T.C.
ISTANBUL MEDENIYET UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
ELECTRICAL AND ELECTRONICS ENGINEERING

TRACKING OF MOVING TARGETS WITH AN AERIAL ROBOT

MASTER THESIS

MUHAMMED KORAY YILMAZ

JANUARY 2022

T.C.
ISTANBUL MEDENIYET UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
ELECTRICAL AND ELECTRONICS ENGINEERING

TRACKING OF MOVING TARGETS WITH AN AERIAL ROBOT

MASTER THESIS

MUHAMMED KORAY YILMAZ

SUPERVISOR

ASST. PROF. HALUK BAYRAM

JANUARY 2022

BİLDİRİM

Hazırladığım tezin tamamen kendi çalışmam olduğunu, akademik ve etik kuralları gözeterek çalıştığımı ve her alıntıya kaynak gösterdiğimi taahhüt ederim.

Muhammed Koray YILMAZ

Danışmanlığını yaptığım işbu tezin tamamen öğrencinin çalışması olduğunu, akademik ve etik kuralları gözeterek çalıştığını taahhüt ederim.

Dr. öğr. Üyesi Haluk BAYRAM

İMZA SAYFASI

Muhammed Koray YILMAZ tarafından hazırlanan “Hareketli Hedeflerin Hava Robotu ile Takibi / Tracking of Moving Targets with an Aerial Robot” başlıklı bu yüksek lisans tezi, Elektrik ve Elektronik Anabilim/bilim Dalında hazırlanmış ve jürimiz tarafından kabul edilmiştir.

JÜRİ ÜYELERİ **İMZA**

Tez Danışmanı:
Dr. Öğr. Üyesi Haluk Bayram ....
Kurumu: İstanbul Medeniyet Üniversitesi

Üyeler:
Dr. Öğr. Üyesi İbrahim Genç ....
Kurumu: İstanbul Medeniyet Üniversitesi

Prof. Dr. H. Işıl Bozma ....
Kurumu: Boğaziçi Üniversitesi

Tez Savunma Tarihi: 26.01.2022

ACKNOWLEDGEMENTS

I would like to thank Asst. Prof. Haluk Bayram for his extensive support, encouragement and guidance that enlightened my way and gave me the power to continue. Hence, I could cope with the problems that I encountered during this journey. I am very grateful and feel fortunate that I had the chance to meet and work with him.

I would like to thank my thesis committee members Prof. H. Işıl Bozma and Asst. Prof. İbrahim Genç for their valuable comments.

I also would like to thank Fatih Çakır for his contribution to making a 3-element Yagi-Uda antenna and Samet Özabacı for the radio transmitter module.

I really appreciate all the help that Ferdi Dumandağ provided in the field experiments. He was there whenever I needed him so that I could perform the field experiments.

I am so happy to have such a nice family that always listens to my complaints with patience, and helps, encourages me. I am very thankful for their support and existence in my life.

I want to thank TUBITAK for their scholarship. It is quite an honor and priceless reward for me to be supported by TUBITAK 2210/C program.

This work has been partially supported by TUBITAK 2232 (grant number 119C008).

ABSTRACT

Tracking of Moving Targets with an Aerial Robot

Muhammed Koray YILMAZ

Master Thesis, Electrical and Electronics Engineering Department, Electrical and Electronics Engineering Program

Supervisor: Asst. Prof. Haluk BAYRAM

January, 2022. 56 Pages.

This thesis considers the problem of tracking a moving target with a radio transmitter using an aerial robot in an online manner. The aerial robot is equipped with a low-cost directional antenna and Software Defined Radio based USB receiver to obtain the signal. The aerial robot rotates around itself and takes a predefined number of measurements to obtain the bearing angle in which the received signal strength is maximized. The measurement uncertainty is assumed to be bounded and represented by two triangular areas divided by a bisector. To localize and track the target, a particle filter based approach is proposed. In this approach, we integrate discrete and bounded measurement model with the particle filter in such a way that the particles' weights are updated based on a novel method which considers the measurement wedge and the particles' location along with a logistic function. We also incorporate the doubling strategy into the particle filter in order to determine the measurement locations. The proposed approach is validated using a target moving with varying velocity and acceleration. For this purpose, we make use of four different motion models (Heterogeneous Behavior, Lévy Walk, Random Walk and Velocity Model), which are used in modelling animal movements in wild life studies. We validated the tracking performance of the approach through a series of extensive simulations. For the field experiments, we developed a low-cost hardware platform and software infrastructure for the proposed tracking system. Using this platform, we conducted field experiments for the stationary and moving targets.

Keywords: particle filter, target tracking, bounded bearing measurements, aerial robots, wildlife monitoring.

ÖZET

Hareketli Hedeflerin Hava Robotu ile Takibi

Muhammed Koray YILMAZ

Yüksek Lisans Tezi, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Elektrik-Elektronik Mühendisliği Programı

Danışman: Dr.Öğr.Üyesi Haluk BAYRAM

Ocak, 2022. 56 Sayfa.

Bu tez, çevrimiçi olarak bir hava robotu ile üzerinde bir radyo verici olan hareketli bir hedefin takibi problemini ele almaktadır. Hava robotu, hedeften gelen sinyali almak için düşük maliyetli bir yönlü anten ve Yazılım Tanımlı Radyo tabanlı bir alıcıya sahiptir. Hava robotu, sinyal gücünün en yüksek olduğu doğrultu açısını belirlemek için kendi etrafında belirli açılarda dönerek, hedeften gelen sinyalleri kaydeder. Ölçüm belirsizliğinin bir açıortay ile bölünmüş iki üçgen alanla sınırlandığı ve temsil edildiği varsayılır. Hedefin yerini belirlemek ve izlemek için parçacık filtresi tabanlı bir yaklaşım önerilmiştir. Bu yaklaşımda, ayrık ve sınırlı ölçüm modeli parçacık filtresi ile entegre edilmiştir. Bu entegrasyon parçacıkların ağırlıklarının güncellenmesi yöntemi ile sağlanmıştır. Parçacıkların ağırlıkları, parçacıkların ölçüm alanına ve bu alanın açıortayına olan yakınlığı kullanılarak, lojistik fonksiyonu yardımıyla güncellenmiştir. Ayrıca ölçüm yerlerini belirlemek için ikiye katlama stratejisi parçacık filtresine dahil edilmiştir. Önerilen yaklaşımın takip performansı, değişen hız ve ivme ile hareket eden bir hedef kullanılarak doğrulanmıştır. Bu amaçla yaban hayatı çalışmalarında hayvan hareketlerinin modellenmesinde kullanılan dört farklı hareket modelinden (Heterojen Davranış, Lévy Yürüş, Rasgele Yürüyüş ve Hız Modeli) yararlanılmıştır. Bir dizi kapsamlı simülasyon aracılığıyla yaklaşımın hedef takip performansı doğrulanmıştır. Saha deneyleri için düşük maliyetli çok-rotorlu bir robotik platform ve yazılım altyapısı geliştirilmiştir. Daha sonra bu platform kullanılarak, sabit ve hareketli hedefler için saha deneyleri yapılmıştır.

Anahtar Kelimeler: parçacık süzgeç, hedef takibi, sınırlı doğrultu açı ölçümü, hava robotları, yaban hayatı gözleme.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

ROS	Robot Operating System
RSS	Received Signal Strength
RSSI	Received Signal Strength Indication
SDR	Software Defined Radio
ssh	Secure Shell
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus

LIST OF SYMBOLS

d	Particle's distance to the origin
d_e	Particle's distance to the edge of measurement wedge
d_b	Particle's distance to the bisector of measurement wedge
d_j	Particle's distance to the UAV, if it is behind of the UAV
d_k	Particle's distance to the edge of measurement wedge, if it is outside of the measurement area
d_{step}	The distance between the UAV and next measurement location
L	The maximum value of the logistic function
$\mathcal{N}$	Gaussian Distribution
$\mathcal{U}$	Uniform Distribution
p	The particle
P	List of the particles
p_i	Particle that is inside of the measurement wedge
p_j	Particle that is behind of the UAV
p_k	Particle that is outside of the measurement wedge
p_w	The weight of the particle

p_x	x coordinate of the the particle
p_y	y coordinate of the particle
$p.velocity$	The velocity of the particle
$p.limitVel$	The velocity limit of the particle
r	Radius of the searching area centered with the UAV
w_T	The total weight of the particles
x_b	x coordinate of the end point of the measurement wedge's bisector
x_{est}	x coordinate of the estimated target location
x_{meas}	x coordinate of the measurement location
x_{UAV}	x coordinate of the UAV
x_0	The midpoint of the logistic function
y_b	y coordinate of the end point of the measurement wedge's bisector
y_{est}	y coordinate of the estimated target location
y_{meas}	y coordinate of the measurement location
y_{UAV}	y coordinate of the UAV
α	Measurement noise

β	Measured bearing
θ	Uniformly chosen particle's angle to the origin
λ	The steepness of the logistic function
σ	Variance of the Gauss distribution
ψ	True bearing

CHAPTER 1: INTRODUCTION

Localizing a particular signal source finds many applications in real life. In wildlife research, animals with a radio transmitter can be tracked, thus the examination of their natural life can be achieved in a short time without the possibility of entering dangerous areas and encountering animals by using the aerial robots in [Bayram et al., 2016a, Bayram et al., 2018, Hui et al., 2021, Nguyen et al., 2019]. These tracked animals can also be invasive fish species in an area. These invasive fish are observed using small autonomous boats and wheeled vehicles in a frozen water area [Vander Hook et al., 2015, Bayram et al., 2016b].

Another application field is natural disasters such as earthquakes and floods in which lost people are required to be localized. If there is a signal-emitting device next to a person such as a mobile phone, the location of a lost person can be detected quickly, and the necessary intervention can be made with an aerial robot, regardless of the environmental conditions. Therefore, UAV systems are used to locate the mobile phone signal source for search and rescue purposes in [Isaacs et al., 2014]. Similarly, localizing the signal source such as signal jamming devices plays an important role in the defense area [Perkins et al., 2015].

1.1 PROBLEM STATEMENT

The problem is to localize a stationary target or track a moving target in an online manner. We consider a target with a radio transmitter, moving with an unknown motion model in a circular area where the UAV is in the center. The target signal must be in the range of the receiver for the localization.[1] The problem illustration can be seen in Figure 1.1.

[1] If the signal is not received by the UAV, meaning the target is out of range, a search strategy can be applied. Since our focus in this work is on target localization and tracking, we will not deal with search problem.

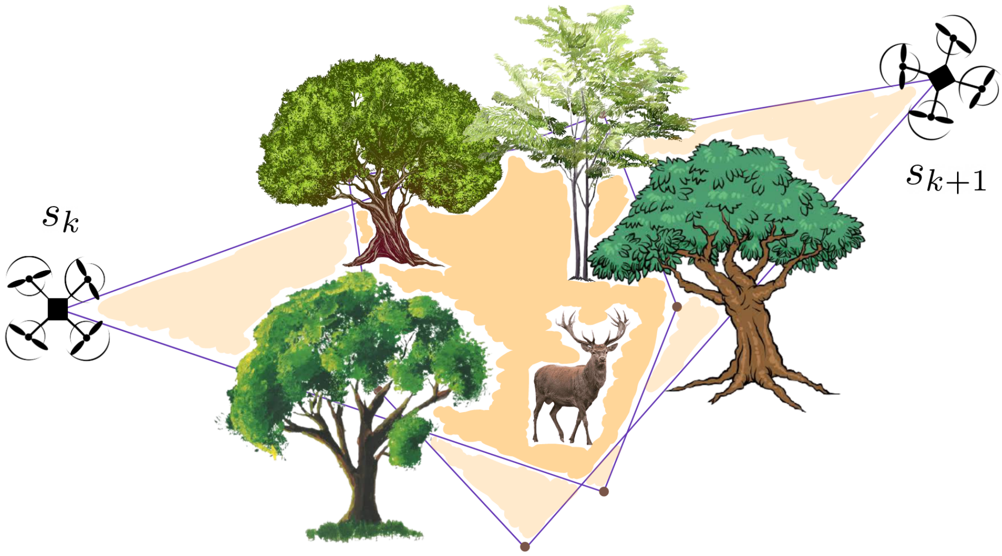

Figure 1.1: Problem illustration: The UAV takes bearing measurements from s_k and s_{k+1}. The target is likely to be inside the intersection of the measurement wedges.

1.2 RELATED LITERATURE

With the use of UAV, the target signal can be detected better without being blocked by obstacles such as trees and hills. Besides, it is possible to operate regardless of the difficulties of terrains, and thus the localization can be done fast [Körner et al., 2010, VonEhr et al., 2016]. The UAVs make use of various information to locate the target signal source. Among these are range-only based on the received signal strength (RSS), the bearing-only based on the direction of the signal, and the use of both range and bearing.

Calculating the target distance based on the signal strength (range-only) may not always give good results. For example, there may be various obstacles between the signal source and the robot. The target may be behind a building, wall, or underground. The characteristics of distance information is highly dependent on environmental conditions [Bayram et al., 2017]. In this case, the signal strength is attenuated, hence does not give us accurate information.

[Cliff et al., 2015] used a two-point phased array antenna and a specially designed signal capture circuit to obtain the directional angle. With their proposed strategy, the authors can reduce localization uncertainty up to 50 meters without guaranteeing localization performance. The advantage of the system we propose is that the antenna can be easily built and the signal receiver is low-cost and available in the market.

In the EKF-based online target localization algorithm in [Vander Hook et al., 2014], a fish tagged with a radio transmitter is localized based on bearing measurements. Meanwhile, the algorithm provides a performance guarantee by comparing the online strategy with the best offline strategy. However, the proposed approach requires an initial estimate for the target location and a sensor noise model. Their choices affect performance. In addition, the system is based on ground and surface vehicles for a stationary target. In our proposed system, we focus on aerial robots for a moving target. Furthermore, it is ensured that our approach is superior to the specified system by aiming to eliminate the dependency on the sensor noise model and initial estimation.

In the work [Bayram et al., 2016a], signals are taken from a radio transmitter using an omnidirectional antenna. If there are beeps in the recorded signal, the UAV is not out of range of the transmitter and the localization process starts. The decrease or increase in the amplitude of the current signal noise by comparison with the last signal noise means that the UAV is approaching or moving away from the transmitter. Thus the location of the transmitter can be determined. At the beginning, the approach requires calibration. However, in our approach, the system performs localization and tracking without requiring any calibration. In addition, it is not assumed that the target is stationary, and the approach will be able to track moving targets.

In another study [Bayram et al., 2018], an online strategy is proposed without requiring a sensor noise model. In this study, the target is within a certain angular aperture of the direction of the measurement. Localization is done by the intersections of these obtained measurements. it is also assumed that the target is stationary. The performance guarantees are given based on this assumption. We make use of the similar measurement technique, but we relax the assumption on the movement of the target. Thus, we can localize and track stationary and moving targets.

The work in [Isaacs et al., 2014] presents a particle filter based approach to track a target with a circular motion in a laboratory environment. The noise model is assumed to be known. This work uses a directional antenna. However, the motion models of the target in our work are Random Walk, Heterogeneous Behavior, Velocity Model [Hooten et al., 2017] and Lévy Walk [James et al., 2011] instead of a circular trajectory in a small area. Furthermore, the target's velocity and acceleration are not fixed in this thesis.

The received signal strength indicator is used in target localization along with particle filter in [Wagle and Frew, 2010, Nguyen et al., 2019, Hasanzade et al., 2018]. The target is a stationary RF source [Wagle and Frew, 2010, Hasanzade et al., 2018], and the system has 200m localization performance in [Wagle and Frew, 2010]. Also, 10 mobile targets as wombats are localized, and their motion models are chosen as Random Walk in [Nguyen et al., 2019]. The system validations have been done with simulations and/or software in the loop experiments in [Nguyen et al., 2019, Hasanzade et al., 2018], except for the work in [Wagle and Frew, 2010] that uses measurement data gathered from the field flights. Then, these data are used as input to the particle filter. On the contrary, the targets are moving with random velocities and accelerations in this thesis. As mentioned above, there are four different motion models, including Random Walk, for the target in simulations. We validate the algorithm and system performance by performing field experiments for stationary and moving targets along with extensive simulations.

1.3 CONTRIBUTION OF THE THESIS

Our main contributions in this thesis are as follows:

- We integrate bounded measurement models with the particle filter in such a way that the particles' weights are updated based on a novel method which considers the measurement wedge and the particles' location along with a logistic function.
- We incorporate the doubling strategy into the particle filter in order to determine the next measurement locations.
- The proposed approach is validated using a target moving with varying velocity and acceleration. For this purpose, a simulator is written in Python language and four different motion models are adopted (Heterogeneous Behavior, Lévy

Walk, Random Walk and Velocity Model) in these simulations, which are used in modeling animal movements in wildlife studies.

- We developed a low-cost hardware and software infrastructure for the proposed tracking system.

1.4 APPROACH

Our aim is to track and localize an RF emitting target in this work. This target has no fixed velocity and acceleration in the simulations and experiments. However, it has a maximum limited velocity. In addition, the target also has no fixed trajectory. An online and bearing-only strategy has been chosen for this work.

As the target has an RF emitting device, a low-cost Yagi-Uda antenna with an SDR is used to catch the signal. According to the antenna directionality sensitivity, the field is divided into a number of measurement wedges and this adds noise to the bearing of the target. The UAV takes samples from each measurement wedges with the attached antenna by rotating around itself. Later, these samples are used to determine the noisy bearing of the target.

Due to the non-linear nature of the localization and tracking of this target, a Particle Filter is used to estimate the location of the target. Particle Filter's measurement update is done according to the measurement area, which is assumed to have the target. And particles' propagation are made by the Guided Walk that is developed for this work.

After each measurement update, the UAV calculates the next measurement location towards the bisector of the bearing angle. Distance to the next measurement location is halved or doubled according to the last and current bearing angles difference.

This approach is tested in the simulator that is written for this work. A UAV was built for the field experiments. After the simulations, parameters such as the number of particles, are determined and field experiments were performed.

1.5 OUTLINE OF THE THESIS

This thesis is composed of 5 chapters. This first chapter has an introduction that includes the problem statement, related literature, contribution of the thesis, proposed approach, and outline of the thesis. The sensing model of the approach and Particle Filter based moving target localization and tracking algorithm are explained in Chapter 2. The target motion models are described in Chapter 3. Detailed information is given about the simulations and their results as well. The experimental platform is explained with its hardware and software components in Chapter4. Also, the field experiments are represented with their results in this chapter. These chapters are followed by Chapters 5 and 6 which are Conclusion and Appendix respectively.

CHAPTER 2: TARGET TRACKING

We aim to localize and track a moving target that has a signal source on it. In order to estimate the target location, a particle filter based tracking approach is proposed. This approach relies solely on bearing measurements. When the UAV reaches the measurement location, it takes bearing measurements by rotating itself towards pre-defined angels. These measurements are received by the Yagi-Uda antenna and SDR connected to the antenna. For each measurement, signal strength is calculated. The bearing angle is obtained such that the direction gives the maximum signal strength. This bearing angle is then used to calculate the posterior distribution of the particles. Following the posterior distribution calculation, a new measurement location is computed by using the bearing angle.

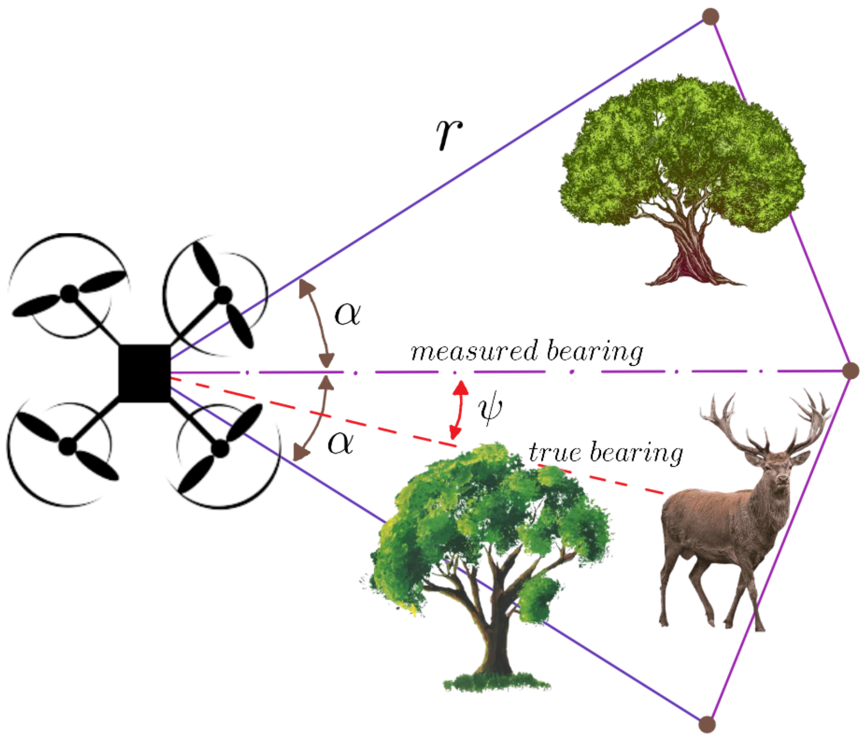

Figure 2.1: Sensing model. The measurement area is represented by two triangular areas on each side of the bisector. The target is known to be likely located in this area.

2.1 SENSING MODEL AND BEARING MEASUREMENTS

When there is no exact sensor model, the bounded noise model can be more effective than probabilistic models. Therefore, we use the bounded noise model in which the measurements produce a bearing angle corrupted with a bounded noise α in each bearing measurement as shown in Figure 2.1. The value of α is determined after the field tests and selected above the real antenna sensitivity in order to stay more reliable. A bearing measurement wedge is assumed to be made of two triangular areas with bounded noise angle left, right edges, and bisector with the length of r. This wedge consisting of a polygonal area is considered to be the most likely area to contain the target.

The UAV takes bearing measurement samples simply by rotating around itself in measurement locations. These bearing measurements are represented as wedges composed of two triangles with α radian clearance. Therefore, this rotation is discrete and the UAV takes π/α number of samples to complete a measurement. Then, these samples are used to determine the bearing measurements such that they have the maximum signal strength in that direction.

2.2 PARTICLE FILTER BASED MOVING TARGET LOCALIZATION AND TRACKING

Bayesian Filters are a regular choice for the state estimation problems such as the localization of a target. The Extended Kalman Filter can be used for nonlinear problems as it linearizes the system about the predicted state. However, the system must be unimodal, and the posterior probability distribution function must be Gaussian [Posch and Sukkarieh, 2009, Soriano et al., 2009].

Among non-parametric Bayesian methods are histogram filters and particle filters. These non-parametric filters estimate the posterior distribution of the system by a finite number of values that represent the system states. Both filters can reduce the approximation error uniformly to zero while the number of values goes to infinity. Histogram filter is prone to use high computational sources. As for the particle filter, its application is easy, and it is the most adaptable among the Bayesian filter algorithms. Particle filters use randomly selected samples from the posterior to present the current posterior of the system. These samples are called particles. [Thrun et al., 2005]. The sampling of the particles is done proportionally to their accompanied weights. However, there are some problems with the particle filters. These problems are degeneracy, sample impoverishment, and particle filter divergence. There are a few methods aimed to prevent these problems, such as resampling for degeneracy and adding adequate process noise for sample impoverishment. The work [Elfring et al., 2021] can be viewed for more details.

Our tracking algorithm is based on the particle filter which is represented in Algorithm 2.4. The tracking process starts in an area with r meters of radius. The algorithm has the following main parts: Particle initialization, particle weight calculation, resampling, particle propagation, and calculations of the estimated target and next measurement locations. Each of these parts will be explained in detail.

2.2.1 Particle Initialization

All particles are spread uniformly over a circular area according to Equation 2.1 where d is the particle's distance to the origin, and ϕ is the angle of the line connecting the particle's location and the origin. x and y are the particle's location calculated by d and ϕ parameters.

$$
\begin{aligned}
&r_d \sim \mathcal{U}(0,1) \\
&d = r\sqrt{r_d} \\
&\phi \sim \mathcal{U}(-\pi,\pi) \\
&x = dcos(\phi),\ y = dsin(\phi)
\end{aligned}
\tag{2.1}
$$

where r is the radius of a circular area in which the target moves.

2.2.2 Particle Weights

After climbing to a predefined altitude and arriving at the measurement location, the UAV starts to take a bearing measurement, resulting in π/α samples. The bearing angle β to the target is obtained from the sample direction which has the maximum signal strength among the others. This angle β is corrupted by the measurement noise angle α in each side.

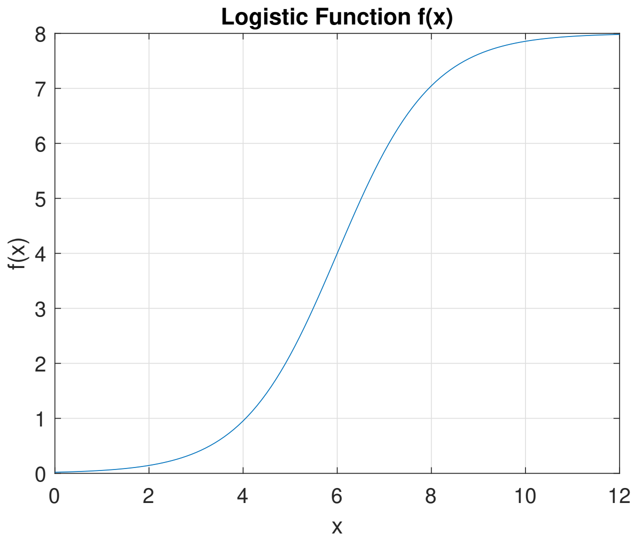

Figure 2.2: Logistic function for the values of $\lambda = 1$, $L = 8$, $x_0 = 6$ and $x = [0, 12]$.

Then, the bearing angle β is used to calculate the weights of particles in line 11 of Algorithm 2.4 (The details of the weight calculation is given in Algorithm 2.1). To calculate the weight of a particle, we make use of the logistic function in Equation 2.2. With the use of the logistic function, we obtain the weights. In the weight calculation, the following three cases represented in Figure 2.3 are considered based on the particle's position with respect to the measurement wedge and the bisector:

- If the particle is inside the wedge, the weight is calculated based on the particle's distance to the bisector and to the closer edge of the measurement wedge,
- If the particle is not inside the wedge, the weight is calculated according to the particle's distance to the closer edge of the measurement wedge,
- If the particle is behind the UAV, the weight is calculated according to the distance between the particle and the UAV.

The distances calculated in these cases above are used as an input argument to the logistic function.

$$f(x) = \frac{L}{1 + e^{\lambda(x - x_0)}} \tag{2.2}$$

where x_0 is value of the sigmoid's midpoint, L is the curve's maximum value and λ is the logistic growth rate or steepness of the curve. The effect of change in the parameters can be seen in Figure 2.2.

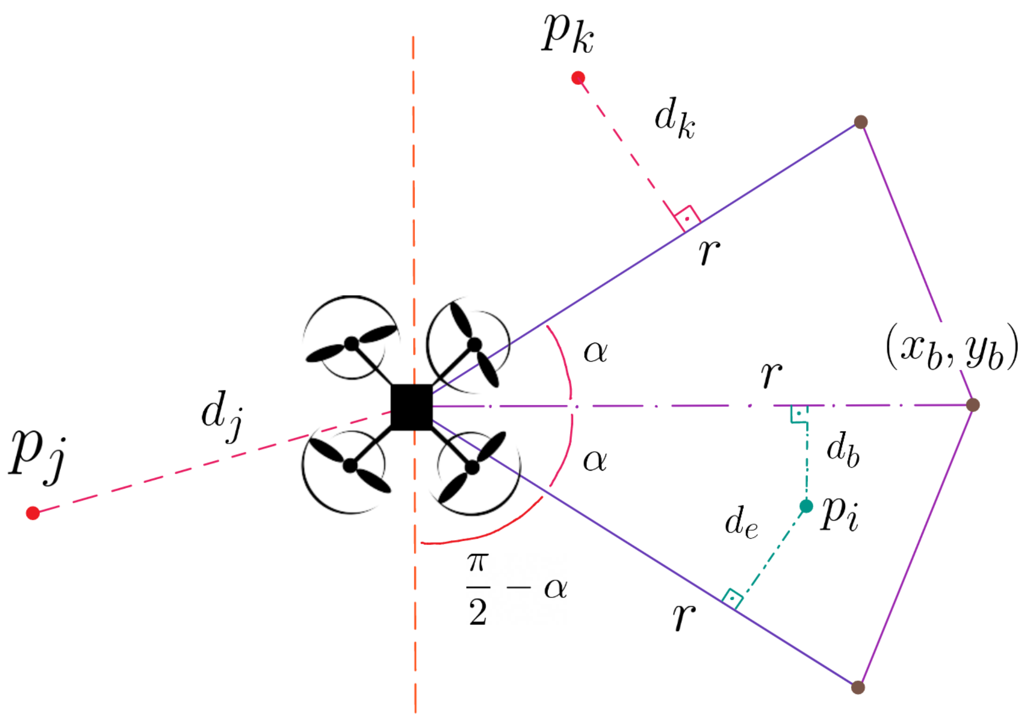

Figure 2.3: Different positions of the particles inside and outside of the measurement wedges. (x_b, y_b) is the coordinate of the end point of the measurement wedge's bisector.

The possible locations of a particle with respect to its corresponding measurement wedge are illustrated in Figure 2.3. In the first case, if a particle (p_i) is inside the measurement wedge, the parameter x in the logistic function is defined as the ratio of the particle's distance to the edge d_e and the particle's distance to the bisector d_b. To prevent low weight values inside the measurement wedge, $\lambda = 1$ and $x_0 = 1$ are chosen. The second and third cases are related to the particles that are outside of the wedge. In the second case, if the particle (p_k) is not behind the UAV, $x = -d_k$ where d_k is the particle's distance to the edge of the measurement wedge. In the third case, if the particle (p_j) is behind the UAV, then $x = -d_j$ where d_j is the particle's distance to the UAV. In this case, $\lambda = 1$ and $x_0 = 0$ are chosen to make weights of the particles low and disappear quickly. λ and x_0 values of the logistic function in Equation 2.2 are determined based on the extensive simulation results.

Algorithm 2.1: `Calculate-Weight`

Input: Particle list P, bearing angle β, UAV location

foreach $p \in P$ **do**
    if p is inside the measurement wedge whose bisector angle is β **then**
        $\lambda \leftarrow 1$ and $x_0 \leftarrow 1$ in Equation 2.2
        $p.w = f(d_e/d_b)$
        // d_e is the particle's distance to the nearest left or right side of the wedge d_b, particle's distance to bisector
    else
        $\lambda \leftarrow 1$ and $x_0 \leftarrow 0$ in Equation 2.2
        if particle p is not behind the UAV **then**
            $p.w = f(-d_k)$
            // d_k is the particle's distance to the nearest edge of measurement wedge
        else
            $p.w = f(-d_j)$
            // d_j is the particle's distance to the UAV location
return P

2.2.3 Particle Resampling

The resampling is carried out when the particles' new weights are calculated. The particle weights denote the probability of the sampling of these particles in the resampling step. Hence, the particles with higher weights are more likely to be selected and the particle filter will overcome the degeneracy problem. The resampled particles' weights are then normalized as seen in the lines 13-15 of Algorithm 2.4. It guarantees that the particles' weights sum up to 1.

2.2.4 Particle Propagation

After the resampling process, the particles are propagated using one of the selected motion models, namely Guided Walk and Random Walk. Guided Walk is the particles' motion model that is used in simulation and experiments. Random Walk is used for comparison with Guided Walk. According to the simulations, Guided Walk has better performance and is chosen for the experiments. Random Walk used as the particle propagation model is shown in Algorithm 2.2. It is the same as the target's Random Walk motion model. However, when Random Walk is used as the particle propagation model, the resulting tracking performance varies much for the motion models of the target.

Algorithm 2.2: `Particle-Random-Walk`

Input: Particle list P

// 2σ rule where 95% percent of the particles velocity will be under the particle velocity limit $p.limitVel$

$\sigma^2 \leftarrow \frac{p.limitVel}{2}$

foreach $\underline{p \in P}$ **do**

  $x_r \sim \mathcal{N}(0, \sigma^2)$

  $y_r \sim \mathcal{N}(0, \sigma^2)$

  $\epsilon_x \leftarrow x_r \times p.velocity$

  $\epsilon_y \leftarrow y_r \times p.velocity$

  // Propagate the particle p

  $p.x \leftarrow p.x + \epsilon_x$

  $p.y \leftarrow p.y + \epsilon_y$

return P

Guided Walk is a motion model that is developed for this thesis to be used in propagating the particles and its algorithm is given in Algorithm 2.3. Each particle moves towards a point on the bisector of the measurement wedge where the signal is the strongest. In order to do that, the bisector equation is generated from the bisector slope and the beginning of the bisector point which is the UAV's location. The x coordinate of the random point x_r is chosen uniformly, and the bisector equation is used to calculate the random point's y coordinate y_r as seen in line 6. The angle from the particle to this point is calculated. Then, the Gaussian noise is added to this angle to prevent sample impoverishment as shown in line 8, especially when this angle is too low. The standard deviation of the noise is set to $\pi/3$ radians based on the simulation results. Finally, the particles move towards the calculated angle. The particles' velocities are calculated as shown in the line 10 of Algorithm 2.3.

Based on the simulation results, the tracking performance drops when the UAV overshoots the target, and the bearing angle β to the target changes significantly. In this situation, even though the target is moving in the same direction, the new measurement leads particles to the opposite path. So, the particles are propagated in the opposite direction of the target. Until the particle filter catches the target again, the tracking error grows. In order to reduce this error, the angle difference of the current and last bisector angle is used. If this angle is more than $5\pi/12$ and less than π, particles' velocities are

Algorithm 2.3: `Particle-Guided-Walk`

Input: Particle list P

// let (x_{UAV}, y_{UAV}), (x_b, y_b)

// where (x_{UAV}, y_{UAV}) is the coordinate of the UAV. and (x_b, y_b) is the coordinate of the end of bisector line

$bisectorSlope \leftarrow \frac{y_{UAV} - y_b}{x_{UAV} - x_b}$

$bisectorAngle \leftarrow \text{arctan}(\frac{y_{UAV} - y_b}{x_{UAV} - x_b})$

$angleDiff \leftarrow |p.lastBisectorAngle - bisectorAngle|$

foreach $p \in P$ **do**

    // Draw a random point over bisector of measurement wedge from uniform distribution

    $x_r \sim \mathcal{U}(x_{UAV}, x_b)$

    // Calculate y

    $y_r \leftarrow bisectorSlope \cdot (x_r - x_{UAV}) + y_{UAV}$

    // Calculate angle of the line connecting the particle and the random point on the bisector

    $angleDeviation \sim \mathcal{N}(0, (\frac{\pi}{3})^2)$

    $angle \leftarrow \text{arctan}(\frac{y_r - p.y}{x_r - p.x}) + angleDeviation$

    // 2σ rule where 95% percent of the particles velocity will be under the particle velocity limit $p.limitVel$

    $velocityDeviation \sim \mathcal{N}(0, (\frac{p.limitVel}{2})^2)$

    $p.velocity \leftarrow |p.velocity + velocityDeviation|$

    // Limit particle velocity

    if $p.velocity < p.limitVel$ **then**

        $p.velocity \leftarrow p.limitVel$

    // If new measurement angle is opposite side of the last measurement, reduce particle velocity

    if $\frac{5\pi}{12} < angleDiff < \pi$ **then**

        $p.velocity \leftarrow p.velocity/3$

    $p.x \leftarrow p.x + p.velocity \times cos(angle)$

    $p.y \leftarrow p.y + p.velocity \times sin(angle)$

$p.lastBisectorAngle \leftarrow bisectorAngle$

return P

divided by 3 stated in the line 14 of Algorithm 2.3. These upper and lower bounds are determined to avoid the problem mentioned above based on the results of the simulations done with varying bounds for $angleDiff$. Reducing the particles' velocity (line 14) is applied when the angle difference between the current and last bisectors is out of the bounds.

2.2.5 Target's Estimated Location and Next Measurement Location

The target's location is estimated by the weighted sum of the particle coordinates as in the lines 16-17 in Algorithm 2.4.

After the particles are propagated, the next measurement location is calculated in between of the lines from 22 to 33 in Algorithm 2.4. How to determine the next measurement locations is a crucial step in the tracking algorithm. Since we do not have any initial guess about the target's position, the target can be arbitrarily far away from the UAV's initial location. Taking this into consideration, we would like to keep the number of measurements bounded. Hence, we adapt the doubling strategy, which is used in solving online problems[1] such as linear search or lost-cow problems [Baeza-Yates et al., 1993, Plonski et al., 2017]. The doubling strategy is employed as long as the target is not overshot. In the doubling strategy, the Euclidean distance between the current and next measurement locations is determined by d_{step}, which is doubled after each measurement. Once the target is overshot, the strategy is changed to halving, which means that the distance between the current and next measurement locations is calculated by halving the previous distance. The condition for detecting the situation of overshooting the target is that the angle between the previous bearing β_{prev} and the current bearing β_{curr} is greater than $\pi/2$ as shown in Equation 2.3.

$$|\beta_{curr} - \beta_{prev}| > \pi/2 \tag{2.3}$$

[1]Unlike offline problems, in online problems we do not have access to all the inputs in advance. Instead, we are provided with inputs in a sequential fashion and have full knowledge of the past inputs. Under these circumstances, online algorithms are designed to solve online problems such that the designed algorithms are competitive with the optimal offline algorithm having perfect knowledge of the future inputs [LaValle, 2006].

Algorithm 2.4: `Track-Moving-Target`

Input: number of particles N, measurement noise α, step size d_{step}, min step size d_{min}, $targetMotionModel$

$strategy \leftarrow$ DOUBLING
// Initialize particles
for $i = 1$ **to** N **do**
    Each particle $p_i(x_i, y_i, w_i)$ is drawn using Equation 2.1
    Add p_i to P list
// First measurement location
$x_{meas} = x_{UAV}$
$y_{meas} = y_{UAV}$
while True **do**
    Move to the measurement location (x_{meas}, y_{meas})
    Take π/α samples from discrete directions
    Determine the bearing angle β
    // Calculate the weights of the particles
    $P \leftarrow$ `Calculate-Weight`$(P, \beta, [x_{UAV}\ y_{UAV}])$
    // Each weight $p_i.w$ is treated as the probability of obtaining the particle p_i at resampling step
    Perform the resampling
    // Normalize particles weights
    $w_T = \sum_{i=0}^{N} w_i$
    foreach $p \in P$ **do**
        $p.w = p.w/w_T$
    // Calculate target's estimated location
    $x_{est} = \sum_{i=0}^{N}(p_i.x \times p_i.w)/N$
    $y_{est} = \sum_{i=0}^{N}(p_i.y \times p_i.w)/N$
    if $targetMotionModel = Guided - Walk$ **then**
        $P \leftarrow$ `Particle-Guided-Walk`(P)
    else
        $P \leftarrow$ `Particle-Random-Walk`(P)
    // Calculate the distance between the next and current measurement locations
    if the UAV overshoots the target **then**
        $strategy \leftarrow$ HALVING
    if $strategy$ is HALVING **then**
        if $d_{step} > d_{min}$ **then**
            $d_{step} = d_{step}/2$ // Halving strategy
        else
            $d_{step} = d_{min}$
            $strategy \leftarrow$ DOUBLING
    else
        $d_{step} = 2d_{step}$ // Doubling strategy
    // Determine the next measurement location
    $x_{meas} = x_{UAV} + d_{step}cos(\beta)$
    $y_{meas} = y_{UAV} + d_{step}sin(\beta)$

CHAPTER 3: SIMULATIONS

In order to evaluate the proposed approach, we conducted a series of extensive simulations. In these simulations, we consider four different motion behaviors for the target which are commonly used to model the animal motion: Heterogeneous Behavior, Lévy Walk, Random Walk, and Velocity Model (the details of the motion behaviors will be given below). In addition, the measurement noises of $\pi/6$, $\pi/4$, and $\pi/3$ radians are used as the antenna directionality sensitivities. For the particle filter, the calculations are done with 200, 500, and 1000 particles. The particles can be propagated based on two different motion models: Random Walk and Guided Walk. Random Walk and Guided Walk algorithms are shown in 2.2 and 2.3, respectively. To determine which model is suitable for the particle propagation, we have conducted a series of simulations. Based on the results of these simulations, we have observed that Guided Walk has better localization and tracking performance than Random Walk. Thus, Guided Walk is chosen as the motion model for the particle propagation in the simulations and experiments.

Each simulation contains 50 measurements. For each target motion model, simulations are repeated 25 times. These simulations were done for each measurement noise and number of the particles. For the performance assessment of the developed tracking algorithm, two measures are used: Euclidean tracking error and distance between the UAV position (x_{UAV}, y_{UAV}) and the target's position. The tracking error is the distance between the target's true location (x_{target}, y_{target}) and its estimated location (x_{est}, y_{est}) which calculated by the particle filter. These measures are defined in Equation 3.1.

$$\begin{aligned} Tracking\_Error &= \sqrt{(x_{UAV} - x_{est})^2 + (y_{UAV} - y_{est})^2} \\ Distance\_btw\_UAV\_and\_target &= \sqrt{(x_{UAV} - x_{target})^2 + (y_{UAV} - y_{target})^2} \end{aligned} \tag{3.1}$$

The UAV's maximum speed is set to 10 m/s. The target and particle speeds are drawn from Gaussian distribution $\mathcal{N}(0, \sigma^2)$. The value of $\sigma = 2.5$ guaranties that the speed stays under 5 m/s with a 95.45% probability. The target's speed is compatible with the red deer speed (Cervus Elaphus) that lives in the Turkey as well, for an example of a target. The speeds of the red deer are between 0.74 m/s and 2.88 m/s in the study [Brockway and Gessaman, 1977]. The target's initial locations are placed uniformly in the workspace with a radius of 1000 meters as shown in Figure 3.1. The target's trajectories were generated with the motion behaviors for every 25 simulations as well. Hence, all the behaviors use the same generated trajectories to eliminate the effect of trajectory differences over the simulation results.

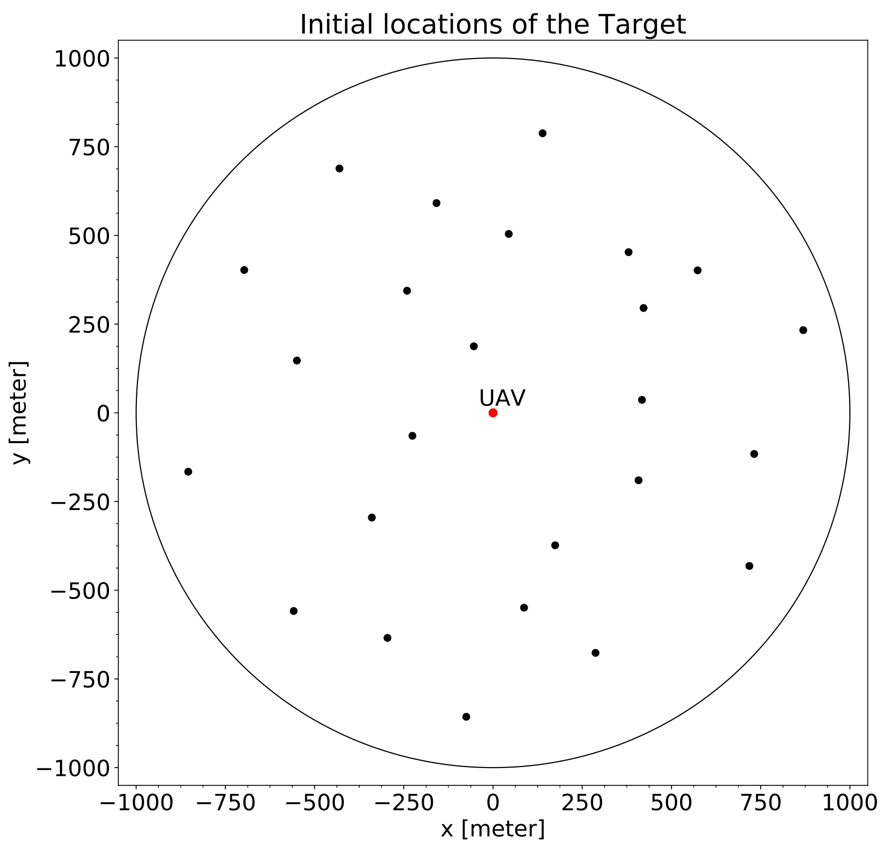

Figure 3.1: Target's uniformly distributed initial locations used in the simulations where the UAV starts at the origin.

3.1 TARGET MOTION MODELS

As the target moves in this work, its motion needs to be simulated. Since our focus is specifically on wild animals, two kinds of models are used to imitate the animal motion. The first model is the position based models which include Random Walk, Lévy Walk, and Heterogeneous Behavior. Lévy Walk is a member of Random Walk. The second model is the Velocity Model. These models are discrete time models.

Random Walk is a motion model whose step size is generated through a Gaussian distribution. The target generally moves in a local area.

In the Velocity Model, the target's position is calculated using its velocity. After the velocity is generated through a Gaussian distribution, it is used to calculate the target's displacement within unit time.

In Heterogeneous Behavior, the target moves randomly between the predefined attraction points. These points are shown in Figure 3.2.

Lévy Walk is the most aggressive model among the target's motion models. While the target moves in a small area, the destination may suddenly change to a distant location. Moreover, the target moves straight to the its destination.

For all the behaviors, the particles' maximum speed is set to 5 m/s and normal distribution with $\sigma = 2.5$ generates 95.45% below the maximum speed. If the particles' speed exceeds the maximum speed, the speed is trimmed to the maximum speed.

3.1.1 Random Walk

In Random Walk, the true location μ of the target at time t depends on all of the previous locations but only through its nearest neighbors in time. The last and previous locations of the target determines the target's trajectory [Hooten et al., 2017, p. 147]. The target's location μ_t at time t can be modeled as:

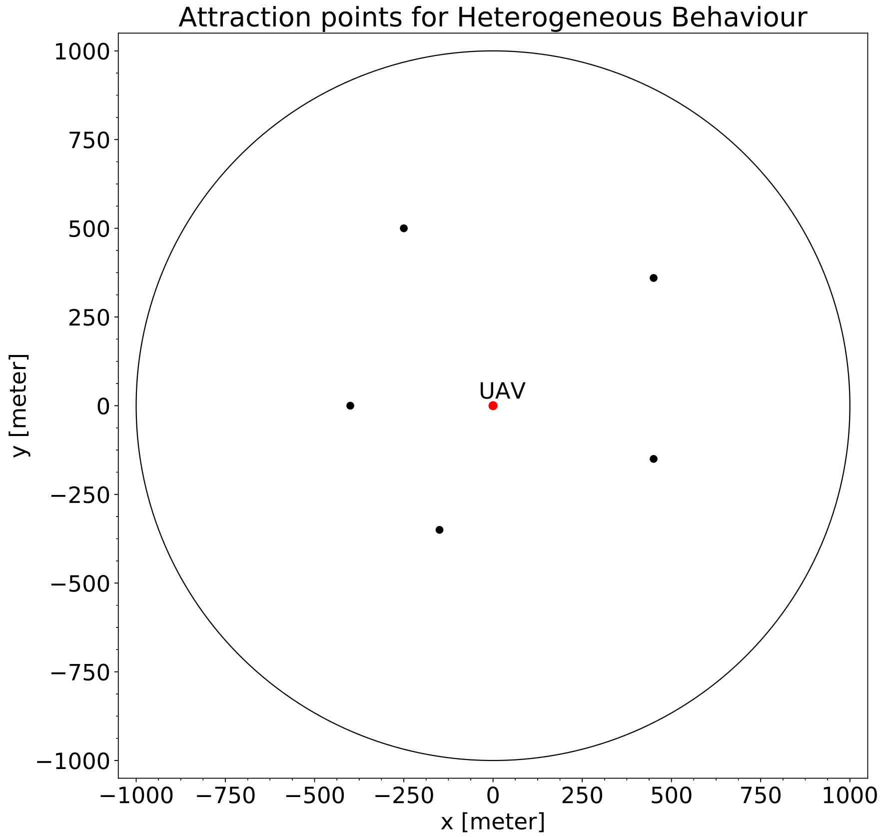

Figure 3.2: Attraction points used for generating the target's trajectory in the Heterogeneous Behavior.

$$\begin{bmatrix}\mu_{t,x}\\ \mu_{t,y}\end{bmatrix} = \begin{bmatrix}\mu_{t-1,x}\\ \mu_{t-1,y}\end{bmatrix} + \begin{bmatrix}\varepsilon_{t,x}\\ \varepsilon_{t,y}\end{bmatrix} \tag{3.2}$$

where ε_t is sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$ and I is a 2x2 identity matrix. A sample of the Random Walk trajectory can be seen in Figure 3.3. σ is set to 2.5 in the simulations.

3.1.2 Heterogeneous Behavior

In wild life, there are some attraction points like water supplies and repulsion points such as residential areas. If these points are a finite set of values as $\mu_1^*, ..., \mu_j^*, ...\mu_J^*$

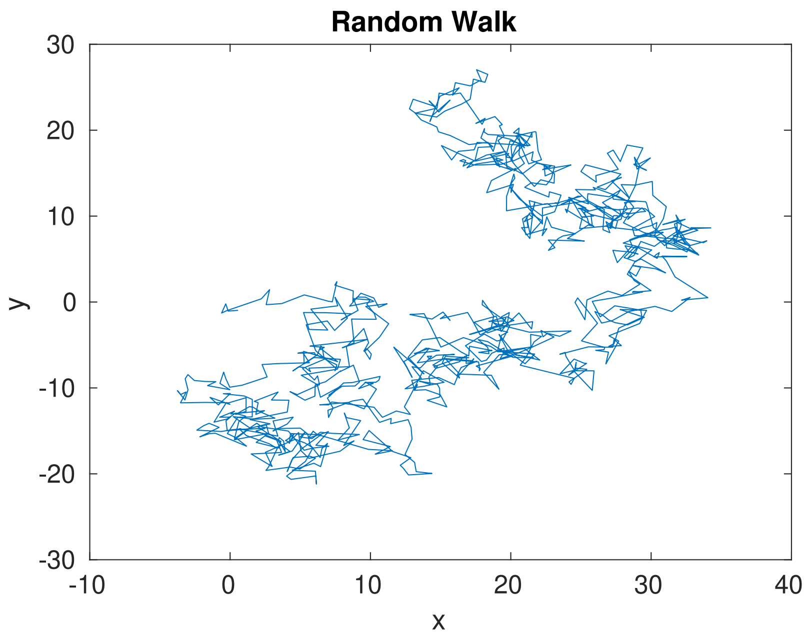

Figure 3.3: Target's trajectory generated using Random Walk for $\sigma^2 = 1$ and 1000 steps.

and are known before, μ_t^* model is given as:

$$\mu_t^* = \begin{cases} \mu_1^*, Prob(\mu_1^*) = p_1 \\ \quad\vdots \\ \mu_J^*, Prob(\mu_J^*) = p_J \end{cases} \tag{3.3}$$

where the probabilities $p_1, .., p_J$ of the attraction points sum to one [Hooten et al., 2017, p. 153].

Using this model for μ_t^*, μ_t can be updated with Equation 3.4.

$$\mu_t = M\mu_{t-1} + (I - M)\mu_t^* + \varepsilon_t \tag{3.4}$$

where I is a 2x2 identity matrix and ε_t is a noise term sampled from a Gaussian distribution $\varepsilon_t \sim N(0, \sigma^2 I)$. σ is set to 2.5 in the simulations. M is calculated from $M = \rho I$, and ρ is used to adjust the weight of the attracting point with M in Equation 3.4. ρ is set to 0.998 based on the simulation observations. If ρ is small, the target's motion results in a straight trajectory towards an attraction point. A sample trajectory of the Heterogeneous Behavior can be seen in Figure 3.4.

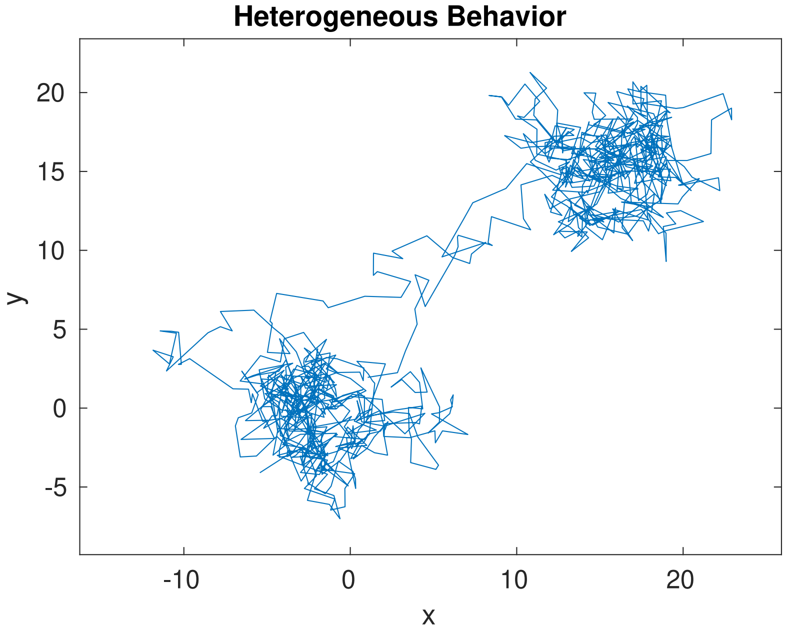

Figure 3.4: Target's trajectory generated using Heterogeneous Behavior for $\sigma^2 = 1$, $\rho = 0.95$, $\mu_1^* = (15, 15)$ for the first 500 steps and $\mu_2^* = (-2, 2)$ for the second 500 steps.

3.1.3 Velocity Model

As an alternative to the direct calculation of position, obtaining position via velocity is used in this method [Hooten et al., 2017, p. 158]. The velocity vector at time t is

$$\begin{bmatrix} V_{t,x} \\ V_{t,y} \end{bmatrix} = M \begin{bmatrix} V_{t-1,x} \\ V_{t-1,y} \end{bmatrix} + \begin{bmatrix} \varepsilon_{t,x} \\ \varepsilon_{t,y} \end{bmatrix} \tag{3.5}$$

where $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ and M is given as:

$$M = \gamma \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \tag{3.6}$$

The angle θ in Equation 3.6 controls the rotation between the consecutive steps and relates the rotation angle with the target's displacement. The rotation angle is bounded between $-\pi$ and π. When θ approaches to 0, the target moves forward. γ is between 0 and 1 and controls the weight of the M matrix in the velocity dynamic Equation 3.5. The resulting equation for the velocity model is

$$\begin{bmatrix} V_{t,x} \\ V_{t,y} \end{bmatrix} = \gamma \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \begin{bmatrix} V_{t-1,x} \\ V_{t-1,y} \end{bmatrix} + \begin{bmatrix} \varepsilon_{t,x} \\ \varepsilon_{t,y} \end{bmatrix} \tag{3.7}$$

where θ and γ can be tuned if the target's motion model is known. The target's location at time t is calculated from Equation 3.8 as follows:

$$\mu_t = \mu_{t-1} + v_t \Delta_t \tag{3.8}$$

The target's trajectories generated for $T = 1000$ steps, $\Delta_t = 1\, sec$, and $\varepsilon \sim \mathcal{N}(\mu_\varepsilon, \sigma_\varepsilon^2)$ where $\mu_\varepsilon = 0$, $\sigma_\varepsilon^2 = 2.5$, which controls the step length of the target, and varying θ and γ are demonstrated in Figure 3.5 and 3.6. In the simulations, θ and γ are set to 0.1π and 0.1, respectively, as chosen in [Hooten et al., 2017, p. 160].

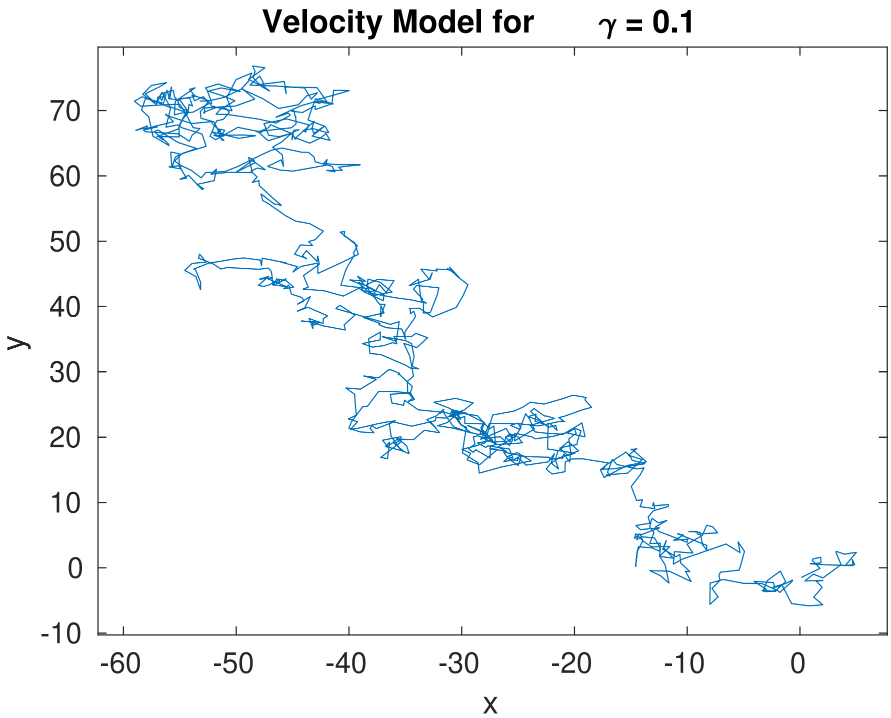

Figure 3.5: Target's trajectory generated using Velocity Model for $\theta = 0.1\pi\, rad$ and $\gamma = 0.1$ in 1000 steps.

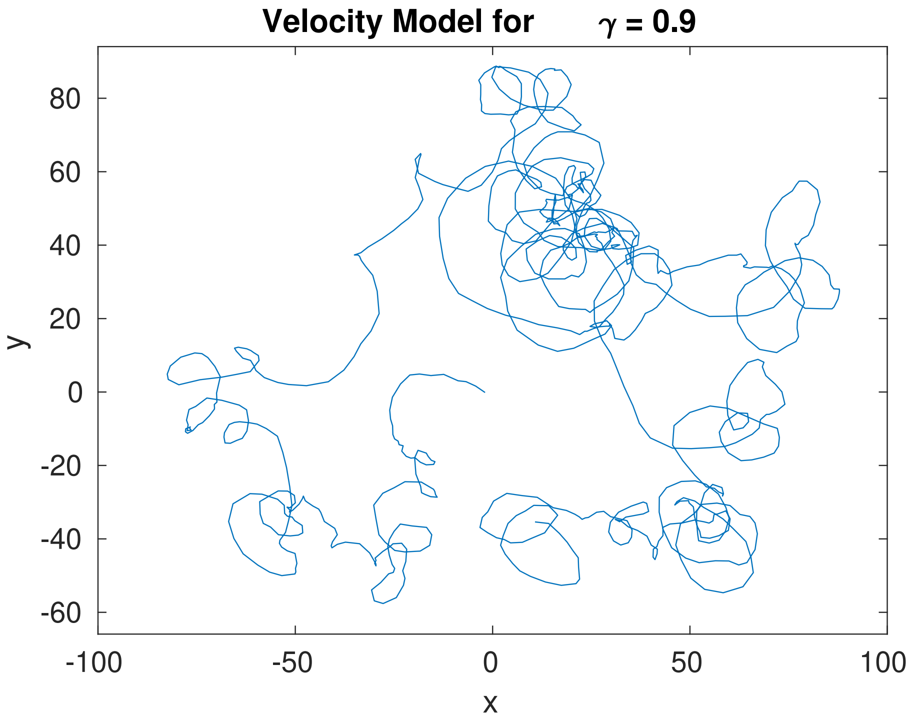

Figure 3.6: Target's trajectory generated using Velocity Model for $\theta = 0.1\pi\, rad$ and $\gamma = 0.9$ in 1000 steps.

3.1.4 Lévy Walk

Lévy Walk is a subclass of Random Walk. The target's movement or walk distance is determined by using one of the heavy-tailed distributions such as the power-law / pareto, Cauchy, Lévy. This motion model is used for analyzing human walk patterns and modeling animal movement patterns [James et al., 2011]. An example trajectory generated using the Lévy Walk is demonstrated in Figure 3.7. The algorithm of Lévy Walk used in simulations is shown in Algorithm 3.1.

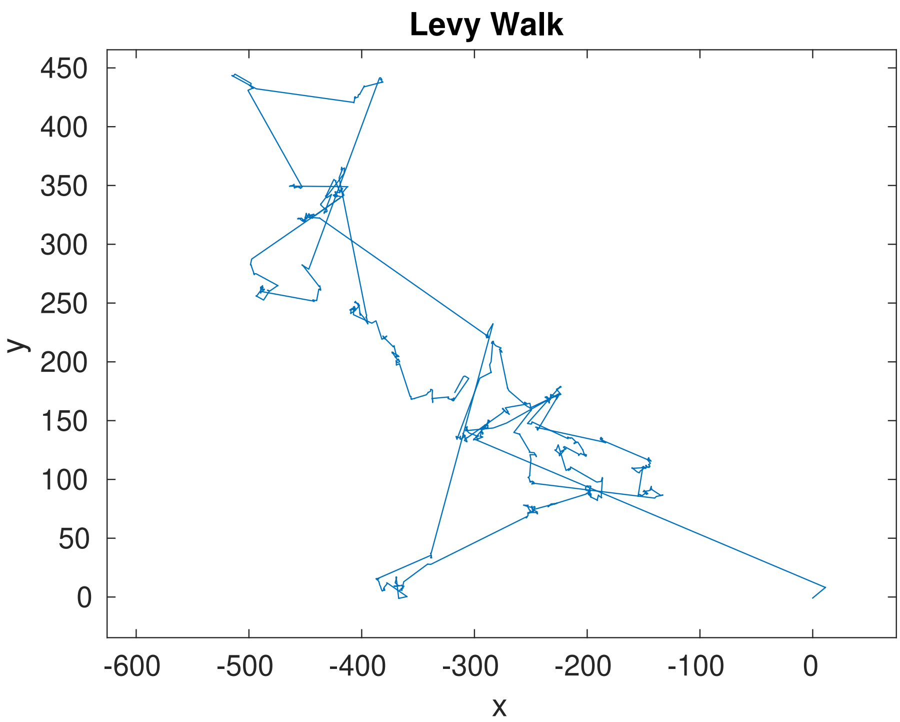

Figure 3.7: Target's trajectory generated using Lévy Walk for $\mu = 1.4$ and $c = 0$ in 1000 steps.

Algorithm 3.1: `Lévy-Walk`

Input: $target$

// let (x_{dest}, y_{dest}), where (x_{dest}, y_{dest}) is the destination for the target.

// let $\mu \leftarrow 1.4$ and $c \leftarrow 0$ where μ is the location parameter and c is the scaling parameter in the Lévy distribution.

if $target$ reaches the destination **then**

  // Draw a random distance from Lévy distribution

  $d_r \sim Levy(\mu, c)$

  // Draw a random angle from uniform distribution

  $\theta_r \sim \mathcal{U}(0, 2\pi)$

  $x_{dest} \leftarrow x_{UAV} + d_r \cos(\theta_r)$

  $y_{dest} \leftarrow y_{UAV} + d_r \sin(\theta_r)$

// Calculate the angle from the target to the destination

$angle2Dest \leftarrow \arctan(\frac{y_{dest} - target.y}{x_{dest} - target.x})$

$V_x \sim \mathcal{U}(0, target.velocity/\sqrt{2})$

$V_y \sim \mathcal{U}(0, target.velocity/\sqrt{2})$

$target.x \leftarrow target.x + V_x \cos(angle2Dest)$

$target.y \leftarrow target.y + V_y \sin(angle2Dest)$

return $target$

3.2 SIMULATION RESULTS

In order to evaluate the simulations, the distance between the target's true and estimated locations is used as a tracking error for the moving target case. These error calculations consider the target's estimated and true locations at which the measurement update is done. To have a fair comparison, the total number of measurement the UAV can takes is limited to 50 measurements. Therefore, each simulation will have 50 error values to be used in calculating the statistical results of the tracking error. For instance, Random Walk is assessed with 3 measurement noise levels and for each measurement noise level, 25 simulation trials are done such that each simulation contains 50 measurements for 200 particles. The same is repeated for 500 and 1000 particles. Hence we have (3 measurement noise levels) $\times$ (25 simulations) $\times$ (50 measurements) error values for each number of particles.

In order to have the statistical analysis of the tracking performance, the error values are used to calculate the mean and standard deviation values. For instance, when the mean and standard deviation values of Random Walk for 200 particles and $\pi/6$ measurement noise are calculated, their error values are used to generate the tables for the tracking error. The total number of the tracking errors used is (1 Random Walk) $\times$ (1 measurement noise level) $\times$ (25 simulations) $\times$ (50 measurements) = 1250.

The samples from the simulation results are demonstrated in Figure 3.8 for Heterogeneous Behavior, Figure 3.9 for Lévy Walk, Figure 3.10 for Random Walk and Figure 3.11 for Velocity Model. In these simulations, the measurement noise and the number of particles are set to $\pi/6$ radians, and 500, respectively. Note that Guided Walk is used as the particles' motion model in all the simulations.

Table 3.1 shows the effects of particle numbers and the measurement noises in terms of the tracking performance which is represented as the mean and standard deviation of the tracking error for each target's motion behavior. The increase in the measurement noise results in an increase in the mean and standard deviations of tracking errors. In general, the best tracking performance is obtained for all the measurement noise levels when the target's motion is Random Walk. This is because the target's motion is local. The tracking performance of Velocity Model comes after Random Walk. Although Velocity Model has a similar trajectory with Random Walk, the target with Velocity Model wanders around in a wider area. Heterogeneous Behavior has a more straight path among attraction points and its tracking performance is in third place. The highest tracking errors and standard deviations are obtained with Lévy Walk because of its aggressive and long straight motion.

The target's speed and acceleration differ during the simulations. The average values of the target's speed and acceleration are approximately 3 m/s and 1.5 m/s^2, respectively.

The lowest mean and standard deviation results of tracking error are achieved for Random Walk in Table 3.1 when the measurement noise is $\pi/6$ and the number of particles is 200. However, with these settings, Lévy Walk exhibits the worst tracking performance. This is because of the insufficient number of particles. When the measurement noise is set to $\pi/4$ and $\pi/3$, the best results are obtained using Heterogeneous Behavior whose motion leads to relatively straight and slower paths than Lévy Walk. The particles move more straight in case of the high measurement noise levels. Since the guided motion attracts the particles to the bisector of the measurement area, the bisector does not often change its direction due to the high noise values at each measurement. Therefore, the particles follow the target efficiently when the target's motion is Heterogeneous Behavior.

For 500 particles as observed in Table 3.1, the mean and standard deviations of the tracking error are comparatively less than the case when the number of particles is 200 and the measurement noise is $\pi/6$ radians especially for Lévy Walk. The tracking error has almost the same mean and standard deviation values for 200 and 500 particles with the measurement noise of $\pi/4$ and $\pi/3$ radians.

For 1000 particles as viewed in Table 3.1, the mean and standard deviation of the tracking errors are similar with 500 particles. Therefore, to avoid the computation load, 500 particles can be chosen instead of 1000 particles.

To sum up, there is an important relation between the measurement noises and the tracking algorithm performance for all the target's behaviors. Considering a trade-off between the tracking performance and the computation power, the number of particle can be chosen as 500 since there is no significant difference in terms of the tracking performance for 500 and 100 particles.

The correlation between each motion model and particle number can be seen from Table 3.2 for Heterogeneous Behavior, Table 3.3 for Lévy Walk, Table 3.4 for Random Walk and Table 3.5 for Velocity Model.

The mean and standard deviations of the tracking error for Heterogeneous Behavior are given in Table 3.2. Heterogeneous Behavior is in the third place in terms of the mean and standard deviation of the tracking errors. These tracking errors are minimum for $\pi/6$ with 500 and 1000 particles. When the number of particles increases, the tracking error mean and standard deviations are almost the same except for $\pi/3$. The mean of the tracking error decreases while the number of particles rises, although the standard deviation stays similar just for the measurement noise of $\pi/3$. As expected, when the measurement noise grows, the tacking performance deteriorates.

Table 3.1: Tracking Error (Mean & Standard Deviation in meters) for all Behaviours with all number of Particles.
(HB: Heterogeneous Behavior, LW: Lévy Walk, RW: Random Walk, VM: Velovity Model)

				Target Motion Models			
				HB	LW	RW	VM
Number of Particles	200	Meas. Noise	$\pi/6$	51.28 8.51	65.09 31.39	47.25 11.57	49.02 9.07
			$\pi/4$	58.98 17.81	88.68 64.32	69.13 58.20	79.07 71.26
			$\pi/3$	97.38 70.52	98.01 77.08	99.59 92.54	88.87 84.62
	500	Meas. Noise	$\pi/6$	49.88 8.83	59.40 16.52	45.45 8.35	47.41 8.48
			$\pi/4$	58.20 18.60	86.75 52.41	70.44 64.78	79.88 80.76
			$\pi/3$	93.07 69.10	102.57 79.30	99.58 89.70	89.88 86.27
	1000	Meas. Noise	$\pi/6$	49.83 9.26	56.66 14.05	46.03 8.35	48.01 8.67
			$\pi/4$	57.81 20.07	87.15 51.95	70.77 62.95	78.31 76.49
			$\pi/3$	90.74 69.47	101.61 77.68	99.42 91.65	90.40 87.83

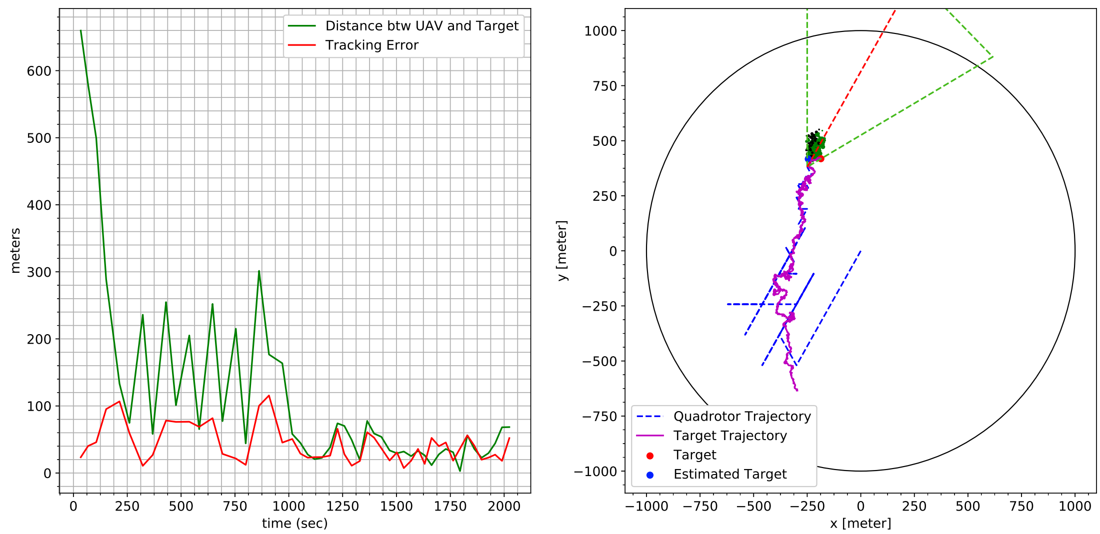

Figure 3.8: Simulation result for the setting in which the target's motion model, measurement noise and number of particles are set to **Heterogeneous Behavior**, $\pi/6$, and 500, respectively. The UAV takes 50 measurements. The average tracking error is about 42 meters.

Table 3.2: Tracking Error (Mean & Standard Deviation in meters) for Heterogeneous Behavior (HB)

HB		Number of Particles		
		200	500	1000
Meas. Noise	$\pi/6$	51.28 8.51	49.88 8.83	49.83 9.26
	$\pi/4$	58.98 17.81	58.20 18.60	57.81 20.07
	$\pi/3$	97.38 70.52	93.07 69.10	90.74 69.47

The tracking error values for Lévy Walk are presented in Table 3.3. The tracking performance is not quite appealing for Lévy Walk in which the target generally moves long distances along a straight trajectory. When the UAV overshoots the target, it takes more time to converge to the target. When the varying levels of measurement noise are examined, the mean and standard deviations of the tracking error increase with the increase in the measurement noise. As to the number of particles, the mean and standard deviations decrease with the increase in the number of particles when the measurement noise is set $\pi/6$. It seems that this type of the target's motion needs a more specific particle motion model.

Table 3.3: Tracking Error (Mean & Standard Deviation in meters) for Lévy Walk (LW)

LW		Number of Particles		
		200	500	1000
Meas. Noise	$\pi/6$	65.09 31.39	59.40 16.52	56.66 14.05
	$\pi/4$	88.68 64.32	86.75 52.41	87.15 51.95
	$\pi/3$	98.01 77.08	102.57 79.30	101.61 77.68

Tracking Error mean and standard deviation values are the lowest for Random Walk as seen in Table 3.4 for measurement noise of $\pi/6$ with all number of particles. Even though there is a notable growth in tracking error means with the increase in measurement noise, when the measurement noise changes from $\pi/6$ to $\pi/4$, the corresponding standard deviations for 200, 500, and 1000 particles increase significantly (6 to 9 times). The change in the number of particles does not have an important effect on the tracking performance when the measurement noise is kept the same. This is because of Random Walk's motion behavior. Its movement is generally local and the number of particle do not improve the tracking performance after the particles converge to the

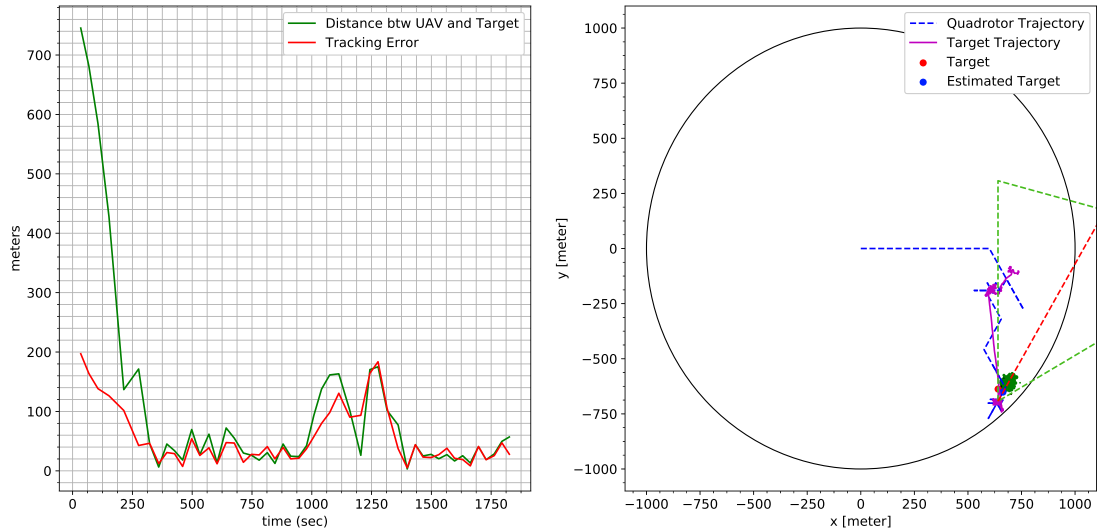

Figure 3.9: Simulation results including tracking error, distance between the target and the UAV, and trajectories of the target and the UAV for **Lévy Walk**, $\pi/6$ measurement noise, and 500 particles. The UAV takes 50 bearing measurements. The average tracking error is 55 meters.

target.

Table 3.4: Tracking Error (Mean & Standard Deviation in meters) for Random Walk (RW)

RW		Number of Particles		
		200	500	1000
Meas. Noise	$\pi/6$	47.25 11.57	45.45 7.65	46.03 8.35
	$\pi/4$	69.13 58.20	70.44 64.78	70.77 62.95
	$\pi/3$	99.59 92.54	99.58 89.70	99.42 91.65

The statistical results for Velocity Model are represented in Table 3.5. The tracking performance of Velocity Model is the second after Random Walk in regard to means and standard deviations. Although tracking errors of Velocity Model are slightly above of Random Walk for $\pi/6$, standard deviations are just below of Random Walk. There is growth in tracking error means and standard deviations with the measurement noise increment. When particle numbers increase, again, there are no significant changes.

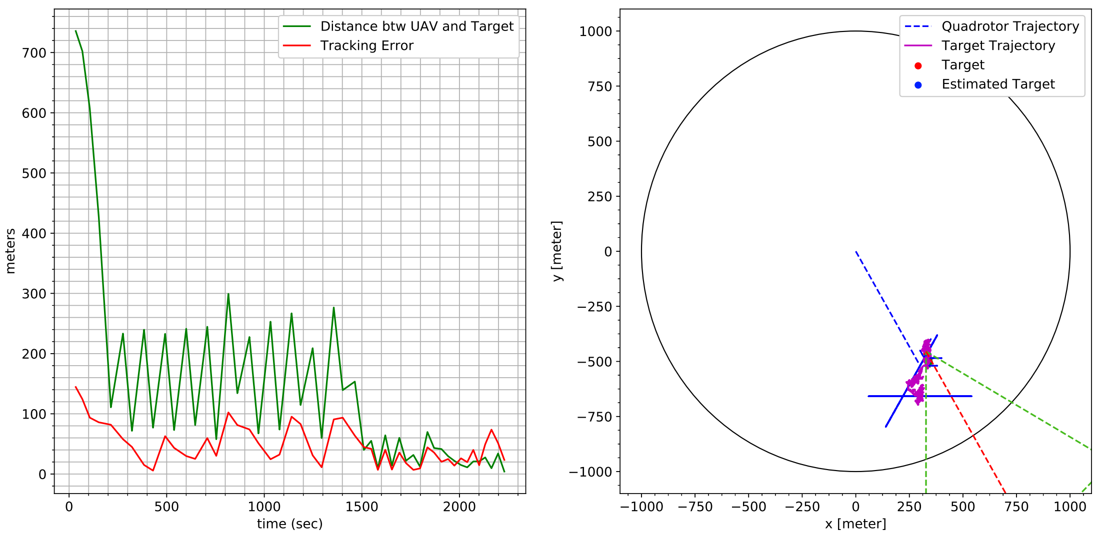

Figure 3.10: Simulation output for **Random Walk**, $\pi/6$ measurement noise, and 500 particles. The UAV takes 50 bearing measurements. The average tracking error is about 47 meters.

Table 3.5: Tracking Error (Mean & Standard Deviation) for Velocity Model (VM)

<table>
<tr><td colspan="2" rowspan="2">VM</td><td colspan="3">Number of Particles</td></tr>
<tr><td>200</td><td>500</td><td>1000</td></tr>
<tr><td rowspan="3">Meas. Noise</td><td>$\pi/6$</td><td>49.02
9.07</td><td>47.41
8.48</td><td>48.01
8.67</td></tr>
<tr><td>$\pi/4$</td><td>79.07
71.26</td><td>79.88
80.76</td><td>78.31
76.49</td></tr>
<tr><td>$\pi/3$</td><td>88.87
84.62</td><td>89.88
86.27</td><td>90.40
87.83</td></tr>
</table>

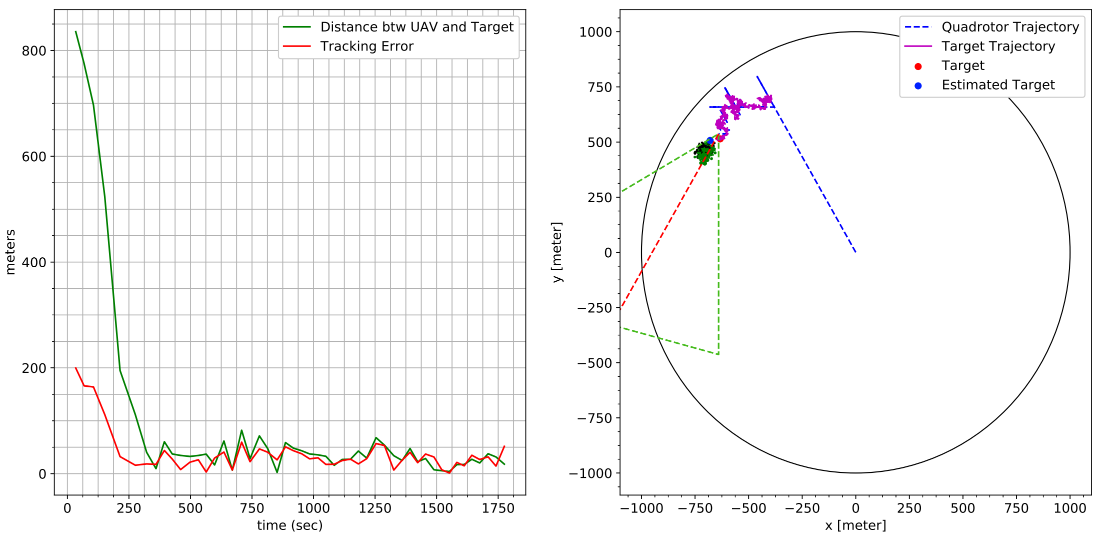

Figure 3.11: Simulation output for **Velocity Model**, $\pi/6$ measurement noise, and 500 particles. The UAV takes 50 bearing measurements. The average tracking error is about 38 meters.

CHAPTER 4: FIELD EXPERIMENTS

We implemented the proposed approach on a multirotor UAV system. To achieve this, we developed hardware and software modules, whose details will be given in the following sections. By using the developed system, field experiments are conducted to validate the performance of the approach.

4.1 EXPERIMENTAL PLATFORM

Both hardware and software components of the experimental platform are developed for this thesis.

4.1.1 Hardware Components

The physical system is primarily over the UAV. The system consists of a Li-Po battery, 4 electronic speed controllers, and brushless motors, a flight computer, a Jetson Nano, the antenna, XBee Pro 2.4 GHz as telemetry, and an SDR device in general. A PC that runs Windows 10 OS with telemetry, is used as the Ground Station. The hardware diagram of the UAV is shown in Figure 4.1.

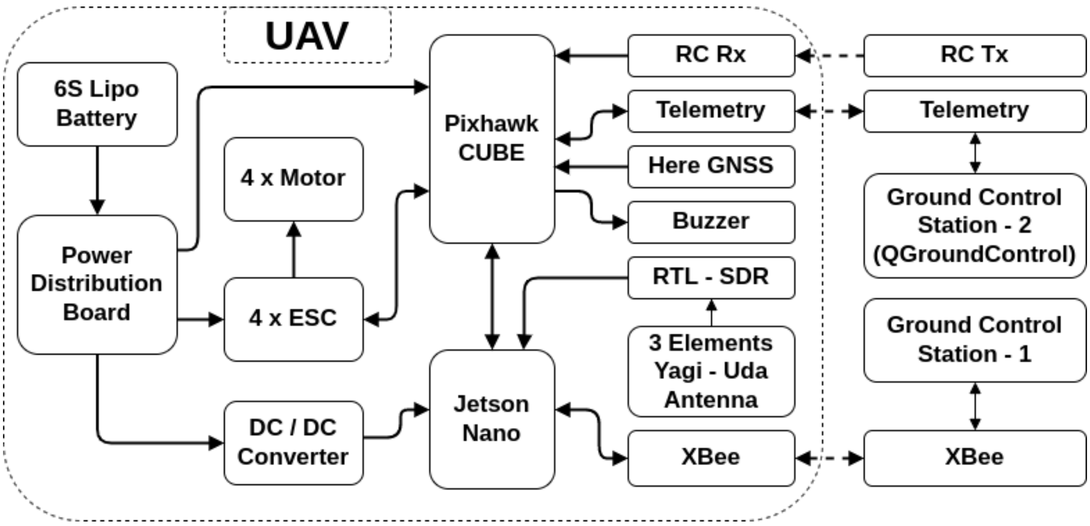

Figure 4.1: Hardware diagram of the system.

- The UAV is a custom-made quadrotor, consisting of carbon fiber and 3D printed parts. It uses The Cube which runs Pixhawk 2 firmware with a standard carrier board for autopilot and Here2 GNNS as seen in Figure 4.2.

Figure 4.2: A custom-made quadrotor UAV with an RTL-SDR receiver, VHF antenna and on-board computer.

- 3-element Yagi-Uda antenna is also a low-cost custom made antenna shown in Figure 4.3 to capture 433MHz signals transmitted by the RF transmitter module on the target.

- The target trajectory logger has a Garmin GPS 18x attached to a Raspberry Pi 3 Model B+ in order to record the target's trajectory. The Raspberry Pi runs 16.04.7 LTS (Xenial Xerus) with ROS as well, hereby ROS communicates with the GPS and a written ROS package logs the current location of the target. The target hardware diagram is illustrated in Figure 4.4.

- A low-cost radio transmitter system acted as a target was developed as seen in Figure 4.5. In this system, PT2262 RF transmitter equipped with Arduino Nano is utilized as a transmitter to generate and transmit the pulses.

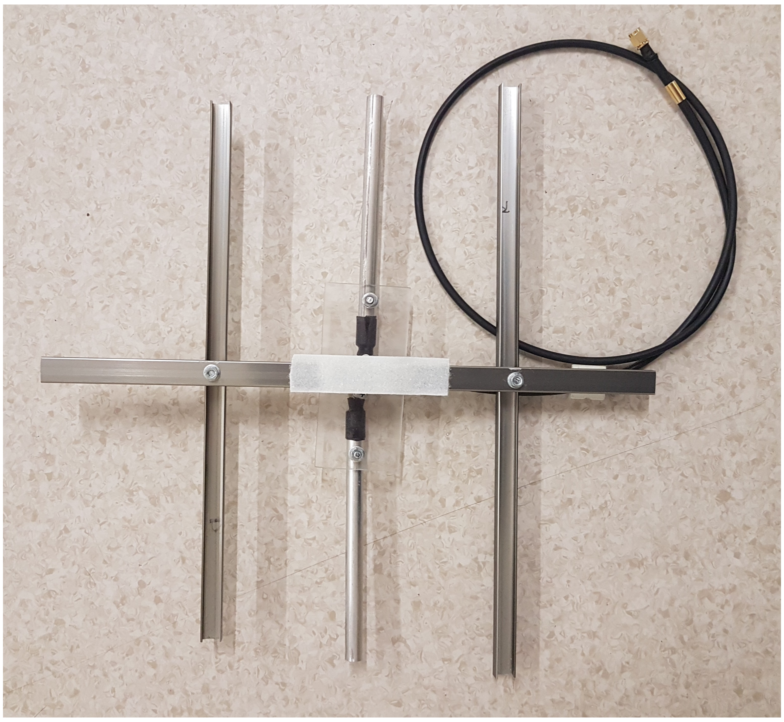

Figure 4.3: 3-Element Yagi-Uda Antenna.

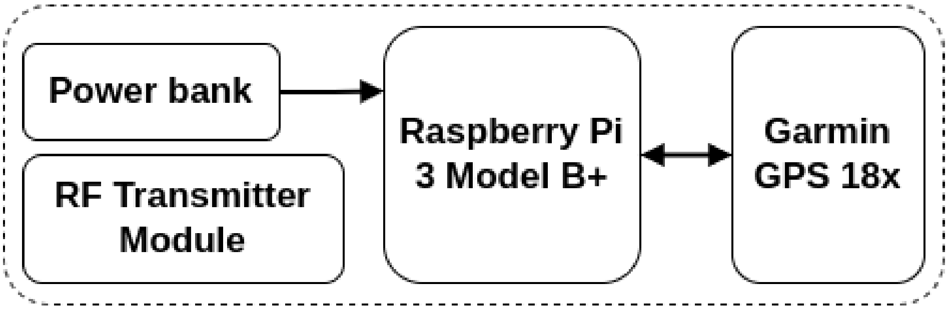

Figure 4.4: Hardware diagram of the target. The target composed of the RF transmitter module and the target trajectory logging system.

- A low-cost RTL-SDR (Software-Defined Radio - SDR) is used as a receiver and tuned to the transmitter frequency. A 3-element Yagi-Uda antenna (Figure 4.3) is used to capture the signal. The recorded signal is in an IQ signal data format and `rtl-str`[1] software records this data using RTL2832 based on DVB-T receivers.

[1]rtl-sdr software is available at http://osmocom.org/projects/sdr/wiki/rtl-sdr

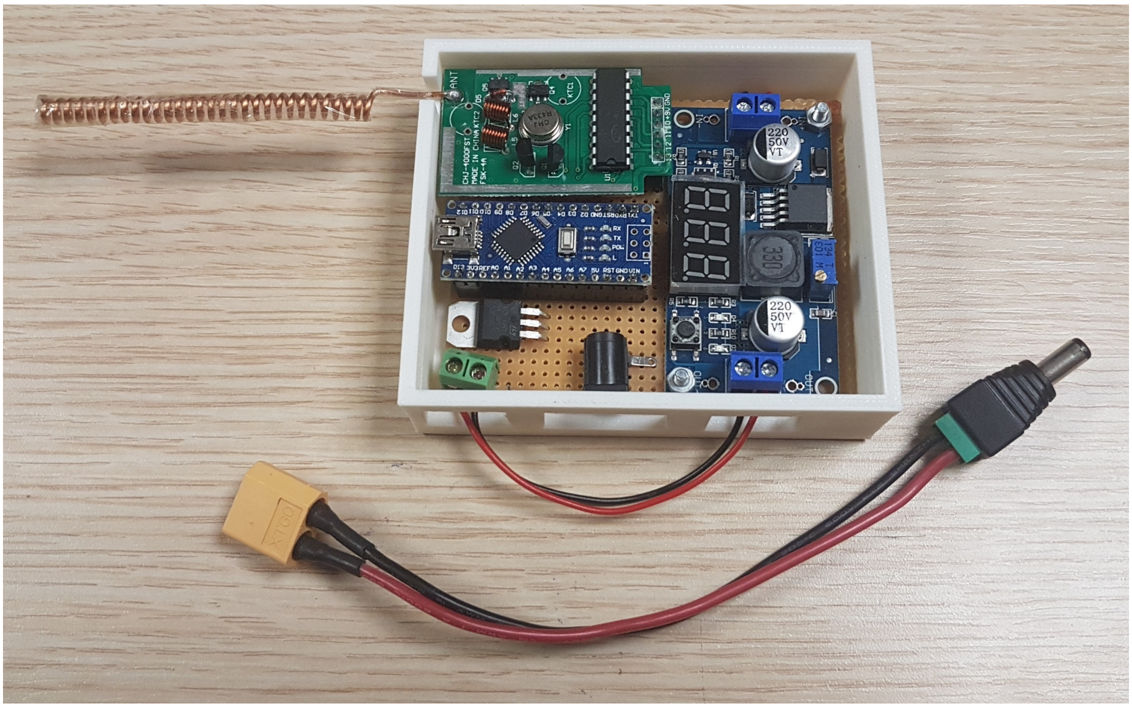

Figure 4.5: Transmitter module composed of Arduino Nano to generate the signal, RF transmitter, DC-DC converter, and regulator.

4.1.2 Software Components

The system consists of ROS nodes and the Ground Station. The ROS nodes are written in C++ and Python. The Ground Station is written in C# to observe and control the system via telemetry. QGroundControl[2] is used only to calibrate the sensors and adjust the safety parameters of the Pixhawk Cube.

Each ROS node takes care of one part of the system. The main ROS node named `Target-Tracking` is responsible for running the tracking algorithm and coordinating the other nodes. The ROS nodes are illustrated in Figure 4.6. All nodes and the Ground Station are explained in detail below.

- *ROS Framework*

  With the help of an on board computer and ROS [Open Robotics,] Melodic which runs on Ubuntu 18.04.5 LTS (Bionic Beaver), all the processing is done online. The online algorithms which were developed for the system combined as ROS nodes on this framework. A Raspberry Pi 3B+ that runs Ubuntu 16.04.7 LTS

[2]QGroundControl is available at http://qgroundcontrol.com/

(Xenial Xerus) is also used to interface with the Garmin GPS module to log the trajectory of the target.

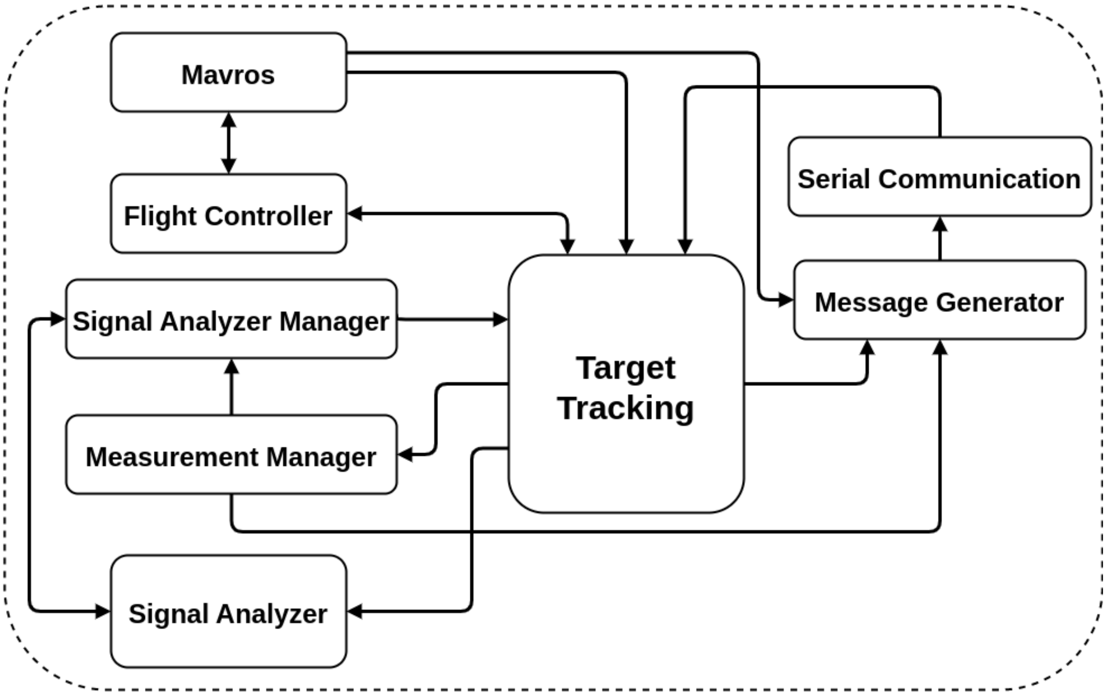

Figure 4.6: The block diagram of the ROS nodes for the target tracking software implemented on the UAV.

The diagram of the ROS nodes is illustrated in Fig. 4.6. Mavros node [Ermakov,] is an interface between MAVLink messaging protocol and ROS framework. This node makes essential information about the flight and UAV status accessible in the ROS framework.

The `Flight Controller` node's aim is flying the quadrotor. The `Measure- ment Manager` node interfaces with SDR and takes measurements. The `Signal Analyzer Manager` node uses Signal Analyzer service, collects results, and sends them to the `Target Tracking` node. The `Signal Analyzer` is a service and its task is analyzing the recorded signals and generating necessary data about the signal. The `Target Tracking` node is the main node and executes the main algorithm presented in Algorithm 2.4 (`Track-Moving- Target`). The `Serial Communication` node is an interface between ROS and Xbee telemetry device. The `Message Generator` node generates messages to be sent via the telemetry.

- *Signal Analyzer*

Each of the completed sample records are processed, while rest of the samples are taken in a measurement location. Later, an envelope of the processed signal is generated. After generating the envelope signal, a pulse signal is created according to hysteresis up and low thresholds. Pulses begin when their amplitudes exceed an up threshold and finish when less than a low threshold [Teledyne SP Devices, 2018]. In order to detect the bearing of the target, average pulse amplitudes are used. Hence it is directly proportional to the heading of the UAV and the distance between the UAV and the target. To analyze the recorded signal, `Signal Analyzer` was coded in C/C++ language using Armadillo library [Sanderson and Curtin, 2016]. A 3-second sample signal was recorded in an I/Q data format at 30 meter altitude and the distance to the target was 334 meters (Figure 4.7). The sample signal is plotted in Figure 4.8. First, since the signal is in the imaginary format, the magnitude of each data point in the signal is computed. Next, the signal analyzing algorithm applies a low-pass filter on the magnitude of the signal. An envelope (the dark blue line) is generated to smooth the resulting signal. In order to detect pulses (magenta color) in the envelope signal, we define thresholds represented with green, red and black colored lines. Green and red thresholds are placed according to the black line as trigger and reset levels, respectively. If the signal amplitude exceeds the trigger level (green line), the algorithm considers it the start of a pulse. When it is below the reset level (red line), the algorithm considers it the end of a pulse. After that, the average pulse amplitude (orange line) is calculated using the detected pulses. The remaining part of the signal excluding the pulses is regarded as noise.

- *Target Tracking*

  The tracking process starts with taking π/α samples (in our case, $\alpha = \pi/6$) in the measurement location. After taking samples, the system evaluates each sample and gets the bearing angle to the target. This angle is used in calculating the next measurement location and in the measurement update step of the particle filter. By considering the desired tracking performance and the battery limitation, the values of the step size d_{step} and the minimum step size d_{min} are set to 40 and 20 meters, respectively. This step size is then doubled or halved according to the previous and current bearing angles as shown in the line 22 of the Algorithm 2.4. This tracking process continues unless it is terminated.

Figure 4.7: Signal sample taken from 30 meters height from ground and 334 meters distance to target. The UAV is in the start of the line, and the target is in the point where distance is shown on. This field is also the area where the experiments are performed.

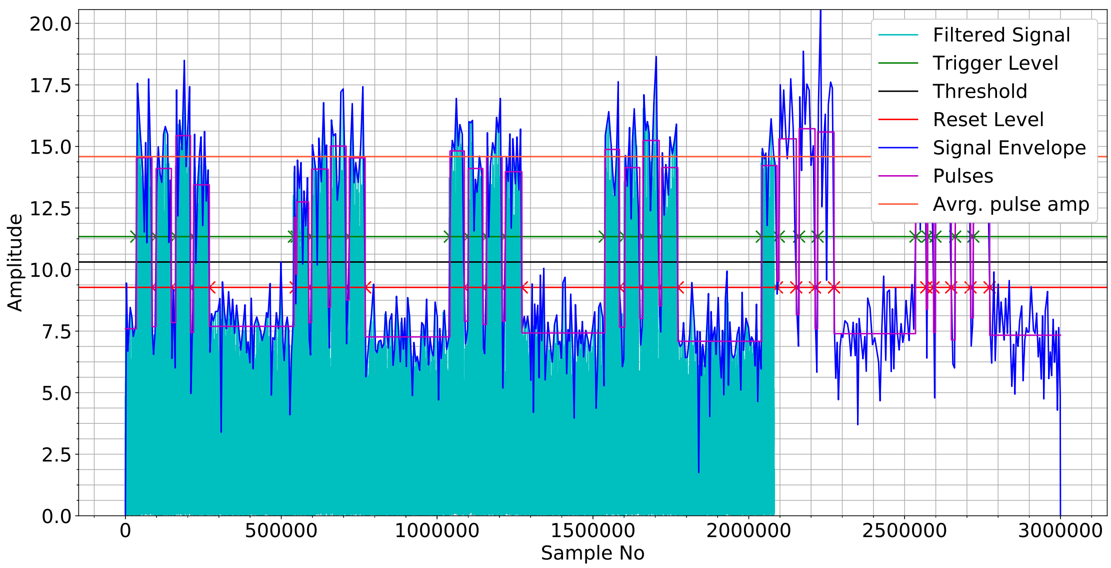

Figure 4.8: Signal sample taken from the altitude of 30 meters from ground and 334 meters distance to target. Orange line is average pulse amplitude. Yellow line is up, red is low threshold and black is the mean of the whole signal. Dark blue line is envelope of the raw signal and light blue is the signal.

- *Ground Station*

  The ground station software was written in C# for Windows OS. Through the application, the UAV trajectory, the estimated target, and the next measurement location can be monitored over a map. The robot and experiment status are observed via the robot status panel. The ground station also logs the flight data during the experiments. For the generic missions, some custom locations can be picked up over the map or read from a file and sent to the UAV or written to a file. On the “Communication” tab, the communication between the UAV and the ground station can be established and monitored.

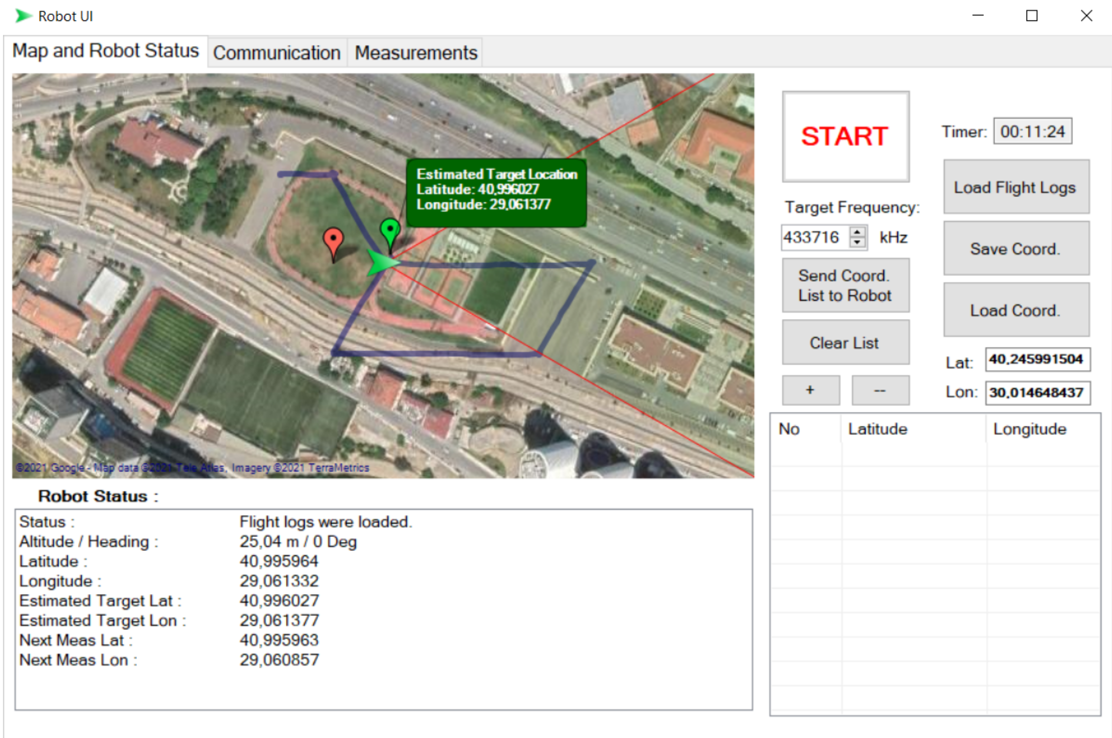

Figure 4.9: Ground station software for monitoring the necessary information regarding the experiments. The green marker shows the target’s estimated location and the red marker is the next measurement location. The measurement area is also shown on the map with green color surrounded by red line.

4.2 EXPERIMENTAL RESULTS

Before the experiments, SDR, antenna, and RF transmitter module are located at various distances. Received signals are recorded as IQ signals from the varying measurement angles with the antenna in these distances. The characteristics of these taken signal samples were analyzed. Thus, the bearing noise α is determined as $\pi/6$. To better detect the pulses in the received signals, we optimized the signal processing algorithm running in `Signal Analyzer`. The number of samples for each bearing measurement is 6 with $\pi/3$ radians of rotation, which means that the UAV takes samples from 0, $\pi/3$, $2\pi/3$, π, $4\pi/3$, $5\pi/3$ radians.

We performed six experiments for each of stationary and moving target scenarios. The distance between the UAV and the target was an average of 270 meters in the stationary experiments. The UAV step size d_{step} is set to 40 meters in the experiments. The step size is doubled or halved according to the measurements results by `Track-Moving-Target` algorithm. The number of particles is set to 500 according to the simulation observations. The UAV altitude is chosen as 25 meters to avoid any collision since the trees and buildings in the experiment area are below this altitude. Measurement numbers for each experiment vary between 4 and 7 in stationary and moving targets. The first three localization errors of the stationary and moving target experiments are ignored in the results of Table 4.1 and 4.2 since the target is far away at the beginning. After the third measurement update, the target's estimated location converges faster to the target's true location and tracks it within a bounded error.

The field experiments were conducted in Istanbul Medeniyet University (IMU), North Campus, Istanbul, Turkey, as seen in Figure 4.7. A snapshot from the field experiments is shown in Figure 4.10.

4.2.1 Stationary Target

The distance between the UAV and the target was approximately 265 meters in this stationary target experiment. The experiment consists of 7 measurement locations. The UAV trajectory, measurement locations, the target's estimated locations, and the target's true location are shown in Figure 4.11. The average speed of the UAV was about 11 m/s.

After the first measurement update, the localization error is large as expected in the first measurement update. Later, the estimated location of the target starts converging to the target's true location. However, the error increases after the UAV overshoots the target's location in the sixth measurements. But the localization error decreases after

Figure 4.10: Footage from the field experiments.

the seventh measurement update.

The particles' states in the stationary target experiment are illustrated in Figure 4.13. These particles' states are uniformly distributed at the initialization using Equation 2.1 and after the first, fifth and seventh measurement updates, respectively. The particles inside the first measurement area are resampled proportional to their weights. After each measurement update, the particles are propagated using Guided Walk. With the fifth and seventh measurement updates, the particles are clustering around the target location. Thus, the estimated target location converges to the target location.

Figure 4.12 shows the tracking error and the distance between the UAV and the target after each measurement update. The tracking was large at the first measurement update. Later, the estimated target location converges to the real target location. When the UAV overshoots the target at the sixth measurement, the tracking error gets larger. At the following measurement update, the tracking error decreases again.

Table 4.1 presents the results calculated from the localization errors obtained from the six stationary target experiments. Due to the constraints on the flight duration, we had the limited number of measurements. The mean of the localization errors is about 50 meters and the minimum localization error can be as low as 20 meters. While the

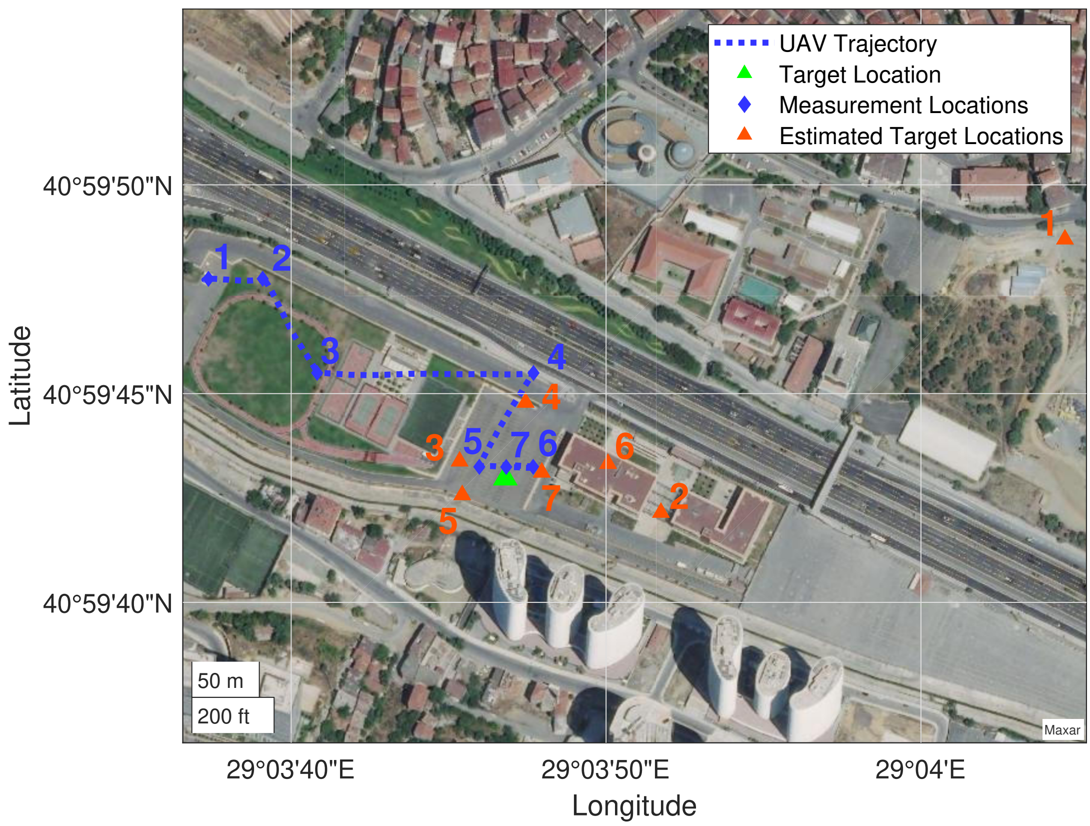

Figure 4.11: The UAV trajectory and the target's true location along with measurements and the target's estimated locations from a stationary target experiment.

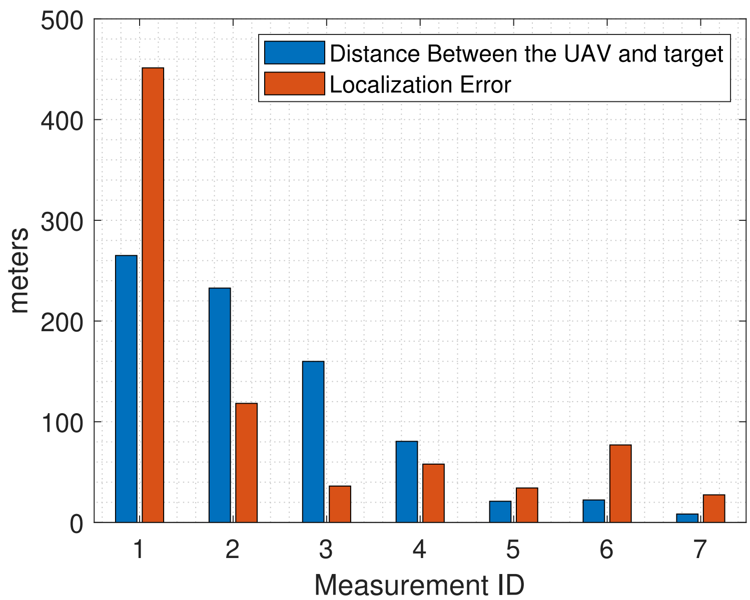

Figure 4.12: Localization error and the distance between the UAV and target results for the stationary target.

distance between the UAV and target mean is more than 70 meters in average, the minimum distance is around 8 meters.

Table 4.1: Stationary Target Experiment Results in meters (Average UAV speed: 11m/sec)

	Min	Max	Mean	Std
Localization Error	19.5	104.4	49.5	26.9
Dist. btw. UAV and Target	8.4	161.3	71.5	55.7

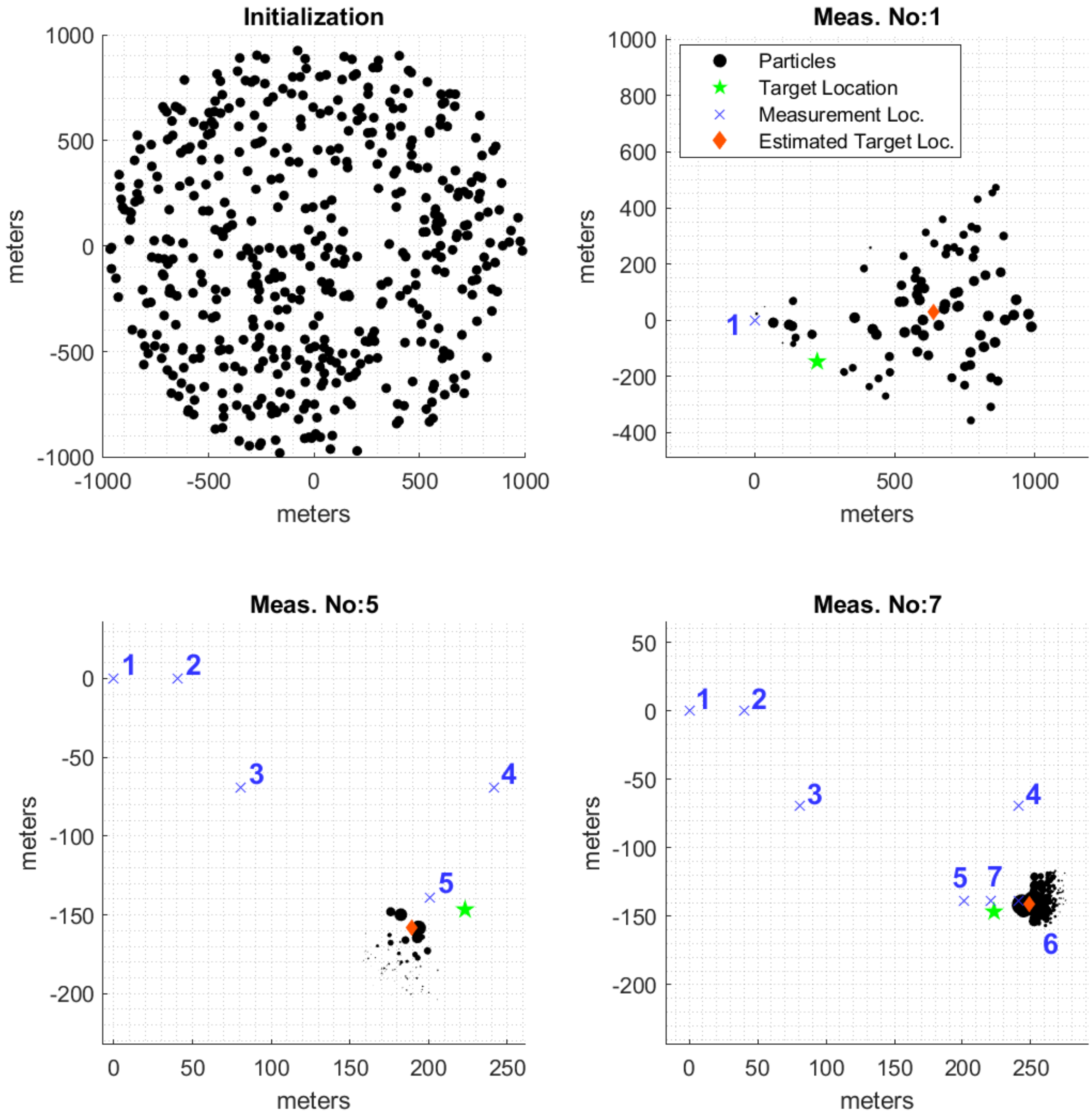

Figure 4.13: The evolution of the particles along with the target's true and estimated locations. The states of the particles were recorded in the stationary target experiment after each measurement update. The plots show the particles, the target's true and estimated locations, and the measurement locations after the first, fifth, and seventh measurements.

4.2.2 Moving Target

The moving target experiment consists of 7 measurements with each having 6 samples in total 42 samples as shown in Figure 4.14. The average speed of the UAV was about 11 m/s The target's velocities and accelerations vary along the experiment. The average values of target velocity and acceleration are 1.3 m/s and 0.17 m/s^2, respectively. This target was a person who walked in the green trajectory as seen in Figure 4.14. The target's trajectory is similar to the Velocity Model. The person carried all the target related hardware during the experiment. Meanwhile, the average speed of the UAV was about 11 m/s.

As in the stationary target, the estimated location of the target gradually converges to the target's true location and follows with the UAV while the target moves as shown in Figure 4.14 and 4.15.

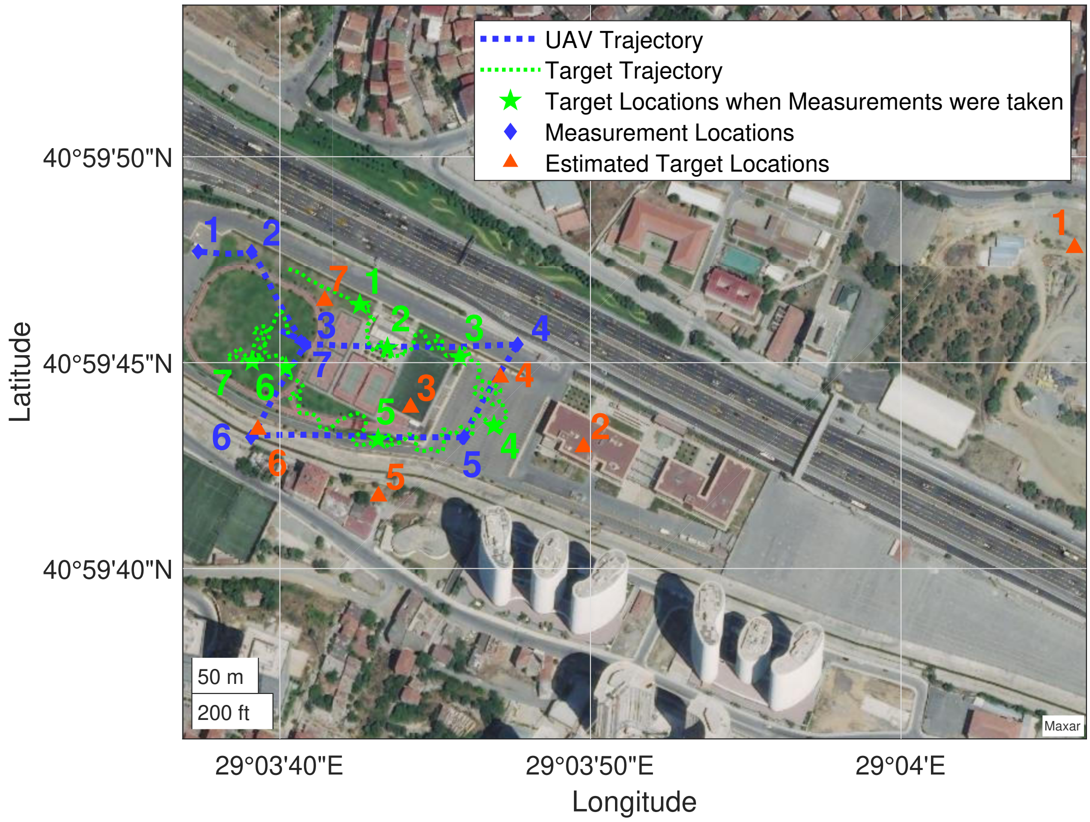

Figure 4.14: The trajectories of the UAV and the moving target along with measurement and target's estimated locations in one of the moving target experiments.

The particles are again distributed uniformly in the area using Equation 2.1. After the first measurement, the particles are resampled according to their weights. After the measurement update, the particles make the Guided Walk. This motion continues in the amount of the passed time that sum of the consumed time during travel from the one measurement location to another, one measurement time, and the computation of the measurement samples. The target's estimated location converges to the moving target location with these steps as seen in the fifth and seventh measurement updates in Figure 4.14.

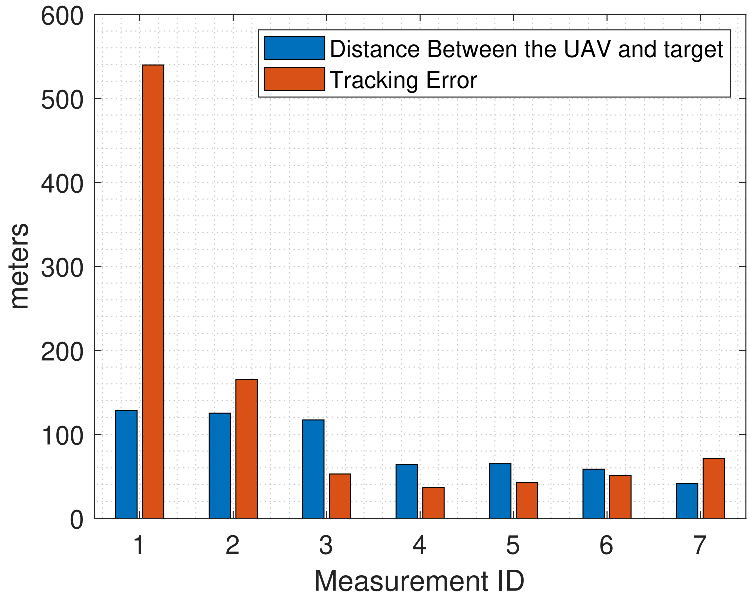

Figure 4.15: Tracking error and the distance between the UAV and the moving target

As shown in Table 4.2, for moving targets indicates the mean as 56 meters and standard deviation as 10 meters of the tracking error. These results are close to the simulation results of the Velocity Model with $\pi/6$ measurement noise and 500 number of particles in Table 3.5. Although the distance between the UAV and the target is short, the tracking error is large at the first measurement update. Later, the tracking error gets smaller in the next measurement updates and closer to the distance between the UAV and the target as seen in Figure 4.15. This trend can also be observed after the fifth measurement update in Figure 4.16. Again, because of the limitations on the flight duration, It is expected that the localization error would be after more measurements as observed from the simulations and system performance from the experiment.

Table 4.2: Moving Target Experiment Results in meters (Average UAV speed: 11m/sec)

	Min	Max	Mean	Std
Tracking Error	36.7	71.1	56.1	10.7
Dist. btw. UAV and Target	41.5	101.3	66.3	13.9

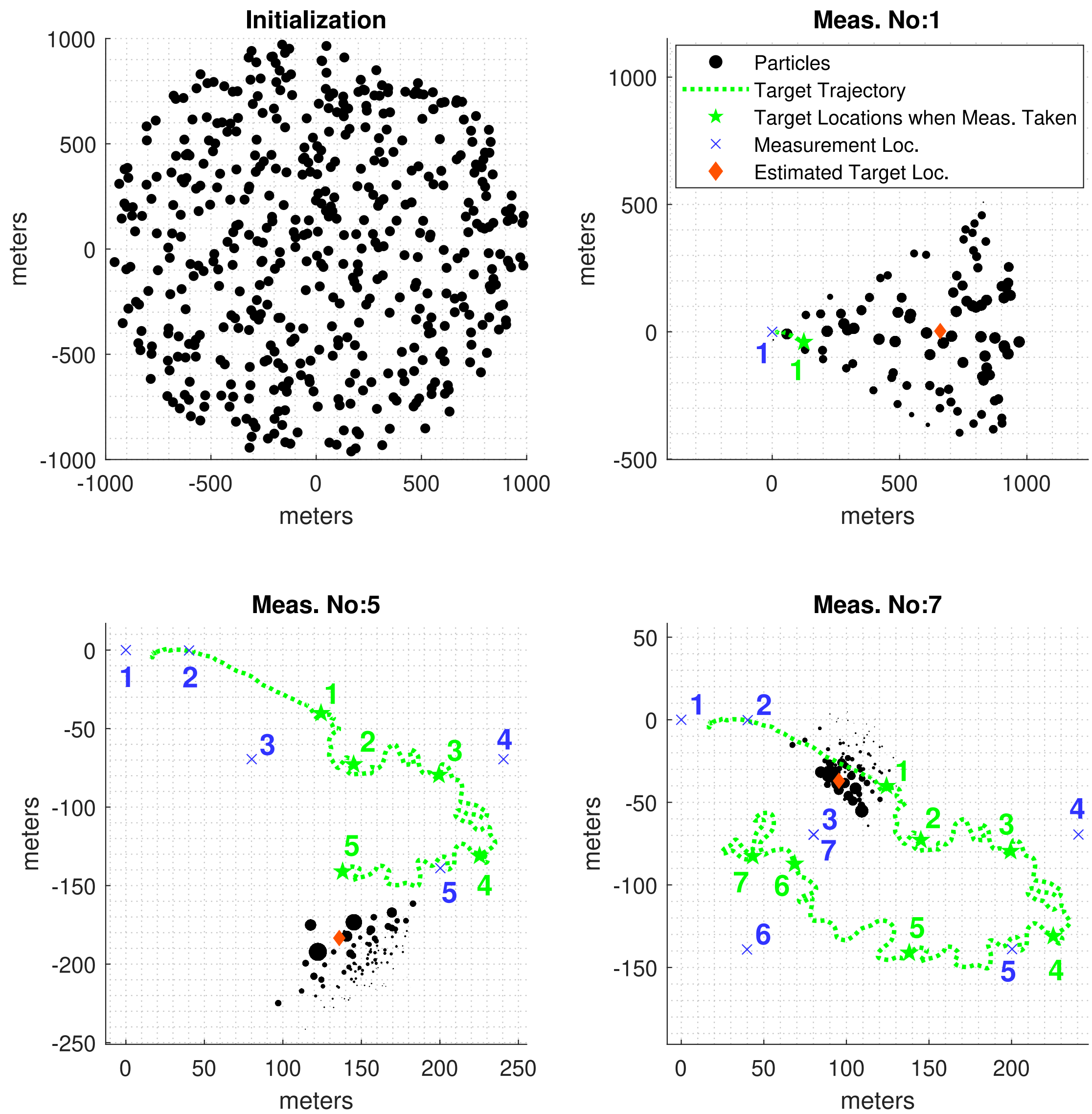

Figure 4.16: The evolution of the particles along with the true and target's estimated locations. The states of the particles were recorded in the moving target experiment after each measurement update. The plots show the particles, the target's trajectory, the target's location when the measurement is taken, the measurement locations, and the target's estimated locations after the first, fifth, and seventh measurements.

CHAPTER 5: CONCLUSION

In this thesis, we presented a particle filter based online strategy to localize and track a moving radio source using a UAV. In general, the UAV takes discrete π/α number of samples around itself with an SDR device and a Yagi-Uda antenna. α is the measurement noise which is the direction sensitivity of the on-board receiver antenna. These discrete samples constitute a bearing measurement, and all samples are analyzed to determine the bearing angle β to the signal source. This bearing information is then used in the measurement update of the particle filter, and in calculating the next measurement location. These particles are propagated using the Guided Walk. All the computations are done on an on-board computer. When the UAV arrives at the next measurement location, this algorithm continues, thus, localization and tracking of the target is performed. To analyze and verify the algorithm, a simulator is written. The target uses the Random Walk, Heterogeneous Behavior, Velocity Model, and Lévy Walk as a motion model. After verifying the algorithm by making extensive simulations, a series of field experiments were performed using the custom-made UAV.

We plan to replace the discrete sampling for measurement in future work with continuous sampling. Hence this reduces the measurement and signal analyzing time.

CHAPTER 6: APPENDIX

6.1 User's Guide

This section explains the initialization procedures of the system. The system works with its dedicated ground station application. This application is written in C# language and runs on the Windows OS. The ground station application also needs a USB telemetry device. When the emergency button is released on the UAV, the hardware and software of the system start automatically.

The dedicated ground station application needs a file named as"config.txt" in the directory `"C:\users\<username>\Desktop\Logs\config.txt"`. This file should include target frequency written as kHz as in the example below,

```
config.txt
433716
```

The application uses this value in every session. If the application cannot find the mentioned file, it uses $433MHz$ as a default target frequency. The application also uses this folder as an output directory for the flight logs. If the folder does not exist, the application creates one with the "config.txt" file in which the default target frequency is written.

First of all, the telemetry communication must be established between the ground station and the UAV. For this, the "Communication" tab can be used, or when pressed the "START", "Send Coord. List to Robot" buttons without any connection, the application will lead the user to the "Communication" tab as seen in Figure 6.1. The user can select a telemetry device and an appropriate baud rate to connect.

In order to start the tracking task, after the communication is established, the "START"

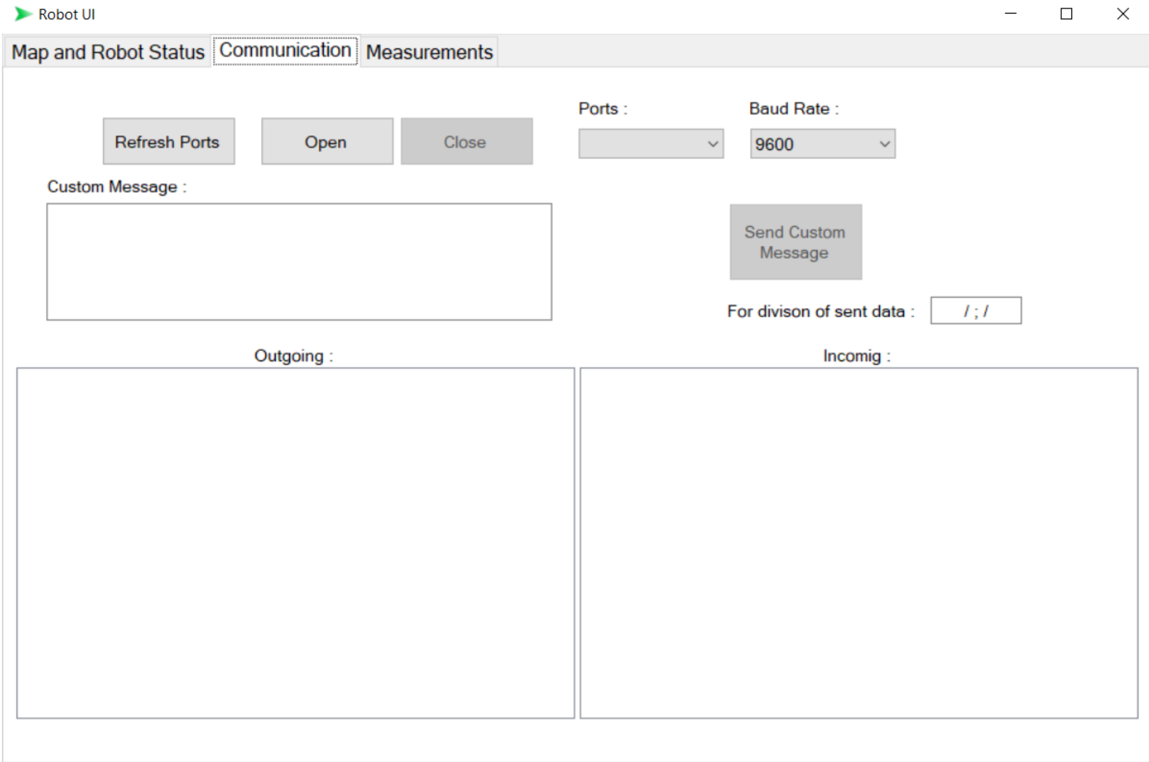

Figure 6.1: Communication tab to select a USB telemetry device and baud rate. The user also can send custom messages to the UAV and monitor incoming/outgoing messages on this tab.

button is used. When it is pressed, its appearance changes, and "STOP" is written on it. The UAV takes off and starts the localization and tracking node. After the "STOP" button is pressed, "STOPPED" appears on the button, and the process exits. This also stops autonomous flight and starts a predefined action stating that the UAV goes back to the landing location.

If the tracking task is halted, the user must restart the ground station application and the UAV by pushing and releasing the emergency button or connecting to the main computer via the `ssh` connection. Jetson Nano which is the main computer does not have a hotspot connection. Hence the user should set up a Wi-Fi network or a hotspot in order to connect to the Jetson Nano. After setting up the connection, the user opens the `ssh` and starts the system again by executing commands below respectively:

```
ssh jetson@<ip address of the jetson nano>
```

or

```
ssh jetson@jetson-nano.local
```

```
roslaunch rf_tracking_pkg start.launch
```

The target has the transmitter module and its battery. Additionally, a Raspberry pi, a Garmin GPS, and a power bank are used to locate the target and log the GPS data.

The transmitter module has a standard power jack that supplies power to the board via its cable with an XT60 male connector. Before a Li-Po battery is plugged in via this connector, the DC-DC converter on the transmitter module is adjusted to supply 9V output. If the user is not sure about the voltage of the DC-DC converter's output, its output cable can be detached by a screwdriver and adjusted using a multi-meter and screwdriver. After all the connections are done, a switch on the DC-DC converter alternates between input an output voltage values, and the transmitter module starts to emit the signal.

Garmin GPS should be connected to the Raspberry pi via a USB port to log the trajectory of the target. A power bank can be used to supply power to the Raspberry pi via its micro USB socket. After the connection is done and the Raspberry pi is initialized, the Raspberry pi establishes a hotspot connection. The user should connect to this Wi-Fi (ssid: ismu_pi, password: ismu4321) and open the `ssh` connection to start the ROS package that logs to the trajectory of the target. To establish `ssh` connection and start the ROS package, commands below should be executed on the terminal respectively:

```
ssh ubuntu@10.42.0.1
```

```
roslaunch gps_logger_pkg gps_logger.launch
```

REFERENCES

[Baeza-Yates et al., 1993] Baeza-Yates, R. A., Culberson, J. C., and Rawlins, G. J. (1993). Searching in the plane. Information and Computation, 106(2):234–252.

[Bayram et al., 2016a] Bayram, H., Doddapaneni, K., Stefas, N., and Isler, V. (2016a). Active localization of vhf collared animals with aerial robots. In IEEE International Conference on Automation Science and Engineering (CASE), pages 934–939.

[Bayram et al., 2017] Bayram, H., Stefas, N., Engin, K. S., and Isler, V. (2017). Tracking wildlife with multiple UAVs: System design, safety and field experiments. In International symposium on multi-robot and multi-agent systems (MRS), pages 97–103.

[Bayram et al., 2018] Bayram, H., Stefas, N., and Isler, V. (2018). Aerial radio-based telemetry for tracking wildlife. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4723–4728.

[Bayram et al., 2016b] Bayram, H., Vander Hook, J., and Isler, V. (2016b). Gathering bearing data for target localization. IEEE Robotics and Automation Letters, 1(1):369–374.

[Brockway and Gessaman, 1977] Brockway, J. and Gessaman, J. (1977). The energy cost of locomotion on the level and on gradients for the red deer (cervus elaphus). Quarterly Journal of Experimental Physiology and Cognate Medical Sciences: Translation and Integration, 62(4):333–339.

[Cliff et al., 2015] Cliff, O. M., Fitch, R., Sukkarieh, S., Saunders, D. L., and Heinsohn, R. (2015). Online localization of radio-tagged wildlife with an autonomous aerial robot system. In Robotics: Science and Systems.

[Elfring et al., 2021] Elfring, J., Torta, E., and van de Molengraft, R. (2021). Particle filters: A hands-on tutorial. Sensors, 21(2):438.

[Ermakov,] Ermakov, V. Mavros - mavlink extendable communication node for ros. `https://github.com/mavlink/mavros`. (Last accessed: 2021-10-28).

[Hasanzade et al., 2018] Hasanzade, M., Herekoğlu, Ö., Yeniçeri, R., Koyuncu, E., and İnalhan, G. (2018). RF source localization using unmanned aerial vehicle with particle filter. In 9th International Conference on Mechanical and Aerospace Engineering (ICMAE), pages 284–289.

[Hooten et al., 2017] Hooten, M. B., Johnson, D. S., McClintock, B. T., and Morales, J. M. (2017). Animal movement: statistical models for telemetry data. CRC press.

[Hui et al., 2021] Hui, N. T., Lo, E. K., Moss, J. B., Gerber, G. P., Welch, M. E., Kastner, R., and Schurgers, C. (2021). A more precise way to localize animals using drones. Journal of Field Robotics, 38(6):917–928.

[Isaacs et al., 2014] Isaacs, J. T., Quitin, F., Carrillo, L. R. G., Madhow, U., and Hespanha, J. P. (2014). Quadrotor control for RF source localization and tracking. In International Conference on Unmanned Aircraft Systems (ICUAS), pages 244–252.

[James et al., 2011] James, A., Plank, M. J., and Edwards, A. M. (2011). Assessing Lévy walks as models of animal foraging. Journal of the Royal Society Interface, 8(62):1233–1247.

[Körner et al., 2010] Körner, F., Speck, R., Göktogan, A. H., and Sukkarieh, S. (2010). Autonomous airborne wildlife tracking using radio signal strength. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 107–112.

[LaValle, 2006] LaValle, S. M. (2006). Planning algorithms. Cambridge university press.

[Nguyen et al., 2019] Nguyen, H. V., Chesser, M., Koh, L. P., Rezatofighi, S. H., and Ranasinghe, D. C. (2019). Trackerbots: Autonomous unmanned aerial vehicle for real-time localization and tracking of multiple radio-tagged animals. Journal of Field Robotics, 36(3):617–635.

[Open Robotics,] Open Robotics. The robot operating system (ros).

[Perkins et al., 2015] Perkins, A., Dressel, L., Lo, S., and Enge, P. (2015). Antenna characterization for UAV based GPS jammer localization. In Proceedings of the 28th

International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015), pages 1684–1695.

[Plonski et al., 2017] Plonski, P. A., Vander Hook, J., Peng, C., Noori, N., and Isler, V. (2017). Environment exploration in sensing automation for habitat monitoring. IEEE Transactions on Automation Science and Engineering, 14(1):25–38.

[Posch and Sukkarieh, 2009] Posch, A. and Sukkarieh, S. (2009). UAV based search for a radio tagged animal using particle filters. In Australasian conference on Robotics and Automation (ACRA).

[Sanderson and Curtin, 2016] Sanderson, C. and Curtin, R. (2016). Armadillo: a template-based c++ library for linear algebra. Journal of Open Source Software, 1(2):26.

[Soriano et al., 2009] Soriano, P., Caballero, F., Ollero, A., and de Tecnologıas Aeroespaciales, C. A. (2009). RF-based particle filter localization for wildlife tracking by using an UAV. In Proceedings of the 40th International Symposium on Robotics, Barcelona, Spain, pages 10–13.

[Teledyne SP Devices, 2018] Teledyne SP Devices (2018). The art of pulse detection.

[Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). Probabilistic Robotics. MIT Press.

[Vander Hook et al., 2014] Vander Hook, J., Tokekar, P., and Isler, V. (2014). Cautious greedy strategy for bearing-only active localization: Analysis and field experiments. Journal of Field Robotics, 31(2):296–318.

[Vander Hook et al., 2015] Vander Hook, J., Tokekar, P., and Isler, V. (2015). Algorithms for cooperative active localization of static targets with mobile bearing sensors under communication constraints. IEEE Transactions on Robotics, 31(4):864–876.

[VonEhr et al., 2016] VonEhr, K., Hilaski, S., Dunne, B. E., and Ward, J. (2016). Software defined radio for direction-finding in UAV wildlife tracking. In IEEE International Conference on Electro Information Technology (EIT), pages 0464–0469.

[Wagle and Frew, 2010] Wagle, N. and Frew, E. (2010). A particle filter approach to wifi target localization. In AIAA guidance, navigation, and control conference, page 7750.

Curriculum Vitae

PERSONAL INFORMATION

Name Surname: Muhammed Koray YILMAZ

EDUCATION

Degree	Institution	Graduation Year
BSc	SAÜ, Faculty of Engineering, Electrical and Electronics Engineering	2011
MSc	İMÜ, Institute of Graduate Studies, Electrical and Electronics Engineering	2021

WORK EXPERIENCE

Date	Institution	Position
11.2018 - 02.2019	Gedik Welding Inc.	R&D Engineer
06.2014 - 12.2016	TEKLAS Inc.	Maintenance Engineer

FOREIGN LANGUAGE

English (Advanced)

ACTIVITIES

Membership of Field Robotics Laboratory, 2019 -

Team member in Team IMU-AV, Teknofest Robotaxi - Full Scale Autonomous Vehicle Competition 2021, Finalist

PUBLICATIONS

E. Ozdemir, H. Gokoglu, M. K. Yilmaz, U. Dumandag, I. H. Savci, H. Bayram. "High Fidelity IMU and Wheel Encoder Models for ROS based AGV Simulations". Robot Operating System (ROS), vol. 7, Springer, 2022 (Accepted)

COURSES

Date	**Institution**	**Subject**
22.12.2018-19.01.2019	BT Akademi	OOP Principles & Design Patterns (32 Hours)
01.04.2017-07.10.2017	BT Akademi	Microsoft Certified Solution Developer (200 Hours)
26.11.2016-29.01.2017	BT Akademi	Programming C# with .NET (70 Hours)
04.03.2013-23.08.2013	Beet Language Centre	English Language
2011-2012	ARD Language	English Language