

**SWARM ROBOTICS ΑΠΟΤΙΜΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΕΛΕΓΧΟΥ  
ΡΟΜΠΟΤΙΚΟΥ ΣΜΗΝΟΥΣ UAV ΓΙΑ ΤΗΝ ΕΠΙΤΗΡΗΣΗ  
ΔΑΣΙΚΗΣ ΕΚΤΑΣΕΩΣ ΚΑΙ ΤΗΝ ΕΓΚΑΙΡΗ ΠΡΟΕΙΔΟΠΟΙΗΣΗ  
ΣΕ ΠΕΡΙΠΤΩΣΗ ΔΑΣΙΚΗΣ ΠΥΡΚΑΓΙΑΣ**



**ΚΑΤΡΑΔΗ ΖΑΧΑΡΟΥΛΑ**

**ΔΑΜΟΚΑ ΚΑΤΕΡΙΝΑ**

**ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ: ΔΡΟΣΟΣ ΧΡΗΣΤΟΣ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ  
ΕΦΑΡΜΟΓΩΝ ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ**

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Πίνακας εικόνων .....	3
Εισαγωγή .....	4
Abstract.....	6
Κεφάλαιο 1: Φυσικό Υπόβαθρο.....	7
1.1. Παράδειγμα συλλογικής ευφύιας στα μυρμήγκια.....	7
1.2. Νοημοσύνη σμήνους (swarm intelligence) .....	9
1.3. Κανόνες του σμήνους σύμφωνα με τον Κρεγκ Ρέινολντς.....	11
1.4. Αλγόριθμοι Νοημοσύνης Σμήνους .....	11
1.4.1. Αλγόριθμος Ant Colony Optimisation (ACO).....	11
1.4.2. Αλγόριθμος Particle Swarm Optimisation (PSO) .....	17
Κεφάλαιο 2: Ρομποτικό Σμήνος – Μη Επανδρωμένα αεροσκάφη UAV (Unmanned Aircraft Vehicles) .	27
2.1. Ιστορικό Υπόβαθρο.....	27
2.2. Κατηγορίες UAV .....	27
2.3. Fire Drones .....	30
2.4. Παραμετρική Παρακολούθηση Φωτιάς για ένα UAV .....	31
2.5. Συνεργασία Ομαδικού Εντοπισμού.....	33
2.5.1. Ελαχιστοποίηση Λανθάνοντα χρόνου .....	33
2.5.2. Αλγόριθμος για παρακολούθηση κατανεμημένης φωτιάς .....	36
2.5.3. Αλγόριθμος Εξισορρόπησης Φορτίου .....	38
2.6. Πλεονεκτήματα- Μειονεκτήματα .....	40
Κεφάλαιο 3: Βασικοί κώδικες.....	43
3.1. Κώδικας αλγορίθμου λειτουργίας σκάφους .....	43
3.2. Συμπληρωματικός κώδικας – αλγόριθμος «ανάγνωσης» ελέγχων .....	64
3.3. Σύγκριση κωδίκων .....	74
Συμπεράσματα .....	76
Βιβλιογραφία .....	78
Πηγές Διαδικτύου.....	79

## Πίνακας εικόνων

<i>Εικόνα 1. 1. Το 1ο μυρμήγκι επιλέγει μια διαδρομή προς την τροφή (F) και επιστρέφει στην φωλιά (N) αφήνοντας ίχνη φερομόνης, 2.Τα άλλα μυρμήγκια ακολουθούν κάποια απο τις 4 διαδρομές, 3. Η διαδρομή με την μεγαλύτερη εναπόθεσης φερομόνης, που είναι και η συντομότερη, γίνεται προτιμητέα.</i>	8
<i>Εικόνα 2. Μη επανδρωμένο αεροσκάφος Predator</i>	28
<i>Εικόνα 3. Μη επανδρωμένο αεροσκάφος Global Hawk</i>	29
<i>Εικόνα 4. Σχήματα μοναδικού και ζεύγους UAV</i>	34
<i>Εικόνα 5. Χρόνος σύγκλισης με βάση των αριθμό των ζευγών UAV</i>	39
<i>Εικόνα 6. Σήματα ρύθμισης θέσης του quadrotor (UAV)</i>	62
<i>Εικόνα 7. Συνιστώσες απόρριψης αναταράξεων στην θέση του quadrotor (UAV)</i>	63
<i>Εικόνα 8. Παρακολούθηση των προκαθορισμένων συνιστωσών τροχιάς στην θέση του quadrotor (UAV)</i>	63
<i>Εικόνα 9. Μεταβολή του ύψους</i>	73
<i>Εικόνα 10. Μεταβολή της γωνίας</i>	73
<i>Εικόνα 11. Χρόνος απόκρισης συστήματος έπειτα από τις αναταράξεις</i>	74

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος/α: Δανάη Αικατερίνη  
του Μισοβίτση, με αριθμό μητρώου 44560 φοιτητής / τρια του  
Τμήματος Μηχανικών Αυτοματισμού Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την  
εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του  
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και  
πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται  
αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη  
αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα  
του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος  
φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα  
του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η  
Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του  
αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα  
καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός  
ημερολογιακού βμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα  
προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Επίσης δηλώνω υπεύθυνα ότι έχω παρακολουθήσει το σεμινάριο συγγραφής και  
εκπόνησης πτυχιακής εργασίας που διοργανώνεται από το Τμήμα Μηχανικών  
Αυτοματισμού Τ.Ε. κατά το Χειμερινό/Εαρινό Εξάμηνο του Ακ. Έτους 2016.

Ο Δηλών

Ημερομηνία

24/5/2016

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος/α: Καρόλη Ζουραζία  
του Παροχίδου, με αριθμό μητρώου 41691 φοιτητής / τρια του  
Τμήματος Μηχανικών Αυτοματισμού Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την  
εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του  
συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και  
πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται  
αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη  
αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα  
του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος  
φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα  
του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η  
Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του  
αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα  
καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός  
ημερολογιακού βμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα  
προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Επίσης δηλώνω υπεύθυνα ότι έχω παρακολουθήσει το σεμινάριο συγγραφής και  
εκπόνησης πτυχιακής εργασίας που διοργανώνεται από το Τμήμα Μηχανικών  
Αυτοματισμού Τ.Ε. κατά το Χειμερινό/Εαρινό Εξάμηνο του Ακ. Έτους 2016.

Ο Δηλών

Ημερομηνία

24/5/2016

## Εισαγωγή

Στην παρούσα εργασία γίνεται μια μελέτη για τα μη επανδρωμένα αεροσκάφη και την λειτουργία τους. Πιο συγκεκριμένα θα αναλύσουμε την αξιοποίηση αυτών των αεροσκαφών και τους αλγόριθμους όπου χρησιμοποιούνται με σκοπό να βγάλουμε συμπεράσματα για την επιτήρηση μιας δασικής έκτασης και έγκαιρης προειδοποίησης και αντιμετώπισης σε περίπτωση πυρκαγιάς.

Τα δάση παίζουν μεγάλο ρόλο στην ισορροπία της φύσης, αφού βοηθούν στον καθαρισμό του νερού, στην σταθεροποίηση του εδάφους, στην παροχή θρεπτικών συστατικών και γενικότερα στη διατήρηση του ήπιου κλίματος. Δυστυχώς κάθε χρόνο οι δασικές πυρκαγιές προκαλούν δισεκατομμύρια σε υλικές ζημιές και το περιβάλλον κάθε χρόνο. Για την καταπολέμησή τους, η έγκαιρη ανίχνευση και η συνεχή παρακολούθηση είναι ζωτικής σημασίας. Με τη βοήθεια των προηγμένων τεχνικών επεξεργασίας εικόνας, πολλές μέθοδοι έχουν αναπτυχθεί για την ανίχνευση των δασικών πυρκαγιών ακόμα και σε απομακρυσμένες περιοχές με τη χρήση δορυφορικών εικόνων. Οι πυροσβέστες χρειάζονται συχνή και υψηλής ποιότητας πληροφόρηση και ενημέρωση σχετικά με την πρόοδο των πυρκαγιών για την αποτελεσματική αλλά και ασφαλή αντιμετώπιση τους. Επειδή όμως πολλές φορές οι τεχνικές παρακολούθησής τους είναι ανεπαρκείς οι πυροσβέστες συχνά απαιτείται να μπαίνουν μέσα στις περιοχές χωρίς να γνωρίζουν για το πώς και κυρίως το που διαδίδεται η φωτιά, βάζοντας έτσι την ζωή τους σε κίνδυνο. Για τους λόγους αυτούς, υπάρχει η ανάγκη για την ανάπτυξη πιο αποτελεσματικών τεχνολογιών παρακολούθησής των πυρκαγιών.

## Abstract

This written work is about a research for the unmanned aerial vehicles and their function. More specifically we will analyze the utilization of these aircrafts and the algorithms which are used in order to draw conclusions for the surveillance of woodland and early warning and response in case of fire.

Forests play a large role in the balance of nature by helping in the purification of water, in soil stabilization, the nutrient supply and generally maintain the mild climate. Unfortunately, every year forest fires cause billions in damage to property and to the environment every year. To fight them, the early detection and monitoring is vital. Using the help of advanced image processing techniques, many methods have been developed for the detection of forest fires even in remote areas using satellite images. Firefighters need frequent and high-quality information and an update on the progress of the fire for the effective and safe treatment. Unfortunately, there are many times that their monitoring techniques are insufficient and the firefighters often required to go into the areas without knowing where they are actually going and especially the spread of fire, thus they are putting their lives in danger. For these reasons, there is a need to develop more efficient fire monitoring technologies.

## Κεφάλαιο 1: Φυσικό Υπόβαθρο

Όλα τα έντομα που συναντάμε στο φυσικό περιβάλλον, που ζουν σε αποικίες έχει παρατηρηθεί ότι με τα χρόνια αναπτύσσουν οργανωμένες συμπεριφορές. Ξεχωριστά το καθένα παίζει το δικό του ρόλο στην αποικία και φτάνοντας στο σύνολο έχουμε μια υψηλού επιπέδου οργάνωση, η οποία αποτελεί αντικείμενο μελέτης και έρευνας για τον άνθρωπο.

Στην φύση όλες αυτές οι δραστηριότητες, οι οποίες πραγματοποιούνται μεμονωμένα δεν απαιτούν καμία εποπτεία. Έχει μελετηθεί ότι η συμπεριφορά των εντόμων αυτών έγκειται στην συνεργασία που έχουν μεταξύ τους, ενώ σε επίπεδο αποικίας είναι αυτοοργανωόμενη.

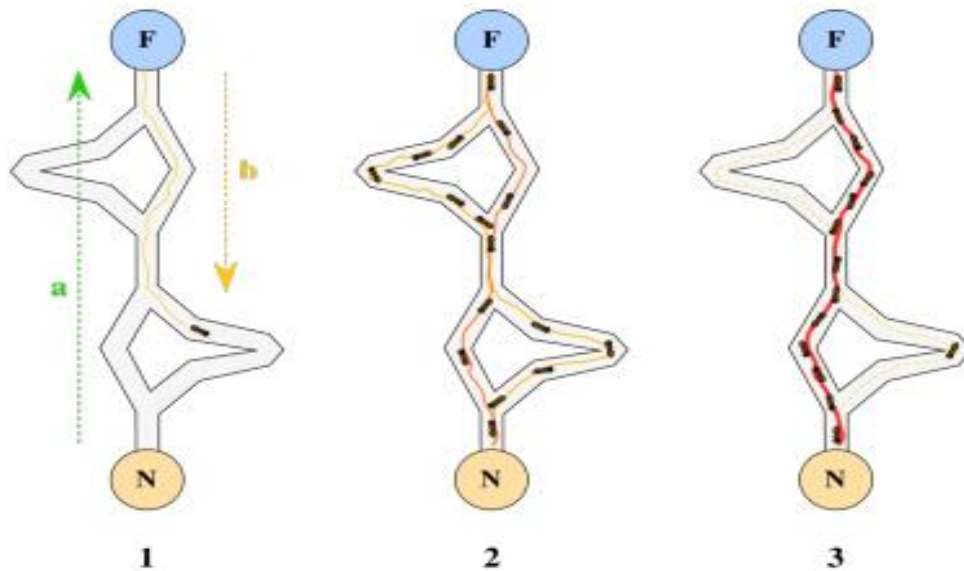
### 1.1. Παράδειγμα συλλογικής ευφυίας στα μυρμήγκια

Η συχνότητα και η ποσότητα των μυρμηγκιών που θα πάνε να φέρουν τροφή στη φωλιά, εξαρτάται από την ποσότητα των «μυρμηγκιών ανιχνευτές» που θα έρθουν σε επαφή με τα υπόλοιπα μυρμήγκια. Δηλαδή αν υπάρχει τροφή, οι «ανιχνευτές» θα βρουν εύκολα την τροφή με αποτέλεσμα να επιστρέψουν γρήγορα και με μεγάλη συχνότητα, δίνοντας το μήνυμα στα άλλα μυρμήγκια να πάνε και αυτά. Ακόμα υπολογίζουν το πόσο ασφαλές είναι εκτός της φωλιάς ανάλογα με τη συχνότητα και τη διάρκεια της ώρας που βγαίνουν έξω οι «ανιχνευτές». Αυτή η επικοινωνία επιτυγχάνεται μέσω των κεραιών.

Ακόμα ένα αξιοσημείωτο παράδειγμα αναφορικά με τα μυρμήγκια είναι το γεγονός ότι ακολουθούν το δρόμο που χάραξε ένα άλλο, ανάμεσα σε πολλές πιθανές διαδρομές ώστε να φτάσουν στη τροφή. Κάθε φορά που τα μυρμήγκια βγαίνουν για να πάρουν τροφή εκκρίνεται μια χημική ουσία που ονομάζεται φερομόνη, η οποία ουσία προσελκύει τα άλλα μυρμήγκια. Με αυτό τον τρόπο δημιουργείται ένα μονοπάτι, που ακολουθούν τα άλλα μυρμήγκια και αυτό έχει ως αποτέλεσμα το να βρίσκουν αυτά τη πιο σύντομη διαδρομή από τη φωλιά τους για να πάρουν

τη τροφή. Η διαδρομή που είναι πιο σύντομη, είναι η πρώτη που έχει διπλό ίχνος φερομόνης και προτιμείται από τα άλλα μυρμήγκια.

Σε σχετικά πειράματα (βλ. Σχήμα 2) δημιουργήθηκαν ανάμεσα στη φωλιά και την τροφή των μυρμηγκιών τέσσερις διαδρομές διαφορετικού μήκους. Μέσα σε μερικά λεπτά η αποικία συνήθως επέλεγε το συντομότερο δρόμο. Τα πρώτα μυρμήγκια που επέστρεφαν στη φωλιά από την τοποθεσία της τροφής ήταν εκείνα που είχαν ακολουθήσει το σύντομο δρόμο τόσο στο πήγαινε, όσο και στο γυρισμό. Επειδή αυτή η διαδρομή ήταν η πρώτη που είχε διπλό ίχνος φερομόνης, τα άλλα μυρμήγκια την προτιμούσαν, με αποτέλεσμα να ενισχύεται όλο και περισσότερο με αποτέλεσμα να γίνει η οδός πρόσβασης προς την τροφή.



Εικόνα 1. 1. Το 1ο μυρμήγκι επιλέγει μια διαδρομή προς την τροφή (F) και επιστρέφει στην φωλιά (N) αφήνοντας ίχνη φερομόνης. 2. Τα άλλα μυρμήγκια ακολουθούν κάποια από τις 4 διαδρομές. 3. Η διαδρομή με την μεγαλύτερη εναπόθεση φερομόνης, που είναι και η συντομότερη, γίνεται προτιμητέα.

Η συλλογική συμπεριφορά που προκύπτει από μια ομάδα κοινωνικών εντόμων ονομάστηκε νοημοσύνη σμήνους. Χαρακτηριστικά άλλα παραδείγματα είναι οργανισμοί όπως



κάποια είδη πτηνών ή ψαριών όπου έχουν την τάση να δημιουργούν σμήνη (κοπάδια) κυρίως για λόγους επιβίωσης. Είναι γεγονός άλλωστε ότι η συσπείρωση των ψαριών σε ένα κοπάδι αυξάνει το βαθμό προστασίας από κάποιο θηρευτή, αν και σε επίπεδο ατόμων ευνοούνται τα άτομα που βρίσκονται στο εσωτερικό της ομάδας ενώ τα άτομα στα άκρα είναι περισσότερο εκτεθειμένα σε κίνδυνο. Συμπεριφορά σμήνους παρουσιάζεται επίσης και σε θηλαστικά για παρόμοιους λόγους επιβίωσης αν και οι αναδυόμενες συμπεριφορές δεν είναι πάντα αμυντικές. Επίσης ένα ακόμη παράδειγμα είναι οι αγέλες των λύκων, μια καθαρά επιθετική περίπτωση συμπεριφοράς σμήνους, η οποία σε επίπεδο παρατήρησης αποσκοπεί στην περικύκλωση-παγίδευση θηράματος από την αγέλη. Συνοψίζοντας τέτοια φαινόμενα μπορούμε να βρούμε σε ποικίλα είδη οργανισμών αν και συνήθως έχουμε στο μυαλό μας την έννοια του σμήνους με την μορφή τεράστιας μάζας εντόμων. Αυτό συμβαίνει για τον λόγο ότι είναι περισσότερο εμφανές η αποτελεσματικότητα της ομάδας σε αντιδιαστολή με τις δυνατότητες και την αποτελεσματικότητα των ατόμων της ομάδας ξεχωριστά.

Έτσι αυτά τα φαινόμενα οδήγησαν πολλούς ερευνητές, να θεμελιώσουν και να χρησιμοποιήσουν τέτοιους βιοεμπνευσμένους (bio-inspired) αλγορίθμους σε προβλήματα βελτιστοποίησης (π.χ. δρομολόγησης σε ασύρματα δίκτυα) και αυτοοργάνωσης (π.χ. σμήνη ρομπότ) με εξαιρετική επιτυχία. Μερικούς απο τους πιο σημαντικούς περιγράφονται σε παρακάτω ενότητα αναλυτικά.

## 1.2. Νοημοσύνη σμήνους (swarm intelligence)

Swarm robotics, ονομάζεται το ρομποτικό σμήνος όπου αποτελεί μια νέα προσέγγιση για το συντονισμό των συστημάτων multirobot που αποτελούνται απο ένα μεγάλο αριθμό απλών φυσικών ρομπότ.

Μια επιθυμητή συλλογική συμπεριφορά προκύπτει από τις αλληλεπιδράσεις μεταξύ των ρομπότ και των αλληλεπιδράσεων των ρομπότ με το περιβάλλον. Αυτή η προσέγγιση προέκυψε

στον τομέα της τεχνητής νοημοσύνης του σμήνους, αφού βιολογικές μελέτες πάνω σε έντομα, μυρμύγκια αλλά και σ' άλλα πεδία στην φύση όπως αναφέρθηκαν δείχνουν να ακολουθούν την συμπεριφορά του σμήνους.

Η νοημοσύνη σμήνους (swarm intelligence) είναι η συλλογική συμπεριφορά μη καταναμημένων, αυτόοργανωμένων φυσικών ή τεχνητών συστημάτων. Η ιδέα εφαρμόζεται στον τομέα της τεχνητής νοημοσύνης. Ο όρος αυτός εισήχθη από τους Gerardo Beni και Jing Wang το 1989. Τα σμήνη παρέχουν επιθυμητή απόδοση, υψηλή αξιοπιστία (ανοχή σε λάθη) αλλά και χαμηλή πολυπλοκότητα σχεδιασμού και κόστους μονάδας σε σχέση με τα κλασσικά ρομποτικά συστήματα.

Η νοημοσύνη του σμήνους είναι μια ιδιότητα των συστημάτων που επιδεικνύουν συλλογικά ευφύη συμπεριφορά. Ένα σύστημα αντιπροσωπεύει μια οντότητα που ανιχνεύει το περιβάλλον προκειμένου να εκτελέσει μια ενέργεια που έχει επιλέξει. Η κάθε οντότητα λειτουργεί αυτόνομα και στο τέλος ως αποτέλεσμα προκύπτει μια συνολική συλλογική συμπεριφορά. Δεν παίρνουν οδηγίες από κανέναν, παίρνουν μόνο αποφάσεις και μέσα από απλούς κανόνες εμφανίζονται διάφορα φαινόμενα (πχ σμήνη πουλιών, κοπάδια ψαριών, άναμμα πυρολαμπίδων, τα μυρμήγκια υπολογίζουν βέλτιστες διαδρομές προς την τροφή τους όπως αναφέρθηκε, μέλισσες να ενημερώνουν τη κυψέλη για νέκταρ κτλ) τα οποία τελικά οδηγούνται μέσα από τη συλλογική συμπεριφορά και ευφυΐα τους.

Τα συστήματα νοημοσύνης σμήνους κατά κανόνα αποτελούνται από έναν πληθυσμό απλών, αυτόνομων πρακτόρων ή διαμεσολαβητών (boids) που αλληλεπιδρούν μεταξύ τους και με το περιβάλλον τους σε τοπικό επίπεδο. Τα συστήματα είναι εμπνευσμένα από τη φύση και ειδικότερα από τα βιολογικά συστήματα. Οι πράκτορες ακολουθούν πολύ απλούς κανόνες, και παρόλο που δεν υπάρχει καμία συγκεντρωτική δομή ελέγχου να υπαγορεύει πως πρέπει να συμπεριφέρονται οι πράκτορες, τοπικές - και έως ένα βαθμό τυχαίες - αλληλεπιδράσεις μεταξύ τέτοιων πρακτόρων οδηγούν στην εμφάνιση μιας ευφυούς, καθολικής συμπεριφοράς, άγνωστης στους αυτόνομους πράκτορες.

### 1.3. Κανόνες του σμήνους σύμφωνα με τον Κρεγκ Ρέινολντς

Κάθε σμήνος ακολουθεί κάποιους κανόνες οι οποίοι καθορίζουν την συμπεριφορά τους. Η ικανότητα οργάνωσης αλλά και η αυτορυθμιζόμενη συμπεριφορά τους είναι κάτι που μέχρι σήμερα παραμένει αντικείμενο μελέτης, αφού εμφανίζεται και εντυπώνεται απο τα πρώτα στάδια της ανάπτυξής τους.

Ο πληροφορικός Κρεγκ Ρεινολντς ήταν ο πρώτος που ξεκίνησε την έρευνα πάνω σε αυτό, και διατύπωσε το 1987 τους 3 βασικούς κανόνες για τη δημιουργία σχηματισμού σμήνους:

- Να υπάρχει μια δύναμη συνάθροισης (clumping), που διατηρεί “ενωμένο” το σμήνος.
- Τα μέλη του σμήνους να είναι σε θέση να ελέγχουν την ταχύτητά τους ως προς τους γείτονες, με αποτέλεσμα το σμήνος να πετάει με κοινή ταχύτητα.
- Τα μέλη του σμήνους να υπόκεινται σε μια δύναμη διαχωρισμού (separation), η οποία αποτρέπει κάθε μέλος του σμήνους από το να πετάει πολύ κοντά με τους γείτονές του.

### 1.4. Αλγόριθμοι Νοημοσύνης Σμήνους

#### 1.4.1. Αλγόριθμος Ant Colony Optimisation (ACO)

Ο αλγόριθμος αυτός επινοήθηκε από τον Marco Dorigo, ο οποίος προσπαθούσε να επιλύσει το πρόβλημα του Περιπλανώμενου Πωλητή (Travelling Salesman Problem). Ήθελε να

βρει την ελάχιστη διαδρομή που θα πραγματοποιούσε ένας πωλητής προκειμένου να επισκεφθεί όλες τις πόλεις από μία φορά. Ο αλγόριθμος ACO είναι εμπνευσμένος από μελέτες που αναφέρονταν στη συμπεριφορά και στην αυτοοργάνωση αποικιών μυρμηγκιών στην προσπάθεια τους να εξασφαλίσουν τροφή.

Είναι ουσιαστικά μια κατηγορία αλγορίθμων βελτιστοποίησης βασισμένη στις ενέργειες μιας αποικίας μυρμηγκιών. Η κάθε λύση για το TSP σχηματίζεται με τη διαδοχική μετάβαση των «ψηφιακών-τεχνικών» μυρμηγκιών από τη μια πόλη στην άλλη με κάποια πιθανότητα. Το «ταξίδι» ολοκληρώνεται όταν το «ψηφιακό-τεχνητό» μυρμήγκι επιστέψει στην αρχική πόλη. Τότε βαθμολογείται η λύση που έχει επιτύχει με κριτήριο το συνολικό μήκος της διαδρομής και προστίθεται η ανάλογη φερομόνη στο μονοπάτι που ακολούθησε. Η παραπάνω διαδικασία επαναλαμβάνεται για κάθε άτομο της αποικίας μέχρι να συμπληρωθεί ο ζητούμενος αριθμός επαναλήψεων.

## Matlab Code

```
% ACO: ant colony optimization for solving the traveling salesperson
```

```
% problem
```

```
% Haupt & Haupt
```

```
% 2003
```

```
clear
```

```
rand('state',11)
```

```
Ncity=30;          % number of cities on tour
```

```
Nants=Ncity;      % number of ants=number of cities
```

```

% city locations

xcity=rand(1,Ncity);ycity=rand(1,Ncity); % cities are
located at (xcity,ycity)

%distance between cities

for ic=1:Ncity

    for id=1:Ncity

dcity(ic,id)=sqrt((xcity(ic)-xcity(id))^2+(ycity(ic)- ycity(id))^2);

    end % id

end % ic

vis=1./dcity;      % visibility equals inverse

                % of distance

phmone=.1*ones(Ncity,Ncity);% initialized pheromones

                % between cities

maxit=600;        % max number of iterations

% a1=0 - closest city is selected

% be=0 - algorithm only works w/ pheromones and not

% distance of city

```

```

% Q - close to the length of the optimal tour

% rr - trail decay

a=2;b=6;rr=0.5;Q=sum(1./(1:8));dbest=9999999;e=5;

% initialize tours

for ic=1:Nants

    tour(ic,:)=randperm(Ncity);

end % ic

tour(:,Ncity+1)=tour(:,1); % tour ends on city it

starts with

for it=1:maxit

% find the city tour for each ant

% st is the current city

% nxt contains the remaining cities to be visited

for ia=1:Nants

    for iq=2:Ncity-1

        [iq tour(ia,:)];
    end
end

```

```
st=tour(ia,iq-1);    nxt=tour(ia,iq:Ncity);    prob=((phmone(st,nxt).^a).*(vis(st,nxt).^b)).  
sum((phmone(st,nxt).^a).*(vis(st,nxt).^b));
```

```
rcity=rand;
```

```
for iz=1:length(prob)
```

```
    if rcity <sum(prob(1:iz))
```

```
        newcity=iq-1+iz;
```

```
to be visited
```

```
        break
```

```
    end % if
```

```
end % iz
```

```
temp=tour(ia,newcity); % puts the new city
```

```
                % selected next in line    tour(ia,newcity)=tour(ia,iq);
```

```
tour(ia,iq)=temp;
```

```
end % iq
```

```
end % ia
```

```
% calculate the length of each tour and pheromone
```

```
distribution
```

```

phtemp=zeros(Ncity,Ncity);

for ic=1:Nants

    dist(ic,1)=0;

    for id=1:Ncity

        dist(ic,1)=dist(ic)+dcity(tour(ic,id),tour(ic,id+1));  phtemp(tour(ic,id),tour(ic,id+1))=Q/dist(ic,1);

    end % id

end % ic

[dmin,ind]=min(dist);

if dmin<dbest

    dbest=dmin;

end % if

% pheromone for elite path

ph1=zeros(Ncity,Ncity);

    for id=1:Ncity

        ph1(tour(ind,id),tour(ind,id+1))=Q/dbest;

    end % id

```



```

% update pheromone trails

phmone=(1-rr)*phmone+phtemp+e*ph1;

dd(it,:)=[dbest dmin];

[it dmin dbest]

end %it

[tour,dist]

figure(1)

plot(xcity(tour(ind,:)),ycity(tour(ind:)),xcity,ycity,'o')

axis square

figure(2);plot([1:maxit],dd(:,1),[1:maxit],dd(:,2),'-')

```

#### 1.4.2. Αλγόριθμος Particle Swarm Optimisation (PSO)

Ο αλγόριθμος PSO αναπτύχθηκε από τους Kennedy και Eberhart για να μοντελοποιήσει τον τρόπο με τον οποίο ένα σμήνος πουλιών ή ψαριών αναζητά τροφή. Ύστερα βελτιστοποιήθηκε από τους Clerc και Kennedy.

Ο συγκεκριμένος αλγόριθμος είναι ένας αλγόριθμος βελτιστοποίησης για την επίλυση προβλημάτων στην οποία η βέλτιστη λύση μπορεί να αναπαραστεί σαν ένα σημείο σε ένα πολυδιάστατο χώρο. Κάθε μεμονωμένο μέλος του καθοδηγείται προς τη βέλτιστη λύση έχοντας ως οδηγό δύο παράγοντες:

- 1) έναν στοχαστικά ζυγισμένο μέσο όρο των προηγούμενων καλύτερων λύσεων του ίδιου και
- 2) έναν στοχαστικά ζυγισμένο μέσο όρο των προηγούμενων καλύτερων λύσεων των υπόλοιπων μελών.

Η διαδικασία στο χώρο τροφοδοτείται με μια αρχική ταχύτητα καθώς αποκαθίστωναται μεταξύ των μελών και ένα κανάλι επικοινωνίας. Η ταχύτητα επιτρέπει στο μέλος του σμήνους να ταξιδεύει στο χώρο λύσης αναζητώντας καλύτερες θέσεις επίλυσης έτσι ώστε να παρουσιάζει καλύτερη απόδοση σε σχέση πάντα με μια κατάλληλα επιλεγμένη συνάρτηση κόστους. Το μεγαλύτερο πλεονέκτημα του PSO είναι ότι μεγάλος αριθμός των μελών καθιστούν την τεχνική ανθεκτική απέναντι στο πρόβλημα των τοπικών ελαχίστων.

Όπως αναφέραμε ο αλγόριθμος αυτός προσομοιώνει τη συμπεριφορά ενός σμήνους πουλιών. Έτσι το λογικό είναι, εάν δεν γνωρίζει κανένα για το που βρίσκεται η τροφή, να ακολουθήσουν εκείνο το οποίο είναι πιο κοντά σε αυτή. Κάθε μοναδική λύση είναι στο παράδειγμα μας ένα πτηνό ως προς τον ανίχνευση χώρο και το αποκαλούμε επιστημονικά σωματίδιο (particle). Όλα τα σωματίδια αξιολογούνται για το αν είναι κατάλληλα απο μια συνάρτηση υπεύθυνη για αυτό (fitness values), την οποία και θέλουμε να βελτιώσουμε κάθε φορά. Έχουν επίσης διανύσματα ταχυτήτων που τα κατευθύνουν στο «πέταγμα» τους. Τα σωματίδια «πετούν» μέσα στο χώρο του προβλήματος προς βελτιστοποίηση και ακολουθούν τα σωματίδια με τις βέλτιστες τιμές καταλληλότητας ως εκείνη τη στιγμή. Οι ταχύτητες των σωματιδίων οι οποίες σε κάθε διάσταση μανδαλώνονται σε μια μέγιστη ταχύτητα  $V_{max}$ . Έτσι αν το άθροισμα των επιταχύνσεων θα προκαλούσε την ταχύτητα σε εκείνη τη διάσταση να υπερβεί την  $V_{max}$  τότε η ταχύτητα σε αυτή τη διάσταση θα περιοριζόταν στη  $V_{max}$ .

Για την αρχικοποίηση του PSO έχουμε ένα σύνολο τυχαίων σωματιδίων και κατόπιν αναζητούμε βέλτιστες λύσεις ενημερώνοντας σε κάθε χρονική στιγμή. Σε κάθε επανάληψη

ενημερώνονται δυο μεταβλητές για κάθε σωματίδιο που ονομάζονται «καλύτερες τιμές» (best values). Η πρώτη μεταβλητή αναφέρεται στην καλύτερη λύση που ανακάλυψε ένα σωματίδιο ως εκείνη τη στιγμή (προσωρινή εμπειρία ή υποκειμενικός παράγοντας) και ονομάζεται pbest και η δεύτερη μεταβλητή αναφέρεται στην γειτονιά με την καλύτερη λύση (αντικειμενικός παράγοντας) και ονομάζεται gbest (global best).

Σε κάθε επανάληψη του αλγορίθμου η ταχύτητα και η θέση κάθε σωματιδίου ενημερώνονται σύμφωνα με τις ακόλουθες εξισώσεις:

$$v = v + c1 \cdot \text{rand}() \cdot (\text{pbest} - \text{present}) + c2 \cdot \text{rand}() \cdot (\text{gbest} - \text{present}) \quad (1)$$

$$\text{present} = \text{present} + v \quad (2)$$

**v:** η ταχύτητα των σωματιδίων

**present:** η τρέχουσα θέση

**pbest, gbest:** όπως αναφέρθηκε

**rand:** γεννήτρια τυχαίων αριθμών που  $\in (0, 1)$

**c1,c2:** συντελεστές βαρών

Ο αλγόριθμος PSO συνοψίζεται στον ακόλουθο ψευδοκώδικα:

∀ (σωματίδιο)

Αρχικοποίησε σωματίδιο

END

DO

∀ (σωματίδιο)

Υπολόγισε τιμή καταλληλότητας

Αν η τιμή καταλληλότητας είναι καλύτερη από pbest

Θέσε τρεχουσα τιμή καταλληλότητας ως pbest

END

Διάλεξε το σωματίδιο με το καλύτερο gbest

∀ (σωματίδιο)

Υπολόγισε ταχύτητα σωματιδίου σύμφωνα με την (1)

Υπολόγισε θέση σωματιδίου σύμφωνα με την (2)

END

While μέγιστος αριθμός επαναλήψεων ή επαλήθευση κριτηρίων σύγκλισης

Τα βασικά βήματα μιας τεχνικής PSO είναι τα εξής:

- **Αρχικοποίηση (initialization):** Θέτουμε τον μετρητή χρόνου στο  $t=0$ , δημιουργούμε κάποια  $n$  τυχαία σωματίδια και δημιουργούμε και αντίστοιχες τυχαίες ταχύτητες για τα σωματίδια. Γίνεται αποτίμηση για κάθε σωματίδιο στον αρχικό πληθυσμό με τη χρήση της

αντικειμενικής συνάρτησης  $J$ . Ψάχνουμε την καλύτερη αντικειμενική τιμή της συνάρτησης  $J_{best}$ . Θέτουμε το σωματίδιο που σχετίζεται με αυτή τη τιμή σαν  $Global\ best$ . Στο τέλος θέτουμε τις αρχικές τιμές των παραμέτρων βαρύτητας  $w_{min}$ ,  $w_{max}$ .

- **Ενημέρωση χρόνου (time updating):** Ενημέρωση του time counter  $t=t+1$
- **Ενημέρωση των παραμέτρων βάρους(weight updating):** Ενημέρωση του  $w(t)=w_{max}-((w_{max}-w_{min})x_{lter}/lter_{max})$
- **Ενημέρωση ταχύτητας (velocity updating):** Χρησιμοποιώντας τα  $global\ best$  και  $individual\ best$  η ταχύτητα του  $j$ th σωματιδίου ενημερώνεται σύμφωνα με την εξής εξίσωση:  $v_{j,k}=w(t)v_{j,k(t-1)}+c_1r_1(x_{j,k(t-1)}-x_{j,k(t-1)})+c_2r_2(x_{best,k(t-1)}-x_{j,k(t-1)})$  όπου  $c_1, c_2$  είναι θετικές σταθερές και  $r_1, r_2$  ομοιόμορφα καταναμημένες τιμές στο διάστημα  $[0,1]$ . Επίσης ελέγχουμε και τα όρια της ταχύτητας γιατί εάν υπερβαίνει αυτά τα όρια την θέτουμε στο πλησιέστερο όριο (άνω ή κάτω αντίστοιχα)
- **Ενημέρωση θέσης (position updating):** Κάθε σωματίδιο ανάλογα με την ταχύτητα που έχει (η οποία μεταβάλλεται) αλλάζει την θέση του σύμφωνα με την παρακάτω εξίσωση  $x_{j,k}(t)=v_{j,k}(t)+x_{j,k}(t-1)$ . Το αποτέλεσμα που λαμβάνουμε στο τέλος μπορεί να μην είναι πάντοτε αυτό που θα θέλαμε λόγω της μη ικανοποίησης των περιορισμών της ταχύτητας. Έτσι η παραπάνω εξίσωση μπορεί να διαμορφωθεί ως εξής:

$$V_{j,k}(t) = v_{j,k}(t) + x_{j,k}(t-1)$$

$$\text{If } x_{j,k} \min \leq v_{j,k}(t) + x_{j,k}(t-1) \leq x_{j,k} \max$$

$$X_{j,k}(t) = \begin{cases} x_{j,k} \min & \text{if } v_{j,k}(t) + x_{j,k}(t-1) < x_{j,k} \min \\ v_{j,k}(t) + x_{j,k}(t-1) & \text{if } x_{j,k} \min \leq v_{j,k}(t) + x_{j,k}(t-1) \leq x_{j,k} \max \\ x_{j,k} \max & \text{if } v_{j,k}(t) + x_{j,k}(t-1) > x_{j,k} \max \end{cases}$$

$$x_{j,k} \max \text{ if } v_{j,k}(t) + x_{j,k}(t-1) > x_{j,k} \max$$

- **Ενημέρωση τοπικού βέλτιστου (individual best updating):** Κάθε σωματίδιο αποτιμάται σύμφωνα με την ενημερωμένη θέση του: If  $J_j > J_j^*$ ,  $j=1,2,\dots,n$  {then update individual best as  $X_j^*(t)=X_j(t)$  and  $J_j^*= J_j$  and goto step 7} else goto step 7.
- **Ενημέρωση ολικού βέλτιστου (Global best updating):** Αναζήτηση για τη μέγιστη τιμή  $J_{max}$  μέσα στο  $J_j^*$  όπου  $J_{max}$  είναι η θέση του σωματιδίου με τη μέγιστη τιμή της αντικειμενικής συνάρτησης.
- **Συνθήκη τέλους (stopping criteria):** Εάν η συνθήκη τέλους ικανοποιείται τότε τελειώσει το πρόγραμμα αλλιώς πήγαινε στο 2ο βήμα.

## Matlab Code

```
% Particle Swarm Optimization - PSO

% Haupt & Haupt

% 2003

clear

ff = 'testfunction'; % Objective Function

% Initializing variables

popsize = 10;           % Size of the swarm

npar = 2;              % Dimension of the problem

maxit = 100;          % Maximum number of iterations
```

```

c1 = 1;          % cognitive parameter

c2 = 4-c1;      % social parameter

C=1;           % constriction factor

% Initializing swarm and velocities

par=rand(popsize,npar); % random population of

                                % continuous values

vel = rand(popsize,npar); % random velocities

% Evaluate initial population

cost=feval(ff,par); % calculates population cost using

                                % ff

minc(1)=min(cost); % min cost

meanc(1)=mean(cost); % mean cost

globalmin=minc(1); % initialize global minimum

% Initialize local minimum for each particle

localpar = par; % location of local minima

localcost = cost; % cost of local minima

% Finding best particle in initial population

```

```

[globalcost,indx] = min(cost);

globalpar=par(indx,:);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Start iterations

iter = 0; % counter

while iter < maxit

    iter = iter + 1;

    % update velocity = vel

    w=(maxit-iter)/maxit;          % inertia weiindxht

    r1 = rand(popsize,npar);      % random numbers

    r2 = rand(popsize,npar);      % random numbers

    vel = C*(w*vel + c1 *r1.*(localpar-par) + c2*r2.*(ones(popsize,1)*globalpar-par)); % update
    particle positions par = par + vel;

    % update particle positions par = par + vel; % updates particle position overlmit=par<=1;

    underlimit=par>=0;

    par=par.*overlimit+not(overlimit);

    par=par.*underlimit;

```



```

% Evaluate the new swarm

    cost = feval(ff,par);          % evaluates cost of swarm

% Updating the best local position for each particle

    bettercost = cost < localcost;

    localcost = localcost.*not(bettercost) + cost.*bettercost; localpar(find(bettercost),:) =
par(find(bettercost),:);

% Updating index g

    [temp, t] = min(localcost);

if temp<globalcost

    globalpar=par(t,:); indx=t; globalcost=temp;

end

    [iter globalpar globalcost] % print output each

                                % iteration

minc(iter+1)=min(cost);          % min for this

                                % iteration    globalmin(iter+1)=globalcost;
% best min so far meanc(iter+1)=mean(cost);    % avg. cost for

                                % this iteration

end% while

```

figure(24)

```
iters=0:length(minc)-1;
```

```
plot(iters, minc, iters, meanc, '- ', iters, globalmin, ':');
```

```
text(0, minc(1), 'best'); text(1, minc(2), 'population
```

```
average')
```

```
xlabel('generation'); ylabel('cost');
```

## Κεφάλαιο 2: Ρομποτικό Σμήνος – Μη Επανδρωμένα αεροσκάφη UAV (Unmanned Aircraft Vehicles)

### 2.1. Ιστορικό Υπόβαθρο

Η ιδέα των αυτόνομων αεροσκαφών ξεκίνησε περίπου απο τότε που δημιουργήθηκαν τα αεροπλάνα. Κατά την διάρκεια του 2ου παγκοσμίου πολέμου οι Η.Π.Α έφτιαξαν τα πρώτα drones. Ο στόχος των UAV ήταν η κατασκοπεία των αντιπάλων κατά τη διάρκεια του πολέμου στο Βιετνάμ. Η ανάπτυξη όμως διαφόρων τύπων drone ξεκίνησαν περίπου το 1990-2000, όταν ξεκίνησε δηλαδή η εμπλοκή των Η.Π.Α στους πολέμους της Μέσης Ανατολής, και αρχικά επικεντρώθηκαν στην παροχή πρακτικής στόχους στην στρατιωτική εκπαίδευση. Τα συγκεκριμένα αεροσκάφη έδιναν την δυνατότητα της οπτικής ενός πτηνού στους στρατιώτες στο πεδίο της μάχης. Μπορούσαν να πετάνε για αρκετή ώρα και να αποστέλλουν φωτογραφικό υλικό πολύ καλής ανάλυσης στους εκάστοτε φορείς, οι οποίοι μπορεί να βρισκόντουσαν μίλια μακριά. Το μεγάλο προσόν τους ήταν ότι εάν υπήρχε πιθανή σύγκρουση δεν θα υπήρχαν απώλειες πιλοτών.

Η ανάπτυξη των UAV συνεχίστηκε κατά την διάρκεια του πρώτου παγκοσμίου πολέμου όταν η Dayton-Wright Airplane Company έχει εφεύρει μια μη επανδρωμένη τορπίλη η οποία θα έπεφτε για να εκραγεί σε ένα προκαθορισμένο χρονικό διάστημα.

### 2.2. Κατηγορίες UAV

Τα μη επανδρωμένα αεροσκάφη (UAV) γνωστά και ως drone είναι αεροσκάφη τα οποία πετούν χωρίς πιλότο και ποικίλουν σε μέγεθος και σχήμα. Η δυνατότητα πτήσης τους μπορεί να ελέγχεται είτε αυτόνομα απο υπολογιστές επι του αεροσκάφους είτε απο τηλεχειριστήριο απο κάποιο πιλότο που βρίσκεται στο έδαφος ή και σε κάποιο όχημα(Ground Control Station-GCS).

Έτσι τα χωρίζουμε σε 2 κατηγορίες:

- Τα αυτόνομα ή προγραμματισμένα αεροσκάφη(Autonomous or Pre-programmed)
- Τα οδηγούμενα απο ένα απομακρυσμένο χειριστή(RPV-Remotely Piloted Vehicle)

Στο παρελθόν ειδικότερα η χρήση αυτών των αεροσκαφών (RPV) ήταν για στρατιωτικά θέματα, όπως το πολύ γνωστό αεροσκάφος Predator, το οποίο άνηκε στην αμερικάνικη πολεμική αεροπορία και χρησιμοποιήθηκε κατα τη διάρκεια του πολέμου. Έμοιαζε με μικρό αεροπλάνο που είχε μεγάλα φτερά. Είχε δυνατότητα πτήσης 24 ώρες, μέχρι και 400 μίλια απόσταση σε ευθεία γραμμή απο τον πλησιέστερο σταθμό εδάφους και σε υψόμετρο 15.000-25.000 ποδιών. Επιπλέον μπορούσε να βιντεοσκοπήσει με καλή ποιότητα απο πολύ μεγάλο υψόμετρο, καθώς και με τη βοήθεια υπέρυθρων κάμερων μπορούσε να έχει λήψεις κατά την διάρκεια της νύχτας ακόμη δε και μέσα απο καπνό ή σύννεφο.



*Εικόνα 2. Μη επανδρωμένο αεροσκάφος Predator*

Τα εντελώς αυτόνομα UAV απο την άλλη παρέχουν έναν αυτόματο ελεγκτή όπου ελέγχει όλη την πτήση και διορθώνει πιθανές αστοχίες των υπόλοιπων επανδρωμένων αεροσκαφών απο το πλάνο πτήσης, το οποίο βρίσκεται στην κεντρική του μνήμη. Το πιο χαρακτηριστικό παράδειγμα είναι το αεροσκάφος Global Hawk όπου σχεδιάστηκε από τον Ryan Aeronautical. Χρησιμοποιήθηκε και αυτο απο τις Η.Π.Α και μπορούσε να λειτουργεί συνεχόμενα για 35 ώρες έχοντας συγκεκριμένο πλάνο πτήσης στην μνήμη του, και με μέγιστο ύψος να φτάνει τα 35000 πόδια.



*Εικόνα 3. Μη επανδρωμένο αεροσκάφος Global Hawk*

### 2.3. Fire Drones

Παραδοσιακά, η εξαγωγή πληροφοριών για την υποστήριξη πυρόσβεσης έχει γίνει από ειδικούς, άμεσα με το έδαφος ή την ανάλυση των δεδομένων που παρέχονται από τους πύργους, τους δορυφόρους ή άλλα μέσα. Λίγο πιο συγκεκριμένα τα UAVs έχουν

- i. Διάφορα frames και αισθητήρια συμπεριλαμβανομένων το παγκόσμια γνωστό σύστημα εντοπισμού θέσης (GPS), αδρανειακές μονάδες μέτρησης (IMUs), κάμερες
- ii. Ειδικούς αλγόριθμους και στρατηγικές για την παρακολούθηση της φωτιάς, την ανίχνευση της και την αντιμετώπισή της
- iii. Τα συστήματα GNC για ένα αλλά και για πολλά UAV
- iv. Συνεργασίες εντοπισμού, συστήματα εγκατάστασης, και ελέγχου για τους στόλους UAVs για να καλύφθούν άριστα περιοχές με φωτιά (αυτά τα συστήματα σε πληροφορίες πραγματικού χρόνου με τη βοήθεια οπτικής ειδικών (για τη διάρκεια της μέρας) και υπέρυθρων (τόσο για τη διάρκεια της νύχτας αλλά και ημέρας))
- v. Έναν ειδικό σταθμό που περιλαμβάνει εξοπλισμό για την επικοινωνία, υπολογισμό του εδάφους, οπτικοποίηση της πυρανίχνευσης, παρακολούθηση και πρόβλεψη με αυτόματο συναγερμό πυρκαγιάς καθώς και όλο τον απαραίτητο εξοπλισμό για την ασφαλή και αποτελεσματική λειτουργία των UAVs.

Πιο πρόσφατα, χρησιμοποιούνται τα αερομεταφερόμενα συστήματα για να έχουν μια ευρεία επισκόπηση της εξέλιξης πυρκαγιάς, αλλά και οι δραστηριότητες παρακολούθησης που πραγματοποιούνται από τους ανθρώπους. Δορυφορικά συστήματα έχουν επίσης προταθεί για την ανίχνευση των δασικών πυρκαγιών (San Miguel Ayanz et al., 2005; Kelhä et al., 2003) και την παρακολούθηση (Chuvieco και Martin, 1994. Gonzalo, 1998). Οι χρονικές και χωρικές αναλύσεις αυτών των συστημάτων είναι ακόμη πολύ χαμηλές για τις ανάγκες της καταπολέμησης των δασικών πυρκαγιών σε πολλές περιπτώσεις. Υπάρχουν κάποιες τεχνικές

ανάλυσης της φωτιάς με βάση τη χρήση του υπολογιστή για τις πυρκαγιές που πραγματοποιούνται σε εργαστήρια, όπως (Martínez de Dios κ.ά. 2006.. Pastora et al, 2006).

Η χρήση των UAVs σε δασικές πυρκαγιές σε σενάρια μάχης με την φωτιά έχει αναλυθεί στο Fire Project στις Ηνωμένες Πολιτείες. Η ALTUS UAV, αποτελεί μια εξέλιξη του UAV Predator, και έχει αποδειχθεί αποτελεσματικό μέσα από πειράματα σε περιπτώσεις φωτιάς στο έργο FIRE (Ambrosia, 2002). Τα δεδομένα που λαμβάνονται στο σταθμό εδάφους γεωαναφοράς εικόνων για τη φωτιά, αποστέλλονται μέσω δορυφορικών κανάλιων uplink / downlink.

Παρόλα αυτά οι εφαρμογές αυτές δεν έχουν χρησιμοποιηθεί ακόμα στην χώρα μας πέρα από πειραματικές πτήσεις που έχουν λάβει χώρα με τη βοήθεια γειτονικών μας χωρών.

#### 2.4. Παραμετρική Παρακολούθηση Φωτιάς για ένα UAV

Η παρακολούθηση των πυρκαγιών βασίζονται στην ικανότητα καθενός ξεχωριστού UAV για να παρακολουθεί την περίμετρο της πυρκαγιάς ανεξάρτητα. Σε αυτή την ενότητα θα περιγράψουμε εν συντομία ένα ισχυρό αλγόριθμο ανίχνευσης πυρκαγιάς που έχει εφαρμοστεί σε προσομοίωση.

Υποθέτουμε ότι κάθε UAV είναι εξοπλισμένο με μια υπέρυθρη κάμερα, κλίση αναρτήρα και έναν αυτόματο πιλότο. Η προσέγγισή μας στη περίμετρο εντοπισμού της φωτιάς μπορεί να συνοψιστεί σε έξι βήματα που εκτελούνται στο ρυθμό καρτέ της κάμερας επί του σκάφους με την χρήση υπέρυθρων.

1. Σάρωση μέσω της υπέρυθρης εικόνας, επισήμανση κάθε pixel ως **α. Καμμένο β. Άκαυτο**

2. Χρησιμοποιήστε ένα γραμμικό ταξινομητή για το τμήμα της εικόνας σε δύο διαμερίσματα, με τα καμμένα στοιχεία στο ένα διαμέρισμα και τα άκαυτα στο άλλο.
3. Σχεδιάστε τη γραμμική τμηματοποίηση από το πλαίσιο λήψης της κάμερας στην παγκόσμιες συντεταγμένες του UAV.
4. Κατασκευάστε το σύνολο των προσβάσιμων σημεία T δευτερολέπτων στο μέλλον και να παραμετροποιήστε αυτό το σύνολο, ως συνάρτηση της γωνίας περιστροφής.
5. Δώστε εντολή στην γωνία περιστροφής που αντιστοιχεί στο σημείο του προσβάσιμου συνόλου, ότι είναι μια απόσταση  $d$  (σε παγκόσμιες συντεταγμένες) επί την άκαυτη πλευρά (διαμέρισμα) στην περίμετρο της φωτιάς.
6. Δώστε την εντολή και τις γωνίες κλίσης του αντίζυγου της κάμερας, ώστε η γραμμή του κατακερματισμού του γραμμικού ταξινομητή να διαιρεί την εικόνα σε δύο ίσα μέρη.

Η προσέγγιση αυτή είναι κατάλληλη για την παρακολούθηση της περιμέτρου της πυρκαγιάς διότι συνήθως εξελίσσεται με ένα τρόπο χωρίς συνέχεια. Η πυρκαγιά θα περάσει πάνω από δρόμους και ρέματα και θα φτάσει ακόμα και σε πετρώδες έδαφος. Επιπλέον, η φωτιά εξαπλώνεται πιο γρήγορα σε ανηφορικό έδαφος παρά σε κατηφορικό. Η γραμμική ταξινόμηση χρησιμοποιείται στο βήμα 2 και λειαιίνει αποτελεσματικά στα μη συνεχόμενα όρια. Η απόσταση  $d$  στο βήμα 5 διασφαλίζει ότι τα UAV δεν πετούν απευθείας πάνω από τη φωτιά, αλλά πετούν σε μια ασφαλή απόσταση  $d$  με ελιγμούς. Τέλος, ο μηχανισμός αντιζύγου στο βήμα 6 εξασφαλίζει τη συνεχή αποτελεσματικότητα της γραμμικής ταξινόμησης στο βήμα 2.



## 2.5. Συνεργασία Ομαδικού Εντοπισμού

Για τους πυροσβέστες στην περίμετρο της πυρκαγιάς, οι συχνές ενημερώσεις σχετικά με την πρόοδο της πυρκαγιάς είναι πολύ σημαντικές για την διατήρηση της ασφάλειας. Στην ενότητα αυτή ανατύσσεται ένα καταναμημένο σύστημα παρακολούθησης που επιτρέπει τις πληροφορίες περιμέτρου να διαβιβάζονται όσο πιο συχνά γίνεται στο σταθμό βάσης.

Υποθέτουμε ότι όλα τα UAV πετούν με πανομοιότυπες σταθερές ταχύτητες και ότι η περίμετρος της φωτιάς είναι κυκλική. Ωστόσο ο αλγόριθμος που θα παρουσιαστεί θα λειτουργεί σωστά εάν η περίμετρος είναι γραμμική και τα UAV έχουν εντολή να αντιστρέψουν την κατεύθυνση, όταν το τέλος της περιμέτρου είναι να επιτευχθεί.

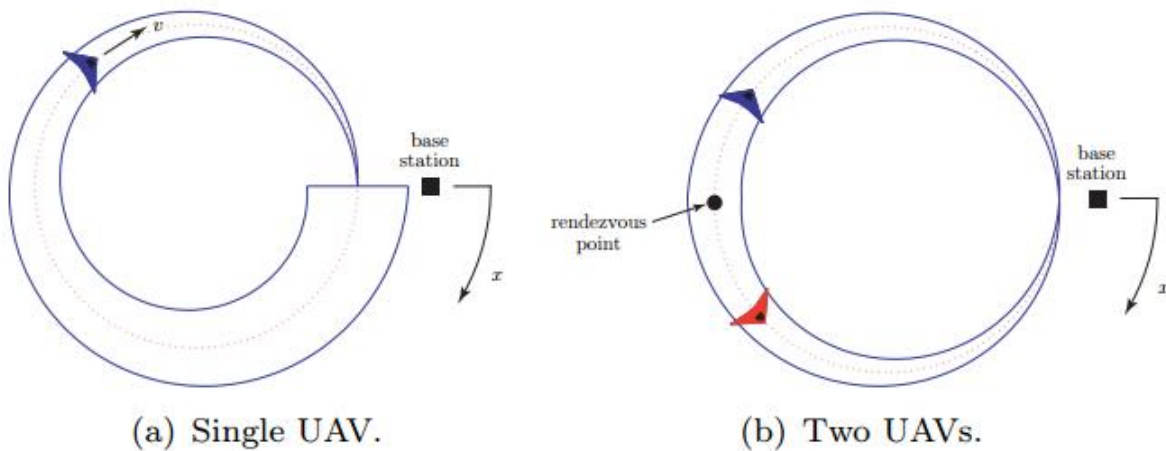
### 2.5.1. Ελαχιστοποίηση Λανθάνοντα χρόνου

Όταν ένα UAV μεταδίδει πληροφορίες προς το σταθμό βάσης, μια συνδεδεμένη λανθάνουσα κατατομή (σε όρους χρόνου-εντυπωμένων εικόνων) συνοδεύει τα δεδομένα. Αφήνουμε τη λανθάνουσα κατάσταση να σχετίζεται με το σημείο  $x$  κατά το χρόνο της ενημέρωσης του σταθμού βάσης, να συμβολίζεται με  $\rho(x)$ . Σημειωνουμε ότι  $\delta(x, t) = \rho(x)$  όταν ένα UAV ενημερώνει το σταθμό βάσης και  $\delta(x, t) = \rho(x) + (t - t_{update})$  μεταξύ των ενημερώσεων. Ο στόχος είναι να σχεδιάσει ένα συνεργατικό σύστημα παρακολούθησης που να ελαχιστοποιεί το  $\rho(x)$  για κάθε  $x$  και να ενημερώνει το σταθμό βάσης όσο το δυνατόν συχνότερα.

Το παρακάτω σχήμα δείχνει τον λανθάνοντα χρόνο που σχετίζεται με την περίμετρο της φωτιάς όταν ένα δεξιόστροφο UAV φτάνει πίσω στο σταθμό βάσης μετά από μια πλήρη περιοδεία της περιμέτρου, όπου το πάχος της διαδρομής δηλώνει την καθυστέρηση της ενημέρωσης του σταθμού βάσης του εν λόγω σημείου. Δεδομένου ότι η κατάσταση της φωτιάς μεταδίδεται μόνο αφού το UAV έχει διαγράψει ολόκληρη την περίμετρο της πυρκαγιάς, η μεγαλύτερη λανθάνουσα κατάσταση συνδέεται με τα δεδομένα που συγκεντρώθηκαν κατά την έναρξη της πτήσης κοντά στο σταθμό βάσης. Επειδή το UAV ταξιδεύει με σταθερή ταχύτητα, το προφίλ λανθάνοντα χρόνου είναι μία γραμμική συνάρτηση της αποστάσεως που διανύθηκε:  $\rho(x) = (P - x) / v$ , όπου  $v$

είναι η ταχύτητα του UAV και  $P$  είναι η περίμετρος. Ο σταθμός βάσης λαμβάνει ενημερώσεις μόνο τόσο γρήγορα όσο τα UAV μπορούν να διασχίζουν ολόκληρη την περίμετρο της φωτιάς. Η συνολική λανθάνουσα κατάσταση συνδέεται με μία διάσχιση όπου δίνεται από τον τύπο  $\mathbf{OSP} \rho(\chi)$   $d(\chi) = 0.5 P^2/v$ .

Αφήνουμε ένα UAV να ανατεθεί να ερευνησει το άνω ήμισυ της περιμέτρου ενώ ένα δεύτερο εκχωρείται στο κατώτερο ήμισυ. Αν τα UAVs αναχωρούν από το σταθμό βάσης ταυτόχρονα και πετάνε με την ίδια ταχύτητα, ο ρυθμός ενημέρωσης θα είναι η ίδιος με την μεμονωμένη περίπτωση UAV (αφού και οι δύο UAVs φτάνουν πίσω στο σταθμό βάσης την ίδια στιγμή), αλλά η καθυστέρηση που σχετίζεται με τις πληροφορίες και στις δύο πλευρές του σταθμού βάσης θα είναι συμμετρική και μειωμένη, όπως μπορεί να φανεί στο παρακάτω σχήμα.



Εικόνα 4. Σχήματα μοναδικού και ζεύγους UAV

(α). Για ένα ενιαίο UAV, όπου παρακολουθεί μια στατική κυκλικής περιμέτρου φωτιάς. Το πάχος της διαδρομής δηλώνει τον λανθάνοντα χρόνο των πληροφοριών σε εκείνο το σημείο, όταν μεταδίδεται στο σταθμό βάσης.

(β). Ένα ζεύγος UAVs όπου παρακολουθούν μια στατική κυκλική πυρκαγιά σε αντίθετες κατευθύνσεις.

Εδώ χρησιμοποιούνται οι τύποι:

$$\rho(x) = \begin{cases} \frac{x}{v} & \text{for } 0 \leq x \leq P/2 \\ \frac{P-x}{v} & \text{for } P/2 < x \leq P \end{cases}$$

Και η συνολική λανθάνουσα κατάσταση που συνδέεται με το σύστημα είναι η μισή από την περίπτωση χρήσης ενός UAV ( $\int_0^P \rho(x) dx = 0.25 P^2/v$ ).

Για τα UAVs που ακολουθούν την περίμετρο και πετούν σε σταθερή ταχύτητα, ο λανθάνων χρόνος του προφίλ που φαίνεται στο παραπάνω σχήμα ( (β).περίπτωση ) είναι η ελάχιστη δυνατή λανθάνουσα για κάθε x κατά μήκος της περιμέτρου της φωτιάς. Για να δείτε ότι αυτό είναι αλήθεια, σημειώστε ότι η ελάχιστη λανθάνουσα κατάσταση που σχετίζεται με στοιχεία που συγκεντρώθηκαν στο x στην περίμετρο είναι ο χρόνος που χρειάζεται για να ταξιδέψει από το x προς το σταθμό βάσης κατά μήκος της συντομότερης διαδρομής. Διαιρώντας την περίμετρο ίσα μεταξύ των δύο UAVs εξασφαλίζει ότι η απόσταση που διανύθηκε μεταξύ οποιουδήποτε σημείου στην περίμετρο και ο σταθμός βάσης είναι ελάχιστη. Η συνέπεια είναι ότι προσθέτοντας περισσότερα από δύο UAVs δεν θα βελτιωθεί η λανθάνουσα  $\rho(x)$  για κανένα συγκεκριμένο σημείο της περιμέτρου. Ωστόσο, ο ρυθμός με τον οποίο συμβαίνουν οι ενημερώσεις θα αυξηθεί γραμμικά με τον αριθμό των ζευγών των UAV που χρησιμοποιούνται και η λανθάνουσα κατάσταση θα μειωθεί αποτελεσματικά. Για να διατηρηθεί η ελάχιστη λανθάνουσα κατάσταση του σχήματος (β) περίπτωση και να μεγιστοποιηθεί η συχνότητα των ενημερώσεων στο σταθμό βάσης, τα UAVs θα πρέπει να διανεμούνται εξ ίσου γύρω από την περίμετρο, με κάθε UAV να του ανατεθεί να παρακολουθεί ένα τμήμα του μήκους  $P / N$  όπου N είναι ο αριθμός των UAVs.

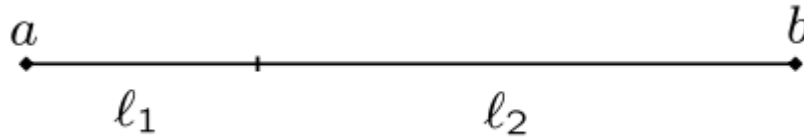
### 2.5.2. Αλγόριθμος για παρακολούθηση κατανεμημένης φωτιάς

Για ένα σταθερό μήκος της περιμέτρου και ένα σταθερό αριθμό των επανδρωμένων αεροσκαφών, η ελάχιστη λανθάνουσα διαμόρφωση είναι όταν ζευγάρια των UAVs έχουν κατανεμηθεί ομοιόμορφα κατά μήκος της περιμέτρου της φωτιάς και από τις 2 πλευρές της (δηλαδή για κάθε ζεύγος ο ένας αρχηγός μαζί με την ομάδα του να είναι δεξιά και ο άλλος μαζί με την δική του αριστερά). Τα ζεύγη των UAVs θα συναντηθούν για να μεταδώσουν τις συλλεγόμενες πληροφορίες τους. Ύστερα το καθένα θα αντιστρέψει την κατεύθυνση του για να συναντήσει το “γείτονα” του στην άλλη κατεύθυνση. Για την διευκόλυνση του ανεφοδιασμού, τα UAVs μπορούν να ανταλλάζουν ρόλους σε κάθε συνάντησή τους έτσι ώστε τα μέλη της ομάδας με τα λιγότερα καύσιμα να είναι πιο κοντά προς τον σταθμό βάσης.

Ο στόχος του αλγορίθμου είναι πολύ απλά το να συγκλίνει σε αυτό η ελάχιστη ορατή διαμόρφωση. Ο αλγόριθμος πρέπει να συγκλιθεί για κάθε μέγεθος της περιμέτρου και πρέπει να προσαρμόζεται κάθε φορά όταν το μήκος της περιμέτρου ή ο αριθμός των ζευγαριών των UAVs αλλάξει. Στον αλγόριθμο οι αλλαγές στις παραμέτρους του συστήματος διαδίδονται σ’όλη την ομάδα με μεγάλη ταχύτητα, και πρέπει να είμαστε σε θέση να αντιμετωπίσουμε την εισαγωγή και την διαγραφή των UAVs όπως και την συστολή και διαστολή της κάθε περιμέτρου.

Η βασική ιδέα είναι για κάθε UAV να προβεί σε ενέργειες που θα επιτρέψουν “γειτονικά” UAV να μοιραστούν την περίμετρο μεταξύ τους. Όταν 2 UAV συναντηθούν, το κάθε ένα έχει γνώση του μήκους που έχει διασχίσει από την τελευταία συνάντησή του. Το άθροισμα αυτών των μηκών μπορεί να διαιρεθεί σε ίσα μέρη μεταξύ τους. Εκείνο με το μικρότερο μήκος ( που έχει δηλαδή ταξιδέψει λιγότερο από τα άλλα) θα “χασομερήσει” με το μεσαίο σημείο αυτού του τμήματος, για να περιμένει το “γείτονα” του για την επόμενη φορά που θα ξανασυναντηθούν.

Για να κάνουμε την ιδέα πιο εμφανή, έστω ότι έχουμε ένα απλό ευθύγραμμο τμήμα AB με 2 UAVs να συγκεντρώνουν μαζί πληροφορίες όπως δείχνουμε παρακάτω:



Έστω το  $l_1$  να είναι η απόσταση που έχει γίνει από το 1ο UAV. Παρατηρούμε ότι έχει ταξιδέψει μικρότερη απόσταση, οπότε μετά την επιστροφή στο σημείο  $a$  θα έχει ταξιδέψει  $(l_1+l_2)/2$  και μετά θα αρχίσει να καθυστερεί (“χασομερεί” όπως αναφέραμε και πιο πάνω). Το 2ο UAV θα επιστρέψει στο σημείο  $b$  και ύστερα θα αντιστρέψει την κατεύθυνση του μέχρι να συναντήσει το UAV1. Έτσι αφού το UAV1 ταξίδεψε την μικρότερη απόσταση, θα φτάσει  $l_1$  στο μέσο του  $ab$ , και όταν φτάσει και το UAV2, θα έχουν κάνει την ίδια απόσταση ( $l_1=l_2$ ), από τα σημεία  $a, b$  που μας οδηγεί στο συμπέρασμα ότι τα ζεύγη των UAVs έχουν πετύχει την ελάχιστη λανθάνουσα διαμόρφωση.

Οποιαδήποτε αλλαγή υπάρξει στο μέγεθος του τμήματος πρέπει να παρακολουθείτε αποτελεσματικά από τα ζεύγη των UAVs, μέχρι κάθε ζεύγος να μετρήσει το τρέχον μήκος της περιμέτρου, αθροίζοντας τις αποστάσεις που έχουν πραγματοποιηθεί από τα τελικά σημεία. Με άλλα λόγια εφόσον τα UAVs έχουν στην μνήμη τους την κατάσταση της περιμέτρου από προηγούμενη επανάληψη η συνεχής εξισορρόπηση του αλγορίθμου θα εντοπίζει τις τετελεσμένες αλλαγές στην περίμετρο. Για να ενεργοποιηθούν πληροφορίες σχετικά με την αύξηση της περιμέτρου και να καταχωρούνται πιο γρήγορα, έχει ανατεθεί στο UAV να περιμένει το “γείτονα” του που θα χασομερίσει το σημείο  $a$  με απόσταση  $(l_1+l_2)/2$  όπου το τελικό σημείο ήταν παρελθόν. Μετρώντας την απόκλιση του  $l_1$  με τη νέα απόσταση πίσω στο τελικό σημείο, το UAV1 μπορεί να ενημερώσει την απόσταση που θα “χασομερήσει” για να αναιρέσει την επίδραση της αύξησης στην περιοχή αυτή.

Προσθέτοντας UAVs στη περίμετρο ισοδυναμεί με μια νοητή κλωστή από περιμετρικά τμήματα που έχουν αλλάξει τα τελικά σημεία. Τα τελικά σημεία για ένα τμήμα μοιρασμένο από ένα ζευγάρι UAVs είναι οι εξωτερικοί γείτονες από τα άλλα UAVs. Θα χρησιμοποιήσουμε τη μέθοδο Monte Carlo για να βεβαιωθούμε ότι ζεύγη εξισορρόπησης φορτίου θα οδηγήσουν την ομάδα σε σύγκλιση. Υποστηρίζουμε ότι με την εξισορρόπηση του μήκους μοιρασμένο σε κάθε ζεύγος των UAVs, η ομάδα στο σύνολό της θα εξαπλωθεί ομοιόμορφα γύρω από την περίμετρο και έτσι θα επιτευχθεί η ελάχιστη λανθάνουσα διαμόρφωση. Εάν ο αλγόριθμος μπορεί να εμφανίσει τη

σύγκλιση για αυθαίρετες αρχικές συνθήκες και με έναν αυθαίρετο αριθμό των μελών της ομάδας, στη συνέχεια, η εισαγωγή / διαγραφή μπορεί να αναλυθεί με την εξέταση του τροποποιημένου συστήματος (μετά την εισαγωγή / διαγραφή), με τις αρχικές συνθήκες οι οποίες δίνονται από την κατάσταση του αρχικού συστήματος κατά τη στιγμή της εισαγωγής / διαγραφής.

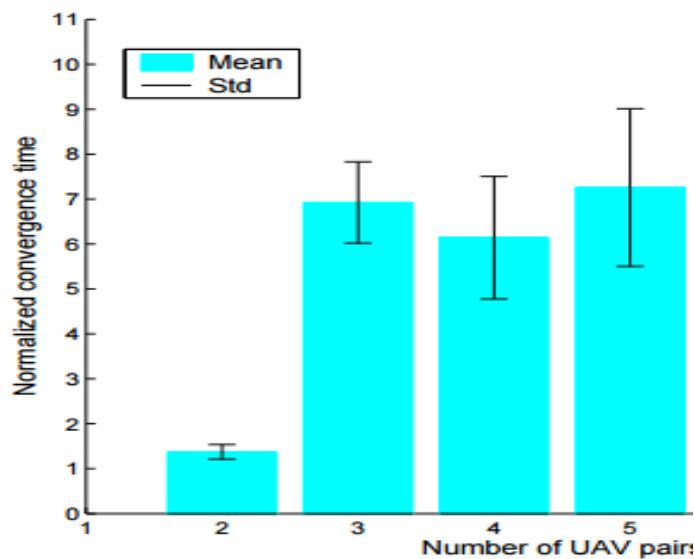
Κάθε UAV υλοποιεί τον ακόλουθο αλγόριθμο:

### 2.5.3. Αλγόριθμος Εξισορρόπησης Φορτίου

1. Διατήρηση μιας εκτιμητέας απόστασης που διανύθηκε από την τελευταία συνάντηση σε κάθε κατεύθυνση (κάθε UAV μοιράζεται ένα τμήμα με το γείτονα στη φορά του ρολογιού και με τον αριστερόστροφο της γείτονα).
2. Σε μια συνάντηση το UAV που έχει διανύσει τη μικρότερη απόσταση από τότε που συνέβει η τελευταία συνάντηση συμφωνεί να “χασομερήσει” στο μεσαίο σημείο του κοινόχρηστου τμήματος και την επόμενη φορά θα παρακολουθεί την περίμετρο σε αυτή την κατεύθυνση (δεξιόστροφα / αριστερόστροφα).
3. Εάν το τελικό σημείο ενός τμήματος έχει αλλάξει (λόγω της αύξησης της περιμέτρου ή λόγω κάποιου γείτονα), τότε η απόσταση που μπορεί το UAV να καθυστερήσει επαυξάνεται από τη μεταβολή της απόστασης του τελικού σημείου. Αυτό κρατά το σημείο καθυστέρησης όπως θα λέγαμε στην ίδια θέση σε σχέση με το μήκος του τμήματος, όπως αυτό ανακοινώθηκε κατά τη διάρκεια της συνάντησης.
4. Τουλάχιστον ένα UAV σε συνάντηση ζεύγους (π.χ. το ένα που έχει διανύσει τη μεγαλύτερη απόσταση) δεν πρέπει να “χασομερήσει” καθ 'οδόν προς την επόμενη αναμενόμενη συνάντηση. Αυτό εξασφαλίζει ότι τα ζεύγη των UAVs θα συναντηθούν και πάλι πάντα, ανεξάρτητα από την αλλαγή της περιμέτρου.

Θα δείξουμε τώρα μέσα από προσομοίωση Monte-Carlo, ότι ο αλγόριθμος εξισορρόπησης φορτίου συγκλίνει στην ελάχιστη διαμόρφωση χρόνου αναμονής αυθαίρετες αρχικές συνθήκες. Ένα παράδειγμα προσομοίωσης αποτελείται από την δρομολόγηση  $N$  ζευγών UAVs από το σταθμό βάσης σε τυχαίες χρονικές στιγμές γύρω από ένα σταθερό μήκος σε κυκλική περίμετρο. Κάθε μέλος της ομάδας ισορροπεί συνεχώς το φορτίο, το οποίο μοιράζεται με κάθε έναν από τους δύο γειτονές του.

Η προσομοίωση συνεχίζεται μέχρι όλοι οι παράγοντες είναι εντός της επίτευξης της ελάχιστης λανθάνουσας διαμόρφωσης ή του μέγιστου χρόνου. Για κάθε  $N \in \{2, \dots, 7\}$ , 100.000 προσομοιώσεις που εκτελέστηκαν και ο χρόνος που απαιτείται για την επίτευξη σταθερής κατάστασης καταγράφονται. Δεδομένου ότι ο χρόνος σύγκλισης είναι μια συνάρτηση της ταχύτητας των UAVs και το μέγεθος της περιμέτρου, ο χρόνος σύγκλισης κανονικοποιείται με το χρόνο που απαιτείται για τη μεταφορά των πληροφοριών γύρω από την περίμετρο. Για παράδειγμα εάν ο χρόνος σύγκλισης ήταν  $T$ , τότε ένα UAV μπορούσε να διασχίσει  $T$  φορές όλη την περίμετρο στην ποσότητα του χρόνου που απαιτείται για την ομάδα να συγκλίνουν στη κατάσταση με την ελάχιστη λανθάνουσα διαμόρφωση. Ας δούμε λίγο το παρακάτω σχήμα το οποίο μας παρουσιάζει τα αποτελέσματα της μεθόδου Monte-Carlo για 100,000 επαναλήψεις για κάθε  $N = 2, \dots, 7$  ζευγάρια των UAVs (με  $\epsilon=0.0003$ ).



Εικόνα 5. Χρόνος σύγκλισης με βάση τον αριθμό των ζευγών UAV

Κάθε παράδειγμα προσομοίωσης συγκλίνει στην ελάχιστη διαμόρφωση λανθάνουσα για κάθε  $N \in \{2, \dots, 7\}$ . Μια εισαγωγή ή διαγραφή ενός ζεύγους UAVs μπορεί να αντιπροσωπεύεται από ένα σύνολο αρχικών συνθηκών, το οποίο δίνεται από την κατάσταση του συστήματος πριν την αλλαγή με μια νέα τιμή του  $N$ . Οι προσομοιώσεις της μεθόδου αυτής δείχνουν ότι κάτω από κάθε αρχικές συνθήκες για κάθε  $N$ , η σταθερότητα του αλγορίθμου δεν θα επηρεάζεται.

Ο αλγόριθμος εξισορρόπησης φορτίου εντοπίζει και παρακολουθεί τη διαστολή / συστολή της περιμέτρου από την κατασκευή: το πραγματικό μήκος της περιμέτρου δοκιμάζεται αποτελεσματικά και συνεχώς από μια ομάδα ατόμων. Επιπλέον οι προσομοιώσεις αυτές έχουν δείξει ότι η εισαγωγή / διαγραφή ζευγών UAV δεν θα επηρεάσει την σταθερότητα του αλγορίθμου. Καταλήγουμε στο συμπέρασμα ότι ο αλγόριθμος εξισορρόπησης φορτίου θα συγκλίνει στην ελάχιστη λανθάνουσα διαμόρφωση με την παρουσία του μέλους της ομάδας εισαγωγής / διαγραφής και τις πεπερασμένες μεταβολές στο μήκος της περιμέτρου.

## 2.6. Πλεονεκτήματα- Μειονεκτήματα

Κάνοντας τον απολογισμό για την χρήση των συγκεκριμένων αεροσκαφών για την επιτήρηση δασικών εκτάσεων, συνοψίζουμε τα εξής πλεονεκτήματα και μειονεκτήματα

### Πλεονεκτήματα:

- Είναι πολύ πιο ασφαλές να χρησιμοποιούνται αυτά αντί των οχημάτων με ανθρώπινη καθοδήγηση, καθώς δεν κινδυνεύουν οι ζωές των εργαζομένων.
- Προλαμβάνεται άμεσα η αντιμετώπιση της πυρκαγιάς, αφού μπορεί να εντοπιστεί από τα UAV, τα οποία παρακολουθούν χωρίς διακοπή τη δασική έκταση.
- Εξαιρετικά υψηλή ανάλυση στο χώρο της δασικής έκτασης. Σε πτήσεις τα μικρά μη επανδρωμένα αεροσκάφη, συνήθως της τάξης των 50-300 m, μας επιτρέπουν την απόκτηση της εξαιρετικά υψηλής χωρικής ανάλυσης εικόνων, με pixels της τάξης των



λίγων εκατοστών (αντί για λίγα μέτρα). Έτσι η λειτουργία αυτή βελτιώνει σημαντικά την οπτική ανάλυση των εικόνων. Η ανάλυση των εικόνων drone θα επιτρέψει την παρακολούθηση των δασών και θα εμπεριέχει πολλά γνωρίσματα που τώρα είναι εφικτά με την βοήθεια της σύγχρονης τεχνολογίας, με άλλα σύνολα δεδομένων τηλεπισκόπησης. Αυτά περιλαμβάνουν την αναγνώριση των μεμονωμένων ειδών δέντρων με σύζευξη εικόνων με τη βοτανική πείρα, η ανίχνευση των χωροκατακτητικών ειδών φυτών και τα παράσιτα, η εκτίμηση της υπέργεια βιομάζα και ο προσδιορισμός των διαφόρων σταδίων της αναγέννησης του δάσους ή υποβάθμιση, τα οποία είναι θεμελιώδους σημασίας για την εκτίμηση της κατάστασης της υγείας των δασών, την αποθήκευση διοξειδίου του άνθρακα και των επιπέδων της βιοποικιλότητας.

- Η υψηλή χρονική ανάλυση. Το συγκριτικά χαμηλότερο κόστος λειτουργίας και συντήρησης των μικρών UAV επιτρέπει στους χρήστες να αποκτήσουν εικόνες πολύ πιο συχνά από ό, τι με τις συμβατικές τεχνολογίες τηλεπισκόπησης, όπως δορυφορικές εικόνες και με κυβερνητικά αεροσκάφη. Αυτό σημαίνει ότι οι χρήστες των drone θα έχουν τη δυνατότητα να ενημερώνουν τις εικόνες τους και να συγκεντρώνουν υψηλής ανάλυσης χρονολογικές σειρές από εικόνες που θα επιτρέψουν με τη σειρά τους τη διεξοδική αξιολόγηση της τοπικής κατάστασης των δασών σε πολύ μικρότερα χρονικά διαστήματα.
- Η δυνατή ορατότητα σε σύννεφα. Στις δασικές εκτάσεις είναι γνωστό ότι συχνά μπορούμε να συναντήσουμε υγρασία και αρκετή ομίχλη και σύννεφα. Με την βοήθεια λοιπόν των UAV έχουμε την δυνατότητα επίβλεψης της δασικής έκτασης ακόμα και σε περιπτώσεις έντονης ομίχλης.
- Τρισδιάστατη εικόνα. Αυτό είναι ένα πλεονέκτημα που μας βοηθάει να κατανοήσουμε καλύτερα τον χώρο (ύψος, πυκνότητα δέντρων κτλ) αλλά και να οργανώσουμε πιο αποτελεσματικές στρατηγικές αντιμετώπισης σε μια πυρκαγιά.
- Η απόκτηση των δεδομένων της αποκέντρωσης. Αυτό έχει σημαντικά πλεονεκτήματα όχι μόνο στις κοινότητες, αλλά στις εταιρίες με δασικές εκτάσεις, συμπεριλαμβανομένων και των κυβερνητικών οργανισμών. Έτσι συλλέγοντας δεδομένα των δασών, μέσω ενός μη

επανδρωμένου αεροσκάφους, η προσέγγιση αυτή θα επιτρέψει τη δημιουργία ή την ενίσχυση των απογραφών των εθνικών δασών αλλά και των τροπικών, οπότε θα έχουμε βελτίωση της διαχείρισης των κοινοτικών δασικών πόρων.

- Ενισχυμένη παρακολούθηση των παράνομων δραστηριοτήτων. Παράνομη υλοτόμηση θα μπορούσε να παρακολουθείται με αυτά τα συστήματα, με την βοήθεια χρονολογημένων φωτογραφιών. Επίσης μπορεί να εντοπίζει ίχνη εξόρυξης και να υπάρχει τακτική παρακολούθηση των ορίων σε πραγματικό χρόνο με βιντεοσκόπηση. Θα μπορούσε επίσης να παρακολουθούνται έγκαιρα η παράνομη εκμετάλλευση των δασικών πόρων και η λαθροθηρία.
- Πρόσβαση σε δυσπρόσιτες ή απομακρυσμένες περιοχές. Δυσπρόσιτες περιοχές εντός του εδάφους που επιθυμούμε (π.χ., απότομες πλαγιές, βραχώδες έδαφος, βάλτους κτλ) θα μπορούσαν να ερευνηθούν με τα μικρά μη επανδρωμένα αεροσκάφη.

#### Μειονεκτήματα:

- Λόγω του ότι τα UAV είναι εφοδιασμένα με υψηλή τεχνολογία και διάφορους αισθητήρες (π.χ., πολυφασματικής, υπερφασματικές, LIDAR, ραντάρ), το κόστος τους φτάνει αρκετά ψηλά.
- Πιθανά προβλήματα για επισκευές και συντήρηση. Η επισκευή τους είναι αρκετά δύσκολη αφού περιέχουν πολλά και πολύπλοκα συστήματα τα οποία μπορεί να μην τα γνωρίζει ένας απλός χρήστης. Για παράδειγμα εάν υπάρξει κάποια σύγκρουση που αφήσει σημαντικό πρόβλημα, αυξάνει σημαντικά το κόστος της επισκευής.
- Η έκθεση μυστικών τοποθεσιών σε άτομα και οργανώσεις που δεν θα έπρεπε να έχουν πρόσβαση σε τέτοιες πληροφορίες.

## Κεφάλαιο 3: Βασικοί κώδικες

### 3.1. Κώδικας αλγορίθμου λειτουργίας σκάφους

Ο βασικός αλγόριθμος στο Matlab βάσει του οποίου λειτουργεί το σκάφος δίνεται παρακάτω:

```
global Jtp Ixx Iyy Izz b d l m g Kpz Kdz Kpp Kdp Kpt Kdt Kpps Kdps ZdF PhidF ThetadF PsidF  
ztime phitime thetatime psitime Zinit Phiinit Thetainit Psiinit Uone Utwo Uthree Ufour Ez Ep Et  
Eps
```

```
% Read the gain values
```

```
GainsK = importdata('Gains.mat');
```

```
Kpz = GainsK(1,1); % Height P controller
```

```
Kdz = GainsK(1,2); % Height D controller
```

```
Kpp = GainsK(1,3); % Roll P controller
```

```
Kdp = GainsK(1,4); % Roll D controller
```

```
Kpt = GainsK(1,5); % Pitch P controller
```

```
Kdt = GainsK(1,6); % Pitch D controller
```

```
Kpps = GainsK(1,7); % Yaw P controller
```

```
Kdps = GainsK(1,8); % Yaw D controller
```

```
%
```

```
% % % Best Gains of the controllers
```

```
%
```

```
% Kpp = 30

% Kdp = 5

%

% Kpt = 30

% Kdt = 5

%

% Kpps = 30

% Kdps = 5

%

% Kpz = 40

% Kdz = 12

% % save gains data

% Gains = [Kpz Kdz Kpp Kdp Kpt Kdt Kpps Kdps];

% save('Gains.mat','Gains') % Write the optimal gains in a .mat file

% Quadrotor constants

Ixx = 8.1*10^(-3); % Quadrotor moment of inertia around X axis
```

```
Iyy = 8.1*10(-3); % Quadrotor moment of inertia around Y axis
```

```
Izz = 14.2*10(-3); % Quadrotor moment of inertia around Z axis
```

```
Jtp = 104*10(-6); % Total rotational moment of inertia around the propeller axis
```

```
b = 54.2*10(5-6); % Thrust factor
```

```
d = 1.1*10(-6); % Drag factor
```

```
l = 0.24; % Distance to the center of the Quadrotor
```

```
m = 1; % Mass of the Quadrotor in Kg
```

```
g = 9.81; % Gravitational acceleration
```

```
stepsize = 0.01;
```

```
t = 0.000000000000000:stepsize:5.000000000000000; % simulation time
```

```
% Initial conditions for the Quadrotor
```

```
x0 = [];
```

```
for i = 1:12,
```

```
    x0 = [x0;0]; % Initial position is the origin O(0,0,0), linear and angular velocities/accelerations  
    = 0
```

```
end
```

```
% Initial values
```

```
Zinit = 0;
```

```
Phiinit = pi/3;
```

```
Thetainit = pi/2;
```

```
Psiinit = -pi/5;
```

```
x0(1) = 3; % Xinit
```

```
x0(3) = 4; % Yinit
```

```
x0(5) = Zinit; % Zinit
```

```
x0(7) = Phiinit; % Phiinit
```

```
x0(9) = Thetainit; % Thetainit
```

```
x0(11) = Psiinit; % Psiinit
```

```
% Desired values
```

```
% Zd = 10*cos(t);
```

```
ZdF = 10;
```

```
PhidF = pi;
```

```
ThetadF = -pi/2;
```

```
PsidF = 2*pi;
```

```
% Solving the differential equation
```

```
options = odeset('OutputFcn','odeplot');
```

```
[t,x] = ode23s(@quad_control_read_fn, t, x0,options); % this function is faster to find the solution
```

```
% ODE15s, ODE23s, and ODE23tb.
```

```
% Desired values
```

```
Zd = 10;
```

```
Phid = pi;
```

```
Thetad = -pi/2;
```

```
Psid = 2*pi;
```

```
% To overcome the error problem:
```



```
% Warning: Failure at t=9.999897e-001. Unable to meet integration tolerances without
```

```
% reducing the step size below the smallest value allowed (1.776357e-015) at time t.
```

```
% We reduce the step size:
```

```
% options=odeset('RelTol',1e-10);
```

```
% [x,y]=ode45(@fn,[0,1],1,options);
```

```
% Plot the path of the Quadrotor in 3-D
```

```
figure(1)
```

```
plot3(x(:,2),x(:,4),x(:,6))
```

```
title('Quadrotor Path in 3-D');
```

```
grid
```

```
% Plot the motion in x, y and z axes
```

```
figure(2)
```

```
hold on
```

```
plot(t,x(:,1),'bx')
```

```
plot(t,x(:,3),'g^')
```

```
plot(t,x(:,5),'r.')
```

```
hold off
```

```
title('Motion in x,y and z axes');
```

```
legend('Motion in x-axis','Motion in y-axis','Motion in z-axis');
```

```
grid
```

```
% Plot the motion in Z-axis
```

```
figure(3)
```

```
% plot(t,Zd,'b-',t,x(:,5),'r--')
```

```
hold on
```

```
plot(t,Zd,'r-.')
```

```
plot(t,x(:,5),'r--','LineWidth',2)
```

```
% legend('Desired motion in Z-axis','Actual motion in z-axis');
```

```
hold off
```

```
title('Height response');
```

```
xlabel('seconds');
```

```
ylabel('meters');
```

```
grid
```

```

% Plot angles' response

figure(4)

hold on

plot(t,Phid,'r',t,x(:,7),'r-.','LineWidth',2)

plot(t,Thetad,'g',t,x(:,9),'g--','LineWidth',2)

plot(t,Psid,'b',t,x(:,11),'b--','LineWidth',2)

hold off

legend('Roll angle in Red','Pitch angle in Green','Yaw angle in Blue');

grid

title('Angle response');

xlabel('seconds');

ylabel('Radians');

% calculate the errors

Z = x(:,5);

Phi = x(:,7);

```

```
Theta = x(:,9);
```

```
Psi = x(:,11);
```

```
Ez = Zd - Z; % find the error in height
```

```
Ep = Phid - Phi; % find the error in height
```

```
Et = Thetad - Theta; % find the error in height
```

```
Eps = Psid - Psi; % find the error in height
```

```
% Plot the errors
```

```
figure(5)
```

```
[p q] = size(Ez);
```

```
t3 = 0:0.01:5; % For exp.3
```

```
hold on
```

```
plot(t3,Ez,'r')
```

```
plot(t3,Ep,'g')
```

```
plot(t3,Et,'b')
```

```
plot(t3,Eps,'k')
```

```

hold off

title('Error vectors');

legend('Z error','Phi error','Theta error','Psi error');

grid

% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% Variables used to decide the effectiveness of the gains

[a b] = size(Ez);

% Calculate the overshoot

ZOvershoot = 50;

ROvershoot = 50;

POvershoot = 50;

YOvershoot = 50;

% Height

if (min(Ez) < 0)

    ZOvershoot = abs(min(Ez))*10; % Height overshoot

end

```

```
if (min(Ez) > 0)

    ZOvershoot = 0; % Height overshoot

end

% Roll

if (min(Ep) < 0)

    ROvershoot = abs(min(Ep))*10; % Roll overshoot

end

if (min(Ep) > 0)

    ROvershoot = 0; % Height overshoot

end

% Pitch

if (min(Et) > 0)

    POvershoot = abs(min(-Et))*10; % Pitch overshoot

end

if (min(Et) < 0)

    POvershoot = 0; % Height overshoot

end
```

```

% Yaw

if (min(Eps) < 0)

    YOvershoot = abs(min(Eps))*10; % Yaw overshoot

end

if (min(Eps) > 0)

    YOvershoot = 0; % Height overshoot

end

Overshoot = [ZOvershoot ROvershoot POvershoot YOvershoot]

% Calculate the steady-state error

Zess = Ez(a);

Ress = Ep(a);

Pess = Et(a);

Yess = Eps(a);

SSError = [Zess Ress Pess Yess]

% Calculate the rise time

Ztr = 10;

```

```
Phitr = 10;
```

```
Thetr = 10;
```

```
Psitr = 10;
```

```
for i = 1:a
```

```
    Ez(i) = Zd - Z(i); % find the error in height
```

```
    if abs(Ez(i)) < 0.05
```

```
        Ztr = i*stepsize;
```

```
        break;
```

```
    end
```

```
end
```

```
for i = 1:a
```

```
    Ep(i) = Phid - Phi(i); % find the error in height
```

```
    if abs(Ep(i)) < 0.05
```

```
        Phitr = i*stepsize;
```

```
        break;
```

```
    end
```

```
end
```



```

for i = 1:a

    Et(i) = Thetad - Theta(i); % find the error in height

    if abs(Et(i)) < 0.05

        Thetr = i*stepsize;

        break;

    end

end

for i = 1:a

    Eps(i) = Psid - Psi(i); % find the error in height

    if abs(Eps(i)) < 0.05

        Psitr = i*stepsize;

        break;

    end

end

Ztr = Ztr - ztime;

Phitr = Phitr - phitime;

Thetr = Thetr - thetotime;

```

Psitr = Psitr - psitime;

Rise\_time = [Ztr Phitr Thetatr Psitr]

%  
% %

%  
% %

% Calculate the controls in order to draw them

U1 = m\*(g + Kpz\*Ez + Kdz\*( - x(:,6)))/(cos(x(:,9)).\*cos(x(:,7))); % Total Thrust on the body  
along z-axis

U2 = (Kpp\*Ep + Kdp\*( - x(:,8))); % Roll input

U3 = (Kpt\*Et + Kdt\*( - x(:,10))); % Pitch input

U4 = (Kpps\*Eps + Kdps\*( - x(:,12))); % Yawing moment

U1 = real(U1);

U2 = real(U2);

U3 = real(U3);

U4 = real(U4);

% Bounding the controls

```
[p q] = size(U2);

for n = 1:p

    % Bounding U1

    if U1(n) > 15.7

        U1(n) = 15.7;

    end

    if U1(n) < 0

        U1(n) = 0;

    end

    % Bounding U2

    if U2(n) > 1

        U2(n) = 1;

    end

    if U2(n) < -1

        U2(n) = -1;

    end

end
```

```
% Bounding U3
```

```
if U3(n) > 1
```

```
    U3(n) = 1;
```

```
end
```

```
if U3(n) < -1
```

```
    U3(n) = -1;
```

```
end
```

```
% Bounding U4
```

```
if U4(n) > 1
```

```
    U4(n) = 1;
```

```
end
```

```
if U4(n) < -1
```

```
    U4(n) = -1;
```

```
end
```

```
end
```

```
% Plot the control vector
```

```
figure(6)
```

```
t2 = 0:0.01:5; % For exp.3
```

```
hold on
```

```
plot(t2,U1,'r')
```

```
plot(t2,U2,'g')
```

```
plot(t2,U3,'b')
```

```
plot(t2,U4,'k')
```

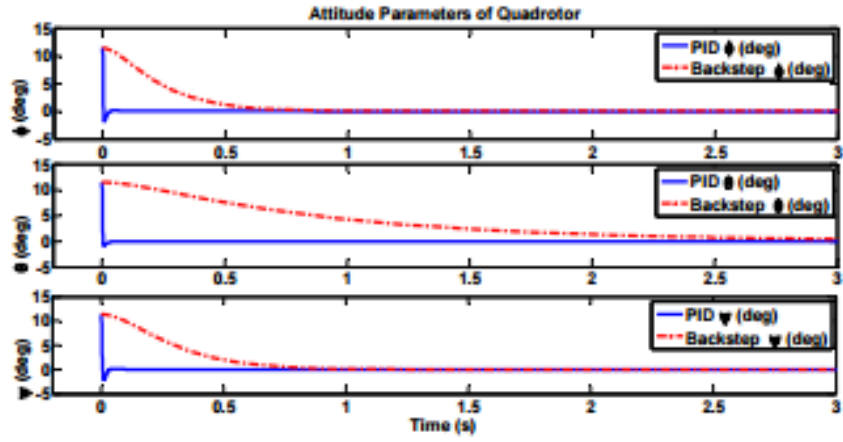
```
hold off
```

```
title('Control vectors');
```

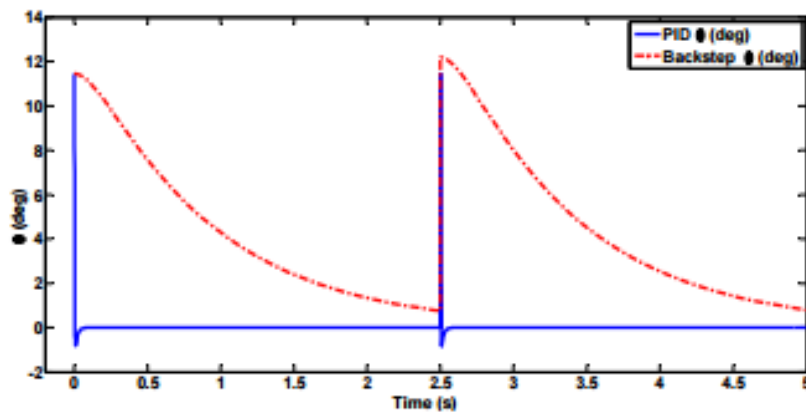
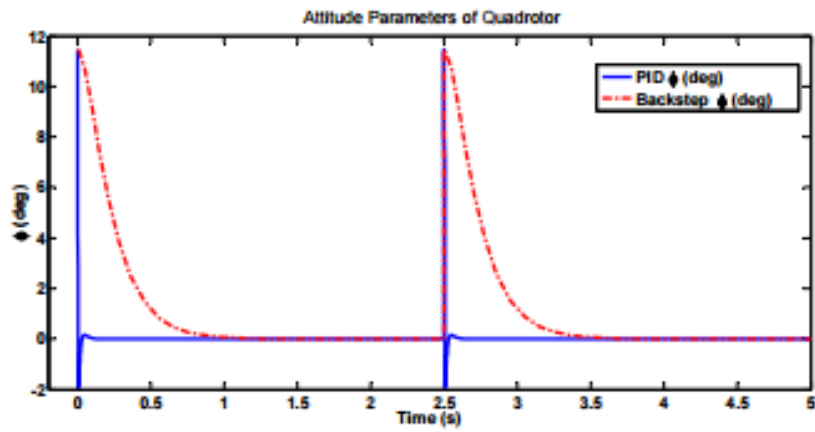
```
legend('U1 Control','U2 Control','U3 Control','U4 Control');
```

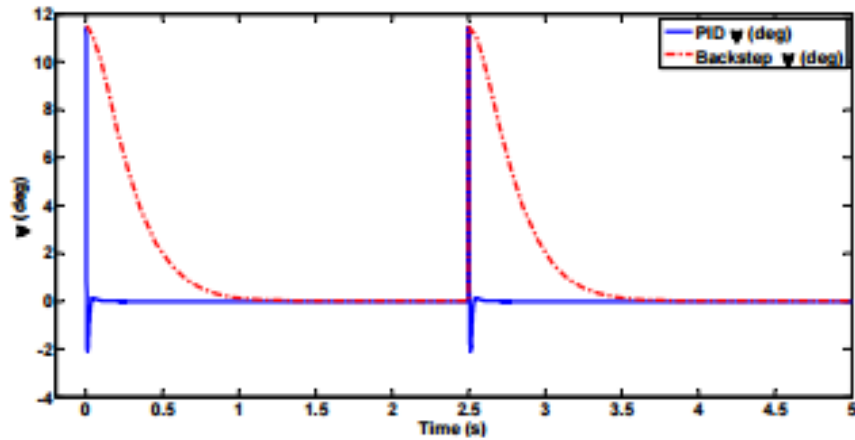
```
grid
```

Παίρνοντας τα αποτελέσματα σε γραφήματα έχουμε τα ακόλουθα:

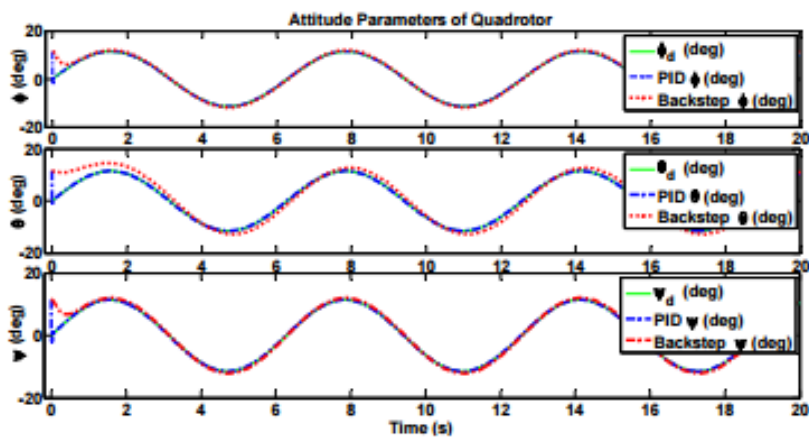


Εικόνα 6. Σήματα ρύθμισης θέσης του quadrotor (UAV)





Εικόνα 7. Συνιστώσες απόρριψης αναταράξεων στην θέση του quadrotor (UAV)



Εικόνα 8. Παρακολούθηση των προκαθορισμένων συνιστωσών τροχιάς στην θέση του quadrotor (UAV)

Τα αποτελέσματα δείχνουν ότι ο επιθυμητός ελεγκτής απορρίπτει τις αναταράξεις σε ελάχιστο χρόνο. Υπάρχει μια υπέρβαση στις αποκρίσεις, γεγονός που είναι συνηθισμένο, λόγω της ταχείας απόκρισης. Τέλος χρειαζόμαστε μια γρήγορη απόκριση.

### 3.2. Συμπληρωματικός κώδικας – αλγόριθμος «ανάγνωσης» ελέγχων

Εδώ θα πρέπει να αναφερθεί ότι έπρεπε να κατασκευαστεί συμπληρωματικός αλγόριθμος για να διαβάσει τους ελέγχους ώστε να λειτουργεί ο βασικός αλγόριθμος του σκάφους. Οι επιπλέον έλεγχοι βρίσκονται στα αρχεία controls, ep, eps, et, ez, gains, overshoot, rise\_time και SSerror. Ο κώδικας των συμπληρωματικών ελέγχων παρατίθεται παρακάτω:

```
function xdot = quad_control_fn(t,x)
```

```
global Jtp Ixx Iyy Izz b d l m g Kpz Kdz Kpp Kdp Kpt Kdt Kpps Kdps ZdF PhidF ThetadF PsidF  
ztime phitime thetetime psitime Zinit Phiinit Thetainit Psiinit Uone Utwo Uthree Ufour Ez Ep Et  
Eps
```

```
% The desired values
```

```
% Changes in Z start at t = 3, changes in Phi start at t = 1, changes in
```

```
% Theta start at the origin, and changes in Psi start at t= 2
```

```
% If you want that all start at the origin simply remove the conditions
```

```
    % time for change start of each variable
```

```
ztime = 3;
```

```
phitime = 1;
```

```
thetetime = 0.2;
```

```
psitime = 2;
```

```
%%% HEIGHT %%%
```



```
if t < ztime
```

```
    Zd = Zinit;
```

```
end
```

```
if t >= ztime
```

```
    Zd = ZdF;
```

```
end
```

```
%%%%%%%%%
```

```
%%% Phi %%%
```

```
if t < phitime
```

```
    Phid = Phiinit;
```

```
end
```

```
if t >= phitime
```

```
    Phid = PhidF;
```

```
end
```

```
%%%%%%%%%
```

```
%%% Theta %%%
```

```
if t < thetetime
```

Thetad = Thetainit;

end

if t >= thetatime

Thetad = ThetadF;

end

%%%%%%%%%

%%% Psi %%%

if t < psitime

Psid = Psiinit;

end

if t >= psitime

Psid = PsidF;

end

%%%%%%%%%

PsidF = 2\*pi;

%%%%%%%%%

%%%%%%%%%

```
thetaddot = 0;
```

```
phiddot = 0;
```

```
psiddot = 0;
```

```
Zddot = 0;
```

```
% Zd = 10;
```

```
% Bounding the angles within the  $-2\pi / 2\pi$  range
```

```
if (x(7) > 2*pi || x(7) < - 2*pi)
```

```
    x(7) = rem(x(7),2*pi);
```

```
end
```

```
if (x(9) > 2*pi || x(9) < - 2*pi)
```

```
    x(9) = rem(x(9),2*pi);
```

```
end
```

```
if (x(11) > 2*pi || x(11) < - 2*pi)
```

```
    x(11) = rem(x(11),2*pi);
```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PD-Z-Control %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Evaluate the Controls
```

```
U = []; % The control vector
```

```
U(1) = m*(g + Kpz*(Zd - x(5)) + Kdz*(- x(6)))/(cos(x(9))*cos(x(7))); % Total Thrust on the  
body along z-axis
```

```
U(2) = (Kpp*(Phid - x(7)) + Kdp*(- x(8))); % Roll input
```

```
U(3) = (Kpt*(Thetad - x(9)) + Kdt*(- x(10))); % Pitch input
```

```
U(4) = (Kpps*(Psid - x(11)) + Kdps*(- x(12))); % Yawing moment
```

```
U = real(U);
```

```
U = [U(1);U(2);U(3);U(4)]; % The control vector
```

```
% % Bounding the controls
```

```
% if U(1) > 15.7
```

```
% U(1) = 15.7;
```

```
% end
```

```
%
```

```
% if U(1) < 0
```

```
% U(1) = 0;
```

```

% end

%

% for j = 2:4

%     if U(j) > 1

%         U(j) = 1;

%     end

%

%     if U(j) < -1

%         U(j) = -1;

%     end

% end

% end

% Calculation of angular velocities

omegasqr(1) = (1/4*b)*U(1) + (1/2*b*1)*U(3) - (1/4*d)*U(4);

omegasqr(2) = (1/4*b)*U(1) - (1/2*b*1)*U(2) + (1/4*d)*U(4);

omegasqr(3) = (1/4*b)*U(1) - (1/2*b*1)*U(3) - (1/4*d)*U(4);

omegasqr(4) = (1/4*b)*U(1) + (1/2*b*1)*U(2) + (1/4*d)*U(4);

omegasqr = real(omegasqr);

```

```
omega(1) = sqrt(omegasqr(1));
```

```
omega(2) = sqrt(omegasqr(2));
```

```
omega(3) = sqrt(omegasqr(3));
```

```
omega(4) = sqrt(omegasqr(4));
```

```
% Bounding the angular velocities
```

```
for j = 1:4
```

```
    if omega(j) > 523
```

```
        omega(j) = 523;
```

```
    end
```

```
    if omega(j) < 125
```

```
        omega(j) = 125;
```

```
    end
```

```
end
```

```
omegasqr(1) = (omega(1))^2;
```

```
omegasqr(2) = (omega(2))^2;
```

```

omegasqr(3) = (omegasqr(3))^2;

omegasqr(4) = (omegasqr(4))^2;

% % Bounding the angular velocities

% for j = 1:4

%     if omegasqr(j) > 523

%         omegasqr(j) = 523;

%     end

%

%     if omegasqr(j) < 125

%         omegasqr(j) = 125;

%     end

% end

% Disturbance

Omega = d*(- sqrt(omegasqr(1)) + sqrt(omegasqr(2)) - sqrt(omegasqr(3)) + sqrt(omegasqr(4)));

% Evaluation of the State space wrt H-frame

xdot(1) = x(2); % Xdot

```

```

xdot(2) = (sin(x(11))*sin(x(7)) + cos(x(11))*sin(x(9))*cos(x(7)))*(U(1)/m); % Xdotdot

xdot(3) = x(4); % Ydot

xdot(4) = (-cos(x(11))*sin(x(7)) + sin(x(11))*sin(x(9))*cos(x(7)))*(U(1)/m); % Ydotdot

xdot(5) = x(6); % Zdot

xdot(6) = - g + (cos(x(9))*cos(x(7)))*(U(1)/m); % Zdotdot

xdot(7) = x(8); % phidot

xdot(8) = ((Iyy - Izz)/Ixx)*x(10)*x(12) - (Jtp/Ixx)*x(10)*Omega + (U(2)/Ixx); % pddot =
phidotdot

xdot(9) = x(10); % thetadot

xdot(10) = ((Izz - Ixx)/Iyy)*x(8)*x(12) + (Jtp/Iyy)*x(8)*Omega + (U(3)/Iyy); % qdot =
thetadotdot

xdot(11) = x(12); % thetadot

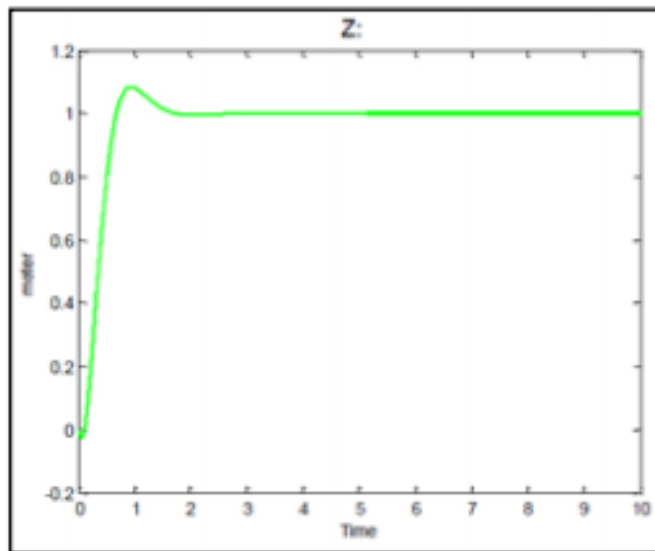
xdot(12) = ((Ixx - Iyy)/Izz)*x(8)*x(10) + (U(4)/Izz); % rdot = psidotdot

xdot = xdot';

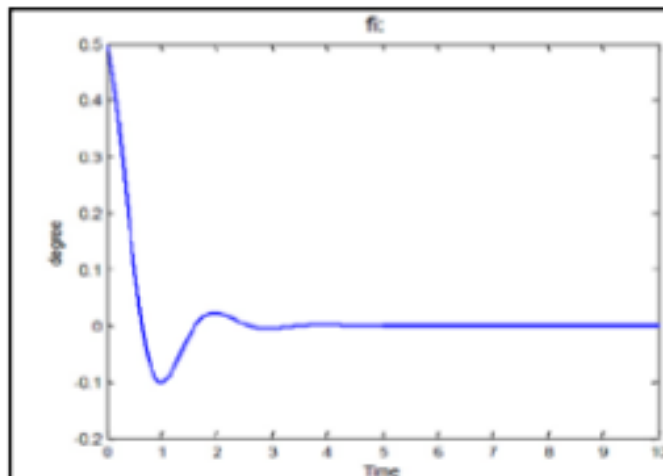
```



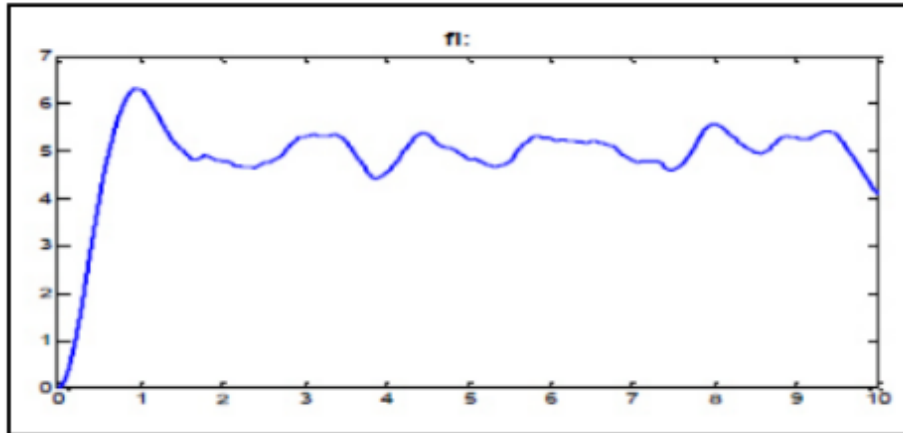
Εδώ έχουμε τα παρακάτω συμπληρωματικά γραφήματα:



Εικόνα 9. Μεταβολή του ύψους



Εικόνα 10. Μεταβολή της γωνίας



*Εικόνα 11. Χρόνος απόκρισης συστήματος έπειτα από τις αναταράξεις*

Μπορούμε να συμπεράνουμε ότι το συγκεκριμένο σύστημα quadrotor ελέγχεται καλά με τον ελεγκτή PID όταν είναι σε μια σταθερή διαδρομή και μακριά από αναταράξεις, αλλά όταν το σύστημα είναι εκτεθειμένο σε αναταράξεις και θόρυβο η απόκριση του συστήματος από μόνη της δεν είναι κατάλληλη με τον έλεγχο PID και πρέπει να αλλάξει, σχετικά με τις διαφορετικές στιγμές των συντελεστών ελέγχου, πράγμα που μπορεί να συμβεί με την παρουσία ασαφούς ελέγχου, μαζί με τον έλεγχο PID.

### 3.3. Σύγκριση κωδίκων

Ο αλγόριθμος λειτουργίας του UAV που εμφανίστηκε προηγουμένως παρουσιάζει τα ακόλουθα πλεονεκτήματα έναντι των υπολοίπων:

1. Επιτεύχθηκε μια αρκετά ρεαλιστική ανάπτυξη ενός μοντέλου quadcopter που είναι το βασικότερο και πιο δύσκολο κομμάτι της μοντελοποίησής μας.

2. Κατορθώθηκε έλεγχος των τεσσάρων μοτέρ σε τέτοιο βαθμό, ώστε να φτάνει αρκετά γρήγορα στην επιθυμητή μας ταχύτητα και να διατηρεί την τιμή αυτή για να μην υπάρχουν αστάθειες κατά την πτήση. Ο PID έλεγχος που αναπτύσσεται στην εργασία αυτή, γίνεται με παραμέτρους που θα μπορούσαν (ίσως) να εφαρμοστούν και σε ένα πραγματικό μοντέλο.
3. “Φιλτραρίστηκαν” οι μετρήσεις μας από τα αισθητήριά μας σε ικανοποιητικό βαθμό, με την χρήση του φίλτρου Kalman. Ο αλγόριθμος αυτός είναι αρκετά πολύπλοκος και χρειάζεται χρόνο και αρκετά πειράματα μέχρι να εξοικειωθούμε με αυτόν. Αποτέλεσε ένα από τα απαιτητικότερα στοιχεία της εργασίας αυτής, και αποδείχθηκε πολύ αποτελεσματικό.
4. Η μοντελοποίηση ενός quadcopter είναι μια πρώτη προσέγγιση ενός πραγματικού τέτοιου συστήματος, στο οποίο μπορούμε να επεμβαίνουμε με περισσότερη ασφάλεια απ’ ότι στο πραγματικό (γιατί κάποιο ενδεχόμενο λάθος, θα σήμαινε αυτομάτως κάποια ενδεχόμενη πτώση). Όμως παρόλο που μπορεί το μοντέλο μας να δείχνει αρκετά ικανοποιητικό, υπάρχει πιθανότητα το ίδιο ακριβώς μοντέλο, να έχει προβλήματα σε πραγματικό χρόνο.
5. Καθώς “τρέχουμε” το μοντέλο μας στο MATLAB-Simulink παρατηρήθηκε πως όταν υπάρχει ο ελεγκτής ασαφούς λογικής στο μοντέλο μας, αυτό γίνεται πολύ αργό, και ο λόγος είναι πως γίνεται μεταφορά δεδομένων από και προς το κεντρικό περιβάλλον του MATLAB με αποτέλεσμα όλη η διαδικασία αυτή να “βαραίνει” το μοντέλο μας.

## Συμπεράσματα

Ανέκαθεν, υπήρχαν παραδείγματα τεχνολογιών που θα μπορούσαν να ελέγχονται εξ αποστάσεως μέσω μηχανικών συνδέσεων (π.χ. μηχανικά χειριστήρια για το χειρισμό επικίνδυνων χημικών ουσιών), καθώς και άλλες τεχνολογίες που παρέχουν προ-προγραμματισμένες κινήσεις (π.χ. βλήματα και τορπίλες). Ωστόσο, μόνο με την ανάπτυξη της μικροηλεκτρονικής και ενσωματωμένων συστημάτων υπολογισμού κατέστη δυνατό να σχεδιαστούν συστήματα που συνδυάζουν τόσο την κινητικότητα όσο και την αυτονομία. Τέσσερις σημαντικές ερευνητικές προκλήσεις έχουν κυριαρχήσει στις εξελίξεις αυτές, και συνεχίζουν να αποτελούν τα βασικά θέματα για διεθνή μελέτη:

### 1. Μηχανισμοί και κινητικότητα

Τόσο η εφαρμοσμένη μηχανική όσο και η βιομηχανική τεχνολογία χρησιμοποιούνται πλέον στο σχεδιασμό ρομποτικών οχημάτων, καθώς και οι τρέχουσες ερευνητικές προσπάθειες συνεχίζουν να εξελίσσονται. Βασικά ερευνητικά θέματα περιλαμβάνουν: αρχές της κίνησης και ιδιότητες των υλικών και σχεδιασμό.

### 2. Ισχύς και προώθηση

Η μακροπρόθεσμη αυτονομία των οχημάτων είναι άμεσα επηρεαζόμενη από τη διαθέσιμη ισχύ και την ενεργειακή απόδοση της κίνησης. Αυτές οι θεωρίες είναι πλέον σημαντικές σε απομακρυσμένες περιοχές, όπως η διαστημικές και υποθαλάσσιες υλοποιήσεις, όταν η ανάκτηση ενέργειας ή ο ανεφοδιασμός είναι ανέφικτο.

### 3. Υπολογισμός και έλεγχος

Η εισαγωγή στα microcomputation επέτρεψε τη χρήση των ενσωματωμένων συστημάτων πληροφορικής που είναι μικρά, ελαφριά και υψηλής ενεργειακής απόδοσης. Τέτοια ενσωματωμένα υπολογιστικά συστήματα έχουν συμβάλει στην ανάπτυξη των ρομποτικών οχημάτων με εξελιγμένες αρχιτεκτονικές υπολογιστών που οργανώνουν αισθητήρες που βασίζονται στην ανατροφοδότηση και τον έλεγχο των ενεργειών επί του οχήματος (onboard).

#### 4. Αισθητήρες και πλοήγηση

Οι πρόσφατες εξελίξεις στην επιτυχία της ρομποτικής τεχνολογίας οχημάτων για το έδαφος, τον αέρα και το νερό έχουν συνδεθεί με την ανάπτυξη βελτιωμένων αισθητήρων και δικτύων αισθητήρων. Τέτοιοι αισθητήρες εξυπηρετούν δύο σημαντικούς σκοπούς για ρομποτικά οχήματα:

a. Αισθητήρες που παρακολουθούν το περιβάλλον και χρησιμοποιούνται για τον έλεγχο διαδραστικών εργασιών.

b. Αισθητήρες είναι απαραίτητοι για τη βασική πλοήγηση ενός ρομποτικού οχήματος

## Βιβλιογραφία

- Şahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application. In *Swarm robotics* (pp. 10-20). Springer Berlin Heidelberg.
- Beni, G. (2004). From swarm intelligence to swarm robotics. In *Swarm robotics* (pp. 1-9). Springer Berlin Heidelberg.
- Brambilla, M., Ferrante, E., Birattari, M., & Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1), 1-41.
- Dorigo, M., Tuci, E., Groß, R., Trianni, V., Labella, T. H., Nouyan, S., ... & Mondada, F. (2004). The swarm-bots project. In *Swarm Robotics* (pp. 31-44). Springer Berlin Heidelberg.
- Payton, D., Estkowski, R., & Howard, M. (2004). Pheromone robotics and the logic of virtual pheromones. In *Swarm Robotics* (pp. 45-57). Springer Berlin Heidelberg.
- Winfield, A. F., Harper, C. J., & Nembrini, J. (2004). Towards dependable swarms and a new discipline of swarm engineering. In *Swarm robotics* (pp. 126-142). Springer Berlin Heidelberg.
- Dorigo, M., & Roosevelt, A. F. (2004). Swarm robotics. In *Special Issue*, *Autonomous Robots*.
- Dorigo, M. (2005). SWARM-BOT: An experiment in swarm robotics. In *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE* (pp. 192-200). IEEE.
- Eisenbeiss, H. (2004). A mini unmanned aerial vehicle (UAV): system overview and image acquisition. *International Archives of Photogrammetry. Remote Sensing and Spatial Information Sciences*, 36(5/W1).
- Tisdale, J., Ryan, A., Kim, Z., Törnqvist, D., & Hedrick, J. K. (2008, June). A multiple UAV system for vision-based search and localization. In *American Control Conference, 2008* (pp. 1985-1990). IEEE.
- Cai, G., Chen, B. M., Lee, T. H., & Dong, M. (2009). Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters. *Mechatronics*, 19(7), 1057-1066.
- Gancet, J., Hattenberger, G., Alami, R., & Lacroix, S. (2005, August). Task planning and control for a multi-UAV system: architecture and algorithms. In *Intelligent Robots and*

Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on (pp. 1017-1022). IEEE.

- Everaerts, J., Lewyckyj, N., & Fransaer, D. (2004). Pegasus: design of a stratospheric long endurance UAV system for remote sensing. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 35(Part B).
- Nikolic, J., Burri, M., Rehder, J., Leutenegger, S., Huerzeler, C., & Siegwart, R. (2013, March). A UAV system for inspection of industrial facilities. In Aerospace Conference, 2013 IEEE (pp. 1-8). IEEE.

## Πηγές Διαδικτύου

- <http://www.naftemporiki.gr/story/647858/smini-autonomon-iptamenon-rompot-kai-aspida-enantion-drones>
- [http://p-comp.di.uoa.gr/resources/nikitidis\\_thesis.pdf](http://p-comp.di.uoa.gr/resources/nikitidis_thesis.pdf)
- [http://nemertes.lis.upatras.gr/jspui/bitstream/10889/630/1/MSc\\_final.pdf](http://nemertes.lis.upatras.gr/jspui/bitstream/10889/630/1/MSc_final.pdf)
- <https://www.ufpe.br/latecgeo/images/PDF/artigo%20aps%20uav%20aplications.pdf>
- [https://books.google.gr/books?hl=el&lr=&id=03gdqhU61C0C&oi=fnd&pg=PT14&dq=UAV+\(Unmanned+Aircraft+Vehicles\)&ots=MvpHgvU4I&sig=0ngWB8A2VBBps3H8bQLTCGTVnf8&redir\\_esc=y#v=onepage&q&f=false](https://books.google.gr/books?hl=el&lr=&id=03gdqhU61C0C&oi=fnd&pg=PT14&dq=UAV+(Unmanned+Aircraft+Vehicles)&ots=MvpHgvU4I&sig=0ngWB8A2VBBps3H8bQLTCGTVnf8&redir_esc=y#v=onepage&q&f=false)
- [https://books.google.gr/books?id=NCcgJhGfWMQC&pg=PA10&hl=el&source=gbs\\_toc\\_r&cad=4#v=onepage&q&f=false](https://books.google.gr/books?id=NCcgJhGfWMQC&pg=PA10&hl=el&source=gbs_toc_r&cad=4#v=onepage&q&f=false)
- <http://fireaviation.com/tag/uav/>
- <http://www.firerescue1.com/fire-products/technology/articles/2543849-Firefighting-and-drones-How-they-work-together/>
- <http://money.cnn.com/2013/08/19/technology/innovation/fire-fighting-drones/>
- [http://nemertes.lis.upatras.gr/jspui/bitstream/10889/8251/1/%CE%A3%CE%BA%CE%AC%CE%BD%CF%84%CE%B6%CE%B9%CE%BA%CE%B1%CF%82\\_%CE%9A%C](http://nemertes.lis.upatras.gr/jspui/bitstream/10889/8251/1/%CE%A3%CE%BA%CE%AC%CE%BD%CF%84%CE%B6%CE%B9%CE%BA%CE%B1%CF%82_%CE%9A%C)

F%8E%CF%83%CF%84%CE%B1%CF%82\_%CE%94%CE%95\_%CE%97%CE%9C  
%CE%A4%CE%A5\_6846.pdf

- <http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.90.7934>
- [https://www.upo.es/isa/lmercab/publications/papers/ICFFR10\\_Merinoetal.pdf](https://www.upo.es/isa/lmercab/publications/papers/ICFFR10_Merinoetal.pdf)
- <http://www.mdpi.com/1999-4907/5/6/1481/htm>
- <http://www.nrcresearchpress.com/doi/pdf/10.1139/cjfr-2014-0347>