ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗ ΡΟΜΠΟΤΙΚΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ, ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΜΟΝΤΕΛΑ VISION-LANGUAGE-ACTION ΓΙΑ ΑΥΤΟΝΟΜΟ ΕΛΕΓΧΟ DRONES

Πτυχιακή Εργασία του

Ιωάννη Πέτσα (136)

Επιβλέπων: Σταύρος Βολογιαννίδης, Επίκουρος Καθηγητής

ΣΕΡΡΕΣ, ΙΑΝΟΥΡΙΟΣ 2026

Υπεύθυνη Δήλωση *: Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών του Διεθνούς Πανεπιστημίου της Ελλάδας.*

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με την ανάπτυξη ενός ολοκληρωμένου συστήματος αυτόνομης πτήσης drone, το οποίο βασίζεται στη συνεργασία ενός Vision-Language-Action (VLA) μοντέλου με το περιβάλλον του ROS και τον αυτόματο πιλότο PX4. Στόχος της εργασίας είναι η δημιουργία ενός πλαισίου όπου το drone μπορεί να αντιλαμβάνεται το περιβάλλον του μέσω κάμερας, να κατανοεί εντολές σε φυσική γλώσσα και να εκτελεί αντίστοιχες κινήσεις χωρίς ανθρώπινη παρέμβαση.

Για την επίτευξη αυτού του στόχου, αναπτύχθηκε ένας ROS bridge που λειτουργεί ως ενδιάμεσος κόμβος επικοινωνίας μεταξύ του VLA server και του drone. Ο κόμβος αυτός λαμβάνει τα δεδομένα εικόνας από το σύστημα πλοήγησης, τα αποστέλλει στον server όπου λειτουργεί το VLA μοντέλο, και στη συνέχεια λαμβάνει πίσω τις υπολογισμένες εντολές πτήσης. Οι εντολές αυτές μετατρέπονται αυτόματα σε σήματα ελέγχου ταχύτητας μέσω MAVROS, επιτρέποντας την επικοινωνία με τον PX4 και την εκτέλεση των εντολών σε πραγματικό χρόνο.

Μέσω της συγκεκριμένης αρχιτεκτονικής επιτεύχθηκε η αυτόνομη πτήση του drone μέσα από καθορισμένες πύλες, σύμφωνα με προκαθορισμένες εντολές που έχουν ενσωματωθεί στον κώδικα, όπως η εντολή "Fly through the gate". Το σύστημα κατάφερε να επεξεργάζεται τα δεδομένα εικόνας, να αντιστοιχεί τις εντολές σε ενέργειες και να ελέγχει το drone με σταθερό και προβλέψιμο τρόπο.

Η εργασία αυτή αποδεικνύει στην πράξη τη δυνατότητα ενσωμάτωσης προηγμένων μοντέλων τεχνητής νοημοσύνης σε συστήματα μη επανδρωμένων αεροσκαφών, προσφέροντας ένα σημαντικό βήμα προς την κατεύθυνση της φυσικής αλληλεπίδρασης ανθρώπου και μηχανής. Το αποτέλεσμα είναι ένα ευέλικτο και επεκτάσιμο σύστημα, το οποίο μπορεί να αποτελέσει τη βάση για μελλοντικές εφαρμογές αυτόνομης πλοήγησης και ελέγχου UAV.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω την ειλικρινή μου ευγνωμοσύνη προς τον επιβλέποντα καθηγητή, κ. Σταύρο Βολογιαννίδη, για την καθοδήγηση, την υπομονή και τις εύστοχες παρατηρήσεις του καθ' όλη τη διάρκεια της πτυχιακής εργασίας. Η επιστημονική του κατάρτιση και η συνεχής διάθεση για συνεργασία συνέβαλαν ουσιαστικά στην ολοκλήρωση της παρούσας μελέτης.

Ευχαριστώ επίσης την εξεταστική επιτροπή, τον κ. Ιωάννη Βουρβουλάκη και τον κ. Αθανάσιο Νικολαΐδη, για τον χρόνο και τις χρήσιμες επισημάνσεις τους, καθώς και το Διεθνές Πανεπιστήμιο της Ελλάδος για την ευκαιρία συμμετοχής στο μεταπτυχιακό πρόγραμμα «ΠΜΣ Ρομποτικής». Η αλληλεπίδραση με τους διδάσκοντες και τους συμφοιτητές αποτέλεσε πηγή γνώσης και έμπνευσης.

Τέλος, θα ήθελα να ευχαριστήσω θερμά τους γονείς μου, Γεώργιο Πέτσα και Ελένη Τζιανοπούλου, για τη διαρκή ηθική τους υποστήριξη, την ενθάρρυνση και την εμπιστοσύνη τους καθ' όλη τη διάρκεια των σπουδών μου, καθώς και τη μέλλουσα σύζυγό μου, Anna Grebennikova, για την αφοσίωση και τη σταθερή παρουσία της σε κάθε στάδιο αυτής της πορείας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....3

ΕΥΧΑΡΙΣΤΙΕΣ.....4

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή.....6

1.1 Μη επανδρωμένα αεροσκάφη.....12

1.2 Ιστορικό πλαίσιο.....13

1.3 Σύχρονη τεχνολογική πραγματικότητα.....15

1.4 Αντικείμενο και στόχος εργασίας.....17

ΚΕΦΑΛΑΙΟ 2: Βασικά εργαλεία ανάπτυξης και ελέγχου drone.....18

2.1 Εισαγωγή.....18

2.2 Διάφορα λειτουργικά συστήματα.....19

2.2.1 Robot operating system.....20

2.2.2 Εφαρμογές του ROS.....20

2.2.3 Πλεονεκτήματα από τη χρήση του ROS.....21

2.2.4 Τεχνολογία ROS.....22

2.3 Gazebo.....22

2.3.1 Προσομοίωση.....24

2.4 Rviz.....25

ΚΕΦΑΛΑΙΟ 3: Px4 structure.....27

3.1 Εισαγωγή στο PX4.....27

3.2 Αρχιτεκτονική PX4.....28

3.2.1 Flight stack.....28

3.2.2 Middleware.....30

3.2.3 Περιβάλλον εκτέλεσης.....31

3.3 uORB.....31

3.4 Mavlink 36
3.5 Δομή λογισμικού 37
3.5.1 System commands (systemcdms) 39
3.5.2 Communication 40
3.5.3 Εκλεκτές 41
3.5.4 Drivers 42
ΚΕΦΑΛΑΙΟ 4: Visual language action system 45
4.1 Μοντέλα βάση γλώσσας 45
4.1.1 Θεμελιώδη μοντέλα όρασης (VFMs) 49
4.1.2 Vision language model 52
4.1.3 Ενσωματωμένα μοντέλα VLA 54
4.2 Vision language action 56
4.2.1 Γενική αρχιτεκτονική VLA 57
4.2.2 Action tokenization και output spaces 58
4.3 Ταξινόμηση και επισκόπηση μοντέλων VLA 58
4.3.1 Ταξινόμηση με βάση την αναπαράσταση δράσης 58
4.3.2 Ενσωματωμένοι τομείς ρομποτικής 59
4.3.3 Τυπικά καθεστώτα εκπαίδευσης και δεδομένων 60
4.4 Visual language action models για μη επανδρωμένα αεροσκάφη 61
4.4.1 RaceVLA 61
4.4.2 Cognitive task solving και reasoning στα UAVs 64
4.4.3 UAV-VLA 67
4.4.4 Κοινά μοτίβα στα συστήματα UAV-VLA 68
4.5 LLM vs VLA 68
4.5.1 Χειρισμός με καθαρά LLM – driven drone agents 69
4.5.2 UAV – VLA 70

ΚΕΦΑΛΑΙΟ 5: Υλοποίηση και ενσωμάτωση LLaMA/VLA στο περιβάλλον ROS–PX4 73

5.1 Προετοιμασία υπολογιστικού περιβάλλοντος 73

5.1.1 Ρύθμιση των locales (απαραίτητο για το ros) 73

5.2 Εγκατάσταση ros noetic 74

5.2.1 Προσθήκη του apt repository του ros 74

5.2.2 Εγκατάσταση ros noetic 75

5.2.3 Εγκατάσταση ros εργαλείων 75

5.3 Εγκατάσταση εργαλείων px4 και εξαρτήσεων 75

5.3.1 Κατέβασμα του αυτόματου πιλότου px4 75

5.4 Εγκατάσταση gazebo και rviz 76

5.5 Εγκατάσταση px4 toolchain και build του sitl 76

5.6 Εγκατάσταση mavros και γεωγραφικών δεδομένων 78

5.7 Κατέβασμα RaceVLA και LLaMA 79

5.7.1 Εγκατάσταση εργαλείων huggingFace 79

5.7.2 Κατέβασμα RaceVLA 79

5.7.3 Κατέβασμα LLaMA-2-7B 80

5.8 Προσθήκη custom scripts στο px4 project 80

5.9 Προσθήκη custom sris model για vla 81

5.10 Προσθήκη custom world file 81

5.11 Εκκίνηση της προσομοίωσης 81

5.11.1 Εκκίνηση του Gazebo World 81

5.11.2 Εκκίνηση MAVROS 82

5.12 Προετοιμασία Περιβάλλοντος για VLA Server 83

5.12.1 Δημιουργία Python Virtual Environment 83

5.12.2 Εγκατάσταση PyTorch (CUDA 11.8) 83

5.12.3 Εγκατάσταση Transformers + TIMM 84

5.12.4 Εγκατάσταση Βιβλιοθηκών VLA Server 84
5.13 Εκκίνηση του Vision–Language–Action Server 84
5.14 Λειτουργία του ολοκληρωμένου συστήματος 87
5.15 Ροή λειτουργίας και επικοινωνία κόμβων 88
5.16 Αποτελέσματα και επιτυχία του συστήματος 88
5.17 Περιορισμοί και προοπτικές βελτίωσης 89
ΣΥΜΠΕΡΑΣΜΑΤΑ 90
ΒΙΒΛΙΟΓΡΑΦΙΑ 92

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

Εικόνα 1.1 UAV Airboxer [2] 12
Εικόνα 1.2 Tesla ράδιο [7] 13
Εικόνα 1.3 Kettering Bug [9] 14
Εικόνα 1.4 UAV Ryan Model 147 [11] 14
Εικόνα 1.5 Drone με lidar yellow scan [13] 15
Εικόνα 2.1 Ros logo [16]. 18
Εικόνα 2.2 Gazebo logo [21] 23
Εικόνα 2.3 Επικοινωνία μεταξύ κόμβων [23] 26
Εικόνα 3.1 PX4 quadcopter σε περιβάλλον εξομοίωσης gazebo 27
Εικόνα 3.2 Διάγραμμα του flight stack [25] 29
Εικόνα 3.3 Ελεγκτής θέσης θέσης [26] 30
Εικόνα 3.4 uORB σχηματικό [28] 32
Εικόνα 3.5 Δομή λογισμικού [34] 38
Εικόνα 3.6 Config arguments [34] 39
Εικόνα 3.7 uORB arguments [34] 41
Εικόνα 3.8 Navigator arguments [34] 42
Εικόνα 3.9 GPS arguments [34] 43
Εικόνα 4.1 Οπτικοποίηση των action tokens σε μία μεμονωμένη ενσωματωμένη εργασία [35] 47

Εικόνα 4.2 Εικόνα 4.2 Χρονοδιάγραμμα εξέλιξης των foundation models, των VLA μοντέλων και των data sources. Το σχήμα σε μορφή "U" αποτυπώνει το πώς η αυξανόμενη ανάπτυξη των VLA μοντέλων υποστηρίζεται από την πρόοδο τόσο των foundation models όσο και των δεδομένων [35] 48
Εικόνα 4.3 Ένα διάγραμμα Venn απεικονίζει τις σχέσεις μεταξύ των βασικών πεδίων της τεχνητής νοημοσύνης. Τα VLA μοντέλα βρίσκονται στο σημείο τομής της digital AI, του hardware και της ρομποτικής, αποτελώντας βασικό υποπεδίο της Embodied AI και κρίσιμο τομέα στην πορεία προς την AGI [35].....56
Εικόνα 4.4 Η γενική αρχιτεκτονική των μοντέλων VLA. Παρουσιάζονται τρεις αντιπροσωπευτικές μέθοδοι για την πρόβλεψη δράσης. Τα σχετικά στοιχεία παρουσιάζονται σε διακεκομμένα πλαίσια [37].....57
Εικόνα 4.5 Το RaceVLA είναι το πρώτο VLA μοντέλο που έχει αναπτυχθεί αποκλειστικά για αγωνιστικά drones. Επεξεργάζεται ροές βίντεο FPV σε συνδυασμό με εντολές φυσικής γλώσσας και παράγει ως έξοδο διανύσματα ταχυτήτων (Vx, Vy, Vz) και γωνιακή ταχύτητα yaw (ω). Με αυτόν τον σχεδιασμό, το σύστημα μπορεί να εκτελεί αυτοδύναμα ένα ευρύ φάσμα πτητικών εργασιών, περιλαμβάνοντας την πλοήγηση σε άγνωστα περιβάλλοντα και νέα σενάρια. Αξιοποιώντας ένα ειδικά διαμορφωμένο σύνολο εκπαίδευσης, το RaceVLA παρουσιάζει ισχυρή ικανότητα γενίκευσης [41].....62
Εικόνα 4.6 Αρχιτεκτονική RaceVLA [41].....63
Εικόνα 4.7 Το CognitiveDrone είναι ένα VLA σύστημα για UAVs που δημιουργεί ομαλές τετραδιάστατες εντολές ελέγχου, βασισμένες σε FPV οπτικά δεδομένα και οδηγίες φυσικής γλώσσας. Το μοντέλο συνδυάζει έναν πυρήνα VLA 7B παραμέτρων, εκπαιδευμένο σε ένα εκτενές ανοικτό σύνολο δεδομένων γνωστικών εργασιών — όπως συλλογισμό, αναγνώριση ανθρώπων και κατανόηση συμβόλων — με ένα πρόσθετο module λογικής επίσης 7B παραμέτρων, το οποίο βελτιώνει και αποσαφηνίζει τις εντολές της αποστολής. Το σύστημα αξιολογείται στο CognitiveDroneBench, το πρώτο benchmark που έχει σχεδιαστεί για VLA μοντέλα με γνωστικό προσανατολισμό σε UAVs. Το drone καλείται να πλοηγηθεί σε μια διαδρομή με πύλες, επιλέγοντας τη σωστή πύλη μέσω της επίλυσης των αντίστοιχων γνωστικών εργασιών. Οι δημιουργοί παρέχουν πλήρως ανοικτό κώδικα για το dataset, το περιβάλλον αξιολόγησης, τα βάρη των μοντέλων και τον κώδικα εκπαίδευσης και inference [42].....64
Εικόνα 4.8 Αρχιτεκτονική CognitiveDrone [42] 66

Εικόνα 4.9 Παραδείγματα προετοιμασμένων εργασιών στο dataset, σχεδιασμένων ώστε τα VLA μοντέλα να επιλύουν γνωστικά καθήκοντα προσαρμοσμένα για UAVs [42]....67
Εικόνα 4.10 LLM – driven drone agent [44] 69
Εικόνα 4.11 Σύστημα UAV-VLA [45] 71
Εικόνα 5.1 Προετοιμασία περιβάλλοντος....73
Εικόνα 5.2 Προετοιμασία περιβάλλοντος....74
Εικόνα 5.21 Προετοιμασία περιβάλλοντος....74
Εικόνα 5.22 Εγκατάσταση Ros....75
Εικόνα 5.23 Εγκατάσταση Ros....75
Εικόνα 5.24 Εγκατάσταση χρήσιμων εργαλείων....75
Εικόνα 5.3 Εγκατάσταση χρήσιμων εργαλείων....76
Εικόνα 5.31 Αυτόματος πιλότος PX4 [46] 76
Εικόνα 5.4 Εγκατάσταση χρήσιμων εργαλείων....76
Εικόνα 5.5 Προετοιμασία αυτόματου πιλότου 77
Εικόνα 5.51 Δοκιμαστικό άνοιγμα....77
Εικόνα 5.52 Δοκιμαστικό άνοιγμα....77
Εικόνα 5.53 Επίλυση σφάλματος....77
Εικόνα 5.54 Έναρξη gazebo με PX4....78
Εικόνα 5.55 Έναρξη gazebo με PX4....78
Εικόνα 5.6 Εγκατάσταση χρήσιμων εργαλείων....79
Εικόνα 5.7 Εγκατάσταση χρήσιμων εργαλείων....79
Εικόνα 5.71 Hugging face login....79
Εικόνα 5.72 Κατέβασμα RaceVLA [47]....80
Εικόνα 5.73 Κατέβασμα LLama [48] 80
Εικόνα 5.8 Δημιουργία executable αρχείων 81
Εικόνα 5.9 Προσθήκη επεξεργασμένου μοντέλου....81
Εικόνα 5.10 Προσθήκη επεξεργασμένου κόσμου 81
Εικόνα 5.11 Έλεγχος topic list....82
Εικόνα 5.111 Έλεγχος mavros....82
Εικόνα 5.112 Εγκατάσταση εργαλείων....83
Εικόνα 5.123 Nvidia smi....83
Εικόνα 5.1221 Εγκατάσταση torch....83
Εικόνα 5.1222 Έλεγχος torch....84

Εικόνα 5.1223 Έλεγχος torch.....84
Εικόνα 5.13 Ξεκίνημα server.....85
Εικόνα 5.131 Ξεκίνημα PX4.....85
Εικόνα 5.132 Ξεκίνημα mavros.....85
Εικόνα 5.133 Ξεκίνημα drone.py.....86
Εικόνα 5.134 Αρχικοποίηση κόσμου.....86
Εικόνα 5.135 Πτήση μέσω πυλών.....87
Εικόνα 5.136 Πτήση μέσω πυλών.....87

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

1.1 Μη επανδρωμένα αεροσκάφη

Τα μη επανδρωμένα αεροσκάφη έχουν γίνει μία από τις πιο βασικές εξελισσόμενες τεχνολογίες του 21ου αιώνα. Αρχικά είχαν σχεδιαστεί και χρησιμοποιηθεί κυρίως για στρατιωτικές εφαρμογές επιτήρησης ή αναγνώρισης, ενώ σήμερα αποτελούν ένα κύριο εργαλείο διαφόρων εφαρμογών. Για παράδειγμα, στον τομέα της γεωργίας διευκολύνουν τη χαρτογράφηση των καλλιεργειών, την ανίχνευση παθήσεων στα φυτά, αλλά και τον ελεγχόμενο ψεκασμό [1] . Στον τομέα της επιτήρησης περιοχών και υποδομών, διευκολύνουν τη συλλογή αεροφωτογραφιών υψηλής ανάλυσης, την επιθεώρηση δύσκολων σημείων και την υποστήριξη διαδικασιών ασφαλείας . Επιπλέον, στον τομέα της αεροφωτογράφησης και κινηματογράφησης, τα drones έχουν λειτουργήσει ως εργαλείο εύκολης πρόσβασης σε αεροφωτογραφικά και κινηματογραφικά δεδομένα για ερευνητές, ιδιώτες και παραγωγούς. Στις ανθρωπιστικές αποστολές και τη μεταφορά αγαθών, η ευελιξία και η ικανότητα πτήσης χωρίς πιλότο καθιστούν τα drones μια εξαιρετική λύση .

Εικόνα 1.1 UAV Airboxer [2].

Η ευκολία χειρισμού τους η βελτιστοποίηση του κόστους κατασκευής και λειτουργίας, καθώς και η βελτίωση των επιδόσεων, όπως η αυξημένη διάρκεια πτήσης, η καλύτερη σταθερότητα και η αυτονομία, έχουν καταστήσει τα drones προσβάσιμα πέρα από

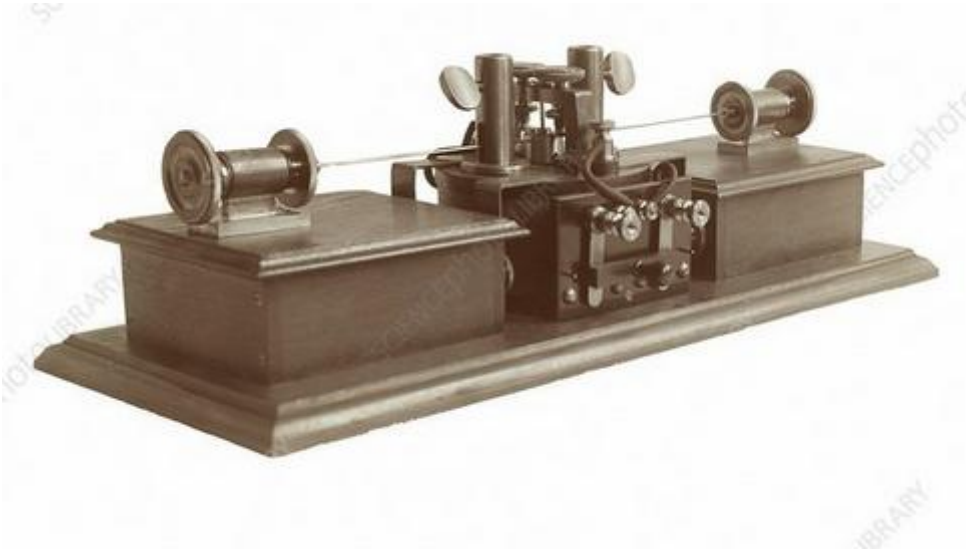
εξειδικευμένα στρατιωτικά ή ερευνητικά περιβάλλοντα, σε ερευνητικά κέντρα, επιχειρήσεις και ιδιώτες.

Η ταχεία αυτή πρόοδος των drones οφείλεται στη συνεργασία ποικίλων τεχνολογιών, όπως η ρομποτική, η τεχνητή νοημοσύνη, η υπολογιστική όραση και τα προηγμένα συστήματα πλοήγησης. Αισθητήρες υψηλής ακρίβειας, όπως το lidar, το GPS, το IMU και οι κάμερες χρώματος και βάθους, όπου παίζουν καθοριστικό ρόλο, επιτρέποντας στα σύγχρονα drones να αντιλαμβάνονται το περιβάλλον τους και να λαμβάνουν κρίσιμες αποφάσεις πτήσης με ελάχιστη ή μηδενική ανθρώπινη παρέμβαση [3]. Επίσης, σημαντικό ρόλο στον τομέα της αυτονομίας έχει η ανάπτυξη αλγορίθμων sensor fusion, SLAM και αυτόνομης πλοήγησης εντός περιβαλλόντων όπου το GPS είναι αναξιόπιστο [4].

1.2 Ιστορικό πλαίσιο

Η έννοια των μη επανδρωμένων πτήσεων δεν είναι καινούργια, καθώς οι ρίζες της ανάγονται σχεδόν δύο αιώνες πριν. Από το 1849, η Αυστριακή Πολεμική Αεροπορία χρησιμοποίησε μη επανδρωμένα αερόστατα με εκρηκτικά για να επιτεθεί στη Βενετία, αποτελώντας μία από τις πρώτες καταγεγραμμένες προσπάθειες απομακρυσμένου ελέγχου πτήσης [5].

Το 1898, ο Νίκολα Τέσλα παρουσίασε στη Διεθνή Έκθεση Ηλεκτρισμού της Νέας Υόρκης ένα ραδιοελεγχόμενο σκάφος, αποδεικνύοντας πρακτικά τη δυνατότητα καθοδήγησης μηχανής μέσω ηλεκτρομαγνητικών κυμάτων, επίτευγμα που θεωρείται θεμέλιο της σύγχρονης ραδιοελεγχόμενης πλοήγησης [6] .



Εικόνα 1.2 Tesla ράδιο [7].

Κατά τη διάρκεια του Α' Παγκοσμίου Πολέμου, αναπτύχθηκαν τα πρώτα τηλεκατευθυνόμενα αεροσκάφη, όπως το Kettering Bug, το οποίο μπορούσε να εκτελέσει προγραμματισμένη πτήση χωρίς πιλότο. [8].

Εικόνα 1.3 Kettering Bug [9].

Στον Β' Παγκόσμιο Πόλεμο, τα drones απέκτησαν πιο πρακτικές εφαρμογές, κυρίως για εκπαίδευση αντιαεροπορικών μονάδων, αναγνωριστικές αποστολές και δοκιμές στόχων. Από τη δεκαετία του 1960, η χρήση τους αναπτύχθηκε στον πόλεμο του Βιετνάμ, όπου αξιοποιήθηκαν για παρακολούθηση και συλλογή πληροφοριών. Ένα βασικό χαρακτηριστικό παράδειγμα αυτής της περιόδου, είναι το UAV Ryan Model 147 Lighting Bug [10].

Εικόνα 1.4 UAV Ryan Model 147 [11].

Κατά τις επόμενες δεκαετίες, η πρόοδος στην ηλεκτρονική, τις επικοινωνίες και την πληροφορική οδήγησε στην ανάπτυξη μικρότερων, ελαφρύτερων και πιο αξιόπιστων UAVs, με αναβαθμισμένα συστήματα πλοήγησης και αυτοματισμού [4].

Μετά το 2000, η τεχνολογία των UAVs εισήλθε δυναμικά στους πολιτικούς και εμπορικούς τομείς, καθιστώντας τα drones κεντρικό στοιχείο της βιομηχανίας, της έρευνας και της καθημερινής ζωής [1].

1.3 Σύγχρονη τεχνολογική πραγματικότητα

Τα σύγχρονα μη επανδρωμένα αεροσκάφη έχουν εξελιχθεί σε πλατφόρμες υψηλής αυτονομίας που συνδυάζουν αισθητήρες, επεξεργασία δεδομένων, λογισμικό και λήψη αποφάσεων, επηρεάζοντας πολλούς τομείς.

Από τη μία πλευρά, η θεμελιώδης λειτουργία πτήσης και ελέγχου παραμένει (κινητήρες, έλικες, σταθεροποίηση), ωστόσο η ουσιαστική μεταβολή έγκειται στην ικανότητα αντίληψης του περιβάλλοντος. Τα σύγχρονα drones διαθέτουν ποικιλία αισθητήρων όπως IMU, GPS/GNSS, κάμερες χρώματος, IR, LiDAR, υπερηχητικούς αισθητήρες και αισθητήρες οπτικής ροής, οι οποίοι συνεργάζονται για την εξασφάλιση σταθερής πτήσης, αποφυγής εμποδίων και χαρτογράφησης του περιβάλλοντος [12].

Εικόνα 1.5 Drone με lidar yellow scan [13].

Για παράδειγμα, οι IMU μετράνε επιταχύνσεις και γωνιακές ταχύτητες ώστε να υπολογίζεται η κίνηση του drone [14]. Οι κάμερες και οι αισθητήρες όρασης υποστηρίζουν πτήσεις σε περιβάλλοντα όπου το GPS είναι μη διαθέσιμο ή αναξιόπιστο, καθώς η τεχνολογία

«οπτικής ροής» μπορεί να εκτιμήσει τη σχετική κίνηση του drone σε σχέση με το έδαφος [12]. Ταυτόχρονα, οι αισθητήρες lidar και οι τρισδιάστατες χαρτογραφήσεις επιτρέπουν στο drone να αντιλαμβάνεται με ακρίβεια το περιβάλλον του και να λαμβάνει αποφάσεις πτήσης σε πραγματικό χρόνο [15].

Η ενσωμάτωση όλων αυτών των αισθητήρων είναι ανεπαρκής από μόνη της, απαιτείται επίσης ισχυρή επεξεργασία των δεδομένων μέσω αλγορίθμων τεχνητής νοημοσύνης και μηχανικής μάθησης. Οι αλγόριθμοι αυτοί βοηθούν στην ανάλυση εικόνας, αναγνώριση αντικειμένων, εκτίμηση τροχιάς και συνδυασμό δεδομένων, επιτρέποντας στο drone να λειτουργεί αυτόνομα.

Η ενισχυμένη υπολογιστική ικανότητα, η ελάττωση του βάρους και του κόστους των αισθητήρων, καθώς και η πρόοδος στις μπαταρίες, έχουν συνδυαστεί ώστε τα drones να θεωρούνται πιο έξυπνα. Σήμερα, είναι ικανά όχι μόνο να πετούν, αλλά και να σκέφτονται σε κάποιο βαθμό, να αναγνωρίζουν περιβάλλοντα, να αποφασίζουν την πορεία που θα ακολουθήσουν, να αποφεύγουν εμπόδια και να εκτελούν αποστολές με περιορισμένη ή χωρίς ανθρώπινη παρέμβαση.

Ένα παράδειγμα είναι η χρήση drones σε ανθρωπιστικές ή επιχειρησιακές αποστολές, όπου η ανάγκη για ταχεία αντίδραση είναι κρίσιμη, όπως σε περιοχές καταστροφών ή σε βιομηχανικές επιθεωρήσεις. Η ικανότητα αυτόνομης πτήσης, σε συνδυασμό με οπτική ή θερμική απεικόνιση, διευκολύνει την ταχεία συλλογή δεδομένων και την άμεση λήψη αποφάσεων.

Συνοπτικά, η σύγχρονη τεχνολογική κατάσταση των drones προσδιορίζεται από:

- Πλούσια ποικιλία αισθητήρων,
- Δυναμική ανάλυση δεδομένων και χρήση τεχνητής νοημοσύνης,
- Αυτόνομη πτήση με σταθερότητα, αξιοπιστία και περιορισμένη ανάγκη ανθρώπινης παρέμβασης,
- Επέκταση της χρήσεως του πέρα από στρατιωτικές εφαρμογές, σε βιομηχανία, επιστήμη, έρευνα και καθημερινές χρήσεις.

Η παρούσα εργασία επιδιώκει να οικοδομηθεί πάνω στην τεχνολογική ωρίμανση, συνδυάζοντας την ικανότητα αντίληψης ενός drone μέσω αισθητήρων την χρήση μοντέλων vision language action η αλλιώς VLA και την δράση δηλαδή τον έλεγχο πτήσης.

1.4 Αντικείμενο και στόχος της εργασίας

Η παρούσα εργασία αποσκοπεί στην ανάπτυξη, υλοποίηση και αξιολόγηση ενός συστήματος αυτόνομης πτήσης drone , το οποίο συνδυάζει δεδομένα μέσω κάμερας, τη επεξεργασία οπτικών δεδομένων μέσω ενός σέρβερ όπου τρέχει το μοντέλο vision language action και τον έλεγχο του drone μέσω μίας γέφυρας όπου επιτρέπει την συνεργασία του PX4 και του ROS.

Ο στόχος του συστήματος είναι η λήψη οπτικών δεδομένων σε πραγματικό χρόνο, η μετατροπή τους σε πληροφορία μέσω του μοντέλου VLA και η απόδοση κατάλληλων εντολών ελέγχου πτήσης, οι οποίες εκτελούνται αυτόνομα μέσω MAVROS/PX4.

Ένας επιπλέον στόχος της εργασίας είναι η ευκολία αναπαραγωγής της υλοποίησης. Το σύστημα έχει σχεδιαστεί ώστε οποιοσδήποτε χρήστης, ερευνητής ή μηχανικός, να μπορεί να εγκαταστήσει, να ρυθμίσει και να επαναλάβει το πείραμα, ακολουθώντας τα βήματα που παρατίθενται στα επόμενα κεφάλαια.

Η παρούσα προσέγγιση επικεντρώνεται στη σύνθεση των τριών θεμελιωδών επιπέδων ενός σύγχρονου αυτόνομου συστήματος:

- Αντίληψη,
- Επεξεργασία και κατανόηση εντολών μέσω VLA,
- Ενέργεια / έλεγχος.

Ο στόχος είναι να αποδειχθεί ότι ακόμη και πειραματικές αρχιτεκτονικές VLA μπορούν να χρησιμοποιηθούν για πραγματικούς ελέγχους drones εντός του οικοσυστήματος ROS–PX4, παρέχοντας μια βάση για μελλοντική έρευνα σε γνωστικά αυτόνομα UAVs.

ΚΕΦΑΛΑΙΟ 2

Βασικά εργαλεία ανάπτυξης και ελέγχου drone

2.1 Εισαγωγή

Το δεύτερο κεφάλαιο της εργασίας εστιάζει στα βασικά εργαλεία που υποστηρίζουν τη σχεδίαση, την υλοποίηση και τη δοκιμή του αυτόνομου συστήματος πτήσης drone. Θα παρουσιαστούν και θα αναλυθούν τρία κύρια υποσυστήματα: το λογισμικό πλαίσιο ROS, ο εξομοιωτής gazebo και ο αυτόματος πιλότος PX4. Κάθε εργαλείο παίζει καθοριστικό ρόλο στην ανάπτυξη του συστήματος για παράδειγμα στην πτήση αλλά και στη σύνδεση των επιμέρους λειτουργικών μονάδων που απαιτούνται για τον έλεγχο, την προσομοίωση και τον πραγματικό χειρισμό του drone.

Εικόνα 2.1 Ros logo [16].

Στην ενότητα 2.2 θα πραγματοποιηθεί λεπτομερής παρουσίαση του ROS της αρχιτεκτονικής του, των θεμελιωδών εννοιών όπως κόμβοι, topics, υπηρεσίες, δράσεις και της ροής δεδομένων μεταξύ των υποσυστημάτων. Το ROS αποτελεί τη θεμελιώδη βάση της λογικής ελέγχου του συστήματος, διευκολύνοντας την αδιάλειπτη επικοινωνία μεταξύ αισθητήρων, μονάδων επεξεργασίας και μηχανισμών δράσης [17].

Η ενότητα 2.3 επικεντρώνεται στον εξομοιωτή Gazebo, ο οποίος επιτρέπει τη δημιουργία ενός ρεαλιστικού εικονικού περιβάλλοντος πτήσης, περιλαμβάνοντας τη μοντελοποίηση του drone και των δυναμικών του χαρακτηριστικών, καθώς και την αλληλεπίδραση με αισθητήρες όπως κάμερες, lidar ή imu. Ο εξομοιωτής διευκολύνει τις δοκιμές σε εικονικό περιβάλλον πριν από την πραγματική πτήση, μειώνοντας τους κινδύνους και το κόστος [18].

Στην ενότητα 2.4 θα εξεταστεί ο αυτόματος πιλότος PX4 και η σύνδεση του με το ROS μέσω του mavros. Θα αναλυθεί ο τρόπος με τον οποίο το PX4 διαχειρίζεται τον έλεγχο μηχανισμών πτήσης, ταχύτητας και στάσης, καθώς και η επικοινωνία του με το ROS για την αποστολή και εκτέλεση εντολών [19].

Τέλος, η ενότητα 2.5 θα αναδείξει τον τρόπο με τον οποίο όλα τα υποσυστήματα συνδέονται για να λειτουργούν ως ενιαίο σύστημα. Θα αναλυθεί η ροή δεδομένων από τους αισθητήρες όπως η κάμερα προς το ROS, η επεξεργασία και διαχείριση των πληροφοριών, καθώς και η μεταφορά εντολών προς το PX4 για την εκτέλεση των αντίστοιχων κινήσεων. Η ολοκληρωμένη αλυσίδα επικοινωνίας μεταξύ αισθητήρων, λογισμικού και αυτόματου πιλότου θα παρουσιαστεί με τρόπο που αναδεικνύει τη συνολική λογική ελέγχου του συστήματος.

Με αυτόν τον τρόπο, το κεφάλαιο παρέχει το αναγκαίο τεχνολογικό υπόβαθρο για τα επόμενα στάδια της εργασίας, προσφέροντας στον αναγνώστη τη γνώση που απαιτείται για την κατανόηση της υλοποίησης και της λειτουργίας του αυτόνομου drone στα επόμενα κεφάλαια.

2.2 Διάφορα λειτουργικά συστήματα

Είναι ευρέως αναγνωρισμένο ότι οι πόροι υλικού και λογισμικού ενός υπολογιστή ελέγχονται από ένα σύνολο προγραμμάτων που ονομάζεται λειτουργικό σύστημα (OS). Οι θεμελιώδεις λειτουργίες, όπως η διαχείριση και η κατανομή μνήμης, η ιεράρχηση αιτημάτων του συστήματος, η εποπτεία συσκευών εισόδου και εξόδου, η διευκόλυνση της δικτύωσης και η διαχείριση αρχείων, εκτελούνται από το λειτουργικό σύστημα. Οι διάφορες υπηρεσίες που προσφέρει το λειτουργικό σύστημα περιλαμβάνουν: διαχείριση διεργασιών, διαχείριση μνήμης, συντήρηση δίσκων και συστημάτων αρχείων, δυνατότητες δικτύωσης, παροχή ασφάλειας και διεπαφές οδηγών συσκευών, μεταξύ άλλων.

Υπάρχει πλήθος λειτουργικών συστημάτων, όπως Windows, Mac OS X, UNIX, Solaris, BS3000, MS-DOS κ.ά. Εντάσσονται σε εξοπλισμό συστήματος ή αυτοματοποιημένο εξοπλισμό που σχετίζεται με τη διαχείριση και τον συντονισμό των δραστηριοτήτων του συγκεκριμένου συστήματος. Το μηχανογραφημένο σύστημα μπορεί να είναι υπολογιστής, σταθμός εργασίας, διακομιστής, PC, μία μέθοδος, smartphone, συσκευή πλοήγησης ή οποιοδήποτε σύστημα με αυτονομία, όπως ένα ρομπότ. Το λειτουργικό σύστημα λειτουργεί ως πλατφόρμα για όλες τις εφαρμογές που εκτελούνται σε οποιοδήποτε υλικό.

Υπάρχουν ποικίλοι τύποι λειτουργικών συστημάτων:

- Λειτουργικά συστήματα πραγματικού χρόνου OS,
- Πολυχρηστικά και μονοχρηστικά λειτουργικά συστήματα,
- Πολυδιεργασιακά και μονοδιεργασιακά λειτουργικά συστήματα,
- Κατανεμημένα λειτουργικά συστήματα OS,
- Ενσωματωμένα λειτουργικά συστήματα OS,

• Εξειδικευμένα λειτουργικά συστήματα, όπως το «Robot OS» [20].

2.2.1 Robot operating system

Το ROS αποτελεί ένα σύνολο λογισμικών δομών για την ανάπτυξη ρομποτικού λογισμικού, παρέχοντας λειτουργίες παρόμοιες με αυτές ενός λειτουργικού συστήματος σε διάφορες κατηγορίες υπολογιστικών συστημάτων. Προσφέρει τυπικές υπηρεσίες λειτουργικού συστήματος, όπως αφαίρεση υλικού, έλεγχο συσκευών χαμηλού επιπέδου, υλοποίηση κοινών λειτουργιών, μεταβίβαση μηνυμάτων μεταξύ διεργασιών και διαχείριση πακέτων.

Οι διαδικασίες που στηρίζονται στο ROS αναπαρίστανται μέσω μιας αρχιτεκτονικής γράφου, όπου η επεξεργασία εκτελείται σε κόμβους που είναι ικανοί να λαμβάνουν, να δημοσιεύουν και να πολυπλέκουν μηνύματα όπως αυτά των αισθητήρων, του ελέγχου, της κατάστασης, του σχεδιασμού και των ενεργοποιητών. Αν και η χαμηλή καθυστέρηση είναι κρίσιμη για τον έλεγχο ρομπότ, το ROS δεν αποτελεί λειτουργικό σύστημα πραγματικού χρόνου. Ωστόσο, δύναται να συνδυαστεί με κώδικα σε πραγματικό χρόνο. Το κύριο λειτουργικό σύστημα σε αυτό το έργο είναι το Debian στο BBB.

Το λογισμικό στο οικοσύστημα ROS διακρίνεται σε τρεις κατηγορίες:

- Εργαλεία ανεξάρτητα από γλώσσα και πλατφόρμα για την ανάπτυξη και διανομή λογισμικού που βασίζεται στο ROS,
- Υλοποιήσεις βιβλιοθηκών πελάτη όπως roscpp, rospy και roslisp,
- Πακέτα με κώδικα εφαρμογών που αξιοποιούν μία ή περισσότερες βιβλιοθήκες πελάτη ROS [20].

2.2.2 Εφαρμογές του ROS

Το ROS είναι εφαρμόσιμο σε ποικίλους τομείς. Ο όρος ROS αναφέρεται στο σύνολο των ετερογενών λογισμικών προγραμμάτων του οικοσυστήματός του, ενώ τα πακέτα ROS περιλαμβάνουν μια συγκεκριμένη ομάδα κώδικα που αξιοποιεί τις βιβλιοθήκες πελάτη.

Βασικές λειτουργίες περιλαμβάνουν:

- Κύριο κόμβο συντονισμού,
- Δημοσίευση ή εγγραφή σε ροές δεδομένων, εικόνες, στερεοσκοπία, λέιζερ, έλεγχο,
- Multiplexing πληροφορίας,
- Δημιουργία και καταστροφή κόμβων,
- Κατανεμημένη λειτουργία σε πολυπύρηνα συστήματα, GPUs και συστοιχίες,

- Καταγραφή,
- Συστήματα δοκιμών.

Τα πακέτα ROS μπορούν να εφαρμοστούν αποτελεσματικά στους εξής τομείς: Μία ομάδα λογισμικού μπορεί να χρησιμοποιηθεί, για παράδειγμα, για αντίληψη και κίνηση, ενώ μια άλλη ομάδα λογισμικών μπορεί να απαιτείται για αναγνώριση αντικειμένων και στερεοσκοπική όραση, και αντίστοιχα μια τρίτη ομάδα.

Εφαρμογές πακέτων ROS περιλαμβάνουν:

- Αντίληψη,
- Αναγνώριση αντικειμένων,
- Τμηματοποίηση και αναγνώριση,
- Αναγνώριση προσώπου,
- Αναγνώριση χειρονομιών,
- Παρακολούθηση κίνησης,
- Κατανόηση κίνησης,
- Δομή από κίνηση,
- Στερεοσκοπική όραση,
- Κινητική ρομποτική,
- Έλεγχο,
- Σχεδιασμό,
- Σύλληψη.

Τόσο τα γλωσσικά ανεξάρτητα εργαλεία όσο και οι κύριες βιβλιοθήκες C++, Python είναι χρήσιμα για την ανάπτυξη πακέτων που σχετίζονται με οδηγούς υλικού, μοντέλα ρομπότ, τύπους δεδομένων, σχεδίαση, αντίληψη, SLAM, εργαλεία και άλλους αλγόριθμους [20].

2.2.3 Πλεονεκτήματα από τη χρήση του ROS

Το ROS είναι ένα μετα-λειτουργικό πλαίσιο, ενδιάμεσο μεταξύ λειτουργικού συστήματος και middleware για ρομποτικές υπηρεσίες. Προσφέρει τόσο θεμελιώδεις λειτουργικές υπηρεσίες όσο και ανώτερες δυνατότητες, όπως ασύγχρονες και σύγχρονες κλήσεις, κεντρική βάση δεδομένων και σύστημα παραμετροποίησης ρομπότ.

Προτού την εμφάνιση του ROS, οι ερευνητές υποχρεώνονταν να σχεδιάζουν εκ του μηδενός το ενσωματωμένο λογισμικό και το υλικό ενός ρομπότ, απαιτώντας γνώσεις μηχανολογίας και ενσωματωμένων συστημάτων. Τα προγράμματα ήταν στενά συνδεδεμένα

με το υλικό, περιορίζοντας την επαναχρησιμοποίηση τους. Ένα επιπλέον πλεονέκτημα είναι η αξιοποίηση γνώσεων από ποικίλους τομείς:

1. Υλικοτεχνική διαχείριση μέσω οδηγών,
2. Διαχείριση μνήμης και διεργασιών,
3. Διαχείριση ταυτόχρονης εκτέλεσης και συγχώνευσης δεδομένων,
4. Χρήση τεχνητής νοημοσύνη [20].

2.2.4 Τεχνολογία ROS

Βασικές έννοιες του ROS περιλαμβάνουν:

• Κόμβους,
• Κύριο κόμβο (Master),
• Topics,
• Υπηρεσίες,
• Bags (αρχεία καταγραφής),

Το ROS αντιπροσωπεύει το μέλλον του λογισμικού για αυτόνομες συσκευές. Λειτουργεί ως ένα ανοιχτό σύστημα εργαλείων για την ανάπτυξη λογισμικού ρομπότ, παρέχοντας υπηρεσίες όπως αφαίρεση υλικού, έλεγχο συσκευών, υλοποίηση κοινών λειτουργιών και διαχείριση πακέτων δεδομένων. Η ανάπτυξη του drone στην παρούσα μελέτη βασίζεται στο ROS.

2.3 Gazebo

Το gazebo είναι ο πιο γνωστός τρισδιάστατος προσομοιωτής που χρησιμοποιείται με το ROS. Αυτό αποτελεί ένα ισχυρό προγραμματιστικό εργαλείο με ενσωματωμένη μηχανή απεικόνισης, γραφικά ισχυρής ανάλυσης και προγραμματιστικά φιλικό γραφικό περιβάλλον διεπαφής, το οποίο, όπως και το ROS, διατίθεται δωρεάν. Το Gazebo παρέχει τη δυνατότητα σχεδίασης ενός ή περισσοτέρων ρομποτικών συστημάτων σε ποικίλα λειτουργικά σενάρια, με ακριβή και αποδοτική προσομοίωση αλγορίθμων. Ο χρήστης έχει τη δυνατότητα να δημιουργήσει εσωτερικούς ή εξωτερικούς χώρους με ποικιλία 30 διαφορετικών μοντέλων, διασφαλίζοντας εξαιρετική ακρίβεια και συνοχή με τις πραγματικές συνθήκες και τα δυναμικά φαινόμενα που συμβαίνουν.

Επιπλέον, παρέχει ποικιλία επιλογών για τις μηχανές φυσικής, με προεπιλεγμένη την open dynamics engine. Το Gazebo συνίσταται από δύο τμήματα: έναν διακομιστή (ο οποίος λειτουργεί ως διεργασία gzserver) και έναν πελάτη (ο οποίος λειτουργεί ως διεργασία gzclient). Μία ακόμη εντυπωσιακή και πολύτιμη ικανότητα του Gazebo είναι η δυνατότητά

του να προσομοιώνει επίσης αισθητήρες, συμπεριλαμβανομένων αισθητήρων μέτρησης δύναμης, επιταχυνσιόμετρων και καμερών.

Εικόνα 2.2 Gazebo logo [21] .

Οι δυνατότητες προσομοίωσης και απεικόνισης που προσφέρει το ROS μέσω του Gazebo αποτελούν πολύτιμα χαρακτηριστικά του. Όταν η ανάπτυξη λογισμικού περιορίζεται σε συγκεκριμένο υλικό, γεγονός που σχετίζεται με τον τομέα της ρομποτικής, η παραγωγικότητα υποβαθμίζεται. Ένα φυσικό ρομπότ συνήθως αποτελεί εμπόδιο στην ανάπτυξη, καθώς είναι ένας περιορισμένος πόρος που πιθανόν να μοιράζεται από μια ομάδα. Επιπλέον, ορισμένα ρομπότ, περιλαμβανομένων οχημάτων και βραχιόνων, απαιτούν από ομάδες την εκτέλεση πειραμάτων. Συχνά, το ρομπότ και η ομάδα παραμένουν σε αδράνεια ενώ ένας προγραμματιστής αναζητά σφάλματα στον κώδικα υπό δοκιμή. Με τη χρήση ενός κατάλληλου προσομοιωτή ρομπότ, ο κώδικας μπορεί να αναπτυχθεί σε περιβάλλον προσομοίωσης, επιταχύνοντας τη διαδικασία και διασφαλίζοντας τη μέγιστη δυνατή ασφάλεια τόσο του προσωπικού όσο και της ρομποτικής διάταξης. Εάν ο προσομοιωτής διαθέτει υψηλή πιστότητα και προσφέρει ακριβή αναπαράσταση της πραγματικής συμπεριφοράς του ρομπότ στον φυσικό κόσμο, πιθανόν να απαιτούνται ελάχιστες ή καθόλου τροποποιήσεις για την εκτέλεση του ίδιου κώδικα στο φυσικό σύστημα. Συνήθως, το λογισμικό που έχει αναπτυχθεί για προσομοίωση στο ROS δεν απαιτεί τροποποιήσεις για να λειτουργήσει στο φυσικό του αντίστοιχο. Με τη χρήση του συστήματος επικοινωνίας του ROS, οι κόμβοι δεν ενδιαφέρονται για την τοποθεσία του άλλου άκρου της επικοινωνίας, αρκεί οι τύποι μηνυμάτων και τα

ονόματα των θεμάτων να είναι ορθά. Η επιτυχία και η χρησιμότητα της ανάπτυξης λογισμικού στην προσομοίωση εξαρτώνται άμεσα από την ποιότητα του προσομοιωτή [22] .

2.3.1 Προσομοίωση

Για να διεξαχθεί μια δυναμική προσομοίωση με ρεαλιστικό και αξιόπιστο τρόπο, τα μοντέλα των αντικειμένων και των κατασκευών πρέπει να πληρούν συγκεκριμένες προϋποθέσεις και να περιγράφονται με επαρκή λεπτομέρεια. Οι απαιτήσεις αυτές διακρίνονται σε τρεις κύριες κατηγορίες: το οπτικό μοντέλο, το δυναμικό μοντέλο και το μοντέλο συγκρούσεων.

Το δυναμικό μοντέλο πρέπει πάντα να περιλαμβάνει τη μάζα και τις αδρανειακές ιδιότητες κάθε αντικειμένου, καθώς αποτελούν τα αναγκαία στοιχεία μιας αφηρημένης φυσικής περιγραφής. Μόνο με αυτά τα χαρακτηριστικά μπορεί να επιτευχθεί μια λειτουργική προσομοίωση. Για παράδειγμα, είναι επαρκή για τον προσδιορισμό της κίνησης ενός δορυφόρου υπό την επίδραση της βαρύτητας.

Όσον αφορά το μοντέλο συγκρούσεων, η ανάλυση της αλληλεπίδρασης δύο σωμάτων, είτε πρόκειται για απλές συγκρούσεις είτε για ρομποτικούς χειρισμούς, όπως η σύλληψη ή η ώθηση αντικειμένων, απαιτεί τον υπολογισμό των δυνάμεων και των ροπών που αναπτύσσονται στα σημεία επαφής. Η προσομοίωση αυτών των φαινομένων είναι απαιτητική και μπορεί να πραγματοποιηθεί με ποικίλες προσεγγίσεις. Η ODE στηρίζεται στην ενέργεια και στη διατήρηση της ορμής για τον υπολογισμό των συνεπειών μιας σύγκρουσης, χωρίς να εξετάζει αναλυτικά την άμεση εξέλιξη των δυνάμεων κατά τη διάρκεια της πρόσκρουσης. Αυτή η μέθοδος αποδίδει εξαιρετικά σε σύντομες κρούσεις, αλλά δεν είναι εξίσου αποτελεσματική σε φαινόμενα συνεχούς επαφής, όπως η κύλιση τροχών στο έδαφος ή η επαφή δακτύλων ρομπότ με αντικείμενα.

Επιπλέον, προκειμένου να αξιολογηθούν τα αποτελέσματα μιας σύγκρουσης, ο προσομοιωτής οφείλει να αναγνωρίζει τον χρόνο και την τοποθεσία εμφάνισης κάθε επαφής. Επομένως, απαιτείται μια κατάλληλη γεωμετρική αναπαράσταση του αντικειμένου. Η εν λόγω αναπαράσταση συνιστά το μοντέλο συγκρούσεων και μπορεί να κυμαίνεται από απλά γεωμετρικά σχήματα, όπως κύβοι και κύλινδροι, έως μοντέλα υψηλής λεπτομέρειας προερχόμενα από αρχεία CAD. Οι επιφάνειες συνήθως αναπαρίστανται με μοντέλα STL που αποτελούνται από τριγωνικές όψεις, μια επιλογή που διευκολύνει τους ταχείς υπολογισμούς τομών μεταξύ των επιφανειών.

Η τρίτη κατηγορία αφορά το οπτικό μοντέλο, το οποίο χρησιμοποιείται αποκλειστικά για γραφική απεικόνιση. Η αναπαράσταση της προσομοιωμένης δυναμικής σε γραφική μορφή διευκολύνει σημαντικά την κατανόηση των πειραματικών αποτελεσμάτων και τον εντοπισμό σφαλμάτων στο λογισμικό. Συχνά, το οπτικό μοντέλο είναι παρόμοιο με το μοντέλο συγκρούσεων, καθώς και τα δύο βασίζονται στη γεωμετρία του αντικειμένου. Ωστόσο, το οπτικό μοντέλο συνήθως διαθέτει υψηλότερη λεπτομέρεια, περιλαμβάνοντας περισσότερες τριγωνικές όψεις. Αυτό οφείλεται στο γεγονός ότι η γραφική απεικόνιση απαιτεί πολύ λιγότερη υπολογιστική ισχύ σε σύγκριση με τον εντοπισμό συγκρούσεων σε σύνθετα μοντέλα. Επιπλέον, το οπτικό και το μοντέλο συγκρούσεων μπορεί να έχουν διαφορετικές επιπρόσθετες παραμέτρους, όπως συντελεστές τριβής ή ελαστικότητας (που δεν σχετίζονται με την εμφάνιση), ή χρωματικές ιδιότητες και χαρακτηριστικά ανακλαστικότητας (που δεν επηρεάζουν τις συγκρούσεις).

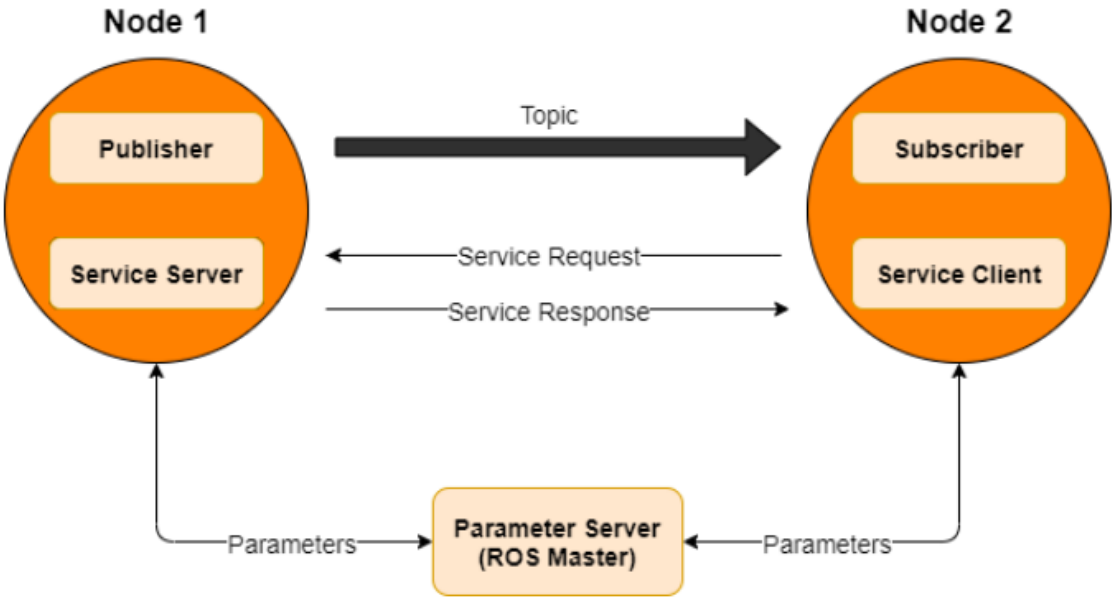
Ένα ρομποτικό μοντέλο που περιλαμβάνει ορθά και τις τρεις κατηγορίες μπορεί να αποτυπωθεί σε ένα αρχείο simulation description format (SDF). Το SDF υποστηρίζεται από το gazebo και είναι θεμελιωμένο στη γλώσσα XML. Ωστόσο, το ROS δεν υποστηρίζει το SDF, αλλά χρησιμοποιεί το unified robot description format, το οποίο είναι παρόμοιο, αν και παλαιότερο και με περιορισμένες δυνατότητες. Το URDF διαρθρώνει τη ρομποτική κατασκευή ως μια ανοιχτή κινηματική αλυσίδα, δηλαδή μια δενδρική δομή όπου κάθε σύνδεσμος έχει έναν μοναδικό γονέα, αλλά μπορεί να συνδέεται με πολλαπλά παιδιά. Αυτή η δομή περιγράφει τους συνδέσμους και τις αρθρώσεις, καθώς και τα δυναμικά, οπτικά και γεωμετρικά χαρακτηριστικά τους. Οι αρθρώσεις περιλαμβάνουν πρόσθετες ιδιότητες, όπως τον τύπο τους, τους άξονες κίνησης και τα όρια λειτουργίας.

Τέλος, είναι σημαντικό να επισημανθεί ότι το rviz και το gazebo εξυπηρετούν διαφορετικούς σκοπούς και δεν πρέπει να συγχέονται. Το rviz απεικονίζει αποκλειστικά την παρούσα κατάσταση της ρομποτικής διάταξης, παραβλέποντας φυσικά φαινόμενα όπως η βαρύτητα ή οι τριβές [22].

2.4 Rviz

Η οπτικοποίηση ROS (Rviz) αποτελεί ένα ισχυρό εργαλείο τρισδιάστατης απεικόνισης ρομπότ για εφαρμογές ROS. Δημιουργεί γραφικό περιβάλλον για την απεικόνιση των καταγεγραμμένων δεδομένων αισθητήρων, του ρομποτικού μοντέλου και των περιβαλλοντικών χαρτών, με στόχο την ανάπτυξη και το debugging των ελεγκτών του ρομπότ. Υποστηρίζει πολυγωνικά σχήματα ή δείκτες που καθορίζονται από τον χρήστη και επιτρέπει

την εκτέλεση διαδραστικών κινήσεων με εντολές και δεδομένα που προέρχονται από τον κόμβο του χρήστη. Στην **Εικόνα 2.3** απεικονίζεται ένα ενδεχόμενο παράδειγμα του μοντέλου του ρομπότ και των εμποδίων που σχηματίζεται στο περιβάλλον γύρω του [23].



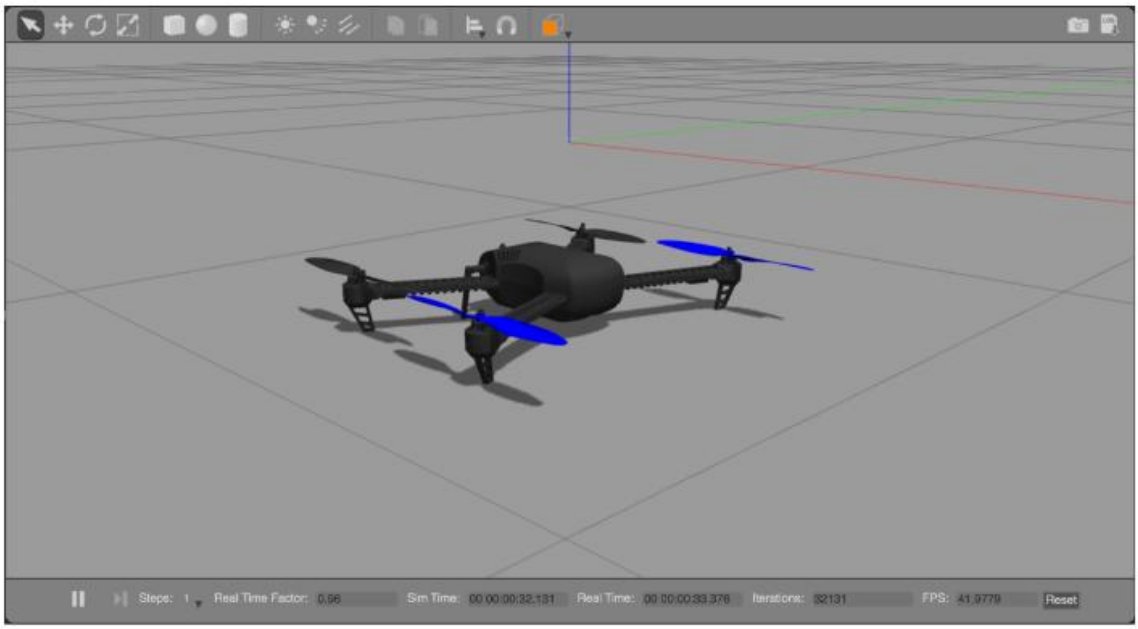
Εικόνα 2.3 Επικοινωνία μεταξύ κόμβων [23].

ΚΕΦΑΛΑΙΟ 3

PX4 structure

3.1 Εισαγωγή στο PX4

Το PX4 αποτελεί ένα λογισμικό ανοιχτού κώδικα για τη διαχείριση πτήσεων drones και άλλων μη επανδρωμένων οχημάτων. Το πρόγραμμα προσφέρει ένα ευέλικτο σύνολο εργαλείων για προγραμματιστές drone, προκειμένου να ανταλλάσσουν τεχνολογίες και να αναπτύσσουν προσαρμοσμένες λύσεις για εφαρμογές drone [24].



Εικόνα 3.1 PX4 quadcopter σε περιβάλλον εξομοίωσης gazebo.

Τα κύρια χαρακτηριστικά του είναι:

• Αρθρωτή αρχιτεκτονική: Το PX4 διαθέτει έντονη modular αρχιτεκτονική και επεκτασιμότητα, τόσο σε επίπεδο υλικού, επιτρέποντας την ενσωμάτωση ποικίλων αισθητήρων, όσο και σε επίπεδο λογισμικού, διευκολύνοντας την άμεση αλληλεπίδραση με το firmware για την υλοποίηση οποιασδήποτε επιθυμητής λειτουργίας.

• Ανοιχτός κώδικας: Η εξέλιξη του PX4 διεξάγεται παγκοσμίως από μια εξαιρετικά δραστήρια κοινότητα. Το flight stack δεν αποσκοπεί στην κάλυψη των αναγκών συγκεκριμένων εργαστηρίων ή εταιρειών, αλλά έχει σχεδιαστεί ως ένα γενικό εργαλείο που μπορεί να χρησιμοποιηθεί τόσο στη βιομηχανία όσο και στα πανεπιστήμια.

• Παραμετροποίηση: Το PX4 παρέχει βελτιστοποιημένα APIs και SDKs για προγραμματιστές που ασχολούνται με την προσθήκη νέων λειτουργιών και την προσαρμογή του συστήματος.

• Autonomy stack: Το PX4 έχει σχεδιαστεί για να συνεργάζεται στενά με ενσωματωμένους υπολογιστές, ιδιαιτέρως σε σχέση με την εφαρμογή όρασης για αυτόνομες λειτουργίες [24].

3.2 Αρχιτεκτονική PX4

Το PX4 αποτελεί από δύο κύρια στρώματα: το flight stack, το οποίο είναι το στρώμα εκτίμησης και ελέγχου πτήσης, και το middleware, το οποίο παρέχει ένα γενικό ρομποτικό στρώμα για την ενσωμάτωση υλικού και την εσωτερική/εξωτερική επικοινωνία. Το κύριο χαρακτηριστικό του PX4 είναι ότι όλα τα αεροσκάφη μοιράζονται την ίδια βάση κώδικα, συμπεριλαμβανομένων και άλλων ρομποτικών συστημάτων όπως οχήματα εδάφους, υποβρύχια και σκάφη.

3.2.1 Flight stack

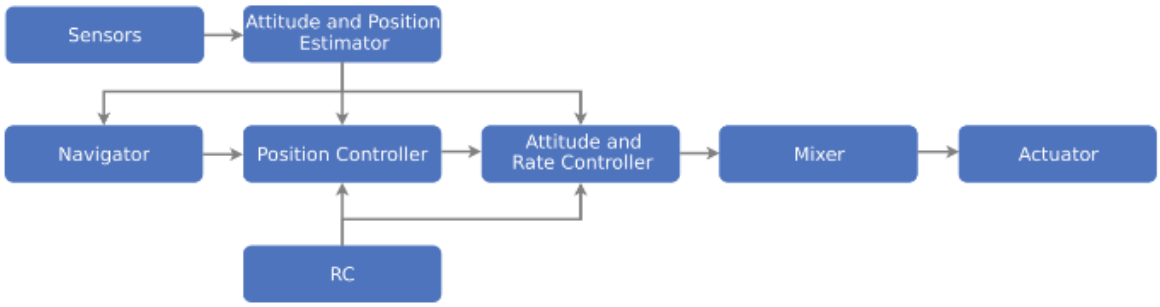
Σύμφωνα με τον οδηγό ανάπτυξης, «το flight stack αποτελεί ένα σύνολο αλγορίθμων καθοδήγησης, πλοήγησης και ελέγχου για αυτόνομα drones. Εμπεριέχει ελεγκτές για αεροσκάφη σταθερής πτέρυγας, πολυκόπτερα και VTOL, καθώς και εκτιμητές για στάση και θέση [25].

Η **Εικόνα 3.2** παρέχει μια συνοπτική επισκόπηση των δομικών στοιχείων του flight stack. Αναφέρεται σε ολόκληρη την αλυσίδα, συμπεριλαμβανομένων των αισθητήρων, της εισόδου του τηλεχειριστηρίου και του αυτόνομου ελέγχου πτήσης ,καθώς και του ελέγχου των κινητήρων ή των σερβομηχανισμών [25].

Ο εκτιμητής αποκτά μία ή περισσότερες εισόδους από αισθητήρες, τις συνδυάζει βάσει καθορισμένων αλγορίθμων και υπολογίζει την κατάσταση του οχήματος (π.χ., τη στάση σύμφωνα με τα δεδομένα του IMU).

Ο ελεγκτής αποτελεί ένα στοιχείο που δέχεται ως είσοδο ένα setpoint (δηλαδή τον στόχο) και μία μέτρηση (εφόσον είναι διαθέσιμη) ή μια εκτιμώμενη κατάσταση (process variable). Ο σκοπός του ελεγκτή είναι να διορθώνει τη συμπεριφορά του συστήματος ώστε να πετυχαίνει τον επιθυμητό στόχο. Η έξοδός του συνιστά τη διόρθωση που απαιτείται για την επίτευξη του setpoint. Ο ελεγκτής θέσης λαμβάνει ως είσοδο τα setpoints θέσης, ως μεταβλητή διαδικασίας την τρέχουσα εκτιμώμενη θέση, και παρέχει ως έξοδο έναν συνδυασμό setpoints στάσης και ώσης που κατευθύνει το όχημα προς την επιθυμητή θέση.

«Ο mixer δέχεται αριθμητικές εντολές (π.χ. στρίψε δεξιά) και τις μετατρέπει σε επιμέρους εντολές προς τους κινητήρες, διασφαλίζοντας ταυτόχρονα ότι δεν παραβιάζονται συγκεκριμένα όρια.» Αυτή η μετάφραση είναι ειδική για κάθε κατηγορία οχήματος και εξαρτάται από ποικίλους παράγοντες, όπως η διάταξη των κινητήρων σε σχέση με το κέντρο βάρους ή η ροπή αδράνειας του οχήματος [25].



Εικόνα 3.2 Διάγραμμα του flight stack [25].

Ελεγκτής θέσης multicopter

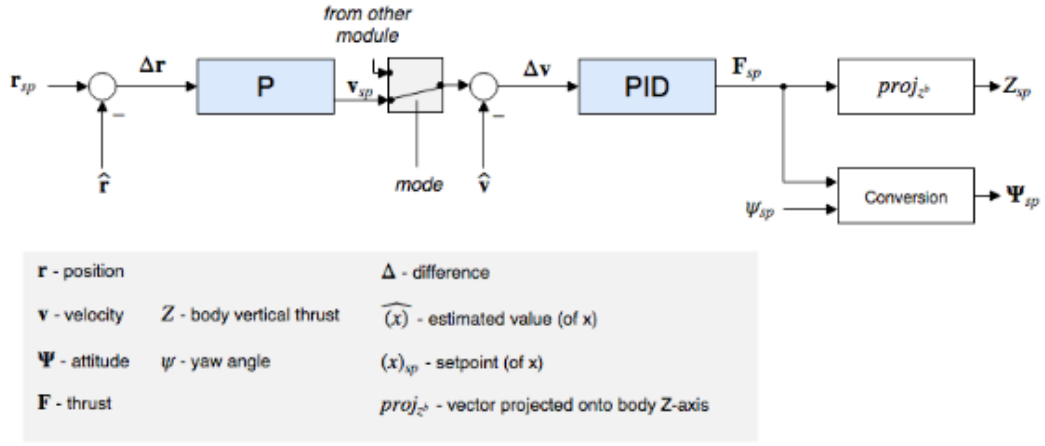
Το διάγραμμα του ελεγκτή θέσης του multicopter απεικονίζεται κατωτέρω. Ο ελεγκτής περιλαμβάνει δύο βρόχους: έναν βρόχο P για το σφάλμα θέσης και έναν βρόχο PID για το σφάλμα ταχύτητας. Η έξοδος του ελεγκτή αποτελεί ένα διάνυσμα ώσης, το οποίο διαχωρίζεται σε δύο συνιστώσες: τη διεύθυνση της ώθησης (δηλαδή τον πίνακα περιστροφής για τον προσανατολισμό του multicopter) και το μέγεθος της ώθησης (δηλαδή την ίδια την ώθηση του multicopter). Ο ελεγκτής δεν απαιτεί γωνίες Euler για τη λειτουργία του, αυτές παράγονται αποκλειστικά για σκοπούς ελέγχου και για την κατανόηση των προγραμματιστών (μέσω καταγραφής).

Ορισμένες παρατηρήσεις σχετικά με αυτό το διάγραμμα ελέγχου, σύμφωνα με τον οδηγό προγραμματιστή:

• Οι εκτιμώμενες καταστάσεις και μεταβλητές προκύπτουν από το EKF2.

• Το διάγραμμα ελέγχου απεικονίζει μια τυπική διαδοχική διάταξη βρόχων θέσης-ταχύτητας (cascaded position velocity loop).

• Ο βρόγχος θέσης ενεργοποιείται αποκλειστικά υπό συγκεκριμένες συνθήκες: όταν το multicopter διατηρεί θέση ή όταν η επιθυμητή ταχύτητα κατά μήκος ενός άξονα είναι μηδενική. Για τον λόγο αυτό, ο εξωτερικός βρόχος μπορεί να παρακαμφθεί ανάλογα με τη χρησιμοποιούμενη λειτουργία (όταν δεν απαιτείται διατήρηση θέσης) [26].



Εικόνα 3.3 Ελεγκτής θέσης [26] .

Ο ολοκληρωτής εντός του εσωτερικού βρόχου του ελεγκτή ταχύτητας περιλαμβάνει μηχανισμό anti-reset windup, εφαρμόζοντας τη μέθοδο του περιορισμού (clamping).

Ελεγκτής στάσης multicopter

Το διάγραμμα του ελεγκτή στάσης του multicopter [27] παρουσιάζει σημαντική ομοιότητα με το διάγραμμα του ελεγκτή θέσης, αν και εκτελεί τον ελεγκτή στάσης και ρυθμού του multicopter. Δέχεται ως εισόδους τα setpoints στάσης ή τα setpoints ρυθμού και παράγει ως έξοδο μηνύματα ελέγχου των ενεργοποιητών. Ο ελεγκτής περιλαμβάνει δύο βρόχους: έναν βρόχο P για το γωνιακό σφάλμα και έναν βρόχο PID για το σφάλμα γωνιακού ρυθμού.

3.2.2 Middleware

Το middleware περιλαμβάνει κυρίως οδηγούς συσκευών για ενσωματωμένους αισθητήρες, επικοινωνία με τον εξωτερικό κόσμο (συνοδευτικός υπολογιστής, GCS κ.λπ.) και το message bus. Το λογισμικό είναι φτιαγμένο από modules/προγράμματα που αλληλοεπιδρούν μέσω ενός συστήματος μηνυμάτων publish και subscribe του uORB.» [25].

Ενώ το flight stack είναι κρίσιμο για την αξιόπιστη πτήση του drone, το middleware αποτελεί το τμήμα του λογισμικού που διευκολύνει την προσθήκη λειτουργιών, την ενσωμάτωση αισθητήρων ή την τροποποίηση της συμπεριφοράς των υφιστάμενων στοιχείων.

Η εφαρμογή ενός σχήματος publish και subscribe υποδηλώνει ότι:

• Το σύστημα είναι αντιδραστικό δηλαδή ασύγχρονο. Επίσης ενημερώνεται αμέσως κάθε φορά όταν προκύπτουν νέα δεδομένα.

• Όλες οι διαδικασίες και οι επικοινωνίες είναι πλήρως παραλληλοποιημένες.

• Ένα στοιχείο του συστήματος μπορεί να αξιοποιεί δεδομένα από οποιαδήποτε πηγή με ασφαλή τρόπο αναφορικά με τα thread.

Επιπλέον, το middleware περιλαμβάνει ένα στρώμα προσομοίωσης που επιτρέπει στο PX4 να λειτουργεί σε λειτουργικό σύστημα επιτραπέζιου υπολογιστή και να ελέγχει ένα μοντελοποιημένο όχημα σε προσομοιωμένο περιβάλλον.

3.2.3 Περιβάλλον εκτέλεσης

Το PX4 λειτουργεί σε ποικιλία λειτουργικών συστημάτων που υποστηρίζουν το POSIX-API (NuttX, macOS, Linux). Η επικοινωνία μεταξύ των modules βασίζεται στο uORB και στηρίζεται σε κοινόχρηστη μνήμη. Ένας ενιαίος χώρος διευθύνσεων αξιοποιείται για την εκτέλεση του συνόλου του middleware του PX4, δεδομένου ότι η μνήμη είναι κοινή σε όλα τα modules. Ωστόσο, το σύστημα έχει σχεδιαστεί ώστε, με ελάχιστες τροποποιήσεις, να επιτρέπεται η εκτέλεση κάθε module σε ιδιωτικό χώρο διευθύνσεων [25].

3.3 uORB

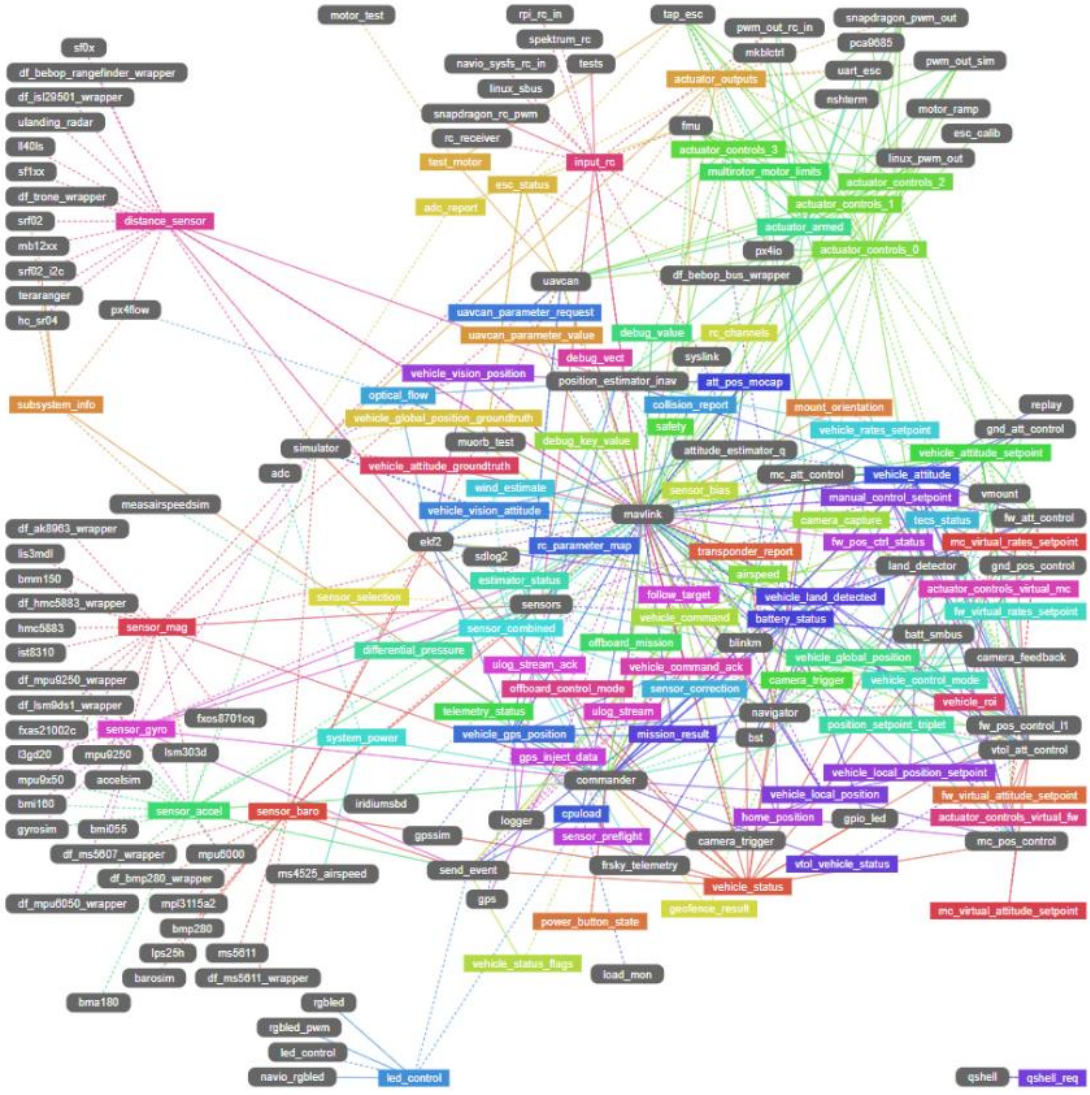
Το uORB αποτελεί τον πυρήνα της επικοινωνίας μεταξύ των εσωτερικών υποσυστημάτων του drone. Αυτό είναι ένα ασύγχρονο messaging API publish() / subscribe() που χρησιμοποιείται για την επικοινωνία μεταξύ threads ή διεργασιών και ενεργοποιείται αυτόματα κατά την εκκίνηση του συστήματος (εντολή uorb start), καθώς πολλές εφαρμογές εξαρτώνται από αυτό [28].

Αν και υπάρχουν πολλά ενσωματωμένα topics, μπορεί να είναι αναγκαία η προσθήκη νέων. Για την προσθήκη ενός νέου topic, απαιτείται η δημιουργία ενός νέου αρχείου .msg στον φάκελο msg/ και η καταχώριση του ονόματος του αρχείου στη λίστα msg/CMakeLists.txt. Στη συνέχεια, ο απαιτούμενος κώδικας C/C++ παράγεται αυτοματοποιημένα κατά τη διαδικασία κατασκευής. Η ανάλυση των υφιστάμενων αρχείων .msg μπορεί να συμβάλει στην επαλήθευση των υποστηριζόμενων τύπων. Ένα μήνυμα μπορεί επίσης να χρησιμοποιηθεί ως ενσωματωμένο μέσα σε άλλα μηνύματα [28].

«Σε κάθε παραγόμενο C/C++ struct προστίθεται ένα πεδίο τύπου uint64_t για την καταγραφή χρονικής σήμανσης.» Αυτό αξιοποιείται από τον publisher και πρέπει να συμπληρώνεται κατά τη δημοσίευση του μηνύματος. [28].

Τα modules απεικονίζονται με γκρι στρογγυλεμένα πλαίσια, ενώ τα topics με χρωματιστά τετράγωνα. Τα modules μπορούν είτε να κάνουν subscribe είτε να κάνουν publish

σε topics: οι διακεκομμένες γραμμές δείχνουν ότι ένα module δημοσιεύει σε ένα topic, ενώ οι συνεχείς γραμμές δείχνουν ότι κάνει subscribe σε αυτό.



Εικόνα 3.4 uORB σχηματικό [28].

Για να γίνει η προσθήκη ενός module με τα άλλα γίνεται:

1. Η αρχική ενέργεια για την εκτέλεση ενός νέου module είναι η κατασκευή του. Στο παρόν παράδειγμα, χρησιμοποιείται το ενσωματωμένο πρότυπο module που βρίσκεται στον φάκελο firmware/src/templates/ module ή στη διαδικτυακή διεύθυνση https://dev.px4.io/master/en/apps/module_template.html. Κάθε module περιλαμβάνει τρία βασικά στοιχεία: το αρχείο .cpp, όπου υλοποιείται ο κώδικας, το αρχείο .h, το οποίο περιέχει όλες τις δηλώσεις κλάσεων και τις

βιβλιοθήκες, και το αρχείο CMakeLists.txt, το οποίο είναι απαραίτητο για την κατασκευή του κώδικα.

2. Η δεύτερη ενέργεια είναι η τοποθέτηση του φακέλου του module, ο οποίος περιλαμβάνει τα τρία προαναφερθέντα αρχεία, στον κατάλληλο κατάλογο. Ο κατάλογος εξαρτάται από την κατηγορία του module που δημιουργήθηκε (driver, module, παράδειγμα εντολές συστήματος κ.λπ.). Ωστόσο, το αρχικό σημείο παραμένει σταθερό: firmware/src. Για παράδειγμα, εάν έχει αναπτυχθεί ένας οδηγός, θα πρέπει να τοποθετηθεί στον φάκελο firmware /src/drivers.
3. Η τρίτη ενέργεια είναι να ενημερωθεί το σύστημα σχετικά με τη διαθεσιμότητα του νέου module. Για να επιτευχθεί αυτό, απαιτείται η μετάβαση στο firmware /board/px4 και η προσθήκη του ονόματος του module στο αρχείο default.cmake της αντίστοιχης πλατφόρμας-στόχου, εντός της σχετικής λίστας (στο παρόν παράδειγμα, στη λίστα των drivers).
4. Τέλος, πραγματοποιείται ο έλεγχος του module (make px4 sitl gazebo εάν εκτελείται σε SITL). Η πληκτρολόγηση της εντολής assistance στο simulations shell θα πρέπει να αποκαλύπτει το όνομα του νέου module. Για τη δοκιμή του, αρκεί η εκτέλεση της εντολής module start.

Actuator outputs

Το topic actuator outputs περιλαμβάνει τις τιμές PWM που αποστέλλονται στις διάφορες θύρες του αυτόματου πιλότου . Οι τιμές κυμαίνονται συνήθως μεταξύ 1000 και 2000, με την τιμή 900 να είναι η προεπιλεγμένη για την κατάσταση disarmed. Ο αριθμός των PWM τιμών εξαρτάται από τον αυτόματο πιλότο για παράδειγμα, στον Pixracer αποστέλλονται μόνον 6 σήματα (4 κινητήρες και 2 επιπλέον ακίδες).

Δυστυχώς, δεν είναι εφικτό να καταχωρηθεί απευθείας μια τιμή PWM σε συγκεκριμένο ακροδέκτη χωρίς να επηρεαστούν τα υπόλοιπα· η μοναδική λύση είναι η αξιοποίηση του topic actuators controls.

Actuator controls

Το topic των actuator controls περιλαμβάνει τις τιμές ελέγχου για όλες τις πιθανές ομάδες ελέγχου. Αφορά κάτι περισσότερο από ένα απλό topic, καθώς περιλαμβάνει όλα τα “sub-topics” που σχετίζονται με τις διάφορες ομάδες ελέγχου . Το actuator_controls_0 περιλαμβάνει τις τιμές ελέγχου της ομάδας ελέγχου 0 (roll, pitch, yaw και thrust) [29].

Οι βασικές ομάδες ελέγχου είναι:

• Actuator controls 0: roll, pitch, yaw και thrust (multicopter).

• Actuator controls 1: roll, pitch, yaw και thrust (VTOL).

• Actuator controls 2: roll, pitch, yaw και shutter (gimbal). Το συγκεκριμένο topic μπορεί να χρησιμοποιηθεί μόνο όταν υπάρχουν AUX θύρες, δηλαδή όχι με τον αυτόματο πιλότο Pixracer.

• Actuator controls 3: manual passthrough. Χρήσιμες τιμές είναι οι 6η και 7η (δηλαδή τα στοιχεία 5 και 6), καθώς επιτρέπουν την αξιοποίηση των original mixer files και τη ρύθμιση της PWM εξόδου στα pins 5 και 6.

• Actuator controls virtual fw.

• Actuator controls virtual ms.

Input rc

Το topic input_rc περιλαμβάνει πληροφορίες σχετικά με τις εντολές του χειριστηρίου (radio commands) που παρέχει ο χρήστης, καθώς και επιπλέον σημάνσεις, όπως η απώλεια σήματος ή η το failsafe mode. Αυτό το topic λειτουργεί ως ενημερωτικός πόρος. Το module που ευθύνεται για την ενημέρωσή του είναι το rc_update.

Sensor_combined

Το topic sensor_combined λειτουργεί ως κόμβος συγκέντρωσης των τιμών των αισθητήρων. Αποτελείται από δύο κύρια στοιχεία:

• gyro_rad: ο μέσος γωνιακός ρυθμός που καταγράφεται στο σύστημα συντεταγμένων του σώματος (XYZ body frame) σε rad/s κατά τη διάρκεια της τελευταίας περιόδου δειγματοληψίας του γυροσκοπίου (η περίοδος δειγματοληψίας καθορίζεται εντός του topic, μεταβλητή gyro_integral_dt σε μς) [30].

• accelerometer_m_s2: η μέση τιμή της επιτάχυνσης που καταγράφεται στο σύστημα συντεταγμένων του σώματος (XYZ body frame) σε m/s^2 κατά τη διάρκεια της τελευταίας περιόδου δειγματοληψίας του επιταχυνσιόμετρου (η περίοδος δειγματοληψίας καθορίζεται εντός του topic, μεταβλητή accelerometer_integral_dt σε μs) [30].

Το topic sensor_combined συλλέγει δεδομένα από το module sensors και τα μεταβιβάζει στους navigator, commander, frsky_telemetry και mavlink.

Sensors module

Το module sensors παίζει έναν καθοριστικό ρόλο στο σύνολο του συστήματος. Αποκτά δεδομένα χαμηλού επιπέδου από τους οδηγούς, τα μετατρέπει σε πιο χρήσιμη μορφή (μέσω φιλτραρίσματος) και τα δημοσιεύει, επιτρέποντας στα υπόλοιπα modules να επωφεληθούν από καθαρές και αξιόπιστες μετρήσεις.

Οι λειτουργείες περιλαμβάνουν:

• Ανάγνωση των εξόδων από τους sensor drivers (sensor_gyro, sensor_accel, sensor_baro, sensor_mag, differential_pressure). Εάν υπάρχουν πολλαπλοί αισθητήρες του ίδιου τύπου, γίνεται διαδικασία voting και failover. Ακολουθεί η εφαρμογή της περιστροφής της πλακέτας και η θερμοκρασιακή βαθμονόμηση (εφόσον είναι ενεργοποιημένη). Τέλος, τα δεδομένα γίνονται publish, ένα από τα topics είναι το sensor_combined, το οποίο αξιοποιείται από πολλά μέρη του συστήματος [31].

• Διασφάλιση ότι οι sensor drivers αποκτούν καλιμπραρισμένες παραμέτρους βαθμονόμησης (κλίμακα και μετατόπιση) όταν οι παράμετροι τροποποιούνται ή κατά την εκκίνηση. Οι οδηγοί αισθητήρων χρησιμοποιούν το interface ioctl για την ενημέρωση παραμέτρων. Για την ορθή λειτουργία αυτού, οι οδηγοί αισθητήρων πρέπει να είναι ήδη ενεργοί κατά την εκκίνηση του module sensors [31].

• Εκτέλεση preflight consistency των αισθητήρων και δημοσίευση του topic sensor_preflight [31].

Vehicle_attitude

Το topic vehicle_attitude περιλαμβάνει δύο κρίσιμες παραμέτρους:

- q: το quaternion που αναπαριστά τη μετατροπή από το σύστημα συντεταγμένων του σώματος (XYZ body frame) στο σύστημα NED (earth frame).
- delta_q_reset: η αλλαγή του quaternion από την τελευταία επαναφορά.

Vehicle gps position

Το topic vehicle_gps_position περιέχει όλες τις πληροφορίες που προέρχονται από τον αισθητήρα GPS, όπως την ταχύτητα σε σχέση με το έδαφος, την ακρίβεια κ.λπ.

Vehicle control mode

Το topic vehicle_control_mode περιέχει μια λίστα από boolean τιμές που χρησιμοποιούνται για τον καθορισμό του είδους του control mode που εφαρμόζεται.

Vehicle status

Το topic vehicle_status περιλαμβάνει πληροφορίες σχετικά με την κατάσταση του drone, ειδικότερα αναφορικά με την κατάσταση arm/disarm, τον τύπο του οχήματος, την κατάσταση πλοήγησης κ.λπ. Δεδομένου ότι λειτουργεί ως κεντρικός κόμβος πληροφόρησης, πολλά modules εγγράφονται σε αυτό. Τα κυριότερα από αυτά είναι: mavlink, commander, navigator.

Vehicle command

Το topic vehicle_command περιλαμβάνει μια εκτενή λίστα εντολών (κάθε μία με συγκεκριμένο αριθμητικό αναγνωριστικό) που παρέχονται στο drone για την εκτέλεση συγκεκριμένων λειτουργιών.

Δεδομένου του υψηλού αριθμού εντολών που περιλαμβάνει, πολλά modules κάνουν subscribe σε αυτό το topic. Τα κύρια στοιχεία είναι: mavlink, fmu, px4io, uavcan, camera_trigger, commander, navigator.

Mount orientation

Το topic mount orientation περιλαμβάνει πληροφορίες σχετικά με την τοποθέτηση.

- Attitude_euler_angle: η στάση/κατεύθυνση του mount εκφρασμένη σε γωνίες Euler (rad).

Το mount_orientation αποκτά δεδομένα από το module vmount και τα κάνει publish μέσω mavlink.

Αυτό το topic χρησιμοποιείται αποκλειστικά όταν υπάρχουν AUX pins, δηλαδή δεν ισχύει για τον αυτόματο πιλότο Pixracer.

Camera trigger

Το topic camera_trigger περιλαμβάνει πληροφορίες σχετικά με τις λειτουργίες της κάμερας:

- Seq: ο αριθμός της ακολουθίας της εικόνας.
- Feedback: Boolean που χρησιμοποιείται για την επαλήθευση ότι η λήψη ολοκληρώθηκε σωστά.

Το camera_trigger συλλέγει δεδομένα από το module camera_trigger και τα κάνει publish στα mavlink και camera_feedback.

3.4 Mavlink

Ενώ το uORB διαχειρίζεται την επικοινωνία μεταξύ των εσωτερικών υποσυστημάτων του drone, το mavlink είναι το πρωτόκολλο μηνυμάτων που έχει σχεδιαστεί για το σύνολο του

οικοσυστήματος του drone, περιλαμβάνοντας το drone και τον ground control station. Συγκεκριμένα, το PX4 χρησιμοποιεί mavlink για την επικοινωνία με το QGroundControl και ως μηχανισμό ενσωμάτωσης για τη σύνδεση με εξωτερικά στοιχεία του drone: companion computers, κάμερες με mavlink υποστήριξη κ.λπ. [32].

Το πρωτόκολλο mavlink καθορίζει τη δομή των μηνυμάτων και τα κριτήρια σειριοποίησης στο επίπεδο εφαρμογής, προσδιορίζοντας τον τρόπο μεταφοράς των πληροφοριών μέσω του δικτύου. Αυτά τα μηνύματα προωθούνται στα κατώτερα επίπεδα για να μεταδοθούν στο δίκτυο.

Ένα από τα κύρια πλεονεκτήματα του mavlink είναι ότι, εξαιτίας της ελαφριάς του δομής, υποστηρίζει ποικιλία τύπων transport layers και μέσων μετάδοσης. Η μετάδοση μπορεί να πραγματοποιηθεί μέσω σειριακών τηλεμετρικών καναλιών χαμηλού εύρους ζώνης, που λειτουργούν στην περιοχή MHz , όπως 433 MHz, 868 MHz ή 915 MHz, καθώς και μέσω wifi και ethernet (TCP/IP δίκτυα). Η χρήση sub-GHz συχνοτήτων εξασφαλίζει μεγάλο εύρος επικοινωνίας για τον χειρισμό του μη επανδρωμένου συστήματος [26].

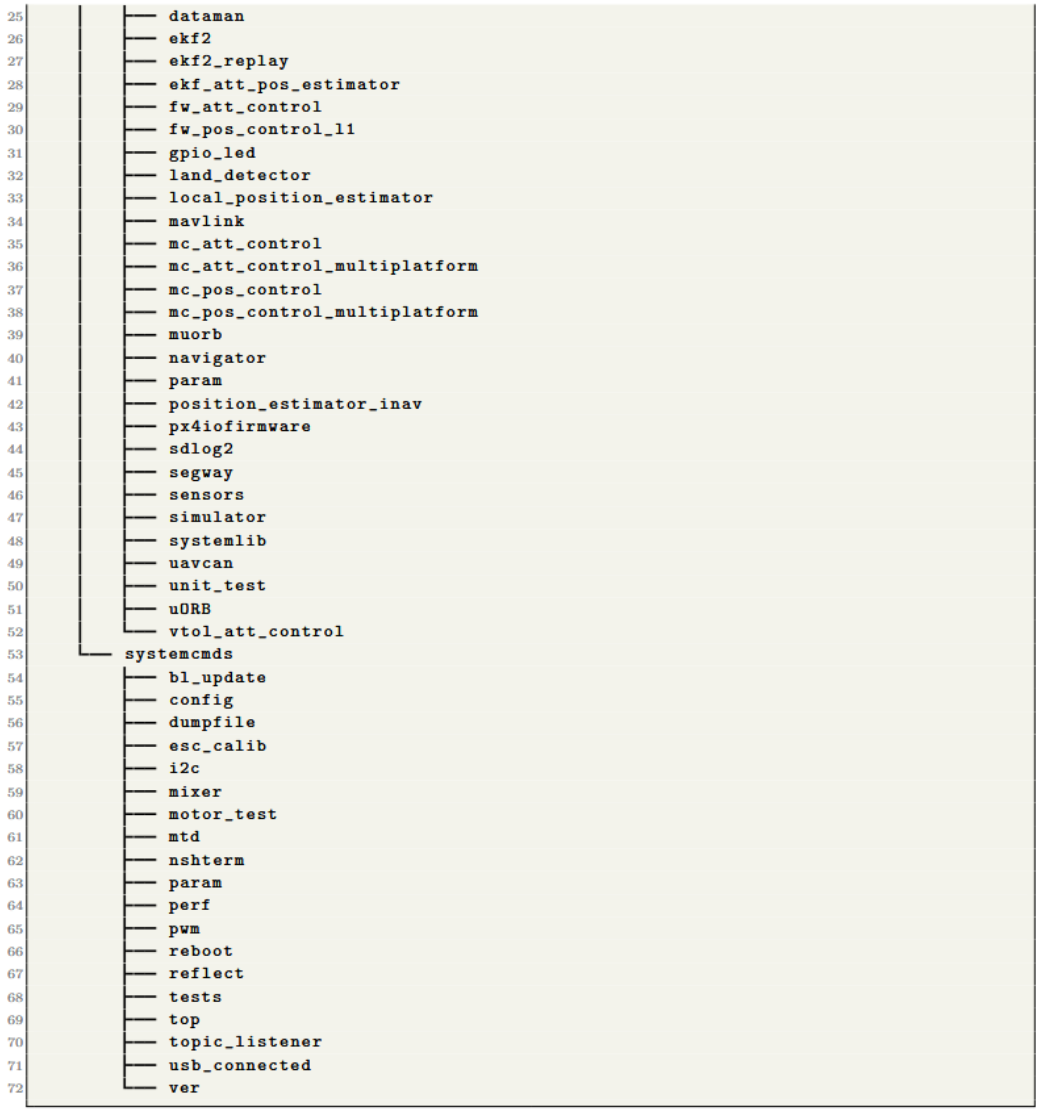
Αντιθέτως, μέσω διεπαφής wifi ή ethernet, τα μηνύματα mavlink μεταδίδονται μέσω δικτύων IP. Τόσο το πρωτόκολλο TCP όσο και το UDP μπορούν να χρησιμοποιηθούν, ανάλογα με την απαιτούμενη αξιοπιστία της εφαρμογής [33].

Η πρώτη έκδοση του mavlink (1.0) δημοσιεύθηκε από τον Lorenz Meier το 2009, ενώ η δεύτερη έκδοση με ουσιαστικές βελτιώσεις κυκλοφόρησε το 2017 (mavlink 2.0). Η επόμενη ενότητα εξετάζει τη δεύτερη έκδοση, η οποία θεωρείται η προτιμώμενη επιλογή.

3.5 Δομή λογισμικού

Η πιο σημαντική δομή του δέντρου παρουσιάζεται στο ακόλουθο δενδροειδές σχήμα:

```
Firmware
├── cmake
├── msg
│   └── templates
│       ├── px4
│       └── uorb
├── ROMFS
│   ├── px4fmu_common
│   │   ├── init.d
│   │   ├── logging
│   │   └── mixers
│   └── px4fmu_test
│       ├── init.d
│       ├── mixers
│       └── unit_test_data
└── src
    ├── drivers
    ├── examples
    ├── modules
    │   ├── attitude_estimator_ekf
    │   ├── attitude_estimator_q
    │   ├── bottle_drop
    │   ├── commander
    │   ├── controllib_test
```



Εικόνα 3.5 Δομή λογισμικού [34] .

- cmake: αρχεία make.
- msg: πρότυπα uORB μηνυμάτων· τα uORB message headers παράγονται από αυτόν τον φάκελο.
- ROMFS: startup scripts, αρχεία ρύθμισης για τα airframes και mixers.
- src/drivers: περιέχει όλους τους drivers (pwm, gyro, gps κλπ.).
- src/examples: απλά παραδείγματα που βοηθούν στην κατανόηση του κώδικα.
- src/modules: estimators, controllers, mavlink, vmount κλπ.

3.5.1 System commands (systemcdms)

config

Επιτρέπει τη ρύθμιση ενός sensor driver (παράμετροι όπως ο ρυθμός δειγματοληψίας και επικοινωνίας, το εύρος κ.λπ.). Οι αισθητήρες που υποστηρίζονται είναι: γυροσκόπιο, επιταχυνσιόμετρα και μαγνητόμετρο.

```
config <command> [arguments...]
 Commands:

 The <file:dev> argument is typically one of /dev/{gyro,accel,mag}i
   block         Block sensor topic publication
     <file:dev>  Sensor device file

   unblock       Unblock sensor topic publication
     <file:dev>  Sensor device file

   sampling      Set sensor sampling rate
     <file:dev> <rate> Sensor device file and sampling rate in Hz

   rate          Set sensor publication rate
     <file:dev> <rate> Sensor device file and publication rate in Hz

   range         Set sensor measurement range
     <file:dev> <rate> Sensor device file and range

   check         Perform sensor self-test (and print info)
     <file:dev>  Sensor device file
```

Εικόνα 3.6 Config arguments [34].

ESC calibration (esc_calib)

Πρόκειται για εργαλείο βαθμονόμησης των ESC. Η διαδικασία βαθμονόμησης είναι η εξής:

• Αφαίρεση ελίκων και διακοπή τροφοδοσίας των ESC.

• Διακοπή attitude controllers: mc_att_control stop, fw_att_control stop.

• Safety πρέπει είναι απενεργοποιημένο.

•Εκτέλεση εντολής.

PWM (pwm)

Πρόκειται για την εντολή που χρησιμοποιείται για τη ρύθμιση των PWM εξόδων για τον έλεγχο servos και ESC.

Η προεπιλεγμένη συσκευή /dev/pwm_output0 αφορά τα main channels, ενώ τα AUX channels βρίσκονται στο /dev/pwm_output1 (με τη χρήση της παραμέτρου -d).

Running processes (top)

Παρακολουθεί τις διεργασίες που εκτελούνται, καθώς και τη χρήση CPU, τη χρήση stack, την προτεραιότητα και την κατάστασή τους.

3.5.2 Communication

Mavlink

Αναφέρεται στο module που εκτελεί το πρωτόκολλο mavlink. Επικοινωνεί με το σύστημα μέσω uORB: ορισμένα μηνύματα υλοποιούνται και διαχειρίζονται απευθείας εντός του module για παράδειγμα το mission protocol, ενώ άλλα απαιτούν δημοσίευση μέσω uORB όπως το vehicle_command.

Η εκτέλεση περιλαμβάνει 2 threads, ένα για αποστολή και ένα για λήψη. Ο αποστολέας λειτουργεί με σταθερό ρυθμό και μειώνει δυναμικά τους ρυθμούς των ροών όταν ο συνδυασμένος ρυθμός μετάδοσης υπερβαίνει την καθορισμένη τιμή (-r) ή όταν ο φυσικός σύνδεσμος φτάνει σε κορεσμό. Αυτό μπορεί να επιβεβαιωθεί με την εντολή mavlink status, ελέγχοντας αν το rate mult είναι πιο λίγο του 1 λεπτού [34].

uORB

Το module που υλοποιεί τον μηχανισμό επικοινωνίας publish και subscribe του uORB είναι το uorb. Αυτό το module ενεργοποιείται κατά την εκκίνηση του συστήματος, καθώς τα περισσότερα από τα υπόλοιπα modules εξαρτώνται από αυτό για την ορθή λειτουργία τους. Δεδομένου ότι η επικοινωνία διεξάγεται μέσω κοινής μνήμης, η εκκίνηση του module απλώς διασφαλίζει την αρχικοποίηση της κοινής καθολικής κατάστασης. Η υλοποίηση είναι ασύγχρονη και χωρίς κλειδώματα, δηλαδή ένας publisher δεν αναμένει έναν συνδρομητή και το αντίστροφο. Αυτό επιτυγχάνεται μέσω ενός διακριτού buffer μεταξύ publisher και subscriber.

Τα μηνύματα καθορίζονται στον φάκελο /msg και μετατρέπονται σε κώδικα C/C++ κατά τη διάρκεια της διαδικασίας κατασκευής.

Η εντολή top μπορεί να χρησιμοποιηθεί και για το uORB, και είναι εξαιρετικά χρήσιμη για την παρακολούθηση του ρυθμού δημοσίευσης των θεμάτων [34].

```
uorb <command> [arguments...]
Commands:
  start

  status        Print topic statistics

  top           Monitor topic publication rates
    [-a]        print all instead of only currently publishing topics
    [-1]        run only once, then exit
    [<filter1> [<filter2>]] topic(s) to match (implies -a)
```

Εικόνα 3.7 uORB arguments [34].

3.5.3 Εκλεκτές

Ελεγκτές fixed wing controllers

Οι ελεγκτές fixed wing χωρίζονται σε δύο κατηγορίες :

• fw_att_control: ελεγκτής στάσης για fixed-wing.

• fw_pos_control_l1: ελεγκτής θέσης για fixed-wing.

Ελεγκτές multicopter controllers

Οι ελεγκτές multicopter controllers χωρίζονται επίσης σε δύο κατηγορίες:

• mc_att_control: ελεγκτής στάσης για multicopter.

• mc_pos_control: ελεγκτής θέσης για multicopter.

Navigator

Το navigator είναι το module που ευθύνεται για τις αυτόνομες λειτουργίες πτήσης. Παραδείγματα τέτοιων λειτουργιών πτήσης είναι οι αποστολές, η απογείωση και η επιστροφή στο σημείο εκκίνησης (RTL). Επιπλέον, το navigator είναι υπεύθυνο για την παρακολούθηση παραβιάσεων geo-fence. Οι ποικίλες εσωτερικές λειτουργίες υλοποιούνται ως διακριτές κλάσεις που κληρονομούν από μια κοινή βασική κλάση, την NavigatorMode. Το navigation_mode περιλαμβάνει το τρέχον ενεργό mode.

Η έξοδος του navigator αποτελεί ένα triplet position setpoint (position_setpoint_triplet_s), το οποίο απαιτείται ως είσοδος από τον ελεγκτή θέσης [34].

Εικόνα 3.8 Navigator arguments [34].

3.5.4 Drivers

FMU driver

FMU driver (px4fmu) είναι το module που ελέγχει τις εξόδους βάσει των εισερχόμενων τιμών ελέγχου. Για πλακέτες που δεν διαθέτουν ξεχωριστό IO chip όπως το Pixracer, χρησιμοποιούνται τα κύρια κανάλια. Αντίθετα, για πλακέτες με IO chip όπως το Pixhawk, χρησιμοποιεί τα AUX channels, ενώ ο driver px4io χρησιμοποιείται για τα Main channels.

Το px4fmu παρακολουθεί τα actuator_controls δηλαδή Firmware/build/px4_fmu-v4_default/uORB/topics, εκτελεί τη διαδικασία mixing σύμφωνα με τα αρχεία mixer στην κάρτα microSD και καταγράφει τις εξόδους PWM .

Το module ρυθμίζεται μέσω των εντολών λειτουργίας. Αυτό προσδιορίζει ποια από τα πρώτα N pins θα αξιοποιηθούν από τον οδηγό. Για παράδειγμα, με τη χρήση του mode pwm4, τα pin 5 και 6 μπορούν να αξιοποιηθούν από τον οδηγό ενεργοποίησης κάμερας ή από έναν οδηγό PWM rangefinder. Εναλλακτικά, το fmu μπορεί να ενεργοποιηθεί σε μία από τις λειτουργίες λήψης, και οι οδηγοί μπορούν να εγγραφούν με callback μέσω callback ioctl.

Κατά την προεπιλογή, το module λειτουργεί στη work queue, προκειμένου να μειωθεί η κατανάλωση RAM. Μπορεί επίσης να εκτελεστεί σε δικό του thread, καθορίζοντας το σημαία -t, προκειμένου να μειωθεί η καθυστέρηση. Κατά την εκτέλεση στο work queue, προγραμματίζεται με σταθερή συχνότητα, και ο ρυθμός PWM περιορίζει την ταχύτητα ενημέρωσης των θεμάτων actuator_controls. Εάν εκτελείται σε δικό του thread, το module πραγματοποιεί polling στο topic actuator_controls. Επιπλέον, ο ρυθμός PWM καθορίζει τους χαμηλούς ρυθμούς των χρονομετρητών IO.

GPS

Ο GPS driver είναι υπεύθυνος για το publish της θέσης μέσω uORB, σύμφωνα με τα δεδομένα που προέρχονται από τον αισθητήρα GPS. Ανάλογα με τον κατασκευαστή του αισθητήρα, ενδέχεται να υποστηρίζει ποικίλα πρωτόκολλα ωστόσο, από προεπιλογή επιλέγει αυτόματα το πλέον κατάλληλο. Είναι επίσης εφικτό να καθοριστεί μια δευτερεύουσα συσκευή GPS μέσω της παραμέτρου -e. Οι πληροφορίες από αυτόν τον δευτερεύοντα αισθητήρα δεν χρησιμοποιούνται για τον προσδιορισμό της θέσης του drone καταγράφονται αποκλειστικά για μελλοντική ανάλυση, προκειμένου να αξιολογηθεί η ακρίβεια του κύριου GPS.

Οι τιμές GPS χρησιμοποιούνται απευθείας για την εκτίμηση θέσης και στάσης, χωρίς να διέρχονται από το sensor hub.

Κάθε συσκευή διαθέτει ένα thread που εκτελεί polling για δεδομένα. Οι κατηγορίες των πρωτοκόλλων GPS υλοποιούνται μέσω callbacks, προκειμένου να είναι επαναχρησιμοποιήσιμες σε άλλα project όπως το QGroundControl.

```
gps <command> [arguments...]
 Commands:
   start
     [-d <val>]  GPS device
                 values: <file:dev>, default: /dev/ttyS3
     [-b <val>]  Baudrate (can also be p:<param_name>)
                 default: 0
     [-e <val>]  Optional secondary GPS device
                 values: <file:dev>
     [-g <val>]  Baudrate (secondary GPS, can also be p:<param_name>)
                 default: 0
     [-f]        Fake a GPS signal (useful for testing)
     [-s]        Enable publication of satellite info
     [-i <val>]  GPS interface
                 values: spi|uart, default: uart
     [-p <val>]  GPS Protocol (default=auto select)
                 values: ubx|mtk|ash|eml

   stop

   status        print status info

   reset         Reset GPS device
     cold|warm|hot Specify reset type
```

Εικόνα 3.9 GPS arguments [34].

Simulation PWM signals (pwm_out_sim)

Drivers για προσομοιωμένες PWM εξόδους. Η μοναδική του λειτουργία είναι να λαμβάνει τα uORB μηνύματα actuator_control, να τα κάνει mixing με τα mixers που έχουν φορτωθεί και να κάνει publish το αποτέλεσμα στο topic actuator_output uORB.

Χρησιμοποιείται τόσο σε SITL όσο και σε HITL [34].

Gimbal control

Driver για έλεγχο του gimbal. Χαρτογραφεί ποικίλες μεθόδους εισόδου για παράδειγμα RC ή mavlink σε μια συγκεκριμένη μέθοδο εξόδου όπως AUX channels ή mavlink. Αυτή η προσέγγιση διασφαλίζει ότι κάθε μέθοδος εισόδου είναι συμβατή με κάθε μέθοδο εξόδου, ενώ νέες είσοδοι και έξοδοι μπορούν να προστεθούν με ελάχιστη προσπάθεια [34].

Boot process

Κατά την ενεργοποίηση της πλακέτας, εκτελείται πρώτα ο bootloader. Ο bootloader λειτουργεί παρόμοια με το BIOS σε έναν υπολογιστή και είναι ήδη προ εγκατεστημένος στην πλακέτα κατά την αγορά της. Ο bootloader εκκινεί το λειτουργικό σύστημα NuttX. Αφού ολοκληρωθεί η αρχικοποίηση του υλικού, της μνήμης, των περιφερειακών κ.λπ., εκτελούνται τα εξής θεμελιώδη βήματα:

- Ανάγνωση του αρχείου παραμέτρων.
- Εκκίνηση των sensor drivers (script rc.sensors).
- Ρύθμιση και φόρτωση του mixer που αντιστοιχεί στην παράμετρο airframe SYS_AUTOSTART, καθώς και ρύθμιση των PWM channel (script rc.autostart, το οποίο δημιουργείται μετά την κατασκευή του κώδικα).
- Εκκίνηση των πτήσεων που αντιστοιχούν στην παράμετρο SYS_AUTOSTART (scripts όπως rc.fw_apps, rc.mc_apps κ.λπ.) [34] .

ΚΕΦΑΛΑΙΟ 4

Visual language action system

4.1 Μοντέλα βάση γλώσσας

Η πρόοδος των μεγάλων γλωσσικών μοντέλων άρχισε με την αρχιτεκτονική transformer, η οποία εισήγαγε πιο αποδοτικούς μηχανισμούς επεξεργασίας ακολουθιών μέσω της προσοχής. Σε αυτή τη βάση αναπτύχθηκαν μοντέλα όπως το BERT, το οποίο εκπαιδεύεται σε εκτενή κείμενα "μαντεύοντας" κρυφές λέξεις, προκειμένου να κατανοήσει καλύτερα τα συμφραζόμενα και να βελτιώσει την απόδοσή του σε ποικίλες γλωσσικές εργασίες. Ο universal sentence encoder έχει σχεδιαστεί για να παρέχει αξιόπιστες αναπαραστάσεις προτάσεων, κατάλληλες για ποικίλες εφαρμογές. Το T5, αντιθέτως, ενοποίησε όλες τις εργασίες φυσικής γλώσσας μετατρέποντάς τες σε προβλήματα τύπου "κείμενο σε κείμενο", εκπαιδεύτηκε σε εκτενή δεδομένα και καθιερώθηκε ως ένα μοντέλο γενικού σκοπού για την παραγωγή και κατανόηση κειμένου. Μέσω αυτών των μοντέλων, η αρχιτεκτονική transformer καθιερώθηκε ως το θεμέλιο για την ανάπτυξη των σύγχρονων γλωσσικών τεχνολογιών [35].

Αντίθετα με άλλα γλωσσικά μοντέλα, τα μοντέλα GPT προσεγγίζουν όλες τις εργασίες NLP ως ένα ενιαίο πρόβλημα πρόβλεψης της επόμενης λέξης. Αυτό οδήγησε στη χρήση αρχιτεκτονικών Transformer που βασίζονται αποκλειστικά σε αποκωδικοποιητές, γνωστών και ως αυτοπαραγωγικοί Transformers. Με την αύξηση του μεγέθους των μοντέλων και την προεκπαίδευση σε δεδομένα διαδικτύου, το GPT-3, με 175 δισεκατομμύρια παραμέτρους, παρουσίασε εντυπωσιακές ικανότητες στην κατανόηση και παραγωγή φυσικής γλώσσας. Ενδιαφέρον είναι το γεγονός ότι παρουσίασε αναδυόμενες συμπεριφορές, όπως το in-context learning, όπου το μοντέλο μπορεί να εκτελεί εργασίες βασιζόμενο αποκλειστικά σε ελάχιστα παραδείγματα που παρέχονται κατά τη διάρκεια της πρόβλεψης. Αυτό υποδεικνύει ότι ο συνδυασμός εκτενών αρχιτεκτονικών, κατάλληλων εκπαιδευτικών στόχων και εκτενών πηγών δεδομένων οδηγεί στη δημιουργία γενικών μοντέλων που υπερβαίνουν τα εξειδικευμένα συστήματα. Η μετατόπιση αυτή συνάδει με το κύριο συμπέρασμα του Bitter Lesson και υποδηλώνει την αρχή της εποχής των μεγάλων γλωσσικών μοντέλων (LLMs).

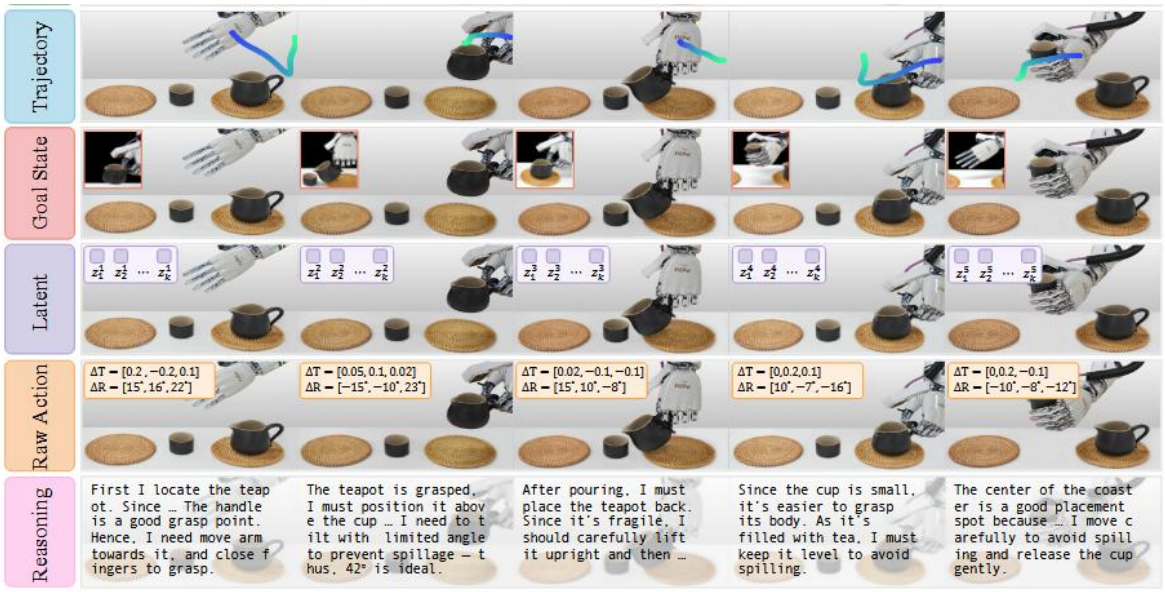
Για την αποτελεσματική κλιμάκωση των LLMs έχουν προταθεί νόμοι κλιμάκωσης που περιγράφουν προβλέψιμες σχέσεις μεταξύ του μεγέθους του μοντέλου, του όγκου δεδομένων, του υπολογιστικού κόστους και της προεκπαιδευτικής απώλειας. Τα ευρήματα αυτά

κατευθύνουν πρακτικές αποφάσεις σχετικά με τον σχεδιασμό μοντέλων και την κατανομή πόρων κατά την εκπαίδευση σε μεγάλη κλίμακα. Το InstructGPT προχώρησε ένα βήμα παραπέρα στην ευθυγράμμιση των LLMs με ανθρώπινες προθέσεις, εφαρμόζοντας επιτηρούμενη βελτίωση σε σύνολα δεδομένων με οδηγίες και στη συνέχεια ενίσχυση μέσω μάθησης από ανθρώπινη ανατροφοδότηση RLHF. Από τότε, οι τεχνικές ευθυγράμμισης έχουν αναλυθεί εκτενώς, με σκοπό την εξασφάλιση ότι τα μεγάλα μοντέλα τεχνητής νοημοσύνης λειτουργούν σύμφωνα με ζητήματα ασφάλειας, ανθρώπινες προτιμήσεις και αξίες.

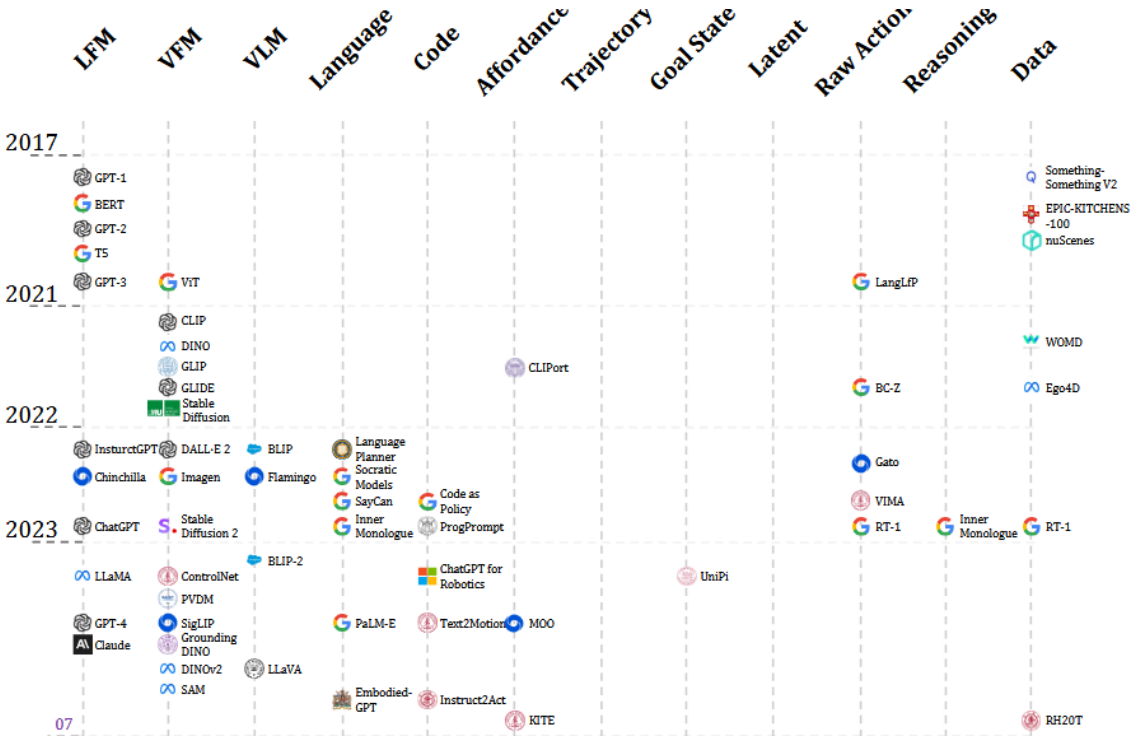
Οι τεχνολογικές εξελίξεις αυτές οδήγησαν στην ανάπτυξη εξαιρετικά ικανών εμπορικών LLMs, όπως το GPT-4 και το Claude, τα οποία επιδεικνύουν υψηλή απόδοση σε ανοιχτού τύπου διάλογο, παραγωγή κώδικα και συλλογιστική chain-of-thought.

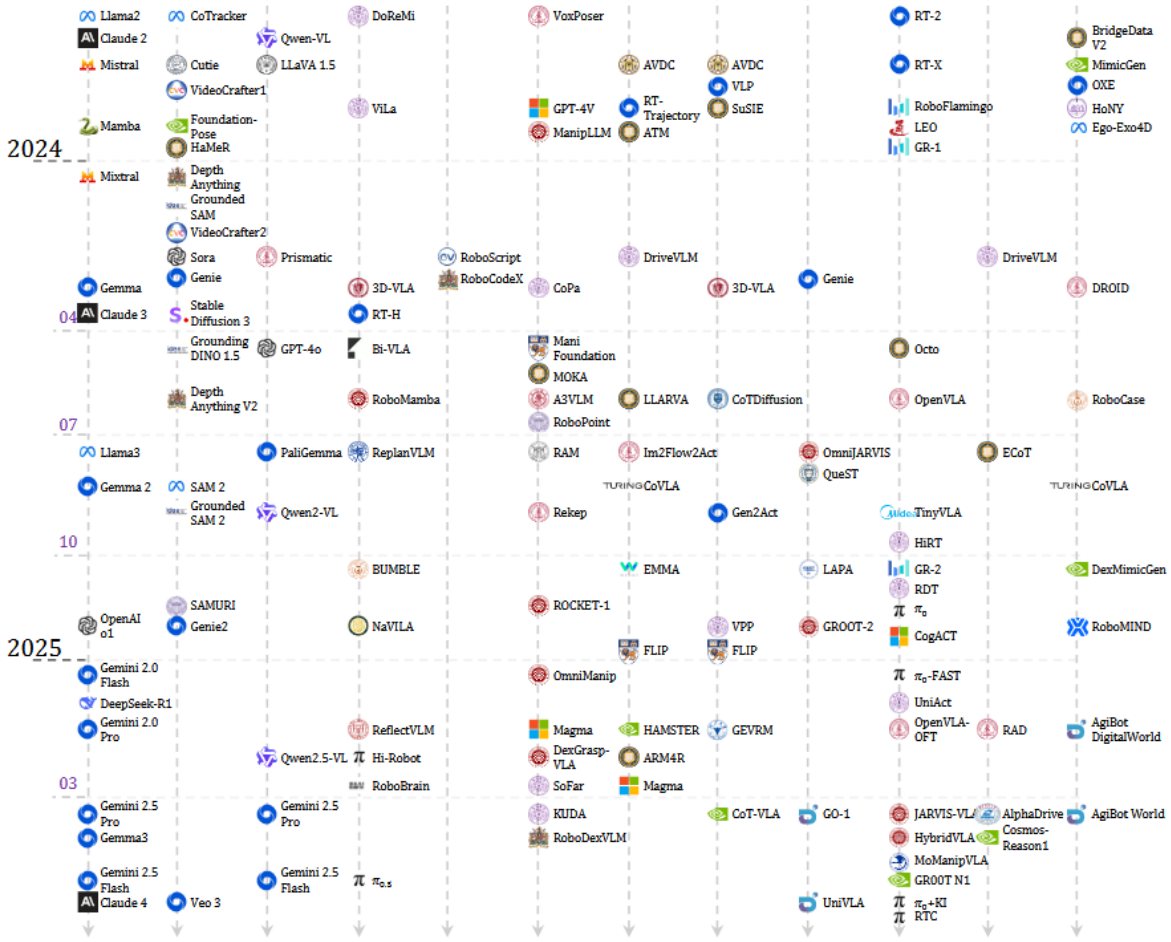
Ενσωματωμένα σε έναν εξελικτικό coding agent με την ονομασία AlphaEvolve, τα Gemini 2.0 Flash και Gemini 2.0 Pro διευκόλυναν σημαντικές προόδους σε ανοικτά επιστημονικά ζητήματα, όπως ο υπολογιστικά βέλτιστος πολλαπλασιασμός πινάκων.

Ωστόσο, εξαιτίας του κλειστού τους χαρακτήρα και της περιορισμένης πρόσβασης μέσω API, αυτά τα μοντέλα παραμένουν δυσχερή στην επιθεώρηση, στην λεπτομερή προσαρμογή και στην ενσωμάτωσή τους σε ευρύτερες ερευνητικές και βιομηχανικές ροές εργασίας.



Εικόνα 4.1 Οπτικοποίηση των action tokens σε μία μεμονωμένη ενσωματωμένη εργασία [35].





Εικόνα 4.2 Χρονοδιάγραμμα εξέλιξης των foundation models, των VLA μοντέλων και των data sources. Το σχήμα σε μορφή "U" αποτυπώνει το πώς η αυξανόμενη ανάπτυξη των VLA μοντέλων υποστηρίζεται από την πρόοδο τόσο των foundation models όσο και των δεδομένων [35].

Για να αντιμετωπιστούν οι παραπάνω περιορισμοί, κυκλοφόρησε μια σειρά από ανοιχτού κώδικα LLMs, όπως τα Llama, Gemma και Mistral, με μεγέθη που κυμαίνονται από 2B έως 70B παραμέτρους ώστε να καλύπτουν διαφορετικές ανάγκες.

Πάνω σε αυτά τα μοντέλα έχουν αναπτυχθεί τεχνικές parameter-efficient fine-tuning, όπως το LoRA, οι οποίες επιτρέπουν την προσαρμογή του μοντέλου σε συγκεκριμένες εργασίες με πολύ μικρότερο αριθμό εκπαιδεύσιμων παραμέτρων και χαμηλότερο υπολογιστικό κόστος. Αυτό καθιστά το fine-tuning εφικτό ακόμη και σε περιβάλλοντα με περιορισμένους πόρους.

Για την περαιτέρω αύξηση της χωρητικότητας των μοντέλων χωρίς αντίστοιχη αύξηση του υπολογιστικού κόστους, εισήχθησαν στις LLM αρχιτεκτονικές τα μοντέλα Mixture-of-Experts , όπως το Switch Transformer και το Mixtral. Τα μοντέλα MoE ενεργοποιούν μόνο ένα

υποσύνολο από υποδίκτυα ειδικών για κάθε είσοδο, επιτρέποντας σημαντικά μεγαλύτερη αποτελεσματική χωρητικότητα μοντέλου ενώ διατηρούν αποδοτική εκτίμηση.

Παράλληλα, για να αντιμετωπιστεί η τετραγωνική πολυπλοκότητα χρόνου της αρχιτεκτονικής transformer, προτάθηκαν εναλλακτικά μοντέλα όπως το Mamba. Το Mamba αντικαθιστά την αυτοπροσοχή με επιλεκτικές ενημερώσεις χώρου καταστάσεων, επιτρέποντας γραμμική χρονική επεξεργασία ακολουθιών, διατηρώντας ταυτόχρονα υψηλή απόδοση σε μεγάλες διαστάσεις συμφραζομένων.

Μια εναλλακτική ερευνητική κατεύθυνση επικεντρώνεται στη βελτίωση των ικανοτήτων συλλογισμού μέσω της ενίσχυσης του υπολογισμού κατά τη διάρκεια της φάσης πρόβλεψης. Ενδεικτικά παραδείγματα είναι τα μοντέλα OpenAI o1 και DeepSeek-R1, τα οποία κατανέμουν δυναμικά περισσότερους υπολογιστικούς πόρους κατά τη διαδικασία inference, με σκοπό τη βελτίωση της απόδοσης σε σύνθετα καθήκοντα λογικής και συλλογισμού. Ειδικότερα, το DeepSeek-R1 αποκτά αυτή την ικανότητα μέσω εκπαίδευσης μεγάλης κλίμακας με reinforcement learning, βασισμένο στη μέθοδο GRPO.

Σημαντική πρόοδος έχει επιτευχθεί στην βελτιστοποίηση της υποδομής για την εκπαίδευση και ανάπτυξη LLMs. Μια σειρά στρατηγικών παραλληλισμού, όπως το data parallelism, το model parallelism, το pipeline parallelism και το tensor parallelism, χρησιμοποιούνται ενεργά για την κλιμάκωση της εκπαίδευσης σε κατανεμημένα υπολογιστικά περιβάλλοντα.

Ταυτόχρονα, έχουν αναπτυχθεί τεχνικές επιτάχυνσης του inference, όπως η μοντελοποίηση με μειωμένη αριθμητική ακρίβεια (quantization), η αφαίρεση περιττών βαρών (weight pruning) και η speculative decoding, με σκοπό τη μείωση της καθυστέρησης και του υπολογιστικού κόστους κατά την ανάπτυξη ενός μοντέλου.

Οι εξελίξεις αυτές καθιστούν τα LLMs εξαιρετικά ικανά στη γνώση, τον διάλογο, την παραγωγή κώδικα και τη συλλογιστική, ενώ παράλληλα διευκολύνουν την αποδοτική εκπαίδευση, ανάπτυξη και fine-tuning μέσω ανεπτυγμένων υποδομών. Επιπλέον, βελτιώνουν σημαντικά τη χρηστικότητα των LLMs και υποστηρίζουν την ανάπτυξη συστημάτων όρασης και πολυτροπικών μοντέλων, αποτελώντας βασικά δομικά στοιχεία για τα embodied VLA συστήματα.

4.1.1 Θεμελιώδη μοντέλα όρασης (VFMs)

Η επιτυχία της αρχιτεκτονικής Transformer στον τομέα της γλώσσας οδήγησε την κοινότητα της υπολογιστικής όρασης να αντικαταστήσει τα παραδοσιακά συνελικτικά

νευρωνικά δίκτυα με το Vision Transformer ως θεμελιώδες υπόβαθρο των μοντέλων όρασης, επιτυγχάνοντας ανώτερη απόδοση κατά την εκπαίδευση με εκτενή σύνολα δεδομένων. Η εν λόγω αρχιτεκτονική θεωρεί τις εικόνες ως ακολουθίες οπτικών συμβόλων, επιτρέποντας την επεξεργασία των οπτικών δεδομένων με τρόπο παρόμοιο με αυτόν των κειμένων, διευκολύνοντας την ευθυγράμμιση και τη σύντηξη μεταξύ διαφορετικών μορφών σε μεταγενέστερα πολυτροπικά μοντέλα [35].

Συγχρόνως, η επιτυχία των LLMs προκάλεσε μια αντίστοιχη αναζήτηση κλιμακωτών στόχων μάθησης στη Vision, με σκοπό την εκπαίδευση γενικών μοντέλων σε εκτενή σύνολα οπτικών δεδομένων χωρίς ανθρώπινη επισημείωση. Ένα από τα πρώτα και πιο επιτυχημένα παραδείγματα είναι το CLIP, το οποίο αξιοποιεί επιτήρηση φυσικής γλώσσας εκπαιδεύοντας ένα μοντέλο σε 400 εκατομμύρια ζεύγη εικόνας-κειμένου με χρήση contrastive loss. Αυτό επιτρέπει στο CLIP να αποκτά ισχυρές και γενικεύσιμες αναπαραστάσεις εικόνων, επιδεικνύοντας εντυπωσιακές δυνατότητες zero-shot.

Το SigLIP αναβαθμίζει το CLIP αντικαθιστώντας το softmax με sigmoid loss, βελτιώνοντας την αποδοτικότητα και την απόδοση. Και τα δύο μοντέλα χρησιμοποιούνται εκτενώς ως κωδικοποιητές εικόνας , ιδίως σε πολυτροπικά συστήματα, λόγω της συνδυασμένης εκπαίδευσής τους με κείμενο. Ωστόσο, αυτή η εξάρτηση από γλωσσική επίβλεψη συνιστά και περιορισμό: οι περιγραφές είναι συχνά αφηρημένες και υψηλού επιπέδου, με αποτέλεσμα οι εικόνες να κωδικοποιούνται χωρίς τις λεπτομερείς πληροφορίες σε επίπεδο pixel, οι οποίες είναι απαραίτητες σε ορισμένες οπτικές εργασίες.

Για την επίλυση αυτού του ζητήματος, το DINO εκπαιδεύεται απευθείας σε επιλεγμένα σύνολα εικόνων με αυτοεπιβλεπόμενη μέθοδο, αποκτώντας πλούσια και γενικής χρήσης οπτικά χαρακτηριστικά, χρήσιμα για λεπτομερείς downstream εργασίες, όπως η semantic segmentation και depth estimation. Είναι σημαντικό ότι οι αναπαραστάσεις του μπορούν να αναγνωρίζουν ομοειδείς περιοχές σε διάφορα αντικείμενα για παράδειγμα φτερά αεροπλάνου και πουλιού, αποδεικνύοντας βαθιά σημασιολογική κατανόηση.

Αναφερόμενη σε αυτές τις καινοτόμες τεχνικές οπτικής κωδικοποίησης, η επόμενη γενιά ερευνών ανέπτυξε θεμελιώδη μοντέλα προσαρμοσμένα σε οπτικές εργασίες όπως η εκτίμηση βάθους και η τμηματοποίηση εικόνας και βίντεο. Το depth anything αξιοποιεί αυτοπαραγόμενα pseudo labels από εκτενή μη επισημασμένα δεδομένα για αξιόπιστη μονοοπτική εκτίμηση βάθους , ενώ το depth anything V2 ενισχύει τη λεπτομερή ανακατασκευή χρησιμοποιώντας δεδομένα αληθείας από συνθετικά σύνολα.

Το segment anything model λειτουργεί ως θεμελιώδες μοντέλο για τμηματοποίηση βάσει prompts, ενώ ο διάδοχός του, το SAM 2 , επεκτείνει αυτή τη δυνατότητα και στο βίντεο. Τα δύο μοντέλα είναι ικανά να παραγάγουν έγκυρες μάσκες segmentation από προτροπές όπως σημεία, bounding boxes, μάσκες και στην περίπτωση του SAM prompts κειμένου.

Το Cutie είναι ένα πρωτοποριακό μοντέλο για την τμηματοποίηση αντικειμένων σε βίντεο και έχει αποδείξει ανθεκτικότητα σε διάφορες οπτικές συνθήκες . Το SAMURAI ενισχύει την απόδοση του visual object tracking του SAM 2 μέσω της ενσωμάτωσης motion modeling και motion-aware μνήμης, διευκολύνοντας τη βελτιωμένη διαχείριση ταχείας κίνησης, απόκρυψης και πυκνών σκηνών. Επιπροσθέτως, το CoTracker εισάγει μια αρχιτεκτονική transformer για πυκνή παρακολούθηση σημείων σε εκτενείς ακολουθίες βίντεο, ενισχύοντας περαιτέρω τις δυνατότητες ανάλυσης δυναμικού οπτικού περιεχομένου.

Στον τομέα της ανίχνευσης και θεμελίωσης open-vocabulary detection, έχει αναπτυχθεί μια σειρά μοντέλων που έχουν σταδιακά βελτιώσει την κατανόηση των οπτικό-γλωσσικών σχέσεων σε επίπεδο περιοχών. Το GLIP συνδυάζει την ανίχνευση και την τοποθέτηση φράσεων σε ένα ενιαίο πλαίσιο προεκπαίδευσης, επεκτείνοντας την ευθυγράμμιση τύπου CLIP σε επίπεδο περιοχών. Το grounding DINO στηρίζεται σε αρχιτεκτονική DETR και σε συγκριτική ευθυγράμμιση κειμένου–περιοχής, επιτυγχάνοντας εξαιρετικές επιδόσεις. Το grounding DINO 1.5 ενισχύει το μοντέλο με αυξημένη κλίμακα δεδομένων και παραμέτρων, επιτυγχάνοντας νέα κορυφαία αποτελέσματα. Το grounded SAM συνδυάζει το grounding DINO με το SAM, διευκολύνοντας την zero-shot segmentation καθοδηγούμενη από γλώσσα. Το grounded SAM 2 διευρύνει αυτές τις δυνατότητες και στο βίντεο για την προσγείωση και παρακολούθηση οποιουδήποτε αντικειμένου.

Για την παραγωγή εικόνας και βίντεο υψηλής πιστότητας, τα μοντέλα διάχυσης έχουν επικρατήσει. Πρώιμα μοντέλα όπως τα GLIDE, DALL·E 2 και Imagen ανέδειξαν τη δύναμη της γλωσσικά καθοδηγούμενης σύνθεσης εικόνας, ενώ το Stable Diffusion κατέστησε δυνατή την αποδοτική και ευρεία δημιουργία εικόνων. Το ControlNet εισήγαγε χωρικούς περιορισμούς, επιτρέποντας λεπτομερή έλεγχο της δομής και της διάταξης.

Στην παραγωγή βίντεο, μοντέλα όπως τα VideoCrafter και PVDM διευρύνουν τη διάχυση στο χρονικό πεδίο. Το Sora προχωρά περαιτέρω αξιοποιώντας τη μέθοδο flow-matching ενσωματώνοντας φυσικούς περιορισμούς, προκειμένου να παράγει μακράς διάρκειας, υψηλής ανάλυσης βίντεο με ισχυρή χρονική συνοχή. Πρόσφατα, το Veo 3 επιδεικνύει εντυπωσιακή παραγωγή πλήρους πολυτροπικότητας, περιλαμβάνοντας συγχρονισμένο ήχο και κίνηση.

Αυτά τα προηγμένα μοντέλα εικόνας και βίντεο συχνά αποκαλούνται global models, καθώς ενσωματώνουν εκτενή φυσική γνώση και κατανόηση του πραγματικού κόσμου. Τα world models, όπως τα Genie και Genie 2, προσομοιώνουν τη μελλοντική οπτική δυναμική βάσει ακολουθιών ενεργειών, επιτρέποντας συνεκτικές και ακριβείς προβολές της εξέλιξης του περιβάλλοντος στο χρόνο.

Επιπλέον, άλλες πρωτοβουλίες επικεντρώνονται στην ανάπτυξη θεμελιωδών μοντέλων για αντιληπτικές εργασίες που σχετίζονται με την χειριστική ικανότητα ρομπότ. Το FoundationPose προσφέρει ένα ομοιογενές μοντέλο όρασης για γενικεύσιμη και ανθεκτική εκτίμηση και παρακολούθηση 6D θέσης νέων αντικειμένων, χωρίς την απαίτηση CAD μοντέλων. Το HaMeR χρησιμοποιεί εκτενή σύνολα δεδομένων και ισχυρές αρχιτεκτονικές transformer για την ακριβή ανάκτηση hand meshes από μονοοπτική εικόνα, διευκολύνοντας εργασίες δεξιοτεχνικής αλληλεπίδρασης.

Συνολικά, οι εξελίξεις στα vision foundation models έχουν δημιουργήσει ισχυρές γενικού σκοπού λύσεις για την αναπαράσταση εικόνας, την ευθυγράμμιση όρασης-γλώσσας, τα οπτικά καθήκοντα και τα γενετικά μοντέλα. Οι ικανότητές τους στη γενικευμένη κατανόηση και παραγωγή οπτικού περιεχομένου έχουν επιταχύνει σημαντικά την πρόοδο της πολυτροπικής μάθησης και έχουν ενισχύσει ένα ευρύ φάσμα πραγματικών εφαρμογών.

4.1.2 Vision language model

Η πρόοδος στα θεμελιώδη μοντέλα για την όραση και τη γλώσσα οδήγησε φυσικά την έρευνα προς μια βαθύτερη πολυτροπική κατανόηση, συλλογιστική και παραγωγή, με αποτέλεσμα την εμφάνιση των μοντέλων όρασης-γλώσσας. Ένα από τα αρχικά βήματα ήταν το BLIP, το οποίο εισάγει μια πολυδιάστατη αρχιτεκτονική multimodal mixture-of-encoder-decoder βασισμένη στα ViT και BERT. Το BLIP επιδιώκει μια ενοποιημένη κατανόηση και παραγωγή εικόνας–κειμένου, χρησιμοποιώντας μια διαδικασία data bootstrapping που δημιουργεί λεζάντες και φιλτράρει θορυβώδη διαδικτυακά δεδομένα, προκειμένου να παραχθούν ποιοτικά ζεύγη εικόνας κειμένου. Για την καλύτερη αξιοποίηση των ήδη υπαρχόντων unimodal foundation models, το BLIP-2 προτείνει έναν Q-Former connector και μια διαδικασία εκπαίδευσης δύο σταδίων, προκειμένου να ευθυγραμμίσει αποτελεσματικά παγωμένους pretrained image encoders με παγωμένα LLMs. Με αυτόν τον τρόπο επιτυγχάνει υψηλή απόδοση σε εργασίες vision-language, απαιτώντας ταυτόχρονα ελάχιστες παραμέτρους εκπαίδευσης [35].

Διερευνήθηκαν επίσης ποικίλα αρχιτεκτονικά παραδείγματα για την ανάπτυξη μοντέλων οράσεως-γλώσσας. Το Flamingo αξιοποιεί έναν Perceiver Resampler και gated cross-attention layers για την αποτελεσματική ευθυγράμμιση εικόνων και κειμένου. Επιπλέον, επεξεργάζεται τα δεδομένα με τρόπο εγγενώς συμβατό με εναλλασσόμενες ακολουθίες οπτικών και γλωσσικών εισροών, επιτρέποντάς του να επιτυγχάνει ισχυρές δυνατότητες few-shot μάθησης.

Το LLaVA αποτέλεσε σημαντικό ορόσημο στην αρχιτεκτονική των VLMs, καθώς συνδέει έναν CLIP vision encoder με το LLM Vicuna μέσω μιας απλής γραμμικής προβολής και εκπαιδεύεται πάνω σε σύνολα οπτικών οδηγιών που έχουν παραχθεί από το GPT-4.

Το LLaVA-1.5 βελτιώνει περαιτέρω το αρχικό μοντέλο, υιοθετώντας ισχυρότερο vision encoder, αντικαθιστώντας τη γραμμική προβολή με MLP και αξιοποιώντας μεγαλύτερο σύνολο δεδομένων για την εκπαίδευση.

Μία ακόμη σημαντική ερευνητική κατεύθυνση είναι η οικογένεια Qwen-VL. Το αρχικό Qwen-VL συνδυάζει το Qwen-7B με έναν ViT μέσω ενός position-aware cross-attention adaptor. Μέσω της σχεδίασης ενός ειδικού συστήματος εισόδου–εξόδου για εικόνες και bounding boxes, καθώς και μιας τριφασικής διαδικασίας εκπαίδευσης, το μοντέλο αποκτά ικανότητες κατανόησης εναλλασσόμενων εικόνων και κειμένου, καθώς και ισχυρές δυνατότητες οπτικής θεμελίωσης.

Ο διάδοχός του, Qwen2-VL, αναβαθμίζει την χωροχρονική κωδικοποίηση μέσω της χρήσης 2D RoPE και M-RoPE, προκειμένου να υποστηρίζει εικόνες και βίντεο με ποικιλία αναλύσεων και αναλογιών. Επιπλέον, διαθέτει ισχυρές πολυγλωσσικές ικανότητες και ανταγωνιστικές επιδόσεις σε εργασίες όπως η περιγραφή εικόνας, η απάντηση σε ερωτήσεις σχετικά με εικόνες και η κατανόηση βίντεο.

Πιο πρόσφατα, το Qwen2.5-VL επεκτείνει τη δυναμική ανάλυση και στο χρονικό πεδίο, ευθυγραμμίζοντας τα M-RoPE time IDs με απόλυτο χρόνο, επιτρέποντας έτσι πιο ακριβή χρονική κατανόηση. Επιπλέον, ενσωματώνει window attention στον vision encoder για αποδοτικότερο inference.

Σε συνδυασμό με εκτενή επιμέλεια υψηλής ποιότητας δεδομένων, το Qwen2.5-VL παρέχει ανώτερη οπτική αναγνώριση, ακριβή προσδιορισμό αντικειμένων, αξιόπιστη ανάλυση εγγράφων και εις βάθος κατανόηση μακρών βίντεο.

Οι Karamcheti και οι συνεργάτες του, εξετάζουν κρίσιμες σχεδιαστικές επιλογές για τα VLMs, όπως η προεπεξεργασία εικόνων, η αρχιτεκτονική και οι μέθοδοι βελτιστοποίησης. Καταλήγουν ότι η εκπαίδευση σε ένα στάδιο, οι ενοποιημένες backbone οπτικών από DINOv2 και SigLIP, η χρήση βασικών LLMs και η συνεκπαίδευση με αποκλειστικά γλωσσικά

δεδομένα συνιστούν αποτελεσματικές στρατηγικές. Σύμφωνα με αυτά τα ευρήματα, αναπτύσσονται τα Prismatic VLMs, τα οποία υπερβαίνουν συστηματικά το LLaVA-1.5 σε δοκιμές και χρησιμοποιήθηκαν αργότερα στο OpenVLA.

Το PaliGemma, ένα 3B VLM που βασίζεται στα SigLIP So400m και Gemma 2B, αναπτύχθηκε με έμφαση στη μεταφερσιμότητα και ακολούθως υιοθετήθηκε ως backbone για τη σειρά VLA μοντέλων π0.

Στην αιχμή των τρεχουσών δυνατοτήτων βρίσκονται δύο ιδιόκτητα μοντέλα: το GPT-4o και το Gemimi 2.5 Pro. Και τα δύο επιδεικνύουν εξαιρετικές επιδόσεις σε γενικά benchmark vision-language και έχουν εκτενή εφαρμογή σε πραγματικά συστήματα. Το GPT-4 διακρίνεται για την ενσωματωμένη υποστήριξη παραγωγής εικόνας, ενώ το Gemini 2.5 Pro επισημαίνει τις εξαιρετικά ισχυρές ικανότητες συλλογισμού, επιβεβαιώνοντας την ταχεία πρόοδο και πρακτική χρησιμότητα των σύγχρονων VLMs.

4.1.3 Ενσωματωμένα μοντέλα VLA

Η ταχεία πρόοδος στα θεμελιώδη μοντέλα ενισχύει τη συζήτηση σχετικά με την τεχνητή γενική νοημοσύνη (AGI) και προωθεί την έρευνα προς πιο γενικές, ευέλικτες μορφές νοημοσύνης. Ενώ τα σημερινά μοντέλα θεμελίωσης λειτουργούν κυρίως στο ψηφιακό περιβάλλον, εκπροσωπώντας την ψηφιακή τεχνητή νοημοσύνη, η ερευνητική κοινότητα μετατοπίζει το ενδιαφέρον της προς την ενσωματωμένη τεχνητή νοημοσύνη, δηλαδή την ανάπτυξη γενικευμένων πρακτόρων που μπορούν να εκτελούν ανθρώπινες εντολές και να δρουν στον φυσικό κόσμο [35].

Εντούτοις, τονίζεται ότι η embodied AI συνιστά έναν πολύ πιο φιλόδοξο στόχο σε σύγκριση με τα ψηφιακά μοντέλα. Αυτό οφείλεται σε πολλαπλούς παράγοντες που καθιστούν τη μετάβαση από τον ψηφιακό στον φυσικό χώρο σαφώς πιο δύσκολη και απαιτητική.

Σε θεμελιώδες επίπεδο, τα ζητήματα που καλείται να αντιμετωπίσει η embodied AI ενσωματώνουν νέες διαστάσεις ανοιχτότητας και πολυπλοκότητας που απουσιάζουν από την digital AI. Ενώ τα απαιτητικά ψηφιακά σενάρια περιορίζονται συνήθως σε out-of-distribution ή adversarial δεδομένα, ο φυσικός κόσμος είναι εγγενώς ανοργάνωτος και απρόβλεπτος, και ακόμη και οι πιο συνηθισμένες καταστάσεις μπορεί να συνιστούν σοβαρή πρόκληση. Παραδείγματα περιλαμβάνουν φυσικές αλληλεπιδράσεις με ανθρώπους, τυχαίες παρεμβάσεις, ανατραπείσες καρέκλες, ακατάστατα δωμάτια, εμπόδια και οπτικές αποκρύψεις, και αυτά είναι μόνο τα απλά περιστατικά.

Ένα συγκρίσιμο και πιο οικείο ζήτημα, το οποίο θεωρούμε επίσης τμήμα της embodied AI, είναι η αυτόνομη οδήγηση. Αν και η αυτόνομη οδήγηση είναι ήδη εξαιρετικά απαιτητική, η γενικού σκοπού embodied νοημοσύνη πρέπει να διαχειρίζεται σημαντικά περισσότερες καταστάσεις, με συνέπεια να αντιμετωπίζει πολύ μεγαλύτερες δυσκολίες και απαιτήσεις. Αυτό υποδηλώνει ότι τα μοντέλα και τα δεδομένα που απαιτούνται για αξιόπιστη embodied AI πρέπει να είναι αναλόγως πολύ πιο εξελιγμένα και εκτενή.

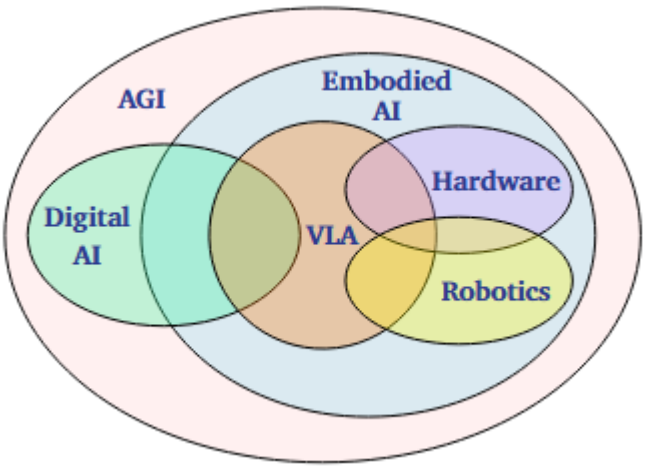
Επιπλέον, μια κρίσιμη παρατήρηση είναι ότι η embodied AI συνεπάγεται απαιτήσεις στο επίπεδο του ρομποτικού υλικού, κάτι που δεν ισχύει για τη digital AI. Για την επίτευξη γενικής σκοπού ενσώματης νοημοσύνης, η ρομποτική πλατφόρμα οφείλει να διαθέτει την επιδεξιότητα και την ανθεκτικότητα που απαιτούνται για καθημερινές εργασίες ικανότητες που παραμένουν σε σημαντική απόσταση από το ανθρώπινο επίπεδο.

Υπάρχουν σημαντικά κενά, όπως η απόσταση των ρομποτικών χεριών και βραχιόνων από την ανθρώπινη δεξιότητα, η υπερβολική εξάρτηση από απλούς αρπάγες, η μεγάλη ποικιλία και ασυμβατότητα μεταξύ διαφορετικών ρομποτικών σωμάτων, καθώς και η έλλειψη πλήρους και ευαίσθητης απτικής αισθητηριοποίησης.

Δεδομένου ότι είναι αδύνατο να επιτευχθεί «τέλειο» hardware σε σύντομο χρονικό διάστημα, μια ρεαλιστική προσδοκία είναι ότι τα μοντέλα, τα δεδομένα και οι ρομποτικές πλατφόρμες θα εξελίσσονται παράλληλα και αμοιβαία, οδηγώντας σταδιακά προς τη γενική ενσώματη νοημοσύνη. Το παρόν κείμενο εστιάζει κυρίως στις πτυχές που αφορούν τα μοντέλα και τα δεδομένα, ενώ επισημαίνει και τις προκλήσεις του hardware, οι οποίες συχνά αποτελούν καθοριστικό παράγοντα στην ανάπτυξη τέτοιων συστημάτων.

Δεδομένου ότι η embodied AI απαιτεί γενικές ικανότητες όρασης και γλώσσας, μια φυσική στρατηγική είναι να στηριχθεί κανείς στα foundation models και να τα επεκτείνει με δυνατότητες δράσης. Αυτή η κατεύθυνση οδήγησε στην ανάπτυξη των ενσωματωμένων μοντέλων VLA, τα οποία αποτελούν πλέον κεντρικό αντικείμενο έρευνας. Το VLA εντοπίζεται στο σημείο σύγκλισης της ψηφιακής τεχνητής νοημοσύνης, της ρομποτικής και του υλικού, συνιστώντας θεμελιώδες υποπεδίο της embodied AI και κρίσιμο βήμα προς την AGI Εικόνα 4.3. Ο υψηλός αριθμός δημοσιευμένων εργασιών VLA υποδεικνύει έναν ταχέως εξελισσόμενο τομέα Εικόνα 4.2. Ο οποίος εμφανίζει πρώιμες αλλά περιορισμένες ενδείξεις νοημοσύνης και γενίκευσης. Παρά την πρόοδο, οι περισσότερες αξιολογήσεις παραμένουν περιορισμένες σε απλοποιημένες εργαστηριακές συνθήκες, κυρίως σε χειρισμούς με grippers, και επομένως απέχουν σημαντικά από τις απαιτήσεις για γενικούς ενσώματους πράκτορες ικανούς να λειτουργούν σε καθημερινά περιβάλλοντα. Ως εκ τούτου, το πεδίο παραμένει σε

πρωτογενές στάδιο και απαιτούνται ουσιαστικά βήματα προόδου. Η εξέλιξη των embodied VLA μοντέλων αναμένεται να παραμείνει το κύριο μέτωπο έρευνας για τα επόμενα χρόνια.



Εικόνα 4.3 Ένα διάγραμμα Venn απεικονίζει τις σχέσεις μεταξύ των βασικών πεδίων της τεχνητής νοημοσύνης. Τα VLA μοντέλα βρίσκονται στο σημείο τομής της digital AI, του hardware και της ρομποτικής, αποτελώντας βασικό υποπεδίο της embodied AI και κρίσιμο τομέα στην πορεία προς την AGI [35].

4.2 Vision language action

Τα μοντέλα Vision Language Action επεκτείνουν τα παραδοσιακά συστήματα όρασης–γλώσσας, ενσωματώνοντας έναν επιπλέον χώρο ενεργειών, τον οποίο το μοντέλο οφείλει πλέον να προβλέπει παράλληλα με την επεξεργασία των οπτικών δεδομένων και των γλωσσικών οδηγιών. Σε αντίθεση με αρχιτεκτονικές που περιορίζονται στην παραγωγή κειμένου, τα VLA θεωρούν τις ενέργειες του ρομπότ ως στοιχεία του ίδιου χώρου εξόδου με τα γλωσσικά tokens. Κατ' αυτόν τον τρόπο, οι ενέργειες κωδικοποιούνται, μετατρέπονται σε tokens και προβλέπονται διαδοχικά εντός ενός ενιαίου πλαισίου μοντελοποίησης ακολουθιών. Τυπικά, ένα μοντέλο VLA υλοποιεί μια χαρτογράφηση από τις αντιληπτικές εισροές και τις προαιρετικές γλωσσικές εντολές προς μια ακολουθία ενεργειών:

$$(I, L) \rightarrow A \tag{1.1}$$

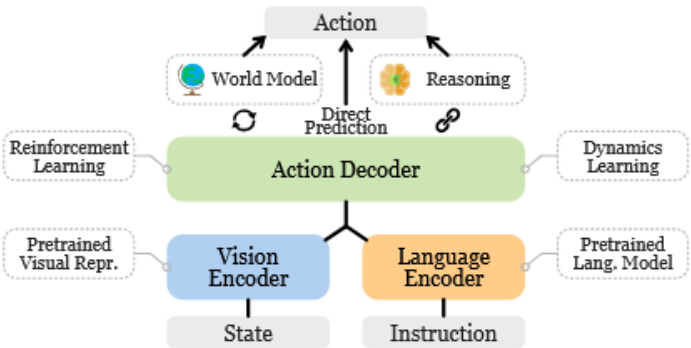
Όπου I αντιπροσωπεύει τις οπτικές παρατηρήσεις, L τη γλωσσική οδηγία ή το πλαίσιο της εντολής, και A την αντίστοιχη αναπαράσταση της ενέργειας. Με αυτόν τον τρόπο, το μοντέλο μπορεί να ενοποιεί αντίληψη, κατανόηση και έλεγχο, παράγοντας action tokens ως άμεση αντίδραση σε πολυτροπικά δεδομένα.

Από την προοπτική της embodied AI, τα VLA μοντέλα συνιστούν μια κατηγορία πολιτικών όπου η επεξεργασία αισθητηριακών πληροφοριών, η γλωσσική λογική και ο χαμηλού επιπέδου έλεγχος υλοποιούνται από μία ενιαία αρχιτεκτονική τύπου transformer. Ενσωματώνοντας οπτικά χαρακτηριστικά, γλωσσικά tokens και action tokens σε έναν κοινό χώρο ακολουθίας, αυτά τα μοντέλα λειτουργούν ως agents που εκτελούν οδηγίες, κατανοούν σκηνές, «γεφυρώνουν» τις εντολές φυσικής γλώσσας με το περιβάλλον και παράγουν αντίστοιχες ενέργειες. Αυτή η ομοιογενής δομή τους διαφοροποιεί από προηγούμενες προσεγγίσεις που στηρίζονταν σε διακριτές μονάδες για την αντίληψη, την ερμηνεία εντολών και τον έλεγχο [36].

4.2.1 Γενική αρχιτεκτονική VLA

Ένα τυπικό μοντέλο VLA στηρίζεται σε τρία κύρια δομικά στοιχεία: έναν οπτικό κωδικοποιητή, έναν γλωσσικό ή multimodal σκελετό, και έναν αποκωδικοποιητή δράσης [37]. Ο οπτικός κωδικοποιητής μετατρέπει τις ακατέργαστες εικόνες της κάμερας ή άλλες αισθητηριακές παρατηρήσεις σε συμπαγείς αναπαραστάσεις χαρακτηριστικών. Το backbone στη συνέχεια συνδυάζει αυτές τις οπτικές αναπαραστάσεις με τις προαιρετικές γλωσσικές εισροές, επιτρέποντας στο σύστημα να κατανοεί τόσο το περιεχόμενο της σκηνής όσο και το νόημα των οδηγιών. Στο τελευταίο στάδιο, ο action decoder, ο οποίος λειτουργεί στο ίδιο token/output space με το backbone, παράγει μια ακολουθία action tokens που αντιστοιχούν σε εντολές προς τον ρομποτικό ενεργοποιητή, όπως κινήσεις, σήματα ελέγχου ή τμήματα τροχιάς [38].

Η ομοιογενής αυτή αρχιτεκτονική επιτρέπει στο ίδιο μοντέλο να εκτελεί ταυτόχρονα ρόλους αντίληψης, κατανόησης και ελέγχου. Οι πολυτροπικές είσοδοι (όραση και γλώσσα) επεξεργάζονται συλλογικά για την παραγωγή ενεργειών, χωρίς την ανάγκη ξεχωριστών υποσυστημάτων για την επεξεργασία εικόνας, την ερμηνεία φυσικής γλώσσας, τον σχεδιασμό ή τον χαμηλού επιπέδου έλεγχο [39].



Εικόνα 4.4 Η γενική αρχιτεκτονική των μοντέλων VLA. Παρουσιάζονται τρεις αντιπροσωπευτικές μέθοδοι για την πρόβλεψη δράσης. Τα σχετικά στοιχεία παρουσιάζονται σε διακεκομμένα πλαίσια [37].

4.2.2 Action tokenization και output spaces

Κεντρικό στοιχείο στον σχεδιασμό των VLA μοντέλων είναι η διαδικασία μετατροπής του action space σε tokens και η αναπαράστασή του στην έξοδο. Σύμφωνα με πρόσφατη βιβλιογραφία, οι κύριες προσεγγίσεις για την αποτύπωση των ενεργειών περιλαμβάνουν: περιγραφές σε φυσική γλώσσα, αποσπάσματα κώδικα, διακριτά low-level control tokens, tokens που περιγράφουν τροχιές, tokens που δηλώνουν επιθυμητές τελικές καταστάσεις, λανθάνουσες αναπαραστάσεις και συνεχείς εξόδους [40].

Διακριτά action tokens: Σε αυτή την κατηγορία, οι ενέργειες ποσοτικοποιούνται σε διακριτά tokens, παρόμοια με τα γλωσσικά tokens, και στη συνέχεια μετατρέπονται εκ νέου σε εντολές ελέγχου. Η εν λόγω μέθοδος υπήρξε η θεμελιώδης βάση των πρώτων συστημάτων VLA [39].

Διαδρομή και υψηλού επιπέδου tokens: Αντί για εντολές χαμηλού επιπέδου, τα tokens μπορούν να περιγράφουν πιο αφηρημένες ενέργειες ή ολόκληρες τροχιές, όπως μετάβαση σε συγκεκριμένο σημείο, αιώρηση ή μεταβολή ύψους. Αυτή η προσέγγιση παρέχει αυξημένη αφαίρεση και διευκολύνει τον σχεδιασμό της κίνησης σε σύνθετα περιβάλλοντα [37].

Λανθάνουσες ή συνεχείς έξοδοι: Ορισμένες εκδόσεις VLA χρησιμοποιούν συνεχείς ή λανθάνουσες αναπαραστάσεις για τη διατήρηση υψηλής ακρίβειας και ομαλότητας στον έλεγχο, γεγονός που είναι κρίσιμο για λεπτούς χειρισμούς ή ενέργειες υψηλής συχνότητας.

Η επιλογή της μεθόδου tokenization έχει σημαντική επίδραση στη δυνατότητα γενίκευσης, την ακρίβεια του ελέγχου και το υπολογιστικό κόστος. Καθορίζει ουσιαστικά αν το μοντέλο λειτουργεί κυρίως ως υψηλού επιπέδου σχεδιαστής ή ως χαμηλού επιπέδου ελεγκτής.

4.3 Ταξινόμηση και επισκόπηση μοντέλων VLA

4.3.1 Ταξινόμηση με βάση την αναπαράσταση δράσης

Ένας κύριος άξονας ταξινόμησης των VLA μοντέλων σχετίζεται με τον τρόπο αναπαράστασης και μετατροπής των ενεργειών σε tokens. Ο Zhong [35] και οι συνεργάτες του προτείνουν μια συστηματική τυπολογία, στην οποία οι υφιστάμενες προσεγγίσεις κατατάσσονται σε πέντε κύριες κατηγορίες βάσει της στρατηγικής tokenization που υιοθετούν.

Η πρώτη κατηγορία περιλαμβάνει γλωσσικά βασισμένα tokens δράσης, όπου οι εντολές ελέγχου εκφράζονται άμεσα ως μονάδες φυσικής γλώσσας. Αυτή η επιλογή εκμεταλλεύεται τα γλωσσικά priors ενός προεκπαιδευμένου μοντέλου, αλλά συχνά περιορίζει το επίπεδο λεπτομέρειας που μπορεί να επιτευχθεί στον έλεγχο. Στη δεύτερη κατηγορία, τα code-based action tokens αναπαριστούν τις ενέργειες ως σύντομες προγραμματιστικές οδηγίες ή δομημένα αποσπάσματα, προσφέροντας μεγαλύτερη ακρίβεια και συνθετικότητα , ιδιαιτέρως σε εργασίες όπου η ρητή διαδικαστική δομή είναι κρίσιμη.

Η τρίτη κατηγορία σχετίζεται με τα trajectory tokens, όπου οι ενέργειες καταγράφονται ως τμήματα κίνησης ή ως διακριτά σημεία μιας τροχιάς. Αυτή η μορφή αποτυπώνει με μεγαλύτερη ακρίβεια τη χρονική και χωρική συνέχεια, καθιστώντας την κατάλληλη για πλοήγηση ή για εργασίες χειρισμού που απαιτούν ακολουθίες πολλαπλών κινήσεων. Η τέταρτη ομάδα, τα goal-state tokens, αναπαριστά την ενέργεια ως επιθυμητή τελική κατάσταση, μεταφέροντας τον λεπτομερή έλεγχο χαμηλού επιπέδου σε έναν ξεχωριστό ελεγκτή.

Τέλος, η κατηγορία των latent ή continuous tokens δράσης αποδίδει τις ενέργειες ως λανθάνοντες διανυσματικούς χώρους ή ως συνεχείς εξόδους, επιτρέποντας ομαλό και υψηλής ακρίβειας έλεγχο, ενώ παραμένει συμβατή με την αυτοπαραγωγική λογική των αρχιτεκτονικών transformer. Η επιλογή της μορφής αναπαράστασης των ενεργειών καθορίζει την ικανότητα ελέγχου, τη γενίκευση και την υπολογιστική αποδοτικότητα ενός μοντέλου VLA, αποτελώντας έναν από τους κύριους άξονες σχεδιασμού του [35].

4.3.2 Ενσωματωμένοι τομείς ρομποτικής

Τα μοντέλα Vision Language Action περιλαμβάνουν ποικιλία εφαρμογών στην embodied AI, με κοινό γνώρισμα την απαίτηση συνδυασμού αντίληψης, κατανόησης οδηγιών και εκτέλεσης δράσης εντός μιας ενιαίας πολιτικής. Ο Ma [37] και οι συνεργάτες του κατηγοριοποιούν τις εφαρμογές των VLA σε διάφορες κύριες κατηγορίες ρομποτικών εργασιών. Μία από αυτές σχετίζεται με τα manipulation tasks, όπου το μοντέλο οφείλει να ερμηνεύσει τη σκηνή και τις γλωσσικές οδηγίες προκειμένου να εκτελέσει αλληλεπιδράσεις με αντικείμενα, όπως λαβές, αναδιάταξη ή χρήση εργαλείων.

Μια δεύτερη σημαντική κατηγορία περιλαμβάνει εργασίες πλοήγησης και εξερεύνησης, στις οποίες τα κινητά ρομπότ οφείλουν να αποκωδικοποιούν υψηλού επιπέδου εντολές, να εντοπίζουν οπτικά στοιχεία του περιβάλλοντος και να εκτελούν κατάλληλες κινήσεις, διατηρώντας ταυτόχρονα επίγνωση της γύρω κατάστασης.

Επιπλέον, οι τομείς εφαρμογής σχετίζονται με ρομποτική για οικιακές και υπηρεσιακές εργασίες, όπου τα μοντέλα VLA μπορούν να υποστηρίξουν καθήκοντα όπως ο καθαρισμός, η βοήθεια στο μαγείρεμα ή η ανάκτηση αντικειμένων σε ημιδομημένα εσωτερικά περιβάλλοντα. Ένας επιπλέον τομέας εφαρμογής είναι η αυτόνομη οδήγηση, όπου η πολυτροπική κατανόηση ενισχύει την αντίληψη, την αναγνώριση σκηνών και τη λήψη αποφάσεων βάσει γλωσσικών οδηγιών.

Τέλος, ο Ma αναφέρει το πεδίο των ενσωματωμένων πρακτόρων σε προσομοιώσεις, όπου τα VLA χρησιμοποιούνται για την εκτέλεση οδηγιών, την ολοκλήρωση σύνθετων εργασιών και την αλληλεπίδραση με εικονικά περιβάλλοντα. Συνολικά, τα προαναφερθέντα παραδείγματα αναδεικνύουν το εύρος των ενσωματωμένων εργασιών στις οποίες η ολιστική προσέγγιση vision–language–action υπερέχει σε σχέση με παραδοσιακά συστήματα που διαθέτουν διακριτές ενότητες αντίληψης, λογικής και ελέγχου [37].

4.3.3 Τυπικά καθεστώτα εκπαίδευσης και δεδομένων

Η εκπαίδευση των VLA μοντέλων συνήθως στηρίζεται σε έναν συνδυασμό εκτενούς πολυτροπικού προεκπαίδευσης και εξειδικευμένων δεδομένων από ρομποτικές εργασίες. Σε συστήματα τύπου transformer, μια συνήθης πρακτική είναι η αρχικοποίηση μέσω vision language pretraining σε δεδομένα διαδικτύου, όπου το backbone αλληλοεπιδρά με ποικίλα ζεύγη εικόνας και κειμένου. Αυτή η εκτενής πολυτροπική θεμελίωση παρέχει στο μοντέλο τους αναγκαίους σημασιολογικούς και αντιληπτικούς προσανατολισμούς για μελλοντικές ενσωματωμένες εφαρμογές. Στηριζόμενο σε αυτό το υπόβαθρο, ακολουθεί εκπαίδευση ή fine-tunning σε ρομποτικά σύνολα δεδομένων τροχιών, στα οποία οι ακολουθίες παρατηρήσεων, οδηγιών και ενεργειών κωδικοποιούνται ως πολυτροπικές “προτάσεις”, επιτρέποντας στο μοντέλο να κατανοήσει τη σχέση μεταξύ αντίληψης, γλώσσας και ελέγχου.

Το παράδειγμα του RT-2 αναδεικνύει χαρακτηριστικά αυτή τη σύνθετη προσέγγιση: ένα backbone vision language, προεκπαιδευμένο σε εκτενή δεδομένα του διαδικτύου, εκπαιδεύεται στη συνέχεια να προβλέπει ενέργειες κωδικοποιημένες ως κείμενο βάσει πραγματικών ρομποτικών επιδείξεων. Με αυτόν τον τρόπο, το ενιαίο μοντέλο αξιοποιεί τη γενική οπτικό-σημασιολογική γνώση και τις εξειδικευμένες τροχιές ελέγχου από πραγματικά καθήκοντα. Παράλληλα, ο Ma [39] και οι συνεργάτες του περιγράφουν διάφορα συνηθισμένα training paradigms για VLA πολιτικές, όπως behavior cloning, imitation learning και reinforcement learning fine-tunning. Οι μέθοδοι αυτές συχνά αξιοποιούν μεικτά δεδομένα από προσομοιώσεις και πραγματικούς χειρισμούς, προκειμένου να διασφαλιστεί η απαραίτητη

κλίμακα και ποικιλία, περιορίζοντας ταυτόχρονα το κόστος και τον κίνδυνο συλλογής πραγματικών ρομποτικών δεδομένων.

Συνολικά, τα καθεστώτα εκπαίδευσης των μοντέλων VLA ακολουθούν πολυσταδιακές διαδικασίες που συνδυάζουν εκτενή πολυτροπική κατανόηση με θεμελιωμένη ενσωματωμένη εμπειρία [39].

4.4 Visual language action models για μη επανδρωμένα αεροσκάφη

Αν και τα συστήματα Vision–Language–Action έχουν μελετηθεί εκτενώς σε πολλές ρομποτικές εφαρμογές, η προσαρμογή τους σε εναέρια οχήματα συνεπάγεται ειδικές απαιτήσεις και νέες δυνατότητες. Τα UAVs επιχειρούν σε τρισδιάστατα, δυναμικά μεταβαλλόμενα περιβάλλοντα, όπου απαιτούνται ταχεία αντίληψη, ακριβής έλεγχος και ισχυρή λογική υπό αυστηρούς περιορισμούς ασφαλείας. Δεδομένου αυτού, πρόσφατες μελέτες έχουν αρχίσει να διευρύνουν το VLA πλαίσιο, το οποίο αρχικά αναπτύχθηκε για εργασίες χειρισμού και επίγειας ρομποτικής, σε τομείς όπως ο racing drones, η γνωσιακή κατανόηση και ο σχεδιασμός αποστολών σε εναέρια συστήματα.

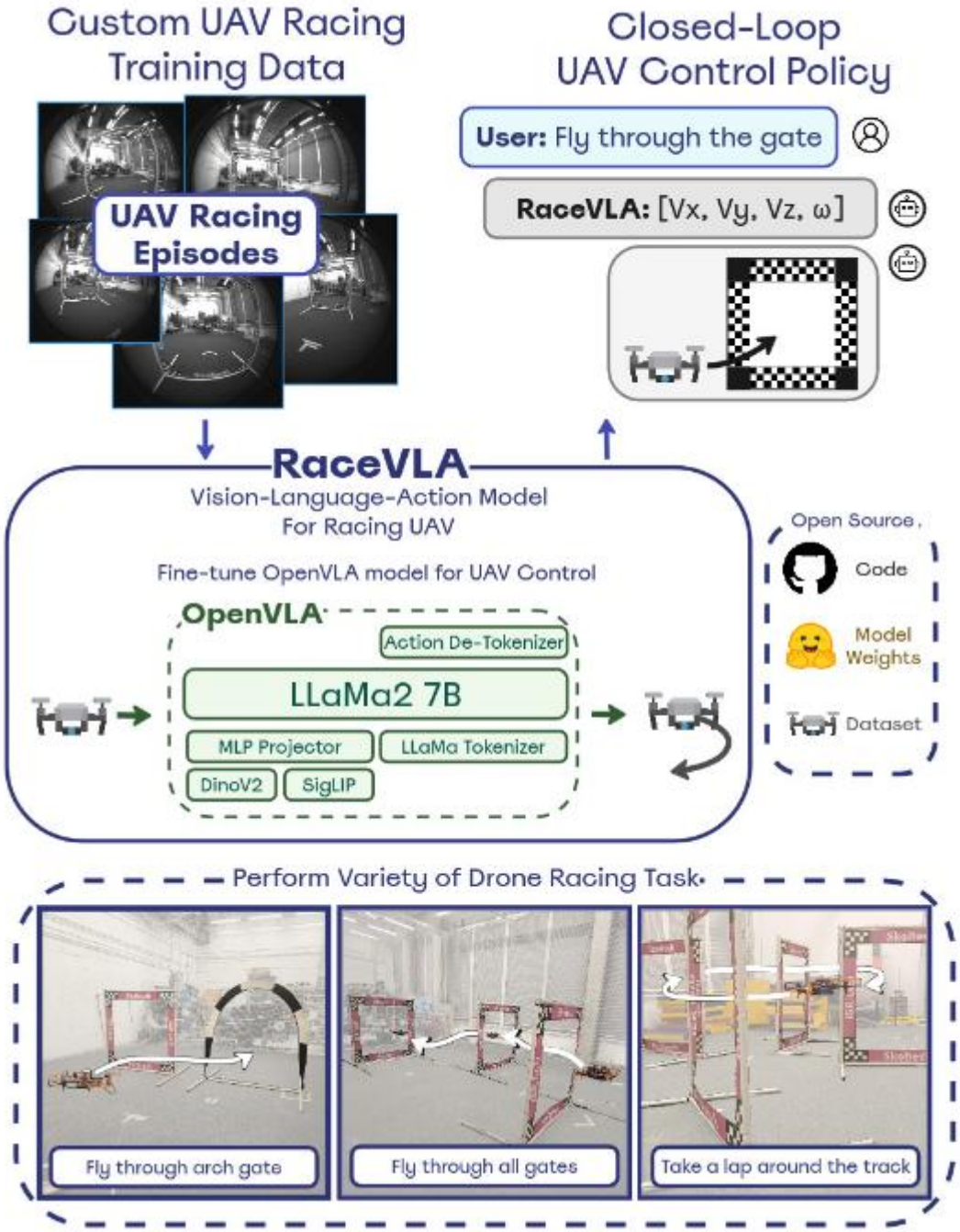
Τρεις χαρακτηριστικές προσεγγίσεις που αποτυπώνουν αυτή τη νέα κατεύθυνση είναι τα RaceVLA, CognitiveDrone και UAV-VLA, τα οποία αντιπροσωπεύουν διαφορετικές πτυχές της μετάβασης των μοντέλων VLA στον τομέα των UAVs.

4.4.1 RaceVLA

Το RaceVLA είναι μία από τις πρώτες αρχιτεκτονικές VLA που αναπτύχθηκαν ειδικά για τον έλεγχο UAV υψηλής ταχύτητας. Η εργασία εστιάζει στον αγώνα FPV drone, όπου το σύστημα οφείλει να ακολουθεί απαιτητικές διαδρομές, αξιοποιώντας ταυτόχρονα φυσικές γλωσσικές οδηγίες και οπτικές πληροφορίες από την κάμερα του drone. Το μοντέλο ενοποιεί τις οπτικές παρατηρήσεις, τα κειμενικά prompts και την πρόβλεψη ενεργειών σε ένα συνεκτικό πλαίσιο βασισμένο σε ακολουθίες. Οι έξοδοι του συνιστούν ένα συνεχές τετραδιάστατο διάνυσμα ελέγχου, συνήθως αναπαριστώντας γραμμικές ταχύτητες και ρυθμούς περιστροφής ,επιτρέποντας στο μοντέλο να λειτουργεί άμεσα ως controller χαμηλού επιπέδου [41].

Οι συγγραφείς αποδεικνύουν ότι το RaceVLA επιτυγχάνει σταθερή πλοήγηση υψηλής ταχύτητας και επιδεικνύει ικανότητα γενίκευσης σε ποικίλες πίστες, περιβαλλοντικές υφές και διαφορετικές μορφές prompting. Η μελέτη συγκρίνει επίσης άλλες προσεγγίσεις VLA, περιλαμβανομένων παραλλαγών OpenVLA, καταγράφοντας βελτιωμένη απόδοση σε οπτικές,

σημασιολογικές και φυσικές μετρικές γενίκευσης. Η εργασία αναδεικνύει τη χρησιμότητα των VLA μοντέλων σε εφαρμογές UAV που απαιτούν ταχείες αποφάσεις, στενή συσχέτιση αντίληψης–ενέργειας και ικανότητα κατανόησης οδηγιών από τον χρήστη.



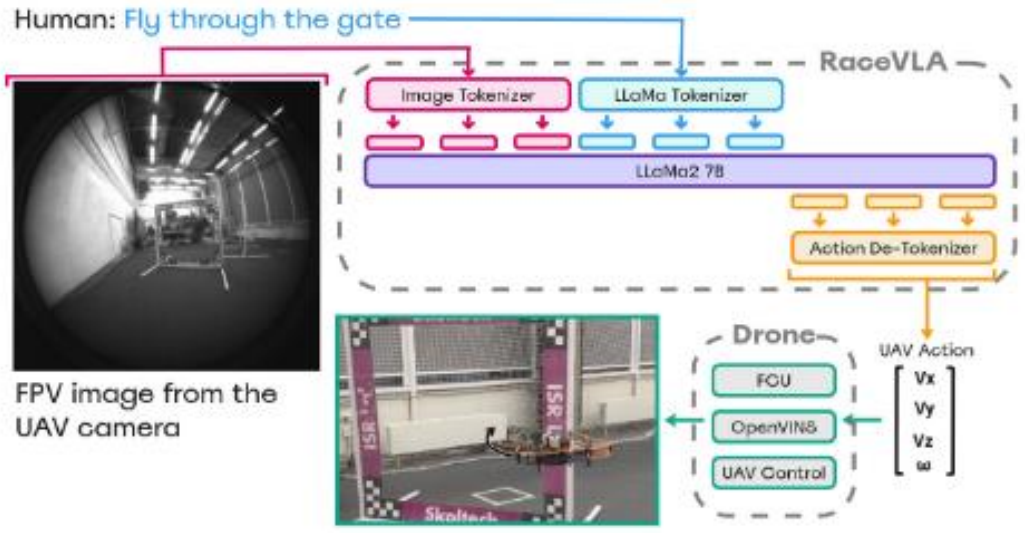
Εικόνα 4.5 Το RaceVLA είναι το πρώτο VLA μοντέλο που έχει αναπτυχθεί αποκλειστικά για αγωνιστικά drones. Επεξεργάζεται ροές βίντεο FPV σε συνδυασμό με εντολές φυσικής γλώσσας και παράγει ως έξοδο διανύσματα ταχυτήτων (Vx, Vy, Vz) και γωνιακή ταχύτητα yaw (ω). Με αυτόν τον σχεδιασμό, το σύστημα μπορεί να εκτελεί αυτοδύναμα ένα ευρύ φάσμα πτητικών εργασιών, περιλαμβάνοντας την πλοήγηση σε άγνωστα περιβάλλοντα και νέα

σενάρια. Αξιοποιώντας ένα ειδικά διαμορφωμένο σύνολο εκπαίδευσης, το RaceVLA παρουσιάζει ισχυρή ικανότητα γενίκευσης [41].

Μία αποτελεσματική μέθοδος ενίσχυσης της παρουσίασης του RaceVLA είναι η λεπτομερής περιγραφή της αρχιτεκτονικής του, όπως απεικονίζεται στο Εικόνα 4.5 , καθώς αναδεικνύει τη συσχέτιση των επιμέρους τμημάτων με το συνολικό VLA πλαίσιο.

Στην εν λόγω αρχιτεκτονική, η ακατέργαστη εικόνα από την εμπρόσθια FPV κάμερα του drone εισέρχεται αρχικά στον οπτικό κωδικοποιητή, ο οποίος εξάγει χωρικά και σημασιολογικά χαρακτηριστικά απαραίτητα για την πλοήγηση. Οι οπτικές αναπαραστάσεις συγχωνεύονται με μια ακολουθία γλωσσικών tokens που προκύπτουν από τη φυσική γλωσσική καθοδήγηση. Και οι δύο μορφές εισόδου ενσωματώνονται στο multimodal transformer, ο οποίος αποτελεί τον κεντρικό πυρήνα της πολιτικής RaceVLA. Ο transformer επεξεργάζεται ταυτόχρονα το οπτικό και γλωσσικό πλαίσιο και προβλέπει αυτοαναδρομικά μια ακολουθία action tokens που αντιστοιχούν σε συνεχείς εντολές χαμηλού επιπέδου.

Η κεφαλή δράσης μετατρέπει αυτές τις προβλέψεις σε ένα τετραδιάστατο διάνυσμα ελέγχου, το οποίο συνήθως περιλαμβάνει κίνηση προς τα εμπρός/πίσω, πλευρική μετατόπιση, κατακόρυφη κίνηση και ρυθμό περιστροφής γύρω από τον κατακόρυφο άξονα. Επιπλέον, η αρχιτεκτονική περιλαμβάνει έναν μηχανισμό χρονικής εξομάλυνσης ή φιλτραρίσματος, διασφαλίζοντας τη σταθερότητα των εντολών ακόμη και σε υψηλές ταχύτητες. Η ενσωμάτωση του σχετικού σχήματος και της περιγραφής του στο κείμενο αναδεικνύει πώς το RaceVLA εφαρμόζει στην πράξη τη γενική VLA αρχιτεκτονική και πώς οι σχεδιαστικές επιλογές του υποστηρίζουν την πλοήγηση UAV σε πραγματικό χρόνο και υψηλές ταχύτητες.

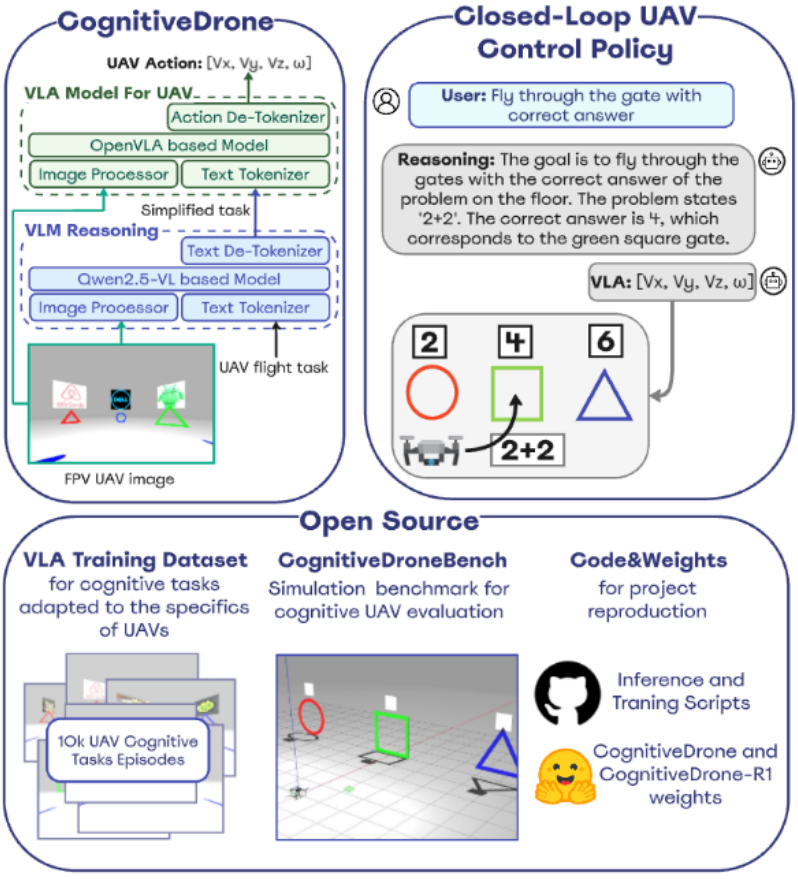


Εικόνα 4.6 Αρχιτεκτονική RaceVLA [41].

4.4.2 Cognitive task solving και reasoning στα UAVs

Ενώ το RaceVLA εστιάζει στον έλεγχο υψηλής ταχύτητας, το CognitiveDrone επιδιώκει γνωστικές εναέριες εργασίες που απαιτούν αναγνώριση, ερμηνεία και λογική επεξεργασία. Αυτές οι εργασίες περιλαμβάνουν τον εντοπισμό ατόμων ή αντικειμένων, την ερμηνεία συμβολικών σημάτων και την επίλυση προβλημάτων που βασίζονται σε οδηγίες και οπτικές πληροφορίες. Για την αξιολόγηση αυτών των ικανοτήτων, οι συγγραφείς εισάγουν το CognitiveDroneBench, ένα benchmark ειδικά σχεδιασμένο για να αξιολογεί την ικανότητα γνωστικής λογικής των UAVs σε ένα ευρύ φάσμα οπτικών και συμβολικών εργασιών.

Η μελέτη αναλύει τρία μοντέλα: το βασικό CognitiveDrone, την ενισχυμένη εκδοχή CognitiveDrone-R1 που περιλαμβάνει πρόσθετο μηχανισμό λογικής, και το RaceVLA. Τα αποτελέσματα υποδεικνύουν ότι το CognitiveDrone-R1 υπερτερεί του CognitiveDrone και του RaceVLA σε σύνθετες εργασίες συλλογισμού, ενώ το RaceVLA διατηρεί την υπεροχή του σε δυναμικά, πτητικά σενάρια. Η σύγκριση αυτή αποκαλύπτει ότι τα VLA συστήματα μπορούν να εστιάσουν σε διάφορα άκρα του φάσματος, από υψηλής ταχύτητας έλεγχο έως γνωστική κατανόηση, ανάλογα με τις απαιτήσεις της εφαρμογής [42].

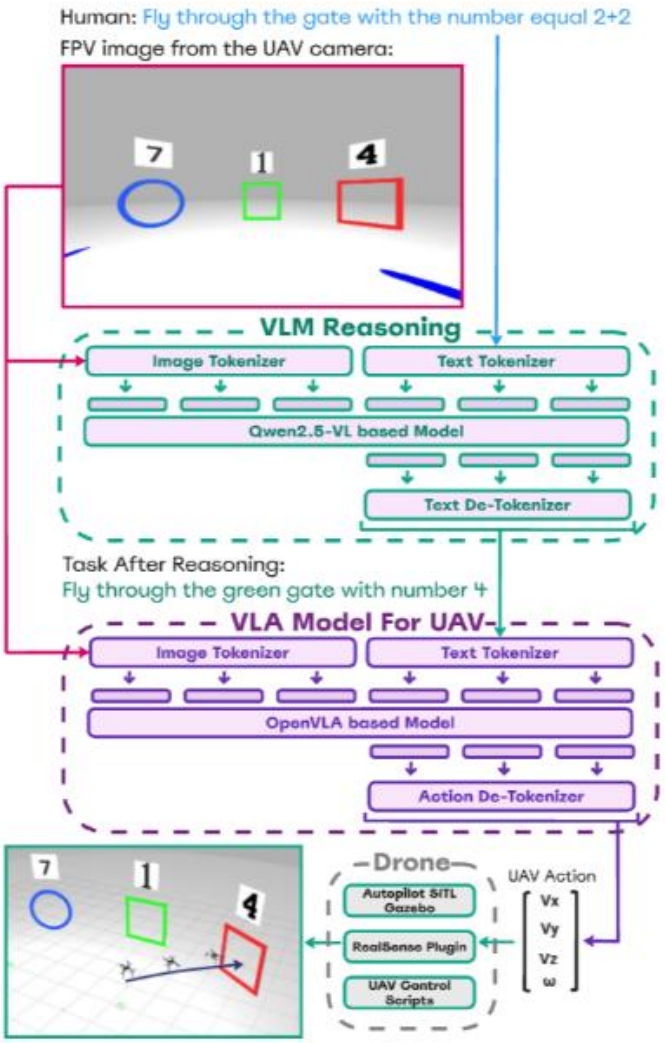


Εικόνα 4.7 Το CognitiveDrone είναι ένα VLA σύστημα για UAVs που δημιουργεί ομαλές τετραδιάστατες εντολές ελέγχου, βασισμένες σε FPV οπτικά δεδομένα και οδηγίες φυσικής γλώσσας. Το μοντέλο συνδυάζει έναν πυρήνα VLA 7B παραμέτρων, εκπαιδευμένο σε ένα εκτενές ανοικτό σύνολο δεδομένων γνωστικών εργασιών — όπως συλλογισμό, αναγνώριση ανθρώπων και κατανόηση συμβόλων με ένα πρόσθετο module λογικής επίσης 7B παραμέτρων, το οποίο βελτιώνει και αποσαφηνίζει τις εντολές της αποστολής. Το σύστημα αξιολογείται στο CognitiveDroneBench, το πρώτο benchmark που έχει σχεδιαστεί για VLA μοντέλα με γνωστικό προσανατολισμό σε UAVs. Το drone καλείται να πλοηγηθεί σε μια διαδρομή με πύλες, επιλέγοντας τη σωστή πύλη μέσω της επίλυσης των αντίστοιχων γνωστικών εργασιών. Οι δημιουργοί παρέχουν πλήρως ανοικτό κώδικα για το dataset, το περιβάλλον αξιολόγησης, τα βάρη των μοντέλων και τον κώδικα εκπαίδευσης και inference [42].

Μία αποτελεσματική μέθοδος ενίσχυσης της παρουσίασης του RaceVLA είναι η λεπτομερής περιγραφή της αρχιτεκτονικής του, όπως απεικονίζεται στην **Εικόνα 4.7** , καθώς αναδεικνύει τη συσχέτιση των επιμέρους τμημάτων με το συνολικό VLA πλαίσιο.

Στην εν λόγω αρχιτεκτονική, η ακατέργαστη εικόνα από την εμπρόσθια FPV κάμερα του drone εισέρχεται αρχικά στον οπτικό κωδικοποιητή, ο οποίος εξάγει χωρικά και σημασιολογικά χαρακτηριστικά απαραίτητα για την πλοήγηση. Οι οπτικές αναπαραστάσεις συγχωνεύονται με μια ακολουθία γλωσσικών tokens που προκύπτουν από τη φυσική γλωσσική καθοδήγηση. Και οι δύο μορφές εισόδου προωθούνται στο multimodal transformer backbone, ο οποίος αποτελεί τον κεντρικό πυρήνα της πολιτικής RaceVLA. Ο transformer επεξεργάζεται ταυτόχρονα το οπτικό και γλωσσικό πλαίσιο και προβλέπει αυτοαναδρομικά μια ακολουθία action tokens που αντιστοιχούν σε συνεχείς εντολές χαμηλού επιπέδου.

Η κεφαλή δράσης μετατρέπει αυτές τις προβλέψεις σε ένα τετραδιάστατο διάνυσμα ελέγχου, το οποίο συνήθως περιλαμβάνει κίνηση προς τα εμπρός/πίσω, πλευρική μετατόπιση, κατακόρυφη κίνηση και ρυθμό περιστροφής γύρω από τον κατακόρυφο άξονα. Επιπλέον, η αρχιτεκτονική περιλαμβάνει έναν μηχανισμό χρονικής εξομάλυνσης ή φιλτραρίσματος, διασφαλίζοντας τη σταθερότητα των εντολών ακόμη και σε υψηλές ταχύτητες. Η ενσωμάτωση του σχετικού σχήματος και της περιγραφής του στο κείμενο αναδεικνύει πώς το RaceVLA εφαρμόζει στην πράξη τη γενική VLA αρχιτεκτονική και πώς οι σχεδιαστικές επιλογές του υποστηρίζουν την πλοήγηση UAV σε πραγματικό χρόνο και υψηλές ταχύτητες.



Εικόνα 4.8 Αρχιτεκτονική CognitiveDrone [42].

Τα παραδείγματα του dataset στην Εικόνα 4.6 αναδεικνύουν τη δομή και την ποικιλία των εργασιών του CognitiveDroneBench. Το σχήμα απεικονίζει αντιπροσωπευτικούς τύπους γνωστικών εργασιών, όπως η αναγνώριση ανθρώπινης παρουσίας, η ταυτοποίηση αντικειμένων στη σκηνή, η ερμηνεία βελών ή συμβόλων, καθώς και η απάντηση σε απλές ερωτήσεις που βασίζονται σε οπτικές ενδείξεις.

Κάθε εργασία συνδυάζει μια εικόνα με μια οδηγία σε φυσική γλώσσα και την αντίστοιχη επιθυμητή ενέργεια, δημιουργώντας ένα ολοκληρωμένο πολυτροπικό δείγμα για εκπαίδευση ή αξιολόγηση. Η ενσωμάτωση αυτού του σχήματος στην εργασία διευκρινίζει τον τρόπο με τον οποίο τα γνωστικά προβλήματα διαμορφώνονται ως VLA tasks, όπου το μοντέλο οφείλει να ενοποιήσει την αντίληψη, τη γλωσσική κατανόηση και τη συμβολική λήψη αποφάσεων. Ταυτόχρονα, προσφέρει το αναγκαίο πλαίσιο για τις συγκρίσεις μεταξύ των CognitiveDrone, CognitiveDrone-R1 και RaceVLA.

Εικόνα 4.9 Παραδείγματα προετοιμασμένων εργασιών στο dataset, σχεδιασμένων ώστε τα VLA μοντέλα να επιλύουν γνωστικά καθήκοντα προσαρμοσμένα για UAVs [42].

4.4.3 UAV-VLA

Το UAV-VLA διευρύνει το VLA πλαίσιο στον τομέα του σχεδιασμού εναέριων αποστολών, όπου ο στόχος δεν είναι ο άμεσος έλεγχος σε χαμηλό επίπεδο, αλλά η δημιουργία ενός οργανωμένου πτητικού σχεδίου βάσει οδηγιών φυσικής γλώσσας και δορυφορικών ή εναέριων εικόνων. Το σύστημα συνίσταται από επιμέρους μονάδες με σαφείς ρόλους:

• Εξαγωγή στόχων μέσω LLM, το οποίο αναλύει τη λεκτική περιγραφή της αποστολής και εντοπίζει τους κύριους στόχους.

• Εξερεύνηση αντικειμένων και περιοχών μέσω VLM, το οποίο επεξεργάζεται δορυφορικές εικόνες για τον εντοπισμό σημείων ενδιαφέροντος για αποστολή.

• Δημιουργία ακολουθίας ενεργειών ή waypoints μέσω LLM, το οποίο συνθέτει ένα ολοκληρωμένο σχέδιο πτήσης ικανό προς υλοποίηση από downstream flight controllers.

Σε αντίθεση με τα RaceVLA και CognitiveDrone, τα οποία λειτουργούν στο επίπεδο του αισθητηριοκινητικού ελέγχου ή της γνωστικής επεξεργασίας, το UAV-VLA εκμεταλλεύεται το VLA πλαίσιο ως μηχανισμό υψηλού επιπέδου σχεδιασμού. Ενοποιεί τη λογική των μεγάλων γλωσσικών μοντέλων με οπτική θεμελίωση, προκειμένου να παραγάγει πτητικές διαδρομές που πληρούν περιορισμούς και στόχους αποστολής. Με αυτόν τον τρόπο, αναδεικνύει μια εναλλακτική, αλλά συμπληρωματική, εφαρμογή των αρχών VLA στη λειτουργία UAVs [43].

4.4.4 Κοινά μοτίβα στα συστήματα UAV-VLA

Ανάμεσα στα τρία αντιπροσωπευτικά μοντέλα αναδύονται ορισμένα κοινά αρχιτεκτονικά και μεθοδολογικά χαρακτηριστικά. Καταρχάς, τα UAV-κεντρικά VLA συστήματα στηρίζονται σταθερά σε LLM backbones, τα οποία παρέχουν σημασιολογικό συλλογισμό, κατανόηση φυσικής γλώσσας και λήψη αποφάσεων υψηλού επιπέδου. Είτε αφορά έλεγχο χαμηλού επιπέδου RaceVLA, είτε γνωστική επεξεργασία CognitiveDrone, είτε σχεδιασμό αποστολών UAV-VLA, ένα γλωσσικό υποσύστημα βασισμένο σε transformer αποτελεί τον πυρήνα της πολιτικής.

Δεύτερον, οι vision encoders ή τα vision-language modules προσφέρουν τη perceptual θεμελίωση που είναι απαραίτητη για τη λειτουργία των UAVs. Στα RaceVLA και CognitiveDrone, οι κωδικοποιητές επεξεργάζονται εικόνες από κάμερες επί του σκάφους, ενώ στο UAV-VLA η εισροή αποτελείται από δορυφορικές εικόνες. Τα προκύπτοντα embeddings συμβάλλουν στην κατανόηση του περιβάλλοντος και καθοδηγούν την επιλογή δράσεων.

Τρίτον, κάθε UAV-VLA σύστημα προσαρμόζει την κεφαλή δράσης στο επίπεδο αφαίρεσης των απαιτούμενων ενεργειών: συνεχείς διανυσματικοί έλεγχοι για αγωνιστική πτήση, συμβολικές γνωστικές εξόδους για το CognitiveDrone ή ακολουθίες waypoints για τον σχεδιασμό αποστολών. Αυτά τα μοτίβα υποδεικνύουν πώς το γενικό VLA πλαίσιο μπορεί να εξειδικευθεί για εναέρια ρομποτική, προσφέροντας μια φυσική εννοιολογική γέφυρα προς UAV συστήματα που αξιοποιούν μοντέλα βασισμένα σε LLaMA, όπως αυτό που αναπτύσσεται στη συγκεκριμένη διατριβή.

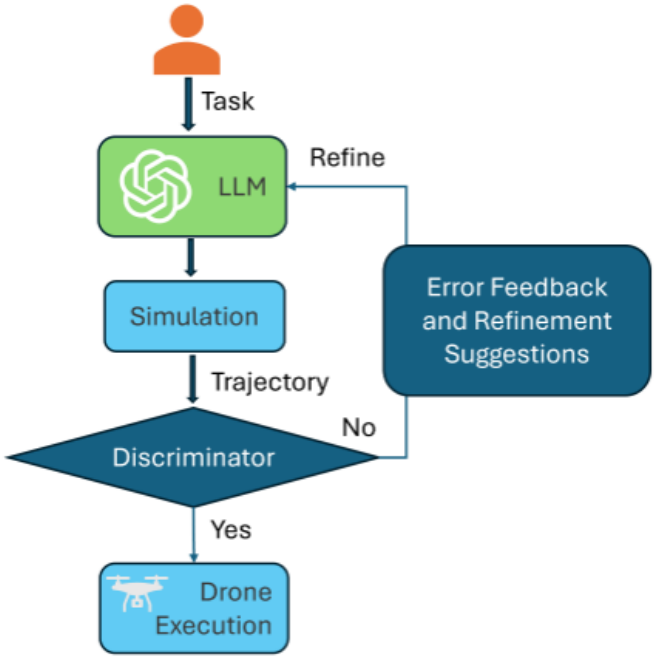
4.5 LLM vs VLA

Καθώς τα μοντέλα Vision language action αρχίζουν να επηρεάζουν την αυτονομία των UAVs, είναι κρίσιμο να διακρίνουμε μεταξύ αρχιτεκτονικών που βασίζονται αποκλειστικά σε μεγάλα γλωσσικά μοντέλα για τη λήψη αποφάσεων και εκείνων που υιοθετούν μια ενοποιημένη προσέγγιση VLA, όπου η αντίληψη, η κατανόηση εντολών και η παραγωγή

δράσεων ενσωματώνονται σε ένα ενιαίο σύστημα. Αυτή η ενότητα διευκρινίζει τη σχέση των προσεγγίσεων που βασίζονται σε LLM με τα ολοκληρωμένα συστήματα VLA.

4.5.1 Χειρισμός με καθαρά LLM – driven drone agents

Μια ειδική κατηγορία μεθοδολογιών ελέγχου drone βασίζεται κυρίως σε ένα εκτενές γλωσσικό μοντέλο ως τον κύριο μηχανισμό λήψης αποφάσεων, αντί να υιοθετεί μια ενοποιημένη αρχιτεκτονική VLA. Ένα χαρακτηριστικό παράδειγμα είναι το σύστημα που αναπτύχθηκε στη μεταπτυχιακή διατριβή του Wang [44], το οποίο εισάγει έναν κλειστό βρόχο συλλογιστικής και επαναληπτικής βελτίωσης για αυτόνομο έλεγχο UAV. Στο συγκεκριμένο πλαίσιο, το LLM λειτουργεί ως ανώτερος γνωστικός πράκτορας: ερμηνεύει τις εντολές του χρήστη, αποδομεί τις εργασίες σε εκτελέσιμα βήματα και παράγει υποψήφιες τροχιές, χωρίς να ενσωματώνει άμεσα τα ακατέργαστα οπτικά δεδομένα στο ίδιο μοντέλο.



Εικόνα 4.10 LLM – driven drone agent [44].

Η **Εικόνα 4.10** παρουσιάζει τον συνολικό βρόχο ελέγχου που χρησιμοποιείται σε αυτή την προσέγγιση. Με την παραλαβή μιας εντολής σε φυσική γλώσσα, το LLM δημιουργεί μια αρχική τροχιά ή μια ακολουθία υψηλού επιπέδου ενεργειών για το drone. Η έξοδος αυτή μεταφέρεται σε ένα περιβάλλον προσομοίωσης, το οποίο αξιολογεί τη σκοπιμότητα και την ασφάλεια της προτεινόμενης κίνησης. Ένα διακριτικό module επιβλέπει την τροχιά για ενδεχόμενα σφάλματα, όπως μη ασφαλείς μανούβρες, κινηματικές ασυνέπειες ή παραβάσεις των περιορισμών της αποστολής. Εάν εντοπιστούν προβλήματα, ένα module

ανατροφοδότησης παράγει προτάσεις διόρθωσης, οι οποίες επιστρέφονται στο LLM για αναθεώρηση της τροχιάς. Η διαδικασία αυτή επαναλαμβάνεται έως ότου το παραγόμενο σχέδιο πληροί τα κριτήρια αξιολόγησης, οπότε και υλοποιείται στο UAV.

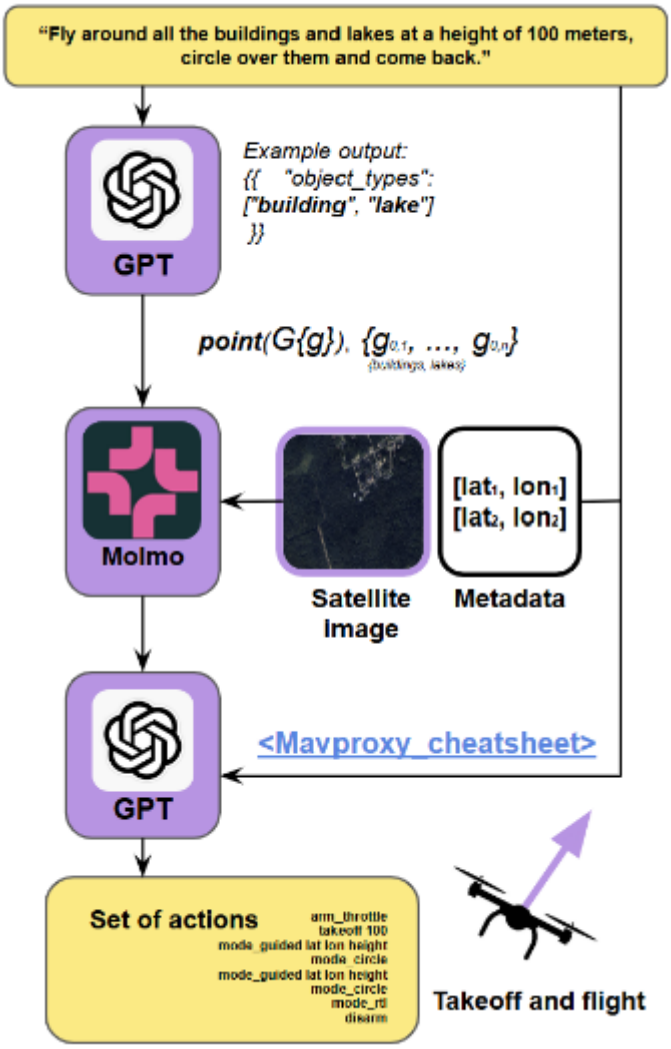
Η συγκεκριμένη αρχιτεκτονική αναδεικνύει ένα θεμελιώδες χαρακτηριστικό των καθαρά LLM driven agents : βασίζονται σε εξωτερικά module, όπως η προσομοίωση, οι διακριτές μονάδες και οι μηχανισμοί ανατροφοδότησης, για την επικύρωση και βελτίωση των αποφάσεών τους. Η αντίληψη, η συλλογιστική και ο έλεγχος παραμένουν διακριτά υποσυστήματα, με το LLM να λειτουργεί κυρίως ως συμβολικός σχεδιαστής και όχι ως μια ενιαία πολυτροπική πολιτική. Σε αντίθεση με τα VLA συστήματα, τα οποία ενοποιούν την οπτική επεξεργασία, την κατανόηση της γλώσσας και τη δημιουργία δράσεων σε ένα ενιαίο μοντέλο βασισμένο σε transformer, οι προσεγγίσεις που βασίζονται σε LLM διατηρούν σαφή διαχωρισμό μεταξύ της υψηλού επιπέδου συλλογιστικής και του χαμηλού επιπέδου ελέγχου. Κατά συνέπεια, παρέχουν υψηλή ερμηνευσιμότητα και ευελιξία στην αποδόμηση σύνθετων εργασιών, αλλά υστερούν σε σχέση με τη στενή σύνδεση αντίληψης-δράσης που διακρίνει τις αρχιτεκτονικές VLA.

4.5.2 UAV – VLA

Τα υβριδικά συστήματα κατέχουν μια ενδιάμεση θέση μεταξύ των καθαρά LLM-based προσεγγίσεων και των πλήρως ενοποιημένων VLA μοντέλων. Ένα χαρακτηριστικό παράδειγμα είναι το πλαίσιο UAV-VLA, το οποίο εκτελεί σχεδιασμό αποστολών υψηλού επιπέδου συνδυάζοντας τις δυνατότητες ενός LLM και ενός VLM μέσα σε μια αρθρωτή αρχιτεκτονική. Αντί να συνδυάζει την όραση, τη γλωσσική κατανόηση και την παραγωγή δράσεων σε έναν ενιαίο μετασχηματιστή, το UAV-VLA διαχωρίζει αυτές τις λειτουργίες σε εξειδικευμένα υποσυστήματα για την ερμηνεία της αποστολής, την οπτική ανάλυση και τη δομημένη παραγωγή ενεργειών [45].

Στην εικόνα 4.7 απεικονίζει τον αγωγό επεξεργασίας του UAV-VLA. Η διαδικασία αρχίζει με μια εντολή σε φυσική γλώσσα, όπως: «Πέτα γύρω από όλα τα κτίρια και τις λίμνες σε ύψος 100 μέτρων, κάνε κύκλους από πάνω τους και επέστρεψε.» Η εντολή αυτή μεταβιβάζεται σε ένα LLM, το οποίο εξάγει τους σημασιολογικούς στόχους και τις αντικειμενικές κατηγορίες ενδιαφέροντος για παράδειγμα κτίριο, λίμνη. Οι κατηγορίες αυτές καθοδηγούν το επόμενο module, ένα VLM (Molmo), το οποίο αναλύει δορυφορικές εικόνες και μεταδεδομένα για τον εντοπισμό σχετικών αντικειμένων ή γεωγραφικών δομών. Μέσω αυτού του βήματος

θεμελίωσης, η υψηλού επιπέδου περιγραφή της αποστολής μετατρέπεται σε χωρικούς στόχους, οι οποίοι εκφράζονται ως γεωγραφικές συντεταγμένες.



Εικόνα 4.11 Σύστημα UAV-VLA [45] .

Οι εντοπισμένες θέσεις επιστρέφονται στο LLM, το οποίο συνθέτει μια δομημένη ακολουθία ενεργειών ή εντολών συμβατών με MAVProxy. Αυτή η έξοδος περιλαμβάνει το ολοκληρωμένο mission script, το οποίο περιέχει ενέργειες όπως απογείωση, ρύθμιση ύψους, κυκλικές κινήσεις, πλοήγηση μεταξύ waypoints και επιστροφή στο σημείο εκκίνησης. Η εκτέλεση από το UAV διεξάγεται μόνο μετά την παραγωγή της τελικής λίστας ενεργειών, καθιστώντας το LLM υπεύθυνο για τον στρατηγικό σχεδιασμό, ενώ το VLM εκτελεί την οπτική θεμελίωση της πληροφορίας.

Η συγκεκριμένη υβριδική αρχιτεκτονική αναδεικνύει ένα σημαντικό εννοιολογικό σημείο: αν και τα LLMs παρέχουν συλλογιστική, ερμηνεία εντολών και παραγωγή ακολουθιών δράσης, δεν ενσωματώνουν τις εικόνες της κάμερας στο εσωτερικό τους αναπαραστατικό μοντέλο. αντίληψη ανατίθεται σε εξωτερικό VLM, ενώ το LLM εργάζεται πάνω σε ήδη ερμηνευμένη οπτική πληροφορία. Αυτός ο διαχωρισμός αντιτίθεται στα ενοποιημένα VLA μοντέλα, όπου οι οπτικές και γλωσσικές αναπαραστάσεις επεξεργάζονται συλλογικά εντός ενός ενιαίου transformer που παράγει άμεσα ενέργειες.

Το UAV-VLA καταγράφει ένα υβριδικό μοντέλο που διατηρεί τη modularity και επιτρέπει προηγμένο σχεδιασμό αποστολών, τοποθετώντας το ακριβώς μεταξύ των καθαρών LLM-based συστημάτων και των πλήρως ενοποιημένων πολιτικών ελέγχου τύπου VLA.

ΚΕΦΑΛΑΙΟ 5

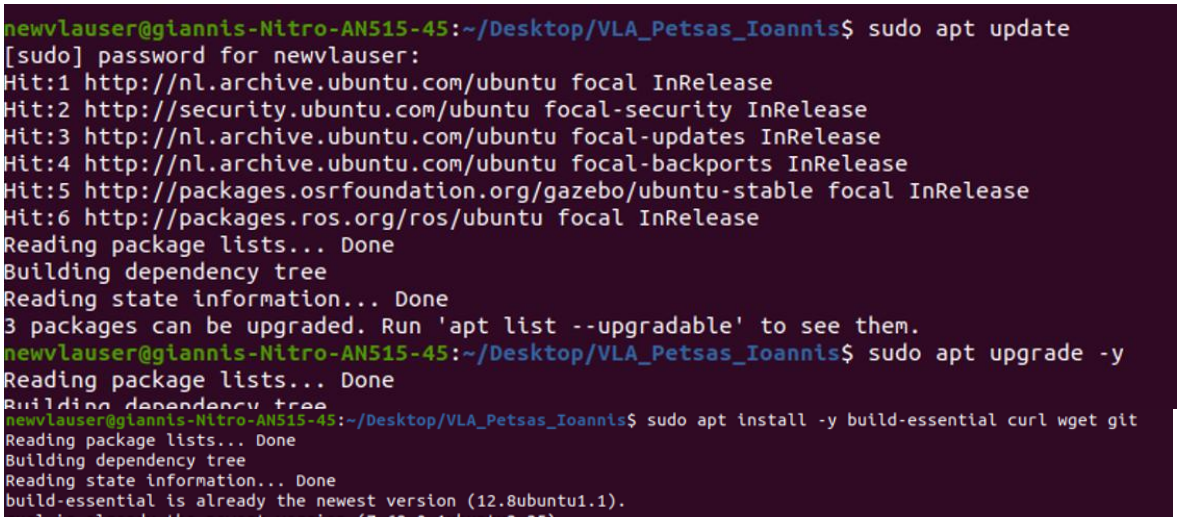
Υλοποίηση και ενσωμάτωση LLaMA/VLA στο περιβάλλον ROS–PX4

Η παρούσα ενότητα αναλύει λεπτομερώς τα βήματα που απαιτούνται για την εγκατάσταση, ρύθμιση και ενσωμάτωση ενός συστήματος VLA βασισμένου σε LLaMA σε περιβάλλον ROS Noetic, PX4 και Gazebo. Ο σκοπός του κεφαλαίου είναι η ολοκληρωμένη αναπαραγωγή της διαδικασίας, προκειμένου οποιοσδήποτε χρήστης να μπορεί να εφαρμόσει το ίδιο σύστημα και να επιτύχει τα ίδια λειτουργικά αποτελέσματα.

5.1 Προετοιμασία υπολογιστικού περιβάλλοντος

Η διαδικασία ξεκινά με την ενημέρωση του συστήματος και την εγκατάσταση των βασικών εργαλείων ανάπτυξης:

```
sudo apt update
sudo apt upgrade -y
sudo apt install -y build-essential curl wget git
```



Εικόνα 5.1 Προετοιμασία περιβάλλοντος.

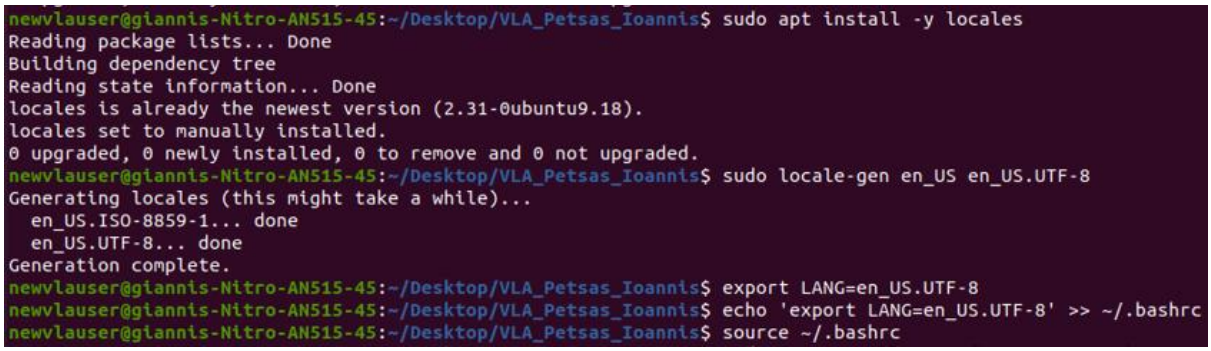
5.1.1 Ρύθμιση των locales (απαραίτητο για το ROS)

```
sudo apt install -y locales
sudo locale-gen en_US en_US.UTF-8
```

export LANG=en_US.UTF-8

echo 'export LANG=en_US.UTF-8' >> ~/.bashrc

source ~/.bashrc



Εικόνα 5.2 Προετοιμασία περιβάλλοντος.

5.2 Εγκατάσταση ROS noetic

Για τη χρήση gazebo και rviz απαιτείται η πλήρης εγκατάσταση του ros noetic.

5.2.1 Προσθήκη του apt repository του ROS

sudo apt install -y software-properties-common

sudo add-apt-repository universe

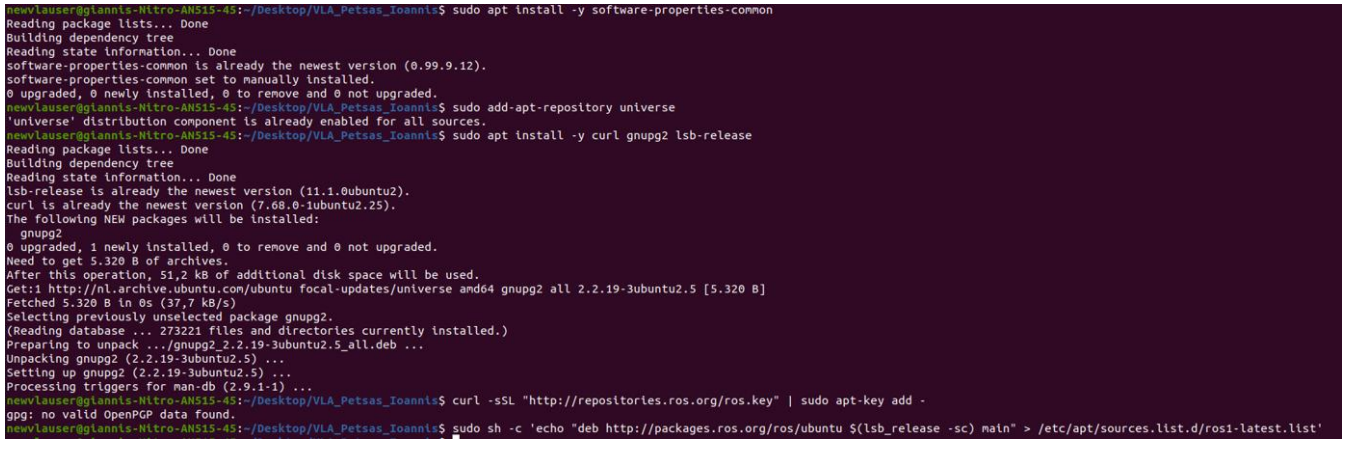
sudo apt install -y curl gnupg2 lsb-release

curl -sSL "http://repositories.ros.org/ros.key" | sudo apt-key add -

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >

/etc/apt/sources.list.d/ros1-latest.list'

sudo apt update



Εικόνα 5.21 Προετοιμασία περιβάλλοντος.

5.2.2 Εγκατάσταση ros noetic

sudo apt install -y ros-noetic-desktop-full

echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc

source ~/.bashrc

echo $ROS_DISTRO

Εικόνα 5.22 Εγκατάσταση ROS.

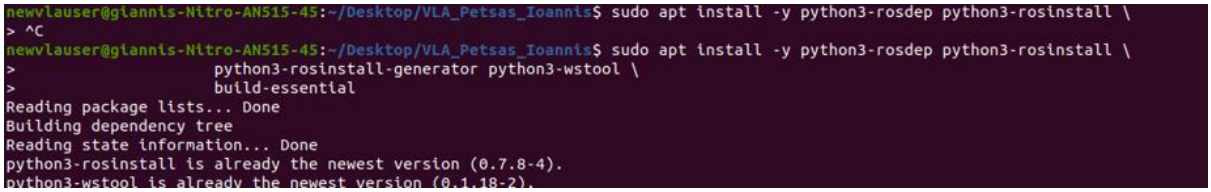
Εικόνα 5.23 Εγκατάσταση ROS.

5.2.3 Εγκατάσταση ROS εργαλείων

sudo apt install -y python3-rosdep python3-rosinstall \

python3-rosinstall-generator python3-wstool \

build-essential



Εικόνα 5.24 Εγκατάσταση χρήσιμων εργαλείων.

5.3 Εγκατάσταση εργαλείων px4 και εξαρτήσεων

Το PX4 απαιτεί επιπλέον βιβλιοθήκες για τη διαδικασία build και τη συνεργασία με Gazebo.

sudo apt install -y git wget zip python3-pip python3-jinja2 \

python3-numpy python3-yaml build-essential ninja-build \

exiftool kmod

```
newvlauser@giannis-Nitro-AN515-45:~/Desktop/VLA_Petsas_Ioannis$ sudo apt install -y git wget zip python3-pip python3-jinja2 \
>     python3-numpy python3-yaml build-essential ninja-build \
>     exiftool kmod
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libimage-exiftool-perl' instead of 'exiftool'
zip is already the newest version (3.0-11build1).
ninja-build is already the newest version (1.10.0-1build1).
build-essential is already the newest version (12.8ubuntu1.1).
```

Εικόνα 5.3 Εγκατάσταση χρήσιμων εργαλείων.

5.3.1 Κατέβασμα του αυτόματου πιλότου PX4

git clone https://github.com/PX4/PX4-Autopilot.git --recursive

```
newvlauser@giannis-Nitro-AN515-45:~/Desktop/VLA_Petsas_Ioannis$ git clone https://github.com/PX4/PX4-Autopilot.git --recursive
Cloning into 'PX4-Autopilot'...
remote: Enumerating objects: 528646, done.
remote: Counting objects: 100% (87/87), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 528646 (delta 56), reused 44 (delta 44), pack-reused 528559 (from 2)
Receiving objects: 100% (528646/528646), 513.82 MiB | 42.82 MiB/s, done.
Resolving deltas: 100% (379248/379248), done.
Updating files: 100% (13478/13478), done.
Submodule 'Tools/flightgear_bridge' (https://github.com/PX4/PX4-FlightGear-Bridge.git) registered for path 'Tools/simulation/flightgear/flightgear_bridge'
Submodule 'Tools/simulation/gazebo-classic/sitl_gazebo-classic' (https://github.com/PX4/PX4-SITL_gazebo-classic.git) registered for path 'Tools/simulation/gazebo
Submodule 'Tools/simulation/gz' (https://github.com/PX4/PX4-gazebo-models.git) registered for path 'Tools/simulation/gz'
Submodule 'Tools/simulation/jmavsim/jMAVSim' (https://github.com/PX4/jMAVSim.git) registered for path 'Tools/simulation/jmavsim/jMAVSim'
Submodule 'Tools/simulation/jsbsim/jsbsim_bridge' (https://github.com/PX4/px4-jsbsim-bridge.git) registered for path 'Tools/simulation/jsbsim/jsbsim_bridge'
Submodule 'boards/modalai/voxl2/libfc-sensor-api' (https://gitlab.com/voxl-public/voxl-sdk/core-libs/libfc-sensor-api.git) registered for path 'boards/modalai/vo
Submodule 'platforms/nuttx/NuttX/apps' (https://github.com/PX4/NuttX-apps.git) registered for path 'platforms/nuttx/NuttX/apps'
Submodule 'platforms/nuttx/NuttX/nuttx' (https://github.com/PX4/NuttX.git) registered for path 'platforms/nuttx/NuttX/nuttx'
Submodule 'src/drivers/actuators/vertiq_io/iq-module-communication-cpp' (https://github.com/PX4/iq-module-communication-cpp.git) registered for path 'src/drivers
```

Εικόνα 5.31 Αυτόματος πιλότος PX4 [46].

5.4 Εγκατάσταση gazebo και rviz

sudo apt install -y gazebo11

sudo apt install -y ros-noetic-gazebo-ros-pkgs ros-noetic-gazebo-ros-control

sudo apt install -y ros-noetic-rviz

```
newvlauser@giannis-Nitro-AN515-45:~/Desktop/VLA_Petsas_Ioannis$ gazebo --version
Gazebo multi-robot simulator, version 11.15.1
Copyright (C) 2012 Open Source Robotics Foundation.
Released under the Apache 2 License.
http://gazebosim.org


newvlauser@giannis-Nitro-AN515-45:~/Desktop/VLA_Petsas_Ioannis$ rviz --version
[INFO] [1765043914.528517335]: rviz version 1.14.26
[INFO] [1765043914.528573911]: compiled against Qt version 5.12.8
[INFO] [1765043914.528597449]: compiled against OGRE version 1.9.0 (Ghadamon)
[ERROR] [1765043914.528908405]: Error parsing command line: unrecognised option '--version'
newvlauser@giannis-Nitro-AN515-45:~/Desktop/VLA_Petsas_Ioannis$ sudo apt install -y gazebo11
Reading package lists... Done
Building dependency tree
Reading state information... Done
gazebo11 is already the newest version (11.15.1-1~focal).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
newvlauser@giannis-Nitro-AN515-45:~/Desktop/VLA_Petsas_Ioannis$ sudo apt install -y ros-noetic-gazebo-ros-pkgs ros-noetic-gazebo-ros-control
Reading package lists... Done
Building dependency tree
Reading state information... Done
ros-noetic-gazebo-ros-control is already the newest version (2.9.3-1focal.20250521.005335).
ros-noetic-gazebo-ros-control set to manually installed.
ros-noetic-gazebo-ros-pkgs is already the newest version (2.9.3-1focal.20250521.011748).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
newvlauser@giannis-Nitro-AN515-45:~/Desktop/VLA_Petsas_Ioannis$ sudo apt install -y ros-noetic-rviz
Reading package lists... Done
Building dependency tree
```

Εικόνα 5.4 Εγκατάσταση χρήσιμων εργαλείων.

5.5 Εγκατάσταση PX4 toolchain και build του sitl

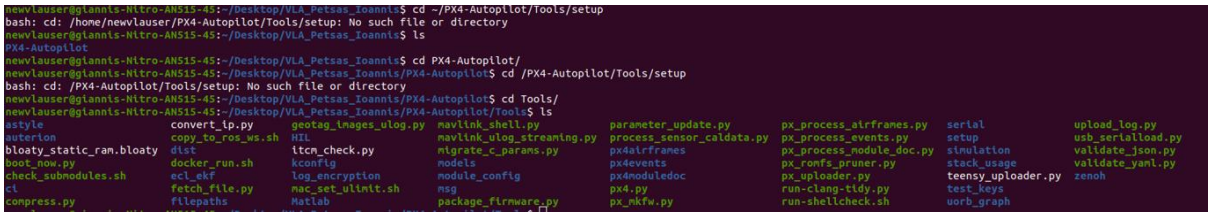
Μεταβαίνουμε στο setup script του PX4:

cd ~/PX4-Autopilot/Tools/setup

sudo ./ubuntu.sh

source ~/.profile

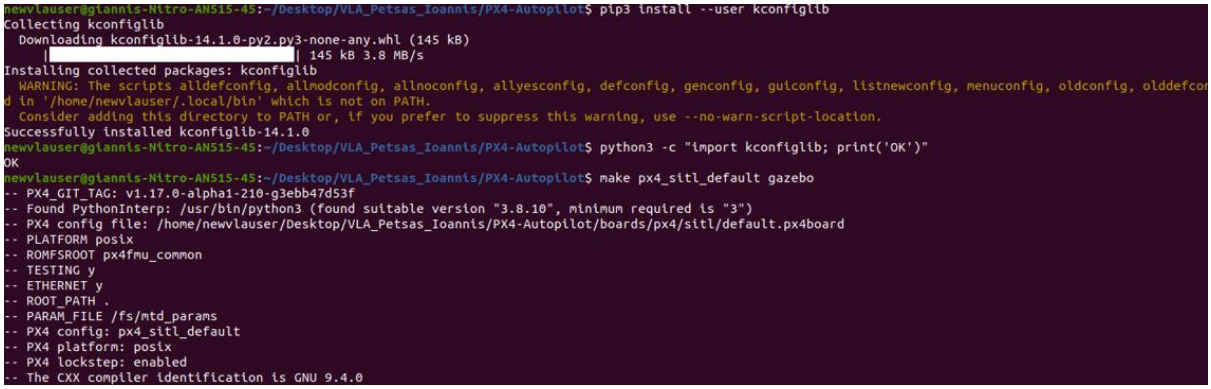
source ~/.bashrc



Εικόνα 5.5 Προετοιμασία αυτόματου πιλότου.

Αν η εντολή make px4_sitl_default gazebo αποτύχει λόγω kconfiglib:

pip3 install --user kconfiglib

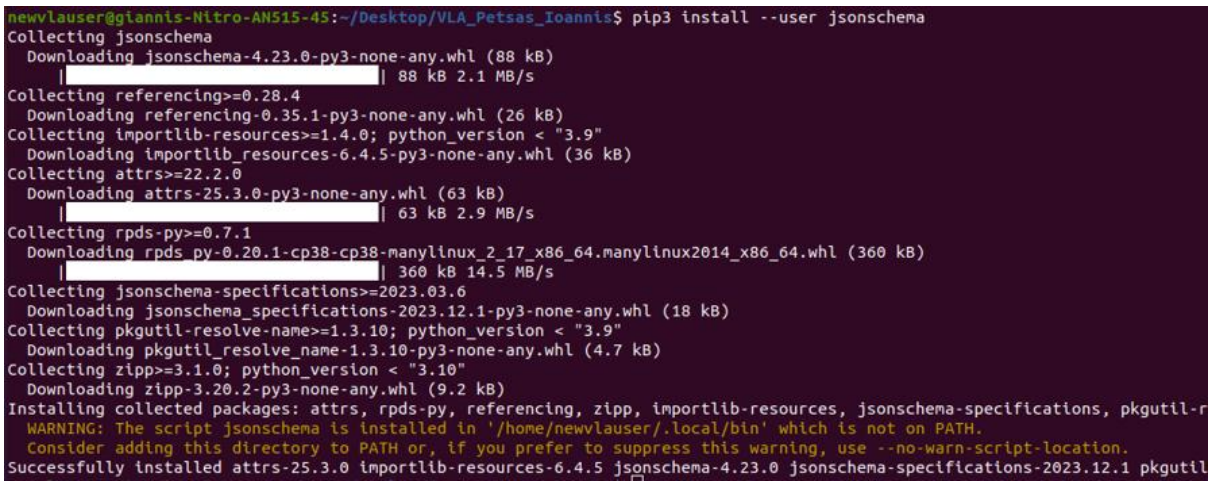


Εικόνα 5.51 Δοκιμαστικό άνοιγμα.

Αν εμφανιστεί σφάλμα για jsonschema:

pip3 install --user jsonschema

Εικόνα 5.52 Δοκιμαστικό άνοιγμα.



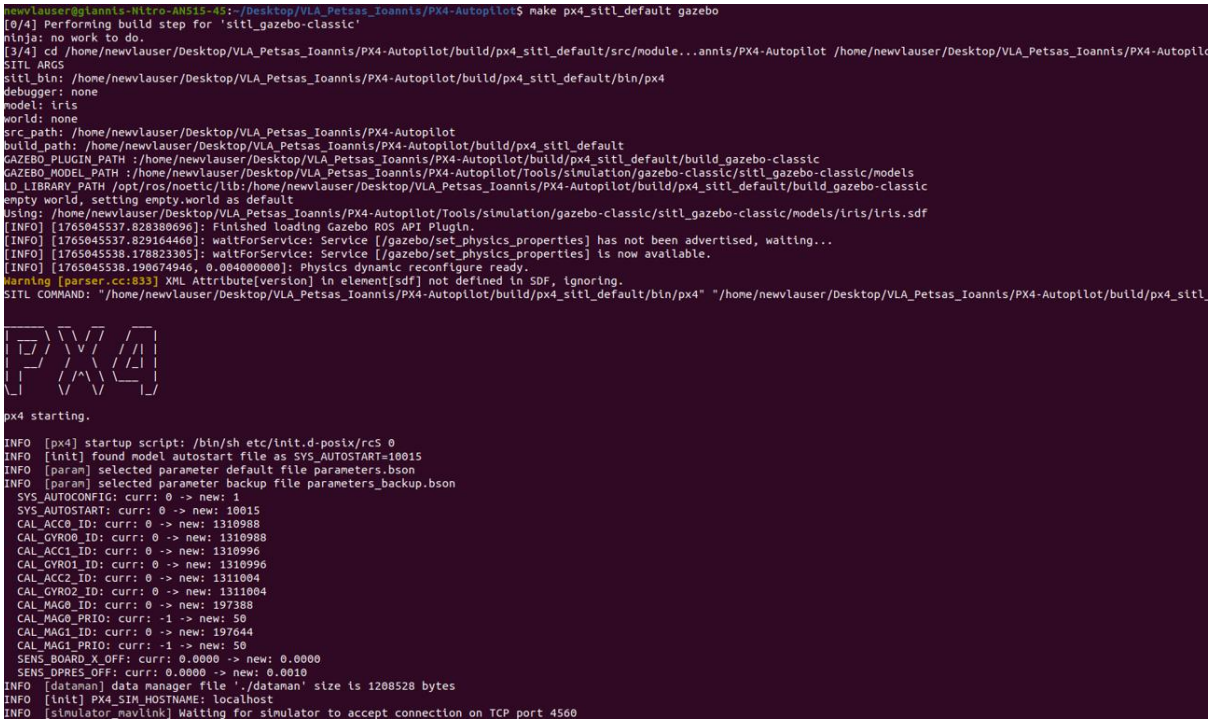
Εικόνα 5.53 Επίλυση σφάλματος.

Στη συνέχεια:

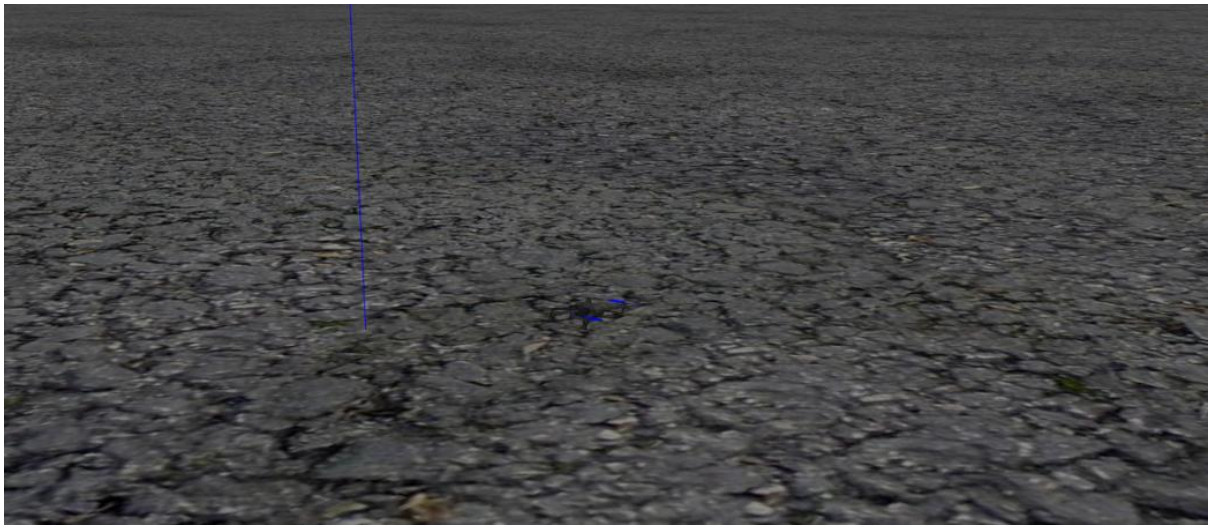
roscore

make px4_sitl_default gazebo

Αν όλα είναι σωστά, θα εμφανιστεί το Gazebo με το Iris drone.



Εικόνα 5.54 Έναρξη gazebo με PX4.



Εικόνα 5.55 Έναρξη gazebo με PX4.

5.6 Εγκατάσταση mavros και γεωγραφικών δεδομένων

sudo apt install -y ros-noetic-mavros ros-noetic-mavros-extras

wget

https://raw.githubusercontent.com/mavlink/mavros/master/mavros/scripts/install_geographiclib_datasets.sh

sudo bash install_geographiclib_datasets.sh

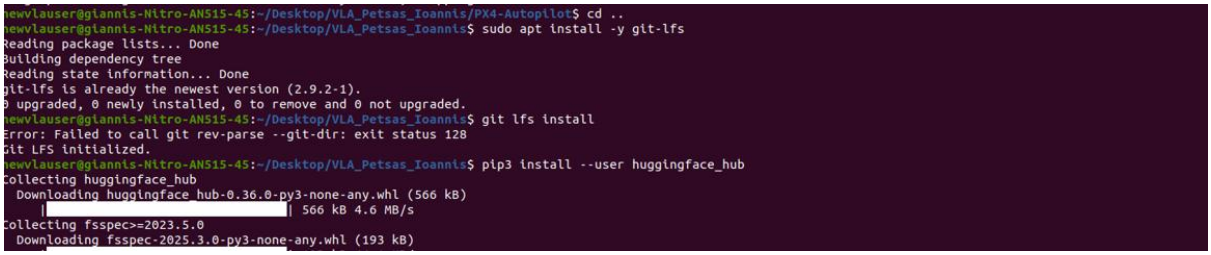
Εικόνα 5.6 Εγκατάσταση χρήσιμων εργαλείων.

5.7 Κατέβασμα RaceVLA και LLaMA

5.7.1 Εγκατάσταση εργαλείων HuggingFace

pip3 install --user huggingface_hub

huggingface-cli login



Εικόνα 5.7 Εγκατάσταση χρήσιμων εργαλείων.

Εικόνα 5.71 Hugging face login.

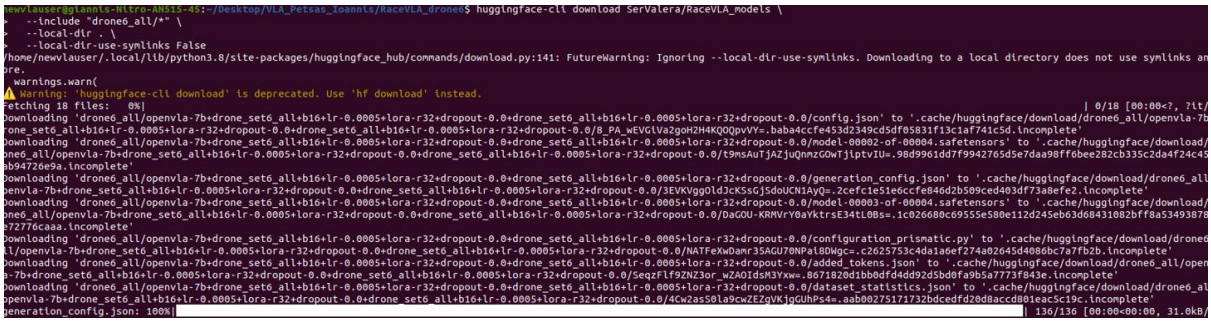
5.7.2 Κατέβασμα RaceVLA

huggingface-cli download SerValera/RaceVLA_models \

--include "drone6_all/*" \

--local-dir . \

--local-dir-use-symlinks False



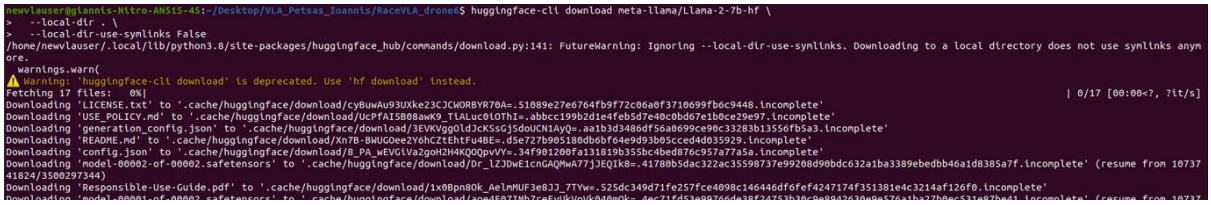
Εικόνα 5.72 Κατέβασμα RaceVLA [47].

5.7.3 Κατέβασμα LLaMA-2-7B

huggingface-cli download meta-llama/Llama-2-7b-hf \

--local-dir . \

--local-dir-use-symlinks False



Εικόνα 5.73 Κατέβασμα LLama [48].

5.8 Προσθήκη custom scripts στο PX4 project

Μεταφέρουμε τα δικά μας scripts:

Τοποθεσία:

~/Desktop/VLA_Petsas_Ioannis/PX4-Autopilot$

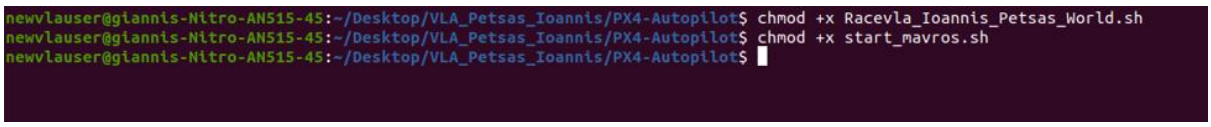
Αρχεία:

- Racevla_Ioannis_Petsas_World.sh
- start_mavros.sh

Κάνουμε τα scripts εκτελέσιμα:

chmod +x Racevla_Ioannis_Petsas_World.sh

chmod +x start_mavros.sh



Εικόνα 5.8 Δημιουργία executable αρχείων.

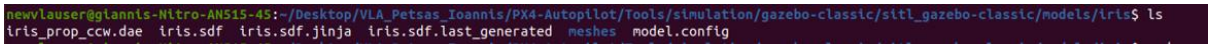
5.9 Προσθήκη custom sris model για vla

Τοποθετούμε τα τροποποιημένα αρχεία του drone:

~/Desktop/VLA_Petsas_Ioannis/PX4-Autopilot/Tools/simulation/gazebo-classic/sitl_gazebo-classic/models/iris

Αρχεία:

- iris.sdf
- iris.sdf.jinja
- iris.sdf.last_generated
- model.config



Εικόνα 5.9 Προσθήκη επεξεργασμένου μοντέλου.

5.10 Προσθήκη custom world file

Τοποθεσία:

~/Desktop/VLA_Petsas_Ioannis/PX4-Autopilot/Tools/simulation/gazebo-classic/sitl_gazebo-classic/worlds

Αρχείο:

- Racevla_Ioannis_Petsas.world

Εικόνα 5.10 Προσθήκη επεξεργασμένου κόσμου.

5.11 Εκκίνηση της προσομοίωσης

5.11.1 Εκκίνηση του Gazebo World

./Racevla_Ioannis_Petsas_World.sh

Σε άλλο terminal:

rostopic list

Επιβεβαιώνουμε ότι εμφανίζονται topics κάμερας όπως /iris/camera/image_raw.

```
newvlauser@giannis-Nitro-AN515-45:~/Desktop/VLA_Petsas_Ioannis/PX4-Autopilot$ rostopic list
/clock
/diagnostics
/gazebo/link_states
/gazebo/model_states
/gazebo/parameter_descriptions
/gazebo/parameter_updates
/gazebo/performance_metrics
/gazebo/set_link_state
/gazebo/set_model_state
/iris_camera/camera_info
/iris_camera/image_raw
/iris_camera/image_raw/compressed
/iris_camera/image_raw/compressed/parameter_descriptions
/iris_camera/image_raw/compressed/parameter_updates
/iris_camera/image_raw/compressedDepth
/iris_camera/image_raw/compressedDepth/parameter_descriptions
/iris_camera/image_raw/compressedDepth/parameter_updates
/iris_camera/image_raw/theora
/iris_camera/image_raw/theora/parameter_descriptions
/iris_camera/image_raw/theora/parameter_updates
/iris_camera/parameter_descriptions
/iris_camera/parameter_updates
/mavlink/from
/mavlink/gcs_ip
/mavlink/to
/mavros/actuator_control
/mavros/adsb/send
/mavros/adsb/vehicle
/mavros/altitude
/mavros/battery
/mavros/cam_imu_sync/cam_imu_stamp
/mavros/camera/image_captured
/mavros/cellular_status/status
/mavros/companion_process/status
/mavros/debug_value/debug
```

Εικόνα 5.11 Έλεγχος topic list .

5.11.2 Εκκίνηση MAVROS

./start_mavros.sh

```
newvlauser@giannis-Nitro-AN515-45:~/Desktop/VLA_Petsas_Ioannis/PX4-Autopilot$ ./start_mavros.sh
Starting MAVROS (PX4, UDP 14540) with ROS1 Noetic...
... logging to /home/newvlauser/.ros/log/ffcdf096-d2ef-11f0-a522-a7b663d78e83/roslaunch-giannis-Nitro-AN515-45-7567.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://giannis-Nitro-AN515-45:39559/

SUMMARY
========

CLEAR PARAMETERS
 * /mavros/

PARAMETERS
 * /mavros/camera/frame_id: base_link
 * /mavros/cmd/use_comp_id_system_control: False
 * /mavros/conn/heartbeat_rate: 1.0
 * /mavros/conn/system_time_rate: 1.0
```

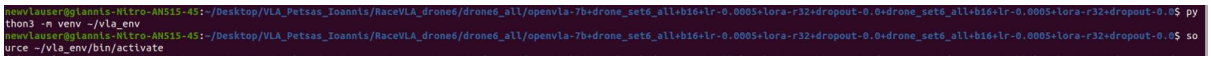
Εικόνα 5.111 Έλεγχος mavros.

5.12 Προετοιμασία Περιβάλλοντος για VLA Server

5.12.1 Δημιουργία Python Virtual Environment

python3 -m venv ~/vla_env

source ~/vla_env/bin/activate

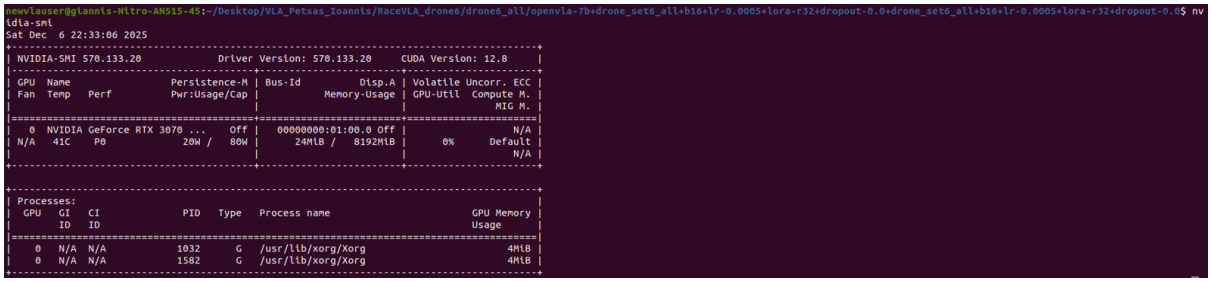


Εικόνα 5.121 Εγκατάσταση εργαλείων.

5.12.2 Εγκατάσταση PyTorch (CUDA 11.8)

Με βάση τα γραφικά του υπολογιστή βάζουμε τα κατάλληλα cuda κτλ.

Nvidia smi



Εικόνα 5.122 Nvidia smi.

pip uninstall -y torch torchvision torchaudio

pip install torch==2.4.1+cu118 torchvision==0.19.1+cu118 torchaudio==2.4.1+cu118 \

  --index-url https://download.pytorch.org/whl/cu118

Εικόνα 5.1221 Εγκατάσταση torch.

Έλεγχος:

python -c "import torch; print('Torch:', torch.__version__); print('CUDA available:', torch.cuda.is_available()); print('Device:', torch.cuda.get_device_name(0))"

Εικόνα 5.1222 Έλεγχος torch.

5.12.3 Εγκατάσταση Transformers + TIMM

pip install "timm==0.9.16" "transformers==4.46.3"

Εικόνα 5.123 Έλεγχος torch.

Έλεγχος:

python -c "import timm, transformers; print('timm:', timm.__version__); print('transformers:', transformers.__version__)"

5.12.4 Εγκατάσταση Βιβλιοθηκών VLA Server

pip install pillow numpy opencv-python

pip install huggingface_hub

5.13 Εκκίνηση του Vision Language Action Server

Στον φάκελο του RaceVLA:

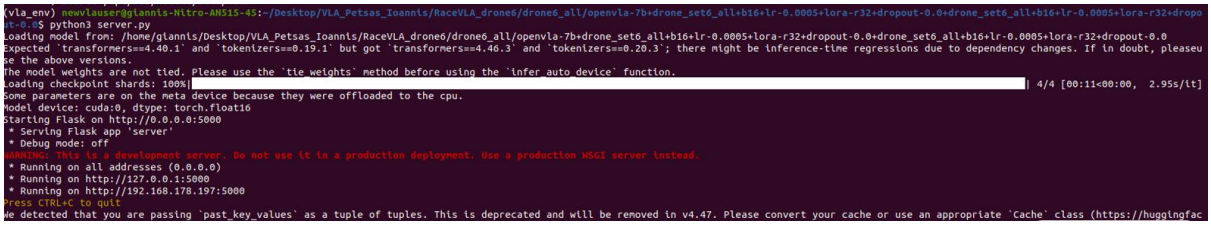
~/Desktop/VLA_Petsas_Ioannis/RaceVLA_drone6/drone6_all/openvla-7b+drone_set6_all+...

Εκτελούμε:

source /opt/ros/noetic/setup.bash

source ~/vla_env/bin/activate

python3 server.py

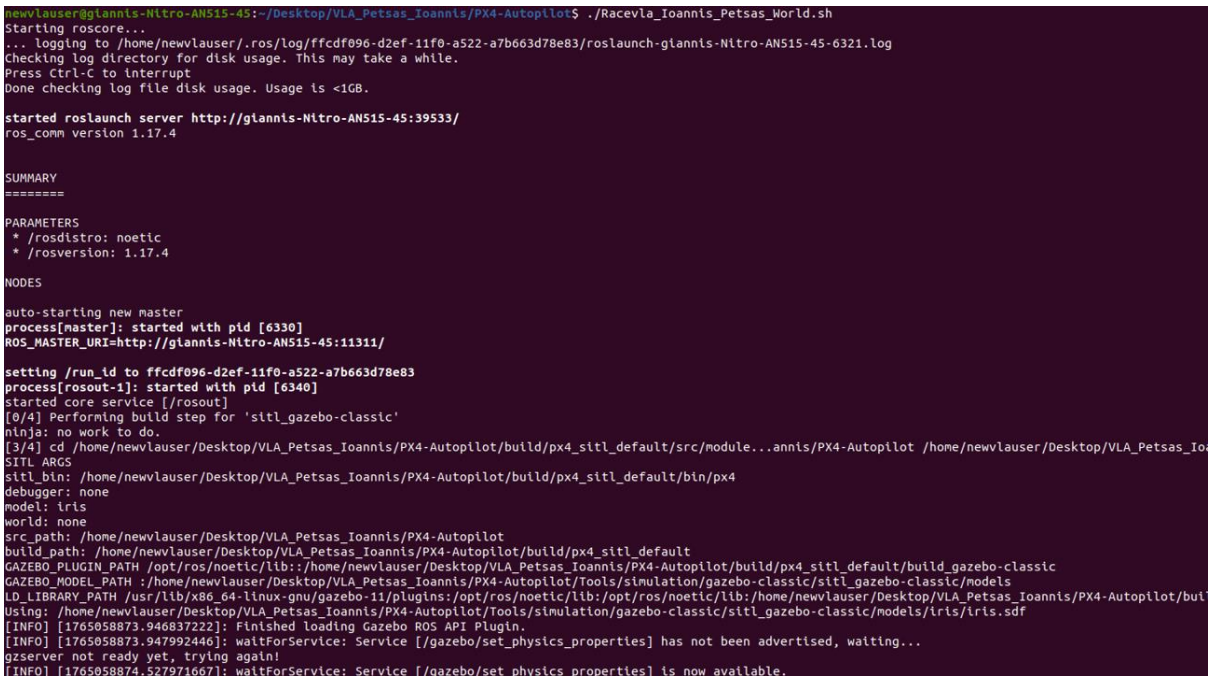


Εικόνα 5.13 Ξεκίνημα server.

Σε άλλο terminal:

cd ~/Desktop/VLA_Petsas_Ioannis/PX4-Autopilot

./Racevla_Ioannis_Petsas_World.sh



Εικόνα 5.131 Ξεκίνημα PX4.

Σε τρίτο terminal:

source /opt/ros/noetic/setup.bash

./start_mavros.sh

Εικόνα 5.132 Ξεκίνημα mavros.

Σε τέταρτο terminal:

cd ~/Desktop/VLA_Petsas_Ioannis/RaceVLA_drone6/drone6_all/openvla-7b+...

python3 drone.py

Εικόνα 5.133 Ξεκίνημα drone.py .

Αν όλα λειτουργούν σωστά θα εμφανίζονται στον σέρβερ logs τύπου:

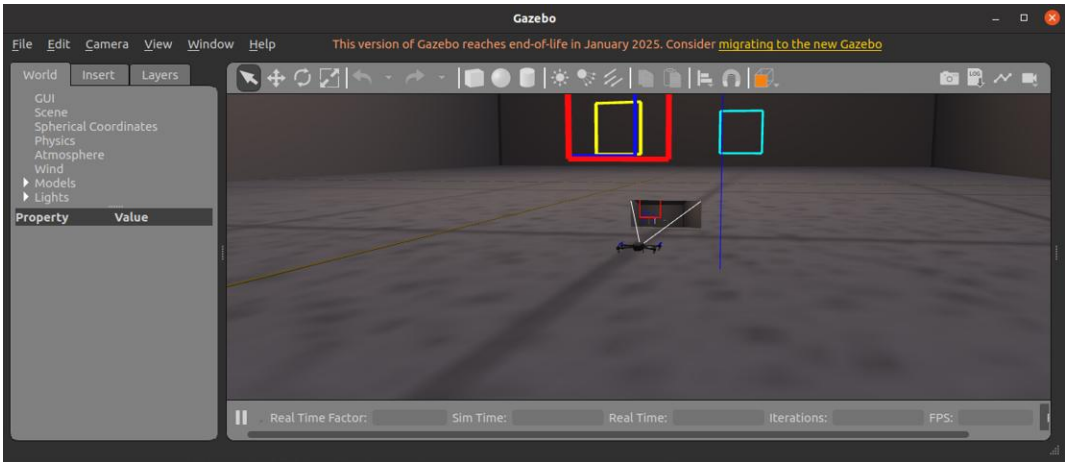
Inference 11.204s

Inference 10.342s

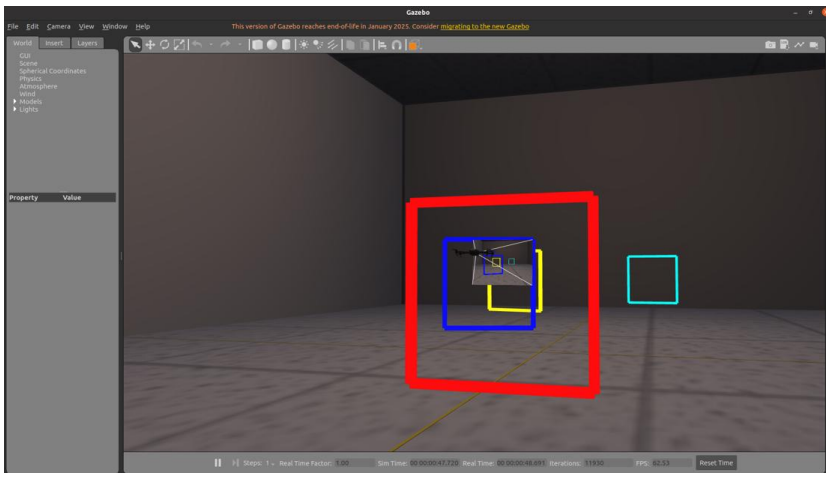
Inference 10.287s

127.0.0.1 - - "POST /predict_action HTTP/1.1" 200 –

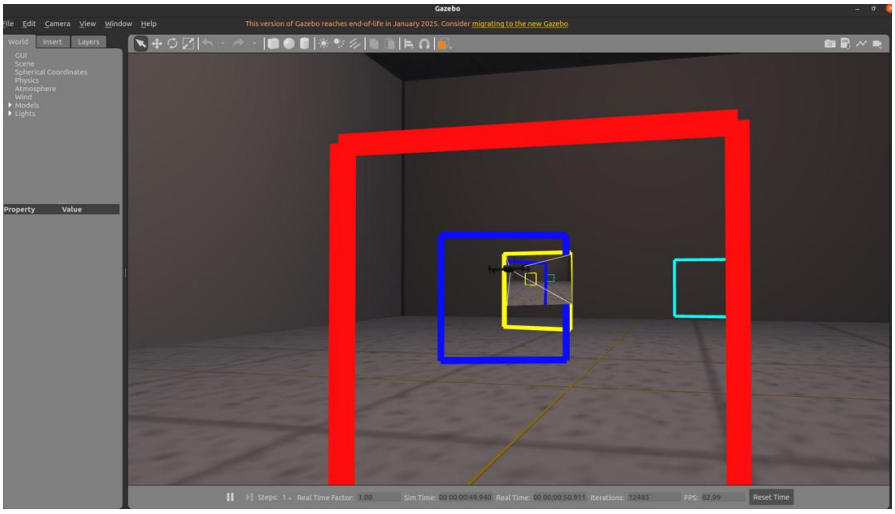
Καθώς και αυτόματη απογείωση (takeoff) του drone στο gazebo.



Εικόνα 5.134 Αρχικοποίηση κόσμου.



Εικόνα 5.135 Πτήση μέσω πυλών.



Εικόνα 5.136 Πτήση μέσω πυλών.

5.14 Λειτουργία του ολοκληρωμένου συστήματος

Μετά την ολοκλήρωση της ενσωμάτωσης όλων των επιμέρους στοιχείων, το σύστημα λειτουργεί βάσει μιας αρχιτεκτονικής client-server που συνδέει τον υπολογιστικό σέρβερ με το περιβάλλον του drone μέσω μιας γέφυρας ROS. Η λειτουργία βασίζεται στη ροή δεδομένων από το drone προς το μοντέλο τεχνητής νοημοσύνης και αντίστροφα, διευκολύνοντας την πλήρη αυτόνομη λήψη αποφάσεων.

Συγκεκριμένα, η κάμερα του drone παρέχει σε πραγματικό χρόνο τα δεδομένα εικόνας, τα οποία αποστέλλονται μέσω της ROS γέφυρας στον VLA σέρβερ. Στο VLA μοντέλο, η εικόνα επεξεργάζεται σε συνδυασμό με τη λεκτική εντολή που έχει καθοριστεί από τον

χρήστη. Το αποτέλεσμα της επεξεργασίας είναι ένα διάνυσμα ενεργειών, το οποίο περιγράφει τις κινήσεις που πρέπει να εκτελέσει το drone, όπως ταχύτητα προς τα εμπρός, στροφή και ύψος.

Η έξοδος αυτή επιστρέφεται μέσω του ίδιου περιβάλλοντος ROS και μετατρέπεται σε μηνύματα twist, τα οποία αποστέλλονται στο mavros, το οποίο είναι υπεύθυνο για τη μετάφραση αυτών των εντολών σε εντολές πτήσης προς τον αυτόματο πιλότο PX4. Με αυτόν τον τρόπο, επιτυγχάνεται η αυτοματοποιημένη εκτέλεση εντολών χωρίς την ανάγκη ανθρώπινης παρέμβασης στον βρόχο ελέγχου.

5.15 Ροή λειτουργίας και επικοινωνία κόμβων

Η συνολική λειτουργία μπορεί να περιγράφει ως εξής:

1. Το drone παρέχει ροή οπτικών δεδομένων.
2. Η γέφυρα ROS συλλέγει τα δεδομένα εικόνας και τα αποστέλλει στον διακομιστή VLA.
3. Το μοντέλο VLA συνδυάζει την εικόνα με τη λεκτική εντολή, εκτελεί συμπερασματική διαδικασία και παράγει την ενέργεια που αντιστοιχεί στη συγκεκριμένη κατάσταση.
4. Οι εντολές επιστρέφουν στη γέφυρα ROS και προωθούνται στο MAVROS.
5. Ο MAVROS διαμορφώνει και αποστέλλει τις εντολές στον PX4, ο οποίος τις εκτελεί στο drone.

Αυτή η αλληλουχία εκτελείται σε πραγματικό χρόνο, επιτρέποντας στο drone να αντιδρά δυναμικά σε περιβαλλοντικά ερεθίσματα, όπως μεταβολές στη θέση μιας πύλης ή σε διαφορετικά χρώματα στόχων.

5.16 Αποτελέσματα και επιτυχία του συστήματος

Η παρούσα υλοποίηση έχει δημιουργήσει ένα σύστημα ικανό να αντιλαμβάνεται, να ερμηνεύει και να εκτελεί εντολές σε φυσική γλώσσα. Το drone έχει τη δυνατότητα να αναγνωρίζει οπτικά στοιχεία στο περιβάλλον, όπως πύλες συγκεκριμένου χρώματος.

Η επικοινωνία μεταξύ ROS, MAVROS και PX4 αποδείχθηκε σταθερή και αξιόπιστη, με ελάχιστη καθυστέρηση. Το VLA μοντέλο επεξεργάζεται αποτελεσματικά τα δεδομένα, επιτυγχάνοντας ικανοποιητική αντιστοίχιση εντολών και ενεργειών, γεγονός που αποδεικνύει την ορθή ενσωμάτωση της λογικής VLA στο σύστημα.

5.17 Περιορισμοί και προοπτικές βελτίωσης

Αν και η ενσωμάτωσή της υπήρξε επιτυχής, το σύστημα δεν λειτουργεί ακόμη απολύτως ομαλά σε όλες τις περιπτώσεις. Ο κύριος λόγος έγκειται στο ότι το μοντέλο VLA έχει εκπαιδευτεί με πραγματικά δεδομένα πτήσεων, χρησιμοποιώντας κάμερες και πύλες που διαφέρουν από αυτές της προσομοίωσης. Έτσι, ορισμένες εντολές δεν ερμηνεύονται με ακρίβεια, ειδικά όταν οι συνθήκες φωτισμού ή οι χρωματικές αποχρώσεις διαφέρουν από το εκπαιδευτικό σύνολο δεδομένων.

Επιπλέον, το μοντέλο απαιτεί επιπλέον fine-tunning για το συγκεκριμένο πτητικό περιβάλλον, ώστε να προσαρμοστεί καταλληλότερα στη γωνία της κάμερας, στις αποστάσεις και στις δυναμικές κινήσεις του drone. Παρά τις αποκλίσεις, το σύστημα αποδεικνύει ότι η ενσωμάτωση ενός VLA σε πραγματικό ή προσομοιωμένο περιβάλλον UAV είναι εφικτή και λειτουργική.

Στο μέλλον, η απόδοση μπορεί να ενισχυθεί με την εκπαίδευση του μοντέλου σε στοχευμένα σύνολα δεδομένων, τα οποία θα περιλαμβάνουν τη συγκεκριμένη διάταξη του drone, τις κάμερες, τα χρώματα των πυλών και τις συνθήκες φωτισμού που επικρατούν κατά τις δοκιμές.

ΣΥΜΠΕΡΑΣΜΑΤΑ

1. Η παρούσα διπλωματική εργασία επικεντρώθηκε στη μελέτη, υλοποίηση και αξιολόγηση ενός σύγχρονου συστήματος αυτόνομης πλοήγησης μη επανδρωμένου αεροσκάφους, βασισμένου σε αρχιτεκτονική vision language action. Ο κύριος στόχος της μελέτης ήταν η εξερεύνηση της δυνατότητας ενός τέτοιου συστήματος να συνδέσει τη φυσική γλώσσα με τον έλεγχο πτήσης ενός UAV, αξιοποιώντας οπτικά δεδομένα και ανώτερη γνωστική κατανόηση του περιβάλλοντος.
2. Στο πλαίσιο αυτό, σχεδιάστηκε και υλοποιήθηκε μια ολοκληρωμένη αρχιτεκτονική που συνδυάζει το ROS, το PX4, το MAVROS και ένα μοντέλο VLA, με αρχιτεκτονική client-server. Η επιλογή αυτής της προσέγγισης επέτρεψε τον σαφή διαχωρισμό μεταξύ του υπολογιστικά απαιτητικού τμήματος VLA server και του συστήματος ελέγχου πτήσης, διασφαλίζοντας επεκτασιμότητα, ευελιξία και δυνατότητα μελλοντικής μεταφοράς σε πραγματικό υλικό.
3. Ένα από τα κύρια επιτεύγματα της εργασίας ήταν η επιτυχής ενσωμάτωση του μοντέλου VLA στο περιβάλλον ROS, διευκολύνοντας την επεξεργασία οπτικών δεδομένων σε πραγματικό χρόνο και τη μετατροπή λεκτικών εντολών σε συνεχείς εντολές κίνησης. Το σύστημα κατόρθωσε να πλοηγηθεί σε προσομοιωμένο περιβάλλον, διασχίζοντας διαδοχικά πολλές πύλες, αποδεικνύοντας ότι η γενική λογική του VLA είναι εφαρμόσιμη σε σύνθετα σενάρια πλοήγησης UAV.
4. Παράλληλα, η εργασία ανέδειξε με σαφήνεια τα πλεονεκτήματα της χρήσης VLA μοντέλων σε σχέση με πιο παραδοσιακές προσεγγίσεις αυτόνομης πτήσης. Η ικανότητα παροχής εντολών σε φυσική γλώσσα επιτρέπει έναν ανώτερο βαθμό αλληλεπίδρασης ανθρώπου–μηχανής, περιορίζοντας την ανάγκη για προκαθορισμένα σενάρια ή αυστηρά προγραμματισμένες διαδρομές. Το UAV αναλαμβάνει έναν πιο «γνωστικό» ρόλο, όπου η πλοήγηση δεν στηρίζεται αποκλειστικά σε αριθμητικά δεδομένα, αλλά σε αντιληπτική κατανόηση του περιβάλλοντος.
5. Ωστόσο, η πειραματική αξιολόγηση του συστήματος κατέδειξε και σημαντικούς περιορισμούς. Ένας από τους βασικότερους αφορά το γεγονός ότι το χρησιμοποιούμενο VLA μοντέλο έχει εκπαιδευτεί κυρίως σε δεδομένα πραγματικών drones, με συγκεκριμένα χαρακτηριστικά καμερών, χρωματισμών και γεωμετρίας πυλών. Η διαφορά μεταξύ του προσομοιωμένου περιβάλλοντος και των πραγματικών

δεδομένων εκπαίδευσης επηρέασε την ακρίβεια και τη σταθερότητα της πλοήγησης, ιδιαίτερα σε οριακές περιπτώσεις.

6. Επιπλέον, παρατηρήθηκαν καθυστερήσεις στην απόκριση του συστήματος, οι οποίες οφείλονται κυρίως στο υπολογιστικό κόστος της επεξεργασίας εικόνας και της παραγωγής ενεργειών από το μοντέλο. Αν και η αρχιτεκτονική server–bridge μετρίασε σε μεγάλο βαθμό αυτό το πρόβλημα, η καθυστέρηση παραμένει κρίσιμος παράγοντας για εφαρμογές πραγματικού χρόνου, ειδικά σε υψηλές ταχύτητες πτήσης.
7. Ένα επιπλέον σημαντικό συμπέρασμα σχετίζεται με τη γενίκευση του μοντέλου. Παρά την ικανότητα του συστήματος να εκτελεί βασικές εντολές πλοήγησης, η απόδοσή του εξαρτάται σε μεγάλο βαθμό από την εγγύτητα των συνθηκών λειτουργίας στα δεδομένα εκπαίδευσης. Αυτό υπογραμμίζει την ανάγκη για περαιτέρω εκπαίδευση ή fine-tuning των VLA μοντέλων σε δεδομένα που αντικατοπτρίζουν καλύτερα το εκάστοτε περιβάλλον χρήσης.
8. Συνολικά, η εργασία απέδειξε ότι η ενσωμάτωση μοντέλων VLA σε συστήματα UAV είναι εφικτή και ιδιαίτερα υποσχόμενη, αλλά παραμένει σε μεταβατικό στάδιο. Τα αποτελέσματα υποδεικνύουν ότι τα VLA συστήματα δύνανται να λειτουργούν ως εξελιγμένοι “γνωστικοί πιλότοι”, οι οποίοι συνεργάζονται με παραδοσιακούς αυτόματους πιλότους όπως το PX4, αντί να τους αντικαθιστούν ολοκληρωτικά.
9. Συμπερασματικά, η παρούσα διπλωματική εργασία προσφέρει συνεισφορά σε πρακτικό και ερευνητικό επίπεδο, παρουσιάζοντας μια πλήρως λειτουργική αρχιτεκτονική VLA–ROS–PX4, αναδεικνύοντας τα πλεονεκτήματα και τους περιορισμούς της. Αποτελεί ένα κρίσιμο βήμα προς την κατεύθυνση της πιο φυσικής, ευέλικτης και έξυπνης αλληλεπίδρασης ανθρώπου και αυτόνομων ιπτάμενων συστημάτων, διευκολύνοντας τη μελλοντική έρευνα και βελτιστοποίηση σε πραγματικές συνθήκες πτήσης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] R. Guebsi, S. Mami, and K. Chokmani, "Drones in Precision Agriculture: A Comprehensive Review of Applications, Technologies, and Challenges," *Drones*, vol. 8, no. 11, p. 686, Nov. 2024, doi: 10.3390/drones8110686.

[2] "Military VTOL UAV | Long-Range Recon Drone | High Eye Airboxer," High Eye. Accessed: Nov. 27, 2025. [Online]. Available: https://www.higheye.com/

[3] S. Guan, Z. Zhu, and G. Wang, "A Review on UAV-Based Remote Sensing Technologies for Construction and Civil Applications," *Drones*, vol. 6, no. 5, p. 117, May 2022, doi: 10.3390/drones6050117.

[4] Y. Chang, Y. Cheng, U. Manzoor, and J. Murray, "A review of UAV autonomous navigation in GPS-denied environments," *Robot. Auton. Syst.*, vol. 170, p. 104533, Dec. 2023, doi: 10.1016/j.robot.2023.104533.

[5] "Unmanned aerial vehicle," *Wikipedia*. Nov. 08, 2025. Accessed: Nov. 10, 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Unmanned_aerial_vehicle&oldid=1321011687

[6] G. Beverley, "A Not-So-Short History of Unmanned Aerial Vehicles (UAV)," Consortiq. Accessed: Nov. 10, 2025. [Online]. Available: https://consortiq.com/uas-resources/short-history-unmanned-aerial-vehicles-uavs

[7] N. T. M. P. LIBRARY, "Tesla's radio, 1890s - Stock Image - C036/8325," Science Photo Library. Accessed: Nov. 27, 2025. [Online]. Available: https://www.sciencephoto.com/media/869867/view/tesla-s-radio-1890s

[8] "Gyroscope, Autopilot, Kettering Liberty Eagle 'Bug' | National Air and Space Museum." Accessed: Nov. 10, 2025. [Online]. Available: https://airandspace.si.edu/collection-objects/gyroscope-autopilot-kettering-liberty-eagle-bug/nasm_A20010524000

[9] "Kettering Aerial Torpedo 'Bug,'" National Museum of the United States Air Force™. Accessed: Nov. 27, 2025. [Online]. Available: https://www.nationalmuseum.af.mil/Visit/Museum-Exhibits/Fact-Sheets/Display/Article/198095/kettering-aerial-torpedo-bug/

[10] "A Brief History of Drones," Imperial War Museums. Accessed: Nov. 10, 2025. [Online]. Available: https://www.iwm.org.uk/history/a-brief-history-of-drones

[11] Ross, "Ryan 147 Lightning Bug," From Balloons to Drones. Accessed: Nov. 27, 2025. [Online]. Available: https://balloonstodrones.com/tag/ryan-147-lightning-bug/

[12] J. Stoner, "Drone Sensors & Cameras," Fly Eye. Accessed: Nov. 13, 2025. [Online]. Available: https://www.flyeye.io/drone-technology-sensors-cameras/

[13] "Lidar Drone: Everything you need to know about LiDARs on UAVs," YellowScan. Accessed: Nov. 27, 2025. [Online]. Available: https://www.yellowscan.com/knowledge/lidar-drone/

[14] Surveygyaan, "Sensors used in drones," Medium. Accessed: Nov. 13, 2025. [Online]. Available: https://surveygyaan.medium.com/sensors-used-in-drones-e6f29be61fb4

[15] Z. Zhang and L. Zhu, “A Review on Unmanned Aerial Vehicle Remote Sensing: Platforms, Sensors, Data Processing Methods, and Applications,” *Drones*, vol. 7, no. 6, p. 398, June 2023, doi: 10.3390/drones7060398.
[16] “ROS: Home.” Accessed: Nov. 27, 2025. [Online]. Available: https://www.ros.org/
[17] “Documentation - ROS Wiki.” Accessed: Nov. 17, 2025. [Online]. Available: https://wiki.ros.org/
[18] “Gazebo Simulation | PX4 Guide (main).” Accessed: Nov. 17, 2025. [Online]. Available: https://docs.px4.io/main/en/sim_gazebo_gz/?utm_source
[19] “Dronecode Calendar.” Accessed: Nov. 17, 2025. [Online]. Available: https://calendar.google.com/calendar/embed?title=Dronecode%20Calendar&mode=WEEK&height=600&wkst=1&bgcolor=%23FFFFFF&src=linuxfoundation.org_g21tvam24m7pm7jhev01bvlqh8%40group.calendar.google.com&color=%23691426&ctz=Europe%2FZurich
[20] G. Kalyani, “A Robot Operating System (ROS) based humanoid robot control,” masters, Middlesex University, 2017. Accessed: Nov. 18, 2025. [Online]. Available: https://repository.mdx.ac.uk/item/86x55
[21] “Gazebo.” Accessed: Nov. 27, 2025. [Online]. Available: https://gazebosim.org/home
[22] Γ. Χρυσάφης and G. Chrysafis, “Υλοποίηση ελεγκτών εμπέδησης και κατασκευή συστήματος συνεργαζόμενων ρομπότ,” Feb. 2022, doi: 10.26240/heal.ntua.22231.
[23] S. Cavallera, “Autonomous Docking System with Obstacle Avoidance for a Differential Drive Robot,” laurea, Politecnico di Torino, 2021. Accessed: Nov. 21, 2025. [Online]. Available: https://webthesis.biblio.polito.it/21022/
[24] “Software Overview,” PX4 Autopilot. Accessed: Nov. 22, 2025. [Online]. Available: https://px4.io/software/software-overview/
[25] “PX4 Architectural Overview | PX4 Guide (main).” Accessed: Nov. 22, 2025. [Online]. Available: https://docs.px4.io/main/en/concept/architecture
[26] “Controller Diagrams | PX4 Guide (main).” Accessed: Nov. 23, 2025. [Online]. Available: https://docs.px4.io/main/en/flight_stack/controller_diagrams
[27] “Nonlinear Quadrocopter Attitude Control.” Accessed: Nov. 23, 2025. [Online]. Available: https://www.research-collection.ethz.ch/entities/publication/f76418e7-82ad-49e2-a4cf-845dcf00207e
[28] “uORB Messaging | PX4 Guide (main).” Accessed: Nov. 24, 2025. [Online]. Available: https://docs.px4.io/main/en/middleware/uorb
[29] “Mixing and Actuators | PX4 User Guide (v1.12).” Accessed: Nov. 25, 2025. [Online]. Available: https://docs.px4.io/v1.12/en/concept/mixing.html
[30] “SensorCombined (UORB message) | PX4 Guide (main).” Accessed: Nov. 25, 2025. [Online]. Available: https://docs.px4.io/main/en/msg_docs/SensorCombined
[31] PX4, “PX4-Autopilot/src/modules/sensors/sensors.cpp at main · PX4/PX4-Autopilot,” GitHub. Accessed: Nov. 25, 2025. [Online]. Available: https://github.com/PX4/PX4-Autopilot/blob/main/src/modules/sensors/sensors.cpp
[32] “MAVLink Messaging | PX4 Guide (main).” Accessed: Nov. 27, 2025. [Online]. Available: https://docs.px4.io/main/en/mavlink/index

[33] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, "Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey," *IEEE Access*, vol. 7, pp. 87658–87680, 2019, doi: 10.1109/ACCESS.2019.2924410.

[34] F. Malacarne, "PX4 autopilot customization for non-standard gimbal and UWB peripherals," laurea, Politecnico di Torino, 2020. Accessed: Nov. 27, 2025. [Online]. Available: https://webthesis.biblio.polito.it/15919/

[35] Y. Zhong *et al.*, "A Survey on Vision-Language-Action Models: An Action Tokenization Perspective," July 02, 2025, *arXiv*: arXiv:2507.01925. doi: 10.48550/arXiv.2507.01925.

[36] A. Brohan *et al.*, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," July 28, 2023, *arXiv*: arXiv:2307.15818. doi: 10.48550/arXiv.2307.15818.

[37] Y. Ma, Z. Song, Y. Zhuang, J. Hao, and I. King, "A Survey on Vision-Language-Action Models for Embodied AI," Aug. 31, 2025, *arXiv*: arXiv:2405.14093. doi: 10.48550/arXiv.2405.14093.

[38] A. Brohan *et al.*, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," July 28, 2023, *arXiv*: arXiv:2307.15818. doi: 10.48550/arXiv.2307.15818.

[39] A. Brohan *et al.*, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control".

[40] *Psi-Robot/Awesome-VLA-Papers*. (Nov. 30, 2025). PsiBot （灵初智能）. Accessed: Dec. 02, 2025. [Online]. Available: https://github.com/Psi-Robot/Awesome-VLA-Papers

[41] V. Serpiva *et al.*, "RaceVLA: VLA-based Racing Drone Navigation with Human-like Behaviour," Mar. 04, 2025, *arXiv*: arXiv:2503.02572. doi: 10.48550/arXiv.2503.02572.

[42] A. Lykov *et al.*, "CognitiveDrone: A VLA Model and Evaluation Benchmark for Real-Time Cognitive Task Solving and Reasoning in UAVs," Mar. 03, 2025, *arXiv*: arXiv:2503.01378. doi: 10.48550/arXiv.2503.01378.

[43] O. Sautenkov *et al.*, "UAV-VLA: Vision-Language-Action System for Large Scale Aerial Mission Generation," May 13, 2025, *arXiv*: arXiv:2501.05014. doi: 10.48550/arXiv.2501.05014.

[44] W. Wang, "An enhanced reasoning framework with closed-loop refinement for reliable LLM-driven drone control: a thesis in Computer Science," Master of Science, University of Massachusetts Dartmouth, 2025. doi: 10.62791/20429.

[45] O. Sautenkov *et al.*, "UAV-VLA: Vision-Language-Action System for Large Scale Aerial Mission Generation," May 13, 2025, *arXiv*: arXiv:2501.05014. doi: 10.48550/arXiv.2501.05014.

[46] *PX4/PX4-Autopilot*. (Dec. 08, 2025). C++. PX4 Autopilot. Accessed: Dec. 08, 2025. [Online]. Available: https://github.com/PX4/PX4-Autopilot

[47] "SerValera/RaceVLA_models at main." Accessed: Dec. 08, 2025. [Online]. Available: https://huggingface.co/SerValera/RaceVLA_models/tree/main/drone6_all/openvla-7b%2Bdrone_set6_all%2Bb16%2Blr-0.0005%2Blora-r32%2Bdropout-0.0%2Bdrone_set6_all%2Bb16%2Blr-0.0005%2Blora-r32%2Bdropout-0.0

[48] "meta-llama/Llama-2-7b-hf · Hugging Face." Accessed: Dec. 08, 2025. [Online]. Available: https://huggingface.co/meta-llama/Llama-2-7b-hf