

TECHNICAL UNIVERSITY OF CRETE

DIPLOMA THESIS

---

# Design and Implementation of a Low Cost Embedded System for Localization of Drones Flying in Swarms

---

*Author:*

Christos SPYRIDAKIS

*Thesis Committee:*

Prof. Apostolos DOLLAS (Supervisor)

Asst. Prof. Eftychios KOUTROULIS

Asst. Prof. Panagiotis PARTSINEVELOS



*A thesis submitted in fulfillment of the requirements  
for the diploma of Electrical and Computer Engineer  
in the*

School of Electrical and Computer Engineering  
Microprocessor and Hardware Laboratory

January 27, 2021



TECHNICAL UNIVERSITY OF CRETE

# *Abstract*

School of Electrical and Computer Engineering

Electrical and Computer Engineer

**Design and Implementation of a Low Cost Embedded System for  
Localization of Drones Flying in Swarms**

by Christos SPYRIDAKIS

TODO: Abstract ...



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

## Περίληψη

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών

Σχεδίαση και Υλοποίηση Ενσωματωμένου Συστήματος Χαμηλού  
Κόστους για Εύρεση Θέσης μη Επανδρωμένων Αεροσκαφών που  
Πετούν σε Σχηματισμό

από τον Χρήστο ΣΠΤΡΙΔΑΚΗ

TODO: Περίληψη ...



# *Acknowledgements*

TODO: Acknowledgements





# Contents

<b>Abstract</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>Contents</b>	<b>ix</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of Algorithms</b>	<b>xv</b>
<b>Physical Constants</b>	<b>xvii</b>
<b>List of Symbols</b>	<b>xix</b>
<b>List of Abbreviations</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Unmanned aerial vehicles . . . . .	2
1.1.1 History and drone types . . . . .	2
1.1.2 Characteristics . . . . .	4
1.2 Motivation . . . . .	5
1.2.1 Nature . . . . .	5
1.2.2 Swarms and Applications . . . . .	6
1.3 Scientific Goals and Contributions . . . . .	7
1.4 Thesis Outline . . . . .	7
<b>2 Theoretical Background</b>	<b>9</b>
2.1 Distance/Angle Estimation . . . . .	12
2.1.1 Distance . . . . .	12
Received Signal Strength . . . . .	12
Propagation Time . . . . .	13

2.1.2	Angle . . . . .	17
	Amplitude response . . . . .	18
	Phase response . . . . .	18
2.2	Position Computation . . . . .	20
2.2.1	Trilateration . . . . .	20
2.2.2	Bounding Box . . . . .	23
2.2.3	Triangulation . . . . .	23
2.2.4	Multilateration . . . . .	25
2.2.5	Probabilistic approaches . . . . .	26
2.3	Localization Algorithm . . . . .	27
2.3.1	One-hop vs Multi-hop . . . . .	28
2.3.2	Anchor-based vs Anchor-free . . . . .	28
2.3.3	Relative vs Absolute Positioning . . . . .	28
2.3.4	Indoor vs Outdoor Scenarios . . . . .	28
2.3.5	Distributed vs Centralized Position Computation . . . . .	29
2.3.6	Range-based vs Range-free . . . . .	30
<b>3</b>	<b>Related Work</b>	<b>33</b>
3.1	Radio Frequency . . . . .	34
3.1.1	UWB, IMU and GPS . . . . .	35
3.2	Thesis Approach . . . . .	37
<b>4</b>	<b>Design Features and Implementation</b>	<b>39</b>
<b>5</b>	<b>Experiments and Results</b>	<b>41</b>
<b>6</b>	<b>Conclusions and Future Work</b>	<b>43</b>
	<b>References</b>	<b>45</b>

# List of Figures

1.1	UAV Examples . . . . .	3
1.2	UAV principal axes . . . . .	4
2.1	Localization System components . . . . .	10
2.2	Localization System overview . . . . .	11
2.3	Ideal RSSI over distance . . . . .	13
2.4	Time of Arrival cases . . . . .	14
2.5	Time Difference of Arrival cases . . . . .	15
2.6	TDoA hyberbolas based on $\Delta t$ . . . . .	16
2.7	Typical anisotropic antenna pattern . . . . .	18
2.8	Antenna Arrays . . . . .	19
2.9	Trilateration examples . . . . .	21
2.10	Bounding Box example . . . . .	23
2.11	Principles of Triangulation . . . . .	24
2.12	Triangulation examples . . . . .	24
2.13	Multilateration example . . . . .	26
2.14	Graph example . . . . .	27
3.1	Important coordinate frames . . . . .	34
3.2	Illustration of Single-sided Two-way ranging . . . . .	35
3.3	Data preprocessing work flow using EKF . . . . .	36
3.4	System overview . . . . .	37



# List of Tables

1.1	Κατηγοριοποίηση των UAV βάσει της δομής τους . . . . .	2
2.1	Nodes' names definitions . . . . .	9
2.2	Distance/angle estimation techniques and position Computation summary . . . . .	26
2.3	Comparison of Centralized vs Distributed . . . . .	29
2.4	Comparison of Range-based vs Range-free . . . . .	30



# List of Algorithms

1	Multi Dimensional Scaling-Mobile Assisted Programming . . . . .	29
2	Approximate Point in Triangle . . . . .	30
3	Distance Vector-Hop . . . . .	31





# Physical Constants

Speed of Light in vacuum  $c_0 = 2.997\,924\,58 \times 10^8 \text{ m s}^{-1}$  (exact)

Pi, Archimedes' constant  $\pi = 3.14159\,26535$  (approximation)



# List of Symbols

$A$	amplitude	
$d$	distance	m
$f$	frequency	Hz
$P$	power	W ( $\text{J s}^{-1}$ )
$r$	radius	m
$s$	speed	$\text{m s}^{-1}$
$t$	time	s
$V$	velocity	$\text{m s}^{-1}$
$\theta$	angle	$^{\circ}$
$\lambda$	wavelength	m
$\phi$	phase	$^{\circ}$
$\omega$	angular frequency	rad



# List of Abbreviations

<b>AoA</b>	<b>A</b> ngle of <b>A</b> rrival
<b>AHD</b>	<b>A</b> verage <b>H</b> op <b>D</b> istance
<b>AHRS</b>	<b>A</b> ltitude <b>H</b> eading <b>R</b> eference <b>S</b> ystem
<b>APIT</b>	<b>A</b> pproximate <b>P</b> oint <b>I</b> n <b>T</b> riangle
<b>BP</b>	<b>B</b> elief <b>P</b> ropagation
<b>BS</b>	<b>B</b> ase <b>S</b> tation
<b>CIF</b>	<b>C</b> ovariance <b>I</b> ntersection <b>F</b> ilter
<b>CL</b>	<b>C</b> ooperative <b>L</b> ocalization
<b>DoA</b>	<b>D</b> irection of <b>A</b> rrival
<b>DoF</b>	<b>D</b> egrees of <b>F</b> reedom
<b>DV-Hop</b>	<b>D</b> istance <b>V</b> ector <b>H</b> op
<b>EKF</b>	<b>E</b> xtended <b>K</b> alman <b>F</b> ilter
<b>ESC</b>	<b>E</b> lectronic <b>S</b> peed <b>C</b> ontrol
<b>FANET</b>	<b>F</b> lying <b>A</b> d-hoc <b>N</b> ETwork
<b>FPV</b>	<b>F</b> irst <b>P</b> erson <b>V</b> iew
<b>FSPL</b>	<b>F</b> ree <b>S</b> pace <b>P</b> ath <b>L</b> oss
<b>GNSS</b>	<b>G</b> lobal <b>N</b> avigation <b>S</b> atellite <b>S</b> ystem
<b>GPS</b>	<b>G</b> lobal <b>P</b> ositioning <b>S</b> ystem
<b>IC</b>	<b>I</b> ntegrated <b>C</b> ircuit
<b>IMU</b>	<b>I</b> ntertial <b>M</b> easurement <b>U</b> nit
<b>ISR</b>	<b>I</b> ntelligence, <b>S</b> urveillance, and <b>R</b> econnaissance
<b>LBSA</b>	<b>L</b> ocalization <b>B</b> ased on <b>S</b> imulated <b>A</b> nnealing
<b>LoS</b>	<b>L</b> ine of <b>S</b> ight
<b>LPS</b>	<b>L</b> ocal <b>P</b> ositioning <b>S</b> ystem <b>A</b> rea <b>N</b> etwork
<b>MCU</b>	<b>M</b> icro <b>C</b> ontroller <b>U</b> nit
<b>MDS-MAP</b>	<b>M</b> ulti <b>D</b> imensional <b>S</b> caling- <b>M</b> obile <b>A</b> ssisted <b>P</b> rogramming
<b>MEMS</b>	<b>M</b> icro-electro-mechanical <b>S</b> ystems
<b>MLAT</b>	<b>M</b> ultilateration
<b>ML</b>	<b>M</b> aximum <b>L</b> ikelihood
<b>MPU</b>	<b>M</b> icro <b>P</b> rocessor <b>U</b> nit
<b>MRS</b>	<b>M</b> inimal <b>R</b> equired number of <b>S</b> ensors

<b>NED</b>	<b>N</b> orth <b>E</b> ast <b>D</b> own
<b>NLoS</b>	<b>N</b> on- <b>L</b> ine of <b>S</b> ight
<b>RF</b>	<b>R</b> adio <b>F</b> requency
<b>RSSI</b>	<b>R</b> eceived <b>S</b> ignal <b>S</b> trength <b>I</b> ndicator
<b>RTLS</b>	<b>R</b> eal <b>T</b> ime <b>L</b> ocation <b>S</b> ystem
<b>RTT</b>	<b>R</b> ound- <b>T</b> rip <b>T</b> ime
<b>SCIF</b>	<b>S</b> plit <b>C</b> ovariance <b>I</b> ntersection <b>F</b> ilter
<b>SR</b>	<b>S</b> warm <b>R</b> obotics
<b>S&amp;R</b>	<b>S</b> earch & <b>R</b> escue
<b>SDP</b>	<b>S</b> emi <b>D</b> efinite <b>P</b> rogramming
<b>SS-TWR</b>	<b>S</b> ingle- <b>S</b> ided <b>T</b> wo- <b>W</b> ay <b>R</b> anging
<b>SNR</b>	<b>S</b> ignal to <b>N</b> oise <b>R</b> atio
<b>TDoA</b>	<b>T</b> ime <b>D</b> ifference of <b>A</b> rrival
<b>ToA</b>	<b>T</b> ime of <b>A</b> rrival
<b>ToF</b>	<b>T</b> ime of <b>F</b> light
<b>UAS</b>	<b>U</b> nmanned <b>A</b> ircraft <b>S</b> ystem
<b>UAV</b>	<b>U</b> nmanned <b>A</b> erial <b>V</b> ehicle
<b>UCAV</b>	<b>U</b> nmanned <b>C</b> ombat <b>A</b> erial <b>V</b> ehicle
<b>UKF</b>	<b>U</b> nscented <b>K</b> alman <b>F</b> ilter
<b>UWB</b>	<b>U</b> ltra- <b>w</b> ideband
<b>VTOL</b>	<b>V</b> ertically <b>H</b> over, <b>T</b> ake-off, and <b>L</b> and
<b>WSN</b>	<b>W</b> ireless <b>S</b> ensor <b>N</b> etworks

*Dedicated to those people who have helped me be the  
person I am today...*





# Chapter 1

## Introduction

“Alone we can do so little,  
together we can do so much”

---

*Hellen Keller*

Τα τελευταία χρόνια η ανάπτυξη της επιστήμης έχει επιφέρει την απόκτηση των τεχνολογικών επιτευγμάτων από το ευρύ κοινό· με ένα πολύ οικονομικό αντίτιμο. Αυτό σημαίνει ότι ο καθένας πολύ εύκολα, μπορεί να έχει στην κατοχή του ακόμα και προϊόντα τα οποία θεωρούνται state-of-the-art, χωρίς να χρειάζεται να δαπανήσει μεγάλα ποσά. Το ίδιο φυσικά συμβαίνει και με τον κλάδο των drone και την - κατά επέκταση - χρήση αυτών· ακόμα και για ψυχαγωγικό σκοπό.

Κατά το τέλος του έτους 2019 μόνο στις Ηνωμένες Πολιτείες της Αμερικής υπήρχαν πάνω από 990 χιλιάδες εγγεγραμμένοι χειριστές drone με πάνω από 1.32 εκατομμύρια drone ψυχαγωγικού χαρακτήρα να χρησιμοποιούνται [1]. Ενώ μέχρι το 2025 υπολογίζεται ότι το μέγεθος αγοράς των υπηρεσιών drone θα κοστολογείται στα 63.6 εκατομμύρια δολάρια [2].

Φυσικά η χρήση τους δεν περιορίζεται μόνο στην ψυχαγωγία, εταιρίες όπως η Amazon έχουν αποκτήσει ήδη τα απαραίτητα πιστοποιητικά και εγκρίσεις, με σκοπό να χρησιμοποιήσουν drone για παράδοση των δεμάτων αρκετά σύντομα [3], καθώς προς το παρόν η διαδικασία βρίσκεται σε στάδιο δοκιμών. Συνεπώς είναι εύκολο να κατανοηθεί ότι ο συγκεκριμένος κλάδος πρόκειται να έχει ακόμα μεγαλύτερη άνθιση, με αρκετά μεγάλο ερευνητικό ενδιαφέρον να του αναλογεί.

Με την αύξηση των drone και την αύξηση των εφαρμογών, υπάρχει η ανάγκη συνεργασίας και η δημιουργία drone swarms για την επιτυχή ολοκλήρωση των στόχων που έχουν οριστεί. Όμως για να καταφέρουν τα drone να συνεργαστούν, χρειάζεται πρώτα να μπορούν να ξεπεράσουν τα προβλήματα τα οποία υπάρχουν.

## 1.1 Unmanned aerial vehicles

Είναι σημαντικό από τα πρώτα βήματα, να έχει γίνει κατανοητό με τον όρο drone σε τι παραπέμπουμε - όπως επίσης πότε θεωρείται ότι ένα σμήνος από drone πετάει σε σχηματισμό (drone swarm).

### 1.1.1 History and drone types

Όταν αναφερόμαστε στον όρο Unmanned Aerial Vehicle (**UAV**) ή απλούστερα drone κάνουμε αναφορά για ένα μη επανδρωμένο ιπτάμενο αεροσκάφος το οποίο ελέγχεται είτε απομακρυσμένα από έναν άνθρωπο, είτε είναι τελείως αυτόνομο. Τα **UAV** μαζί με ένα Base Station (**BS**) και την από κοινού επικοινωνίας του σταθμού - drone, δημιουργούν αυτό που ονομάζουμε Unmanned Aircraft System (**UAS**) [4] [5].

Η πρώτη εμφάνιση των **UAV** έγινε κατά το 1849 στα πλαίσια μάχης, ενώ οι πρώτες καινοτομίες πάνω σε αυτά ξεκίνησαν ήδη από τις αρχές του 20ου αιώνα. Το 2013 τουλάχιστον 50 χώρες χρησιμοποιούσαν **UAVs** για κάποιον σκοπό, με μερικές από αυτές φυσικά να σχεδιάζουν τα δικά τους [5]. Αυτήν την στιγμή υπάρχουν πάνω από 1000 διαφορετικά μοντέλα **UAV** που χρησιμοποιούνται ανά τον κόσμο, με τα περισσότερα από αυτά να μην έχουν ψυχαγωγικό χαρακτήρα [6].

Είναι λοιπόν ξεκάθαρο ότι το πλήθος των drone είναι τόσο μεγάλο, λόγω των διαφορετικών αναγκών - και ότι κάποια έχουν καλύτερα αποτελέσματα από ότι άλλα σε συγκεκριμένες αποστολές. Για αυτό, έχουν γίνει ήδη προσπάθειες για την κατηγοριοποίηση των **UAVs** σύμφωνα με τα διάφορα χαρακτηριστικά που μπορεί να έχουν. Ενδεικτικά με βάση το μέγεθος, την αυτονομία, το βάρος ή το μηχανολογικό σχεδιασμό των **UAV** να είναι μερικές από τις υπάρχουσες [4] [7] [8]. Στο *Table 1.1* υπάρχει μία απλουστευμένη κατηγοριοποίηση η οποία προτάθηκε από τους συγγραφείς του [8] σύμφωνα με τη βασική μηχανολογική δομή που μπορεί να έχει ένα drone καθώς και τα πλεονεκτήματα της κάθε δομής.

TABLE 1.1: Κατηγοριοποίηση των **UAV** βάση της δομής τους [8].

Drones	Main features
Fixed-Wing	long endurance and fast flight speed
Fixed-Wing Hybrid	<b>VTOL</b> and long endurance flight
Single Rotor	<b>VTOL</b> , hover, and long endurance flight
Multicopter	<b>VTOL</b> , hover, and short endurance flight



FIGURE 1.1: UAV Examples

Τυπικά τα Fixed-Wing drones είναι αρκετά ακριβά, χρειάζονται εξειδικευμένους χειριστές για να λειτουργήσουν, όπως επιπλέον και περισσότερο χώρο για την απογείωση και την προσγείωση. Είναι ιδανικά για εφαρμογές που χρειάζεται να καλύψουμε μεγάλες περιοχές και συχνά έχουν αυτονομία τουλάχιστον μερικών ωρών. Για αυτούς τους λόγους χρησιμοποιούνται κυρίως από κυβερνήσεις, στρατιωτικές μονάδες ή επιχειρήσεις για την γρήγορη επίβλεψη μεγάλων εκτάσεων [9].

Τα Fixed-Wing Hybrid προσπαθούν να λύσουν τα μειονεκτήματα που έχουν τα Fixed-Wing drones, την μη ικανότητα δηλαδή για Vertically Hover, Take-off, and Land (VTOL) όμως είναι ακόμα σε αρχικά στάδια [8].

Τα Single Rotor είναι επίσης αρκετά ακριβά, πολύπλοκα μηχανολογικά μηχανήματα, που δέχονται πολλούς κραδασμούς, απαιτούν εξειδικευμένους χειριστές όμως μπορούν να μεταφέρουν αρκετά βαριά payloads, θετικό στην χρήση τους ότι μπορούν να πραγματοποιήσουν VTOL [8].

Τα Multirotor είναι ίσως τα πιο ευρέως διαδεδομένα. Καθώς είναι τα πιο οικονομικά από τα παραπάνω και εύκολο να κατασκευαστούν. Μπορούν να βρεθούν στο εμπόριο με διάφορο πλήθος από έλικες και είναι το κύριο είδος που χρησιμοποιείται από ερασιτέχνες ή χομπίστες για λόγους αναψυχής [8].

Στο *Figure 1.1* δίνονται κάποια ενδεικτικά παραδείγματα UAVs με βάση την κατηγοριοποίηση του *Table 1.1*. Φυσικά αυτή η κατηγοριοποίηση δεν περιλαμβάνει όλα τα είδη drone, είναι όμως ικανοποιητική για να γίνουν ξεκάθαρες δύο βασικές ιδέες. Αρχικά ανάλογα με την εφαρμογή που μας ενδιαφέρει, θα πρέπει να επιλέξουμε την χρήση του πλέον κατάλληλου τύπου drone. Όπως επίσης με βάση την επιλογή του συγκεκριμένου τύπου - αυτόματα έχουμε να διαχειριστούμε τα πλεονεκτήματα ή τα μειονεκτήματα που έχει.

Σε περίπτωση που μας ενδιαφέρει, οι συγγραφείς του [7] παρουσιάζουν με εκτενέστερο τρόπο, διάφορες κατηγοριοποιήσεις και είδη drone τα οποία δεν εμπίπτουν στα πλαίσια αυτής της διπλωματικής και κυμαίνονται από smart dust, bio-drones, hybrid drones και άλλα πολλά.

### 1.1.2 Characteristics

Σε όποια από τις κατηγορίες και αν αντιστοιχεί ένα drone από την στιγμή που είναι ένα ιπτάμενο αντικείμενο, θα πρέπει να έχει την δυνατότητα να κινείται - φυσικά - στον αέρα. Στο *Figure 1.2* παρουσιάζονται στους 3 άξονες, τα 6 Degrees of Freedom (DoF) - τόσο *Transitional* όσο και *Rotational* - ενός UAV [10] καθώς και το όνομα που δίνεται στην κίνηση ανάλογα με τον άξονα που πραγματοποιείται.

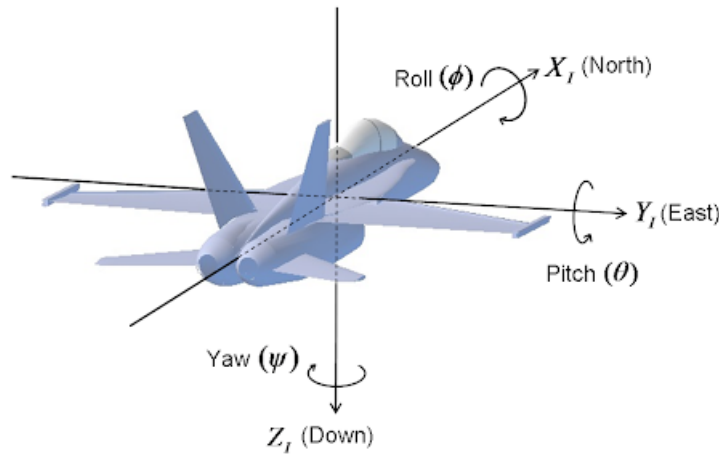


FIGURE 1.2: UAV principal axes (URL)

Ζούμε σε μία ψηφιακή εποχή, στην οποία ένα από τα πιο σημαντικά κατορθώματα είναι η ανάπτυξη των Integrated Circuits (ICs) & Micro-electro-mechanical Systems (MEMS) [11], πράγμα το οποίο έχει βοηθήσει - έξυπνες συσκευές γεμάτες

με αισθητήρες να βρίσκονται γύρω μας. Τέτοια τεχνολογικά επιτεύγματα όπως τα drones είναι συνεπώς εξοπλισμένα με Micro Controller Units (MCU) [12] ή Micro Processor Units (MPU) [13] για τον έλεγχο τους, ενώ δεν θα μπορούσαν να μην περιλαμβάνουν πλέον και ένα μεγάλο πλήθος και εύρος διαφορετικών τύπων sensors on-board. Με δύο από τους πιο σημαντικούς να είναι το Electronic Speed Control (ESC) [14] και το Inertial Measurement Unit (IMU) [15] - τα οποία χρησιμοποιούνται ώστε το drone να μπορεί να διατηρεί μία σταθερή και ελεγχόμενη πτήση [8]. Εκτός από αυτούς βέβαια, ένα drone μπορεί να διαθέτει Global Positioning System (GPS), camera για λήψη οπτικού υλικού ή ελέγχου μέσω First Person View (FPV), Obstacle avoidance sensors ή ακόμα και άλλους. Με κύριο περιορισμό τα αισθητήρια όργανα να βρίσκονται στο εύρος βάρους του payload, που μπορεί να μεταφέρει το drone.

Σε αυτό το section πραγματοποιήθηκε μία πρώτη οριοθέτηση του όρου drone, παρόλα αυτά δεν αναφέρθηκαν λόγοι ύπαρξης τους, καθώς και εφαρμογές τους. Η ύπαρξη των swarms είναι ουσιαστικά η κάλυψη των αναγκών από των individual drones σε μεγαλύτερη κλίμακα, για αυτό μερικές από τις εφαρμογές των drone - βρίσκονται στο *Section 1.2.2*.

## 1.2 Motivation

Ξεκινώντας από τα individual drones, αξίζει να σημειωθεί ότι το Πολυτεχνείο Κρήτης έχει ένα ενεργό ερευνητικό ιστορικό στον τομέα των αεροχημάτων. Το SenseLab στο οποίο υπεύθυνος καθηγητής είναι ο κύριος Παρτσινέβελος της Σχολής Μηχανικών Ορυκτών Πόρων να είναι ένα από αυτά - και μάλιστα με πολλαπλές διακρίσεις σε διεθνείς διαγωνισμούς [16] [17].

### 1.2.1 Nature

Στην φύση είναι αρκετά συχνή η ύπαρξη έμβιων οργανισμός που κινούνται σε ομάδες, με μερικά από αυτά ως παράδειγμα, τα σμήνη από πουλιά και έντομα, κοπάδια ψαριών ή και αγέλες ζώων. Σκοπός της συνεργασίας τους είναι κυρίως η προστασία από θηρευτές ή άλλοι σχετικοί με την επιβίωση. Αρκετές έρευνες έχουν γίνει εμπνευσμένες σε αυτό, ακόμα και στον τομέα των fleet of drones [18] [19] [20].

### 1.2.2 Swarms and Applications

Από μόνο του ένα UAV σε πολλές περιπτώσεις θα μπορέσει να φέρει εις πέραν την αποστολή που του έχει ανατεθεί. Πολύ εύκολα όμως μπορούν να δημιουργηθούν ζητήματα αξιοπιστίας, όταν ένα μεμονωμένο drone χρειάζεται ως παράδειγμα να χαρτογραφήσει σε μικρό χρονικό διάστημα ένα άγνωστο και μεγάλης έκτασης περιβάλλον. Όπως επίσης, εάν θα θέλαμε να διαθέτει πολλαπλούς αισθητήρες για να έχουμε πιο λεπτομερείς δεδομένα σε μία αποστολή, όμως αυτό να είναι αδύνατον διότι υπερβαίνουν το δυνατόν payload που μπορεί να μεταφέρει το drone. Η ακόμα, και για το ενδεχόμενο αποτυχίας της αποστολής, σε κάποιο πιθανό malfunction που θα πραγματοποιηθεί στο ιπτάμενου όχημα. Μπορούμε να πούμε λοιπόν· σε μία abstract σύγκριση με αυτή των ζώων, ότι ωθούμαστε για ανάλογους σκοπούς στην χρήση των swarms ώστε να ξεπεράσουμε αυτά τα προβλήματα [21].

Με τον όρο Swarm Robotics (SR) συχνά αναφερόμαστε στην μεθοδολογία που ακολουθούμε ώστε να συντονίζουμε πολλαπλές ανεξάρτητες ρομποτικές οντότητες να συμπεριφέρονται συνεργατικά ως ένα ενιαίο σύστημα [22]. Με τον όρο συνεργατικά, αυτό σημαίνει είτε να μπορούν να κινηθούν ως ομάδα [23] [24] [25], είτε να επικοινωνούν ώστε γνώση πληροφορίας την οποία έχει συλλέξει ένα από αυτά να μεταδίδεται και στα υπόλοιπα (π.χ. σε περίπτωση που θέλουμε να κάνουμε 3D reconstruction μίας περιοχής ή γνώση της θέσης από την οποία λαμβάνεται - με χρήση της κάμερας - υλικό είναι σημαντική για την δημιουργία του ψηφιακού μοντέλου [26]).

Σήμερα τα drones καθώς και τα fleet of drones χρησιμοποιούνται σε ένα μεγάλος εύρος εφαρμογών. Έχει γίνει αναφορά είδη από την αρχή του chapter ότι χρησιμοποιούνται για λόγους αναψυχής, με μερικά παραδείγματα, την διεκπεραίωση shows [27] [28] [29] και την καταγραφή οπτικού υλικού για δημιουργία ταινιών [30]. Άλλες πιο ζωτικής σημασίας χρήσεις τους, είναι σχετικές με environment mapping [31], police surveillance [32], natural disaster inspection [33], Search & Rescue (S&R) [34], light cargo transportation [35] και πολλές άλλες. Ακόμα και στρατιωτικές επιχειρήσεις χρησιμοποιούν drones. Συνήθως σε αυτές τις περιπτώσεις αναφερόμαστε στην χρήση Unmanned Combat Aerial Vehicle (UCAV) [36] [37] και πολλές από τις εφαρμογές στο συγκεκριμένο τομέα έχουν να κάνουν με Intelligence Surveillance, and Reconnaissance (ISR) [38].

Για περισσότερες εφαρμογές των drones/swarms [39] ή λεπτομέρειες σχετικά με αυτές, μπορούμε να ανατρέξουμε στην υφιστάμενη βιβλιογραφία [40] [41] [42].

## 1.3 Scientific Goals and Contributions

Όπως μπορεί να γίνει εύκολα αντιληπτό η γνώση της σχετικής θέσης ενός UAV σε σχέση με τα υπόλοιπα του σμήνους, είναι αρκετά σημαντική για ένα πολύ μεγάλο πλήθος εφαρμογών. Σκοπός της συγκεκριμένης διπλωματικής εργασίας, είναι να γίνει η απαραίτητη ερευνητική αναζήτηση, ώστε να επιτευχθεί ο σχεδιασμός καθώς και η υλοποίηση ενός ενσωματωμένου συστήματος - το οποίο να είναι σε θέση με όσο το δυνατόν πιο οικονομικό τρόπο - να υπολογίζει την θέση των μεμονωμένων αεροχημάτων όταν αυτά πετούν σε σχηματισμό.

## 1.4 Thesis Outline

- **Chapter 2 - Theoretical Background:** Σε αυτό το κεφάλαιο παρουσιάζεται το απαραίτητο μαθηματικό υπόβαθρο, καθώς και βασικές μεθοδολογίες εύρεσης τοποθεσία, που προέρχονται από τον τομέα των Wireless Sensor Networks (WSN).
- **Chapter 3 - Related Work:** Στην συγκεκριμένη ενότητα γίνεται μία συνοπτική αναφορά σε τεχνικές που χρησιμοποιούνται ήδη, σύμφωνα με την βιβλιογραφία, για την εύρεσης θέσης μέσα στα swarms.
- **Chapter 4 - Design Features and Implementation:** Σε αυτό το σημείο παρουσιάζεται η διαδικασία σχεδιασμού του ενσωματωμένου συστήματος που σχετίζεται αυτή η διπλωματική.
- **Chapter 5 - Experiments and Results:** Στο παρόν κεφάλαιο παρουσιάζονται οι απαραίτητοι έλεγχοι που έγιναν για την επαλήθευση της ορθής λειτουργίας του συστήματος.
- **Chapter 6 - Conclusions and Future Work:** Στο συγκεκριμένο κεφάλαιο παρουσιάζονται τα τελικά συμπεράσματα τα οποία βγήκαν από το σύνολο της διπλωματικής - καθώς και κάποιες από τις πιθανές μελλοντικές εξελίξεις της.





## Chapter 2

# Theoretical Background

"Let no one ignorant of  
geometry enter"

*Plato*

Ο χώρος των **WSN** έχει και αυτός τα τελευταία χρόνια υψηλό ερευνητικό ενδιαφέρον. Θα μπορούσε να πει κανείς - λόγω το ότι αποτελεί ένα πιο γενικό κλάδο - περισσότερο από ότι αυτό των drones. Συνεπώς θα μεταφερθούμε σε πρώτο επίπεδο στο πιο γενικό φάσμα, αυτό των **WSN** για να προσεγγίσουμε το localization problem.

Όταν μιλάμε για **WSN**, αναφερόμαστε σε αυτόνομα ηλεκτρονικά συστήματα, χωρικά διασκορπισμένα σε ένα πεδίο - τα οποία συχνά περιλαμβάνουν αισθητήρες και επικοινωνούν με τα γειτονικά τους ή **BS** για να μεταφέρουν πληροφορία [43] [44].

Το καθένα από αυτά τα ανεξάρτητα συστήματα ονομάζεται **Node**. Ενώ για το κάθε μεμονωμένο node μπορεί να έχουμε στην διάθεση μας location information ή όχι. Μία πρώτη σκέψη θα ήταν κάθε node ενός συστήματος αισθητήρων να περιλαμβάνει **GPS** ώστε να γνωρίζουμε την θέση του. Αυτό μπορεί γρήγορα να καταρριφθεί σαν σκέψη, αν αναλογιστούμε αρχικά ότι το Global Navigation Satellite System (**GNSS**) δεν είναι διαθέσιμο σε κάθε περιβάλλον λειτουργίας (π.χ. εσωτερικούς χώρους), όπως επίσης μπορεί να μην είναι δυνατή η χρήση του σε όλους τους κόμβους ενός συστήματος, λόγω περιορισμών όπως το κόστος, μέγεθος του node και energy consumption [44].

TABLE 2.1: Nodes' names definitions

Node name	Definition
Unknown/Free/Dumb/Non-anchors	Their position is unknown
Beacons/Anchors/Landmarks	Nodes with known location information
Settled	Initial unknown nodes with estimated position

Στην υπάρχουσα βιβλιογραφία [44] [45] [46] βρίσκουμε ότι nodes των οποίων η θέση είναι γνωστή ή άμεσα υπολογίσιμη, συχνά ονομάζονται **Beacons**. Πληροφορία σχετικά με την θέση αυτών των nodes είναι γνωστή, είτε γιατί έχουν τοποθετηθεί από εμάς σε προκαθορισμένες θέσεις, είτε μέσου ενός εξωτερικού συστήματος όπως το **GPS** [47]. Αντίθετα κόμβοι για τους οποίους δεν έχουμε αρχικά πληροφορία της θέσης τους, ονομάζονται **Non-anchors**. Άλλος ένας σημαντικός ορισμός, που θα πρέπει να αναφερθεί είναι ότι συχνά ονομάζουμε **Settle** nodes, αυτά τα οποία αρχικά δεν γνωρίζαμε την θέση τους αλλά στην συνέχεια την εκτιμήσαμε. Στο *Table 2.1* παρουσιάζονται συνοπτικά τα διάφορα ονόματα που έχουν δοθεί ανά καιρούς για το κάθε τύπο node.

Σκοπός ενός localization system είναι, με χρήση της γνώσης που έχουμε για τα beacon nodes να εκτιμήσουμε την θέση όσο περισσότερων unknown nodes ώστε να τα μετατρέψουμε σε settled nodes και η εκτίμηση της κάθε θέσης να είναι με όσο το δυνατόν μικρότερο error απόκλισης.



FIGURE 2.1: Localization System components based on [45] ([URL](#))

Οι συγγραφείς του [45] επιχειρούν να χωρίσουν ένα localization system, ώστε αυτό να αποτελείται από τρία διακριτά components, αυτή τη κατηγοριοποίηση την υιοθετούν κατά την έρευνα τους και οι ερευνητές του [48]. Πρώτο μπορεί να θεωρηθεί αυτό του **Distance/Angle Estimation**, που σκοπό έχει να υπολογίσει την γωνία ή απόσταση που έχουν δύο nodes του συστήματος μεταξύ τους. Η πληροφορία που θα παραχθεί από αυτό το component θα χρησιμοποιηθεί στα άλλα μέρη του συστήματος. Στην συνέχεια υπάρχει το **Position Computation**, δουλειά του οποίου είναι να υπολογίσει την θέση ενός node με βάση την γνώση που έχουμε για τα beacons και την πληροφορία που λάβαμε από το πρώτο component. Ενώ τέλος είναι το κύριο μέρος του συστήματος, με όνομα **Localization Algorithm** και ουσιαστικά είναι ο προκαθορισμένος τρόπος που θα ακολουθηθεί για να υπολογιστεί η θέση των unknown nodes με βάση όλες τις πληροφορίες που έχουμε. Στο *Figure 2.1* δίνεται η

απεικόνιση που έδωσαν οι συγγραφείς του [45] για να εξηγήσουν το παραπάνω. Ενώ στο Figure 2.2 έχει γίνει μία προσπάθεια να κατηγοριοποιηθούν τα κομμάτια καθώς και τεχνικές των Localization Systems, με βάση τα [44] [45] [46] και αναλύονται στην συνέχεια του κεφαλαίου.

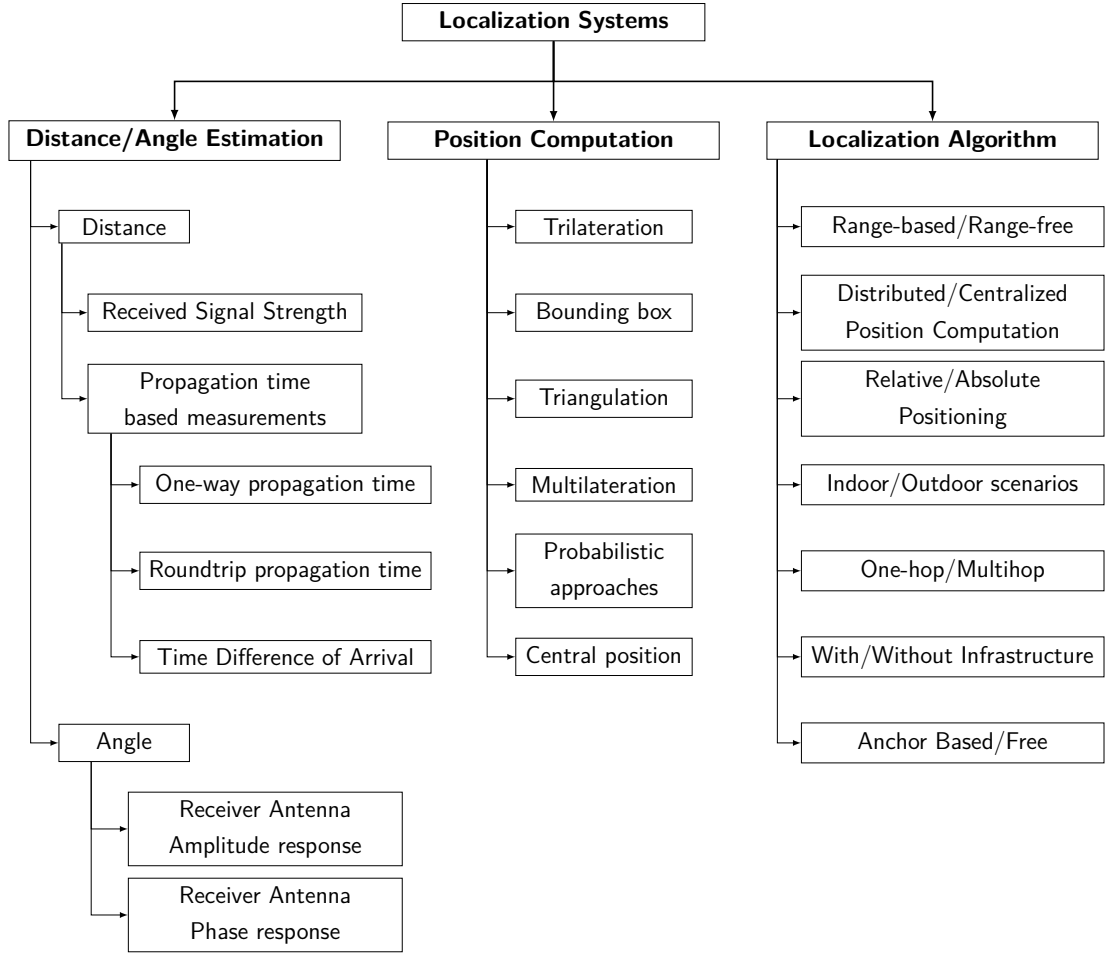


FIGURE 2.2: Localization System overview

Με βάση τον παραπάνω διαχωρισμό των Localization Systems σε τρία διακριτά μέρη, μπορούμε να καταλάβουμε ότι η απόκλιση της εκτίμησης του συνολικού συστήματος, εξαρτάται από τα σφάλματα του κάθε μεμονωμένου μέρους.

## 2.1 Distance/Angle Estimation

### 2.1.1 Distance

Θα ξεκινήσουμε με μία σύντομη ανάλυση των τεχνικών εκτίμησης της απόστασης μεταξύ δύο nodes που χρησιμοποιούνται ήδη.

#### Received Signal Strength

Η πρώτη τεχνική η οποία έχει χρησιμοποιηθεί για τον υπολογισμό απόστασης στα WSN, είναι αυτή με όνομα Received Signal Strength Indicator (RSSI) και έχει ως αρχή την χρήση της έντασης της ισχύς ενός σήματος που λαμβάνουμε στον δέκτη, ως τρόπο υπολογισμού της απόστασης του πομπού από αυτόν. Path loss ή path attenuation [49] ονομάζεται η μείωση της ισχύς ενός σήματος καθώς αυτό διαδίδεται. Στον ελεύθερο χώρο η λαμβανόμενη ισχύς  $P_r(d)$  που ανιχνεύει ο πομπός μπορεί να περιγραφεί από το μοντέλο του Free Space Path Loss (FSPL) [50], το οποίο προκύπτει μέσω της Friis transmission equation [46] [51] [52] - σχέση (2.1).

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2} \quad (2.1)$$

Όπου  $P_t$  είναι η ισχύς που στέλνει ο πομπός,  $G_t$  είναι το gain της κεραίας του πομπού,  $G_r$  το gain της κεραίας του δέκτη,  $\lambda$  είναι το μήκος κύματος του σήματος το οποίο μεταδίδουμε και  $d$  η απόσταση του πομπού από τον δέκτη. Αν θεωρήσουμε ότι τα  $G_t$ ,  $G_r$  και  $\lambda$  είναι μη μεταβλητές τιμές - με  $C_f = \frac{G_t G_r \lambda^2}{(4\pi)^2}$  - τότε μπορούμε να καταλήξουμε στην (2.2) [53].

$$P_r(d) = C_f \frac{P_t}{d^2} \quad (2.2)$$

Αυτό που μπορούμε να δούμε από την παραπάνω σχέση είναι ότι, ιδανικά στον ελεύθερο χώρο σε Line of Sight (LoS) μετάδοση - η ισχύς του σήματος που λαμβάνει ο πομπός εξαρτάται από το αντίστροφο τετράγωνο της απόστασης των δύο nodes. Η σχέση (2.2) συχνά αναφέρεται σε watt, όμως όταν μιλάμε για transmitting power αντί για την χρήση των watt, είναι αρκετά βολική η χρήση του dBm [54]. Για αυτό τον λόγω παρακάτω παραπείθονται τα conversion equations, από το ένα στο άλλο [51] [54].

$$\begin{cases} P[dBm] = 10 \cdot \log_{10} \left( \frac{P[mW]}{1[mW]} \right) \\ P[mW] = 10^{\frac{(P[dBm])}{10}} \end{cases} \quad (2.3)$$

Ενώ το *Figure 2.3* περιγράφει σχηματικά το ιδανικό μοντέλο της εξάρτησης της ισχύος με την αύξηση της απόστασης όταν χρησιμοποιούμε στον δέκτη μέτρηση σε dBm.

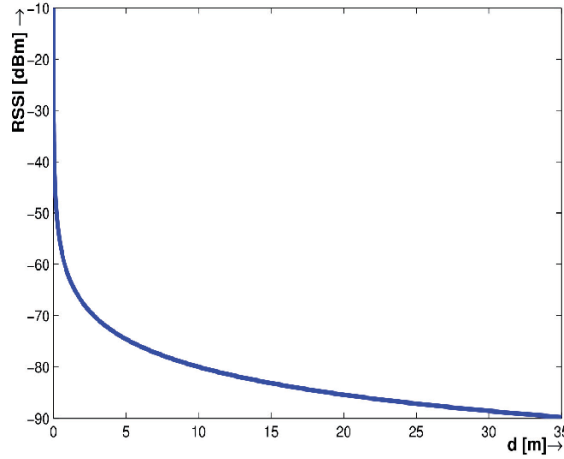


FIGURE 2.3: Ideal RSSI over distance based on [55]

Αν λοιπόν, μετράμε την ισχύ με την οποία λαμβάνουμε ένα σήμα, μπορούμε να είμαστε σε θέση να υπολογίσουμε την απόσταση που βρίσκεται ο πομπός από εμάς. Αυτή η μέθοδος παρόλο που είναι αρκετά δημοφιλής και οικονομική για τον υπολογισμό της απόστασης - λόγω του ότι δεν απαιτεί επιπλέον αισθητήρες - σε πραγματικές συνθήκες αντιμετωπίζει αρκετά προβλήματα, καθώς οι μετρήσεις μπορούν να επηρεαστούν από θόρυβο, ανακλάσεις του σήματος, διαθλάσεις, δυναμικά περιβάλλοντα ή εμπόδια σε αυτά, Non-Line of Sight (NLoS) μετάδοση, ή ακόμα και errors στο hardware [45] [55]. Σε ένα βαθμό μπορεί να βελτιωθεί η απόδοση με στατικό ή δυναμικό calibration του συστήματος, όμως μέχρι τώρα δεν χρησιμοποιείται για εκτίμηση απόστασης σε εφαρμογές όπου nodes έχουν μεγάλη απόσταση μεταξύ τους ή μας ενδιαφέρει να έχουμε μεγάλη ακρίβεια προσέγγισης της απόστασης [55].

## Propagation Time

Σε αυτήν την κατηγορία εκτίμησης απόστασης μεταξύ nodes - η οποία βασίζεται σε χρονικές μετρήσεις της διάδοσης του σήματος Time of Flight (ToF) - κατά κύριο

λόγο χρησιμοποιούνται δύο βασικές τεχνικές, η Time of Arrival (**ToA**) και η Time Difference of Arrival (**TDoA**) [45].

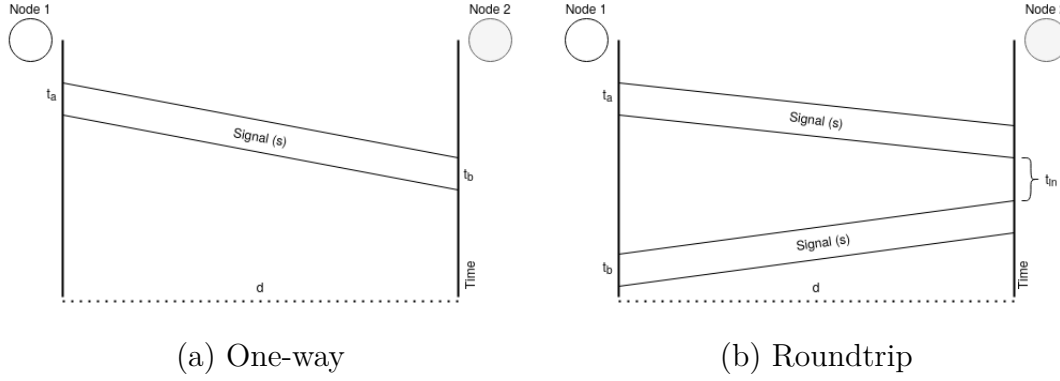


FIGURE 2.4: Time of Arrival cases

Θα ξεκινήσουμε με μία σύντομη περιγραφή της **ToA**. Από την κινηματική γνωρίζουμε την σχέση (2.4), η οποία συσχετίζει την ταχύτητα κίνησης ενός σώματος  $V$  ως το πηλίκο της μεταβολής της θέσης  $ds$  - που έκανε το σώμα, προς τον χρόνο  $dt$  που χρειάστηκε για να πραγματοποιηθεί η μεταβολή [56].

$$V = \frac{ds}{dt} \quad (2.4)$$

$$d = s(t_b - t_a) \quad (2.5)$$

Αξιοποιώντας την (2.4) ως αρχή, μπορούμε να καταλήξουμε στην (2.5) για να εκτιμήσουμε την απόσταση  $d$  που βρίσκονται δύο nodes μεταξύ τους, αν ένα κύμα κινείται με ταχύτητα  $s$  και χρειάστηκε χρόνο  $t$  για να μεταδοθεί από το ένα node στο άλλο. Στο Figure 2.4 (a) απεικονίζεται σχηματικά αυτό, όπου  $t = t_b - t_a$  με  $t_b$  η χρονική στιγμή που φτάνει το κύμα στο receiver και  $t_a$  η χρονική στιγμή η οποία ξεκινάει από τον transmitter. Σε περίπτωση που μιλάμε για Radio Frequency (**RF**) η ταχύτητα μετάδοσης του κύματος είναι ίση με την ταχύτητα μετάδοσης του φωτός  $c_o$ , το οποίο σε  $0.1\mu s$  διανύει περίπου  $30m$ . Με βάση αυτό, μπορούμε εύκολα να καταλάβουμε ότι για να έχουμε ακριβή αποτελέσματα είναι αρκετά σημαντικό τα clocks των δύο nodes να είναι απόλυτα συγχρονισμένα για να μην έχουμε error απόκλισης, πράγμα που απαιτεί να κάνουμε το συνολικό σύστημα αρκετά πιο πολύπλοκο σχεδιαστικά ώστε η απόκλιση μας να είναι σε ανεκτά σημεία για την εφαρμογή [45] [46].

Μπορούμε να το παρακάμψουμε αυτό, με το να γίνει η μέτρηση σε Round-Trip Time (**RTT**), Figure 2.4 (b). Σε αυτήν την περίπτωση το ένα node στέλνει ένα σήμα, και

μόλις το λάβει ένα γειτονικό node, απαντάει πίσω στο πρώτο. Με αυτόν τον τρόπο η μέτρηση του χρόνου εκκίνησης  $t_a$  και άφιξης  $t_b$  του σήματος γίνονται στο ίδιο node - άρα δεν χρειάζεται συγχρονισμός, και η πραγματική απόσταση είναι η μισή από αυτή που θα υπολογιστεί. Ο κύριος παράγοντας σφάλματος σε αυτή την μέθοδο, είναι ο υπολογισμός του χρόνου που χρειάστηκε το δεύτερο node για να διαχειριστεί το σήμα που έλαβε και να απαντήσει. Αυτό το internal delay  $t_{in}$  μπορεί να είναι είτε γνωστό από ένα a priori calibration, είτε μπορεί να μετριέται και να στέλνεται μαζί με το σήμα απάντησης - ώστε να αφαιρείται από τον χρόνο μετάδοσης του κύματος [46]. Με αυτά τα δεδομένα, η σχέση (2.6) περιγράφει τον τρόπο υπολογισμού της απόστασης  $d$  μεταξύ των δύο nodes.

$$d = \frac{s(t_b - t_a - t_{in})}{2} \quad (2.6)$$

Όσον αφορά την τεχνική **TDoA**, και σε αυτήν την περίπτωση υπάρχουν δύο παραλλαγές της [45] - όπου και οι δύο είναι βασισμένες στην αρχή το ότι δεν μας ενδιαφέρει η χρονική στιγμή που ξεκίνησε η αποστολή ενός σήματος, αλλά μόνο οι χρονική στιγμή που το λάβαμε.

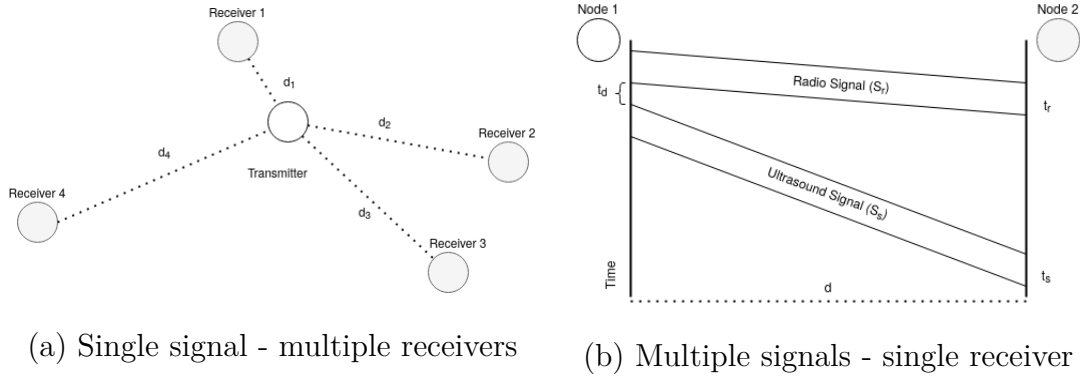


FIGURE 2.5: Time Difference of Arrival cases

Η πρώτη περίπτωση σχετίζεται με single signal και multiple receivers, παρουσιάζεται στο Figure 2.5 (a), χρησιμοποιείται συνήθως στα cellular network και απαιτεί την ύπαρξη τουλάχιστον 4 beacons για την απόκτηση location πληροφορίας ενός free node στον τρισδιάστατο χώρο. Υπολογίζει την χρονική διαφορά που έφτασε σε καθένα από τα beacons το σήμα που έστειλε το free node, για την εκτίμηση της θέσης του free node από το καθένα από αυτά. Σημαντικό σε αυτήν την περίπτωση είναι και εδώ, να είναι απόλυτα συγχρονισμένα τα beacons. Βασίζεται στις ιδιότητες των υπερβολικών καμπύλων και στο ότι με δεδομένη μία συγκεκριμένη χρονική διαφορά

του σήματος, το free node θα πρέπει να βρίσκεται κάπου πάνω σε μία υπερβολική καμπύλη όπως παρουσιάζεται από το *Figure 2.6* [57].

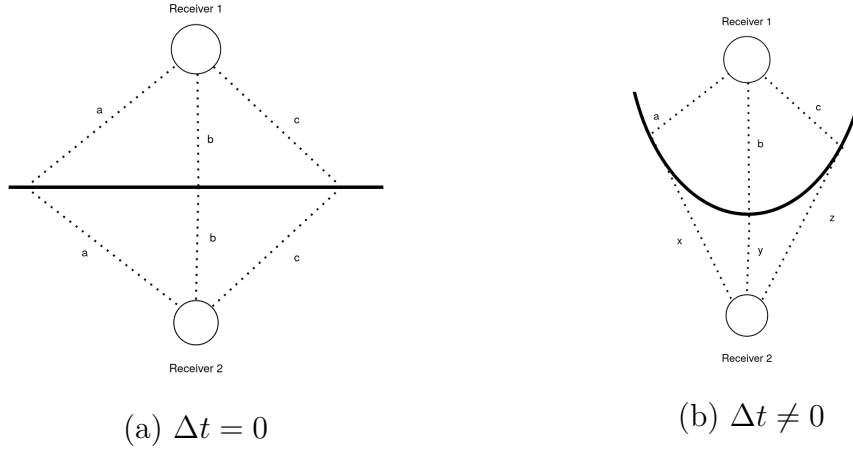


FIGURE 2.6: TDoA hyperbolas based on  $\Delta t$

$$\Delta t = a - a = 0$$

$$\Delta t = x - a$$

$$\Delta t = b - b = 0$$

$$\Delta t = y - b$$

$$\Delta t = c - c = 0$$

$$\Delta t = z - c$$

Μόνο από αυτόν τον υπολογισμό δεν δύναται να κάνουμε ακριβή εκτίμηση απόστασης, μπορούμε να συμπεράνουμε όμως τον γεωμετρικό τόπο στον οποίο βρίσκεται το free node. Στο *Section 2.2.4* - και σε επέκταση αυτού - παρουσιάζεται όμως, ο τρόπος με τον οποίο μπορούμε να συνδυάσουμε τέτοια πληροφορία από πολλαπλά beacon για να εκτιμήσουμε την ακριβή θέση του free node.

Η δεύτερη εκδοχή της χρήσης **TDoA** - η οποία είναι πιο συχνή στα **WSN** και μας ενδιαφέρει περισσότερο, χρησιμοποιεί multiple signals με single receiver και παρουσιάζεται στο *Figure 2.5* (b). Αρχή λειτουργίας έχει ότι ο transmitter θα στείλει πολλαπλά διαφορετικού είδους σήματα και ο δέκτης θα μετρήσει την χρονική διαφορά που τα έλαβε. Ως παράδειγμα μπορεί το ένα σήμα να είναι σε **RF** και να κινείται με ταχύτητα  $s_r = c_o$  και το άλλο ηχητικό με ταχύτητα  $s_s \approx 343m/s$ <sup>1</sup>. Αν  $t_r$  η χρονική στιγμή που λαμβάνει το **RF** σήμα,  $t_s$  το ηχητικό και  $t_d$  το delay που μεσολάβησε από την αποστολή του ενός στο άλλο, τότε μπορούμε να υπολογίσουμε την απόσταση μεταξύ των κόμβων από την εξίσωση (2.7) [45] [59].

<sup>1</sup>Για μετάδοση σε ξηρό αέρα υπό θερμοκρασία 20 °C [58]



$$d = (s_r - s_s)(t_s - t_r - t_d) \quad (2.7)$$

Θετικό σε αυτήν την μέθοδο είναι ότι το error μπορεί να είναι της τάξης των μερικών εκατοστών, όμως αρχικά απαιτεί επιπλέον εξοπλισμό στο node - ώστε να μπορεί να στείλει και να λάβει πολλαπλά είδη σήματος - πράγμα που μπορεί να το κάνει αντισυμβατικό ή αρκετά μεγαλύτερο σε διαστάσεις από το επιθυμητό. Όπως επίσης - και μάλιστα σημαντικότερο - η απόσταση η οποία μπορεί να χρησιμοποιηθεί, επηρεάζεται σε μεγάλο βαθμό από τα χαρακτηριστικά του δεύτερου σήματος. Ως παράδειγμα τα ηχητικά σήματα δεν μπορούν να μεταφερθούν σε μεγάλες αποστάσεις ή το ότι η ταχύτητα τους μπορεί να επηρεαστεί σημαντικά από περιβαλλοντολογικούς παράγοντες [44].<sup>2</sup>

### 2.1.2 Angle

Άλλη μία χρήσιμη μέτρηση η οποία μας ενδιαφέρει, είναι η εκτίμηση της γωνίας από την οποία λαμβάνουμε το σήμα ενός γειτονικού node σε σχέση με έναν άξονα αναφοράς. Ο άξονας αυτός μπορεί να είναι είτε κοινός για όλα τα nodes (π.χ. ως προς το βόρειο γεωγραφικό πόλο), είτε μπορεί να είναι για το κάθε node ξεχωριστός, ως παράδειγμα με βάση τον προσανατολισμό του ίδιου του node ή με βάση την γωνία λήψης ενός επιπλέον σήματος [45]. Την πληροφορία αυτή την βρίσκουμε στην βιβλιογραφία να ονομάζεται Angle of Arrival (AoA) ή Direction of Arrival (DoA).

$$s(t) = A \sin(2\pi ft + \phi) \quad (2.8)$$

Γνωρίζουμε ότι ένα σήμα καθορίζεται πλήρως από τρεις παραμέτρους, το πλάτος του  $A$ , την συχνότητα του  $f$  καθώς επίσης και από την φάση του  $\phi$  - σχέση (2.8). Για τον υπολογισμό της γωνίας AoA μπορούν να χρησιμοποιηθούν τεχνικές με γνώμονα τα παραπάνω.

<sup>2</sup>Η συγκεκριμένη μέθοδος είναι διαισθητικά παρόμοια, με αυτό που θα κάναμε για να εκτιμήσουμε την απόσταση μίας αστραπής από εμάς, μετρώντας το χρονικό διάστημα μέχρι να ακούσουμε την βροντή.

## Amplitude response

Ο όρος *Beamforming* [60] χρησιμοποιείται για να περιγράψει το pattern της ευαισθησίας του σήματος που στέλνει/λαμβάνει μία *directional antenna* [61]. Το pattern αυτού του τύπου κεραίας, παρουσιάζεται στο *Figure 2.7* [46] και μπορούμε να χρησιμοποιήσουμε αυτό το χαρακτηριστικό για τον υπολογισμό της AoA. Εάν χρησιμοποιούμε *directional antenna* στον receiver και μπορούμε είτε με μηχανικό είτε με ηλεκτρονικό τρόπο να στρέψουμε την κεραία σε διάφορες κατευθύνσεις, τότε μπορούμε μετρώντας το πλάτος του σήματος που λαμβάνουμε, να εκτιμήσουμε και την κατεύθυνση του - όπου ιδανικά βρίσκεται εκεί από όπου έχουμε την μέγιστη τιμή του amplitude του κύματος. Σημαντικές παράμετροι για την εκτίμηση της γωνίας σε αυτήν την περίπτωση, είναι επίσης η ευαισθησία του δέκτη καθώς και το width του beam της. Λανθασμένη εκτίμηση μπορεί να προκύψει, αν για κάποιο λόγο το πλάτος του σήματος που λαμβάνουμε δεν είναι σταθερό. Σε αυτή την περίπτωση, μία λύση είναι να χρησιμοποιήσουμε μία δεύτερη - μη κινούμενη - *omnidirectional* κεραία και να κανονικοποιήσουμε τις μετρήσεις της *directional* με βάση τις μετρήσεις της δεύτερης [46].

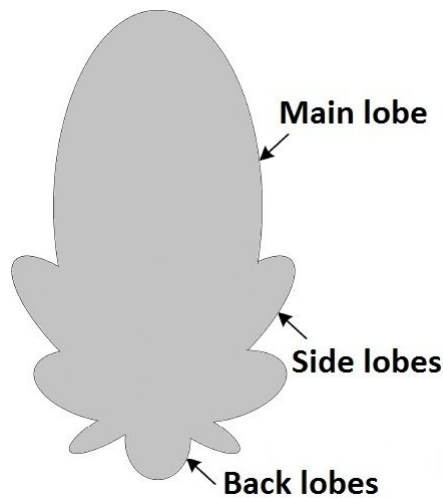


FIGURE 2.7: Typical anisotropic antenna pattern based on [46]  
(URL)

## Phase response

Άλλος τρόπος υπολογισμού του AoA είναι με την αξιοποίηση της πληροφορία για την φάση που έχουμε για ένα σήμα. Σε αυτήν την περίπτωση χρειαζόμαστε ή κεραίες αρκετά μεγαλύτερες από το μήκος κύματος του σήματος που στέλνουμε, ή *antenna arrays* [46].

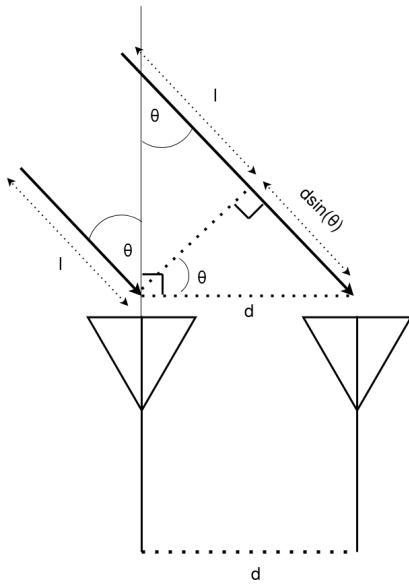
Για τα κύματα γνωρίζουμε ότι η διαφορά φάσης  $\Delta\phi$  δύο σημείων που απέχουν μεταξύ τους  $\Delta\chi$  μπορεί να υπολογιστεί από την σχέση (2.9) [62].

$$\Delta\phi = 2\pi \frac{\Delta\chi}{\lambda} = 2\pi \frac{\Delta t}{T} \quad (2.9)$$

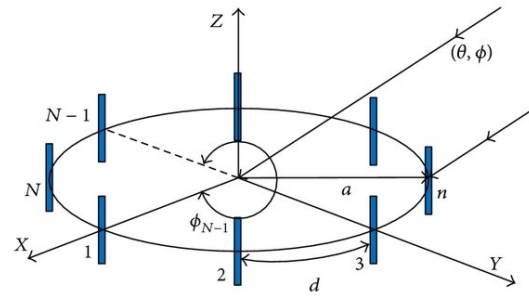
Συνεπώς, αν σχεδιάσουμε ένα antenna array, όπου η κάθε κεραία απέχει από την γειτονική της σε συγκεκριμένη απόσταση  $d$  και μετρώντας το phase difference του σήματος που λαμβάνουμε μεταξύ τους, τότε μπορούμε να υπολογίσουμε την γωνία πρόσληψης της ακτινοβολίας μέσω της σχέσης (2.10) [46] [63]. Παράδειγμα αυτού βρίσκεται στο *Figure 2.8* (a). Σε αυτήν την περίπτωση χρησιμοποιούμε *omnidirectional* κεραίες και ο υπολογισμός της διαφοράς φάσης μπορεί να γίνει μέσω μετρήσεων TDoA.

$$\begin{aligned} \phi &= \frac{2\pi}{\lambda} d \sin(\theta) \Rightarrow \\ \theta &= \sin^{-1} \left( \frac{\lambda \phi}{2\pi d} \right) \end{aligned} \quad (2.10)$$

Δεν υπάρχουν όμως μόνο τα linear antenna arrays, μπορούμε να τα σχεδιάσουμε σε διάφορες διατάξεις, παράδειγμα μίας εξ' αυτών βρίσκεται στο *Figure 2.8* (b).



(a) Linear Two Antenna Array



(b) Circular Antenna Array (URL)

FIGURE 2.8: Antenna Arrays

Η μέθοδος με την αξιοποίηση της σχέσης (2.10) λειτουργεί αρκετά καλά όταν έχουμε υψηλό Signal to Noise Ratio (SNR), αλλά μπορεί να έχει λανθασμένα αποτελέσματα όταν έχουμε διασυμβολικές παρεμβολές ή multipath signals. Επειδή στην πράξη εμφανίζονται και τα δύο, υπάρχει αρκετά μεγάλο επιστημονικό ενδιαφέρον να αντιμετωπιστεί αυτός ο περιορισμός. Αρκετές τεχνικές έχουν προταθεί οι οποίες στρέφονται γύρω από την λογική του Maximum Likelihood (ML) για τον υπολογισμό τελικά της γωνίας [46].

## 2.2 Position Computation

Αν έχουμε δύο nodes, οι μετρήσεις που λάβαμε από το Section 2.1 είναι αρκετές για να γνωρίζουμε την θέση του καθενός, συνεπώς ενδιαφέρον βρίσκεται στην ύπαρξη τριών ή παραπάνω nodes σε ένα σύστημα. Με βάση τις πληροφορίες που έχουμε συλλέξει - μέσω των τεχνικών που περιγράφηκαν στο προηγούμενο κεφάλαιο - θα προσπαθήσουμε πλέον να εκτιμήσουμε την θέση ενός node. Στην συνέχεια αυτού του section περιγράφονται μέθοδοι οι οποίοι μπορούν να χρησιμοποιηθούν για να το επιτύχουν αυτό. Κύρια διαφορά τους είναι η απόδοση που μπορεί να έχουν, η οποία όμως σχετίζεται με την αύξηση της πολυπλοκότητας στους υπολογισμούς που θα χρειαστούν καθώς επίσης και ποιες από τις παραπάνω πληροφορίες θα εκμεταλλευτούν.

Για την εύρεση της θέσης ενός unknown node στο τρισδιάστατο χώρο ( $\mathfrak{R}^3$ ) το Minimal Required number of Sensors (MRS) είναι τέσσερα nodes. Στις παρακάτω περιγραφές όμως, και χωρίς βλάβη της γενικότητας θα χρησιμοποιηθούν τρία για την απλούστευση της περιγραφής, με την παραδοχή ότι μας ενδιαφέρει η δισδιάστατη ανάλυση ( $\mathfrak{R}^2$ ).

### 2.2.1 Trilateration

Η συγκεκριμένη μέθοδος είναι ίσως η πιο απλή διαισθητικά, βασίζεται στην γεωμετρία κύκλων, είναι αυτή που χρησιμοποιείται από τα GPS για τον υπολογισμό θέσης [64], ενώ αξιοποιεί μόνο πληροφορία απόστασης και όχι γωνίας [65]. Η εξίσωση ενός κύκλου από την γεωμετρία γνωρίζουμε ότι περιγράφεται από την εξίσωση (2.11) όπου  $(x_i, y_i)$  οι συντεταγμένες του κέντρου του κύκλου με ακτίνα  $r_i$ .

$$(x - x_i)^2 + (y - y_i)^2 = r_i^2 \quad (2.11)$$

Εάν χρησιμοποιούμε *omnidirectional* κεραίες [66] ή στην γενικότερη περίπτωση το μοντέλο της *isotropic antenna* [67] και υπολογίσουμε την απόσταση ενός beacon από το node στο οποίο θέλουμε να υπολογίσουμε την θέση του - τότε μπορούμε να συμπεράνουμε ότι το free node είναι κάπου πάνω στην περιφέρεια ενός κύκλου, με κέντρο το beacon και ακτίνα την απόσταση μεταξύ του beacon και του free node. Επαναλαμβάνοντας αυτό για ακόμα δύο beacons, τελικά το free node στο δισδιάστατο επίπεδο ( $\mathcal{R}^2$ ) θα πρέπει να βρίσκεται στην τομή των τριών κύκλων, πράγμα που γραφικά απεικονίζεται στο *Figure 2.9* (a) [68].

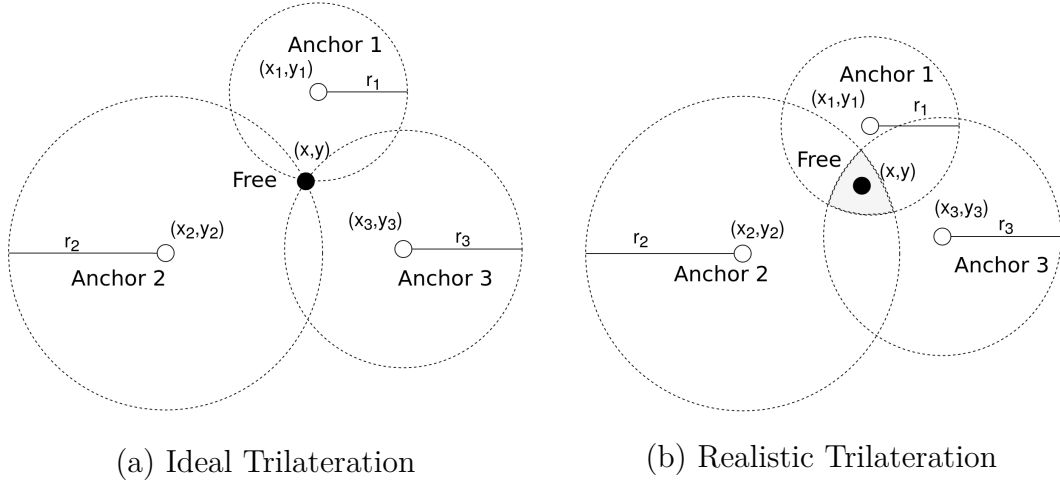


FIGURE 2.9: Trilateration examples

$$x^2 - 2x_1x + x_1^2 + y^2 - 2y_1y + y_1^2 = r_1^2 \quad (2.12)$$

$$x^2 - 2x_2x + x_2^2 + y^2 - 2y_2y + y_2^2 = r_2^2 \quad (2.13)$$

$$x^2 - 2x_3x + x_3^2 + y^2 - 2y_3y + y_3^2 = r_3^2 \quad (2.14)$$

Οι εξισώσεις (2.12), (2.13) και (2.14) περιγράφουν πλήρως τους κύκλους του κάθε node από το παράδειγμα του *Figure 2.9* (a), όπου  $(x_i, y_i)$  το κέντρο του κύκλου,  $r_i$  η ακτίνα του - για κάθε beacon  $i = 1, 2, 3$  και τελικά  $(x, y)$  οι συντεταγμένες του free nodes τις οποίες και ψάχνουμε. Ένας τρόπος να υπολογίσουμε τις συντεταγμένες αυτές είναι να αφαιρέσουμε από την (2.13) την (2.12) και όμοια από την (2.14) την (2.13) ώστε να καταλήξουμε στις παρακάτω δύο εξισώσεις [69] [70].

$$\begin{aligned} (-2x_1 + 2x_2)x + (-2y_1 + 2y_2)y &= r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \\ (-2x_2 + 2x_3)x + (-2y_2 + 2y_3)y &= r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \end{aligned}$$

Ο λόγος που κάναμε το παραπάνω βήμα είναι διότι πλέον έχουμε ένα σύστημα με δύο εξισώσεις και δύο αγνώστους, οπότε μπορούμε εύκολα να θεωρήσουμε τους παρακάτω πίνακες.

$$A = \begin{bmatrix} -2x_1 + 2x_2 & -2y_1 + 2y_2 \\ -2x_2 + 2x_3 & -2y_2 + 2y_3 \end{bmatrix} \quad X = \begin{bmatrix} x \\ y \end{bmatrix} \quad B = \begin{bmatrix} r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \\ r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \end{bmatrix}$$

Με την χρήση των οποίων καταλήγουμε ότι έχουμε να λύσουμε το γραμμικό σύστημα πινάκων που περιγράφεται από την εξίσωση (2.15) για τον υπολογισμό των συντεταγμένων  $(x, y)$  του free node οι οποίες μας ενδιαφέρουν.

$$AX = B \quad (2.15)$$

Όπως αναφέρθηκε και στις προηγούμενες ενότητες, ο υπολογισμός της απόστασης πολύ πιθανόν να περιλαμβάνει μια μικρή απόκλιση  $\hat{r}_i = r_i - \varepsilon$ , με το  $\varepsilon$  συχνά να θεωρείται μία ανεξάρτητη κανονική τυχαία μεταβλητή με μηδενικό μέσο. Αυτό σημαίνει ότι τότε οι κύκλοι δεν έχουν ένα κοινό σημείο τομής, αλλά το free node βρίσκεται κάπου μέσα στο χωρίο επικάλυψης των τριών κύκλων, σχηματικά αυτό παρουσιάζεται στο *Figure 2.9 (b)* και σε αυτήν την περίπτωση καταλήγουμε σε ένα μη πεπερασμένο πλήθος λύσεων, με τη συνάρτηση του κάθε κύκλου να περιγράφεται από την εξίσωση (2.16) [45].

$$(x - x_i)^2 + (y - y_i)^2 = r_i^2 - e \quad (2.16)$$

Το αρνητικό με αυτήν την μέθοδο, είναι η ανάγκη πραγματοποίησης floating point operations για τον υπολογισμό των συντεταγμένων  $(x, y)$  σε πραγματικές συνθήκες - όπου το πλήθος των οποίων εξαρτάται από τον τρόπο που θα επιλέξουμε να επιλύσουμε το σύστημα. Μία από τις μεθόδους επίλυσης της γραμμικής εξίσωσης είναι μέσω του least square method, όπου τότε το πλήθος των floating point operations

που απαιτούνται είναι  $(m + \frac{n}{3}) \cdot n^2$ , με  $m$  τον αριθμό των αγνώστων και  $n$  τον αριθμό των δοθέντων εξισώσεων [45].

### 2.2.2 Bounding Box

Σε αυτήν την μέθοδο χρησιμοποιούνται τετράγωνα αντί για κύκλους του Trilateration, ενώ και εδώ θεωρούμε  $(x_i, y_i)$  τις συντεταγμένες των beacon και  $d_i$  η απόσταση που έχουμε υπολογίσει από το free node - για κάθε beacon  $i$ . Δημιουργούμε τετράγωνα μήκος πλευράς  $2d_i$  με κέντρο το κέντρο του beacon και συντεταγμένες  $(x_i - d_i, y_i - d_i)$  &  $(x_i + d_i, y_i + d_i)$ . Θετικό πλέον είναι ότι δεν χρειάζεται να κάνουμε floating point operations για τον υπολογισμό του χωρίου τομής - αλλά μπορούμε να το υπολογίσουμε με απλή γεωμετρία. Αφού έχουμε υπολογίσει το χωρίο τομής των τετραγώνων μπορούμε να θεωρήσουμε ότι στο κέντρο του βρίσκεται το free node. Παράδειγμα αυτής της μεθόδου βρίσκεται στο *Figure 2.10*. Η συγκεκριμένη μέθοδος μπορεί να είναι ευκολότερη υπολογιστικά και να απαιτεί λιγότερα processor resources από το Trilateration, όμως ταυτόχρονα προκύπτει και μεγαλύτερο σφάλμα απόκλισης [45].

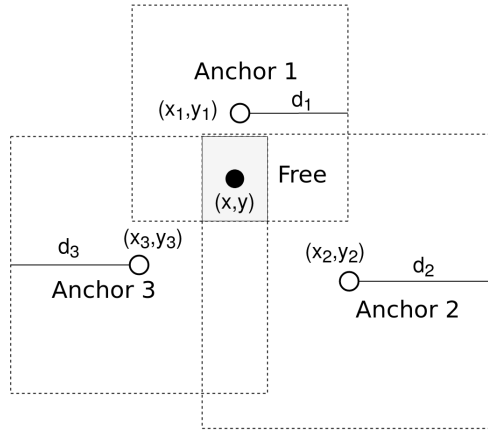


FIGURE 2.10: Bounding Box example

### 2.2.3 Triangulation

Σε αντίθεση με τις παραπάνω μεθόδους, η τεχνική Triangulation εκτιμάει την θέση του node που μας ενδιαφέρει, χρησιμοποιώντας γνώση που έχουμε για γωνίες και όχι εκτίμηση της απόστασης. Ένα απλούστερο παράδειγμα παρουσιάζεται στο *Figure 2.11*, όπου σε αυτό το παράδειγμα από την τριγωνομετρία, μπορούμε εύκολα να συμπεράνουμε ότι για τις γωνίες  $\theta_1$  και  $\theta_2$  ισχύουν οι σχέσεις (2.17).

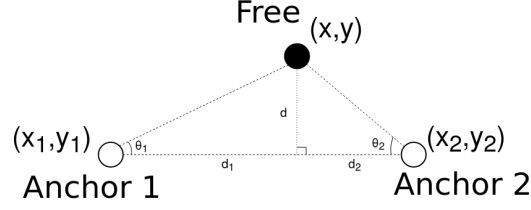


FIGURE 2.11: Principles of Triangulation

$$\theta_1 = \frac{d}{d_1} = \tan^{-1} \left( \frac{y_1 - y}{x_1 - x} \right) \quad \theta_2 = \frac{d}{d_2} = \tan^{-1} \left( \frac{y_2 - y}{x_2 - x} \right) \quad (2.17)$$

Από τις σχέσεις (2.17) μπορούμε να καταλήξουμε στις σχέσεις (2.18) [71], μέσω των οποίων τελικά για δεδομένες θέσεις των beacon - γνωστά δηλαδή τα  $(x_i, y_i)$  για κάθε beacon  $i$  - και με βάση την μέτρηση των γωνιών  $\theta_1$  και  $\theta_2$ , να υπολογίσουμε την θέση του free node  $(x, y)$ .

$$x = \frac{y_2 - y_1 - x_2 \tan \theta_2 + x_1 \tan \theta_1}{\tan \theta_1 - \tan \theta_2} \quad (2.18)$$

$$y = x \tan \theta_1 + y_1 - x_1 \tan \theta_1 = x \tan \theta_2 + y_2 - x_2 \tan \theta_2$$

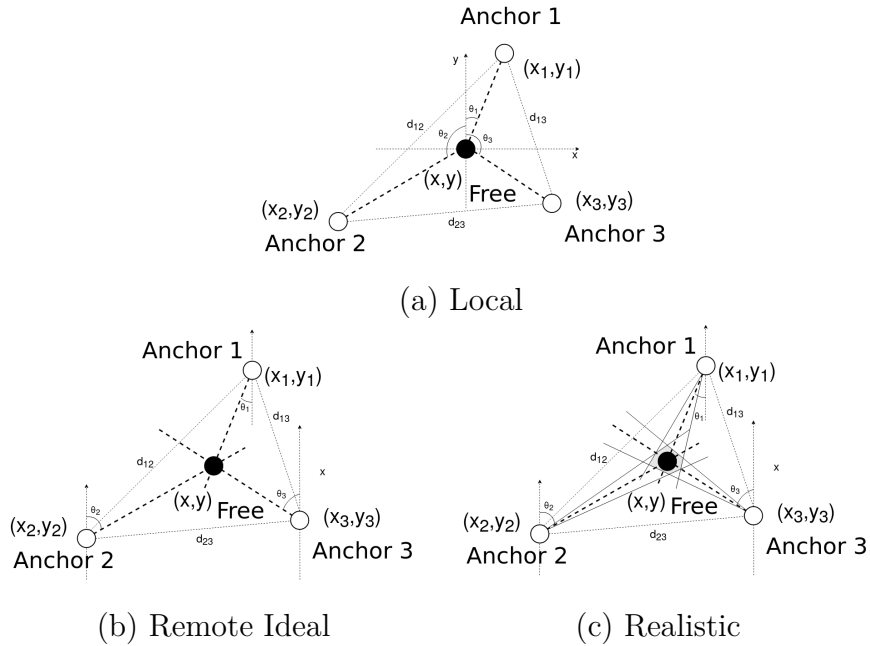


FIGURE 2.12: Triangulation examples



Αυτή η τεχνική εκτίμησης της θέσης του node, μπορεί να γίνει είτε από τοπικές μετρήσεις του ίδιου του node - *Figure 2.12* (a) - είτε από μετρήσεις γωνιών που κάνουν τα anchors - *Figure 2.12* (b) [45]. Φυσικά σε μία πραγματική εφαρμογή, όπως έχει αναφερθεί παραπάνω, οι μετρήσεις παρουσιάζουν αποκλίσεις από τα ιδανικά μοντέλα. Συνεπώς το *Figure 2.12* (c) παρουσιάζει σε πιο ρεαλιστικά επίπεδα - συνυπολογίζοντας την ύπαρξη των αποκλίσεων - το χωρίο στο οποίο με μεγαλύτερη πιθανότητα βρίσκεται το node.

### 2.2.4 Multilateration

Η μέθοδος του Multilateration<sup>3</sup> (MLAT), έχει ως βάση την γεωμετρία των υπερβολικών καμπύλων και συχνά μπορεί να βρεθεί επίσης και ως hyperbolic positioning στην βιβλιογραφία [72] [73] [74]. Στο *Section 2.1.1* έγινε αναφορά ότι η χρήση του TDoA βασίζεται στις υπερβολικές καμπύλες, συνεπώς μπορούμε εύκολα να χρησιμοποιήσουμε της μετρήσεις αυτές για τον υπολογισμό της θέσης του free node.

Η γενική ιδέα σε αυτήν την περίπτωση περιγράφεται στην εξίσωση (2.19) [46] [75] - και αυτό που θέλουμε, είναι να υπολογίσουμε την θέση του  $k_t$  (free node) γνωρίζοντας για κάθε δύο διαφορετικά  $k_i$  και  $k_j$  beacon nodes την χρονική στιγμή  $t_i$  και  $t_j$  που έφτασε το σήμα στο καθένα - εάν αυτό κινούταν με ταχύτητα  $s$  και το  $\|\cdot\|$  συμβολίζει την ευκλείδεια απόσταση μεταξύ τους.

$$\Delta t_{ij} \triangleq t_i - t_j = \frac{1}{s} (\|k_i - k_t\| - \|k_j - k_t\|), \quad i \neq j \quad (2.19)$$

Αυτό που αναπαριστά ουσιαστικά η σχέση (2.19) δεν είναι τίποτα άλλο από την προσπάθεια εύρεσης του σημείου τομής των καμπύλων. Παράδειγμα αυτής της μεθόδου βρίσκεται στο *Figure 2.13*.

<sup>3</sup>Κάποιες φορές ο όρος αυτός χρησιμοποιείται επίσης σε αναφορά της χρήσης του Trilateration ή του Triangulation με πάνω από 3 beacons [45] [71].

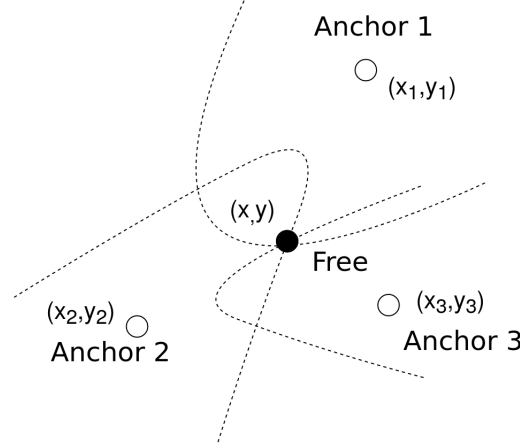


FIGURE 2.13: Multilateration example

### 2.2.5 Probabilistic approaches

Το γεγονός ότι οι μετρήσεις απόστασης και γωνίας σε πραγματικές συνθήκες εμπεριέχουν σφάλματα, έχει ωθείσει στην έρευνα probabilistic μεθόδων εκτίμησης τοποθεσίας των free nodes. Αυτού του τύπου οι προσεγγίσεις λαμβάνουν υπόψη τα μετρικά σφάλματα και συχνά τα μοντελοποιούν ως κανονικές τυχαίες μεταβλητές. Το μεγάλο μειονέκτημα σε αυτήν την μέθοδο είναι τα μεγάλα υπολογιστικά και χωρικά - για την αποθήκευση των δεδομένων - κόστη [45].

Στο Table 2.2 γίνεται σύνοψη των μεθόδων εκτίμησης απόστασης που αναλύθηκαν παραπάνω, μαζί με τις μεθόδους εκτίμησης θέσης που συχνά συνδυάζονται, καθώς και θετικά/αρνητικά όπως και συχνές τους εφαρμογές.

TABLE 2.2: Distance/angle estimation techniques and position Computation summary [76]

Method	Position Computation	Advantages	Disadvantages	Applications
RSSI	Trila/Multilateration, Bounding box	Simple, inexpensive, no need for extra hardware, synchronization not needed	Multipath interference, NLoS and environment can affect readings	In sort-distance with LoS applications and low accuracy needed (e.g. indoor)
ToA	Trila/Multilateration, Bounding box	High accuracy	Need to precisely measure time, LoS is normally assumed, more sophisticated design	Common in cellular networks or systems assisted by GPS
TDoA	Multilateration	High accuracy	Need for extra hardware or precisely synchronized nodes	Common in WSN or systems assisted by GPS
AoA	Triangulation	MRS is two nodes (Ideal $\mathbb{R}^2$ ), synchronization not needed	LoS is normally assumed, smart antenna design, multipath can affect readings	Common in radar scenarios

## 2.3 Localization Algorithm

Στις προηγούμενες ενότητες αναφέρθηκαν κάποιες από τις βασικές τεχνικές που χρησιμοποιούνται, προκειμένου να εκτιμηθεί η θέση ενός node στον χώρο. Σε ένα πραγματικό σύστημα όμως, πρέπει να συνυπολογιστούν και άλλοι παράγοντες πέρα από - απλά - τον υπολογισμό απόστασης/γωνίας για να καταλήξουμε να αποκτήσουμε location information. Για αυτό το λόγο, στο παρόν *Section* παρουσιάζονται επιπλέον πτυχές - του συστήματος - που πρέπει να λάβουμε υπόψιν· καθώς και αναφορά αλγόριθμων από την υπάρχουσα βιβλιογραφία, με τους οποίους μπορούμε να καλύψουμε αυτές τις ανάγκες.

Αφού έχουμε δει τα παραπάνω, αρχικά ας ορίσουμε το localization problem με ένα πιο αυστηρό μαθηματικό φορμαλισμό. Έστω ότι έχουμε στην διάθεση μας  $n$  αριθμό από nodes, και για λόγους απλότητας· συμμετρικά και όμοια δίκτυα επικοινωνίας - με εμβέλεια  $r$  - για κάθε node. Συνεπώς, ένα node  $u$  επικοινωνεί με το  $v$  αν και μόνο αν, και το  $v$  μπορεί να επικοινωνήσει πίσω στο  $u$ , τα οποία είναι διασκορπισμένα σε ένα διδιάστατο τετραγωνικό πεδίο  $Q = [0, s] \times [0, s]$ . Τότε μπορούμε να μοντελοποιήσουμε το δίκτυο από nodes, ως ένα μη κατευθυνόμενο γράφο  $\mathcal{G} = (V, E)$  με τα παρακάτω χαρακτηριστικά [45].

- $V = \{v_1, v_2, \dots, v_n\}$  το σύνολο των nodes, όπου έχει την έννοια των κορυφών του γράφου, με  $|V| = n$ .
- $\langle i, j \rangle \in E$  εάν το  $u_i$  μπορεί να επικοινωνήσει με το  $v_i$ . Πράγμα που σημαίνει - με βάση τα παραπάνω - ότι η ευκλείδεια απόσταση τους· είναι μικρότερη από  $r$  και έχει την έννοια των ακμών του γράφου, με  $|E| = m$ .
- $0 \leq w(e) \leq r$  με  $e = \langle i, j \rangle$ , να είναι το βάρος της κάθε ακμής και να χρησιμοποιείται για την έννοια της απόστασης  $d_{ij}$  μεταξύ του node  $u_i$  με το  $v_i$ .

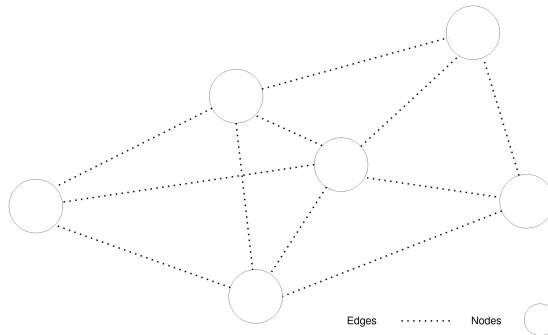


FIGURE 2.14: Graph example

### 2.3.1 One-hop vs Multi-hop

Η πρώτη κατηγοριοποίηση στην οποία θα αναφερθούμε - για αυτό το δίκτυο γράφου - προκειμένου να εκτιμήσουμε την θέση των nodes, είναι από ποιους κόμβους τελικά θα αξιοποιήσουμε πληροφορία. Στους αλγόριθμους **One-hop** αξιοποιείται πληροφορία, μόνο από άμεσους γείτονες των nodes' προκειμένου να υπολογίσουμε την θέση του κόμβου στο χώρο. Αντίθετα στις τεχνικές **Multi-hop**, χρησιμοποιούμε και πληροφορία που λαμβάνουμε με έμμεσο τρόπο' από τους γείτονες των γειτόνων μας - προκειμένου να αποφανθούμε για την θέση ενός node.

### 2.3.2 Anchor-based vs Anchor-free

Ένας ακόμα σημαντικός διαχωρισμός είναι η ύπαρξη ή όχι τελικά στο δίκτυο από anchor nodes. Καθώς, υπάρχουν εφαρμογές στις οποίες μπορεί να μην χρειάζεται ή να μην έχουμε δυνατότητα να τοποθετηθούν beacons - ακόμα και αν αναφερόμαστε σε mobile beacons - οπότε τότε θα χρησιμοποιήσουμε **Anchor-free** αλγορίθμους. Αν υπάρχει η δυνατότητα χρήσης τους, τότε ενδιαφερόμαστε για **Anchor-based**.

### 2.3.3 Relative vs Absolute Positioning

Ανάλογα με τις προδιαγραφές της εφαρμογής που θέλουμε να καλύψουμε, θα πρέπει να χρησιμοποιήσουμε ανάλογες εκτιμήσεις θέσεις. Υπάρχουν εφαρμογές, όπως για παράδειγμα - εκτίμηση θέσεις των nodes ενός κινούμενου swarm' για χαρτογράφηση σε άγνωστο terrain. Σε αυτού, του τύπου τις εφαρμογές μας ενδιαφέρει η απόλυτη θέση στον χώρο' οπότε θέλουμε μεθοδολογίες για **Absolute Positioning**, ενώ υπάρχουν άλλες εφαρμογές στις οποίες μας ενδιαφέρει μόνο πληροφορία θέσεις σε σχέση με τα υπόλοιπα στοιχεία του περιβάλλοντος της εφαρμογής, οπότε χρειαζόμαστε **Relative Positioning**.

### 2.3.4 Indoor vs Outdoor Scenarios

Κάτι ακόμα που πρέπει να προσέξουμε είναι το περιβάλλον στο οποίο θα βρίσκεται η εφαρμογή, καθώς ανάλογα με το περιβάλλον' έχουμε να διαχειριστούμε και τις δυσκολίες που μας παρουσιάζει - εμπόδια, ανακλάσεις σημάτων, κλπ. Με δύο κύριους διαχωρισμούς, τα **Indoor Scenarios** και τα **Outdoor Scenarios**.

### 2.3.5 Distributed vs Centralized Position Computation

Πρέπει να αναφερθεί επίσης, ότι είναι πολύ σημαντικό να είναι ξεκάθαρο το σημείο στο οποίο θα γίνεται η επεξεργασία της πληροφορίας. Υπάρχουν δύο ενδεχόμενα. Το πρώτο είναι, ένα **BS** να αντιλαμβάνεται και να υπολογίζει την θέση των nodes, τα οποία έπειτα θα ενημερώνονται μέσω αυτού· για την θέση τους - οπότε αναφερόμαστε για **Centralized** τεχνικές. Αντίθετα άλλη εναλλακτική είναι να γίνει τοπικά η επεξεργασία. Σε αυτή την περίπτωση, αναφερόμαστε για **Decentralized** ή **Distributed** αλγόριθμους· όπου κάθε node θα επικοινωνεί από μόνο του με τα γειτονικά του για την απόκτηση πληροφορίας· και έπειτα υπολογίζει μεμονωμένα την θέση του. Το μεγάλο πλεονέκτημα των Centralized αλγορίθμων είναι ότι απομακρύνουμε το πρόβλημα του computation από το κάθε node, όμως την ίδια στιγμή προκύπτουν δυσκολίες όπως καθυστερήσεις στις επικοινωνίες, μεγαλύτερο power consumption & bandwidth, ενώ τέλος και προβλήματα με το scalability του συστήματος - καθώς προτείνεται για μικρότερα networks. Το οποίο όμως σε ένα βαθμό· μπορεί να λυθεί με το να χρησιμοποιηθούν πολλαπλά **BSs**, συνεπώς να έχουμε ένα multi-tier network. Γνωστοί Centralized αλγόριθμοι είναι ο Multi Dimensional Scaling-Mobile Assisted Programming (**MDS-MAP**), Semi Definite Programming (**SDP**) και Localization Based on Simulated Annealing (**LBSA**) [77]. Παρακάτω στον *Algorithm 1* - ως παράδειγμα χρήσης των *Centralized* αλγορίθμων - παρουσιάζεται ο pseudo-code του **MDS-MAP**.

TABLE 2.3: Comparison of Centralized vs Distributed [78]

	Centralized	Distributed
Accuracy	Data collection	Position merging
Energy consumption	Signal transition	Position merging
Coverage area	Network topology	Anchor deployment
Costs	Central module	Anchor equipment

---

**Algorithm 1** Multi Dimensional Scaling-Mobile Assisted Programming [59]

---

**procedure** MDS-MAP ( $r_{i,j} = w(e) \quad \forall \quad e \in E$ )

- 1: Form a sparse matrix  $R$  from  $r_{i,j}$
  - 2: Run a standard all pairs shortest path algorithm on  $R$  to produce inter-node distances  $D$  (e.g. Dijkstra's, Floyd's)
  - 3: Run classical metric MDS on  $D$  to find estimated positions  $X$
  - 4: Transform the solution  $X$  into global coordinates  $G$
-

### 2.3.6 Range-based vs Range-free

Στο *Section 2.1* αναλύθηκαν τεχνικές εκτίμησης απόστασης ή γωνίας μεταξύ των nodes - συγκεκριμένα οι **RSSI**, **ToA**, **TDoA** και **AoA**. Σε αυτήν την περίπτωση, όταν χρησιμοποιούμε δηλαδή έξτρα hardware ή γενικά εκτεταμένες ranging τεχνικές για να καταλήξουμε στις θέσεις των free nodes στον χώρο, αναφερόμαστε σε **Range-based** μεθόδους. Θετικό σε αυτήν την περίπτωση είναι ότι έχουμε μεγαλύτερη ακρίβεια της θέσης, όμως ταυτόχρονα πιθανόν να χρειαστούμε ακριβότερα nodes - λόγω των έξτρα components ή του πιο *sophisticated hardware*.

Για να το αποφύγουμε αυτό, μπορούμε να χρησιμοποιήσουμε τεχνικές οι οποίες ονομάζονται **Range-free**. Αυτές οι τεχνικές δεν απαιτούν κανένα επιπρόσθετο hardware, διότι δεν χρησιμοποιούν εκτιμήσεις απόστασης/γωνίας, συνεπώς έχουμε μικρότερο κόστος ανά node. Έχουν αρκετό ενδιαφέρον λόγω της απλότητας τους παρά την μικρότερη ακρίβεια που παρέχουν. Γνωστοί Range-free αλγόριθμοι είναι ο Approximate Point in Triangle (**APIT**), Distance Vector-Hop (**DV-Hop**), Centroid και Gradient [77]. Όμοια με παραπάνω, στους *Algorithms 2* και *3* παρουσιάζονται οι pseudo-codes των **APIT** και **DV-Hop** αντίστοιχα.

TABLE 2.4: Comparison of Range-based vs Range-free [78]

	Range-based	Range-free
Accuracy	Ranging algorithm	Geometric algorithm
Energy consumption	Signal transition	Instruction execution
Coverage area	Signal cover	Network topology
Costs	Ranging module	Execution module

---

#### Algorithm 2 Approximate Point in Triangle [59]

---

**procedure** APIT

- 1: Receive beacon positions from hear-able beacons.
  - 2: Initialize inside-set to be empty
  - 3: **for** each triangle  $T_i$  in possible triangles formed over beacons **do**
  - 4:   Add  $T_i$  to inside-set if node is inside  $T_i$ . Go to next Step when accuracy of inside-set is sufficient.
  - 5: Compute position estimate as the center of mass of the intersection of all triangles in inside-set.
-

**Algorithm 3** Distance Vector-Hop [79] [80] [81]

---

```

procedure DV-Hop
1: for Each anchor do
2:   Broadcast a message with it's ID, location and hop count initialized to 0
3: for all nodes received all messages from neighbors do
4:   Neighbor nodes record the node ID, the coordinate values, and the smallest
     hop. Then forward packet after incrementing hop by one
5: for each anchor i to each anchor j do
6:   Estimate Average Hop Distance (AHD)

$$HopSize_i = \frac{\sum \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum hop_{i,j}}, i \neq j$$

7:   Broadcast AHD to network
8: for each free node u do
9:   Save the first AHD you hear, then broadcast it to your neighbors
10:  Calculate your distance from your closest anchor k,  $d_{uk} = HopSize_i \times hop_{uk}$ 
11: Based on known beacons' coordinates and estimated distances, use one of the
     methods described in Section 2.2 to compute free nodes locations

```

---

Σε ευρύτερη κλίμακα, στην βιβλιογραφία υπάρχουν επιπλέον αλγόριθμοι ή παραλλαγές χρήσης αυτών που αναφέρθηκαν, για την εύρεση τοποθεσίας nodes σε ένα **WSN**. Παρόλα αυτά, σε αυτό το σημείο έγινε μία συνοπτική αναφορά τους, και μία πρώτη ανάλυση γύρω από το localization problem. Στην συνέχεια, με βάση αυτές τις αρχές θα γίνει η μοντελοποίηση του συστήματος που συσχετίζεται αυτή η διπλωματική.





## Chapter 3

### Related Work

“This is where technology is  
now, imagine where we can go  
in the future”

---

*Timothy Chung*

Σε αυτό το *Chapter* περιγράφονται τρόποι - από την βιβλιογραφία - με τους οποίους, οι υπάρχουσες εφαρμογές από drone swarms επιλύουν το localization problem. Κάποια από τα συστήματα στις υπάρχουσες έρευνες έχουν καθαρά θεωρητική πλευρά, άλλα έχουν δοκιμαστεί σε real-life scenarios. Ενώ, σε αρκετές από αυτές τις εφαρμογές χρησιμοποιούνται - και για αυτό γίνεται αναφορά - τεχνικές που αναφέρθηκαν στο *Chapter 2*.

Κυρίως, αυτά που θα μελετήσουμε θα αφορούν Cooperative Localization (CL) των UAVs, ενώ επίσης μας ενδιαφέρουν τα Distributed Computation - Real Time Location System (RTLS) [82]. Το πρόβλημα του Local Positioning System (LPS) [83] έχει ερευνηθεί από διάφορες οπτικές οι οποίες μπορεί να βασίζονται σε RF, sound waves ή ακόμα και να είναι image oriented.

Σχεδόν βέβαιο είναι επίσης, ότι από όποια κατεύθυνση και αν προσεγγίσουμε το LPS να έχουμε την ανάγκη να χρησιμοποιήσουμε κάποια τεχνική για να συνενώσουμε όλες τις πληροφορίες που έχουμε από τους διάφορους αισθητήρες για κάθε μεμονωμένο time-frame. Την ανάγκη αυτού έρχεται να καλύψει η έννοια του *sensor fusion* [84], με μερικούς γνωστούς αλγόριθμους που να το επιτυγχάνουν - να είναι το Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), Covariance Intersection Filter (CIF), Split Covariance Intersection Filter (SCIF) και Belief Propagation (BP) [85].

Τέλος, είναι επίσης σημαντικό να αναφερθούν τα βασικά *coordinate frames* τα οποία χρησιμοποιούνται. Αισθητήρες όπως το IMU και το compass κάνουν μετρήσεις με γνώμονα το ίδιο το σώμα του αντικειμένου για αυτό - σε αυτήν την περίπτωση

το σύστημα αξόνων ονομάζεται body frame (B frame) [86]. Συχνά είναι όμως βολικό να μετατρέψουμε αυτές τις μετρήσεις σε ένα κοινό North East Down (NED) σύστημα, οπότε τότε αναφερόμαστε σε N frame [86] [87]. Παράδειγμα μεταξύ των δύο συστημάτων βρίσκεται στο *Figure 3.1*.

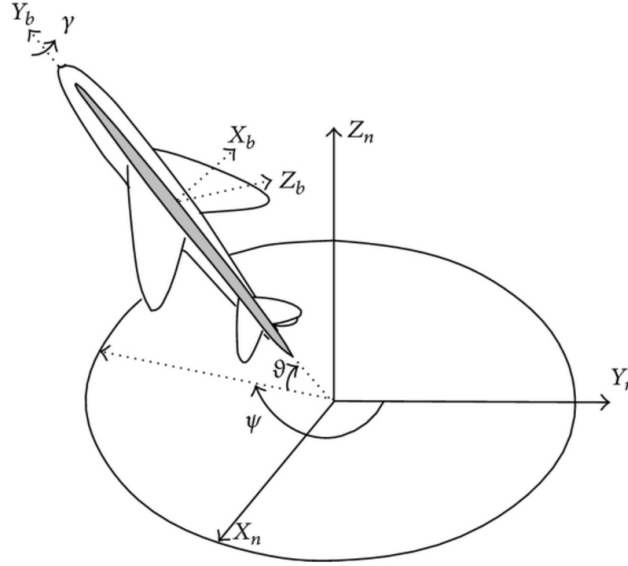


FIGURE 3.1: Important coordinate frames [87]

### 3.1 Radio Frequency

Το πιο εύκολο που μπορούμε να φανταστούμε είναι να χρησιμοποιήσουμε RF τεχνολογίες όπως WiFi, Zigbee και Ultra-Wideband (UWB) για την επικοινωνία και την υλοποίηση του LPS. Το WiFi είναι εύκολα προσβάσιμο<sup>1</sup> με μικρό κόστος ενώ το Zigbee προτιμάται για low power consumption εφαρμογές. Όμως προβλήματα αυτών των δύο είναι, ότι το WiFi έχει μικρή εμβέλεια<sup>1</sup> όπως επίσης είναι πολύ εύκολο να υπάρχουν παρεμβολές από τις υπόλοιπες γειτονικές συσκευές<sup>2</sup> άρα δεν το καθιστά καλή λύση για swarms. Το Zigbee επιλύει κάποια από αυτά τα προβλήματα<sup>3</sup>. Συχνότερη εφαρμογή έχουν τα UWB καθώς επιλύει θέματα σχετικά με την ακρίβεια των μετρήσεων, το κόστος καθώς και την εμβέλεια<sup>1</sup> ταυτόχρονα [88]. Τέλος, με την χρήση του UWB επίσης, μπορούμε εύκολα να δημιουργήσουμε ένα Flying Ad-hoc Network (FANET) το οποίο βοηθάει στην ευελιξία του συστήματος.

<sup>1</sup>Αυτό είναι κυρίως πρόβλημα για outdoor scenarios

<sup>2</sup>Αυτό είναι κυρίως πρόβλημα για indoor scenarios

<sup>3</sup>Σε indoor σενάρια παραμένει η δυσκολία για collision-free formation

### 3.1.1 UWB, IMU and GPS

Στο [86] θεωρείται ότι υπάρχουν  $N$  αριθμημένα UAV, τα οποία επικοινωνούν σε UWB, αξιοποιώντας την λογική του γράφου - για την αναπαράσταση τους - όπως περιγράφηκε στο *Section 2.3*. Ενώ κάθε node περιλαμβάνει IMU, Compass, GPS και Altitude and Heading Reference System (AHRS)<sup>4</sup> με την χρήση των οποίων μπορούν σε  $N$  frame να συλλέξουν πληροφορίες όπως παρουσιάζονται στους παρακάτω πίνακες για κάθε time-frame  $k$ .

$$\text{GPS Position: } r_{i,GPS}^k = [x_{i,GPS}^k \quad y_{i,GPS}^k \quad z_{i,GPS}^k]^T$$

$$\text{Velocity estimation: } \hat{v}_i^k = [\hat{v}_{xi}^k \quad \hat{v}_{yi}^k \quad \hat{v}_{zi}^k]^T$$

$$\text{Acceleration estimation: } \hat{a}_i^k = [\hat{a}_{xi}^k \quad \hat{a}_{yi}^k \quad \hat{a}_{zi}^k]^T$$

Σε αυτό το UWB δίκτυο είναι εύκολο το κάθε drone να μετρήσει την απόσταση του από ένα γειτονικό μέσω ToA-based RTT μεθόδου, με όνομα Single-sided Two-way Ranging (SS-TWR), όπως περιγράφει η εξίσωση (3.1). Τα  $T_{round}$  και  $T_{reply}$  είναι οι χρόνοι που μετρήθηκαν για την εκτίμηση απόστασης - η οποία όμως περιλαμβάνει σφάλμα ως εξής,  $\|r_i - r_j\|$  η πραγματική απόσταση και  $n_{UWB}$  τυχαία μεταβλητή με τυπική απόκλιση ίση με  $0.1m^5$  και στο *Figure 3.2* υπάρχει παράδειγμα αυτής.

$$d_{ij} = \frac{1}{2}(T_{round} - T_{reply}) \times c = \|r_i - r_j\| + n_{UWB}, \quad \text{with } n_{UWB} \sim N(0, \sigma_{UWB}^2) \quad (3.1)$$

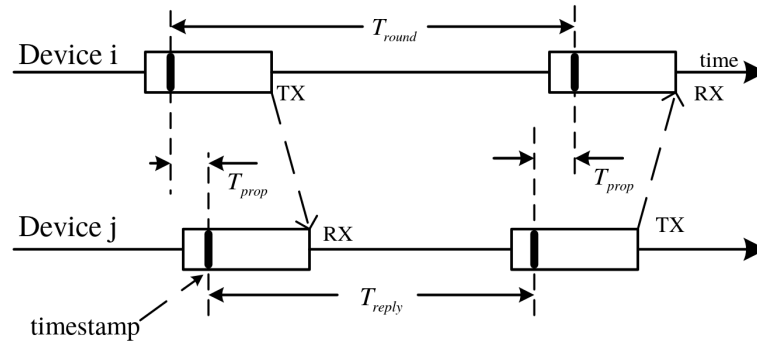


FIGURE 3.2: Illustration of Single-sided Two-way ranging [86]

<sup>4</sup>Αξιοποιείται από ένα flight control unit

<sup>5</sup>Βάση θεωρήσης των συγγραφών της έρευνας

Παρόμοιας λογικής σφάλματα θεωρείται ότι συμπεριλαμβάνουν και οι μετρήσεις από τους άλλους αισθητήρες, ενώ επίσης ότι το κάθε node μπορεί να μεταφέρει τις πληροφορίες αυτές στα υπόλοιπα.

Πριν το relative position estimation, για λόγους που έχουν να κάνουν με Ranging outlier rejection και Ranging failure data completion γίνεται ένα preprocessing με χρήση του EKF στα δεδομένα - όπως παρουσιάζεται στο Figure 3.3 - ώστε να έχουμε σαν αποτέλεσμα μετά από αυτό το στάδιο μία καλύτερη εκτίμηση των αποστάσεων  $d_{ij}$  για κάθε  $i$  και  $j$  nodes.

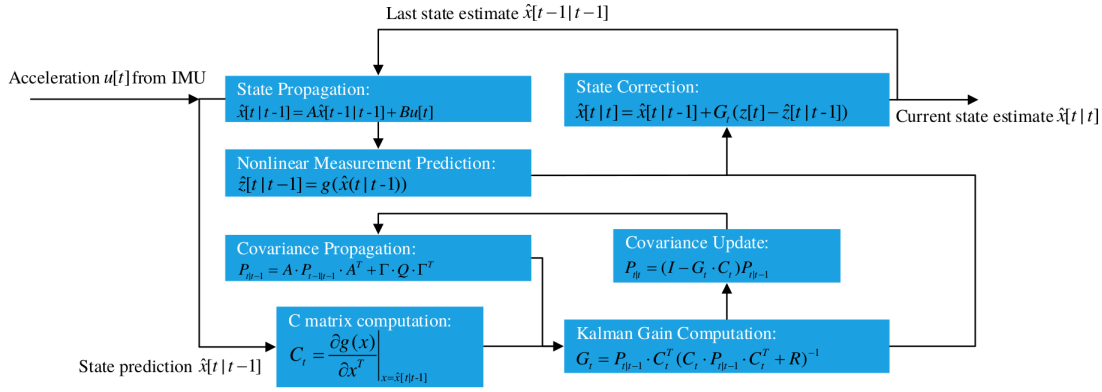


FIGURE 3.3: Data preprocessing work flow using EKF[86]

Ενώ σε επόμενο στάδιο, για το κομμάτι του relative position estimation, θεωρούν το σύστημα ως ένα γράφο με λογική ενός *spring system* - με διαφορετικούς  $k_o$  συντελεστές σκληρότητας - όπου στην κατάσταση όπου το σύστημα σταθεροποιηθεί, θεωρείται ότι είναι το σημείο το οποίο περιγράφει την τοποθεσία των nodes. Ψάχνουν να βρουν το σημείο στο οποίο η συνολική δυναμική ενέργεια στο σύστημα είναι η μικρότερη. Ουσιαστικά μετατρέπουν το position estimation πρόβλημα σε ένα non-convex optimization πρόβλημα και προσπαθούν να το λύσουν. Ένα overview του συστήματος που δημιούργησαν παρουσιάζεται στο Figure 3.4.

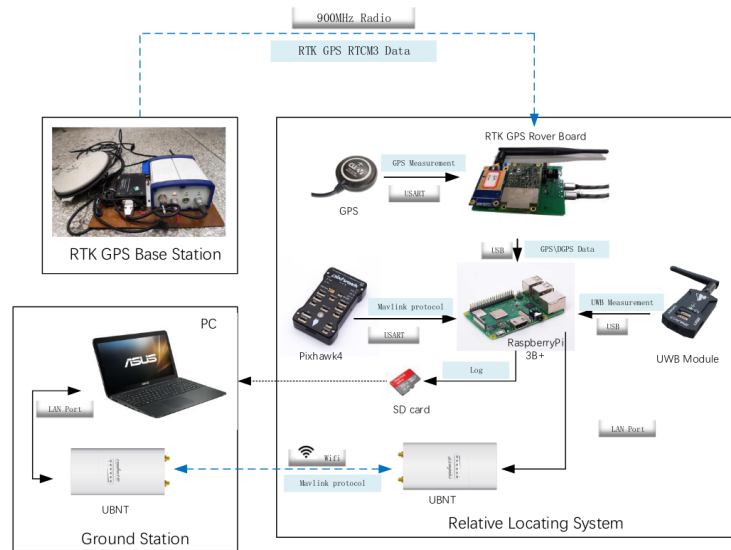


FIGURE 3.4: System overview for paper [86]

## 3.2 Thesis Approach

TODO: As last section of this chapter



## Chapter 4

# Design Features and Implementation

”

---

TODO:





## Chapter 5

# Experiments and Results

”

---

TODO:



## Chapter 6

# Conclusions and Future Work

”

---

TODO:



# References

- [4] S. G. Gupta, D. Ghonge, P. M. Jawandhiya, *et al.*, “Review of unmanned aircraft system (uas)”, *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume*, vol. 2, 2013. [Online]. Available: [http://refhub.elsevier.com/S0376-0421\(16\)30134-8/sbref2](http://refhub.elsevier.com/S0376-0421(16)30134-8/sbref2).
- [7] M. Hassanalian and A. Abdelkefi, “Classifications, applications, and design challenges of drones: A review”, *Progress in Aerospace Sciences*, vol. 91, pp. 99–131, 2017, ISSN: 0376-0421. DOI: <https://doi.org/10.1016/j.paerosci.2017.04.003>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0376042116301348>.
- [8] A. Tahir, J. Böling, M.-H. Haghbayan, H. T. Toivonen, and J. Plosila, “Swarms of unmanned aerial vehicles — a survey”, *Journal of Industrial Information Integration*, vol. 16, p. 100–106, 2019, ISSN: 2452-414X. DOI: <https://doi.org/10.1016/j.jii.2019.100106>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2452414X18300086>.
- [26] S. Milani and A. Memo, “Impact of drone swarm formations in 3d scene reconstruction”, pp. 2598–2602, 2016. DOI: [10.1109/ICIP.2016.7532829](https://doi.org/10.1109/ICIP.2016.7532829).
- [40] A. Bürkle, F. Segor, and M. Kollmann, “Towards autonomous micro uav swarms”, *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 339–353, 2011, ISSN: 1573-0409. DOI: [10.1007/s10846-010-9492-x](https://doi.org/10.1007/s10846-010-9492-x). [Online]. Available: <https://doi.org/10.1007/s10846-010-9492-x>.
- [41] E. Mitka and S. Mouroutsos, “Classification of drones”, *American Journal of Engineering research*, vol. 6, pp. 36–41, Jul. 2017.
- [44] M. F. i Azam and M. N. Ayyaz, *Location and position estimation in wireless sensor networks*, 2016. arXiv: [1611.03420](https://arxiv.org/abs/1611.03420) [cs.NI].
- [45] A. Boukerche, H. A.B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, “Localization systems for wireless sensor networks”, *IEEE Wireless Communications*, vol. 14, no. 6, pp. 6–12, 2007. DOI: [10.1109/MWC.2007.4407221](https://doi.org/10.1109/MWC.2007.4407221).
- [46] G. Mao, B. Fidan, and B. D. Anderson, “Wireless sensor network localization techniques”, *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007, ISSN: 1389-1286. DOI: <https://doi.org/10.1016/j.comnet.2006.11.018>.

- [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128606003227>.
- [47] R. Peng and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks", in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, vol. 1, 2006, pp. 374–382. DOI: [10.1109/SAHCN.2006.288442](https://doi.org/10.1109/SAHCN.2006.288442).
- [51] B. Yamamoto, A. Wong, P. J. Agcanas, K. Jones, D. Gaspar, R. Andrade, and A. Z. Trimble, "Received signal strength indication (rss) of 2.4 ghz and 5 ghz wireless local area network systems projected over land and sea for near-shore maritime robot operations", *Journal of Marine Science and Engineering*, vol. 7, no. 9, p. 290, Aug. 2019, ISSN: 2077-1312. DOI: [10.3390/jmse7090290](https://doi.org/10.3390/jmse7090290). [Online]. Available: <http://dx.doi.org/10.3390/jmse7090290>.
- [53] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, "Using rss value for distance estimation in wireless sensor networks based on zigbee", pp. 303–306, 2008. DOI: [10.1109/IWSSIP.2008.4604427](https://doi.org/10.1109/IWSSIP.2008.4604427).
- [55] M. Botta and M. Simek, "Adaptive distance estimation based on rss in 802.15.4 network", *Radioengineering*, vol. 22, pp. 1162–1168, Dec. 2013.
- [59] J. Bachrach and C. Taylor, *1 localization in sensor networks*, 2004. [Online]. Available: <https://www.semanticscholar.org/paper/1-Localization-in-Sensor-Networks-Bachrach-Taylor/a70a5ac4c092fec8c140fc3850e2e1f6d7b698cf>.
- [71] A. M. H. KHALEL, "Position location techniques in wireless communication systems", 2010. [Online]. Available: <http://urn.kb.se/resolve?urn=urn:nbn:se:bth-4796>.
- [72] R. Kaune, C. Steffes, S. Rau, W. Konle, and J. Pagel, "Wide area multilateration using ads-b transponder signals", in *2012 15th International Conference on Information Fusion*, 2012, pp. 727–734.
- [75] B. O'Keefe, "Finding location with time of arrival and time difference of arrival techniques", *ECE Senior Capstone Project*, 2017. [Online]. Available: [https://sites.tufts.edu/eeseniordesignhandbook/files/2017/05/FireBrick\\_OKeefe\\_F1.pdf](https://sites.tufts.edu/eeseniordesignhandbook/files/2017/05/FireBrick_OKeefe_F1.pdf).
- [76] X. Li, Z. D. Deng, L. T. Rauchenstein, and T. J. Carlson, "Contributed review: Source-localization algorithms and applications using time of arrival and time difference of arrival measurements", *Review of Scientific Instruments*, vol. 87, no. 4, p. 041502, 2016. DOI: [10.1063/1.4947001](https://doi.org/10.1063/1.4947001). eprint: <https://doi.org/10.1063/1.4947001>. [Online]. Available: <https://doi.org/10.1063/1.4947001>.

- [77] S. P. Singh and S. Sharma, “Range free localization techniques in wireless sensor networks: A review”, *Procedia Computer Science*, vol. 57, pp. 7–16, 2015, 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2015.07.357>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877050915018864>.
- [78] C. Zhou, T. Xu, and H. Dong, “Distributed locating algorithm mds-map (lf) based on low-frequency signal”, *Computer Science and Information Systems*, vol. 12, pp. 55–55, Nov. 2015. DOI: [10.2298/CSIS140801055Z](https://doi.org/10.2298/CSIS140801055Z).
- [79] X. Wang and N. Yunfeng, “An improved distance vector-hop localization algorithm based on coordinate correction”, *International Journal of Distributed Sensor Networks*, vol. 13, p. 155 014 771 774 183, Nov. 2017. DOI: [10.1177/1550147717741836](https://doi.org/10.1177/1550147717741836).
- [80] X. Yang and W. Zhang, “An improved dv-hop localization algorithm based on hop distance and hops correction”, *International Journal of Multimedia and Ubiquitous Engineering*, vol. 11, pp. 319–328, Jun. 2016. DOI: [10.14257/ijmue.2016.11.6.28](https://doi.org/10.14257/ijmue.2016.11.6.28).
- [81] L. Yin, “A new distance vector-hop localization algorithm based on half-measure weighted centroid”, *Mobile Information Systems*, vol. 2019, pp. 1–9, Jan. 2019. DOI: [10.1155/2019/9892512](https://doi.org/10.1155/2019/9892512).
- [85] S. Goel, “A distributed cooperative uav swarm localization system: Development and analysis”, Sep. 2017. DOI: [10.33012/2017.15217](https://doi.org/10.33012/2017.15217).
- [86] Y. Qi, Y. Zhong, and Z. Shi, “Cooperative 3-d relative localization for UAV swarm by fusing UWB with IMU and GPS”, *Journal of Physics: Conference Series*, vol. 1642, p. 012 028, Sep. 2020. DOI: [10.1088/1742-6596/1642/1/012028](https://doi.org/10.1088/1742-6596/1642/1/012028). [Online]. Available: <https://doi.org/10.1088/1742-6596/1642/1/012028>.
- [87] L. Zhao and Q. Wang, “Design of an attitude and heading reference system based on distributed filtering for small uav”, *Mathematical Problems in Engineering*, vol. 2013, Jul. 2013. DOI: [10.1155/2013/498739](https://doi.org/10.1155/2013/498739).
- [88] J. Li, Y. Bi, K. Li, K. Wang, F. Lin, and B. M. Chen, *Accurate 3d localization for mav swarms by uwb and imu fusion*, 2018. arXiv: [1807.10913](https://arxiv.org/abs/1807.10913) [cs.R0].





## External Links

- [1] M. Satell, *Ultimate list of drone stats for 2020*, Jul. 2020. [Online]. Available: <https://www.phillybyair.com/blog/drone-stats/> (visited on 11/2020).
- [2] B. I. Intelligence, *Drone market outlook: Industry growth trends, market stats and forecast*, Mar. 2020. [Online]. Available: <https://www.businessinsider.com/drone-industry-analysis-market-trends-growth-forecasts> (visited on 11/2020).
- [3] C. de León, *Drone delivery? amazon moves closer with f.a.a. approval*, Aug. 2020. [Online]. Available: <https://www.nytimes.com/2020/08/31/business/amazon-drone-delivery.html> (visited on 11/2020).
- [5] *Unmanned aerial vehicle*. [Online]. Available: [https://en.wikipedia.org/wiki/Unmanned\\_aerial\\_vehicle](https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle) (visited on 11/2020).
- [6] *List of unmanned aerial vehicles*. [Online]. Available: [https://en.wikipedia.org/wiki/List\\_of\\_unmanned\\_aerial\\_vehicles](https://en.wikipedia.org/wiki/List_of_unmanned_aerial_vehicles) (visited on 11/2020).
- [9] *A guide to fixed wing drones*. [Online]. Available: <https://www.coptrz.com/a-guide-to-fixed-wing-drones/> (visited on 11/2020).
- [10] *Aircraft principal axes*. [Online]. Available: [https://en.wikipedia.org/wiki/Aircraft\\_principal\\_axes](https://en.wikipedia.org/wiki/Aircraft_principal_axes) (visited on 11/2020).
- [11] *Microelectromechanical systems*. [Online]. Available: [https://en.wikipedia.org/wiki/Microelectromechanical\\_systems](https://en.wikipedia.org/wiki/Microelectromechanical_systems) (visited on 01/2021).
- [12] *Microcontroller*. [Online]. Available: <https://en.wikipedia.org/wiki/Microcontroller> (visited on 11/2020).
- [13] *Microprocessor*. [Online]. Available: <https://en.wikipedia.org/wiki/Microprocessor> (visited on 11/2020).
- [14] *Electronic speed control*. [Online]. Available: [https://en.wikipedia.org/wiki/Electronic\\_speed\\_control](https://en.wikipedia.org/wiki/Electronic_speed_control) (visited on 11/2020).
- [15] *Inertial measurement unit*. [Online]. Available: [https://en.wikipedia.org/wiki/Inertial\\_measurement\\_unit](https://en.wikipedia.org/wiki/Inertial_measurement_unit) (visited on 11/2020).
- [16] *Tuc researcher's night 2020: Drones, satellites, and more ...* Nov. 2020. [Online]. Available: <https://www.youtube.com/watch?v=9ZZW-XQKGlw&feature=youtu.be&fbclid=IwAR2VdxWuF0wiCbEdEGJcBDxapixuPtExsViDCoBEPQLnU8YQLg3hGRBn47w> (visited on 11/2020).

- [17] *Senselab research*. [Online]. Available: <http://senselab.tuc.gr/> (visited on 11/2020).
- [18] *Optimized flocking of autonomous drones in confined environments*, Jul. 2018. [Online]. Available: <https://www.youtube.com/watch?v=E4XpyG4eMKE> (visited on 11/2020).
- [19] *An autonomous swarm of flying robots*, Feb. 2014. [Online]. Available: [https://www.youtube.com/watch?v=W7yJx\\_IY9Mg](https://www.youtube.com/watch?v=W7yJx_IY9Mg) (visited on 11/2020).
- [20] *Drone swarm performance and applications*, Jun. 2019. [Online]. Available: <https://www.embention.com/news/drone-swarm-performance-and-applications/> (visited on 11/2020).
- [21] M. Murison, *The potential of drone swarms*, Apr. 2019. [Online]. Available: <https://blog.dronebase.com/the-potential-of-drone-swarms> (visited on 11/2020).
- [22] *Swarm robotics*. [Online]. Available: [https://en.wikipedia.org/wiki/Swarm\\_robotics](https://en.wikipedia.org/wiki/Swarm_robotics) (visited on 11/2020).
- [23] *Controlling robotic swarms*, Dec. 2014. [Online]. Available: <https://www.youtube.com/watch?v=stzQNjtDg0g> (visited on 11/2020).
- [24] *Kargu - the kamikaze drones getting ready for the swarm operation*, Jul. 2019. [Online]. Available: <https://www.youtube.com/watch?v=3d28APIfwSI> (visited on 11/2020).
- [25] *A swarm of nano quadrotors*, Pennsylvania, Jan. 2012. [Online]. Available: <https://www.youtube.com/watch?v=YQIMGV5vtd4> (visited on 11/2020).
- [27] *How intel made its world record-breaking drone show*, Jul. 2018. [Online]. Available: [https://www.youtube.com/watch?v=zdLjoqa\\_oUs](https://www.youtube.com/watch?v=zdLjoqa_oUs) (visited on 11/2020).
- [28] *Ford motor company / firefly drone shows*, Aug. 2019. [Online]. Available: <https://www.youtube.com/watch?v=Bdsr1cojSxo> (visited on 11/2020).
- [29] *A swarm of mini drones makes ... magic! / marco tempest*. [Online]. Available: <https://www.youtube.com/watch?v=u2bQSKvZ2qI> (visited on 12/2020).
- [30] *Hollywood welcomes drones to the set*, Jan. 2015. [Online]. Available: <https://www.youtube.com/watch?v=GXhlgNHK14s> (visited on 12/2020).
- [31] *3d mapping with a dji phantom & drone deploy*, Aug. 2016. [Online]. Available: <https://www.youtube.com/watch?v=SATijfXnshg> (visited on 12/2020).
- [32] *Dji - m300 rtk - coordinating police missions from the sky*, May 2020. [Online]. Available: <https://www.youtube.com/watch?v=FpkEsmU1NEo> (visited on 12/2020).

- [33] *Drones for disaster management (nepal earthquake)*, Aug. 2015. [Online]. Available: <https://www.youtube.com/watch?v=xYrhAoX2H8I> (visited on 12/2020).
- [34] *Drone search and rescue - wake forest fire department*, Jun. 2018. [Online]. Available: [https://www.youtube.com/watch?v=Ih\\_ayi318X0](https://www.youtube.com/watch?v=Ih_ayi318X0) (visited on 12/2020).
- [35] *Amazon gets faa approval to operate a fleet of delivery drones*, Aug. 2020. [Online]. Available: <https://www.youtube.com/watch?v=beZ8iKGuDeo> (visited on 12/2020).
- [36] *Unmanned combat aerial vehicle*. [Online]. Available: [https://en.wikipedia.org/wiki/Unmanned\\_combat\\_aerial\\_vehicle](https://en.wikipedia.org/wiki/Unmanned_combat_aerial_vehicle) (visited on 11/2020).
- [37] *Military robot*. [Online]. Available: [https://en.wikipedia.org/wiki/Military\\_robot](https://en.wikipedia.org/wiki/Military_robot) (visited on 11/2020).
- [38] *Isr unmanned aerial vehicles*. [Online]. Available: <https://www.militaryfactory.com/aircraft/uav-intelligence-surveillance-reconnaissance.asp> (visited on 12/2020).
- [39] *List of unmanned aerial vehicle applications*. [Online]. Available: [https://en.wikipedia.org/wiki/List\\_of\\_unmanned\\_aerial\\_vehicle\\_applications](https://en.wikipedia.org/wiki/List_of_unmanned_aerial_vehicle_applications) (visited on 11/2020).
- [42] *Drone swarms: The buzz of the future*, Nov. 2015. [Online]. Available: <https://www.youtube.com/watch?v=0so5Fc6B7W4> (visited on 11/2020).
- [43] *Wireless sensor network*. [Online]. Available: [https://en.wikipedia.org/wiki/Wireless\\_sensor\\_network](https://en.wikipedia.org/wiki/Wireless_sensor_network) (visited on 12/2020).
- [48] *Research on localization algorithms and technologies*, Nov. 2015. [Online]. Available: <https://www.youtube.com/watch?v=XDFVtYEh0tM> (visited on 12/2020).
- [49] *Path loss*. [Online]. Available: [https://en.wikipedia.org/wiki/Path\\_loss](https://en.wikipedia.org/wiki/Path_loss) (visited on 12/2020).
- [50] *Free-space path loss*. [Online]. Available: [https://en.wikipedia.org/wiki/Free-space\\_path\\_loss](https://en.wikipedia.org/wiki/Free-space_path_loss) (visited on 12/2020).
- [52] *Friis transmission equation*. [Online]. Available: [https://en.wikipedia.org/wiki/Friis\\_transmission\\_equation](https://en.wikipedia.org/wiki/Friis_transmission_equation) (visited on 12/2020).
- [54] *Dbm*. [Online]. Available: <https://en.wikipedia.org/wiki/DBm> (visited on 12/2020).
- [56] *Kinematics*. [Online]. Available: <https://en.wikipedia.org/wiki/Kinematics> (visited on 12/2020).
- [57] *An introduction to direction finding*, Feb. 2020. [Online]. Available: <https://www.youtube.com/watch?v=N8rZIAHxAH4> (visited on 12/2020).

- [58] *Speed of sound*. [Online]. Available: [https://en.wikipedia.org/wiki/Speed\\_of\\_sound](https://en.wikipedia.org/wiki/Speed_of_sound) (visited on 12/2020).
- [60] *Beamforming*. [Online]. Available: <https://en.wikipedia.org/wiki/Beamforming> (visited on 12/2020).
- [61] *Directional antenna*. [Online]. Available: [https://en.wikipedia.org/wiki/Directional\\_antenna](https://en.wikipedia.org/wiki/Directional_antenna) (visited on 12/2020).
- [62] *Phase difference*, Dec. 2015. [Online]. Available: <https://www.miniphysics.com/phase-difference.html> (visited on 12/2020).
- [63] *Fmcw radars lec 5: Angle estimation*, May 2018. [Online]. Available: <https://www.youtube.com/watch?v=0k2HarUxS8o> (visited on 12/2020).
- [64] *How gps works? trilateration explained*, Mar. 2014. [Online]. Available: <https://www.youtube.com/watch?v=403ZVHVFhes> (visited on 12/2020).
- [65] *Trilateration vs triangulation – how gps receivers work*, Dec. 2020. [Online]. Available: <https://gisgeography.com/trilateration-triangulation-gps/> (visited on 12/2020).
- [66] *Omnidirectional antenna*. [Online]. Available: [https://en.wikipedia.org/wiki/Omnidirectional\\_antenna](https://en.wikipedia.org/wiki/Omnidirectional_antenna) (visited on 12/2020).
- [67] *Isotropic radiator*. [Online]. Available: [https://en.wikipedia.org/wiki/Isotropic\\_radiator](https://en.wikipedia.org/wiki/Isotropic_radiator) (visited on 12/2020).
- [68] *Rssi-based accurate indoor localization scheme for wireless sensor networks*, Nov. 2015. [Online]. Available: <https://www.youtube.com/watch?v=CWvRJdF7oVE> (visited on 12/2020).
- [69] J. Wedding, *Find x location using 3 known (x,y) location using trilateration*, Jul. 2017. [Online]. Available: <https://math.stackexchange.com/questions/884807/find-x-location-using-3-known-x-y-location-using-trilateration> (visited on 12/2020).
- [70] *Hybrid localization algorithm for wireless sensor networks*, Oct. 2015. [Online]. Available: <https://www.youtube.com/watch?v=kf-hXqZHkAA> (visited on 12/2020).
- [73] I. Biswas, *Triangulation vs trilateration vs multilateration – for indoor positioning systems*, Jun. 2019. [Online]. Available: <https://www.pathpartnertech.com/triangulation-vs-trilateration-vs-multilateration-for-indoor-positioning-systems/> (visited on 12/2020).
- [74] *Multilateration*. [Online]. Available: <https://en.wikipedia.org/wiki/Multilateration> (visited on 12/2020).
- [82] *Real-time locating system*. [Online]. Available: [https://en.wikipedia.org/wiki/Real-time\\_locating\\_system](https://en.wikipedia.org/wiki/Real-time_locating_system) (visited on 01/2021).

- 
- [83] *Positioning system*. [Online]. Available: [https://en.wikipedia.org/wiki/Positioning\\_system](https://en.wikipedia.org/wiki/Positioning_system) (visited on 01/2021).
- [84] *Sensor fusion*. [Online]. Available: [https://en.wikipedia.org/wiki/Sensor\\_fusion](https://en.wikipedia.org/wiki/Sensor_fusion) (visited on 01/2021).



*Last Page*