



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

Τμ. ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

Π.Μ.Σ. «ΜΗ ΕΠΑΝΔΡΩΜΕΝΑ ΑΥΤΟΝΟΜΑ ΚΑΙ

ΤΗΛΕΚΑΤΕΥΘΥΝΟΜΕΝΑ ΣΥΣΤΗΜΑΤΑ»

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«ΜΗ ΕΠΑΝΔΡΩΜΕΝΑ ΑΥΤΟΝΟΜΑ ΚΑΙ ΤΗΛΕΚΑΤΕΥΘΥΝΟΜΕΝΑ

ΣΥΣΤΗΜΑΤΑ»

ΤΙΤΛΟΣ

ΔΙΑΧΕΙΡΙΣΗ ΣΜΗΝΟΥΣ ΜΕ ΑΠΟΜΑΚΡΥΣΜΕΝΟ ΕΛΕΓΧΟ

ΤΙΤΛΟΣ ΑΓΓΛΙΚΑ

SWARM COORDINATION WITH REMOTE CONTROL

Όνοματεπώνυμο Φοιτητή:

ΑΪΒΑΤΙΔΗΣ ΙΟΡΔΑΝΗΣ

Όνοματεπώνυμο Υπεύθυνου Καθηγητή:

ΑΒΡΑΑΜ ΧΑΤΖΟΠΟΥΛΟΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΟΚΤΩΒΡΙΟΣ 2024

Μέλη Εξεταστικής Επιτροπής

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Αϊβατίδης Ιορδάνης του Ιωάννη, με αριθμό μητρώου 8096625 φοιτητής του Προγράμματος Μεταπτυχιακών Σπουδών «Μη Επανδρωμένα Αυτόνομα και Τηλεκατευθυνόμενα Συστήματα» του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής της Σχολής Μηχανικών Πανεπιστημίου Δυτικής Αττικής, δηλώνω υπεύθυνα ότι: «Είμαι συγγραφέας αυτής της μεταπτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου».

Ο δηλών

Ημερομηνία

29/10/2024



ΠΕΡΙΛΗΨΗ

Το παρόν έργο διερευνά τη διαμόρφωση και τον έλεγχο ενός σμήνους drones Tello EDU, εστιάζοντας σε εκπαιδευτικούς και πειραματικούς σκοπούς. Με χρήση της γλώσσας Python και της βιβλιοθήκης djitellopy, πραγματοποιήθηκε μια σειρά πειραμάτων για την επαλήθευση συγχρονισμένων κινήσεων των drones. Αρχικά, το έργο περιλάμβανε την καθιέρωση μιας αξιόπιστης επικοινωνίας μέσω δικτύου Wi-Fi με τα drones, τη ρύθμισή τους σε λειτουργία "σταθμού" και την εκτέλεση διαφόρων εντολών πτήσης. Εξετάστηκαν διάφορες εφαρμογές και κινήσεις, από απλή αιώρηση έως πιο περίπλοκους σχηματισμούς, όπως η πτήση σε σχηματισμό "κουτιού" και ακροβατικοί ελιγμοί όπως ανατροπές (flips). Ο τελικός στόχος ήταν η παροχή ενός ευέλικτου πλαισίου για τον έλεγχο πολλαπλών drones ταυτόχρονα και η ανάλυση της συμπεριφοράς τους σε περιβάλλοντα σμήνους, προσφέροντας σημαντικές εκπαιδευτικές δυνατότητες και γνώσεις στον τομέα της ρομποτικής σμήνους.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Tello EDU, Έλεγχος Σμήνους, Έλεγχος Πτήσης, Συγχρονισμένες Κινήσεις, Σχηματισμός Drone

ABSTRACT

This project explores the configuration and control of a swarm of Tello EDU drones, focusing on educational and experimental purposes. Using the Python language and the djitellopy library, a series of experiments were conducted to verify the synchronized movements of the drones. Initially, the project involved establishing reliable communication via a Wi-Fi network with the drones, setting them up in "station mode," and executing various flight commands. Various applications and movements were examined, from simple hovering to more complex formations, such as flying in a "box" formation and performing acrobatic maneuvers like flips. The ultimate goal was to provide a flexible framework for controlling multiple drones simultaneously and analyzing their behavior in swarm environments, offering significant educational opportunities and insights into the field of swarm robotics.

KEYWORDS

Tello EDU, Swarm Control, Flight Control, Synchronized Movements, Drone Formation

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ.....	iv
ABSTRACT	v
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	vi
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	viii
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	ix
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ.....	x
ΕΙΣΑΓΩΓΗ.....	1
Μεθοδολογία	3
ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	5
Νοημοσύνη Σμήνους	5
Ορισμός και χαρακτηριστικά ρομποτικού σμήνους.....	6
Χαρακτηριστικά των σμηνών της φύσης	6
Πλεονεκτήματα της ρομποτικής σμήνους.....	7
Πεδίο εφαρμογών της ρομποτικής σμήνους.....	11
Σύστημα ρομποτικής σμήνους στην πραγματική ζωή.....	12
Γενικό μοντέλο ρομποτικής σμήνους.....	13
Ενότητα ανταλλαγής πληροφοριών.....	14
Μέθοδοι Μοντελοποίησης για τη Ρομποτική Σμήνους.....	16
Packet sender.....	17
ΛΕΙΤΟΥΡΓΙΚΗ ΑΝΑΛΥΣΗ ΚΑΙ ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ TELLO EDU ΓΙΑ ΤΗΝ ΟΛΟΚΛΗΡΩΜΕΝΗ ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ	19
Λειτουργικά Μέρη Κατασκευής	20
Βασικό Περιβάλλον πριν από τον προγραμματισμό	22
Πίνακες Αποστολής – Mission Pads	23
Εφαρμογή	24
Εντολές που μπορούν να σταλούν στο Tello EDU Drone	26
Καταστάσεις λειτουργίας (Mode) Tello Drone.....	27
Ρύθμιση Σμήνους (Swarm Setup).....	28
Οικονομικός προϋπολογισμός.....	29
ΑΝΑΛΥΣΗ ΚΑΙ ΣΥΓΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ (SOFTWARE)	30
Διαδικασία ασύρματης σύνδεσης Tello drone 1/3	30

Διαδικασία ασύρματης σύνδεσης Tello drone 2/3	31
Διαδικασία ασύρματης σύνδεσης Tello drone 3/3	32
Κώδικας.....	33
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 1/8.....	34
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 2/8.....	34
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 3/8.....	35
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 4/8.....	35
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 5/8.....	37
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 6/8.....	38
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 7/8.....	39
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 8/8.....	39
1 ^ο Πείραμα.....	40
2 ^ο Πείραμα	44
3 ^ο Πείραμα.....	49
ΑΠΟΤΕΛΕΣΜΑΤΑ	54
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	55
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	56

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Περιγραφή του drone.....	20
Πίνακας 2: Λεπτομέρειες στη λειτουργία	21
Πίνακας 3: Λεπτομέρειες μπαταρίας.....	21
Πίνακας 4: Λεπτομέρειες κάμερας.....	22
Πίνακας 5: Εντολές Ελέγχου μέσω Python (Cavadas, 2024).....	26

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1 – Comparison of swarm robotics and other systems (Ying TAN*, 2013).....	10
Εικόνα 2 - General model of swarm robotics (Ying TAN*, 2013).....	14
Εικόνα 11 – GUI Packet Sender (Packet Sender documentation, 2024).....	18
Εικόνα 3 -Tello EDU.....	19
Εικόνα 4 – Τεχνικά χαρακτηριστικά (Cavadas, 2024).....	20
Εικόνα 5 – Mission Pads	23
Εικόνα 6 – Detection Range (Cavadas, 2024).....	24
Εικόνα 7 – Εφαρμογή για τον έλεγχο του drone μέσω κινητού- Tello app	25
Εικόνα 8 - Flight modes	25
Εικόνα 9 – Καταστάσεις λειτουργίας Tello Drone.....	28
Εικόνα 10 – Drones in Station mode.....	29
Εικόνα 12 – Επιβεβαίωση σύνδεσης Tello drone με το laptop(Wi-Fi).....	30
Εικόνα 13 – Εντολή επικοινωνίας μέσω Packet Sender.....	31
Εικόνα 14 - Εντολή για τον έλεγχο της μπαταρίας μέσω Packet Sender	32
Εικόνα 15 – Αλλαγή "Wireless Router" σε "Access Point"	34
Εικόνα 16 – Αλλαγή από Dynamic σε Static IP.....	34
Εικόνα 17 – Εύρος IP διευθύνσεων.....	35
Εικόνα 18 – Εντολή σύνδεσης drone με το Access point.....	36
Εικόνα 19 – Αναζήτηση νέας IP	37
Εικόνα 20 – Επιβεβαίωση επικοινωνίας με την καινούργια IP.....	38
Εικόνα 21 – Επιβεβαίωση ποσοστού μπαταρίας σε λειτουργία σμήνους.....	39
Εικόνα 22 – Αιώρηση στο 1 μέτρο σε λειτουργία σμήνους.....	44
Εικόνα 23 – Πτήση σε ‘κουτί’ και αλλαγή κατεύθυνσης (ΦΑΣΗ 1)	52
Εικόνα 24 – Πτήση και flip (ΦΑΣΗ 2)	53

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

Συντομογραφία (Abbreviation)	Πλήρης Όρος στα Αγγλικά (Full Term in English)	Περιγραφή (Description)
AP	Access Point	Σημείο Πρόσβασης: Συσκευή δικτύου που επιτρέπει στα drones να συνδεθούν μέσω Wi-Fi.
DHCP	Dynamic Host Configuration Protocol	Πρωτόκολλο Δυναμικής Ρύθμισης Κεντρικού Υπολογιστή: Πρωτόκολλο για την αυτόματη ανάθεση διευθύνσεων IP.
IP	Internet Protocol	Πρωτόκολλο Διαδικτύου: Πρωτόκολλο για τη διεύθυνση και την επικοινωνία σε δίκτυο.
UDP	User Datagram Protocol	Πρωτόκολλο Χρήστη Datagram: Πρωτόκολλο δικτύωσης που χρησιμοποιείται για την αποστολή και λήψη δεδομένων μεταξύ του υπολογιστή και των drones.
Tello EDU	Tello Educational Drone	Εκπαιδευτικό Drone Tello: Μοντέλο drone της Ryze Tech σχεδιασμένο για εκπαιδευτικούς σκοπούς και προγραμματισμό.
SSID	Service Set Identifier	Ταυτότητα Συνόλου Υπηρεσιών: Το όνομα ενός δικτύου Wi-Fi.
Wi-Fi	Wireless Fidelity	Ασύρματη Πιστότητα: Τεχνολογία για ασύρματη επικοινωνία σε δίκτυα υπολογιστών.
RC	Remote Control	Τηλεχειριστήριο: Αποστολή εντολών κίνησης και ελέγχου στα drones.
ASCII	American Standard Code for Information Interchange	Αμερικανικός Τυποποιημένος Κώδικας για την Ανταλλαγή Πληροφοριών: Πρότυπο κωδικοποίησης χαρακτήρων που

		χρησιμοποιείται για την αποστολή εντολών κειμένου στα drones.
HEX	Hexadecimal	Δεκαεξαδικό: Μορφή αναπαράστασης αριθμών που χρησιμοποιείται για την αναγνώριση εντολών στα drones.
CW	Clockwise	Δεξιόστροφα: Κατεύθυνση περιστροφής των drones κατά τη διάρκεια ελιγμών.

ΕΙΣΑΓΩΓΗ

Η ανάπτυξη και ο συντονισμός σμήνους drones αποτελεί ένα αναδυόμενο πεδίο έρευνας με πολλές εφαρμογές τόσο στον εκπαιδευτικό όσο και στον βιομηχανικό τομέα. Τα drones έχουν γίνει ευρέως γνωστά για την ευελιξία, την απλότητα και τις δυνατότητες τους σε διάφορους τομείς, όπως η λήψη εικόνων, η επιτήρηση, οι αποστολές διάσωσης και πολλές άλλες εφαρμογές. Ωστόσο, ο συντονισμός πολλαπλών drones που δρουν ως σμήνος (swarm) προσφέρει νέες ευκαιρίες και προκλήσεις, ειδικά όσον αφορά τη συνεργασία μεταξύ τους και τον συνδυασμό των κινήσεων και των ενεργειών τους.

Στο πλαίσιο αυτής της έρευνας, διερευνάται η διαμόρφωση και ο έλεγχος ενός σμήνους drones τύπου Tello EDU για εκπαιδευτικούς και πειραματικούς σκοπούς. Τα Tello EDU drones είναι μικρά και εύχρηστα, καθιστώντας τα ιδανικά για εφαρμογές σε περιβάλλοντα που απαιτούν προγραμματισμό και συγχρονισμό. Με τη χρήση της γλώσσας προγραμματισμού Python και της βιβλιοθήκης djitellory, πραγματοποιήθηκαν διάφορα πειράματα, με στόχο την επαλήθευση συγχρονισμένων κινήσεων μεταξύ των drones. Αρχικά, το έργο επικεντρώθηκε στη δημιουργία ενός αξιόπιστου δικτύου επικοινωνίας μέσω Wi-Fi, τη ρύθμιση των drones σε λειτουργία "σταθμού" και την εκτέλεση διαφόρων εντολών πτήσης. Κατά τη διάρκεια των πειραμάτων εξετάστηκαν διάφορα είδη κινήσεων, ξεκινώντας από απλή αιώρηση και φτάνοντας μέχρι πιο σύνθετες δομές, όπως η πτήση σε σχηματισμό "κουτιού" και οι ακροβατικοί ελιγμοί, όπως ανατροπές (flips).

Ο τελικός στόχος αυτής της έρευνας ήταν να παρασχεθεί ένα ευέλικτο και αποτελεσματικό πλαίσιο για τον έλεγχο πολλαπλών drones ταυτόχρονα, επιτρέποντας την ανάλυση της συμπεριφοράς τους σε περιβάλλον σμήνους. Αυτό το πλαίσιο θα μπορούσε να αποτελέσει πολύτιμο εργαλείο για την προώθηση των εκπαιδευτικών δεξιοτήτων στον τομέα της ρομποτικής σμήνους και της τεχνητής νοημοσύνης, καθώς και για την ενίσχυση της κατανόησης σχετικά με τη συλλογική νοημοσύνη και τον συντονισμό πολλαπλών πρακτόρων (agents).

Η ρομποτική σμήνους αντιπροσωπεύει έναν δυναμικά αναπτυσσόμενο τομέα έρευνας, εμπνευσμένο από τη συνεργατική συμπεριφορά που παρατηρείται σε φυσικά σμήνη, όπως αυτά των μυρμηγκιών, των μελισσών και των πουλιών. Ο κύριος στόχος του παρόντος έργου ήταν η μελέτη και η ανάπτυξη μεθόδων για τον συγχρονισμένο έλεγχο ενός σμήνους drones, αντιμετωπίζοντας τις προκλήσεις που ανακύπτουν όταν απαιτείται η ταυτόχρονη διαχείριση πολλαπλών drones με αποδοτικό, προσαρμοστικό και ακριβή τρόπο. Παρότι ο έλεγχος ενός μόνο drone είναι ένα θέμα που έχει μελετηθεί εκτενώς, η ταυτόχρονη διαχείριση ενός σμήνους drones παρουσιάζει νέες προκλήσεις, ιδιαίτερα όσον αφορά τη συνεργασία και το συγχρονισμό.

Συγκεκριμένα, οι προκλήσεις που επιχειρεί να αντιμετωπίσει αυτό το έργο περιλαμβάνουν:

1. **Δημιουργία αξιόπιστης δικτυακής επικοινωνίας** μεταξύ του υπολογιστή ελέγχου και πολλαπλών drones, διασφαλίζοντας σταθερή και αποτελεσματική μετάδοση δεδομένων και εντολών.
2. **Ανάπτυξη μιας αρθρωτής προσέγγισης ελέγχου συγχρονισμένων κινήσεων σμήνους** με χρήση Python, που να επιτρέπει την εύκολη επέκταση, τροποποίηση και παραμετροποίηση των drones και των κινήσεών τους.
3. **Δοκιμή και αξιολόγηση της δυνατότητας εκτέλεσης σύνθετων ελιγμών** από ένα σμήνος drones και ανάλυση της απόδοσης τους, διασφαλίζοντας τόσο τη διατήρηση των σχηματισμών όσο και την ακρίβεια στην εκτέλεση εντολών σε πραγματικό χρόνο.

Το προτεινόμενο έργο έχει ως τελικό στόχο να παρέχει μια ευέλικτη πλατφόρμα για την εκπαίδευση στον προγραμματισμό, τη ρομποτική και τα δίκτυα. Τα drones Tello EDU προσφέρουν μια ιδανική βάση για τον πειραματισμό και τη μάθηση, ενισχύοντας τη γνώση στον τομέα της ρομποτικής σμήνους και προωθώντας τις εκπαιδευτικές ευκαιρίες σε σχετικούς τομείς.

Μεθοδολογία

Η μεθοδολογία που ακολουθήθηκε για την επίτευξη των στόχων του έργου περιλαμβάνει τα εξής βήματα:

1. Έρευνα Αγοράς:

Πριν την υλοποίηση, πραγματοποιήθηκε συγκριτική έρευνα αγοράς για την αξιολόγηση άλλων πλατφορμών drone που είναι κατάλληλες για εκπαιδευτικούς σκοπούς (π.χ. Parrot Mambo, Robolink coDrone, Flybrix). Το Tello EDU επιλέχθηκε λόγω των προγραμματιστικών του δυνατοτήτων και της ευκολίας χρήσης του σε διαμόρφωση σμήνους.

2. Ρύθμιση Δικτύου (Network Setup):

Τα Tello EDU drones ρυθμίστηκαν σε λειτουργία "σταθμού" (station mode), ώστε να συνδέονται σε έναν router που λειτουργεί ως access point. Αυτό επιτρέπει την επικοινωνία μέσω Wi-Fi μεταξύ των drones και του υπολογιστή ελέγχου. Κάθε drone τέθηκε ξεχωριστά σε λειτουργία σταθμού με την εντολή ap ssid pass, όπου ssid είναι το όνομα του δικτύου Wi-Fi του router και pass ο κωδικός πρόσβασης.

3. Επικοινωνία με τα Drones (Communication Setup):

Δημιουργήθηκε ένα τοπικό UDP socket για κάθε drone για την αποστολή εντολών και τη λήψη απαντήσεων. Ο υπολογιστής ήταν υπεύθυνος για την αποστολή εντολών στα drones, ενώ κάθε drone είχε μια μοναδική θύρα επικοινωνίας για ανεξάρτητη ανταλλαγή μηνυμάτων.

4. Έλεγχος και Δοκιμές Σύνδεσης (Testing Connectivity):

Χρησιμοποιήθηκε το λογισμικό Packet Sender για την επαλήθευση της συνδεσιμότητας. Βασικές εντολές, όπως command και battery, στάλθηκαν για να επιβεβαιωθεί η λειτουργικότητα και η επικοινωνία με τα drones, με κάθε drone να λαμβάνει μια μοναδική IP διεύθυνση εντός του δικτύου.

5. Σύνταξη Κώδικα Ελέγχου Πτήσης (Flight Control Code):

Μέσω της βιβλιοθήκης dji-tellopy και της Python, αναπτύχθηκαν scripts για τον έλεγχο κινήσεων των drones, όπως απογείωση, αιώρηση, περιστροφή και προσγείωση. Οι εντολές στάλθηκαν στα drones με καθυστερήσεις, ώστε να εξασφαλιστεί η σωστή εκτέλεση των πτήσεων.

6. Διαχείριση Σμήνους (Swarm Control):

Αναπτύχθηκε μια κλάση (TelloSwarm) για τον συντονισμό πολλαπλών drones ταυτόχρονα. Αυτό επέτρεψε τον συγχρονισμό των κινήσεων των drones σε σμήνος, εξασφαλίζοντας ότι οι εντολές αποστέλλονται ταυτόχρονα σε όλα τα drones.

7. Εκτέλεση Πειραμάτων (Experiment Execution):

- Πραγματοποιήθηκαν διάφορα πειράματα για την αξιολόγηση των κινήσεων των drones σε συγχρονισμό:
 - a) Το πρώτο πείραμα επικεντρώθηκε στην απλή αιώρηση και προσγείωση.
 - b) Τα επόμενα πειράματα εισήγαγαν πιο πολύπλοκους σχηματισμούς, όπως το "κουτί" και ακροβατικά ελιγμούς (flips).

8. Ανάλυση Αποτελεσμάτων (Results Analysis):

Τα αποτελέσματα καταγράφηκαν και αξιολογήθηκαν για να ελεγχθεί η ακρίβεια της πτήσης και η ανταπόκριση των drones στις εντολές. Οι παρατηρήσεις επικεντρώθηκαν στο εάν τα drones διατηρούσαν συγχρονισμένες κινήσεις και αν οι εντολές εκτελούνταν με ακρίβεια.

ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Νοημοσύνη Σμήνους

Η συλλογική νοημοσύνη, ως ένας ανερχόμενος ερευνητικός τομέας, έχει τραβήξει την προσοχή πολλών επιστημόνων από τότε που εισήχθη τη δεκαετία του 1980. Σήμερα, αποτελεί ένα διεπιστημονικό σημείο αναφοράς που εμπλέκει επιστήμες όπως η τεχνητή νοημοσύνη, η οικονομία, η κοινωνιολογία και η βιολογία. Εδώ και καιρό έχει παρατηρηθεί ότι ορισμένα είδη καταφέρνουν να επιβιώσουν σε δύσκολες φυσικές συνθήκες, αξιοποιώντας τη δύναμη της συλλογικότητας αντί για την ατομική σοφία. Τα άτομα σε αυτές τις ομάδες δεν διαθέτουν υψηλή νοημοσύνη από μόνα τους, αλλά μέσω της συνεργασίας και του καταμερισμού εργασίας, καταφέρνουν να ολοκληρώνουν σύνθετες αποστολές. Η συνολική τους νοημοσύνη, ως ομάδα, είναι αυτο-οργανωμένη και προσαρμόζεται συνεχώς, επιδεικνύοντας εξαιρετική αποτελεσματικότητα (Ying TAN, 2013).

Η συλλογική νοημοσύνη μπορεί να θεωρηθεί ως μια μορφή «μαλακής» βιονικής, που αναπαράγει τις κοινωνικές δομές και αλληλεπιδράσεις των σμηνών στη φύση, αντί να εστιάζει στην ατομική νοημοσύνη, όπως γίνεται στην παραδοσιακή τεχνητή νοημοσύνη. Τα άτομα σε ένα τέτοιο σύστημα λειτουργούν σαν πράκτορες με απλές και περιορισμένες ικανότητες, ενώ κάποια από αυτά έχουν τη δυνατότητα να εξελίσσονται καθώς αντιμετωπίζουν προβλήματα, ώστε να προσαρμόζονται καλύτερα. Συνήθως, ένα σύστημα συλλογικής νοημοσύνης αποτελείται από μια ομάδα απλών ατόμων που δρουν αυτόνομα, καθοδηγούμενα από ένα σύνολο βασικών κανόνων και τοπικών αλληλεπιδράσεων. Αν και τα άτομα αυτά δεν είναι αφελή, η λειτουργία τους είναι σχετικά απλή σε σχέση με τη συνολική νοημοσύνη που επιτυγχάνεται συλλογικά. Όταν πολλά άτομα συνεργάζονται ή ανταγωνίζονται, προκύπτουν έξυπνες συμπεριφορές που δεν θα εμφανίζονταν ποτέ σε ένα μόνο άτομο. Το σμήνος μπορεί να εκτελεί καθήκοντα που κανονικά θα απαιτούσαν ένα πολύπλοκο άτομο, ενώ επιδεικνύει μεγαλύτερη ανθεκτικότητα και ευελιξία με χαμηλό κόστος. Η συλλογική νοημοσύνη εκμεταλλεύεται πλήρως τα πλεονεκτήματα της ομαδικής δράσης χωρίς την ανάγκη για κεντρικό έλεγχο ή ένα παγκόσμιο μοντέλο, αποτελώντας μια ιδανική λύση για μεγάλης κλίμακας σύνθετα προβλήματα.

Ορισμός και χαρακτηριστικά ρομποτικού σμήνους

Η ρομποτική σμήνους αποτελεί μια σύγχρονη προσέγγιση για τον συντονισμό συστημάτων πολλαπλών ρομπότ, τα οποία αποτελούνται από μεγάλο αριθμό κυρίως απλών ρομπότ. Η συλλογική συμπεριφορά προκύπτει από τις αλληλεπιδράσεις μεταξύ των ρομπότ, καθώς και από τις αλληλεπιδράσεις τους με το περιβάλλον. Αυτή η προσέγγιση αντλεί έμπνευση από τον τομέα της τεχνητής συλλογικής νοημοσύνης και από βιολογικές μελέτες σε έντομα, όπως τα μυρμήγκια και άλλα είδη της φύσης, που επιδεικνύουν συμπεριφορά σμήνους (Ying TAN, 2013).

Η έρευνα στη ρομποτική σμήνους επικεντρώνεται στον σχεδιασμό μεγάλου αριθμού απλών ρομπότ, στην κατασκευή του φυσικού τους σώματος, καθώς και στους μηχανισμούς ελέγχου της συμπεριφοράς τους. Τα ρομπότ του σμήνους είναι συνήθως απλά, μικρά και χαμηλού κόστους, ώστε να αξιοποιείται το πλεονέκτημα του μεγάλου αριθμού τους. Κομβικό ρόλο παίζει η τοπική επικοινωνία μεταξύ των ρομπότ, η οποία συμβάλλει στην επεκτασιμότητα και την ανθεκτικότητα του συστήματος.

Ένα απλό σύνολο κανόνων σε επίπεδο μεμονωμένων ρομπότ μπορεί να οδηγήσει στην εμφάνιση σύνθετων συμπεριφορών σε επίπεδο σμήνους. Οι κανόνες που διέπουν τη συμπεριφορά τους είναι εμπνευσμένοι από τη συνεργατική δράση που παρατηρείται στα σμήνη της φύσης. Το σύστημα είναι πλήρως κατανεμημένο και αποκεντρωμένο, προσφέροντας υψηλή αποδοτικότητα, παράλληλη επεξεργασία, επεκτασιμότητα και ανθεκτικότητα.

Οι εφαρμογές της ρομποτικής σμήνους είναι πολλές και ποικίλες. Περιλαμβάνουν εργασίες που απαιτούν μικροσκοπικά ρομπότ, όπως η κατανεμημένη ανίχνευση σε μικρομηχανές ή στο ανθρώπινο σώμα, ενώ ταυτόχρονα μπορεί να προσαρμόζεται σε αποστολές που απαιτούν οικονομικές λύσεις, όπως η εξόρυξη ή η γεωργική αναζήτηση τροφής. Επιπλέον, η ρομποτική σμήνους είναι κατάλληλη για αποστολές με υψηλό κόστος σε χρόνο και χώρο, ή για εργασίες επικίνδυνες για τον άνθρωπο ή τα ίδια τα ρομπότ, όπως η παροχή βοήθειας μετά από καταστροφές, η αναζήτηση στόχων και οι στρατιωτικές εφαρμογές.

Χαρακτηριστικά των σμηνών της φύσης

Δεδομένου ότι η ρομποτική σμήνους αντλεί την έμπνευσή της κυρίως από τα σμήνη στη φύση, αποτελεί ένα ιδανικό πλαίσιο για την ανάλυση των χαρακτηριστικών των φυσικών σμηνών. Η μελέτη της ρομποτικής σμήνους ξεκίνησε πριν από περίπου έναν αιώνα. Αρχικά, υποτέθηκε ότι κάθε άτομο στο σμήνος έχει μια μοναδική ταυτότητα (ID) για να διευκολύνει τη συνεργασία και

την επικοινωνία, ενώ η ανταλλαγή πληροφοριών θεωρούνταν ότι γινόταν μέσω ενός κεντρικού δικτύου. Οι βασίλισσες σε αποικίες μυρμηγκιών και μελισσών θεωρούνταν υπεύθυνες για τη μετάδοση και διανομή των πληροφοριών στους υπόλοιπους πράκτορες.

Ωστόσο, η έρευνα των τελευταίων δεκαετιών έχει αποδείξει ότι δεν υπάρχουν μοναδικές ταυτότητες ή πληροφορίες αποθηκευμένες κεντρικά στο δίκτυο. Κανένας μεμονωμένος πράκτορας δεν έχει πρόσβαση σε όλες τις πληροφορίες του δικτύου, και επομένως δεν υπάρχει κάποιος κεντρικός ρυθμιστής. Πλέον, οι βιολόγοι θεωρούν ότι τα κοινωνικά σμήνη λειτουργούν ως αποκεντρωμένα συστήματα, καταναμημένα σε όλο το περιβάλλον, και μπορούν να περιγράψουν μέσω πιθανολογικών μοντέλων (Ying TAN, 2013).

Οι πράκτορες στα σμήνη ακολουθούν δικούς τους κανόνες, βασισμένους στις πληροφορίες που λαμβάνουν τοπικά. Η συλλογική συμπεριφορά προκύπτει από αυτούς τους τοπικούς κανόνες, οι οποίοι επηρεάζουν τόσο την ανταλλαγή πληροφοριών όσο και τη δομή της τοπολογίας του σμήνους. Αυτοί οι κανόνες αποτελούν επίσης το βασικό στοιχείο που διασφαλίζει την ευελιξία και την ανθεκτικότητα της συνολικής δομής, ακόμα και όταν αναπτύσσονται πιο σύνθετες συμπεριφορές.

Πλεονεκτήματα της ρομποτικής σμήνους

Τα πλεονεκτήματα και τα χαρακτηριστικά ενός συστήματος ρομποτικής σμήνους αναδεικνύονται μέσα από τη σύγκριση με ένα μεμονωμένο ρομπότ και άλλα συστήματα που περιλαμβάνουν πολλαπλά άτομα. Αυτά τα χαρακτηριστικά εμφανίζουν πολλές ομοιότητες με εκείνα που παρατηρούνται στα φυσικά σμήνη.

Σύγκριση με ένα μεμονωμένο ρομπότ

Για να εκτελέσει ένα πολύπλοκο έργο, ένα μεμονωμένο ρομπότ πρέπει να έχει περίπλοκη δομή και συστήματα ελέγχου, κάτι που αυξάνει σημαντικά το κόστος σχεδιασμού, κατασκευής και συντήρησης. Ένα τέτοιο ρομπότ είναι επίσης ευάλωτο, καθώς μια μικρή βλάβη σε κάποιο μέρος του μπορεί να επηρεάσει ολόκληρη τη λειτουργία του συστήματος, καθιστώντας δύσκολο να προβλεφθούν οι συνέπειες. Αντίθετα, η ρομποτική σμήνους μπορεί να πετύχει αντίστοιχες δυνατότητες μέσω της συνεργασίας μεταξύ των μελών του σμήνους, αξιοποιώντας την επαναχρησιμοποίηση απλών πρακτόρων και το χαμηλό κόστος κατασκευής και συντήρησης.

Επιπλέον, η ρομποτική σμήνους επωφελείται από την υψηλή παραλληλία και είναι ιδιαίτερα κατάλληλη για εργασίες μεγάλης κλίμακας.

Σε αντίθεση με τα μεμονωμένα ρομπότ που αντλούν έμπνευση από τις ανθρώπινες συμπεριφορές, η ρομποτική σμήνους βασίζεται στη μελέτη κοινωνικών ζώων. Λόγω των περιορισμών της τρέχουσας τεχνολογίας, είναι δύσκολο να προσομοιωθούν οι ανθρώπινες αλληλεπιδράσεις με μηχανές ή υπολογιστές, ενώ οι μηχανισμοί των κοινωνικών ζώων είναι πιο εύκολα εφαρμόσιμοι. Αυτό προσδίδει στη ρομποτική σμήνους σημαντικές προοπτικές για την αντιμετώπιση σύνθετων και μεγάλης κλίμακας προβλημάτων. Τα πλεονεκτήματα της ρομποτικής σμήνους αναλύονται παρακάτω.

- 1) **Παράλληλο.** Στη ρομποτική σμήνους, το μέγεθος του πληθυσμού είναι συνήθως αρκετά μεγάλο, επιτρέποντας την ταυτόχρονη διαχείριση πολλαπλών στόχων σε μια εργασία. Αυτό σημαίνει ότι το σμήνος μπορεί να αναλάβει αποστολές με πολλούς στόχους που είναι καταναμημένοι σε ευρεία περιοχή, μειώνοντας έτσι σημαντικά τον χρόνο που απαιτείται για την αναζήτησή τους.
- 2) **Επεκτάσιμο.** Η αλληλεπίδραση μεταξύ των μελών του σμήνους είναι τοπική, επιτρέποντας σε οποιοδήποτε άτομο να συμμετέχει ή να αποχωρεί από την εργασία ανά πάσα στιγμή, χωρίς να επηρεάζεται η συνολική λειτουργία του σμήνους. Το σμήνος έχει την ικανότητα να προσαρμόζεται στις αλλαγές του πληθυσμού, αναδιανέμοντας τις εργασίες μέσω έμμεσων μηχανισμών, χωρίς να απαιτείται εξωτερική παρέμβαση. Αυτό δείχνει επίσης ότι το σύστημα μπορεί να προσαρμόζεται σε διαφορετικά μεγέθη πληθυσμού χωρίς να χρειάζεται τροποποίηση του λογισμικού ή του υλικού, κάτι που το καθιστά ιδιαίτερα χρήσιμο για εφαρμογές στον πραγματικό κόσμο.
- 3) **Σταθερό.** Όπως συμβαίνει με την επεκτασιμότητα, τα συστήματα ρομποτικής σμήνους δεν επηρεάζονται σοβαρά ακόμη και όταν κάποια μέλη του σμήνους αποχωρούν λόγω απρόβλεπτων συνθηκών. Το σμήνος μπορεί να συνεχίσει να εργάζεται για την ολοκλήρωση της αποστολής, αν και η απόδοσή του ενδέχεται να μειωθεί με τη μείωση του αριθμού των ρομπότ. Αυτή η ιδιότητα είναι ιδιαίτερα ωφέλιμη για αποστολές σε επικίνδυνα περιβάλλοντα.
- 4) **Οικονομικό.** Όπως προαναφέρθηκε, το κόστος στη ρομποτική σμήνους παραμένει χαμηλό όσον αφορά τον σχεδιασμό, την παραγωγή και τη συντήρηση. Το συνολικό κόστος

ενός συστήματος ρομποτικής σμήνους είναι χαμηλότερο από εκείνο ενός μεμονωμένου, πολύπλοκου ρομπότ, ακόμα και όταν το σμήνος περιλαμβάνει εκατοντάδες ή χιλιάδες ρομπότ. Τα ρομπότ στο σμήνος μπορούν να παραχθούν μαζικά με χαμηλό κόστος, σε αντίθεση με τα μεμονωμένα ρομπότ που απαιτούν πιο περίπλοκη και ακριβή μηχανολογική κατασκευή.

- 5) **Ενεργειακή αποδοτικότητα.** Επειδή τα άτομα που αποτελούν το σμήνος είναι πολύ μικρότερα και απλούστερα σε σύγκριση με ένα μεγάλο και πολύπλοκο ρομπότ, το ενεργειακό κόστος είναι σημαντικά χαμηλότερο σε σχέση με το μέγεθος της μπαταρίας. Αυτό έχει ως αποτέλεσμα τη μεγαλύτερη διάρκεια ζωής του σμήνους. Σε περιβάλλοντα όπου δεν υπάρχουν διαθέσιμες υποδομές για ανεφοδιασμό ή όπου η χρήση καλωδιωμένης ηλεκτρικής ενέργειας δεν είναι εφικτή, η ρομποτική σμήνους μπορεί να αποδειχθεί πολύ πιο πρακτική και αποτελεσματική από τα παραδοσιακά μεμονωμένα ρομπότ.

Συμπερασματικά, η ρομποτική σμήνους μπορεί να εφαρμοστεί σε πολύπλοκα προβλήματα που περιλαμβάνουν μεγάλη ποσότητα χρόνου, χώρου ή στόχων, και όπου υπάρχει ένας βαθμός κινδύνου στο περιβάλλον. Οι τυπικές εφαρμογές περιλαμβάνουν: έλεγχο μη επανδρωμένων αεροσκαφών (UAV), ανακούφιση μετά από καταστροφές, εξόρυξη, γεωλογικές έρευνες, στρατιωτικές εφαρμογές και συνεργατική μεταφορά. Η ρομποτική σμήνους μπορεί να εκτελέσει αυτές τις αποστολές μέσω της συνεργασίας που αναπτύσσεται ανάμεσα στα μέλη του σμήνους, ενώ ένα μεμονωμένο ρομπότ θα δυσκολευόταν να προσαρμοστεί σε τέτοιες συνθήκες. Γι' αυτόν τον λόγο η ρομποτική σμήνους έχει εξελιχθεί σε σημαντικό πεδίο έρευνας την τελευταία δεκαετία.

Διαφορετικό από άλλα συστήματα πολλαπλών πρακτόρων

Υπάρχουν διάφοροι ερευνητικοί τομείς που αντλούν έμπνευση από τα σμήνη στη φύση και συχνά συγχέονται με τη ρομποτική σμήνους, όπως τα συστήματα πολλαπλών πρακτόρων και τα δίκτυα αισθητήρων. Αυτοί οι τομείς επίσης εκμεταλλεύονται τη συνεργατική συμπεριφορά που αναπτύσσεται μεταξύ των πρακτόρων για την εκτέλεση εξειδικευμένων εργασιών. Ωστόσο, υπάρχουν ορισμένες θεμελιώδεις διαφορές που τα ξεχωρίζουν από τη ρομποτική σμήνους, όπως παρουσιάζεται στον Πίνακα 1 (Ying TAN, 2013).

Από τον **Πίνακα 1**, είναι ξεκάθαρο ότι οι κύριες διαφορές αφορούν τον αριθμό των μελών, τον τρόπο ελέγχου, την ομοιογένεια και τη δυνατότητα επεκτασιμότητας των συστημάτων.

Comparison of swarm robotics and other systems.

	Swarm robotics	Multi-robot system	Sensor network	Multi-agent system
Population Size	Variation in great range	Small	Fixed	In a small range
Control	Decentralized and autonomous	Centralized or remote	Centralized or remote	Centralized or hierarchical or network
Homogeneity	Homogeneous	Usually heterogeneous	Homogeneous	Homogeneous or heterogeneous
Flexibility	High	Low	Low	Medium
Scalability	High	Low	Medium	Medium
Environment	Unknown	Known or unknown	Known	Known
Motion	Yes	Yes	No	Rare
Typical applications	Post-disaster relief Military application Dangerous application	Transportation Sensing Robot football	Surveillance Medical care Environmental protection	Net resources management Distributed control

Εικόνα 1 – Comparison of swarm robotics and other systems (Ying TAN, 2013)

Δεδομένου ότι η ομοιογένεια και η δυνατότητα επέκτασης λαμβάνονται υπόψη από την αρχή του σχεδιασμού, η ρομποτική σμήνους παρουσιάζει σημαντική ευελιξία και προσαρμοστικότητα σε σύγκριση με άλλα συστήματα. Τα συστήματα πολλαπλών ρομπότ συνήθως αποτελούνται από ετερογενή ρομπότ, που αν και επιτυγχάνουν καλύτερη απόδοση σε εξειδικευμένες εργασίες, θυσιάζουν την ευελιξία, την επαναχρησιμοποίηση και την επεκτασιμότητα. Εκτός από τη δυνατότητα κλιμάκωσης, που συζητήθηκε νωρίτερα, τα χαρακτηριστικά της ρομποτικής σμήνους.

- 1) **Αυτόνομο.** Τα ρομπότ σε ένα σύστημα ρομποτικής σμήνους πρέπει να είναι αυτόνομα, δηλαδή ικανά να αλληλοεπιδρούν με το περιβάλλον και να κινούνται. Με αυτές τις βασικές δυνατότητες, μπορούν να εισαχθούν συνεργατικοί μηχανισμοί εμπνευσμένοι από τη φύση. Αν και τα συστήματα, όπως τα δίκτυα αισθητήρων, διαφέρουν από τη ρομποτική σμήνους ως προς αυτό, η έρευνα σε αυτούς τους τομείς μπορεί να συμβάλει στην κατανόηση της ρομποτικής σμήνους.
- 2) **Αποκέντρωση.** Με τη χρήση κατάλληλων συνεργατικών κανόνων, τα άτομα μπορούν να ολοκληρώσουν τις αποστολές χωρίς κεντρικό έλεγχο, εξασφαλίζοντας έτσι την επεκτασιμότητα και την ευελιξία του συστήματος. Επιπλέον, το σμήνος έχει πλεονεκτήματα σε περιβάλλοντα όπου οι επικοινωνίες μπορεί να διακοπούν ή να καθυστερήσουν, βελτιώνοντας την απόκριση και την ακρίβεια.
- 3) **Τοπική ανίχνευση και επικοινωνίες.** α ρομπότ στο σμήνος διαθέτουν περιορισμένο εύρος ανίχνευσης και επικοινωνίας, με αποτέλεσμα το σύστημα να είναι κατανεμημένο στο περιβάλλον. Η χρήση παγκόσμιας επικοινωνίας θα μείωνε την επεκτασιμότητα και την ευελιξία, καθώς το κόστος της επικοινωνίας αυξάνεται εκθετικά με τον πληθυσμό. Ωστόσο, κάποιες μορφές ελεγχόμενης παγκόσμιας επικοινωνίας είναι αποδεκτές, όπως η

αποστολή τελικών σημάτων ή η ενημέρωση στρατηγικών ελέγχου, εφόσον δεν παρεμβαίνουν στις τοπικές αλληλεπιδράσεις.

- 4) **Ομοιογενές.** Σε ένα σμήνος ρομπότ, τα ρομπότ θα πρέπει να εκτελούν όσο το δυνατόν λιγότερους διακριτούς ρόλους, με πολλά ρομπότ να αναλαμβάνουν κάθε ρόλο. Αν οι ρόλοι των ρομπότ είναι ιδιαίτερα εξειδικευμένοι, το σύστημα δεν θεωρείται ρομποτική σμήνους. Για παράδειγμα, ρομπότ σε ομάδες ποδοσφαίρου δεν θεωρούνται ρομποτική σμήνους, καθώς κάθε ρομπότ έχει συγκεκριμένο ρόλο κατά τη διάρκεια του παιχνιδιού
- 5) **Ευελξία.** Ένα σμήνος υψηλής ευελξίας μπορεί να εκτελέσει πολλές διαφορετικές αποστολές με το ίδιο υλικό και ελάχιστες τροποποιήσεις στο λογισμικό, όπως ακριβώς τα φυσικά σμήνη εκτελούν διαφορετικές εργασίες. Τα ρομπότ μπορούν να προσαρμόσουν τις ικανότητες και τις συνεργατικές τους στρατηγικές ανάλογα με την αποστολή και το περιβάλλον. Μέσω μηχανικής μάθησης, το σμήνος μπορεί να βελτιώνει τις στρατηγικές του με βάση προηγούμενες εμπειρίες (Spezzano, 2024).

Πεδίο εφαρμογών της ρομποτικής σμήνους

Πεδίο εφαρμογών της ρομποτικής σμήνους Τα τελευταία χρόνια, η έρευνα στον τομέα της ρομποτικής σμήνους έχει επικεντρωθεί σε εργασίες σε δύσκολα προσβάσιμες ή επικίνδυνες περιοχές. Αυτά τα προβλήματα κατηγοριοποιούνται σε δύο κύριες κατηγορίες: η πρώτη αφορά πρότυπα συμπεριφοράς, όπως η χαρτογράφηση και η μετανάστευση, ενώ η δεύτερη εστιάζει στις οντότητες του περιβάλλοντος, όπως η αναζήτηση στόχων ή η διάσωση σε περιοχές καταστροφής. Επιπλέον, υπάρχουν και πιο σύνθετα προβλήματα που συνδυάζουν στοιχεία και από τις δύο κατηγορίες (Ying TAN, 2013).

Πιθανές εφαρμογές της ρομποτικής σμήνους:

1. Εργασίες μεγάλης κλίμακας: Η ρομποτική σμήνους είναι ιδιαίτερα κατάλληλη για εργασίες που απαιτούν κάλυψη μεγάλης περιοχής, όπως η ανίχνευση περιβαλλοντικών αλλαγών. Τα ρομπότ μπορούν να περιπολούν, να ανιχνεύουν και να αντιδρούν δυναμικά στο περιβάλλον, προσφέροντας καλύτερη παρακολούθηση από ένα σταθερό δίκτυο αισθητήρων.
2. Επικίνδυνες εργασίες: Η ρομποτική σμήνους μπορεί να αντεπεξέλθει σε επικίνδυνες εργασίες χάρη στην ανθεκτικότητά της, ακόμα και αν χαθούν ρομπότ κατά τη διάρκεια της αποστολής. Λόγω του χαμηλού κόστους των ρομπότ, οι αποστολές σε επικίνδυνες περιοχές είναι οικονομικά αποδεκτές.

3. Κλιμάκωση πληθυσμού: Οι αποστολές όπου το έργο μεταβάλλεται με τον χρόνο μπορούν να επωφεληθούν από την προσαρμοστική κλιμάκωση του πληθυσμού των ρομπότ, ανάλογα με τις ανάγκες της αποστολής.
4. Πλεονασματικότητα: Η ικανότητα του σμήνους να συνεχίζει να λειτουργεί ακόμα και με απώλειες ρομπότ εξασφαλίζει την επιτυχή ολοκλήρωση της αποστολής.

Σύστημα ρομποτικής σμήνους στην πραγματική ζωή

Τα τελευταία χρόνια, οι ερευνητές έχουν ήδη χρησιμοποιήσει τη ρομποτική σμήνους σε αρκετές εφαρμογές της πραγματικής ζωής, περιλαμβάνοντας τις περισσότερες από τις προαναφερθείσες εργασίες.

Ο William et al. πρότεινε ένα πλαίσιο, το οποίο ονόμασαν Physicomimetics, για τον καταναμημένο έλεγχο των σμηνών. Επικεντρώθηκαν στις ρομποτικές συμπεριφορές που μοιάζουν με αυτές που παρουσιάζουν τα στερεά, τα υγρά και τα αέρια. Οι διάφοροι σχηματισμοί υιοθετούνται για τις διάφορες εργασίες, περιλαμβάνοντας την καταναμημένη ανίχνευση, την αποφυγή εμποδίων, την επιτήρηση και τη σάρωση.

Ο Correll πρότεινε ένα σύστημα επιθεώρησης βασισμένο σε σμήνος με νοημοσύνη, για την επιθεώρηση λεπίδων σε στρόβιλο αεριωθουμένων. Το σύστημα βασίζεται σε ένα σμήνος αυτόνομων, μικρορομπότ, χρησιμοποιώντας μόνο ενσωματωμένους, τοπικούς αισθητήρες.

Το Senseable City Lab του MIT ανέπτυξε ένα στόλο ρομπότ απορρόφησης πετρελαίου χαμηλού κόστους, που ονομάζεται Seaswarm, για τη συλλογή πετρελαίου από την επιφάνεια της θάλασσας. Ένα νανοϋλικό ρομπότ μπορεί να απορροφήσει πετρέλαιο έως και 20 φορές το βάρος του. Το σύστημα παρέχει μια αυτόνομη και χαμηλού κόστους λύση για την προστασία του θαλάσσιου περιβάλλοντος (Ying TAN, 2013).

Τα Roombots είναι ένα καινοτόμο, αυτοαναδιαμορφούμενο αρθρωτό ρομποτικό σύστημα. Τα αυτόνομα αρθρωτά ρομπότ μπορούν να αλλάξουν το σχήμα τους για να προσαρμοστούν σε μια δεδομένη εργασία και περιβάλλον εργασίας, όπως η αυτόματη συναρμολόγηση και η αναδιαμόρφωση στατικών αντικειμένων, όπως έπιπλα στο καθημερινό περιβάλλον.

Η Formica είναι μια πλατφόρμα ρομποτικής σμήνους με βιολογική έμπνευση, που μπορεί να κλιμακωθεί. Ο πρωτοποριακός μηχανικός της σχεδιασμός επιτρέπει την παραγωγή μέσω γραμμών

συναρμολόγησης κυκλωμάτων. Το σύστημα αξιοποιεί μικρά, οικονομικά και ανθεκτικά ρομπότ και μπορεί να επεκταθεί σε πληθυσμούς εκατοντάδων ατόμων. Οι επιστήμονες θεωρούν ότι τέτοια σμήνη είναι κατάλληλα για αποστολές, όπως η εξερεύνηση του Άρη ή η διάσωση μετά από σεισμούς.

Η ρομποτική σμήνους έχει επίσης πιθανές στρατιωτικές εφαρμογές. Ο Pettinaro και οι συνεργάτες του ανέπτυξαν ένα σύστημα αυτοαναδιαμορφούμενων ρομπότ για αποστολές αναζήτησης, διάσωσης και εύρεσης τροφής, το οποίο μπορεί να διαχειριστεί τυχαίες αποτυχίες. Οι στρατιωτικοί ειδικοί πιστεύουν ότι τα βιονικά εναέρια οχήματα, βασισμένα στην τεχνολογία σμήνους, θα είναι λειτουργικά μέσα στα επόμενα χρόνια. Προβλέπεται ότι μηχανικές μέλισσες ή κατσαρίδες, εξοπλισμένες με συστήματα αναγνώρισης και βόμβες, ενδέχεται να χρησιμοποιηθούν σε μελλοντικές στρατιωτικές συγκρούσεις.

Γενικό μοντέλο ρομποτικής σμήνους

Το μοντέλο της ρομποτικής σμήνους είναι ουσιαστικό για τη λειτουργία του συνεργατικού αλγορίθμου που κατευθύνει τις συμπεριφορές και τις αλληλεπιδράσεις των ρομπότ στο σμήνος. Σε αυτό το μοντέλο, τα ρομπότ πρέπει να έχουν συγκεκριμένες βασικές δυνατότητες, όπως η ικανότητα ανίχνευσης, επικοινωνίας και κίνησης.

Το μοντέλο διαρθρώνεται σε τρεις κύριες ενότητες, καθεμία από τις οποίες εστιάζει σε διαφορετικούς μηχανισμούς που καθορίζουν συγκεκριμένες συμπεριφορές: ανταλλαγή πληροφοριών, βασικές συμπεριφορές και πιο εξελιγμένες συμπεριφορές. Η διαδικασία ανταλλαγής πληροφοριών ανάμεσα σε αυτές τις ενότητες είναι καθοριστική. Τα ρομπότ στο σμήνος ανταλλάσσουν δεδομένα μεταξύ τους και διαχέουν τις πληροφορίες σε όλη την ομάδα μέσω των αυτόνομων ενεργειών τους, οδηγώντας σε συντονισμένη συνεργασία σε επίπεδο σμήνους (Ying TAN, 2013).

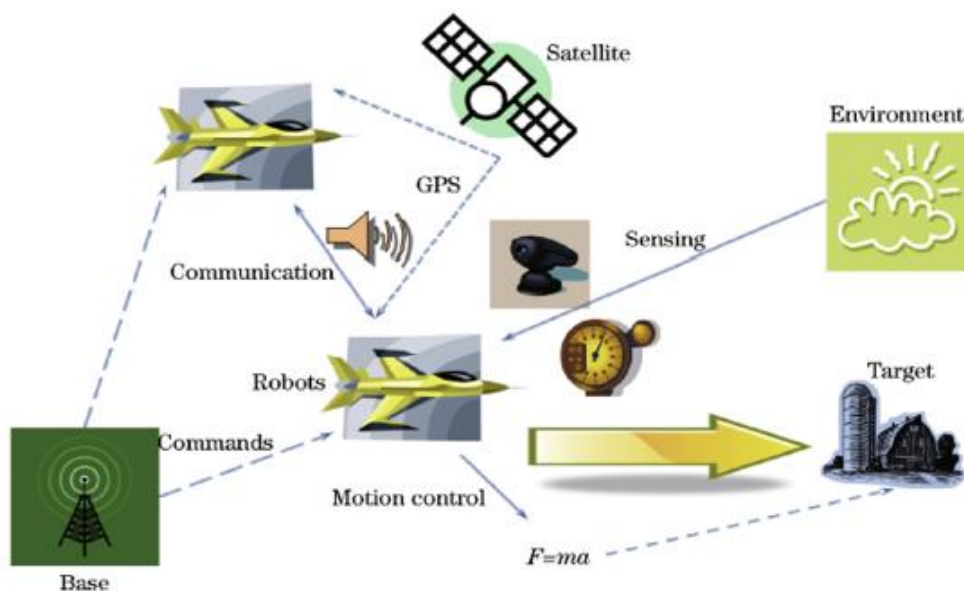
Το γενικό μοντέλο ρομποτικής σμήνους παρουσιάζεται στην Εικόνα 2. Τα ρομπότ επικοινωνούν μεταξύ τους. Σε ορισμένες περιπτώσεις, μπορεί να εισαχθούν παγκόσμια θέση ή κεντρικές εντολές, αλλά το σμήνος πρέπει ακόμα να είναι ικανό να ολοκληρώσει την εργασία αν η παγκόσμια επικοινωνία διακοπεί.

Ενότητα ανταλλαγής πληροφοριών

Η ανταλλαγή πληροφοριών είναι απαραίτητη όταν τα ρομπότ συνεργάζονται και αποτελεί κρίσιμο στοιχείο για τον συντονισμό των συμπεριφορών του σμήνους. Οι κύριες λειτουργίες που εκτελούν τα ρομπότ σε αυτό το πλαίσιο περιλαμβάνουν την περιορισμένη ανίχνευση και την τοπική επικοινωνία. Οι πληροφορίες που διαμοιράζεται κάθε ρομπότ κατατάσσονται σε δύο βασικούς τύπους: αλληλεπιδράσεις με άλλα ρομπότ ή με το περιβάλλον. Οι στρατηγικές αλληλεπίδρασης μπορεί να είναι είτε κοινές είτε να ποικίλουν, ανάλογα με τις ανάγκες της αποστολής.

Στη φύση, τα μέλη ενός σμήνους μπορούν να αλληλεπιδρούν απευθείας μέσω φυσικής επαφής, κινήσεων ή ήχων. Παράλληλα, υπάρχουν και πιο έμμεσοι τρόποι αλληλεπίδρασης. Για παράδειγμα, τα άτομα μπορεί να ανιχνεύσουν πληροφορίες από το περιβάλλον, να αντιδράσουν σε αυτές και να αφήσουν σημάδια για τα υπόλοιπα μέλη του σμήνους. Αυτή η διαδικασία, που ενισχύεται από θετική ανατροφοδότηση, συμβάλλει στη βελτίωση των επιμέρους συμπεριφορών και τελικά οδηγεί σε σύνθετες συμπεριφορές σε επίπεδο σμήνους (Ying TAN, 2013).

Υπάρχουν τρεις βασικές μέθοδοι για τη μεταφορά πληροφοριών στο σμήνος: απευθείας επικοινωνία, ανταλλαγή μέσω περιβάλλοντος και ανίχνευση. Μάλιστα, συχνά μπορεί να χρησιμοποιηθούν συνδυασμοί αυτών των μεθόδων για την αποτελεσματική συνεργασία των ρομπότ.



Εικόνα 2 - General model of swarm robotics (Ying TAN, 2013)

- 1) **Άμεση επικοινωνία.** Η άμεση επικοινωνία στα σμήνη ρομπότ είναι παρόμοια με τα ασύρματα δίκτυα και μπορεί να χωριστεί σε δύο τύπους: επικοινωνία peer-to-peer και εκπομπή (broadcast). Υπάρχουν ήδη τεχνολογίες κινητής που μπορούν να εφαρμοστούν άμεσα για αυτήν την επικοινωνία. Ωστόσο, οι ασύρματοι αισθητήρες, που συνήθως αποτελούν το μισό κόστος ενός ρομπότ, αυξάνουν το κόστος καθώς μεγαλώνει το σμήνος. Επιπλέον, η αύξηση του πληθυσμού των ρομπότ εκτοξεύει το απαιτούμενο εύρος ζώνης, καθιστώντας αναγκαίο τον περιορισμό της άμεσης επικοινωνίας.
Παρά τη διαθεσιμότητα των ασύρματων τεχνολογιών, τα πρωτόκολλα και οι τοπολογίες που είναι εξειδικευμένες για τη ρομποτική σμήνους δεν είναι ακόμα πλήρως γνωστά. Η επικοινωνία σε αυτά τα συστήματα πρέπει να αξιοποιεί την τοπική ανίχνευση και την κίνηση, με έμφαση στην ενίσχυση της συνεργασίας και των δυναμικών τοπολογιών του σμήνους.
- 2) **Επικοινωνία μέσω περιβάλλοντος.** Το περιβάλλον παίζει ρόλο μεσολαβητή στην αλληλεπίδραση μεταξύ των ρομπότ. Τα ρομπότ αφήνουν ίχνη μετά από μια ενέργεια, τα οποία ανιχνεύουν άλλα ρομπότ, χωρίς να υπάρχει άμεση επικοινωνία μεταξύ τους. Αυτή η διαδικασία δημιουργεί μια αλληλουχία ενεργειών, οδηγώντας τελικά σε συντονισμένη συμπεριφορά του σμήνους. Αυτός ο τύπος επικοινωνίας μιμείται τις φερομόνες που χρησιμοποιούν τα μυρμήγκια και οι μέλισσες. Ωστόσο, για να λειτουργήσει αυτή η προσέγγιση, το περιβάλλον πρέπει να υποστηρίζει την «αποθήκευση» των φερομονών, κάτι που μπορεί να δημιουργήσει περιορισμούς.
- 3) **Ανίχνευση.** Τα ρομπότ χρησιμοποιούν ενσωματωμένους αισθητήρες για να ανιχνεύσουν άλλα ρομπότ και το περιβάλλον τους. Αυτή η ανίχνευση τα βοηθά να εκτελούν εργασίες όπως η αποφυγή εμποδίων, η αναζήτηση στόχων και ο σχηματισμός ομάδων. Η πρόκληση εδώ είναι να συνδυαστούν οι αισθητήρες για αποτελεσματική συνεργασία εντός του σμήνους.
Η βασική διαφορά μεταξύ επικοινωνίας και ανίχνευσης είναι ότι η επικοινωνία είναι ενεργή (δηλαδή τα ρομπότ στέλνουν πληροφορίες), ενώ η ανίχνευση είναι παθητική (λαμβάνουν πληροφορίες). Ενώ η ακριβής επικοινωνία απαιτεί πιο περίπλοκο εξοπλισμό και συγχρονισμό, αυτό ανεβάζει σημαντικά το κόστος σε επίπεδο χρόνου, εύρους ζώνης και ενέργειας καθώς ο πληθυσμός αυξάνεται. Ένα μοντέλο ρομποτικής σμήνους θα πρέπει να επιδιώκει την απλούστευση της επικοινωνίας και να βασίζεται περισσότερο στην

ανίχνευση, χρησιμοποιώντας στοιχεία όπως χρώματα και φωτεινότητα για την παροχή πληροφοριών (Ying TAN, 2013).

Μέθοδοι Μοντελοποίησης για τη Ρομποτική Σμήνους

Η μοντελοποίηση είναι χρήσιμη για την κατανόηση των λειτουργιών των συστημάτων σμήνους, καθώς ένας αλγόριθμος για ρομποτική σμήνους πρέπει να μπορεί να κλιμακώνεται σε εκατοντάδες ή χιλιάδες ρομπότ. Δεδομένου ότι η πραγματοποίηση τέτοιων πειραμάτων είναι δαπανηρή και χρονοβόρα, η μοντελοποίηση προσφέρει έναν πιο πρακτικό τρόπο να μελετηθεί η συμπεριφορά.

Οι μέθοδοι μοντελοποίησης διακρίνονται σε τέσσερις τύπους:

1. **Μοντελοποίηση βασισμένη σε αισθητήρες:** Σε αυτή την προσέγγιση, οι αισθητήρες και οι ενεργοποιητές των ρομπότ είναι τα κύρια συστατικά που μοντελοποιούνται, μαζί με τα αντικείμενα του περιβάλλοντος. Αυτή η μέθοδος επικεντρώνεται σε ρεαλιστικές αλληλεπιδράσεις των ρομπότ με το περιβάλλον και χρησιμοποιείται εδώ και καιρό για πειράματα στη ρομποτική.
2. **Μικροσκοπική μοντελοποίηση:** Εδώ τα ρομπότ και οι αλληλεπιδράσεις τους μοντελοποιούνται σαν μηχανές πεπερασμένων καταστάσεων. Οι συμπεριφορές των ρομπότ ορίζονται από διάφορες καταστάσεις, με συνθήκες που βασίζονται στην επικοινωνία και την ανίχνευση. Το μοντέλο αυτό επαναλαμβάνεται πολλές φορές για να προβλεφθούν οι μέσες συμπεριφορές του σμήνους.
3. **Μακροσκοπική μοντελοποίηση:** Αυτή η μέθοδος μοντελοποίησης αντιπροσωπεύει την παγκόσμια συμπεριφορά του συστήματος με εξισώσεις διαφορών, αντί να επικεντρώνεται στις επιμέρους συμπεριφορές των ρομπότ. Αντί να επαναλαμβάνει την ίδια τη συμπεριφορά, το μοντέλο δείχνει την τελική εικόνα και παρέχει μια γενική εικόνα της λειτουργίας του σμήνους.
4. **Μοντελοποίηση με βάση αλγόριθμους νοημοσύνης σμήνους:** Σχήματα συνεργασίας από την νοημοσύνη σμήνους έχουν εισαχθεί στη ρομποτική, με πιο συνηθισμένο παράδειγμα την βελτιστοποίηση σμήνους σωματιδίων – Particle Swarm Optimazation (PSO). Η PSO μιμείται την κίνηση σμηνών πουλιών καθώς αναζητούν τον βέλτιστο στόχο, κάτι που είναι παρόμοιο με τη συμπεριφορά των ρομπότ σε ένα σμήνος.

Αν και οι αλγόριθμοι αυτοί είναι ώριμοι, υπάρχουν ακόμα προκλήσεις όταν εφαρμόζονται στη ρομποτική σμήνους. Οι αλγόριθμοι συχνά βασίζονται σε παγκόσμιες αλληλεπιδράσεις ή τυχαίες κινήσεις, ενώ στη ρομποτική σμήνους αυτές οι λειτουργίες μπορεί να είναι δύσκολο να εφαρμοστούν άμεσα (Ying TAN, 2013).

Packet sender

Το Packet Sender είναι ένα λογισμικό ανοιχτού κώδικα που επιτρέπει την αποστολή και λήψη πακέτων TCP και UDP. Υποστηρίζει επίσης συνδέσεις TCP με χρήση SSL, έντονη δημιουργία κίνησης, αιτήματα HTTP(S) GET/POST και δημιουργία πάνελ. Είναι διαθέσιμο για Windows, Mac, και Linux. Διατίθεται υπό την άδεια GNU General Public License v2 και είναι ελεύθερο λογισμικό. Η ιστοσελίδα του Packet Sender αναφέρει ότι "Σχεδιάστηκε για να είναι πολύ εύκολο στη χρήση, προσφέροντας ταυτόχρονα αρκετές δυνατότητες ώστε οι power users να μπορούν να κάνουν αυτό που χρειάζονται" (Packet Sender, 2024).

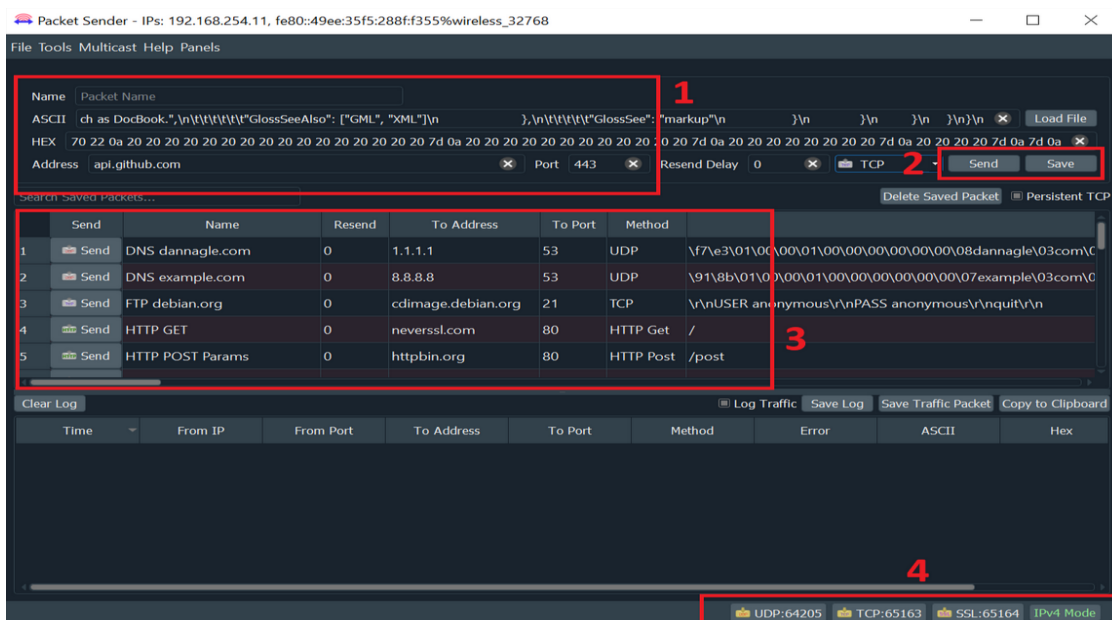
Χρήσεις Τυπικές εφαρμογές του Packet Sender περιλαμβάνουν:

- Αντιμετώπιση προβλημάτων σε συσκευές δικτύου που χρησιμοποιούν network servers (αποστολή πακέτου και ανάλυση της απάντησης).
- Αντιμετώπιση προβλημάτων σε συσκευές δικτύου που χρησιμοποιούν network clients (συσκευές που "καλούν σπίτι" μέσω UDP, TCP ή SSL—το Packet Sender μπορεί να καταγράψει αυτά τα αιτήματα).
- Δοκιμή και ανάπτυξη νέων πρωτοκόλλων δικτύου (αποστολή πακέτου, έλεγχος της συμπεριφοράς της συσκευής).
- Αντίστροφη μηχανική (reverse-engineering) πρωτοκόλλων δικτύου για ανάλυση ασφάλειας (όπως κακόβουλο λογισμικό).
- Αντιμετώπιση προβλημάτων ασφαλών συνδέσεων (με χρήση του SSL server και client).
- Αυτοματοποίηση (μέσω του command line interface του Packet Sender ή της δυνατότητας resend).
- Δοκιμή αντοχής μιας συσκευής (με χρήση του εργαλείου έντονης δημιουργίας δικτυακής κίνησης).
- Κοινή χρήση/Αποθήκευση/Συνεργασία με τη χρήση της υπηρεσίας Packet Sender Cloud.

Το Packet Sender έρχεται με έναν ενσωματωμένο TCP, UDP και SSL server σε πολλαπλές θύρες που ορίζει ο χρήστης. Αυτός παραμένει σε λειτουργία, ακούγοντας για πακέτα ενώ ταυτόχρονα αποστέλλει άλλα πακέτα.

GUI

Το Packet Sender είναι πανομοιότυπο για όλες τις εκδόσεις desktop. Η μόνη διαφορά είναι το θέμα του που προσαρμόζεται στο λειτουργικό σύστημα.



Εικόνα 3 – GUI Packet Sender (Packet Sender documentation, 2024)

1. Ένα πακέτο έχει ένα όνομα, διεύθυνση προορισμού (τα ονόματα των τομέων επιλέγονται από προεπιλογή ακριβώς πριν την αποστολή), την θύρα και τα δεδομένα που σχετίζονται με αυτό.
2. Στον πίνακα υπάρχει μια λίστα αποθηκευμένων πακέτων. Μπορείτε να κάνετε διπλό κλικ για να επεξεργαστείτε απευθείας τα πεδία αυτού του πίνακα.
3. Στο κάτω δεξί μέρος βρίσκονται η κατάσταση των servers UDP, TCP και SSL και οι θύρες τους. Κάνοντας κλικ πάνω σε αυτά ενεργοποιείτε ή απενεργοποιείτε το πρωτόκολλο. Το Packet Sender υποστηρίζει δέσμευση σε οποιονδήποτε αριθμό θυρών.
4. Υπάρχει επίσης ένα κουμπί εναλλαγής IP. Κάνοντας κλικ σε αυτό, αλλάζει σε IPv4 (προεπιλογή), IPv6, ή προσαρμοσμένη IP.

ΛΕΙΤΟΥΡΓΙΚΗ ΑΝΑΛΥΣΗ ΚΑΙ ΠΡΟΣΑΡΜΟΓΗ ΤΟΥ TELLO EDU ΓΙΑ ΤΗΝ ΟΛΟΚΛΗΡΩΜΕΝΗ ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ

Το Tello EDU (Εικόνα 3) είναι ένα προγραμματιζόμενο drone ιδανικό για εκπαιδευτική χρήση. Σε αυτή την ενότητα, θα εξετάσουμε πιο αναλυτικά το drone, τα χαρακτηριστικά του και άλλες πληροφορίες που είναι απαραίτητες για την ανάπτυξη εφαρμογών.

Το Tello EDU κατασκευάζεται από την Shenzhen Ryze Technology και ενσωματώνει την τεχνολογία ελέγχου πτήσης της DJI και επεξεργαστές της Intel.

Πριν από αυτή την έκδοση υπήρχε το Tello, το οποίο, σε αντίθεση με αυτό το νέο μοντέλο, ήταν λευκό και δεν επέτρεπε τον προγραμματισμό σμήνους με απλό τρόπο.

Αυτή η νέα έκδοση διαθέτει επίσης ένα νέο SDK, το οποίο είναι πιο διαδεδομένο από το προηγούμενο (Cavadas, 2024).

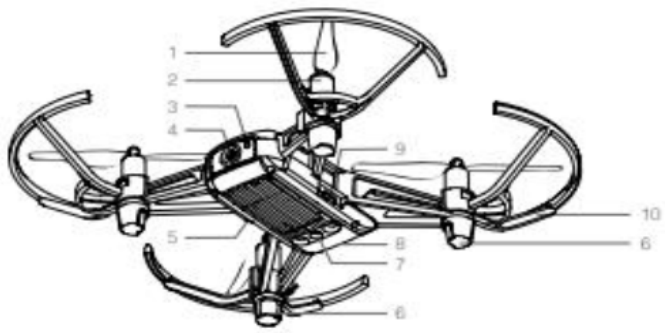


Εικόνα 4 -Tello EDU

Λειτουργικά Μέρη Κατασκευής

Τα κύρια στοιχεία του drone είναι τα εξής:

Περιγραφή των στοιχείων του drone Tello EDU

	1.Propellers
	2.Engines
	3.Drone Status Indicator
	4.Camera
	5.Power Button
	6.Antennas
	7.Vision Positioning System
	8.Battery
	9.Micro USB port
	10.Propeller protectors

Εικόνα 5 – Τεχνικά χαρακτηριστικά (Cavadas, 2024)

Οι διαστάσεις και τα χαρακτηριστικά του το καθιστούν πολύ ευέλικτο.

Πίνακας 1: Περιγραφή του drone

Βάρος	87 g
Διαστάσεις	98×92.5×41 mm
Προπέλα	3 ίντσες
Ενσωματωμένες Λειτουργίες	Αισθητήρας τηλεμετρίας
	Βαρόμετρο
	LED
	Σύστημα Όρασης
	Wi-Fi 2.4 GHz 802.11n
	Ροή βίντεο σε πραγματικό χρόνο 720p

Θύρα	Θύρα φόρτισης μπαταρίας USB
Εύρος Θερμοκρασίας Λειτουργίας	από 0° έως 40°
Εύρος Συχνοτήτων Λειτουργίας	από 2.4 έως 2.4835 GHz
Πομπός (EIRP)	20 dBm (FCC)
	19 dBm (CE)
	19 dBm (SRRC)

Ως drone που προορίζεται κυρίως για χρήση σε εσωτερικούς χώρους, τα χαρακτηριστικά του που καθορίζουν τις λειτουργίες του είναι αρκετά περιορισμένα.

Πίνακας 2.3. Λεπτομέρειες στη λειτουργία

Πίνακας 2: Λεπτομέρειες στη λειτουργία

Μέγιστη απόσταση πτήσης	100 m
Μέγιστη ταχύτητα	8 m/s
Μέγιστη διάρκεια πτήσης	13 λεπτά
Μέγιστο ύψος πτήσης	30 m

Διαθέτει μια πολύ εύκολη μπαταρία για αλλαγή· μπορεί να φορτιστεί απευθείας με καλώδιο USB με την μπαταρία μέσα στο drone ή με έναν σταθμό φόρτισης.

Πίνακας 3: Λεπτομέρειες μπαταρίας

Αποσπώμενη	Ναι
Χωρητικότητα	1100 mAh
Τάση	3.8 V
Τύπος	LiPo
Ενέργεια	4.18 Wh
Καθαρό Βάρος	25 ± 2 g

Εύρος Θερμοκρασίας Φόρτισης	από 5° έως 45°
Μέγιστη Ισχύς Φόρτισης	10 W

Η κάμερα μας επιτρέπει να αποκτούμε βίντεο με καλή ποιότητα (HD)· μετά την επεξεργασία της εικόνας, είναι δυνατόν να αναπτυχθούν διάφορες εφαρμογές.

Πίνακας 4: Λεπτομέρειες κάμερας

Φωτογραφία	5 MP (2592x1936)
Πεδίο Όρασης	82.6°
Βίντεο	HD 720p 30 fps
Μορφή	JPG (Φωτογραφία)
	MP4 (Βίντεο)
Ηλεκτρονική Σταθεροποίηση	Ναι

Σύστημα Οπτικής Τοποθέτησης: Αποτελείται από μια κάμερα και ένα υπέρυθρο τριδιάστατο module. Αυτό το σύστημα είναι ικανό να λειτουργεί σε εύρος ύψους από 0,3 μ. έως 30 μ., αλλά οι βέλτιστες συνθήκες λειτουργίας του είναι από 0,3 μ. έως 6 μ. Είναι ένα LED που διαθέτει το drone και υποδεικνύει σε ποια κατάσταση βρίσκεται το drone κάθε στιγμή (Ενεργοποίηση, χωρίς λήψη μηνυμάτων κ.λπ.).

Πεδίο Θέασης: Είναι η ανοιχτή παρατηρήσιμη περιοχή που μπορεί να δει η κάμερα του drone.

Ηλεκτρονική Σταθεροποίηση: Είναι μια τεχνική βελτίωσης εικόνας που χρησιμοποιεί ηλεκτρονική επεξεργασία.

Βασικό Περιβάλλον πριν από τον προγραμματισμό

Για να προγραμματίσετε το drone, είναι δυνατόν να χρησιμοποιήσετε διάφορα λειτουργικά συστήματα (Windows, Linux, MAC κ.λπ.). Για αυτό το έργο, έχει προγραμματιστεί σε Linux λόγω της ευκολίας λειτουργίας με τις διάφορες βιβλιοθήκες που επρόκειτο να χρησιμοποιηθούν.

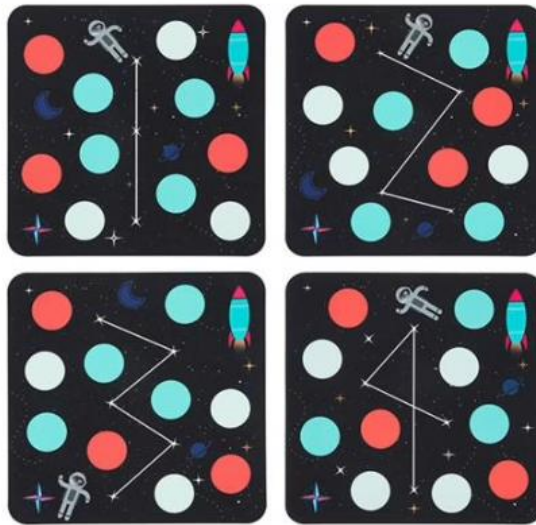
Είναι δυνατός ο προγραμματισμός σε διάφορες γλώσσες (Java, C++, Python, Scratch κ.λπ.). Για αυτό το έργο, έχει χρησιμοποιηθεί η Python.

Για να εκτελέσετε ένα script, είναι απαραίτητο να συνδέσετε τον υπολογιστή με το drone Tello EDU. Για να το κάνετε αυτό, πρέπει να συνδεθείτε στο Wi-Fi δίκτυο του Tello EDU. Το όνομα του Wi-Fi δικτύου, από προεπιλογή, μπορεί να φανεί μέσα στο drone όταν αφαιρεθεί η μπαταρία. Είναι δυνατή η αλλαγή του ονόματος του δικτύου και η προσθήκη κωδικού πρόσβασης για να αποτραπεί τυχόν ξένος να αποκτήσει έλεγχο του drone.

Πίνακες Αποστολής – Mission Pads

Το drone Tello EDU συνοδεύεται από τέσσερις πίνακες. Αυτοί οι πίνακες είναι γνωστοί ως Πίνακες Αποστολής (Εικόνα 5). Ο Πίνακας Αποστολής είναι ένας πίνακας διαστάσεων 15x15 εκ. που χρησιμεύει ως πρότυπο που μπορεί να αναγνωριστεί από το drone Tello EDU. Κάθε πλευρά περιέχει ένα μοναδικό πρότυπο που αποτελείται από:

- Πλανήτες: διάφορα πρότυπα που υποδεικνύουν την ID
- Μια ροκέτα: ακολουθεί τη διεύθυνση του άξονα x
- Μια ID: αριθμός μεταξύ 1 και 8



Εικόνα 6 – Mission Pads

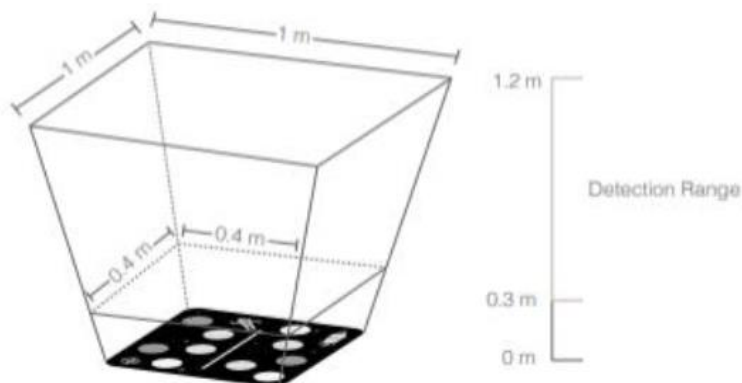
Η συχνότητα αντίληψης των καμερών είναι 20 Hz εάν είναι συνδεδεμένες ατομικά. Εάν οι δύο αρθρώσεις είναι συνδεδεμένες, έχουν 10 Hz. Η εμβέλεια αντίληψης για τον Πίνακα Αποστολής από το drone Tello EDU είναι η εξής (Εικόνα 6):

Detection Range

Height: 0.3 - 1.2 m

Range (height of 0.3 m): 0.4x0.4 m

Range (height of 1.2 m): 1x1 m



Εικόνα 7 – Detection Range (Cavadas, 2024)

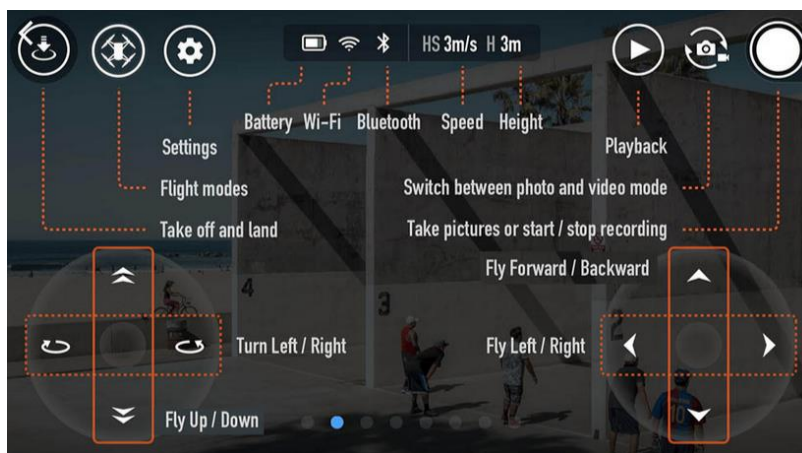
SDK 2.0

Το drone Tello EDU διαθέτει ένα κιτ ανάπτυξης λογισμικού (SDK), το οποίο χρησιμεύει ως βάση για την ανάπτυξη των διαφόρων εφαρμογών που πραγματοποιούνται σε αυτό το έγγραφο. Στο διαδίκτυο, μπορείτε να βρείτε έναν οδηγό χρήστη που έχει δημιουργηθεί από την RYZE Technologies, στον οποίο εξηγείται πώς να κάνετε τις συνδέσεις, όλες τις διάφορες εντολές που δέχεται το drone και πώς να το επαναφέρετε στην αρχική του λειτουργία.

Εφαρμογή

Το drone Tello EDU μπορεί να ελεγχθεί μέσω της εφαρμογής Tello της Shenzhen RYZE Tech Co. Ltd.

Στην κύρια διεπαφή (Εικ. 2.4), έχουμε και τους δύο joystick για να ελέγχουμε το drone και να βλέπουμε τι καταγράφει η κάμερα του drone. Σε αυτήν την ίδια διεπαφή, έχουμε δεδομένα όπως η μπαταρία, αν είναι συνδεδεμένο στο Wi-Fi ή Bluetooth, το ύψος και η ταχύτητα. Μπορούμε επίσης να τραβήξουμε φωτογραφίες και να ξεκινήσουμε την καταγραφή βίντεο.

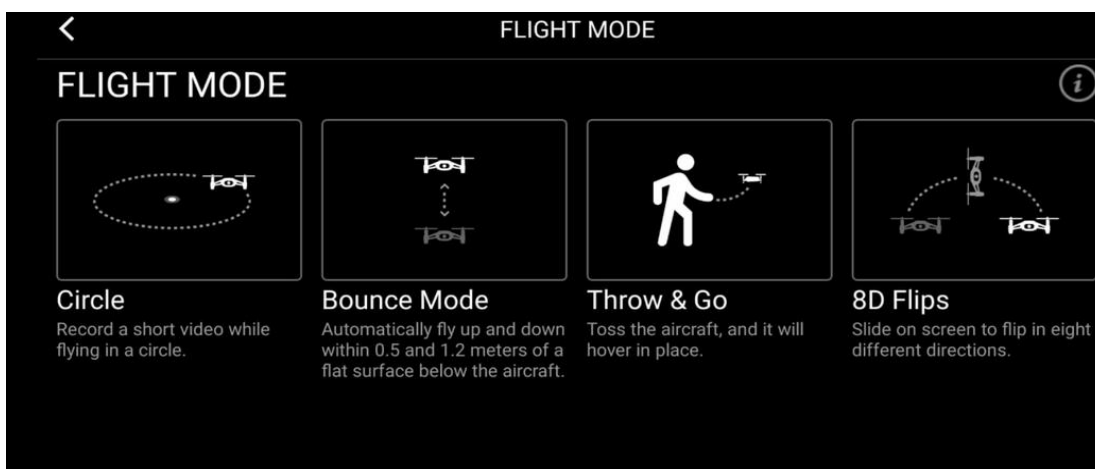


Εικόνα 8 – Εφαρμογή για τον έλεγχο του drone μέσω κινητού- Tello app

Τέλος, μπορούμε να έχουμε πρόσβαση σε διάφορες λειτουργίες, όπως απογειώση και προσγειώση, κάνοντας κύλιση στην οθόνη.

Και μπορούμε να έχουμε πρόσβαση στα διάφορα flight modes:

1. Απομακρυσμένο: Καταγράψτε ένα σύντομο βίντεο ενώ το drone απομακρύνεται και ανυψώνεται.
- 8D ανατροπές: Ανατρέψτε σε 8 διαφορετικές κατευθύνσεις.
2. Κύκλος: Καταγράψτε ένα σύντομο βίντεο ενώ πετάτε σε κύκλο.
3. 360°: Καταγράψτε ένα σύντομο βίντεο ενώ περιστρέφεστε 360°.
4. Ρίψη στον αέρα: Μπορείτε να ρίξετε το drone σε μια θέση στον αέρα. Επιπλέει εκεί.
5. Λειτουργία αναπήδησης: Πετάει αυτόματα μεταξύ 0,5 και 1,2 μέτρων.



Εικόνα 9 - Flight modes

Εντολές που μπορούν να σταλούν στο Tello EDU Drone

Σε αυτή την ενότητα θα εξηγηθούν οι διάφοροι τύποι εντολών που μπορούν να σταλούν στο Tello EDU Drone, και μέσα σε αυτούς τους τύπους θα αναλυθούν ορισμένες εντολές (όλες οι εντολές εμφανίζονται στον οδηγό χρήσης που αναφέρεται παραπάνω).

Οι εντολές μπορούν να οργανωθούν ως εξής:

Control commands (Εντολές Ελέγχου): Είναι οι βασικές εντολές για τον έλεγχο του drone.

Set commands (Εντολές Ρύθμισης): Χρησιμοποιούνται για τον ορισμό νέων τιμών υπο-παραμέτρων.

Read commands (Εντολές Ανάγνωσης): Χρησιμοποιούνται για την ανάγνωση της τρέχουσας τιμής των υπο-παραμέτρων.

Control Commands (Εντολές Ελέγχου):

Πίνακας 5: Εντολές Ελέγχου μέσω Python (Cavadas, 2024)

Command	Περιγραφή (Description)
command	Εισαγωγή σε λειτουργία SDK (Enter SDK mode)
streamon	Ενεργοποίηση ροής βίντεο (Enable video stream)
emergency	Διακοπή κινητήρων αμέσως
takeoff	Αυτόματη Απογείωση (Auto Takeoff)
up x	Ανύψωση έως 'x' εκ. $20 < x < 500$ (Ascend to 'x' cm)
cw x	Περιστροφή 'x' μοίρες δεξιόστροφα. $0 < x < 360$ (Rotate 'x' degrees clockwise)
flip x	Εκτέλεση περιστροφής προς 'x' κατεύθυνση. 'l', 'r', 'f', 'b' l: Αριστερά (Left) r: Δεξιά (Right)

	f: Μπροστά (Forward) b: Πίσω (Backward)
go x y z speed	Πέτα προς 'x' 'y' 'z' με ταχύτητα 'speed'. $-500 < x,y,z \text{ (cm)} < 500$. $10 < \text{ταχύτητα (cm/s)} < 100$
stop	Ακινητοποιείται στον αέρα. Σημείωση: λειτουργεί ανά πάσα στιγμή
go x y z speed mid	Πέτα προς 'x' 'y' 'z' του 'mid' με ταχύτητα 'speed'. $-500 < x,y,z < 500$. $10 < \text{ταχύτητα} < 100$

Καταστάσεις λειτουργίας (Mode) Tello Drone

Στο πρώτο μέρος της εικόνας 10 βλέπουμε ότι το drone λειτουργεί ως σημείο πρόσβασης (access point). Μέσω της εφαρμογής που συνοδεύει το drone, μπορούμε να εντοπίσουμε το Wi-Fi δίκτυό του και να συνδέσουμε το κινητό μας. Με αυτή τη σύνδεση, το κινητό μας αποκτά άμεση πρόσβαση στον έλεγχο και την παρακολούθηση του drone μέσω της εφαρμογής.

Στο δεύτερο μέρος της εικόνας 10 βλέπουμε ότι το drone παραμένει και πάλι ως σημείο πρόσβασης, αλλά αυτή τη φορά το Wi-Fi του εντοπίζεται και συνδέεται με ένα laptop. Ο έλεγχος του drone πραγματοποιείται μέσω Python scripts, δίνοντας έτσι τη δυνατότητα για μεγαλύτερη ευελιξία και δυνατότητα για αυτοματισμούς ή εκτέλεση πιο σύνθετων εντολών.

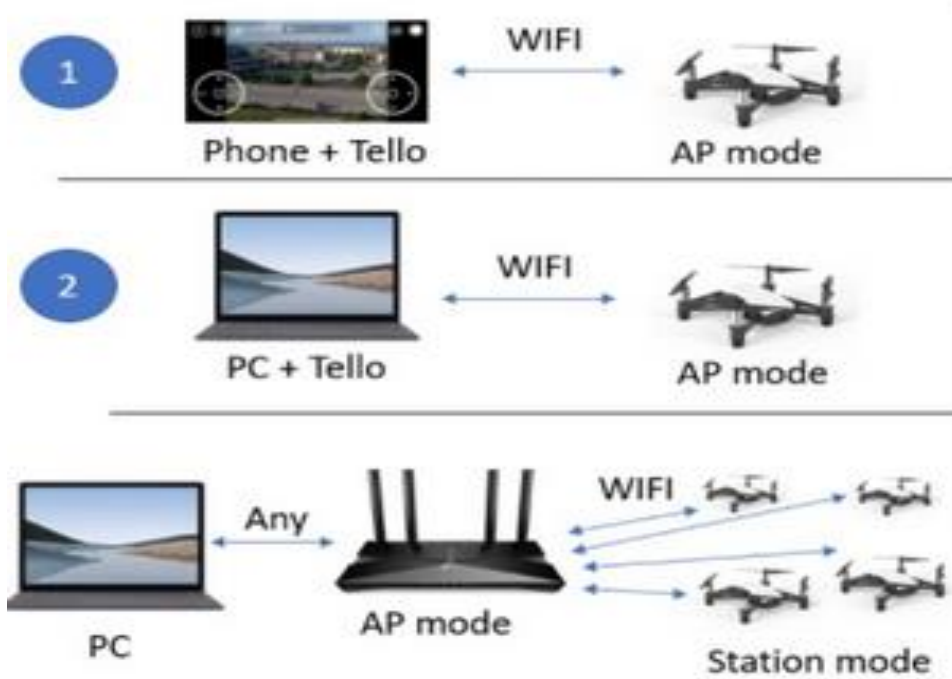
Στο τρίτο μέρος της εικόνας 9 τα στοιχεία που χρησιμοποιούνται για τη ρύθμιση του περιβάλλοντος σμήνους είναι τα εξής:

1. **2 x Tello EDU Drones**
2. **Laptop**
3. **Router**

Τα Tello EDU Drones θα είναι οι κύριοι πρωταγωνιστές μας, υπεύθυνοι για την εκτέλεση των υποδεικνυόμενων λειτουργιών.

Το laptop είναι το σημείο όπου γίνεται η επεξεργασία των δεδομένων και δίνονται οι εντολές στα drones μέσω ενός Python script.

Το router λειτουργεί ως σημείο διασύνδεσης μεταξύ του υπολογιστή και των drones, με όλα αυτά τα στοιχεία να είναι συνδεδεμένα στον router.



Εικόνα 10 – Καταστάσεις λειτουργίας Tello Drone

Ρύθμιση Σμήνους (Swarm Setup)

Για να ρυθμίσουμε το περιβάλλον, αυτό που πρέπει να κάνουμε είναι να συνδέσουμε τα drones στον router. Για αυτό, πρέπει να θέσουμε τα drones σε λειτουργία "σταθμού" (station mode). Αυτό επιτυγχάνεται με την αποστολή της εντολής **'ap ssid pass'**: • **ssid: Όνομα Wi-Fi (router)**

• **pass: Κωδικός πρόσβασης Wi-Fi (router)**

Η διαδικασία αυτή γίνεται ξεχωριστά για κάθε drone. Για να στείλουμε αυτή την εντολή, πρώτα συνδέουμε τον υπολογιστή με το drone που θέλουμε να θέσουμε σε λειτουργία "σταθμού". Το επόμενο βήμα είναι να στείλουμε την εντολή **'command'** και, τέλος, να στείλουμε την εντολή **'ap ssid pass'**. Ο router εκχωρεί μια διεύθυνση IP (τύπου 192.168.10.x) στο drone αυτόματα. Όταν το drone είναι σε λειτουργία "σταθμού", δεν μπορούμε να συνδεθούμε στο δίκτυό του από τον υπολογιστή. Για να επιστρέψουμε στην αρχική λειτουργία, πρέπει να επαναφέρουμε το drone πατώντας το κουμπί ενεργοποίησης για 5 δευτερόλεπτα ενώ είναι αναμμένο.



Εικόνα 11 – Drones in Station mode

Οικονομικός προϋπολογισμός

Πριν από την υλοποίηση του έργου, πραγματοποιήθηκε συγκριτική έρευνα αγοράς για την αξιολόγηση πλατφορμών drone που είναι κατάλληλες για εκπαιδευτικούς σκοπούς, όπως τα Parrot Mambo, Robolink coDrone και Flybrix. Το Tello EDU επιλέχθηκε λόγω των προγραμματιστικών του δυνατοτήτων και της ευκολίας χρήσης του για διαμόρφωση σχήματος (formation flying).

Κατά την έρευνα, διαπιστώθηκε ότι το κόστος αγοράς δύο Tello EDU είναι χαμηλότερο σε σχέση με τον εξοπλισμό και την κατασκευή δύο ή περισσότερων drones από το μηδέν. Επιπλέον, η δημιουργία ενός drone από το μηδέν συνοδεύεται από την ιδιαίτερα χρονοβόρα διαδικασία του tuning. Ένα σωστό tuning μπορεί να απαιτήσει εβδομάδες για να ολοκληρωθεί, γεγονός που θα καθυστερούσε σημαντικά την ολοκλήρωση του έργου.

Κατά συνέπεια, κρίθηκε πιο αποτελεσματικό να επενδυθεί χρόνος στο κύριο αντικείμενο της διπλωματικής, το οποίο αφορά τις επικοινωνίες μεταξύ των drones, παρά στη διαδικασία δημιουργίας και ρύθμισης ενός νέου drone. Αυτό θα επέτρεπε την καλύτερη επίτευξη των στόχων της διπλωματικής εργασίας, δεδομένου ότι η ανάπτυξη και το tuning ενός drone θα μπορούσαν να αποτελούν ένα ανεξάρτητο έργο από μόνα τους.

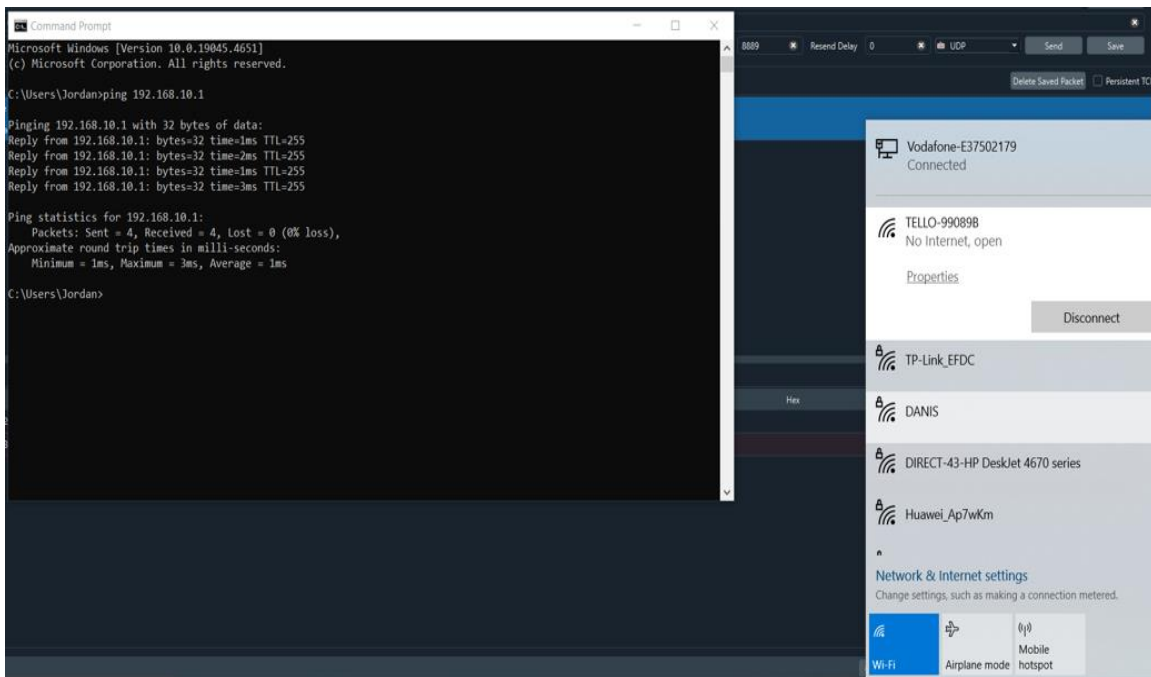
Τέλος, σε οικονομικό επίπεδο, ο προϋπολογισμός ήταν πιο συμφέρων με την επιλογή του Tello EDU, καθώς επιτυγχάνονται τα ίδια εκπαιδευτικά και ερευνητικά αποτελέσματα με μικρότερη επένδυση σε χρόνο και χρήματα. (Erkin Bahceci, 2005) (Autonomous rescue robot swarms for first responders, 2005) (Houxiang Zhang, 2006) (Sahin, 2004)

ΑΝΑΛΥΣΗ ΚΑΙ ΣΥΓΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ (SOFTWARE)

Διαδικασία ασύρματης σύνδεσης Tello drone 1/3

Η πρώτη διαδικασία για την υλοποίηση του έργου είναι η σύνδεση του drone Tello με τον υπολογιστή (laptop) μέσω ασύρματης σύνδεσης Wi-Fi. Αυτή η σύνδεση εξασφαλίζει την επικοινωνία και τον έλεγχο του drone μέσω του λογισμικού που θα αναπτυχθεί. Αρχικά, ελέγχουμε την επιτυχία της σύνδεσης, ανοίγοντας το command window (παράθυρο εντολών) στον υπολογιστή και εκτελώντας την εντολή ping προς τη διεύθυνση IP (192.168.10.1) του drone. Με αυτόν τον τρόπο, επιβεβαιώνουμε ότι υπάρχει σωστή επικοινωνία μεταξύ του Tello και του laptop.

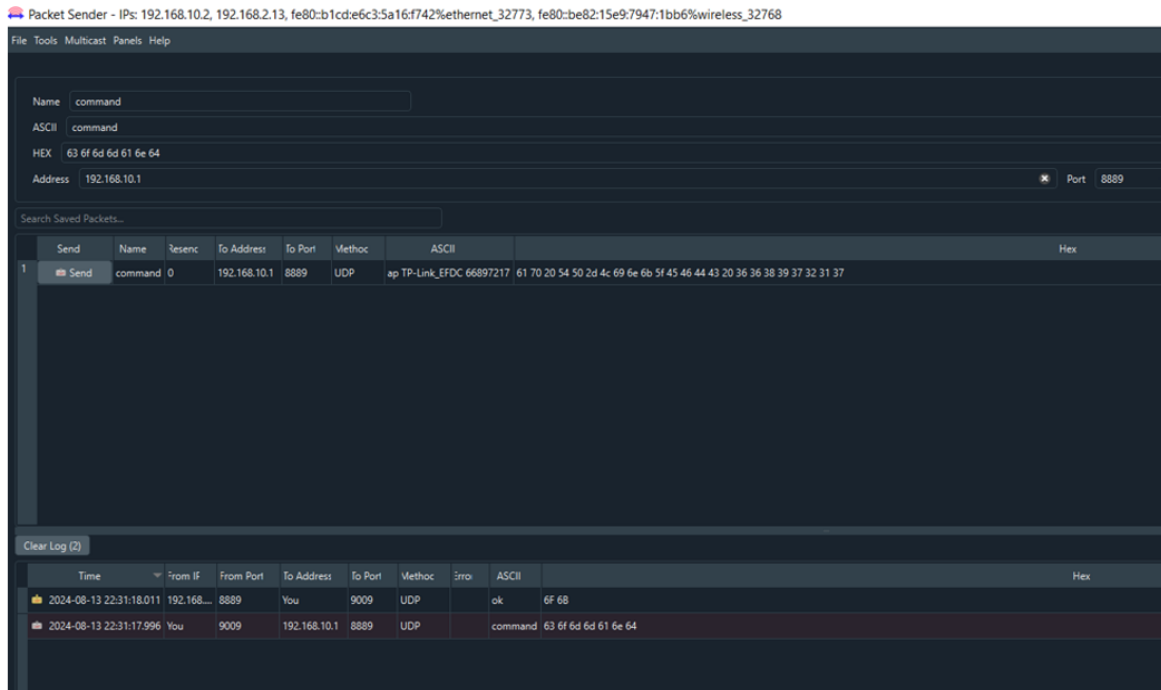
Η γνώση της IP διεύθυνσης του drone είναι απαραίτητη, καθώς θα αξιοποιηθεί στην επόμενη φάση για την αλλαγή της, ώστε να επιτρέψει τη δημιουργία ενός σμήνους με δύο Tello EDU drones. Αυτό θα μας επιτρέψει να αναπτύξουμε και να διαχειριστούμε ένα σύστημα με πολλαπλά drones, ενισχύοντας την πολυπλοκότητα και τη λειτουργικότητα του λογισμικού μας.



Εικόνα 12 – Επιβεβαίωση σύνδεσης Tello drone με το laptop(Wi-Fi)

Στη συνέχεια, χρησιμοποιώντας ένα λογισμικό που ονομάζεται Packet Sender, θα αποστείλουμε συγκεκριμένα μηνύματα στο drone Tello, με σκοπό να επιβεβαιώσουμε την αμφίδρομη επικοινωνία. Αυτή η διαδικασία μας επιτρέπει να ελέγξουμε αν το drone είναι σε θέση όχι μόνο να λαμβάνει τα μηνύματα αλλά και να ανταποκρίνεται σε αυτά, διασφαλίζοντας έτσι την ομαλή αλληλεπίδραση μεταξύ του λογισμικού και του drone. Η επιβεβαίωση αυτής της επικοινωνίας είναι κρίσιμη για την επιτυχημένη εκτέλεση των εντολών που θα αποστέλλονται στο drone κατά τη διάρκεια της λειτουργίας του.

Διαδικασία ασύρματης σύνδεσης Tello drone 2/3

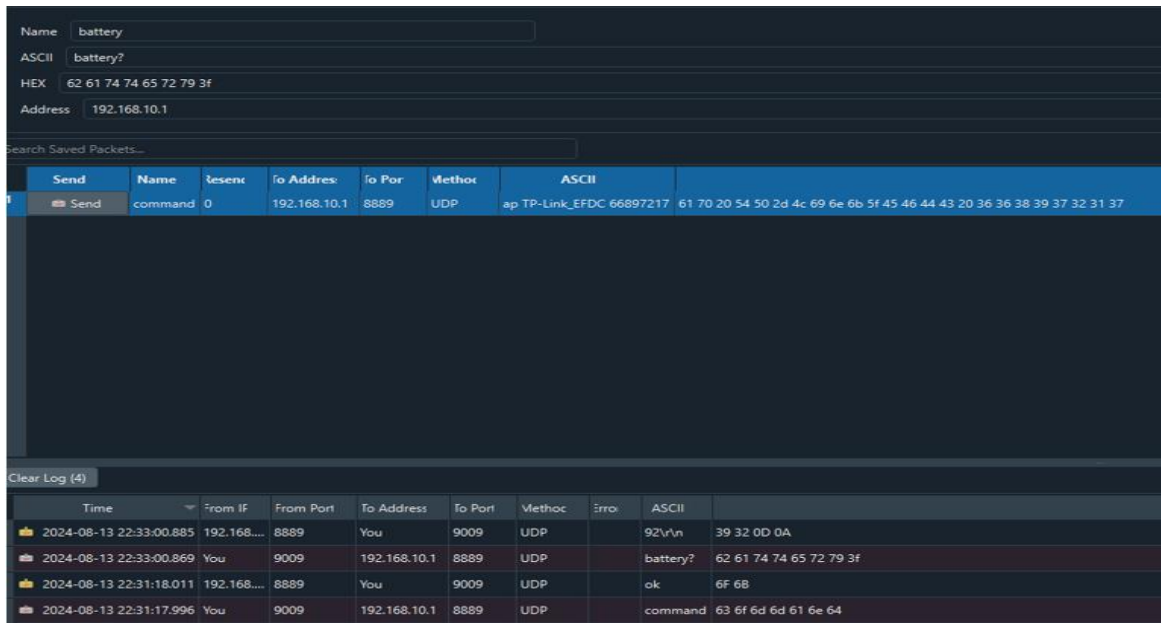


Εικόνα 13 – Εντολή επικοινωνίας μέσω Packet Sender

Η πρώτη εντολή που αποστέλλεται ονομάζεται "command", όπως φαίνεται και στην παραπάνω εικόνα. Κάθε φορά που αποστέλλεται μια εντολή, είναι απαραίτητο να καθορίζονται το όνομα της εντολής (Name), η αναπαράστασή της σε ASCII, η διεύθυνση IP του drone, και το πρόγραμμα επιστρέφει έναν συγκεκριμένο αριθμό σε μορφή HEX ανάλογα με την εντολή που δόθηκε, όπως

φαίνεται στην εικόνα. Στη συνέχεια, το drone απαντά με "OK", επιβεβαιώνοντας ότι τα μηνύματα-πακέτα στάλθηκαν επιτυχώς.

Διαδικασία ασυρματης συνδεσης Tello drone 3/3



Εικόνα 14 - Εντολή για τον έλεγχο της μπαταρίας μέσω Packet Sender

Η δεύτερη εντολή που αποστάλθηκε είχε ως σκοπό την επιβεβαίωση του ποσοστού της μπαταρίας. Γι' αυτό, το όνομα της εντολής (Name) ορίστηκε ως "battery", και η αναπαράστασή της σε ASCII περιλαμβάνει τη λέξη "battery" συνοδευόμενη από ένα ερωτηματικό. Σε αυτήν την περίπτωση, το ποσοστό της μπαταρίας ήταν 92%.

Στη συνέχεια, χρησιμοποιώντας το PyCharm, προχωρούμε στην εγκατάσταση ορισμένων βιβλιοθηκών, όπως η "djitello". Ακολούθως, θα εκτελεστεί ένα απλό script για να ελέγξουμε τη λειτουργικότητα, τη συνδεσιμότητα, καθώς και την ακρίβεια με την οποία το drone δέχεται εντολές μέσω ενός script γραμμένου σε Python, αντί να χρησιμοποιήσουμε την εφαρμογή στο κινητό.

Κώδικας

```
from djitellopy import tello
from time import sleep

me = tello.Tello()
me.connect()
print(me.get_battery())

me.takeoff()
me.send_rc_control(0, 50, 0, 0) ###move forward with this velocity###
sleep(2)###delay###
me.send_rc_control(0, 0, 0, 30) ### right yaw###
sleep(2)
me.send_rc_control(0, 0, 0, 0) ###just hover###
me.land()
```

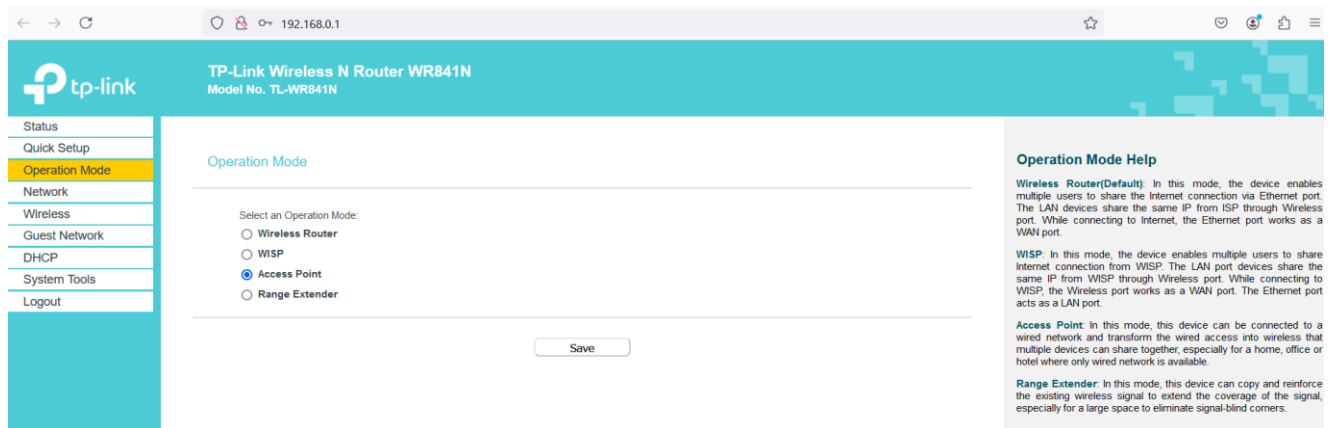
Με τον παραπάνω κώδικα, το drone έλαβε εντολή να κινηθεί μπροστά (pitch) με σταθερή ταχύτητα, να στρίψει δεξιά και στη συνέχεια να προσγειωθεί. Για να αντιστρέψουμε τις κινήσεις αυτές, δηλαδή να κινηθεί το drone προς τα πίσω και να εκτελέσει μια στροφή αριστερά (yaw), θα αρκούσε να χρησιμοποιηθούν αρνητικές τιμές στις αντίστοιχες εντολές (π.χ., -50 για την κίνηση προς τα πίσω, -30 για αριστερή στροφή).

Αφού ολοκληρώθηκε το πρώτο βασικό πείραμα, επιβεβαιώνοντας ότι το drone μπορεί να ελεγχθεί και να εκτελέσει βασικές κινήσεις μέσω ενός script γραμμένου σε Python, μπορούμε πλέον να προχωρήσουμε στο επόμενο βήμα, το οποίο είναι η διαχείριση πολλαπλών drones σε λειτουργία σμήνους.

Για να δημιουργηθεί ένα σμήνος, αρχικά απαιτείται η σύνδεση των drones σε ένα κοινό access point (router). Το router θα αποτελέσει το κεντρικό σημείο σύνδεσης και θα συνδεθεί με το laptop που θα εκτελεί τα scripts ελέγχου. Στη συνέχεια, θα ελεγχθούν και θα διαμορφωθούν οι ρυθμίσεις του access point, έτσι ώστε να διασφαλιστεί η σωστή και απρόσκοπτη επικοινωνία μεταξύ του laptop και των drones, εξασφαλίζοντας παράλληλα τη δυνατότητα αποστολής εντολών σε πραγματικό χρόνο προς όλα τα drones του σμήνους.

Αυτή η διαδικασία θέτει τα θεμέλια για τον συγχρονισμένο έλεγχο και την ομαλή εκτέλεση των προγραμματισμένων κινήσεων, καθώς και για τη μετέπειτα υλοποίηση πιο πολύπλοκων λειτουργιών του σμήνους.

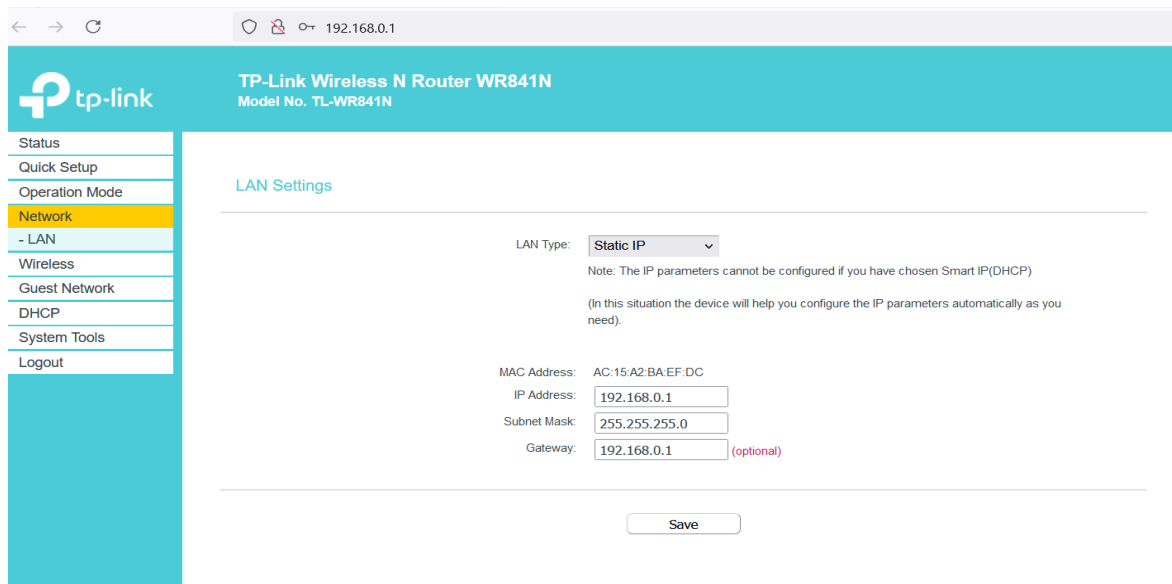
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 1/8



Εικόνα 15 – Αλλαγή "Wireless Router" σε "Access Point"

Η αλλαγή λειτουργίας από "Wireless Router" σε "Access Point" διασφαλίζει ότι όλα τα drones βρίσκονται εντός του ίδιου τοπικού δικτύου (LAN), επιτρέποντας τη συγχρονισμένη διαχείριση και τον έλεγχο του σμήνους. Επιπλέον, επιτρέπει την ταυτόχρονη αποστολή εντολών προς όλα τα drones, διευκολύνοντας έτσι την ακριβή εκτέλεση των κινήσεων και λειτουργιών που καθορίζονται από τα scripts.

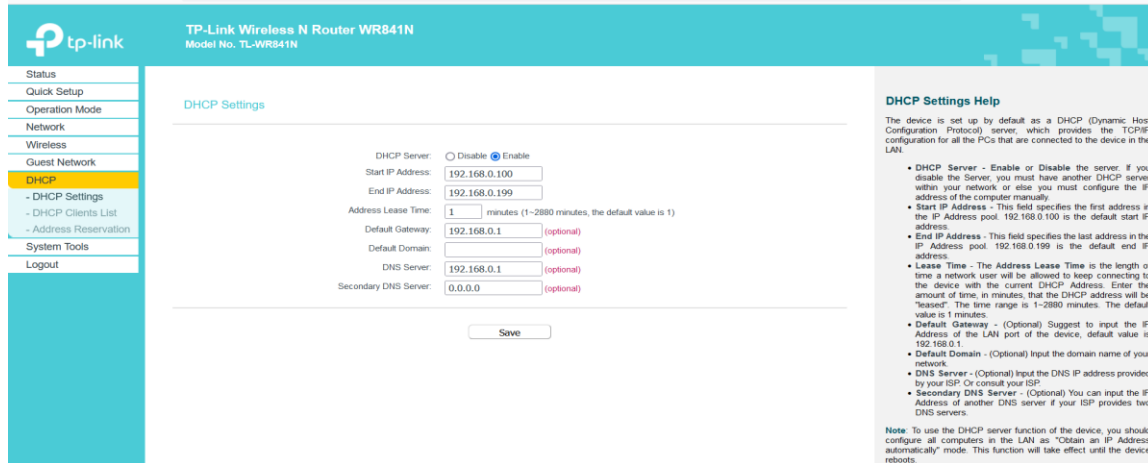
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 2/8



Εικόνα 16 – Αλλαγή από Dynamic σε Static IP

Στη συνέχεια, η διεύθυνση IP άλλαξε από Dynamic σε Static, ώστε να διασφαλιστεί ότι η IP των drones παραμένει σταθερή και δεν αλλάζει κάθε φορά που πραγματοποιείται επανεκκίνηση. Αυτό το βήμα είναι απαραίτητο για την αξιόπιστη επικοινωνία μεταξύ του laptop και των drones, καθώς μια στατική IP διευκολύνει την αναγνώριση και τον έλεγχο κάθε drone.

Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 3/8



The screenshot displays the DHCP Settings interface for a TP-Link Wireless N Router (Model No. TL-WR841N). The DHCP Server is set to 'Enable'. The Start IP Address is 192.168.0.100 and the End IP Address is 192.168.0.199. The Address Lease Time is set to 1 minute. The Default Gateway is 192.168.0.1, the Default Domain is blank, the DNS Server is 192.168.0.1, and the Secondary DNS Server is 0.0.0.0. A 'Save' button is located at the bottom of the form. To the right, a 'DHCP Settings Help' section provides detailed instructions for each field.

DHCP Settings Help

The device is set up by default as a DHCP (Dynamic Host Configuration Protocol) server, which provides the TCP/IP configuration for all the PCs that are connected to the device in the LAN.

- **DHCP Server** - Enable or Disable the server. If you disable the Server, you must have another DHCP server within your network or else you must configure the IP address of the computer manually.
- **Start IP Address** - This field specifies the first address in the IP Address pool. 192.168.0.100 is the default start IP address.
- **End IP Address** - This field specifies the last address in the IP Address pool. 192.168.0.199 is the default end IP address.
- **Lease Time** - The Address Lease Time is the length of time a network user will be allowed to keep connecting to the device with the current DHCP Address. Enter the amount of time, in minutes, that the DHCP address will be "leased". The time range is 1-2880 minutes. The default value is 1 minutes.
- **Default Gateway** - (Optional) Suggest to input the IP Address of the LAN port of the device, default value is 192.168.0.1.
- **Default Domain** - (Optional) Input the domain name of your network.
- **DNS Server** - (Optional) Input the DNS IP address provided by your ISP. Or consult your ISP.
- **Secondary DNS Server** - (Optional) You can input the IP Address of another DNS server if your ISP provides two DNS servers.

Note: To use the DHCP server function of the device, you should configure all computers in the LAN as "Obtain an IP Address automatically" mode. This function will take effect until the device reboots.

Εικόνα 17 – Εύρος IP διευθύνσεων

Τέλος, στο DHCP καθορίζεται το εύρος των IP διευθύνσεων που θα λάβουν τα drones όταν συνδεθούν με το access point μέσω του Packet Sender. Συγκεκριμένα, τόσο τα drones όσο και το access point θα έχουν IP διευθύνσεις εντός του εύρους 192.168.0.100 έως 192.168.0.199. Αυτό το εύρος εξασφαλίζει ότι όλες οι συσκευές του σμήνους βρίσκονται στο ίδιο δίκτυο και μπορούν να επικοινωνούν μεταξύ τους.

Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 4/8

Μετά την ολοκλήρωση της ρύθμισης του Access Point, μπορούμε να προχωρήσουμε στη σύνδεση των drones μέσω του Packet Sender. Αυτό το βήμα επιτρέπει την αποστολή και λήψη πακέτων δεδομένων, διασφαλίζοντας τον έλεγχο των drones και την ομαλή ροή των εντολών για τον συγχρονισμένο έλεγχο και τη διαχείριση του σμήνους.

Name

ASCII

HEX

Address

Search Saved Packets...

Send	Name	Resent	Io Address	Io Por	Method	ASCII	Hex
<input type="button" value="Send"/>	command	0	192.168.10.1	8889	UDP	ap TP-Link_EFDC 66897217	61 70 20 54 50 2d 4c 69 6e 6b 5f 45 46 44 43 20 36 36 38 39 37 32 31 37

near Log (6)

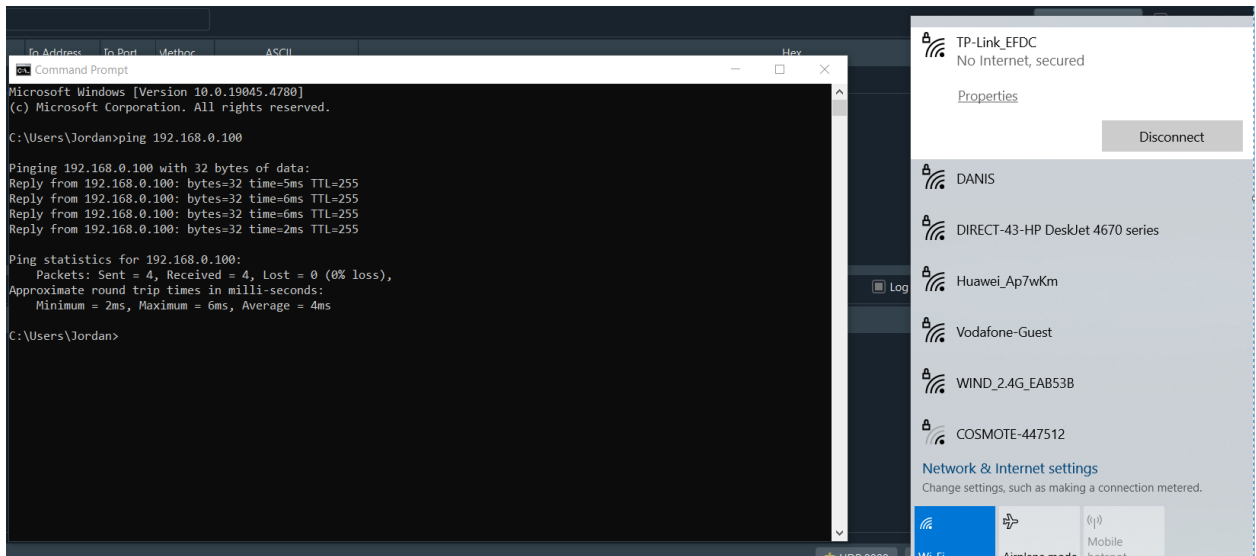
Time	From IP	From Port	To Address	To Port	Method	Erro	ASCII	Hex
2024-08-13 22:40:59.864	192.168.10.118	8889	You	9009	UDP		87\r\n	38 37 0D 0A
2024-08-13 22:41:10.118	192.168.10.118	8889	You	9009	UDP		OK,drone will reboot in 3s	4F 4B 2C 64 72 6F 6E 65 20 77 69 6C 6C 20 72 65 62 6F 6F 74 20 69 6E 20 33 73
2024-08-13 22:41:09.868	You	9009	192.168.10.1	8889	UDP		ap TP-Link_EFDC 66897217	61 70 20 54 50 2d 4c 69 6e 6b 5f 45 46 44 43 20 36 36 38 39 37 32 31 37
2024-08-13 22:40:59.848	You	9009	192.168.10.1	8889	UDP		battery?	62 61 74 74 65 72 79 3f
2024-08-13 22:40:49.760	You	9009	192.168.10.1	8889	UDP		command	63 6f 6d 6d 61 6e 64
2024-08-13 22:40:49.776	192.168.10.118	8889	You	9009	UDP		ok	6F 68

Εικόνα 18 – Εντολή σύνδεσης drone με το Access point

Η εντολή που στάλθηκε μέσω του Packet Sender για τη σύνδεση του drone με το access point (router) έχει την ακόλουθη μορφή: **Name:** command και **ASCII:** ap TP-Link_EFDC 66897217. Σε αυτήν την εντολή περιλαμβάνονται το όνομα και ο κωδικός πρόσβασης του access point, επιτρέποντας στο drone να πραγματοποιήσει σύνδεση.

Όπως φαίνεται στην παραπάνω εικόνα, το drone εκτελεί επανεκκίνηση (reboot) μετά από 3 δευτερόλεπτα και στη συνέχεια συνδέεται με το access point, αποκτώντας μια νέα IP διεύθυνση. Αυτή η νέα IP είναι απαραίτητο να εντοπιστεί για να συνεχιστεί η διαδικασία ελέγχου του drone, καθώς όλες οι επόμενες εντολές θα πρέπει να αποστέλλονται στη σωστή IP διεύθυνση για την ορθή λειτουργία και διαχείριση του σμήνους.

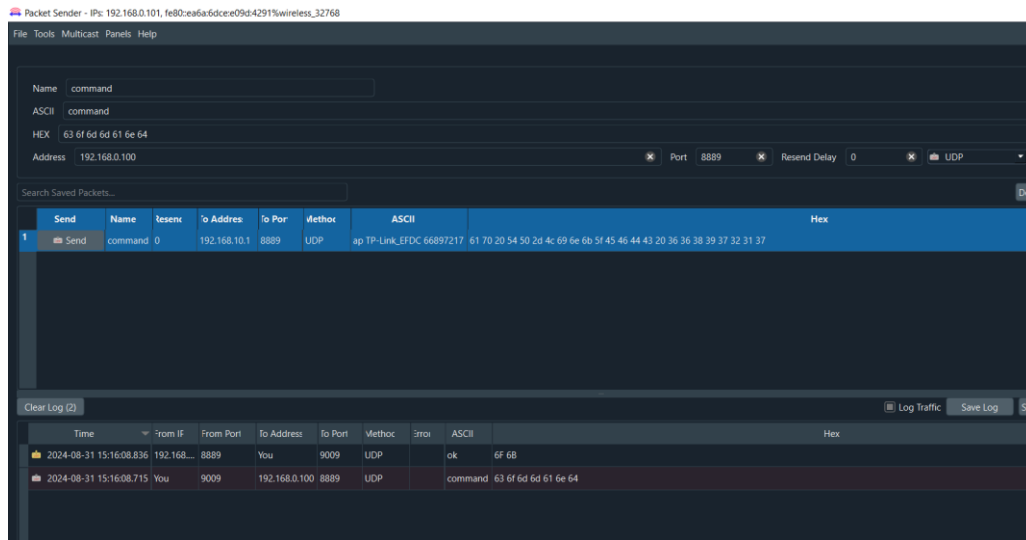
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 5/8



Εικόνα 19 – Αναζήτηση νέας IP

Αφού αποστείλαμε την εντολή μέσω του Packet Sender και ολοκληρώθηκε το reboot, στη συνέχεια συνδέσαμε το laptop με το Access Point, όπως φαίνεται στην παραπάνω εικόνα. Στη συνέχεια, αναζητήσαμε τη νέα IP διεύθυνση του drone, καθώς και την IP του Access Point. Μετά από λεπτομερή αναζήτηση, διαπιστώθηκε ότι η IP του πρώτου drone ήταν 192.168.0.100, ενώ η IP του Access Point ήταν 192.168.0.101. Αυτή η πληροφορία είναι σημαντική για την περαιτέρω επικοινωνία και έλεγχο του drone μέσω του δικτύου, διασφαλίζοντας ότι οι εντολές στέλνονται στις σωστές διευθύνσεις IP για την επιτυχή λειτουργία του σμήνους.

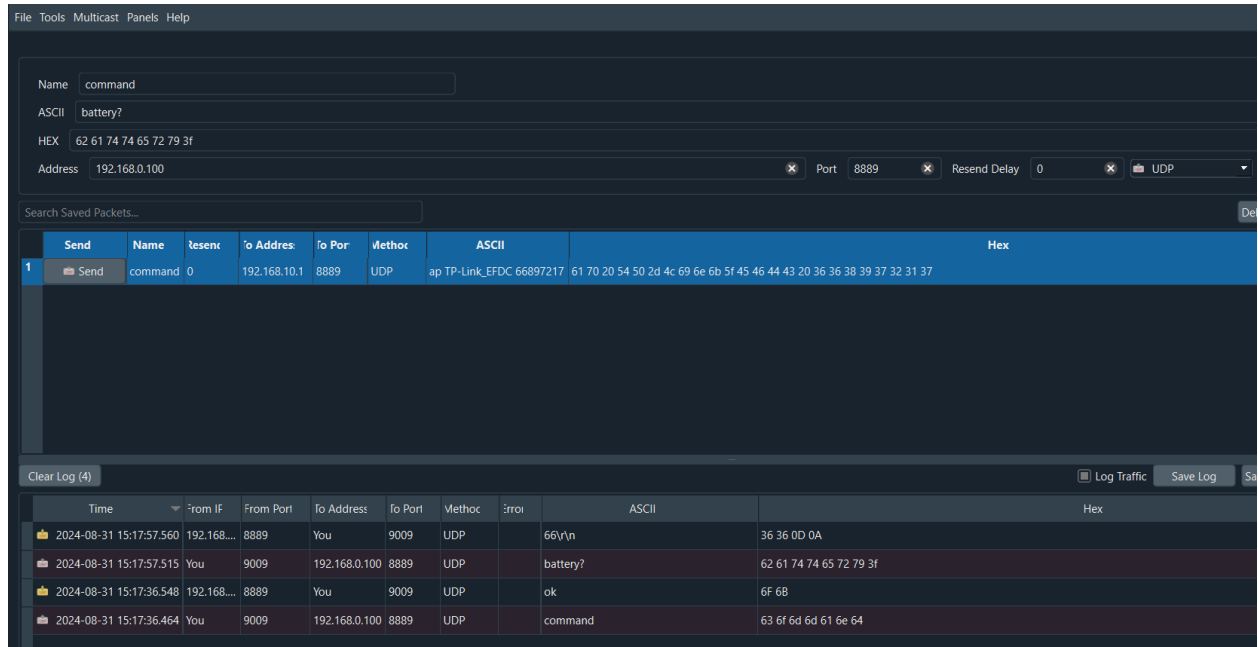
Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 6/8



Εικόνα 20 – Επιβεβαίωση επικοινωνίας με την καινούργια IP

Όπως φαίνεται στην παραπάνω εικόνα, αφού εντοπίσαμε την IP του πρώτου drone, αποστείλαμε εκ νέου την εντολή command, αυτή τη φορά προς τη νέα IP (192.168.0.100). Ο σκοπός ήταν διττός: αφενός να επιβεβαιώσουμε την επικοινωνία, κάτι που επιβεβαιώνεται από την απάντηση "OK" στην εικόνα, και αφετέρου να καθορίσουμε την IP που θα έχει κάθε drone στο σμήνος, καθώς και τις κινήσεις που θα εκτελούν στα επόμενα σενάρια. Είναι σημαντικό να σημειωθεί ότι όταν το drone βρίσκεται σε λειτουργία σμήνους, το χρώμα του φωτισμού που αναβοσβήνει δίπλα στην κάμερα αλλάζει από κίτρινο σε μωβ, παρέχοντας μια οπτική ένδειξη της αλλαγής λειτουργίας και της επιτυχούς ένταξης του drone στο σμήνος.

Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 7/8



Εικόνα 21 – Επιβεβαίωση ποσοστού μπαταρίας σε λειτουργία σμήνους

Στη συνέχεια, επιβεβαιώσαμε το ποσοστό της μπαταρίας για το συγκεκριμένο drone. Τα δύο τελευταία βήματα επαναλήφθηκαν και για το δεύτερο drone, του οποίου η IP διεύθυνση βρέθηκε να είναι 192.168.0.102. Για κάθε επιπλέον drone που προστίθεται στο σμήνος, επαληθεύουμε την IP του και επαναλαμβάνουμε τη διαδικασία αποστολής εντολών, επιβεβαιώνοντας τη συνδεσιμότητα και την επικοινωνία με κάθε drone.

Διαδικασία σύνδεσης του router για τον έλεγχο πολλαπλών drone (Station Mode) 8/8

Συνεπώς, οι IPs που παρατηρήθηκαν ήταν οι εξής:

- Drone 1: 192.168.0.100
- Drone 2: 192.168.0.102
- Access Point: 192.168.0.101

1^ο Πείραμα

```
import socket # Χρησιμοποιείται για επικοινωνία UDP
import threading # Για εκτέλεση threads (π.χ., ακρόαση απαντήσεων)
import time # Για παύση του προγράμματος

# Ορισμός IP και port για κάθε drone Tello
tello1_address = ('192.168.0.100', 8889) # IP και command port του
Tello 1
tello2_address = ('192.168.0.102', 8889) # IP και command port του
Tello 2

# Ορισμός IP και port του τοπικού υπολογιστή για κάθε drone
local1_address = ('', 9010) # Τοπικό port για επικοινωνία με το Tello 1
local2_address = ('', 9011) # Τοπικό port για επικοινωνία με το Tello 2

# Κλάση για τον χειρισμό επικοινωνίας με ένα drone Tello
class Tello:
    def __init__(self, local_address, tello_address):
        """
        Αρχικοποιεί ένα drone Tello δεσμεύοντας τοπική διεύθυνση και
        ρυθμίζοντας επικοινωνία με το drone.
        - local_address: Tuple που περιέχει την τοπική IP και port (για
        λήψη απαντήσεων)
        - tello_address: Tuple που περιέχει την IP και port του drone
        Tello
        """
        self.local_address = local_address
        self.tello_address = tello_address

        # Δημιουργία UDP socket για επικοινωνία
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

        # Δέσμευση του socket στη τοπική IP και port για λήψη απαντήσεων
        self.socket.bind(self.local_address)

        # Έναρξη ενός thread για ακρόαση απαντήσεων από το drone
```

```

        self.receive_thread =
threading.Thread(target=self.receive_thread)
        self.receive_thread.daemon = True # Εξασφαλίζει ότι το thread
θα κλείσει όταν το πρόγραμμα τελειώσει#
        self.receive_thread.start()

    def send_command(self, command):
        """
        Στέλνει εντολή στο drone Tello χρησιμοποιώντας UDP.
        - command: Η εντολή σε string μορφή που θα σταλεί (π.χ.
'takeoff', 'land')
        """
        self.socket.sendto(command.encode('utf-8'), self.tello_address)

    def receive_thread(self):
        """
        Αυτή η συνάρτηση εκτελείται σε ξεχωριστό thread και ακούει για
απαντήσεις από το drone Tello.
        Ακούει συνεχώς για μηνύματα και τα εκτυπώνει όταν λαμβάνονται.
        """
        while True:
            try:
                # Λήψη απάντησης (μέχρι 128 bytes) από το drone
                response, ip_address = self.socket.recvfrom(128)

                # Αποκωδικοποίηση και εκτύπωση του μηνύματος απάντησης
                print('Received message:
{}'.format(response.decode(encoding='utf-8')))

            except Exception as e:
                # Διαχείριση σφαλμάτων κατά τη λήψη
                print('Error receiving: ' + str(e))
                break

    def takeoff(self):
        """Στέλνει την εντολή 'takeoff' στο drone Tello."""
        self.send_command('takeoff')

    def land(self):

```

```

        """Στέλνει την εντολή 'land' στο drone Tello."""
        self.send_command('land')

# Κλάση για χειρισμό ενός swarm (ομάδας) drones Tello
class TelloSwarm:
    def __init__(self, num_drones):
        """
        Αρχικοποιεί ένα TelloSwarm για τον έλεγχο πολλών drones.
        - num_drones: Αριθμός drones στην ομάδα
        """
        self.num_drones = num_drones # Αριθμός drones στην ομάδα

        # Δημιουργία αντικειμένων Tello για κάθε drone στην ομάδα
        # Αναθέτει σε κάθε αντικείμενο Tello τις αντίστοιχες τοπικές και
        Διευθύνσεις Tello
        self.drones = [
            Tello(local_address, tello_address)
            for local_address, tello_address in zip(
                [local1_address, local2_address][:num_drones], #
                [tello1_address, tello2_address][:num_drones] #
                Διευθύνσεις των drones Tello
            )
        ]

    def takeoff(self):
        """Δίνει εντολή σε όλα τα drones της ομάδας να απογειωθούν."""
        for drone in self.drones:
            drone.takeoff()

    def land(self):
        """Δίνει εντολή σε όλα τα drones της ομάδας να προσγειωθούν."""
        for drone in self.drones:
            drone.land()

if __name__ == "__main__":

```

```
# Δημιουργία ενός TelloSwarm με 2 drones
swarm = TelloSwarm(num_drones=2)

# Εντολή σε όλη την ομάδα να απογειωθεί
swarm.takeoff()

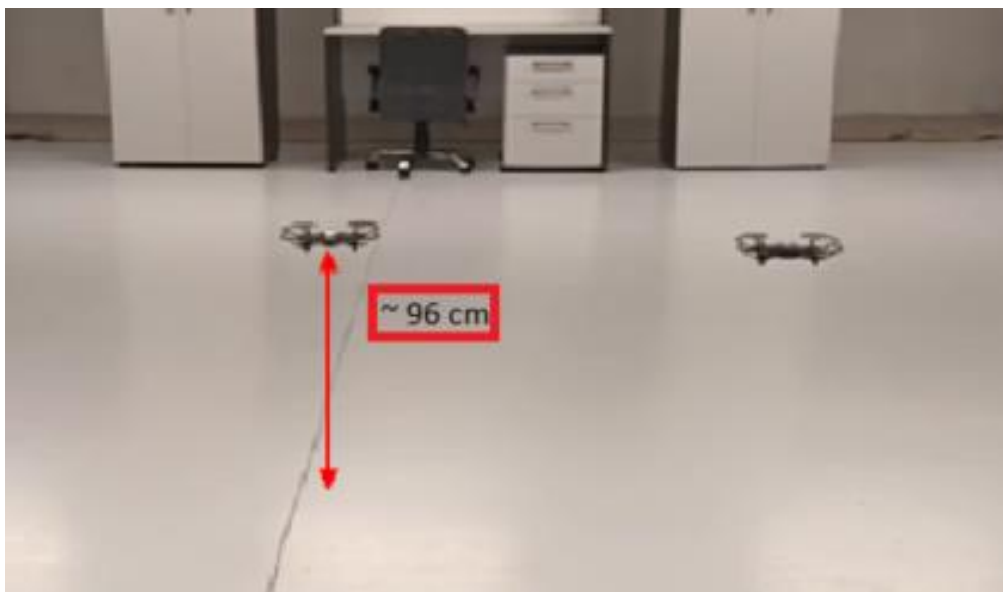
# Τα drones παραμένουν στον αέρα για 5 δευτερόλεπτα
time.sleep(5)

# Εντολή σε όλη την ομάδα να προσγειωθεί
swarm.land()
```

Ανάλυση Κώδικα:

- **Κλάση Tello (Tello Class):**
 - Η κλάση Tello αναπαριστά ένα μεμονωμένο drone και χειρίζεται την επικοινωνία με αυτό μέσω UDP.
 - Δημιουργεί ένα UDP socket, το δεσμεύει σε τοπικό port και στέλνει εντολές όπως takeoff και land.
 - Περιέχει επίσης ένα thread που ακούει συνεχώς για μηνύματα (όπως απαντήσεις από το drone).
- **Κλάση TelloSwarm:**
 - Η κλάση TelloSwarm ελέγχει πολλαπλά drones.
 - Αρχικοποιεί πολλές instances της κλάσης Tello (με βάση τον αριθμό των drones) και στέλνει εντολές σε όλη την ομάδα για απογείωση ή προσγείωση.
- **Κύριο Πρόγραμμα:**
 - Η κλάση TelloSwarm δημιουργείται με 2 drones.
 - Δίνεται εντολή στην ομάδα να απογειωθεί, να αιωρείται για 5 δευτερόλεπτα και στη συνέχεια να προσγειωθεί.

Αυτή η δομή επιτρέπει τον εύκολο έλεγχο πολλαπλών drones ταυτόχρονα χρησιμοποιώντας την κλάση TelloSwarm.



Εικόνα 22 – Αιώρηση στο 1 μέτρο σε λειτουργία σμήνους

Σε αυτή την εικόνα παρατηρείται ότι ο κώδικας εκτελείται σωστά, καθώς τα δύο drones αιωρούνται σε ύψος περίπου 1 μέτρο από το έδαφος για 5 δευτερόλεπτα. Στη συνέχεια, θα παρουσιαστούν δύο επιπλέον πειράματα, όπου αυτή τη φορά τα drones θα κινούνται στον χώρο εκτελώντας συγκεκριμένους ελιγμούς.

2ο Πείραμα

```
# Αυτό το παράδειγμα δείχνει πώς να χρησιμοποιήσετε Python για να
εκτελέσετε αποστολή πτήσης "κουτί" με drones Tello.
# Το σενάριο στέλνει εντολές σε δύο drones Tello ταυτόχρονα για να
εκτελέσουν την αποστολή.
# Χρησιμοποιεί επικοινωνία UDP για να ελέγξει τα drones και να λάβει
απαντήσεις.

# Εισαγωγή των απαραίτητων modules
import socket # Για δικτυακή επικοινωνία με τα drones
import threading # Για δημιουργία ξεχωριστού thread που ακούει μηνύματα
από τα drones
import time # Για παύσεις μεταξύ των εντολών

# IP και port των δύο drones Tello
tello1_address = ('192.168.0.100', 8889) # Διεύθυνση IP και command
```

```

port του πρώτου drone Tello
tello2_address = ('192.168.0.102', 8889) # Διεύθυνση IP και command
port του δεύτερου drone Tello

# IP και port του τοπικού υπολογιστή για κάθε drone
local1_address = ('', 9010) # Τοπικό port για τη λήψη μηνυμάτων από το
Tello #1
local2_address = ('', 9011) # Τοπικό port για τη λήψη μηνυμάτων από το
Tello #2

# Δημιουργία δύο UDP sockets για επικοινωνία με τα drones
sock1 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # Socket για
το Tello #1
sock2 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # Socket για
το Tello #2

# Δέσμευση κάθε socket στη τοπική διεύθυνση και port (για λήψη
απαντήσεων)
sock1.bind(local1_address) # Δέσμευση του socket1 στο τοπικό port
sock2.bind(local2_address) # Δέσμευση του socket2 στο τοπικό port

# Συνάρτηση για αποστολή εντολής και καθυστέρηση
def send(message, delay):
    """
    Στέλνει μια εντολή και στα δύο drones Tello και προσθέτει καθυστέρηση
    μετά την εντολή.
    - message: Η εντολή που θα σταλεί (π.χ., 'takeoff', 'forward 100')
    - delay: Πόσο χρόνο να περιμένει μετά την εντολή (σε δευτερόλεπτα)
    """
    try:
        # Αποστολή της εντολής και στα δύο drones μέσω UDP
        sock1.sendto(message.encode(), tello1_address)
        sock2.sendto(message.encode(), tello2_address)
        print("Sending message: " + message) # Εκτύπωση της εντολής για
        παρακολούθηση
    except Exception as e:
        # Αν υπάρξει σφάλμα κατά την αποστολή, εκτυπώνεται το μήνυμα

```



```

εξαίρεσης
    print("Error sending: " + str(e))

    # Καθυστέρηση για την ορισμένη διάρκεια ώστε να εκτελέσει την εντολή
το drone
    time.sleep(delay)

# Συνάρτηση για συνεχή λήψη μηνυμάτων από τα drones
def receive():
    """
    Λαμβάνει απαντήσεις από τα δύο drones Tello σε συνεχή βρόχο.
    Η συνάρτηση εκτελείται σε ξεχωριστό thread και ακούει τις απαντήσεις
από τα drones.
    """
    while True:
        try:
            # Λήψη μηνυμάτων και από τα δύο drones (μέχρι 128 bytes)
            response1, ip_address = sock1.recvfrom(128) # Ακρόαση για
μήνυμα από το Tello #1
            response2, ip_address = sock2.recvfrom(128) # Ακρόαση για
μήνυμα από το Tello #2

            # Αποκωδικοποίηση και εκτύπωση των μηνυμάτων από τα δύο
drones
            print("Received message: from Tello EDU #1: " +
response1.decode(encoding='utf-8'))
            print("Received message: from Tello EDU #2: " +
response2.decode(encoding='utf-8'))
        except Exception as e:
            # Σε περίπτωση σφάλματος, κλείνουμε τα sockets και βγαίνουμε
από τον βρόχο
            sock1.close()
            sock2.close()
            print("Error receiving: " + str(e))
            break

```

```

# Δημιουργία ξεχωριστού thread για την εκτέλεση της συνάρτησης 'receive'
στο παρασκήνιο
# Αυτό το thread θα ακούει συνεχώς για μηνύματα από τα drones
receiveThread = threading.Thread(target=receive)
receiveThread.daemon = True # Το thread ρυθμίζεται ως daemon για να
κλείσει με το πρόγραμμα
receiveThread.start() # Εκκίνηση του thread

# Παράμετροι για την αποστολή "κουτί"
box_leg_distance = 100 # Κάθε πλευρά του κουτιού θα έχει μήκος 100 cm
yaw_angle = 90 # Το Tello θα στρίψει 90 μοίρες σε κάθε γωνία
yaw_direction = "cw" # 'cw' σημαίνει περιστροφή δεξιόστροφα (yaw right)

# Βάλε και τα δύο drones σε λειτουργία εντολών (απαραίτητο για να
δεχτούν περαιτέρω εντολές)
send("command", 3)

# Δώσε εντολή και στα δύο drones να απογειωθούν
send("takeoff", 8)

# Βρόχος για να δημιουργήσει κάθε πλευρά του κουτιού
for i in range(4):
    # Εντολή και στα δύο drones να πετάξουν μπροστά για 100 cm (η κάθε
πλευρά του κουτιού)
    send("forward " + str(box_leg_distance), 4)
    # Εντολή και στα δύο drones να στρίψουν 90 μοίρες δεξιόστροφα
    send("cw " + str(yaw_angle), 3)

# Εντολή και στα δύο drones να προσγειωθούν μετά την ολοκλήρωση της
αποστολής
send("land", 5)

# Εκτύπωση μηνύματος επιτυχούς ολοκλήρωσης αποστολής
print("Mission completed successfully!")

# Κλείσιμο των δύο sockets για να τερματιστεί η επικοινωνία με τα drones
sock1.close()
sock2.close()

```

Επεξήγηση των σχολίων:

1. **Εισαγωγή modules:** Τα modules socket, threading, και time εισάγονται για επικοινωνία UDP, εκτέλεση background threads και παύσεις αντίστοιχα.
2. **Διευθύνσεις drones και τοπικού υπολογιστή:**
 - Ορίζονται οι διευθύνσεις IP και τα ports για τα drones και οι τοπικές διευθύνσεις για τη λήψη μηνυμάτων.
3. **UDP Sockets:**
 - Δημιουργούνται τα UDP sockets για την αποστολή εντολών και τη λήψη απαντήσεων από κάθε drone.
4. **Συνάρτηση αποστολής εντολών (send):**
 - Αυτή η συνάρτηση στέλνει εντολές και στα δύο drones και προσθέτει καθυστέρηση για την εκτέλεση της εντολής.
5. **Συνάρτηση λήψης απαντήσεων (receive):**
 - Αυτή η συνάρτηση ακούει για μηνύματα από τα drones σε συνεχή βρόχο και εκτυπώνει τις απαντήσεις.
6. **Παρασκηνακό Thread για ακρόαση:**
 - Δημιουργείται ένα thread που εκτελεί τη συνάρτηση λήψης σε παρασκήνιο, ενώ το κυρίως πρόγραμμα συνεχίζει.
7. **Αποστολή "κουτί":**
 - Στέλνονται εντολές απογείωσης, πτήσης σε ευθεία και περιστροφής σε κάθε γωνία για να σχηματιστεί ένα τετράγωνο.
 - Τα drones προσγειώνονται μετά την ολοκλήρωση της αποστολής.
8. **Κλείσιμο των Sockets:**
 - Κλείνουν τα sockets για να τερματιστεί η επικοινωνία με τα drones.

3^ο Πείραμα

```
# Εισαγωγή των απαραίτητων modules
# Χρησιμοποιούνται για επικοινωνία UDP, παράλληλη εκτέλεση διαδικασιών
και παύσεις.
import socket # Για την επικοινωνία με τα drones μέσω UDP
import threading # Για τη δημιουργία παράλληλων threads για ακρόαση
μηνυμάτων
import time # Για τη διαχείριση παύσεων μεταξύ εντολών

# IP και port των drones Tello
tello1_address = ('192.168.0.100', 8889) # Διεύθυνση και port του
πρώτου drone Tello
tello2_address = ('192.168.0.102', 8889) # Διεύθυνση και port του
δεύτερου drone Tello

# IP και port του τοπικού υπολογιστή (για λήψη μηνυμάτων από τα drones)
local1_address = ('', 9010) # Τοπική διεύθυνση και port για το Tello #1
local2_address = ('', 9011) # Τοπική διεύθυνση και port για το Tello #2

# Δημιουργία σύνδεσης UDP για αποστολή εντολών
sock1 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # Socket για
το Tello #1
sock2 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # Socket για
το Tello #2

# Δέσμευση των sockets στις τοπικές διευθύνσεις και ports
sock1.bind(local1_address) # Δέσμευση του πρώτου socket
sock2.bind(local2_address) # Δέσμευση του δεύτερου socket

# Συνάρτηση για αποστολή εντολής και καθυστέρηση πριν από την επόμενη
ενέργεια
def send(message, delay):
```

```

"""
    Στέλνει μια εντολή στα δύο drones Tello και καθυστερεί πριν από την
    επόμενη ενέργεια.
    - message: Η εντολή που θα σταλεί (π.χ., 'takeoff', 'land', 'flip
    b')
    - delay: Ο χρόνος καθυστέρησης σε δευτερόλεπτα πριν από την επόμενη
    εντολή.
"""
try:
    # Αποστολή της εντολής και στα δύο drones
    sock1.sendto(message.encode(), tello1_address) # Αποστολή στο
    πρώτο drone
    sock2.sendto(message.encode(), tello2_address) # Αποστολή στο
    δεύτερο drone
    print("Sending message: " + message) # Εκτύπωση της εντολής για
    έλεγχο
except Exception as e:
    # Αν υπάρξει κάποιο σφάλμα κατά την αποστολή της εντολής,
    εμφανίζεται το σφάλμα
    print("Error sending: " + str(e))

# Καθυστέρηση για τη διάρκεια που ορίζεται από τον χρήστη (delay)
time.sleep(delay)

# Συνάρτηση για συνεχή λήψη μηνυμάτων από τα drones
def receive():
    """
    Λαμβάνει απαντήσεις από τα δύο drones σε συνεχή βρόχο.
    Εκτελείται σε ξεχωριστό thread και ακούει τα μηνύματα από τα drones.
    """
    while True:
        try:
            # Λήψη μηνυμάτων από το Tello #1 και Tello #2
            response1, ip_address = sock1.recvfrom(128) # Λήψη
            μηνύματος από το πρώτο drone
            response2, ip_address = sock2.recvfrom(128) # Λήψη
            μηνύματος από το δεύτερο drone

```

```

        # Εκτύπωση των αποκωδικοποιημένων μηνυμάτων
        print("Received message: from Tello EDU #1: " +
response1.decode(encoding='utf-8'))
        print("Received message: from Tello EDU #2: " +
response2.decode(encoding='utf-8'))
    except Exception as e:
        # Αν υπάρξει κάποιο σφάλμα, κλείνουμε τα sockets και
τερματίζουμε τον βρόχο
        sock1.close()
        sock2.close()
        print("Error receiving: " + str(e))
        break

# Δημιουργία και εκκίνηση ενός thread για συνεχή ακρόαση των μηνυμάτων
από τα drones
# Το thread αυτό θα εκτελεί τη συνάρτηση receive σε παρασκήνιο.
receiveThread = threading.Thread(target=receive)
receiveThread.daemon = True # Ορίζεται ως daemon thread για να
σταματήσει μαζί με το πρόγραμμα
receiveThread.start() # Εκκίνηση του thread

# Ορισμός παραμέτρων για την αποστολή "κουτί" (box mission)
box_leg_distance = 100 # Κάθε πλευρά του κουτιού θα έχει μήκος 100 cm
yaw_angle = 90 # Τα drones θα στρίψουν 90 μοίρες σε κάθε γωνία
yaw_direction = "cw" # 'cw' σημαίνει δεξιόστροφη περιστροφή (clockwise)

# Βάλτε τα drones σε "λειτουργία εντολών" για να δεχθούν τις εντολές
send("command", 3)

# Απογείωση και των δύο drones
send("takeoff", 8)

# Βρόχος για να εκτελεστεί κάθε πλευρά του κουτιού και μια περιστροφή
(flip)
for i in range(2): # Εκτέλεση για δύο πλευρές
    # Εντολή στα drones να πετάξουν μπροστά για 100 cm
    send("forward " + str(box_leg_distance), 4)
    # Εντολή για flip (π.χ. 'b' για flip προς τα πίσω)

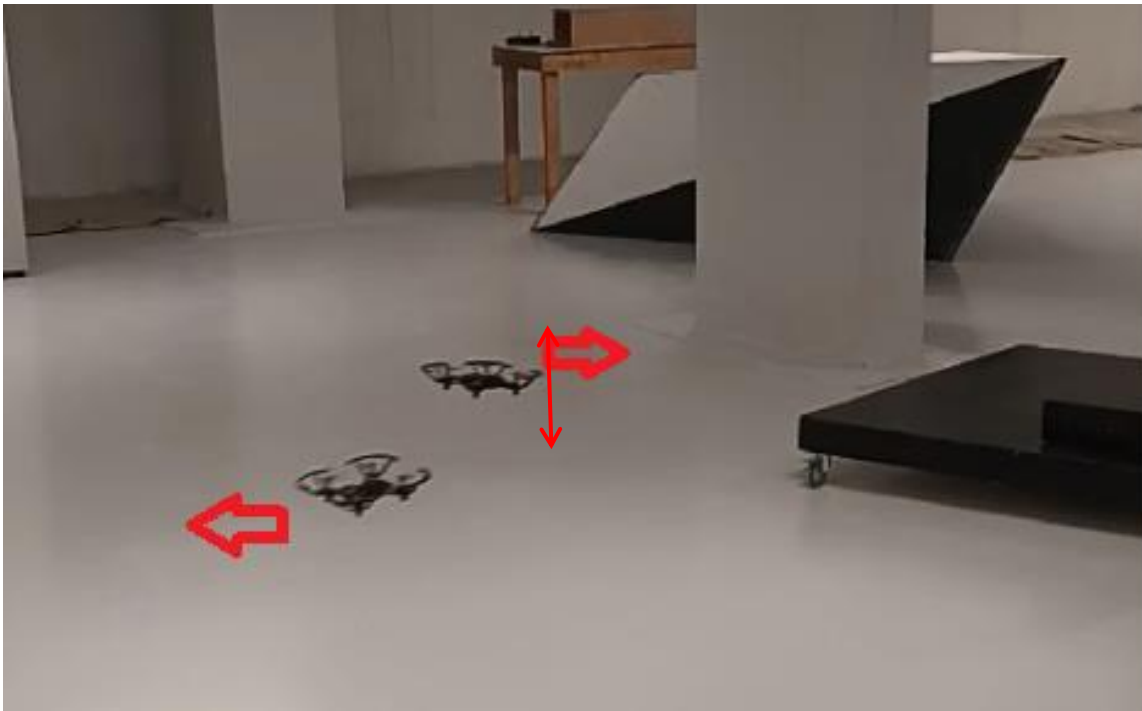
```

```
send("flip " + "b", 2) # Εκτέλεση flip προς τα πίσω
# Εντολή να στρίψουν 90 μοίρες δεξιόστροφα
send("cw " + str(yaw_angle), 3)

# Εντολή προσγείωσης και των δύο drones μετά την ολοκλήρωση της
αποστολής
send("land", 5)

# Εκτύπωση μηνύματος επιτυχούς ολοκλήρωσης της αποστολής
print("Mission completed successfully!")

# Κλείσιμο των sockets για να τερματιστεί η επικοινωνία με τα drones
sock1.close() # Κλείσιμο του πρώτου socket
sock2.close() # Κλείσιμο του δεύτερου socket
```



Εικόνα 23 – Πτήση σε ‘κουτί’ και αλλαγή κατεύθυνσης (ΦΑΣΗ 1)



Εικόνα 24 – Πτήση και flip (ΦΑΣΗ 2)

Ο κώδικας αυτός επιτρέπει σε δύο drones **Tello** να εκτελέσουν μια αποστολή με τη μορφή "κουτιού", ενώ παράλληλα εκτελούν μια περιστροφή και ένα flip (ανατροπή) κατά την πτήση τους. Χρησιμοποιεί το **UDP πρωτόκολλο επικοινωνίας** για την αποστολή εντολών στα drones και τη λήψη απαντήσεων από αυτά. Τα drones ελέγχονται ταυτόχρονα και εκτελούν την ίδια αποστολή σε συγχρονισμό.

Εκτέλεση της αποστολής "κουτί"

Η αποστολή "κουτί" εκτελείται σε έναν βρόχο. Κάθε drone:

- **Πετάει μπροστά** για 100 cm (`send("forward 100", 4)`).
- **Εκτελεί μια ανατροπή (flip)** προς τα πίσω με την εντολή `send("flip b", 2)`.
- **Στρέφεται δεξιόστροφα** κατά 90 μοίρες με την εντολή `send("cw 90", 3)`.

Αυτό επαναλαμβάνεται για δύο πλευρές του "κουτιού".

Προσγείωση των drones

Αφού ολοκληρώσουν την αποστολή, η εντολή `send("land", 5)` στέλνει τα drones για προσγείωση, και το πρόγραμμα περιμένει 5 δευτερόλεπτα για να ολοκληρωθεί η διαδικασία.

Κλείσιμο της σύνδεσης

Τέλος, το πρόγραμμα κλείνει τα **sockets** για να τερματίσει την επικοινωνία με τα drones. Αυτό είναι απαραίτητο για να εξασφαλιστεί ότι δεν παραμένουν ανοιχτές συνδέσεις που μπορεί να προκαλέσουν προβλήματα.

ΑΠΟΤΕΛΕΣΜΑΤΑ

Το έργο ανέδειξε την επιτυχή εφαρμογή ελέγχου και συγχρονισμού πολλαπλών drones Tello EDU μέσω δικτύου Wi-Fi, με τη χρήση Python. Τα βασικά αποτελέσματα περιλαμβάνουν:

1. **Επιτυχής Δικτύωση και Επικοινωνία:** Δημιουργήθηκε ένα σταθερό και αξιόπιστο δίκτυο, το οποίο επέτρεψε την απρόσκοπτη επικοινωνία και τον συντονισμένο έλεγχο των drones.
2. **Ανάπτυξη Αποτελεσματικού Πλαισίου Ελέγχου:** Ένα πλήρες πλαίσιο ελέγχου αναπτύχθηκε σε γλώσσα Python, επιτρέποντας τον συγχρονισμό κινήσεων των drones με υψηλή ακρίβεια. Αυτό το πλαίσιο επέτρεψε την εκτέλεση σύνθετων σχηματισμών και ελιγμών με απόλυτο συντονισμό.
3. **Επιτυχία στον Συντονισμό και την Εκτέλεση Πτήσεων:** Τα drones μπόρεσαν να εκτελέσουν μια σειρά από σύνθετες πτητικές κινήσεις με ακρίβεια και συγχρονισμό, δείχνοντας ότι η πλατφόρμα Tello EDU μπορεί να λειτουργήσει ως μια προσιτή και ευέλικτη βάση για εκπαιδευτικά και ερευνητικά έργα στον τομέα της ρομποτικής σμήνους.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Το έργο ανέδειξε με επιτυχία την ικανότητα ελέγχου και συγχρονισμού πολλαπλών drones Tello EDU μέσω ενός δικτύου Wi-Fi, αξιοποιώντας την Python. Τα drones εκτέλεσαν συντονισμένες κινήσεις με ακρίβεια, διατηρώντας σχηματισμό και ανταποκρινόμενα με επιτυχία στις εντολές. Η έρευνα κατέδειξε ότι η πλατφόρμα Tello EDU είναι ιδιαίτερα κατάλληλη για εκπαιδευτική χρήση, προσφέροντας μια αξιόπιστη και ευέλικτη βάση για πειραματισμό στον τομέα της ρομποτικής σμήνους.

Η επιτυχημένη υλοποίηση του περιβάλλοντος σμήνους παρέχει πολύτιμες γνώσεις για τον έλεγχο πολλαπλών πρακτόρων, με δυνατότητες εφαρμογής σε τομείς όπως ο συντονισμός αυτόνομων οχημάτων, η συλλογική νοημοσύνη και η εκπαιδευτική ρομποτική. Το έργο καταδεικνύει επίσης τη δυνατότητα βελτίωσης της ακρίβειας και της απόκρισης των drones μέσω αλγορίθμων συγχρονισμού και ανίχνευσης θέσης.

Στο μέλλον, η εργασία αυτή θα μπορούσε να επεκταθεί με τη χρήση αισθητήρων, όπως κάμερες και υπερήχων, που θα επιτρέπουν την αυτόματη αποφυγή εμποδίων και την ακριβέστερη αλληλεπίδραση με το περιβάλλον. Επιπλέον, η ενσωμάτωση αλγορίθμων τεχνητής νοημοσύνης και μηχανικής μάθησης μπορεί να συμβάλει στην ανάλυση και βελτίωση του συντονισμού σε δυναμικά περιβάλλοντα, καθώς και στην ανάπτυξη πιο αυτόνομων και ευφυών σχηματισμών drones. Αυτή η προοπτική μπορεί να οδηγήσει σε εφαρμογές όπως η ανίχνευση και χαρτογράφηση περιοχών, η παρακολούθηση και διάσωση σε καταστάσεις έκτακτης ανάγκης, καθώς και η επιτήρηση μεγάλων περιοχών με ελάχιστη ανθρώπινη παρέμβαση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Autonomous rescue robot swarms for first responders.* (2005). Retrieved from https://www.researchgate.net/publication/4170387_Autonomous_rescue_robot_swarms_for_first_responders
- Cavadas, J. A. (2024, 9 30). *Using Drone Tello Edu for educational purposes.* Retrieved from <https://upcommons.upc.edu/bitstream/handle/2117/173834/memoria.pdf>
- Erkin Bahceci, E. S. (2005, 7). *Evolving aggregation behaviors for swarm robotic systems: a systematic case study.* Retrieved from http://www.weissgerhard.info/publications/MATES2012_RS_W_N.pdf
- Houxiang Zhang, W. W. (2006, 11). *A Novel Reconfigurable Robot for Urban Search and Rescue.* Retrieved from https://www.researchgate.net/publication/221785993_A_Novel_Reconfigurable_Robot_for_Urban_Search_and_Rescue
- Jorge Cortés, S. M. (2004, 4). *Coverage Control for Mobile Sensing Networks.* Retrieved from https://www.researchgate.net/publication/3299441_Coverage_Control_for_Mobile_Sensing_Networks
- Packet Sender.* (2024, 9 30). Retrieved from https://en.wikipedia.org/wiki/Packet_Sender#External_links
- Packet Sender documentation.* (2024, 9 30). Retrieved from <https://packetsender.com/documentation>
- Sahin, E. (2004). *Swarm Robotics: From Sources of Inspiration to Domains of Application.* Retrieved from https://www.researchgate.net/publication/225703852_Swarm_Robotics_From_Sources_of_Inspiration_to_Domains_of_Application
- Simon Garnier, J. G. (2007, 7 17). *The biological principles of swarm intelligence.* Retrieved from https://www.researchgate.net/publication/220058931_The_biological_principles_of_swarm_intelligence
- Spezzano, G. (2024, 9 30). *Swarm Robotics.* Retrieved from https://mdpi-res.com/bookfiles/book/1294/Swarm_Robotics.pdf?v=1726362069
- Ying TAN*, Z.-y. Z. (2013, 3 2). *Research Advance in Swarm Robotics.* Retrieved 9 30, 2024, from https://www.sciencedirect.com/science/article/pii/S221491471300024X?ref=pdf_download&fr=RR-2&rr=8cb1e2c5cd2fee77