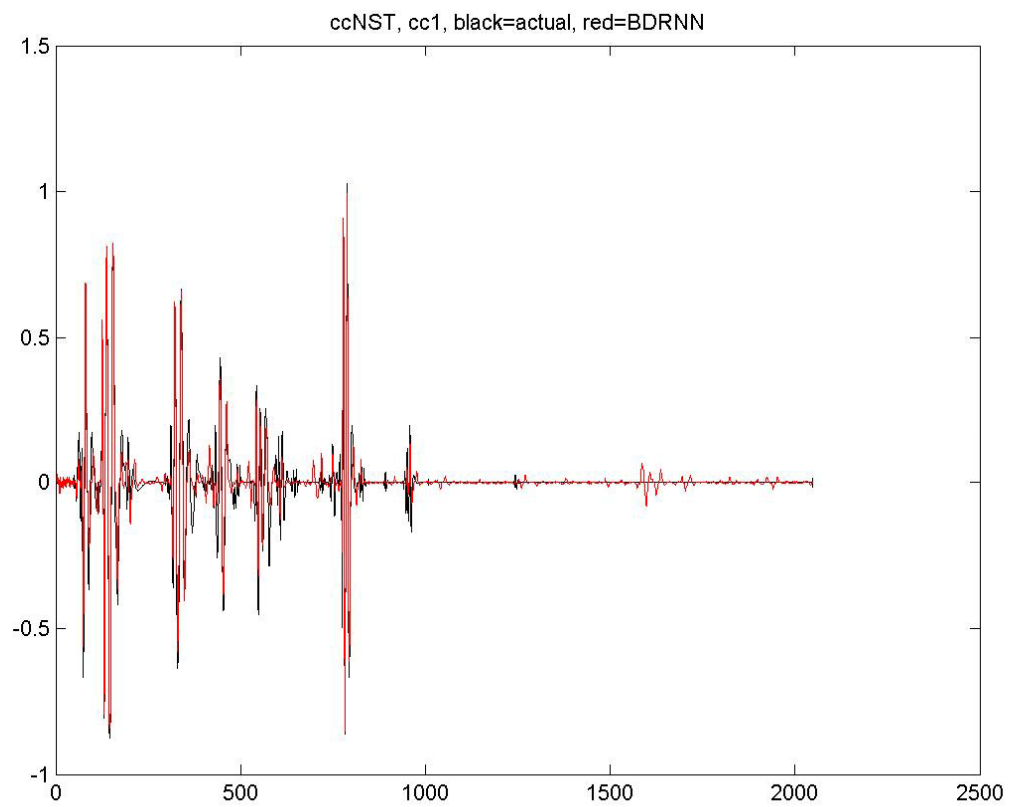
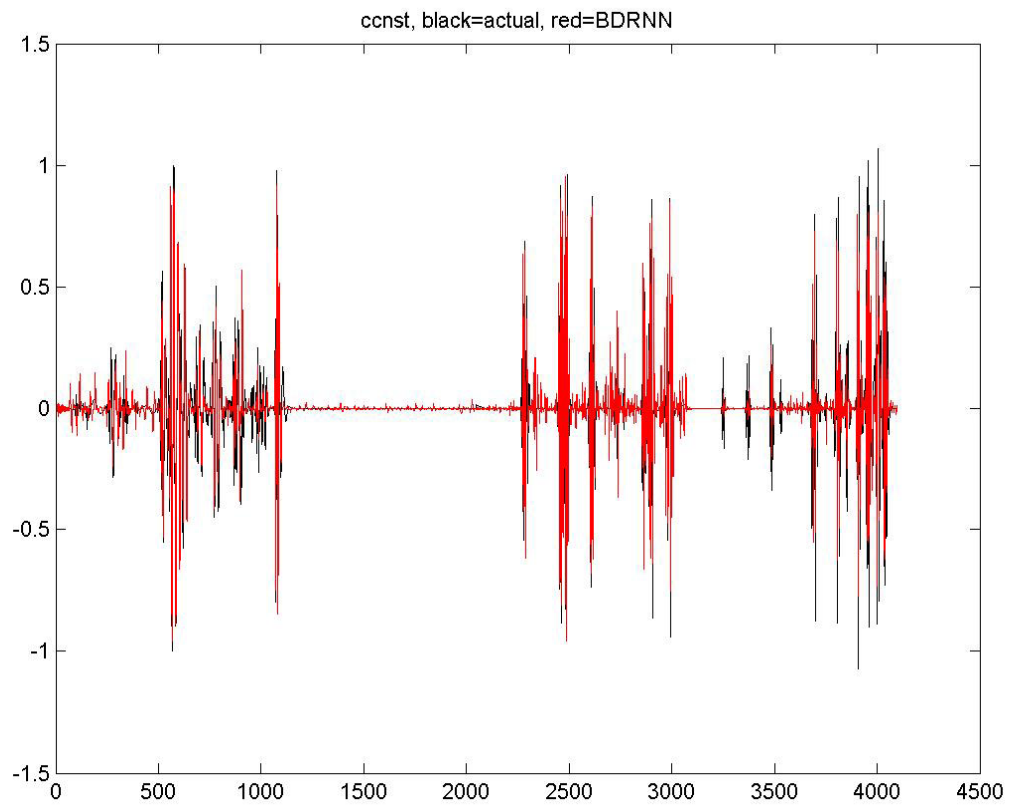


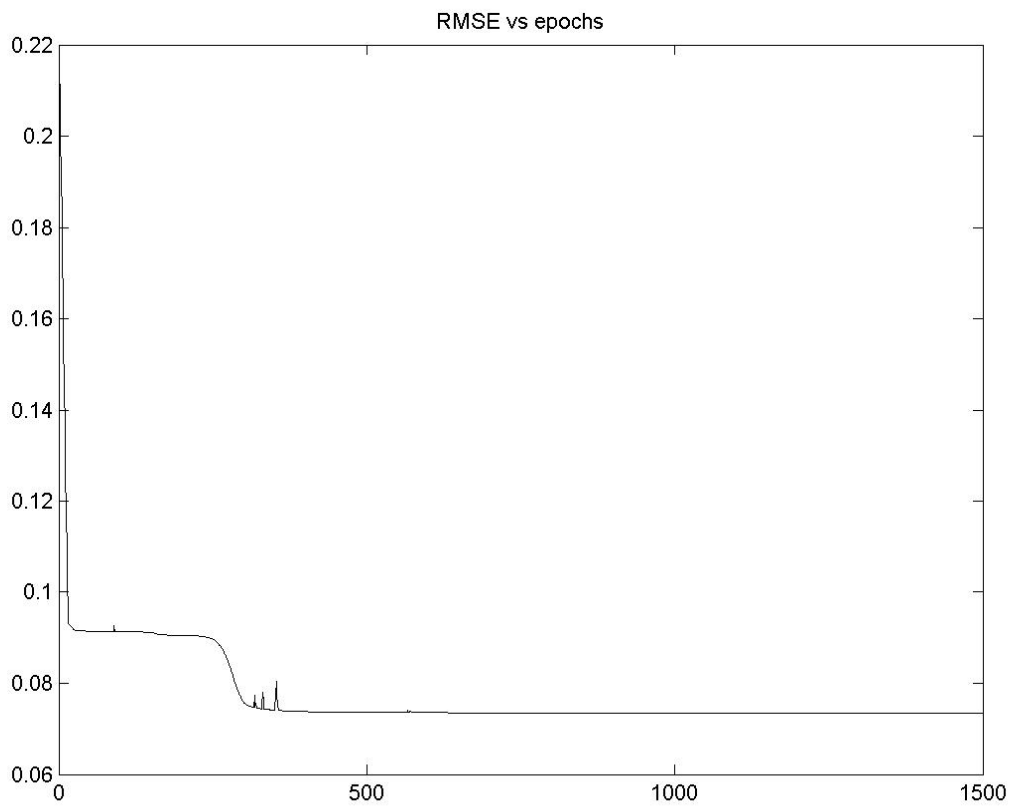
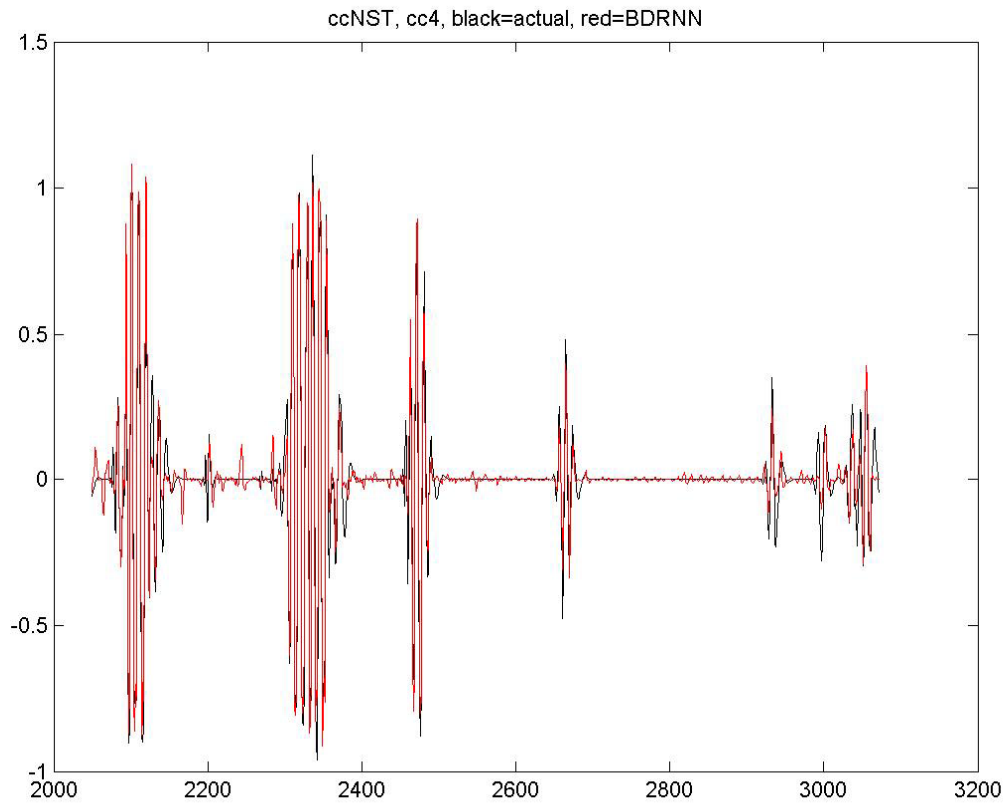
46^η Επανάληψη



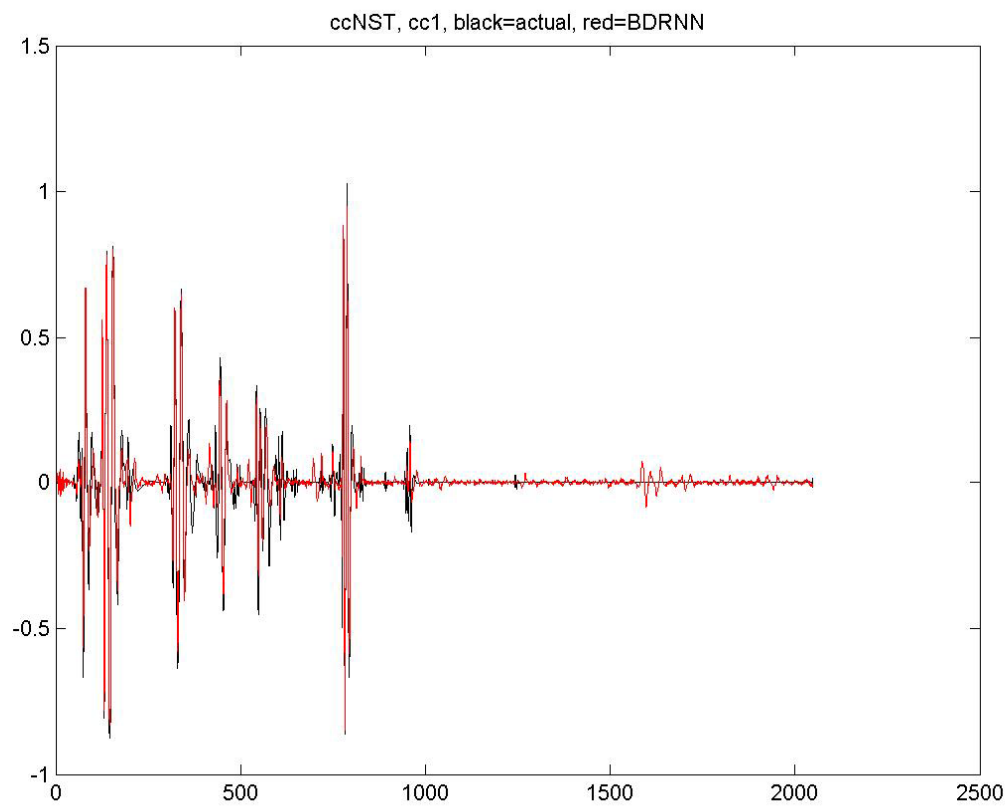
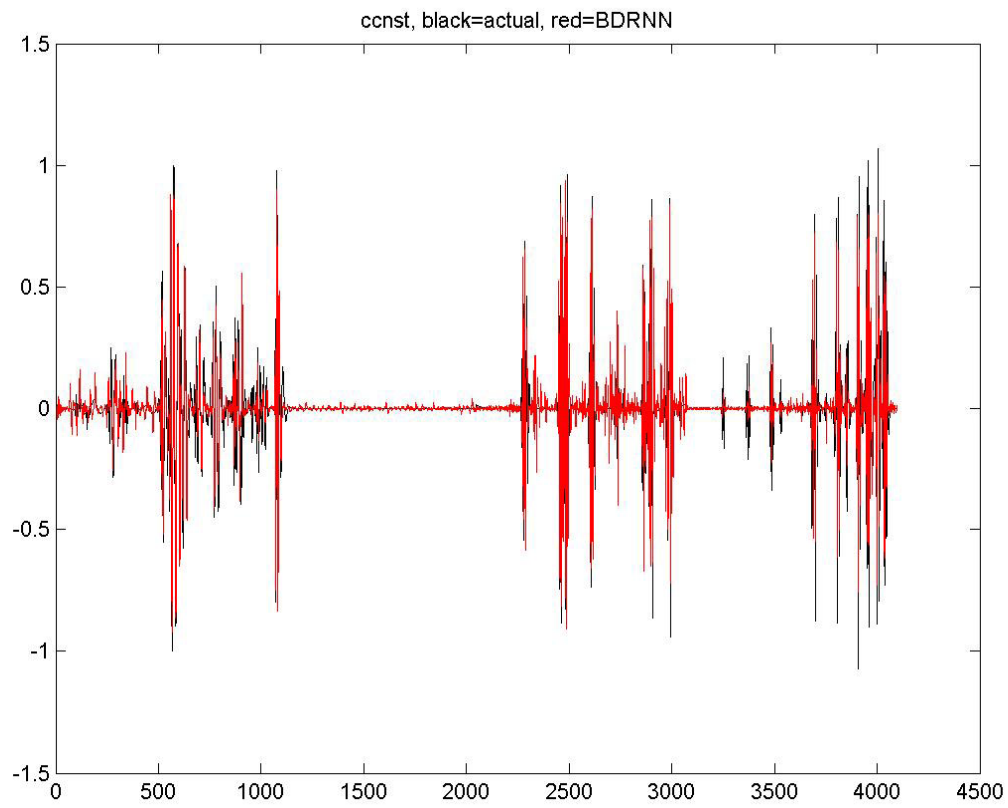


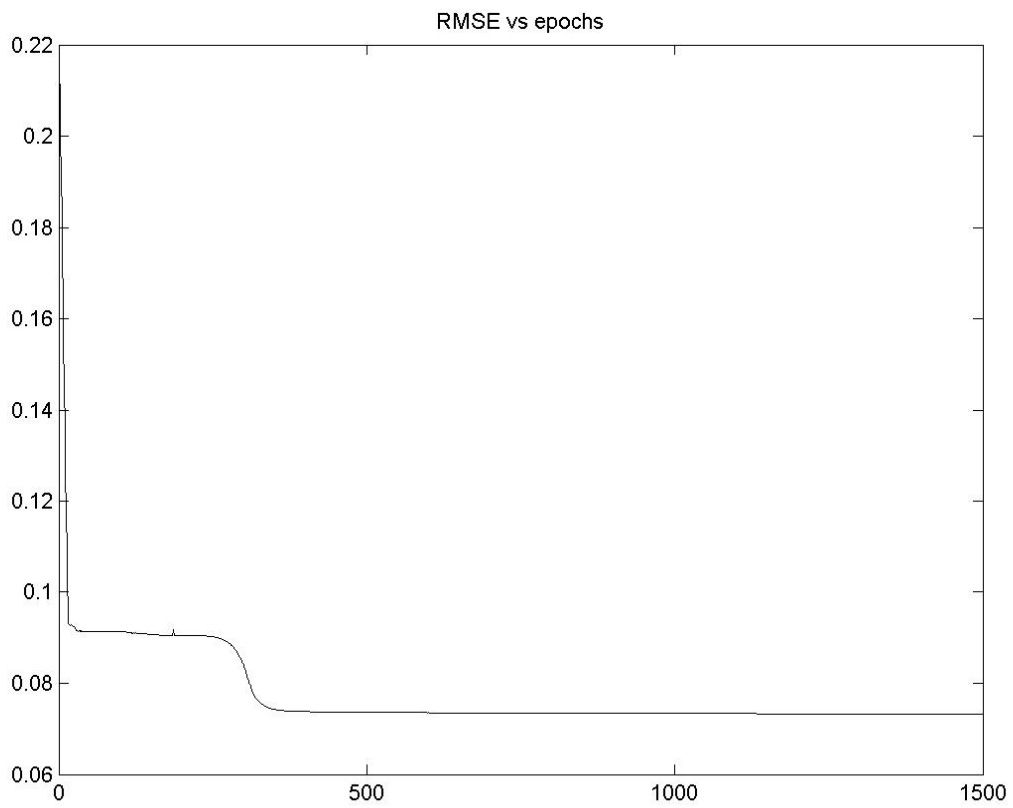
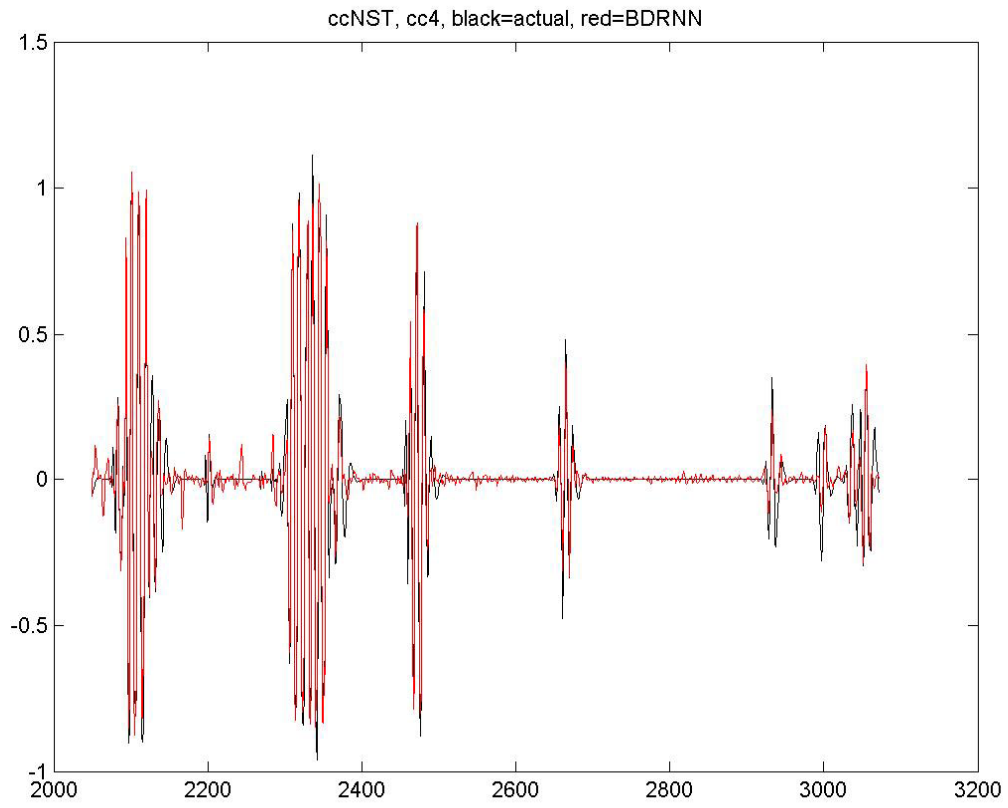
47^η Επανάληψη



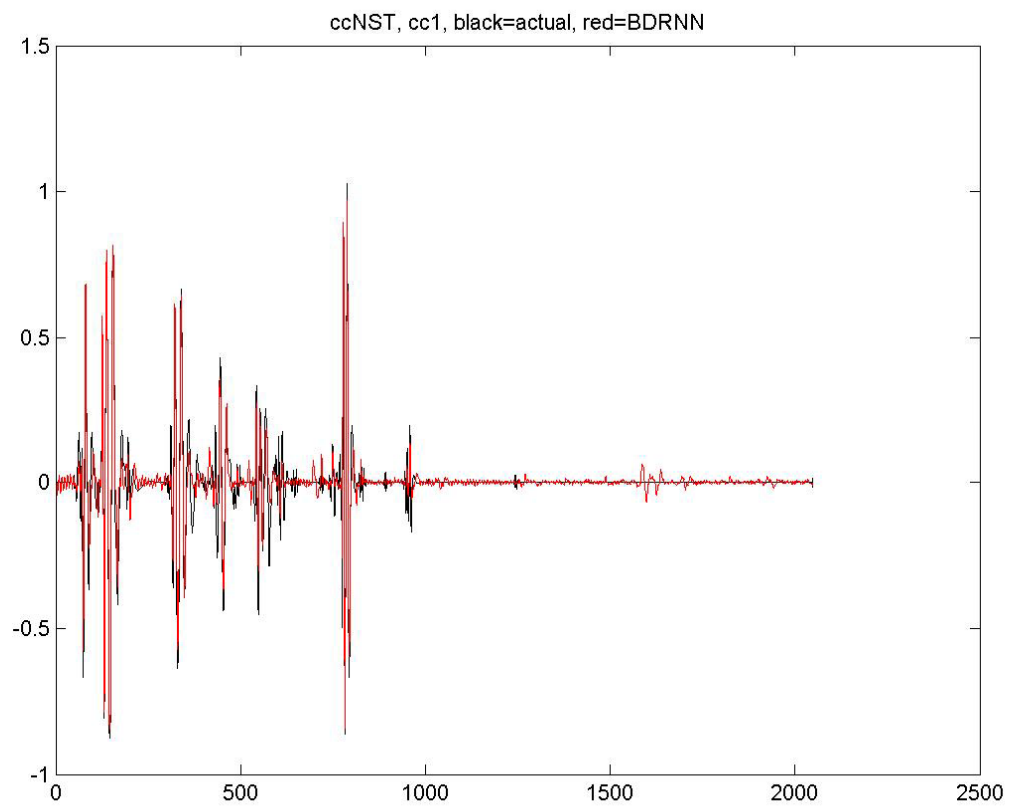
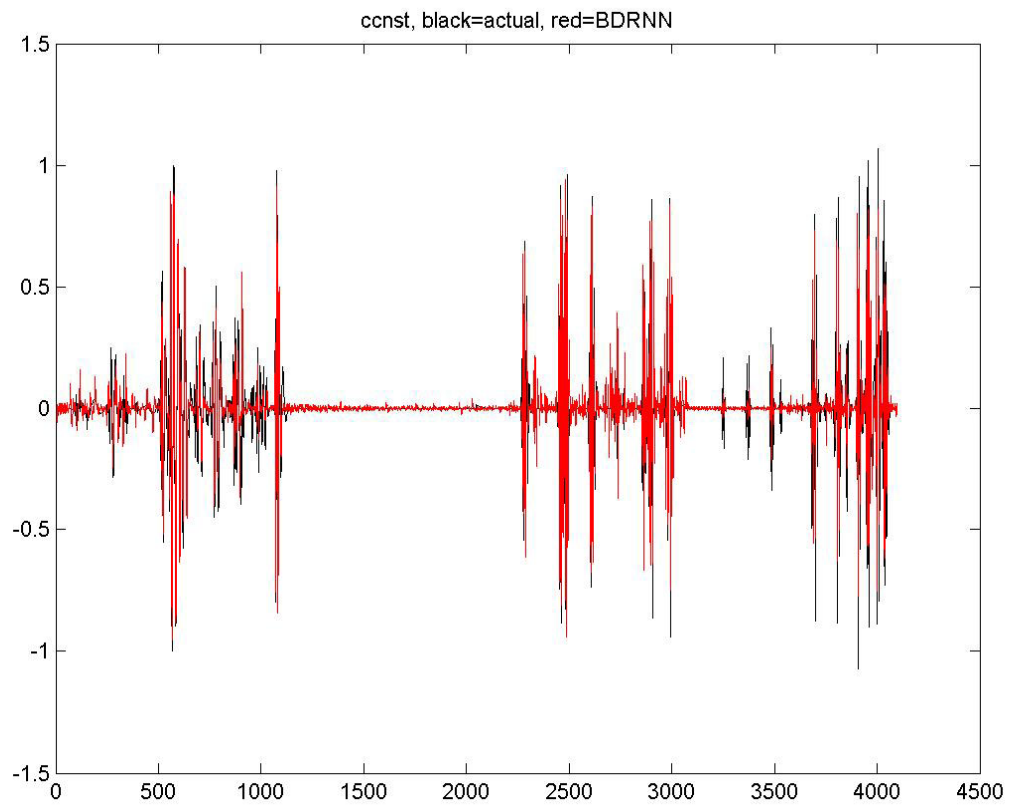


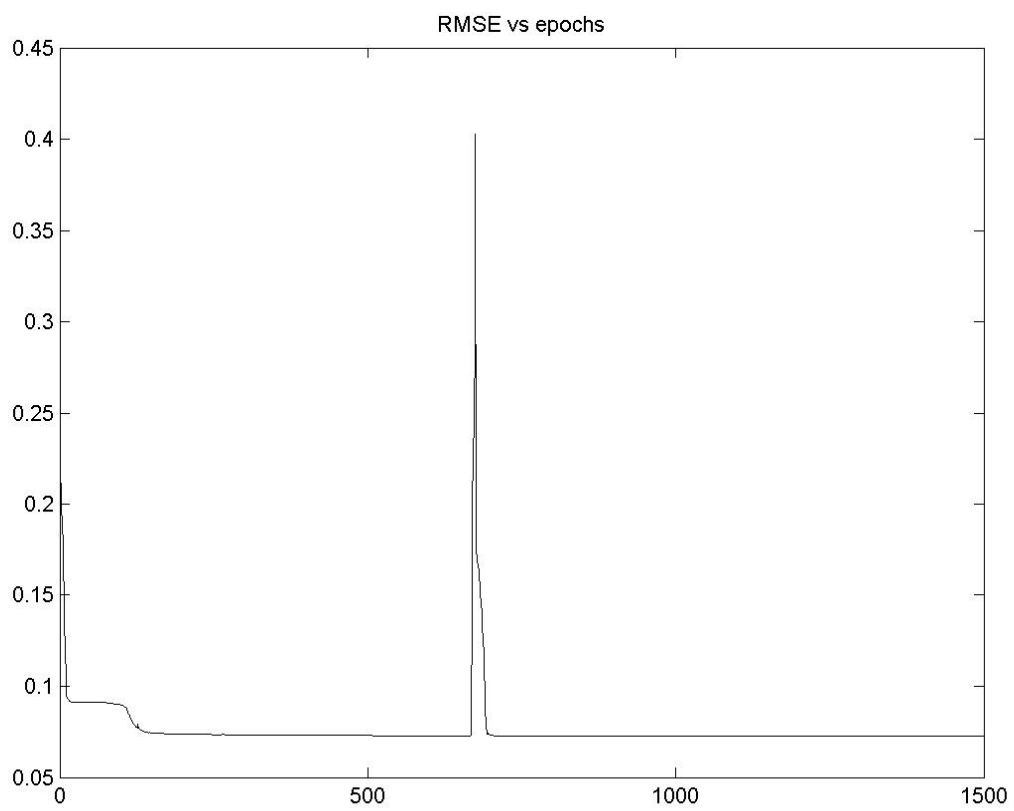
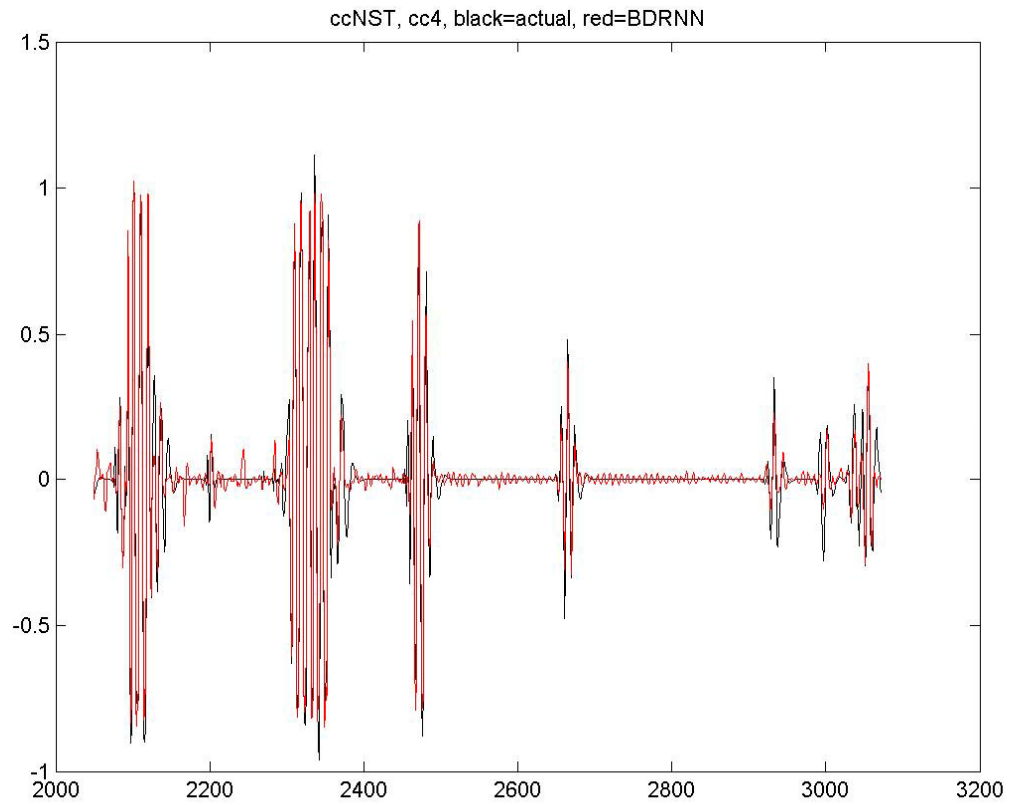
48^η Επανάληψη



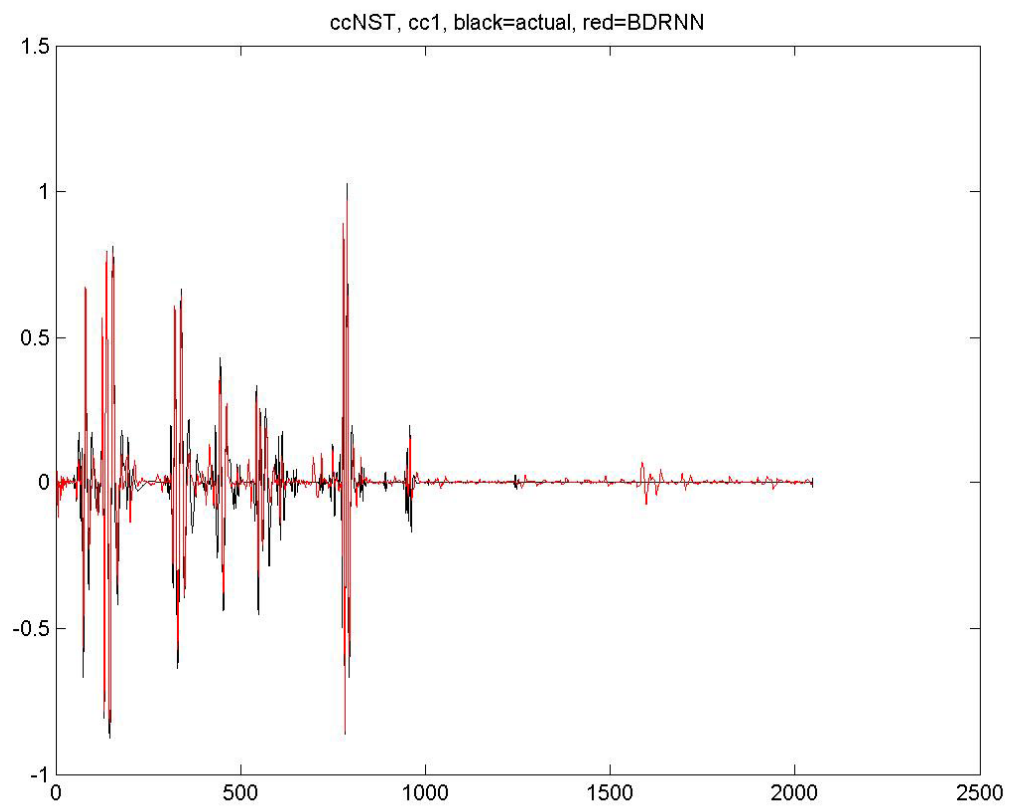
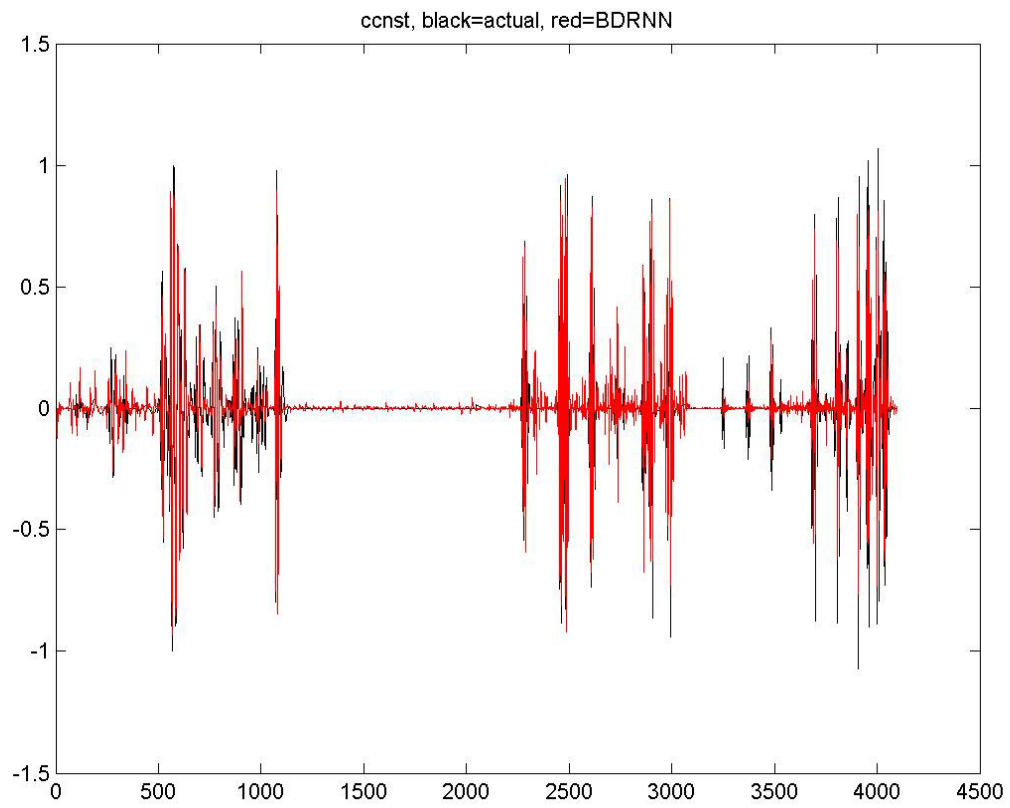


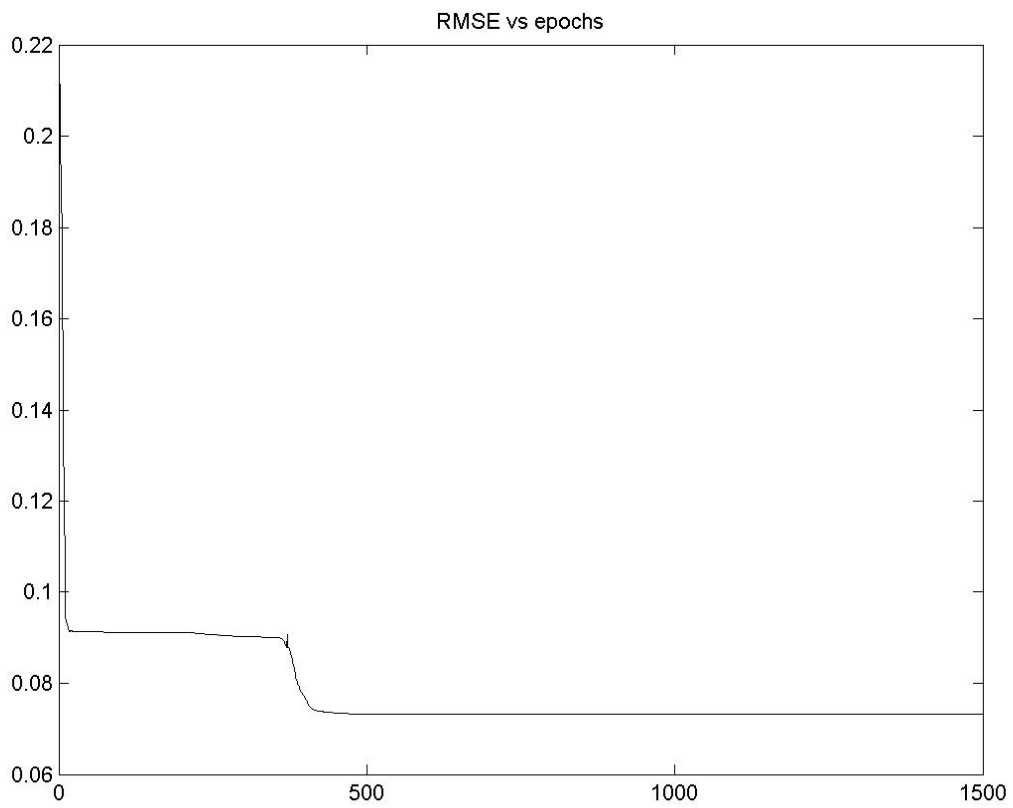
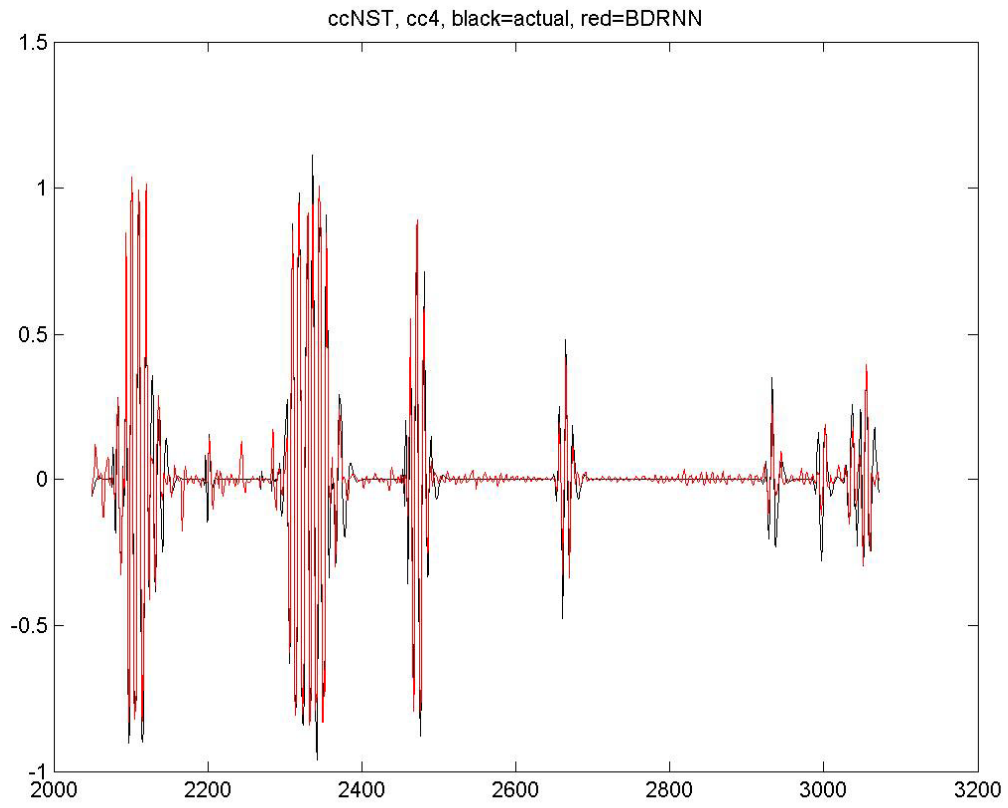
49^η Επανάληψη



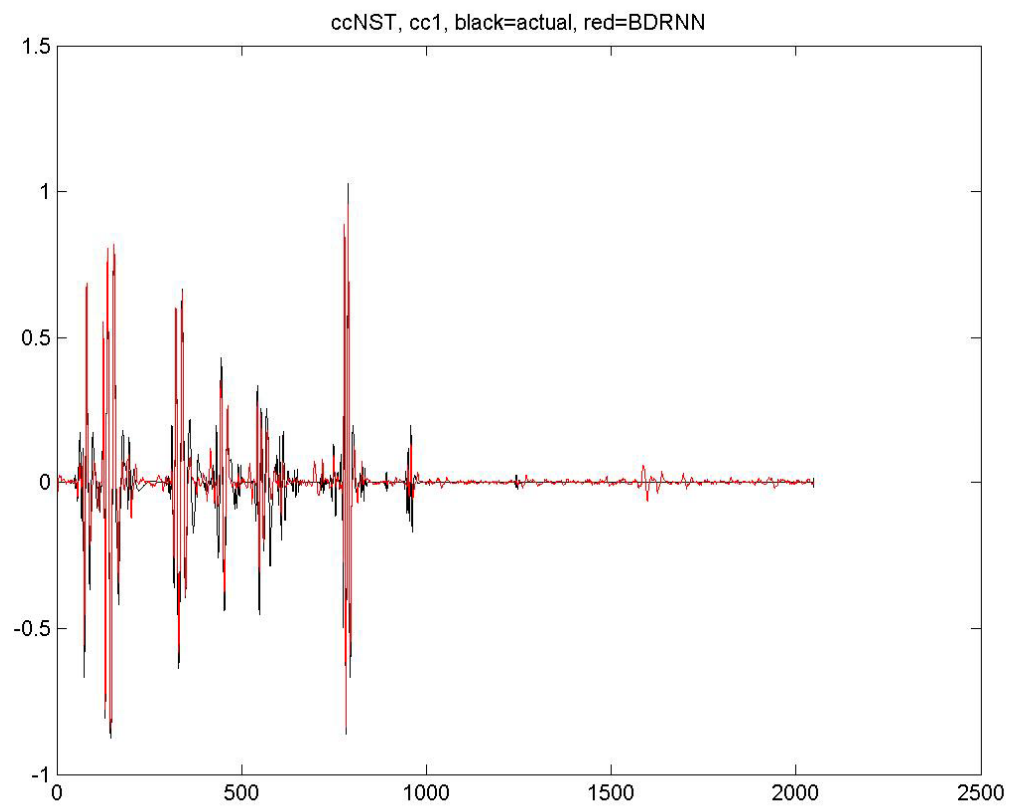
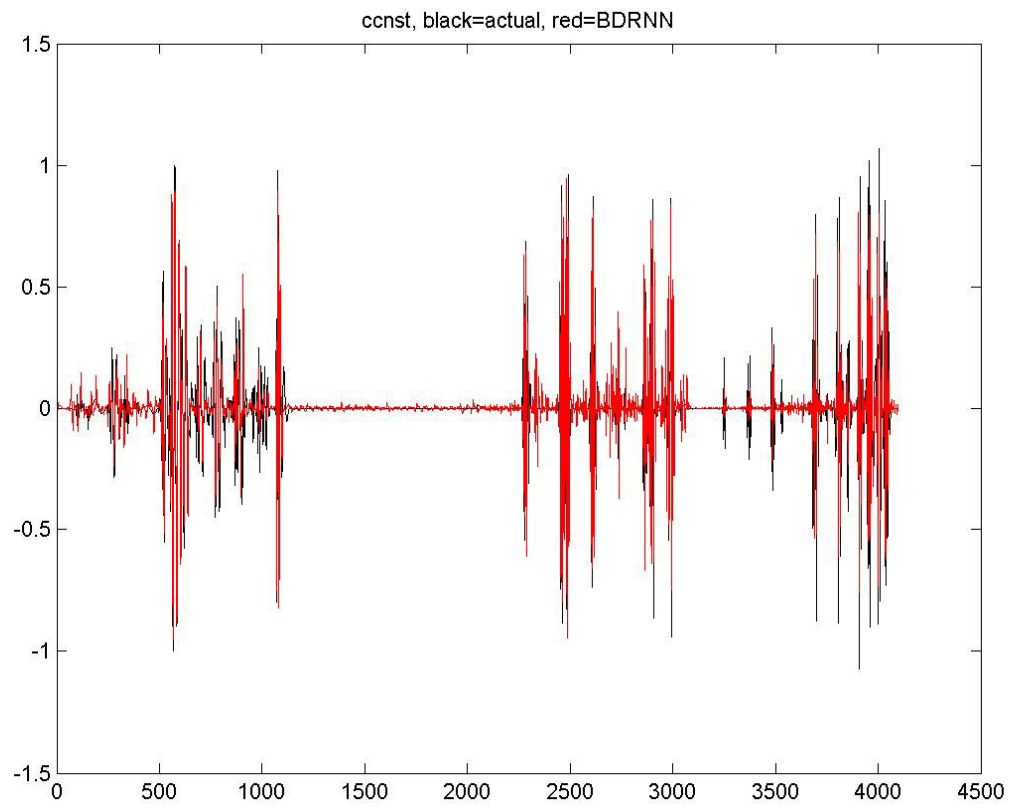


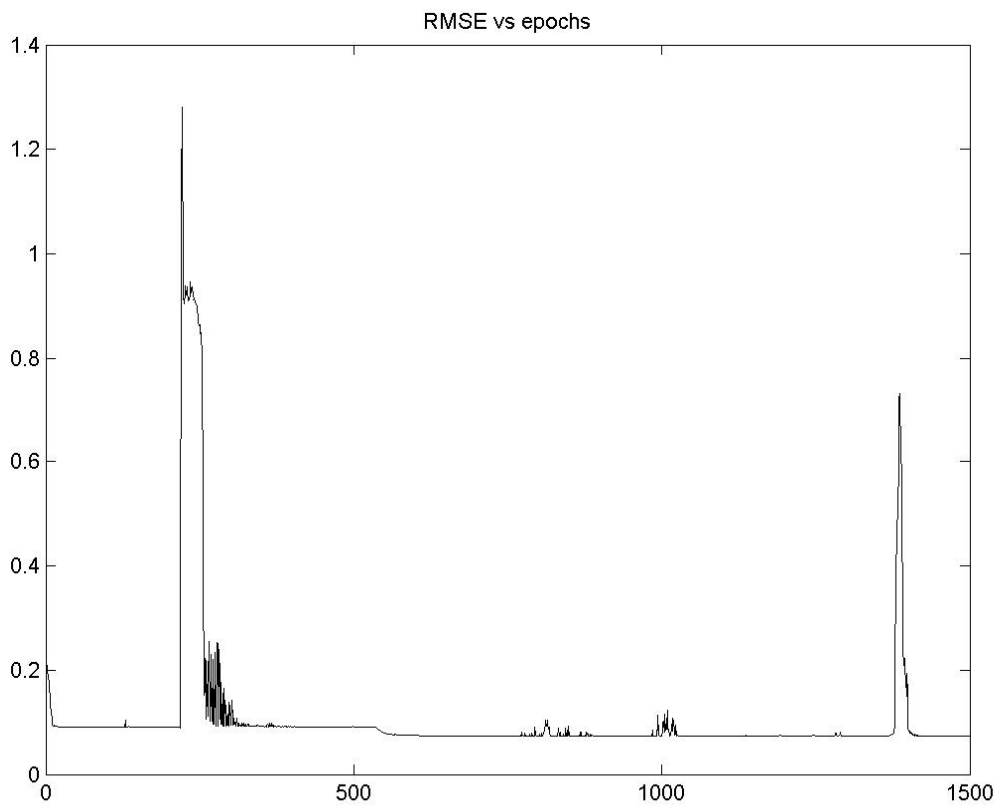
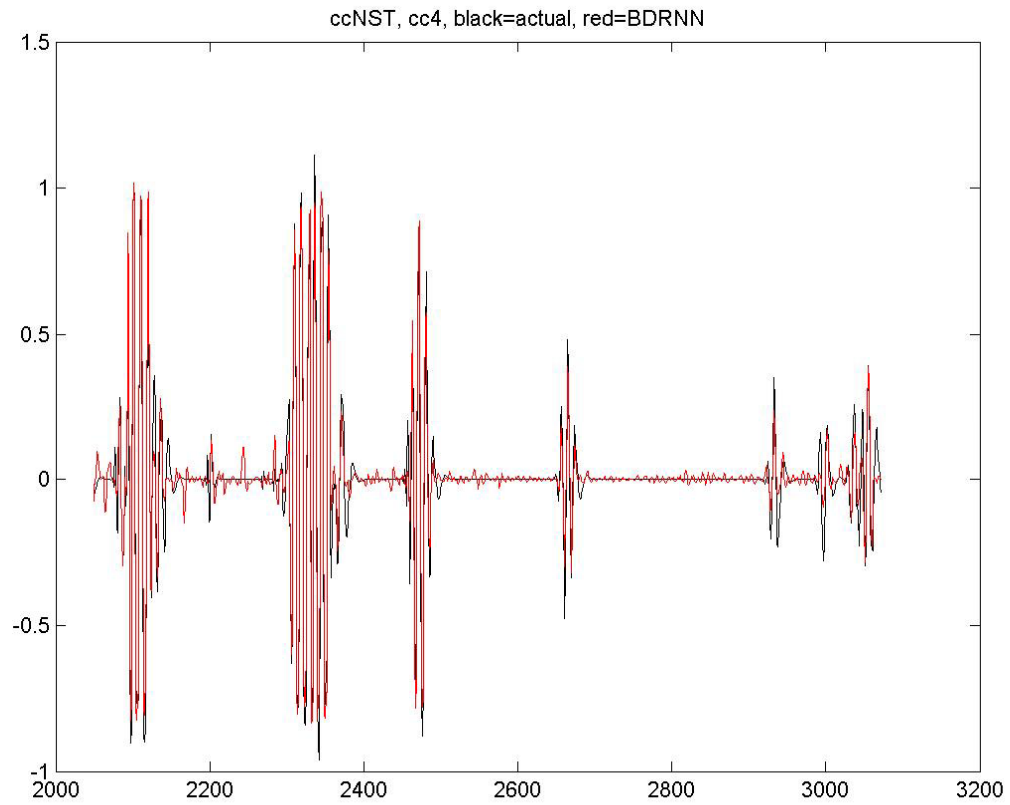
50^η Επανάληψη



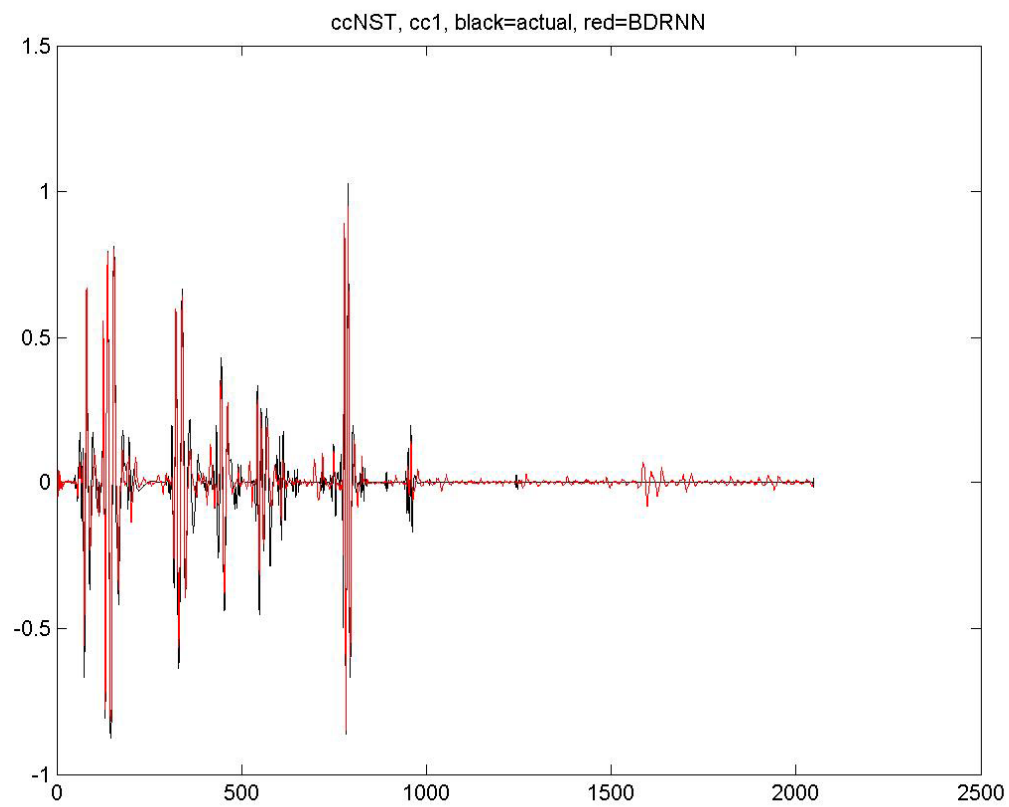
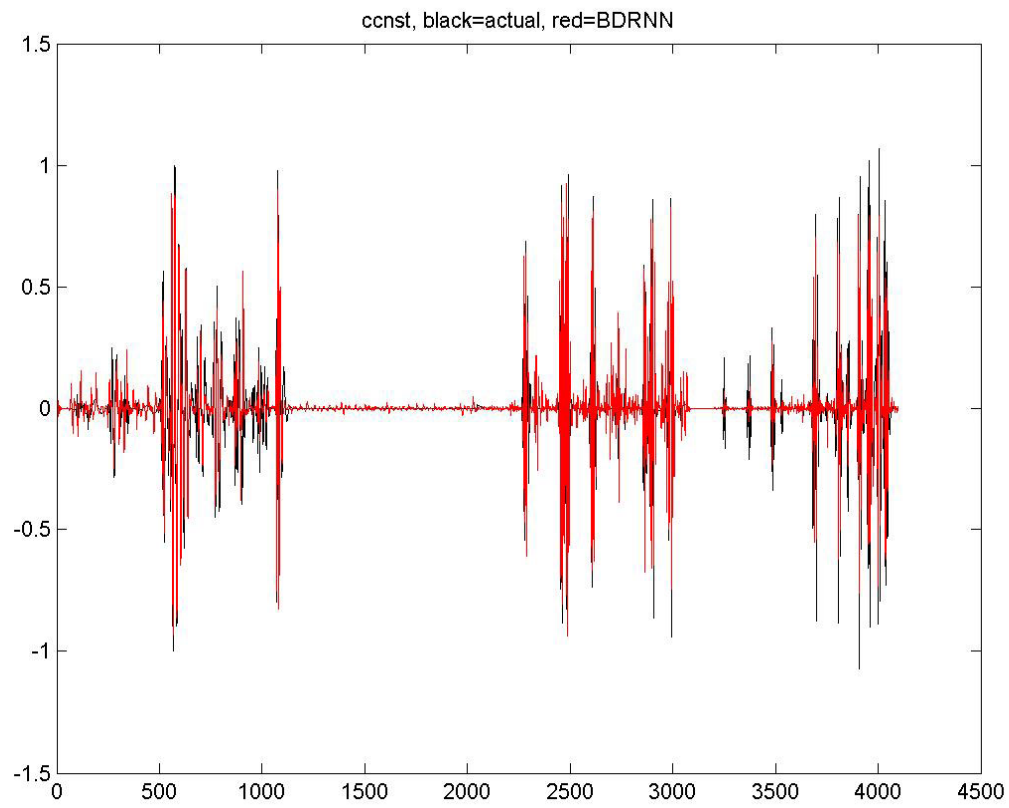


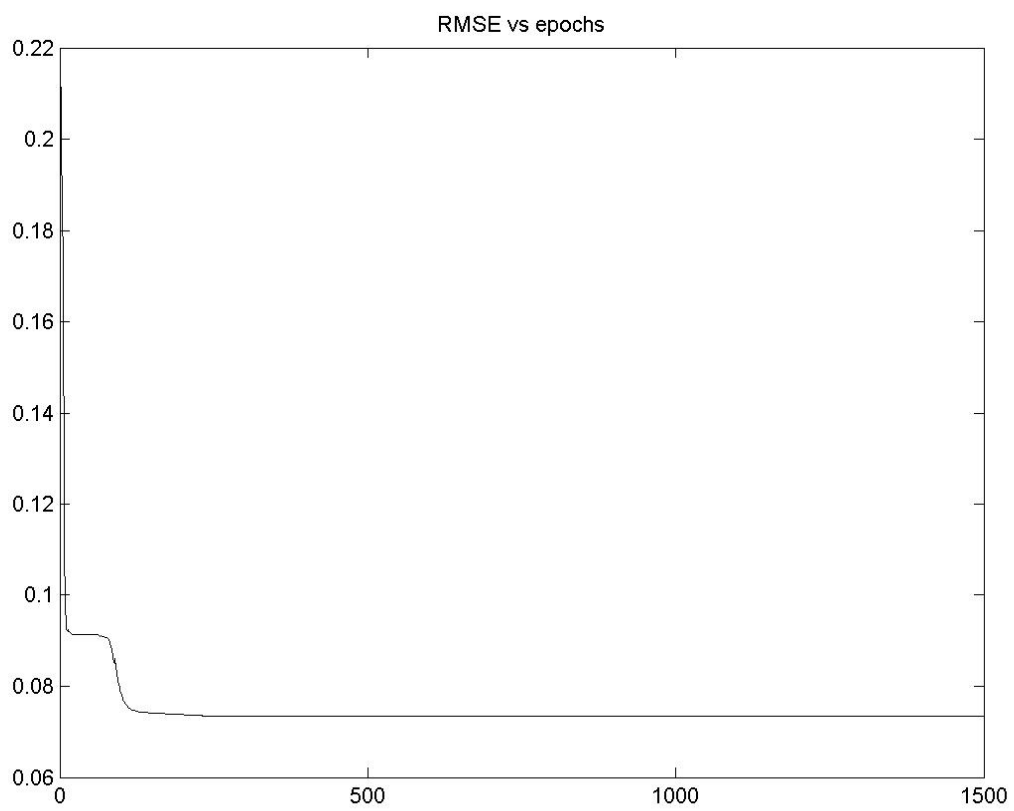
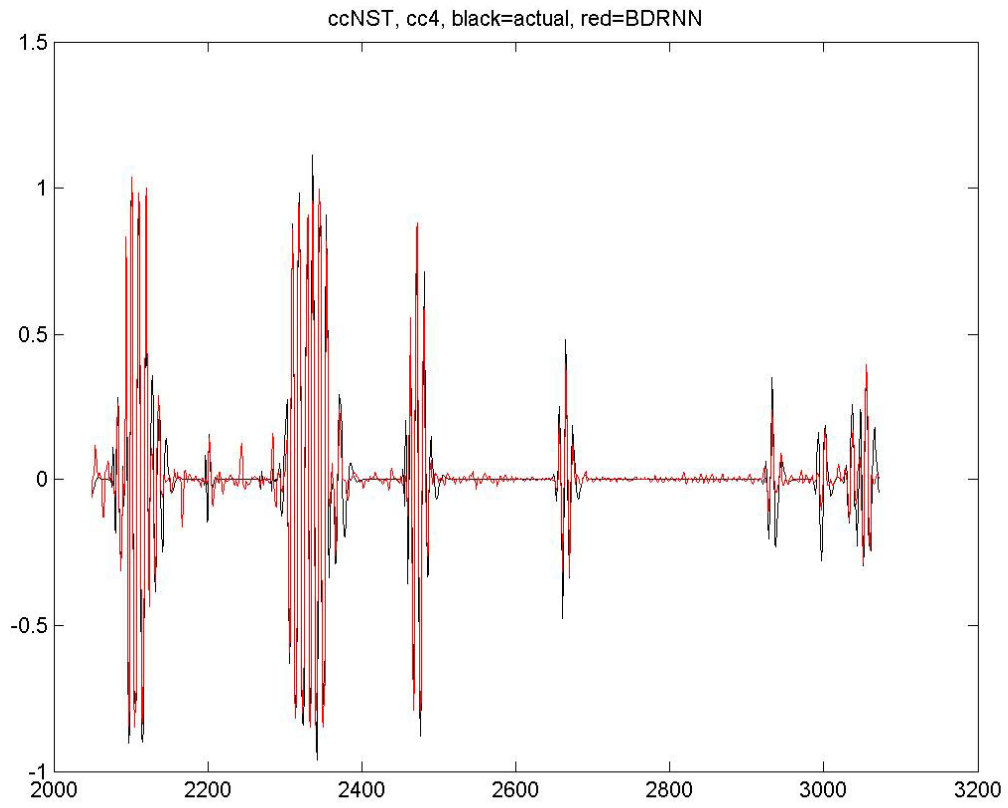
51^η Επανάληψη



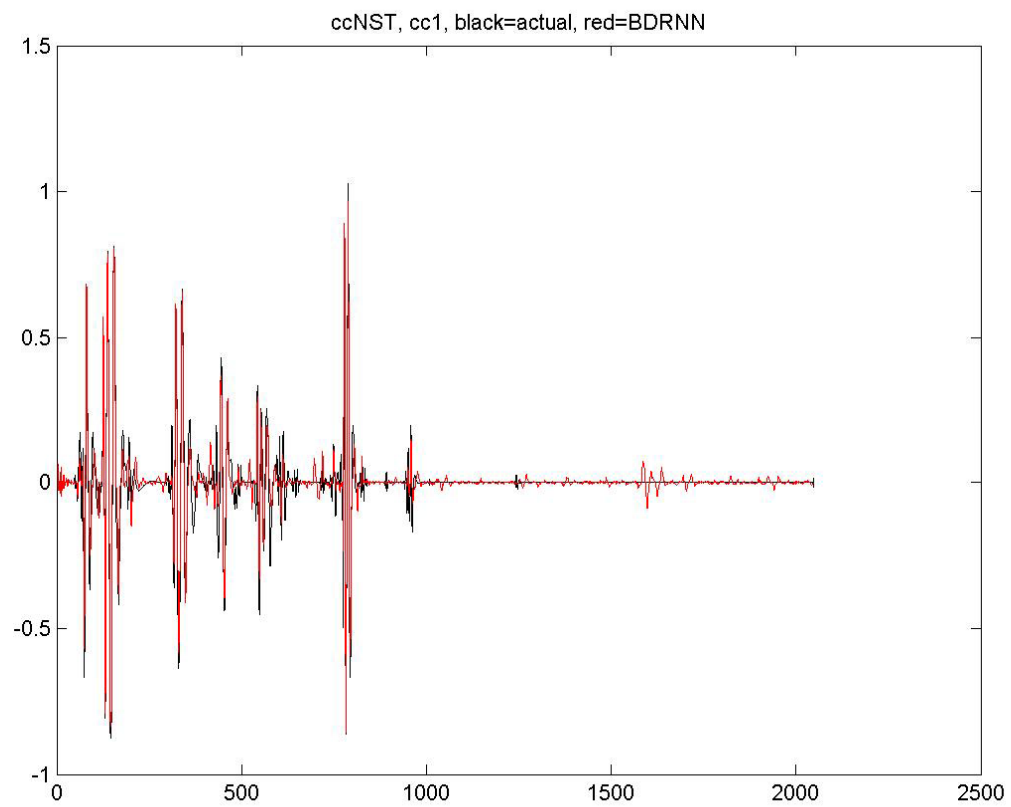
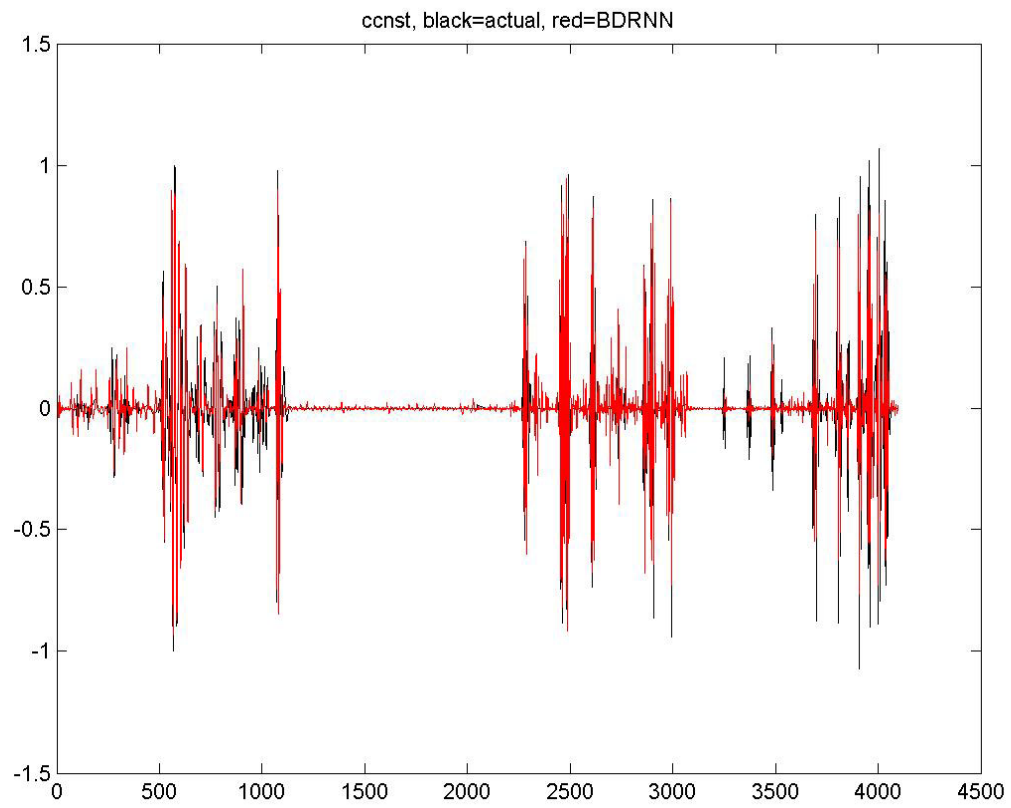


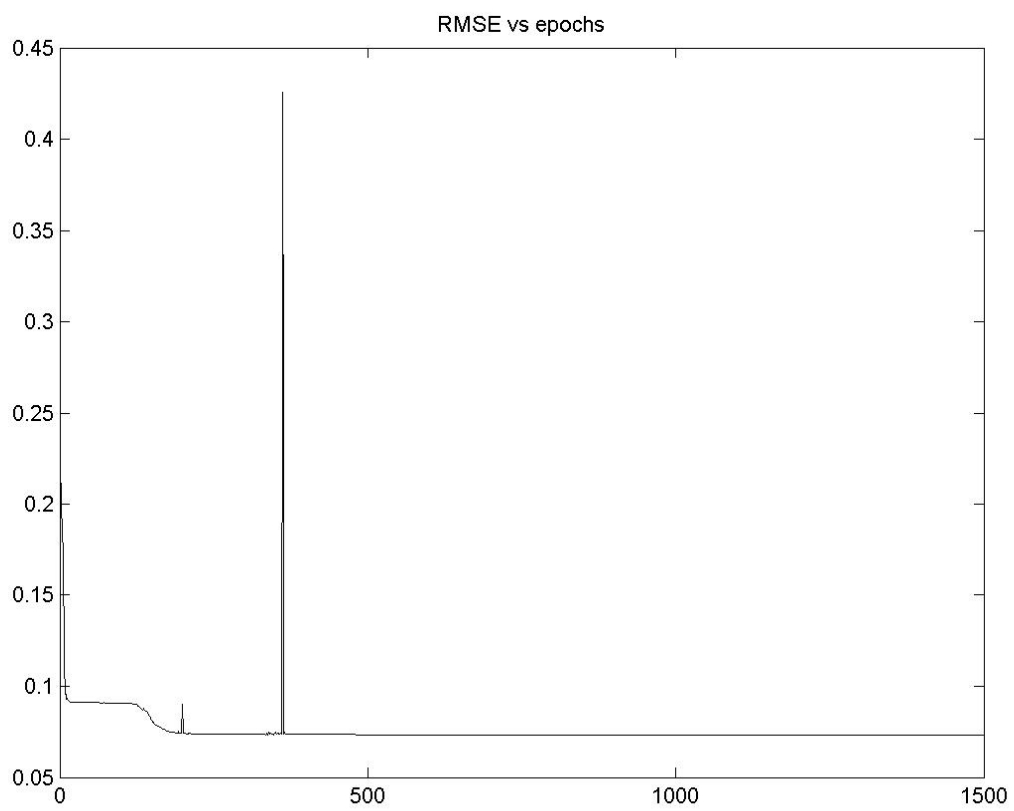
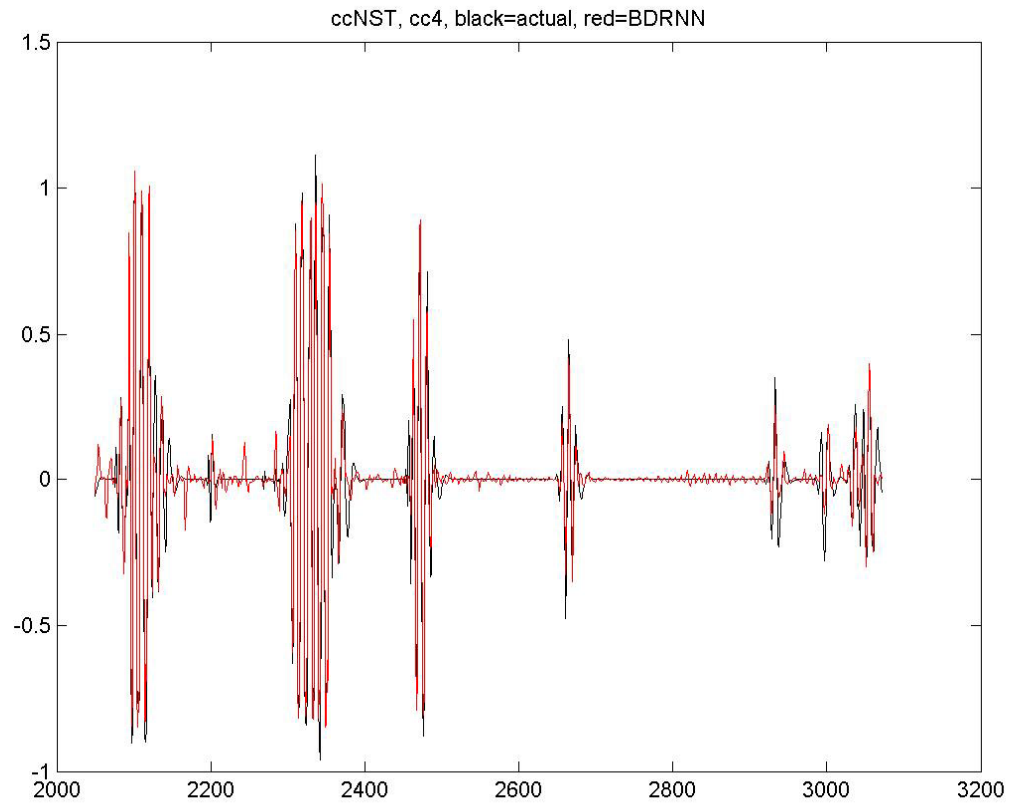
52^η Επανάληψη



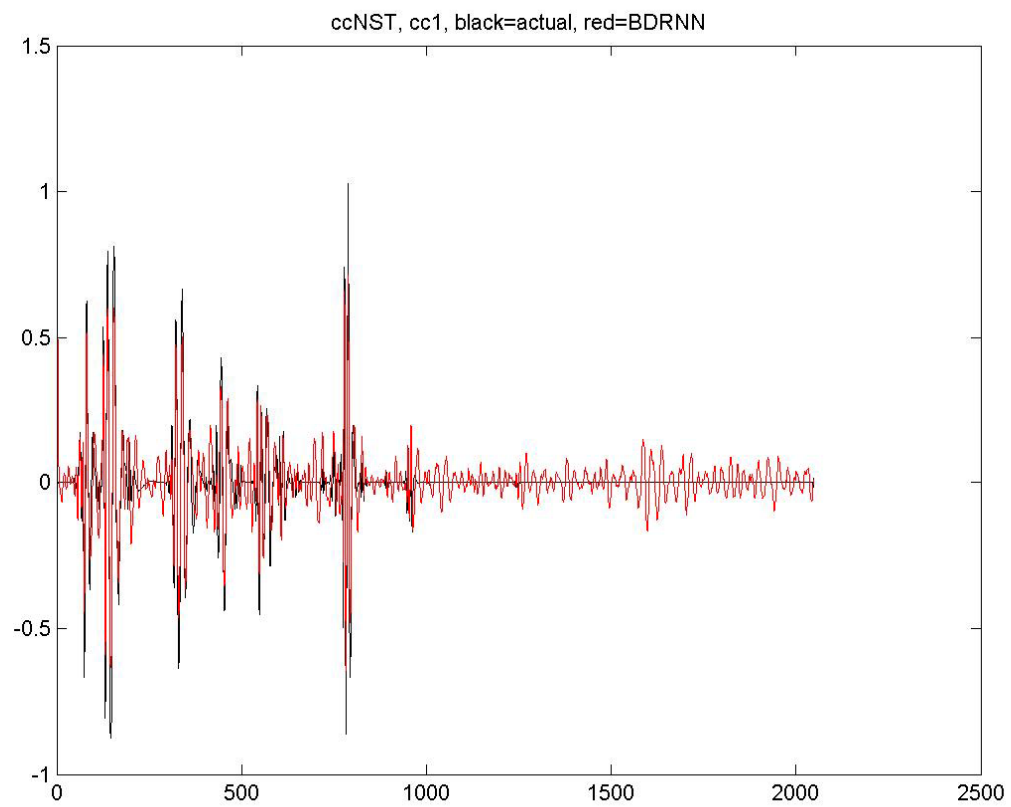
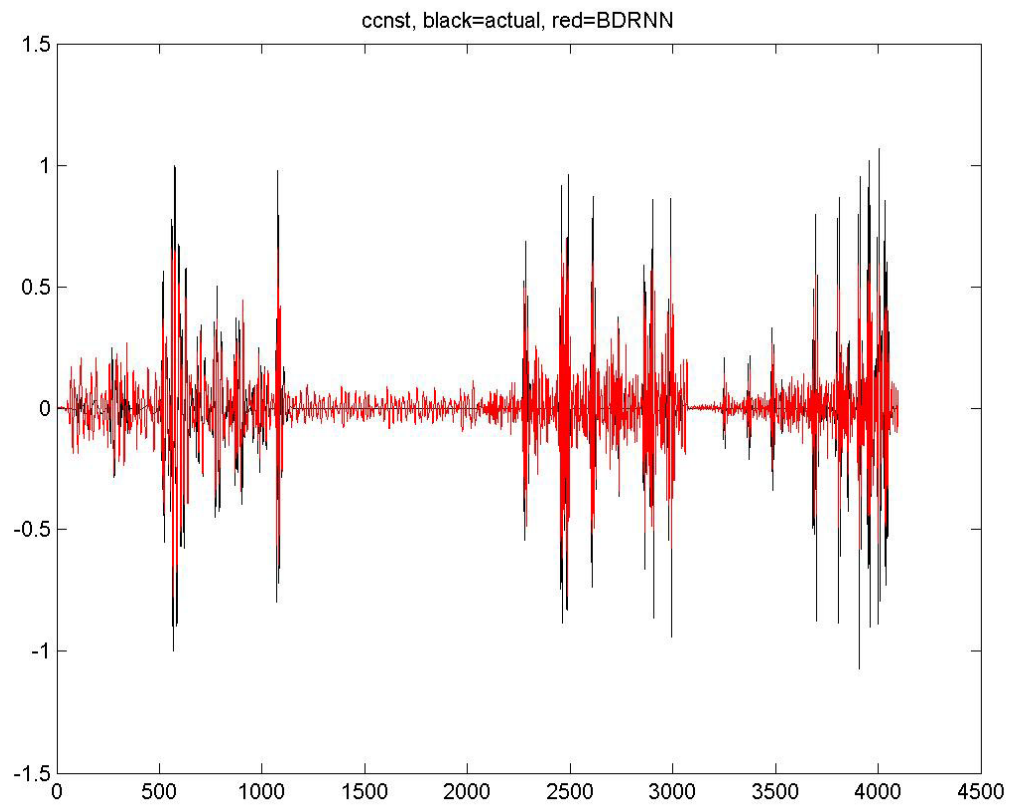


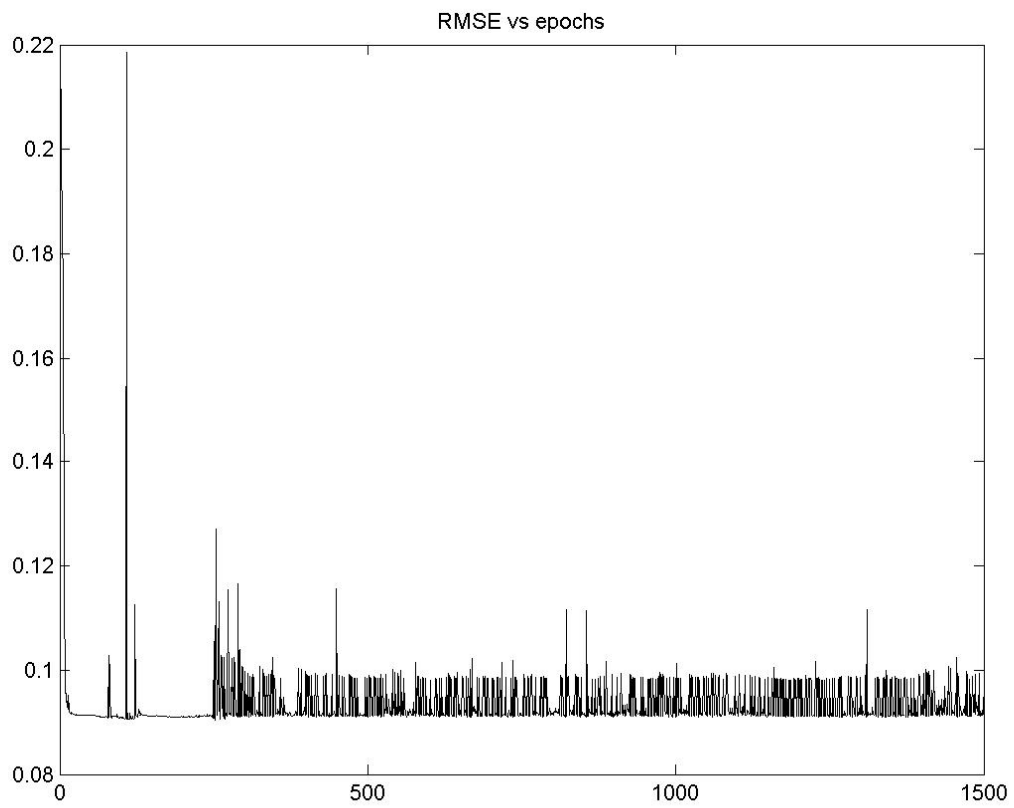
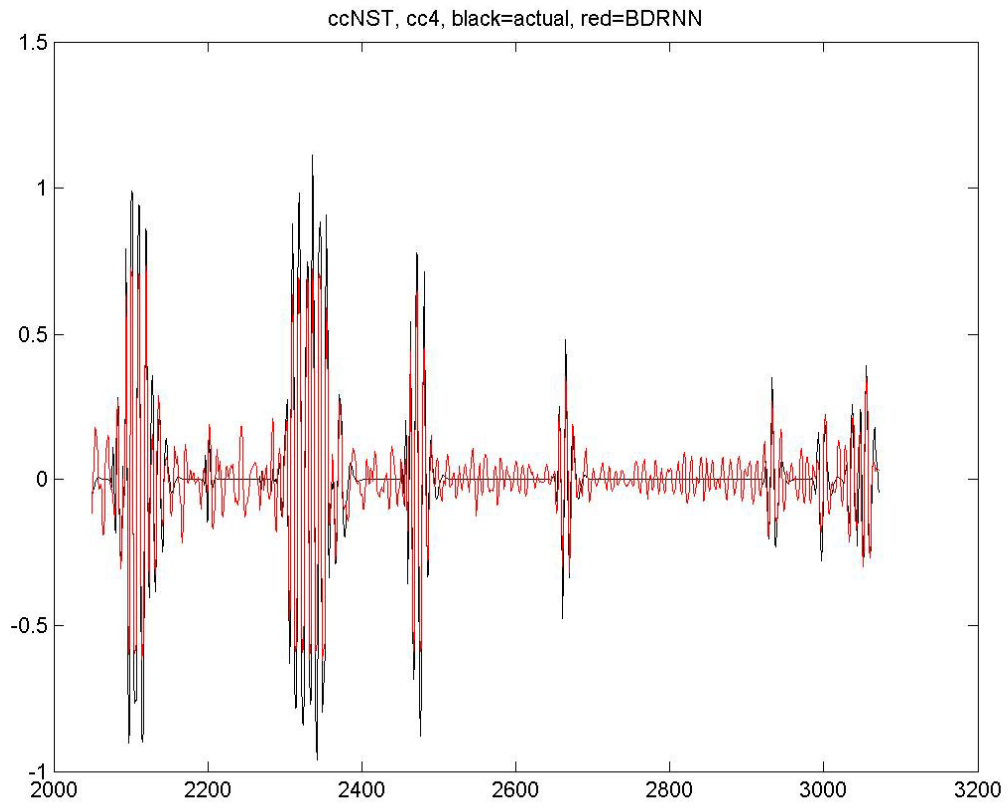
53^η Επανάληψη





54^η Επανάληψη





Πίνακας 3**Συνδυασμός παραμέτρων και αποτέλεσμα για την περίπτωση του πνευμονικού ήχου FC**

Αριθμός Επανάληψης	n_hid	alpha stability	n2	ep_size	n+	n-	max_iter	RMSE (error)
1	12	6	2	4096	1,05	0,5	4000	0,024011
2	12	6	2	4096	1,05	0,7	4000	0,016783
3	12	6	2	4096	1,05	0,9	4000	0,007772
4	12	6	2	4096	1,15	0,5	4000	0,008559
5	12	6	2	4096	1,15	0,7	4000	0,004925
6	12	6	2	4096	1,15	0,9	4000	0,005056
7	12	6	2	4096	1,25	0,5	4000	0,031529
8	12	6	2	4096	1,25	0,7	4000	0,006423
9	12	6	2	4096	1,25	0,9	4000	0,005312
10	12	6	6	4096	1,05	0,5	4000	0,010835
11	12	6	6	4096	1,05	0,7	4000	0,033348
12	12	6	6	4096	1,05	0,9	4000	0,067793
13	12	6	6	4096	1,15	0,5	4000	0,047044
14	12	6	6	4096	1,15	0,7	4000	0,014615
15	12	6	6	4096	1,15	0,9	4000	0,008118
16	12	6	6	4096	1,25	0,5	4000	0,010157
17	12	6	6	4096	1,25	0,7	4000	0,025747
18	12	6	6	4096	1,25	0,9	4000	0,025467
19	18	9	2	4096	1,05	0,5	4000	0,008908
20	18	9	2	4096	1,05	0,7	4000	0,016994
21	18	9	2	4096	1,05	0,9	4000	0,023601
22	18	9	2	4096	1,15	0,5	4000	0,010386
23	18	9	2	4096	1,15	0,7	4000	0,006211
24	18	9	2	4096	1,15	0,9	4000	0,012316
25	18	9	2	4096	1,25	0,5	4000	0,042614
26	18	9	2	4096	1,25	0,7	4000	0,011439
27	18	9	2	4096	1,25	0,9	4000	0,045924
28	18	9	6	4096	1,05	0,5	4000	0,004999
29	18	9	6	4096	1,05	0,7	4000	0,020116
30	18	9	6	4096	1,05	0,9	4000	0,006553
31	18	9	6	4096	1,15	0,5	4000	0,019307
32	18	9	6	4096	1,15	0,7	4000	0,010528
33	18	9	6	4096	1,15	0,9	4000	0,011902
34	18	9	6	4096	1,25	0,5	4000	0,015421
35	18	9	6	4096	1,25	0,7	4000	0,014010
36	18	9	6	4096	1,25	0,9	4000	0,041839
37	24	12	2	4096	1,05	0,5	4000	0,039648
38	24	12	2	4096	1,05	0,7	4000	0,033219
39	24	12	2	4096	1,05	0,9	4000	0,015429
40	24	12	2	4096	1,15	0,5	4000	0,053026
41	24	12	2	4096	1,15	0,7	4000	0,024571
42	24	12	2	4096	1,15	0,9	4000	0,042050
43	24	12	2	4096	1,25	0,5	4000	0,027010
44	24	12	2	4096	1,25	0,7	4000	0,009045
45	24	12	2	4096	1,25	0,9	4000	0,024411
46	24	12	6	4096	1,05	0,5	4000	0,035346
47	24	12	6	4096	1,05	0,7	4000	0,005709
48	24	12	6	4096	1,05	0,9	4000	0,006832
49	24	12	6	4096	1,15	0,5	4000	0,006256

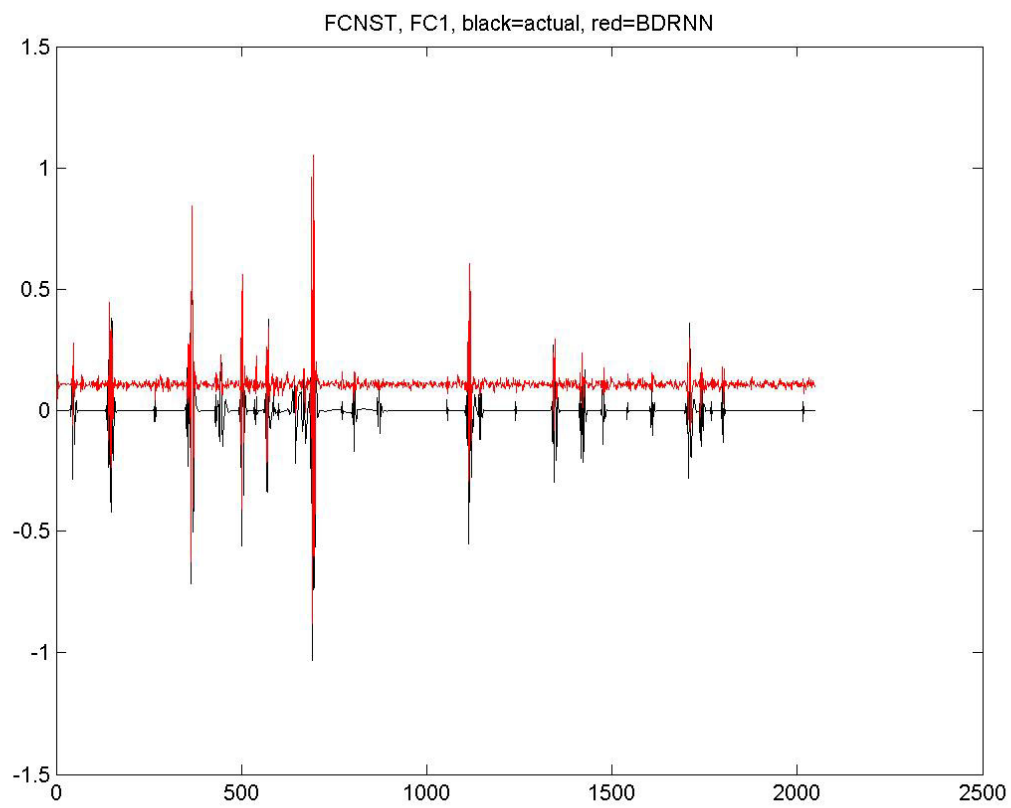
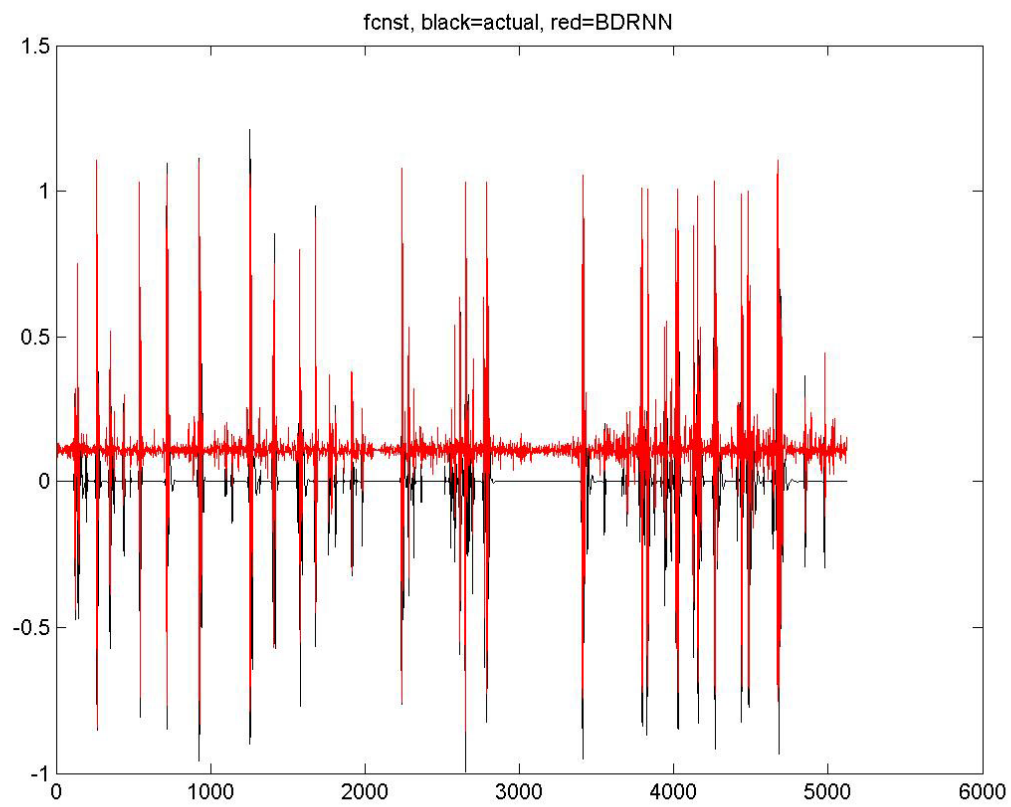
50	24	12	6	4096	1,15	0,7	4000	0,005130
51	24	12	6	4096	1,15	0,9	4000	0,029521
52	24	12	6	4096	1,25	0,5	4000	0,015183
53	24	12	6	4096	1,25	0,7	4000	0,018102
54	24	12	6	4096	1,25	0,9	4000	0,027772

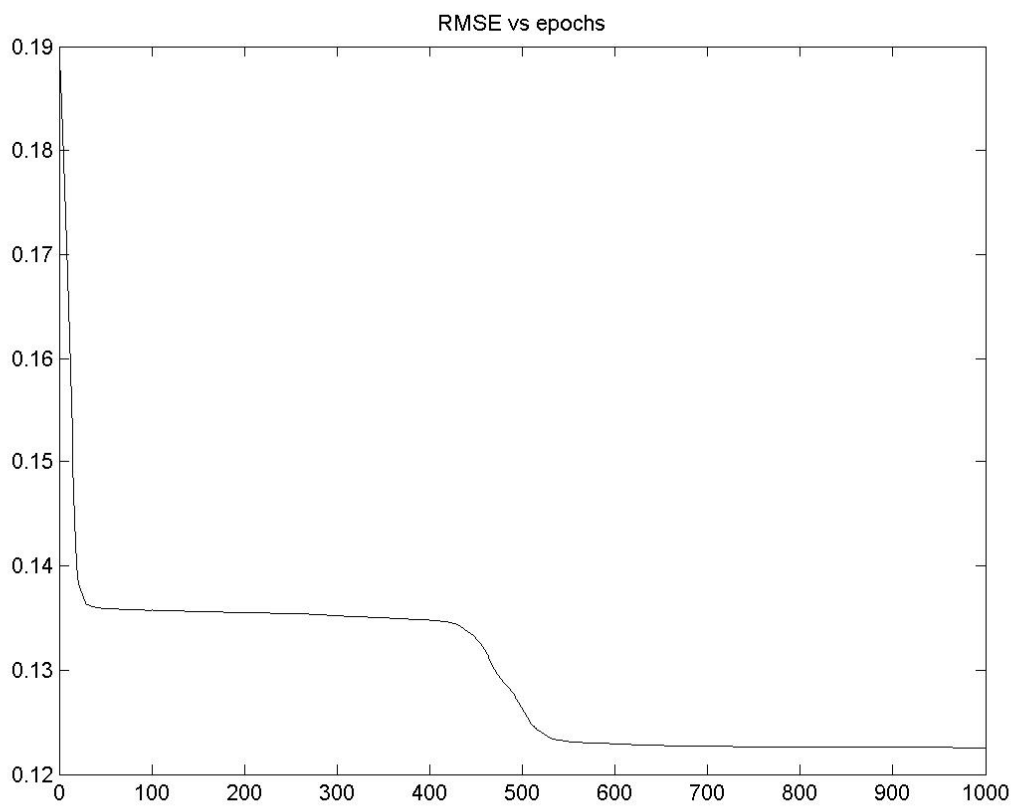
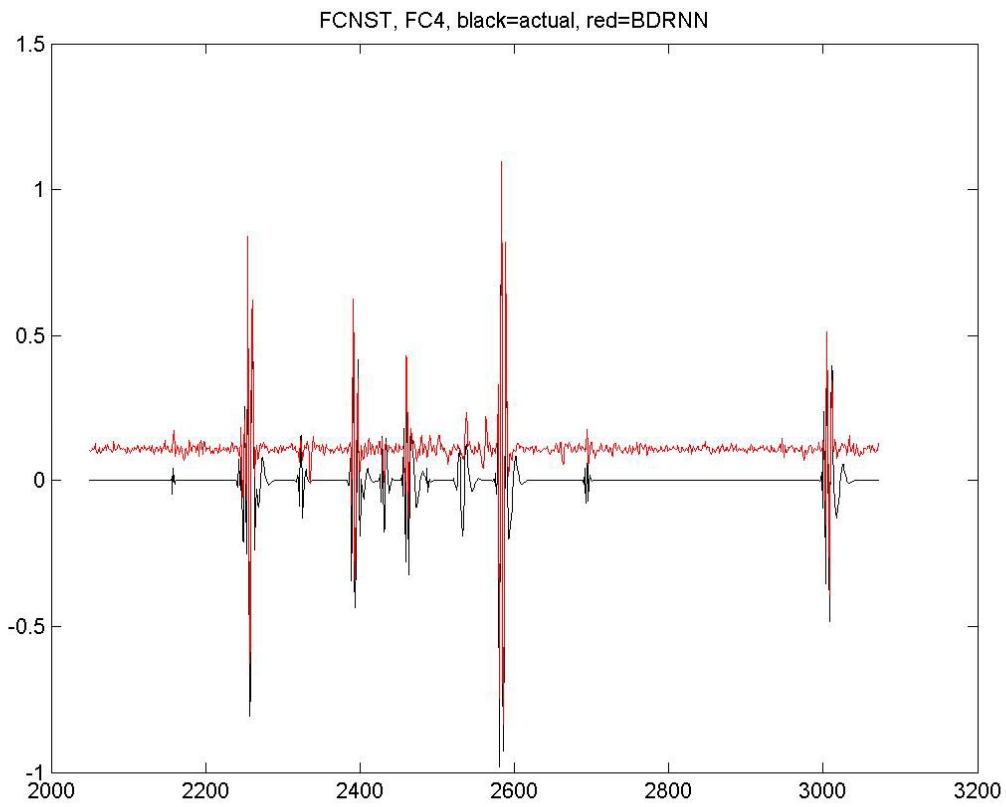
Πίνακας 3.1

Ελάχιστο MSE	Μέγιστο MSE	Μέσος Όρος	Τυπική Απόκλιση	(Τυπική Απόκλιση / Μέσος Όρος)*100
0,01413	0,037767	0,020091	0,005374	26,74813 %

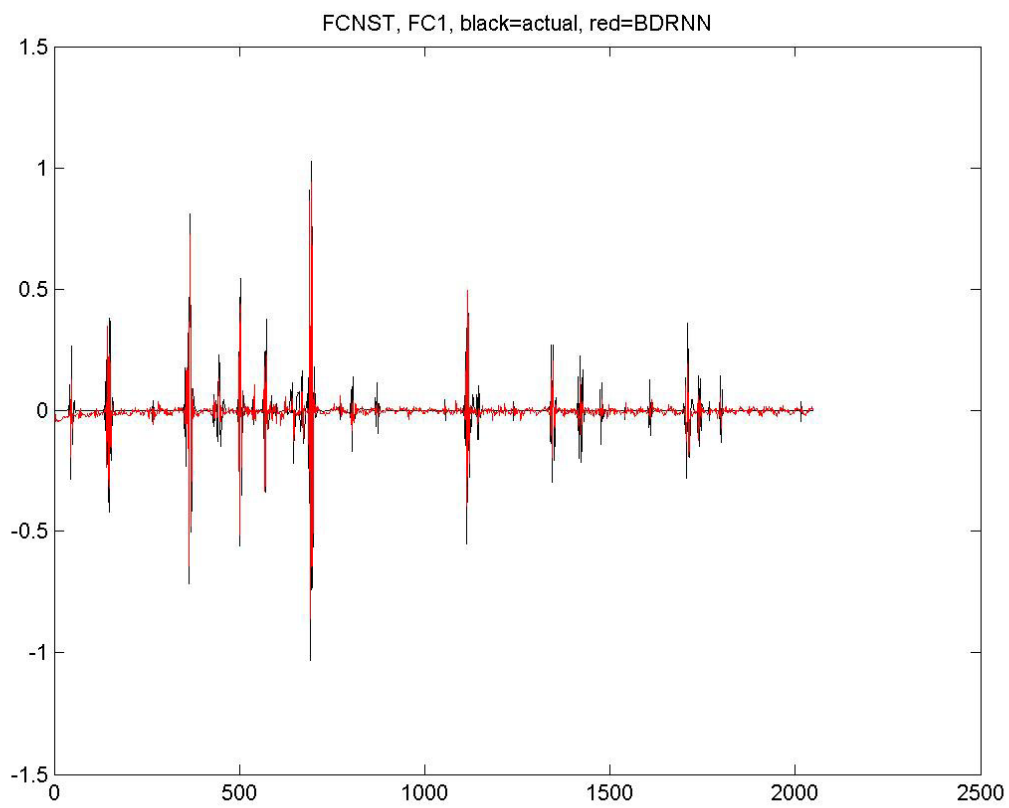
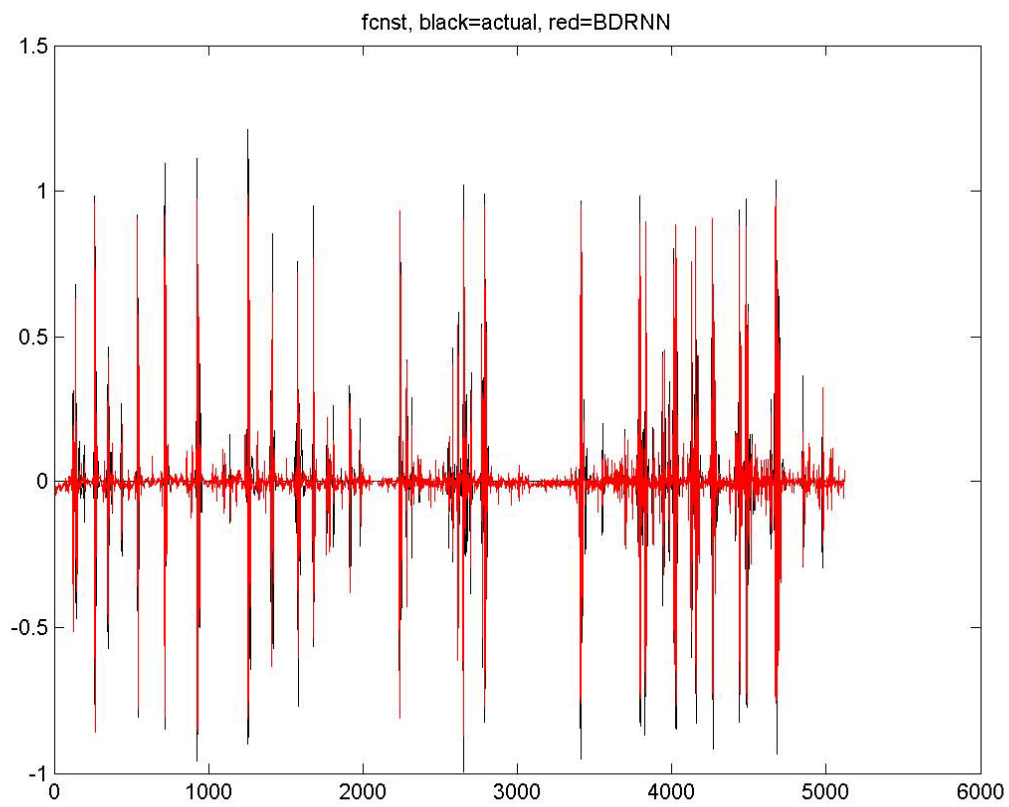
Στις επόμενες σελίδες ακολουθούν οι γραφικές απεικονίσεις των αποτελεσμάτων για κάθε ένα από τους 54 συνδυασμούς παραμέτρων εκπαίδευσης που εκτελέστηκαν για την περίπτωση του πνευμονικού ήχου FC.

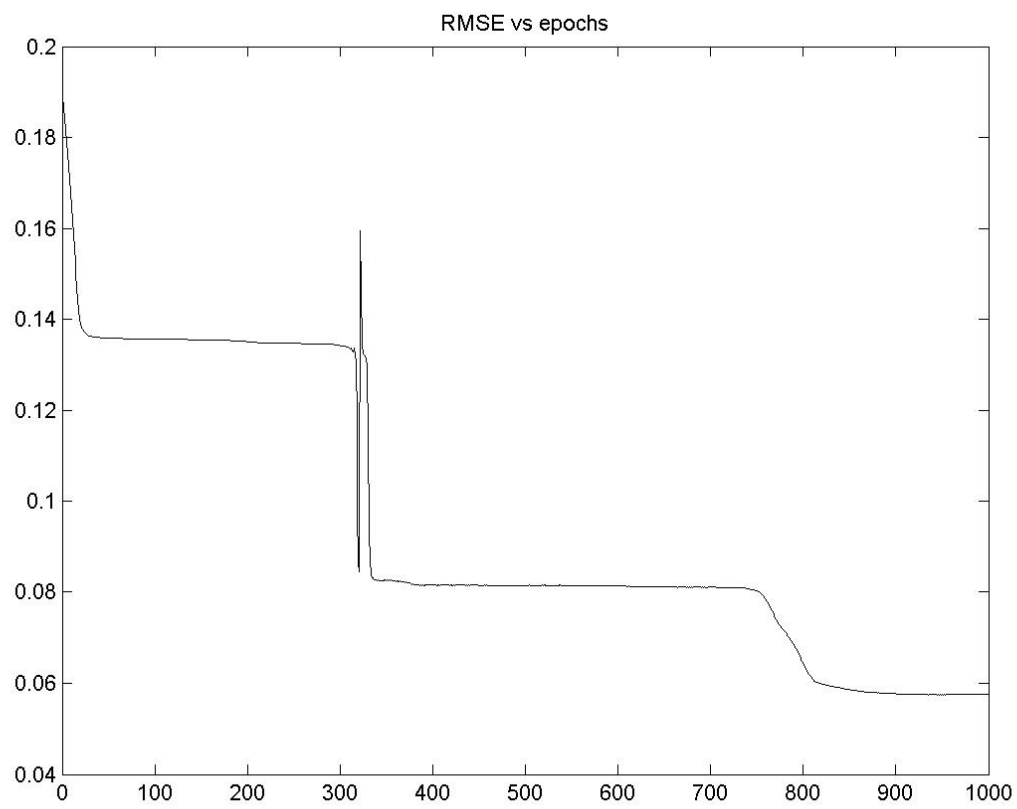
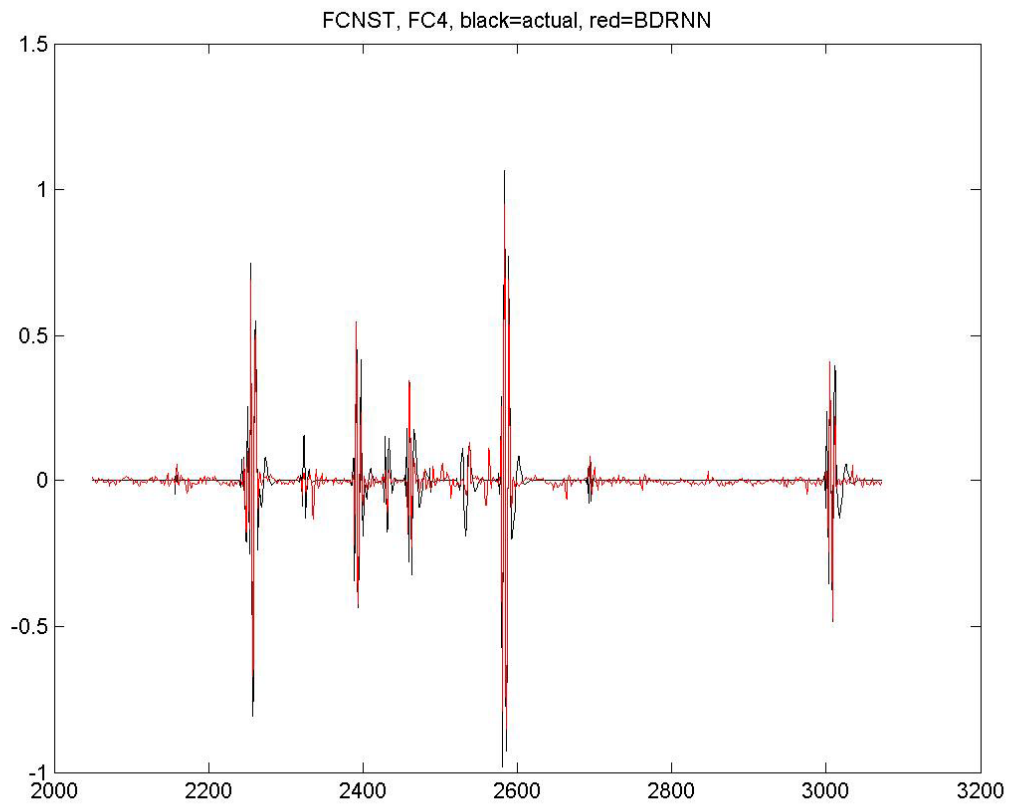
1^η Επανάληψη



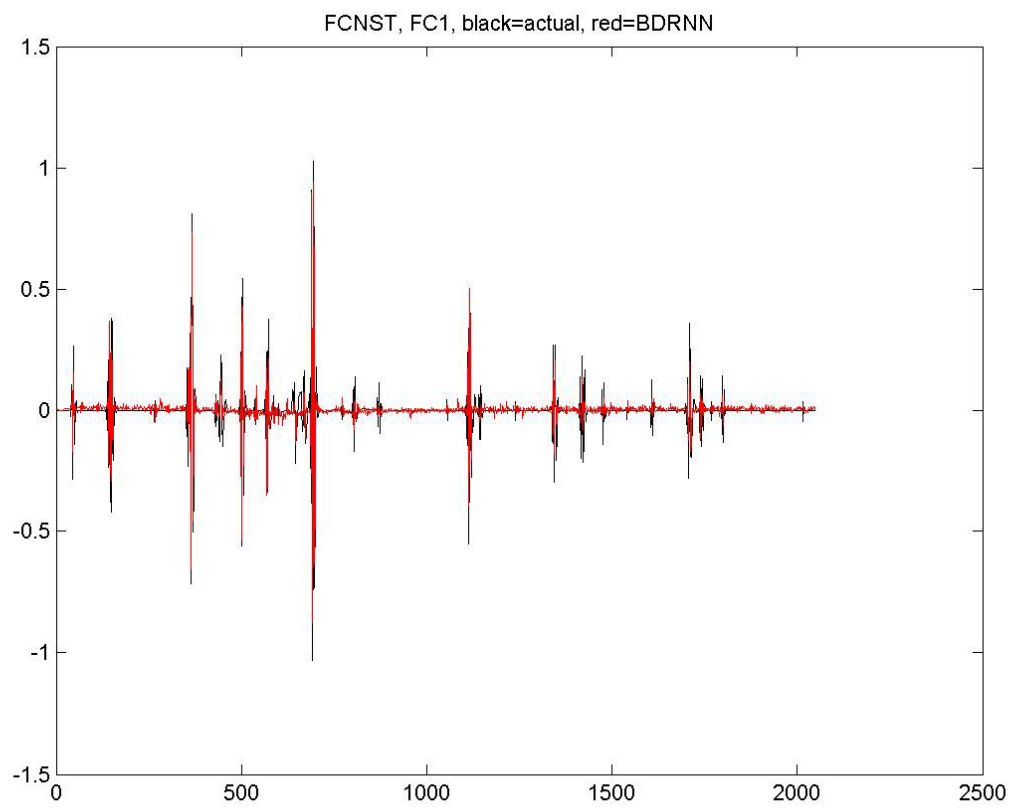
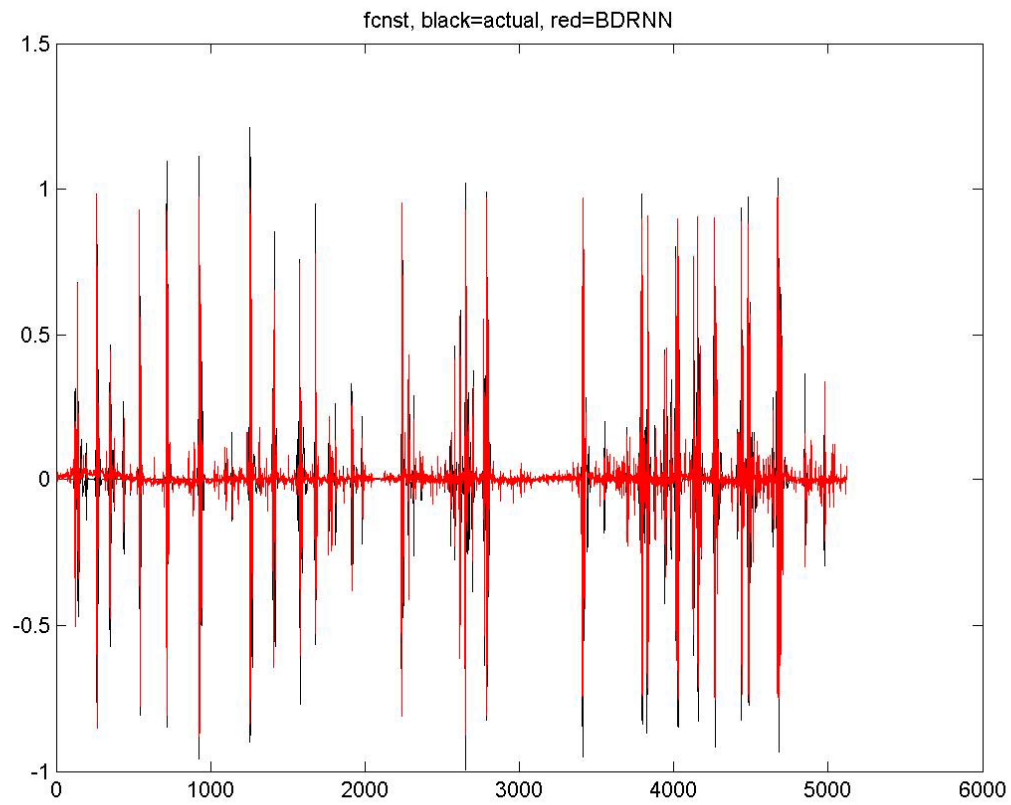


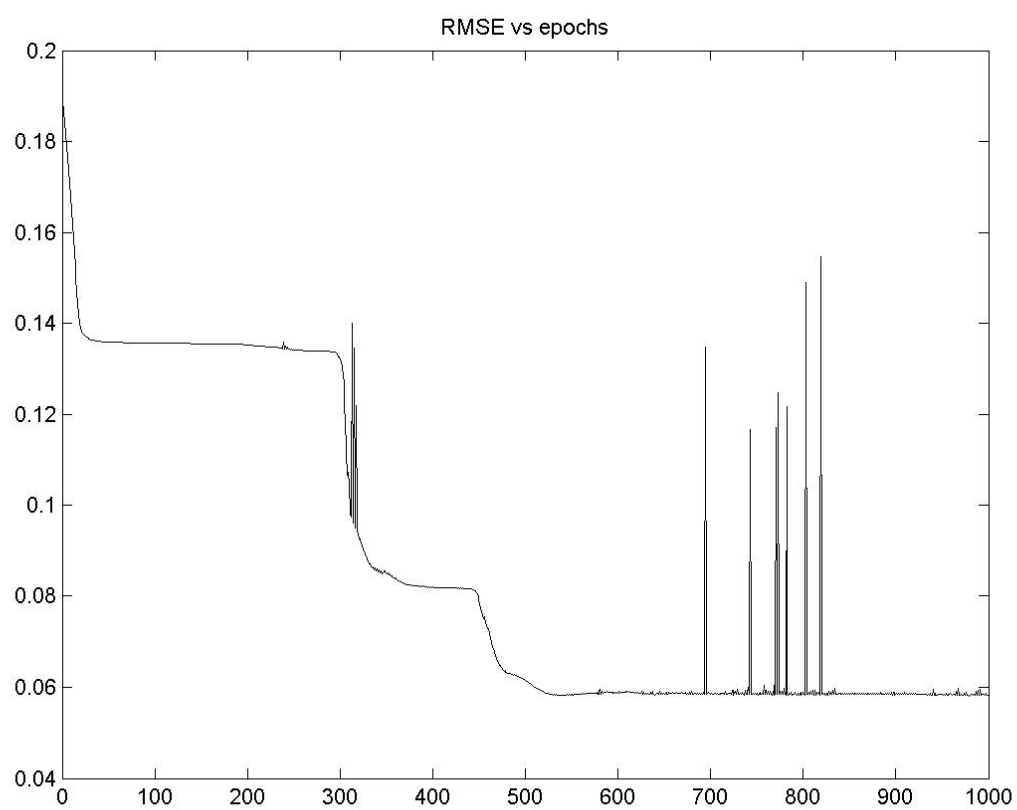
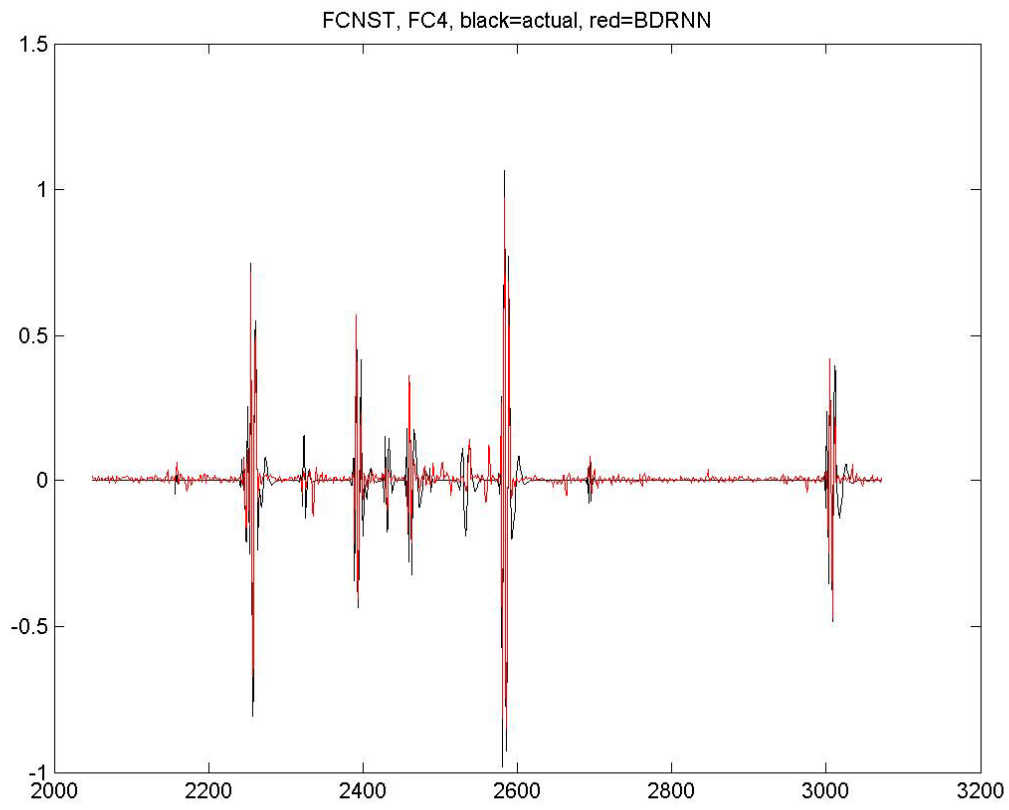
2^η Επαναληψη



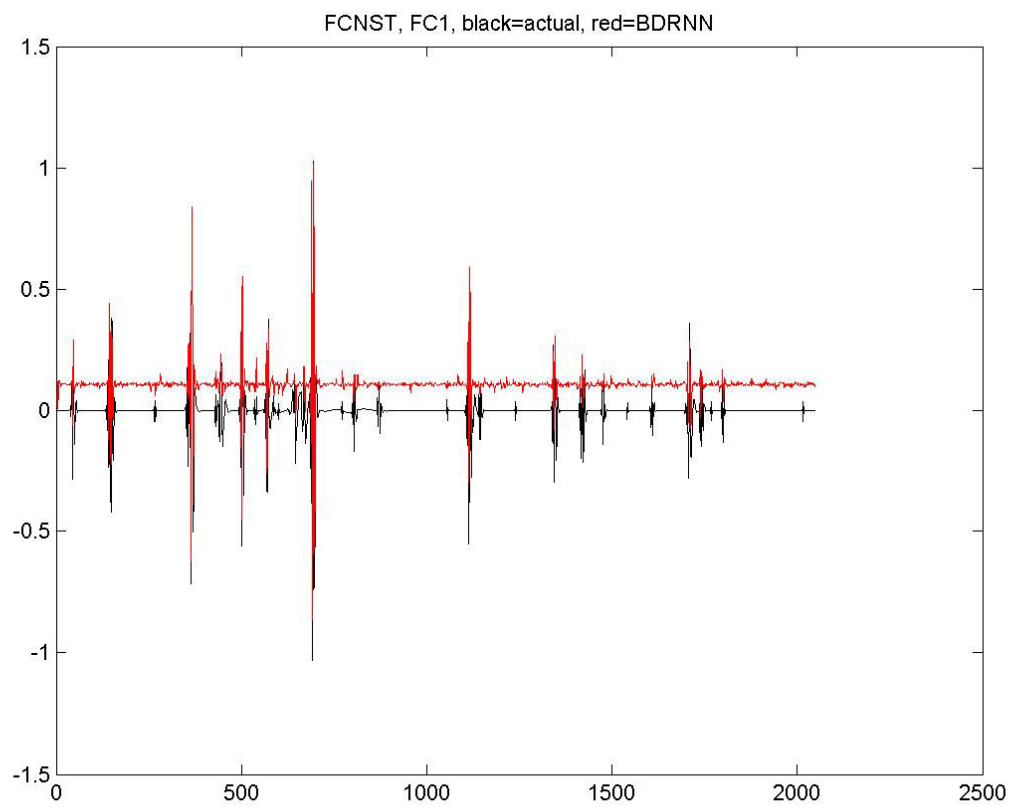
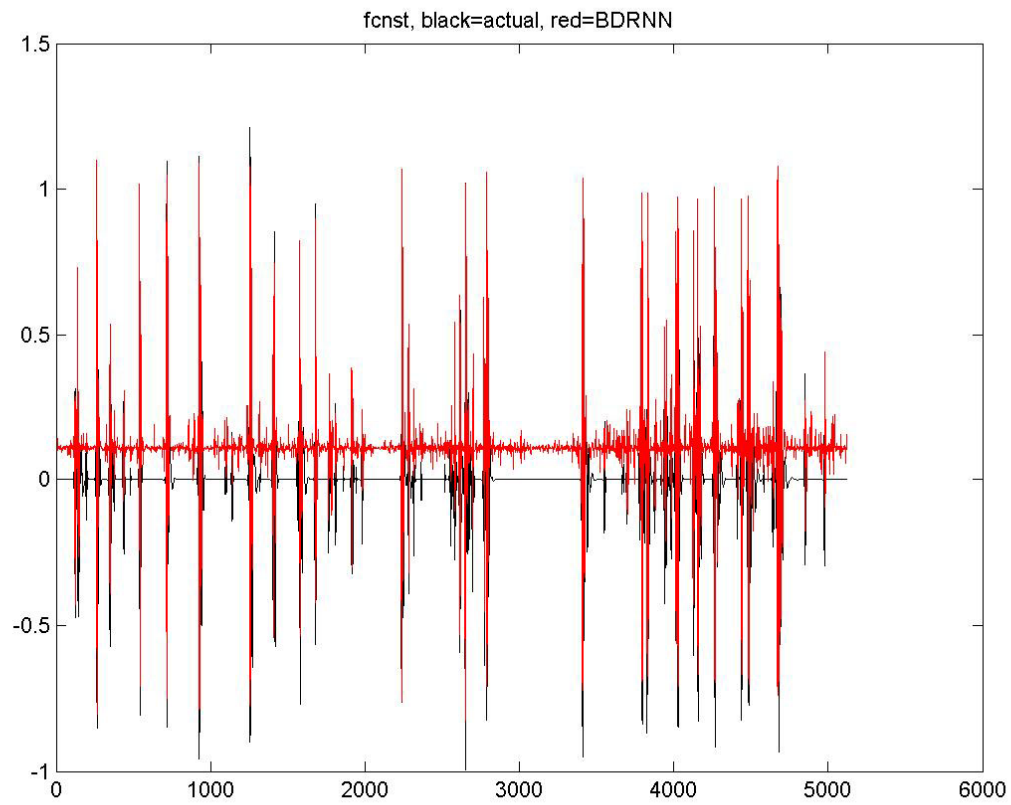


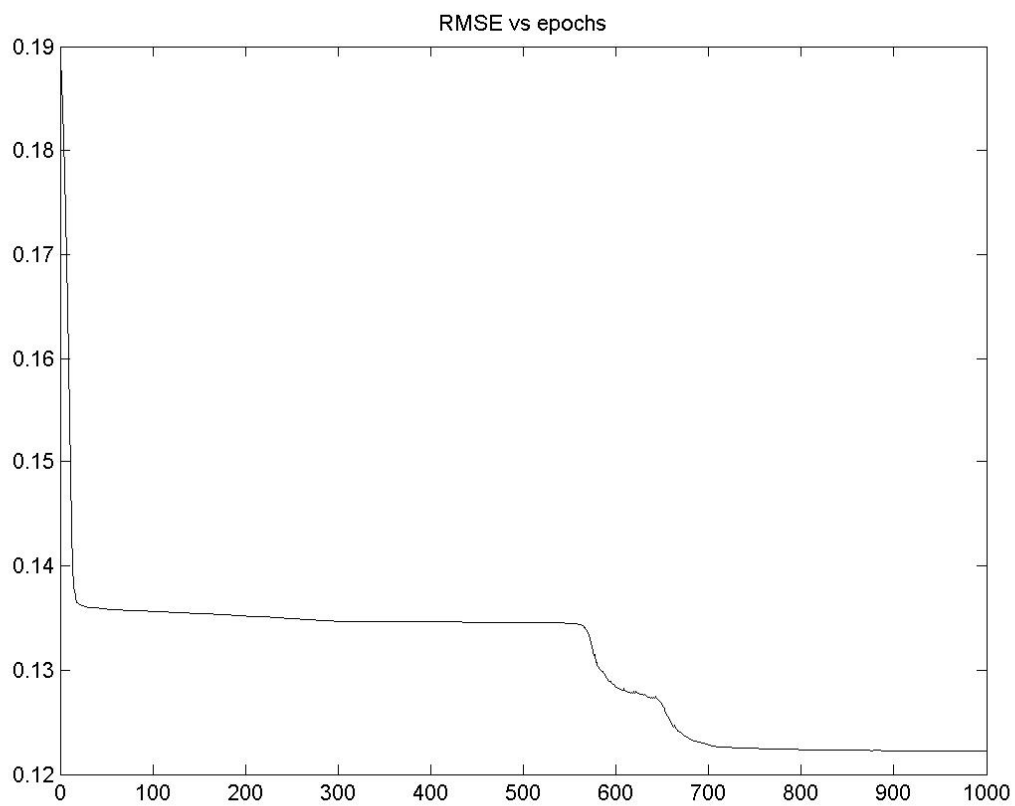
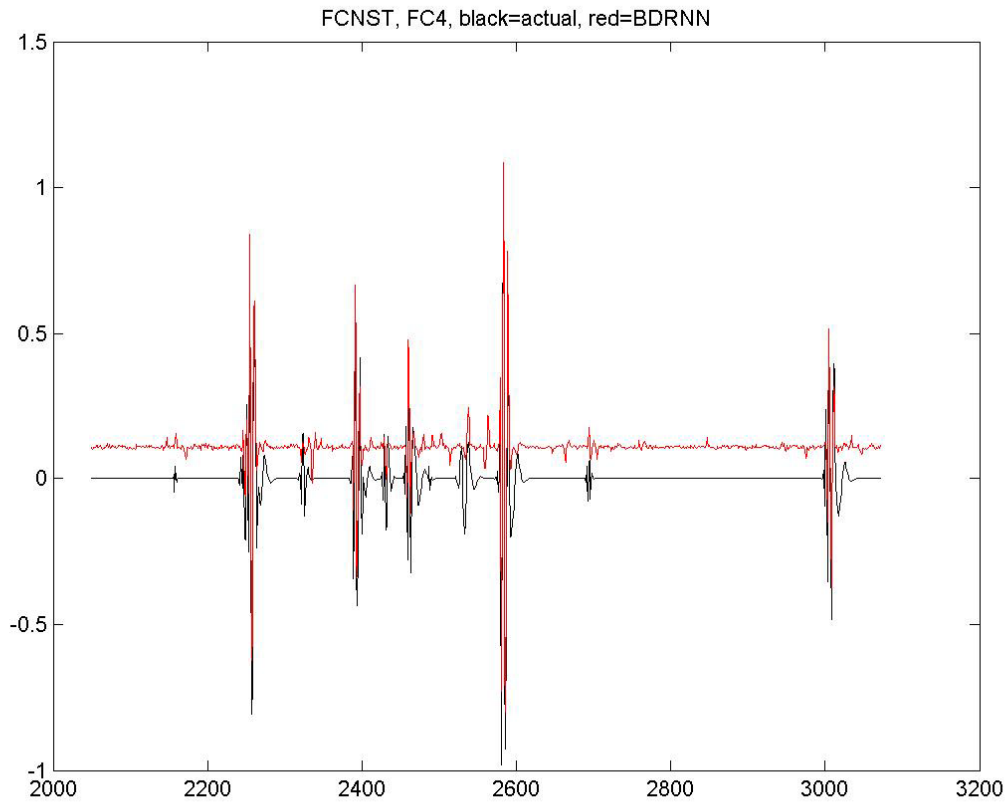
3^η Επανάληψη



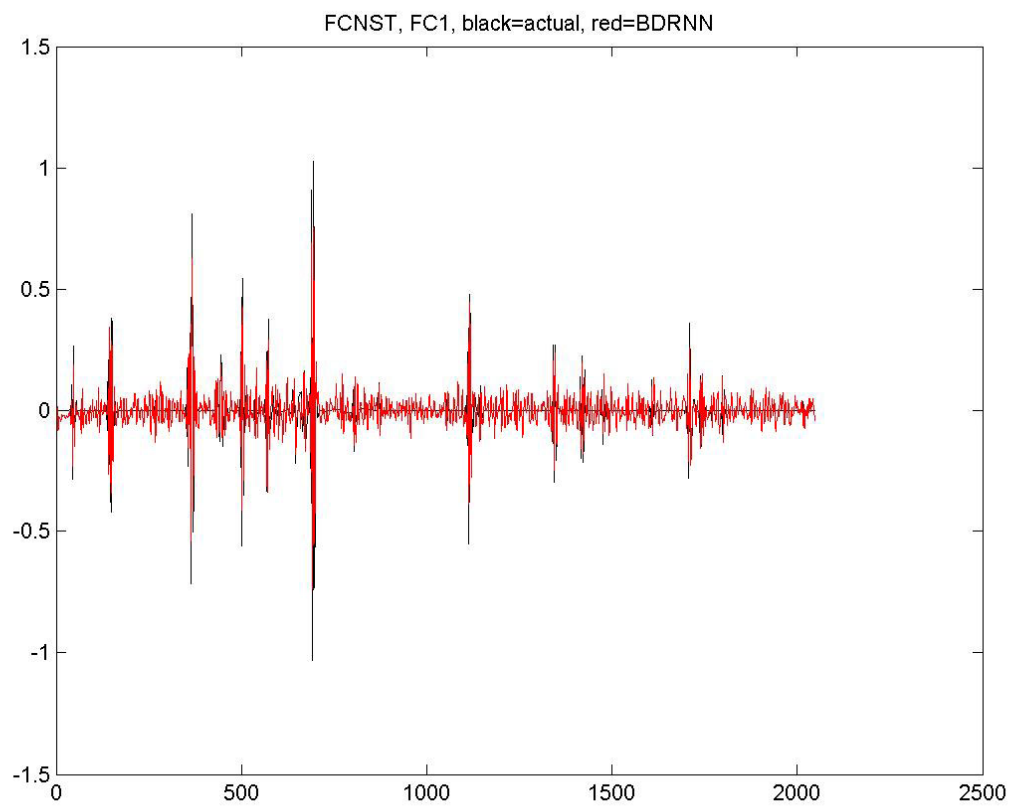
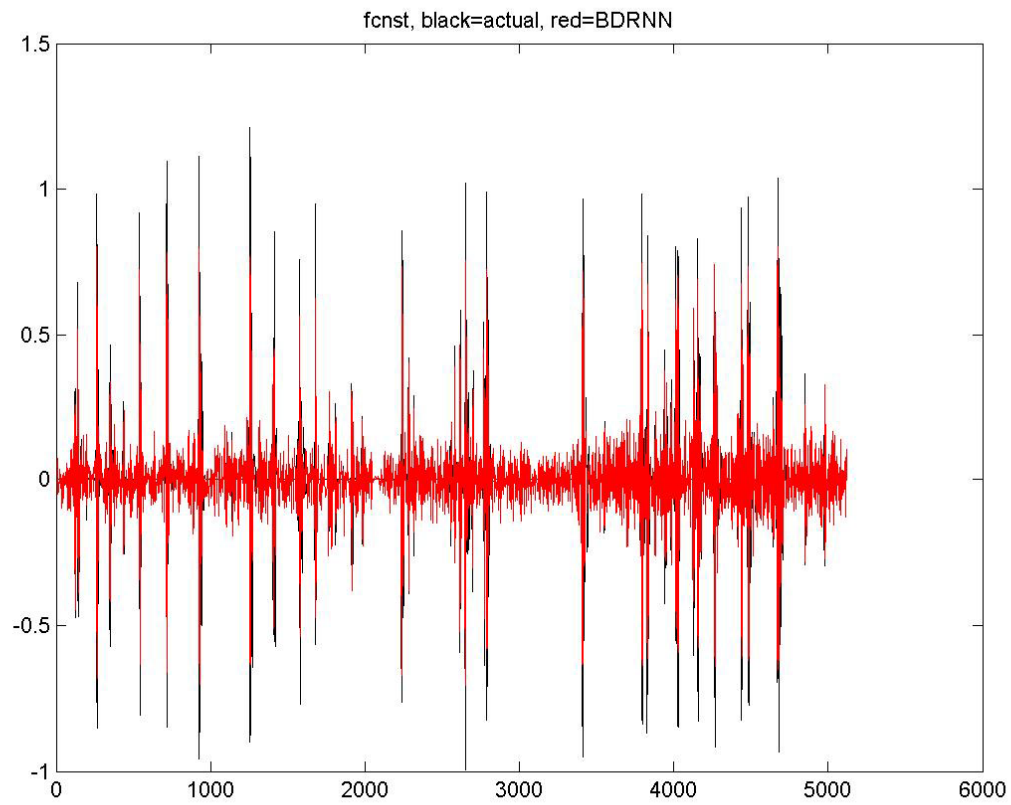


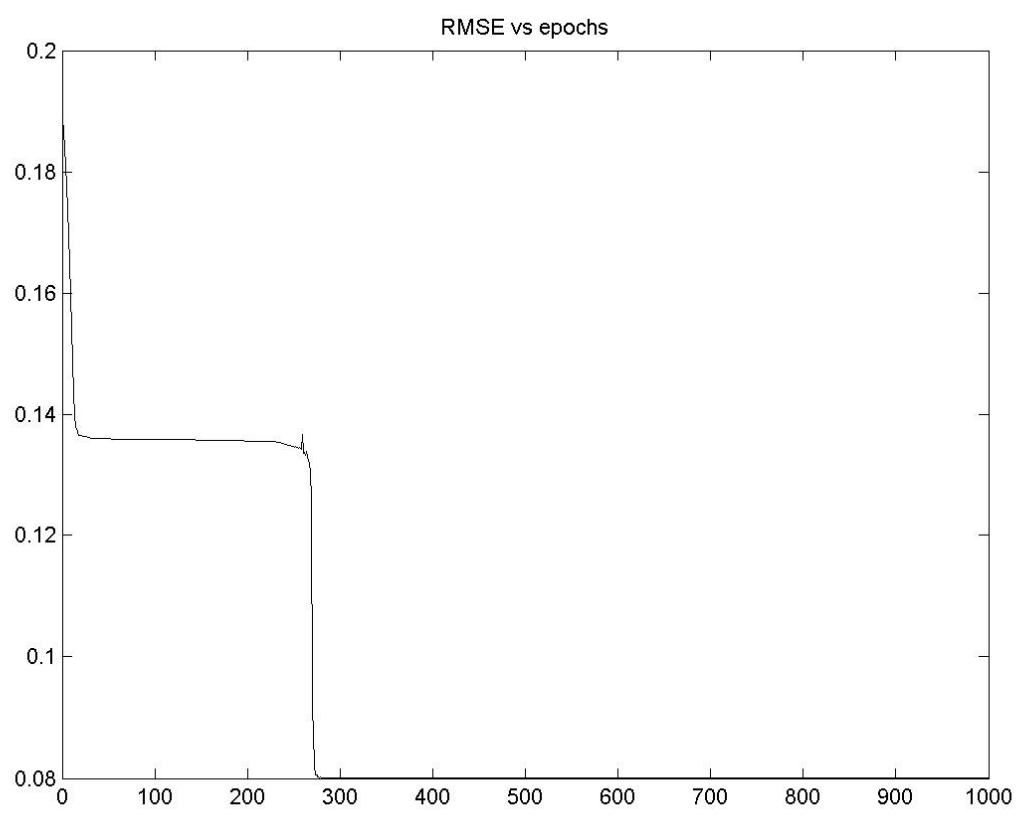
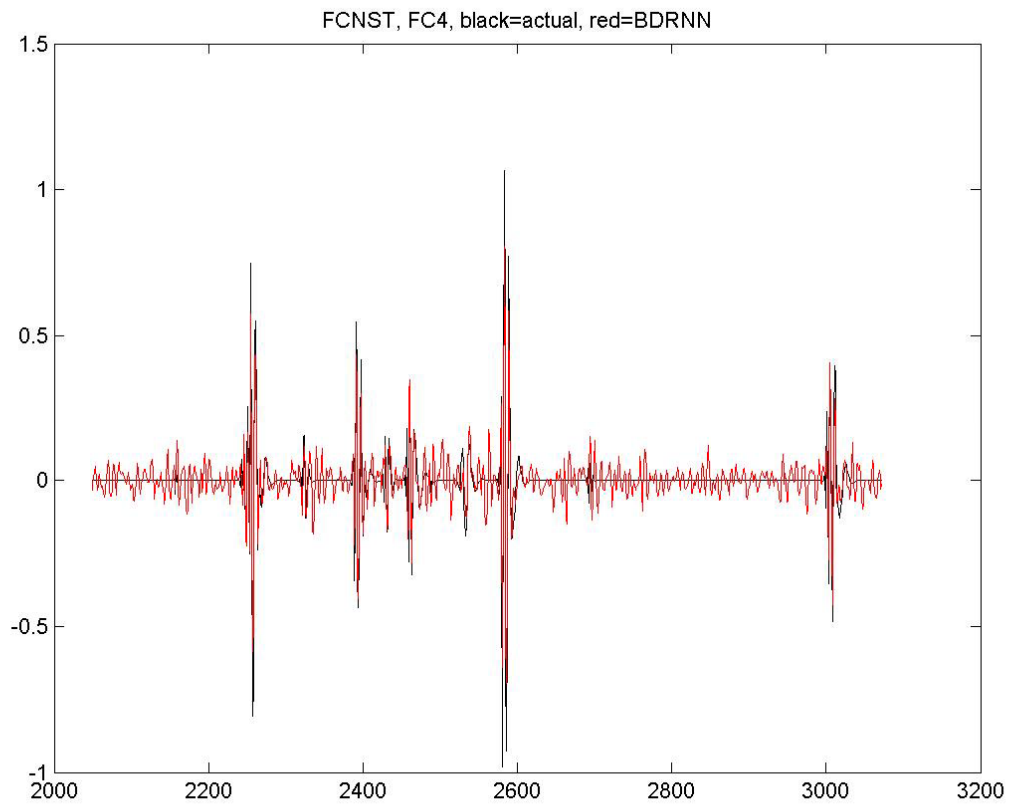
4^η Επανάληψη



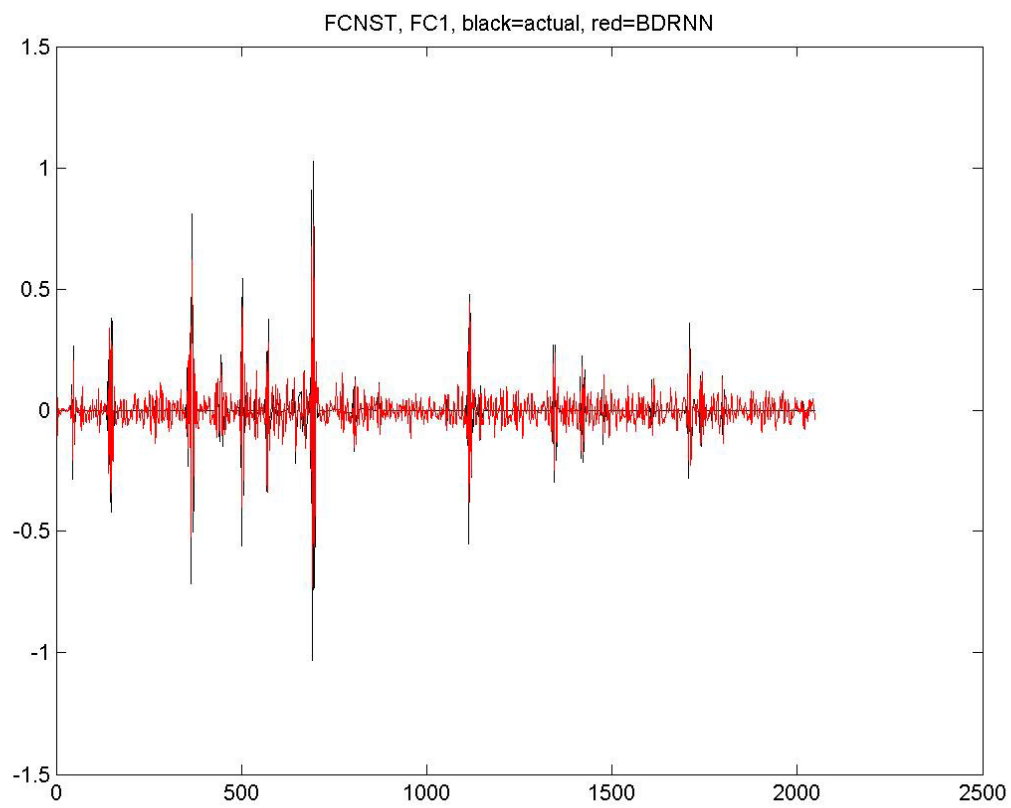
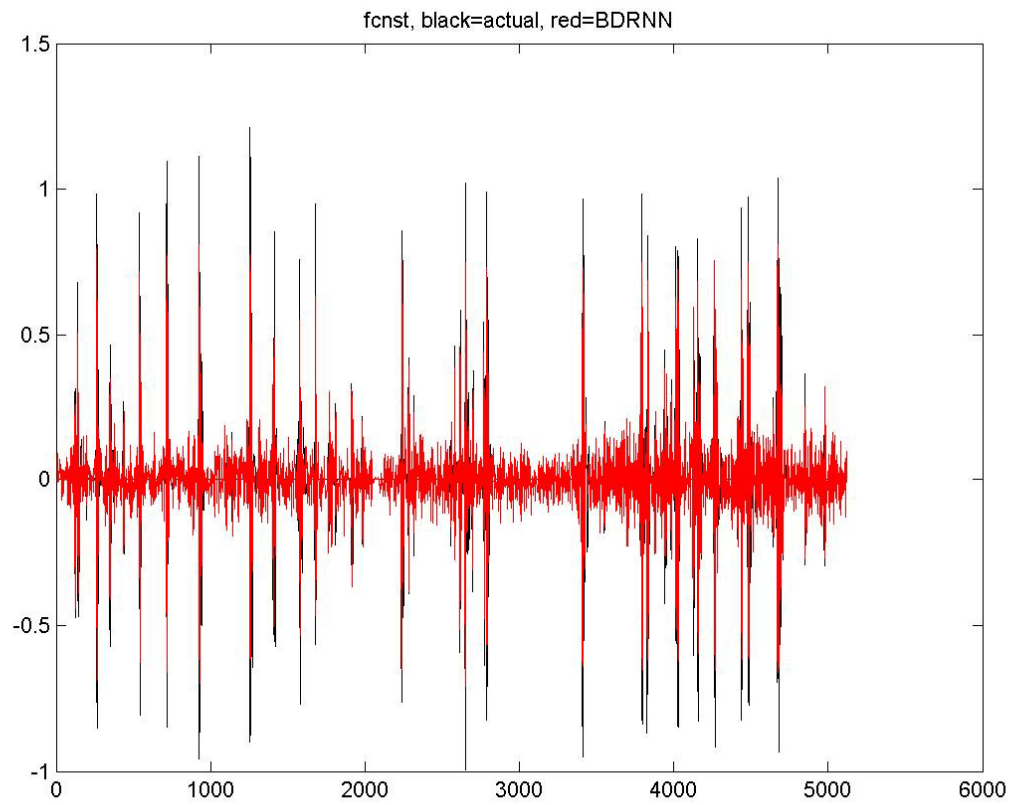


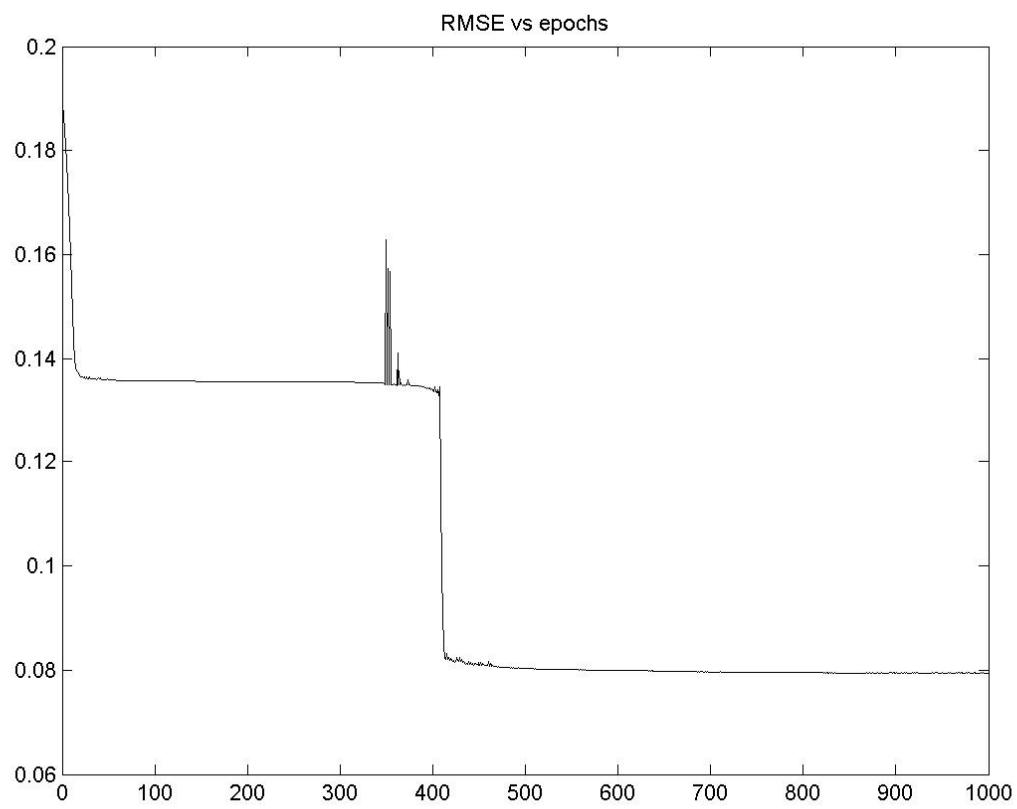
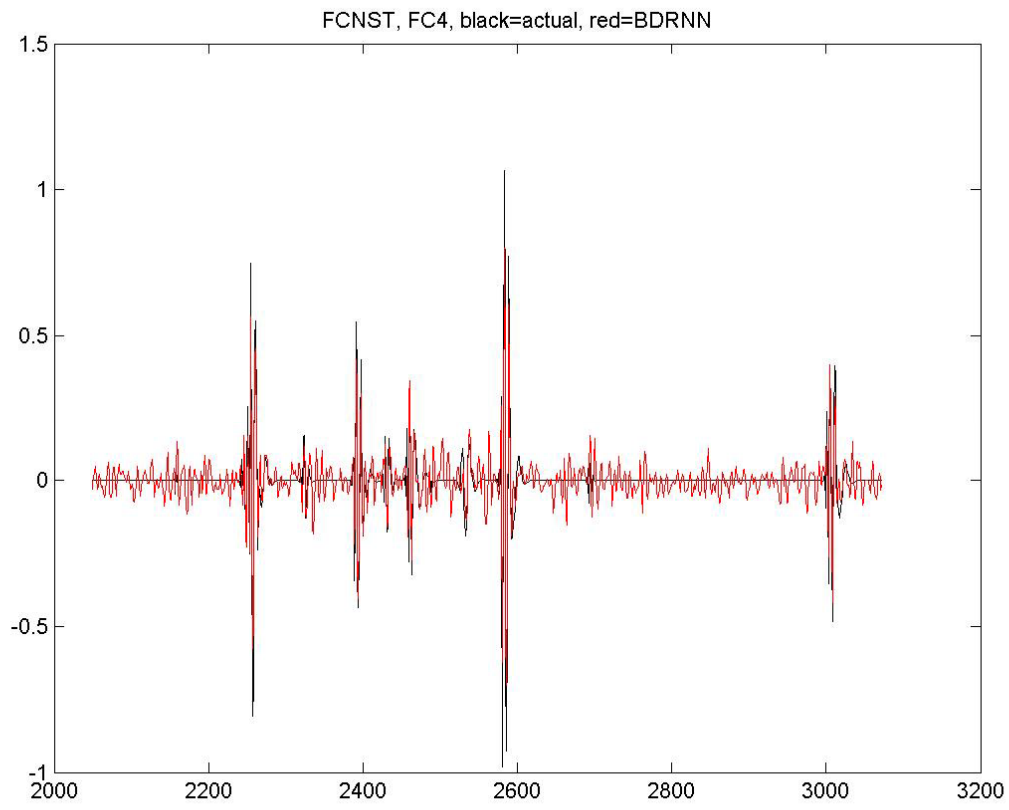
5^η Επανάληψη



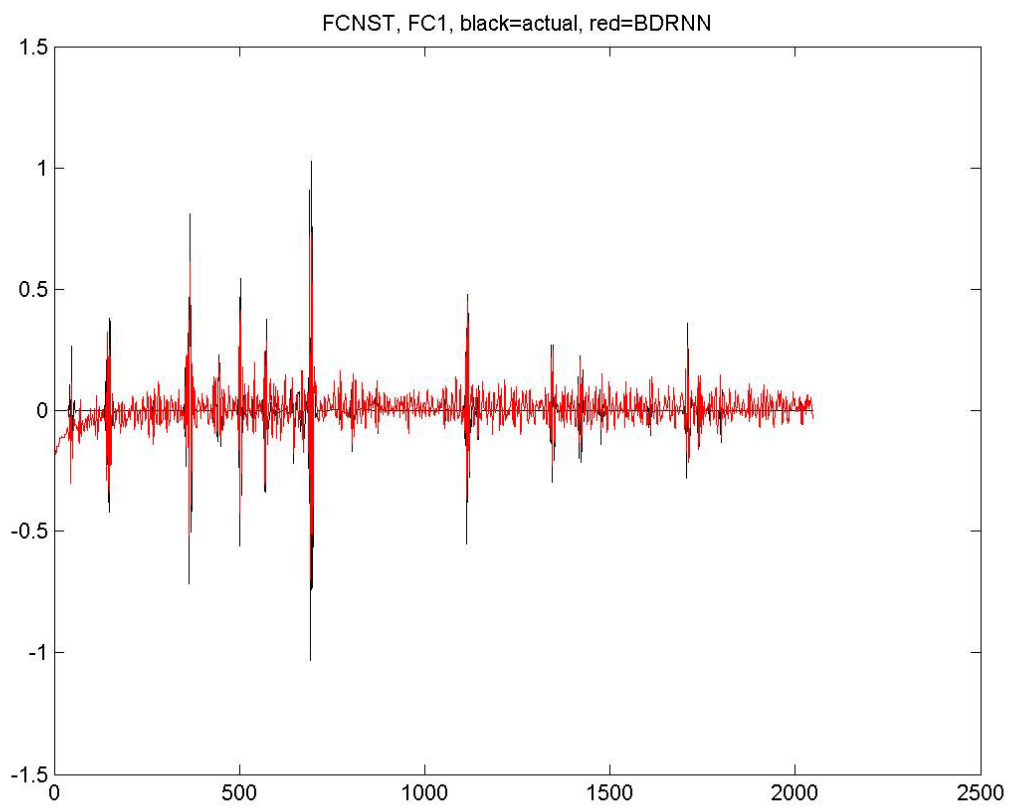
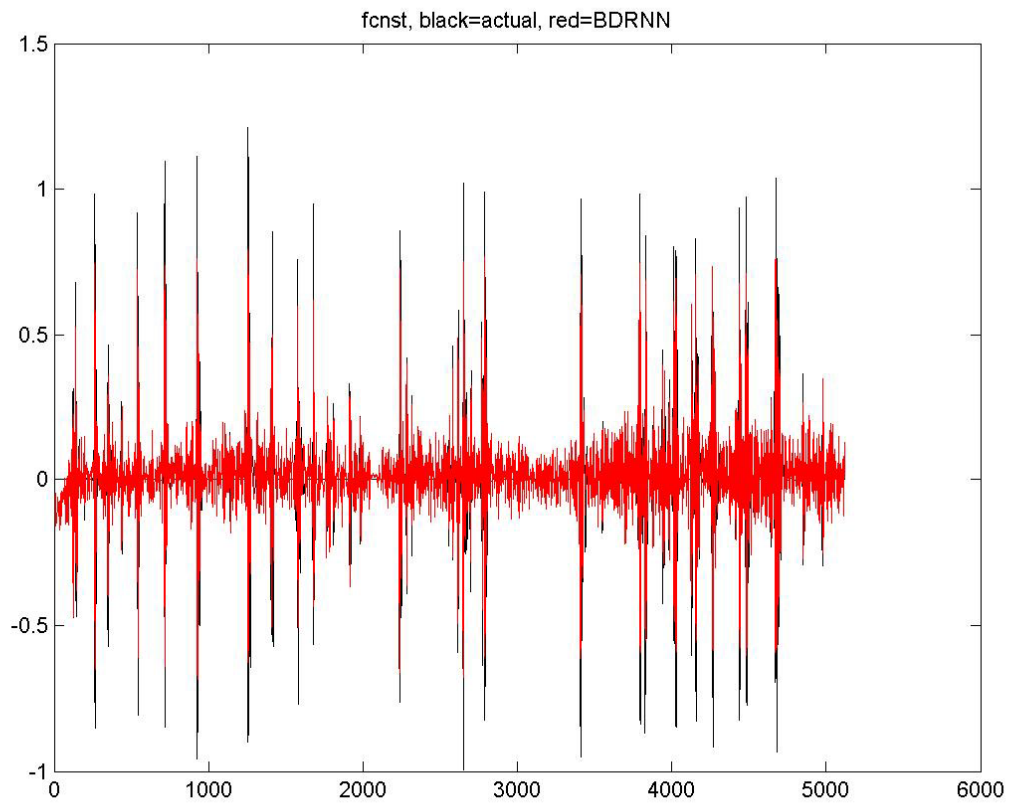


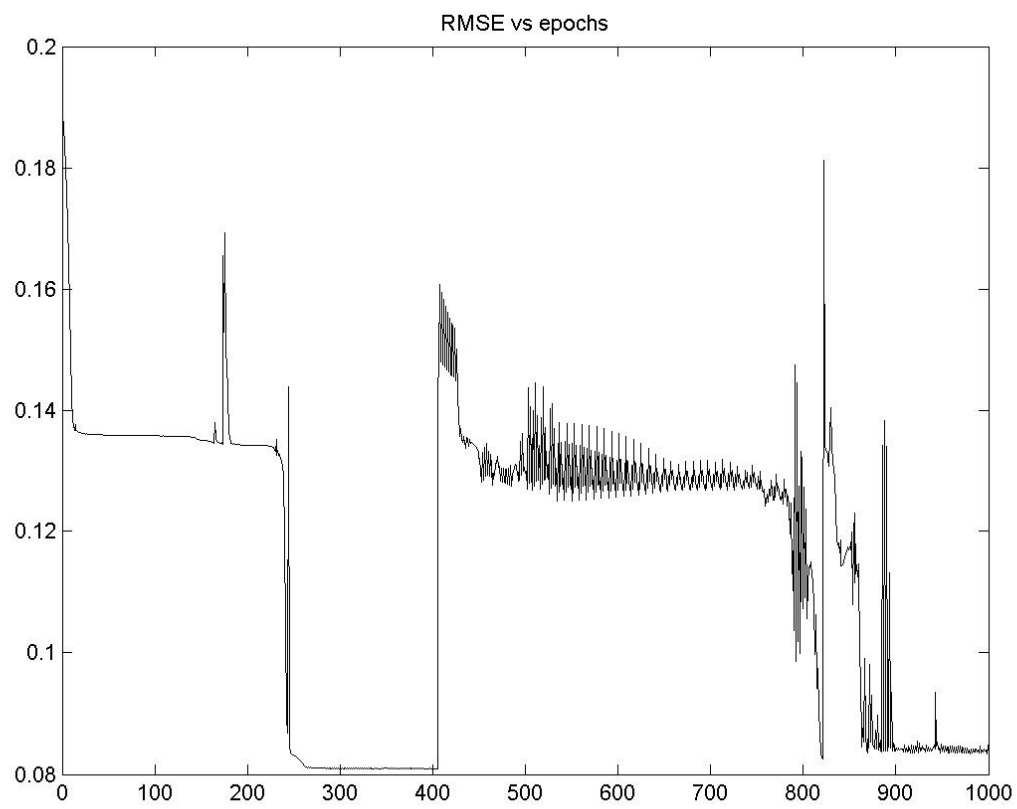
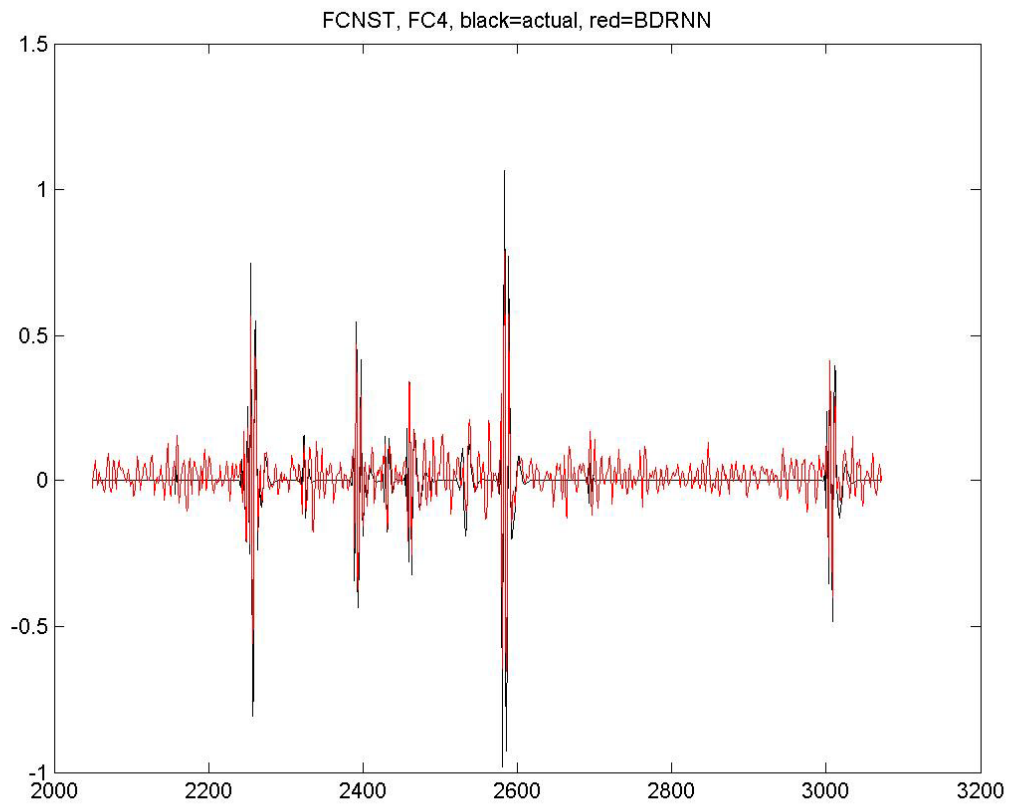
6^η Επανάληψη



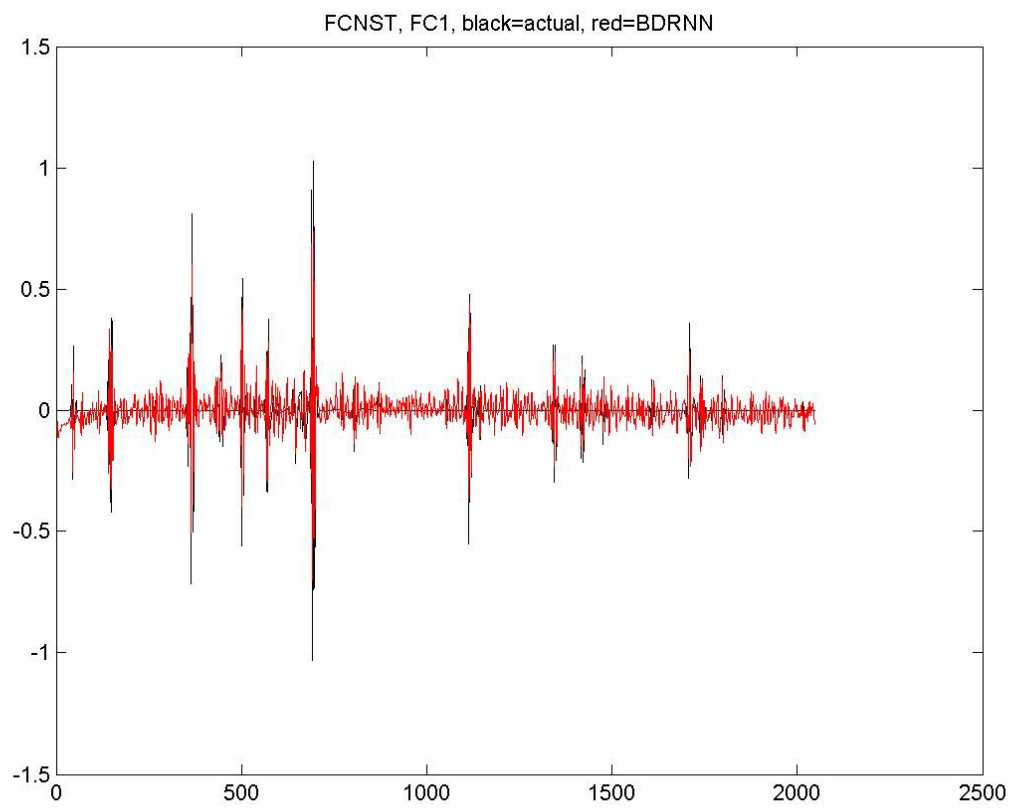
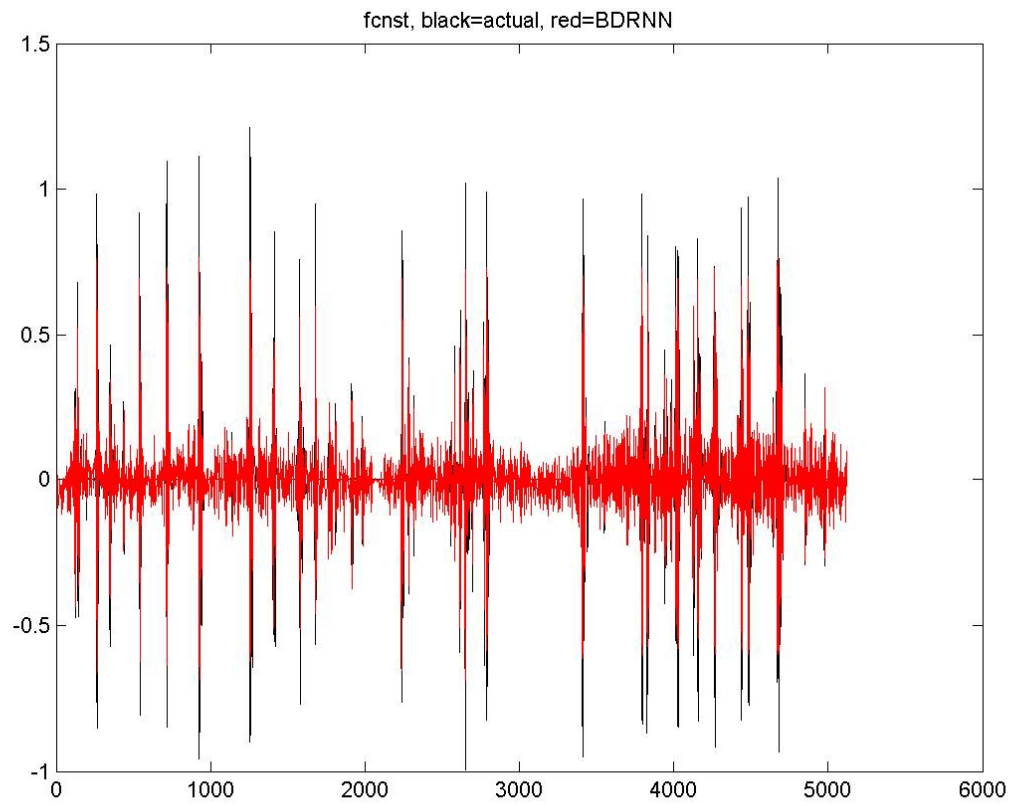


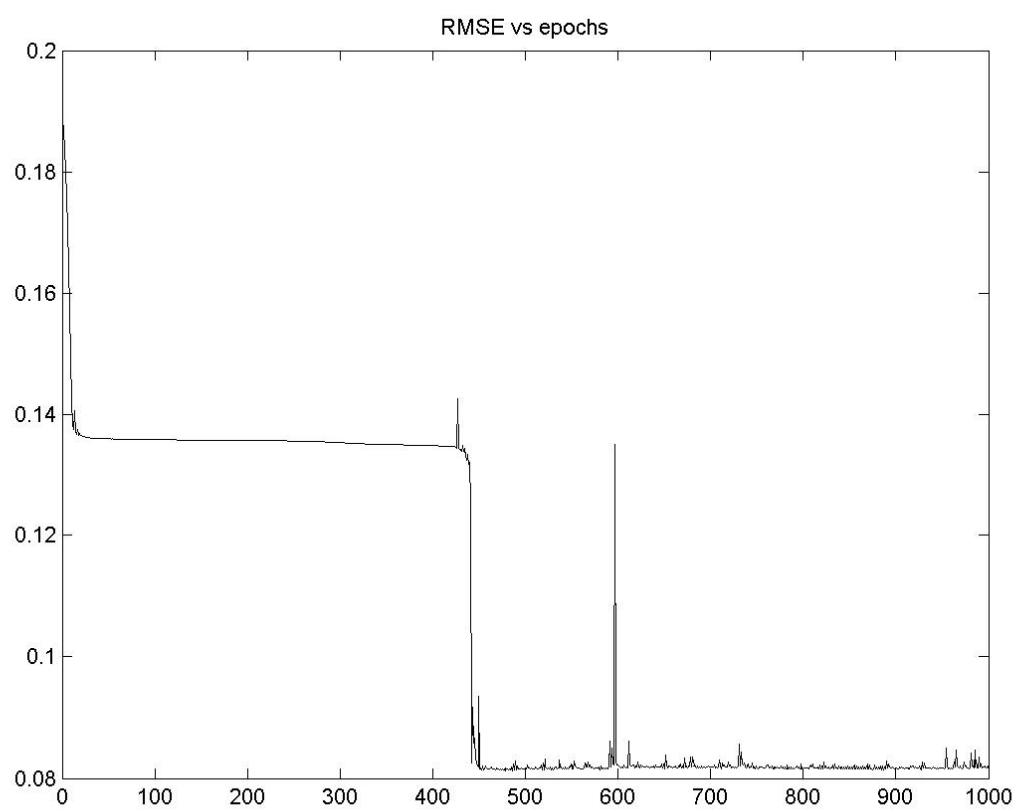
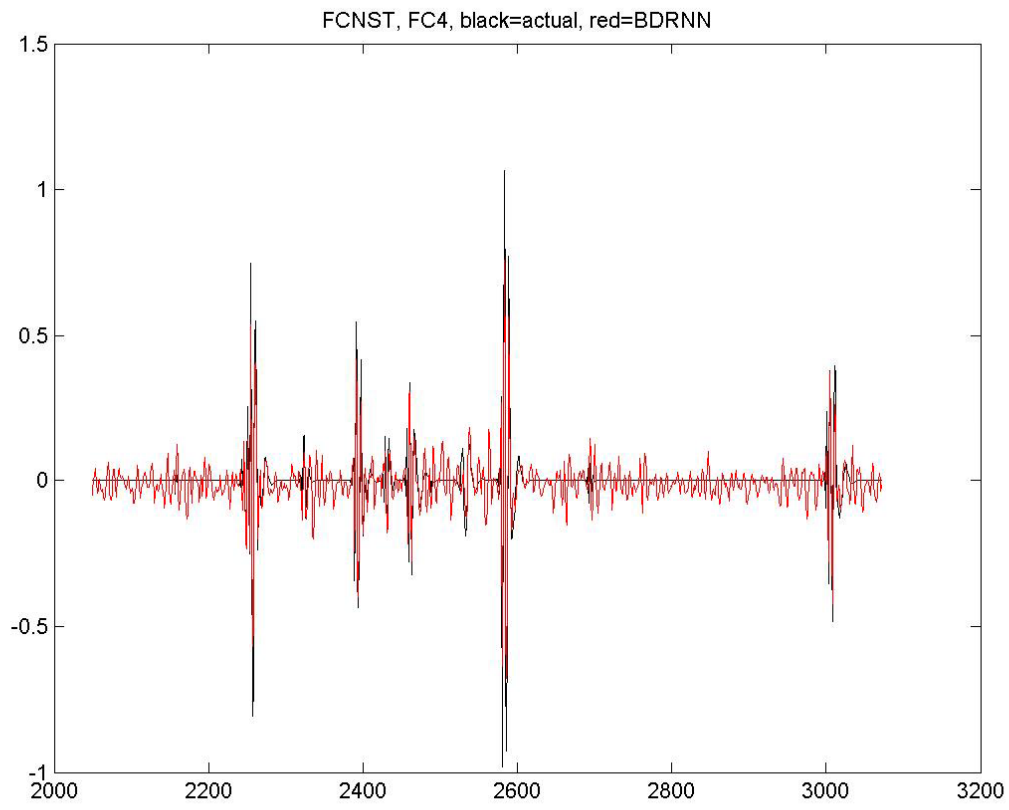
7^η Επανάληψη



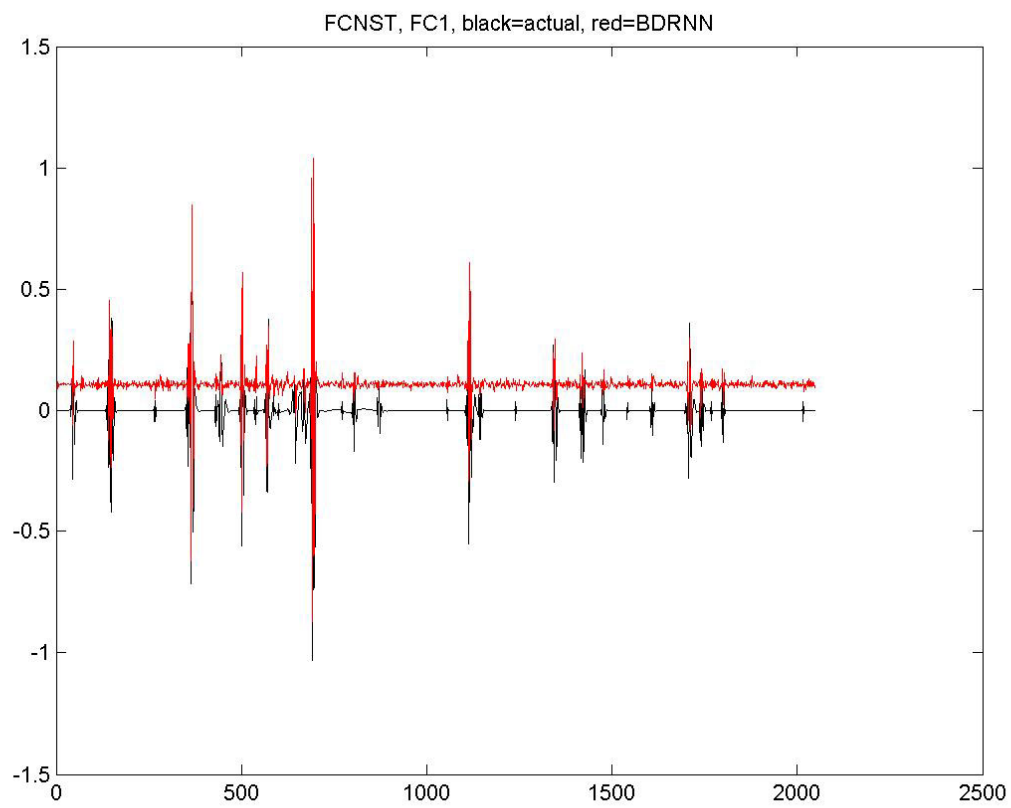
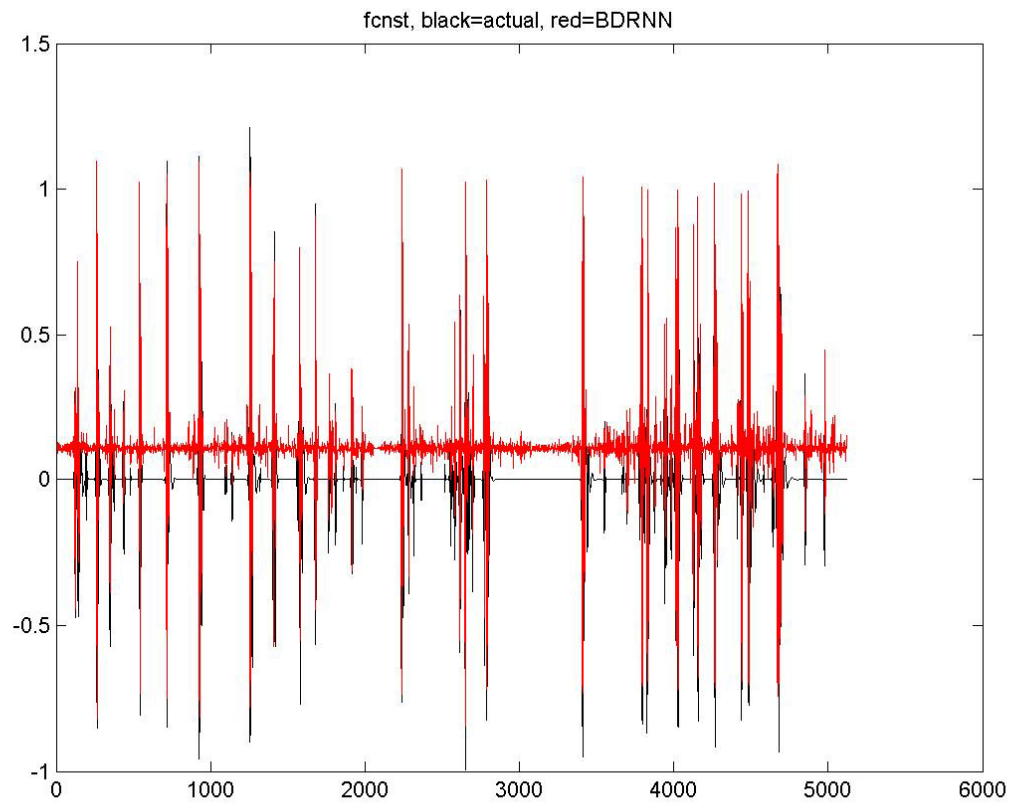


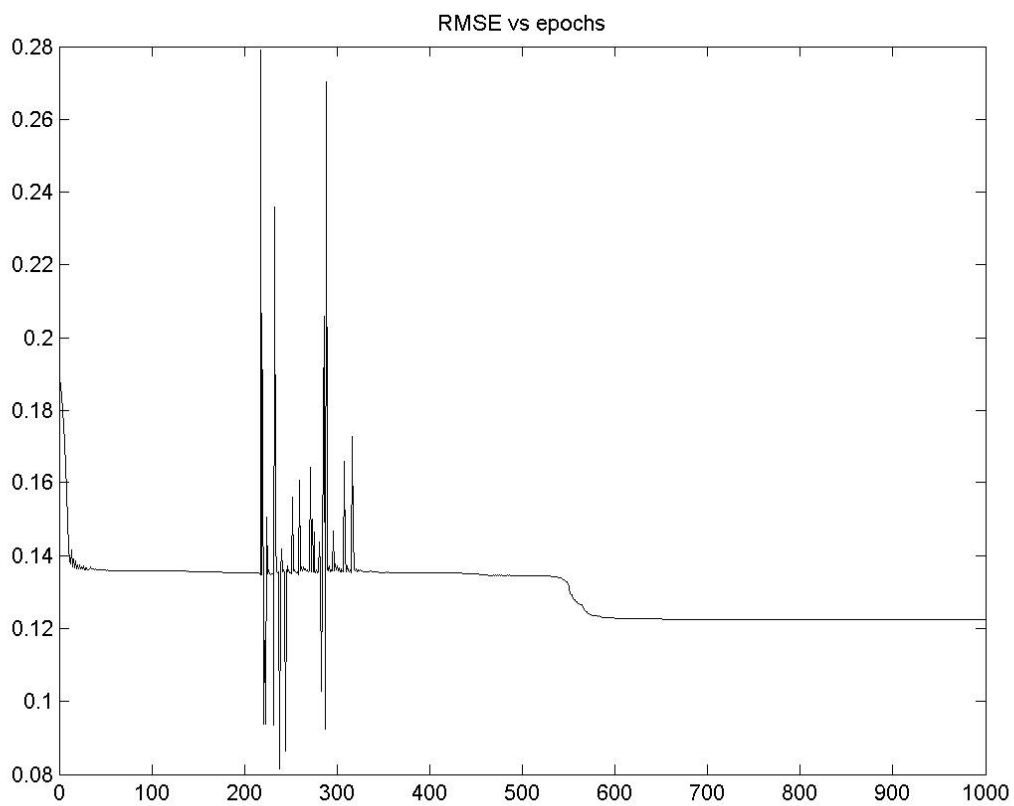
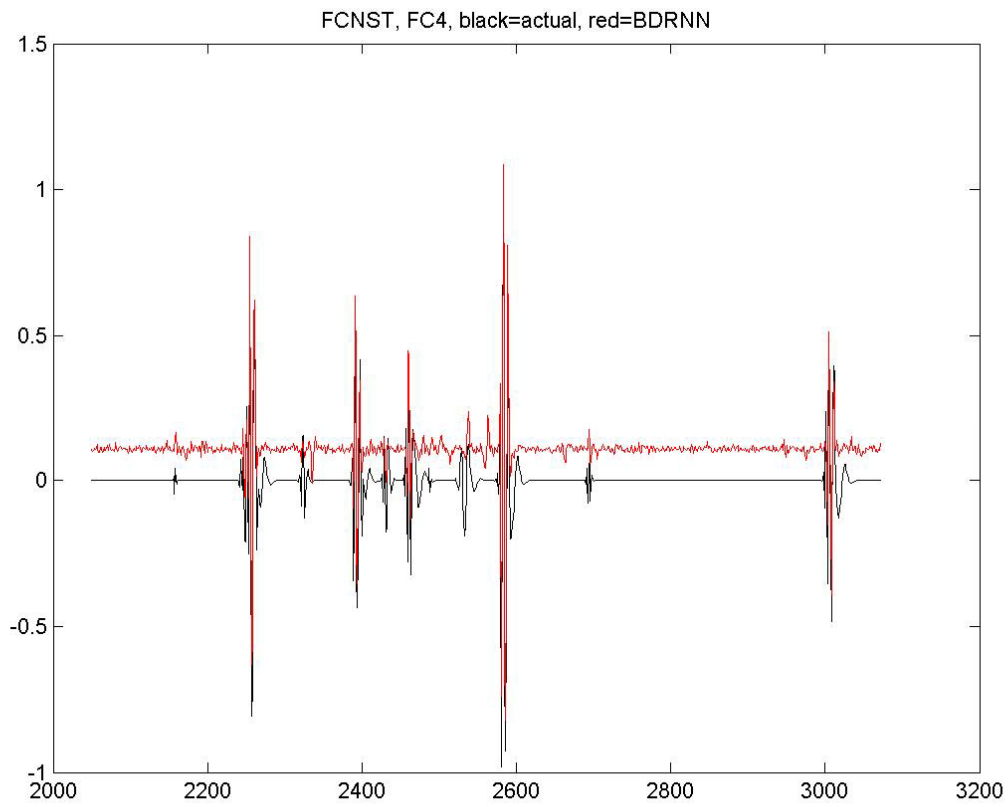
8^η Επανάληψη



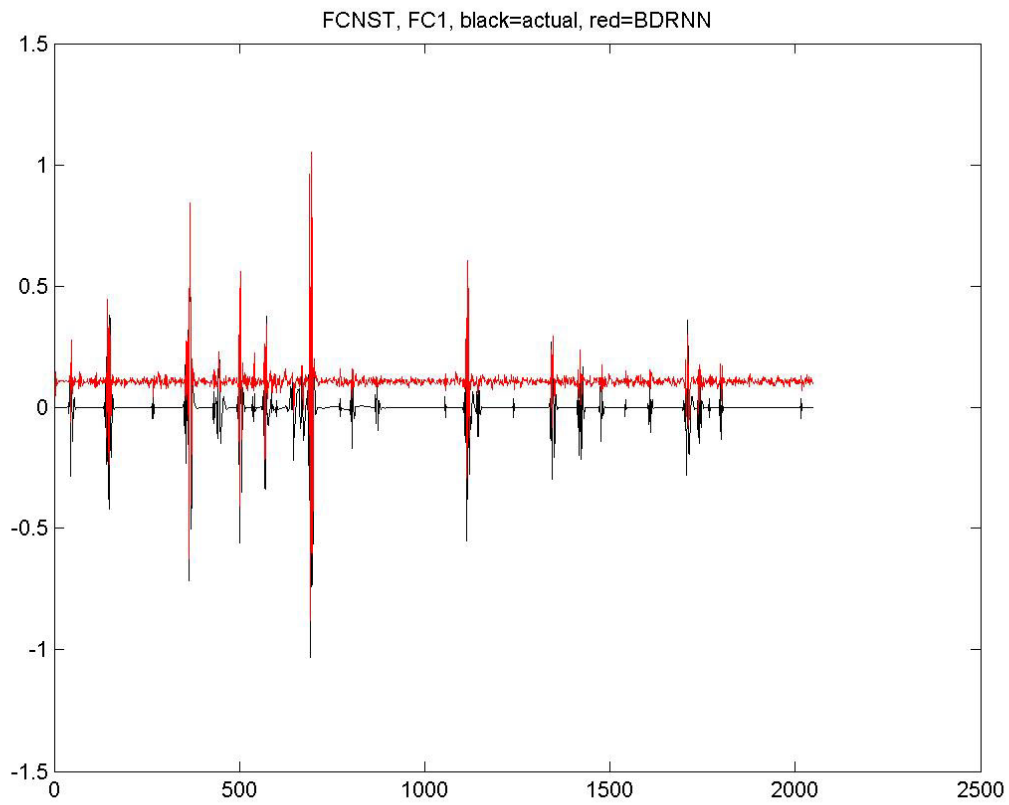
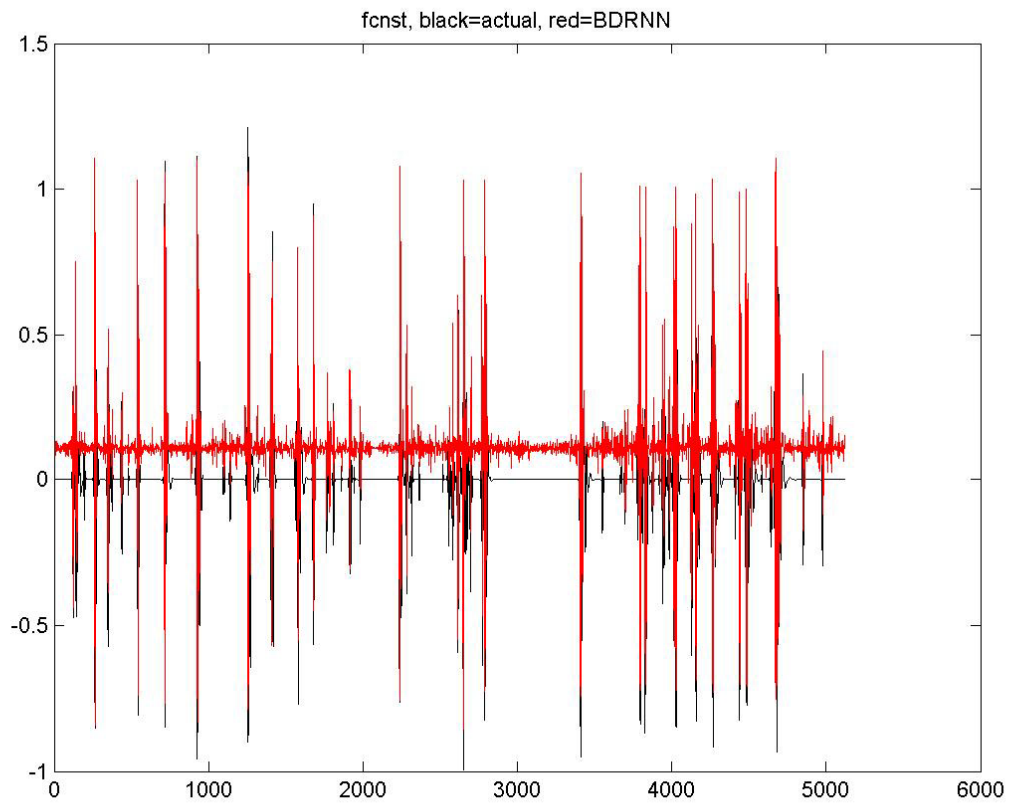


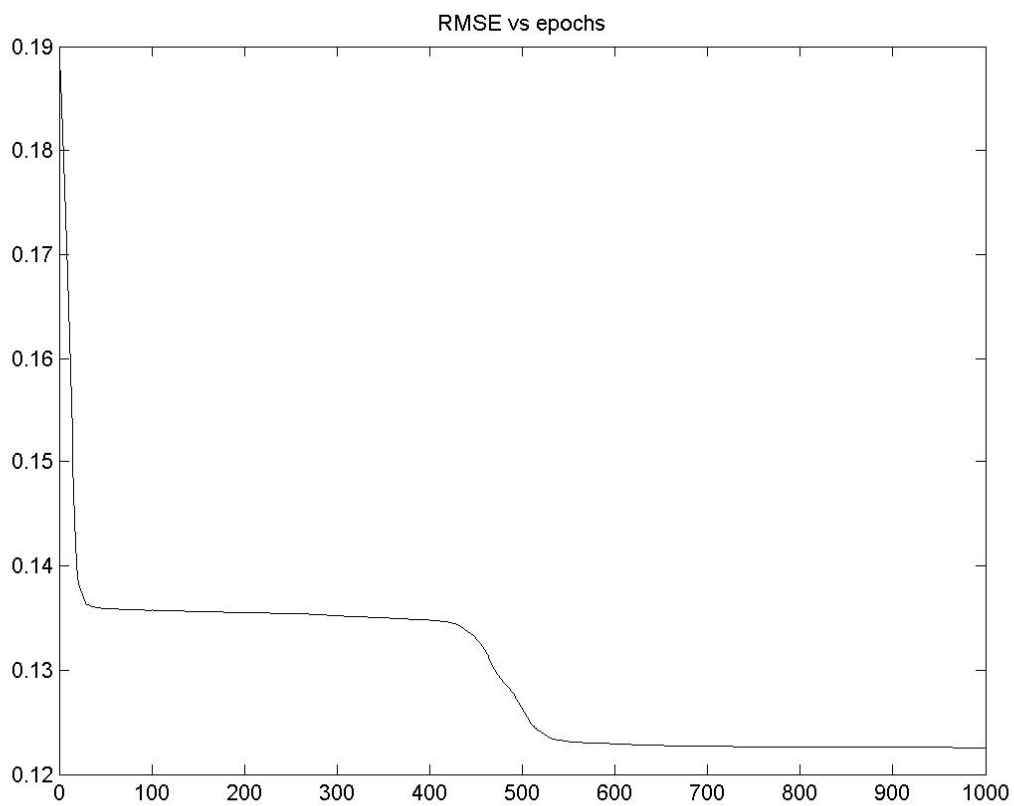
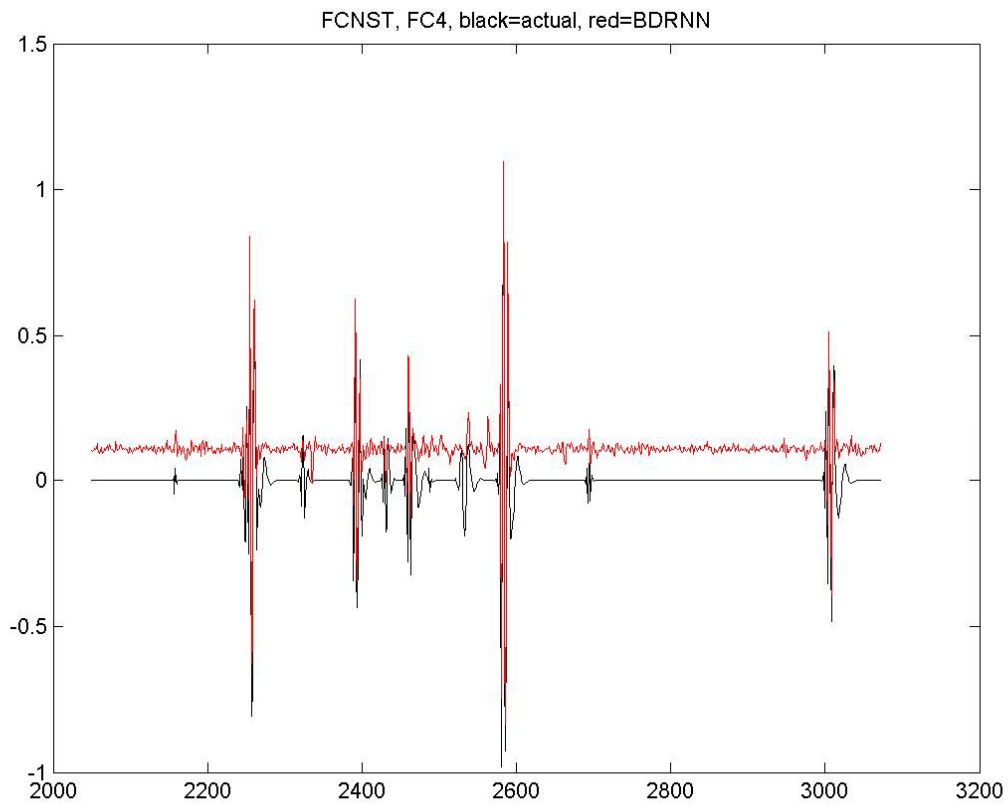
9^η Επανάληψη



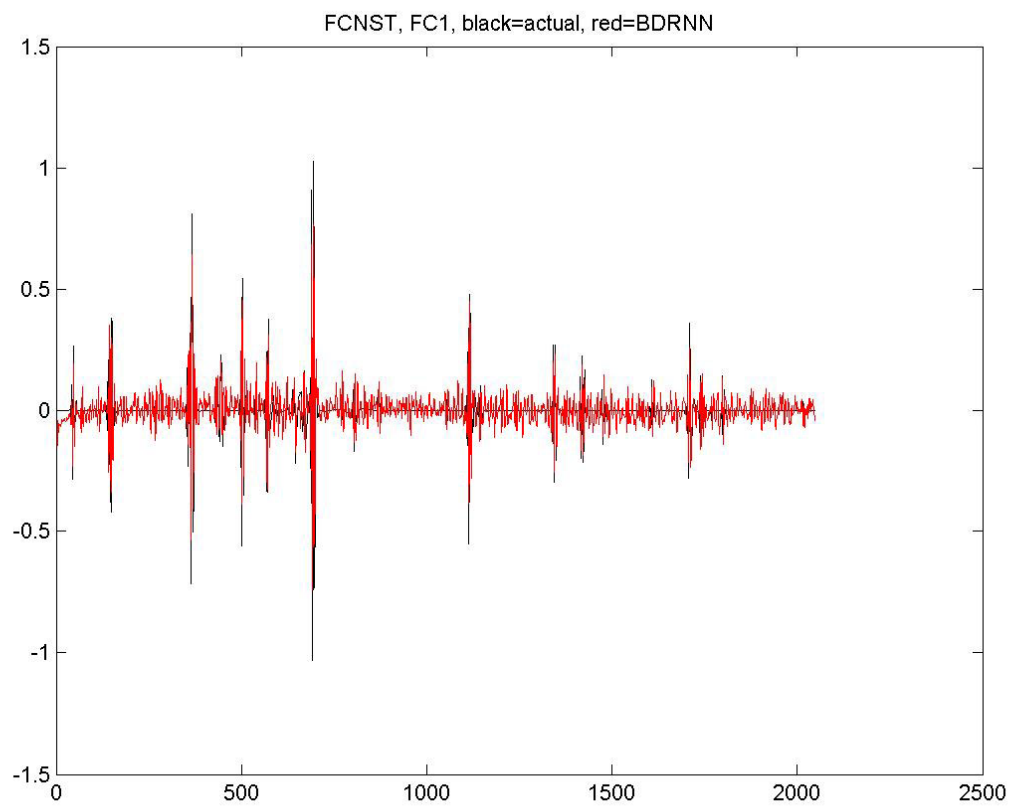
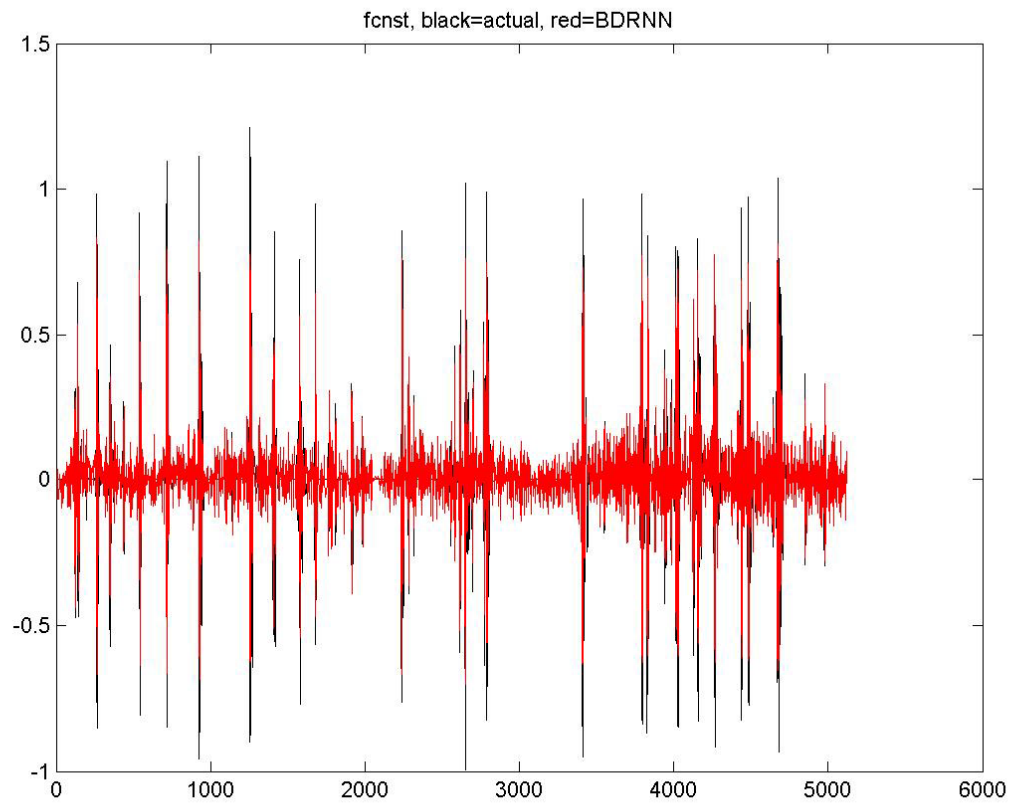


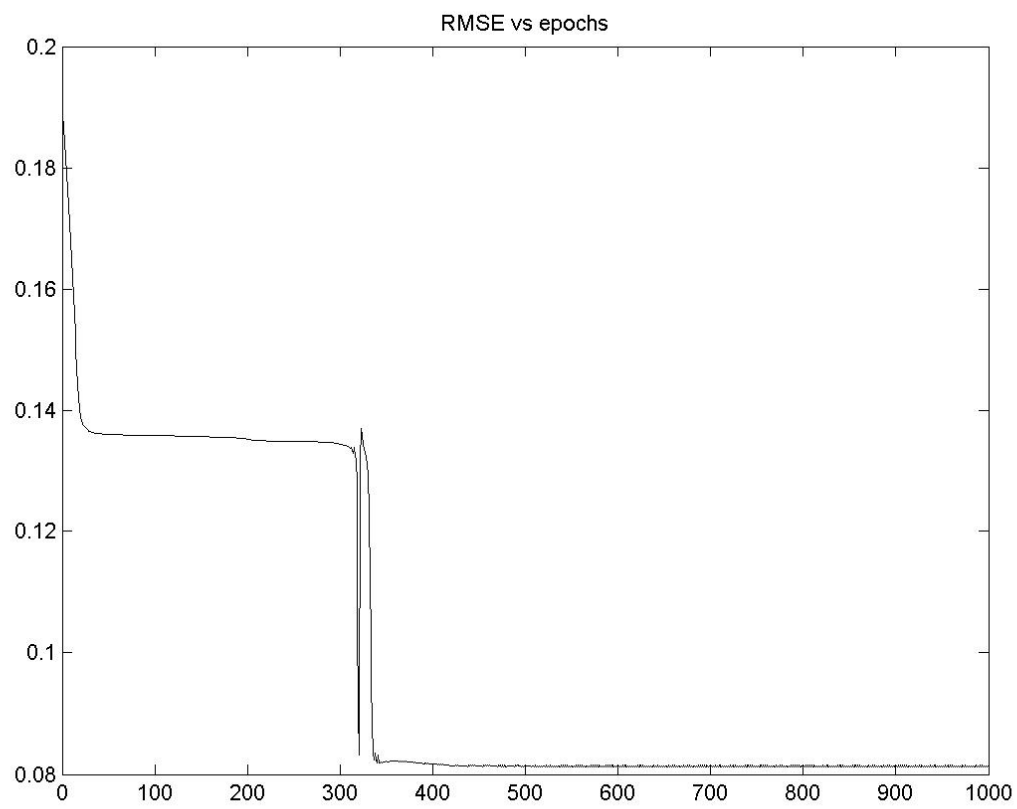
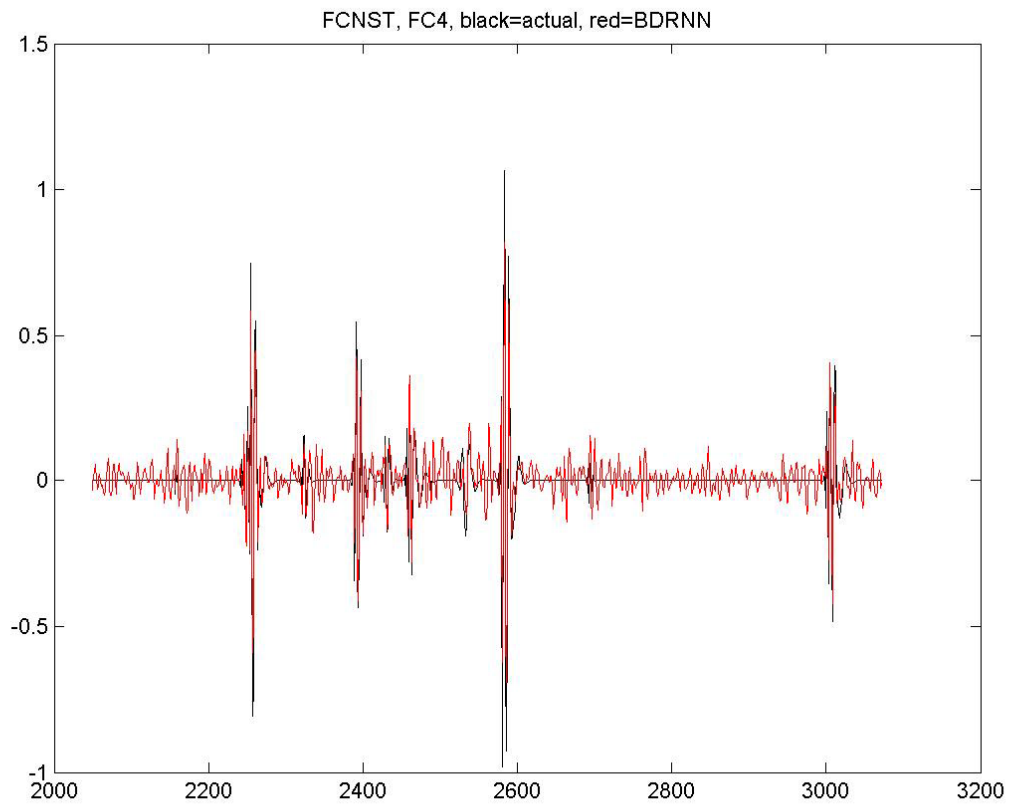
10¹ Επανάληψη



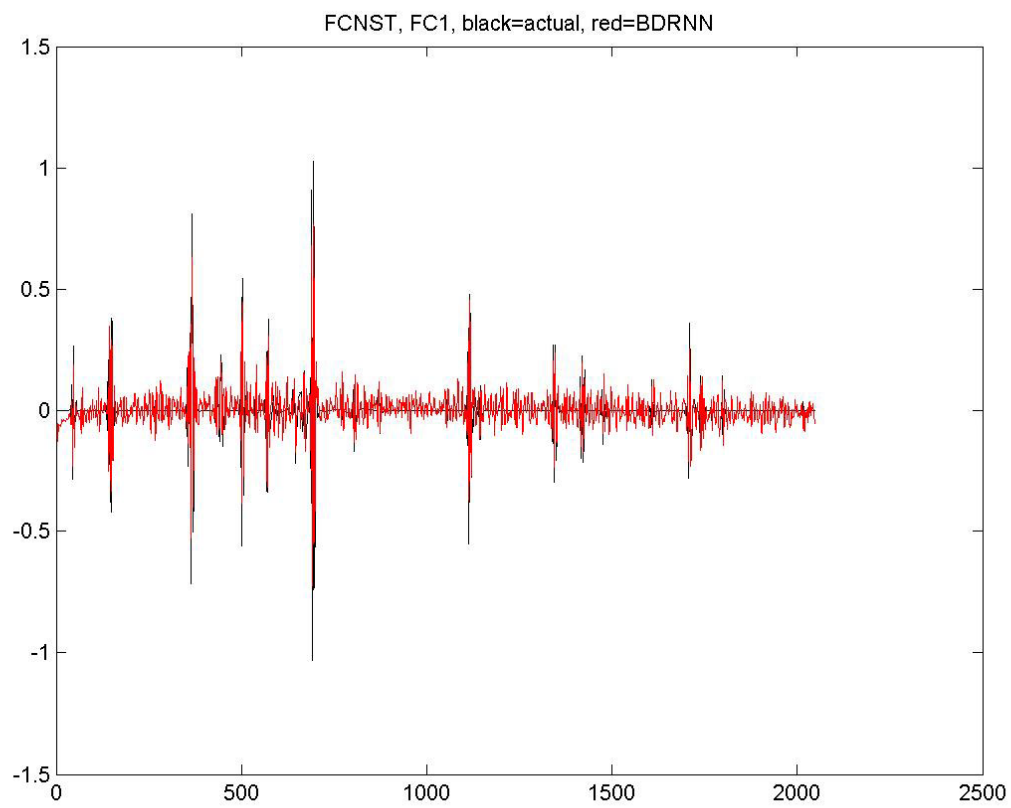
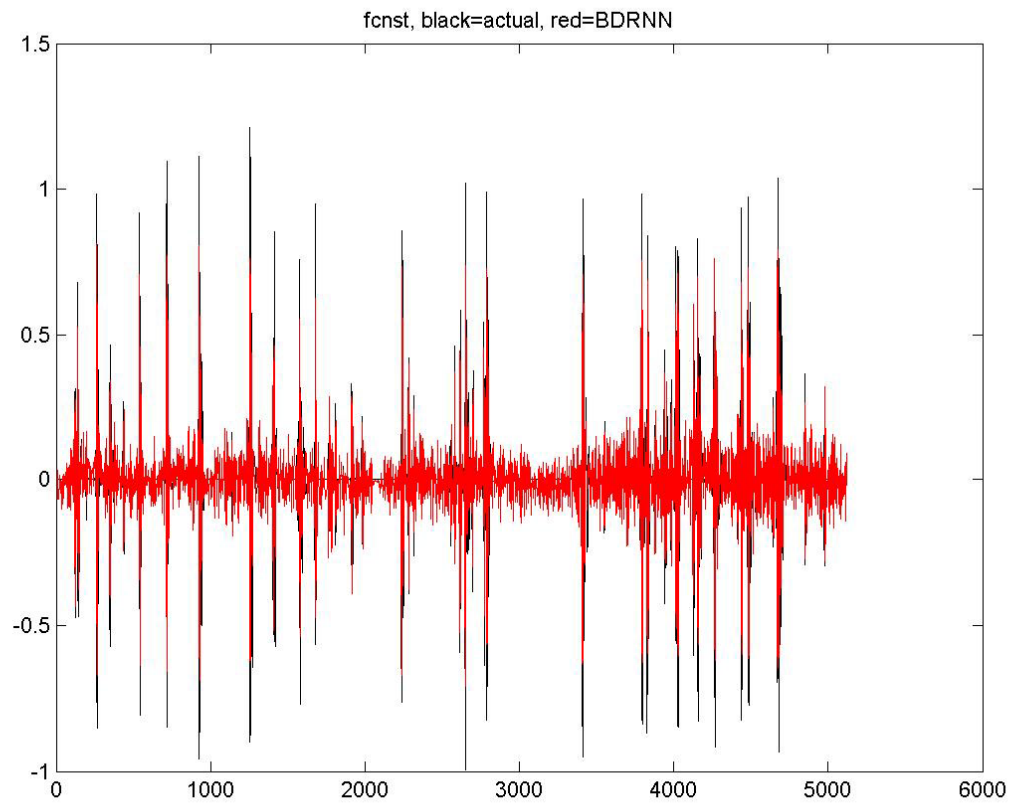


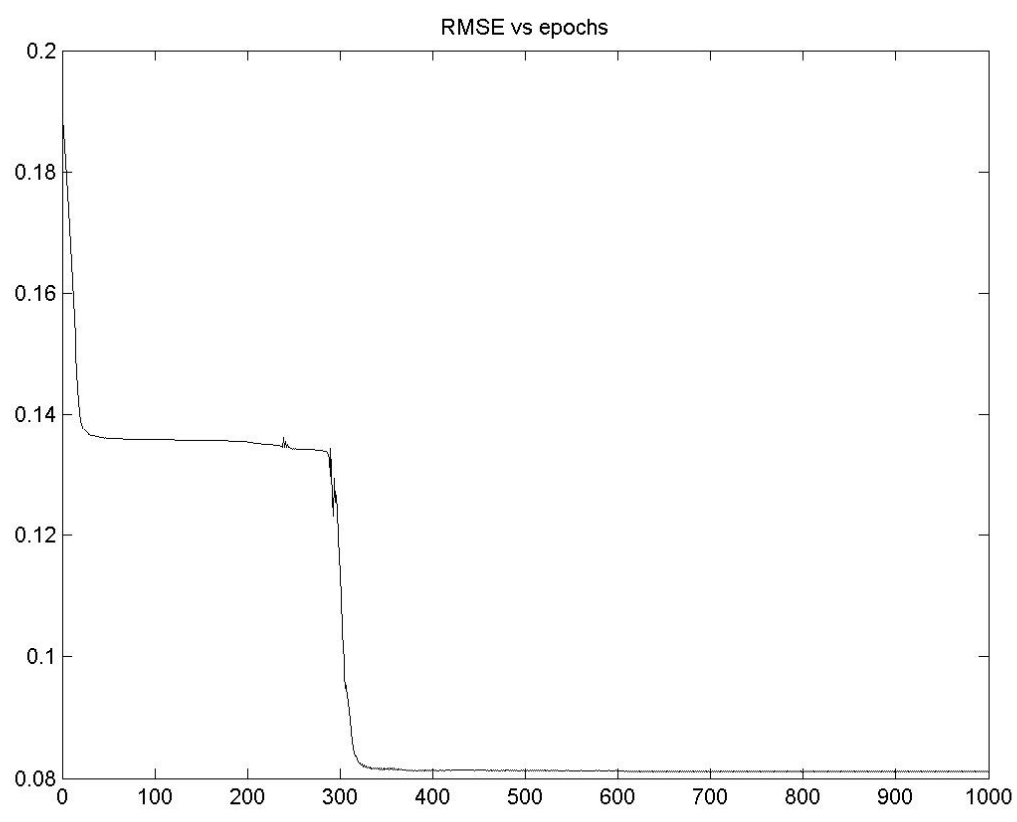
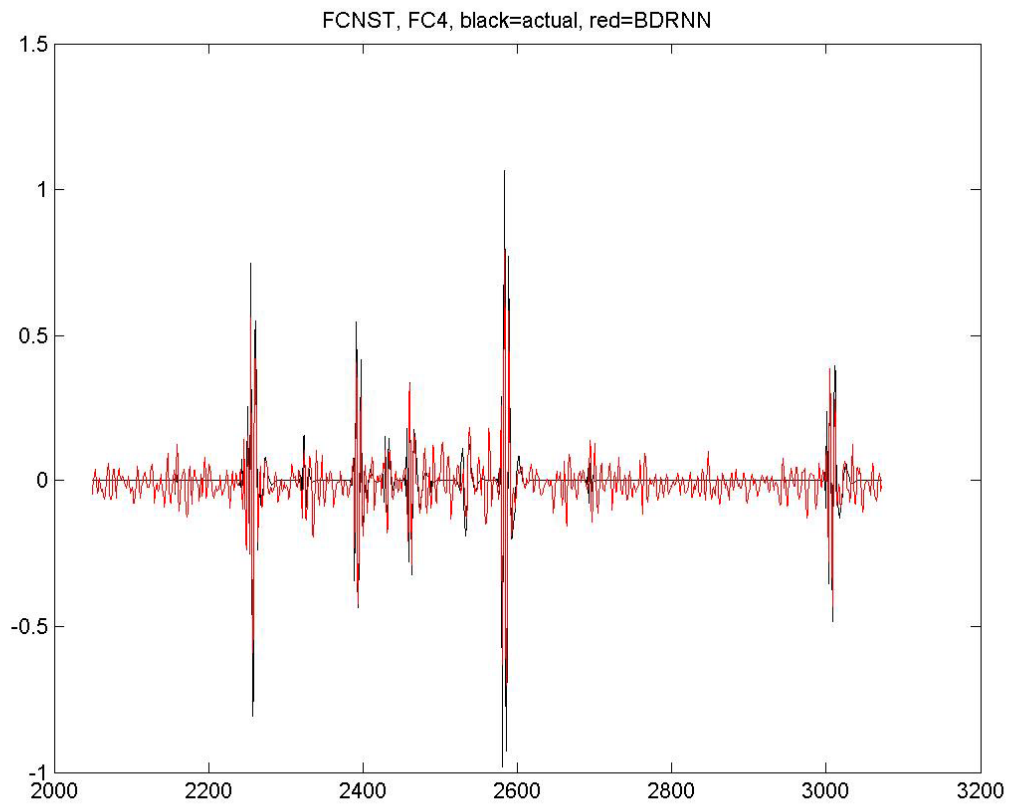
11^η Επανάληψη



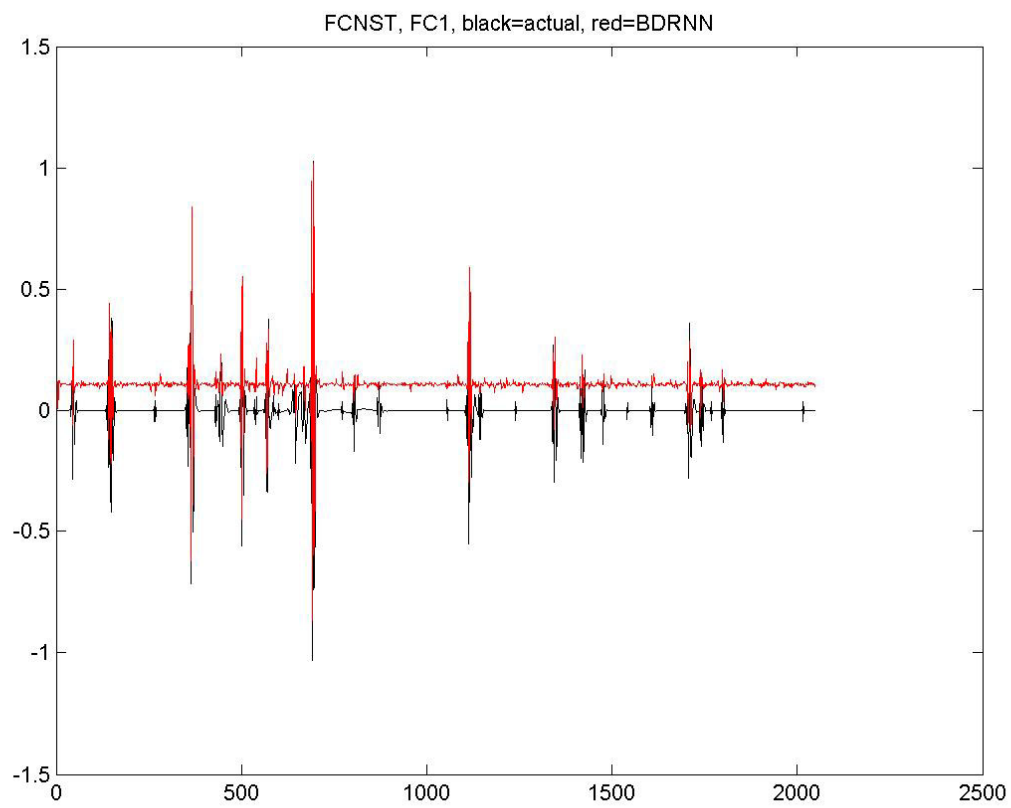
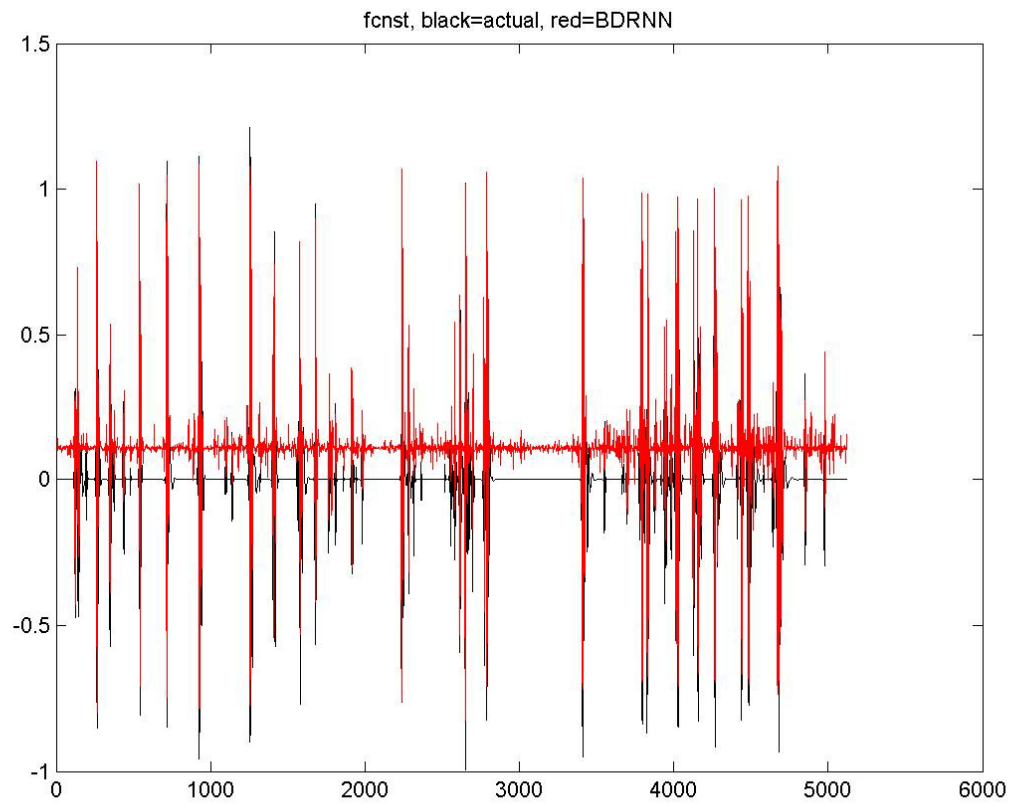


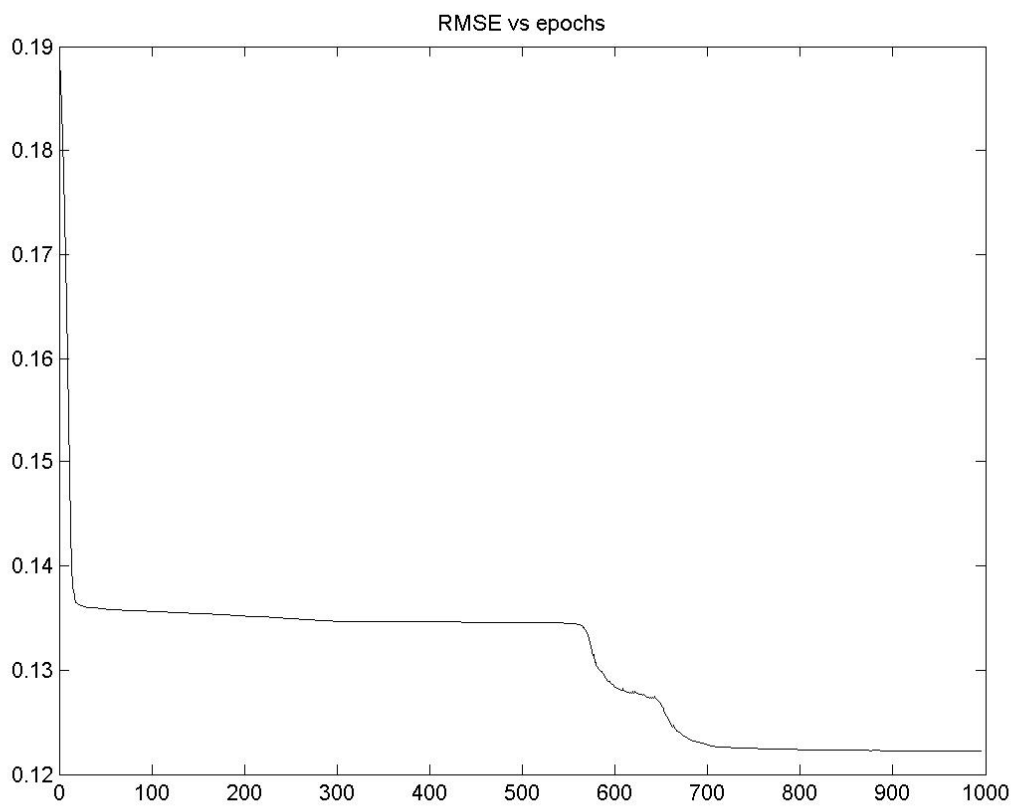
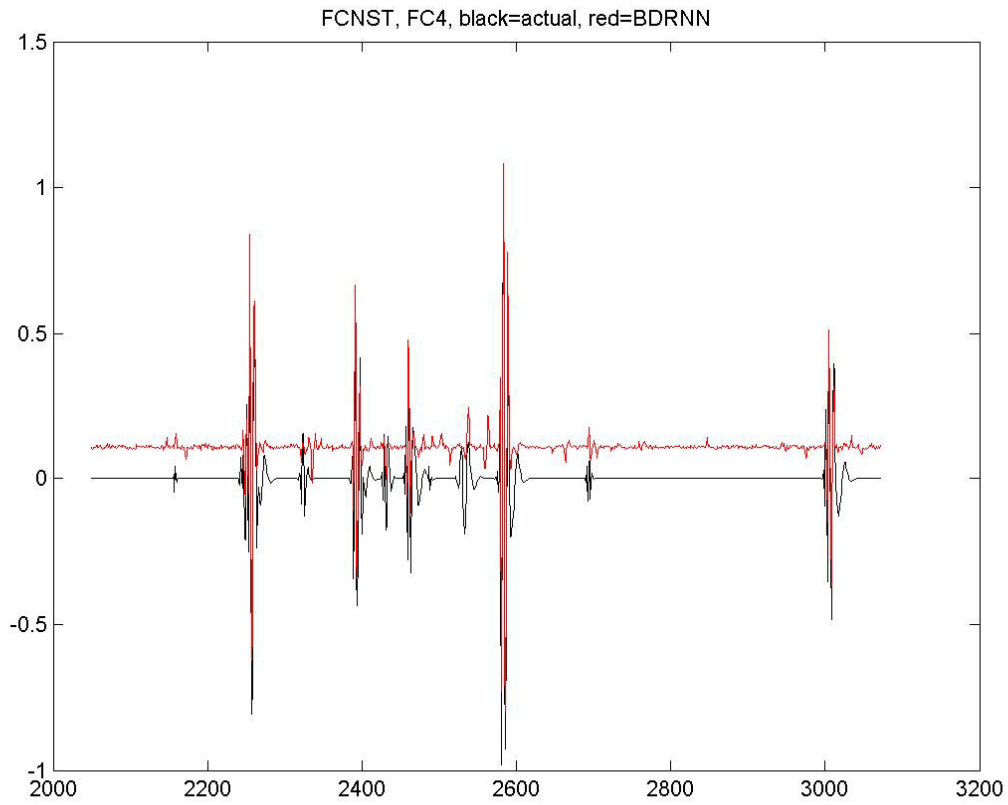
12^η Επανάληψη



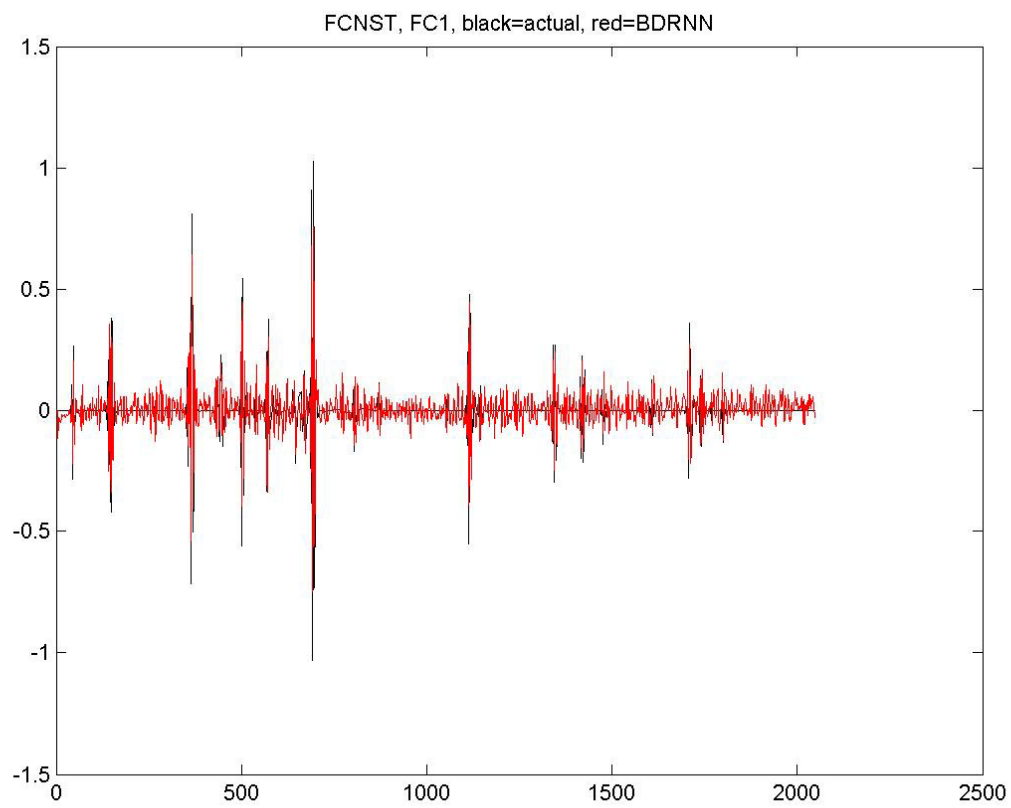
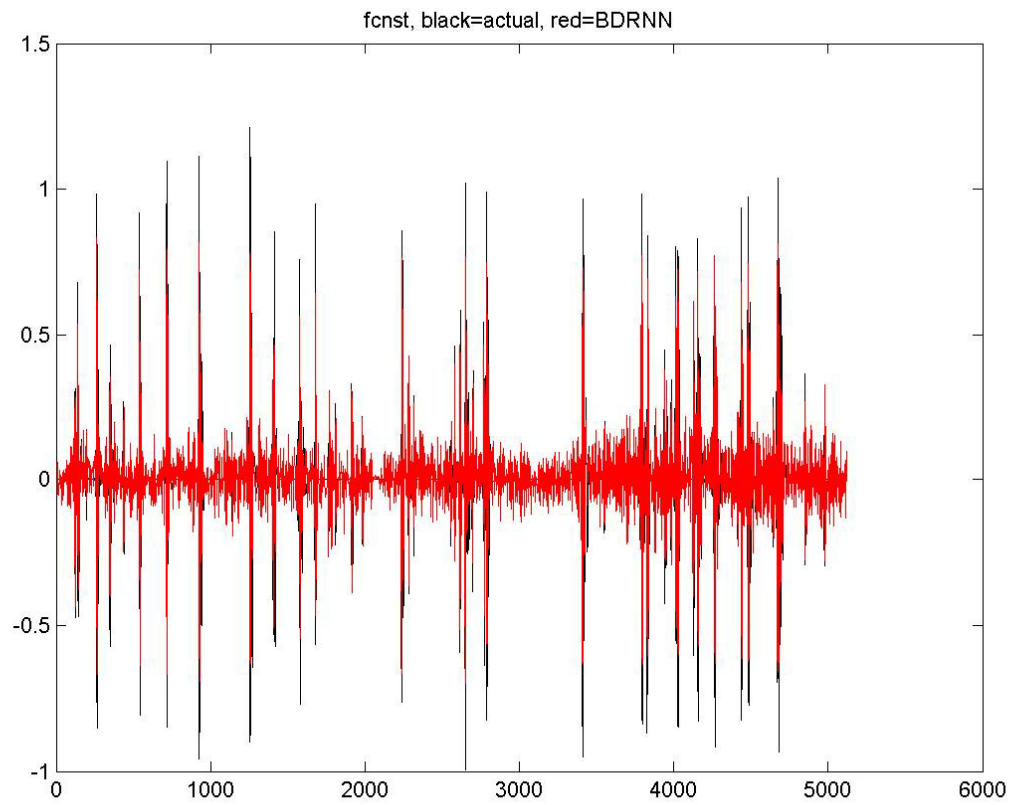


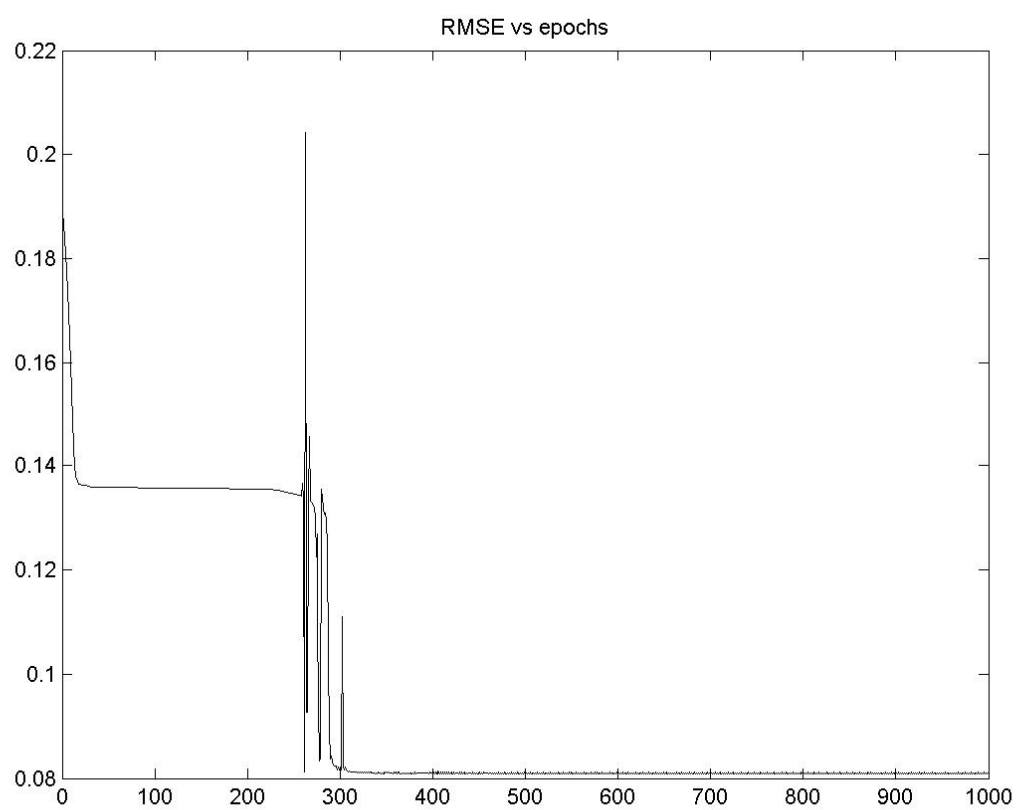
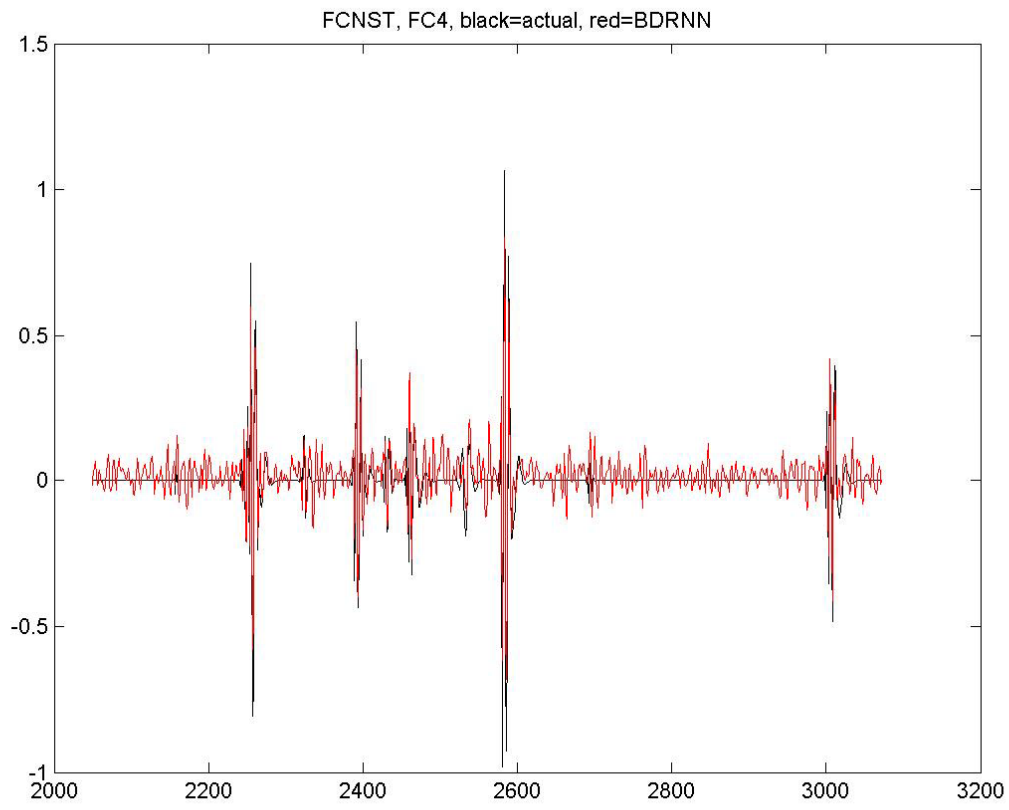
13^η Επανάληψη



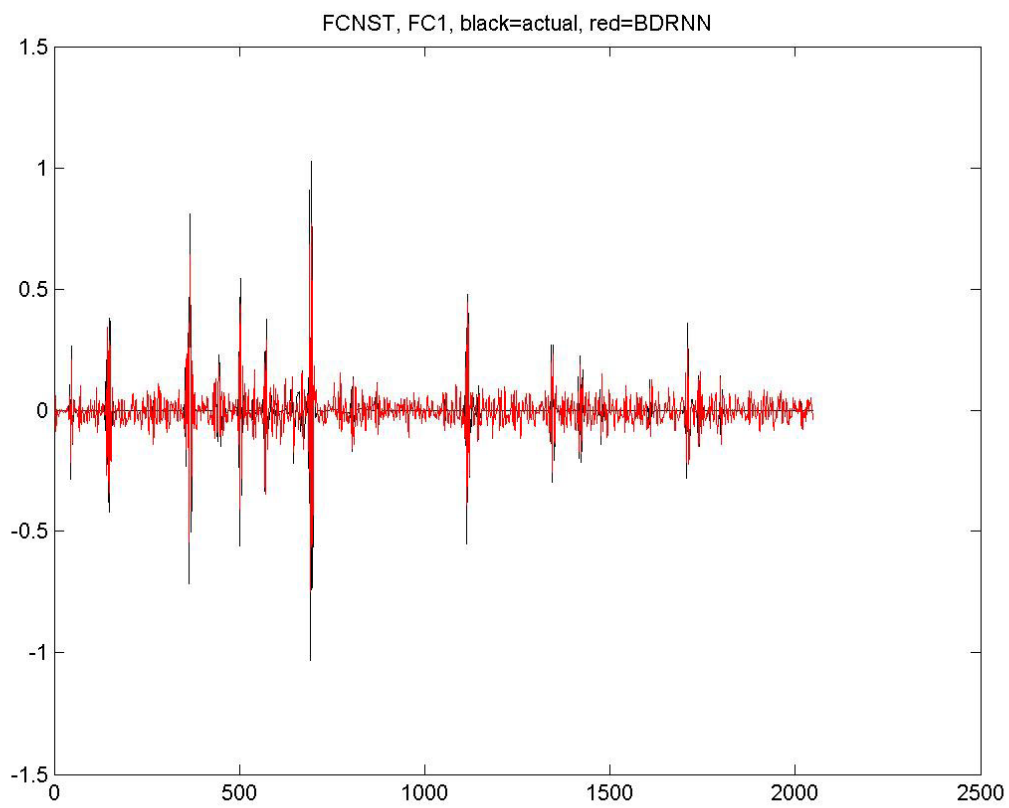
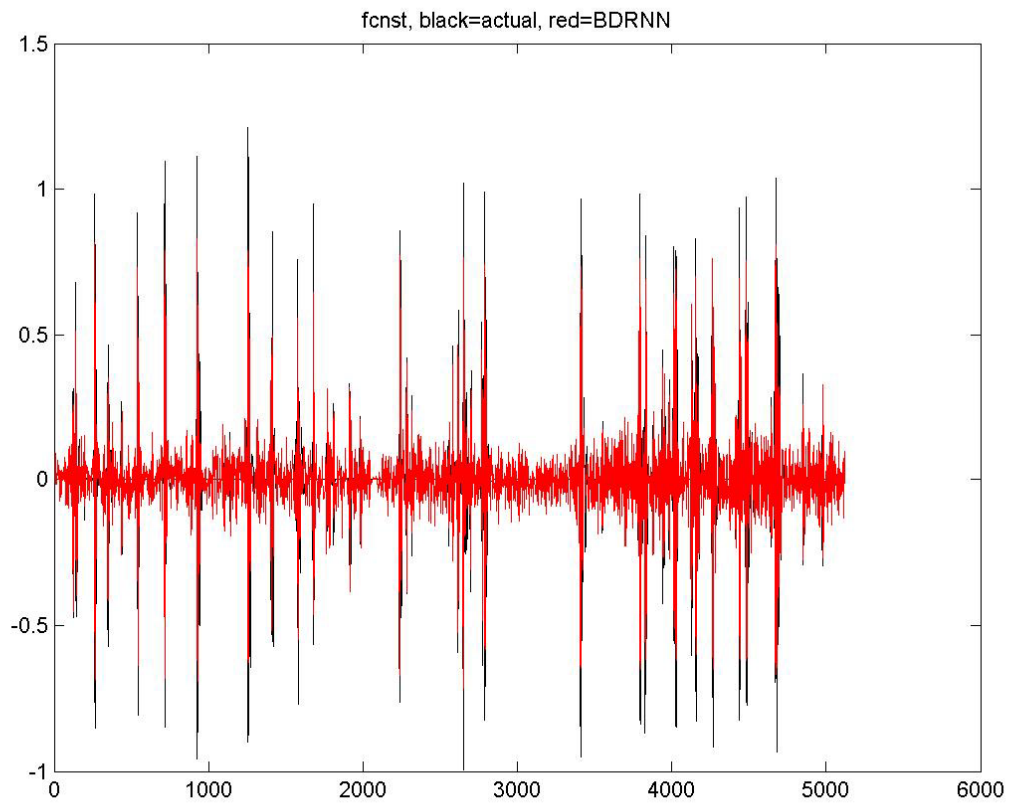


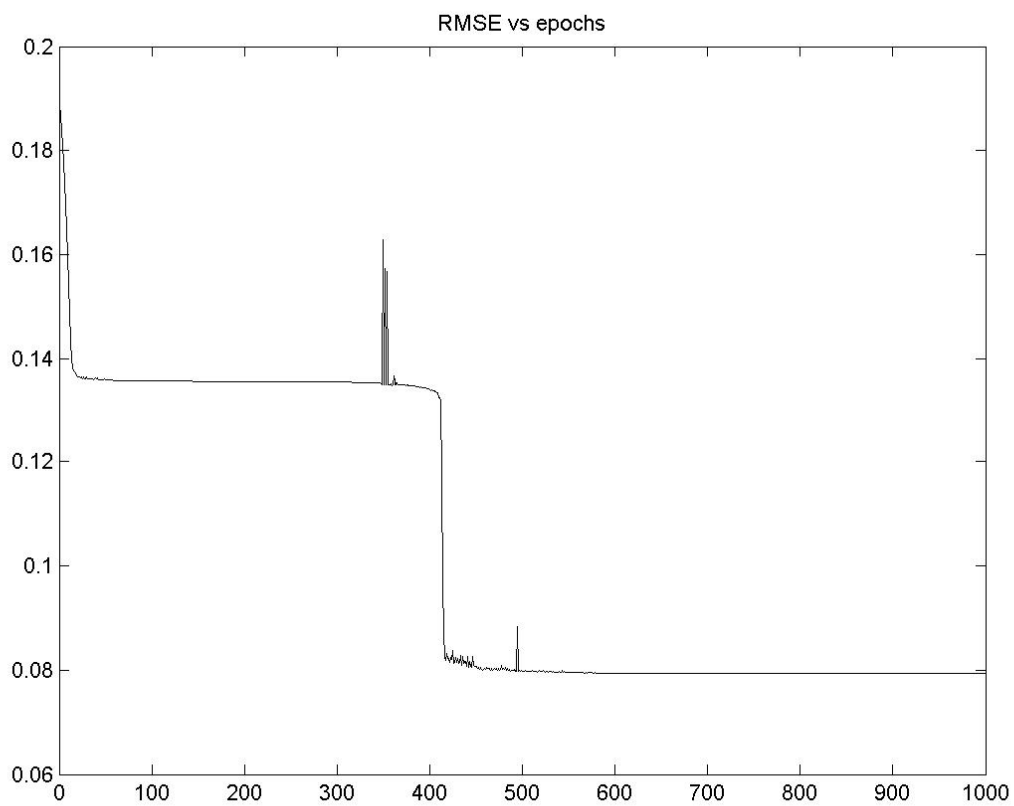
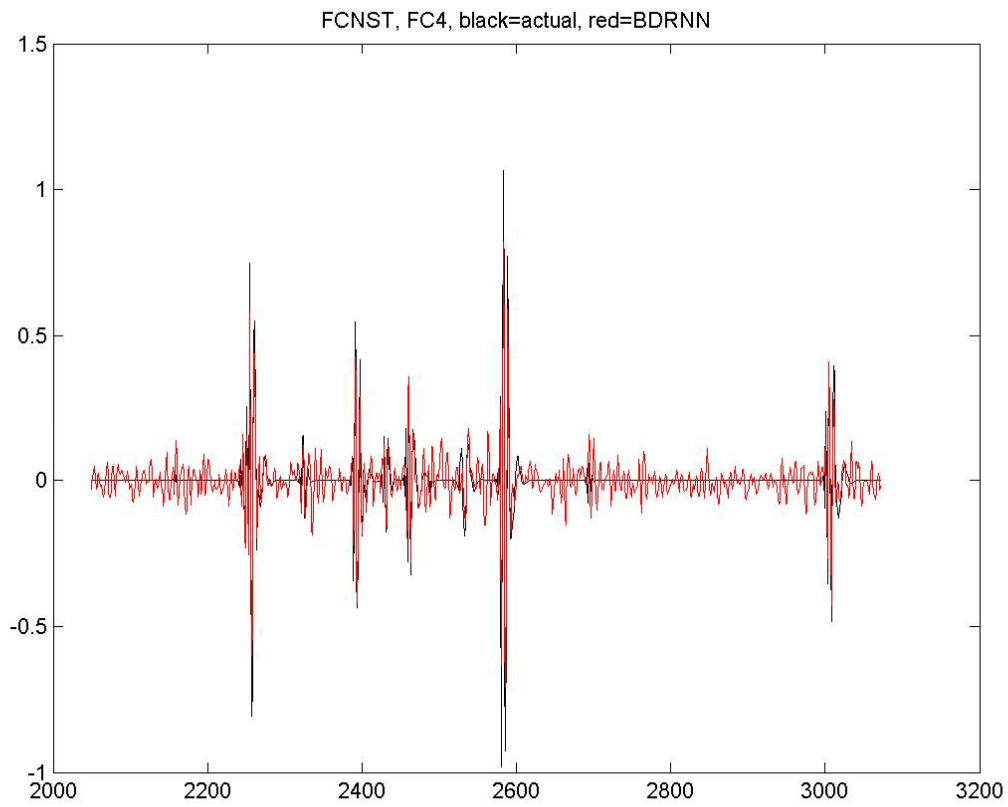
14^η Επανάληψη



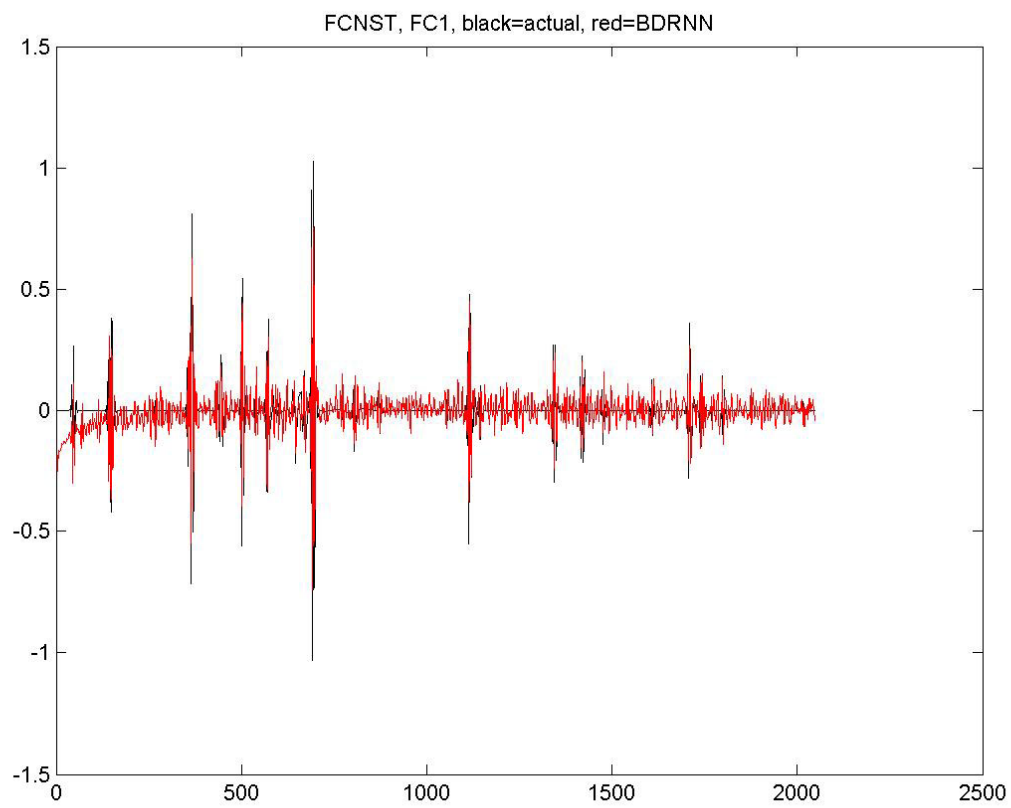
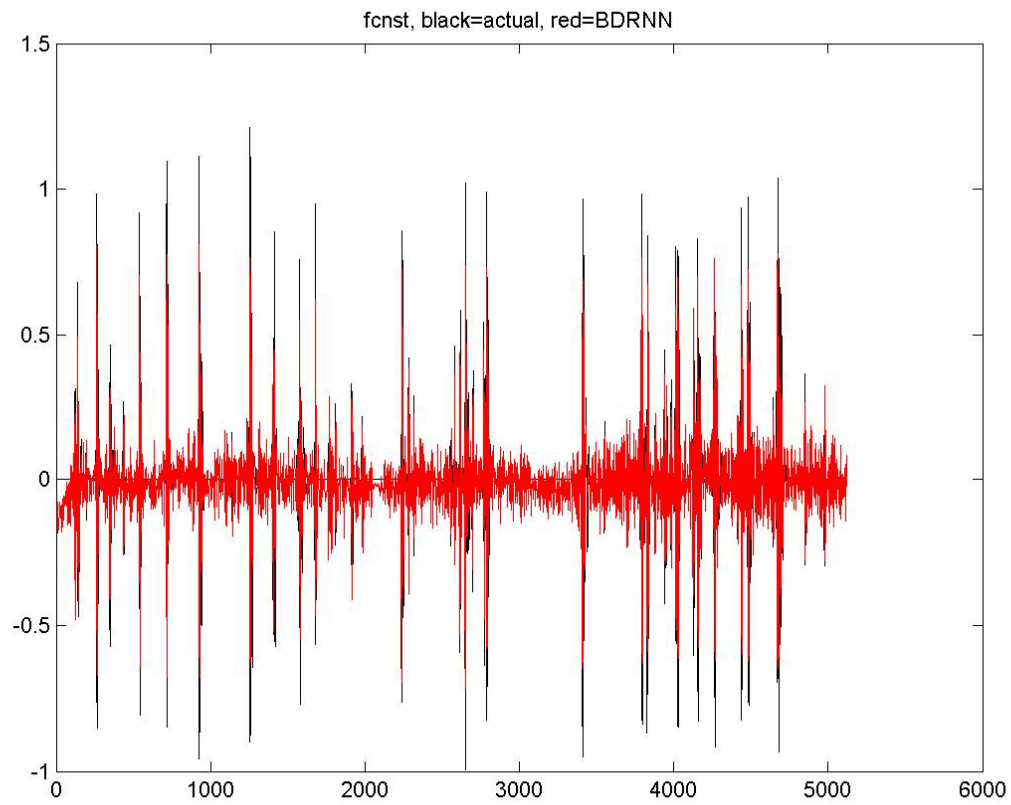


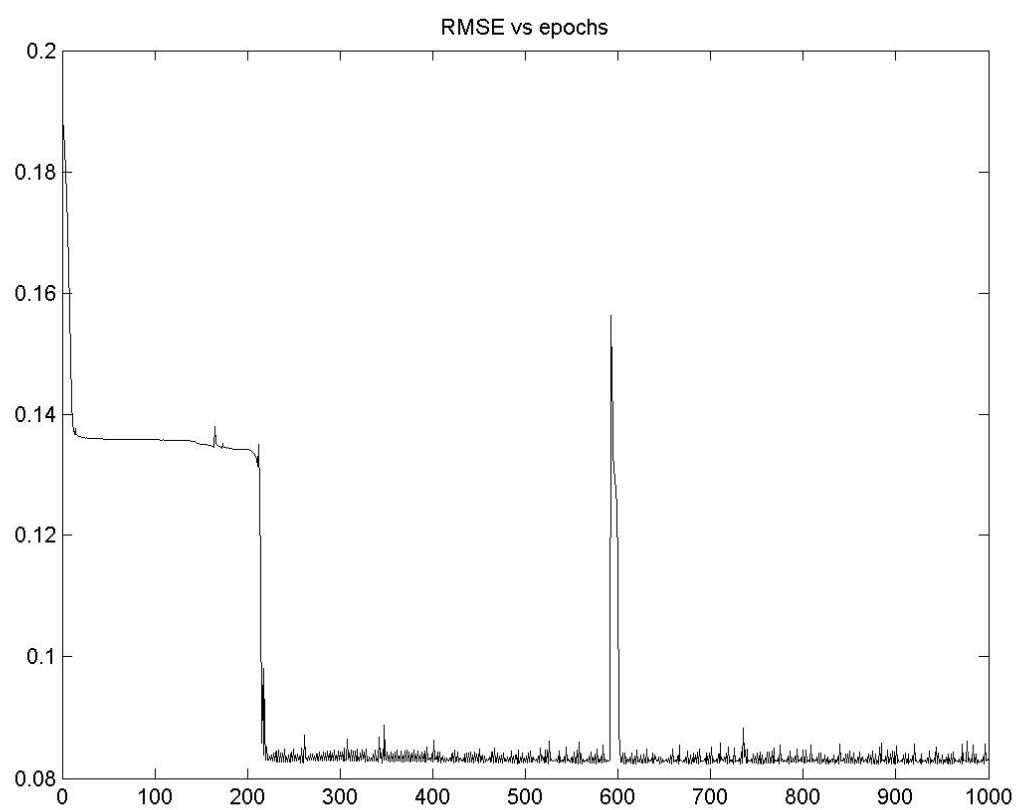
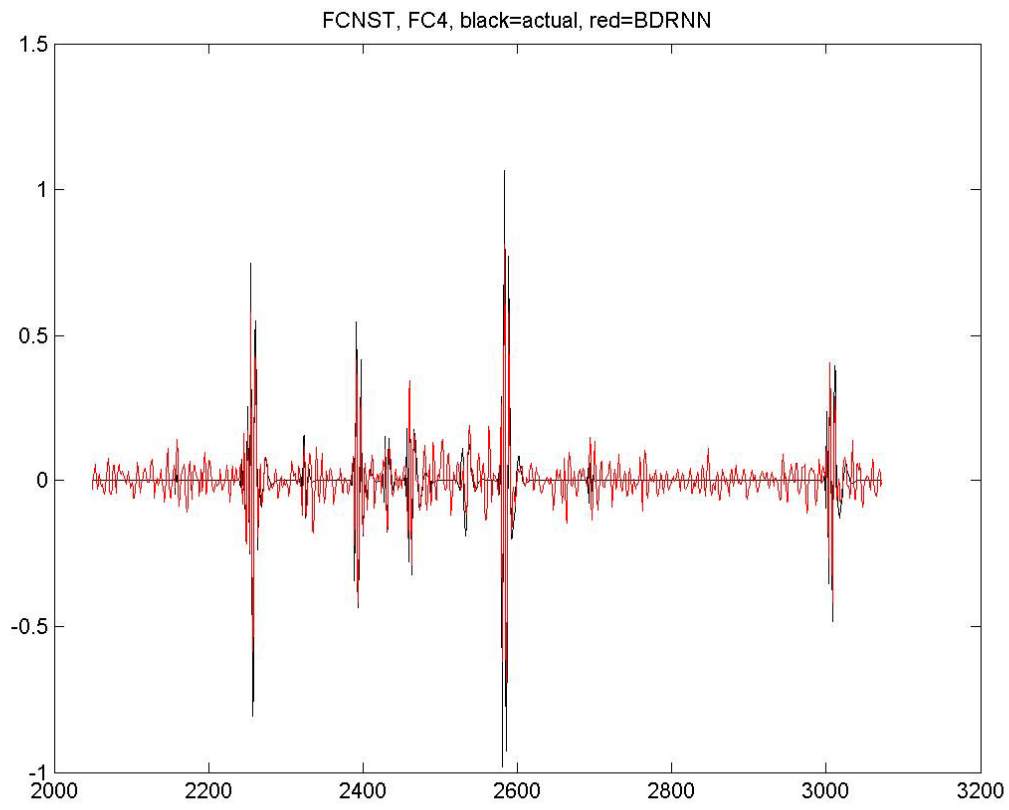
15^η Επανάληψη



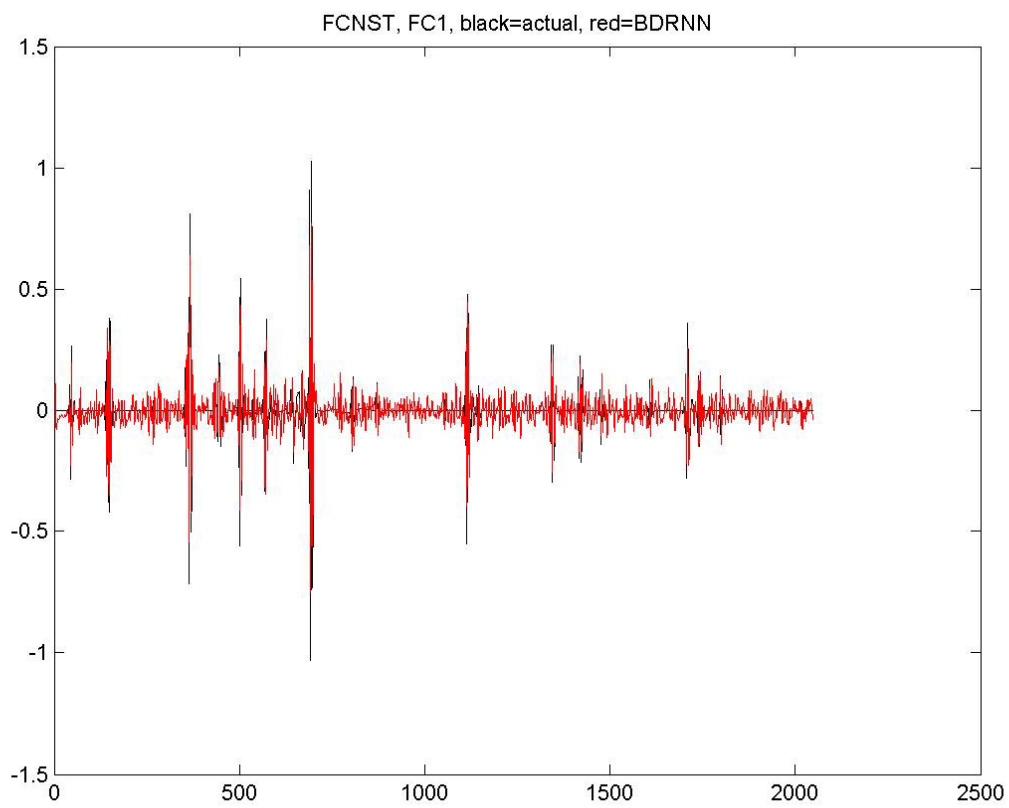
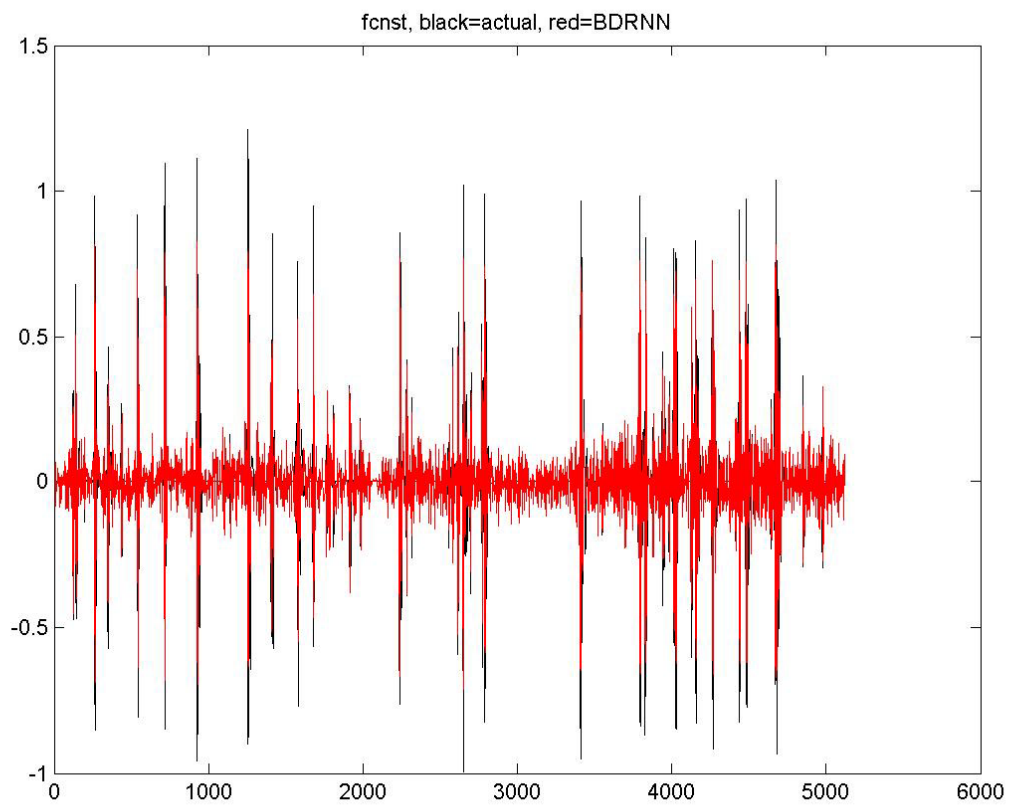


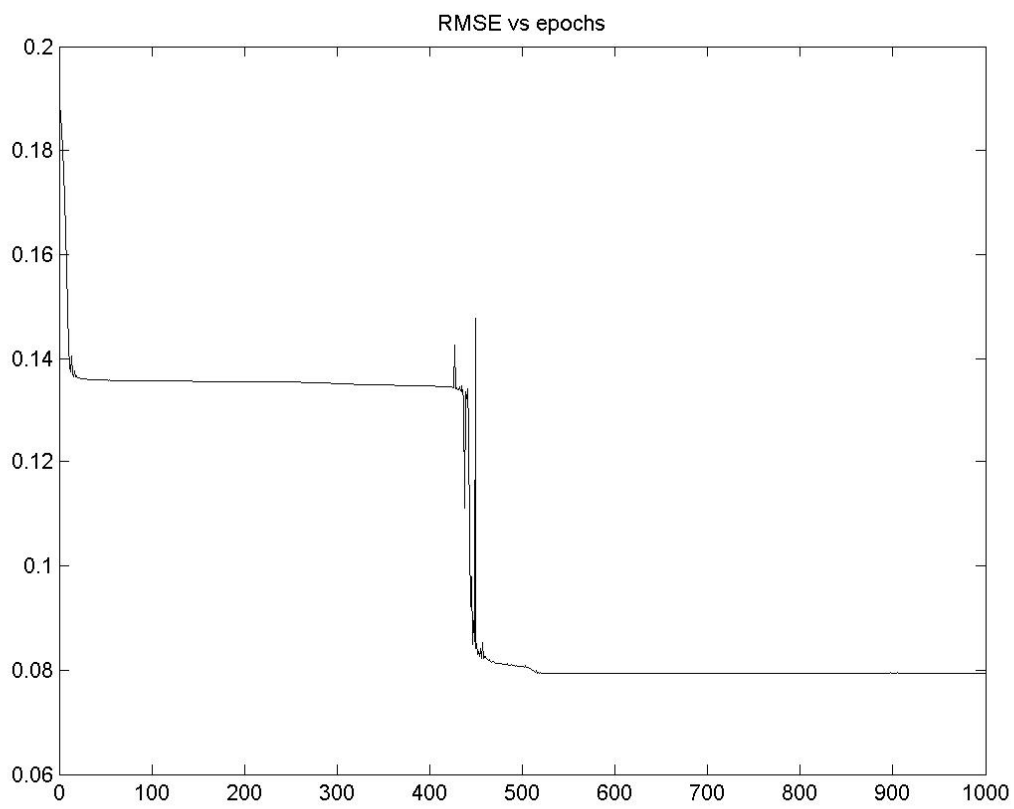
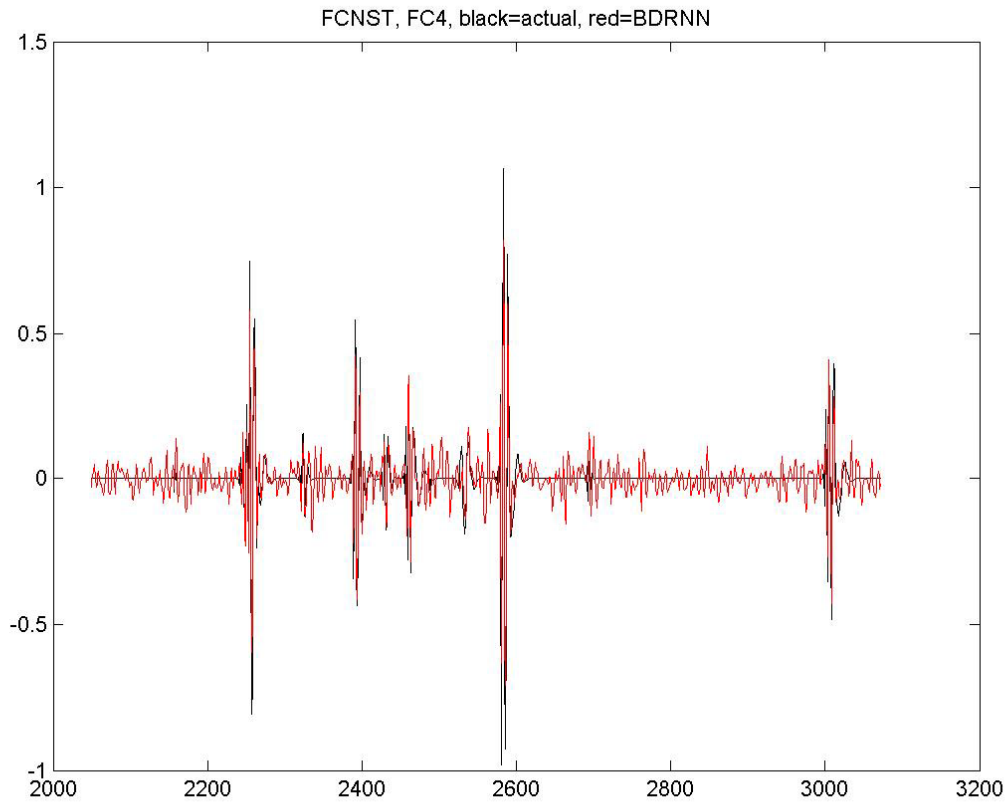
16^η Επανάληψη



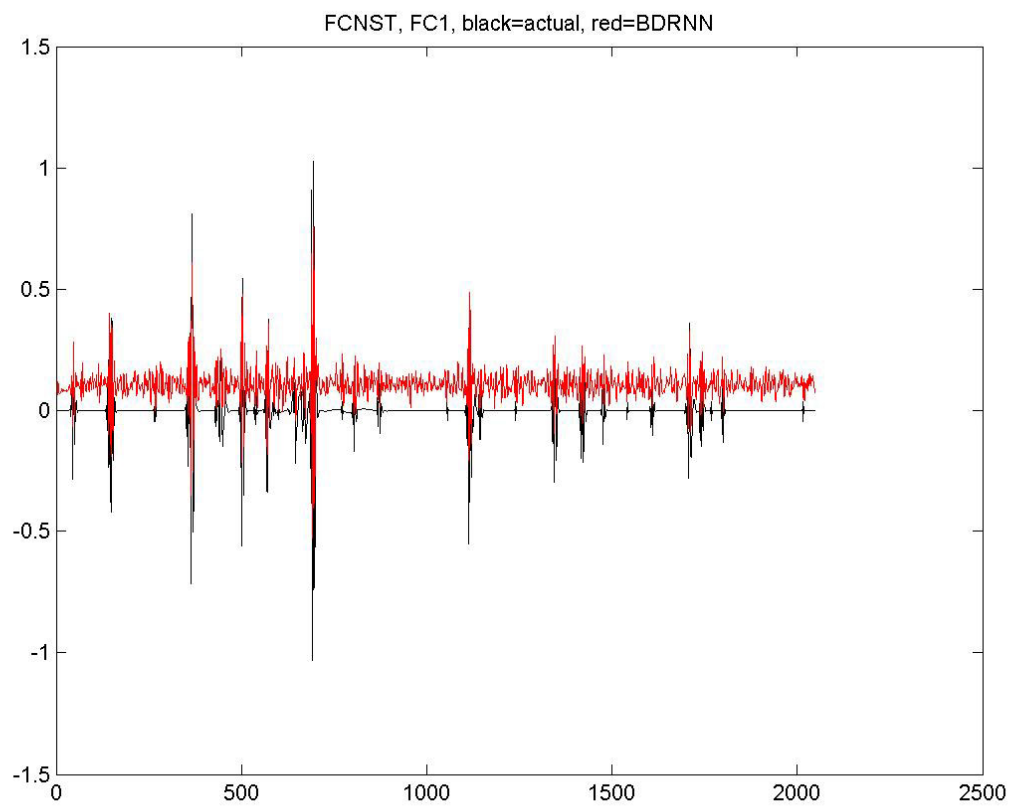
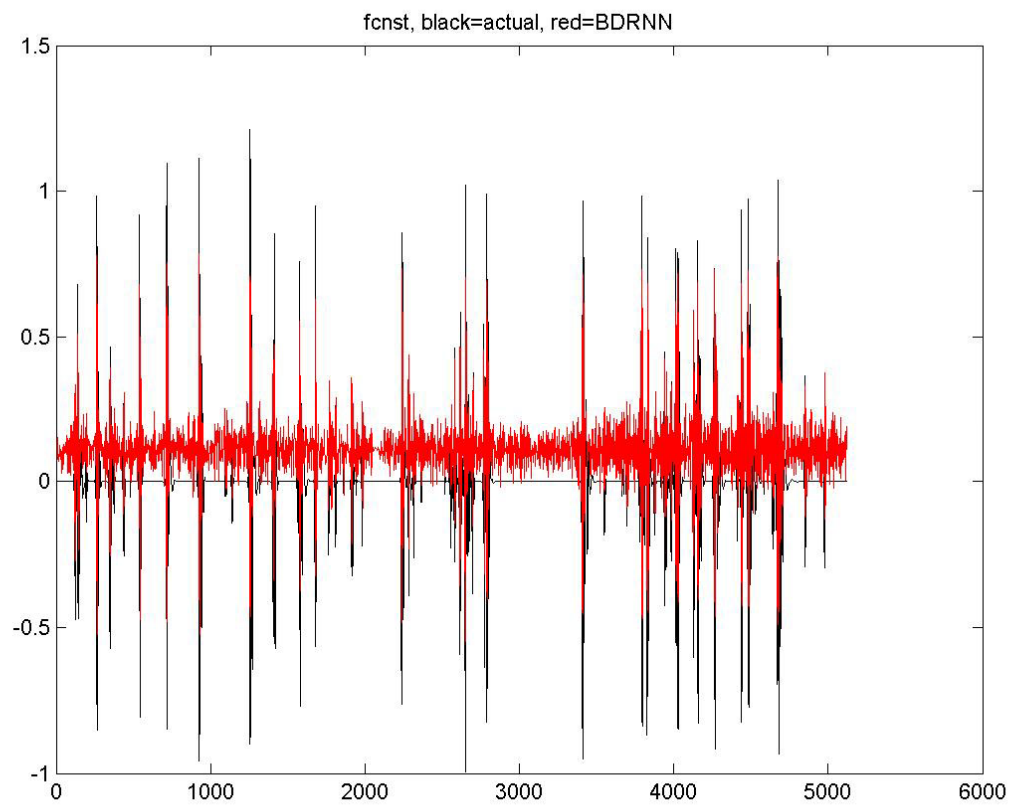


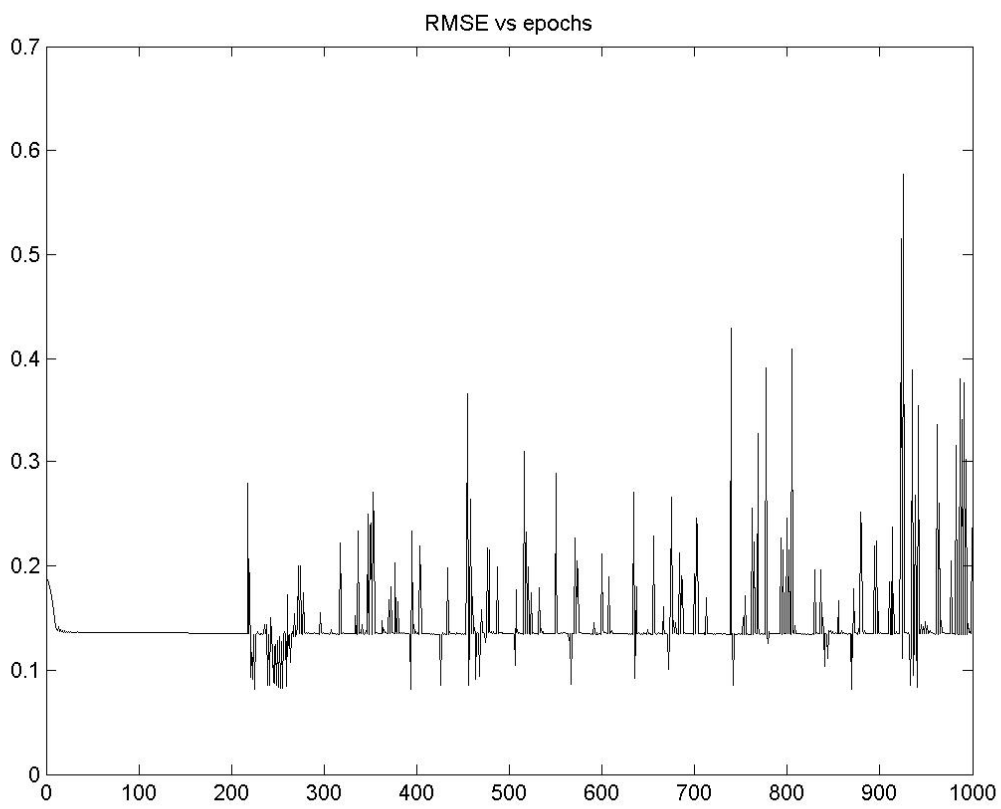
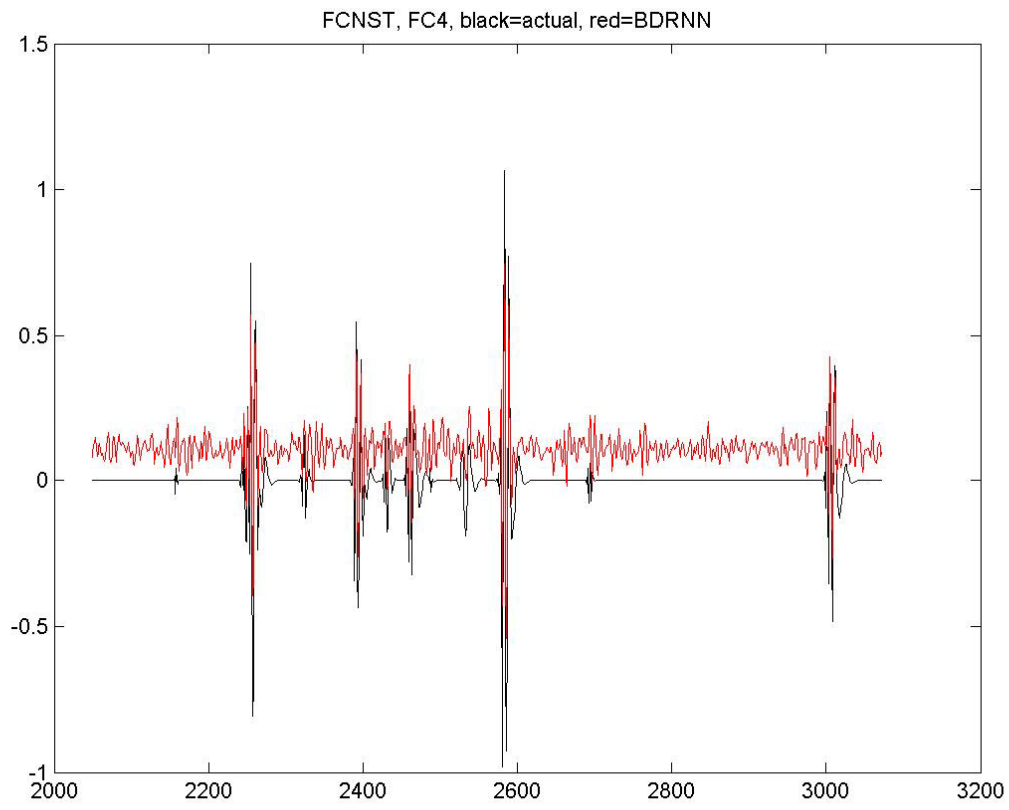
17^η Επανάληψη



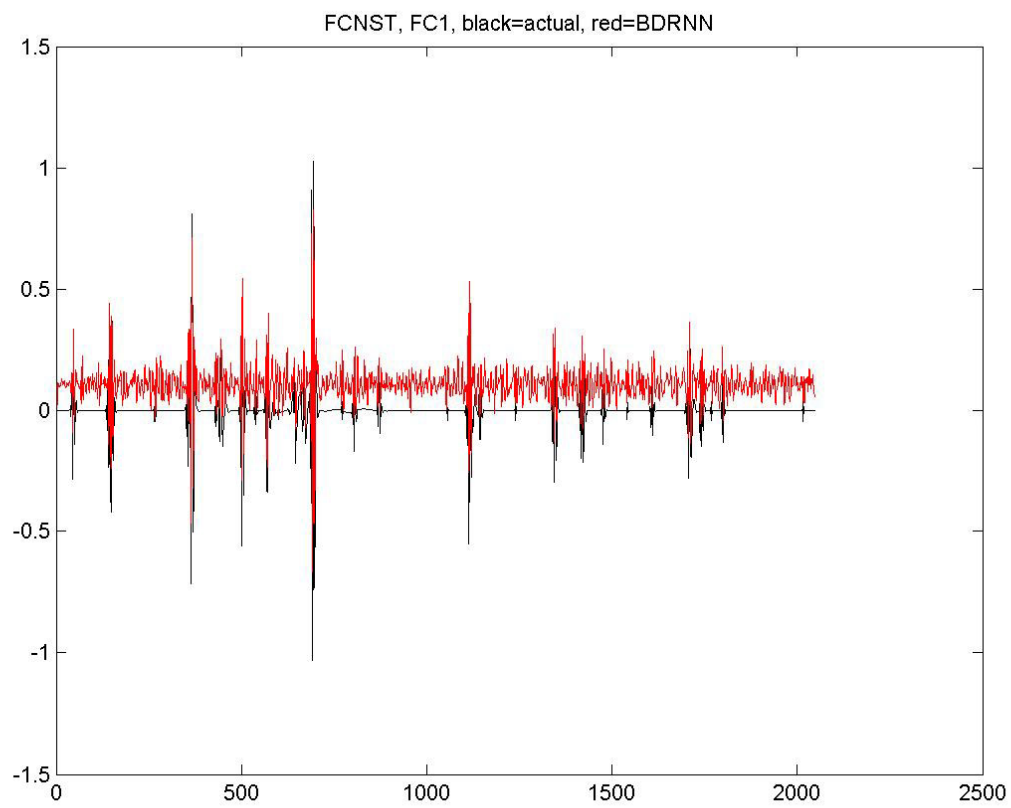
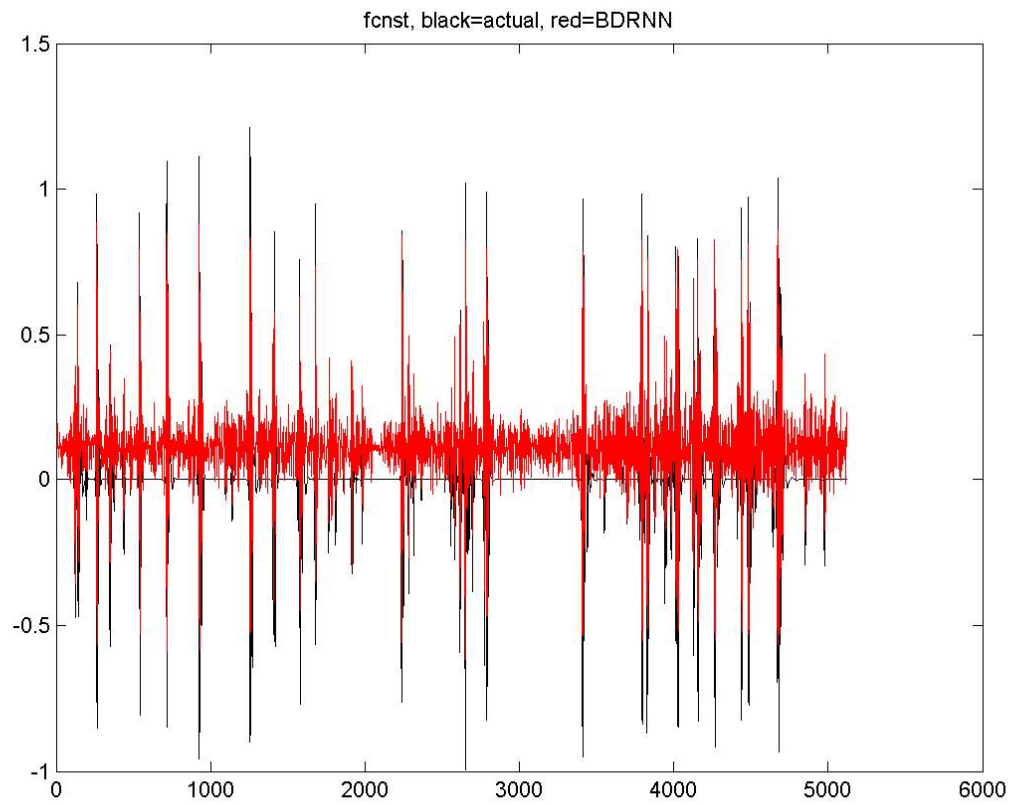


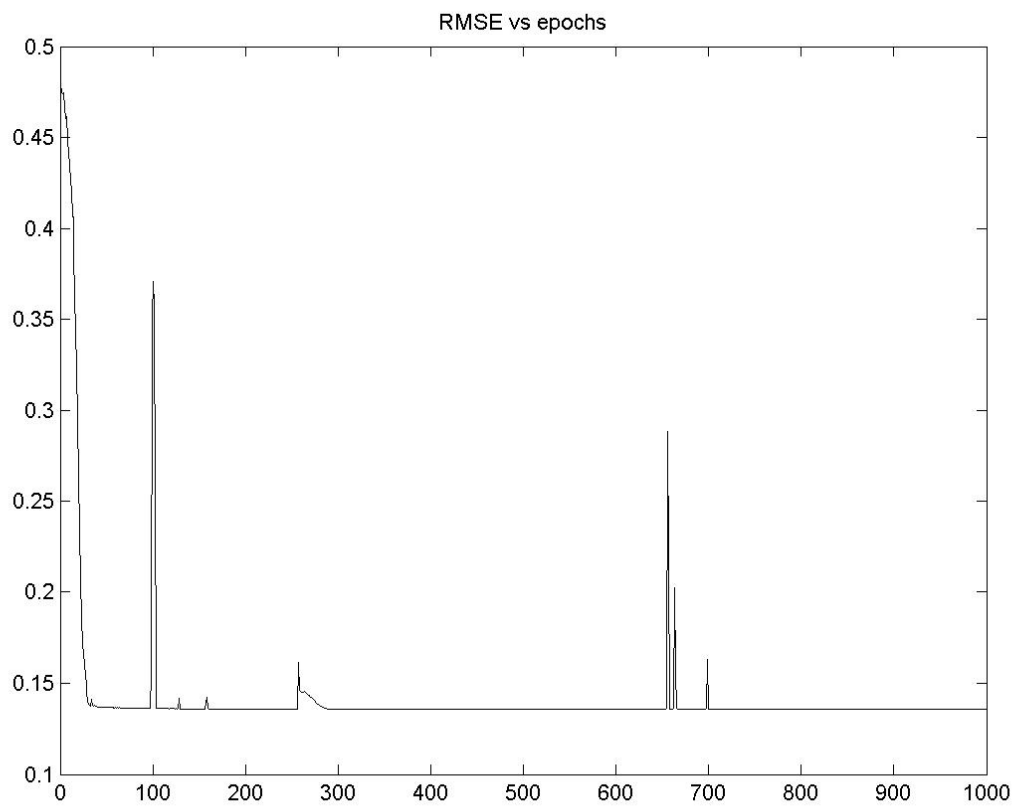
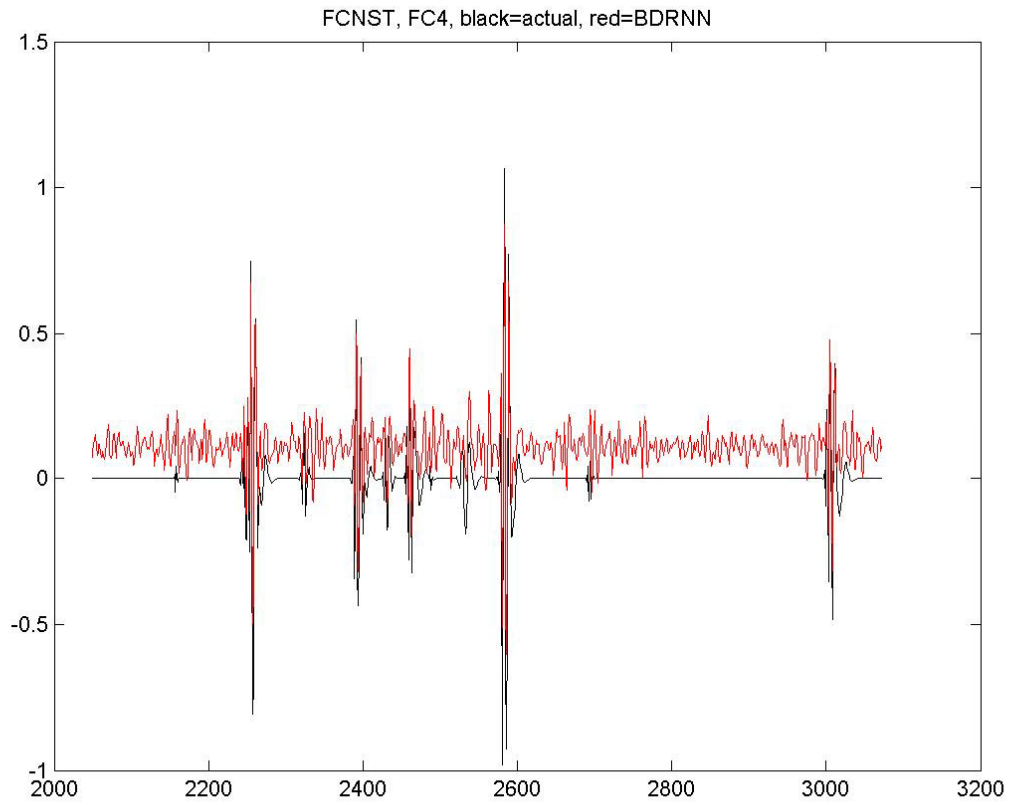
18^η Επανάληψη



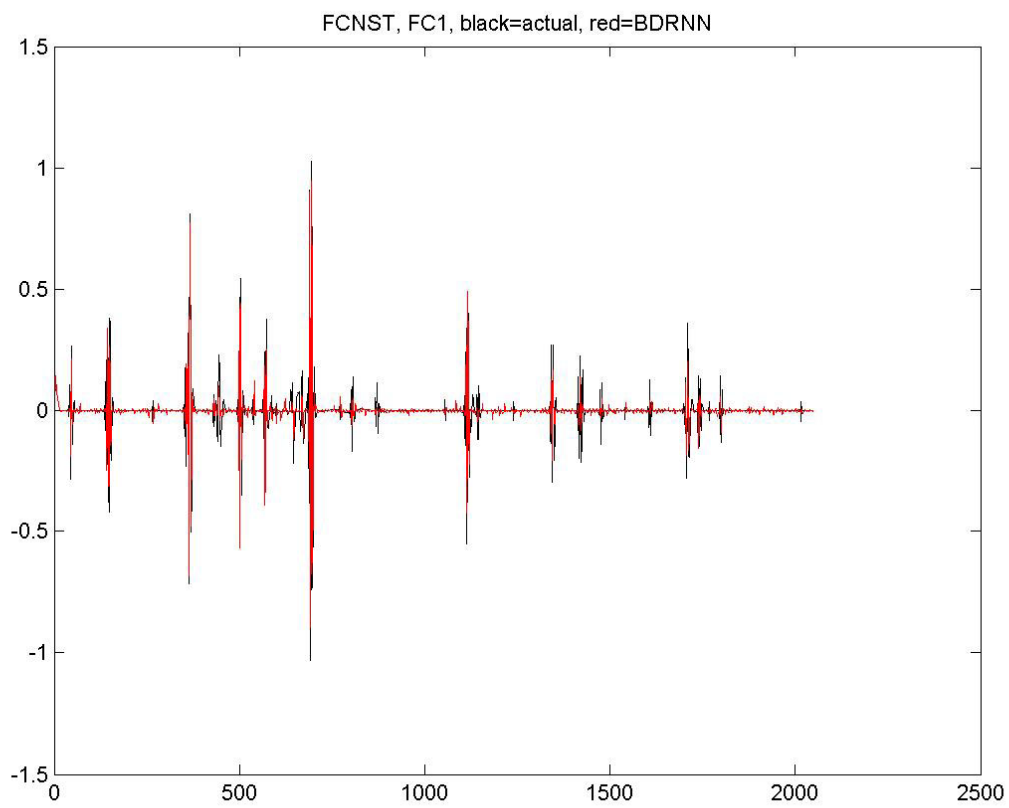
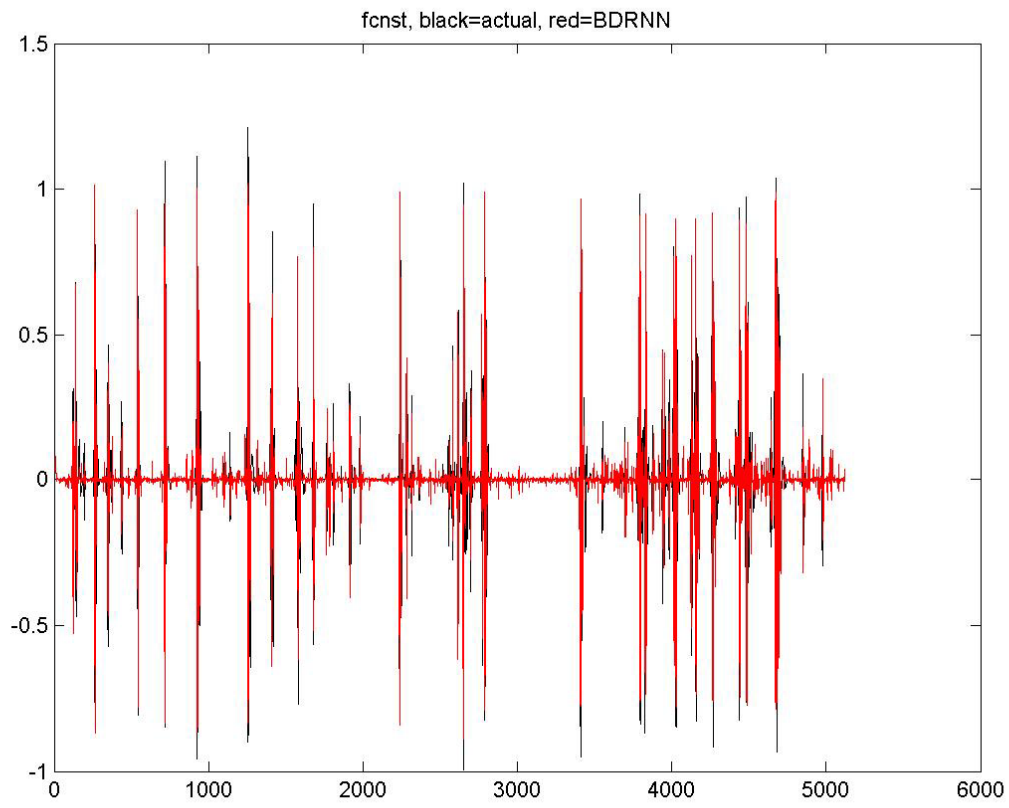


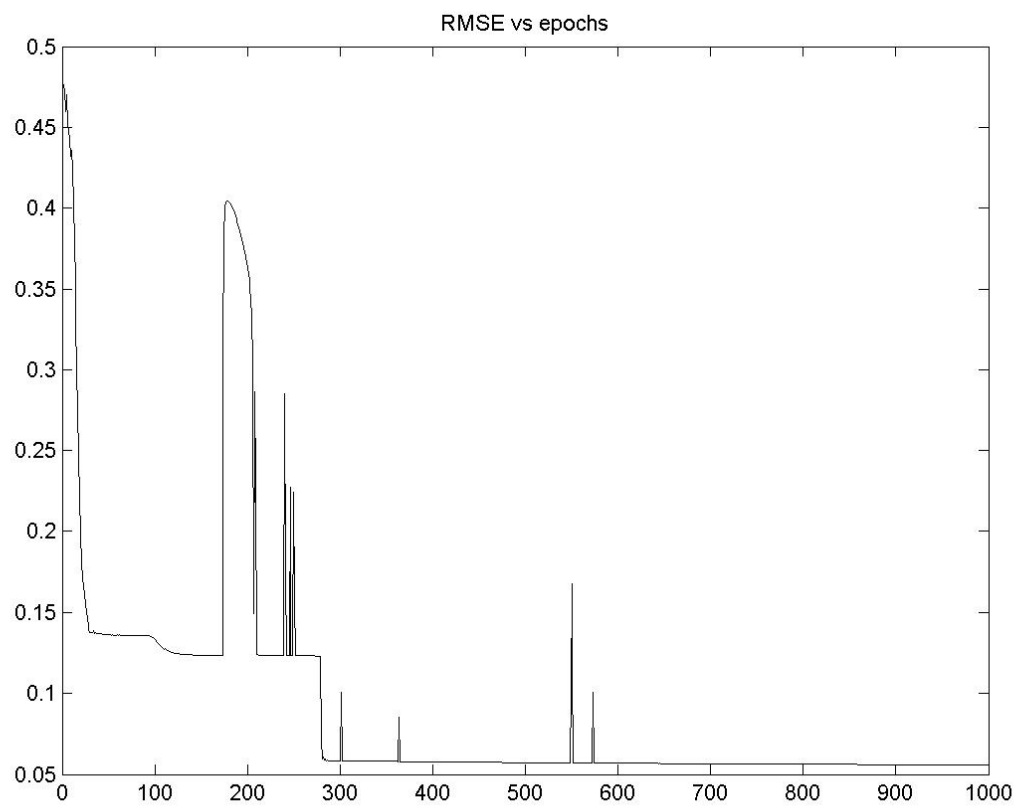
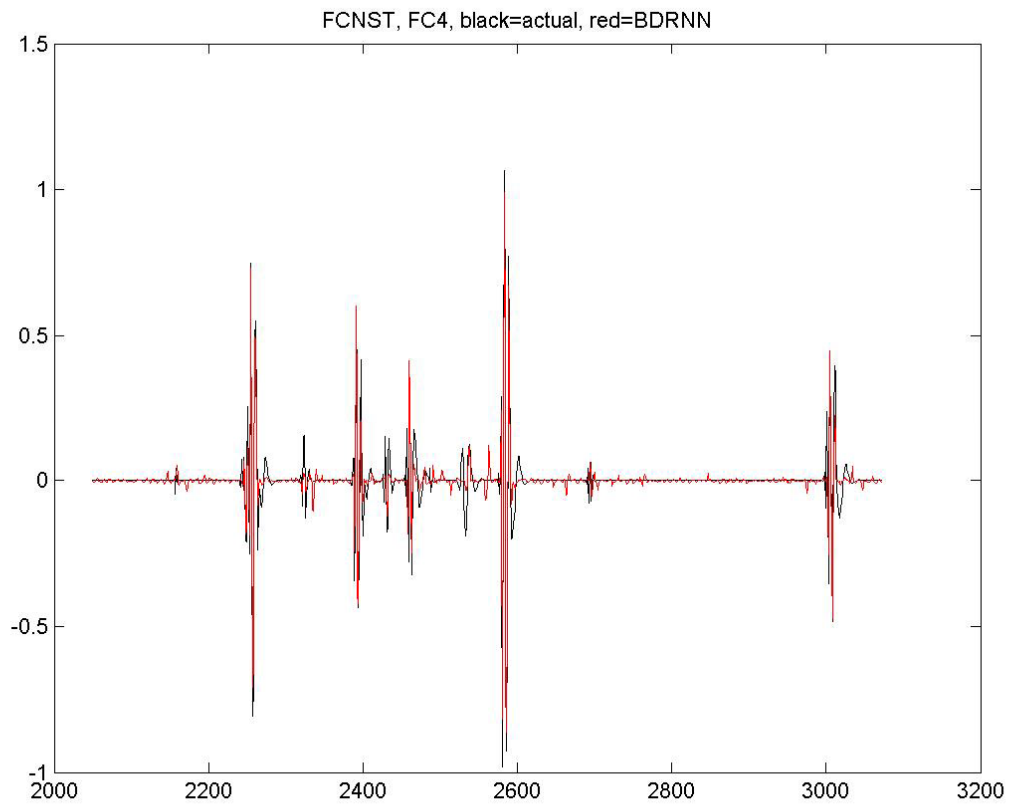
19^η Επανάληψη



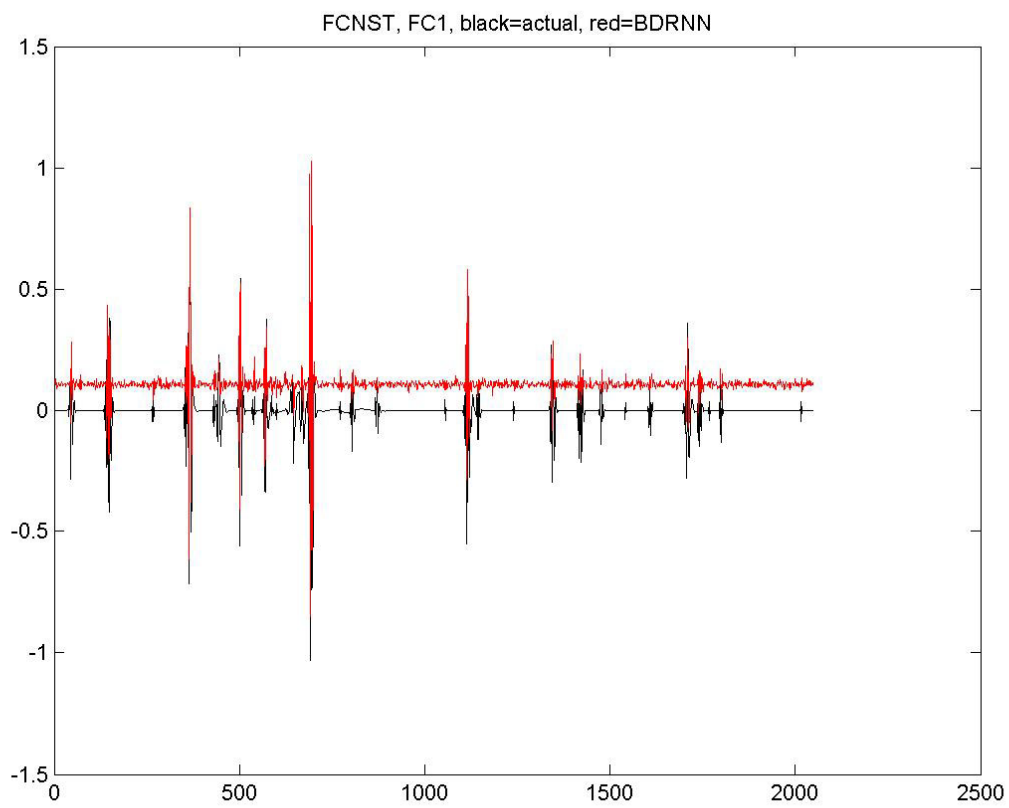
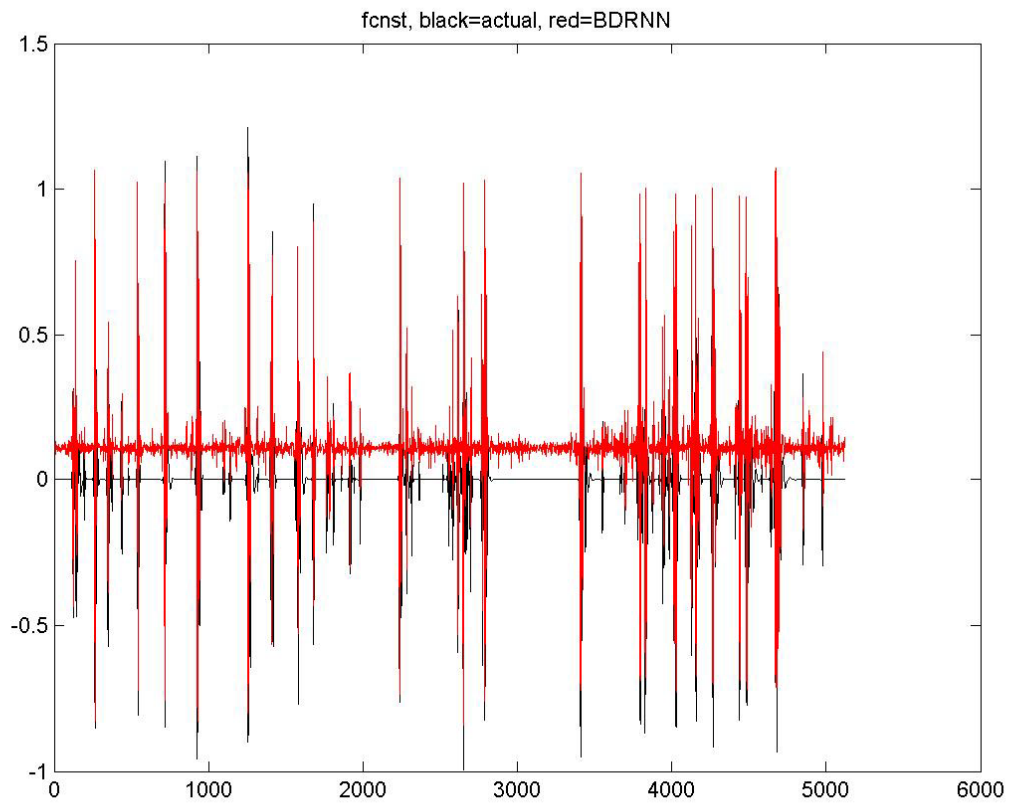


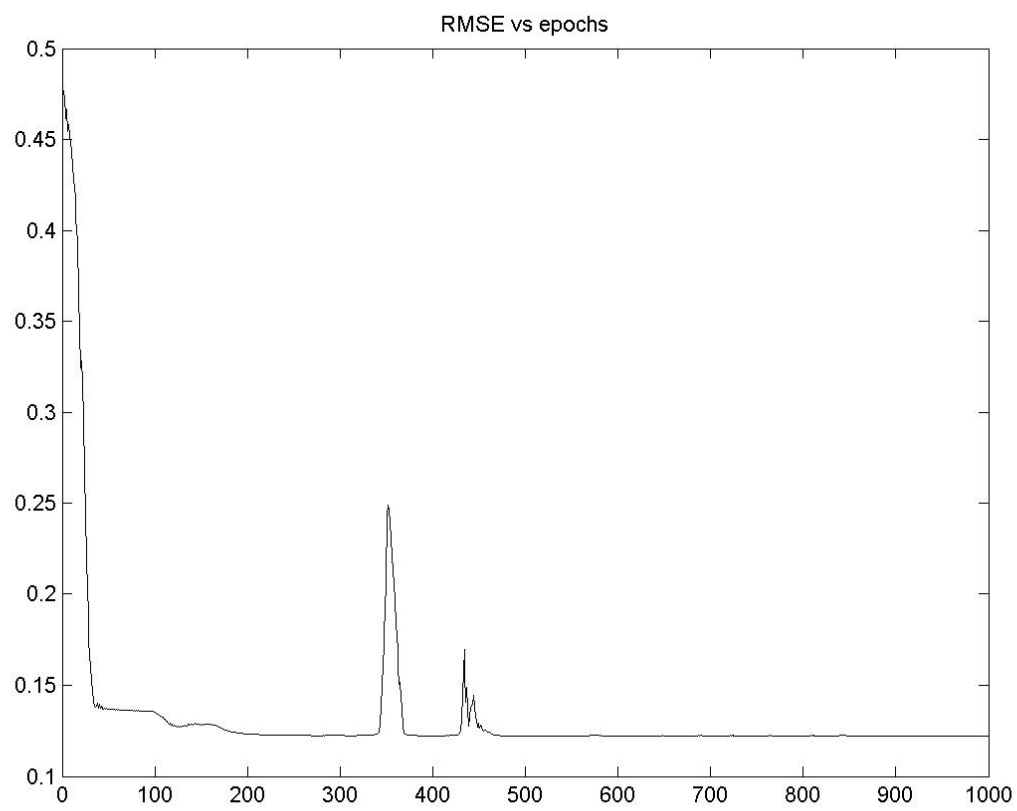
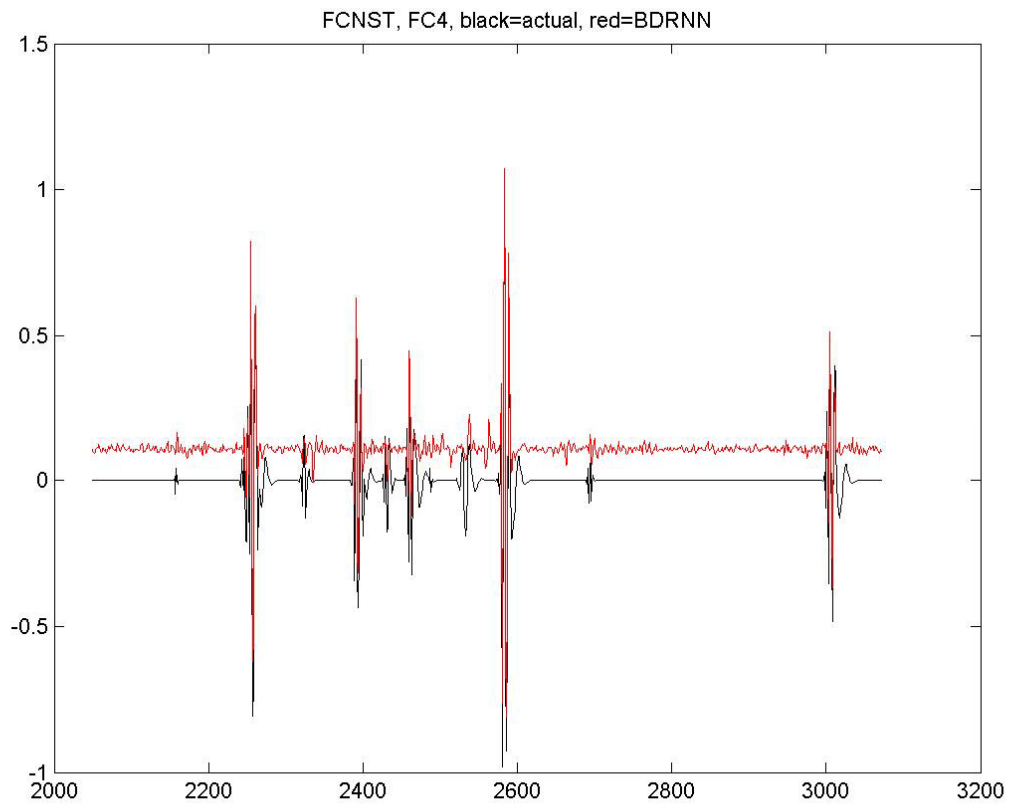
20th Επανάληψη



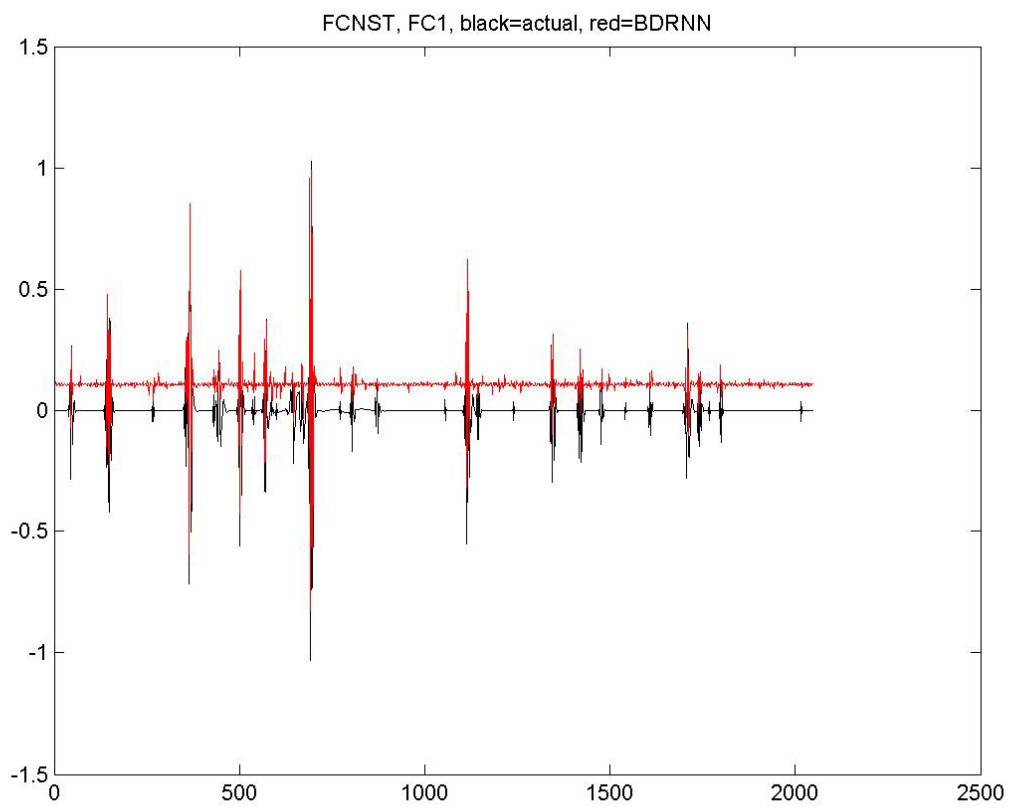
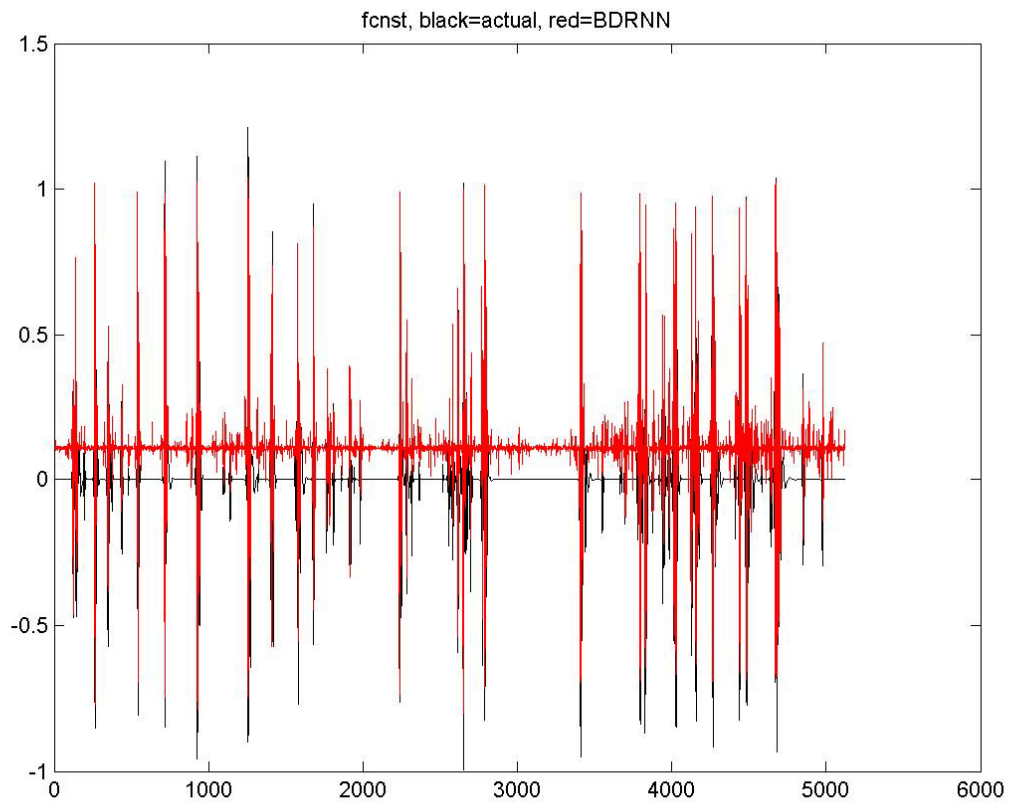


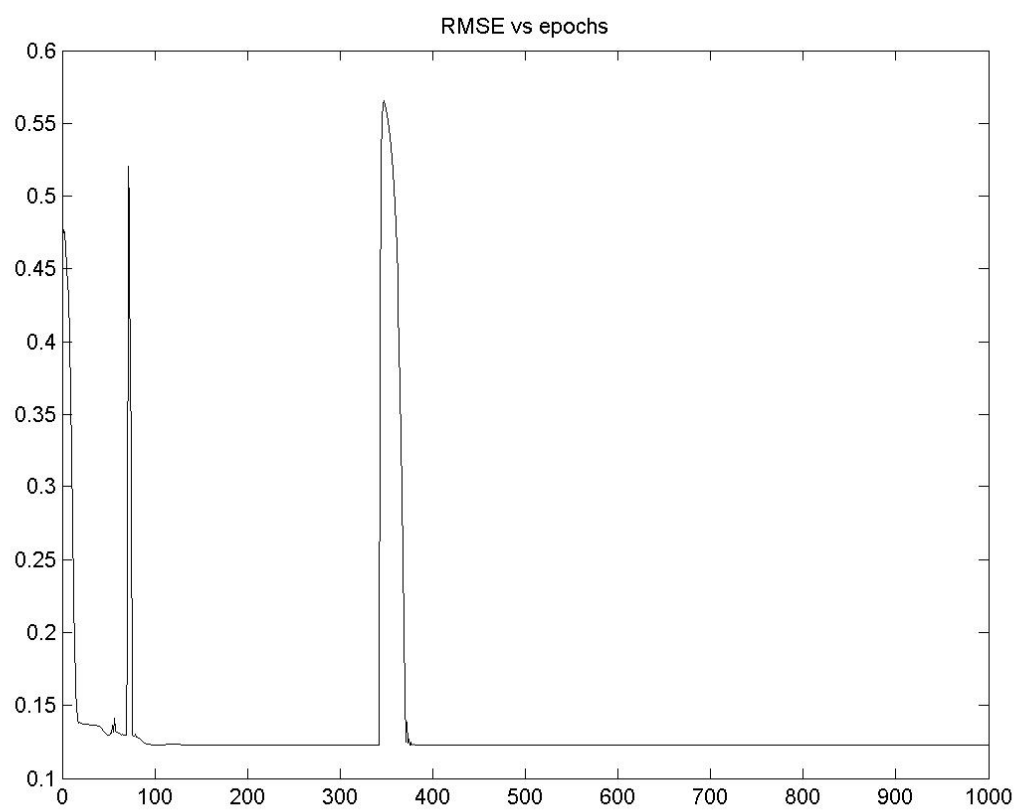
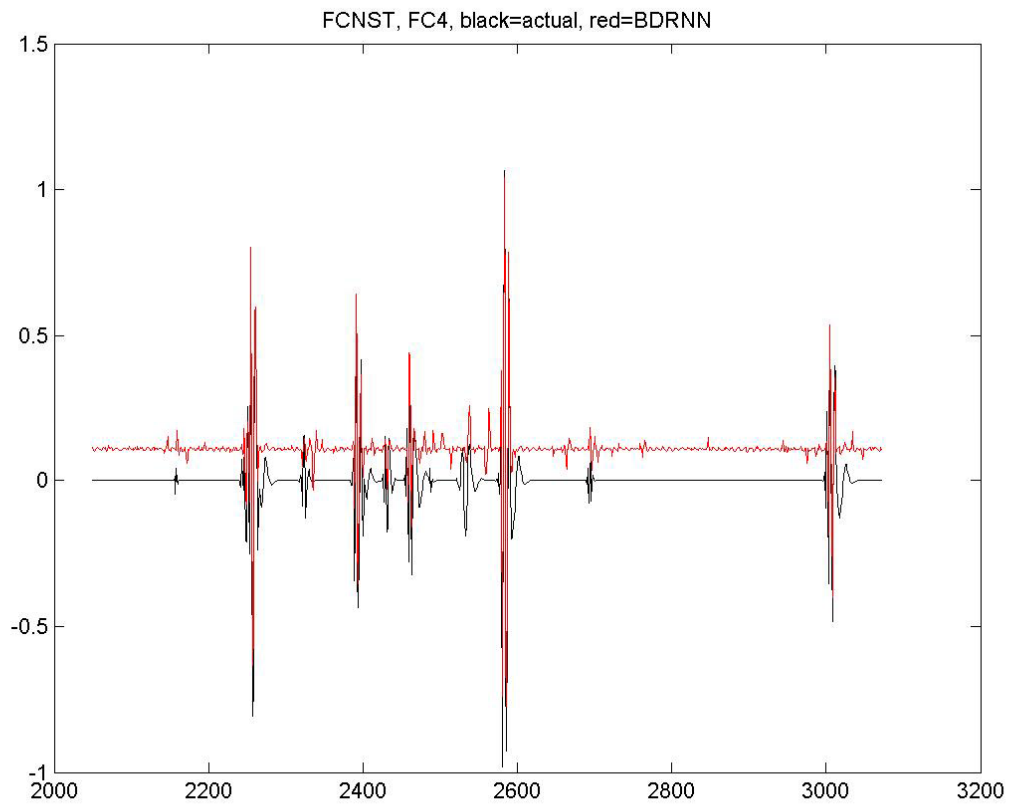
21^η Επανάληψη



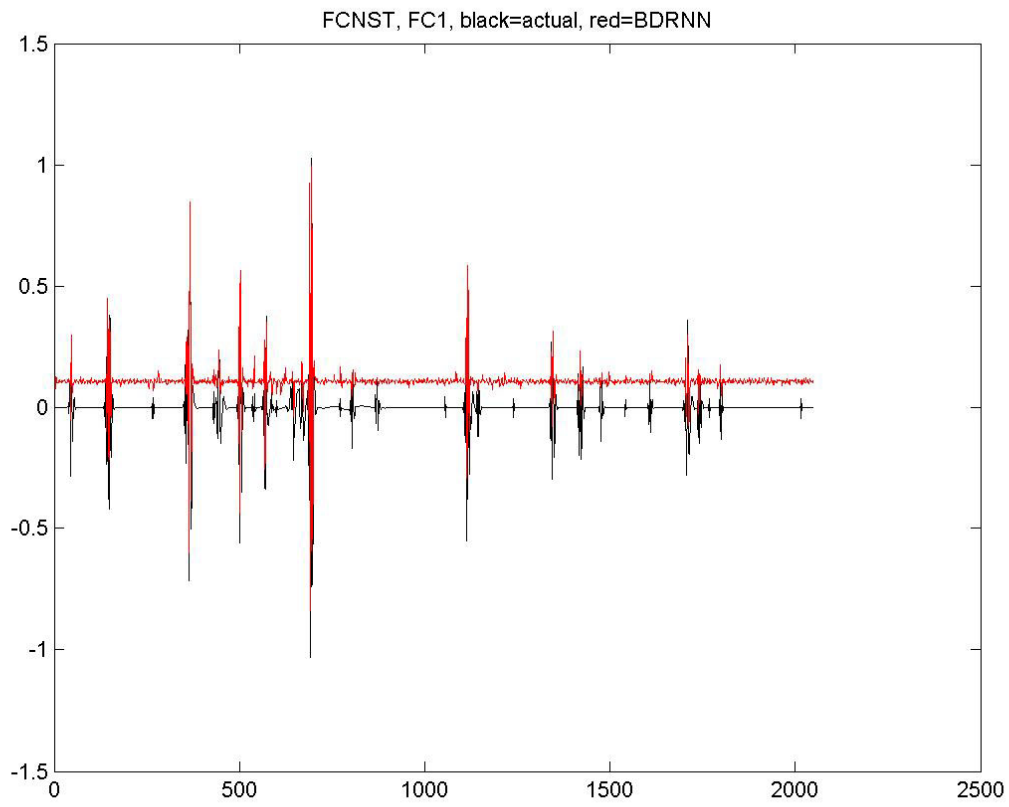
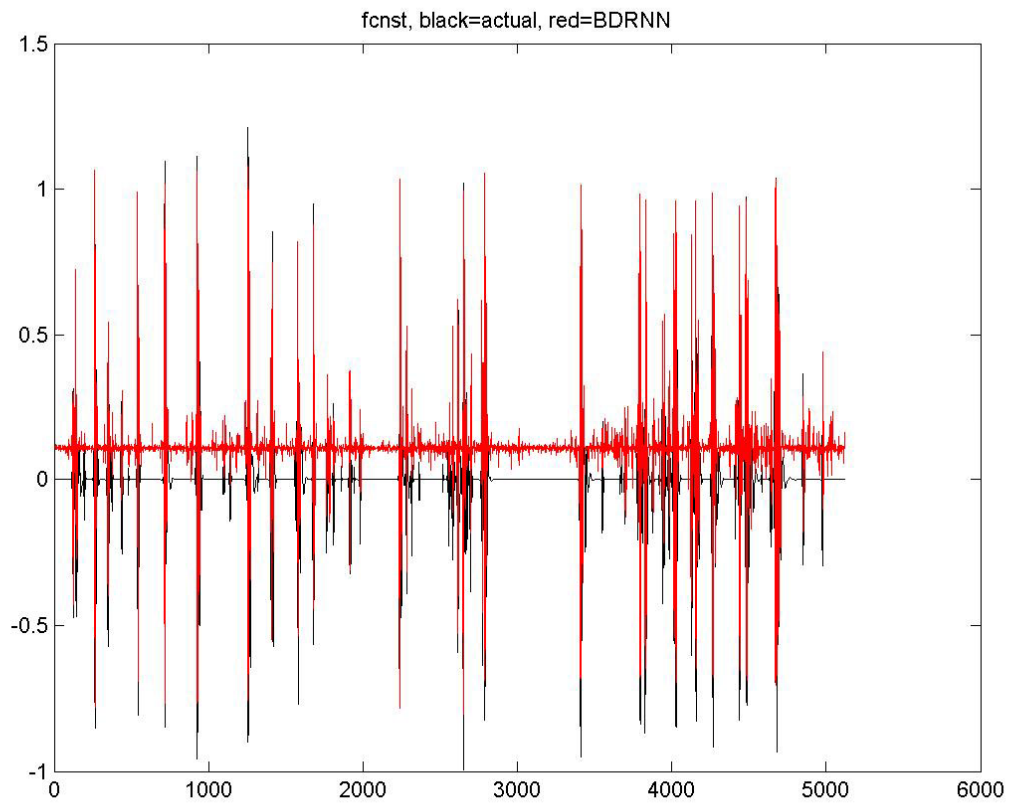


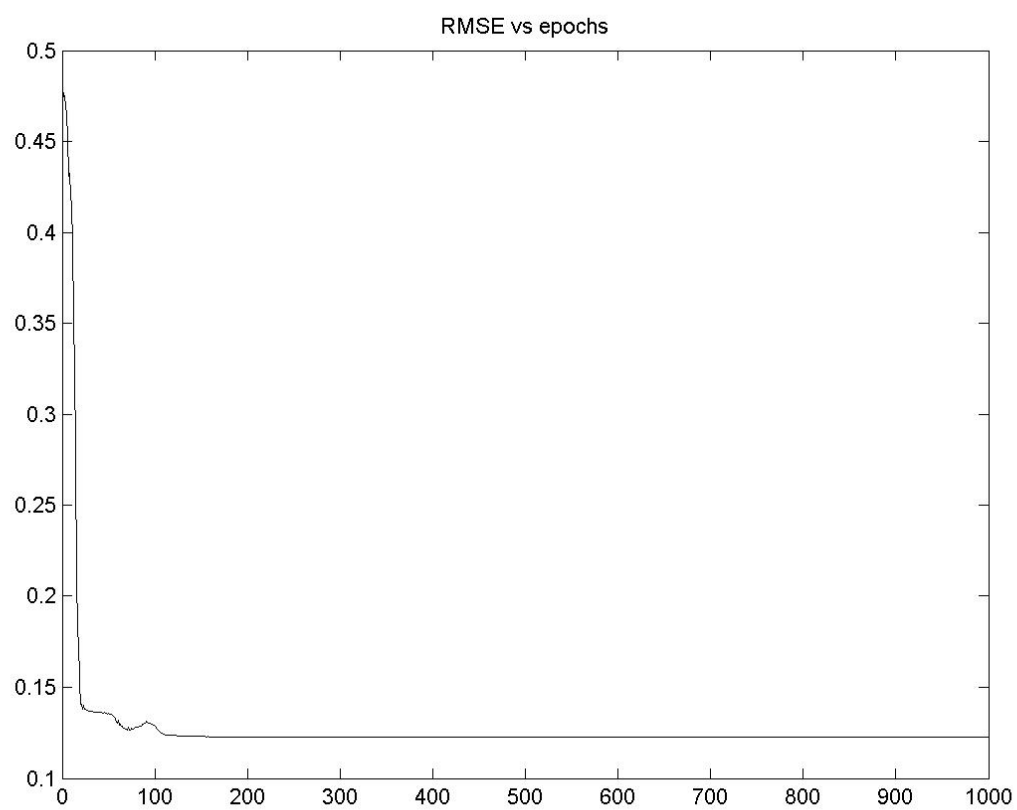
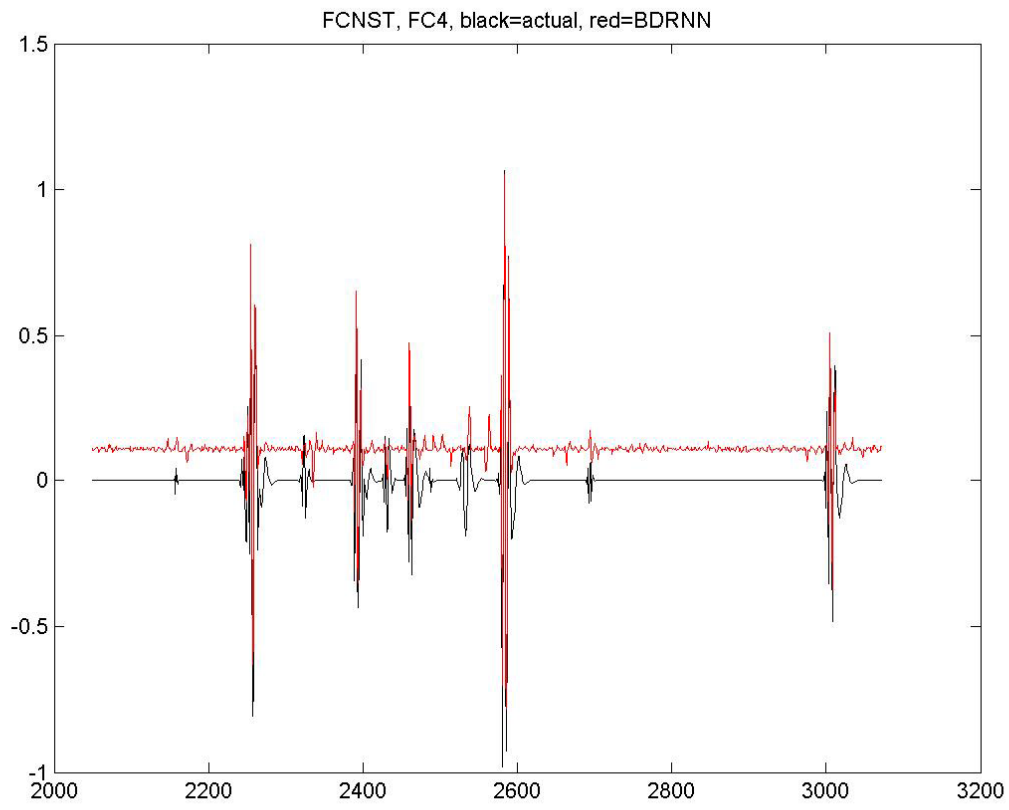
22^η Επανάληψη



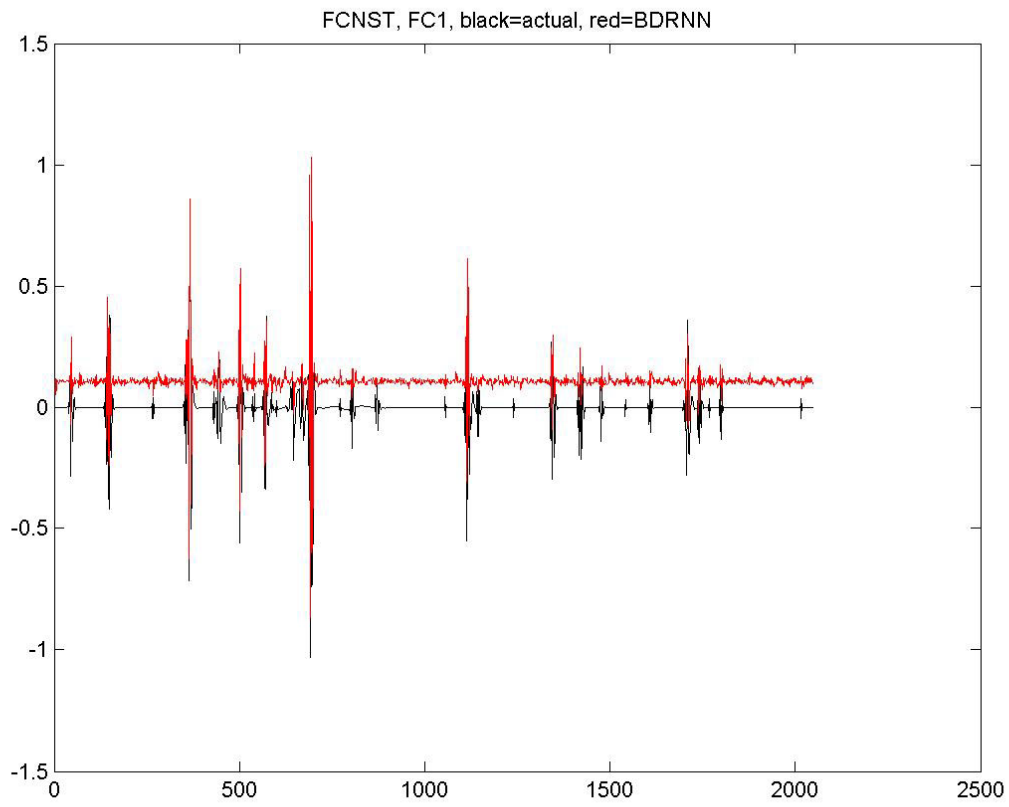
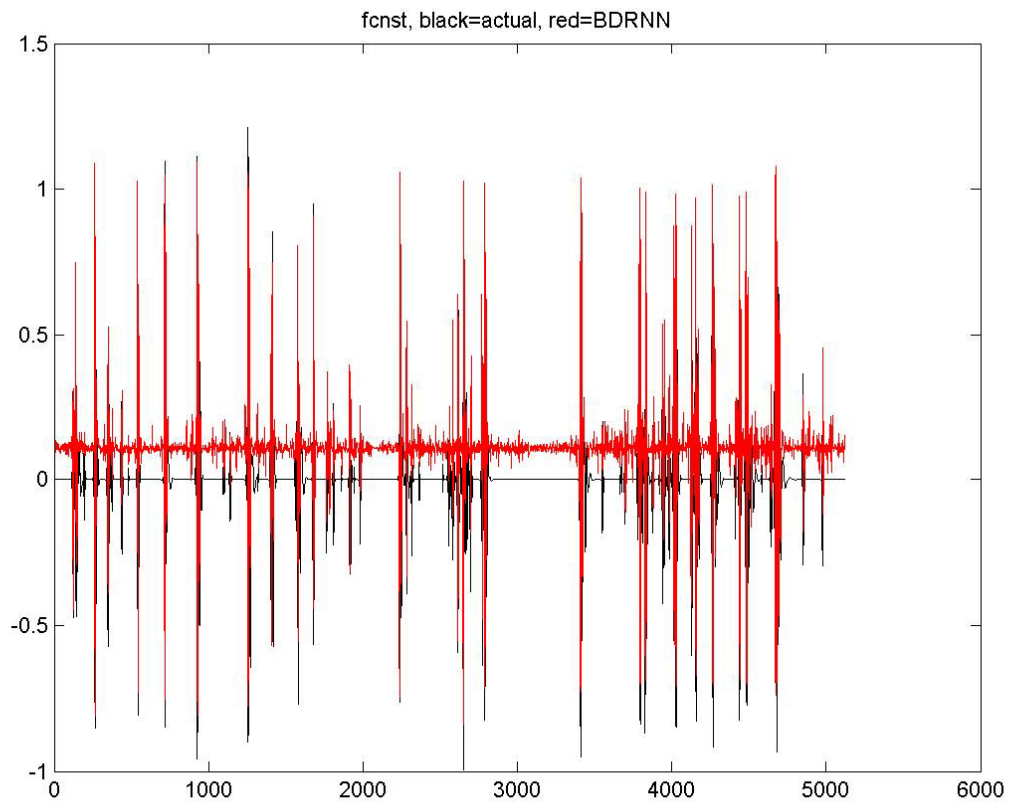


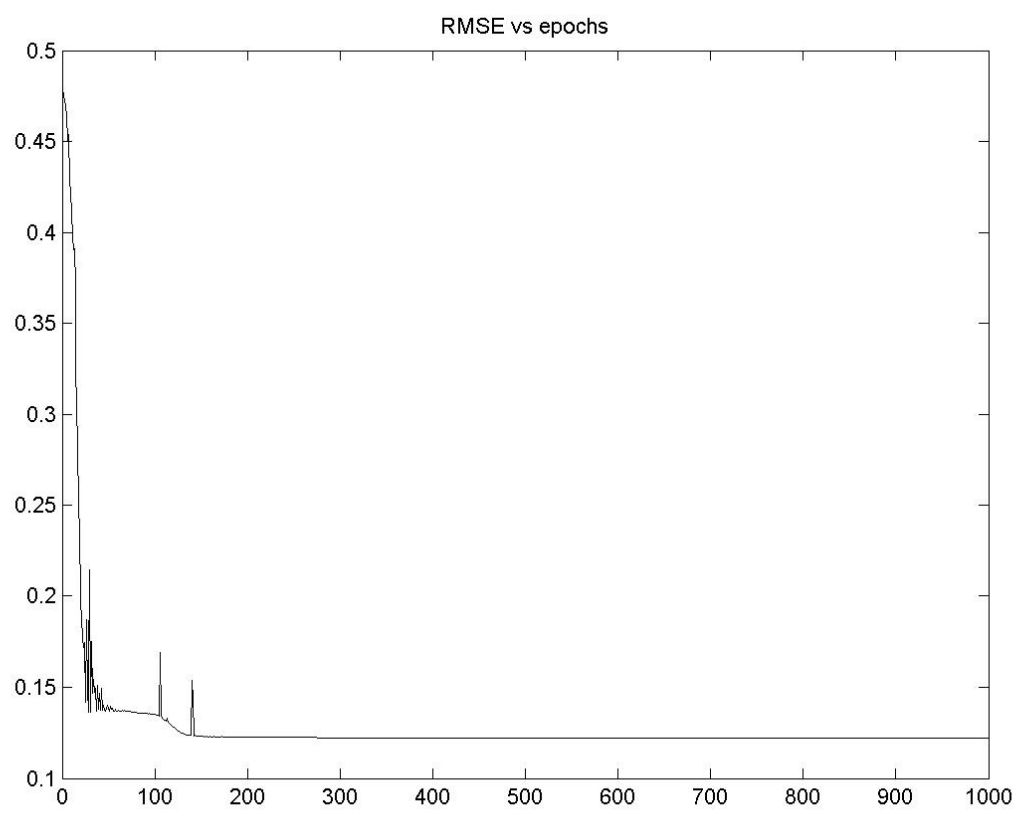
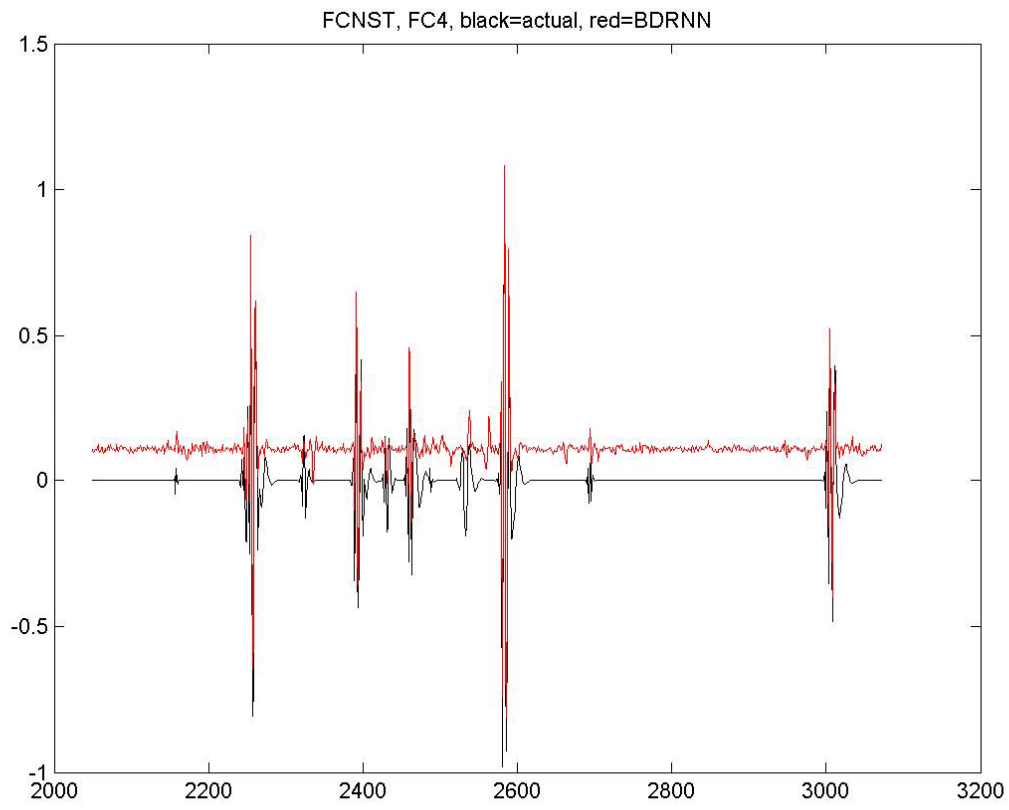
23^η Επανάληψη



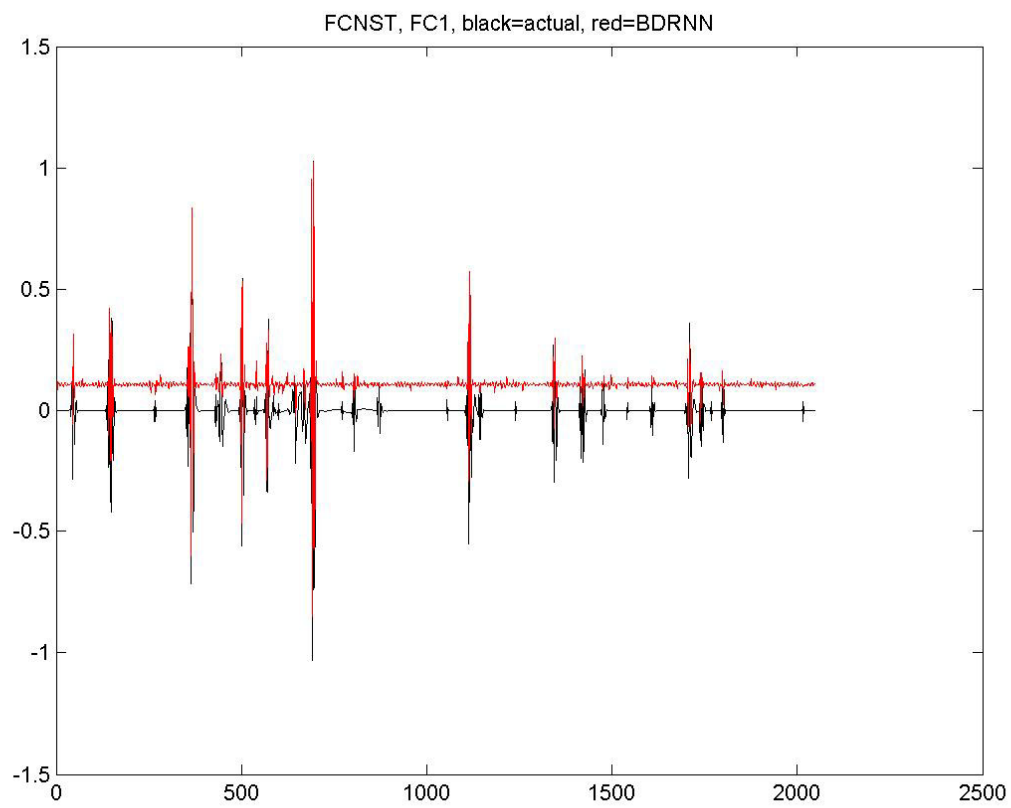
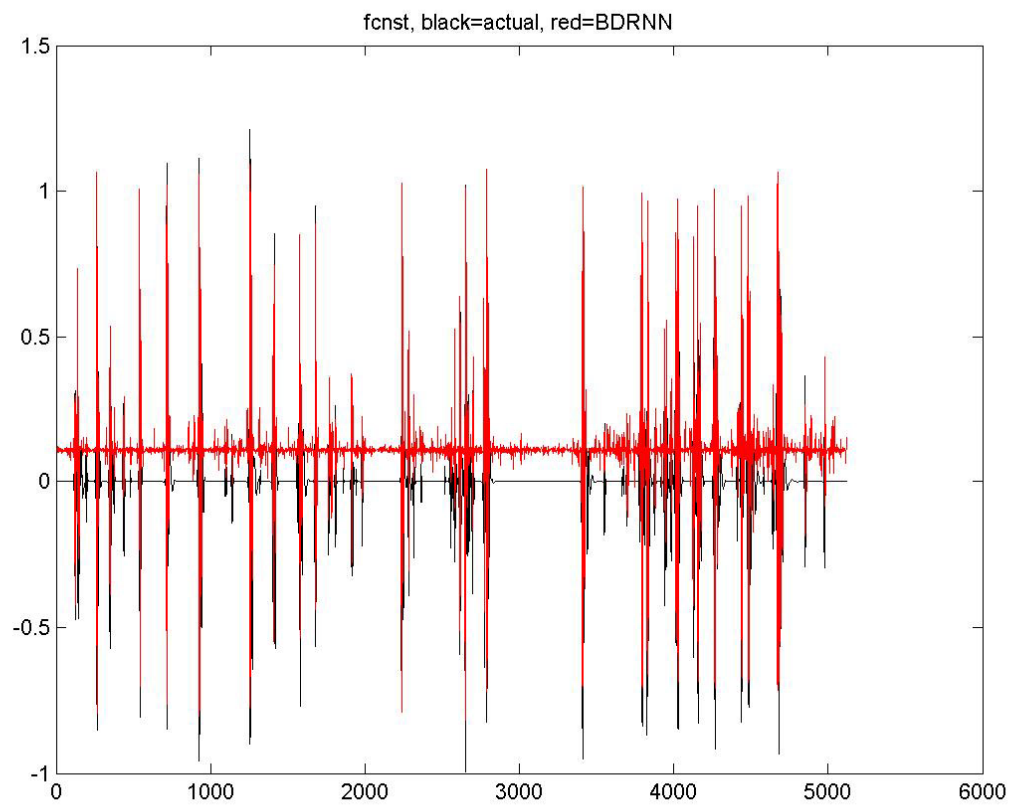


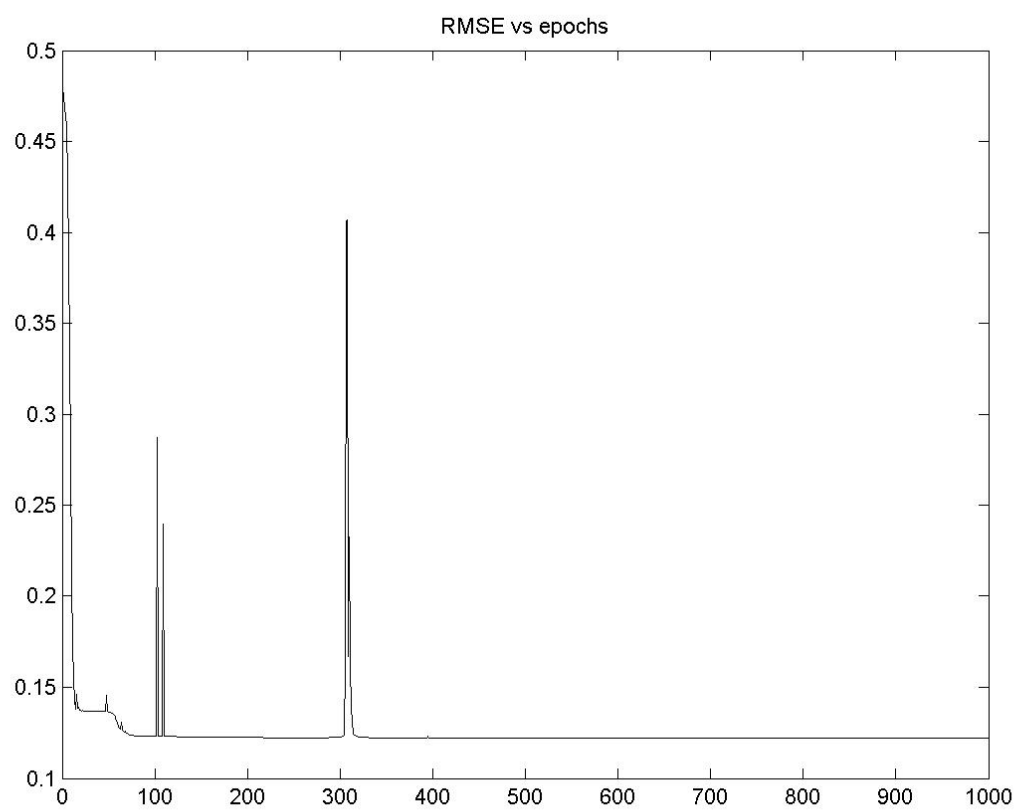
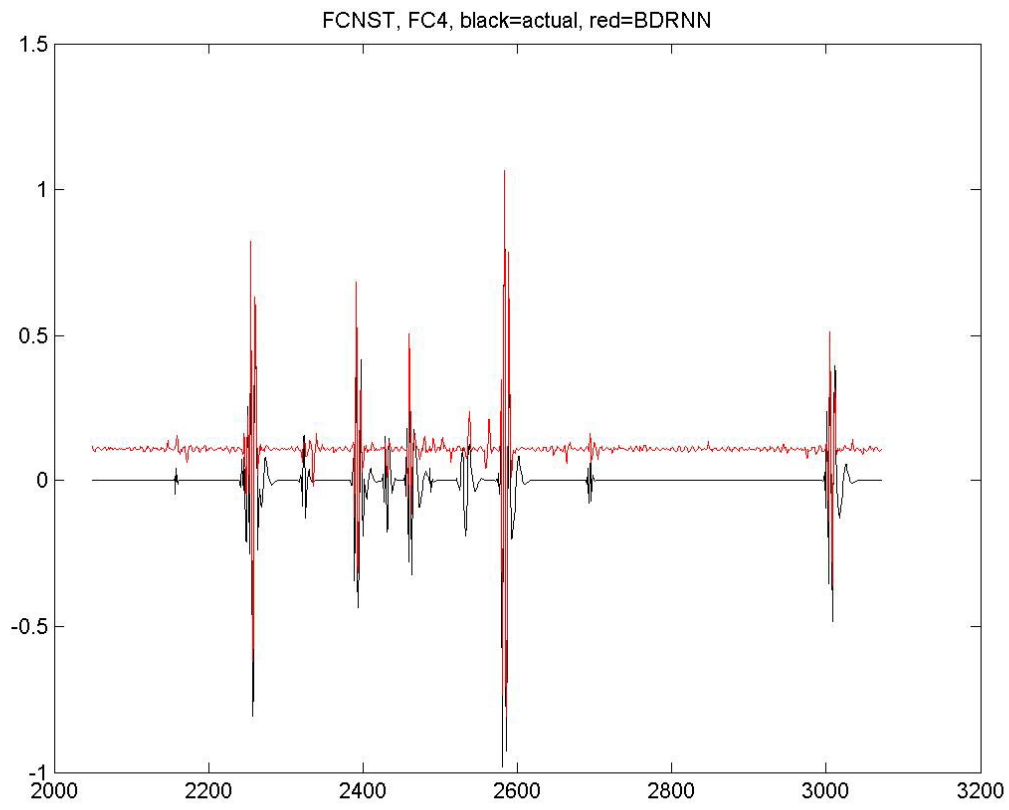
24^η Επανάληψη



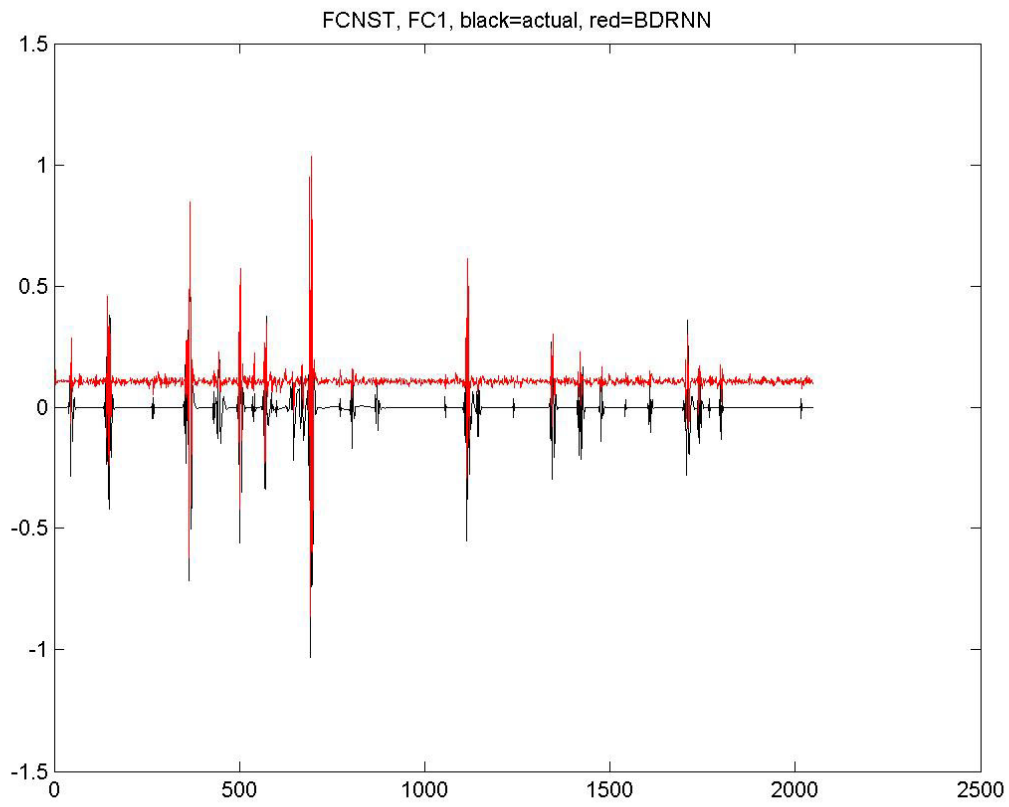
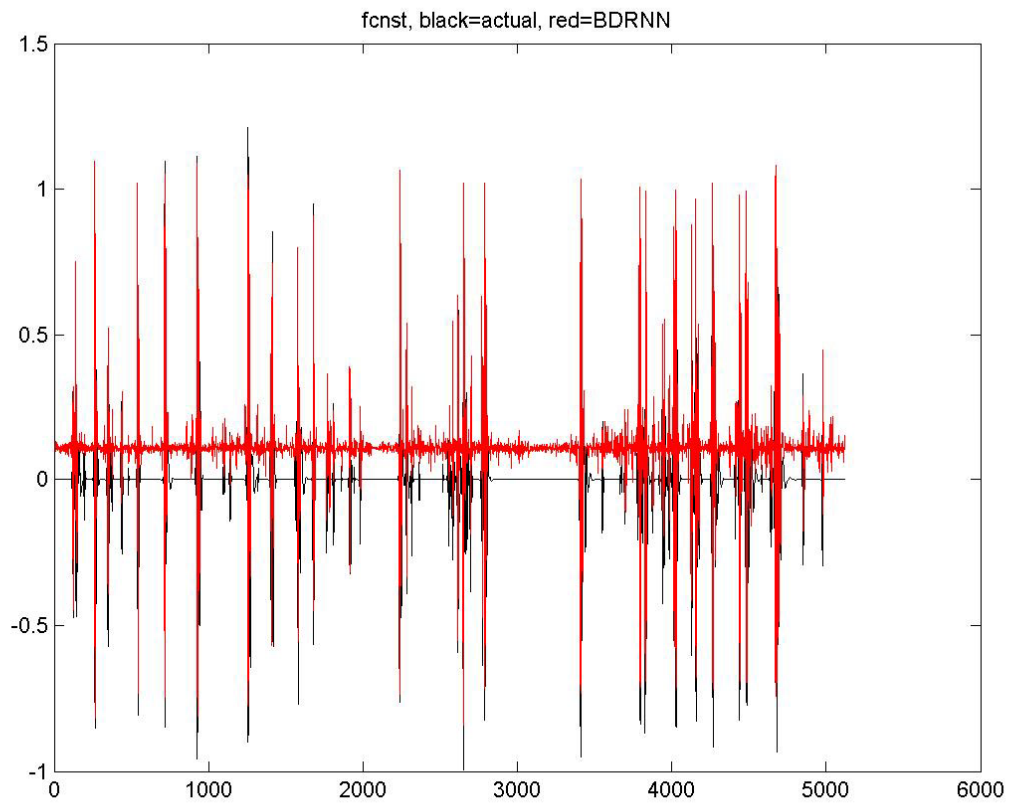


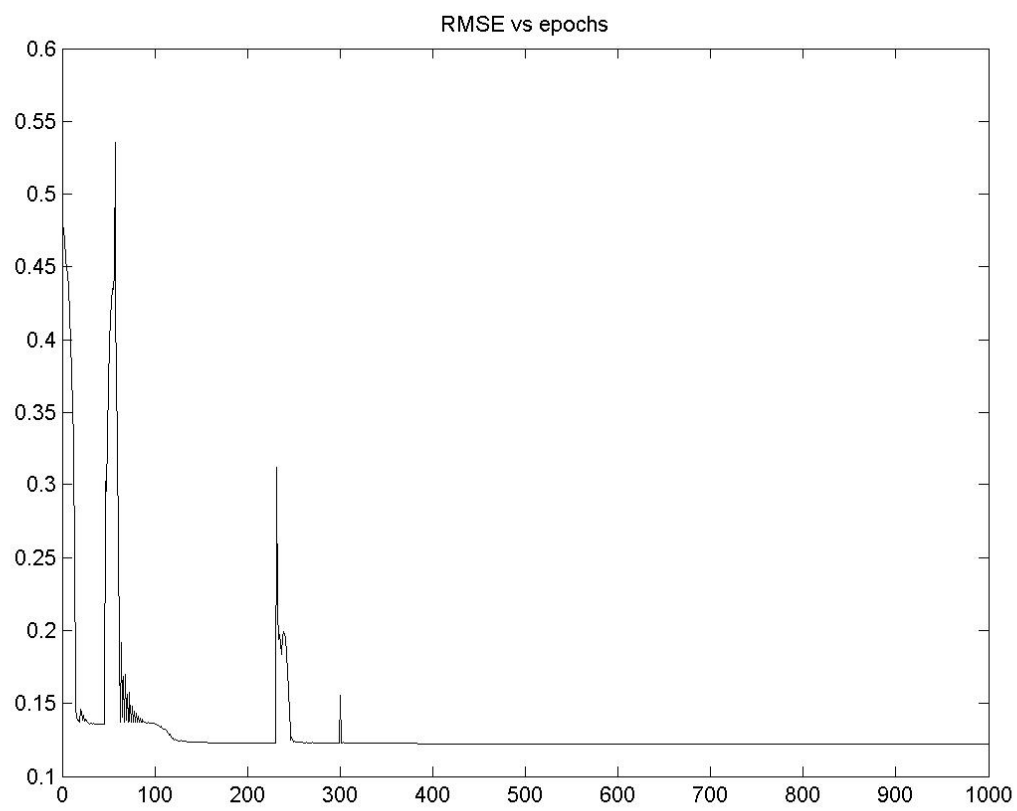
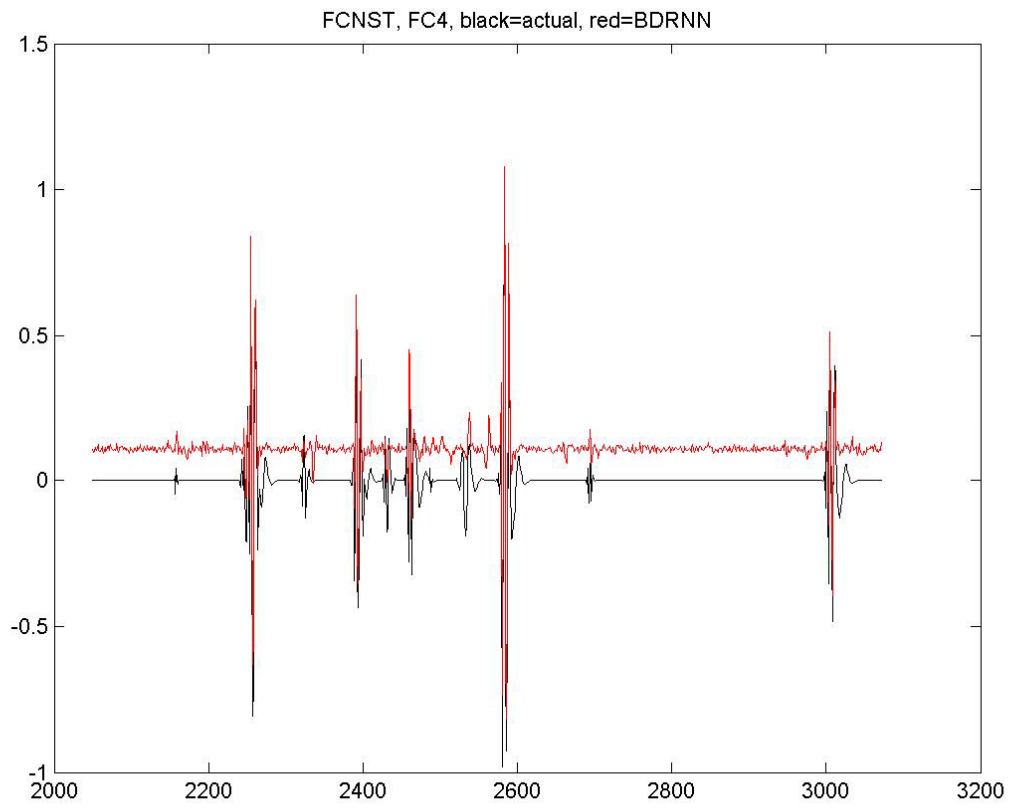
25^η Επανάληψη



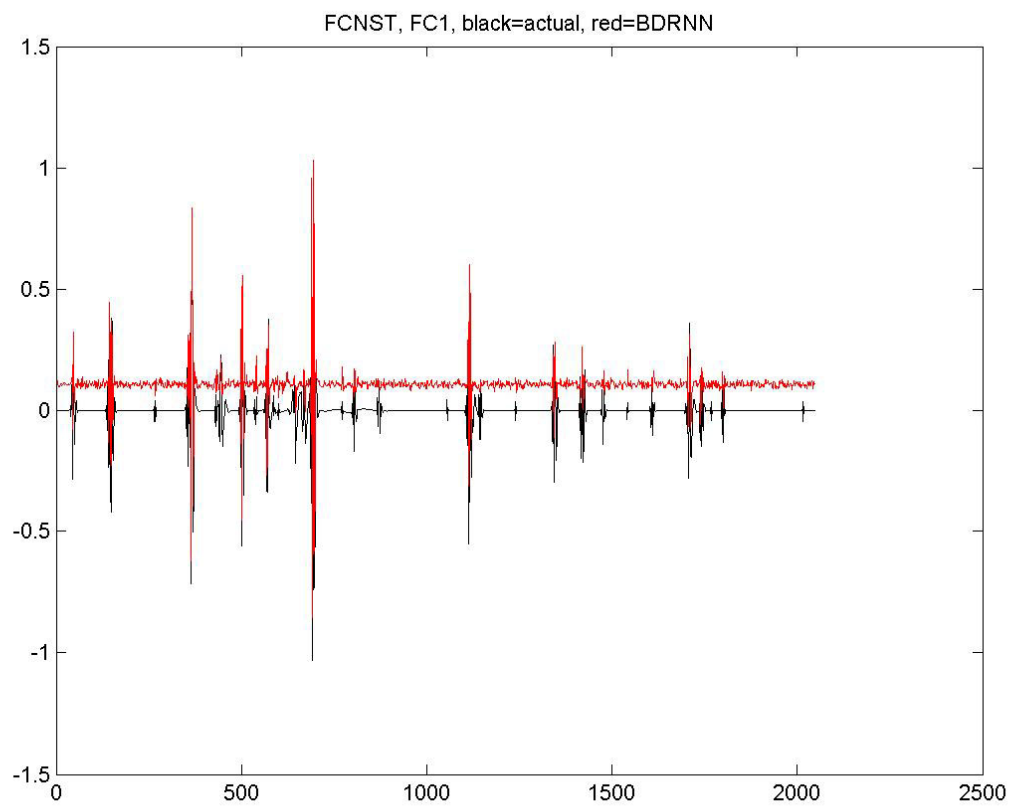
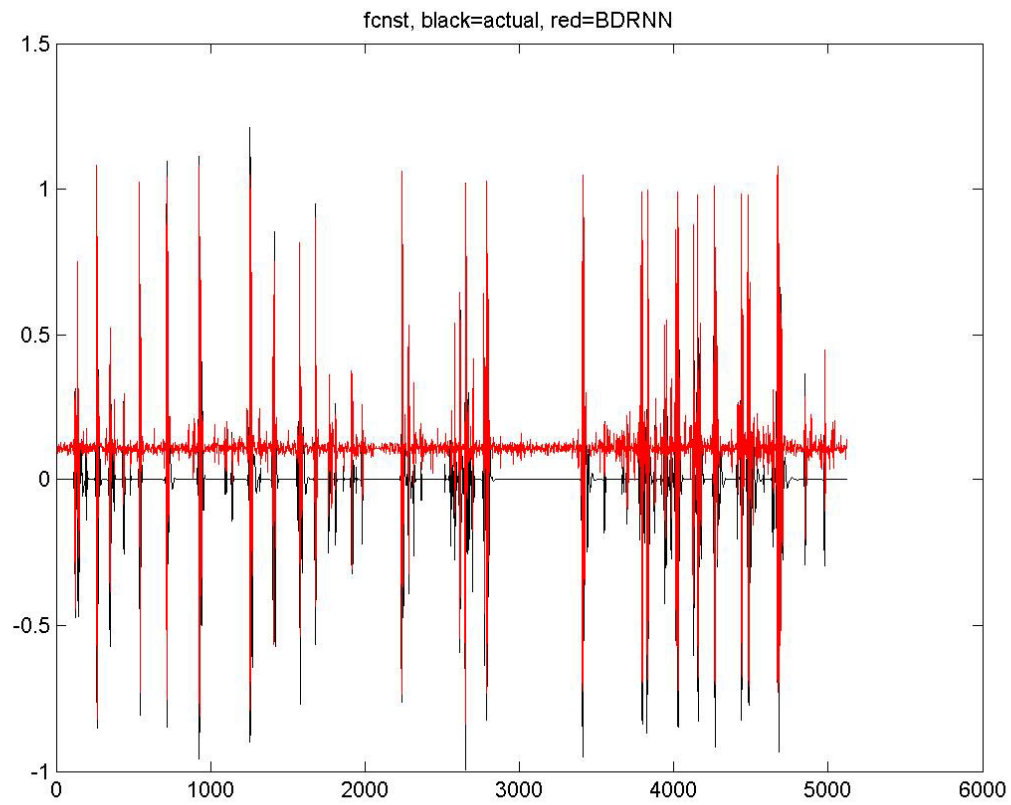


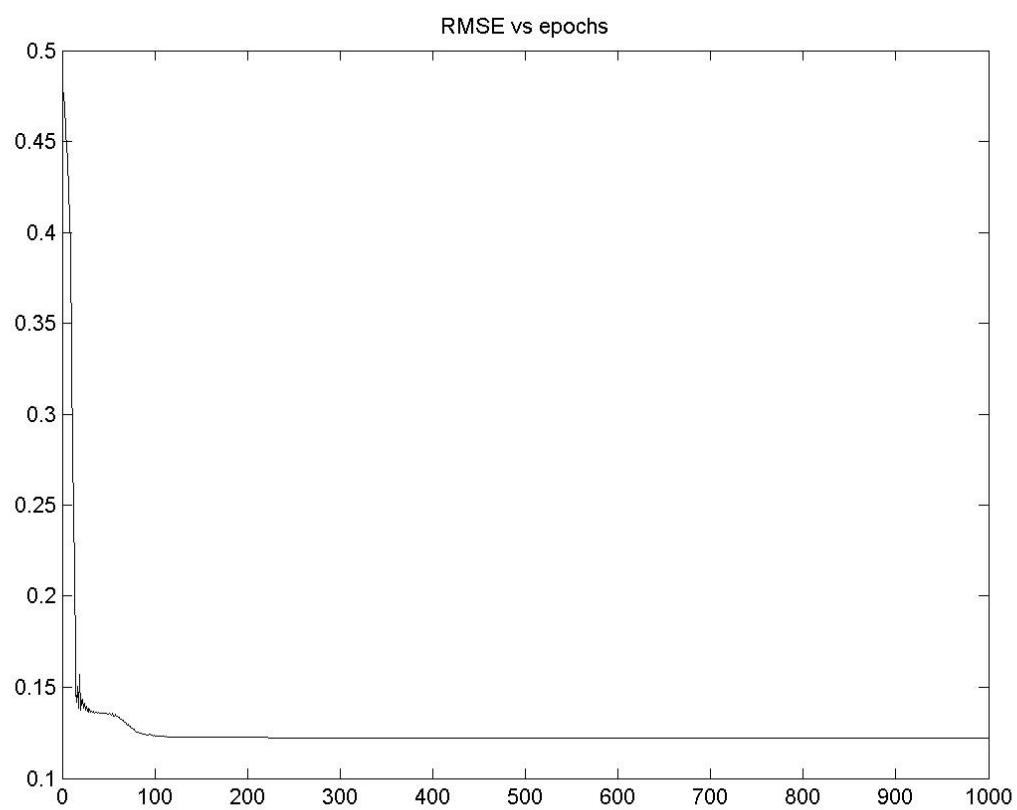
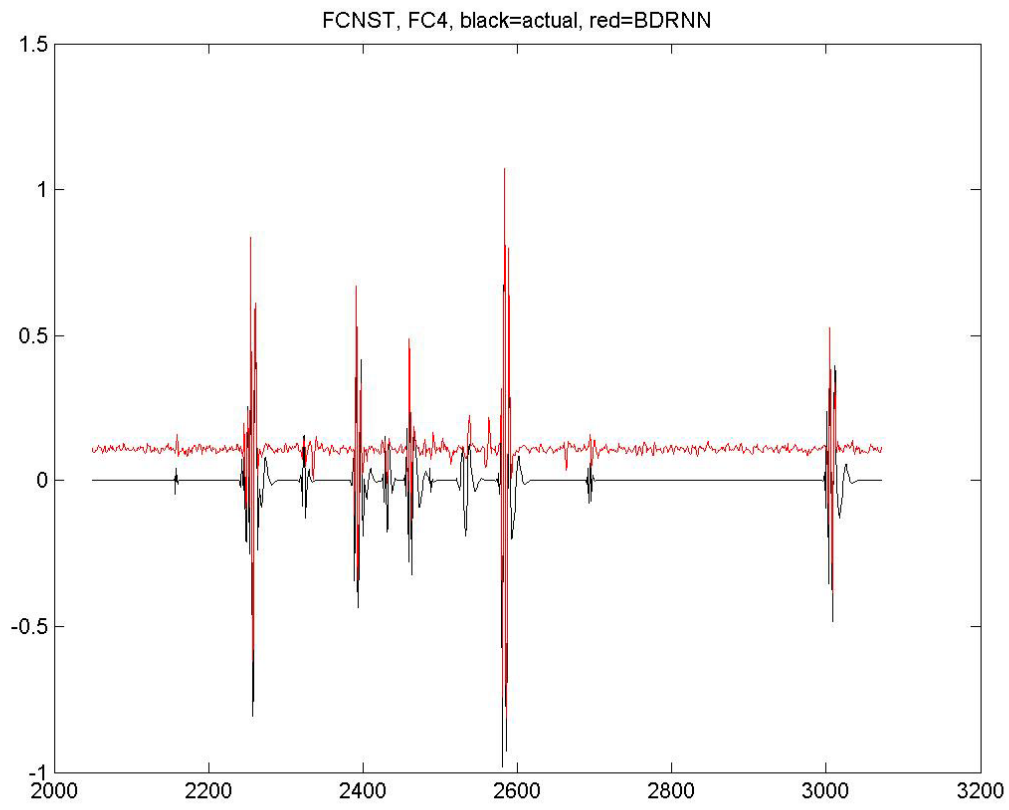
26^η Επανάληψη



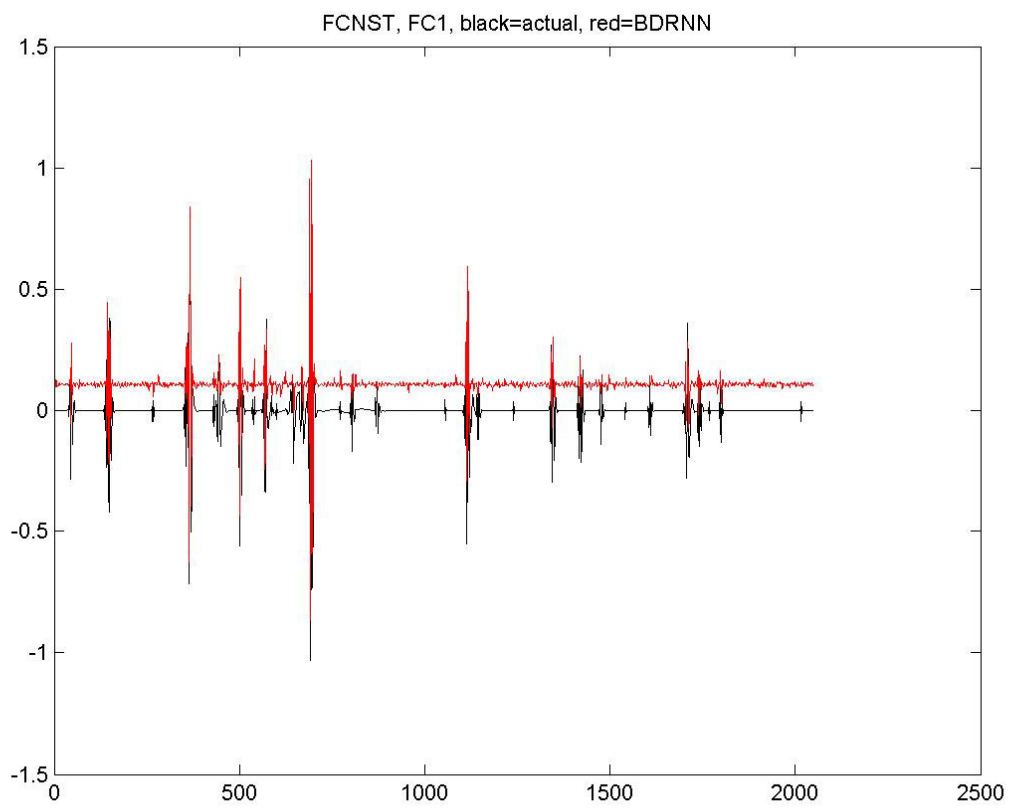
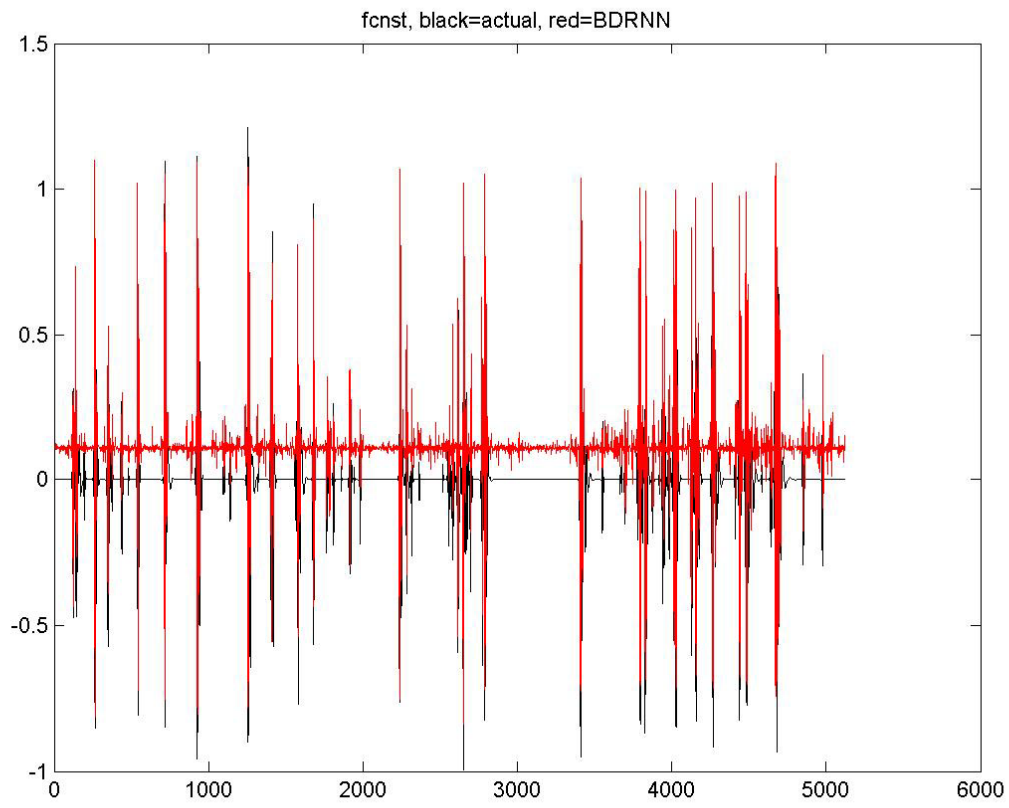


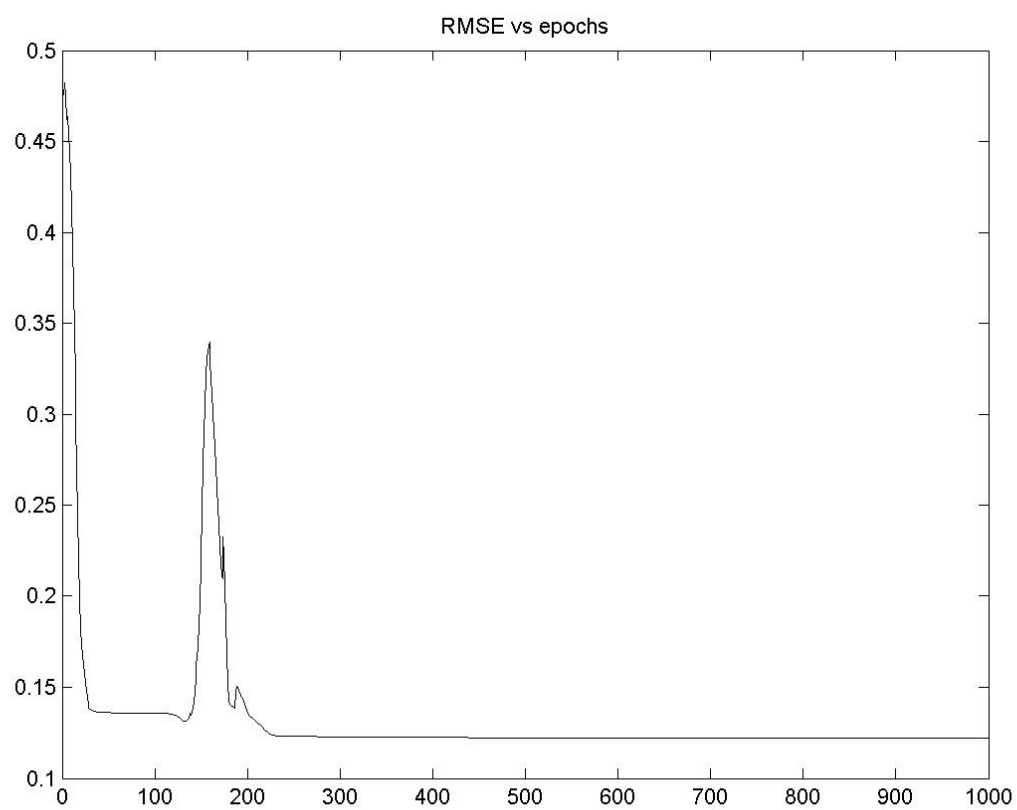
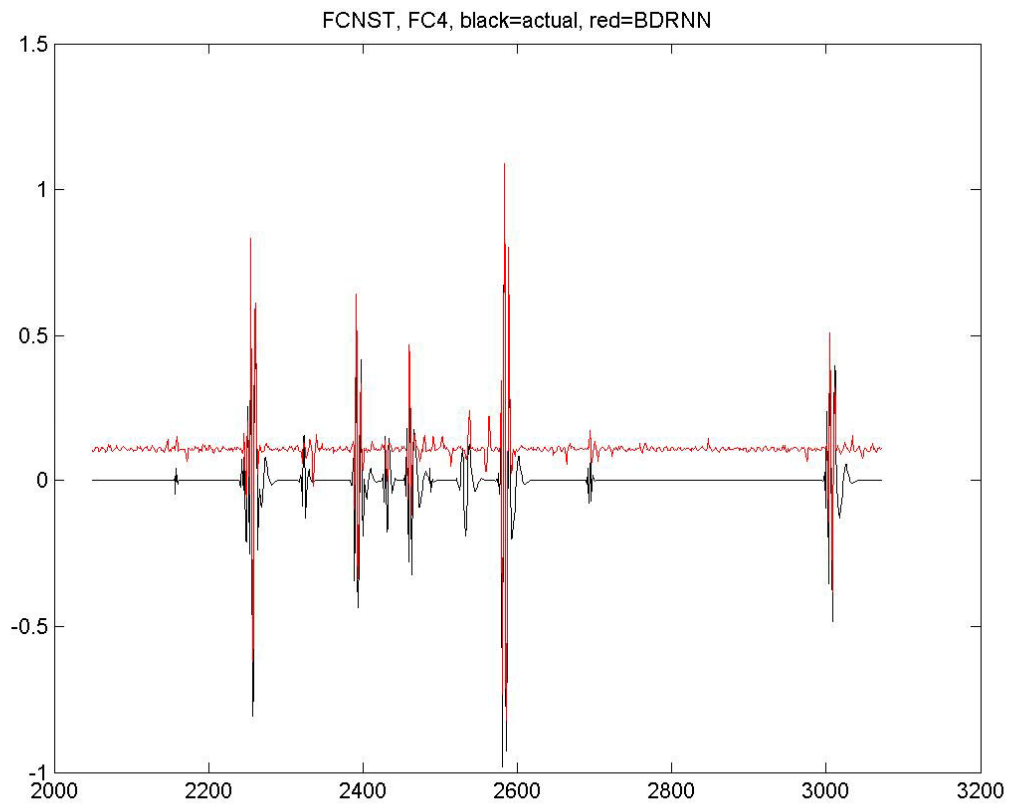
27^η Επανάληψη



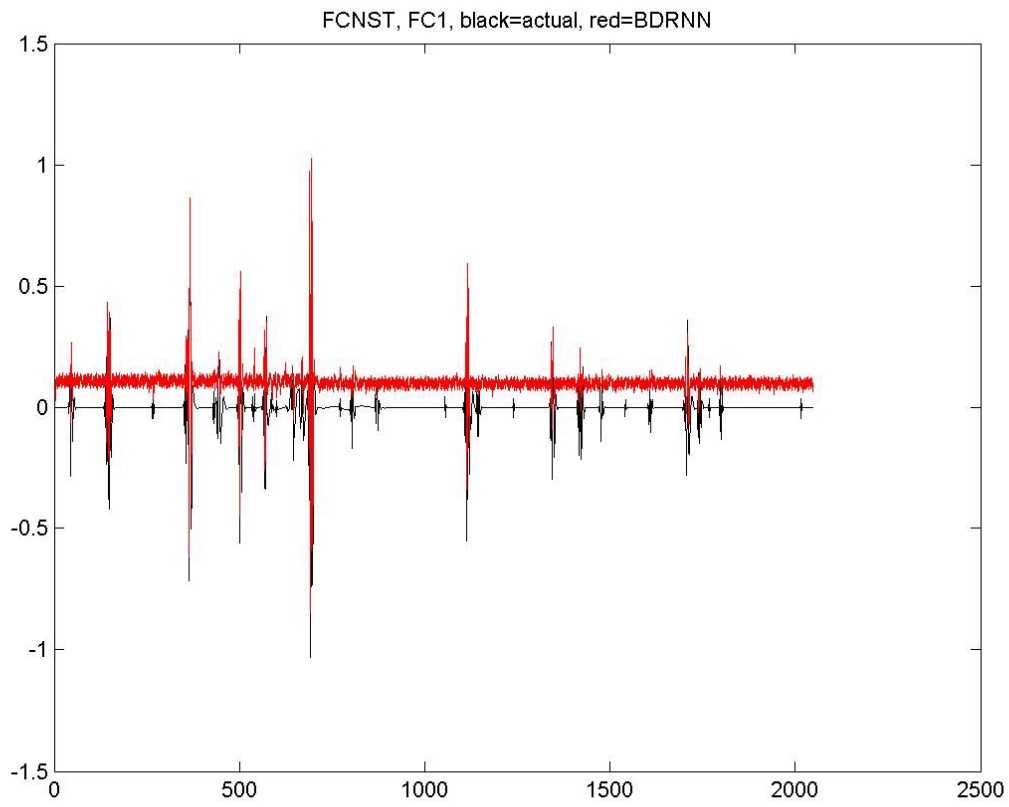
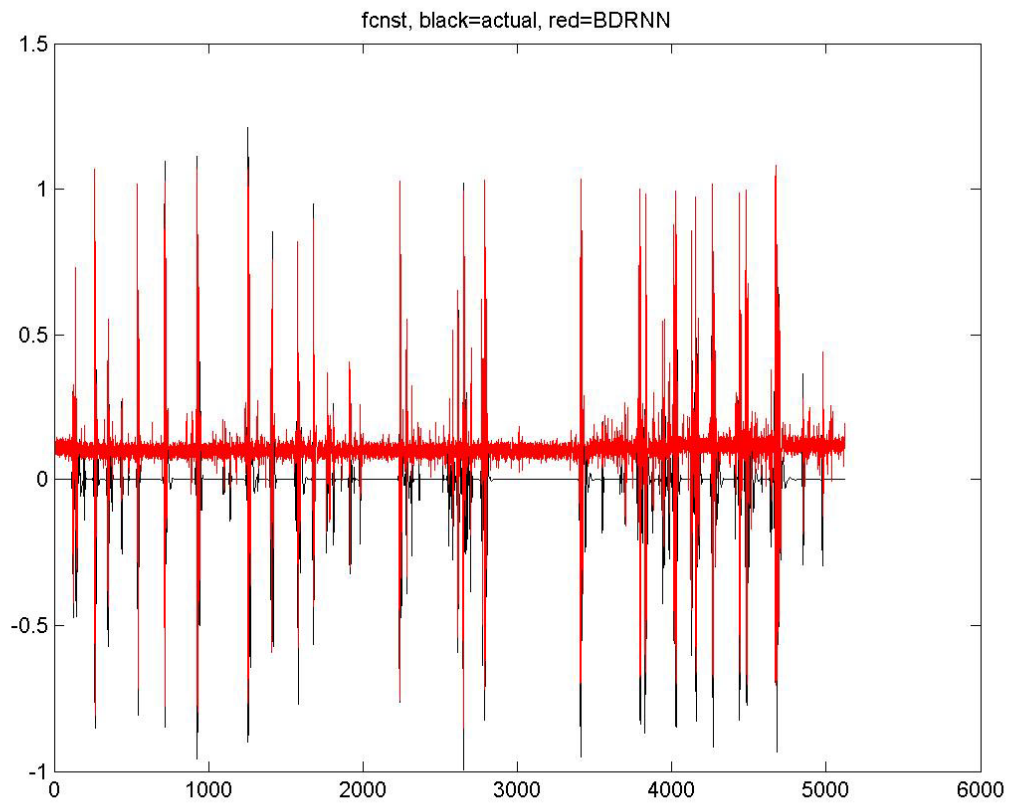


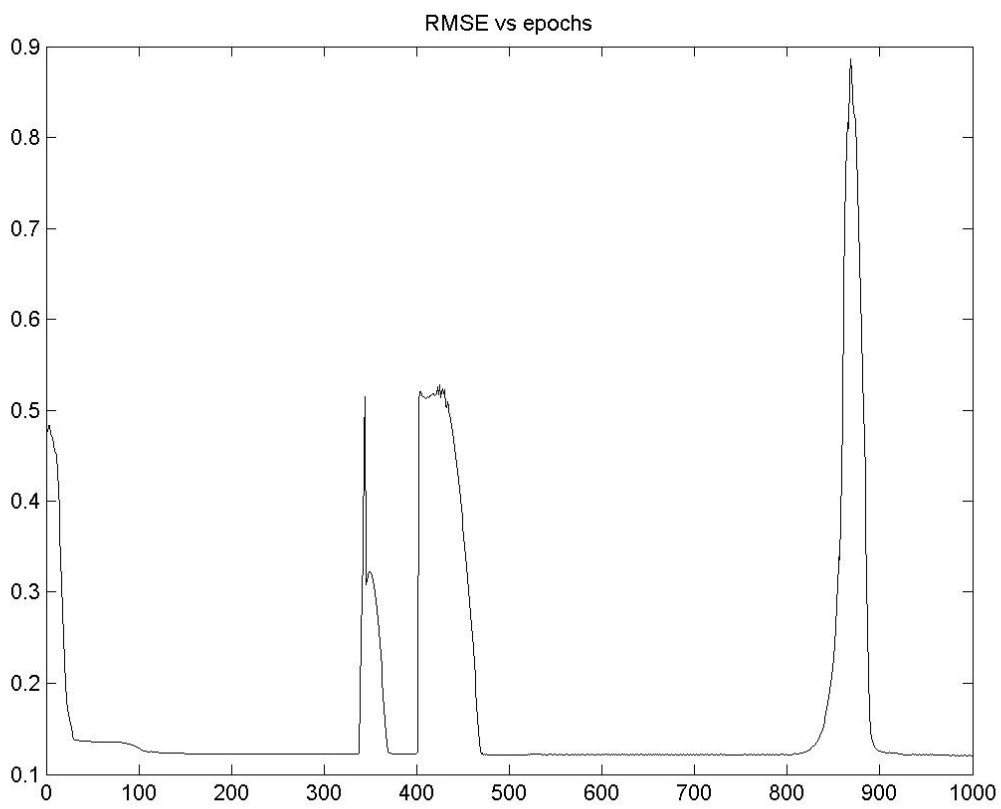
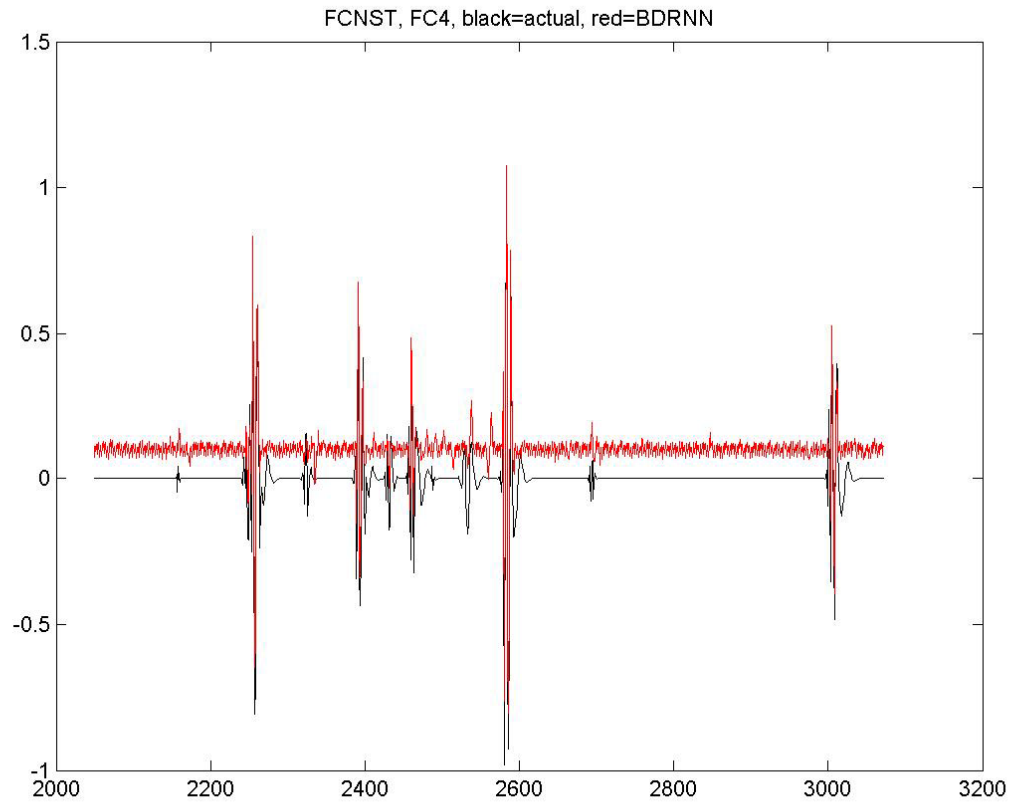
28^η Επανάληψη



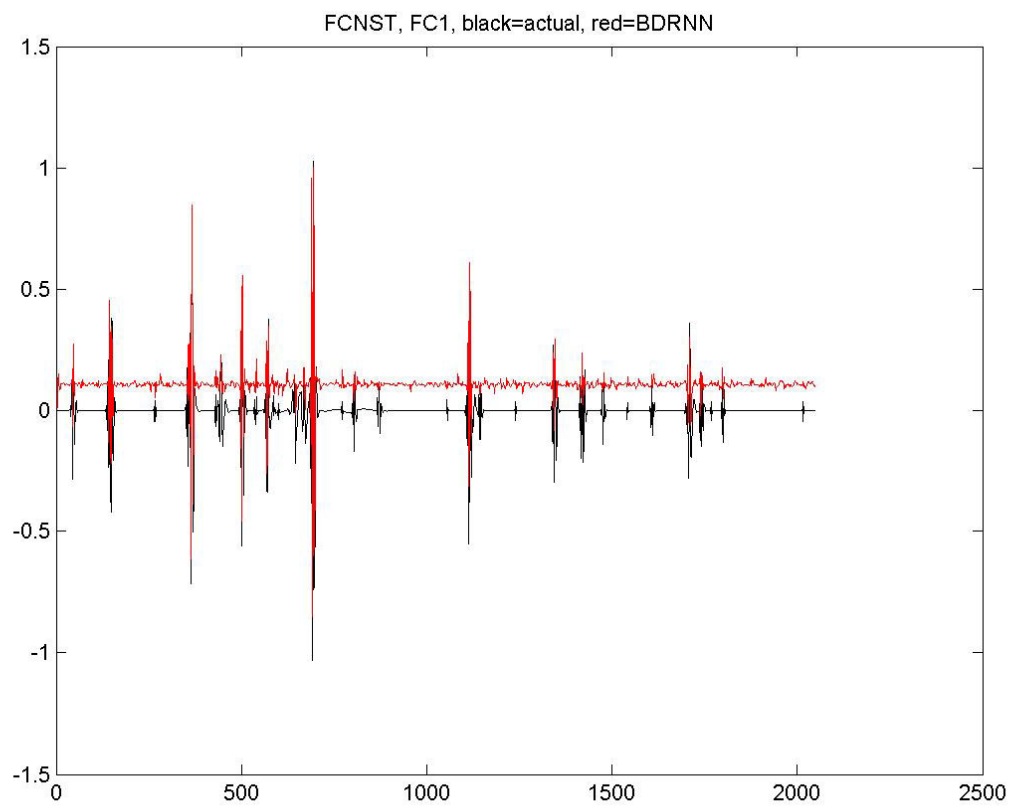
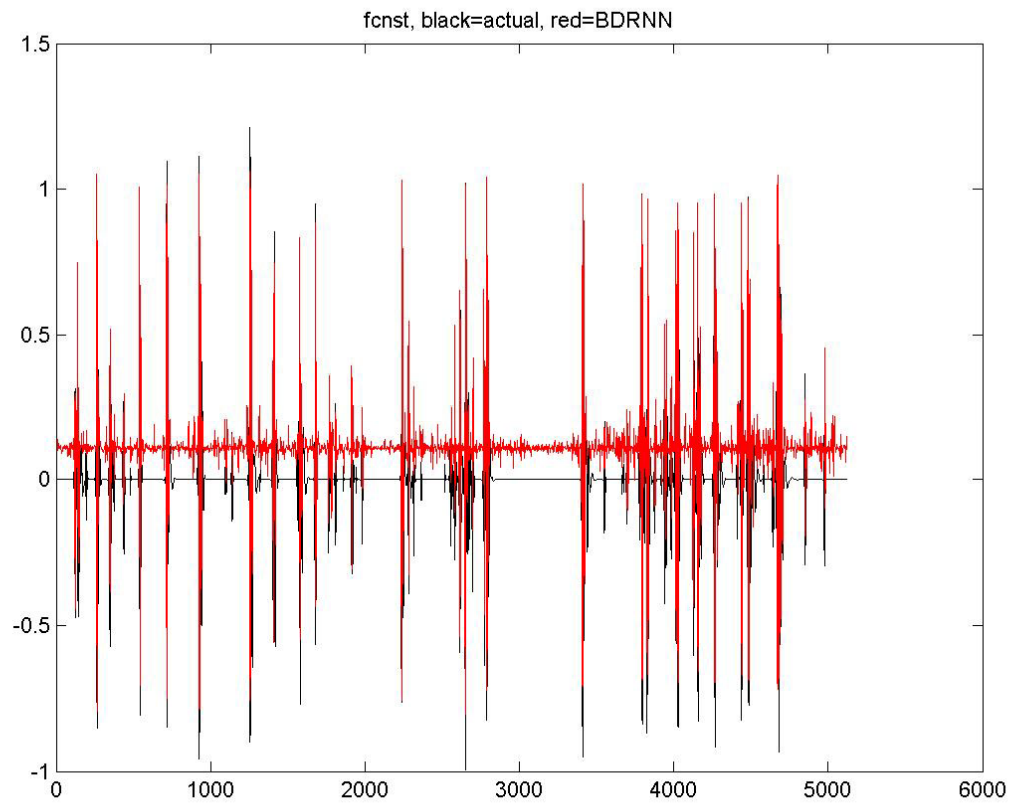


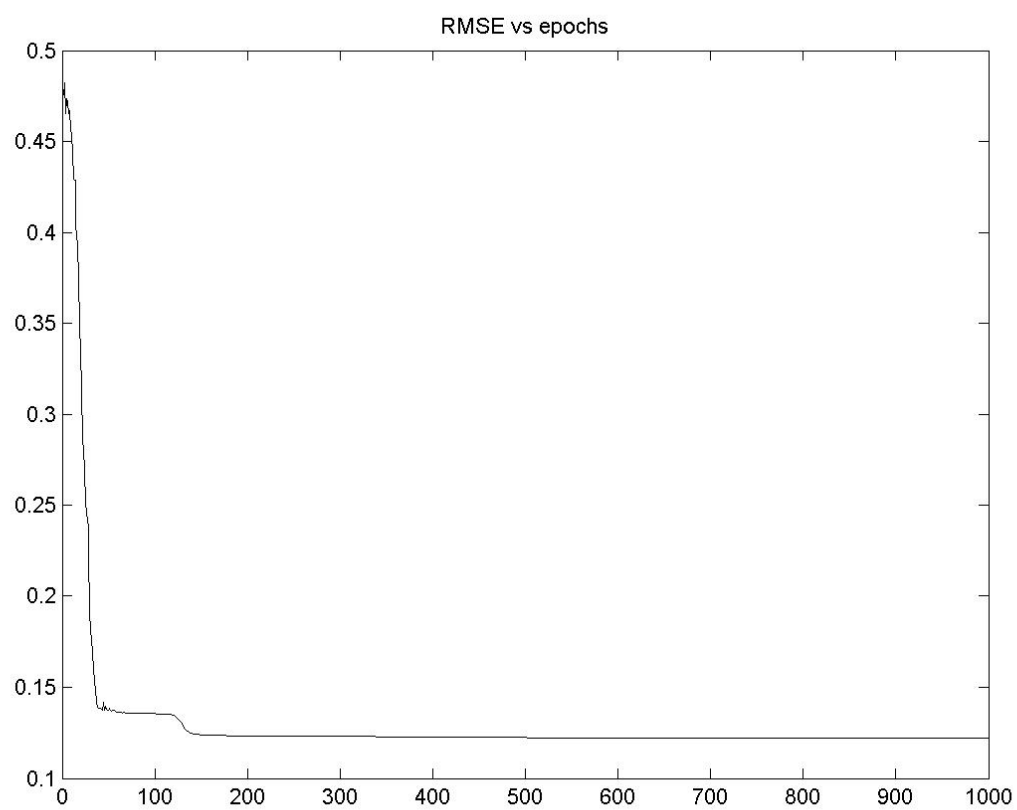
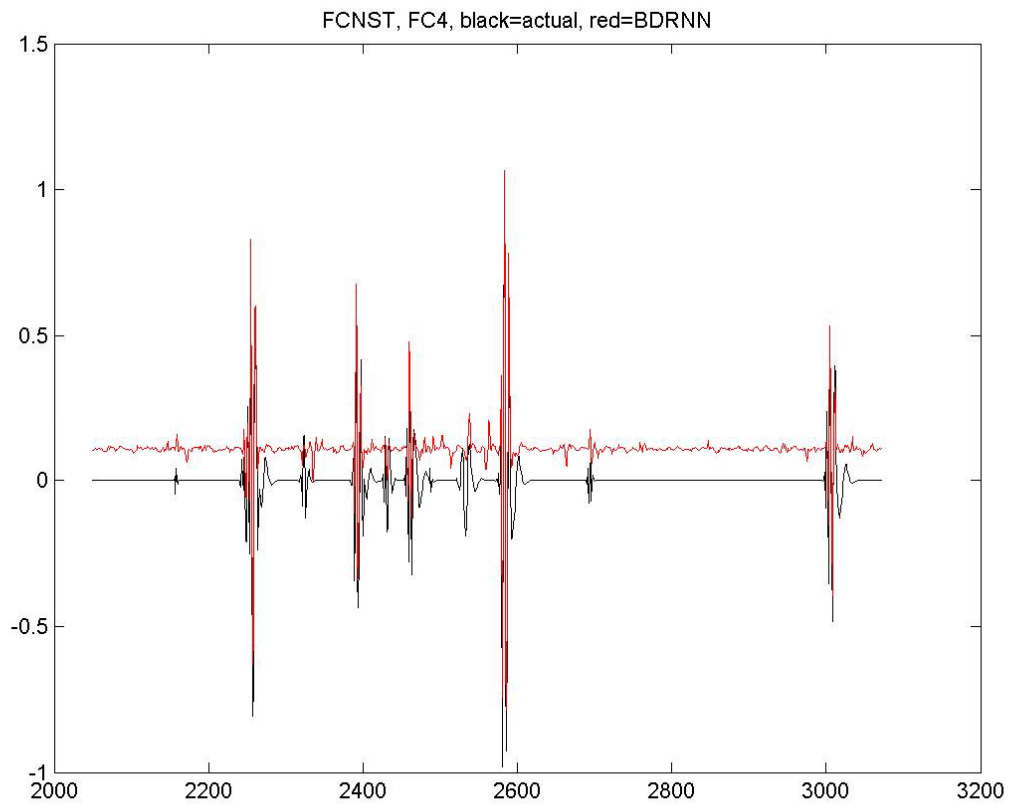
29^η Επανάληψη



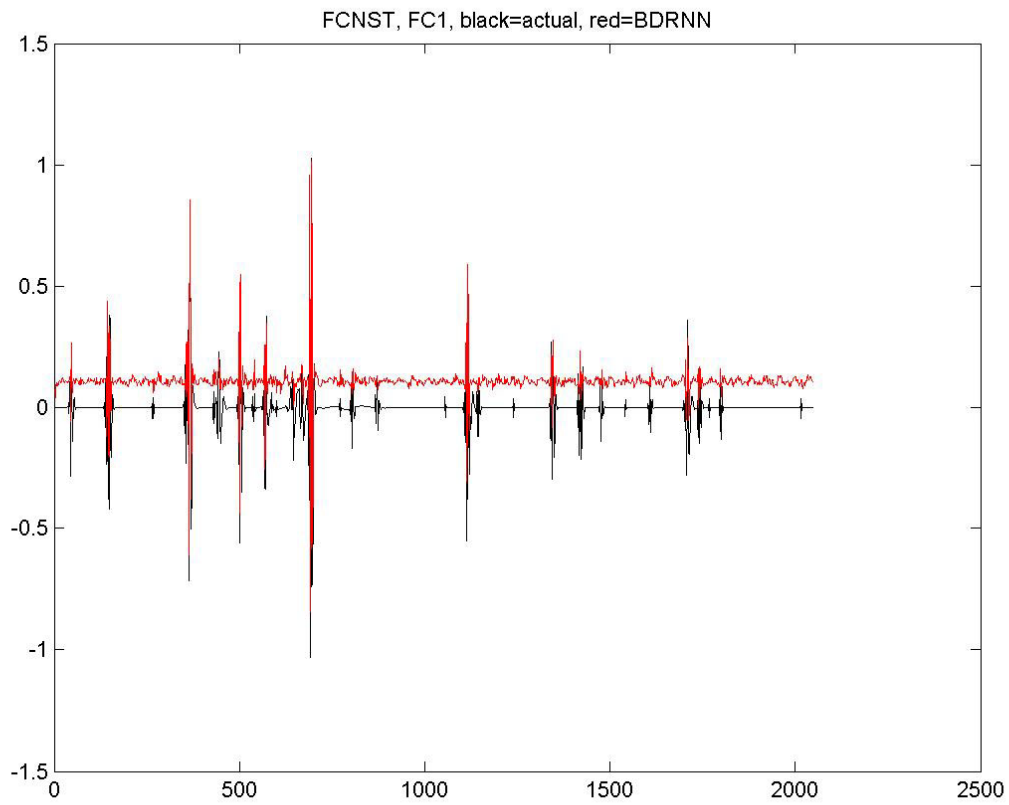
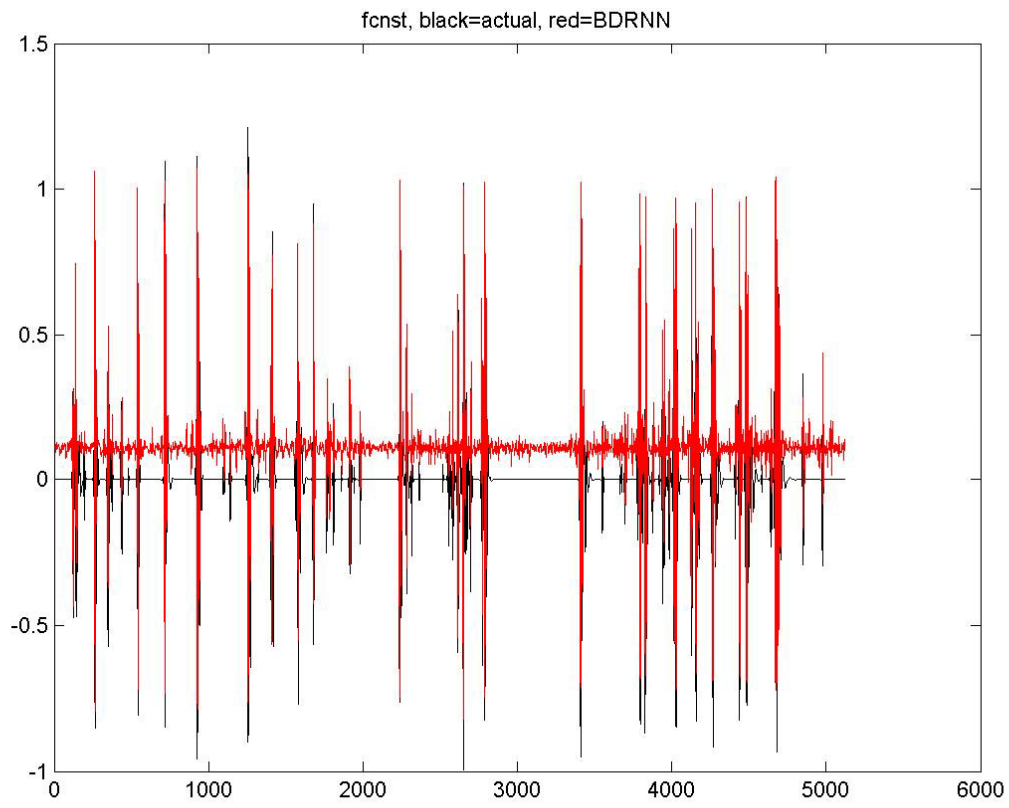


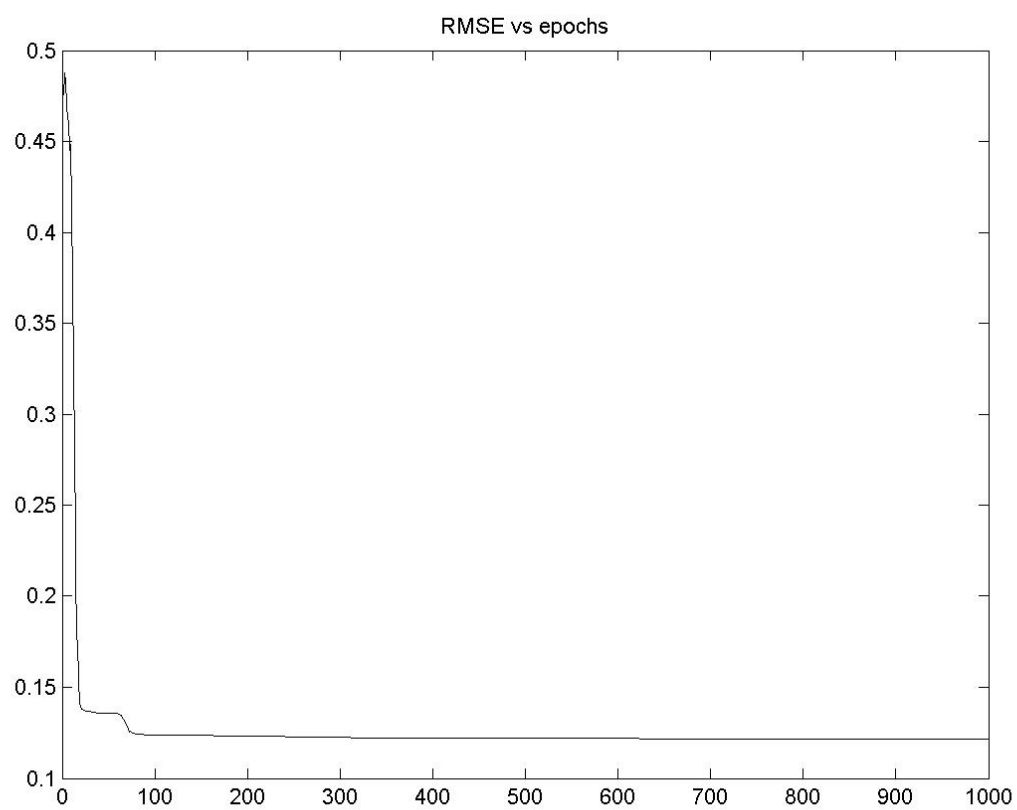
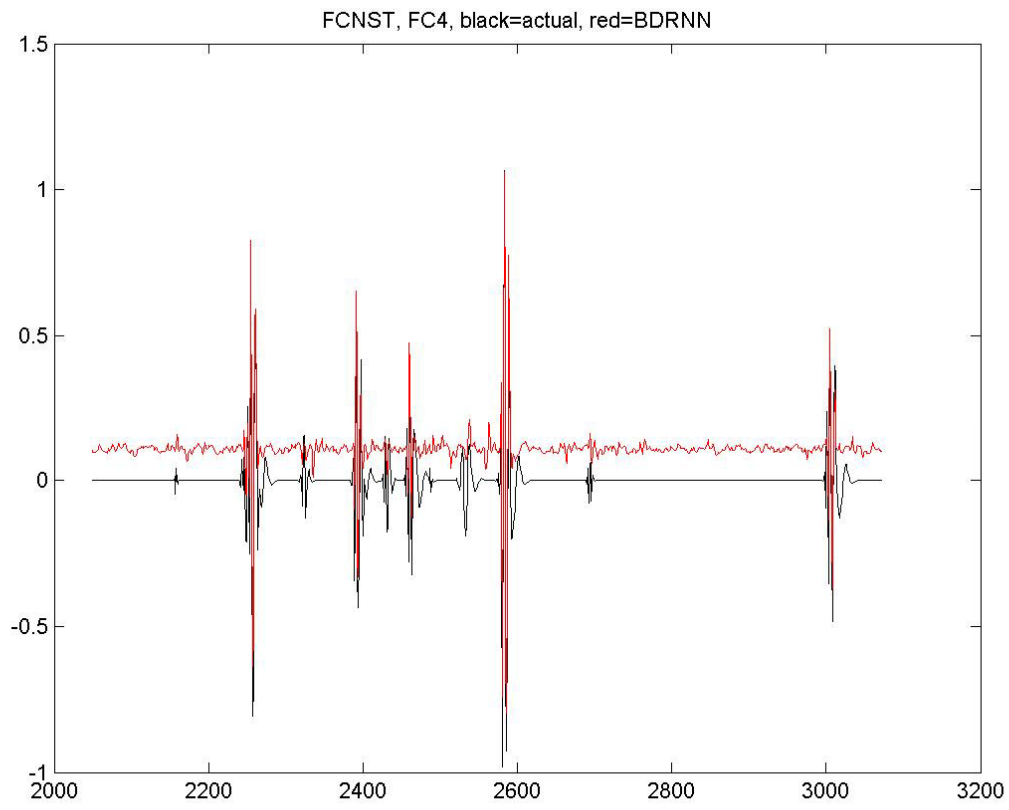
30^η Επανάληψη



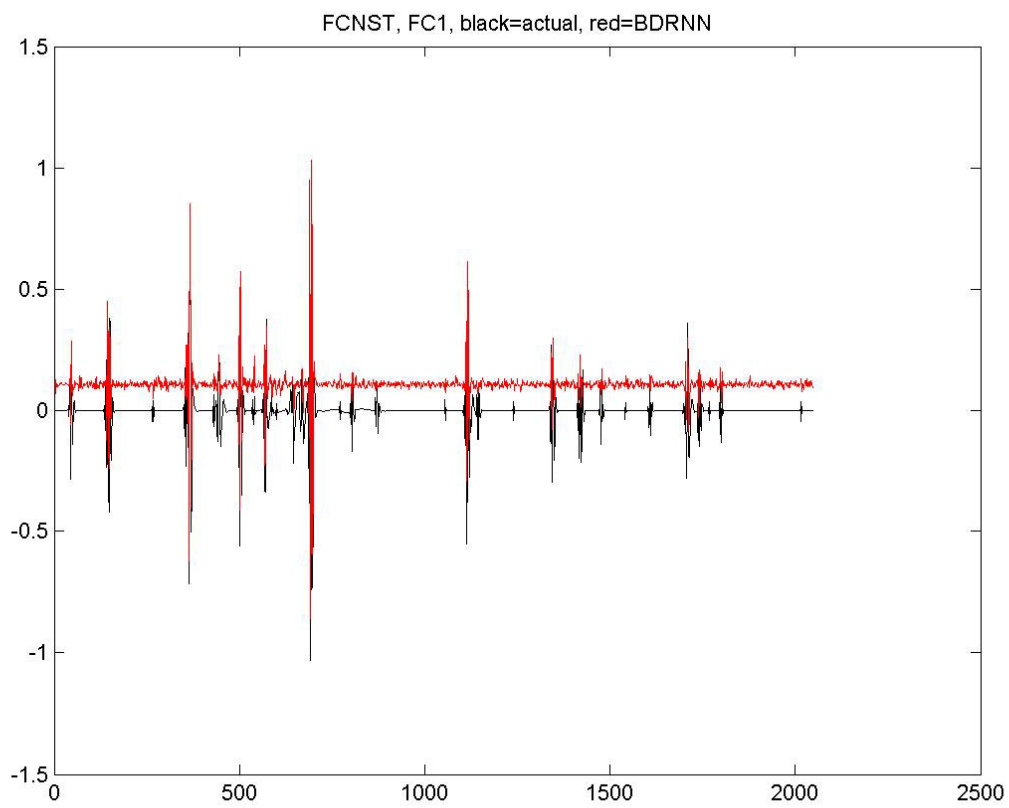
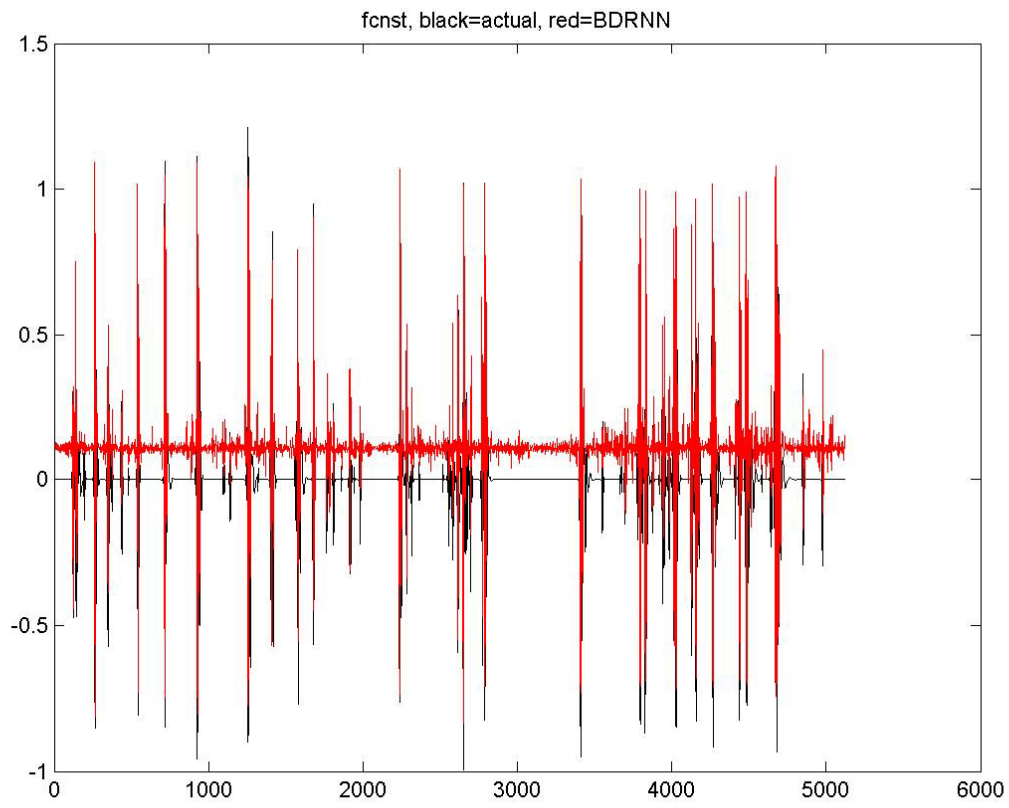


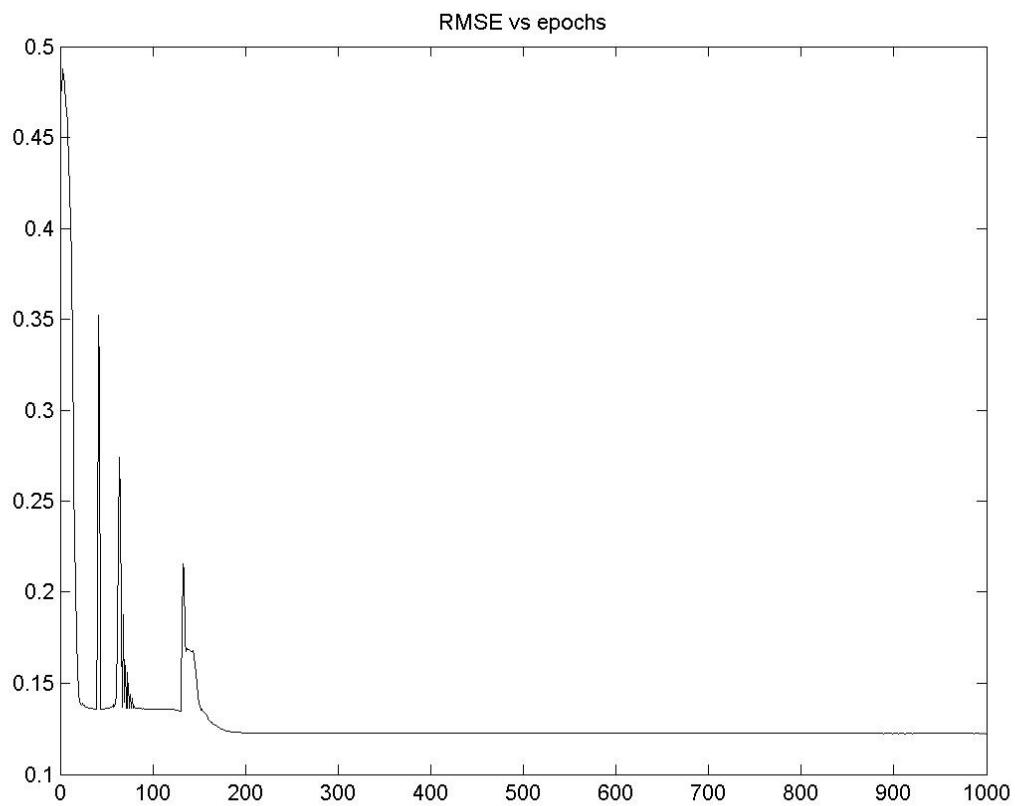
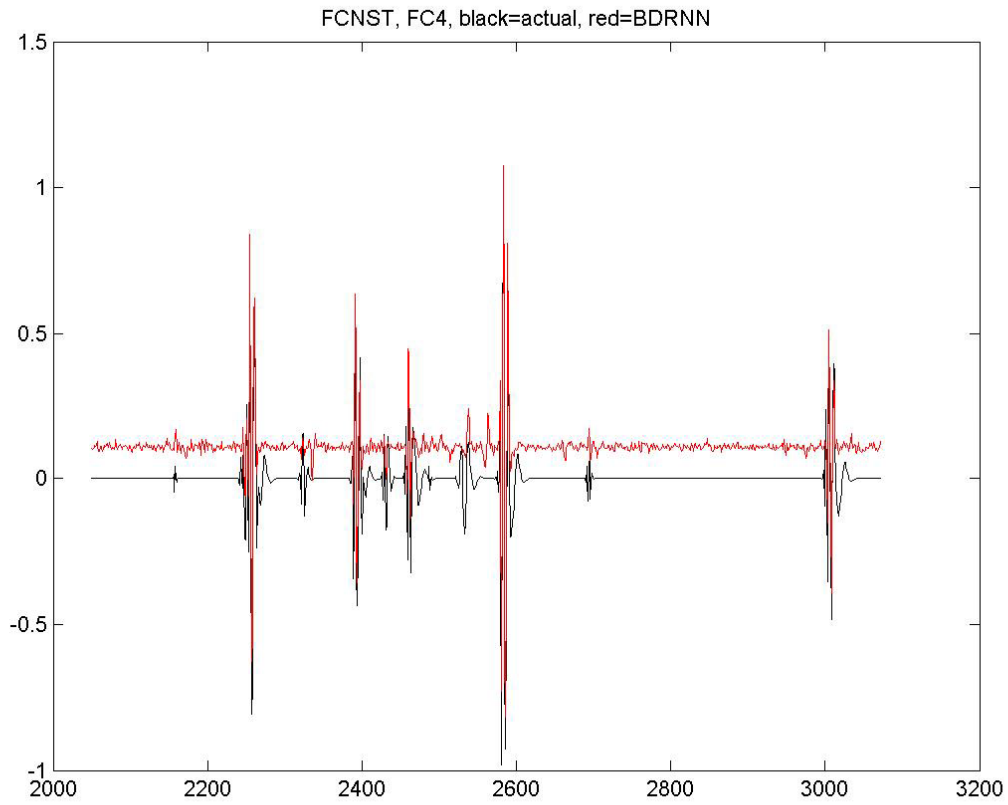
31^η Επανάληψη



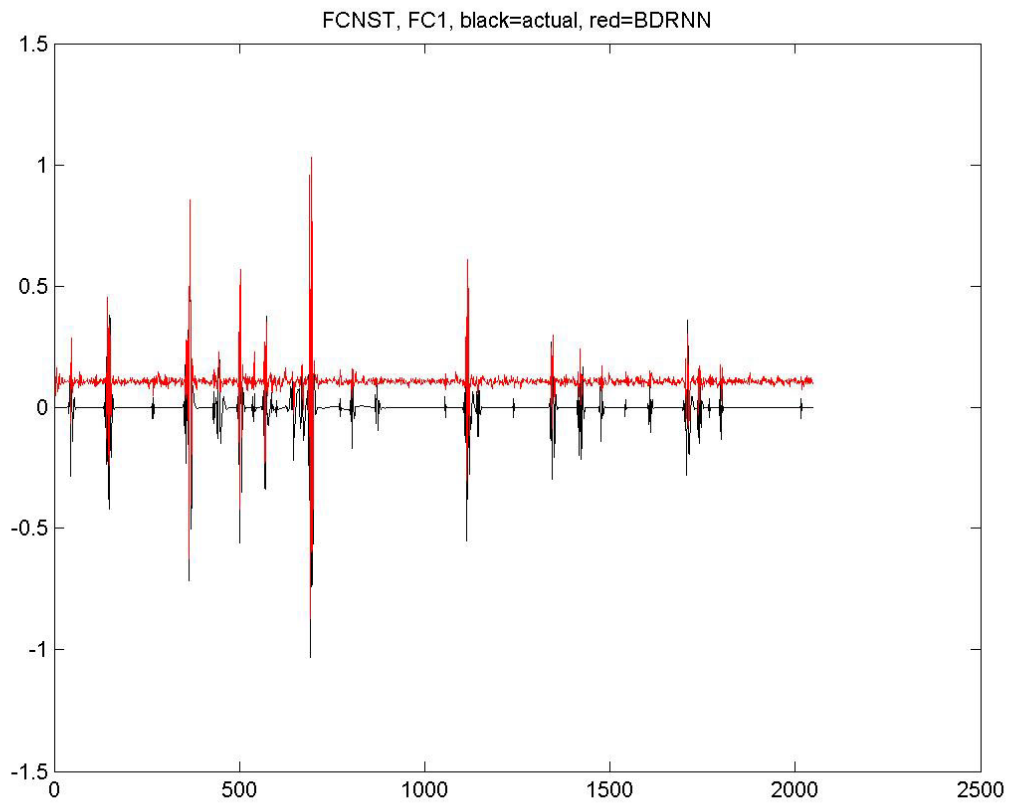
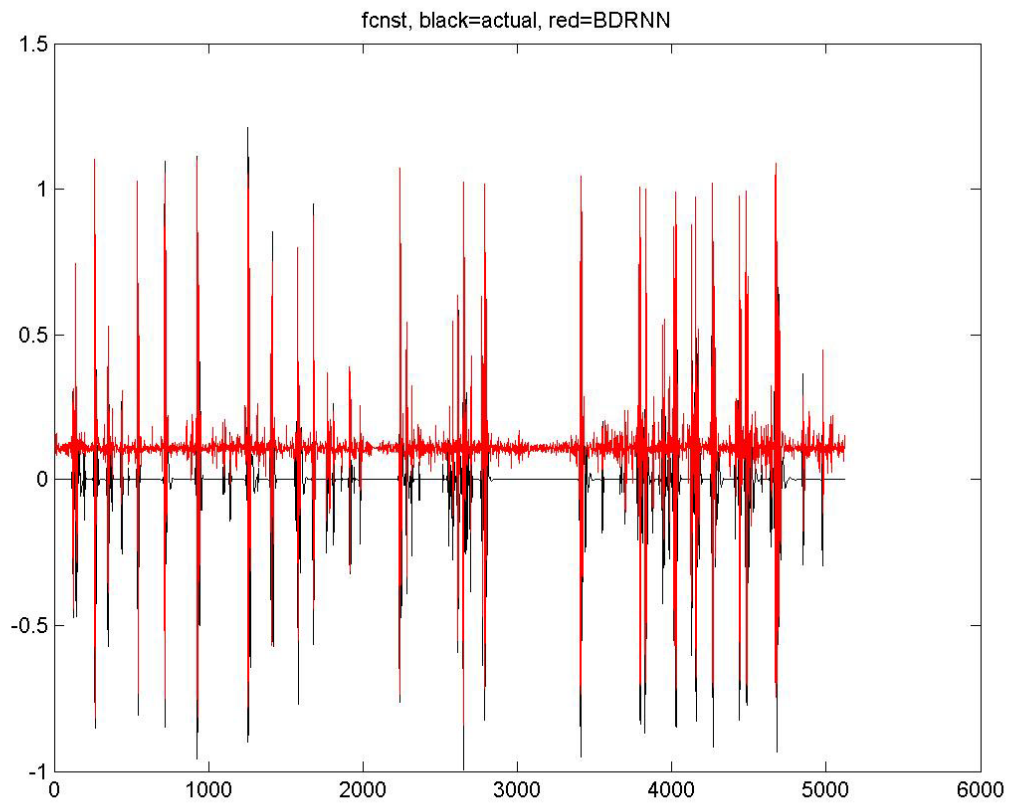


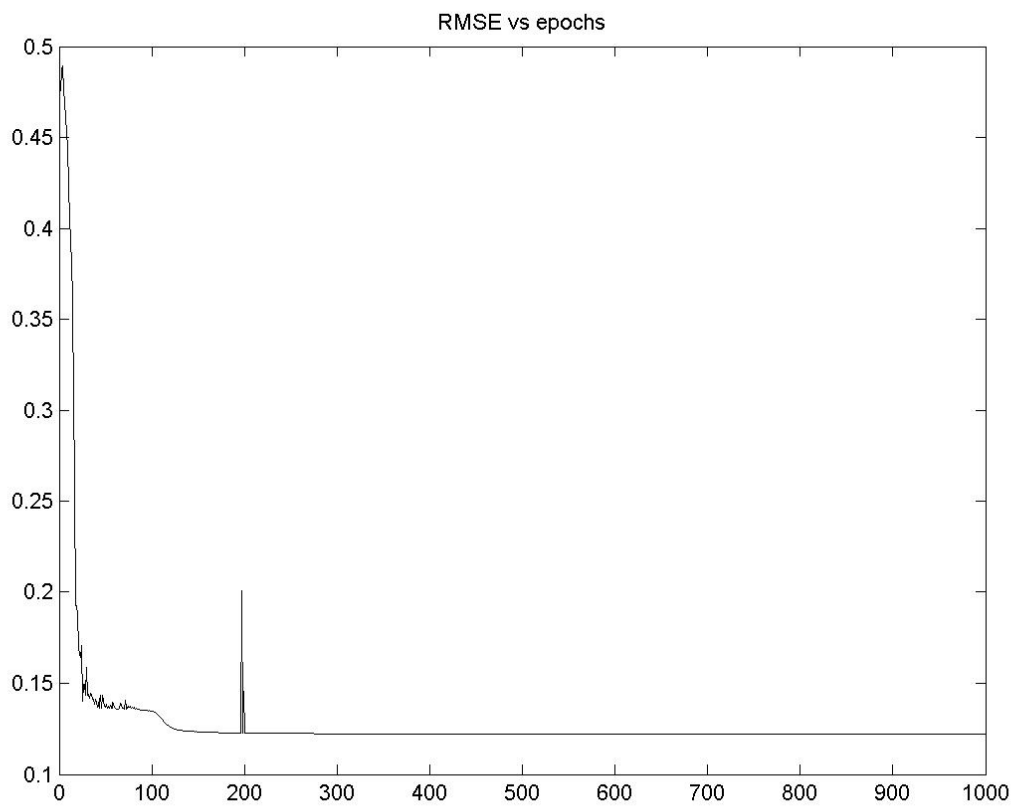
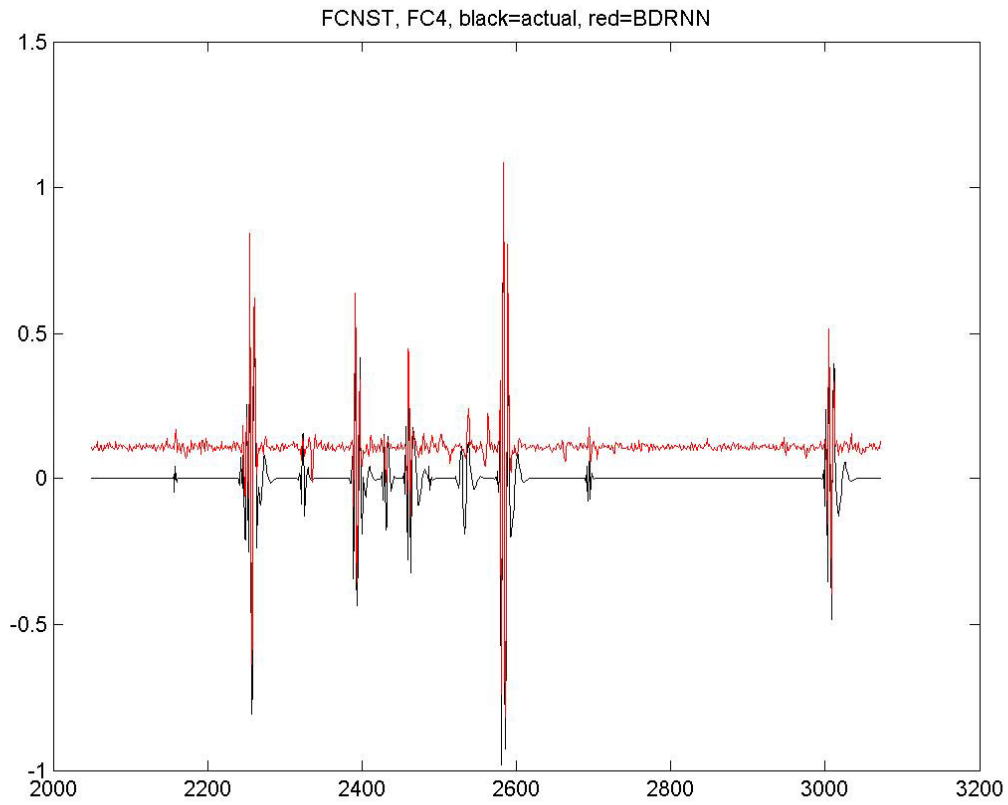
32^η Επανάληψη



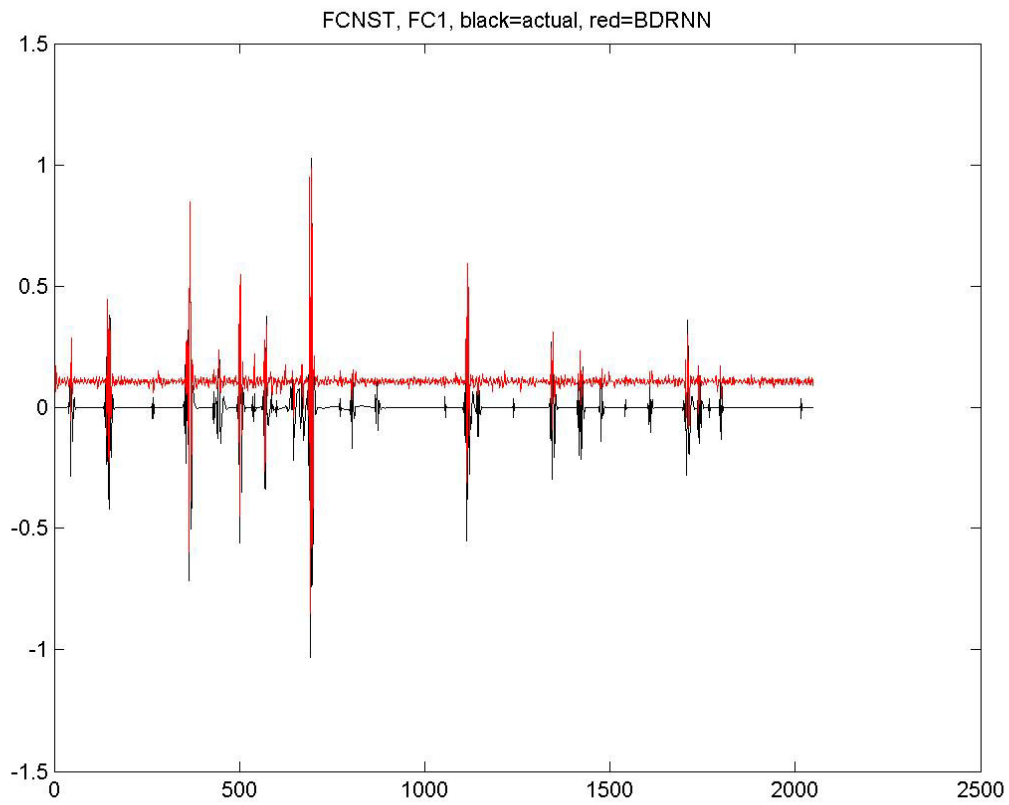
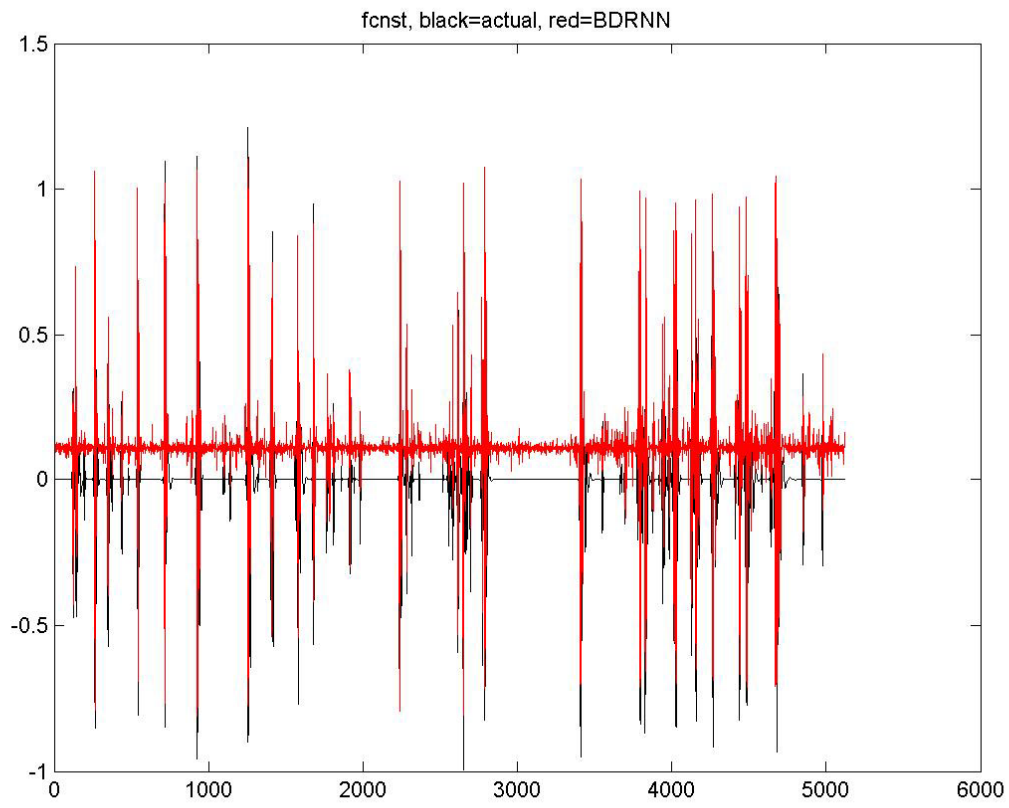


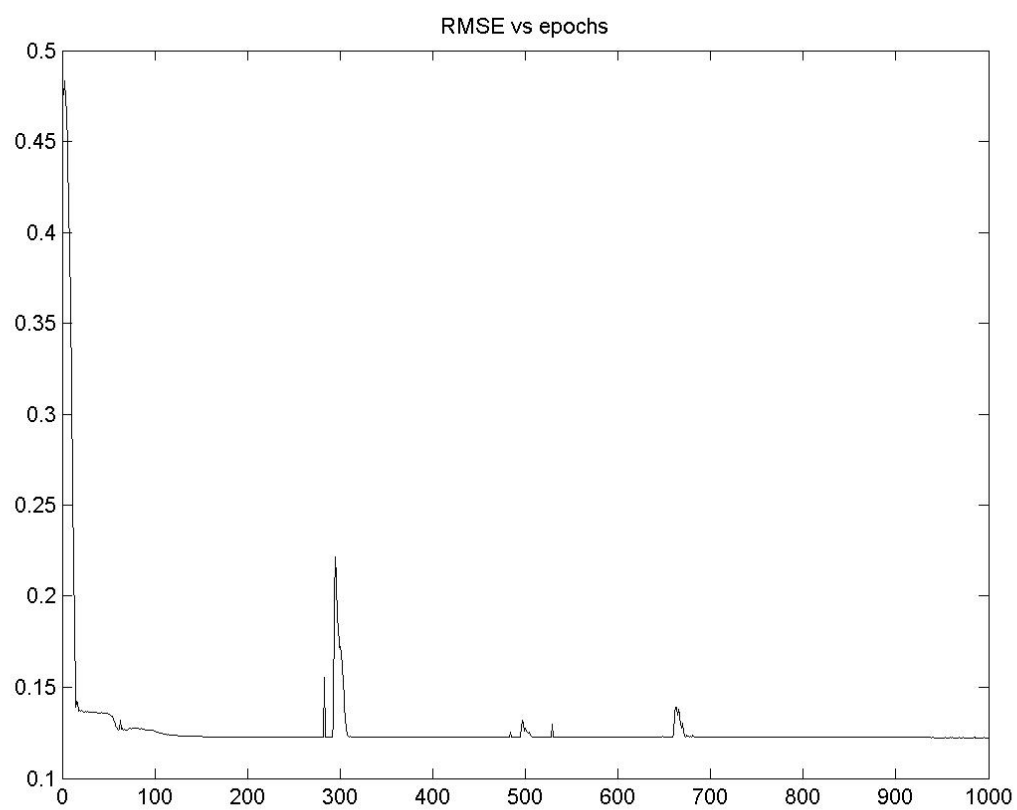
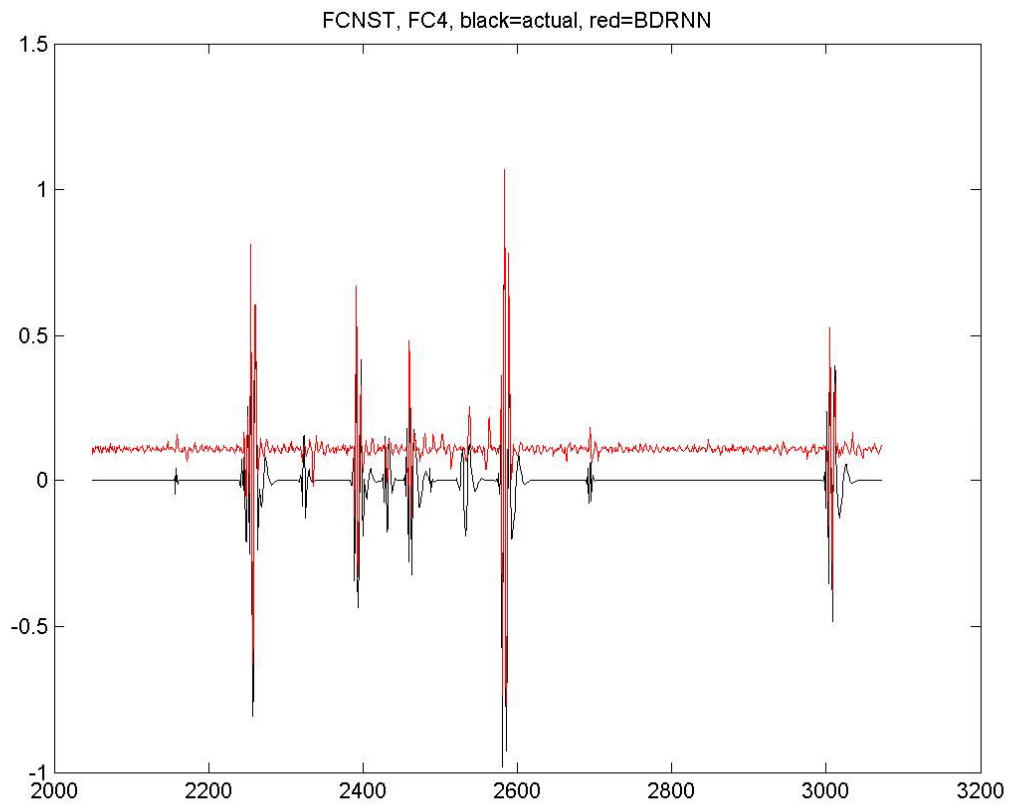
33^η Επανάληψη



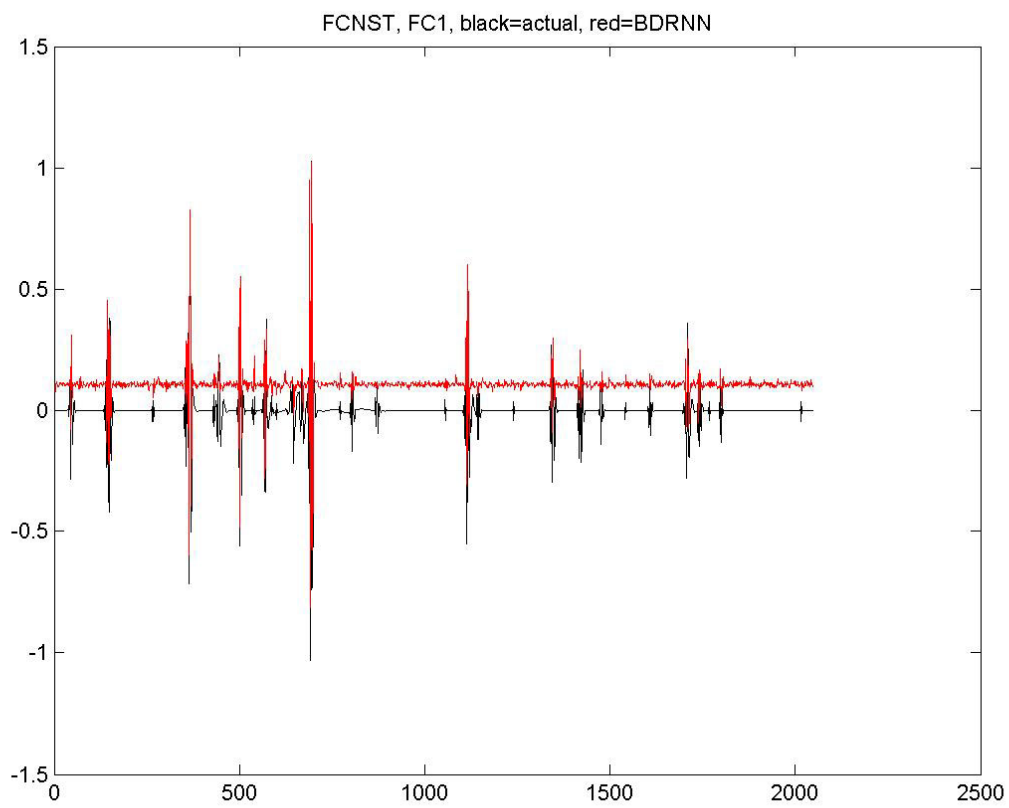
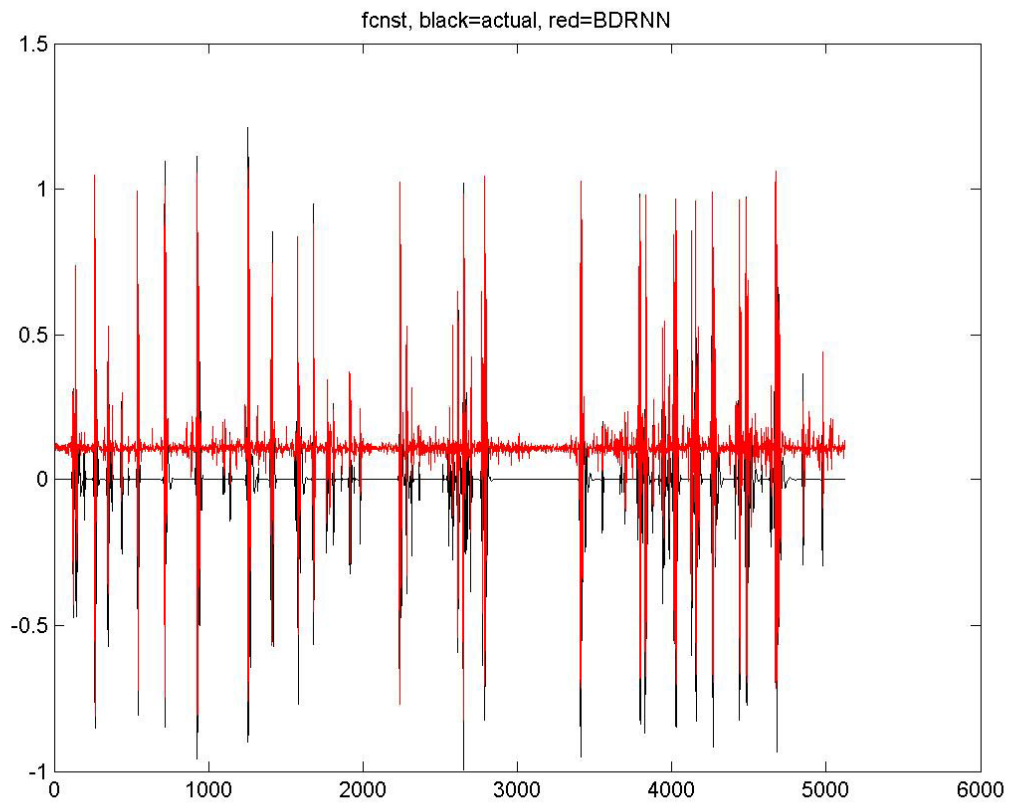


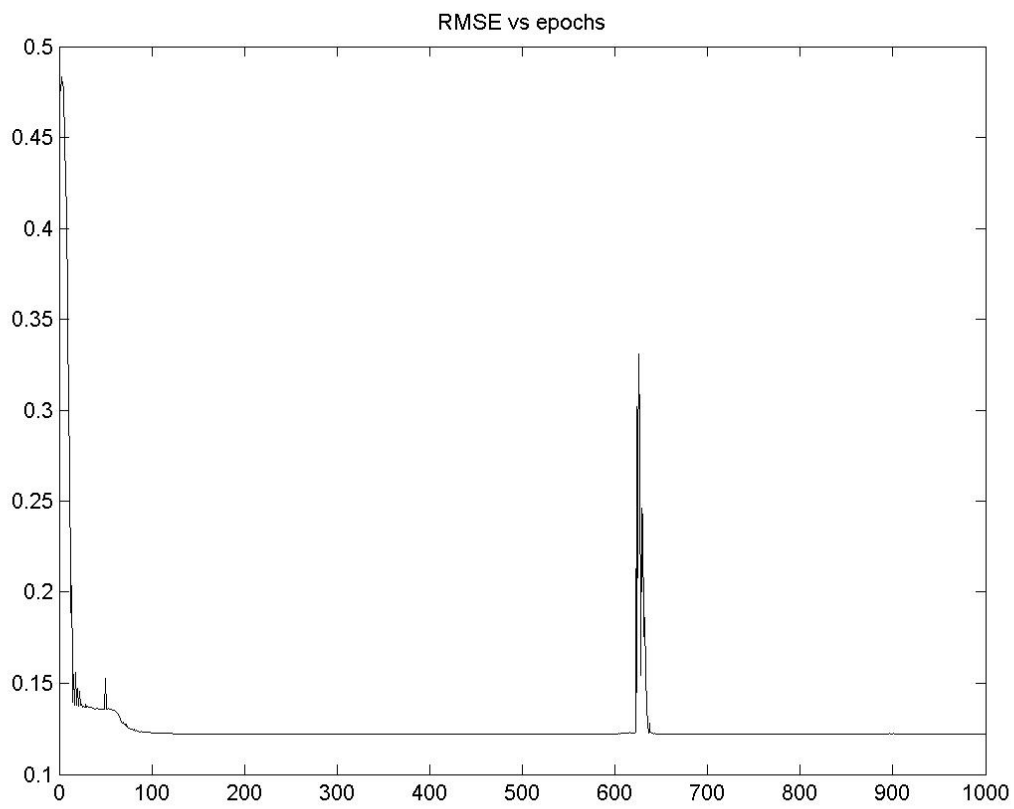
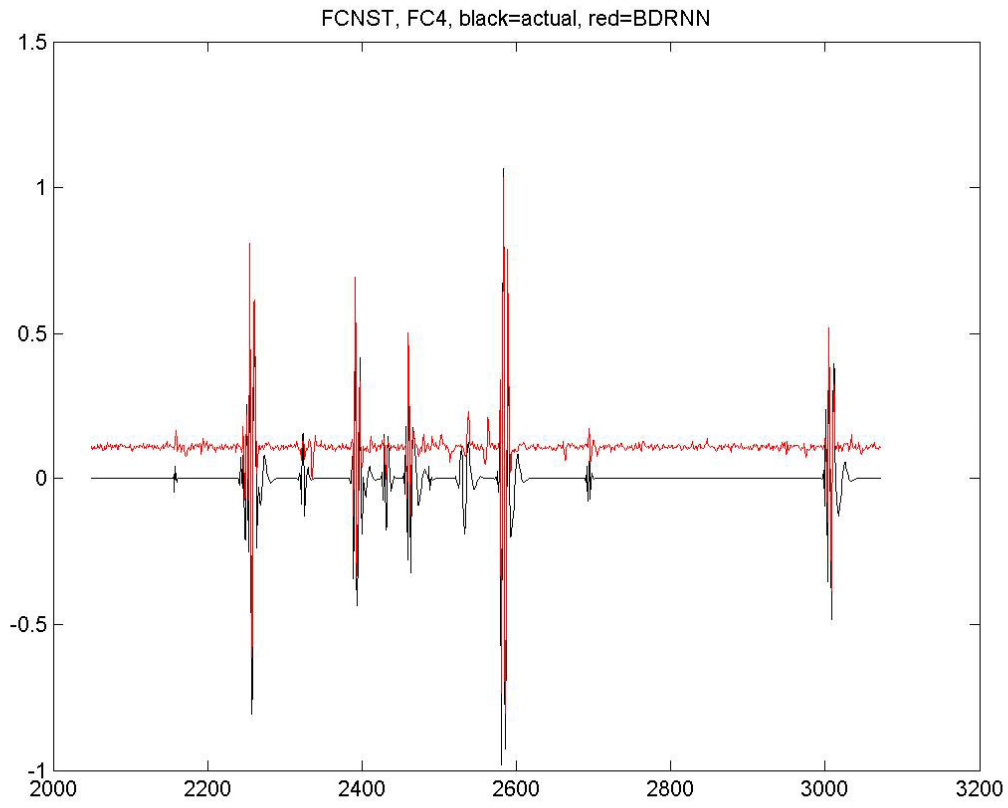
34^η Επανάληψη



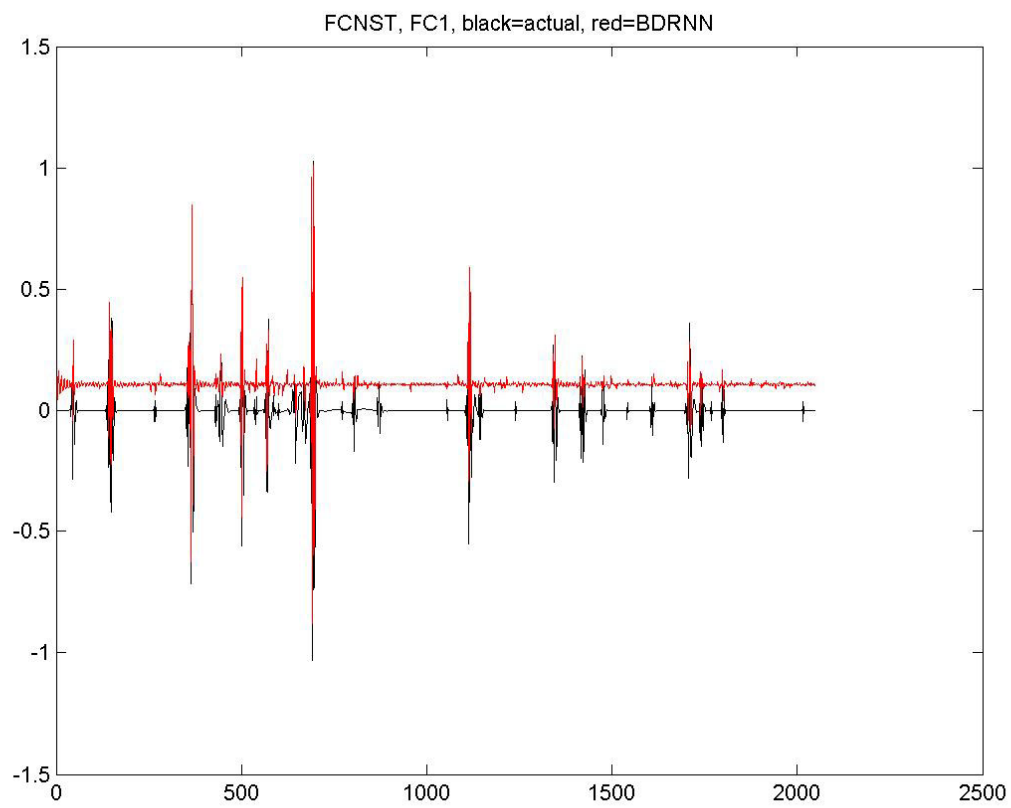
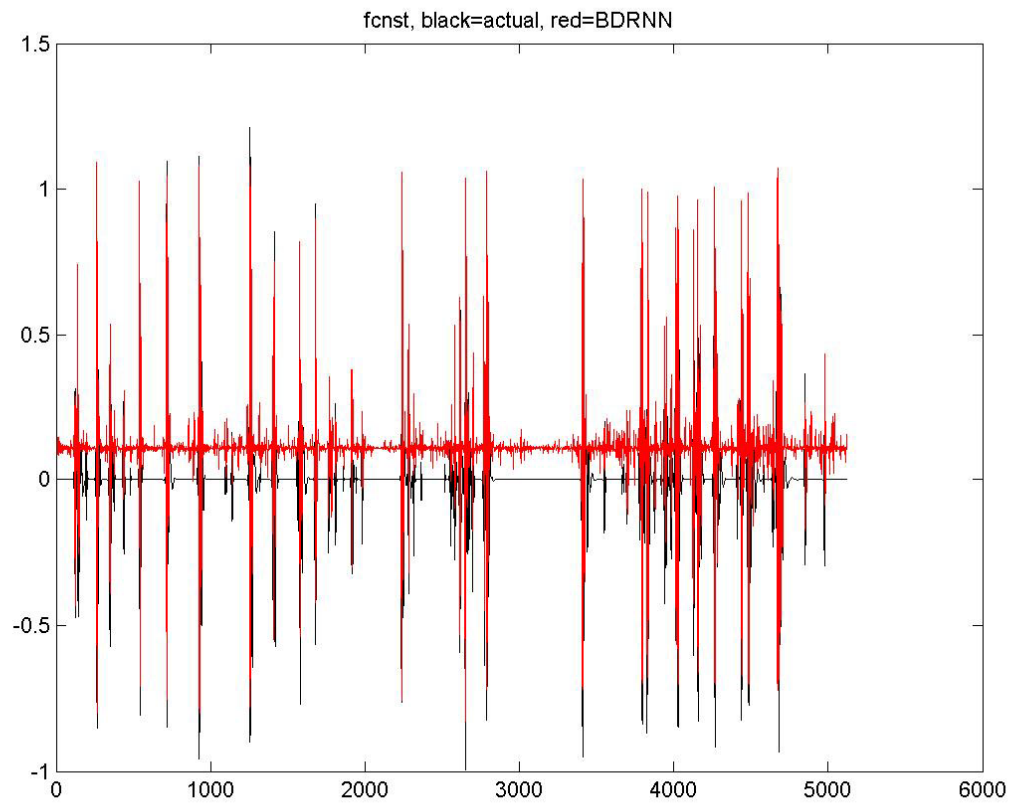


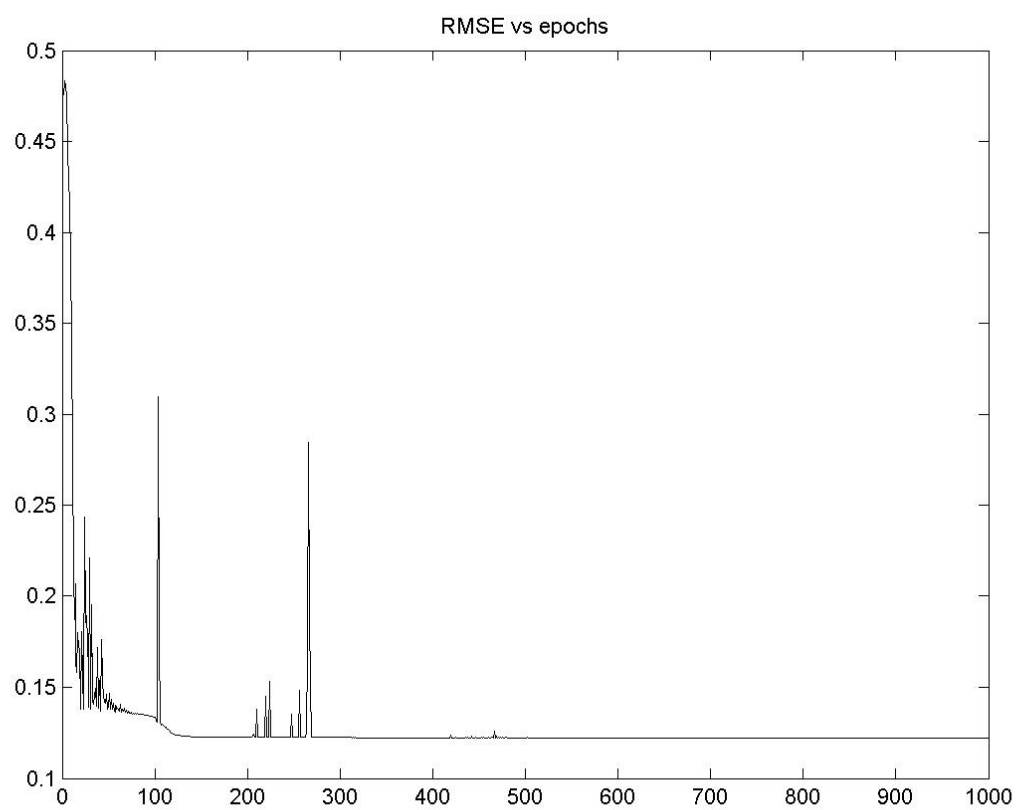
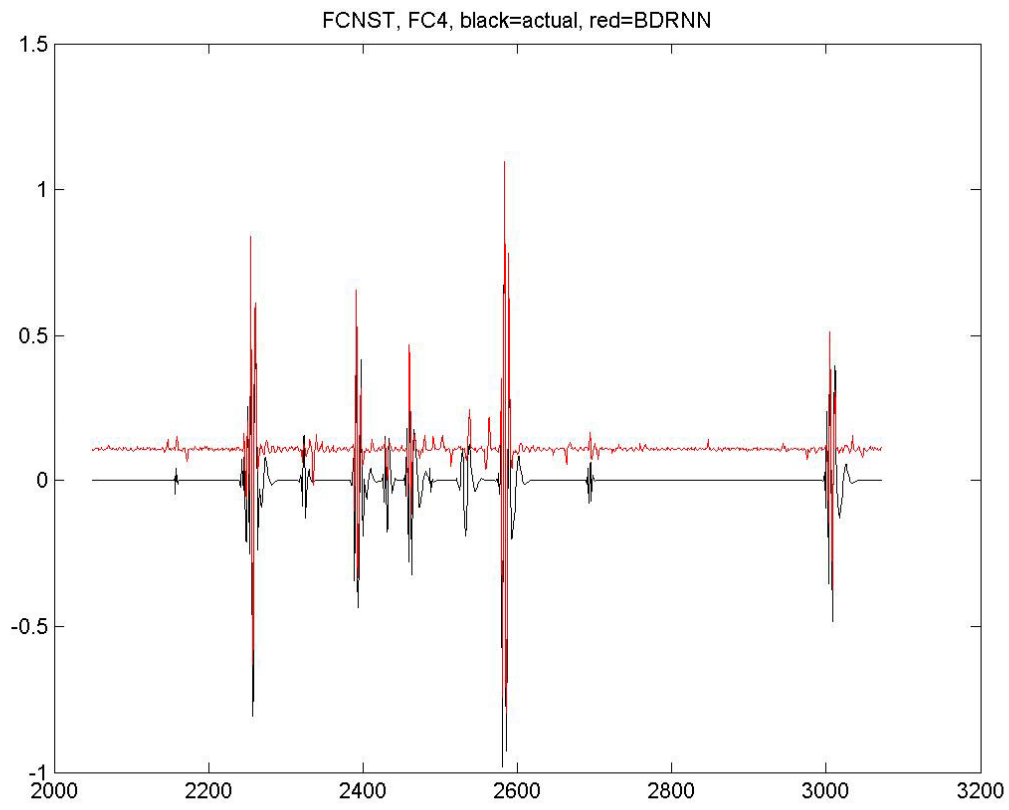
35^η Επανάληψη



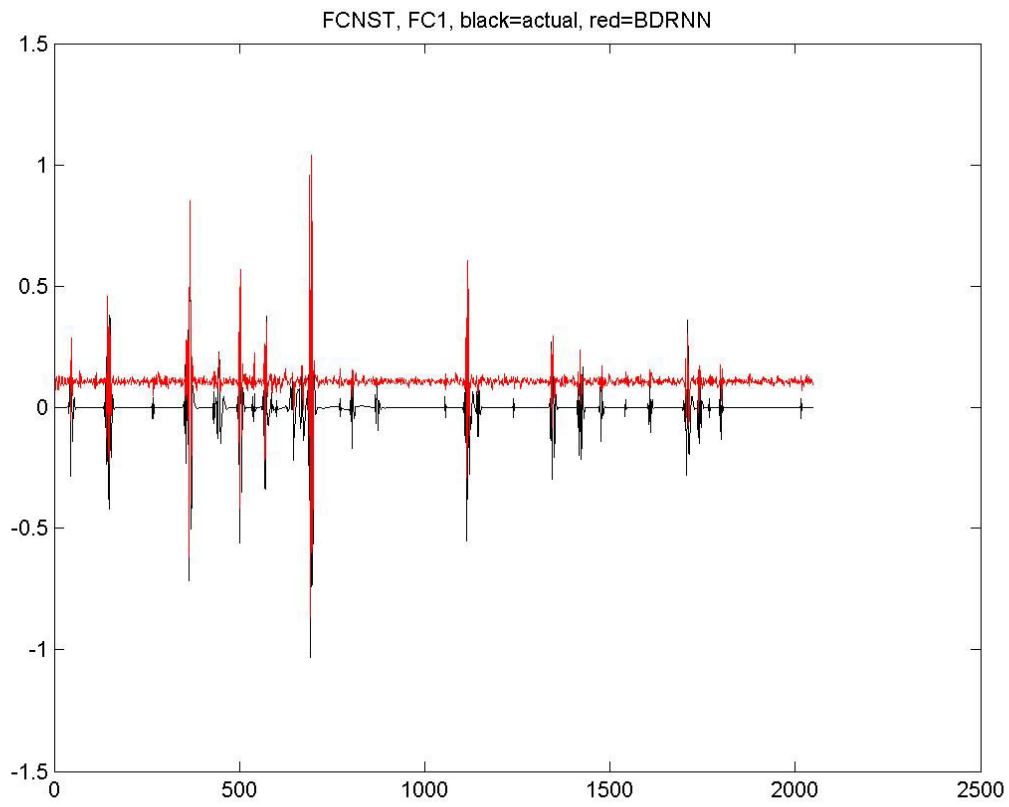
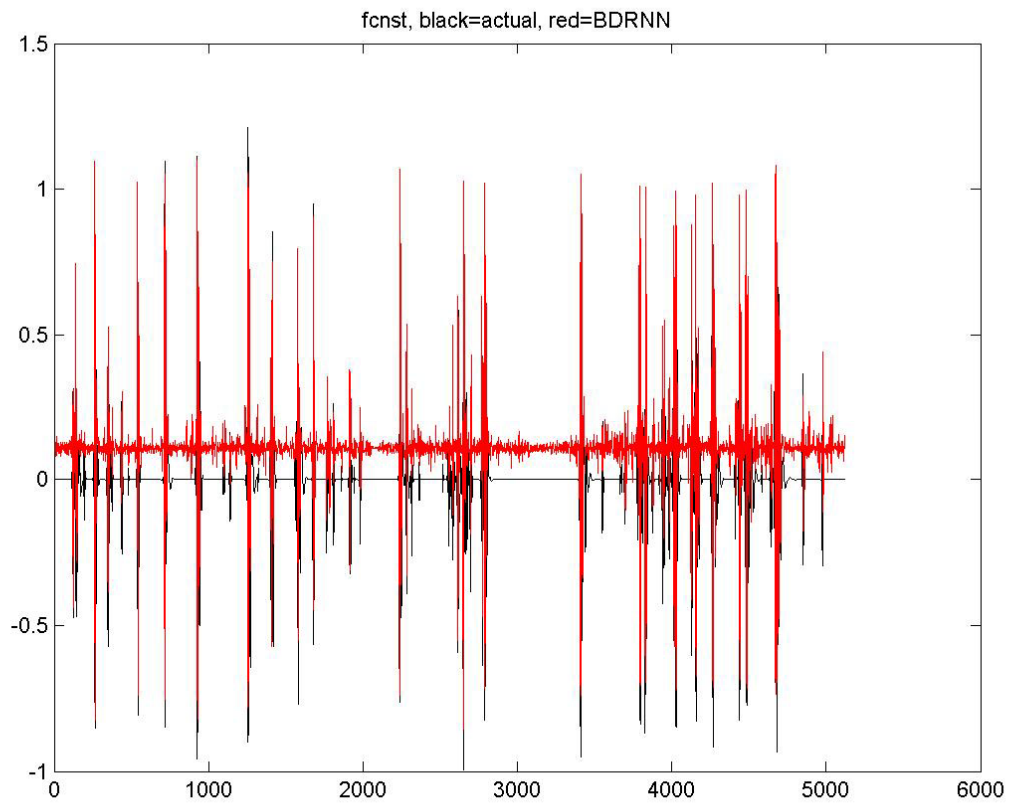


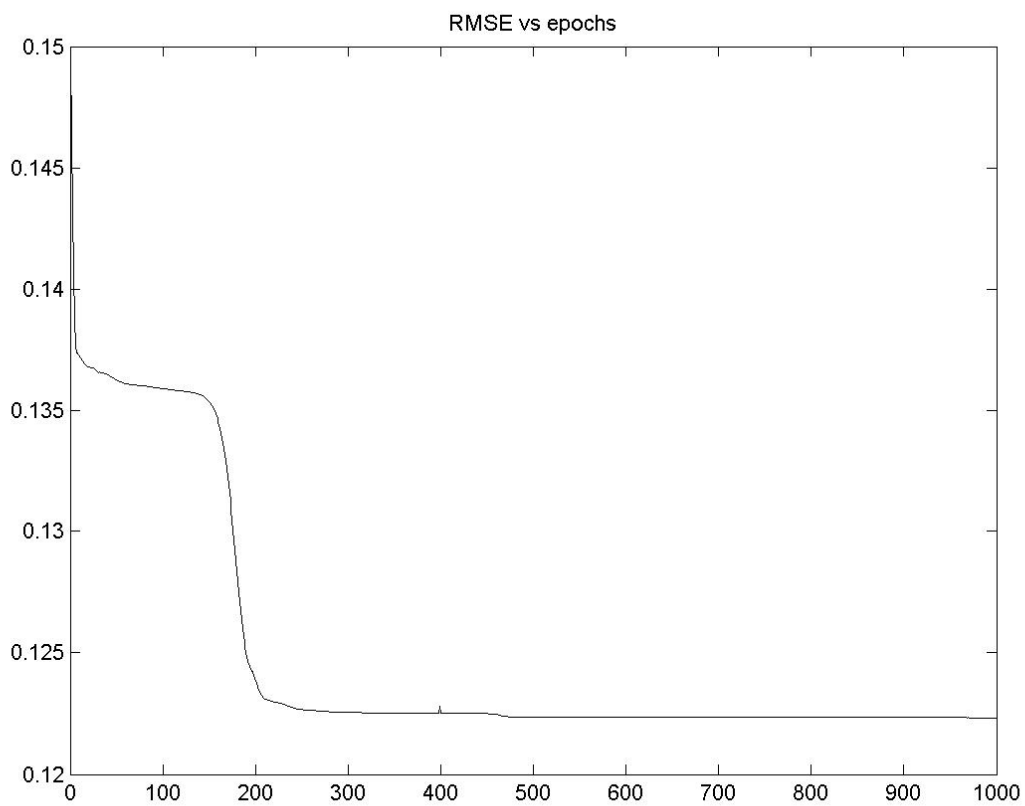
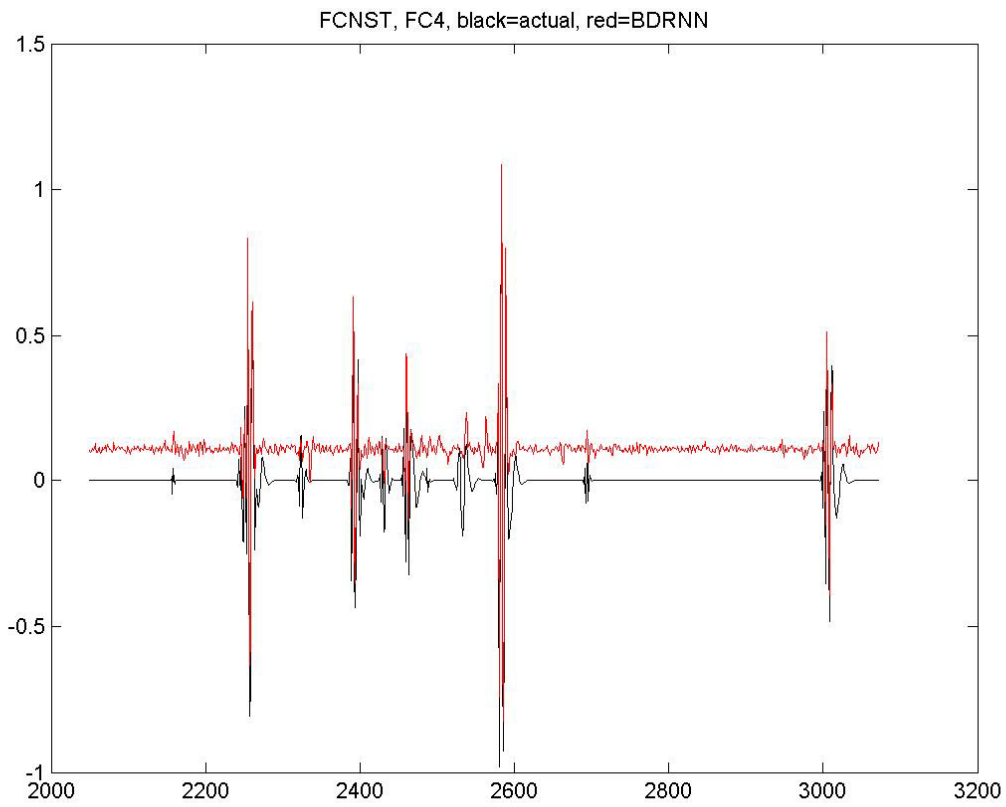
36^η Επανάληψη



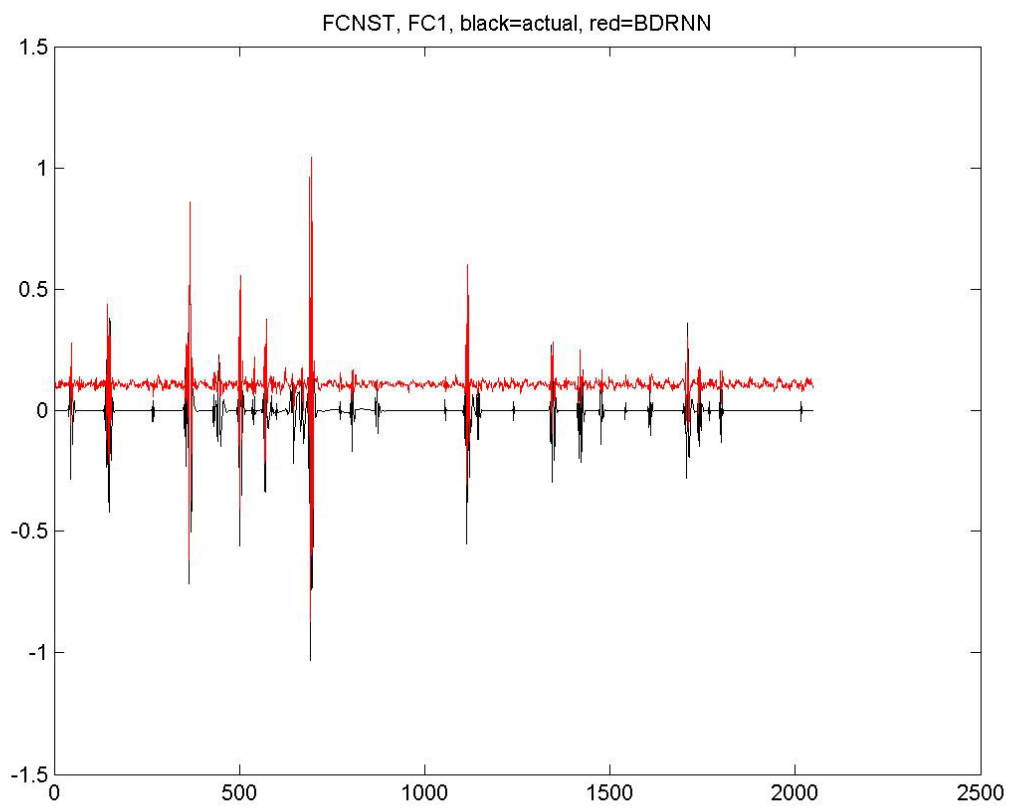
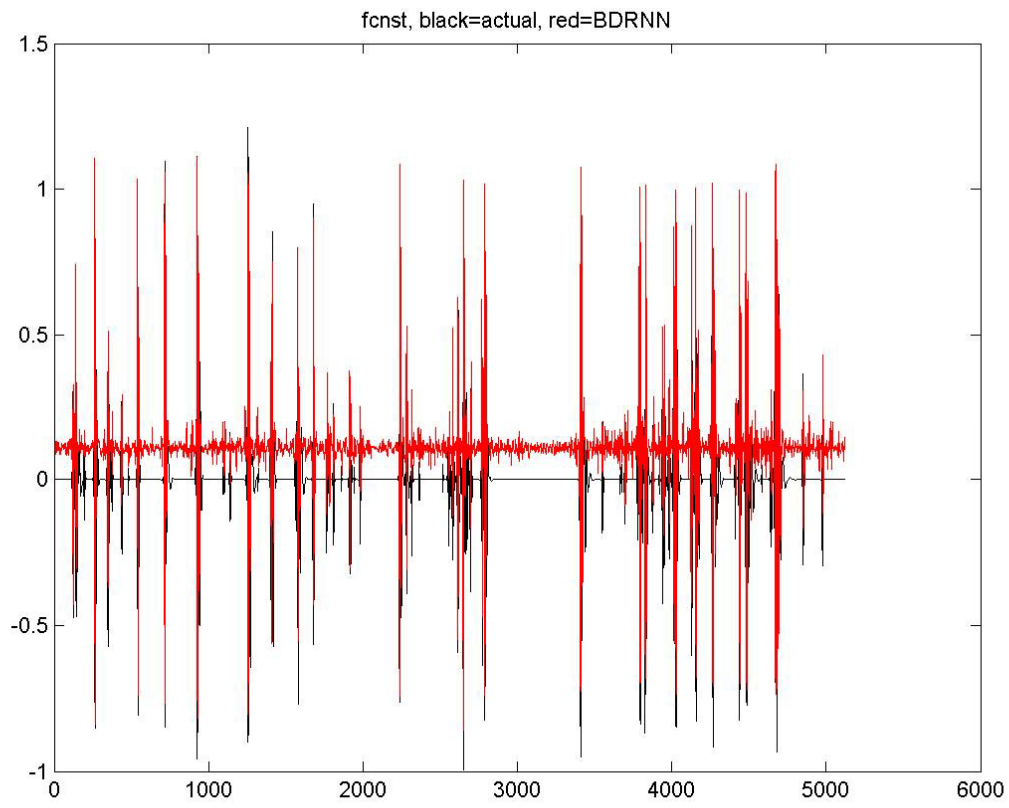


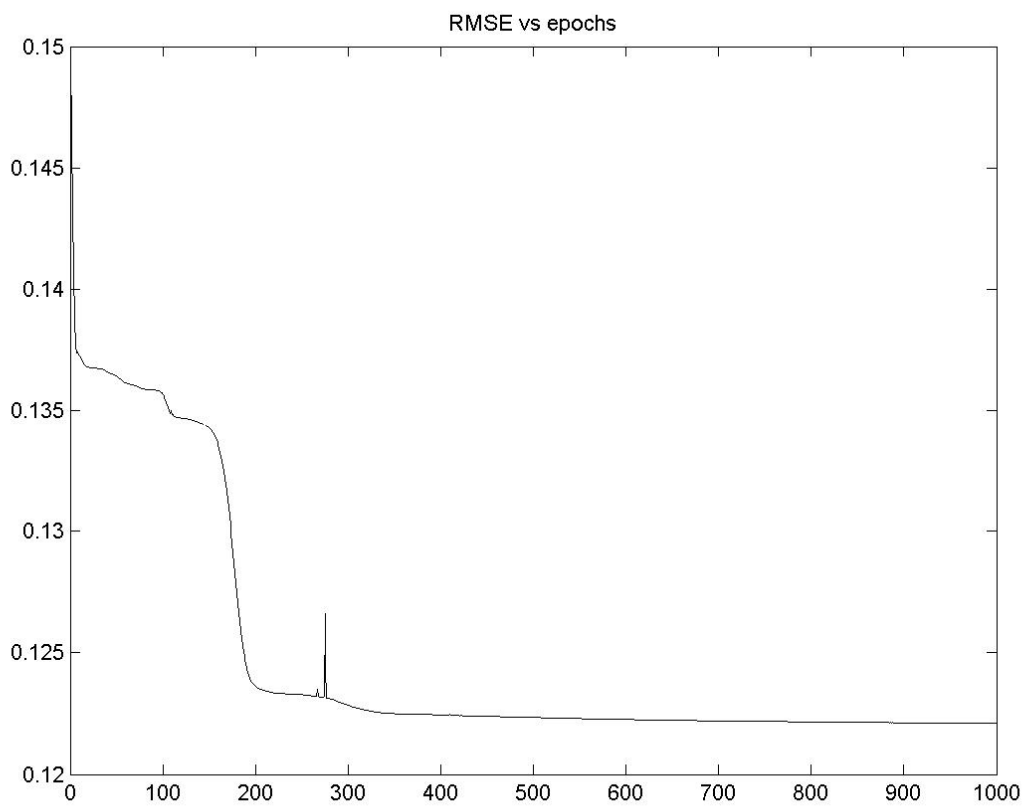
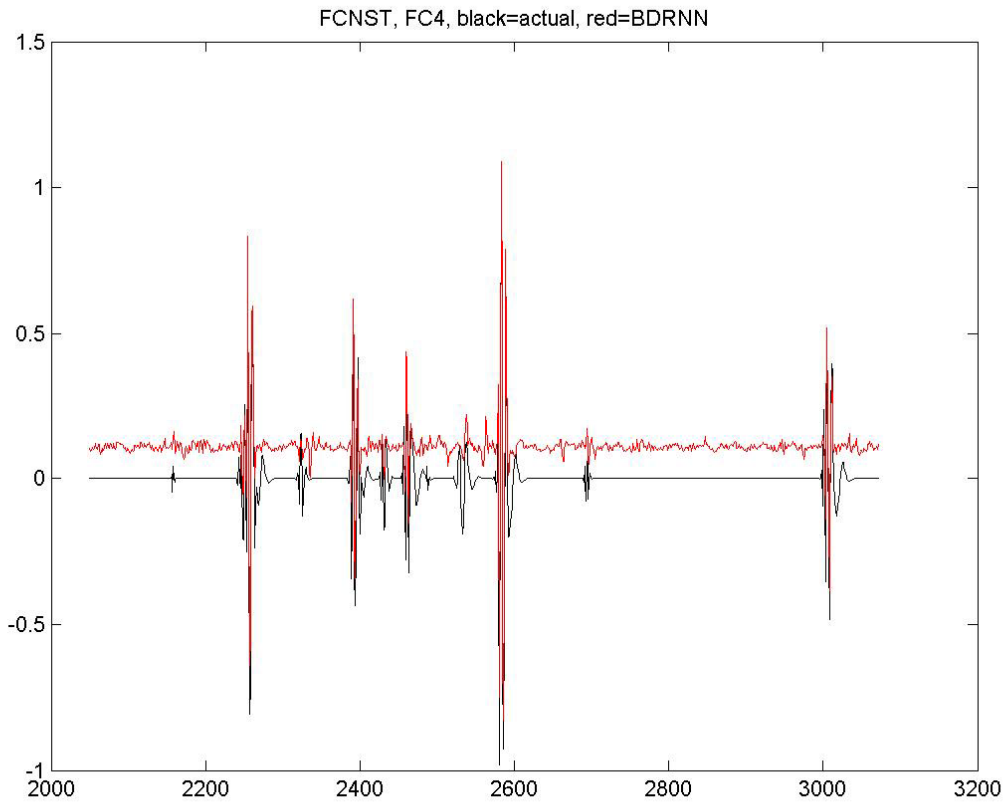
37^η Επανάληψη



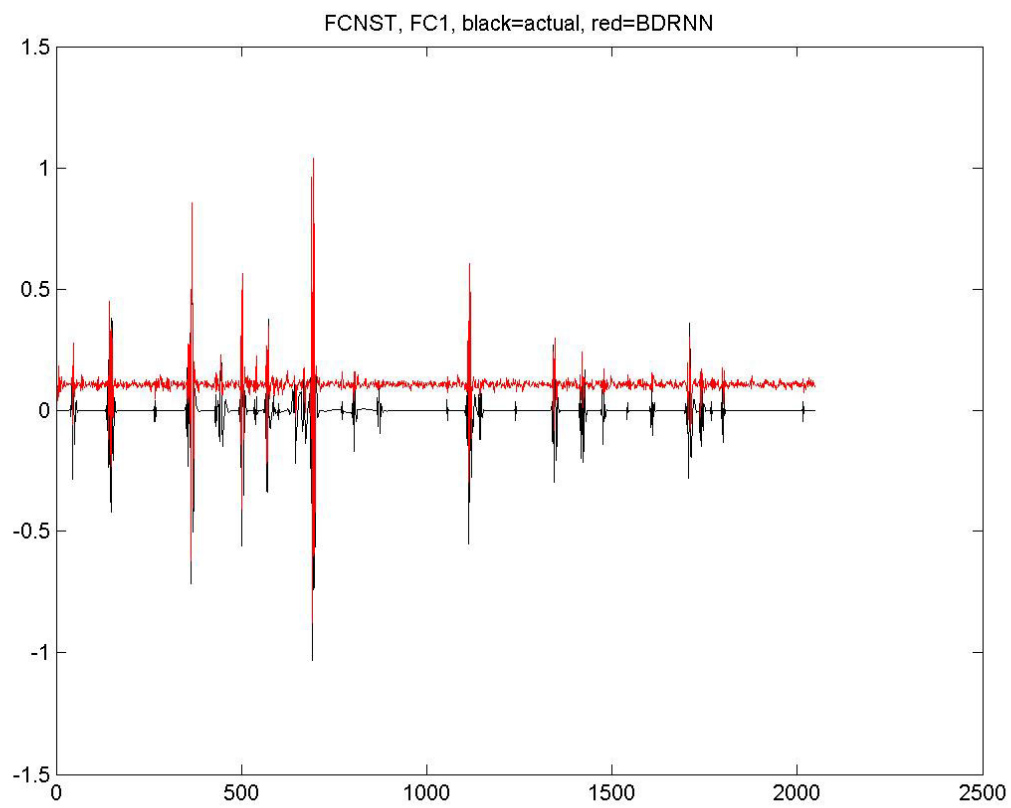
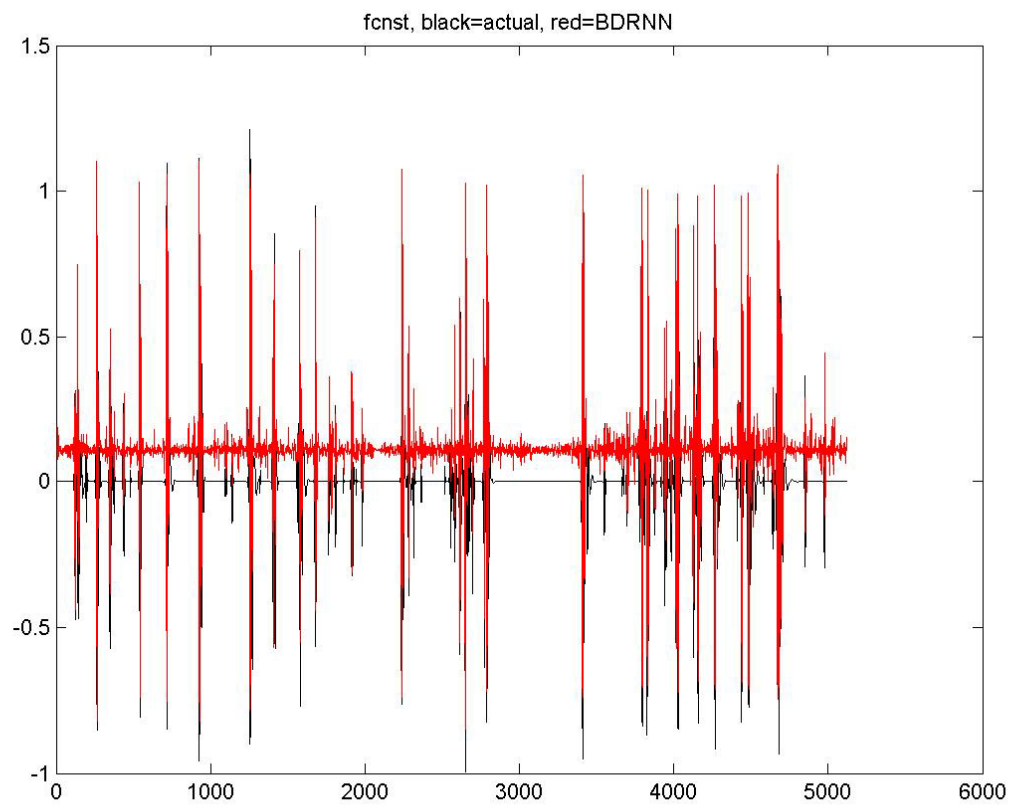


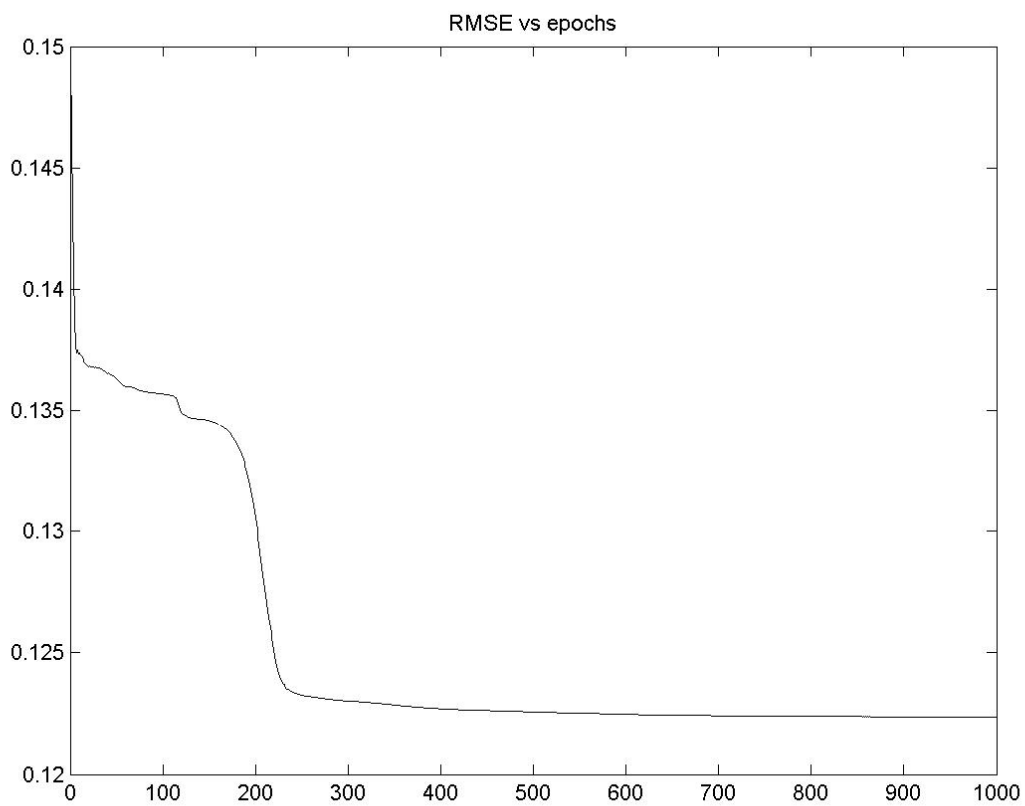
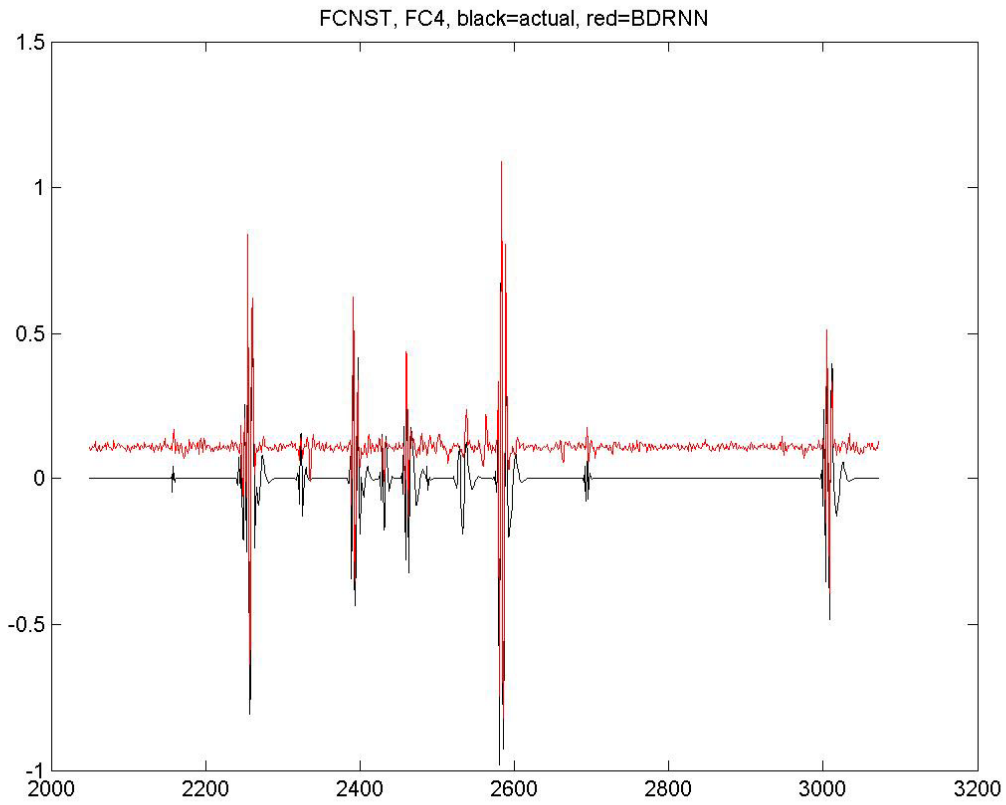
38^η Επανάληψη



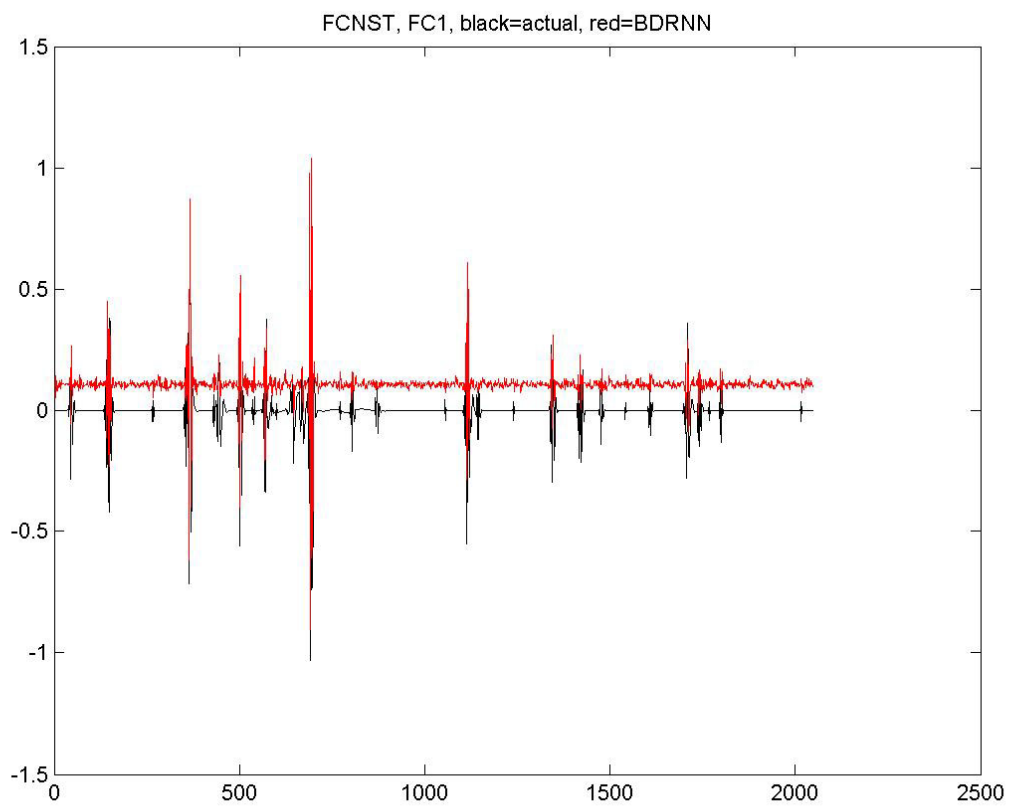
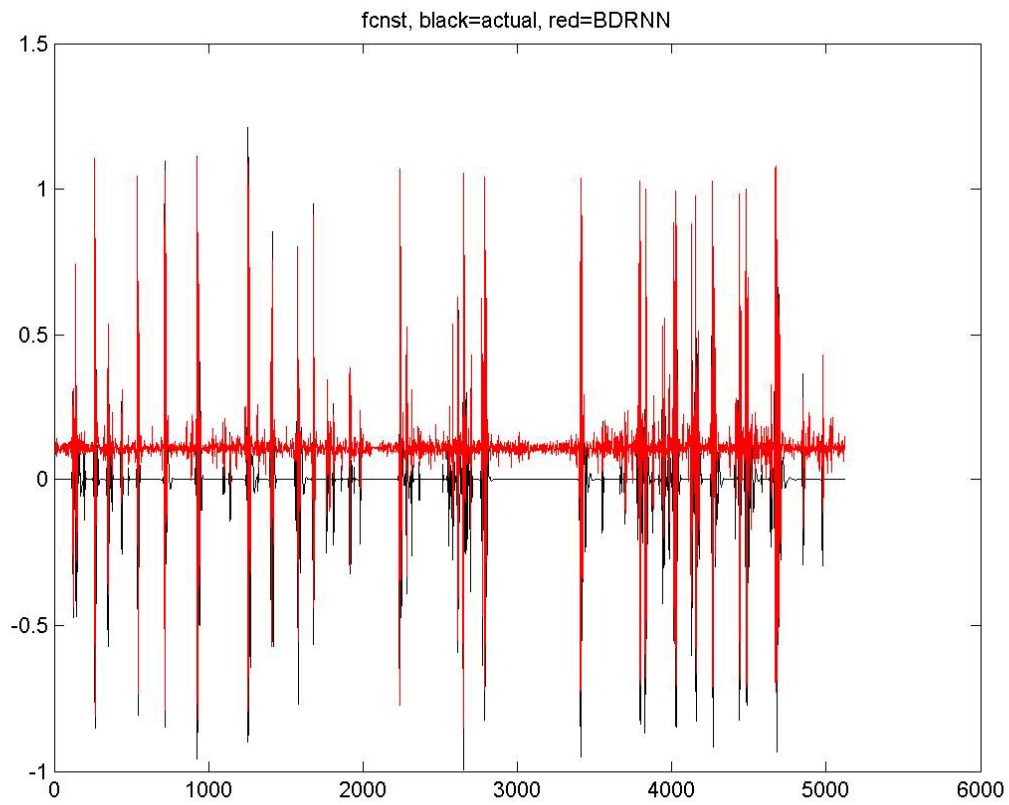


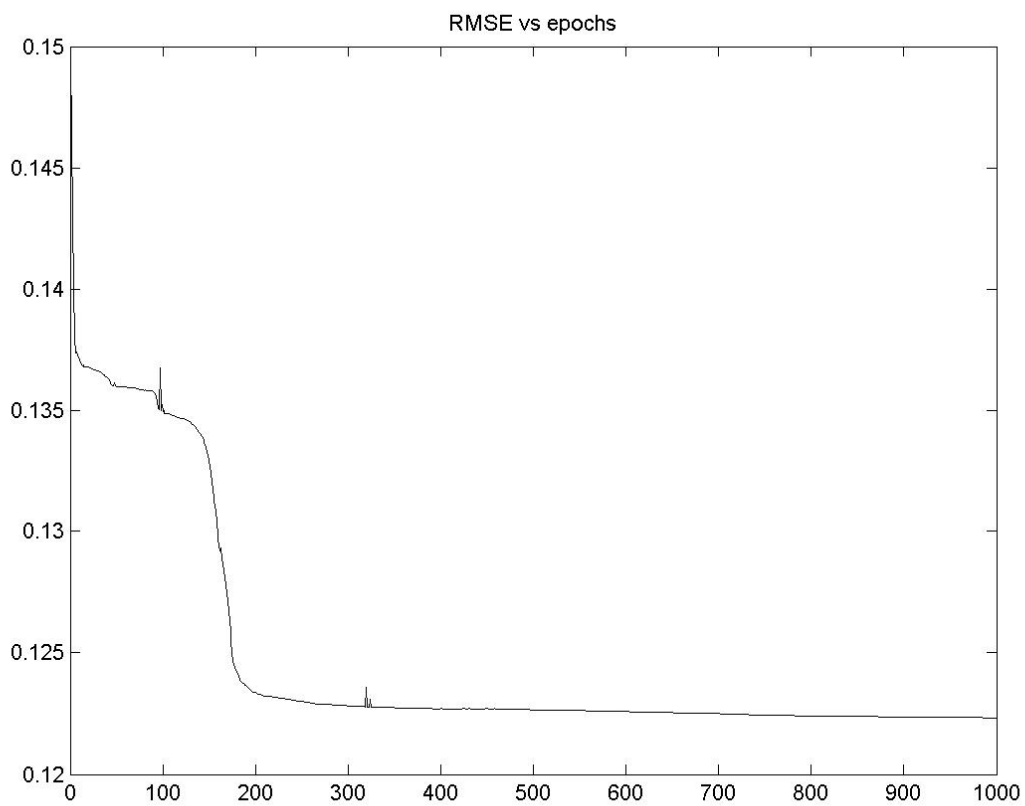
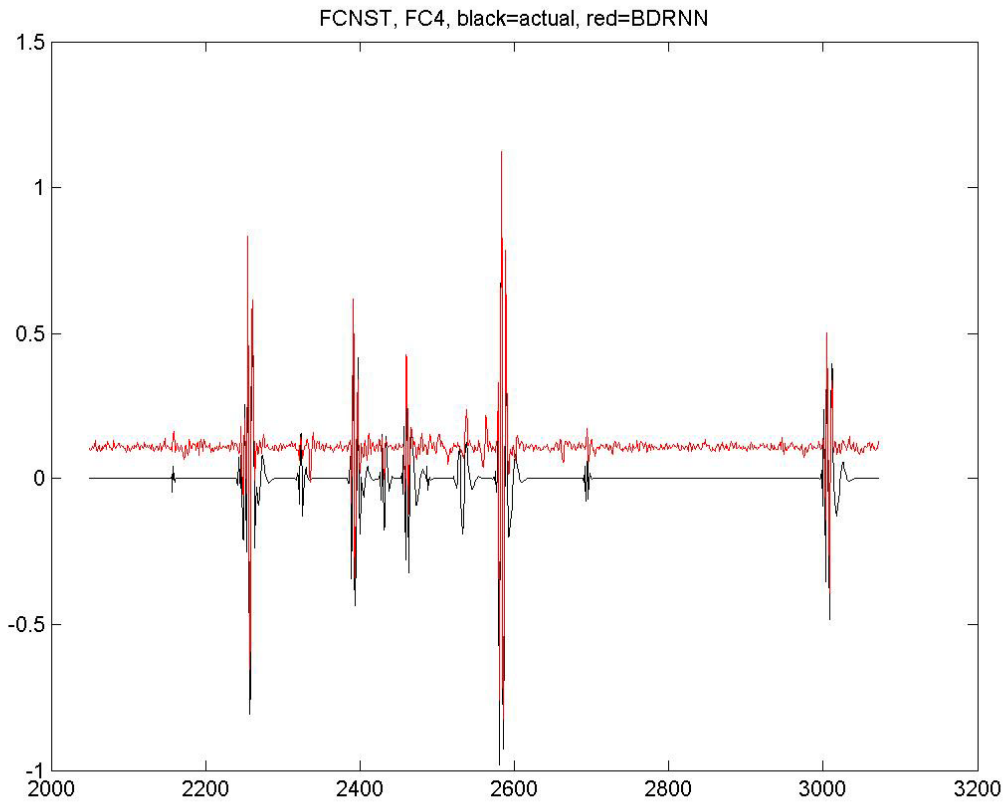
39^η Επανάληψη



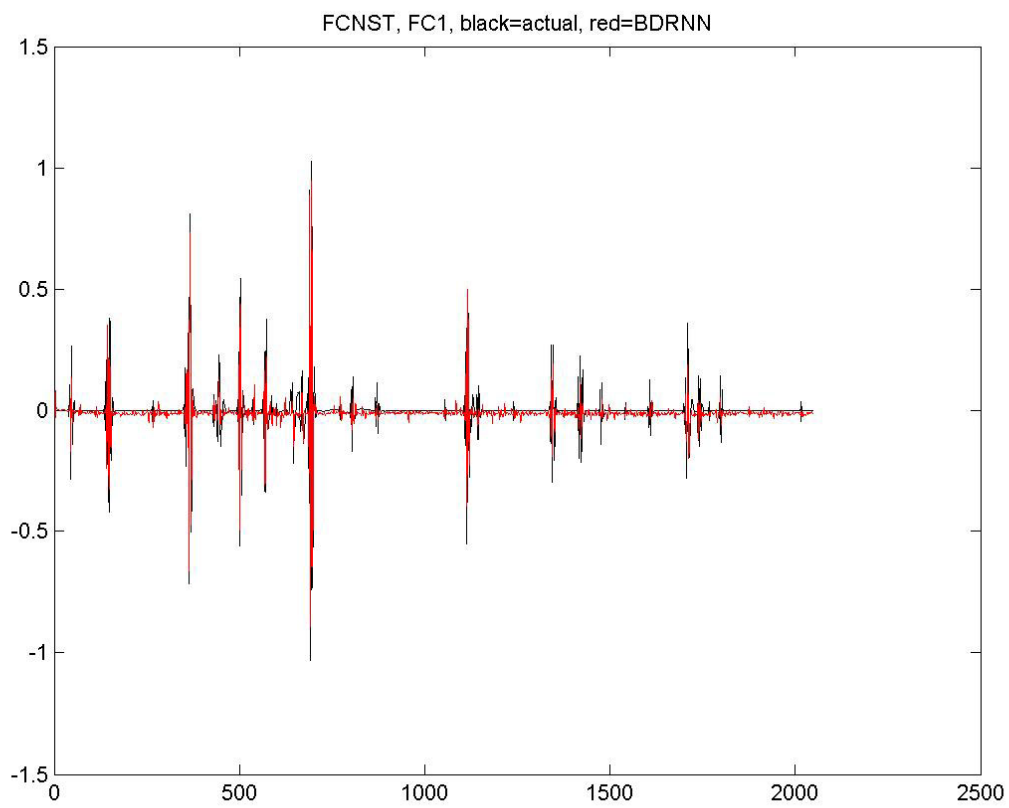
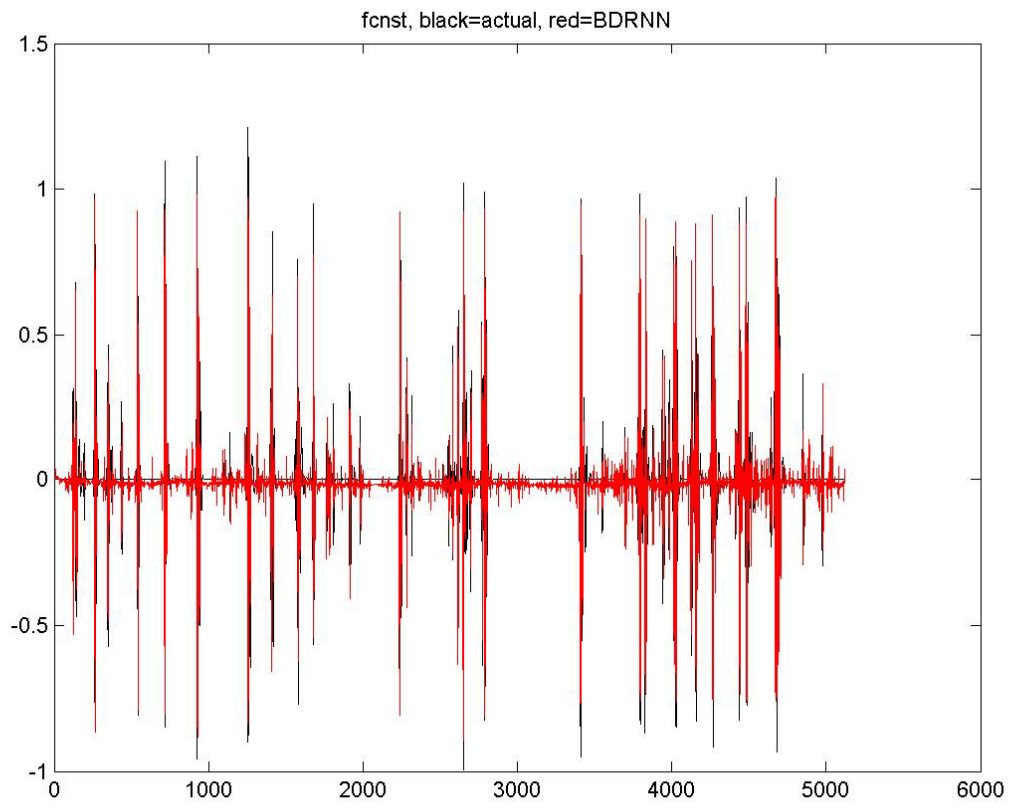


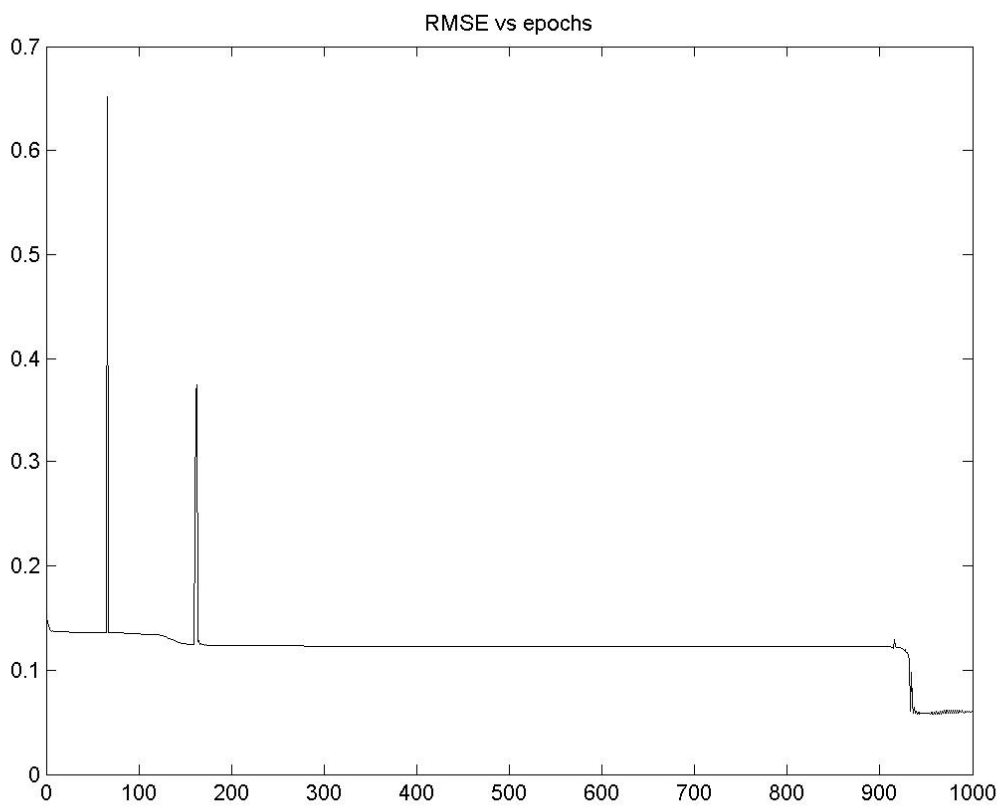
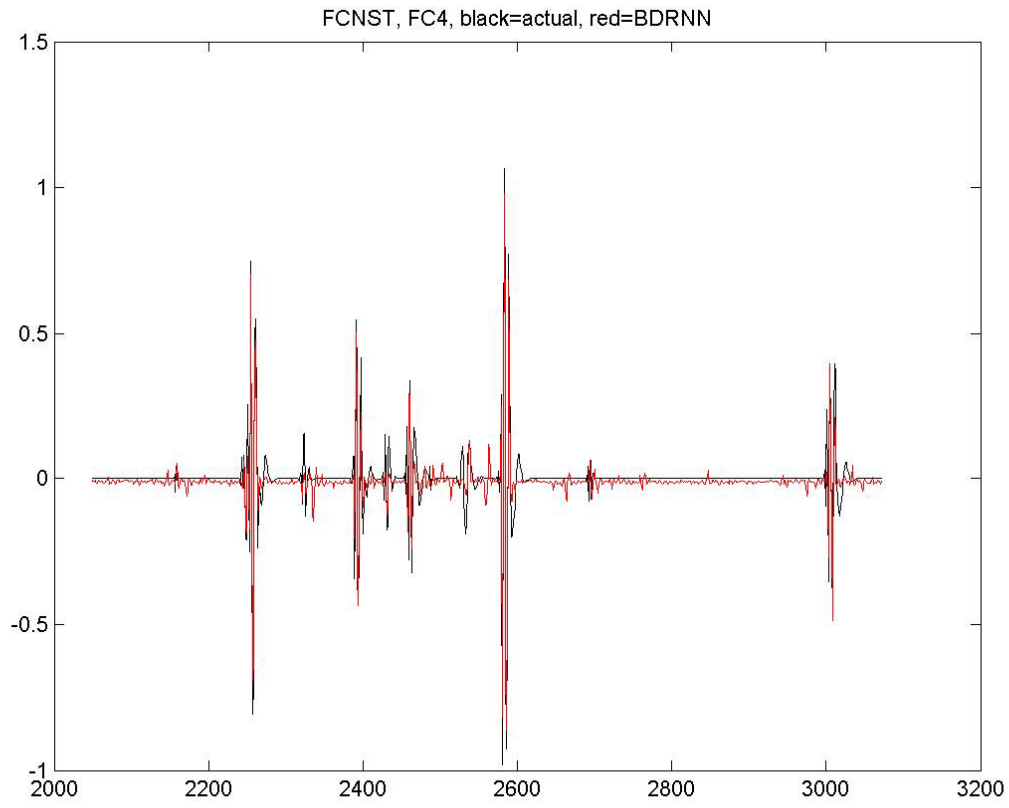
40^η Επανάληψη



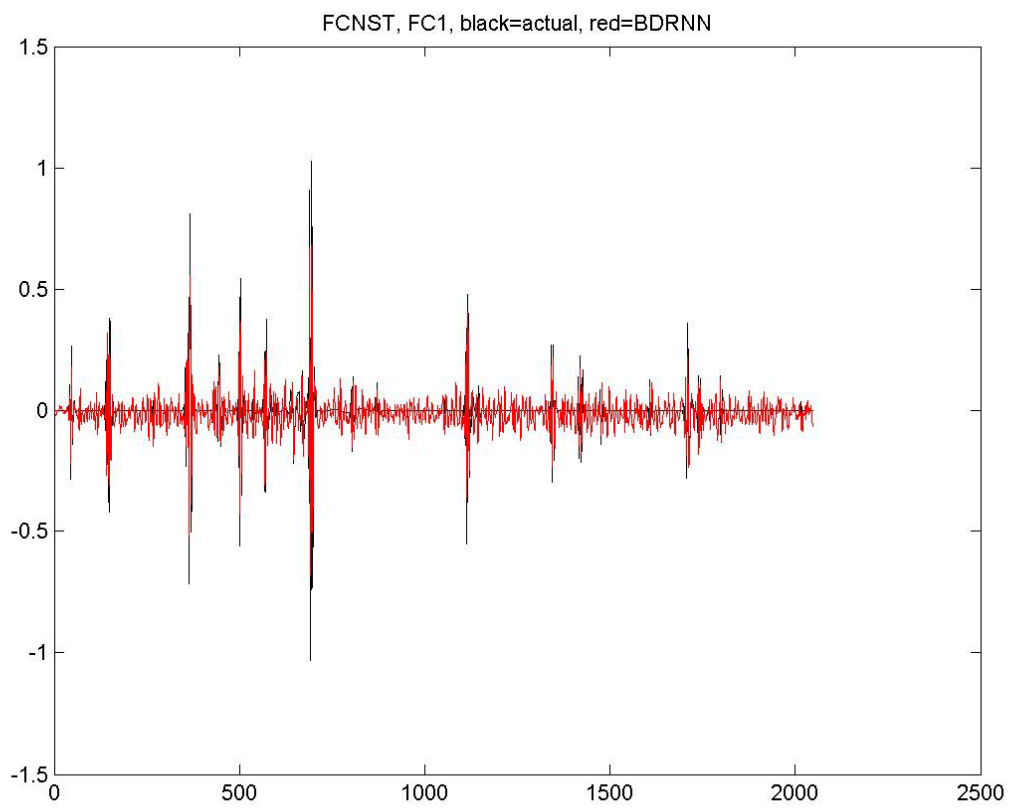
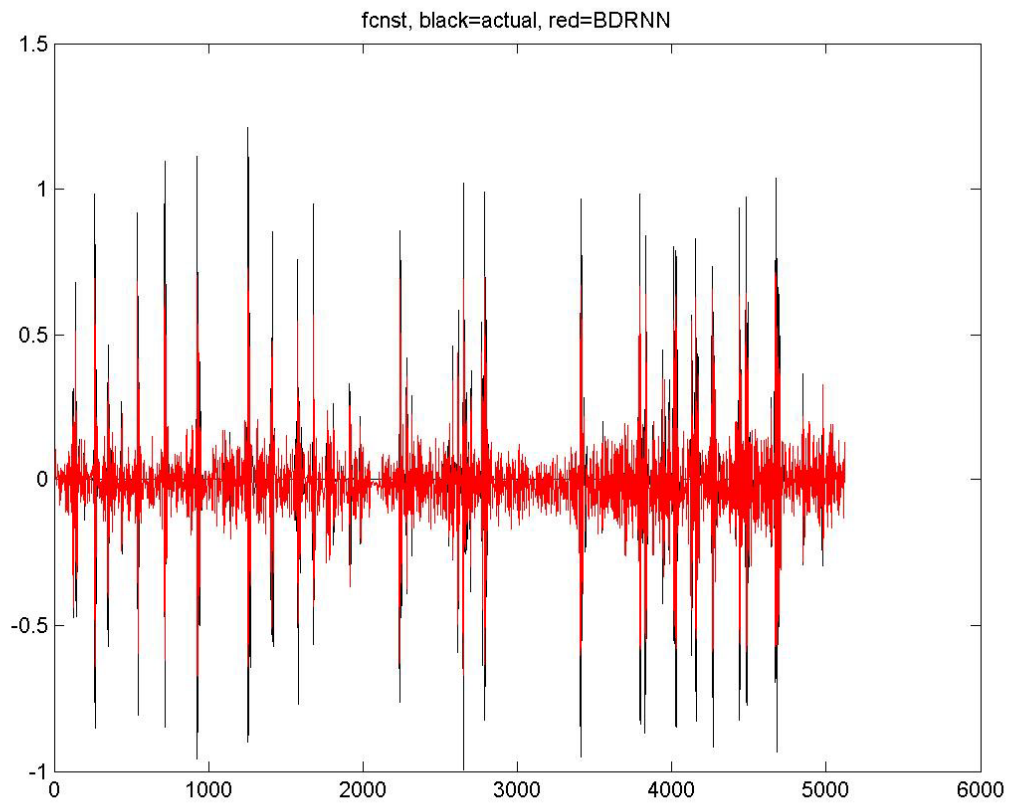


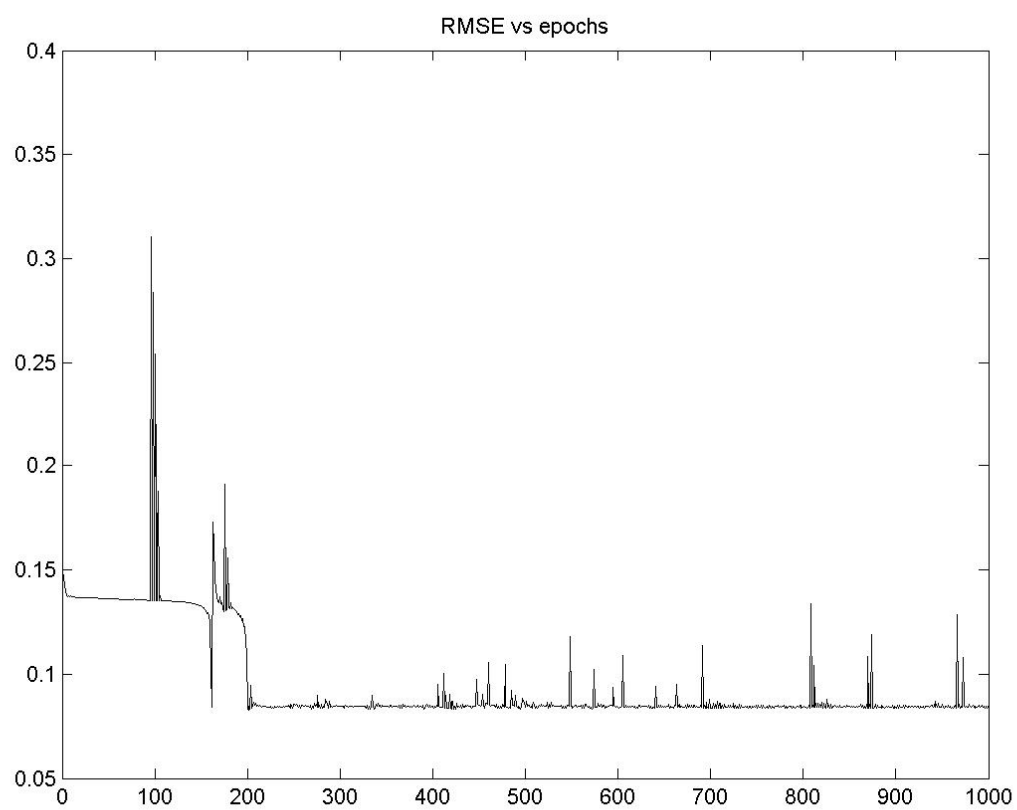
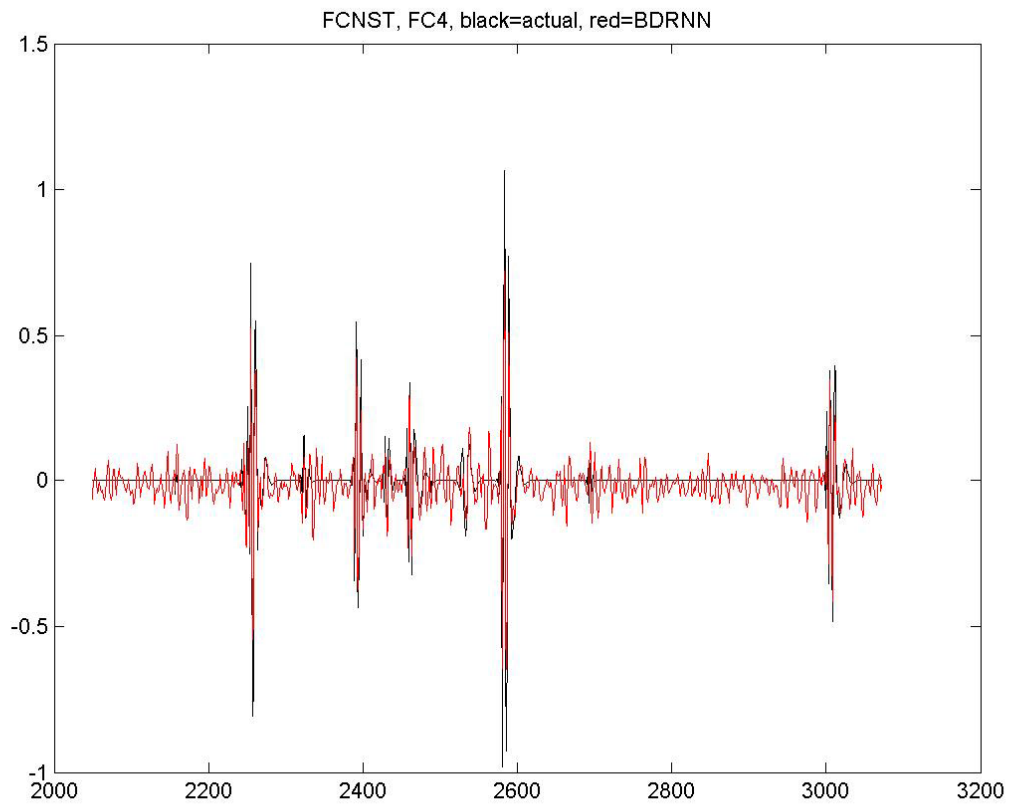
41^η Επανάληψη



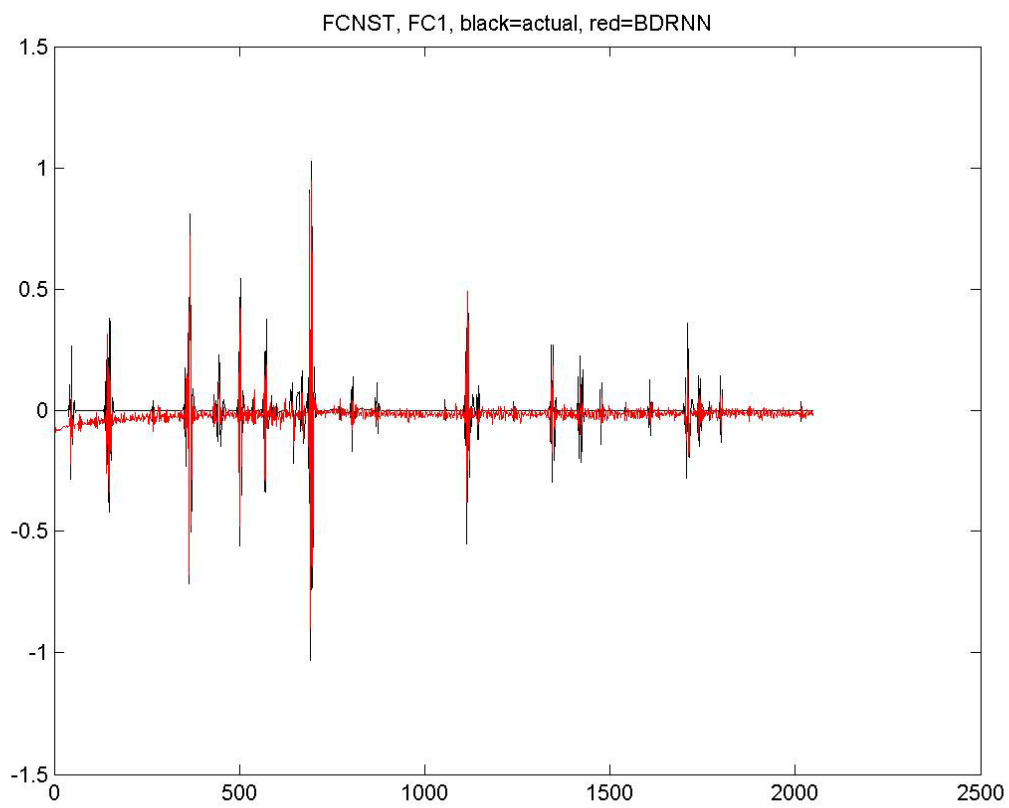
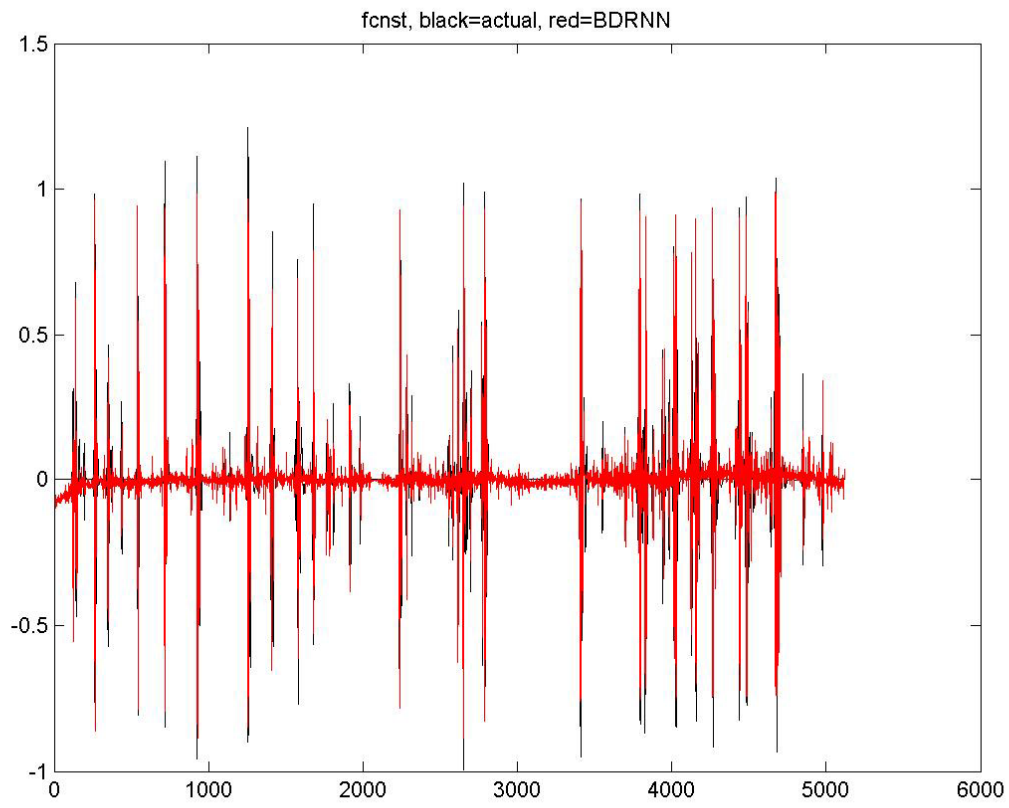


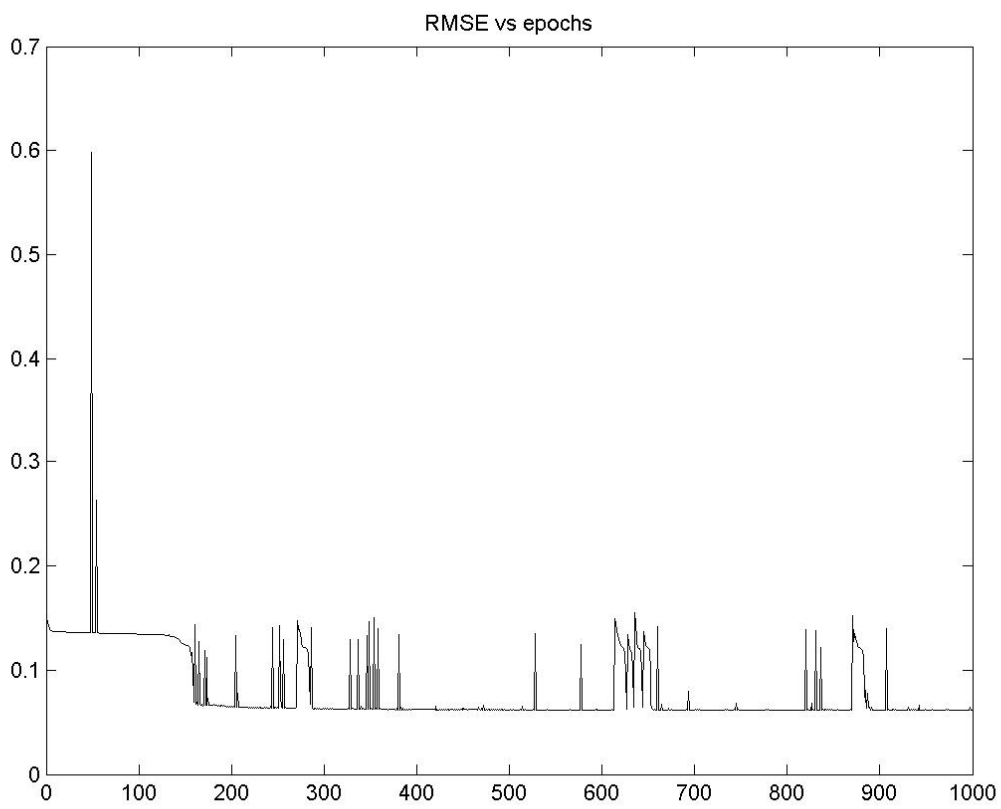
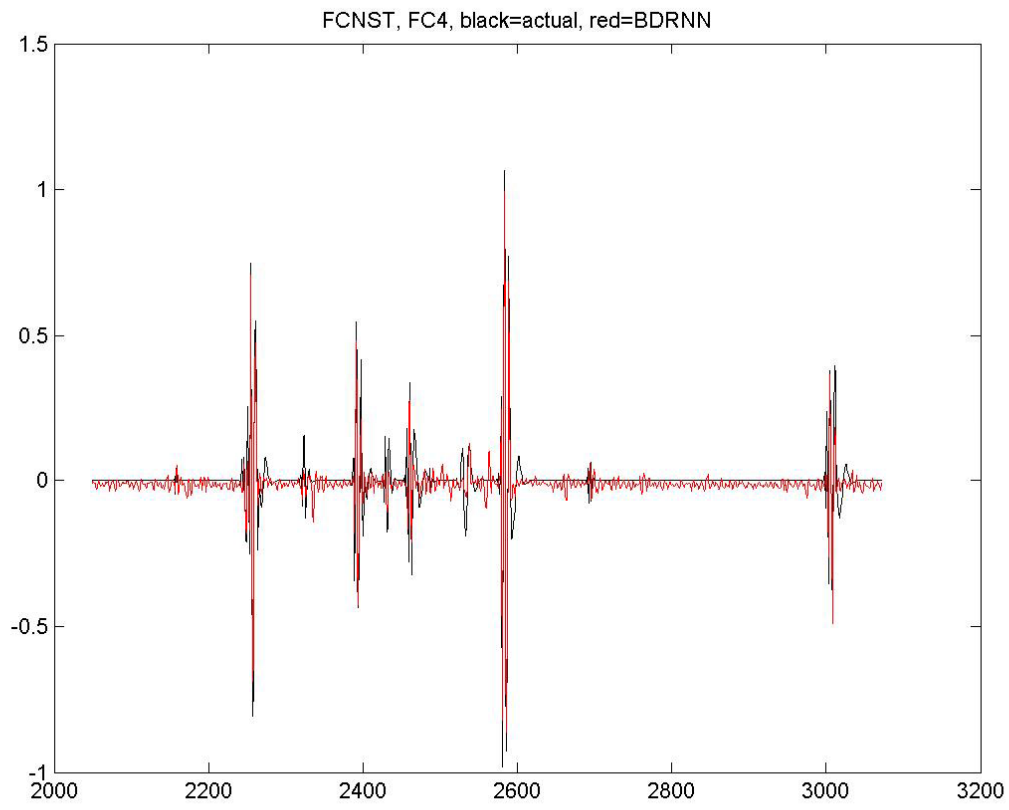
42^η Επανάληψη



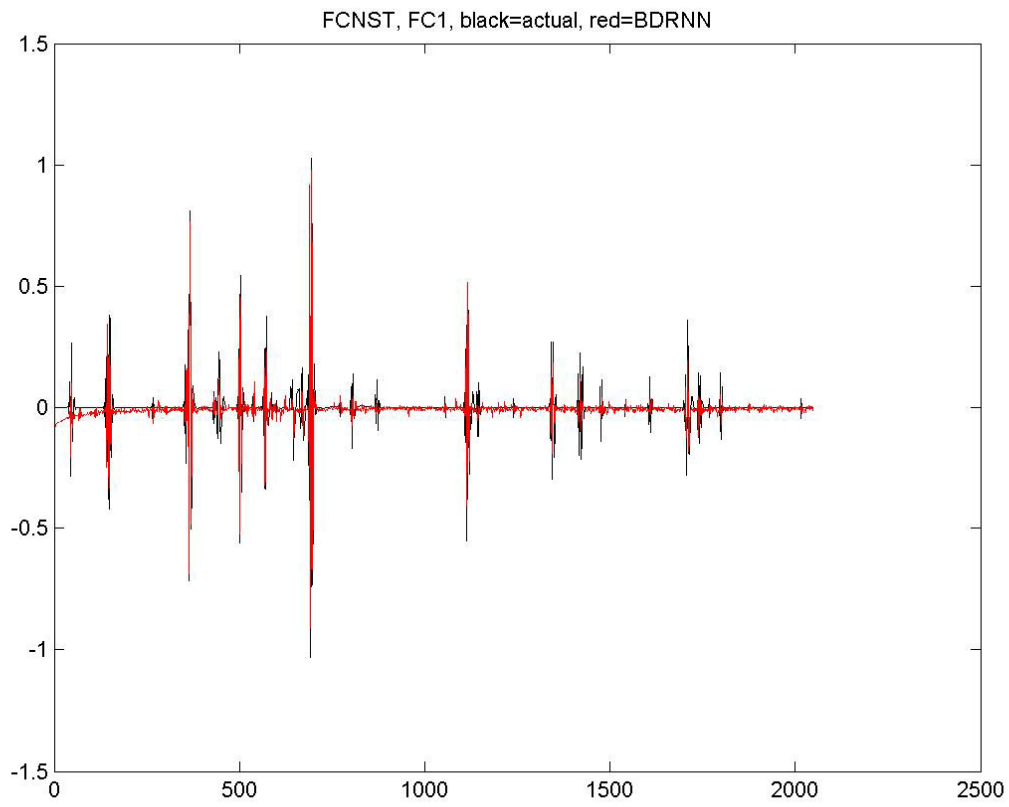
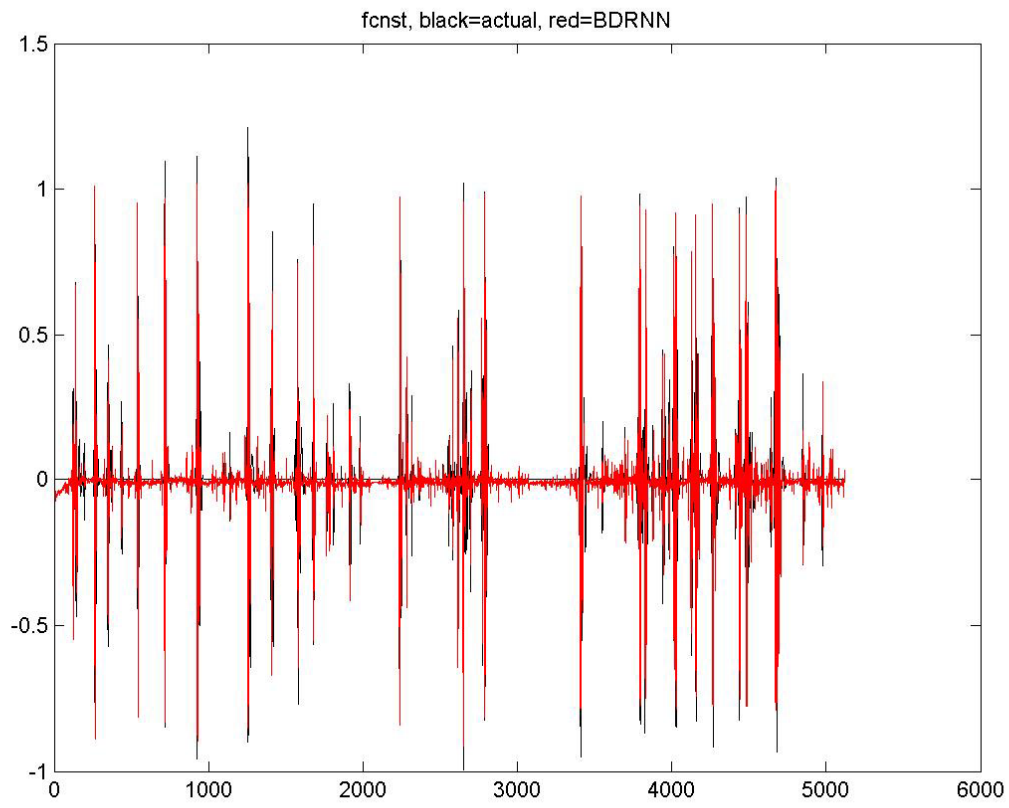


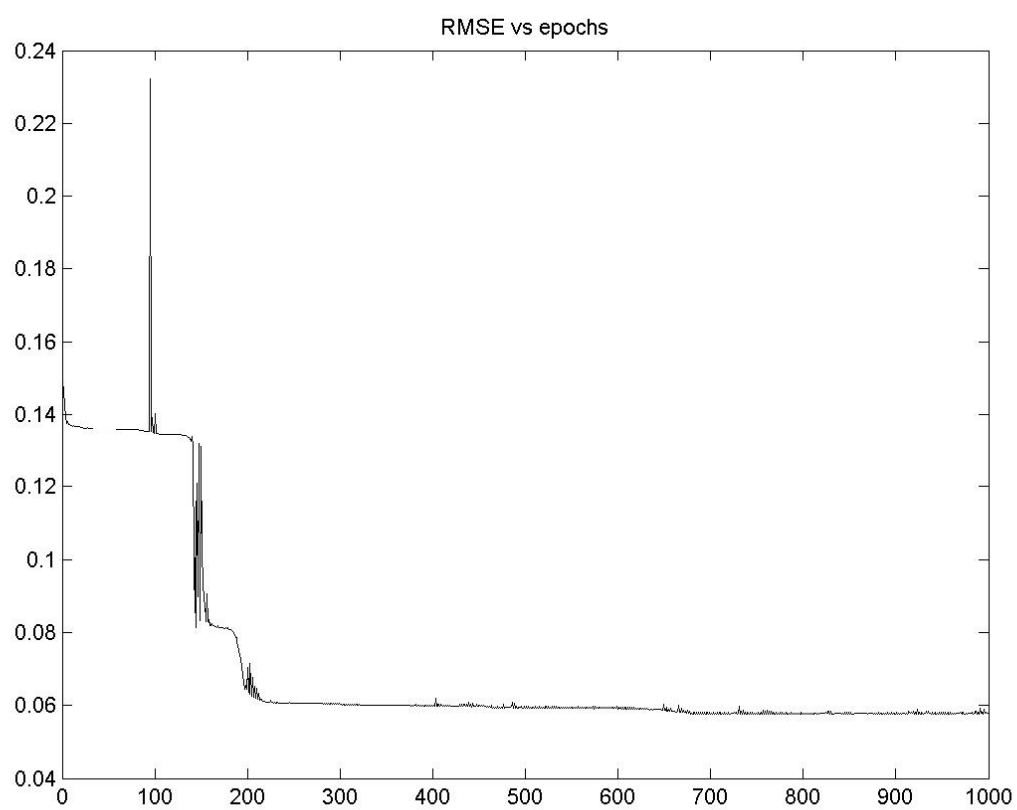
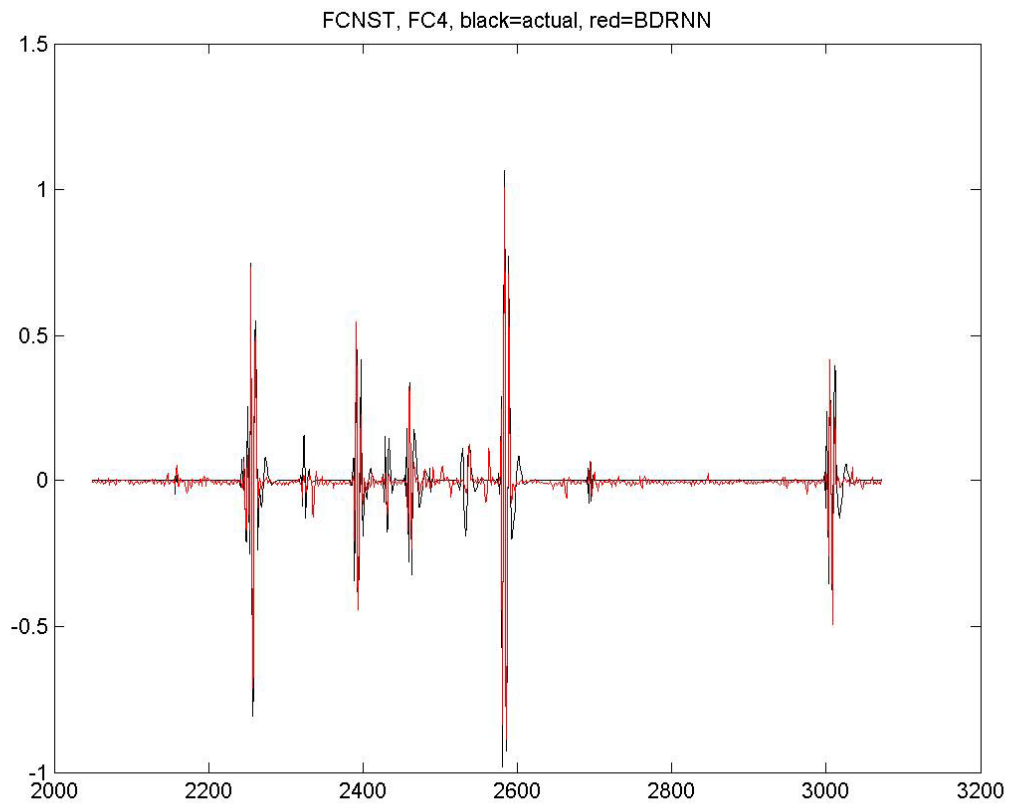
43^η Επανάληψη



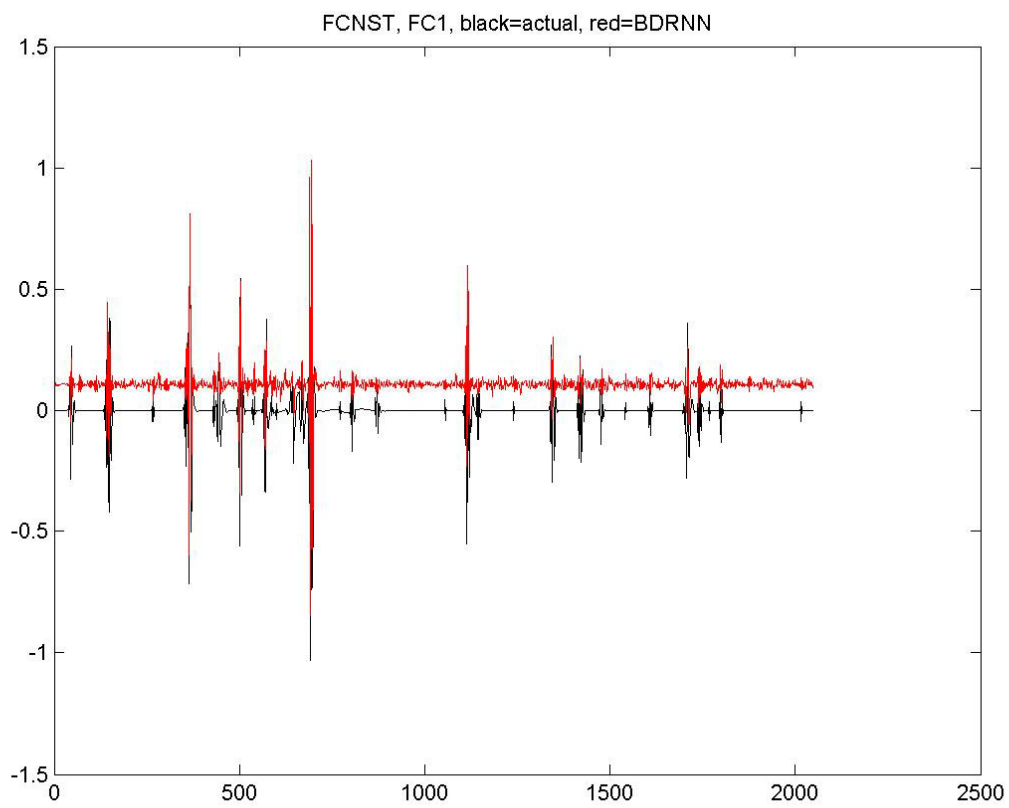
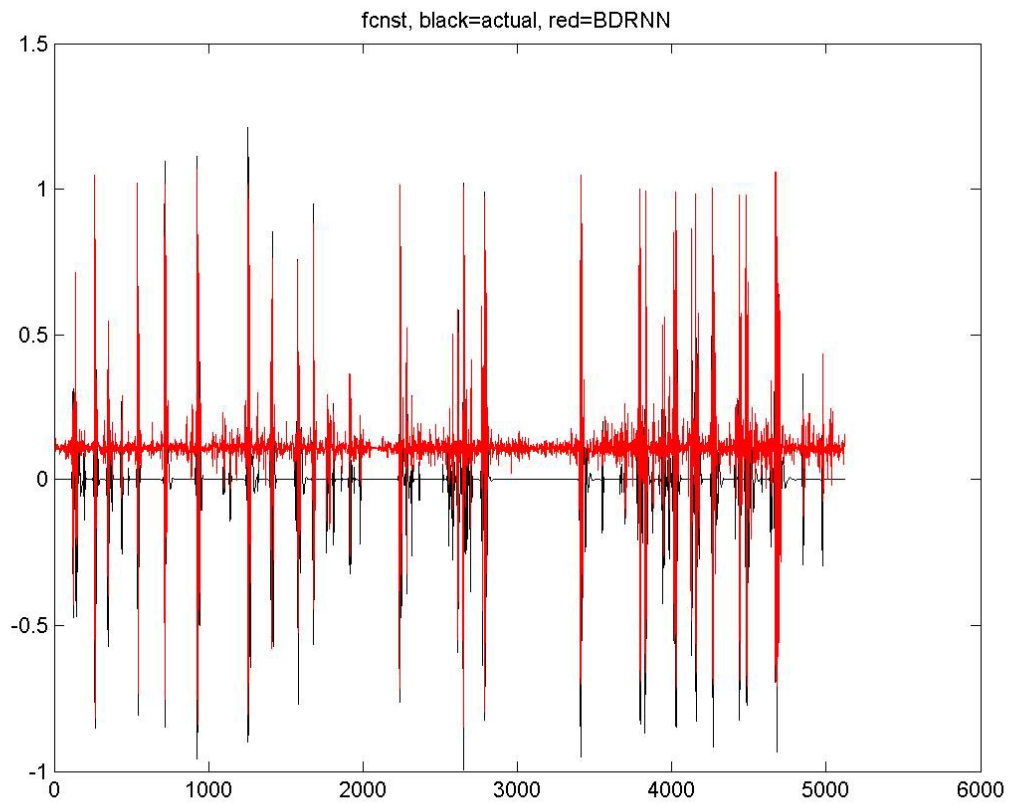


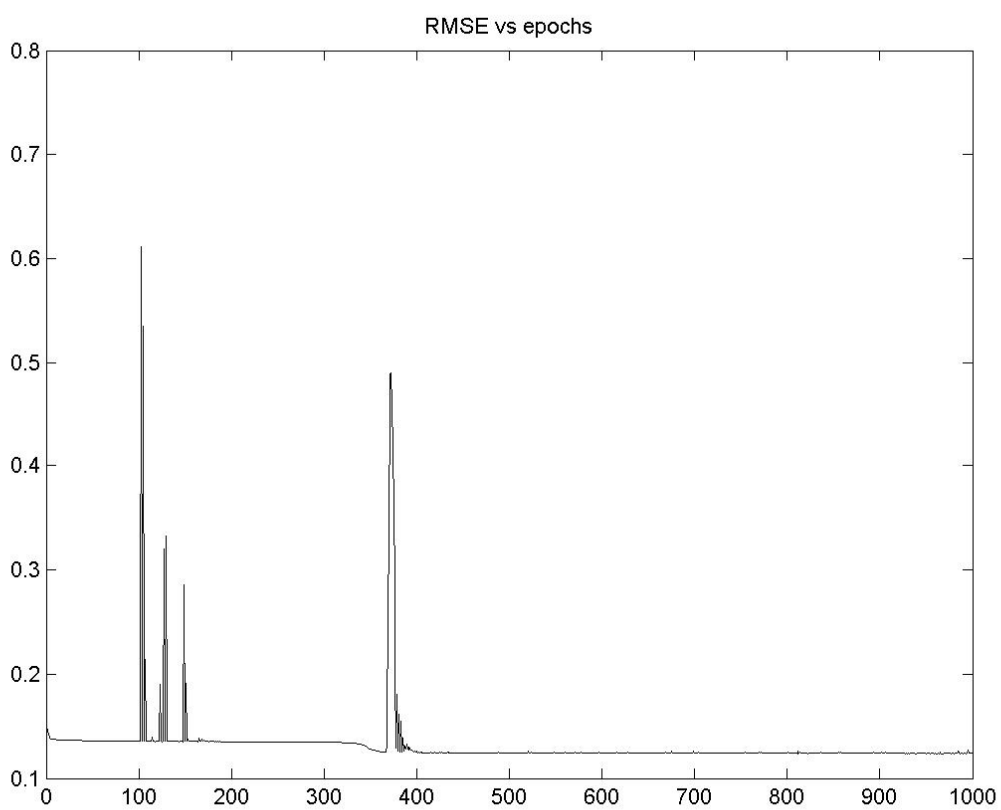
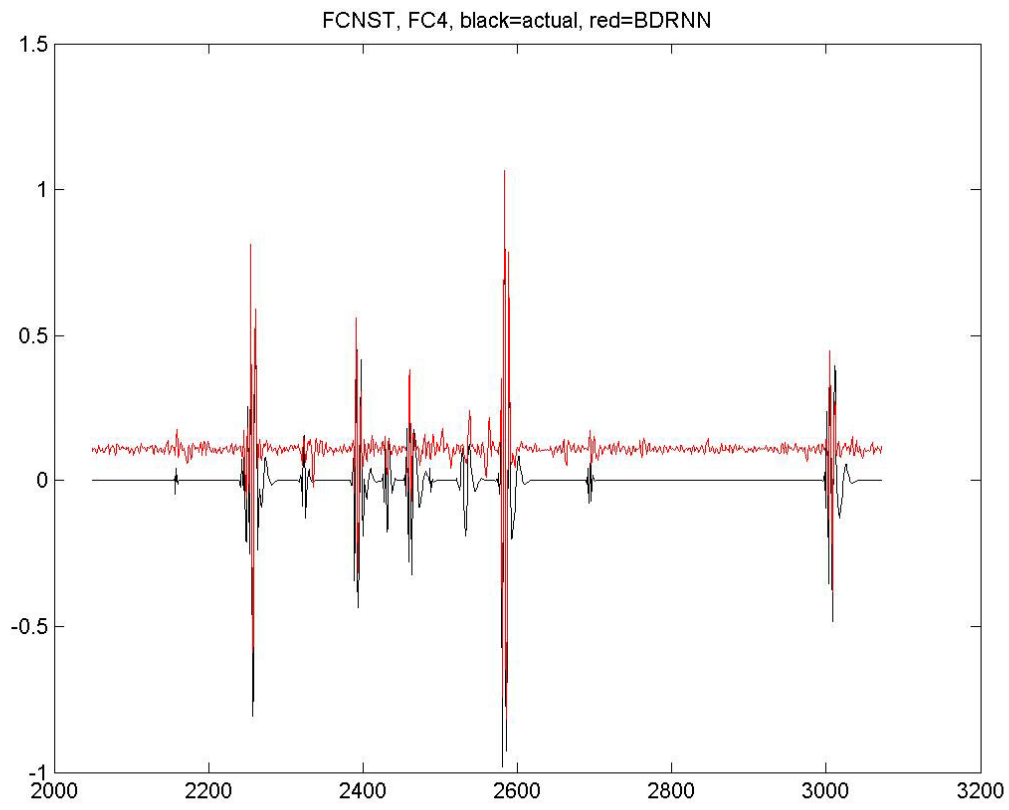
44^η Επανάληψη



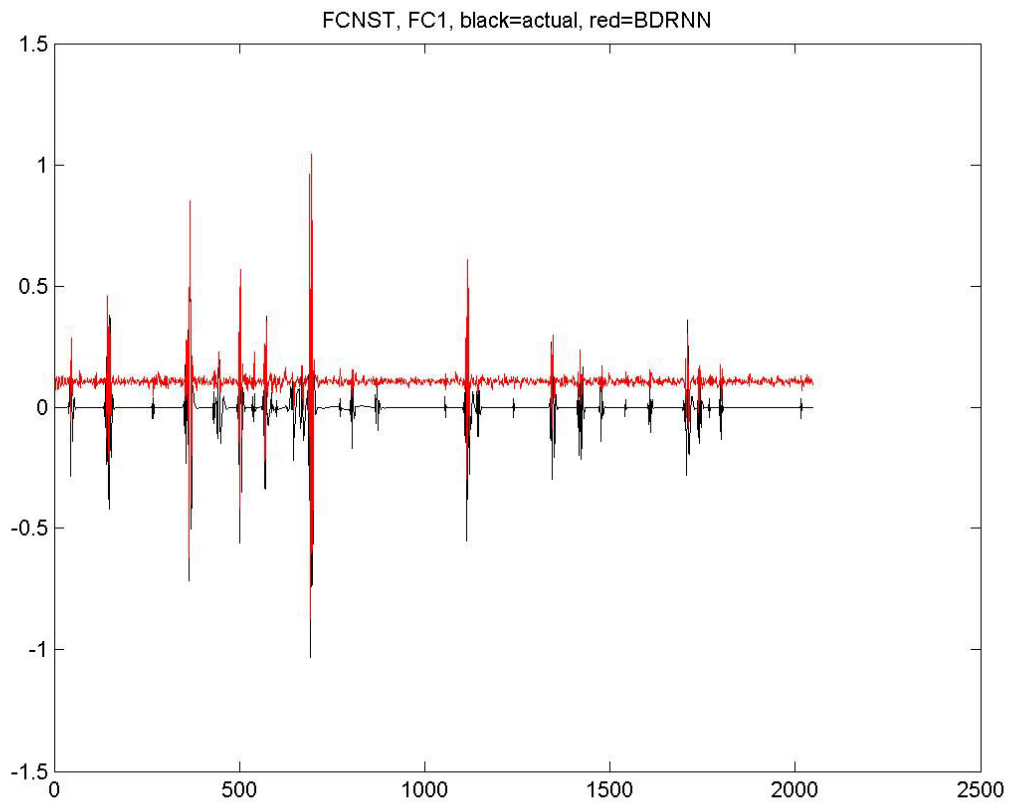
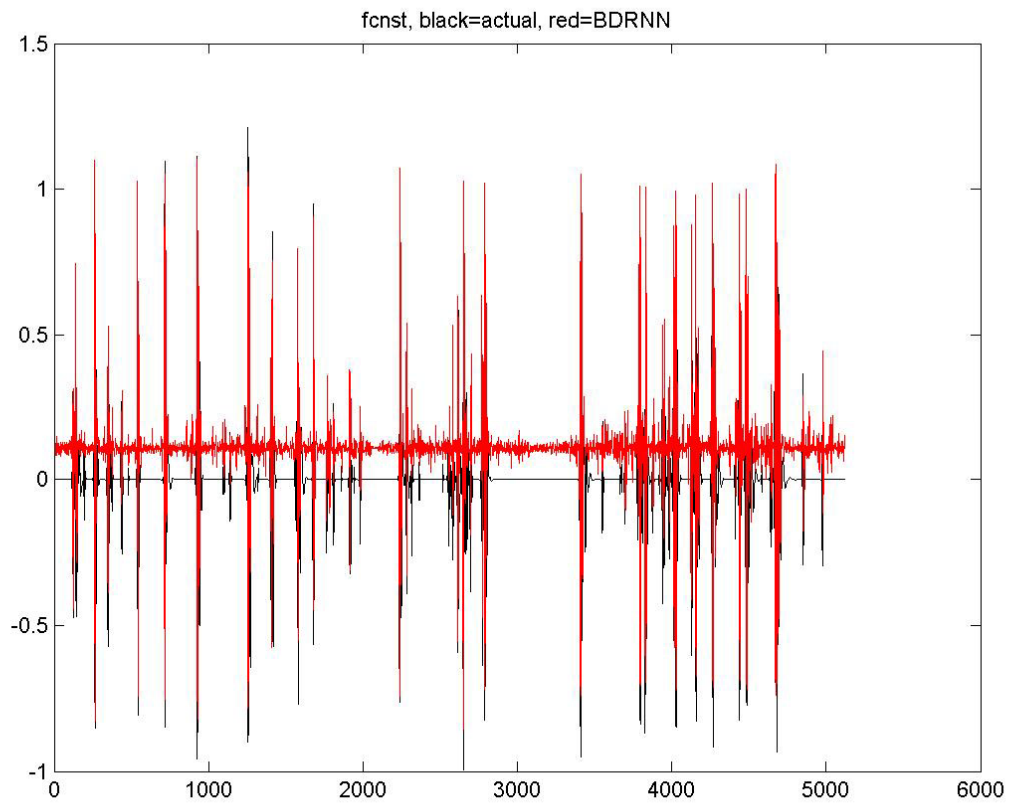


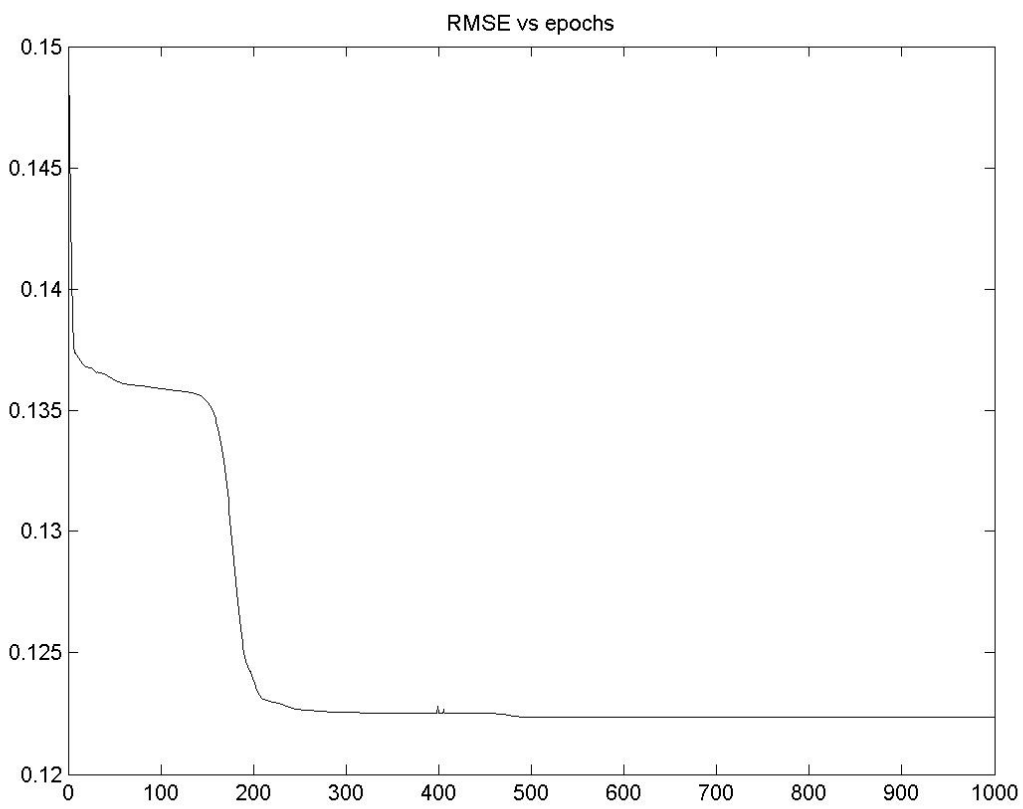
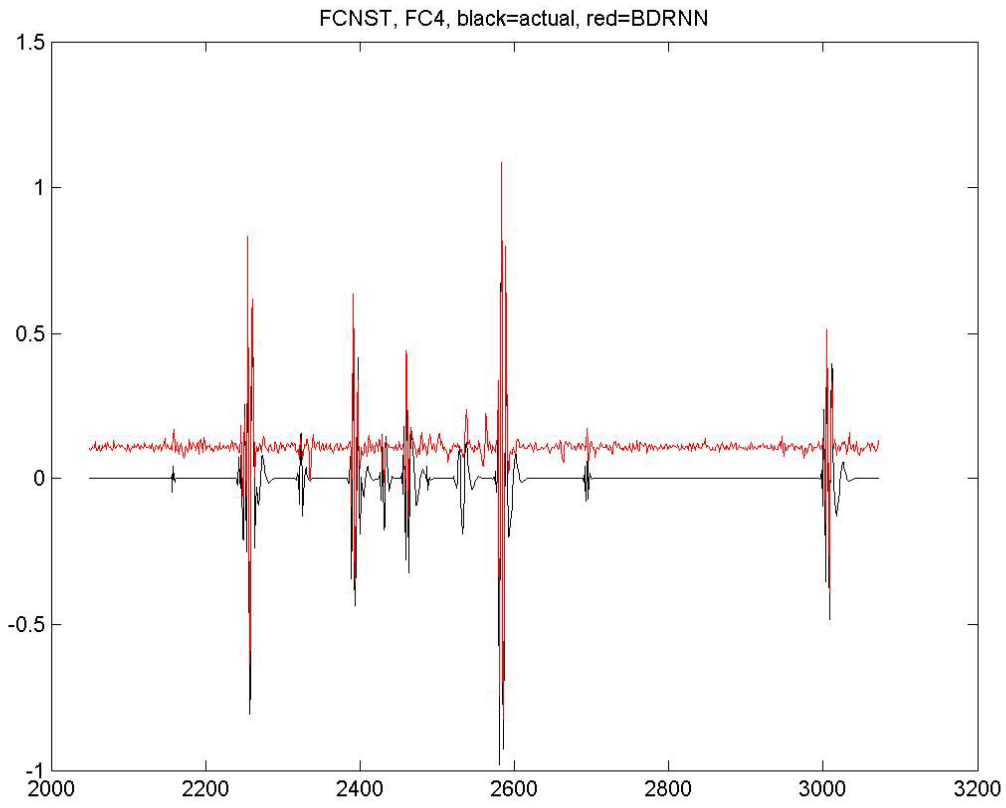
45^η Επανάληψη



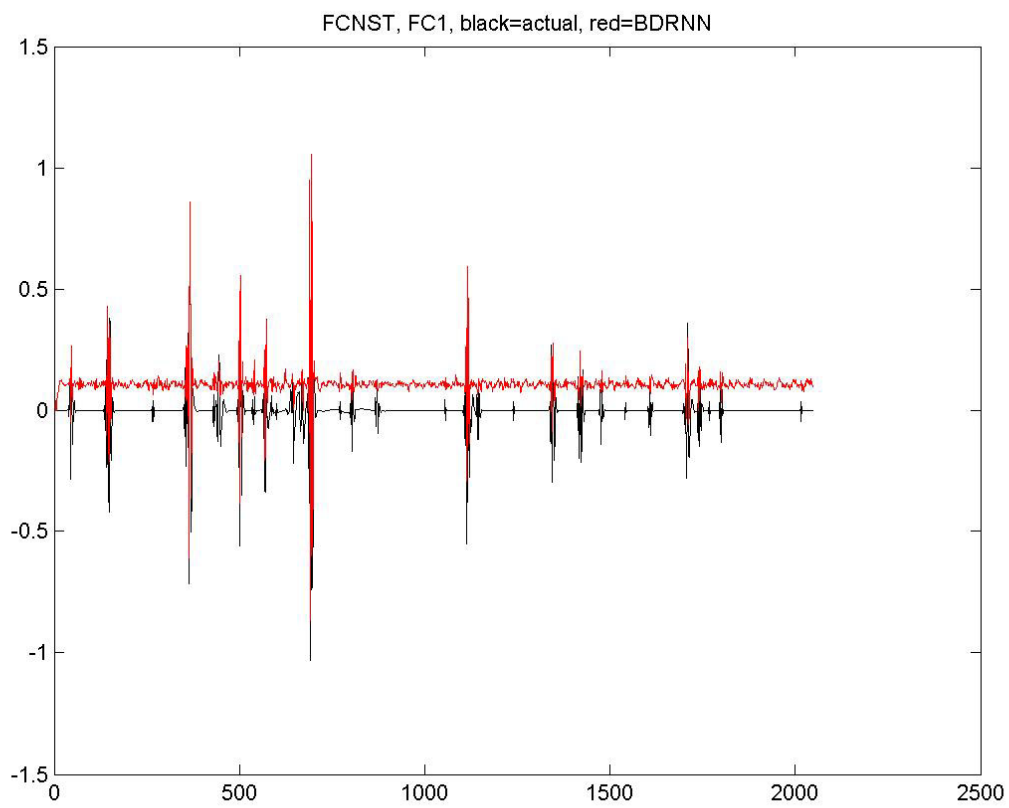
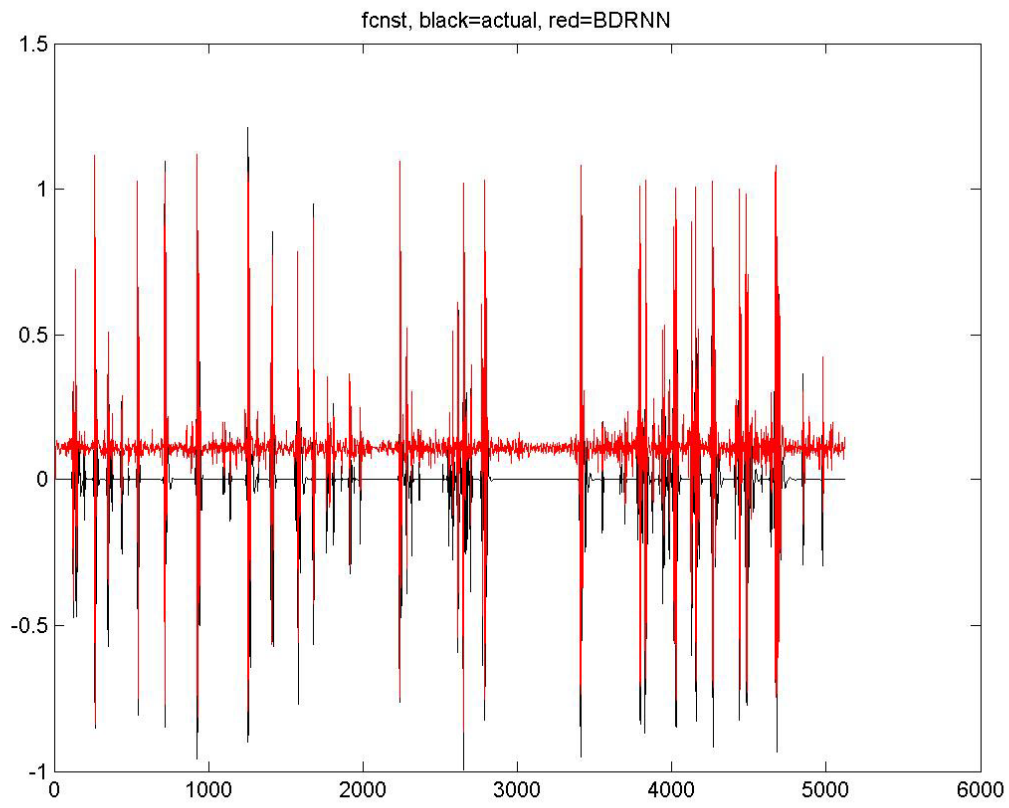


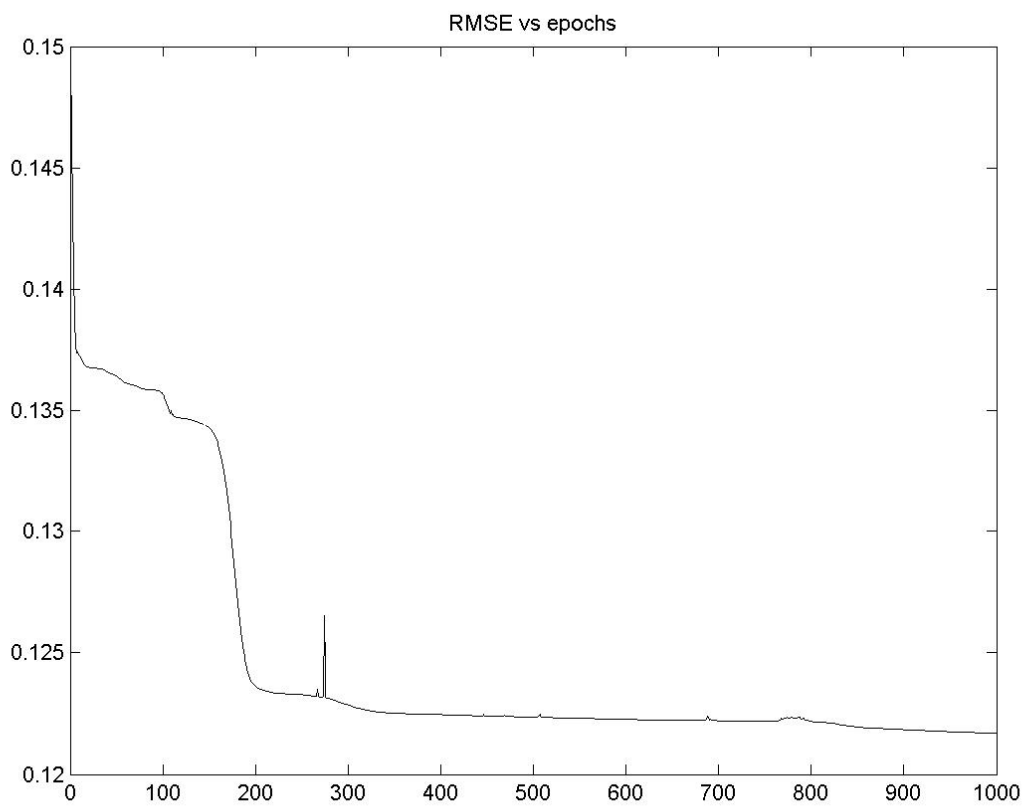
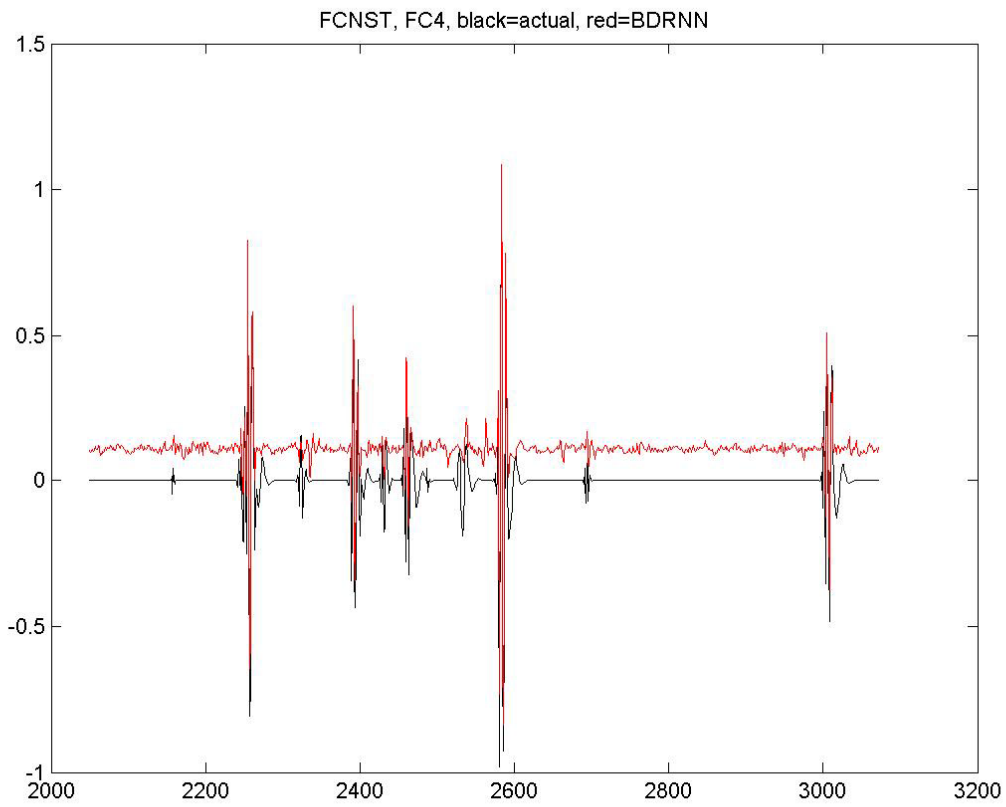
46^η Επανάληψη



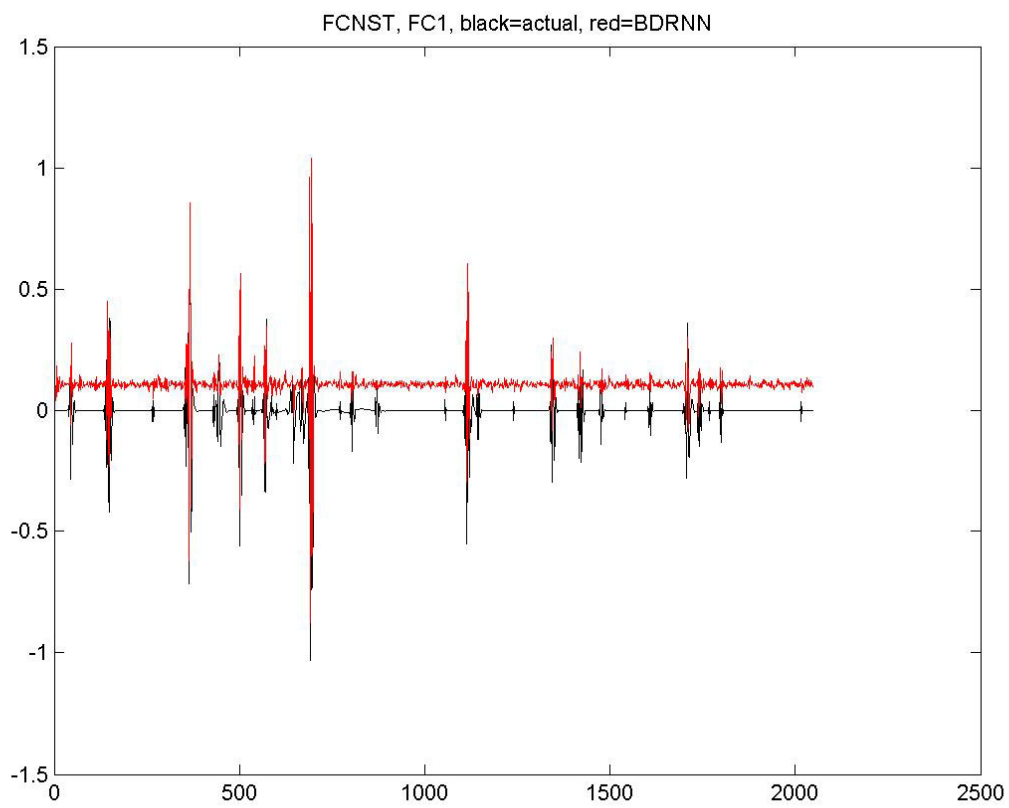
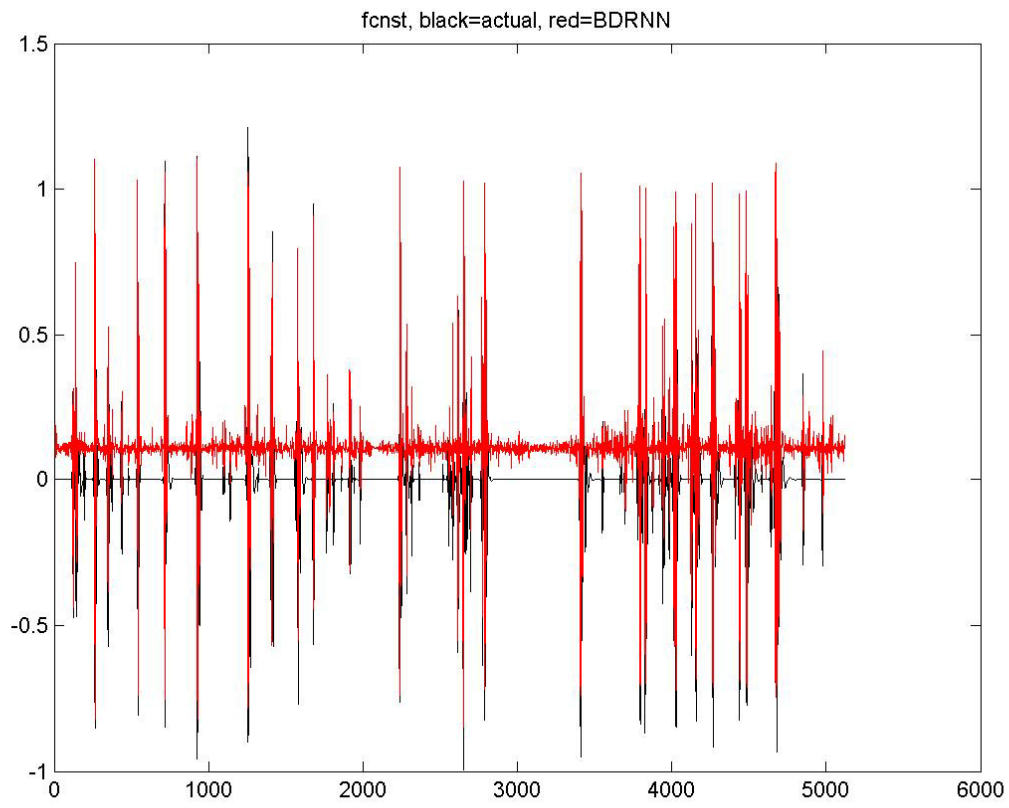


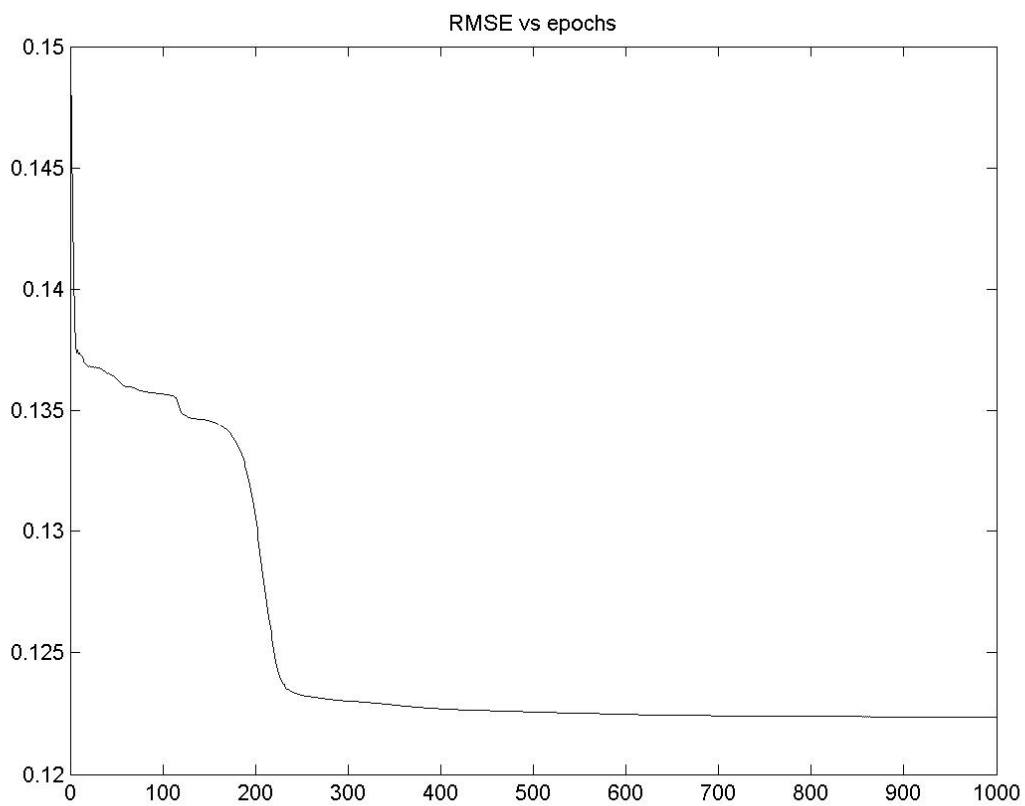
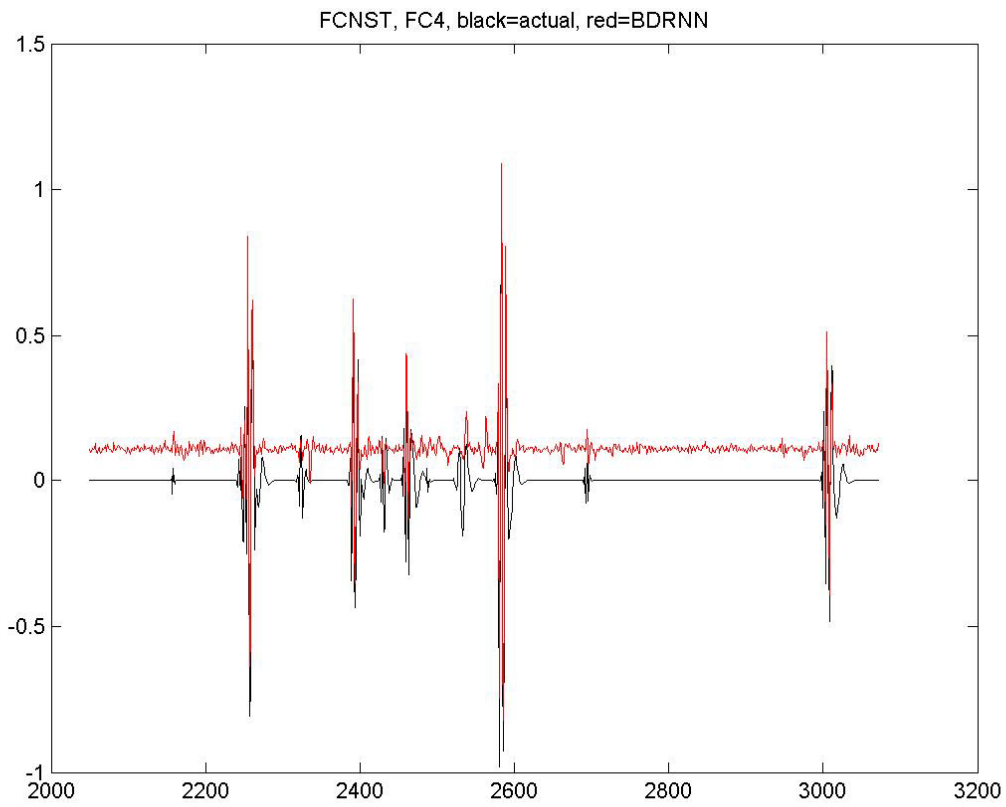
47^η Επανάληψη



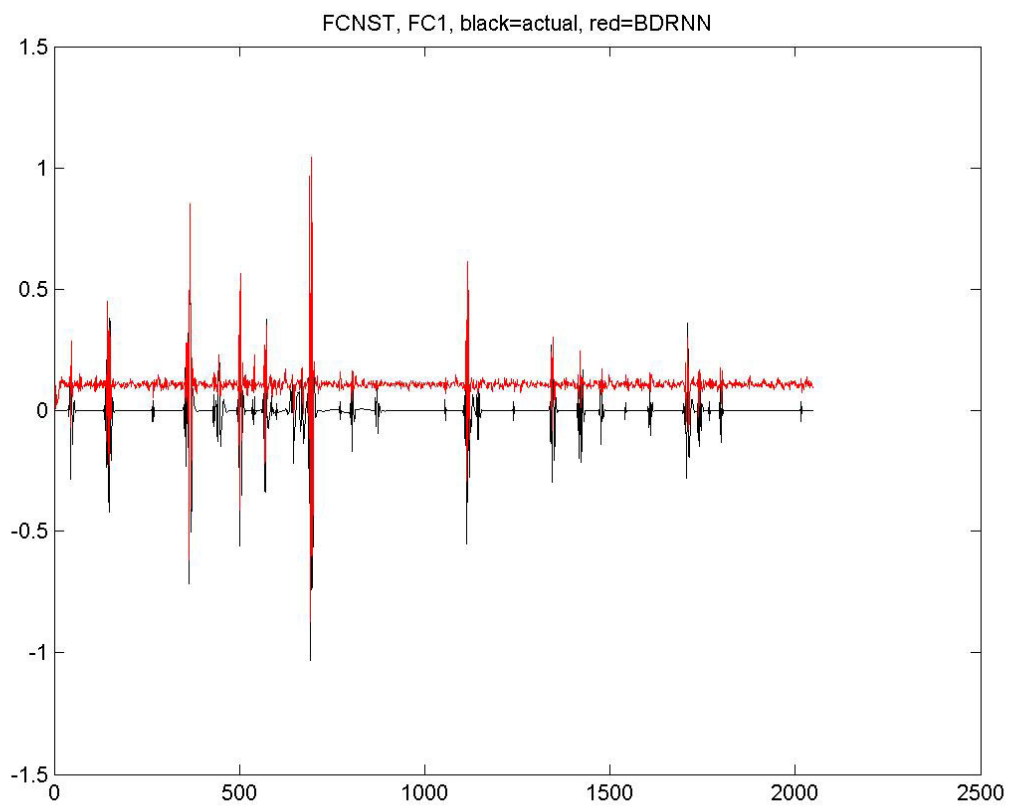


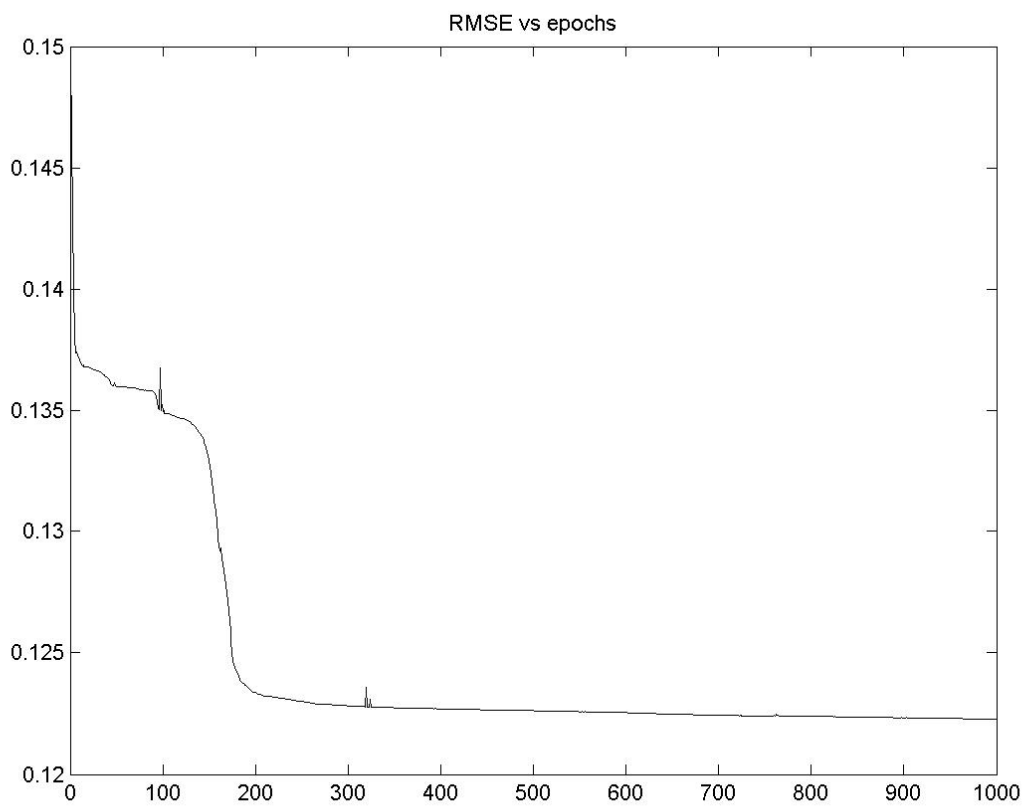
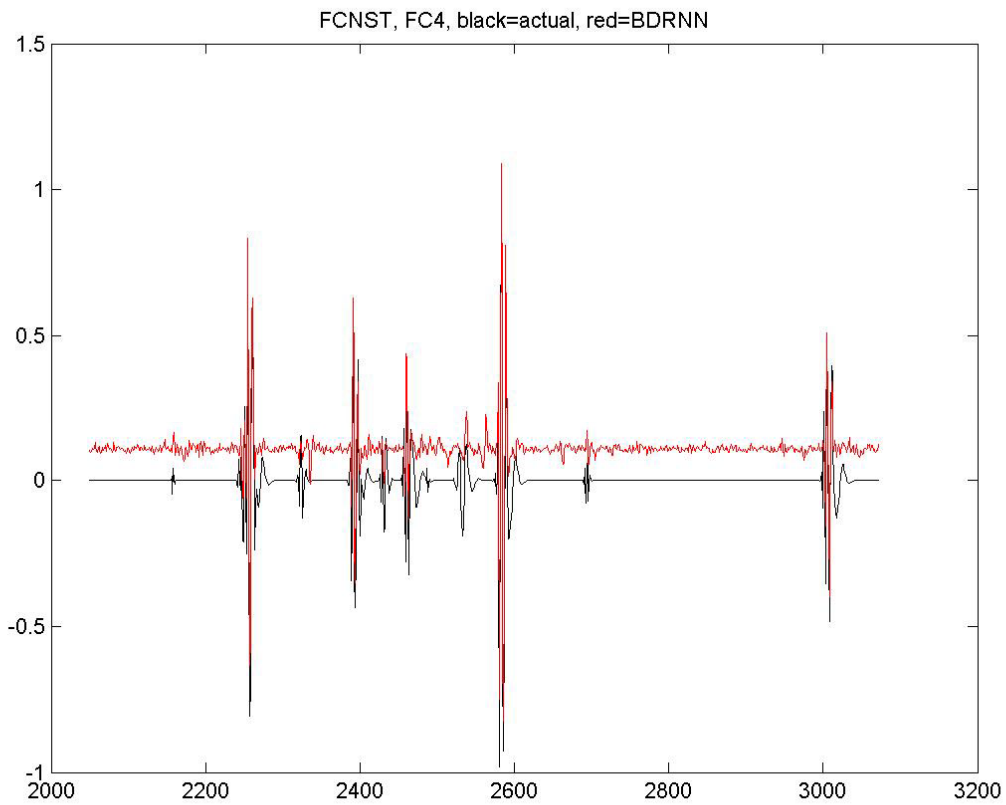
48^η Επανάληψη



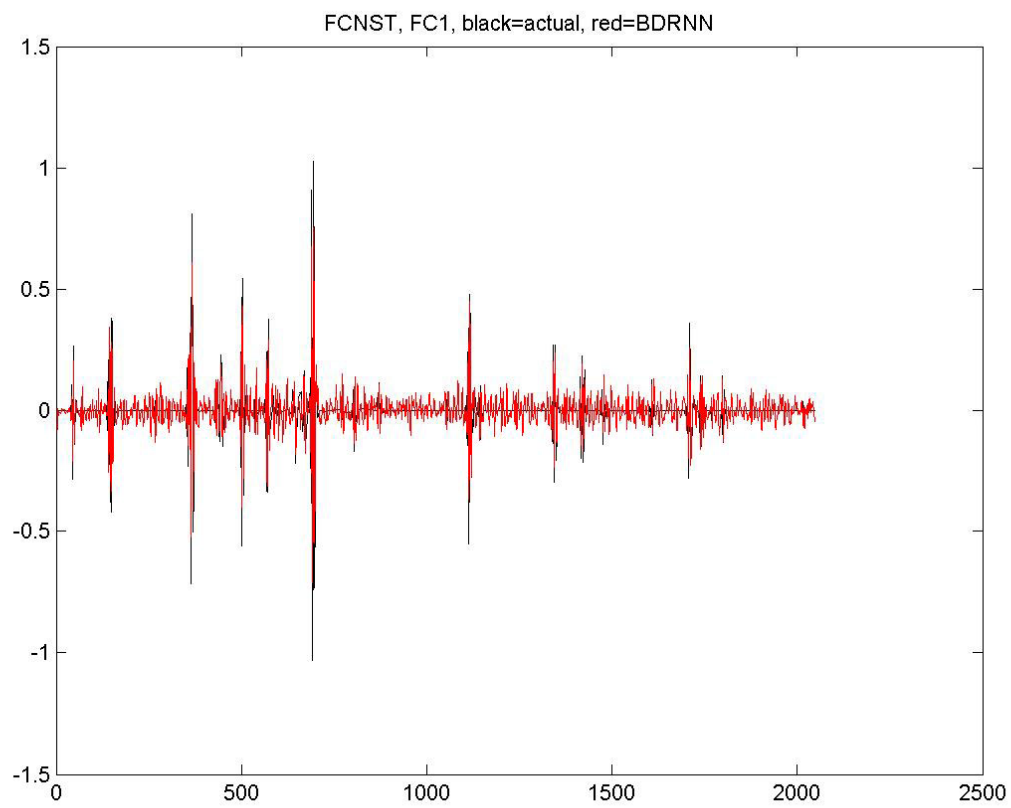
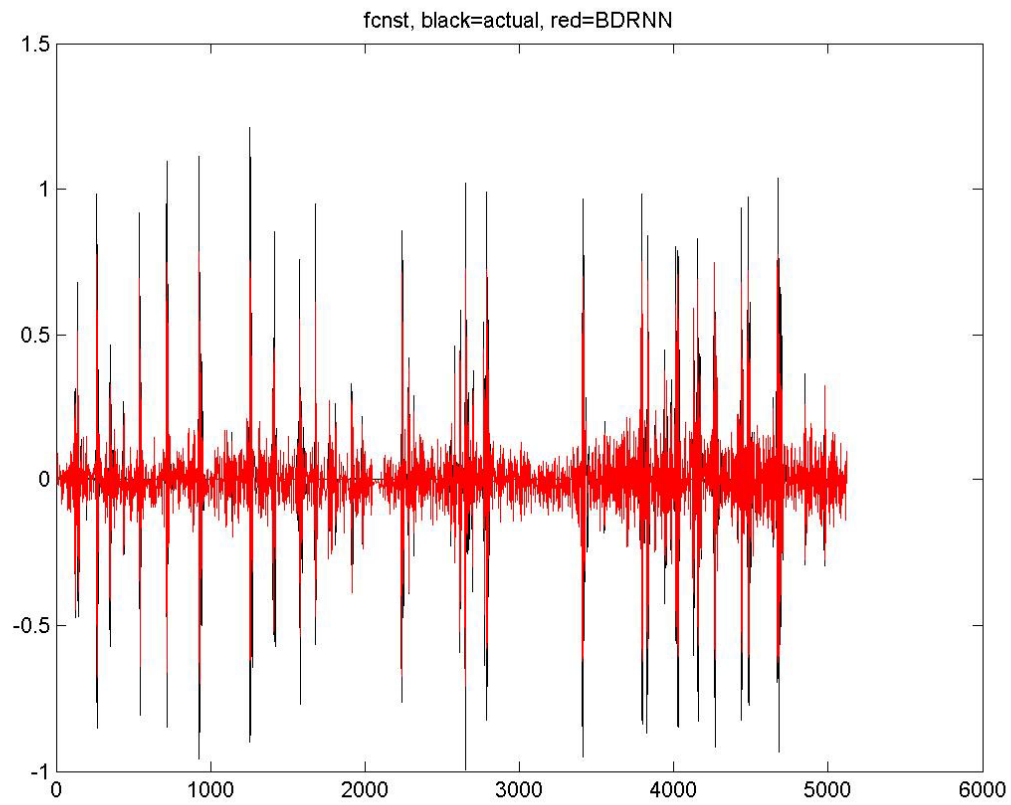


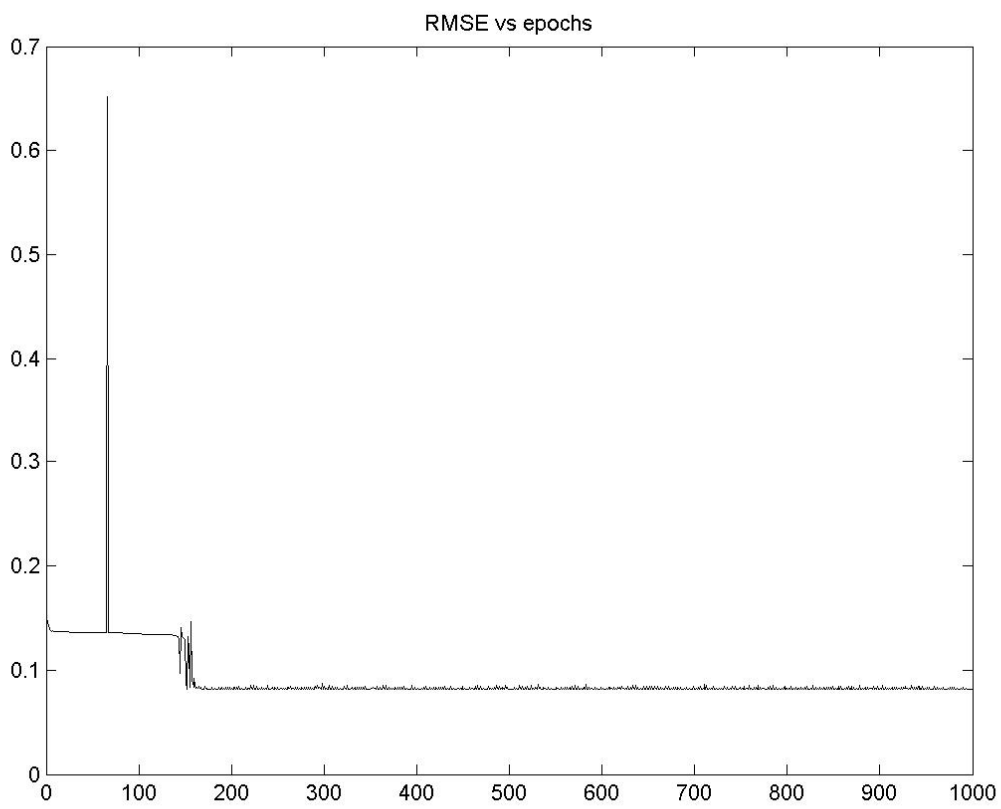
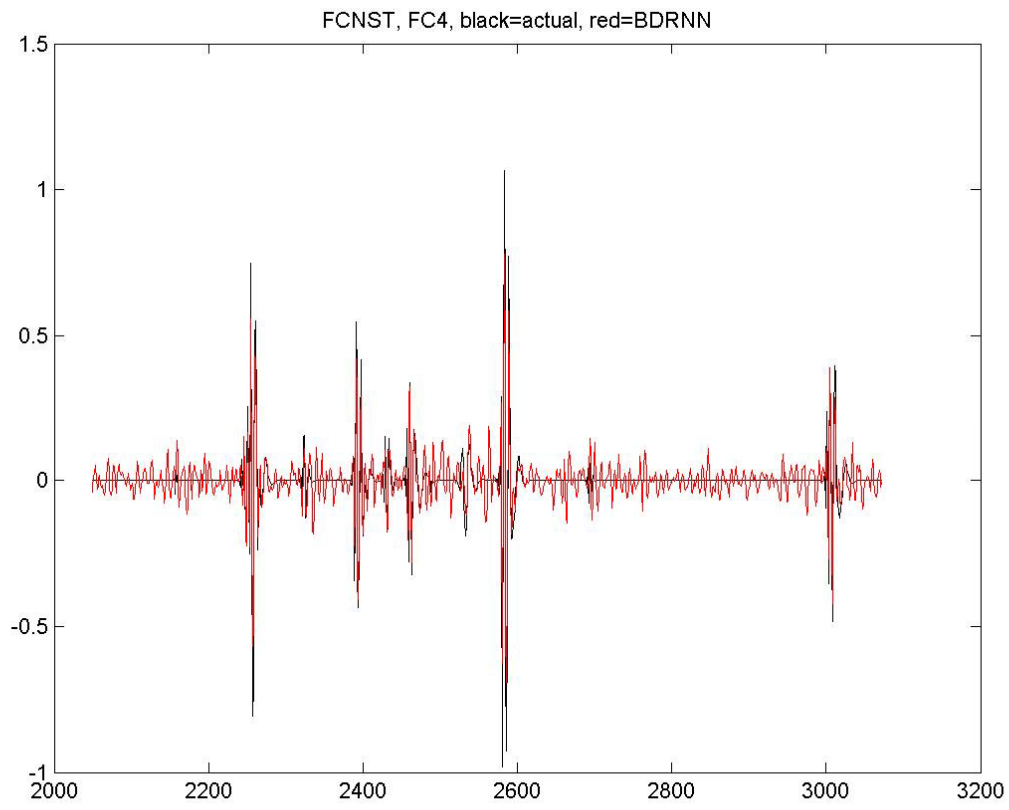
49^η Επανάληψη



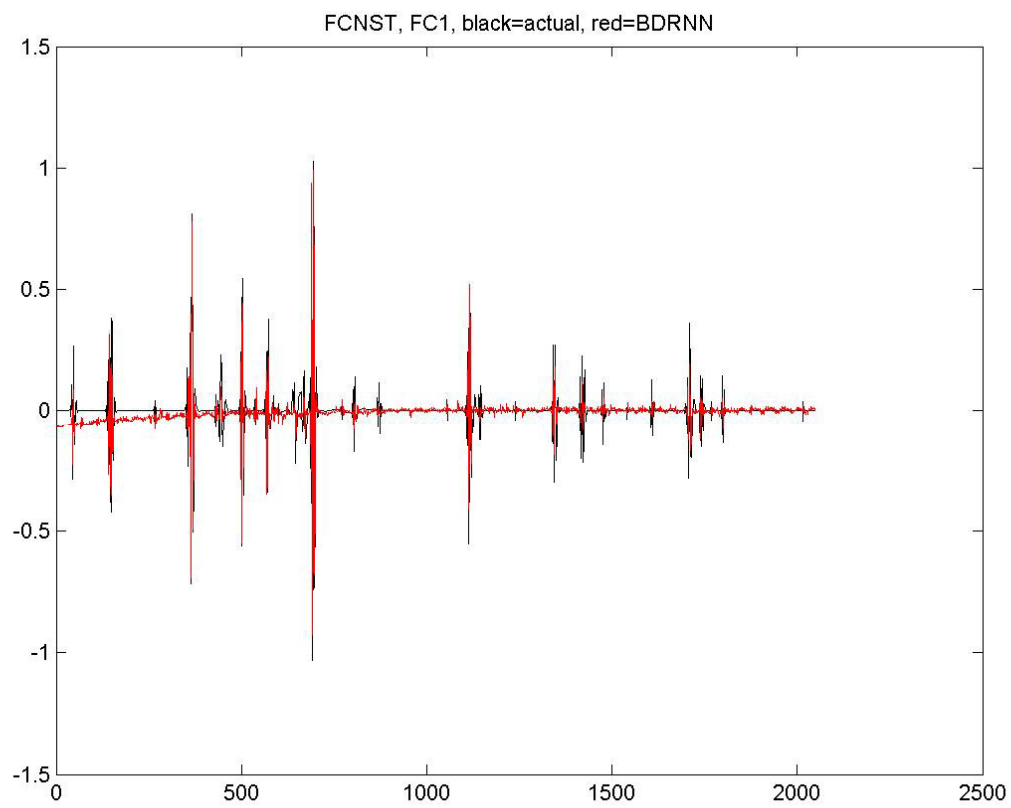
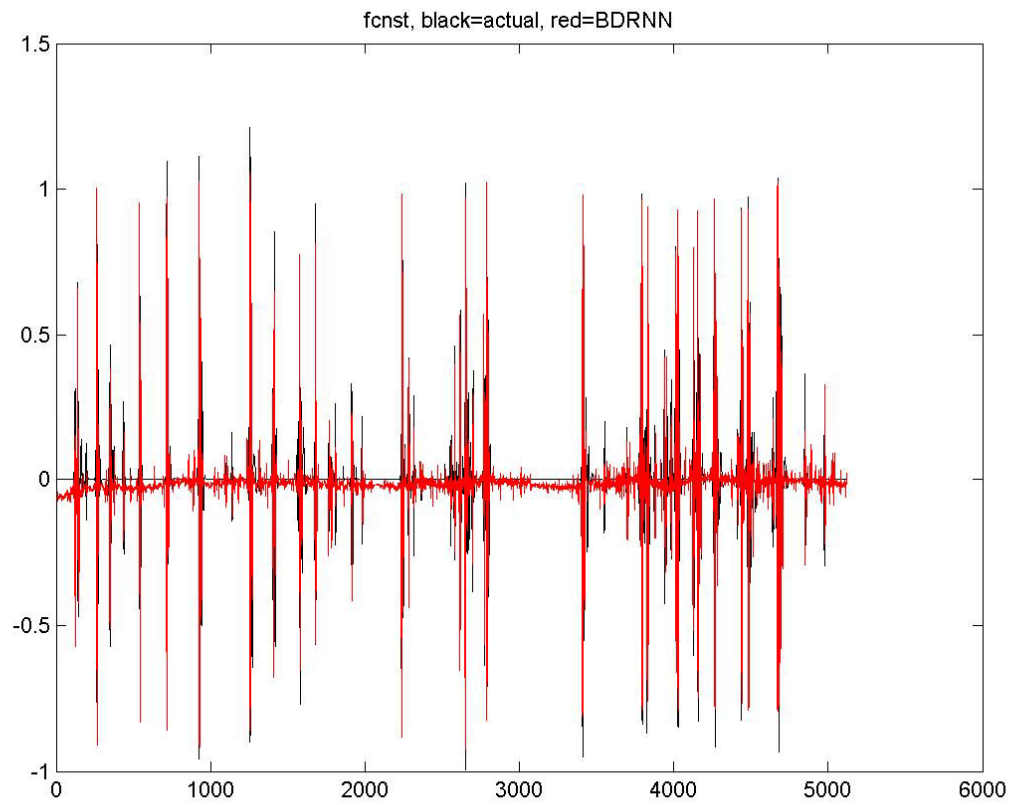


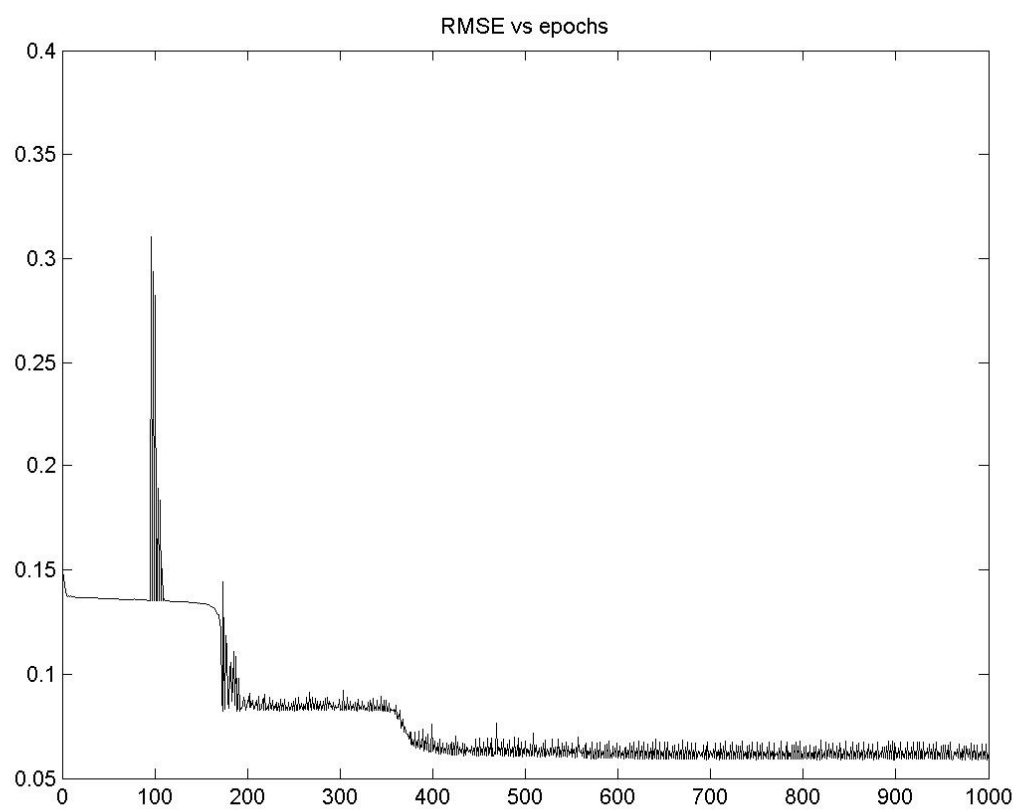
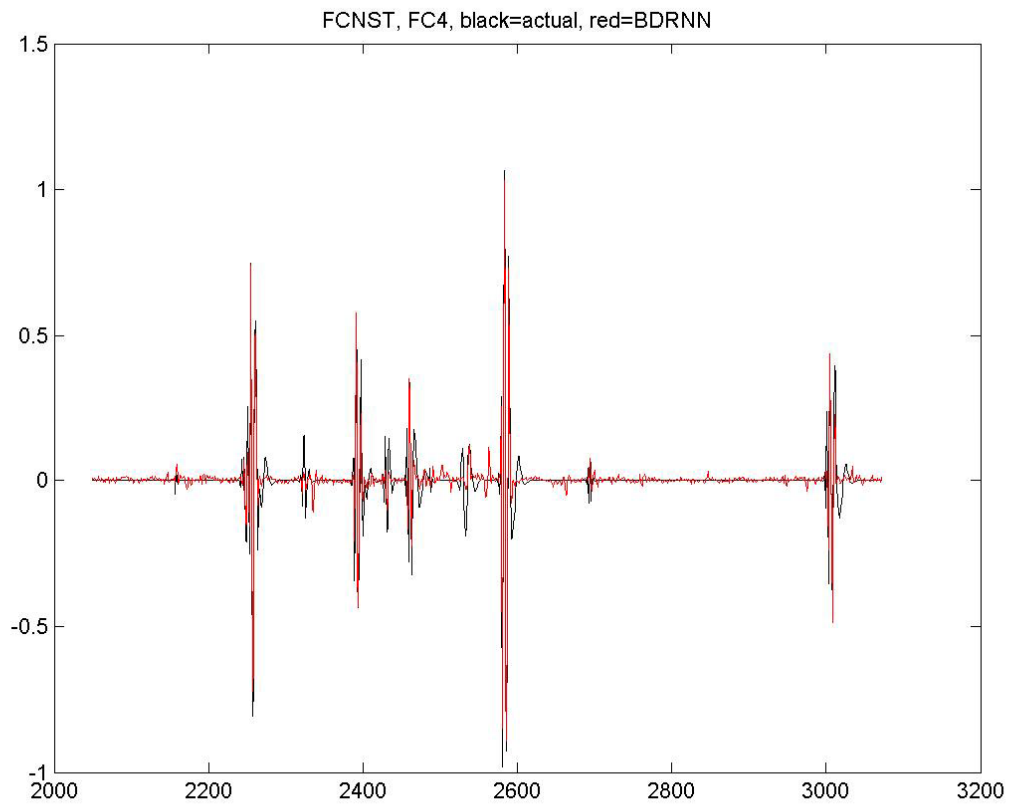
50^η Επανάληψη



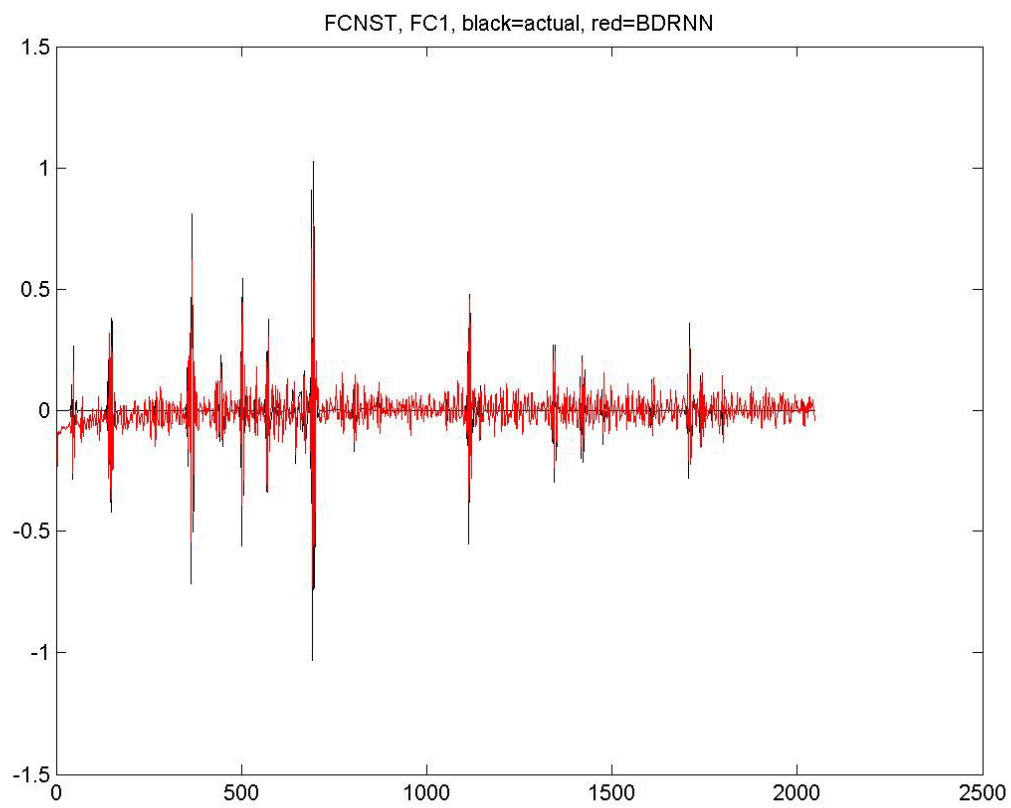
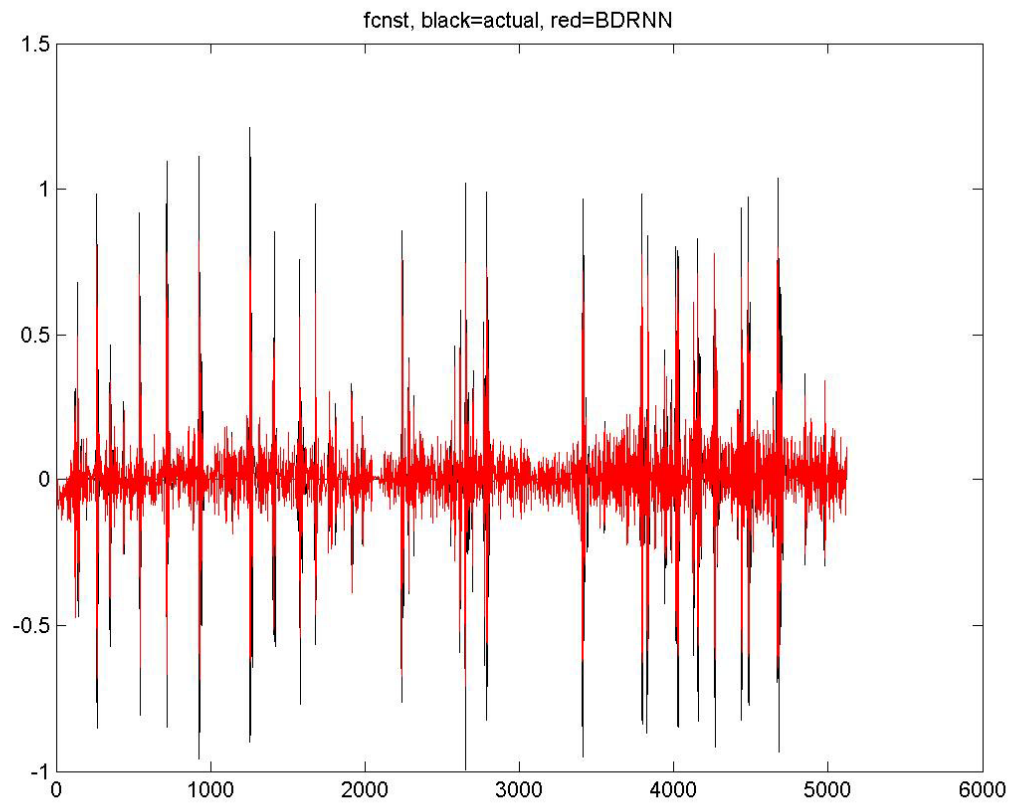


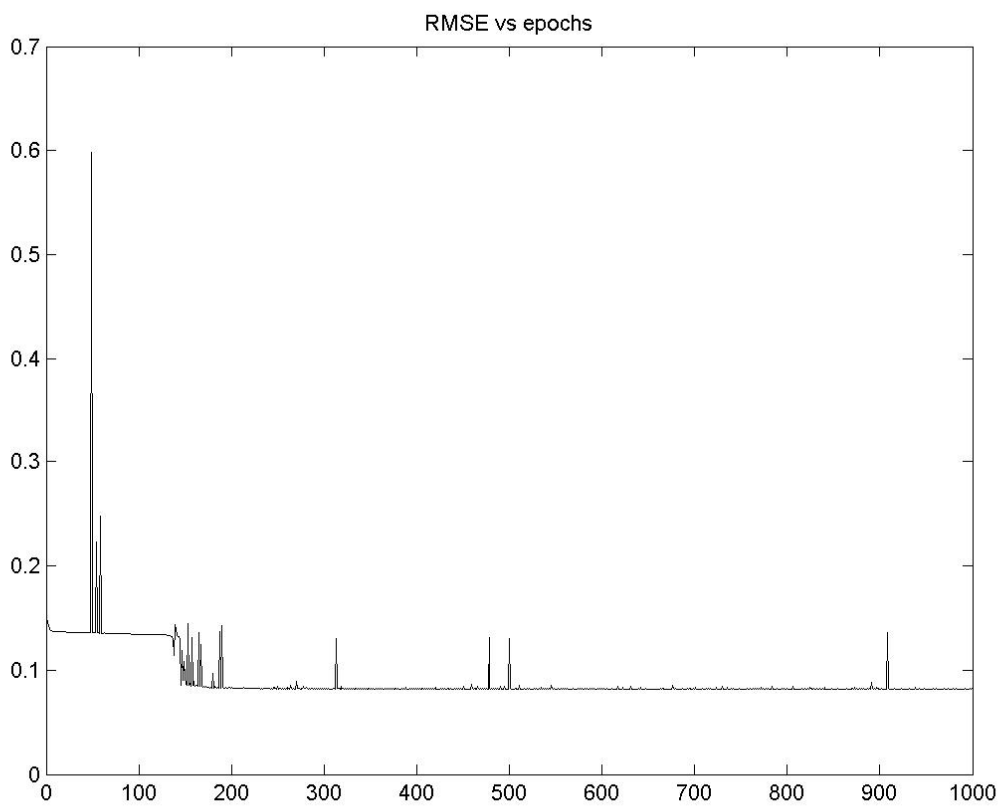
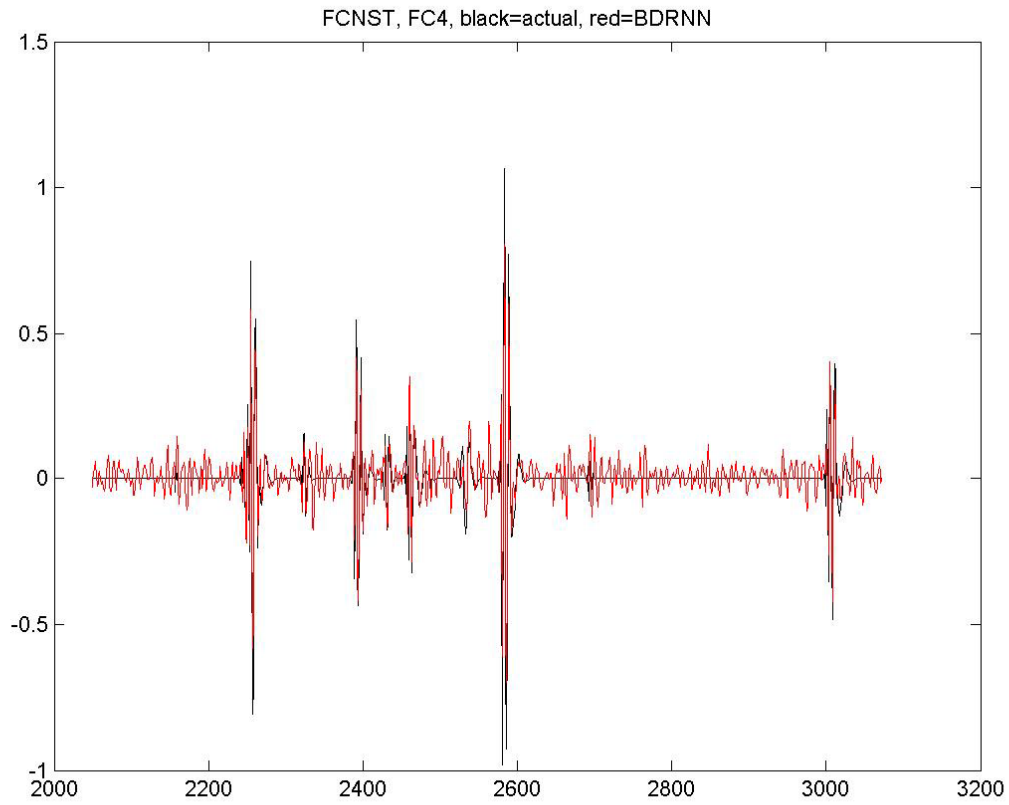
51^η Επανάληψη



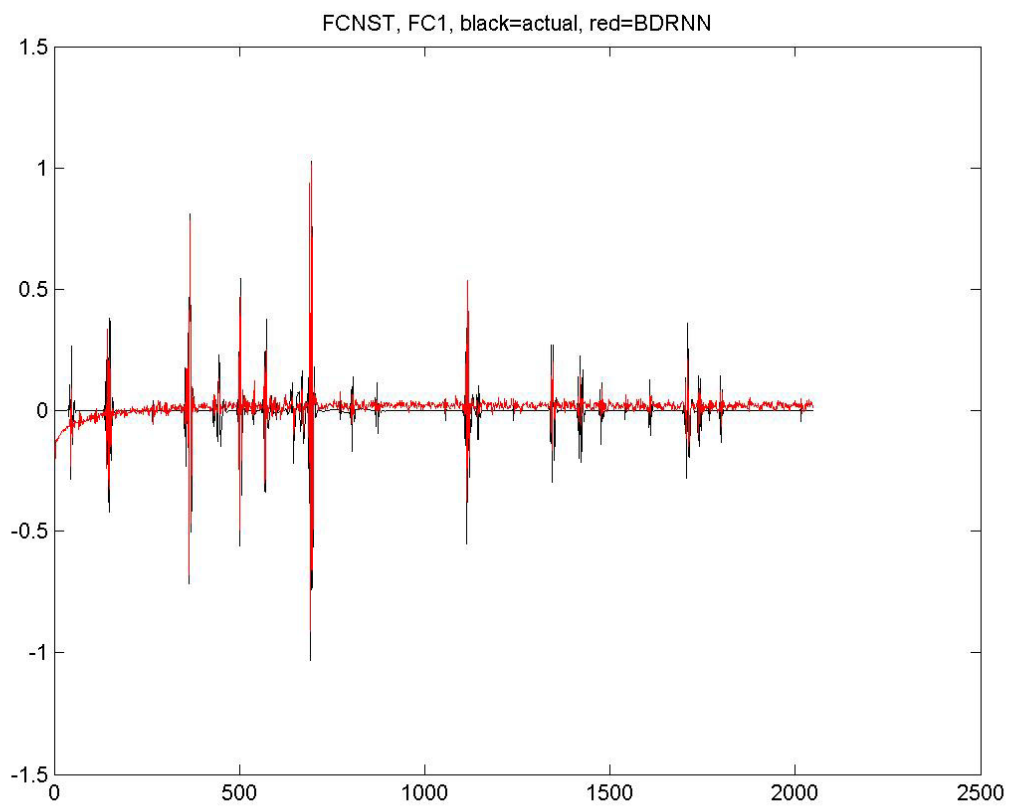
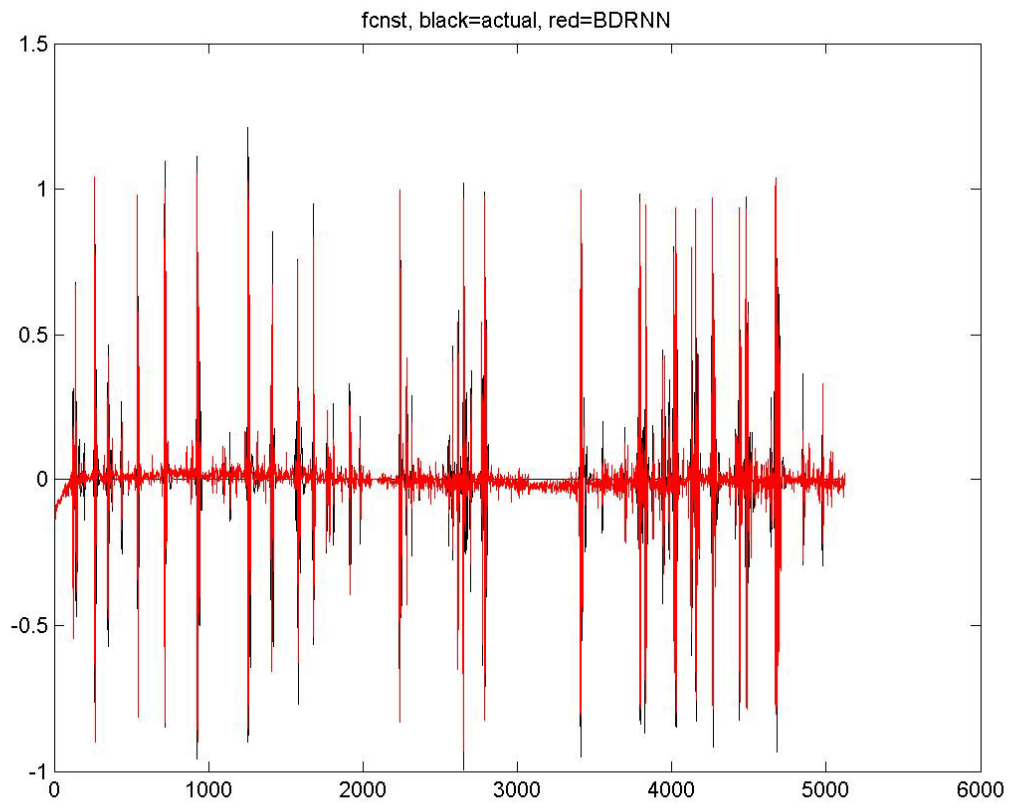


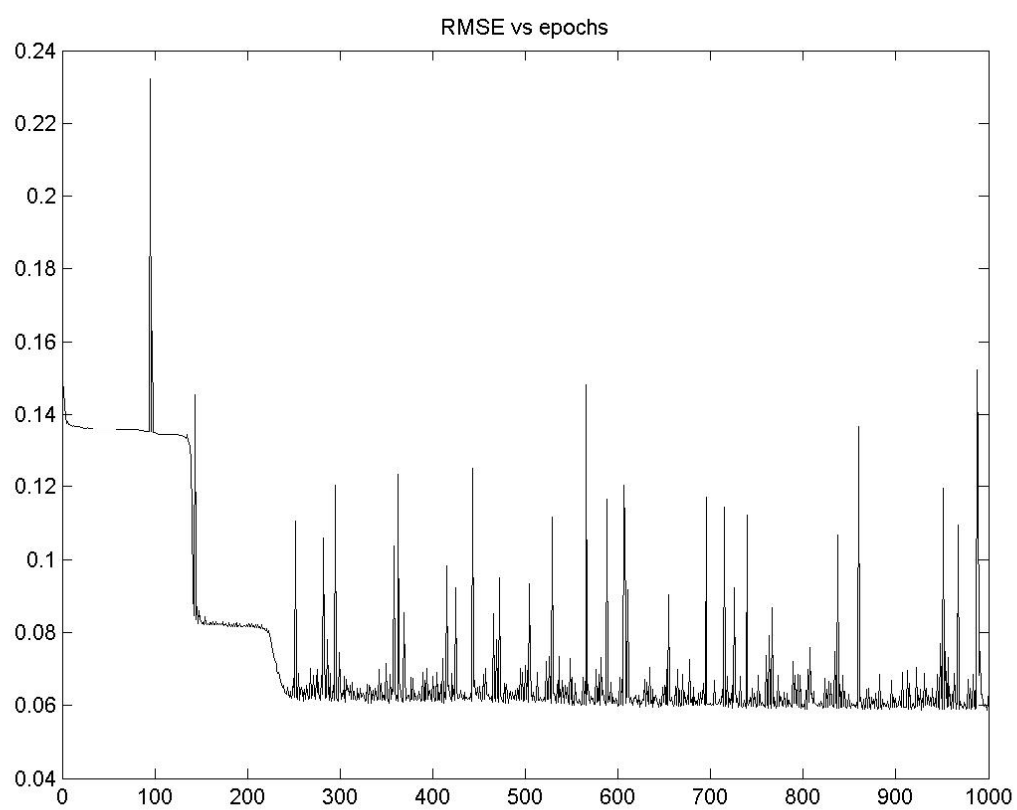
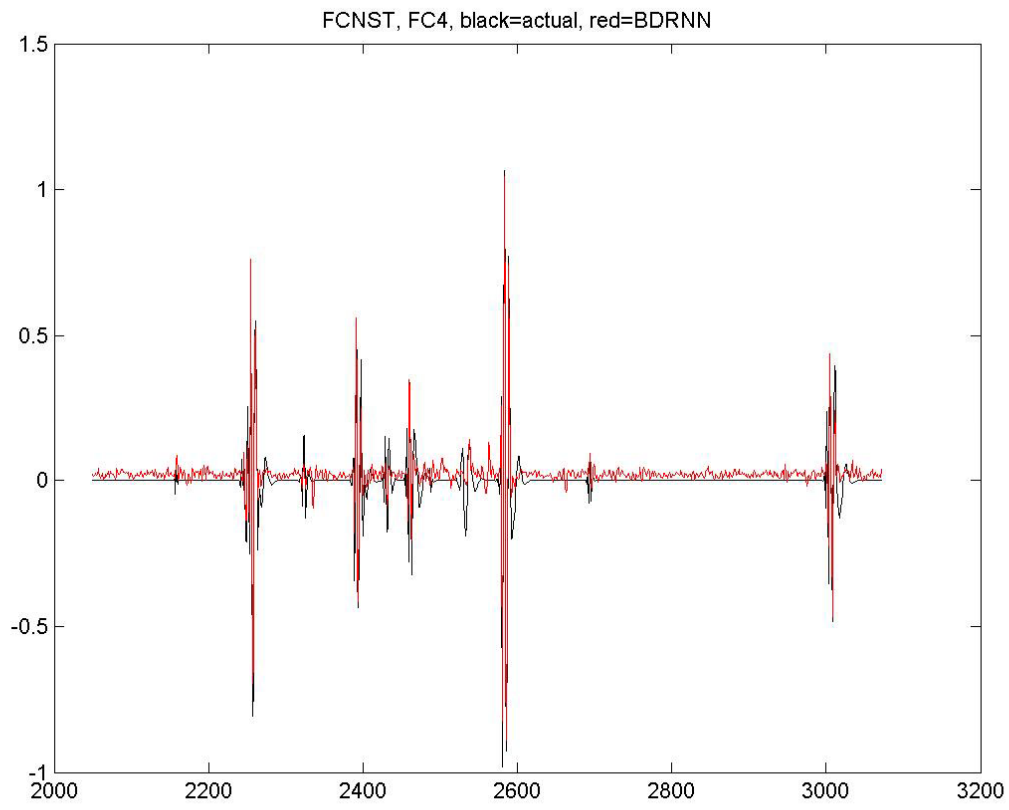
52^η Επανάληψη



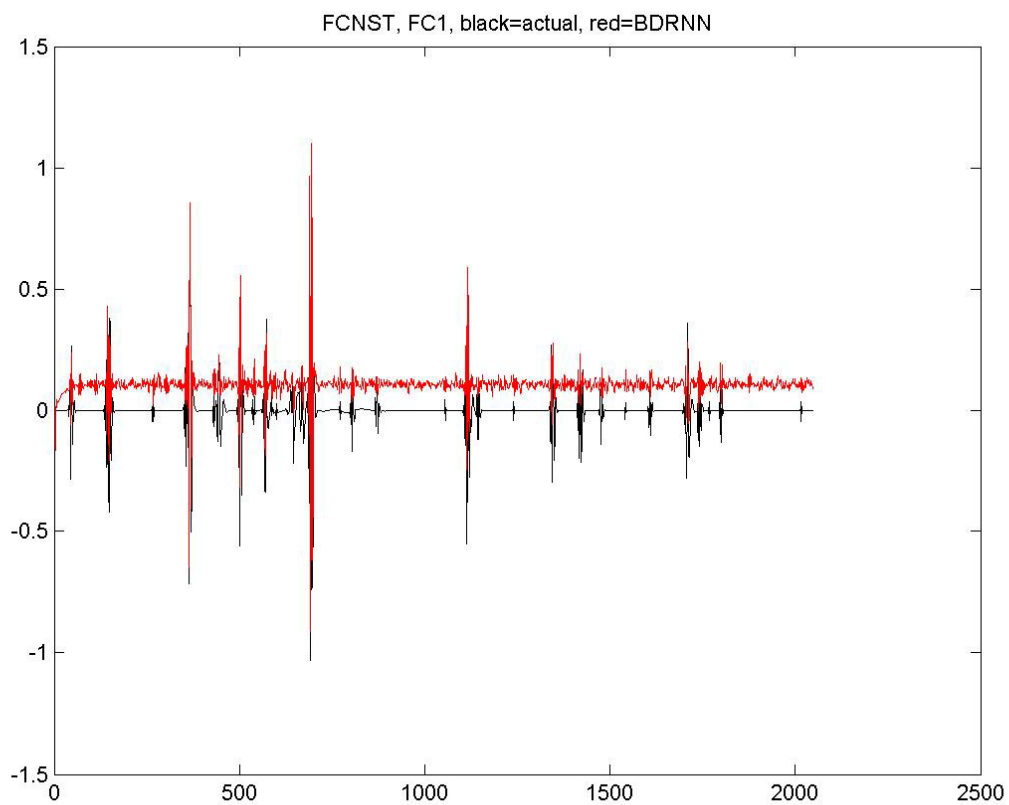
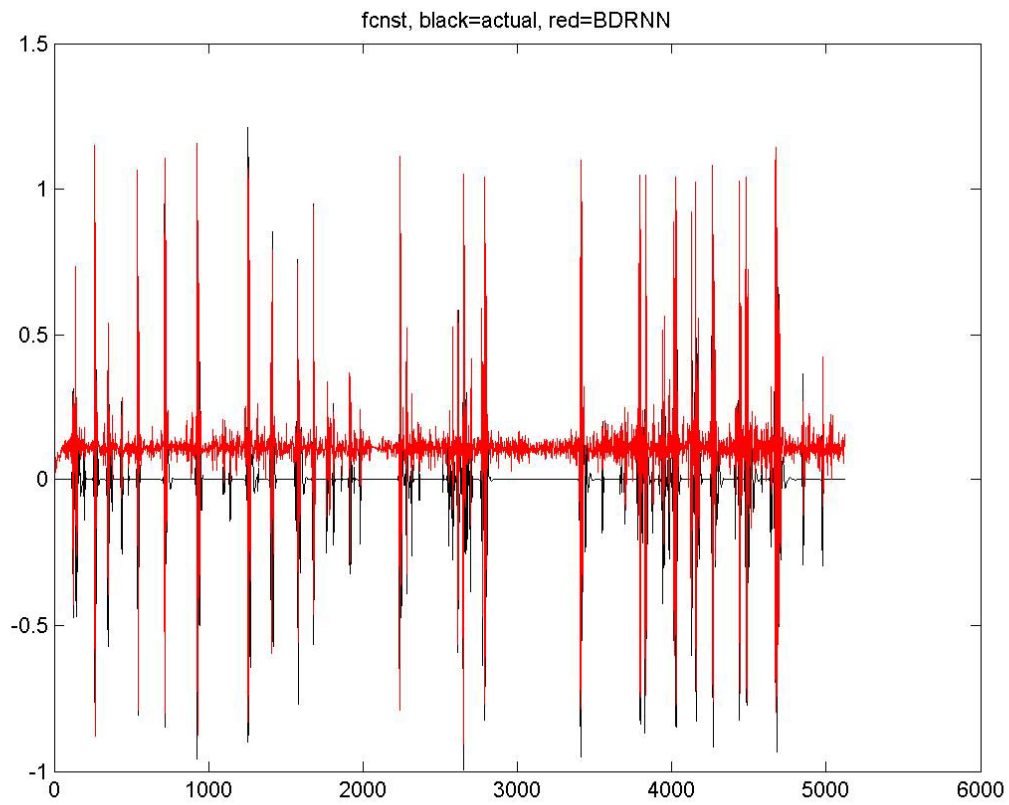


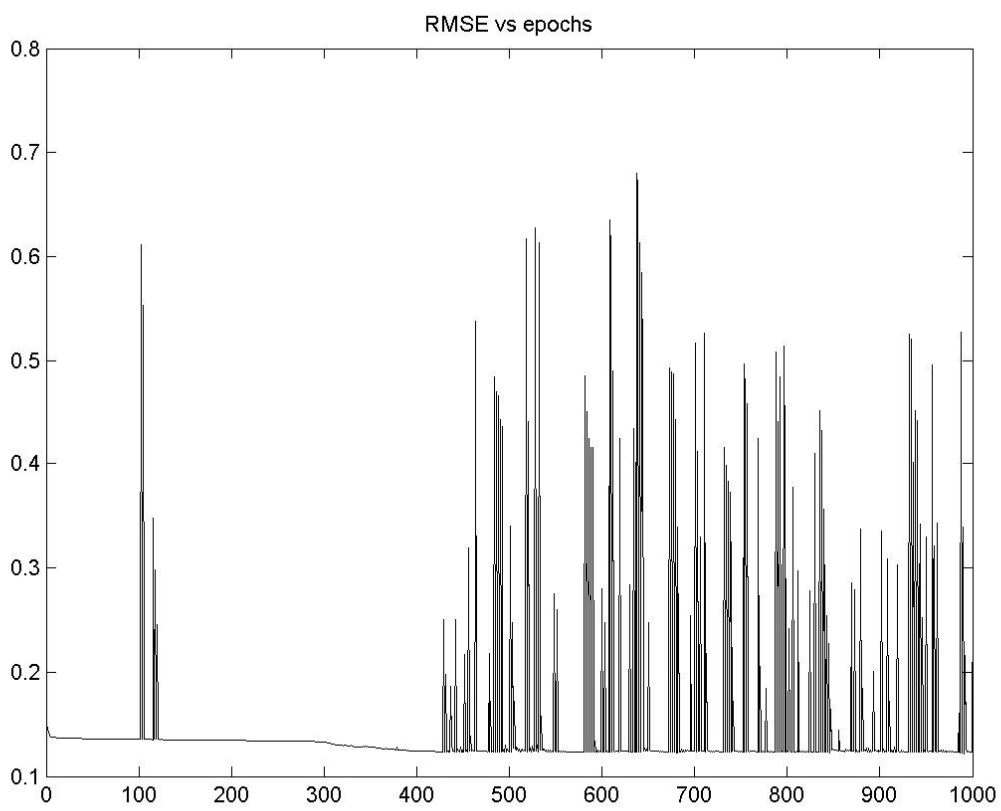
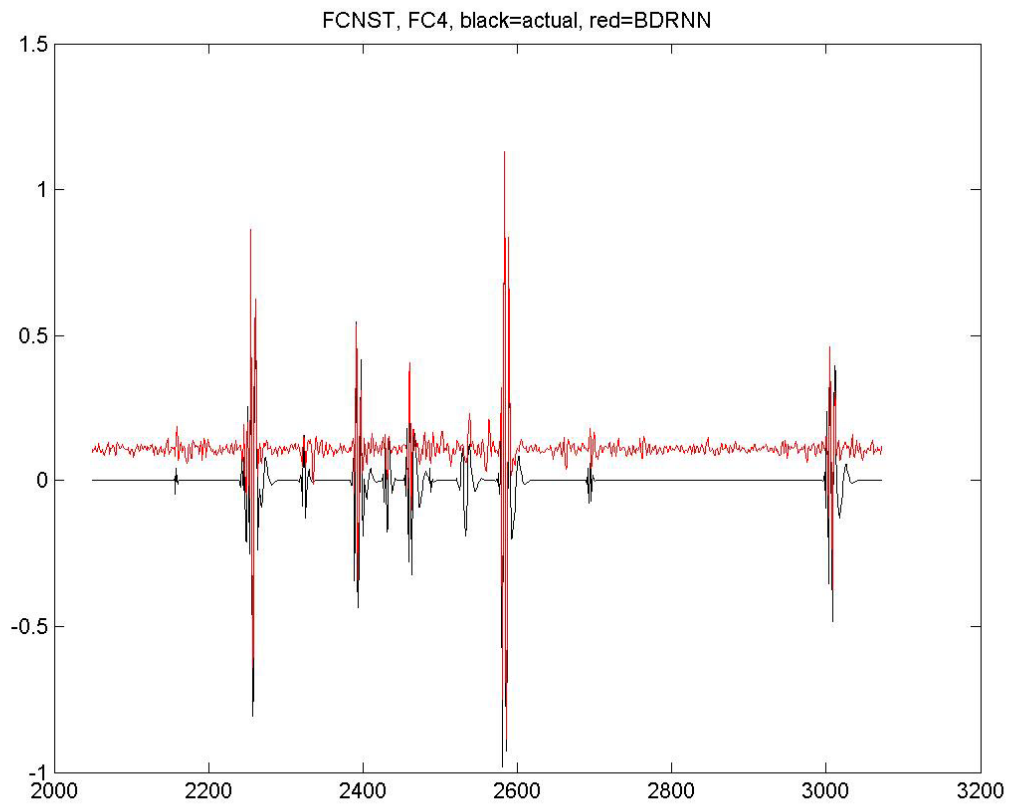
53^η Επανάληψη





54^η Επανάληψη





Πίνακας 4**Συνδυασμός παραμέτρων και αποτέλεσμα για την περίπτωση του πνευμονικού ήχου SQ**

Αριθμός Επανάληψης	n_hid	alpha stability	n2	ep_size	n+	n-	max_iter	RMSE (error)
1	12	6	2	5120	1,05	0,5	4000	0,010212
2	12	6	2	5120	1,05	0,7	4000	0,009186
3	12	6	2	5120	1,05	0,9	4000	0,021395
4	12	6	2	5120	1,15	0,5	4000	0,008711
5	12	6	2	5120	1,15	0,7	4000	0,007224
6	12	6	2	5120	1,15	0,9	4000	0,031606
7	12	6	2	5120	1,25	0,5	4000	0,009593
8	12	6	2	5120	1,25	0,7	4000	0,028214
9	12	6	2	5120	1,25	0,9	4000	0,007054
10	12	6	6	5120	1,05	0,5	4000	0,006761
11	12	6	6	5120	1,05	0,7	4000	0,012772
12	12	6	6	5120	1,05	0,9	4000	0,018250
13	12	6	6	5120	1,15	0,5	4000	0,008385
14	12	6	6	5120	1,15	0,7	4000	0,011700
15	12	6	6	5120	1,15	0,9	4000	0,025189
16	12	6	6	5120	1,25	0,5	4000	0,026983
17	12	6	6	5120	1,25	0,7	4000	0,020134
18	12	6	6	5120	1,25	0,9	4000	0,013626
19	18	9	2	5120	1,05	0,5	4000	0,005234
20	18	9	2	5120	1,05	0,7	4000	0,015608
21	18	9	2	5120	1,05	0,9	4000	0,024600
22	18	9	2	5120	1,15	0,5	4000	0,003759
23	18	9	2	5120	1,15	0,7	4000	0,005311
24	18	9	2	5120	1,15	0,9	4000	0,015950
25	18	9	2	5120	1,25	0,5	4000	0,004102
26	18	9	2	5120	1,25	0,7	4000	0,014945
27	18	9	2	5120	1,25	0,9	4000	0,008299
28	18	9	6	5120	1,05	0,5	4000	0,039937
29	18	9	6	5120	1,05	0,7	4000	0,020625
30	18	9	6	5120	1,05	0,9	4000	0,004569
31	18	9	6	5120	1,15	0,5	4000	0,006672
32	18	9	6	5120	1,15	0,7	4000	0,009317
33	18	9	6	5120	1,15	0,9	4000	0,007412
34	18	9	6	5120	1,25	0,5	4000	0,012413
35	18	9	6	5120	1,25	0,7	4000	0,004164
36	18	9	6	5120	1,25	0,9	4000	0,004653
37	24	12	2	5120	1,05	0,5	4000	0,043563
38	24	12	2	5120	1,05	0,7	4000	0,008339
39	24	12	2	5120	1,05	0,9	4000	0,017654
40	24	12	2	5120	1,15	0,5	4000	0,015738
41	24	12	2	5120	1,15	0,7	4000	0,004757
42	24	12	2	5120	1,15	0,9	4000	0,006099
43	24	12	2	5120	1,25	0,5	4000	0,017995
44	24	12	2	5120	1,25	0,7	4000	0,007665
45	24	12	2	5120	1,25	0,9	4000	0,005820
46	24	12	6	5120	1,05	0,5	4000	0,030021
47	24	12	6	5120	1,05	0,7	4000	0,005010
48	24	12	6	5120	1,05	0,9	4000	0,020178
49	24	12	6	5120	1,15	0,5	4000	0,004782

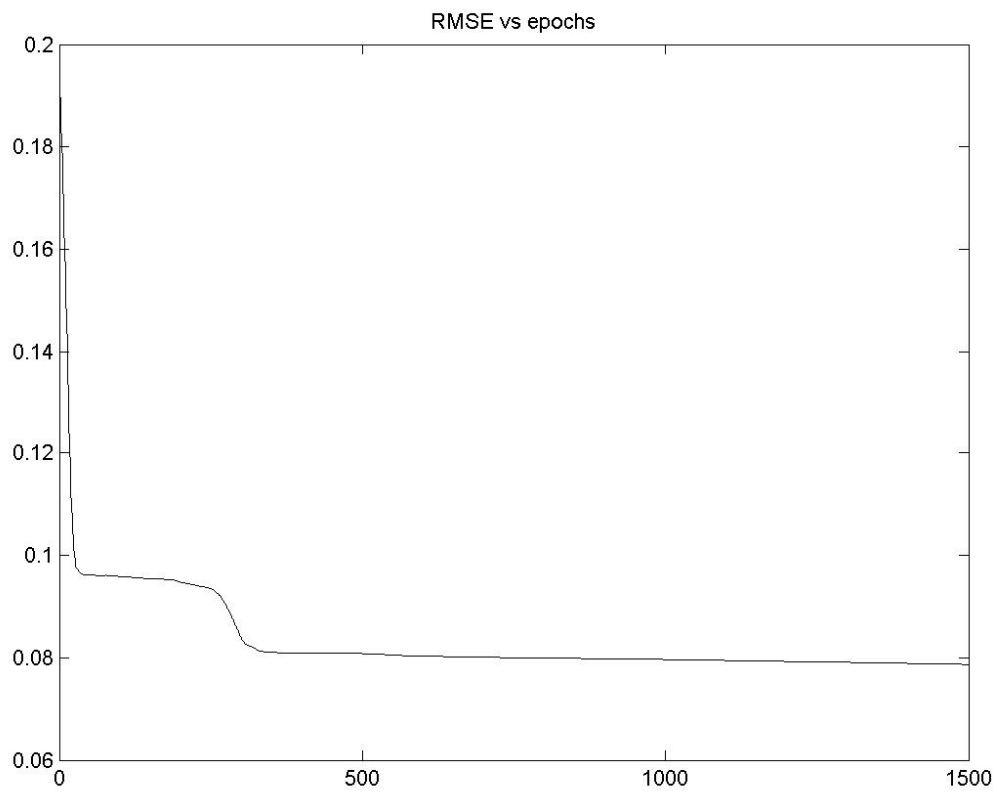
50	24	12	6	5120	1,15	0,7	4000	0,017651
51	24	12	6	5120	1,15	0,9	4000	0,017057
52	24	12	6	5120	1,25	0,5	4000	0,004515
53	24	12	6	5120	1,25	0,7	4000	0,009908
54	24	12	6	5120	1,25	0,9	4000	0,005270

Πίνακας 4.1

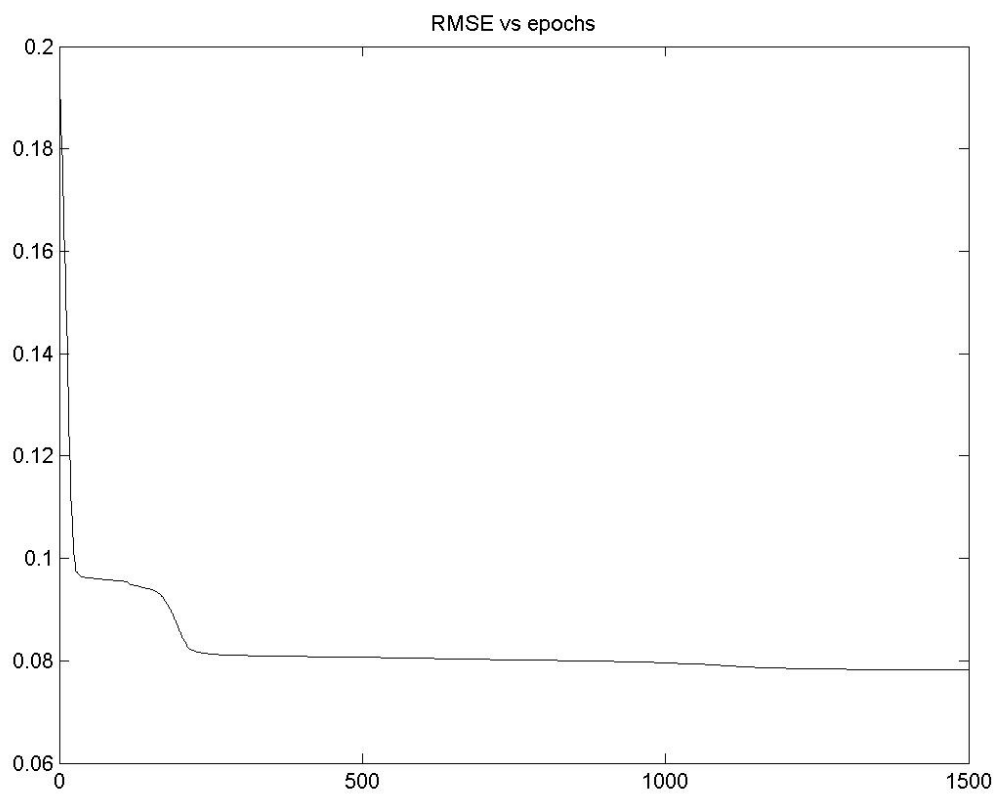
Ελάχιστο MSE	Μέγιστο MSE	Μέσος Όρος	Τυπική Απόκλιση	(Τυπική Απόκλιση / Μέσος Όρος)*100
0,003759	0,043563	0,013455	0,009365	69,59951 %

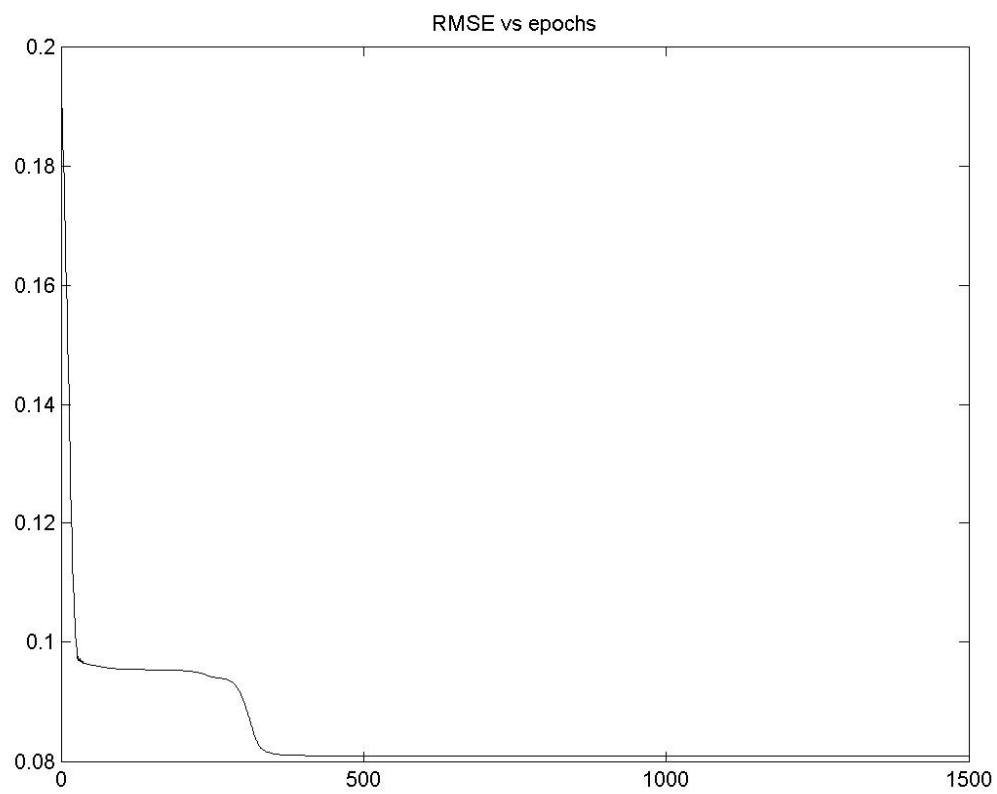
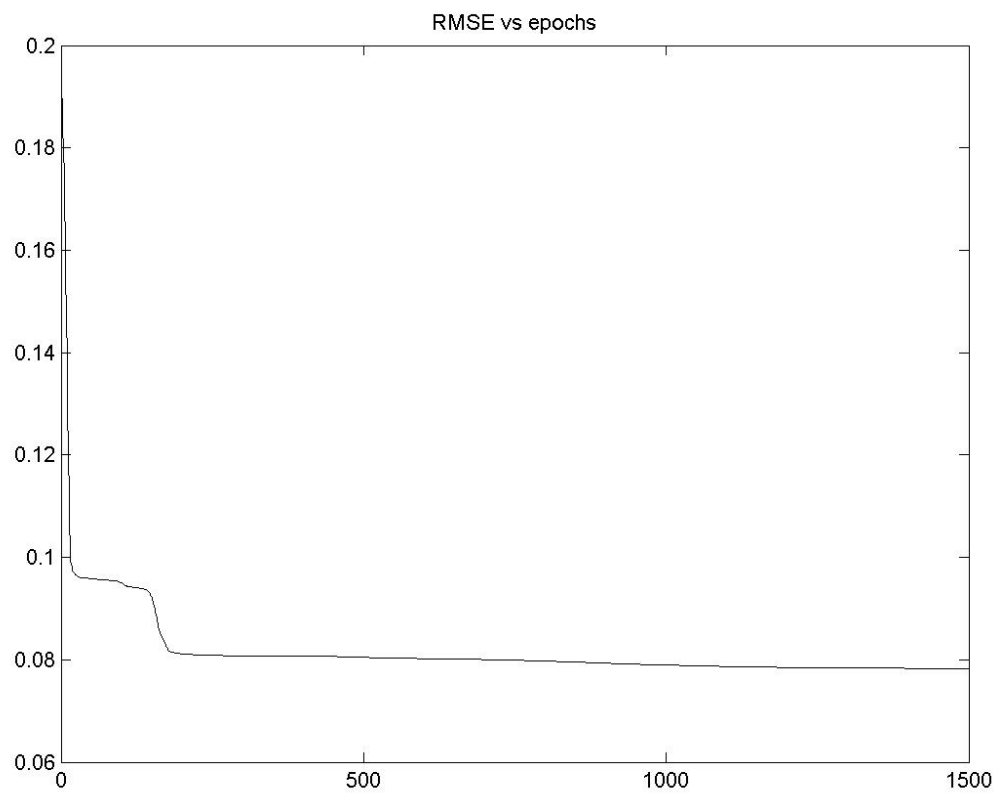
Στις επόμενες σελίδες ακολουθούν οι γραφικές απεικονίσεις των αποτελεσμάτων για κάθε ένα από τους 54 συνδυασμούς παραμέτρων εκπαίδευσης που εκτελέστηκαν για την περίπτωση του πνευμονικού ήχου SQ.

1^η Επανάληψη

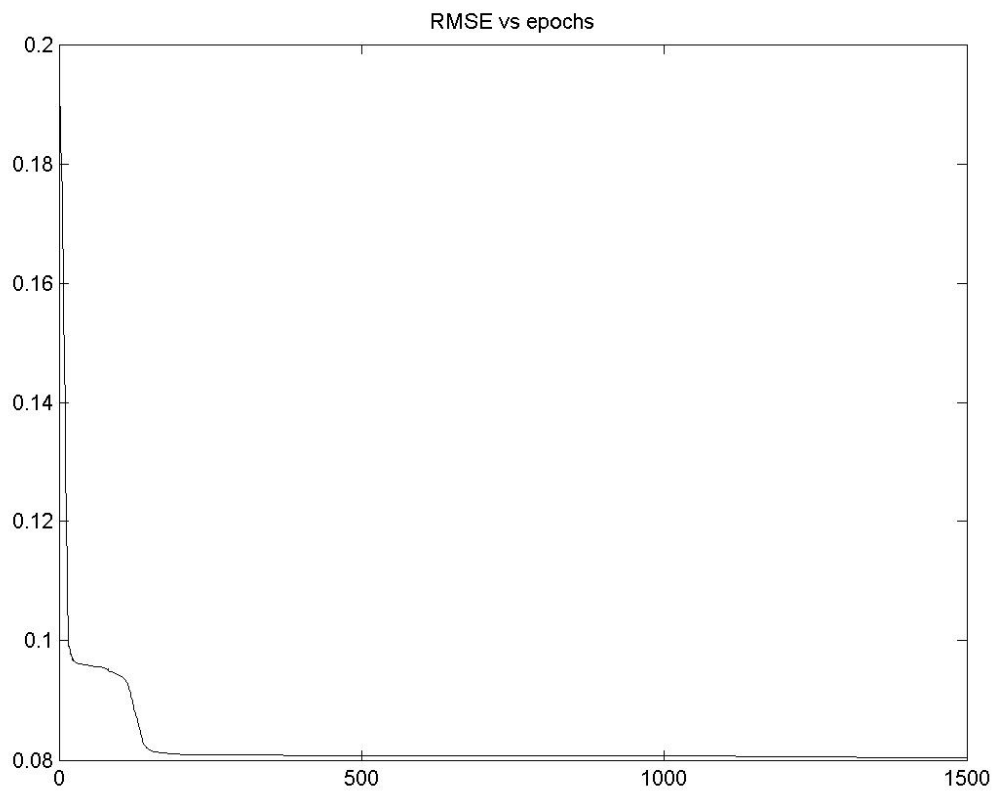


2^η Επανάληψη

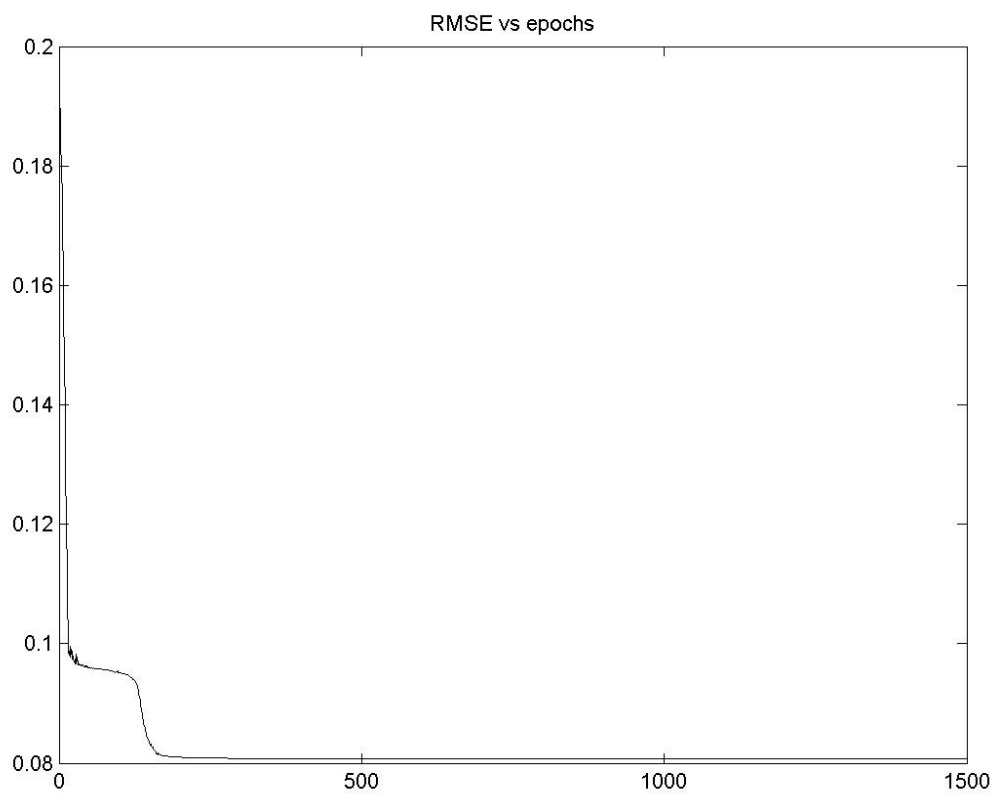


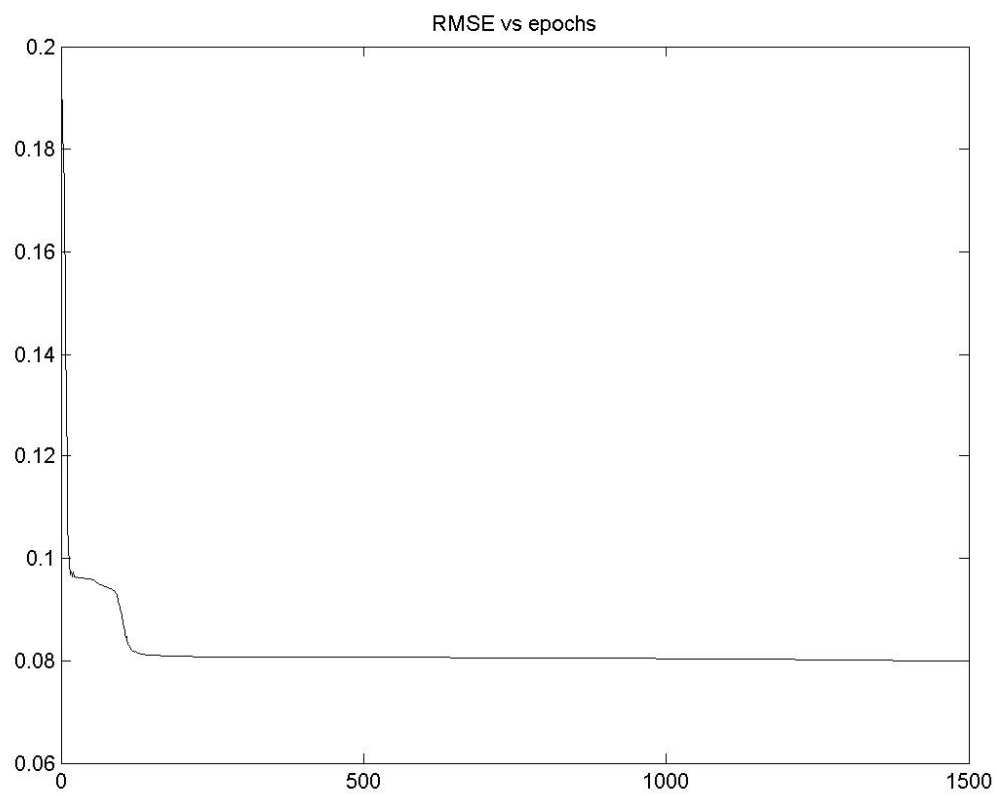
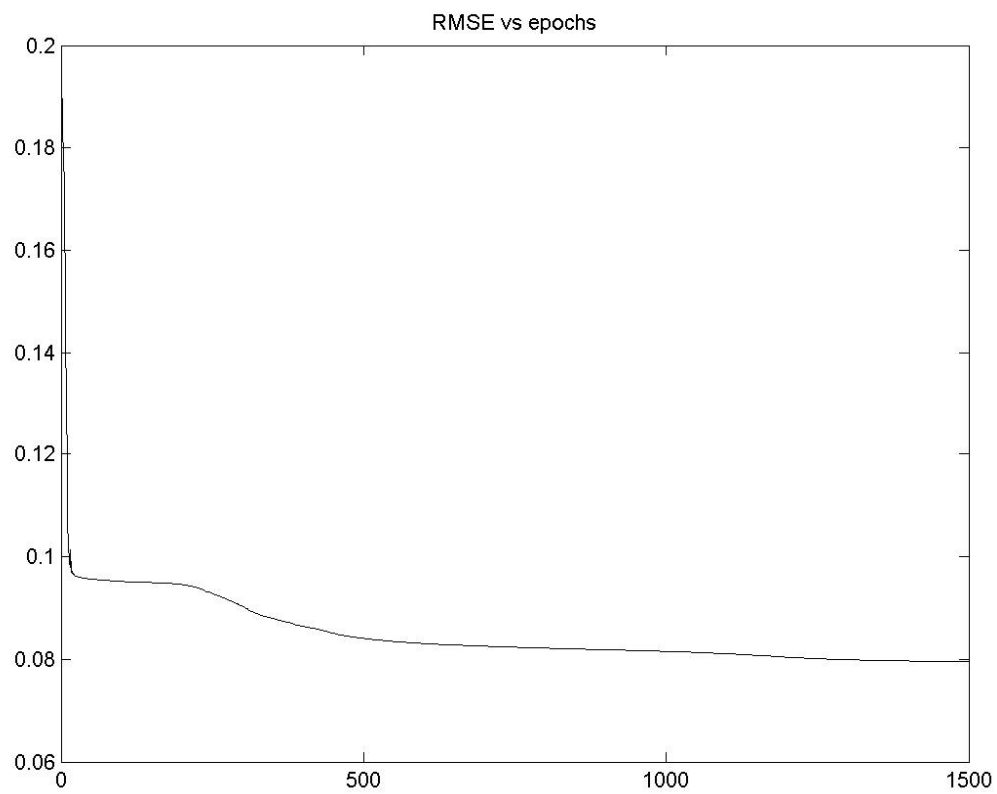
3^η Επανάληψη4^η Επανάληψη

5^η Επανάληψη

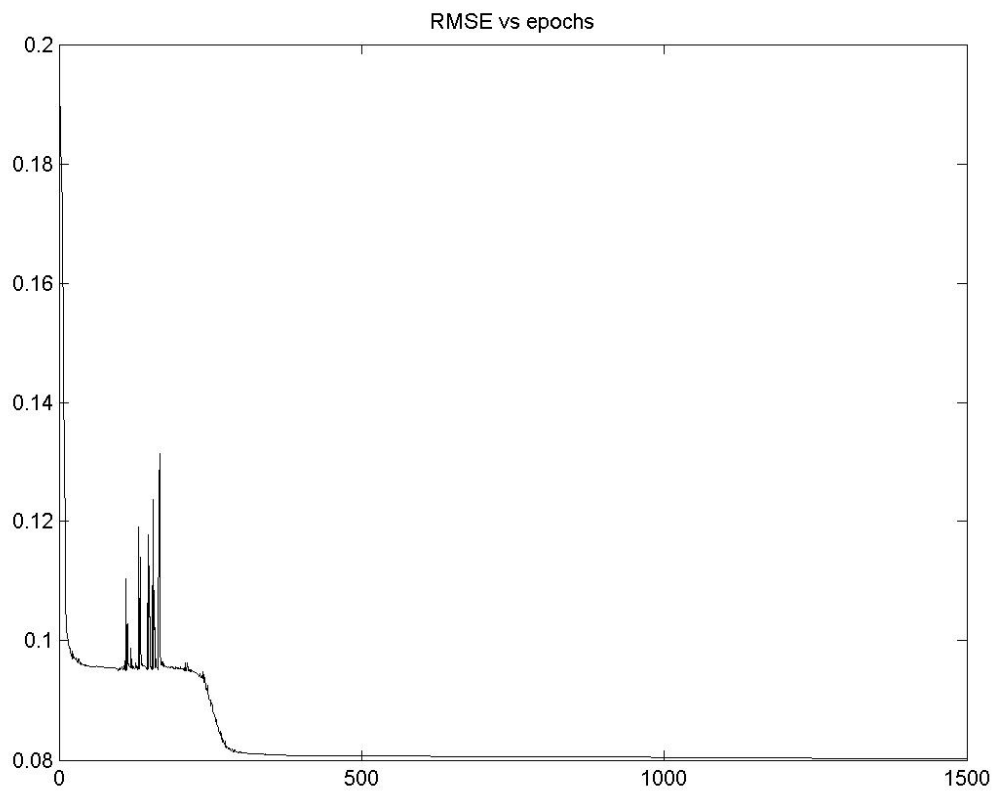


6^η Επανάληψη

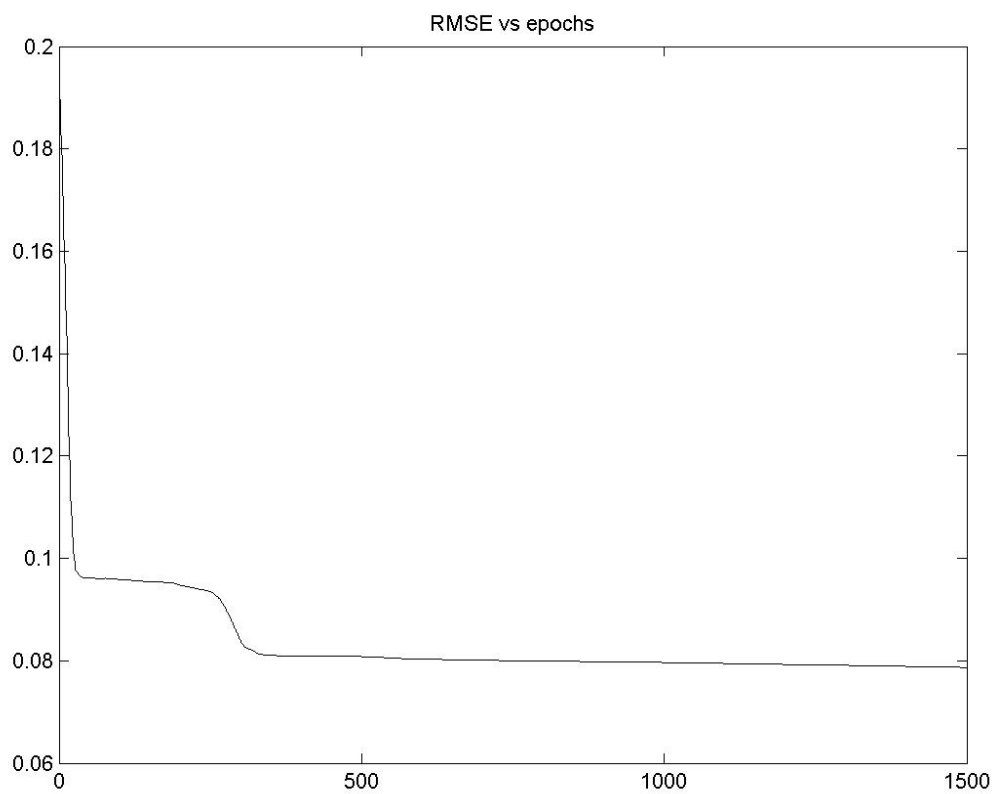


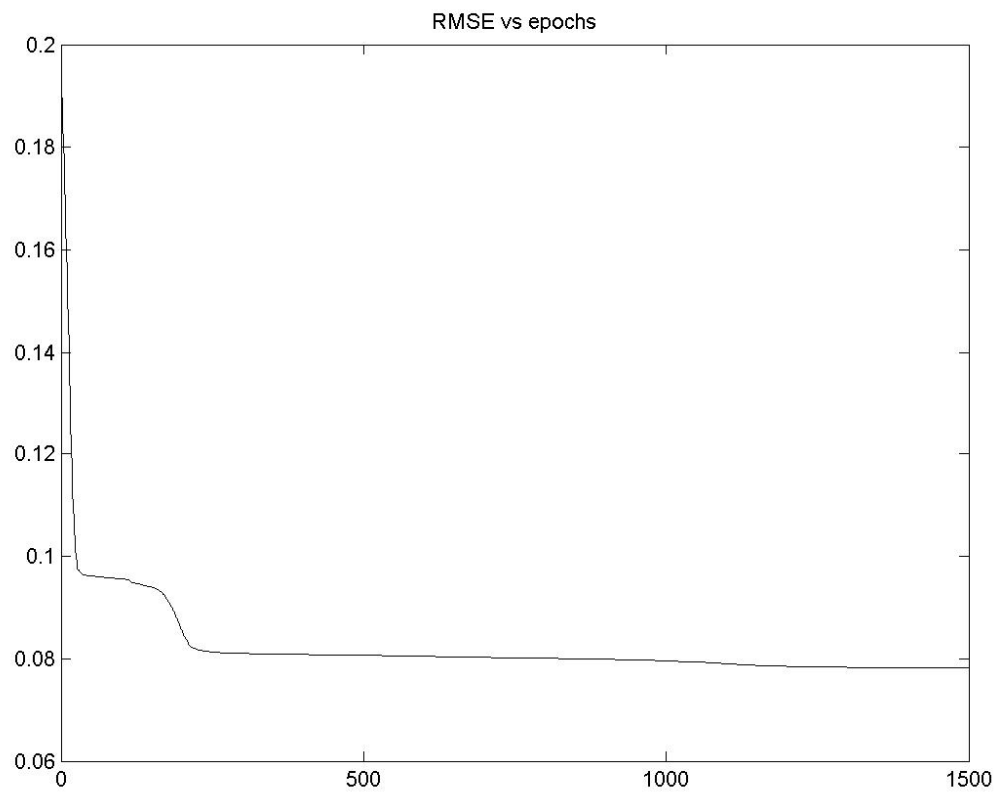
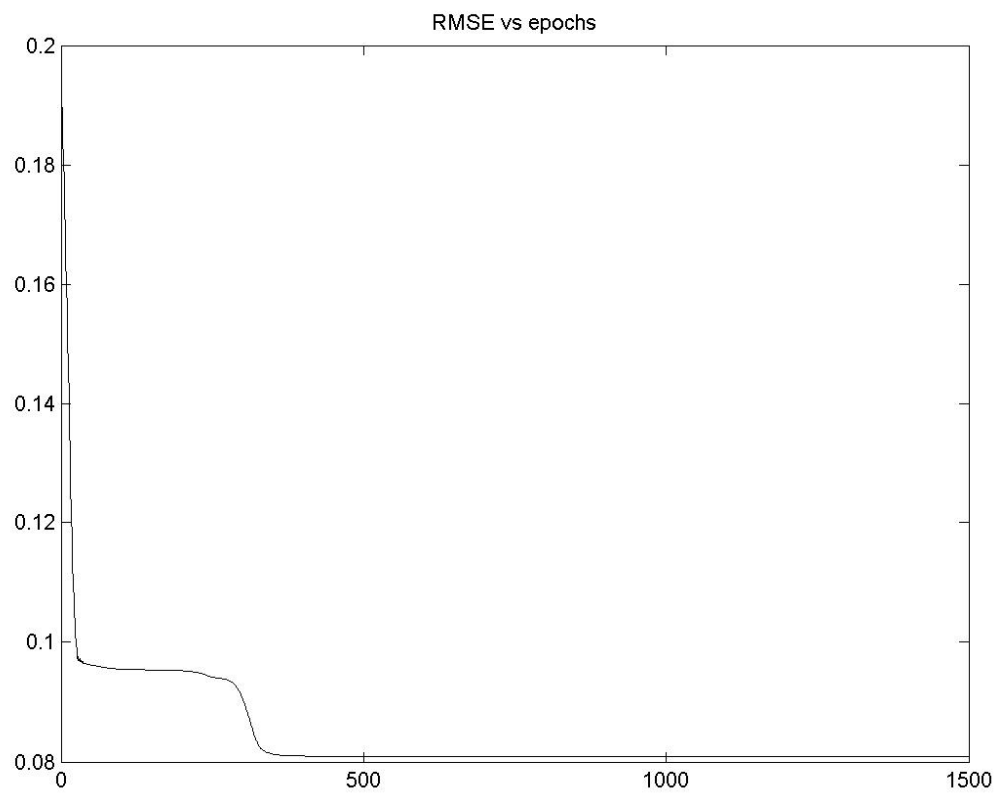
7^η Επανάληψη8^η Επανάληψη

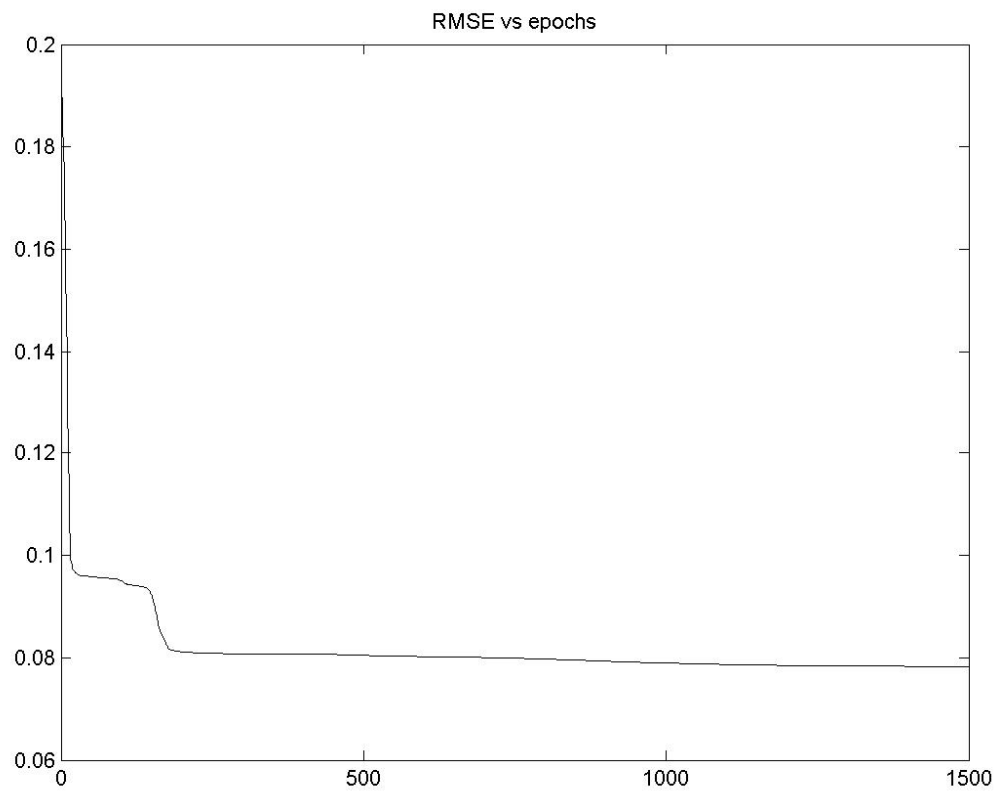
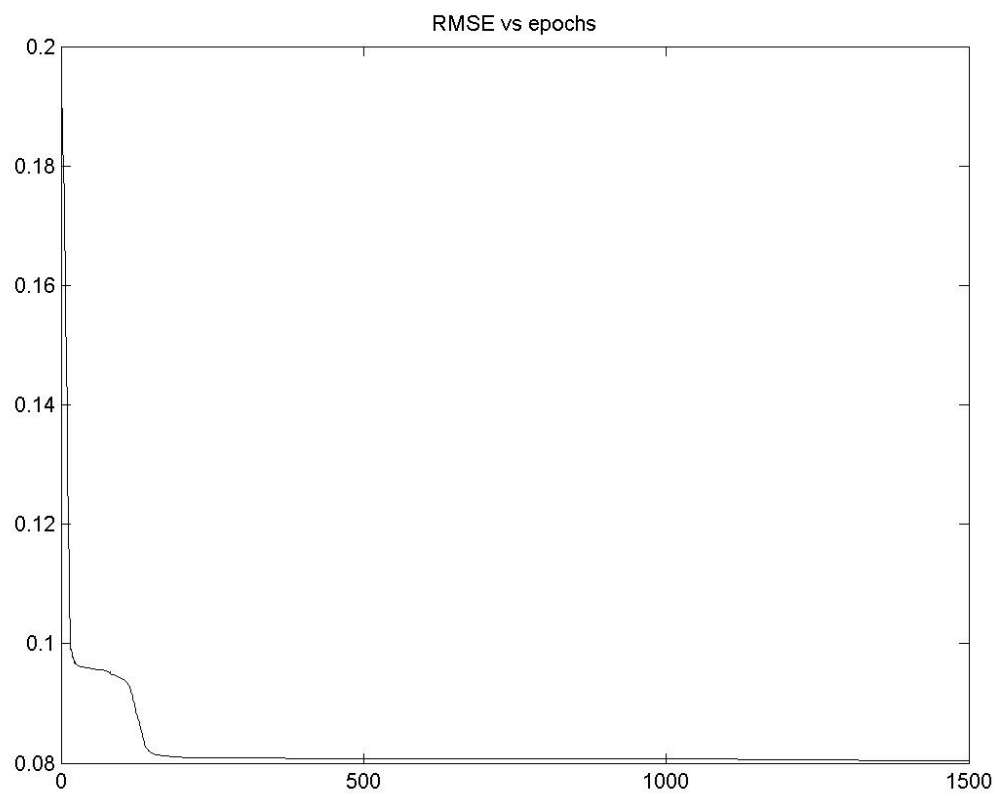
9^η Επανάληψη

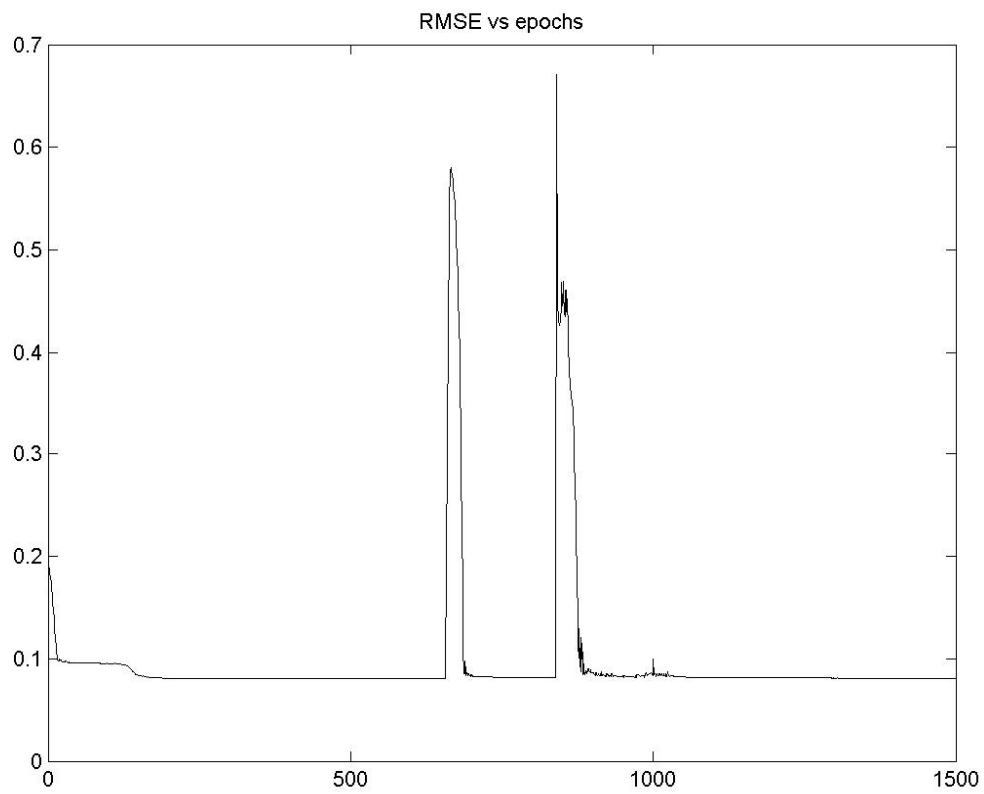
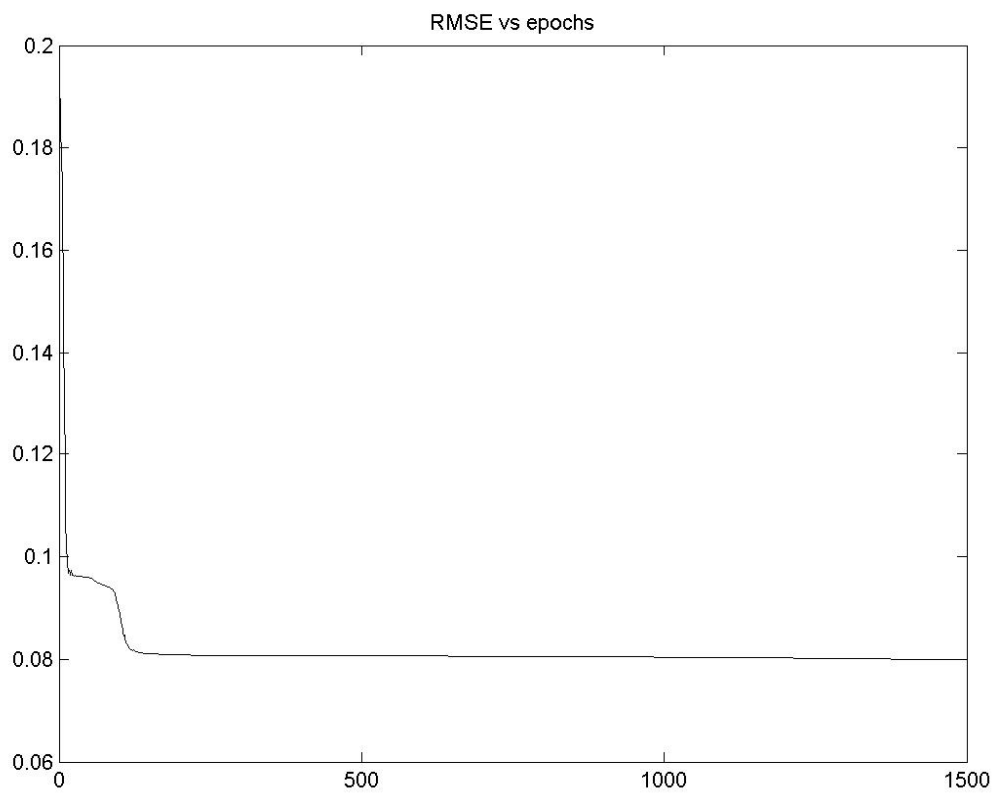


10^η Επανάληψη

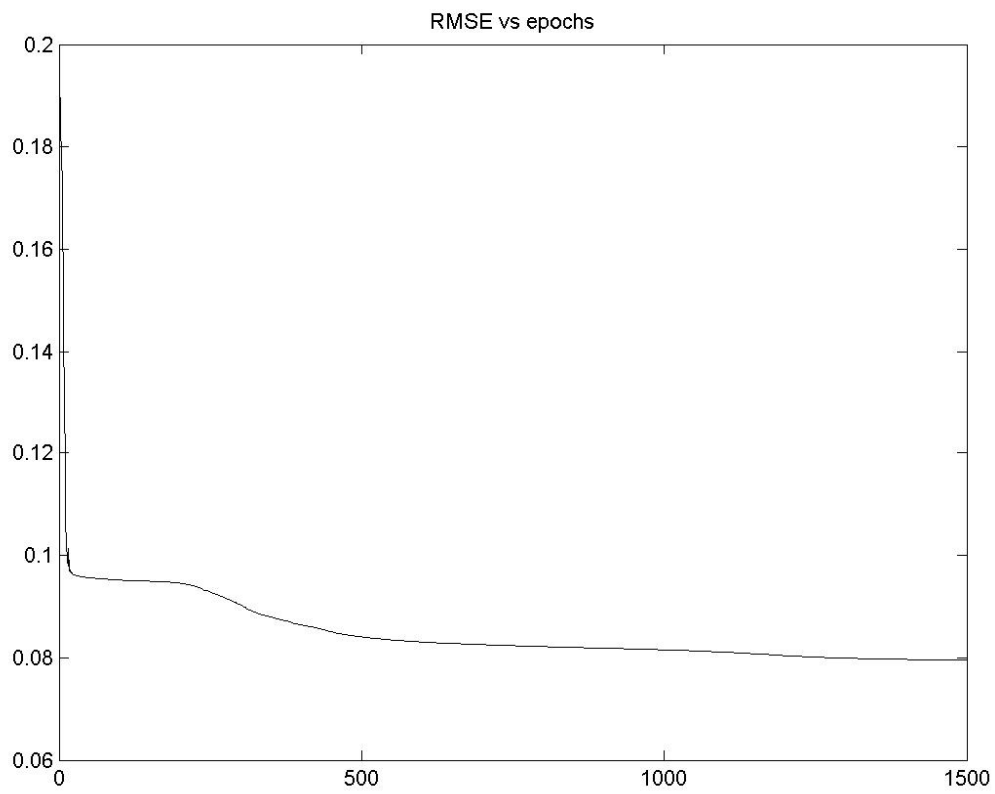


11^η Επανάληψη12^η Επανάληψη

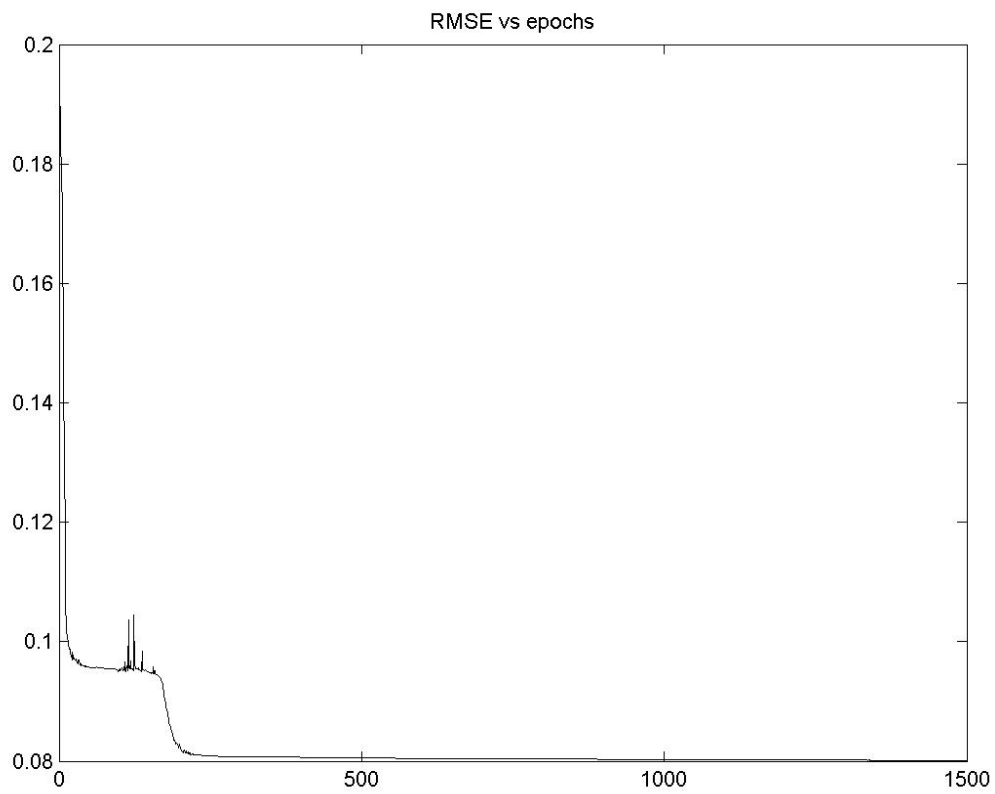
13^η Επανάληψη14^η Επανάληψη

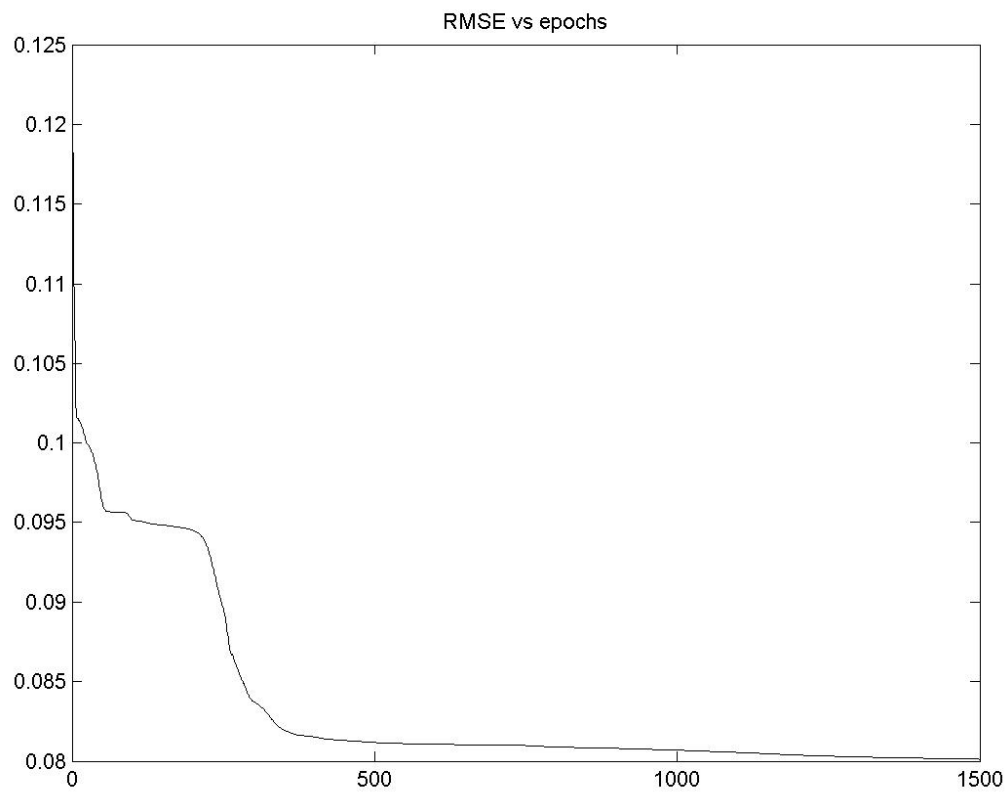
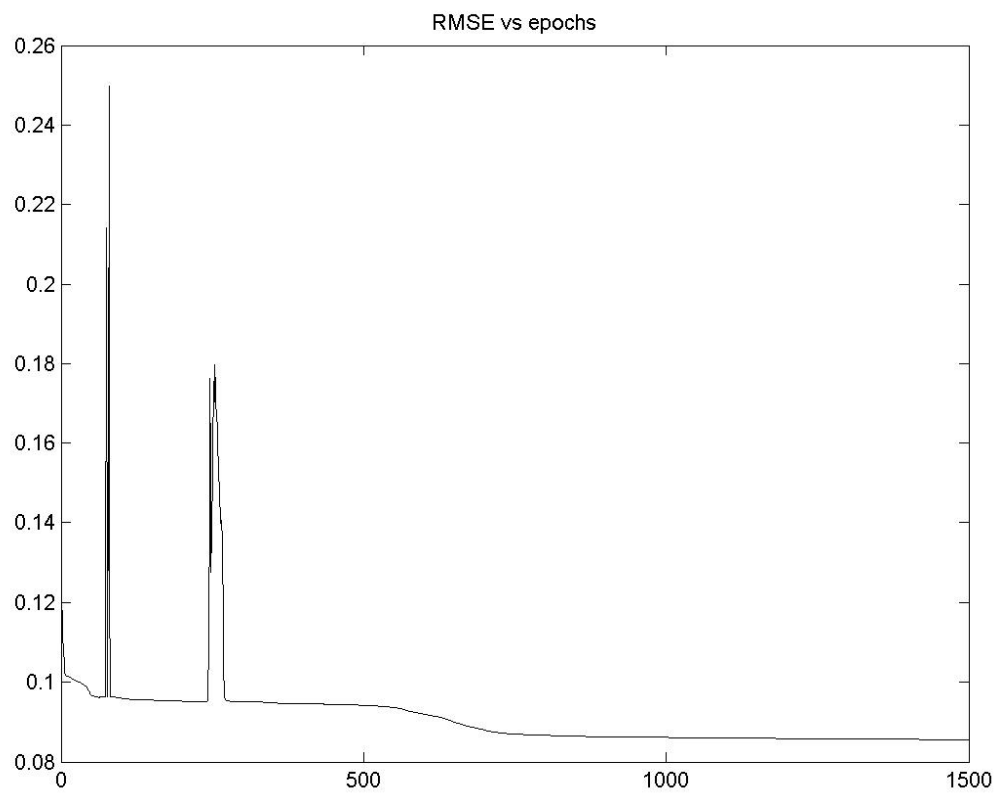
15^η Επανάληψη16^η Επανάληψη

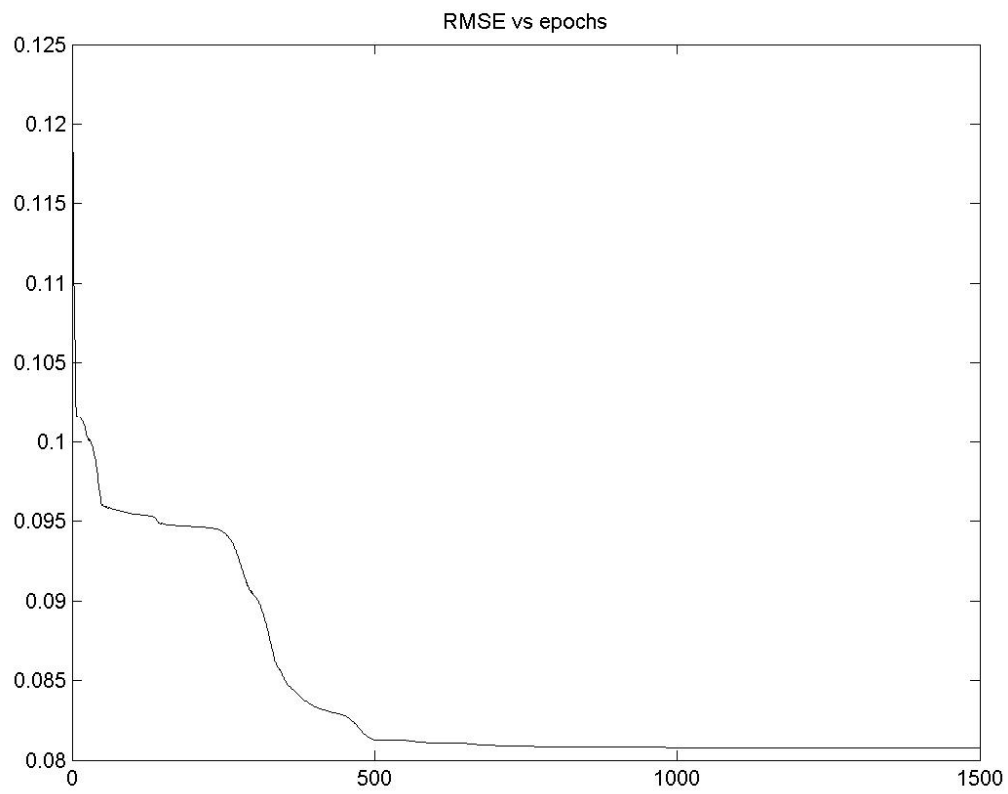
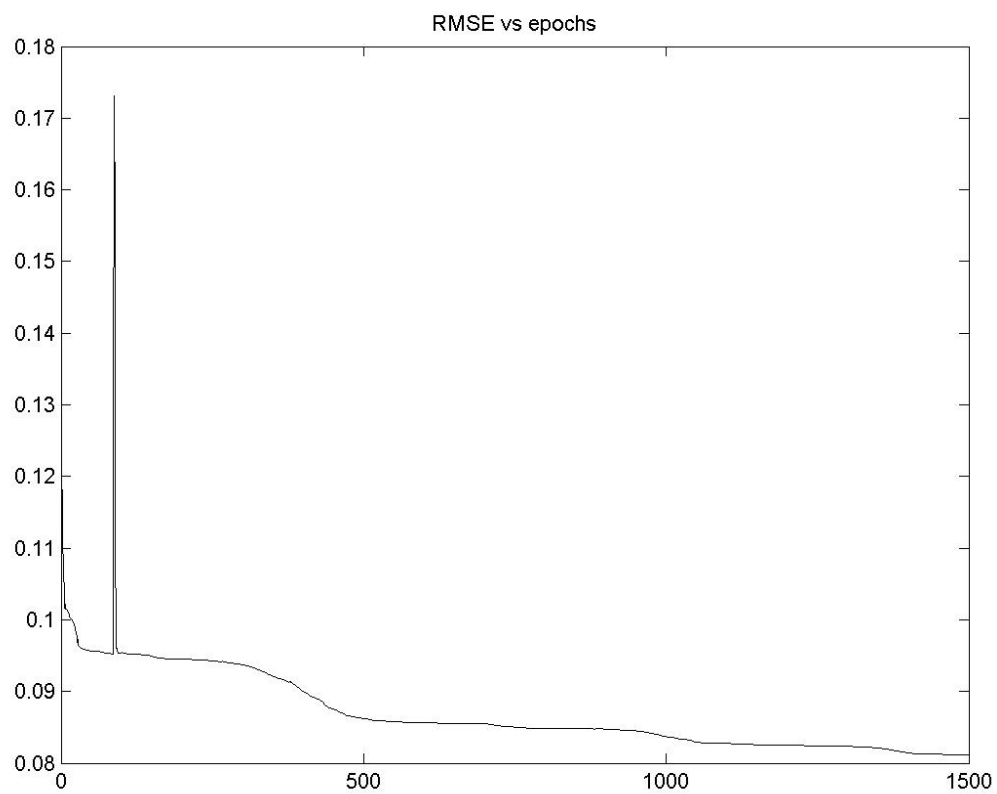
17^η Επανάληψη

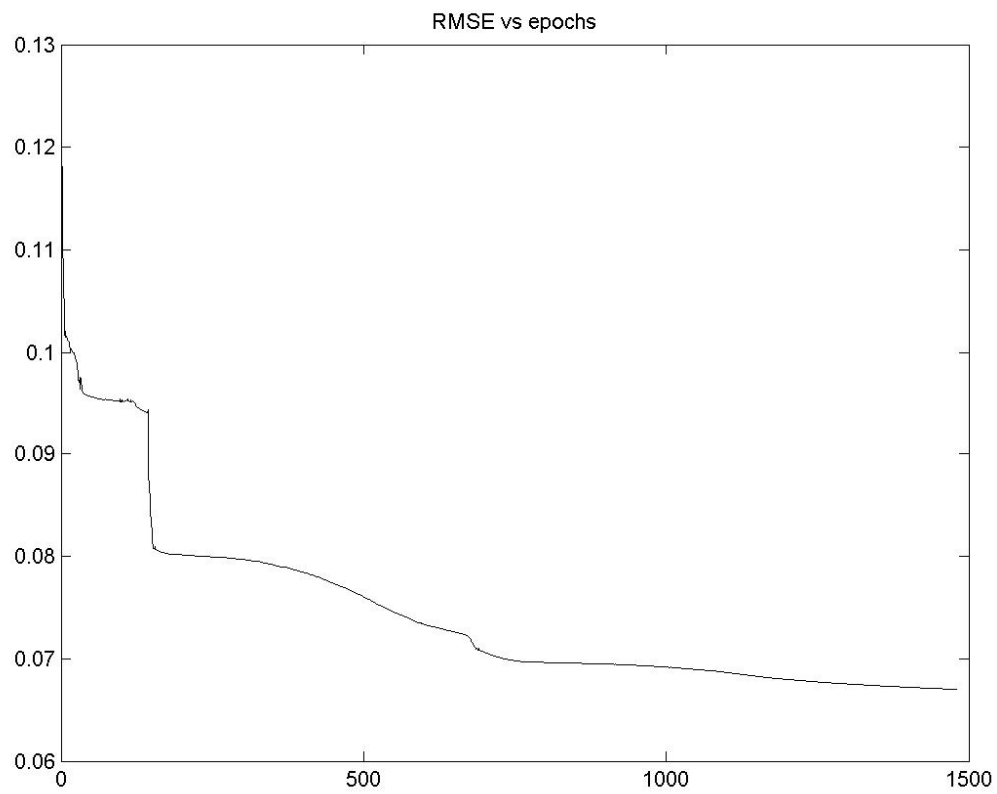
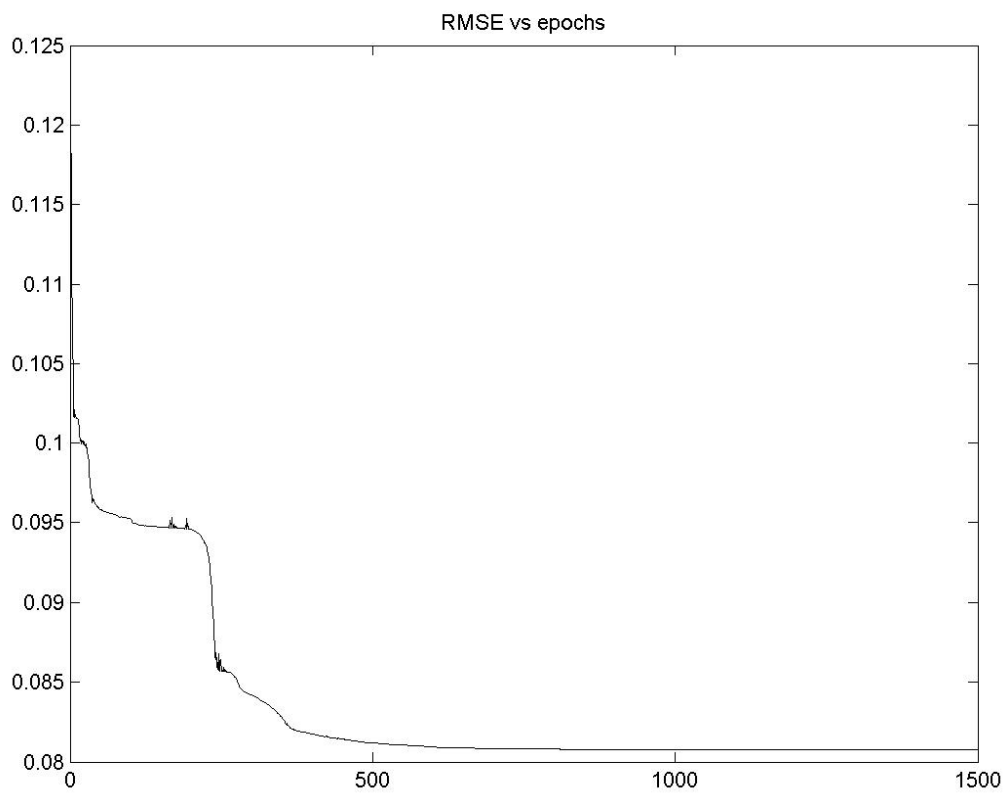


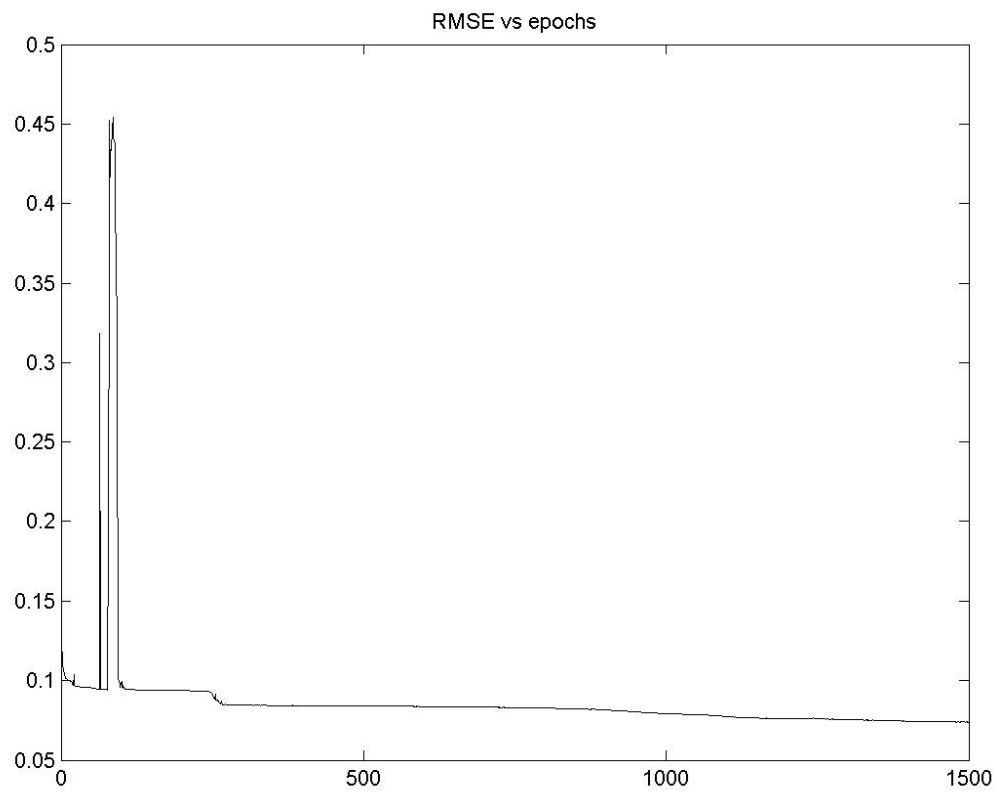
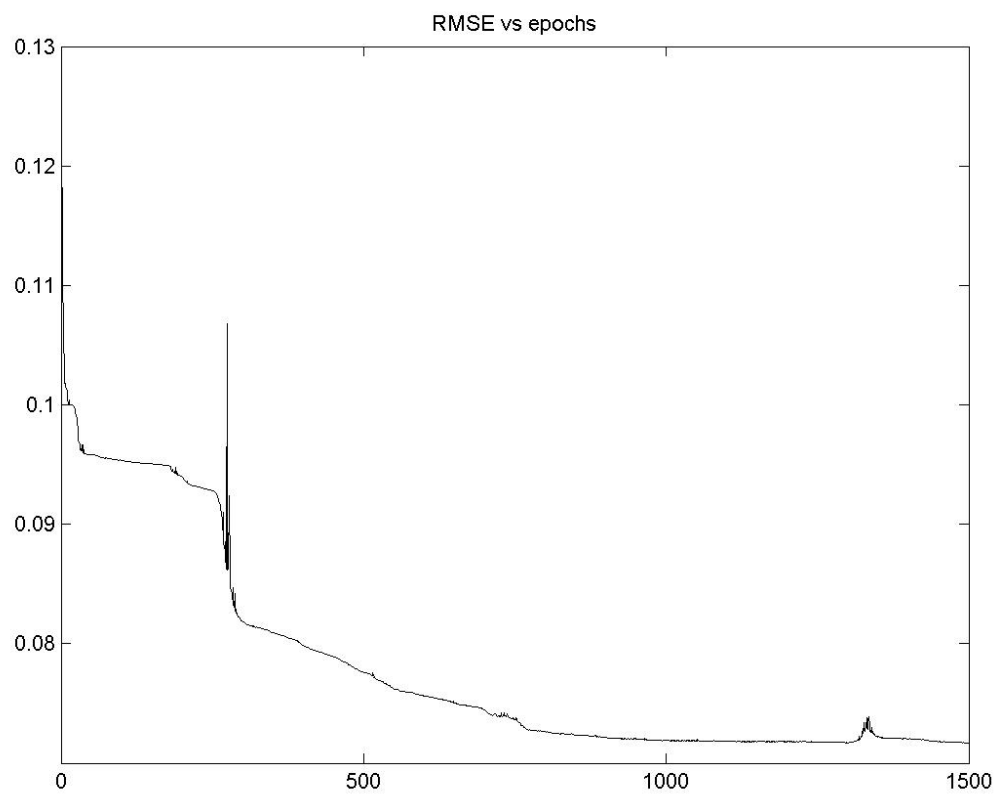
18^η Επανάληψη

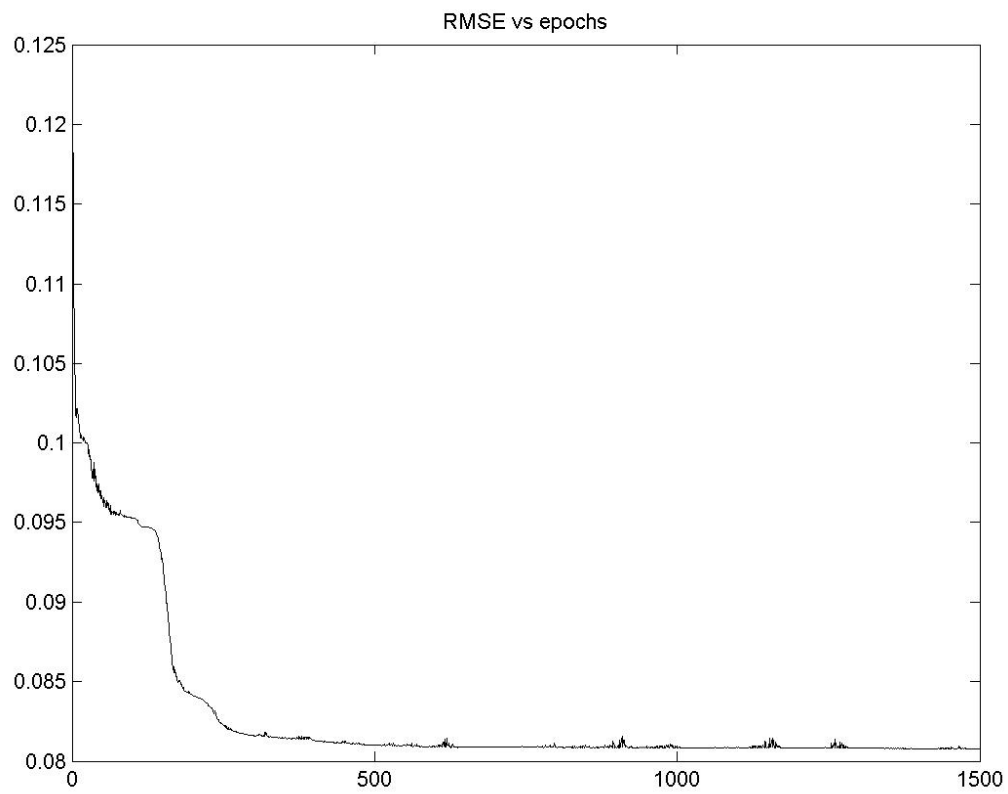
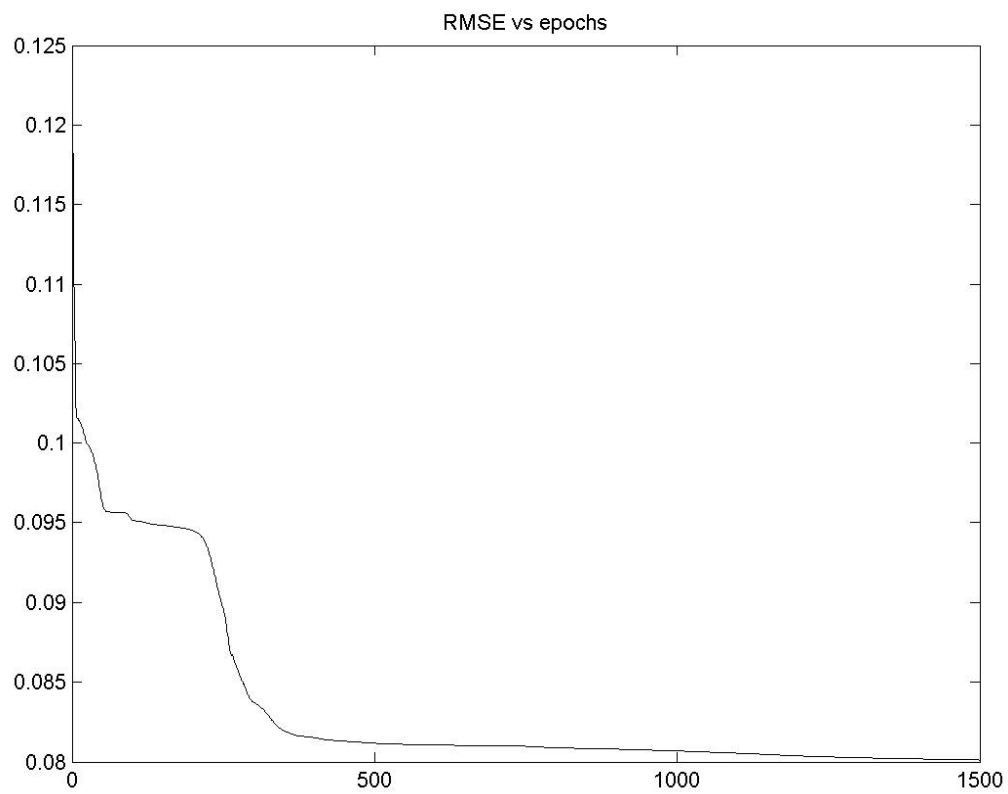


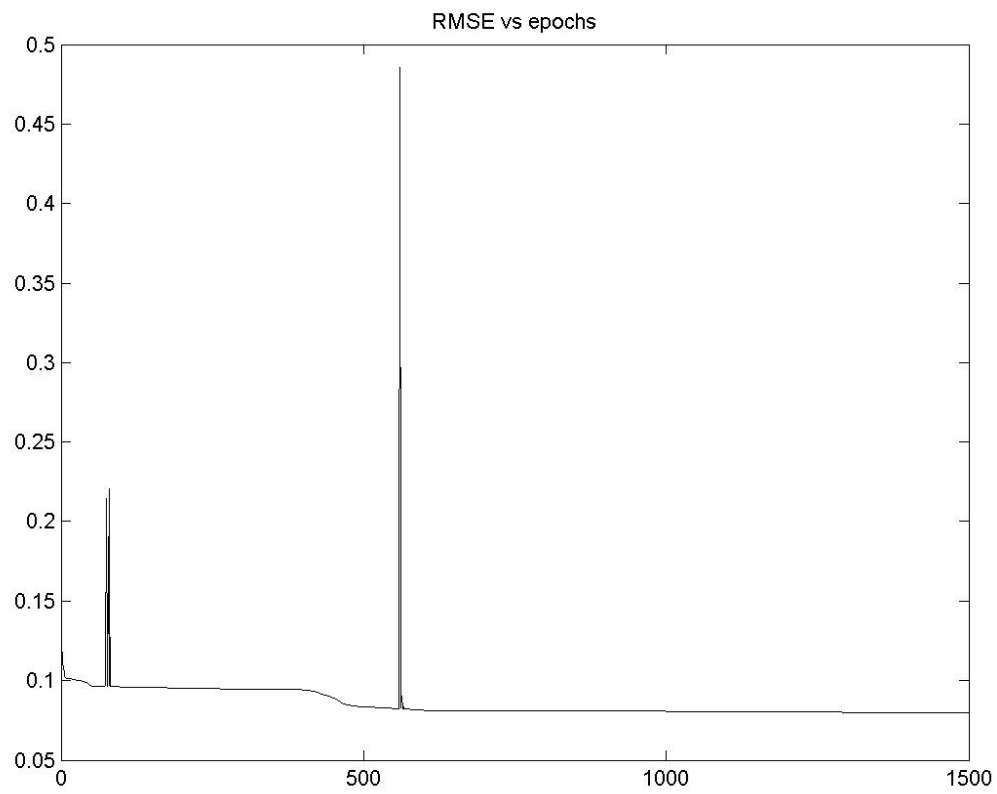
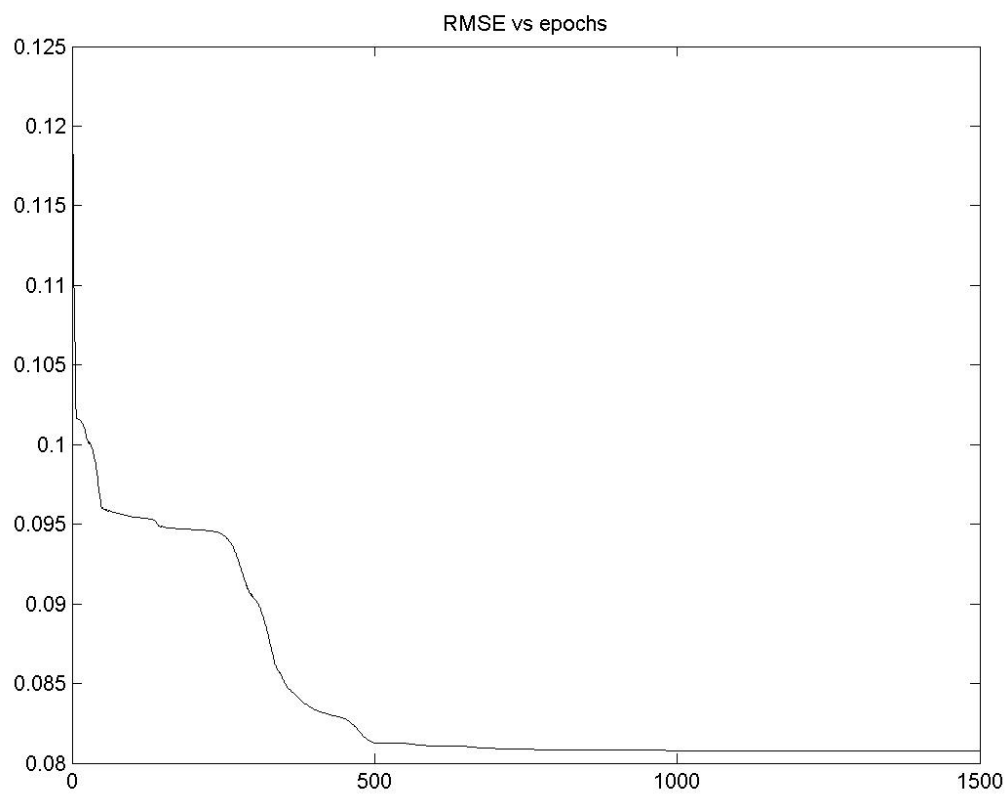
19^η Επανάληψη20^η Επανάληψη

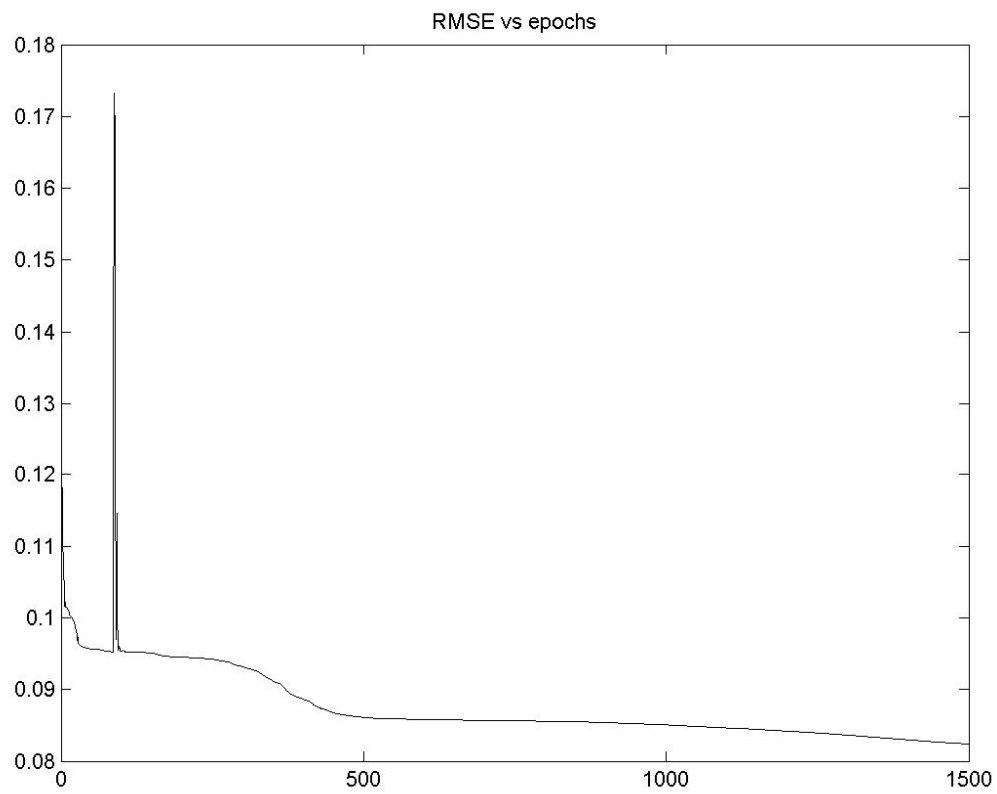
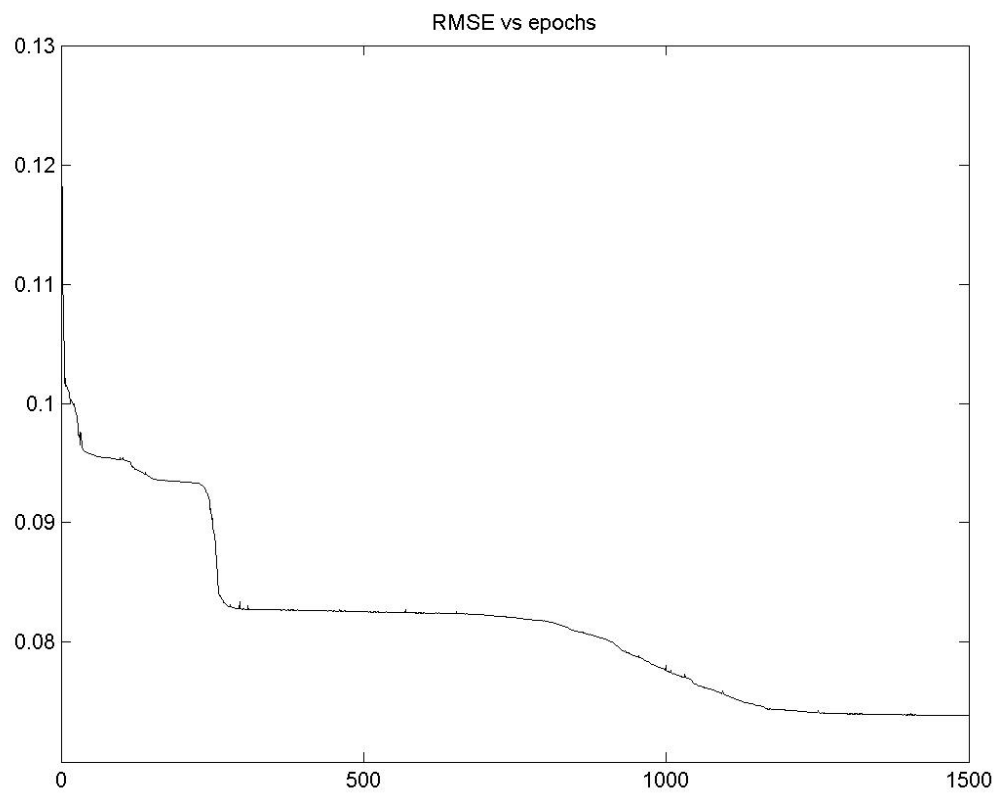
21^η Επανάληψη22^η Επανάληψη

23^η Επανάληψη24^η Επανάληψη

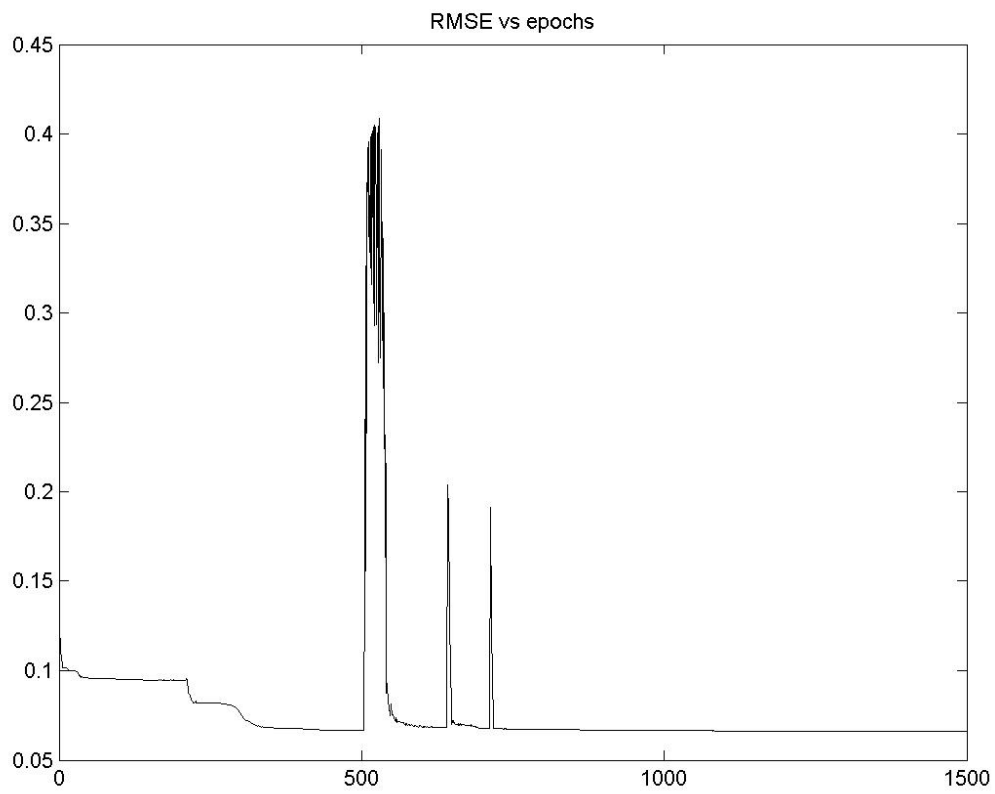
25^η Επανάληψη26^η Επανάληψη

27^η Επανάληψη28^η Επανάληψη

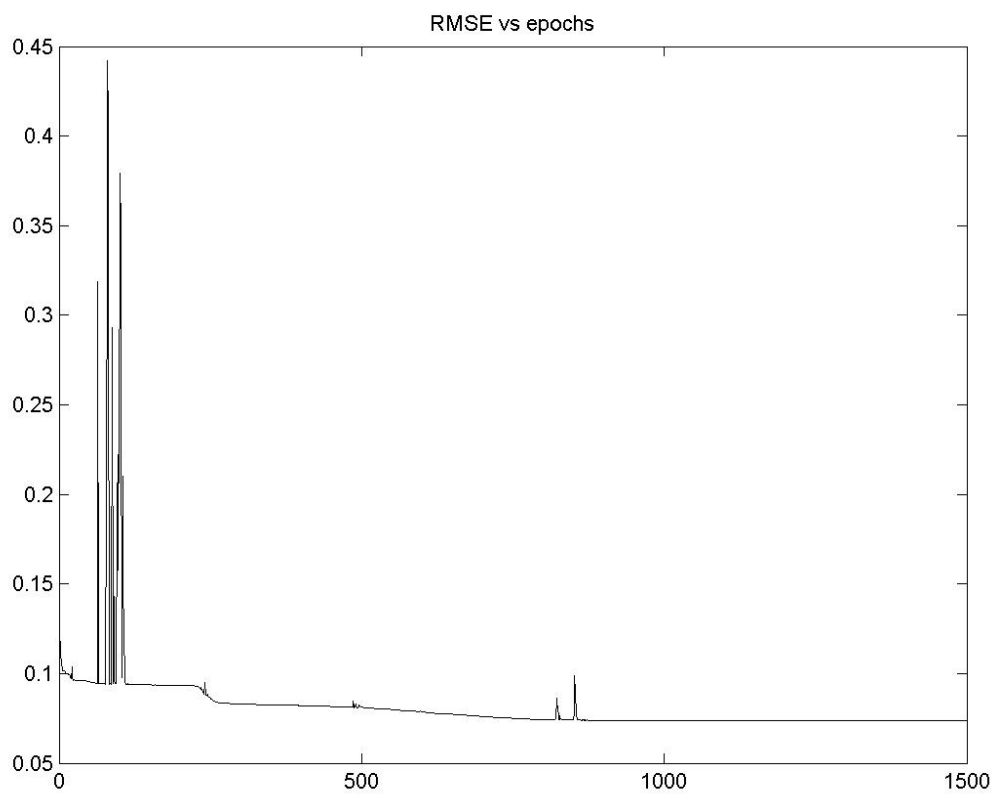
29^η Επανάληψη30^η Επανάληψη

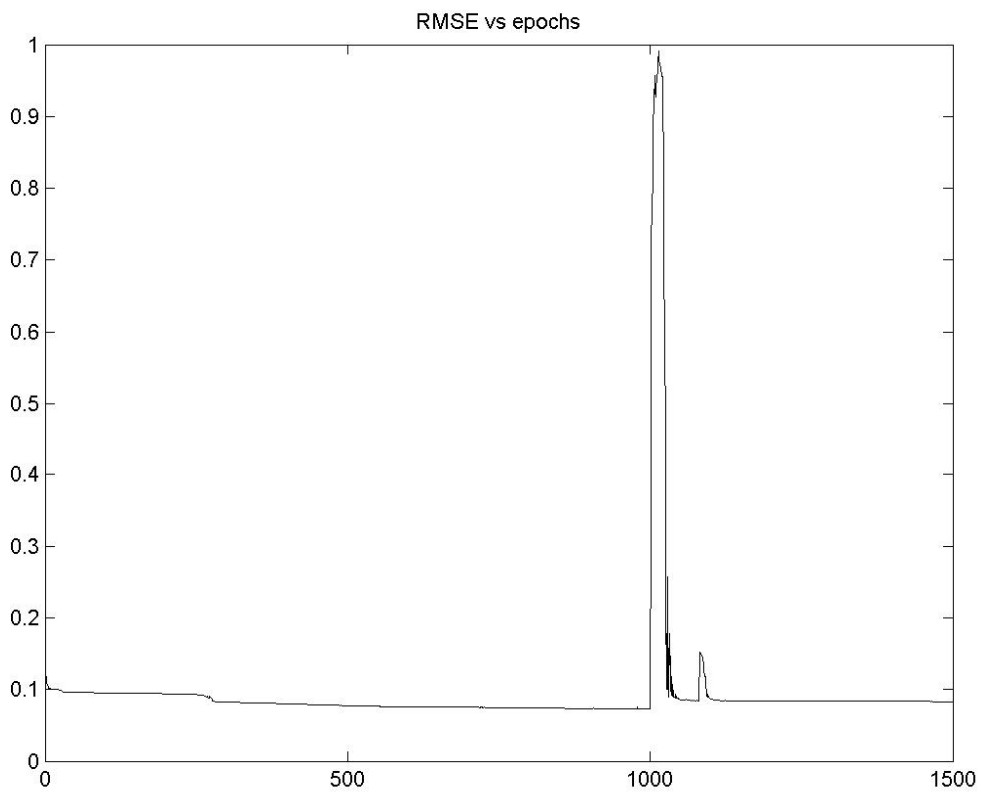
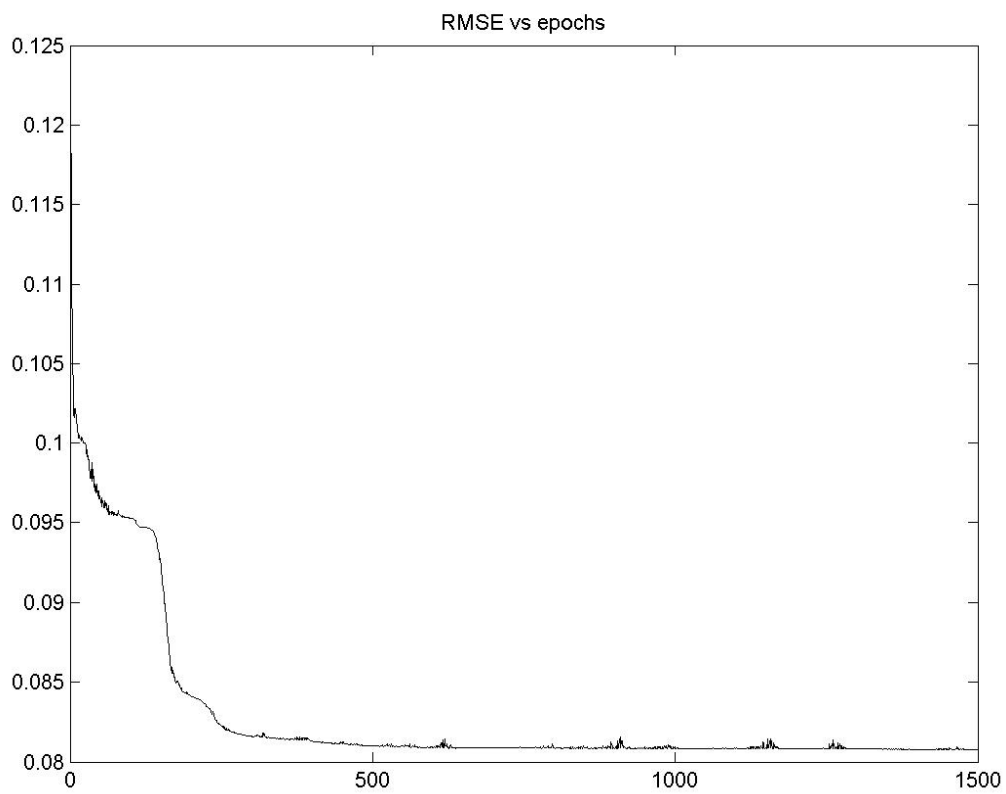
31^η Επανάληψη32^η Επανάληψη

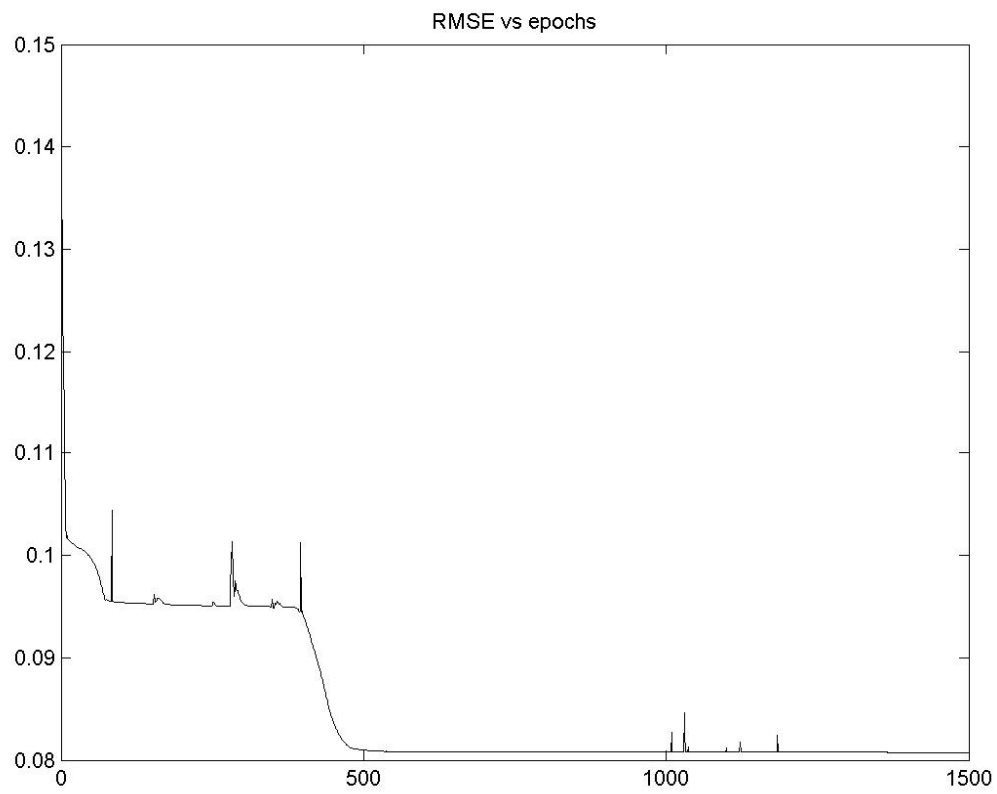
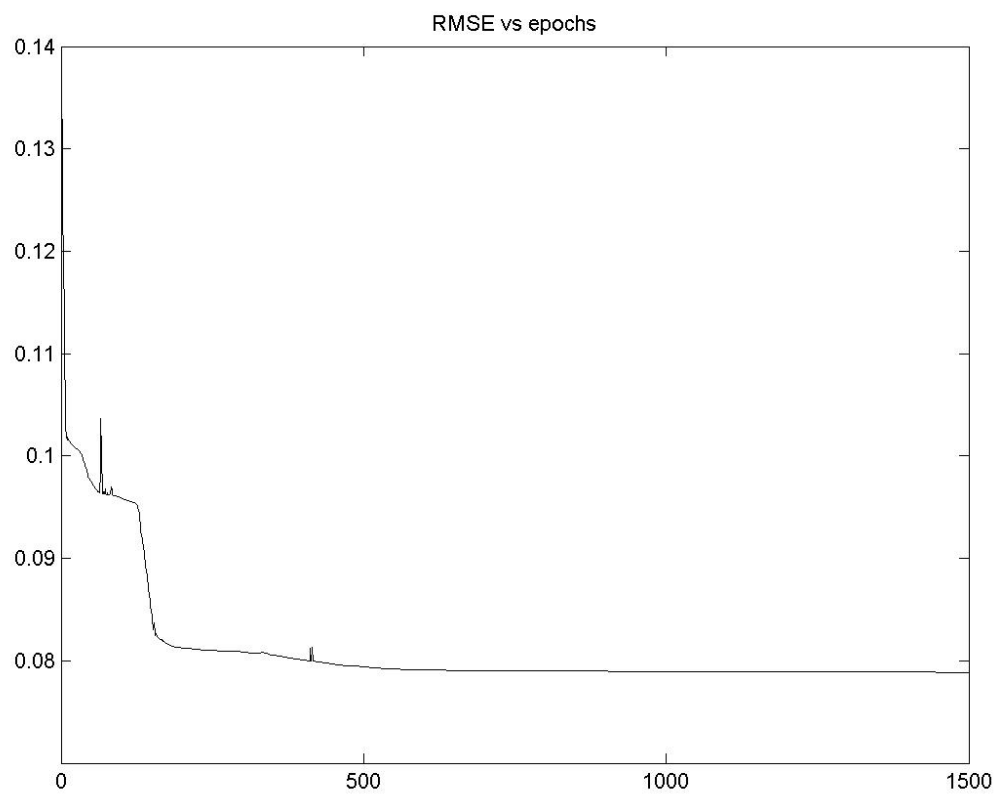
33^η Επανάληψη

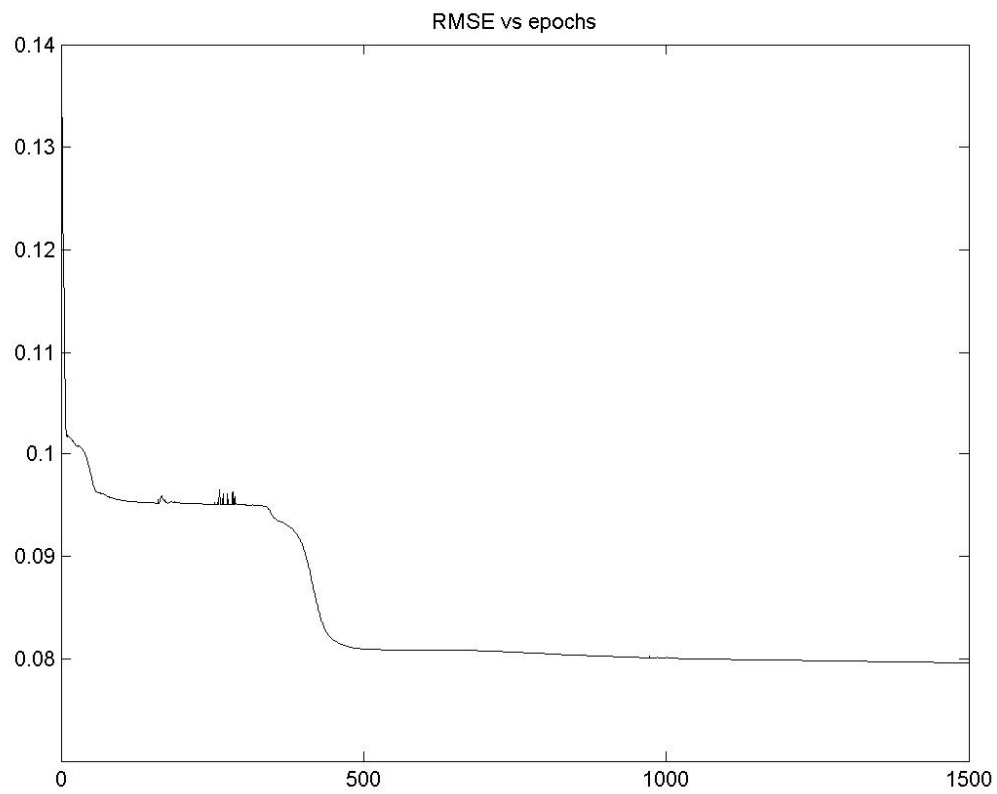
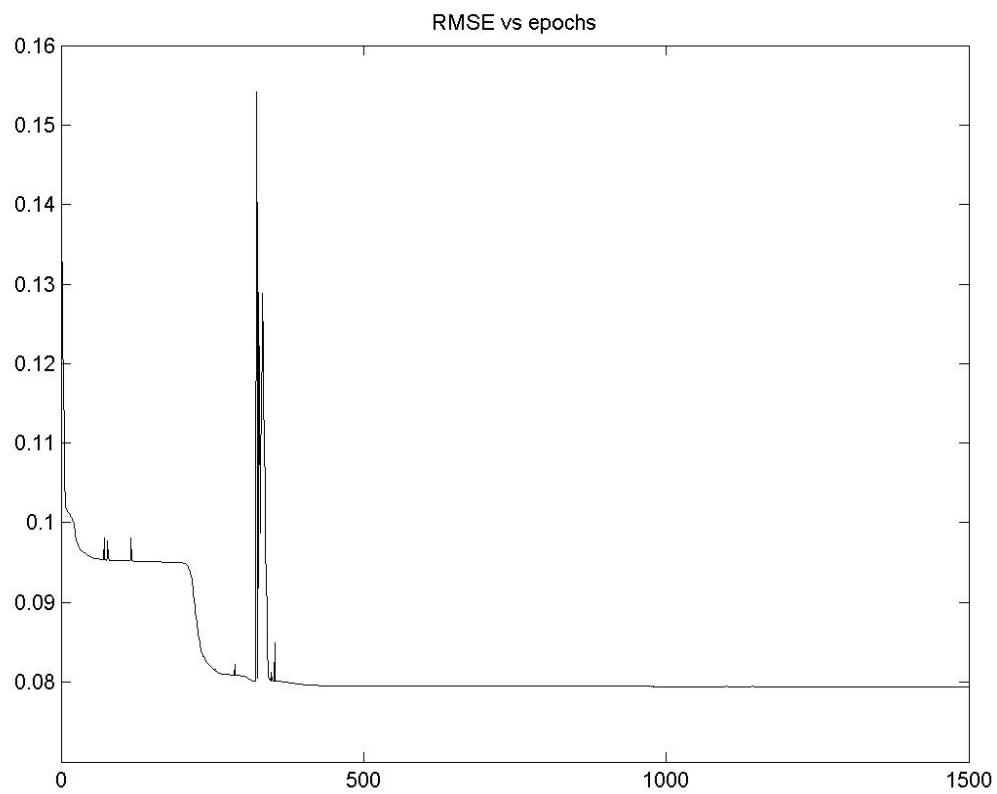


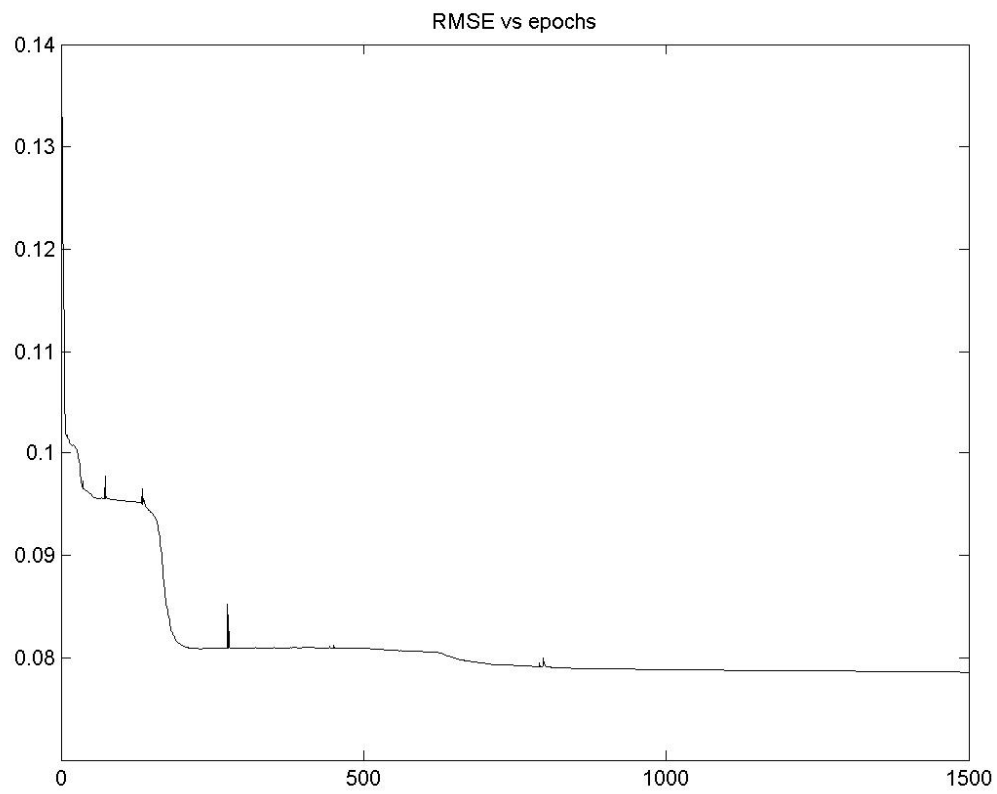
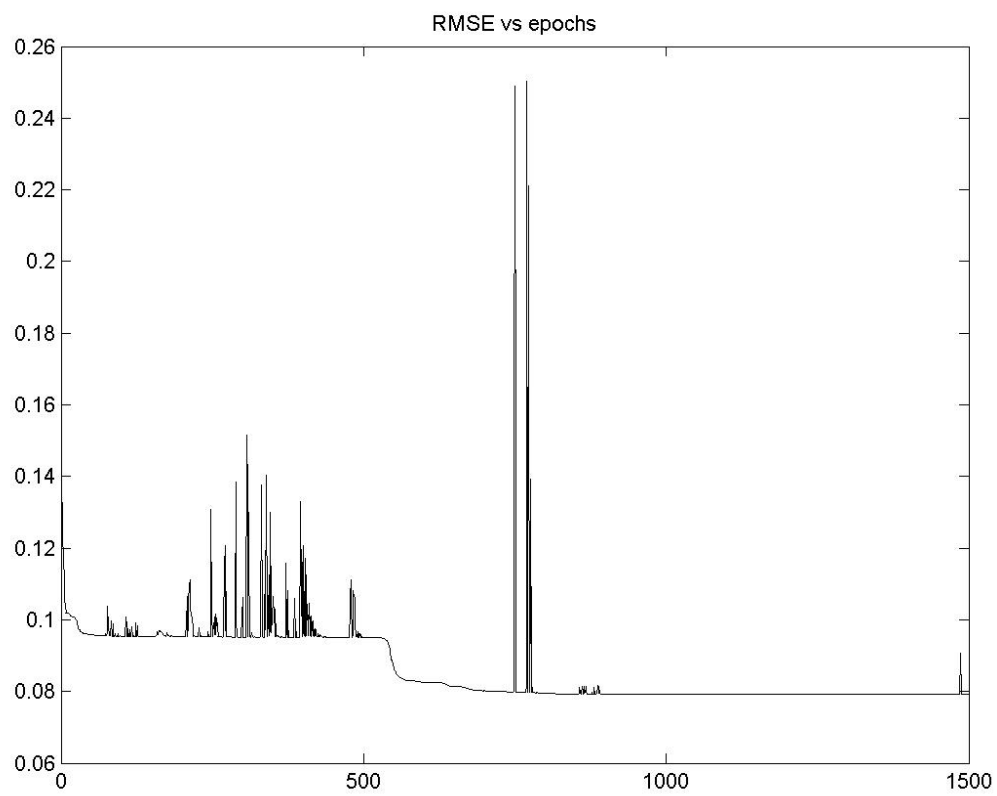
34^η Επανάληψη

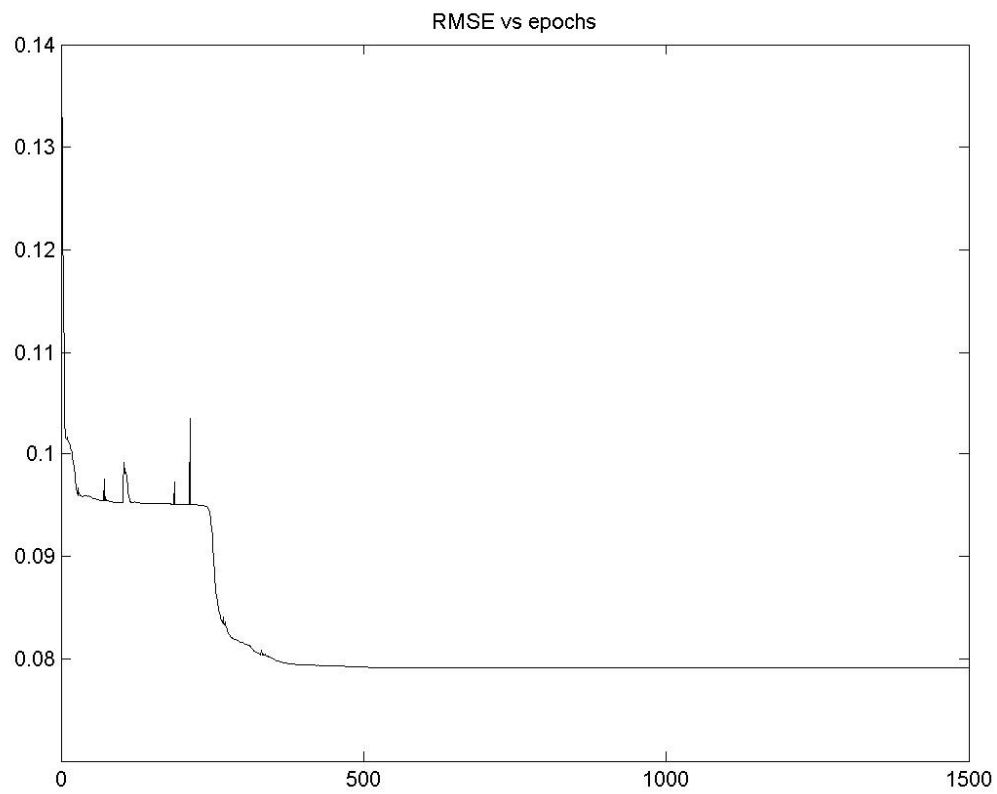
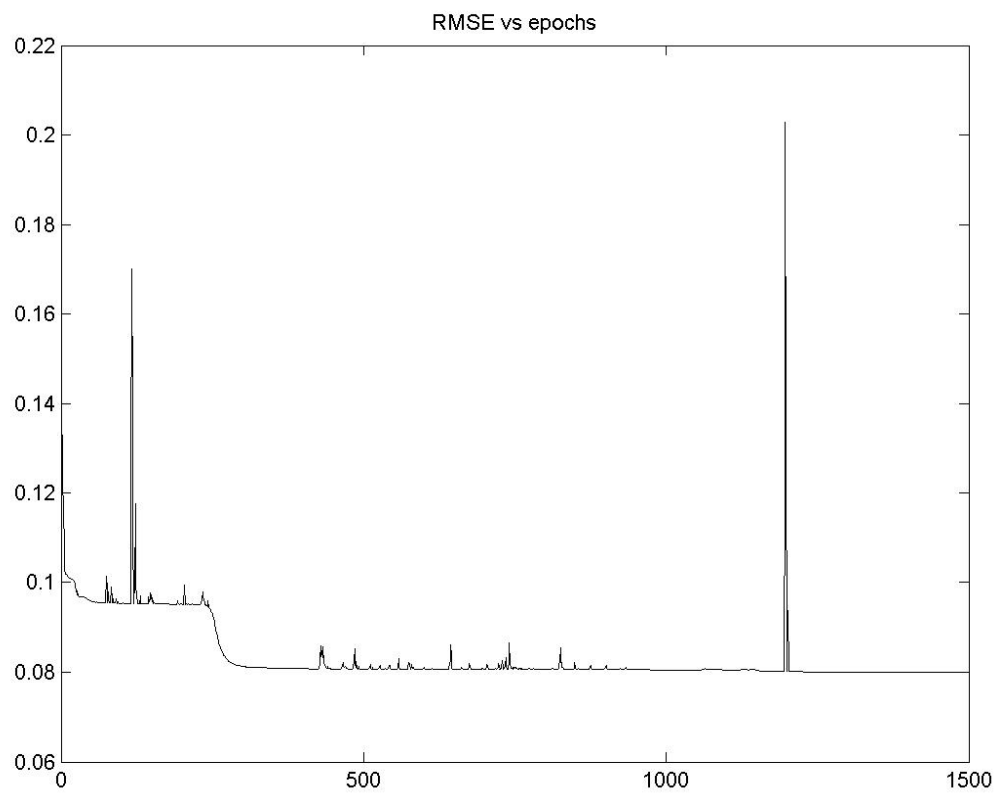


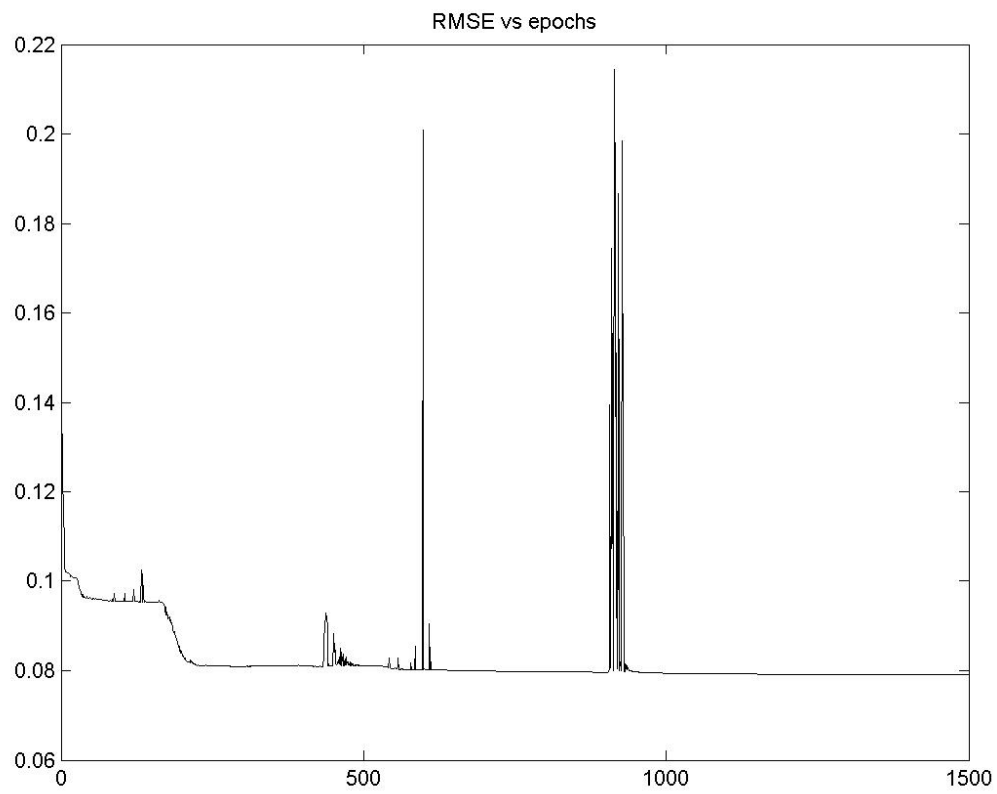
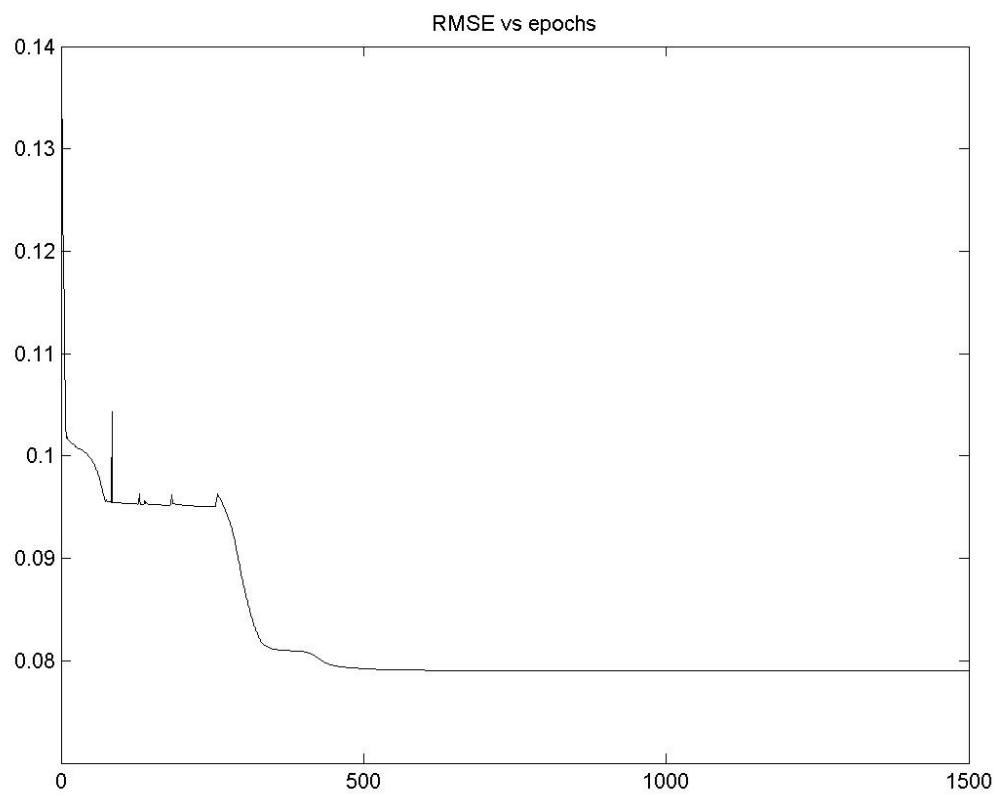
35^η Επανάληψη36^η Επανάληψη

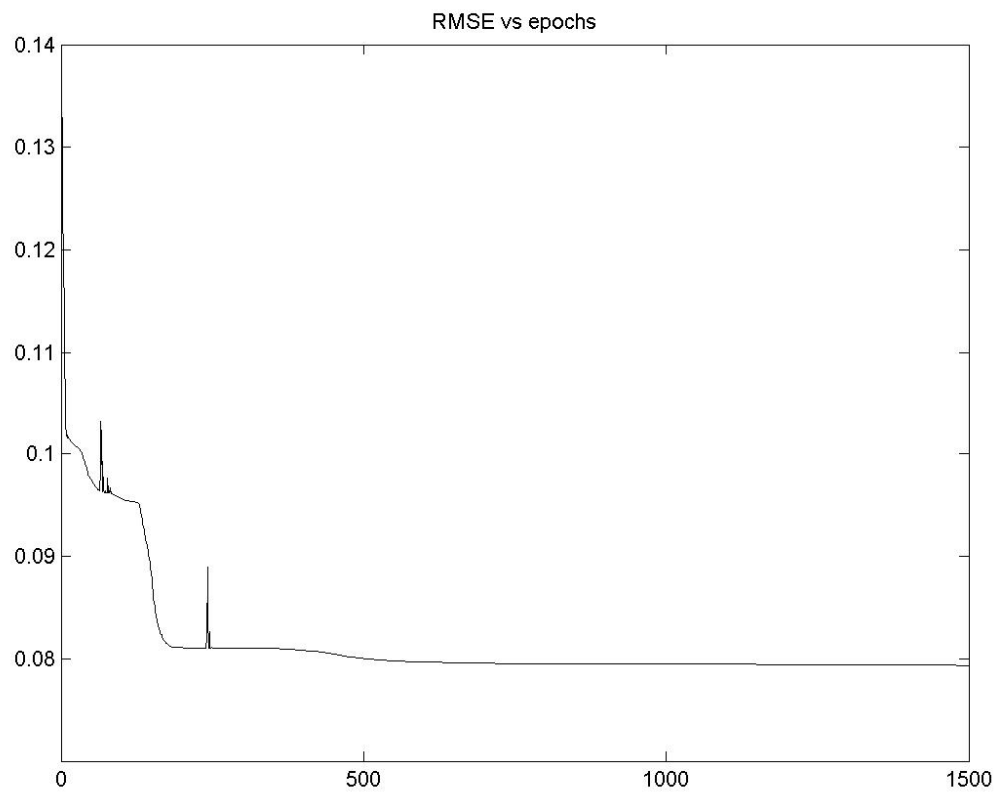
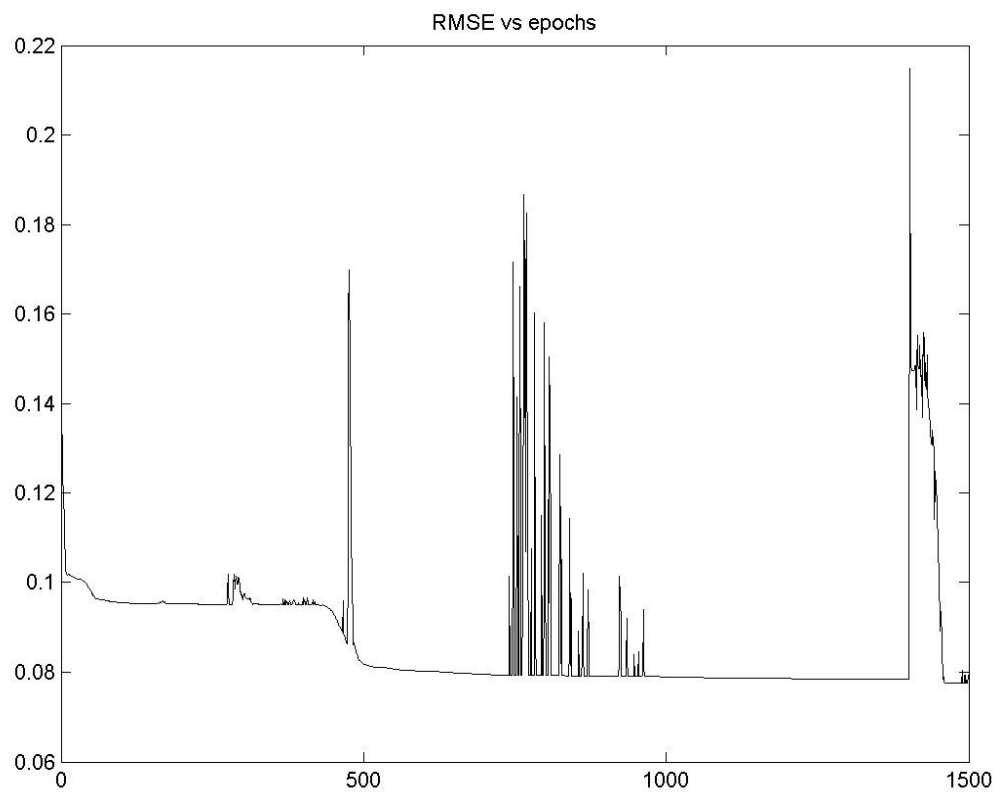
37^η Επανάληψη38^η Επανάληψη

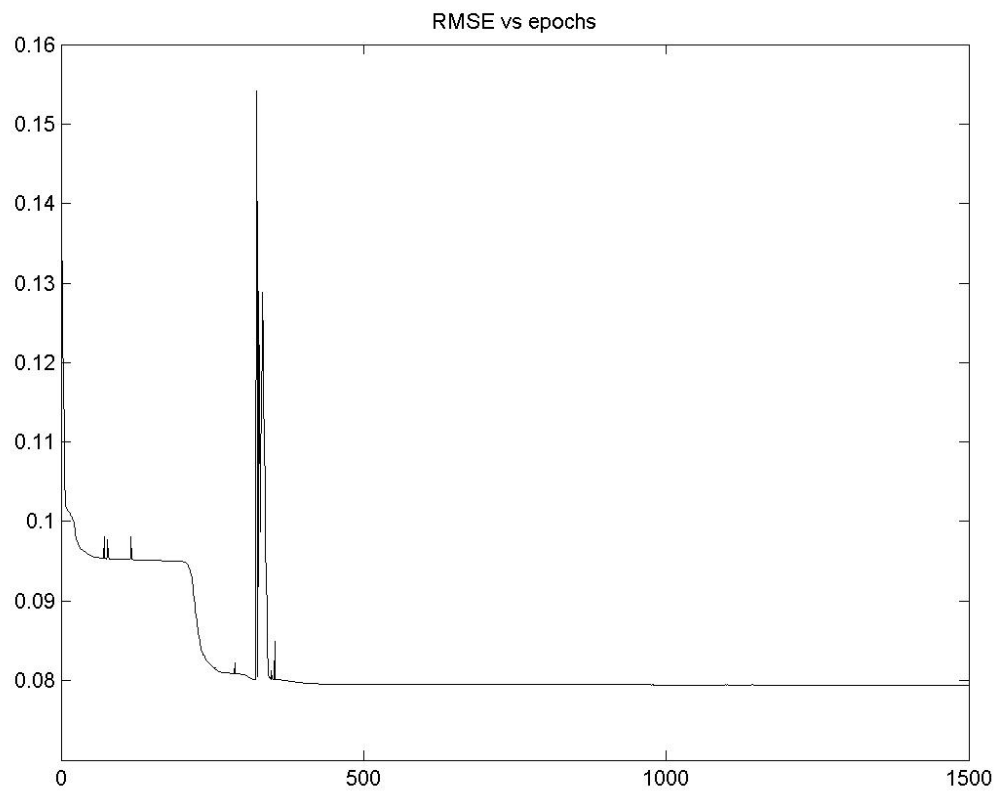
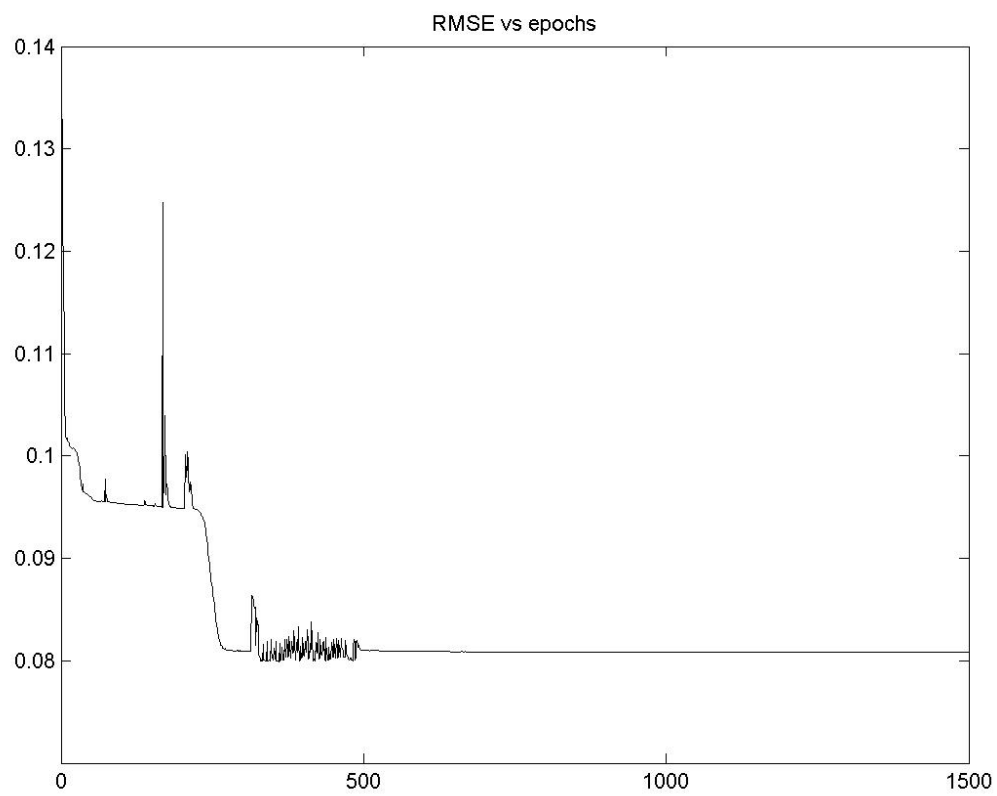
39^η Επανάληψη40^η Επανάληψη

41^η Επανάληψη42^η Επανάληψη

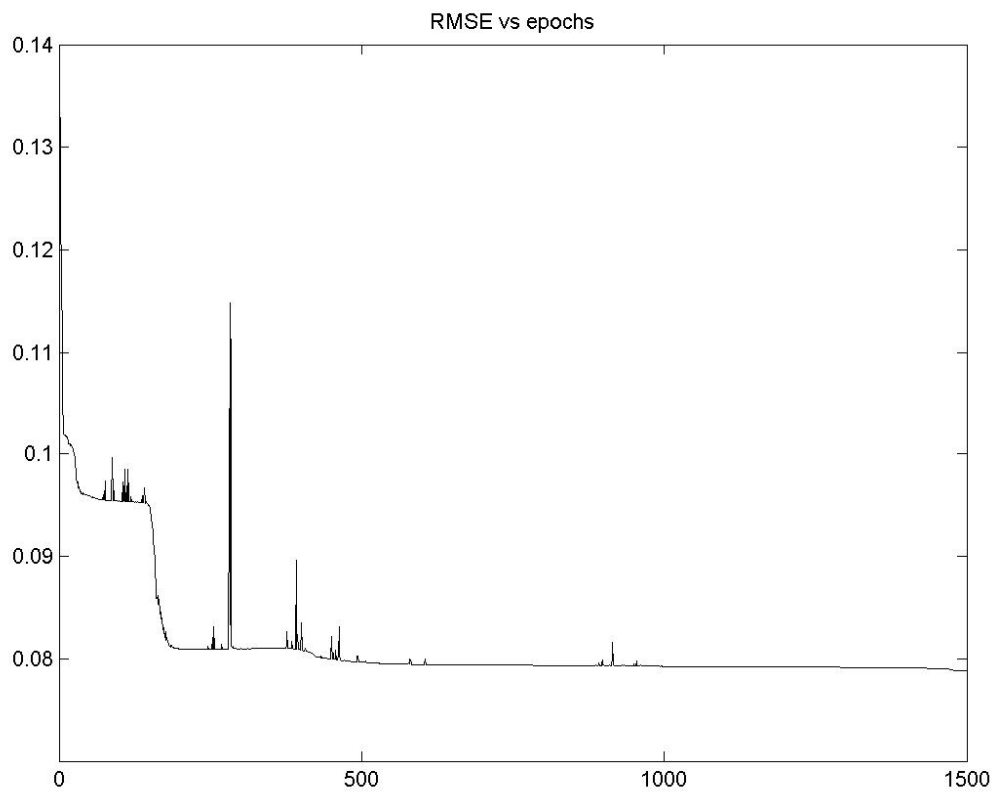
43^η Επανάληψη44^η Επανάληψη

45^η Επανάληψη46^η Επανάληψη

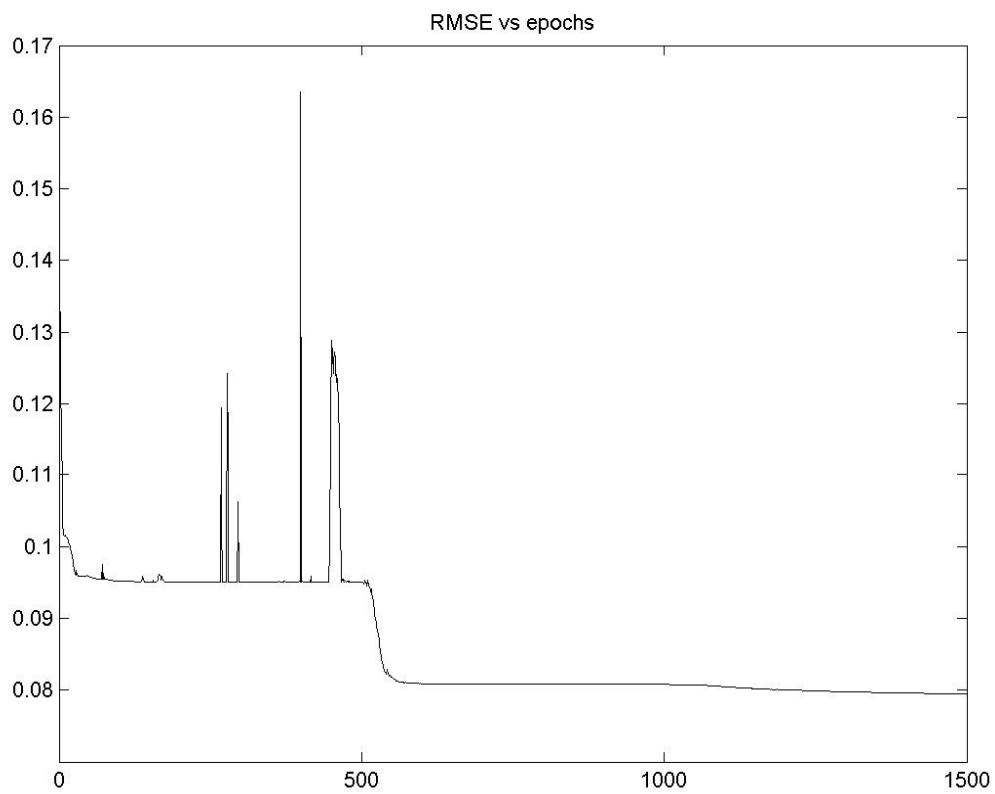
47^η Επανάληψη48^η Επανάληψη

49^η Επανάληψη50^η Επανάληψη

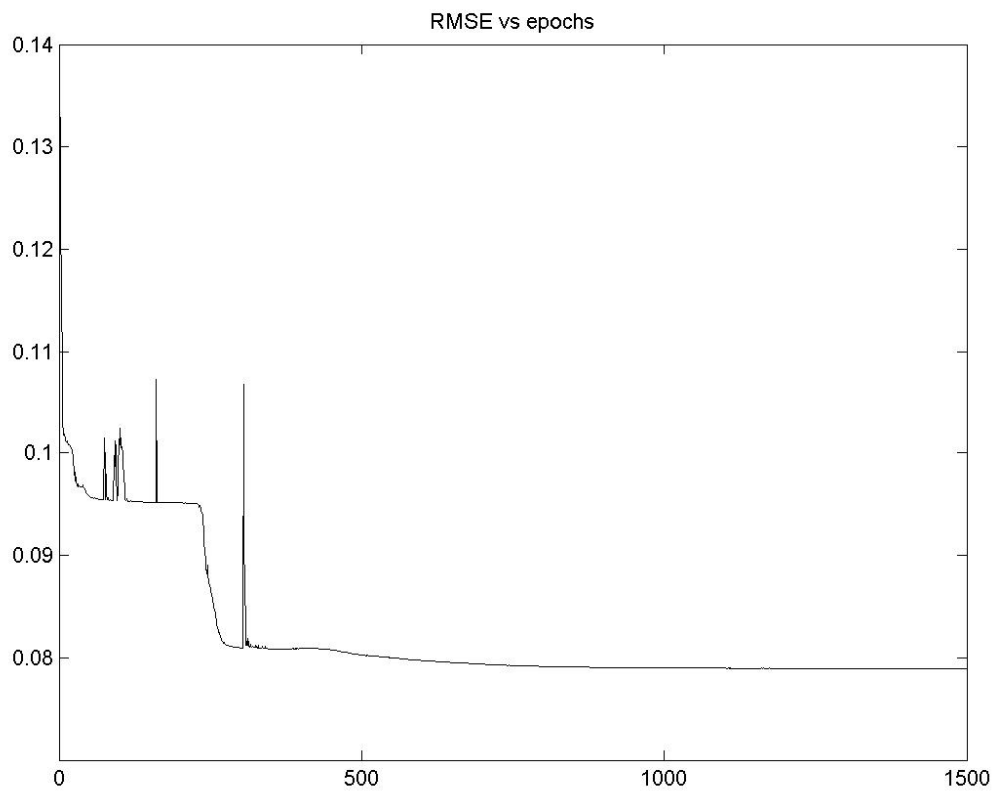
51^η Επανάληψη



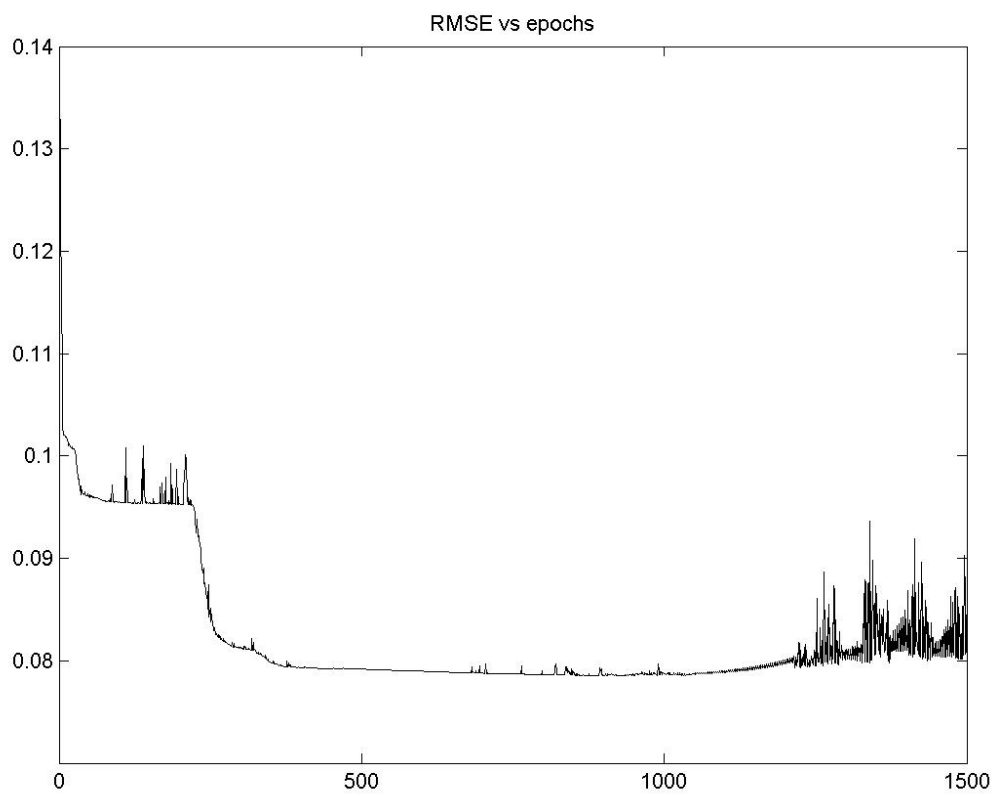
52^η Επανάληψη



53^η Επανάληψη



54^η Επανάληψη



ΠΑΡΑΡΤΗΜΑ

Η εκμάθηση του δικτύου αποσκοπεί στην επιλογή του $d\theta$ έτσι ώστε να μεγιστοποιηθεί το $d\Phi$ για δεδομένες τιμές των dE και $d\lambda$. Για να καθορισθούν οι αλλαγές που επιφέρουν στις Φ και E μικρές μεταβολές $d\theta$ του τρέχοντος διανύσματος ελέγχου, θεωρούνται οι

$$\Phi = \Phi(\theta) + (\lambda_\Phi)^T \cdot f(st, \theta) \quad (\text{A.1}\alpha)$$

$$E = E(st, \theta) + (\lambda_E)^T \cdot f(st, \theta) \quad (\text{A.1}\beta)$$

όπου λ_Φ^T , λ_E^T είναι πίνακες γραμμική που περιέχουν τους πολλαπλασιαστές Lagrange. Παραγωγίζοντας τις (A.1) έχουμε

$$d\Phi = \left(\frac{\partial \Phi}{\partial st} + (\lambda_\Phi)^T \cdot \frac{\partial f}{\partial st} \right) \cdot dst + \left(\frac{\partial \Phi}{\partial \theta} + (\lambda_\Phi)^T \cdot \frac{\partial f}{\partial \theta} \right) \cdot d\theta \quad (\text{A.2}\alpha)$$

$$dE = \left(\frac{\partial E}{\partial st} + (\lambda_E)^T \cdot \frac{\partial f}{\partial st} \right) \cdot dst + \left(\frac{\partial E}{\partial \theta} + (\lambda_E)^T \cdot \frac{\partial f}{\partial \theta} \right) \cdot d\theta \quad (\text{A.2}\beta)$$

Οι πολλαπλασιαστές Lagrange επιλέγοντες έτσι ώστε να εκλείψει η εξάρτηση του dst επί του dE , δηλαδή να μηδενισθεί ο συντελεστής του dst στην (A.2):

$$\frac{\partial \Phi}{\partial st} + (\lambda_\Phi)^T \cdot \frac{\partial f}{\partial st} = 0 \Rightarrow \lambda_\Phi = \mathbf{0} \quad (\text{A.3}\alpha)$$

$$\frac{\partial E}{\partial st} + (\lambda_E)^T \cdot \frac{\partial f}{\partial st} = 0 \quad (\text{A.3}\beta)$$

και οι (A.2) μετασχηματίζονται στις ακόλουθες σχέσεις:

$$d\Phi = \frac{\partial \Phi}{\partial \theta} \cdot d\theta \quad (\text{A.4}\alpha)$$

$$dE = \left(\frac{\partial E}{\partial \theta} + (\lambda_E)^T \cdot \frac{\partial f}{\partial \theta} \right) \cdot d\theta = (\lambda_E)^T \cdot \frac{\partial f}{\partial \theta} \cdot d\theta \quad (\text{A.4}\beta)$$

Οι πολλαπλασιαστές Lagrange καθορίζονται από τις αναδρομικές εξισώσεις:

$$\lambda_{E_l}^{(2)}(k) = \frac{1}{k_f} \cdot [y_l(k) - \hat{y}_l(k)] \quad (\text{A.5}\alpha)$$

$$\lambda_{E_{2i-1}}^{(1)}(k) = \frac{1}{k_f} \cdot \sum_{l=1}^r \left\{ c_{l,2i-1} \cdot f_b^{(l)}(k) \cdot [y_l(k) - \hat{y}_l(k)] \right\} + \sum_{l=1}^r \left\{ \lambda_{E_l}^{(2)}(k) \cdot c_{l,2i-1} \cdot f_b^{(l)}(k) \right\} +$$

$$+ \lambda_{E_{2i-1}}^{(1)}(k+1) \cdot w_i^{(1)} \cdot f_a^{(2i-1)}(k+1) - \lambda_{E_{2i}}^{(1)}(k+1) \cdot w_i^{(2)} \cdot f_a^{(2i)}(k+1) \quad (\text{A.5}\beta)$$

$$\lambda_{E_{2i}}^{(1)}(k) = \frac{1}{k_f} \cdot \sum_{l=1}^r \left\{ c_{l,2i} \cdot f_b^{(l)}(k) \cdot [y_l(k) - \hat{y}_l(k)] \right\} + \sum_{l=1}^r \left\{ \lambda_{E_l}^{(2)}(k) \cdot c_{l,2i} \cdot f_b^{(l)}(k) \right\} +$$

$$+ \lambda_{E_{2i-1}}^{(1)}(k+1) \cdot w_i^{(2)} \cdot f_a^{(2i-1)}(k+1) + \lambda_{E_{2i}}^{(1)}(k+1) \cdot w_i^{(1)} \cdot f_a^{(2i)}(k+1) \quad (\text{A.5}\gamma)$$

όπου $i=1, \dots, \frac{N}{2}, l=1, \dots, r$ και $f_b^{(l)}(k), f_1^{(2i-1)}(k+j), f_1^{(2i)}(k+j)$ είναι οι παράγωγοι των $y_j(k+l)$ και $x_{2i-1}(k+j), x_{2i}(k+j)$, αντίστοιχα, ως προς τα ορίσματά τους. Οι εξισώσεις (A.5) είναι αναδρομικές εξισώσεις διαφορών και επιλύονται για $k=k_f, k_f-1, \dots, 1$ με χρήση των ακόλουθων οριακών συνθηκών:

$$\lambda_{E_l}^{(2)}(k_f) = \frac{1}{k_f} \cdot [y_l(k_f) - \hat{y}_l(k_f)] \quad (\text{A.6}\alpha)$$

$$\lambda_{E_{2i-1}}^{(1)}(k_f) = \frac{1}{k_f} \cdot \sum_{l=1}^r \left\{ c_{l,2i-1} \cdot f_b^{(l)}(k_f) \cdot [y_l(k_f) - \hat{y}_l(k_f)] \right\} + \sum_{l=1}^r \left\{ \lambda_{E_l}^{(2)}(k_f) \cdot c_{l,2i-1} \cdot f_b^{(l)}(k_f) \right\} \quad (\text{A.6}\beta)$$

$$\lambda_{E_{2i}}^{(1)}(k_f) = \frac{1}{k_f} \cdot \sum_{l=1}^r \left\{ c_{l,2i} \cdot f_b^{(l)}(k_f) \cdot [y_l(k_f) - \hat{y}_l(k_f)] \right\} + \sum_{l=1}^r \left\{ \lambda_{E_l}^{(2)}(k_f) \cdot c_{l,2i} \cdot f_b^{(l)}(k_f) \right\} \quad (\text{A.6}\gamma)$$

Το διάνυσμα της ενημέρωσης των παραμέτρων κείται στην επιφάνεια ενός μικρού υπερελλειψοειδούς μέσα στο χώρο των παραμέτρων, οι άξονες του οποίου είναι οι Δ_i και το κέντρο του είναι το σημείο που καθορίζει το τρέχον διάνυσμα ελέγχου:

$$d\theta^T \cdot (\Delta^2)^{-1} \cdot d\theta = 1 \quad (\text{A.7})$$

Η εξίσωση (A.4α) και οι περιρισμοί που θέτουν οι εξισώσεις (A.4β) και (A.7) πρέπει να πληρούνται σε κάθε επανάληψη. Κατά συνέπεια, θεωρώντας ένα γραμμικό συνδυασμό των (A.4) και (A.7), έπεται ότι

$$d\Phi = \frac{\partial \Phi}{\partial \theta} \cdot d\theta + \nu \cdot \left[dE - (\lambda_E)^T \cdot \frac{\partial f}{\partial \theta} \cdot d\theta \right] + \mu \cdot \left[1 - d\theta^T \cdot (\Delta^2)^{-1} \cdot d\theta \right] \quad (\text{A.8})$$

Λαμβάνοντας το δεύτερο διαφορικό της (A.8) προκύπτει:

$$d^2\Phi = \left[\frac{\partial^2 \Phi}{\partial \theta^2} - \nu \cdot (\lambda_E)^T \cdot \frac{\partial^2 f}{\partial \theta^2} - 2 \cdot \mu \cdot d\theta \cdot (\Delta^2)^{-1} \right] \cdot d^2\theta \quad (\text{A.9})$$

Έτσι το μέγιστο του $d\Phi$ λαμβάνει χώρα όταν ο συντελεστής του $d^2\theta$ στην (A.9) μηδενισθεί, όταν δηλαδή

$$d\theta \cdot (\Delta^2)^{-1} = \frac{\frac{\partial^2 \Phi}{\partial \theta^2} - \nu \cdot (\lambda_E)^T \cdot \frac{\partial^2 f}{\partial \theta^2}}{2 \cdot \mu} \quad (\text{A.10})$$

Ορίζοντας τις βοηθητικές μεταβλητές:

$$\Lambda_\Phi = \left(\frac{\partial^2 \Phi}{\partial \theta^2} \right)^T \quad (\text{A.11}\alpha) \quad \text{και} \quad \Lambda_E = \left((\lambda_E)^T \cdot \frac{\partial^2 f}{\partial \theta^2} \right)^T \quad (\text{A.11}\beta)$$

η (A.10) γίνεται:

$$d\theta \cdot (\Delta^2)^{-1} = \frac{\Lambda_\Phi - \nu \cdot \Lambda_E}{2 \cdot \mu} \quad (\text{A.12})$$

Μετά τη διεξαγωγή των υπολογισμών στις (A.12), οι βοηθητικές μεταβλητές δίνονται από τις ακόλουθες εξισώσεις:

$$\Lambda_{E_{l,i}}^{(c)} = \sum_{k=1}^{k_f} \left\{ \lambda_{E_l}^{(2)}(k) \cdot x_i(k) \cdot f_b^{(l)}(k) \right\} \quad l=1, \dots, r, \quad i=1, \dots, N \quad (\text{A.13}\alpha)$$

$$A_{E_{i,j}}^{(b)} = \sum_{k=1}^{k_f} \left\{ \lambda_{E_i}^{(1)}(k) \cdot u_j(k) \cdot f_a'^{(i)}(k) \right\} \quad i=1,\dots,N, \quad j=1,\dots,m \quad (\text{A.13}\beta)$$

$$A_{E_i}^{(w^{(1)})} = \sum_{k=1}^{k_f} \left\{ \lambda_{E_{2i-1}}^{(1)}(k) \cdot x_{2i-1}(k-1) \cdot f_a'^{(2i-1)}(k) + \lambda_{E_{2i}}^{(1)}(k) \cdot x_{2i}(k-1) \cdot f_a'^{(2i)}(k) \right\} \quad i=1,\dots,\frac{N}{2} \quad (\text{A.13}\gamma)$$

$$A_{E_i}^{(w^{(2)})} = \sum_{k=1}^{k_f} \left\{ \lambda_{E_{2i-1}}^{(1)}(k) \cdot x_{2i}(k-1) \cdot f_a'^{(2i-1)}(k) - \lambda_{E_{2i}}^{(1)}(k) \cdot x_{2i-1}(k-1) \cdot f_a'^{(2i)}(k) \right\} \quad i=1,\dots,\frac{N}{2} \quad (\text{A.13}\delta)$$

$$A_{\Phi_{l,i}}^{(c)} = \begin{cases} 2(c_{l,i}|_{cur} - c_{l,i}|_{prev}) & \text{ΕΑΝ ΟΙ ΙΔΙΟΤΙΜΕΣ ΚΕΙΝΤΑΙ ΕΝΤΟΣ ΤΟΥ ΜΟΝΑΔΙΑΙΟΥ ΚΥΚΛΟΥ} \\ 0 & \text{ΣΤΗΝ ΑΝΤΙΘΕΤΗ ΠΕΡΙΠΤΩΣΗ} \end{cases}$$

$l=1,\dots,r, \quad i=1,\dots,N \quad (\text{A.14}\alpha)$

$$A_{\Phi_{i,j}}^{(b)} = \begin{cases} 2(b_{i,j}|_{cur} - b_{i,j}|_{prev}) & \text{ΕΑΝ ΟΙ ΙΔΙΟΤΙΜΕΣ ΚΕΙΝΤΑΙ ΕΝΤΟΣ ΤΟΥ ΜΟΝΑΔΙΑΙΟΥ ΚΥΚΛΟΥ} \\ 0 & \text{ΣΤΗΝ ΑΝΤΙΘΕΤΗ ΠΕΡΙΠΤΩΣΗ} \end{cases}$$

$i=1,\dots,N, \quad j=1,\dots,m \quad (\text{A.14}\beta)$

$$A_{\Phi_i}^{(w^{(1)})} = \begin{cases} 2(w_i^{(1)}|_{cur} - w_i^{(1)}|_{prev}) & \text{ΕΑΝ ΟΙ ΙΔΙΟΤΙΜΕΣ ΚΕΙΝΤΑΙ ΕΝΤΟΣ ΤΟΥ ΜΟΝΑΔΙΑΙΟΥ ΚΥΚΛΟΥ} \\ \left\{ 1 + f_d[(w_i^{(1)})^2 + (w_i^{(2)})^2 - 1] \right\} \cdot f_d'[(w_i^{(1)})^2 + (w_i^{(2)})^2 - 1] \cdot 2 \cdot w_i^{(1)} & \text{ΣΤΗΝ ΑΝΤΙΘΕΤΗ ΠΕΡΙΠΤΩΣΗ} \end{cases}$$

$i=1,\dots,\frac{N}{2} \quad (\text{A.14}\gamma)$

$$A_{\Phi_i}^{(w^{(2)})} = \begin{cases} 2(w_i^{(2)}|_{cur} - w_i^{(2)}|_{prev}) & \text{ΕΑΝ ΟΙ ΙΔΙΟΤΙΜΕΣ ΚΕΙΝΤΑΙ ΕΝΤΟΣ ΤΟΥ ΜΟΝΑΔΙΑΙΟΥ ΚΥΚΛΟΥ} \\ \left\{ 1 + f_d[(w_i^{(1)})^2 + (w_i^{(2)})^2 - 1] \right\} \cdot f_d'[(w_i^{(1)})^2 + (w_i^{(2)})^2 - 1] \cdot 2 \cdot w_i^{(2)} & \text{ΣΤΗΝ ΑΝΤΙΘΕΤΗ ΠΕΡΙΠΤΩΣΗ} \end{cases}$$

$i=1,\dots,\frac{N}{2} \quad (\text{A.14}\delta)$

Οι αλλαγές στις μεταβλητές ελέγχου δίνονται από τις (20), όπου οι I_{EE} , $I_{E\Phi}$, $I_{\Phi\Phi}$ και δE υπολογίζονται από τις (22) και (23), αντίστοιχα.

Βιβλιογραφία

A. Αναφορικά με το μοντέλο, τον αλγόριθμο εκμάθησης και τις εφαρμογές

- [A1] A. Cohen, "Signal processing methods for upper airway and pulmonary dysfunction diagnosis," *IEEE Engineering in Medicine and Biology Magazine*, pp. 72-75, 1990.
- [A2] Λ.Ι. Χατζηλεοντιάδης, *Ανάλυση και επεξεργασία των αναπνευστικών ήχων με τη χρήση της στατιστικής - φάσματος ανώτερης τάξης και του μετασχηματισμού κυματιδίων*, Διδακτορική διατριβή, Αριστοτέλειο Πανεπιστήμιο, Θεσσαλονίκη 1997.
- [A3] L.J. Hadjileontiadis and S.M. Panas, "Separation of Discontinuous Adventitious Sounds from Vesicular Sounds Using a Wavelet-Based Filter," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 12, pp. 1269-1281, 1997.
- [A4] S. S. Kraman, *Lung Sounds: An introduction to the interpretation of the auscultatory finding*, Northbrook, IL: Amer. College of Chest Physicians, 1993, audio tape.
- [A5] S. Lehrer, *Understanding Lung Sounds*. Philadelphia, PA: Saunders, 1993, workbook.
- [A6] S. Lehrer, *Understanding Lung Sounds*. Philadelphia, PA: Saunders, 1993, audio tape.
- [A7] P. Mastorocostas, J. Theocharis, "On Stable Learning of Block-Diagonal Recurrent Neural Networks - Part I: The RENNCOM Algorithm," *Proceedings of 2004 IEEE International Joint Conference on Neural Networks*, Budapest, Hungary, July 2004, pp. 815-820.
- [A8] P. Mastorocostas, J. Theocharis, "On Stable Learning of Block-Diagonal Recurrent Neural Networks - Part II: Application to the Analysis of Lung Sounds," *Proceedings of 2004 IEEE International Joint Conference on Neural Networks*, Budapest, Hungary, July 2004, pp. 821-826.
- [A9] P. Mastorocostas, J. Theocharis, "A Stable Learning Method for Block-Diagonal Recurrent Neural Networks: Application to the Analysis of Lung Sounds," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no 2, pp. 242-254, April 2006.
- [A10] R.L. Murphy, Jr., S.K. Holford, and W.C. Knowler, "Visual lung sound characterization by time-expanded waveform analysis," *New England Journal of Medicine*, vol. 296, pp. 968-971, 1977.
- [A11] A.G. Tilikian and M.B. Conover, *Understanding Heart Sounds and Murmurs with an Introduction to Lung Sounds*. Philadelphia, PA: Saunders, 1993, audio tape.

B. Αναφορικά με τη σχεδίαση και ανάπτυξη του λογισμικού

- [B1] *MATLAB Creating Graphical User Interfaces* © COPYRIGHT 2000 - 2005 by The MathWorks, Inc.
- [B2] Δημήτριος Βαρσάμης, *Matlab – Graphical User Interfaces Σημειώσεις GUI για το εργαστήριο του μαθήματος Ασαφή Συστήματα*, Σέρρες 2004