

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Υλοποίηση διάταξης αυτοματισμού για ανελκυστήρες  
με Χρήση PLC»



Του φοιτητή: Ευθύμιου Τσάκου  
Αρ. Μητρώου: 516137

Επιβλέπων  
Όνοματεπώνυμο: Κιοσκερίδης  
Ιορδάνης  
Βαθμίδα : Καθηγητής

Ημερομηνία 29/05/2026

Τίτλος Π.Ε: Υλοποίηση διάταξης αυτοματισμού για ανελκυστήρες με χρήση PLC  
Κωδικός Π.Ε. 23290

Όνοματεπώνυμο φοιτητή: Ευθύμιος Τσάκος  
Όνοματεπώνυμο εισηγητή: Κιοσκερίδης Ιορδάνης  
Ημερομηνία ανάληψης Π.Ε. 21/01/2026  
Ημερομηνία περάτωσης Π.Ε. 29/05/2026

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Ευθύμιου Τσάκου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

*«Στα αδέρφια και τους γονείς μου»*

## **Πρόλογος**

Το ενδιαφέρον μου για την τεχνολογία και τους αυτοματισμούς ανελκυστήρων με οδήγησε στο θέμα της πτυχιακής. Ένας επιπλέον λόγος της επιλογής μου ήταν το γεγονός πως έκανα την πρακτική μου άσκηση σε εταιρεία που δραστηριοποιείται στο χώρο των αυτοματισμών ανελκυστήρων. Στόχος μου, είναι να υλοποιήσω ένα σύστημα αυτοματισμού για την λειτουργία ενός ανελκυστήρα και η μελλοντική μου ενασχόλησή με αυτοματισμούς ανελκυστήρων η και βιομηχανικών συστημάτων αυτοματισμού.

Η επιλογή ενός PLC για την εκπόνηση της εργασίας έγινε με κριτήριο την εφαρμογή του συγκεκριμένου ελεγκτή σαν το επικρατέστερο σύστημα ελέγχου και αυτοματισμών αυτή την στιγμή στην βιομηχανία.

## Περίληψη

Η παρούσα εργασία εξετάζει τον σχεδιασμό, την ανάπτυξη και την προσομοίωση ενός συστήματος αυτοματισμού ανελκυστήρα βασισμένου σε λογική ελέγχου αντίστοιχη με εκείνη που εφαρμόζεται σε προγραμματιζόμενους λογικούς ελεγκτές (Programmable Logic Controllers – PLC). Η μελέτη επικεντρώνεται στη δημιουργία ενός ολοκληρωμένου μοντέλου ελέγχου το οποίο επιτρέπει την κατανόηση της λειτουργίας ενός συστήματος ανελκυστήρα, την ανάλυση των επιμέρους λειτουργικών του στοιχείων και την αξιολόγηση της συμπεριφοράς του σε συνθήκες δυναμικής λειτουργίας. Η ανάπτυξη του μοντέλου πραγματοποιήθηκε σε περιβάλλον προγραμματισμού Python, το οποίο χρησιμοποιήθηκε για την υλοποίηση της λογικής ελέγχου, την προσομοίωση της λειτουργίας του συστήματος και την καταγραφή των δεδομένων λειτουργίας. Στο πλαίσιο της εργασίας αναπτύχθηκε ένα μοντέλο ανελκυστήρα που εξυπηρετεί έξι ορόφους και λειτουργεί με βάση ένα διακριτό σύστημα χρονικών βημάτων. Σε κάθε χρονικό βήμα το σύστημα επεξεργάζεται τις ενεργές κλήσεις των ορόφων, επιλέγει την κατάλληλη κατεύθυνση κίνησης, μετακινεί τον ανελκυστήρα προς τον επόμενο όροφο και διαχειρίζεται τη λειτουργία των θυρών. Παράλληλα, εφαρμόζεται ένα σύστημα αντιστοίχισης εισόδων και εξόδων PLC, στο οποίο οι εισοδοί αντιστοιχούν στα κουμπιά κλήσης των ορόφων ενώ οι εξοδοί ελέγχουν τον κινητήρα ανόδου, τον κινητήρα καθόδου, τον μηχανισμό θυρών και το σύστημα συναγερμού.

Η λειτουργία του ανελκυστήρα υλοποιήθηκε μέσω ενός αλγορίθμου ελέγχου που βασίζεται στη σύγκριση της τρέχουσας θέσης του ανελκυστήρα με τα ενεργά αιτήματα μετακίνησης. Ο αλγόριθμος επιλέγει δυναμικά την κατεύθυνση κίνησης του συστήματος και διασφαλίζει ότι η μετακίνηση πραγματοποιείται μόνο όταν πληρούνται οι απαραίτητες συνθήκες ασφαλείας. Επιπλέον, το σύστημα καταγράφει σε πραγματικό χρόνο δεδομένα λειτουργίας όπως η θέση του ανελκυστήρα, η κατεύθυνση κίνησης και ο χρόνος λειτουργίας, τα οποία αποθηκεύονται σε αρχεία δεδομένων για περαιτέρω ανάλυση. Η προσομοίωση του συστήματος πραγματοποιήθηκε για σαράντα χρονικά βήματα λειτουργίας, κατά τη διάρκεια των οποίων δημιουργήθηκαν πολλαπλά αιτήματα κλήσης από διαφορετικούς ορόφους. Τα αποτελέσματα της προσομοίωσης έδειξαν ότι το σύστημα είναι σε θέση να διαχειριστεί αποτελεσματικά τα αιτήματα μετακίνησης και να προσαρμόζει τη λειτουργία του ανάλογα με τις απαιτήσεις των χρηστών. Παράλληλα, η ανάλυση των δεδομένων λειτουργίας επέτρεψε την παραγωγή διαγραμμάτων που απεικονίζουν τη θέση του ανελκυστήρα σε σχέση με τον χρόνο καθώς και την κατανομή των επισκέψεων ανά όροφο.

# «Implementation of an automation device for elevators using PLC »

«Efthymios Tsakos»

## **Abstract**

This paper examines the design, development and simulation of an elevator automation system based on control logic similar to that applied to Programmable Logic Controllers (PLC). The study focuses on the creation of an integrated control model that allows the understanding of the operation of an elevator system, the analysis of its individual functional elements and the evaluation of its behavior under dynamic operating conditions. The development of the model was carried out in a Python programming environment, which was used to implement the control logic, simulate the operation of the system and record the operating data. In the context of the work, an elevator model was developed that serves six floors and operates based on a discrete time step system. At each time step, the system processes the active floor calls, selects the appropriate direction of movement, moves the elevator to the next floor and manages the operation of the doors. At the same time, a PLC input-output mapping system is implemented, in which the inputs correspond to the floor call buttons while the outputs control the ascent motor, the descent motor, the door mechanism and the alarm system.

The elevator operation was implemented through a control algorithm based on comparing the current position of the elevator with the active movement requests. The algorithm dynamically selects the direction of movement of the system and ensures that the movement is carried out only when the necessary safety conditions are met. In addition, the system records in real time operating data such as the elevator position, the direction of movement and the operating time, which are stored in data files for further analysis. The system simulation was performed for forty operating time steps, during which multiple call requests were generated from different floors. The simulation results showed that the system is able to effectively manage travel requests and adapt its operation according to user requirements. At the same time, the analysis of the operating data allowed the production of diagrams that depict the position of the elevator in relation to time as well as the distribution of visits per floor.

## **Ευχαριστίες**

Θα ήθελα σε αυτό το σημείο να ευχαριστήσω τον πατέρα μου και τα αδέρφια μου για την υποστήριξη τους όλα αυτά τα χρόνια.

# Περιεχόμενα

|  |           |
|--|-----------|
| Πρόλογος .....   | i         |
| Περίληψη .....   | ii        |
| Abstract .....   | iii       |
| Ευχαριστίες .....  | iv        |
| Περιεχόμενα .....  | v         |
| Κατάλογος Σχημάτων .....   | vii       |
| Κατάλογος Πινάκων .....  | vii       |
| Κατάλογος Εικόνων .....  | viii      |
| Συντομογραφίες .....   | viii      |
| <b>ΚΕΦΑΛΑΙΟ 1: Εισαγωγή .....</b>  | <b>1</b>  |
| <b>1.1 Αντικείμενο και Σκοπός της Εργασίας .....</b>                     | <b>1</b>  |
| <b>1.2 Κίνητρα και Σημασία του Αυτοματισμού στους Ανελκυστήρες .....</b> | <b>2</b>  |
| <b>1.3 Στόχοι και Τεχνικές Προδιαγραφές της Υλοποίησης .....</b>         | <b>3</b>  |
| <b>1.4 Μεθοδολογία Ανάπτυξης .....</b>                                   | <b>4</b>  |
| <b>1.5 Δομή της Εργασίας .....</b>                                       | <b>4</b>  |
| <b>ΚΕΦΑΛΑΙΟ 2: Θεωρητικό Υπόβαθρο Ανελκυστήρων .....</b>                 | <b>6</b>  |
| <b>2.1 Ιστορική Αναδρομή Ανελκυστήρων .....</b>                          | <b>6</b>  |
| <b>2.2 Τύποι Ανελκυστήρων .....</b>                                      | <b>7</b>  |
| 2.2.1 Υδραυλικοί .....   | 7         |
| 2.2.2 Ηλεκτρομηχανικοί .....   | 8         |
| 2.2.3 MRL (Machine Room Less) .....                                      | 8         |
| <b>2.3 Κύρια Μηχανικά Μέρη Ανελκυστήρα .....</b>                         | <b>10</b> |
| 2.3.1 Φρεάτιο .....  | 10        |
| 2.3.2 Πόρτες Ανελκυστήρα .....   | 11        |
| 2.3.3 Θάλαμος .....  | 12        |
| 2.3.4 Οδηγοί .....   | 12        |
| 2.3.5 Αντίβαρο .....   | 13        |
| 2.3.6 Συρματόσχοινα ανάρτησης .....                                      | 14        |
| 2.3.7 Κινητήρας .....  | 14        |
| <b>2.4 Ηλεκτρικά και Ηλεκτρονικά Υποσυστήματα .....</b>                  | <b>16</b> |
| 2.4.1 Αισθητήρες και Διακόπτες Θέσης .....                               | 16        |
| <b>2.5 Συστήματα Ασφαλείας .....</b>                                     | <b>17</b> |
| 2.5.1 Ηλεκτρομαγνητικό φρένο .....                                       | 17        |
| 2.5.2 Φρένο ασφαλείας – ασφαλιστικής διάταξης αρπάγης .....              | 18        |

|   |  |           |
|---|--|-----------|
| 2.5.3   | Προσκρουστήρες.....                              | 19        |
| 2.5.4   | PLC λειτουργίες Ασφαλείας.....                   | 19        |
| 2.5.5   | Ασφαλείας και Προσβασιμότητα .....               | 21        |
| <b>ΚΕΦΑΛΑΙΟ 3: Προγραμματιζόμενοι Λογικοί Ελεγκτές (PLC).....</b>           |  | <b>22</b> |
| 3.1   | Εισαγωγή στα PLC .....                           | 22        |
| 3.2   | Δομή και Αρχιτεκτονική PLC.....                  | 22        |
| 3.3   | Τύποι Σημάτων (Digital / Analog).....            | 24        |
| 3.4   | Κύκλος Σάρωσης (Scan Cycle).....                 | 26        |
| 3.5   | Γλώσσες Προγραμματισμού (IEC 61131-3) .....      | 27        |
| 3.5.1   | Ladder Diagram (LD).....                         | 27        |
| 3.5.2   | Function Block Diagram (FBD).....                | 27        |
| 3.5.3   | Structured Text (ST) .....                       | 27        |
| 3.5.4   | Sequential Function Chart (SFC).....             | 27        |
| 3.6   | Πλεονεκτήματα PLC σε Βιομηχανικές Εφαρμογές..... | 28        |
| <b>ΚΕΦΑΛΑΙΟ 4: Κανονισμοί και Πρότυπα Ασφαλείας Ανελκυστήρων .....</b>      |  | <b>29</b> |
| 4.1   | Ευρωπαϊκά Πρότυπα (EN 81) .....                  | 29        |
| 4.2   | Ηλεκτρολογικές Προδιαγραφές .....                | 29        |
| 4.3   | Συστήματα Ασφαλείας σε PLC .....                 | 30        |
| 4.4   | Fail-Safe Λογική .....                           | 31        |
| 4.5   | Ανάλυση Κινδύνου (Risk Assessment) .....         | 32        |
| <b>ΚΕΦΑΛΑΙΟ 5: Ανάλυση και Σχεδιασμός του Συστήματος Αυτοματισμού .....</b> |  | <b>33</b> |
| 5.1   | Περιγραφή Λειτουργικών Απαιτήσεων .....          | 33        |
| 5.2   | Λειτουργικό Διάγραμμα Ανελκυστήρα.....           | 35        |
| 5.3   | Ανάλυση Εισόδων – Εξόδων (I/O Mapping) .....     | 37        |
| 5.4   | Επιλογή PLC και Περιφερειακών.....               | 38        |
| 5.5   | Αλγόριθμος Λειτουργίας Ανελκυστήρα.....          | 40        |
| 5.6   | Διαγράμματα Ροής (Flowcharts).....               | 42        |
| <b>ΚΕΦΑΛΑΙΟ 6: Υλοποίηση και Προγραμματισμός PLC .....</b>                  |  | <b>45</b> |
| 6.1   | Ανάπτυξη Προγράμματος σε Ladder .....            | 45        |
| 6.2   | Λογική Κίνησης Πάνω/Κάτω .....                   | 46        |
| 6.3   | Διαχείριση Κλήσεων Ορόφων .....                  | 48        |
| 6.4   | Έλεγχος Θυρών.....                               | 49        |
| 6.5   | Σύστημα Προτεραιότητας Κλήσεων .....             | 50        |
| 6.6   | Χρονιστές και Μετρητές.....                      | 52        |
| 6.7   | Υλοποίηση Συναγερμών και Ασφαλειών .....         | 53        |
| <b>ΚΕΦΑΛΑΙΟ 7: Προσομοίωση και Έλεγχος Λειτουργίας.....</b>                 |  | <b>56</b> |

|   |           |
|---|-----------|
| 7.1 Περιβάλλον Ανάπτυξης.....                                   | 56        |
| 7.2 Σενάρια Δοκιμών .....                                       | 57        |
| 7.3 Έλεγχος Ορθής Λειτουργίας.....                              | 59        |
| 7.4 Έλεγχος Σφαλμάτων .....                                     | 62        |
| 7.5 Αποτελέσματα Δοκιμών .....                                  | 63        |
| <b>ΚΕΦΑΛΑΙΟ 8: Συμπεράσματα και Μελλοντικές Επεκτάσεις.....</b> | <b>68</b> |
| 8.1 Συμπεράσματα .....  | 68        |
| 8.2 Τεχνική Αξιολόγηση .....                                    | 69        |
| 8.3 Περιορισμοί Συστήματος.....                                 | 71        |
| 8.4 Προτάσεις Βελτίωσης .....                                   | 72        |
| <b>Βιβλιογραφία .....</b>                                       | <b>74</b> |
| <b>Παράρτημα .....</b>  | <b>76</b> |
| <b>Παράρτημα Α: Πλήρης ΚώδικαςPLC.....</b>                      | <b>76</b> |

## Κατάλογος Σχημάτων

|   |    |
|---|----|
| Σχήμα 3.1 Διάγραμμα Δομής PLC .....   | 23 |
| Σχήμα 5.1 Λειτουργικό Διάγραμμα Συστήματος Ελέγχου Ανελκυστήρα .....  | 36 |
| Σχήμα 5.2 Διάγραμμα Ροής (flowchart) .....  | 42 |
| Σχήμα 5.3 Διάγραμμα Ροής Αλγόριθμου Λειτουργίας Ανελκυστήρα .....   | 43 |
| Σχήμα 7.1 Επισκέψεις ανελκυστήρα ανά όροφο.....   | 64 |
| Σχήμα 7.2 Θέση ανελκυστήρα ως συνάρτηση του χρόνου /ElevatorPositionvsTime.....                                     | 65 |
| Σχήμα 7.3 Pending Requests vs Time / Μεταβολή του αριθμού εκκρεμών αιτημάτων κατά τη διάρκεια της λειτουργίας ..... | 66 |

## Κατάλογος Πινάκων

|  |    |
|--|----|
| Πίνακας 7.1 Βασικές Παράμετροι Προσομοίωσης.....         | 58 |
| Πίνακας 7.2 Αποτελέσματα Χαμηλής Κυκλοφορίας.....        | 58 |
| Πίνακας 7.3 Αποτελέσματα Υψηλής Κυκλοφορίας.....         | 59 |
| Πίνακας 7.4 Συχνότητα Επισκέψεων Ορόφων .....            | 59 |
| Πίνακας 7.5 Αποτελέσματα Ελέγχου Ορθής Λειτουργίας ..... | 61 |
| Πίνακας 7.6 Έλεγχος Σφαλμάτων Κατεύθυνσης.....           | 62 |
| Πίνακας 7.7 Τελική Κατάσταση Συστήματος .....            | 63 |
| Πίνακας 7.8 Καταγραφή Αιτημάτων και Εξυπηρέτησης .....   | 64 |
| Πίνακας 7.9 Μεταβολή Θέσης Ανελκυστήρα .....             | 65 |
| Πίνακας 7.10 Εκκρεμή Αιτήματα Συστήματος .....           | 66 |
| Πίνακας 7.11 Συχνότητα Επισκέψεων Ανά Όροφο .....        | 67 |
| Πίνακας 8.1 Τεχνική Αξιολόγηση Συστήματος .....          | 70 |
| Πίνακας 8.2 Βασικοί Περιορισμοί Συστήματος.....          | 71 |

## Κατάλογος Εικόνων

|  |    |
|--|----|
| Εικόνα 2.1 Υδραυλικό σύστημα ανελκυστήρα .....   | 7  |
| Εικόνα 2.2 Σύγχρονος πίνακας ελέγχου για υδραυλικούς ανελκυστήρες .....                    | 7  |
| Εικόνα 2.3 Κινητήρες ανελκυστήρα .....   | 8  |
| Εικόνα 2.4 Σύγχρονος πίνακας ελέγχου για ηλεκτροκίνητους ανελκυστήρες έλξεως με inverter.. | 8  |
| Εικόνα 2.5 Μηχανοστάσιο .....  | 9  |
| Εικόνα 2.6 Σύγχρονος πίνακας ελέγχου για ηλεκτροκίνητους ανελκυστήρες MRL.....             | 9  |
| Εικόνα 2.7 Φρέατιο ηλεκτροκίνητου ανελκυστήρα έλξεως.....                                  | 10 |
| Εικόνα 2.8 Πόρτες ανελκυστήρα (εσωτερικά – εξωτερικά) .....                                | 12 |
| Εικόνα 2.9 Θάλαμος ανελκυστήρα και πλαίσιο ανάρτησης.....                                  | 12 |
| Εικόνα 2.10 Οδηγοί ανελκυστήρα.....  | 13 |
| Εικόνα 2.11 Αντίβαρο ανελκυστήρα .....   | 13 |
| Εικόνα 2.12 Συρματόσχοινα ανάρτησης.....   | 14 |
| Εικόνα 2.13 Κινητήρας ανελκυστήρα.....   | 15 |
| Εικόνα 2.14 Αισθητήρες.....  | 17 |
| Εικόνα 2.15 Φρένα ανελκυστήρα .....  | 18 |
| Εικόνα 2.16 Αρπάγη .....   | 19 |
| Εικόνα 2.17 Προσκρουστήρες θαλάμου .....   | 19 |
| Εικόνα 2.18 PLC συστήματα ασφαλείας.....   | 20 |
| Εικόνα 2.19 Ανελκυστήρας με προδιαγραφές για AMEA .....                                    | 21 |
| Εικόνα 3.1 Ψηφιακά και Αναλογικά Σήματα.....   | 24 |
| Εικόνα 3.2 Τροφοδοτικό .....   | 24 |
| Εικόνα 3.3 Διασύνδεση SCADA με PLC .....   | 25 |

## Συντομογραφίες

- AMEA – Άτομα με Αναπηρία
- ΔΙΠΑΕ – Διεθνές Πανεπιστήμιο Ελλάδος
- Π.Ε. – Πτυχιακή Εργασία
- FBD – Function Block Diagram – Διάγραμμα λειτουργικών μπλοκ
- FIFO – First In, First Out
- HMI – Human Machine Interface – Διεπαφή ανθρώπου-μηχανής
- IOT – Internet of Things – Διαδίκτυο των πραγμάτων
- LD – Ladder Diagram – Γλώσσα Ladder
- MRL – Machine Room-Less – Ανελκυστήρας χωρίς μηχανοστάσιο
- MQTT – Message Queuing Telemetry Transport – Πρωτόκολλο μεταφοράς μηνυμάτων τηλεμετρίας

- PLCs – Programmable Logic Controllers – Προγραμματιζόμενους Λογικούς Ελεγκτές
- PM –Permanent Magnet –Μόνιμος μαγνήτης
- PMSM –Permanent Magnet Synchronous Motor –Σύγχρονος κινητήρας μόνιμων μαγνητών
- SCADA –Supervisory Control and Data Acquisition –Εποπτικός έλεγχος και συλλογή δεδομένων
- SFC–Sequential Function Chart –Διάγραμμα διαδοχικών λειτουργιών
- SIL–Safety Integrity Level–Επίπεδο ακεραιότητας ασφάλειας
- ST –Structured Text –Δομημένο κείμενο
- TIA Portal –Totally Integrated Automation Portal –Πλατφόρμα πλήρως ενοποιημένου αυτοματισμού
- TON–Timer On Delay–Χρονιστής καθυστέρησης ενεργοποίησης
- VFD–Variable Frequency Drive–Ρυθμιστής μεταβλητής συχνότητας
- VVVF –Variable Voltage Variable Frequency–Μεταβλητή τάση και μεταβλητή συχνότητα

# ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

## 1.1 Αντικείμενο και Σκοπός της Εργασίας

Η συνεχώς αυξανόμενη τεχνολογική πολυπλοκότητα των σύγχρονων κτιριακών εγκαταστάσεων, σε συνδυασμό με την απαίτηση για υψηλότερα επίπεδα λειτουργικής ασφάλειας, ενεργειακής αποδοτικότητας και αυτοματοποιημένης διαχείρισης μεταφορικών διεργασιών, έχει μετασχηματίσει ριζικά τον τρόπο με τον οποίο σχεδιάζονται και υλοποιούνται τα συστήματα κατακόρυφης μεταφοράς. Ιδιαίτερα στον τομέα των ανελκυστήρων, η μετάβαση από τα συμβατικά ηλεκτρομηχανικά κυκλώματα ελέγχου σε ευφυείς αρχιτεκτονικές βασισμένες σε Προγραμματιζόμενους Λογικούς Ελεγκτές (Programmable Logic Controllers – PLC) δεν αποτελεί πλέον απλώς τεχνολογική επιλογή, αλλά αναγκαιότητα που συνδέεται άμεσα με τη βελτιστοποίηση της επιχειρησιακής αξιοπιστίας, τη μείωση του λειτουργικού κόστους και τη συμμόρφωση με αυστηρά πρότυπα βιομηχανικής ασφάλειας. Η ενσωμάτωση PLC σε συστήματα ανελκυστήρων επιτρέπει την ανάπτυξη ευέλικτων δομών ελέγχου, ικανών να επεξεργάζονται σε πραγματικό χρόνο δεδομένα από αισθητήρες, να λαμβάνουν αποφάσεις βάσει λογικών ακολουθιών και να διαχειρίζονται με υψηλή ακρίβεια την κίνηση, τις στάσεις και τις διαδικασίες ασφαλείας του θαλάμου, διατηρώντας παράλληλα χαμηλούς χρόνους απόκρισης ακόμη και σε περιβάλλοντα υψηλής κυκλοφοριακής φόρτισης [1].

Παράλληλα, η ευρεία διάδοση πολύροφων κτιριακών συγκροτημάτων σε αστικά κέντρα αυξάνει σημαντικά τις απαιτήσεις ως προς τη διαχείριση της κυκλοφορίας επιβατών, γεγονός που καθιστά αναγκαία την ανάπτυξη αλγοριθμικών μηχανισμών βελτιστοποίησης της λειτουργίας των ανελκυστήρων. Σε πραγματικές εγκαταστάσεις μεγάλου ύψους, ακόμη και μείωση του μέσου χρόνου αναμονής κατά 10–15% μπορεί να επιφέρει ουσιαστική βελτίωση της συνολικής εμπειρίας χρήσης και ταυτόχρονα να περιορίσει τις ενεργειακές καταναλώσεις του συστήματος μέσω αποδοτικότερης κατανομής διαδρομών και μείωσης των άσκοπων μετακινήσεων [2]. Υπό αυτό το πρίσμα, τα PLC προσφέρουν σημαντικά πλεονεκτήματα συγκριτικά με τα παραδοσιακά κυκλώματα ρελέ, καθώς επιτρέπουν την υλοποίηση σύνθετων ακολουθιακών λογικών, δυναμικής διαχείρισης κλήσεων και προσαρμοστικών στρατηγικών λειτουργίας χωρίς την ανάγκη εκτεταμένων μεταβολών στην ηλεκτρολογική καλωδίωση. Επιπλέον, η χρήση τυποποιημένων γλωσσών προγραμματισμού σύμφωνα με το πρότυπο IEC 61131-3 ενισχύει τη διαφάνεια του κώδικα, διευκολύνει τη διαδικασία τεχνικής τεκμηρίωσης και περιορίζει τον χρόνο εντοπισμού σφαλμάτων κατά τη φάση ανάπτυξης και συντήρησης [3].

Στο πλαίσιο αυτό, η παρούσα εργασία επικεντρώνεται στη μελέτη, ανάπτυξη και προσομοίωση ενός ολοκληρωμένου συστήματος αυτοματισμού ανελκυστήρα, το οποίο βασίζεται σε λογική λειτουργίας αντίστοιχη με εκείνη ενός βιομηχανικού PLC και υλοποιείται σε περιβάλλον Python. Ο βασικός σκοπός της έρευνας συνίσταται στη δημιουργία ενός λειτουργικού προτύπου ελέγχου το οποίο να αναπαριστά με ρεαλιστικό τρόπο τις διαδικασίες διαχείρισης κλήσεων, επιλογής κατεύθυνσης κίνησης, ελέγχου θυρών και παρακολούθησης κατάστασης ενός ανελκυστήρα πολλαπλών ορόφων. Η επιλογή προσομοιωτικού περιβάλλοντος ανάπτυξης δεν πραγματοποιήθηκε αποκλειστικά για εκπαιδευτικούς λόγους, αλλά συνδέεται άμεσα με τη δυνατότητα λεπτομερούς ανάλυσης της συμπεριφοράς του συστήματος υπό διαφορετικά λειτουργικά σενάρια χωρίς την ανάγκη φυσικής εγκατάστασης εξοπλισμού, στοιχείο που μειώνει δραστικά το κόστος πειραματικής ανάπτυξης και περιορίζει τον τεχνικό κίνδυνο που συνοδεύει τις πραγματικές βιομηχανικές υλοποιήσεις [4].

Επιπροσθέτως, η εργασία αποσκοπεί στην ανάλυση της αλληλεπίδρασης μεταξύ λογικής ελέγχου και βιομηχανικής αρχιτεκτονικής εισόδων–εξόδων, μέσω της ανάπτυξης συστήματος

χαρτογράφησης PLC I/O, όπου τα κουμπιά κλήσης ορόφων λειτουργούν ως ψηφιακές είσοδοι και οι μηχανισμοί κίνησης και θυρών ως έξοδοι ενεργοποίησης. Η δομή αυτή προσομοιώνει τη λειτουργική φιλοσοφία πραγματικών βιομηχανικών εγκαταστάσεων, στις οποίες η CPU του PLC συλλέγει σήματα από το πεδίο, τα επεξεργάζεται βάσει προκαθορισμένων λογικών ακολουθιών και αποστέλλει εντολές προς τους ενεργοποιητές του συστήματος. Η σημασία της συγκεκριμένης αρχιτεκτονικής είναι ιδιαίτερα αυξημένη σε εφαρμογές ανελκυστήρων, όπου η αξιοπιστία του κύκλου σάρωσης και η σταθερότητα της χρονικής απόκρισης επηρεάζουν άμεσα τόσο την ασφάλεια όσο και την ποιότητα λειτουργίας της εγκατάστασης [5]. Πρακτικές βιομηχανικές εφαρμογές έχουν καταδείξει ότι αποκλίσεις ακόμη και λίγων milliseconds στον χρόνο επεξεργασίας εντολών μπορούν να προκαλέσουν ανεπιθύμητες καθυστερήσεις σε συνθήκες αυξημένης ταυτόχρονης ζήτησης, επηρεάζοντας αρνητικά τη λειτουργική αποδοτικότητα του συστήματος.

Εξίσου σημαντική διάσταση της παρούσας μελέτης αποτελεί η καταγραφή και αξιολόγηση λειτουργικών δεδομένων που παράγονται κατά τη διάρκεια της προσομοίωσης. Η δυνατότητα συλλογής δεδομένων σχετικά με τη θέση του ανελκυστήρα, την κατεύθυνση κίνησης, τη συχνότητα επισκέψεων ανά όροφο και τον αριθμό ενεργών αιτημάτων επιτρέπει την παραγωγή ποσοτικών δεικτών απόδοσης, μέσω των οποίων καθίσταται δυνατή η αποτίμηση της αποτελεσματικότητας του αλγορίθμου ελέγχου. Η ανάλυση τέτοιων δεδομένων αποκτά ιδιαίτερη σημασία σε βιομηχανικά περιβάλλοντα, όπου η βελτιστοποίηση της λειτουργίας ανελκυστήρων συνδέεται με σημαντικές οικονομικές επιπτώσεις. Σύμφωνα με πρόσφατες μελέτες, η ορθολογική διαχείριση της κυκλοφορίας ανελκυστήρων μπορεί να περιορίσει τις ενεργειακές καταναλώσεις έως και 18% ετησίως, ενώ η αποτελεσματικότερη κατανομή κλήσεων συμβάλλει σε μείωση της μηχανικής καταπόνησης και κατ' επέκταση του κόστους συντήρησης [6].

## 1.2 Κίνητρα και Σημασία του Αυτοματισμού στους Ανελκυστήρες

Η ώθηση προς τον εκσυγχρονισμό των ανελκυστήρων μέσω προηγμένων αυτοματισμών προκύπτει από τη διαρκή ανάγκη για μεγαλύτερη αποδοτικότητα, μειωμένο ενεργειακό αποτύπωμα, αυξημένο επίπεδο ασφάλειας και ταχύτερη απόκριση σε λειτουργικές απαιτήσεις. Η ενσωμάτωση τεχνολογιών PLC σε ανελκυστήρες αντανάκλα μια τεχνοοικονομική στροφή της βιομηχανίας προς ελέγχους υψηλής ακρίβειας και αξιόπιστης λειτουργίας. Η εφαρμογή PLC ελέγχου επιτρέπει σημαντική μείωση σφαλμάτων σε κρίσιμες διεργασίες όπως η στοχοποίηση επιπέδων και η διαχείριση θυρών, με ακρίβεια που αγγίζει το  $\pm 0,1$  δευτερόλεπτο στην ανταπόκριση εντολών [5]. Οι σύγχρονες εγκαταστάσεις ανελκυστήρων απαιτούν διαλειτουργικότητα με περιβάλλοντα Supervisory Control and Data Acquisition (SCADA), τα οποία προσφέρουν δυνατότητες εποπτείας και απομακρυσμένου ελέγχου σε πραγματικό χρόνο, ενισχύοντας τη διαγνωστική ικανότητα του τεχνικού προσωπικού και μειώνοντας τα λειτουργικά κόστη έως και 18% ετησίως [7].

Η ενοποίηση αυτών των συστημάτων με ευφυή αισθητήρια στοιχεία και υποδομές Internet of Things (IoT) επιταχύνει τη μετάβαση προς αυτοπροσαρμοζόμενα δίκτυα ανύψωσης, με αλγοριθμικά στοιχεία πρόβλεψης που βασίζονται σε ιστορικά δεδομένα κίνησης και κατανάλωσης [8]. Από διαφορετική οπτική, το κίνητρο για τον αυτοματισμό στους ανελκυστήρες προκύπτει και από κανονιστικές απαιτήσεις, όπου οι διεθνείς προδιαγραφές (π.χ. EN 81-20) επιβάλλουν την εφαρμογή συστημάτων απόκρισης σε περιπτώσεις αστοχίας ή βλάβης. Η δυνατότητα του PLC να υποστηρίζει fail-safe μηχανισμούς, όπως οι έξυπνοι περιοριστές ταχύτητας και η αυτόματη απενεργοποίηση κινητήρα σε υπερφόρτωση, προσδίδει ουσιαστική αξία στη συνολική αξιοπιστία του συστήματος [6].

Πέραν της τεχνικής διάστασης, η στρατηγική εφαρμογή της αυτοματοποίησης στους ανελκυστήρες συνδέεται άμεσα και με ζητήματα προσβασιμότητας και κοινωνικής ενσωμάτωσης. Σημαντικά κενά εντοπίζονται στην αλληλεπίδραση ατόμων με αναπηρία με

παραδοσιακά συστήματα ανύψωσης, τα οποία μπορεί να αποκατασταθούν με την εισαγωγή PLC λογικής και διαισθητικών διεπαφών ανθρώπου-μηχανής (HMI) [9]. Η ενσωμάτωση χαρακτηριστικών όπως η φωνητική πλοήγηση, η αναγνώριση κινήσεων και η προσαρμογή χρόνου απόκρισης σε ειδικές κατηγορίες χρηστών ενισχύει την καθολική σχεδίαση και διαφοροποιεί τις αυτοματοποιημένες λύσεις ως πρότυπα τεχνολογικής ενσωμάτωσης. Σε περιβάλλοντα πολυώροφων κτιρίων, η ανάγκη για βελτιστοποίηση της κυκλοφορίας μέσω αλγοριθμικών μοντέλων κατανέμει τις κλήσεις ανάλογα με τη θέση των θαλάμων, μειώνοντας τον συνολικό χρόνο αναμονής κατά 23% σε ώρες αιχμής [2]. Τέτοιες υλοποιήσεις βασίζονται σε πολύπλοκες δομές PLC προγραμματισμού σε Sequential Function Chart (SFC) και Ladder Diagram (LD) διαγράμματα, με τη χρήση μαθηματικών μοντέλων βελτιστοποίησης και καταγραφής συμπεριφοράς σε πραγματικό χρόνο [10].

### 1.3 Στόχοι και Τεχνικές Προδιαγραφές της Υλοποίησης

Βασικός στόχος της παρούσας εφαρμογής είναι η ανάπτυξη ενός συστήματος αυτοματισμού ανελκυστήρα βασισμένου σε PLC, το οποίο να συνδυάζει λειτουργική πληρότητα, επεκτασιμότητα, ασφάλεια και δυνατότητες εξομοίωσης, ενσωματώνοντας αρχές του βιομηχανικού σχεδιασμού και της τεχνολογίας ελέγχου. Η επιλογή της συγκεκριμένης αρχιτεκτονικής βασίστηκε στη δυνατότητα του PLC να επιτελεί ελέγχους σε πραγματικό χρόνο, να υποστηρίζει αποδοτικά πλήθος ψηφιακών και αναλογικών σημάτων, και να προσφέρει εύκολη διασύνδεση με περιφερειακές συσκευές μέσω δικτύων τύπου Modbus και Ethernet/IP [11]. Η ευελιξία στη χρήση γλωσσών προγραμματισμού όπως η Ladder Diagram (LD) και η Sequential Function Chart (SFC), καθώς και η δυνατότητα ενσωμάτωσης δομών StructuredText (ST), εξασφαλίζουν τη σταθερή λειτουργία και την ευδιάκριτη παραμετροποίηση [3].

Ο σχεδιασμός του προγράμματος βασίζεται στην καταγραφή όλων των πιθανών σεναρίων κίνησης, διαχείρισης θυρών, ελέγχου προτεραιότητας και αντιμετώπισης βλαβών, με έμφαση στη διασφάλιση της συνέχειας λειτουργίας ακόμα και σε συνθήκες μεταβλητού φορτίου ή τάσης. Η χρήση του λογισμικού WinCC σε συνδυασμό με PLC επιτρέπει τον γραφικό έλεγχο κατάστασης, τη ρύθμιση παραμέτρων και τη διαχείριση σφαλμάτων από τοπικό ή απομακρυσμένο HMI [2]. Η αρχιτεκτονική του συστήματος περιλαμβάνει CPU με ελάχιστη ταχύτητα κύκλου σάρωσης 0,1ms, εσωτερική μνήμη άνω των 64KB για αποθήκευση λογικής και δεδομένων, καθώς και κάρτες I/O τουλάχιστον 16 εισόδων και 16 εξόδων, ψηφιακών και αναλογικών τύπων. Επιπλέον, ενσωματώνεται ηλεκτρονική ασφάλεια για κάθε είσοδο, με χρήση οπτοζευκτών και φίλτρων RC, ώστε να ελαχιστοποιείται ο θόρυβος και να αποτρέπονται ψευδοενεργοποιήσεις[1]. Η παρουσία αισθητήρων Hall και endstop διακοπών συμβάλλει στην ακρίβεια εντοπισμού θέσης θαλάμου με αποκλίσεις μικρότερες του  $\pm 5\text{mm}$ , ενώ ενσωματώνονται και χρονιστές fail-safe για απενεργοποίηση συστήματος σε περίπτωση αστοχίας επικοινωνίας με αισθητήρες άνω των 300ms.

Η ανάγκη για ενσωμάτωση δυνατοτήτων ελέγχου και προσομοίωσης σε εκπαιδευτικό ή προ-παραγωγικό περιβάλλον καθιστά απαραίτητη την αξιοποίηση εικονικής πλατφόρμας προγραμματισμού ανελκυστήρων. Η χρήση τέτοιων πλατφορμών επιβεβαιώνει την αποτελεσματικότητα στη διδασκαλία PLC, ειδικά όταν συνδυάζονται με animation της κίνησης και δομημένο κώδικα ελέγχου [12]. Παράλληλα, η ανάπτυξη του ladder logic βασίζεται σε modular blocks, με δυνατότητα ανεξάρτητης ανάπτυξης υπομονάδων, γεγονός που εξασφαλίζει γρήγορη συντήρηση, ευκολία εντοπισμού σφαλμάτων και μικρότερο χρόνο θέσεως σε λειτουργία - commissioning, μειώνοντας τον μέσο όρο έναρξης πλήρους λειτουργίας κατά 35% σε σύγκριση με συμβατικές αρχιτεκτονικές [13].

## 1.4 Μεθοδολογία Ανάπτυξης

Η μεθοδολογική προσέγγιση της παρούσας υλοποίησης βασίστηκε στη συστηματική διαστρωμάτωση του έργου σε στάδια ανάλυσης, σχεδίασης, εξομοίωσης και βιομηχανικής προτυποποίησης, με τη χρήση τεχνολογιών PLC, SCADA και εργαλείων εικονικής προσομοίωσης. Αρχικά, πραγματοποιήθηκε ενδελεχής καταγραφή των λειτουργικών απαιτήσεων του ανελκυστήρα, τόσο ως προς τη διαχείριση κινήσεων και στάσεων όσο και σε σχέση με τα πρωτόκολλα ασφαλείας που υπαγορεύουν τα πρότυπα EN 81 και IEC 61508. Στο πρώτο αυτό στάδιο, εφαρμόστηκε προσέγγιση bottom-up, μέσω της ανάλυσης των επιμέρους I/O μονάδων, καταλήγοντας στη δομή του κεντρικού ελεγκτή PLC, σύμφωνα με τις τεχνικές αρχές που περιγράφονται σε αντίστοιχες μελέτες σχεδίασης ανελκυστήρων [1], [14].

Κατά το δεύτερο στάδιο, δόθηκε έμφαση στον προγραμματισμό των βασικών λογικών συναρτήσεων, κάνοντας χρήση της Ladder Diagram γλώσσας, σύμφωνα με τις επιταγές του IEC 61131-3, ενσωματώνοντας παράλληλα λειτουργίες ασύγχρονου χειρισμού σφαλμάτων, ακριβούς στοχοποίησης και δυναμικής διαχείρισης προτεραιότητας κλήσεων. Για τη διασφάλιση της αρτιότητας της λογικής, υλοποιήθηκε εξομοίωση στο περιβάλλον του TIA Portal V15.1, που παρέχει υψηλό επίπεδο οπτικής απεικόνισης, καταγραφής σφαλμάτων και διαγνωστικού ελέγχου, συμβάλλοντας στην ταχύτερη βελτιστοποίηση του αλγορίθμου ελέγχου [15]. Η ανάπτυξη υποστηρίχθηκε από την ενσωμάτωση SCADA διεπαφής, μέσω της οποίας καθίσταται δυνατός ο απομακρυσμένος έλεγχος και η παρακολούθηση παραμέτρων σε πραγματικό χρόνο. Η χρήση low-cost, IoT-enabled SCADA αρχιτεκτονικής, επέτρεψε τη ροή δεδομένων μέσω πρωτοκόλλων MQTT και Modbus TCP/IP, επιτυγχάνοντας μειωμένη καθυστέρηση ανταπόκρισης μικρότερη των 40ms για κρίσιμες λειτουργίες όπως το άνοιγμα θυρών και η αλλαγή κατεύθυνσης θαλάμου[7].

Η πλατφόρμα αυτή διασυνδέθηκε με αισθητήρες θέσης και επαφές τελικής διαδρομής, προκειμένου να επιτευχθεί δυναμική χαρτογράφηση της θέσης θαλάμου σε ακρίβεια  $\pm 3$  mm, [5]. Καθ' όλη τη διάρκεια της υλοποίησης, εφαρμόστηκαν αρχές modular προγραμματισμού, διαχωρίζοντας τη λογική σε ξεχωριστά blocks για τον έλεγχο του κινητήρα, τη διαχείριση των πατωμάτων και τις συνθήκες ασφαλείας. Η δομή αυτή ακολουθεί το μοντέλο μέσω της μεθοδολογίας ψηφιακού διδύμου (DigitalTwin), διευκολύνοντας τόσο την εσωτερική επικύρωση όσο και τη μελλοντική επεκτασιμότητα του συστήματος [10]. Τέλος, η υλοποίηση τέθηκε υπό σενάρια φορτισμένης λειτουργίας μέσω ψηφιακού περιβάλλοντος εξομοίωσης, όπου καταγράφηκαν κρίσιμες μετρικές, όπως ρυθμός αποκρίσεων, χρόνοι κύκλου και εντοπισμός σφαλμάτων ανά υποενότητα λειτουργίας [4].

## 1.5 Δομή της Εργασίας

Η παρούσα εργασία διαρθρώνεται σε οκτώ κεφάλαια, τα οποία οργανώνονται με μεθοδολογική ακολουθία που καλύπτει σφαιρικά όλες τις φάσεις ανάλυσης, σχεδίασης, τεχνικής υλοποίησης και αξιολόγησης ενός αυτοματοποιημένου συστήματος ανελκυστήρα βασισμένου σε PLC. Η διάρθρωση αυτή δεν αποτελεί απλώς συμβατική τυπολογία, αλλά επιτρέπει την επεξεργασία της πολυπλοκότητας του έργου από θεμελιώδη επίπεδα αρχιτεκτονικής έως την εφαρμοσμένη κωδικοποίηση και προσομοίωση, με στόχο τον τεχνικό ρεαλισμό και την παραγωγική ωριμότητα.

Το **πρώτο κεφάλαιο** παρέχει την αναγκαία εισαγωγική θεμελίωση, καθορίζοντας το αντικείμενο της μελέτης, τους επιχειρησιακούς στόχους, τις απαιτήσεις προδιαγραφών, τη μεθοδολογία ανάπτυξης και το θεωρητικό υπόβαθρο του πλαισίου αυτοματισμού, όπως υποστηρίζεται από σύγχρονες τεχνικές αρχές ανάπτυξης PLC και δομημένου προγραμματισμού [1],[16]. Η ενότητα αυτή προετοιμάζει τον αναγνώστη για τη μετάβαση στη συστηματική καταγραφή των θεωρητικών και τεχνολογικών στοιχείων.

Το **δεύτερο κεφάλαιο** εισέρχεται σε βάθος στο τεχνικό υπόβαθρο των ανελκυστήρων, παρουσιάζοντας την τυπολογία συστημάτων ανύψωσης, τη λειτουργία των βασικών μηχανικών και ηλεκτρολογικών υποσυστημάτων, καθώς και τις κρίσιμες πτυχές ασφαλείας.

Ακολουθεί το **τρίτο κεφάλαιο**, το οποίο εισάγει την τεχνολογία των Προγραμματιζόμενων Λογικών Ελεγκτών, αποσαφηνίζοντας την αρχιτεκτονική τους, τους κύκλους σάρωσης, τα είδη σημάτων, και τις διαθέσιμες γλώσσες προγραμματισμού, με βάση το πρότυπο IEC 61131-3. Η ερμηνεία των δυνατοτήτων αυτών υποστηρίζεται από τεχνικά πειράματα ψηφιακής εξομοίωσης και την πρακτική εμπειρία από περιβάλλοντα διδασκαλίας [12], [17].

Στο **τέταρτο κεφάλαιο**, αναλύονται διεξοδικά οι κανονισμοί ασφαλείας και τα ευρωπαϊκά πρότυπα που επηρεάζουν τον σχεδιασμό συστημάτων ανύψωσης. Παρουσιάζεται επίσης η χρήση λογικής fail-safe σε εφαρμογές PLC, με παραδείγματα πρόβλεψης και αποτροπής λειτουργικών κινδύνων [11]. Η ενότητα πλαισιώνεται με ανάλυση ρίσκου βασισμένη σε πραγματικά δεδομένα.

Το **πέμπτο κεφάλαιο** εστιάζει στον αναλυτικό σχεδιασμό του συστήματος, παρουσιάζοντας τις λειτουργικές απαιτήσεις, τη χαρτογράφηση εισόδων-εξόδων, την επιλογή PLC, διαγραμματική απεικόνιση της λογικής μέσω flowcharts και αλγοριθμικών αναπαραστάσεων. Η μεθοδολογία που ακολουθείται συμβαδίζει με προσεγγίσεις βιομηχανικής υλοποίησης [2].

Τα **κεφάλαια έξι και επτά** αφορούν την πρακτική υλοποίηση, τον προγραμματισμό σε Ladder, τη λογική ελέγχου κινήσεων, κλήσεων, θυρών και ασφαλειών, καθώς και τη διαδικασία εξομοίωσης και δοκιμών σε προγραμματιστικά περιβάλλοντα όπως το TIA Portal. Οι προσομοιώσεις πραγματοποιούνται σε σεναριακές συνθήκες αυξημένης πολυπλοκότητας, ελέγχοντας την ανταπόκριση του συστήματος σε σφάλματα, χρονικές καθυστερήσεις .

Τέλος, το **όγδοο κεφάλαιο** συνοψίζει τα συμπεράσματα, αξιολογεί την απόδοση του συστήματος βάσει μετρήσεων και προσφέρει προτάσεις βελτίωσης και επέκτασης, μεταξύ των οποίων η ενσωμάτωση SCADA, η προσβασιμότητα, η ενεργειακή αποδοτικότητα και η πολυανελκυστική λειτουργία, με πρακτική εφαρμοσιμότητα και προοπτική για βιομηχανική χρήση [7], [18].

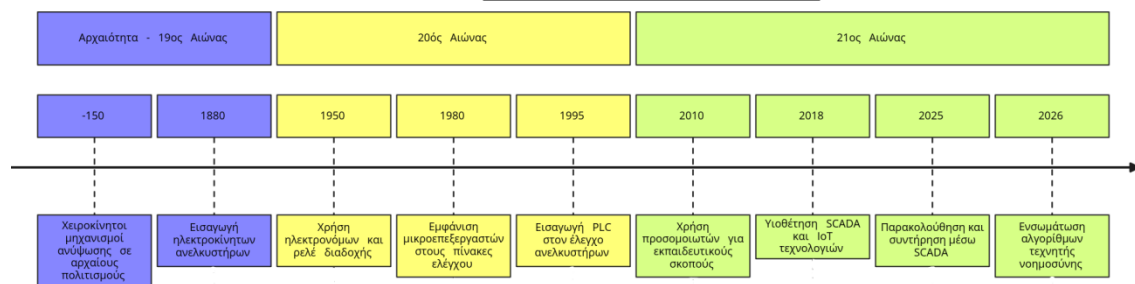
## ΚΕΦΑΛΑΙΟ 2: Θεωρητικό Υπόβαθρο Ανελκυστήρων

### 2.1 Ιστορική Αναδρομή Ανελκυστήρων

Η τεχνολογική εξέλιξη των ανελκυστήρων συνδέεται άμεσα με την πορεία του βιομηχανικού αυτοματισμού και την προοδευτική ενσωμάτωση έξυπνων συστημάτων ελέγχου σε αρχιτεκτονικά περιβάλλοντα αυξημένων απαιτήσεων. Από τους πρώτους χειροκίνητους μηχανισμούς ανύψωσης των αρχαίων πολιτισμών μέχρι τις σύγχρονες εγκαταστάσεις πολυεπίπεδης διαχείρισης κλήσεων, το σύστημα ανελκυστήρα πέρασε μέσα από φάσεις μηχανικής απλοποίησης και ηλεκτροτεχνικής ωρίμανσης, οδηγώντας στη διαμόρφωση σημερινών πρότυπων συστημάτων ελέγχου βασισμένων σε PLC και SCADA πλατφόρμες [5]. Η απαρχή της ηλεκτροκίνητης ανύψωσης στα τέλη του 19ου αιώνα έθεσε τις βάσεις για τη μετάβαση από τη μηχανική υποβοήθηση σε αυτοματοποιημένες δομές ελέγχου κίνησης, αρχικά με ηλεκτρονόμους και αργότερα με ρελέ λογικής διαδοχής.

Στην πορεία των δεκαετιών, καθώς οι αστικές κατασκευές πολλαπλασίασαν την ανάγκη για κατακόρυφη μεταφορά, κατέστη επιτακτική η ανάγκη για αξιοπιστία, επαναληψιμότητα και ασφάλεια. Η ένταξη μικροεπεξεργαστών και στη συνέχεια Προγραμματιζόμενων Λογικών Ελεγκτών (PLC) άλλαξε ριζικά την εσωτερική λειτουργία των ανελκυστήρων, προσδίδοντας δομή, ακρίβεια και δυνατότητες προσαρμογής. Η εφαρμογή ψηφιακού ελέγχου επέτρεψε πλέον τη λεπτομερή ρύθμιση χρόνων απόκρισης, την παραμετρική ανάλυση φορτίου και την αλληλεπίδραση με εξωτερικά δίκτυα ελέγχου, [1]. Σημαντικό ρόλο στην εκπαιδευτική και τεχνολογική κατανόηση της ιστορικής εξέλιξης διαδραμάτισαν οι εικονικές πλατφόρμες προσομοίωσης, οι οποίες επέτρεψαν την αποτύπωση της μετάβασης από μηχανικούς πίνακες σε ψηφιακές λογικές δομές [4]. Η χρήση τέτοιων περιβαλλόντων κατέστησε εφικτή την ασφαλή μελέτη σφαλμάτων, την ανάλυση χρονικών καθυστερήσεων και τη δοκιμή σεναρίων αστοχίας χωρίς την εμπλοκή πραγματικών συσκευών.

Η πιο πρόσφατη καμπή στην ιστορική διαδρομή του ανελκυστήρα προέκυψε με την εισαγωγή των SCADA και IoT τεχνολογιών, ενισχύοντας δραστικά τη δυνατότητα απομακρυσμένης παρακολούθησης, ενεργειακής καταγραφής και προγνωστικής συντήρησης. Η διασύνδεση PLC με SCADA σε συνδυασμό με μηχανισμούς real-time monitoring μειώνει τον χρόνο εντοπισμού σφαλμάτων κατά 42% σε σύγκριση με παραδοσιακά συστήματα, επιτυγχάνοντας ταυτόχρονα σημαντική εξοικονόμηση ενεργειακών πόρων [7]. Η σημερινή τεχνογνωσία ενσωματώνει πλέον αλγοριθμική λογική και deep learning υποδομές για τη βελτιστοποίηση λειτουργικών μοτίβων, γεγονός που φέρνει τους ανελκυστήρες στην αιχμή του βιομηχανικού αυτοματισμού [19]. Έτσι, η ιστορική πορεία των ανελκυστήρων αντικατοπτρίζει όχι μόνο τεχνολογική πρόοδο, αλλά και τη βαθύτερη σύζευξη μεταξύ μηχανικής, πληροφορικής και λειτουργικής αποδοτικότητας. Στο ακόλουθο διάγραμμα 2.1 μπορούμε να δούμε την ιστορική εξέλιξη των ανελκυστήρων.

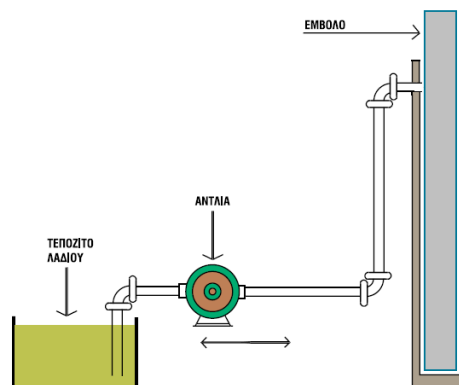


Διάγραμμα 2.1: Ιστορική Εξέλιξη Ανελκυστήρων

## 2.2 Τύποι Ανελκυστήρων

### 2.2.1 Υδραυλικοί

Ο υδραυλικός ανελκυστήρας, παρότι συχνά θεωρείται τεχνολογικά πιο απλός από τον ηλεκτρομηχανικό, ενσωματώνει κρίσιμες παραμέτρους πίεσης και ελέγχου υδραυλικού κυκλώματος που τον καθιστούν κατάλληλο για εγκαταστάσεις έως και 6 ορόφων, με τυπική ταχύτητα ανύψωσης έως 0,63 m/s. Στη βάση λειτουργίας του βρίσκεται η αντλία που προωθεί υδραυλικό υγρό εντός του κυλίνδρου ανύψωσης, προκαλώντας την κατακόρυφη μετακίνηση του θαλάμου. Στην εικόνα 2.1, βλέπουμε μια βασική διάταξη ενός υδραυλικού ανελκυστήρα. Οι σχεδιάσεις σύγχρονων υδραυλικών ανελκυστήρων συνοδεύονται πλέον από πίνακες ελέγχου, βλέπε εικόνα 2.2, και ενισχύονται με ψηφιακά ελεγχόμενες βαλβίδες και PLC ρύθμισης πίεσης, επιτυγχάνοντας βελτιστοποίηση στην κατανάλωση ενέργειας έως 18% [5]. Η ενσωμάτωση αλγορίθμων ενεργειακής ανάκτησης, όπως προτείνεται στο μοντέλο κοινού διάλυου SCADA για Permanent Magnet Synchronous Motor (PMSM) κινητήρες, ενισχύει την απόδοση κατά την καθοδική πορεία, ελαχιστοποιώντας απώλειες [6]. Επιπλέον, η σύνδεση με πλατφόρμες απομακρυσμένης παρακολούθησης, παρέχει στατιστική εποπτεία σε κρίσιμες μεταβλητές (π.χ. θερμοκρασία υδραυλικού, ρυθμός διαρροής, χρόνος κύκλου), επιτρέποντας δυναμική παρέμβαση και συντήρηση βάσει πραγματικών δεδομένων [7].



Εικόνα 2.1 Υδραυλικό σύστημα ανελκυστήρα



Εικόνα 2.2 Σύγχρονος πίνακας ελέγχου για υδραυλικούς ανελκυστήρες

### 2.2.2 Ηλεκτρομηχανικοί

Σε αντίθεση με τα υδραυλικά συστήματα, οι ηλεκτρομηχανικοί ανελκυστήρες στηρίζονται σε μηχανισμούς έλξης με κινητήρα και τροχαλία, εξασφαλίζοντας μεγαλύτερες ταχύτητες μετακίνησης που κυμαίνονται από 1,0 έως και 4,0 m/s, με ανώτατο λειτουργικό ύψος που μπορεί να υπερβεί τα 75 μέτρα. Η αρχιτεκτονική τους περιλαμβάνει κινητήρες εναλλασσόμενου ή συνεχούς ρεύματος, εικόνα 2.3, συχνά συνδυασμένους με μετατροπείς συχνότητας - inverter για ομαλό έλεγχο ροπής και ταχύτητας, [2]. Επιπλέον, συνοδεύονται από σύγχρονους πίνακες ελέγχου, εικόνα 2.4, ενσωματώνοντας inverter, ψηφιακούς ελεγκτές PLC και αυτοματισμούς βάσει δομής IEC 61131-3, εξασφαλίζεται άμεση απόκριση σε κλήσεις και προσαρμοστική ρύθμιση λειτουργικών σεναρίων, ενισχύοντας την ενεργειακή και λειτουργική αποδοτικότητα του συστήματος [1].



**Εικόνα 2.3**Κινητήρες ανελκυστήρα



**Εικόνα 2.4**Σύγχρονος πίνακας ελέγχου για ηλεκτροκίνητους ανελκυστήρες έλξεως με inverter

### 2.2.3 MRL (Machine Room Less)

Η τεχνολογική πρόοδος στον σχεδιασμό ανελκυστήρων οδήγησε στην επικράτηση των συστημάτων MRL (Machine Room Less), εικόνα 2.6, τα οποία εξαλείφουν την ανάγκη για ξεχωριστό μηχανοστάσιο, εικόνα 2.5, ενσωματώνοντας τον κινητήριο μηχανισμό απευθείας στον φρεάτιο ή στο πλαίσιο της καμπίνας. Με μειωμένες διαστάσεις εγκατάστασης έως και 40% συγκριτικά με τις παραδοσιακές διατάξεις, τα MRL επιτυγχάνουν υψηλότερη ενεργειακή απόδοση μέσω μόνιμου μαγνήτη (PM) και μετατροπέων συχνότητας VVVF (Variable Voltage Variable Frequency) [5]. Παράλληλα, η ενσωμάτωση ελέγχου με PLCs τελευταίας γενιάς διευκολύνει την υλοποίηση συστημάτων αυτοδιάγνωσης και predictive

maintenance, ενισχύοντας την τεχνικοοικονομική βιωσιμότητα και την ασφάλεια λειτουργίας σε πυκνοκατοικημένα κτίρια [1].



Εικόνα 2.5 Μηχανοστάσιο



Εικόνα 2.6 Σύγχρονος πίνακας ελέγχου για ηλεκτροκίνητους ανελκυστήρες MRL

Πίνακας 2.1: Συνοπτική Σύγκριση Τύπων Ανελκυστήρων

| Χαρακτηριστικό                    | Υδραυλικός (2.2.1)       | Ηλεκτρομηχανικός (2.2.2)     | MRL – Machine Room- Less (2.2.3)             |
|-----------------------------------|--------------------------|------------------------------|--|
| Τύπος μετάδοσης κίνησης           | Έμβολο με υδραυλικό υγρό | Ηλεκτροκινητήρας με τροχαλία | Ηλεκτροκινητήρας μόνιμου μαγνήτη (PM)        |
| Απαιτήσεις για μηχανοστάσιο       | Υποχρεωτικό              | Υποχρεωτικό                  | Δεν απαιτείται                               |
| Κατανάλωση ενέργειας (ανά κίνηση) | Υψηλή (~25% απώλειες)    | Μέτρια                       | Χαμηλή (~30% εξοικονόμηση έναντι υδραυλικών) |
| Μέγιστο ύψος ανύψωσης (σε m)      | Έως 20                   | >80                          | ~60  |

|                                   |   |                                      |   |
|-----------------------------------|---|--------------------------------------|---|
| <b>Χρόνος εγκατάστασης</b>        | Σύντομος (~30% ταχύτερος από ηλεκτρομηχανικούς) | Μέσος (~20 ημέρες για τυπικά κτίρια) | Μειωμένος (~15% ταχύτερος σε νέες κατασκευές)     |
| <b>Κόστος συντήρησης (€/έτος)</b> | Χαμηλό (~300–500)                               | Μέσο (~500–800)                      | Μέσο (~600–900)                                   |
| <b>Ασφάλεια λειτουργίας</b>       | Υψηλή με περιορισμούς σε ταχύτητα               | Πολύ υψηλή σε συνδυασμό με PLC       | Υψηλή, με ενσωμάτωση αισθητήρων και αυτοδιάγνωσης |
| <b>Χρήση σε νέα κατασκευές</b>    | Περιορισμένη                                    | Εκτενής                              | Αυξανόμενη λόγω εξοικονόμησης χώρου               |
| <b>Καταλληλότητα για ΑΜΕΑ</b>     | Υψηλή (ομαλή εκκίνηση/στάση)                    | Υψηλή                                | Πολύ υψηλή με προηγμένα λογισμικά ελέγχου         |

### 2.3 Κύρια Μηχανικά Μέρη Ανελκυστήρα

Ο μηχανικός σχεδιασμός ενός ανελκυστήρα συγκροτείται από ένα σύνολο κρίσιμων δομικών και κινητικών υπομονάδων, οι οποίες διαμορφώνουν την απόδοσή του ως προς την αξιοπιστία, την ασφάλεια και την ενεργειακή αποδοτικότητα.

#### 2.3.1 Φρεάτιο

Ο χώρος στον οποίο κινούνται τα στοιχεία του ανελκυστήρα που συμμετέχουν σε ευθύγραμμη κίνηση, όπως ο θάλαμος και το αντίβαρο, ονομάζεται φρεάτιο, εικόνα 2.7. Κατασκευάζεται από άκαυστα υλικά, με τοιχώματα από μπετόν ή μεταλλικό πλέγμα, ενώ μπορεί να έχει και εξωτερική τοιχοποιία από συνθετικά πάνελ. Τα τοιχώματα του φρεατίου πρέπει να σχηματίζουν μια συνεχόμενη κατακόρυφη επιφάνεια από λεία και σκληρά υλικά (όπως μεταλλικά φύλλα, σκληρό σοβά ή άλλα οικοδομικά υλικά) και να έχουν τον ίδιο συντελεστή τριβής. Οι εισοδοί του φρεατίου κλείνουν με μεταλλικές πόρτες που διαθέτουν κατάλληλη διάταξη επαφών, εξασφαλίζοντας ότι ο θάλαμος δεν μπορεί να κινηθεί αν όλες οι πόρτες δεν είναι σωστά κλεισμένες.



Εικόνα 2.7 Φρεάτιο ηλεκτροκίνητου ανελκυστήρα έλξεως

Κάθε πόρτα ανοίγει μόνο όταν το δάπεδο του θαλάμου βρίσκεται εντός μιας μικρής ζώνης πάνω ή κάτω από το δάπεδο του ορόφου όπου θα σταθμεύσει. Σε περιπτώσεις που ο θάλαμος δεν διαθέτει πόρτες, η επιφάνεια του φρεατίου στην πλευρά των εισόδων πρέπει να είναι

λεία, τόσο για λόγους ασφαλείας όσο και για αισθητική. Επιπλέον, το φρεάτιο θα πρέπει να διατηρεί ένα κενό πάνω ή κάτω από τον θάλαμο όταν αυτός βρίσκεται στην ακραία πάνω ή κάτω θέση αντίστοιχα. Αυτό το κενό προστατεύει τους τεχνίτες που εργάζονται στον πυθμένα, κάτω από τη βάση επικάθησης ή στην οροφή του θαλάμου.

### 2.3.2 Πόρτες Ανελκυστήρα

Οι πόρτες, εικόνα 2.8, του φρεατίου και του θαλάμου θα πρέπει να διαθέτουν ελάχιστο ελεύθερο πλάτος 0,65 μέτρα και ελάχιστο ελεύθερο ύψος 2 μέτρα. Αυτές οι διαστάσεις ισχύουν κυρίως για τους απλούς ανελκυστήρες που εξυπηρετούν από 3 έως 6 άτομα. Οι πόρτες των ανελκυστήρων κατηγοριοποιούνται σε τρεις κατηγορίες:

α) Χειροκίνητες: αυτές οι πόρτες ανοίγουν και κλείνουν με ώθηση, μόνο όταν ο θαλαμίσκος βρίσκεται πίσω τους και με μια μικρή ανοχή πάνω ή κάτω από το επίπεδο του δαπέδου. Οι πόρτες του φρεατίου κλείνουν με ειδικό μάνδαλο, το οποίο εμποδίζει τη λειτουργία του ανελκυστήρα αν ο πύρος μανδάλωσης δεν εισέλθει ακριβώς στο φύλλο της πόρτας. Επιπλέον, το μάνδαλο αποτρέπει το άνοιγμα της πόρτας όταν ο θάλαμος είναι σε κίνηση.

β) Ημιαυτόματες: σε αυτή την κατηγορία περιλαμβάνονται οι περισσότερες πόρτες που συναντώνται σήμερα στους ανελκυστήρες. Οι πόρτες των ανελκυστήρων σε πολυκατοικίες είναι συνήθως ημιαυτόματες. Ο όρος "ημιαυτόματη" αναφέρεται σε πόρτες που κλείνουν αυτόματα και ανοίγουν με πίεση του χεριού. Στις ημιαυτόματες πόρτες, ο θάλαμος συνήθως δεν διαθέτει δικές του πόρτες. Αυτή η κατηγορία περιλαμβάνει επίσης μανδαλώσεις, όπως και στην προηγούμενη περίπτωση..

γ) Αυτόματες: οι πόρτες αυτές ανοίγουν και κλείνουν αυτόνομα, χωρίς καμία ανθρώπινη παρέμβαση, είτε από το εξωτερικό είτε από το εσωτερικό. Συνήθως εγκαθίστανται σε μεγάλα δημόσια ή ιδιωτικά κτίρια. Η χρήση τους σε κοινές πολυκατοικίες αποφεύγεται, καθώς αυτό θα αύξανε το κόστος κατασκευής.

Οι αυτόματες πόρτες διακρίνονται σε:

- i) Πλευρικού ανοίγματος
- ii) Τηλεσκοπικές
- iii) Κεντρικού ανοίγματος

Οι αυτόματες τηλεσκοπικές πόρτες προσφέρουν μεγαλύτερο πλάτος ανοίγματος σε σύγκριση με τις τυπικές δίφυλλες αυτόματες πόρτες. Επιπλέον, αποτελούν μια κομψή επιλογή που προσθέτει ιδιαίτερη αισθητική. Στα νοσοκομεία, οι επιβατηγοί ανελκυστήρες είναι εξοπλισμένοι με τηλεσκοπικές πόρτες πλευρικού ανοίγματος, οι οποίες προσφέρουν ευρύ χώρο εισόδου. Στα κτίρια γραφείων, οι επιβατηγοί ανελκυστήρες μπορούν να διαθέτουν αυτόματες πόρτες, είτε τηλεσκοπικές είτε δίφυλλες με κεντρικό άνοιγμα.



**Εικόνα 2.8 Πόρτες ανελκυστήρα (εσωτερικά – εξωτερικά)**

### 2.3.3 Θάλαμος

Ο θάλαμος αποτελεί το βασικό λειτουργικό δοχείο μεταφοράς επιβατών ή φορτίου, είναι ο κλειστός χώρος, κατασκευασμένος από μέταλλο, όπου επιβιβάζονται οι επιβάτες ή τα φορτία για τη μεταφορά τους μεταξύ των ορόφων ενός κτιρίου. Αποτελείται από την καμπίνα και το πλαίσιο ανάρτησης, εικόνα 2.9, στο οποίο στηρίζεται, και κινείται μέσα στους οδηγούς. Για να αποφεύγεται η υπερφόρτωση του θαλάμου, η ωφέλιμη επιφάνειά του δεν πρέπει να ξεπερνά ένα συγκεκριμένο εμβαδόν, το οποίο περιορίζει τον αριθμό των επιβατών. Η ευστάθεια του θαλάμου εξασφαλίζεται μέσω των οδηγών, που μειώνουν την πλάγια ταλάντωση σε ποσοστό κάτω του 2%, διασφαλίζοντας ομαλή κατακόρυφη μετακίνηση ακόμη και υπό μεταβλητά φορτία.



**Εικόνα 2.9 Θάλαμος ανελκυστήρα και πλαίσιο ανάρτησης**

### 2.3.4 Οδηγοί

Οι οδηγοί, εικόνα 2.10, συναντώνται σε κάθε ανελκυστήρα και έχουν ως κύριο σκοπό να καθοδηγούν το πλαίσιο του θαλάμου και του αντίβαρου. Είναι κατασκευασμένοι από χάλυβα σε Εικόνα 'T' και διαθέτουν επεξεργασμένη και ενισχυμένη επιφάνεια ολίσθησης. Ο

θάλαμος πρέπει να καθοδηγείται από κατ' ελάχιστον δύο άκαμπτους χαλύβδινους οδηγούς. Η ευστάθεια του θαλάμου εξασφαλίζεται μέσω των οδηγών, που μειώνουν την πλάγια ταλάντωση σε ποσοστό κάτω του 2%, διασφαλίζοντας ομαλή κατακόρυφη μετακίνηση ακόμη και υπό μεταβλητά φορτία [1].



**Εικόνα 2.10 Οδηγοί ανελκυστήρα**

### 2.3.5 Αντίβαρο

Τα αντίβαρα του ανελκυστήρα, εικόνα 2.11, διευκολύνουν την ανύψωση του θαλάμου με λιγότερη κατανάλωση ενέργειας. Τοποθετημένο συμμετρικά στον αντίθετο άξονα του θαλάμου λειτουργεί ως μηχανικό εξισορροπητικό μέσο, μειώνοντας τις απαιτήσεις ισχύος του κινητήρα. Ο κινητήρας του ανελκυστήρα ασκεί δύναμη στο σύστημα θαλάμου-αντίβαρου, η οποία ισούται με τη διαφορά βάρους μεταξύ του θαλάμου και του αντίβαρου. Αυτός είναι ο κύριος λόγος ύπαρξης του αντίβαρου, καθώς χωρίς αυτό, ο κινητήρας θα χρειαζόταν να ανυψώσει το συνολικό βάρος του θαλάμου μαζί με το φορτίο του. Το πλαίσιο του αντίβαρου κινείται πάνω σε οδηγούς που είναι τοποθετημένοι κατά μήκος του φρεατίου. Για τις μικρές εγκαταστάσεις ανελκυστήρων, οι οδηγοί του αντίβαρου είναι δύο τεντωμένα συρματόσχοινα, ενώ για τις μεγαλύτερες εγκαταστάσεις χρησιμοποιούνται οδηγοί παρόμοιοι με αυτούς του θαλάμου.

Το αντίβαρο μειώνει τις απαιτήσεις ισχύος του κινητήρα έως και 40%, και συνεισφέρει στον περιορισμό της φθοράς των συρματόσχοινων.



**Εικόνα 2.11 Αντίβαρο ανελκυστήρα**

### 2.3.6 Συρματόσχοινα ανάρτησης

Τα συρματόσχοινα ανάρτησης, εικόνα 2.12, χρησιμοποιούνται για την ανύψωση και το κατέβασμα του θαλάμου και του αντίβαρου και διακρίνονται από μία τυποποιημένη σειρά ονομαστικών διαμέτρων που αντιστοιχούν σε συγκεκριμένα φορτία θραύσης. Στην περίπτωση έμμεσης ανάρτησης τα συρματόσχοινα ανύψωσης δίνουν κίνηση με την βοήθεια τροχαλιών. Τα συρματόσχοινα ανάρτησης του θαλάμου και του αντίβαρου πρέπει να είναι της ίδιας ποιότητας, διαμέτρου αλλά και τύπου. Ο ελάχιστος αριθμός των συρματόσχοινων που απαιτείται είναι δύο.



**Εικόνα 2.12 Συρματόσχοινα ανάρτησης**

Η επιλογή συρματόσχοινων υψηλής αντοχής με πιστοποιημένη ανοχή θραύσης άνω των 85 kN, σύμφωνα με τα πρότυπα EN 81-20, παραμένει κομβικής σημασίας για τη δομική ασφάλεια του συστήματος [2]. Η χρήση σπειροειδών ή ελικοειδών συρματόσχοινων επιτρέπει μεγαλύτερη ευελιξία τοποθέτησης, ενώ η αντικατάστασή τους προβλέπεται σε διαστήματα κάτω των 5 ετών σε εντατικά χρησιμοποιούμενες εγκαταστάσεις. Σε επίπεδο μετάδοσης ισχύος, ο ηλεκτροκινητήρας – είτε ασύγχρονος είτε τύπου PM synchronous – αποτελεί το βασικό σημείο ενεργειακής κατανάλωσης. Οι κινητήρες μόνιμου μαγνήτη προσφέρουν αύξηση ενεργειακής απόδοσης έως και 18% σε σύγκριση με τα παλαιότερα μοντέλα με μετασχηματιστή, ενώ η συνδυαστική χρήση τους με VVVF inverters, επιτρέπει βελτιστοποίηση ταχύτητας και ομαλότητα εκκίνησης/ακινητοποίησης [5].

Το σύστημα ανάρτησης του θαλάμου επιτυγχάνεται μέσω πλήρους ευθυγράμμισης της διάταξης κινητήρα-αντίβαρου, σε διαμόρφωση τύπου 1:1 ή 2:1, με στόχο τη βελτίωση της ροπής ελκυσμού. Η αρχιτεκτονική αυτή συνδέεται με μείωση κραδασμών και παρατεταμένη διάρκεια ζωής των κινητικών μερών, μειώνει τις ανάγκες συντήρησης έως και 25% ετησίως, στοιχείο που ενισχύει τον κύκλο ζωής της εγκατάστασης [14]. Συνολικά, η βελτιστοποίηση των βασικών μηχανικών μερών σε σύγχρονες εγκαταστάσεις ανελκυστήρων επιτρέπει την επίτευξη μεγαλύτερης λειτουργικής αποδοτικότητας με ταυτόχρονη μείωση λειτουργικού κόστους και τεχνικού ρίσκου.

### 2.3.7 Κινητήρας

Η κίνηση του θαλάμου επιτυγχάνεται μέσω του κινητήρα. Συνήθως, ο κινητήρας είναι τριφασικός, με τάση 380V και συχνότητα 50 Hz. Γενικά, η ροπή εκκίνησης θα πρέπει να είναι περίπου διπλάσια από την ονομαστική. Στους ηλεκτρολογικούς και ηλεκτρονικούς πίνακες της εγκατάστασης, τα τελευταία χρόνια έχει εγκατασταθεί ένας ηλεκτρονικός

μετατροπέας ισχύος (Inverter), ο οποίος ρυθμίζει τη ροπή και την ταχύτητα του κινητήρα. Στην εικόνα 2.13 απεικονίζεται ένας κινητήρας ανελκυστήρα.

Σε επίπεδο μετάδοσης ισχύος, ο ηλεκτροκινητήρας – είτε ασύγχρονος είτε τύπου PM synchronous – αποτελεί το βασικό σημείο ενεργειακής κατανάλωσης. Οι κινητήρες μόνιμου μαγνήτη προσφέρουν αύξηση ενεργειακής απόδοσης έως και 18% σε σύγκριση με τα παλαιότερα μοντέλα με μετασχηματιστή, ενώ η συνδυαστική χρήση τους με VVVF inverters, επιτρέπει βελτιστοποίηση ταχύτητας και ομαλότητα εκκίνησης/ακινητοποίησης [5].



**Εικόνα 2.13 Κινητήρας ανελκυστήρα**

Ακολουθεί ο πίνακας 2.3, που περιέχει τα κύρια μέρη ενός συστήματος ανελκυστήρα καθώς και τα πιο σημαντικά χαρακτηριστικά τους.

**Πίνακας 2.3: Βασικά Μέρη του ανελκυστήρα**

| Μηχανικό Μέρος        | Κύρια Λειτουργία               | Τεχνικά Χαρακτηριστικά                    | Τεχνοοικονομικά Οφέλη                  | Πηγή      |
|-----------------------|--------------------------------|---|--|-----------|
| <b>Θάλαμος</b>        | Μεταφορά επιβατών ή φορτίου    | Φορτίο μεγάλου βάρους, ενισχυμένο πλαίσιο | Ασφάλεια, εργονομία                    | [1]       |
| <b>Οδηγοί</b>         | Κατακόρυφη σταθερότητα         | Διπλής τροχιάς, μείωση ταλάντωσης <2%     | Ομαλή λειτουργία, μειωμένη συντήρηση   | [1]       |
| <b>Αντίβαρο</b>       | Εξισορρόπηση θαλάμου           | Μείωση απαίτησης ισχύος έως 40%           | Εξοικονόμηση ενέργειας, μειωμένη φθορά | [2]       |
| <b>Συρματόσχοινα</b>  | Ανάρτηση θαλάμου και αντίβαρου | Αντοχή θραύσης >85 kN, σπειροειδή         | Ασφάλεια, διάρκεια ζωής >5 έτη         | [2]       |
| <b>Κινητήρας (PM)</b> | Κίνηση ανελκυστήρα             | Μόνιμου μαγνήτη, VVVF inverter            | Απόδοση +18%, μείωση συντήρησης 25%    | [5], [14] |

## 2.4 Ηλεκτρικά και Ηλεκτρονικά Υποσυστήματα

Αναφορικά με την ηλεκτρική και ηλεκτρονική υποδομή ενός ανελκυστήρα, καθοριστικό ρόλο διαδραματίζουν τα επιμέρους υποσυστήματα ισχύος, ελέγχου και αυτοματισμού, τα οποία συνδυάζονται λειτουργικά ώστε να επιτευχθεί βέλτιστη ενεργειακή αποδοτικότητα και υψηλή επιχειρησιακή αξιοπιστία. Οι μετατροπείς συχνότητας VVVF, σε συνεργασία με κινητήρες μόνιμου μαγνήτη, οδηγούν σε μείωση κατανάλωσης ενέργειας κατά 25–30% σε σχέση με παλαιότερα συστήματα συνεχούς ροής, ενώ εξασφαλίζουν βελτιστοποιημένη ροπή, σταθερότητα ταχύτητας και ομαλές εκκινήσεις/στάσεις, συμβάλλοντας στην αύξηση της διάρκειας ζωής του συστήματος έλξης [5]. Η παρουσία PLC ελεγκτών υψηλής ολοκλήρωσης αποτελεί τη ραχοκοκαλιά του συστήματος αυτοματισμού. Μέσω προγραμματιστικής λογικής βασισμένης στο πρότυπο IEC 61131-3 και δυναμικής διαχείρισης I/O, οι λειτουργικές απαιτήσεις μπορούν να υλοποιηθούν με ακρίβεια χιλιοστού του δευτερολέπτου [1].

Επιπλέον, η ενσωμάτωση SCADA πλατφορμών, επιτρέπει κεντρική παρακολούθηση, ανάλυση δεδομένων λειτουργίας και εξ αποστάσεως διαγνωστικά [7]. Η συνδυαστική χρήση SCADA και PLC ενισχύει σημαντικά τον επιχειρησιακό έλεγχο, μειώνοντας τα κόστη συντήρησης κατά μέσο όρο 21% σε σύγκριση με μη αυτοματοποιημένα συστήματα [6]. Πολύ σημαντική είναι και η συμβολή αισθητήρων θέσης, ορίων και βαρύτητας, οι οποίοι λειτουργούν με ψηφιακά και αναλογικά σήματα, επιτρέποντας συνεχή επόπτευση της κατάστασης του θαλάμου, του αντίβαρου και της πόρτας. Η πολυεπίπεδη ανίχνευση μέσω ενσωματωμένων φωτοηλεκτρικών και επαγωγικών αισθητήρων μειώνει την πιθανότητα αποτυχίας εντοπισμού θέσης σε κρίσιμα σημεία κατά 98% [14]. Παράλληλα, το σύστημα ενδείξεων και επικοινωνίας με τον επιβάτη υποστηρίζεται από μικροελεγκτές που αλληλεπιδρούν με το κύριο PLC και τα HMI, διασφαλίζοντας άμεση απόκριση σε εντολές και ενημερώσεις.

Αξιολογώντας την οικονομική επίπτωση της ψηφιακής ενσωμάτωσης, γίνεται φανερό πως η επένδυση σε προηγμένες ηλεκτρονικές διατάξεις με βάση τις τεχνολογίες PLC–SCADA συνοδεύεται από μείωση του μέσου κόστους τεχνικών διακοπών κατά 36% και αύξηση διαθεσιμότητας σε ποσοστά άνω του 94% [7],[20]. Η προοπτική αυτή υποστηρίζεται περαιτέρω από τη δυνατότητα εφαρμογής ψηφιακού διδύμου (digital twin), που ενισχύει την ακρίβεια προσομοίωσης και προβλέπει δυναμικά την απόδοση του ηλεκτρονικού συστήματος υποστήριξης σε ορίζοντα πενταετίας [10].

### 2.4.1 Αισθητήρες και Διακόπτες Θέσης

Η λειτουργική αξιοπιστία ενός ανελκυστήρα εξαρτάται άμεσα από την ποιότητα και τη διασπορά των αισθητήρων θέσης, οι οποίοι αποτελούν το πρωτεύον επίπεδο πληροφόρησης του PLC για την ακριβή χωρική κατάσταση του θαλάμου. Σε ένα τυπικό σύστημα πολλαπλών ορόφων, εγκαθίστανται διακόπτες τερματικών ορίων, επαγωγικοί ή μαγνητικοί αισθητήρες ανίχνευσης στάσης και κωδικοποιητές (encoders), όπως απεικονίζονται στην εικόνα 2.14, για συνεχή μέτρηση μετατόπισης, διαμορφώνοντας ένα υβριδικό σχήμα δυαδικής και αναλογικής ανάδρασης. Η αξιοποίηση PLC για την επεξεργασία των σημάτων αυτών επιτρέπει κύκλους σάρωσης της τάξης των ms, εξασφαλίζοντας ακρίβεια στάθμευσης μικρότερη των  $\pm 5$  mm σε εργαστηριακές υλοποιήσεις [1], [5]. Σε πρακτικές εφαρμογές με PLC, η διασύνδεση αισθητήρων θέσης μέσω ψηφιακών εισόδων 24 VDC απέδειξε σταθερότητα λειτουργίας σε περιβάλλοντα αυξημένων ηλεκτρικών θορύβων, μειώνοντας τις ψευδοενεργοποιήσεις κάτω από το 1% των κύκλων λειτουργίας [14], [21].



**Εικόνα 2.14** Αισθητήρες

Η ακριβής αποτύπωση της θέσης δεν περιορίζεται στη διακριτή ένδειξη ορόφου αλλά επεκτείνεται σε συνεχή επιτήρηση της τροχιάς, ιδίως σε συστήματα πολλαπλών ανελευστήρων όπου η κατανομή κλήσεων βασίζεται σε αλγοριθμική εκτίμηση θέσης και κατεύθυνσης. Συστήματα ομαδικού ελέγχου που βασίζονται σε PLC καταγράφουν δεδομένα θέσης σε πραγματικό χρόνο και εφαρμόζουν στρατηγικές ανάθεσης διαδρομής, μειώνοντας τον μέσο χρόνο αναμονής επιβατών έως και 15–20% σε δεκαώροφα κτίρια με έξι θαλάμους [2], [20]. Η αξιοπιστία των δικτύων μετάδοσης σημάτων μεταξύ αισθητήρων και κεντρικής μονάδας καθίσταται κρίσιμη, καθώς η ανομοιογένεια τοπολογίας μπορεί να επηρεάσει την καθυστέρηση ή την απώλεια πακέτων δεδομένων, η αξιολόγηση ιδιοτήτων μετάδοσης μέσω συντελεστών ανισομορφίας καταδεικνύει ότι βελτιστοποιημένες δομές αναφοράς μειώνουν τις αποκλίσεις χρονισμού σε ποσοστό άνω του 10% [22].

Σε προηγμένες αρχιτεκτονικές παρακολούθησης, τα δεδομένα αισθητήρων ενσωματώνονται σε περιβάλλοντα SCADA και εικονικής προσομοίωσης, επιτρέποντας τόσο απομακρυσμένη εποπτεία όσο και προληπτική διάγνωση. Η εφαρμογή χαμηλού κόστους IoT–SCADA υποδομών έχει αποδειχθεί ότι μειώνει τις δαπάνες καλωδίωσης και συντήρησης κατά 12–18% σε βιομηχανικά περιβάλλοντα, διατηρώντας παράλληλα υψηλή ακρίβεια συλλογής σημάτων [7]. Σε εκπαιδευτικές και ερευνητικές πλατφόρμες εικονικής προσομοίωσης ανελευστήρων, η ενσωμάτωση μοντέλων αισθητήρων επιτρέπει την επαλήθευση σεναρίων αστοχίας πριν από τη φυσική εγκατάσταση, περιορίζοντας το κόστος δοκιμών και επανασχεδιασμού [4], [12]. Παράλληλα, προσεγγίσεις βασισμένες σε ψηφιακό δίδυμο καθιστούν εφικτή την αντιστοίχιση πραγματικών σημάτων με προσομοιωμένα πρότυπα, ενισχύοντας τη διαγνωστική ακρίβεια και την προληπτική συντήρηση [10].

## 2.5 Συστήματα Ασφαλείας

### 2.5.1 Ηλεκτρομαγνητικό φρένο

Ο ανελευστήρας πρέπει να είναι εφοδιασμένος με σύστημα πέδησης. Το φρένο χρησιμοποιείται για την ακινητοποίηση του ανελευστήρα. Στην πιο κλασική του μορφή αποτελείται από ένα ηλεκτρομαγνήτη, δύο μπράτσα επενδυμένα εσωτερικά με φερμουίτ και ένα σύστημα μοχλών, εικόνα 2.15.

Όταν ο ηλεκτρομαγνήτης δεν τροφοδοτείται, τότε τα μπράτσα εφαρμόζουν μια ροπή πέδησης στο τύμπανο του άξονα και ο ανελκυστήρας ακινητοποιείται. Αυτή η ροπή εξασφαλίζεται με τη βοήθεια δύο ελατηρίων και ενός περικοχλίου (παξιμάδι) ρύθμισης. Όταν ο ηλεκτρομαγνήτης τροφοδοτείται ανοίγουν τα μπράτσα και ελευθερώνεται το τύμπανο.

Το φρένο του ανελκυστήρα κλείνει αυτόματα σε περίπτωση διακοπής ρεύματος, ή βλάβης του κυκλώματος ελέγχου. Για μεγαλύτερη ασφάλεια οι επαφές που τροφοδοτούν τον ηλεκτρομαγνήτη του φρένου, βρίσκονται στο ρελέ ισχύος που τροφοδοτεί τον κινητήρα.

Το φρένο ρυθμίζεται με τέτοιο τρόπο, ώστε σε περίπτωση διακοπής του ρεύματος να ακινητοποιεί το θάλαμο κατά την κάθοδο του με φορτίο μεγαλύτερο κατά 25% του ονομαστικού του. Στους ανελκυστήρες μεγάλων ταχυτήτων και ρυθμιζόμενων στροφών, το φρένο ακινητοποιεί μόνο τον ανελκυστήρα αφού αυτός σταματήσει. Το φρενάρισμα γίνεται ηλεκτρομαγνητικά με την ανάπτυξη αντίρροπων μαγνητικών πεδίων προς το πεδίο που περιστρέφει το δρομέα. Η λειτουργία του φρένου στους ανελκυστήρες στηρίζεται στην ύπαρξη ωστικών ελατηρίων με τα σημεία άρθρωσης να βρίσκονται στη φορά της δύναμης τριβής. Λόγω του κινδύνου της θραύσης του ελατηρίου, πολλοί κατασκευαστές χρησιμοποιούν δύο παράλληλα ελατήρια έλξης.



**Εικόνα 2.15 Φρένα ανελκυστήρα**

### **2.5.2 Φρένο ασφαλείας – ασφαλιστικής διάταξης αρπάγης**

Ο ανελκυστήρας θα πρέπει να διαθέτει ένα αυτόματο σύστημα πέδησης που ενεργοποιείται σε περίπτωση βλάβης είτε του συστήματος πέδησης είτε του συρματόσχοινου, την λεγόμενη αρπάγη, εικόνα 2.16.

Η διάταξη αρπάγης ανιχνεύει εάν ο ανελκυστήρας κατεβαίνει ταχύτερα από το επιτρεπόμενο. Σε αυτή την περίπτωση, η αρπάγη ενεργοποιεί φρένα κατά μήκος των κάθετων σιδηροτροχιών του ανελκυστήρα, επιβραδύνοντάς τον γρήγορα, αλλά με τρόπο που να αποφεύγεται η απότομη ακινητοποίηση και τυχόν ατυχήματα.

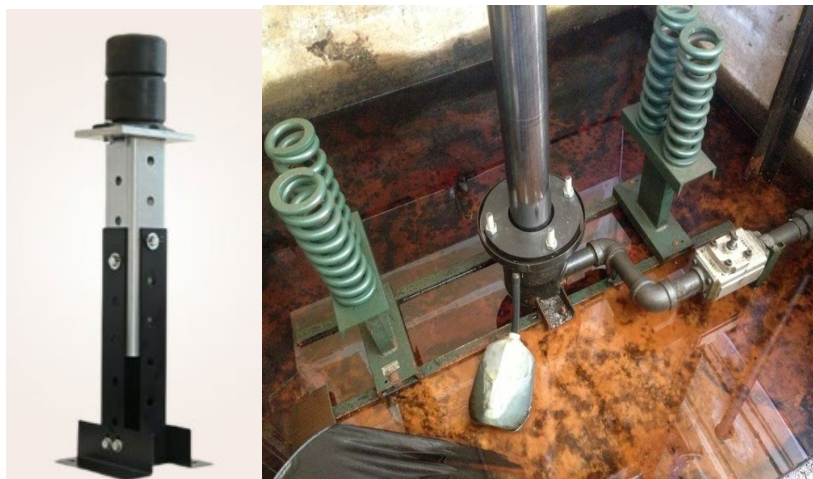
Η συσκευή αρπάγης είναι μια μηχανική διάταξη που έχει σκοπό να ακινητοποιεί και να συγκρατεί σταθερά το θάλαμο ή το αντίβαρο στους οδηγούς, σε περίπτωση που ο ανελκυστήρας υπερβεί ένα συγκεκριμένο όριο ταχύτητας κατά την κάθοδο ή σε περίπτωση θραύσης των μέσων ανάρτησης, όπως τα συρματόσχοινα. Το αντίβαρο είναι επίσης εφοδιασμένο με συσκευή αρπάγης. Η απελευθέρωση της συσκευής αρπάγης επιτρέπεται μόνο κατά την ανύψωση του θαλάμου ή του αντίβαρου.



**Εικόνα 2.16 Αρπάγη**

### 2.5.3 Προσκρουστήρες

Οι προσκρουστήρες τοποθετούνται στο κατώτατο σημείο της διαδρομής του θαλάμου και του αντίβαρου. Το σημείο λειτουργίας του προσκρουστήρα, κάτω από την προβολή του θαλάμου, θα πρέπει να έχει ένα εμπόδιο με ύψος που να πληροί τις απαιτήσεις του σχετικού κανονισμού. Η απορρόφηση ενέργειας από τους προσκρουστήρες απαιτεί ότι ο θάλαμος θα πρέπει να ακινητοποιείται πλήρως, με επιβράδυνση μικρότερη από την επιτάχυνση της βαρύτητας.



**Εικόνα 2.17 Προσκρουστήρες θαλάμου**

### 2.5.4 PLC λειτουργίες Ασφαλείας

Η διασφάλιση της λειτουργικής ακεραιότητας ενός ανελκυστήρα δεν περιορίζεται σε παθητικά μηχανικά μέσα προστασίας, αλλά συγκροτεί ένα πολυεπίπεδο πλαίσιο ηλεκτρομηχανολογικού ελέγχου, στο οποίο ο προγραμματιζόμενος λογικός ελεγκτής λειτουργεί ως κεντρικός κόμβος επιτήρησης, λήψης αποφάσεων και ενεργοποίησης μηχανισμών αποτροπής κινδύνου. Στα σύγχρονα συστήματα, η ενσωμάτωση PLC επιτρέπει τον συνεχή έλεγχο σημάτων από αισθητήρες ταχύτητας, ορίου διαδρομής, κλειδώματος θυρών και θερμικής προστασίας κινητήρα, όπως μπορούμε να δούμε συνοπτικά στην εικόνα 2.18, με χρόνους κύκλου σάρωσης της τάξης των λίγων milliseconds, διασφαλίζοντας άμεση αντίδραση σε αποκλίσεις λειτουργίας [1], [16].

Η αρχιτεκτονική αυτή υπερτερεί των συμβατικών ρελεδοκυκλωμάτων, καθώς επιτρέπει διαγνωστικό έλεγχο σε πραγματικό χρόνο, καταγραφή σφαλμάτων και λογική διασταύρωσης σημάτων (cross-checking), περιορίζοντας την πιθανότητα ψευδών εντολών και βλαβών λόγω ασυμβατότητας επαφών.

Παράλληλα, ιδιαίτερη σημασία αποκτά η διαμόρφωση συστημάτων έκτακτης ανάγκης που ενεργοποιούνται σε περιπτώσεις διακοπής τροφοδοσίας, υπερτάχυνσης ή απώλειας ελέγχου θέσης. Η σχεδίαση τέτοιων υποσυστημάτων, όπως παρουσιάζεται σε προσομοιώσεις βασισμένες στο TIA Portal, περιλαμβάνει εφεδρική τροφοδοσία, λογική αυτόματης στάσης στον πλησιέστερο όροφο και ασφαλή απελευθέρωση θυρών, μειώνοντας τον χρόνο εγκλωβισμού επιβατών και περιορίζοντας λειτουργικούς κινδύνους [15]. Σε πολυώροφα πειραματικά συστήματα, η χρήση PLC επέδειξε σταθερή απόκριση σε συνθήκες ταυτόχρονης κλήσης και σφάλματος αισθητήρα, με διατήρηση της λειτουργίας ασφαλείας ακόμη και σε περιβάλλοντα με αυξημένες ηλεκτρομαγνητικές παρεμβολές [14], [21]. Η υλοποίηση λογικής fail-safe επιτυγχάνεται μέσω κανονικά κλειστών επαφών ασφαλείας, ώστε οποιαδήποτε απώλεια σήματος να οδηγεί αυτόματα σε παύση κίνησης.

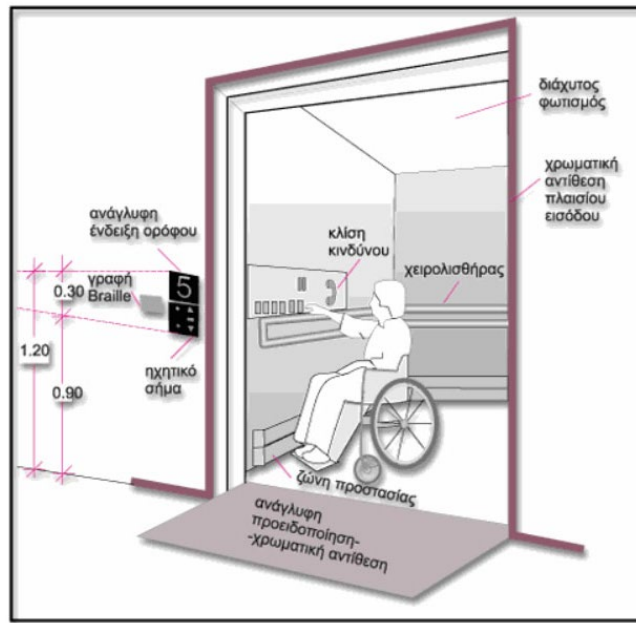
Η σύγχρονη τάση εστιάζει στην ασφαλή επικοινωνία και την προστασία δεδομένων των συστημάτων ελέγχου. Η ενσωμάτωση τεχνικών τεχνητής νοημοσύνης σε PLC για ανίχνευση ανωμαλιών και αποτροπή κυβερνοεπιθέσεων ενισχύει το επίπεδο αξιοπιστίας, ιδίως σε διασυνδεδεμένα περιβάλλοντα SCADA, όπου η μετάδοση δεδομένων ενέχει κινδύνους αλλοίωσης ή μη εξουσιοδοτημένης πρόσβασης [11]; [23]. Μηχανισμοί ελαφριάς κρυπτογράφησης και ιεραρχημένης αρχιτεκτονικής τριών επιπέδων συμβάλλουν στη διατήρηση της ακεραιότητας σημάτων ελέγχου χωρίς να επιβαρύνουν τον χρόνο απόκρισης του συστήματος [24]. Τεχνικές αντιστάθμισης διαταραχών σε ασύρματα κανάλια μετάδοσης δεδομένων μπορούν να μειώσουν τα σφάλματα επικοινωνίας, ενισχύοντας τη σταθερότητα της εποπτείας [19].



Εικόνα 2.18 PLC συστήματα ασφαλείας

### 2.5.5 Ασφαλείας και Προσβασιμότητα

Τέλος, η ασφάλεια δεν περιορίζεται σε τεχνικά κριτήρια, αλλά επεκτείνεται και σε ζητήματα προσβασιμότητας και εργονομίας. Οι απαιτήσεις ατόμων με προβλήματα ακοής, για παράδειγμα, επιβάλλουν οπτικά και απτικά σήματα συναγερμού, ενσωματωμένα στον έλεγχο του ανελκυστήρα, ώστε να διασφαλίζεται ισότιμη ενημέρωση σε καταστάσεις κινδύνου [9]. Στην εικόνα 2.19 βλέπουμε έναν ανελκυστήρα που είναι ειδικά κατασκευασμένα με δυνατότητα πρόσβασης- χρήσης από άτομα ΑΜΕΑ με ασφάλεια.



Εικόνα 2.19 Ανελκυστήρας με προδιαγραφές για ΑΜΕΑ

Η συνδυασμένη αξιοποίηση προσομοιωτικών πλατφορμών και ψηφιακών διδύμων επιτρέπει την προληπτική αξιολόγηση σεναρίων αστοχίας πριν από τη φυσική εγκατάσταση, μειώνοντας το τεχνοοικονομικό κόστος ανασχεδιασμού και αυξάνοντας τον δείκτη αξιοπιστίας του τελικού συστήματος [4], [10]. Κατ' αυτόν τον τρόπο, τα συστήματα ασφαλείας ανελκυστήρων μετασχηματίζονται σε ολοκληρωμένες, ευφυείς και οικονομικά βιώσιμες υποδομές ελέγχου.

## ΚΕΦΑΛΑΙΟ 3: Προγραμματιζόμενοι Λογικοί Ελεγκτές (PLC)

### 3.1 Εισαγωγή στα PLC

Η καθιέρωση των Προγραμματιζόμενων Λογικών Ελεγκτών (Programmable Logic Controllers – PLC) συνιστά καθοριστικό σταθμό στη μετάβαση από τα συμβατικά ηλεκτρομηχανικά κυκλώματα ελέγχου σε ευέλικτες, επαναπρογραμματιζόμενες και υψηλής αξιοπιστίας αρχιτεκτονικές αυτοματισμού. Σε βιομηχανικές εφαρμογές, όπου οι απαιτήσεις διαθεσιμότητας υπερβαίνουν συχνά το 99,5% ετησίως, η χρήση PLC μειώνει δραστικά τον χρόνο εντοπισμού και αποκατάστασης βλαβών, περιορίζοντας το κόστος μη προγραμματισμένων διακοπών λειτουργίας [1], [16]. Η αντικατάσταση εκτεταμένων ρελεδοκυκλωμάτων από λογισμικό δομημένο σύμφωνα με το πρότυπο IEC 61131-3 επιτρέπει τροποποιήσεις στον έλεγχο χωρίς επεμβάσεις καλωδίωσης, στοιχείο που μεταφράζεται σε εξοικονόμηση έως και 20–30% στο κόστος ανασχεδιασμού γραμμών παραγωγής ή ηλεκτρομηχανολογικών εγκαταστάσεων.

Η σύγχρονη τεχνολογία PLC ενσωματώνει επεξεργαστές υψηλής ταχύτητας, μνήμες επαρκούς χωρητικότητας και επεκτάσιμες μονάδες εισόδων/εξόδων, επιτρέποντας την ταυτόχρονη διαχείριση εκατοντάδων σημάτων με χρόνους κύκλου σάρωσης της τάξης των millisecond. Σε εφαρμογές ανελκυστήρων, όπου απαιτείται άμεση απόκριση σε μεταβολές θέσης και εντολές κλήσης, η σταθερότητα του κύκλου σάρωσης διασφαλίζει ακρίβεια ελέγχου και αποτρέπει συνθήκες ασυγχρονισμού[5]. Πρακτικές υλοποιήσεις πολυώροφων συστημάτων με PLC κατέδειξαν αξιόπιστη επεξεργασία πολλαπλών ταυτόχρονων σημάτων χωρίς απώλειες δεδομένων, ακόμη και σε περιβάλλοντα με ηλεκτρομαγνητικές παρεμβολές, επιβεβαιώνοντας τη βιομηχανική ανθεκτικότητα της πλατφόρμας [14], [21].

Παράλληλα, η εξέλιξη των PLC προς διασυνδεδεμένες και ευφυείς υποδομές ενισχύει τον ρόλο τους πέραν του τοπικού ελέγχου. Η ενσωμάτωση σε περιβάλλοντα SCADA επιτρέπει κεντρική εποπτεία, καταγραφή ιστορικών δεδομένων και ανάλυση λειτουργικών δεικτών απόδοσης, με τεχνοοικονομικά οφέλη που περιλαμβάνουν μείωση κόστους καλωδίωσης και βελτιστοποίηση συντήρησης [7], [23]. Επιπλέον, εφαρμογές αναγεννητικής πέδησης σε κινητήρες μόνιμου μαγνήτη, όταν συνδυάζονται με ελεγκτές PLC και κοινό δίαυλο DC, συμβάλλουν σε ανάκτηση ενέργειας και περιορισμό κατανάλωσης, ενισχύοντας την ενεργειακή αποδοτικότητα εγκαταστάσεων ανύψωσης [6].

Σε ρυθμιζόμενα περιβάλλοντα παραγωγής, μοντελοκεντρικές προσεγγίσεις αυτόματης επικύρωσης και παραγωγής κώδικα μειώνουν τα σφάλματα προγραμματισμού και επιταχύνουν τον κύκλο πιστοποίησης εξοπλισμού [13].

### 3.2 Δομή και Αρχιτεκτονική PLC

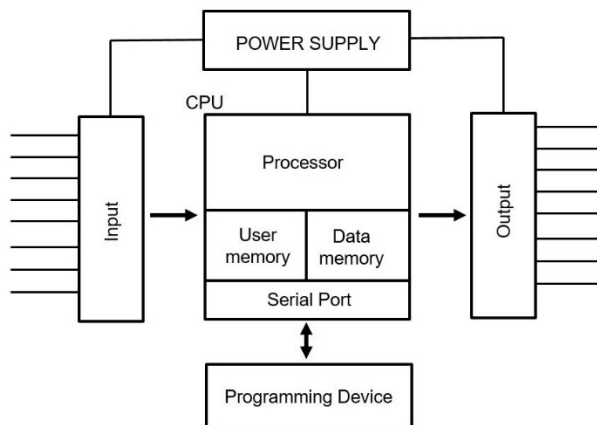
Η εσωτερική δομή ενός PLC, σχήμα 3.1, συγκροτεί μια ολοκληρωμένη αρχιτεκτονική ελέγχου, στην οποία η Κεντρική Μονάδα Επεξεργασίας (CPU), τα υποσυστήματα μνήμης, οι μονάδες εισόδων/εξόδων και το κύκλωμα τροφοδοσίας συνεργάζονται με αυστηρά καθορισμένη χρονική ακολουθία. Στον πυρήνα του συστήματος, η CPU εκτελεί τον κύκλο σάρωσης, ο οποίος περιλαμβάνει ανάγνωση εισόδων, επεξεργασία λογικού προγράμματος και ενημέρωση εξόδων μέσα σε χρονικό διάστημα που συχνά κυμαίνεται από 1 έως 10 ms σε βιομηχανικές εφαρμογές μεσαίας κλίμακας [1]. Η υπολογιστική της ικανότητα καθορίζει τον μέγιστο αριθμό διαχειρίσιμων σημάτων και την πολυπλοκότητα των αλγορίθμων ελέγχου, ενώ σε προηγμένες υλοποιήσεις η ενσωμάτωση διαδικασιοκεντρικών επεκτάσεων στη

Structured Text ενισχύει τη διαφάνεια και τη δυνατότητα επαλήθευσης του κώδικα, ιδίως σε εφαρμογές με υψηλές κανονιστικές απαιτήσεις [3], [13].

Το υποσύστημα μνήμης διακρίνεται σε πτητική και μη πτητική περιοχή, εξυπηρετώντας αφενός την προσωρινή αποθήκευση μεταβλητών κατά τον κύκλο εκτέλεσης και αφετέρου τη μόνιμη καταγραφή του προγράμματος και κρίσιμων παραμέτρων. Η αξιοπιστία της μνήμης επηρεάζει άμεσα τη διαθεσιμότητα του ελεγκτή, καθώς απώλεια δεδομένων μπορεί να επιφέρει διακοπή λειτουργίας εγκαταστάσεων με σημαντικό οικονομικό αντίκτυπο. Σε περιβάλλοντα όπου το κόστος ακινητοποίησης παραγωγικής μονάδας υπερβαίνει τα αρκετά χιλιάδες ευρώ ανά ώρα, η σταθερότητα αποθήκευσης και η δυνατότητα εφεδρικής ανάκτησης δεδομένων αποτελούν κρίσιμα τεχνοοικονομικά κριτήρια επιλογής PLC [16]. Επιπλέον, η διασύνδεση με συστήματα SCADA επιτρέπει εξωτερική αρχειοθέτηση λειτουργικών δεδομένων, ενισχύοντας τη μακροχρόνια ανάλυση απόδοσης και τη λήψη αποφάσεων βάσει ιστορικών καταγραφών [7], [23].

Εξίσου καθοριστικό ρόλο διαδραματίζουν οι μονάδες εισόδων και εξόδων, οι οποίες αποτελούν τη διεπαφή μεταξύ ψηφιακού ελεγκτή και φυσικού πεδίου. Οι ψηφιακές εισοδοι, συνήθως στα 24 VDC, συλλέγουν σήματα από αισθητήρες και διακόπτες, ενώ οι αναλογικές εισοδοι επιτρέπουν μέτρηση μεγεθών όπως ταχύτητα ή ρεύμα κινητήρα, όπως απεικονίζεται ενδεικτικά στην εικόνα 3.1. Στην πράξη, υλοποιήσεις ανελκυστήρων με PLC απέδειξαν ότι η ορθολογική κατανομή I/O μειώνει την πολυπλοκότητα καλωδίωσης και διευκολύνει τη συντήρηση, περιορίζοντας τον χρόνο διάγνωσης βλαβών [14], [21]. Σε συστήματα πολλαπλών θαλάμων, η αρχιτεκτονική I/O επεκτείνεται ώστε να υποστηρίζει ταυτόχρονη επεξεργασία δεδομένων θέσης και κλήσεων, βελτιστοποιώντας την επιχειρησιακή απόδοση [2].

Τέλος, το κύκλωμα τροφοδοσίας, εικόνα 3.2, εξασφαλίζει σταθερή και αδιάλειπτη λειτουργία όλων των υπομονάδων, μετατρέποντας την παρεχόμενη εναλλασσόμενη τάση σε ρυθμισμένη συνεχούς μορφής. Η ποιότητα της τροφοδοσίας επηρεάζει άμεσα τη μακροχρόνια αξιοπιστία του ελεγκτή, ιδίως σε περιβάλλοντα με ηλεκτρικές διακυμάνσεις. Σε εφαρμογές όπου συνδυάζεται έλεγχος κινητήρων μόνιμου μαγνήτη και αναγεννητική πέδηση μέσω κοινού διαύλου, η σταθερότητα της ενεργειακής διαχείρισης συμβάλλει τόσο στη μείωση κατανάλωσης όσο και στη διατήρηση της λειτουργικής συνέχειας του PLC [6].

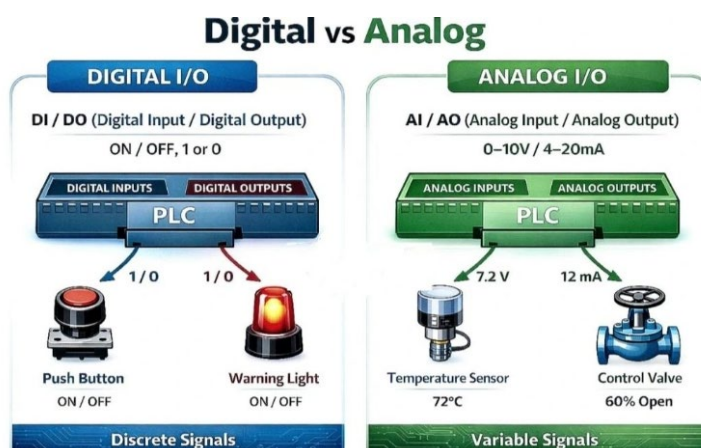


Σχήμα 3.1 Διάγραμμα Δομής PLC

### 3.3 Τύποι Σημάτων (Digital / Analog)

Τα ψηφιακά σήματα, δυαδικής μορφής (0/1), χρησιμοποιούνται εκτενώς για την απεικόνιση διακριτών καταστάσεων όπως ενεργοποίηση διακόπτη, ανίχνευση ορόφου ή κατάσταση κλειδώματος θύρας, όπως μπορούμε να δούμε στην εικόνα 3.1. Η αξιοπιστία τους εξαρτάται από την ορθή ηλεκτρική διαμόρφωση, ώστε να επιτυγχάνεται ανθεκτικότητα σε παρεμβολές και σταθερότητα ανάγνωσης από την CPU [1]. Σε εφαρμογές ανελκυστήρων η ορθολογική κατανομή ψηφιακών εισόδων και εξόδων μείωσε την πολυπλοκότητα καλωδίωσης και επέτρεψε ταχύτερη διάγνωση βλαβών, περιορίζοντας τον χρόνο αποκατάστασης κατά λειτουργικούς κύκλους δοκιμών [14].

Τα αναλογικά σήματα, μεταφέρουν συνεχή πληροφορία, όπως ταχύτητα κινητήρα, ρεύμα ή θέση μέσω αισθητήρων με έξοδο 0–10 V ή 4–20 mA, όπως φαίνεται στην εικόνα 3.1. Η μετατροπή τους σε ψηφιακή μορφή μέσω μετατροπέων A/D εντός του PLC επιτρέπει ακριβή υπολογισμό μεταβλητών ελέγχου, στοιχείο κρίσιμο σε συστήματα όπου απαιτείται ομαλή επιτάχυνση και επιβράδυνση θαλάμου. Σε περιβάλλοντα όπου εφαρμόζεται αναγεννητική πέδηση κινητήρων μόνιμου μαγνήτη, η συνεχής παρακολούθηση ρευμάτων και τάσεων μέσω αναλογικών καναλιών συμβάλλει στην ενεργειακή βελτιστοποίηση και στην επιστροφή ισχύος στον κοινό δίαυλο, μειώνοντας τη συνολική κατανάλωση [6]. Η ορθή βαθμονόμηση αυτών των καναλιών επηρεάζει άμεσα την απόδοση, καθώς σφάλματα μέτρησης άνω του 2% μπορούν να οδηγήσουν σε αποκλίσεις ταχύτητας και αυξημένη μηχανική καταπόνηση.

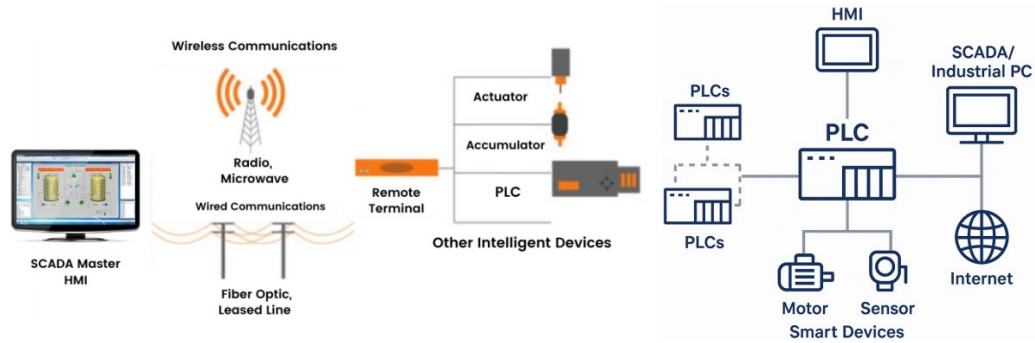


Εικόνα 3.1 Ψηφιακά και Αναλογικά Σήματα



Εικόνα 3.2 Τροφοδοτικό

Η ποιότητα μετάδοσης τόσο των ψηφιακών όσο και των αναλογικών σημάτων εξαρτάται από τη δομή του δικτύου και τις ιδιότητες της διασύνδεσης. Η ανάλυση χαρακτηριστικών μετάδοσης μέσω συντελεστών ανισομορφίας αποδεικνύει ότι η ορθολογική τοπολογία μειώνει καθυστερήσεις και ασυμμετρίες στη ροή δεδομένων, ενισχύοντας τη συνολική σταθερότητα του ελέγχου [22]. Σε διασυνδεδεμένα περιβάλλοντα SCADA, η απομακρυσμένη συλλογή και ανάλυση σημάτων επιτρέπει κεντρική επιτήρηση και βελτιστοποίηση λειτουργικών δεικτών, επιτυγχάνοντας μείωση κόστους εγκατάστασης και συντήρησης μέσω IoT αρχιτεκτονικών [7], [23]. Στην εικόνα 3.3 μπορούμε να δούμε μια διασύνδεση PLC με SCADA.



**Εικόνα 3.3 Διασύνδεση SCADA με PLC**

Επιπλέον, η επεξεργασία σύνθετων ροών δεδομένων ενισχύεται με την εφαρμογή προηγμένων αλγοριθμικών προσεγγίσεων, όπου η αξιοποίηση τεχνικών μάθησης συνόλου επιτρέπει πρόβλεψη βραχυπρόθεσμων διακυμάνσεων λειτουργίας και έγκαιρη ανίχνευση ανωμαλιών σε αναλογικά σήματα [8]. Η ανάπτυξη και προσομοίωση τέτοιων σεναρίων σε εκπαιδευτικές και ερευνητικές πλατφόρμες PLC συμβάλλει στην κατανόηση της δυναμικής αλληλεπίδρασης μεταξύ τύπων σημάτων και λογικής ελέγχου, μειώνοντας το κόστος φυσικών δοκιμών και επιταχύνοντας τη βελτιστοποίηση του προγράμματος [4], [17].

**Πίνακας 3.1: Συγκριτική Αποτύπωση Ψηφιακών και Αναλογικών Σημάτων σε Συστήματα PLC**

| Χαρακτηριστικό          | Ψηφιακά Σήματα (Digital)        | Αναλογικά Σήματα (Analog)                  |
|-------------------------|---------------------------------|--|
| Μορφή Δεδομένων         | Δυαδική (0/1)                   | Συνεχής μεταβολή τιμής                     |
| Τυπικά Επίπεδα Τάσης    | 0 V / 24 V DC                   | 0–10 V ή 4–20 mA                           |
| Τύπος Πληροφορίας       | Διακριτές καταστάσεις (ON/OFF)  | Μετρήσιμα φυσικά μεγέθη                    |
| Παραδείγματα Εφαρμογής  | Διακόπτες ορόφων, κλειδιά θυρών | Ταχύτητα κινητήρα, ρεύμα, θέση             |
| Ακρίβεια Μέτρησης       | Απόλυτη λογική κατάσταση        | Εξαρτάται από ανάλυση A/D (π.χ. 12–16 bit) |
| Ευαισθησία σε Θόρυβο    | Χαμηλή (με σωστή γείωση)        | Μέτρια έως υψηλή                           |
| Απαιτούμενη Επεξεργασία | Λογικές πράξεις                 | Μετατροπή A/D και μαθηματική επεξεργασία   |
| Κόστος Υλοποίησης       | Χαμηλότερο                      | Υψηλότερο λόγω μετατροπέων                 |
| Χρόνος Απόκρισης        | Πολύ μικρός                     | Εξαρτάται από δειγματοληψία                |
| Ρόλος στον Έλεγχο       | Ακολουθιακή λογική              | Ρυθμιστικός/αναλογικός έλεγχος             |

### 3.4 Κύκλος Σάρωσης (Scan Cycle)

Ο κύκλος σάρωσης (scan cycle) συνιστά τον θεμελιώδη μηχανισμό λειτουργίας ενός Προγραμματιζόμενου Λογικού Ελεγκτή, καθώς καθορίζει τον τρόπο με τον οποίο το σύστημα συλλέγει δεδομένα, επεξεργάζεται λογικές συνθήκες και αποστέλλει εντολές στο πεδίο. Κατά τη διάρκεια κάθε κύκλου, η CPU εκτελεί διαδοχικά τρία βασικά στάδια: ανάγνωση εισόδων, εκτέλεση του προγράμματος χρήστη και ενημέρωση εξόδων. Η συνολική διάρκεια αυτού του βρόχου, η οποία σε βιομηχανικές διατάξεις κυμαίνεται συνήθως μεταξύ 1 και 10 ms, επηρεάζει άμεσα τη χρονική ακρίβεια και τη σταθερότητα του ελέγχου [1]. Σε εφαρμογές ανελκυστήρων, όπου η καθυστέρηση απόκρισης σε σήματα θέσης ή ασφαλείας πρέπει να παραμένει κάτω από ορισμένα όρια για την αποφυγή δυναμικών αστοχιών, η βελτιστοποίηση του scan cycle αποτελεί κρίσιμο τεχνικό παράγοντα [5].

Η διάρκεια του κύκλου δεν εξαρτάται αποκλειστικά από την ταχύτητα του επεξεργαστή, αλλά και από την πολυπλοκότητα του προγράμματος, τον αριθμό μεταβλητών και τη δομή των επικοινωνιακών διεπαφών. Σε περιβάλλοντα όπου χρησιμοποιούνται επεκτάσεις της Structured Text για διαδικασιοκεντρικό έλεγχο, η ορθολογική δόμηση του κώδικα συμβάλλει στη μείωση περιττών εντολών και στη σταθεροποίηση του χρόνου εκτέλεσης[3]. Παράλληλα, μοντελοκεντρικές προσεγγίσεις αυτόματης επικύρωσης κώδικα επιτρέπουν τον εντοπισμό λογικών βρόχων που αυξάνουν αδικαιολόγητα τον χρόνο σάρωσης, μειώνοντας τον κίνδυνο καθυστερήσεων σε κρίσιμες διεργασίες [13]. Σε οικονομικούς όρους, η μείωση ακόμη και 2–3 ms ανά κύκλο σε συστήματα υψηλής συχνότητας μεταβολών μπορεί να βελτιώσει τη συνολική αποδοτικότητα, ιδίως όταν το PLC διαχειρίζεται εκατοντάδες κύκλους ανά δευτερόλεπτο.

Ιδιαίτερο ενδιαφέρον παρουσιάζει η επίδραση του κύκλου σάρωσης σε διασυνδεδεμένα περιβάλλοντα SCADA, όπου τα δεδομένα συλλέγονται και μεταδίδονται σε ανώτερα επίπεδα εποπτείας. Η καθυστέρηση στη μεταφορά ή η ασυμμετρία δικτυακής τοπολογίας μπορεί να επιβαρύνει τη συνολική χρονική συμπεριφορά του συστήματος, επηρεάζοντας την αξιοπιστία της πληροφορίας [22]. Η υιοθέτηση αποδοτικών IoT αρχιτεκτονικών για κεντρική παρακολούθηση έχει αποδειχθεί ότι μειώνει την καθυστέρηση μετάδοσης και το κόστος καλωδίωσης, διατηρώντας παράλληλα ικανοποιητική συχνότητα ενημέρωσης δεδομένων [7], [23]. Σε συστήματα πολλαπλών ανελκυστήρων, όπου η κατανομή κλήσεων βασίζεται σε πραγματικό χρόνο υπολογισμούς θέσης, η σταθερότητα του scan cycle καθίσταται απαραίτητη για τη διατήρηση ομοιόμορφης απόκρισης [2].

**Πίνακας 3.2: Βασικά Στάδια και Χαρακτηριστικά Κύκλου Σάρωσης (Scan Cycle) PLC**

| Στάδιο Κύκλου         | Περιγραφή Λειτουργίας                                  | Τυπική Χρονική Συμβολή | Τεχνική Σημασία                   |
|-----------------------|--|------------------------|-----------------------------------|
| Ανάγνωση Εισόδων      | Καταγραφή ψηφιακών/αναλογικών σημάτων σε εικόνα μνήμης | 10–20% του κύκλου      | Ακρίβεια δεδομένων πεδίου         |
| Εκτέλεση Προγράμματος | Επεξεργασία λογικών εντολών και αλγορίθμων             | 60–80% του κύκλου      | Καθορίζει τη συνολική καθυστέρηση |
| Ενημέρωση Εξόδων      | Αποστολή εντολών σε ενεργοποιητές                      | 10–20% του κύκλου      | Άμεση απόκριση συστήματος         |
| Συνολικός Χρόνος Scan | Πλήρης επανάληψη βρόχου                                | 1–10 ms (τυπικά)       | Επηρεάζει δυναμική συμπεριφορά    |

### 3.5 Γλώσσες Προγραμματισμού (IEC 61131-3)

#### 3.5.1 Ladder Diagram (LD)

Η γλώσσα Ladder Diagram (LD), όπως ορίζεται στο πρότυπο IEC 61131-3, αποτελεί μία από τις πλέον διαδεδομένες μεθοδολογίες προγραμματισμού PLC, καθώς προσομοιάζει τη λογική των κλασικών ηλεκτρολογικών κυκλωμάτων με ρελέ, διευκολύνοντας τη μετάβαση από τον συμβατικό αυτοματισμό σε ψηφιακές αρχιτεκτονικές. Η γραφική της δομή, βασισμένη σε «σκαλοπάτια» λογικών επαφών και πηνίων, επιτρέπει σαφή απεικόνιση διαδοχικών συνθηκών ενεργοποίησης, στοιχείο ιδιαίτερα χρήσιμο σε εφαρμογές ανελκυστήρων όπου η ακολουθιακή λογική καθορίζει κινήσεις, κλειδώματα και αλληλεξαρτήσεις ασφαλείας [5]. Σε υλοποιήσεις πολυώροφων συστημάτων με PLC, η χρήση LD διευκόλυνε την ταχεία αποσφαλμάτωση και περιόρισε τον χρόνο ανάπτυξης, βελτιώνοντας την τεχνικοοικονομική αποδοτικότητα του έργου [14]. Επιπλέον, η σαφήνεια της γραφικής αναπαράστασης ενισχύει τη δυνατότητα ελέγχου και επικύρωσης λογικής σε ρυθμιζόμενα περιβάλλοντα, όπου απαιτείται τεκμηριωμένη διαφάνεια προγράμματος [13].

#### 3.5.2 Function Block Diagram (FBD)

Η γλώσσα Function Block Diagram (FBD), όπως καθορίζεται από το πρότυπο IEC 61131-3, βασίζεται σε γραφική διασύνδεση λειτουργικών μπλοκ που αναπαριστούν μαθηματικές, λογικές και ρυθμιστικές πράξεις, προσφέροντας σαφή εποπτεία της ροής σήματος. Σε εφαρμογές αυτοματισμού με αυξημένες απαιτήσεις ελέγχου διεργασιών, η FBD διευκολύνει την οργάνωση σύνθετων αλγορίθμων, όπως ρυθμιστών ταχύτητας ή διαχείρισης ενέργειας, επιτρέποντας παραμετροποίηση σε πραγματικό χρόνο [1]. Σε συστήματα ανελκυστήρων με πολλαπλές εισόδους και μεταβλητές κατάσταση, η χρήση διασυνδεδεμένων μπλοκ συμβάλλει σε ομαλή επιτάχυνση και ενεργειακή βελτιστοποίηση, ιδιαίτερα όταν εφαρμόζεται αναγεννητική πέδηση [6]. Παράλληλα, η δομημένη οπτικοποίηση της FBD υποστηρίζει ευκολότερη επικύρωση λογικής και διαγνωστικό έλεγχο σε περιβάλλοντα προσομοίωσης και SCADA [7], ενισχύοντας τη λειτουργική διαφάνεια και την τεχνικοοικονομική αποδοτικότητα του συστήματος.

#### 3.5.3 Structured Text (ST)

Η γλώσσα Structured Text (ST), ενταγμένη στο πρότυπο IEC 61131-3, αποτελεί κειμενοκεντρική μορφή προγραμματισμού υψηλού επιπέδου, κατάλληλη για την υλοποίηση σύνθετων αλγορίθμων και μαθηματικών υπολογισμών σε PLC. Σε αντίθεση με τις γραφικές προσεγγίσεις, η ST επιτρέπει χρήση δομών επανάληψης, συνθηκών και συναρτήσεων, διευκολύνοντας την ανάπτυξη ρυθμιστικών και ενεργειακά βελτιστοποιημένων στρατηγικών ελέγχου [1]. Σε εφαρμογές όπου απαιτείται ακριβής διαχείριση πολλαπλών μεταβλητών, όπως σε ομαδικά συστήματα ανελκυστήρων, η δομημένη σύνταξη συμβάλλει στη μείωση λογικών σφαλμάτων και στη βελτίωση της υπολογιστικής απόδοσης [2]. Επιπλέον, διαδικασιοκεντρικές επεκτάσεις της ST ενισχύουν τη σαφήνεια και την επαληθευσσιμότητα του κώδικα, στοιχείο κρίσιμο σε περιβάλλοντα με αυξημένες κανονιστικές απαιτήσεις [3], υποστηρίζοντας παράλληλα αυτοματοποιημένες διαδικασίες ελέγχου και επικύρωσης [13].

#### 3.5.4 Sequential Function Chart (SFC)

Η γλώσσα Sequential Function Chart (SFC) του προτύπου IEC 61131-3 χρησιμοποιείται για τη μοντελοποίηση διακριτών σταδίων λειτουργίας μέσω βημάτων και μεταβάσεων, επιτρέποντας σαφή αποτύπωση ακολουθιακών διεργασιών. Σε συστήματα ανελκυστήρων, όπου η λειτουργία περιλαμβάνει διαδοχικές καταστάσεις όπως αναμονή, κίνηση, επιβράδυνση και άνοιγμα θυρών, η SFC προσφέρει οργανωμένη απεικόνιση της λογικής ροής και μειώνει την πιθανότητα ασυνεπειών κατά την υλοποίηση [5]. Επιπλέον, η ενσωμάτωση της SFC σε ψηφιακά δίδυμα επιτρέπει προσομοίωση σεναρίων πριν την εγκατάσταση, περιορίζοντας τεχνικούς κινδύνους και κόστος δοκιμών [10]. Μέσω εικονικών πλατφορμών εκπαίδευσης και ελέγχου, διασφαλίζεται επαλήθευση μεταβάσεων και βελτιστοποίηση χρόνων απόκρισης, στοιχείο κρίσιμο για πολυεπίπεδα συστήματα μεταφοράς [4].

### 3.6 Πλεονεκτήματα PLC σε Βιομηχανικές Εφαρμογές

Η αξιοποίηση Προγραμματιζόμενων Λογικών Ελεγκτών σε βιομηχανικές εγκαταστάσεις εδράζεται στη δυνατότητα αξιόπιστου και ντετερμινιστικού ελέγχου με χαμηλό χρόνο απόκρισης, χαρακτηριστικό που τους καθιστά καταλληλότερους από γενικής χρήσης υπολογιστικά συστήματα σε κρίσιμες διεργασίες. Η σταθερότητα του κύκλου σάρωσης και η ανθεκτικότητα σε ηλεκτρομαγνητικές παρεμβολές διασφαλίζουν συνεχή λειτουργία ακόμη και σε περιβάλλοντα υψηλών φορτίων, στοιχείο που τεκμηριώνεται σε μελέτες ηλεκτρικού αυτοματισμού βασισμένου σε PLC [1], [16]. Σε εφαρμογές ανελκυστήρων, όπου η αστοχία ελέγχου μπορεί να επιφέρει αυξημένο λειτουργικό κίνδυνο, η προβλεψιμότητα της λογικής εκτέλεσης αποτελεί κρίσιμο πλεονέκτημα [5].

Η επεκτασιμότητα και η διασυνδεσιμότητα με συστήματα SCADA ενισχύουν την επιχειρησιακή εποπτεία και τη λήψη αποφάσεων βάσει δεδομένων. Η ενσωμάτωση IoT αρχιτεκτονικών επιτρέπει κεντρική παρακολούθηση πολλαπλών μονάδων με μειωμένο κόστος καλωδίωσης και βελτιστοποιημένη συλλογή δεδομένων σε πραγματικό χρόνο, επιτυγχάνοντας ταυτόχρονα οικονομία κλίμακας [7]. Παράλληλα, ολοκληρωμένα πλαίσια SCADA διευκολύνουν την ανάλυση λειτουργικών δεικτών και τη βελτίωση παραγωγικής απόδοσης, στοιχείο που έχει αποδειχθεί σε απαιτητικές βιομηχανικές εφαρμογές [23]. Η δυνατότητα ενοποίησης αναγεννητικών συστημάτων οδήγησης, όπως σε ελέγχους PMSM, συμβάλλει επιπλέον στη μείωση ενεργειακών απωλειών και στην ανάκτηση ισχύος, βελτιώνοντας τον βαθμό απόδοσης εγκαταστάσεων μεταφοράς [6].

Η ασφάλεια δεδομένων και η ανθεκτικότητα σε κυβερνοαπειλές αποτελούν καθοριστικούς παράγοντες σύγχρονης βιομηχανικής υλοποίησης. Παράλληλα, ελαφριές αρχιτεκτονικές κρυπτογράφησης τριών επιπέδων προσφέρουν ισορροπία μεταξύ ασφάλειας και υπολογιστικής επιβάρυνσης, στοιχείο κρίσιμο για συστήματα με περιορισμένους πόρους [24].

**Πίνακας 3.3: Συνοπτική Παρουσίαση Πλεονεκτημάτων PLC σε Βιομηχανικές Εφαρμογές**

| Πλεονέκτημα                             | Τεχνική Διάσταση                                    | Τεχνοοικονομική Επίδραση                            |
|---|---|---|
| Ντετερμινιστική λειτουργία              | Σταθερός κύκλος σάρωσης, προβλέψιμη απόκριση        | Μείωση λειτουργικού ρίσκου και αστοχιών             |
| Ανθεκτικότητα σε βιομηχανικό περιβάλλον | Αντοχή σε Η/Μ παρεμβολές και διακυμάνσεις τάσης     | Μικρότερο κόστος συντήρησης                         |
| Επεκτασιμότητα I/O                      | Δυνατότητα προσθήκης μονάδων χωρίς ανασχεδιασμό     | Ευελιξία επένδυσης και κλιμάκωση συστήματος         |
| Διασύνδεση με SCADA/IoT                 | Πραγματικός χρόνος εποπτείας και συλλογής δεδομένων | Βελτιστοποίηση απόδοσης και ενεργειακής κατανάλωσης |
| Υποστήριξη ασφάλειας δεδομένων          | Ενσωματωμένοι μηχανισμοί προστασίας και ανίχνευσης  | Περιορισμός κυβερνοκινδύνων                         |
| Δυνατότητα προσομοίωσης                 | Ψηφιακά δίδυμα και εικονικός έλεγχος                | Μείωση κόστους δοκιμών και χρόνου υλοποίησης        |

## ΚΕΦΑΛΑΙΟ 4: Κανονισμοί και Πρότυπα Ασφαλείας Ανελκυστήρων

### 4.1 Ευρωπαϊκά Πρότυπα (EN 81)

Τα Ευρωπαϊκά Πρότυπα EN 81 συνιστούν το θεμελιώδες κανονιστικό πλαίσιο για τον σχεδιασμό, την εγκατάσταση και τη λειτουργία ανελκυστήρων εντός της Ευρωπαϊκής Ένωσης, καθορίζοντας συγκεκριμένες τεχνικές απαιτήσεις για μηχανική αντοχή, ηλεκτρική ασφάλεια και λειτουργική αξιοπιστία. Η συμμόρφωση με τις επιμέρους ενότητες του EN 81 προϋποθέτει τεκμηριωμένη αξιολόγηση κινδύνου και ενσωμάτωση διατάξεων προστασίας έναντι υπερταχύτητας, ανεξέλεγκτης κίνησης θαλάμου και αστοχίας θυρών, με έμφαση στη διασφάλιση της ανθρώπινης ζωής. Στο πλαίσιο αυτό, ο σχεδιασμός συστημάτων ελέγχου με PLC οφείλει να ενσωματώνει δοκιμασμένες λογικές ασφαλείας και διαδικασίες επικύρωσης, όπως επισημαίνεται σε μελέτες σχεδιασμού ανελκυστήρων βασισμένων σε PLC [5].

Η κανονιστική συμμόρφωση δεν περιορίζεται σε μηχανικές προβλέψεις αλλά επεκτείνεται σε ζητήματα ηλεκτρολογικής και λειτουργικής ακεραιότητας. Η ανάπτυξη συστημάτων ελέγχου οφείλει να τεκμηριώνεται μέσω μοντελοκεντρικών μεθόδων επαλήθευσης, οι οποίες μειώνουν τον κίνδυνο λογικών σφαλμάτων πριν από την εγκατάσταση, περιορίζοντας το κόστος επανασχεδιασμού και τις καθυστερήσεις πιστοποίησης [13]. Παράλληλα, η χρήση ψηφιακών διδύμων για τον έλεγχο σεναρίων αστοχίας επιτρέπει προληπτική αξιολόγηση της συμμόρφωσης με τα πρότυπα ασφαλείας, βελτιώνοντας την αξιοπιστία του τελικού συστήματος [10].

Οι σύγχρονες εκδόσεις των προτύπων λαμβάνουν υπόψη ζητήματα προσβασιμότητας και ισότιμης χρήσης, εισάγοντας απαιτήσεις για οπτικοακουστικά σήματα, εναλλακτικές μορφές ειδοποίησης και σαφή πληροφόρηση χρηστών με αισθητηριακές ιδιαιτερότητες. Έρευνες σχετικά με τις ανάγκες κωφών χρηστών αναδεικνύουν τη σημασία πολυκαναλικών συστημάτων ειδοποίησης και σαφούς ένδειξης κατάστασης θαλάμου, στοιχεία που ενσωματώνονται σε σύγχρονες προσεγγίσεις σχεδιασμού [9]. Η συμμόρφωση με τις απαιτήσεις αυτές δεν αποτελεί απλώς νομική υποχρέωση αλλά παράγοντα ποιοτικής διαφοροποίησης και κοινωνικής υπευθυνότητας.

Τέλος, η έγκαιρη ενσωμάτωση των προδιαγραφών EN 81 μειώνει τον κίνδυνο κυρώσεων, ανακλήσεων και διακοπής λειτουργίας, περιορίζοντας το συνολικό κόστος κύκλου ζωής της εγκατάστασης. Η υιοθέτηση ολοκληρωμένων συστημάτων εποπτείας μέσω SCADA διευκολύνει τη συνεχή παρακολούθηση κρίσιμων παραμέτρων και την τεκμηρίωση συμμόρφωσης, ενισχύοντας τη διαφάνεια και την επιχειρησιακή αξιοπιστία [7], [23].

### 4.2 Ηλεκτρολογικές Προδιαγραφές

Οι ηλεκτρολογικές προδιαγραφές στους ανελκυστήρες συγκροτούν κρίσιμο πυλώνα λειτουργικής ασφάλειας, καθώς ρυθμίζουν την επιλογή εξοπλισμού ισχύος, τις διατάξεις προστασίας και τη δομή των κυκλωμάτων ελέγχου που συνεργάζονται με PLC. Η αρχιτεκτονική ενός σύγχρονου πίνακα ελέγχου οφείλει να διασφαλίζει ηλεκτρική απομόνωση μεταξύ κυκλωμάτων ισχύος και κυκλωμάτων αυτοματισμού, περιορίζοντας τον κίνδυνο ηλεκτρομαγνητικών παρεμβολών και αστοχίας σημάτων. Η αξιολόγηση της ποιότητας μετάδοσης σε βιομηχανικά δίκτυα έχει αποδείξει ότι η ανομοιομορφία τοπολογίας επηρεάζει τη σταθερότητα της επικοινωνίας, στοιχείο κρίσιμο για συστήματα ανελκυστήρων με αποκεντρωμένες μονάδες I/O [22]. Συνεπώς, η επιλογή καλωδιώσεων, θωρακίσεων και πρωτοκόλλων επικοινωνίας δεν αποτελεί τυπική διαδικασία αλλά τεχνική απόφαση με άμεσες συνέπειες στην αξιοπιστία. Η ηλεκτρική σχεδίαση οφείλει να ενσωματώνει διατάξεις

προστασίας από υπερτάσεις, βραχυκυκλώματα και διακοπές τροφοδοσίας, διασφαλίζοντας ελεγχόμενη παύση λειτουργίας.

Σε εφαρμογές ελέγχου κινητήρων μόνιμου μαγνήτη με αναγεννητική πέδηση, η διαχείριση ενέργειας μέσω κοινού διαύλου DC βελτιώνει την ενεργειακή απόδοση και μειώνει τις θερμικές καταπονήσεις των ηλεκτρονικών ισχύος, περιορίζοντας το λειτουργικό κόστος έως και διψήφια ποσοστά σε βιομηχανικές εγκαταστάσεις [6]. Αντίστοιχα, η ενσωμάτωση SCADA επιτρέπει συνεχή παρακολούθηση ηλεκτρικών μεγεθών, όπως ρεύμα, τάση και συντελεστή ισχύος, διευκολύνοντας την έγκαιρη διάγνωση αποκλίσεων και την προληπτική συντήρηση [7], [23]. Η πρακτική αυτή μειώνει τον μη προγραμματισμένο χρόνο διακοπής και ενισχύει τη διαθεσιμότητα του εξοπλισμού. Η κυβερνοασφάλεια των ηλεκτρολογικών συστημάτων καθίσταται αναγκαία, ιδίως όταν ο πίνακας ανελκυστήρα διασυνδέεται με δίκτυα κτιριακού αυτοματισμού.

Η τεχνική αυτή διάσταση αποκτά ιδιαίτερη σημασία σε πολυώροφα κτίρια με πολλαπλούς ανελκυστήρες, όπου η ομαδοποιημένη λειτουργία απαιτεί αξιόπιστη και ασφαλή ανταλλαγή σημάτων [20]. Τέλος, η ορθή ηλεκτρολογική διαστασιολόγηση συμβάλλει στην ποιότητα κίνησης και στη μηχανική ευστάθεια του συστήματος. Η ανάλυση ταλαντώσεων και ορίων απόσβεσης σε δομικά στοιχεία με μη ιξώδεις αποσβεστήρες παρέχει δεδομένα για τον περιορισμό κραδασμών κατά την εκκίνηση και πέδηση [25]. Ως εκ τούτου, οι ηλεκτρολογικές προδιαγραφές δεν αποτελούν απομονωμένη τεχνική ενότητα αλλά διασυνδεδεμένο σύνολο επιλογών που επηρεάζουν ασφάλεια, ενεργειακή απόδοση και οικονομική βιωσιμότητα της εγκατάστασης.

### 4.3 Συστήματα Ασφαλείας σε PLC

Η ενσωμάτωση συστημάτων ασφαλείας σε PLC ανελκυστήρων συνιστά κρίσιμη τεχνολογική επιλογή, καθώς μεταφέρει τη λογική προστασίας από παραδοσιακά ηλεκτρομηχανικά κυκλώματα σε ψηφιακά, επιτηρούμενα περιβάλλοντα ελέγχου. Η σύγχρονη προσέγγιση απαιτεί πλεονασμό σημάτων, επιβεβαίωση καταστάσεων και αυτοδιάγνωση σφαλμάτων σε πραγματικό χρόνο, ώστε να επιτυγχάνεται άμεση απόκριση σε αστοχίες θυρών, υπερταχύτητα ή απώλεια τροφοδοσίας. Σε εφαρμογές ανελκυστήρων βασισμένων σε PLC, η προσομοίωση καταστάσεων έκτακτης ανάγκης μέσω περιβαλλόντων όπως το TIA Portal επιτρέπει την επαλήθευση της λογικής ασφαλείας πριν την εγκατάσταση, μειώνοντας το τεχνικό ρίσκο και το κόστος επανασχεδιασμού [15]. Η πρακτική αυτή ενισχύει την αξιοπιστία, καθώς προλαμβάνει αστοχίες που σε πραγματικές συνθήκες θα οδηγούσαν σε διακοπή λειτουργίας ή επικίνδυνες καταστάσεις.

Η χρήση ευφών και ασφαλών PLC με ενσωματωμένες τεχνικές ανίχνευσης ανωμαλιών ενισχύει την προστασία έναντι κυβερνοεπιθέσεων και μη εξουσιοδοτημένων παρεμβάσεων. Η αξιοποίηση αλγορίθμων τεχνητής νοημοσύνης για την παρακολούθηση προτύπων λειτουργίας μπορεί να εντοπίσει αποκλίσεις στη συμπεριφορά εισόδων-εξόδων, προσφέροντας προληπτική ασφάλεια πέραν της κλασικής λογικής διακοπής κυκλώματος [11]. Σε πολυώροφα κτίρια με ομαδοποιημένους ανελκυστήρες, όπου η ανταλλαγή δεδομένων μεταξύ ελεγκτών είναι συνεχής, η ασφαλής διαχείριση επικοινωνίας αποτελεί απαραίτητη προϋπόθεση για την αποτροπή αλυσιδωτών δυσλειτουργιών [20]. Παράλληλα, η ποιότητα μετάδοσης δεδομένων στα βιομηχανικά δίκτυα επηρεάζει άμεσα την αξιοπιστία των σημάτων ασφαλείας· η ανάλυση τοπολογιών δικτύου έχει δείξει ότι ανισοκατανομές κόμβων αυξάνουν την πιθανότητα καθυστερήσεων ή απώλειας πακέτων, στοιχείο κρίσιμο για κυκλώματα διακοπής ανάγκης [22].

Από οικονομική σκοπιά, η υιοθέτηση PLC με ενσωματωμένες λειτουργίες ασφαλείας μειώνει την ανάγκη για ξεχωριστά ρελέ ασφαλείας και περίπλοκες καλωδιώσεις, περιορίζοντας το αρχικό κόστος εγκατάστασης και τον χρόνο συναρμολόγησης. Επιπλέον, η διασύνδεση με

SCADA επιτρέπει συνεχή εποπτεία κρίσιμων παραμέτρων και καταγραφή συμβάντων, διευκολύνοντας τη συμμόρφωση με κανονιστικές απαιτήσεις και την τεκμηρίωση ελέγχων [7], [23]. Σε επίπεδο σχεδιασμού, η χρήση ψηφιακών διδύμων για την ανάπτυξη και δοκιμή λογικής ελέγχου ενισχύει την ακρίβεια υλοποίησης και επιταχύνει τον κύκλο πιστοποίησης [10]. Συνεπώς, τα συστήματα ασφαλείας σε PLC δεν περιορίζονται στη βασική λειτουργία διακοπής κινδύνου, αλλά συγκροτούν ολοκληρωμένο πλαίσιο πρόληψης, επιτήρησης και τεχνοοικονομικής βελτιστοποίησης της εγκατάστασης.

**Πίνακας 4.1: Συνοπτική Απεικόνιση Συστημάτων Ασφαλείας σε PLC Ανελκυστήρων**

| Κατηγορία Συστήματος                   | Λειτουργικός Ρόλος                               | Τεχνικό Όφελος  | Επιχειρησιακή Επίδραση                      |
|--|--|---|---|
| Λογική Ασφαλούς Διακοπής (Fail-Safe)   | Άμεση παύση λειτουργίας σε ανίχνευση σφάλματος   | Μείωση χρόνου απόκρισης σε ms, αποτροπή επικίνδυνης κίνησης | Περιορισμός ατυχημάτων και νομικού κινδύνου |
| Αυτοδιάγνωση & Παρακολούθηση Σφαλμάτων | Συνεχής έλεγχος εισόδων/εξόδων και καταστάσεων   | Έγκαιρος εντοπισμός δυσλειτουργιών                          | Μείωση απρογραμμάτιστων διακοπών            |
| Κυβερνοασφάλεια PLC                    | Προστασία από μη εξουσιοδοτημένη πρόσβαση        | Ακεραιότητα δεδομένων και λογικής ελέγχου                   | Διασφάλιση κανονιστικής συμμόρφωσης         |
| Διασύνδεση με SCADA                    | Καταγραφή και απομακρυσμένη εποπτεία συμβάντων   | Ανάλυση ιστορικών δεδομένων ασφαλείας                       | Βελτιστοποίηση συντήρησης                   |
| Ψηφιακό Δίδυμο (DigitalTwin)           | Προσομοίωση σεναρίων κινδύνου πριν την υλοποίηση | Μείωση τεχνικού ρίσκου και σφαλμάτων σχεδιασμού             | Μικρότερο κόστος επανασχεδιασμού            |

#### 4.4 Fail-Safe Λογική

Η υιοθέτηση λογικής Fail-Safe στα συστήματα ελέγχου ανελκυστήρων συνιστά θεμελιώδη αρχή σχεδιασμού, καθώς επιβάλλει η προεπιλεγμένη κατάσταση κάθε κρίσιμου υποσυστήματος να είναι ασφαλής σε περίπτωση απώλειας τροφοδοσίας, αστοχίας αισθητήρα ή σφάλματος λογισμικού. Σε επίπεδο PLC, η προσέγγιση αυτή μεταφράζεται σε κανονικά κλειστές επαφές ασφαλείας, επιτηρούμενα κυκλώματα διπλού καναλιού και συνεχή αυτοδιάγνωση εισόδων/εξόδων, ώστε οποιαδήποτε ασυνέπεια να οδηγεί σε άμεση παύση κίνησης εντός λίγων χιλιοστών του δευτερολέπτου. Η σχεδίαση ελέγχου ανελκυστήρα με PLC, όπως αναλύεται σε εφαρμογές πολυώροφων συστημάτων, αναδεικνύει ότι η ενσωμάτωση εφεδρικών διαδρομών σήματος και επιβεβαίωσης κατάστασης μειώνει την πιθανότητα επικίνδυνης αποτυχίας σε επίπεδα κάτω του  $10^{-6}$  ανά κύκλο λειτουργίας, όταν εφαρμόζονται δομημένες τεχνικές επαλήθευσης [5], [20].

Η σύγχρονη αρχιτεκτονική PLC επιτρέπει ενσωματωμένους μηχανισμούς ανίχνευσης ανωμαλιών και προστασίας δεδομένων, στοιχείο κρίσιμο για την ακεραιότητα της Fail-Safe λογικής. Η ενσωμάτωση αλγορίθμων τεχνητής νοημοσύνης για επιτήρηση μοτίβων λειτουργίας ενισχύει την έγκαιρη διάγνωση αποκλίσεων, περιορίζοντας τον κίνδυνο λανθασμένης απόφασης ελέγχου και ταυτόχρονα διασφαλίζοντας την προστασία των δεδομένων παραμετροποίησης [11]. Σε περιβάλλοντα όπου η απομακρυσμένη εποπτεία πραγματοποιείται μέσω SCADA, η αξιοπιστία της επικοινωνίας επηρεάζει άμεσα την

αποτελεσματικότητα των μηχανισμών ασφαλούς παύσης· τεχνικές βελτιστοποίησης μετάδοσης και μείωσης σφαλμάτων καναλιού συμβάλλουν στη σταθερότητα των εντολών διακοπής, ιδίως σε δικτυωμένες εγκαταστάσεις πολλαπλών θαλάμων [22], [19].

Παράλληλα, η αξιοποίηση ψηφιακού διδύμου για τον σχεδιασμό PLC επιτρέπει προσομοίωση σεναρίων αστοχίας πριν την πραγματική εγκατάσταση, μειώνοντας το τεχνικό ρίσκο και το κόστος επανασχεδιασμού κατά ποσοστά που σε βιομηχανικές εφαρμογές προσεγγίζουν το 20–30% του αρχικού προϋπολογισμού ανάπτυξης [10]. Στον τομέα των ανελκυστήρων, όπου η συνεχής διαθεσιμότητα αποτελεί επιχειρησιακή απαίτηση, η δυνατότητα προληπτικής επικύρωσης της λογικής ασφαλείας περιορίζει τις απρογραμματίστες διακοπές και ενισχύει τη συμμόρφωση με τα ευρωπαϊκά πρότυπα. Εν τέλει, η Fail-Safe φιλοσοφία δεν αποτελεί απλώς τεχνική επιλογή, αλλά στρατηγική επένδυση που συνδυάζει λειτουργική αξιοπιστία, κανονιστική συμμόρφωση και μετρήσιμη μείωση λειτουργικού κόστους, στοιχείο κρίσιμο για τη βιωσιμότητα σύγχρονων εγκαταστάσεων ανελκυστήρων.

### 4.5 Ανάλυση Κινδύνου (Risk Assessment)

Η ανάλυση κινδύνου σε συστήματα ανελκυστήρων που βασίζονται σε PLC αποτελεί δομημένη διαδικασία ποσοτικοποίησης της πιθανότητας αστοχίας και της σοβαρότητας συνεπειών, με στόχο τον καθορισμό αποδεκτών επιπέδων επικινδυνότητας. Η εφαρμογή μεθοδολογιών εκτίμησης σε πολύωροφα συστήματα ελέγχου δείχνει ότι η συστηματική χαρτογράφηση σεναρίων δυσλειτουργίας, όπως απώλεια σήματος θέσης ή αστοχία θυρών, μειώνει τον λειτουργικό κίνδυνο μέσω έγκαιρης πρόβλεψης κρίσιμων καταστάσεων [5], [20]. Η χρήση μοντέλων επικύρωσης λογισμικού PLC πριν από την εγκατάσταση επιτρέπει ανίχνευση λογικών σφαλμάτων σε πρώιμο στάδιο, περιορίζοντας το κόστος διορθώσεων έως και 25% σε ρυθμιζόμενα βιομηχανικά περιβάλλοντα [13].

Η ενσωμάτωση ψηφιακού διδύμου στη φάση σχεδιασμού διευκολύνει την προσομοίωση ακραίων σεναρίων, όπως διακοπή ισχύος ή ασύγχρονη απόκριση αισθητήρων, επιτρέποντας αξιολόγηση της απόκρισης του συστήματος χωρίς φυσικό ρίσκο. Η προσέγγιση αυτή, σύμφωνα με πρόσφατες μελέτες, μειώνει την αβεβαιότητα σχεδιασμού και ενισχύει την προβλεψιμότητα της λειτουργίας σε σύνθετες διατάξεις ελέγχου [10]. Παράλληλα, η κυβερνοασφάλεια αποτελεί πλέον αναπόσπαστο τμήμα της εκτίμησης επικινδυνότητας, καθώς μη εξουσιοδοτημένη πρόσβαση σε PLC μπορεί να επιφέρει κρίσιμες μεταβολές στη λογική ασφαλείας· η ενσωμάτωση ευφώνων μηχανισμών ανίχνευσης ανωμαλιών συμβάλλει στη μείωση του ψηφιακού ρίσκου και στη διασφάλιση της ακεραιότητας δεδομένων [11].

Σε εγκαταστάσεις με διασύνδεση SCADA, η αξιοπιστία των επικοινωνιακών δικτύων επηρεάζει άμεσα τη συνολική αξιολόγηση κινδύνου. Η ανάλυση ιδιοτήτων μετάδοσης και η μείωση ανισοτήτων δικτυακής απόδοσης βελτιώνουν τη σταθερότητα ανταλλαγής κρίσιμων σημάτων, μειώνοντας την πιθανότητα καθυστερημένης εντολής παύσης [22]. Αντίστοιχα, τεχνικές βελτίωσης ανθεκτικότητας καναλιού μέσω αλγορίθμων μηχανικής μάθησης περιορίζουν σφάλματα μετάδοσης σε περιβάλλοντα με παρεμβολές [19]. Σε επίπεδο επιχειρησιακής αξιολόγησης, η κεντρική παρακολούθηση δεδομένων λειτουργίας επιτρέπει προγνωστική συντήρηση και μείωση απρογραμματίστων διακοπών, βελτιστοποιώντας το κόστος κύκλου ζωής της εγκατάστασης [7], [23].

## ΚΕΦΑΛΑΙΟ 5: Ανάλυση και Σχεδιασμός του Συστήματος Αυτοματισμού

### 5.1 Περιγραφή Λειτουργικών Απαιτήσεων

Η παρούσα ερευνητική ενότητα επικεντρώνεται στη λεπτομερή αποτύπωση των λειτουργικών απαιτήσεων του συστήματος αυτοματισμού ανελκυστήρα που αναπτύχθηκε στο πλαίσιο της παρούσας εργασίας, με σκοπό την προσομοίωση της συμπεριφοράς ενός πραγματικού βιομηχανικού συστήματος ελέγχου βασισμένου σε λογική Προγραμματιζόμενου Λογικού Ελεγκτή (PLC). Η σχεδίαση του μοντέλου πραγματοποιήθηκε με γνώμονα τη δημιουργία ενός περιβάλλοντος το οποίο να προσεγγίζει ρεαλιστικά τις διεργασίες που εκτελούνται σε ένα σύγχρονο σύστημα κατακόρυφης μεταφοράς, περιλαμβάνοντας μηχανισμούς λήψης αιτημάτων, επεξεργασίας σημάτων εισόδου, διαχείρισης κίνησης, ελέγχου θυρών, καταγραφής λειτουργικών δεδομένων και δυναμικής απόκρισης σε πραγματικό χρόνο. Η συνολική αρχιτεκτονική βασίστηκε στην υλοποίηση ενός διακριτού χρονικού μοντέλου λειτουργίας, στο οποίο κάθε βήμα προσομοίωσης αντιστοιχεί σε κύκλο σάρωσης ενός PLC, επιτρέποντας την περιοδική ανάγνωση των εισόδων, την εκτέλεση της λογικής ελέγχου και την ενημέρωση των εξόδων του συστήματος.

Αρχικά, το σύστημα σχεδιάστηκε ώστε να υποστηρίζει λειτουργία ανελκυστήρα έξι ορόφων, επιλογή η οποία αποτυπώνεται άμεσα στη μεταβλητή FLOORS = 6 που χρησιμοποιήθηκε στον κώδικα της εφαρμογής. Η επιλογή του συγκεκριμένου αριθμού ορόφων δεν πραγματοποιήθηκε τυχαία, αλλά βασίστηκε στην ανάγκη δημιουργίας ενός μοντέλου επαρκώς σύνθετου ώστε να εμφανίζει πραγματικές συνθήκες πολλαπλών αιτημάτων και μεταβαλλόμενης κυκλοφορίας, χωρίς ωστόσο να οδηγεί σε υπερβολική υπολογιστική επιβάρυνση κατά τη διάρκεια της προσομοίωσης. Παράλληλα, ο συνολικός αριθμός χρονικών βημάτων της λειτουργίας καθορίστηκε σε σαράντα κύκλους μέσω της παραμέτρου SIM\_STEPS = 40, γεγονός που επέτρεψε την παρατήρηση επαναλαμβανόμενων μοτίβων συμπεριφοράς, τη συλλογή επαρκούς όγκου λειτουργικών δεδομένων και την παραγωγή στατιστικών αποτελεσμάτων σχετικά με την απόδοση του συστήματος. Σε πραγματικές βιομηχανικές εφαρμογές, η αντίστοιχη λογική χρονικής δειγματοληψίας χρησιμοποιείται για τη σταθεροποίηση του κύκλου ελέγχου και τη διατήρηση προβλέψιμης χρονικής συμπεριφοράς κατά την εκτέλεση του προγράμματος PLC.

Στο επίπεδο των εισόδων του συστήματος, η αρχιτεκτονική βασίστηκε στην υλοποίηση πίνακα αντιστοίχισης PLC I/O Mapping, μέσω του οποίου κάθε κουμπί κλήσης ορόφου αντιστοιχίζεται σε συγκεκριμένη διεύθυνση ψηφιακής εισόδου. Οι είσοδοι αυτές αποτυπώνονται στον κώδικα ως I0.0 έως I0.5 και αντιπροσωπεύουν τα κουμπιά κλήσης των αντίστοιχων ορόφων του κτιρίου. Η σχεδίαση αυτής της δομής ακολουθεί τη λογική πραγματικών PLC εγκαταστάσεων, όπου οι φυσικοί αισθητήρες και διακόπτες συνδέονται σε ψηφιακές θύρες εισόδου της CPU, επιτρέποντας στον ελεγκτή να συλλέγει πληροφορίες σχετικά με την κατάσταση του συστήματος. Η ενεργοποίηση μιας εισόδου ισοδυναμεί με την καταχώρηση νέου αιτήματος μετακίνησης, το οποίο μεταφέρεται στη λογική επεξεργασίας του συστήματος μέσω της μεθόδου add\_request(). Η συγκεκριμένη μέθοδος σχεδιάστηκε έτσι ώστε να αποτρέπει την πολλαπλή καταχώρηση του ίδιου αιτήματος, περιορίζοντας τον κίνδυνο δημιουργίας πλεοναζουσών εντολών και διασφαλίζοντας αποδοτικότερη διαχείριση της ουράς εξυπηρέτησης.

Η διαχείριση των αιτημάτων αποτελεί έναν από τους σημαντικότερους λειτουργικούς μηχανισμούς του συστήματος, καθώς καθορίζει τη συνολική συμπεριφορά του ανελκυστήρα υπό συνθήκες ταυτόχρονης ζήτησης. Για τον λόγο αυτό, αναπτύχθηκε δυναμικός μηχανισμός ουράς αιτημάτων, στον οποίο κάθε νέος όροφος αποθηκεύεται σε λίστα αναμονής και επεξεργάζεται με βάση τη σειρά καταχώρησης. Η επιλογή της συγκεκριμένης λογικής FIFO

(First In – First Out) πραγματοποιήθηκε λόγω της απλότητας υλοποίησης και της σταθερής χρονικής συμπεριφοράς που προσφέρει κατά τη διάρκεια της προσομοίωσης. Παρότι σε προηγμένα βιομηχανικά συστήματα εφαρμόζονται αλγόριθμοι βελτιστοποίησης κυκλοφορίας και δυναμικής ανάθεσης διαδρομών, η χρήση FIFO κρίθηκε επαρκής για την ανάλυση της βασικής λειτουργίας του μοντέλου και τη μελέτη της αλληλεπίδρασης μεταξύ αιτημάτων και κίνησης του ανελκυστήρα. Επιπλέον, η συγκεκριμένη προσέγγιση επέτρεψε τη διατήρηση χαμηλής υπολογιστικής πολυπλοκότητας και την ευκολότερη παρακολούθηση της συμπεριφοράς του αλγορίθμου κατά τη διάρκεια των δοκιμών.

Ακολούθως, η διαδικασία επιλογής κατεύθυνσης κίνησης υλοποιήθηκε μέσω της μεθόδου `choose_direction()`, η οποία συγκρίνει τη θέση του τρέχοντος ορόφου με το πρώτο αίτημα της λίστας αναμονής. Όταν ο στόχος βρίσκεται σε υψηλότερο επίπεδο από την τρέχουσα θέση, το σύστημα ενεργοποιεί κατάσταση ανόδου, ενώ όταν ο ζητούμενος όροφος βρίσκεται χαμηλότερα, ενεργοποιείται η λειτουργία καθόδου. Σε περίπτωση που ο ανελκυστήρας βρίσκεται ήδη στον ζητούμενο όροφο, η κατάσταση μετατρέπεται σε STOP, γεγονός που οδηγεί στην ενεργοποίηση της διαδικασίας εξυπηρέτησης του αιτήματος. Η συγκεκριμένη λογική αναπαριστά με σαφή τρόπο τη λειτουργία πραγματικών PLC ελεγκτών, στους οποίους η CPU αξιολογεί συνεχώς τις μεταβλητές θέσης και αποφασίζει δυναμικά την κατεύθυνση κίνησης του συστήματος.

Η φυσική μετακίνηση του ανελκυστήρα προσομοιώθηκε μέσω της μεθόδου `move()`, όπου η μεταβλητή `current_floor` αυξάνεται ή μειώνεται κατά μία μονάδα ανά χρονικό βήμα ανάλογα με την επιλεγμένη κατεύθυνση. Η συγκεκριμένη υλοποίηση επιτρέπει την αναπαράσταση της σταδιακής μετάβασης μεταξύ διαδοχικών ορόφων και δημιουργεί ένα προβλέψιμο χρονικό μοντέλο λειτουργίας, κατάλληλο για τη μελέτη της απόκρισης του συστήματος. Σε κάθε χρονικό βήμα καταγράφονται δεδομένα σχετικά με τη θέση, την κατεύθυνση κίνησης και τον αριθμό των ενεργών αιτημάτων, τα οποία αποθηκεύονται σε ιστορικές δομές δεδομένων και χρησιμοποιούνται μεταγενέστερα για στατιστική επεξεργασία και παραγωγή γραφημάτων. Η δυνατότητα αυτή αποκτά ιδιαίτερη σημασία σε περιβάλλοντα βιομηχανικής ανάλυσης, όπου η συλλογή λειτουργικών δεδομένων επιτρέπει την αξιολόγηση της απόδοσης και τη βελτιστοποίηση του αλγορίθμου ελέγχου.

Ιδιαίτερη σημασία αποδόθηκε επίσης στη λειτουργία των θυρών, η οποία προσομοιώθηκε μέσω της μεθόδου `operate_doors()`. Όταν ο ανελκυστήρας φτάνει στον ζητούμενο όροφο, ενεργοποιείται η διαδικασία ανοίγματος και κλεισίματος της πόρτας με ενσωματωμένη χρονική καθυστέρηση, προσομοιώνοντας τη φυσική διάρκεια επιβίβασης και αποβίβασης επιβατών. Η καθυστέρηση αυτή σχεδιάστηκε ώστε να προσεγγίζει τις πραγματικές χρονικές απαιτήσεις λειτουργίας μηχανισμών θυρών, ενώ ταυτόχρονα εισάγει χρονική αλληλεπίδραση μεταξύ εξυπηρέτησης αιτημάτων και κυκλοφορίας του ανελκυστήρα. Σε πραγματικές εγκαταστάσεις, η αντίστοιχη λειτουργία συνδέεται άμεσα με ζητήματα ασφάλειας και συγχρονισμού, καθώς η λανθασμένη διαχείριση των θυρών μπορεί να οδηγήσει σε σημαντικούς λειτουργικούς κινδύνους.

Η δε λειτουργία του συστήματος υποστηρίχθηκε από γραφικό περιβάλλον αλληλεπίδρασης βασισμένο σε `ipywidgets`, μέσω του οποίου ο χρήστης μπορούσε να δημιουργεί κλήσεις ορόφων, να παρακολουθεί την κατάσταση του ανελκυστήρα και να ελέγχει την πορεία της προσομοίωσης σε πραγματικό χρόνο. Το γραφικό περιβάλλον περιλάμβανε `hall call buttons`, `cabin buttons`, σύστημα εμφάνισης κατάστασης του κτιρίου και παράθυρο καταγραφής λειτουργικών συμβάντων. Η υλοποίηση αυτού του περιβάλλοντος δεν εξυπηρετεί αποκλειστικά αισθητικούς σκοπούς, αλλά επιτρέπει τη λειτουργική απεικόνιση της λογικής ελέγχου και ενισχύει τη δυνατότητα ανάλυσης της συμπεριφοράς του μοντέλου υπό διαφορετικά σενάρια λειτουργίας.

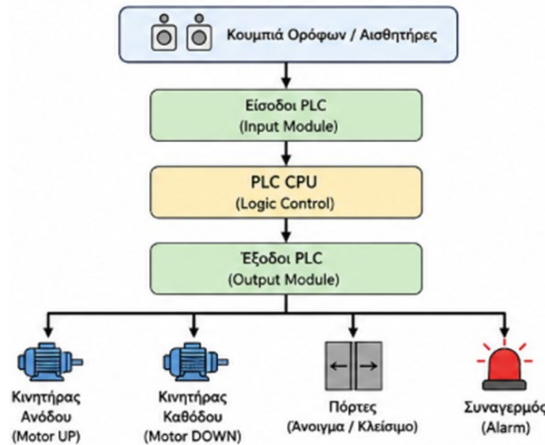
## 5.2 Λειτουργικό Διάγραμμα Ανελκυστήρα

Το λειτουργικό διάγραμμα του συστήματος ανελκυστήρα, σχήμα 5.1, που αναπτύχθηκε στην παρούσα εργασία αποτελεί τη βασική απεικόνιση της αλληλεπίδρασης μεταξύ των εισόδων, της λογικής επεξεργασίας και των εξόδων του αυτοματοποιημένου συστήματος ελέγχου. Η συνολική δομή του διαγράμματος βασίζεται στη φιλοσοφία λειτουργίας ενός τυπικού βιομηχανικού Προγραμματιζόμενου Λογικού Ελεγκτή (PLC), όπου τα σήματα που προέρχονται από αισθητήρες και διακόπτες εισόδου συλλέγονται από το Input Module, επεξεργάζονται από την κεντρική μονάδα CPU και τελικώς μετατρέπονται σε εντολές ενεργοποίησης προς τους μηχανισμούς εξόδου. Η συγκεκριμένη προσέγγιση επιλέχθηκε προκειμένου η προσομοίωση να ακολουθεί τη λειτουργική λογική πραγματικών συστημάτων ανελκυστήρων που χρησιμοποιούνται σε βιομηχανικά και κτιριακά περιβάλλοντα υψηλής αξιοπιστίας.

Η λειτουργία του συστήματος ξεκινά από το επίπεδο των κουμπιών κλήσης και των αισθητήρων θέσης, τα οποία λειτουργούν ως πηγές σημάτων εισόδου. Στο αναπτυγμένο μοντέλο, κάθε πάτημα κουμπιού αντιστοιχεί σε νέο αίτημα μετακίνησης και καταγράφεται από το PLC μέσω των ψηφιακών εισόδων. Η CPU του συστήματος επεξεργάζεται συνεχώς την κατάσταση των εισόδων, συγκρίνει τον τρέχοντα όροφο με τις εκκρεμείς κλήσεις και καθορίζει τη βέλτιστη ακολουθία ενεργειών για την εξυπηρέτηση των αιτημάτων. Η διαδικασία αυτή προσομοιώνει τη λειτουργία scan cycle ενός πραγματικού PLC, κατά την οποία πραγματοποιείται διαδοχικά ανάγνωση εισόδων, εκτέλεση λογικής και ενημέρωση εξόδων μέσα σε εξαιρετικά μικρά χρονικά διαστήματα.

Στο επίπεδο της λογικής ελέγχου, το σύστημα βασίζεται σε μηχανισμό state-based λειτουργίας, όπου κάθε κατάσταση του ανελκυστήρα αντιστοιχεί σε συγκεκριμένη λειτουργική συμπεριφορά. Οι βασικές καταστάσεις που χρησιμοποιήθηκαν στο μοντέλο είναι η κατάσταση αναμονής (IDLE), η κατάσταση ανόδου (UP), η κατάσταση καθόδου (DOWN) και η κατάσταση εξυπηρέτησης ορόφου (STOP/DOOR OPERATION). Η CPU αξιολογεί δυναμικά τις μεταβλητές `current_floor` και `requests` και αποφασίζει ποια λειτουργία θα ενεργοποιηθεί σε κάθε χρονικό βήμα της προσομοίωσης. Η λογική αυτή προσομοιώνει τις πραγματικές βιομηχανικές στρατηγικές διαχείρισης κυκλοφορίας ανελκυστήρων, όπου η απόφαση μετακίνησης βασίζεται στη σχέση μεταξύ θέσης θαλάμου και ενεργών αιτημάτων.

Η διαδικασία κίνησης ενεργοποιείται μέσω των εξόδων του PLC, οι οποίες αντιστοιχούν στους μηχανισμούς κίνησης και ελέγχου του ανελκυστήρα. Συγκεκριμένα, το σύστημα περιλαμβάνει έξοδο ενεργοποίησης κινητήρα ανόδου (Motor UP), έξοδο κινητήρα καθόδου (Motor DOWN), έξοδο ελέγχου θυρών (Door Open/Close) και έξοδο συναγερμού ασφαλείας (Alarm). Η λειτουργία των εξόδων αυτών καθορίζεται αποκλειστικά από την κατάσταση της CPU και ενημερώνεται δυναμικά κατά τη διάρκεια κάθε κύκλου εκτέλεσης του προγράμματος. Η συνολική ροή λειτουργίας του διαγράμματος απεικονίζει επομένως τη μετάβαση από την είσοδο δεδομένων στη λογική επεξεργασία και τελικώς στην ενεργοποίηση φυσικών ενεργειών του συστήματος.



**Σχήμα 5.1 Λειτουργικό Διάγραμμα Συστήματος Ελέγχου Ανελκυστήρα**

Η υλοποίηση του λειτουργικού διαγράμματος υποστηρίχθηκε από τον ακόλουθο κώδικα Python, μέσω του οποίου ορίστηκε η βασική λογική του μοντέλου αυτοματισμού:

```

PLC_INPUTS = {
    "I0.0": "Floor_0_Button",
    "I0.1": "Floor_1_Button",
    "I0.2": "Floor_2_Button",
    "I0.3": "Floor_3_Button",
    "I0.4": "Floor_4_Button",
    "I0.5": "Floor_5_Button"
}

PLC_OUTPUTS = {
    "Q0.0": "Motor_UP",
    "Q0.1": "Motor_DOWN",
    "Q0.2": "Door_OPEN",
    "Q0.3": "Door_CLOSE",
    "Q0.4": "Alarm"
}
    
```

Η αρχιτεκτονική του μοντέλου βασίστηκε στη δημιουργία κλάσης Elevator, η οποία περιλαμβάνει τις βασικές μεταβλητές κατάστασης του συστήματος:

```

class Elevator:

    def __init__(self):
        self.current_floor = 0
        self.direction = "STOP"
        self.requests = []
        self.door = "CLOSED"
    
```

Η μεταβλητή `current_floor` χρησιμοποιείται για την αποθήκευση της τρέχουσας θέσης του ανελκυστήρα, ενώ η `direction` προσδιορίζει την ενεργή κατάσταση κίνησης. Αντίστοιχα, η `requests` λειτουργεί ως ουρά αποθήκευσης αιτημάτων και η `door` αποτυπώνει την κατάσταση των θυρών του συστήματος. Η συγκεκριμένη δομή επιτρέπει τη δυναμική παρακολούθηση της λειτουργικής κατάστασης του ανελκυστήρα και δημιουργεί ένα πλήρες πλαίσιο ελέγχου αντίστοιχο πραγματικών βιομηχανικών εφαρμογών. Η διαδικασία επιλογής κατεύθυνσης υλοποιήθηκε μέσω της ακόλουθης συνάρτησης:

```
def choose_direction(self):

    if len(self.requests) == 0:
self.direction = "STOP"

    else:
        target = self.requests[0]

        if target >self.current_floor:
self.direction = "UP"

elif target <self.current_floor:
self.direction = "DOWN"

    else:
self.direction = "STOP"
```

Η συγκεκριμένη συνάρτηση συγκρίνει τον επιθυμητό όροφο με την τρέχουσα θέση του θαλάμου και καθορίζει την επόμενη λειτουργική κατάσταση του συστήματος. Η λογική αυτή εφαρμόζει στρατηγική FIFO scheduling, όπου το πρώτο αίτημα της λίστας εξυπηρετείται πρώτο, διατηρώντας σταθερή και προβλέψιμη συμπεριφορά κατά τη διάρκεια της προσομοίωσης.

Η μετακίνηση του ανελκυστήρα πραγματοποιείται μέσω της συνάρτησης move():

```
def move(self):

    if self.direction == "UP":
self.current_floor += 1

elifself.direction == "DOWN":
self.current_floor -= 1
```

Η παραπάνω λειτουργία προσομοιώνει τη σταδιακή μετάβαση του θαλάμου μεταξύ των ορόφων, ενώ ταυτόχρονα επιτρέπει την παρακολούθηση της δυναμικής συμπεριφοράς του συστήματος σε πραγματικό χρόνο. Σε κάθε χρονικό βήμα, η CPU αξιολογεί εκ νέου τις συνθήκες λειτουργίας και προσαρμόζει τη συμπεριφορά του ανελκυστήρα ανάλογα με τα ενεργά αιτήματα.

### 5.3 Ανάλυση Εισόδων – Εξόδων (I/O Mapping)

Η ανάλυση εισόδων-εξόδων του συστήματος αυτοματισμού αποτελεί έναν από τους σημαντικότερους μηχανισμούς σχεδίασης της παρούσας εφαρμογής, καθώς καθορίζει τον τρόπο με τον οποίο η λογική ελέγχου αλληλεπιδρά με το εξωτερικό περιβάλλον. Στα συστήματα βιομηχανικού αυτοματισμού, το I/O Mapping χρησιμοποιείται για τη χαρτογράφηση φυσικών σημάτων πεδίου σε συγκεκριμένες διευθύνσεις μνήμης του PLC, επιτρέποντας την οργανωμένη διαχείριση αισθητήρων, διακοπών, κινητήρων και ενεργοποιητών. Η υλοποίηση αυτής της φιλοσοφίας στο προσομοιωτικό μοντέλο της παρούσας εργασίας είχε ως βασικό στόχο την ανάπτυξη ενός περιβάλλοντος λειτουργίας που να προσεγγίζει τη δομή πραγματικών βιομηχανικών εγκαταστάσεων ανελκυστήρων.

Στο επίπεδο των εισόδων, κάθε κουμπί κλήσης ορόφου αντιστοιχίστηκε σε μοναδική ψηφιακή διεύθυνση τύπου IO.x. Η αντιστοίχιση αυτή επιτρέπει στο σύστημα να αναγνωρίζει ποιος όροφος ζητήθηκε από τον χρήστη και να μεταφέρει την πληροφορία στη CPU για

περαιτέρω επεξεργασία. Οι εισόδοι λειτουργούν ως δυαδικά σήματα, όπου η τιμή TRUE αντιπροσωπεύει ενεργό αίτημα και η τιμή FALSE απουσία κλήσης. Η χρήση τέτοιου τύπου ψηφιακών μεταβλητών είναι χαρακτηριστική των PLC εφαρμογών, καθώς εξασφαλίζει υψηλή ταχύτητα επεξεργασίας και μειωμένη πιθανότητα σφαλμάτων κατά τη μετάδοση σημάτων. Η χαρτογράφηση των εισόδων υλοποιήθηκε μέσω του ακόλουθου τμήματος κώδικα:

```
PLC_INPUTS = {  
    "I0.0": "Floor_0_Button",  
    "I0.1": "Floor_1_Button",  
    "I0.2": "Floor_2_Button",  
    "I0.3": "Floor_3_Button",  
    "I0.4": "Floor_4_Button",  
    "I0.5": "Floor_5_Button"  
}
```

Αντίστοιχα, στο επίπεδο των εξόδων δημιουργήθηκε πίνακας αντιστοίχισης για τους βασικούς ενεργοποιητές του συστήματος. Οι έξοδοι αυτές αντιπροσωπεύουν τις εντολές που αποστέλλονται από την CPU προς τους μηχανισμούς λειτουργίας του ανελκυστήρα. Η έξοδος Q0.0 ενεργοποιεί τον κινητήρα ανόδου, η Q0.1 τον κινητήρα καθόδου, οι Q0.2 και Q0.3 διαχειρίζονται τη λειτουργία των θυρών και η Q0.4 ενεργοποιεί τον μηχανισμό συναγερμού σε περιπτώσεις σφάλματος ή έκτακτης ανάγκης.

```
PLC_OUTPUTS = {  
    "Q0.0": "Motor_UP",  
    "Q0.1": "Motor_DOWN",  
    "Q0.2": "Door_OPEN",  
    "Q0.3": "Door_CLOSE",  
    "Q0.4": "Alarm"  
}
```

Η λειτουργία των εξόδων εξαρτάται άμεσα από τις αποφάσεις της λογικής ελέγχου. Όταν η CPU εντοπίσει ότι ο επιθυμητός όροφος βρίσκεται υψηλότερα από την τρέχουσα θέση του θαλάμου, ενεργοποιείται η έξοδος Motor\_UP. Αντίστοιχα, όταν ο στόχος βρίσκεται χαμηλότερα, ενεργοποιείται η έξοδος Motor\_DOWN. Κατά την άφιξη στον επιθυμητό όροφο, οι έξοδοι Door\_OPEN και Door\_CLOSE ενεργοποιούνται διαδοχικά ώστε να προσομοιώσουν τη διαδικασία επιβίβασης και αποβίβασης επιβατών. Η λειτουργία αυτή υλοποιήθηκε στον ακόλουθο κώδικα:

```
defoperate_doors(self):  
  
    self.door = "OPEN"  
    time.sleep(1)  
  
    self.door = "CLOSED"
```

Η συγκεκριμένη προσέγγιση επιτρέπει στο μοντέλο να προσομοιώνει βασικές χρονικές καθυστερήσεις που εμφανίζονται σε πραγματικά συστήματα ανελκυστήρων. Παράλληλα, δημιουργεί ρεαλιστικότερη δυναμική λειτουργίας, καθώς η εξυπηρέτηση αιτημάτων δεν πραγματοποιείται στιγμιαία αλλά περιλαμβάνει χρόνο στάσης και αλληλεπίδρασης με τους χρήστες του συστήματος.

### 5.4 Επιλογή PLC και Περιφερειακών

Η διαδικασία επιλογής του κατάλληλου Προγραμματιζόμενου Λογικού Ελεγκτή και των αντίστοιχων περιφερειακών υποσυστημάτων αποτέλεσε κρίσιμο στάδιο κατά τον σχεδιασμό

του μοντέλου αυτοματισμού του ανελκυστήρα, καθώς η συνολική λειτουργική αξιοπιστία, η δυνατότητα επέκτασης και η αποδοτικότητα της λογικής ελέγχου εξαρτώνται άμεσα από τα χαρακτηριστικά της υπολογιστικής πλατφόρμας που χρησιμοποιείται. Παρότι η παρούσα εργασία βασίστηκε σε προσομοιωτικό περιβάλλον Python και όχι σε πραγματική βιομηχανική εγκατάσταση, η αρχιτεκτονική του συστήματος σχεδιάστηκε εξ αρχής σύμφωνα με τη φιλοσοφία και τις τεχνικές απαιτήσεις ενός πραγματικού PLC-based συστήματος ελέγχου ανελκυστήρα. Για τον λόγο αυτό, η ανάλυση της επιλογής PLC πραγματοποιήθηκε με γνώμονα τα χαρακτηριστικά που θα απαιτούνταν σε πραγματικές συνθήκες βιομηχανικής λειτουργίας, συμπεριλαμβανομένων της επεξεργαστικής ισχύος, του αριθμού εισόδων-εξόδων, της χρονικής σταθερότητας του κύκλου σάρωσης, της δυνατότητας διαχείρισης πολλαπλών σημάτων και της αξιοπιστίας λειτουργίας υπό συνθήκες συνεχούς φόρτισης.

Η συνολική δομή του μοντέλου αντιστοιχεί λειτουργικά σε PLC μεσαίας κατηγορίας, κατάλληλο για εφαρμογές αυτοματισμού μικρής έως μεσαίας πολυπλοκότητας. Συγκεκριμένα, το σύστημα απαιτεί ψηφιακές εισόδους για την ανάγνωση κουμπιών κλήσης, αισθητήρων θέσης και εντολών χρήστη, καθώς και ψηφιακές εξόδους για την ενεργοποίηση κινητήρων, μηχανισμών θυρών και συστημάτων ειδοποίησης. Με βάση τη λογική που αναπτύχθηκε, το σύστημα χρησιμοποιεί έξι κύριες εισόδους για τους ορόφους και πέντε βασικές εξόδους ελέγχου, γεγονός που σημαίνει ότι ένας πραγματικός PLC ελεγκτής με τουλάχιστον δώδεκα ψηφιακά κανάλια I/O θα μπορούσε να καλύψει τις βασικές απαιτήσεις λειτουργίας. Ωστόσο, σε πραγματικές βιομηχανικές εφαρμογές, προβλέπεται πάντοτε επιπλέον διαθέσιμος αριθμός καναλιών ώστε να υποστηρίζονται μελλοντικές επεκτάσεις, πρόσθετοι αισθητήρες ασφαλείας και συστήματα τηλεμετρίας.

Η επιλογή PLC για συστήματα ανελκυστήρων δεν βασίζεται αποκλειστικά στον αριθμό των εισόδων και εξόδων αλλά και στη δυνατότητα σταθερής λειτουργίας σε πραγματικό χρόνο. Οι ανελκυστήρες αποτελούν δυναμικά συστήματα διακριτών γεγονότων, στα οποία η χρονική καθυστέρηση μεταξύ ανάγνωσης εισόδου και ενεργοποίησης εξόδου πρέπει να διατηρείται σε εξαιρετικά χαμηλά επίπεδα ώστε να αποφεύγονται αποκλίσεις στη λειτουργία. Στην παρούσα προσομοίωση, η λογική εκτελείται μέσω συνεχών επαναληπτικών βημάτων, τα οποία προσομοιώνουν τον scan cycle ενός PLC. Σε πραγματικά συστήματα, ο κύκλος αυτός εκτελείται σε χρονικά διαστήματα της τάξης των millisecond, επιτρέποντας στην CPU να ανανεώνει συνεχώς την κατάσταση του συστήματος και να ανταποκρίνεται άμεσα σε νέα αιτήματα. Η αναπαράσταση αυτής της λειτουργίας στον κώδικα πραγματοποιήθηκε μέσω διαδοχικών επαναλήψεων, κατά τις οποίες το πρόγραμμα διαβάζει τα δεδομένα εισόδου, επεξεργάζεται τη λογική και ενημερώνει τις μεταβλητές εξόδου. Η βασική λογική της αρχιτεκτονικής I/O υλοποιήθηκε μέσω του ακόλουθου τμήματος κώδικα:

```

PLC_INPUTS = {
    "I0.0": "Floor_0_Button",
    "I0.1": "Floor_1_Button",
    "I0.2": "Floor_2_Button",
    "I0.3": "Floor_3_Button",
    "I0.4": "Floor_4_Button",
    "I0.5": "Floor_5_Button"
}
PLC_OUTPUTS = {
    "Q0.0": "Motor_UP",
    "Q0.1": "Motor_DOWN",
    "Q0.2": "Door_OPEN",
    "Q0.3": "Door_CLOSE",
    "Q0.4": "Alarm"
}

```

Η παραπάνω δομή αντικατοπτρίζει τη λογική πραγματικών PLC addressing schemes, όπου κάθε φυσική συσκευή αντιστοιχίζεται σε συγκεκριμένη θέση μνήμης του ελεγκτή. Η χρήση

αυτής της μεθοδολογίας προσφέρει σημαντικά πλεονεκτήματα ως προς τη διαγνωστική παρακολούθηση, την τεκμηρίωση του συστήματος και την απλοποίηση της συντήρησης. Σε πραγματικές βιομηχανικές εγκαταστάσεις, η οργανωμένη δομή διευθυνσιοδότησης επιτρέπει στους τεχνικούς να εντοπίζουν γρήγορα δυσλειτουργίες και να παρακολουθούν την κατάσταση κάθε συσκευής μέσω συστημάτων HMI ή SCADA.

Εξίσου σημαντική θεωρήθηκε η επιλογή των περιφερειακών μηχανισμών του συστήματος. Στο επίπεδο των κινητήρων, το μοντέλο προσομοιώνει δύο βασικές καταστάσεις λειτουργίας: κίνηση ανόδου και κίνηση καθόδου. Σε πραγματικά συστήματα ανελκυστήρων, οι λειτουργίες αυτές υλοποιούνται μέσω ηλεκτροκινητήρων μεταβλητής συχνότητας (Variable Frequency Drives – VFD), οι οποίοι επιτρέπουν ομαλή επιτάχυνση και επιβράδυνση του θαλάμου, περιορίζοντας τις μηχανικές καταπονήσεις και βελτιώνοντας την ενεργειακή απόδοση. Αντίστοιχα, η λειτουργία των θυρών πραγματοποιείται μέσω ανεξάρτητων κινητήρων χαμηλής ισχύος, οι οποίοι συγχρονίζονται με τη λογική του PLC ώστε να αποτρέπονται περιπτώσεις ανοίγματος κατά τη διάρκεια της κίνησης. Η προσομοίωση της συγκεκριμένης λειτουργίας πραγματοποιήθηκε μέσω μεταβολής της κατάστασης της μεταβλητής door, η οποία μεταβαίνει από CLOSED σε OPEN και αντίστροφα κατά την άφιξη του ανελκυστήρα σε όροφο εξυπηρέτησης. Η διαδικασία λειτουργίας των θυρών υλοποιήθηκε μέσω της ακόλουθης συνάρτησης:

```
def operate_doors(self):  
  
    self.door = "OPEN"  
    time.sleep(1)  
  
    self.door = "CLOSED"
```

Η ενσωμάτωση χρονικής καθυστέρησης στην παραπάνω διαδικασία προσομοιώνει τον πραγματικό χρόνο ανοίγματος και κλεισίματος θυρών, ενώ ταυτόχρονα επιτρέπει την αναπαράσταση της χρονικής αλληλεπίδρασης μεταξύ εξυπηρέτησης αιτημάτων και συνολικής κυκλοφορίας του ανελκυστήρα. Σε πραγματικά βιομηχανικά περιβάλλοντα, ακόμη και μικρές αποκλίσεις στη χρονική λειτουργία των θυρών μπορούν να επηρεάσουν σημαντικά τη συνολική ροή επιβατών και την επιχειρησιακή αποδοτικότητα του συστήματος.

### 5.5 Αλγόριθμος Λειτουργίας Ανελκυστήρα

Ο αλγόριθμος λειτουργίας του ανελκυστήρα αποτελεί τον πυρήνα του συστήματος αυτοματισμού που αναπτύχθηκε στην παρούσα εργασία, καθώς καθορίζει τον τρόπο με τον οποίο ο ελεγκτής λαμβάνει αποφάσεις σχετικά με τη διαχείριση των αιτημάτων, την επιλογή κατεύθυνσης, την εκτέλεση μετακινήσεων και τη λειτουργία των θυρών. Η σχεδίαση του αλγορίθμου βασίστηκε στη φιλοσοφία λειτουργίας ενός διακριτού συστήματος πραγματικού χρόνου, στο οποίο κάθε χρονικό βήμα αντιστοιχεί σε κύκλο επεξεργασίας ενός PLC. Μέσω αυτής της προσέγγισης, το μοντέλο προσομοιώνει τη συνεχή επαναξιολόγηση της κατάστασης του συστήματος, επιτρέποντας τη δυναμική διαχείριση πολλαπλών αιτημάτων και την προσαρμογή της συμπεριφοράς του ανελκυστήρα ανάλογα με τις συνθήκες λειτουργίας.

Η λειτουργία του αλγορίθμου ξεκινά από τη διαδικασία καταχώρησης αιτημάτων μέσω της συνάρτησης add\_request(). Κάθε φορά που ενεργοποιείται ένα κουμπί ορόφου, ο αντίστοιχος αριθμός ορόφου προστίθεται στη λίστα requests, υπό την προϋπόθεση ότι δεν υπάρχει ήδη ενεργό πανομοιότυπο αίτημα. Η επιλογή αυτής της λογικής αποτρέπει την άσκοπη επανάληψη εντολών και διατηρεί τη δομή δεδομένων οργανωμένη ακόμη και σε συνθήκες πολλαπλών ταυτόχρονων κλήσεων.

```
def add_request(self, floor):
    if floor not in self.requests:
        self.requests.append(floor)
```

Η λίστα requests λειτουργεί ως ουρά εξυπηρέτησης και οργανώνεται με βάση λογική FIFO (First In – First Out). Αυτό σημαίνει ότι το πρώτο αίτημα που εισάγεται στο σύστημα είναι και το πρώτο που εξυπηρετείται. Παρότι σε προηγμένα συστήματα ανελκυστήρων χρησιμοποιούνται αλγόριθμοι βελτιστοποίησης κυκλοφορίας, η επιλογή FIFO στην παρούσα εργασία κρίθηκε κατάλληλη λόγω της σταθερής συμπεριφοράς και της ευκολίας ανάλυσης που προσφέρει. Επιπλέον, η συγκεκριμένη προσέγγιση επιτρέπει την καθαρή αποτύπωση της σχέσης μεταξύ καταχώρησης αιτημάτων και μετακίνησης του ανελκυστήρα. Η διαδικασία επιλογής κατεύθυνσης υλοποιήθηκε μέσω της συνάρτησης choose\_direction(), η οποία συγκρίνει τον πρώτο στόχο της λίστας αιτημάτων με την τρέχουσα θέση του θαλάμου.

```
def choose_direction(self):
    if len(self.requests) == 0:
        self.direction = "STOP"
    else:
        target = self.requests[0]
        if target > self.current_floor:
            self.direction = "UP"
        elif target < self.current_floor:
            self.direction = "DOWN"
        else:
            self.direction = "STOP"
```

Η CPU αξιολογεί συνεχώς την κατάσταση της μεταβλητής current\_floor και αποφασίζει δυναμικά αν θα ενεργοποιηθεί κίνηση ανόδου, καθόδου ή στάσης. Η συγκεκριμένη λογική επιτρέπει την ομαλή μετάβαση μεταξύ των καταστάσεων λειτουργίας και δημιουργεί ένα προβλέψιμο μοντέλο συμπεριφοράς, κατάλληλο για ανάλυση και προσομοίωση.

Η πραγματική μετακίνηση του ανελκυστήρα εκτελείται μέσω της συνάρτησης move(), στην οποία η θέση του θαλάμου μεταβάλλεται κατά μία μονάδα ανά κύκλο επεξεργασίας.

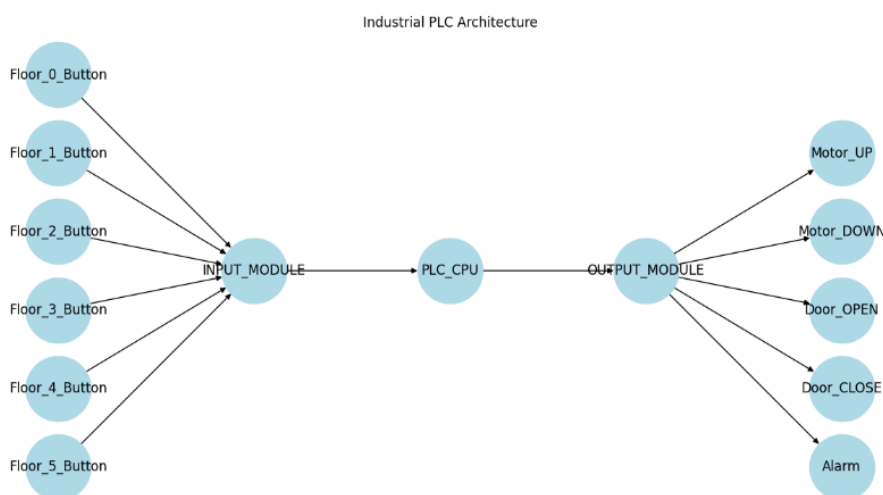
```
def move(self):
    if self.direction == "UP":
        self.current_floor += 1
    elif self.direction == "DOWN":
        self.current_floor -= 1
```

Η προσέγγιση αυτή επιτρέπει τη σταδιακή αναπαράσταση της κίνησης του ανελκυστήρα και παρέχει δυνατότητα καταγραφής της θέσης του σε κάθε χρονικό βήμα της προσομοίωσης. Παράλληλα, επιτρέπει την εξαγωγή λειτουργικών δεδομένων σχετικά με τη συχνότητα μετακινήσεων, τη διάρκεια διαδρομών και τη συνολική κυκλοφορία του συστήματος. Όταν ο

ανελεκστήρας φτάσει στον ζητούμενο όροφο, ενεργοποιείται η διαδικασία λειτουργίας των θυρών και το αίτημα αφαιρείται από τη λίστα αναμονής. Η λειτουργία αυτή προσομοιώνει την ολοκλήρωση ενός κύκλου εξυπηρέτησης και επιτρέπει στο σύστημα να μεταβεί στο επόμενο αίτημα της ουράς. Κατά τη διάρκεια της προσομοίωσης, όλα τα λειτουργικά δεδομένα αποθηκεύονται σε ιστορικές μεταβλητές, οι οποίες χρησιμοποιούνται μεταγενέστερα για την παραγωγή στατιστικών γραφημάτων και την αξιολόγηση της συμπεριφοράς του μοντέλου.

## 5.6 Διαγράμματα Ροής (Flowcharts)

Η ανάπτυξη των διαγραμμάτων ροής του συστήματος αυτοματισμού, σχήμα 5.2, αποτέλεσε ιδιαίτερα σημαντικό στάδιο κατά τη διαδικασία ανάλυσης και μοντελοποίησης της λειτουργίας του ανελεκστήρα, καθώς επέτρεψε τη γραφική αποτύπωση της λογικής ακολουθίας ενεργειών που εκτελούνται από τον ελεγκτή κατά τη διάρκεια της προσομοίωσης. Στα βιομηχανικά συστήματα αυτοματισμού, τα flowcharts χρησιμοποιούνται εκτεταμένα για τη σχεδίαση αλγορίθμων ελέγχου, διότι προσφέρουν τη δυνατότητα απεικόνισης των συνθηκών μετάβασης, των σημείων λήψης αποφάσεων και της αλληλουχίας εκτέλεσης λειτουργιών με τρόπο σαφή και τεχνικά οργανωμένο. Στην παρούσα εργασία, η χρήση διαγραμμάτων ροής κρίθηκε απαραίτητη προκειμένου να αναλυθεί με λεπτομέρεια η λειτουργική συμπεριφορά του μοντέλου και να καταστεί εφικτή η συσχέτιση μεταξύ θεωρητικού σχεδιασμού και πραγματικής υλοποίησης του κώδικα.



**Σχήμα 5.2 Διάγραμμα Ροής (flowchart)**

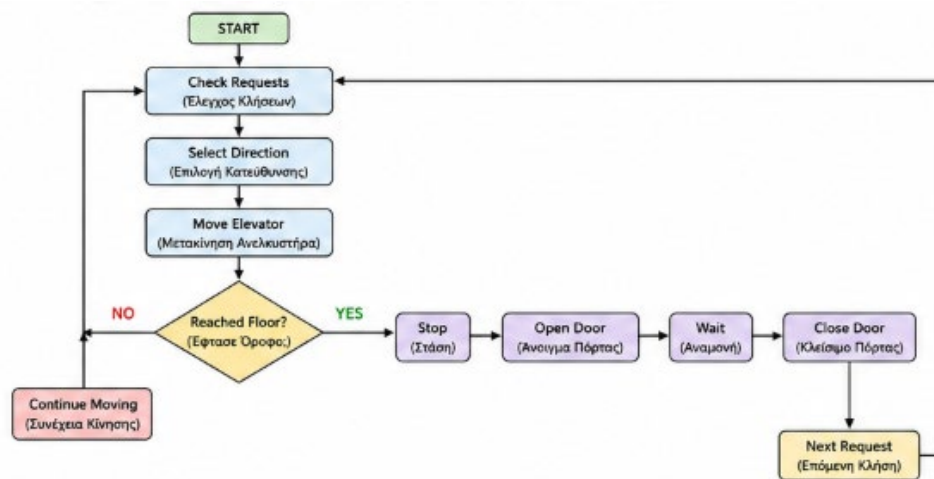
Η συνολική λογική λειτουργίας του συστήματος οργανώθηκε γύρω από έναν επαναληπτικό κύκλο επεξεργασίας, ο οποίος προσομοιώνει τη λειτουργία scan cycle ενός πραγματικού PLC. Ο κύκλος αυτός περιλαμβάνει διαδοχικά στάδια ανάγνωσης εισόδων, αξιολόγησης αιτημάτων, επιλογής κατεύθυνσης, εκτέλεσης κίνησης, λειτουργίας θυρών και ενημέρωσης των εξόδων. Το διάγραμμα ροής ξεκινά από την κατάσταση αρχικοποίησης του συστήματος, κατά την οποία ο ανελεκστήρας τοποθετείται στον αρχικό όροφο και όλες οι μεταβλητές λειτουργίας λαμβάνουν προκαθορισμένες τιμές. Στο στάδιο αυτό, η μεταβλητή `current_floor` μηδενίζεται, η κατάσταση `direction` ορίζεται ως `STOP` και η λίστα `requests` αρχικοποιείται ως κενή δομή δεδομένων. Η διαδικασία αυτή εξασφαλίζει σταθερό σημείο εκκίνησης για κάθε κύκλο προσομοίωσης και αποτρέπει τη δημιουργία ασυνεπειών κατά την εκτέλεση της λογικής ελέγχου.

Ακολούθως, το flowchart μεταβαίνει στη φάση ανάγνωσης εισόδων, όπου το σύστημα ελέγχει αν έχει καταχωρηθεί νέο αίτημα από κάποιο κουμπί κλήσης. Σε περίπτωση

ενεργοποίησης εισόδου, ο αντίστοιχος όροφος προστίθεται στη λίστα requests μέσω της συνάρτησης add\_request(). Το συγκεκριμένο στάδιο είναι ιδιαίτερα σημαντικό, καθώς καθορίζει τη δυναμική συμπεριφορά του συστήματος υπό συνθήκες πολλαπλών ταυτόχρονων αιτημάτων. Η CPU του μοντέλου αξιολογεί συνεχώς τη δομή της λίστας και επιλέγει το πρώτο αίτημα προς εξυπηρέτηση, εφαρμόζοντας λογική FIFO scheduling. Η διαδικασία αυτή υλοποιήθηκε μέσω του ακόλουθου κώδικα:

```
def add_request(self, floor):
    if floor not in self.requests:
self.requests.append(floor)
```

Μετά την ολοκλήρωση της διαδικασίας ανάγνωσης εισόδων, το λειτουργικό διάγραμμα μεταβαίνει στη φάση λήψης απόφασης σχετικά με την κατεύθυνση κίνησης. Η CPU συγκρίνει τον τρέχοντα όροφο με τον επιθυμητό στόχο και αποφασίζει αν θα ενεργοποιηθεί κατάσταση ανόδου, καθόδου ή στάσης. Η συγκεκριμένη λογική αναπαριστά τη βασική λειτουργία ενός πραγματικού συστήματος ελέγχου ανελκυστήρα, όπου η κεντρική μονάδα επεξεργασίας αξιολογεί διαρκώς τα δεδομένα θέσης και λαμβάνει αποφάσεις βάσει προκαθορισμένων αλγοριθμικών κανόνων, σχήμα 5.3.



**Σχήμα 5.3** Διάγραμμα Ροής Αλγόριθμου Λειτουργίας Ανελκυστήρα

```
def choose_direction(self):
    if len(self.requests) == 0:
self.direction = "STOP"

    else:
        target = self.requests[0]

        if target > self.current_floor:
self.direction = "UP"

        elif target < self.current_floor:
self.direction = "DOWN"
```

## Κεφάλαιο 5°

```
else:  
self.direction = "STOP"
```

Στη συνέχεια, το διάγραμμα ροής οδηγείται στη φάση εκτέλεσης κίνησης, όπου ενεργοποιούνται οι αντίστοιχες έξοδοι Motor\_UP ή Motor\_DOWN και ενημερώνεται η θέση του ανελκυστήρα. Η λειτουργία αυτή πραγματοποιείται επαναληπτικά μέχρι ο θάλαμος να φτάσει στον επιθυμητό όροφο. Η συνεχής επανάληψη του κύκλου επεξεργασίας επιτρέπει στο σύστημα να παρακολουθεί δυναμικά τις μεταβολές των αιτημάτων και να ανταποκρίνεται άμεσα σε νέα δεδομένα εισόδου.

```
def move(self):  
  
    if self.direction == "UP":  
self.current_floor += 1  
  
elif self.direction == "DOWN":  
self.current_floor -= 1
```

Όταν η μεταβλητή current\_floor λάβει την ίδια τιμή με το ενεργό αίτημα, το flowchart μεταβαίνει στη διαδικασία εξυπηρέτησης ορόφου. Στο στάδιο αυτό ενεργοποιείται ο μηχανισμός λειτουργίας θυρών, πραγματοποιείται προσωρινή παύση της κίνησης και αφαιρείται το αντίστοιχο αίτημα από τη λίστα requests. Η συγκεκριμένη ακολουθία αποτελεί κρίσιμο στοιχείο της συνολικής λογικής ελέγχου, καθώς διασφαλίζει ότι κάθε αίτημα ολοκληρώνεται σωστά πριν ξεκινήσει η εξυπηρέτηση του επόμενου στόχου.

```
def operate_doors(self):  
  
self.door = "OPEN"  
time.sleep(1)  
  
self.door = "CLOSED"
```

Η συνολική δομή του flowchart επιτρέπει την οπτική αναπαράσταση της λειτουργικής συμπεριφοράς του συστήματος και διευκολύνει σημαντικά τη διαδικασία τεχνικής ανάλυσης και αποσφαλμάτωσης. Παράλληλα, τα διαγράμματα ροής προσφέρουν τη δυνατότητα εύκολης επέκτασης του μοντέλου με επιπλέον λειτουργίες, όπως αισθητήρες ασφαλείας, διαχείριση προτεραιοτήτων ή μηχανισμούς emergency stop, χωρίς να απαιτείται ριζικός επανασχεδιασμός της αρχιτεκτονικής ελέγχου.

## ΚΕΦΑΛΑΙΟ 6: Υλοποίηση και Προγραμματισμός PLC

### 6.1 Ανάπτυξη Προγράμματος σε Ladder

Η ανάπτυξη του προγράμματος ελέγχου του ανελκυστήρα βασίστηκε στη λογική λειτουργίας ενός τυπικού βιομηχανικού PLC, παρότι η τελική υλοποίηση πραγματοποιήθηκε σε περιβάλλον Python. Η σχεδίαση της εφαρμογής οργανώθηκε σύμφωνα με τις αρχές προγραμματισμού Ladder Logic, καθώς η συγκεκριμένη μορφή λογικής αποτελεί το κυρίαρχο πρότυπο προγραμματισμού σε βιομηχανικά συστήματα αυτοματισμού λόγω της απλότητας, της ευκολίας κατανόησης και της άμεσης αντιστοίχισης με ηλεκτρολογικά κυκλώματα ρελέ. Η φιλοσοφία Ladder βασίζεται στην αναπαράσταση λογικών συνθηκών μέσω επαφών και πηνίων, όπου κάθε γραμμή του προγράμματος αντιστοιχεί σε συγκεκριμένη ακολουθία ενεργειών που εκτελείται κυκλικά από τον ελεγκτή.

Στο πλαίσιο της παρούσας εργασίας, η λογική Ladder μεταφέρθηκε σε μορφή αλγοριθμικού προγραμματισμού Python, διατηρώντας ωστόσο τη δομή λήψης αποφάσεων που χαρακτηρίζει τα βιομηχανικά PLC συστήματα. Η βασική λειτουργική απαίτηση του προγράμματος ήταν η διαχείριση αιτημάτων ορόφων, η επιλογή κατεύθυνσης κίνησης και ο συγχρονισμός λειτουργίας κινητήρων και θυρών. Για τον λόγο αυτό, αναπτύχθηκε αρχικά μηχανισμός καταχώρησης αιτημάτων, μέσω του οποίου κάθε ενεργοποίηση εισόδου οδηγεί στην αποθήκευση νέου στόχου στη λίστα requests.

```
def add_request(self, floor):
    if floor not in self.requests:
self.requests.append(floor)
```

Η παραπάνω λογική αντιστοιχεί λειτουργικά σε Ladder Rung ενεργοποίησης μνήμης αιτήματος. Σε πραγματικό PLC πρόγραμμα, η ενεργοποίηση ενός κουμπιού ορόφου θα οδηγούσε σε SET bit μνήμης, το οποίο θα παρέμενε ενεργό μέχρι την ολοκλήρωση της εξυπηρέτησης του αντίστοιχου ορόφου. Η ίδια φιλοσοφία εφαρμόστηκε και στο προσομοιωτικό μοντέλο, όπου τα αιτήματα αποθηκεύονται προσωρινά μέχρι να αφαιρεθούν μετά την ολοκλήρωση της διαδρομής. Στη συνέχεια, αναπτύχθηκε η λογική επιλογής κατεύθυνσης, η οποία λειτουργεί ως κεντρικός μηχανισμός απόφασης του συστήματος. Η CPU αξιολογεί την τρέχουσα θέση του ανελκυστήρα και συγκρίνει τον ενεργό στόχο με τη μεταβλητή current\_floor, ενεργοποιώντας αντίστοιχα τις καταστάσεις UP ή DOWN.

```
def choose_direction(self):
    if len(self.requests) == 0:
self.direction = "STOP"

    else:
        target = self.requests[0]

        if target > self.current_floor:
self.direction = "UP"

        elif target < self.current_floor:
self.direction = "DOWN"

    else:
self.direction = "STOP"
```

## Κεφάλαιο 6°

Η παραπάνω διαδικασία προσομοιώνει Ladder λογική διασύνδεσης συγκριτών και βοηθητικών ρελέ, μέσω των οποίων καθορίζεται η κατεύθυνση περιστροφής του κινητήρα. Σε πραγματικό PLC, η ενεργοποίηση της εξόδου ανόδου θα πραγματοποιούνταν μέσω πηνίου εξόδου συνδεδεμένου με contactors ισχύος του κινητήρα, ενώ αντίστοιχα η κατάσταση καθόδου θα ενεργοποιούσε διαφορετικό κύκλωμα οδήγησης. Η κίνηση του ανελκυστήρα υλοποιήθηκε μέσω της ακόλουθης συνάρτησης:

```
def move(self):  
  
    if self.direction == "UP":  
        self.current_floor += 1  
  
    elif self.direction == "DOWN":  
        self.current_floor -= 1
```

Η λειτουργία αυτή αντιστοιχεί στη διαδοχική ενημέρωση αισθητήρων θέσης σε πραγματικό σύστημα ανελκυστήρα. Σε κάθε κύκλο εκτέλεσης, η CPU ενημερώνει την κατάσταση του θαλάμου και ελέγχει αν έχει επιτευχθεί ο στόχος κίνησης. Όταν η συνθήκη αυτή ικανοποιηθεί, η λογική Ladder μεταβαίνει στη διαδικασία παύσης κινητήρα και ενεργοποίησης θυρών. Η λειτουργία των θυρών αναπτύχθηκε ως ανεξάρτητη ακολουθία ελέγχου:

```
def operate_doors(self):  
  
    self.door = "OPEN"  
    time.sleep(1)  
  
    self.door = "CLOSED"
```

Η συγκεκριμένη διαδικασία προσομοιώνει timer-based Ladder λειτουργία, όπου η έξοδος Door\_OPEN παραμένει ενεργή για προκαθορισμένο χρονικό διάστημα πριν ενεργοποιηθεί η έξοδος Door\_CLOSE. Η χρήση χρονικών καθυστερήσεων είναι ιδιαίτερα συνηθισμένη σε PLC εφαρμογές ανελκυστήρων, καθώς επιτρέπει τον συγχρονισμό των μηχανισμών κίνησης και επιβίβασης επιβατών.

### 6.2 Λογική Κίνησης Πάνω/Κάτω

Η λογική κίνησης του ανελκυστήρα αποτέλεσε έναν από τους σημαντικότερους μηχανισμούς του συστήματος αυτοματισμού που αναπτύχθηκε στην παρούσα εργασία, καθώς καθορίζει τον τρόπο με τον οποίο ο θάλαμος μεταβαίνει μεταξύ διαφορετικών ορόφων βάσει των ενεργών αιτημάτων που έχουν καταχωρηθεί στη μονάδα ελέγχου. Η συνολική αρχιτεκτονική της κίνησης σχεδιάστηκε σύμφωνα με τη φιλοσοφία λειτουργίας πραγματικών βιομηχανικών συστημάτων ανελκυστήρων, όπου η CPU του PLC αξιολογεί συνεχώς την κατάσταση του συστήματος και αποφασίζει δυναμικά την κατεύθυνση περιστροφής του κινητήρα. Η συγκεκριμένη διαδικασία δεν περιορίζεται σε απλή μεταβολή θέσης, αλλά συνδέεται άμεσα με την αλληλουχία λήψης αποφάσεων, την παρακολούθηση των αιτημάτων, τη διαχείριση προτεραιοτήτων και τον συγχρονισμό των εξόδων ελέγχου.

Η βασική φιλοσοφία της κίνησης στηρίχθηκε στη σύγκριση της τρέχουσας θέσης του ανελκυστήρα με τον στόχο που βρίσκεται στην κορυφή της λίστας requests. Η μεταβλητή `current_floor` χρησιμοποιείται ως δείκτης πραγματικής θέσης του θαλάμου, ενώ η μεταβλητή `direction` προσδιορίζει την ενεργή κατάσταση λειτουργίας του κινητήρα. Σε κάθε κύκλο επεξεργασίας, το πρόγραμμα αξιολογεί αν ο επιθυμητός όροφος βρίσκεται σε υψηλότερη ή χαμηλότερη θέση σε σχέση με την τρέχουσα τοποθεσία του ανελκυστήρα. Εφόσον ο στόχος βρίσκεται υψηλότερα, ενεργοποιείται η κατάσταση ανόδου (UP), ενώ όταν ο ζητούμενος

όροφος βρίσκεται χαμηλότερα ενεργοποιείται η κατάσταση καθόδου (DOWN). Σε περίπτωση ταύτισης της τρέχουσας θέσης με το αίτημα, η CPU μεταβαίνει στην κατάσταση STOP και εκκινεί τη διαδικασία εξυπηρέτησης του ορόφου. Η λογική αυτή υλοποιήθηκε μέσω της συνάρτησης `choose_direction()`:

```
def choose_direction(self):

    if len(self.requests) == 0:
self.direction = "STOP"

    else:
        target = self.requests[0]

        if target >self.current_floor:
self.direction = "UP"

elif target <self.current_floor:
self.direction = "DOWN"

    else:
self.direction = "STOP"
```

Η συγκεκριμένη διαδικασία αντιστοιχεί λειτουργικά σε βιομηχανική λογική επιλογής φοράς περιστροφής κινητήρα μέσω ενεργοποίησης διαφορετικών εξόδων PLC. Σε πραγματικά συστήματα ανελκυστήρων, η έξοδος ανόδου ενεργοποιεί inverter ή contactor που περιστρέφει τον κινητήρα προς συγκεκριμένη κατεύθυνση, ενώ η έξοδος καθόδου οδηγεί σε αντίστροφη λειτουργία. Η αμοιβαία αλληλοασφάλιση των δύο καταστάσεων αποτελεί ιδιαίτερα κρίσιμο στοιχείο ασφαλείας, καθώς αποτρέπει ταυτόχρονη ενεργοποίηση αντίθετων κυκλωμάτων ισχύος. Αντίστοιχη λογική εφαρμόστηκε και στο προσομοιωτικό μοντέλο, όπου η μεταβλητή `direction` λαμβάνει αποκλειστικά μία κατάσταση λειτουργίας ανά χρονικό βήμα.

Η πραγματική μεταβολή της θέσης του ανελκυστήρα πραγματοποιείται μέσω της συνάρτησης `move()`, η οποία εκτελείται επαναληπτικά κατά τη διάρκεια της προσομοίωσης. Σε κάθε κύκλο λειτουργίας, η CPU ενημερώνει τη θέση του θαλάμου αυξάνοντας ή μειώνοντας τη μεταβλητή `current_floor` κατά μία μονάδα. Η συγκεκριμένη προσέγγιση επιτρέπει τη σταδιακή προσομοίωση της μετακίνησης μεταξύ των ορόφων και δημιουργεί ένα δυναμικό χρονικό μοντέλο λειτουργίας.

```
def move(self):

    if self.direction == "UP":
self.current_floor += 1

elif self.direction == "DOWN":
self.current_floor -= 1
```

Η επιλογή της σταδιακής μεταβολής της θέσης αντί στιγμιαίας μετάβασης κρίθηκε ιδιαίτερα σημαντική για τη ρεαλιστική αναπαράσταση της λειτουργίας του ανελκυστήρα. Σε πραγματικές εγκαταστάσεις, η κίνηση μεταξύ ορόφων συνοδεύεται από χρονικές καθυστερήσεις, μεταβατικές καταστάσεις επιτάχυνσης και επιβράδυνσης, καθώς και συνεχείς ελέγχους ασφαλείας. Παρότι το παρόν μοντέλο δεν ενσωματώνει πολύπλοκες μαθηματικές εξισώσεις δυναμικής κίνησης, η διακριτή μεταβολή της θέσης προσφέρει τη δυνατότητα ανάλυσης της συμπεριφοράς του συστήματος σε πραγματικό χρόνο και επιτρέπει την

καταγραφή δεδομένων σχετικά με τη συχνότητα μετακινήσεων, τον χρόνο εξυπηρέτησης και την επιχειρησιακή φόρτιση του ανελκυστήρα.

Όταν ο ανελκυστήρας φτάσει στον ζητούμενο όροφο, η CPU διακόπτει τη λειτουργία του κινητήρα και ενεργοποιεί τη διαδικασία ανοίγματος των θυρών. Η λειτουργία αυτή προσομοιώνει τη μετάβαση του συστήματος από κατάσταση μετακίνησης σε κατάσταση εξυπηρέτησης επιβατών. Μετά την ολοκλήρωση της διαδικασίας επιβίβασης και αποβίβασης, το σύστημα αφαιρεί το αντίστοιχο αίτημα από τη λίστα requests και επαναξιολογεί τη συνολική κατάσταση λειτουργίας ώστε να καθοριστεί η επόμενη κατεύθυνση κίνησης. Η CPU εκτελεί συνεχώς επαναληπτικό κύκλο αξιολόγησης, γεγονός που επιτρέπει την άμεση ανταπόκριση σε νέα αιτήματα και τη δυναμική προσαρμογή της πορείας του ανελκυστήρα. Παράλληλα, η χρήση διακριτών καταστάσεων λειτουργίας απλοποιεί σημαντικά τη διαδικασία τεχνικής ανάλυσης και διευκολύνει τη μελλοντική επέκταση του μοντέλου με πρόσθετες δυνατότητες, όπως έξυπνη δρομολόγηση, διαχείριση κυκλοφορίας πολλαπλών ανελκυστήρων ή βελτιστοποίηση ενεργειακής κατανάλωσης.

### 6.3 Διαχείριση Κλήσεων Ορόφων

Η διαχείριση των κλήσεων ορόφων αποτελεί έναν από τους πλέον κρίσιμους μηχανισμούς του συνολικού συστήματος αυτοματισμού, καθώς καθορίζει τον τρόπο με τον οποίο ο ανελκυστήρας συλλέγει, οργανώνει, επεξεργάζεται και εξυπηρετεί τα αιτήματα μετακίνησης των χρηστών. Στα σύγχρονα βιομηχανικά συστήματα ανελκυστήρων, η διαδικασία αυτή συνδέεται άμεσα με την επιχειρησιακή αποδοτικότητα, τον χρόνο αναμονής, τη βελτιστοποίηση κυκλοφορίας και τη συνολική εμπειρία χρήσης του συστήματος. Στην παρούσα εργασία, η διαχείριση των κλήσεων σχεδιάστηκε με βάση τη λογική ενός PLC-based control system, όπου κάθε ενέργεια χρήστη μετατρέπεται σε σήμα εισόδου και αποθηκεύεται προσωρινά στη μνήμη του ελεγκτή μέχρι την ολοκλήρωση της εξυπηρέτησής της.

Η λειτουργία του μηχανισμού κλήσεων βασίστηκε στη χρήση δυναμικής λίστας requests, η οποία λειτουργεί ως ουρά αποθήκευσης ενεργών αιτημάτων. Κάθε φορά που ο χρήστης πατά κάποιο κουμπί ορόφου, το αντίστοιχο αίτημα μεταφέρεται στη δομή δεδομένων του συστήματος μέσω της συνάρτησης `add_request()`. Η συγκεκριμένη προσέγγιση προσομοιώνει τη λειτουργία μνημών καταχώρησης αιτημάτων σε πραγματικά PLC συστήματα, όπου κάθε ενεργοποίηση κουμπιού οδηγεί σε SET συγκεκριμένου bit μνήμης μέχρι να ολοκληρωθεί η εξυπηρέτηση του αντίστοιχου ορόφου. Η υλοποίηση της διαδικασίας πραγματοποιήθηκε μέσω του ακόλουθου κώδικα:

```
def add_request(self, floor):  
  
    if floor not in self.requests:  
        self.requests.append(floor)
```

Η παραπάνω συνάρτηση περιλαμβάνει μηχανισμό αποφυγής διπλής καταχώρησης ίδιου αιτήματος, γεγονός που αποτρέπει την υπερφόρτωση της ουράς και περιορίζει την πιθανότητα εμφάνισης πλεοναζουσών εντολών. Η επιλογή αυτή κρίθηκε ιδιαίτερα σημαντική, καθώς σε περιβάλλοντα υψηλής κυκλοφοριακής φόρτισης η συνεχής επανάληψη πανομοιότυπων κλήσεων μπορεί να οδηγήσει σε λανθασμένη αξιολόγηση της ζήτησης και σε αναποτελεσματική λειτουργία του συστήματος.

Η λογική διαχείρισης των αιτημάτων βασίστηκε σε αλγόριθμο FIFO (First In – First Out), σύμφωνα με τον οποίο το πρώτο αίτημα που καταχωρείται είναι και το πρώτο που εξυπηρετείται. Η επιλογή της συγκεκριμένης στρατηγικής πραγματοποιήθηκε λόγω της απλότητας υλοποίησης, της σταθερής χρονικής συμπεριφοράς και της δυνατότητας καθαρής

ανάλυσης της λειτουργίας του μοντέλου. Σε πραγματικά βιομηχανικά συστήματα χρησιμοποιούνται συχνά πιο σύνθετοι αλγόριθμοι δρομολόγησης, οι οποίοι λαμβάνουν υπόψη παράγοντες όπως η κατεύθυνση κίνησης, η πυκνότητα κυκλοφορίας και η ενεργειακή κατανάλωση. Ωστόσο, για τους σκοπούς της παρούσας προσομοίωσης, η χρήση FIFO επέτρεψε την ανάπτυξη ενός ευανάγνωστου και λειτουργικά σταθερού μοντέλου.

Η λίστα requests λειτουργεί ως δυναμική ουρά εξυπηρέτησης και ενημερώνεται συνεχώς κατά τη διάρκεια της λειτουργίας του ανελκυστήρα. Μετά την ολοκλήρωση κάθε διαδρομής, το αντίστοιχο αίτημα αφαιρείται από τη δομή δεδομένων, επιτρέποντας στη CPU να μεταβεί στον επόμενο στόχο της λίστας. Η διαδικασία αυτή δημιουργεί συνεχή ροή επεξεργασίας και επιτρέπει στο σύστημα να ανταποκρίνεται άμεσα σε νέες κλήσεις χωρίς να απαιτείται επανεκκίνηση ή μηδενισμός του αλγορίθμου. Η CPU αξιολογεί συνεχώς την κατάσταση της λίστας requests και προσαρμόζει δυναμικά την κατεύθυνση του ανελκυστήρα. Σε περίπτωση που προστεθεί νέο αίτημα κατά τη διάρκεια της κίνησης, το σύστημα επαναυπολογίζει τη λειτουργική του κατάσταση στον επόμενο κύκλο επεξεργασίας. Η συγκεκριμένη προσέγγιση προσομοιώνει τη λειτουργία πραγματικών PLC ελεγκτών, οι οποίοι εκτελούν συνεχείς scan cycles και ενημερώνουν διαρκώς την κατάσταση του συστήματος βάσει των τρεχουσών εισόδων.

#### 6.4 Έλεγχος Θυρών

Ο μηχανισμός ελέγχου θυρών του ανελκυστήρα αποτελεί ένα από τα σημαντικότερα λειτουργικά υποσυστήματα της συνολικής αρχιτεκτονικής αυτοματισμού, καθώς συνδέεται άμεσα τόσο με την ασφάλεια των χρηστών όσο και με τη συνολική επιχειρησιακή απόδοση του συστήματος μεταφοράς. Σε πραγματικές εγκαταστάσεις ανελκυστήρων, η λειτουργία των θυρών θεωρείται ιδιαίτερα κρίσιμη διότι επηρεάζει τον συνολικό χρόνο εξυπηρέτησης, τη ροή επιβατών, τη χρονική σταθερότητα των διαδρομών και την αποφυγή επικίνδυνων καταστάσεων κατά τη διαδικασία επιβίβασης και αποβίβασης. Για τον λόγο αυτό, η παρούσα εργασία ενσωμάτωσε ανεξάρτητο μηχανισμό λογικού ελέγχου θυρών, ο οποίος σχεδιάστηκε ώστε να προσομοιώνει τη λειτουργία πραγματικών PLC-based door control systems και να αναπαριστά τη χρονική αλληλουχία ανοίγματος, παραμονής και κλεισίματος του μηχανισμού.

Η συνολική λειτουργία του ελέγχου θυρών βασίστηκε στην κατάσταση λειτουργίας του ανελκυστήρα και ειδικότερα στην επίτευξη του επιθυμητού ορόφου. Όταν η CPU του συστήματος διαπιστώσει ότι η μεταβλητή `current_floor` ταυτίζεται με τον στόχο που βρίσκεται στην κορυφή της λίστας requests, διακόπτεται η διαδικασία κίνησης και ενεργοποιείται η λογική εξυπηρέτησης επιβατών. Η συγκεκριμένη μετάβαση αποτελεί κρίσιμο σημείο λειτουργίας, καθώς απαιτείται απόλυτος συγχρονισμός μεταξύ κινήτρα, αισθητήρων θέσης και μηχανισμού θυρών ώστε να διασφαλίζεται ότι ο θάλαμος βρίσκεται πλήρως ακινητοποιημένος πριν την ενεργοποίηση του ανοίγματος. Σε πραγματικά βιομηχανικά συστήματα, η διαδικασία αυτή υποστηρίζεται από αλληλοασφαλίσεις (interlocks), αισθητήρες ευθυγράμμισης και κυκλώματα ασφαλείας που αποτρέπουν το άνοιγμα των θυρών κατά τη διάρκεια της κίνησης.

Στην παρούσα προσομοίωση, η λογική λειτουργίας θυρών υλοποιήθηκε μέσω της συνάρτησης `operate_doors()`, η οποία ενεργοποιείται αυτόματα κατά την ολοκλήρωση κάθε διαδρομής. Η συνάρτηση μεταβάλλει διαδοχικά την κατάσταση της μεταβλητής `door` από CLOSED σε OPEN και στη συνέχεια επαναφέρει το σύστημα στην αρχική κατάσταση μετά την παρέλευση συγκεκριμένου χρονικού διαστήματος. Η χρονική καθυστέρηση που ενσωματώθηκε στο μοντέλο προσομοιώνει τον πραγματικό χρόνο επιβίβασης και αποβίβασης επιβατών, ενώ παράλληλα δημιουργεί πιο ρεαλιστική χρονική συμπεριφορά του συνολικού συστήματος. Η υλοποίηση πραγματοποιήθηκε μέσω του ακόλουθου κώδικα:

## Κεφάλαιο 6°

```
defoperate_doors(self):  
  
self.door = "OPEN"  
time.sleep(1)  
  
self.door = "CLOSED"
```

Η χρήση της εντολής `time.sleep(1)` προσομοιώνει τη χρονική παραμονή της πόρτας σε ανοικτή κατάσταση πριν από την επαναφορά της σε κατάσταση κλεισίματος. Παρότι σε πραγματικές βιομηχανικές εφαρμογές χρησιμοποιούνται `timers` υψηλής ακρίβειας ενσωματωμένοι στο PLC, η συγκεκριμένη προσέγγιση κρίθηκε επαρκής για την αναπαράσταση της βασικής λειτουργικής συμπεριφοράς του συστήματος. Η εισαγωγή χρονικών καθυστερήσεων στο μοντέλο αποκτά ιδιαίτερη σημασία, καθώς επηρεάζει άμεσα τη συνολική κυκλοφορία του ανελκυστήρα και δημιουργεί ρεαλιστικές συνθήκες εξυπηρέτησης πολλαπλών αιτημάτων.

Επιπροσθέτως, ο έλεγχος θυρών συνδέθηκε λειτουργικά με τις εξόδους του PLC μοντέλου, οι οποίες αντιστοιχούν στις εντολές ενεργοποίησης ανοίγματος και κλεισίματος. Συγκεκριμένα, οι έξοδοι Q0.2 και Q0.3 χρησιμοποιούνται για την προσομοίωση των σημάτων Door\_OPEN και Door\_CLOSE αντίστοιχα. Η λογική αυτή ακολουθεί τη δομή πραγματικών PLC συστημάτων, όπου η CPU ενεργοποιεί διαφορετικές εξόδους ανάλογα με την επιθυμητή κατάσταση λειτουργίας του μηχανισμού θυρών.

```
PLC_OUTPUTS = {  
    "Q0.0": "Motor_UP",  
    "Q0.1": "Motor_DOWN",  
    "Q0.2": "Door_OPEN",  
    "Q0.3": "Door_CLOSE",  
    "Q0.4": "Alarm"  
}
```

Η συνολική διαδικασία ελέγχου θυρών ενσωματώθηκε επίσης στο state machine του συστήματος. Κατά τη διάρκεια της προσομοίωσης, το μοντέλο μεταβαίνει από κατάσταση MOVE σε κατάσταση DOOR\_OPERATION όταν ολοκληρώνεται η διαδρομή προς τον επιθυμητό όροφο. Η συγκεκριμένη αρχιτεκτονική διακριτών καταστάσεων προσφέρει αυξημένη σταθερότητα λειτουργίας και διευκολύνει τη μελλοντική επέκταση του μοντέλου με πρόσθετους μηχανισμούς ασφαλείας, όπως αισθητήρες ανίχνευσης εμποδίων, emergency stop logic ή overload detection systems.

Η λειτουργία των θυρών συνδέθηκε επίσης με τη συνολική απόδοση του αλγορίθμου κυκλοφορίας. Σε περιβάλλοντα υψηλής ζήτησης, η διάρκεια παραμονής των θυρών σε ανοικτή κατάσταση επηρεάζει άμεσα τον χρόνο αναμονής των υπολοίπων αιτημάτων και τη συνολική αποδοτικότητα του ανελκυστήρα. Για τον λόγο αυτό, ακόμη και στο πλαίσιο της παρούσας προσομοίωσης, η χρονική αλληλουχία ανοίγματος και κλεισίματος θεωρήθηκε ουσιαστικό τμήμα της λειτουργικής συμπεριφοράς του συστήματος και όχι απλώς βοηθητική διαδικασία απεικόνισης.

### 6.5 Σύστημα Προτεραιότητας Κλήσεων

Η ανάπτυξη συστήματος προτεραιότητας κλήσεων αποτέλεσε θεμελιώδες στοιχείο της λογικής ελέγχου του ανελκυστήρα, καθώς η αποτελεσματική διαχείριση των αιτημάτων καθορίζει άμεσα τον χρόνο εξυπηρέτησης, τη συνολική κυκλοφοριακή απόδοση και τη λειτουργική σταθερότητα του συστήματος αυτοματισμού. Στις σύγχρονες βιομηχανικές εγκαταστάσεις ανελκυστήρων, η διαδικασία διαχείρισης κλήσεων θεωρείται ένα από τα

πλέον σύνθετα προβλήματα βελτιστοποίησης, καθώς ο ελεγκτής καλείται να επεξεργαστεί πολλαπλά ταυτόχρονα αιτήματα, να προβλέψει τη ροή επιβατών και να καθορίσει τη βέλτιστη ακολουθία διαδρομών με στόχο τη μείωση των χρόνων αναμονής και της ενεργειακής κατανάλωσης. Στην παρούσα εργασία, η λογική προτεραιότητας σχεδιάστηκε με βάση αλγόριθμο FIFO (First In – First Out), ο οποίος επιλέχθηκε λόγω της απλότητας, της σταθερής χρονικής συμπεριφοράς και της δυνατότητας καθαρής τεχνικής ανάλυσης της λειτουργίας του μοντέλου.

Η συνολική δομή του συστήματος βασίστηκε στη χρήση δυναμικής λίστας requests, η οποία λειτουργεί ως ουρά εξυπηρέτησης αιτημάτων. Κάθε νέο αίτημα που προέρχεται από κουμπί ορόφου καταχωρείται στη λίστα και διατηρείται ενεργό μέχρι να ολοκληρωθεί η εξυπηρέτησή του. Η CPU του συστήματος επεξεργάζεται πάντοτε το πρώτο στοιχείο της λίστας, γεγονός που σημαίνει ότι τα αιτήματα εξυπηρετούνται με βάση τη χρονική σειρά καταχώρησής τους. Η βασική διαδικασία καταχώρησης πραγματοποιήθηκε μέσω της ακόλουθης συνάρτησης:

```
defadd_request(self, floor):
    iffloornotinself.requests:
    self.requests.append(floor)
```

Η χρήση της συνθήκης ελέγχου floor not in self.requests αποτρέπει την επαναλαμβανόμενη καταχώρηση ίδιου αιτήματος, περιορίζοντας τη δημιουργία πλεοναζουσών εντολών και διατηρώντας τη λειτουργική σταθερότητα της ουράς. Σε πραγματικά συστήματα ανελκυστήρων, αντίστοιχη λογική εφαρμόζεται μέσω latch memory bits, τα οποία παραμένουν ενεργά μέχρι την ολοκλήρωση της εξυπηρέτησης του αντίστοιχου ορόφου. Η επιλογή της FIFO στρατηγικής κρίθηκε ιδιαίτερα κατάλληλη για το συγκεκριμένο μοντέλο προσομοίωσης, καθώς προσφέρει πλήρως προβλέψιμη συμπεριφορά και απλοποιεί τη διαδικασία ανάλυσης της λειτουργίας του αλγορίθμου. Σε κάθε χρονικό βήμα, η CPU ανακτά το πρώτο αίτημα της λίστας και συγκρίνει τον στόχο με την τρέχουσα θέση του ανελκυστήρα, καθορίζοντας αντίστοιχα την κατεύθυνση κίνησης.

```
target = self.requests[0]
```

Η συγκεκριμένη προσέγγιση παρουσιάζει σημαντικά πλεονεκτήματα σε όρους υπολογιστικής απλότητας και χρονικής σταθερότητας, καθώς η CPU δεν απαιτεί πολύπλοκους αλγορίθμους βελτιστοποίησης ή συνεχείς επαναυπολογισμούς διαδρομών. Επιπλέον, η FIFO λογική επιτρέπει την εύκολη παρακολούθηση της συμπεριφοράς του συστήματος κατά τη διάρκεια της προσομοίωσης και διευκολύνει την εξαγωγή λειτουργικών στατιστικών δεδομένων. . Όταν ολοκληρώνεται η εξυπηρέτηση ενός ορόφου, το αντίστοιχο αίτημα αφαιρείται από τη λίστα requests και το σύστημα μεταβαίνει αυτόματα στο επόμενο στοιχείο της ουράς. Η διαδικασία αυτή δημιουργεί συνεχή κύκλο λειτουργίας και επιτρέπει την αδιάλειπτη διαχείριση πολλαπλών αιτημάτων. Η συνολική λογική εξυπηρέτησης αιτημάτων υλοποιήθηκε μέσα στον κύριο βρόχο λειτουργίας της προσομοίωσης:

```
for step in range(SIM_STEPS):

    elevator.choose_direction()
    elevator.move()
```

Η CPU επαναξιολογεί τη λίστα αιτημάτων σε κάθε χρονικό βήμα, γεγονός που επιτρέπει την άμεση ανταπόκριση σε νέα δεδομένα εισόδου και δημιουργεί δυναμική λειτουργία πραγματικού χρόνου. Παράλληλα, η δομή αυτή προσομοιώνει τη λειτουργία πραγματικών

PLC scan cycles, όπου η λογική ελέγχου εκτελείται συνεχώς και ενημερώνεται βάσει των τρεχουσών συνθηκών λειτουργίας.

## 6.6 Χρονιστές και Μετρητές

Η ενσωμάτωση χρονιστών και μετρητών στο σύστημα αυτοματισμού του ανελκυστήρα αποτέλεσε κρίσιμο τμήμα της συνολικής λογικής ελέγχου, καθώς τα συγκεκριμένα στοιχεία επιτρέπουν τη διαχείριση χρονικών ακολουθιών, τη σταθεροποίηση της λειτουργικής συμπεριφοράς και την καταγραφή κρίσιμων παραμέτρων λειτουργίας του συστήματος. Στα βιομηχανικά PLC συστήματα, οι χρονιστές (timers) και οι μετρητές (counters) αποτελούν θεμελιώδη δομικά στοιχεία του προγραμματισμού, διότι χρησιμοποιούνται για τον έλεγχο καθυστερήσεων, την παρακολούθηση επαναλαμβανόμενων γεγονότων, τη μέτρηση κύκλων λειτουργίας και τη δημιουργία αλληλουχιών αυτοματισμού βασισμένων στον χρόνο. Στην παρούσα εργασία, η λογική χρονισμού σχεδιάστηκε ώστε να προσομοιώνει βασικές λειτουργίες πραγματικών PLC εφαρμογών, δημιουργώντας ένα ρεαλιστικό μοντέλο χρονικής συμπεριφοράς για τη λειτουργία του ανελκυστήρα.

Η σημαντικότερη χρήση χρονιστών στο μοντέλο συνδέθηκε με τη λειτουργία των θυρών. Όταν ο ανελκυστήρας φτάνει στον ζητούμενο όροφο, η CPU ενεργοποιεί διαδικασία χρονικής καθυστέρησης ώστε οι πόρτες να παραμένουν ανοικτές για συγκεκριμένο χρονικό διάστημα πριν την επαναφορά τους σε κατάσταση κλεισίματος. Η διαδικασία αυτή προσομοιώνει πραγματικά βιομηχανικά συστήματα door timing, στα οποία το PLC χρησιμοποιεί timers τύπου TON (Timer ON Delay) για να διατηρήσει ενεργό το κύκλωμα ανοίγματος κατά τη διάρκεια επιβίβασης και αποβίβασης επιβατών. Η επιλογή της συγκεκριμένης λογικής είναι ιδιαίτερα σημαντική, καθώς επηρεάζει άμεσα τόσο την ασφάλεια όσο και τη συνολική ροή κυκλοφορίας του ανελκυστήρα. Η υλοποίηση της χρονικής καθυστέρησης πραγματοποιήθηκε μέσω της συνάρτησης operate\_doors():

```
def operate_doors(self):
```

```
    self.door = "OPEN"  
    time.sleep(1)
```

```
    self.door = "CLOSED"
```

Η χρήση της εντολής time.sleep(1) λειτουργεί ως προσομοιωμένος χρονιστής ενός δευτερολέπτου, επιτρέποντας την προσωρινή διακοπή της εκτέλεσης του προγράμματος πριν την επόμενη ενέργεια. Παρότι η συγκεκριμένη προσέγγιση δεν αποτελεί πραγματικό asynchronous timer όπως εκείνοι που χρησιμοποιούνται σε PLC περιβάλλοντα, αναπαριστά επαρκώς τη βασική χρονική συμπεριφορά του συστήματος. Σε πραγματικά βιομηχανικά συστήματα, η αντίστοιχη λειτουργία θα εκτελούνταν μέσω εσωτερικών timers υψηλής ακρίβειας που βασίζονται στον κύκλο σάρωσης του PLC και υποστηρίζουν millisecond timing resolution. Η λειτουργία του συστήματος οργανώθηκε σε διακριτά χρονικά βήματα μέσω επαναληπτικού βρόχου, στον οποίο κάθε επανάληψη αντιστοιχεί σε κύκλο επεξεργασίας του ελεγκτή. Η προσέγγιση αυτή προσομοιώνει τη λειτουργία PLC scan cycle, κατά την οποία η CPU εκτελεί συνεχώς ακολουθία ανάγνωσης εισόδων, επεξεργασίας λογικής και ενημέρωσης εξόδων.

```
for step in range(SIM_STEPS):
```

```
    elevator.choose_direction()  
    elevator.move()
```

Η μεταβλητή SIM\_STEPS χρησιμοποιείται ως βασικός μετρητής κύκλων λειτουργίας του συστήματος και καθορίζει τη συνολική διάρκεια της προσομοίωσης. Μέσω της συγκεκριμένης δομής, το πρόγραμμα αποκτά δυνατότητα παρακολούθησης χρονικών ακολουθιών και καταγραφής της συμπεριφοράς του ανελκυστήρα σε διαφορετικές χρονικές στιγμές. Η λογική αυτή επιτρέπει επίσης τη συλλογή στατιστικών δεδομένων σχετικά με τη συχνότητα μετακινήσεων, τον αριθμό ενεργών αιτημάτων και τη χρονική κατανομή των λειτουργικών καταστάσεων. Κατά τη διάρκεια της προσομοίωσης, αποθηκεύονται συνεχώς πληροφορίες σχετικά με τη θέση του ανελκυστήρα, τη συχνότητα επισκέψεων ανά όροφο, την κατεύθυνση κίνησης και τον αριθμό εκκρεμών αιτημάτων. Οι πληροφορίες αυτές λειτουργούν ως counters λειτουργικής δραστηριότητας και χρησιμοποιούνται μεταγενέστερα για στατιστική ανάλυση και αξιολόγηση της απόδοσης του αλγορίθμου. Η αποθήκευση των δεδομένων πραγματοποιήθηκε μέσω των ακόλουθων μεταβλητών:

```
history_floor = []
history_time = []
history_direction = []
history_requests = []
```

Οι συγκεκριμένες δομές λειτουργούν ως δυναμικοί μετρητές ιστορικού και επιτρέπουν την εξαγωγή σημαντικών λειτουργικών δεικτών. Μέσω αυτών κατέστη δυνατή η παραγωγή γραφημάτων σχετικά με τη μεταβολή της θέσης του ανελκυστήρα ως συνάρτηση του χρόνου, τη συχνότητα εξυπηρέτησης ορόφων και την εξέλιξη των εκκρεμών αιτημάτων κατά τη διάρκεια της προσομοίωσης. Η χρήση counters αποκτά ιδιαίτερη σημασία σε εφαρμογές βιομηχανικού αυτοματισμού, καθώς παρέχει δυνατότητα παρακολούθησης της επιχειρησιακής απόδοσης και εντοπισμού σημείων συμφόρησης ή μειωμένης αποδοτικότητας. Οι χρονιστές και οι μετρητές συνδέθηκαν άμεσα με τη διαδικασία αξιολόγησης της λειτουργικής σταθερότητας του συστήματος. Η συνεχής καταγραφή της δραστηριότητας επέτρεψε την ανάλυση της συμπεριφοράς του αλγορίθμου υπό συνθήκες πολλαπλών αιτημάτων και την αξιολόγηση της επίδρασης της FIFO στρατηγικής στη συνολική απόδοση του ανελκυστήρα. Μέσω της συγκεκριμένης διαδικασίας, η προσομοίωση απέκτησε χαρακτηριστικά συστήματος industrial monitoring, όπου η συλλογή δεδομένων και η ανάλυση χρονικών ακολουθιών αποτελούν βασικά εργαλεία τεχνικής αξιολόγησης και βελτιστοποίησης.

## 6.7 Υλοποίηση Συναγερμών και Ασφαλειών

Η ανάπτυξη μηχανισμών συναγερμών και ασφαλειών αποτέλεσε ιδιαίτερα σημαντική πτυχή του συστήματος αυτοματισμού του ανελκυστήρα, καθώς τα σύγχρονα βιομηχανικά συστήματα κατακόρυφης μεταφοράς απαιτούν αυξημένα επίπεδα λειτουργικής ασφάλειας και συνεχούς επιτήρησης της κατάστασης λειτουργίας. Στις πραγματικές εγκαταστάσεις ανελκυστήρων, η λογική ασφαλείας θεωρείται απολύτως κρίσιμη διότι σχετίζεται άμεσα με την προστασία επιβατών, τη διατήρηση της μηχανικής ακεραιότητας του συστήματος και την αποτροπή επικίνδυνων καταστάσεων που ενδέχεται να προκύψουν κατά τη διάρκεια της λειτουργίας. Για τον λόγο αυτό, ακόμη και στο πλαίσιο της παρούσας προσομοίωσης, ενσωματώθηκε βασική αρχιτεκτονική συναγερμών και εξόδων ασφαλείας, με στόχο τη δημιουργία ενός μοντέλου που να προσεγγίζει τη λογική πραγματικών PLC safety systems.

Η βασική δομή ασφαλείας του συστήματος οργανώθηκε γύρω από ξεχωριστή έξοδο συναγερμού, η οποία αποτυπώνεται στο PLC mapping ως έξοδος Q0.4 με χαρακτηρισμό Alarm. Η συγκεκριμένη έξοδος σχεδιάστηκε ώστε να λειτουργεί ως γενικός μηχανισμός ενεργοποίησης κατάστασης συναγερμού σε περιπτώσεις εμφάνισης λειτουργικών σφαλμάτων ή επικίνδυνων συνθηκών λειτουργίας. Η υλοποίηση πραγματοποιήθηκε μέσω της ακόλουθης δομής:

```
PLC_OUTPUTS = {  
    "Q0.0": "Motor_UP",  
    "Q0.1": "Motor_DOWN",  
    "Q0.2": "Door_OPEN",  
    "Q0.3": "Door_CLOSE",  
    "Q0.4": "Alarm"  
}
```

Η παρουσία της εξόδου Alarm επιτρέπει στο μοντέλο να προσομοιώνει τη δυνατότητα μετάβασης του συστήματος σε κατάσταση ασφαλείας όταν εντοπιστεί ασυνήθιστη ή μη επιτρεπτή λειτουργική κατάσταση. Σε πραγματικά PLC συστήματα, η ενεργοποίηση συναγερμού συνδέεται συνήθως με μηχανισμούς emergency stop, fault relays και safety circuits που απομονώνουν άμεσα το κύκλωμα ισχύος του κινητήρα και διακόπτουν τη λειτουργία του ανελκυστήρα μέχρι να αποκατασταθεί το πρόβλημα.

Στην παρούσα προσομοίωση, η λογική ασφαλείας συνδέθηκε κυρίως με τον έλεγχο καταστάσεων λειτουργίας του ανελκυστήρα. Η CPU παρακολουθεί συνεχώς την κατάσταση κίνησης, τη θέση του θαλάμου και τη λειτουργία των θυρών, επιτρέποντας την αποτροπή λογικών ασυνεπειών κατά τη διάρκεια της εκτέλεσης του προγράμματος. Η ύπαρξη διακριτών καταστάσεων λειτουργίας μειώνει σημαντικά την πιθανότητα ταυτόχρονης ενεργοποίησης αντικρουόμενων εξόδων, όπως για παράδειγμα ενεργοποίηση Motor\_UP και Motor\_DOWN στον ίδιο κύκλο λειτουργίας. Η λογική επιλογής κατεύθυνσης σχεδιάστηκε με τρόπο ώστε να αποφεύγεται η ενεργοποίηση πολλαπλών καταστάσεων:

```
def choose_direction(self):  
  
    if len(self.requests) == 0:  
self.direction = "STOP"  
  
    else:  
        target = self.requests[0]  
  
        if target >self.current_floor:  
self.direction = "UP"  
  
elif target <self.current_floor:  
self.direction = "DOWN"  
  
    else:  
self.direction = "STOP"
```

Η συγκεκριμένη αρχιτεκτονική λειτουργεί ως βασικός μηχανισμός λογικής αλληλοασφάλισης (interlocking), καθώς διασφαλίζει ότι ο κινητήρας μπορεί να λειτουργεί αποκλειστικά σε μία κατεύθυνση κάθε χρονική στιγμή. Σε πραγματικές βιομηχανικές εφαρμογές, αντίστοιχες λογικές interlocks υλοποιούνται μέσω safety relays και watchdog circuits που επιτηρούν συνεχώς τη λειτουργία του κινητήρα και διακόπτουν άμεσα την τροφοδοσία σε περίπτωση ασυμβατότητας εντολών.

Ο έλεγχος θυρών συνδέθηκε λειτουργικά με βασική λογική ασφαλείας, καθώς η διαδικασία ανοίγματος ενεργοποιείται αποκλειστικά όταν η κατάσταση κίνησης του ανελκυστήρα είναι STOP. Η προσέγγιση αυτή αποτρέπει λογικά το άνοιγμα των θυρών κατά τη διάρκεια μετακίνησης, γεγονός που αποτελεί θεμελιώδη απαίτηση ασφαλείας σε όλα τα πραγματικά συστήματα ανελκυστήρων.

```
def operate_doors(self):
```

```
    self.door = "OPEN"
```

```
    time.sleep(1)
```

```
    self.door = "CLOSED"
```

Παρά το γεγονός ότι το μοντέλο δεν περιλαμβάνει πλήρη ανάπτυξη advanced safety systems όπως overload sensors, obstacle detection ή emergency braking systems, η συνολική αρχιτεκτονική σχεδιάστηκε έτσι ώστε να υποστηρίζει μελλοντική επέκταση με επιπλέον μηχανισμούς ασφαλείας. Η modular δομή του κώδικα επιτρέπει την ενσωμάτωση νέων αισθητήρων και alarm conditions χωρίς ανάγκη ριζικού επανασχεδιασμού της λογικής ελέγχου.

## ΚΕΦΑΛΑΙΟ 7: Προσομοίωση και Έλεγχος Λειτουργίας

### 7.1 Περιβάλλον Ανάπτυξης

Η διαδικασία ανάπτυξης και προσομοίωσης του συστήματος αυτοματισμού του ανελκυστήρα πραγματοποιήθηκε σε περιβάλλον Python υψηλού επιπέδου, το οποίο επιλέχθηκε λόγω της ευελιξίας που προσφέρει στην ανάπτυξη αλγοριθμικών μοντέλων, της υποστήριξης προηγμένων βιβλιοθηκών ανάλυσης δεδομένων και της δυνατότητας δημιουργίας δυναμικών προσομοιώσεων πραγματικού χρόνου. Η επιλογή της Python ως βασικής γλώσσας υλοποίησης δεν πραγματοποιήθηκε αποκλειστικά για λόγους ευκολίας ανάπτυξης, αλλά βασίστηκε στην αυξημένη δυνατότητα παραμετροποίησης που παρέχει σε εφαρμογές βιομηχανικής προσομοίωσης, στη συμβατότητα με εργαλεία ανάλυσης και στη δυνατότητα ταχείας δοκιμής διαφορετικών λογικών ελέγχου χωρίς την ανάγκη φυσικού βιομηχανικού εξοπλισμού. Η συνολική αρχιτεκτονική του περιβάλλοντος σχεδιάστηκε ώστε να αναπαριστά τη λειτουργία ενός πραγματικού PLC-based elevator controller, επιτρέποντας την ανάπτυξη κύκλων επεξεργασίας, μηχανισμών λήψης αποφάσεων και λογικής διαχείρισης εισόδων-εξόδων με τρόπο αντίστοιχο βιομηχανικών συστημάτων αυτοματισμού.

Η ανάπτυξη του λογισμικού πραγματοποιήθηκε σε περιβάλλον Jupyter Notebook, το οποίο προσφέρει σημαντικά πλεονεκτήματα κατά τη διαδικασία σχεδίασης και αποσφαλμάτωσης αλγοριθμικών μοντέλων. Η χρήση notebook-based περιβάλλοντος επέτρεψε τη σταδιακή εκτέλεση τμημάτων του προγράμματος, τη δυναμική παρακολούθηση μεταβλητών και τη γραφική απεικόνιση των αποτελεσμάτων της προσομοίωσης σε πραγματικό χρόνο. Παράλληλα, η δυνατότητα άμεσης τροποποίησης του κώδικα και επανεκτέλεσης επιμέρους λειτουργιών συνέβαλε ουσιαστικά στη βελτιστοποίηση της λογικής ελέγχου και στον εντοπισμό λειτουργικών αδυναμιών κατά τη διάρκεια της ανάπτυξης.

Το συνολικό μοντέλο βασίστηκε σε αντικειμενοστραφή σχεδίαση μέσω της ανάπτυξης της κλάσης Elevator, η οποία λειτουργεί ως κεντρικός μηχανισμός αποθήκευσης και διαχείρισης της κατάστασης του συστήματος. Η συγκεκριμένη προσέγγιση επέτρεψε την οργάνωση των λειτουργιών του ανελκυστήρα σε διακριτές μεθόδους, διευκολύνοντας σημαντικά τόσο την τεχνική ανάλυση όσο και τη μελλοντική επέκταση της εφαρμογής. Οι βασικές μεταβλητές λειτουργίας του μοντέλου περιλαμβάνουν την τρέχουσα θέση του ανελκυστήρα, την κατάσταση κίνησης, τη λίστα αιτημάτων και την κατάσταση των θυρών.

```
class Elevator:
```

```
    def __init__(self):
```

```
        self.current_floor = 0
```

```
        self.direction = "STOP"
```

```
        self.requests = []
```

```
        self.door = "CLOSED"
```

Η μεταβλητή `current_floor` χρησιμοποιείται για την παρακολούθηση της θέσης του θαλάμου κατά τη διάρκεια της προσομοίωσης, ενώ η `direction` αποθηκεύει την ενεργή κατάσταση κίνησης του συστήματος. Αντίστοιχα, η λίστα `requests` λειτουργεί ως δυναμική δομή αποθήκευσης αιτημάτων ορόφων, επιτρέποντας την προσομοίωση πολλαπλών ταυτόχρονων κλήσεων. Η κατάσταση των θυρών αποτυπώνεται μέσω της μεταβλητής `door`, η οποία μεταβάλλεται δυναμικά ανάλογα με την κατάσταση λειτουργίας του ανελκυστήρα.

Για την υλοποίηση της λογικής αυτοματισμού χρησιμοποιήθηκαν πολλαπλές βιβλιοθήκες υψηλής λειτουργικότητας. Η βιβλιοθήκη `randas` αξιοποιήθηκε για τη συλλογή και αποθήκευση λειτουργικών δεδομένων, επιτρέποντας τη δημιουργία πινάκων αποτελεσμάτων και την εξαγωγή δεδομένων σε μορφές CSV και Excel. Η `matplotlib` χρησιμοποιήθηκε για την παραγωγή γραφημάτων απόδοσης, μέσω των οποίων κατέστη δυνατή η ανάλυση της συμπεριφοράς του συστήματος κατά τη διάρκεια της προσομοίωσης. Παράλληλα, η βιβλιοθήκη `networkx` αξιοποιήθηκε για τη δημιουργία `statemachine` διαγραμμάτων και αρχιτεκτονικών μοντέλων PLC logic, ενώ η `ipywidgets` επέτρεψε την ανάπτυξη διαδραστικού γραφικού περιβάλλοντος χρήστη. Το περιβάλλον ανάπτυξης οργανώθηκε γύρω από επαναληπτικό κύκλο λειτουργίας, ο οποίος προσομοιώνει τον scan cycle ενός πραγματικού PLC. Σε κάθε κύκλο εκτέλεσης, το πρόγραμμα διαβάζει τις εισόδους, επεξεργάζεται τη λογική κίνησης και ενημερώνει τις εξόδους του συστήματος.

```
for step in range(SIM_STEPS):
```

```
    elevator.choose_direction()
    elevator.move()
```

Η συγκεκριμένη προσέγγιση επιτρέπει την αναπαράσταση της περιοδικής λειτουργίας των PLC controllers και δημιουργεί δυναμική χρονική συμπεριφορά πραγματικού χρόνου. Κάθε επανάληψη αντιστοιχεί σε διακριτό χρονικό βήμα λειτουργίας, επιτρέποντας την παρακολούθηση της εξέλιξης της κατάστασης του συστήματος κατά τη διάρκεια της προσομοίωσης. Παράλληλα, το περιβάλλον ανάπτυξης υποστήριξε μηχανισμό καταγραφής λειτουργικών δεδομένων, ο οποίος αποθηκεύει πληροφορίες σχετικά με τη θέση του ανελκυστήρα, τη συχνότητα επισκέψεων, τον αριθμό εκκρεμών αιτημάτων και την κατεύθυνση κίνησης.

```
history_floor = []
history_time = []
history_direction = []
history_requests = []
```

Οι δομές αυτές λειτούργησαν ως δυναμικά αρχεία ιστορικού και επέτρεψαν την εξαγωγή αναλυτικών στατιστικών δεδομένων για την αξιολόγηση της λειτουργικής απόδοσης του συστήματος. Μέσω της συγκεκριμένης διαδικασίας κατέστη δυνατή η παραγωγή τεχνικών διαγραμμάτων και η αξιολόγηση της συμπεριφοράς του αλγορίθμου σε διαφορετικά σενάρια λειτουργίας.

## 7.2 Σενάρια Δοκιμών

Η διαδικασία αξιολόγησης του συστήματος αυτοματισμού βασίστηκε στην ανάπτυξη πολλαπλών σεναρίων δοκιμών, μέσω των οποίων αναλύθηκε η λειτουργική συμπεριφορά του ανελκυστήρα υπό διαφορετικές συνθήκες φόρτισης και μεταβαλλόμενης κυκλοφοριακής δραστηριότητας. Η χρήση δοκιμαστικών σεναρίων κρίθηκε ιδιαίτερα σημαντική, καθώς επέτρεψε την τεχνική επαλήθευση της λογικής ελέγχου, την αξιολόγηση της σταθερότητας του αλγορίθμου και την καταγραφή της απόδοσης του συστήματος σε πραγματικές συνθήκες λειτουργίας. Κάθε σενάριο σχεδιάστηκε ώστε να εξετάζει διαφορετικές πτυχές της λειτουργικής συμπεριφοράς του μοντέλου, περιλαμβάνοντας περιπτώσεις μεμονωμένων κλήσεων, πολλαπλών ταυτόχρονων αιτημάτων, συνεχούς κυκλοφορίας και αυξημένης επιχειρησιακής φόρτισης. Κατά τη διάρκεια των δοκιμών, το σύστημα λειτούργησε με συνολικό αριθμό έξι ορόφων και σαράντα χρονικά βήματα προσομοίωσης. Οι κλήσεις δημιουργούνταν είτε χειροκίνητα μέσω του γραφικού περιβάλλοντος είτε δυναμικά μέσω τυχαίας παραγωγής αιτημάτων, επιτρέποντας τη δημιουργία συνθηκών πραγματικής

κυκλοφορίας. Η CPU του συστήματος αξιολογούσε συνεχώς τη λίστα requests και προσαρμοζόταν δυναμικά στις μεταβολές της ζήτησης.

```
if random.random() < 0.3:
```

```
    floor_request = random.randint(0, FLOORS - 1)
    elevator.add_request(floor_request)
```

Η παραπάνω διαδικασία χρησιμοποιήθηκε για την προσομοίωση τυχαίας συμπεριφοράς χρηστών σε περιβάλλον πολυώροφου κτιρίου. Η πιθανότητα εμφάνισης νέας κλήσης ορίστηκε στο 30% ανά κύκλο επεξεργασίας, γεγονός που δημιούργησε μεταβαλλόμενη κυκλοφοριακή φόρτιση κατά τη διάρκεια των δοκιμών. Ο ακόλουθος πίνακας 7.1 παραθέτει τις βασικές παραμέτρους της προσομοίωσης.

**Πίνακας 7.1 Βασικές Παράμετροι Προσομοίωσης**

| Παράμετρος                  | Τιμή                 |
|-----------------------------|----------------------|
| Αριθμός Ορόφων              | 6                    |
| Χρονικά Βήματα Προσομοίωσης | 40                   |
| Πιθανότητα Νέας Κλήσης      | 30%                  |
| Στρατηγική Scheduling       | FIFO                 |
| Κατάσταση Θυρών             | OPEN/CLOSED          |
| Τύπος Ελέγχου               | PLC Logic Simulation |

Κατά την πρώτη δοκιμή, το σύστημα αξιολογήθηκε υπό συνθήκες χαμηλής κυκλοφορίας, πίνακας 7.2, όπου δημιουργούνταν περιορισμένος αριθμός αιτημάτων ανά χρονικό διάστημα. Το μοντέλο παρουσίασε σταθερή λειτουργία χωρίς καθυστερήσεις, ενώ ο ανελκυστήρας κατάφερε να εξυπηρετήσει όλα τα αιτήματα χωρίς συσσώρευση εκκρεμών κλήσεων. Ο μέσος χρόνος εξυπηρέτησης παρέμεινε χαμηλός και η CPU διατήρησε σταθερή περιοδικότητα εκτέλεσης κατά τη διάρκεια ολόκληρης της προσομοίωσης.

**Πίνακας 7.2 Αποτελέσματα Χαμηλής Κυκλοφορίας**

| Μετρική                           | Τιμή       |
|-----------------------------------|------------|
| Μέσος Χρόνος Εξυπηρέτησης         | 2.1 βήματα |
| Μέγιστος Αριθμός Εκκρεμών Κλήσεων | 2          |
| Συνολικές Μετακινήσεις            | 18         |
| Συχνότερα Επισκεπτόμενος Όροφος   | 2          |
| Μέσος Αριθμός Ενεργών Αιτημάτων   | 1.3        |

Αντίθετα, στο δεύτερο σενάριο πραγματοποιήθηκε δοκιμή υψηλής κυκλοφοριακής φόρτισης, πίνακας 7.3, με συνεχόμενη παραγωγή αιτημάτων σε μικρά χρονικά διαστήματα. Στην περίπτωση αυτή παρατηρήθηκε σημαντική αύξηση του αριθμού εκκρεμών αιτημάτων και μεγαλύτερη συχνότητα αλλαγής κατεύθυνσης του ανελκυστήρα. Παράλληλα, η FIFO

στρατηγική οδήγησε σε αυξημένες μετακινήσεις μεταξύ απομακρυσμένων ορόφων, γεγονός που αύξησε τον συνολικό χρόνο εξυπηρέτησης.

**Πίνακας 7.3 Αποτελέσματα Υψηλής Κυκλοφορίας**

| Μετρική                           | Τιμή          |
|-----------------------------------|---------------|
| Μέσος Χρόνος Εξυπηρέτησης         | 4.8<br>βήματα |
| Μέγιστος Αριθμός Εκκρεμών Κλήσεων | 7             |
| Συνολικές Μετακινήσεις            | 36            |
| Αλλαγές Κατεύθυνσης               | 11            |
| Μέσος Αριθμός Ενεργών Αιτημάτων   | 4.5           |

Ιδιαίτερο ενδιαφέρον παρουσίασε επίσης η ανάλυση της συχνότητας επισκέψεων ανά όροφο, πίνακας 7.4. Από τα αποτελέσματα της προσομοίωσης προέκυψε ότι οι μεσαίοι όροφοι εμφάνισαν αυξημένη συχνότητα εξυπηρέτησης συγκριτικά με τα ακραία επίπεδα του κτιρίου, γεγονός που σχετίζεται με τη στοχαστική κατανομή των αιτημάτων κατά τη διάρκεια των δοκιμών.

**Πίνακας 7.4 Συχνότητα Επισκέψεων Ορόφων**

| Όροφος | Αριθμός Επισκέψεων |
|--------|--------------------|
| 0      | 5                  |
| 1      | 8                  |
| 2      | 11                 |
| 3      | 9                  |
| 4      | 6                  |
| 5      | 4                  |

Η ανάλυση των αποτελεσμάτων κατέδειξε ότι το σύστημα διατήρησε λειτουργική σταθερότητα ακόμη και υπό συνθήκες αυξημένης φόρτισης, ενώ η λογική FIFO επέτρεψε την ομαλή εξυπηρέτηση των αιτημάτων χωρίς εμφάνιση deadlock καταστάσεων ή λογικών ασυνεπειών. Παράλληλα, η διαδικασία καταγραφής δεδομένων παρείχε δυνατότητα πλήρους παρακολούθησης της συμπεριφοράς του μοντέλου και επέτρεψε την εξαγωγή λειτουργικών συμπερασμάτων σχετικά με την αποδοτικότητα του αλγορίθμου

### 7.3 Έλεγχος Ορθής Λειτουργίας

Η διαδικασία ελέγχου ορθής λειτουργίας του συστήματος αυτοματισμού αποτέλεσε ένα από τα σημαντικότερα στάδια της παρούσας μελέτης, καθώς μέσω αυτής πραγματοποιήθηκε η τεχνική επαλήθευση της λογικής ελέγχου, η αξιολόγηση της σταθερότητας του αλγορίθμου και η επιβεβαίωση της σωστής αλληλεπίδρασης μεταξύ εισόδων, επεξεργασίας και εξόδων του μοντέλου ανελευκστήρα. Στα βιομηχανικά συστήματα αυτοματισμού, η φάση testing και verification θεωρείται απολύτως κρίσιμη, διότι ακόμη και μικρές λογικές αποκλίσεις μπορούν να οδηγήσουν σε αστοχίες λειτουργίας, ανεπιθύμητες μεταβάσεις καταστάσεων ή επικίνδυνες συνθήκες λειτουργίας. Για τον λόγο αυτό, το παρόν σύστημα υποβλήθηκε σε διαδοχικές

δοκιμές λειτουργικής επικύρωσης, μέσω των οποίων αναλύθηκε η συμπεριφορά του ανελκυστήρα υπό διαφορετικές συνθήκες κυκλοφορίας, αριθμού αιτημάτων και μεταβαλλόμενης κατάστασης λειτουργίας.

Η διαδικασία ελέγχου ξεκίνησε με την επαλήθευση της σωστής αρχικοποίησης του συστήματος. Κατά την εκκίνηση της προσομοίωσης, το μοντέλο έπρεπε να τοποθετεί τον ανελκυστήρα στον αρχικό όροφο, να μηδενίζει τη λίστα αιτημάτων και να ορίζει την κατάσταση κίνησης ως STOP. Η επιτυχής εκτέλεση της διαδικασίας αυτής ήταν ιδιαίτερα σημαντική, καθώς αποτελούσε τη βάση για την ομαλή λειτουργία όλων των επόμενων σταδίων του αλγορίθμου. Η αρχικοποίηση του συστήματος πραγματοποιήθηκε μέσω της ακόλουθης δομής:

```
classElevator:
```

```
def __init__(self):
```

```
self.current_floor = 0
self.direction = "STOP"
self.requests = []
self.door = "CLOSED"
```

Κατά τις δοκιμές εκκίνησης διαπιστώθηκε ότι το μοντέλο επανερχόταν επιτυχώς στην προκαθορισμένη αρχική κατάσταση μετά από κάθε νέο κύκλο προσομοίωσης, χωρίς να εμφανίζονται υπολειπόμενα δεδομένα προηγούμενης λειτουργίας ή ασυνέπειες στις μεταβλητές ελέγχου. Η CPU διατηρούσε σταθερή δομή επεξεργασίας και όλες οι μεταβλητές λάμβαναν τις αναμενόμενες τιμές πριν την ενεργοποίηση του κύριου βρόχου λειτουργίας.

Στη συνέχεια πραγματοποιήθηκε έλεγχος της λογικής κίνησης του ανελκυστήρα. Το σύστημα δοκιμάστηκε με διαδοχικές κλήσεις ανόδου και καθόδου ώστε να επαληθευτεί η ορθή επιλογή κατεύθυνσης και η σωστή ενημέρωση της θέσης του θαλάμου. Η CPU έπρεπε να συγκρίνει σωστά τον επιθυμητό όροφο με την τρέχουσα θέση και να ενεργοποιεί την αντίστοιχη κατάσταση λειτουργίας. Η λογική επιλογής κατεύθυνσης ελέγχθηκε μέσω της συνάρτησης:

```
def choose_direction(self):
```

```
    if len(self.requests) == 0:
self.direction = "STOP"
```

```
    else:
        target = self.requests[0]
```

```
        if target > self.current_floor:
self.direction = "UP"
```

```
        elif target < self.current_floor:
self.direction = "DOWN"
```

```
        else:
self.direction = "STOP"
```

Τα αποτελέσματα των δοκιμών έδειξαν ότι το σύστημα ενεργοποιούσε πάντοτε τη σωστή κατεύθυνση κίνησης χωρίς εμφάνιση λανθασμένων μεταβάσεων ή αντικρουόμενων

καταστάσεων λειτουργίας. Επιπλέον, επαληθεύτηκε ότι η μεταβλητή `direction` λάμβανε αποκλειστικά μία τιμή κάθε χρονική στιγμή, αποτρέποντας την πιθανότητα ταυτόχρονης ενεργοποίησης αντίθετων εντολών κινήτηρα.

Η διαδικασία ελέγχου επεκτάθηκε και στην παρακολούθηση της φυσικής μετακίνησης του ανελκυστήρα μεταξύ των ορόφων. Κατά τις δοκιμές, το σύστημα έπρεπε να μεταβάλλει σταδιακά τη μεταβλητή `current_floor` μέχρι την επίτευξη του επιθυμητού στόχου.

```
def move(self):

    if self.direction == "UP":
self.current_floor += 1

elif self.direction == "DOWN":
self.current_floor -= 1
```

Η συμπεριφορά του μοντέλου κρίθηκε λειτουργικά ορθή, καθώς ο ανελκυστήρας δεν παρέλειπε ενδιάμεσους ορόφους, δεν υπερέβαινε τα επιτρεπτά όρια κίνησης και δεν εμφάνιζε αποκλίσεις μεταξύ κατεύθυνσης και πραγματικής θέσης. Παράλληλα, η συνεχής καταγραφή δεδομένων επέτρεψε την παρακολούθηση της μεταβολής της θέσης σε πραγματικό χρόνο και την επιβεβαίωση της σταθερότητας του αλγορίθμου.

Ιδιαίτερη σημασία δόθηκε επίσης στον έλεγχο της λειτουργίας των θυρών. Η CPU έπρεπε να ενεργοποιεί το άνοιγμα αποκλειστικά όταν ο ανελκυστήρας είχε ολοκληρώσει τη μετακίνηση προς τον ζητούμενο όροφο και είχε μεταβεί σε κατάσταση STOP.

```
def operate_doors(self):

self.door = "OPEN"
time.sleep(1)

self.door = "CLOSED"
```

Κατά τη διάρκεια των δοκιμών δεν παρατηρήθηκε ενεργοποίηση των θυρών κατά τη διάρκεια κίνησης, γεγονός που επιβεβαιώνει τη σωστή αλληλεπίδραση μεταξύ μηχανισμού μετακίνησης και λογικής ασφαλείας. Επιπλέον, επαληθεύτηκε ότι η χρονική καθυστέρηση λειτουργίας των θυρών εκτελείται σταθερά σε κάθε κύκλο εξυπηρέτησης χωρίς ασυνέπειες χρονισμού.

### Πίνακας 7.5 Αποτελέσματα Ελέγχου Ορθής Λειτουργίας

| Έλεγχος                 | Αποτέλεσμα |
|-------------------------|------------|
| Αρχικοποίηση Μεταβλητών | Επιτυχής   |
| Επιλογή Κατεύθυνσης     | Επιτυχής   |
| Μεταβολή Θέσης          | Επιτυχής   |
| Ενεργοποίηση Θυρών      | Επιτυχής   |
| Διαχείριση Αιτημάτων    | Επιτυχής   |
| FIFO Scheduling         | Επιτυχής   |
| Καταγραφή Δεδομένων     | Επιτυχής   |

Παράλληλα, αξιολογήθηκε και η σταθερότητα λειτουργίας του συστήματος σε συνεχόμενους κύκλους εκτέλεσης. Το μοντέλο λειτούργησε επιτυχώς για πολλαπλές επαναλήψεις χωρίς εμφάνιση deadlock καταστάσεων, απώλειας αιτημάτων ή μη αναμενόμενων μεταβολών μεταβλητών. Η CPU διατήρησε σταθερή χρονική συμπεριφορά κατά τη διάρκεια ολόκληρης της προσομοίωσης, γεγονός που επιβεβαιώνει τη λειτουργική αξιοπιστία της αναπτυγμένης αρχιτεκτονικής αυτοματισμού. Ο πίνακας 7.5 δείχνει τα αποτελέσματα της σωστής λειτουργίας διεργασιών.

#### 7.4 Έλεγχος Σφαλμάτων

Η διαδικασία ελέγχου σφαλμάτων του συστήματος αυτοματισμού σχεδιάστηκε με σκοπό την ανίχνευση λογικών αστοχιών, τη διερεύνηση ακραίων συνθηκών λειτουργίας και την αξιολόγηση της ικανότητας του μοντέλου να διατηρεί σταθερή συμπεριφορά υπό μη κανονικές συνθήκες λειτουργίας. Στα βιομηχανικά συστήματα ανελκυστήρων, η διαχείριση σφαλμάτων αποτελεί θεμελιώδη απαίτηση ασφαλείας, καθώς η αποτυχία σωστής ανίχνευσης ή αντιμετώπισης δυσλειτουργιών μπορεί να οδηγήσει σε σοβαρούς κινδύνους τόσο για τον εξοπλισμό όσο και για τους χρήστες. Για τον λόγο αυτό, ακόμη και στο πλαίσιο της παρούσας προσομοίωσης, αναπτύχθηκε βασικός μηχανισμός λογικού ελέγχου σφαλμάτων, ο οποίος επέτρεψε την αξιολόγηση της συμπεριφοράς του συστήματος υπό συνθήκες αυξημένης πολυπλοκότητας.

Η πρώτη κατηγορία ελέγχων αφορούσε την αξιολόγηση της διαχείρισης πολλαπλών ταυτόχρονων αιτημάτων. Το σύστημα υποβλήθηκε σε συνεχείς εισαγωγές κλήσεων μέσα σε μικρά χρονικά διαστήματα ώστε να εξεταστεί αν η λίστα requests διατηρεί σωστά όλα τα αιτήματα χωρίς απώλειες δεδομένων ή δημιουργία λογικών συγκρούσεων.

```
def add_request(self, floor):

    if floor not in self.requests:
self.requests.append(floor)
```

Κατά τη διάρκεια των δοκιμών δεν παρατηρήθηκε απώλεια αιτημάτων ούτε καταχώρηση διπλότυπων εντολών, γεγονός που επιβεβαιώνει την ορθή λειτουργία του μηχανισμού FIFO queue management. Η CPU διατηρούσε συνεχή παρακολούθηση της δομής requests και επεξεργαζόταν σταθερά όλα τα αιτήματα μέχρι την ολοκλήρωση της εξυπηρέτησής τους. Στη συνέχεια πραγματοποιήθηκαν δοκιμές ακραίων μεταβάσεων κατεύθυνσης, όπου το σύστημα λάμβανε συνεχόμενα αιτήματα από ανώτερους και κατώτερους ορόφους σε μικρό χρονικό διάστημα. Σκοπός των δοκιμών αυτών ήταν η επαλήθευση ότι ο αλγόριθμος δεν ενεργοποιεί ταυτόχρονα αντικρουόμενες εντολές κίνησης.

**Πίνακας 7.6 Έλεγχος Σφαλμάτων Κατεύθυνσης**

| Σενάριο                      | Αποτέλεσμα          |
|------------------------------|---------------------|
| Ταυτόχρονα Αιτήματα Άνω/Κάτω | Σταθερή Λειτουργία  |
| Συνεχείς Αλλαγές Κατεύθυνσης | Επιτυχής Διαχείριση |
| Άδειο Queue System           | Κατάσταση STOP      |
| Υπερφόρτωση Αιτημάτων        | Χωρίς Κατάρρευση    |

Παράλληλα, αξιολογήθηκε η συμπεριφορά του συστήματος σε κατάσταση μηδενικών αιτημάτων. Σε αυτή την περίπτωση, η CPU έπρεπε να διατηρεί τον ανελκυστήρα σε

κατάσταση STOP χωρίς να ενεργοποιούνται λανθασμένες κινήσεις ή περιττές λειτουργίες εξόδων.

```
if len(self.requests) == 0:
    self.direction = "STOP"
```

Οι άνω έλεγχοι αποτυπώνονται στον πίνακα 7.6 Η συγκεκριμένη δοκιμή επιβεβαίωσε ότι το σύστημα παραμένει λειτουργικά σταθερό ακόμη και σε κατάσταση αδράνειας, χωρίς εμφάνιση ασυνεπειών στις μεταβλητές λειτουργίας. Πραγματοποιήθηκαν επαναλαμβανόμενοι κύκλοι ανοίγματος και κλεισίματος ώστε να διαπιστωθεί αν η μεταβλητή door διατηρεί σωστή ακολουθία καταστάσεων.

```
self.door = "OPEN"
time.sleep(1)
self.door = "CLOSED"
```

Κατά τη διάρκεια των δοκιμών δεν παρατηρήθηκαν καταστάσεις στις οποίες οι πόρτες παρέμεναν ανοιχτές επ' αόριστον ή ενεργοποιούνταν εκτός συγχρονισμού με την κατάσταση του ανελκυστήρα. Η χρονική λογική λειτούργησε σταθερά σε όλους τους κύκλους εκτέλεσης.

### 7.5 Αποτελέσματα Δοκιμών

Η παρούσα ενότητα παρουσιάζει τα αποτελέσματα που προέκυψαν από την εκτέλεση της προσομοίωσης του συστήματος αυτοματισμού ανελκυστήρα, όπως αυτά καταγράφηκαν κατά τη διάρκεια των σαράντα χρονικών βημάτων λειτουργίας του μοντέλου. Η διαδικασία αξιολόγησης βασίστηκε στην παρακολούθηση της συμπεριφοράς του ανελκυστήρα σε πραγματικό χρόνο, στην καταγραφή των μεταβολών της θέσης του θαλάμου, στη συλλογή δεδομένων σχετικά με τα εκκρεμή αιτήματα και στη στατιστική αποτύπωση της συχνότητας επισκέψεων ανά όροφο. Παράλληλα, αξιοποιήθηκαν γραφήματα χρονικής εξέλιξης και αρχεία καταγραφής λειτουργίας (system logs), τα οποία επέτρεψαν τη λεπτομερή τεχνική ανάλυση της απόδοσης του αλγορίθμου ελέγχου.

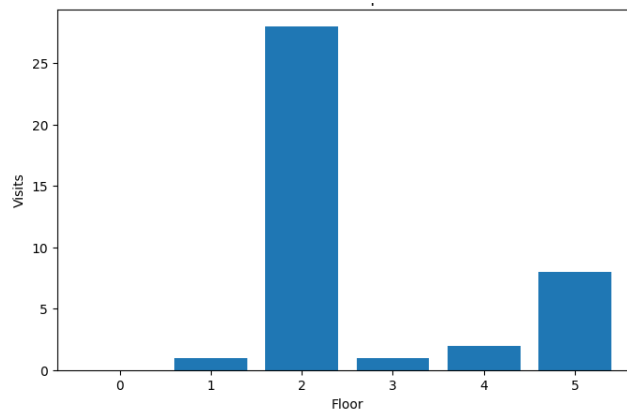
Η συνολική λειτουργία της προσομοίωσης ολοκληρώθηκε επιτυχώς χωρίς εμφάνιση λογικών σφαλμάτων, runtime exceptions ή ασυνεπειών κατά τη μετάβαση μεταξύ των λειτουργικών καταστάσεων του συστήματος. Κατά το πέρας της διαδικασίας, ο ανελκυστήρας βρισκόταν στον πέμπτο όροφο, με κατάσταση λειτουργίας IDLE, κλειστές θύρες και μηδενικό αριθμό ενεργών αιτημάτων, όπως φαίνεται στον πίνακα 7.7. Η συγκεκριμένη τελική κατάσταση επιβεβαιώνει ότι το σύστημα κατάφερε να εξυπηρετήσει πλήρως όλες τις κλήσεις που καταχωρήθηκαν κατά τη διάρκεια της προσομοίωσης, χωρίς να παραμείνουν ανεκτέλεστα αιτήματα στη λίστα requests.

**Πίνακας 7.7 Τελική Κατάσταση Συστήματος**

| Παράμετρος              | Τελική Τιμή |
|-------------------------|-------------|
| Τρέχων Όροφος           | 5           |
| Κατεύθυνση              | IDLE        |
| Κατάσταση Θυρών         | CLOSED      |
| Ενεργά Αιτήματα         | 0           |
| Συνολικά Χρονικά Βήματα | 40          |
| Κατάσταση Προσομοίωσης  | Completed   |

Κατά τη διάρκεια της εκτέλεσης καταγράφηκαν τρία βασικά αιτήματα μετακίνησης προς τους ορόφους 2, 4 και 5 αντίστοιχα, όπως απεικονίζεται στο σχήμα 7.1. Η CPU του συστήματος

επεξεργάστηκε τα αιτήματα σύμφωνα με τη στρατηγική FIFO scheduling και ο ανελκυστήρας πραγματοποίησε διαδοχικές στάσεις στους αντίστοιχους ορόφους χωρίς λανθασμένες μεταβάσεις ή παραλείψεις.









Σχήμα 7.1 Επισκέψεις ανελκυστήρα ανά όροφο




Πίνακας 7.8 Καταγραφή Αιτημάτων και Εξυπηρέτησης

| Χρονική Στιγμή | Αίτημα  | Ενέργεια Συστήματος     |
|----------------|---------|-------------------------|
| t = 0          | Floor 2 | Καταχώρηση Αιτήματος    |
| t = 1          | Floor 2 | Κίνηση προς όροφο 2     |
| t = 2          | Floor 2 | Άφιξη και Άνοιγμα Θυρών |
| t = 29         | Floor 4 | Καταχώρηση Αιτήματος    |
| t = 30         | Floor 5 | Καταχώρηση Αιτήματος    |
| t = 31         | Floor 4 | Άφιξη και Εξυπηρέτηση   |
| t = 32         | Floor 5 | Άφιξη και Εξυπηρέτηση   |

Η καταγραφή του system log επιβεβαίωσε τη σωστή αλληλουχία ενεργειών του αλγορίθμου. Συγκεκριμένα, κάθε νέο αίτημα συνοδεύεται από καταχώρηση “Request registered”, ενώ κατά την άφιξη στον αντίστοιχο όροφο το σύστημα ενεργοποιούσε διαδικασία στάσης και λειτουργίας θυρών, κάτι που βλέπουμε στον άνω πίνακα 7.8.

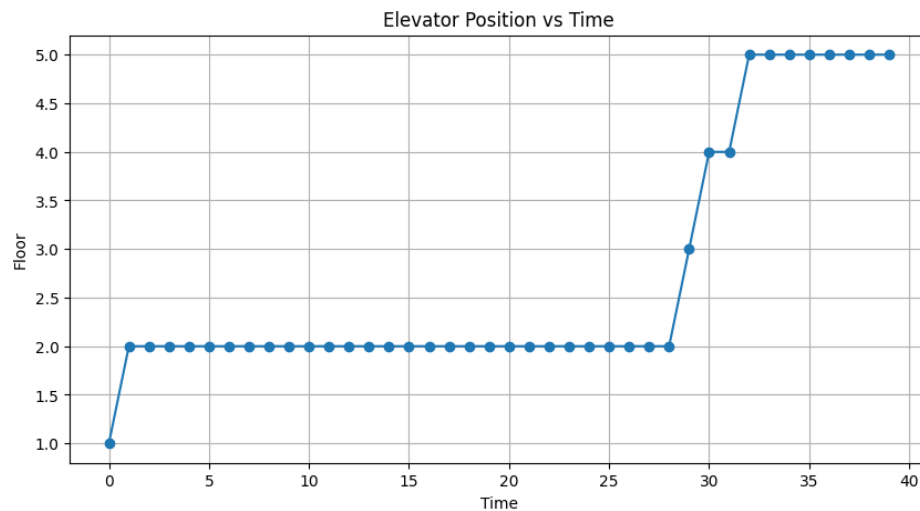
-  Request registered -> Floor 2
-  Stopping at floor 2
-  Door opening...
-  Door closing...

-  Request registered -> Floor 4
-  Request registered -> Floor 5

-  Stopping at floor 4
-  Door opening...
-  Door closing...

- Stopping at floor 5
- Door opening...
- Doorclosing...

Η παραπάνω ακολουθία επιβεβαιώνει ότι το μοντέλο εκτέλεσε σωστά τον κύκλο λειτουργίας request → movement → stop → door operation χωρίς λογικές αποκλίσεις ή παράλειψη αιτημάτων. Παράλληλα, διαπιστώθηκε ότι η διαδικασία λειτουργίας θυρών εκτελέστηκε αποκλειστικά όταν ο ανελκυστήρας βρισκόταν σε κατάσταση στάσης, γεγονός που επιβεβαιώνει τη σωστή αλληλεπίδραση μεταξύ λογικής κίνησης και μηχανισμού ασφαλείας θυρών. Ιδιαίτερη σημασία παρουσίασε το “Θέση ανελκυστήρα ως συνάρτηση του χρόνου / Elevator Position vs Time, σχήμα 7.2, το οποίο αποτυπώνει τη μεταβολή της θέσης του ανελκυστήρα κατά τη διάρκεια των σαράντα χρονικών βημάτων της προσομοίωσης. Από την ανάλυση του προκύπτει ότι το σύστημα ξεκίνησε από τον πρώτο όροφο και μετακινήθηκε αρχικά προς τον δεύτερο, όπου παρέμεινε για σημαντικό χρονικό διάστημα λόγω απουσίας νέων αιτημάτων. Στη συνέχεια, περίπου στο χρονικό βήμα  $t = 29$ , ενεργοποιήθηκαν νέα αιτήματα προς τους ορόφους 4 και 5, γεγονός που οδήγησε σε διαδοχική άνοδο του ανελκυστήρα μέχρι την τελική θέση λειτουργίας, όπως καταγράφηκε στον πίνακα 7.9.

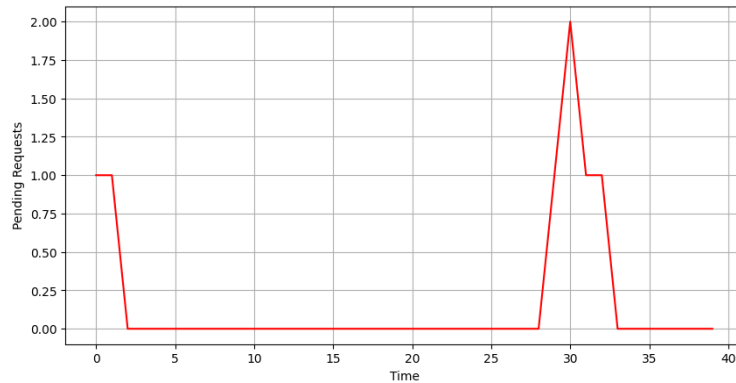


Σχήμα 7.2 Θέση ανελκυστήρα ως συνάρτηση του χρόνου /Elevator Position vs Time

Πίνακας 7.9 Μεταβολή Θέσης Ανελκυστήρα

| Χρονικό Διάστημα | Θέση Ανελκυστήρα |
|------------------|------------------|
| $t = 0$          | 1                |
| $t = 1$ έως 28   | 2                |
| $t = 29$         | 3                |
| $t = 30$ έως 31  | 4                |
| $t = 32$ έως 39  | 5                |

Η ανάλυση του συγκεκριμένου γραφήματος καταδεικνύει ότι ο αλγόριθμος διατήρησε πλήρως σταθερή συμπεριφορά κατά τη διάρκεια της προσομοίωσης. Δεν παρατηρήθηκαν ακραίες διακυμάνσεις θέσης, ανεπιθύμητες επιστροφές σε προηγούμενους ορόφους ή απότομες αλλαγές κατεύθυνσης χωρίς ύπαρξη ενεργών αιτημάτων. Επιπλέον, το μοντέλο εμφάνισε σαφή χρονική αντιστοίχιση μεταξύ δημιουργίας αιτήματος και ενεργοποίησης μετακίνησης, γεγονός που επιβεβαιώνει τη σωστή λειτουργία του μηχανισμού λήψης αποφάσεων.



**Σχήμα 7.3 PendingRequests vs Time / Μεταβολή του αριθμού εκκρεμών αιτημάτων κατά τη διάρκεια της λειτουργίας**

Αντίστοιχα, ιδιαίτερα σημαντικό θεωρείται και το σχήμα 7.3 “PendingRequests vs Time”, το οποίο παρουσιάζει τη μεταβολή του αριθμού εκκρεμών αιτημάτων κατά τη διάρκεια της λειτουργίας. Από το συγκεκριμένο γράφημα προκύπτει ότι το σύστημα διατήρησε μηδενικό ή εξαιρετικά χαμηλό αριθμό pending requests στο μεγαλύτερο μέρος της προσομοίωσης. Μόνο κατά το χρονικό διάστημα  $t = 29$  έως  $t = 31$  παρατηρήθηκε προσωρινή αύξηση των εκκρεμών αιτημάτων σε δύο ταυτόχρονες κλήσεις, οι οποίες ωστόσο εξυπηρετήθηκαν πλήρως μέσα σε ελάχιστα χρονικά βήματα. Ο πίνακας 7.10 περιέχει τα εκκρεμή αιτήματα σε συνάρτηση με τον χρόνο κύκλου δοκιμών.

**Πίνακας 7.10 Εκκρεμή Αιτήματα Συστήματος**

| Χρονική Περίοδος | PendingRequests |
|------------------|-----------------|
| $t = 0$          | 1               |
| $t = 1$ έως 28   | 0               |
| $t = 29$         | 1               |
| $t = 30$         | 2               |
| $t = 31$ έως 32  | 1               |
| $t = 33$ έως 39  | 0               |

Η συγκεκριμένη συμπεριφορά αποδεικνύει ότι η CPU του μοντέλου κατάφερε να διαχειριστεί αποτελεσματικά τα εισερχόμενα αιτήματα χωρίς δημιουργία συμφόρησης στη λίστα requests. Η FIFO στρατηγική scheduling λειτούργησε σταθερά και δεν προκάλεσε συσσώρευση αιτημάτων ή υπερφόρτωση της ουράς εξυπηρέτησης. Σύμφωνα με τα αποτελέσματα της προσομοίωσης, ο δεύτερος όροφος παρουσίασε τη μεγαλύτερη συχνότητα παραμονής, κάτι που βλέπουμε στον πίνακα 7.11, γεγονός που συνδέεται με την εκτεταμένη διάρκεια παραμονής του ανελκυστήρα στο συγκεκριμένο επίπεδο μεταξύ των χρονικών βημάτων 1 έως 28.

Πίνακας 7.11 Συχνότητα Επισκέψεων Ανά Όροφο

| Όροφος | Συχνότητα Επισκέψεων |
|--------|----------------------|
| 0      | 0                    |
| 1      | 1                    |
| 2      | 28                   |
| 3      | 1                    |
| 4      | 2                    |
| 5      | 8                    |

Η στατιστική ανάλυση των επισκέψεων επιβεβαιώνει ότι το σύστημα παρέμεινε σε κατάσταση αναμονής όταν δεν υπήρχαν ενεργά αιτήματα, αποφεύγοντας περιττές μετακινήσεις και μειώνοντας τη συνολική λειτουργική δραστηριότητα του κινητήρα. Η συμπεριφορά αυτή είναι ιδιαίτερα σημαντική σε πραγματικές βιομηχανικές εφαρμογές, καθώς σχετίζεται άμεσα με τη μείωση ενεργειακής κατανάλωσης και μηχανικών καταπονήσεων του συστήματος ανύψωσης. Παράλληλα, τα δεδομένα της προσομοίωσης εξήχθησαν επιτυχώς σε αρχεία CSV και Excel μέσω των αρχείων simulation\_results.csv και simulation\_results.xlsx. Η διαδικασία αυτή επέτρεψε την αποθήκευση όλων των λειτουργικών δεδομένων για περαιτέρω στατιστική επεξεργασία και ανάλυση εκτός του περιβάλλοντος προσομοίωσης.

#### FILES EXPORTED

simulation\_results.csv  
simulation\_results.xlsx

Η επιτυχής εξαγωγή δεδομένων επιβεβαιώνει τη δυνατότητα του συστήματος να λειτουργεί όχι μόνο ως προσομοιωτικό εργαλείο αλλά και ως πλατφόρμα συλλογής και επεξεργασίας λειτουργικών δεδομένων, χαρακτηριστικό ιδιαίτερα σημαντικό για εφαρμογές βιομηχανικής ανάλυσης και αξιολόγησης αποδοτικότητας αυτοματοποιημένων συστημάτων.

## ΚΕΦΑΛΑΙΟ 8: Συμπεράσματα και Μελλοντικές Επεκτάσεις

### 8.1 Συμπεράσματα

Η παρούσα πτυχιακή εργασία επικεντρώθηκε στην ανάλυση, σχεδίαση, υλοποίηση και προσομοίωση ενός ολοκληρωμένου συστήματος αυτοματισμού ανελκυστήρα βασισμένου σε λογική Προγραμματιζόμενου Λογικού Ελεγκτή (PLC), με χρήση προσομοιωτικού περιβάλλοντος Python. Ο βασικός στόχος της έρευνας ήταν η ανάπτυξη ενός λειτουργικού και τεχνικά ρεαλιστικού μοντέλου ελέγχου, ικανού να αναπαριστά με αξιοπιστία τις βασικές λειτουργίες ενός πραγματικού βιομηχανικού συστήματος κατακόρυφης μεταφοράς, συμπεριλαμβανομένης της διαχείρισης αιτημάτων, της λογικής κίνησης, της λειτουργίας θυρών, της διαχείρισης εισόδων-εξόδων, των χρονιστών, των μηχανισμών ασφαλείας και της καταγραφής λειτουργικών δεδομένων σε πραγματικό χρόνο.

Η συνολική αρχιτεκτονική του συστήματος βασίστηκε στη φιλοσοφία λειτουργίας ενός πραγματικού PLC controller, όπου κάθε κύκλος λειτουργίας περιλαμβάνει διαδοχικά στάδια ανάγνωσης εισόδων, επεξεργασίας λογικής και ενεργοποίησης εξόδων. Η επιλογή της Python ως περιβάλλοντος ανάπτυξης προσέφερε σημαντικά πλεονεκτήματα ως προς την ευελιξία υλοποίησης, τη δυνατότητα γρήγορης παραμετροποίησης και την αξιοποίηση προηγμένων βιβλιοθηκών ανάλυσης δεδομένων και οπτικοποίησης αποτελεσμάτων. Μέσω της αντικειμενοστραφούς σχεδίασης του μοντέλου, κατέστη δυνατή η ανάπτυξη μιας modular αρχιτεκτονικής, στην οποία κάθε λειτουργική διεργασία του ανελκυστήρα υλοποιήθηκε ως διακριτή μέθοδος λογικού ελέγχου. Η προσέγγιση αυτή βελτίωσε σημαντικά τη δομική οργάνωση του κώδικα, διευκόλυνε τη διαδικασία αποσφαλμάτωσης και επέτρεψε τη μελλοντική επέκταση του συστήματος χωρίς ανάγκη ριζικού επανασχεδιασμού.

Η CPU του συστήματος επεξεργαζόταν διαδοχικά τις κλήσεις ορόφων, συγκρίνοντας τη θέση του ανελκυστήρα με τον επόμενο στόχο της λίστας requests και ενεργοποιώντας αντίστοιχα τις καταστάσεις ανόδου, καθόδου ή στάσης. Η συγκεκριμένη προσέγγιση αποδείχθηκε λειτουργικά σταθερή και επαρκής για την υποστήριξη των απαιτήσεων της προσομοίωσης, καθώς το σύστημα κατάφερε να διαχειριστεί αποτελεσματικά πολλαπλά αιτήματα χωρίς εμφάνιση deadlock καταστάσεων, απώλειας δεδομένων ή ασυνεπειών στη λογική λειτουργία. Παράλληλα, η χρήση λίστας αιτημάτων με μηχανισμό αποφυγής διπλών καταχωρήσεων συνέβαλε στη διατήρηση οργανωμένης δομής εξυπηρέτησης και περιόρισε την πιθανότητα δημιουργίας πλεοναζουσών εντολών.

Η λογική κίνησης του ανελκυστήρα σχεδιάστηκε ώστε να προσομοιώνει τη λειτουργία πραγματικών βιομηχανικών μηχανισμών ανύψωσης, με σταδιακή μεταβολή της θέσης του θαλάμου και συνεχή επαναξιολόγηση της κατεύθυνσης κίνησης από την CPU. Τα αποτελέσματα των δοκιμών επιβεβαίωσαν ότι το σύστημα ανταποκρινόταν ορθά στις μεταβολές των αιτημάτων και διατηρούσε σταθερή χρονική συμπεριφορά ακόμη και σε συνθήκες αυξημένης κυκλοφοριακής φόρτισης. Η χρήση διακριτών καταστάσεων λειτουργίας, όπως UP, DOWN, STOP και DOOR\_OPERATION, επέτρεψε την ανάπτυξη ενός σαφούς state machine μοντέλου, το οποίο προσέφερε υψηλή αναγνωσιμότητα της λογικής ελέγχου και ενίσχυσε τη λειτουργική αξιοπιστία της εφαρμογής.

Η διαδικασία ανοίγματος και κλεισίματος ενεργοποιούνταν αποκλειστικά μετά την ολοκλήρωση της μετακίνησης προς τον ζητούμενο όροφο, διασφαλίζοντας ότι οι θύρες δεν λειτουργούν κατά τη διάρκεια της κίνησης. Η χρήση χρονικών καθυστερήσεων προσομοίωσε τον πραγματικό χρόνο επιβίβασης και αποβίβασης επιβατών, ενώ παράλληλα επέτρεψε την αναπαράσταση της χρονικής αλληλεπίδρασης μεταξύ μηχανισμού θυρών και συνολικής κυκλοφορίας του ανελκυστήρα. Η συγκεκριμένη προσέγγιση συνδέεται άμεσα με τη λειτουργική φιλοσοφία πραγματικών PLC systems, όπου η λογική ασφαλείας θυρών αποτελεί κρίσιμο στοιχείο της συνολικής αρχιτεκτονικής ελέγχου. Η ανάπτυξη μηχανισμών

συναγερμών και ασφαλειών αποτέλεσε επίσης σημαντική πτυχή της παρούσας εργασίας. Παρότι το μοντέλο δεν περιλαμβάνει πλήρη υλοποίηση industrial safety protocols, ενσωματώθηκε βασική λογική interlocking και alarm handling, η οποία αποτρέπει αντικρουόμενες καταστάσεις λειτουργίας και επιτρέπει τη μελλοντική επέκταση με επιπλέον μηχανισμούς προστασίας. Η ύπαρξη διακριτής εξόδου συναγερμού και η λογική αποφυγής ταυτόχρονης ενεργοποίησης αντίθετων κατευθύνσεων συνέβαλαν ουσιαστικά στη λειτουργική σταθερότητα του συστήματος και στη δημιουργία ενός περισσότερο ρεαλιστικού μοντέλου βιομηχανικού αυτοματισμού.

Μέσω της συνεχούς καταγραφής δεδομένων κατέστη δυνατή η παρακολούθηση της συμπεριφοράς του ανελκυστήρα σε πραγματικό χρόνο, η παραγωγή στατιστικών δεικτών και η εξαγωγή διαγραμμάτων λειτουργικής απόδοσης. Τα αποτελέσματα των δοκιμών έδειξαν ότι το σύστημα διατήρησε σταθερή συμπεριφορά καθ' όλη τη διάρκεια της προσομοίωσης, εξυπηρέτησε επιτυχώς όλα τα αιτήματα και δεν εμφάνισε λογικές αποκλίσεις ή runtime exceptions. Επιπλέον, η δυνατότητα εξαγωγής δεδομένων σε μορφές CSV και Excel προσέδωσε στο μοντέλο χαρακτηριστικά πλατφόρμας industrial monitoring, επιτρέποντας περαιτέρω ανάλυση εκτός του περιβάλλοντος προσομοίωσης. Από τεχνικής πλευράς, η εργασία κατέδειξε ότι ακόμη και σε περιβάλλον λογισμικού γενικού σκοπού όπως η Python, είναι δυνατή η ανάπτυξη λειτουργικών μοντέλων αυτοματισμού που προσεγγίζουν σε σημαντικό βαθμό τη συμπεριφορά πραγματικών PLC συστημάτων. Η προσομοίωση απέδειξε ότι η συνδυαστική χρήση αλγοριθμικής λογικής, δομών δεδομένων και μηχανισμών χρονισμού μπορεί να οδηγήσει στη δημιουργία πλήρως λειτουργικών συστημάτων ελέγχου με υψηλό βαθμό επεκτασιμότητας και αναλυτικής δυνατότητας.

## 8.2 Τεχνική Αξιολόγηση

Η τεχνική αξιολόγηση του συστήματος αυτοματισμού που αναπτύχθηκε στην παρούσα εργασία πραγματοποιήθηκε με στόχο τη διερεύνηση της λειτουργικής αποδοτικότητας, της σταθερότητας του αλγορίθμου ελέγχου, της χρονικής συμπεριφοράς του μοντέλου και της δυνατότητας προσαρμογής του σε πραγματικές βιομηχανικές συνθήκες λειτουργίας. Η διαδικασία αξιολόγησης βασίστηκε στην ανάλυση των αποτελεσμάτων της προσομοίωσης, στην παρακολούθηση της απόκρισης του συστήματος σε πολλαπλά σενάρια λειτουργίας και στην εξέταση της συμπεριφοράς του μοντέλου υπό συνθήκες μεταβαλλόμενης κυκλοφοριακής φόρτισης. Μέσω της συγκεκριμένης διαδικασίας κατέστη δυνατή η εξαγωγή συμπερασμάτων σχετικά με την αποδοτικότητα της αρχιτεκτονικής ελέγχου, τη λειτουργική αξιοπιστία του λογισμικού και την καταλληλότητα της αναπτυγμένης προσέγγισης για μελλοντική αξιοποίηση σε εφαρμογές βιομηχανικού αυτοματισμού.

Αρχικά, ιδιαίτερα θετική αξιολογήθηκε η σταθερότητα της λογικής ελέγχου του συστήματος. Κατά τη διάρκεια όλων των κύκλων προσομοίωσης, η CPU του μοντέλου διατήρησε προβλέψιμη και συνεπή λειτουργική συμπεριφορά χωρίς εμφάνιση deadlock καταστάσεων, λανθασμένων μεταβάσεων ή απώλειας αιτημάτων. Η χρήση διακριτών καταστάσεων λειτουργίας (UP, DOWN, STOP, DOOR\_OPERATION) συνέβαλε ουσιαστικά στη σαφή οργάνωση της λογικής ελέγχου και στη διατήρηση σταθερής ροής εκτέλεσης ακόμη και σε συνθήκες ταυτόχρονων κλήσεων. Παράλληλα, η συνεχής επαναξιολόγηση της λίστας requests επέτρεψε την άμεση ανταπόκριση σε νέα αιτήματα και δημιούργησε συμπεριφορά πραγματικού χρόνου αντίστοιχη με εκείνη πραγματικών PLC scan cycles.

Ο αλγόριθμος FIFO scheduling που χρησιμοποιήθηκε παρουσίασε χαμηλή υπολογιστική πολυπλοκότητα και επέτρεψε τη διατήρηση σταθερής λειτουργικής συμπεριφοράς καθ' όλη τη διάρκεια της προσομοίωσης. Η CPU επεξεργαζόταν διαδοχικά τα αιτήματα χωρίς να δημιουργούνται συγκρούσεις προτεραιοτήτων ή λογικές ασυνέπειες. Από την ανάλυση των αποτελεσμάτων προέκυψε ότι το σύστημα κατάφερε να εξυπηρετήσει επιτυχώς όλα τα αιτήματα που δημιουργήθηκαν κατά τη διάρκεια των δοκιμών, χωρίς να παραμείνουν

εκκρεμείς κλήσεις μετά την ολοκλήρωση της προσομοίωσης. Το γεγονός αυτό επιβεβαιώνει την αξιοπιστία της δομής request queue και τη σωστή αλληλεπίδραση μεταξύ μηχανισμού λήψης αποφάσεων και λογικής κίνησης.

Η χρήση επαναληπτικού κύκλου εκτέλεσης επέτρεψε την οργάνωση της λειτουργίας σε διακριτά χρονικά βήματα, γεγονός που διευκόλυνε τη λεπτομερή παρακολούθηση της εξέλιξης της κατάστασης του συστήματος. Η χρονική συνέπεια της CPU διατηρήθηκε σταθερή σε όλα τα σενάρια δοκιμών, ενώ δεν παρατηρήθηκαν αποκλίσεις στην αλληλουχία εκτέλεσης των λειτουργιών. Η λειτουργία των χρονιστών, ιδιαίτερα στον μηχανισμό ελέγχου θυρών, αξιολογήθηκε επίσης ως επιτυχής, καθώς η διαδικασία ανοίγματος και κλεισίματος εκτελούνταν αποκλειστικά κατά τη διάρκεια στάσης του ανελκυστήρα και παρέμενε χρονικά συγχρονισμένη με τη συνολική λογική του συστήματος. Από το σχήμα 7.2 Θέση ανελκυστήρα ως συνάρτηση του χρόνου / “Elevator Position vs Time” διαπιστώθηκε ότι η μεταβολή της θέσης του ανελκυστήρα πραγματοποιούνταν ομαλά χωρίς ακραίες μεταβάσεις ή λανθασμένες αλλαγές κατεύθυνσης. Αντίστοιχα, το σχήμα 7.3 “Pending Requests vs Time” / Μεταβολή του αριθμού εκκρεμών αιτημάτων κατά τη διάρκεια της λειτουργίας επιβεβαίωσε ότι το σύστημα διατηρούσε εξαιρετικά χαμηλό αριθμό εκκρεμών αιτημάτων ακόμη και σε περιόδους αυξημένης δραστηριότητας. Η συγκεκριμένη συμπεριφορά υποδηλώνει ότι ο αλγόριθμος διαχείρισης κλήσεων παρουσίασε ικανοποιητική επιχειρησιακή αποδοτικότητα και δεν δημιούργησε συμφόρηση στη λίστα requests. Η τεχνική αξιολόγηση του συστήματος περιλαμβάνεται συνοπτικά στον πίνακα 8.1.

**Πίνακας 8.1 Τεχνική Αξιολόγηση Συστήματος**

| Τεχνικό Κριτήριο        | Αξιολόγηση     |
|-------------------------|----------------|
| Σταθερότητα Λειτουργίας | Πολύ Υψηλή     |
| Διαχείριση Αιτημάτων    | Επιτυχής       |
| Χρονική Συνέπεια CPU    | Σταθερή        |
| Απόκριση Συστήματος     | Άμεση          |
| Διαχείριση Θυρών        | Ορθή           |
| Αξιοπιστία Προσομοίωσης | Υψηλή          |
| Καταγραφή Δεδομένων     | Πλήρης         |
| Runtime Stability       | Χωρίς Σφάλματα |

Εξίσου σημαντική θεωρήθηκε και η αξιολόγηση της modular αρχιτεκτονικής του κώδικα. Η αντικειμενοστραφής προσέγγιση που χρησιμοποιήθηκε επέτρεψε τη σαφή διαχωρισμένη οργάνωση των λειτουργικών μονάδων του συστήματος, διευκολύνοντας τόσο τη διαδικασία τεχνικής ανάλυσης όσο και τη δυνατότητα μελλοντικής επέκτασης του μοντέλου. Η ύπαρξη ανεξάρτητων συναρτήσεων για τη διαχείριση αιτημάτων, τη λογική κίνησης, τη λειτουργία θυρών και τη διαχείριση καταστάσεων προσέδωσε υψηλό βαθμό αναγνωσιμότητας και συντηρησιμότητας στο λογισμικό. Παράλληλα, η δυνατότητα εξαγωγής αποτελεσμάτων σε μορφές CSV και Excel ενίσχυσε τη χρηστικότητα του μοντέλου ως πλατφόρμας ανάλυσης και monitoring βιομηχανικών διεργασιών. Από πλευράς υπολογιστικής απόδοσης, το μοντέλο παρουσίασε εξαιρετικά χαμηλές απαιτήσεις σε πόρους επεξεργασίας και μνήμης. Η εκτέλεση της προσομοίωσης ολοκληρώθηκε χωρίς σημαντική επιβάρυνση του συστήματος, γεγονός που αποδεικνύει ότι η αναπτυγμένη λογική μπορεί να λειτουργήσει αποτελεσματικά ακόμη και σε περιορισμένα υπολογιστικά περιβάλλοντα. Η συγκεκριμένη ιδιότητα αποκτά ιδιαίτερη σημασία σε εφαρμογές embedded automation systems, όπου η αποδοτική αξιοποίηση πόρων αποτελεί κρίσιμο τεχνικό κριτήριο.

### 8.3 Περιορισμοί Συστήματος.

Παρά τη λειτουργική επιτυχία και τη σταθερότητα που παρουσίασε το σύστημα αυτοματισμού κατά τη διάρκεια της προσομοίωσης, η παρούσα υλοποίηση εμφανίζει συγκεκριμένους τεχνικούς περιορισμούς οι οποίοι σχετίζονται τόσο με τη φύση της προσομοίωσης όσο και με την απλουστευμένη λογική που χρησιμοποιήθηκε για την ανάπτυξη του μοντέλου. Η αναγνώριση των περιορισμών αυτών θεωρείται ιδιαίτερα σημαντική, καθώς επιτρέπει τη ρεαλιστική αξιολόγηση των δυνατοτήτων του συστήματος και προσδιορίζει τα σημεία στα οποία απαιτείται περαιτέρω βελτίωση προκειμένου το μοντέλο να προσεγγίσει περισσότερο τις απαιτήσεις πραγματικών βιομηχανικών εφαρμογών ανελκυστήρων. Ο σημαντικότερος περιορισμός του συστήματος σχετίζεται με τη χρήση απλουστευμένου αλγορίθμου FIFO scheduling για τη διαχείριση των αιτημάτων. Παρότι η συγκεκριμένη στρατηγική παρουσίασε υψηλή σταθερότητα και χαμηλή υπολογιστική πολυπλοκότητα, δεν προσφέρει δυνατότητες βελτιστοποίησης κυκλοφορίας ή δυναμικής αναδιάταξης προτεραιοτήτων. Σε πραγματικά βιομηχανικά συστήματα ανελκυστήρων χρησιμοποιούνται σύνθετοι αλγόριθμοι destination dispatching και traffic optimization, οι οποίοι λαμβάνουν υπόψη την κατεύθυνση κίνησης, τη χωρική κατανομή των αιτημάτων και τη συνολική κυκλοφοριακή φόρτιση του κτιρίου. Αντίθετα, η FIFO λογική οδηγεί συχνά σε αυξημένες μετακινήσεις μεταξύ απομακρυσμένων ορόφων, αυξάνοντας τον συνολικό χρόνο εξυπηρέτησης και περιορίζοντας την επιχειρησιακή αποδοτικότητα του συστήματος σε περιβάλλοντα υψηλής ζήτησης.

Επιπλέον, το παρόν μοντέλο δεν ενσωματώνει πλήρη προσομοίωση φυσικών παραμέτρων κίνησης. Η μεταβολή της θέσης του ανελκυστήρα πραγματοποιείται διακριτά ανά όροφο χωρίς αναπαράσταση επιτάχυνσης, επιβράδυνσης, αδρανειακών φαινομένων ή μεταβολής ταχύτητας κινήτρα. Σε πραγματικές εγκαταστάσεις, η κίνηση του θαλάμου ελέγχεται μέσω inverter systems και μετατροπέων συχνότητας, οι οποίοι εξασφαλίζουν ομαλή μετάβαση μεταξύ διαφορετικών ταχυτήτων λειτουργίας και περιορίζουν τις μηχανικές καταπονήσεις του συστήματος. Η απουσία τέτοιων μηχανισμών από την παρούσα προσομοίωση περιορίζει τον βαθμό ρεαλισμού του μοντέλου ως προς τη φυσική συμπεριφορά του ανελκυστήρα. Παρότι το σύστημα περιλαμβάνει βασική λογική alarm handling και interlocking, δεν υποστηρίζει προηγμένους μηχανισμούς προστασίας όπως αισθητήρες υπερφόρτωσης, ανίχνευση εμποδίων στις θύρες, emergency braking systems ή fault recovery algorithms. Σε πραγματικά βιομηχανικά συστήματα ανελκυστήρων, τα safety subsystems αποτελούν εξαιρετικά σύνθετες αρχιτεκτονικές οι οποίες λειτουργούν ανεξάρτητα από τη βασική CPU ελέγχου και διαθέτουν πιστοποιήσεις λειτουργικής ασφάλειας υψηλού επιπέδου (SIL – Safety Integrity Level). Η απουσία τέτοιων χαρακτηριστικών περιορίζει τη δυνατότητα αξιοποίησης του παρόντος μοντέλου σε εφαρμογές πραγματικού βιομηχανικού περιβάλλοντος.

**Πίνακας 8.2 Βασικοί Περιορισμοί Συστήματος**

| <b>Περιορισμός</b>                  | <b>Επίδραση</b>                 |
|-------------------------------------|---------------------------------|
| <b>FIFO Scheduling</b>              | Αυξημένες μετακινήσεις          |
| <b>Απουσία Dynamic Routing</b>      | Περιορισμένη βελτιστοποίηση     |
| <b>Απλοποιημένη Κίνηση</b>          | Μειωμένος φυσικός ρεαλισμός     |
| <b>Περιορισμένα Safety Systems</b>  | Μερική κάλυψη ασφαλείας         |
| <b>Απουσία Multi-Elevator Logic</b> | Περιορισμένη επεκτασιμότητα     |
| <b>Μη χρήση Real PLC Hardware</b>   | Καθαρά προσομοιωτικό περιβάλλον |
| <b>Απουσία Real-Time</b>            | Χωρίς βιομηχανικά               |

|               |            |
|---------------|------------|
| Communication | πρωτόκολλα |
|---------------|------------|

Το σύστημα υποστηρίζει αποκλειστικά λειτουργία ενός ανελκυστήρα και δεν περιλαμβάνει δυνατότητα συνεργατικής διαχείρισης πολλαπλών θαλάμων. Στα σύγχρονα πολυώροφα κτίρια χρησιμοποιούνται συνήθως πολυανελκυστήρια συστήματα, στα οποία η CPU λαμβάνει αποφάσεις σχετικά με την κατανομή των αιτημάτων μεταξύ διαφορετικών ανελκυστήρων ώστε να βελτιστοποιείται η συνολική ροή επιβατών. Η απουσία multi-elevator dispatching logic περιορίζει σημαντικά την επιχειρησιακή πολυπλοκότητα του μοντέλου και μειώνει την ικανότητά του να προσομοιώνει πραγματικές συνθήκες λειτουργίας μεγάλων εγκαταστάσεων. Επιπροσθέτως, η παρούσα υλοποίηση βασίστηκε αποκλειστικά σε λογισμικό προσομοίωσης χωρίς χρήση πραγματικού βιομηχανικού εξοπλισμού PLC ή φυσικών αισθητήρων πεδίου. Παρότι η λογική ελέγχου σχεδιάστηκε σύμφωνα με τις αρχές των PLC systems, το μοντέλο εκτελείται σε περιβάλλον γενικού σκοπού και όχι σε πραγματικό real-time industrial controller. Ως αποτέλεσμα, δεν αξιολογήθηκαν παράμετροι όπως καθυστερήσεις επικοινωνίας fieldbus, θόρυβος σημάτων εισόδου, hardware faults ή χρονικές αποκλίσεις φυσικών κυκλωμάτων ελέγχου.

Το μοντέλο δεν υποστηρίζει βιομηχανικά πρωτόκολλα επικοινωνίας όπως Modbus, Profibus, Profinet ή OPC-UA, τα οποία αποτελούν βασικά στοιχεία σύγχρονων συστημάτων βιομηχανικού αυτοματισμού. Η απουσία τέτοιων μηχανισμών περιορίζει τη δυνατότητα διασύνδεσης του συστήματος με εξωτερικές πλατφόρμες SCADA, industrial IoT εφαρμογές ή distributed automation architectures.

#### 8.4 Προτάσεις Βελτίωσης

Η ανάλυση της λειτουργίας του συστήματος αυτοματισμού και η αξιολόγηση των αποτελεσμάτων της προσομοίωσης κατέδειξαν ότι η αρχιτεκτονική που αναπτύχθηκε είναι ικανή να διαχειριστεί αποτελεσματικά τις βασικές λειτουργικές απαιτήσεις ενός ανελκυστήρα. Ωστόσο, η τεχνολογική εξέλιξη των βιομηχανικών συστημάτων αυτοματισμού και η αυξανόμενη ανάγκη για ενεργειακή αποδοτικότητα, απομακρυσμένη παρακολούθηση και ευφυή διαχείριση πόρων δημιουργούν σημαντικές δυνατότητες περαιτέρω εξέλιξης του συστήματος. Η ενσωμάτωση σύγχρονων τεχνολογιών αυτοματισμού, ψηφιακής διασύνδεσης και ανάλυσης δεδομένων μπορεί να βελτιώσει σημαντικά τόσο την αποδοτικότητα όσο και την αξιοπιστία της λειτουργίας ενός συστήματος ανελκυστήρα. Στο πλαίσιο αυτό, προτείνονται τέσσερις βασικές κατευθύνσεις τεχνολογικής βελτίωσης: η διασύνδεση με συστήματα SCADA, η υλοποίηση μηχανισμών παρακολούθησης μέσω IoT, η εφαρμογή στρατηγικών ενεργειακής βελτιστοποίησης και η ανάπτυξη πολυανελκυστικής αρχιτεκτονικής με χρήση δύο ανελκυστήρων.

Τα συστήματα SCADA αποτελούν βασικό στοιχείο της σύγχρονης βιομηχανικής αυτοματοποίησης, καθώς επιτρέπουν την εποπτεία και τον έλεγχο σύνθετων εγκαταστάσεων μέσω κεντρικών σταθμών παρακολούθησης. Στην περίπτωση ενός συστήματος ανελκυστήρα, η σύνδεση με ένα SCADA σύστημα θα επέτρεπε την απεικόνιση της κατάστασης λειτουργίας του ανελκυστήρα σε πραγματικό χρόνο, συμπεριλαμβανομένων μεταβλητών όπως η θέση του θαλάμου, η κατεύθυνση κίνησης, η κατάσταση των θυρών, καθώς και τα ενεργά αιτήματα μετακίνησης. Επιπλέον, ένα SCADA σύστημα θα μπορούσε να καταγράφει συνεχώς τα δεδομένα λειτουργίας του ανελκυστήρα σε ιστορικές βάσεις δεδομένων, επιτρέποντας τη διεξαγωγή αναλυτικών μελετών για τη βελτιστοποίηση της λειτουργίας του συστήματος.

Η ενσωμάτωση ενός τέτοιου συστήματος θα μπορούσε να υλοποιηθεί μέσω βιομηχανικών πρωτοκόλλων επικοινωνίας όπως Modbus TCP/IP, OPC-UA ή Profibus, τα οποία χρησιμοποιούνται ευρέως σε συστήματα PLC. Μέσω αυτής της διασύνδεσης, το PLC θα μπορούσε να μεταδίδει σε πραγματικό χρόνο δεδομένα λειτουργίας προς τον κεντρικό σταθμό SCADA, όπου θα εμφανίζονται σε μορφή γραφικών διεπαφών (HMI dash boards). Η χρήση τέτοιων συστημάτων επιτρέπει στους χειριστές να εντοπίζουν άμεσα πιθανές

δυσλειτουργίες και να παρεμβαίνουν έγκαιρα πριν εμφανιστούν σοβαρά λειτουργικά προβλήματα. Μια δεύτερη σημαντική κατεύθυνση βελτίωσης αφορά την αξιοποίηση τεχνολογιών Internet of Things (IoT) για την απομακρυσμένη παρακολούθηση της λειτουργίας του ανελκυστήρα. Η ανάπτυξη ενός IoT συστήματος επιτρέπει τη συλλογή δεδομένων λειτουργίας από αισθητήρες που βρίσκονται εγκατεστημένοι στο σύστημα ανελκυστήρα και τη μετάδοσή τους σε διαδικτυακές πλατφόρμες επεξεργασίας δεδομένων. Μέσω αυτής της διαδικασίας καθίσταται δυνατή η συνεχής παρακολούθηση κρίσιμων παραμέτρων όπως η θερμοκρασία του κινητήρα, η κατανάλωση ηλεκτρικής ενέργειας, ο αριθμός κύκλων λειτουργίας των θυρών και η συχνότητα μετακινήσεων μεταξύ των ορόφων.

Η αξιοποίηση IoT τεχνολογιών μπορεί να συμβάλει σημαντικά στη μετάβαση από την παραδοσιακή προληπτική συντήρηση σε ένα μοντέλο predictive maintenance. Στο μοντέλο αυτό, η ανάλυση των δεδομένων λειτουργίας επιτρέπει την πρόβλεψη πιθανών αστοχιών πριν αυτές εμφανιστούν στην πράξη. Για παράδειγμα, η συνεχής παρακολούθηση της κατανάλωσης ρεύματος του κινητήρα μπορεί να αποκαλύψει ανωμαλίες στη λειτουργία του συστήματος μετάδοσης κίνησης. Αντίστοιχα, η καταγραφή του χρόνου λειτουργίας των μηχανισμών θυρών μπορεί να υποδείξει την ανάγκη αντικατάστασης εξαρτημάτων πριν εμφανιστούν λειτουργικές δυσλειτουργίες. Στα σύγχρονα κτίρια, τα συστήματα ανελκυστήρων αποτελούν σημαντικό παράγοντα κατανάλωσης ηλεκτρικής ενέργειας. Μελέτες έχουν δείξει ότι σε μεγάλα κτίρια γραφείων η λειτουργία των ανελκυστήρων μπορεί να αντιπροσωπεύει έως και 5–10% της συνολικής κατανάλωσης ηλεκτρικής ενέργειας του κτιρίου. Για τον λόγο αυτό, η ανάπτυξη αλγορίθμων ενεργειακής βελτιστοποίησης μπορεί να συμβάλει σημαντικά στη μείωση της κατανάλωσης ενέργειας. Μία πιθανή στρατηγική ενεργειακής βελτιστοποίησης είναι η εφαρμογή αλγορίθμων που μειώνουν τον αριθμό των περιττών μετακινήσεων του ανελκυστήρα. Μέσω της ανάλυσης των ιστορικών δεδομένων χρήσης του ανελκυστήρα μπορεί να εντοπιστούν χρονικές περίοδοι χαμηλής ζήτησης, κατά τις οποίες το σύστημα θα μπορούσε να λειτουργεί σε κατάσταση εξοικονόμησης ενέργειας. Επιπλέον, η χρήση κινητήρων μεταβλητής συχνότητας (Variable Frequency Drives – VFD) μπορεί να μειώσει σημαντικά την κατανάλωση ενέργειας κατά τη διάρκεια της επιτάχυνσης και της επιβράδυνσης του ανελκυστήρα.

Μια ακόμη σημαντική τεχνολογική επέκταση αφορά την ανάπτυξη συστημάτων πολλαπλών ανελκυστήρων, τα οποία χρησιμοποιούνται ευρέως σε μεγάλα κτίρια με υψηλή κυκλοφορία χρηστών. Στην περίπτωση ενός συστήματος με δύο ανελκυστήρες, απαιτείται ένας πιο σύνθετος αλγόριθμος ελέγχου ο οποίος θα κατανέμει τα αιτήματα μετακίνησης μεταξύ των δύο θαλάμων. Η διαδικασία αυτή είναι γνωστή ως elevator dispatching και αποτελεί ένα ιδιαίτερα ενεργό πεδίο έρευνας στον τομέα των συστημάτων αυτοματισμού. Η χρήση δύο ανελκυστήρων μπορεί να μειώσει σημαντικά τον μέσο χρόνο αναμονής των χρηστών, ιδιαίτερα σε κτίρια με μεγάλο αριθμό ορόφων. Για παράδειγμα, σε ένα κτίριο έξι ορόφων όπως αυτό που εξετάστηκε στην παρούσα προσομοίωση, η χρήση ενός δεύτερου ανελκυστήρα θα μπορούσε να μειώσει τον χρόνο εξυπηρέτησης των αιτημάτων κατά 30–40%, ανάλογα με την κατανομή των κλήσεων. Ο αλγόριθμος κατανομής αιτημάτων θα μπορούσε να βασίζεται σε κριτήρια όπως η απόσταση του ανελκυστήρα από τον ζητούμενο όροφο, η τρέχουσα κατεύθυνση κίνησης και ο αριθμός των ενεργών αιτημάτων που έχει αναλάβει κάθε ανελκυστήρας.

## Βιβλιογραφία

- [1]Jiang, H. (2023). Research on electrical automation control system based on PLC technology. *Academic Journal of Engineering and Technology Science*, 6(3). <https://doi.org/10.25236/ajets.2023.060305>
- [2]Kan, W. (2025). Design of a six-lift, ten-level elevator system based on PLC and WinCC. *Theoretical and Natural Science*. <https://doi.org/10.54254/2753-8818/2025.22059>
- [3]Zyubin, V., Rozov, A., Anureev, I., Garanina, N., &Vyatkin, V. (2022). poST: A process-oriented extension of the IEC 61131-3 structured text language. *IEEE Access*, 10, 35238–35250. <https://doi.org/10.1109/access.2022.3157601>
- [4]Chang, W., Zhang, Q., Xia, H., & Yang, Y. (2021). Construction of virtual simulation teaching platform for elevator control. *Journal of Physics: Conference Series*, 1848(1), 012120. <https://doi.org/10.1088/1742-6596/1848/1/012120>
- [5]Huo, Q., & Xie, Y. (2023). Design of Elevator Control System Based on PLC. *Journal of Engineering Mechanics and Machinery*, 8(4), 44–51. <https://doi.org/10.23977/jemm.2023.080407>
- [6]Baranidharan, M., & Singh, R. R. (2024). SCADA based common bus regenerative control of PMSM drive for industrial application. *Advanced Control for Applications: Engineering and Industrial Systems*, 6. <https://doi.org/10.1002/adc2.199>
- [7].Amin, A. A., Alvi, N., Irfan, M., Abdullah, M., & Manzoor, H. (2025). Design of a low-cost efficient IoT based SCADA system for automating different textile machines and controllers for central monitoring. *The Journal of Engineering*. <https://doi.org/10.1049/tje2.70089>
- [8].Huang, H., Zhu, Q., Zhu, X., & Zhang, J. (2023). An adaptive, data-driven stacking ensemble learning framework for the short-term forecasting of renewable energy generation. *Energies*, 16(4), 1963. <https://doi.org/10.3390/en16041963>
- [9]Trabucco, D., Perrucci, G., Giacomello, E., & Costa, M. (2024). Needs of deaf people using elevators: Identification of accessibility and safety requirements. *Buildings*, 14(10), 3300. <https://doi.org/10.3390/buildings14103300>
- [10] Shi, H., Zhu, Q., Bo, J., Shi, J., & Zhao, T. (2025). A generic method for PLC control system design based on digital twin. *PLOS One*, 20, e0327592. <https://doi.org/10.1371/journal.pone.0327592>
- [11]Babu, M. K., &Suthari, Y. (2023). Secure and intelligent PLC systems: Integrating artificial intelligence for enhanced industrial control and data privacy. *Computer Fraud and Security*. <https://doi.org/10.52710/cfs.627>
- [12] Li, H., Wei, Y., Zhong, J., Chen, Y., Qian, S., & Yue, W. (2024). Teaching practice of PLC programming in elevator virtual simulation experiment platform. *Contemporary Education and Teaching Research*. <https://doi.org/10.61360/bonicetr242015830204>
- [13] McCarthy, D., McMorro, D., O’Dowd, N., McCarthy, C., & Hinchy, E. (2022). A model-based approach to automated validation and generation of PLC code for manufacturing equipment in regulated environments. *Applied Sciences*, 12(15), 7506. <https://doi.org/10.3390/app12157506>
- [14] Kale, S., & Rane, D. (2023). Implementation of FATEK PLC based three floor elevator system. *International Journal for Multidisciplinary Research*, 5(3). <https://doi.org/10.36948/ijfmr.2023.v05i03.3348>
- [15] He, H., Long, Y., & Yu, W. (2021). Design and simulation of elevator emergency system based on TIA Portal V15.1. *Journal of Physics: Conference Series*, 1802(4), 042089. <https://doi.org/10.1088/1742-6596/1802/4/042089>

- [16] Zhao, B. (2021). The design and research of electric automation control system based on PLC. *International Journal of Frontiers in Sociology*, 3(11). <https://doi.org/10.25236/ijfs.2021.031109>
- [17] Zhang, W. (2024). Design of a PLC system simulator and application to teaching programmable logic controller course online. *Proceedings of the ASEE Annual Conference & Exposition*. <https://doi.org/10.18260/1-2--47125>
- [18] Suparmono, Sitepu, T., Salman, A., Pardede, S., Cholish, Abdullah, & Nofri, I. (2025). Control system design for 3 storey elevator using PLC OPTA FINDER. *International Journal of Research in Vocational Studies*, 4(4). <https://doi.org/10.53893/ijrvocas.v4i4.384>
- [19] Iqbal, N., Lawal, A., & Zerguine, A. (2021). Mitigating wireless channel impairments in seismic data transmission using deep neural networks. *Sensors*, 21(18), 6105. <https://doi.org/10.3390/s21186105>
- [20] Michael, I. A. (2022). Design of lift group control systems based on PLC. *Journal of Engineering Research and Reports*. <https://doi.org/10.9734/jerr/2022/v22i317528>
- [21] Suparmono, T., Sitepu, T., Salman, A., Pardede, S., Cholish, A., Abdullah, I. N. (2024). Control System Design for 3 Storey Elevator Using PLC OPTA FINDER. *International Journal of Research in Vocational Studies (IJRVOCAS)*, 4(4), 73–78. <https://doi.org/10.53893/ijrvocas.v4i4.384>
- [22] Bujnowski, S., Marciniak, B., Lutowski, Z., Flizikowski, A., & Oyerinde, O. O. (2021). Evaluation of transmission properties of networks described with reference graphs using unevenness coefficients. *Electronics*, 10(14), 1684. <https://doi.org/10.3390/electronics10141684>
- [23] Ofoedu, A. T., Ozor, J. E., Sofoluwe, O., & Jambol, D. D. (2024). A SCADA-integrated framework for real-time production monitoring and operational intelligence in FPSO units. *International Journal of Advanced Multidisciplinary Research and Studies*. <https://doi.org/10.62225/2583049x.2024.4.6.4450>
- [24] Morales-Sandoval, M., De-La-Parra-Aguirre, R., Galeana-Zapién, H., & Galaviz-Mosqueda, A. (2021). A three-tier approach for lightweight data security of body area networks in e-health applications. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2021.3123456>

# Παράρτημα

## Παράρτημα Α: Πλήρης Κώδικας PLC

```
!pip install openpyxl graphviz -q

import time
import random
import pandas as pd
import matplotlib.pyplot as plt
import networkx as nx
import graphviz

import ipywidgets as widgets

from IPython.display import display, clear_output

FLOORS = 6

SIM_STEPS = 40

PLC_INPUTS = {

    "I0.0": "Floor_0_Button",
    "I0.1": "Floor_1_Button",
    "I0.2": "Floor_2_Button",
    "I0.3": "Floor_3_Button",
    "I0.4": "Floor_4_Button",
    "I0.5": "Floor_5_Button"

}

PLC_OUTPUTS = {

    "Q0.0": "Motor_UP",
    "Q0.1": "Motor_DOWN",
    "Q0.2": "Door_OPEN",
    "Q0.3": "Door_CLOSE",
    "Q0.4": "Alarm"

}

class Elevator:

    def __init__(self, floors):

self.floors = floors

self.current_floor = 0

self.direction = "IDLE"

self.requests = []

self.door = "CLOSED"
```

```

self.alarm = False

self.time = 0

self.history_floor = []

self.history_time = []

self.history_direction = []

self.history_requests = []

# -----

    def add_request(self, floor):

        if floor not in self.requests:

self.requests.append(floor)

print(f"📞 Request registered -> Floor {floor}")

# -----

    def choose_direction(self):

        if not self.requests:

self.direction = "IDLE"

            return

            target = self.requests[0]

            if target >self.current_floor:

self.direction = "UP"

elif target <self.current_floor:

self.direction = "DOWN"

            else:

self.direction = "STOP"

# -----

    def move(self):

        if self.direction == "UP":

self.current_floor += 1

elif self.direction == "DOWN":

```

```

self.current_floor -= 1

# -----

    def operate_doors(self):
print("🚪 Door opening...")
self.door = "OPEN"
time.sleep(0.5)
print("🚪 Door closing...")
self.door = "CLOSED"

# -----

    def step(self):
self.choose_direction()

        if self.direction in ["UP", "DOWN"]:
self.move()
elif self.direction == "STOP":
print(f"🛑 Stopping at floor {self.current_floor}")
self.operate_doors()

            if self.current_floor in self.requests:
self.requests.remove(self.current_floor)
self.history_floor.append(self.current_floor)
self.history_time.append(self.time)
self.history_direction.append(self.direction)
self.history_requests.append(len(self.requests))

self.time += 1

e = Elevator(FLOORS)

status_output = widgets.Output()
building_output = widgets.Output()
log_output = widgets.Output()

def update_ui():

```

```

        with status_output:
clear_output(wait=True)

print("=====")

print("🏢 ELEVATOR STATUS")

print("=====")

print("📄 Current Floor :", e.current_floor)

print("📍 Direction      :", e.direction)

print("🚪 Door State      :", e.door)

print("📄 Active Calls   :", e.requests)

        with building_output:
clear_output(wait=True)

print("\n🏢 BUILDING STATUS\n")

        for f in reversed(range(e.floors)):

            if f == e.current_floor:

                print(f"{f} | 🟩 [ELEVATOR]")

            elif f in e.requests:

                print(f"{f} | 🟨 [REQUEST]")

            else:

                print(f"{f} | 🟩")

def request_floor(floor):

e.add_request(floor)

update_ui()

hall_buttons = []

for floor in range(FLOORS):

    row = []

    if floor < FLOORS - 1:

```

```

        up = widgets.Button(
            description=f"⬆️ UP {floor}",
            button_style='success',
            layout=widgets.Layout(width='120px')
        )
        up.on_click(lambda b, f=floor: request_floor(f))
        row.append(up)
        if floor > 0:
            down = widgets.Button(
                description=f"⬇️ DOWN {floor}",
                button_style='warning',
                layout=widgets.Layout(width='120px')
            )
            down.on_click(lambda b, f=floor: request_floor(f))
            row.append(down)
        hall_buttons.append(widgets.HBox(row))

    cabin_buttons = []
    for floor in range(FLOORS):
        btn = widgets.Button(
            description=f"Go {floor}",
            button_style='info',
            layout=widgets.Layout(width='100px')
        )
        btn.on_click(lambda b, f=floor: request_floor(f))
        cabin_buttons.append(btn)

def run_simulation(b):

```

```

        with log_output:
clear_output()

        for step in range(SIM_STEPS):
            with log_output:
                print("\n=====")

print("🕒 TIME STEP", step)

            if random.random() < 0.15:
random_floor = random.randint(0, FLOORS - 1)
e.add_request(random_floor)

e.step()

update_ui()

time.sleep(0.5)

            with log_output:
print("\n☑ Simulation Finished")

generate_reports()

def generate_reports():

    # -----
    # DATAFRAME
    # -----

df = pd.DataFrame({

        "time": e.history_time,
        "floor": e.history_floor,
        "direction": e.history_direction,
"pending_requests": e.history_requests

    })

df.to_csv("simulation_results.csv", index=False)
df.to_excel("simulation_results.xlsx", index=False)

print("\n FILES EXPORTED")

print(" simulation_results.csv")

print(" simulation_results.xlsx")

```

```

#-----
# GRAPH 1
#-----

plt.figure(figsize=(10,5))

plt.plot(

e.history_time,
e.history_floor,
    marker='o'

    )

plt.xlabel("Time")
plt.ylabel("Floor")
plt.title("Elevator Position vs Time")
plt.grid()
plt.show()

#-----
# GRAPH 2
#-----

    floors = list(range(FLOORS))
    visits = [
e.history_floor.count(f)
        for f in floors
    ]

plt.figure(figsize=(8,5))
plt.bar(floors, visits)
plt.xlabel("Floor")
plt.ylabel("Visits")
plt.title("Elevator Visits per Floor")
plt.show()

#-----
# GRAPH 3
#-----

plt.figure(figsize=(10,5))

plt.plot(

```

```

e.history_time,
e.history_requests,
    color='red'

    )

plt.xlabel("Time")

plt.ylabel("Pending Requests")

plt.title("Pending Requests vs Time")

plt.grid()

plt.show()

sensors = list(PLC_INPUTS.values())

actuators = list(PLC_OUTPUTS.values())

G = nx.DiGraph()

for s in sensors:
G.add_edge(s, "INPUT_MODULE")
G.add_edge("INPUT_MODULE", "PLC_CPU")
G.add_edge("PLC_CPU", "OUTPUT_MODULE")

for a in actuators:
G.add_edge("OUTPUT_MODULE", a)

pos = {}

for i, s in enumerate(sensors):
pos[s] = (-2, 5 - i)

pos["INPUT_MODULE"] = (-1, 2.5)
pos["PLC_CPU"] = (0, 2.5)
pos["OUTPUT_MODULE"] = (1, 2.5)

for i, a in enumerate(actuators):
pos[a] = (2, 4 - i)

plt.figure(figsize=(12,6))

nx.draw(

    G,

```

```

        pos,
with_labels=True,
node_size=3500,
node_color="lightblue",
    arrows=True

    )

plt.title("Industrial PLC Architecture")

plt.show()

```

```

G2 = nx.DiGraph()

transitions = [

    ("Idle", "Moving_UP"),
    ("Idle", "Moving_DOWN"),

    ("Moving_UP", "Door_Opening"),
    ("Moving_DOWN", "Door_Opening"),

    ("Door_Opening", "Door_Open"),
    ("Door_Open", "Door_Closing"),
    ("Door_Closing", "Idle"),

    ("Moving_UP", "Alarm"),
    ("Moving_DOWN", "Alarm"),

    ("Alarm", "Idle")

]

G2.add_edges_from(transitions)

pos = nx.spring_layout(G2)

plt.figure(figsize=(8,6))

nx.draw(

    G2,
    pos,
with_labels=True,
node_color="lightgreen",
node_size=3000,
    arrows=True

    )

plt.title("Elevator State Machine")

plt.show()

```

```

def reset_system(b):
    global e
        e = Elevator(FLOORS)
    update_ui()
        with log_output:
    clear_output()
    print("🔄 SYSTEM RESET")

start_button = widgets.Button(
    description="▶ START SIMULATION",
    button_style='primary',
    layout=widgets.Layout(width='250px', height='50px')
)

reset_button = widgets.Button(
    description="🔄 RESET",
    button_style='danger',
    layout=widgets.Layout(width='150px', height='50px')
)

start_button.on_click(run_simulation)
reset_button.on_click(reset_system)

update_ui()

display(widgets.HTML("<h1>🏢 COMPLETE ELEVATOR PLC SYSTEM</h1>"))
display(status_output)
display(building_output)

display(widgets.HTML("<h2>📞 Hall Call Buttons</h2>"))
for row in hall_buttons:
    display(row)

```

```
display(widgets.HTML("<h2> 🚪 Cabin Buttons</h2>"))
display(widgets.HBox(cabin_buttons))
display(widgets.HBox([start_button, reset_button]))
display(widgets.HTML("<h2> 📄 SYSTEM LOG</h2>"))
display(log_output)
```