ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΗΛΕΚΤΡΙΚΗΣ ΕΝΕΡΓΕΙΑΣ

ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΙΣΧΥΟΣ, ΑΝΑΝΕΩΣΙΜΗΣ ΚΑΙ ΚΑΤΑΝΕΜΗΜΕΝΗΣ ΠΑΡΑΓΩΓΗΣ

Οικιακός μεσολαβητής για διαχείριση μη-κρίσιμων φορτίων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Παναγιώτη Μαρίνη

Επιβλέπων: Παναγής Βοβός
Αναπληρωτής Καθηγητής

ΠΑΤΡΑ, ΦΕΒΡΟΥΑΡΙΟΣ 2026

Πανεπιστήμιο Πατρών, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών.

Μαρίνης Παναγιώτης

© [2026] – Με την επιφύλαξη παντός δικαιώματος

Το σύνολο της εργασίας αποτελεί πρωτότυπο έργο, παραχθέν από τον Μαρίνη Παναγιώτη, και δεν παραβιάζει δικαιώματα τρίτων καθ' οιονδήποτε τρόπο. Αν η εργασία περιέχει υλικό, το οποίο δεν έχει παραχθεί από τον/την ίδιο/α, αυτό είναι ευδιάκριτο και αναφέρεται ρητώς εντός του κειμένου της εργασίας ως προϊόν εργασίας τρίτου, σημειώνοντας με παρομοίως σαφή τρόπο τα στοιχεία ταυτοποίησής του, ενώ παράλληλα βεβαιώνει πως στην περίπτωση χρήσης αυτούσιων γραφικών αναπαραστάσεων, εικόνων, γραφημάτων κ.λπ., έχει λάβει τη χωρίς περιορισμούς άδεια του κατόχου των πνευματικών δικαιωμάτων για την συμπερίληψη και επακόλουθη δημοσίευση του υλικού αυτού.

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η Διπλωματική Εργασία με τίτλο

ΟΙΚΙΑΚΟΣ ΜΕΣΟΛΑΒΗΤΗΣ ΓΙΑ ΔΙΑΧΕΙΡΙΣΗ ΜΗ-ΚΡΙΣΙΜΩΝ ΦΟΡΤΙΩΝ

Του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

ΠΑΝΑΓΙΩΤΗ ΜΑΡΙΝΗ ΤΟΥ **ΓΕΩΡΓΙΟΥ**

Αριθμός Μητρώου: 1072794

Παρουσιάστηκε δημόσια στο Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών στις

27/02/2026

και εξετάστηκε από την ακόλουθη εξεταστική επιτροπή:

Παναγής Βοβός, Αναπληρωτής Καθηγητής του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών

Δημήτριος Παπαδασκαλόπουλος, Επίκουρος Καθηγητής του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών

Καμπίτσης Γεώργιος, Επίκουρος Καθηγητής του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών

Ο Επιβλέπων	Ο Διευθυντής του Τομέα
Παναγής Βοβός	Καλαντώνης Βασίλειος
Αναπληρωτής Καθηγητής	Καθηγητής

ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κ. Παναγή Βοβό, για την καθοδήγησή του και για την παρότρυνσή του να ασχοληθώ με ένα τόσο επίκαιρο και ενδιαφέρον θέμα. Η επιστημονική του υποστήριξη συνέβαλε ουσιαστικά στην ολοκλήρωση της διπλωματικής εργασίας μου.

Θα ήθελα επίσης να ευχαριστήσω φίλους και συγγενείς, και ιδιαίτερα στους γονείς και στους παππούδες μου, για την οικονομική και ψυχολογική στήριξη που μου παρείχαν καθ' όλη τη διάρκεια των σπουδών μου.

Σας ευχαριστώ όλους θερμά!

Θέμα: «Οικιακός μεσολαβητής για διαχείριση μη-κρίσιμων φορτίων»

ΦΟΙΤΗΤΗΣ: Μαρίνης Παναγιώτης

ΕΠΙΒΛΕΠΩΝ: Βοβός Παναγής

Περίληψη

Η αυξανόμενη διείσδυση των Ανανεώσιμων Πηγών Ενέργειας στα σύγχρονα Συστήματα Ηλεκτρικής Ενέργειας δημιουργεί προκλήσεις στη διατήρηση της ισορροπίας παραγωγής–ζήτησης καθώς και στην ευστάθεια τους. Η στοχαστική τους φύση και η μείωση της διαθέσιμης στρεφόμενης αδράνειας εντείνουν την ανάγκη για ευελιξία στην πλευρά της ζήτησης. Η χρονική και ποσοτική μετατόπιση φορτίου μπορεί να συμβάλει στη διαχείριση των διακυμάνσεων ισχύος και στην αποτελεσματικότερη αξιοποίηση των ΑΠΕ. Στην παρούσα διπλωματική σχεδιάζεται και υλοποιείται ένα πλαίσιο οικιακού μεσολαβητή διαχείρισης μη κρίσιμων φορτίων, βασισμένο σε αρχιτεκτονική πολλών πρακτόρων. Η προσέγγιση συνδυάζει αποκεντρωμένη βελτιστοποίηση φορτίων με κεντρικό συντονισμό σε επίπεδο οικίας. Κάθε φορτίο ενσωματώνει αυτόνομο πράκτορα που προγραμματίζει την οικονομικότερη λειτουργία του, ενώ ο πράκτορας οικίας συγκεντρώνει τα αποτελέσματα και τα εκθέτει σε διαχειριστές δικτύου η φορείς σωρευτικής εκπροσώπησης. Η λειτουργία του πλαισίου αξιολογείται σε μικροδίκτυο ως ενδεικτική περίπτωση χρήσης του εργαλείου. Τα αποτελέσματα δείχνουν μείωση του κόστους για τον τελικό χρήστη, ενώ σε επίπεδο μικροδικτύου επιτυγχάνεται πλήρης αξιοποίηση της τοπικής παραγωγής και οικονομικότερη διαχείριση της ανταλλαγής ισχύος με το δίκτυο. Το προτεινόμενο πλαίσιο μπορεί να εφαρμοστεί σε σχήματα συγκέντρωσης ευελιξίας ή σε ενεργειακές κοινότητες. Η εργασία δομείται ως εξής: αρχικά παρουσιάζεται το πλαίσιο λειτουργίας των σύγχρονων δικτύων και η έννοια της ευελιξίας. Στη συνέχεια αναπτύσσονται τα μοντέλα οικιακών φορτίων και η αρχιτεκτονική των πρακτόρων. Έπειτα διατυπώνεται το πρόβλημα βελτιστοποίησης σε επίπεδο μικροδικτύου και παρουσιάζονται τα αποτελέσματα προσομοίωσης.

Abstract

The increasing penetration of Renewable Energy Sources (RES) in modern Power Systems introduces challenges in maintaining generation–demand balance and overall system stability. Their stochastic nature and the reduction of available rotational inertia intensify the need for flexibility on the demand side. Temporal and quantitative load shifting can contribute to managing power fluctuations and improving the effective utilization of RES. This thesis designs and implements a residential mediator framework for the management of non-critical loads, based on a multi-agent architecture. The proposed approach combines decentralized load optimization with centralized coordination at the household level. Each load integrates an autonomous agent that schedules its operation in a cost-optimal manner, while the house agent aggregates the results and communicates the overall flexibility profile to distribution system operators or aggregators. The framework is evaluated within a microgrid environment as an indicative use case. Simulation results prove cost reduction for the end user, while at the microgrid level full utilization of local generation and more economical power exchange with the main grid are achieved. The proposed framework can be extended to flexibility aggregation schemes or energy communities. The thesis is structured as follows: it first presents the operational context of modern power systems and the concept of demand-side flexibility. It then develops residential load models and the multi-agent architecture. Finally, the microgrid-level optimization problem is formulated and the simulation results are presented.

Πρόλογος

Στην παρούσα διπλωματική εργασία μελετάται η ανάπτυξη ενός οικιακού μεσολαβητή για τη διαχείριση μη-κρίσιμων φορτίων, με στόχο την αξιοποίηση της ευελιξίας στην πλευρά της ζήτησης. Η εργασία συνδυάζει μοντελοποίηση φορτίων, πολυπρακτορική αρχιτεκτονική και αλγόριθμο βελτιστοποίησης, εξετάζοντας την εφαρμογή του πλαισίου σε περιβάλλον μικροδικτύου.

Η διπλωματική δομείται ως εξής:

Στο **Κεφάλαιο 1** παρουσιάζεται το πλαίσιο της ενεργειακής μετάβασης και οι προκλήσεις που προκύπτουν από την αυξημένη διείσδυση ΑΠΕ. Εισάγονται βασικές έννοιες όπως η διαχείριση ζήτησης, τα μη κρίσιμα φορτία, τα μικροδίκτυα και διατυπώνεται ο στόχος της εργασίας.

Στο **Κεφάλαιο 2** αναλύεται η έννοια της ευελιξίας στην οικιακή κατανάλωση και τα χαρακτηριστικά της. Παρουσιάζονται τα έξυπνα φορτία, τα οικιακά συστήματα ενεργειακής διαχείρισης και τα κίνητρα παροχής ευελιξίας μέσω πολιτικών δυναμικής τιμολόγησης.

Στο **Κεφάλαιο 3** αναπτύσσονται γενικευμένα μαθηματικά μοντέλα για την ποσοτικοποίηση της ευελιξίας και τη μοντελοποίηση οικιακών φορτίων. Τα μοντέλα στη συνέχεια εξειδικεύονται σε θερμοσίφωνες, συστήματα άντλησης νερού και μπαταρίες, διότι θα χρησιμοποιηθούν στην δοκιμή του προτεινόμενου πλαισίου ως παραδείγματα ελεγχόμενων μη κρίσιμων οικιακών φορτίων.

Στο **Κεφάλαιο 4** παρουσιάζεται η αρχιτεκτονική πολλών πρακτόρων για τη διαχείριση των οικιακών φορτίων. Διατυπώνεται το πρόβλημα βελτιστοποίησης με δυναμικό προγραμματισμό, περιγράφεται ο μηχανισμός εξαγωγής ευελιξίας και αναλύεται ο κεντρικός συντονισμός.

Στο **Κεφάλαιο 5** διατυπώνεται το πρόβλημα Οικονομικής Κατανομής Φορτίου σε μικροδίκτυο με ενσωμάτωση οικιακής ευελιξίας. Παρουσιάζεται η μαθηματική μοντελοποίηση των μονάδων παραγωγής, της ανταλλαγής ισχύος με το δίκτυο και της συνάρτησης που μοντελοποιεί το στόχο του διαχειριστή του συστήματος στον οποίο προσφέρεται η οικιακή ευελιξία .

Στο **Κεφάλαιο 6** παρουσιάζονται τα σενάρια προσομοίωσης και αναλύονται τα αποτελέσματα τόσο σε επίπεδο οικίας όσο και σε επίπεδο μικροδικτύου. Αξιολογείται η απόδοση του προτεινόμενου πλαισίου υπό διαφορετικές συνθήκες λειτουργίας και τιμολόγησης.

Τέλος στο **Κεφάλαιο 7** γίνεται ανακεφαλαίωση της διπλωματικής εργασίας και αποτιμάται η συνεισφορά του προτεινόμενου πλαισίου για διαχείριση μη κρίσιμων φορτίων. Επιπλέον, συζητούνται πιθανές μελλοντικές βελτιώσεις και προεκτάσεις της προσέγγισης, καθώς και προσωπικά συμπεράσματα από τη διαδικασία εκπόνησης της εργασίας.

Περιεχόμενα

1 Εισαγωγή 11

1.1 Ενεργειακή μετάβαση 11
1.2 Ισορροπία παραγωγής-ζήτησης και διαχείριση της ζήτησης 13
1.3 Μη κρίσιμα οικιακά φορτία 15
1.4 Μικροδίκτυα 16
1.5 Συστήματα ενεργειακής διαχείρισης 18
1.6 Ευελιξία στη ζήτηση 19
1.7 Η σύγχρονη οικία 20
1.8 Στόχος Διπλωματικής Εργασίας 21

2 Ευελιξία στην Οικιακή Κατανάλωση 25

2.1 Εισαγωγή 25
2.2 Χαρακτηριστικά της ευελιξίας 25
2.3 Μη κρίσιμα και έξυπνα φορτία 28
2.4 Οικιακά Συστήματα Ενεργειακής Διαχείρισης 30
2.5 Πολιτική Τιμολόγησης και Κίνητρα για Παροχή Ευελιξίας 30
2.5.1 Οφέλη δυναμικής τιμολόγησης 31
2.6 Συμπεράσματα 31

3 Ποσοτικοποίηση ευελιξίας και μοντελοποίηση φορτίων 33

3.1 Εισαγωγή 33
3.2 Γενικευμένα μοντέλα φορτίων 33
3.2.1 Φορτία Ελεγχόμενης Θερμοκρασίας 34
3.2.2 Συστήματα Αποθήκευσης Ενέργειας 35
3.3 Ειδικά μοντέλα φορτίων 35
3.3.1 Θερμοσίφωνας 36
3.3.2 Σύστημα άντλησης νερού 40
3.3.3 Μπαταρία 42
3.4 Ποσοτικοποίηση ευελιξίας 43
3.5 Συμπεράσματα 45

4 Πράκτορες για τη διαχείριση οικιακών φορτίων 47

4.1 Εισαγωγή 47
4.2 Κατηγορίες πρακτόρων και εφαρμογές στην οικιακή ενεργειακή διαχείριση 47

4.3 Αρχιτεκτονική πολλών πρακτόρων 49
4.4 Ζητήματα Ασφάλειας και Ιδιωτικότητας 51
4.5 Μοντελοποίηση με προσέγγιση μαύρου κουτιού 51
4.6 Χρονοπρογραμματισμός εντολών πρακτόρων φορτίου 53
4.6.1 Ορισμός δυναμικού προγραμματισμού 53
4.6.2 Διατύπωση του προβλήματος 54
4.6.3 Εξισώσεις μετάβασης κατάστασης 56
4.6.4 Συνάρτηση κόστους και εξισώσεις Bellman 57
4.6.5 Κατασκευή βέλτιστης τροχιάς 58
4.7 Εξαγωγή προφίλ ευελιξίας 60
4.7.1 Όρια ισχύος μπαταρίας 61
4.7.2 Όρια ισχύος θερμοσίφωνα 62
4.7.3 Όρια ισχύος αντλίας 62
4.8 Κεντρικός συντονισμός 63
4.9 Κυλιόμενος προγραμματισμός βάσει εισερχόμενων δεδομένων πραγματικού χρόνου 65
4.10 Κατανομή εντολής ευελιξίας στα φορτία 65
4.11 Συμπεράσματα 67

5 Οικονομική κατανομή φορτίου σε μικροδίκτυο με αξιοποίηση υπηρεσιών ευελιξίας από οικίες 69

5.1 Εισαγωγή 69
5.2 Οικονομική Κατανομή Φορτίου 69
5.3 Περιγραφή του υπό Μελέτη Μικροδικτύου 70
5.4 Μαθηματική διατύπωση οικονομικής κατανομής φορτίου στο μικροδίκτυο 71
5.4.1 Μοντελοποίηση οικιακής ευελιξίας 72
5.4.2 Μοντελοποίηση μονάδων παραγωγής 72
5.4.3 Ανταλλαγή ισχύος με το κεντρικό δίκτυο 75
5.5 Συνάρτηση στόχος 76
5.6 Συμπεράσματα 77

6 Προσομοιώσεις και αποτελέσματα 79

6.1 Εισαγωγή 79
6.2 Αυτόνομη λειτουργία πρακτόρων οικιακών φορτίων 79
6.2.1 Σενάριο 1ο : Σταθερή τιμολόγηση 81
6.2.2 Σενάριο 2ο : Δυναμική τιμολόγηση 83
6.3 Οικονομική κατανομή φορτίου με αξιοποίηση οικιακής ευελιξίας 86

6.3.1 Σενάριο 1ο : Πλεόνασμα φωτοβολταϊκής παραγωγής 86
6.3.2 Σενάριο 2ο : Περιορισμένη παραγωγή Φ/Β 91
6.3.3 Σενάριο 3ο : Μεγιστοποίηση εσόδων Μικροδικτύου 92
6.4 Συμπεράσματα 96
7 Συμπεράσματα Διπλωματικής Εργασίας 99
7.1 Ανακεφαλαίωση 99
7.2 Μελλοντικές βελτιώσεις και προεκτάσεις 100
7.3 Προσωπικά συμπεράσματα 101
8 Βιβλιογραφία 103
9 Παράρτημα 107
9.1 Κώδικας υλοποίησης οικιακού μεσολαβητή 107
9.1.1 Λύτης δυναμικού προγραμματισμού 107
9.1.2 Διεπαφή υλοποίησης πράκτορα φορτίου 110
9.1.3 Πράκτορας θερμοσίφωνα 112
9.1.4 Πράκτορας μπαταρίας 119
9.1.5 Πράκτορας αντλίας 126
9.1.6 Πράκτορας οικίας 133
9.2 Κώδικας επίλυσης Οικονομικής Κατανομής Φορτίου 139
9.2.1 Αρχείο επίλυσης οικονομικής κατανομής φορτίου 139
9.3 Κύριο αρχείο προσομοίωσης οικιακού πολυπρακτορικού συστήματος 143

Ευρετήριο Εικόνων

Εικόνα 1 Μηνιαία παραγωγή ενέργειας από ΑΠΕ [4]. 11

Εικόνα 2 Εξέλιξη των συστημάτων ηλεκτρικής ενέργειας, από τα παραδοσιακά κεντρικά συστήματα, στη σημερινή μεταβατική κατάσταση και στα μελλοντικά ψηφιοποιημένα δίκτυα [5]. 12

Εικόνα 3 Eurostat: Αύξηση ηλεκτρικών οχημάτων [7]..... 12

Εικόνα 4 Eurostat: Κατανάλωση ενέργειας στις Ευρωπαϊκές οικίες [8]. 13

Εικόνα 5 Κεντρική και Διεσπαρμένη παραγωγή ενέργειας [11]. 15

Εικόνα 6 Ενδεικτική αρχιτεκτονική μικροδικτύου με τοπικούς ελεγκτές και κεντρικό ελεγκτή μικροδικτύου [15]. 17

Εικόνα 7 Σύγχρονη οικία [20]. 21

Εικόνα 8 Γραφική απεικόνιση υπηρεσίας ευελιξίας [23]..... 27

Εικόνα 9 Έξυπνο φορτίο και μπλοκ ελέγχου [5]. 29

Εικόνα 10 Ηλεκτρικός θερμοσίφωνας: (α) Απεικόνιση βασικών μερών του (β) Σχηματική αναπαράσταση του θερμοδυναμικού RC μοντέλου του [31]..... 37

Εικόνα 11 Προφίλ κατανάλωσης νερού [34]..... 40

Εικόνα 12 Σύστημα άντλησης νερού [35]. 41

Εικόνα 13 Αρχιτεκτονική πολλών πρακτόρων. 49

Εικόνα 14 Πράκτορας φορτίου. 50

Εικόνα 15 Σχηματική αναπαράσταση προβλήματος δυναμικού προγραμματισμού. . 54

Εικόνα 16 Αλγόριθμος δυναμικού προγραμματισμού. 59

Εικόνα 17 Προφίλ ευελιξίας. 61

Εικόνα 18 Ακολουθιακό διάγραμμα ημερήσιου προγραμματισμού. 64

Εικόνα 19 Κατανομή εντολής ευελιξίας και κυλιόμενος προγραμματισμός. 67

Εικόνα 20 Μικροδίκτυο υπό μελέτη. 71

Εικόνα 21 Τμηματικά γραμμική προσέγγιση τετραγωνικής συνάρτησης. 75

Εικόνα 22 Σταθερή τιμολόγηση: αποτελέσματα οικίας. 82

Εικόνα 23 Σταθερή τιμολόγηση: Κατανομή εντολής ευελιξίας. 82

Εικόνα 24 Σταθερή τιμολόγηση: λειτουργία φορτίων. 83

Εικόνα 25 Δυναμική τιμολόγηση: αποτελέσματα οικίας. 84

Εικόνα 26 Δυναμική τιμολόγηση: Κατανομή εντολής ευελιξίας. 84

Εικόνα 27 Δυναμική τιμολόγηση: λειτουργία φορτίων. 85

Εικόνα 28 Σενάριο 1ο: Προσφορά ευελιξίας κατοικίας και ενεργοποίηση σε πραγματικό χρόνο 88

Εικόνα 29 Σενάριο 1ο: Κατανομή ισχύος μικροδικτύου σε πραγματικό χρόνο 89

Εικόνα 30 Σενάριο 1ο: Κατανομή ισχύος μικροδικτύου σε πραγματικό χρόνο χωρίς ευελιξία. 90

Εικόνα 31 Σενάριο 2ο: Προσφορά ευελιξίας κατοικίας και ενεργοποίηση σε πραγματικό χρόνο. 91

Εικόνα 32 Σενάριο 2ο: Κατανομή ισχύος μικροδικτύου σε πραγματικό χρόνο. 92

Εικόνα 33 Σενάριο 3ο: Προσφορά ευελιξίας κατοικίας και ενεργοποίηση σε πραγματικό χρόνο. 93

Εικόνα 34 Σενάριο 3ο: Κατανομή ισχύος μικροδικτύου σε πραγματικό χρόνο. 94

Εικόνα 35 Σενάριο 3ο: Κατανομή ισχύος μικροδικτύου σε πραγματικό χρόνο χωρίς ευελιξία. 95

Ευρετήριο Πινάκων

Πίνακας 1 Ακρωνύμια 9
Πίνακας 2 Σύμβολα 9
Πίνακας 3 Χαρακτηριστικά Ευελιξίας 26
Πίνακας 4 Δυνατότητες και περιγραφή ευελιξίας ανά φορτίο που μελετήθηκε 44
Πίνακας 5 Διακριτοποίηση καταστάσεων συστήματος 56
Πίνακας 6 Παράμετροι πρακτόρων φορτίων 81
Πίνακας 7 Παραμετροποίηση μικροδικτύου 87
Πίνακας 8 Σενάριο 1ο: Σύγκριση με και χωρίς οικιακή ευελιξία. 90
Πίνακας 9 Σενάριο 3ο: Σύγκριση με και χωρίς οικιακή ευελιξία. 95

Ορολογία

Πίνακας 1 Ακρωνύμια

Ακρωνύμιο	Επεξήγηση
ΑΠΕ	Ανανεώσιμες Πηγές Ενέργειας
ΔΕΔΔΗΕ	Διαχειριστής Ελληνικού Δικτύου Διανομής Ηλεκτρικής Ενέργειας
ΔΕΠ	Διεσπαρμένοι Ενεργειακοί Πόροι
ΔΜ	Διαχειριστής Μικροδικτύου
ΔΠ	Δυναμικός Προγραμματισμός
ΟΚΦ	Οικονομική Κατανομή Φορτίου
ΟΣΕΔ	Οικιακό Σύστημα Ενεργειακής Διαχείρισης
ΣΕΔ	Σύστημα Ενεργειακής Διαχείρισης
ΣΗΕ	Συστήματα Ηλεκτρικής Ενέργειας
Φ/Β	Φωτοβολταϊκά
ΦοΣΕ	Φορέας Σωρευτικής Εκπροσώπησης

Πίνακας 2 Σύμβολα

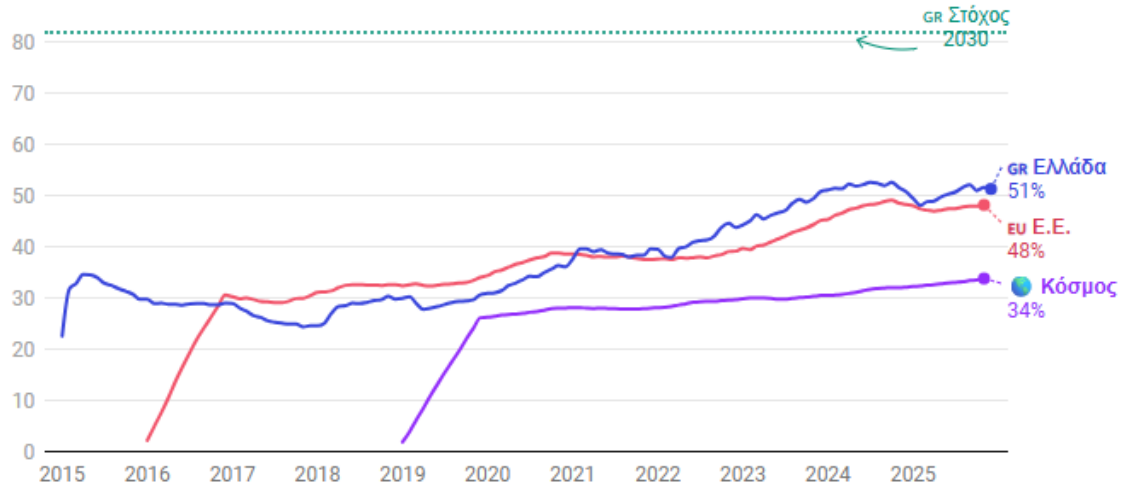
Σύμβολο	Ορισμός	Μονάδα μέτρησης
c_{flex}	Κόστος ενεργοποίησης ευελιξίας	€/kW
C_w	Ισοδύναμη θερμική χωρητικότητα δεξαμενής	J/°C
C_{water}	Ειδική θερμότητα του νερού	J/°C
E_{cap}	Ονομαστική ενεργειακή χωρητικότητα μπαταρίας	J
g	Επιτάχυνση βαρύτητας	m/s²
h	Μανομετρικό ύψος αντλίας	m
L	Ανελαστική ζήτηση φορτίου	W
R_{ewh}	Ισοδύναμη θερμική αντίσταση	°C/W

V_{tank}	Όγκος θερμοσίφωνα	Λίτρα
ρ	Πυκνότητα νερού	kg/m^3
η	Βαθμός απόδοσης	-

1 Εισαγωγή

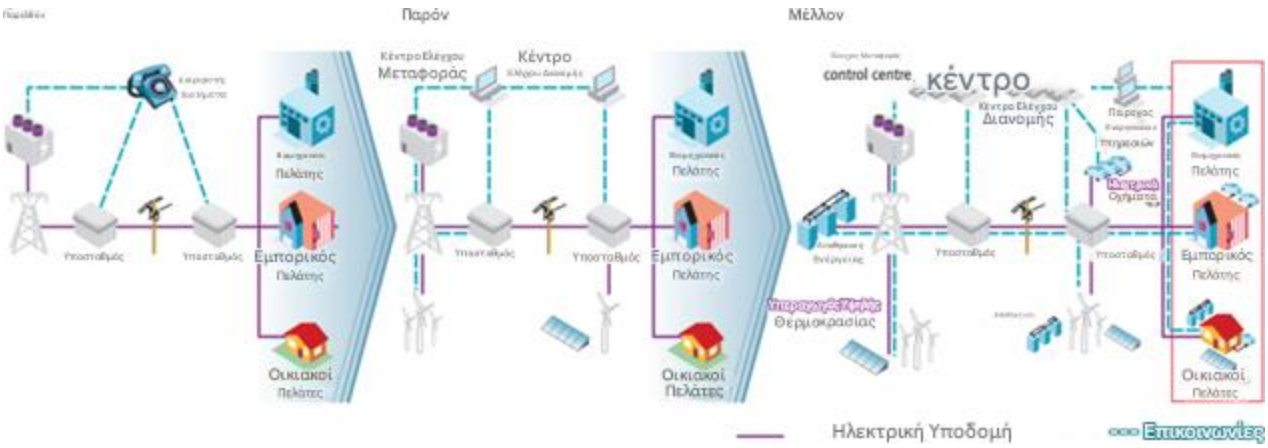
1.1 Ενεργειακή μετάβαση

Η ενεργειακή μετάβαση αναφέρεται στον μακροπρόθεσμο μετασχηματισμό του ενεργειακού συστήματος από ένα μοντέλο βασισμένο κυρίως στα ορυκτά καύσιμα προς ένα σύστημα χαμηλών ή μηδενικών εκπομπών άνθρακα. Στο ευρωπαϊκό πλαίσιο, η ενεργειακή μετάβαση υποστηρίζεται από πολιτικές και κανονισμούς που στοχεύουν στην επίτευξη κλιματικής ουδετερότητας έως το 2050 [1] και ήδη έχει οδηγήσει σε ραγδαία αύξηση της διείσδυσης των Ανανεώσιμων Πηγών Ενέργειας (ΑΠΕ) στο ενεργειακό μείγμα. Το 2023, η παραγωγή ηλεκτρικής ενέργειας από ΑΠΕ στην Ευρωπαϊκή Ένωση ανήλθε στο 45.3% της συνολικής παραγωγής, παρουσιάζοντας αύξηση κατά 4.1% σε σχέση με το 2022 [2]. Το 2024 η αύξηση συνεχίστηκε κατά 2% [3]. Στην Ελλάδα, όπως απεικονίζεται στην Εικόνα 1, το ποσοστό συμμετοχής των ΑΠΕ στην παραγωγή ηλεκτρικής ενέργειας ανήρθε στο 51% για το έτος 2024, ποσοστό μεγαλύτερο του μέσου όρου της Ε.Ε [4].



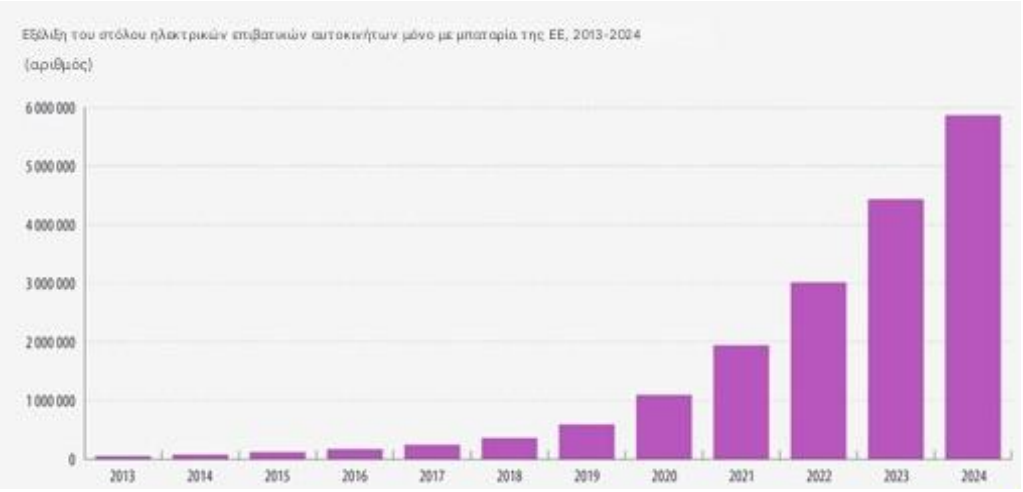
Εικόνα 1 Μηνιαία παραγωγή ενέργειας από ΑΠΕ [4].

Η αυξημένη διείσδυση των ΑΠΕ έχει ως αποτέλεσμα η κάλυψη του φορτίου να γίνεται πιο στοχαστική διαδικασία, καθώς η διαθεσιμότητα τους εξαρτάται από μεταβλητούς και μη ελεγχόμενους παράγοντες. Η στοχαστικότητα αυτή δημιουργεί σημαντικές προκλήσεις για τους διαχειριστές των δικτύων ηλεκτρικής ενέργειας. Οι παραδοσιακές πρακτικές εξισορρόπησης που βασίζονται αποκλειστικά στον απόλυτο έλεγχο της παραγωγής παύουν να εφαρμόζονται, καθιστώντας αναγκαία τη διερεύνηση λύσεων από την πλευρά της ζήτησης ή την αποθήκευση.



Εικόνα 2 Εξέλιξη των συστημάτων ηλεκτρικής ενέργειας, από τα παραδοσιακά κεντρικά συστήματα, στη σημερινή μεταβατική κατάσταση και στα μελλοντικά ψηφιοποιημένα δίκτυα [5].

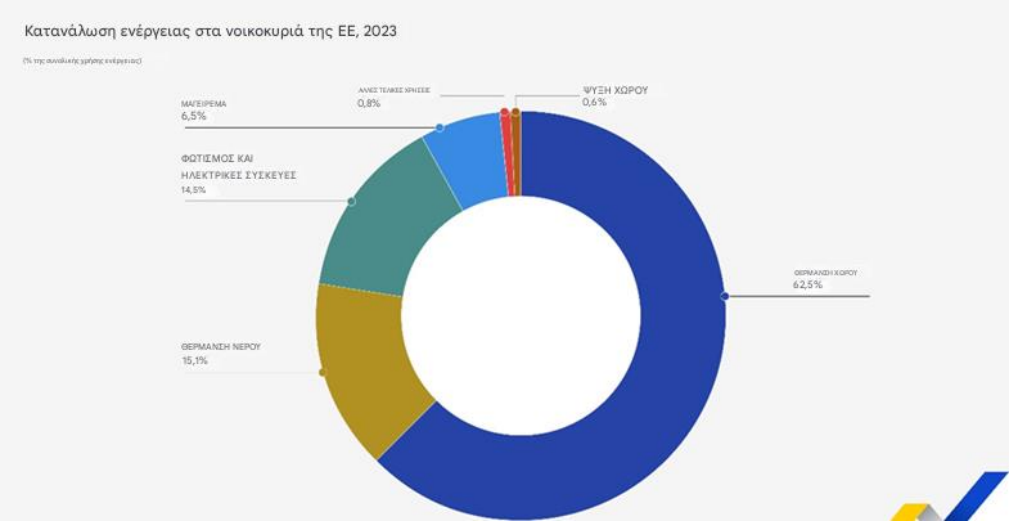
Η αντιμετώπιση των παραπάνω προκλήσεων δεν περιορίζεται μόνο σε τεχνολογίες παραγωγής και αποθήκευσης, αλλά συνδέεται άμεσα με τον ευρύτερο μετασχηματισμό της αρχιτεκτονικής και της λειτουργίας των δικτύων ηλεκτρικής ενέργειας. Η εγκατάσταση έξυπνων μετρητών και η υιοθέτηση οικιακών συστημάτων ενεργειακής διαχείρισης, επιτρέπουν στους καταναλωτές να συμμετέχουν ενεργά στην εξισορρόπηση του δικτύου [6], παρέχοντας υπηρεσίες ευελιξίας στη ζήτηση και συμβάλλοντας ουσιαστικά στη διατήρηση της αξιοπιστίας του συστήματος. Στην Εικόνα 2 παρουσιάζεται ο μετασχηματισμός προς ένα πιο έξυπνο δίκτυο [5].



Εικόνα 3 Eurostat: Αύξηση ηλεκτρικών οχημάτων [7].

Σημαντικές αλλαγές τα τελευταία χρόνια γίνονται και στην πλευρά της ζήτησης. Έχει αυξηθεί σημαντικά τα τελευταία χρόνια, κυρίως λόγω της αύξησης της ηλεκτροκίνησης (Εικόνα 3). Συγκεκριμένα τα έτη, 2023 κ’ 2024 στην ΕΕ αυξήθηκε ο στόλος των ηλεκτρικών οχημάτων κατά 1.55 εκατ. και 1.45 εκατ. οχήματα αντίστοιχα.

Η αυξημένη μεταβλητότητα στην παραγωγή σε συνδυασμό με την αυξημένη ζήτηση εντείνουν την ανάγκη καλύτερου και ταχύτερου ελέγχου του ισοζύγιού ισχύος.



Εικόνα 4 Eurostat: Κατανάλωση ενέργειας στις Ευρωπαϊκές οικίες [8].

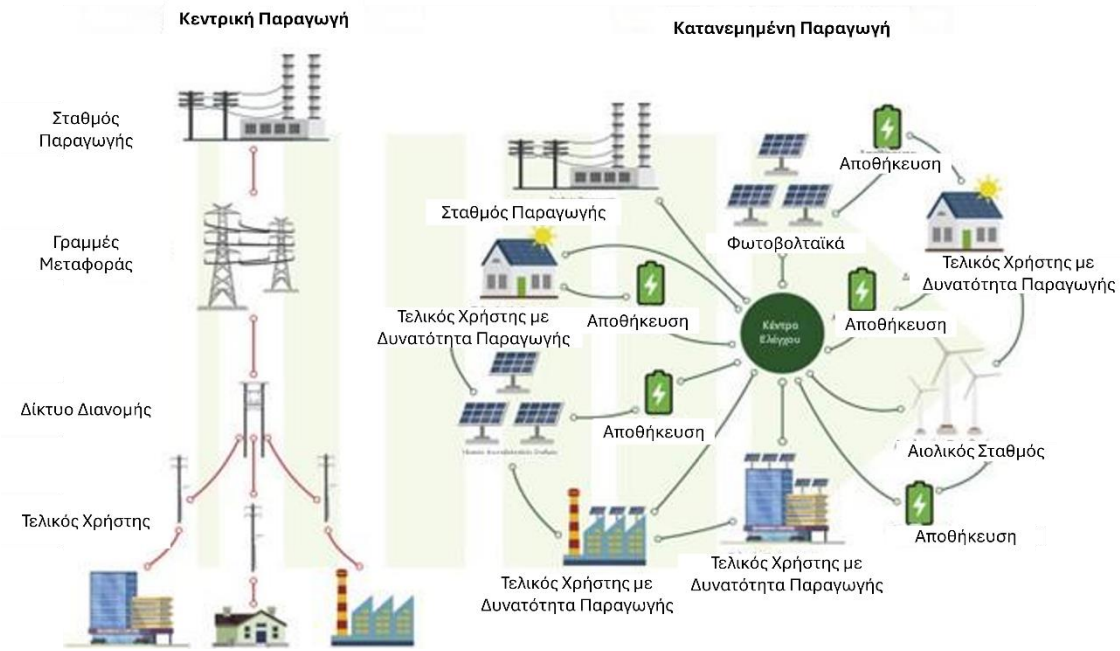
Γενικότερα, ο οικιακός τομέας αποτελεί έναν από τους σημαντικότερους καταναλωτές ηλεκτρικής ενέργειας. Συγκεκριμένα η κατανάλωση ενέργειας, για το έτος 2023, των ευρωπαϊκών νοικοκυριών αντιστοιχούσε στο 26.2% της συνολικής κατανάλωσης ηλεκτρικής ενέργειας, το οποίο αντιστοιχεί σε περίπου 2.67 TkWh [8]. Από την ενέργεια αυτή, το 15.1% αντιστοιχεί σε κατανάλωση ενέργειας για θέρμανση νερού και το 62.5% σε θέρμανση του χώρου (Εικόνα 4) [8], γεγονός που αναδεικνύει τα αντίστοιχα ηλεκτρικά φορτία (όπως θερμοσίφωνες και κλιματιστικά) ως σημαντικά για την διερεύνηση τρόπων χειρισμού τους.

1.2 Ισορροπία παραγωγής-ζήτησης και διαχείριση της ζήτησης

Τα συστήματα ηλεκτρικής ενέργειας (ΣΗΕ) αποτελούν κρίσιμες υποδομές για την υποστήριξη της σύγχρονης κοινωνίας καθώς η ποιότητα ζωής μας και η οικονομική δραστηριότητα εξαρτώνται άμεσα από την καθημερινή χρήση ηλεκτρικής ενέργειας. Η παραδοσιακή αρχιτεκτονική των ΣΗΕ περιλαμβάνει τρία στάδια: παραγωγή, μεταφορά και διανομή της ηλεκτρικής ενέργειας όπου γίνεται και η κατανάλωση από τους τελικούς χρήστες. Στα παραδοσιακά συστήματα, η παραγωγή ηλεκτρικής ενέργειας βασίζονταν κυρίως σε μεγάλες μονάδες (θερμικές, υδροηλεκτρικές κλπ.), οι οποίες τροφοδοτούσαν το δίκτυο μεταφοράς υψηλής τάσης και στην συνέχεια, μέσω σταθμών υποβιβασμού τάσης, η ενέργεια μεταφέρονταν στο δίκτυο διανομής και τελικά στους καταναλωτές [9]. Η λειτουργία των ΣΗΕ τότε, αλλά σε μεγάλο βαθμό

ακόμα και τώρα λόγω της περιορισμένης δυνατότητας αποθήκευσης, βασίζεται στην διατήρηση της ισορροπίας μεταξύ προσφοράς και ζήτησης ενέργειας σε πραγματικό χρόνο. Απόκλιση αυτού του ισοζυγίου ενέργειας μπορεί να οδηγήσει σε διαταραχές συχνότητας και τάσης, προκαλώντας αστάθεια στο δίκτυο και σε ακραίες περιπτώσεις να καταρρεύσει το δίκτυο.

Η αυξανόμενη διείσδυση των ΑΠΕ, όπως η αιολική και η φωτοβολταϊκή ενέργεια, μεταβάλλει θεμελιωδώς τον τρόπο λειτουργίας των ΣΗΕ. Ένα σημαντικό μέρος της παραγωγής ηλεκτρικής ενέργειας βασίζεται πλέον σε τεχνολογίες με εγγενή μεταβλητότητα και περιορισμένη προβλεψιμότητα. Η παραγωγή από ΑΠΕ εξαρτάται άμεσα από μετεωρολογικές συνθήκες, οι οποίες μεταβάλλονται σε διαφορετικές χρονικές κλίμακες (λεπτά, ώρες, ημέρες), δημιουργώντας αβεβαιότητα τόσο στον προγραμματισμό όσο και στη λειτουργία του συστήματος σε πραγματικό χρόνο. Η στοχαστική αυτή συμπεριφορά δυσχεραίνει τη διατήρηση της ισορροπίας παραγωγής–ζήτησης και αυξάνει τις απαιτήσεις για εφεδρείες. Επιπλέον, περιορίζει την αποτελεσματικότητα των συμβατικών μηχανισμών ελέγχου που βασίζονται αποκλειστικά στη ρύθμιση της παραγωγής. Ως αποτέλεσμα, καθίσταται αναγκαία η εισαγωγή συμπληρωματικών μηχανισμών ευελιξίας, ικανών να εξισορροπούν τις διακυμάνσεις της παραγωγής από ΑΠΕ. Παράλληλα με την ραγδαία αύξηση της διείσδυσης των ΑΠΕ, εντείνεται και εισαγωγή των Διεσπαρμένων Ενεργειακών Πόρων (ΔΕΠ), μετασχηματίζοντας τη δομή των ΣΗΕ προς ένα πιο αποκεντρωμένο μοντέλο λειτουργίας. Ως ΔΕΠ ορίζονται οι μονάδες παραγωγής ηλεκτρικής ενέργειας και τα συστήματα αποθήκευσης που συνδέονται απευθείας στα δίκτυα διανομής μέσης ή χαμηλής τάσης, αντί για το δίκτυο μεταφοράς. Στην κατηγορία αυτή περιλαμβάνονται τεχνολογίες όπως, φωτοβολταϊκά συστήματα, μικρές ανεμογεννήτριες, κυψέλες καυσίμου, μικροτουρμπίνες, καθώς και συστήματα αποθήκευσης ενέργειας [10]. Συνεπώς, σε αντίθεση με το κεντρικό μοντέλο παραγωγής, η αποκεντρωμένη παραγωγή βασίζεται σε μικρότερες μονάδες, οι οποίες εγκαθίστανται πλησιέστερα στο φορτίο, συνήθως σε επίπεδο δικτύων διανομής (βλ. Εικόνα 5) [11].



Εικόνα 5 Κεντρική και Διεσπαρμένη παραγωγή ενέργειας [11].

Επομένως, στο νέο αυτό μοντέλο παραγωγής ηλεκτρικής ενέργειας, δημιουργούνται επιπλέον προκλήσεις. Η αυξημένη στοχαστικότητα της παραγωγής εντείνει τις διακυμάνσεις του ισοζυγίου ισχύος. Ταυτόχρονα, η ευελιξία των κεντρικών μονάδων παραγωγής παραμένει περιορισμένη λόγω του ότι καταλαμβάνουν μικρότερο ποσοστό στην παραγωγή. Επιπλέον, η αυξημένη διείσδυση ΔΕΠ στα δίκτυα διανομής μπορεί να οδηγήσει σε φαινόμενα συμφόρησης και τοπικές παραβιάσεις λειτουργικών ορίων. Μια προσέγγιση αντιμετώπισης αυτών των θεμάτων είναι η εξισορρόπηση ισχύος σε τοπικό επίπεδο [11]. Λόγω αυτών των προκλήσεων, αναπτύσσονται εργαλεία στην πλευρά της ζήτησης τα οποία διαχειρίζονται αποτελεσματικά κάποια μη κρίσιμα φορτία ώστε να βελτιωθούν οι δυνατότητες εξισορρόπησης των ΣΗΕ.

1.3 Μη κρίσιμα οικιακά φορτία

Τα οικιακά ηλεκτρικά φορτία διακρίνονται σε κρίσιμα και μη κρίσιμα. Τα κρίσιμα φορτία είναι απαραίτητα για την άμεση εξυπηρέτηση των αναγκών των χρηστών και η λειτουργία τους δεν μπορεί να μεταβληθεί χωρίς να επηρεαστεί η άνεση των χρηστών. Αντίθετα, τα μη κρίσιμα φορτία μπορούν να μετατοπιστούν χρονικά, υπό τον όρο ότι η λειτουργία τους θα ολοκληρωθεί εντός ενός αποδεκτού χρονικού διαστήματος [12].

Χαρακτηριστικά παραδείγματα μη κρίσιμων φορτίων αποτελούν οι συσκευές που λειτουργούν σε διακριτά χρονικά διαστήματα, όπως πλυντήρια ρούχων, πλυντήρια

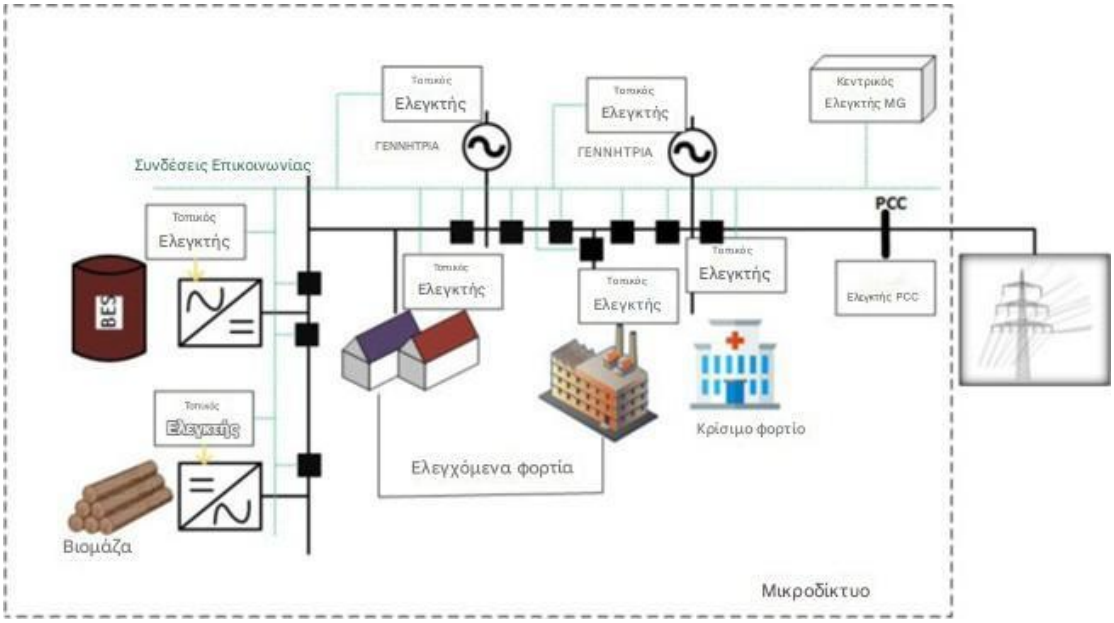
πιάτων και στεγνωτήρια. Η λειτουργία τους μπορεί να καθυστερήσει ή να προγραμματιστεί σε χρονικές περιόδους χαμηλότερης ζήτησης χωρίς ουσιαστική ενόχληση για τους χρήστες. Στην ίδια κατηγορία εντάσσονται επίσης φορτία όπως οι ηλεκτρικοί θερμοσίφωνες και οι φορτιστές ηλεκτρικών οχημάτων, όταν δεν απαιτείται άμεση διαθεσιμότητα του οχήματος [12].

Η διαχείριση των μη κρίσιμων φορτίων προσφέρει σημαντικές δυνατότητες ευελιξίας στα συστήματα ηλεκτρικής ενέργειας. Μέσω κατάλληλων στρατηγικών χρονοπρογραμματισμού, είναι δυνατή η μείωση των αιχμών της ζήτησης, η καλύτερη αξιοποίηση της παραγωγής από ΑΠΕ και η μείωση του συνολικού κόστους ενέργειας για τους καταναλωτές. Για τον λόγο αυτό, τα μη κρίσιμα οικιακά φορτία αποτελούν βασικό πεδίο εφαρμογής σύγχρονων συστημάτων ενεργειακής διαχείρισης και μηχανισμών διαχείρισης της ζήτησης.

1.4 Μικροδίκτυα

Τα μικροδίκτυα αποτελούν ένα οριοθετημένο ηλεκτρικό δίκτυο που περιλαμβάνει καταναλωτές, διεσπαρμένους ενεργειακούς πόρους, μονάδες αποθήκευσης ενέργειας και συστήματα τοπικού ελέγχου και ενεργεί ως μια ενιαία οντότητα [13]. Κάθε επιμέρους στοιχείο συνοδεύεται από έναν τοπικό ελεγκτή (βλ. Εικόνα 6), ο οποίος είναι υπεύθυνος για τη λειτουργία του σε τοπικό επίπεδο, ενώ ο κεντρικός ελεγκτής του μικροδικτύου αναλαμβάνει τον συνολικό συντονισμό και την βελτιστοποίηση της λειτουργίας του συστήματος.

Η λειτουργία ενός μικροδικτύου επιβλέπεται από τον διαχειριστή του μικροδικτύου [14], οποίος ο οποίος έχει την ευθύνη για τη διατήρηση της ισορροπίας ισχύος μεταξύ παραγωγής και ζήτησης. Επιπλέον, είναι αρμόδιος για τη συνεχή παρακολούθηση και τον έλεγχο της κατάστασης του μικροδικτύου. Βασικός του στόχος είναι η μεγιστοποίηση της αξιοποίησης της τοπικής παραγωγής ενέργειας, καθώς και η ελαχιστοποίηση του συνολικού κόστους λειτουργίας του συστήματος. Παράλληλα, ο διαχειριστής του μικροδικτύου δύναται να συντονίζει την παροχή επικουρικών υπηρεσιών προς το κεντρικό δίκτυο, συμβάλλοντας στη συνολική ευστάθεια του ΣΗΕ.



Εικόνα 6 Ενδεικτική αρχιτεκτονική μικροδικτύου με τοπικούς ελεγκτές και κεντρικό ελεγκτή μικροδικτύου [15].

Ένα μικροδίκτυο μπορεί να λειτουργεί είτε συνδεδεμένο με το κεντρικό δίκτυο (grid-connected mode) είτε σε αυτόνομη λειτουργία, δηλαδή «νησιδοποιημένο» (islanded mode), διατηρώντας σταθερά τα επίπεδα τάσης και συχνότητας. Η ικανότητα αυτή, να λειτουργούν στις δύο καταστάσεις, αφενός επιτρέπει στο μικροδίκτυο να αλληλοεπιδρά με την αγορά ενέργειας όταν είναι συνδεδεμένο, αφετέρου να εξασφαλίζει τροφοδότηση κρίσιμων φορτίων κατά την διάρκεια διαταραχών του κεντρικού δικτύου, όπου λειτουργεί αυτόνομα αξιοποιώντας την τοπική παραγωγή και αποθήκευση [14].

Στην διασυνδεδεμένη λειτουργία, το μικροδίκτυο παραμένει συνδεδεμένο με το δίκτυο διανομής μέσω του σημείου κοινής σύνδεσης. Στην κατάσταση αυτή, μπορεί να εισάγει ισχύ για να καλύψει την ζήτηση των τοπικών φορτίων [16], όταν η τοπική παραγωγή δεν αρκεί, ενώ μπορεί να εξάγει ισχύ, όταν προκύπτει πλεονάζουσα ενέργεια από τοπική παραγωγή. Οι απαιτήσεις για σταθερότητα και έλεγχο είναι απλοποιημένες σε αυτή την λειτουργία, διότι οι τοπικές μονάδες παραγωγής που βασίζονται σε μετατροπείς ισχύος, λειτουργούν με σημείο αναφοράς τάσης και συχνότητας που δίνει το κεντρικό δίκτυο [16]. Ακόμα, μπορεί να παρέχει υπηρεσίες ευελιξίας και προσφορές εφεδρειών, καθιστώντας σημαντικό τον έλεγχο των μη κρίσιμων ή λιγότερων σημαντικών οικιακών φορτίων, τα οποία δύναται να προσαρμόσουν την κατανάλωση τους. Αυτή η λειτουργία εστιάζει κυρίως στο οικονομικό όφελος του μικροδικτύου, σε αντίθεση με την αυτόνομη λειτουργία όπου η προτεραιότητα είναι η ασφάλεια τροφοδοσίας.

Στην αυτόνομη λειτουργία, το μικροδίκτυο απομονώνεται από το κεντρικό δίκτυο είτε λόγω σφάλματος είτε λόγω προγραμματισμένης αποσύνδεσης. Η απουσία του κεντρικού δικτύου ως σημείο αναφοράς, απαιτεί πιο πολύπλοκες υπηρεσίες ελέγχου, ιδιαίτερα σε μικρά οικιακά μικροδίκτυα [15]. Επίσης, επειδή η διαθέσιμη ισχύς περιορίζεται από τους τοπικούς πόρους, καθίστανται αναγκαία τα ευέλικτα φορτία ώστε να διασφαλίζεται η ισορροπία ισχύος. Η πολυπλοκότητα της δομής των μικροδικτύων και η ανάγκη συντονισμού των επιμέρους στοιχείων του, όπως φαίνονται στην Εικόνα 6, καθιστούν αναγκαία την ύπαρξη συστημάτων ενεργειακής διαχείρισης, τα οποία αναλύονται στην επόμενη ενότητα.

1.5 Συστήματα ενεργειακής διαχείρισης

Τα συστήματα ενεργειακής διαχείρισης (ΣΕΔ) είναι συστήματα ελέγχου, τα οποία έχουν ως στόχο τη βέλτιστη λειτουργία ενός ενεργειακού συστήματος μέσω της παρακολούθησης, ανάλυσης και ελέγχου της παραγωγής, της κατανάλωσης και της αποθήκευσης ενέργειας. Τα ΣΕΔ βρίσκουν γενικότερα εφαρμογή στα ΣΗΕ, από κεντρικά δίκτυα και βιομηχανικές εγκαταστάσεις έως οικίες, με στόχο τη βελτίωση της αποδοτικότητας και τη μείωση του κόστους λειτουργίας του εκάστοτε ενεργειακού συστήματος.

Βασικές λειτουργίες ενός ΣΕΔ περιλαμβάνουν τη συλλογή και επεξεργασία δεδομένων, την πρόβλεψη της ζήτησης και της παραγωγής, καθώς και τη λήψη αποφάσεων μέσω αλγορίθμων βελτιστοποίησης ή κανόνων ελέγχου. Ιδιαίτερα στα σύγχρονα ΣΗΕ, όπου η στοχαστικότητα της παραγωγής και η ενεργός συμμετοχή των καταναλωτών αυξάνονται, τα ΣΕΔ αποτελούν εργαλείο για την αξιοποίηση της ευελιξίας και την εξισορρόπηση της παραγωγής με τη ζήτηση [17].

Στο πλαίσιο των μικροδικτύων, τα ΣΕΔ καθιστούν εφικτή τη συνεχή παρακολούθηση του μικροδικτύου και την βέλτιστη λειτουργία του. Στα πλαίσια του μικροδικτύου, το κεντρικό ΣΕΔ οφείλει να διαχειρίζεται την τοπική παραγωγή από ΔΕΠ, την αποθήκευση ενέργειας και την αξιοποίηση της ευελιξίας των φορτίων. Η αρχιτεκτονική των ΣΕΔ κατηγοριοποιείται σε κεντρική και κατανεμημένη. Στην κεντρική αρχιτεκτονική, όλα τα δεδομένα (προβλέψεις ζήτησης και παραγωγής, καιρικά δεδομένα) συλλέγονται κεντρικά για να καθοριστεί ο βέλτιστος ενεργειακός προγραμματισμός για το μικροδίκτυο. Στη συνέχεια οι αποφάσεις του ΣΕΔ αποστέλλονται στα τοπικά υποσυστήματα για υλοποίηση [15].

Αντίθετα, στην κατανεμημένη αρχιτεκτονική, κάθε τοπικός ελεγκτής επικοινωνεί με τον κεντρικό ελεγκτή του μικροδικτύου σε πραγματικό χρόνο. Οι τοπικοί ελεγκτές ενημερώνονται ή εκτιμούν τη τρέχουσα και μελλοντική ζήτηση ή παραγωγή. Ο κεντρικός ελεγκτής καθορίζει τον βέλτιστο προγραμματισμό και τον αποστέλλει στους τοπικούς ελεγκτές. Με τη σειρά τους αυτοί τον επεξεργάζονται για να διαπιστώσουν αν ικανοποιούνται οι τοπικοί στόχοι και διαπραγματεύονται συνεχώς με το κεντρικό ελεγκτή διαφοροποιήσεις του προγραμματισμού μέχρι να επιτευχθούν τοπικοί και συνολικοί στόχοι λειτουργίας του μικροδικτύου.

Στο επίπεδο των οικιών, οι τοπικοί ελεγκτές υλοποιούνται μέσω οικιακών συστημάτων ενεργειακής διαχείρισης (ΟΣΕΔ), τα οποία είναι υπεύθυνα για την πρόβλεψη της κατανάλωσης, τον έλεγχο των ευέλικτων φορτίων και την εκτίμηση της διαθέσιμης ευελιξίας. Φαίνεται λοιπόν ότι η συνολική λειτουργία του μικροδικτύου, εξαρτάται από το κεντρικό ΣΕΔ αλλά και τους επιμέρους τοπικούς ελεγκτές και τα ΟΣΕΔ.

1.6 Ευελιξία στη ζήτηση

Η μετάβαση σε αποκεντρωμένα δίκτυα βασίζεται κατά μεγάλο μέρος στην αύξηση συμμετοχής των τελικών καταναλωτών, παρέχοντας υπηρεσίες ευελιξίας στο δίκτυο. Η έννοια της ευελιξίας στην πλευρά της ζήτησης περιγράφει την ικανότητα των οικιακών και βιομηχανικών καταναλωτών να προσαρμόζουν το προφίλ κατανάλωσης ηλεκτρικής ενέργειας σύμφωνα με εξωτερικά σήματα που λαμβάνουν.

Η προσαρμογή αυτή πραγματοποιείται μέσω μηχανισμών Διαχείρισης Ζήτησης, οι οποίοι διακρίνονται σε έμμεσους και άμεσους μηχανισμούς. Στους έμμεσους μηχανισμούς, η απόκριση των καταναλωτών προκύπτει μέσω οικονομικών σημάτων, όπως οι μεταβολές των τιμών ηλεκτρικής ενέργειας. Χαρακτηριστικό παράδειγμα αποτελούν τα δυναμικά (πορτοκαλί) τιμολόγια στα οποία η τιμή της ηλεκτρικής ενέργειας διαμορφώνεται σε ωριαία βάση και αντανακλά τις συνθήκες της χονδρεμπορικής αγοράς. Στους άμεσους μηχανισμούς, η προσαρμογή της ζήτησης πραγματοποιείται κατόπιν αιτημάτων ή εντολών από τους διαχειριστές του δικτύου που έχουν προκαθοριστεί σε συμβάσεις παροχής ευελιξίας [18].

Η αξιοποίηση της ευελιξίας στην πλευρά της ζήτησης, προσφέρει στους διαχειριστές δικτύου τη δυνατότητα να διαχειρίζονται αποτελεσματικότερα την

εξισορρόπηση μεταξύ παραγωγής και ζήτησης. Παράλληλα, συμβάλλει στην αποφυγή αιχμών φορτίου, μετατοπίζοντας τα φορτία σε ώρες χαμηλής ζήτησης, με αποτέλεσμα την αντιμετώπιση προβλημάτων συμφόρησης στα δίκτυα διανομής. Επιπλέον, διευκολύνει την διείσδυση των ΑΠΕ, μέσω καλύτερης προσαρμογής της ζήτησης στη στοχαστική παραγωγή. Οι εφαρμογές της ευελιξίας στη ζήτηση εκτείνονται σε διαφορετικές χρονικές κλίμακες, από τη λειτουργία σε πραγματικό χρόνο έως τον ημερήσιο προγραμματισμό [18].

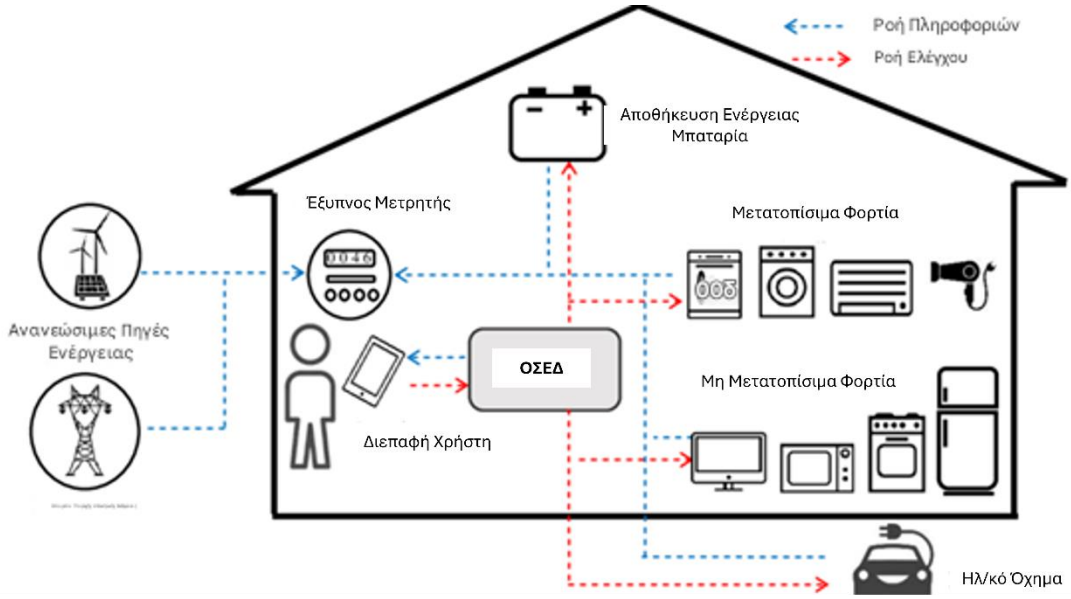
Σε πολύ μικρές χρονικές κλίμακες, της τάξης των δευτερολέπτων, η ευελιξία αξιοποιείται για την εξισορρόπηση απότομων και ταχέων μεταβολών της ισχύος, συμβάλλοντας στη διατήρηση της συχνότητας του συστήματος. Σε χρονικές κλίμακες λεπτών, η ευελιξία στη ζήτηση χρησιμοποιείται για την αντιμετώπιση σφαλμάτων πρόβλεψης τόσο της παραγωγής όσο και του φορτίου [18]. Η δυνατότητα αυτή επιτρέπει τη γρήγορη προσαρμογή της κατανάλωσης σε αποκλίσεις από τον αρχικό προγραμματισμό. Σε μεγαλύτερες χρονικές κλίμακες, που εκτείνονται από ώρες έως και ημέρες, η ευελιξία εφαρμόζεται για την εξισορρόπηση της μεταβλητότητας του «καθαρού» φορτίου. Το καθαρό φορτίο ορίζεται ως το φορτίο του συστήματος μετά την αφαίρεση της παραγωγής από ΑΠΕ. Σε αυτή την περίπτωση, η ζήτηση προσαρμόζεται ώστε να ακολουθεί τις διακυμάνσεις της παραγωγής από ΑΠΕ. Τέλος, σε ακόμη μεγαλύτερες χρονικές κλίμακες, της τάξης των μηνών, η ευελιξία στη ζήτηση μπορεί να συμβάλει στην εξισορρόπηση της εποχιακής διαθεσιμότητας ενέργειας. Η εφαρμογή αυτή σχετίζεται κυρίως με τον μακροχρόνιο ενεργειακό προγραμματισμό και τη βέλτιστη αξιοποίηση των διαθέσιμων ενεργειακών πόρων.

1.7 Η σύγχρονη οικία

Η σύγχρονη οικία αποτελείται από ένα σύνολο διασυνδεδεμένων συστημάτων με κεντρικά στοιχεία τον έξυπνο μετρητή και το ΟΣΕΔ (βλ. Εικόνα 7). Ο έξυπνος μετρητής επιτρέπει τη συνεχή παρακολούθηση της κατανάλωσης και την αμφίδρομη επικοινωνία με τον διαχειριστή δικτύου, ενώ το ΟΣΕΔ συλλέγει δεδομένα από τα επιμέρους συστήματα και λαμβάνει αποφάσεις ελέγχου. Στην Ελλάδα αναμένεται η εγκατάσταση περίπου 3,12 εκατ. έξυπνων μετρητών μέχρι το 2026, με στόχο να αντικατασταθούν όλοι οι μετρητές μέχρι το 2030 [19].

Η σύγχρονη κατοικία μπορεί επίσης να διαθέτει τοπική παραγωγή ηλεκτρικής ενέργειας, όπως φωτοβολταϊκά συστήματα, καθώς και δυνατότητα αποθήκευσης

ενέργειας μέσω συσσωρευτών. Επιπλέον, περιλαμβάνει έξυπνα κ' ευέλικτα οικιακά φορτία, όπως συστήματα θέρμανσης, κλιματισμού, θερμοσίφωνες και ηλεκτρικά οχήματα, τα οποία έχουν ενσωματωμένα ηλεκτρονικά ισχύος με ικανότητες ρύθμισης της κατανάλωσης τους (βλ. Ενότητα 2.3).



Εικόνα 7 Σύγχρονη οικία [20].

Για να συντονιστούν όλα αυτά τα συστήματα χρειάζονται προηγμένες στρατηγικές ελέγχου. Σε αυτό το πλαίσιο, τα επιμέρους φορτία μπορούν να αντιμετωπιστούν ως αυτόνομες οντότητες λήψης αποφάσεων, δηλαδή ως πράκτορες, όπου διαθέτουν τοπική πληροφόρηση, συγκεκριμένους λειτουργικούς περιορισμούς και στόχους και συνεργάζονται με έναν κεντρικό πράκτορα στο ΟΣΕΔ.

1.8 Στόχος Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία επικεντρώνεται στην σχεδίαση και ανάπτυξη ενός οικιακού μεσολαβητή διαχείρισης μη κρίσιμων φορτίων, βασισμένο σε αρχιτεκτονική πολλών πρακτόρων, με αποκεντρωμένη βελτιστοποίηση και κεντρικό συντονισμό. Στο προτεινόμενο πλαίσιο, η ημερήσια βελτιστοποίηση της λειτουργίας των μη-κρίσιμων φορτίων πραγματοποιείται τοπικά από αυτόνομους πράκτορες φορτίου (load agents), ενώ ο πράκτορας οικίας (house agent) αναλαμβάνει τον συντονισμό των επιμέρους πρακτόρων και την συσσώρευση των αποτελεσμάτων, λειτουργώντας ως οικιακός μεσολαβητής μεταξύ οικίας και διαχειριστή δικτύου ή φορέα σωρευτικής εκπροσώπησης (ΦοΣΕ). Ένας πράκτορας φορτίου ενσωματώνεται

σε κάθε συγκεκριμένο οικιακό φορτίο και επιλύει ανεξάρτητα το πρόβλημα ημερήσιου προγραμματισμού με κριτήριο την οικονομικότερη διαχείριση. Ο πράκτορας οικίας είναι εγκατεστημένος στο ΟΣΕΔ και είναι υπεύθυνος για τον συντονισμό των επιμέρους πρακτόρων και τη διαμόρφωση ενός συνολικού προφίλ ευελιξίας. Με την αρχιτεκτονική αυτή, ο οικιακός μεσολαβητής δρα ως ενδιάμεσο επίπεδο μεταξύ των επιμέρους φορτίων και του ανώτερου επιπέδου ελέγχου, επιτρέποντας την αφαίρεση λεπτομερειών από τη προσωπική ζωή του χρήστη (που προκύπτει από τη λειτουργία των φορτίων) και τη διατήρηση της αποκεντρωμένης λήψης αποφάσεων. Οι στόχοι του προτεινόμενου πλαισίου λειτουργίας του οικιακού μεσολαβητή είναι οι εξής:

1. Αποκεντρωμένη βελτιστοποίηση της λειτουργίας των μη-κρίσιμων οικιακών φορτίων μέσω αυτόνομων πρακτόρων, με στόχο τη μείωση του κόστους κατανάλωσης, λαμβάνοντας υπόψη τις τιμές ενέργειας βάση πολιτικής δυναμικής τιμολόγησης και τις προβλεπόμενες ανάγκες κατανάλωσης, μέσω αλγοριθμικής προσέγγισης δυναμικού προγραμματισμού.
2. Κεντρικός συντονισμός των επιμέρους πρακτόρων και διαμόρφωση ενιαίου προφίλ ευελιξίας της οικίας, το οποίο αποστέλλεται στο ανώτερο επίπεδο ελέγχου για την εξισορρόπηση της παραγωγής με τη ζήτηση.

Η υλοποίηση ενός τέτοιου συστήματος συμβάλλει στην αποκεντρωμένη αξιοποίηση της ευελιξίας στην πλευρά της ζήτησης, επιτρέποντας την ενεργό συμμετοχή των κατοικιών στη λειτουργία των ηλεκτρικών δικτύων. Από τη μια πλευρά ο τελικός καταναλωτής αποκομίζει οικονομικό όφελος μέσω της βελτιστοποίησης της κατανάλωσης του και της παροχής υπηρεσιών ευελιξίας στο δίκτυο. Από την άλλη πλευρά, ο διαχειριστής του δικτύου αποκτά ποσοτικοποιημένες υπηρεσίες ευελιξίας, τις οποίες αξιοποιεί για τη βέλτιστη λειτουργία του δικτύου και την εξασφάλιση της αξιοπιστίας του.

Όπως αναφέρθηκε στις προηγούμενες ενότητες, η Ευρωπαϊκή Ένωση στοχεύει να αυξηθεί το ποσοστό παραγωγής ΑΠΕ, αλλά η στοχαστική τους φύση ελλοχεύει κινδύνους για τα ηλεκτρικά δίκτυα. Η εκμετάλλευση της ευελιξίας στην πλευρά της ζήτησης καθίσταται αναγκαία ως εργαλείο αντιμετώπισης της στοχαστικότητας στην παραγωγή, διότι υπάρχει η δυνατότητα χρονικής και ποσοτικής μετατόπισης του φορτίου σύμφωνα με το προφίλ παραγωγής των ΑΠΕ. Παράλληλα, η εγκατάσταση έξυπνων μετρητών σε συνδυασμό με τους μηχανισμούς δυναμικής τιμολόγησης, όπως

είναι και το «πορτοκαλί τιμολόγιο» που υπάρχει στο Ελληνικό πλαίσιο, επιτρέπουν την εφαρμογή έξυπνων πρακτόρων που μπορούν να λαμβάνουν αυτόνομες αποφάσεις για βελτιστοποίηση της κατανάλωσης.

Ένας οικιακός πράκτορας διαχείρισης ευελιξίας, μπορεί γενικότερα να αξιοποιηθεί ποικιλοτρόπως εντός του ΣΗΕ. Ενδεικτικά, μπορεί να συμμετέχει σε σχήματα συγκέντρωσης ευελιξίας από πολλαπλές οικίες μέσω ΦοΣΕ. Ακόμα, μπορεί να παρέχει υπηρεσίες ευελιξίας απευθείας στον διαχειριστή δικτύου διανομής (ΔΕΔΔΗΕ στο ελληνικό πλαίσιο) ή να λειτουργεί στο πλαίσιο ενός μικροδικτύου, όπου η ευελιξία αξιοποιείται για την βελτιστοποίηση της τοπικής λειτουργίας.

Στην παρούσα διπλωματική εργασία, εξετάζεται η λειτουργία του οικιακού μεσολαβητή στο πλαίσιο ενός μικροδικτύου, χωρίς όμως να σημαίνει ότι οι βασικές αρχές και η τελική υλοποίηση του δεν μπορεί να επεκταθεί σχετικά εύκολα και στα υπόλοιπα πλαίσια (ΦοΣΕ, ΔΕΔΔΗΕ κλπ.). Η συμπεριφορά του πράκτορα και τα οφέλη του αξιολογούνται μέσω προσομοιώσεων υπό διαφορετικά σενάρια λειτουργίας με στόχο την κατανόηση της συνεισφοράς της οικιακής ευελιξίας στη συνολική λειτουργία του μικροδικτύου.

Τέλος, δίνεται έμφαση στο ζήτημα της ιδιωτικότητας των οικιακών καταναλωτών. Ο σχεδιασμός του προτεινόμενου πλαισίου οικιακού μεσολαβητή, βασίζεται στη λογική τοπικής λήψης αποφάσεων, ώστε να ελαχιστοποιείται η κοινοποίηση αναλυτικών δεδομένων κατανάλωσης. Προς το ανώτερο επίπεδο ελέγχου αποκρύπτονται πληροφορίες όπως το είδος ή το σύνολο των φορτίων και η πληροφορία που αποστέλλεται προς τον διαχειριστή του μικροδικτύου είναι μόνο η συσσωρευμένη προσφορά.

2 Ευελιξία στην Οικιακή Κατανάλωση

2.1 Εισαγωγή

Στο κεφάλαιο 1, αναδείχθηκε ότι η ενεργειακή μετάβαση, η αυξανόμενη διείσδυση ΑΠΕ και ο ραγδαίος εξηλεκτρισμός των ανθρώπινων δραστηριοτήτων επιβάλλουν την ανάγκη για αυξημένη ευελιξία στη λειτουργία του ΣΗΕ. Για τη διατήρηση της ισορροπίας παραγωγής–ζήτησης απαιτείται η ενεργός συμμετοχή των καταναλώσεων. Στο πλαίσιο αυτό, η οικία θα μπορούσε να αντιμετωπιστεί ως ένας ακόμα ενεργειακός πόρος για το σύστημα, γεγονός που καθιστά απαραίτητη την ανάλυση της ευελιξίας σε επίπεδο οικιακής κατανάλωσης. Το παρόν κεφάλαιο εστιάζει στα χαρακτηριστικά και την ποσοτικοποίηση αυτής της ευελιξίας. Επίσης περιγράφονται οι κατηγορίες οικιακών φορτίων βάσει υπηρεσιών που μπορούν να προσφέρουν. Ακόμα, περιγράφεται ο ρόλος των ΟΣΕΔ και οι πολιτικές δυναμικής τιμολόγησης, ως μηχανισμοί διαχείρισης της ζήτησης. Στόχος είναι η θεμελίωση ενός θεωρητικού υπόβαθρου που αποσαφηνίζει ποιες μορφές ευελιξίας μπορούν να προσφερθούν από μια οικία και υπό ποιες προϋποθέσεις, ώστε στο επόμενο κεφάλαιο να καταστεί δυνατή η ποσοτικοποίηση, η μοντελοποίηση και η αλγοριθμική διαχείρισή τους.

2.2 Χαρακτηριστικά της ευελιξίας

Η ποσοτικοποίηση της ευελιξίας αποτελεί σημαντικό βήμα για την συσσώρευση και ορθή χρήση της από ένα μικροδίκτυο ή μια οικία. Οι διαθέσιμοι πόροι πρέπει να μπορούν να ανταποκριθούν σε σήματα απενεργοποίησης από τον Διαχειριστή Δικτύου Διανομής ή από κάποιον ΦοΣΕ. Η ευελιξία που απαιτείται από τον διαχειριστή ή τον συσσωρευτή περιγράφεται μέσω παραμέτρων οι οποίες καθορίζουν το μέγεθος και την διάρκεια της παρεχόμενης υπηρεσίας [21]. Έχοντας ποσοτικοποιήσει την ευελιξία βάσει των απαιτήσεων που θέτουν οι διαχειριστές, η κατοικία μπορεί να προσφέρει υπηρεσίες ευελιξίας ή να συμμετάσχει απευθείας στο μέλλον σε αγορές ευελιξίας, οι οποίες είναι υπό ανάπτυξη [22].

Μια οικία διαθέτει συγκεκριμένες δυνατότητες μεταβολής φορτίου, οι οποίες μπορούν να αποδοθούν υπό μορφή προσφοράς ευελιξίας. Για να είναι αξιοποιήσιμη αυτή η ευελιξία, πρέπει η προσφορά που θα κατατεθεί από ΦοΣΕ, είτε ως προσφορά ενέργειας στη χονδρεμπορική αγορά είτε ως προσφορά ευελιξίας στην αγορά

εξισορρόπησης, να είναι συμβατή με τις απαιτήσεις του διαχειριστή. Αυτό σημαίνει, να δηλώνει το εύρος της εφικτής μεταβολής του φορτίου και το αντίστοιχο χρονικό διάστημα παροχής υπηρεσίας. Πέραν αυτών των χαρακτηριστικών, λαμβάνονται υπ' όψη και άλλα χαρακτηριστικά της ευελιξίας [23] τα οποία παρουσιάζονται στον Πίνακας 3.

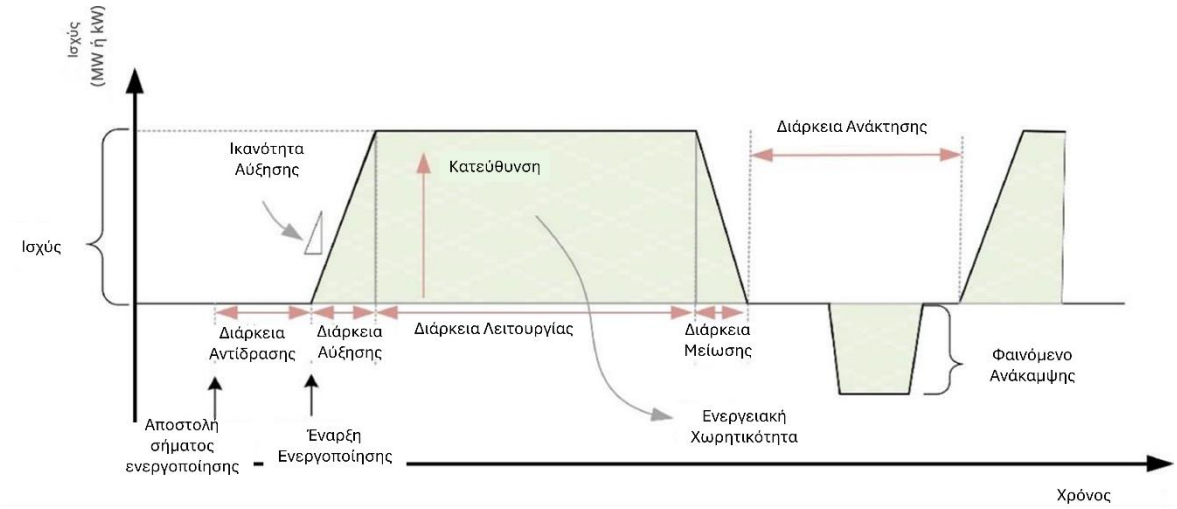
Πίνακας 3 Χαρακτηριστικά Ευελιξίας

Χαρακτηριστικά Ευελιξίας	**Μονάδα Μέτρησης**	**Ορισμός**
Ικανότητα ισχύος	W ή VA	Το μέγιστο μέγεθος μεταβολής ισχύος που μπορεί να προσφέρει ο πόρος (προς τα πάνω και προς τα κάτω).
Κατεύθυνση Ευελιξίας	+/-	Αν ο πόρος μπορεί να προσφέρει αύξηση ή μείωση της ισχύος (π.χ. αύξηση παραγωγής ή μείωση κατανάλωσης, και αντίστροφα).
Χρόνος παροχής ευελιξίας	Mins	Το χρονικό διάστημα στο οποίο ο πόρος μπορεί να συνεχίσει να προσφέρει ευελιξία πριν παραβιαστεί κάποιο τεχνικό του όριο.
Ενεργειακή ικανότητα	Wh	Η συνολική ενέργεια που μπορεί να αποθηκεύσει ή να αποδώσει ο πόρος, δηλαδή το ενεργειακό του περιθώριο.
Ικανότητα ρυθμού μεταβολής (ramp)	W/sec	Πόσο γρήγορα μπορεί να αλλάξει η ισχύς ανά μονάδα χρόνου.
Διάρκεια μεταβολής	sec	Ο απαιτούμενος χρόνος για να φτάσει από την ενεργοποίηση στη μέγιστη ισχύ (ή αντίστοιχα να μειωθεί)
Φαινόμενο ανάκαμψης (rebound effect)	W	Η μεταβολή της ισχύος αμέσως μετά τη λήξη της περιόδου ευελιξίας

Τα χαρακτηριστικά της ευελιξίας περιγράφουν τα τεχνικά όρια και τη δυναμική συμπεριφορά του οικιακού πόρου κατά την παροχή της υπηρεσίας. Η ικανότητα ισχύος και η κατεύθυνση ευελιξίας καθορίζουν το μέγεθος και το πρόσημο της επιτρεπτής μεταβολής της ισχύος, ενώ ο χρόνος παροχής ευελιξίας και η ενεργειακή ικανότητα προσδιορίζουν τη μέγιστη διάρκεια για την οποία μπορεί να διατηρηθεί η απόκλιση από τη βασική λειτουργία. Παράλληλα, η ικανότητα ρυθμού μεταβολής και η διάρκεια μεταβολής εκφράζουν τη δυναμική απόκριση του φορτίου, δηλαδή το πόσο γρήγορα μπορεί να ενεργοποιηθεί ή να απενεργοποιηθεί η παρεχόμενη ευελιξία. Τέλος, το φαινόμενο ανάκαμψης αναφέρεται στη μεταβολή της ισχύος του φορτίου μετά τη λήξη

της ενεργοποίησης της ευελιξίας, καθώς το σύστημα επανέρχεται σταδιακά στην αρχική του κατάσταση. Το φαινόμενο αυτό αποτελεί κρίσιμο στοιχείο για την αποφυγή νέων αιχμών ή παραβίασης τεχνικών περιορισμών.

Η Εικόνα 8 παρουσιάζει με οπτικό τρόπο τα βασικά χαρακτηριστικά της ευελιξίας τα οποία περιεγράφηκαν στον Πίνακας 3. Με την αποστολή του σήματος ενεργοποίησης, ακολουθεί μια σύντομη χρονική καθυστέρηση αντίδρασης μέχρι το φορτίο να ξεκινήσει να μεταβάλλει την ισχύ του. Στο χρονικό διάστημα παροχής υπηρεσίας η ισχύς παραμένει σταθερή βάσει της κατατιθέμενης προσφοράς από τον ΦοΣΕ που έγινε αποδεκτή από το διαχειριστή.



Εικόνα 8 Γραφική απεικόνιση υπηρεσίας ευελιξίας [23].

Μετά το τέλος της ενεργοποίησης, ακολουθεί το φαινόμενο ανάκαμψης (rebound effect), όπου στην προκειμένη η ισχύς μειώνεται και επανέρχεται σταδιακά σε χαμηλότερο επίπεδο του αρχικού. Αυτό συμβαίνει διότι μπορεί να παραβιαστεί κάποιο τεχνικό όριο αν συνεχίσει να λειτουργεί με το αρχικό επίπεδο ισχύος. Αυτό το διάστημα λέγεται χρόνος ανάκτησης και σηματοδοτεί τον χρόνο που χρειάζεται το σύστημα για να επανέλθει στην αρχική του κατάσταση. Τα παραπάνω χαρακτηριστικά δεν αποτελούν εγγενείς ιδιότητες των φορτίων, αλλά προκύπτουν από τη διαδικασία λήψης αποφάσεων ενός οικιακού μεσολαβητή, ο οποίος καθορίζει πότε και σε ποιο βαθμό τα μη κρίσιμα φορτία μπορούν να αποκλίνουν από τη βασική τους λειτουργία, χωρίς να δημιουργείται πρόβλημα στη λειτουργία τους ή τη λειτουργία των εγκαταστάσεων και την άνεση των χρηστών τους.

2.3 Μη κρίσιμα και έξυπνα φορτία

Τα μη κρίσιμα φορτία αναφέρονται σε οικιακές ηλεκτρικές συσκευές, των οποίων η κατανάλωση μπορεί να μεταβληθεί χρονικά ή ποσοτικά, χωρίς να επηρεάζουν την άνεση του χρήστη ή τη λειτουργικότητα τους. Αυτή η δυνατότητα, επιτρέπει στους καταναλωτές να προγραμματίζουν την κατανάλωση τους σε ώρες με χαμηλό κόστος ενέργειας ή σε ώρες με υψηλή διαθεσιμότητα ΑΠΕ, συμβάλλοντας έτσι στην ευσταθέστερη και οικονομικότερη λειτουργία του δικτύου. Η ευελιξία που μπορούν να παρέχουν αυτά τα φορτία χωρίζεται στις εξής κατηγορίες:

1. **Μετατόπιση φορτίου**: Αναβολή της λειτουργίας για τις ώρες εκτός αιχμής
2. **Αποκοπή φορτίου**: Προσωρινή διακοπή της λειτουργίας
3. **Αλλαγή ρυθμού λειτουργίας**: Ρύθμιση της ισχύος λειτουργίας, με ηλεκτρονικά ισχύος για ή/και κυκλώματα αυτοματισμού.

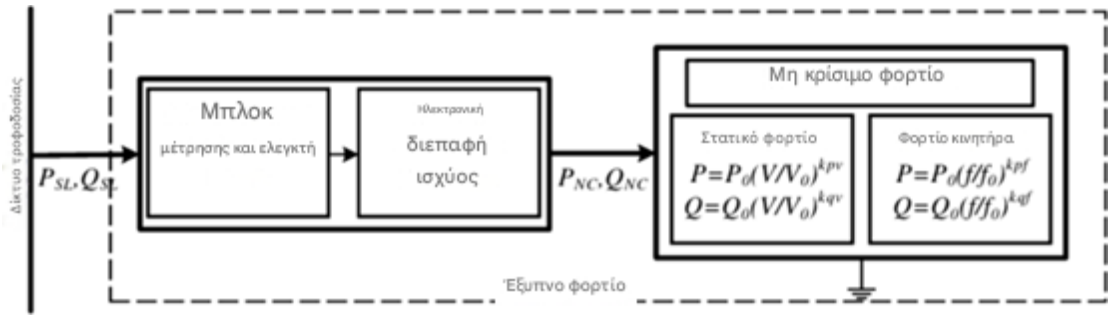
Τα κυριότερα οικιακά φορτία που ανήκουν σε αυτή την κατηγορία είναι συσκευές οι οποίες έχουν δυνατότητα αποθήκευσης ενέργειας. Παραδείγματα αποτελούν οι ηλεκτρικοί θερμοσίφωνες και οι αντλίες θερμότητας που αποθηκεύουν θερμική ενέργεια (είτε σε δεξαμενή νερού, είτε στον αέρα εντός οικίας), οι μπαταρίες ηλεκτρικών οχημάτων που αποθηκεύουν χημική ενέργεια και οι αντλίες νερού όπου αποθηκεύουν ενέργεια έμμεσα με τη χρήση δυναμικής ενέργειας.

Στην βιβλιογραφία, αναφέρονται διάφορες κατηγορίες μη κρίσιμων φορτίων, με κριτήρια τον τύπο της συσκευής, το ενεργειακό προφίλ και το είδος ευελιξίας που μπορεί να προσφέρει. Βέβαια, μια συσκευή μπορεί να ανήκει σε παραπάνω από μια κατηγορίες. Οι κατηγορίες φορτίων που αναφέρονται στην βιβλιογραφία είναι οι εξής:

- **Θερμοστατικά Ελεγχόμενα Φορτία**: Συσκευές οι οποίες διατηρούν την τιμή μιας θερμοκρασίας εντός συγκεκριμένων ορίων π.χ. κλιματιστικά, καταψύξεις, αντλίες θερμότητας και ηλεκτρικοί θερμοσίφωνες. Χάρη στην ιδιότητα της θερμικής αδράνειας, αυτά τα φορτία μπορούν διακόψουν προσωρινά την λειτουργία τους ή να την μετατοπίσουν [24] [25].
- **Αναβαλλόμενα Φορτία:** Η λειτουργία αυτών των φορτίων μπορεί να αναβληθεί εντός ενός αποδεκτού χρονικού ορίου, όπως για παράδειγμα τα πλυντήρια και η φόρτιση ηλεκτρικών οχημάτων. Η απόκριση σε σήματα

δυναμικής τιμολόγησης, είναι βασική λειτουργικότητα που πρέπει να υλοποιούν αυτά τα φορτία, ώστε να μειώνεται το κόστος κατανάλωσης [26].

- **Διακοπτόμενα Φορτία:** Υπάρχει η δυνατότητα διακοπής της λειτουργίας τους πριν ολοκληρωθεί ο κύκλος εργασίας, χωρίς να προκύπτει μείωση της άνεσης [27]. Σε αυτή την κατηγορία ανήκουν όλα τα μη κρίσιμα φορτία, όπου η λειτουργία τους δύναται να διακοπεί βραχυπρόθεσμα, όπως για παράδειγμα οι ηλεκτρικοί θερμοσίφωνες, οι αντλίες νερού και οι αντλίες θερμότητας.



Εικόνα 9 Έξυπνο φορτίο και μπλοκ ελέγχου [5].

Ορισμένα σύγχρονα οικιακά φορτία χαρακτηρίζονται και ως έξυπνα φορτία, καθώς ενσωματώνουν αισθητήρες, ηλεκτρονικά ισχύος και δυνατότητες επικοινωνίας εξ αποστάσεως με το χρήστη, κάποιο σύστημα διαχείρισης της οικίας ή ακόμα και από το διαχειριστή. Όπως απεικονίζεται, ένα έξυπνο φορτίο περιλαμβάνει μονάδες μέτρησης και ελέγχου, οι οποίες επιτρέπουν την ρύθμιση της κατανάλωσης και την απόκριση σε εξωτερικά σήματα ελέγχου (Εικόνα 9). Η είσοδος του συστήματος, είναι η ενεργός και άεργος ισχύς του έξυπνου φορτίου, P_{SL} και Q_{SL} που «βλέπει» το δίκτυο. Το μπλοκ μέτρησης και ελέγχου παρακολουθεί συνεχώς τις μεταβολές της τάσης και της συχνότητας και παράγει σήματα ελέγχου. Τα σήματα αυτά οδηγούνται στη διεπαφή ηλεκτρονικών ισχύος, η οποία απομονώνει το φορτίο από το δίκτυο και επιτρέπει ελεγχόμενη προσαρμογή της κατανάλωσης. Για την παρούσα εργασία, μελετάται το στατικό μη-κρίσιμο φορτίο, όπως περιεγράφηκε σε προηγούμενη ενότητα. Το φορτίο αυτό χαρακτηρίζεται από σχέσεις εξάρτησης της ενεργού και άεργου ισχύος του από την τάση, μέσω των εκθετών k_{pv} και k_{qv}. Οι ισχύεις P_{NC} και Q_{NC} αποτελούν την έξοδο της διεπαφής ηλεκτρονικών ισχύος προς το φορτίο. Το τελικό αποτέλεσμα είναι ότι το έξυπνο φορτίο μπορεί να μεταβάλλει ελεγχόμενα την κατανάλωσή του, χωρίς να διαταράσσεται η λειτουργία του χρήστη, και να συμβάλλει στη στήριξη της τάσης και της συχνότητας του δικτύου [5].

2.4 Οικιακά Συστήματα Ενεργειακής Διαχείρισης

Τα ΟΣΕΔ είναι ενσωματωμένα συστήματα υλικού και λογισμικού που επιτρέπουν την αποδοτική ενεργειακή διαχείριση εντός μιας οικίας [28]. Η δομή των ΟΣΕΔ περιλαμβάνει υλικό όπως αισθητήρες, μετρητές και συστήματα ελέγχου οικιακών συσκευών και η λειτουργία του βασίζεται σε τρείς πυλώνες. Αρχικά, επιτελούν τη συνεχή παρακολούθηση της κατανάλωσης και της παραγωγής ηλεκτρικής ενέργειας μέσω μετρητών και αισθητήρων υψηλής χρονικής ανάλυσης, επιτρέποντας την εξαγωγή αξιόπιστων προφίλ ζήτησης. Δεύτερον, αξιοποιούν τα συλλεγόμενα δεδομένα για τον υπολογισμό βέλτιστων προγραμμάτων λειτουργίας των οικιακών συσκευών και της φόρτισης ηλεκτρικών οχημάτων, λαμβάνοντας υπόψη τις τιμές της ενέργειας και τους περιορισμούς άνεσης των χρηστών. Τέλος, υλοποιούν τον έλεγχο των φορτίων μέσω της αποστολής εντολών στα συστήματα ελέγχου, για την εκτέλεση των αποφάσεων σε πραγματικό χρόνο [28].

Η εξέλιξη των ΟΣΕΔ αποτελεί βασική προϋπόθεση για τον μετασχηματισμό των κατοικιών σε ενεργειακούς πόρους που μπορούν να αξιοποιηθούν από διαχειριστές δικτύων. Στο ίδιο πλαίσιο με τα ΟΣΕΔ, αναπτύσσονται αρχιτεκτονικές και συστήματα «πρακτόρων» για να διαχειριστούν την πολυπλοκότητα της σύγχρονης οικίας. Στα πλαίσια αυτής της εργασίας, το ΟΣΕΔ αξιοποιείται ως το κεντρικό επίπεδο εποπτείας και συντονισμού των φορτίων.

2.5 Πολιτική Τιμολόγησης και Κίνητρα για Παροχή Ευελιξίας

Οι πολιτικές τιμολόγησης της ηλεκτρικής ενέργειας διακρίνονται σε στατικές και δυναμικές. Οι στατικές πολιτικές διατηρούν σταθερή τιμή ανά κιλοβατώρα καθ' όλη τη χρονική περίοδο τιμολόγησης (στο Ελληνικό πλαίσιο, το λεγόμενο μπλε τιμολόγιο). Παρότι παρέχουν σταθερότητα στους καταναλωτές, δεν αντικατοπτρίζουν το πραγματικό ωριαίο κόστος παραγωγής και μεταφοράς ενέργειας το οποίο αυξάνεται σημαντικά στις ώρες αιχμής.

Αντίθετα, οι δυναμικές πολιτικές τιμολόγησης μεταβάλλουν την τιμή της ενέργειας ανάλογα με το χρόνο ή τη ζήτηση κατά τη διάρκεια της ημέρας. Η δυναμική τιμολόγηση έχει εισαχθεί τα τελευταία χρόνια ως μια τεχνική διαχείρισης ζήτησης που προσαρμόζει την τιμή της ηλεκτρικής ενέργειας σε πραγματικό χρόνο, με βάση τις μεταβολές στη ζήτηση και το κόστος παραγωγής. Είναι ένας μηχανισμός οπού οι τιμές

αντικατοπτρίζουν το πραγματικό οριακό κόστος παραγωγής και προσφέρει κίνητρο στους καταναλωτές να μεταβάλλουν την συμπεριφορά τους σύμφωνα με τις ανάγκες του δικτύου. Στην Ελλάδα, εντός του 2026, αναμένεται η εφαρμογή δυναμικών τιμολογίων (το λεγόμενο πορτοκαλί τιμολόγιο). Η εφαρμογή τους προβλέπεται να ξεκινήσει την 1^{η} Φεβρουαρίου για τους μεγάλους καταναλωτές και την 1^{η} Απριλίου για τα νοικοκυριά [29]. Καθοριστικό ρόλο για την υλοποίηση των πορτοκαλί τιμολογίων αποτέλεσε η εγκατάσταση έξυπνων μετρητών. Ο ΔΕΔΔΗΕ έχει εγκαταστήσει περίπου 1,3 εκατομμύρια έξυπνους μετρητές έως τις αρχές του 2026 [29], ενώ ο συνολικός αριθμός αναμένεται να φτάσει τα 3,2 εκατομμύρια έως το τέλος του έτους [19].

2.5.1 Οφέλη δυναμικής τιμολόγησης

Στην βιβλιογραφία αναγνωρίζονται πολλαπλά οφέλη από την εφαρμογή δυναμικής τιμολόγησης, αλλά και ή ανάγκη μηχανισμών για εκμετάλλευση της ευελιξίας στην πλευρά της ζήτησης. Μέσω της μετατόπισης της κατανάλωσης σε χρονικές περιόδους χαμηλότερης τιμής, οι καταναλωτές μπορούν να επιτύχουν μείωση του κόστους ηλεκτρικής ενέργειας χωρίς να απαιτείται απαραίτητα περιορισμός της συνολικής κατανάλωσης ενέργειας. Παράλληλα, σε επίπεδο παραγωγής, περιορίζεται η ανάγκη επενδύσεων σε μονάδες αιχμής ή αποθήκευσης και αποφεύγεται η υπερδιαστασιολόγηση της εγκατεστημένης ισχύος. Στην μεγάλη εικόνα των ΣΗΕ, η δυναμική τιμολόγηση συμβάλλει στην εξομάλυνση της καμπύλης φορτίου και στη μείωση της εξάρτησης από ενεργοβόρες θερμικές μονάδες κατά τις ώρες αιχμής ή έντονης ή/και απρόβλεπτης μεταβολής των ΑΠΕ.

Στο πλαίσιο της παρούσας διπλωματικής, θεωρείται ότι εφαρμόζεται πολιτική δυναμικής τιμολόγησης. Αυτή λαμβάνεται υπόψη από τον πράκτορα διαχείρισης των οικιακών φορτίων, ώστε να ρυθμίζεται η κατανάλωση των μη κρίσιμων φορτίων με δυνατότητες ελέγχου, με αποτέλεσμα να ελαχιστοποιείται το κόστος για το χρήστη.

2.6 Συμπεράσματα

Στο παρόν κεφάλαιο αναλύθηκε η έννοια της ευελιξίας στην οικιακή κατανάλωση και οι βασικοί μηχανισμοί μέσω των οποίων μια σύγχρονη κατοικία μπορεί να μετατραπεί από παθητική κατανάλωση σε ενεργό πόρο για το διαχειριστή του ΣΗΕ. Παρουσιάστηκαν τα βασικά χαρακτηριστικά της ευελιξίας και η ανάγκη ποσοτικοποίησής της ώστε να μπορεί να προσφερθεί ως υπηρεσία προς ανώτερα

επίπεδα ελέγχου. Επίσης, εξετάστηκαν οι κατηγορίες οικιακών φορτίων με βάση τις δυνατότητες που προσφέρουν για παροχή ευελιξίας. Παράλληλα, αναδείχθηκε ο ρόλος των ΟΣΕΔ ως το απαραίτητο τεχνολογικό υπόβαθρο για τον συντονισμό των φορτίων, καθώς και η σημασία της δυναμικής τιμολόγησης ως μηχανισμού έμμεσης διαχείρισης της ζήτησης μέσω οικονομικών σημάτων προς τα ΟΣΕΔ. Από την ανάλυση αυτή προέκυψε ότι τα φορτία με δυνατότητες αποθήκευσης ενέργειας, σε συνδυασμό με λειτουργίες επικοινωνίας και ελέγχου, αποτελούν τους βασικούς πόρους για την παροχή υπηρεσιών ευελιξίας σε οικιακό επίπεδο. Στο επόμενο κεφάλαιο, ποσοτικοποιείται η ευελιξία και μοντελοποιούνται τα υπό μελέτη φορτία.

3 Ποσοτικοποίηση ευελιξίας και μοντελοποίηση φορτίων

3.1 Εισαγωγή

Στο κεφάλαιο 2 αναλύθηκε η έννοια της ευελιξίας στην οικιακή κατανάλωση. Παρουσιάστηκαν τα βασικά χαρακτηριστικά που επιτρέπουν σε μια κατοικία να συμμετέχει ενεργά στη λειτουργία του ΣΗΕ. Ωστόσο, για την πρακτική αξιοποίηση της ευελιξίας απαιτείται η ποσοτικοποίηση της. Παράλληλα, είναι αναγκαία η ανάπτυξη μαθηματικών μοντέλων που περιγράφουν την ηλεκτρική συμπεριφορά των οικιακών φορτίων. Το παρόν κεφάλαιο εστιάζει στη μοντελοποίηση των μη-κρίσιμων οικιακών φορτίων και στην ποσοτικοποίηση της ευελιξίας που αυτά μπορούν να προσφέρουν. Εξετάζονται δύο γενικές κατηγορίες: τα φορτία ελεγχόμενης θερμοκρασίας και τα συστήματα αποθήκευσης ενέργειας. Αρχικά, παρουσιάζονται γενικευμένα μοντέλα φορτίων, τα οποία βασίζονται σε απλοποιημένες μαθηματικές εξισώσεις. Η προσέγγιση αυτή επιτρέπει την ενιαία περιγραφή ενός σημαντικού εύρους ετερογενών καταναλώσεων. Ακολούθως, εξειδικεύονται τα μοντέλα αυτά για την μελέτη συγκεκριμένων οικιακών φορτίων. Εξετάζονται ο ηλεκτρικός θερμοσίφωνας, το σύστημα άντλησης νερού και η μπαταρία. Στην συνέχεια, το κεφάλαιο επικεντρώνεται στην ποσοτικοποίηση της ευελιξίας, που μπορούν να προσφέρουν τα παραπάνω οικιακά φορτία, ώστε να μπορεί να αξιοποιηθεί από το διαχειριστή του συστήματος, μέσω ενιαίων χαρακτηριστικών.

3.2 Γενικευμένα μοντέλα φορτίων

Όπως αναφέρθηκε σε προηγούμενα κεφάλαια, ο οικιακός πράκτορας θα πρέπει να συσσωρεύει τα διαθέσιμα μη κρίσιμα οικιακά φορτία για την παροχή ευελιξίας σε μια ενιαία οντότητα ως «μαύρο κουτί», με στόχο την ανάπτυξη ενός γενικού μοντέλου, το οποίο έχει κοινές εισόδους και εξόδους ανεξαρτήτως του τύπου των φορτίων που διαχειρίζεται. Για τον λόγο αυτό, απαιτείται η χρήση γενικευμένων μαθηματικών μοντέλων, τα οποία περιγράφουν τη δυναμική εξέλιξη διαφορετικών κατηγοριών φορτίων με ενιαίο τρόπο, που είναι όμως κατάλληλα για ανάλυση σε βάθος χρόνου.

Στη συνέχεια παρουσιάζονται δύο βασικές κατηγορίες γενικευμένων εξισώσεων κατάστασης [30]. Αυτές οι κατηγορίες, καλύπτουν το σύνολο των φορτίων που εξετάζονται στην παρούσα εργασία και μπορούν να ενσωματωθούν απευθείας

στον αλγόριθμο βελτιστοποίησης που υλοποιεί ο πράκτορας (δυναμικός προγραμματισμός).

3.2.1 Φορτία Ελεγχόμενης Θερμοκρασίας

Τα φορτία ελεγχόμενης θερμοκρασίας, όπως οι ηλεκτρικοί θερμοσίφωνες, τα συστήματα κλιματισμού και οι αντλίες θερμότητας, χαρακτηρίζονται από την ικανότητα απόδοσης και αποθήκευσης ενέργειας υπό τη μορφή θερμότητας. Η λειτουργία τους δεν απαιτεί συνεχή παροχή ηλεκτρικής ισχύος, καθώς η θερμική αδράνεια του συστήματος επιτρέπει τη μετατόπιση της κατανάλωσης της στο χρόνο χωρίς άμεσο αντίκτυπο στην άνεση του χρήστη [30].

Η δυναμική εξέλιξη της εσωτερικής θερμοκρασίας ενός φορτίου ελεγχόμενης θερμοκρασίας μπορεί να περιγραφεί μέσω ενός ισοδύναμου θερμοδυναμικού μοντέλου πρώτης τάξης. Συγκεκριμένα, η θερμοκρασία του φορτίου στο επόμενο χρονικό βήμα t+1 δίνεται από την εξίσωση κατάστασης (3.1), η οποία εκφράζει τη θερμική ισορροπία μεταξύ της αποθηκευμένης θερμότητας, των απωλειών προς το περιβάλλον και της παρεχόμενης θερμικής ισχύος:

$$T_{in}^{t+1} = T_0^{t+1} + R * Q_{TC}^t - (T_0^{t+1} + R * Q_{TC}^t - T_{in}^t) * e^{(-\Delta t/RC)} \quad (3.1)$$

όπου T_{in}^t είναι η εσωτερική θερμοκρασία του φορτίου στο χρόνο t, T_0^{t+1} η θερμοκρασία περιβάλλοντος στο επόμενο χρονικό βήμα, R η ισοδύναμη θερμική αντίσταση, C η ισοδύναμη θερμική χωρητικότητα και Δt το χρονικό βήμα διακριτοποίησης. Η παρεχόμενη θερμική ισχύς Q_{TC}^t ισούται με την ηλεκτρική ισχύ λειτουργίας του φορτίου πολλαπλασιασμένη με έναν συντελεστή απόδοσης λ, όπως φαίνεται στην εξίσωση:

$$Q_{TC}^t = \lambda * P_{TC}^t \quad (3.2)$$

Αντικαθιστώντας την (3.2) στην (3.1), η εξίσωση κατάστασης μπορεί να γραφεί στη μορφή (3.3), η οποία είναι γραμμική ως προς την κατάσταση και τη μεταβλητή ελέγχου:

$$T_{in}^{t+1} = e^{-\Delta t/RC} * T_{in}^t + \lambda * R * P_{TC}^t * \left(1 - e^{-\Delta t/RC}\right) + T_0^{t+1} * \left(1 - e^{-\Delta t/RC}\right) \quad (3.3)$$

Με τον τρόπο αυτό, η εσωτερική θερμοκρασία του θερμοσίφωνα ή του κλιματιστικού αντιμετωπίζεται ως μεταβλητή κατάστασης, ενώ η ηλεκτρική ισχύς

λειτουργίας P_{TC}^{t} αποτελεί τη μεταβλητή ελέγχου που επιλέγει ο πράκτορας. Το μοντέλο αυτό χρησιμοποιείται στην παρούσα εργασία για την περιγραφή της δυναμικής του ηλεκτρικού θερμοσίφωνα.

3.2.2 Συστήματα Αποθήκευσης Ενέργειας

Τα συστήματα αποθήκευσης ενέργειας, όπως οι μπαταρίες, διαθέτουν τη δυνατότητα τόσο φόρτισης όσο και εκφόρτισης, παρέχοντας αυξημένο βαθμό ελέγχου στον πράκτορα. Σε μικρά χρονικά διαστήματα, η μεταβολή της κατάστασης φόρτισης μπορεί να περιγραφεί μέσω της εξίσωσης:

$$S_{ES}^{t+1} = S_{ES}^{t} + (\alpha * P_{ES}^{t}/E_{ES}) * \Delta t \tag{3.4}$$

όπου S_{ES}^{t} είναι η κατάσταση φόρτισης της μπαταρίας και E_{ES} η ονομαστική χωρητικότητα. Ο συντελεστής α λαμβάνει διαφορετικές τιμές ανάλογα με το αν η μπαταρία φορτίζεται ή εκφορτίζεται, όπως φαίνεται στην εξίσωση:

$$\alpha = \begin{cases} \eta_c, & P_{ES}^{t} > 0 \\ 1/\eta_d, & P_{ES}^{t} \leq 0 \end{cases} \tag{3.5}$$

όπου η_c και η_d είναι οι αποδόσεις φόρτισης και εκφόρτισης αντίστοιχα. Το μοντέλο αυτό χρησιμοποιείται στην παρούσα εργασία για την περιγραφή της μπαταρίας, αλλά και για το σύστημα άντλησης νερού. Στην περίπτωση της μπαταρίας, η κατάσταση φόρτισης αντιστοιχεί στο ποσοστό αποθηκευμένης ηλεκτρικής ενέργειας, ενώ στην περίπτωση του συστήματος άντλησης νερού αντιστοιχεί στο επίπεδο αποθηκευμένου νερού σε μια δεξαμενή. Η γραμμική μορφή της εξίσωσης επιτρέπει τη διακριτοποίηση των καταστάσεων, καθιστώντας δυνατή την εφαρμογή αλγοριθμικών μεθόδων βελτιστοποίησης.

3.3 Ειδικά μοντέλα φορτίων

Στην ενότητα αυτή, εξειδικεύονται τα γενικευμένα μοντέλα που παρουσιάστηκαν στην προηγούμενη ενότητα και χρησιμοποιούνται για την εξομοίωση των επιμέρους φορτίων που παρουσιάζονται στο επόμενο κεφάλαιο της εργασίας, ως εργαλεία ευελιξίας. Συγκεκριμένα, τα γενικευμένα μοντέλα θα εφαρμοστούν στον ηλεκτρικό θερμοσίφωνα, στο σύστημα άντλησης νερού και στην οικιακή μπαταρία.

3.3.1 Θερμοσίφωνας

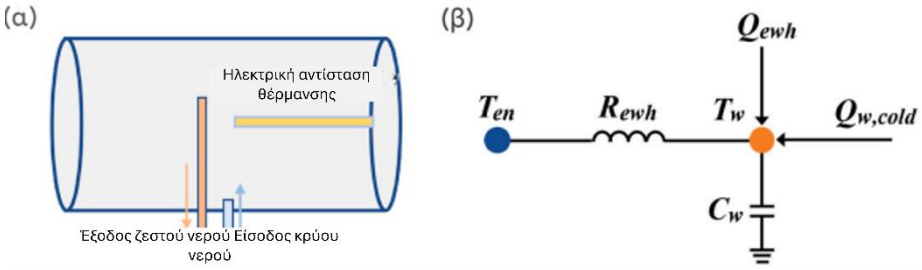
Ο ηλεκτρικός θερμοσίφωνας αποτελεί ένα από τα πιο συνήθη φορτία μιας οικίας, ενώ είναι το 2^{ο} σε σειρά πιο ενεργοβόρο φορτίο στις Ευρωπαϊκές κατοικίες (ενότητα 1.1). Πρόκειται για μια θερμοστατικά ελεγχόμενη συσκευή με υψηλή κατανάλωση ισχύος, η οποία λειτουργεί με κύκλους ενεργοποίησης και απενεργοποίησης της ηλεκτρικής της αντίστασης, με στόχο τη διατήρηση της θερμοκρασίας του νερού μιας δεξαμενής/δοχείου εντός προκαθορισμένων ορίων άνεσης για το χρήστη.

Η βασική ιδιαιτερότητα του θερμοσίφωνα, η οποία τον καθιστά ιδιαίτερα ελκυστικό για εφαρμογές διαχείρισης ζήτησης, είναι η ύπαρξη θερμικής αποθήκευσης με σημαντική χωρητικότητα και ελάχιστες απώλειες. Η αποθηκευμένη θερμική ενέργεια στη δεξαμενή επιτρέπει την προσωρινή διακοπή λειτουργίας της συσκευής χωρίς άμεση υποβάθμιση της παρεχόμενης υπηρεσίας προς τον χρήστη, δηλαδή παρουσιάζει χαμηλή ευαισθησία σε μικρής διάρκειας διακοπές τροφοδοσίας. Συνεπώς, ο θερμοσίφωνας μπορεί να θεωρηθεί ως ένα φορτίο με εγγενή χρονική ευελιξία [31].

Οι θερμοσίφωνες που είναι εξοπλισμένοι με ικανότητες ανοδικής και καθοδικής ρύθμισης της κατανάλωσης ισχύος, είναι ιδανικά εργαλεία για μείωση αιχμών και αποκοπή φορτίου. Η μετατόπιση της κατανάλωσης ισχύος μπορεί να πραγματοποιηθεί είτε μέσω προθέρμανσης της δεξαμενής πριν από περιόδους αιχμής, είτε μέσω προσωρινής απενεργοποίησης της αντίστασης κατά τη διάρκεια αυτών. Η μέγιστη χρονική διάρκεια κατά την οποία ένας θερμοσίφωνας μπορεί να μείνει απενεργοποιημένος χωρίς να μειωθεί η θερμοκρασία κάτω από το επιτρεπτό για το χρήστη όριο εξαρτάται από την ποσότητα της αποθηκευμένης θερμικής ενέργειας, τις θερμικές απώλειες προς το περιβάλλον και το προφίλ χρήσης ζεστού νερού [31].

Σύμφωνα με την κατηγοριοποίηση της ενότητας 3.2.1, ο θερμοσίφωνας ανήκει στα Φορτία Ελεγχόμενης Θερμοκρασίας. Η μοντελοποίηση του θερμοσίφωνα μπορεί να πραγματοποιηθεί με μεγάλη ακρίβεια μέσω ενός απλοποιημένου θερμικού RC μοντέλου πρώτης τάξης όπως φαίνεται στην Εικόνα 10. Συγκεκριμένα στην εικόνα (10α) φαίνεται η απεικόνιση του θερμοσίφωνα και στην εικόνα (10β) φαίνεται η σχηματική αναπαράσταση του θερμοδυναμικού RC μοντέλου του θερμοσίφωνα, όπου T_{en} είναι η θερμοκρασία του περιβάλλοντος, T_w η θερμοκρασία του ζεστού νερού, Q_{ewh} η παρεχόμενη θερμότητα από την ηλεκτρική αντίσταση, $Q_{w,cold}$ οι θερμικές απώλειες

λόγω χρήσης ζεστού νερού, R_{ewh} η ισοδύναμη θερμική αντίσταση και C_w η ισοδύναμη θερμική χωρητικότητα της δεξαμενής.



Εικόνα 10 Ηλεκτρικός θερμοσίφωνας: (α) Απεικόνιση βασικών μερών του (β) Σχηματική αναπαράσταση του θερμοδυναμικού RC μοντέλου του [31].

Η θερμική κατάσταση της δεξαμενής περιγράφεται από την ισορροπία μεταξύ εισερχόμενης θερμότητας από την ηλεκτρική αντίσταση, απωλειών προς το περιβάλλον και θερμικής ανακίνησης λόγω εξόδου θερμού και την ταυτόχρονη είσοδο ψυχρού νερού κατά την χρήση, όπως περιγράφεται από την εξίσωση:

$$C_w * \frac{dT_w}{dt} = \frac{T_{en}(t) - T_w(t)}{R_{ewh}} + Q_{ewh}(t) - Q_{w,cold}(t) \quad (3.6)$$

$$Q_{ewh}(t) = P_{boiler,in}(t) * \eta_{boiler} \quad (3.7)$$

όπου η_{boiler} ο βαθμός απόδοσης της αντίστασης. Η ισοδύναμη θερμική χωρητικότητα της δεξαμενής υπολογίζεται ως:

$$C_w = C_{water} * \rho * V_{tank} \quad (3.8)$$

όπου με C_{water} συμβολίζεται η ειδική θερμότητα του νερού, με ρ η πυκνότητα νερού και με V_{tank} ο όγκος του νερού που χωράει ο θερμοσίφωνας. Η θερμική ενέργεια που αφαιρείται από τη δεξαμενή κατά τη χρήση ζεστού νερού οφείλεται στην αντικατάσταση μέρους του θερμού νερού της δεξαμενής από το κρύο νερό του δικτύου ύδρευσης. Η ανάμιξη θερμού και ψυχρού νερού, στην οικία πραγματοποιείται στον μίκτη της βρύσης. Μια βαλβίδα επιτρέπει ταυτόχρονη ροή από τις δύο παροχές (ζεστό και κρύο) και ένας χειροκίνητος μηχανισμός (π.χ. κινούμενο έμβολο) ρυθμίζει την αναλογία μεταξύ ζεστού και ψυχρού νερού έτσι ώστε η θερμοκρασία εξόδου να ρυθμίζεται χειροκίνητα από το χρήστη στην επιθυμητή θερμοκρασία $T_{desired}$. Για τα σενάρια προσομοίωσης που εξετάζονται σε επόμενο κεφάλαιο, θεωρείται ότι $T_{desired}$ ισούται με 40 °C [32].

Ο συνολικός όγκος νερού που καταναλώνεται $V_{consumed}$, ισούται με το άθροισμα του ζεστού V_{hot} και του κρύου νερού V_{cold} που θα οδηγήσει σε κάθε χρονική στιγμή στην $T_{desired}$. Από το ισοζύγιο θερμικής ενέργειας ισχύει η εξίσωση (3.9). Στην εξίσωση αυτή θεωρήθηκε ότι T_w είναι η θερμοκρασία του νερού στη δεξαμενή και T_{cold} η θερμοκρασία του ψυχρού νερού τροφοδοσίας από το δίκτυο ύδρευσης.

$$V_{hot} * T_w + V_{cold} * T_{cold} = (V_{hot} + V_{cold}) * T_{desired} \quad (3.9)$$

Λύνοντας ως προς τον όγκο ζεστού νερού V_{hot} που αντλείται από τη δεξαμενή προκύπτει η εξίσωση:

$$V_{hot} = V_{consumed} * \frac{T_{desired} - T_{cold}}{T_w - T_{cold}} \quad (3.10)$$

Ο όγκος αυτός είναι ίσος με τον όγκο ψυχρού νερού που εισέρχεται στη δεξαμενή και προκαλεί την αντίστοιχη απώλεια θερμικής ενέργειας. Η θερμική ενέργεια που αφαιρείται από τη δεξαμενή στο χρονικό διάστημα Δτ δίνεται από την εξίσωση:

$$E_{w,cold} = C_{water} * \rho * V_{hot} * (T_w - T_{cold}) \quad (3.11)$$

Η θερμική ισχύς δίνεται από την εξίσωση:

$$Q_{w,cold}(t) = \frac{C_w * \rho * V_{consumed}(t) * (T_{desired} - T_{cold})}{\Delta\tau} \quad (3.12)$$

Κατά την θέρμανση του νερού της δεξαμενής εμφανίζεται το φαινόμενο της θερμικής διαστρωμάτωσης. Το φαινόμενο αυτό, παρατηρείται με τη δημιουργία κατακόρυφου προφίλ θερμοκρασίας εντός της δεξαμενής, όπου το θερμότερο νερό τείνει να συγκεντρώνεται στα ανώτερα στρώματα, ενώ το ψυχρότερο παραμένει στα κατώτερα [33]. Το φαινόμενο είναι εντονότερο σε κατακόρυφες δεξαμενές και σε περιπτώσεις χαμηλής ανάμιξης, καθώς επηρεάζεται από τη γεωμετρία του δοχείου και τη θέση της αντίστασης. Για οριζόντιους κυλινδρικούς θερμοσίφωνες, η υπόθεση ενός μοναδικού θερμικού κόμβου (αντίσταση θερμοσίφωνα) δεν οδηγεί σε μεγάλη απώλεια ακρίβειας (τάξης μεγέθους 2%), καθώς η απόδοση της ηλεκτρικής αντίστασης επηρεάζεται ελάχιστα από φαινόμενα θερμικής διαστρωμάτωσης εντός της δεξαμενής [31].

Για διακριτό χρονικό βήμα Δt, η εξίσωση (3.7) μετασχηματίζεται στη διακριτοποιημένη εξίσωση κατάστασης (3.13) όπως δηλαδή περιεγράφηκε και στην προηγούμενη ενότητα η γενικευμένη μορφή της εξίσωσης των θερμοστατικά ελεγχόμενων φορτίων.

$$T_{t+1} = a * T_t + (1 - a) * \left[T_{en}(t) + \left(Q_{ewh}(t) - Q_{w,cold}(t)\right) * R_{ewh}\right] \quad (3.13)$$

$$a = \exp\left(-\frac{\Delta t}{R_{ewh} * C_w}\right) \quad (3.14)$$

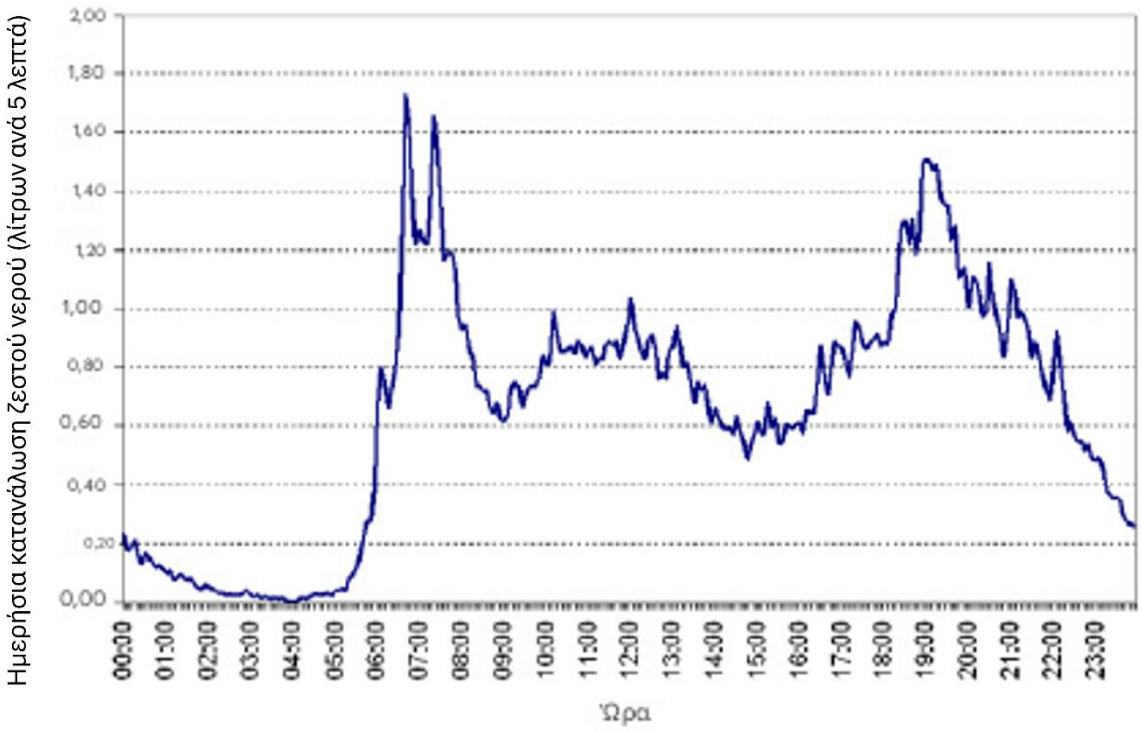
Η θερμοκρασία του νερού περιορίζεται από τα όρια του θερμοστάτη, όπου το κάτω όριο T_{min} ορίζεται από το σημείο ρύθμισης ενώ το άνω όριο T_{max} ορίζεται από τεχνικές προδιαγραφές ασφαλούς λειτουργίας τον θερμοσίφωνα:

$$T_{min} \leq T_w \leq T_{max} \quad (3.15)$$

Αντίστοιχα, περιορίζεται η δυνατότητα ρύθμισης της ισχύος του θερμοσίφωνα, με άνω όριο την ονομαστική του ισχύ $P_{boiler,in}^{nominal}$:

$$0 \leq P_{boiler,in} \leq P_{boiler,in}^{nominal} \quad (3.16)$$

Ο ρυθμός κατανάλωσης του συνολικού όγκου νερού $V_{consumed}(t)$, εκφράζεται σε λίτρα ανά μονάδα χρόνου (L/min) και αντιστοιχεί στο ημερήσιο προφίλ κατανάλωσης νερού της κατοικίας (Εικόνα 11). Το προφίλ αυτό θεωρείται δεδομένο στην προκειμένη περίπτωση. Υποθέτουμε ότι ο πράκτορας διαχείρισης του ηλεκτρικού θερμοσίφωνα είναι ικανός να εκτιμήσει με άριστη ακρίβεια το προφίλ κατανάλωσης από ιστορικά δεδομένα καθημερινής χρήσης, αξιοποιώντας π.χ. μεθόδους μηχανικής μάθησης ή άλλες μεθόδους πρόβλεψης χρονοσειρών.



Εικόνα 11 Προφίλ κατανάλωσης νερού [34].

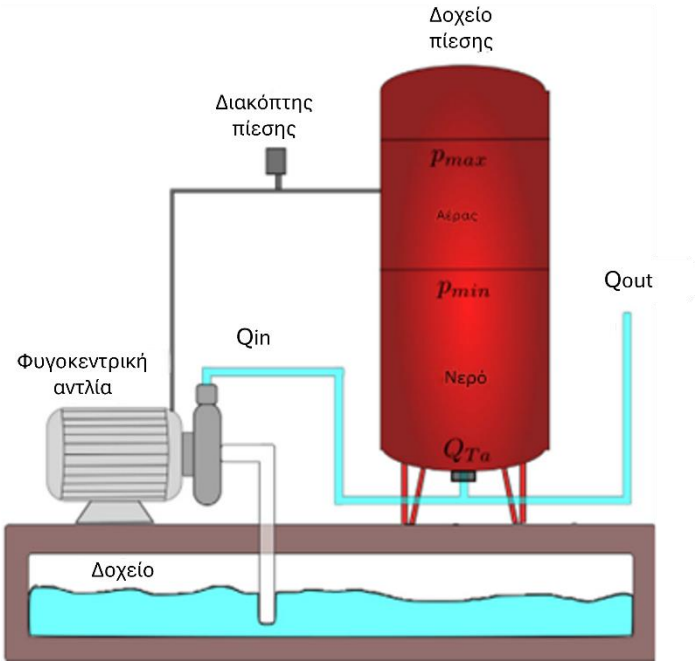
Για να ενσωματωθεί το προφίλ κατανάλωσης $V_{consumed}(t)$ στην διακριτοποιημένη εξίσωση κατάστασης του θερμοσίφωνα (3.13), απαιτείται η μετατροπή της παροχής νερού σε καταναλισκόμενο όγκο εντός κάθε διακριτού χρονικού διαστήματος Δτ. Συγκεκριμένα σε κάθε χρονικό διάστημα Δτ, ο συνολικός όγκος νερού που καταναλώνεται $V_{consumed}$ σε λίτρα, υπολογίζεται από το άθροισμα των επιμέρους όγκων ανά Δτ:

$$V_{consumed} = \int_{t}^{t+\Delta\tau} V_{consumed}(\tau)\, d\tau \qquad (3.17)$$

Ο υπολογισμός αυτός πραγματοποιείται πριν την εκτέλεση του βρόγχου βελτιστοποίησης, ώστε να μην επηρεάζεται ο χρόνος εκτέλεσης του αλγόριθμου.

3.3.2 Σύστημα άντλησης νερού

Συστήματα άντλησης νερού χρησιμοποιούνται για την παροχή νερού με επαρκή πίεση σε κτίρια με αυξημένο ύψος ή σε οικιακά αρδευτικά συστήματα που διαθέτουν πηγάδι. Ένα τυπικό σύστημα άντλησης νερού φαίνεται στην Εικόνα 12. Περιλαμβάνει φυγοκεντρική αντλία και δεξαμενή πίεσης. Η δεξαμενή λειτουργεί ως ενδιάμεση αποθήκη μηχανικής ενέργειας, καθώς αποθηκεύει νερό σε μεγαλύτερο ύψος, ενώ στην έξοδο της η πίεση του νερού ρυθμίζεται από τους καταναλωτές [35].



Εικόνα 12 Σύστημα άντλησης νερού [35].

Η ευελιξία που προσφέρει το σύστημα άντλησης νερού οφείλεται στο γεγονός ότι η παροχή νερού με επαρκή πίεση προς τους χρήστες δεν απαιτεί τη συνεχή λειτουργία της αντλίας σε κάθε χρονική στιγμή. Η ύπαρξη της δεξαμενής επιτρέπει τον χρονικό διαχωρισμό μεταξύ της κατανάλωσης ηλεκτρικής ισχύος από την αντλία και της πραγματικής ζήτησης νερού, καθώς το νερό μπορεί να αντληθεί και να αποθηκευτεί εκ των προτέρων, ώστε να μπορεί να καταναλωθεί αργότερα. Με τον τρόπο αυτό, η λειτουργία της αντλίας μπορεί να μετατοπιστεί σε χρονικές περιόδους χαμηλότερου φορτίου ή κόστους, ενώ είναι δυνατή και η προσωρινή μείωση ή διακοπή της κατανάλωσης ηλεκτρικής ισχύος, χωρίς άμεση επίπτωση στην παρεχόμενη υπηρεσία, εφόσον η δεξαμενή έχει επαρκή όγκο [35]. Επιπλέον, ευελιξία μπορεί να επιτευχθεί μέσω προσαρμογής της κατανάλωσης ισχύος της αντλίας, με αντίστοιχη προσαρμογή του ρυθμού άντλησης.

Στη λογική της ενότητας 3.2.2 (Συστήματα αποθήκευσης ενέργειας), το σύστημα άντλησης με δεξαμενή μπορεί να μοντελοποιηθεί ως αποθήκη, όπου το επίπεδο φόρτισης, ή αλλιώς η «κατάσταση» του φορτίου (state of charge) εκφράζει τον αποθηκευμένο όγκο νερού. Το φυσικό μοντέλο περιγράφεται από τις ακόλουθες εξισώσεις. Μετατροπή της ηλεκτρικής ισχύος της αντλίας σε υδραυλική P_{hyd} περιγράφεται από την εξίσωση (3.18). Η υδραυλική ισχύς P_{hyd} σχετίζεται με την παροχή νερού Q στους καταναλωτές σε σταθερό ύψος h, μέσω της εξίσωσης (3.19). Οι

σταθερές g και ρ περιγράφουν την επιτάχυνση της βαρύτητας και την ειδική πυκνότητα του νερού αντίστοιχα.

$$P_{hyd} = \eta_{pump} * P_t^{pump} \quad (3.18)$$

$$\text{και } P_{hyd} = \rho * g * h * Q_t \quad (3.19)$$

Η μεταβολή του όγκου του νερού $V_{t+1} - V_t$ δεξαμενή σε χρονικό βήμα $\Delta\tau$ δίνεται από ισοζύγιο όγκου νερού, δηλαδή την εισερχόμενη ροή από την λειτουργία της αντλίας μείον την εξερχόμενη λόγω ζήτησης από την εξίσωση (3.20).

$$V_{t+1} - V_t = Q_t * \Delta\tau - V_t^{out,Pump} \quad (3.20)$$

όπου $V_t^{out,Pump}$ είναι ο όγκος νερού που αφαιρείται από τη δεξαμενή στη χρονική περίοδο Δt και Q_t η παροχή της αντλίας. Αντικαθιστώντας την εξίσωση (3.19) στην εξίσωση (3.20), προκύπτει η εξίσωση (3.21), η οποία είναι συνεπής με το γενικευμένο μοντέλο αποθήκευσης που παρουσιάστηκε στην ενότητα 3.2.2.

$$V_{t+1} = V_t + \eta_{pump} \frac{P_t^{pump} \Delta\tau}{\rho g h} - V_t^{out,Pump} \quad (3.21)$$

Όπως και ο θερμοσίφωνας, έτσι και η αντλία, χαρακτηρίζεται από τα λειτουργικά της όρια που περιγράφονται στην εξίσωση (3.22), δηλαδή τον ελάχιστο και μέγιστο επιτρεπτό όγκο νερού και την μέγιστη και ελάχιστη δυνατότητα ρύθμισης της κατανάλωσης ισχύος, άρα και ρυθμού άντλησης.

$$V_{min} \leq V_t \leq V_{max} \quad \text{και} \quad P_{min}^{pump} \leq P_t^{pump} \leq P_{max}^{pump} \quad (3.22)$$

Αντίστοιχα με την περίπτωση του θερμοσίφωνα, το προφίλ κατανάλωσης νερού του συστήματος άντλησης $V_t^{out,Pump}(t)$ θεωρείται γνωστό και εκφράζεται ως χρονική συνάρτηση παροχής νερού προς τον χρήστη. Για να ενσωματωθεί το προφίλ αυτό στη διακριτοποιημένη εξίσωση κατάστασης του συστήματος άντλησης (3.23), ακολουθείται η ίδια διαδικασία με την περίπτωση του θερμοσίφωνα και εφαρμογή της εξίσωσης (3.17).

3.3.3 Μπαταρία

Οι μπαταρίες, ως συστήματα αποθήκευσης ηλεκτρικής ενέργειας, βρίσκουν πλέον μεγάλη απήχηση στις οικίες απορροφώντας ενέργεια σε χρονικές περιόδους

χαμηλού κόστους (φόρτιση), ή υψηλής τοπικής παραγωγής (π.χ. Φ/Β στην στέγη), και την απόδοσή της (εκφόρτιση) σε περιόδους αυξημένης τιμής [36]. Το κόστος πλέον έχει μειωθεί σε προσιτά επίπεδα για το μέσο νοικοκυριό. Ενδεικτικά, στην ευρωπαϊκή αγορά, οι μέσες τιμές πώλησης μειώθηκαν από περίπου 1.332 €/kWh το πρώτο εξάμηνο του 2023 σε περίπου 711 €/kWh το δεύτερο εξάμηνο του 2025, παρουσιάζοντας δηλαδή μείωση 46% σε διάστημα μόλις δύο ετών [37].

Η ευελιξία που προσφέρει μια μπαταρία οφείλεται στην δυνατότητά της να αποσυνδέει χρονικά την κατανάλωση από την παραγωγή ηλεκτρικής ενέργειας. Όπως εξηγήθηκε την ενότητα 2.2, η χρονική μετατόπιση του φορτίου είναι από τις βασικές υπηρεσίες ευελιξίας. Η μοντελοποίηση της μπαταρίας μπορεί να γίνει σύμφωνα με την περιγραφή της ενότητας 3.2.2.

$$\mathrm{SoC}_{t+1} = \mathrm{SoC}_t + \eta_{bat} * \frac{\mathrm{P}_t^{bat} * \Delta\tau}{\mathrm{E}_{cap}} \tag{3.23}$$

$$\mathrm{SoC}_{min} \leq \mathrm{SoC}_t \leq \mathrm{SoC}_{max} \quad \text{και} \quad \mathrm{P}_{min}^{bat} \leq \mathrm{P}_t^{bat} \leq \mathrm{P}_{max}^{bat} \tag{3.24}$$

όπου E_{cap} είναι η ονομαστική της χωρητικότητα ενέργειας (σε kWh), απόδοση διαχείρισης της ενέργειας η_{bat} ισούται με η_c στην περίπτωση φόρτισης και με $1/\eta_d$ στην περίπτωση εκφόρτισης. Αντίστοιχα με τα άλλα φορτία που παρέχουν ευελιξία, περιγράφεται από τους λειτουργικούς της περιορισμούς, δηλαδή το μέγιστο επίπεδο φόρτισης SoC_{max} και το ελάχιστο επίπεδο φόρτισης SoC_{min}, όπως επίσης και την μέγιστη P_{max}^{bat} και ελάχιστη P_{min}^{bat} δυνατή κατανάλωση και παροχή ισχύος, αντίστοιχα. Η P_{min}^{bat} έχει αρνητική τιμή, αφού περιορίζει το ρυθμό εκφόρτισης και η P_{max}^{bat} θετική αφού περιορίζει το ρυθμό φόρτισης.

3.4 Ποσοτικοποίηση ευελιξίας

Στον Πίνακας 4, καταγράφονται τα βασικά χαρακτηριστικά που χρειάζεται να είναι γνωστά ή να μπορούν να εκτιμηθούν, ώστε κάθε οικιακό φορτίο που μελετάται στην παρούσα εργασία να μπορεί να παρέχει υπηρεσίες ευελιξίας. Έγινε μια προσπάθεια δημιουργίας γενικών κατηγοριών όπου ποσοτικοποιούν την διαθέσιμη ευελιξία βάση των χαρακτηριστικών ευελιξίας που περιεγράφηκαν στον Πίνακας 3. Οι εξισώσεις που παρουσιάζονται στον Πίνακας 4, εμπεριέχονται στις εξισώσεις των φορτίων που αναλύθηκαν ήδη. Επίσης, οι εξισώσεις αυτές, καταδεικνύουν τον τρόπο με τον οποίο τα τεχνικά και λειτουργικά χαρακτηριστικά κάθε μη κρίσιμου φορτίου

αξιοποιούνται στο μοντέλο του πράκτορα. Μέσω αυτών των εξισώσεων, προκύπτει ότι τα φορτία που αναλύθηκαν, μπορούν να ικανοποιήσουν τις απαιτήσεις που παρουσιάστηκαν στον Πίνακας 3 και να παρέχουν υπηρεσίες ευελιξίας. Τα χαρακτηριστικά που σημειώνονται με «✓» αποτελούν δεδομένα εισόδου για τον πράκτορα φορτίου, τα οποία εξάγονται από τα τεχνικά χαρακτηριστικά της κάθε συσκευής ή εισάγονται από τον χρήστη (π.χ. το άνω και κάτω όριο άνεσης). Βάσει μοντελοποίησης με χαρακτηριστικά λειτουργίας «μαύρου κουτιού» ως προς την προστασία των προσωπικών δεδομένων του χρήστη, η 1η στήλη του Πίνακας 4, αφορά τις εξόδους του που είναι ικανός να διαθέσει ο πράκτορας.

Πίνακας 4 Δυνατότητες και περιγραφή ευελιξίας ανά φορτίο που μελετήθηκε

Χαρακτηριστικά	Φορτία		
	Θερμοσίφωνας	Μπαταρία	Αντλία νερού
Τάση (V) Ρεύμα (A) Συχνότητα (Hz) Ισχύς (kW)	✓	✓	✓
Συντελεστής Ισχύος (cosφ)	✓	✓	✓
Ταχύτητα (rpm)	-	-	✓
Όρια άνεσης	✓	✓	✓
Χωρητικότητα Ενέργειας (kWh)	$E = C_w * (T_{max} - T_{min})$	✓	$E_{max}^{kWh} = \frac{\rho * g * h * V_{max}}{3.6 \times 10^6}$
Απώλειες αδράνειας (kW)	$P_{loss}(t) = \frac{T_w(t) - T_{en}}{R_{ewh}}$	Τεχνικά χαρακτηριστικά «self-discharge»	-
Χρόνος φόρτισης στην μέγιστη χωρητικότητα (minutes)	$t = -\tau\ ln\left(\frac{T_{max} - T_{ss}}{T_0 - T_{ss}}\right)$ $\tau = R_{ewh} * C_w$ $T_{ss} = T_{en} + Q_{nom} * R_{ewh}$	$t = \frac{(SoC_{max} - So\ C_0) * E}{\eta_{bat} P_{ch}}$	$t = \frac{(V_{max} - V_0)\,\rho g h}{\eta_{pump} P}$
Ρυθμός φόρτισης (kW)	$R_t = \frac{\Delta E}{\Delta t}$ $\Delta E = C_w * (T_{t+1} - T_t)$	$R_t = \eta_{bat} * P_t$	$R_t = \frac{\Delta E_t}{\Delta \tau}$ $= \eta_{pump} \backslash astP$
Επίπεδο φόρτισης (%)	$SoC(t) = 100\frac{T_w(t) - T_{min}}{T_{max} - T_{min}}$	SoC(t)	$SoC(t)$ $= 100\frac{V_t - V_{min}}{V_{max} - V_{min}}$

Κάθε φορτίο έχει διαφορετικούς περιορισμούς και χαρακτηριστικά αλλά η ποσοτικοποίηση της ευελιξίας πρέπει να γίνεται με ενιαίο τρόπο. Στο προτεινόμενο πλαίσιο, κάθε φορτίο ελέγχεται από έναν ανεξάρτητο πράκτορα φορτίου, ο οποίος

λειτουργεί ως μαύρο κουτί ως προς τον πράκτορα οικίας και αποκρύπτει τα εσωτερικά χαρακτηριστικά του. Ο Πίνακας 4 ορίζει το κοινό σύνολο μεγεθών και παραμέτρων μέσω των οποίων οι πράκτορες φορτίων εκφράζουν τη δυναμική και τα όρια της ευελιξίας τους, ανεξάρτητα από τον τύπο του φορτίου. Οι παράμετροι αυτές περιλαμβάνουν ηλεκτρικά χαρακτηριστικά, όρια άνεσης ή λειτουργίας, ισοδύναμη ενεργειακή χωρητικότητα, απώλειες, ρυθμούς φόρτισης και επίπεδα φόρτισης. Με τον τρόπο αυτό, διαφορετικά φορτία, όπως ο θερμοσίφωνας, η μπαταρία και η αντλία νερού, μπορούν να περιγραφούν με έναν ενιαίο λειτουργικό τρόπο. Ουσιαστικά, ο Πίνακας 4 λειτουργεί ως διεπαφή επικοινωνίας μεταξύ του πράκτορα φορτίου και του πράκτορα οικίας, δηλαδή ως ένα τυποποιημένο σύνολο μεγεθών και παραμέτρων μέσω των οποίων κάθε πράκτορας φορτίου εκφράζει τις δυνατότητες και τους περιορισμούς του. Επιτρέπει στον πράκτορα οικίας να συνθέτει τη συνολική διαθέσιμη οικιακή ευελιξία χωρίς να απαιτείται γνώση των εσωτερικών δυναμικών των επιμέρους φορτίων. Αντίστοιχα, ο πράκτορας οικίας εκθέτει προς τα ανώτερα επίπεδα ελέγχου (διαχειριστής δίκτυου/ ΦοΣΕ) μόνο συγκεντρωτικά μεγέθη ευελιξίας, λειτουργώντας με τη σειρά του ως μαύρο κουτί στο επίπεδο του συστήματος.

3.5 Συμπεράσματα

Στο κεφάλαιο αυτό παρουσιάστηκε ένα πλαίσιο για την μοντελοποίηση μη-κρίσιμων οικιακών φορτίων και την ποσοτικοποίησης της ευελιξίας που μπορούν να παρέχουν. Αρχικά, παρουσιάστηκαν γενικευμένα μοντέλα φορτίων, τα οποία επιτρέπουν την κατηγοριοποίηση διαφορετικών καταναλώσεων με βάση κοινά χαρακτηριστικά. Έμφαση δόθηκε σε φορτία με θερμική αδράνεια και σε συστήματα αποθήκευσης ενέργειας, καθώς αυτά εμφανίζουν σημαντικές δυνατότητες παροχής ευελιξίας. Η μοντελοποίηση των επιμέρους φορτίων είναι απαραίτητη για την εφαρμογή υπολογιστικών μεθόδων ελέγχου και βελτιστοποίησης της χρήσης τους. Η προσέγγιση αυτή συμβάλλει στη βελτίωση της οικονομικής αποδοτικότητας τόσο για τους καταναλωτές όσο και για το ΣΗΕ. Επιπλέον, επιτρέπει την ποσοτικοποίηση των δυνατοτήτων των φορτίων σε επίπεδο ευελιξίας. Στο επόμενο κεφάλαιο, το αναπτυγμένο πλαίσιο μοντελοποίησης αξιοποιείται για τον σχεδιασμό ενός πράκτορα διαχείρισης των μη-κρίσιμων φορτίων μιας οικίας.

4 Πράκτορες για τη διαχείριση οικιακών φορτίων

4.1 Εισαγωγή

Στο προηγούμενο κεφάλαιο αναπτύχθηκε το θεωρητικό και μαθηματικό υπόβαθρο για την ποσοτικοποίηση της ευελιξίας και τη μοντελοποίηση των μη-κρίσιμων οικιακών φορτίων. Η ανάλυση αυτή καθιστά δυνατή την περιγραφή της συμπεριφοράς των φορτίων και των περιορισμών λειτουργίας τους ως στοιχεία ευελιξίας. Ωστόσο, για την αξιοποίηση της ευελιξίας, απαιτείται ένας μηχανισμός λήψης αποφάσεων που να συντονίζει τα επιμέρους φορτία. Το παρόν κεφάλαιο εστιάζει στη σχεδίαση και λειτουργία πρακτόρων, ως μηχανισμοί λήψης αποφάσεων για τη διαχείριση οικιακών φορτίων. Αρχικά παρουσιάζονται οι βασικές κατηγορίες πρακτόρων και οι εφαρμογές τους στην οικιακή ενεργειακή διαχείριση. Στη συνέχεια περιγράφεται μια ιεραρχική αρχιτεκτονική πολλών πρακτόρων, όπου διαχωρίζεται ο τοπικός έλεγχος των φορτίων από τον συντονισμό τους σε επίπεδο οικίας. Ιδιαίτερη έμφαση δίνεται σε ζητήματα ασφάλειας και ιδιωτικότητας, μέσω της υιοθέτησης της προσέγγισης «μαύρου κουτιού». Ακολούθως, αναπτύσσεται ένας αλγόριθμος δυναμικού προγραμματισμού για τον βέλτιστο προγραμματισμό των φορτίων. Στόχος του κεφαλαίου είναι η υλοποίηση πρακτόρων φορτίου και ο συντονισμός τους από τον πράκτορα οικίας, ο οποίος είναι σε θέση να λαμβάνει αυτόνομες αποφάσεις και να παρέχει ποσοτικοποιημένες υπηρεσίες ευελιξίας προς το διαχειριστή δικτύου.

4.2 Κατηγορίες πρακτόρων και εφαρμογές στην οικιακή ενεργειακή διαχείριση

Στα πλαίσια της σύγχρονης οικίας, οι πράκτορες μπορούν να αποτελέσουν τη νοημοσύνη των ΟΣΕΔ ή/και να υλοποιούνται ως αυτόνομες οντότητες λογισμικού εγκατεστημένες στα συστήματα ελέγχου των επιμέρους συσκευών. Στην βιβλιογραφία οι «πράκτορες» ορίζονται ως εξής: «*μια φυσική ή εικονική οντότητα, η οποία μπορεί να δρά, να αντιλαμβάνεται το περιβάλλον της και να επικοινωνεί με άλλους, είναι αυτόνομη και διαθέτει δεξιότητες για να πετύχει τους στόχους της*» [38]. Υπάρχουν διάφορες κατηγορίες πρακτόρων για διαφορετικές περιπτώσεις χρήσης αλλά και διαφορετική πολυπλοκότητα υλοποίησης, οι οποίες παρουσιάζονται ακολούθως:

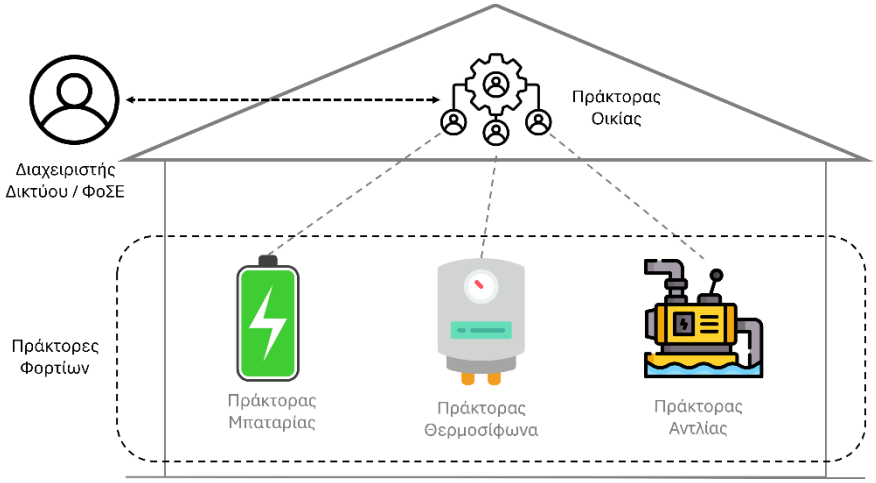
1. **Πράκτορες Απλής Αντίδρασης**: Λειτουργούν αποκλειστικά βάση κανόνων συνθήκης–δράσης, δηλαδή δεν διατηρούν ιστορικό, απλά αντιδράνε βάση της τωρινής κατάστασης. Για παράδειγμα, ένας τέτοιος πράκτορας θα μπορούσε να αντιπροσωπεύει έναν θερμοστάτη που ενεργοποιείται ή απενεργοποιείται όταν η θερμοκρασία υπερβεί τα προκαθορισμένα όρια.
2. **Πράκτορες Αντίδρασης με Μοντέλα:** Διατηρούν ένα μοντέλο περιβάλλοντος, ώστε να εκτελούν πιο τεκμηριωμένες αποφάσεις, λαμβάνοντας υπ' όψη την επίδραση των πράξεων τους στο περιβάλλον. Στο πλαίσιο ενός ΟΣΕΔ, ένας τέτοιος πράκτορας μπορεί να ενσωματώνει μοντέλα φορτίων για να λαμβάνει πιο ρεαλιστικές αποφάσεις.
3. **Πράκτορες με Στόχους:** Οι πράκτορες αυτοί διαθέτουν καθορισμένους στόχους και λαμβάνουν απόφαση με βάση το αν μια ενέργεια τους συμβάλλει στην επίτευξη του στόχου τους. Σε ΟΣΕΔ, θα μπορούσε να τεθεί στόχος ελαχιστοποίησης κόστους ή αύξησης αυτοκατανάλωσης.
4. **Πράκτορες Ωφελιμότητας:** Οι πράκτορες ωφελιμότητας επεκτείνουν τους «πράκτορες με στόχους» αξιολογώντας τις δράσεις τους βάση του κατά πόσο μεγιστοποιούν την συνάρτηση ωφελιμότητας. Αυτή η προσέγγιση επιτρέπει να λαμβάνουν αποφάσεις όταν υπάρχουν αντικρουόμενοι στόχοι, για παράδειγμα στα πλαίσια ΟΣΕΔ, όταν λαμβάνονται ταυτόχρονα ως κριτήρια το κόστος κατανάλωσης και η άνεση των χρηστών.
5. **Πράκτορες Μάθησης:** Διαθέτουν τη δυνατότητα προσαρμογής της συμπεριφοράς τους μέσω εμπειρίας. Βελτιώνουν τις αποφάσεις τους παρατηρώντας τις συνέπειες των ενεργειών τους, προσαρμόζοντας τα εσωτερικά τους μοντέλα, ώστε να πετυχαίνουν καλύτερα αποτελέσματα στο μέλλον.

Στην παρούσα διπλωματική εργασία αναπτύσσεται ένα πολυπρακτορικό σύστημα, όπως θα αναλυθεί διεξοδικά στις επόμενες ενότητες, στο οποίο οι επιμέρους πράκτορες φορτίων υλοποιούνται ως πράκτορες ωφελιμότητας. Κάθε πράκτορας φορτίου λαμβάνει αποφάσεις με βάση την ελαχιστοποίηση μιας συνάρτησης κόστους, η οποία ενσωματώνει οικονομικά κριτήρια, όπως το κόστος ενέργειας, αλλά και λειτουργικούς περιορισμούς που απορρέουν από τα φυσικά χαρακτηριστικά του αντίστοιχου φορτίου. Με τον τρόπο αυτό, οι πράκτορες είναι σε θέση να διαχειρίζονται

αποτελεσματικά αντικρουόμενους στόχους, όπως η ελαχιστοποίηση του κόστους κατανάλωσης και η διατήρηση της άνεσης ή της ασφαλής λειτουργίας.

4.3 Αρχιτεκτονική πολλών πρακτόρων

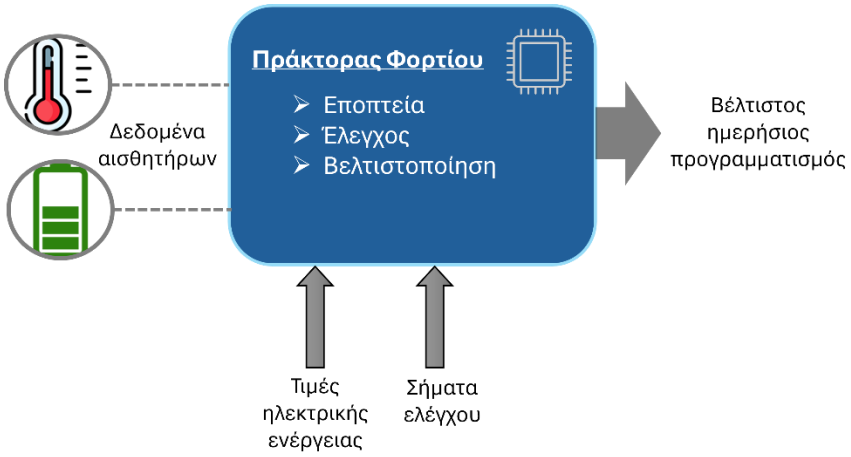
Η προτεινόμενη αρχιτεκτονική βασίζεται σε ιεραρχική προσέγγιση συντονισμού για την διαχείριση των μη-κρίσιμων φορτίων. Η ιεραρχία αυτή διαχωρίζει τον αποκεντρωμένο έλεγχο των επιμέρους μη κρίσιμων φορτίων από τον κεντρικό συντονισμό σε επίπεδο οικίας [39]. Όπως απεικονίζεται στην Εικόνα 13, η αρχιτεκτονική οργανώνεται σε δύο βασικά επίπεδα συντονισμού: το επίπεδο του πράκτορα φορτίου και το επίπεδο του πράκτορα οικίας.



Εικόνα 13 Αρχιτεκτονική πολλών πρακτόρων.

Στο πρώτο επίπεδο, κάθε οικιακό φορτίο συνοδεύεται από έναν πράκτορα φορτίου, ο οποίος είναι υπεύθυνος για τον τοπικό έλεγχο και τη λειτουργία της συσκευής. Ο πράκτορας φορτίου παρακολουθεί και καταγράφει την κατάσταση του φορτίου, συλλέγοντας δεδομένα από αισθητήρες που περιγράφουν την τρέχουσα κατανάλωση ισχύος και κρίσιμες μεταβλητές λειτουργίας, όπως η θερμοκρασία ή το επίπεδο φόρτισης. Με βάση τις πληροφορίες αυτές, μπορεί να εκτελεί τοπικές αποφάσεις ελέγχου και να υπολογίζει τη δυνατότητα μεταβολής της κατανάλωσης, χωρίς να παραβιάζονται οι λειτουργικοί περιορισμοί ή η άνεση του χρήστη. Η γενική λειτουργία ενός πράκτορα φορτίου απεικονίζεται στην Εικόνα 14. Παράλληλα, λαμβάνει εξωτερικές πληροφορίες, όπως τις τιμές ηλεκτρικής ενέργειας και εντολές από το ΟΣΕΔ, το οποίο με την σειρά του δέχεται σήματα από τον διαχειριστή δικτύου ή τον ΦοΣΕ. Με βάση τα δεδομένα αυτά, ο πράκτορας φορτίου εκτιμά την εξέλιξη της κατάστασης του φορτίου μέσω του φυσικού του μοντέλου και αξιολογεί τις διαθέσιμες

δράσεις. Ενδεικτικές δράσεις περιλαμβάνουν την ενεργοποίηση ή απενεργοποίηση της συσκευής, τη ρύθμιση της ισχύος λειτουργίας ή την προσωρινή αναβολή της λειτουργίας.



Εικόνα 14 Πράκτορας φορτίου.

Στο δεύτερο επίπεδο, ο πράκτορας της οικίας λειτουργεί ως κεντρικός συντονιστής και είναι ενσωματωμένος στο ΟΣΕΔ. Ο πράκτορας της οικίας συλλέγει τις πληροφορίες που αποστέλλονται από τους επιμέρους πράκτορες φορτίου. Στη συνέχεια, επεξεργάζεται τις πληροφορίες αυτές, σχηματίζοντας μια συνολική εικόνα της ευελιξίας που μπορεί να προσφέρει η οικία. Όπως φαίνεται στην Εικόνα 13, η πληροφορία αυτή μεταφέρεται από τον πράκτορα της οικίας προς ανώτερα επίπεδα ελέγχου, όπως έναν ΦοΣΕ ή απευθείας προς τον ΔΕΔΔΗΕ. Με αυτή την προσέγγιση, η ευελιξία των μεμονωμένων φορτίων παρουσιάζεται ως συνολική και ποσοτικοποιημένη προσφορά, η οποία μπορεί να αξιοποιηθεί σε σχήματα διαχείρισης ζήτησης ή παροχής επικουρικών υπηρεσιών από διαχειριστές ΣΗΕ.

Με την ιεραρχική δομή πετυχαίνουμε τον διαχωρισμό ρόλων μεταξύ τοπικού και κεντρικού ελέγχου. Οι πράκτορες φορτίου επικεντρώνονται στην διαχείριση των επιμέρους συσκευών, ενώ ο πράκτορας της οικίας αναλαμβάνει τον συντονισμό των οικιακών πρακτόρων και την επικοινωνία με το εξωτερικό σύστημα. Παράλληλα, περιορίζεται η ανάγκη αποστολής αναλυτικών δεδομένων για τις συνήθειες των χρηστών ή τα είδη και το πλήθος των φορτίων που διαθέτουν, ενισχύοντας την προστασία της ιδιωτικότητας των χρηστών.

4.4 Ζητήματα Ασφάλειας και Ιδιωτικότητας

Η ευρεία τοποθέτηση έξυπνων μετρητών, εισάγει σημαντικές προκλήσεις ως προς την προστασία της ιδιωτικότητας των καταναλωτών. Η υψηλή χρονική ανάλυση των μετρήσεων επιτρέπει την εξαγωγή λεπτομερών προφίλ κατανάλωσης, από τα οποία θα μπορούσαν να εξαχθούν ευαίσθητες πληροφορίες για τους καταναλωτές. Ενδεικτικά, από τα δεδομένα κατανάλωσης μπορούν να προκύψουν συμπεράσματα για τις καθημερινές δραστηριότητες, καθώς και την παρουσία ή την απουσία των ενοίκων από την κατοικία.

Η ανάλυση των δεδομένων του έξυπνου μετρητή με τεχνικές «*μη παρεμβατικής αναγνώρισης φορτίων*» (non-intrusive load monitoring) επιτρέπει την αναγνώριση των συμπεροφορικών μοτίβων των καταναλωτών ακόμα και χωρίς άμεση πρόσβαση στις συσκευές [40]. Επιπλέον, οι μετρητικές και επικοινωνιακές υποδομές είναι ευάλωτες σε διαρροές ή κακόβουλες επιθέσεις. Σε συνδυασμό με το ανεπαρκές υπάρχον κανονιστικό πλαίσιο, το οποίο δεν παρέχει επαρκή προστασία ως προς την προστασία των δεδομένων που προκύπτουν από την ενεργειακή κατανάλωση, εγείρονται ανησυχίες για την προστασία της ιδιωτικότητας του καταναλωτή. Δεδομένου ότι ο οικιακός πράκτορας χειρίζεται αναλυτικά δεδομένα κατανάλωσης, τα οποία θα μπορούσαν να αποκαλύψουν ευαίσθητες πληροφορίες αν κοινοποιούνταν π.χ. απευθείας στο ΦοΣΕ, ο σχεδιασμός του βασίζεται στην ελαχιστοποίηση της διακινούμενης πληροφορίας. Ο πράκτορας υλοποιείται ως οντότητα τύπου «μαύρου κουτιού», ώστε προς τα ανώτερα επίπεδα ελέγχου να αποστέλλεται μόνο απολύτως απαραίτητη πληροφορία, χωρίς αποκάλυψη επιμέρους δεδομένων συσκευών ή καταναλωτικών προτύπων.

4.5 Μοντελοποίηση με προσέγγιση μαύρου κουτιού

Ο όρος «μαύρο κουτί» περιγράφει ένα σύστημα του οποίου η εσωτερική λειτουργία δεν είναι ορατή ή δεν χρειάζεται να αποκαλυφθεί. Το σύστημα δέχεται δεδομένα εισόδου, εκτελεί μια εσωτερική επεξεργασία και παράγει αποτελέσματα, χωρίς ο εξωτερικός παρατηρητής να έχει πρόσβαση στη διαδικασία λήψης αποφάσεων [41]. Στην παρούσα εργασία, η προσέγγιση μαύρου κουτιού εφαρμόζεται στο επίπεδο της οικίας προς τα ανώτερα επίπεδα ελέγχου, αλλά μερικώς και στο επίπεδο των πρακτόρων φορτίων. Ο πράκτορας οικίας, που είναι ενσωματωμένος στο ΟΣΕΔ,

επιλύει τοπικά το πρόβλημα βελτιστοποίησης και εκθέτει προς το ανώτερο επίπεδο ελέγχου (ΔΕΔΔΗΕ, ΦοΣΕ κλπ) μόνο ένα ενιαίο προφίλ ευελιξίας F_k (4.1), χωρίς να αποκαλύπτει πληροφορίες για τα επιμέρους φορτία. Το προφίλ αυτό περιγράφει, για κάθε χρονικό διάστημα Δt, τη μέγιστη δυνατή αύξηση και μείωση της ενεργού ισχύος της οικίας.

$$F_k = \{\Delta P_k^+,\ \Delta P_k^-\}, \quad \forall k \in \{1, \dots, \mathrm{N}\} \tag{4.1}$$

όπου k είναι ο δείκτης του διακριτού χρονικού βήματος, N είναι ο συνολικός αριθμός χρονικών διαστημάτων του ορίζοντα βελτιστοποίησης, ΔP_k^+ είναι η μέγιστη επιτρεπτή αύξηση της ενεργού ισχύος της οικίας κατά το χρονικό διάστημα k σε σχέση με το προγραμματισμένο προφίλ λειτουργίας της, ενώ ΔP_k^- είναι η μέγιστη επιτρεπτή μείωση της ενεργού ισχύος της οικίας.

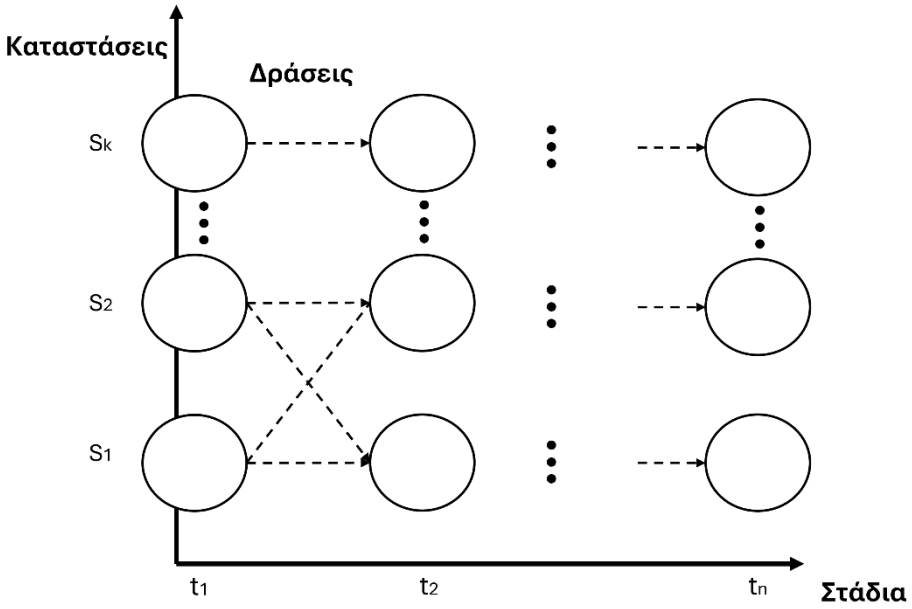
Σε εσωτερικό επίπεδο, κάθε πράκτορας φορτίου διαθέτει πλήρη πρόσβαση στα δεδομένα κατάστασης του αντίστοιχου φορτίου. Ο πράκτορας φορτίου είναι υπεύθυνος για την εκτέλεση ενεργειών ελέγχου, όπως η ρύθμιση της ισχύος λειτουργίας ή η ενεργοποίηση και απενεργοποίηση του φορτίου, σύμφωνα με τις εντολές που λαμβάνει από τον πράκτορα οικίας, όπως τα προτεινόμενα σημεία λειτουργίας. Προς τον πράκτορα οικίας εκθέτει μόνο το κατά πόσο ένα αίτημα είναι εφικτό να ικανοποιηθεί, χωρίς να αποκαλύπτονται οι λεπτομέρειες εσωτερικής λειτουργίας του φορτίου. Η μοντελοποίηση των πρακτόρων φορτίου με προσέγγιση μαύρου κουτιού επιτρέπει την παροχή υπηρεσιών ευελιξίας με ενιαίο τρόπο, ανεξάρτητα από το είδος της συσκευής την οποία επιτηρούν και ελέγχουν. Με τον τρόπο αυτό, αποσαφηνίζονται οι διεπαφές εισόδων και εξόδων, χωρίς να απαιτείται γνώση του φυσικού φορτίου που ενεργοποιείται. Στην προκειμένη περίπτωση, ο πράκτορας οικίας, και κατ' επέκταση ο ΦοΣΕ ή ο διαχειριστής δικτύου, αντιλαμβάνονται την οικία ως έναν ενιαίο πόρο ευελιξίας και όχι ως σύνολο επιμέρους συσκευών με ανεξάρτητα χρονοπρογράμματα και δυνατότητες. Με αυτή την μοντελοποίηση επιτυγχάνεται προστασία της ιδιωτικότητας, αλλά και απλοποίηση της διαλειτουργικότητας, καθώς η πολυπλοκότητα της διαχείρισης του κάθε φορτίου δεν απασχολεί τα ανώτερα επίπεδα ελέγχου.

4.6 Χρονοπρογραμματισμός εντολών πρακτόρων φορτίου

Κάθε πράκτορας φορτίου οφείλει να είναι ικανός να υπολογίζει τον βέλτιστο χρονοπρογραμματισμό λειτουργίας του φορτίου που ελέγχει, λαμβάνοντας υπόψη τους τεχνικούς περιορισμούς του συστήματος και οικονομικά κριτήρια λειτουργίας. Για τον σκοπό αυτό, στην παρούσα ενότητα αναπτύσσεται ένα γενικευμένο πλαίσιο δυναμικού προγραμματισμού, το οποίο επιτρέπει την επίλυση ενός διακριτού προβλήματος βελτιστοποίησης σε βάθος χρόνου. Το πλαίσιο αυτό εφαρμόζεται τοπικά από κάθε πράκτορα φορτίου, ανεξάρτητα από τους υπόλοιπους πράκτορες. Η επίλυση του προβλήματος οδηγεί στην κατασκευή μιας βέλτιστης τροχιάς κατάστασης και τη συνεχή ανανέωσή του με δεδομένα πραγματικού χρόνου, καθώς και του αντίστοιχου προγραμματισμένου προφίλ κατανάλωσης ισχύος για τον εξεταζόμενο χρονικό ορίζοντα. Τα παραγόμενα αποτελέσματα εκτίθενται στον πράκτορα οικίας, χωρίς να απαιτείται γνώση από μέρους του των επιμέρους φυσικών μοντέλων. Στις επόμενες υποενότητες παρουσιάζεται το γενικευμένο πλαίσιο του δυναμικού προγραμματισμού.

4.6.1 Ορισμός δυναμικού προγραμματισμού

Ο δυναμικός προγραμματισμός (ΔΠ) αποτελεί μια μέθοδο βελτιστοποίησης προβλημάτων με δομή πολλών σταδίων, στα οποία η λήψη απόφασης εξελίσσεται σε διακριτά χρονικά βήματα. Σε κάθε στάδιο, το σύστημα βρίσκεται σε μια συγκεκριμένη κατάσταση και η επιλογή μιας δράσης επηρεάζει την εξέλιξη της κατάστασης στα επόμενα στάδια [42]. Η δομή του ΔΠ περιλαμβάνει τα «*στάδια*» της διαδικασίας, τις «*καταστάσεις*» του συστήματος και τις «*δράσεις*» που επιτρέπεται να ληφθούν σε κάθε στάδιο, όπως φαίνεται στην Εικόνα 15.



Εικόνα 15 Σχηματική αναπαράσταση προβλήματος δυναμικού προγραμματισμού.

Η βασική ιδέα είναι ότι ένα πολύπλοκο πρόβλημα μπορεί να αναλυθεί σε μια ακολουθία απλούστερων υποπροβλημάτων. Η επίλυση κάθε υποπροβλήματος οδηγεί στην εύρεση της βέλτιστης λύσης του υποπροβλήματος, η οποία αποτελεί τμήμα της συνολικής λύσης και χρησιμοποιείται για την εύρεση της συνολικής βέλτιστης λύσης. Σε κάθε χρονικό στάδιο, η μόνη πληροφορία που χρειάζεται να είναι γνωστοποιημένη είναι η κατάσταση του συστήματος ώστε να μπορούν να υπολογιστούν οι δυνατές δράσεις. Βασικό χαρακτηριστικό του δυναμικού προγραμματισμού αποτελεί η αρχή της «*βέλτιστης πολιτικής*». Σύμφωνα με αυτήν, μια βέλτιστη ακολουθία αποφάσεων έχει την ιδιότητα ότι, ανεξάρτητα από τις προηγούμενες ενέργειες, οι μελλοντικές αποφάσεις πρέπει να είναι βέλτιστες σε σχέση με την κατάσταση που προκύπτει από την τρέχουσα δράση.

4.6.2 Διατύπωση του προβλήματος

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, ο δυναμικός προγραμματισμός αξιοποιείται από τους αυτόνομους πράκτορες φορτίου για τον υπολογισμό του βέλτιστου προγράμματος λειτουργίας του κάθε μη-κρίσιμου οικιακού φορτίου που επιτηρούν και ελέγχουν. Κάθε πράκτορας φορτίου επιλύει τοπικά το δικό του πρόβλημα βελτιστοποίησης, λαμβάνοντας υπόψη τους λειτουργικούς περιορισμούς του αντίστοιχου φορτίου, καθώς και τα ενδεχόμενα όρια άνεσης που μπορεί να προκύπτουν στην περίπτωση θερμοστατικά ελεγχόμενου φορτίου. Ο πράκτορας οικίας λειτουργεί ως κεντρικός συντονιστής των φορτίων, συλλέγοντας από

κάθε πράκτορα φορτίου τα ελάχιστα και μέγιστα επιτρεπτά όρια ισχύος σε κάθε χρονικό διάστημα και τα εκθέτει ως ένα συσσωρευμένο προφίλ προς τον διαχειριστή δικτύου. Το πρόβλημα είναι πεπερασμένου χρονικού ορίζοντα, δηλαδή πολλαπλών, αλλά μετρήσιμων σταδίων. Η κατάσταση του «συστήματος» σε κάθε χρονικό στάδιο περιγράφεται από το διάνυσμα κατάστασης S_t το οποίο ισούται με T_t για την θερμοκρασία του θερμοσίφωνα, με V_t τον όγκο αποθηκευμένου νερού της δεξαμενής και με SoC_t για το επίπεδο φόρτισης της μπαταρίας. Οι μεταβολές στις καταστάσεις των φορτίων περιγράφονται από τις εξισώσεις κατάστασης που παρουσιάστηκαν στην ενότητα 3.3 (Ειδικά μοντέλα φορτίων). Για κάθε δυνατή κατάσταση του φορτίου, υπολογίζεται το ελάχιστο δυνατό συνολικό κόστος, έως εκείνο το στάδιο, καθώς και το κόστος της αντίστοιχης δράσης, ώστε να επιτευχθεί το ελάχιστο κόστος [43].

Η υλοποίηση ενός αλγόριθμου δυναμικού προγραμματισμού περιλαμβάνει τρία βήματα: την διακριτοποίηση του χώρου καταστάσεων, τον ορισμό του χώρου δράσεων και τον υπολογισμό της συνάρτησης αξίας, δηλαδή το ελάχιστο κόστος από ένα δεδομένο χρονικό στάδιο έως το τέλος του χρονικού ορίζοντα. Ο χρόνος t θεωρείται διακριτή μεταβλητή, με βήμα μετάβασης Δτ, το οποίο αντιστοιχεί στο χρονικό διάστημα κατά το οποίο τα φορτία θεωρείται ότι λειτουργούν με σταθερή ισχύ για να μεταβεί το σύστημα από την κατάσταση s_t στην s_{t+1}. Ο χρονικός ορίζοντας του προβλήματος ορίζεται ως $t \in T$, όπου $T = \{0, \dots, N-1\}$ με N να είναι το πλήθος των χρονικών διαστημάτων.

Στην συνέχεια, γίνεται διακριτοποίηση του χώρου καταστάσεων, δηλαδή της θερμοκρασίας του θερμοσίφωνα, του αποθηκευμένου όγκου νερού της δεξαμενής που γεμίζει η αντλία και της αποθηκευμένης ενέργειας της μπαταρίας. Προσδιορίζονται διακριτά επίπεδα κατάστασης που μπορεί να βρεθεί το σύστημα, έτσι ώστε να περιοριστεί ο χώρος καταστάσεων S, σε πεπερασμένο για να συγκλίνει ο αλγόριθμος. Στον Πίνακα 5 παρουσιάζεται η διακριτοποίηση των καταστάσεων του συστήματος που χρησιμοποιείται στο πλαίσιο του προτεινόμενου αλγορίθμου δυναμικού προγραμματισμού. Συγκεκριμένα, ορίζονται τα πλέγματα καταστάσεων για τη θερμοκρασία του θερμοσίφωνα (T_{grid}), τον αποθηκευμένο όγκο νερού της δεξαμενής (V_{grid}) και το επίπεδο φόρτισης της μπαταρίας (SoC_{grid}). Για κάθε μεταβλητή κατάστασης καθορίζονται τα κατώτερα και ανώτερα όρια λειτουργίας, καθώς και το αντίστοιχο βήμα διακριτοποίησης r. Η επιλογή των βημάτων διακριτοποίησης επηρεάζει άμεσα την ακρίβεια της αναπαράστασης των φυσικών καταστάσεων και το

υπολογιστικό κόστος του αλγορίθμου. Όσο μικρότερο είναι το βήμα διακριτοποίησης τόσο μεγαλύτερος είναι ο αριθμός των καταστάσεων που πρέπει να εξεταστούν.

Πίνακας 5 Διακριτοποίηση καταστάσεων συστήματος

$T_i \in T_{grid}$	$T_{grid} = (T_{min}, T_{max})$	$\Delta T = \frac{T_{max} - T_{min}}{r}$
$V_i \in V_{grid}$	$V_{grid} = (V_{min}, V_{max})$	$\Delta V = \frac{V_{max} - V_{min}}{r}$
$SoC_i \in SoC_{grid}$	$SoC_{grid} = (SoC_{min}, SoC_{max})$	$\Delta SoC = \frac{SoC_{max} - SoC_{min}}{r}$

Έπειτα ορίζεται ο χώρος δράσεων A και αντιστοιχεί στο σύνολο των επιτρεπτών διακριτών τιμών ισχύος του φορτίου στο χρονικό διάστημα Δτ, σύμφωνα με τους τεχνικούς περιορισμούς λειτουργίας του. Όσες περισσότερες οι δυνατές δράσεις τόσο μεγαλώνει ο χώρος δράσεων, άρα αυξάνεται και ο χρόνος εκτέλεσης του αλγόριθμου. Για κάθε φορτίο επομένως, πρέπει να επιλέγεται μικρός αριθμός δράσεων, ώστε να μειωθεί το υπολογιστικό κόστος που δίνει επαρκή ακρίβεια. Κάθε δράση a για κάθε φορτίο i στο χρονικό στάδιο t γράφεται ως $a_t = P_t^i \in A$.

4.6.3 Εξισώσεις μετάβασης κατάστασης

Για κάθε χρονικό στάδιο, εξετάζεται το σύνολο των επιτρεπτών δράσεων a_t και υπολογίζεται η μετάβαση του συστήματος στην επόμενη κατάσταση. Οι μεταβάσεις αυτές υπολογίζονται μέσω των εξισώσεων κατάστασης f_T, f_V, f_S οι οποίες προσεγγίζουν την φυσική λειτουργία των φορτίων. Οι εξισώσεις κατάστασης για κάθε φορτίο, υλοποιήθηκαν στις προηγούμενες ενότητες και εδώ χρησιμοποιούνται για τον υπολογισμό της δυναμικής εξέλιξης του συστήματος:

$$T_{t+1} = f_T\left(T_t, P_t^{boiler}, V_t^{out,boiler}, \Delta\tau\right) \quad (4.2)$$

$$V_{t+1} = f_V\left(V_t, P_t^{pump}, V_t^{out,Pump}, \Delta\tau\right) \quad (4.3)$$

$$SoC_{t+1} = f_S\left(SoC_t, P_t^{battery}, \Delta\tau\right) \quad (4.4)$$

Οι μεταβλητές $V_t^{out,boiler}$ και $V_t^{out,pump}$ εκφράζουν τα αντίστοιχα προφίλ κατανάλωσης νερού, οι οποίες θεωρούνται γνωστές ή προβλέψιμες είσοδοι του συστήματος. Οι εξισώσεις κατάστασης γραμμένες σε αυτή την μορφή είναι πολύ εύκολο και γρήγορο να υπολογιστούν συγκριτικά με την προσομοίωση λεπτομερών φυσικών μοντέλων. Αυτό είναι σημαντικό για την εφαρμογή του δυναμικού προγραμματισμού, όπου απαιτείται η αξιολόγηση μεγάλου αριθμού πιθανών καταστάσεων και δράσεων, καθώς και για εφαρμογές σε πραγματικό ή σχεδόν πραγματικό χρόνο, όπως είναι επί το πλείστων οι υπηρεσίες ευελιξίας.

4.6.4 Συνάρτηση κόστους και εξισώσεις Bellman

Εάν υπάρχει τουλάχιστον μια εφικτή μετάβαση μεταξύ δύο καταστάσεων διαδοχικών σταδίων τότε, υπολογίζεται και αποθηκεύεται για κάθε τέτοια κατάσταση s_t κάθε σταδίου και για κάθε επιτρεπτή δράση a_t, το κόστος σταδίου c_t (s_t ,a_t). Στην προκειμένη, το κόστος αυτό εξαρτάται από την ωριαία τιμή της ηλεκτρικής ενέργειας λ_t, έχοντας θεωρήσει ότι ισχύει πολιτική δυναμικής τιμολόγησης. Η αντικειμενική συνάρτηση συνολικού κόστους που ελαχιστοποιεί ο κάθε πράκτορας φορτίου είναι J και στόχος του αλγόριθμου είναι να βρεθεί η ακολουθία δράσεων a_t που το ελαχιστοποιεί.

$$c_t(s_t, a_t) = \lambda_t * P_t^{\mathrm{i}} * \Delta\tau \tag{4.5}$$

$$\mathrm{J} = \sum_{t=0}^{N-1} c_t(s_t, a_t) \tag{4.6}$$

$$\min_{\{a_t\}_{t=0}^{N-1}} [\,\mathrm{J}\,] \tag{4.7}$$

όπου στην προκειμένη περίπτωση, ο πεπερασμένος χρονικός ορίζοντας T αντιστοιχεί σε χρονική διάρκεια 24 ωρών, ίση με τον ορίζοντα διαθεσιμότητας των τιμών ηλεκτρικής ενέργειας και P_t^{i}, η ισχύς του i-οστού φορτίου στο βήμα t. Η μεταβλητή t αναφέρεται στο t-οστό διακριτό χρονικό βήμα Δτ εντός του 24ώρου, με το πλήθος των χρονικών διαστημάτων να ισούται με $N = T/\Delta\tau$. Δεδομένου ότι η μπαταρία δεν λειτουργεί ως αυτόνομο φορτίο, αλλά ως στοιχείο εξισορρόπησης της συνολικής κατανάλωσης της οικίας, η βελτιστοποίησή της βασίζεται στο συνολικό φορτίο της οικίας που προκύπτει μετά τον προγραμματισμό των υπόλοιπων φορτίων. Για να διασφαλιστεί ότι η λειτουργία της μπαταρίας περιορίζεται αποκλειστικά σε αυτοκατανάλωση εντός της οικίας και δεν επιτρέπεται η εξαγωγή ισχύος προς το δίκτυο, επιβάλλεται ο ανισοτικός περιορισμός:

$$P_t^{\text{battery}} + P_t^{\text{house}} \geq 0 \tag{4.8}$$

όπου, P_t^{house} το συνολικό φορτίο της οικίας στο χρονικό βήμα t και P_t^{battery} η ισχύς φόρτισης ή εκφόρτισης της μπαταρίας.

Οι δύο βασικές εξισώσεις που χαρακτηρίζουν τον αλγόριθμο είναι οι συναρτήσεις Bellman [43]. Η συνάρτηση αξίας που εκφράζεται από την V_t, υπολογίζει το ελάχιστο δυνατό συνολικό κόστος που μπορεί να επιτευχθεί ξεκινώντας από την κατάσταση s στον χρόνο t, μέχρι το τέλος του χρονικού ορίζοντα. Η σχέση αυτή υποδεικνύει ότι η αξία μιας κατάστασης ισούται με το άθροισμα του τρέχοντος κόστους για μια συγκεκριμένη δράση και του ελάχιστου κόστους που μπορεί να επιτευχθεί στην συνέχεια. Η συνάρτηση (4.10), είναι η συνάρτηση βέλτιστης πολιτικής π_t, η οποία προσδιορίζει ποια συγκεκριμένη δράση πρέπει να αναλάβει ο πράκτορας όταν το σύστημα βρίσκεται στην κατάσταση s και στάδιο t. Η συνάρτηση πολιτικής λοιπόν, μπορεί να ερμηνευτεί ως ένας πίνακας αποφάσεων, όπου για κάθε πιθανή κατάσταση του συστήματος καθορίζει την δράση που ελαχιστοποιεί το συνολικό κόστος. Συγκεκριμένα:

$$V_t(s) = \min_{a \in \mathcal{A}(s)} \{ c_t(s,a) + V_{t+1}(f(s,a,t)) \}, \qquad t = N-1, \dots, 0 \tag{4.9}$$

$$\pi_t^*(s) = \arg\min_{a \in \mathcal{A}(s)} \{ c_t(s,a) + V_{t+1}(f(s,a,t)) \} \tag{4.10}$$

όπου $\arg\min\{V_t(s)\}$ επιστρέφει την τιμή s για την οποία η συνάρτηση βέλτιστης πολιτικής ελαχιστοποιείται. Αφού υλοποιηθούν οι πίνακες με τα κόστη μετάβασης και τις δράσεις μετάβασης, ο κάθε πράκτορας φορτίου διαθέτει πλέον την δυνατότητα προγραμματισμού της βέλτιστης «τροχιάς» λειτουργίας, του φορτίου που ελέγχει, για το επερχόμενο χρονικό διάστημα 24ων ωρών, ή ισοδύναμα, των 95 διακριτών χρονικών βημάτων. Στην συνέχεια, ο πράκτορας οικίας συντονίζει τα επιμέρους αποτελέσματα συλλέγοντας τα προφίλ από όλους τους πράκτορες φορτίου.

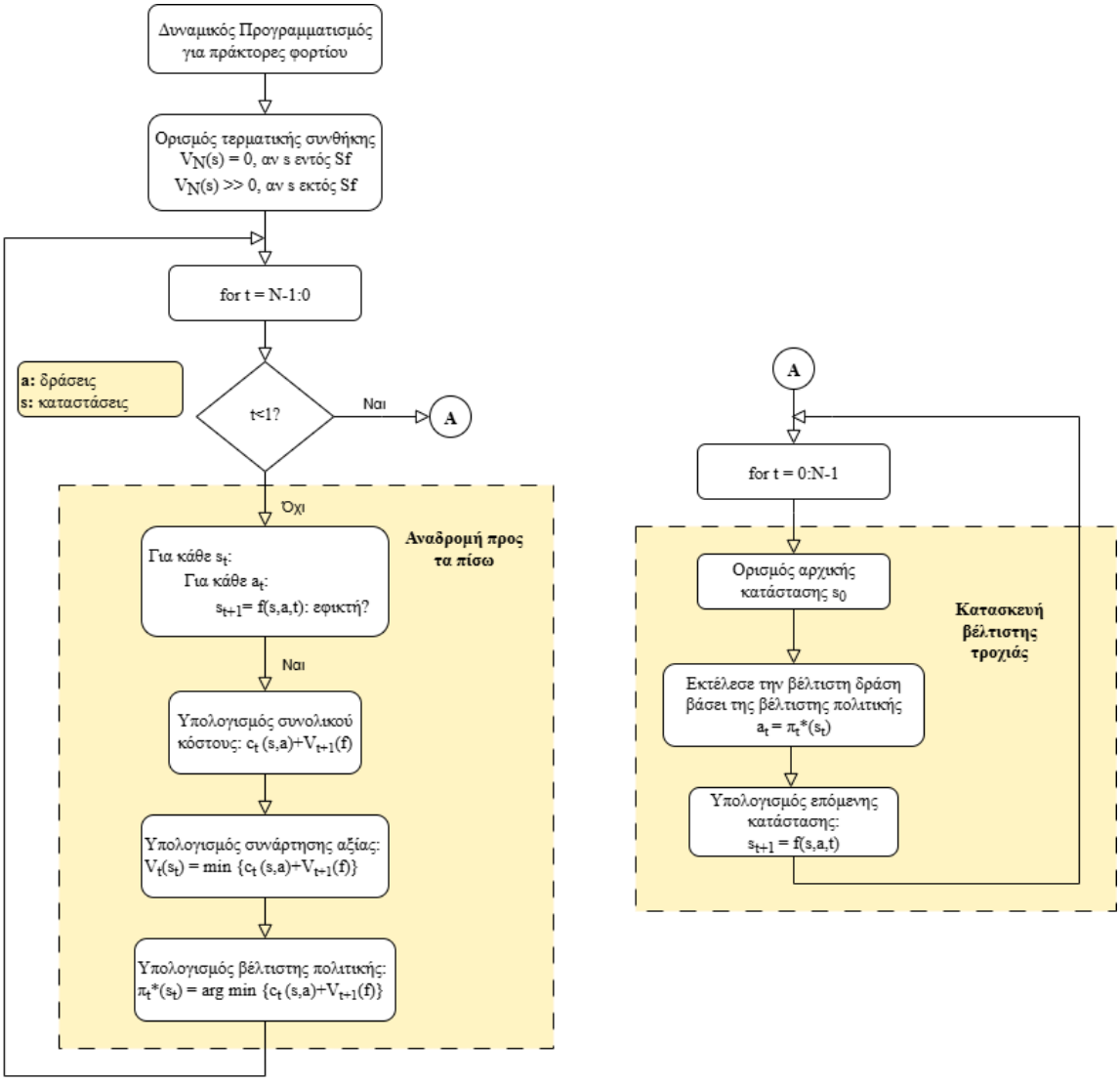
4.6.5 Κατασκευή βέλτιστης τροχιάς

Δεδομένης της αρχικής κατάστασης του κάθε φορτίου, ο πράκτορας φορτίου ακολουθεί την βέλτιστη πολιτική που έχει υπολογιστεί μέσω του ΔΠ για κάθε χρονικό στάδιο. Με τον τρόπο αυτό κατασκευάζεται μια ακολουθία δράσεων που οδηγεί σε προγραμματισμό λειτουργίας που ελαχιστοποιεί το συνολικό κόστος κατανάλωσης ενέργειας, χωρίς να θυσιάζει την άνεση του χρήστη. Για την εύρεση της τελικής λύσης,

επιβάλλεται ως οριακή συνθήκη τερματισμού ένα σύνολο επιθυμητών καταστάσεων S_f, έτσι ώστε να εξασφαλίζεται ότι ο αλγόριθμος συγκλίνει ομαλά σε εφικτές λύσεις:

$$S_f = \{|x - x^*| \leq \Delta x\} \tag{4.11}$$

Οι καταστάσεις που ανήκουν στο S_f θεωρούνται αποδεκτές και μηδενικού τερματικού κόστους, ενώ όλες οι υπόλοιπες καταστάσεις επιβαρύνονται με υψηλό κόστος (κάποιο κόστος ώστε να αποκλείονται στην συνέχεια, δεν έχει φυσική σημασία). Στην παρακάτω εικόνα απεικονίζεται ένα διάγραμμα ροής του αλγόριθμου που υλοποιήθηκε:



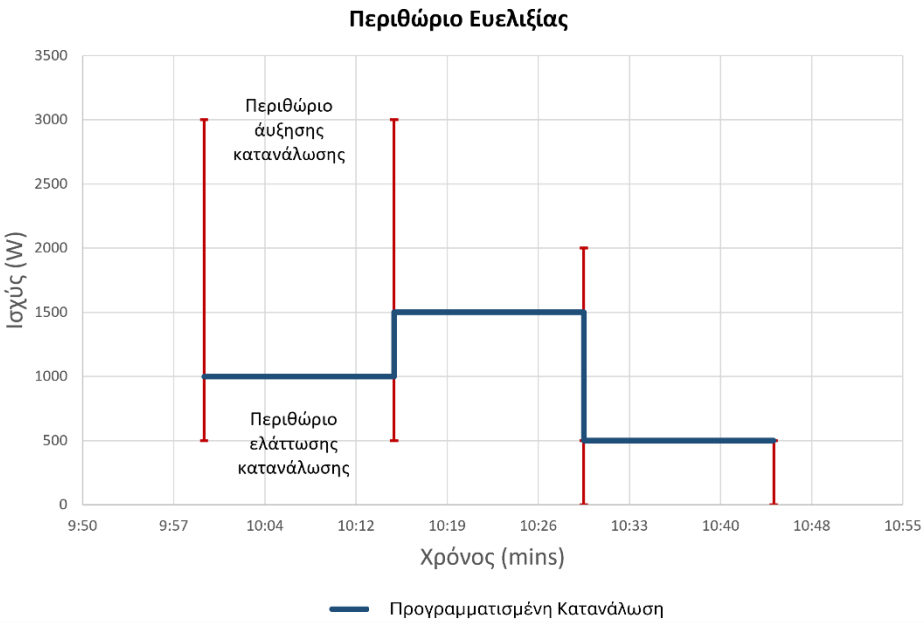
Εικόνα 16 Αλγόριθμος δυναμικού προγραμματισμού.

Η λύση του προβλήματος (βέλτιστη διαδρομή) υπολογίζεται με αντίστροφη αναδρομή. Ο υπολογισμός ξεκινά από το τελευταίο χρονικό βήμα t = N-1 εφαρμόζοντας τιμές τερματισμού και προχωρά αναδρομικά προς το πρώτο στάδιο t =

0. Για κάθε χρονικό στάδιο και για κάθε δυνατή κατάσταση s_t, εξετάζεται το σύνολο των επιτρεπτών δράσεων a_t. Για κάθε τέτοια δράση υπολογίζεται η επόμενη κατάσταση s_{t+1} και ελέγχεται η εφικτότητα της μετάβασης. Εφόσον η μετάβαση είναι εφικτή, υπολογίζεται το συνολικό κόστος που προκύπτει ως άθροισμα του κόστους του τωρινού σταδίου $c_t(s_t, a_t)$ και της συνάρτησης αξίας του επόμενου σταδίου $V_{t+1}(s_{t+1})$. Η ελάχιστη τιμή αυτού του κόστους καθορίζει τη συνάρτηση αξίας $V_t(s_t)$ για την αντίστοιχη κατάσταση και στάδιο, ενώ ταυτόχρονα αποθηκεύεται η δράση που την επιτυγχάνει, σχηματίζοντας τη βέλτιστη πολιτική $\pi_t^\star(s)$. Μετά την ολοκλήρωση της διαδικασίας, ο πράκτορας φορτίου διαθέτει στην μνήμη του έναν πλήρη πίνακα βέλτιστων δράσεων για κάθε κατάσταση. Η βέλτιστη τροχιά προκύπτει στη συνέχεια, ξεκινώντας από την αρχική κατάσταση s_0 και εφαρμόζοντας διαδοχικά τη βέλτιστη δράση που οδηγεί στο επόμενο στάδιο κάθε φορά και υπαγορεύεται από τη συνάρτηση βέλτιστης πολιτικής $\pi_t^\star(s)$. Η εφαρμογή της δράσης αυτής οδηγεί στον υπολογισμό της επόμενης κατάστασης μέσω του προσεγγιστικού φυσικού μοντέλου του φορτίου και η διαδικασία επαναλαμβάνεται έως την ολοκλήρωση του χρονικού ορίζοντα. Με τον τρόπο αυτό προκύπτει μια ακολουθία δράσεων, για τον πράκτορα φορτίου, που ελαχιστοποιεί το συνολικό κόστος κατανάλωσης ενέργειας, ενώ ταυτόχρονα ικανοποιεί τους λειτουργικούς περιορισμούς του φορτίου.

4.7 Εξαγωγή προφίλ ευελιξίας

Αφού εκτελεστεί ο αλγόριθμος ΔΠ, ο κάθε πράκτορας φορτίου έχει προγραμματίσει ένα προφίλ λειτουργίας του φορτίου που επιτηρεί και ελέγχει, δηλαδή τη βέλτιστη λειτουργία του για το δεδομένο χρονικό ορίζοντα με τον πιο οικονομικό τρόπο χωρίς την παροχή ευελιξίας. Στόχος είναι ο κάθε πράκτορας φορτίου να είναι ικανός να υπολογίζει το εφικτό εύρος λειτουργίας του φορτίου, δηλαδή τα όρια κατανάλωσης ισχύος του σε κάθε χρονικό διάστημα (βλ. Εικόνα 17). Τα όρια αυτά προκύπτουν λαμβάνοντας υπόψη τα όρια άνεσης του χρήστη, καθώς και τεχνικούς περιορισμούς των επιμέρους φορτίων και προκύπτουν από τις εξισώσεις κατάστασης που περιεγράφηκαν στην Ενότητα 3.3 (Ειδικά μοντέλα φορτίων).



Εικόνα 17 Προφίλ ευελιξίας.

Το αποτέλεσμα είναι η δημιουργία ενός προφίλ βάσης με άνω και κάτω όρια για τη δυνατότητα μεταβολής ισχύος, όπως φαίνεται στην Εικόνα 17, που συνιστούν το διαθέσιμο περιθώριο ευελιξίας. Το εύρος αυτό [t, ΔP(Δτ)] καθορίζει την ισχύ που μπορεί να αυξομειώσει ο κάθε πράκτορας φορτίου, εντός του χρονικού διαστήματος (t, t+ Δt). Ο πράκτορας οικίας προσφέρει αυτή την πληροφορία στον διαχειριστή, δηλαδή μια σαφώς ορισμένη υπηρεσία ευελιξίας.

4.7.1 Όρια ισχύος μπαταρίας

Για κάθε φορτίο ο αντίστοιχος πράκτορας υπολογίζει τα άνω και κάτω όρια της εφικτής ισχύος επιλύοντας την εξίσωση κατάστασης του φορτίου ως προς την ισχύ. Συγκεκριμένα, η εξίσωση κατάστασης αντιστρέφεται ως προς την ισχύ και η επόμενη κατάσταση αντικαθίσταται με την ελάχιστη και τη μέγιστη επιτρεπτή τιμή της, ώστε να εξαχθούν τα αντίστοιχα όρια ισχύος. Για τη μπαταρία, η εσωτερική κατάσταση του συστήματος περιγράφεται από την κατάσταση φόρτισης SoC_t. Τα όρια ισχύος προκύπτουν επιβάλλοντας τον περιορισμό ότι η επόμενη κατάσταση φόρτισης πρέπει να παραμείνει εντός των επιτρεπτών ορίων. Η μέγιστη επιτρεπτή εκφόρτιση, η οποία αντιστοιχεί στο κάτω όριο ισχύος, καθώς και η μέγιστη επιτρεπτή φόρτιση, που αντιστοιχεί στο άνω όριο ισχύος, προκύπτουν από τις εξισώσεις:

$$P_{min}^{SoC}(t) = \left((SoC_{min} - SoC_t) \cdot E_{cap} \cdot \eta_d\right) / \Delta t \qquad (4.12)$$

$$P_{max}^{SoC}(t) = \left((SoC_{max} - SoC_t) \cdot E_{cap}\right) / \Delta t \cdot \eta_c \quad (4.13)$$

Οι εξισώσεις αυτές εκφράζουν τη μέγιστη ποσότητα ενέργειας που μπορεί να απορροφηθεί ή να αποδοθεί από τη μπαταρία εντός του χρονικού διαστήματος Δτ, χωρίς να παραβιαστούν τα όρια της κατάστασης φόρτισης. Τα όρια αυτά εκφράζουν φυσικούς περιορισμούς κατάστασης, ανεξάρτητους από τα ηλεκτρικά χαρακτηριστικά της μπαταρίας. Για τον λόγο αυτό, τα τελικά όρια ισχύος που εκθέτει ο πράκτορας μπαταρίας στον πράκτορα οικίας προκύπτουν λαμβάνοντας επιπλέον υπόψη τα ονομαστικά όρια φόρτισης και εκφόρτισης της, όπως φαίνεται στις εξισώσεις:

$$P_{min}^{SoC}(t) = \max\left(P_{min}^{bat}, P_{min}^{SoC}(t)\right) \quad (4.14)$$

$$P_{max}^{SoC}(t) = \min\left(P_{max}^{bat}, P_{max}^{SoC}(t)\right) \quad (4.15)$$

4.7.2 Όρια ισχύος θερμοσίφωνα

Για τον πράκτορα του θερμοσίφωνα, η εσωτερική κατάσταση περιγράφεται από τη θερμοκρασία του αποθηκευμένου νερού T_t. Τα όρια ισχύος προκύπτουν από την απαίτηση η θερμοκρασία στο επόμενο χρονικό βήμα να παραμείνει εντός ενός άνω λειτουργικού ορίου και ενός κάτω ορίου άνεσης. Η απαιτούμενη ισχύς ώστε η θερμοκρασία να παραμείνει εντός των επιτρεπτών ορίων προκύπτει από τις εξισώσεις:

$$P_{min}^{T}(t) = \left(\left((T_{min} - a \cdot T_t)/(1-a) - T_{en}\right)/R_{ewh} + Q_{w,cold}(t)\right)/\eta_{boiler} \quad (4.16)$$

$$P_{max}^{T}(t) = \left(\left((T_{max} - a \cdot T_t)/(1-a) - T_{en}\right)/R_{th} + Q_{w,cold}(t)\right)/\eta_{boiler} \quad (4.17)$$

Τα τελικά όρια ισχύος που εκθέτει ο πράκτορας του θερμοσίφωνα στον πράκτορα οικίας λαμβάνουν επίσης υπόψη τα ονομαστικά όρια λειτουργίας:

$$P_{min}^{T}(t) = \max\left(0, P_{min}^{T}(t)\right) \quad (4.18)$$

$$P_{max}^{T}(t) = \min\left(P_{boiler,in}^{nominal}, P_{max}^{T}(t)\right) \quad (4.19)$$

4.7.3 Όρια ισχύος αντλίας

Για τον πράκτορα της αντλίας, η κατάσταση του συστήματος περιγράφεται από τον αποθηκευμένο όγκο νερού στη δεξαμενή V_t. Τα όρια ισχύος προκύπτουν επιβάλλοντας τον περιορισμό ότι ο όγκος της δεξαμενής στο επόμενο χρονικό βήμα πρέπει να παραμείνει εντός των επιτρεπτών ορίων λειτουργίας. Αντικαθιστώντας τον επόμενο όγκο με τα επιτρεπτά όρια, προκύπτουν τα όρια ισχύος:

$$P_{min}^{vol}(t) = \left(\left(V_{min} - V_t + V^{out,Pump}(t)\right) \cdot \rho \cdot g \cdot h\right) / \left(\eta_{pump} \cdot \Delta t\right) \quad (4.20)$$

$$P_{min}^{vol}(t) = \left(\left(V_{min} - V_t + V^{out,Pump}(t)\right) \cdot \rho \cdot g \cdot h\right) / \left(\eta_{pump} \cdot \Delta t\right) \quad (4.21)$$

Όπως και στους υπόλοιπους πράκτορες φορτίων, τα τελικά όρια ισχύος που εκθέτει ο πράκτορας της αντλίας στον πράκτορα οικίας λαμβάνουν επίσης υπόψη τα ονομαστικά όρια λειτουργίας:

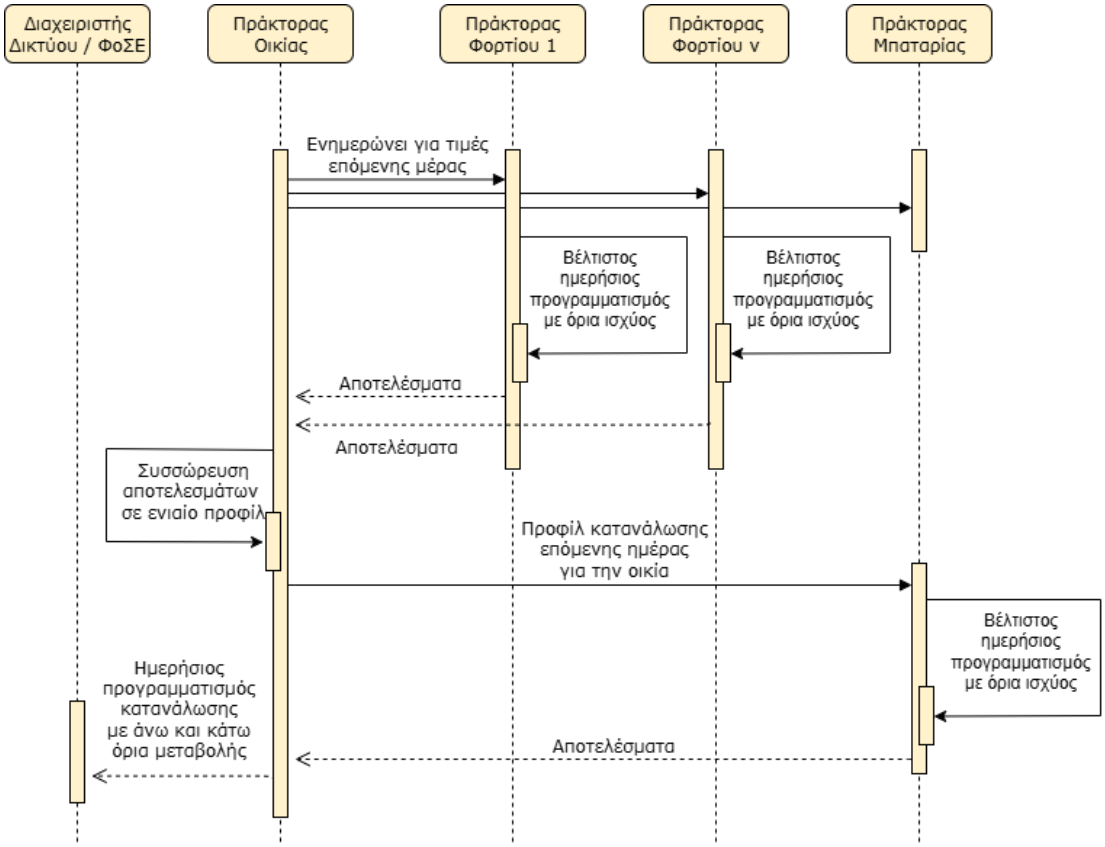
$$P_{min}^{vol}(t) = \max\left(P_{min}^{pump}, P_{min}^{vol}(t)\right) \quad (4.22)$$

$$P_{min}^{vol}(t) = \min\left(P_{max}^{pump}, P_{min}^{vol}(t)\right) \quad (4.23)$$

Με τον παραπάνω τρόπο, κάθε πράκτορας φορτίου λειτουργεί ως μαύρο κουτί που εκθέτει αποκλειστικά το προγραμματισμένο προφίλ βάσης λειτουργίας και τα αντίστοιχα άνω και κάτω όρια ισχύος. Ο πράκτορας οικίας δεν απαιτεί γνώση του φυσικού μοντέλου κάθε φορτίου, αλλά συντονίζει τα επιμέρους φορτία αποκλειστικά βάσει των εκτεθειμένων ορίων ισχύος. Η προσέγγιση αυτή επιτυγχάνει καθαρό διαχωρισμό αρμοδιοτήτων και απλοποιεί τον κεντρικό συντονισμό.

4.8 Κεντρικός συντονισμός

Στο προτεινόμενο πλαίσιο οικιακής μεσολάβησης μη κρίσιμων φορτίων, εισάγεται ο κεντρικός συντονισμός μέσω του πράκτορα οικίας. Οι επιμέρους πράκτορες φορτίων είναι αποκλειστικά υπεύθυνοι για την επιτήρηση και τον έλεγχο των αντίστοιχων φορτίων τους και δεν επικοινωνούν άμεσα μεταξύ τους. Δεδομένου ότι η μπαταρία δεν αποτελεί ένα φορτίο που εξυπηρετεί αυτόνομα μια σκοπιμότητα, αλλά έχει ως κύριο ρόλο τη λειτουργία της ως ενεργειακό απόθεμα, η λειτουργία της απαιτεί συντονισμό με τα υπόλοιπα φορτία της οικίας. Ο συντονισμός αυτός υλοποιείται μέσω του πράκτορα οικίας, ο οποίος αναλαμβάνει τον ρόλο της κεντρικής οντότητας που συντονίζει τα φορτία που αποκλειστικά καταναλώνουν ενέργεια με τη λειτουργία της μπαταρίας. Η διαδοχή των επιμέρους ενεργειών, ώστε η οικία να παρέχει τον ημερήσιο προγραμματισμό της, απεικονίζεται στο παρακάτω ακολουθιακό διάγραμμα:



Εικόνα 18 Ακολουθιακό διάγραμμα ημερήσιου προγραμματισμού.

Αρχικά, όταν δημοσιεύονται οι τιμές ηλεκτρικής ενέργειας από την αγορά της επόμενης ημέρας καταγράφονται από τον πράκτορα οικίας. Στη συνέχεια, ο πράκτορας οικίας ενημερώνει όλους τους πράκτορες φορτίων για το δυναμικό τιμολόγιο, το οποίο ισχύει για τον ημερήσιο χρονικό ορίζοντα βάσει των τιμών της αγοράς επόμενης ημέρας. Οι πράκτορες φορτίων, με εξαίρεση τον πράκτορα της μπαταρίας, υπολογίζουν τον βέλτιστο ημερήσιο προγραμματισμό λειτουργίας των φορτίων τους. Ο προγραμματισμός αυτός περιλαμβάνει το βασικό προφίλ λειτουργίας όσο και τα αντίστοιχα κατώτερα και ανώτερα όρια ισχύος σε κάθε χρονικό διάστημα. Τα αποτελέσματα αυτά αποστέλλονται στον πράκτορα οικίας. Ο πράκτορας οικίας συγκεντρώνει τα επιμέρους προφίλ των φορτίων και τα συσσωρεύει σε ένα ενιαίο προφίλ κατανάλωσης, το οποίο αποστέλλεται στον πράκτορα της μπαταρίας. Με βάση το δυναμικό τιμολόγιο και το συνολικό φορτίο της οικίας, ο πράκτορας της μπαταρίας υπολογίζει τον βέλτιστο ημερήσιο προγραμματισμό λειτουργίας της μπαταρίας και αποστέλλει το αποτέλεσμα στον πράκτορα οικίας. Τέλος, ο πράκτορας οικίας συνθέτει το τελικό ενιαίο προφίλ λειτουργίας της οικίας και το αποστέλλει στον ΦοΣΕ ή στον διαχειριστή του δικτύου.

4.9 Κυλιόμενος προγραμματισμός βάσει εισερχόμενων δεδομένων πραγματικού χρόνου

Στην πράξη, οι προβλέψεις που χρησιμοποιούνται στον ημερήσιο προγραμματισμό από τους πράκτορες φορτίων (π.χ. προφίλ κατανάλωσης ζεστού νερού, ζήτηση νερού στη δεξαμενή, κ.λπ.) ενδέχεται να παρουσιάσουν αποκλίσεις από την πραγματική συμπεριφορά του χρήστη. Επιπλέον, όταν ο διαχειριστής αποστέλλει εκκαθαρισμένη εντολή ευελιξίας $\Delta P_{\text{house}}^{\text{target}}(t)$, η υλοποίηση της εντολής απαιτεί προσαρμογή στις πραγματικές τρέχουσες καταστάσεις των φορτίων. Για τον λόγο αυτό, η προτεινόμενη αρχιτεκτονική εφαρμόζει κυλιόμενη βελτιστοποίηση σε διακριτό χρόνο, ώστε συνεχώς να βελτιώνεται και να επικαιροποιείται ο προτεινόμενος προγραμματισμός βάσει της πραγματικής κατάστασης για την οποία ενημερώνεται σε πραγματικό χρόνο. Σε κάθε χρονικό στάδιο t, κάθε πράκτορας φορτίου i εκτιμά την πραγματική τρέχουσα κατάσταση s_t του αντίστοιχου φορτίου (π.χ. T_t, V_t, SoC_t). Επιλύει εκ νέου το πρόβλημα βελτιστοποίησης, δηλαδή τον προγραμματισμό λειτουργίας, για τον υπόλοιπο χρονικό ορίζοντα [n, …, N-1):

$$\min_{\{a_t\}_{t=n}^{N-1}} \sum_{t=n}^{N-1} c_t(s_t, a_t) \tag{4.24}$$

Ο κάθε πράκτορας φορτίου εφαρμόζει την εντολή ευελιξίας μόνο για το διάστημα [n, n+1] και εκθέτει το νέο προφίλ ευελιξίας για τα επόμενα στάδια. Αυτή η διαδικασία επαναλαμβάνεται για όλο τον εξεταζόμενο χρονικό ορίζοντα, ώστε να διορθώνονται οι αποκλίσεις μεταξύ προβλέψεων και πραγματικότητας και να βελτιώνεται η αξιοπιστία της υπηρεσίας ευελιξίας προς τον διαχειριστή δικτύου.

4.10 Κατανομή εντολής ευελιξίας στα φορτία

Όταν ο πράκτορας οικίας λαμβάνει από το ανώτερο επίπεδο ελέγχου μια εκκαθαρισμένη εντολή ευελιξίας, αυτή εκφράζεται ως στόχος μεταβολής ισχύος $\Delta P_{\text{house}}^{\text{target}}(t)$ για κάθε χρονικό διάστημα t. Η εντολή αυτή μεταβάλλει τη συνολική ισχύ που οφείλει να καταναλώσει η οικία στο αντίστοιχο χρονικό στάδιο. Ο πράκτορας οικίας, έχοντας ήδη συλλέξει από τους πράκτορες φορτίων τα επιτρεπτά όρια ισχύος και το βασικό προφίλ λειτουργίας, καλείται να κατανείμει τη ζητούμενη μεταβολή ισχύος στα επιμέρους φορτία, τηρώντας όλους τους τεχνικούς περιορισμούς τους. Για

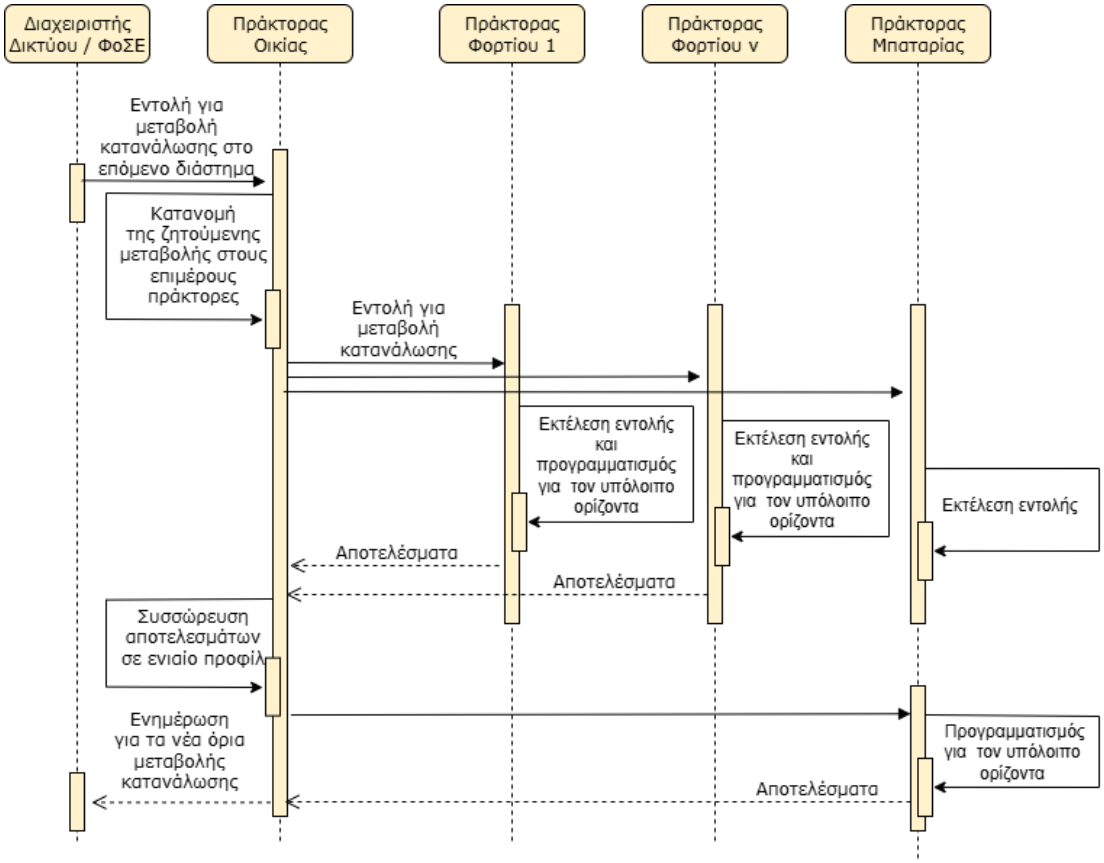
τον σκοπό αυτό επιλύεται το ακόλουθο πρόβλημα κατανομής, στο οποίο για κάθε χρονικό διάστημα ισχύει ο ισοτικός περιορισμός:

$$\sum_{t=0}^{N-1}\sum_{i}\Delta P^{i}(t) = \sum_{t=0}^{N-1}\Delta P_{house}^{target}(t) \tag{4.25}$$

Για κάθε φορτίο i, η μεταβολή ισχύος περιορίζεται από τα τοπικά όρια λειτουργίας του, που έχουν προκύψει κατά την εκτέλεση του δυναμικού προγραμματισμού:

$$\Delta P_{min}^{i}(t) \leq \Delta P^{i}(t) \leq \Delta P_{max}^{i}(t) \tag{4.26}$$

Η λύση του προβλήματος προσδιορίζει τα προτεινόμενα σημεία λειτουργίας για κάθε φορτίο, τα οποία αποστέλλονται στους επιμέρους πράκτορες φορτίων ώστε να τα υλοποιήσουν, όπως απεικονίζεται στην Εικόνα 19. Μετά την επίλυση του προβλήματος κατανομής από τον πράκτορα οικίας, οι πράκτορες φορτίων, με τη σειρά τους, εκτελούν την εντολή για το τρέχον χρονικό βήμα και λαμβάνοντας υπόψη την πραγματική τρέχουσα κατάσταση του φορτίου τους, υπολογίζουν εκ νέου τον βέλτιστο προγραμματισμό τους για τον υπόλοιπο χρονικό ορίζοντα. Η διαδικασία αυτή υλοποιείται σε κυλιόμενο χρονικό ορίζοντα, επιτρέποντας την προσαρμογή του προγραμματισμού σε αποκλίσεις μεταξύ προβλεπόμενων και πραγματικών συνθηκών λειτουργίας.



Εικόνα 19 Κατανομή εντολής ευελιξίας και κυλιόμενος προγραμματισμός.

Στο πλαίσιο αυτό, ο πράκτορας οικίας καλείται εκ νέου να συντονίσει τη λειτουργία του πράκτορα της μπαταρίας με τους πράκτορες των λοιπών φορτίων, ώστε να διασφαλίζεται η συνεπής και ασφαλής κατανομή της ευελιξίας στο επίπεδο της οικίας. Μετά τον επαναϋπολογισμό των επιμέρους προγραμμάτων, ο πράκτορας οικίας συνθέτει το νέο συνολικό προγραμματισμό κατανάλωσης και τα επικαιροποιημένα όρια δυνατότητας μεταβολής της ισχύος. Τέλος, ο πράκτορας οικίας ενημερώνει τον διαχειριστή δικτύου ή τον ΦοΣΕ για τον νέο προγραμματισμό και τα αντίστοιχα όρια ευελιξίας. Με τον τρόπο αυτό ελέγχεται ότι οι καταστάσεις των φορτίων, όπως η θερμοκρασία του νερού στον θερμοσίφωνα ή το επίπεδο φόρτισης της μπαταρίας, παραμένουν εντός των επιτρεπτών λειτουργικών ορίων, επιβεβαιώνοντας ότι η παρεχόμενη ευελιξία δεν παραβιάζει κανέναν λειτουργικό περιορισμό των φορτίων, ούτε τα όρια άνεσης.

4.11 Συμπεράσματα

Στο παρόν κεφάλαιο παρουσιάστηκε ένα πολυπρακτορικό πλαίσιο για τη διαχείριση μη-κρίσιμων οικιακών φορτίων με στόχο την αξιοποίηση της ευελιξίας στην πλευρά

της ζήτησης. Αναπτύχθηκε μια ιεραρχική αρχιτεκτονική πολλών πρακτόρων, η οποία επιτρέπει τον διαχωρισμό μεταξύ αποκεντρωμένου ελέγχου σε επίπεδο επιμέρους φορτίων και κεντρικού συντονισμού σε επίπεδο οικίας. Η υιοθέτηση μιας προσέγγισης τύπου «μαύρου κουτιού» για τους πράκτορες φορτίων, αλλά και για τον πράκτορα οικίας, ενισχύει την προστασία της ιδιωτικότητας των τελικών χρηστών και απλοποιεί τη διαλειτουργία της οικίας με ανώτερα επίπεδα ελέγχου, όπως ο διαχειριστής δικτύου ή ο ΦοΣΕ. Στη συνέχεια, ο δυναμικός προγραμματισμός χρησιμοποιήθηκε ως εργαλείο λήψης αποφάσεων για τον βέλτιστο προγραμματισμό των φορτίων, λαμβάνοντας υπόψη τις τιμές ενέργειας και τους τεχνικούς περιορισμούς. Με αυτό τον τρόπο, πεπεραστικά προβλήματα μη γραμμικότητας λόγω της συνθετότητας των επι μέρους ελέγχων των ρυθμιζόμενων φορτίων αντιμετωπίστηκαν και εξασφαλίστηκε για το χρήστη το ολικό βέλτιστο στην παρεχόμενη ευελιξία προς το δίκτυο. Παράλληλα, παρουσιάστηκε ο κυλιόμενος προγραμματισμός ως εγγενής λειτουργικότητα του προτεινόμενου πλαισίου, η οποία επιτρέπει τη συνεχή προσαρμογή των αποφάσεων ελέγχου σε αποκλίσεις μεταξύ προβλεπόμενων και πραγματικών συνθηκών λειτουργίας. Τέλος, παρουσιάστηκε η διαδικασία εκτέλεσης εντολών ευελιξίας μέσω της κατανομής της ζητούμενης μεταβολής ισχύος στα επιμέρους φορτία από τον πράκτορα οικίας. Το κεφάλαιο αποσαφήνισε ότι ένας οικιακός πράκτορας μπορεί να λειτουργήσει ως αυτόνομος μηχανισμός αξιοποίησης της σωρευτικής ευελιξίας στην πλευρά της ζήτησης. Στο επόμενο κεφάλαιο, το προτεινόμενο πλαίσιο εφαρμόζεται σε ένα περιβάλλον μικροδικτύου, το οποίο αποτελεί παράδειγμα ενός ΣΗΕ με αυξημένες ανάγκες συντονισμού παραγωγής και ζήτησης. Η χρήση του μικροδικτύου ως ενδεικτικής περίπτωσης μελέτης επιτρέπει την αξιολόγηση της συνεισφοράς της οικιακής ευελιξίας στη συνολική λειτουργία του συστήματος.

5 Οικονομική κατανομή φορτίου σε μικροδίκτυο με αξιοποίηση υπηρεσιών ευελιξίας από οικίες

5.1 Εισαγωγή

Η ανάπτυξη και υλοποίηση ενός οικιακού μεσολαβητή διαχείρισης μη κρίσιμων φορτίων αποκτά πρακτική σημασία όταν εντάσσεται σε ένα ευρύτερο πλαίσιο διαχείρισης ενέργειας από ΣΗΕ. Αν και η παρούσα διπλωματική εργασία εστιάζει στον σχεδιασμό και τη λειτουργία του συστήματος των πρακτόρων φορτίων και οικίας, η πραγματική αξία της ευελιξίας αναδεικνύεται όταν η οικία συμμετέχει ενεργά σε ένα ΣΗΕ με ενεργή συμμετοχή μικροπαραγωγών και καταναλωτών όπως είναι το μικροδίκτυο. Για τον λόγο αυτό, στο παρόν κεφάλαιο το μικροδίκτυο χρησιμοποιείται ως ενδεικτική περίπτωση χρήσης του προτεινόμενου μεσολαβητή, με στόχο την αξιολόγηση της συμβολής της οικιακής ευελιξίας στην λειτουργία του. Η ευελιξία στην πλευρά της ζήτησης ενσωματώνεται στο πρόβλημα κατανομής ως ένας επιπλέον βαθμός ελευθερίας, ο οποίος επιτρέπει στον Διαχειριστή του Μικροδικτύου (ΔΜ) να προσαρμόζει ως ένα βαθμό την κατανάλωση των οικιών στις συνθήκες λειτουργίας του μικροδικτύου. Στις επόμενες ενότητες διατυπώνεται το πρόβλημα της οικονομικής κατανομής φορτίου και η αντίστοιχη μαθηματική του μοντελοποίηση και στη συνέχεια παρουσιάζεται το υπό μελέτη μικροδίκτυο.

5.2 Οικονομική Κατανομή Φορτίου

Η Οικονομική Κατανομή Φορτίου (ΟΚΦ) είναι ένα από τα βασικά προβλήματα βελτιστοποίησης στα ΣΗΕ. Αφορά τον καθορισμό του τρόπου λειτουργίας του συστήματος σε βραχυχρόνιο χρονικό ορίζοντα και στόχος της είναι να καλύπτεται η ζήτηση ισχύος σε κάθε χρονική περίοδο με το μικρότερο δυνατό λειτουργικό κόστος. Παράλληλα, πρέπει να τηρούνται οι λειτουργικοί περιορισμοί των μονάδων παραγωγής. Η ΟΚΦ μπορεί να θεωρηθεί ειδική περίπτωση του προβλήματος βέλτιστης ροής ισχύος, όπου επιπλέον λαμβάνονται υπόψη περιορισμοί του δικτύου μεταφοράς. Στην πράξη, χρησιμοποιούνται συχνά απλουστευμένα γραμμικά μοντέλα, διότι σε εφαρμογές πραγματικού χρόνου ή ενδοημερήσιου προγραμματισμού τα μοντέλα αυτά επιτρέπουν γρήγορη και αξιόπιστη επίλυση [44]. Η ΟΚΦ επιλύεται σε διακριτά χρονικά βήματα και συνήθως χρησιμοποιείται χρονική ανάλυση μίας ώρας ή δεκαπέντε λεπτών. Σε κάθε χρονικό βήμα, ο διαχειριστής του συστήματος αποφασίζει ποιοι

ενεργειακοί πόροι θα χρησιμοποιηθούν και καθορίζει το επίπεδο ισχύος κάθε πόρου όπως και τη σειρά ενεργοποίησής τους. Οι αποφάσεις αυτές βασίζονται στο κόστος λειτουργίας, στη διαθεσιμότητα των πόρων και στους περιορισμούς του δικτύου.

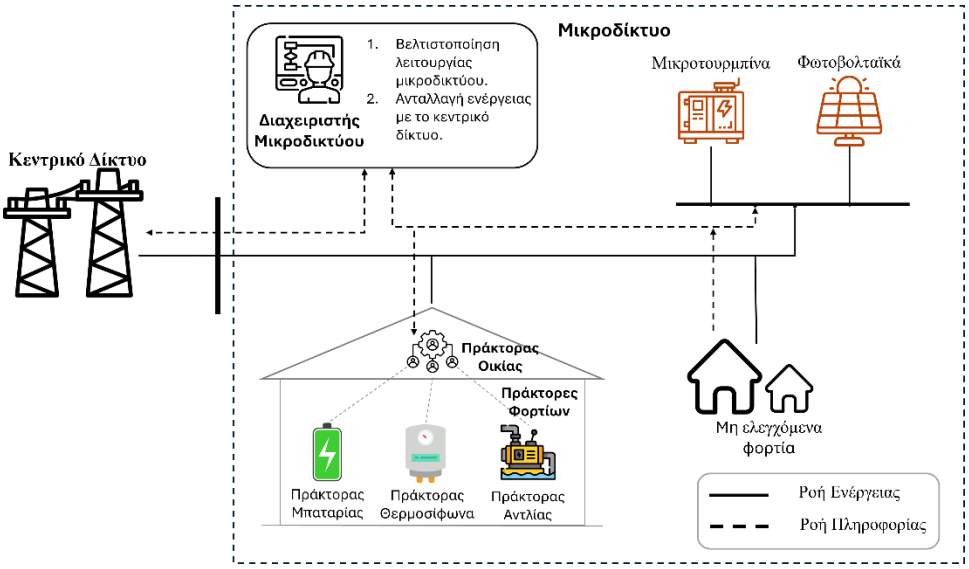
Στο πλαίσιο ενός μικροδικτύου, η ΟΚΦ περιλαμβάνει τόσο ελεγχόμενες όσο και μη ελεγχόμενες πηγές ενέργειας. ΑΠΕ όπως τα φωτοβολταϊκά συστήματα ανήκουν στη δεύτερη κατηγορία. Η παραγόμενη ισχύς τους θεωρείται δεδομένη για κάθε χρονικό βήμα και συνήθως ίση με τη μέγιστη που μπορεί τεχνικά να απορροφηθεί από το δίκτυο. Αντίθετα, συμβατικές μονάδες παραγωγής όπως οι μικροτουρμπίνες ή οι μονάδες αποθήκευσης όπως οι μπαταρίες μπορούν να ελεγχθούν από τον διαχειριστή. Η ύπαρξη ενέργειας και φορτίων που μπορούν να προσφέρουν ευελιξία στην πλευρά της ζήτησης αυξάνει τις δυνατότητες βελτιστοποίησης της λειτουργίας του συστήματος. Οι πόροι αυτοί επιτρέπουν τη μετατόπιση της κατανάλωσης ή της παραγωγής από ΑΠΕ στον χρόνο. Για παράδειγμα, σε περιόδους αυξημένης παραγωγής από φωτοβολταϊκά ο διαχειριστής μπορεί να επιλέξει τη φόρτιση των μπαταριών ή να αυξήσει την κατανάλωση μη κρίσιμων φορτίων. Με τον τρόπο αυτό αξιοποιείται καλύτερα η τοπική παραγωγή. Επιτρέπουν επίσης την αποθήκευση ενέργειας όταν το κόστος είναι χαμηλό, ενώ αντίστοιχα δίνουν τη δυνατότητα μείωσης της κατανάλωσης ή εκφόρτισης των μπαταριών όταν το κόστος είναι υψηλό. Αντίθετα, όταν η ενέργεια από το κεντρικό δίκτυο είναι ακριβή, μπορεί να μειωθεί η κατανάλωση ή να χρησιμοποιηθεί αποθηκευμένη ενέργεια.

Με τη χρήση της ΟΚΦ σε ένα σύστημα με τα παραπάνω χαρακτηριστικά, ο διαχειριστής του μικροδικτύου μπορεί να μειώσει το συνολικό κόστος λειτουργίας αυξάνοντας την αξιοποίηση της τοπικής παραγωγής. Τέλος, μπορεί να λαμβάνει αποφάσεις για την εισαγωγή ή την εξαγωγή ενέργειας από και προς το κεντρικό δίκτυο. Στην παρούσα εργασία, η οικιακή ευελιξία αντιμετωπίζεται ως ελεγχόμενος πόρος στην πλευρά της ζήτησης. Με τον τρόπο αυτό διευρύνονται οι διαθέσιμες επιλογές και βελτιώνεται η συνολική απόδοση του συστήματος.

5.3 Περιγραφή του υπό Μελέτη Μικροδικτύου

Το υπό μελέτη μικροδίκτυο αποτελείται από ένα σύνολο φορτίων και μονάδων παραγωγής, τα οποία συντονίζονται από τον διαχειριστή του μικροδικτύου με στόχο την οικονομική κατανομή της ισχύος. Η ανάλυση περιορίζεται στη διασυνδεδεμένη λειτουργία, όπου το μικροδίκτυο έχει τη δυνατότητα να ανταλλάσσει ισχύ με το

κεντρικό δίκτυο μέσω του σημείου κοινής σύνδεσης. Στην Εικόνα 20 απεικονίζεται η δομή του υπό μελέτη μικροδικτύου.



Εικόνα 20 Μικροδίκτυο υπό μελέτη.

Στην πλευρά της ζήτησης, το μικροδίκτυο περιλαμβάνει αφενός μη ελεγχόμενα φορτία, των οποίων η κατανάλωση θεωρείται δεδομένη και δεν μεταβάλλεται και αφετέρου μη κρίσιμα φορτία, τα οποία διαθέτουν δυνατότητα μεταβολής της κατανάλωσής τους εντός κάποιων ορίων. Τα μη κρίσιμα φορτία αντιπροσωπεύουν την πηγή οικιακής ευελιξίας και ελέγχονται μέσω του προτεινόμενου οικιακού μεσολαβητή, ο οποίος εκθέτει στο επίπεδο του μικροδικτύου το προγραμματισμένο προφίλ λειτουργίας και τα αντίστοιχα άνω και κάτω όρια ισχύος. Στην πλευρά της παραγωγής, το μικροδίκτυο περιλαμβάνει φωτοβολταϊκά συστήματα, τα οποία μοντελοποιούνται ως στοχαστικές πηγές ενέργειας με μη ελεγχόμενη ισχύ, καθώς και μία μικροτουρμπίνα, η οποία λειτουργεί ως ελεγχόμενη μονάδα παραγωγής. Τέλος, το μικροδίκτυο διαθέτει σύνδεση με το κεντρικό δίκτυο, μέσω της οποίας μπορεί να εισάγει ή να εξάγει ισχύ, ανάλογα με τις συνθήκες λειτουργίας. Η ανταλλαγή αυτή επιτρέπει στο μικροδίκτυο να εξισορροπεί τη ζήτηση και την τοπική παραγωγή, λαμβάνοντας υπόψη τη τιμή της ενέργειας από και προς το κεντρικό δίκτυο.

5.4 Μαθηματική διατύπωση οικονομικής κατανομής φορτίου στο μικροδίκτυο

Στην παρούσα ενότητα παρουσιάζεται η μαθηματική διατύπωση του προβλήματος οικονομικής κατανομής φορτίου στο υπό μελέτη μικροδίκτυο. Στόχος

είναι η διαμόρφωση του προβλήματος βελτιστοποίησης, μέσω της αποτύπωσης των αλληλεπιδράσεων μεταξύ παραγωγής, ζήτησης και οικιακής ευελιξίας. Η διατύπωση πραγματοποιείται σε διακριτό χρονικό ορίζοντα και ακολουθεί τη λογική της κλασσικής ΟΚΦ, επεκταμένης ώστε να περιλαμβάνει ελεγχόμενους πόρους στην πλευρά της ζήτησης. Η οικιακή ευελιξία ενσωματώνεται στο ισοζύγιο ισχύος και στη συνάρτηση κόστους, επιτρέποντας στον Διαχειριστή του μικροδικτύου να λαμβάνει αποφάσεις τόσο για την παραγωγή όσο και για την προσαρμογή της κατανάλωσης. Στις επόμενες υποενότητες ορίζονται οι μεταβλητές απόφασης, οι συναρτήσεις κόστους και οι περιορισμοί λειτουργίας του συστήματος, οδηγώντας στη διατύπωση ενός κυρτού προβλήματος τετραγωνικού προγραμματισμού, το οποίο επιλύεται ανεξάρτητα σε κάθε χρονικό βήμα.

5.4.1 Μοντελοποίηση οικιακής ευελιξίας

Ο οικιακός πράκτορας παράγει σε κάθε χρονικό διάστημα ένα εύρος διαθέσιμης ευελιξίας ΔP_t^{flex} το οποίο εκφράζει τη δυνατότητα μείωσης ή αύξησης της κατανάλωσης της οικίας και ικανοποιεί τους περιορισμούς:

$$\Delta P_t^{-} \leq \Delta P_t^{flex} \leq \Delta P_t^{+} \tag{5.1}$$

όπου ΔP_t^{-} αντιστοιχεί σε μείωση κατανάλωσης, ενώ ΔP_t^{+} σε αύξηση. Θεωρούμε ότι ο Διαχειριστής μπορεί να αξιοποιήσει την ευελιξία προσαρμόζοντας την συνολική ζήτηση της οικίας, έναντι τιμής αποζημίωσης c^{flex} (€/kW).

5.4.2 Μοντελοποίηση μονάδων παραγωγής

Η διαθέσιμη ισχύς από ΑΠΕ πρέπει να απορροφάται από το μικροδίκτυο κατά προτεραιότητα, οπότε δεν εισάγεται κάποιο κόστος για την Φ/Β παραγωγή P_t^{PV}. Λόγω του ισοτικού περιορισμού ζήτησης με παραγωγής (5.15) θα δίνεται πάντα προτεραιότητα στην αξιοποίησή της, αλλά σε περίπτωση περίσσειας ισχύος, επιτρέπεται η περικοπή της παραγωγής:

$$0 \leq P_{used}^{PV} \leq P_t^{PV} \tag{5.2}$$

όπου P_t^{PV}, η διαθέσιμη παραγωγή από Φ/Β σε κάθε χρονικό διάστημα και P_{used}^{PV} η ισχύς που αξιοποιήθηκε. Η συμβατική παραγωγή (όπως μια μικροτουρμπίνα) αποτελεί το μόνο πλήρως ελεγχόμενο μέσο παραγωγής στο μικροδίκτυο. Το κόστος

λειτουργίας περιγράφεται από πολυωνυμική συνάρτηση δευτέρου βαθμού, όπως και στο κλασσικό πρόβλημα ΟΚΦ:

$$C^{MT}(P_t^{MT}) = a^{MT} * (P_t^{MT})^2 + b^{MT} * P_t^{MT} + c^{MT},\ P_{min}^{MT} \leq P_t^{MT} \leq P_{max}^{MT} \quad (5.3)$$

όπου a^{MT} ο τετραγωνικός όρος κόστους (€/kW^2), b^{MT} ο γραμμικός συντελεστής κόστους (€/kW) και c^{MT} ο σταθερός συντελεστής κόστους (€) της μονάδας παραγωγής. Η παραπάνω διατύπωση οδηγεί σε κυρτό πρόβλημα τετραγωνικού προγραμματισμού. Ωστόσο, στη συγκεκριμένη εφαρμογή ο σταθερός όρος c^{MT} αντιπροσωπεύει κόστος ενεργοποίησης της μονάδας και πρέπει να λαμβάνεται υπόψη μόνο όταν η μονάδα τίθεται σε λειτουργία. Η διατύπωση του τετραγωνικού προγραμματισμού δεν επιτρέπει την μοντελοποίηση απόφασης ενεργοποίησης/απενεργοποίησης. Για τον λόγο αυτό απαιτείται μετασχηματισμός του προβλήματος σε πρόβλημα μικτού ακέραιου γραμμικού προγραμματισμού (Mixed Integer Linear Programming - MILP), με γραμμικοποίηση της τετραγωνικής συνάρτησης κόστους της μικροτουρμπίνας [45]. Εισάγεται δυαδική μεταβλητή $u_t^{MT} \in 0,1$, όπου $u_t^{MT} = 1$ αν η μικροτουρμπίνα είναι ενεργή και $u_t^{MT} = 0$ αν είναι απενεργοποιημένη. Το σταθερό κόστος λειτουργίας γράφεται τώρα ως $u_t^{MT} * c^{MT}$ ώστε να επιβάλλεται μόνο όταν η μονάδα λειτουργεί. Οι όροι $a^{MT} * (P_t^{MT})^2 + b^{MT} * P_t^{MT}$, θα αντικατασταθούν από τμηματικά γραμμική προσέγγιση [46]. Το εύρος λειτουργίας της μικροτουρμπίνας $[P_{min}^{MT},\ P_{max}^{MT}]$, χωρίζεται σε m ίσα τμήματα και επιλέγεται ένα σύνολο σημείων P_m, όπου:

$$P_m = P_{min}^{MT} + (m-1) * \Delta P_m \quad (5.4)$$

$$\Delta P_m = (P_{max}^{MT} - P_{min}^{MT})/(m-1) \quad (5.5)$$

και για κάθε σημείο υπολογίζεται το κόστος:

$$C_m(P_m) = a^{MT} * (P_m)^2 + b^{MT} * P_m \quad (5.6)$$

Τα σημεία (P_m, C_m) ορίζουν την τμηματικά γραμμική προσέγγιση της αρχικής παραβολής. Η απλή επιλογή των σημείων δεν είναι επαρκής για τη διατύπωση του προβλήματος ως γραμμικού. Απαιτείται ένας μηχανισμός γραμμικής παρεμβολής μεταξύ διαδοχικών σημείων που να επιτρέπει στην παραγόμενη ισχύ P_t^{MT} να λαμβάνει οποιαδήποτε τιμή εντός του επιτρεπτού διαστήματος και ταυτόχρονα το κόστος C^{MT} να υπολογίζεται ως γραμμική συνάρτηση του P_t^{MT}. Για τον σκοπό αυτό χρησιμοποιείται η μέθοδος του κυρτού συνδυασμού η οποία αποτελεί τυπική διατύπωση τμηματικά γραμμικής προσέγγισης κυρτών συναρτήσεων. Για κάθε

χρονικό βήμα t εισάγονται βοηθητικές μεταβλητές βαρών $\lambda_{m,t} \geq 0$ οι οποίες ικανοποιούν τον περιορισμό:

$$\sum_{m=1} \lambda_{m,t} = u_t^{MT} \tag{5.7}$$

Η παραγόμενη ισχύς της μικροτουρμπίνας πλέον εκφράζεται ως:

$$P_t^{MT} = \sum_{m=1} P_m * \lambda_{m,t} \tag{5.8}$$

Το μεταβλητό κόστος λειτουργίας μετασχηματίστηκε σε:

$$a^{MT} * (P_m^{MT})^2 + b^{MT} * P_m^{MT} = \sum_{m=1} C_m * \lambda_{m,t} \tag{5.9}$$

Το συνολικό κόστος της μικροτουρμπίνας τελικά γράφεται ως:

$$C^{MT}(P_t^{MT}) = \left[\sum_{m=1} C_m * \lambda_{m,t}\right] + u_t^{MT} * c^{MT} \tag{5.10}$$

Ο κυρτός συνδυασμός χρειάστηκε διότι η τετραγωνική συνάρτηση κόστους της μικροτουρμπίνας είναι συνεχής και κυρτή. Η αντικατάσταση της από τμηματικά γραμμική προσέγγιση σημαίνει ότι μεταξύ δύο διαδοχικών σημείων (P_m, C_m) και (P_{m+1}, C_{m+1}) το κόστος πρέπει να υπολογίζεται μέσω γραμμικής παρεμβολής. Έστω ότι η ισχύς της μικροτουρμπίνας βρίσκεται μεταξύ [P_m, P_{m+1}] τότε εισάγονται οι βοηθητικές μεταβλητές $\lambda_{m,t}$ και $\lambda_{m+1,t}$ με:

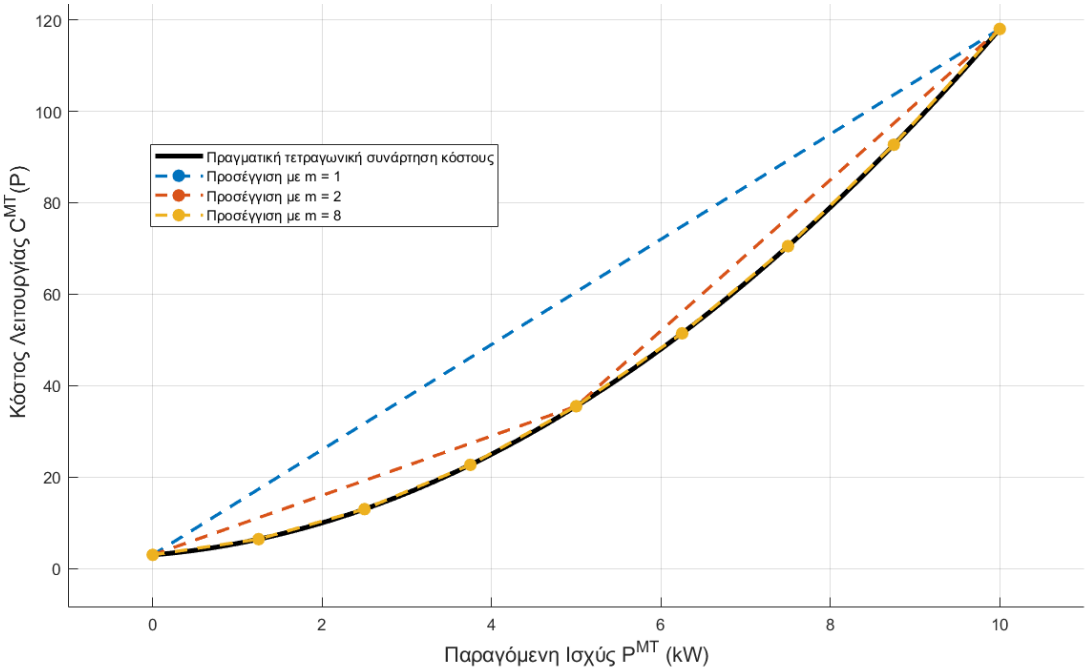
$$\lambda_{m,t} + \lambda_{m+1,t} = u_t^{MT} \tag{5.11}$$

Η παραγόμενη ισχύς και το κόστος λειτουργίας υπολογίζονται ως γραμμικός συνδυασμός αυτών των δύο σημείων:

$$P_t^{MT} = P_m * \lambda_{m,t} + P_{m+1} * \lambda_{m+1,t} \tag{5.12}$$

$$C^{MT} = C_m * \lambda_{m,t} + C_{m+1} * \lambda_{m+1,t} \tag{5.13}$$

Οπότε καταλήγουμε με ένα γραμμικό σύστημα τριών εξισώσεων και ο επιλυτής του MILP λύνει ως προς $\lambda_{m,t}$ και $\lambda_{m+1,t}$, έτσι ώστε να ελαχιστοποιείται η συνάρτηση στόχος υπό τους δεδομένους περιορισμούς.



Εικόνα 21 Τμηματικά γραμμική προσέγγιση τετραγωνικής συνάρτησης.

Στο παραπάνω γράφημα (Εικόνα 21) παρουσιάζεται η διαδικασία που περιεγράφηκε, δηλαδή η τμηματικά γραμμική προσέγγιση της τετραγωνικής συνάρτησης κόστους της μικροτουρμπίνας. Η μαύρη καμπύλη είναι η τετραγωνική συνάρτηση, ενώ οι διακεκομμένες γραμμές απεικονίζουν την τμηματικά γραμμική προσέγγιση της παραπάνω καμπύλης για διαφορετικό αριθμό τμημάτων m. Όπως φαίνεται στο γράφημα, όσο αυξάνεται το πλήθος των τμημάτων m τόσο αυξάνεται η ακρίβεια της προσέγγισης.

5.4.3 Ανταλλαγή ισχύος με το κεντρικό δίκτυο

Η ανταλλαγή ισχύος με το δίκτυο γίνεται μέσω μίας γραμμής σύνδεσης. Η ισχύς που εισάγεται στο μικροδίκτυο από το δίκτυο σε κάθε χρονικό διάστημα P_t^{grid}, περιορίζεται από τη μέγιστη ικανότητα της γραμμής σύνδεσης:

$$-P_{max}^{Line} \leq P_t^{grid} \leq P_{max}^{Line} \tag{5.14}$$

όπου P_{max}^{Line} μέγιστη ικανότητα μεταφοράς ισχύος της γραμμής και για $P_{max}^{Line} > 0$, το μικροδίκτυο εισάγει ισχύ, ενώ για $P_{max}^{Line} < 0$ εξάγει ισχύ στο κεντρικό δίκτυο. Σε κάθε χρονική στιγμή πρέπει να διατηρείται το ισοζύγιο ισχύος εντός του μικροδικτύου:

$$P_t^{MT} + P_t^{PV} + P_t^{grid} = L_t + \Delta P_t \tag{5.15}$$

όπου L_t είναι η ανελαστική προβλεπόμενη ζήτηση. Η παραπάνω εξίσωση εξασφαλίζει ότι η συνολική παραγόμενη και ανταλλασσόμενη με το δίκτυο ισχύς ισούται με το άθροισμα της ανελαστικής ζήτησης και της μεταβολής που προκύπτει για το συνολικό φορτίο από την αξιοποίηση της οικιακής ευελιξίας.

5.5 Συνάρτηση στόχος

Ο στόχος του διαχειριστή μικροδικτύου είναι η ελαχιστοποίηση του συνολικού κόστους λειτουργίας του μικροδικτύου και το πρόβλημα επιλύεται ενδοημερήσια με τα επικαιροποιημένα δεδομένα σε κάθε χρονικό διάστημα t για το επόμενο διάστημα (t+1). Το κόστος αυτό περιλαμβάνει το κόστος λειτουργίας της μικροτουρμπίνας, το κόστος αγοράς ενέργειας από το δίκτυο ή το κέρδος πώλησης ενέργειας προς το δίκτυο και την αποζημίωση για ενεργοποίηση οικιακής ευελιξίας. Η αντικειμενική συνάρτηση δίνεται στην:

$$j = \min[C^{MT}(P_t^{MT}) + c_t^{grid} * P_t^{grid} + c_t^{flex} * |\Delta P_t^{flex}|] \quad (5.16)$$

Για την επίλυση του προβλήματος ορίζεται το διάνυσμα μεταβλητών απόφασης x_t για κάθε χρονική στιγμή t, όπου:

$$x_t = \left[P_t^{MT}\ P_t^{grid}\ \Delta P_t^{flex}\ u_t^{MT}\ \lambda_{1,t}\ \lambda_{2,t}\ \dots\ \lambda_{m,t}\right]^T$$

Το πρόβλημα ΟΚΦ που επιλύει ο Διαχειριστής Μικροδικτύου αποτελεί ένα πρόβλημα μικτού ακέραιου γραμμικού προγραμματισμού με γραμμική αντικειμενική συνάρτηση, καθώς και γραμμικούς και ανισοτικούς περιορισμούς. Ουσιαστικά ο Διαχειριστής, επιλύοντας την ΟΚΦ, καθορίζει με ποιον τρόπο το μικροδίκτυο θα καλύψει την ζήτηση του στο ελάχιστο δυνατό κόστος. Σε κάθε χρονικό βήμα θα πρέπει να ικανοποιεί το ισοζύγιο ισχύος και να τηρεί τους τεχνικούς περιορισμούς των μονάδων. Για να μεγιστοποιήσει την κατανάλωση από παραγωγή Φ/Β, ενεργοποιεί υπηρεσία ανοδικής εφεδρείας από την ευέλικτη ζήτηση μετατοπίζοντας έτσι το φορτίο τις ώρες όπου υπάρχει περίσσεια παραγωγή. Αντίστοιχα, τις ώρες όπου δεν υπάρχει παραγωγή από Φ/Β επιλέγει να ενεργοποιήσει υπηρεσία καθοδικής εφεδρείας, διότι είναι πιο φθηνή επιλογή από το να καλύψει το φορτίο αγοράζοντας ενέργεια από το δίκτυο ή ενεργοποιώντας την μικροτουρμπίνα.

5.6 Συμπεράσματα

Στο παρόν κεφάλαιο διατυπώθηκε το πρόβλημα της οικονομικής κατανομής φορτίου σε ένα διασυνδεδεμένο μικροδίκτυο, με ενσωμάτωση υπηρεσιών ευελιξίας από την πλευρά της ζήτησης. Το μικροδίκτυο χρησιμοποιήθηκε ως ενδεικτική περίπτωση χρήσης, με σκοπό να αναδειχθεί ο ρόλος και η προστιθέμενη αξία του προτεινόμενου οικιακού μεσολαβητή στο επίπεδο λειτουργίας ενός ΣΗΕ. Η οικιακή ευελιξία μοντελοποιήθηκε ως ελεγχόμενος πόρος, ο οποίος εκφράζεται μέσω άνω και κάτω ορίων μεταβολής της κατανάλωσης και ενός κόστους ενεργοποίησης της ευελιξίας και ενσωματώνεται απευθείας στο ισοζύγιο ισχύος του μικροδικτύου. Με τον τρόπο αυτό, η ευελιξία δεν αντιμετωπίζεται ως εξωγενής ή επικουρικός μηχανισμός, αλλά ως μεταβλητή απόφασης στο πρόβλημα της ΟΚΦ. Η προσέγγιση αυτή επιτρέπει στον Διαχειριστή του μικροδικτύου να προσαρμόζει τη ζήτηση, αξιοποιώντας τη διαθεσιμότητα μη κρίσιμων φορτίων, με στόχο τη μείωση του συνολικού κόστους λειτουργίας του.

Το πρόβλημα ΟΚΦ διατυπώθηκε ως πρόβλημα μικτού ακέραιου γραμμικού προγραμματισμού, λόγω της ύπαρξης αρχικού κόστους όταν αποφασίζεται η λειτουργία της συμβατικής μονάδας, με γραμμικούς και ανισοτικούς περιορισμούς. Από τη διατύπωση αυτή, καθίσταται σαφές ότι η αξιοποίηση της οικιακής ευελιξίας παρέχει στον Διαχειριστή επιπλέον βαθμούς ελευθερίας λειτουργίας του μικροδικτύου, τόσο σε περιόδους αυξημένης τοπικής παραγωγής όσο και σε περιόδους περιορισμένης διαθεσιμότητας ΑΠΕ. Στο επόμενο κεφάλαιο παρουσιάζονται οι προσομοιώσεις που εκτελέστηκαν και τα αντίστοιχα αποτελέσματα, μέσω των οποίων αναλύεται η συμπεριφορά του συστήματος υπό διαφορετικά σενάρια λειτουργίας και αποτιμάται στην πράξη η επίδραση της ευελιξίας στη μείωση του κόστους και στη βελτιστοποίηση της λειτουργίας του μικροδικτύου.

6 Προσομοιώσεις και αποτελέσματα

6.1 Εισαγωγή

Στο παρόν κεφάλαιο παρουσιάζονται τα αποτελέσματα των προσομοιώσεων που πραγματοποιήθηκαν για την αξιολόγηση του προτεινόμενου πολυπρακτορικού συστήματος διαχείρισης οικιακής ευελιξίας. Αρχικά εξετάζεται η αυτόνομη λειτουργία των πρακτόρων φορτίων σε επίπεδο οικίας, υπό διαφορετικές πολιτικές τιμολόγησης, με στόχο την αποτίμηση της ικανότητάς τους να προσαρμόζουν τη λειτουργία των μη κρίσιμων φορτίων βάσει οικονομικών σημάτων και λειτουργικών περιορισμών. Στη συνέχεια, η διαθέσιμη οικιακή ευελιξία ενσωματώνεται σε επίπεδο μικροδικτύου και αξιοποιείται στο πλαίσιο της ΟΚΦ. Εξετάζονται διαφορετικές συνθήκες τοπικής παραγωγής, όπως πλεόνασμα και έλλειμμα Φ/Β παραγωγής, προκειμένου να διερευνηθεί ο ρόλος της ευελιξίας στη βελτίωση της διατήρησης ισοζυγίου ισχύος και στη διαμόρφωση οικονομικής στρατηγικής. Στόχος του κεφαλαίου είναι να τεκμηριωθεί η λειτουργικότητα του προτεινόμενου πλαισίου διαχείρισης ευελιξίας που αναπτύχθηκε, καθώς και η συμβολή της οικιακής ευελιξίας στη βέλτιστη λειτουργία του ΣΗΕ.

6.2 Αυτόνομη λειτουργία πρακτόρων οικιακών φορτίων

Για την αξιολόγηση του πολυπρακτορικού οικιακού μεσολαβητή που υλοποιήθηκε, εξετάζονται δύο σενάρια προσομοίωσης τα οποία εστιάζουν αποκλειστικά στη λειτουργία των πρακτόρων φορτίων. Η συνολική διάρκεια της προσομοίωσης είναι 12 ώρες με χρονικό βήμα 15 λεπτών. Οι πράκτορες φορτίων προγραμματίζουν τη λειτουργία των μη κρίσιμων φορτίων που ελέγχουν σε ημερήσιο ορίζοντα βάσει των οικονομικών σημάτων, δηλαδή των τιμών της αγοράς επόμενης ημέρας και των προβλεπόμενων προφίλ κατανάλωσης ζεστού νερού. Εντός της ημέρας πραγματοποιείται επαναπρογραμματισμός σε κάθε χρονικό βήμα, ώστε να λαμβάνονται υπόψη αποκλίσεις από τις προβλεπόμενες συνθήκες, όπως για παράδειγμα μεγαλύτερη κατανάλωση ζεστού νερού από τη προβλεπόμενη. Ο ενδοημερήσιος επαναπρογραμματισμός επιτρέπει στους πράκτορες να προσαρμόσουν τη λειτουργία των φορτίων καθώς οι προβλέψεις ζήτησης διορθώνονται σε πραγματικό χρόνο, διασφαλίζοντας τη τήρηση των περιορισμών άνεσης. Σε κάθε χρονικό διάστημα, ο διαχειριστής δικτύου αποστέλλει εντολές ευελιξίας προς την οικία, οπότε

ο πράκτορας φορτίου κατανείμει την εντολή στους πράκτορες φορτίου, δηλαδή την ισχύ με την οποία θα πρέπει να λειτουργήσουν και αυτοί με την σειρά τους υπολογίζουν τον επαναπρογραμματισμό των φορτίων τους για τον υπόλοιπο χρονικό ορίζοντα. Στόχος της ανάλυσης είναι να αξιολογηθεί η ικανότητα κάθε πράκτορα φορτίου να προσαρμόζει τη λειτουργία του φορτίου που ελέγχει, λαμβάνοντας υπόψη τόσο τις εσωτερικές καταστάσεις της οικίας (π.χ. κατάσταση φόρτισης μπαταρίας, θερμοκρασία θερμοσίφωνα, στάθμη δεξαμενής άντλησης) όσο και τα εξωτερικά οικονομικά σήματα (τιμές τιμολογόγησης).

Η παραμετροποίηση των πρακτόρων των μη κρίσιμων οικιακών φορτίων που χρησιμοποιούνται στα σενάρια προσομοίωσης παρουσιάζεται στον Πίνακα Πίνακας 6. Συνοψίζονται τα κύρια μεγέθη των φορτίων τα οποία αξιοποιεί ως δεδομένα εισόδου ο κάθε πράκτορας φορτίου, ενώ πιο αναλυτικά η παραμετροποίηση φαίνεται στο Παράρτημα όπου έχει περιληφθεί ο κώδικας υλοποίησής τους (Κεφάλαιο 9). Η εστίαση δίνεται στα βασικά λειτουργικά χαρακτηριστικά κάθε πράκτορα φορτίου, τα οποία καθορίζουν τα όρια ισχύος (Ενότητα 4.7), τις επιτρεπτές καταστάσεις λειτουργίας (Ενότητα 4.6.5) και τις συνθήκες άνεσης της οικίας. Επίσης, θεωρήθηκε ότι ο ρυθμός κατανάλωσης νερού τόσο για την αντλία όσο και για τον θερμοσίφωνα παραμένει σταθερός καθ' όλη τη διάρκεια της προσομοίωσης, ώστε οι αποκλίσεις μεταξύ ημερήσιου και ενδοημερήσιου προγραμματισμού να οφείλονται αποκλειστικά στην επίδραση της τιμολόγησης και όχι σε στοχαστικές μεταβολές της ζήτησης νερού. Έτσι, τα σενάρια που εξετάζονται επιτρέπουν την καθαρή αποτίμηση της επίδρασης της τιμολόγησης στη συνολική κατανάλωση, χωρίς να εισάγονται περιττές παραδοχές. Επιπλέον, για όλους τους πράκτορες φορτίου εφαρμόζεται διακριτοποίηση του χώρου καταστάσεων και δράσεων. Συγκεκριμένα, κάθε φυσική ποσότητα που ρυθμίζετια από το φορτίο μεταξύ αποδεκτών ορίων (θερμοκρασία, όγκος νερού και κατάσταση φόρτισης μπαταρίας) διακριτοποιείται σε 50 ίσα διαστήματα. Αντίστοιχα, οι δυνατές δράσεις, δηλαδή τα επίπεδα ισχύος των ελεγχόμενων φορτίων, διακριτοποιούνται σε 10 ίσα διαστήματα.

Φορτίο	**Παράμετρος**	**Τιμή**
Πράκτορας Μπαταρίας	Ονομαστική ισχύς φόρτισης/εκφόρτισης	2 kW
	Χωρητικότητα ενέργειας	5 kWh
	Βαθμός απόδοσης φόρτισης/εκ φόρτισης	0,95
	Ελάχιστη / μέγιστη κατάσταση φόρτισης	(10 / 100)%

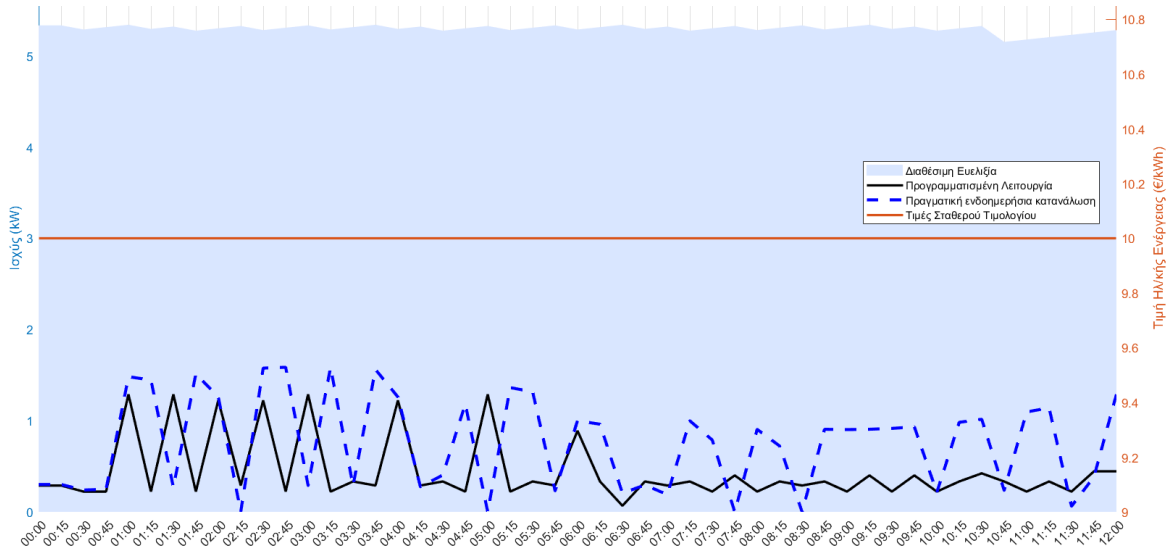
	Τελικός στόχος φόρτισης	50%
	Παράμετρος ανοχής	± 0,3
Πράκτορας Αντλίας	Ονομαστική ισχύς	0,6 kW
	Βαθμός απόδοσης	0,55
	Ελάχιστος / μέγιστος όγκος δεξαμενής	(0,2 / 1) m^3
	Τελικός στόχος όγκου	0,25 m^3
	Παράμετρος ανοχής	± 0,3 m^3
Πράκτορας Θερμοσίφωνα	Ονομαστική Ισχύς	3 kW
	Βαθμός απόδοσης	0,95
	Ελάχιστη / μέγιστη θερμοκρασία	(35/70) ^oC
	Τελική επιθυμητή θερμοκρασία	45 ^oC
	Παράμετρος ανοχής	± 5 ^oC

Πίνακας 6 Παράμετροι πρακτόρων φορτίων

Με βάση την παραπάνω παραμετροποίηση, εξετάζονται τα ακόλουθα σενάρια. Στο πρώτο σενάριο θεωρείται πολιτική σταθερής τιμολόγησης (μπλε τιμολόγιο), όπου η τιμή της ηλεκτρικής ενέργειας παραμένει σταθερή καθ' όλη τη διάρκεια της προσομοίωσης. Στο δεύτερο σενάριο θεωρείται πολιτική δυναμικής τιμολόγησης (πορτοκαλί τιμολόγιο), κατά την οποία οι τιμές διαφοροποιούνται ανά ώρα.

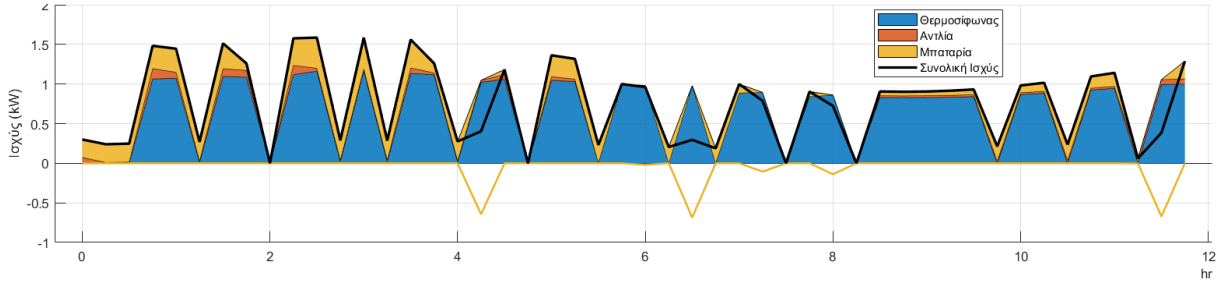
6.2.1 Σενάριο 1^o : Σταθερή τιμολόγηση

Το σενάριο αυτό χρησιμοποιείται ως σημείο αναφοράς, προκειμένου να αξιολογηθεί η συμπεριφορά των πρακτόρων σε συνθήκες ουδέτερου οικονομικού κινήτρου. Θεωρείται ότι η οικία περιλαμβάνει όλο το σύστημα του πολυπρακτορικού οικιακού μεσολαβητή, ωστόσο η τιμή της ηλεκτρικής ενέργειας παραμένει σταθερή καθ' όλη τη διάρκεια της προσομοίωσης, με αποτέλεσμα να μην υφίσταται οικονομικό σήμα που να ενθαρρύνει τη μετατόπιση της κατανάλωσης στον χρόνο. Επομένως, ο αρχικός προγραμματισμός του πράκτορα στοχεύει στη λειτουργία του φορτίου εντός των ορίων άνεσης του χρήστη με την ελάχιστη δυνατή κατανάλωση ενέργειας. Στην Εικόνα 22, παρουσιάζεται ο ημερήσιος προγραμματισμός της οικίας, όπως αυτός υπολογίζεται την προηγούμενη ημέρα, βάσει της σταθερής τιμολόγησης και των λειτουργικών περιορισμών των φορτίων. Στο ίδιο σχήμα απεικονίζεται επίσης η διαθέσιμη ευελιξία της οικίας, δηλαδή τα όρια εντός των οποίων θα μπορούσε να μεταβληθεί η συνολική κατανάλωση όπως και η πραγματική ενδοημερήσια κατανάλωση που προέκυψε λόγω εκτέλεσης εντολών ευελιξίας.



Εικόνα 22 Σταθερή τιμολόγηση: αποτελέσματα οικίας.

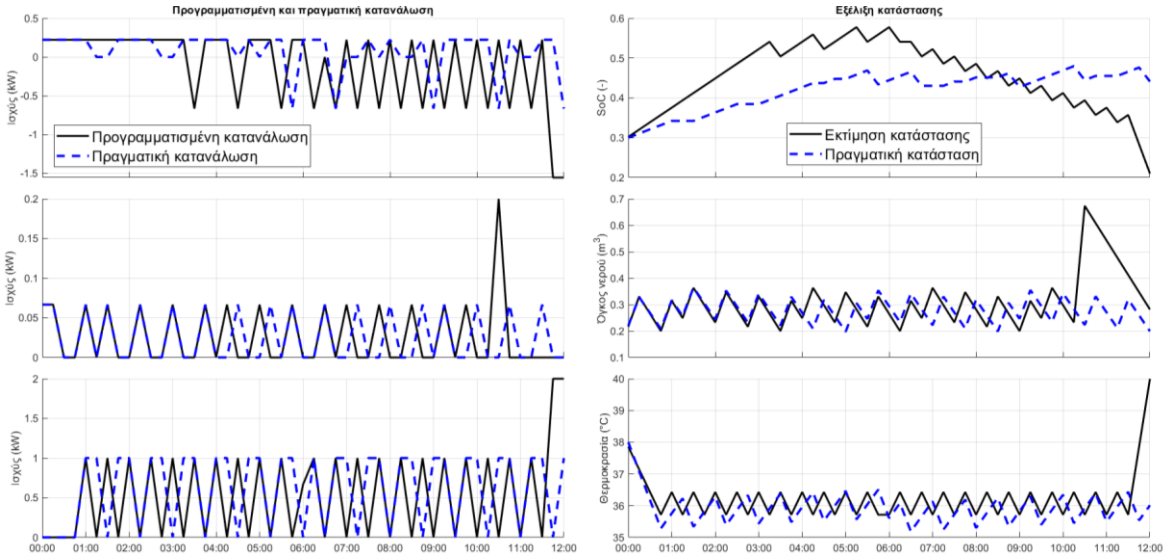
Θεωρούμε ότι, σε κάθε χρονικό διάστημα, κάποιος εξωτερικός φορέας π.χ. διαχειριστής δικτύου ή ΦοΣΕ, αποστέλλει μια εντολή για μεταβολή κατανάλωσης στο εύρος ±30%. Κατά τη διάρκεια της προσομοίωσης, ανά χρονικό βήμα των 15 λεπτών, ο πράκτορας οικίας κατανείμει την εντολή στους πράκτορες φορτίων (Εικόνα 23), οι πράκτορες φορτίων επικαιροποιούν τις εσωτερικές καταστάσεις των φορτίων που ελέγχουν (π.χ. κατάσταση φόρτισης μπαταρίας, θερμοκρασία θερμοσίφωνα, στάθμη δεξαμενής) και εκτελούν την εντολή από τον πράκτορα οικίας. Με βάση τις επικαιροποιημένες καταστάσεις, επιλύουν εκ νέου το πρόβλημα ΔΠ, υπολογίζοντας τον βέλτιστο προγραμματισμό για τον υπόλοιπο χρονικό ορίζοντα. Στο συγκεκριμένο σενάριο, η διαδικασία αυτή αποσκοπεί αποκλειστικά στη διατήρηση των λειτουργικών περιορισμών και των ορίων άνεσης, καθώς δεν υπάρχει οικονομικό κίνητρο για μεταβολή του προφίλ κατανάλωσης.



Εικόνα 23 Σταθερή τιμολόγηση: Κατανομή εντολής ευελιξίας.

Στην Εικόνα 24 παρουσιάζονται τα αποτελέσματα σε επίπεδο επιμέρους φορτίων. Στην αριστερή στήλη γραφημάτων απεικονίζεται η προγραμματισμένη κατανάλωση κάθε φορτίου σε σύγκριση με την πραγματική, ενδοημερήσια κατανάλωση που προκύπτει κατά την εκτέλεση. Στη δεξιά στήλη απεικονίζεται η

εκτιμώμενη εξέλιξη των καταστάσεων των φορτίων, όπως αυτή προκύπτει από τον ημερήσιο προγραμματισμό, σε σύγκριση με το πραγματικό προφίλ εξέλιξης των καταστάσεων κατά τη διάρκεια του εξεταζόμενου 12ώρου.



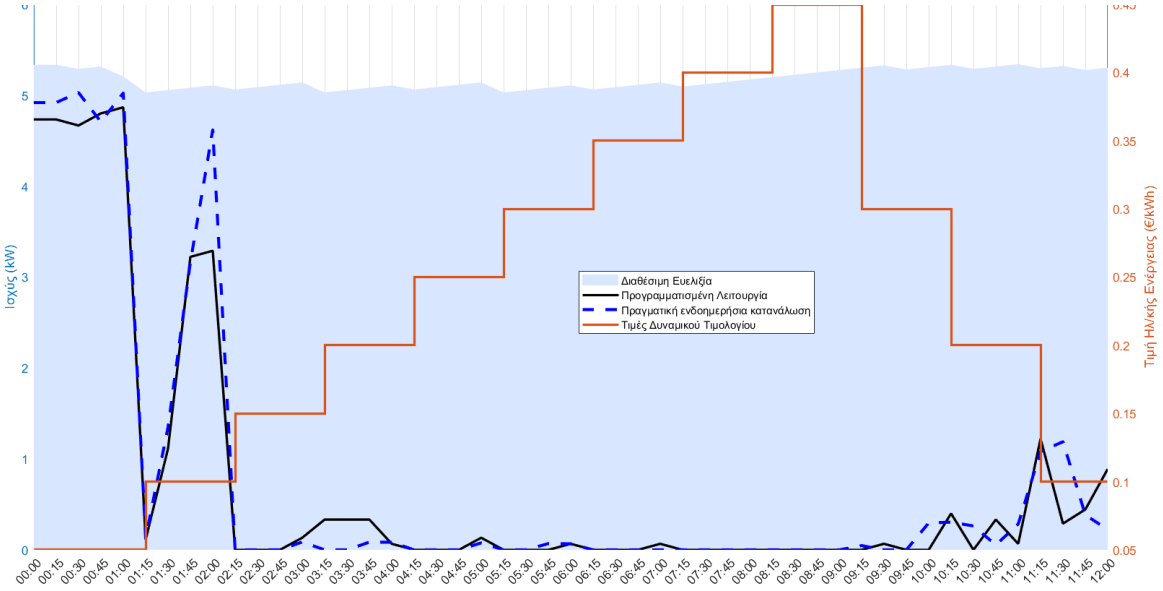
Εικόνα 24 Σταθερή τιμολόγηση: λειτουργία φορτίων.

Από τα αποτελέσματα παρατηρείται ότι όλα τα φορτία παραμένουν εντός των δηλωμένων ορίων λειτουργίας και άνεσης, επιβεβαιώνοντας την ορθότητα και τη σταθερότητα του αλγορίθμου ΔΠ. Οι αποκλίσεις που εντοπίζονται μεταξύ του ημερήσιου και του ενδοημερήσιου προγραμματισμού, δημιουργούνται καθώς η οικία έχει διαθέσει την ευελιξία της στο ΣΗΕ και οι πράκτορες φορτίων έχουν κληθεί να λειτουργήσουν εκτός του αρχικά προγραμματισμένου τους προφίλ, άρα απαιτήθηκε επαναπρογραμματισμός του προφίλ λειτουργίας των φορτίων, ώστε να προσφέρουν την απραίτητη ευελιξία.

6.2.2 Σενάριο 2ο : Δυναμική τιμολόγηση

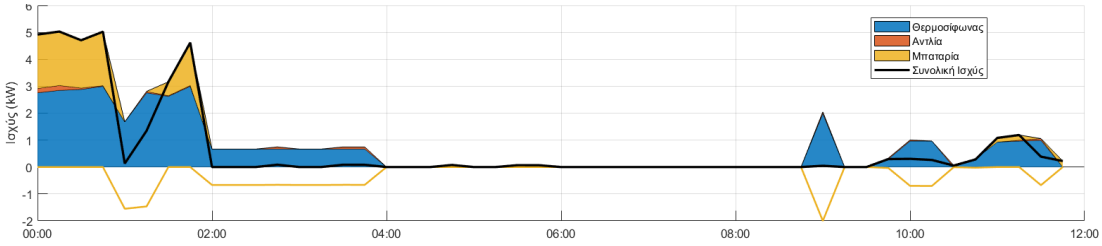
Αντίστοιχα με το πρώτο σενάριο, στο δεύτερο σενάριο εξετάζεται αποκλειστικά η λειτουργία της οικίας, λαμβάνοντας υπόψη την εμπλοκή του διαχειριστή δικτύου. Ωστόσο, εδώ εφαρμόζεται πολιτική δυναμικής τιμολόγησης, κατά την οποία η τιμή της ηλεκτρικής ενέργειας μεταβάλλεται ανά 1 ώρα εντός του 12ώρου προσομοίωσης. Η χρονική μεταβολή της τιμής λειτουργεί ως εξωτερικό οικονομικό σήμα για τους πράκτορες φορτίου. Ως αποτέλεσμα, οι πράκτορες δεν περιορίζονται πλέον μόνο στην ικανοποίηση των λειτουργικών περιορισμών και των ορίων άνεσης, αλλά επιδιώκουν επιπλέον τη μείωση του συνολικού ενεργειακού κόστους, προσαρμόζοντας χρονικά τη λειτουργία των φορτίων που ελέγχουν. Στην Εικόνα 25 παρουσιάζεται ο ημερήσιος

προγραμματισμός της οικίας με την εκτιμώμενη διαθέσιμη ευελιξία, την πραγματική ενδοημερήσια κατανάλωση, καθώς και το δυναμικό τιμολόγιο.



Εικόνα 25 Δυναμική τιμολόγηση: αποτελέσματα οικίας.

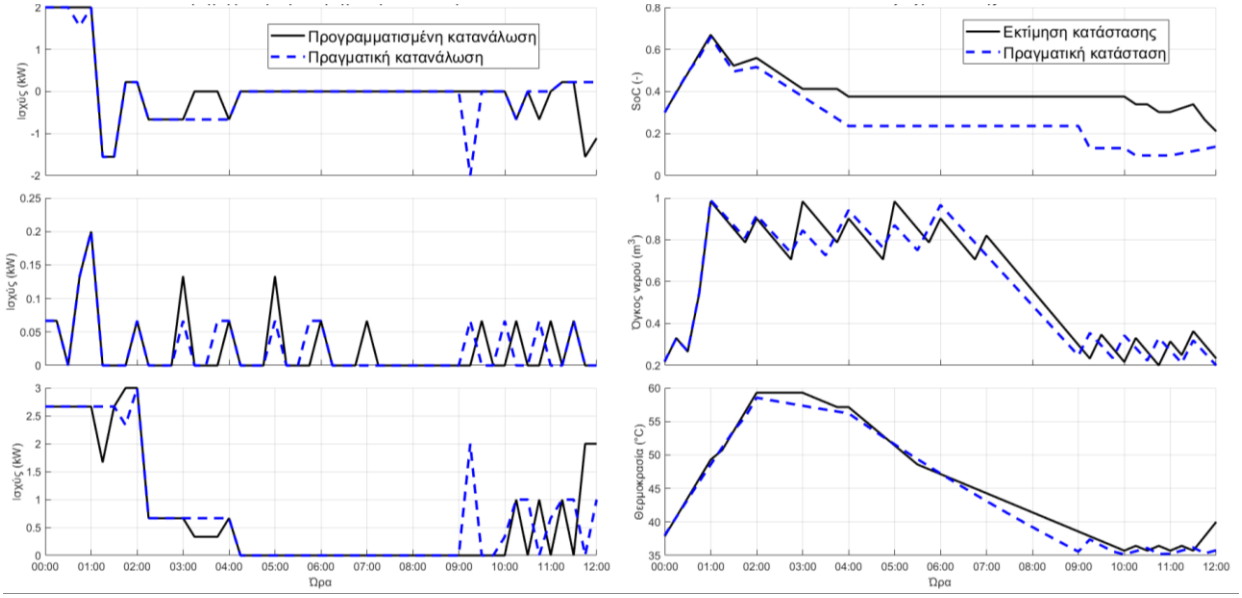
Ο πράκτορας οικίας ενημερώνει τους πράκτορες φορτίου και αυτοί επιλύουν τοπικά προβλήματα ΔΠ για να εκτιμήσουν την βέλτιστη λειτουργία των φορτίων τους. Εντός της ημέρας γίνονται αναπροσαρμογές λόγω εκτέλεσης εντολών ευελιξίας από τον διαχειριστή δικτύου ή ΦοΣΕ. Σε σύγκριση με το 1ο σενάριο, παρατηρείται σαφέστερη χρονική διαφοροποίηση της κατανάλωσης, με μετατόπιση φορτίων προς χρονικές περιόδους χαμηλότερου κόστους ενέργειας (Εικόνα 25). Η κατανομή της εντολής ευελιξίας, από τον πράκτορα οικίας στους επιμέρους πράκτορες φορτίων, φαίνεται παρακάτω. Για κάθε χρονικό διάστημα απεικονίζεται η συνολική ισχύς κατανάλωσης της οικίας που ζητήθηκε από τον διαχειριστή δικτύου, καθώς και ο τρόπος με τον οποίο αυτή κατανέμεται στα επιμέρους φορτία από τον πράκτορα οικίας.



Εικόνα 26 Δυναμική τιμολόγηση: Κατανομή εντολής ευελιξίας.

Στην Εικόνα 27 απεικονίζεται, η προγραμματισμένη κατανάλωση των επιμέρους φορτίων (αριστερά) σε σύγκριση με την πραγματική ενδοημερήσια κατανάλωση, ενώ δεξιά παρουσιάζεται η εκτιμώμενη εξέλιξη των καταστάσεων των

φορτίων βάσει του ημερήσιου προγραμματισμού, σε αντιπαραβολή με την πραγματική εξέλιξή τους κατά την εκτέλεση. Σε αντίθεση με το 1ο σενάριο, οι αποκλίσεις μεταξύ ημερήσιου και ενδοημερήσιου προγραμματισμού είναι σχεδόν αμελητέες, παρόλο που σε κάθε χρονικό βήμα το προφίλ λειτουργίας των φορτίων αναπροσαρμόζεται. Αυτό οφείλεται στο γεγονός ότι ο αλγόριθμος ΔΠ που υλοποιεί κάθε πράκτορας φορτίου προγραμματίζει την κατανάλωση κυρίως στις χρονικές περιόδους όπου η τιμή της ενέργειας είναι χαμηλή, περιορίζοντας έτσι τη δυνατότητα για ουσιαστικές μεταβολές κατά τον ενδοημερήσιο επαναπρογραμματισμό.



Εικόνα 27 Δυναμική τιμολόγηση: λειτουργία φορτίων.

Συνολικά, το σενάριο δυναμικής τιμολόγησης αναδεικνύει τη δυνατότητα του πολυπρακτορικού συστήματος να αξιοποιεί οικονομικά σήματα για την διαμόρφωση της κατανάλωσης. Όπως φαίνεται στην Εικόνα 25, μεταξύ της 3ης και 9ης ώρας οι πράκτορες φορτίου επιλέγουν να περιορίσουν την κατανάλωση τους λόγω υψηλών τιμών. Επίσης, αναδεικνύεται η αξία της μπαταρίας, διότι κατά την 1η ώρα όπου η ενέργεια ήταν φθηνή φρόντισε να φορτίσει, ενώ κατά την 2η και 9η ώρα, όπου τα λοιπά φορτία χρειάστηκε να κάνουν κατανάλωση, τροφοδοτήθηκε από την μπαταρία ώστε να ενισχυθεί η αυτοκατανάλωση της οικίας. Επιβεβαιώνεται η λειτουργικότητα του αλγόριθμου που υλοποιήθηκε στην παρούσα εργασία, όπως και η πρακτική χρησιμότητα του. Ο οικιακός μεσολαβητής μπορεί να μειώνει το κόστος για τον τελικό καταναλωτή, διασφαλίζοντας παράλληλα τη λειτουργική ευστάθεια των φορτίων και την τήρηση των ορίων άνεσης της οικίας.

6.3 Οικονομική κατανομή φορτίου με αξιοποίηση οικιακής ευελιξίας

Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα της Οικονομικής Κατανομής Φορτίου στο μικροδίκτυο, με χρήση της οικιακής ευελιξίας που προκύπτει από την λειτουργία του προτεινόμενου οικιακού μεσολαβητή. Στόχος είναι να εξεταστεί πώς η διαθέσιμη ευελιξία των οικιακών μη κρίσιμων φορτίων επηρεάζει τη λειτουργία του μικροδικτύου καθώς και το συνολικό κόστος του.

Δεδομένου ότι στην προηγούμενη ενότητα παρουσιάστηκε η ικανότητα του μεσολαβητή να μετατοπίζει το φορτίο ως ανταπόκριση σε οικονομικά σήματα, η παρούσα ενότητα εστιάζει στη συμβολή της ευελιξίας σε δύο διαφορετικές συνθήκες λειτουργίας του συστήματος από άποψη αυτοπαραγωγής. Συγκεκριμένα, εξετάζεται αφενός η περίπτωση αυξημένης παραγωγής από ανανεώσιμες πηγές ενέργειας και αφετέρου η περίπτωση περιορισμένης παραγωγής από φωτοβολταϊκά συστήματα, οι οποίες οδηγούν σε περίσσεια και έλλειμα αυτοπαραγωγής αντίστοιχα. Οι πράκτορες φορτίων επικαιροποιούν τον ενδοημερήσιο προγραμματισμό τους ανά 15 λεπτά, λαμβάνοντας υπόψη τις πραγματικές καταστάσεις των φορτίων τους. Τα αποτελέσματα αποστέλλονται στον πράκτορα οικίας, ο οποίος τα συγκεντρώνει και τα μεταφέρει στον ΔΜ ως ένα ενιαίο προφίλ. Το προφίλ αυτό περιγράφει τη διαθέσιμη μεταβολή ισχύος που μπορεί να προσφέρει η οικία στο σύστημα. Η παραμετροποίηση των πρακτόρων φορτίων διατηρείται ίδια με τα προηγούμενα σενάρια, ώστε οι παρατηρούμενες διαφοροποιήσεις στα αποτελέσματα να αποδίδονται στις συνθήκες αυτοπαραγωγής (φωτοβολταϊκά).

6.3.1 Σενάριο 1ο : Πλεόνασμα φωτοβολταϊκής παραγωγής

Το πρώτο σενάριο εξετάζει τη χρονική περίοδο από τις πρωινές έως τις μεσημεριανές ώρες, κατά την οποία επικρατεί έντονη ηλιοφάνεια και συνεπώς υψηλή διαθεσιμότητα Φ/Β παραγωγής και επιπλέον το μικροδίκτυο βρίσκεται σε νησιδοποιημένη λειτουργία. Την ίδια περίοδο, μεγάλο ποσοστό καταναλωτών απουσιάζει από την κατοικία (εργασία και σχολείο), με αποτέλεσμα η ανελαστική ζήτηση του μικροδικτύου να παραμένει σε σχετικά χαμηλά επίπεδα. Υπό αυτές τις συνθήκες, ο ΔΜ επιδιώκει τη μέγιστη δυνατή αξιοποίηση της τοπικής Φ/Β παραγωγής και την ελαχιστοποίηση της ανταλλαγής ισχύος με το κεντρικό δίκτυο. Η κατοικία, μέσω του πράκτορα οικίας, δηλώνει στο ΔΜ τόσο το προγραμματισμένο προφίλ λειτουργίας της όσο και τα επιτρεπτά όρια ελάχιστης και μέγιστης κατανάλωσης ανά

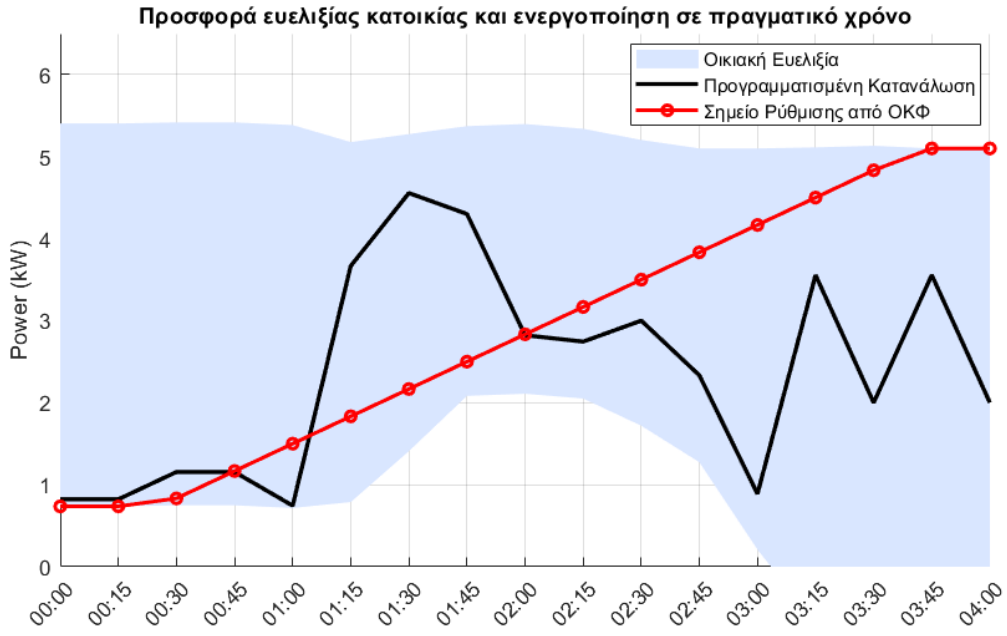
χρονικό διάστημα. Με τον τρόπο αυτό προσφέρει τη διαθέσιμη οικιακή ευελιξία προς αξιοποίηση, έναντι ενός μικρού κόστους ενεργοποίησης. Στόχος του σεναρίου είναι η αξιολόγηση της ικανότητας του μικροδικτύου να αξιοποιεί ενεργά την οικιακή ευελιξία, ώστε να αυξηθεί η αυτοκατανάλωση της ανανεώσιμης ενέργειας και να ενισχυθεί η ενεργειακή αυτάρκεια του μικροδικτύου σε σχέση με το κεντρικό δίκτυο. Στον Πίνακας 7 παρουσιάζεται η παραμετροποίηση του μικροδικτύου για το συγκεκριμένο σενάριο. Η διαθέσιμη ισχύς από Φ/Β επιλέγεται να κυμαίνεται στο εύρος 60–160% της συνολικής κατανάλωσης της οικίας, ώστε να αναδειχθεί με σαφήνεια ο ρόλος της ευελιξίας στη διαμόρφωση της καμπύλης απορρόφησης της Φ/Β παραγωγής.

Πίνακας 7 Παραμετροποίηση μικροδικτύου

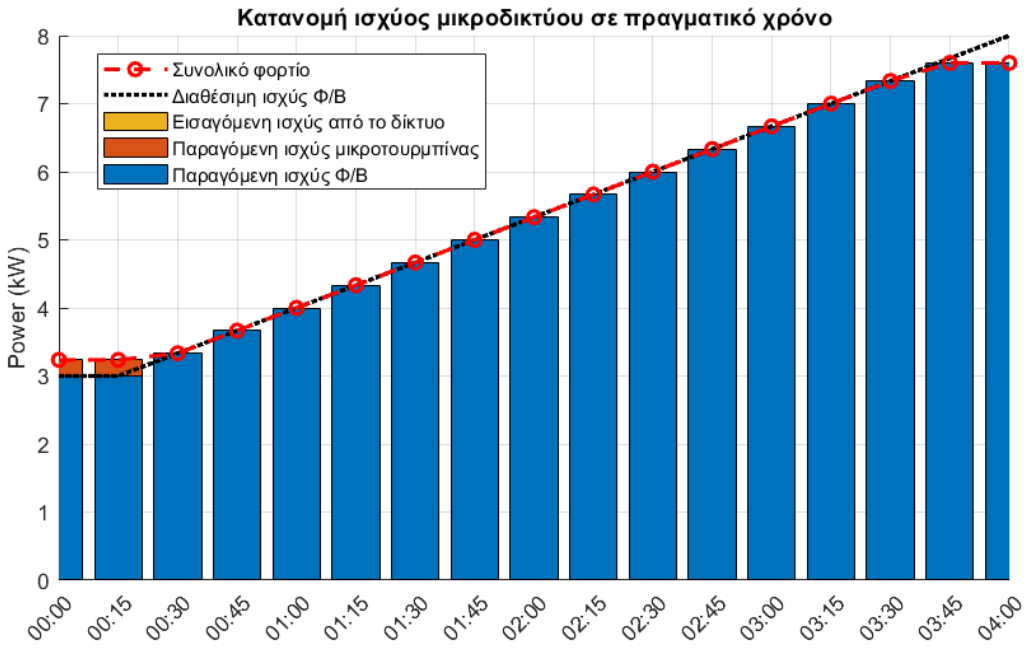
Παράμετρος		**Σύμβολο**	**Τιμή**	**Μονάδα μέτρησης**
Διάρκεια προσομοίωσης		T	4	ώρες
Χρονικό βήμα		Δτ	0,25	ώρες
Ανελαστική ζήτηση		L	2,5	kW
Διαθέσιμη ισχύς Φ/Β (μεταβάλλεται ανά 15 λεπτά)		P_t^{PV}	3 έως 8	kW
Μικροτουρμπίνα	Τετραγωνικός συντελεστής κόστους	a^{MT}	10^{-2}	-
	Γραμμικός συντελεστής κόστους	b^{MT}	0.1	€/kW
	Σταθερός όρος κόστους	c^{MT}	0.15	€
	Ελάχιστη ισχύς	P_{min}^{MT}	0	kW
	Μέγιστη ισχύς	P_{max}^{MT}	4	kW
Μέγιστη εισαγόμενη ισχύς από κεντρικό δίκτυο		P_{max}^{Line}	3	kW
Κόστος ενεργοποίησης ευελιξίας		c_t^{flex}	0.03	€/kW
Τιμή αγοράς/πώλησης από/προς δίκτυο		c_t^{grid}	0.25	€/kW

Κατά τη διάρκεια της αιχμής της Φ/Β παραγωγής, ο ΔΜ επιλύει το πρόβλημα κατανομής φορτίου και επιλέγει να ενεργοποιήσει την ανοδική ευελιξία της κατοικίας εντός των δηλωμένων ορίων, όπως φαίνεται και στα αποτελέσματα των προσομοιώσεων (Εικόνα 28). Έτσι, επιτυγχάνεται η απορρόφηση του πλεονάσματος Φ/Β ισχύος μέσω της ενεργοποίησης των μη κρίσιμων φορτίων της κατοικίας, δηλαδή

προθέρμανση του θερμοσίφωνα, γέμισμα της δεξαμενής νερού και φόρτιση της μπαταρίας. Το αποτέλεσμα αυτό αποτυπώνεται στο περιεχόμενο των Εικόνων 28 και 29. Στην Εικόνα 28 παρουσιάζεται ο αρχικός ημερήσιος προγραμματισμός της οικίας, ο οποίοςλαμβάνοντας υπόψη την προκαθορισμένη διακύμανση τιμών της δυναμικής τιμολόγησης με στόχο την οικονομικότερη λειτουργία για το χρήστη. Ανά 15 λεπτά ο ΔΜ επιλύει το πρόβλημα ΟΚΦ και αποστέλλει στην οικία το προτεινόμενο σημείο λειτουργίας, το οποίο όμως σέβεται τα όρια που έχει ορίσει ο οικιακός πράκτορας. Στο τέλος του εξεταζόμενου ορίζοντα φαίνεται το τελικό προφίλ λειτουργίας, συνολικά της οικίας, έναντι του αρχικού ημερήσιου προγραμματισμού που είχαν κάνει οι επιμέρους πράκτορες φορτίων.



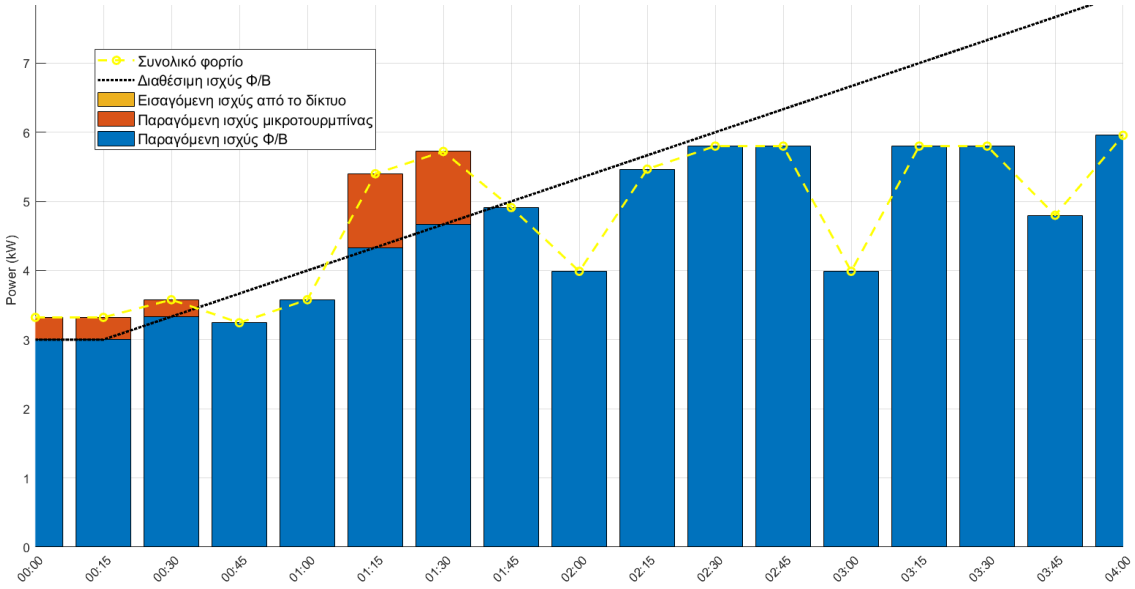
Εικόνα 28 Σενάριο 1ο: Προσφορά ευελιξίας κατοικίας και ενεργοποίηση σε πραγματικό χρόνο



Εικόνα 29 Σενάριο 1ο: Κατανομή ισχύος μικροδικτύου σε πραγματικό χρόνο

Στην Εικόνα 29 παρουσιάζονται τα αποτελέσματα της ΟΚΦ. Παρατηρείται ότι, η καμπύλη διαθέσιμης Φ/Β ισχύος ταυτίζεται με την τελική παραγόμενη ισχύ και δεν πραγματοποιείται περικοπή της Φ/Β παραγωγής. Παράλληλα, το μικροδίκτυο καλύπτει τις ενεργειακές του ανάγκες σχεδόν αποκλειστικά από τη Φ/Β παραγωγή, ενεργοποιώντας σε κάθε χρονικό βήμα την κατάλληλη ποσότητα ανοδικής ευελιξίας από τα μη κρίσιμα φορτία. Με τον τρόπο αυτό, η συνολική ζήτηση του μικροδικτύου προσαρμόζεται ώστε να ισούται με τη διαθέσιμη Φ/Β παραγωγή. Στο 1ο σενάριο διαπιστώνεται ότι η αξιοποίηση της ευελιξίας στην πλευρά της ζήτησης είναι ιδιαίτερα αποτελεσματική, επιτρέποντας την πλήρη απορρόφηση της τοπικής Φ/Β παραγωγής και τη σημαντική μείωση της εξάρτησης από το κεντρικό δίκτυο.

Για την αποτίμηση της συμβολής της ανοδικής ευελιξίας της κατοικίας, το ίδιο σενάριο προσομοιώθηκε θεωρώντας ότι η ευελιξία των μη κρίσιμων οικιακών φορτίων δεν είναι διαθέσιμη. Τα αποτελέσματα της κατανομής ισχύος του μικροδικτύου χωρίς ευελιξία παρουσιάζονται στην ακόλουθη εικόνα:



Εικόνα 30 Σενάριο 1ο: Κατανομή ισχύος μικροδικτύου σε πραγματικό χρόνο χωρίς ευελιξία.

Στον Πίνακα 8 συγκρίνονται οι δύο εκδοχές του 1ου σεναρίου, δηλαδή η λειτουργία με και χωρίς ενεργοποίηση οικιακής ευελιξίας. Επειδή τα δεδομένα της προσομοίωσης είναι συνθετικά και δεν αντιστοιχούν σε πραγματικές τιμές, η αξιολόγηση βασίζεται κυρίως σε ποσοστιαίες μεταβολές και όχι στις απόλυτες τιμές, για το εξεταζόμενο χρονικό διάστημα των τεσσάρων ωρών.

Πίνακας 8 Σενάριο 1ο: Σύγκριση με και χωρίς οικιακή ευελιξία.

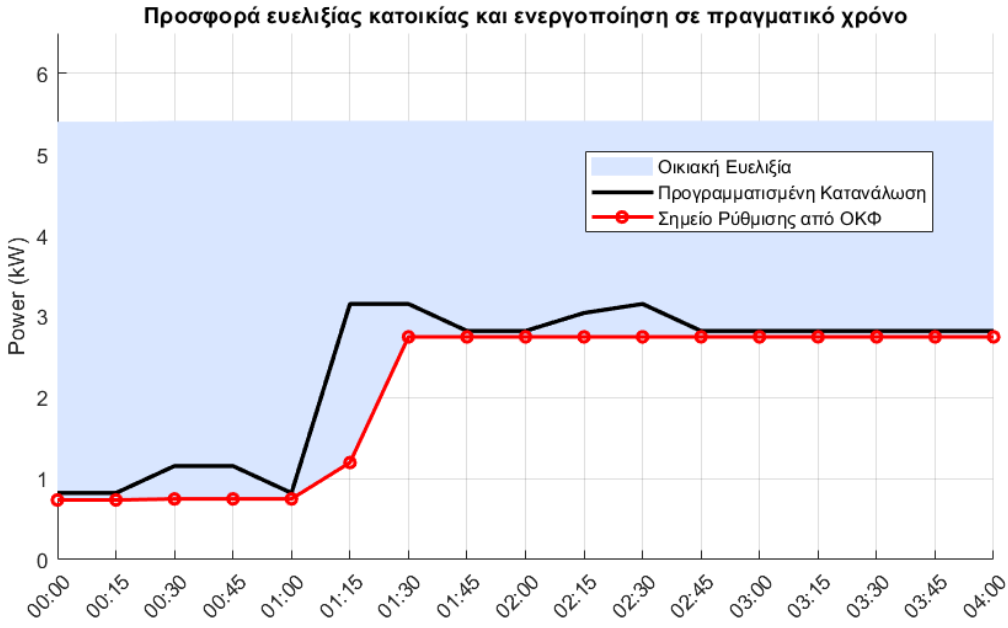
Μέγεθος (για 4 ώρες)	**Χωρίς ευελιξία**	**Με ευελιξία**	**Ποσοστιαία μεταβολή**
Συνολικό κόστος παραγωγής μικροτουρμπίνας (€)	2,81	0,57	-79,7%
Συνολική ενέργεια από μικροτουρμπίνα (kWh)	0,67	0,06	-91,0%
Συνολικό κόστος μικροδικτύου (€)	2,81	0,79	- 71.89%

Η ενεργοποίηση της ανοδικής οικιακής ευελιξίας οδηγεί σε αισθητή βελτίωση της λειτουργίας του μικροδικτύου. Η συνολική παραγωγή ενέργειας της μικροτουρμπίνας μειώνεται κατά 91%, γεγονός που συνεπάγεται και αντίστοιχη

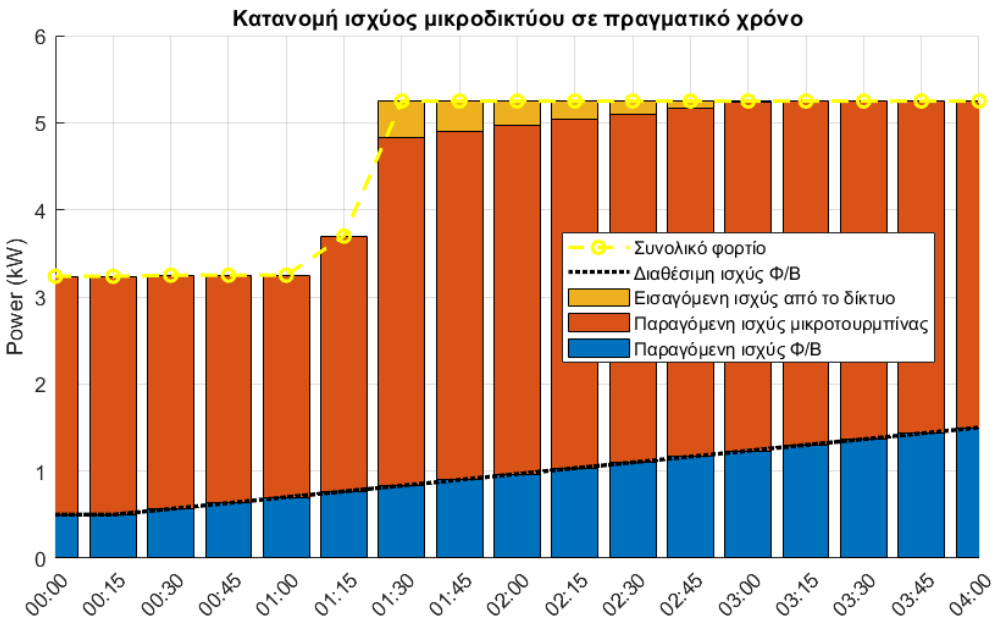
μείωση του κόστους λειτουργίας της κατά 79,7%. Η σημαντική αυτή μείωση της συμβατικής παραγωγής οφείλεται στη δυνατότητα αύξησης της οικιακής κατανάλωσης όταν υπάρχει διαθέσιμη Φ/Β παραγωγή. Με τον τρόπο αυτό, η πλεονάζουσα ενέργεια απορροφάται τοπικά, περιορίζοντας την ανάγκη ενεργοποίησης της μικροτουρμπίνας. Ως αποτέλεσμα, το συνολικό κόστος λειτουργίας του μικροδικτύου μειώνεται κατά 71,9% σε σχέση με την περίπτωση χωρίς ευελιξία.

6.3.2 Σενάριο 2ο : Περιορισμένη παραγωγή Φ/Β

Το δεύτερο σενάριο εξετάζει τη λειτουργία του μικροδικτύου σε συνθήκες περιορισμένης Φ/Β παραγωγής, όπως αυτές που προκύπτουν λόγω έντονης συννεφιάς ή κατά τις απογευματινές ώρες. Σε αντίθεση με το σενάριο πλεονάσματος Φ/Β παραγωγής, η διαθέσιμη ανανεώσιμη παραγωγή είναι χαμηλή και δεν επαρκεί για την κάλυψη της συνολικής ζήτησης του μικροδικτύου. Οι υπόλοιπες παράμετροι του προβλήματος διατηρούνται ίδιες με αυτές που παρουσιάστηκαν στον Πίνακας 7. Σε αυτό το σενάριο, ο ΔΜ στοχεύει στη μείωση του συνολικού φορτίου, ώστε να περιορίσει τη λειτουργία των κοστοβόρων συμβατικών μονάδων (μικροτουρμπίνα) και την εισαγωγή ενέργειας από το κεντρικό δίκτυο. Αντίστοιχα με το 1ο σενάριο, στις εικόνες 31 και 32 παρουσιάζονται τα αποτελέσματα της προσομοίωσης του σεναρίου με περιορισμένη παραγωγή.



Εικόνα 31 Σενάριο 2ο: Προσφορά ευελιξίας κατοικίας και ενεργοποίηση σε πραγματικό χρόνο.



Εικόνα 32 Σενάριο 2ο: Κατανομή ισχύος μικροδικτύου σε πραγματικό χρόνο.

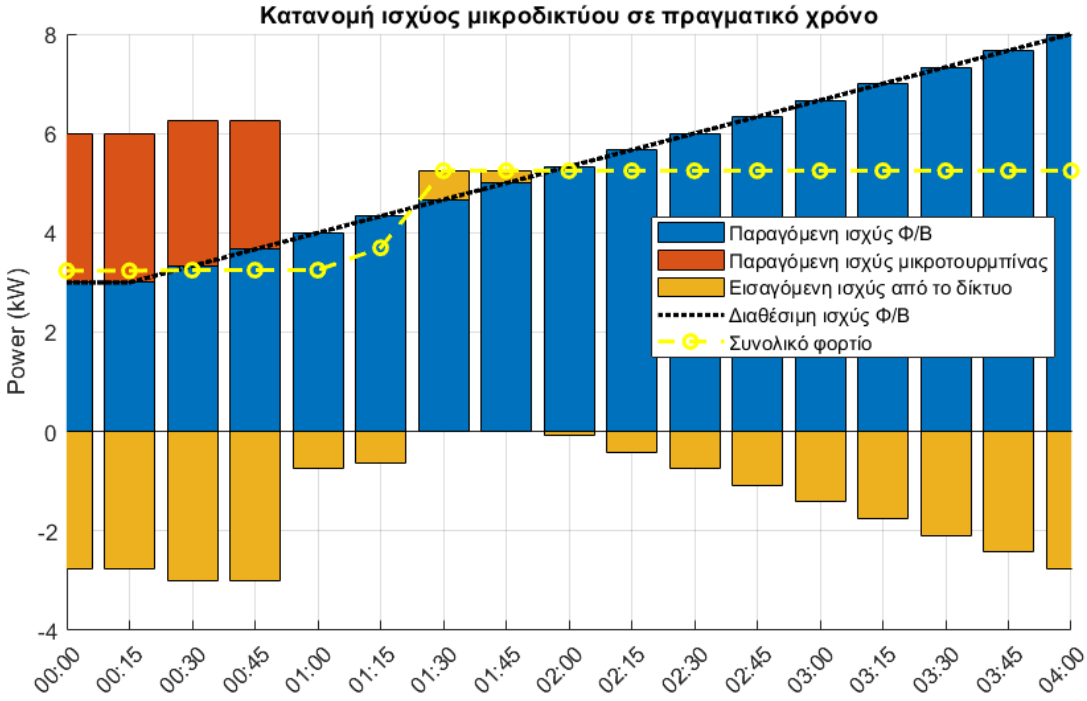
Συγκεκριμένα, στην Εικόνα 31 απεικονίζεται ο αρχικός ημερήσιος προγραμματισμός λειτουργίας της οικίας και η ενεργοποίησή της ευελιξίας της σε πραγματικό χρόνο. Κατά τη διάρκεια της προσομοίωσης, ο ΔΜ αξιοποιεί την καθοδική ευελιξία της κατοικίας, αποστέλλοντας μειωμένο σημείο κατανάλωσης εντός των δηλωμένων ορίων, καθιστώντας την οικία ως έναν πόρο καθοδικής εφεδρείας. Στην Εικόνα 32, όπου παρουσιάζονται τα αποτελέσματα της ΟΚΦ, παρατηρείται ότι, λόγω της χαμηλής Φ/Β παραγωγής, το μικροδίκτυο δεν μπορεί να καλύψει πλήρως τις ανάγκες του από ανανεώσιμες πηγές. Ωστόσο, μέσω της αξιοποίησης της καθοδικής ευελιξίας των μη κρίσιμων φορτίων, η συνολική ζήτηση μειώνεται σημαντικά, περιορίζοντας την ανάγκη εισαγωγής ενέργειας από το κεντρικό δίκτυο. Στο 2ο σενάριο αναδεικνύεται ο ρόλος της οικιακής ευελιξίας ως μηχανισμού καθοδικής ευελιξίας. Η ικανότητα μείωσης της ζήτησης επιτρέπει στο μικροδίκτυο να λειτουργεί πιο αποδοτικά υπό δυσμενείς συνθήκες αυτοπαραγωγής και υψηλής ζήτησης, διατηρώντας σε υψηλό βαθμό την ενεργειακή του αυτάρκεια.

6.3.3 Σενάριο 3ο : Μεγιστοποίηση εσόδων Μικροδικτύου

Το τρίτο σενάριο εξετάζει τη λειτουργία του μικροδικτύου σε συνθήκες υψηλής τιμής ηλεκτρικής ενέργειας από και προς το κεντρικό δίκτυο. Η αυξημένη τιμή αγοράς/πώλησης δημιουργεί ισχυρό οικονομικό κίνητρο για την εξαγωγή ισχύος, μετατοπίζοντας τον στόχο της λειτουργίας από την απλή κάλυψη της ζήτησης με

ελάχιστο κόστος, προς τη μεγιστοποίηση των εσόδων του μικροδικτύου. Στα προηγούμενα σενάρια, οι παράμετροι του μικροδικτύου είχαν επιλεγεί ώστε η μικροτουρμπίνα να αποτελεί οικονομικότερο μέσο κάλυψης της ζήτησης απ' ότι η αγορά ισχύος από το κεντρικό δίκτυο, ενώ η τιμή αγοράς/πώλησης παρέμενε σταθερή καθ' όλη τη διάρκεια της προσομοίωσης. Αντιθέτως, στο παρόν σενάριο επιλέγεται η τιμή αγοράς/πώλησης να μεταβάλλεται χρονικά, ακολουθώντας ένα δυναμικό προφίλ τιμολόγησης (π.χ. βάσει της εκκαθάρισης της κεντρικής χονδρεμπορικής αγοράς επόμενης ημέρας), ώστε να διερευνηθεί η συμπεριφορά του συστήματος όταν η τιμή είναι άλλοτε υψηλότερη και άλλοτε χαμηλότερη από το οριακό κόστος της μικροτουρμπίνας. Παράλληλα, επιλέγεται αυξημένη Φ/Β παραγωγή, ώστε να εξεταστούν οι επιλογές του ΔΜ ως προς τη βέλτιστη αξιοποίησή της. Σε αυτό το σενάριο ο ΔΜ επιδιώκει να αυξήσει την εξαγόμενη ισχύ κατά τις χρονικές περιόδους υψηλής τιμής, περιορίζοντας τη τοπική κατανάλωση μέσω ενεργοποίησης καθοδικής ευελιξίας που παίρνει από τα μη κρίσιμα φορτία. Στην Εικόνα 33 παρουσιάζεται η προσφορά ευελιξίας της οικίας και η ενεργοποίησή της σε πραγματικό χρόνο, όπου φαίνεται πως απαιτήθηκε από την οικία να λειτουργεί στο κατώτερο επιτρεπτό επίπεδο κατανάλωσης σε κάθε διάστημα, χωρίς όμως να παραβιάζονται τα όρια άνεσης για το χρήστη.

Εικόνα 33 Σενάριο 3ο: Προσφορά ευελιξίας κατοικίας και ενεργοποίηση σε πραγματικό χρόνο.

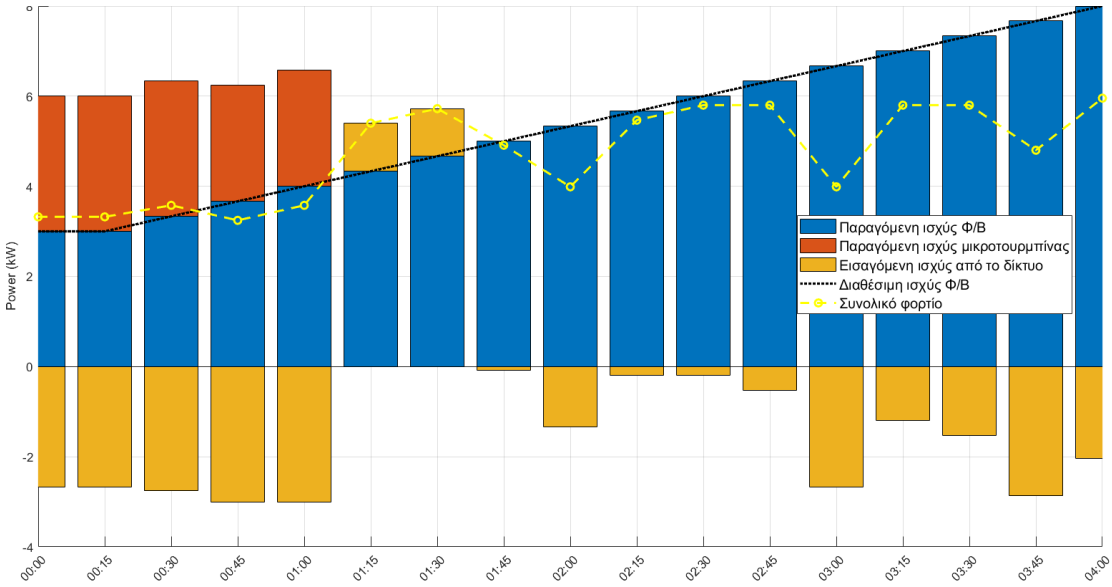


Εικόνα 34 Σενάριο 3ο: Κατανομή ισχύος μικροδικτύου σε πραγματικό χρόνο.

Στην Εικόνα 34 απεικονίζονται τα αποτελέσματα της ΟΚΦ και η κατανομή ισχύος του μικροδικτύου. Το προφίλ της τιμής αγοράς/πώλησης ακολουθεί την διαθέσιμη Φ/Β παραγωγή. Την 1η ώρα όπου φαίνεται να μην υπάρχει ηλιοφάνεια, η τιμή αγοράς/πώλησης είναι ακριβή και όσο αυξάνεται η ηλιοφάνεια τόσο φθίνει η τιμή της. Την 1η ώρα φαίνεται ότι λόγω ακριβής τιμής, ο ΔΜ επιλέγει να περιορίσει την τοπική κατανάλωση και να εξάγει όλη την διαθέσιμη παραγωγή ισχύος στο κεντρικό δίκτυο, ενεργοποιώντας ακόμα και την μικροτουρμπίνα στην ονομαστική της ισχύ, αφού το κόστος παραγωγής της είναι μικρότερο από την τιμή αγοράς ηλεκτρικής ενέργειας από το δίκτυο, ώστε να μεγιστοποιήσει το κέρδος του. Κατά την 2η ώρα, όπου η τιμή έχει μειωθεί, παρατηρείται ότι το 1ο μισάωρο εξάγει την πλεονάζουσα Φ/Β ισχύ στο δίκτυο ενώ κατά το 2ο μισάωρο εισάγει ισχύ από το κεντρικό δίκτυο για να καλύψει την συνολική του ζήτηση. Από την αρχή της 3ης ώρας έως το τέλος της 4ης, παρά την ύπαρξη αυξημένης Φ/Β παραγωγής, ο ΔΜ συνεχίζει να διατηρεί μειωμένη τη τοπική κατανάλωση, εκμεταλλευόμενος τη δυνατότητα εξαγωγής της πλεονάζουσας ενέργειας. Η επιλογή αυτή επιβεβαιώνει ότι, πέραν από μηχανισμό διαχείρισης ελλείμματος παραγωγής, η οικιακή ευελιξία μπορεί να αποτελέσει και εργαλείο οικονομικής στρατηγικής. Συνολικά, στο 3ο σενάριο αναδεικνύεται ο ρόλος της ευελιξίας ως μέσου ενεργούς συμμετοχής του μικροδικτύου στην αγορά ενέργειας. Η

προσαρμογή της ζήτησης επιτρέπει τη βέλτιστη εκμετάλλευση των διακυμάνσεων της τιμής ενέργειας, μεγιστοποιώντας το οικονομικό όφελος του μικροδικτύου.

Για την ποσοτικοποίηση της αξίας της ευελιξίας στη ζήτηση, το ίδιο σενάριο προσομοιώθηκε εκ νέου υπό την υπόθεση ότι η ευελιξία των μη κρίσιμων οικιακών φορτίων δεν είναι διαθέσιμη. Τα αποτελέσματα της προσομοίωσης αυτής παρουσιάζονται στην ακόλουθη εικόνα:



Εικόνα 35 Σενάριο 3ο: Κατανομή ισχύος μικροδικτύου σε πραγματικό χρόνο χωρίς ευελιξία.

Με βάση την παραμετροποίηση του μικροδικτύου (βλ. Πίνακα Πίνακας 7), στον παρακάτω πίνακα συγκρίνονται οι δύο εκδοχές του 3ου σεναρίου, δηλαδή η λειτουργία με και χωρίς ενεργοποίηση οικιακής ευελιξίας. Δεδομένου ότι τα δεδομένα που χρησιμοποιήθηκαν για τις προσομοιώσεις, είναι συνθετικά και δεν αντιστοιχούν σε πραγματικές τιμές, η ανάλυση επικεντρώνεται κυρίως στη σχετική μεταβολή των μεγεθών (ποσοστιαία μεταβολή) όπως έγινε και στο 1ο Σενάριο προσομοίωσης.

Πίνακας 9 Σενάριο 3ο: Σύγκριση με και χωρίς οικιακή ευελιξία.

Μέγεθος (για 4 ώρες)	**Χωρίς ευελιξία**	**Με ευελιξία**	**Ποσοστιαία μεταβολή**
Συνολικό κόστος παραγωγής μικροτουρμπίνας (€)	5,26	3,99	-24,1%
Εξαγόμενη ενέργεια (kWh)	5,73	6,03	+5,0%

Εισαγόμενη ενέργεια (kWh)	0,53	0,21	-60,4%
Συνολικό κόστος μικροδικτύου (€)	-1,74	-2,31	-32,8%

Από τα αποτελέσματα προκύπτει ότι η ενεργοποίηση της καθοδικής οικιακής ευελιξίας οδηγεί σε ουσιαστική βελτίωση της οικονομικής απόδοσης του μικροδικτύου, με επιπλέον καθαρό όφελος της τάξης του 32,8%. Η βελτίωση αυτή οφείλεται κυρίως στη σημαντική μείωση της παραγωγής από τη μικροτουρμπίνα (κατά 24,1%), καθώς η δυνατότητα προσαρμογής της ζήτησης επέτρεψε στον ΔΜ να περιορίσει τη λειτουργία της συμβατικής μονάδας. Παρατηρείται επίσης ότι, με ενεργοποίηση της ευελιξίας, η εξαγόμενη ενέργεια αυξάνεται κατά 5%, ενώ η εισαγόμενη ενέργεια μειώνεται σημαντικά, κατά 60,4%. Το αποτέλεσμα αυτό υποδηλώνει ότι η ευελιξία δεν λειτουργεί αποκλειστικά ως μέσο μεγιστοποίησης των εξαγωγών, αλλά κυρίως ως εργαλείο περιορισμού της εξάρτησης τόσο από την τοπική συμβατική παραγωγή όσο και από εισαγωγές ισχύος από το κεντρικό δίκτυο. Αν και τα αποτελέσματα αφορούν μόνο χρονικό ορίζοντα τεσσάρων ωρών, καταδεικνύουν σαφώς ότι η οικιακή ευελιξία μπορεί να ενισχύσει την οικονομική αποδοτικότητα του μικροδικτύου, μειώνοντας τη χρήση ακριβών πόρων και βελτιστοποιώντας τη συμμετοχή του στην αγορά ηλεκτρικής ενέργειας.

6.4 Συμπεράσματα

Από τα αποτελέσματα των προσομοιώσεων προκύπτει ότι το προτεινόμενο πολυπρακτορικό σύστημα διαχείρισης οικιακής ευελιξίας λειτουργεί αποτελεσματικά. Σε όλα τα εξεταζόμενα σενάρια τηρούνται οι λειτουργικοί περιορισμοί και τα όρια άνεσης των φορτίων και αυτό επιτυγχάνεται με το μικρότερο δυνατό κόστος. Σε επίπεδο οικίας διαπιστώθηκε ότι, υπό δυναμική τιμολόγηση, οι πράκτορες φορτίων δρουν αυτόνομα με στόχο τη μείωση του κόστους για τον χρήστη, μετατοπίζοντας χρονικά την κατανάλωση προς περιόδους χαμηλότερης τιμής ενέργειας.

Σε επίπεδο μικροδικτύου, η οικιακή ευελιξία αποδείχθηκε ιδιαίτερα χρήσιμη τόσο σε συνθήκες πλεονάσματος όσο και έλλειψης οικονομικής αυτοπαραγωγής (όπως ΑΠΕ). Στην περίπτωση υψηλής παραγωγής, συνέβαλε στην πλήρη απορρόφηση της διαθέσιμης ανανεώσιμης ενέργειας μετατοπίζοντας κατάλληλα την ζήτηση των μη κρίσιμων φορτίων. Αντίστοιχα, σε συνθήκες περιορισμένης παραγωγής, λειτούργησε

ως μηχανισμός καθοδικής εφεδρείας, περιορίζοντας τη ζήτηση και μειώνοντας την ανάγκη εισαγωγής ενέργειας ή λειτουργίας της μικροτουρμπίνας. Τέλος, στο σενάριο μεγιστοποίησης εσόδων αναδείχθηκε η δυνατότητα αξιοποίησης της ευελιξίας όχι μόνο ως τεχνικού μηχανισμού εξισορρόπησης, αλλά και ως οικονομικού εργαλείου στρατηγικής λειτουργίας. Η προσαρμογή της ζήτησης επέτρεψε στο μικροδίκτυο να ανταποκρίνεται στις διακυμάνσεις της τιμής ενέργειας, ενισχύοντας τη συμμετοχή του στην αγορά και βελτιώνοντας τη συνολική οικονομική του απόδοση. Συνολικά, τα αποτελέσματα επιβεβαιώνουν ότι ο οικιακός μεσολαβητής μπορεί να αποτελέσει αποτελεσματικό μηχανισμό διαχείρισης ευελιξίας, συμβάλλοντας τόσο στη μείωση του κόστους για τον τελικό καταναλωτή όσο και στη βελτίωση της λειτουργίας του ΣΗΕ που συμμετέχει, αυξάνοντας το μερίδιο και την αξία της παραγόμενης από ΑΠΕ ενέργειας.

7 Συμπεράσματα Διπλωματικής Εργασίας

7.1 Ανακεφαλαίωση

Τα σύγχρονα ΣΗΕ μετασχηματίζονται με ταχύ ρυθμό, λόγω συνεχόμενης ενσωμάτωσης διεσπαρμένων πηγών, κυρίως ΑΠΕ, και του εξηλεκτρισμού των καταναλώσεων. Οι απαιτήσεις για αποδοτική διαχείριση των δικτύων αυξάνονται, ενώ η πλευρά της ζήτησης παραμένει σε μεγάλο βαθμό ανεκμετάλλευτη, παρά το γεγονός ότι διαθέτει σημαντικές δυνατότητες συνεισφοράς στην αποδοτική λειτουργία των τους. Η ενεργός συμμετοχή των καταναλωτών μπορεί να συμβάλει στην αποσυμφόρηση των δικτύων, στη βελτίωση της αξιοποίησης των ΑΠΕ και στον περιορισμό της ανάγκης για δαπανηρές επενδύσεις σε υποδομές των δικτύων ή μεγάλης κλίμακας αποθήκευση ενέργειας. Η επίλυση του προβλήματος της αποτελεσματικής ενσωμάτωσης της ευελιξίας που προσφέρεται από οικιακούς καταναλωτές είναι επομένως κρίσιμη τόσο τεχνικά όσο και οικονομικά.

Η παρούσα διπλωματική εργασία ανέπτυξε ένα ολοκληρωμένο σύστημα οικιακής μεσολάβησης για τη διαχείριση μη κρίσιμων φορτίων. Η προτεινόμενη λύση βασίστηκε σε πολυπρακτορική αρχιτεκτονική, επιτρέποντας σαφή διαχωρισμό ρόλων, επεκτασιμότητα και αποκεντρωμένη λήψη αποφάσεων. Στη λογική της επεκτασιμότητας, παρουσιάστηκαν γραμμικά μοντέλα για 2 βασικές κατηγορίες φορτίων, τα θερμοστατικά ελεγχόμενα φορτία και η αποθήκευση ενέργειας. Τα μοντέλα αυτά στη συνέχεια εξειδικεύτηκαν σε 3 κοινά οικιακά φορτία που μπορούν να παρέχουν σημαντική ευελιξία. Αναπτύχθηκε ένας γενικευμένος αλγόριθμος δυναμικού προγραμματισμού για τη βελτιστοποίηση της λειτουργίας του κάθε φορτίου, τον οποίο ενσωματώνει ο αντίστοιχος πράκτορας φορτίου. Οι πράκτορες φορτίου σχεδιάστηκαν με δυνατότητες ημερήσιου προγραμματισμού και ενδοημερήσιου προγραμματισμού της λειτουργίας του φορτίου και διαμόρφωση προφίλ ευελιξίας. Αναπτύχθηκε επίσης ο πράκτορας οικίας με ρόλο συντονιστή μεταξύ μπαταρίας και φορτίων. Ο ρόλος του περιλαμβάνει επίσης την συγκέντρωση των αποτελεσμάτων των επιμέρους φορτίων και την έκθεση ενός ενιαίου προφίλ ευελιξίας προς τον διαχειριστή δικτύου και την κατανομή των εντολών ενεργοποίησης ευελιξίας στα επιμέρους φορτία.

Τα αποτελέσματα έδειξαν ότι υπό καθεστώς δυναμικής τιμολόγησης επιτυγχάνεται μείωση του κόστους ηλεκτρικής ενέργειας για τον τελικό χρήστη, μέσω χρονικής μετατόπισης των μη κρίσιμων, ελεγχόμενων φορτίων του. Σε επίπεδο μικροδικτύου, η ενσωμάτωση της οικιακής ευελιξίας συνέβαλε σε αυξημένη απορρόφηση της τοπικής παραγωγής από Φ/Β και σε οικονομικότερη διαχείριση της ανταλλαγής ισχύος με το κεντρικό δίκτυο. Η αρχιτεκτονική διατηρεί τον αποκεντρωμένο χαρακτήρα λήψης αποφάσεων και περιορίζει την ανταλλαγή ευαίσθητων δεδομένων κατανάλωσης με συστήματα ή διαχειριστές εκτός οικίας. Αυτό το χαρακτηριστικό είναι ιδιαίτερα σημαντικό για την αποδοχή τέτοιων συστημάτων από τους χρήστες.

Συνολικά, αποδείχθηκε ότι ένας οικιακός μεσολαβητής βασισμένος σε πράκτορες μπορεί να λειτουργήσει ως αποτελεσματικό ενδιάμεσο επίπεδο οικονομικής και λειτουργικής ενσωμάτωσης των φορτίων στη βέλτιστη διαχείριση του συνολικού δικτύου. Παρέχει ποσοτικοποιημένη και αξιοποιήσιμη ευελιξία, χωρίς να απαιτεί πλήρη κεντρικό έλεγχο ή λεπτομερή γνώση της εσωτερικής λειτουργίας της οικίας. Η εργασία συμβάλλει έτσι στη γεφύρωση του κενού μεταξύ θεωρητικής έννοιας της ευελιξίας και της πρακτικής υλοποίησης σε πραγματικά ενεργειακά συστήματα.

7.2 Μελλοντικές βελτιώσεις και προεκτάσεις

Το προτεινόμενο πλαίσιο επικοινωνίας μεταξύ πρακτόρων φορτίων, πράκτορα οικίας και διαχειριστή δικτύου είναι λειτουργικά ολοκληρωμένο. Ωστόσο, υπάρχουν σημαντικά περιθώρια βελτίωσης στις επιμέρους λειτουργικότητες που υλοποιούν οι πράκτορες. Αρχικά, για λόγους απλοποίησης, στην παρούσα διπλωματική εργασία, θεωρήθηκε ότι οι πράκτορες της αντλίας νερού και του θερμοσίφωνα μπορούν να προβλέπουν ικανοποιητικά την κατανάλωση νερού της επόμενης ημέρας. Στην πράξη όμως, η πρόβλεψη αυτή αποτελεί ένα ξεχωριστό πρόβλημα. Θα μπορούσε να ενσωματωθεί σε κάθε πράκτορα ένα μοντέλο μηχανικής μάθησης, το οποίο, με βάση ιστορικά δεδομένα κατανάλωσης, θα προβλέπει με μεγαλύτερη ακρίβεια τις ανάγκες του τελικού χρήστη. Με τον τρόπο αυτό, ο ημερήσιος προγραμματισμός θα βασίζεται σε πιο ρεαλιστικές παραδοχές και θα βελτιώνεται η ποιότητα των αποφάσεων.

Επιπλέον, οι πράκτορες σχεδιάστηκαν ώστε να αποκρίνονται κυρίως σε αιτήματα ευελιξίας που αφορούν ημερήσιο προγραμματισμό και παροχή επικουρικών υπηρεσιών σε χρονικό ορίζοντα 15 λεπτών, όπως η τριτεύουσα εφεδρεία. Σε μελλοντική επέκταση, το πλαίσιο θα μπορούσε να υποστηρίξει και ταχύτερες

αποκρίσεις, επιτρέποντας συμμετοχή σε δευτερεύουσα ή ακόμη και πρωτεύουσα εφεδρεία. Η δυνατότητα αυτή θα αποκτούσε ιδιαίτερη σημασία σε συστήματα με υψηλή διείσδυση ΑΠΕ, όπου η μειωμένη αδράνεια καθιστά αναγκαία την ταχεία εξισορρόπηση ισχύος.

Παράλληλα, έχει ήδη υλοποιηθεί ένα γενικευμένο μοντέλο οικίας, το οποίο μπορεί να επεκταθεί σε επίπεδο ενεργειακής κοινότητας. Κάθε οικία θα μπορούσε να διαθέτει διαφορετικό σύνολο φορτίων και διαφορετικές ανάγκες, διατηρώντας όμως την ίδια αρχιτεκτονική. Η απουσία άμεσης εξάρτησης μεταξύ πρακτόρων φορτίων, καθώς και μεταξύ πρακτόρων και πράκτορα οικίας, καθιστά το πλαίσιο εύκολα επεκτάσιμο. Η δομή αυτή αποτυπώνεται και στον κώδικα που παρατίθεται στο Παράρτημα.

Τέλος, μία ιδιαίτερα ενδιαφέρουσα προοπτική αφορά την ενσωμάτωση της συμπεριφοράς ή/και ανάδρασης του καταναλωτή στο αποτέλεσμα της λειτουργίας του συστήματος. Ο χρήστης θα μπορούσε να αξιολογεί την ποιότητα του παραγόμενου προγράμματος, για παράδειγμα, ως προς το αν η θερμοκρασία του νερού ήταν ικανοποιητική. Η πληροφορία αυτή θα μπορούσε να χρησιμοποιηθεί από τον αντίστοιχο πράκτορα για προσαρμογή των παραμέτρων βελτιστοποίησης, οδηγώντας σε ένα σύστημα πιο ανθρωποκεντρικό, προσαρμοζόμενο δηλαδή στον εκάστοτε χρήστη.

7.3 Προσωπικά συμπεράσματα

Η εκπόνηση της διπλωματικής εργασίας, ως το τελευταίο στάδιο των σπουδών, αποτελεί μια διαδικασία ουσιαστικής μετάβασης από τον ρόλο του φοιτητή σε αυτόν του μηχανικού. Η αναζήτηση και αξιολόγηση πληροφορίας, με στόχο την αναγνώριση ενός πραγματικού προβλήματος και τη διατύπωση μιας εφαρμόσιμης λύσης, γεφυρώνει τη θεωρία με την πράξη. Η προσπάθεια να σχεδιαστεί ένα σύστημα που να έχει πρακτική αξία, τόσο σε μικρή κλίμακα, μέσω της μείωσης του κόστους κατανάλωσης για τον τελικό χρήστη, όσο και σε μεγαλύτερη κλίμακα, μέσω της διευκόλυνσης της διείσδυσης των ΑΠΕ και της βελτίωσης της λειτουργίας των δικτύων, ανέδειξε τον κοινωνικό ρόλο του μηχανικού.

Η επίλυση τεχνικών προβλημάτων δεν αποτελεί αυτοσκοπό, αποκτά ουσιαστικό νόημα όταν συμβάλλει στη βελτίωση της καθημερινότητας και της ποιότητας ζωής των ανθρώπων. Η διαδικασία ανάλυσης επιμέρους προβλημάτων και η σύνθεσή τους σε

ένα λειτουργικό σύστημα συνέβαλαν σημαντικά στην ανάπτυξη ορθολογικής σκέψης και μεθοδικότητας. Συνολικά, η εκπόνηση της διπλωματικής εργασίας αποτέλεσε εμπειρία ωρίμανσης, ενισχύοντας την αντίληψη ότι ο ρόλος του μηχανικού συνδέεται άμεσα με την τεκμηριωμένη συμβολή του στην κοινωνία.

8 Βιβλιογραφία

[1] E. Commission, «Energy and the Green Deal».

[2] Eurostat, European Commission –, 21 Feb. 2025. [Ηλεκτρονικό]. Available: https://ec.europa.eu/eurostat/web/products-eurostat-news/w/ddn-20250221-3.

[3] Eurostat, European Commission –, 19 March 2025. [Ηλεκτρονικό]. Available: https://ec.europa.eu/eurostat/web/products-eurostat-news/w/ddn-20250319-1.

[4] Δ. Μιχαηλίδης, «Παραγωγή Ηλεκτρικής Ενέργειας & Ενεργειακή Μετάβαση,» GreeceInFigures.com, 20 Σεπτέμβρης 2024. [Ηλεκτρονικό]. Available: https://www.greeceinfigures.com/paragogi-ilektrikis-energeias/.

[5] I. Diahovchenko, M. Kolcun, Z. Čonka και V. Savkiv, «Progress and Challenges in Smart Grids: Distributed Generation, Smart Metering, Energy Storage and Smart Loads,» *Iranian Journal of Science and Technology, Transactions of Electrical Engineering,* τόμ. 44, αρ. 4, p. 1319–1333, December 2020.

[6] C. P. Ohanu, S. A. Rufai και U. C. Oluchi, «A comprehensive review of recent developments in smart grid through renewable energy resources integration,» *Heliyon,* τόμ. 10, αρ. 3, p. e25705, Feb 2024.

[7] Eurostat, 2025.

[8] E. C. –. Eurostat, 25 June 2025. [Ηλεκτρονικό]. Available: https://ec.europa.eu/eurostat/web/products-eurostat-news/w/ddn-20250625-2.

[9] B. M. Weedy, B. J. Cory, N. Jenkins και B. E. Janaka , Power System Analysis, New York: McGraw-Hill, 1994.

[10] M. F. Akorede, H. Hizam και a. E. Pouresmaeil, «Distributed energy resources and benefits to the environment,» *Renewable and Sustainable Energy Reviews,* τόμ. 14, αρ. 2, p. 724–734, February 2010.

[11] M. Asif, «Sustainable energy transition in the 21st century,» *Elsevier eBooks,* p. 27–38, January 2022.

[12] G. Tsagarakis, R. C. Thomson, A. J. C. G. P. H. και a. A. E. Kiprakis, «Assessment of the cost and environmental impact of demand side management on residential sector,» *Ninth International Conference on Ecological Vehicles and Renewable Energies (EVER),* pp. 1-7, 2014.

[13] Gaur, None Manisha και P., «Scope of architecture improvement for standalone microgrids,» *Elsevier eBooks,* p. 113–135, Jan 2021.

[14] N. Hatziargyriou, H. Asano, R. Iravani και C. Marn, «Microgrids,» *IEEE Power and Energy Magazine,* τόμ. 5, αρ. 4, pp. 78-94, July-August 2007.

[15] S. Ishaq, . I. Khan, . S. Rahman, . T. Hussain και . A. Iqbal, «A review on recent developments in control and optimization of micro grids,» *Energy Reports,* τόμ. 8, p. 4085–4103, Nov 2022.

[16] Freitas, M. E. T. Souza και L. C. G., «Grid-Connected and Seamless Transition Modes for Microgrids: An Overview of Control Methods, Operation Elements, and General Requirements,» *IEEE Access,* τόμ. 10, pp. 97802-97834, 12 September 2022.

[17] Cheng, D. Lee και C. -C., «Energy savings by energy management systems: A review,» *Renewable and Sustainable Energy Reviews*, τόμ. 56, p. 760–777, April 2016.

[18] F. D. e. al., «Exploiting demand-side flexibility: State-of-the-art, open issues and social perspective,» *Renewable and Sustainable Energy Reviews*, τόμ. 165, p. 112605, Sep 2022.

[19] European Investment Bank, «HEDNO SMART METERS I,» August 2023. [Ηλεκτρονικό]. Available: https://www.eib.org/en/projects/pipelines/all/20220823.

[20] F. Alfaverh και M. Denai, «Demand Response Strategy Based on Reinforcement Learning and Fuzzy Reasoning for Home Energy Management,» *IEEE Access*, p. 99, February 2020.

[21] W. Li, R. Li, H. Yu, H. Gao και a. L. Yang, «Study on the year-round demand flexibility in residential buildings: Control strategies and quantification methods,» *Energy and Buildings*, τόμ. 336, p. 115582, Jun 2025.

[22] S. Chondrogiannis, J. Vasiljevska, A. Marinopoulos, I. Papaioannou και G. and Flego, «Local electricity flexibility markets in Europe,» Publications Office of the European Union, 2022.

[23] M. Z. Degefa, I. B. Sperstad και a. H. Sæle, «Comprehensive classifications and characterizations of power system flexibility resources,» *Electric Power Systems Research*, τόμ. 194, p. 107022, May 2021.

[24] Z. Luo και e. al, «Demand Flexibility of Residential Buildings: Definitions, Flexible Loads, and Quantification Methods,» *Engineering*, τόμ. 16, p. 123–140, 2022.

[25] A. Arteconi, N. J. Hewitt και F. Polonara, «State of the art of thermal storage for demand-side management,» *Applied Energy*, τόμ. 93, αρ. May, p. 371–389, 2012.

[26] C. Li, D. Srinivasan και a. T. Reindl, «Real-time scheduling of time-shiftable loads in smart grid with dynamic pricing and photovoltaic power generation,» *IEEE Innovative Smart Grid Technologies - Asia*, pp. 1-6, 2015.

[27] X. Lu, K. Zhou, X. Zhang και a. S. Yang, «A systematic review of supply and demand side optimal load scheduling in a smart grid environment,» *Journal of Cleaner Production*, τόμ. 203, p. 757–768, December 2018.

[28] B. Han, Y. Zahraoui, M. Mubin, S. Mekhilef, M. Seyedmahmoudian και a. A. Stojcevski, «Home Energy Management Systems: A Review of the Concept, Architecture, and Scheduling Strategies,» *IEEE Access*, τόμ. 11, pp. 19999-20025, 2023.

[29] Χτύπος, «Πορτοκαλί τιμολόγια ρεύματος: Πώς αλλάζουν οι χρεώσεις από Φεβρουάριο και Απρίλιο,» 2026.

[30] C. Xiao, «Research on Flexible Load Aggregation and Coordinated Control Methods Considering Dynamic Demand Response,» *Energy Engineering*, τόμ. 122, αρ. 7, p. 2719–2750, 2025.

[31] Z. Luo, J. Peng, X. Zhang, H. Jiang και a. M. Lv, «Load flexibility quantification of electric water heaters under various demand-side management strategies and seasons,» *Journal of Building Engineering*, τόμ. 97, p. 110724, September 2024.

[32] D. Mousdell, «Understanding Mixing Valves, Tempering Valves, and TMVs,» h2x, May 2024. [Ηλεκτρονικό]. Available: https://www.h2xengineering.com/blogs/understanding-mixing-tempering-valves-tmvs/.

[33] P. Pieter D. van Schalkwyk, J. A. A. Engelbrecht και a. M. J. Booysen, «Thermal Stratification and Temperature Variation in Horizontal Electric Water Heaters: A Characterisation Platform,» *Energies,* τόμ. 15, αρ. 8, p. 2840, April 2022.

[34] A. Moreau, «Control Strategy for Domestic Water Heaters during Peak Periods and its Impact on the Demand for Electricity,» *Energy Procedia,* τόμ. 12, p. 1074–1082, 2011.

[35] C. Diaz, F. Ruiz και a. D. Patino, «Modeling and control of water booster pressure systems as flexible loads for demand response,» *Applied Energy,* τόμ. 204, p. 106–116, October 2017.

[36] O. P. Marta J.N. , N. M. Mateus και d. G. G. Carrilho , «Measured and modeled performance of internal mass as a thermal energy battery for energy flexible residential buildings,» *Applied Energy,* τόμ. 219, p. 252–267, April 2019.

[37] EUPD Group, «European Residential Battery Storage Market: Strong Momentum Continues in 2025 – EUPD Research Highlights Strategic Growth Opportunities,» July 2025. [Ηλεκτρονικό]. Available: https://eupd-group.com/european-residential-battery-storage-market-strong-momentum-continues-in-2025-eupd-research-highlights-strategic-growth-opportunities/.

[38] Y. Izmirlioglu, L. Pham, . T. C. Son και E. Pontell, «A Survey of Multi-Agent Systems for Smartgrids,» *Energies,* τόμ. vol. 17, αρ. no. 15, pp. p. 3620,, 2024.

[39] P. Michailidis, I. Michailidis και E. Kosmatopoulos, «Review and Evaluation of Multi-Agent Control Applications for Energy Management in Buildings,» *Energies,* τόμ. 17, αρ. 19, p. 4835, 2024.

[40] E. Erdemir, D. Gündüz και D. P. , «Privacy in Dynamical Systems,» *Smart Meter Privacy,* 2020.

[41] O. Loyola-González, «Black-Box vs. White-Box: Understanding Their Advantages and Weaknesses From a Practical Point of View,» *IEEE Access,* τόμ. 7, pp. 154096-154113, 2019.

[42] J. Rust, «Dynamic Programming,» *The New Palgrave Dictionary of Economics,* p. 1471–1489, 2008.

[43] R. Bellman, «The theory of dynamic programming,» *Bulletin of the American Mathematical Society,* pp. 503-515, 1954.

[44] Yang Zhifang, Yong Pei και a. X. Mingxu, «Revisit power system dispatch: Concepts, models, and solutions,» *iEnergy,* τόμ. 2, pp. 43-62, 2023.

[45] V. V. G. T. H. a. J. Q. H. Zhang, «A Mixed-Integer Linear Programming Approach for Multi-Stage Security-Constrained Transmission Expansion Planning,» *IEEE Transactions on Power Systems,* τόμ. 27, αρ. 2, pp. 1125-1133, 2012.

[46] R. F. C. Misener, «Piecewise-Linear Approximations of Multidimensional Functions,» *Journal of Optimization Theory and Applications,* p. 120–147, 2010.

9 Παράρτημα

9.1 Κώδικας υλοποίησης οικιακού μεσολαβητή

Η υλοποίηση των πρακτόρων έγινε με χρήση κλάσεων ώστε ο κώδικας να είναι επεκτάσιμος και να υπάρχει σαφής διαχωρισμός αρμοδιοτήτων του κάθε πράκτορα. Κάθε πράκτορας αξιοποιεί το την γενικευμένη υλοποίηση του λύτη ΔΠ για να βελτιστοποιήσει τον προγραμματισμό του φορτίου του. Ο συντονισμός των πρακτόρων φορτίου και του πράκτορα μπαταρίας γίνεται μέσω του πράκτορα οικίας.

9.1.1 Λύτης δυναμικού προγραμματισμού

```
classdef DynamicProgrammingSolver
    %DYNAMICPROGRAMMINGSOLVER
    % Generic finite-horizon DP solver for discrete state/action spaces.
    %
    % Solves:
    %   min sum_t l(x_t, u_t, t) + lT(x_{T+1})
    %   s.t. x_{t+1} = f(x_t, u_t, t)

    properties
        T           % horizon
        X           % state grid (1 x nX)
        A           % action grid (1 x nA)
        f           % dynamics: x_next = f(x,u,t)
        l           % stage cost: l(x,u,t)
        lT          % terminal cost: lT(x)
        feasible    % feasible(x_next, x, u, t)
        strict_rollout (1,1) logical = true  % error if rollout hits
infeasibility
    end

    methods
        function obj = DynamicProgrammingSolver(T, X, A, f, l, lT,
feasible)
            if nargin < 7 || isempty(feasible)
                feasible = @(x_next, x, u, t) true;
            end
            obj.T  = T;
            obj.X  = X(:)';   % ensure row
            obj.A  = A(:)';   % ensure row
            obj.f  = f;
            obj.l  = l;
            obj.lT = lT;
            obj.feasible = feasible;
        end

        function result = solve(obj, x0)

            nX = numel(obj.X);
            nA = numel(obj.A);

            % Cost-to-go
```

```
            J  = inf(obj.T+1, nX);
            pi = zeros(obj.T, nX, 'uint16');  % store action indices

            % Terminal cost
            for i = 1:nX
                v = obj.lT(obj.X(i));
                if ~isfinite(v)
                    % inf is OK (hard terminal constraint), NaN is not
                    if isnan(v)
                        error("DP: lT returned NaN at terminal state
X(%d)=%.6g. Fix lT (avoid 0*Inf).", i, obj.X(i));
                    end
                end
                J(obj.T+1, i) = v;
            end

            % Backward recursion
            for t = obj.T:-1:1
                for i = 1:nX
                    x = obj.X(i);

                    best_cost = inf;
                    best_a = uint16(0);  % 0 => no feasible action found

                    for a = 1:nA
                        u = obj.A(a);
                        x_next = obj.f(x, u, t);

                        if ~obj.feasible(x_next, x, u, t)
                            continue
                        end
                        % Project to grid
                        [~, j] = min(abs(obj.X - x_next));
                        % Future cost must be finite (otherwise cannot
reach terminal-feasible set)
                        future = J(t+1, j);
                        if ~isfinite(future)
                            continue;
                        end

                        stage = obj.l(x, u, t);
                        if ~isfinite(stage)
                            % NaN/Inf stage costs are treated as invalid
                            continue;
                        end

                        cost = stage + future;

                        % cost should be finite here; still guard
                        if ~isfinite(cost)
                            continue;
                        end

                        if cost < best_cost
                            best_cost = cost;
                            best_a = uint16(a);
                        end
                    end
```

```
                J(t, i)  = best_cost;
                pi(t, i) = best_a; % 0 if infeasible
            end
        end

        % ---------------------------
        % Forward rollout
        % ---------------------------
        x_traj = zeros(1, obj.T+1);
        u_traj = zeros(1, obj.T);

        [~, s] = min(abs(obj.X - x0));
        x_traj(1) = obj.X(s);

        for t = 1:obj.T
            a = pi(t, s);

            if a == 0
                if obj.strict_rollout
                    error("DP rollout: no feasible action at t=%d,
state index=%d (x=%.6g).", ...
                          t, s, obj.X(s));
                else
                    % stop rollout gracefully
                    u_traj(t:end) = nan;
                    x_traj(t+1:end) = x_traj(t);
                    break;
                end
            end

            u = obj.A(double(a));
            u_traj(t) = u;

            % Continuous transition
            x_next = obj.f(x_traj(t), u, t);

            % Project to grid and store projected state
            [~, s_next] = min(abs(obj.X - x_next));
            x_traj(t+1) = obj.X(s_next);

            % Optional: enforce feasibility consistency during rollout
            if obj.strict_rollout
                if ~obj.feasible(x_next, x_traj(t), u, t)
                    error("DP rollout: reached infeasible projected
state at t=%d (x=%.6g -> x_next=%.6g).", ...
                          t, x_traj(t), x_next);
                end
            end
            % x_traj(t+1) = x_next;
            s = s_next;
        end
        % ---------------------------
        % Output
        % ---------------------------
        result = struct();
        result.state_traj  = x_traj;
        result.action_traj = u_traj;
        result.policy_idx  = pi;
        result.cost_to_go  = J;
```

```
        end
    end
end
```

9.1.2 Διεπαφή υλοποίησης πράκτορα φορτίου

```
classdef (Abstract) LoadAgent < handle
    %LOADAGENT Abstract interface for any non critical load agent.
    %
    % This class defines the common contract for all load agents.
    % ============================================================
    % CORE DESIGN PRINCIPLES
    % ============================================================
    %  - Each load agent encapsulates its OWN physical state
    %    (thermal, hydraulic, electrochemical, etc.).
    %
    %  - Planning is performed locally using
    %    optimization algorithm (e.g., Dynamic Programming).
    %
    %  - NO physical execution or state update is allowed in plan().
    %
    %  - Physical execution happens ONLY via applySetpoint(),
    %    and ONLY for ONE real time step.
    %
    %  - After execution, each agent MUST replan the remaining
    %    horizon starting from the updated real state.
    %
    %  - The HomeAgent acts as a COORDINATOR, not a controller:
    %      * it requests plans,
    %      * evaluates flexibility,
    %      * selects setpoints,
    %      * but never manipulates internal agent states.
    %
    % ============================================================
    % TYPICAL CONTROL SEQUENCE
    % ============================================================
    %   1) HomeAgent calls plan() on all load agents.
    %   2) Each agent proposes:
    %         - a baseline power profile
    %         - a physically feasible flexibility envelope.
    %   3) HomeAgent decides a targeted power for the CURRENT step.
    %   4) HomeAgent calls applySetpoint() on selected agents.
    %   5) Each agent:
    %         - executes ONE step,
    %         - updates its physical state,
    %         - replans the remaining horizon.

    properties
        % ============================================================
        % Planning horizon parameters
        % ============================================================
        T (1,1) double {mustBeInteger, mustBePositive} = 24
        % Number of discrete time steps in the planning horizon

        dt_hours (1,1) double {mustBePositive} = 0.25
        % Duration of each time step [hours]
    end
```

```
methods
    function obj = LoadAgent(T, dt_hours)
        % Constructor for load agents
        % INPUTS:
        %   T         : planning horizon (number of steps)
        %   dt_hours  : duration of each step [hours]
        if nargin >= 1
            obj.T = T;
        end
        if nargin >= 2
            obj.dt_hours = dt_hours;
        end
    end
end

methods (Abstract)
    % ============================================================
    % Agent identification
    % ============================================================
    label = name(obj)
    % Returns a short identifier for the load agent
    % (e.g., "EWH", "Pump", "Battery").

    % ============================================================
    % Rolling-horizon planning (proposal-only)
    % ============================================================
    [P_kw, info] = plan(obj, signals)
    % Computes a rolling-horizon optimization proposal.
    %
    % IMPORTANT:
    %   - MUST NOT execute any physical action.
    %   - MUST NOT update the internal physical state.
    %
    % INPUT:
    %   signals : struct with exogenous inputs, e.g.
    %       signals.dynamic_price        [1xT]
    %       signals.V_out_EWH            [1xT] (EWH)
    %       signals.V_out_pump           [1xT] (Pump)
    %       signals.residual_load_kw     [1xT] (Battery)
    %
    % OUTPUT:
    %   P_kw  : [1xT] baseline power proposal (policy output)
    %   info  : struct with auxiliary data:
    %             - predicted state trajectories
    %             - disturbances
    %             - optimization policy
    %             - any debug/plotting info
    %
    % NOTE:
    %   The returned profile is a PROPOSAL.
    %   No physical execution occurs here.

    % ============================================================
    % Flexibility exposure (state-based)
    % ============================================================
    caps = getFlexibility(obj)
    % Returns the flexibility envelope associated with the most
    % recent planning or execution step.
```

```
        %
        % OUTPUT:
        %   caps.P_min   [1xT_rem] minimum feasible power
        %   caps.P_base  [1xT_rem] baseline power
        %   caps.P_max   [1xT_rem] maximum feasible power
        %
        % SEMANTICS:
        %   - Flexibility bounds MUST be derived from the agent's
        %     physical state and model constraints.
        %   - Bounds MUST guarantee feasibility of the NEXT state.
        %
        % NOTE:
        %   - Immediately after plan():     T_rem = T
        %   - After applySetpoint():        T_rem = T - executed steps
        %   Flexibility ALWAYS refers to the REMAINING horizon.

        % ============================================================
        % Control execution (one-step only)
        % ============================================================
        applySetpoint(obj, P_set)
        % Executes the control action for the CURRENT time step.
        %
        % INPUT:
        %   P_set :
        %     - scalar : targeted power for the next step
        %     - vector : optional trajectory (only first element executed)
        %
        % REQUIRED BEHAVIOR:
        %   - Quantize P_set to admissible actions
        %   - Update the internal physical state for ONE step
        %   - Replan the remaining horizon starting from the updated state
        %   - Update flexibility envelopes accordingly
        %
        % This method is the ONLY place where:
        %   - physical execution
        %   - real state updates
        % are allowed.
    end
end
```

9.1.3 Πράκτορας θερμοσίφωνα

```
classdef EWHAgent < LoadAgent
    properties
        % Electrical limits
        % ============================================================
        Pmin_kw (1,1) double
        Pmax_kw (1,1) double

        % Thermal model parameters
        % ============================================================
        T0 (1,1) double              % current water temperature [°C]
        T_env (1,1) double = 20.0    % ambient temperature [°C]
        T_min (1,1) double           % comfort lower bound [°C]
        T_max (1,1) double           % safety upper bound [°C]

        % Water tank physical parameters
        Cw (1,1) double = 4184       % specific heat [J/(kg·K)]
        rho (1,1) double = 997       % water density [kg/m^3]
```

```
        V_tank (1,1) double = 0.15    % tank volume [m^3]
        UA (1,1) double = 10          % overall heat transfer [W/K]
        eta (1,1) double = 0.95       % heater efficiency

        % Cold water properties
        T_cold (1,1) double = 20      % cold water inlet temp [°C]
        T_desired (1,1) double = 40   % draw reference temp [°C]

        % Derived (computed once)
        C_thermal (1,1) double        % [J/K]
        R_thermal (1,1) double        % [K/W]

        % Discretization parameters
        % =================================================================
        number_of_discrete_states  (1,1) double
        number_of_discrete_actions (1,1) double

        % User settings
        % =================================================================
        T_final (1,1) double           % terminal condition temp [°C]
        Tolerance (1,1) double         % |Temperature - T_desired| <
Tolerance

        % Interface book keeping
        % =================================================================
        last_info struct = struct()
        last_caps struct = struct()
        P_levels_last double = []
        P_set_traj double = []
        set_idx (1,1) double {mustBeInteger, mustBeNonnegative} = 0
    end

    methods (Access = private)
        function T_next = physicalModel(obj, T, P_exec, V_out)
            % One-step thermal state transition for the EWH
            %
            % T      : current water temperature [°C]
            % P_exec : executed electrical power [kW]
            % V_out  : hot water draw during this step [m^3]
            %=================================================================

            % Time step
            dt = obj.dt_hours * 3600;   % [s]

            % Heater thermal power
            Q_boiler = obj.eta * (P_exec * 1000);   % [W]

            % Thermal loss due to hot water draw
            Q_cold = obj.Cw * obj.rho * V_out * ...
                     (obj.T_desired - obj.T_cold) / dt;   % [W]

            % Thermal inertia coefficient (PHYSICAL, dt-dependent)
            a = exp( -dt / (obj.R_thermal * obj.C_thermal) );

            % Steady-state temperature
            T_inf = obj.T_env + (Q_boiler - Q_cold) * obj.R_thermal;

            % State update
            T_next = a * T + (1 - a) * T_inf;
```

```
        end

        function v = terminalCost(obj, Tw)
            if abs(Tw - obj.T_final) <= obj.Tolerance
                v = 0;
            else
                v = realmax;  % or Inf
            end
        end
        function [Pmin, Pmax] = powerBounds(obj, T_t, V_out_t)
            % Compute feasible power bounds for ONE step
            % based on thermal state and physical limits

            dt = obj.dt_hours * 3600;

            % Thermal coefficients
            a = exp( -dt / (obj.R_thermal * obj.C_thermal) );

            % Cold water thermal loss
            Q_cold = obj.Cw * obj.rho * V_out_t * ...
                     (obj.T_desired - obj.T_cold) / dt;   % [W]

            % Invert model for Tmin and Tmax
            Q_req_min = ( ...
                (obj.T_min - a*T_t)/(1-a) - obj.T_env ) / obj.R_thermal +
Q_cold;

            Q_req_max = ( ...
                (obj.T_max - a*T_t)/(1-a) - obj.T_env ) / obj.R_thermal +
Q_cold;

            % Convert to electrical power [kW]
            Pmin_state = Q_req_min / (obj.eta * 1000);
            Pmax_state = Q_req_max / (obj.eta * 1000);

            % Combine with electrical limits
            Pmin = max(obj.Pmin_kw, Pmin_state);
            Pmax = min(obj.Pmax_kw, Pmax_state);
        end

        function replanRemainingHorizon(obj)
            % Re-solve DP for the remaining horizon starting
            % from the CURRENT real state obj.T0

            % Remaining horizon length
            T_remain = obj.T - obj.set_idx;
            if T_remain <= 0
                return;
            end

            % Shift signals
            dynamic_price = obj.last_info.dynamic_price(obj.set_idx+1:end);
            V_out         = obj.last_info.V_out_EWH(obj.set_idx+1:end);

            % Discretization
            P_levels = obj.P_levels_last;
            T_grid = linspace(obj.T_min, obj.T_max,
obj.number_of_discrete_states);
```

```
            % Dynamics
            f = @(Tw, P, t) obj.physicalModel(Tw, P, V_out(t));

            % Cost
            l = @(Tw, P, t) dynamic_price(t) * P;
            lT = @(Tw) obj.terminalCost(Tw);

            feasible = @(Tw_next, Tw, P, t) ...
                (Tw_next >= obj.T_min) && ...
                (Tw_next <= obj.T_max);

            % Solve DP from UPDATED real state
            solver = DynamicProgrammingSolver( ...
                T_remain, T_grid, P_levels, f, l, lT, feasible);
            solver.strict_rollout = true;
            res = solver.solve(obj.T0);

            % Build new caps
            nA = numel(P_levels);
            P_base = zeros(1, T_remain);
            P_min  = zeros(1, T_remain);
            P_max  = zeros(1, T_remain);

            for t = 1:T_remain
                idx = find(P_levels >= res.action_traj(t) - 1e-9, 1,
'first');
                if isempty(idx), idx = nA; end
                P_base(t) = P_levels(idx);
                T_t = res.state_traj(t);
                Vt  = V_out(t);
                [Pmin_t, Pmax_t] = obj.powerBounds(T_t, Vt);

                P_min(t) = Pmin_t;
                P_max(t) = Pmax_t;
            end

            % Overwrite caps with NEW solution
            obj.last_caps = struct( ...
                'P_min',  P_min, ...
                'P_base', P_base, ...
                'P_max',  P_max );

            % Update info
            obj.last_info.P_opt = res.action_traj;
            obj.last_info.temperature = res.state_traj;
        end
    end

    methods
        function obj = EWHAgent(T, dt_hours, params)
            % Constructor: initializes horizon and optional parameters
            obj@LoadAgent(T, dt_hours);
            if nargin >= 3
                f = fieldnames(params);
                for i = 1:numel(f)
                    if isprop(obj, f{i})
                        obj.(f{i}) = params.(f{i});
                    end
                end
```

```
            end
            obj.C_thermal   = obj.Cw * obj.rho * obj.V_tank;   % J/K
            obj.R_thermal = 1 / obj.UA;                        % K/W
        end

        function label = name(obj)
            label = "EWH";
        end

        function [P_kw, info] = plan(obj, signals)
            % ============================================================
            % PLAN (proposal only)
            %
            % - Solves DP from CURRENT state obj.T0 over horizon T.
            % - DOES NOT execute any action and DOES NOT update obj.T0.
            % - Exposes baseline and flexibility envelope for horizon T.
            %
            % Execution happens ONLY in applySetpoint().
            % ============================================================

            T = obj.T; % planning horizon

            % ---- Price signal ----
            % Dynamic energy pricing signal (exogenous policy)
            % time-varying electricity tariff (e.g., TOU / RTP) used in the
cost function.
            % ------------------------------------------------------------
            dynamic_price = signals.dynamic_price(:)';
            dynamic_price = dynamic_price / (max(dynamic_price) + eps);

            % Discretization
            % ------------------------------------------------------------
            % Discrete power actions
            P_levels = linspace( ...
                obj.Pmin_kw, ...
                obj.Pmax_kw, ...
                obj.number_of_discrete_actions);
            % Discrete temperature states
            T_grid = linspace( ...
                obj.T_min, ...
                obj.T_max, ...
                obj.number_of_discrete_states);

            % Thermal disturbance (hot water consumption)
            % ------------------------------------------------------------
            if isfield(signals,'V_out_EWH')
                V_out = signals.V_out_EWH(:)';
            else
                V_out = zeros(1,T);
            end


            % System dynamics: first-order RC thermal model
            Tw = obj.T0;
            % ------------------------------------------------------------
            f = @(Tw, P, t) obj.physicalModel(Tw, P, V_out(t));

            % Stage cost: energy cost
            % ------------------------------------------------------------
```

```
            l = @(Tw, P, t) dynamic_price(t) * P;

            % Hard terminal comfort constraint
            lT = @(Tw) obj.terminalCost(Tw);

            feasible = @(Tw_next, Tw, P, t) ...
                (Tw_next >= obj.T_min) && ...
                (Tw_next <= obj.T_max);


            % Dynamic Programming solution
            % ---------------------------------------------------------
            solver = DynamicProgrammingSolver( ...
                T, T_grid, P_levels, f, l, lT, feasible);

            % Optional but recommended
            solver.strict_rollout = true;

            res = solver.solve(obj.T0);

            P_opt = res.action_traj; % [1xT]
            T_traj = res.state_traj;

            % Build (Pmin, Pbase, Pmax) profiles for remaining horizon
            nA = numel(P_levels);
            P_base = zeros(1,T);
            P_min  = zeros(1,T);
            P_max  = zeros(1,T);
            for t = 1:T
                % Baseline
                idx = find(P_levels >= P_opt(t) - 1e-9, 1, 'first');
                if isempty(idx), idx = nA; end
                P_base(t) = P_levels(idx);

                % SoC-analogue: temperature-based bounds
                T_t = T_traj(t);
                Vt  = V_out(t);
                [Pmin_t, Pmax_t] = obj.powerBounds(T_t, Vt);

                P_min(t) = Pmin_t;
                P_max(t) = Pmax_t;
            end


            % ---- Output to home agent ----
            % baseline proposal for the horizon
            P_kw = P_base;
            % Info for debugging/plots
            info = struct();
            info.P_levels = P_levels;
            info.P_opt = P_opt;
            info.P_min = P_min;
            info.P_base = P_base;
            info.P_max = P_max;
            info.temperature = T_traj;
            info.V_out_EWH = V_out;
            info.dynamic_price = dynamic_price;
            info.policy = "DP-based EWH (proposal only; execution via
applySetpoint)";
```

```
            % ---- Tracking ----
            obj.P_levels_last = P_levels;
            obj.last_caps = struct( ...
                'P_min',  P_min, ...
                'P_base', P_base, ...
                'P_max',  P_max );
            obj.last_info = info;
        end

        function caps = getFlexibility(obj)
            % Returns the last computed envelope (could be full horizon
            % right after plan(), or remaining horizon after
applySetpoint()).
            if isempty(fieldnames(obj.last_caps))
                error("EWHAgent.getFlexibility(): call plan() first to
compute flexibility.");
            end
            caps = obj.last_caps;
        end

        function applySetpoint(obj, P_set)
            % ============================================================
            % APPLYSETPOINT (execution + rolling-horizon replan)
            %
            % - Executes the received power setpoint for ONE real step
            % - Updates the REAL physical state (obj.T0)
            % - Re-solves Dynamic Programming for the REMAINING horizon
            %   starting from the updated real state
            % ============================================================

            % Initialize setpoint tracking
            % ------------------------------------------------------------
            if isempty(obj.P_set_traj)
                obj.P_set_traj = nan(1, obj.T);
                obj.set_idx = 0;
            end

            % Accept scalar setpoint only (current step)
            % ------------------------------------------------------------
            if ~isscalar(P_set)
                % Keep support for trajectory, but execution is stepwise:
                % we apply only the first element NOW.
                P_set = P_set(:)';
                if isempty(P_set)
                    error("EWHAgent.applySetpoint(): empty P_set.");
                end
                P_set = P_set(1);
            end

            % Quantize to nearest discrete action
            % ------------------------------------------------------------
            P_exec = obj.quantizeToLevels(P_set);

            % Advance executed setpoint index
            % ------------------------------------------------------------
            obj.set_idx = obj.set_idx + 1;
            if obj.set_idx > obj.T
```

```
                warning("EWHAgent.applySetpoint(): received more than T
scalar setpoints; ignoring extra.");
                return;
            end
            obj.P_set_traj(obj.set_idx) = P_exec;

            % State update must use the same disturbance as planning:
            % We retrieve Qw(t) from the latest plan() call. This enforces
consistency
            % between proposed envelope and executed transition.
            if isempty(fieldnames(obj.last_info)) ||
~isfield(obj.last_info,'P_base')
                error("EWHAgent.applySetpoint(): call plan() before
applySetpoint() so disturbance is known.");
            end

            V_out = obj.last_info.V_out_EWH(:)';
            if numel(V_out) < obj.set_idx
                error("EWHAgent.applySetpoint(): disturbance horizon
inconsistent.");
            end

            % REAL physical state update (t = 1 relative to current
horizon)
            % -----------------------------------------------------------
            obj.T0 = obj.physicalModel(obj.T0, P_exec, V_out(1));

            % Store executed action for traceability
            % -----------------------------------------------------------
            obj.last_info.P_exec = P_exec;

            % Re-plan for the remaining horizon (ROLLING HORIZON)
            % -----------------------------------------------------------
            obj.replanRemainingHorizon();
        end

        function Pq = quantizeToLevels(obj, P)
             % Quantizes a continuous power setpoint to nearest discrete
action
            if isempty(obj.P_levels_last)
                Pq = P;
                return;
            end
            [~, k] = min(abs(obj.P_levels_last - P));
            Pq = obj.P_levels_last(k);
        end
    end
end
```

9.1.4 Πράκτορας μπαταρίας

```
classdef BatteryAgent < LoadAgent
    properties
        % Power limits
        %   P > 0  => charging (consumption from grid)
        %   P < 0  => discharging (supplying load / reducing import)
        P_ch_max_kw (1,1) double
        P_dis_max_kw (1,1) double
```

```
        % SoC limits
        soc0 (1,1) double
        soc_min (1,1) double
        soc_max (1,1) double

        % Battery parameters
        eta (1,1) double = 0.95
        E_kWh (1,1) double

        % Discretization parameters
        % ================================================================
        number_of_discrete_states  (1,1) double
        number_of_discrete_actions (1,1) double

        % User settings
        % ================================================================
        soc_final (1,1) double       % desired soc at the end of the day
        Tolerance (1,1) double         % |Temperature - T_desired| <
Tolerance

        % Interface bookkeeping
        last_info struct = struct()
        last_caps struct = struct()
        P_levels_last double = []
        P_set_traj double = []
        set_idx (1,1) double {mustBeInteger, mustBeNonnegative} = 0
    end

    methods (Access = private)
        function soc_next = physicalModel(obj, soc, P_exec)
            % One-step SoC transition
            %
            % soc    : current state of charge [p.u.]
            % P_exec : executed battery power [kW]
            % dt     : obj.dt_hours [h]
            %
            % Charging:  soc_{k+1} = soc_k + (P*dt*eta)/E
            % Discharge: soc_{k+1} = soc_k + (P*dt)/(eta*E)   (P is
negative)

            dt = obj.dt_hours; % [h]

            soc_next = soc + ...
                (P_exec >= 0) .* ( P_exec * dt * obj.eta / obj.E_kWh ) +
...
                (P_exec <  0) .* ( P_exec * dt / (obj.eta * obj.E_kWh) );
        end
        function v = terminalCost(obj, soc)
            if abs(soc - obj.soc_final) <= obj.Tolerance
                v = 0;
            else
                v = realmax;  % or Inf
            end
        end
        function [Pmin, Pmax] = powerBounds(obj, soc_t, residual_t)
            % Compute physically feasible power bounds for ONE step
            % based on SoC and no-export logic.
            %
```

```
            % Output semantics:
            %   Pmin <= P <= Pmax   (P in kW, battery-side)
            %
            % residual_t is used only for no-export.

            dt = obj.dt_hours;

            % SoC-based bounds
            % Max discharge (most negative) such that soc_{t+1} >= soc_min
            Pmin_soc = (obj.soc_min - soc_t) * obj.E_kWh * obj.eta / dt;
% <= 0

            % Max charge (most positive) such that soc_{t+1} <= soc_max
            Pmax_soc = (obj.soc_max - soc_t) * obj.E_kWh / (dt * obj.eta);
% >= 0

            % Actual Limits
            Pmin = max(-obj.P_dis_max_kw, Pmin_soc);
            Pmax = min( obj.P_ch_max_kw, Pmax_soc);

            % no-export constraint (only constrains discharging)
            if nargin >= 3 && ~isempty(residual_t)
                Pmin = max(Pmin, -max(residual_t,0));
            end
        end
        function replanRemainingHorizon(obj)
            % Re-solve DP for the remaining horizon starting
            % from the CURRENT real state obj.soc0

            T_remain = obj.T - obj.set_idx;
            if T_remain <= 0
                return;
            end

            % Shift signals (must exist from plan())
            dynamic_price = obj.last_info.dynamic_price(obj.set_idx+1:end);
            residual      =
obj.last_info.residual_load_kw(obj.set_idx+1:end);

            % Discretization (reuse last action grid!)
            P_levels = obj.P_levels_last;
            soc_grid = linspace(obj.soc_min, obj.soc_max,
obj.number_of_discrete_states);

            % Dynamics
            f = @(soc, P, t) obj.physicalModel(soc, P);
            % Cost
            l  = @(soc, P, t) dynamic_price(t) * P;
            lT = @(soc) obj.terminalCost(soc);
            % Feasibility (no-export constraint carried over)
            feasible = @(soc_next, soc, P, t) ...
                (soc_next >= obj.soc_min) && ...
                (soc_next <= obj.soc_max) && ...
                P >= -max(residual(t),0);

            solver = DynamicProgrammingSolver( ...
                T_remain, soc_grid, P_levels, f, l, lT, feasible);
            solver.strict_rollout = true;
            res = solver.solve(obj.soc0);
```

```
            % Build new caps for remaining horizon
            nA = numel(P_levels);
            P_base = zeros(1, T_remain);
            P_min  = zeros(1, T_remain);
            P_max  = zeros(1, T_remain);

            for t = 1:T_remain
                % Baseline from DP policy (keep your "ceiling to next
level" behavior)
                P_opt_t = res.action_traj(t);
                idx = find(P_levels >= P_opt_t - 1e-9, 1, 'first');
                if isempty(idx), idx = numel(P_levels); end
                P_base(t) = P_levels(idx);
                % Enforce no-discharge baseline when no residual load
                if residual(t) <= 0
                    P_base(t) = max(P_base(t), 0);
                end
                % SoC-based feasible bounds at predicted soc(t)
                soc_t = res.state_traj(t);
                [Pmin_t, Pmax_t] = obj.powerBounds(soc_t, residual(t));

                P_min(t) = Pmin_t;
                P_max(t) = Pmax_t;
            end
            obj.last_caps = struct( ...
                'P_min',  P_min, ...
                'P_base', P_base, ...
                'P_max',  P_max );

            % Update info (debug/plots)
            obj.last_info.P_opt = res.action_traj;
            obj.last_info.soc   = res.state_traj;
        end
    end

    methods
        function obj = BatteryAgent(T, dt_hours, params)
            % Constructor: initializes horizon and optional parameters
            obj@LoadAgent(T, dt_hours);
            if nargin >= 3
                f = fieldnames(params);
                for i = 1:numel(f)
                    if isprop(obj, f{i})
                        obj.(f{i}) = params.(f{i});
                    end
                end
            end
        end

        function label = name(obj)
            label = "Battery";
        end

        function [P_kw, info] = plan(obj, signals)
            % ========================================================
            % This agent differs conceptually from other loads:
            % - It is not a "service" that must be executed (like
EWH/pump),
```

```
            %   but an energy buffer that is coordinated by the home agent.
            % - Nevertheless, it still solves a local DP to propose a
profile
            %   consistent with SoC dynamics and constraints.
            % ========================================================

            T = obj.T;

            % Residual load is required for NO-EXPORT logic
            % residual_load_kw > 0 : house needs power (import needed)
            % residual_load_kw <= 0: there is no demand to serve
            % --------------------------------------------------------
            if isfield(signals,'residual_load_kw')
                residual = signals.residual_load_kw(:)';
            else
                error("BatteryAgent requires signals.residual_load_kw");
            end
            if numel(residual) ~= T
                error("BatteryAgent: signals.residual_load_kw must have
length T.");
            end

            % ---- Price signal:
            % Dynamic energy pricing signal (exogenous policy)
            % time-varying electricity tariff (e.g., TOU / RTP) used in the
cost function.
            % --------------------------------------------------------
            dynamic_price = signals.dynamic_price(:)';
            dynamic_price = dynamic_price / (max(dynamic_price) + eps);

            % Discretization: action space and state space
            % --------------------------------------------------------
            % Build P_levels with N discrete actions between -P_dis_max and
+P_ch_max.
            Np = obj.number_of_discrete_actions;
            if Np == 1
                P_levels = 0;
            else
                P_levels = linspace( ...
                    -obj.P_dis_max_kw, ...
                    obj.P_ch_max_kw, Np);

                % Ensure exact zero is included
                [~, k0] = min(abs(P_levels - 0));
                P_levels(k0) = 0;
            end
            % SoC discrete grid
            soc_grid = linspace( ...
                obj.soc_min, ...
                obj.soc_max, ...
                obj.number_of_discrete_states);

            % ---- Battery dynamics ----
            % Charging:  soc_{t+1} = soc_t + (P*dt*eta)/E
            % Discharge: soc_{t+1} = soc_t + (P*dt)/(eta*E)   (P is
negative)
            f = @(soc, P, t) obj.physicalModel(soc, P);

            % Stage cost: energy cost
```

```
            l = @(soc, P, t) dynamic_price(t) * P;
            % Terminal cost (keep SoC above minimum)
            lT = @(soc) obj.terminalCost(soc);
            % Hard stage feasibility (physical + coordination constraints)
            % Interpretation:
            % - If residual(t) is ~0 or negative, discharging (P<0) would
            %   create export (or unnecessary injection). We penalize it
heavily.
            feasible = @(soc_next, soc, P, t) ...
                (soc_next >= obj.soc_min) && ...
                (soc_next <= obj.soc_max) && ...
                P >= -max(residual(t),0);

            % Solve DP from CURRENT SoC state
            % ---------------------------------------------------------
            solver = DynamicProgrammingSolver( ...
                T, soc_grid, P_levels, f, l, lT, feasible);
            solver.strict_rollout = true;
            res = solver.solve(obj.soc0);

            P_opt = res.action_traj;
            soc_traj = res.state_traj;

            % Build flexibility envelope over full horizon
            nA = numel(P_levels);

            P_base = zeros(1, T);
            P_min  = zeros(1, T);
            P_max  = zeros(1, T);

            for t = 1:T
                % Baseline from DP action (quantized)
                P_opt_t = res.action_traj(t);
                idx = find(P_levels >= P_opt_t - 1e-9, 1, 'first');
                if isempty(idx), idx = numel(P_levels); end
                P_base(t) = P_levels(idx);
                % Enforce no-discharge baseline when no residual load
                if residual(t) <= 0
                    P_base(t) = max(P_base(t), 0);
                end
                % SoC-based bounds at predicted soc(t)
                soc_t = res.state_traj(t);
                [Pmin_t, Pmax_t] = obj.powerBounds(soc_t, residual(t));

                P_min(t) = Pmin_t;
                P_max(t) = Pmax_t;
            end
            % Output to home agent
            % ---------------------------------------------------------
            P_kw = P_base;

            info = struct();
            info.P_levels = P_levels;
            info.P_opt = P_opt;
            info.P_min = P_min;
            info.P_base = P_base;
            info.P_max = P_max;
            info.soc = soc_traj;
            info.dynamic_price = dynamic_price;
```

```
            info.residual_load_kw = residual; % Store disturbances for
consistent execution
            info.policy = "DP-based battery with SoC state and no-export
constraint";

            % ---- Tracking ----
            obj.P_levels_last = P_levels;
            obj.last_caps = struct( ...
                'P_min',  P_min, ...
                'P_base', P_base, ...
                'P_max',  P_max );
            obj.last_info = info;
        end

        function caps = getFlexibility(obj)
            if isempty(fieldnames(obj.last_caps))
                error("BatteryAgent.getFlexibility(): call plan() first to
compute flexibility.");
            end
            caps = obj.last_caps;
        end

        function applySetpoint(obj, P_set)
            % ============================================================
            % APPLYSETPOINT (execution-only)
            %
            % Called by Home Agent AFTER it decides targeted power for the
            % first step (following DSO request).
            %
            % - Quantizes the setpoint to admissible actions
            % - Updates internal state obj.soc0 for ONE step
            % - Shifts stored envelopes to represent remaining horizon
            % ============================================================
            if isempty(obj.P_set_traj)
                obj.P_set_traj = nan(1, obj.T);
                obj.set_idx = 0;
            end

            % Accept scalar or trajectory, execute only first element now
            if ~isscalar(P_set)
                P_set = P_set(:)';
                if isempty(P_set)
                    error("BatteryAgent.applySetpoint(): empty P_set.");
                end
                P_set = P_set(1);
            end

            % Must have a plan first (for action levels and residual at
step 1)
            if isempty(fieldnames(obj.last_info)) ||
~isfield(obj.last_info,'residual_load_kw')
                error("BatteryAgent.applySetpoint(): call plan() before
applySetpoint() so residual is known.");
            end

            residual = obj.last_info.residual_load_kw(:)';
            if isempty(residual)
                error("BatteryAgent.applySetpoint(): invalid stored
residual_load_kw.");
```

```
            end

            % Quantize to nearest admissible level
            P_exec = obj.quantizeToLevels(P_set);

            % Enforce no-export at execution time as a safety layer:
            % If there is no residual demand, do not allow discharge.
            if (residual(1) <= 1e-3) && (P_exec < 0)
                P_exec = 0;
            end

            % Store executed setpoint
            obj.set_idx = obj.set_idx + 1;
            if obj.set_idx > obj.T
                warning("BatteryAgent.applySetpoint(): received more than T
scalar setpoints; ignoring extra.");
                return;
            end
            obj.P_set_traj(obj.set_idx) = P_exec;

            % One-step SoC update
            obj.soc0 = obj.physicalModel(obj.soc0, P_exec);

            % Clamp SoC to physical bounds
            % ---------------------------------------------------------
            % obj.soc0 = min(max(obj.soc0, obj.soc_min), obj.soc_max);
            % Store executed action for traceability
            % ---------------------------------------------------------
            obj.last_info.P_exec = P_exec;
            % Re-plan for the remaining horizon (ROLLING HORIZON)
            % ---------------------------------------------------------
            obj.replanRemainingHorizon();
        end

        function Pq = quantizeToLevels(obj, P)
            % Quantize continuous setpoint to nearest discrete action
            if isempty(obj.P_levels_last)
                Pq = P;
                return;
            end
            [~, k] = min(abs(obj.P_levels_last - P));
            Pq = obj.P_levels_last(k);
        end

    end
end
```

9.1.5 Πράκτορας αντλίας

```
classdef PumpAgent < LoadAgent
    properties
        % Electrical limits
        Pmin_kw (1,1) double
        Pmax_kw (1,1) double   % max pump power [kW]

        % Hydraulic state
        % =========================================================
        V0 (1,1) double       % current tank volume [m^3]
        V_min (1,1) double     % minimum allowable volume [m^3]
```

```
        V_max (1,1) double    % maximum tank volume [m^3]

        % Physical parameters
        % =================================================================
        eta_pump (1,1) double = 0.55
        rho (1,1) double = 1000        % water density [kg/m^3]
        g (1,1) double = 9.81          % gravity [m/s^2]
        h (1,1) double = 20            % pumping head [m]

        % Discretization parameters (NEW)
        % =================================================================
        number_of_discrete_states  (1,1) double
        number_of_discrete_actions (1,1) double

        % User settings
        % =================================================================
        V_final (1,1) double          % desired water state [m^3]
        Tolerance (1,1) double          % |Volume - V_desired| < Tolerance

        % Interface book keeping
        % =================================================================
        last_info struct = struct()
        last_caps struct = struct()
        P_levels_last double = []
        P_set_traj double = []
        set_idx (1,1) double {mustBeInteger, mustBeNonnegative} = 0
    end

    methods (Access = private)
        function V_next = physicalModel(obj, V, P_exec, V_out)
            % One-step hydraulic state transition
            % V      : current tank volume [m^3]
            % P_exec : executed pump power [kW]
            % V_out  : water outflow during this step [m^3]

            dt = obj.dt_hours * 3600;   % [s]

            V_next = V + ...
                obj.eta_pump * (P_exec * 1000) * dt / ...
                (obj.rho * obj.g * obj.h) ...
                - V_out;
        end
        function v = terminalCost(obj, V)
            if abs(V - obj.V_final) <= obj.Tolerance
                v = 0;
            else
                v = realmax;  % or Inf
            end
        end
        function [Pmin, Pmax] = powerBounds(obj, V_t, V_out_t)
            % Compute feasible power bounds for ONE step
            % based on tank volume dynamics and physical limits
            %
            % Output:
            %   Pmin <= P <= Pmax   (battery-side power, kW)

            dt = obj.dt_hours * 3600; % [s]

            % Volume-based bounds
```

```
            Pmin_vol = ( ...
                obj.V_min - V_t + V_out_t ) * obj.rho * obj.g * obj.h ...
                / (obj.eta_pump * 1000 * dt);

            Pmax_vol = ( ...
                obj.V_max - V_t + V_out_t ) * obj.rho * obj.g * obj.h ...
                / (obj.eta_pump * 1000 * dt);

            % Combine with electrical limits
            Pmin = max(obj.Pmin_kw, Pmin_vol);
            Pmax = min(obj.Pmax_kw, Pmax_vol);
        end
        function replanRemainingHorizon(obj)
            % Re-solve DP for the remaining horizon starting
            % from the CURRENT real state obj.V0

            T_remain = obj.T - obj.set_idx;
            if T_remain <= 0
                return;
            end

            % Shift stored signals
            dynamic_price = obj.last_info.dynamic_price(obj.set_idx+1:end);
            V_out         = obj.last_info.V_out_pump(obj.set_idx+1:end);

            % Discretization (reuse last action levels!)
            P_levels = obj.P_levels_last;
            V_grid   = linspace(obj.V_min, obj.V_max,
obj.number_of_discrete_states);

            % Dynamics
            f = @(V, P, t) obj.physicalModel(V, P, V_out(t));

            % Cost
            l  = @(V, P, t) dynamic_price(t) * P;
            lT = @(V) obj.terminalCost(V);

            feasible = @(V_next, V, P, t) ...
                (V_next >= obj.V_min) && ...
                (V_next <= obj.V_max);

            solver = DynamicProgrammingSolver( ...
                T_remain, V_grid, P_levels, f, l, lT, feasible);
            solver.strict_rollout = true;
            res = solver.solve(obj.V0);

            % Build new caps for remaining horizon
            nA = numel(P_levels);
            P_base = zeros(1, T_remain);
            P_min  = zeros(1, T_remain);
            P_max  = zeros(1, T_remain);

            for t = 1:T_remain
                idx = find(P_levels >= res.action_traj(t) - 1e-9, 1,
'first');
                if isempty(idx), idx = nA; end
                P_base(t) = P_levels(idx);

                % Volume-based physical bounds
```

```
            V_t = res.state_traj(t);
            Vd  = V_out(t);
            [Pmin_t, Pmax_t] = obj.powerBounds(V_t, Vd);

            P_min(t) = Pmin_t;
            P_max(t) = Pmax_t;
        end

        obj.last_caps = struct( ...
            'P_min',  P_min, ...
            'P_base', P_base, ...
            'P_max',  P_max );

        % Update info for debugging/plots
        obj.last_info.P_opt  = res.action_traj;
        obj.last_info.volume = res.state_traj;
    end
end

methods
    function obj = PumpAgent(T, dt_hours, params)
        % Constructor: initializes horizon and optional parameters
        obj@LoadAgent(T, dt_hours);
        if nargin >= 3
            f = fieldnames(params);
            for i = 1:numel(f)
                if isprop(obj, f{i})
                    obj.(f{i}) = params.(f{i});
                end
            end
        end
    end

    function label = name(obj)
        label = "Pump";
    end

    function [P_kw, info] = plan(obj, signals)
        % ============================================================
        % PLAN (proposal-only)
        %
        % - Solves DP from CURRENT volume state obj.V0 over horizon T.
        % - DOES NOT execute any action and DOES NOT update obj.V0.
        % - Exposes baseline and flexibility envelope for horizon T.
        %
        % Execution happens ONLY in applySetpoint().
        % ============================================================

        T = obj.T;

        % Disturbance: water outflow from tank
        % ------------------------------------------------------------
        if isfield(signals,'V_out_pump')
            V_out = signals.V_out_pump(:)';
        else
            V_out = zeros(1,T);
        end
        if numel(V_out) ~= T
```

```
                error("PumpAgent.plan(): signals.V_out_pump must have
length T (or be omitted).");
            end

            % ---- Price signal:
            % Dynamic energy pricing signal (exogenous policy)
            % time-varying electricity tariff (e.g., TOU / RTP) used in the
cost function.
            % -----------------------------------------------------------
            dynamic_price = signals.dynamic_price(:)';
            dynamic_price = dynamic_price / (max(dynamic_price) + eps);

            % Discretization
            % -----------------------------------------------------------
            % Discrete power actions
            P_levels = linspace( ...
                obj.Pmin_kw, ...
                obj.Pmax_kw, ...
                obj.number_of_discrete_actions);
            % Discrete volume states
            V_grid = linspace( ...
                obj.V_min , ...
                obj.V_max, ...
                obj.number_of_discrete_states);

            %  Hydraulic dynamics
            % -----------------------------------------------------------
            f = @(V, P, t) ...
                obj.physicalModel(V, P, V_out(t));

            % Stage cost
            % -----------------------------------------------------------
            l = @(V, P, t) dynamic_price(t) * P;

            % Terminal cost: penalty to enforce volume sufficiency
            % -----------------------------------------------------------
            lT = @(V) obj.terminalCost(V);

            feasible = @(V_next, V, P, t) ...
                (V_next >= obj.V_min) && ...
                (V_next <= obj.V_max);

            % Solve DP from CURRENT volume state
            % -----------------------------------------------------------
            solver = DynamicProgrammingSolver( ...
                T, V_grid, P_levels, f, l, lT, feasible);
            solver.strict_rollout = true;
            res = solver.solve(obj.V0);

            P_opt = res.action_traj;
            V_traj = res.state_traj;

            % Build (Pmin, Pbase, Pmax) profiles for remaining horizon
            nA = numel(P_levels);
            P_base = zeros(1,T);
            P_min  = zeros(1,T);
            P_max  = zeros(1,T);
            for t = 1:T
                idx = find(P_levels >= P_opt(t) - 1e-9, 1, 'first');
```

```
                if isempty(idx), idx = nA; end
                P_base(t) = P_levels(idx);

                V_t = V_traj(t);
                Vd  = V_out(t);
                [Pmin_t, Pmax_t] = obj.powerBounds(V_t, Vd);

                P_min(t) = Pmin_t;
                P_max(t) = Pmax_t;
            end

            % ---------------------------------------------------------
            % Output to home agent: baseline proposal for horizon
            % ---------------------------------------------------------
            P_kw = P_base;

            info = struct();
            info.P_levels = P_levels;
            info.P_opt = P_opt;
            info.P_min = P_min;
            info.P_base = P_base;
            info.P_max = P_max;
            info.volume = V_traj;
            info.V_out_pump = V_out; % store disturbance for consistent
execution
            info.dynamic_price = dynamic_price;
            info.policy = "DP-based hydraulic pump with volume state";

            % ---- Tracking ----
            obj.P_levels_last = P_levels;
            obj.last_caps = struct( ...
                'P_min',  P_min, ...
                'P_base', P_base, ...
                'P_max',  P_max );
            obj.last_info = info;
        end

        function caps = getFlexibility(obj)
            % Returns last stored envelope.
            % After plan(): full horizon.
            % After applySetpoint(): remaining horizon (shifted).
            if isempty(fieldnames(obj.last_caps))
                error("PumpAgent.getFlexibility(): call plan() first to
compute flexibility.");
            end
            caps = obj.last_caps;
        end

        function applySetpoint(obj, P_set)
            % =========================================================
            % APPLYSETPOINT (execution-only)
            %
            % Called by HomeAgent AFTER it decides targeted power for
            % the first step (e.g., following a DSO request).
            %
            % - Quantizes setpoint to admissible actions
            % - Updates internal state obj.V0 for ONE step
            % - Shifts stored envelopes to represent remaining horizon
            % =========================================================
```

```
            % Initialize tracking
            if isempty(obj.P_set_traj)
                obj.P_set_traj = nan(1, obj.T);
                obj.set_idx = 0;
            end
            % Support scalar or trajectory (apply only first now)
            if ~isscalar(P_set)
                P_set = P_set(:)';
                if isempty(P_set)
                    error("PumpAgent.applySetpoint(): empty P_set.");
                end
                P_set = P_set(1);
            end

            % Quantize to nearest discrete power level
            P_exec = obj.quantizeToLevels(P_set);

            % Store executed setpoint
            obj.set_idx = obj.set_idx + 1;
            if obj.set_idx > obj.T
                warning("PumpAgent.applySetpoint(): received more than T
scalar setpoints; ignoring extra.");
                return;
            end
            obj.P_set_traj(obj.set_idx) = P_exec;

            % Ensure plan() has been called so we have the disturbance used
in planning
            if isempty(fieldnames(obj.last_info)) ||
~isfield(obj.last_info,'V_out_pump')
                error("PumpAgent.applySetpoint(): call plan() before
applySetpoint() so disturbance is known.");
            end
            V_out = obj.last_info.V_out_pump(:)';
            if numel(V_out) < 1
                error("PumpAgent.applySetpoint(): invalid stored
V_out_pump.");
            end

            % One-step state update (t = 1 relative to current horizon)
            obj.V0 = obj.physicalModel(obj.V0, P_exec, V_out(1));
            % Store executed action for traceability
            obj.last_info.P_exec = P_exec;
            % Re-plan for the remaining horizon (ROLLING HORIZON)
            % ------------------------------------------------------------
            obj.replanRemainingHorizon();
        end

        function Pq = quantizeToLevels(obj, P)
            % Quantize continuous setpoint to nearest discrete action
            if isempty(obj.P_levels_last)
                Pq = P; % fallback
                return;
            end
            [~, k] = min(abs(obj.P_levels_last - P));
            Pq = obj.P_levels_last(k);
        end
    end
```

```
end
```

9.1.6 Πράκτορας οικίας

```
classdef HouseAgent < handle
    %HOUSEAGENT
    % Central coordinator for a household multi-agent EMS.
    %
    % Responsibilities:
    %   (i)   Call plan() on all agents (proposal-only)
    %   (ii)  Aggregate household baseline and flexibility bounds
    %   (iii) Allocate a CURRENT-STEP external target power to agents
    %         and trigger execution via applySetpoint() (scalar only)
    %
    % Notes:
    % - HouseAgent does NOT solve load dynamics (DP stays inside each
load).
    % - Physical execution happens ONLY in applySetpoint() of each agent.

    properties
        T (1,1) double
        dt_hours (1,1) double

        agents   % cell array of LoadAgent objects

        % Latest per-agent outputs (from plan)
        labels    string
        P_each    double    % [nA x T] baseline proposal per agent
        infos     cell      % {nA} info structs returned by each agent
        caps_each cell         % {nA x 1} each contains struct with
P_min/P_base/P_max

        % Aggregated household profiles (proposal)
        P_base_house  double  % [1 x T]
        P_min_house   double  % [1 x T]
        P_max_house   double  % [1 x T]

        planned_flag (1,1) logical = false
    end

    methods
        function obj = HouseAgent(agents, T, dt_hours)
            obj.agents = agents;
            obj.T = T;
            obj.dt_hours = dt_hours;

            obj.reset();
        end

        % ================================================================
        % Reset stored planning results
        % ================================================================
        function reset(obj)
            nA = numel(obj.agents);

            obj.labels = strings(nA,1);
            obj.P_each = nan(nA, obj.T);
            obj.infos  = cell(nA,1);
```

```
            obj.caps_each = cell(nA,1);

            obj.P_base_house = zeros(1, obj.T);
            obj.P_min_house  = zeros(1, obj.T);
            obj.P_max_house  = zeros(1, obj.T);

            obj.planned_flag = false;
        end


        % ================================================================
        % Planning (loads first -> residual -> battery -> aggregate all)
        % ================================================================
        function [P_total_base, infos, labels, P_each] = plan(obj, signals)
            % Plans all agents with correct coupling for the battery.
            %
            % Output:
            %   P_total_base: [1xT] baseline household (loads+battery)
            %   infos, labels, P_each: per-agent outputs

            obj.reset();
            nA = numel(obj.agents);


            % Identify battery indices by class
            is_battery = false(nA,1);
            for i = 1:nA
                is_battery(i) = isa(obj.agents{i}, 'BatteryAgent');
            end
            idx_nonbat = find(~is_battery);
            idx_bat    = find(is_battery);

            % ----------------------------
            % (1) Plan NON-battery loads (proposal-only)
            % ----------------------------
            P_loads_nonbat = zeros(1, obj.T);

            for k = 1:numel(idx_nonbat)
                i = idx_nonbat(k);

                obj.labels(i) = obj.agents{i}.name(); % Store label
                [P_kw, info]  = obj.agents{i}.plan(signals);  % Proposal-
only plan (agent does NOT update its state here)

                obj.P_each(i,:) = P_kw; % Cache baseline and info
                obj.infos{i} = info;

                % Cache caps from agent interface
                obj.caps_each{i} = obj.agents{i}.getFlexibility();

                % Residual demand seen by the battery
                P_loads_nonbat = P_loads_nonbat + obj.P_each(i,:);
            end

            % ----------------------------
            % (2) Plan BATTERY (needs residual_load_kw)
            % ----------------------------
            if ~isempty(idx_bat)
                signals_b = signals;
```

```
                signals_b.residual_load_kw = P_loads_nonbat;

                for k = 1:numel(idx_bat)
                    i = idx_bat(k);

                    obj.labels(i) = obj.agents{i}.name();
                    [P_kw, info]  = obj.agents{i}.plan(signals_b);

                    obj.P_each(i,:) = P_kw;
                    obj.infos{i} = info;
                    obj.caps_each{i} = obj.agents{i}.getFlexibility();
                end
            end

            % ----------------------------
            % (3) Aggregate household flexibility (INCLUDING battery)
            % ----------------------------
            obj.P_base_house(:) = 0;
            obj.P_min_house(:)  = 0;
            obj.P_max_house(:)  = 0;
            %{
            for i = 1:nA
                info = obj.infos{i};
                % Use advertised caps if available
                if isfield(info,'P_base') && isfield(info,'P_min') &&
isfield(info,'P_max')
                    obj.P_base_house = obj.P_base_house + info.P_base;
                    obj.P_min_house  = obj.P_min_house  + info.P_min;
                    obj.P_max_house  = obj.P_max_house  + info.P_max;
                else
                    % Fallback: treat as fixed baseline
                    obj.P_base_house = obj.P_base_house + obj.P_each(i,:);
                    obj.P_min_house  = obj.P_min_house  + obj.P_each(i,:);
                    obj.P_max_house  = obj.P_max_house  + obj.P_each(i,:);
                end
            end
            %}
            for i = 1:nA
                caps = obj.caps_each{i};
                if isempty(caps) || ~isfield(caps,'P_base') ||
~isfield(caps,'P_min') || ~isfield(caps,'P_max')
                    error("HouseAgent.plan(): agent %d did not provide caps
via getFlexibility().", i);
                end
                obj.P_base_house = obj.P_base_house + caps.P_base;
                obj.P_min_house  = obj.P_min_house  + caps.P_min;
                obj.P_max_house  = obj.P_max_house  + caps.P_max;
            end
            % Baseline household proposal (includes battery baseline)
            P_total_base = obj.P_base_house;
            % Mark planning complete
            obj.planned_flag = true;
            % Return for convenience
            infos      = obj.infos;
            labels     = obj.labels;
            P_each     = obj.P_each;
        end

        % ================================================================
```

```
        % Household flexibility bid (proposal)
        % =================================================================
        function bid = getFlexibilityBid(obj)
            obj.assertPlanned();

            bid = struct();
            bid.P_base = obj.P_base_house;
            bid.P_min  = obj.P_min_house;
            bid.P_max  = obj.P_max_house;

            bid.DP_min = obj.P_min_house - obj.P_base_house; % <= 0
            bid.DP_max = obj.P_max_house - obj.P_base_house; % >= 0
        end

        % =================================================================
        % Allocate & EXECUTE for the CURRENT STEP ONLY (t = 1)
        % (Quadratic Programming formulation)
        % =================================================================
        function alloc = allocateTarget(obj, P_target)
            % This function allocates a household-level power target for
the
            % CURRENT time step (t = 1) among the individual load agents.
            %
            % The allocation problem is formulated as a Quadratic Program
(QP):
            %
            %   minimize    sum_i (ΔP_i)^2
            %
            %   subject to  sum_i ΔP_i = ΔP_house_target
            %               ΔP_min_i ≤ ΔP_i ≤ ΔP_max_i
            %
            % where:
            %   ΔP_i        : power deviation of load i from its baseline
            %   ΔP_house_target = P_target - P_base_house
            %
            % After solving the QP, the resulting setpoints are:
            %   P_set_i = P_base_i + ΔP_i
            %
            % Only the CURRENT step is executed (rolling horizon control).

            obj.assertPlanned();
            nA = numel(obj.agents);

            % =============================================================
            % (1) Extract per-agent baseline and bounds at current step
(t=1)
            % =============================================================
            P_base_1 = zeros(nA,1);
            P_min_1  = zeros(nA,1);
            P_max_1  = zeros(nA,1);

            for i = 1:nA
                caps = obj.caps_each{i};
                P_base_1(i) = caps.P_base(1);
                P_min_1(i)  = caps.P_min(1);
                P_max_1(i)  = caps.P_max(1);
            end

            % Household baseline at current step
```

```
            P_base_house_1 = sum(P_base_1);

            % Requested household-level deviation
            dP_house_target = P_target - P_base_house_1;

            % ============================================================
            % (2) Define decision variables and bounds
            % ============================================================
            % Decision variables:
            %   x_i = ΔP_i = P_set_i - P_base_i
            %
            % Lower and upper bounds on deviations:
            dP_min = P_min_1 - P_base_1;   % ≤ 0
            dP_max = P_max_1 - P_base_1;   % ≥ 0

            % ============================================================
            % (3) Quadratic Programming formulation
            % ============================================================
            % Objective:
            %   minimize (1/2) * x' * H * x
            %
            % where H = 2 * I ensures:
            %   (1/2) * x' * (2I) * x = sum_i (ΔP_i)^2
            H = 2 * eye(nA);
            f = zeros(nA,1);   % no linear term

            % Equality constraint: sum_i ΔP_i = ΔP_house_target
            Aeq = ones(1,nA);
            beq = dP_house_target;

            % No inequality constraints beyond box bounds
            A = [];
            b = [];

            % ============================================================
            % (4) Solve QP
            % ============================================================
            options = optimoptions('quadprog', ...
                'Display', 'off', ...
                'Algorithm', 'interior-point-convex');

            [dP_opt, ~, exitflag] = quadprog( ...
                H, f, A, b, Aeq, beq, dP_min, dP_max, [], options);

            % If the QP is infeasible, fall back to saturation of the
target
            if exitflag <= 0
                warning("HouseAgent: QP allocation infeasible at t=1.
Saturating target.");

                % Project target to feasible interval
                dP_house_feasible = min( ...
                    max(dP_house_target, sum(dP_min)), ...
                    sum(dP_max));

                beq = dP_house_feasible;

                [dP_opt, ~, exitflag] = quadprog( ...
                    H, f, A, b, Aeq, beq, dP_min, dP_max, [], options);
```

```
                if exitflag <= 0
                    error("HouseAgent: QP allocation failed even after
saturation.");
                end
            end

            % ============================================================
            % (5) Compute final setpoints
            % ============================================================
            P_set_1 = P_base_1 + dP_opt;

            % ============================================================
            % (6) Enforce NO-EXPORT constraint at household level
            % ============================================================
            % If total power becomes negative, reduce battery discharge.
            bat_idx = 0;
            for i = 1:nA
                if isa(obj.agents{i}, 'BatteryAgent')
                    bat_idx = i;
                    break;
                end
            end

            if bat_idx ~= 0
                P_total_1 = sum(P_set_1);

                if P_total_1 < 0
                    % Increase battery power to eliminate export
                    P_set_1(bat_idx) = P_set_1(bat_idx) - P_total_1;

                    % Clamp to battery bounds
                    P_set_1(bat_idx) = min(P_set_1(bat_idx),
P_max_1(bat_idx));
                    P_set_1(bat_idx) = max(P_set_1(bat_idx),
P_min_1(bat_idx));

                    if sum(P_set_1) < -1e-6
                        warning("HouseAgent: NO-EXPORT violated at t=1
(battery saturated).");
                    end
                end
            else
                if sum(P_set_1) < -1e-6
                    warning("HouseAgent: Export detected but no battery is
present.");
                end
            end

            % ============================================================
            % (7) EXECUTE current step
            % ============================================================
            for i = 1:nA
                obj.agents{i}.applySetpoint(P_set_1(i));
            end

            % ============================================================
            % (8) Output structure (unchanged interface)
            % ============================================================
```

```
            alloc = struct();
            alloc.P_target            = P_target;
            alloc.P_base_house_now    = P_base_house_1;
            alloc.P_set_each_now      = P_set_1;
            alloc.P_total_set_now     = sum(P_set_1);
            alloc.labels              = obj.labels;
        end

    end

    methods (Access = private)
        function assertPlanned(obj)
            if isempty(obj.infos) || any(cellfun(@isempty, obj.infos)) ||
any(isnan(obj.P_each(:)))
                error("HouseAgent: call plan(signals) before
allocation/execution.");
            end
        end
    end
end
```

9.2 Κώδικας επίλυσης Οικονομικής Κατανομής Φορτίου

9.2.1 Αρχείο επίλυσης οικονομικής κατανομής φορτίου

```
function out = solve_ed_microgrid_qp(data)
%   MILP for microgrid economic dispatch with:
%   - absolute household power as decision variable
%   - MT quadratic cost linearized via piecewise-linear (convex-
combination) model
%   - MT commitment variable u_MT(t) to account for fixed activation cost
c_MT (paid only if ON)

    T = data.T;

    % =====================================================================
    % PIECEWISE-LINEAR SETTINGS
    % =====================================================================
    % Number of breakpoints (K) => segments = K-1
    % Tradeoff: larger K => more accurate, slower MILP
    K = 8;

    % Breakpoints across MT range. Include 0 explicitly to allow "OFF".
    % We use [0, ..., P_MT_max].
    Pk = linspace(0, data.P_MT_max, K)';                    % [K x 1]
    Ck = data.a_MT * (Pk.^2) + data.b_MT * Pk;              % variable cost
at breakpoints

    % =====================================================================
    % VARIABLE INDEXING
    % =====================================================================
    % Variables:
    %   P_MT, P_PV, P_grid, P_house, s_abs
    %
    % Add:
    %   u_MT(t)    binary commitment (ON/OFF)
    %   lambda(k,t) convex-combination weights for P_MT and cost
    %
```

```
% Total vars:
%   5T + T + K*T = (6+K)*T
nVar_base = 5*T;
nVar = nVar_base + T + K*T;

idx.P_MT    = 1:T;
idx.P_PV    = T+1:2*T;
idx.P_grid  = 2*T+1:3*T;
idx.P_house = 3*T+1:4*T;
idx.s_abs   = 4*T+1:5*T;

idx.u_MT    = nVar_base + (1:T);                  % binary
idx.lam     = (nVar_base + T) + (1:K*T);          % continuous >=0

% Helper to access lambda(k,t) in the flat vector:
lam_index = @(k,t) idx.lam( (t-1)*K + k );

% ================================================================
% COST FUNCTION (linear):  f'*x
% ================================================================
f = zeros(nVar,1);

% MT linearized variable cost via lambdas: sum_k Ck * lambda(k,t)
for t = 1:T
    for k = 1:K
        f(lam_index(k,t)) = Ck(k);
    end
end

% MT fixed activation cost: c_MT * u_MT(t)
% If c_MT is scalar: ok. If it is [T x 1], also ok.
cMT = data.c_MT;
if isscalar(cMT)
    f(idx.u_MT) = cMT * ones(T,1);
else
    f(idx.u_MT) = cMT(:);
end

% Grid cost and flexibility deviation cost
f(idx.P_grid) = data.c_grid(:);
f(idx.s_abs)  = data.c_flex(:);

% ================================================================
% EQUALITY CONSTRAINTS  Aeq*x = beq
% ================================================================
% (1) Power balance:
%   P_MT + P_PV + P_grid = L_inflex + P_house
Aeq_pb = zeros(T, nVar);
beq_pb = data.L_inflex(:);

for t = 1:T
    Aeq_pb(t, idx.P_MT(t))    = 1;
    Aeq_pb(t, idx.P_PV(t))    = 1;
    Aeq_pb(t, idx.P_grid(t))  = 1;
    Aeq_pb(t, idx.P_house(t)) = -1;
end

% (2) Link P_MT(t) to lambdas:
%   P_MT(t) = sum_k Pk(k) * lambda(k,t)
```

```
Aeq_linkP = zeros(T, nVar);
beq_linkP = zeros(T, 1);

for t = 1:T
    Aeq_linkP(t, idx.P_MT(t)) = 1;
    for k = 1:K
        Aeq_linkP(t, lam_index(k,t)) = -Pk(k);
    end
end

% (3) Convex-combination "on/off":
%   sum_k lambda(k,t) = u_MT(t)
Aeq_lamsum = zeros(T, nVar);
beq_lamsum = zeros(T, 1);

for t = 1:T
    Aeq_lamsum(t, idx.u_MT(t)) = -1;
    for k = 1:K
        Aeq_lamsum(t, lam_index(k,t)) = 1;
    end
end

Aeq = [Aeq_pb; Aeq_linkP; Aeq_lamsum];
beq = [beq_pb; beq_linkP; beq_lamsum];

% ================================================================
% INEQUALITY CONSTRAINTS  A*x <= b
% ================================================================
% |P_house - P_base| <= s_abs (same as before)
A = zeros(2*T, nVar);
b = zeros(2*T, 1);

for t = 1:T
    %  P_house - P_base <= s_abs  ->  P_house - s_abs <= P_base
    A(2*t-1, idx.P_house(t)) =  1;
    A(2*t-1, idx.s_abs(t))   = -1;
    b(2*t-1)                 =  data.P_base(t);

    % -(P_house - P_base) <= s_abs -> -P_house - s_abs <= -P_base
    A(2*t,   idx.P_house(t)) = -1;
    A(2*t,   idx.s_abs(t))   = -1;
    b(2*t)                   = -data.P_base(t);
end

% Optional: enforce MT min output when ON:
%   P_MT(t) >= P_MT_min * u_MT(t)
if isfield(data, 'P_MT_min') && any(data.P_MT_min(:) > 0)
    A_minON = zeros(T, nVar);
    b_minON = zeros(T, 1);
    for t = 1:T
        % -P_MT + P_MT_min*u <= 0  ->  P_MT >= P_MT_min*u
        A_minON(t, idx.P_MT(t))  = -1;
        A_minON(t, idx.u_MT(t))  =  data.P_MT_min;
        b_minON(t)               =  0;
    end
    A = [A; A_minON];
    b = [b; b_minON];
end
```

```
    % ================================================================
    % BOUNDS
    % ================================================================
    lb = -inf(nVar,1);
    ub =  inf(nVar,1);
    % MT bounds: allow OFF by setting lower bound to 0 (since u handles
ON/OFF)
    lb(idx.P_MT) = 0;
    ub(idx.P_MT) = data.P_MT_max;
    % PV bounds
    lb(idx.P_PV) = 0;
    ub(idx.P_PV) = data.Ppv_max(:);
    % Grid import/export capability
    lb(idx.P_grid) = -data.P_import;
    ub(idx.P_grid) = +data.P_import;
    % Absolute house power bounds
    lb(idx.P_house) = data.P_min(:);
    ub(idx.P_house) = data.P_max(:);
    % Slack variable
    lb(idx.s_abs) = 0;
    % Lambda bounds
    lb(idx.lam) = 0;
    % Binary u bounds
    lb(idx.u_MT) = 0;
    ub(idx.u_MT) = 1;

    % ================================================================
    % INTEGRAL CONSTRAINTS
    % ================================================================
    intcon = idx.u_MT;  % only u_MT is integer (binary)

    % ================================================================
    % SOLVER (MILP)
    % ================================================================
    options = optimoptions('intlinprog', ...
        'Display','none');

    [x, J_opt, exitflag] = intlinprog(f, intcon, A, b, Aeq, beq, lb, ub,
options);

    % ================================================================
    % OUTPUT
    % ================================================================
    out.P_MT     = x(idx.P_MT);
    out.P_PV     = x(idx.P_PV);
    out.P_grid   = x(idx.P_grid);
    out.P_house  = x(idx.P_house);
    out.s_abs    = x(idx.s_abs);
    out.J_opt    = J_opt;
    out.exitflag = exitflag;

end
```

9.3 Κύριο αρχείο προσομοίωσης οικιακού πολυπρακτορικού συστήματος

```
clear all; clc;
close all;

%% PATH SETUP
% ========================================================================

thisFile = mfilename('fullpath');
thisDir  = fileparts(thisFile);
rootDir  = fileparts(thisDir);

addpath(genpath(rootDir));
rehash path

%% LOAD SCENARIO
% ========================================================================
% s = 0 → base scenario
% s = 1 → dynamic pricing case
% s = 2 → fixed pricing case

s = 1;   % <-- change only this

if s == 0
    scenario_name = "scenario_consumption";
elseif s == 1
    scenario_name = "scenario_consumption_dynamic_price_12h";
elseif s == 2
    scenario_name = "scenario_consumption_fixed_price_12h";
end
S = feval(scenario_name);

% SIMULATION HORIZON
T        = S.T;         % planning horizon steps seen by each agent
dt_hours = S.dt_hours;  % 15 minutes per step
N_sim    = S.N_sim;     % number of REAL executed steps (can be different)
% INPUT SIGNALS (exogenous)
signals_all = S.signals;
cfg         = S.cfg;


%% BUILD AGENTS
% ========================================================================
agents = build_agents(cfg, T, dt_hours);

%% HOUSE AGENT (orchestrates proposal -> step execution)
% ========================================================================
house = HouseAgent(agents, T, dt_hours);

% LOGGING ARRAYS (executed step-by-step)
% ========================================================================
nA = numel(agents);
labels = strings(nA,1);
for i = 1:nA
    labels(i) = agents{i}.name();
end
```

```
P_each_exec   = nan(nA, N_sim);    % executed power per agent per real step
P_total_exec  = nan(1,  N_sim);    % executed household power

P_base_now    = nan(1,  N_sim);    % household baseline at current step
P_min_now     = nan(1,  N_sim);    % household min at current step
P_max_now     = nan(1,  N_sim);    % household max at current step
P_target_now  = nan(1,  N_sim);    % DSO target (current step)

% Log internal states by reading agent properties (reliable)
soc_log = nan(1, N_sim+1);
V_log   = nan(1, N_sim+1);
Tw_log  = nan(1, N_sim+1);

% Initial states
bat_idx  = find(labels == "Battery", 1);
pump_idx = find(labels == "Pump",    1);
ewh_idx  = find(labels == "EWH",     1);

if ~isempty(bat_idx),  soc_log(1) = agents{bat_idx}.soc0; end
if ~isempty(pump_idx), V_log(1)   = agents{pump_idx}.V0;  end
if ~isempty(ewh_idx),  Tw_log(1)  = agents{ewh_idx}.T0;   end


%% ================================================================
% INITIAL PROPOSAL (FULL HORIZON, NO EXECUTION)
% ================================================================
sig_dayAhead = struct();
sig_dayAhead.dynamic_price = pad_to_T(signals_all.dynamic_price(1:end), T);
sig_dayAhead.V_out_pump    = pad_to_T(signals_all.V_out_pump(1:end), T);
sig_dayAhead.V_out_EWH    = pad_to_T(signals_all.V_out_EWH(1:end), T);

house.plan(sig_dayAhead);
bid_dayAhead = house.getFlexibilityBid();

P_base_house_0 = bid_dayAhead.P_base; % house-level planned
P_min_house_0  = bid_dayAhead.P_min;
P_max_house_0  = bid_dayAhead.P_max;

P_each_base_0  = house.P_each;         % per-agent baseline
infos_0        = house.infos;          % DP info for plotting

% DAY-AHEAD PLANNED CONSUMPTION (REFERENCE)
% ================================================================
P_house_planned = P_base_house_0;    % [1 x T] house-level planned
P_load_planned  = P_each_base_0;     % [nLoads x T] load-level planned


%% Intra-day snapshots configuration
% ------------------------------------------------------------
snap_steps = unique([ ...
    1:3, ...                    % first 5 real-time steps
    (N_sim-1):N_sim ...         % last 3 real-time steps
]);
snapshots = struct();                          % will store results
snap_count = 0;


%% ========================================================================
```

```
% ROLLING-HORIZON LOOP
% =========================================================================
for k = 1:N_sim

    % ---------------------------------------------------------
    % Build horizon signals window for current time step k
    % Each agent sees a horizon of length T (pad last value if required).
    % ---------------------------------------------------------
    sig = struct();
    sig.dynamic_price = pad_to_T(signals_all.dynamic_price(k:end), T);
    sig.V_out_pump = pad_to_T(signals_all.V_out_pump(k:end), T);
    sig.V_out_EWH = pad_to_T(signals_all.V_out_EWH(k:end), T);

    % ---------------------------------------------------------
    % (1) Proposal stage: plan all agents (proposal-only)
    % ---------------------------------------------------------
    house.plan(sig);

    % Household flexibility at current step (t=1 of the horizon)
    bid = house.getFlexibilityBid();
    P_base_now(k) = bid.P_base(1);
    P_min_now(k)  = bid.P_min(1);
    P_max_now(k)  = bid.P_max(1);

    % ---------------------------------------------------------
    % (2) GRID OPERATOR sends TARGETED CONSUMPTION for current step ONLY
    % Controlled variation within ±30% of baseline
    % ---------------------------------------------------------
    omega = 2*pi / N_sim;
    variation_factor = 0.30 * sin(omega * k);
    P_desired = P_base_now(k) * (1 + variation_factor);
    P_target_now(k) = min( max(P_desired, P_min_now(k)), P_max_now(k) );


    % ---------------------------------------------------------
    % (3) Allocation + execution for current step ONLY
    % ---------------------------------------------------------
    alloc = house.allocateTarget(P_target_now(k));

    % Log executed powers
    P_each_exec(:,k)  = alloc.P_set_each_now;
    P_total_exec(k)   = alloc.P_total_set_now;

    % ---------------------------------------------------------
    % (4) Log states after execution
    % ---------------------------------------------------------
    if ~isempty(bat_idx),  soc_log(k+1) = agents{bat_idx}.soc0; end
    if ~isempty(pump_idx), V_log(k+1)   = agents{pump_idx}.V0;  end
    if ~isempty(ewh_idx),  Tw_log(k+1)  = agents{ewh_idx}.T0;   end

    % ===========================================================
    % SNAPSHOT FOR INTRA-DAY COMPARISON
    % ===========================================================
    if ismember(k, snap_steps)
        snap_count = snap_count + 1;

        bid_k = house.getFlexibilityBid();

        snapshots(snap_count).k = k;
```

```
        snapshots(snap_count).time_offset = (k-1) * dt_hours;

        snapshots(snap_count).P_base_house = bid_k.P_base;
        snapshots(snap_count).P_min_house  = bid_k.P_min;
        snapshots(snap_count).P_max_house  = bid_k.P_max;

        snapshots(snap_count).P_each_base = house.P_each;
        snapshots(snap_count).infos = house.infos;

        % Store real states after execution
        if ~isempty(bat_idx),  snapshots(snap_count).soc =
agents{bat_idx}.soc0; end
        if ~isempty(pump_idx), snapshots(snap_count).V   =
agents{pump_idx}.V0;  end
        if ~isempty(ewh_idx),  snapshots(snap_count).Tw  =
agents{ewh_idx}.T0;   end
    end
end

%% FACTORY FUNCTION
% =========================================================================
function agents = build_agents(cfg, T, dt_hours)

    agents = {};

    for k = 1:numel(cfg.enabled)
        key = cfg.enabled(k);

        params = struct();
        if isfield(cfg.params, key)
            params = cfg.params.(key);
        end

        switch key
            case "EWH"
                agents{end+1} = EWHAgent(T, dt_hours, params);
            case "Pump"
                agents{end+1} = PumpAgent(T, dt_hours, params);
            case "Battery"
                agents{end+1} = BatteryAgent(T, dt_hours, params);
            otherwise
                error("Unknown or excluded agent type: %s", key);
        end
    end
end
```