



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ**

**Ανάπτυξη εφαρμογής SCADA για την παροχή πρωτεύουσας
ρύθμισης συχνότητας από ενεργά δίκτυα διανομής στο σύστημα
μεταφοράς**

Διπλωματική Εργασία

Χαράλαμπος Βλέτσης

Επιβλέπων: Ελευθέριος Κόντης

Ιούλιος 2025



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ**

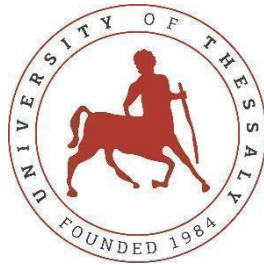
**Ανάπτυξη εφαρμογής SCADA για την παροχή πρωτεύουσας
ρύθμισης συχνότητας από ενεργά δίκτυα διανομής στο σύστημα
μεταφοράς**

Διπλωματική Εργασία

Χαράλαμπος Βλέτσης

Επιβλέπων: Ελευθέριος Κόντης

Ιούλιος 2025



**UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**Development of a SCADA application for the provision of primary
frequency response from active distribution networks to the
transmission system**

Diploma Thesis

Charalampos Vletsis

Supervisor: Eleftherios Kontis

July 2025

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων

Ελευθέριος Κόντης

Επίκουρος Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος

Δημήτριος Μπαργιώτας

Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος

Ασπασία Δασκαλοπούλου

Αναπληρώτρια Καθηγήτρια, Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

**ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ
ΔΙΚΑΙΩΜΑΤΩΝ**

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελούν αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλουν οποιασδήποτε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχουν έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Δηλώνω επίσης ότι τα αποτελέσματα της εργασίας δεν έχουν χρησιμοποιηθεί για την απόκτηση άλλου πτυχίου. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο Δηλών

Βλέτσης Χαράλαμπος

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

The Declarant

Vletsis Charalampos

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας θα ήθελα να ευχαριστήσω όλους όσους συνέβαλαν στην επίτευξη της.

Πρωτίστως θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα της διπλωματικής μου εργασίας, κ. Κόντη Ελευθέριο, για την πολύτιμη βοήθεια του τόσο για την επιλογή του θέματος όσο και για την καθοδήγηση καθ' όλη την διάρκεια της εκπόνησης.

Επίσης, οφείλω ένα μεγάλο ευχαριστώ στην οικογένεια μου που με στήριξε και με εμπιστεύθηκε όλα αυτά τα χρόνια.

Ανάπτυξη εφαρμογής SCADA για την παροχή πρωτεύουσας ρύθμισης συχνότητας από ενεργά δίκτυα διανομής στο σύστημα μεταφοράς

Χαράλαμπος Βλέτσης

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται την ανάπτυξη ενός ολοκληρωμένου Συστήματος Εποπτικού Ελέγχου και Απόκτησης Δεδομένων (SCADA), το οποίο έχει ως κύριο στόχο την αποστολή σημάτων κατανομής προς τις μονάδες Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ) του δικτύου διανομής μέσης τάσης, με σκοπό τη διασφάλιση προκαθορισμένης καμπύλης στατισμού στο σημείο διασύνδεσης με το σύστημα μεταφοράς. Παράλληλα, το σύστημα λειτουργεί ως πλατφόρμα διεπαφής με τον διαχειριστή του δικτύου, παρέχοντας δυνατότητες ρύθμισης παραμέτρων και εποπτείας της λειτουργίας του, όπως η παρακολούθηση τάσεων και ρευμάτων, καθώς και ο εντοπισμός ανεπιθύμητων ή μη επιτρεπτών καταστάσεων. Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε με τη χρήση του λογισμικού «Winlog Eno 4.0» σε συνδυασμό με τη γλώσσα προγραμματισμού «Python». Επιπλέον, αξιοποιήθηκε το προγραμματιστικό περιβάλλον «MATLAB», στο οποίο υλοποιείται διαδικασία βελτιστοποίησης για την παραγωγή των σημάτων κατανομής, με σκοπό την επίτευξη συγκεκριμένης απόκρισης ρύθμισης συχνότητας, λαμβάνοντας υπόψη τα όρια τάσης και θερμικά όρια γραμμών. Επίσης, με την χρήση του λογισμικού «OpenDSS» επιτυγχάνεται η μοντελοποίηση του δικτύου διανομής ηλεκτρικής ενέργειας. Το προτεινόμενο σύστημα αξιολογήθηκε με την βοήθεια προσομοιώσεων σε ένα πρότυπο δίκτυο μέσης τάσης το οποίο φιλοξενεί οικιακά και βιομηχανικά φορτία καθώς και ανεμογεννήτριες.

Development of a SCADA application for the provision of primary frequency response from active distribution networks to the transmission system

Charalampos Vletsis

Abstract

This thesis presents the development of a Supervisory Control and Data Acquisition (SCADA) system, designed to send dispatch signals to Distributed Energy Sources (DER) units within a medium voltage distribution network, to ensure a standard droop curve at the point of interconnection with the transmission system. Additionally, it functions as an interface for the power system operator for parameter configuration and network monitoring, including supervision of voltages and currents as well as detection of abnormal grid conditions. The application was developed using “Winlog Evo 4.0” software and “Python” programming language. Grid optimization is performed in MATLAB, which is used to generate dispatch signals for all DER units of the grid aiming to achieve a specific frequency regulation response while taking into account voltage limits and line thermal constraints. “OpenDSS”, is used for modeling of the electrical distribution network. The proposed system was evaluated with the help of simulations on a medium voltage network that hosts residential and industrial loads as well as wind turbines.

Πίνακας Περιεχομένων

Περίληψη.....	8
Abstract	9
Πίνακας Περιεχομένων	10
Κατάλογος Εικόνων	12
Κατάλογος Τμημάτων Κώδικα.....	13
Κατάλογος Πινάκων	14
Συνοτομογραφίες	15
Κεφάλαιο 1 Εισαγωγή	16
Κεφάλαιο 2 Παροχή Πρωτεύουσας Ρύθμισης Συχνότητας Από τις Μονάδες ΑΠΕ του Δικτύου Διανομής στο Σύστημα Μεταφοράς.....	19
2.1 Εισαγωγή.....	19
2.2 Υλοποίηση καμπύλης στατισμού	20
2.3 Υλοποίηση διαδικασίας βελτιστοποίησης και συνάρτησης στόχου	21
2.4 Πλεονεκτήματα αξιοποίησης της υπηρεσίας.....	23
Κεφάλαιο 3 Ανάλυση Της Λειτουργίας της Προτεινόμενης Εφαρμογής SCADA	25
3.1 Εισαγωγή.....	25
3.2 Λειτουργία του OpenDSS	26
3.3 Λειτουργία του MATLAB	28
3.4 Αρχεία CSV	29
3.5 Λειτουργία της Winlog.....	29
3.5.1 Gates	30
3.5.2 Code	32
3.5.3 Templates.....	38
3.6 Λειτουργία της Python.....	39

Κεφάλαιο 4 Εγχειρίδιο Χρήσης Της Εφαρμογής	48
4.1 Εγκατάσταση Λογισμικών.....	48
4.1.1 Εγκατάσταση του OpenDSS	48
4.1.2 Εγκατάσταση του MATLAB	48
4.1.3 Εγκατάσταση της Winlog Eno 4.0.....	49
4.1.4 Εγκατάσταση της Python	49
4.2 Εγκατάσταση βιβλιοθηκών της Python.....	49
4.3 Προετοιμασία των Λογισμικών και Κώδικα	52
4.4 Εκτέλεση και Λειτουργία της Εφαρμογής	53
Κεφάλαιο 5 Συμπεράσματα	58
Βιβλιογραφία.....	60

Κατάλογος Εικόνων

Εικόνα 1 - Καμπύλη στατισμού	19
Εικόνα 2 - Διασύνδεση Λογισμικών	22
Εικόνα 3 - Μονογραμμικό Διάγραμμα	23
Εικόνα 4 - Γραφικό περιβάλλον Winlog	34
Εικόνα 5 - Python paths	35
Εικόνα 6 - Anaconda Prompt	44
Εικόνα 7 - Anaconda Navigator	44
Εικόνα 8 - Επιλογή μεγιστοποίησης εισαγόμενης ισχύος	47
Εικόνα 9 - Επιλογή ελαχιστοποίησης απωλειών συστήματος	47
Εικόνα 10 - Επιλογή Same Pheads	48
Εικόνα 11 - Επιλογή Proportional Pheads	48
Εικόνα 12 - Επιλογή Standard Pheads	49
Εικόνα 13 - Solve Circuit κουμπί	49
Εικόνα 14 - Περιβάλλον με αποτελέσματα	50
Εικόνα 15 - Διαθέσιμες ενέργειες χειριστή	51

Κατάλογος Τμημάτων Κώδικα

Κώδικας 1 - Υλοποίηση του δικτύου(OpenDSS)	27
Κώδικας 2 - Αρχικοποίηση μεταβλητών της γραφικής παράστασης	32
Κώδικας 3 - Animation Code (Winlog)	33
Κώδικας 4 - Συνάρτηση init (Winlog)	34
Κώδικας 5 – Συνάρτηση Objective_OPF (Winlog)	35
Κώδικας 6 - Συνάρτηση OPF_operation (Winlog)	35
Κώδικας 7 - Συναρτήσεις OPF_select (Winlog)	36
Κώδικας 8 - Συνάρτηση reset (Winlog)	37
Κώδικας 9 - Συνάρτηση turn_off (Winlog)	37
Κώδικας 10 - Συνάρτηση save_data (Winlog)	38
Κώδικας 11 - Συνάρτηση csv_init (Python)	41
Κώδικας 12 - Συνάρτηση phead_calc (Python)	42
Κώδικας 13 - Συνάρτηση objective_opf (Python)	43
Κώδικας 14 - Συνάρτηση loads_export (Python)	44
Κώδικας 15 - Συνάρτηση save_data (Python)	45
Κώδικας 16 - Αυτόματος Έλεγχος (Python)	46

Κατάλογος Πινάκων

Πίνακας 1 - Numeric Gates	25
Πίνακας 2 - Digital Gates	25
Πίνακας 3 - String Gates	26

Συντομογραφίες

1. ΑΠΕ: Ανανεώσιμες Πηγές Ενέργειας
2. ΣΗΕ: Συστήματα Ηλεκτρικής Ενέργειας
3. API: Application Programming Interface (Διασύνδεση Προγραμματισμού Εφαρμογών)
4. DER: Distributed Energy Resource (Κατανεμημένοι Ενεργειακοί Πόροι)
5. DSOs: Distribution System Operators (Διαχειριστές Συστημάτων Διανομής)
6. PFR: Primary Frequency Response (Πρωτεύουσα Ρύθμιση Συχνότητας)
7. SCADA: Supervisory Control And Data Acquisition (Σύστημα Εποπτικού Ελέγχου και Απόκτησης Δεδομένων)

Κεφάλαιο 1 Εισαγωγή

Παραδοσιακά, τα συστήματα διανομής ηλεκτρικής ενέργειας ήταν παθητικά τμήματα των Συστημάτων Ηλεκτρικής Ενέργειας (ΣΗΕ) με ελάχιστες δυνατότητες επίβλεψης και ανάλυσης πραγματικού χρόνου [1]. Αυτή η αρχιτεκτονική όμως, σταδιακά αλλάζει εξαιτίας της εγκατάστασης μονάδων Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ) στα σύγχρονα δίκτυα διανομής. Η αυξημένη διείσδυση μονάδων ΑΠΕ στα σύγχρονα ΣΗΕ δημιουργεί ζητήματα ευστάθειας και ελέγχου συχνότητας και τάσης [2]. Προβλήματα ελέγχου τάσης εμφανίζονται κυρίως εξαιτίας της εγχεόμενης ενεργού ισχύος και της συνεπακόλουθης αντιστροφής στη ροή της ισχύος. Ζητήματα ευστάθειας συχνότητας εμφανίζονται αφενός εξαιτίας της έλλειψης αδράνειας των περισσότερων μονάδων ΑΠΕ και αφετέρου εξαιτίας του γεγονότος πως οι μονάδες ΑΠΕ των δικτύων διανομής δε συμμετέχουν ενεργά στον έλεγχο συχνότητας του ΣΗΕ [3]. Για να εξασφαλιστεί καλή ποιότητα ισχύος και να αντιμετωπιστούν ζητήματα ευστάθειας και ελέγχου συχνότητας θα πρέπει να αναπτυχθούν μέθοδοι ελέγχου των μονάδων ΑΠΕ των δικτύων διανομής, ώστε αυτές να συμβάλουν ενεργά στη διαδικασία εξισορρόπησης του ΣΗΕ [3].

Η ΑΠΕ που αναμένεται να έχει το μεγαλύτερο αντίκτυπο στην ευστάθεια των συστημάτων μεταφοράς είναι τα φωτοβολταϊκά πάνελ, για τα οποία έχουν είδη αναπτυχθεί μέθοδοι για τον έλεγχο της συχνότητας. Πιο συγκεκριμένα, το σύστημα ελέγχου συνεχούς τάσης των φωτοβολταϊκών έχει τροποποιηθεί καταλλήλως ώστε να προκαλεί μια εικονική αδράνεια στο σύστημα. Παρόλα αυτά, για την δημιουργία της εικονικής αδράνειας απαιτείται η χρήση μονάδων αποθήκευσης, όπως πυκνωτές συνεχούς τάσης ή μπαταρίες, με αποτέλεσμα οι δυνατότητες ρύθμισης συχνότητας του συστήματος να μην μεγιστοποιούνται [2]. Οπότε θα πρέπει να εξεταστούν άλλες λύσεις που να μην εξαρτώνται από μηχανισμούς αποθήκευσης ενέργειας και να μπορούν να εφαρμοστούν σε όλα τα συστήματα μεταφοράς ανεξάρτητα από το μέγεθος (μικρή ή μέση τάση) και το είδος των ΑΠΕ που αξιοποιούν. Τέτοιου είδους λύσεις που στοχεύουν σε ομοιόμορφες κατανομές ενέργειάς στο δίκτυο, τόσο αναφορικά με την ενεργό ισχύ όσο και με τους ίδιους τους χρήστες, είναι είδη υπό ανάπτυξη όπως αναλύεται στις [4], [5]. Μια παρόμοιά, αλλά καινοτόμα, πρόταση παροχής πρωτεύουσας ρύθμισης συχνότητας θα μελετηθεί σε αυτή την εργασία, όπως περιγράφεται στην [3].

Στο πρώτο στάδιο της πρότασης ορίζονται τα όρια λειτουργίας των ΑΠΕ, δηλαδή η μέγιστη και η ελάχιστη τιμή της παροχής πρωτεύουσας ρύθμισης στο σημείο διασύνδεσης. Η τιμή αυτή ποσοτικοποιείται με τον ορισμό των καμπυλών χρονισμού στο σημείο διασύνδεσης. Στη συνέχεια μια διαδικασία βελτιστοποίησης διαχειρίζεται τις καμπύλες χρονισμού για κάθε ΑΠΕ, ώστε να παρέχουν τις ζητούμενες ποσότητες ισχύος στο σημείο διασύνδεσης. Στόχος της διαδικασίας είναι η ελαχιστοποίηση μιας συνάρτησης στόχου, αξιοποιώντας παράλληλα με τον βέλτιστο τρόπο τους τεχνικούς περιορισμούς του δικτύου, μεγιστοποιώντας έτσι την εγγεόμενη ισχύς [3].

Παράλληλα με την ανάπτυξη τεχνολογιών ρύθμισης συχνότητας, έγινε γρήγορα αντιληπτό πως θα πρέπει να αναπτυχθούν επίσης νέες τεχνολογίες αυτοματισμού και ελέγχου με σκοπό την καλύτερη επίβλεψη του δικτύου [6]. Με την εισαγωγή των συστημάτων SCADA δημιουργείται η προοπτική της προσομοίωσής σε συνδυασμό με την παρακολούθηση σε πραγματικό χρόνο [7]. Επιπλέον προσφέρουν τις δυνατότητες βελτίωσης της υπηρεσίας, μείωσης του κόστους, επίλυσης προβλημάτων του περιβάλλοντος και βελτιστοποίησης της διαδικασίας καθώς και της αξιοποίησης των διαθέσιμων πόρων [8]. Επιπλέον, ο χρήστης αποκτά την δυνατότητα οπτικοποίησης των δεδομένων, με αποτέλεσμα να έχει καλύτερη αντίληψη των καταστάσεων και των συμβάντων την στιγμή που συμβαίνουν αλλά και σε βάθος χρόνου, καθώς έχει την επιλογή της αποθήκευσης των δεδομένων [9].

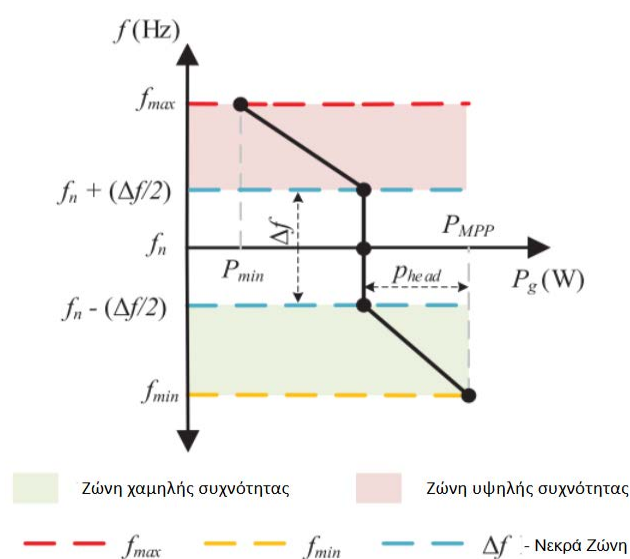
Για τους προαναφερθείς λόγους, στην παρούσα εργασία αναπτύχθηκε μια εφαρμογή SCADA ως ένα διαχειριστικό εργαλείο επίβλεψης της διαδικασίας βελτιστοποίησης για την παροχή πρωτεύουσας ρύθμισης συχνότητας από ενεργά δίκτυα διανομής που αναλύεται στην [3]. Στόχος είναι η συγκέντρωση και προβολή των δεδομένων σε πραγματικό χρόνο λαμβάνοντας υπόψιν τις παραμέτρους που θέτονται από τον χειριστή. Η εφαρμογή SCADA αναπτύχθηκε με την χρήση του λογισμικού Winlog Eno 4.0, στο οποίο γίνεται η προβολή των δεδομένων και η επιλογή των παραμέτρων. Για την εκτέλεση της διαδικασίας βελτιστοποίησης χρησιμοποιείται το προγραμματιστικό περιβάλλον του MATLAB, από όπου παράγονται τα δεδομένα που προβάλλονται από το Winlog Eno 4.0. Επιπλέον, αξιοποιείται το λογισμικό OpenDSS για την μοντελοποίηση και προσομοίωση του ΣΗΕ. Τέλος, για την μεταφορά των δεδομένων από το MATLAB στο γραφικό περιβάλλον του Winlog Eno 4.0 αναπτύχθηκε κώδικας στο προγραμματιστικό περιβάλλον της Python.

Η υπόλοιπη εργασία οργανώνεται ως ακολούθως: Στο Κεφάλαιο 2 αναλύεται η μεθοδολογία που αξιοποιείται για την παροχή πρωτεύουσας ρύθμισης συχνότητας. Ο τρόπος λειτουργίας της εφαρμογής καθώς και οι τεχνικές λεπτομέρειες που αφορούν το κάθε λειτουργικό αναπτύσσονται στο Κεφάλαιο 3. Στο Κεφάλαιο 4 παρατίθεται το εγχειρίδιο χρήσης της εφαρμογής και στο Κεφάλαιο 5 συνοψίζονται τα συμπεράσματα της διπλωματικής εργασίας.

Κεφάλαιο 2 Παροχή Πρωτεύουσας Ρύθμισης Συχνότητας Από τις Μονάδες ΑΠΕ του Δικτύου Διανομής στο Σύστημα Μεταφοράς

2.1 Εισαγωγή

Η παροχή πρωτεύουσας ρύθμισης συχνότητας από ενεργά δίκτυα διανομής περιγράφει την δυνατότητα των μονάδων ΑΠΕ του δικτύου διανομής να μεταβάλλουν την εγγεόμενη ενεργό ισχύ τους, ώστε να προσφέρουν στο σύστημα μεταφοράς ηλεκτρικής ενέργειας ενεργό ισχύ ανάλογα με τη συχνότητα του ΣΗΕ. Η μεταβολή της ενεργού ισχύος πραγματοποιείται με βάση την απόκλιση της συχνότητας του ΣΗΕ από την ονομαστική της τιμή. Πρακτικά αυτό σημαίνει πως οι μονάδες ΑΠΕ, δεν λειτουργούν μόνιμα προσφέροντας τη μέγιστη δυνατή ισχύ τους. Αντίθετα, εγγέουν στο δίκτυο ισχύ μικρότερη από τη μέγιστη δυνατή, ώστε να είναι σε θέση να μεταβάλουν αυτόματα την παραγωγή τους ανάλογα με τη συχνότητα του ΣΗΕ. Έτσι, σε περίπτωση μείωσης της συχνότητας από την ονομαστική της τιμή, οι μονάδες ΑΠΕ μπορούν άμεσα να συνεισφέρουν σε περισσότερη ενεργό ισχύ, στηρίζοντας αποτελεσματικά τη λειτουργία του ΣΗΕ. Η ισχύς που εγγέεται από τις μονάδες ΑΠΕ καθορίζεται με τη βοήθεια καμπυλών στατισμού, οι οποίες περιγράφουν τη σχέση συχνότητας ισχύος. Μια χαρακτηριστική καμπύλη στατισμού μονάδας ΑΠΕ απεικονίζεται στην Εικόνα 1.



Εικόνα 1 - Καμπύλη στατισμού [3]

2.2 Υλοποίηση καμπύλης στατισμού

Η ελάχιστη επιτρεπτή τιμή της συχνότητας στην παρούσα υλοποίηση ορίστηκε στα 49 Hz (f_{min}) και η μέγιστη στα 51 Hz (f_{max}). Την ονομαστική συχνότητα, η οποία είναι στα 50 Hz, περικλείει η νεκρά ζώνη Δf , με τιμή 0.4, στην οποία δεν ενεργοποιείται απόκριση από τις μονάδες ΑΠΕ με αποτέλεσμα η εγχεόμενη ισχύς να παραμένει σταθερή. Με βάση τα παραπάνω δεδομένα ορίζονται οι τρεις παρακάτω ζώνες λειτουργίας των ΑΠΕ:

1. Ζώνη κανονικής λειτουργίας: $f_n - \frac{\Delta f}{2} \leq f_n \leq f_n + \frac{\Delta f}{2}$
2. Ζώνη χαμηλής συχνότητας: $f_{min} \leq f_n < f_n - \frac{\Delta f}{2}$
3. Ζώνη υψηλής συχνότητας: $f_n + \frac{\Delta f}{2} < f_n \leq f_{max}$

Για κάθε μια από τις παραπάνω ζώνες, η ενεργός ισχύς που εγχέεται από την εκάστοτε μονάδα ΑΠΕ, είναι όπως φαίνεται και στην Εικόνα 1, ίση με:

1. Ζώνη κανονικής λειτουργίας: $P_k = P_c = P_{MPP,k}(1 - p_{h,k})$ (1)

Στη ζώνη αυτή η εγχεόμενη ισχύς έχει σταθερή τιμή. Η μεταβλητή $p_{h,k}$ δηλώνει το headroom της κάθε μονάδας (Εικόνα 1).

2. Ζώνη χαμηλής συχνότητας: $P_k = P_{MPP,k} + \frac{P_{MPP,k}[p_{h,k}(f - f_{min})]}{f_{min} + \frac{\Delta f}{2} - f_n}$ (2)

Στη ζώνη αυτή η ισχύς περιγράφεται από μια γραμμική συνάρτηση που ξεκινά από το ακρότατο σημείο της ζώνης κανονικής λειτουργίας, όπου η συχνότητα είναι ελάχιστη της ζώνης ($f_n - \frac{\Delta f}{2}$), και καταλήγει στο σημείο όπου η συχνότητα έχει την ελάχιστη επιτρεπτή τιμή και η ισχύς είναι η μέγιστη (P_{MPP}).

3. Ζώνη υψηλής συχνότητας: $P_k = \frac{P_c(f - f_{max}) + P_{min,k}(f_n + \frac{\Delta f}{2} - f)}{f_n + \frac{\Delta f}{2} - f_{max}}$ (3)

Στη ζώνη αυτή η ισχύς περιγράφεται από μια γραμμική συνάρτηση που ξεκινά από το σημείο όπου η συχνότητα έχει την μέγιστη επιτρεπτή τιμή και η ισχύς είναι η ελάχιστη (P_{min}) και καταλήγει ακρότατο σημείο της ζώνης κανονικής λειτουργίας, όπου η συχνότητα είναι μέγιστη της ζώνης ($f_n + \frac{\Delta f}{2}$).

Όπου $P_{MPP,k}$ είναι η μέγιστη τιμή ισχύος που μπορεί να προσφέρει η k-οστή ΑΠΕ, P_c είναι η τιμή ισχύος που μπορεί να προσφέρει η k-οστή για ονομαστική συχνότητα και $P_{min,k}$ η ελάχιστη ισχύς που μπορεί να προσφέρει η k-οστή ΑΠΕ.

2.3 Υλοποίηση διαδικασίας βελτιστοποίησης και συνάρτησης στόχου

Για την μοντελοποίηση του δικτύου χρησιμοποιήθηκε ένας γράφος $G = (V, E)$, όπου $V = \{0, 1, \dots, p\}$ είναι ο αριθμός των κόμβων. Έτσι το σύνολο των γραμμών ορίζεται ως $E \subseteq V \times V$. Επιπλέον, η $N_i := \{j: (j, i), (i, j) \in E\}$ ορίζει τους κόμβους που συνδέονται με τον κόμβο i και $D_i = \{1, 2, \dots, d_i\}$ είναι το σύνολο των μονάδων ΑΠΕ στον κόμβο. Τέλος, θεωρείται πως ο μετασχηματιστής τοποθετείται ανάμεσα στους κόμβους 0 και 1, άρα ο κόμβος 0 ανήκει στην πλευρά της υψηλής τάσης του μετασχηματιστή και ο κόμβος 1 είναι ο πρώτος από την πλευρά της μέσης τάσης.

Για την μοντελοποίηση του δικτύου ισχύουν:

$$\sum_{k \in D_i} S_k - S_i^L = \sqrt{3U_i I_i^*}, \quad \forall i \in V \quad (4)$$

$$\sum_{j \in N_i} I_{i,j} = I_i, \quad \forall i \in V, j \in N_i \quad (5)$$

$$U_i - U_j = Z_{i,j} I_{i,j}, \quad \forall i \in V, j \in N_i \quad (6)$$

Στον κόμβο i , S_k είναι η συνολική παραγόμενη ισχύς της k-οστής ΑΠΕ και S_i^L είναι η συνολική κατανάλωση. Επιπλέον, το U_i αντιπροσωπεύει τη φασική τάση και το ρεύμα στον κόμβο i αντίστοιχα. Το $I_{i,j}$ αντιπροσωπεύει το ρεύμα που διαρρέει την γραμμή (i, j) και $Z_{i,j}$ είναι η αντίσταση της αντίστοιχης γραμμής. Τέλος, πρέπει να αναφερθεί ότι η ενεργός ισχύς που εγχέεται από την k-οστή ΑΠΕ εξαρτάται από την συχνότητα του δικτύου και υπολογίζεται από τις συναρτήσεις (1), (2) και (3).

Τα όρια λειτουργίας για τις επιτρεπόμενες τάσεις και τα επιτρεπόμενα ρεύματα περιγράφονται από τις δυο παρακάτω ανισότητες.

$$|I_{i,j}| \leq I_{i,j}^{max}, \quad \forall i \in V, j \in N_i \quad (7)$$

$$V^{min} \leq |U_i| \leq V^{max}, \quad \forall i \in V \quad (8)$$

Όπου $I_{i,j}^{max}$ είναι το μέγιστο θερμικό όριο της κάθε γραμμής και V^{min}, V^{max} είναι η ελάχιστη και η μέγιστη επιτρεπτή τάση αντίστοιχα.

Η συνάρτηση βελτιστοποίησης ορίζεται ως:

$$(\underline{P}^{PFR}, \underline{Q}^{PFR}) = \arg \min C_{(\underline{P}, \underline{Q})} \quad (9)$$

Όπου ο δείκτης PFR δηλώνει την τιμή των εγγεόμενων ισχύων των μονάδων ΑΠΕ που ελαχιστοποιούν την (9) και το C ορίζεται ως:

$$C = w_1 P_{loss}^{line} + w_2 C^d \quad (10)$$

Το C^d είναι η συνολική αμοιβή των ΑΠΕ και P_{loss}^{line} είναι οι συνολικές απώλειες ισχύος στις γραμμές του δικτύου. Οι μεταβλητές w_1 και w_2 είναι βάρη που μπορούν να χρησιμοποιηθούν από τον διαχειριστή του συστήματος για να δοθεί διαφορετική βαρύτητα στις επιμέρους συνιστώσες της συνάρτησης κόστους.

Οι απώλειες, P_{loss}^{line} ορίζονται ως εξής:

$$P_{loss}^{line} = \sum_{i,j \in E} R_{i,j} (I_{i,j})^2 \quad (11)$$

Με το $R_{i,j}$ να δηλώνει την αντίσταση της γραμμής. Επίσης, το C^d ορίζεται ως:

$$C_d = \sum_{k \in D_i, i \in V} (P_{MPP,k} - P_k) \quad (12)$$

Όπως φαίνεται, η μεγιστοποίηση του κέρδους C_d επιτυγχάνεται όταν οι μονάδες εγχέουν τη μέγιστη δυνατή ισχύ τους.

Τέλος, η ενεργός ισχύς που πρέπει να δοθεί στο σημείο διασύνδεσης ορίζεται ως:

$$P_0 = ct \quad (13)$$

Σε περίπτωση που θέλουμε οι μονάδες ΑΠΕ να δουλεύουν με το ίδιο headroom, τότε προστίθεται ακόμα ο περιορισμός:

$$p_{h,k} = ct \quad \forall k \quad (14)$$

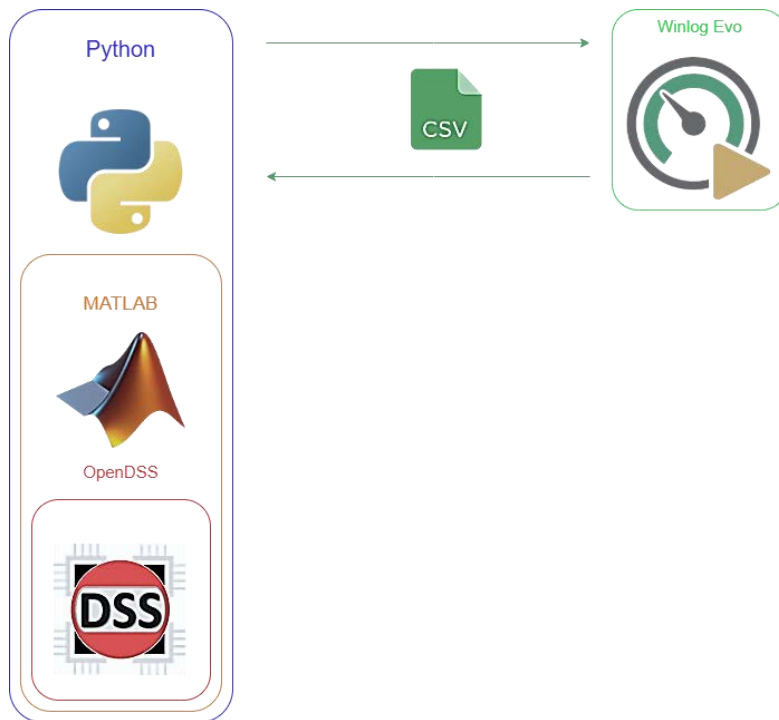
2.4 Πλεονεκτήματα αξιοποίησης της υπηρεσίας

Βασικός στόχος της υπηρεσίας είναι η παροχή πρωτεύουσας ρύθμισης συχνότητας στο σημείο διασύνδεσης με το σύστημα μεταφοράς. Η λειτουργία της υπηρεσίας βασίζεται σε διαδικασίες βελτιστοποίησης οι οποίες αξιοποιούν πλήρως τις δυνατότητες των ΑΠΕ, συνυπολογίζοντας παράλληλα και τεχνικούς περιορισμούς του δικτύου, όπως τα θερμικά όρια των γραμμών, τα όρια των τάσεων και της παραγόμενης ισχύος. Επιπλέον, παρέχεται η δυνατότητα προσαρμογής της υπηρεσίας με την αξιοποίηση της συνάρτησης στόχου. Αναλυτικότερα προσφέρονται δύο επιλογές, η πρώτη στοχεύει στην μεγιστοποίηση της εγχεόμενης ισχύος εντός των δυναμικών περιορισμών του συστήματος και η δεύτερη στοχεύει στην ελαχιστοποίηση των απωλειών του συστήματος. Η ευελιξία που προσφέρουν οι συναρτήσεις επιτρέπει την κάλυψη μεγαλύτερου εύρους λειτουργικών σεναρίων.

Κεφάλαιο 3 Ανάλυση Της Λειτουργίας της Προτεινόμενης Εφαρμογής SCADA

3.1 Εισαγωγή

Η λειτουργία της προτεινόμενης εφαρμογής βασίζεται κατά κύριο λόγο στο λογισμικό Winlog Eno 4.0 [10]. Επιπλέον, για την ανάπτυξη της αξιοποιήθηκαν τα προγραμματιστικά περιβάλλοντά Python [11], MATLAB [12] και OpenDSS [13]. Η διασύνδεση τους έγινε όπως φαίνεται στην Εικόνα 2. Όπως φαίνεται και στην εικόνα η εφαρμογή Winlog λαμβάνει και αποστέλλει δεδομένα στην Python μέσω CSV αρχείων. Η Python για να αποθηκεύσει τα δεδομένα στα CSV καλεί το περιβάλλον του MATLAB μέσω ενός application programming interface (API), το οποίο με την σειρά του καλεί ένα API του OpenDSS. Η ακριβής λειτουργία κάθε εφαρμογής αναλύεται εκτενέστερα στη συνέχεια αυτού του κεφαλαίου.



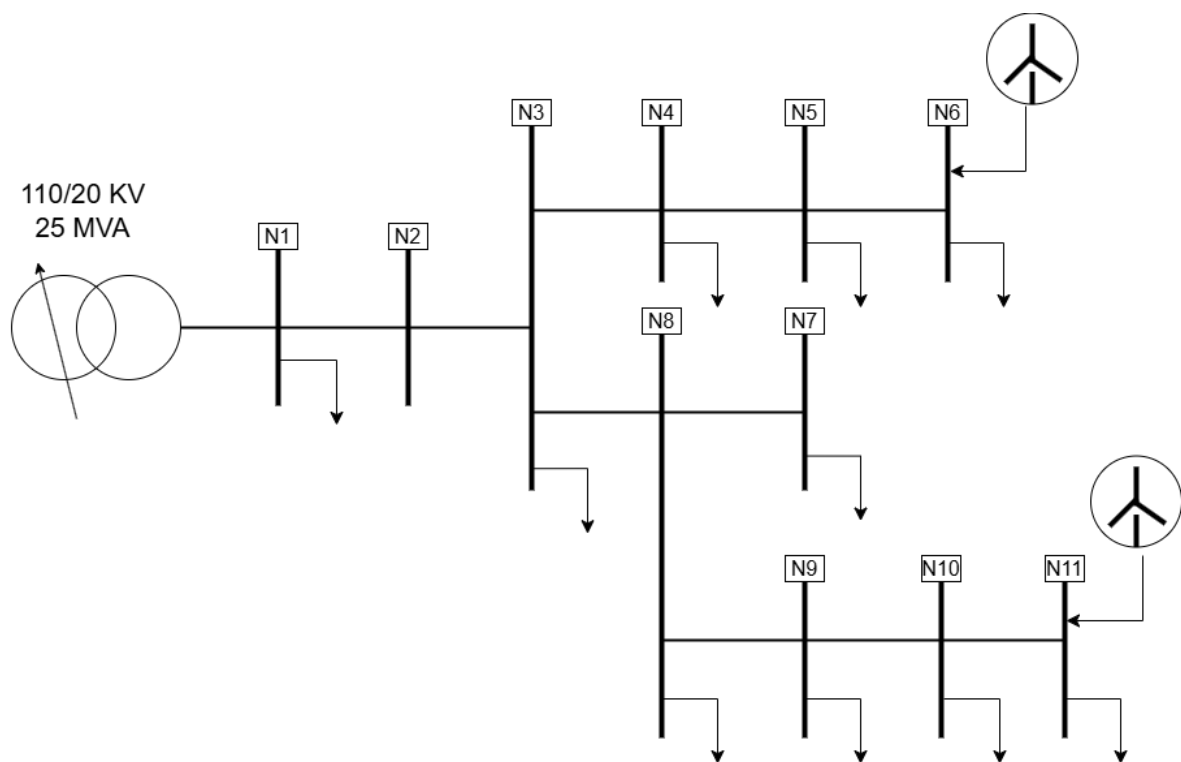
Εικόνα 2 - Διασύνδεση Λογισμικών

3.2 Λειτουργία του OpenDSS

Στην παρούσα εργασία το λειτουργικό περιβάλλον του OpenDSS χρησιμοποιείται για την προσομοίωση και την ανάλυση του δικτύου διανομής. Συγκεκριμένα, ο κώδικας που αξιοποιήθηκε σε αυτή την εργασία περιγράφει το κύκλωμα μέσης τάσης που φαίνεται στην Εικόνα 3, το οποίο τροφοδοτείται μέσω ενός μετασχηματιστή (110kV/20kV) συνδεσμολογίας Dyn5. Το εν λόγω κύκλωμα περιέχει δυο ανεμογεννήτριες, στους ζυγούς έξι και έντεκα, και δεκατρία φορτία μοιρασμένα από τον ζυγό ένα έως και τον τελευταίο ζυγό που είναι ο ζυγός έντεκα. Αναλυτικότερα:

- Οι ζυγοί ένα, τρία και δέκα έχουν διπλά φορτία, το βιομηχανικό και το οικιακό.
- Οι ζυγοί τέσσερα, πέντε, έξι, οκτώ και έντεκα έχουν μόνο οικιακό φορτίο.
- Οι ζυγοί επτά και εννέα έχουν μόνο βιομηχανικό.

Ο διαχωρισμός των φορτίων σε αυτές τις δύο κατηγορίες (οικιακό και βιομηχανικό) γίνεται προκειμένου να προσομοιάζει καλύτερα ένα κοινό κύκλωμα μέσης τάσης. Επιπλέον, για τις γραμμές του δικτύου έχουν οριστεί τα τεχνικά του στοιχεία, όπως το μήκος τους σε χιλιόμετρα, ο αριθμός των φάσεων και η επαγωγική αντίσταση τους, όπως φαίνεται στις γραμμές 6 έως 15 του Κώδικα 1.



Εικόνα 3 - Μονογραμμικό Διάγραμμα

Στον κώδικα του OpenDSS λοιπόν περιγράφονται οι δύο μονάδες ΑΠΕ, τα δεκατρία φορτία, ο μετασχηματιστής που μεταφέρει την ενέργεια στο κύκλωμα καθώς και οι δέκα γραμμές που ενώνουν τα στοιχεία του κυκλώματος. Σκοπός του αρχείου που περιγράφεται στην Κώδικα 1 είναι η προσομοίωση του δικτύου μέσης τάσης με στόχο την εξαγωγή δεδομένων για την ροή της ισχύος, των απωλειών του συστήματος και την ρύθμιση της τάσης. Η διαδικασία αυτή μπορεί να πραγματοποιηθεί χειροκίνητα από το λογισμικό του OpenDSS, αλλά στην παρούσα υλοποίηση καλείται και εκτελείται από το περιβάλλον του MATLAB όπως περιγράφεται στο επόμενο υποκεφάλαιο.

```

clear
set DefaultBaseFrequency=50

new circuit.name bus1=n0 baseKv=110 angle=0 frequency=50 phases=3 MVAsc1=5000000 MVAsc3=5000000 x1r1=10 x0r0=10

new line.line_10_11 bus1=n10 bus2=n11 length=0.33 phases=3 units=km R1=0.501 X1=0.716 C1=0 C0=0 Rg=0 Xg=0
new line.line_1_2 bus1=n1 bus2=n2 length=2.82 phases=3 units=km R1=0.501 X1=0.716 C1=0 C0=0 Rg=0 Xg=0
new line.line_2_3 bus1=n2 bus2=n3 length=4.42 phases=3 units=km R1=0.501 X1=0.716 C1=0 C0=0 Rg=0 Xg=0
new line.line_3_4 bus1=n3 bus2=n4 length=0.61 phases=3 units=km R1=0.501 X1=0.716 C1=0 C0=0 Rg=0 Xg=0
new line.line_3_8 bus1=n8 bus2=n3 length=1.3 phases=3 units=km R1=0.501 X1=0.716 C1=0 C0=0 Rg=0 Xg=0
new line.line_4_5 bus1=n4 bus2=n5 length=0.56 phases=3 units=km R1=0.501 X1=0.716 C1=0 C0=0 Rg=0 Xg=0
new line.line_5_6 bus1=n5 bus2=n6 length=1.54 phases=3 units=km R1=0.501 X1=0.716 C1=0 C0=0 Rg=0 Xg=0
new line.line_7_8 bus1=n8 bus2=n7 length=1.67 phases=3 units=km R1=0.501 X1=0.716 C1=0 C0=0 Rg=0 Xg=0
new line.line_8_9 bus1=n8 bus2=n9 length=0.32 phases=3 units=km R1=0.501 X1=0.716 C1=0 C0=0 Rg=0 Xg=0
new line.line_9_10 bus1=n9 bus2=n10 length=0.77 phases=3 units=km R1=0.501 X1=0.716 C1=0 C0=0 Rg=0 Xg=0

new transformer.tr1 phases=3 windings=2
~ buses=[n0,n1]
~ conns=[delta,wy]
~ kvs=[110,20]
~ kvas=[25000,25000]
~ taps=[1,1.0625]
~ Xhl=12 %loadloss=0 %noloadloss=0 Sub=Yes
~ Maxtap=1.1 MinTap=0.9 NumTap=32 %imag=0

new load.l10_ind bus1=n10 phases=3 kV=20 kw=68 kvar=42.1 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l10_res bus1=n10 phases=3 kV=20 kw=475.3 kvar=119 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l11_res bus1=n11 phases=3 kV=20 kw=329.8 kvar=82.7 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l1_ind bus1=n1 phases=3 kV=20 kw=0 kvar=0 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l1_res bus1=n1 phases=3 kV=20 kw=0 kvar=0 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l3_ind bus1=n3 phases=3 kV=20 kw=225.25 kvar=140 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l3_res bus1=n3 phases=3 kV=20 kw=276.45 kvar=69.3 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l4_res bus1=n4 phases=3 kV=20 kw=431.65 kvar=108 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l5_res bus1=n5 phases=3 kV=20 kw=727.5 kvar=182 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l6_res bus1=n6 phases=3 kV=20 kw=548.0499 kvar=137 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l7_ind bus1=n7 phases=3 kV=20 kw=76.5 kvar=47.4 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l8_res bus1=n8 phases=3 kV=20 kw=586.8501 kvar=147 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2
new load.l9_ind bus1=n9 phases=3 kV=20 kw=573.75 kvar=356 model=4 CVRwatts=0 CVRvars=0 status=fixed Vminpu=0.1 Vmaxpu=2

new generator.gen1 bus1=n6 phases=3 kV=20 kw=435.5208263 kvar=0 model=1 status=fixed Vminpu=0.1 Vmaxpu=2
new generator.gen2 bus1=n11 phases=3 kV=20 kw=649.999999 kvar=0 model=1 status=fixed Vminpu=0.1 Vmaxpu=2

set Algorithm=Normal
set maxiterations=150
set tolerance=0.0001
set VoltageBases=[110 20]
calc voltagebases

```

Κώδικας 1 - Υλοποίηση του δικτύου(OpenDSS)

3.3 Λειτουργία του MATLAB

Στο προγραμματιστικό περιβάλλον του MATLAB υλοποιείται η διαδικασία βελτιστοποίησης που αναφέρθηκε στο Κεφάλαιο 2.3. Η διαδικασία αυτή αφορά την βελτιστοποίηση της λειτουργίας των ΑΠΕ ενός δικτύου, δεδομένων των τεχνικών περιορισμών και της συνάρτησης στόχου που επιλέγεται από τον χειριστή. Οι τεχνικοί περιορισμοί αφορούν τα όρια των τάσεων, των ρευμάτων και της εγχεόμενης ισχύος και αφορούν όλες τις υποπεριπτώσεις που θα αναλυθούν παρακάτω. Η επιλογή του χρήστη αφορά τον στόχο της βελτιστοποίησης στο σύστημα. Η πρώτη είναι η ελαχιστοποίηση των απωλειών του συστήματος και η δεύτερη είναι η μεγιστοποίηση της εγχεόμενης ισχύος στο σύστημα και άρα του κέρδους των παραγωγών. Η επιλογή αυτή γίνεται από τον χειριστή κατά την χρήση της εφαρμογής και αποθηκεύεται στο αρχείο «Objective_OPF.m» μέσω της Python όπως περιγράφεται στο Κεφάλαιο 3.6.

Η διαδικασία βελτιστοποίησης υλοποιείται με βάση ένα από τα εξής τρία διαφορετικά μοντέλα τα οποία αξιοποιούνται στα πλαίσια αυτής της εφαρμογής, `Main_OPF`, `Main_OPF_proportional_Pheads` και `Main_OPF_same_Pheads`. Ο όρος `Phead` χρησιμοποιείται σε κάθε μια από υποπεριπτώσεις και δηλώνει την ποσοστιαία ποσότητα ισχύος η οποία αποκόπτεται από την κάθε ΑΠΕ (δηλώνει, δηλαδή, το headroom που αναλύθηκε στο Κεφάλαιο 2). Στην πρώτη περίπτωση η διαδικασία βελτιστοποίησης λαμβάνει υπόψιν μόνο τους τεχνικούς περιορισμούς του δικτύου και υπολογίζει δυο `Phead` (ένα για κάθε ανεμογεννήτρια). Στη περίπτωση των αναλογικών `Phead` προστίθεται μια επιπλέον παράμετρος στην διαδικασία, η οποία την αναγκάζει να περιορίζει τις ΑΠΕ αναλογικά με το μέγεθος τους. Οι τιμές των `Phead` εξακολουθούν να είναι ευέλικτες και άνισες όπως και στη γενική περίπτωση αλλά πλέον είναι πιο δίκαιο το μοίρασμα της αποκοπής, επεκτείνοντας τις εμπορικές εφαρμογές της διαδικασίας. Τέλος, στην πιο αυστηρή περίπτωση των ίδιων `Phead` έχουμε ίση αποκοπή από όλες τις ΑΠΕ του δικτύου. Το μοντέλο αυτό θυσιάζοντας κομμάτι της ευελιξίας της διαδικασίας εξασφαλίζεται η ίση συνεισφορά όλων των ΑΠΕ στην εγχεόμενη ισχύ.

Η επιλογή για το μοντέλο που θα χρησιμοποιηθεί γίνεται από το χειριστή μέσω του γραφικού περιβάλλοντος της εφαρμογής όπως περιγράφεται στο Κεφάλαιο 3.5 και αποθηκεύεται στο αρχείο «`Phead_calc.m`» από την Python παρόμοια με το «`Objective_OPF.m`».

3.4 Αρχεία CSV

Για την επικοινωνία μεταξύ της Winlog και της Python πρέπει να δημιουργηθούν δυο αρχεία με κατάληξη `.csv`. Το πρώτο λέγεται `Winlog_output.csv` και περιέχει δύο στήλες, την `operations` και την `digits`. Η πρώτη περιέχει τα ονόματα `objective_OPF`, `OPF_op`, `Calc_load`, `Reset`, `Turn_off` και `Save_Data`, τα οποία αντιπροσωπεύουν τις έξι διαφορετικές επιλογές που έχει ο χειριστής στο περιβάλλον της εφαρμογής. Κάθε μια από τις επιλογές αναλύεται στο Κεφάλαιο 3.5. Η δεύτερη στήλη περιέχει την τιμή για κάθε μια από τις επιλογές, η οποία είναι αρχικοποιημένη σε μηδέν. Το «0» είναι η αρχική τιμή για όλες τις μεταβλητές και αντιπροσωπεύει την αρχική τους κατάσταση όταν ανοίγει η εφαρμογή προτού ο χειριστής κάνει οποιαδήποτε αλλαγή. Ο τρόπος με τον οποίο γίνονται οι αλλαγές αναλύεται στο Κεφάλαιο 3.5.3 και το πως αξιοποιούνται στα κεφάλαια 3.5.2 και 3.6. Στην πράξη το αρχείο αυτό είναι υπεύθυνο για την μεταφορά της πληροφορίας, από την Winlog στην Python, για

τις επιλογές που κάνει η χειριστής. Το δεύτερο αρχείο λέγεται Winlog_input.txt και παρότι έχει κατάληξη .txt διαχειρίζεται και από τις δύο εφαρμογές σαν .csv αρχείο. Ο λόγος για τον οποίο επιλέχθηκε η κατάληξη .txt είναι επειδή όλες οι κατηγορίες μεταβλητών ακολουθούν μια προκαθορισμένη μορφή (π.χ. οι τάσεις έχουν δύο ψηφία πριν την υποδιαστολή και τέσσερα μετά την υποδιαστολή), την οποία τα αρχεία με κατάληξη .csv δεν ακολουθούν όταν τα ψηφία μετά την υποδιαστολή είναι «0». Η αυστηρότητα αυτή ως προς την μορφή των δεδομένων αφορά τον τρόπο που διαβάζει τα αρχεία η Winlog και περιγράφεται αναλυτικότερα στο Κεφάλαιο 3.5.2. Το αρχείο αυτό αποσκοπεί στην αποθήκευση των δεδομένων που παράγει το MATLAB μέσω της Python, όπως περιγράφεται στο Κεφάλαιο 3.6, και την ανάγνωση τους από την Winlog, όπως περιγράφεται στο Κεφάλαιο 3.5.2.

3.5 Λειτουργία της Winlog

Η λειτουργία του λειτουργικού συστήματος της Winlog βασίζεται κυρίως σε ένα ενσωματωμένο προγραμματιστικό περιβάλλον (Code) και σε ένα περιβάλλον δημιουργίας προτύπων (Templates), τα οποία αποτελούν το γραφικό περιβάλλον με το οποίο αλληλοεπιδρά ο χειριστής. Επιπλέον, προσφέρει ξεχωριστό πεδίο ώστε να δηλώνονται οι πύλες (Gates), οι οποίες λειτουργούν σαν μεταβλητές.

3.5.1 Gates

Το πεδίο που δηλώνονται οι πύλες χωρίζεται σε πέντε υποκατηγορίες: Numeric, Digital, Compound, String και Event and Alarm. Στην παρούσα υλοποίηση αξιοποιήθηκαν μόνο τρεις κατηγορίες, η Numeric, η Digital και η String. Ο τρόπος με τον οποίο αξιοποιείται η κάθε πύλη θα αναλυθεί παρακάτω παράλληλα με την επεξήγηση του κώδικα στο Κεφάλαιο 3.5.2, εδώ επεξηγείται μόνο η δήλωση τους. Στην κατηγορία Numeric δηλώθηκαν συνολικά οκτώ μεταβλητές όπως φαίνονται στον Πίνακα 1, με την χρησιμότητα της κάθε μίας να φαίνεται στο Description.

Πίνακας 1 - Numeric Gates

	Channel	Device	Gate ID	N ID	Address	Description	Measure	Variable type	Tolerance	Min. value	Max. value	Start value
1	0	0	fmax	2		This is the gate for the maximum frequency at the point of interest.		DOUBLE		0	50.2	0
2	0	0	fmin	2		This is the gate for the minimum frequency at the point of interest.		DOUBLE		0	49.8	0
3	0	0	fmax	1		This is the gate for the maximum frequency.		DOUBLE		0	51	0
4	0	0	fmin	1		This is the gate for the minimum frequency.		DOUBLE		0	49	0.0
5	0	0	Pfeeder_max	1		This is the gate for the maximum value for Pfeeder.		DOUBLE		-9000	9000	0.0
6	0	0	Pfeeder_min	1		This is the gate for the minimum value for the Pfeeder.		DOUBLE		-9000	9000	0.0
7	0	0	Pfeeder	1		This is the gate for the FIRST droop curve.		DOUBLE		-9000	9000	0.0
8	0	0	OPF_select	1		This gate selects the OPF function that will be called.		DOUBLE		1	3	1

Αντίστοιχα στην κατηγορία Digital δηλώθηκαν οι παρακάτω δύο μεταβλητές που φαίνονται στον Πίνακα 2.

Πίνακας 2 - Digital Gates

	Channel	Device	Gate ID	N ID	Address	Description	Start value
1	0	0	Droop_Curve_LED	1		This is the gate that controls the LED above the Droop Curve diagram.	0
2	0	0	Objective_OPF	1		This Gate represents the switch for the Objective OPF	0

Τέλος, στην κατηγορία String δηλώθηκαν όλες οι μεταβλητές που αφορούν τις τάσεις, τα ρεύματα, τις ενεργούς και άεργους ισχύς, όπως φαίνεται στον Πίνακα 3.

Πίνακας 3 - String Gates

	Channel	Device	Gate ID	N ID	Address	Description	Max dimension	Start value
1	0	0	Currents	10		This is the gate for the line 9_10.	20	0.0
2	0	0	Currents	9		This is the gate for the line 8_9.	20	0.0
3	0	0	Currents	8		This is the gate for the line 7_8.	20	0.0
4	0	0	Currents	7		This is the gate for the line 5_6.	20	0.0
5	0	0	Currents	6		This is the gate for the line 4_5.	20	0.0
6	0	0	Currents	5		This is the gate for the line 3_8.	20	0.0
7	0	0	Currents	4		This is the gate for the line 3_4.	20	0.0
8	0	0	Currents	3		This is the gate for the line 2_3.	20	0.0
9	0	0	Currents	2		This is the gate for the line 1_2.	20	0.0
10	0	0	Currents	1		This is the gate for the line 10_11.	20	0.0
11	0	0	Voltages	11		This is the gate for Bus11.	20	0.0
12	0	0	Voltages	10		This is the gate for Bus10.	20	0.0
13	0	0	Voltages	9		This is the gate for Bus9.	20	0.0
14	0	0	Voltages	8		This is the gate for Bus8.	20	0.0
15	0	0	Voltages	7		This is the gate for Bus7.	20	0.0
16	0	0	Voltages	6		This is the gate for Bus6.	20	0.0
17	0	0	Voltages	5		This is the gate for Bus5.	20	0.0
18	0	0	Voltages	4		This is the gate for Bus4.	20	0.0
19	0	0	Voltages	3		This is the gate for Bus3.	20	0.0
20	0	0	Voltages	2		This is the gate for Bus2.	20	0.0
21	0	0	Voltages	1		This is the gate for Bus1.	20	0.0
22	0	0	Voltage_pu	11		This is the gate in pu for Bus11.	20	0.0
23	0	0	Voltage_pu	10		This is the gate in pu for Bus10.	20	0.0
24	0	0	Voltage_pu	9		This is the gate in pu for Bus9.	20	0.0
25	0	0	Voltage_pu	8		This is the gate in pu for Bus8.	20	0.0
26	0	0	Voltage_pu	7		This is the gate in pu for Bus7.	20	0.0
27	0	0	Voltage_pu	6		This is the gate in pu for Bus6.	20	0.0
28	0	0	Voltage_pu	5		This is the gate in pu for Bus5.	20	0.0
29	0	0	Voltage_pu	4		This is the gate in pu for Bus4.	20	0.0
30	0	0	Voltage_pu	3		This is the gate in pu for Bus3.	20	0.0
31	0	0	Voltage_pu	2		This is the gate in pu for Bus2.	20	0.0
32	0	0	Voltage_pu	1		This is the gate in pu for Bus1.	20	0.0
33	0	0	Pload	11		This is the gate for Load11.	20	0.0
34	0	0	Pload	10		This is the gate for Load10.	20	0.0
35	0	0	Pload	9		This is the gate for Load9.	20	0.0
36	0	0	Pload	8		This is the gate for Load8.	20	0.0
37	0	0	Pload	7		This is the gate for Load7.	20	0.0
38	0	0	Pload	6		This is the gate for Load6.	20	0.0
39	0	0	Pload	5		This is the gate for Load5.	20	0.0
40	0	0	Pload	4		This is the gate for Load4.	20	0.0
41	0	0	Pload	3		This is the gate for Load3.	20	0.0
42	0	0	Pload	1		This is the gate for Load1.	20	0.0
43	0	0	Qload	11		This is the gate for Load11.	20	0.0
44	0	0	Qload	10		This is the gate for Load10.	20	0.0
45	0	0	Qload	9		This is the gate for Load9.	20	0.0
46	0	0	Qload	8		This is the gate for Load8.	20	0.0
47	0	0	Qload	7		This is the gate for Load7.	20	0.0
48	0	0	Qload	6		This is the gate for Load6.	20	0.0
49	0	0	Qload	5		This is the gate for Load5.	20	0.0
50	0	0	Qload	4		This is the gate for Load4.	20	0.0
51	0	0	Qload	3		This is the gate for Load3.	20	0.0
52	0	0	Qload	1		This is the gate for Load1.	20	0.0
53	0	0	Plosses	1		This is the gate for the Plosses.	20	0.0
54	0	0	Qlosses	1		This is the gate for the Qlosses.	20	0.0
55	0	0	P_inj	6		This is the gate for the injected power of the Generator on bus 6.	20	0.0
56	0	0	P_inj	11		This is the gate for the injected power of the Generator on bus 11.	20	0.0
57	0	0	Current_Load	11		This is the gate for the curent in Load11.	20	0.0
58	0	0	Current_Load	10		This is the gate for the curent in Load10.	20	0.0
59	0	0	Current_Load	9		This is the gate for the curent in Load9.	20	0.0
60	0	0	Current_Load	8		This is the gate for the curent in Load8.	20	0.0
61	0	0	Current_Load	7		This is the gate for the curent in Load7.	20	0.0
62	0	0	Current_Load	6		This is the gate for the curent in Load6.	20	0.0
63	0	0	Current_Load	5		This is the gate for the curent in Load5.	20	0.0
64	0	0	Current_Load	4		This is the gate for the curent in Load4.	20	0.0
65	0	0	Current_Load	3		This is the gate for the curent in Load3.	20	0.0
66	0	0	Current_Load	2		This is the gate for the curent in Load2.	20	0.0
67	0	0	Current_Load	1		This is the gate for the curent in Load1.	20	0.0

Στο σημείο αυτό πρέπει να αναφερθεί πως οι πύλες των τάσεων, ρευμάτων και ισχύων ανήκουν στην κατηγορία String και όχι Numeric διότι θα πρέπει σε κάθε περίπτωση να είναι σε μια προκαθορισμένη μορφή.

3.5.2 Code

Ο κώδικας ο οποίος αναπτύχθηκε για τις ανάγκες της εφαρμογής χωρίζεται σε τρία αρχεία με τα παρακάτω ονόματα: Animation, Loads, Main.

3.5.2.1 Loads

Το αρχείο αυτό καλείται με όρισμα μια δυαδική μεταβλητή διότι έχει δύο χρήσεις. Η πρώτη χρήση που ισχύει όταν η μεταβλητή είναι ψευδής αποσκοπεί στην επανεκκίνηση του συστήματος. Η επιλογή αυτή καλείται σε δύο περιπτώσεις, στην εκκίνηση της εφαρμογής για να διαγραφούν δεδομένα από προηγούμενες χρήσεις ή όταν ο χρήστης καλέσει την συνάρτηση reset() (έχει υλοποιηθεί στο αρχείο Main). Σε κάθε περίπτωση η επανεκκίνηση γίνεται με την αρχικοποίηση των μεταβλητών που χρησιμεύουν στην γραφική παράσταση όπως φαίνεται στον Κώδικα 1. Έπειτα ο κώδικας μπαίνει σε μια λούπα μέχρις ότου το μέγεθος του αρχείου Winlog_input.txt να γίνει μικρότερο από 10KB, το οποίο σημαίνει ότι η Python το έχει αδειάσει από πιθανά ανεπιθύμητα δεδομένα που μπορεί να περιείχε. Στην συνέχεια ο κώδικας προχωρά στην ανάγνωση του αρχείου και στην ανάθεση κενών χαρακτήρων στις πύλες, όπως φαίνεται στον Κώδικα 2.

```
SetNumGateValue("fmin",1,0.0);  
SetNumGateValue("fmin",2,0.0);  
SetNumGateValue("fmax",1,0.0);  
SetNumGateValue("fmax",2,0.0);  
SetNumGateValue("Pfeeder_min",1,0.0);  
SetNumGateValue("Pfeeder_max",1,0.0);
```

Κώδικας 2 - Αρχικοποίηση μεταβλητών της γραφικής παράστασης

Η δεύτερη χρήση ισχύει όταν η δυαδική μεταβλητή είναι αληθής με σκοπό την ανάκτηση των αποτελεσμάτων από την προσομοίωση και την ανάθεση τους σε πύλες. Αρχικά, όταν καλείται με αυτόν τον τρόπο η συνάρτηση ανοίγει το αρχείο Winlog_output.csv και δίνει την τιμή 1 στο κελί της στήλης digits που αντιστοιχεί στην επιλογή Calc_load. Έπειτα ο κώδικας μπαίνει σε μια λούπα μέχρι το μέγεθος του αρχείου Winlog_input.txt να γίνει μεγαλύτερο από 10KB, το οποίο σε αντίθεση με την προηγούμενη περίπτωση σημαίνει ότι η

Python έχει γεμίσει το αρχείο με τα δεδομένα τα οποία χρειάζονται. Τα δεδομένα αυτά αφορούν όλες τις String πύλες καθώς και την αριθμητική πύλη Pfeeder. Επιπλέον, όπως και πριν ο κώδικας προχωρά στην ανάγνωση του αρχείου και στην ανάθεση αυτή τη φορά τιμών σε κάθε μια από τις μεταβλητές. Ο λόγος για τον οποίο οι μεταβλητές πρέπει να έχουν προκαθορισμένη μορφή είναι επειδή η Winlog διαβάζει το αρχείο ανά byte, οπότε δεν είναι δυνατή η ανάγνωση του εάν δεν είναι γνωστό εκ των προτέρων το μέγεθος (σε χαρακτήρες) της κάθε μεταβλητής.

3.5.2.2 Animation

Στο αρχείο Animation, που περιγράφεται στον Κώδικα 3, έχουν οι υλοποιηθεί οι έξι συναρτήσεις που φαίνονται στον Κώδικα 2.

```
function void Calculate_Loads_animation()
#Macro
    TObjSetSize(46,115,115);
    Sleep(150);
    TObjSetSize(46,130,130);
end

function void Save_Data_animation()
#Macro
    TObjSetSize(90,115,115);
    Sleep(150);
    TObjSetSize(90,130,130);
end

function void OPF_UP_animation()
#Macro
    TObjSetSize(77,66.5,66.5);
    Sleep(100);
    TObjSetSize(77,75,75);
end

function void OPF_DOWN_animation()
#Macro
    TObjSetSize(78,66.5,66.5);
    Sleep(100);
    TObjSetSize(78,75,75);
end

function void reset_animation()
#Macro
    TObjSetSize(84,85,85);
    Sleep(100);
    TObjSetSize(84,100,100);
end

function void turn_off_animation()
#Macro
    TObjSetSize(85,87,87);
    Sleep(100);
    TObjSetSize(85,100,100);
end
```

Κώδικας 3 - Animation Code (Winlog)

Σκοπός των συναρτήσεων αυτών είναι να κάνουν τα κουμπιά που υπάρχουν στο γραφικό περιβάλλον να μοιάζουν περισσότερο με αληθινά δίνοντας στον χειριστή την αίσθηση του πατήματος. Αυτό γίνεται μειώνοντας στιγμιαία το μέγεθος της εικόνας και επαναφέροντας την μετά στην αρχική της κατάσταση. Η ετικέτα #Macro υπάρχει σε όλες τις συναρτήσεις ώστε να λειτουργούν μόνο όταν ο χειριστής επιλέξει να καλέσει την εκάστοτε συνάρτηση.

3.5.2.3 Main

Στο αρχείο αυτό ορίζονται δυο global μεταβλητές: η output_csv και η input_csv, οι οποίες αφορούν το μονοπάτι για τη θέση μνήμης στην οποία είναι αποθηκευμένα τα αρχεία Winlog_output.csv και Winlog_input.txt αντίστοιχα. Οι μεταβλητές είναι τύπου global string και όχι απλώς string προκειμένου να είναι αναγνώσιμες και από τα δύο αρχεία κώδικα που έχουν ήδη προαναφερθεί. Το υπόλοιπο αρχείο περιέχει οκτώ συναρτήσεις, που αναλύονται στους Κώδικες 4 έως 9. Η συνάρτηση init() έχει την ετικέτα #Startup γιατί καλείται κατά την εκκίνηση της εφαρμογής και ρόλος της είναι η αρχικοποίηση όλων των μεταβλητών. Όπως φαίνεται και στον Κώδικα 4 αρχικά αρχικοποιούνται οι δύο μεταβλητές Objective_OPF και OPF_select που αφορούν τις επιλογές του χειριστή και στη συνέχεια καλείται η συνάρτηση Calculate_Loads() με όρισμα False για να αρχικοποιήσει τις υπόλοιπες μεταβλητές με τον τρόπο που περιεγράφηκε στο 3.5.2.1.

```
function void init()  
#Startup  
  
SetDigGateValue("Objective_OPF",1,0);  
SetNumGateValue("OPF_select",1,1);  
  
Calculate_Loads(False);  
  
end
```

Κώδικας 4 - Συνάρτηση init (Winlog)

Η συνάρτηση Objective_OPF(), αυτή του Κώδικα 5, καλείται όποτε ο χειριστής επιθυμεί να αλλάξει την επιλογή του για τον στόχο της υλοποίησης. Οι δυο επιλογές είναι η ελαχιστοποίηση των απωλειών και η μεγιστοποίηση της εισαγόμενης ισχύος. Η επιλογή αυτή υλοποιείται γράφοντας «0» ή «1» στο κελί της στήλης digits του αρχείου Winlog_output.csv που αντιστοιχεί στην επιλογή objective_OPF. Το 1 αντιπροσωπεύει την επιλογή της ελαχιστοποίησης των απωλειών και το 0 της μεγιστοποίησης της εισαγόμενης ισχύος.

```

function void Objective_OPF()
#Macro
int fileHandle;

fileHandle=FileOpen(output_csv, "r+b");

if (GetDigGateValue("Objective_OPF",1)) then
    FileSeek(fileHandle, 33, 1);
    FileWrite(fileHandle, "0", 1);          //Maximization
    SetDigGateValue("Objective_OPF",1,0);
else
    FileSeek(fileHandle, 33, 1);
    FileWrite(fileHandle, "1", 1);          //Minimization
    SetDigGateValue("Objective_OPF",1,1);
end

FileClose(fileHandle);

end

```

Κώδικας 5 – Συνάρτηση Objective_OPF (Winlog)

Η συνάρτηση OPF_operation() στον Κώδικα 6 αντιπροσωπεύει την δεύτερη επιλογή του χειριστή και αφορά τον τρόπο της υλοποίησης. Οι τρεις επιλογές είναι η βέλτιστη επιλογή, η επιλογή με αναλογικά Phead και η επιλογή με ίσα Phead. Ομοίως, η επιλογή υλοποιείται γράφοντας «0», «1» ή «2» στο κελί της στήλης digits που αντιστοιχεί στο OPF_op για κάθε μια αντίστοιχα.

```

function void OPF_operation()
#Macro
int fileHandle;

fileHandle=FileOpen(output_csv, "r+b");

if (GetNumGateValue("OPF_select",1) == 1) then
    FileSeek(fileHandle, 43, 1);
    FileWrite(fileHandle, "0", 1);          //Standard operation
else
    if (GetNumGateValue("OPF_select",1) == 2) then
        FileSeek(fileHandle, 43, 1);
        FileWrite(fileHandle, "1", 1);    //Proportional Phead
    else
        FileSeek(fileHandle, 43, 1);
        FileWrite(fileHandle, "2", 1);    //Same Phead
    end
end

FileClose(fileHandle);

end

```

Κώδικας 6 - Συνάρτηση OPF_operation (Winlog)

Οι δύο συναρτήσεις του Κώδικα 7 διαχειρίζονται την προβολή της προηγούμενης επιλογής στο γραφικό περιβάλλον της εφαρμογής και κατά επέκταση και την επιλογή του χειριστή. Ο τρόπος με τον οποίο προβάλλονται οι επιλογές θα περιγραφεί αναλυτικότερα στο επόμενο υποκεφάλαιο, αλλά πρακτικά η λειτουργία είναι αυτή ενός μενού. Όταν ο χειριστής πατάει το κουμπί «up» τότε η πύλη OPF_select αυξάνεται κατά ένα και προβάλλεται η επόμενη επιλογή. Όταν ο χειριστής πατήσει το κουμπί «down» τότε η πύλη μειώνεται κατά ένα και προβάλλεται η προηγούμενη επιλογή. Το μενού δεν είναι κυκλικό, το οποίο σημαίνει πως αν επιλεγεί η προηγούμενη επιλογή από την πρώτη ή η επόμενη από την τρίτη δεν πραγματοποιείται κάποια αλλαγή.

```
function void OPF_select_up()  
#Macro  
  
    if (GetNumGateValue("OPF_select",1) < 3) then  
        SetNumGateValue("OPF_select",1,GetNumGateValue("OPF_select",1) + 1);  
    end  
  
    OPF_operation();  
end  
  
function void OPF_select_down()  
#Macro  
  
    if (GetNumGateValue("OPF_select",1) > 1) then  
        SetNumGateValue("OPF_select",1,GetNumGateValue("OPF_select",1) - 1);  
    end  
  
    OPF_operation();  
end
```

Κώδικας 7 - Συναρτήσεις OPF_select (Winlog)

Στον Κώδικα 8 η συνάρτηση reset() ενεργοποιείται όταν ο χειριστής πατήσει το αντίστοιχο κουμπί και έχει ως σκοπό την επαναφορά όλων των μεταβλητών του περιβάλλοντος στην αρχική τους κατάσταση. Αρχικά, όπως και σε όλες τις άλλες επιλογές του χειριστή η συνάρτηση όταν καλείτε ανοίγει το αρχείο Winlog_output.csv και γράφει «1» στο κελί της στήλης digits που αντιστοιχεί στην επιλογή reset. Στην συνέχεια, παρόμοια με την συνάρτηση init() καλεί την συνάρτηση Calculate_Loads() με όρισμα False ορίζοντας την τιμή των μεταβλητών σε κενούς χαρακτήρες.

```

function void reset()
#Macro
int fileHandle;

    fileHandle=FileOpen(output_csv, "r+b");

    FileSeek(fileHandle, 66, 1);
    FileWrite(fileHandle, "1", 1);

    FileClose(fileHandle);

    Calculate_Loads(False);

end

```

Κώδικας 8 - Συνάρτηση reset (Winlog)

Η συνάρτηση turn_off(), που περιγράφεται στον Κώδικα 9, καλείται όταν ο χειριστής πατήσει το κουμπί «turn_off». Παρόμοια με άλλες επιλογές η συνάρτηση ανοίγει το αρχείο Winlog_output.csv και γράφει «1» στο κελί της στήλης digits που αντιστοιχεί στην επιλογή turn_off. Έπειτα καλεί την συνάρτηση CloseSession() της Winlog, χωρίς ορίσματα, η οποία τερματίζει την λειτουργία της εφαρμογής.

```

function void turn_off()
#Macro
int fileHandle;

    fileHandle=FileOpen(output_csv, "r+b");

    FileSeek(fileHandle, 78, 1);
    FileWrite(fileHandle, "1", 1);

    FileClose(fileHandle);

    CloseSession();

end

```

Κώδικας 9 - Συνάρτηση turn_off (Winlog)

Η συνάρτηση save_data(), Κώδικας 10, καλείται όταν ο χειριστής πατήσει το κουμπί «Save_data» και το μόνο που κάνει είναι να γράφει «1» στο κελί της στήλης digits που αντιστοιχεί στην επιλογή save_data στο αρχείο Winlog_output.csv.

```

function void save_data()
#Macro
int fileHandle;

fileHandle=FileOpen(output_csv, "r+b");

FileSeek(fileHandle, 91, 1);
FileWrite(fileHandle, "1", 1);

FileClose(fileHandle);

end

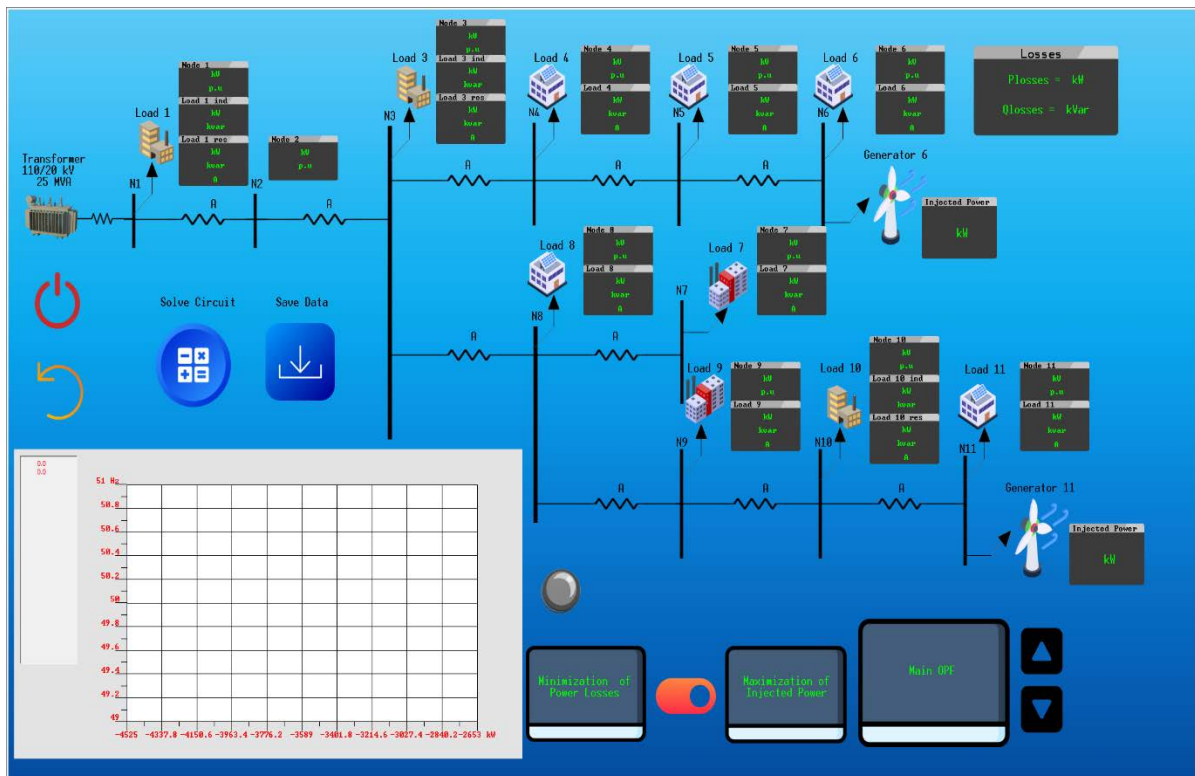
```

Κώδικας 10 - Συνάρτηση save_data (Winlog)

3.5.3 Templates

Τα Templates αποτελούν το κομμάτι του λειτουργικού συστήματος της Winlog στο οποίο σχεδιάζεται το γραφικό περιβάλλον της εφαρμογής. Το περιβάλλον για την παρούσα εργασία είναι αυτό της Εικόνας 4, όπου φαίνεται το κύκλωμα και όλες οι επιλογές που περιεγράφηκαν παραπάνω. Κάθε φορτίο του κυκλώματος συμβολίζεται με ένα κτήριο, ανάλογα με το αν είναι βιομηχανικό, οικιακό ή και τα δύο, και οι μονάδες ΑΠΕ με το σύμβολο της ανεμογεννήτριας. Δίπλα σε κάθε φορτίο προβάλλονται τα παρακάτω στοιχεία: ρεύμα σε Aber (A), ενεργός και άεργος ισχύς σε kilo Watt (kW) και kilo Volt Aber Reactive (kvar) αντίστοιχα. Επίσης, προβάλλονται οι τάσεις σε kilo volt (kV) και per unit (p.u) για τον εκάστοτε ζυγό. Δίπλα σε κάθε ανεμογεννήτρια προβάλλεται η ενεργός ισχύς που προσφέρεται στο σύστημα. Οι συνολικές απώλειες του συστήματος προβάλλονται στο παράθυρο με τίτλο Losses στο πάνω δεξί μέρος του παραθύρου. Τέλος, τα ρεύματα των γραμμών προβάλλονται πάνω από την κάθε γραμμή. Στο κάτω αριστερά μέρος του παραθύρου πραγματοποιείται η γραφική παράσταση της καμπύλης στατισμού στο σημείο διασύνδεσης με την επιλογή της παρακολούθησης της τιμής κάθε σημείου με τον κέρσορα του ποντικιού. Δεξιά της γραφικής υπάρχει ένα LED το οποίο ανάβει όποτε πραγματοποιείται κάποια αλλαγή στις τιμές του συστήματος. Τα κουμπιά με τα οποία μπορεί να αλληλεπιδράσει ο χειριστής είναι: το «turn_off» που είναι το κόκκινο κουμπί κάτω από το σύμβολο του μετασχηματιστή, το «reset» κουμπί που βρίσκεται κάτω από το «turn_off, το «Save Data» που είναι αυτό που αποθηκεύει τα δεδομένα σε csv αρχείο, το «Solve Circuit» το οποίο λύνει το σύστημα με τις επιλεγμένες παραμέτρους, ο πορτοκαλί διακόπτης με τον οποίο επιλέγεται η συνάρτηση στόχου, δηλαδή η ελαχιστοποίηση των απωλειών ή η

μεγιστοποίηση της εισαγόμενης ισχύος και τα «up» και «down» κουμπιά που μεταβάλουν την πύλη «OPF_select» προβάλλοντας παράλληλα την αντίστοιχη επιλογή στο πλαίσιο αριστερά από τα κουμπιά.



Εικόνα 4 - Γραφικό περιβάλλον Winlog

3.6 Λειτουργία της Python

Η ανάγκη για την χρήση της Python για την μεταφορά δεδομένων είναι εμφανής από την πλευρά και των δύο εφαρμογών. Από την μία το MATLAB χρειάζεται τις επιλογές του χειριστή για να λειτουργήσει και για τον λόγο αυτό, όπως και το OpenDSS, καλείται μέσω ενός API και όχι από το αντίστοιχο περιβάλλον. Από την άλλη όπως αναφέρθηκε παραπάνω η Winlog διαβάζει τα δεδομένα ανά byte, οπότε είναι απολύτως απαραίτητο τα δεδομένα να αποθηκεύονται με προκαθορισμένη μορφή. Ο κώδικας στην Python είναι δομημένος στις παρακάτω πέντε συναρτήσεις: `csv_init()`, `rhead_calc()`, `objective_orpf()`, `loads_export()` και `save_data()` και σε μια `while(True)` λούπα που αποτελεί τον έλεγχο του συστήματος. Προτού αναλυθεί η λειτουργία του κώδικα πρέπει να αναφερθεί πως είναι απολύτως απαραίτητες οι εξής πέντε βιβλιοθήκες:

1. `matlab.engine`: Για το API του MATLAB
2. `pandas`: Για την διαχείριση των data frames
3. `os`: Για την διαχείριση των μονοπατιών των θέσεων μνήμης
4. `datetime`: Για την ανάκτηση της ημέρας και της ώρας
5. `shutil`: Για την αντιγραφή αρχείου

Επιπλέον, πρέπει να οριστούν οι μεταβλητές που αφορούν τις θέσεις που βρίσκονται αποθηκευμένα τα αρχεία, όπως φαίνεται στην Εικόνα 5.

```
working_path = "C:/Users/harry/Desktop/cigre_MV"

Winlog_output_csv = os.path.join(working_path, "Winlog_output.csv")
Winlog_input_csv = os.path.join(working_path, "Winlog_input.txt")
phead_matlab_script = os.path.join(working_path, "Phead_calc.m")
objectiveOPF_matlab_script = os.path.join(working_path, "Objective_OPF.m")
```

Εικόνα 5 - Python paths

Η μεταβλητή `working_path` αφορά το φάκελο στον οποίο βρίσκονται όλα τα αρχεία και είναι διαφορετική ανάλογα με τη συσκευή. Οι υπόλοιπες μεταβλητές αλλάζουν κατάλληλα παίρνοντας ως όρισμα την `working_path` μεταβλητή ώστε να καταλήγουν στο αντίστοιχο αρχείο.

Η συνάρτηση `csv_init()` είναι αυτή που περιγράφεται παρακάτω στον Κώδικα 11 και δημιουργεί τα αρχεία `Winlog_output.csv` και `Winlog_input.txt` αρχικοποιημένα. Όταν καλείται δημιουργεί δυο λίστες. Την `operations`, η οποία περιέχει όλες τις κατηγορίες επιλογών του χειριστή και την `digits` η οποία συμβολίζει τις τιμές που μπορούν να πάρουν οι παραπάνω επιλογές, αρχικοποιημένες σε «0». Για τον λόγο αυτό, οι δύο λίστες έχουν το ίδιο μήκος. Στην συνέχεια οι λίστες μετατρέπονται σε `DataFrames` με την βοήθεια της βιβλιοθήκης `pandas` και συμπύσσονται σε ένα κοινό `DataFrame`, το οποίο μετατρέπεται τελικά στο αρχείο «`Winlog_output.csv`». Επιπλέον δημιουργείται ένα κενό `DataFrame` το οποίο μετατρέπεται στο αρχείο «`Winlog_input.txt`». Τέλος, τυπώνεται το μήνυμα: «`Winlog_output and Winlog_input have been created successfully!`» ώστε να γνωρίζει (εάν χρειαστεί) πως αυτή η διεργασία ολοκληρώθηκε επιτυχώς.

```

def csv_init(): 3 usages
    # Lists with all the operations for the Winlog_output file.
    operations = ["objective_OPF", "OPF_op", "Calc_loads", "Reset", "Turn_off", "Save_Data"]
    digits = [0, 0, 0, 0, 0, 0]

    # Convert the lists to DataFrames.
    df_operations = pd.DataFrame({"operations": operations})
    df_digits = pd.DataFrame({"digits": digits})

    # Concat the two separate DataFrames to one.
    df = pd.concat([df_operations, df_digits], axis=1)

    # Create one empty DataFrame.
    empty_df = pd.DataFrame()

    # Export the DataFrames to csv files.
    df.to_csv(Winlog_output_csv, index=False)
    empty_df.to_csv(Winlog_input_csv)

    print("Winlog_output and Winlog_input have been created successfully!")

```

Κώδικας 11 - Συνάρτηση csv_init (Python)

Η συνάρτηση phead_calc() στον Κώδικα 12 δημιουργεί το αρχείο «Phead_calc.m» που χρησιμοποιεί το MATLAB. Όταν καλείται η συνάρτηση διαβάζει τα περιεχόμενα του αρχείου Winlog_output.csv και το αποθηκεύει σε ένα DataFrame με όνομα «df». Αυτό που χρειάζεται πρακτικά από το DataFrame είναι η μεταβλητή «df.digits[1]» η οποία περιέχει την πληροφορία για την είσοδο του χειριστή στην επιλογή OPF_op. Στη συνέχεια καλεί το API του MATLAB ώστε να τρέξει ένα από τα OPF αρχεία (Main_OPF, Main_OPF_same_Pheads, Main_OPF_proportional_Pheads) ανάλογα με την επιλογή του χειριστή. Από την κλήση αυτή ανακτώνται τα όρια του Phead και αποθηκεύονται σε μια string μεταβλητή όπως φαίνεται παρακάτω. Το περιεχόμενο της μεταβλητής αυτής γίνεται το περιεχόμενο του αρχείου «Phead_calc.m» και τυπώνεται το μήνυμα «The Phead_calc.m has changed successfully!».

```

def phead_calc(): 1 usage
# Read the Winlog_output.csv file to know which mode will be used.
df = pd.read_csv(Winlog_output_csv)
# Change the Phead depending on the mode chosen.
if df.digits[1] == 0:
    # Start MATLAB engine.
    eng = matlab.engine.start_matlab()
    # Go to the path that the file is saved at.
    eng.addpath(working_path)
    # Run the specified file.
    eng.Main_OPF(nargout=0)
    # Retrieve the Pfeeder from MATLAB.
    Pfeeder = f"{eng.eval('Pfeeder') / 1000:0=+10.4f}"
    # Get the Phead boundaries.
    Phead_min = eng.eval('[x(37)]')
    Phead_max = eng.eval('[x(36)]')
    # Write the script for the Phead_calc.m file.
    script_content = f"""
function Phead = Phead_calc()
    Phead=[{Phead_min} {Phead_max}];
end
"""
    # Close MATLAB engine.
    eng.quit()
elif df.digits[1] == 2:
    # Start MATLAB engine.
    eng = matlab.engine.start_matlab()
    # Go to the path that the file is saved at.
    eng.addpath(working_path)
    # Run the specified file.
    eng.Main_OPF_same_Pheads(nargout=0)
    # Retrieve the Pfeeder from MATLAB.
    Pfeeder = f"{eng.eval('Pfeeder') / 1000:0=+10.4f}"
    # Get the Phead boundaries.
    Phead_min = eng.eval('[x(37)]')
    Phead_max = eng.eval('[x(36)]')
    # Write the script for the Phead_calc.m file.
    script_content = f"""
function Phead = Phead_calc()
    Phead=[{Phead_min} {Phead_max}];
end
"""
    # Close MATLAB engine.
    eng.quit()
elif df.digits[1] == 1:
    # Start MATLAB engine.
    eng = matlab.engine.start_matlab()
    # Go to the path that the file is saved at.
    eng.addpath(working_path)
    # Run the specified file.
    eng.Main_OPF_proportional_Pheads(nargout=0)
    # Retrieve the Pfeeder from MATLAB.
    Pfeeder = f"{eng.eval('Pfeeder') / 1000:0=+10.4f}"
    # Get the Phead boundaries.
    Phead_min = eng.eval('[x(37)]')
    Phead_max = eng.eval('[x(36)]')
    # Write the script for the Phead_calc.m file.
    script_content = f"""
function Phead = Phead_calc()
    Phead=[{Phead_min} {Phead_max}];
end
"""
    # Close MATLAB engine.
    eng.quit()
else:
    # Write the default case script for the Phead_calc.m file.
    script_content = """
function Phead = Phead_calc()
    Phead=[0.8 0.00001]; % default
end
"""
    Pfeeder = -3068027.93737460
# Open the Phead_calc.m file to write.
with open(phead_matlab_script, 'w') as file:
    # Apply the changes to the Phead_calc.m file.
    file.write(script_content)

print("The Phead_calc.m has changed successfully!")
return Pfeeder

```

Κώδικας 12 - Συνάρτηση phead_calc (Python)

Η συνάρτηση `objective_opf()`, Κώδικας 13, παρόμοια με την συνάρτηση `rhead_calc()` προετοιμάζει ένα αρχείο MATLAB (`Objective_OPF.m`). Αντίστοιχα με πριν αρχικά η συνάρτηση διαβάζει το `Winlog_output.csv` αρχείο σε ένα `DataFrame` και διαβάζει την επιλογή `objective_OPF` από την μεταβλητή «`df.digits[0]`». Ανάλογα με την επιλογή του χειριστή αλλάζει το περιεχόμενο του αρχείου «`Objective_OPF.m`» με την βοήθεια μιας string μεταβλητής, όπως φαίνεται παρακάτω, και τυπώνεται το μήνυμα «`The Objective_OPF.m has changed successfully!`».

```
def objective_opf(): 1usage
    # Read the Winlog_output.csv file to know which mode will be used.
    df = pd.read_csv(Winlog_output_csv)
    # Change the Objective_OPF.m depending on the mode.
    if df.digits[0] == 0:
        script_content = """
        function f = Objective_OPF( x )

        % Maximization of injected power
        f=-sum(x(34:35));

        % Minimization of power losses
        %f=sum(x(12:22));

        end
        """
    elif df.digits[0] == 1:
        script_content = """
        function f = Objective_OPF( x )

        % Minimization of power losses
        f=sum(x(12:22));

        % Maximization of injected power
        %f=-sum(x(34:35));

        end
        """
    # Open the Objective_OPF file to write
    with open(objectiveOPF_matlab_script, "w") as file:
        # Apply the changes to the Objective_OPF file
        file.write(script_content)
    print("The Objective_OPF.m has changed successfully!")
```

Κώδικας 13 - Συνάρτηση `objective_opf` (Python)

Η συνάρτηση `loads_export()` είναι υπεύθυνη για την ανάκτηση όλων των δεδομένων από το MATLAB. Όπως φαίνεται στον Κώδικα 14 αυτό γίνεται εκτελώντας το αρχείο

Desing_Power_at_PCC ώστε να αποθηκευτούν τα παρακάτω δεδομένα: Pfeeder, Voltage_ru, Voltage, Pload, Qload, Current_Line, Current_Load, PGen, Plosses και Qlosses. Όσα από αυτά είναι πίνακες μεταβλητών μετατρέπονται σε λίστες, προκαθορίζοντας ταυτόχρονα την μορφή της κάθε μεταβλητής, και στη συνέχεια μετατρέπονται όλα σε DataFrames όπως φαίνεται παρακάτω. Τέλος, συμπύσσονται όλα τα επιμέρους DataFrames σε ένα τελικό DataFrame το οποίο αποθηκεύεται στο αρχείο «Winlog_input.txt» και τυπώνεται το μήνυμα «Loads and Currents have been exported successfully!».

```

def loads_export(Pfeeder): 1 usage
    # Start MATLAB engine.
    eng = matlab.engine.start_matlab()

    # Go to the path that the file is saved at.
    eng.addpath(working_path)

    # Run the specified file.
    eng.Design_Power_at_PCC(nargout=0)

    # Retrieve the Voltages, Currents, Ploads, Qloads, Losses from MATLAB.
    Voltage_pu = eng.eval('Voltage_pu')
    Voltage = eng.eval('Voltage.VAN_mag')
    Pload = eng.eval('Load(:,5);')
    Qload = eng.eval('Load(:,6);')
    Current_Line = eng.eval('Currents.pha_mag')
    Current_Load = eng.eval('cur')
    PGen = eng.eval('Gen(:,5);')
    Plosses = f"{eng.eval('losses(:,1);') / 1000:0=8.4f}"
    Qlosses = f"{eng.eval('losses(:,2);') / 1000:0=8.4f}"

    # Close MATLAB engine.
    eng.quit()

    # Convert MATLAB arrays to Python lists.
    Voltage_pu_py = [f"{x:5.4f}".format(Voltage_pu[i][0]) for i in range(len(Voltage_pu))]
    Voltage_py = [f"{x / 1000:7.4f}" for x in Voltage[0]] # Voltages/1000 kW
    Pload_py = [f"{x:8.4f}" for x in list(Pload)]
    Qload_py = [f"{x:8.4f}" for x in list(Qload)]
    Current_Line_py = [f"{x:8.4f}" for x in list(Current_Line[0])]
    Current_Load_py = [f"{x:8.4f}" for x in list(Current_Load[0])]
    PGen_py = [f"{x:9.4f}" for x in list(PGen)]

    # Convert the lists to Dataframes.
    df_Voltage_pu = pd.DataFrame({"Voltage_pu": Voltage_pu_py})
    df_Voltages = pd.DataFrame({"Voltages": Voltage_py})
    df_Pfeeder = pd.DataFrame({"Pfeeder": [Pfeeder]})
    df_Pload = pd.DataFrame({"Pload": Pload_py})
    df_Qload = pd.DataFrame({"Qload": Qload_py})
    df_Plosses = pd.DataFrame({"Plosses": [Plosses]})
    df_Qlosses = pd.DataFrame({"Qlosses": [Qlosses]})
    df_Currents_Line = pd.DataFrame({"Currents_Line": Current_Line_py})
    df_Currents_Load = pd.DataFrame({"Currents_Load": Current_Load_py})
    df_Pinj = pd.DataFrame({"P_inj": PGen_py})

    # Merge the Dataframes to one complete Dataframe.
    df_out = pd.concat([df_Pfeeder, df_Plosses, df_Qlosses, df_Pinj, df_Voltages,
                        df_Voltage_pu, df_Currents_Load, df_Currents_Line, df_Pload, df_Qload], axis=1)

    # Export the final Dataframe to Winlog_input.txt
    df_out.to_csv(Winlog_input_csv, index=False)

    print("Loads and Currents have been exported successfully!")

```

Κώδικας 14 - Συμπύληση loads_export (Python)

Η συνάρτηση `save_data()` χρησιμοποιείται για να αποθηκεύσει τα δεδομένα μίας προσομοίωσης σε ένα CSV αρχείο. Αυτό γίνεται δημιουργώντας ένα αντίγραφο του αρχείου «Winlog_input.txt» με κατάληξη `.csv` στον φάκελο «Saved_Data» που βρίσκεται στην ίδια τοποθεσία με τα υπόλοιπα αρχεία. Το όνομα του νέου αρχείου είναι η ημερομηνία και ώρα κατά την οποία αποθηκεύτηκε ώστε να γίνεται εύκολα διαχωρισμός όλων των δεδομένων και έπειτα τυπώνεται το μήνυμα «Data Saved Successfully!». Η υλοποίηση της συνάρτησης `save_data()` παρουσιάζεται παρακάτω στον Κώδικα 15.

```
def save_data(): 1 usage
    # Create a string with the current date and time.
    date_str = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")

    # Create a string with the destination path to save the file.
    destination = os.path.join(working_path, f"Saved_Data/{date_str}.csv")

    # Save a copy of Winlog_input.txt to the destination file.
    shutil.copyfile(Winlog_input_csv, destination)

    print("Data Saved Successfully!")
```

Κώδικας 15 - Συνάρτηση `save_data` (Python)

Στον Κώδικα 16 παρακάτω φαίνεται η ατέρμονη λούπα που ελέγχει διαρκώς εάν ο χειριστής έχει κάνει κάποια αλλαγή στις επιλογές του. Όταν συμβαίνει αυτό τελούνται οι απαραίτητες ενέργειες. Αυτό σημαίνει πως καλούνται οι εκάστοτε συναρτήσεις που περιεγράφηκαν παραπάνω, όταν αυτό είναι απαραίτητο, και η επιλογή αρχικοποιείται σε «0».

```

#Initialize the files.
csv_init()

while True:

    # Read the Winlog_output.csv file to know if any option has changed.
    df = pd.read_csv(Winlog_output_csv)

    # Calculate Loads Option
    if df.digits[2] == 1:
        objective_opf()
        Pfeeder = phead_calc()
        csv_init()
        loads_export(Pfeeder)
        # Return to the original Value.
        df.loc[2, "digits"] = 0
        # Save the change.
        df.to_csv(Winlog_output_csv,index=False)
    elif df.digits[3] == 1:
        csv_init()
        # Return to the original Value.
        df.loc[3, "digits"] = 0
        # Save the change.
        df.to_csv(Winlog_output_csv,index=False)
    elif df.digits[4] == 1:
        # Return to the original Value.
        df.loc[4, "digits"] = 0
        # Save the change.
        df.to_csv(Winlog_output_csv,index=False)
        # End the run.
        break
    elif df.digits[5] == 1:
        save_data()
        # Return to the original Value.
        df.loc[5, "digits"] = 0
        # Save the change.
        df.to_csv(Winlog_output_csv,index=False)

```

Κώδικας 16 - Αυτόματος Έλεγχος (Python)

Κεφάλαιο 4 Εγχειρίδιο Χρήσης Της Εφαρμογής

4.1 Εγκατάσταση Λογισμικών

Για την λειτουργία της εφαρμογής είναι απαραίτητη η εγκατάσταση όλων των λογισμικών που αναφέρθηκαν στα προηγούμενα κεφάλαια (Winlog Evo 4.0, Python, MATLAB, OpenDSS). Στο κεφάλαιο αυτό θα παρατεθούν οδηγίες για την εγκατάσταση των παραπάνω και την ρύθμιση τους όπως αυτή είναι απαραίτητη. Για την αναπαραγωγή της εφαρμογής σε μια νέα συσκευή είναι απαραίτητο το λειτουργικό σύστημα της να είναι Windows, να έχει διαθέσιμο αποθηκευτικό χώρο για την εγκατάσταση των εφαρμογών και να έχει πρόσβαση στο διαδίκτυο για την λήψη των απαραίτητων αρχείων. Τέλος, θεωρείται δεδομένο πως υπάρχει πρόσβαση στα αρχεία που αναπτύχθηκαν για το κάθε λογισμικό ώστε να λειτουργεί η εφαρμογή.

4.1.1 Εγκατάσταση του OpenDSS

Για την εγκατάσταση του λογισμικού OpenDSS ο χειριστής θα πρέπει να μεταβεί στον παρακάτω σύνδεσμο: <https://sourceforge.net/projects/electricdss/> και να κάνει λήψη του αρχείου «OpenDSSInstaller_1020_1.exe» ακολουθώντας τις οδηγίες της σελίδας. Το όνομα του αρχείου μπορεί να διαφέρει ανάλογα με την τρέχουσα έκδοση. Αφού γίνει η λήψη του αρχείου ο χειριστής θα πρέπει να το ανοίξει και να ακολουθήσει τα βήματα της εγκατάστασης που θα εμφανιστούν στην οθόνη. Η αλληλεπίδραση του χειριστή με το γραφικό περιβάλλον του OpenDSS δεν είναι απαραίτητη και για τον λόγο αυτό δεν χρειάζεται να γίνει οποιαδήποτε αλλαγή.

4.1.2 Εγκατάσταση του MATLAB

Η εγκατάσταση του λογισμικού MATLAB γίνεται από τον ακόλουθο σύνδεσμο: <https://www.mathworks.com/help/install/ug/install-products-with-internet-connection.html> ακολουθώντας τις οδηγίες που αναγράφονται. Αφού γίνει η λήψη του αρχείου (π.Χ. matlab_R2024b_Windows.exe) ο χειριστής θα πρέπει να το ανοίξει και να ακολουθήσει τις οδηγίες εγκατάστασης παρόμοια με την περίπτωση του OpenDSS, με την μόνη διαφορά να είναι πως θα πρέπει να συνδεθεί με τον λογαριασμό του ώστε να λειτουργεί η εγκατάσταση. Μετά την εγκατάσταση η αλληλεπίδραση του χειριστή δεν είναι

απαραίτητη ούτε με το περιβάλλον του MATLAB καθώς οι λειτουργίες του τελούνται μέσω της Python.

4.1.3 Εγκατάσταση της Winlog Evo 4.0

Η λήψη του εκτελέσιμου αρχείου για την εγκατάσταση της Winlog γίνεται από τον παρακάτω σύνδεσμο: https://www.sielcosistemi.com/en/download/public/winlog_evo.html. Ο χειριστής θα πρέπει πρώτα να συμπληρώσει τα προσωπικά του στοιχεία (Όνομα, Επώνυμο, Εταιρία/Σχολή, Χώρα, Διεύθυνση Ηλεκτρονικού Ταχυδρομείου) και στην συνέχεια να πραγματοποιήσει την λήψη του αρχείου με όνομα «WinlogEvo_Setup.exe». Η εγκατάσταση του είναι σαφής και υλοποιείται ακολουθώντας τα βήματα μετά το άνοιγμα του αρχείου.

4.1.4 Εγκατάσταση της Python

Η εγκατάσταση ενός περιβάλλοντος για την εκτέλεση της Python έχει περισσότερους βαθμούς ελευθερίας καθώς ο κάθε χειριστής μπορεί να επιλέξει περιβάλλον βάσει προτίμησης του. Στην παρούσα εργασία χρησιμοποιήθηκε το περιβάλλον PyCharm του οποίου η λήψη μπορεί να γίνει από τον παρακάτω σύνδεσμο: <https://www.jetbrains.com/pycharm/download/?section=windows>. Μετά την λήψη του αρχείου (π.χ. pycharm-2025.1.1.1.exe) ο χειριστής θα πρέπει να ανοίξει το αρχείο και να ακολουθήσει τα βήματα της εγκατάστασης όπως και με τα προηγούμενα λογισμικά. Με το περιβάλλον της Python απαιτείται ελάχιστη αλληλεπίδραση από τον χειριστή, όμως για να λειτουργήσει θα πρέπει να εγκατασταθούν οι βιβλιοθήκες που αναφέρθηκαν στο Κεφάλαιο 3.6.

4.2 Εγκατάσταση βιβλιοθηκών της Python

Υπάρχουν πολλοί τρόποι για να εγκατασταθούν οι βιβλιοθήκες που χρειάζονται για την υλοποίηση της εφαρμογής, οπότε ο κάθε χειριστής μπορεί να επιλέξει αυτή που ο ίδιος θεωρεί ότι είναι η καλύτερη. Η πιο εύκολη επιλογή είναι, πιθανόν, η εγκατάσταση μέσω του ενσωματωμένου τερματικού που προσφέρει το περιβάλλον της PyCharm, εκτελώντας της εκάστοτε εντολή. Στην παρούσα υλοποίηση, παρά όλα αυτά επιλέχθηκε μια πιο ασφαλής και αξιόπιστη επιλογή κάνοντας χρήση ενός βοηθητικού λογισμικού, του Anaconda [14]. Το λογισμικό αυτό χρησιμοποιείται για τη δημιουργία και διαχείριση ενός περιβάλλοντος το

οποίο θα περιλαμβάνει όλες τις απαραίτητες βιβλιοθήκες για να λειτουργήσει ο κώδικας. Η λήψη του λογισμικού γίνεται από τον ακόλουθο σύνδεσμο: <https://www.anaconda.com/download>, παρέχοντας μια διεύθυνση ηλεκτρονικού ταχυδρομείου. Μετά την λήψη του αρχείου (π.Χ. Anaconda3-2024.10-1-Windows-x86_64.exe) η εγκατάσταση είναι ίδια με την εγκατάσταση όλων των λογισμικών που προαναφέρθηκαν. Μετά την εγκατάσταση υπάρχουν δύο επιλογές για την αλληλεπίδραση με την εφαρμογή. Η πρώτη είναι μέσω του Anaconda Prompt (Εικόνα 6) το οποίο εγκαθίσταται κατά την εγκατάσταση της εφαρμογής. Η δεύτερη επιλογή είναι μέσω του Anaconda Navigator (Εικόνα 7) που είναι η εγκατεστημένη εφαρμογή και προσφέρει ένα γραφικό περιβάλλον για να γίνουν όλες οι διεργασίες.



Εικόνα 6 - Anaconda Prompt



Εικόνα 7 - Anaconda Navigator

Σε αυτή την εργασία επιλέχθηκε η υλοποίηση και περιγραφή μέσω του Anaconda Prompt διότι δεν εξαρτάται από πιθανές αλλαγές του γραφικού περιβάλλοντος με την πάροδο των εκδόσεων. Ανοίγοντας αρχικά το Anaconda Prompt θα είναι ορατή μια γραμμή παρόμοια με αυτήν: (base) C:\Users\User>. Το «(base)» συμβολίζει το περιβάλλον στο οποίο θα εκτελεστούν οι ενέργειες και το υπόλοιπο της γραμμής την διεύθυνση μνήμης. Το «base» αντιπροσωπεύει την βάση όλων των περιβαλλόντων που θα δημιουργηθούν, το οποίο σημαίνει πως οποιαδήποτε αλλαγή στην βάση επηρεάζει όλα τα υπόλοιπα. Στο τερματικό θα πρέπει να εκτελεστούν με την σειρά που παρουσιάζονται οι ακόλουθες εντολές:

1. `conda create -n name python=x`

Αντικαθιστώντας όπου name το όνομα του περιβάλλοντος (my_environment) και όπου x την έκδοση της Python που θα χρησιμοποιηθεί (π.Χ. 3.12). Με την παραπάνω εντολή δημιουργείται ένα νέο περιβάλλον με όνομα name και προκαθορισμένο να λειτουργεί για Python interpreter με έκδοση x.

2. `conda activate name`

Με την εντολή «2» ενεργοποιείται το περιβάλλον που δημιουργήθηκε προηγουμένως. Αυτό σημαίνει πως όλες οι παρακάτω αλλαγές θα ισχύουν μόνο για αυτό το περιβάλλον. Επίσης πλέον στην παρένθεση στην αρχή της πρότασης θα πρέπει να αναγράφεται το όνομα που δόθηκε στο περιβάλλον αντί για «base».

3. `conda install -c conda-forge pandas`

Με την εντολή «3» θα εγκατασταθεί η βιβλιοθήκη με όνομα «pandas» στο περιβάλλον. Κατά την εγκατάσταση θα ζητηθεί από τον χρήστη να συμφωνήσει στην λήψη των απαραίτητων αρχείων για την βιβλιοθήκη. Αυτό γίνεται πατώντας τον χαρακτήρα «y» και Enter.

Η παραπάνω εντολή θα πρέπει να επαναληφθεί και για τις υπόλοιπες βιβλιοθήκες (matlab.engine, os, datetime, shutil) ολοκληρώνοντας έτσι την διαδικασία προετοιμασίας του περιβάλλοντος και κατά επέκταση της εγκατάστασης των βιβλιοθηκών.

4.3 Προετοιμασία των Λογισμικών και Κώδικα

Όπως αναφέρθηκε και παραπάνω για την λειτουργία της εφαρμογής θα πρέπει να έχουν υλοποιηθεί κάποια αρχεία για όλες τις εφαρμογές τα οποία θα θεωρηθεί ότι βρίσκονται όλα σε ένα κοινό φάκελο, στην ενδεικτική διεύθυνση: C:\Users\User\Functions.

Αρχικά μέσα στον φάκελο που βρίσκονται τα αρχεία θα πρέπει να δημιουργηθεί ένας νέος φάκελος με όνομα «Saved_Data».

Στο περιβάλλον της PyCharm θα πρέπει να δημιουργηθεί ένα νέο Project στο οποίο θα αντιγραφούν οι κώδικες που έχουν υλοποιηθεί. Ανοίγοντας το περιβάλλον της PyCharm ο χειριστής θα πρέπει να κάνει ένα νέο Project δίνοντας του το όνομα της επιλογής του και την διεύθυνση στην οποία επιθυμεί να αποθηκευτεί. Οι υπόλοιπες επιλογές που υπάρχουν διαθέσιμες όμως θα πρέπει να συμπληρωθούν αυστηρά ως εξής:

- Interpreter type: Custom environment
- Environment: Select existing
- Type: Conda
- Path to conda: C:\Users\User\Anaconda3\condabin\conda.bat

Η παραπάνω επιλογή αποτελεί παράδειγμα και θα διαφέρει ανάλογα με την συσκευή. Επίσης θα συμπληρώνεται αυτόματα από την PyCharm.

- Environment: Το όνομα του περιβάλλοντος που δημιουργήθηκε από το Anaconda.

Αφού δημιουργηθεί το Project θα πρέπει να προστεθεί το .py αρχείο που περιέχει της υλοποίηση της εφαρμογής στην Python στον αντίστοιχο φάκελο του Project. Ο φάκελος αυτός βρίσκεται σε μία αντίστοιχη τοποθεσία με την εξής: «C:\Users\User\PycharmProjects\ProjectName». Αφού αντιγραφεί το αρχείο σε αυτόν τον φάκελο ο χειριστής θα πρέπει να ανοίξει το .py αρχείο και να αλλάξει την μεταβλητή «working_path» που δηλώνεται στην γραμμή οκτώ του κώδικα ώστε να έχει την διεύθυνση για τον φάκελο στον οποίο βρίσκονται όλα τα αρχεία που χρησιμοποιούνται από την εφαρμογή (π.Χ. C:\Users\User\Functions).

Αντίστοιχες αλλαγές θα πρέπει να γίνουν επίσης στα MATLAB αρχεία. Συγκεκριμένα από τα MATLAB θα πρέπει να αλλάξουν αυτά που καλούν το API του OpenDSS ώστε να δείχνουν στην διεύθυνση που βρίσκεται το .dss αρχείο (π.Χ. C:\Users\User\Functions\opendss_file.dss). Το ένα είναι το «data_write.m» που δημιουργεί το .dss αρχείο και το άλλο είναι το «Design_Power_at_PCC.m» που λύνει το κύκλωμα.

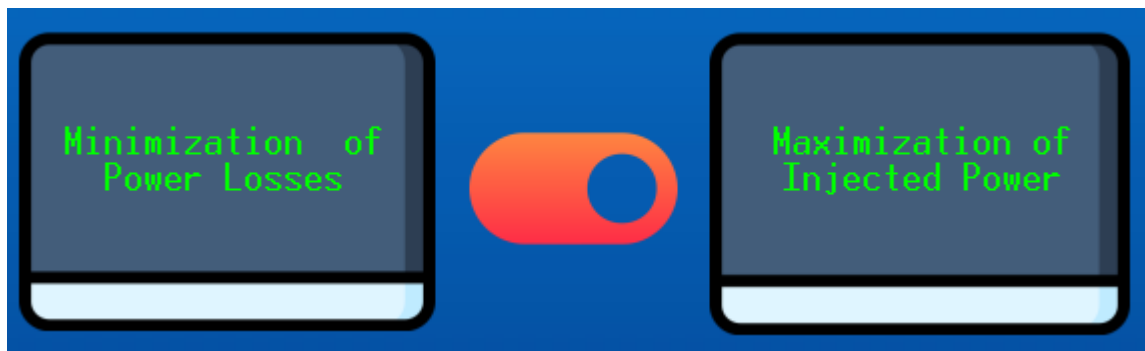
Όσων αφορά την Winlog ο χειριστής θα πρέπει να εισάγει το Project που αναπτύχθηκε από την επιλογή «Import a Project» στο κύριο πάνελ του περιβάλλοντος. Έπειτα θα πρέπει ανοίξει των κώδικα με όνομα «Main» του Project και να αλλαχθούν οι μεταβλητές «output_csv» και «input_csv» στις διευθύνσεις των αρχείων «Winlog_output.csv» και

«Winlog_input.txt» αντίστοιχα (π.Χ. C:\Users\User\Functions\Winlog_output.csv και C:\Users\User\Functions\Winlog_input.txt).

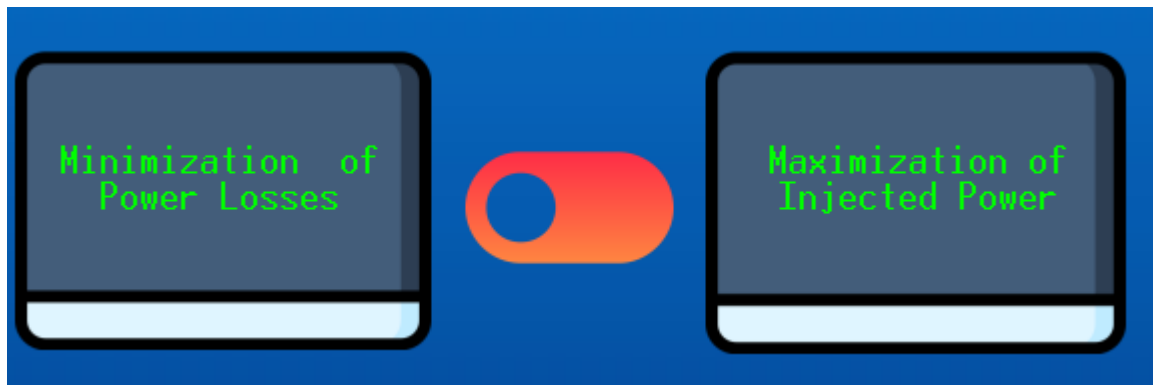
4.4 Εκτέλεση και Λειτουργία της Εφαρμογής

Για την λειτουργία της εφαρμογής θα πρέπει να εκτελεστούν το αρχείο στην Python και το Project στην Winlog. Είναι πολύ σημαντικό να εκτελεστεί το αρχείο στην Python πριν την Winlog ώστε να εκτελεστεί η συνάρτηση `csv_init()` και να αρχικοποιηθούν τα δεδομένα πρώτου προβληθούν. Το .py αρχείο εκτελείται από το περιβάλλον της PyCharm και έχει υλοποιηθεί η δυνατότητα επίβλεψης της εκτέλεσης των συναρτήσεων με την προβολή κατάλληλων μηνυμάτων στο τερματικό.

Κατά την εκτέλεση του Project στην Winlog ο χειριστής βλέπει το περιβάλλον που φαίνεται στην Εικόνα 4. Για να «τρέξει» την προσομοίωση και να προβληθούν τα δεδομένα ο χειριστής θα πρέπει πρώτα να εισάγει τις επιλογές του, δηλαδή, θα πρέπει να επιλέξει εάν επιθυμεί ελαχιστοποίηση των απωλειών ή μεγιστοποίηση της εγχεόμενης ισχύος πατώντας των διακόπτη όπως φαίνεται στις Εικόνες 8 και 9.

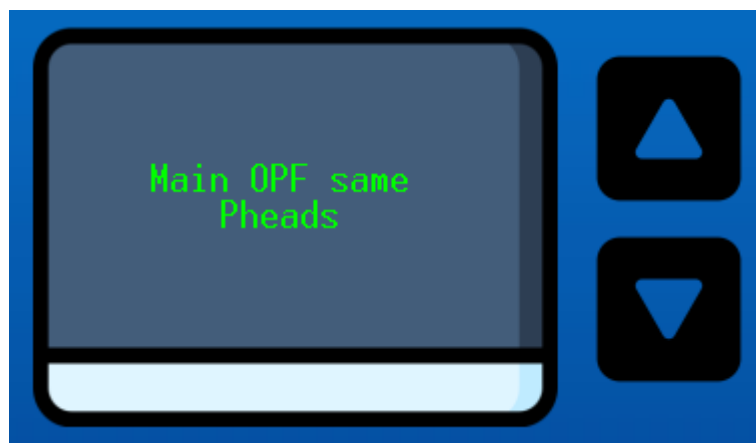


Εικόνα 8 - Επιλογή μεγιστοποίησης εισαγόμενης ισχύος

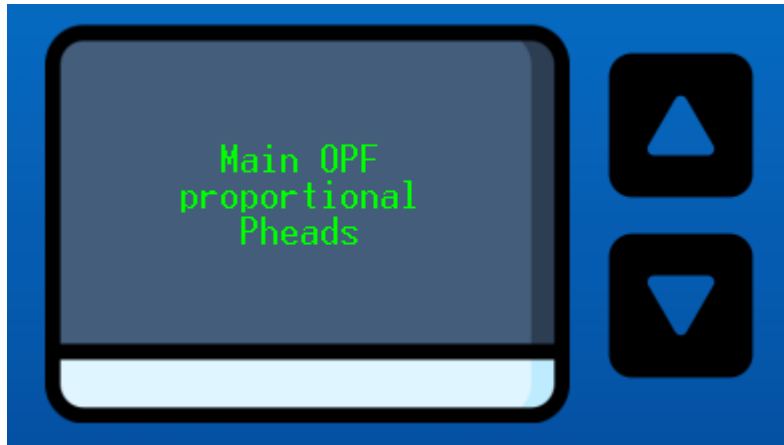


Εικόνα 9 - Επιλογή ελαχιστοποίησης απωλειών συστήματος

Στη συνέχεια θα πρέπει να επιλέξει ποια προσομοίωση θέλει να υλοποιήσει διαλέγοντας από το μενού πατώντας τα πάνω και κάτω βέλη που φαίνονται στις Εικόνες 10, 11 και 12.



Εικόνα 10 - Επιλογή Same Pheads

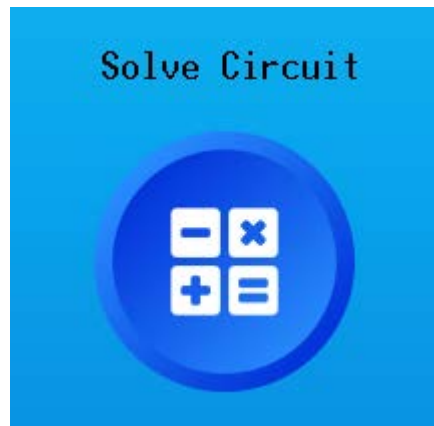


Εικόνα 11 - Επιλογή Proportional Pheads



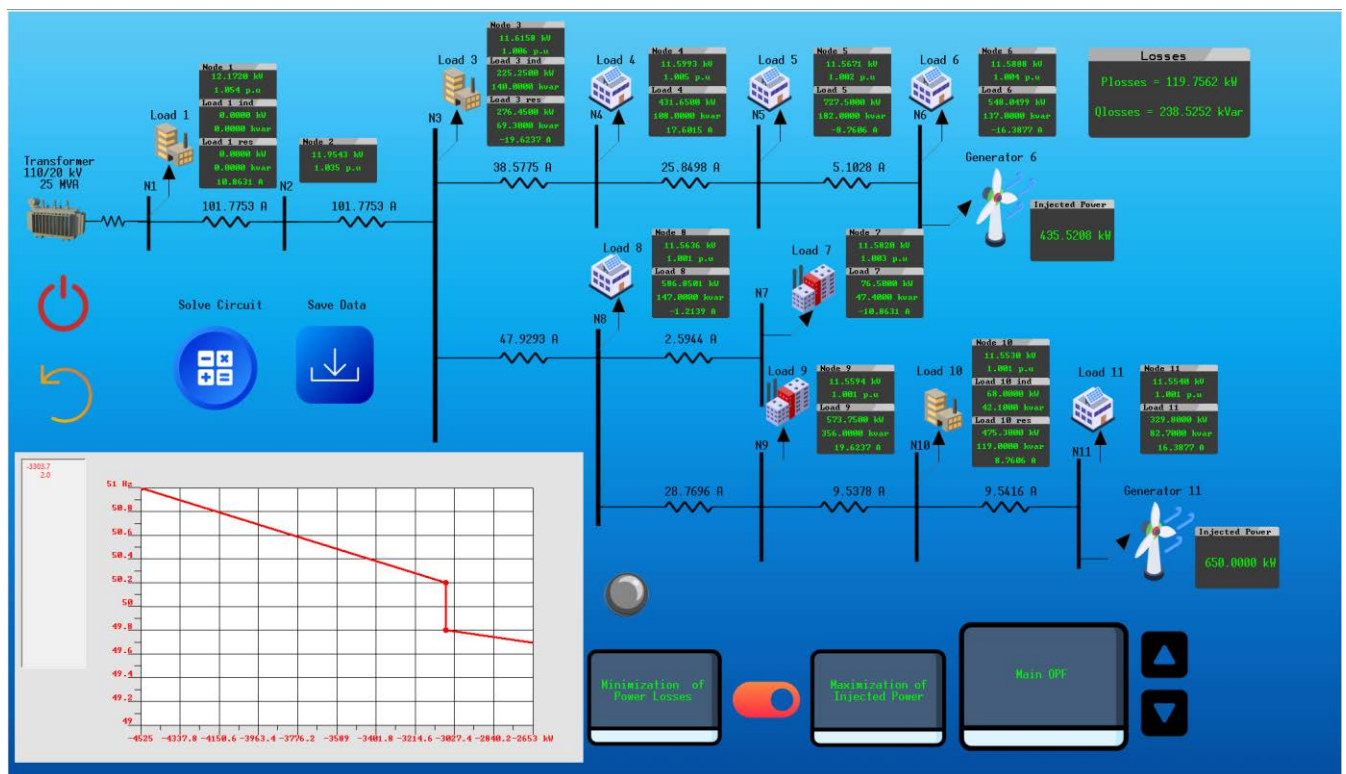
Εικόνα 12 - Επιλογή Standard Pheads

Οι επιλογές είναι διατεταγμένες με την ίδια σειρά που είναι και οι Εικόνες στο κείμενο. Δηλαδή η επιλογή «Main OPF proportional Pheads» είναι “πάνω” από την επιλογή «Main OPF» και η επιλογή «Main OPF same Pheads» είναι “πάνω” από την επιλογή «Main OPF proportional Pheads». Αφού ολοκληρωθούν οι επιλογές η κλήση της προσομοίωσης γίνεται με το πάτημα του κουμπιού «Solve Circuit» που φαίνεται στην Εικόνα 13.



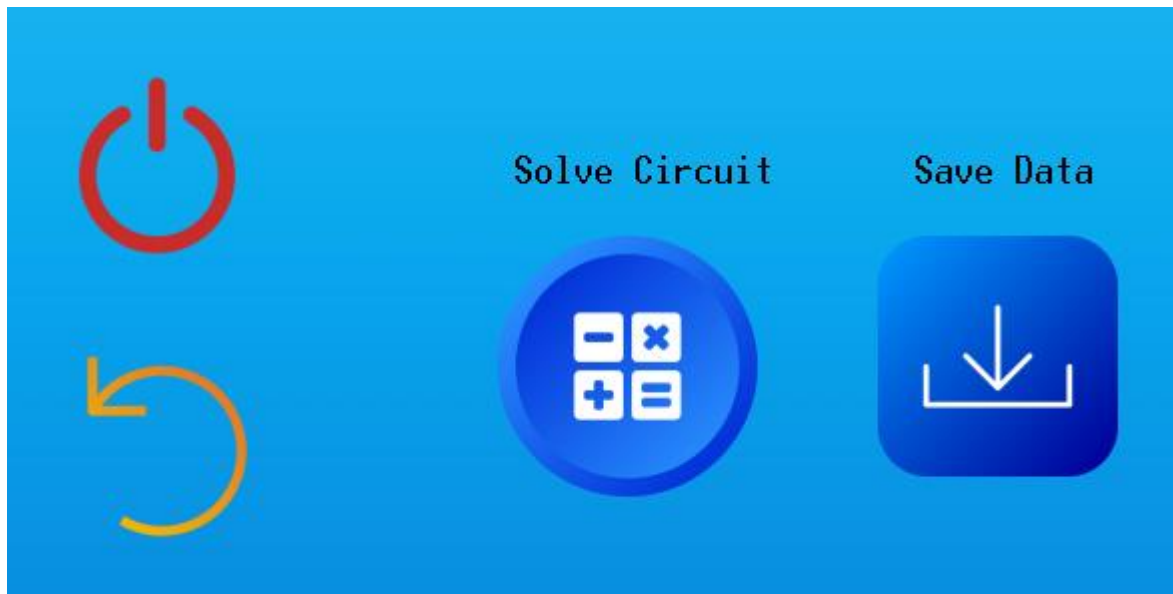
Εικόνα 13 - Solve Circuit κουμπί

Η προσομοίωση έχει ολοκληρωθεί όταν το LED που βρίσκεται δεξιά της γραφικής παράστασης γίνει πράσινο και συμπληρωθούν με μη κενές τιμές όλες οι μεταβλητές. Παράδειγμα εκτέλεσης του προγράμματος παρουσιάζεται στην Εικόνα 14, όπου φαίνονται τα δεδομένα για όλα τα στοιχεία του συστήματος και η καμπύλη στατισμού, με την επιλογή παρακολούθησης των τιμών της με τον κέρσορα.



Εικόνα 14 - Περιβάλλον με αποτελέσματα

Κατά την διάρκεια εκτέλεσης της προσομοίωσης συνίσταται ο χειριστής να μην μεταβάλει οποιαδήποτε κατάσταση των επιλογών του και να μην πατήσει οποιοδήποτε άλλο κουμπί που πραγματοποιεί κάποια ενέργεια. Τα κουμπιά αυτά είναι τα: Solve Circuit, Save Data, Reset και Turn off, τα οποία φαίνονται στην Εικόνα 15.



Εικόνα 15 - Διαθέσιμες ενέργειες χειριστή

Η λειτουργία του «Save Data» είναι η αποθήκευση των δεδομένων που προβάλλονται τη στιγμή που θα πατηθεί σε ένα .csv αρχείο με όνομα την ημερομηνία και την ώρα της στιγμής στον φάκελο «Saved Data». Το «reset» αρχικοποιεί όλες τις μεταβλητές επιστρέφοντας το περιβάλλον σε αυτό της Εικόνας 4. Τέλος, το «turn off» τερματίζει την λειτουργία της Winlog και της Python.

Κεφάλαιο 5 Συμπεράσματα

Στην παρούσα διπλωματική εργασία υλοποιήθηκε η ανάπτυξη μίας πλήρους εφαρμογής SCADA για την επιτήρηση της παροχής πρωτεύουσας ρύθμισης συχνότητας από ενεργά δίκτυα διανομής στο σύστημα μεταφοράς. Στόχος της εφαρμογής είναι η δυνατότητα εποπτείας και παραμετροποίησης της διαδικασίας βελτιστοποίησης σε πραγματικό χρόνο.

Η μελέτη έδειξε πως η επιλογή της συνάρτησης στόχου επηρέασε σημαντικά τα αποτελέσματα των επιβλεπόμενων τιμών και ιδιαίτερα της εγχεόμενης ισχύος. Ακόμη αποδείχθηκε πως η επιλογή του πως θα αντιμετωπιστούν οι μονάδες ΑΠΕ κατά την παροχή της πρωτεύουσας ρύθμισης συχνότητας μπορεί να επηρεάσει σημαντικά τα αποτελέσματα. Συγκεκριμένα, το μοντέλο των ίδιων Phead αποτελεί την αυστηρότερη επιλογή καθώς ισορροπεί την αποκοπή των ΑΠΕ σε μια κοινή τιμή ενώ το μοντέλο των ανάλογων Phead αποτελεί το δικαιότερο μοντέλο καθώς η αποκοπή ισχύος εξαρτάται από το μέγεθος της κάθε ΑΠΕ.

Μέσω της υλοποίησης της εφαρμογής αναδείχθηκε η χρησιμότητα και οι δυνατότητες των λειτουργικών προγραμμάτων που αξιοποιήθηκαν. Κυρίαρχο ρόλο έχει το λειτουργικό περιβάλλον Winlog Eno 4.0, μέσω του οποίου υλοποιήθηκε το γραφικό περιβάλλον της εφαρμογής, παρέχοντας τη δυνατότητα δημιουργίας ενός πλήρους προσαρμόσιμου γραφικού περιβάλλοντος για την προβολή των δεδομένων. Ακόμη, αποδείχθηκε πως τα προγραμματιστικά περιβάλλοντα των OpenDSS και MATLAB είναι τα κατάλληλα εργαλεία για την μοντελοποίηση του δικτύου και την υλοποίηση της διαδικασίας βελτιστοποίησης αντίστοιχα, καθώς προσφέρουν δυνατότητα εύκολης επικοινωνίας μεταξύ τους επιλύοντας τις προσομοιώσεις των δικτύων σε σχεδόν πραγματικό χρόνο. Τέλος, η ευρεία γκάμα εργαλείων σε συνδυασμό με την ευκολία ανάπτυξης και χρήσης, καθιστούν την γλώσσα προγραμματισμού Python την βέλτιστη επιλογή για την ασφαλή και γρήγορη μεταφορά των δεδομένων μεταξύ όλων των εμπλεκόμενων λογισμικών.

Παρά τα πλεονεκτήματα της εφαρμογής πρέπει να σημειωθεί πως υπάρχει περιθώριο βελτίωσης τόσο στο τεχνικό όσο και στο ερευνητικό κομμάτι της. Σχετικά με το πρακτικό κομμάτι της εφαρμογής, είναι εμφανές πως η διαδικασία εγκατάστασης όλων των απαραίτητων εργαλείων δεν είναι απλή και απαιτείται πραγματοποίηση σε κάθε διαφορετική συσκευή. Επιπλέον, η λειτουργία της κοστίζει τόσο σε αποθηκευτικό χώρο όσο

σε υπολογιστική ισχύ της εκάστοτε συσκευής. Μια πιθανή επίλυση αυτού του προβλήματος θα ήταν η υλοποίηση της εφαρμογής σε μια web εκδοχή της, μια επιλογή η οποία προσφέρεται από το λειτουργικό Winlog Eno 4.0, ώστε να είναι προσβάσιμη από οποιαδήποτε συσκευή. Παράλληλα θα πρέπει να αναπτυχθεί μια cloud τύπου υπηρεσία η οποία θα είναι υπεύθυνη για τον υπολογισμό των προβαλλόμενων τιμών, ώστε η εφαρμογή να μην εξαρτάται από την εκάστοτε συσκευή. Στο κομμάτι της έρευνας σε πρώτο στάδιο μια μικρή αλλά σημαντική βελτίωση θα ήταν η προσθήκη περισσότερων παραμέτρων που θα συμβάλουν στην διαδικασία βελτιστοποίησης, όπως για παράδειγμα η εκάστοτε τιμή του ρεύματος. Σε μεγαλύτερης κλίμακα έρευνα, αλγόριθμοι αναπτυγμένοι με την χρήση μηχανικής μάθησης θα μπορούσαν να εξελίσσουν σημαντικά την χρησιμότητα της εφαρμογής. Με την δυνατότητα πρόβλεψης του φορτίου και των μεταβολών της παραγωγής η διαδικασία θα μπορεί να προσαρμόζεται σε πραγματικό χρόνο στις αλλαγές με τον βέλτιστο τρόπο.

Συνοψίζοντας, η ανάπτυξη της εφαρμογής SCADA στα πλαίσια της διπλωματικής εργασίας αποδείχθηκε επιτυχής ως προς την ανάπτυξη και την δοκιμή ενός συστήματος εποπτείας και παραμετροποίησης της παροχής πρωτεύουσας ρύθμισης συχνότητας. Το συνολικό περιβάλλον όλων των λειτουργικών συστημάτων και γλωσσών προγραμματισμού αποδείχθηκε αποδοτικό αλλά και επεκτάσιμο. Η παρούσα πρόταση μπορεί να αξιοποιηθεί για την εφαρμογή έρευνας σε ζητήματα που αφορούν την ενεργή συμμετοχή των ΑΠΕ στον έλεγχο του ΣΗΕ και στην έξυπνη διαχείριση της ενέργειας, καθώς και για εκπαιδευτικούς σκοπούς στα πλαίσια πανεπιστημιακών μαθημάτων.

Βιβλιογραφία

- [1] I. G. Abdul Kareem, S. M. H. Rizvi, S. Kaduvelty, and T. Ghaoud, "An Open-source Automated Distribution Network Analysis (O-ADNA) tool for Modeling and Visualization," in *2023 IEEE IAS Global Conference on Emerging Technologies, GlobConET 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/GlobConET56651.2023.10149941.
- [2] Q. Peng, Y. Yang, T. Liu, and F. Blaabjerg, "Coordination of virtual inertia control and frequency damping in PV systems for optimal frequency support," *CPSS Transactions on Power Electronics and Applications*, vol. 5, no. 4, pp. 305–316, Dec. 2020, doi: 10.24295/CPSSTPEA.2020.00025.
- [3] E. O. Kontis, A. R. Del Nozal, J. M. Mauricio, and C. S. Demoulias, "Provision of Primary Frequency Response as Ancillary Service from Active Distribution Networks to the Transmission System," *IEEE Trans Smart Grid*, vol. 12, no. 6, pp. 4971–4982, Nov. 2021, doi: 10.1109/TSG.2021.3103060.
- [4] Eleftherios O. Kontis, Georgios C. Kryonidis, Andreas I. Chrysochos, Charis S. Demoulias, and Grigoris K. Papagiannis, "Long-term Evaluation of DRES Penetration in LV Networks using Droop Control Techniques," *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE, 2016. doi: 978-1-5090-3358-4/16/\$31.00.
- [5] M. M. Viyathukattuva *et al.*, "Fair Power Curtailment of Distributed Renewable Energy Sources to Mitigate Overvoltages in Low-Voltage Networks." [Online]. Available: <http://www.project-increase.eu/>. pn: 608998.
- [6] C.-D. Dumitru and A. Gligor, "SCADA Based Software for Renewable Energy Management System," *Procedia Economics and Finance*, vol. 3, pp. 262–267, 2012, doi: 10.1016/s2212-5671(12)00150-5.
- [7] A. Soetedjo and Sotyohadi, "Simulation and Visualization of Occupancy-Based Energy Consumption in a Campus Building Using SCADA Software," in *5th IEEE Conference on Energy Conversion, CENCON 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/CENCON51869.2021.9627262.
- [8] N. Mahdavi, T. Ali, J. Aghbolaghi, N. Bizon, and F. Blaabjerg, "Reactive Power Control in AC Power Systems Fundamentals and Current Issues." [Online]. Available: <http://www.springer.com/series/4622>. pp. 136–141.

- [9] Tumininu Lawanson, Roozbeh Karandeh, and Valentina Cecchi, "Improving Power Distribution System Situational Awareness Using Visual Analytics," *2018 IEEE 19th Workshop on Control and Modeling for Power Electronics (COMPEL)*. IEEE, 2018. doi: 978-1-5386-5541-2/18/\$31.00.
- [10] Sielco Sistemi, "Winlog Evo 4.0 Documentation," [Online]. Available: <https://www.sielcosistemi.com/en/download/public/download.html>. Accessed: May 26, 2025.
- [11] Python Software Foundation, "Python 3.13.3 Documentation," Python.org. Last updated May 26, 2025. [Online]. Available: <https://docs.python.org/3/>. Accessed: May 26, 2025.
- [12] MathWorks, Inc., "MATLAB Documentation," The MathWorks Inc., Natick, MA, USA. [Online]. Available: <https://www.mathworks.com/help/matlab/index.html>. Accessed: May 26, 2025.
- [13] Electric Power Research Institute, "OpenDSS Manual," Oct. 2020. [Online]. Available: <https://sourceforge.net/projects/electricdss/files/OpenDSS/OpenDSSManual.pdf>. Accessed: May 26, 2025.
- [14] Anaconda, Inc., "Anaconda Documentation," [Online]. Available: <https://docs.anaconda.com/docs/main>. Accessed: May 26, 2025.