



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΗΣ ΙΣΧΥΟΣ

ΒΕΛΤΙΣΤΗ ΡΥΘΜΙΣΗ ΗΛΕΚΤΡΟΝΟΜΩΝ ΥΠΕΡΕΝΤΑΣΗΣ - ΚΑΤΕΥΘΥΝΣΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΜΑΡΙΑΣ Ν. ΚΑΛΥΒΙΩΤΗ

Επιβλέπων καθηγητής :

Γεώργιος Κορρές
Αναπληρωτής Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2008



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΗΣ ΙΣΧΥΟΣ

**ΒΕΛΤΙΣΤΗ ΡΥΘΜΙΣΗ ΗΛΕΚΤΡΟΝΟΜΩΝ
ΥΠΕΡΕΝΤΑΣΗΣ - ΚΑΤΕΥΘΥΝΣΗΣ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
της
ΜΑΡΙΑΣ Ν. ΚΑΛΥΒΙΩΤΗ

Επιβλέπων καθηγητής : Γεώργιος Κορρές
Αναπληρωτής Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30^η Οκτωβρίου 2008.

.....
Γεώργιος Κορρές
Αναπληρωτής Καθηγητής
ΕΜΠ

.....
Κωνσταντίνος Βουρνάς
Καθηγητής
ΕΜΠ

.....
Σταύρος Παπαθανασίου
Επίκουρος Καθηγητής
ΕΜΠ

Αθήνα, Οκτώβριος 2008

.....
ΚΑΛΥΒΙΩΤΗ Ν. ΜΑΡΙΑ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Καλυβιώτη Ν. Μαρία, 2008

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το γράφοντα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Πρόλογος

Θα ήθελα να ευχαριστήσω θερμά τον κ. Γεώργιο Κορρέ, αναπληρωτή καθηγητή του τομέα Ηλεκτρικής Ισχύος του Εθνικού Μετσόβιου Πολυτεχνείου, για την συμπαράστασή του και την συνδρομή του στην εκπόνηση αυτής της εργασίας.

Επιπλέον θεωρώ υποχρέωσή μου να ευχαριστήσω τον υποψήφιο Διδάκτορα κ. Χρήστο Αποστολόπουλο για την πολύτιμη καθοδήγηση που μου παρείχε καθ' όλη τη διάρκεια της διπλωματικής εργασίας.

Αθήνα, 30 Οκτωβρίου 2008

Μαρία Καλυβιώτη

Περίληψη

Η ρύθμιση των στοιχείων προστασίας ενός Συστήματος Ηλεκτρικής Ενέργειας είναι ζωτικής σημασίας για την εύρυθμη λειτουργία του συστήματος. Στη διπλωματική αυτή εργασία πραγματοποιείται η ρύθμιση των ηλεκτρονόμων υπερέντασης με στοιχεία κατεύθυνσης, κάνοντας χρήση εναλλακτικών μεθόδων. Οι υπολογισμοί που προηγούνται της επιλογής και ρύθμισης των στοιχείων προστασίας είναι η ανάλυση ροών φορτίου, η ανάλυση βραχυκυκλωμάτων και η τοπολογική ανάλυση του συστήματος. Η τελευταία περιλαμβάνει την εύρεση των βρόχων του συστήματος, των Break Points, του Διανύσματος Σχετικής Ακολουθίας (*RSV*) και του Συνόλου Διαδοχικών Ζευγών (*SSP*). Για την εύρεση των Break Points και του διανύσματος *RSV* χρησιμοποιούνται δυο εναλλακτικές μέθοδοι. Οι διαδικασίες αυτές υλοποιούνται και τα αποτελέσματά τους χρησιμοποιούνται για τη ρύθμιση των ηλεκτρονόμων υπερέντασης – κατεύθυνσης και των μετασχηματιστών έντασης. Συγκεκριμένα υπολογίζονται ο λόγος μετασχηματισμού (*CTR*) των μετασχηματιστών έντασης, το ρεύμα ρύθμισης (I_{pick_up}) και η χρονική σταθερά (*TDS*) των ηλεκτρονόμων. Η σταθερά *TDS* των ηλεκτρονόμων υπολογίζεται κάνοντας χρήση δυο εναλλακτικών μεθόδων, της μεθόδου τοπολογικής ανάλυσης και της μεθόδου γραμμικού προγραμματισμού. Η πρώτη από αυτές τις μεθόδους χρησιμοποιεί το Σύνολο Διαδοχικών Ζευγών (*SSP*), ενώ για την υλοποίηση της δεύτερης χρησιμοποιήθηκε μια βελτιωμένη διαμόρφωση του προβλήματος που οδηγεί σε μείωση της διάστασής του. Τέλος, πραγματοποιείται εφαρμογή των χρησιμοποιούμενων μεθοδολογιών σε ένα πρότυπο σύστημα των 14 ζυγών και εξάγονται τα απαραίτητα συμπεράσματα.

Λέξεις κλειδιά: < Ηλεκτρονόμοι υπερέντασης – κατεύθυνσης, συνεργασία ηλεκτρονόμων, τοπολογική ανάλυση συστήματος, Break Points, *RSV*, *SSP*, ρεύμα ρύθμισης, χρονική σταθερά *TDS*, μέθοδος τοπολογικής ανάλυσης, γραμμικός προγραμματισμός >

Abstract

The coordination of the protection devices of a Power System is of tremendous importance for the proper operation of the system. In this thesis coordination of directional overcurrent relays using alternative methods is fulfilled. The computations that proceed the protection coordination are the Power Flow Analysis, the Fault Analysis and the Topological Analysis of the system. The last one is the enumeration of the loops of the system, Break Points Set, Relative Sequence Vector (RSV) and Set of Sequential Pairs (SSP). Two different methods are used for the enumeration of Break Points Set and Vector RSV . These procedures are completed and their results are used for the coordination of directional overcurrent relays and their current transformers. In particular, the current transformer ratio (CTR) of the current transformers, the pick up current (I_{pick_up}) and the time dial setting (TDS) of the relays are calculated. The TDS of the relays are calculated using two alternative methods, the topological analysis method and the linear programming method. The first method uses the Set of Sequential Pairs (SSP), while for the implementation of the second method an improved formulation of the problem that leads to dimensionality reduction is used. At last, the described techniques are implemented on a 14-bus test system and the essential conclusions are drawn.

Key words: < Directional overcurrent relays, protection coordination, topological analysis of the system, Break Points, RSV, SSP, pick up current, time dial setting TDS, topological analysis method, linear programming >

Πίνακας Περιεχομένων

1	Εισαγωγή.....	1
1.1	Προστασία Συστημάτων Ηλεκτρικής Ενέργειας	1
1.2	Αντικείμενο διπλωματικής	1
1.3	Δομή διπλωματικής	1
1.4	Οργάνωση κειμένου	2
2	Βασικές έννοιες	3
2.1	Προστασία	3
2.2	Προστασία Υπερέντασης - Κατεύθυνσης	3
2.2.1	Ηλεκτρονόμοι υπερέντασης – κατεύθυνσης	3
2.2.2	Ζώνες προστασίας.....	5
2.3	Ανάλυση ροών φορτίου.....	7
2.3.1	Μεταβλητές συστήματος	7
2.3.2	Χαρακτηρισμός ζυγών	8
2.3.3	Εξισώσεις ροών φορτίου	8
2.4	Ανάλυση βραχυκυκλωμάτων.....	10
2.4.1	Βραχυκυκλώματα	10
2.4.2	Ασύμμετρα τριφασικά συστήματα.....	11
2.5	Τοπολογική ανάλυση συστήματος.....	11
2.6	Επιλογή - ρύθμιση εξοπλισμού προστασίας.....	13
3	Άλλες τεχνικές.....	15
3.1	Ανάλυση ροών φορτίου.....	15
3.1.1	Μέθοδος Gauss – Siedel	15
3.2	Ανάλυση βραχυκυκλωμάτων.....	17
3.2.1	Μέθοδος στο πεδίο των ακολουθιακών κυκλωμάτων	17
3.2.2	Μέθοδοι στο πεδίο των φάσεων.....	17
3.3	Τοπολογική ανάλυση συστήματος.....	19
3.3.1	Break Points.....	19
3.3.2	RSM	19
3.4	Ρύθμιση των ηλεκτρονόμων	20
3.4.1	Μέθοδος δοκιμής και λάθους	20
3.4.2	Μέθοδος τοπολογικής ανάλυσης	21
3.4.3	Μέθοδος βελτιστοποίησης	21
4	Χρησιμοποιούμενες τεχνικές	27
4.1	Ανάλυση ροών φορτίου.....	27
4.1.1	Μέθοδος Newton – Raphson	27
4.1.2	Υπολογισμός ρευμάτων.....	30
4.2	Ανάλυση βραχυκυκλωμάτων.....	30
4.2.1	Τριφασικός πίνακας αγωγιμοτήτων Y	30
4.2.2	Υπολογισμός τριφασικών σφαλμάτων	34
4.3	Τοπολογική ανάλυση συστήματος.....	36
4.3.1	Βρόχοι συστήματος	36
4.3.2	Break Points.....	37

4.3.3	<i>RSV</i>	39
4.3.4	<i>SSP</i>	40
4.4	Επιλογή - ρύθμιση εξοπλισμού προστασίας	41
4.4.1	Επιλογή των μετασχηματιστών εντάσεως (<i>CT</i>)	41
4.4.2	Ρύθμιση των <i>H/N</i> υπερέντασης	41
5	Υλοποίηση των χρησιμοποιούμενων τεχνικών	47
5.1	Ανάλυση ροών φορτίου	48
5.1.1	Αρχείο εισόδου	48
5.1.2	Αλγόριθμος	49
5.2	Ανάλυση βραχυκυκλωμάτων	50
5.2.1	Αρχεία εισόδου	51
5.2.2	Αλγόριθμος	52
5.3	Τοπολογική ανάλυση συστήματος	57
5.3.1	Βρόχοι του συστήματος	58
5.3.2	<i>Break Points</i>	62
5.3.3	<i>RSV</i>	66
5.3.4	<i>SSP</i>	68
5.4	Επιλογή - ρύθμιση εξοπλισμού προστασίας	70
5.4.1	Επιλογή του λόγου των μετασχηματιστών εντάσεως (<i>CTR</i>) και του ρεύματος ρύθμισης των ηλεκτρονόμων ($I_{pick-up}$)	70
5.4.2	Χρονική σταθερά (<i>TDS</i>)	73
6	Εφαρμογή	81
6.1	Ανάλυση ροών φορτίου και ανάλυση βραχυκυκλωμάτων	82
6.2	Τοπολογική ανάλυση συστήματος	84
6.3	Επιλογή - ρύθμιση εξοπλισμού προστασίας	87
6.3.1	Επιλογή του λόγου των μετασχηματιστών εντάσεως (<i>CTR</i>) και του ρεύματος ρύθμισης των ηλεκτρονόμων ($I_{pick-up}$)	87
6.3.2	Χρονική σταθερά (<i>TDS</i>)	89
7	Αξιολόγηση – Σύγκριση αποτελεσμάτων	93
7.1	Αξιολόγηση αποτελεσμάτων	93
7.2	Σύγκριση εναλλακτικών μεθόδων	97
7.2.1	Για εύρεση των <i>Break Points</i> , <i>RSV</i>	97
7.2.2	Για εύρεση των <i>TDS</i>	97
7.3	Συμπεράσματα	98
8	Βιβλιογραφία	99
	Παράρτημα	101
	Προγράμματα	101
	Αρχεία εισόδου	131

1 Εισαγωγή

1.1 Προστασία Συστημάτων Ηλεκτρικής Ενέργειας

Η λειτουργία των Συστημάτων Ηλεκτρικής Ενέργειας δεν μπορεί να εξασφαλιστεί χωρίς την ύπαρξη κατάλληλης προστασίας, η οποία παρέχεται από τα στοιχεία προστασίας. Η σωστή ρύθμιση των στοιχείων αυτών εξασφαλίζει την επιθυμητή λειτουργία τους στην περίπτωση εκδήλωσης βραχυκυκλώματος και συνεπώς στον περιορισμό των συνεπειών του. Η βέλτιστη επιλογή και ρύθμιση των στοιχείων προστασίας είναι λοιπόν εξαιρετικής σημασίας.

1.2 Αντικείμενο διπλωματικής

Αντικείμενο της διπλωματικής αυτής είναι η ρύθμιση Ηλεκτρονόμενων Υπερέντασης-Κατεύθυνσης χρησιμοποιώντας δυο εναλλακτικές τεχνικές, τη μέθοδο της τοπολογικής ανάλυσης και τη μέθοδο του γραμμικού προγραμματισμού. Σκοπός είναι να ρυθμιστούν με αποτελεσματικό τρόπο όλες οι παράμετροι των ηλεκτρονόμενων ώστε το σύστημα προστασίας να χαρακτηρίζεται από αξιοπιστία, επιλογικότητα και ασφαλή ταχύτητα λειτουργίας.

1.3 Δομή διπλωματικής

Για να επιτευχθεί ο σκοπός της διπλωματικής, δηλαδή να ρυθμιστούν οι Ηλεκτρονόμοι Υπερέντασης-Κατεύθυνσης, θα πρέπει πρωτίστως να εκτελεστούν και κάποιες άλλες εργασίες. Οι επιμέρους λοιπόν εργασίες που απαιτούνται για την ολοκλήρωση της διαδικασίας ρύθμισης είναι οι ακόλουθες:

1. Ανάλυση Ροών Φορτίου
2. Ανάλυση Βραχυκυκλωμάτων
3. Τοπολογική ανάλυση συστήματος
4. Ρύθμιση των ηλεκτρονόμενων

Το τελευταίο από τα επιμέρους αυτά τμήματα χρησιμοποιεί στις διαδικασίες του και τα αποτελέσματα των προηγούμενων τμημάτων. Έτσι, χωρίζεται μεν η εργασία σε επιμέρους τμήματα, υπάρχει όμως μια αλληλεξάρτηση μεταξύ των τμημάτων αυτών.

1.4 Οργάνωση κειμένου

Η διπλωματική εργασία έχει την ακόλουθη δομή:

Στο κεφάλαιο 2 αναφέρονται έννοιες που σχετίζονται με το αντικείμενο της διπλωματικής και δίνεται μια περιγραφή ή εξήγηση καθεμιάς από αυτές. Στο κεφάλαιο 3 παρατίθενται υπάρχουσες μέθοδοι που επιλύουν καθένα από τα προαναφερθέντα επιμέρους τμήματα της διπλωματικής. Οι μέθοδοι αυτές δεν πρόκειται να χρησιμοποιηθούν στη συνέχεια. Στο τέταρτο κεφάλαιο παρουσιάζονται οι τεχνικές που πρόκειται να χρησιμοποιηθούν στη διπλωματική αυτή για την ολοκλήρωση καθενός από τα τμήματά της. Η παρουσίαση είναι περιγραφική ή θεωρητική. Στο κεφάλαιο 5 παρουσιάζεται αναλυτικά η υλοποίηση των μεθόδων αυτών. Εξηγείται η δομή των αλγόριθμων και παρουσιάζονται λεπτομέρειες σχετικά με το αντίστοιχο πρόγραμμα που χρησιμοποιήθηκε. Στο κεφάλαιο 6 γίνεται εφαρμογή των αλγορίθμων αυτών σε ένα πρότυπο σύστημα και λαμβάνονται τα αντίστοιχα αποτελέσματα ρύθμισης των ηλεκτρονικών υπερέντασης. Στο έβδομο κεφάλαιο γίνονται συγκρίσεις μεταξύ των αποτελεσμάτων και εξάγονται τα απαραίτητα συμπεράσματα. Τέλος, στο κεφάλαιο 8 παρατίθεται η χρησιμοποιούμενη βιβλιογραφία και στο παράρτημα παρατίθενται όλοι κώδικες που αποτελούν την υλοποίηση σε Matlab της μεθοδολογίας που χρησιμοποιήθηκε.

2

Βασικές έννοιες

Στο κεφάλαιο αυτό θα δοθεί ο ορισμός και μια σύντομη ερμηνεία των εννοιών που χρησιμοποιούνται σε αυτή τη διπλωματική εργασία. Θα εξηγηθούν δηλαδή οι έννοιες τις οποίες θα υπολογίσουμε στη συνέχεια.

2.1 Προστασία

Η ανάγκη της προστασίας των Συστημάτων Ηλεκτρικής Ενέργειας διαπιστώθηκε από τα πρώτα χρόνια της ανάπτυξής τους και παρουσιάζει και σήμερα συνεχή εξέλιξη. Ιδιαίτερη σημασία έχει η ορθή επιλογή και ρύθμιση των Ηλεκτρονόμων (H/N-relays), οι οποίοι μαζί με τους Διακόπτες Ισχύος (Δ/Ι, Circuit Breakers- CB) αποτελούν τα μέσα προστασίας. Ο ρόλος της προστασίας ενός ηλεκτρικού συστήματος συνίσταται στο να προλάβει μια βλάβη ή, εάν για οποιοδήποτε λόγο συμβεί ένα σφάλμα, να περιορίσει κατά το δυνατόν περισσότερο τις επιπτώσεις του στο υπόλοιπό σύστημα.

2.2 Προστασία Υπερέντασης - Κατεύθυνσης

2.2.1 Ηλεκτρονόμοι υπερέντασης – κατεύθυνσης

Η προστασία υπερεντάσεως είναι η απλούστερη και οικονομικότερη προστασία, η οποία και εφαρμόζεται ευρύτατα είτε ως πρωτεύουσα προστασία σε δίκτυα διανομής και συστήματα υπομεταφοράς είτε ως δευτερεύουσα προστασία σε συστήματα μεταφοράς. Συγκεκριμένα, χρησιμοποιείται εκτεταμένα στις Εσωτερικές Ηλεκτρικές Εγκαταστάσεις και τα Δίκτυα Διανομής XT αλλά και MT.

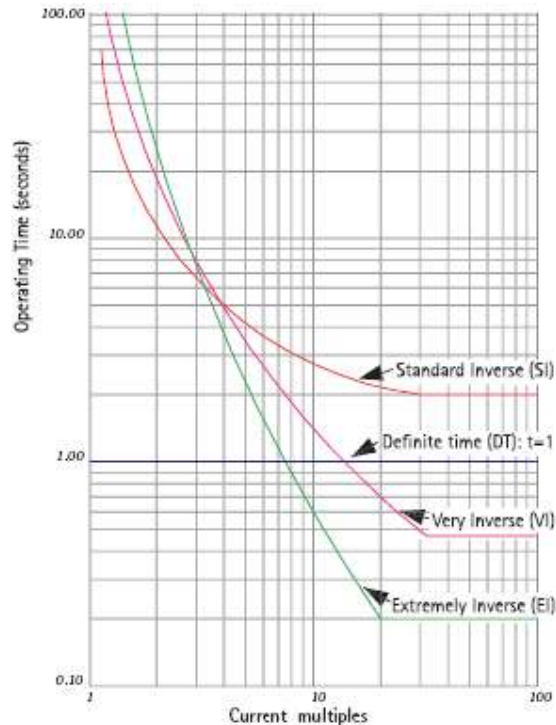
Ο ηλεκτρονόμος υπερεντάσεως είναι μια συσκευή που μετράει το ρεύμα που τη διαρρέει (υποβιβασμένο πάντα μέσω του Μ/Σ εντάσεως στο επίπεδο λειτουργίας του) και το συγκρίνει με μια τιμή ρεύματος αναφοράς ή ρύθμισης ή κατωφλιού (Pick up Current). Αν το ρεύμα αυτό είναι μεγαλύτερο από το ρεύμα αναφοράς, όπως στην περίπτωση του ρεύματος βραχυκύκλωσης, τότε λειτουργεί αποστέλλοντας ένα σήμα στο διακόπτη ισχύος με τον οποίο επικοινωνεί, ώστε να αποκόψει το τμήμα του δικτύου που έχει υποστεί βλάβη. Η ενέργεια αυτή θα έχει ως αποτέλεσμα τη συνέχιση της ομαλής τροφοδότησης των υπολοίπων φορτίων του δικτύου. Εκτός από το ρεύμα ρύθμισης θα πρέπει να οριστεί και η χρονική σταθερά (Time Dial Setting- TDS), που καθορίζει το χρόνο λειτουργίας του ηλεκτρονόμου για κάθε τιμή του ρεύματος.

Ανάλογα με τη σχέση χρόνου – έντασης οι H/N υπερέντασης διακρίνονται σε:

1. «Σταθερού χρόνου» (Definite Time), οι οποίοι ανεξάρτητα από το μέγεθος της υπερεντάσεως λειτουργούν στον ίδιο χρόνο που καθορίζεται από τη ρύθμισή τους.

2. «Αντιστρόφου χρόνου», στους οποίους ο χρόνος λειτουργίας μειώνεται όσο η υπερένταση αυξάνεται. Σύμφωνα με τις υφιστάμενες τυποποιήσεις, ανάλογα με το ρυθμό μείωσης του χρόνου διακρίνονται τρεις τύποι ηλεκτρονόμων:
- Οι απλώς αντιστρόφου χρόνου (Standard Inverse)
 - Οι πολύ αντιστρόφου χρόνου (Very Inverse)
 - Οι εξαιρετικώς αντιστρόφου χρόνου (Extremely Inverse)

Στο Γράφημα 1 δίνονται οι γραφικές παραστάσεις του χρόνου λειτουργίας συναρτήσει του πολλαπλασίου του ρεύματος ρύθμισης για TDS=1.0 και για διάφορους τύπους ηλεκτρονόμων.



Γράφημα 1: Χαρακτηριστικές διάφορων τύπων ηλεκτρονόμων με TDS=1.0

Όταν τα προστατευόμενα κυκλώματα δε λειτουργούν ακτινικά (δηλαδή τροφοδοτούμενα από ένα σημείο), αλλά λειτουργούν κλειστά (δηλαδή τροφοδοτούμενα από περισσότερα του ενός σημεία), όπως ορισμένες φορές συμβαίνει ιδίως σε δίκτυα Διανομής όπου επιζητείται ένας μεγαλύτερος βαθμός συνέχειας της τροφοδοσίας, τότε χρησιμοποιούνται H/N υπερεντάσεως κατευθύνσεως. Γενικότερα, οι περιπτώσεις στις οποίες είναι απαραίτητη η χρήση αυτού του τύπου των ηλεκτρονόμων είναι [10]:

- Σε συστήματα με πολλές πηγές
- Σε βροχοειδή συστήματα και σε συστήματα με παράλληλες γραμμές
- Για ανίχνευση μη κανονικής κατεύθυνσης της ροής ενεργού και άεργου ισχύος (γεννήτριες)

Στη διπλωματική αυτή θα ασχοληθούμε με τη γενική περίπτωση των βροχοειδών δικτύων, οπότε οι ηλεκτρονόμοι που θα χρησιμοποιηθούν θα είναι H/N υπερεντάσεως με στοιχεία κατευθύνσεως.

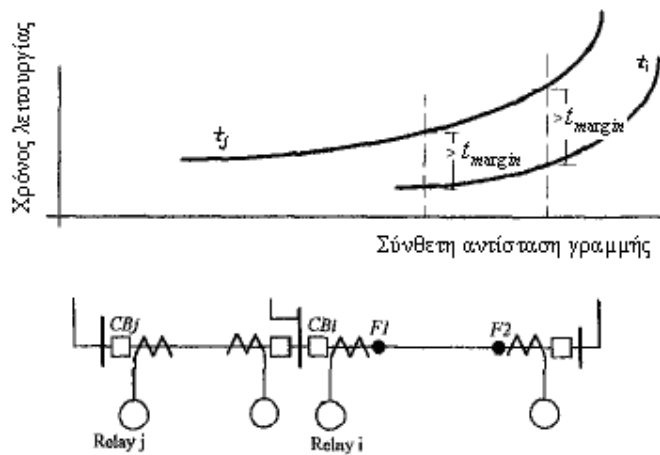
Τα στοιχεία κατεύθυνσης είναι πολύ σημαντικά για την επιλογικότητα, καθώς ελέγχουν τη λειτουργία των ηλεκτρονόμενων υπερέντασης. Αυτό που κάνουν είναι να ανιχνεύουν την κατεύθυνση προς την οποία έχει γίνει το σφάλμα, καταγράφοντας την κατεύθυνση του ρεύματος σφάλματος ή με άλλα λόγια τη διαφορά φάσης μεταξύ του ρεύματος και της τάσης. Οι ηλεκτρονόμοι υπερέντασης με στοιχεία κατεύθυνσης είναι πιο ευαίσθητοι από τους ηλεκτρονόμους υπερέντασης χωρίς στοιχεία κατεύθυνσης. Επιπλέον η συνεργασία μεταξύ των ηλεκτρονόμενων, η οποία θα εξηγηθεί στη συνέχεια, απλοποιείται, αφού το στοιχείο κατεύθυνσης περιορίζει τη λειτουργία του ηλεκτρονόμου μόνο σε σφάλματα προς τη μια κατεύθυνση.

2.2.2 Ζώνες προστασίας

Ζώνες προστασίας είναι τα τμήματα του συστήματος που καλύπτονται από μια ορισμένη προστασία [1]. Η μια ζώνη καλύπτει την άλλη, ώστε κανένα σημείο του δικτύου να μην μένει απροστάτευτο. Σε περίπτωση που για οποιονδήποτε λόγο η προστασία μιας ζώνης δε λειτουργήσει, ώστε να απομονωθεί το εντός αυτής σφάλμα, προβλέπεται από τη σχεδίαση (μελέτη) της προστασίας ότι θα λειτουργήσει η προστασία της προηγούμενης ζώνης. Η προστασία αυτή είναι η προστασία δευτέρης βαθμίδας ή δευτερεύουσα προστασία (back-up protection) σε αντίθεση με την κύρια προστασία της ζώνης που αποτελεί την προστασία πρώτης βαθμίδας ή πρωτεύουσα προστασία (primary protection). Στη συνέχεια, όποτε αναφέρουμε την έννοια συνεργασία ηλεκτρονόμενων θα εννοούμε τη λειτουργία του ηλεκτρονόμου δευτερεύουσας προστασίας σε σχέση με τη λειτουργία του ηλεκτρονόμου πρωτεύουσας προστασίας. Στο Σχήμα 1 για σφάλμα F1 ή F2 ο ηλεκτρονόμος i αποτελεί πρωτεύουσα προστασία, ενώ ο ηλεκτρονόμος j δευτερεύουσα προστασία. Στο διάγραμμα του σχήματος αυτού δίνεται ο χρόνος λειτουργίας κάθε ηλεκτρονόμου συναρτήσει της σύνθετης αντίστασης της γραμμής, η οποία είναι αντιστρόφως ανάλογη του ρεύματος σφάλματος.

Χρονικό περιθώριο πρωτεύουσας- δευτερεύουσας προστασίας

Πρόκειται για το χρονικό διάστημα που πρέπει να μεσολαβήσει μεταξύ της λειτουργίας του ηλεκτρονόμου που παρέχει δευτερεύουσα προστασία και αυτού που παρέχει πρωτεύουσα προστασία. Δηλαδή, αν κάποιος primary ηλεκτρονόμος έχει ρυθμιστεί να λειτουργήσει σε χρόνο t_1 , ο back-up ηλεκτρονόμος θα πρέπει να ρυθμιστεί να λειτουργήσει σε χρόνο $t_2 = t_1 + t_{margin}$. Το Σχήμα 1 δείχνει το χρονικό αυτό περιθώριο για δυο ηλεκτρονόμους i και j . Το χρονικό αυτό περιθώριο τίθεται για να αποφευχθεί η ταυτόχρονη λειτουργία των δυο ηλεκτρονόμενων, η οποία θα οδηγούσε σε αβεβαιότητα σχετικά με το σημείο στο οποίο έγινε το βραχυκύκλωμα και σε άσκοπη μη τροφοδότηση κάποιων φορτίων. Πρέπει όμως το χρονικό περιθώριο να είναι και αρκούντως μικρό, ώστε σε περίπτωση που αποτύχει η λειτουργία του primary ηλεκτρονόμου να λειτουργήσει έγκαιρα ο back-up ηλεκτρονόμος. Η εύρεση της κατάλληλης τιμής του χρονικού αυτού περιθωρίου είναι λοιπόν ένα ζήτημα.



Σχήμα 1 : Χρονικό περιθώριο t_{margin}

Οι παράγοντες από τους οποίους εξαρτάται το χρονικό περιθώριο είναι [16]:

- i. Ο χρόνος λειτουργίας του διακόπτη
Ο χρόνος που χρειάζεται ένας διακόπτης για να ενεργήσει σε περίπτωση που ανιχνευτεί ρεύμα σφάλματος εξαρτάται από το είδος του διακόπτη και την τιμή του ρεύματος σφάλματος. Οι κατασκευαστές παρέχουν συνήθως το χρόνο διακοπής σφάλματος για βαθμονομημένη χωρητικότητα διακοπής.
- ii. Το σφάλμα στη χρονομέτρηση του ηλεκτρονόμου
Όλοι οι ηλεκτρονόμοι παρουσιάζουν σφάλμα στη χρονομέτρησή τους συγκρινόμενοι με τα ιδανικά χαρακτηριστικά που ορίζονται από το IEC 60255.
- iii. Ο επιπλέον χρόνος λειτουργίας του ηλεκτρονόμου
Όταν ένας ηλεκτρονόμος απενεργοποιείται, η λειτουργία του μπορεί να συνεχίσει για λίγο ακόμα έως ότου σπαταληθεί η ενέργεια, η οποία μπορεί να είναι αποθηκευμένη στους πυκνωτές. Ο σχεδιασμός των ηλεκτρονόμων στοχεύει στην ελαχιστοποίηση και την απορρόφηση αυτής της ενέργειας, αλλά θα πρέπει να δίνεται και κάποιο περιθώριο. Ο επιπλέον χρόνος λειτουργίας του ηλεκτρονόμου ορίζεται ως η διαφορά μεταξύ του χρόνου λειτουργίας του ηλεκτρονόμου για ένα συγκεκριμένο ρεύμα σφάλματος και της μέγιστης διάρκειας που ο ηλεκτρονόμος δέχεται ρεύμα στην είσοδό του.
- iv. Τα σφάλματα των μετασχηματιστών έντασης
Οι μετασχηματιστές έντασης παρουσιάζουν σφάλματα λόγω του πολύ μεγάλου ρεύματος που απαιτείται για να μαγνητιστεί ο πυρήνας. Το αποτέλεσμα είναι ότι το ρεύμα στο δευτερεύον του μετασχηματιστή δεν είναι ένα αντίγραφο βαθμονομημένο με την ίδια κλίμακα. Συνεπώς, δημιουργούνται προβλήματα στη λειτουργία των ηλεκτρονόμων, όσον αφορά το χρόνο λειτουργίας.
- v. Τελικό χρονικό περιθώριο
Αφού έχουν ληφθεί υπόψη όλοι οι προηγούμενοι παράγοντες, ίσως χρειάζεται ένα επιπλέον χρονικό περιθώριο, έτσι ώστε ο ηλεκτρονόμος δευτερεύουσας προστασίας

να συνεργάζεται με τον ηλεκτρονόμο πρωτεύουσας προστασίας όπως ακριβώς απαιτείται.

Λαμβάνοντας υπόψη όλους αυτούς τους παράγοντες καθορίζεται το χρονικό περιθώριο t_{margin} . Κάποτε το σύνηθες χρονικό περιθώριο ήταν 0,5 sec, ενώ με τα σημερινά δεδομένα μια λογική τιμή του χρονικού περιθωρίου είναι 0,3 sec. Συνήθως χρησιμοποιείται σταθερό χρονικό περιθώριο για όλους τους ηλεκτρονόμους του υπό μελέτη συστήματος. Η τακτική αυτή είναι κατάλληλη στις περιπτώσεις που τα ρεύματα σφάλματος είναι μεγάλα, οπότε και οι χρόνοι λειτουργίας των ηλεκτρονόμων είναι μικροί. Για χαμηλότερα επίπεδα ρευμάτων σφάλματος, οπότε οι χρόνοι λειτουργίας είναι μεγαλύτεροι, το επιτρεπόμενο σφάλμα στο χρόνο λειτουργίας του ηλεκτρονόμου μπορεί να είναι μεγαλύτερο από το χρονικό περιθώριο, δημιουργώντας έτσι πρόβλημα στη συνεργασία των ηλεκτρονόμων. Σε αυτές τις περιπτώσεις απαιτείται επιλογή διαφορετικού t_{margin} για κάθε ζευγάρι ηλεκτρονόμων, ανάλογα με τη θέση στην οποία βρίσκονται. Στην παρούσα διπλωματική όμως δε θα μας απασχολήσει αυτή η περίπτωση και θα επιλεγεί ενιαίο t_{margin} για όλα τα ζεύγη ηλεκτρονόμων.

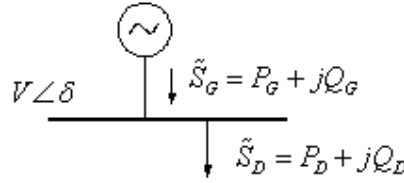
2.3 Ανάλυση ροών φορτίου

Το πρόβλημα των ροών φορτίου συνίσταται στο προσδιορισμό των μεταβλητών του συστήματος (ισχύων, ρευμάτων, τάσεων) σε μια δεδομένη μόνιμη κατάσταση λειτουργίας [2]. Η μόνιμη κατάσταση λειτουργίας αντιστοιχεί σε μια ορισμένη εικόνα φορτίων, μια αντίστοιχη εικόνα παραγόμενης ισχύος και αντίστοιχη εικόνα τάσεων και ροών στο δίκτυο. Κάθε άλλη εικόνα ροών, ή φορτίων, συνιστά μια άλλη κατάσταση λειτουργίας και περιγράφεται από άλλες τιμές μεταβλητών.

2.3.1 Μεταβλητές συστήματος

Θεωρώντας τη γενική περίπτωση ενός ζυγού του συστήματος με παραγωγή και φορτίο, διακρίνουμε τα ακόλουθα έξι μεγέθη (Σχήμα 2):

- P_G : παραγόμενη ενεργός ισχύς
- Q_G : παραγόμενη άεργος ισχύς
- P_D : ενεργός ισχύς φορτίου
- Q_D : άεργος ισχύς φορτίου
- V : μέτρο τάσης ζυγού
- δ : γωνία τάσης ζυγού



Σχήμα 2: Χαρακτηριστικά ηλεκτρικά μεγέθη ζυγού

Τα ηλεκτρικά αυτά μεγέθη αποτελούν τις μεταβλητές του προβλήματος και είναι 6 ανά ζυγό. Αντίστοιχα ορίζεται το ρεύμα ζυγού:

$$\tilde{J} = \frac{\tilde{S}^*}{\tilde{V}^*} = \frac{\tilde{S}_G^* - \tilde{S}_D^*}{\tilde{V}^*} = \frac{P_G - P_D - j(Q_G - Q_D)}{\tilde{V}^*} = \frac{P - jQ}{\tilde{V}^*}$$

Η εφαρμογή του νόμου ρευμάτων του Kirchhoff σε κάθε ζυγό ενός συστήματος N ζυγών, οδηγεί σε ένα σύστημα 2N εξισώσεων με 6N μεταβλητές η διαμόρφωση και η επίλυση του οποίου αποτελεί το στόχο της μελέτης ροών φορτίου.

2.3.2 Χαρακτηρισμός ζυγών

Για τις ανάγκες της μελέτης ροών φορτίου ορίζονται οι ακόλουθοι τρεις χαρακτηριστικοί τύποι ζυγών:

- Ζυγοί φορτίου ή PQ, στους οποίους είναι γνωστές οι ισχείς φορτίου και άγνωστη η τάση κατά μέτρο και γωνία.
- Ζυγοί παραγωγής ή PV, όπου είναι γνωστά η παραγόμενη ενεργός ισχύς και το μέτρο της τάσης και προσδιορίζονται η άεργος ισχύς και η γωνία της τάσης.
- Ζυγός ταλάντωσης ή αναφοράς ή slack, ο οποίος είναι ζυγός παραγωγής, χρησιμοποιούμενος για να αντισταθμίσει τις απώλειες του δικτύου, που δεν είναι γνωστές εξ' αρχής, και γενικά για άμεση αντιμετώπιση των μεταβολών ισχύος. Στο ζυγό αυτό θεωρείται σταθερή τάση κατά μέτρο και γωνία και προσδιορίζονται οι ισχείς παραγωγής.

2.3.3 Εξισώσεις ροών φορτίου

Η μαθηματική ανάλυση καταλήγει στη διαμόρφωση ενός συστήματος 2N μη γραμμικών εξισώσεων, όπου N ο αριθμός των ζυγών του δικτύου. Οι εγκάρσιες αγωγιμότητες προς γη των γραμμών μεταφοράς συγκεντρώνονται στους ζυγούς, οπότε για έναν οποιονδήποτε ζυγό i του συστήματος θα είναι:

$$Y_{i0} = \sum_{j=1}^N Y_{ij0}$$

Η εξίσωση των ρευμάτων στο ζυγό γράφεται:

$$\tilde{J}_i = \tilde{V}_i Y_{i0} + \sum_{j \neq i}^N (\tilde{V}_i - \tilde{V}_j) Y_{ij} = \left(Y_{i0} + \sum_{j \neq i}^N Y_{ij} \right) \tilde{V}_i - Y_{i1} \tilde{V}_1 - Y_{i2} \tilde{V}_2 - \dots - Y_{iN} \tilde{V}_N$$

Όπου $Y_{i1}, Y_{i2}, \dots, Y_{iN}$ οι αγωγιμότητες σειράς των γραμμών που συνδέονται στο ζυγό i και $V_1, V_2, \dots, V_N$ οι τάσεις προς γη των ζυγών του δικτύου.

Εισάγοντας τις ακόλουθες νέες παραμέτρους,

$$y_{ii} = Y_{i0} + \sum_{j \neq i}^N Y_{ij}$$

$$y_{ij} = -Y_{ij} = -Y_{ji}$$

η εξίσωση των ρευμάτων του ζυγού γράφεται: $\tilde{J}_i = \sum_{j=i}^N y_{ij} \tilde{V}_j$. Έχουμε συνεπώς για τους N ζυγούς του συστήματος:

$$\begin{bmatrix} \tilde{J}_1 \\ \tilde{J}_2 \\ \dots \\ \tilde{J}_N \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1N} \\ y_{21} & y_{22} & \dots & y_{2N} \\ \dots & \dots & \dots & \dots \\ y_{N1} & \dots & \dots & y_{NN} \end{bmatrix} \begin{bmatrix} \tilde{V}_1 \\ \tilde{V}_2 \\ \dots \\ \tilde{V}_N \end{bmatrix}$$

και συμβολικά $\begin{bmatrix} \tilde{J}_z \end{bmatrix} = \begin{bmatrix} Y_z \end{bmatrix} \begin{bmatrix} \tilde{V}_z \end{bmatrix}$,

όπου $\begin{bmatrix} Y_z \end{bmatrix}$ η μήτρα αγωγιμοτήτων ζυγών του συστήματος.

Δεδομένου ότι για έναν ζυγό ισχύει $\tilde{J}_i = \frac{P_i - jQ_i}{\tilde{V}_i^*}$, η τυπική μιγαδική εξίσωση ροών φορτίου γράφεται:

$$P_i - jQ_i = \tilde{V}_i^* \sum_{j=i}^N y_{ij} \tilde{V}_j \quad (2.1)$$

Χωρίζοντας πραγματικά και φανταστικά μέρη, προκύπτουν από την (2.1) δυο εξισώσεις για κάθε ζυγό και συνεπώς για το σύστημα N ζυγών έχουμε $2N$ εξισώσεις ροών φορτίου της μορφής:

$$P_i = |V_i| \cdot \sum_{j=i}^N |V_j| \left[g_{ji} \cos(\delta_j - \delta_i) - b_{ji} \sin(\delta_j - \delta_i) \right]$$

$$-Q_i = |V_i| \cdot \sum_{j=i}^N |V_j| \left[g_{ji} \sin(\delta_j - \delta_i) + b_{ji} \cos(\delta_j - \delta_i) \right]$$

Οι εξισώσεις αυτές είναι αλγεβρικές, εφόσον αναφέρονται στη μόνιμη κατάσταση λειτουργίας και μη γραμμικές.

- Η ανάλυση ροών φορτίου είναι λοιπόν η διαδικασία κατά την οποία, κάνοντας χρήση των παραπάνω εξισώσεων φορτίου, υπολογίζουμε την τάση κάθε ζυγού του συστήματος κατά την κανονική λειτουργία. Χρησιμοποιώντας αυτές τις τάσεις θα υπολογίσουμε το ρεύμα που ρέει σε κάθε γραμμή του συστήματος, που είναι και η μεταβλητή που μας ενδιαφέρει για τη ρύθμιση των ηλεκτρονόμων.

2.4 Ανάλυση βραχυκυκλωμάτων

2.4.1 Βραχυκυκλώματα

Με τον όρο συμμετρικά βραχυκυκλώματα χαρακτηρίζονται οι ανώμαλες μεταβατικές καταστάσεις, στις οποίες περιέρχεται το σύστημα όταν λόγω βλάβης σε κάποιο σημείο έρθουν σε επαφή μεταξύ τους και οι τρεις φάσεις [3]. Πλην των συμμετρικών βραχυκυκλωμάτων υπάρχουν και τα ασύμμετρα βραχυκυκλώματα, όπως είναι τα διφασικά και τα μονοφασικά βραχυκυκλώματα. Τα τελευταία είναι και τα συνηθέστερα. Τα περισσότερα βραχυκυκλώματα, ή σφάλματα, γίνονται στα εκτεθειμένα μέρη του δικτύου, όπως στις εναέριες γραμμές, στους ακροδέκτες του εξοπλισμού κλπ, είναι δε και τα πιο ανώδυνα. Αντίθετα, τα σφάλματα στο εσωτερικό μιας συσκευής, π.χ. ενός μετασχηματιστή, ή ενός καλωδίου είναι μεν σπανιότερα αλλά και καταστρεπτικότερα.

Τα βραχυκυκλώματα, σαν μεταβατικά φαινόμενα, είναι ηλεκτρικής φύσεως και δυναμικής μορφής. Εν σχέσει με τις υπερτάσεις, οι οποίες είναι ηλεκτρικά επίσης μεταβατικά φαινόμενα αλλά διηλεκτρικής μορφής, τα βραχυκυκλώματα είναι βραδύτερα στην εξέλιξή τους διότι καθορίζονται βασικά από τη μαγνητική αντίδραση και αλληλεπίδραση των τυλιγμάτων των γεννητριών.

Τα βραχυκυκλώματα συνοδεύονται από στιγμιαίο μηδενισμό ή βύθιση της τάσεως σε όλο το δίκτυο, σε διαφορετικό βέβαια βαθμό για κάθε ζυγό. Με την ξαφνική μείωση αυτή της τάσεως μειώνεται απότομα η εξερχόμενη ισχύς των γεννητριών ενώ προς στιγμήν η μηχανική ισχύς των στροβίλων μένει σταθερή. Αυτή η κατάσταση δημιουργεί ένα προσωρινό περίσσειμα κινητήριας ισχύος που δημιουργεί ένα ζεύγος επιταχύνσεως με πιθανές δυσμενείς συνέπειες για την ευστάθεια λειτουργίας του συστήματος.

Εκτός από τις επιπτώσεις αυτές στη λειτουργία του συστήματος τα βραχυκυκλώματα ως υπερεντάσεις μπορούν να φθάσουν σε τιμές καταστρεπτικές για τον εξοπλισμό, κυρίως τις γεννήτριες και τους μετασχηματιστές. Οι υπερεντάσεις αυτές προκαλούν μηχανικές και θερμικές καταπονήσεις στον εξοπλισμό του συστήματος, πολλές φορές με καταστρεπτικά αποτελέσματα, αν δεν εξουδετερωθούν έγκαιρα. Το καθήκον της εξουδετέρωσης των ρευμάτων βραχυκύκλωσης έχουν οι διακόπτες ισχύος και το σύστημα προστασίας του δικτύου. Ο υπολογισμός των ρευμάτων βραχυκύκλωσης, που αποτελεί το κύριο έργο των μελετών των βραχυκυκλωμάτων, ενδιαφέρει πρώτιστα τα ανωτέρω μέσα διακοπής και προστασίας. Η ικανότητα διακοπής των διακοπών βασίζεται στη δυσμενέστερη περίπτωση βραχυκυκλώματος, ενώ το σύστημα προστασίας βασίζει τη λειτουργία του στο μέγεθος και τη φορά κατευθύνσεως των ρευμάτων βραχυκύκλωσης.

- Η ανάλυση βραχυκυκλωμάτων είναι η διαδικασία εύρεσης του ρεύματος σφάλματος για σφάλμα σε κάποιο σημείο του δικτύου, καθώς και των ρευμάτων που ρέουν σε κάθε κλάδο του συστήματος μετά την εκδήλωση του σφάλματος και πριν την εκκαθάρισή του από τα μέσα προστασίας. Για τη ρύθμιση των ηλεκτρονόμων υπερέντασης ενδιαφερόμαστε για τα σοβαρότερα σφάλματα, γι' αυτό κατά την ανάλυση βραχυκυκλωμάτων θεωρούνται

συμμετρικά (τριφασικά) βραχυκυκλώματα. Σε κάθε επανάληψη της ανάλυσης θεωρείται σφάλμα σε ένα ζυγό του συστήματος και υπολογίζονται οι συνεισφορές από τις διάφορες γραμμές του δικτύου.

2.4.2 Ασύμμετρα τριφασικά συστήματα

Τα τριφασικά συστήματα μπορούν να αναλυθούν και να μελετηθούν ευκολότερα με τη βοήθεια ισοδύναμων μονοφασικών κυκλωμάτων. Ένα τριφασικό σύστημα μπορεί να παρασταθεί από ισοδύναμο μονοφασικό κύκλωμα όταν είναι συμμετρικό, δηλαδή όλες οι φάσεις είναι ίδιες, δεν υπάρχουν ζεύξεις μεταξύ των φάσεων ή όταν υπάρχουν ζεύξεις μεταξύ φάσεων αυτές είναι όλες ίσες μεταξύ τους, και επιπλέον το κύκλωμα τροφοδοτείται από συμμετρικό τριφασικό σύστημα τάσεων. Η συμμετρία του συστήματος στην περίπτωση αυτή είναι διαγώνια και κυκλική. Τα τριφασικά συστήματα τα οποία συναντώνται στην πράξη δεν είναι απολύτως συμμετρικά στις περισσότερες περιπτώσεις. Σε πολλές περιπτώσεις όμως οι ασυμμετρίες, οι οποίες οφείλονται σε ζεύξεις μεταξύ φάσεων, δηλαδή αμοιβαίες επαγωγικές και χωρητικές αντιδράσεις, είναι σχετικά πολύ μικρές και μπορούν να αγνοηθούν ή να θεωρηθούν ίσες μεταξύ τους, οπότε το σύστημα αποκτά τη δυνατότητα μονοφασικής ισοδυναμίας.

Σε άλλες περιπτώσεις οι ασυμμετρίες μπορούν να εξουδετερωθούν, όπως γίνεται με την αντιμετάθεση των αγωγών τριφασικών γραμμών. Υπενθυμίζεται ότι η γενική ασύμμετρη τριφασική γραμμή χωρίς αντιμετάθεση αγωγών παρουσιάζει επαγωγικές και χωρητικές ζεύξεις μεταξύ των φάσεων, ώστε οι τρεις φάσεις δεν μπορούν να ανεξαρτητοποιηθούν και να επιτευχθεί ισοδύναμο μονοφασικό κύκλωμα.

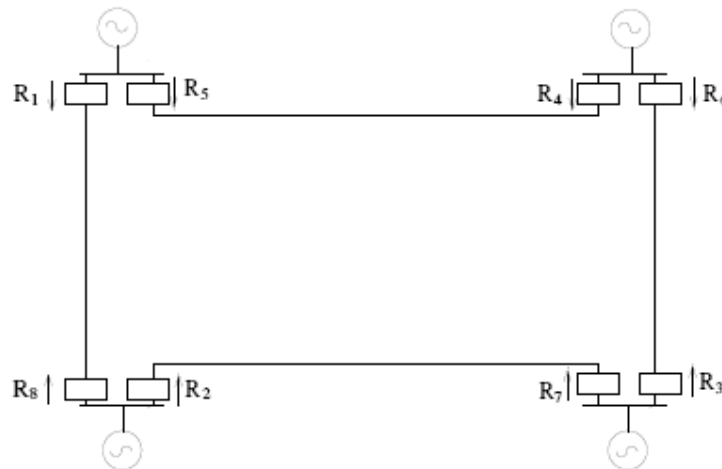
Σε άλλες περιπτώσεις ασυμμετρίας, το ασύμμετρο τριφασικό σύστημα είναι δυνατόν να αναλυθεί σε τρία ανεξάρτητα μονοφασικά κυκλώματα διασυνδεδεμένα μεταξύ τους στα σημεία της ασυμμετρίας. Αυτό μπορεί να επιτευχθεί με μια αλλαγή μεταβλητών, δηλαδή ένα μετασχηματισμό συνιστωσών. Τέτοιοι είναι ο γνωστός μετασχηματισμός συμμετρικών συνιστωσών και ο μετασχηματισμός Clarke. Ο μετασχηματισμός συμμετρικών συνιστωσών μετασχηματίζει ένα τριφασικό σύστημα σε δυο συμμετρικά τριφασικά συστήματα, ευθύ (θετικής ακολουθίας) και αντίστροφο (αρνητικής ακολουθίας), και σε ένα ομοπολικό (μηδενικής ακολουθίας). Κυκλώματα ακολουθίας είναι τα κυκλώματα στα οποία ρέουν τα αντίστοιχα ρεύματα ακολουθίας. Το κύκλωμα θετικής ακολουθίας περιλαμβάνει αντιστάσεις θετικής ακολουθίας και εμφανίζει τάσεις θετικής ακολουθίας. Τα κυκλώματα αρνητικής και μηδενικής ακολουθίας δεν περιλαμβάνουν ΗΕΔ, παρά μόνο τις αντίστοιχες αντιστάσεις ακολουθίας.

2.5 Τοπολογική ανάλυση συστήματος

Η συνεργασία ηλεκτρονόμων υπερέντασης με στοιχεία κατευθύνσεως απαιτεί σύγκριση της λειτουργίας του ηλεκτρονόμου πρωτεύουσας προστασίας για κάποιο σφάλμα με τη λειτουργία όλων των ηλεκτρονόμων δευτερεύουσας προστασίας. Έτσι, πριν την έναρξη της διαδικασίας ρύθμισης ώστε να επιτυγχάνεται η επιθυμητή συνεργασία θα πρέπει να αναγνωριστούν όλα τα ζευγάρια ηλεκτρονόμων πρωτεύουσας - δευτερεύουσας προστασίας (primary - backup, P-B). Για τη ρύθμιση

κάθε ηλεκτρονόμου x θα πρέπει να έχει προηγηθεί και να λαμβάνεται υπόψη η ρύθμιση των ηλεκτρονόμων για τους οποίους ο x λειτουργεί ως backup. Το πρόβλημα που προκύπτει όμως ακολουθώντας αυτή την τακτική εξηγείται με το επόμενο παράδειγμα.

Ας θεωρήσουμε το βρόχο του Σχήματος 3. Έστω ότι η ηλεκτρονόμος R_1 ρυθμίζεται πρώτος. Στη συνέχεια θα ρυθμιστεί ο R_4 έτσι ώστε να συνεργάζεται με τον R_1 . Έπειτα ρυθμίζουμε τον R_3 ώστε να συνεργάζεται με τον R_4 , τον R_2 ώστε να συνεργάζεται με τον R_3 και τον R_5 ώστε να συνεργάζεται με τον R_1 . Καθώς οι ρυθμίσεις του ηλεκτρονόμου R_1 άλλαξαν, πρέπει να ξαναεπισκεφτούμε τον ηλεκτρονόμο R_4 και να τον ρυθμίσουμε εκ νέου ώστε να συνεργάζεται σωστά με τον ηλεκτρονόμο R_1 . Έτσι, διασχίζουμε ξανά τον βρόχο. Συνεπώς, έχουμε μια επαναλαμβανόμενη διαδικασία ρύθμισης των ηλεκτρονόμων σε κάθε βρόχο του συστήματος. Σε ένα σύστημα μεταφοράς έχουμε πολλούς τέτοιους βρόχους, παρακείμενο τον έναν στον άλλο, και συνεπώς πρέπει να γίνει ένας μεγάλος αριθμός επαναλαμβανόμενων υπολογισμών ώστε να επιτύχουμε συνεργασία όλων των ηλεκτρονόμων του συστήματος.



Σχήμα 3

Συμπεραίνουμε λοιπόν ότι είναι σημαντικό να βρεθεί ο ελάχιστος αριθμός των αρχικών ηλεκτρονόμων έτσι ώστε να ελαχιστοποιηθεί ο αριθμός των επαναλήψεων. Αυτό το ελάχιστο σύνολο ηλεκτρονόμων αναφέρεται ως Break Point (BP) ηλεκτρονόμοι και αποτελείται από τους ηλεκτρονόμους των οποίων το άνοιγμα οδηγεί στο άνοιγμα όλων των βρόχων του συστήματος. Το άνοιγμα των ηλεκτρονόμων αυτών δηλαδή μετατρέπει το βροχοειδές σύστημα σε ακτινικό. Από τη λογική υπολογισμού των Break Points συμπεραίνουμε ότι θα πρέπει να έχει προηγηθεί η εύρεση όλων των βρόχων (loops) του συστήματος. Η εύρεση του ελάχιστου συνόλου των BP ηλεκτρονόμων είναι ένα κύριο αποτέλεσμα αυτής της τοπολογικής ανάλυσης.

Από τη στιγμή που βρεθεί αυτό το σύνολο, το επόμενο βήμα είναι να τοποθετηθούν όλοι οι ηλεκτρονόμοι σε συγκεκριμένη αλληλουχία, η οποία υποδεικνύει τη σειρά

ρύθμισής τους. Η δεδομένη αυτή διάταξη των ηλεκτρονόμων ονομάζεται Διάνυσμα Σχετικής Ακολουθίας (Relative Sequence Vector- RSV).

Αφού η συνεργασία των ηλεκτρονόμων βασίζεται στην έννοια των ζευγαριών P/B, το επόμενο βήμα είναι να καθορίσουμε το Σύνολο των Διαδοχικών Ζευγών (Set of Sequential Pairs- SSP), που είναι η διατεταγμένη ακολουθία των ζευγαριών P/B του συστήματος.

Κατά την τοπολογική ανάλυση του συστήματος λοιπόν θα πρέπει να υπολογιστούν τα ακόλουθα στοιχεία [18]:

- Οι βρόχοι του συστήματος (loops)
- Το σύνολο των Break Points
- Το Διάνυσμα Σχετικής Ακολουθίας (RSV)
- Το Σύνολο των Διαδοχικών Ζευγών (SSP)

2.6 Επιλογή - ρύθμιση εξοπλισμού προστασίας

Για την προστασία υπερεντάσεως, η επιλογή του εξοπλισμού προστασίας περιλαμβάνει σύμφωνα με το [4] τα ακόλουθα:

1. Επιλογή των μετασχηματιστών εντάσεως (Current Transformers- CT)

Για την επιλογή των Μ/Σ έντασης πρέπει να ληφθούν υπόψη οι ακόλουθες παράμετροι:

- Το ρεύμα στο πρωτεύον του μετασχηματιστή έντασης πρέπει να είναι μεγαλύτερο από το ρεύμα της γραμμής σε καταστάσεις ομαλής φόρτισης ή αποδεκτών υπερτάσεων.
- Το ρεύμα στο δευτερεύον του μετασχηματιστή έντασης πρέπει να ταιριάζει με το αναγραφόμενο ρεύμα λειτουργίας του χρησιμοποιούμενου ρεύμα υπερέντασης (τυπικά 5A ή 1A).
- Πρέπει να αποφεύγονται φαινόμενα κορεσμού στους μετασχηματιστές έντασης όταν αυτοί διαρρέονται από τα μέγιστα ρεύματα σφάλματος.

2. Επιλογή των διακοπών ισχύος (CB)

Για την επιλογή των διακοπών ισχύος πρέπει να ληφθούν υπόψη τα ακόλουθα:

- Το μέγιστο ρεύμα που μπορούν να αποκόψουν οι διακόπτες ισχύος πρέπει να είναι μεγαλύτερο από το ρεύμα σφάλματος που μπορεί να προκύψει από το δίκτυο.
- Ο μέγιστος χρόνος λειτουργίας ενός διακόπτη ισχύος πρέπει να είναι γνωστός εκ των προτέρων για το σωστό καθορισμό της απόκρισης των συνδεδεμένων Η/Ν υπερέντασης.

Σε αυτή τη διπλωματική δε θα μας απασχολήσει η επιλογή των διακοπών, αλλά θα θεωρήσουμε ότι αυτοί επιλέγονται σωστά και συμφωνούν με τα παραπάνω κριτήρια.

3. Επιλογή των Η/Ν υπερέντασης

Οι ακόλουθες παράμετροι πρέπει να ληφθούν υπόψη για την επιλογή των ηλεκτρονόμων υπερέντασης:

- Το αναγραφόμενο ρεύμα λειτουργίας του ηλεκτρονόμου πρέπει να ταιριάζει με το ρεύμα στο δευτερεύον του μετασχηματιστή έντασης ή να είναι της ίδιας τάξης μεγέθους.
- Οι επιλογή των παραμέτρων του ηλεκτρονόμου (ρεύμα ρύθμισης και χρονική σταθερά) πρέπει να ταιριάζει με τις εκάστοτε ανάγκες προστασίας που καθορίζονται από τη θέση του ηλεκτρονόμου.

Με την έννοια «παράμετροι ηλεκτρονόμου υπερέντασης» εννοούμε λοιπόν τα εξής χαρακτηριστικά τους:

- Ρεύμα ρύθμισης (Pick up Current)
- Χρονική σταθερά (Time Dial Setting- TDS)

Ο καθορισμός των παραμέτρων αυτών αποτελεί και το σκοπό της διπλωματικής αυτής. Ο ρόλος όλων των άλλων υπολογισμών και διαδικασιών που θα προηγηθούν είναι να παρέχουν τα δεδομένα που είναι απαραίτητα για τη διαδικασία ρύθμισης των ηλεκτρονόμων υπερέντασης. Ο τρόπος ρύθμισης πρόκειται να καθοριστεί σε επόμενο κεφάλαιο.

3

Άλλες τεχνικές

Το κεφάλαιο αυτό αποτελείται από τέσσερις υποενότητες. Σε κάθε μια από αυτές παρουσιάζονται τεχνικές για την ολοκλήρωση καθενός από τα τέσσερα βασικά βήματα που απαιτούνται για τη ρύθμιση των ηλεκτρονόμων υπερέντασης και τα οποία αναφέρθηκαν στο εισαγωγικό κεφάλαιο. Δεν πρόκειται για τις τεχνικές οι οποίες θα χρησιμοποιηθούν στην παρούσα διπλωματική εργασία, αλλά για εναλλακτικές τεχνικές. Η παρουσίασή τους γίνεται για λόγους πληρότητας και σύγκρισης με τις μεθόδους που θα χρησιμοποιηθούν στη συνέχεια.

3.1 Ανάλυση ροών φορτίου

Οι μέθοδοι που χρησιμοποιούνται για τους υπολογισμούς των ροών φορτίου μπορούν αν λύσουν τις μη γραμμικές εξισώσεις των ροών φορτίου ακόμα και για μεγάλα συστήματα με ικανοποιητική ακρίβεια. Οι δυο βασικές μέθοδοι που εφαρμόζονται στις μελέτες ροών φορτίου είναι:

- a) Η μέθοδος Gauss - Siedel
- b) Η μέθοδος Newton – Raphson

Και οι δυο αυτές μέθοδοι ξεκινούν από μια αρχική λύση και με διαδοχικές ανακυκλώσεις προσεγγίζουν την τελική ως εξής: Αρχική λύση χρησιμοποιείται στις εξισώσεις του προβλήματος για να βρεθεί μια νέα καλύτερη λύση. Η δεύτερη λύση χρησιμοποιείται στη συνέχεια για να βρεθεί μια τρίτη καλύτερη λύση κ.ο.κ. Μια τέτοια συγκλίνουσα διαδικασία λέγεται μέθοδος ή διαδικασία ανακύκλωσης και η ποιότητά της κρίνεται από την ταχύτητα συγκλίσεώς της.

Στη διπλωματική αυτή θα χρησιμοποιηθεί η μέθοδος Newton – Raphson. Συνεπώς στο κεφάλαιο αυτό, όπου παρουσιάζονται τεχνικές διαφορετικές από αυτές που χρησιμοποιήθηκαν, θα εξηγηθεί η μέθοδος Gauss – Siedel, όπως αυτή παρουσιάζεται στο [2].

3.1.1 Μέθοδος Gauss – Siedel

Υποθέτουμε γνωστή μια αρχική σειρά τιμών όλων των τάσεων των ζυγών (μόνο του ζυγού ταλάντωσης η τάση είναι επιβεβλημένη) και έστω $\tilde{V}_k$ η αρχική αυτή τιμή για το ζυγό k. Θεωρώντας τη γενική μορφή των εξισώσεων ροών φορτίου (2.1) για το ζυγό k και λύνοντας ως προς $\tilde{V}_k$ έχουμε:

$$\tilde{V}_k = \frac{1}{y_{kk}} \left\{ \frac{P_k - jQ_k}{\tilde{V}_k^*} - \sum_{\substack{j=1 \\ j \neq k}}^N y_{kj} \tilde{V}_j \right\}, \quad j=1,2,3,\dots,N \quad (3.1)$$

Η σχέση αυτή χρησιμοποιείται επαναληπτικά σύμφωνα με τη διαδικασία Gauss – Siedel. Για τον προσδιορισμό της $\tilde{V}_k$ στο βήμα $(n+1)$ χρησιμοποιούνται οι προσεγγίσεις του βήματος $(n+1)$ των τάσεων $\tilde{V}_1 - \tilde{V}_{k-1}$. Με τον τρόπο αυτό η (3.1) κατά τη $(n+1)$ επανάληψη θα είναι:

$$\tilde{V}_k^{(n+1)} = \frac{1}{y_{kk}} \left\{ \frac{P_k^{(n)} - jQ_k^{(n)}}{\tilde{V}_k^{(n)*}} - \sum_{j=1}^{k-1} y_{kj} \tilde{V}_j^{(n+1)} - \sum_{j=k+1}^N y_{kj} \tilde{V}_j^{(n)} \right\} \quad (3.2)$$

Στους ζυγούς φορτίου ο τύπος (3.2) εφαρμόζεται κατευθείαν δεδομένου ότι τα P_k και Q_k είναι γνωστά. Στους ζυγούς παραγωγής όμως δεν είναι γνωστό το Q_k και χρειάζεται να υπολογιστεί από τον τύπο:

$$Q_k^{(n)} = -\text{Im} \left\{ \tilde{V}_k^{(n)*} \left[\sum_{j=1}^{k-1} y_{kj} \tilde{V}_j^{(n+1)} + \sum_{j=k}^N y_{kj} \tilde{V}_j^{(n)} \right] \right\} \quad (3.3)$$

Η μέθοδος συγκλίνει γρηγορότερα αν χρησιμοποιηθεί συντελεστής επιτάχυνσης. Η διαφορά δηλαδή μεταξύ δυο βημάτων της μεθόδου πολλαπλασιάζεται με έναν αριθμό α (γύρω στο 1,6 συνήθως).

$$\tilde{V}_k^{(n+1)} = \tilde{V}_k^{(n)} + \alpha \left(\tilde{V}_k^{(n+1)'} - \tilde{V}_k^{(n)} \right) \quad (3.4)$$

Στους ζυγούς παραγωγής ο συντελεστής επιτάχυνσης εφαρμόζεται στη γωνία δ .

$$\delta_k^{(n+1)} = \delta_k^{(n)} + \alpha \left(\delta_k^{(n+1)'} - \delta_k^{(n)} \right) \quad (3.5)$$

Όπου οι τονούμενες τιμές είναι αυτές που προκύπτουν από τον τύπο (3.2).

Ένας συνοπτικός αλγόριθμος της μεθόδου Gauss – Siedel είναι:

- 1) Δίνονται αρχικές τιμές στις τάσεις V των ζυγών φορτίου και στις γωνίες δ των ζυγών παραγωγής.
- 2) Για κάθε ζυγό του συστήματος εκτός του ζυγού ταλάντωσης:
 - a. Αν είναι ζυγός παραγωγής υπολογίζουμε το Q_k από τον τύπο (3.3).
 - b. Υπολογίζεται η νέα τιμή της τάσης $\tilde{V}_k^{(n+1)'}$ από τον τύπο (3.2).
 - c. Αν είναι ζυγός φορτίου διορθώνουμε το $\tilde{V}_k$ σύμφωνα με τον τύπο (3.4) και έχουμε τη νέα προσέγγιση $\tilde{V}_k^{(n+1)}$.
 - d. Αν είναι ζυγός παραγωγής διορθώνουμε τη γωνία $\delta_k^{(n+1)'}$ του 2b, σύμφωνα με τον τύπο (3.5). Η νέα τάση είναι $\tilde{V}_k^{(n+1)} = |\tilde{V}_k| \angle \delta_k^{(n+1)}$.
- 3) Αν οι νέες τιμές των τάσεων διαφέρουν από τις προηγούμενες περισσότερο από ένα όριο ανοχής ξαναγυρίζουμε στο βήμα 2.
- 4) Εφόσον η μέθοδος έχει συγκλίνει υπολογίζουμε τις ροές ισχύων και τις απώλειες.

3.2 Ανάλυση βραχυκυκλωμάτων

3.2.1 Μέθοδος στο πεδίο των ακολουθιακών κυκλωμάτων

Η πρώτη και πιο κοινή μέθοδος για την ανάλυση βραχυκυκλωμάτων σε συστήματα μεταφοράς είναι η ακολουθιακή μέθοδος. Σε αυτή τη μέθοδο το ασύμμετρο τριφασικό σύστημα αναλύεται σε τρία συμμετρικά συστήματα, θετικής, αρνητικής και μηδενικής ακολουθίας, όπως εξηγήθηκε στην υποενότητα 2.4.2, και δημιουργούνται οι αντίστοιχοι πίνακες ακολουθιακών σύνθετων αντιστάσεων ή αγωγιμοτήτων. Στην περίπτωση των τριφασικών βραχυκυκλωμάτων που μας ενδιαφέρει, χρειάζεται μόνο ο πίνακας σύνθετων αντιστάσεων θετικής ακολουθίας, ο οποίος δηλώνεται με το δείκτη 1 στις εξισώσεις που ακολουθούν και προκύπτει μετά από αντιστροφή του πίνακα αγωγιμοτήτων θετικής ακολουθίας. Υπολογίζουμε το ρεύμα σφάλματος και στη συνέχεια τις τάσεις των κόμβων μετά το σφάλμα. Οι σχέσεις που χρησιμοποιούνται [15] έχουν ως εξής:

$$\begin{aligned} Z_1 &= Y_1^{-1} \\ I_i^f &= \frac{-V_i^1}{Z_{ii}^1} \\ \Delta V_j^1 &= Z_{ij}^1 I_i^f \quad \Rightarrow \quad V_j^{final} = V_j^1 + \Delta V_j^1 \\ \Delta I_{jk}^1 &= \frac{\Delta V_j^1 - \Delta V_k^1}{Z_{jk}^1} \quad \Rightarrow \quad I_{jk}^{final} = I_{jk}^1 + \Delta I_{jk}^1 \end{aligned}$$

Όπου i ο κόμβος στον οποίο γίνεται το σφάλμα και j οποιοσδήποτε άλλος κόμβος του συστήματος.

Τα βασικά βήματα της μεθόδου αυτής είναι λοιπόν:

- Δημιουργία των πινάκων σύνθετων αντιστάσεων
- Υπολογισμός του ρεύματος σφάλματος
- Υπολογισμός των μετά-σφάλματος τάσεων
- Υπολογισμός των μετά-σφάλματος ρευμάτων γραμμής

Μειονεκτήματα

Παρ' όλο που η ακολουθιακή μέθοδος συνεισφέρει ακόμα αρκετά στην ανάλυση των βραχυκυκλωμάτων ενός συστήματος, έχει κάποια κενά που περιορίζουν την εφαρμογή της. Η παραδοσιακή αυτή μέθοδος δεν μπορεί να χειριστεί σύνθετα σφάλματα, όπου υπάρχει αμοιβαία σύζευξη μεταξύ των ακολουθιακών κυκλωμάτων.

3.2.2 Μέθοδοι στο πεδίο των φάσεων

1) Για να λυθεί αυτό το πρόβλημα της ακολουθιακής μεθόδου, ο Alvarado παρουσίασε το 1985 ένα σημαντικό αλγόριθμο που εμπεριέχει το πεδίο των φάσεων. Το ίδιο έτος ο Tinney παρουσίασε επίσης μια μικτή μέθοδο για ανάλυση βραχυκυκλωμάτων, η οποία είναι σχεδόν ίδια με τη μέθοδο του Alvarado. Υπάρχει

και μια άλλη παρόμοια μέθοδος που βασίζεται στις μεθόδους αυτών των δυο. Στις μεθόδους τους η μήτρα ακολουθιακών αγωγιμοτήτων μετατρέπεται πρώτα στο πεδίο των φάσεων και έπειτα αλληλοκαλύπτεται με τον πίνακα σφάλματος. Ένας πίνακας 3×3 προκύπτει συμπτίσσοντας τον φασικό πίνακα, όπως υποδεικνύεται από το σφάλμα. Έστερα, οι φασικές τάσεις στο σημείο σφάλματος υπολογίζονται από αυτόν τον καινούριο πίνακα. Μετά από αυτό, οι φασικές τάσεις μετατρέπονται και πάλι στο ακολουθιακό πεδίο και χρησιμοποιούνται σε αυτή τη μορφή για να υπολογιστούν το ρεύμα σφάλματος και οι τάσεις μετά το σφάλμα στο πεδίο των ακολουθιακών κυκλωμάτων. Είναι σαφές ότι μόνο ένα μέρος των υπολογισμών πραγματοποιείται στο πεδίο φάσης.

2) Από την άλλη, μια εναλλακτική μέθοδος στο φασικό πεδίο αναπτύχθηκε το 1968. Η μέθοδος αυτή περιλαμβάνει δυο μεθόδους επίλυσης για τον υπολογισμό των βραχυκυκλωμάτων, τη μέθοδο μετασχηματισμού των πηγών και τη μέθοδο των κατανεμημένων πηγών. Η μέθοδος αυτή είναι η πρώτη μέθοδος στο πεδίο φάσεων για την ανάλυση βραχυκυκλωμάτων.

3) Αργότερα, παρουσιάστηκε μια γενικευμένη μέθοδος στο πεδίο φάσεων [17], η οποία βασίζεται στη μέθοδο των κατανεμημένων πηγών. Στη μέθοδο αυτή όλα τα στοιχεία του ηλεκτρικού συστήματος μοντελοποιούνται στο πεδίο των φάσεων και έπειτα υπολογίζεται ο τριφασικός πίνακας αγωγιμοτήτων. Μετά το σφάλμα, ο πίνακας αγωγιμοτήτων Y μετατρέπεται σε Y' για να συμπεριλάβει τις αλλαγές στη συνδεσμολογία του δικτύου σύμφωνα με το σφάλμα.

Στη συνέχεια μπορούν να υπολογιστούν οι πτώσεις τάσης που δημιουργούνται εξ' αιτίας του σφάλματος:

$$Y' \cdot V' = I' - Y'_k \cdot V_k \quad (3.6)$$

Όπου k είναι ο κόμβος σφάλματος, για τον οποίο συμπεριλαμβάνονται και οι τρεις φάσεις p, q, r ,

n είναι ο συνολικός αριθμός των ζυγών,

Y' είναι ο 3×3 τροποποιημένος πίνακας αγωγιμοτήτων, που διαφέρει από τον Y στο ότι δεν περιέχει τις p, q και r γραμμές και στήλες,

V' είναι η ζητούμενη πτώση τάσεων,

$I' = 0$, καθώς το ρεύμα I'_k έχει απαλειφθεί από αυτό το διάνυσμα,

Y'_k είναι οι p, q, r στήλες της μήτρας Y' που όμως δεν περιλαμβάνει τα $Y'_{pp}, Y'_{qq}, Y'_{rr}$,

V_k είναι η προ σφάλματος τάση του κόμβου k συμπεριλαμβανομένων των V'_p, V'_q, V'_r .

Οι μετά σφάλματος τάσεις των ζυγών δίνονται από τη σχέση:

$$E_i = V_i + V'_i \quad i = 1, 2, \dots, n$$

Το ρεύμα σφάλματος προκύπτει από την εξίσωση:

$$I_k = \sum_{i=1}^n Y'_{ki} \cdot E_i \quad k = p, q, r$$

Παρ' όλο που αυτή η μέθοδος ξεκίνησε επιτυχώς την προσέγγιση της ανάλυσης βραχυκυκλωμάτων με υπολογισμούς στο πεδίο των φάσεων, χρησιμοποιούταν μόνο σαν εναλλακτική μέθοδος, καθώς εκείνη την εποχή οι ηλεκτρονικοί υπολογιστές δεν ήταν αρκετά ισχυροί ώστε να πραγματοποιήσουν τον υπολογισμό στο πεδίο των φάσεων.

3.3 Τοπολογική ανάλυση συστήματος

Έχουν αναπτυχθεί διάφορες μέθοδοι για τον καθορισμό της σειράς με την οποία θα ρυθμίζονται οι ηλεκτρονόμοι υπερέντασης- κατεύθυνσης, καθώς μόνο στη περίπτωση που οι ηλεκτρονόμοι ρυθμίζονται με συγκεκριμένη σειρά επιτυγχάνεται γρήγορη σύγκλιση της επαναληπτικής διαδικασίας ρύθμισης.

3.3.1 Break Points

Ο Knable ήταν αυτός που εφηύρε τα επονομαζόμενα Break Points, που είναι οι ηλεκτρονόμοι από τους οποίους πρέπει να ξεκινήσει η διαδικασία ρύθμισης. Ο Damborg και οι συνεργάτες του, καθώς και ο Jenkines, έχουν εφαρμόσει τη μέθοδο της τοπολογικής ανάλυσης (χρησιμοποιώντας γραμμική μέθοδο γράφων και συναρτησιακή κτήση) για να ορίσουν το βέλτιστο σύνολο Break Points. Ο Bapeswara Rao και οι συνεργάτες του χρησιμοποίησαν την έννοια της ελάχιστης κάλυψης στηλών του Πίνακα Βρόχων L_D ¹. Διάφορες πράξεις της άλγεβρας Boole οδηγούν στην ελάχιστη κάλυψη, που είναι η κάλυψη με τον ελάχιστο αριθμό ηλεκτρονόμων.

3.3.2 RSM

Ο Knable έχει προτείνει επίσης ένα ιεραρχικό σχήμα που δίνει αυτή τη συγκεκριμένη αλληλουχία ηλεκτρονόμων. Οι Dwarakanath και Nowitz χρησιμοποίησαν την θεωρία των γράφων για να λάβουν τον Πίνακα Σχετικής Ακολουθίας (Relative Sequence Matrix - RSM), ο οποίος καθορίζει τη βέλτιστη αλληλουχία ρύθμισης των ηλεκτρονόμων. Αυτή η τεχνική αναπτύχθηκε λεπτομερώς από τον Damborg και τους συνεργάτες του, οι οποίοι δημιούργησαν ένα πρόγραμμα CAD για τη ρύθμιση των ηλεκτρονόμων υπερέντασης με στοιχεία κατεύθυνσης.

¹ Ο Πίνακας Βρόχων L_D έχει διάσταση ίση με το συνολικό αριθμό των βρόχων επί το συνολικό αριθμό των ηλεκτρονόμων. Μια γραμμή του L_D δίνει τους ηλεκτρονόμους ενός βρόχου. Ένα στοιχείο $L_D(i, j)$ είναι ίσο με 1 εάν ο ηλεκτρονόμος j περιλαμβάνεται στο βρόχο i .

3.4 Ρύθμιση των ηλεκτρονόμων

Ρύθμιση των ηλεκτρονόμων είναι ο προσδιορισμός του ρεύματος ρύθμισης $I_{pick-up}$ και της χρονικής σταθεράς TDS κάθε ηλεκτρονόμου. Ένα βασικό πρόβλημα που προκύπτει με τους ηλεκτρονόμους υπερέντασης- κατεύθυνσης είναι η περιπλοκότητα που εισάγεται από την ανάγκη συνεργασίας των ηλεκτρονόμων, δηλαδή την ανάγκη σωστής λειτουργίας του ηλεκτρονόμου δευτερεύουσας βαθμίδας σε σχέση με τον ηλεκτρονόμο πρωτεύουσας βαθμίδας. Η περιπλοκότητα είναι μεγαλύτερη σε συστήματα με πολλούς βρόχους και πολλές πηγές. Συνεπώς, από τη δεκαετία του 1960 έχει πραγματοποιηθεί μεγάλη προσπάθεια επίλυσης αυτού του προβλήματος με χρήση προσομοίωσης με ηλεκτρονικούς υπολογιστές.

Αρχικά θα πρέπει να αναφερθούμε στον τύπο των ηλεκτρονόμων που χρησιμοποιούνται σε αυτή τη διπλωματική εργασία. Οι ηλεκτρονόμοι υπερέντασης περιγράφονται από την ακόλουθη IEC χαρακτηριστική:

$$T = \frac{k_1}{M^{k_2} - 1} \cdot TDS \quad (3.7)$$

Όπου

T	ο χρόνος λειτουργίας του ηλεκτρονόμου,
k_1, k_2	σταθερές που δηλώνουν τον τύπο του ηλεκτρονόμου (αντιστρόφου, πολύ αντιστρόφου, εξαιρετικά αντιστρόφου χρόνου κτλ),
$M = \frac{I}{I_{pick-up}}$	ο λόγος του ρεύματος που διαρρέει τον ηλεκτρονόμο προς το ρεύμα ρύθμισης αυτού.

Τρεις είναι οι προσεγγίσεις που έχουν χρησιμοποιηθεί για τη ρύθμιση των ηλεκτρονόμων: οι μέθοδος δοκιμής και λάθους, η τοπολογική ανάλυση και η μέθοδος βελτιστοποίησης.

3.4.1 Μέθοδος δοκιμής και λάθους

Η μέθοδος δοκιμής και λάθους είναι η παραδοσιακή μέθοδος για τη ρύθμιση των ηλεκτρονόμων. Η αναζήτηση των κατάλληλων ρυθμίσεων γίνεται εμπειρικά δοκιμάζοντας διάφορες πιθανές τιμές. Η ελαχιστοποίηση των χρόνων λειτουργίας των ηλεκτρονόμων γίνεται θέτοντας αρχικά τις τιμές τους στην ελάχιστη επιτρεπτή τιμή και αυξάνοντας τις βαθμιαία.

Μειονέκτημα

Η μέθοδος αυτή έχει ένα σημαντικό μειονέκτημα που είναι ο μεγάλος χρόνος που απαιτείται για την εύρεση λύσης.

3.4.2 Μέθοδος τοπολογικής ανάλυσης

Η μέθοδος της τοπολογικής ανάλυσης είναι μια από τις μεθόδους που πρόκειται να χρησιμοποιηθούν στη συνέχεια για τη ρύθμιση των ηλεκτρονόμων. Αναφέρεται εδώ απλά για λόγους πληρότητας, ενώ θα εξηγηθεί στο επόμενο κεφάλαιο.

Πλεονεκτήματα - μειονεκτήματα

Παρά τα πλεονεκτήματα της τοπολογικής προσέγγισης, δεν υπάρχει εγγύηση ότι επιλέγονται οι βέλτιστες ρυθμίσεις για τους ηλεκτρονόμους. Εξασφαλίζει βέβαια την απαιτούμενη συνεργασία, αλλά οι χρόνοι λειτουργίας των ηλεκτρονόμων δεν είναι οι ελάχιστοι δυνατοί.

3.4.3 Μέθοδος βελτιστοποίησης

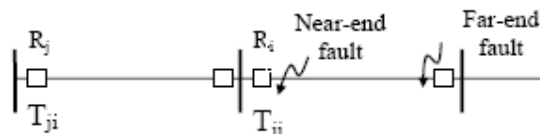
3.4.3.1 Γραμμικός προγραμματισμός

Διατύπωση του προβλήματος γραμμικού προγραμματισμού

Το πρόβλημα ρύθμισης των ηλεκτρονόμων υπερέντασης έτσι ώστε να ακολουθούν τους κανόνες της μεταξύ τους συνεργασίας μπορεί να θεωρηθεί ως ένα παραμετρικό πρόβλημα βελτιστοποίησης. Βελτιστοποιούμε την αντικειμενική συνάρτηση που περιλαμβάνει τους χρόνους λειτουργίας των primary ηλεκτρονόμων, ενώ παράλληλα λαμβάνουμε υπόψη τους περιορισμούς που εκφράζουν τη συνεργασία πρωτεύουσας και δευτερεύουσας προστασίας. Μια πιθανή προσέγγιση του προβλήματος θα ήταν να ελαχιστοποιήσουμε το άθροισμα των χρόνων λειτουργίας όλων των primary ηλεκτρονόμων, ελπίζοντας ότι με την κίνηση αυτή και ο χρόνος λειτουργίας κάθε primary ηλεκτρονόμου θα είναι κοντά στην ελάχιστη δυνατή τιμή του. Θεωρώντας ένα σύστημα με n ηλεκτρονόμους, η αρχική αντικειμενική συνάρτηση J , η οποία πρόκειται να ελαχιστοποιηθεί είναι:

$$J = \sum_i T_{ii} \quad (3.8)$$

Όπου T_{ii} είναι ο χρόνος λειτουργίας του ηλεκτρονόμου R_i πρωτεύουσας προστασίας για ένα σφάλμα αμέσως μετά τον ηλεκτρονόμο αυτό (close-in ή near-end fault), όπως φαίνεται και στο παρακάτω σχήμα.



Σχήμα 4 : Χρόνοι λειτουργίας πρωτεύοντος και δευτερεύοντος ηλεκτρονόμου

Όπως αναφέρθηκε και παραπάνω, ο χρόνος λειτουργίας του ηλεκτρονόμου δευτερεύουσας προστασίας (back up) πρέπει να είναι μεγαλύτερος από το άθροισμα του χρόνου λειτουργίας του ηλεκτρονόμου πρωτεύουσας προστασίας (primary) και του χρονικού περιθωρίου μεταξύ των χρόνων των δυο ηλεκτρονόμων. Αυτός ο περιορισμός εκφράζεται από τη ανίσωση:

$$T_{ji} \geq T_{ii} + CTI \quad (3.9)$$

όπου T_{ji} είναι ο χρόνος λειτουργίας του ηλεκτρονόμου R_j που προσφέρει δευτερεύουσα προστασία για το ίδιο close-in σφάλμα i και CTI είναι το χρονικό περιθώριο πρωτεύουσας – δευτερεύουσας προστασίας.

Στην εξίσωση (3.7) είναι εμφανές ότι η σχέση μεταξύ του χρόνου λειτουργίας T του ηλεκτρονόμου υπερέντασης και του πολλαπλασίου του ρεύματος ρύθμισης M είναι μη γραμμική. Αφού όμως το ρεύμα ρύθμισης των ηλεκτρονόμων μπορεί να προκαθοριστεί, η εξίσωση (3.7) γίνεται γραμμική για κάθε τιμή του ρεύματος σφάλματος. Η γραμμικότητα εκφράζεται από τη σχέση:

$$T = a \cdot TDS$$

$$\text{όπου } a = \frac{k_1}{M^{k_2} - 1}.$$

Με αντικατάσταση στη σχέση (3.8) η αντικειμενική συνάρτηση γράφεται:

$$J = \sum_{i=1}^n a_i \cdot TDS_i \quad (3.10)$$

Το πρόβλημα ρύθμισης των TDS των ηλεκτρονόμων είναι ένα πρόβλημα γραμμικού προγραμματισμού με αντικειμενική συνάρτηση την (3.10), η οποία ελαχιστοποιείται, και περιορισμούς που εκφράζονται από την εξίσωση (3.9). Επιπλέον υπάρχουν και περιορισμοί που θέτουν τις επιτρεπόμενες τιμές των TDS , οι οποίοι εκφράζονται από τη σχέση:

$$TDS_{\min} \leq TDS_i \leq TDS_{\max}$$

Συνεπώς το πρόβλημα γραμμικού προγραμματισμού διατυπώνεται ως εξής:

$$\begin{aligned} \min & \left(J = \sum_{i=1}^n a_i \cdot TDS_i \right) \\ & T_{ji} \geq T_{ii} + CTI \\ & TDS_{\min} \leq TDS_i \leq TDS_{\max} \end{aligned} \quad (3.11)$$

Προσεγγίσεις του προβλήματος γραμμικού προγραμματισμού

Κάποιοι ερευνητές εφάρμοσαν την τεχνική του γραμμικού προγραμματισμού για να ελαχιστοποιήσουν τις χρονικές σταθερές των ηλεκτρονόμων, ενώ τα ρεύματα αναφοράς επιλέχθηκαν εμπειρικά. Το πρόβλημα γραμμικού προγραμματισμού λύθηκε με τη μέθοδο Simplex, Dual Simplex ή Two Phase Simplex.

Οι Laway και Gubta χρησιμοποίησαν γραμμικό προγραμματισμό για τον προσδιορισμό των TDS . Έλαβα υπόψη για κάθε ηλεκτρονόμο δυο περιορισμούς σχετικούς με τη συνεργασία, έναν για σφάλμα αμέσως μετά τον ηλεκτρονόμο (near

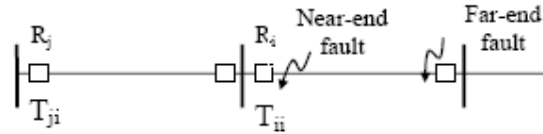
end / close-in fault) και έναν για σφάλμα στο τέλος της γραμμής (far-end fault). Τα σφάλματα αυτά φαίνονται στο Σχήμα 5 που ακολουθεί. Η αντικειμενική συνάρτηση είναι άθροισμα δυο προσθετέων: ο πρώτος είναι το σταθμισμένο άθροισμα των χρόνων λειτουργίας των ηλεκτρονόμων πρωτεύουσας προστασίας T_{ii} για close-in σφάλμα και ο δεύτερος είναι σταθμισμένο άθροισμα των χρόνων λειτουργίας των ηλεκτρονόμων πρωτεύουσας προστασίας για far-end σφάλμα. Αφού όμως οι γραμμές είναι μικρές και του ίδιου σχεδόν μήκους, χρησιμοποιούνται τα ίδια βάρη (=1) στα σταθμισμένα αθροίσματα της αντικειμενικής συνάρτησης. Η εξίσωση (3.11) λαμβάνει λοιπόν την εξής μορφή:

$$\min \left(J = \sum_{i=1}^n T_{ii, near-end} + \sum_{i=1}^n T_{ii, far-end} \right) \Leftrightarrow \min \left(J = \sum_{i=1}^n a_i \cdot TDS_{i, near-end} + \sum_{i=1}^n a_i \cdot TDS_{i, far-end} \right)$$

$$T_{ji, near-end} \geq T_{ii, near-end} + CTI$$

$$T_{ji, far-end} \geq T_{ii, far-end} + CTI$$

$$TDS_{\min} \leq TDS_i \leq TDS_{\max}$$



Σχήμα 5 : Μεγέθη χρησιμοποιούμενα από το πρόβλημα γραμμικού προγραμματισμού

Αντί για ένα σφάλμα στην αρχή της γραμμής και ένα σφάλμα στο τέλος της, ο Urdenta και οι συνεργάτες του θεώρησαν σφάλμα στο μέσο της γραμμής για να μειώσουν τη διάσταση του προβλήματος.

Αργότερα ο Urdenta και οι συνεργάτες του πρότειναν τέσσερις περιορισμούς για κάθε ηλεκτρονόμο, τους δυο που χρησιμοποίησαν οι Laway και Gubta και δυο για τις τοπολογικές αλλαγές κατά τη μεταβατική κατάσταση λειτουργίας. Τους τέσσερις αυτούς περιορισμούς χρησιμοποίησαν και οι Perez και Urdaneta στη δική τους προσέγγιση [9] της επίλυσης με γραμμικό προγραμματισμό. Με αυτόν τον τρόπο όμως, η διάσταση του προβλήματος, που ήταν ήδη μεγάλη, αυξήθηκε περισσότερο.

3.4.3.2 Μη γραμμικός προγραμματισμός

1. Για την εύρεση των TDS

Το πρόβλημα του προσδιορισμού των παραμέτρων των ηλεκτρονόμων υπερέντασης έχει λάβει και μη γραμμική μορφή [8]. Η περίπτωση αυτή διαφέρει από την προηγούμενη στο ότι τα ρεύματα ρύθμισης δεν προεπιλέγονται. Η χαρακτηριστική

του ηλεκτρονόμου $T = \frac{k_1}{M^{k_2} - 1} \cdot TDS$ μπορεί να γραφεί στη μορφή:

$$T = f(I_{pick-up}, I) \cdot TDS$$

Χρησιμοποιώντας το μη γραμμικό μοντέλο του Sachdev για τους συντελεστές $f(I_{pick-up,i}, I_i)$, το οποίο εκφράζει το ρεύμα ρύθμισης κάθε ηλεκτρονόμου σαν μη γραμμική συνάρτηση της χρονικής του σταθεράς TDS , η αντικειμενική συνάρτηση του προβλήματος δίνεται από τη σχέση:

$$J = \sum_i T_i = \sum_i f(I_{pick-up,i}, I_i) \cdot TDS_i = \sum_i (b_0 + b_1 \cdot TDS_i + b_2 \cdot TDS_i^2 + b_3 \cdot TDS_i^3 + b_4 \cdot TDS_i^4)$$

Στη συνάρτηση δηλαδή περιλαμβάνονται τα TDS των ηλεκτρονόμων και έχει μη γραμμική μορφή. Οι μεταβλητές της συνάρτησης αυτής υπόκεινται στους εξής περιορισμούς:

$$TDS_{\min} \leq TDS_i \leq TDS_{\max}$$

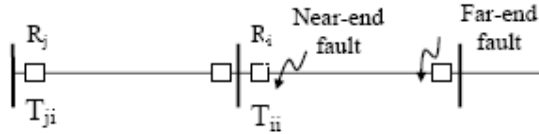
$$T_j - T_i \geq CTI \Leftrightarrow f(I_{pick-up,j}, I_j) \cdot TDS_j - f(I_{pick-up,i}, I_i) \cdot TDS_i \geq CTI$$

Όπου T_i ο χρόνος λειτουργίας κάθε πρωτεύοντος ηλεκτρονόμου,

T_j ο χρόνος λειτουργίας κάθε δευτερεύοντος ηλεκτρονόμου,

CTI το χρονικό περιθώριο δευτερεύουσας – πρωτεύουσας προστασίας.

Τα μεγέθη αυτά φαίνονται χαρακτηριστικά και στο ακόλουθο σχήμα.



Σχήμα 6 : Χρησιμοποιούμενα μεγέθη

Μειονέκτημα

Το μειονέκτημα του μη γραμμικού προγραμματισμού είναι ότι πρόκειται για μια χρονοβόρα τεχνική. Αυτό συμβαίνει επειδή για κάθε επανάληψη του μη γραμμικού προγραμματισμού καλείται ένας άλλος γραμμικός προγραμματισμός για να καθοριστεί η κατεύθυνση αναζήτησης.

2. Για την εύρεση των $I_{pick-up}$ και των TDS

Κάποιοι ερευνητές χρησιμοποίησαν μη γραμμικό προγραμματισμό για να καθορίσουν τις βέλτιστες τιμές των ρευμάτων αναφοράς και γραμμικό προγραμματισμό (μέθοδος Simplex) για βελτιστοποιήσουν τις τιμές των χρονικών σταθερών TDS , λαμβάνοντας πάντα υπόψιν τους περιορισμούς σχετικά με τη συνεργασία των ηλεκτρονόμων.

Άλλοι περιέλαβαν τόσο τα ρεύματα ρύθμισης όσο και τις χρονικές σταθερές των ηλεκτρονόμων στο ίδιο πρόβλημα μη γραμμικού προγραμματισμού σύμφωνα με το γεννητικό αλγόριθμο. Η αντικειμενική συνάρτηση στην περίπτωση αυτή έχει, σύμφωνα με το [7], την ακόλουθη μορφή:

$$J = \left(\chi \cdot \sum_i T_i \right) + \left(\beta \cdot \sum_i (TDS_i - TDS_{pref,i})^2 \right) + \left(\alpha \cdot \sum_i (I_{p-up,i} - I_{p-up_pref,i})^2 \right) + \left(\delta \cdot \sum (T_j - T_i - CTI)^2 \right)$$

Με $TDS_{pref,i}$ το επιθυμούμενο από το χρήστη TDS για τον ηλεκτρονόμο i ,

$I_{p-up_pref,i}$ το επιθυμούμενο από το χρήστη $I_{pick-up}$ τον ηλεκτρονόμο i και

$j - i$ το κάθε ζεύγος δευτερεύουσας – πρωτεύουσας προστασίας.

Είναι φανερό ότι οι περιορισμοί έχουν προστεθεί στην αντικειμενική συνάρτηση ύστερα από ύψωση στο τετράγωνο και πολλαπλασιασμό με τον αντίστοιχο παράγοντα βάρους.

Πλεονεκτήματα - μειονεκτήματα

Το βασικό πλεονέκτημα αυτής της μεθόδου είναι ότι τα ρεύματα ρύθμισης των ηλεκτρονόμων δεν είναι προκαθορισμένα, αλλά υπολογίζονται. Έτσι, λαμβάνουμε μια διαφορετική βέλτιστη λύση, καλύτερη από αυτή που λαμβάνουμε στην περίπτωση που τα ρεύματα ρύθμισης είναι προκαθορισμένα.

Στα μειονεκτήματα θα πρέπει να συμπεριληφθεί η πολυπλοκότητα που εισάγει η μεθοδολογία αυτή. Επίσης, λόγω της μη γραμμικής μορφής του προβλήματος, υπάρχει η πιθανότητα να παγιδευτεί το πρόβλημα σε μια τοπική βέλτιστη λύση.

Σε αυτή τη διπλωματική δε θα μας απασχολήσουν μέθοδοι μη γραμμικού προγραμματισμού. Από τις μεθόδους βελτιστοποίησης θα χρησιμοποιηθεί μια μέθοδος γραμμικού προγραμματισμού, η οποία θα παρουσιαστεί στη συνέχεια στο αντίστοιχο κεφάλαιο.

4

Χρησιμοποιούμενες τεχνικές

Ύστερα από την παρουσίαση των εναλλακτικών μεθόδων, θα παρουσιαστούν οι τεχνικές που ακολουθήθηκαν στην εργασία αυτή. Θα παρατεθεί το θεωρητικό υπόβαθρο κάθε τεχνικής, όπου αυτό υπάρχει, και θα εξηγηθεί η μεθοδολογία η οποία ακολουθείται για την ολοκλήρωση κάθε μιας από τις τέσσερις επιμέρους διαδικασίες.

4.1 Ανάλυση ροών φορτίου

4.1.1 Μέθοδος Newton – Raphson

Όπως αναφέρθηκε και στην παράγραφο 3.1, η μέθοδος ανάλυσης ροών φορτίου που θα χρησιμοποιηθεί στη διπλωματική αυτή εργασία είναι η επαναληπτική μέθοδος Newton – Raphson. Η μέθοδος αυτή είναι πιο σύνθετη από τη μέθοδο Gauss – Siedel και απαιτεί σημαντική μνήμη υπολογιστή, προσφέρει όμως τα ακόλουθα πλεονεκτήματα:

- Στις περισσότερες περιπτώσεις δεν έχει πρόβλημα σύγκλισης.
- Κατά κανόνα συγκλίνει ταχύτερα από τη μέθοδο Gauss – Siedel.

Έστω προς επίλυση το σύστημα των εξισώσεων

$$f_1(x_1, x_2, \dots, x_n) = 0$$

$$f_2(x_1, x_2, \dots, x_n) = 0$$

...

$$f_n(x_1, x_2, \dots, x_n) = 0$$

Θεωρώντας το ανάπτυγμα Taylor όλων των συναρτήσεων f διαμορφώνουμε το γενικό σύστημα των εξισώσεων:

$$[\Delta f] = [J][\Delta x]$$
$$\text{Όπου } [J] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \text{ η Ιακωβιανή μήτρα του συστήματος.}$$

Στη περίπτωση των ροών φορτίου οι μεταβλητές x και f παριστάνουν τα ακόλουθα μεγέθη:

$$[x] = \begin{bmatrix} \delta_i \\ |\tilde{V}_i| \end{bmatrix} = \begin{bmatrix} \delta_1 \\ \dots \\ \delta_n \\ |\tilde{V}_1| \\ \dots \\ |\tilde{V}_n| \end{bmatrix} \text{ το διάνυσμα καταστάσεως}$$

$$\text{με } [x_0] = \begin{bmatrix} \delta_{0i} \\ |\tilde{V}_{0i}| \end{bmatrix} \text{ το διάνυσμα αρχικών τιμών,}$$

$$[f(x)] = \begin{bmatrix} P_{Si} \\ Q_{Si} \end{bmatrix} = \begin{bmatrix} P_{S1} \\ \dots \\ P_{Sk} \\ Q_{S1} \\ \dots \\ Q_{Sk} \end{bmatrix} \text{ το διάνυσμα των καθαρών ισχύων}$$

$$\text{με } [f(x_0)] = \begin{bmatrix} P_i \\ Q_i \end{bmatrix} \text{ το διάνυσμα των κάθε φορά προσεγγίσεων των ισχύων P και Q.}$$

Η Ιακωβιανή μήτρα έχει μορφή:

$$[J] = \begin{bmatrix} \frac{\partial P}{\partial \delta} & \frac{\partial P}{\partial |V|} \\ \frac{\partial Q}{\partial \delta} & \frac{\partial Q}{\partial |V|} \end{bmatrix} = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix} \quad (4.1)$$

Θυμίζουμε στο σημείο αυτό τις εξισώσεις ροών φορτίου που υπολογίστηκαν στην παράγραφο 2.3.

$$P_i = |V_i| \cdot \sum_{j=i}^N |V_j| \left[g_{ji} \cos(\delta_j - \delta_i) - b_{ji} \sin(\delta_j - \delta_i) \right]$$

$$-Q_i = |V_i| \cdot \sum_{j=i}^N |V_j| \left[g_{ji} \sin(\delta_j - \delta_i) + b_{ji} \cos(\delta_j - \delta_i) \right]$$

Θεωρούμε $\tilde{V}_i = |\tilde{V}_i| \angle \delta_i$ και $\tilde{y}_{ij} = |\tilde{y}_{ij}| \angle -\theta_{ij}$, οπότε οι εξισώσεις ροών φορτίου γράφονται:

$$P_i = |\tilde{V}_i^2 y_{ii}| \cos(\theta_{ii}) + \sum_{\substack{j=1 \\ j \neq i}}^N |\tilde{V}_i y_{ij} \tilde{V}_j| \cos(\theta_{ij} + \delta_i - \delta_j) \quad (4.2)$$

$$Q_i = |\tilde{V}_i^2 y_{ii}| \sin(\theta_{ii}) + \sum_{\substack{j=1 \\ j \neq i}}^N |\tilde{V}_i y_{ij} \tilde{V}_j| \sin(\theta_{ij} + \delta_i - \delta_j) \quad (4.3)$$

Χρησιμοποιούμε αυτές τις εξισώσεις για να υπολογίσουμε τις μερικές παραγώγους της Ιακωβιανής μήτρας, οπότε για παράδειγμα για το στοιχείο J_{1ij} έχουμε:

$$J_{1ij} = \frac{\partial P_i}{\partial \delta_j} = |\tilde{V}_i \tilde{V}_j y_{ii}| \sin(\theta_{ij} + \delta_i - \delta_j) \quad , j \neq i$$

Στην πραγματικότητα όμως δε χρειάζεται να υπολογίσουμε όλες τις παραγώγους, γιατί στο ζυγό αναφοράς τα $|\tilde{V}|$ και δ είναι γνωστά, όπως στους ζυγούς παραγωγής τα $|\tilde{V}|$, οπότε δεν απαιτείται να τα υπολογίσουμε. Έτσι, εάν αριθμήσουμε τους ζυγούς ώστε ο ζυγός 1 να είναι ο ζυγός αναφοράς, οι ζυγοί 2 έως m να είναι ζυγοί παραγωγής και οι (m+1) έως N ζυγοί φορτίου η Ιακωβιανή μήτρα που μας χρειάζεται έχει διαστάσεις $(2N - m - 1) \times (2N - m - 1)$ και ισούται με:

$$\begin{bmatrix} \Delta P_2 \\ \dots \\ \Delta P_N \\ \Delta Q_{m+1} \\ \dots \\ \Delta Q_N \end{bmatrix} = \begin{bmatrix} \frac{\partial P_2}{\partial \delta_2} & \dots & \frac{\partial P_2}{\partial \delta_N} & \frac{\partial P_2}{\partial |V|_{m+1}} & \dots & \frac{\partial P_2}{\partial |V|_N} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial P_N}{\partial \delta_2} & \dots & \frac{\partial P_N}{\partial \delta_N} & \frac{\partial P_N}{\partial |V|_{m+1}} & \dots & \frac{\partial P_N}{\partial |V|_N} \\ \frac{\partial Q_{m+1}}{\partial \delta_2} & \dots & \frac{\partial Q_{m+1}}{\partial \delta_N} & \frac{\partial Q_{m+1}}{\partial |V|_{m+1}} & \dots & \frac{\partial Q_{m+1}}{\partial |V|_N} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial Q_N}{\partial \delta_2} & \dots & \frac{\partial Q_N}{\partial \delta_N} & \frac{\partial Q_N}{\partial |V|_{m+1}} & \dots & \frac{\partial Q_N}{\partial |V|_N} \end{bmatrix} \begin{bmatrix} \Delta \delta_2 \\ \dots \\ \Delta \delta_N \\ \Delta |\tilde{V}|_{m+1} \\ \dots \\ \Delta |\tilde{V}|_N \end{bmatrix}$$

$$\text{Η συνοπτικά} \begin{bmatrix} [\Delta P] \\ [\Delta Q] \end{bmatrix} = \begin{bmatrix} \left[\frac{\partial P}{\partial \delta} \right] & \left[\frac{\partial P}{\partial |V|} \right] \\ \left[\frac{\partial Q}{\partial \delta} \right] & \left[\frac{\partial Q}{\partial |\tilde{V}|} \right] \end{bmatrix} \begin{bmatrix} [\Delta \delta] \\ [\Delta |\tilde{V}|] \end{bmatrix} \quad (4.4)$$

Συνοπτικά η μέθοδος Newton – Raphson περιλαμβάνει τα ακόλουθα βήματα:

- 1) Δίνονται αρχικές τιμές στα $|\tilde{V}|$ και δ , όπου αυτά δεν είναι καθορισμένα (διάνυσμα $[x_0]$).
- 2) Υπολογίζονται οι ισχείς των ζυγών από τις εξισώσεις (4.2), (4.3). Είναι γνωστά τα P για όλους τους ζυγούς πλην του ζυγού αναφοράς και τα Q για τους ζυγούς φορτίου. Συνεπώς χρειάζεται ο υπολογισμός των $P_2, \dots, P_N$ και μόνο των $Q_{m+1}, \dots, Q_N$ που θα δώσουν τις αποκλίσεις $[\Delta P]$ και $[\Delta Q]$ που χρειάζεται η (4.4).
- 3) Υπολογίζεται η Ιακωβιανή από τη σχέση (4.1).

- 4) Επιλύεται το σύστημα (4.4) και υπολογίζονται οι αποκλίσεις $[\Delta V]$, $[\Delta \delta]$, δηλαδή οι διορθώσεις $[\Delta x]$.
- 5) Αν οι αποκλίσεις είναι μεγαλύτερες από τα όρια ανοχής, επανερχόμαστε στο βήμα 2.
- 6) Μετά τη σύγκλιση των αποτελεσμάτων υπολογίζονται τα P_i , Q_i του ζυγού αναφοράς.

4.1.2 Υπολογισμός ρευμάτων

Με την ολοκλήρωση της διαδικασίας που ορίζει η μέθοδος Newton – Raphson είναι γνωστές οι ισχείς και οι τάσεις σε όλους τους κόμβους. Χρησιμοποιούμε τις τάσεις αυτές για να υπολογίσουμε το ρεύμα που ρέει σε κάθε γραμμή του συστήματος. Αυτές είναι οι μεταβλητές που μας ενδιαφέρουν, γιατί αυτές θα χρειαστούμε στη συνέχεια για τη ρύθμιση των ηλεκτρονόμων.

Έστω δυο ζυγοί i , j του συστήματος, οι τάσεις $\tilde{V}_i$ και $\tilde{V}_j$ των οποίων έχουν υπολογιστεί από την ανάλυση ροών φορτίου. Το ρεύμα που ρέει στον κλάδο $i - j$ δίνεται από τη σχέση:

$$\tilde{I}_{ij} = \tilde{Y}_{ij} (\tilde{V}_i - \tilde{V}_j)$$

Όπου $\tilde{Y}_{ij}$ η αγωγιμότητα του κλάδου $i-j$.

4.2 Ανάλυση βραχυκυκλωμάτων

Ο αλγόριθμος για την επίλυση βραχυκυκλωμάτων που θα χρησιμοποιηθεί στη συνέχεια παρουσιάστηκε από τον Jun Zhu το 2004 [17]. Ο αλγόριθμος αυτός περιλαμβάνει υπολογισμούς αυστηρά στο πεδίο των φάσεων. Δεν περιλαμβάνει τα ακολουθιακά κυκλώματα, καθώς δημιουργείται ο τριφασικός πίνακας αγωγιμοτήτων. Αφού πραγματοποιεί τους υπολογισμούς σφάλματος στο πεδίο των φάσεων, μπορεί να αντιμετωπίσει περίπλοκες περιπτώσεις βραχυκυκλωμάτων. Αυτή η μέθοδος δεν τροποποιεί τον πίνακα αγωγιμοτήτων, όπως άλλες μέθοδοι, αλλά χρησιμοποιεί τον πίνακα υπο-αντιστάσεων του ζυγού σφάλματος για να υπολογίσει το ρεύμα σφάλματος. Οι λεπτομερείς υπολογισμοί παρουσιάζονται στη συνέχεια.

4.2.1 Τριφασικός πίνακας αγωγιμοτήτων Y

Αφού έχουν μοντελοποιηθεί όλα τα στοιχεία του συστήματος, μπορούν να υπολογιστούν οι πίνακες αγωγιμοτήτων για τα τρία ακολουθιακά κυκλώματα. Πρώτα δημιουργούνται οι αρχικοί πίνακες αγωγιμοτήτων. Για παράδειγμα, ο αρχικός πίνακας σύνθετων αντιστάσεων μηδενικής ακολουθίας είναι:

$$z_0 = \begin{bmatrix} z_{r1} & 0 & 0 & \dots & 0 \\ 0 & z_{r2} & z_{mu} & \dots & 0 \\ 0 & z_{mu} & z_{r3} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & z_{rm} \end{bmatrix} \quad (4.5)$$

Όπου z_r είναι η σειριακή αγωγιμότητα της γραμμής, z_{mu} είναι η αμοιβαία αγωγιμότητα των γραμμών 2 και 3 και m είναι ο συνολικός αριθμός των γραμμών. Εάν η γραμμή p είναι κλάδος μετασχηματιστή, τότε η z_{rp} ισούται με 1.0.

Ο αρχικός πίνακας αγωγιμοτήτων μηδενικής ακολουθίας είναι συνεπώς:

$$y_0 = z_0^{-1}$$

και εάν η γραμμή p είναι κλάδος μετασχηματιστή, τότε $y_{pp} = 1$. Αυτό γίνεται για να είμαστε βέβαιοι ότι το στοιχείο του κλάδου μετασχηματιστή δε θα επηρεάσει τα άλλα στοιχεία του πίνακα κατά την πραγματοποίηση του γινομένου της εξίσωσης (4.7).

Η πλήρης μήτρα πρόσπτωσης ορίζεται ως

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{11} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad (4.6)$$

Όπου

$$a_{pq} = \begin{cases} 1 & \text{εάν q είναι ο ζυγός αφετηρίας του κλάδου p} \\ -1 & \text{εάν q είναι ο ζυγός τερματισμού του κλάδου p} \\ 0 & \text{αλλιώς} \end{cases}$$

Με m τον αριθμό των γραμμών και n τον αριθμό των ζυγών.

Τώρα, μπορεί να υπολογιστεί από την παρακάτω εξίσωση ο πίνακας αγωγιμοτήτων για κάθε ακολουθία i:

$$Y_{l-i} = A' \cdot y_i \cdot A \quad (4.7)$$

Σημειωτέον ότι η μήτρα αυτή δεν περιλαμβάνει τα παράλληλα στοιχεία, τις γεννήτριες και τις παράλληλες χωρητικότητες των γραμμών μεταφοράς. Προσθέτοντας τις παράλληλες χωρητικότητες των γραμμών έχουμε

$$Y_s = \begin{bmatrix} Y_{l-pp} + y_s & Y_{s-pq} - y_s \\ Y_{s-qp} - y_s & Y_{l-qq} + y_s \end{bmatrix}$$

Όπου p και q είναι οι κόμβοι της γραμμής και y_s είναι η παράλληλη αγωγιμότητα της γραμμής αυτής.

Προσθέτοντας τα παράλληλα στοιχεία και τις γεννήτριες,

$$Y_{pp} = Y_{s-pp} + y_g$$

Όπου p είναι ο κόμβος όπου συνδέεται το παράλληλο στοιχείο ή η γεννήτρια και y_g είναι η αγωγιμότητα του παράλληλου στοιχείου ή της γεννήτριας. Σημειώνεται ότι εάν η συνδεσμολογία της γεννήτριας δεν είναι αστέρας γειωμένος, η αγωγιμότητα μηδενικής ακολουθίας y_{g0} της γεννήτριας έχει μηδενική τιμή.

Εκτελώντας τα παραπάνω βήματα και για τις τρεις ακολουθίες, λαμβάνουμε τους πίνακες αγωγιμοτήτων Y_0 , Y_1 και Y_2 μη συμπεριλαμβανομένων των κλάδων μετασχηματιστών.

Στη συνέχεια αυτοί οι πίνακες αγωγιμοτήτων μετασχηματίζονται στο πεδίο των φάσεων. Για κάθε ζυγό, για παράδειγμα το ζυγό 1, το διαγώνιο στοιχείο του πίνακα αγωγιμοτήτων στο φασικό πεδίο έχει τη μορφή:

$$Y_{11}^{(a,b,c)} = \begin{bmatrix} Y_{a1a1} & Y_{a1b1} & Y_{a1c1} \\ Y_{b1a1} & Y_{b1b1} & Y_{b1c1} \\ Y_{c1a1} & Y_{c1b1} & Y_{c1c1} \end{bmatrix} = T \cdot \begin{bmatrix} Y_{0-11} & 0 & 0 \\ 0 & Y_{1-11} & 0 \\ 0 & 0 & Y_{2-11} \end{bmatrix} \cdot T^{-1} \quad (4.8)$$

Όπου Y_{0-11} το στοιχείο στη θέση (1,1) του πίνακα αγωγιμοτήτων μηδενικής ακολουθίας και T ο συμμετρικός πίνακας μετασχηματισμού:

$$T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & a^2 & a \\ 1 & a & a^2 \end{bmatrix}$$

Όπου $a = 1 \angle 120^\circ$.

Έτσι, ολόκληρος ο τριφασικός πίνακας αγωγιμοτήτων είναι:

$$Y_{abc} = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1n} \\ Y_{21} & Y_{22} & \dots & Y_{2n} \\ \dots & \dots & \dots & \dots \\ Y_{n1} & Y_{n2} & \dots & Y_{nn} \end{bmatrix} \quad (4.9)$$

Ο τελικός πίνακας αγωγιμοτήτων λαμβάνεται συνδυάζοντας τον πίνακα Y_{abc} με όλους τους πίνακες αγωγιμοτήτων μετασχηματιστών.

$$Y_{final} = Y_{abc} + \sum_{j=1}^N Y'_{Tj}$$

$$\text{Όπου } Y'_{Tj} = \begin{bmatrix} p_j \left[\frac{1}{\alpha_j^2} \cdot Y_{A_j} & -\frac{1}{\alpha_j \beta_j} \cdot Y_{C_j} \right] \\ q_j \left[-\frac{1}{\alpha_j \beta_j} \cdot Y_{C_j}^T & \frac{1}{\beta_j^2} \cdot Y_{B_j} \right] \end{bmatrix}$$

N : ο αριθμός των μετασχηματιστών

Y_{A_j} , Y_{B_j} , Y_{C_j} : παράμετροι του μετασχηματιστή j

p_j , q_j : οι τερματικοί ζυγοί του μετασχηματιστή j

Όπου α και β είναι τα taps του πρωτεύοντος και του δευτερεύοντος αντίστοιχα.

Οι πίνακες Y_A , Y_B , Y_C δίνονται από τους πίνακες Y_I , Y_{II} , Y_{III} ανάλογα με τη συνδεσμολογία του μετασχηματιστή, όπως δίνεται από τον ακόλουθο πίνακα:

Συνδεσμολογία	Y_A	Y_B	Y_C
$Y_g - Y_g$	Y_I	Y_I	Y_I
$Y_g - Y$	Y_I	Y_{II}	Y_{II}
$Y_g - D$	Y_I	Y_{II}	Y_{III}
$Y - Y$	Y_{II}	Y_{II}	Y_{II}
$Y - D$	Y_{II}	Y_{II}	Y_{III}
$D - D$	Y_{II}	Y_{II}	Y_{II}

Πίνακας 1: Ορισμός των Y_A , Y_B , Y_C για διάφορες συνδεσμολογίες μετασχηματιστή

Οι πίνακες Y_I , Y_{II} , Y_{III} ορίζονται ως εξής:

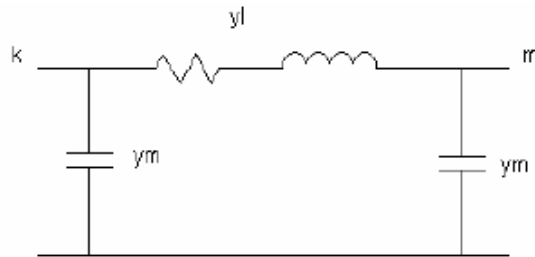
$$Y_I = \frac{1}{3} \begin{bmatrix} y_0 + 2y_1 & y_0 - y_1 & y_0 - y_1 \\ y_0 - y_1 & y_0 + 2y_1 & y_0 - y_1 \\ y_0 - y_1 & y_0 - y_1 & y_0 + 2y_1 \end{bmatrix}$$

$$Y_{II} = \frac{1}{3} \begin{bmatrix} 2y_1 & -y_1 & -y_1 \\ -y_1 & 2y_1 & -y_1 \\ -y_1 & -y_1 & 2y_1 \end{bmatrix}$$

$$Y_{III} = \frac{1}{\sqrt{3}} \begin{bmatrix} y_1 & -y_1 & 0 \\ 0 & y_1 & -y_1 \\ -y_1 & 0 & y_1 \end{bmatrix}$$

Όπου για τους πίνακες Y_A και Y_B είναι $y_1 = y_{l1} + \frac{1}{2}y_{m1}$ και για τον πίνακα Y_C είναι

$y_1 = y_{l1}$. Το y_0 είναι πάντα ίσο με y_{l0} . Οι δείκτες 1 και 0 αναφέρονται στη θετική και τη μηδενική ακολουθία αντίστοιχα. Σε κάθε ακολουθία ο μετασχηματιστής μοντελοποιείται όπως φαίνεται στο Σχήμα 7, όπου y_l η εν σειρά αγωγιμότητα και y_m η παράλληλη αγωγιμότητα στην αντίστοιχη ακολουθία.



Σχήμα 7: Μοντέλο μετασχηματιστή

Έτσι, ο τελικός τριφασικός πίνακας αγωγιμοτήτων Y_{final} , που στο εξής θα αναφέρεται απλά ως Y , έχει υπολογιστεί.

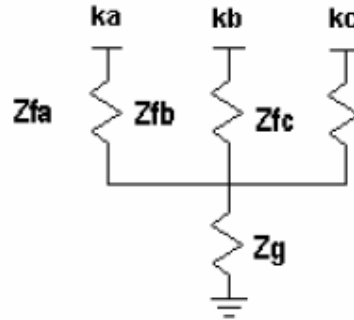
4.2.2 Υπολογισμός τριφασικών σφαλμάτων

Έστω ότι το σφάλμα συμβαίνει στο ζυγό k . Οι γραμμές και οι στήλες που αντιστοιχούν στο ζυγό σφάλματος τοποθετούνται τελευταίες στον τριφασικό πίνακα αγωγιμοτήτων Y .

$$Y' = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1k} \\ Y_{21} & Y_{22} & \dots & Y_{2k} \\ \dots & \dots & \dots & \dots \\ Y_{k1} & Y_{k2} & \dots & Y_k \end{bmatrix}$$

Εφαρμόζοντας παραγοντοποίηση LU στον πίνακα Y' λαμβάνουμε τον 3×3 υποπίνακα σύνθετων αντιστάσεων Z_{kk} .

Στο Σχήμα 8 απεικονίζεται η αντίσταση σφάλματος για διάφορα είδη σφαλμάτων στο ζυγό k .



Σχήμα 8 : Αντίσταση σφάλματος

Ο τρόπος υπολογισμού του ρεύματος βραχυκύκλωσης διαφέρει ανάλογα με το εάν πρόκειται για σφάλμα προς γη ή όχι. Στη συνέχεια παρουσιάζεται μόνο η περίπτωση των σφαλμάτων προς γη, αφού τα σφάλματα που μας ενδιαφέρουν είναι μόνο τα τριφασικά προς γη.

Τα ρεύματα σφάλματος για τις τρεις φάσεις I_{fa} , I_{fb} , I_{fc} υπολογίζονται από την ακόλουθη εξίσωση:

$$\begin{bmatrix} I_{fa} \\ I_{fb} \\ I_{fc} \end{bmatrix} = \begin{bmatrix} Z_{akak} + Z_{fa} & Z_{akbk} & Z_{akck} \\ Z_{bkak} & Z_{bkbk} + Z_{fb} & Z_{bkck} \\ Z_{ckak} & Z_{ckbk} & Z_{ckck} + Z_{fc} \end{bmatrix}^{-1} \cdot \begin{bmatrix} V_a - Z_g (I_{fa} + I_{fb} + I_{fc}) \\ V_b - Z_g (I_{fa} + I_{fb} + I_{fc}) \\ V_c - Z_g (I_{fa} + I_{fb} + I_{fc}) \end{bmatrix} \quad (4.10)$$

Όπου k ο ζυγός σφάλματος και Z_g η αντίσταση σφάλματος μεταξύ του ουδετέρου και της γης, όπως φαίνεται και στο Σχήμα 8.

Αφού έχουν υπολογιστεί τα ρεύματα σφάλματος, οι μεταβολές των τάσεων μετά την εκδήλωση του σφάλματος προκύπτουν από την εξίσωση:

$$Y \cdot \Delta V = -I_f$$

Εφαρμόζοντας παραγοντοποίηση LU στον πίνακα Y έχουμε:

$$L \cdot U \cdot \begin{bmatrix} \Delta V_{1a} \\ \Delta V_{1b} \\ \Delta V_{1c} \\ \dots \\ \Delta V_{ka} \\ \Delta V_{kb} \\ \Delta V_{kc} \\ \dots \\ \Delta V_{na} \\ \Delta V_{nb} \\ \Delta V_{nc} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ -I_{fa} \\ -I_{fb} \\ -I_{fc} \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Όπου k είναι ο ζυγός σφάλματος.

Εφαρμόζοντας παραγοντοποίηση LU στον πίνακα αγωγιμοτήτων Y , αποφεύγεται η ανάγκη δημιουργίας και αποθήκευσης του πίνακα συνθέτων αντιστάσεων Z . Σε ένα σύνθετο ηλεκτρικό σύστημα οι περισσότεροι ζυγοί συνδέονται με λιγότερους από δέκα άλλους ζυγούς, οδηγώντας έτσι σε έναν πολύ αραιό πίνακα αγωγιμοτήτων Y . Αντιθέτως, ο πίνακας συνθέτων αντιστάσεων Z είναι γενικά ένας πλήρης πίνακας. Εφαρμόζοντας παραγοντοποίηση LU, όχι μόνο εξοικονομείται αποθηκευτικός χώρος, αλλά και αυξάνεται η ταχύτητα υπολογισμού αποφεύγοντας την αντιστροφή του πίνακα.

Αφού ο Y είναι αραιός πίνακας, οι L και U είναι επίσης αραιοί πίνακες και το I_f είναι αραιό διάνυσμα, μπορούμε εύκολα να λύσουμε την εξίσωση με προς τα εμπρός και προς τα πίσω αντικατάσταση και να λάβουμε όλα τα ΔV .

Οι μετά σφάλματος τάσεις μπορούν συνεπώς να υπολογιστούν.

$$\begin{bmatrix} V_{pa}^{new} \\ V_{pb}^{new} \\ V_{pc}^{new} \end{bmatrix} = \begin{bmatrix} V_{pa} \\ V_{pb} \\ V_{pc} \end{bmatrix} + \begin{bmatrix} \Delta V_{pa} \\ \Delta V_{pb} \\ \Delta V_{pc} \end{bmatrix}$$

Όπου p είναι ο αριθμός του ζυγού.

Η πτώση τάσης κατά μήκος κάθε γραμμής του συστήματος μπορεί να υπολογιστεί και να αποθηκευτεί σε ένα διάνυσμα.

$$V_{line} = A \cdot V^{new}$$

Με A την πλήρη μήτρα πρόσπτωσης που ορίστηκε στην εξίσωση (4.6).

Στη συνέχεια δημιουργείται ο αρχικός πίνακας σύνθετων αντιστάσεων που δίνεται στην εξίσωση (4.5), αλλά σε αυτή την περίπτωση συμπεριλαμβάνονται και οι σύνθετες αντιστάσεις των μετασχηματιστών. Μετασχηματίζοντας αυτούς τους τρεις ακολουθιακούς πίνακες στο πεδίο των φάσεων, όπως υποδεικνύεται στις εξισώσεις

(4.8) και (4.9), λαμβάνουμε τον πίνακα αγωγιμοτήτων κλάδων Y_{line} στο πεδίο των φάσεων. Έτσι, υπολογίζουμε τα μετά σφάλματος ρεύματα των κλάδων:

$$I_{line} = Y_{line} \cdot V_{line}$$

και η ανάλυση βραχυκυκλωμάτων ολοκληρώνεται.

Συγκρινόμενη με άλλες μεθόδους, η χρησιμοποιούμενη μέθοδος δε χρειάζεται να μετατρέψει τον πίνακα αγωγιμοτήτων ανάλογα με τον τύπο του σφάλματος, αλλά απαιτούνται μόνο οι σύνθετες αντιστάσεις του ζυγού σφάλματος. Ο υπολογισμός της εξίσωσης (3.6), που δόθηκε στην υποενότητα 3.2.2, υποκαθίσταται τώρα από αυτόν της εξίσωσης (4.10).

Τα πλεονεκτήματα αυτής της μεθόδου είναι:

- Δημιουργεί τον τριφασικό πίνακα αγωγιμοτήτων και πραγματοποιεί τους υπολογισμούς σφάλματος στο πεδίο των φάσεων.
- Χρησιμοποιεί τον υπο-πίνακα σύνθετων αντιστάσεων του ζυγού σφάλματος για να υπολογίσει τα ρεύματα σφάλματος
- Αποφεύγει τη χρησιμοποίηση του πλήρους τριφασικού πίνακα σύνθετων αντιστάσεων.

4.3 Τοπολογική ανάλυση συστήματος

4.3.1 Βρόχοι συστήματος

Το πρώτο βήμα για την πραγματοποίηση της τοπολογικής ανάλυσης είναι η εύρεση όλων των βρόχων του συστήματος. Αναπτύσσεται αλγόριθμος που βρίσκει τους βρόχους του συστήματος χρησιμοποιώντας μια δεδομένη μέθοδο ανίχνευσης. Αυτή η μέθοδος ανίχνευσης είναι μια τροποποιημένη εκδοχή της Μεθόδου Αναζήτησης κατά Βάθος (Depth First Search- DFS) και της Διαδικασίας Επανιχνηλάτησης (Back Tracking- BT) που προτάθηκε από τον Tarjan [18]. Η εύρεση των Break Points προϋποθέτει την εύρεση όλων των βρόχων του συστήματος και προς τις δυο κατευθύνσεις. Η εύρεση των βρόχων πραγματοποιείται με τα ακόλουθα βασικά βήματα:

1. Ξεκινάμε από ένα ζυγό και προχωράμε μέσα στο δίκτυο μέχρι να βρεθεί ένας βρόχος.
2. Οπισθοχωρούμε στον προηγούμενο ζυγό και αναζητούμε άλλους πιθανούς βρόχους.
3. Εάν έχουμε οπισθοχωρήσει στο ζυγό από τον οποίο ξεκινήσαμε, διαγράφουμε το ζυγό αυτό από το σύστημα. Θεωρούμε τον επόμενο ζυγό και επιστρέφουμε στο βήμα 1 μέχρι να εξαλειφθούν όλοι οι ζυγοί.

Αποτέλεσμα αυτού του τρόπου υπολογισμού των βρόχων είναι μια συστηματική και αποτελεσματική διαδικασία εύρεσης κάθε βρόχου μόνο μια φορά.

4.3.2 Break Points

Από τη στιγμή που έχουν βρεθεί οι βρόχοι του συστήματος, το επόμενο βήμα είναι ο καθορισμός του συνόλου των Break Points, δηλαδή του ελάχιστου συνόλου ηλεκτρονόμεων που ανοίγουν όλους αυτούς τους βρόχους. Σκοπός είναι να βρεθεί το ελάχιστο σύνολο των ηλεκτρονόμεων που έχουν αυτή την ιδιότητα. Πρόκειται να χρησιμοποιηθούν δυο εναλλακτικές μέθοδοι για τον προσδιορισμό των Break Points.

4.3.2.1 Μέθοδος 1

Ακολουθούμε τη μέθοδο που προτάθηκε από τον Gajbhiye και τους συνεργάτες του [11] εισάγοντας μια μικρή τροποποίηση. Για την κατανόηση της μεθόδου θα πρέπει πρώτα να εισάγουμε την έννοια Βαθμός Συμμετοχής Ζυγού στους Βρόχους (Loop Node Degree- LND), η οποία αναφέρεται σε κάθε ζυγό του συστήματος.

LND: Ισούται με το πόσες φορές συμμετέχει ο ζυγός αυτός στους βρόχους του συστήματος.

Χρησιμοποιώντας την έννοια LND αναλύουμε τη διαδικασία εύρεσης των Break Points στα ακόλουθα βήματα:

1. Υπολογίζουμε τον LND κάθε ζυγού.
2. Επιλέγουμε το ζυγό με τον μεγαλύτερο LND.
3. Εάν περισσότεροι του ενός ζυγοί έχουν το ίδιο LND, επιλέγουμε το ζυγό με το μικρότερο αριθμό ηλεκτρονόμεων.
4. Προσθέτουμε τους ηλεκτρονόμους του ζυγού αυτού στο σύνολο των Break Points.
5. Απαλείφουμε τους βρόχους που περιλαμβάνουν κάποιον από αυτούς τους ηλεκτρονόμους αυτούς.
6. Εάν δεν έχουν απαλειφθεί όλοι οι βρόχοι επιστρέφουμε στο Βήμα 1.

Σε αυτό το σημείο θα πρέπει να αναφέρουμε ότι τα φανταστικά στοιχεία (ζυγοί, ηλεκτρονόμοι) δε λαμβάνονται υπόψη σε αυτό το στάδιο. Όποτε ένας φανταστικός ηλεκτρονόμος επιλέγεται ως Break Point, αγνοείται και ο αλγόριθμος συνεχίζει στον αμέσως επόμενο ζυγό με το μεγαλύτερο LND.

4.3.2.2 Μέθοδος 2

Η δεύτερη μέθοδος εύρεσης των Break Points δε χρησιμοποιεί τους βρόχους που υπολογίστηκαν παραπάνω, αλλά μόνο σε ποιους ηλεκτρονόμους προσφέρει κάθε ηλεκτρονόμος δευτερεύουσα προστασία. Η περίπλοκη διαδικασία εύρεσης των βρόχων του συστήματος δεν απαιτείται δηλαδή για τη εφαρμογή αυτής της μεθόδου.

Μια μέθοδος εύρεσης των Break Points ενός συστήματος προτάθηκε από τους Yue, Lu, Yu και Wang [12]. Η μέθοδος αυτή κάνει χρήση των μεταβλητών PRDD και PPDS για να δηλώσει τα ζεύγη πρωτεύουσας- δευτερεύουσας προστασίας. Η τιμή της μεταβλητής PRDD για κάθε ηλεκτρονόμο x δηλώνει το πλήθος των ηλεκτρονόμεων που σε καθενός τη ζώνη ο x προσφέρει δευτερεύουσα προστασία και η μεταβλητή PPDS είναι ένα σύνολο που αποτελείται από τους ηλεκτρονόμους αυτούς. Η

συνάρτηση $f(n)$ χρησιμοποιείται λοιπόν για να απεικονίσει τις παραπάνω μεταβλητές και έχει την ακόλουθη μορφή:

$$f(n) = P : \{n_1, n_2, \dots, n_p\}$$

Όπου n ο εν λόγω ηλεκτρονόμος, P η μεταβλητή PRDD και $\{n_1, n_2, \dots, n_p\}$ η μεταβλητή PPDS.

Αφού έχει δημιουργηθεί ο πίνακας M που περιλαμβάνει τις τιμές των παραπάνω μεταβλητών για καθέναν από τους ηλεκτρονόμους του συστήματος, προχωρούμε στον υπολογισμό των Break Points. Τα Break Points θα αποτελέσουν στοιχεία της διάταξης MBPS (Minimum Break Point Set). Η μέθοδος συνίσταται από τα εξής βήματα:

1. Ορισμός των i , k , RSM, N και r . Ο ακέραιος i χρησιμοποιείται για να δηλώσει τον αριθμό των εξωτερικών επαναλήψεων και έχει αρχικά τιμή 1. Ο ακέραιος k είναι ο αριθμός της γραμμής του πίνακα RSM και η αρχική του τιμή είναι επίσης 1. Το r , ο αριθμός του ηλεκτρονόμου, του οποίου το PRDD έχει τη μεγαλύτερη τιμή, προκύπτει από τη διάταξη N . Η διάταξη N περιέχει κάθε φορά τους ηλεκτρονόμους με το μεγαλύτερο PRDD. Οι ηλεκτρονόμοι αυτοί μπορεί να είναι περισσότεροι από ένας. Από αυτούς επιλέγεται τυχαία ένας, ο οποίος ονομάζεται r .
2. Συμπεριλαμβάνουμε τον ηλεκτρονόμο r στη διάταξη MBPS και έτσι αυτός αποτελεί το πρώτο Break Point. Επίσης, τοποθετούμε τον αριθμό r στην πρώτη γραμμή του πίνακα RSM και συνεπώς ο ηλεκτρονόμος r θα είναι αυτός που θα ρυθμιστεί πρώτος. Στην πρώτη σειρά του RSM τοποθετούνται τα Break Points του συστήματος.
3. Διαγράφουμε τον ηλεκτρονόμο r από τα σύνολα PPDS του πίνακα M . Συνέπεια αυτού είναι ότι πρέπει να μειωθούν κατά 1 το PRDD των ηλεκτρονόμων, από το PPDS των οποίων διαγράφηκε ο ηλεκτρονόμος r . Υπάρχουν δυο τρόποι για να διαγραφεί ο ηλεκτρονόμος r . Η πρώτη είναι να βρεθεί ο αριθμός r του ηλεκτρονόμου ύστερα από αναζήτηση του r στον πίνακα M και να διαγραφεί. Ο δεύτερος είναι να έχει δοθεί αρχικά η ταμπέλα “true” σε όλους τους ηλεκτρονόμους του πίνακα M και όταν πρέπει να διαγραφεί ένας ηλεκτρονόμος, να λαμβάνει την ταμπέλα “false”. Οι ηλεκτρονόμοι με ταμπέλα “false” δε θα λαμβάνονται στο εξής υπόψη.
4. Ελέγχουμε αν υπάρχουν μηδενικά σύνολα στον πίνακα M , δηλαδή εάν υπάρχουν ηλεκτρονόμοι των οποίων το σύνολο PRDD είναι μηδενικό. Τοποθετούμε τους ηλεκτρονόμους αυτούς στην $k+1$ γραμμή του πίνακα RSG. Εν τω μεταξύ, διαγράφουμε τους ηλεκτρονόμους αυτούς από τον πίνακα M και μειώνουμε κατά 1 το PRDD των ηλεκτρονόμων, από το PPDS των οποίων διαγράφηκε ο ηλεκτρονόμος r . Στη συνέχεια ελέγχουμε εάν προέκυψαν μετά την αφαίρεση ηλεκτρονόμοι με μηδενικό PRDD. Εάν συμβαίνει αυτό επιστρέφουμε στο βήμα 4, αλλιώς επιστρέφουμε στο βήμα 1 και αναζητούμε του ηλεκτρονόμου με το μεγαλύτερο PRDD, βρίσκοντας έτσι το επόμενο Break Point.
5. Εάν όλα τα PRDD στον πίνακα M είναι μηδενικά, η επαναληπτική διαδικασία σταματάει και οι πίνακες MBPS και RSG έχουν βρεθεί. Τα Break Points, που καταγράφονται στον πίνακα MBPS, αποτελούν τους ηλεκτρονόμους που ρυθμίζονται πρώτοι. Οι ηλεκτρονόμοι κάθε σειράς του πίνακα RSG πρέπει να συνεργάζονται με τους ηλεκτρονόμους ανώτερων σειρών. Έτσι η ρύθμιση των

υπολοίπων ηλεκτρονόμων ακολουθεί τη σειρά τους στον πίνακα RSG με φορά από πάνω προς τα κάτω.

Τα Break Points που προκύπτουν από την εφαρμογή της μεθόδου αυτής είναι αποθηκευμένα στην πρώτη γραμμή του πίνακα RSM.

Πλεονεκτήματα

Ένα πλεονέκτημα της μεθόδου αυτής είναι η απλότητά της. Στα πλεονεκτήματα θα πρέπει να συμπεριληφθούν επίσης οι μικρές απαιτήσεις σε μνήμη και η γρήγορη σύγκλισή της. Μάλιστα η σύγκλιση της μεθόδου αυτής δεν εξαρτάται από τον αριθμό των ζυγών και των κλάδων του υπό μελέτη συστήματος.

4.3.3 RSV

Αφού έχει βρεθεί το ελάχιστο σύνολο Break Points, το επόμενο βήμα είναι να τοποθετηθούν όλοι οι ηλεκτρονόμοι σε συγκεκριμένη αλληλουχία, η οποία υποδεικνύει τη σειρά ρύθμισής τους. Η δεδομένη αυτή διάταξη των ηλεκτρονόμων ονομάζεται Διάνυσμα Σχετικής Ακολουθίας (Relative Sequence Vector- RSV). Η διαδικασία δημιουργίας του εξαρτάται από τη μέθοδο εύρεσης των Break Points. Έτσι έχουμε και στην περίπτωση αυτή τις ακόλουθες δυο μεθόδους:

4.3.3.1 Μέθοδος 1

Όταν τα Break Points του συστήματος έχουν υπολογιστεί με την πρώτη μέθοδο το διάνυσμα RSV δημιουργείται με την ακόλουθη λογική [18]:

1. Πρώτα τοποθετούνται τα Break Points και αυτό σημαίνει ότι αυτοί είναι οι πρώτοι ηλεκτρονόμοι που ρυθμίζονται.
2. Στη συνέχεια τοποθετούνται οι υπόλοιποι ηλεκτρονόμοι, έτσι ώστε για κάθε ηλεκτρονόμο που πρόκειται να τοποθετηθεί, όλοι οι πρωτεύοντες ηλεκτρονόμοι του να έχουν ήδη τοποθετηθεί στο RSV. Αυτό εξασφαλίζει ότι ο εν λόγω ηλεκτρονόμος θα ρυθμιστεί μετά από τους πρωτεύοντες ηλεκτρονόμους του και συνεπώς η ρύθμιση θα γίνει με τρόπο ώστε να συνεργάζεται με τους πρωτεύοντές του.

4.3.3.2 Μέθοδος 2

Στην περίπτωση υπολογισμού των Break Points με τη δεύτερη μέθοδο, έχει δημιουργηθεί ταυτόχρονα και ο πίνακας RSM. Ο πίνακας αυτός περιέχει την επιθυμητή διαδοχή των ηλεκτρονόμων, δηλαδή για κάθε ηλεκτρονόμο όλοι οι πρωτεύοντες ηλεκτρονόμοι του προηγούνται στον RSM. Για τη δημιουργία του Διανύσματος Σχετικής Ακολουθίας RSV, μένει συνεπώς απλά να διαβαστούν διαδοχικά οι γραμμές του RSM και τα στοιχεία να αποθηκευτούν στο διάνυσμα RSV.

Τοποθετώντας τους ηλεκτρονόμους σε αυτή τη «βέλτιστη» ακολουθία, είτε με την πρώτη είτε με τη δεύτερη μέθοδο, εξασφαλίζουμε επίσης ότι κάθε ηλεκτρονόμος επισκέπτεται μόνο μια φορά κατά τη διάρκεια των επαναλήψεων. Αυτό εξηγείται ως εξής: Έστω ότι πρόκειται να τοποθετηθεί στο RSV ένας ηλεκτρονόμος x . Η

τοποθέτηση του x προϋποθέτει, όπως αναφέρθηκε προηγουμένως, ότι όλοι οι πρωτεύοντες ηλεκτρονόμοι του έχουν ήδη τοποθετηθεί. Σημαίνει όμως επίσης ότι κανένας από τους ηλεκτρονόμους που προσφέρουν δευτερεύουσα προστασία στον x είτε δεν έχει τοποθετηθεί στον RSV. Για να μπορούσε να τοποθετηθεί θα έπρεπε να υπήρχαν ήδη στον RSV όλοι οι πρωτεύοντές του, άρα και ο x , πράγμα που δε συμβαίνει. Αφού λοιπόν δεν υπάρχουν οι ηλεκτρονόμοι αυτοί στον RSV, δεν έχουν ρυθμιστεί, άρα δεν οδηγούμαστε μετά τη ρύθμιση του x στην ανάγκη εκ νέου ρύθμισης των ηλεκτρονόμων αυτών ώστε να προσφέρουν σωστή δευτερεύουσα προστασία στον x . Υπάρχει όμως και το ενδεχόμενο οι δευτερεύοντες ηλεκτρονόμοι του x ή κάποιοι από αυτούς να βρίσκονται ήδη στον RSV, αλλά στην περίπτωση αυτή πρόκειται αναγκαστικά για break points. Ακόμη και σε αυτή την περίπτωση δεν υπάρχει εκ νέου ρύθμιση των ηλεκτρονόμων αυτών μετά τη ρύθμιση του x , καθώς πρόκειται για break points, οι ρυθμίσεις των οποίων παραμένουν αμετάβλητες. Συμπεραίνουμε λοιπόν ότι σε κάθε περίπτωση δεν απαιτείται εκ νέου ρύθμιση κάποιων ηλεκτρονόμων, δηλαδή κάθε ηλεκτρονόμος επισκέπτεται (ρυθμίζεται) μόνο μια φορά.

Η απουσία τέτοιας αλληλουχίας μπορεί να οδηγήσει στο να επισκέπτεται κάθε ηλεκτρονόμος περισσότερες από μια φορές κατά τη διάρκεια κάθε επανάληψης. Έχει αποδειχθεί ότι εάν οι ηλεκτρονόμοι υπερέντασης ρυθμίζονται χρησιμοποιώντας το διάνυσμα RSV, η διαδικασία ρύθμισης των ηλεκτρονόμων συγκλίνει πολύ γρήγορα.

4.3.4 SSP

Αφού η συνεργασία των ηλεκτρονόμων βασίζεται στην έννοια των ζευγαριών P/B, το επόμενο βήμα είναι να καθορίσουμε το Σύνολο των Διαδοχικών Ζευγών (Set of Sequential Pairs- SSP), που είναι η διατεταγμένη ακολουθία των ζευγαριών P/B του συστήματος. Τα ζευγάρια τοποθετούνται σε σειρά με τέτοιο τρόπο ώστε οι ηλεκτρονόμοι δευτερεύουσας προστασίας να ακολουθούν τη σειρά των ηλεκτρονόμων στο διάνυσμα RSV. Για κάθε ηλεκτρονόμο δηλαδή του RSV τοποθετούμε στο SSP όλα τα ζεύγη που δημιουργούνται με back-up τον ηλεκτρονόμο αυτό και συνεχίζουμε τη διαδικασία με τον επόμενο ηλεκτρονόμο του RSV.

Με την εύρεση του συνόλου SSP ολοκληρώνεται η τοπολογική ανάλυση του υπό μελέτη συστήματος. Τα αποτελέσματά της πρόκειται να χρησιμοποιηθούν στη συνέχεια για τη ρύθμιση των ηλεκτρονόμων (με την πρώτη μέθοδο, αυτή της τοπολογικής ανάλυσης).

4.4 Επιλογή - ρύθμιση εξοπλισμού προστασίας

Παρουσιάζουμε στο σημείο αυτό τη μεθοδολογία που ακολουθήθηκε στη διπλωματική αυτή εργασία για τη επιλογή ή ρύθμιση του εξοπλισμού προστασίας. Όπως κάθε μέθοδος ρύθμισης των στοιχείων προστασίας, έτσι και αυτή προσαρμόζεται στις ενδεχόμενες αλλαγές του συστήματος που συνεπάγονται αλλαγές στις ρυθμίσεις των ηλεκτρονόμων, όπως είναι οι αλλαγές στα φορτία, στις γεννήτριες και στην τοπολογία του συστήματος.

4.4.1 Επιλογή των μετασχηματιστών εντάσεως (CT)

Επιλέγουμε το λόγο μετασχηματισμού καθενός από τους μετασχηματιστές εντάσεως με βάσει τα κριτήρια που αναφέρθηκαν στη υποενότητα 2.6. Επιλέγεται τιμή 5A για το δευτερεύον του μετασχηματιστή. Η τιμή του πρωτεύοντος καθορίζεται από το ρεύμα κανονικής λειτουργίας, αλλά δεν πρέπει να παρατηρούνται φαινόμενα κορεσμού κατά την εκδήλωση βραχυκυκλωμάτων. Συγκεκριμένα η διαδικασία που ακολουθείται έχει ως εξής:

1. Θεωρείται το ρεύμα κανονικής λειτουργίας (σε A) που διαρρέει τη γραμμή στο σημείο που βρίσκεται ο ηλεκτρονόμος.
2. Θεωρείται το ρεύμα σφάλματος (σε A) που διαρρέει τη γραμμή στο σημείο που βρίσκεται ο ηλεκτρονόμος για σφάλμα αμέσως μετά τον ηλεκτρονόμο.
3. Επιλέγεται ως προσωρινή τιμή του πρωτεύοντος και η τιμή της αμέσως μεγαλύτερης εκατοντάδας.
4. Η τιμή που προέκυψε στο Βήμα 3 αυξάνεται ανά 100, έως ότου το αποτέλεσμα του πολλαπλασιασμού της επί 20 να είναι μεγαλύτερο από την τιμή του Βήματος 2.
5. Ο λόγος μετασχηματισμού ισούται με την τιμή που τελικά προκύπτει στο Βήμα 4 διαιρεμένη με 5.

4.4.2 Ρύθμιση των H/N υπερέντασης

4.4.2.1 Ρεύμα ρύθμισης ($I_{pick-up}$)

Το ρεύμα ρύθμισης επιλέγεται με βάση το σύστημα και το σχεδιασμό των ηλεκτρονόμων και η επιλογή είναι συνήθως εμπειρική. Στη διπλωματική αυτή εργασία το ρεύμα ρύθμισης κάθε ηλεκτρονόμου λαμβάνεται ίσο με το ρεύμα κανονικής λειτουργίας που διαρρέει τον ηλεκτρονόμο πολλαπλασιασμένο με 1,5. Το ρεύμα κανονικής λειτουργίας είναι αυτό που προκύπτει από την ανάλυση ροών φορτίου. Επιτρεπόμενη περιοχή τιμών για το ρεύμα ρύθμισης είναι το διάστημα [1,12]. Επειδή όμως το ρεύμα ρύθμισης λαμβάνει, όπως έχει ήδη αναφερθεί, διακριτές τιμές, οι επιτρεπόμενες τιμές θεωρούνται οι ακέραιες και οι δεκαδικές με δεκαδικό μέρος ίσο με 0,5.

Για τον καθορισμό του ρεύματος ρύθμισης κάθε ηλεκτρονόμου ακολουθούνται τα παρακάτω βήματα:

1. Κατ' αρχήν θεωρούμε το ρεύμα κανονικής λειτουργίας (σε A) που διαρρέει τη γραμμή στο σημείο που βρίσκεται ο ηλεκτρονόμος.
2. Το ρεύμα αυτό διαιρείται με το λόγο μετασχηματισμού του μετασχηματιστή έντασης του ηλεκτρονόμου για να βρεθεί το ρεύμα που διαρρέει τον ηλεκτρονόμο.
3. Η τιμή αυτή πολλαπλασιάζεται επί 1,5 και προκύπτει το ρεύμα ρύθμισης.

4.4.2.2 Χρονική σταθερά (TDS)

Ο προσδιορισμός της σταθεράς TDS κάθε ηλεκτρονόμου αποτελεί και το βασικότερο υπολογισμό αυτής της εργασίας. Για την πραγματοποίηση του υπολογισμού θα χρησιμοποιηθούν δυο από τις τεχνικές που παρουσιάστηκαν στην παράγραφο 3.4, η μέθοδος της τοπολογικής ανάλυσης και η μέθοδος του γραμμικού προγραμματισμού. Οι τεχνικές αυτές είναι ανεξάρτητες και υπολογίζουν και οι δυο τα TDS των ηλεκτρονόμων υπερέντασης. Γίνεται χρήση και των δυο για να μπορούν στη συνέχεια να συγκριθούν τα αποτελέσματά τους. Στη συνέχεια παρουσιάζεται η διαδικασία που ακολουθείται σε καθεμία από αυτές τις μεθόδους.

4.4.2.2.1 Μέθοδος τοπολογικής ανάλυσης

Η πρώτη μέθοδος που θα εφαρμόσουμε χρησιμοποιεί τα αποτελέσματα της τοπολογικής ανάλυσης του συστήματος, δηλαδή το σύνολο των Break Points, το διάνυσμα RSV και τον πίνακα SSP. Τα δεδομένα αυτά υποδεικνύουν τη βέλτιστη σειρά ρύθμισης των ηλεκτρονόμων. Μοναδική απαίτηση κατά τον υπολογισμό των TDS είναι να υπάρχει σωστή συνεργασία σε κάθε ζεύγος ηλεκτρονόμων πρωτεύουσας – δευτερεύουσας προστασίας. Θα πρέπει δηλαδή ο χρόνος λειτουργίας του back up ηλεκτρονόμου να είναι τουλάχιστον ίσος με το άθροισμα του χρόνου λειτουργίας του primary ηλεκτρονόμου και του χρονικού περιθωρίου που επιβάλλει η συνεργασία, δηλαδή

$$T_{ji} \geq T_{ii} + CTI \quad (4.11)$$

Όπου

- T_{ii} ο χρόνος λειτουργίας του primary ηλεκτρονόμου i για close-in σφάλμα i,
- T_{ji} ο χρόνος λειτουργίας του back up ηλεκτρονόμου j για το ίδιο σφάλμα i,
- CTI το χρονικό περιθώριο συνεργασίας.

Η γενική ιδέα της μεθόδου αναφέρεται στο [18]. Η διαδικασία η οποία ακολουθείται έχει ως ακολούθως:

1. Δίνονται αρχικές ρυθμίσεις ίσες με την ελάχιστη επιτρεπόμενη τιμή του TDS σε όλους τους ηλεκτρονόμους που αποτελούν το σύνολο των Break Points.
2. Ξεκινώντας από το ζεύγος ηλεκτρονόμων που είναι αμέσως μετά τα Break Points στον πίνακα SSP, διατρέχουμε τον πίνακα και ρυθμίζουμε και τους υπόλοιπους ηλεκτρονόμους. Θεωρούμε ένα ζεύγος κάθε φορά και ο ηλεκτρονόμος δευτερεύουσας προστασίας (back up) ρυθμίζεται έτσι ώστε να συνεργάζεται σωστά με τον ηλεκτρονόμο πρωτεύουσας προστασίας (primary), ο οποίος έχει ήδη ρυθμιστεί σε προηγούμενο στάδιο. Γίνεται δηλαδή χρήση της ισότητας της σχέσης (4.11).

3. Μετά τη ρύθμιση του τελευταίου ηλεκτρονόμου, μετακινούμαστε πάλι στην αρχή του SSP και ελέγχουμε αν οι back up ηλεκτρονόμοι συνεργάζονται σωστά με τους primary ηλεκτρονόμους τους. Εάν δε συμβαίνει αυτό, οι ρυθμίσεις των ηλεκτρονόμων πρέπει να αλλάξουν. Συνεπώς διατρέχουμε τον πίνακα για δεύτερη φορά, ορίζοντας τα νέα TDS.
4. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να επιτευχθεί η επιθυμητή συνεργασία σε όλα τα ζεύγη primary – back up προστασίας.

Αποδεικνύεται ότι ο πίνακας SSP παρέχει γρήγορη σύγκλιση της επαναληπτικής αυτής διαδικασίας.

4.4.2.2 Μέθοδος γραμμικού προγραμματισμού

Στη διπλωματική αυτή ακολουθείται η μέθοδος γραμμικού προγραμματισμού που χρησιμοποιήθηκε από τους Talaat, Nosseir και τους συνεργάτες τους [6]. Πρόκειται για μια τροποποιημένη διαμόρφωση του προβλήματος με σκοπό την επιτάχυνση της διαδικασίας. Εφαρμόζεται δηλαδή μια μέθοδος γραμμικού προγραμματισμού με μείωση στη διάσταση του προβλήματος. Σε πρώτη φάση υπολογίζεται μια εφικτή λύση, ενώ η δεύτερη φάση περιλαμβάνει μια επαναληπτική ακολουθία εφικτών λύσεων, οι οποίες συγκλίνουν στη βέλτιστη λύση. Η αντικειμενική συνάρτηση δεν περιλαμβάνει το άθροισμα των χρόνων λειτουργίας όλων των ηλεκτρονόμων που παρέχουν πρωτεύουσα προστασία (primary relays) για μέγιστο ρεύμα σφάλματος και για σφάλμα αμέσως μετά τον ηλεκτρονόμο (close-in fault), όπως θεωρείται συνήθως. Είναι απλά το άθροισμα των TDS όλων των ηλεκτρονόμων, όπως θα αποδειχθεί στη συνέχεια. Οι περιορισμοί αναφέρονται και αυτοί σε μέγιστο ρεύμα σφάλματος για close-in σφάλματα.

Το πρόβλημα γραμμικού προγραμματισμού που θα εφαρμόσουμε περιγράφεται από τις ακόλουθες εξισώσεις:

$$\begin{aligned} \min \left(J = \sum_{i=1}^n TDS_i \right) \\ T_{ji} \geq T_{ii} + CTI \\ TDS_{\min} \leq TDS_i \leq TDS_{\max} \end{aligned} \quad (4.12)$$

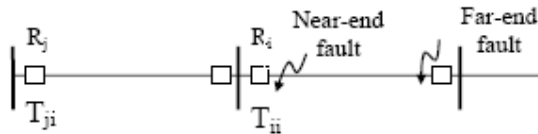
Σκοπός είναι ο προσδιορισμός των μεταβλητών TDS_i ελαχιστοποιώντας την αντικειμενική συνάρτηση J , η οποία αποτελεί το άθροισμα των TDS όλων των ηλεκτρονόμων του συστήματος. Παράλληλα όμως θα πρέπει οι μεταβλητές αυτές να ικανοποιούν τους περιορισμούς που επιβάλλονται από το γεγονός ότι για κάθε ζεύγος ηλεκτρονόμων πρωτεύουσας – δευτερεύουσας προστασίας θα πρέπει ο δευτερεύων ηλεκτρονόμος να καθυστερεί τη λειτουργία του κατά 0,2 sec σε σχέση με τον πρωτεύον. Αν το σύστημα λοιπόν έχει m ζεύγη ηλεκτρονόμων, θα έχουμε m ανισώσεις της μορφής $T_{ji} \geq T_{ii} + CTI$. Τέλος, οι ζητούμενες μεταβλητές μπορούν να λάβουν τιμές από ένα συγκεκριμένο εύρος τιμών, γι' αυτό στους περιορισμούς περιλαμβάνονται και οι περιορισμοί της μορφής $TDS_{\min} \leq TDS_i \leq TDS_{\max}$.

Το πρόβλημα γραμμικού προγραμματισμού που θα χρησιμοποιηθεί διαφέρει στη διαμόρφωσή του από άλλες μεθόδους γραμμικού προγραμματισμού που χρησιμοποιούνται για τον ίδιο σκοπό, δηλαδή για τον υπολογισμό των TDS. Η διαφοροποίησή του έγκειται στα ακόλουθα χαρακτηριστικά:

- Η αντικειμενική συνάρτηση δεν αποτελεί το άθροισμα των χρόνων λειτουργίας των ηλεκτρονόμων για close-in σφάλμα σε καθέναν από αυτούς, όπως θεωρείται συνήθως. Στην παρούσα μεθοδολογία χρησιμοποιείται το άθροισμα των TDS των ηλεκτρονόμων, δηλαδή παραλείπονται οι συντελεστές

$$a_i \text{ της εξίσωσης } J = \sum_{i=1}^n a_i \cdot TDS_i .$$

- Το σφάλμα το οποίο θεωρούμε για τον υπολογισμό του χρόνου λειτουργίας κάθε ηλεκτρονόμου είναι σφάλμα αμέσως μετά από τον ηλεκτρονόμο (close-in fault). Σε αυτά τα σφάλματα αναφέρονται οι προσθετοί της αντικειμενικής συνάρτησης και οι περιορισμοί, οι σχετικοί με τη συνεργασία των ηλεκτρονόμων, στους οποίους υπόκεινται τα TDS. Δε θεωρούμε επιπλέον σφάλματα στο τέλος της ζώνης προστασίας κάθε ηλεκτρονόμου (far-end fault), πράγμα που γίνεται σε άλλες διατυπώσεις του προβλήματος. Τα σφάλματα αυτά υπενθυμίζονται στο παρακάτω σχήμα.
- Η μεταβατική κατάσταση δε λαμβάνεται υπόψη κατά τους υπολογισμούς για την εύρεση των TDS. Η κατάσταση αυτή, όπως και οι παράμετροι που αναφέρθηκαν αμέσως πριν, δε επηρεάζει την εύρεση της λύσης, όπως θα εξηγηθεί παρακάτω.



Σχήμα 9 : Τύποι σφαλμάτων

Αιτιολόγηση της διαμόρφωσης του προβλήματος

a. Η μη επιρροή των συντελεστών a_i

Σύμφωνα με τη μαθηματική διατύπωση ου προβλήματος γραμμικού προγραμματισμού, μπορεί ναδειχθεί ότι η αντικειμενική συνάρτηση έχει αρνητική κλίση, ενώ οι περιορισμοί $(a_i X - b_i Y \geq c)$ έχουν θετική κλίση. Η βέλτιστη λύση δε θα αλλάξει εάν οι συντελεστές των προσθετών της αντικειμενικής συνάρτησης λάβουν διάφορες τιμές στο διάστημα $(0, \infty)$. Η βέλτιστη λύση θα αλλάξει μόνο στην περίπτωση που οι συντελεστές αυτοί λάβουν αρνητικές τιμές. Αυτή η περίπτωση όμως είναι αδύνατη, καθώς οι συντελεστές είναι θετικοί πραγματικοί αριθμοί. Αν αποδεχόμασταν και αρνητικές τιμές των a_i , κάποιοι από τους χρόνους λειτουργίας των ηλεκτρονόμων θα μεγιστοποιούνταν αντί να ελαχιστοποιούνταν. Μια ανάλυση ευαισθησίας σε οποιοδήποτε παράδειγμα γραμμικού προγραμματισμού, στο οποίο η αντικειμενική συνάρτηση και οι περιορισμοί είναι ίδιου τύπου με αυτούς του δικού μας προβλήματος, αποδεικνύει τη μη επίδραση των συντελεστών a_i . Αφού λοιπόν η

τιμή των συντελεστών δεν επηρεάζει τη λύση, οι συντελεστές a_i λαμβάνουν μοναδιαία τιμή.

b. Η μη επιρροή του σφάλματος στο τέλος της ζώνης προστασίας (far-end fault)

Η θεώρηση και των σφαλμάτων στο τέλος της γραμμής, στην αρχή της οποίας βρίσκεται ο ηλεκτρονόμος, αυξάνει τη διάσταση του προβλήματος χωρίς παρ' όλα αυτά να οδηγεί σε διαφορετική βέλτιστη λύση. Οι μελετητές που συμπεριέλαβαν τα σφάλματα αυτά στη μελέτη τους πίστευαν ότι οι χρόνοι λειτουργίας των ηλεκτρονόμων θα ελαχιστοποιούνταν.

Στην πραγματικότητα η θεώρηση αυτή δεν έχει κανένα πλεονέκτημα, καθώς οποιαδήποτε αλλαγή στους συντελεστές της αντικειμενικής συνάρτησης δεν έχει καμία επίδραση στη βέλτιστη λύση, όπως εξηγήθηκε αμέσως πριν. Άλλωστε η ελαχιστοποίηση του χρόνου λειτουργίας για close-in σφάλμα οδηγεί αυτόματα στην ελαχιστοποίηση των χρόνων λειτουργίας για σφάλμα σε οποιοδήποτε σημείο τη γραμμής, όπως στο μέσο ή το άλλο άκρο της γραμμής (far-end fault), χωρίς όμως η σχέση αυτή να ισχύει και αντίστροφα. Επιπλέον, κάθε ηλεκτρονόμος θα έχει δυο περιορισμούς, αλλά ο ένας από αυτούς θα είναι περιττός, καθώς θα ακυρώνεται από τον άλλο που θα έχει μικρότερη κλίση.

Συνεπώς, η θεώρηση και σφαλμάτων στο τέλος της ζώνης προστασίας είναι περιττή και γι' αυτό παραλείπεται στη συγκεκριμένη διαμόρφωση του προβλήματος. Συνέπεια αυτού είναι να μειωθεί η διάσταση του προβλήματος στο μισό και να επιτυγχάνεται γρήγορη σύγκλιση (αφού ελαχιστοποιείται ο αριθμός των επαναλήψεων), άρα και επιτάχυνση της διαδικασίας ρύθμισης των ηλεκτρονόμων.

c. Η μη επιρροή της μεταβατικής κατάστασης

Μεταβατική είναι η κατάσταση που επικρατεί κατά τη διάρκεια εκκαθάρισης του βραχυκυκλώματος. Κάποιοι μελετητές έχουν περιλάβει τις τοπολογικές αλλαγές, που συμβαίνουν στο σύστημα κατά τη μεταβατική κατάσταση, στη διαμόρφωση του προβλήματος, όπως αναφέρθηκε στην παράγραφο 3.4.3.1. Έτσι, η διάσταση του προβλήματος αυξάνεται δυο ή τέσσερις φορές, καθώς για κάθε ηλεκτρονόμο θεωρούνται πλέον τέσσερις περιορισμοί, και το πρόβλημα γίνεται πιο περίπλοκο από ότι προηγούμενως.

Στην πραγματικότητα, η θεώρηση της μεταβατικής κατάστασης λειτουργίας είναι αποτελεσματική μόνο σε μια σπάνια περίπτωση: όταν η μεταβολή του ρεύματος που διαρρέει τον ηλεκτρονόμο δευτερεύουσας προστασίας είναι μεγαλύτερη από τη μεταβολή του ρεύματος που διαρρέει τον ηλεκτρονόμο πρωτεύουσας προστασίας. Αυτό συμβαίνει μόνο στην περίπτωση δυο παράλληλων γραμμών που τροφοδοτούνται από πηγές και στα δυο άκρα τους. Στις άλλες περιπτώσεις δεν υπάρχει περίπτωση εάν θεωρήσουμε τη μεταβατική κατάσταση λειτουργίας. Αυτό μπορεί να αποδειχθεί με προσομοίωση, γραφικά ή μαθηματικά. Είναι λοιπόν προτιμότερο να μην τη συμπεριλάβουμε στη διαμόρφωση του προβλήματος, με σκοπό να μειώσουμε τη διάσταση του προβλήματος και τον αριθμό των επαναλήψεων και συνεπώς το χρόνο της διαδικασίας.

Πλεονέκτημα ακολουθούμενης τεχνικής

Από τα παραπάνω είναι εμφανές το πλεονέκτημα που προσφέρει αυτή η διαμόρφωση του προβλήματος γραμμικού προγραμματισμού. Μειώνεται η διάσταση του προβλήματος στο ένα τέταρτο, ελαττώνοντας έτσι τον αριθμό των επαναλήψεων που απαιτούνται για την εύρεση λύσης. Συνεπώς η διαδικασία ρύθμισης των ηλεκτρονόμων γίνεται γρηγορότερη. Το χαρακτηριστικό αυτό καθιστά τη μέθοδο αυτή κατάλληλη και για διαδικτυακές εφαρμογές.

5

Υλοποίηση των χρησιμοποιούμενων τεχνικών

Στο προηγούμενο κεφάλαιο παρουσιάστηκαν οι τεχνικές που θα χρησιμοποιηθούν σε καθένα από τα τμήματα στα οποία μπορεί να χωριστεί αυτή η διπλωματική εργασία. Έγινε μια αναφορά στη γενική ιδέα, τη διαδικασία και το θεωρητικό υπόβαθρο, όπου αυτό υφίσταται. Στο κεφάλαιο αυτό θα ασχοληθούμε με την υλοποίηση των μεθόδων αυτών, δηλαδή τη δημιουργία προγράμματος που εκτελεί τις διαδικασίες αυτές. Θα παρουσιάζονται με διαγράμματα ροής τα βήματα των προγραμμάτων και στη συνέχεια θα εξηγούνται συνοπτικά ή αναλυτικότερα τα βήματα αυτά, αναφέροντας και μεταβλητές του προγράμματος. Δεν πρόκειται να δοθούν κώδικες στο σημείο αυτό, αλλά θα παρατεθούν συγκεντρωτικά στο Παράρτημα.

Η υλοποίηση των αλγορίθμων έγινε στο πρόγραμμα Matlab. Πρόκειται για μια γλώσσα προγραμματισμού υψηλού επιπέδου και ένα διαδραστικό περιβάλλον που επιτρέπει την πραγματοποίηση υπολογιστικών εργασιών πιο γρήγορα από τις παραδοσιακές γλώσσες, όπως οι C, C++ και Fortran.

Τα σημαντικά χαρακτηριστικά που διαθέτει το Matlab είναι:

- Γλώσσα υψηλού επιπέδου για τεχνικούς υπολογισμούς
- Περιβάλλον για διαχείριση κώδικα, αρχείων και δεδομένων
- Διαδραστικά εργαλεία για επαναληπτική αναζήτηση, σχεδίαση και επίλυση προβλημάτων
- Μαθηματικές συναρτήσεις για γραμμική άλγεβρα, στατιστική, βελτιστοποίηση
- Ευκολία στη διαχείριση πινάκων
- Γραφική απεικόνιση δεδομένων
- Συναρτήσεις για ολοκλήρωση των αλγορίθμων που βασίζονται στο Matlab με άλλες εφαρμογές και γλώσσες, όπως C, C++, Fortran, Java, COM, and Microsoft Excel

Η δομή του προγράμματος έχει ως εξής:

Υπάρχει ένα κυρίως πρόγραμμα, από το οποίο καλούνται τα ακόλουθα υποπρογράμματα:

- Μια συνάρτηση που πραγματοποιεί ανάλυση ροών φορτίου
- Μια συνάρτηση που εκτελεί ανάλυση βραχυκυκλωμάτων
- Ένα υποπρόγραμμα που πραγματοποιεί τοπολογική ανάλυση του συστήματος

Στο κυρίως πρόγραμμα πραγματοποιούνται όσοι επιπλέον υπολογισμοί χρειάζονται για να ολοκληρωθεί τελικά η διαδικασία με τη ρύθμιση των ηλεκτρονικών υπερέντασης.

Για την ανάλυση ροών φορτίου και την ανάλυση βραχυκυκλωμάτων χρησιμοποιούνται έτοιμα προγράμματα, τα οποία έχουν δημιουργηθεί από τους

Alvarado και Zhu αντίστοιχα. Βέβαια, στην ανάλυση βραχυκυκλωμάτων γίνεται παρέμβαση, ώστε τελικά να πραγματοποιείται ο υπολογισμός του ρεύματος σφάλματος τόσες φορές όσοι είναι οι ζυγοί του συστήματος. Το κυρίως πρόγραμμα, το οποίο καλεί όλες τις συναρτήσεις, πραγματοποιεί τη δημιουργία ενδιάμεσων πινάκων και τελικά τη ρύθμιση των ηλεκτρονόμων, καθώς και το υποπρόγραμμα τοπολογικής ανάλυσης, που υπολογίζει τους βρόχους, τα Break Points, το διάνυσμα RSV και τον πίνακα SSP, δημιουργήθηκαν σε αυτή τη διπλωματική εργασία.

5.1 Ανάλυση ροών φορτίου

5.1.1 Αρχείο εισόδου

Το πρώτο βήμα είναι η εισαγωγή των δεδομένων του συστήματος. Τα δεδομένα αυτά αποθηκεύονται και δε χρησιμοποιούνται μόνο κατά την ανάλυση ροών φορτίου, αλλά κάποια από αυτά χρησιμοποιούνται και μετέπειτα στο πρόγραμμα. Τα δεδομένα του υπό μελέτη συστήματος υπάρχουν αποθηκευμένα σε ένα αρχείο με κατάληξη “.cf”. Οργανώνονται σε τμήματα, καθένα από τα οποία έχει τίτλο και τερματίζεται με μια σημαία (συνήθως -9 ή -999). Τα δεδομένα είναι αποθηκευμένα σε συγκεκριμένες στήλες του αρχείου αυτού, καθώς η συνάρτηση που διαβάζει το αρχείο αυτό και εισάγει τα δεδομένα στο πρόγραμμα αναζητά σε κάθε συγκεκριμένη θέση το συγκεκριμένο στοιχείο που πρέπει να έχει αποθηκευτεί στη θέση αυτή. Ο τύπος δεδομένων μπορεί να είναι αλφαριθμητικός, ακέραιος ή κινητής υποδιαστολής.

Μερικά από τα δεδομένα του αρχείου εισόδου, η πρώτη γραμμή του οποίου περιλαμβάνει τον τίτλο του και τη βάση ισχύος του συστήματος, είναι:

Δεδομένα ζυγών

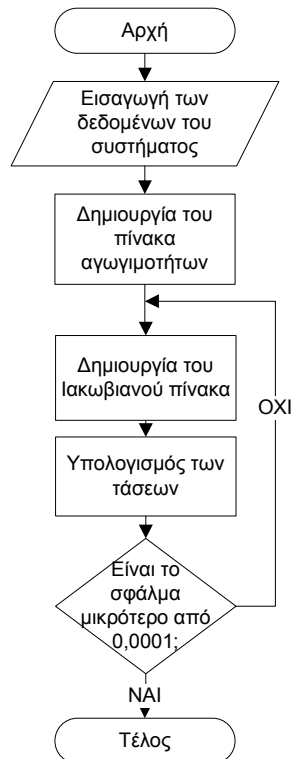
- Αριθμός του ζυγού
- Ο τύπος του ζυγού ($PV, PQ, swing$)
- Παραγόμενη ενεργός ισχύς $P(MW)$ και άεργος ισχύς $Q(MVAR)$
- Φορτίο σε MW και $MVAR$
- Βασική ισχύς (kV)
- Σύνθετη αγωγιμότητα παράλληλου στοιχείου σε ανά μονάδα (α.μ.)

Δεδομένα γραμμών

- Οι αριθμοί των ακραίων ζυγών της γραμμής αναφέροντας πρώτα το ζυγό στον οποίο καταλήγει η γραμμή
- Ο τύπος της γραμμής
- Η αντίσταση της γραμμής σε ανά μονάδα
- Η αντίδραση της γραμμής σε ανά μονάδα
- Ο λόγος μετασχηματισμού στην περίπτωση που πρόκειται για μετασχηματιστή
- Το όριο φόρτισης

5.1.2 Αλγόριθμος

Για τον προσδιορισμό των ρευμάτων κανονικής λειτουργίας χρησιμοποιείται το πρόγραμμα που δίνεται στο [5], το οποίο δημιουργήθηκε από τον C.DeMarco και αναπτύχθηκε από τον F.Alvarado. Η μέθοδος την οποία εφαρμόζει το πρόγραμμα αυτό είναι η μέθοδος Newton – Raphson, η θεωρητική ανάλυση της οποίας έγινε στην υποενότητα 4.1. Η υλοποίησή της γίνεται με κλήση επιμέρους συναρτήσεων και το τμήμα του κυρίως προγράμματος που πραγματοποιεί την ανάλυση ροών φορτίου μπορεί να περιγραφεί από το ακόλουθο διάγραμμα ροής:



Διάγραμμα 1: Ανάλυση ροών φορτίου

Τα δεδομένα διαβάζονται από το αρχείο εισόδου χρησιμοποιώντας τη συνάρτηση “readcf”. Το αποτέλεσμα αυτής της ανάγνωσης στοιχείων αποθηκεύεται σε κάποιους πίνακες: Τον πίνακα z που περιέχει τα στοιχεία των κλάδων, τον πίνακα L που περιέχει τα δεδομένα φορτίου και τον πίνακα A που περιέχει τα δεδομένα των γεννητριών. Επιπλέον αναγνωρίζονται οι ζυγοί αναφοράς, παραγωγής και φορτίου, καθώς και ο αριθμός των γραμμών ($nLines$) και ο αριθμός των ζυγών ($nNodes$) του συστήματος.

Στη συνέχεια δημιουργείται ο πίνακας αγωγιμοτήτων, ο οποίος έχει διαστάσεις ($nNodes \times nNodes$) με $nNodes$ τον αριθμό των ζυγών του υπό μελέτη συστήματος. Η δημιουργία του πίνακα αγωγιμοτήτων γίνεται καλώντας τη συνάρτηση “bldybus”.

Η βασική ρουτίνα στο παραπάνω πρόγραμμα είναι η “pfsolve. Το πρώτο βήμα είναι η αρχικοποίηση του σφάλματος, το οποίο λαμβάνει άπειρη τιμή. Στη συνέχεια ακολουθεί η επαναληπτική διαδικασία για τον υπολογισμό της λύσης ενημερώνοντας

σε κάθε επανάληψη τον Ιακωβιανό πίνακα. Βασική ιδέα της συνάρτησης είναι αυτή του υπολογισμού ενός μιγαδικού διανύσματος για να αποκτήσουμε μέσω αυτού το μιγαδικό Ιακωβιανό πίνακα. Από τον πίνακα αυτό προκύπτει ο πραγματικός Ιακωβιανός πίνακας. Η συνάρτηση αυτή επιστρέφει ως έξοδο τη μιγαδική τάση σε κάθε ζυγό και το σφάλμα σύγκλισης. Σε περίπτωση που το σφάλμα είναι μικρότερο από 0.001 επιτυγχάνεται σύγκλιση.

Μέχρι στιγμής έχουν υπολογιστεί το μέτρο και η γωνία της τάσης σε κάθε ζυγό του συστήματος. Οι μεταβλητές όμως που χρειαζόμαστε στη συνέχεια για τον καθορισμό των παραμέτρων των ηλεκτρονόμων είναι τα ρεύματα που ρέουν στο σύστημα. Με κλήση της συνάρτησης “flows” υπολογίζεται το ρεύμα σε κάθε γραμμή του συστήματος και αναφέρονται υπερφορτίσεις σε περίπτωση που αυτές υπάρχουν.

Με την ολοκλήρωση της εκτέλεσης του τμήματος του προγράμματος που πραγματοποιεί ανάλυση ροών φορτίου έχει δημιουργηθεί ένα αρχείο S. Το αρχείο αυτό περιέχει διάφορες μεταβλητές, από τις οποίες αυτές που θα χρησιμοποιήσουμε στη συνέχεια είναι:

S.Misc.BaseMVA	Η βάση ισχύος.
S.Bus.KV	Η βάση τάσης σε κάθε ζυγό του συστήματος (διάσταση $(nNodes \times 1)$).
S.Bus.Voltages	Η μιγαδική τάση (α.μ.) κάθε ζυγού κατά την κανονική λειτουργία (διάσταση $(nNodes \times 1)$).
S.Machine.BusRef	Οι αριθμοί των ζυγών όπου συνδέονται γεννήτριες.
S.Branch.From	Οι αριθμοί των ζυγών από τους οποίους ξεκινούν οι γραμμές (διάσταση $(nLines \times 1)$).
S.Branch.To	Οι αριθμοί των ζυγών στους οποίους καταλήγουν οι γραμμές (διάσταση $(nLines \times 1)$).
S.Branch.Z	Οι σύνθετη αντίσταση κάθε γραμμής (διάσταση $(nLines \times 1)$).
S.Flow	Το μιγαδικό ρεύμα (α.μ.) που ρέει σε κάθε γραμμή κατά την κανονική λειτουργία (διάσταση $(nLines \times 1)$).

Πίνακας 2: Μερικά από τα περιεχόμενα του αρχείου S

5.2 Ανάλυση βραχυκυκλωμάτων

Η διαδικασία ανάλυσης βραχυκυκλωμάτων, η οποία αναλύθηκε στο εδάφιο 4.2, μπορεί να περιγραφεί από τα εξής βήματα:

1. Δημιουργία του πίνακα Y για γραμμές μεταφοράς στο πεδίο των φάσεων.
2. Δημιουργία των πινάκων αγωγιμοτήτων για μετασχηματιστές στο πεδίο των φάσεων.
3. Συνδυασμός των πινάκων αυτών ώστε να δημιουργηθεί ο τελικός πίνακας Y στο πεδίο των φάσεων.

4. Υπολογισμός του σφάλματος χρησιμοποιώντας τον πίνακα αγωγιμοτήτων.
5. Υπολογισμός των μετά σφάλματος τάσεων των ζυγών.
6. Υπολογισμός των μετά σφάλματος ρευμάτων των γραμμών.

5.2.1 Αρχεία εισόδου

Αρχικά θα πρέπει να γίνει η εισαγωγή των δεδομένων του συστήματος στο οποίο επιθυμούμε να κάνουμε ανάλυση βραχυκυκλωμάτων. Οι τιμές των δεδομένων αυτών ανατίθενται στις αντίστοιχες τοπικές μεταβλητές της συνάρτησης που πραγματοποιεί την ανάλυση βραχυκυκλωμάτων και η οποία καλείται από το κυρίως πρόγραμμα. Τα δεδομένα είναι αποθηκευμένα σε τέσσερα αρχεία εισόδου:

5.2.1.1 *scparam.txt*

Αυτό το αρχείο περιέχει μόνο δυο παραμέτρους. Η πρώτη καθορίζει τον τρόπο με τον οποίο θα γίνει η ανάλυση βραχυκυκλωμάτων, αν δηλαδή θα συμπεριληφθούν τα ρεύματα κανονικής λειτουργίας ή όχι. Στην πρώτη περίπτωση οι αρχικές τάσεις των ζυγών έχουν τις τιμές που προκύπτουν από την ανάλυση ροών φορτίου, ενώ στη δεύτερη οι αρχικές τάσεις θεωρούνται ίσες με $1 pu$. Η δεύτερη μεταβλητή του αρχείου αυτού είναι η βασική ισχύς του συστήματος.

5.2.1.2 *busotp.txt*

Το αρχείο αυτό περιέχει τις προ σφάλματος τάσεις των ζυγών. Από εδώ διαβάζονται οι τάσεις αυτές στην περίπτωση που η μεταβλητή του προηγούμενου αρχείου καθορίζει τρόπο λύσης που να λαμβάνει υπόψη τα ρεύματα κανονικής λειτουργίας. Συγκεκριμένα, δίνεται για κάθε ζυγό ο αριθμός του ζυγού, το μέτρο της τάσης της φάσης a, η γωνία της τάσης αυτής, το μέτρο της τάσης της φάσης b, η γωνία της, το μέτρο της τάσης της φάσης c και η γωνία της τάσης αυτής.

Για κάθε ζυγό τα μέτρα των φασικών τάσεων είναι ίσα και οι γωνίες των φάσεων a, b, c είναι αντίστοιχα $0^\circ, -120^\circ, 120^\circ$.

5.2.1.3 *scinput.txt*

Το τρίτο αρχείο εισόδου παρέχει πληροφορίες για το σύστημα, όπως τα δεδομένα των γεννητριών, των παράλληλων στοιχείων, των φορτίων, των γραμμών και των αμοιβαίων αντιστάσεων.

- Τα δεδομένα των γεννητριών περιλαμβάνουν τη συνδεσμολογία κάθε γεννήτριας, το ζυγό στον οποίο συνδέεται, την αντίσταση και την αντίδρασή της για κάθε ακολουθία.
- Για τα παράλληλα στοιχεία δίνεται ο τρόπος συνδεσμολογίας, ο ζυγός στον οποίο συνδέεται καθένα από αυτά, την αντίσταση και την αντίδρασή του για κάθε ακολουθία.
- Σχετικά με τα φορτία του συστήματος δίνονται ο αντίστοιχος ζυγός, η ενεργός και η άεργος ισχύς του φορτίου.
- Τα δεδομένα των κλάδων περιλαμβάνουν τον αριθμό της γραμμής, τους ακραίους ζυγούς, τον τύπο της γραμμής και τη σύνθετη αντίστασή της για

κάθε ακολουθία. Στην περίπτωση που πρόκειται για μετασχηματιστή δίνονται επίσης ο τρόπος σύνδεσης και τα taps.

- Σχετικά με την αμοιβαία σύζευξη των γραμμών δίνονται οι αριθμοί των γραμμών που την εμφανίζουν, οι ακραίοι ζυγοί τους και η αμοιβαία σύνθετη αντίσταση. Στη διπλωματική αυτή εργασία θα αγνοηθεί η αμοιβαία σύζευξη μεταξύ των γραμμών και συνεπώς το τμήμα αυτό του αρχείου εισόδου δεν περιέχει δεδομένα.

5.2.1.4 flinput.txt

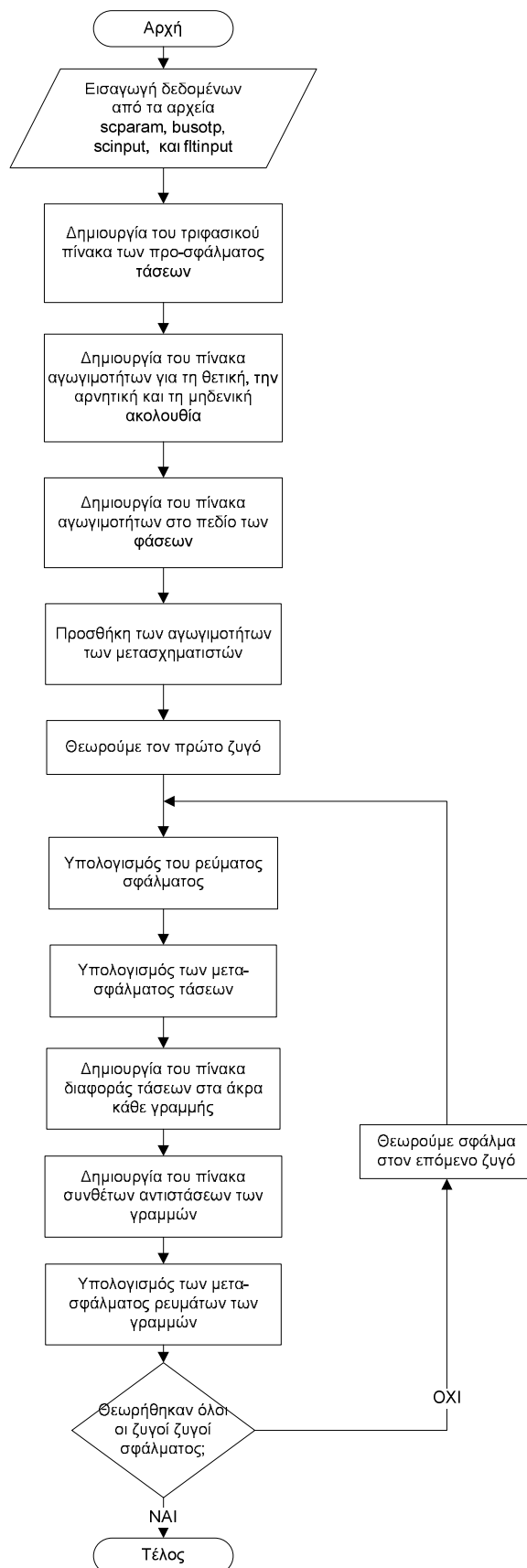
Το τελευταίο αρχείο εισόδου παρέχει πληροφορίες για το σφάλμα. Δίνεται ο ζυγός στον οποίο πραγματοποιείται, ο τύπος του και οι σύνθετες αντιστάσεις σφάλματος, z_a, z_b, z_c, z_g , που αναφέρθηκαν στην παράγραφο 4.2.2.

Για τη μελέτη προστασίας χρειάζονται τα ρεύματα σε κάθε ηλεκτρονόμο για σφάλμα σε καθέναν από τους ζυγούς. Τροποποιούμε λοιπόν τη συνάρτηση βραχυκυκλωμάτων του Jun Zhu, ώστε αυτή να υπολογίζει το ρεύμα σε κάθε κλάδο του συστήματος για σφάλμα σε έναν από τους ζυγούς κάθε φορά. Εισάγεται δηλαδή ένας επαναληπτικός βρόχος ώστε η ανάλυση βραχυκυκλωμάτων να γίνει τόσες φορές όσοι είναι οι ζυγοί του συστήματος.

Επίσης καταργούμε τα αρχεία εξόδου του προγράμματος του Jun Zhu, καθώς δε μας ενδιαφέρει η παρουσίαση των αποτελεσμάτων της ανάλυσης σφαλμάτων, αλλά απλά η αποθήκευσή τους σε μεταβλητές και η χρησιμοποίησή τους στη ρύθμιση των ηλεκτρονόμων.

5.2.2 Αλγόριθμος

Όπως έχει ήδη αναφερθεί, ο αλγόριθμος για την επίλυση βραχυκυκλωμάτων στο πεδίο των φάσεων παρουσιάστηκε από τον Jun Zhu [17]. Χρησιμοποιούμε τον αλγόριθμο αυτό με τη διαφοροποίηση ότι εισάγουμε έναν επαναληπτικό βρόχο που πραγματοποιεί την ανάλυση βραχυκυκλωμάτων τόσες φορές όσοι είναι οι ζυγοί του συστήματος. Ο αλγόριθμος εξηγείται από το παρακάτω διάγραμμα ροής:



Διάγραμμα 2: Διάγραμμα ροής αλγόριθμου ανάλυσης βραχυκυκλωμάτων

Η ανάλυση βραχυκυκλωμάτων γίνεται από μια συνάρτηση με το όνομα “shortcircuits”, η οποία καλείται από το κυρίως πρόγραμμα. Κατ’ αρχήν γίνεται η εισαγωγή των δεδομένων που βρίσκονται στα αρχεία εισόδου και την αποθήκευσή τους σε μεταβλητές για μελλοντική τους χρήση από τη συνάρτηση. Μια από αυτές είναι ο πίνακας V_{pre} που περιέχει τις προ σφάλματος τάσεις των ζυγών. Στη συνέχεια δημιουργούνται οι πίνακες αγωγιμοτήτων θετικής, αρνητικής και μηδενικής ακολουθίας ακολουθώντας τη θεωρητική ανάλυση που έγινε στην παράγραφο 4.2. Λαμβάνουμε δηλαδή υπόψη τα δεδομένα όλων των στοιχείων εκτός από τους μετασχηματιστές και γίνεται χρήση της εξίσωσης (4.7). Μετασχηματίζουμε τους πίνακες αγωγιμοτήτων στο πεδίο των φάσεων και προκύπτει έτσι ο πίνακας Y_{matrix} διαστάσεων $(3 \cdot nNodes \times 3 \cdot nNodes)$. Η διάσταση αυτή δικαιολογείται από το γεγονός ότι λαμβάνονται υπόψη και οι τρεις φάσεις, αφού βρισκόμαστε στο πεδίο των φάσεων. Στη συνέχεια προστίθενται και οι αγωγιμότητες των μετασχηματιστών, ανάλογα με τη συνδεσμολογία του μετασχηματιστή και ο πίνακας που προκύπτει αντιστρέφεται, λαμβάνοντας έτσι τον πίνακα Z_{matrix} .

Ακολουθεί το τμήμα της συνάρτησης στο οποίο γίνεται ο υπολογισμός των σφαλμάτων. Ένας επαναληπτικός βρόχος for βοηθάει ώστε να υπολογιστεί το ρεύμα σφάλματος για σφάλμα σε καθέναν από τους ζυγούς του συστήματος. Για τριφασικό βραχυκύκλωμα υπολογίζεται αρχικά το ρεύμα σφάλματος σε καθεμία από τις τρεις φάσεις (τα μέτρα των ρευμάτων στην περίπτωση αυτή προκύπτουν ίσα) κάνοντας χρήση τη εξίσωσης (4.10) και τα ρεύματα αυτά αποθηκεύονται στον πίνακα I_{short} . Από αυτόν δημιουργείται το διάνυσμα I_{fault} διάστασης $nNodes$, όπου έχουμε για καθένα από τα σφάλματα το ρεύμα σφάλματος σε μια από τις τρεις φάσεις. Από τις σχέσεις

$$\begin{aligned} \Delta V &= Z_{matrix} \cdot I_{short} \\ V_{short} &= V_{pre} + \Delta V \end{aligned}$$

Υπολογίζονται οι τάσεις των ζυγών μετά την εκδήλωση του σφάλματος. Γνωρίζοντας τις τάσεις αυτές μπορούμε να υπολογίσουμε την πτώση τάσης σε καθεμία από τις γραμμές του συστήματος και συνεπώς το ρεύμα που τις διαρρέει. Τα ρεύματα των γραμμών αποθηκεύονται στον πίνακα I_{fault_line} διάστασης $(nLines \times nNodes)$, του οποίου κάθε στοιχείο $I_{fault_line}(i, j)$ δηλώνει το ρεύμα που ρέει στη γραμμή i για σφάλμα στο ζυγό j . Αφού γίνει και αυτός ο υπολογισμός περνάμε στην επόμενη επανάληψη του βρόχου for, δηλαδή στη μελέτη του σφάλματος στο επόμενο ζυγό.

Μετά την ανάλυση ροών φορτίου και βραχυκυκλωμάτων δημιουργούνται στο κυρίως πρόγραμμα κάποιοι πίνακες, οι οποίοι πρόκειται να χρησιμοποιηθούν στη συνέχεια. Αυτοί είναι:

Numbers

Πρόκειται για πίνακα διάστασης $(nLines \times nNodes)$, όπου $nLines$ ο αριθμός των γραμμών του συστήματος και $nNodes$ ο αριθμός των ζυγών. Σε κάθε γραμμή του πίνακα αυτού δυο στοιχεία είναι μη μηδενικά. Κάθε μη μηδενικό στοιχείο $Numbers(i, j)$ έχει τιμή ίση με τον αριθμό του ηλεκτρονόμου που βρίσκεται στη

γραμμή i του δικτύου και πρόσκειται στο ζυγό j . Η δημιουργία του πίνακα Numbers γίνεται με τη βοήθεια του πίνακα κατάστασης A , του οποίου κάθε γραμμή περιέχει μηδενικά στοιχεία, εκτός από δυο στοιχεία με τιμές 1 (στη στήλη που αντιστοιχεί στην αρχικό ζυγό της γραμμής) και -1 (στη στήλη που αντιστοιχεί στον τελικό ζυγό της γραμμής). Μετά τη δημιουργία του πίνακα Numbers, βρίσκουμε το μέγιστο στοιχείο του, το οποίο δηλώνει τον αριθμό των ηλεκτρονόμων του συστήματος και αποθηκεύεται στη μεταβλητή $nRelays$.

PhantomRelays

Είναι ο πίνακας που περιέχει τους αριθμούς των φανταστικών ηλεκτρονόμων του συστήματος. Εξετάζεται οι περίπτωση του μετασχηματιστή τριών τυλιγμάτων, για την παράσταση του οποίου χρησιμοποιείται ένας φανταστικός ζυγός, όπως θα εξηγηθεί και στη συνέχεια. Το πρόγραμμα βρίσκει τέτοιους φανταστικούς ζυγούς, διαβάζει από τον πίνακα Numbers τους ηλεκτρονόμους τους και τους αποθηκεύει στο διάνυσμα phantomRelays. Φανταστικός θεωρείται επίσης ο ηλεκτρονόμος που βρίσκεται αμέσως μετά από μια γεννήτρια, όταν αυτοί συνδέεται με έναν μετασχηματιστή. Εξετάζεται και αυτοί οι περίπτωση και οι αντίστοιχοι ηλεκτρονόμοι συμπεριλαμβάνονται στο διάνυσμα phantomRelays.

Primaries

Πρόκειται για πίνακα που περιέχει $nRelays$ γραμμές και που σε κάθε γραμμή g αποθηκεύονται οι ηλεκτρονόμοι εκείνου για τους οποίους ο g αποτελεί δευτερεύουσα προστασία. Ο πίνακας primaries δηλαδή περιέχει τους πρωτεύοντες ηλεκτρονόμους κάθε ηλεκτρονόμου. Για τη δημιουργία του πίνακα χρησιμοποιείται ο πίνακας Numbers. Συγκεκριμένα, για κάθε ηλεκτρονόμο $Numbers(i, j)$ του πίνακα Numbers αναζητούμε στην ίδια γραμμή i το άλλο μη μηδενικό στοιχείο, δηλαδή τον ηλεκτρονόμο στο άλλο άκρο της γραμμής. Έστω ότι αυτό βρίσκεται στη στήλη j' , δηλαδή ο ηλεκτρονόμος ανήκει στο ζυγό j' . Στη συνέχεια διατρέχουμε τη στήλη j' και κάθε μη μηδενικό στοιχείο διάφορο του $Numbers(i, j')$ που βρίσκουμε το αποθηκεύουμε στη γραμμή $Numbers(i, j)$ του πίνακα primaries. Πρόκειται για ηλεκτρονόμο που βρίσκεται σε κάποια άλλη γραμμή και πρόσκειται στο ζυγό j' . Είναι δηλαδή ο ηλεκτρονόμος στη ζώνη του οποίου ο $Numbers(i, j)$ προσφέρει δευτερεύουσα προστασία. Κάθε φορά ελέγχεται αν ο υποψήφιος primary ηλεκτρονόμος είναι φανταστικός. Αν ισχύει κάτι τέτοιο, ο ηλεκτρονόμος δεν συμπεριλαμβάνεται στον πίνακα primaries και η διαδικασία που περιγράφηκε για τη διάσχιση του πίνακα Numbers επαναλαμβάνεται με αφετηρία τώρα τον φανταστικό αυτό ηλεκτρονόμο. Η διαδικασία αυτή συνεχίζεται μέχρι να βρεθεί ηλεκτρονόμος που να μην είναι φανταστικός. Ο νέος ηλεκτρονόμος που θα προκύψει θα είναι ο primary του ηλεκτρονόμου από τον οποίο αρχικά ξεκινήσαμε.

BreakersOfNode

Είναι ένα διάνυσμα διάστασης $(nNodes \times 1)$ που το στοιχείο κάθε γραμμής i έχει τιμή ίση με τον αριθμό των ηλεκτρονόμων που ανήκουν στο ζυγό i .

I_base

Το διάνυσμα αυτό έχει διάσταση ($nRelays \times 1$) και περιέχει τη βάση ισχύος που αντιστοιχεί στο σημείο που βρίσκεται ο κάθε ηλεκτρονόμος. Ο υπολογισμός της βάσης ισχύος γίνεται από τον τύπο:

$$I_{base} = \frac{S_{base}}{\sqrt{3} V_{base}}$$

Όπου η βασική τάση και η βασική ισχύς είναι αποθηκευμένες στις εξόδους S.Misc.BaseMVA και S.Bus.KV αντίστοιχα του προγράμματος ανάλυσης ροών φορτίου.

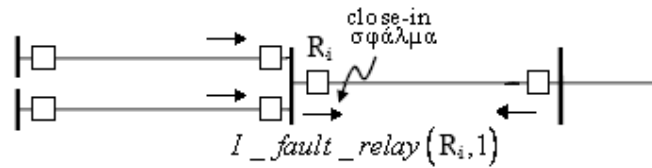
I_flow_relay

Είναι διάνυσμα με διάσταση ίδια με αυτή του προηγούμενου διανύσματος, δηλαδή ($nRelays \times 1$). Περιέχει για κάθε ηλεκτρονόμο του ρεύμα κανονικής λειτουργίας σε A που διαρρέει τη γραμμή στο σημείο που βρίσκεται ο ηλεκτρονόμος. Το ρεύματα ροής είναι αποτέλεσμα της ανάλυσης βραχυκυκλωμάτων και αποθηκεύονται στο διάνυσμα S.Flow. Το διάνυσμα όμως αυτό περιέχει τα ρεύματα σε μιγαδική ανά μονάδα μορφή. Για κάθε ηλεκτρονόμο $Numbers(i, j)$ βρίσκουμε το ρεύμα της γραμμής του $S.Flow(i, 1)$, λαμβάνουμε το μέτρο του ρεύματος αυτού και το πολλαπλασιάζουμε επί το βασικό ρεύμα στην περιοχή του ηλεκτρονόμου, δηλαδή επί $I_{base}(Numbers(i, j), 1)$. Το αποτέλεσμα αποθηκεύεται στη θέση $I_{flow_relay}(Numbers(i, j), 1)$.

I_fault_relay

Πρόκειται για διάνυσμα διάστασης ($nRelays \times 1$), το οποίο αρχικοποιείται μηδενικό. Κάθε γραμμή του αντιστοιχεί σε έναν ηλεκτρονόμο και περιέχει το ρεύμα σε A που διαρρέει τη γραμμή στο σημείο που βρίσκεται ο ηλεκτρονόμος για close-in στον ηλεκτρονόμο σφάλμα. Στο Σχήμα 10 που ακολουθεί παρουσιάζεται ένα close-in στον ηλεκτρονόμο R_i σφάλμα. Ο υπολογισμός του ρεύματος αυτού γίνεται ως εξής: Για κάθε ηλεκτρονόμο προσθέτουμε τα ρεύματα όλων των γραμμών που έχουν ένα από τα δυο άκρα τους στο ζυγό του ηλεκτρονόμου, εκτός από τη γραμμή στην οποία βρίσκεται ο εν λόγω ηλεκτρονόμος. Το αποτέλεσμα πολλαπλασιάζεται με το βασικό ρεύμα στο σημείο του ηλεκτρονόμου για να προκύψει το ρεύμα σε A. Με αναφορά στους χρησιμοποιούμενους πίνακες έχουμε: Για κάθε στοιχείο $Numbers(i, j)$ διατρέχουμε τη στήλη j του πίνακα Numbers αναζητώντας άλλα μη μηδενικά στοιχεία, δηλαδή αναζητούμε τους ηλεκτρονόμους του ζυγού j που ανήκουν σε γραμμή διαφορετική από αυτή που ανήκει ο $Numbers(i, j)$. Για κάθε στοιχείο που βρίσκουμε, έστω το $Numbers(line, j)$ που βρίσκεται στη γραμμή line, προσθέτουμε στο στοιχείο $I_{fault_relay}(Numbers(i, j), 1)$ το γινόμενο $I_{fault_line}(line, j) \cdot I_{base}(Numbers(line, j), 1)$, δηλαδή το ρεύμα σε A που διαρρέει τη γραμμή line. Αφού προστεθούν τα ρεύματα από όλους τις γραμμές εκτός

από αυτή που βρίσκεται ο $Numbers(i, j)$, έχουμε το ρεύμα που διαρρέει τον ηλεκτρονόμο $Numbers(i, j)$ για close-in σε αυτόν σφάλμα.



Σχήμα 10 : Close-in σφάλμα

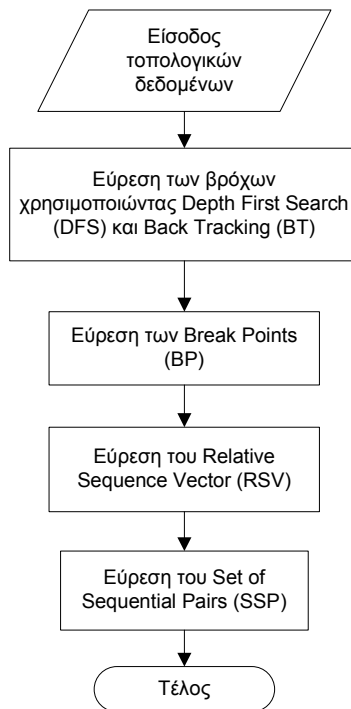
Μετά την ανάλυση ροών φορτίου, την ανάλυση βραχυκυκλωμάτων και τους υπολογισμούς που αναλύθηκαν αμέσως πριν έχουν δημιουργηθεί κάποιοι πίνακες, οι οποίοι θα χρησιμοποιηθούν στη συνέχεια. Συγκεντρωτικά αυτοί δίνονται στον επόμενο πίνακα:

$Numbers(i, j)$	Κάθε μη μηδενικό στοιχείο ισούται με τον αριθμό του ηλεκτρονόμου που βρίσκεται στη γραμμή i και στο ζυγό j .
$phantomRelays$	Περιέχει όλους τους μη φανταστικούς ηλεκτρονόμους.
$primaries(i, j)$	Η γραμμή i περιέχει όλους τους ηλεκτρονόμους που για τη ζώνη του καθενός ο i αποτελεί δευτερεύουσα προστασία.
$BreakersOfNode(i, 1)$	Περιέχει τον αριθμό των ηλεκτρονόμων του ζυγού i .
$I_base(i, 1)$	Περιέχει τη βάση ισχύος του ηλεκτρονόμου i .
$I_flow_relay(i, 1)$	Το ρεύμα κανονικής λειτουργίας σε A που ρέει στη γραμμή στη θέση που βρίσκεται ο ηλεκτρονόμος i .
$I_fault_line(i, j)$	Το ρεύμα σφάλματος σε α.μ. που αντιστοιχεί στον ηλεκτρονόμο i όταν συμβεί σφάλμα στο ζυγό j .
$I_fault_relay(i, 1)$	Το ρεύμα σφάλματος σε A που ρέει στη γραμμή στη θέση που βρίσκεται ο ηλεκτρονόμος i όταν συμβεί close-in σφάλμα στον ηλεκτρονόμο i .

Πίνακας 3 : Πίνακες που θα χρησιμοποιηθούν στη συνέχεια

5.3 Τοπολογική ανάλυση συστήματος

Ο αλγόριθμος που θα παρουσιαστεί στη συνέχεια διεξάγει την τοπολογική ανάλυση κάθε συστήματος με πολλούς βρόχους και καταλήγει σε μια «βέλτιστη» ακολουθία ηλεκτρονόμων. Ένα διάγραμμα ροής που δείχνει τα βασικά βήματα της τοπολογικής ανάλυσης δίνεται στο Διάγραμμα 3.

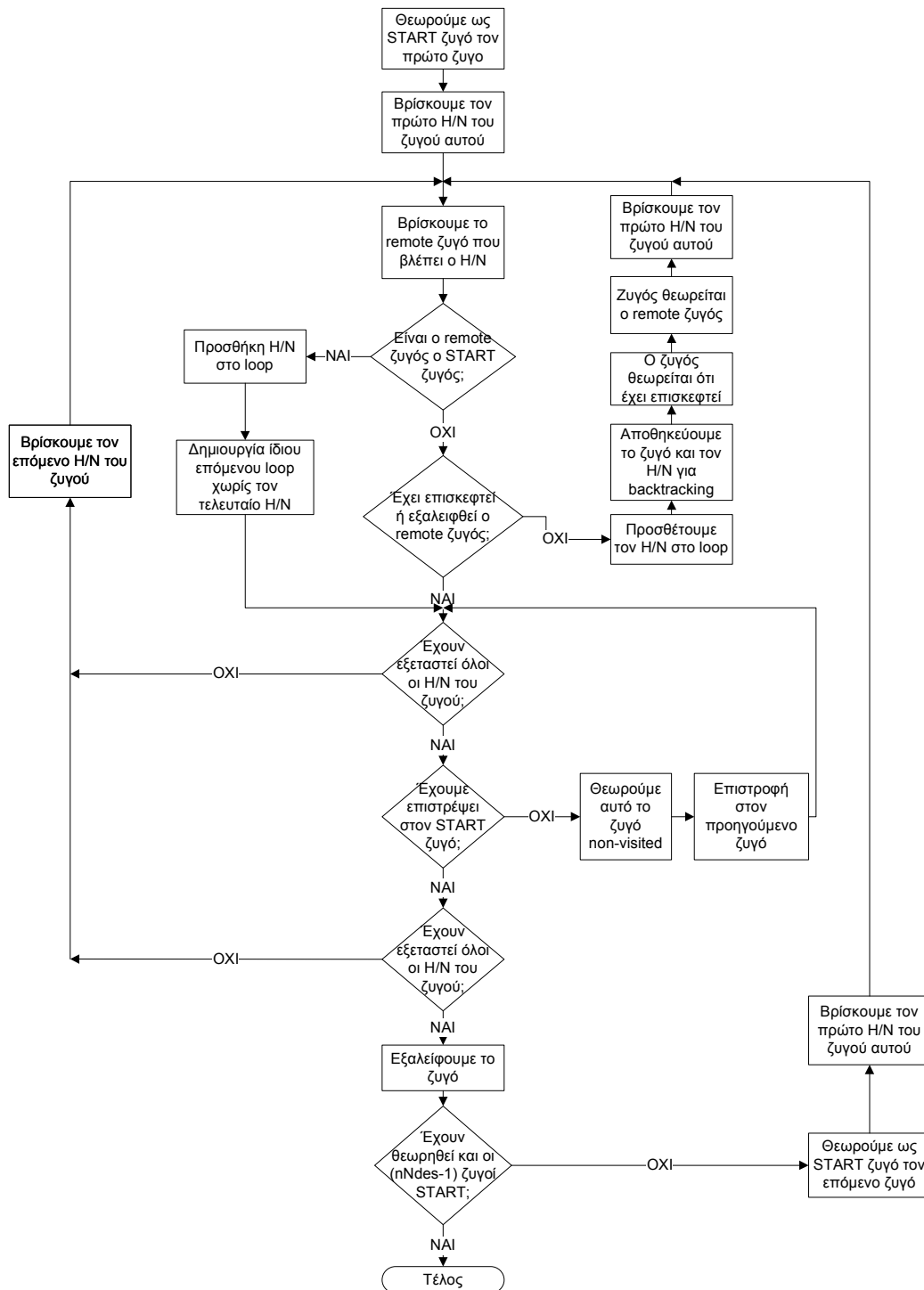


Διάγραμμα 3 : Διάγραμμα ροής τοπολογικής ανάλυσης συστήματος

Η τοπολογική ανάλυση του συστήματος γίνεται από ένα υποπρόγραμμα με όνομα “Topol_anal”, το οποίο καλείται από το κυρίως πρόγραμμα. Στη συνέχεια παρουσιάζεται η υλοποίηση της εύρεσης καθενός από τα στοιχεία που υπολογίζει το υποπρόγραμμα αυτό.

5.3.1 Βρόχοι του συστήματος

Όπως αναφέρθηκε στην παράγραφο 4.3.1, η μέθοδος ανίχνευσης που θα χρησιμοποιηθεί για τη εύρεση των βρόχων είναι μια τροποποιημένη εκδοχή της Μεθόδου Αναζήτησης κατά Βάθος (Depth First Search- DFS) και της Διαδικασίας Επανιχνηλάτησης (Back Tracking- BT). Ακολουθήσαμε τη γενική ιδέα που περιγράφεται στο [18] τροποποιώντας τη, και δημιουργήσαμε πρόγραμμα που να υλοποιεί τον αλγόριθμο αυτό. Το πρόγραμμα αυτό αποτελεί συνάρτηση, η οποία καλείται από το κυρίως πρόγραμμα και επιστρέφει δυο πίνακες που έχουν τόσες γραμμές όσοι οι βρόχοι του συστήματος. Ο πρώτος ονομάζεται “loopsBreakers” και κάθε γραμμή του δίνει τους ηλεκτρονόμους που σχηματίζουν έναν βρόχο. Ο δεύτερος ονομάζεται “loopsBuses” και περιέχει σε κάθε γραμμή του τους ζυγούς που σχηματίζουν βρόχο, δηλαδή κάθε στοιχείο $\text{loopsBuses}(i,j)$ είναι ο ζυγός στον οποίο ανήκει το στοιχείο $\text{loopsBreakers}(i,j)$. Το διάγραμμα ροής που περιγράφει το πρόγραμμα δίνεται παρακάτω:



Διάγραμμα 4: Διάγραμμα ροής αλγόριθμου εύρεσης των βρόχων

Επεξηγηματικά μπορούμε να πούμε τα ακόλουθα:

Ο αλγόριθμος αποτελείται από τρεις επαναληπτικούς βρόχους. Ένας εξωτερικός βρόχος for, ο οποίος καθορίζει το ζυγό από τον οποίο ξεκινάμε, περιέχει ένα βρόχο while, ο οποίος προχωράει μέσα στο σύστημα προς αναζήτηση βρόχου

ακολουθώντας προς τα εμπρός πορεία, και αυτός με τη σειρά του περιέχει έναν άλλο βρόχο `while`, ο οποίος ακολουθεί πορεία οπισθοδρόμησης αναζητώντας ηλεκτρονόμους που δεν έχουν εξεταστεί.

Αφετηρία είναι ο πρώτος ζυγός με βάση τη σειρά που έχουν δηλωθεί, έστω ο ζυγός j , και το πρώτο βήμα είναι η αναζήτηση στον πίνακα `Numbers` του πρώτου μη μηδενικού στοιχείου της στήλης j . Αυτό σημαίνει ότι θεωρούμε τον πρώτο ηλεκτρονόμο (με βάση τη σειρά που έχουν δηλωθεί) του ζυγού αυτού. Έστω ότι ο ηλεκτρονόμος αυτός βρίσκεται στη γραμμή i του πίνακα `Numbers`, πράγμα που σημαίνει ότι ανήκει στη γραμμή i του συστήματος. Αναζητούμε στη γραμμή i το άλλο μη μηδενικό στοιχείο, εκτός από το `Numbers( $i, j$ )`, δηλαδή τον ηλεκτρονόμο στο άλλο άκρο της γραμμής. Αυτός είναι ο ηλεκτρονόμος που στο διάγραμμα ροής αναφέρεται ως ζεύγος. Ο ηλεκτρονόμος και το ζεύγος του προστίθενται στον πίνακα “`autos`”, όπου αποθηκεύονται όλοι οι ηλεκτρονόμοι που εξετάζονται.

Στη συνέχεια ξεκινάει ο πρώτος `while` επαναληπτικός βρόχος, ο οποίος διατρέχει το σύστημα προς εύρεση βρόχου. Ονομάζουμε τη στήλη στην οποία βρίσκεται το ζεύγος του υπό εξέταση ηλεκτρονόμου, δηλαδή του `Numbers( $i, j$ )`, “`remote bus`”. Μέσω δηλαδή της γραμμής του ηλεκτρονόμου `Numbers( $i, j$ )` προχωρούμε στον επόμενο υπό εξέταση ζυγό.

Στη συνέχεια ελέγχουμε εάν ο `remote` ζυγός είναι αυτός από τον οποίο ξεκινήσαμε, δηλαδή ο j . Εάν δε συμβαίνει αυτό, ελέγχουμε επίσης εάν ο `remote` ζυγός έχει ήδη επισκεφτεί ή εξαλειφθεί από το σύστημα. Εάν τίποτα από όλα αυτά δε συμβαίνει, αυτό σημαίνει ότι δεν έχει ολοκληρωθεί ο βρόχος, οπότε η πορεία θα πρέπει να συνεχιστεί. Αποθηκεύουμε τον εν λόγω ηλεκτρονόμο, τον `Numbers( $i, j$ )`, στην πρώτη σειρά του πίνακα `loopsBreakers` και το ζυγό του, τον j , στην πρώτη σειρά του πίνακα `loopsBuses`, δηλαδή θεωρείται το πρώτο στοιχείο του υπό κατασκευή βρόχου. Ο ηλεκτρονόμος αποθηκεύεται επίσης στο διάνυσμα “`runningbreakers`”, όπου αποθηκεύονται όλοι οι ηλεκτρονόμοι που συμμετέχουν σε έναν βρόχο. Ο ζυγός του ηλεκτρονόμου, ο j , προστίθεται στο διάνυσμα “`visitedBus`”, στο οποίο θα αποθηκεύονται οι ζυγοί που επισκεπτόμαστε, δηλαδή αυτού από τους οποίους περνάμε κατά τη δημιουργία του βρόχου. Τώρα ζυγός θεωρείται ο `remote` ζυγός και η διαδικασία επαναλαμβάνεται, δηλαδή βρίσκουμε τον πρώτο ηλεκτρονόμο του ζυγού και το ζεύγος του και μεταβαίνουμε στην αρχή του επαναληπτικού βρόχου `while`.

Στην περίπτωση που ο `remote` ζυγός είναι αυτός από τον οποίο ξεκινήσαμε, δηλαδή ο j που στο εξής θα αναφέρεται ως `start` ζυγός, τότε έχει δημιουργηθεί ένας βρόχος, αφού ξεκινήσαμε και καταλήξαμε στον ίδιο ζυγό. Αποθηκεύουμε τον ηλεκτρονόμο που εξετάζεται τη δεδομένη στιγμή στην τρέχουσα γραμμή του πίνακα `loopsBreakers` και το ζυγό του στην τρέχουσα γραμμή του πίνακα `loopsBuses`. Πρόκειται για τα τελευταία στοιχεία του βρόχου, γι’ αυτό μετά την καταγραφή τους προχωρούμε στην επόμενη γραμμή των πινάκων `loopsBreakers` και `loopsBuses` και αντιγράφουμε σε αυτή όλα τα στοιχεία της προηγούμενης γραμμής εκτός από το τελευταίο. Η νέα αυτή γραμμή καταγράφει ένα τμήμα του επόμενου πιθανού βρόχου. Αυτό σημαίνει ότι θα παραμείνουμε στον τελευταίο ζυγό και θα θεωρήσουμε κάποιον άλλο ηλεκτρονόμο του, όπως θα φανεί στη συνέχεια. Αντιγράφουμε όλους τους ηλεκτρονόμους εκτός από τον τελευταίο, αφού ως εκείνο το σημείο ο νέος πιθανός βρόχος έχει την ίδια

πορεία με τον προηγούμενο. Στη συνέχεια θα προστεθούν στο νέο βρόχο οι επόμενοι ηλεκτρονόμοι που απαιτούνται για την ολοκλήρωσή του.

Είτε στην περίπτωση που ο remote ζυγός είναι ο start είτε στην περίπτωση που ο remote ζυγός έχει επισκεφτεί, δηλαδή βρίσκεται στο διάνυσμα visitedBus, ή εξαλειφθεί από το σύστημα μπαίνουμε στον επόμενο while βρόχο, ο οποίος περιέχεται στον προηγούμενο βρόχο while. Αυτός βρίσκει άλλους ηλεκτρονόμους για να συνεχιστεί η αναζήτηση εξετάζοντας τους υπόλοιπους ηλεκτρονόμους του τελευταίου ζυγού και αν από την εξέταση αυτή δεν προκύψει αποτέλεσμα, οπισθοχωρεί στους προηγούμενους ζυγούς μέχρι να βρεθεί νέος βρόχος. Ανακόπτεται δηλαδή η προς τα εμπρός πορεία που ακολουθούσαμε μέχρι τώρα. Πρώτα εξετάζουμε εάν υπάρχει άλλος ηλεκτρονόμος στον τρέχοντα ζυγό (currentBus), ο οποίος είναι ο τελευταίος υπό εξέταση ζυγός. Για να γίνει αυτό διατρέχουμε τη στήλη currentBus του πίνακα Numbers και όποτε βρίσκουμε μη μηδενικό στοιχείο, δηλαδή ηλεκτρονόμο στο ζυγό, ελέγχουμε εάν αυτός υπάρχει στο διάνυσμα autos, δηλαδή εάν έχει εξεταστεί ήδη.

Αν από την παραπάνω διαδικασία δεν βρεθεί κάποιος ηλεκτρονόμος, οπότε έχουν εξεταστεί όλοι οι ηλεκτρονόμοι του τρέχοντα ζυγού, ακολουθεί ένας ακόμη έλεγχος. Ελέγχουμε εάν έχουμε οπισθοχωρήσει στον start ζυγό, δηλαδή σε αυτόν από τον οποίο ξεκινήσαμε. Εάν δε συμβαίνει αυτό, μπορούμε να οπισθοχωρήσουμε στον προηγούμενο ζυγό και να συνεχίσουμε από εκεί την αναζήτηση βρόχων. Αυτό συνεπάγεται ότι θα πρέπει να βρεθεί στο διάνυσμα visitedBus ο τρέχον ζυγός και να διαγραφεί από το διάνυσμα αυτό. Το ίδιο θα πρέπει να γίνει και στην τρέχουσα γραμμή του πίνακα loopsBuses. Επίσης διαγράφονται οι αντίστοιχοι ηλεκτρονόμοι από τα διανύσματα running, autos και από την τρέχουσα γραμμή του πίνακα loopsBreakers. Διαγράφονται δηλαδή όλα τα στοιχεία τα οποία είχαν καταγραφεί επειδή εξετάστηκαν, δεν οδήγησαν όμως στη δημιουργία βρόχου. Ως currentBus θεωρείται τώρα ο ζυγός που βρίσκεται στην αμέσως προηγούμενη θέση του visitedBuses, δηλαδή ο αμέσως προηγούμενος ζυγός στην πορεία που είχαμε διαγράψει. Αυτό που στην ουσία κάνουμε είναι να αγνοούμε τον τελευταίο ζυγό, αφού δεν υπάρχουν άλλοι ηλεκτρονόμοι που μπορούμε να εξετάσουμε στο ζυγό αυτό, και να γυρνάμε στον προτελευταίο ζυγό, όπου και συνεχίζουμε τη διαδικασία. Στη συνέχεια επιστρέφουμε στο σημείο όπου εξετάζεται εάν υπάρχουν ηλεκτρονόμοι στον τρέχοντα ζυγό που δεν έχουν εξεταστεί.

Εάν στο τελευταίο έλεγχο προκύψει θετική απάντηση, δηλαδή εάν οπισθοχωρώντας έχουμε επιστρέψει στον start ζυγό, εξετάζουμε εάν υπάρχει ηλεκτρονόμος στο ζυγό αυτό που δεν έχει εξεταστεί. Εάν ναι, θεωρούμε τον ηλεκτρονόμο του start ζυγού που δεν έχει εξεταστεί και το ζεύγος του και συνεχίζουμε τις επαναλήψεις στον αρχικό while βρόχο χωρίς να βγούμε από αυτόν. Εάν όχι, εξαλείφουμε αυτόν το ζυγό, δηλαδή τον προσθέτουμε στο διάνυσμα eliminatedBus. Βγαίνουμε από το while βρόχο και βρισκόμαστε πλέον στο for βρόχο. Εκεί θεωρούμε start ζυγό τον επόμενο ζυγό και η διαδικασία επαναλαμβάνεται από την αρχή, δηλαδή βρίσκουμε τον πρώτο ηλεκτρονόμο του start ζυγού και το ζεύγος του και μπαίνουμε ξανά στον αρχικό while βρόχο. Για συνολικό αριθμό ζυγών nNodes, οι (nNodes-1) ζυγοί θα θεωρηθούν start, ένας βέβαια κάθε φορά, καθώς δεν έχει νόημα να θεωρηθεί start και ο τελευταίος. Ο επαναληπτικός βρόχος for έχει λοιπόν άνω όριο (nNodes-1).

5.3.2 Break Points

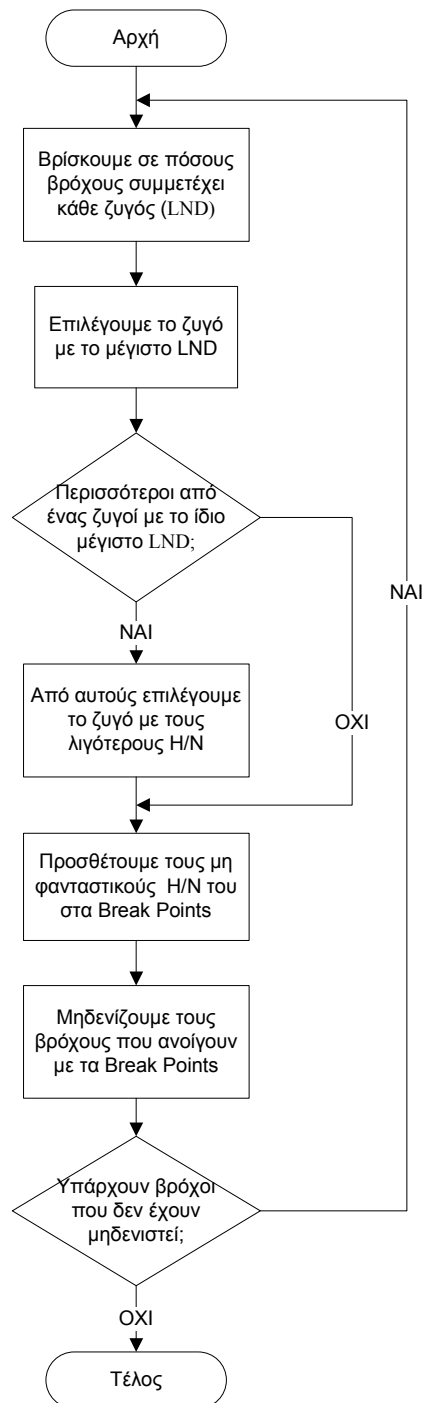
Το επόμενο βήμα είναι ο καθορισμός του ελάχιστου συνόλου των Break Points, δηλαδή του ελάχιστου συνόλου ηλεκτρονόμων που ανοίγουν όλους τους βρόχους. Η εύρεση των ηλεκτρονόμων αυτών θα γίνει με δυο εναλλακτικές μεθόδους. Γι' αυτό το λόγο εμφανίζεται κατά τη εκτέλεση στην οθόνη ένα μήνυμα που καλεί το χρήστη να επιλέξει τη μέθοδο με την οποία θα συνεχιστεί η διαδικασία..

5.3.2.1 Μέθοδος 1

Χρησιμοποιούμε στον αντίστοιχο κώδικα, που είναι τμήμα του κυρίως προγράμματος, την έννοια Βαθμός Συμμετοχής Ζυγού στους Βρόχους (Loop Node Degree- LND), η οποία εξηγήθηκε στην παράγραφο 4.3.2.1. Θυμίζουμε εδώ τι παριστάνει η έννοια αυτή για κάθε ζυγό.

LND: Ισούται με το πόσες φορές συμμετέχει ο ζυγός αυτός στους βρόχους του συστήματος.

Ο αλγόριθμος που δημιουργήθηκε για την εύρεση του συνόλου των Break Points εξηγείται καλύτερα με ένα διάγραμμα ροής.



Διάγραμμα 5: Διάγραμμα ροής για εύρεση των Break Points με την πρώτη μέθοδο

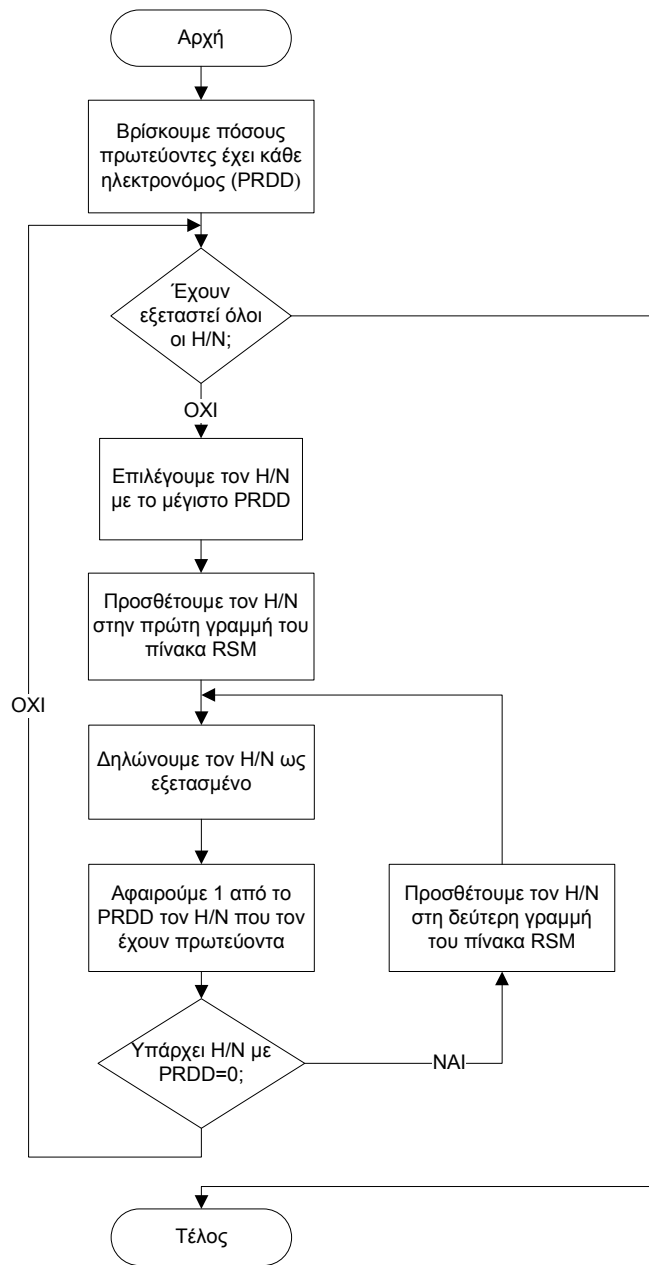
Το τμήμα του προγράμματος που εκτελεί την εύρεση των Break Points αποτελείται από ένα βρόχο while, ο οποίος επαναλαμβάνεται μέχρι να βρεθούν τα Break Points και περιέχει άλλους επαναληπτικούς βρόχους και ελέγχους. Αρχικά αναφέρουμε ότι μετά την ολοκλήρωση της εύρεσης των βρόχων του συστήματος, οι πίνακες loopsBreakers και loopsBuses αντιγράφονται δημιουργώντας δυο νέους ίδιους πίνακες, που τους ονομάζουμε loopsBreakers_br και loopsBuses_br. Αυτοί οι πίνακες θα χρησιμοποιηθούν για την εύρεση των Break Points. Ο λόγος που γίνεται αυτό

είναι ότι επιθυμούμε να υπάρχουν πίνακες που περιέχουν τους ηλεκτρονόμους και τους ζυγούς κάθε βρόχου μετά την ολοκλήρωση της διαδικασίας εύρεσης των Break Points, κατά την οποία οι αντίστοιχοι πίνακες χρησιμοποιούνται και εν τέλει μηδενίζονται. Διατρέχουμε λοιπόν τον πίνακα `loopsBuses_br` και αποθηκεύουμε σε έναν πίνακα-στήλη “`times`”, με διάσταση ίση με το πλήθος των ζυγών, πόσες φορές συναντάται σε βρόχους ο αντίστοιχος ζυγός. Υπολογίζουμε δηλαδή το LND κάθε ζυγού. Στη συνέχεια βρίσκουμε το μέγιστο στοιχείο του `times`, και θεωρούμε τη γραμμή στην οποία βρίσκεται, δηλαδή το ζυγό με το μεγαλύτερο LND, αν ο ζυγός αυτός δεν είναι φανταστικός. Αν περισσότεροι του ενός ζυγοί έχουν το ίδιο LND, το πρόγραμμα ανατρέχει στις αντίστοιχες γραμμές του πίνακα `BreakersOfNode` και επιλέγει από τις γραμμές αυτές εκείνη τη γραμμή `i` που έχει το μικρότερο `BreakersOfNode(i,1)`. Έτσι επιλέγεται από τους υποψήφιους ζυγούς αυτός με το μικρότερο αριθμό ηλεκτρονόμων. Στην στήλη `i` του πίνακα `Numbers` βρίσκουμε τα μη μηδενικά στοιχεία και αν πρόκειται για μη φανταστικούς ηλεκτρονόμους, τους προσθέτουμε στο διάνυσμα `breakpoints`. Περιλαμβάνονται έτσι οι H/N του ζυγού `i` στα Break Points.

Έπειτα εξετάζουμε για κάθε γραμμή του πίνακα `loopsBreakers_br` και για κάθε Break Point εάν υπάρχει σε αυτή τη γραμμή το Break Point αυτό. Ελέγχουμε δηλαδή εάν ο συγκεκριμένος βρόχος ανοίγει με κάποιο από τα Break Points. Αν συμβαίνει αυτό, μηδενίζουμε τη γραμμή αυτή, δηλαδή το αντίστοιχο βρόχο, από τους πίνακες `loopsBreakers` και `loopsBuses`. Το εάν υπάρχουν βρόχοι που εξακολουθούν να είναι μη μηδενικοί, διαπιστώνεται από την τιμή μιας μεταβλητής, που έχει αρχικά μηδενική τιμή και η τιμή της αυξάνεται κάθε φορά που βρίσκεται ένας τέτοιος βρόχος. Αν η μεταβλητή αυτή έχει μη μηδενική τιμή, ξαναμπαίνουμε στον εξωτερικό βρόχο `while` και η διαδικασία επαναλαμβάνεται. Στην αντίθετη περίπτωση, η μεταβλητή ελέγχου της εντολής `while` αλλάζει τιμή και έτσι η είσοδος στο βρόχο δεν είναι πλέον δυνατή. Έχουν ανοίξει δηλαδή όλοι οι βρόχοι, συνεπώς έχουν βρεθεί τα Break Points και είναι αποθηκευμένα στο διάνυσμα “`breakpoints`”.

5.3.2.2 Μέθοδος 2

Ακολουθώντας την περιγραφική διαδικασία που αναλύθηκε στην παράγραφο 4.3.2.2 γράφουμε πρόγραμμα που υπολογίζει τα Break Points με τη δεύτερη μέθοδο. Θυμίζουμε ότι στη μέθοδο αυτή χρησιμοποιείται το μέγεθος PRDD. Το PRDD κάθε ηλεκτρονόμου x ισούται με τον αριθμό των ηλεκτρονόμων που σε καθενός τη ζώνη ο x προσφέρει δευτερεύουσα προστασία.



Διάγραμμα 6 : Διάγραμμα ροής για εύρεση των Break Points με τη δεύτερη μέθοδο

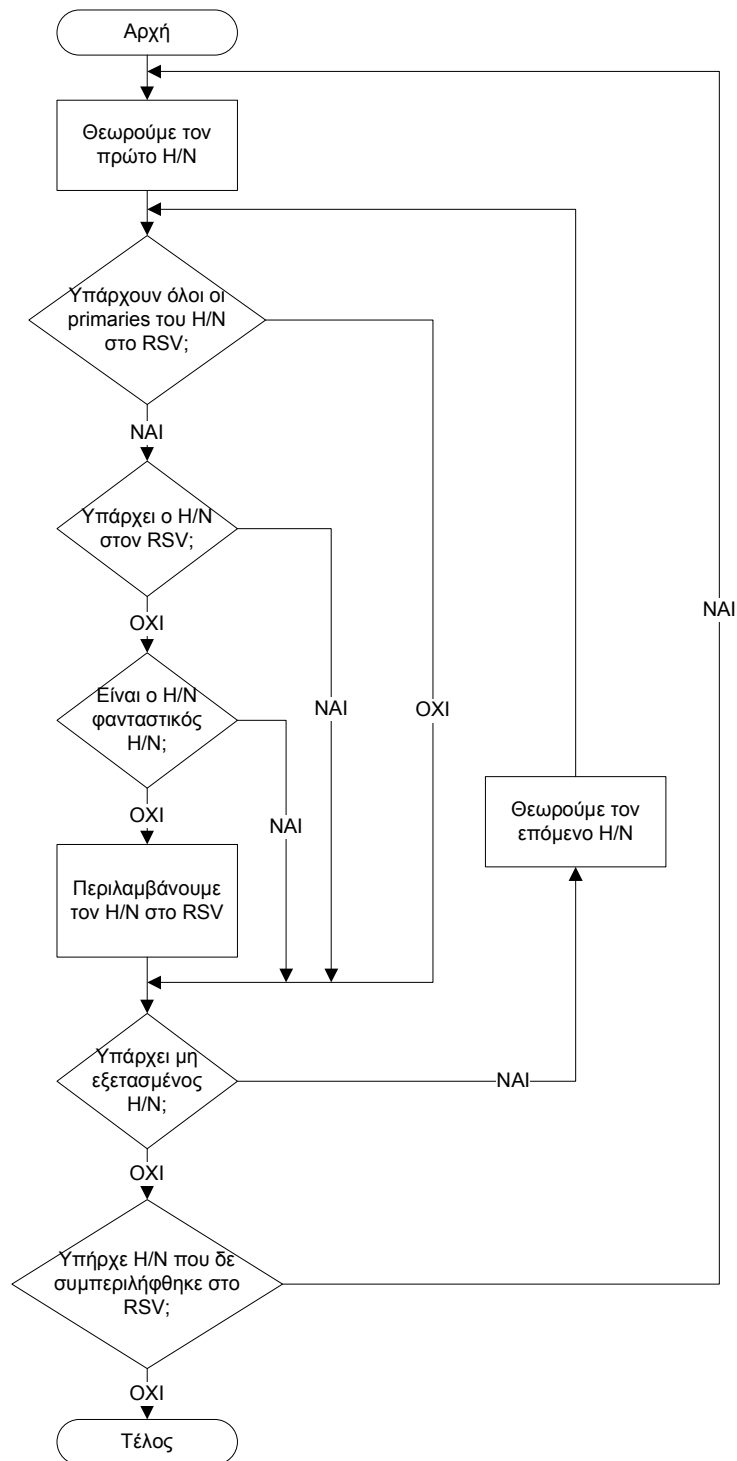
Αρχικά δημιουργείται ο πίνακας $NumberOfPrim$ διάστασης $(n Relays \times 1)$, που κάθε στοιχείο του x_{i1} ισούται με τον αριθμό των ηλεκτρονόμων στη ζώνη των οποίων ο ηλεκτρονόμος i παρέχει δευτερεύουσα προστασία. Πρόκειται δηλαδή για το PRDD κάθε ηλεκτρονόμου. Στη συνέχεια δίνουμε την τιμή (-1) στα στοιχεία $NumberOfPrim(i,1)$ με i τον αριθμό κάθε φανταστικού ηλεκτρονόμου. Ακολουθεί ένας επαναληπτικός βρόχος `while`, ο οποίος επαναλαμβάνεται όσο δεν υπάρχει ηλεκτρονόμος με $PRDD=0$. Εντός του βρόχου αυτού βρίσκουμε πρώτα τον ηλεκτρονόμο με το μεγαλύτερο PRDD, δηλαδή με τους περισσότερους πρωτεύοντες ηλεκτρονόμους, τον τοποθετούμε στο διάνυσμα `breakpoints` και στην πρώτη σειρά του πίνακα RSM. Δίνουμε επίσης τιμή (-1) στο στοιχείο του πίνακα `NumberOfPrim`

που αντιστοιχεί στον ηλεκτρονόμο αυτό, δηλώνοντας έτσι ότι ο ηλεκτρονόμος έχει εξεταστεί, και μειώνουμε κατά 1 την τιμή των στοιχείων του πίνακα NumberOfPrim που αντιστοιχούν σε ηλεκτρονόμους που έχουν πρωτεύοντα των ηλεκτρονόμο που μόλις εξετάσαμε. Όταν ένας ηλεκτρονόμος είναι εξετασμένος, δεν πρόκειται να εξεταστεί πάλι, αφού στις αντίστοιχες συνθήκες που κάθε φορά ελέγχονται περιλαμβάνεται και το εάν η τιμή του ηλεκτρονόμου στον πίνακα NumberOfPrim είναι μηδενική. Στη συνέχεια, χωρίς να βγούμε από τον εξωτερικό βρόχο while, εισερχόμαστε σε έναν εσωτερικό while βρόχο, ο οποίος επαναλαμβάνεται όσο υπάρχουν ηλεκτρονόμοι με $PRDD = 0$. Εντός του βρόχου αυτού ελέγχονται ένας – ένας οι ηλεκτρονόμοι και αν βρεθεί κάποιος με μηδενικό $PRDD$, καταγράφεται στη δεύτερη σειρά του πίνακα RSM, δηλώνεται ως εξετασμένος και μειώνεται κατά 1 το $PRDD$ των στοιχείων που τον περιέχουν στους πρωτεύοντες ηλεκτρονόμους τους. Μετά την έξοδο από το βρόχο αυτό, δηλαδή όταν δεν υπάρχουν ηλεκτρονόμοι με $PRDD = 0$, προχωρούμε στην επόμενη επανάληψη του εξωτερικού βρόχου while, αναζητούμε δηλαδή τον ηλεκτρονόμο με το μεγαλύτερο $PRDD$. Έξοδο από το βρόχο αυτό έχουμε όταν έχουν εξεταστεί όλοι οι ηλεκτρονόμοι, πράγμα που δηλώνεται από την τιμή μιας μεταβλητής, η οποία αυξάνει κατά 1 όταν βρεθεί στον πίνακα NumberOfPrim ηλεκτρονόμος με τιμή (-1). Στην πρώτη γραμμή του πίνακα RSM είναι αποθηκευμένα τα Break Points, ενώ το διάνυσμα RSV προκύπτει από τη διαδοχική ανάγνωση της πρώτης και της δεύτερης σειράς του πίνακα RSM.

5.3.3 RSV

5.3.3.1 Μέθοδος 1

Ο πρώτος τρόπος τοποθέτησης των ηλεκτρονόμων στο Διάνυσμα Σχετικής Ακολουθίας (Relative Sequence Vector- RSV) αναφέρθηκε στην παράγραφο 4.3.3.1. Εδώ παρουσιάζονται λεπτομέρειες σχετικές με την υλοποίησή του και τη δημιουργία του αντίστοιχου προγράμματος.



Διάγραμμα 7: Διάγραμμα ροής για εύρεση Διάνυσματος Σχετικής Ακολουθίας (RSV)

Πρόκειται να εξηγήσουμε το πρόγραμμα αναφέροντας και τις αντίστοιχες μεταβλητές που χρησιμοποιούνται. Αρχικά πρέπει να ειπωθεί ότι στο διάνυσμα RSV υπάρχουν ήδη κάποια στοιχεία και δεν είναι άλλα από τα Break Points. Αποθηκεύτηκαν στο RSV κατά το προηγούμενο στάδιο, δηλαδή αμέσως μετά την εύρεση του καθενός. Όπως στην εύρεση των Break Points, έτσι και στην περίπτωση αυτή έχουμε έναν

εξωτερικό βρόχο while που επαναλαμβάνεται μέχρι να βρεθεί το RSV. Αυτός περιέχει έναν βρόχο for που επαναλαμβάνεται τόσες φορές όσοι είναι οι H/N του συστήματος. Ο βρόχος αυτός διατρέχει για κάθε H/N i την αντίστοιχη γραμμή του πίνακα primaries, δηλαδή την i -οστή, και για κάθε έναν primary H/N του i ελέγχει αν ο H/N αυτός βρίσκεται στον RSV. Αν βρει κάποιον ηλεκτρονόμο που δεν υπάρχει στο RSV, τότε διακόπτεται η διαδικασία ελέγχου και δεν εξετάζονται οι επόμενοι primaries του ηλεκτρονόμου αυτού. Αρκεί ένας primary να μην υπάρχει ήδη στο διάνυσμα RSV για να μην μπορεί να συμπεριληφθεί σε αυτό και ο υπό εξέταση ηλεκτρονόμος i . Επίσης, αυξάνεται κατά ένα η τιμή μιας μεταβλητής (againRSV) που είχε αρχικοποιηθεί να έχει μηδενική τιμή και ο ρόλος της οποίας θα εξηγηθεί παρακάτω. Η εκτέλεση συνεχίζεται ξαναμπαίνοντας στο βρόχο for, δηλαδή θεωρώντας τον επόμενο ηλεκτρονόμο και εξετάζοντας αν μπορεί αυτός να προστεθεί στο RSV.

Αντίθετα, αν προέκυψε ότι όλοι οι primaries ενός H/N υπάρχουν ήδη στο RSV, ελέγχουμε εάν υπάρχει ήδη ο εν λόγω ηλεκτρονόμος στο RSV και εάν είναι φανταστικός ηλεκτρονόμος. Αν τίποτα από αυτά δεν ισχύει, τον περιλαμβάνουμε στο RSV και συνεχίζουμε με την επόμενη εκτέλεση του βρόχου for, δηλαδή ελέγχοντας τον επόμενο H/N. Όταν τελειώσουν όλοι οι ηλεκτρονόμοι, το εάν θα επαναληφθεί ο εξωτερικός while βρόχος εξαρτάται από την τιμή της μεταβλητής που αναφέρθηκε παραπάνω. Αν η μεταβλητή αυτή έχει τιμή μεγαλύτερη του μηδενός, αυτό σημαίνει ότι υπάρχουν ηλεκτρονόμοι που δε συμπεριλήφθηκαν στο RSV, πρέπει συνεπώς η όλη διαδικασία να επαναληφθεί. Άρα, έχουμε μια ακόμα επανάληψη του βρόχου while.

5.3.3.2 Μέθοδος 2

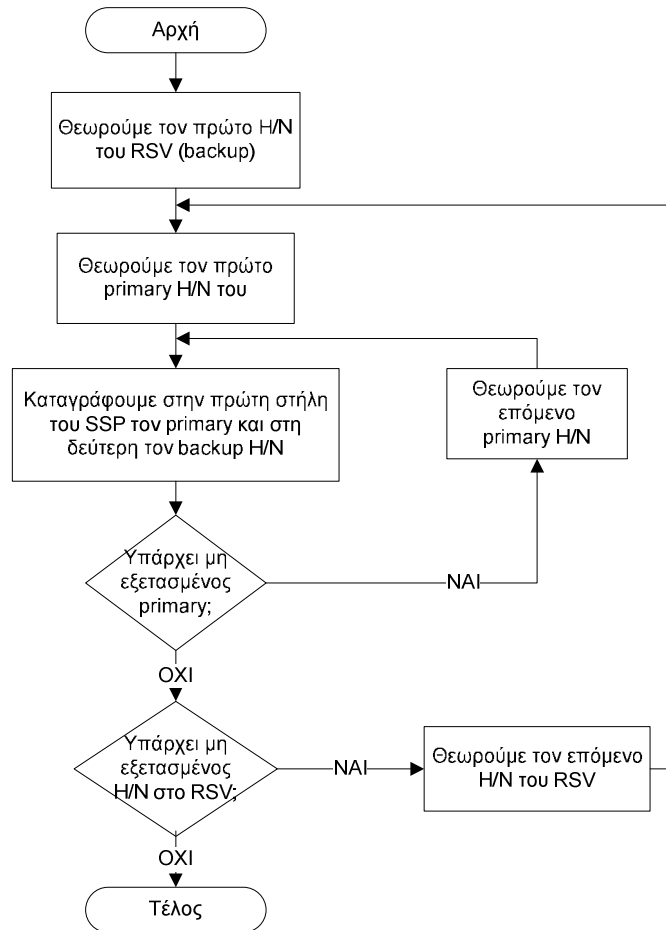
Στην περίπτωση της εύρεσης των Break Points με την πρώτη μέθοδο, η υλοποίηση της διαδικασίας δημιουργίας του διανύσματος RSV είναι πολύ απλή, αφού η αλληλουχία των ηλεκτρονόμων είναι ουσιαστικά έτοιμη. Πρόκειται για την αλληλουχία των στοιχείων του πίνακα RSM, ο οποίος δημιουργήθηκε κατά την εύρεση των Break Points του συστήματος. Αυτό που κάνει το πρόγραμμα στο στάδιο αυτό είναι να διαβάσει πρώτα τα στοιχεία της πρώτης γραμμής και στη συνέχεια τα στοιχεία της δεύτερης γραμμής του πίνακα RSM και να τα αποθηκεύει διαδοχικά στο διάνυσμα RSV. Το πρόγραμμα εκτελεί δηλαδή τα ακόλουθα απλά βήματα:

1. Αποθηκεύει τα στοιχεία της πρώτης γραμμής του RSM στο RSV.
2. Αποθηκεύει στη συνέχεια στην ίδια γραμμή του RSV τα στοιχεία της δεύτερης γραμμής του RSM.

5.3.4 SSP

Το τμήμα αυτό του κυρίως προγράμματος αποσκοπεί στη διάταξη των ζευγών primary- backup με συγκεκριμένη αλληλουχία. Τα ζεύγη αυτά των ηλεκτρονόμων είναι ήδη γνωστά και ουσιαστικά βρίσκονται αποθηκευμένα στον πίνακα primaries που κάθε γραμμή του i περιέχει τους primaries του ηλεκτρονόμου i . Το επιπλέον που θα επιτευχθεί τώρα είναι να διαταχθούν αυτά τα ζεύγη με συγκεκριμένη σειρά και να

δημιουργηθεί το Σύνολο των Διαδοχικών Ζευγών (Set of Sequential Pairs- SSP). Το παρακάτω διάγραμμα ροής εξηγεί τη διαδικασία:



Διάγραμμα 8: Διάγραμμα ροής για τη εύρεση του Συνόλου των Διαδοχικών Ζευγών (SSP)

Το τμήμα αυτό του κώδικα αποτελείται από έναν εξωτερικό βρόχο for, ο οποίος επαναλαμβάνεται τόσες φορές όση είναι η διάσταση του διανύσματος RSV και περιλαμβάνει έναν βρόχο while. Ξεκινάμε θεωρώντας τον πρώτο ηλεκτρονόμο του RSV. Στο while βρόχο θεωρούμε ένα- ένα τα στοιχεία της σειράς του primaries που αντιστοιχεί στον θεωρούμενο ηλεκτρονόμο και καταγράφουμε στην πρώτη στήλη του πίνακα SSP τον primary που διαβάζεται από τον πίνακα primaries και στη δεύτερη στήλη της ίδιας σειράς του SSP τον backup που είναι ο υπό εξέταση ηλεκτρονόμος. Στη συνέχεια μετακινούμαστε στην επόμενη σειρά του SSP και καταγράφουμε σε αυτήν το επόμενο ζευγάρι primary- backup. Εξερχόμαστε από το βρόχο while όταν έχουν εξεταστεί όλοι οι primaries ηλεκτρονόμοι της υπό εξέταση γραμμής του πίνακα primaries. Θεωρούμε τον επόμενο ηλεκτρονόμο του RSV και επαναλαμβάνουμε τη διαδικασία. Ο πίνακας SSP έχει βρεθεί όταν έχουν εξεταστεί όλα τα στοιχεία του διανύσματος RSV.

Μετά το τέλος της τοπολογικής ανάλυσης του συστήματος έχουν δημιουργηθεί οι παρακάτω πίνακες:

$loopsBreakers(i, j)$	Κάθε γραμμή i περιέχει όλους τους ηλεκτρονόμους που σχηματίζουν ένα βρόχο .
$loopsBuses(i, j)$	Κάθε γραμμή i περιέχει όλους τους ζυγούς που σχηματίζουν ένα βρόχο .
$breakpoints(1, j)$	Περιέχει όλους τους ηλεκτρονόμους που αποτελούν Break Points.
$RSV(1, j)$	Περιέχει τη διατεταγμένη ακολουθία των ηλεκτρονόμων.
$SSP(1, j)$	Περιέχει τη διατεταγμένη ακολουθία των ζευγών ηλεκτρονόμων σύμφωνα με την οποία θα ρυθμιστούν οι ηλεκτρονόμοι.

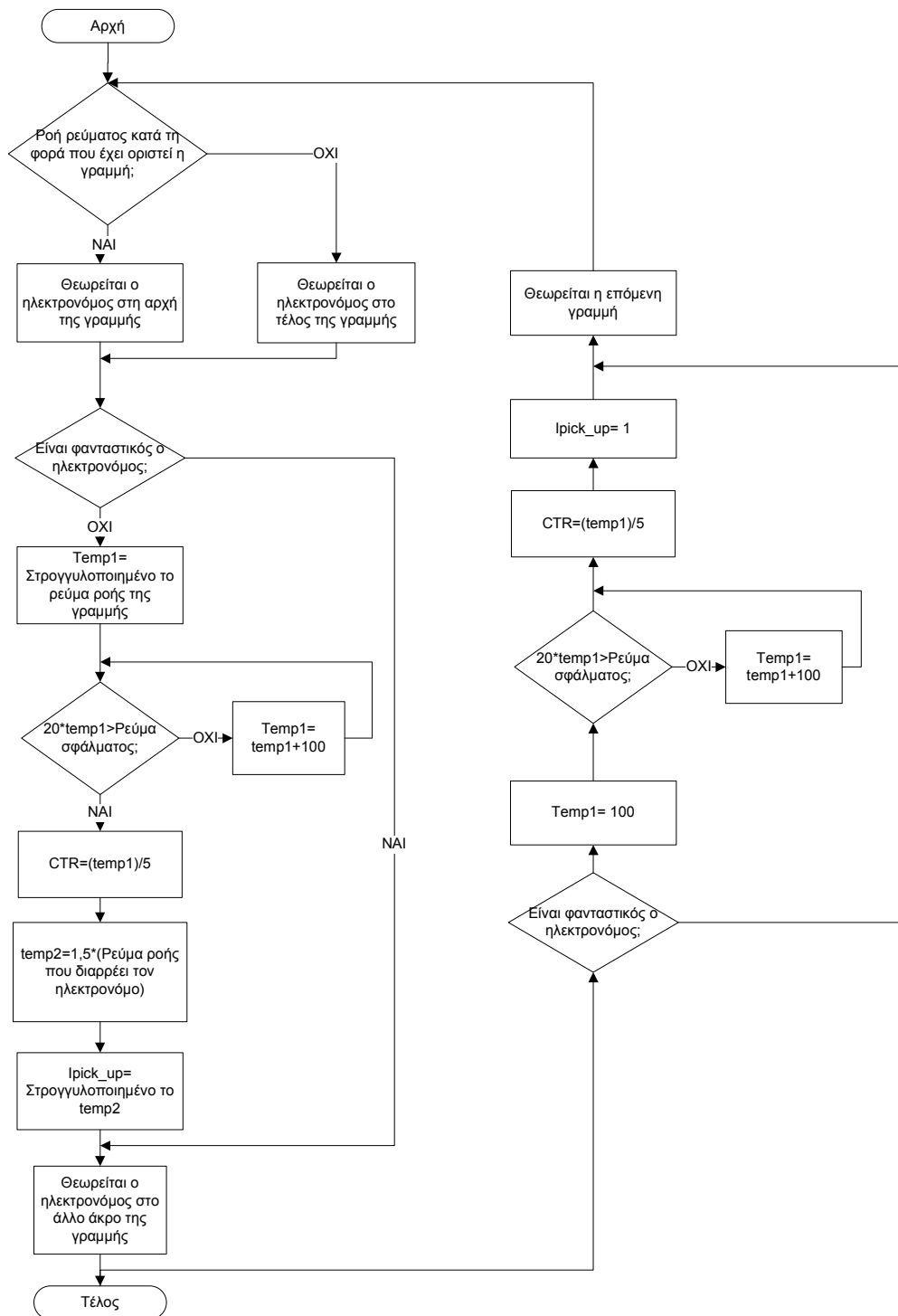
Πίνακας 4 : Δεδομένα που προκύπτουν από την τοπολογική ανάλυση του συστήματος

5.4 Επιλογή - ρύθμιση εξοπλισμού προστασίας

Στη συνέχεια παρουσιάζεται η υλοποίηση της διαδικασίας επιλογής και ρύθμισης των στοιχείων προστασίας, η οποία εξηγήθηκε στο εδάφιο 4.4. Στο σημείο αυτό θα περιγραφούν όχι μόνο οι αντίστοιχοι αλγόριθμοι που αναπτύχθηκαν στη διπλωματική αυτή εργασία, αλλά και η γενική πορεία του προγράμματος σε καθένα από τα επιμέρους τμήματα.

5.4.1 Επιλογή του λόγου των μετασχηματιστών εντάσεως (CTR) και του ρεύματος ρύθμισης των ηλεκτρονόμων ($I_{pick-up}$)

Ακολουθούμε τη διαδικασία που έχει ήδη περιγραφεί και υπολογίζουμε το λόγο μετασχηματισμού (Current Transformer Ratio – CTR) κάθε μετασχηματιστή έντασης και το ρεύμα ρύθμισης κάθε ηλεκτρονόμου ($I_{pick-up}$). Ο υπολογισμός γίνεται με τη βοήθεια ενός επαναληπτικού βρόχου for, εντός του οποίου υπολογίζονται και οι δυο παράμετροι. Ο αλγόριθμος παρουσιάζεται στο διάγραμμα ροής που ακολουθεί και το οποίο επαναλαμβάνεται τόσες φορές όσες είναι και οι γραμμές του συστήματος. Μετά το διάγραμμα ροής εξηγείται αναλυτικά το αντίστοιχο τμήμα του κυρίως προγράμματος.



Διάγραμμα 9: Αλγόριθμος για τον υπολογισμό των CTR των μετασχηματιστών και των $I_{pick-up}$ των ηλεκτρονόμων κάθε γραμμής

Αρχικά αρχικοποιούνται λαμβάνοντας μηδενική τιμή το διάνυσμα CT_Ratio , στο οποίο θα αποθηκευτούν οι λόγοι μετασχηματισμού (CTR) των μετασχηματιστών και το διάνυσμα $I_{pick-up}$, στο οποίο θα αποθηκευτούν τα ρεύματα ρύθμισης ($I_{pick-up}$) των

ηλεκτρονόμων. Και τα δυο αυτά διανύσματα έχουν διαστάσεις $(nRelays \times 1)$, όπου $nRelays$ ο αριθμός των ηλεκτρονόμων. Με τη βοήθεια ενός επαναληπτικού βρόχου `for`, ο κώδικας που υλοποιεί τον αλγόριθμο που περιγράφεται στο παραπάνω διάγραμμα ροής επαναλαμβάνεται $nLines$ φορές, όπου $nLines$ ο αριθμός των γραμμών του συστήματος.

Εντός του βρόχου `for` υπάρχει αρχικά ένας έλεγχος για το εάν η ροή ρεύματος είναι κατά τη φορά που έχει οριστεί η γραμμή. Συγκεκριμένα, για κάθε γραμμή ελέγχουμε στις θέσεις του διανύσματος `S.Bus.Voltages` που αντιστοιχούν στους ζυγούς στα άκρα της γραμμής, ποια από τις δυο αυτές τάσεις είναι μεγαλύτερη. Ροή ρεύματος έχουμε από το ζυγό με τη μεγαλύτερη τάση προς τον άλλο ζυγό. Αν η φορά της ροής ταυτίζεται με τη φορά ορισμού της γραμμής, τότε ρυθμίζουμε τον μετασχηματιστή και τον ηλεκτρονόμο στην αρχή της γραμμής με βάση το ρεύμα κανονικής λειτουργίας και στα στοιχεία στο τέλος της γραμμής δίνουμε την ελάχιστη επιτρεπόμενη τιμή του CTR , ελέγχοντας βέβαια αν παρατηρείται κορεσμός, και την ελάχιστη επιτρεπόμενη τιμή του $I_{pick-up}$ αντίστοιχα. Αν η ροή ρεύματος έχει αντίθετη φορά, η ρύθμιση με βάση το ρεύμα γίνεται στα στοιχεία στο τέλος της γραμμής, ενώ ελάχιστες τιμές λαμβάνουν οι παράμετροι των στοιχείων στην αρχή της γραμμής.

Θεωρούμε ροή ρεύματος κατά τη φορά ορισμού της γραμμής. Αρχικά ελέγχουμε αν ο ηλεκτρονόμος στην αρχή της γραμμής είναι φανταστικός αναζητώντας τον στο διάνυσμα `phantomRelays`. Αν βρεθεί στο διάνυσμα αυτό η διαδικασία συνεχίζει στον επόμενο ηλεκτρονόμο, χωρίς να ρυθμιστούν ο μετασχηματιστής του και αυτός. Μηδενική τιμή στα CTR και $I_{pick-up}$ σημαίνει ότι ο ηλεκτρονόμος είναι φανταστικός, οπότε δεν έχει ρυθμιστεί.

Η ρύθμιση των μετασχηματιστών γίνεται ως εξής: Αν ο ηλεκτρονόμος δεν είναι φανταστικός βρίσκουμε στο διάνυσμα I_flow_relay το ρεύμα κανονικής λειτουργίας της γραμμής. Η προσωρινή μεταβλητή `temp1` λαμβάνει τιμή ίση με την τιμή του ρεύματος αυτού στρογγυλοποιημένο στην εκατοντάδα. Στη συνέχεια ελέγχεται αν η τιμή του `temp` πολλαπλασιαζόμενη επί 20 δίνει τιμή μεγαλύτερη από το ρεύμα που περνά από το πρωτεύον του μετασχηματιστή σε περίπτωση `close-in` βραχυκυκλώματος, το οποίο είναι αποθηκευμένο στη θέση του διανύσματος I_fault_relay που αντιστοιχεί στον εν λόγω ηλεκτρονόμο. Αν ο έλεγχος ικανοποιείται ο λόγος μετασχηματισμού του μετασχηματιστή είναι $CTR = temp1/5$. Διαφορετικά η τιμή `temp` αυξάνεται ανά 100, έως ότου ικανοποιηθεί ο έλεγχος. Όσο για τον ηλεκτρονόμο στο τέλος της γραμμής, αυτός λαμβάνει αρχική τιμή `temp` την ελάχιστη δυνατή τιμή πρωτεύοντος μετασχηματιστή (100), ενώ στη συνέχεια η διαδικασία είναι η ίδια με αυτή για τον ηλεκτρονόμο στην αρχή της γραμμής.

Για το ρεύμα ρύθμισης των ηλεκτρονόμων της γραμμής έχουμε: Διαβάζουμε το ρεύμα κανονικής λειτουργίας που διαρρέει τον ηλεκτρονόμο από τον πίνακα I_flow_relay , το πολλαπλασιάζουμε επί 1,5 και το αποτέλεσμα αποθηκεύεται στη μεταβλητή `temp2`. Αν το δεκαδικό μέρος της `temp2` είναι μικρότερο από 0,5, ως ρεύμα ρύθμισης του ηλεκτρονόμου λαμβάνεται το ακέραιο μέρος της `temp2`. Αν το δεκαδικό μέρος είναι μεγαλύτερο του 0,5, θεωρούμε ρεύμα ρύθμισης με τιμή ίση με (ακέραιο μέρος της `temp2` + 0,5). Ο ηλεκτρονόμος στο τέλος της γραμμής λαμβάνει

τιμή $I_{pick-up}=1$, δηλαδή την ελάχιστη δυνατή, και η ρύθμιση των στοιχείων της γραμμής ολοκληρώνεται.

Για γραμμές με φορά ρεύματος αντίθετη από τη φορά ορισμού της γραμμής, η διαδικασία είναι η ίδια, μόνο που τώρα ο ηλεκτρονόμος στο τέλος της γραμμής ρυθμίζεται με βάση το ρεύμα και ο ηλεκτρονόμος στην αρχή της γραμμής λαμβάνει τις ελάχιστες δυνατές ρυθμίσεις.

5.4.2 Χρονική σταθερά (TDS)

Το πρώτο βήμα για τον υπολογισμό των TDS των ηλεκτρονόμων προστασίας είναι η δημιουργία ενός πίνακα που χρησιμοποιείται για την εύρεση λύσης.

Πίνακας A_{lin}

Πρόκειται για τον πίνακα A_{lin} , ο οποίος αρχικοποιείται μηδενικός και έχει διαστάσεις $(zeugi \times n Relays)$, όπου $zeugi$ είναι ο αριθμός των ζευγών πρωτεύουσας – δευτερεύουσας προστασίας, δηλαδή ο αριθμός των γραμμών του πίνακα SSP . Κάθε γραμμή του πίνακα A_{lin} περιέχει δυο μη μηδενικά στοιχεία, ένα που αναφέρεται στον primary ηλεκτρονόμο και ένα που αναφέρεται στον back-up ηλεκτρονόμο. Αρχικά διαβάζουμε τους ηλεκτρονόμους κάθε ζεύγους από τον πίνακα SSP και τους βρίσκουμε στον πίνακα Numbers. Μας ενδιαφέρουν η στήλη c_{pr} του primary ηλεκτρονόμου και η γραμμή και η στήλη, l και c αντίστοιχα, του back up ηλεκτρονόμου. Ενδιαφερόμαστε δηλαδή για το ζυγό του primary και για το ζυγό και τη γραμμή του back up ηλεκτρονόμου.

Για τον ηλεκτρονόμο πρωτεύουσας προστασίας (pr) επιλέγουμε από τον πίνακα “ I_{fault_relay} ” το στοιχείο $I_{fault_relay}(pr, c_{pr})$, δηλαδή το ρεύμα που περνάει από τη γραμμή στο σημείο που βρίσκεται ο ηλεκτρονόμος για close-in σφάλμα και διαιρούμε το ρεύμα αυτό με το στοιχείο $CT_Ratio(pr, l)$ του πίνακα “ CT_Ratio ” για να βρούμε το ρεύμα στο δευτερεύον, δηλαδή το ρεύμα που διαρρέει τον ηλεκτρονόμο για close-in σφάλμα. Το αποτέλεσμα διαιρείται με το ρεύμα ρύθμισης του εν λόγω ηλεκτρονόμου που είναι αποθηκευμένο στον πίνακα “ I_{pick_up} ”, δηλαδή με $I_{pick_up}(pr, l)$, και έτσι λαμβάνουμε την ποσότητα M για τον ηλεκτρονόμο. Τέλος, από τη σχέση $a = \frac{k_1}{M^{k_2} - 1}$ υπολογίζουμε το a και το αποθηκεύουμε στη θέση $A_{lin}(line, pr)$ του πίνακα A_{lin} , όπου $line$ η γραμμή του ζεύγους στον SSP .

Για τον ηλεκτρονόμο δευτερεύουσας προστασίας (bu) η διαδικασία είναι παρόμοια, μόνο που δεν αναφερόμαστε σε σφάλμα αμέσως μετά τον back-up, αλλά σε σφάλμα αμέσως μετά τον primary, δηλαδή το σφάλμα που θεωρήθηκε προηγουμένως. Θεωρούμε το στοιχείο $I_{fault_line}(l, c_{pr})$, δηλαδή το ρεύμα που περνάει από τη γραμμή του δευτερεύοντος ηλεκτρονόμου για close-in σφάλμα στον πρωτεύοντα

ηλεκτρονόμο, και το πολλαπλασιάζουμε επί $I_{base}(bu,1)$, αφού τα στοιχεία του πίνακα I_{fault_line} είναι σε α.μ. Δε χρησιμοποιούμε στην περίπτωση αυτή τον πίνακα I_{fault_relay} , καθώς ο πίνακας αυτός περιέχει για κάθε ηλεκτρονόμο το ρεύμα που διαρρέει το πρωτεύον για close-in σε αυτόν σφάλμα. Εδώ δεν πρόκειται για close-in στον δευτερεύοντα ηλεκτρονόμο σφάλμα. Διαιρούμε το αποτέλεσμα που προέκυψε με το λόγο μετασχηματισμού $CT_Ratio(bu,1)$ για να βρούμε το ρεύμα που διαρρέει τον ηλεκτρονόμο. Το αποτέλεσμα διαιρείται με το ρεύμα ρύθμισης του εν λόγω ηλεκτρονόμου, δηλαδή με $I_{pick_up}(bu,1)$, και έτσι λαμβάνουμε την ποσότητα

M. Χρησιμοποιούμε τη σχέση $a = \frac{k_1}{M^{k_2} - 1}$ και αποθηκεύουμε το $(-a)$ στη θέση

$A_lin(line, bu)$ του πίνακα A_lin . Ο αλγόριθμος της δημιουργίας του πίνακα A_lin περιγράφεται από το διάγραμμα ροής που δίνεται στο Διάγραμμα 10.

Για τη δημιουργία του πίνακα A_lin δεν απαιτείται να ελέγχουμε αν κάθε ηλεκτρονόμος είναι φανταστικός, αφού οι ηλεκτρονόμοι διαβάζονται από τον πίνακα SSP , όπου υπάρχουν μόνο μη φανταστικοί ηλεκτρονόμοι.

Τελικά στον πίνακα A_lin περιέχονται οι ποσότητες a ή $(-a)$ όλων των ηλεκτρονόμων και από εδώ θα διαβάζονται για να χρησιμοποιηθούν στις παρακάτω δυο μεθοδολογίες. Θεωρώντας τη χαρακτηριστική του ηλεκτρονόμου

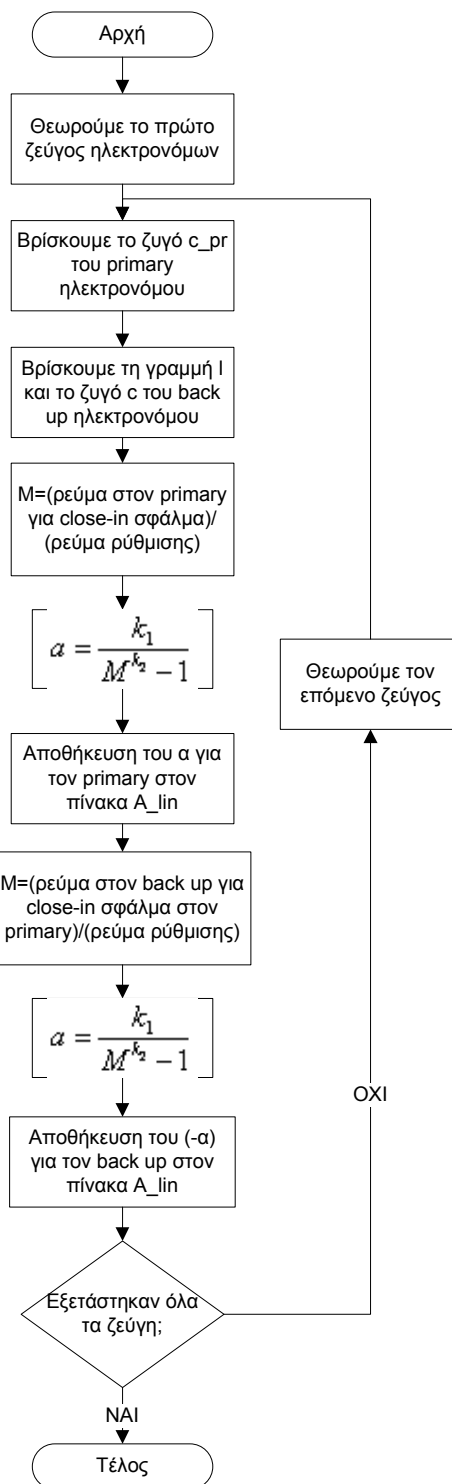
$$T = \frac{k_1}{M^{k_2} - 1} \cdot TDS = a \cdot TDS$$

έχουμε για τον πίνακα A_lin :

Κάθε γραμμή περιέχει το a του πρωτεύοντα ηλεκτρονόμου του ζεύγους A_lin στη στήλη που αντιστοιχεί στο ζυγό του ηλεκτρονόμου αυτού και το $(-a)$ του δευτερεύοντα ηλεκτρονόμου του ζεύγους στη στήλη που αντιστοιχεί στο ζυγό του ηλεκτρονόμου αυτού.

Πίνακας 5 : Πίνακας A_lin

Η δημιουργία του πίνακα αυτού περιγράφεται με το ακόλουθο διάγραμμα ροής:



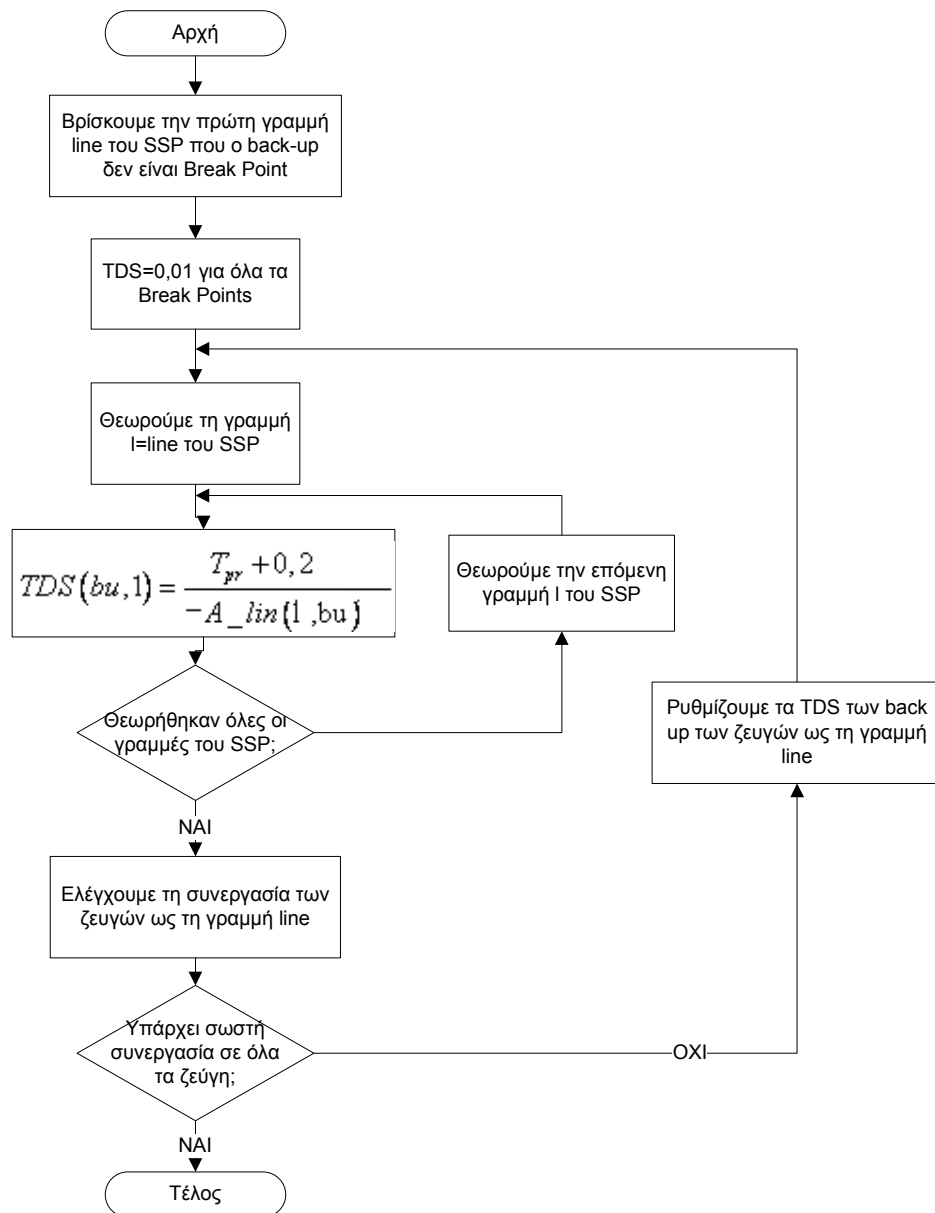
Διάγραμμα 10 : Αλγόριθμος δημιουργίας του πίνακα A_{lin}

Ο υπολογισμός των χρονικών σταθερών TDS των ηλεκτρονόμων θα γίνει με δυο εναλλακτικούς τρόπους. Γι' αυτό κατά τη εκτέλεση εμφανίζεται στην οθόνη ένα μήνυμα που καλεί το χρήστη να επιλέξει ποια μέθοδος επιθυμεί να χρησιμοποιηθεί. Στη συνέχεια δίνονται τιμές στις σταθερές k_1 και k_2 της χαρακτηριστικής του

ηλεκτρονόμου. Επιλέχθηκε να χρησιμοποιηθούν ηλεκτρονόμοι απλώς αντιστρόφου χρόνου, γι' αυτό οι σταθερές λαμβάνουν τιμές $k_1 = 0,14$ και $k_2 = 0,02$. Το πρόγραμμα συνεχίζει με την εκτέλεση της επιλεγμένης μεθόδου. Στη συνέχεια περιγράφεται αναλυτικά η υλοποίηση των δυο αυτών μεθόδων.

5.4.2.1 Μέθοδος τοπολογικής ανάλυσης

Με τη βοήθεια ενός διαγράμματος ροής παρουσιάζουμε τον αλγόριθμο της μεθόδου αυτής, τον οποίο εξηγήσαμε συνοπτικά στην παράγραφο 4.4.2.2.1. Εν συνεχεία, εξηγείται το τμήμα του προγράμματος που υλοποιεί τον αλγόριθμο αυτό.



Διάγραμμα 11 : Αλγόριθμος της πρώτης μεθόδου εύρεσης των TDS των ηλεκτρονόμων

Αναλυτικά μπορούμε να πούμε τα ακόλουθα: Αρχικοποιούμε το διάνυσμα TDS , διάστασης $(nRelays \times 1)$, μηδενίζοντας το. Στη συνέχεια βρίσκουμε την πρώτη γραμμή, έστω $line$, του SSP που ο ηλεκτρονόμος δευτερεύουσας προστασίας δεν είναι Break Point. Αυτό γίνεται ελέγχοντας για κάθε ζεύγος $primary - back up$ του SSP αν ο $back up$ ηλεκτρονόμος υπάρχει στο διάνυσμα $phantomRelays$ των φανταστικών ηλεκτρονόμων. Δίνουμε την ελάχιστη επιτρεπόμενη τιμή του TDS στους ηλεκτρονόμους που είναι Break Points. Δηλαδή $TDS(i,1) = 0,01$ για όλες τις θέσεις i του διανύσματος TDS που αντιστοιχούν σε Break Points.

Ένας επαναληπτικός βρόχος $while$ που ακολουθεί διατρέχει συνεχώς το διάνυσμα TDS ανανεώνοντας τον, έως ότου τα στοιχεία του οδηγούν στην επιθυμητή συνεργασία των ηλεκτρονόμων. Εντός του βρόχου, ρυθμίζονται αρχικά οι ηλεκτρονόμοι που δεν είναι Break Points. Αν βρισκόμαστε στην πρώτη επανάληψη, υπολογίζουμε για κάθε ζεύγος ηλεκτρονόμων, που βρίσκεται στη γραμμή i του SSP αρχίζοντας από τη γραμμή $line$ που βρέθηκε προηγουμένως, το χρόνο λειτουργίας του $primary$ από τη σχέση $T_{pr} = A_lin(i,pr) \cdot TDS(pr,1)$. Η μεταβλητή $TDS(pr,1)$ έχει ήδη οριστεί, αφού για κάθε ζεύγος ο $primary$ ηλεκτρονόμος ορίζεται πριν από τον $back up$. Ύστερα απαιτούμε

$$T_{bu} = T_{pr} + CTI \Leftrightarrow TDS(bu,1) = \frac{T_{pr} + 0,2}{-A_lin(bu)} \quad (5.1)$$

και από τη σχέση αυτή βρίσκουμε το TDS του $back up$ ηλεκτρονόμου και το αποθηκεύουμε στην αντίστοιχη θέση του διανύσματος TDS . Αν πρόκειται για μια από τις επόμενες επαναλήψεις του $while$, υπολογίζουμε για κάθε ζεύγος ηλεκτρονόμων το χρόνο λειτουργίας του $primary$ από τη σχέση $T_{pr} = A_lin(i,pr) \cdot TDS(pr,1)$ και του $back up$ από τη σχέση $T_{bu} = -A_lin(i,bu) \cdot TDS(bu,1)$. Ελέγχουμε αν οι χρόνοι αυτοί διαφέρουν λιγότερο από το χρονικό περιθώριο συνεργασίας $CTI = 0,2sec$ και αν συμβαίνει αυτό αλλάζουμε το $SSP TDS$ του $back up$ ηλεκτρονόμου, έτσι ώστε οι χρόνοι να διαφέρουν $0,2sec$.

Μετά την ανανέωση των TDS των ηλεκτρονόμων που δεν είναι Break Points, θα πρέπει να επιστρέψουμε στην αρχή του πίνακα SSP και να ελέγξουμε αν τα ζεύγη στα οποία ο $back up$ ηλεκτρονόμος είναι Break Point συνεργάζονται σωστά. Αν σε κάποιο ζεύγος υπάρχει πρόβλημα, ανανεώνεται η τιμή $TDS(bu,1)$ χρησιμοποιώντας τον τύπο (5.1). Παράλληλα, κάθε φορά που αλλάζει το TDS κάποιου Break Point, αυξάνεται κατά ένα η τιμή μιας αρχικά μηδενικής μεταβλητής ($change$). Αν η μεταβλητή αυτή έχει μη μηδενική τιμή μετά τον έλεγχο όλων των ζευγών με $back up$ ηλεκτρονόμο ένα Break Point, αυτό σημαίνει ότι έγιναν κάποιες αλλαγές στο διάνυσμα TDS . Θα πρέπει λοιπόν να ξαναδιασχίσουμε τον πίνακα SSP και να ελέγξουμε τη συνεργασία με τις νέες τιμές των TDS . Έξοδο από το βρόχο $while$ έχουμε όταν $change > 0$. Οι τιμές του διανύσματος είναι τότε οι τελικές τιμές των TDS .

5.4.2.2 Μέθοδος γραμμικού προγραμματισμού

Ο δεύτερος τρόπος υπολογισμού των TDS είναι με χρήση της μεθόδου γραμμικού προγραμματισμού. Με τη βοήθεια ενός διαγράμματος ροής παρουσιάζουμε τον αλγόριθμο της μεθόδου αυτής, ο οποίος βασίζεται στις σχέσεις (4.12) της παραγράφου 4.4.2.2.2. Επαναλαμβάνουμε για λόγους συνοχής τις σχέσεις αυτές:

$$\begin{aligned} \min & \left(J = \sum_{i=1}^n TDS_i \right) \\ & T_{ji} - T_{ii} \geq CTI \\ & TDS_{\min} \leq TDS_i \leq TDS_{\max} \end{aligned} \quad (4.12)$$

Για την επίλυση με γραμμικό προγραμματισμό χρησιμοποιείται η εντολή “linprog” του Matlab. Πρόκειται για μια έτοιμη συνάρτηση του προγράμματος, η οποία επιλύει προβλήματα της μορφής

$$\begin{aligned} \min & (J = f \cdot x) \\ & A \cdot x \leq b \\ & Aeq \cdot x = beq \\ & lb \leq x \leq ub \end{aligned}$$

δίνοντας ως αποτέλεσμα το διάνυσμα x . Συντάσσεται ως εξής:

$$x = \text{linprog}(f, A, b, Aeq, beq, lb, ub)$$

Το πρόβλημα γραμμικού προγραμματισμού που επιθυμούμε να επιλύσουμε αποκτάει τη μορφή που απαιτεί η συνάρτηση του Matlab αν η πρώτη ομάδα περιορισμών θεωρηθεί με αντίθετη φορά. Για το λόγο αυτό πολλαπλασιάζουμε επί (-1) τις ανισότητες αυτές. Επίσης παρατηρούμε ότι στο πρόβλημα εύρεσης των TDS δεν υπάρχουν περιορισμοί που εκφράζονται με ισότητα. Για το λόγο αυτό, ο πίνακας Aeq και το διάνυσμα beq που απαιτούνται από τη διατύπωση της συνάρτησης θεωρούνται στην περίπτωσή μας κενοί. Οι αντιστοιχίες που ισχύουν μεταξύ των πινάκων που χρησιμοποιούνται από τη συνάρτηση και των πινάκων του προβλήματός μας είναι:

x	Το διάνυσμα TDS διάστασης $(n \text{ Relays} \times 1)$,
f	Μοναδιαίο διάνυσμα διάστασης $(1 \times n \text{ Relays})$,
A	ο πίνακας A_{lin} διάστασης $(zeugi \times n \text{ Relays})$,
b	διάνυσμα διάστασης $(zeugi \times 1)$ με στοιχεία ίσα με $(-CTI)$,
Aeq	κενός πίνακας,
beq	κενό διάνυσμα,
lb	διάνυσμα διάστασης $(n \text{ Relays} \times 1)$ με στοιχεία ίσα με 0,01,
ub	διάνυσμα διάστασης $(n \text{ Relays} \times 1)$ με στοιχεία ίσα με 1,1.

Πίνακας 6 : Στοιχεία της συνάρτησης επίλυσης γραμμικού προγραμματισμού

Παρατηρούμε ότι κάποιος από τους πίνακες που χρειαζόμαστε έχουν ήδη δημιουργηθεί, ενώ άλλοι θα πρέπει να οριστούν σε αυτό το στάδιο. Ακολουθούν κάποια σχόλια για τους χρησιμοποιούμενους πίνακες και ο ορισμός όσων δεν έχουν ήδη οριστεί.

Διάνυσμα f

Το διάνυσμα αυτό χρησιμοποιείται για να δημιουργήσει την αντικειμενική συνάρτηση πολλαπλασιαζόμενο με το διάνυσμα των αγνώστων. Πρόκειται για διάνυσμα διάστασης ίσης με τον αριθμό των ηλεκτρονόμων, που τα στοιχεία του έχουν τιμή ίση με τη μονάδα. Εξαιρούνται οι θέσεις του διανύσματος που αντιστοιχούν στους φανταστικούς ηλεκτρονόμους. Δεν επιθυμούμε να ρυθμίσουμε τους ηλεκτρονόμους αυτούς, αφού δεν έχουν φυσική υπόσταση. Γι' αυτό το λόγο οι αντίστοιχες θέσεις στο διάνυσμα f έχουν μηδενική τιμή. Επίσης οι ηλεκτρονόμοι αυτοί δεν εμφανίζονται στους περιορισμούς. Αναφέρουμε ότι το διάνυσμα f περιέχει μοναδιαία στοιχεία, καθώς χρησιμοποιούμε τη συγκεκριμένη διαμόρφωση του προβλήματος, στην οποία η αντικειμενική συνάρτηση είναι το άθροισμα των TDS και όχι των χρόνων λειτουργίας των ηλεκτρονόμων.

Πίνακας A και διάνυσμα b

Πρόκειται για στοιχεία που χρησιμοποιούνται για να δηλωθούν οι περιορισμοί που σχετίζονται με τη συνεργασία πρωτεύουσας-δευτερεύουσας προστασίας. Ο πίνακας A είναι ο γνωστός πίνακας A_{lin} , ο οποίος ορίστηκε στην αρχή της παραγράφου 5.4.2 και έχει διάσταση ($zeugi \times n Relays$). Το διάνυσμα b έχει διάσταση ίση με τον αριθμό των ζευγών πρωτεύουσας-δευτερεύουσας προστασίας. Αποτελείται από στοιχεία με τιμή ίση με το αντίθετο του χρονικού περιθωρίου που ισχύει σε κάθε ζεύγος προστασίας, δηλαδή ($-CTI$). Οι πίνακες αυτοί είναι κατάλληλοι για να εκφράσουν τους περιορισμούς

$$T_{bu} - T_{pr} \geq CTI \Leftrightarrow -T_{bu} + T_{pr} \leq -CTI$$

αφού το γινόμενο κάθε γραμμής του πίνακα A_{lin} επί το διάνυσμα των αγνώστων TDS δίνει ως αποτέλεσμα $-a_{bu} \cdot TDS_{bu} + a_{pr} TDS_{pr} = -T_{bu} + T_{pr}$, το οποίο θα πρέπει να είναι μικρότερο ή ίσο με το χρονικό περιθώριο. Αφού το πλήθος των γραμμών του πίνακα A_{lin} ισούται με τον αριθμό των ζευγών, τόσοι είναι συνολικά και οι περιορισμοί που προκύπτουν.

Πίνακας A_{eq} και διάνυσμα beq

Όπως αναφέρθηκε και προηγουμένως, τα στοιχεία αυτά χρησιμοποιούνται για να εκφράσουν περιορισμούς με μορφή ισότητας. Στην περίπτωσή μας είναι κενοί, αφού δεν υπάρχουν τέτοιοι περιορισμοί, η χρησιμοποίησή τους όμως επιβάλλεται από τη σύνταξη της συνάρτησης linprog.

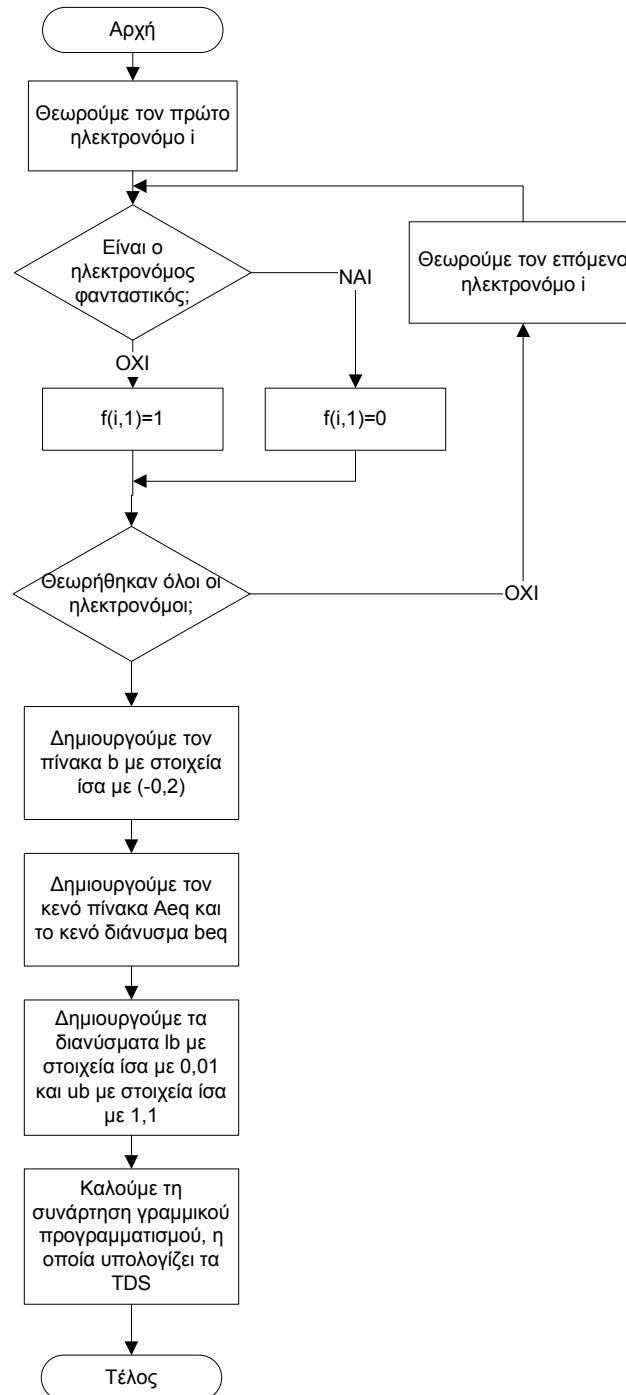
Διανύσματα lb και ub

Πρόκειται για διανύσματα διάστασης ($n Relays \times 1$), τα οποία χρησιμοποιούνται για να εκφραστούν οι περιορισμοί οι σχετικοί με τα επιτρεπόμενα όρια του TDS . Τα διανύσματα εκφράζουν το κάτω και το πάνω όριο των περιορισμών, οι οποίοι έχουν την ακόλουθη μορφή:

$$0,01 \leq TDS \leq 1,1$$

Συνεπώς τα στοιχεία των διανυσμάτων lb και ub έχουν τιμή ίση με 0,01 και 1,1 αντίστοιχα.

Όσα εξηγήθηκαν παραπάνω παρουσιάζονται συνοπτικά με το ακόλουθο διάγραμμα ροής:



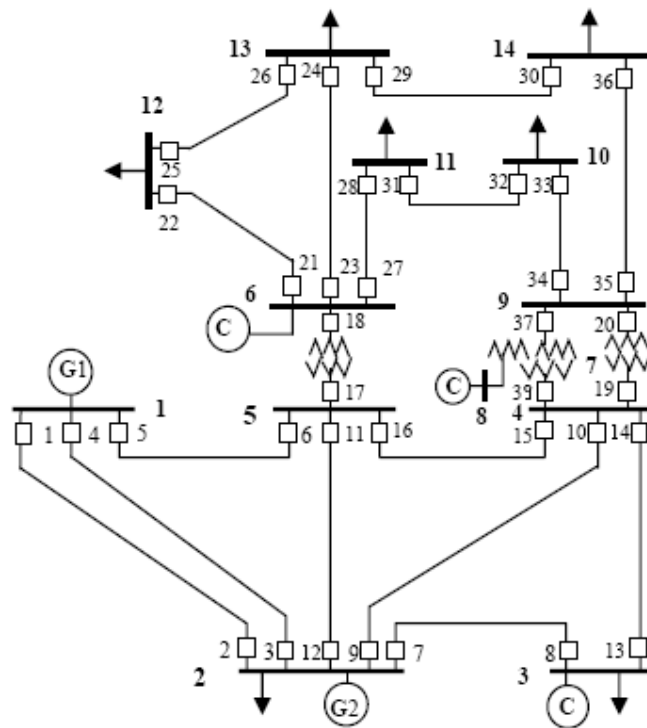
Διάγραμμα 12 : Αλγόριθμος της δεύτερης μεθόδου εύρεσης των TDS των ηλεκτρονόμων

6

Εφαρμογή

Εφαρμόζουμε τη μέθοδο ρύθμισης των ηλεκτρονόμων υπερέντασης, που αναλύθηκε στα δυο προηγούμενα κεφάλαια, στο IEEE 14-ζυγών σύστημα δοκιμής, το οποίο απεικονίζεται στο Σχήμα 11. Το σύστημα αυτό αποτελείται από 14 ζυγούς και 21 κλάδους. Έχουμε παραγωγή ενεργού ισχύος στους ζυγούς 1 και 2 και άεργου ισχύος στους ζυγούς 3, 6 και 8. Όσον αφορά τα φορτία, έχουμε κατανάλωση ενεργού ισχύος σε όλους τους ζυγούς εκτός από τους 1, 7, 8 και κατανάλωση άεργου ισχύος σε όλους τους ζυγούς εκτός από τους 1, 4, 7 και 8. Στο σύστημα υπάρχουν επίσης δυο μετασχηματιστές, μεταξύ των ζυγών 5-6 και 4-9, και ένας μετασχηματιστής τριών τυλιγμάτων με άκρα στους ζυγούς 4, 9 και 8.

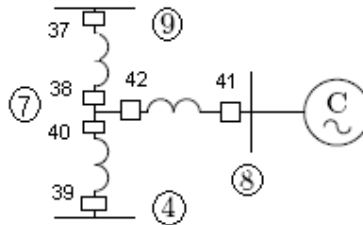
Ακολουθούμε τη σειρά με την οποία ορίζονται οι γραμμές στο αντίστοιχο αρχείο εισόδου και τοποθετούμε ηλεκτρονόμους στην αρχή και το τέλος κάθε γραμμής, διαδικασία που υλοποιείται με τη δημιουργία το πίνακα Numbers. Στο Σχήμα 8 φαίνεται επίσης η αρίθμηση των ηλεκτρονόμων.



Σχήμα 11 : IEEE 14-ζυγών σύστημα δοκιμής

Όπως αναφέρθηκε ήδη, το σύστημα περιλαμβάνει έναν μετασχηματιστή τριών τυλιγμάτων με άκρα τους ζυγούς 4, 9 και 8. Το ισοδύναμο του μετασχηματιστή αυτού φαίνεται σε μεγέθυνση στο σχήμα που ακολουθεί. Θεωρείται ένας επιπλέον φανταστικός ζυγός, ο 7, στον οποίο συνδέονται και τα τρία τυλίγματα του μετασχηματιστή. Κατά τη δήλωση των γραμμών στο αρχείο εισόδου καταγράφονται στο συγκεκριμένο τμήμα του συστήματος τρεις γραμμές, οι 7-4, 7-9 και 8-7. Έτσι,

κατά την τοποθέτηση και αρίθμηση των ηλεκτρονόμων, στις γραμμές αυτές τοποθετούνται 6 ηλεκτρονόμοι με την αρίθμηση που φαίνεται στο σχήμα. Όμως, από τους ηλεκτρονόμους αυτούς μόνο οι 37 και 39 υφίστανται στην πραγματικότητα, ενώ οι υπόλοιποι (38, 40, 41, 42) είναι φανταστικοί ηλεκτρονόμοι (phantom relays). Στη συνέχεια θα λαμβάνονται υπόψη, γιατί βοηθούν στον καθορισμό των ζευγών πρωτεύουσας και δευτερεύουσας προστασίας, δεν πρόκειται όμως οι ηλεκτρονόμοι αυτοί τελικά να λάβουν ρυθμίσεις. Γι' αυτό και δεν εμφανίζονται στο Σχήμα 11.



Σχήμα 12 : Μετασχηματιστής τριών τυλιγμάτων

6.1 Ανάλυση ροών φορτίου και ανάλυση βραχυκυκλωμάτων

Τα πρώτα στοιχεία που υπολογίζονται είναι τα ρεύματα κανονικής λειτουργίας που ρέουν στους κλάδους του συστήματος. Τα αρχεία εισόδου που χρησιμοποιούν οι μέθοδοι αυτοί κατασκευάζονται από τα στοιχεία που υπάρχουν στα [4], [13], [14] και δίνονται στο παράρτημα. Αρχικά το αποτέλεσμα για κάθε κλάδο είναι μιγαδικό και εκφρασμένο στο ανά μονάδα σύστημα. Στη συνέχεια υπολογίζεται για κάθε ηλεκτρονόμο το ρεύμα κανονικής λειτουργίας σε A που ρέει στο πρωτεύον του μετασχηματιστή του και η τιμή αποθηκεύεται στο διάνυσμα I_flow_relay . Το διάνυσμα αυτό στην περίπτωσή μας έχει διάσταση (42×1) .

Το κυρίως πρόγραμμα συνεχίζει καλώντας τη συνάρτηση `shortcircuits`. Η συνάρτηση αυτή υπολογίζει το ρεύμα σε κάθε κλάδο του συστήματος για σφάλμα σε καθέναν από τους ζυγούς. Εν προκειμένω, η συνάρτηση θα εκτελέσει τον υπολογισμό βραχυκυκλωμάτων 14 φορές και θα επιστρέψει έναν πίνακα I_fault_line διαστάσεων (21×14) , όπου το στοιχείο x_{ij} του πίνακα ισούται με το α.μ. ρεύμα που διαρρέει τη γραμμή i για σφάλμα στο ζυγό j . Στη συνέχεια δημιουργείται και το διάνυσμα I_fault_relay , όπου αποθηκεύεται το ρεύμα σε A στο πρωτεύον κάθε ηλεκτρονόμου για close-in σε αυτόν σφάλμα. Το διάνυσμα αυτό έχει διάσταση (42×1) .

Στον πίνακα που ακολουθεί δίνονται τα περιεχόμενα του διανύσματος I_flow_relay και I_fault_relay , δηλαδή για κάθε ηλεκτρονόμο το ρεύμα κανονικής λειτουργίας και το ρεύμα για close-in σφάλμα.

Ηλεκτρονόμος	$I_{flow_relay} (A)$	$I_{fault_relay} (A)$
1	361.95	2804.85
2	361.95	3656.39
3	358.77	3702.98
4	358.77	2760.16
5	212.95	3725.20
6	212.95	3283.60
7	305.60	3649.71
8	305.60	2783.81
9	255.51	3772.99
10	255.51	3285.72
11	212.12	3196.85
12	212.12	3776.99
13	84.66	2855.66
14	84.66	3211.02
15	212.13	2757.04
16	212.13	2195.12
17	181.32	3229.71
18	906.61	12554.73
19	67.58	3739.67
20	337.89	8602.58
21	160.72	17323.44
22	160.72	4514.67
23	376.04	16314.97
24	376.04	4945.02
25	36.92	6147.23
26	36.92	11094.11
27	166.77	15045.04
28	166.77	5993.42
29	120.08	10379.36
30	120.08	6812.27
31	91.04	6942.75
32	91.04	11655.37
33	127.42	3565.94
34	127.42	8516.88
35	200.92	9239.12
36	200.92	4158.10
37	584.08	5325.98
38	584.08	10333.66
39	127.61	3344.40
40	638.04	12556.29
41	407.93	13467.21
42	312.75	7276.50

Πίνακας 7 : Ρεύμα στο πρωτεύον για κανονική λειτουργία και close-in σφάλμα

6.2 Τοπολογική ανάλυση συστήματος

Το υποπρόγραμμα τοπολογικής ανάλυσης δίνει ως πρώτο αποτέλεσμα όλους τους βρόχους του συστήματος. Στη συγκεκριμένη εφαρμογή προκύπτουν 112 βρόχοι, που καθένας τους περιλαμβάνει από 2 έως 11 ηλεκτρονόμους. Ένα μέρος του πίνακα που περιέχει τους βρόχους με αναφορά στους αντίστοιχους ηλεκτρονόμους είναι:

1	3								
1	7	13	15	6					
1	7	13	19	34	32	28	18	6	
1	7	13	19	35	30	24	18	6	
1	7	13	19	35	30	26	22	18	6
1	7	13	39	38	34	32	28	18	6
1	7	13	39	38	35	30	24	18	6
1	7	13	39	38	35	30	26	22	18
1	9	15	6						
1	9	19	34	32	28	18	6		
1	9	19	35	30	24	18	6		
1	9	19	35	30	26	22	18	6	
1	9	39	38	34	32	28	18	6	
1	9	39	38	35	30	24	18	6	
1	9	39	38	35	30	26	22	18	6
1	12	6							
4	2								
4	7	13	15	6					
4	7	13	19	34	32	28	18	6	
4	7	13	19	35	30	24	18	6	
4	7	13	19	35	30	26	22	18	6
4	7	13	39	38	34	32	28	18	6
4	7	13	39	38	35	30	24	18	6
4	7	13	39	38	35	30	26	22	18
4	9	15	6						
4	9	19	34	32	28	18	6		
4	9	19	35	30	24	18	6		
4	9	19	35	30	26	22	18	6	
4	9	39	38	34	32	28	18	6	
4	9	39	38	35	30	24	18	6	
4	9	39	38	35	30	26	22	18	6
4	12	6							
5	11	2							
5	11	3							
5	16	10	2						
5	16	10	3						
5	16	14	8	2					
5	16	14	8	3					
5	17	21	25	29	36	20	10	2	
5	17	21	25	29	36	20	10	3	

Πίνακας 8 : Μερικοί από τους βρόχους του συστήματος

Στη συνέχεια, το υποπρόγραμμα προχωρεί στον υπολογισμό των Break Points. Με τη μέθοδο εύρεσής τους από τους βρόχους προκύπτουν 16 Break Points, ενώ με τη μέθοδο εύρεσής τους από τους πρωτεύοντες ηλεκτρονόμους προκύπτουν 18 Break Points. Ο πίνακας breakpoints που δημιουργείται από καθεμία από τις μεθόδους αυτές είναι:

<i>Break Points 1 (προκύπτουν με τη μέθοδο 1)</i>															
6	11	16	17	24	26	29	2	3	7	9	12	20	34	35	37

<i>Break Points 2 (προκύπτουν με τη μέθοδο 2)</i>																	
1	4	8	10	11	9	13	17	19	22	24	28	2	3	5	12	16	20

Πίνακας 9 : Οι ηλεκτρονόμοι που είναι Break Points

Για κάθε μια από τις παραπάνω μεθόδους, χρησιμοποιώντας τους πίνακες breakpoints και primaries (ο οποίος περιέχει για κάθε ηλεκτρονόμο τους ηλεκτρονόμους στη ζώνη των οποίων ο εν λόγω παρέχει δευτερεύουσα προστασία), δημιουργείται το διάνυσμα *RSV*. Δεν περιέχονται σε αυτό οι φανταστικοί ηλεκτρονόμοι, συνεπώς η διάστασή του είναι (38×1):

RSV 1 (προκύπτει με τη μέθοδο 1)

6 11 16 17 24 26 29 2 3 7 9 12 20 34 35 37 1 4 5 8 10 14 15 18 19 23 25 30
33 36 39 13 21 28 31 32 27 22

RSV 2 (προκύπτει με τη μέθοδο 2)

1 4 8 10 11 9 13 17 19 22 24 28 2 3 5 12 16 20 6 14 7 15 37 26 30 35 32 34
18 33 36 39 29 31 23 25 27 21

Πίνακας 10 : Το Διάνυσμα Σχετικής Ακολουθίας (RSV)

Από το διάνυσμα αυτό και τον πίνακα primaries προκύπτει εύκολα ο πίνακας SSP, ο οποίος περιέχει όλα τα ζεύγη πρωτεύουσας – δευτερεύουσας προστασίας. Τα ζεύγη αυτά είναι 98 για το σύστημα των 14 ζυγών. Τμήμα του πίνακα αυτού για καθένα από τα σύνολα Break Points είναι:

<i>SSP 1 (χρησιμοποιώντας το RSV 1)</i>	
<i>primary</i>	<i>back-up</i>
1	6
4	6
2	11
3	11
7	11
9	11
10	16
14	16

<i>SSP 2 (χρησιμοποιώντας το RSV 2)</i>	
<i>primary</i>	<i>back-up</i>
3	1
7	1
9	1
12	1
2	4
7	4
9	4
12	4

<i>SSP 1</i> (χρησιμοποιώντας το RSV 1)	
19	16
39	16
21	17
23	17
27	17
18	24
21	24
27	24
22	26
36	29
4	2
5	2
1	3
5	3
13	7
14	9
15	9
19	9
39	9
6	12
16	12
17	12
10	20
14	20
15	20
39	20
32	34
30	35
10	37
14	37
15	37
19	37
3	1
7	1
9	1
12	1
2	4
7	4
9	4
12	4
11	5
16	5
17	5
2	8
3	8
9	8
12	8
2	10
3	10
7	10
12	10
8	14
6	15
11	15
17	15

<i>SSP 2</i> (χρησιμοποιώντας το RSV 2)	
2	8
3	8
9	8
12	8
2	10
3	10
7	10
12	10
2	11
3	11
7	11
9	11
14	9
15	9
19	9
39	9
10	13
15	13
19	13
39	13
21	17
23	17
27	17
34	19
35	19
37	19
18	22
23	22
27	22
18	24
21	24
27	24
18	28
21	28
23	28
4	2
5	2
1	3
5	3
11	5
16	5
17	5
6	12
16	12
17	12
10	16
14	16
19	16
39	16
10	20
14	20
15	20
39	20
1	6
4	6

<i>SSP 1</i> (χρησιμοποιώντας το RSV 1)	
6	18
11	18
16	18
34	19
35	19
37	19
26	23
29	23
24	25
29	25
24	30
26	30
20	33
35	33
37	33
20	36
34	36
37	36
20	39
34	39
35	39
10	13
15	13
19	13
39	13
25	21
18	28
21	28
23	28
33	31
28	32
31	27
18	22
23	22
27	22

<i>SSP 2</i> (χρησιμοποιώντας το RSV 2)	
8	14
13	7
6	15
11	15
17	15
10	37
14	37
15	37
19	37
22	26
24	30
26	30
30	35
28	32
32	34
6	18
11	18
16	18
20	33
35	33
37	33
20	36
34	36
37	36
20	39
34	39
35	39
36	29
33	31
26	23
29	23
24	25
29	25
31	27
25	21

Πίνακας 11 : Τμήμα του Σύνολο Διαδοχικών Ζευγών (SSP)

Πρόκειται για τα ίδια ζεύγη πρωτεύουσας – δευτερεύουσας προστασίας μόνο που η σειρά με την οποία εμφανίζονται στον πίνακα *SSP* είναι διαφορετική.

6.3 Επιλογή - ρύθμιση εξοπλισμού προστασίας

6.3.1 Επιλογή του λόγου των μετασχηματιστών εντάσεως (*CTR*) και του ρεύματος ρύθμισης των ηλεκτρονόμων ($I_{pick-up}$)

Τα δυο αυτά μεγέθη υπολογίζονται με τη διαδικασία που έχει ήδη περιγραφεί. Τα αποτελέσματα αποθηκεύονται στους πίνακες *CT_Ratio* και *Ipick_up*, οι οποίοι για το σύστημα των 14 ζυγών έχουν την ακόλουθη μορφή:

Ηλεκτρονόμος	CT_Ratio	$I_{pick_up} (A)$
1	80	6.5
2	40	1
3	40	1
4	80	6.5
5	60	5
6	40	1
7	80	5.5
8	40	1
9	60	6
10	40	1
11	40	1
12	60	5
13	40	1
14	40	3
15	40	1
16	60	5
17	40	1
18	200	6.5
19	40	1
20	100	5
21	180	1
22	60	1
23	180	3
24	60	1
25	80	1
26	120	1
27	160	1.5
28	60	1
29	120	1.5
30	80	1
31	80	1.5
32	120	1
33	40	1
34	100	1.5
35	100	3
36	60	1
37	60	1
38	0	0
39	40	1
40	0	0
41	0	0
42	0	0

Πίνακας 12 : Λόγος μετασχηματισμού CT_Ratio κάθε μετασχηματιστή έντασης και ρεύμα ρύθμισης I_{pick_up} κάθε ηλεκτρονόμου

Σχετικά με τους λόγους μετασχηματισμού, οι τιμές που καταγράφονται στον παραπάνω πίνακα είναι το αποτέλεσμα της διαίρεσης της ονομαστικής τιμής του πρωτεύοντος με την ονομαστική τιμή του δευτερεύοντος, που είναι $5A$. Οι ηλεκτρονόμοι που έχουν μηδενική τιμή στο λόγο μετασχηματισμού και στο ρεύμα

ρύθμισης είναι οι φανταστικοί ηλεκτρονόμοι. Οι ηλεκτρονόμοι αυτοί δεν υφίστανται στην πραγματικότητα, συνεπώς δε χρειάζεται να ρυθμιστούν. Οι μηδενικές τιμές είναι ένας τρόπος να δηλωθεί ότι οι ηλεκτρονόμοι αυτοί δε ρυθμίζονται.

6.3.2 Χρονική σταθερά (TDS)

Ο υπολογισμός της χρονικής σταθεράς είναι ο κυριότερος από τους υπολογισμούς που περιλαμβάνει όλη η διαδικασία. Παρακάτω δίνονται τα αποτελέσματα που προκύπτουν για το TDS κάθε ηλεκτρονόμου από κάθε μια από τις χρησιμοποιούμενες μεθόδους.

6.3.2.1 Μέθοδος τοπολογικής ανάλυσης

Ο πίνακας TDS που προκύπτει με εφαρμογή αυτής της μεθόδου είναι:

Ηλεκτρονόμος	TDS
1	0.0124
2	0.2756
3	0.2783
4	0.01
5	0.1224
6	0.0854
7	0.1254
8	0.2439
9	0.1283
10	0.223
11	0.2222
12	0.1402
13	0.5079
14	0.1795
15	0.5791
16	0.2143
17	0.5437
18	0.1415
19	0.4764
20	0.2001
21	0.5506
22	0.2899
23	0.4582
24	0.5222
25	0.5531
26	0.3492
27	0.5771
28	0.6221
29	0.5696
30	0.5241
31	0.5801
32	0.6409

<i>Ηλεκτρονόμος</i>	<i>TDS</i>
33	0.6423
34	0.7383
35	0.4554
36	0.6511
37	0.5504
38	0
39	0.5248
40	0
41	0
42	0

Πίνακας 13 : Χρονική σταθερά (*TDS*) κάθε ηλεκτρονόμου με χρήση της μεθόδου τοπολογικής ανάλυσης

Οι παραπάνω τιμές της σταθεράς *TDS* προκύπτουν είτε εφαρμοστεί η μέθοδος προσδιορισμού των Break Points με χρήση των βρόχων (Break Points 1) είτε η μέθοδος προσδιορισμού με χρήση των πρωτευόντων ηλεκτρονόμων (Break Points 2). Δηλαδή το αποτέλεσμα είναι ακριβώς το ίδιο όποια μέθοδος εύρεσης Break Points, συνεπώς οποιαδήποτε από της δυο αλληλουχίες ρύθμισης που ορίζουν οι πίνακες SSP, και αν έχει χρησιμοποιηθεί.

6.3.2.2 Μέθοδος γραμμικού προγραμματισμού

Αν αντί της προηγούμενης μεθόδου υπολογισμού των *TDS* χρησιμοποιηθεί η μέθοδος του γραμμικού προγραμματισμού, ο πίνακας που προκύπτει είναι:

<i>Ηλεκτρονόμος</i>	<i>TDS</i>
1	0.0124
2	0.2756
3	0.2783
4	0.01
5	0.1224
6	0.0854
7	0.1254
8	0.2439
9	0.1283
10	0.223
11	0.2222
12	0.1402
13	0.5079
14	0.1795
15	0.5791
16	0.2143
17	0.5437
18	0.1415
19	0.4764
20	0.2001
21	0.5506
22	0.2899
23	0.4582

<i>Ηλεκτρονόμος</i>	<i>TDS</i>
24	0.5222
25	0.5531
26	0.3492
27	0.5771
28	0.6221
29	0.5696
30	0.5241
31	0.5801
32	0.6409
33	0.6423
34	0.7383
35	0.4554
36	0.6511
37	0.5504
38	0
39	0.5248
40	0
41	0
42	0

Πίνακας 14 : Χρονική σταθερά (*TDS*) κάθε ηλεκτρονόμου με χρήση της μεθόδου γραμμικού προγραμματισμού

Και στην περίπτωση των *TDS* μηδενική ρύθμιση σημαίνει απλά ότι ο ηλεκτρονόμος είναι φανταστικός και δε ρυθμίζεται. Η μηδενική τιμή δεν ανταποκρίνεται σε πραγματική ρύθμιση, αλλά είναι μια ένδειξη μη ρύθμισης του ηλεκτρονόμου.

Παρατηρούμε ότι προκύπτουν ακριβώς ίδιες τιμές με αυτές που προέκυψαν με την προηγούμενη μέθοδο. Επίσης, αν εφαρμοστεί η μέθοδος γραμμικού προγραμματισμού για τον υπολογισμό των *TDS*, τα παραπάνω αποτελέσματα δεν εξαρτώνται προφανώς από τα Break Points, οπότε είναι τα ίδια είτε εφαρμοστεί η πρώτη μέθοδος προσδιορισμού των Break Points είτε η δεύτερη. Οποιοσδήποτε λοιπόν από τους τέσσερις συνδυασμούς μεθόδων οδηγεί στην ίδια λύση για τη χρονική σταθερά των ηλεκτρονόμων.

7

Αξιολόγηση – Σύγκριση αποτελεσμάτων

Στο κεφάλαιο αυτό παρατίθενται σχολιασμοί και συμπεράσματα σχετικά με τις χρησιμοποιούμενες τεχνικές και τα αποτελέσματα που προέκυψαν από την εφαρμογή τους. Πραγματοποιούνται επίσης οι απαραίτητες συγκρίσεις μεταξύ των διαφορετικών αποτελεσμάτων που προέκυψαν από τη χρήση εναλλακτικών μεθόδων, όπου αυτό πραγματοποιήθηκε.

7.1 Αξιολόγηση αποτελεσμάτων

Τα αποτελέσματα που προέκυψαν σχετικά με τις παραμέτρους ρύθμισης των ηλεκτρονόμων και των αντίστοιχων μετασχηματιστών τους είναι αρκετά ικανοποιητικά. Τόσο οι λόγοι μετασχηματισμού CT_Ratio των μετασχηματιστών έντασης, όσο και τα ρεύματα ρύθμισης I_{pick_up} και η χρονικές σταθερές TDS των ηλεκτρονόμων υπερέντασης λαμβάνουν τιμές που βρίσκονται εντός των επιτρεπτών ορίων και ανταποκρίνονται στις πραγματικές συνθήκες.

Όσον αφορά συγκεκριμένα τις χρονικές σταθερές των ηλεκτρονόμων, οι τιμές τους είναι ικανοποιητικά μικρές και δεν προσεγγίζουν το άνω επιτρεπτό όριο. Μάλιστα, σε κάποιους ηλεκτρονόμους η σταθερά TDS λαμβάνει την ελάχιστη δυνατή τιμή ή τιμή κοντά στη ελάχιστη δυνατή (ηλεκτρονόμοι 1, 4, 6). Η επιθυμία μας να λάβουν οι σταθερές TDS των ηλεκτρονόμων όσο το δυνατόν μικρότερες τιμές εξηγείται από το γεγονός ότι για μικρότερες τιμές των TDS έχουμε και μικρότερους χρόνους λειτουργίας των ηλεκτρονόμων. Αυτό είναι εμφανές από τη μορφή της χαρακτηριστικής των ηλεκτρονόμων υπερέντασης:

$$T = \frac{k_1}{M^{k_2} - 1} \cdot TDS$$

Οι μικροί χρόνοι λειτουργίας των ηλεκτρονόμων επιδιώκονται, καθώς σημαίνουν γρήγορη ανταπόκριση των μέσων προστασίας σε περίπτωση εκδήλωσης βραχυκυκλώματος στο σύστημα. Προϋπόθεση είναι βέβαια ότι θα τηρούνται οι περιορισμοί που επιβάλλονται από τη συνεργασία των ηλεκτρονόμων.

Για να ελέγξουμε ότι οι περιορισμοί αυτοί ικανοποιούνται, δηλαδή ότι για κάθε ζεύγος πρωτεύουσας – δευτερεύουσας προστασίας ο ηλεκτρονόμος δευτερεύουσας προστασίας ανταποκρίνεται στο σφάλμα τουλάχιστον 0,2sec μετά τον ηλεκτρονόμο πρωτεύουσας προστασίας, ελέγχουμε τους χρόνους λειτουργίας των ηλεκτρονόμων. Για κάθε γραμμή i του SSP, δηλαδή για κάθε ζεύγος πρωτεύουσας – δευτερεύουσας προστασίας, έχουμε:

$$pr=SSP(i,1) \quad bu=SSP(i,2)$$

Και οι χρόνοι λειτουργίας των ηλεκτρονόμων κάθε ζεύγους είναι:

$$t_{pr} = A_{lin}(i,pr) \cdot TDS(pr,1)$$

$$t_{bu} = A_{lin}(i,bu) \cdot TDS(bu,1)$$

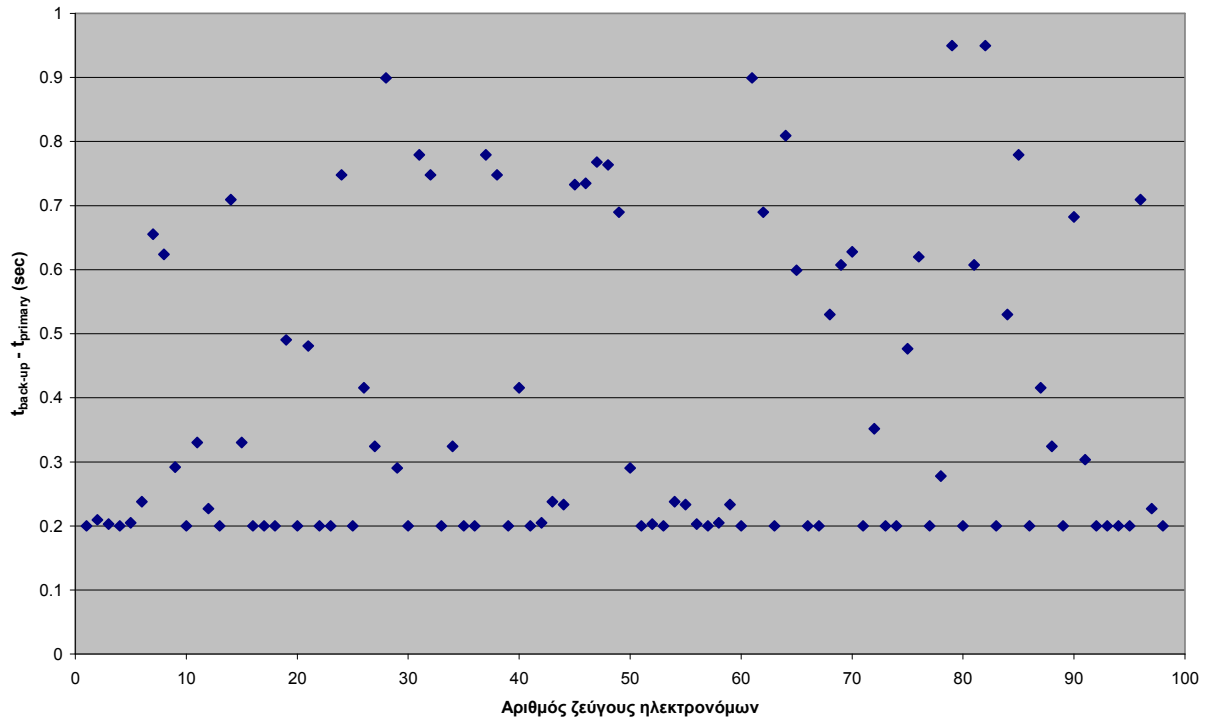
Υπολογίζουμε τους χρόνους αυτούς και τους αποθηκεύουμε σε έναν πίνακα. Μετά την τοποθέτηση όλων των ζευγών ο πίνακας αυτός είναι:

<i>H/N πρωτεύουσας</i> <i>προστασίας (pr)</i>	<i>Χρόνος λειτουργίας</i> (sec)	<i>H/N δευτερεύουσας</i> <i>προστασίας (bu)</i>	<i>Χρόνος λειτουργίας</i> (sec)
1	0.0508	6	0.2508
4	0.0412	6	0.2508
2	0.4083	11	0.6111
3	0.4111	11	0.6111
7	0.4063	11	0.6111
9	0.3733	11	0.6111
10	0.3386	16	0.9938
14	0.3698	16	0.9938
19	0.7021	16	0.9938
39	0.7938	16	0.9938
21	0.8061	17	1.1363
23	0.9094	17	1.1363
27	0.9363	17	1.1363
18	0.4271	24	1.1363
21	0.8061	24	1.1363
27	0.9363	24	1.1363
22	0.4497	26	0.6497
36	1.0303	29	1.2303
4	0.0412	2	0.5315
5	0.3315	2	0.5315
1	0.0508	3	0.5315
5	0.3315	3	0.5315
13	0.7979	7	0.9979
14	0.3698	9	1.1177
15	0.9177	9	1.1177
19	0.7021	9	1.1177
39	0.7938	9	1.1177
6	0.1297	12	1.0292
16	0.7389	12	1.0292
17	0.8292	12	1.0292
10	0.3386	20	1.1177
14	0.3698	20	1.1177
15	0.9177	20	1.1177
39	0.7938	20	1.1177
32	0.9363	34	1.1363
30	0.7893	35	0.9893
10	0.3386	37	1.1177
14	0.3698	37	1.1177
15	0.9177	37	1.1177
19	0.7021	37	1.1177
3	0.4111	1	0.6111
7	0.4063	1	0.6111
9	0.3733	1	0.6111
12	0.3777	1	0.6111
2	0.4083	4	1.1412

<i>H/N πρωτεύουσας προστασίας</i>	<i>Χρόνος λειτουργίας (sec)</i>	<i>H/N δευτερεύουσας προστασίας</i>	<i>Χρόνος λειτουργίας (sec)</i>
7	0.4063	4	1.1412
9	0.3733	4	1.1412
12	0.3777	4	1.1412
11	0.3397	5	1.0292
16	0.7389	5	1.0292
17	0.8292	5	1.0292
2	0.4083	8	0.6111
3	0.4111	8	0.6111
9	0.3733	8	0.6111
12	0.3777	8	0.6111
2	0.4083	10	0.6111
3	0.4111	10	0.6111
7	0.4063	10	0.6111
12	0.3777	10	0.6111
8	0.3855	14	0.5855
6	0.1297	15	1.0292
11	0.3397	15	1.0292
17	0.8292	15	1.0292
6	0.1297	18	0.9389
11	0.3397	18	0.9389
16	0.7389	18	0.9389
34	1.2284	19	1.4284
35	0.8985	19	1.4284
37	0.8209	19	1.4284
26	0.5159	23	1.144
29	0.944	23	1.144
24	0.7925	25	1.144
29	0.944	25	1.144
24	0.7925	30	0.9925
26	0.5159	30	0.9925
20	0.4785	33	1.0985
35	0.8985	33	1.0985
37	0.8209	33	1.0985
20	0.4785	36	1.4284
34	1.2284	36	1.4284
37	0.8209	36	1.4284
20	0.4785	39	1.4284
34	1.2284	39	1.4284
35	0.8985	39	1.4284
10	0.3386	13	1.1177
15	0.9177	13	1.1177
19	0.7021	13	1.1177
39	0.7938	13	1.1177
25	0.8536	21	1.0536
18	0.4271	28	1.1094
21	0.8061	28	1.1094
23	0.9094	28	1.1094
33	0.9571	31	1.1571
28	0.9029	32	1.1029
31	0.9606	27	1.1606
18	0.4271	22	1.1363
23	0.9094	22	1.1363
27	0.9363	22	1.1363

Πίνακας 15 : Χρόνοι λειτουργίας πρωτεύουσας και δευτερεύουσας προστασίας

Ελέγχοντας ένα – ένα τα ζεύγη ηλεκτρονόμων διαπιστώνουμε ότι όντως ο χρόνος λειτουργίας του ηλεκτρονόμου δευτερεύουσας προστασίας είναι τουλάχιστον 0,2sec μεγαλύτερος από το χρόνο λειτουργίας του ηλεκτρονόμου πρωτεύουσας προστασίας. Δηλαδή οι περιορισμοί που σχετίζονται με τη συνεργασία των ηλεκτρονόμων πληρούνται. Σε πολλά ζεύγη μάλιστα η διαφορά των δυο χρόνων είναι ακριβώς 0,2sec, οδηγώντας έτσι στην ταχύτερη δυνατή λειτουργία του ηλεκτρονόμου δευτερεύουσας προστασίας σε περίπτωση που αποτύχει η λειτουργία του πρωτεύοντος ηλεκτρονόμου. Το Γράφημα 2 παρουσιάζει με παραστατικό τρόπο τη διαφορά των χρόνων λειτουργίας των ηλεκτρονόμων κάθε ζεύγους.



Γράφημα 2 : Διαφορά χρόνων λειτουργίας των ηλεκτρονόμων κάθε ζεύγους

Στο γράφημα αυτό φαίνεται εμφανώς ότι οι διαφορές στους χρόνους λειτουργίας των ηλεκτρονόμων κάθε ζεύγους είναι μικρές και κυμαίνονται από 0,2 έως 0,95sec. Σχεδόν όλα τα ζεύγη έχουν διαφορά χρόνων λειτουργίας μικρότερη από 0,8sec, ενώ υπάρχουν αρκετά ζεύγη στα οποία ο δευτερεύων ηλεκτρονόμος ανταποκρίνεται 0,2sec μετά τον πρωτεύον. Όλα αυτά σημαίνουν ότι οι ηλεκτρονόμοι δευτερεύουσας προστασίας ανταποκρίνονται, σε περίπτωση αποτυχίας λειτουργίας της πρωτεύουσας προστασίας, ικανοποιητικά γρήγορα, οδηγώντας έτσι σύντομα σε απομόνωση της περιοχής όπου εκδηλώθηκε το σφάλμα.

7.2 Σύγκριση εναλλακτικών μεθόδων

Στην παράγραφο αυτή πρόκειται να συγκριθούν τα αποτελέσματα που προέκυψαν ύστερα από τη χρησιμοποίηση εναλλακτικών μεθόδων. Περισσότερες από μια μέθοδοι χρησιμοποιήθηκαν για την εύρεση των Break Points και του διάνυσματος RSV και για τον υπολογισμό της χρονικής σταθεράς *TDS* κάθε ηλεκτρονόμου.

7.2.1 Για εύρεση των *Break Points* , *RSV*

Συγκρίνοντας τις δυο μεθόδους προσδιορισμού των Break Points του συστήματος, δηλαδή τη μέθοδο εύρεσής τους με χρήση των βρόχων και τη μέθοδο εύρεσής τους με χρήση αποκλειστικά των πρωτευόντων ηλεκτρονόμων, μπορούμε να κάνουμε τα ακόλουθα σχόλια: Η πρώτη μέθοδος οδηγεί σε μικρότερο πλήθος Break Points, και αφού σκοπός είναι να βρεθεί το ελάχιστο σύνολο Break Points, η μέθοδος αυτή μπορεί να θεωρηθεί καλύτερη. Η αλληλουχία των ηλεκτρονόμων στο διάνυσμα RSV και των ζευγών στον πίνακα SSP διαφέρουν επίσης ανάλογα με τη μέθοδο που χρησιμοποιείται, αλλά αυτό είναι αναμενόμενο από τη στιγμή που διαφέρουν τα Break Points των δυο μεθόδων.

Παρ' όλες όμως αυτές τις αλλαγές, το τελικό αποτέλεσμα που είναι η ρύθμιση των ηλεκτρονόμων δε διαφέρει όποια από τις δυο μεθόδους εύρεσης των Break Points και αν έχει χρησιμοποιηθεί, όπως θα φανεί αμέσως μετά.

7.2.2 Για εύρεση των *TDS*

7.2.2.1 Επίδραση των *Break Points*

Κατ' αρχήν επιθυμούμε να βρούμε τι επίδραση έχει το σύνολο των Break Points και το διάνυσμα RSV στις τιμές των *TDS*. Θεωρούμε συνεπώς ότι εφαρμόζεται η μέθοδος τοπολογικής ανάλυσης για τον προσδιορισμό των χρονικών σταθερών *TDS* των ηλεκτρονόμων, αφού μόνο στη μέθοδο αυτή χρησιμοποιούνται τα Break Points και το RSV. Συγκρίνουμε τις τιμές των *TDS* που προκύπτουν με χρήση της μεθόδου αυτής στις ακόλουθες δυο περιπτώσεις: εύρεση των Break Points με τη πρώτη μέθοδο και εύρεση των Break Points με τη δεύτερη μέθοδο. Τα αποτελέσματα που προκύπτουν για τις χρονικές σταθερές των ηλεκτρονόμων είναι ακριβώς τα ίδια είτε έχει χρησιμοποιηθεί η πρώτη είτε η δεύτερη μέθοδος. Η διαφοροποίηση δηλαδή των αποτελεσμάτων των δυο μεθόδων όσον αφορά τα Break Points και το RSV δε συνεπάγεται διαφοροποίηση και στις χρονικές σταθερές των ηλεκτρονόμων.

7.2.2.2 Επίδραση της μεθόδου εύρεσης των *TDS*

Στη συνέχεια συγκρίνουμε τα αποτελέσματα που προέκυψαν για τη χρονική σταθερά των ηλεκτρονόμων με χρήση της μεθόδου τοπολογικής ανάλυσης και της μεθόδου γραμμικού προγραμματισμού. Όπως αναφέρθηκε και κατά την παράθεση των αποτελεσμάτων, οι τιμές των *TDS* των ηλεκτρονόμων είναι ακριβώς οι ίδιες όποια από τις δυο μεθόδους και αν εφαρμοστεί. Θα ανέμενε κανείς ότι η μέθοδος τοπολογικής ανάλυσης θα οδηγούσε σε λιγότερο καλά αποτελέσματα, δηλαδή σε μεγαλύτερες τιμές των χρονικών σταθερών. Αυτό υποστηρίζεται, επειδή όπως

αναφέρθηκε στην παράγραφο 3.4.3 η μέθοδος τοπολογικής ανάλυσης δεν εγγυάται ότι επιλέγονται οι βέλτιστες, δηλαδή οι ελάχιστες δυνατές, ρυθμίσεις για τους ηλεκτρονόμους. Στο σύστημα των 14 ζυγών όμως δεν παρατηρήθηκε να μειονεκτεί η μέθοδος αυτή έναντι της μεθόδου γραμμικού προγραμματισμού, καθώς προέκυψαν οι ίδιες ρυθμίσεις. Ενδεχομένως, σε πιο περίπλοκα συστήματα να είναι εμφανής η υπεροχή της μιας μεθόδου έναντι της άλλης.

7.3 Συμπεράσματα

Στη διπλωματική αυτή εργασία πραγματοποιήθηκε ρύθμιση των ηλεκτρονόμων υπερέντασης με στοιχεία κατεύθυνσης, κάνοντας χρήση εναλλακτικών μεθόδων. Ο χρήστης έχει συνεπώς τη δυνατότητα να επιλέξει ποια από τις μεθόδους επιθυμεί να χρησιμοποιηθεί σε κάθε εφαρμογή. Επίσης, δίνεται η δυνατότητα σύγκρισης των αποτελεσμάτων ύστερα από εκτέλεση όλων των εναλλακτικών μεθόδων.

Τα αποτελέσματα που προκύπτουν από την εκτέλεση του προτεινόμενου προγράμματος είναι, όπως διαπιστώθηκε και στην εφαρμογή, λογικά και ικανοποιητικά. Εξασφαλίζεται δηλαδή η απαραίτητη συνεργασία των ηλεκτρονόμων βελτιστοποιώντας τις ρυθμίσεις των στοιχείων προστασίας. Βέβαια, ενδεχομένως να υπάρχουν περιθώρια περαιτέρω βελτίωσης των ρυθμίσεων.

Τα πλεονεκτήματα που προσφέρει η χρησιμοποιούμενη μεθοδολογία είναι λοιπόν η επικοινωνία με το χρήστη και η δυνατότητα σύγκρισης διαφορετικών μεθόδων. Επίσης, θα πρέπει να συμπεριληφθούν τα πλεονεκτήματα της βελτιωμένης διαμόρφωσης του προβλήματος γραμμικού προγραμματισμού, που είναι η μείωση της διάστασής του προβλήματος και συνεπώς η μείωση των απαιτήσεων σε μνήμη και η επιτάχυνση της διαδικασίας.

8

Βιβλιογραφία

- [1] Μ.Π. Παπαδόπουλος, «Προστασία Συστημάτων Ηλεκτρικής Ενέργειας», Ε.Μ. Πολυτεχνείο, Αθήνα, 1997, σελ 1-3.
- [2] Β.Κ. Παπαδιάς, «Ανάλυση Συστήματος Ηλεκτρικής Ενέργειας- Μόνιμη Κατάσταση Λειτουργίας», Ε.Μ. Πολυτεχνείο, Αθήνα, 1985, σελ 128-135, 149-159.
- [3] Β.Κ. Παπαδιάς, «Ανάλυση Συστήματος Ηλεκτρικής Ενέργειας- Ασύμμετρες και Μεταβατικές Καταστάσεις», Ε.Μ. Πολυτεχνείο, Αθήνα, 1993, σελ 1-2, 41-43.
- [4] Γ. Κορρές, «Θεωρία Προστασίας Υπερέντασης», Ε.Μ. Πολυτεχνείο, Αθήνα, 2005, σελ 1-2.
- [5] F.L. Alvarado, «Simulation and Analysis of Power Systems and Power Electronics in MATLAB», Montreal, pp 3-6, 12-17, 2001.
- [6] H.E.A Talaat, A.Y. Abdelaziz, A.I. Nasseir, A.A. Hajjar, «Optimal Coordination of Overcurrent Relays using Linear Programming», Ain Shams Univerity, Tishreen University, pp 1-4, 5-8.
- [7] H. Zeineldin, E. F. El-Saadany, M. A. Salama, «Optimal Coordination of Directional Overcurrent Relays Coordination», Power Engineering Society General Meeting, 2005, IEEE, vol. 2, pp 1101-1103, June 2005
- [8] H.A. Abyaneh, M. Dabbagh, H.K Hossein, Kazemi Karegar, S.H.H Sadeghi, R.A.J Khan, «A New Optimal Approach for Coordination of Overcurrent Relays in Interconnected Power Systems», IEEE Transactions on Power Delivery, vol. 18, No 2, pp 430-431, April 2003.
- [9] L.G. Perez, A.J. Urdaneta, «Optimal Coordination of Directional Overcurrent Relays Considering Definite Time Backup Relaying», », IEEE Transactions on Power Delivery, vol. 14, No 4, p 1277, October 1999.
- [10] P. Bertrand, «Directional protection equipment», Groupe Schneider, No 181, p 4.
- [11] R.K. Gajbhiye, R. Helwade, S.A. Soman, «A Simple and Efficient Approach to Determination of Minimum Set of Break Point Relays for Transmission Protection System Coordination», International Conference on Future Power Systems, pp 1-3, November 2005.
- [12] Q. Yue, F. Lu, W. Yu, J. Wang, «A Novel Algorithm to Determine Minimum Break Point Set for Optimum Cooperation of Directional Protection Relays in Multiloop Networks», IEEE Transactions on Power Delivery, vol. 21, No 3, pp 1114-1116, July 2006.
- [13] S.K.M Kodsí, C.A. Canizares, «Modeling and Simulation of IEEE 14 BUS System with Facts Controllers», Technical Report #2003-3, pp 37-40.
- [14] Federico Milano, «Power System Analysis Toolbox Documentation for PSAT», version 1.3.4, pp 390-392, July 2005.
- [15] F.L. Alvarado, «Short Circuit Applications of Matlab», p 9, 1999, 2002.
- [16] «Overcurrent Protection for Phase and Earth Faults», Network Protection and Automation Guide, Chapter 9, pp 126-127, 130-131.
- [17] Jun Zhu, «Analysis of Transmission System Faults in the Phase Domain», Texas A&M University, pp 19-29, 50-55, 63-84, August 2004.

- [18] R. Ramaswami, M.J Damborg, S.S Venkata, A.K Jampala, J. Postforoosh, «Enhanced Algorithms for Transmission Protective Relay Coordination», IEEE Transactions on Power Delivery, vol PWRD-1, No 1, pp 280-284, January 1986.

Παράρτημα

Στη συνέχεια παρατίθενται οι κώδικες που χρησιμοποιήθηκαν για την πραγματοποίηση των υπολογισμών που περιλαμβάνει η διπλωματική αυτή εργασία. Δίνονται επίσης τα αρχεία εισόδου για το σύστημα των 14 ζυγών, στο οποίο εφαρμόστηκαν αυτοί οι κώδικες.

Προγράμματα

Το κυρίως πρόγραμμα, το οποίο καλεί άλλα υποπρογράμματα και πραγματοποιεί στο τέλος τη ρύθμιση των ηλεκτρονόμων είναι:

```
%-----Analusi rown fortiou
S=readcf;
S.Ybus=bldybus(S);      %Dimiourgia tou pinaka agwgimotitwn Y-bus
S.snet=
sparse(S.Machine.BusRef,1,S.Machine.MW+j*S.Machine.MVAR,S.Bus.n,1)...
-sparse(S.Load.BusRef,1,S.Load.MW+j*S.Load.MVAR,S.Bus.n,1);
S=pfsolve(S);
if S.mismatch>0.001,
    disp('H sugklisi apetuxe');
else
    disp('Olokliwthike i analusi rown fortiou');
end;
%Ypologismos twn reumatwn kai twn uperfortisewn
[S.Flow,S.iOvld]=flows(S);

%-----Analusi vraxukuklwmatwn
[nLines,one]=size(S.Branch.To);
nNodes=max(max(S.Branch.To),max(S.Branch.From));
buildingc;
[I_fault,I_fault_line,Vnodes]=shortcircuit(S.Branch.From,S.Branch.To,A);
disp('Ypologistikan ta 3ph vraxukuklwmatata');

%-----Dimiourgia tou pinaka Numbers[nLines,nNodes]
k=1;
for i=1:nLines
    for c=1:nNodes
        if A(i,c)==1
            Numbers(i,c)=k+1;
        end;
        if A(i,c)==(-1)
            Numbers(i,c)=k;
        end;
    end;
    k=k+2;
end;
nRelays=max(max(Numbers(:,1:nNodes)));

%-----Euresi twn phantom relays (periptwsi metasximatisti 3 tuligmatwn)
[mach_numb,y]=size(S.Machine.BusRef);
t=1;
c=1;
times_ph=zeros(nNodes,1);
for i=1:nLines
    if real(S.Branch.Z(i,1))==0 %R=0
        times_ph(S.Branch.From(i,1),1)=times_ph(S.Branch.From(i,1),1)+1;
        times_ph(S.Branch.To(i,1),1)=times_ph(S.Branch.To(i,1),1)+1;
    end;
end;
end;
```

```

for i=1:nNodes
    if times_ph(i,1)==3 %o i einai zugos metasximatisti 3 tuligmatwn
        phantomBuses(1,t)=i;
        for line=1:nLines
            if Numbers(line,i)~=0 %vriskei tous relays tou zugou i
                phantomRelays(1,c)=Numbers(line,i);
                c=c+1;
                for col=1:nNodes
                    if Numbers(line,col)~=0 && col~=c %vriskei ton relay sto
                                                                allo akro tis grammis
                        for h=1:mach_numb
                            if S.Machine.BusRef(h,1)==col
                                phantomRelays(1,c)=Numbers(line,col);
                                c=c+1;
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;
end;
[one,numberOfPhantom]=size(phantomRelays);
[one,numberOfPhantom_B]=size(phantomBuses);

primaries=zeros(nRelays,nRelays);
newBus=0;
newLine=0;
%-----Euresi tw n primaries kathe breaker
for i=1:nLines
    column1=1;
    column2=1;
    c=1;
    while Numbers(i,c)==0 && (c<nNodes) %vriskoume ton back-up1
        c=c+1;
    end;
    bus1=c;
    c=c+1;
    while Numbers(i,c)==0
        c=c+1;
    end;
    bus2=c; %vriskoume ton back-up2

    isPhantom=0;
    for h=1:numberOfPhantom
        if phantomRelays(1,h)==Numbers(i,bus1) %elegxei an o relay1 einai
                                                                phantom
            isPhantom=1;
            break;
        end;
    end;
    if (isPhantom==0)
        for k=1:nLines %vriskoume tin primary prostasia
            if Numbers(k,bus2)~=0 && (k~=i)
                found=0;
                for h=1:numberOfPhantom
                    if Numbers(k,bus2)==phantomRelays(1,h) %elegxei an o
                                                                primary einai phantom
                        found=1;
                        break;
                    end;
                end;
            end;
            if (found==0)
                primaries(Numbers(i,bus1),column1)=Numbers(k,bus2); %an
                                                                den einai phantom ton grafei ston primaries_RSV
                column1=column1+1;
            end;
        end;
    end;
end;

```

```

elseif (found==1)% an einai phantom xanakanw ti diadikasia
    c=1;
    while Numbers(k,c)==0 || (c==bus2)
        c=c+1;
    end;
    newBus=c;
    for newLine=1:nLines
        if Numbers(newLine,newBus)~=0 && (newLine~=k)
            found=0;
            for h=1:numberOfPhantom
                if
Numbers(newLine,newBus)==phantomRelays(1,h) %elegxei an o primary einai
                    phantom
                        found=1;
                        break;
                    end;
                end;
            if (found==0)
primaries(Numbers(i,bus1),column1)=Numbers(newLine,newBus); %an den einai
                    phantom ton grafei ston primaries_RSV
                    column1=column1+1;
                elseif (found==1) % an einai kai autos
                    phantom
                        c=1;
                        while Numbers(newLine,c)==0 || (c==newBus)
                            c=c+1;
                        end;
                        newBus=c;
                        continue;
                    end;
                end;
            end;
        end;
    end;
end;

isPhantom=0;
for h=1:numberOfPhantom
    if phantomRelays(1,h)==Numbers(i,bus2) %elegxei an o relay2 einai
        phantom
            isPhantom=1;
            break;
        end;
end;
if (isPhantom==0)
    for k=1:nLines %vriskoume tin primary prostasia
        if Numbers(k,bus1)~=0 && (k~=i)
            found=0;
            for h=1:numberOfPhantom
                if Numbers(k,bus1)==phantomRelays(1,h) %elegxei an o
                    primary einai phantom
                        found=1;
                        break;
                    end;
                end;
            if (found==0)
                primaries(Numbers(i,bus2),column2)=Numbers(k,bus1); %an
                    den einai phantom ton grafei ston primaries_RSV
                    column2=column2+1;

                elseif (found==1)% an einai phantom xanakanw ti diadikasia
                    c=1;
                    while Numbers(k,c)==0 || (c==bus1)
                        c=c+1;
                    end;

```

```

        newBus=c;
        for newLine=1:nLines
            if Numbers(newLine,newBus)~=0 && (newLine~=k)
                found=0;
                for h=1:numberOfPhantom
                    if
Numbers(newLine,newBus)==phantomRelays(1,h) %elegxei an o primary einai
                    phantom
                        found=1;
                        break;
                    end;
                end;
                if (found==0)
primaries(Numbers(i,bus2),column2)=Numbers(newLine,newBus); %an den einai
                    phantom ton grafei ston primaries_RSV
                    column2=column2+1;
                    elseif (found==1) % an einai kai autos
                        phantom
                            c=1;
                            while Numbers(newLine,c)==0 || (c==newBus)
                                c=c+1;
                            end;
                            newBus=c;
                            continue;
                        end;
                    end;
                end;
            end;
        end;
    end;
    primaries_RSV=primaries;
    [linesPrim,columnsPrim]=size(primaries);

%-----vriskw posa breakers exei kathe zugos
BreakersOfNode=zeros(nNodes,1);
for c=1:nNodes
    for i=1:nLines
        if Numbers(i,c)~=0
            BreakersOfNode(c,1)=BreakersOfNode(c,1)+1;
        end;
    end;
end;

Topol_anal %Topologiki analusi tou systimatos

%-----upologismos tw'n I_base gia kathe relay
I_base=zeros(nRelays,1);
for i=1:nLines
    for c=1:nNodes
        if Numbers(i,c)~=0
I_base(Numbers(i,c),1)=(S.Misc.BaseMVA*1000)/(sqrt(3)*S.Bus.KV(c,1));
            end;
        end;
    end;

%-----upologismos tou I_flow apo kathe relay
I_flow_relay=zeros(nRelays,1);
for i=1:nLines
    for c=1:nNodes
        if Numbers(i,c)~=0

```

```

I_flow_relay(Numbers(i,c),1)=abs(S.Flow(i,1))*I_base(Numbers(i,c),1);
    %To S.Flow (pu) anaferetai stin antistoixi grammi. To
    I_flow_relay (A) anaferetai ston ilektronomo
end;
end;
end;

%-----upologismos tou I_fault_relay (A) pou pernaei apo kathe relay gia
close-in fault
I_fault_relay=zeros(nRelays,1);
for i=1:nLines
    for c=1:nNodes
        if Numbers(i,c)~=0
            for line=1:nLines
                if Numbers(line,c)~=0 && line~=i %allos relay pou sundeetai
                    ston idio zugo kai einai se alli grammi
I_fault_relay(Numbers(i,c),1)=I_fault_relay(Numbers(i,c),1)+I_fault_line(lin
e,c)*I_base(Numbers(line,c),1);
                end;
            end;
            for h=1:mach_numb
                if S.Machine.BusRef(h,1)==c
                    I_fault_relay(Numbers(i,c),1)=(I_fault(c,1)-
I_fault_line(i,c))*I_base(Numbers(i,c),1);
                end;
            end;
        end;
    end;
end;

%-----kathorismos tw'n CTR kai tw'n pick_up currents
CT_Ratio=zeros(nRelays,1);
Ipick_up=zeros(nRelays,1);
for i=1:nLines
    if
S.Bus.Voltages(S.Branch.From(i,1),1)>S.Bus.Voltages(S.Branch.To(i,1),1)
    %exoume roi isxu'os kata ti fora pou exei oristei i grammi
    phant=0; %elegxei an o relay stin arxi tis grammis einai phantom
    for h=1:numberOfPhantom
        if Numbers(i,S.Branch.From(i,1))==phantomRelays(1,h)
            phant=1;
            break;
        end;
    end;
    if phant==0 %ean den einai phantom upologizei to CTR
        found=0;
        temp1=I_flow_relay(Numbers(i,S.Branch.From(i,1)),1)+(100-
mod(I_flow_relay(Numbers(i,S.Branch.From(i,1)),1),100)); %pithani timi tou
                2ontos tou CT (den exei elegxthei o koresmos)
        while found==0 %epanaliptikos vroxos mexri na vrethei i timi tou
                2ontos tou CT
            if
(20*temp1)>(I_fault_relay(Numbers(i,S.Branch.From(i,1)),1)) %ean den
                    paratireitai koresmos kata to 3ph fault
                CT_Ratio(Numbers(i,S.Branch.From(i,1)),1)=temp1/5;
                found=1;
            else
                temp1=temp1+100;
            end;
        end;
    end;

temp2=1.5*(I_flow_relay(Numbers(i,S.Branch.From(i,1)),1)/CT_Ratio(Numbers(i,
S.Branch.From(i,1)),1));
    akeraio=fixdec(temp2,0);
    dekadiko=mod(temp2,akeraio);

```

```

        if dekadiko<0.5
            Ipick_up(Numbers(i,S.Branch.From(i,1)),1)=akeraio;
        else
            Ipick_up(Numbers(i,S.Branch.From(i,1)),1)=(akeraio+0.5);
        end;
        if Ipick_up(Numbers(i,S.Branch.From(i,1)),1)>12
            Ipick_up(Numbers(i,S.Branch.From(i,1)),1)=12;
        end;
        if Ipick_up(Numbers(i,S.Branch.From(i,1)),1)<1
            Ipick_up(Numbers(i,S.Branch.From(i,1)),1)=1;
        end;
    end;

    phant=0; %elegxei an o relay sto telos tis grammis einai phantom
    for h=1:numberOfPhantom
        if Numbers(i,S.Branch.To(i,1))==phantomRelays(1,h)
            phant=1;
            break;
        end;
    end;
    if phant==0 %ean den einai phantom upologizei to CTR
        found=0;
        temp1=100; %pithani timi tou 2ontos tou CT (den exei elegxthei o
            %koresmos)
        while found==0 %epanaliptikos vroxos mexri na vrethei i timi tou
            %2ontos tou CT
            if (20*temp1)>(I_fault_relay(Numbers(i,S.Branch.To(i,1)),1))
                %ean den paratireitai koresmos kata to 3ph fault
                CT_Ratio(Numbers(i,S.Branch.To(i,1)),1)=temp1/5;
                found=1;
            else
                temp1=temp1+100;
            end;
        end;
        Ipick_up(Numbers(i,S.Branch.To(i,1)),1)=1;
    end;

    else %exoume roi isxuos kata fora antitheti apo auti pou exei oristei i
        %grammi
        phant=0; %elegxei an o relay stin arxi tis grammis einai phantom
        for h=1:numberOfPhantom
            if Numbers(i,S.Branch.To(i,1))==phantomRelays(1,h)
                phant=1;
                break;
            end;
        end;
        if phant==0 %ean den einai phantom upologizei to CTR
            found=0;
            temp1=I_flow_relay(Numbers(i,S.Branch.From(i,1)),1)+(100-
                mod(I_flow_relay(Numbers(i,S.Branch.From(i,1)),1),100)); %pithani timi tou
                %2ontos tou CT (den exei elegxthei o koresmos)
            while found==0 %epanaliptikos vroxos mexri na vrethei i timi tou
                %2ontos tou CT
                if (20*temp1)>(I_fault_relay(Numbers(i,S.Branch.To(i,1)),1))
                    %ean den paratireitai koresmos kata to 3ph fault
                    CT_Ratio(Numbers(i,S.Branch.To(i,1)),1)=temp1/5;
                    found=1;
                else
                    temp1=temp1+100;
                end;
            end;
        end;

        temp2=1.5*(I_flow_relay(Numbers(i,S.Branch.To(i,1)),1)/CT_Ratio(Numbers(i,S.
            Branch.To(i,1)),1));
        akeraio=fixdec(temp2,0);
        dekadiko=mod(temp2,akeraio);
        if dekadiko<0.5
            Ipick_up(Numbers(i,S.Branch.To(i,1)),1)=akeraio;
        end;
    end;
end;

```

```

else
    Ipick_up(Numbers(i,S.Branch.To(i,1)),1)=(akeraio+0.5);
end;
if Ipick_up(Numbers(i,S.Branch.To(i,1)),1)>12
    Ipick_up(Numbers(i,S.Branch.To(i,1)),1)=12;
end;
if Ipick_up(Numbers(i,S.Branch.To(i,1)),1)<1
    Ipick_up(Numbers(i,S.Branch.To(i,1)),1)=1;
end;
end;

phant=0; %elegxei an o relay sto telos tis grammis einai phantom
for h=1:numberOfPhantom
    if Numbers(i,S.Branch.From(i,1))==phantomRelays(1,h)
        phant=1;
        break;
    end;
end;
if phant==0 %ean den einai phantom upologizei to CTR
    found=0;
    temp1=100;
    while found==0 %epanaliptikos vroxos mexri na vrethei i timi tou
        2ontos tou CT
        if
            (20*temp1)>(I_fault_relay(Numbers(i,S.Branch.From(i,1)),1)) %ean den
                paratireitai koresmos kata to 3ph fault
                CT_Ratio(Numbers(i,S.Branch.From(i,1)),1)=temp1/5;
                found=1;
            else
                temp1=temp1+100;
            end;
        end;
        Ipick_up(Numbers(i,S.Branch.From(i,1)),1)=1;
    end;
end;
end;
%telos kathorismou tw n CTR kai tw n pick up currents

%-----dimiourgia tou pinaka A_lin
k1=0.14;
k2=0.02;
[zeugi,two]=size(SSP);
A_lin=zeros(zeugi,nRelays);
for line=1:zeugi
    pr=SSP(line,1);
    bu=SSP(line,2);
    found=0;
    for l=1:nLines %vriskei to zugo(c_pr) kai ti grammi(l) pou anikei o pr
        for c_pr=1:nNodes
            if Numbers(l,c_pr)==pr
                found=1;
                break;
            end;
        end;
        if found==1
            break;
        end;
    end;
    M=I_fault_relay(pr,1)/(CT_Ratio(pr,1)*Ipick_up(pr,1));
    A_lin(line,pr)=(k1/(M^k2)-1)); %sumplirwnoume ton pinaka A_lin me
        stoixeia pou anaferontai stous pr relays

    found=0;
    for l=1:nLines %vriskei to zugo(c) kai ti grammi(l) pou anikei o bu
        for c=1:nNodes
            if Numbers(l,c)==bu
                found=1;

```

```

        break;
    end;
end;
if found==1
    break;
end;
end;
M=(I_fault_line(l,c_pr)*I_base(bu,1))/(CT_Ratio(bu,1)*Ipick_up(bu,1));
A_lin(line,bu)=(-k1/((M^k2)-1)); %sumplirwnoume ton pinaka A_lin me
                                stoixeia pou anaferontai stous bu relays
end;

%-----ruthmisi tw n ilektronomwn
swsti=0;
while swsti==0
    methodos=input('Epilexe methodo upologismou tw n TDS, topologiki i
grammikou programmatismou (1 i 2?) ');
    if methodos==1
        %ruthmisi tw n TDS me methodo topologikis analisis-arxi
        swsti=1;
        TDS=zeros(nRelays,1);
        for line=1:zeugi %vriskei tin prwti grammi (line) tou SSP pou o b_u
                        den einai breakpoint
            found=0;
            for t=1:(columnBreakPoints-1)
                if breakpoints(1,t)==SSP(line,2)
                    found=1;
                    continue;
                end;
            end;
            if found==0
                break;
            end;
        end;

        for t=1:(columnBreakPoints-1) %arxiki timi tw n TDS tw n breakpoints
            TDS(breakpoints(1,t),1)=0.01;
        end;

        change=1;
        while change>0
            change=0;
            for i=line:zeugi %ruthmisi tw n relays pou den einai breakpoints
                t_pr=A_lin(i,SSP(i,1))*TDS(SSP(i,1),1);
                if TDS(SSP(i,2),1)~=0
                    t_bu=(-A_lin(i,SSP(i,2)))*TDS(SSP(i,2),1);
                    margin=(t_bu-t_pr);
                    if (margin<0.2)
                        TDS(SSP(i,2),1)=(t_pr+0.2)/(-A_lin(i,SSP(i,2)));
                    end;
                else
                    TDS(SSP(i,2),1)=(t_pr+0.2)/(-A_lin(i,SSP(i,2)));
                end;
            end;

            for i=1:(line-1) %xanaruthmizei ta breakpoints
                t_bu=(-A_lin(i,SSP(i,2)))*TDS(SSP(i,2),1);
                t_pr=(A_lin(i,SSP(i,1)))*TDS(SSP(i,1),1);
                margin=(t_bu-t_pr);
                if (margin<0.2)
                    old=TDS(SSP(i,2),1);
                    TDS(SSP(i,2),1)=(t_pr+0.2)/(-A_lin(i,SSP(i,2)));
                    if old~=TDS(SSP(i,2),1)
                        change=change+1;
                    end;
                end;
            end;
        end;
    end;
end;
end;

```

```

disp('Ruthmistikan oi ilektronomoi');
%ruthmisi tw'n TDS me methodo topologikis analisis-telos

elseif methodos==2 %grammikos programmatismos
swsti=1;
f=zeros(1,nRelays);%orismos tou pinaka f
for i=1:nRelays
phant=0; %elegxei an o relay einai phantom
for h=1:numberOfPhantom
if i==phantomRelays(1,h)
phant=1;
break;
end;
end;
if phant==0 %ean den einai phantom
f(1,i)=1;
end;
end;

for i=1:zeugi %orismos tou pinaka b
b(i,1)=(-0.2);
end;

Aeq=[];
beq=[];
for i=1:nRelays
lb=0.01;
ub=1.1;
end;

TDS=linprog(f,A_lin,b,Aeq,beq,lb,ub);
disp('Ruthmistikan oi ilektronomoi');
else
disp('Lathos methodos!');
end;
end;

for i=1:nRelays %ellexisti dunati timi to 0.01
if TDS(i,1)<0.01 && TDS(i,1)>0
TDS(i,1)=0.01;
end;
end;

xronoi=zeros(zeugi,2); %upologizei se kathe zeugos to xrono leitourgias tw'n
ilektronomwn
for i=1:zeugi
xronoi(i,1)=A_lin(i,SSP(i,1))*TDS(SSP(i,1),1);
xronoi(i,2)=-A_lin(i,SSP(i,2))*TDS(SSP(i,2),1);
end;
%telos tis ruthmisis tw'n ilektronomwn

```

Το κυρίως πρόγραμμα καλεί κάποια υποπρογράμματα, τα οποία με τη σειρά που καλούνται είναι:

Συναρτήσεις χρησιμοποιούμενες από την ανάλυση ροών φορτίου

```

function S=readcf(fname)
% readcf.m
% Read the common format file and create a data dictionary structure
if nargin<1, [fname,pname]=uigetfile('*.cf'); fname=[pname fname]; end
S.Misc.BaseMVA=100;
fcf=fopen(fname,'r'); % Open common format file
s=fgetl(fcf);

```

```

while strcmp(s(1:min(3,length(s))), 'BUS')~=1, % Find the start of bus data
    s=fgetl(fcf);
end
s=fgetl(fcf);
while s(1)=='%', s=fgetl(fcf); end;
nL=0; ng=0; n=0;
S.PQlist=[]; S.SlackList=[]; S.BlackList=[]; S.PVlist=[];
S.Bus.Voltages=sparse(n,1);
S.Interch.I=[];
S.Interch.areaCount=zeros(100,1);
while strcmp(s(1:4), '-999')~=1,
    n=n+1;
    Pg=scanreal(s,59,67);
    Qg=scanreal(s,68,75);
    Pd=scanreal(s,41,49);
    Qd=scanreal(s,50,58);
    S.Bus.busType(n,1)=sscanf(s(26), '%d');
    if (s(26)=='1') || (s(26)=='2') || (s(26)=='3')
        if s(26)=='3', S.SlackList=[S.SlackList n];
        else S.PVlist=[S.PVlist n];
        end;
    else
        S.PQlist=[S.PQlist n];
    end;
    bus=scanint(s,1,5);
    S.Bus.Number(n,1)=bus;
    S.Bus.Name{n}=s(6:17);
    S.newBus(bus)=n;
    S.Machine.newGen(bus,1)=0;
    if (Pg~=0) || (Qg~=0) || (s(26)=='3') || (s(26)=='2'),
        ng=ng+1;
        S.Machine.newGen(bus,1)=ng;
        S.Machine.BusRef(ng,1)=bus; % Generation bus number
        S.Machine.MW(ng,1)=Pg/S.Misc.BaseMVA; % Active Power Generation
                                         (actual)
        S.Machine.MVAR(ng,1)=Qg/S.Misc.BaseMVA; % Reactive Power Generation
        S.Machine.Status(ng,1)=1;
    end
    if (Pd~=0) || (Qd~=0),
        nL=nL+1;
        newLoad(bus,1)=nL;
        S.Load.BusRef(nL,1)=bus; % Load bus number
        S.Load.MW(nL,1)=Pd/S.Misc.BaseMVA; % Net active power load
        S.Load.MVAR(nL,1)=Qd/S.Misc.BaseMVA; % Net reactive power load
        S.Load.Status(nL,1)=1;
    end;
    S.Bus.Generation(n,1)=(Pg+j*Qg)/S.Misc.BaseMVA;
    S.Bus.Load(n,1)=(Pd+j*Qd)/S.Misc.BaseMVA;
    S.Bus.Voltages(n,1)=scanreal(s,28,33,1)*...
        exp(sqrt(-1)*scanreal(s,34,40)*pi/180);
    S.Bus.area(n,1)=sscanf(s(19:20), '%d'); % Now this is numeric, +1
    S.Bus.Zone(n,:)=s(21:23);
    S.Bus.KV(n,1)=scanreal(s,77,83);
    if S.Machine.newGen(bus),
        S.Machine.MinOperatingVolt(ng,1)=scanreal(s,85,90);
        S.Machine.MaxOperatingVolt(ng,1)=scanreal(s,85,90);
        S.Machine.MaxQOutput(ng,1)=scanreal(s,91,98)/S.Misc.BaseMVA;
        S.Machine.MinQOutput(ng,1)=scanreal(s,99,106)/S.Misc.BaseMVA;
        S.Machine.ControlledBusRef(ng,1)=scanreal(s,124,128);
    end
    S.Bus.G(n,1)=scanreal(s,107,114);
    S.Bus.B(n,1)=scanreal(s,115,122);
    s=fgetl(fcf);
    while s(1)=='%', s=fgetl(fcf); end;
end;
minArea=min(S.Bus.area);
dArea=1-minArea;
S.Interch.areaCount=sparse(S.Bus.area+dArea,1,1);

```

```

while ~strcmp(s(1:3),'BRA'), s=fgetl(fcf); end
s=fgetl(fcf);
while s(1)=='%', s=fgetl(fcf); end;
nn=0;
while ~strcmp(s(1:4),'-999'),
    nn=nn+1;
    S.Branch.To(nn,1)=S.newBus(scanint(s,1,5));
    S.Branch.From(nn,1)=S.newBus(scanint(s,6,10));
    R=scanreal(s,20,29); % Resistance
    X=scanreal(s,30,40); % Reactance
    S.Branch.Z(nn,1)=R+j*X;
    S.Branch.B(nn,1)=scanreal(s,41,49); % Line Charging
    S.Branch.RateValue(nn,1)=scanreal(s,51,55)/S.Misc.BaseMVA;
    S.Branch.Type(nn,1)=scanint(s,19,19); % Type
    if S.Branch.Type(nn,1), % If dealing with a transformer
        tap=scanreal(s,77,82);
        alpha=scanreal(s,84,90)*pi/180;
        S.Branch.TAP(nn,1)=tap.*exp(j*alpha);
    else
        S.Branch.TAP(nn,1)=1;
    end;
    s=fgetl(fcf); % Next line
    while s(1)=='%', s=fgetl(fcf); end; % Skip comments
end
S.Branch.YI=zeros(nn,1);
S.Branch.YJ=zeros(nn,1);
S.Branch.Status=ones(nn,1);
kzero=find(S.Branch.RateValue==0); % No zero ratings permitted
S.Branch.RateValue(kzero)=Inf;

while ~strcmp(s(1:3),'INT'),
    s=fgetl(fcf);
end
s=fgetl(fcf);
while s(1)=='%', s=fgetl(fcf); end;
nInterch=0;
while ~strcmp(s(1:2),'-9'),
    nInterch=nInterch+1;
    S.Interch.I(nInterch,1)=scanint(s,1,2);
    S.Interch.ControlGenRef(nInterch,1)=scanint(s,3,7);
    S.Interch.SchedNetMW(nInterch,1)=scanreal(s,21,29)/S.Misc.BaseMVA;
    s=fgetl(fcf);
    while s(1)=='%', s=fgetl(fcf); end;
end
fclose(fcf);
S.Machine.BusRef=S.newBus(S.Machine.BusRef)'; % Use internal bus numbers
S.Load.BusRef=S.newBus(S.Load.BusRef)'; % "
ky=find(S.Machine.ControlledBusRef); % "
kx=S.Machine.ControlledBusRef(ky);
S.Machine.ControlledBusRef(ky)=S.newBus(kx);
S.Bus.n=length(S.Bus.Voltages);
S.Branch.nn=length(S.Branch.From);
S.Bus.area=S.Bus.area+dArea;

```

```

function Ybus=bldybus(S)
% bldybus.m Build the y-bus matrices
Ybus=sparse(S.Bus.n,S.Bus.n);
YL=[];
if isfield(S.Bus,'G'),
    YL=S.Bus.G;
end
if isfield(S.Bus,'B'),

```

```

YL=YL+i*S.Bus.B;
end
if ~isempty(YL),
    Ybus=Ybus+diag(sparse(YL));
end
n=S.Bus.n;
Y11=(1./S.Branch.Z+(i*S.Branch.B)/2).*S.Branch.Status;
Ybus=Ybus+sparse(S.Branch.From,S.Branch.From,Y11,n,n);
Y12=(-1./(S.Branch.TAP.*S.Branch.Z)).*S.Branch.Status;
Ybus=Ybus+sparse(S.Branch.From,S.Branch.To,Y12,n,n);
Y21=(-1./(conj(S.Branch.TAP).*S.Branch.Z)).*S.Branch.Status;
Ybus=Ybus+sparse(S.Branch.To,S.Branch.From,Y21,n,n);
Y22=(1./((abs(S.Branch.TAP).^2).*S.Branch.Z)+(i*S.Branch.B)/2).*S.Branch.Sta
tus;
Ybus=Ybus+sparse(S.Branch.To,S.Branch.To,Y22,n,n);
if isfield(S.Branch,'YI'),
    Ybus=Ybus+sparse(S.Branch.From,S.Branch.From,S.Branch.YI,n,n);
end
if isfield(S.Branch,'YJ'),
    Ybus=Ybus+sparse(S.Branch.To,S.Branch.To,S.Branch.YJ,n,n);
end
if isfield(S,'Shunt'),
    Ybus=Ybus+sparse(S.Shunt.I,S.Shunt.I,i*S.Shunt.BINIT,n,n);
end;

```

```

function S=pfsolve(S)
% This routine solves the power flow problem using a Newton-Raphson
% iteration with full Jacobian update at every iteration.
% Originally developed by Professor Christopher DeMarco, 1997
% and expanded by Fernando Alvarado 1998, 1999
%
% Compute initial mismatch
MAXITER=15;    n=S.Bus.n;
vbus=S.Bus.Voltages;
S.mismatch = Inf; % pick off largest component of relevant mismatch
% Newton-Raphson Iteration
S.itcnt = 0;
while S.mismatch > .0001
    fullmiss = pfmisss(S);
    rmiss=[real(fullmiss(S.PVlist));real(fullmiss(S.PQlist)); ...
        imag(fullmiss(S.PQlist))];
    S.mismatch = max(abs(rmiss));
    [dsdd, dsdv] = pflowjac(S.Ybus,vbus);

% dsdd and dsdv are composed of all (complex) partial derivatives
% rjac is a selection from these
    rjac = [ ...
        real(dsdd(S.PVlist,S.PVlist)) real(dsdd(S.PVlist,S.PQlist)) ...
        real(dsdv(S.PVlist,S.PQlist)); ...
        real(dsdd(S.PQlist,S.PVlist)) real(dsdd(S.PQlist,S.PQlist)) ...
        real(dsdv(S.PQlist,S.PQlist)); ...
        imag(dsdd(S.PQlist,S.PVlist)) imag(dsdd(S.PQlist,S.PQlist)) ...
        imag(dsdv(S.PQlist,S.PQlist)) ];
    x=[angle(vbus(S.PVlist))*pi/180;
        angle(vbus(S.PQlist))*pi/180;
        abs(vbus(S.PQlist))];
    dx = -rjac\rmiss; % Here is the actual update
    del=angle(vbus);
    vmag=abs(vbus);
    n1=length(S.PVlist); n2=n1+length(S.PQlist); n3=length(dx);
    del(S.PVlist)=del(S.PVlist)+dx(1:n1);
    del(S.PQlist)=del(S.PQlist)+dx((n1+1):n2);
    vmag(S.PQlist)=vmag(S.PQlist)+dx((n2+1):n3);
    vbus=vmag.*exp(j*del);
    S.Bus.Voltages=vbus;
    S.itcnt=S.itcnt+1;
    if (S.itcnt > MAXITER) || ((S.itcnt>3)&&(S.mismatch>100)),

```

```

        S.itcnt=Inf;
        break;
    end;
end
if (S.itcnt<=MAXITER),
    S.Bus.Voltages=vbus;
    disp([int2str(S.itcnt) ' iterations']);
end

```

```

function nmiss = pfmisss(S)
% Form the vector of complex power mismatches.
% Usage: nmiss = pfmisss(Y,vb,s_net_inject)
% Arguments: complex bus admittance matrix, Y;
%            vector of complex bus voltage phasors, vb;
%            vector of complex bus power demand (net),
%            S.snet; (note that s_net_inject will be
%            negative at buses demanding power)
% Returns:   a full complex n-vector of power mismatches.
%            Power LEAVING the bus is positive
% Comments:  Note that we return ALL mismatches at EVERY bus.
ib=S.Ybus*S.Bus.Voltages;
nmiss = S.Bus.Voltages.*conj(ib)-S.snet;

```

```

function [dSdd,dSdv] = pflowjac(Y,vb)
% Construct the power flow Jacobian matrix in complex form
% Usage: [dSdd,dSdv] = pflowjac(Y,vb)
% Arguments: complex bus admittance matrix, Y;
%            vector of complex bus voltage phasors, vb.
% Returns:   Two full, complex nxn matrices of partial derivatives;
%            dSdd = partial of complex power w.r.t. delta;
%            dSdv = partial of complex power w.r.t. voltage magnitude.
% Comments:  We return ALL the partial derivatives of power absorbed at
EVERY
%            bus with respect to EVERY voltage magnitude and phase angle.
% The operation of this routine is transparent if one recognizes that:
% S = diag(vb)*conj(ib) = diag(conj(ib))*vb;
% hence:
% dS/d(delta) =
diag(vb)*conj(d(ib)/d(delta))+diag(conj(ib))*d(vb)/d(delta)
% dS/d(vmag) = diag(vb)*conj(d(ib)/d(vmag))+diag(conj(ib))*d(vb)/d(vmag)
% This routine was developed by Chris DeMarco of the University of
Wisconsin
ib=Y*vb;
dSdd=j*diag(conj(ib).*vb)-j*diag(vb)*conj(Y)*diag(conj(vb));
dSdv=diag(conj(ib).*(vb./abs(vb)))+diag(vb)*conj(Y)*diag(conj(vb)./abs(vb));

```

```

function i=scanint(s,beg,last,default)
% Function to read an integer from a designated field
if nargin<4, default=0; end;
i=default;
[dummy,ns]=size(s);
if ns<beg, return; end;
if last<beg, return; end;
last=min(last,ns);
if isspace(s(beg:last)), return; end;
i=sscanf(s(beg:last),'%d');
return;

```

```

function i=scanreal(s,beg,last,default)
% Function to read a real from a designated field
if nargin<4, default=0; end;
i=default;
[dummy,ns]=size(s);
if ns<beg, return; end;
if last<beg, return; end;

```

```

last=min(last,ns);
if isspace(s(beg:last)), return; end;
i=sscanf(s(beg:last),'%f');
return;

```

```

function [Sflow,iOvld]=flows(S)
% Calculate all the line flows and report overloads
Vf=S.Bus.Voltages;
Sflow=(Vf(S.Branch.From,1)-Vf(S.Branch.To,1))./S.Branch.Z+ ...
    Vf(S.Branch.From,1).*(j*S.Branch.B/2);
Sflow=Sflow/2;
iOvld=find((S.Branch.Status.*abs(Sflow))>S.Branch.RateValue);

```

Συναρτήσεις χρησιμοποιούμενες από την ανάλυση βραχυκυκλωμάτων

```

function [I_fault,I_fault_line,Vnodes] =
shortcircuit(SBranchFrom,SBranchoTo,A)
% start to input data
finput=fopen('scparam.txt'); % input calculation mode and base MVA
ReadLine=fgets(finput);
BranchData=str2num(ReadLine);
while BranchData(1)~-99
    prefaultrv = BranchData(1); % The calculation mode: including the load
    currents or not
    MVAbase= BranchData(2);
    ReadLine=fgets(finput);
    BranchData=str2num(ReadLine);
end;
fclose(finput);
finput=fopen('scinput.txt'); % input generator data
ReadLine=fgets(finput);
BranchData=str2num(ReadLine);
I=1;
while BranchData(1)~-99
    gennum(I)=BranchData(1);
    genconnect(I)=BranchData(2);
    negative(I)=BranchData(3);
    genzr(I)=BranchData(4);
    genzx(I)=BranchData(5);
    genpr(I)=BranchData(6);
    genpx(I)=BranchData(7);
    if negative(I)==1 % if data of negative sequence = positive sequence
        gennr(I)=genpr(I); % use the same data
        gennx(I)=genpx(I);
    else
        gennr(I)=BranchData(8);
        gennx(I)=BranchData(9);
    end;
    ReadLine=fgets(finput);
    BranchData=str2num(ReadLine);
    I=I+1;
end;
total_gen=I-1;
ReadLine=fgets(finput); % input shunt elements data
BranchData=str2num(ReadLine);
I=1;
while BranchData(1)~-99
    shuntnum(I)=BranchData(1);
    shuntconnect(I)=BranchData(2);
    negative(I)=BranchData(3);
    shuntzr(I)=BranchData(4);
    shuntzx(I)=BranchData(5);
    shuntpr(I)=BranchData(6);
    shuntpx(I)=BranchData(7);

```

```

        if negative(I)==1
            shuntnr(I)=shuntpr(I);
            shuntnx(I)=shuntpx(I);
        else
            shuntnr(I)=BranchData(8);
            shuntnx(I)=BranchData(9);
        end;
        ReadLine=fgets(finput);
        BranchData=str2num(ReadLine);
        I=I+1;
    end;
    total_shunt=I-1;
    ReadLine=fgets(finput); % input load data
    BranchData=str2num(ReadLine);
    I=1;
    while BranchData(1) ~= -99
        loadnum(I)=BranchData(1);
        loadbus(I)=BranchData(2);
        loadP(I)=BranchData(3);
        loadQ(I)=BranchData(4);
        ReadLine=fgets(finput);
        BranchData=str2num(ReadLine);
        I=I+1;
    end;
    total_load=I-1;
    ReadLine=fgets(finput); % input lines data
    BranchData=str2num(ReadLine);
    I=1;
    while BranchData(1) ~= -99
        startbus(I) = BranchData(2);
        endbus(I) = BranchData(3);
        negative(I)=BranchData(4);
        branchtype(I) = BranchData(5);
        frombuscon(I) = BranchData(6);
        tobuscon(I) = BranchData(7);
        ratiop(I) = BranchData(8);
        ratios(I) = BranchData(9);
        zr(I) = BranchData(10);
        zx(I) = BranchData(11);
        zcon(I) = BranchData(12);
        zsup(I) = BranchData(13);
        pr(I) = BranchData(14);
        px(I) = BranchData(15);
        pg(I) = BranchData(16);
        pb(I) = BranchData(17);
        if negative(I)==1
            nr(I)=pr(I);
            nx(I)=px(I);
            ng(I)=pg(I);
            nb(I)=pb(I);
        else
            nr(I)=BranchData(18);
            nx(I)=BranchData(19);
            ng(I)=BranchData(20);
            nb(I)=BranchData(21);
        end;
        ReadLine=fgets(finput);
        BranchData=str2num(ReadLine);
        I=I+1;
    end;
    total_line=I-1;
    total_bus=max(max(startbus),max(endbus));
    ReadLine=fgets(finput); % input mutual couplings data
    BranchData=str2num(ReadLine);
    I=1;
    while BranchData(1) ~= -99
        mutualset(I,:)=BranchData(1:2);
        mutualendbus(I,:)=BranchData(3:6);
    end;

```

```

        mutualzim(I)=BranchData(7)+i*BranchData(8);
        mutualpim(I)=BranchData(9)+i*BranchData(10);
        mutualnim(I)=BranchData(11)+i*BranchData(12);
        ReadLine=fgets(fininput);
        BranchData=str2num(ReadLine);
        I=I+1;
    end;
    total_mutual=I-1;
    if total_mutual==0
        mutual=0;
    else
        mutual=1;
    end;
    fclose(fininput);

    fininput=fopen('fltinput.txt'); % input fault data
    ReadLine=fgets(fininput);
    BranchData=str2num(ReadLine);
    I=1;
    while BranchData(1)~-99
        shortbus(I,1)=BranchData(1);
        z(I,1)=BranchData(2)+i*BranchData(3);
        z(I,2)=BranchData(4)+i*BranchData(5);
        z(I,3)=BranchData(6)+i*BranchData(7);
        z(I,4)=BranchData(8)+i*BranchData(9);
        ReadLine=fgets(fininput);
        BranchData=str2num(ReadLine);
        I=I+1;
    end;

    total_fault=I-1;
    fclose(fininput);
    if prefaultv==1 % not including load currents
        v=ones(total_bus,1); % Initial value for bus voltages = desired voltage
                                % in data file
        va=v;
        vb=v*cos(-2/3*pi)+v*sin(-2/3*pi)*i;
        vc=v*cos(2/3*pi)+v*sin(2/3*pi)*i;
    else % including load currents
        fininput=fopen('BUSOTP.txt'); % Initial value for bus voltages = voltages
                                        % of power flow
        results
        ReadLine=fgets(fininput);
        BranchData=str2num(ReadLine);
        I=1;
        if BranchData(1)==0
            I=I+1;
            while I<=total_bus+1
                ReadLine=fgets(fininput);
                BranchData=str2num(ReadLine);
                va(I-
1)=BranchData(2)*cos(BranchData(3)*pi/180)+BranchData(2)*sin(BranchData(3)*p
i/180)*i;
                vb(I-
1)=BranchData(4)*cos(BranchData(5)*pi/180)+BranchData(4)*sin(BranchData(5)*p
i/180)*i;
                vc(I-
1)=BranchData(6)*cos(BranchData(7)*pi/180)+BranchData(6)*sin(BranchData(7)*p
i/180)*i;
                I=I+1;
            end;
        end;
        fclose(fininput);
    end;
    a=-0.5+sqrt(3)*j/2;
    T=[1 1 1;1 a^2 a;1 a a^2]; % transforming matrix
    S=inv(T);
    C=1/sqrt(3)*[1 -1 0;0 1 -1;-1 0 1]; % matrix used in transformer model

```

```

% Build prefault three-phase voltage matrix
vpre=zeros(total_bus*3,1);
for x=1:total_bus
    vpre(x*3-2)=va(x);
    vpre(x*3-1)=vb(x);
    vpre(x*3)=vc(x);
end;

% Build the Y Matrix of zero-sequence
zz=zeros(total_line,total_line);
for x=1:total_line
    if branchtype(x)==0 % input elements data other than transformer
        zz(x,x)=zr(x)+zx(x)*i;
    else
        zz(x,x)=1;
    end;
end;
if total_mutual>0 % input mutual couplings data
    for x=1:total_mutual
        zz(mutualset(x,1),mutualset(x,2))=mutualzim(x);
        zz(mutualset(x,2),mutualset(x,1))=mutualzim(x);
    end;
end;
yz=inv(zz);
for x=1:total_line
    if branchtype(x)==1
        yz(x,x)=0;
    end;
end;
Ymatrixz=A'*yz*A;
if total_gen~=0
    for x=1:total_gen % add the generator admittance
        if genconnect(x)==0
            bus=gennum(x);
            Ymatrixz(bus,bus)=Ymatrixz(bus,bus)+1/(genzr(x)+genzx(x)*i);
        end;
    end;
end;
if total_shunt~=0
    for x=1:total_shunt % add the shunt element admittance
        if shuntconnect(x)==0
            bus=shuntnum(x);
            Ymatrixz(bus,bus)=Ymatrixz(bus,bus)+1/(shuntzr(x)+shuntzx(x)*i);
        end;
    end;
end;
for x=1:total_line % add line shunt capacitance
    from=startbus(x);
    to=endbus(x);
    if branchtype(x)==0
        Ymatrixz(from,from)=Ymatrixz(from,from)+zcon(x)+zsup(x)*i;
        Ymatrixz(to,to)=Ymatrixz(to,to)+zcon(x)+zsup(x)*i;
    end;
end;
if prefaultv==0 % if considering load currents, add the equivalent load
    impedance
    for x=1:total_load
        bus=loadbus(x);
        PQ=(loadP(x)-loadQ(x)*j)/MVAbase;
        zload(x)=va(bus)*conj(va(bus))/PQ;
        Ymatrixz(bus,bus)=Ymatrixz(bus,bus)+1/zload(x);
    end;
end;
% Build the Y Matrix of positive-sequence
zp=zeros(total_line,total_line);
for x=1:total_line
    if branchtype(x)==0 % input elements data other than transformer

```

```

        zp(x,x)=pr(x)+px(x)*i;
    else
        zp(x,x)=1;
    end;
end;
if total_mutual>0 % input mutual couplings data
    for x=1:total_mutual
        zp(mutualset(x,1),mutualset(x,2))=mutualpim(x);
        zp(mutualset(x,2),mutualset(x,1))=mutualpim(x);
    end;
end;
yp=inv(zp);
for x=1:total_line
    if branchtype(x)==1
        yp(x,x)=0;
    end;
end;
Ymatrixp=A'*yp*A;
if total_gen~=0
    for x=1:total_gen % add the generator admittance
        bus=gennum(x);
        Ymatrixp(bus,bus)=Ymatrixp(bus,bus)+1/(genpr(x)+genpx(x)*i);
    end;
end;
if total_shunt~=0
    for x=1:total_shunt % add the shunt element admittance
        bus=shuntnum(x);
        Ymatrixp(bus,bus)=Ymatrixp(bus,bus)+1/(shuntpr(x)+shuntpx(x)*i);
    end;
end;
for x=1:total_line % add line shunt capacitance
    from=startbus(x);
    to=endbus(x);
    if branchtype(x)==0
        Ymatrixp(from,from)=Ymatrixp(from,from)+pg(x)+pb(x)*i;
        Ymatrixp(to,to)=Ymatrixp(to,to)+pg(x)+pb(x)*i;
    end;
end;
if prefaultv==0 % if considering load currents, add the equivalent load
    impedance
    for x=1:total_load
        bus=loadbus(x);
        Ymatrixp(bus,bus)=Ymatrixp(bus,bus)+1/zload(x);
    end;
end;
% Build the Y Matrix of negative-sequence
zn=zeros(total_line,total_line);
for x=1:total_line
    if branchtype(x)==0 % input elements data other than transformer
        zn(x,x)=nr(x)+nx(x)*i;
    else
        zn(x,x)=1;
    end;
end;
if total_mutual>0 % input mutual couplings data
    for x=1:total_mutual
        zn(mutualset(x,1),mutualset(x,2))=mutualnim(x);
        zn(mutualset(x,2),mutualset(x,1))=mutualnim(x);
    end;
end;
yn=inv(zn);
for x=1:total_line
    if branchtype(x)==1
        yn(x,x)=0;
    end;
end;
Ymatrixn=A'*yn*A;
if total_gen~=0

```

```

        for x=1:total_gen % add the generator admittance
            bus=gennum(x);
            Ymatrixn(bus,bus)=Ymatrixn(bus,bus)+1/(gennr(x)+gennx(x)*i);
        end;
    end;
    if total_shunt~=0
        for x=1:total_shunt % add the shunt element admittance
            bus=shuntnum(x);
            Ymatrixn(bus,bus)=Ymatrixn(bus,bus)+1/(shuntnr(x)+shuntnx(x)*i);
        end;
    end;
    for x=1:total_line % add line shunt capacitance
        from=startbus(x);
        to=endbus(x);
        if branchtype==0
            Ymatrixn(from,from)=Ymatrixn(from,from)+ng(x)+nb(x)*i;
            Ymatrixn(to,to)=Ymatrixn(to,to)+ng(x)+nb(x)*i;
        end;
    end;
    if prefaultv==0 % if considering load currents, add the equivalent load
                    impedance
        for x=1:total_load
            bus=loadbus(x);
            Ymatrixn(bus,bus)=Ymatrixn(bus,bus)+1/zload(x);
        end;
    end;

    % Convert to Y matrix in phase domain
    Ymatrix=zeros(total_bus*3,total_bus*3);
    for x=1:total_bus
        for y=1:total_bus
            if y<x
                Ymatrix(3*x-2:3*x,3*y-2:3*y)=Ymatrix(3*y-2:3*y,3*x-2:3*x);
            else
                Ymatrix(((3*x)-2):(3*x),((3*y)-2):(3*y))=T*[Ymatrixz(x,y) 0 0;0
Ymatrixp(x,y) 0;0 0 Ymatrixn(x,y)]*S;
            end;
        end;
    end;
    % Add transformer components
    for x=1:total_line
        from=startbus(x);
        to=endbus(x);
        if branchtype(x)==1
            ys1=1/(pr(x)+px(x)*i)+pg(x)+pb(x)*i;
            ys0=1/(zr(x)+zx(x)*i)+zcon(x)+zsup(x)*i;
            y1=1/(pr(x)+px(x)*i);
            y0=1/(zr(x)+zx(x)*i);
            alpha=ratiop(x);
            beta=ratios(x);
            Y1=1/3*[ys0+2*ys1 ys0-ys1 ys0-ys1;ys0-ys1 ys0+2*ys1 ys0-ys1;ys0-ys1
ys0+2*ys1];
            Y2=1/3*[2*ys1 -ys1 -ys1;-ys1 2*ys1 -ys1;-ys1 -ys1 2*ys1];
            Y1x=1/3*[y0+2*y1 y0-y1 y0-y1;y0-y1 y0+2*y1 y0-y1;y0-y1 y0-y1
y0+2*y1];
            Y2x=1/3*[2*y1 -y1 -y1;-y1 2*y1 -y1;-y1 -y1 2*y1];
            Y3=y1*C;
            Y3t=y1*C';
            fromco=frombuscon(x);
            toco=tobuscon(x);
            if fromco==0 % for different connections, use different functions
                switch toco
                    case 0
                        Ya=Y1;
                        Yb=Y1x;
                        Yc=Y1x;
                        Yd=Y1;
                    case 1

```

```

        Ya=Y1;
        Yb=Y2x;
        Yc=Y2x;
        Yd=Y2;
    otherwise
        Ya=Y1;
        Yb=Y3;
        Yc=Y3t;
        Yd=Y2;
    end;
elseif fromco==1
    switch toco
    case 0
        Yd=Y1;
        Yb=Y2x;
        Yc=Y2x;
        Ya=Y2;
    case 1
        Ya=Y2;
        Yb=Y2x;
        Yc=Y2x;
        Yd=Y2;
    otherwise
        Ya=Y2;
        Yb=Y3;
        Yc=Y3t;
        Yd=Y2;
    end;
else
    switch toco
    case 0
        Ya=Y2;
        Yb=Y3t;
        Yc=Y3;
        Yd=Y1;
    case 1
        Ya=Y2;
        Yb=Y3t;
        Yc=Y3;
        Yd=Y2;
    otherwise
        Ya=Y2;
        Yb=Y2x;
        Yc=Y2x;
        Yd=Y2;
    end;
end;
Ytr=[Ya/alpha^2 -Yb/(alpha*beta);-Yc/(alpha*beta) Yd/beta^2]; %
                                     build the final matrix
Ymatrix(from*3-2:from*3,from*3-2:from*3)=Ymatrix(from*3-
2:from*3,from*3-2:from*3)+Ytr(1:3,1:3); % overlap it to the Y amtrix
Ymatrix(to*3-2:to*3,to*3-2:to*3)=Ymatrix(to*3-2:to*3,to*3-
2:to*3)+Ytr(4:6,4:6);
Ymatrix(from*3-2:from*3,to*3-2:to*3)=Ymatrix(from*3-2:from*3,to*3-
2:to*3)+Ytr(1:3,4:6);
Ymatrix(to*3-2:to*3,from*3-2:from*3)=Ymatrix(to*3-2:to*3,from*3-
2:from*3)+Ytr(4:6,1:3);
end;
end;
Zmatrix=inv(Ymatrix); % get the Z matrix
% Short circuit study
Ishort=zeros(total_bus*3,1);
I_fault_line=zeros(total_line,total_bus);
for faultNode=1:total_fault
    k=shortbus(faultNode,1);
    if z(faultNode,4)==-1 % for non-grounded fault
        if z(faultNode,1)==-1

```

```

        Zx=Zmatrix(3*k-1:3*k,3*k-1:3*k)+[z(faultNode,2) 0;0
z(faultNode,3)];
        Zf=[Zx(1,:)-Zx(2,:);1 1];
        Ip=inv(Zf)*[vb(k)-vc(k);0];
        If=[0;-Ip];
        elseif z(faultNode,2)==-1
            Zx=[Zmatrix(3*k-2,3*k-2)+z(faultNode,1) Zmatrix(3*k-
2,3*k);Zmatrix(3*k,3*k-2)
            Zmatrix(3*k,3*k)+z(faultNode,3)];
            Zf=[Zx(1,:)-Zx(2,:);1 1];
            Ip=inv(Zf)*[va(k)-vc(k);0];
            If=[-Ip(1);0;-Ip(2)];
        elseif z(faultNode,3)==-1
            Zx=Zmatrix(3*k-2:3*k-1,3*k-2:3*k-1)+[z(faultNode,1) 0;0
z(faultNode,2)];
            Zf=[Zx(1,:)-Zx(2,:);1 1];
            Ip=inv(Zf)*[va(k)-vb(k);0];
            If=[-Ip;0];
        else %3-phase fault
            Zx=Zmatrix(3*k-2:3*k,3*k-2:3*k)+[z(faultNode,1) 0 0;0
z(faultNode,2) 0;0 0 z(faultNode,3)];
            Zf=[Zx(1,:)-Zx(2,:);Zx(2,:)-Zx(3,:);1 1 1];
            Ip=inv(Zf)*[va(k)-vb(k);vb(k)-vc(k);0];
            If=-Ip;
        end;
    else % for grounded fault
        if (z(faultNode,1)==-1)&&(z(faultNode,2)==-1)
            Ic=vc(k)/(z(faultNode,3)+z(faultNode,4)+Zmatrix(3*k,3*k));
            If=[0;0;-Ic];
        elseif (z(faultNode,1)==-1)&&(z(faultNode,3)==-1)
            Ib=vb(k)/(z(faultNode,2)+z(faultNode,4)+Zmatrix(3*k-1,3*k-1));
            If=[0;-Ib;0];
        elseif (z(faultNode,2)==-1)&&(z(faultNode,3)==-1)
            Ia=va(k)/(z(faultNode,1)+z(faultNode,4)+Zmatrix(3*k-2,3*k-2));
            If=[-Ia;0;0];
        elseif z(faultNode,1)==-1
            Zx=Zmatrix(3*k-1:3*k,3*k-1:3*k)+[z(faultNode,2) 0;0
z(faultNode,3)]+z(faultNode,4)*ones(2,2);
            Ip=inv(Zx)*[vb(k);vc(k)];
            If=[0;-Ip];
        elseif z(faultNode,2)==-1
            Zx=[Zmatrix(3*k-2,3*k-2)+z(faultNode,1) Zmatrix(3*k-
2,3*k);Zmatrix(3*k,3*k-2)
            Zmatrix(3*k,3*k)+z(faultNode,3)]+z(faultNode,4)*ones(2,2);
            Ip=inv(Zx)*[va(k);vc(k)];
            If=[-Ip(1);0;-Ip(2)];
        elseif z(faultNode,3)==-1
            Zx=Zmatrix(3*k-2:3*k-1,3*k-2:3*k-1)+[z(faultNode,1) 0;0
z(faultNode,2)]+z(faultNode,4)*ones(2,2);
            Ip=inv(Zx)*[va(k);vb(k)];
            If=[-Ip;0];
        else % (3-phase ground fault)
            Zx=Zmatrix(3*k-2:3*k,3*k-2:3*k)+[z(faultNode,1) 0 0;0
z(faultNode,2) 0;0 0 z(faultNode,3)]+z(faultNode,4)*ones(3,3);
            Ip=inv(Zx)*[va(k);vb(k);vc(k)];
            If=-Ip;
        end;
    end;
    Ishort=zeros(3*total_bus,1);
    Vshort=zeros(3*total_bus,1);
    Ishort(3*k-2:3*k)=If; % calculate fault currents
    I_fault(faultNode,1)=abs(Ishort(3*k));

    deltaV=Zmatrix*Ishort;
    Vshort=vpre+deltaV; % calculate post-fault voltages

    for h=1:total_bus
        Vnodes(h,faultNode)=abs(Vshort(3*h));
    end
end

```

```

end;

Iline=zeros(total_line*3,1);
Vline=zeros(total_line*3,1);
for x=1:total_line % build the voltage difference matrix for the line
    from=startbus(x);
    to=endbus(x);
    Vline(3*x-2:3*x)=Vshort(from*3-2:from*3)-Vshort(to*3-2:to*3);
end;
Yline=zeros(total_line*3,total_line*3); % build the line impedance
                                         matrix
for x=1:total_line
    if branchtype(x)==0
        Ym=T*[yz(x,x) 0 0;0 yp(x,x) 0;0 0 yn(x,x)]*S;
    else
        Ym=T*[1/(zr(x)+zx(x)*i) 0 0;0 1/(pr(x)+px(x)*i) 0;0 0
1/(nr(x)+nx(x)*i)]*S;
    end;
    Yline(3*x-2:3*x,3*x-2:3*x)=Ym;
end;
if total_mutual>0 % add mutual couplings
    for x=1:total_mutual
        a=mutualset(x,1);
        b=mutualset(x,2);
        for y=1:total_line
            Ym1=T*[yz(a,y) 0 0;0 yp(a,y) 0;0 0 yn(a,y)]*S;
            Ym2=T*[yz(b,y) 0 0;0 yp(b,y) 0;0 0 yn(b,y)]*S;
            Yline(3*a-2:3*a,3*y-2:3*y)=Ym1;
            Yline(3*b-2:3*b,3*y-2:3*y)=Ym2;
        end;
    end;
end;
Iline=Yline*Vline; % calculate the post-fault branch currents
shuntIline=zeros(total_line*3,2); % calculate the flow in and out
                                   currents of the line
for x=1:total_line
    from=startbus(x);
    to=endbus(x);
    yshunt=T*[zcon(x)+zsup(x)*i 0 0;0 pg(x)+pb(x)*i 0;0 0
ng(x)+nb(x)*i]*S;
    shuntIline(3*x-2:3*x,1)=yshunt*Vshort(from*3-2:from*3);
    shuntIline(3*x-2:3*x,2)=yshunt*Vshort(to*3-2:to*3);
end;
for k=1:total_line
    if SBranchFrom(k,1)==startbus(k) && SBranchTo(k,1)==endbus(k)
I_fault_line(k,faultNode)=abs(Iline(k*3)+shuntIline(k*3,1)+shuntIline(k*3,2)
);
        end;
    end;
end;
end;

```

```

% buildinc.m build the incidence matrices by means of a loop
A=zeros(nLines,nNodes);
for i=1: nLines,
    if S.Branch.From(i,1), A(i,S.Branch.From(i,1))=1; end;
    if S.Branch.To(i,1), A(i,S.Branch.To(i,1))=-1; end;
end;

```

Συνάρτηση τοπολογικής ανάλυσης συστήματος

```

startBus=0;
currentBreaker=0;
currentBus=0;
pairOfCurrentBreaker=0;
searchingLine=0;

```

```

remoteBus=0;
loopsBreakers=[];
loopsBuses=[];
loopsBreakers_br=[];
loopsBuses_br=[];
autos_y=1;
currentLoopLine=1;
currentLoopColumn=1;
eliminatedBus=zeros(1,nNodes);
elBus_y=1;

for startBus=1:(nNodes-1)
    newBreaker=1;
    isBusVisited=0;
    isBusEliminated=0;
    backTrackStart=0;
    runningBreakers=zeros(1,nRelays);
    visitedBus=zeros(1,nNodes);
    autos=zeros();
    autos_y=1;
    rBr_y=1;
    vBu_x=1;
    vBu_y=1;

    %Epilogi zygou
    currentBus=startBus;

    %Euresi prwtou breaker zygou
    for i=1:nLines
        if Numbers(i,currentBus)~=0
            currentBreaker=Numbers(i,currentBus);
            searchingLine=i;
            autos(1,autos_y)=currentBreaker;
            autos_y=autos_y+1;
            break;
        end;
    end;

    while ((newBreaker==1 || (isBusVisited==0 && isBusEliminated==0)) &&
(backTrackStart==0))
        isBusVisited=0;
        isBusEliminated=0;

        %Euresi zeugous breaker
        for i=1:nNodes
            if ((Numbers(searchingLine,i)~=0) &&
(Numbers(searchingLine,i)~=currentBreaker))
                pairOfCurrentBreaker=Numbers(searchingLine,i);
                autos(1,autos_y)=pairOfCurrentBreaker;
                autos_y=autos_y+1;
                break;
            end;
        end;
        %Euresi remote bus pou vlepei o breaker
        for i=1:nNodes
            if
(Numbers(searchingLine,i)~=0) && (Numbers(searchingLine,i)~=currentBreaker)
                remoteBus=i;
                break;
            end;
        end;

        %o remote pou vrika einai o start?
        if (remoteBus~=startBus)
            %vriskw an o remote bus einai visited i eliminated
            for i=1:nNodes
                if (remoteBus==visitedBus(1,i))

```

```

        isBusVisited=1;
        break;
    end;
    if (remoteBus==eliminatedBus(1,i))
        isBusEliminated=1;
        break;
    end;
end;
% An den einai o remote bus visited ouite eliminated
if (isBusVisited==0 && isBusEliminated==0)
    synthiki2=1;

loopsBreakers(currentLoopLine,currentLoopColumn)=currentBreaker;
loopsBuses(currentLoopLine,currentLoopColumn)=currentBus;
currentLoopColumn=currentLoopColumn+1;
runningBreakers(1,rBr_y)=currentBreaker;
rBr_y=rBr_y+1;
if vBu_y==1
    visitedBus(vBu_x,vBu_y)=currentBus;

    vBu_y=vBu_y+1;
else
    if currentBus~=visitedBus(1,vBu_y-1)
        visitedBus(vBu_x,vBu_y)=currentBus;

        vBu_y=vBu_y+1;
    end;
end;
currentBus=remoteBus;

%vriskei ton epomeno breaker
for i=1:nLines
    if
((Numbers(i,currentBus)~=0) && (Numbers(i,currentBus)~=pairOfCurrentBreaker))
        currentBreaker=Numbers(i,currentBus);
        searchingLine=i;
        autos(1,autos_y)=currentBreaker;
        autos_y=autos_y+1;
        break;
    end;
end;
continue;
else
    loopsBreakers(currentLoopLine,currentLoopColumn)=currentBreaker;
    loopsBuses(currentLoopLine,currentLoopColumn)=currentBus;
    for i=1:currentLoopColumn-1

loopsBreakers(currentLoopLine+1,i)=loopsBreakers(currentLoopLine,i);

loopsBuses(currentLoopLine+1,i)=loopsBuses(currentLoopLine,i);
        end
        currentLoopLine=currentLoopLine+1;
        currentLoopColumn=currentLoopColumn;
    end;

    %Euresi allou breaker
    backTrackStart=0;
    newBreaker=0;
    while (backTrackStart==0) && (newBreaker==0)
        f=0;
        l=0;
        while (l<nLines) && (f==0)    %vriskw allon breaker se auto to
                                     zugo
            l=l+1;
            if Numbers(l,currentBus)~=0
                x=0;
                f=1;

```

```

        while (x<autos_y-1) && (f==1) %an exei exetastei o
                                   breaker
            x=x+1;
            if (Numbers(1,currentBus)==autos(1,x))
                f=0;
                break;
            end;
        end;
    end;
end;

if (f==0) %den vrike mi exetasmemo breaker->exoun exetastei oloi
%Backtracked to start?
if (currentBus~=startBus)
    %vriskw ton proigoumeno zugo (auton pou einai prin apo
    auton pou aporrripsame)
    x=1;
    while (x<=nNodes)
        if (visitedBus(1,x)~=currentBus)
            x=x+1;
        else
            break; %to vrike stous visited
        end;
    end;
    if (x>nNodes) %den vrike ton current stous visited
        x=1;
        while (visitedBus(1,x)~=0) %vriskw sto visited to
            teleutaio mi mideniko kai xerw ti thesi tou
            x=x+1;
        end;
    else
        visitedBus(1,x)=0; %svinw ton current apo tous
                           visited
        vBu_y=x;
    end;

    %svisimo stoixeiwn apo tous pinakes running,autos kai loops
    f=0;
    l=0;
    bus=visitedBus(1,vBu_y-1);
    while (l<nLines) && (f==0)
        l=l+1;
        if Numbers(1,bus)~=0 %vrike kapoion breaker
            f=1;
        end;

        i=0;
        g=0;
        while (i<(rBr_y-1)) && (g==0)
            i=i+1;
            if runningBreakers(1,i)==Numbers(1,bus) %ton
                                                        vrike sto running
                g=1;
            end;
        end;

        if (g==1)
            c=0;
            d=0;
            while (c<(autos_y-1)) && (d==0)
                c=c+1;
                if autos(1,c)==Numbers(1,bus)
                    d=1;
                end;
            end;
            for k=(c+1):(autos_y-1) %tous svinei apo ton
                                    autos
                autos(1,k)=0;
            end;
        end;
    end;
end;

```

```

        end;
        autos_y=c+1;

        n=0;
        m=0;
        while n<(currentLoopColumn-1) && (m==0) %to
            vriskei sto loops
            n=n+1;
            if
loopsBreakers(currentLoopLine,n)==Numbers(1,bus)
                m=1;
            end;
        end;

        for t=n:(currentLoopColumn-1)
            loopsBreakers(currentLoopLine,t)=0; %tous
            svinw apo to loops
            loopsBuses(currentLoopLine,t)=0;
        end;
        currentLoopColumn=n;

        for k=i:(rBr_y-1) %tous svinei apo to running
            runningBreakers(1,k)=0;
        end;
        rBr_y=i;
    else
        f=0;

        end;
    end;
    %telos svisimatos stoixeiwn

    currentBus=visitedBus(1,x-1);

    elseif currentBus==startBus
        backTrackStart=1;
        %Eliminate bus from system
        eliminatedBus(1,elBus_y)=currentBus;
        elBus_y=elBus_y+1;
    end;

    elseif f==1 % exei vrei mi exetasmeno breaker
        currentBreaker=Numbers(1,currentBus);
        searchingLine=1;
        autos(1,autos_y)=currentBreaker;
        autos_y=autos_y+1;
        newBreaker=1;
    end;
end;
end;
end;

[x,y]=size(loopsBreakers);

loopsBreakers(x,:)=[]; %svinw tin teleutaia grammi (mideniki)
loopsBuses(x,:)=[];
loopsBreakers(:,y)=[]; %svinw tin teleutaia stili (mideniki)
loopsBuses(:,y)=[];

loopsBreakers_br=loopsBreakers;
loopsBuses_br=loopsBuses;

%-----Euresis breakpoints-RSV
swsti=0;
while swsti==0

```

```

methodos_br=input('Epilexte methodo euresis twn Break Points, apo
vroxous i apo primaries (1 i 2) ');

%euresi twn break points me tin prwti methodo
if methodos_br==1 %
    swsti=1;
    times=zeros(nNodes,1);
    breakpoints=[];
    RSV=[];
    [numberOfLoops,maxNodesinLoop]=size(loopsBuses);
    breakPointsFound=0;
    columnBreakPoints=1;
    columnRSV=1;

    while (breakPointsFound==0)
        %dimiourgia tou pinaka times(se posa loops summetexei kathe
        zugos)
        for i=1:numberofLoops
            c=1;
            while loopsBuses_br(i,c)~=0
                times(loopsBuses_br(i,c))=times(loopsBuses_br(i,c))+1;
                if c<maxNodesinLoop
                    c=c+1;
                else
                    break;
                end;
            end;
        end;

        %euresi tou zugou pou summetexei sta perissotera loops
        pos=1;
        maxtimes=0;
        for i=1:nNodes
            found=0;
            for h=1:numberOfPhantom_B
                if i==phantomBuses(1,h) %elegxei an o bus einai phantom
                    found=1;
                    break;
                end;
            end;
            if (times(i,1)>maxtimes) && (found==0)
                pos=i;
                maxtimes=times(i,1);
            elseif (times(i,1)==maxtimes) && (found==0) %an 2 zugoi
            exoun to idio times, pos thewrei autwn me tous ligoterous breakers
                if BreakersOfNode(i,1)<BreakersOfNode(pos,1)
                    pos=i;
                end;
            end;
        end;

        %euresi twn breakpoints
        for i=1:nLines
            if Numbers(i,pos)~=0

                isPhantom=0;
                for h=1:numberOfPhantom
                    if phantomRelays(1,h)==Numbers(i,pos) %elegxei an o
                    relay einai phantom
                        isPhantom=1;
                        break;
                    end;
                end;
                if (isPhantom==0)
                    breakpoints(1,columnBreakPoints)=Numbers(i,pos);
                    RSV(1,columnRSV)=Numbers(i,pos);

                    c=1;

```

```

        while primaries_RSV(Numbers(i,pos),c)~=0 % svinei
            ti grammi tou apo ton primaries_RSV
            primaries_RSV(Numbers(i,pos),c)=0;
            if c<columnsPrim
                c=c+1;
            end;
        end;
        columnBreakPoints=columnBreakPoints+1;
        columnRSV=columnRSV+1;
    end;
end;

%dimiourgia tou neou pinaka loopsBuses_br (poia loops den
%anoigoun me ta parapanw breakpoints)
breakerFound=0;
again=0;
c=1;
for line=1:numberofLoops
    while (breakerFound==0) && (c<=maxNodesinLoop) &&
(loopsBreakers_br(line,c)~=0)
        for t=1:(columnBreakPoints-1)
            if (loopsBreakers_br(line,c)==breakpoints(1,t))
                breakerFound=1;
                break;
            end;
        end;
        if c<maxNodesinLoop
            c=c+1;
        else
            break;
        end;
    end;
    if (breakerFound==1) %an to loop sti grammi line anoigei me
        ta breakpoints to midenizoume
        c=0;
        while (c<maxNodesinLoop)
            c=c+1;
            loopsBuses_br(line,c)=0;
            loopsBreakers_br(line,c)=0;
        end;
    else %to loop den anoigei kai den einai mideniko ->tha
        xanaginei i diadikasia
        if c>1
            again=again+1;
        end;
    end;
    breakerFound=0;
    c=1;
end;

if (again==0)
    breakPointsFound=1;
end;
times=zeros(nNodes,1);
end;

%dimiourgia tou pinaka RSV
RSV_found=0;
while (RSV_found==0)
    againRSV=0;
    for i=1:nRelays %gia kathe breaker
        c=1;
        primaryFound=1;
        while (primaries_RSV(i,c)~=0) && (c<=columnsPrim) &&
(primaryFound==1) %gia kathe primary tou
            g=0;
            for t=1:(columnRSV-1)

```

```

        if (primaries_RSV(i,c)==RSV(1,t)) %o primary
                                         uparxei ston RSV
            g=1;
            break;
        end;
    end;

    if (g==0) %DEN vrethike o primary
        primaryFound=0;
    end;

    if c<columnsPrim
        c=c+1;
    else
        break;
    end;
end;

if (primaryFound==1) %uparxoun oloi oi primaries ston RSV
    exists=0;
    for h=1:(columnRSV-1)
        if RSV(1,h)==i %elegxei an uparxei o relay ston RSV
            exists=1;
            break;
        end;
    end;
    isPhantom=0;
    for h=1:numberOfPhantom
        if phantomRelays(1,h)==i %elegxei an o relay einai
                                phantom
            isPhantom=1;
            break;
        end;
    end;
    if (exists==0) && (isPhantom==0) %an DEN uparxei
                                    ston RSV kai DEN einai phantom, ton prostheti
        RSV(1,columnRSV)=i;
        columnRSV=columnRSV+1;
        c=1;
    end;
elseif (primaryFound==0)
    againRSV=againRSV+1;
end;
end;
if (againRSV==0)
    RSV_found=1;
end;
end;

%euresi tw'n break points me ti deuteri methodo
elseif methodos_br==2
    swsti=1;
    breakpoints=[];
    RSM=[];
    columnBreakPoints=1;
    columnRSM=1;
    columnRSM_next=1;
    rowRSM=1;
    NumberOfPrim=zeros(nRelays,1); %dimiourgia tou pinaka NumberOfPrim
    for i=1:nRelays
        c=1;
        while primaries(i,c)~=0
            NumberOfPrim(i,1)=NumberOfPrim(i,1)+1;
            c=c+1;
        end;
    end;

    for g=1:numberOfPhantom

```

```

        NumberOfPrim(phantomRelays(1,g),1)=(-1);
end;

found=0;
while found==0;
    relay=1;% euresi tou relay me tous perissoterous primaries
    maxtimes=0;
    examined=0;
    for i=1:nRelays
        if (NumberOfPrim(i,1)>maxtimes)
            relay=i;
            maxtimes=NumberOfPrim(i,1);
        end;
        if NumberOfPrim(i,1)==(-1)
            examined=examined+1;
        end;
    end;

    if examined==nRelays
        break;
    end;

    breakpoints(1,columnBreakPoints)=relay;
    columnBreakPoints=columnBreakPoints+1;
    RSM(rowRSM,columnRSM)=relay;
    columnRSM=columnRSM+1;
    NumberOfPrim(relay,1)=-1;

    for i=1:nRelays %de thewrei primary ton relay
        for c=1:nRelays
            if primaries(i,c)==relay && NumberOfPrim(i,1)~=(-1)
                NumberOfPrim(i,1)=NumberOfPrim(i,1)-1;
            end;
        end;
    end;

    found=1;
    while found==1 %epanalamvanetai mexri na min uparxei relay me
        miden primaries

        found=0;
        for h=1:nRelays %an uparxei relay me miden primaries
            if NumberOfPrim(h,1)==0
                RSM(rowRSM+1,columnRSM_next)=h;
                columnRSM_next=columnRSM_next+1;
                relay=h;
                NumberOfPrim(relay,1)=-1;
                found=1;
                for i=1:nRelays %de thewrei primary ton relay
                    for c=1:nRelays
                        if primaries(i,c)==relay &&
NumberOfPrim(i,1)~=(-1)
                            NumberOfPrim(i,1)=NumberOfPrim(i,1)-1;
                        end;
                    end;
                end;
            end;
        end;
    end;

    %dimiourgia tou dianusmatos RSV apo ton pinaka RSM
    [two,col]=size(RSM);
    i=1;
    while RSM(1,i)~=0
        RSV(1,i)=RSM(1,i);
        i=i+1;
    end;
end;

```

```

end;
column=i;
i=1;
while RSM(2,i)~=0
    RSV(1,column)=RSM(2,i);
    column=column+1;
    if i<col
        i=i+1;
    else
        break;
    end;
end;
columnRSV=column;
else
    disp('Lathos methodos!');
end;
end;
% telos euresis breakpoints-RSV

%-----dimiourgia tou pinaka SSP
lineSSP=1;
for i=1:(columnRSV-1) %diavazei enan breaker apo ton RSV
    c=1;
    while (primaries(RSV(1,i),c)~=0) && (c<=columnsPrim)
        SSP(lineSSP,1)=primaries(RSV(1,i),c);
        SSP(lineSSP,2)=RSV(1,i);
        lineSSP=lineSSP+1;
        if c<columnsPrim
            c=c+1;
        else
            break;
        end;
    end;
end;
end;

```

Αρχεία εισόδου

Το αρχείο εισόδου για την ανάλυση ροών φορτίου στο σύστημα των 14 ζυγών είναι:

```

08/19/93 UW ARCHIVE      100.0 1962 W IEEE 14 Bus Test Case
BUS DATA FOLLOWS      14 ITEMS
1 Bus 1  HV 1 1 3 1.060  0.0  0.0  0.0 232.4 -16.9  69 1.060  0.0  0.0  0.0  0.0  0
2 Bus 2  HV 1 1 2 1.045 -4.98 21.7 12.7  40.0 42.4  69 1.045 50.0 -40.0  0.0  0.0  0
3 Bus 3  HV 1 1 2 1.010 -12.72 94.2 19.0  0.0 23.4  69 1.010 40.0  0.0  0.0  0.0  0
4 Bus 4  HV 1 1 0 1.019 -10.33 47.8 -3.9  0.0  0.0  69 0.0  0.0  0.0  0.0  0.0  0
5 Bus 5  HV 1 1 0 1.020 -8.78  7.6  1.6  0.0  0.0  69 0.0  0.0  0.0  0.0  0.0  0
6 Bus 6  LV 1 1 2 1.070 -14.22 11.2  7.5  0.0 12.2 13.8 1.070 24.0 -6.0  0.0  0.0  0
7 Bus 7  ZV 1 1 0 1.062 -13.37  0.0  0.0  0.0  0.0 13.8  0.0  0.0  0.0  0.0  0.0  0
8 Bus 8  TV 1 1 2 1.090 -13.36  0.0  0.0  0.0 17.4  18 1.090 24.0 -6.0  0.0  0.0  0
9 Bus 9  LV 1 1 0 1.056 -14.94 29.5 16.6  0.0  0.0 13.8  0.0  0.0  0.0  0.0  0.19 0
10 Bus 10 LV 1 1 0 1.051 -15.10  9.0  5.8  0.0  0.0 13.8  0.0  0.0  0.0  0.0  0.0  0
11 Bus 11 LV 1 1 0 1.057 -14.79  3.5  1.8  0.0  0.0 13.8  0.0  0.0  0.0  0.0  0.0  0
12 Bus 12 LV 1 1 0 1.055 -15.07  6.1  1.6  0.0  0.0 13.8  0.0  0.0  0.0  0.0  0.0  0
13 Bus 13 LV 1 1 0 1.050 -15.16 13.5  5.8  0.0  0.0 13.8  0.0  0.0  0.0  0.0  0.0  0
14 Bus 14 LV 1 1 0 1.036 -16.04 14.9  5.0  0.0  0.0 13.8  0.0  0.0  0.0  0.0  0.0  0
-999
BRANCH DATA FOLLOWS      20 ITEMS
1  2 1 1 1 0 0.01938 0.05917 0.0528 0 0 0 0 0 0.0 0.0 0.0 0.0 0.0 0.0
1  5 1 1 1 0 0.05403 0.22304 0.0492 0 0 0 0 0 0.0 0.0 0.0 0.0 0.0 0.0
2  3 1 1 1 0 0.04699 0.19797 0.0438 0 0 0 0 0 0.0 0.0 0.0 0.0 0.0 0.0
2  4 1 1 1 0 0.05811 0.17632 0.0340 0 0 0 0 0 0.0 0.0 0.0 0.0 0.0 0.0
2  5 1 1 1 0 0.05695 0.17388 0.0346 0 0 0 0 0 0.0 0.0 0.0 0.0 0.0 0.0
3  4 1 1 1 0 0.06701 0.17103 0.0128 0 0 0 0 0 0.0 0.0 0.0 0.0 0.0 0.0

```

```

4 5 1 1 1 0 0.01335 0.04211 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0 0.0
4 7 1 1 1 0 0.0 0.20912 0.0 0 0 0 0 0 0.978 0.0 0.0 0.0 0.0
4 9 1 1 1 0 0.0 0.55618 0.0 0 0 0 0 0 0.969 0.0 0.0 0.0 0.0
5 6 1 1 1 0 0.0 0.25202 0.0 0 0 0 0 0 0.932 0.0 0.0 0.0 0.0
6 11 1 1 1 0 0.09498 0.19890 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
6 12 1 1 1 0 0.12291 0.25581 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
6 13 1 1 1 0 0.06615 0.13027 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
7 8 1 1 1 0 0.0 0.17615 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
7 9 1 1 1 0 0.0 0.11001 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
9 10 1 1 1 0 0.03181 0.08450 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
9 14 1 1 1 0 0.12711 0.27038 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
10 11 1 1 1 0 0.08205 0.19207 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
12 13 1 1 1 0 0.22092 0.19988 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
13 14 1 1 1 0 0.17093 0.34802 0.0 0 0 0 0 0 0.0 0.0 0.0 0.0
-999
LOSS ZONES FOLLOWS 1 ITEMS
1 IEEE 14 BUS
-99
INTERCHANGE DATA FOLLOWS 1 ITEMS
1 2 Bus 2 HV 0.0 999.99 IEEE14 IEEE 14 Bus Test Case
-9
TIE LINES FOLLOWS 0 ITEMS
-999
END OF DATA

```

Τα αρχεία εισόδου για την ανάλυση βραχυκυκλωμάτων είναι:

Αρχείο scparam.txt

```

1 100
-99

```

Αρχείο busotp.txt

```

0
1 1.060 0.000 1.060 -120.000 1.060 120.000
2 1.045 0.000 1.045 -120.000 1.045 120.000
3 1.010 0.000 1.010 -120.000 1.010 120.000
4 1.019 0.000 1.019 -120.000 1.019 120.000
5 1.020 0.000 1.020 -120.000 1.020 120.000
6 1.070 0.000 1.070 -120.000 1.070 120.000
7 1.062 0.000 1.062 -120.000 1.062 120.000
8 1.090 0.000 1.090 -120.000 1.090 120.000
9 1.056 0.000 1.056 -120.000 1.056 120.000
10 1.051 0.000 1.051 -120.000 1.051 120.000
11 1.057 0.000 1.057 -120.000 1.057 120.000
12 1.055 0.000 1.055 -120.000 1.055 120.000
13 1.050 0.000 1.050 -120.000 1.050 120.000
14 1.036 0.000 1.036 -120.000 1.036 120.000
-999

```

Αρχείο scinput.txt

```

1 0 1 0.000 0.899 0.000 0.2995
2 0 1 0.000 0.555 0.000 0.1850
3 0 1 0.000 0.555 0.000 0.1850
6 0 1 0.000 0.696 0.000 0.232
8 0 1 0.000 0.696 0.000 0.232
-99
9 0 1 0.000 0.19 0.000 0.19
-99
1 2 21.7 12.7
2 3 94.2 19.0
3 4 47.8 0.0

```

```

4 5 7.6 1.6
5 6 11.2 7.5
6 9 29.5 16.6
7 10 9.0 5.8
8 11 3.5 1.8
9 12 6.1 1.6
10 13 13.5 5.8
11 14 14.9 5.0
-99
1 2 1 1 0 -1 -1 -1 -1 0.00969 0.029585 0.00000 0.0088 0.00969 0.029585 0.00000 0.0264
2 1 2 1 0 -1 -1 -1 -1 0.00969 0.029585 0.00000 0.0088 0.00969 0.029585 0.00000 0.0264
3 5 1 1 0 -1 -1 -1 -1 0.05403 0.22304 0.00000 0.0164 0.05403 0.22304 0.00000 0.0492
4 3 2 1 0 -1 -1 -1 -1 0.04699 0.19797 0.00000 0.0146 0.04699 0.19797 0.00000 0.0438
5 4 2 1 0 -1 -1 -1 -1 0.05811 0.17632 0.00000 0.0125 0.05811 0.17632 0.00000 0.0374
6 2 5 1 0 -1 -1 -1 -1 0.05695 0.17388 0.00000 0.0113 0.05695 0.17388 0.00000 0.0340
7 4 3 1 0 -1 -1 -1 -1 0.06701 0.17103 0.00000 0.0115 0.06701 0.17103 0.00000 0.0346
8 5 4 1 0 -1 -1 -1 -1 0.01335 0.04211 0.00000 0.0043 0.01335 0.04211 0.00000 0.0128
9 6 5 1 0 -1 -1 1 1 0.00000 0.25202 0.00000 0.0000 0.00000 0.25202 0.00000 0.00000
10 9 4 1 0 -1 -1 1 1 0.00000 0.55618 0.00000 0.0000 0.00000 0.55618 0.00000 0.00000
11 12 6 1 0 -1 -1 -1 -1 0.12291 0.25581 0.00000 0.0000 0.12291 0.25581 0.00000 0.00000
12 13 6 1 0 -1 -1 -1 -1 0.06615 0.13027 0.00000 0.0000 0.06615 0.13027 0.00000 0.00000
13 13 12 1 0 -1 -1 -1 -1 0.22092 0.19988 0.00000 0.0000 0.22092 0.19988 0.00000 0.00000
14 11 6 1 0 -1 -1 -1 -1 0.09498 0.19890 0.00000 0.0000 0.09498 0.19890 0.00000 0.00000
15 14 13 1 0 -1 -1 -1 -1 0.17093 0.34802 0.00000 0.0000 0.17093 0.34802 0.00000 0.00000
16 10 11 1 0 -1 -1 -1 -1 0.08205 0.19207 0.00000 0.0000 0.08205 0.19207 0.00000 0.00000
17 9 10 1 0 -1 -1 -1 -1 0.03181 0.08450 0.00000 0.0000 0.03181 0.08450 0.00000 0.00000
18 14 9 1 0 -1 -1 -1 -1 0.12711 0.27038 0.00000 0.0000 0.12711 0.27038 0.00000 0.00000
19 7 9 1 0 -1 -1 -1 -1 0.00000 0.11001 0.00000 0.0000 0.00000 0.11001 0.00000 0.00000
20 7 4 1 0 -1 -1 1 1 0.00000 0.20912 0.00000 0.0000 0.00000 0.20912 0.00000 0.00000
21 8 7 1 0 -1 -1 -1 -1 0.00000 0.17615 0.00000 0.0000 0.00000 0.17615 0.00000 0.00000
-99
-99

```

Αρχείο flinput.txt

```

1 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
2 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
3 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
4 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
5 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
6 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
7 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
8 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
9 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
10 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
11 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
12 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
13 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
14 0.1 0.1 0.1 0.1 0.1 0.1 -1 0
-99

```