Development of an Adaptive Clustering Algorithm for Localization of Wireless Signals

by

Joseph B. Smith

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 02, 2026

Keywords: Localization, Clustering, Algorithm Selection

Approved by

Scott Martin, Chair, Assistant Professor of Mechanical Engineering
David Bevly, Bill and Lana McNair Distinguished Professor of Mechanical Engineering
Chad Rose, Assistant Professor of Mechanical Engineering

Abstract

The algorithm selection problem is an idea John Rice proposed in the 1970's in which he suggests that selecting the proper algorithm to be used to solve a problem is a problem itself [1]. Since then, there have been several ways proposed to address how to best select an algorithm for a given dataset [4], and more specifically for the use case of multiple target localization [3], which clustering algorithm will give the best results. One popular approach to solving the algorithm selection problem problem is with meta-learning, meta-learning operates off of heuristics in the data set and how it performs with a given algorithm to provide feedback for later iterations. The method described in this thesis foregoes the analysis of the data set going into the clustering algorithm and instead proposes an a priori method that uses the characteristics of the scenario that forms the data set to decide which algorithm is best suited resulting in less overall computation on a large dataset. This thesis covers a particle filter based multi target tracking algorithm developed around wireless signal localization. The Adaptive Clustering Engine (ACE) is introduced as a sub-algorithm to the particle filter that decides the clustering to be used. Algorithm selection is driven by a set of scenario characteristic estimates to determine quantity of targets, observability of targets, and proximity between targets. Simulation tests were run to develop a selection map that correlates algorithm performance to localization scenario characteristics. Additional simulation testing is performed to validate the system. Validation testing shows ACE is capable of selecting a clustering algorithm that correctly estimated number of targets 88.09% of validation runs.

Artificial Intelligence (AI) Use Disclosure Statement

In the preparation of this thesis, no Artificial Intelligence (AI) tools were used.

Digital Accessibility Use Disclosure Statement

In the preparation of this thesis, the following digital accessibility tools were used to ensure this document complies with federal requirements: Adobe Acrobat Pro. The author acknowledges full responsibility for the intellectual content of this work and has made a good faith effort to comply with digital accessibility requirements in publishing, wherein the nature of the content does not significantly change in order to do so. Furthermore, all content has been reviewed and revised to meet these requirements prior to final publication.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACE	Adaptive Clustering Engine
AOA	Angle of Arrival
ASP	Algorithm Selection Problem
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DOP	Dilution of Precision
MTT	Multiple Target Tracking
NFL	No Free Lunch
RF	Radio Frequency
RFS	Random Finite Set
SIR	Sequential Importance Resampling
TDOA	Time Difference of Arrival
TOA	Time of Arrival

Chapter 1

Introduction

1.1 Motivation

Wireless radio frequency (RF) localization has been an area of research for over a century dating back to the early 1900's using direction finding measurements to track ships [16]. Modern use cases are varied for applications across the electromagnetic spectrum. Even more varied are the algorithms developed to tackle the wireless localization problem as well as the more generalized multiple target tracking (MTT) problem. Algorithm selection for the MTT problem becomes more difficult as scenario factors change and alter over time leading to the need for more flexible methods. The process of selecting an algorithm to address a problem statement becomes a problem itself.

This thesis problem looks to address the higher level problem of which algorithm is best to use. The algorithm selection problem is an idea John Rice proposed in the 1970's in which Rice suggests that selecting the proper algorithm to be used to solve a problem is a problem itself [1]. Since then, several ways have been proposed to address how to best select an algorithm for a given data set [4], and more specifically for the case of multiple target localization, which clustering algorithm will give the best results [3]. One popular approach to solving the algorithm selection problem problem is with meta-learning, meta-learning operates off of heuristics in the data set and how it performs with a given algorithm to provide feedback for later iterations. The method described in this thesis foregoes the analysis of the data set going into the clustering algorithm and instead proposes an a priori method that uses the characteristics of the scenario that forms the data set to decide which algorithm is best suited, resulting in less overall computation on a large dataset.

1.2 Prior Art

Locating targets through the use of wireless signals has a long and storied history, covering applications as small as locating Wi-Fi devices to radar systems tracking aircraft miles away. From these different methods, there is a set of generalized measurement types that geometrically relate sensor and target locations in three-dimensional space. These measurements convert signal characteristics into geometrical relations such as angles and distances which can be combined to form a position solution. The wireless signals evaluated in this thesis are angle of arrival (AOA), time of arrival (TOA), and time difference of arrival (TDOA) [17]. These measurements have been thoroughly researched including the concepts of sensor geometry, measurement error values, and relation to localization performance [18], [20], [21].

Multiple Target Tracking (MTT) is a more generalized field for estimating number of targets and their locations. Many methods exist but one that has been popular in literature in recent years is Random Finite Set (RFS) approaches that avoids traditional measurement and track association [5]. A subset of the RFS approaches are particle approaches that use many potential target states that get filtered out instead of direct solve methods [10] [14]. Particle methods have been implemented for a wide variety of use cases because of their algorithmic flexibility covering use cases such as searching for emergency beacons [11], autonomous mobile robots [8], and tracking zebrafish [12].

One particle based method of MTT that has shown promise is the inclusion of machine learning clustering techniques to split the particle based probability density function into individual state estimates. The approach has been evaluated through multiple research efforts in which several independent clustering algorithms have been implemented [6], [8], [9].

The process of selecting an algorithm is itself a field of study. Originating with John Rice's Algorithm Selection Problem (ASP) which looks at linking different application characteristics to algorithm performance to help in making a well informed algorithm selection

[1]. Selecting a best algorithm is not always an option; there are often tradeoffs where algorithm A might be better in some areas but algorithm B performs better in others. The No Free Lunch (NFL) theorem suggests that the implementation of multiple approaches can lead to better overall performance [2]. Meta-learning is field that shares common features with ASP and provides implementations to select best performing algorithms using reference data referred to as meta-data [4]. As an example, meta-learning can be used to select a clustering algorithm based on particle distribution parameters [3]

1.3 Contributions

The contributions presented in this thesis cover two distinct areas related to wireless signal localizations. The first is mapping out what scenario factors lead to different algorithm performances. In addition to mapping out the factors, methodologies are explored for estimating the factors using deterministic values calculated during operation. The second area is the implementation of adaptable algorithms that can fully replace portions of the algorithm to adapt to real time changes in scenarios. The full contributions to this research field are as follows:

- Introduction of the Adaptive Clustering Engine (ACE)
- Mapping of scenario characteristics that impact clustering performance
- Methodology for characterizing scenarios without external information
- Adapted particle filter algorithm to allow for real time sub-algorithm selection

1.4 Thesis Outline

This thesis is organized by first providing the technical background for wireless signal localization and the algorithm selection problem, then introduces the algorithm that combines the concepts of the two mentioned concepts and ends with software simulation.

Chapter 2 introduces wireless signal localization and covers topics such as measurement types and localization algorithms. The chapter includes visual representations of different measurement types, describes how localization algorithms work, and introduces the concept of particle clustering, which is one of the main focuses of this thesis.

Chapter 3 introduces the algorithm selection problem, which provides an abstract model for selecting the best algorithm to fit a certain application. The algorithm selection problem model is described with details of how it is applied to the wireless signal localization problem. Clustering algorithms are explained, and example scenarios are provided to show how different algorithms perform better or worse in different scenarios.

Chapter 4 introduces the Adaptive Clustering Engine (ACE), an algorithm that runs within a wireless signal localization architecture and estimates problem set parameters that can lead to better clustering algorithm selections. ACE estimates problem set characteristics that are unknown at the time of operation using sensor location and measurement relations. The process of creating meta-data used by ACE for algorithm selection is shown. Along with meta-data results, results showing having multiple algorithms perform better than any one singular algorithm including the selection map that is used to pick the better performing algorithms.

Chapter 5 demonstrates software validation of the ACE algorithm with a variety of simulation parameters. Results are shown for ACE related to clustering performance and the problem set estimation portion of ACE.

Chapter 6 provides conclusions drawn from the simulation validation testing. Both successful and areas of improvement are explored with future work provided that could push the concepts further.

Chapter 2

Wireless Signal Localization

Wireless signal localization is the general term for locating the source of a wireless signal received or using signals to locate a target using one or more sensors. There are several different measurement types that can be formed through signal processing. This thesis focuses on three of the most common measurements: Angle of Arrival (AOA), Time of Arrival (TOA), and Time Difference of Arrival (TDOA). These three measurements are divided into two groups, based on the types of signals used to create the measurements, to create realistic pairings of what measurements could be available at the same time [17].

Multiple Target Tracking (MTT) is another field of research that is devoted to estimating and tracking multiple targets in the environment at the same time. One subset of MTT algorithms is the Random Finite Set (RFS) approach that simultaneously tracks optimal and non optimal target states [5]. Particle based methods such as [9], [12], and [10] are examples of RFS algorithms that use many potential target states to estimate the number of targets and target locations. This thesis uses the approach of a single set of particles that are separated using machine learning clustering algorithms.

This chapter will provide an overview for the types of measurements generated using wireless signal methods and apply them to an MTT particle filter implementation to provide a technical background for the main subject of this thesis.

2.1 Multiple Target Tracking

Multiple Target Tracking (MTT), also commonly referred to as Multiple Object Tracking, is a field of research pertaining to the problem of simultaneously estimating number of

targets and their states. While there are many methods of addressing the problem, random finite set (RFS) based algorithms have shown promising results in recent decades. The distinguishing feature of RFS algorithms is the departure from traditional tracking algorithms that require measurement association to individual tracks. This data association problem can be very complex and made even more difficult by missed detections or false measurements. Instead, RFS algorithms track both optimal and suboptimal estimates and can eliminate the need for data association [5]. In this thesis, an adaptation of a multi-target particle filter will be discussed as the core MTT algorithm being tested.

2.2 Cooperative vs Non-Cooperative Signals

This thesis focuses on wireless signal based measurements for both cooperative and non-cooperative signals. These measurements include Time of Arrival (TOA), Time Difference of Arrival (TDOA), and Angle of Arrival (AOA). Cooperative signals are classified as signals that provide information within the signal, have a known signal structure, or any other factors that allow sensors to interpret more information from the signal. Examples of cooperative signals are GNSS which measures the time the signal is sent into the signal, radar systems that emit a pulsed signal and time the time of flight for the signal to return, and two way radio ranging. Non-cooperative signals do not have any additional information that is easily extracted from the signal. For example, a broadband noise signal that is random in the time-domain has no reference to transmission time and would require correlation between two sensors to extract the time difference of arrival.

For the analysis in this thesis cooperative signals will use the TOA and AOA measurements and non-cooperative signals will use TDOA and AOA. This chapter will briefly cover each measurement types geometric relation between sensor or sensors and the target and the error models. The way measurements are formed will not be included, but some examples can be found in [17].

2.3 Measurement Types

The three measurement types included in this thesis are Time of Arrival (TOA), Time Difference of Arrival (TDOA), and Angle of Arrival (AOA). Each measurement has a unique geometric relation that directly affects the localization performance. The analysis of different measurement types is broken down into five categories. The first three are the individual measurement types that run separately. The next two are combinations of measurements based on the relation of cooperative and non-cooperative signal types, which is AOA and TOA for cooperative and AOA and TDOA for non-cooperative.

2.3.1 Angle of Arrival

Angle of arrival (AOA), sometimes referred to as direction of arrival (DOA) or line of bearing (LOB), measurements are a relative angle between the sensor and target. AOA signals can be estimated for both cooperative and non-cooperative signals and only require a single senor to produce the measurement.

The AOA measurements are composed of an azimuth value for the Two-Dimensional direction and an elevation value for the third dimensional component. A single AOA measurement defines two of the three dimensions for a spherical coordinate system, only missing the range component, meaning that they cannot fully estimate a position without additional measurements. Often times, AOA measurements are combined from multiple source locations through triangulation to estimate a target location.

AOA measurements are shown geometrically as a cone protruding from the sensor location, the width of the cone being proportional to that of the AOA uncertainty value. Because the cone grows as a function of distance, AOA measurements create less accurate location estimates the farther away a sensor is from a target, as shown in Figure 2.1.

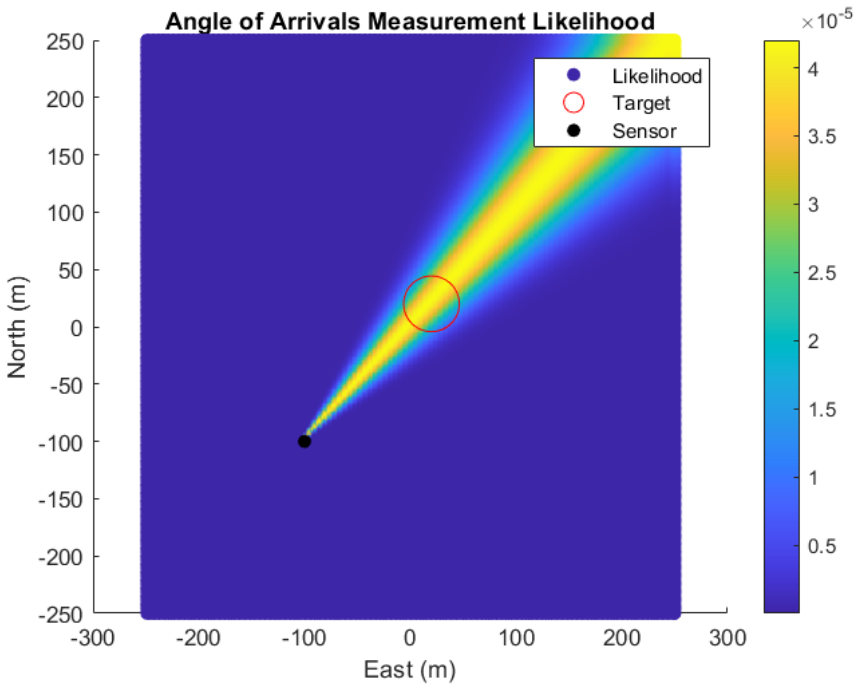

Figure 2.1: Angle of Arrival Measurement Likelihood

2.3.2 Time of Arrival

Range measurements report the 3-Dimensional Euclidean distance between the sensor and target. Range measurements are generally derived using the time it takes for a signal to propagate and the rate at which the signal travels, referred to as Time of Arrival (TOA). Radar, for example, emits a signal and measures the time it takes for the signal to reflect off the object and return. Dividing this time by two gets the one direction time and can be converted to a range by multiplying by the propagation rate of the signal used. Global Navigation Satellite System (GNSS) positioning is another range based system with the additional complexity of estimating the sensors time bias. As a simplified explanation, GNSS satellites emit their time and location and GNSS receivers use their own time to calculate a time of flight of the signal to find ranges to the satellite [21]. Geometrically, range measurements create a sphere around the sensor where all points are equal distant from the sensor or a circle when simplified to two dimensions. Measurement noise changes

the circle around the sensor to a ring around the sensor of higher likelihood as shown in Figure 2.2.

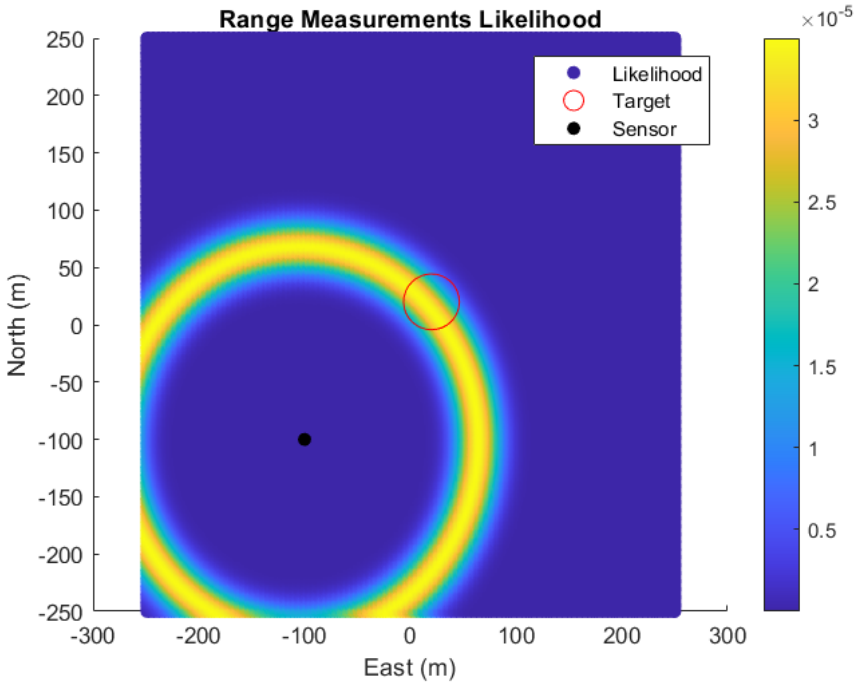

Figure 2.2: Time of Arrival Measurement Likelihood

2.3.3 Time Difference of Arrival

If the targets use non-cooperative signal types, it may not be possible to derive time of arrival measurements because there is no known time signature in the signal. Instead, the same signal received by a geometrically diverse pair of sensors can be used to calculate time difference of arrival (TDOA) of the signal by correlating two time synced snippets of the received signal to estimate the difference in received time. TDOA measurements are equivalent to range difference through the signal propagation rate and creates a hyperbola in two dimensions and a hyperboloid in three dimensions where every point on either shape is equal distant between each sensor. A visual representation of a two-dimensional hyperbola with the likelihood function is shown in Figure 2.3.

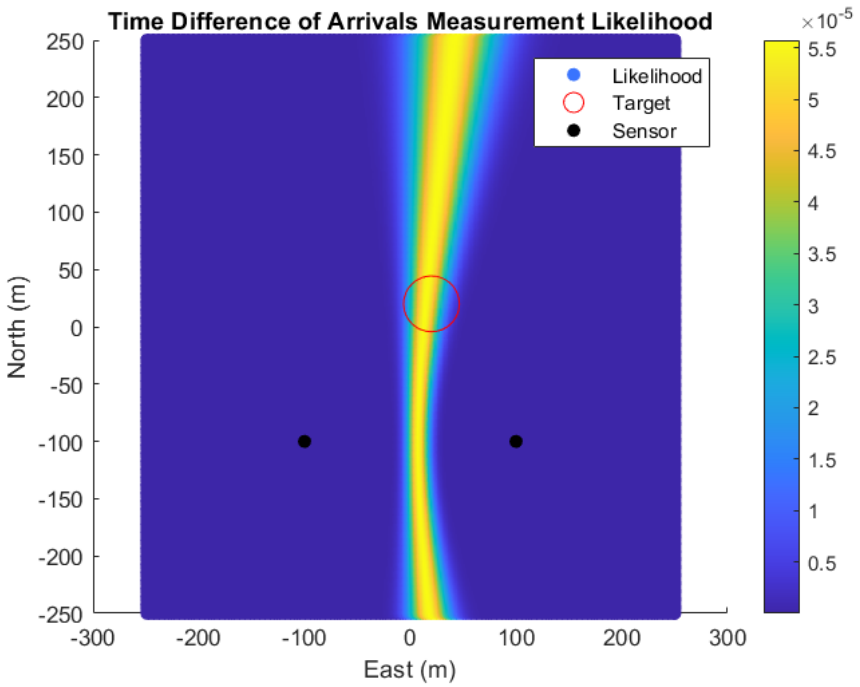

Figure 2.3: Time Difference of Arrival Measurement Likelihood

The error model for TDOA measurements, like TOA, is based on timing uncertainty. However, for TDOA you have the combination of two timing uncertainties because it takes two sensor measurements to create one measurement, increasing the measurement uncertainty relative to TOA measurements even if the sensors have the same timing quality.

2.4 Particle Filters

A particle filter (PF) is a form of a sequential Monte Carlo estimator capable of estimating linear and non-linear states. The particle filter operates using a large set of randomized state estimates, referred to as particles, to produce the probability density function for the environment. Each potential state, or particle, has an associated weight value that is linked to the likelihood that that particle represents the true state being estimated. A higher weight particle represents a state solution that is more likely to represent the truth given the

available measurements. The particle filter in this thesis is a sequential importance resampling (SIR) filter with adaptations to track multiple states simultaneously [7]. The general operation of a traditional SIR particle filter is outlined in Figure 2.4.

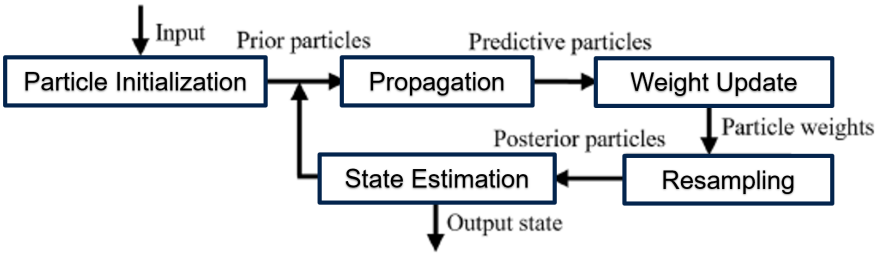

Figure 2.4: Particle Filter Operation

2.4.1 Particle Initialization

The particles are initialized in a uniform grid pattern to ensure coverage of the observable space. The initial grid size is calculated using the sensor locations and relative distances from each other, as well as the perceived dilution of precision (DOP). More information about the perceived DOP is available in later sections. The particles are initialized with a normalized weight of $1/N$ where N is the total number of particles [6].

2.4.2 Propagation

Particles are propagated forward in the state transfer using a basic motion model to allow tracking of dynamic targets. Without the state transfer, dynamic targets would move out of the area observed by the particles and the filter would lose track. Additive noise is used to spread the particle distribution out to cover more of the observable space and to prevent particles from converging to a single location over time as seen in Equation 2.2.

$$x_k^i \sim \mathcal{N}(x_{k-1}^i + B_k u_k, Q) \tag{2.1}$$

Where x_k^i is a sample from a Gaussian distribution with mean vector $x_{k-1}^i + B_k u_k$ and covariance Q [7].

2.4.3 Weight Calculation

Particle weights are updated using measurements by first computing what measurement the particle states would produce given the same sensor location(s) then calculating the probability of that particle state producing the sensor measurement as seen in Equation 2.2.

$$\omega_k^i = \omega_{k-1}^i p(z_k|x_k^i), i = 1, 2, ..., N \tag{2.2}$$

Where $p(z_k|x_k^i)$ is the conditional probability of measurement z_k given the particle state x_k^i. The particle filter weight calculation can be adapted to use non-Gaussian probability distributions unlike other common filters. This process is repeated for each particle before the particle weights are normalized to have a sum of 1 using Equation 2.3.

$$\omega_k^i = \frac{1}{\sum_{j=1}^{N} \omega_k^j}, i = 1, 2, ..., N \tag{2.3}$$

2.4.4 Resampling

SIR resamples lower weight particles into higher weight particles to better fill out the observable space in areas of higher likelihood. Resampling is performed once the number of effective particles N_{eff} falls below a threshold N_{th}. N_{eff} decreases from a maximum value of N as the particles are weighted unequally and peaks start to form in the weight of the particles. N_{eff} is calculated as shown in Equation 2.4.

$$N_{eff} = \frac{1}{\sum_{j=1}^{N} (\omega_k^j)^2} \tag{2.4}$$

N_{th} is a tunable parameter for particle filters with higher values causing more frequent resamples and lower values taking more measurements to resample. The particle filter used in this thesis uses a N_{th} value of 0.6 which allows more sensor movement between resampling updates to allow overall tighter particle clusters. For more dynamic targets, a higher N_{th} should be considered to prevent particles from diverging or losing the target because of high dynamics. Once resampled, the particle weights are reset to $1/N$ to allow the distribution to develop again. A visual example of the SIR process is shown in Figure 2.5.

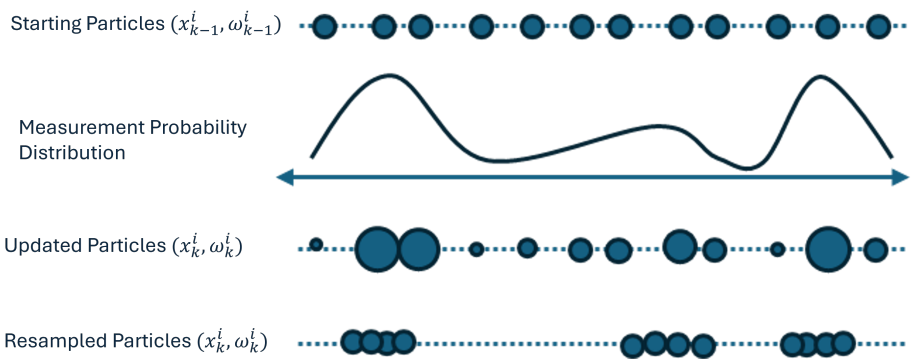

Figure 2.5: Particle Filter Operation

2.4.5 Estimation

Once the posterior distribution of the filter is set, the states can be estimated using a weighted mean of all of the particles, and covariance can be calculated using a population covariance of the particle distribution since the density of the particles reflect the distribution through the resampling step. Target states are estimated using the weighted mean Equation 2.5.

$$\hat{x}_k = \sum_{j=1}^{N} \omega_k^j x_k^j \tag{2.5}$$

Target state covariance is calculated using the sample covariance Equation 2.6.

$$P = \frac{\sum_{j=1}^{N} (x_{k,x}^j - \hat{x}_{k,x})(x_{k,y}^j - \hat{x}_{k,y})}{N} \tag{2.6}$$

Where $\hat{x}_k$ is the state estimate at time k with covariance P [6].

2.5 Chapter Conclusion

In conclusion, key components of the technical background are covered to provide context for later chapters in this thesis. The topic of multiple target tracking is introduced as the field for simultaneous estimation of the number of targets and their locations/trajectories. The wireless signal localization problem is introduced as an example application for MTT algorithms. Common measurements used for wireless signal localization problems are introduced including the measurement's geometric relation between sensor and target. Signal characteristics are used to split the potential measurement types into sub-categories that will be used in following chapters. Finally, a technical background for particle filtering is provided that will be expanded upon in Chapter 3.

Chapter 3

Particle Filter Adaptations for Multiple Target Tracking

Adaptations are required for a particle filter to operate in an environment with multiple targets. The first is with the weight calculation. The generic particle filter weight update multiplies the existing weight with a likelihood value that is less than one, decreasing all weights, and requiring the normalization step to bring weights back up. With multiple targets, measurements from one target would negatively impact particles that are associated with other targets. The second adaptation needed is to separate the particles into subgroups or clusters to calculate accurate target states. The separation of particles is achieved through a machine learning process referred to as clustering.

3.1 Additive Weight Calculation

In the generic particle filter case updating particle weights acts as a destructive process, and particles that are not associated well with the measurements have their weights decreased so that they are removed. In the case of multiple targets, an adaptation needs to be made so that particle weight does not decrease when the particle does not match, but rather increases when it fits. In order to change the relation, a simple value of one is added to the probability therefore increasing weight by a small factor for well fitting particles, and particles that do not match the measurement have their weights remain the same. In this case, the normalization step brings the weights down to a sum of one rather than increasing them from lower values as seen in Equation 3.1.

$$\omega_k^i = \omega_{k-1}^i(1 + p(z_k|x_k^i)), i = 1, 2, ..., N \tag{3.1}$$

Visual comparisons of how this change affects the particle distribution are shown below for basic cases with the different measurement types that are being used in Figure 3.1, 3.2, and 3.3 for AOA, TDOA, and TOA measurements.

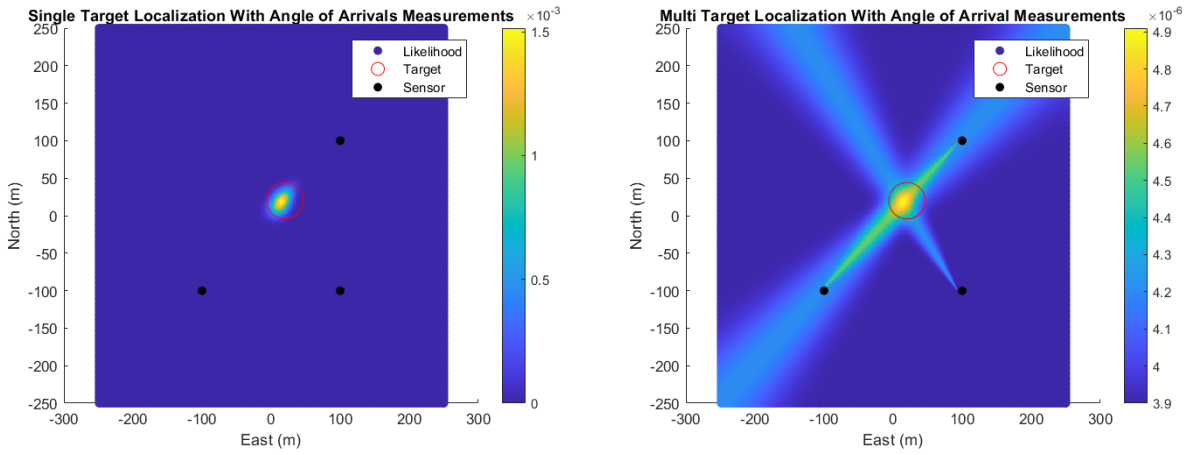

Figure 3.1: Comparison of Particle Weighting for AOA Measurements, Single Target PF (left) and MTT PF (right)

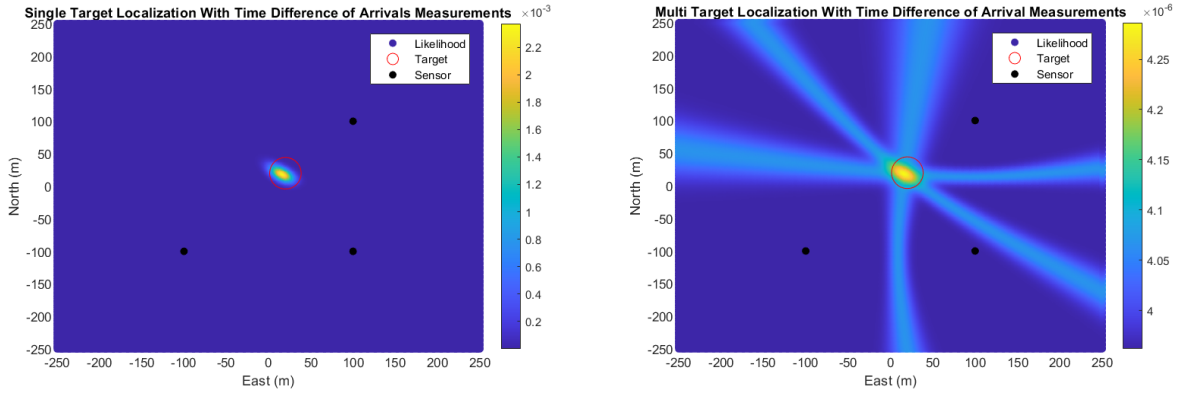

Figure 3.2: Comparison of Particle Weighting for TDOA Measurements, Single Target PF (left) and MTT PF (right)

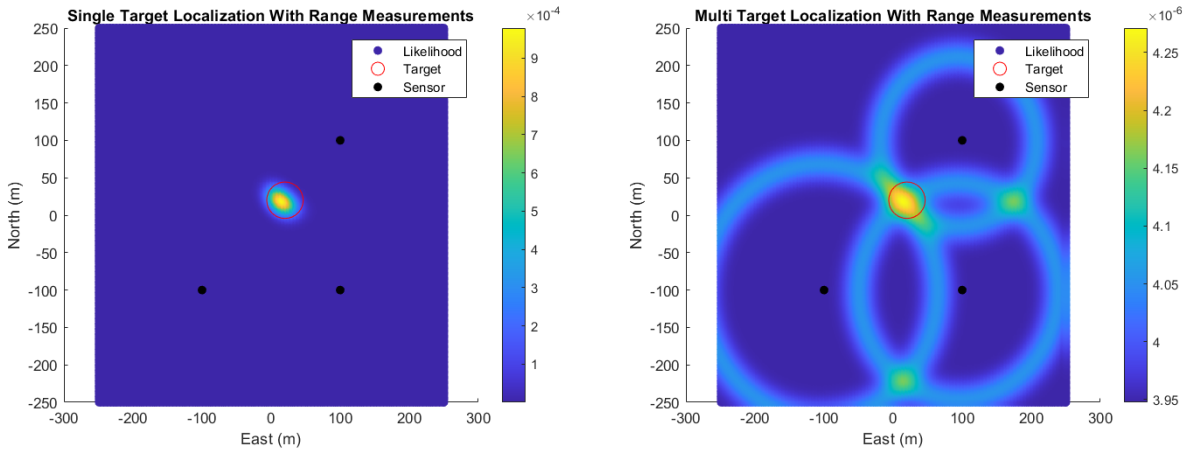

Figure 3.3: Comparison of Particle Weighting for TOA Measurements, Single Target PF (left) and MTT PF (right)

Through the additive nature of the new weight calculation in Equation 3.1, the whole area of measurements and the measurement uncertainty can be seen in the weight plots. The additive weight allows multiple particle peaks to form in the distribution and allows estimation of multiple targets.

3.2 Clustering

In order to split the larger particle set into individual groups to represent each individual target, the particles must go through a clustering step. Clustering of particles assigns individual cluster IDs to allow other processes in the particle filter to isolate groups of particles and estimate states for the corresponding target without being affected by unrelated particles. The process of clustering data points, as performed in this application, is an example of unsupervised machine learning. Unsupervised learning is a machine learning technique that works on unlabeled data, allowing grouping without human intervention. More details on clustering and the algorithms selected for evaluation are available in Section 3.5.

With a large list of potential clustering algorithms available in the literature, the process of selecting the best algorithm becomes an algorithm itself. Different factors are needed to

be taken into consideration that can affect clustering performance, and the methods in which scoring algorithm performance needs to be formulated to enable selecting the best clustering algorithm. The remaining sections of this chapter cover the algorithm selection problem and outline methodologies applied to make an informed algorithm selection.

3.3 Algorithm Selection Problem and Meta-Learning

When developing or implementing an algorithm to address a problem, there are often multiple options or algorithms to choose from. Implementers can evaluate which algorithm is best suited for the problem based on experimental data or insight from working in similar applications. The way in which an algorithm is selected can be arbitrary or heavily investigated. This process of selecting an algorithm can itself be treated as an algorithm. This is the basis for the Algorithm Selection Problem (ASP), an abstract framework designed to address the higher level question of which algorithm would perform best for a specific problem introduced by John Rice in the 1970's [1].

The algorithm selection problem is an abstract model consisting of three components, the problem space, the algorithm space, and the performance space. Between these three components are two sets of mappings. Between the problem space and the algorithm space is the selection mapping $S(p)$, responsible for taking in the problem set p from the problem space and selecting the corresponding algorithm a. The second mapping is the performance mapping that takes in a problem set p and a selected algorithm a to map which performance measures should be evaluated $y(a, p)$.

The ASP is modeled in three components: the problem space, the algorithm space, and the performance space. Together, these components can be used to outline a problem, collect all applicable algorithms, and derive a way to rank algorithm performance to solve the overarching problem of which algorithm is the best suited for the original problem [1].

The machine learning community took notice of the algorithm selection problem and derived their own version referred to as meta-learning, or "learning about learning". Common principles shared between ASP and meta-learning include:

1) Large collections of problem instances

2) Large number of diverse algorithms

3) Performance metrics to evaluate algorithm performance

4) Features to characterize properties of problem instances

From the No Free Lunch Theorem of [2], the concept of implementing a single algorithm to address a broad problem domain is not reasonable. Different algorithms have tradeoffs that cause improved performance for parts of the problem space but worse performance in other parts. Instead, multiple algorithms should be available for selection. Meta-learning uses machine learning principles to combine the shared principles above into a comprehensive set of training data, referred to as meta-data, that can determine algorithm selection based on the problem set similar to that of the selection map in the algorithm selection problem [4].

Combining principles of the two fields, this thesis develops a two part system that creates a comprehensive set of meta-data and a deployment method that estimates the active problem set for accurate algorithm selection. The meta-data creation portion matches that outlined in [4] and is visualized in Figure 3.4.

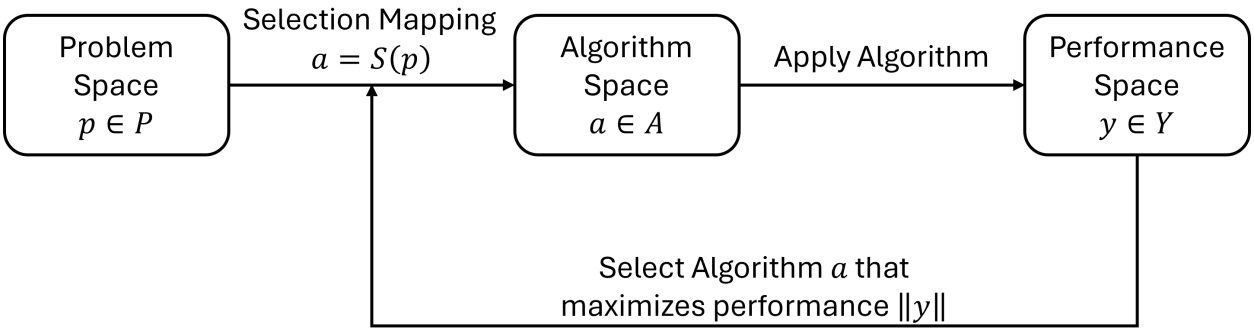

Figure 3.4: Meta-Data Creation Framework

A similar approach of using meta-learning principles has been taken in [3] to dictate clustering algorithms; however, the approach outlined uses the characteristics of the output data to determine the problem space. The proposed implementation in this thesis takes an a priori approach and determines the problem state using characteristics that shape the particle distribution instead of the particles themselves. A visual representation of this implementation is shown in Figure 3.5.

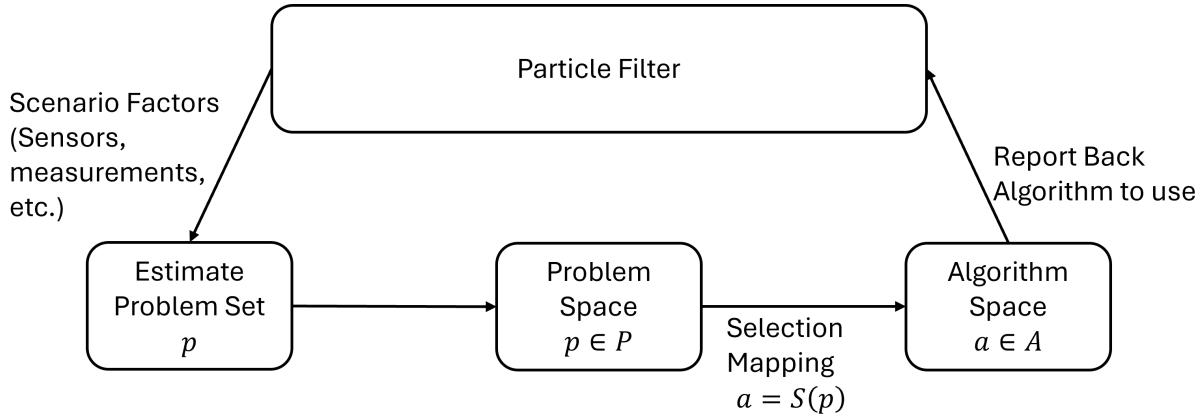

Figure 3.5: Implemented Algorithm Selection Process

3.4 Problem Space

The problem space is the set of independent characteristics that affect algorithm selection and performance. The determination of the characteristics that define the problem space is one of the most important and tedious aspects of the algorithm selection problem. An appropriate sample of the problem space is needed to make reasonable selections.

The scenario factors that impact localization performance can be split into two categories. The first is sensor dependent factors and include what types of measurement are available for the sensors being used and their relative noise values. The other set of factors are target dependent and include how many targets are present, and if there are multiple how close are they to each other, and the dilution of precision of the scenario which relates

to the relative location of targets compared to sensors. These two subsets are used to create a matrix of performance results that will be evaluated for each algorithm.

3.4.1 Target Quantity

The target quantity subspace can be further divided into three categories: single target, multiple targets, and multiple targets within close proximity.

The single target case is generally easier for localization because all the measurements are associated to a single target. The challenge with clustering for a single target is the possibility for clustering algorithms to subdivide a single cluster incorrectly. Because the number of targets is considered unknown, there is no way to ensure that there is only one target present; therefore, the filter still needs to go through the clustering process.

Increasing the number of targets shows the need for clustering in a particle filter. Since different particles become estimates for different targets, the particles should be clustered and labeled before the estimation step. A range of 2-6 targets was used to generate the meta-data in simulation. The multiple target section of the problem space is designed to have geometrically diverse target locations to allow for more isolated particle distributions and should result in better clustering performance.

In real world applications, it is uncommon for targets to be optimally spaced in a way that benefits algorithm performance. Instead, it requires algorithm flexibility to handle the more difficult cases that could arise. In the instance of MTT one of the most difficult use cases is multiple targets being in close proximity to each other relative to the sensors. The close proximity can have several knock on affects, such as only producing one measurement in the direction of the two targets, or a blended measurement that measures between the two measurements. Additional effects can include particle distributions that blend together, and it becomes more challenging for clustering algorithms to correctly split the two particle distributions. The clustered case is designed to have more frequent measurement overlaps

by putting targets within a three standard deviations for the measurements from another target resulting in more often particle distribution overlaps.

3.4.2 Measurement Noise Level

Increased levels of sensor measurement noise affect the particle filter by increasing the overall size of particle distributions around each target. The size of the distribution could impact the clustering algorithms in a few different ways. The measurement noise used in simulations are all zero mean Gaussian distributions with different variances to create three different noise levels. The noise levels used are shown in 3.1. One important note is that TDOA measurements use the same values as the TOA measurements because they are both time based errors, but TDOA adds two random values because it takes two sensors to produce one measurement.

Measurement Noise Levels in Terms of Standard Deviation				
Measurement Type	Units	Low Noise	Medium Noise	High Noise
Angle of Arrival	degrees (°)	2	4	6
Time of Arrival	meters (m)	50	100	150
Time Difference of Arrival*	meters (m)	50	100	150

Table 3.1: Noise Levels for Different Measurement Types

3.4.3 Dilution Of Precision

The Dilution of precision (DOP) is a value that incorporates relative geometries and measurement error to produce a theoretical estimation error. Different measurement types and the corresponding geometries influence the DOP value differently depending on the measurements. The DOP of a localization problem is calculated using the measurement matrix of all the measurements used in the localization H. To compute the DOP for a given measurement matrix, Equation 3.2 is used.

$$DOP = \sqrt{(H^T H)^{-1}} \tag{3.2}$$

The main diagonal terms of the resulting matrix can be used to estimate the error in the corresponding direction. Example measurement matrices for the different measurement types are included in the following sub sections.

A simplified DOP level is implemented to simplify the wide range of possible DOP values into three bins that can be used to outline the problem subspace. The different bins or levels are classified on the basis of the ratio of DOP values for each direction, or the ratio of the semi-major and semi-minor axis of the corresponding ellipse that can be derived from the DOP value. A circular or near circular ellipse indicates a well defined sensor and measurement geometry, and the corresponding size of the ellipse is due to other sources such as range and measurement uncertainties. A semi-major to semi-minor ratio $EllR$ near $1:1$ is considered a low DOP level. As target locations start to move further away from the sensors, the ellipses start to show asymmetries that can impact clustering performance, increasing to the medium DOP level and eventually to high as the ratio continues to increase. The DOP Level values are shown in the piecewise Equation 3.3.

$$DOP = \begin{cases} 0 & \text{if } EllR \leq 1 \\ 1 & \text{if } 1 < EllR \leq 2.5 \\ 2 & \text{if } EllR > 2.5 \end{cases} \tag{3.3}$$

Angle of Arrival

For AOA measurements, DOP increases as the range between sensor and target increases. Although this affect does influence localization performance, it is a known relation and therefore is not used to drive the decision making process for what DOP level is estimated. Instead, the relative angle between the sensors is used to create an estimated DOP. The ideal case is to have orthogonal AOA intersections to minimize the DOP for a given range to target. As the relative angle moves away from 90 degrees the ellipsoid created by

the DOP begins to lengthen as AOA measurements do not have a range associated. An example AOA measurement matrix is provided in Equation 3.4 .

$$H = \begin{bmatrix} -\frac{x_y - s_{1,y}}{||x - s_1||} & \frac{x_x - s_{1,x}}{||x - s_1||} \\ -\frac{x_y - s_{2,y}}{||x - s_2||} & \frac{x_x - s_{2,x}}{||x - s_2||} \\ \cdots & \\ -\frac{x_y - s_{i,y}}{||x - s_i||} & \frac{x_x - s_{i,x}}{||x - s_i||} \end{bmatrix} \tag{3.4}$$

Where x is the target position vector and s_i is the position vector of the ith sensor with $s_{i,x}$ and $s_{i,y}$ being the x and y components of the position vector, respectively. A visual representation of the AOA DOP is shown in Figure 3.6.

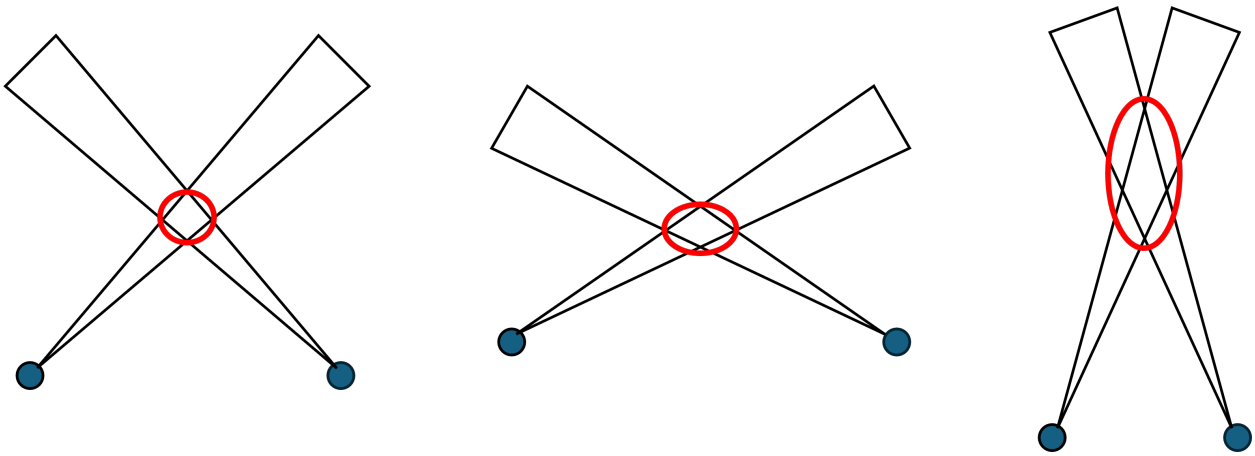

Figure 3.6: AOA DOP, Ideal (Left), Pointing Towards Each Other (Center), Overlapping Pointing in the Same Direction (Right)

Figure 3.6 shows the DOP ellipse shape for three intersecting angles. The ideal case is 90 degrees, and as the angle shifts in either direction, the ellipse begins to elongate as the area where the AOA measurements overlaps grows.

Time of Arrival

TOA measurements create similar DOP relations as AOA measurements except that DOP is independent of the range to the target, since the width of the measurement uncertainty is only related to the measurement uncertainty, as shown in 2.2. The difference from AOA is the relative angle of the semi-major and semi-minor axis of the DOP ellipse is rotated compared to that of AOA for the same sensor positions. As sensors move to a smaller relative angle, the confidence increases in the range estimate but less confident orthogonally. Similarly, if sensors are on either side of a target, separated by roughly 180 degrees, longer sections of the circles created by the measurement overlap, and the relative angles between sensors and target become less confident. An example measurement matrix for TOA measurements is shown in Equation 3.5.

$$H = \begin{bmatrix} \frac{x_x - s_{1,x}}{||x - s_1||} & \frac{x_y - s_{1,y}}{||x - s_1||} \\ \frac{x_x - s_{2,x}}{||x - s_2||} & \frac{x_y - s_{2,y}}{||x - s_2||} \\ \cdots & \\ \frac{x_x - s_{i,x}}{||x - s_i||} & \frac{x_y - s_{2,y}}{||x - s_i||} \end{bmatrix} \tag{3.5}$$

The ideal case has the measurement range rings intersecting in a location where the tangential angles are perpendicular to each other. If the target is located in line between two sensors, the range rings over lap in a geometry that is accurate in terms of range but relative angle to the sensors becomes less obvious. Similarly, as two sensors converge, the range rings begin to overlap creating larger areas of similar weights. A visual representation of the DOP for TOA measurements is shown in Figure 3.7.

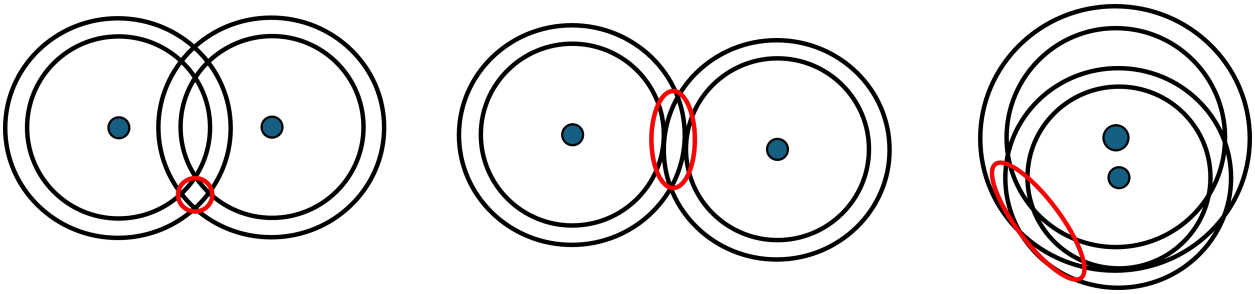

Figure 3.7: TOA DOP, Ideal (Left), In-Line (Center), Overlapping (Right)

Time Difference of Arrival

TDOA measurements have a unique DOP relation due to the hyperbolic shape. The relation between DOP and the relative distance from the target to the sensor pair is a logarithmic relationship, as shown in [20]. That is, a greatest increase in dilution of precision is seen as the target starts to move away from the sensor pair. This causes TDOA only estimates to be very long and narrow when the sensors do not surround the target. An example measurement matrix for TDOA measurements is shown in Equation 3.6.

$$H = \begin{bmatrix} \frac{x-s_1}{||x-s_1||} - \frac{x}{||x||} \\ \frac{x-s_2}{||x-s_2||} - \frac{x}{||x||} \\ \cdots \\ \frac{x-s_i}{||x-s_i||} - \frac{x}{||x||} \end{bmatrix} \tag{3.6}$$

Where x represents the target location with respect to the reference sensor, s_i represents the relative location of the ith sensor relative to the reference sensor [20].

The TDOA dilution of precision is best when the target is surrounded by sensors. A visual representation of different DOP values for the same sensor layout shows the increase in hyperbola overlap as the target moves away from the center, increasing the DOP value as shown in Figure 3.8.

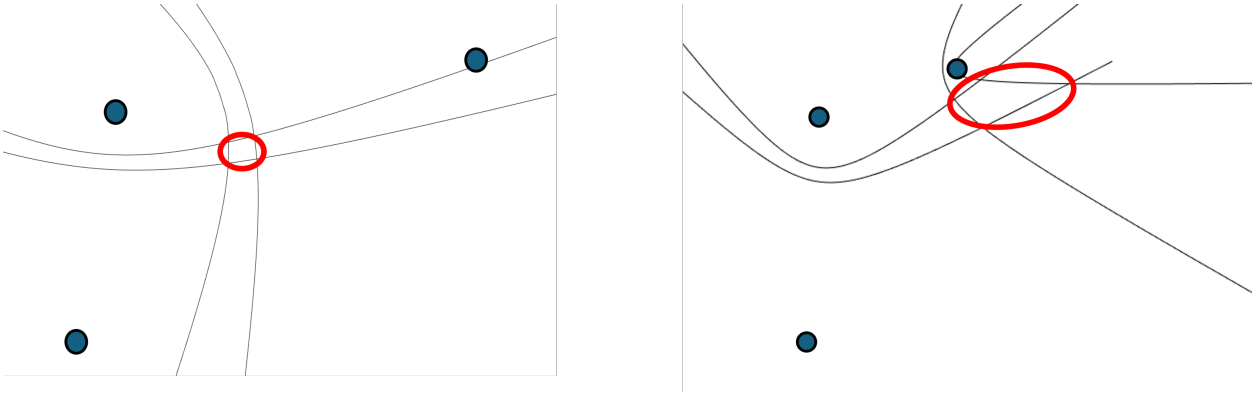

Figure 3.8: TDOA DOP, Ideal (Left), Non-Ideal (Right)

Hybrid

When different measurement types are combined, the combined DOP is produced by combining the measurement matrices of the two measurement types into a single matrix before calculating the DOP. Combining the different measurement types can reduce the DOP relative to the individual measurement types because the geometric relations of the measurements are different. For example, at any range or angle, combined AOA and TOA measurements from the same source intersect orthogonally, resulting in a DOP value that is mostly limited to the measurement quality rather than sensor geometry.

3.5 Algorithm Space

The algorithm space covers all of the different algorithms that are included as options for the algorithm selection problem. These algorithms include K-Means, Mean Shift, and a density-based scan with additive noise (DBSCAN). The selected clustering algorithms are examples of different clustering methodologies. K-Means is considered centroid based, DBSCAN is considered density based, and Mean Shift is a hybrid of both density based and centroid based.

The algorithm space is separated by functional parameters, the adjustment of individual parameters does not count as multiple algorithms per [**?**]. Instead, additional tests were

performed to improve the various tuning parameters for each algorithm. More details about the tuning process are included in each algorithm overview.

3.5.1 K-Means

K-Means is a popular option for clustering algorithms because of its history as one of the first clustering algorithms and for the ease of implementation in different applications. K-Means clusters particles by associating particles to the nearest centroid in an iterative process and recomputing centroid locations. The clustering process is performed for $\boldsymbol{K}$ number of clusters at a given time. Centroid initialization is performed by a sub-algorithm known as K-Means++. If the total number of clusters is unknown, the K-Means process is repeated across a range of potential numbers of clusters, and scores are computed for each number to find the optimal number of clusters for the data set. The final output of the K-Means algorithm is the number of clusters, the cluster centroids, and the particle association to the clusters based on the number of clusters that scored the best.

K-Means++

K-Means++ is an initialization algorithm designed to reduce the total number of iterations needed for K-Means clustering to converge. This reduces the number of iterations by predetermining the most spatially diverse particles from the data set based on the distance to all other centroids. The initial centroid is randomly selected from the set of particles, then subsequent centroids are selected by finding the particle that has the maximum distance to all previously selected centroids. A labeled example of the particle selection process is shown in Figure 3.9

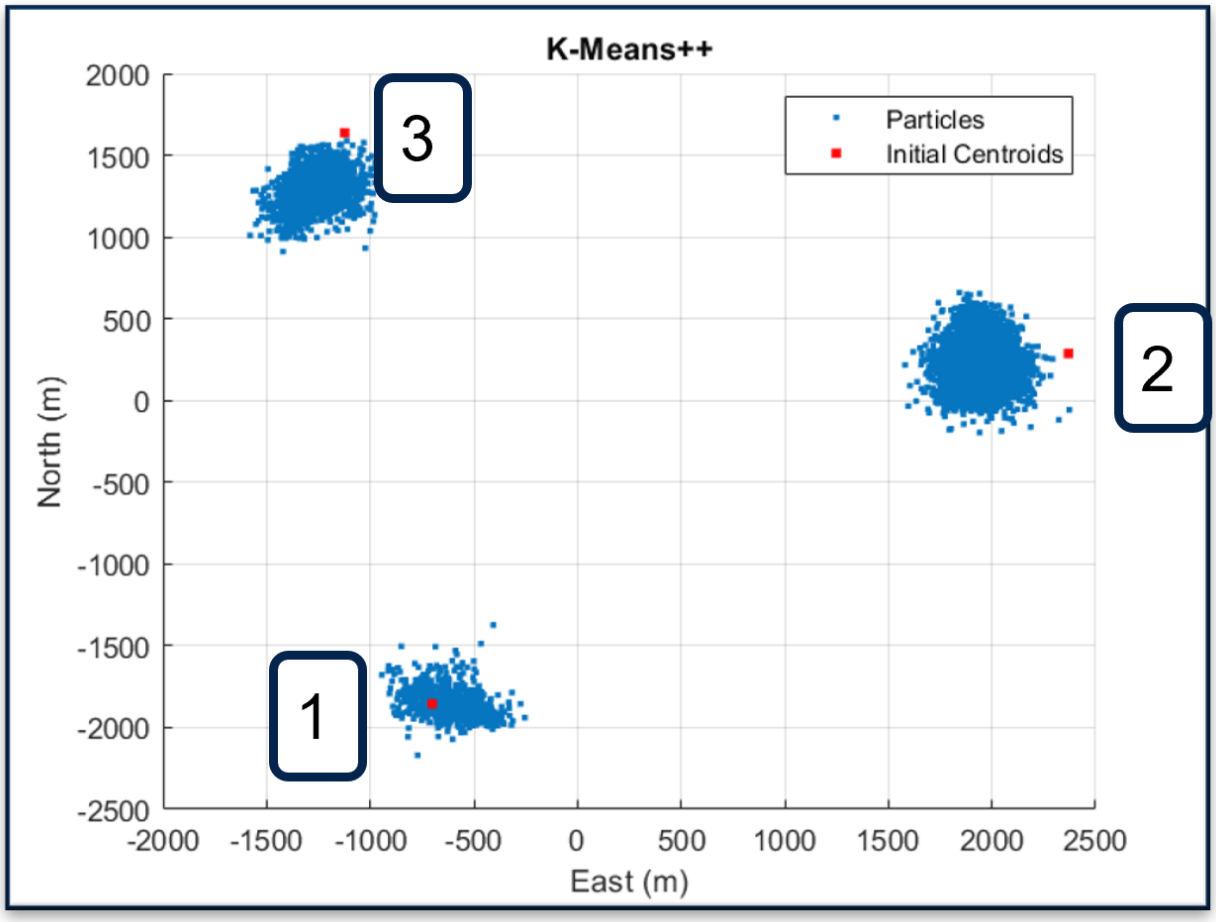

Figure 3.9: K-Means++ Particle Selection

K-Means Clustering For Set K

K-Means for a set K value is an iterative process that cycles between assigning particles to a cluster then recalculating the cluster centroid. The initial centroids come from the K-Means++ process. Particles are associated to the closest centroid using a minimum distance. Once labeled, the centroids are updated to the mean location of all particles that are associated to the cluster. These two steps are repeated iteratively until the centroids no longer move between iterations or the deltas are below a predetermined threshold [23].

Scoring Criteria

K-Means does not directly calculate the number of clusters; instead, a range of values for K needs to be tested. Clustering performance is scored to determine what number of clusters best fits the current particle set. Three different scoring criteria are evaluated for use with K-Means clustering. First, is the Elbow Test which is a traditional approach based on visual characteristics [23]. The second is the Silhouette Test which is a distance-based scoring criteria [25]. Third, the Bayesian information criterion (BIC), which is an information-based approach [26].

Elbow Test

The Elbow Test is based of the visual representation of clustering performance when plotting the relation of within cluster variance on the y-axis and the number of clusters on the x-axis. The "Elbow" is the inflection point where there is the most notable decrease in the delta between one number of clusters and the next, as shown in Figure 3.10.

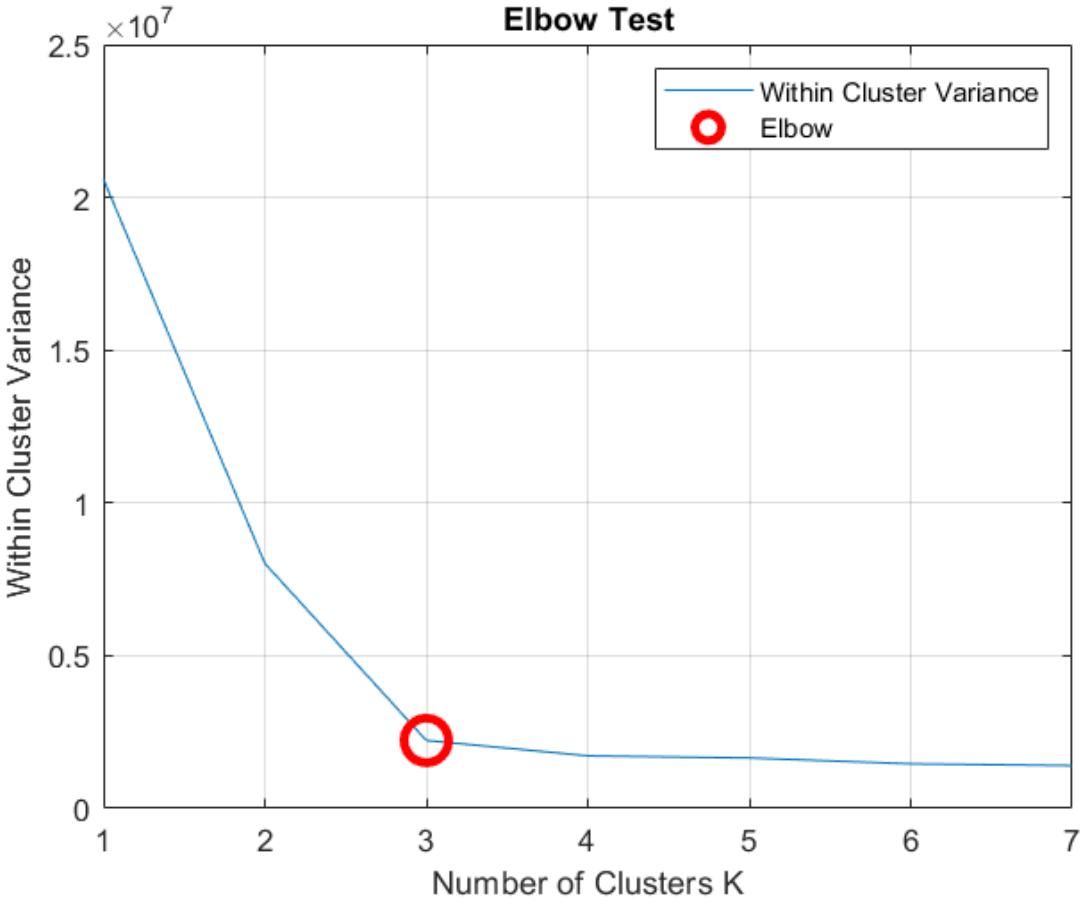

Figure 3.10: Example Elbow Test

The Elbow Test is visually based on the within-cluster variance, but mathematically is based on the derivative of the within-cluster variance with respect to the number of clusters. The Elbow of the curve is where there is the largest delta between two derivative values.

Silhouette Test

The silhouette test is a distance based criterion that evaluates particles on the basis of how similar they are to their labeled group A versus the other clusters C. $a(i)$ is the average dissimilarity of particle i to all other particles in its cluster calculated using Equation 3.7.

$$a(i) = \frac{1}{N_A - 1} \sum_{j \in A, i \neq j} d(i,j) \tag{3.7}$$

Where A is the cluster for particle i, N_A is the number of particles in cluster A, and $d(i,j)$ is the distance between particles i and j [25]. $b(i)$ is the minimum average dissimilarity for all other clusters calculated using Equation 3.8.

$$a(i) = min_{C \neq A}(\frac{1}{N_C} \sum_{l \in C} d(i,l)) \tag{3.8}$$

Where C is all clusters evaluated independently except for cluster A, and l is the individual particles in these other algorithms.

A silhouette value is calculated to score clustering performance combining terms $a(i)$ and $b(i)$. Note that this is only calculated when there is more than one cluster, if only one cluster $s(i) = 0$. Silhouette valuse are calculated using Equation 3.9.

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{3.9}$$

The silhouette value $s(i)$ is averaged $\tilde{s}$ for all particles to create a singular score value for each number of clusters, K, being evaluated. The maximum average $max\tilde{s}(K)$ is selected as the best number of clusters [25].

The silhouette test offers good performance, but is computationally expensive with a time complexity of $O(N^2)$ and was not selected for use. Visually, the difference in ranges are visualized as seen in Figure 3.11.

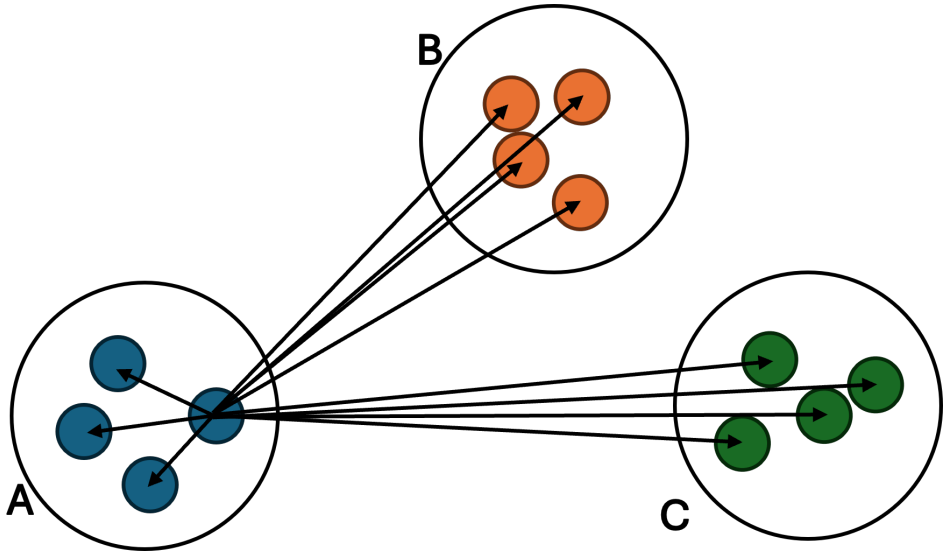

Figure 3.11: Particle Ranges Within Cluster and With Other Clusters

Bayesian Information Criterion

The Bayesian Information Criterion (BIC) score is a third option for K-Means scoring that uses a likelihood given different distribution parameters and a core model to score clustering performance. The BIC score is calculated using Equation 3.10.

$$BIC = k\ln(N) - 2\ln(\hat{L}) \tag{3.10}$$

Where k is the number of parameters estimated, N is the number of particles, and $\hat{L}$ is the maximum value of the model's likelihood function. For clustering, the BIC score is using a Gaussian Mixture MLE model as part of $\hat{L}$. k is calculated using Equation 3.11.

$$k = (K-1) + K \cdot D + K \cdot CovParams \tag{3.11}$$

Where D is the number of dimensions, in this case 2, and $CovParams$ are set to 1 to make a spherical covariance [27].

From prior testing, the BIC score was the most consistent of the scoring algorithms and had a smaller time complexity of $O(N \cdot k)$.

3.5.2 Mean Shift Clustering

Mean shift, also commonly referred to as mode seeking or gradient ascent, is a general procedure designed to shift data points to the average within a local area. The underlying principles of mean shift can be applied to a set of particles to act as a clustering algorithm by adding convergence parameters and labeling functionality. Traditionally, mean shift clustering is only tuned using a bandwidth parameter ϵ, but a minimum points $minPts$ value was added to reduce the amount of clusters formed by noise particles. The general function of the mean shift algorithm is a two step iterative process from [28]:

Repeat steps until $\nabla f(x_i) = 0$:

1) Calculate the mean shift vector $m(x_i^t)$ using Equation 3.12

$$m(x_i^t) = \frac{\sum_{i=1}^{N} x_i g(||\frac{x-x_i}{\epsilon}||^2)}{\sum_{i=1}^{N} g(||\frac{x-x_i}{\epsilon}||^2)} - x \tag{3.12}$$

2) Translate density estimation window: $x_i^{t+1} = x_i^t + m(x_i^t)$

Where $m(x_i^t)$ is the mean shift vector for particle x_i at time t and g is the kernel function used to change the particles. A visual example of the iterative process is shown in Figure 3.12.

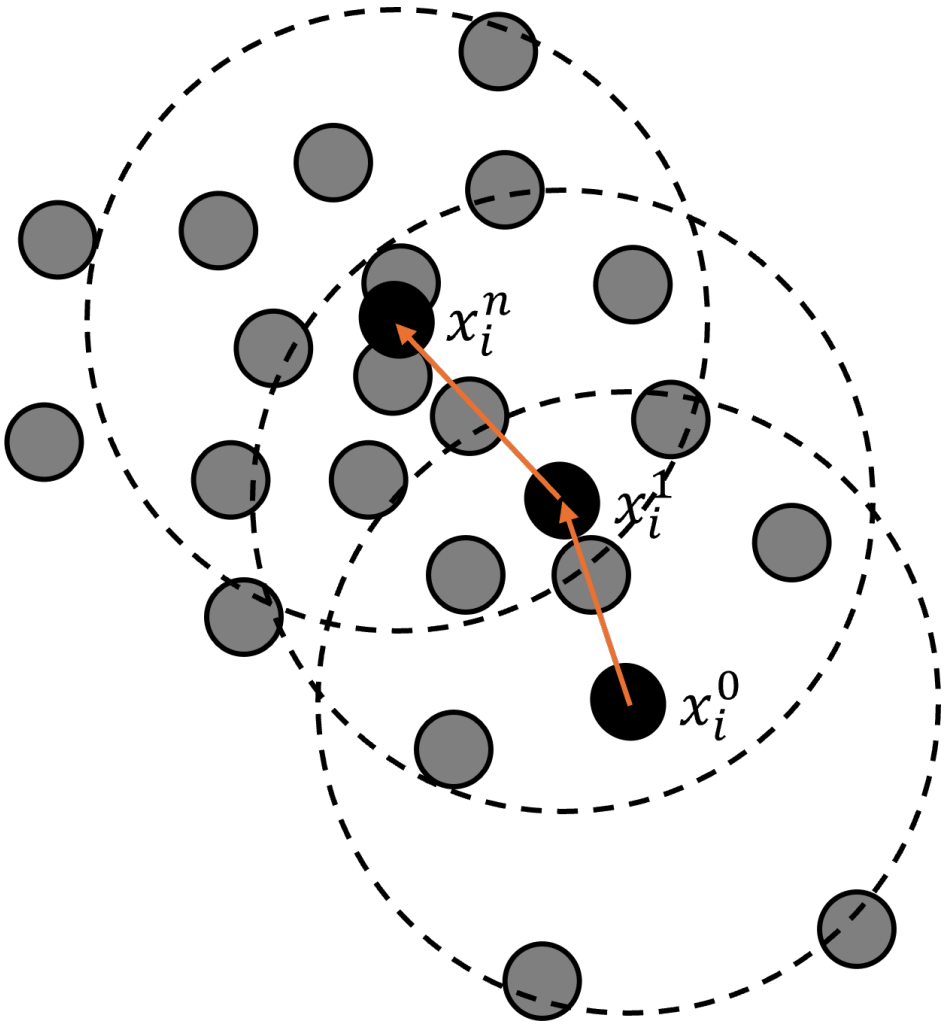

Figure 3.12: Example of Particle Movement During Mean Shift Clustering

Different Kernel functions are used to select which particles are included in the gradient calculation and how they are weighted. This implementation uses a flat kernel so that all particles outside of the bandwidth are not included in calculating the local gradient. Equation 3.13 shows the mathematical representation of a flat kernel with d being the distance between two particles.

$$F(d) = \begin{cases} 1 & \text{if } ||d|| \leq \epsilon \\ 0 & \text{if } ||d|| > \epsilon \end{cases} \tag{3.13}$$

3.5.3 Density-Based Spatial Clustering of Applications with Noise

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering algorithm that identifies clusters based on the density of particles. DBSCAN operates by finding continuous areas of particles that maintain a minimal particle density, allowing for ambiguous shaped clusters. Unlike K-Means, DBSCAN is capable of simultaneously clustering particles and estimating the number of clusters. DBSCAN algorithm is tuned using a bandwidth ϵ and minimum points value $minPts$.

DBSCAN operates by first marking all particles as unlabeled. The algorithm then starts with the first particle and calculates the number of neighboring particles. Neighbor particles are particles in which the distance to the center particle is less than the bandwidth ϵ. If there are more neighbor particles than $minPts$ the current particle being evaluated is labeled as a core particle for the current cluster label. Then, each neighbor particle is labeled as part of this new cluster and checked if it meets the criteria to be a core particle. If one of the original neighbor particles qualifies as a core particle, it adds any new neighbor particles to the cluster. This process is continued recursively for every neighbor particle and neighbor particles added to the original grouping until the cluster has evaluated all potential neighbor particles. This process results in a cluster comprised of core particles and neighbor particles, all labeled as part of the cluster. This process is then completed for any non labeled particles. If a particle is evaluated for starting a new cluster and does not reach $minPts$ within ϵ test, it is labeled a noise particle.If the noise particle is not picked up by any other cluster by the end of the algorithm, it becomes ignored during the state estimation step. Figure 3.13 shows the relation between different particle labels.

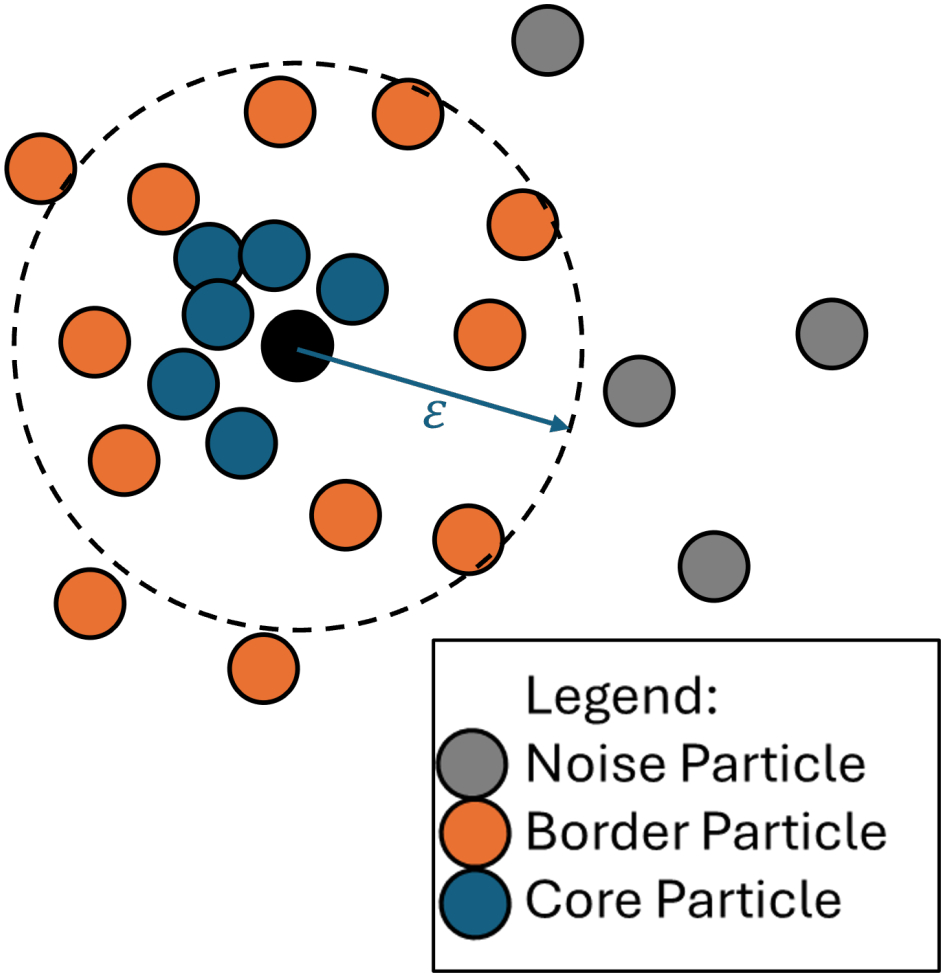

Figure 3.13: Example of Particle Labeling During DBSCAN Clustering

3.5.4 Clustering Errors Due to Improper Algorithm Selection

The motivation behind developing an adaptive clustering mechanism comes from experiences in which certain clustering algorithms would not work under certain circumstances. Each algorithm has an area in which it performs better than the other algorithms, and an adaptive selection highlights those areas to get the best performance across the board.

Example Case 1 Non-Symmetrical Clusters

An example scenario where some of the clustering algorithms fall short is in the case of non-symmetrical clusters. K-Means clustering uses variance as part of its scoring mechanic,

which results in favoring symmetrical clusters that maximize area with minimal variance. Non-symmetrical clusters can form when the higher weight areas of measurement likelihood run closer to parallel and create longer and narrower clusters such as the one shown in Figure 3.14.

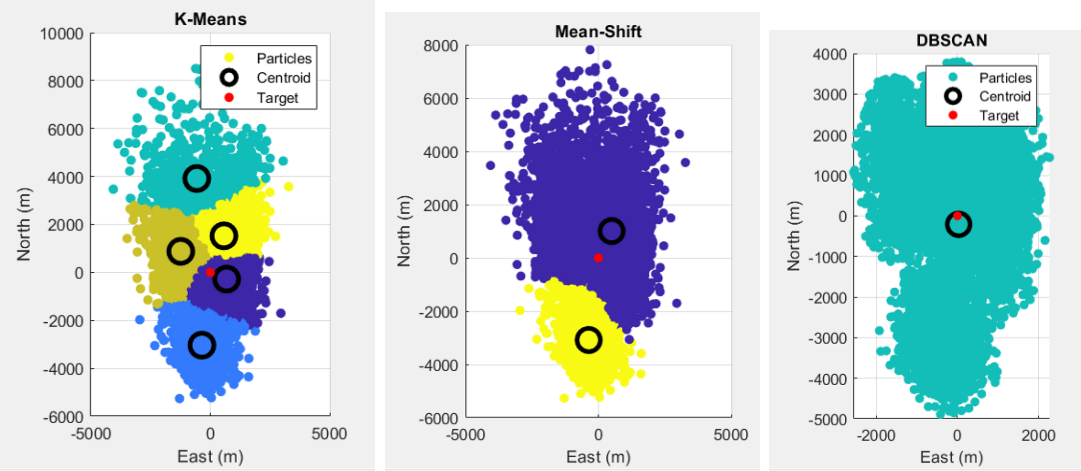

Figure 3.14: Clustering Results for Non-Symmetrical Cluster

Without user input or adaptation, the K-Means algorithm clusters the narrow group of particles into multiple sections to reduce the average variance in each cluster. The number of splits depends on the relative dimensions of the cluster. Mean shift can also split apart larger non-symmetric clusters if the tuning parameters are tuned incorrectly or the particle density does not form a steep enough gradient relative to the size of the cluster. Alternatively, DBSCAN operates by finding continuous groups of particles rather than variance allowing for non-symmetrical and diverse shapes of particle distributions. In the case of a longer narrow cluster DBSCAN groups all particles together into one distribution.

Example Case 2 Close Proximity Clusters

The same characteristics that made DBSCAN a better algorithm choice in the first example make it a non-ideal choice in situations where multiple targets are in close proximity

to one another. If measurement noise values are large enough so that the two corresponding peaks overlap, the DBSCAN algorithm will cluster the two distributions into one. An example scenario is shown below in Figure 3.15 where multiple clusters are forming in close proximity to each other.

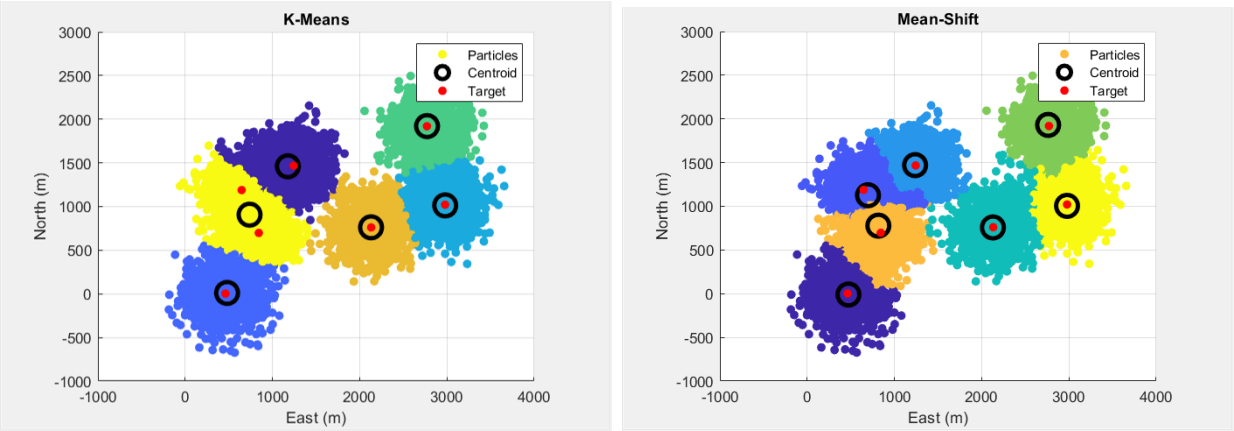

Figure 3.15: Clustering Results for Close Proximity Clusters - K-Means (left) and Mean Shift (right)

In this example, Figure 3.15 shows K-Means and Mean Shift algorithms are capable of clustering multiple densely packed targets. In this sample K-Means does combine two targets into one cluster, which is a possible outcome when targets are too close together. The clustering results for DBSCAN are shown in Figure 3.16.

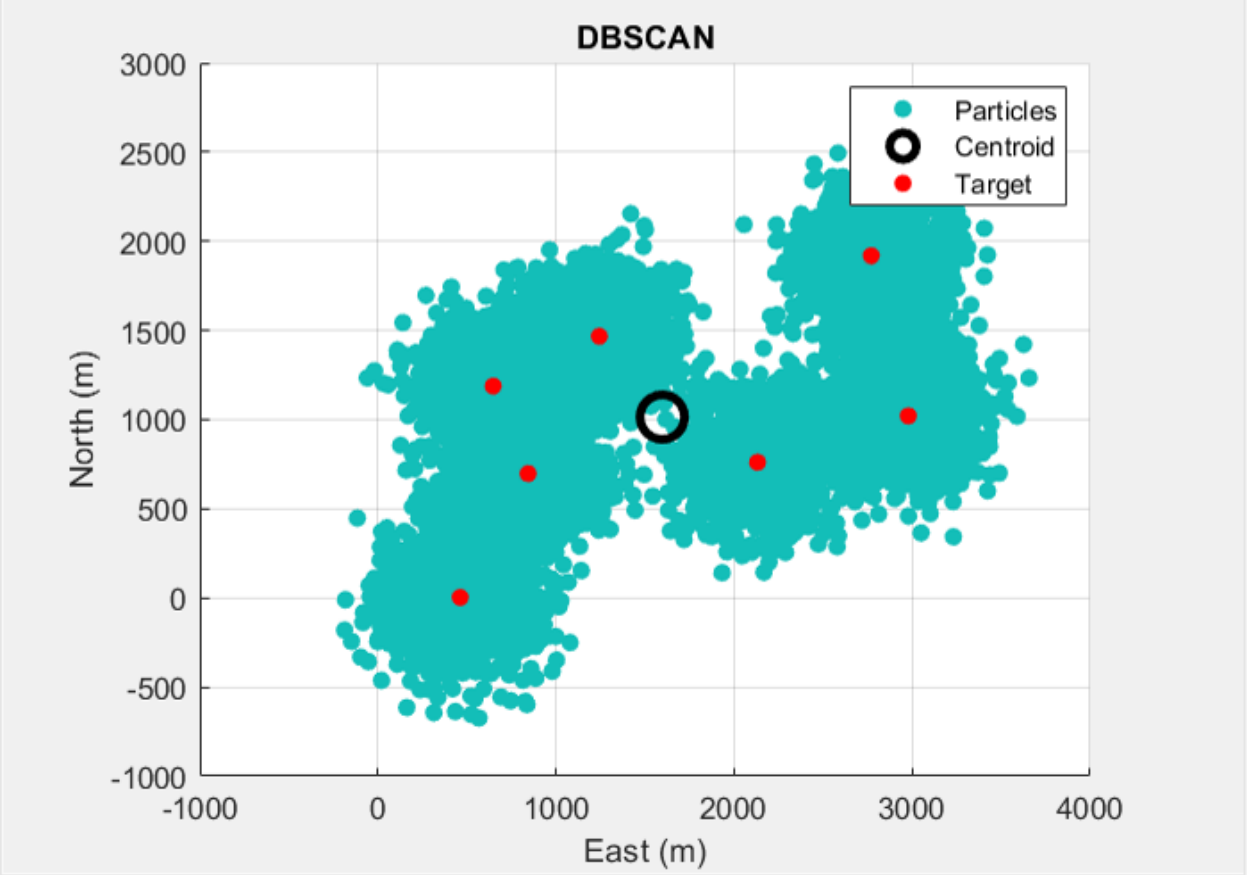

Figure 3.16: Clustering Results for Close Proximity Clusters - DBSCAN

In the case of overlapping clusters, DBSCAN will link continuous clusters if there is a high enough density of particles at the boundary of the two clusters. In this example case, the clusters create a continuous chain, resulting in DBSCAN assigning every particle to the same cluster.

3.6 Performance Space

The performance space takes the output of the algorithm space and evaluates the algorithm performance. Algorithm performance is evaluated on clustering the correct number of clusters and the overall localization error to ensure that the clusters generated are accurate for the overall system.

The performance space acts as a feedback loop to aid in selection map creation. As different algorithms are scored for each set within the problem space, the best performing algorithm is assigned in the selection map.

3.6.1 Evaluating Algorithm Performance

Algorithm performance was evaluated using a several step verification process. First, the general performance of the particle filter was checked to see if the run was valid for clustering. The run validation required the minimum number of particles points to be around each target to ensure the possibility of success for the clustering algorithm. These runs were ignored because they are representative of errors with the particle filter specifically rather than the clustering algorithm and could skew results. Valid run percentages can be found in the appendix.

Valid runs were clustered by each algorithm under test, and results for each run were stored. The results saved for each run include particle locations, cluster ids for each particle for each algorithm, and cluster centroids for each cluster for each algorithm for each cluster.

Algorithm performance is post processed to gather four main performance statistics, average number of clusters error, number of clusters variance, average centroid error, and centroid error variance. A simple scoring parameter is computed to weight the performance to a single term $Score_{cluster}$ combining the average and variance with the minimum score value corresponding to the best algorithm choice using Equation 3.14.

$$Score_{cluster} = l \cdot \mu^2_{cluster} + m \cdot \sigma^2_{cluster} \tag{3.14}$$

Here l and m are arbitrarily set tuning parameters to adjust weighting between accuracy and precision. A value of $l = 1$ and $m = 2$ is used for meta-data creation. The same formula is used for scoring the clustering error. The average error and standard deviation are squared to make the values always positive and to make the higher errors worse. The standard deviation value is scaled to a higher value using a constant value to prioritize consistency.

The same value is computed for centroid error and used as a tie-breaker in the case the algorithms have similar performance. In the case where multiple algorithms have similar performances for both clustering and localization accuracies, the surrounding problem subspaces were evaluated to make the logic tree simpler to make the subspaces consistent.

3.6.2 Selection Mapping

After meta-data is created, algorithm performance is processed to evaluate which algorithms map best to which subset of the problem space. In operation, the selection map operates as a logic tree within the particle filter that takes in scenario parameters and selects which algorithm to run, removing the Performance space from the loop. Figure 3.17 is a visual representation of the selection map in matrix form showing the different problem space sets and the corresponding algorithm for that set.

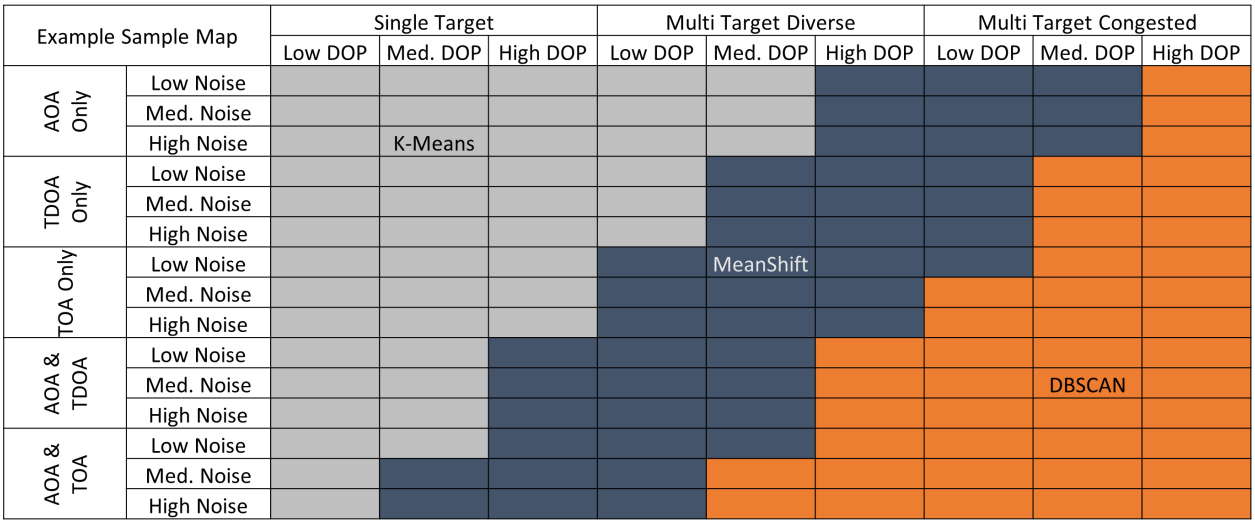

Figure 3.17: Visual Representation of Sample Mapping

3.7 Chapter Conclusion

Chapter 3 expands upon the use of particle filtering in multiple target tracking applications and adaptations needed to allow for estimating multiple targets including the concept

of clustering. Clustering is an unsupervised machine learning process that has many different algorithms available in literature. To select the best clustering algorithms to use and when to use them, the concepts of the ASP and meta-learning are introduced. These concepts provide a framework to thoroughly evaluate algorithm performance across a large range of scenario factors referred to as the problem space.

Three clustering algorithms are introduced as part of the algorithm space. K-Means is a traditional clustering technique that uses particle centroids to create distinct cluster regions. Mean shift clustering uses local gradients in particle densities to form clusters around density centers. Lastly, DBSCAN uses continuous regions of a minimum clustering distance to associate particles together.

Performance metrics for clustering accuracy is defined to form the performance space. The performance space correlates problem sets, combinations of scenario factors, with clustering algorithms. The concept of a selection map is introduced to link the problem and algorithm space using the performance space to drive selections.

Chapter 4

Adaptive Clustering Engine

The Adaptive Clustering Engine (ACE) is a logic based algorithm that takes in scenario factors and determines which clustering algorithm is expected perform best. ACE is designed with adaptability in mind to not limit overall localization accuracy based on clustering technique selection. The internal logic tree of ACE was created using Monte Carlo simulation results evaluating different scenarios and combinations of measurements. ACE was tested against the same training data used to create the algorithm map, as well as additional randomly generated scenarios with varying parameters to validate performance.

4.1 Meta-Data Creation

Meta-Data for ACE was generated using a series of Monte-Carlo simulations with varying parameters, sensor positions, and target positions. This section discusses how the different simulation parameters are simulated.

4.1.1 Clustering Simulation

The three clustering algorithms were evaluated using a series of simulation runs with varying parameters that fully encompass the Problem Space. The scenario variables include single target versus multiple targets, levels of Dilution of Precision (DOP), different levels of measurement noise, and combinations of different measurement types. For the multiple target case, a variable number of targets is used in addition to the proximity of targets to each other. Every combination of the problem space was evaluated independently to create a full set of meta-data. Simulations were run starting with an uninitialized particle filter

and then run through five resampling steps to allow the filter to start forming a well defined distribution.

The general scenario layout is a 2,000 meter ring of sensors randomly spaced out moving in different directions. The different levels of DOP correspond to what range bands from the center of the ring the targets will be present in starting with low DOP scenarios having sensors fully encompassing the targets. The medium DOP case has targets starting at the same radius as the sensors and extending outward a limited amount. Lastly, high DOP scenarios operate in a range beyond the medium case.

Target locations are randomly generated using a random distribution for range between the DOP limits and angularly distributed using a random starting phase and different angular spacings depending on the number of targets and if it is running the close proximity case. These scenario outlines allow for a large variety of controlled scenarios for use in Monte Carlo simulation. A visual representation of the simulation layout is available in Figure 4.1.

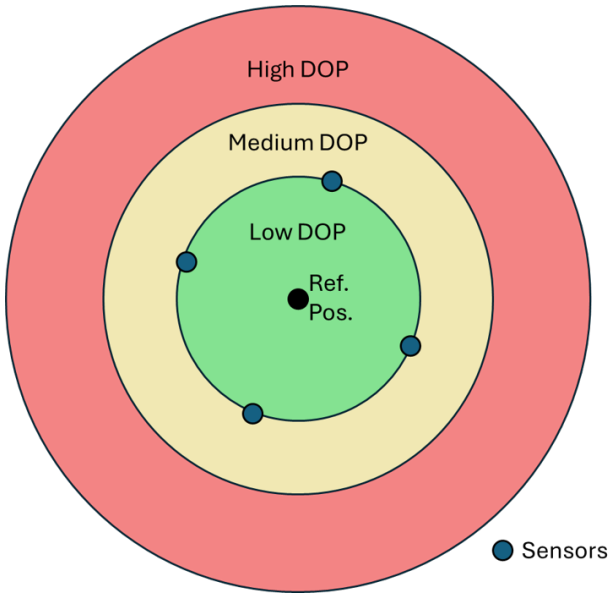

Figure 4.1: Visual of Scenario Layout

4.1.2 Clustering Algorithm Performance

The simulation data was processed to select the best algorithms for each entry of the selection map following the methodology presented in 3.6. The first set of results is the error for the number of clusters formed. The ideal answer for all values in this table is zero, a negative value means that the clustering algorithm estimated fewer clusters than targets, and a positive value means that it estimated more clusters than targets.

Number of Clusters		Single Target			Multi Target Diverse			Multi Target Congested		
K-Means		Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP
AOA Only	Low Noise	0.0	0.0	2.0	0.0	-0.2	1.0	-1.6	-0.4	0.8
	Med. Noise	0.0	0.0	2.2	-0.3	-0.1	0.9	-1.5	-0.1	0.7
	High Noise	0.0	0.0	2.3	-1.0	-0.2	1.0	-3.0	-0.5	0.6
TDOA Only	Low Noise	0.1	2.9	5.8	-1.0	0.6	2.1	-1.3	1.0	2.2
	Med. Noise	0.2	4.3	5.9	-0.9	1.4	2.4	-0.6	1.8	2.7
	High Noise	0.1	4.0	5.9	-0.4	1.8	2.5	-0.8	2.1	2.7
TOA Only	Low Noise	0.1	0.5	2.7	-0.3	-0.2	0.2	-0.3	-0.1	1.4
	Med. Noise	0.1	0.5	3.0	-0.4	-0.2	0.1	-1.0	-0.1	1.6
	High Noise	0.0	0.3	3.0	-0.6	-0.1	0.3	-1.2	0.3	1.6
AOA + TDOA	Low Noise	0.0	1.0	2.3	0.0	-0.4	0.9	-0.4	-0.4	1.0
	Med. Noise	0.0	0.0	2.1	0.0	-0.2	1.0	-1.0	-0.2	0.9
	High Noise	0.0	0.0	2.6	-1.1	-0.2	1.0	-1.5	-0.3	0.9
AOA + TOA	Low Noise	0.0	0.0	2.4	-0.3	-0.2	0.3	-0.4	-0.2	0.3
	Med. Noise	0.0	0.1	2.7	-0.5	-0.5	0.4	-1.1	-0.4	0.4
	High Noise	0.0	0.2	2.6	-0.2	-0.2	0.0	-1.3	-0.4	0.2

Table 4.1: K-Means Clustering Error

Number of Clusters		Single Target			Multi Target Diverse			Multi Target Congested		
Mean Shift		Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP
AOA Only	Low Noise	0.0	0.0	0.0	0.0	-0.2	0.0	-0.6	0.4	0.2
	Med. Noise	0.0	0.0	0.0	-0.1	-0.1	0.0	0.2	0.3	2.8
	High Noise	0.4	0.1	0.1	0.4	0.0	0.8	1.9	6.6	15.2
TDOA Only	Low Noise	0.0	0.0	0.1	-0.4	-0.5	1.0	-0.3	-0.7	1.7
	Med. Noise	0.0	0.1	0.2	-0.2	2.7	3.9	-0.2	3.8	6.0
	High Noise	0.0	1.0	0.4	-0.3	8.1	8.0	-0.5	8.0	7.6
TOA Only	Low Noise	0.3	1.2	1.6	-0.3	-0.3	-0.3	-0.5	-0.4	-0.2
	Med. Noise	0.0	0.0	0.0	-0.4	-0.4	-0.5	-1.3	-0.5	-0.1
	High Noise	0.0	0.0	0.0	-0.5	-0.1	-0.2	-1.3	-0.1	0.1
AOA + TDOA	Low Noise	0.0	0.1	0.0	0.1	-0.4	0.0	-0.6	-0.4	0.0
	Med. Noise	0.0	0.0	0.0	0.1	-0.2	0.0	-1.0	-0.3	0.0
	High Noise	0.3	0.0	0.1	-0.1	-0.1	0.7	-1.0	-0.1	0.7
AOA + TOA	Low Noise	0.0	0.0	0.0	-0.3	-0.3	-0.3	-0.5	-0.2	-0.4
	Med. Noise	0.0	0.0	0.0	-0.5	-0.5	-0.3	-1.3	-0.6	-0.3
	High Noise	0.0	0.0	0.0	-0.8	-0.3	-0.5	-1.7	-0.6	-0.4

Table 4.2: Mean Shift Clustering Error

From the clustering error tables general trends can be made that are validated by the scoring value for the performance space. For example, in lower DOP scenarios, K-Means starts to overestimate number of targets because it is splitting larger clusters into multiples as discussed in section 3.5.4. Another informative trend is the low error for DBSCAN for all

Number of Clusters		Single Target			Multi Target Diverse			Multi Target Congested		
DBSCAN		Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP
AOA Only	Low Noise	0.0	0.0	0.0	-0.4	-0.3	0.0	-2.3	-0.3	-0.1
	Med. Noise	0.0	0.0	0.0	-1.6	-0.1	0.0	-2.2	0.2	0.2
	High Noise	0.0	0.0	0.1	-2.3	-0.4	0.1	-1.3	4.6	3.8
TDOA Only	Low Noise	0.0	0.0	0.0	-0.4	-0.8	-0.5	-0.6	-0.9	-0.8
	Med. Noise	0.0	0.0	0.0	-0.5	-0.7	-0.8	-0.9	-1.0	-0.6
	High Noise	0.0	0.0	0.0	-0.6	-0.8	-0.7	-1.4	-0.7	-1.2
TOA Only	Low Noise	0.0	0.0	0.1	-0.4	-0.3	-0.3	-0.8	-0.4	-0.2
	Med. Noise	0.0	0.0	0.0	-0.7	-0.4	-0.6	-1.5	-0.5	-0.1
	High Noise	0.0	0.0	0.0	-1.2	-0.2	-0.3	-1.8	-0.5	-0.1
AOA + TDOA	Low Noise	0.0	0.0	0.0	-0.3	-0.4	0.0	-2.0	-0.5	0.0
	Med. Noise	0.0	0.0	0.0	-1.5	-0.4	0.0	-2.5	-0.4	-0.1
	High Noise	0.0	0.0	0.0	-2.4	-0.3	0.1	-2.7	-0.8	0.0
AOA + TOA	Low Noise	0.0	0.0	0.0	-0.3	-0.3	-0.3	-0.6	-0.3	-0.4
	Med. Noise	0.0	0.0	0.0	-0.8	-0.6	-0.3	-1.5	-0.7	-0.3
	High Noise	0.0	0.0	0.0	-0.8	-0.6	-0.6	-2.0	-0.9	-0.5

Table 4.3: DBSCAN clustering Error

single target cases, This result shows the method in which DBSCAN creates clusters aids in the single target case because the particles are all in one continuous group and it is more difficult to split clusters compared to alternative algorithms.

The sample table format can be used to display the standard deviation for the estimate distribution. The standard deviation is used in the scoring criteria to help select algorithms that are more consistent in the case the average errors are comparable.

Std. Dev. of Clusters		Single Target			Multi Target Diverse			Multi Target Congested		
K-Means		Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP
AOA Only	Low Noise	0.2	0.0	1.2	0.3	0.6	0.1	1.0	0.7	0.6
	Med. Noise	0.0	0.0	1.0	0.8	0.4	0.2	1.6	1.1	0.7
	High Noise	0.0	0.0	0.8	1.2	0.9	0.0	0.0	1.3	0.7
TDOA Only	Low Noise	0.5	2.3	0.4	1.0	1.3	0.8	1.0	1.5	1.0
	Med. Noise	0.7	1.7	0.3	1.0	1.1	0.9	1.1	1.1	0.7
	High Noise	0.6	2.2	0.3	1.2	1.2	0.7	1.4	1.2	1.0
TOA Only	Low Noise	0.8	1.1	1.1	0.7	0.7	0.7	1.0	0.9	1.2
	Med. Noise	0.6	1.0	1.1	0.9	1.1	1.0	1.6	1.0	1.1
	High Noise	0.0	0.7	1.1	0.9	0.8	0.8	1.4	1.1	1.3
AOA + TDOA	Low Noise	0.0	1.4	1.0	0.3	0.6	0.3	1.0	0.7	0.1
	Med. Noise	0.0	0.0	1.1	0.5	0.5	0.1	1.3	0.9	0.3
	High Noise	0.0	0.0	0.9	1.3	0.7	0.0	1.3	1.0	0.4
AOA + TOA	Low Noise	0.0	0.0	0.9	0.7	0.7	0.8	0.7	0.5	0.9
	Med. Noise	0.0	0.5	0.9	0.9	0.9	0.6	1.1	0.8	0.8
	High Noise	0.0	0.6	0.9	0.7	0.7	0.7	1.2	1.0	1.1

Table 4.4: K-Means Clustering Standard Deviation

Using the algorithm scoring equation, Equation 3.14, and the tabulated results in this section, a table of algorithm scores is created that can then be used to map out which algorithm performs best for each problem set. Algorithm scores are shown in Table 4.1.2.

Std. Dev. of Clusters		Single Target			Multi Target Diverse			Multi Target Congested		
Mean Shift		Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP
AOA Only	Low Noise	0.0	0.0	0.0	0.2	0.6	0.0	1.5	1.4	1.7
	Med. Noise	0.2	0.0	0.0	0.7	0.3	0.1	2.7	2.7	5.5
	High Noise	0.5	0.2	0.2	2.0	0.7	1.1	6.6	6.1	10.9
TDOA Only	Low Noise	0.0	0.3	0.3	0.6	1.2	2.4	0.7	0.9	2.6
	Med. Noise	0.0	0.5	0.5	1.3	4.2	3.7	2.1	5.3	5.2
	High Noise	0.0	2.0	1.0	0.9	7.5	6.3	2.3	5.5	7.8
TOA Only	Low Noise	1.5	8.7	11.6	0.9	0.6	0.5	1.2	0.6	0.7
	Med. Noise	0.0	0.0	0.0	0.9	1.1	0.8	1.4	0.8	0.7
	High Noise	0.0	0.0	0.0	1.1	0.9	0.8	1.6	1.1	1.1
AOA + TDOA	Low Noise	0.0	0.2	0.0	0.2	0.6	0.0	1.1	0.8	0.0
	Med. Noise	0.1	0.0	0.0	0.5	0.5	0.0	1.3	0.8	0.2
	High Noise	0.5	0.1	0.3	1.8	0.5	0.8	1.6	2.5	1.1
AOA + TOA	Low Noise	0.0	0.0	0.0	0.6	0.6	0.7	0.7	0.5	0.7
	Med. Noise	0.0	0.0	0.0	0.8	0.9	0.5	1.1	0.8	0.6
	High Noise	0.0	0.0	0.0	0.9	0.7	0.7	1.4	1.0	1.1

Table 4.5: Mean Shift Clustering Standard Deviation

Std. Dev. of Clusters		Single Target			Multi Target Diverse			Multi Target Congested		
DBSCAN		Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP
AOA Only	Low Noise	0.0	0.0	0.0	0.7	0.6	0.0	1.0	0.7	0.5
	Med. Noise	0.0	0.0	0.0	1.0	0.3	0.0	1.2	2.4	1.6
	High Noise	0.1	0.1	0.2	0.8	0.7	0.3	2.1	4.5	4.2
TDOA Only	Low Noise	0.0	0.0	0.0	0.6	0.9	0.8	1.0	1.0	0.9
	Med. Noise	0.0	0.0	0.0	0.7	0.8	0.9	1.1	1.0	0.8
	High Noise	0.0	0.0	0.0	0.8	1.0	1.1	1.4	1.1	1.2
TOA Only	Low Noise	0.0	0.0	0.6	0.8	0.7	0.5	1.3	0.7	0.7
	Med. Noise	0.0	0.0	0.0	0.8	1.0	0.9	1.3	0.9	0.8
	High Noise	0.0	0.0	0.0	1.0	0.8	0.7	1.3	1.2	1.0
AOA + TDOA	Low Noise	0.0	0.0	0.0	0.6	0.6	0.0	1.1	0.8	0.0
	Med. Noise	0.0	0.0	0.0	1.0	0.6	0.0	0.9	0.9	0.5
	High Noise	0.1	0.1	0.0	0.9	0.8	0.2	0.6	1.1	0.5
AOA + TOA	Low Noise	0.0	0.0	0.0	0.6	0.6	0.6	0.8	0.6	0.7
	Med. Noise	0.0	0.0	0.0	1.0	0.9	0.5	1.2	0.9	0.6
	High Noise	0.0	0.0	0.0	1.1	0.7	0.7	1.2	0.7	0.9

Table 4.6: DBSCAN Clustering Standard Deviation

Algorithm Scoring for Each Problem Set			Single Target			Multi Target Diverse			Multi Target Close Proximity		
			Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP
AoA only	Low Noise	K-Means	0.0727	0.0000	6.5322	0.2037	0.8542	1.0004	4.6824	1.3134	1.3250
		MS	0.0000	0.0000	0.0000	0.0800	0.8542	0.0000	5.0947	4.3077	5.8124
		DBSCAN	0.0000	0.0000	0.0000	1.1150	0.8850	0.0000	7.4591	1.0071	0.5917
	Med. Noise	K-Means	0.0000	0.0000	7.0745	1.4694	0.3642	0.9987	7.0867	2.5152	1.3593
		MS	0.0800	0.0000	0.0000	1.0167	0.1597	0.0404	15.0421	14.2898	67.7864
		DBSCAN	0.0000	0.0000	0.0000	4.4376	0.1975	0.0000	7.9725	11.5717	4.8762
	High Noise	K-Means	0.0000	0.0000	6.8061	3.8049	1.4722	1.0000	9.0000	3.3769	1.5072
		MS	0.7314	0.1187	0.1187	8.2196	1.1246	3.0811	91.6733	118.8215	467.9603
		DBSCAN	0.0404	0.0404	0.1187	6.3816	1.1003	0.2382	10.5130	61.7688	50.6353
TDoA only	Low Noise	K-Means	0.5456	18.9623	33.5498	2.8333	3.7488	5.7916	3.9587	5.2727	6.9968
		MS	0.0000	0.1616	0.1937	0.7761	3.0110	12.4285	1.1389	2.1077	16.8842
		DBSCAN	0.0000	0.0000	0.0000	0.7761	2.4483	1.4168	2.2914	2.8166	2.0622
	Med. Noise	K-Means	0.9867	24.3310	34.9937	2.7486	4.3607	7.3390	2.9119	5.6431	8.2547
		MS	0.0000	0.4286	0.4969	3.1875	42.0662	42.5025	8.6356	70.7077	89.3853
		DBSCAN	0.0000	0.0000	0.0000	1.1455	1.7596	2.4154	3.4644	2.8074	1.7424
	High Noise	K-Means	0.6735	26.1959	35.1966	2.9415	6.0424	7.1892	4.7570	7.2487	9.1111
		MS	0.0000	9.4049	2.2475	1.8791	178.8945	143.1429	10.8342	125.4545	180.1420
		DBSCAN	0.0000	0.0000	0.0000	1.7166	2.6940	2.9206	5.6293	2.7677	4.3827
TOA only	Low Noise	K-Means	1.1848	2.7085	9.6746	1.1899	1.1233	1.1552	2.2505	1.6929	4.7013
		MS	4.8146	152.2230	272.8055	1.6489	0.8889	0.6323	3.2954	0.9815	1.1446
		DBSCAN	0.0000	0.0000	0.6336	1.4038	0.9441	0.6323	4.0263	1.2116	1.0862
	Med. Noise	K-Means	0.6336	2.1901	11.7172	1.7310	2.5487	1.8622	5.8713	1.9233	4.8559
		MS	0.0000	0.0000	0.0000	1.9223	2.3697	1.5158	5.2582	1.5158	0.9377
		DBSCAN	0.0000	0.0000	0.0000	1.8350	2.3265	1.9070	5.7283	1.7459	1.1626
	High Noise	K-Means	0.0000	1.2044	11.4020	1.9985	1.1406	1.4448	5.4429	2.6530	5.9314
		MS	0.0000	0.0000	0.0000	2.4881	1.5326	1.2841	7.1159	2.5372	2.4235
		DBSCAN	0.0000	0.0000	0.0000	3.2322	1.3314	1.2225	6.4965	2.9323	1.8521
AOA + TDOA	Low Noise	K-Means	0.0000	4.6735	7.2142	0.2037	0.9067	0.9966	2.1047	1.1734	1.0002
		MS	0.0000	0.1187	0.0000	0.1187	0.9067	0.0000	2.9623	1.4129	0.0000
		DBSCAN	0.0000	0.0000	0.0000	0.8980	0.9329	0.0000	6.4969	1.5157	0.0000
	Med. Noise	K-Means	0.0000	0.0000	6.9610	0.5000	0.6114	1.0004	4.2111	1.5506	0.9956
		MS	0.0242	0.0000	0.0000	0.4475	0.5619	0.0000	4.4930	1.2944	0.0460
		DBSCAN	0.0000	0.0000	0.0000	4.3964	0.9517	0.0000	7.8559	1.7067	0.4137
	High Noise	K-Means	0.0000	0.0000	8.1331	4.4016	1.0433	1.0000	5.6589	2.1895	1.0947
		MS	0.5715	0.0400	0.1937	6.1510	0.4565	1.8345	5.9784	12.2215	3.1548
		DBSCAN	0.0404	0.0201	0.0000	7.4591	1.2245	0.1236	7.8714	3.1934	0.5783
AOA + TOA	Low Noise	K-Means	0.0000	0.0000	7.2836	1.0001	1.1403	1.5097	1.1161	0.6193	1.8001
		MS	0.0000	0.0000	0.0000	0.9403	0.9112	0.9530	1.3476	0.6193	1.1996
		DBSCAN	0.0000	0.0000	0.0000	0.9403	0.8810	0.8110	1.7444	0.7764	1.1734
	Med. Noise	K-Means	0.0000	0.4160	8.9872	1.7262	1.6914	0.9932	3.5475	1.4737	1.4736
		MS	0.0000	0.0000	0.0000	1.5357	1.7468	0.5896	4.1002	1.5987	0.7584
		DBSCAN	0.0000	0.0000	0.0000	2.4889	1.8896	0.5896	5.1238	2.2243	0.7968
	High Noise	K-Means	0.0000	0.7180	8.5525	1.1045	0.9877	1.0000	4.4779	2.1843	2.3400
		MS	0.0000	0.0000	0.0000	2.3097	1.0001	1.3506	6.9021	2.4040	2.3599
		DBSCAN	0.0000	0.0000	0.0000	2.9980	1.4271	1.4785	6.6667	1.7491	1.7509

Table 4.7: Algorithm Scores

4.2 Selection Map for Problem Space

Analyzing the meta-data produces a selection map that can be used to determine which algorithm should be run at based on the current scenario factors. The selection map was created using the score values computed from the meta-data. The initial selection map is shown in Table 4.8.

Best Algorithm		Single Target			Multi Target Diverse			Multi Target Close Proximity		
Measurements		Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP
AOA Only	Low Noise	DBSCAN	DBSCAN	DBSCAN	MeanShift	MeanShift	MeanShift	K-Means	DBSCAN	DBSCAN
	Med. Noise	DBSCAN	DBSCAN	DBSCAN	MeanShift	MeanShift	DBSCAN	K-Means	K-Means	K-Means
	High Noise	DBSCAN	DBSCAN	DBSCAN	K-Means	DBSCAN	DBSCAN	K-Means	K-Means	K-Means
TDOA Only	Low Noise	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	MeanShift	MeanShift	DBSCAN
	Med. Noise	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	K-Means	DBSCAN	DBSCAN
	High Noise	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	K-Means	DBSCAN	DBSCAN
TOA Only	Low Noise	DBSCAN	DBSCAN	DBSCAN	K-Means	MeanShift	MeanShift	K-Means	MeanShift	DBSCAN
	Med. Noise	DBSCAN	DBSCAN	DBSCAN	K-Means	DBSCAN	MeanShift	MeanShift	MeanShift	MeanShift
	High Noise	DBSCAN	DBSCAN	DBSCAN	K-Means	K-Means	DBSCAN	K-Means	MeanShift	DBSCAN
AOA + TDOA	Low Noise	DBSCAN	DBSCAN	DBSCAN	MeanShift	MeanShift	MeanShift	K-Means	K-Means	MeanShift
	Med. Noise	DBSCAN	DBSCAN	DBSCAN	MeanShift	MeanShift	MeanShift	K-Means	MeanShift	MeanShift
	High Noise	K-Means	K-Means	DBSCAN	K-Means	MeanShift	DBSCAN	K-Means	K-Means	DBSCAN
AOA + TOA	Low Noise	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	K-Means	K-Means	DBSCAN
	Med. Noise	DBSCAN	DBSCAN	DBSCAN	MeanShift	K-Means	MeanShift	K-Means	K-Means	MeanShift
	High Noise	DBSCAN	DBSCAN	DBSCAN	K-Means	K-Means	K-Means	K-Means	DBSCAN	DBSCAN

Table 4.8: Initial Selection Map From Algorithm Performance Scores

After initial review some changes were made to the selection map where performance across algorithms is comparable, so the algorithm selection was changed to create simpler mappings with more consistent selections for the different subspaces. The updated map is shown in Table 4.9.

Best Algorithm		Single Target			Multi Target Diverse			Multi Target Close Proximity		
Measurements		Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP
AOA Only	Low Noise	DBSCAN	DBSCAN	DBSCAN	MeanShift	MeanShift	MeanShift	MeanShift	DBSCAN	DBSCAN
	Med. Noise	DBSCAN	DBSCAN	DBSCAN	MeanShift	MeanShift	DBSCAN	K-Means	K-Means	K-Means
	High Noise	DBSCAN	DBSCAN	DBSCAN	K-Means	DBSCAN	DBSCAN	K-Means	K-Means	K-Means
TDOA Only	Low Noise	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	MeanShift	MeanShift	DBSCAN
	Med. Noise	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	K-Means	DBSCAN	DBSCAN
	High Noise	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	K-Means	DBSCAN	DBSCAN
TOA Only	Low Noise	DBSCAN	DBSCAN	DBSCAN	K-Means	MeanShift	MeanShift	K-Means	MeanShift	DBSCAN
	Med. Noise	DBSCAN	DBSCAN	DBSCAN	K-Means	DBSCAN	MeanShift	MeanShift	MeanShift	MeanShift
	High Noise	DBSCAN	DBSCAN	DBSCAN	K-Means	K-Means	DBSCAN	K-Means	MeanShift	DBSCAN
AOA + TDOA	Low Noise	DBSCAN	DBSCAN	DBSCAN	MeanShift	MeanShift	MeanShift	K-Means	K-Means	MeanShift
	Med. Noise	DBSCAN	DBSCAN	DBSCAN	MeanShift	MeanShift	MeanShift	MeanShift	MeanShift	MeanShift
	High Noise	DBSCAN	DBSCAN	DBSCAN	K-Means	MeanShift	DBSCAN	MeanShift	K-Means	DBSCAN
AOA + TOA	Low Noise	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	DBSCAN	K-Means	DBSCAN	DBSCAN
	Med. Noise	DBSCAN	DBSCAN	DBSCAN	MeanShift	MeanShift	MeanShift	K-Means	K-Means	DBSCAN
	High Noise	DBSCAN	DBSCAN	DBSCAN	K-Means	K-Means	K-Means	K-Means	DBSCAN	DBSCAN

Table 4.9: Final Selection Map From Algorithm Performance Scores and Simplification

One of the largest and most notable adjustments made is changing all single target sets to use DBSCAN because it is the least susceptible to split clusters into multiple sections and greatly simplifies the logic structure.

4.3 Adaptive Clustering Engine Implementation

The Adaptive Clustering engine slots into the particle filter architecture described previously as a precursor step to the state estimation. ACE evaluates the measurements received so far in the particle filter's operation and provides a clustering algorithm recommendation to the state estimator. A visual example of the updated data flow path is provided in Figure 4.2.

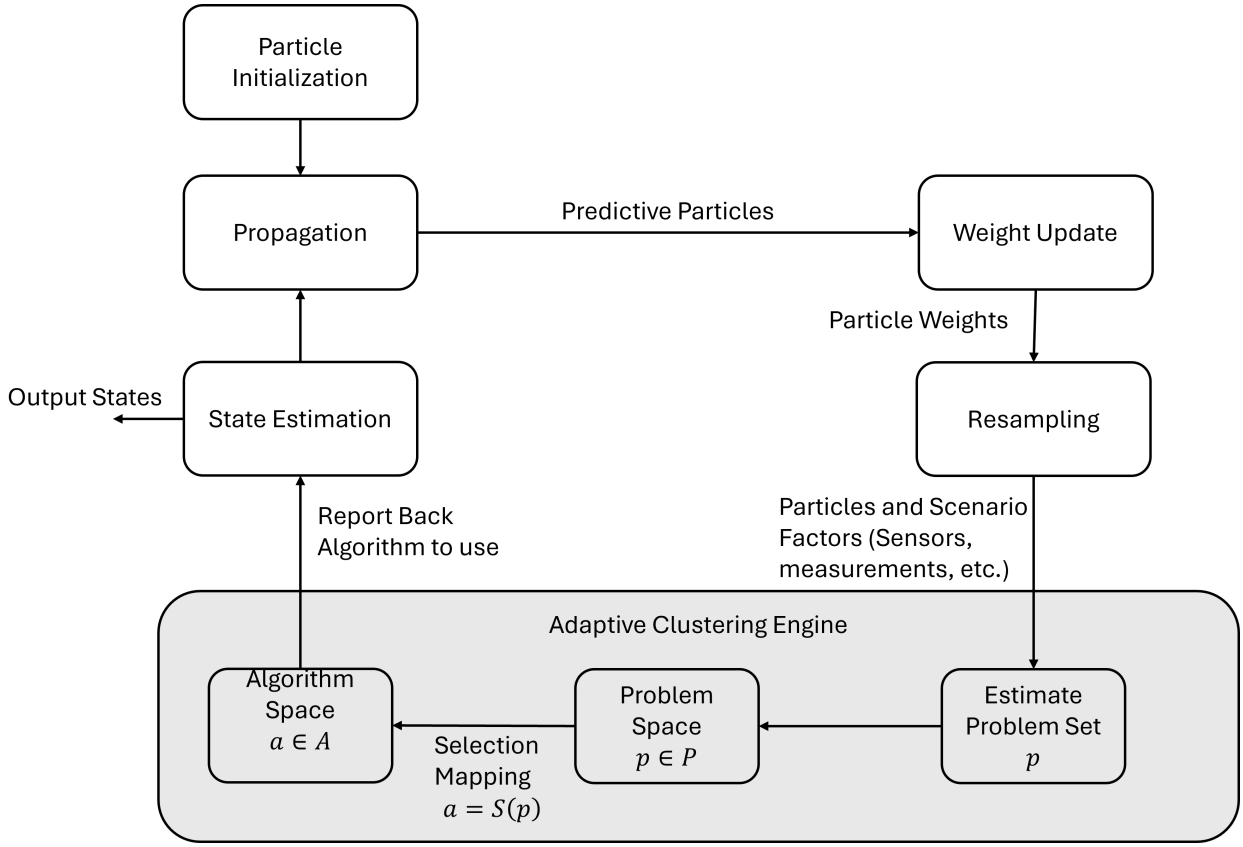

Figure 4.2: Updated Particle Filter with ACE Included

4.3.1 Problem Space Estimation

In order to properly select which algorithm to use from the selection map, the scenario characteristics themselves need to be estimated. Although some fields are deterministic, others need to be estimated using available information from the filter. One challenge with this approach is the fields are not directly observable either, for example computing DOP requires the target location. The following sections describe how ACE evaluates the sensor measurements and locations to determine which problem space the algorithm is operating in.

Deterministic Fields

The following subspaces are deterministic given some form of previous knowledge before operation. The noise level of the measurements could be preset if the sensors used are known and have defined error distributions that can be rounded to the terms used in Table 3.1. The measurement types used in the estimation would be known or could easily be checked and updated in real time operation as the filter receives measurements. For verification testing, these fields were considered deterministic and did not use an estimation step.

DOP Level Estimation

The components required to produce a DOP value for any given localization problem require an understanding of the target location as part of the measurement matrix equations. Since the position of a target has not yet been estimated, it is not possible to calculate the DOP in real time. Instead, relative sensor measurements can be used to create a rough idea of the DOP level that can help guide ACE in selecting the correct entry of the selection map.

This process is completed differently depending on the types of measurements. For each scenario, a reference position x_{ref} is calculated that is the average of all sensor locations using Equation 4.1.

$$x_{ref} = \frac{\sum_{i=1}^{M}(s_{k,x}^{i} + s_{k,y}^{i})}{M} \tag{4.1}$$

Where s is the sensor locations comprised of a x and y component and M is the total number of active sensors. Depending on how the measurements interact or relate to that relative position or measurements from other sensors can be used to infer the DOP level.

Default Case

The default approach to estimating the DOP level is to calculate the average range from the reference position to each particle. This method is measurement agnostic and can be applied for any measurement type, but it is also computationally expensive to interact with each particle. Therefore, it is only used when necessary and other measurement based methods are prioritized. The average particle range (APR) is computed as shown in Equation 4.2 and then the DOP level is assigned based on the piecewise function of Equation 4.3.

$$APR = \frac{\sum_{j=1}^{N}\sqrt{(x_{k,x}^{j} - x_{ref,x})^2 + (x_{k,y}^{j} - x_{ref,y})^2}}{N} \tag{4.2}$$

$$DOP = \begin{cases} 0 & \text{if } APR \leq 1.15 \\ 1 & \text{if } 1.15 < APR \leq 2.5 \\ 2 & \text{if } APR > 2.5 \end{cases} \tag{4.3}$$

A visual representation of the default case is shown in Figure 4.3.

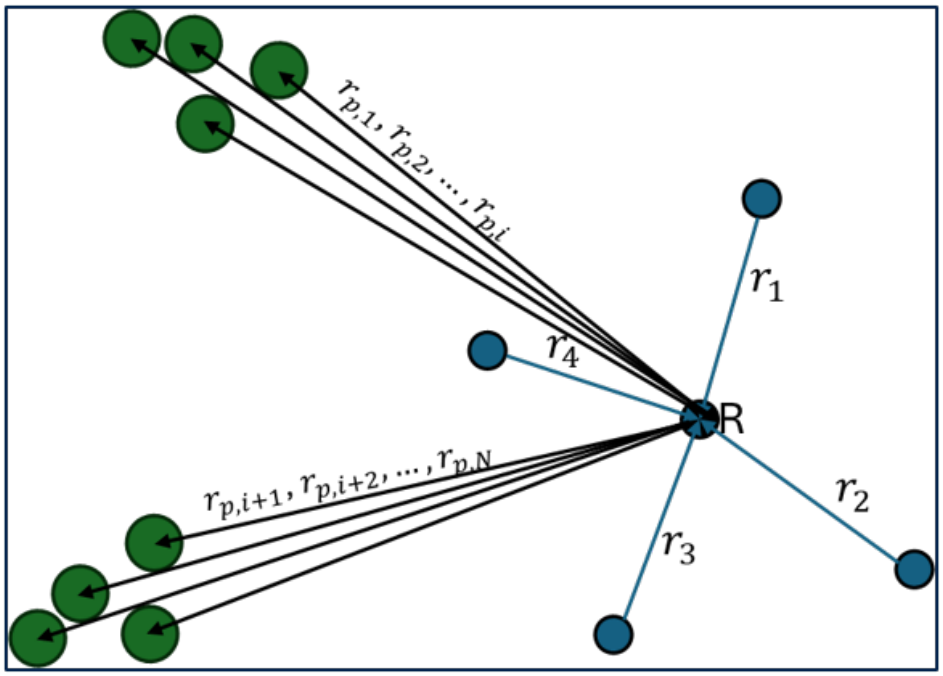

Figure 4.3: Default Case For DOP Estimation

TOA Case

For TOA measurements, the DOP level is calculated by comparing the average measurement ranges to the average range from each sensor to the average sensor position. The rationale behind this method is that if the sensors are surrounding the targets, some ranges will be less than the range to the reference position and some will be higher, resulting in an average around one. An upper limit value of 1.3 is set for the measurement to sensor range ratio for the low DOP level. If the DOP level is higher, some measurements that are on the side closer to the target will be lower, but the sensors further away will increase the average range. The average measurement to sensor range ratio for the medium DOP level is between 1.3 and 2.5. Anything above 2.5 is considered to be a high DOP level. A visual reference of the senor ranges compared to measurement ranges is shown for a medium DOP case shown in Figure 4.4.

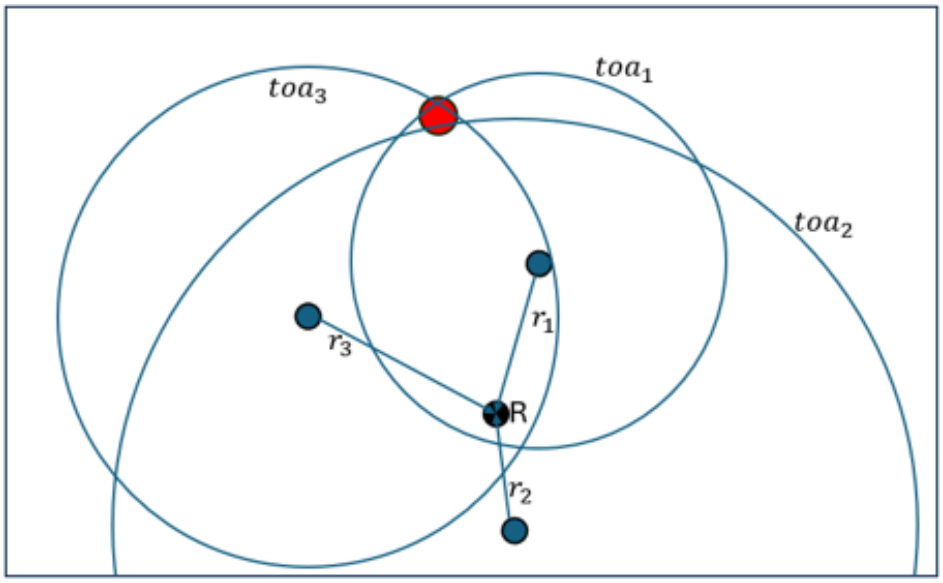

Figure 4.4: TOA Case For DOP Estimation

AOA Case

The DOP level for the AOA only case is estimated using the relative angle differences between each sensor. The simplest case to demonstrate the process is with a single target. The AOA measurement produced from two sensors intersect at or near the target depending on measurement noise. The two lines of the AOA measurements produce an angle value with the vertex being the AOA intersection point. The relative angle measured from the intersecting point is directly related to the ratio of DOP values that are used to define the different DOP levels. Using parallel line theorem the relative angle at the intersection location is the same as the difference in the two measurement angles. This relation allows the relative intersection angle to be determined without finding the intersection location. A visual interpretation of this process is shown in Figure 4.5.

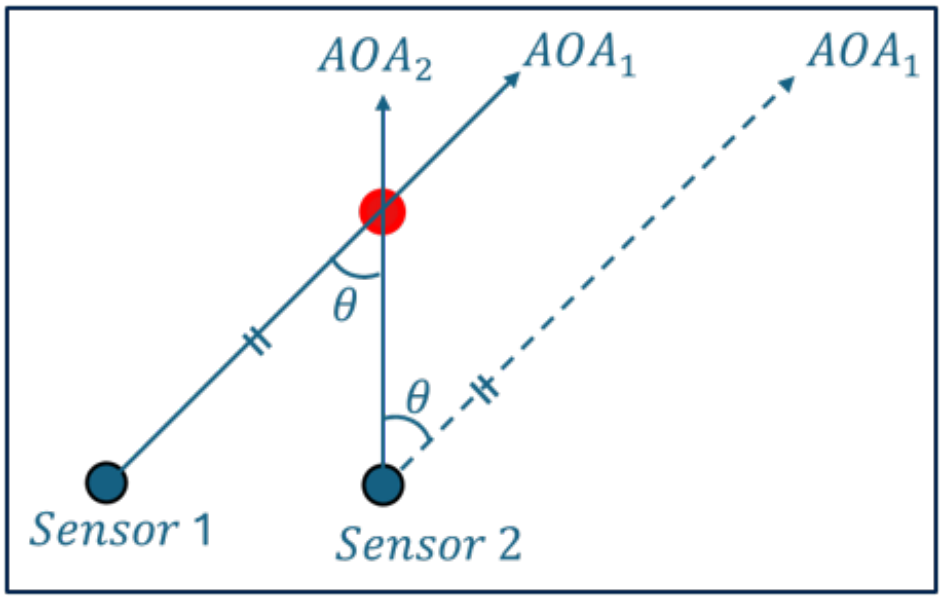

Figure 4.5: Relative Angle

The optimal AOA geometry is when measurements intersect at a relative angle of 90°. Any values higher than 90 degrees indicates the target is between the two sensors, so it is considered a low DOP level. As relative angles decrease the target is presumably further away and thus the DOP ratio starts to increase into a longer ellipse. Starting at 90°and moving towards zero the ellipse ratio slowly increases until a value of 2.5 near the 45 degree relative angle value. This region creates the medium DOP level region, relative angles between 45°and 90°. Any relative angles less than 45°considered the high DOP level.

This method of using relative angles can be expanded to the multiple target case by mapping out measurements from two independent sensors on top of each other and averaging out measurement offsets. This process first requires creating a map of an individual sensor measurements and calculating the spacing between each AOA measurement. The spaces between measurements are associated across sensors to see if there is correlation. If the targets are between the two sensors the measurement maps will either have measurement spacings in alternate orders or only be a small overall measurement window resulting in a large range of angles with no measurements. If the large ranges of no measurements are in

opposite directions this implies a low DOP level. If the map spacings are near identical but slightly rotated, the majority of targets are well outside of the sensor group and a higher DOP level is selected. Example AOA measurement mappings are shown in Figure 4.6.

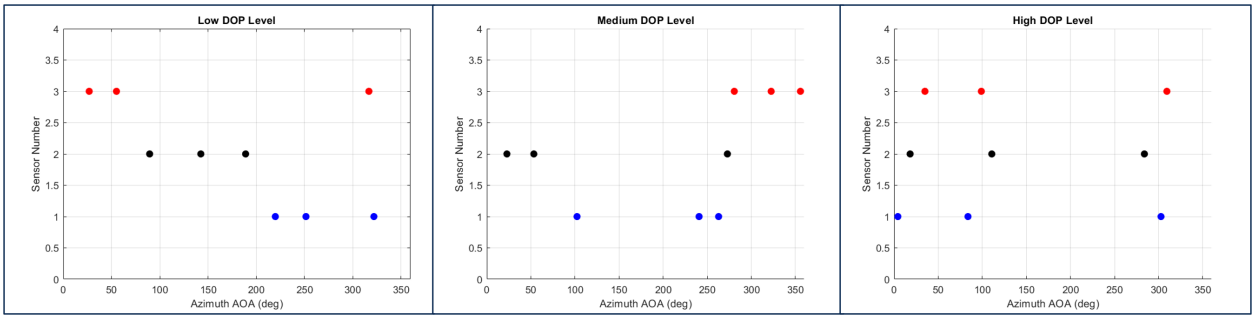

Figure 4.6: Comparison of AOA Measurement Maps for Different DOP Levels

Hybrid Case

For the hybrid case of AOA and TDOA, only the AOA method is used to avoid running the default case that requires performing an operation on all particles. With the AOA and TOA case, the DOP is estimated independently. If there is a disagreement in the results the higher DOP Level is selected.

4.3.2 Target Quantity Estimation

ACE calculates the number of independent measurements received for each set of detections to get an average number of measurements per sensor. This average for the detection period is then added to a moving mean over a longer time period. The use of a moving mean helps to smooth out the value over time to account for missed detections or false detections without rapidly switching states. If the average number of measurements produced per each sensor is greater than 1.5 the scenario is considered to have multiple targets. The added buffer of half a measurement is to account for situations where a sensor could have stray measurements that increase the number. Similarly, using an average across all sensors

helps from underestimating the number of targets in the instance that a sensor has a missed detection either from a signal processing reason or lack of visibility of a target.

Estimating the target quantity is only used to guide algorithm selection in the selection map. Meaning, if single target is estimated incorrectly, the only downside would be a non-optimal clustering algorithm is selected. The clustering algorithms in use are not limited to estimating only one target if the target quantity is inaccurate and estimates a single target.

4.3.3 Target Proximity Estimation

Target proximity checks are evaluated on a single sensor basis by checking how different the measurements of each type are from each other. The proximity estimation step uses an altered version of innovation filtering to determine the proximity of the measurements. This method was selected using the understanding that particle distributions reflect the uncertainty of sensor measurements and if sensor measurements overlap with their uncertainties, the respective particle distributions will also overlap.

First, each measurement from a single sensor at a single detection time is evaluated by subtracting all the other measurements to get residual values. These residual values represent the offset from the central measurement being evaluated. The residuals are then divided by the square root of the measurement uncertainty to create innovation values. The innovation represents how many standard deviations apart the different measurements are for a given uncertainty. If any of the innovations are less than three, meaning that two measurements are within three standard deviations, the sensors are considered to be seeing clustered targets [33]. If majority of sensors are producing measurements that are close to each other, the scenario is considered to be close proximity.

4.4 Chapter Conclusion

Chapter 4 combines takes concepts covered in Chapter 3 and creates a functional algorithm. The two staged approach of ACE is introduced to cover the training side to create

a selection map and the implementation side where the selection map is used to select clustering algorithms during filter operation.

ACE is trained through the creation of meta-data, simulated data that encompasses the entire problem space. Simulated particle filter runs are clustered using each algorithm of the algorithm space and were scored using the process outlined in Chapter 3. The resultant meta-data is then used to create a selection map that can be used when the filter is running.

The implementation of ACE and how it fits into the overall particle filter architecture in Section 4.3. The sub-processes ACE uses to estimate which problem set the filter is operating in are defined. The estimated fields include relative DOP level, single or multiple targets present, and if the multiple targets are in close proximity to one another. ACE uses these estimated parameters to pull the clustering algorithm to be used from the selection map produced using the meta-data.

Chapter 5

Validation Testing of Adaptive Clustering Engine

This chapter covers the testing performed on the various components of the Adaptive Clustering Engine. First, the simulation setup is reviewed for comparison to how meta-data was created. Secondly, the scenario factor estimation of ACE is evaluated across the entire problem space. Lastly, the cumulative clustering performance with ACE is compared to individual clustering algorithms.

5.1 Simulation Validation Testing

ACE was validated the same three ring simulation setup used meta-data creation with some alterations to add diversity between runs. The range in number of sensors and number of targets was increased to encourage more variation in geometries. Stray measurements or false detections were added randomly to prevent knowing the number of targets. Detection rate for targets was decreased so some sensors would miss targets at different times throughout simulation runs. The DOP level, target quantity, and target proximity were not known and had to be estimated. The deterministic measurement characteristics were also tested at all incremental levels.

5.2 Scenario Factor Estimation Results

This section covers performance metrics of the scenario factor estimation portion of the Adaptive Clustering Engine. The estimated values are the DOP level on a scale from 0-2, whether or not there are multiple targets, and if there are multiple targets were they

within close proximity. Measurement types and measurement noise values are considered deterministic, as described in 4.3.1, and therefore not estimated.

5.2.1 DOP Level Estimation Results

The accuracy of DOP level estimation was tracked as it is one of the main influences to algorithm selection. Overall estimation results show promising results across the entire problem space with very few errors. Performance is shown with a three by three histogram where the rows indicate the estimated DOP level, and the various columns represent the different noise levels tested. The noise levels are separate to show the correlation between the estimation error and the measurement noise if there is correlation. The true DOP levels are the same for each subplot and are in a grid so the leftmost column of subplots all have a low DOP level and it increases each column to the right. The rows of the outer grid represent the three target quantity based fields. Lastly, there is a separate set of results for each measurement type/pairing. Result plots start with the AOA only scenarios in Figure 5.1.

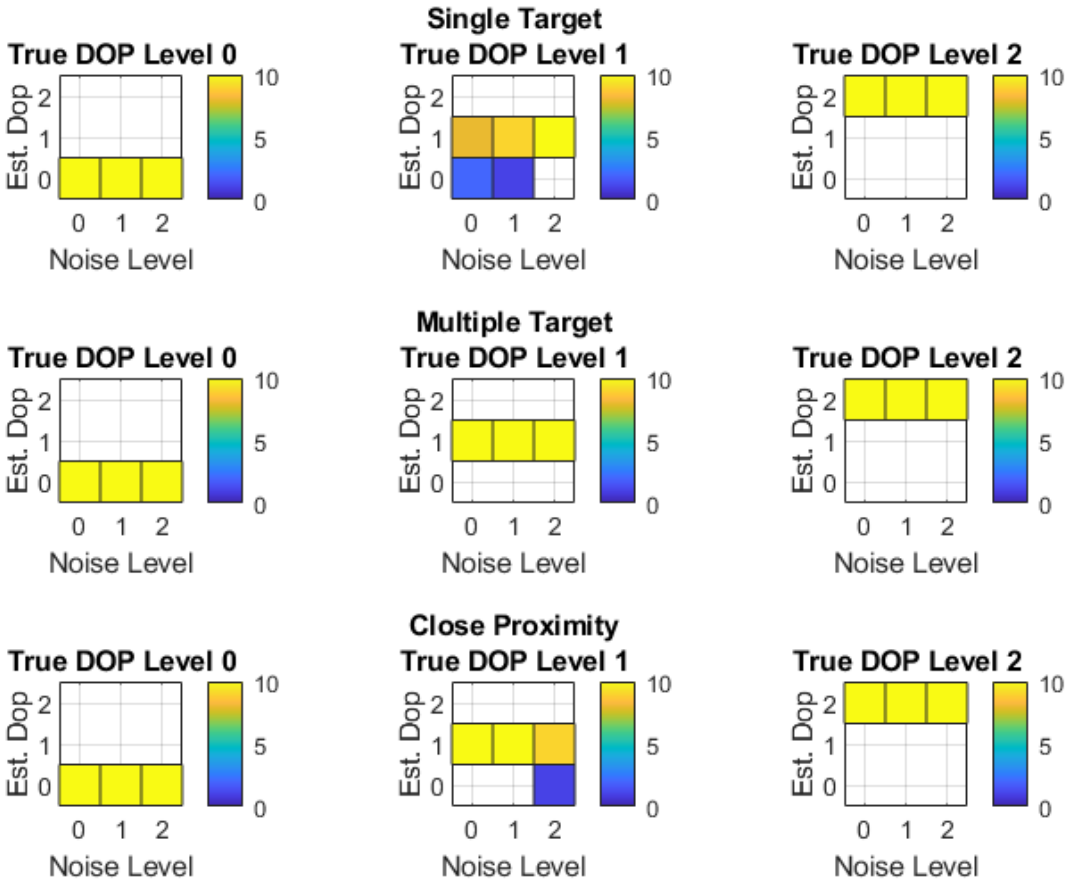

Figure 5.1: AOA Only DOP Estimation

For AOA only scenarios, the only errors are with a few of the medium DOP level estimates underestimating down to a low DOP level. For AOA measurements the transition in clustering performance and based on the selection map the algorithm choice for all of the missed estimates would result in using the same algorithm. For AOA measurements only, seven of the nine subspaces have the same algorithm selection between low and medium DOP levels. For all measurement types, the same is true for 33 of the 45 subspaces.

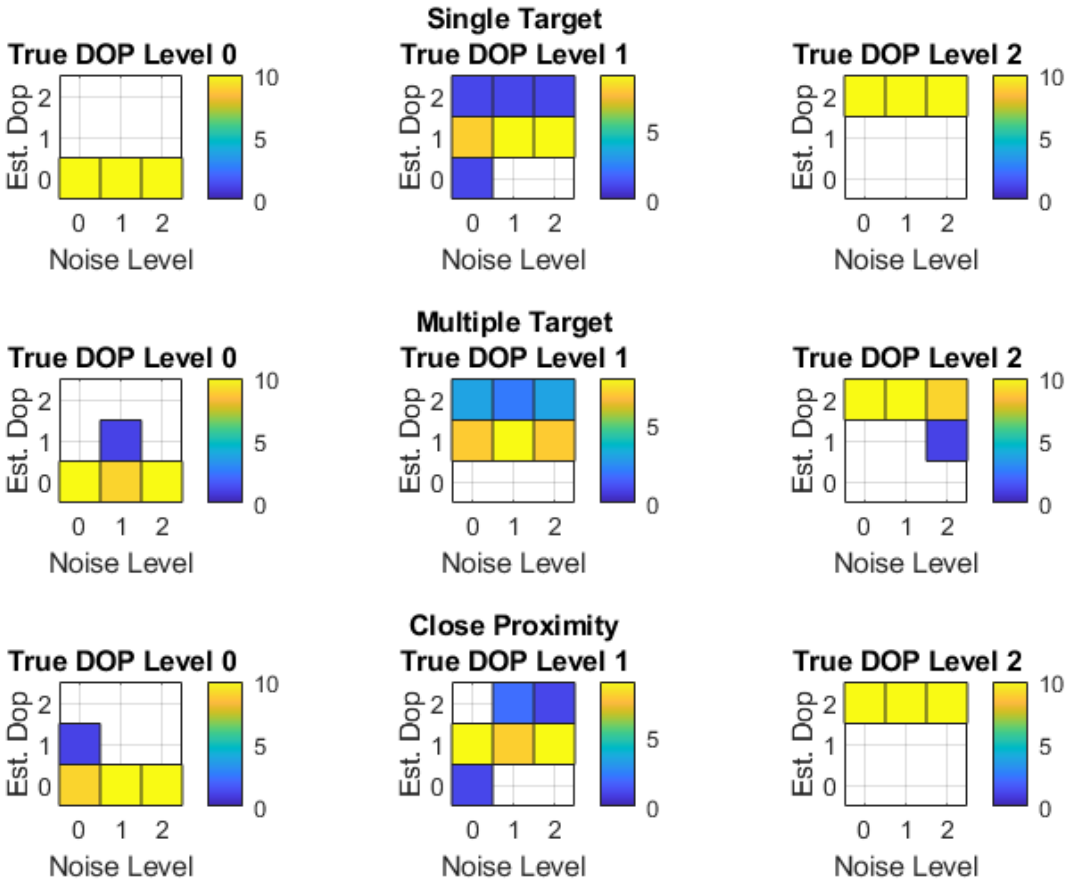

Figure 5.2: TDOA Only DOP Estimation

TDOA only measurements have an opposite result where it more commonly estimates a higher DOP level. Eight of the nine subspaces have the same clustering algorithm between medium and high DOP levels. The close proximity medium DOP level and low measurement noise subspace is the only subspace where the medium and high DOP levels are different, but the inaccurate estimates for this subspace under estimated DOP which has the same mean shift entry for the selection map. This results in the optimal algorithm being selected for each run regardless of the errors.

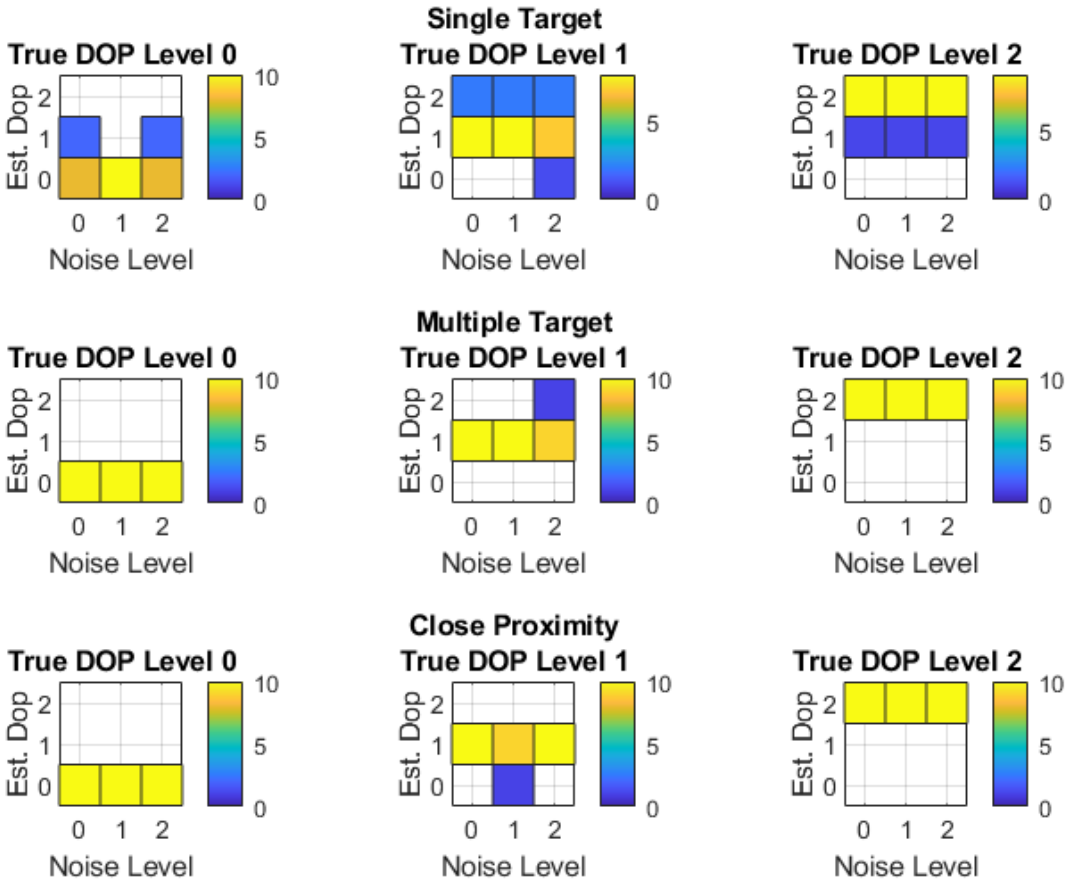

Figure 5.3: TOA Only DOP Estimation

The errors for the TOA only case in Figure 5.3 show that most errors are in the single target case. All of these errors result in the same selection of the clustering algorithm and do not affect localization performance. The multiple target error in the central table results in a change from K-Means to DBSCAN. The error in the close proximity row results in selecting the same mean shift algorithm resulting in only one error that changes algorithm selection.

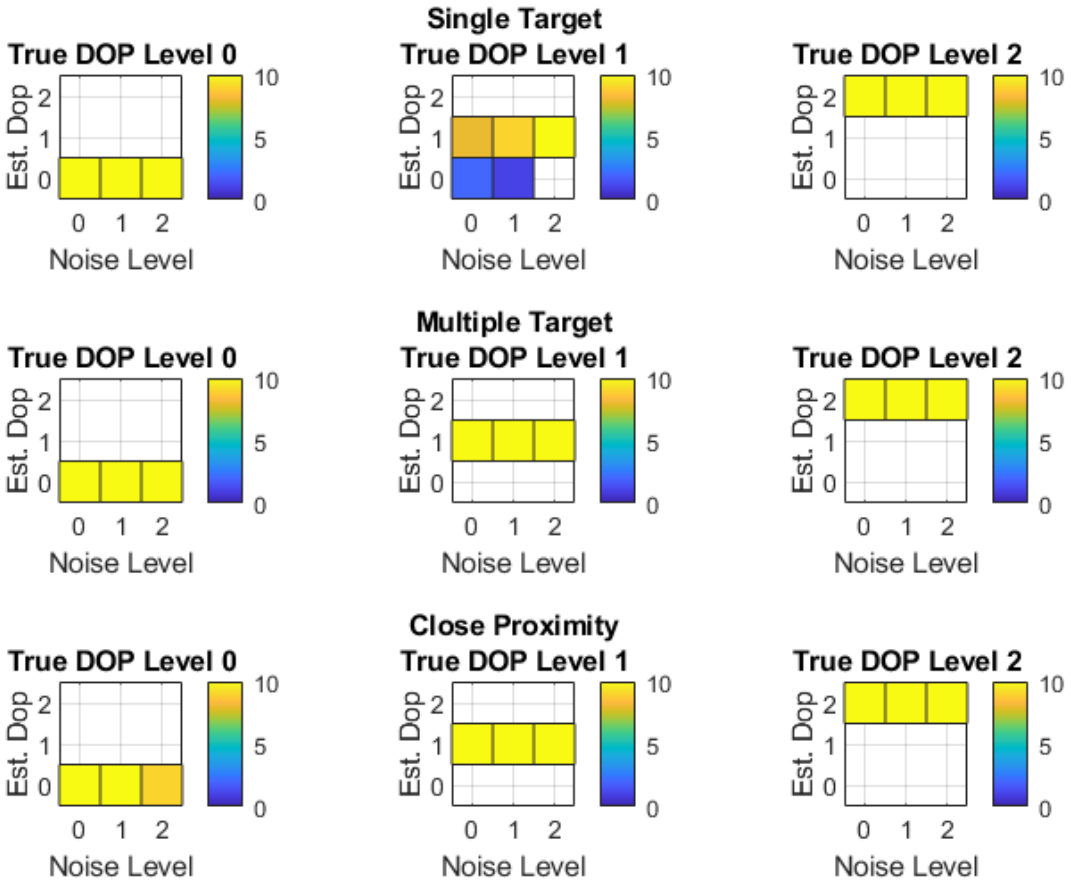

Figure 5.4: AOA and TDOA DOP Estimation

For the AOA and TDOA subspace every DOP estimate is correct except for three runs, all falling under the single target and medium DOP level subspace. As mentioned in Section 4.2, all single target selection maps are set to use DBSCAN, so the errors shown in Figure 5.4 do not alter the algorithm selection.

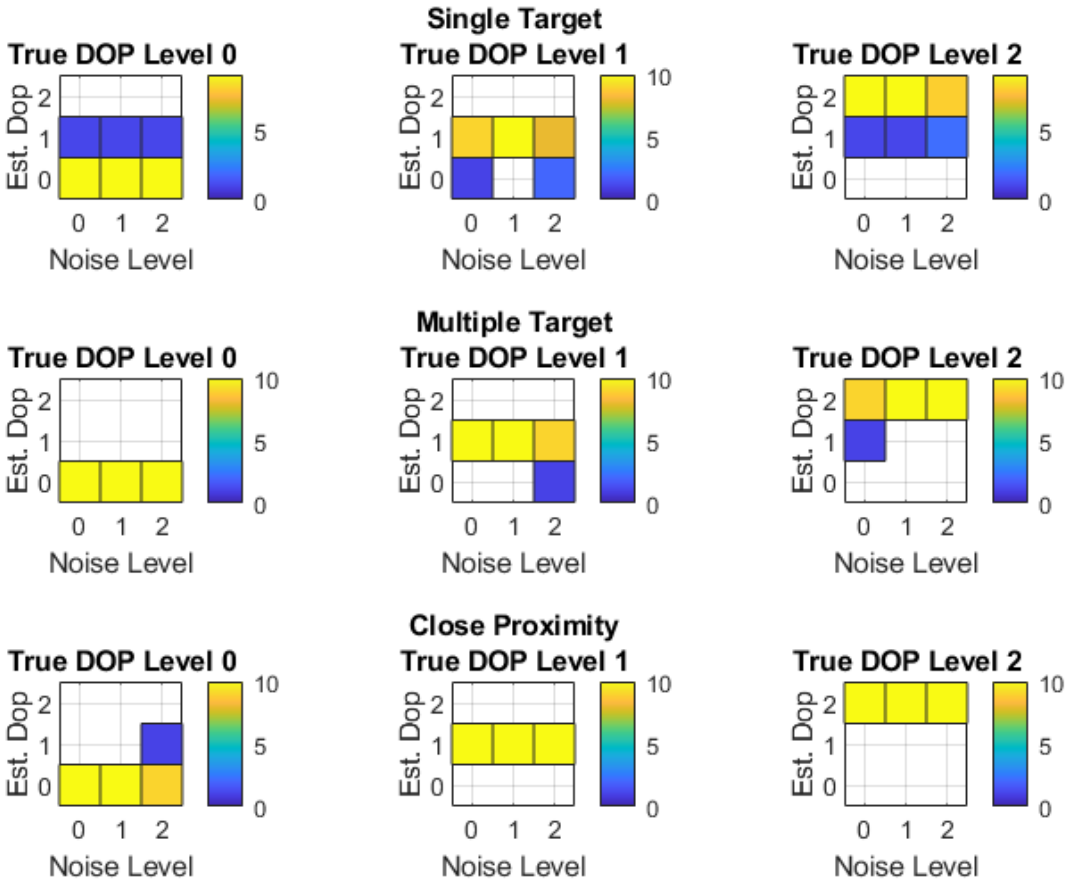

Figure 5.5: AOA and TOA Only DOP Estimation

The single target estimates make up the majority of the DOP estimation errors. These errors in DOP estimation do not effect clustering algorithm selection as all single targets use DBSCAN. The AOA and TOA multiple target case is the only subspaces where DOP level does not affect algorithm selection, but measurement noise level results in three different algorithm selections, implicating DOP estimation errors does not change algorithm selection. The one error in the close proximity cases does result in a change from using K-Means to DBSCAN which has a tendency of underestimate total number of clusters with low DOP close proximity targets.

5.2.2 Target Quantity Estimation Results

The target quantity is estimated to a binary resolution of single target versus multiple targets. Confusion matrices are used to show estimation accuracy. A confusion matrix summarizes binary selections into four components: true positives, true negatives, false positives, and false negative. True positives and true negatives are the percent of runs that have the correct results split based on what the true value is. For the target quantity case, a positive is the multiple target case and a negative is a single target. The layout of the confusion matrix is shown in 5.1.

	Estimated True	Estimated False
Actual True	True Positive	False Negative
Actual False	False Positive	True Negative

Table 5.1: Example Confusion Matrix

Table 5.2 shows individual confusion matrices for each subspace using the same outline outlined by Table 5.1.

Table 5.2 shows that the accuracy for estimating if there are multiple targets is above 90% except for three instances that result in an average for all subspaces of 88% true positive rate. The true negative rate has an average of 89% with the lowest rate of 70% true negative rate. Additionally, results are better in the hybrid cases where more measurements are averaged. For reference, the detection rate for targets is 80% and a stray measurement rate of 15%. As the detection rate decreases, it becomes more likely that a scenario with fewer targets is falsely marked as a single target.

Further investigation reveals that all false negative runs are scenarios in which only two targets were present. In total there were 48 runs where this happened, 44 of which still estimated two targets in the clustering step, two estimated one target, and two estimated three targets. Of the false negative scenarios, there is a 91.67% accuracy in the number of targets estimated after the clustering step.

Single/Multi		Low DOP		Med. DOP		High DOP	
AOA Only	Low Noise	0.8	0.2	0.95	0.05	0.95	0.05
		0	1	0.1	0.9	0	1
	Med. Noise	0.9	0.1	0.9	0.1	0.9	0.1
		0	1	0	1	0	1
	High Noise	0.95	0.05	0.85	0.15	0.85	0.15
		0.2	0.8	0	1	0	1
TDOA Only	Low Noise	0.9	0.1	0.95	0.05	0.95	0.05
		0.3	0.7	0.2	0.8	0	1
	Med. Noise	0.95	0.05	0.95	0.05	0.95	0.05
		0.2	0.8	0.1	0.9	0.3	0.7
	High Noise	0.95	0.05	0.95	0.05	0.95	0.05
		0.1	0.9	0	1	0.1	0.9
TOA Only	Low Noise	0.95	0.05	0.95	0.05	0.95	0.05
		0	1	0	1	0	1
	Med. Noise	1	0	0.9	0.1	0.95	0.05
		0	1	0	1	0	1
	High Noise	1	0	0.9	0.1	0.95	0.05
		0	1	0.1	0.9	0	1
AOA + TDOA	Low Noise	0.9	0.1	1	0	1	0
		0	1	0	1	0	1
	Med. Noise	0.95	0.05	1	0	0.95	0.05
		0	1	0	1	0	1
	High Noise	1	0	0.95	0.05	0.95	0.05
		0	1	0	1	0	1
AOA + TOA	Low Noise	1	0	0.95	0.05	1	0
		0	1	0	1	0	1
	Med. Noise	0.95	0.05	1	0	0.95	0.05
		0	1	0	1	0	1
	High Noise	1	0	1	0	0.95	0.05
		0	1	0	1	0	1

Table 5.2: Multiple Versus Single Target Confusion Matrix

5.2.3 Target Proximity Estimation

The target proximity estimator results in a binary value of whether the targets are clustered or not. The results for target proximity estimation are reported as confusion matrices using the same formatting outline in Section 5.2.2. True positives are cases where the estimator predicts that the targets are close in proximity.

Close Proximity		Low DOP		Med. DOP		High DOP	
AOA Only	Low Noise	0.2	0.8	0	1	0.1	0.9
		0.1	0.9	0.1	0.9	0.1	0.9
	Med. Noise	0.3	0.7	0.3	0.7	0.2	0.8
		0.3	0.7	0.1	0.9	0.2	0.8
	High Noise	0.4	0.6	0.2	0.8	0.1	0.9
		0.2	0.8	0.1	0.9	0.4	0.6
TDOA Only	Low Noise	0.3	0.7	0.3	0.7	0.3	0.7
		0	1	0	1	0	1
	Med. Noise	0.8	0.2	0.5	0.5	0.3	0.7
		0	1	0	1	0	1
	High Noise	0.6	0.4	0.6	0.4	0.6	0.4
		0	1	0	1	0	1
TOA Only	Low Noise	0.3	0.7	0.2	0.8	0.2	0.8
		0.4	0.6	0	1	0	1
	Med. Noise	0.3	0.7	0.3	0.7	0.2	0.8
		0.6	0.4	0.4	0.6	0.2	0.8
	High Noise	0.7	0.3	0.7	0.3	0.2	0.8
		0.6	0.4	0.4	0.6	0.1	0.9
AOA + TDOA	Low Noise	0.2	0.8	0	1	0.3	0.7
		0	1	0	1	0	1
	Med. Noise	0.2	0.8	0.2	0.8	0.2	0.8
		0.4	0.6	0.1	0.9	0	1
	High Noise	0.3	0.7	0	1	0.5	0.5
		0.5	0.5	0	1	0.2	0.8
AOA + TOA	Low Noise	0.4	0.6	0.3	0.7	0.2	0.8
		0.1	0.9	0.3	0.7	0.1	0.9
	Med. Noise	0.7	0.3	0.2	0.8	0.3	0.7
		0.4	0.6	0.2	0.8	0.1	0.9
	High Noise	1	0	0.3	0.7	0.4	0.6
		0.5	0.5	0.6	0.4	0.2	0.8

Table 5.3: Close Target Proximity Confusion Matrix

Table 5.3 shows a propensity to estimate targets that are not within close proximity with only a true positive rate of 29.4%. This poor result suggests a divergence from how close proximity scenarios are generated in the simulation versus the innovation filtering approach used to check proximity.

5.2.4 Combined Performance

Combining results of the three scenario factor estimation results in error statistics for how many times the correct problem set was selected correctly. The first combined statistic is the percentage of runs that selected the correct problem set shown in Table

Correct Problem Set		Single Target			Multi Target Diverse			Multi Target Close Proximity		
Validation Runs		Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP
AOA Only	Low Noise	100%	70%	100%	70%	90%	90%	20%	0%	10%
	Med. Noise	100%	90%	100%	70%	80%	70%	30%	30%	20%
	High Noise	80%	100%	100%	70%	80%	40%	40%	20%	10%
TDOA Only	Low Noise	70%	60%	100%	100%	60%	100%	30%	30%	30%
	Med. Noise	80%	90%	70%	90%	70%	100%	80%	30%	30%
	High Noise	90%	90%	90%	90%	70%	90%	60%	50%	60%
TOA Only	Low Noise	80%	80%	90%	50%	100%	100%	30%	20%	20%
	Med. Noise	100%	80%	90%	40%	40%	70%	30%	30%	20%
	High Noise	80%	70%	90%	40%	30%	80%	70%	70%	20%
AOA + TDOA	Low Noise	100%	80%	100%	90%	100%	100%	20%	0%	30%
	Med. Noise	100%	90%	100%	60%	90%	90%	20%	20%	20%
	High Noise	100%	100%	100%	50%	100%	70%	30%	0%	50%
AOA + TOA	Low Noise	90%	90%	90%	90%	60%	80%	40%	30%	20%
	Med. Noise	90%	100%	90%	50%	80%	80%	70%	20%	30%
	High Noise	90%	80%	80%	50%	40%	80%	90%	30%	40%

Table 5.4: Total Estimations of Each Problem Set

The next result obtained by combining all three estimation functions is the total number of times each problem set was estimated. Table 5.5 shows the subspaces in which ACE more commonly estimates. Most errors as expected are related to the close proximity case causing the average number of normal multiple target runs to be higher.

5.3 Clustering Performance Results

The complete ACE process is tested using the same process as the meta-data creation but uses the estimated scenario parameters with the selection map to decide algorithm instead of the true parameters. An important note before analyzing these results is that ACE only selects which algorithm to use based on estimated scenario factors. The actual

Total Runs		Single Target			Multi Target Diverse			Multi Target Close Proximity		
Validation Runs		Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP	Low DOP	Med. DOP	High DOP
AOA Only	Low Noise	16	8	11	13	19	17	3	1	2
	Med. Noise	13	11	12	12	14	14	6	4	4
	High Noise	10	12	13	15	14	12	6	3	5
TDOA Only	Low Noise	10	9	14	28	22	28	3	3	3
	Med. Noise	9	11	10	22	19	34	8	3	4
	High Noise	12	11	11	21	17	30	6	5	7
TOA Only	Low Noise	9	12	12	12	17	17	7	2	2
	Med. Noise	10	11	12	12	10	15	9	7	4
	High Noise	8	12	12	8	6	17	13	11	3
AOA + TDOA	Low Noise	14	8	10	16	20	17	2	0	3
	Med. Noise	12	9	11	13	17	17	6	3	2
	High Noise	10	11	11	11	19	12	8	0	7
AOA + TOA	Low Noise	10	12	9	15	14	16	5	6	3
	Med. Noise	10	12	10	8	16	15	11	4	4
	High Noise	11	11	9	5	11	13	15	9	6

Table 5.5: Total Runs Estimated for Each Problem Set

clustering tuning and performance is not changed by ACE, rather the result performances should be comparable to the best algorithm performances from the meta-data for each set of the problem space.

The performance of each algorithm and ACE is shown in Figure 5.6. The max value is lowered to 150 runs to show the dynamics of the different algorithms even though each has a peak at zero. Zero means that the algorithm created the correct number of clusters.

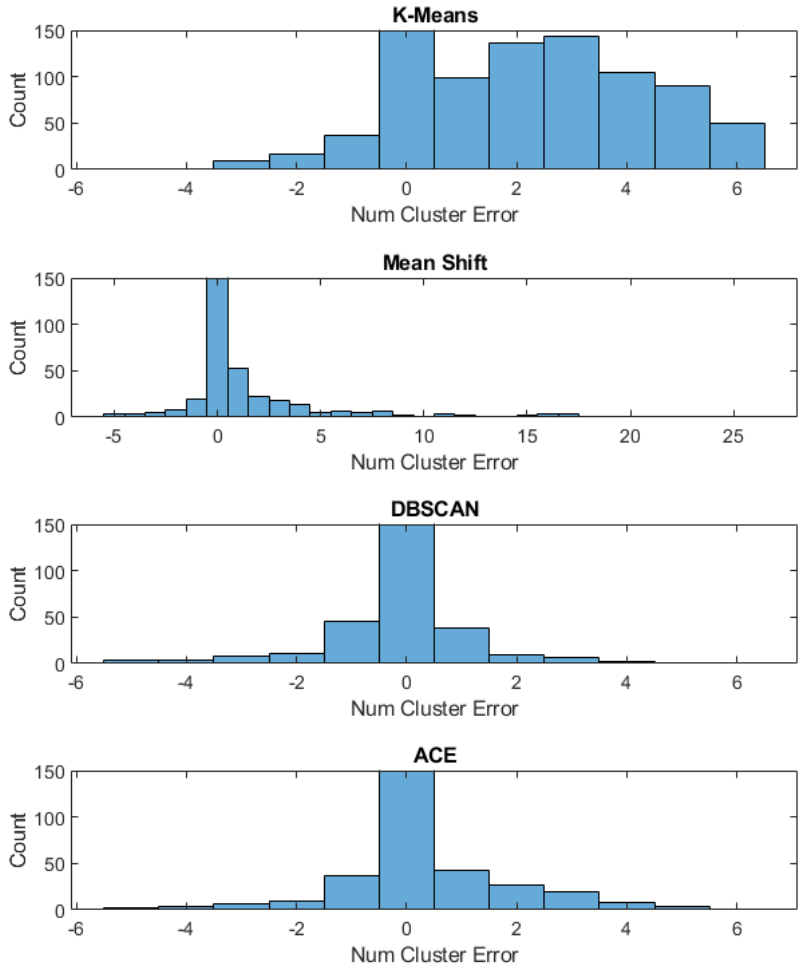

Figure 5.6: Clustering Error By Algorithm

General results that can be inferred from Figure 5.6 include K-Means has a tendency to over estimate number of clusters but is bounded due to the limitation of the range of K values tested. Mean Shift also had runs where it overestimated the number of clusters and in some cases it greatly overestimated the number of targets. However, ACE avoided most of the high error results from Mean Shift and K-Means. Using ACE also lowered the number of runs in which DBSCAN underestimated number of targets resulting in a more centered distribution. In total ACE selected an algorithm that correctly estimated the number of targets 88.09% of runs. The percentages for each algorithm are shown below in Table 5.6

K-Means	Mean Shift	DBSCAN	ACE
49.26%	85.87%	90.38%	88.09%

Table 5.6: Percent of Perfect Runs

In total, 1,352 scenarios were run evenly across the problem space. The number of times that ACE selected each clustering algorithm is shown in Table 5.7.

K-Means	Mean Shift	DBSCAN
209	357	786

Table 5.7: Total Runs ACE Selected Each Algorithm

A direct comparison of ACE performance versus the others is shown in Table 5.8.

	ACE Better	ACE Worse
K-Means	566	33
Mean Shift	108	54
DBSCAN	29	68

Table 5.8: Comparison of Ace Performance To Individual Algorithms

ACE outperformed both K-Means and Mean Shift more times than individual algorithms outperformed ACE. DBSCAN was the only algorithm of the three evaluated that was more accurate more times than ACE. 19 of the 68 times DBSCAN clustered better ACE should have selected DBSCAN but the scenario parameters were incorrectly estimated. An additional 16 runs had DBSCAN performing the best of the three algorithms in contradiction to the meta-data analysis. This happened in total 46 times out of the 1,352 split between Mean Shift with 13 runs and K-Means with 17 runs. 42 runs were times ACE selected K-Means, but performed worse than both Mean Shift and DBSCAN which is consistent with the results shown in Figure 5.6.

Additional investigation shows that in the majority of the runs where DBSCAN performed better, the measurement noise was higher, 66.18%, and the DOP was high, 50%, and the combination of these two parameters constitutes 41.18% of the runs where DBSCAN

outperformed ACE. Nearly half of the scenarios, 47.06%, that performed worse with ACE were isolated to the TOA only measurement set, the next closest is 26.47% for AOA only. This information can be used to further tune or reevaluate the selection map. Of the scenarios where ACE performed better than DBSCAN, 82.76% of the runs were clustered scenarios showing a bias in certain areas of the problem space.

5.4 Chapter Conclusions

In conclusion, the performance of the Adaptive Clustering Engine was validated in two respects. First, the ability to estimate scenario characteristics and identify the problem set was shown to be effective. The area that could benefit from further research is the determination of how clustered targets relate to each other. Validation simulation results showed ACE is stricter than the original sim setup and had a large number of scenarios falsely labeled non-clustered. ACE was also tested using a series of simulation tests in which the result of the selected algorithm was compared with the other two algorithms in the algorithm space. The results of these tests showed that there is a noticeable improvement using ACE compared to the K-Means and Mean Shift algorithms; however, ACE only performed equally or slightly worse than DBSCAN. This poor performance may be linked to the fact that the simulation setup for generating close proximity targets is too lenient with the how close targets are placed. Testing results show that the majority of scenarios in which ACE performs better than DBSCAN only are the close proximity scenarios, so the two issues may share the same root cause but further investigation is required.

The initial results of the simulation testing show that ACE works as a proof of concept and that there are other areas in which performance can be improved.

Chapter 6

Conclusions

The following chapter will present conclusions drawn from the contents of this thesis. The first section covers the scenario characterization estimation process. The second section will discuss the Adaptive Clustering Engine performance and its ability to dynamically select the optimal algorithm to improve the localization performance. The last section will discuss future work that could push the concepts covered in this thesis further.

6.1 Scenario Factor Estimation

The scenario factor estimation is the key component that enables ACE to navigate the problem space. There are three scenario factors that need to be estimated that drive the selection process. The scenario factors are the DOP level, the quantity of targets, and the proximity of the targets to other targets. The estimation capabilities of each factor was analyzed separately. The deterministic factors relating to measurement type and quantity were treated as known variables.

This thesis defines the DOP level using the ratio of the individual components of the two dimensional error ellipse that is produced from the DOP. A higher ratio corresponds to a longer, thinner ellipse and is considered to be a higher DOP level. The DOP level is estimated using measurement relationships between sensors, which is a more computationally efficient approach than evaluating the particle distribution. The overall DOP level estimation had an accuracy well above 80% for all subspaces evaluated. When there was an error, it only moved one level up or down. The majority of error runs resulted in a problem set where the same algorithm would have been selected, so there was no change in the algorithm selection.

The target quantity estimator uses the moving mean in number of measurements produced to determine if there are multiple targets present. The actual number of targets is estimated using the clustering process. The target quantity estimator estimated the correct scenario type for over 88% of validation runs. Of the runs that falsely determined the scenario to be single target each run only had two targets present and a 91.67% accuracy in estimating the correct number of clusters with the incorrect problem set being used to select clustering algorithm.

The close proximity estimator had the worst performance of the scenario estimators favoring to mark scenarios as non-clustered. The tendency to declare scenarios as non close proximity is an area that could be reviewed and developed further. Initial development and testing in this thesis shows the current method is too strict or the simulation method used to create close proximity scenarios is not consistent enough.

6.2 Adaptive Clustering Engine

The Adaptive Clustering Engine (ACE) uses scenario estimation to aid in selecting the optimal clustering algorithm to be used for a given scenario without user input. Results shown in this thesis show the benefits of this approach over the use of a single algorithm for all scenarios. ACE performance is still limited to that of the algorithms provided in the algorithm space but has good coverage for the problem space outlined in this thesis. ACE was evaluated using general parameters allowing for implementation into a broad range of different applications.

6.3 Future Work

There are two main focuses in this thesis, the first is the scenario factor estimation and the second is the ASP framework implementation for cluster algorithm selection. Each area has the opportunity to be improved or expanded through future work.

The first area to revisit in the scenario estimation problem is the close proximity estimation and meta-data creation. This was the area with the greatest room for improvement in terms of estimation performance.

Other areas of future work pertain to expanding out the problem and algorithm spaces to include more measurement types and clustering algorithms. The problem space used in this thesis uses the field of wireless signal localization to guide the measurement selection, but there are other measurement types that could be incorporated such as received power, frequency difference of arrival, and others. Additionally, some of the problem space could be split up so that there are mixed parameters for different targets. The problem space could also be expanded to new dimensions to include different levels of sensor and target dynamics.

For the algorithm space, a select group of clustering algorithms were selected based on experience and researching other sources that have done a similar particle filter approaches. Looking more into the machine-learning field, there is a wide range of clustering algorithms that could be adapted and used in this application.

Bibliography

[1] Rice, John. "The Algorithm Selection Problem." Department of Computer Science Technical Reports, Jan. 1975, https://docs.lib.purdue.edu/cstech/99.

[2] Wolpert, D. H., and W. G. Macready. "No Free Lunch Theorems for Optimization." IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, Apr. 1997, pp. 67–82. DOI.org (Crossref), https://doi.org/10.1109/4235.585893.

[3] Ferrari, Daniel Gomes, and Leandro Nunes De Castro. "Clustering Algorithm Selection by Meta-Learning Systems: A New Distance-Based Problem Characterization and Ranking Combination Methods." Information Sciences, vol. 301, Apr. 2015, pp. 181–94. DOI.org (Crossref), https://doi.org/10.1016/j.ins.2014.12.044.

[4] Smith-Miles, Kate A. "Cross-Disciplinary Perspectives on Meta-Learning for Algorithm Selection." ACM Computing Surveys, vol. 41, no. 1, Jan. 2009, pp. 1–25. DOI.org (Crossref), https://doi.org/10.1145/1456650.1456656.

[5] Vo, Ba-ngu, et al. "Multitarget Tracking." Wiley Encyclopedia of Electrical and Electronics Engineering, edited by John G. Webster, 1st ed., Wiley, 2015, pp. 1–15. DOI.org (Crossref), https://doi.org/10.1002/047134608X.W8275.

[6] Cheng, Yi, et al. "Improved Particle Filter Algorithm for Multi-Target Detection and Tracking." Sensors, vol. 24, no. 14, July 2024, p. 4708. DOI.org (Crossref), https://doi.org/10.3390/s24144708.

[7] Elfring, Jos, et al. "Particle Filters: A Hands-On Tutorial." Sensors, vol. 21, no. 2, Jan. 2021, p. 438. DOI.org (Crossref), https://doi.org/10.3390/s21020438.

[8] Romera, Marta Marrón, et al. "Tracking Multiple and Dynamic Objects with an Extended Particle Filter and an Adapted K-Means Clustering Algorithm." IFAC Proceedings Volumes, vol. 37, no. 8, July 2004, pp. 310–15. DOI.org (Crossref), https://doi.org/10.1016/S1474-6670(17)31994-8.

[9] Raihan, Dilshad, and Suman Chakravorty. "Particle Gaussian Mixture Filters: Application and Performance Evaluation." 2019 22th International Conference on Information Fusion (FUSION) [Ottawa, ON, Canada], 2019, pp. 1–8. DOI.org (Crossref), https://doi.org/10.23919/FUSION43075.2019.9011177.

[10] Ristic, Branko, et al. "An Overview of Particle Methods for Random Finite Set Models." Information Fusion, vol. 31, Sept. 2016, pp. 110–26. DOI.org (Crossref), https://doi.org/10.1016/j.inffus.2016.02.004.

[11] Khalaf-Allah, Mohamed. “Emitter Location with Azimuth and Elevation Measurements Using a Single Aerial Platform for Electronic Support Missions.” Sensors, vol. 21, no. 12, June 2021, p. 3946. DOI.org (Crossref), https://doi.org/10.3390/s21123946.

[12] Cong, Heng, et al. “Multi-Target Tracking of Zebrafish Based on Particle Filter.” 2016 35th Chinese Control Conference (CCC) [Chengdu], 2016, pp. 10308–13. DOI.org (Crossref), https://doi.org/10.1109/ChiCC.2016.7554987.

[13] Bagnell, Drew. ”Good, Bad, and Ugly of Particle Filters.” Carnegie Mellon, Lecture Notes #5

[14] Ba-Ngu Vo, et al. “Random Finite Sets and Sequential Monte Carlo Methods in Multi-Target Tracking.” 2003 Proceedings of the International Conference on Radar (IEEE Cat. No.03EX695) [Adelaide, SA, Australia], 2003, pp. 486–91. DOI.org (Crossref), https://doi.org/10.1109/RADAR.2003.1278790.

[15] Hue, C., et al. “Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion.” IEEE Transactions on Signal Processing, vol. 50, no. 2, Feb. 2002, pp. 309–25. DOI.org (Crossref), https://doi.org/10.1109/78.978386.

[16] John Stone Stone, Apparatus for determining the direction of space telegraph signals, US Patent 716,135, December 16, 1902.

[17] Bensky, Alan. Wireless Positioning Technologies and Applications. Second edition, Artech House, 2016. GNSS Technology and Applications Series.

[18] Zhang, Jiao, and Jianfeng Lu. “Analytical Evaluation of Geometric Dilution of Precision for Three-Dimensional Angle-of-Arrival Target Localization in Wireless Sensor Networks.” International Journal of Distributed Sensor Networks, vol. 16, no. 5, May 2020, p. 155014772092047. DOI.org (Crossref), https://doi.org/10.1177/1550147720920471.

[19] Zhao, Qingqing, et al. “An Improved Non-Cooperative Signal Detection and Extraction Method Based on Template Matching.” 2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC) [Macau, China], 2017, pp. 365–68. DOI.org (Crossref), https://doi.org/10.1109/ICEIEC.2017.8076583.

[20] Bard, J. D., and F. M. Ham. “Time Difference of Arrival Dilution of Precision and Applications.” IEEE Transactions on Signal Processing, vol. 47, no. 2, Feb. 1999, pp. 521–23. DOI.org (Crossref), https://doi.org/10.1109/78.740135.

[21] Li, Binghao, et al. “Dilution of Precision in Positioning Systems Using Both Angle of Arrival and Time of Arrival Measurements.” IEEE Access, vol. 8, 2020, pp. 192506–16. DOI.org (Crossref), https://doi.org/10.1109/ACCESS.2020.3033281.

[22] A. Nagpal, A. Jatain and D. Gaur, ”Review based on data clustering algorithms,” 2013 IEEE Conference on Information & Communication Technologies, Thuckalay, India, 2013, pp. 298-303, doi: 10.1109/CICT.2013.6558109.

[23] Kodinariya, Trupti and Makwana, Prashant. "Review on Determining of Cluster in K-means Clustering." International Journal of Advance Research in Computer Science and Management Studies. 1, 2013, 90-95.

[24] Sinaga, Kristina P., and Miin-Shen Yang. "Unsupervised K-Means Clustering Algorithm." IEEE Access, vol. 8, 2020, pp. 80716–27. DOI.org (Crossref), https://doi.org/10.1109/ACCESS.2020.2988796.

[25] Rousseeuw, Peter J. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." Journal of Computational and Applied Mathematics, vol. 20, Nov. 1987, pp. 53–65. DOI.org (Crossref), https://doi.org/10.1016/0377-0427(87)90125-7.

[26] Schubert, Erich. "Stop Using the Elbow Criterion for K-Means and How to Choose the Number of Clusters Instead." ACM SIGKDD Explorations Newsletter, vol. 25, no. 1, June 2023, pp. 36–42. DOI.org (Crossref), https://doi.org/10.1145/3606274.3606278.

[27] Schwarz, Gideon. "Estimating the Dimension of a Model." The Annals of Statistics, vol. 6, no. 2, Mar. 1978. DOI.org (Crossref), https://doi.org/10.1214/aos/1176344136.

[28] Derpanis, Konstantinos G. "Mean shift clustering." Lecture Notes 32.1-4 (2005): 16.

[29] Yizong Cheng. "Mean Shift, Mode Seeking, and Clustering." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 8, Aug. 1995, pp. 790–99. DOI.org (Crossref), https://doi.org/10.1109/34.400568.

[30] Fukunaga, K., and L. Hostetler. "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition." IEEE Transactions on Information Theory, vol. 21, no. 1, Jan. 1975, pp. 32–40. DOI.org (Crossref), https://doi.org/10.1109/TIT.1975.1055330.

[31] Morris, Katherine, et al. "Asymmetric Clusters and Outliers: Mixtures of Multivariate Contaminated Shifted Asymmetric Laplace Distributions." Computational Statistics & Data Analysis, vol. 132, Apr. 2019, pp. 145–66. DOI.org (Crossref), https://doi.org/10.1016/j.csda.2018.12.001.

[32] Deng, Dingsheng. "DBSCAN Clustering Algorithm Based on Density." 2020 7th International Forum on Electrical Engineering and Automation (IFEEA) [Hefei, China], 2020, pp. 949–53. DOI.org (Crossref), https://doi.org/10.1109/IFEEA51475.2020.00199.

[33] P. D. Groves, "Principles of GNSS, inertial, and multisensor integrated navigation systems, 2nd edition [Book review]," IEEE Aerospace and Electronic Systems Magazine, vol. 30, no. 2, pp. 26–27, Feb. 2015, Conference Name: IEEE Aerospace and Electronic Systems Magazine, issn: 1557-959X. doi: 10.1109/MAES.2014.14110.