

# **Data Fusion Algorithms for Collaborative Localisation**

A thesis submitted in partial fulfilment of the degree of  
Bachelor of Engineering (Honours)

by  
**Caitlin Lovejoy**  
**U6959326**

Supervisor: Prof. Jochen Trumpf  
Examiner: Dr. Behzad Zamani



College of Engineering, Computing & Cybernetics  
The Australian National University

Nov. 2023

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of the author's knowledge, it contains no material previously published or written by another person, except where due reference is made in the text.

Caitlin Lovejoy  
17 November 2023

# Acknowledgements

I would like to express my deepest gratitude to my supervisor Prof. Jochen Trumpf for all his support, patience, and invaluable guidance in the completion of this project.

I would also like to extend my thanks to my friends who provided constant support and encouragement, particularly to all who listened to me talk about my project topic for hours on end over the last year, and those who helped me in the proof reading and editing of this report.

Lastly, I would like to thank my Mum and Dad for their continued belief in me and unwavering support through this process.

# Abstract

Localisation is one of the fundamental components of navigation for a mobile robot or autonomous system, with accurate localisation being a key requirement for the successful completion of higher-level tasks such as path planning and object avoidance. This report considers the case of bearing only localisation, which often needs to utilise collaborative localisation techniques. Data fusion algorithms are examined in their application to this collaborative localisation problem in a bottom-up problem architecture. Their performance is analysed and along with an assessment of the measures of performance applied in testing. This report makes its main contribution in the comparison of the Ellipsoidal Intersection (EI) method with methods such as Covariance Intersection (CI) and the Convex Combination Ellipsoid (CCE). It shows that EI demonstrates promise in this application but has some properties that need further exploration in future works.

# Contents

<b>Acknowledgements</b> .....	<b>i</b>
<b>Abstract</b> .....	<b>ii</b>
<b>Contents</b> .....	<b>iii</b>
<b>List of figures</b> .....	<b>v</b>
<b>List of tables</b> .....	<b>vii</b>
<b>Chapter 1 Introduction</b> .....	<b>1</b>
<b>Chapter 2 Background</b> .....	<b>4</b>
2.1 Ellipsoidal Representations .....	4
2.1.1 Matrix Representation Ellipsoids .....	4
2.1.2 Ellipsoidal Representation of Uncertainty and Noise .....	5
2.2 Localisation .....	5
2.2.1 Bearing Only Localisation .....	6
2.3 Collaborative Localisation Problem Approaches .....	7
2.3.1 Top-down Approaches and Centralised Architecture .....	7
2.3.2 Bottom-up Approaches and Decentralised Architecture .....	8
2.4 Data Fusion Algorithms .....	9
2.4.1 Kalman Fusion Method .....	10
2.4.2 Covariance Intersection .....	10
2.4.3 Convex Combination Ellipsoid .....	11
2.4.4 S-Procedure for Ellipsoidal Containment .....	12
2.4.5 Ellipsoidal Intersection .....	12
2.4.6 Inverse Covariance Intersection .....	13
2.5 Measures of Performance .....	13
2.5.1 Ideal characteristics of fused estimates .....	13
2.5.2 Confidence .....	14
2.5.3 Statistical Consistency .....	14
2.5.4 Noise floor .....	15
<b>Chapter 3 Methodology</b> .....	<b>16</b>
3.1 Selection of Data Fusion Algorithms .....	16
3.2 Implementation Environment .....	16
3.3 Experimental Designs .....	17
3.3.1 Isolated Data Fusion .....	17
3.3.2 Collaborative Localisation Problem Formulation .....	17
3.3.2.1 Measurement Model .....	18

3.3.2.2	Collaborative Bearing Localisation Algorithm .....	18
3.3.2.3	Dealing with Non-overlapping Cases .....	19
3.3.3	Data Fusion in Collaborative Localisation, Plausibility Study .....	20
3.3.4	Data Fusion in Collaborative Localisation, Monte Carlo Simulation .....	21
<b>Chapter 4</b>	<b>Results and Analysis .....</b>	<b>22</b>
4.1	Verification of Implementation .....	22
4.1.1	Verification of EI .....	22
4.1.2	Verification of Kalman Fusion, CI, CCE, and ICI .....	23
4.2	Isolated Data Fusion .....	24
4.2.1	Property 1, Intersection Containment .....	24
4.2.2	Property 2, Boundary Intersections .....	25
4.2.3	Property 3, Containment in the Union .....	26
4.3	Plausibility Study Simulation .....	27
4.3.1	Noise Floor Measurements .....	30
4.4	Monte Carlo Simulation .....	32
<b>Chapter 5</b>	<b>Conclusions and Future Work .....</b>	<b>34</b>
<b>References</b>	<b>.....</b>	<b>A</b>
<b>Appendix A</b>	<b>.....</b>	<b>C</b>

# List of figures

Figure 1. Visual representation of ellipse geometry calculated from matrix representation.....4

Figure 2. The measurement ellipse for the sensor of an agent located at position  $\mathbf{x}$  measuring a noisy relative bearing angle  $\theta$  towards a target  $\mathbf{p}$ . It is assumed that the sensor has a measurement range  $[\mathbf{r}, \mathbf{r}]$  and a standard deviation error of  $\sigma$  [7] .....7

Figure 3. (a) Graphic (b) Network architecture, representation of centralised fusion for a team of 4 mobile robots where agent 1 acts as the central node ..... 8

Figure 4. (a) Graphic (b) Network architecture, representation of decentralised fusion for a pair of mobile robots ..... 9

Figure 5. A 2 agent bearing localisation scenario with stationary agents and a stationary reference point. Where the shared initial belief is circular and is centred at  $\mathbf{2}, -\mathbf{1T}$  and the target reference point is located at  $\mathbf{10}, -\mathbf{12T}$ . The noisy measurements of each agent are depicted in their respective colour. ....20

Figure 6. The prior and fused estimates according to EI and CI, with two cases of EI,  $\zeta\mathbf{1} = \mathbf{10} - \mathbf{6}, \zeta\mathbf{2} = \mathbf{0.1}$  (a) shows a replication of (b) sourced from [14].....23

Figure 7. The prior and fused estimates according to Kalman, CCE, CI and ICI (a) shows a replication of (b) sourced from [7] .....24

Figure 8. Isolated fusion example demonstrating ideal fusion property 1. ....25

Figure 9. Isolated fusion example demonstrating ideal fusion property 2. ....26

Figure 10. Isolated fusion example demonstrating ideal fusion property 3. ....27

Figure 11. Estimation error over time for the non-collaborative, Kalman, CI, ICI, CCE, and EI methods for each agent. ....27

Figure 12. The determinant of the estimated error covariance, tracked over time, for the non-collaborative, Kalman, CI, ICI, CCE, and EI methods for each agent. The figure is zoomed to the first 50 steps of the simulation. ....28

Figure 13. The NEES over time for the non-collaborative, CI, ICI, CCE and EI methods for each agent. ....29

Figure 14. The NEES over time for the non-collaborative, Kalman, CI, ICI, CCE and EI methods for each agent. The Kalman fusion method’s NEES grows to such a magnitude that measures cannot be seen for any other method. The final marking on the y axis is  $\mathbf{1.6} \times \mathbf{1054}$ .....29

Figure 15. Estimation error over time for the non-collaborative, Kalman, CI, ICI, CCE, and EI methods for each agent. ....30

Figure 16. Average change of position estimate over time for the non-collaborative, Kalman, CI, ICI, CCE, and EI methods for each agent ..... 31

Figure 17. Histogram of the final estimation error of the Kalman, CI, ICI, CCE, and EI methods at the final fusion step. Run for 1000 simulations to 150 steps. .... 32

Figure 18. Histogram of the rankings of the final estimation error of the Kalman, CI, ICI, CCE, and EI methods in context to each other at the final simulation step. Run for 1000 simulations to 150 steps..... 33

# List of tables

Table 1. Summary of Results of Noise Floor Test .....31

# Chapter 1 Introduction

Localisation is one of the fundamental components of navigation for a mobile robot or autonomous system, referring to the ability of a robot to determine its position, and sometimes orientation, within an environment [1]. Accurate localisation is essential for the successful execution of more advanced processes such as path-planning and target tracking [2]. Localisation can be conducted using a variety of methods to solve different problem formulations, making use of various proprioceptive and exteroceptive sensors [3].

In scenarios such as search and rescue operations, indoor navigation, and military applications, where the environment may be GPS denied or the use of GPS and other various active exteroceptive sensors may not be applicable, it is essential to be able to perform fast and accurate localisation through the means of different sensor technologies [2]. One such category of sensor that can be applied in these situations are bearing measurement sensors and therefore, the focus of this report will be on the problem of bearing only localisation [4].

However, due to the nature of bearing measurements being unable to provide information about the distance from the autonomous robot to its reference point, it is often infeasible for a single agent to be able to make precise estimates about its location.

Due to this limitation a technique known as collaborative localisation is often employed [2][5][6]. Collaborative localisation is the practice of sharing information between agents within a team of mobile robots to improve location accuracy [6] beyond what could have been possible from each agent's individual interactions with the environment alone [7]

For a team of mobile robots, it has been shown that the use of collaborative localisation significantly improves the performance of each of the individual agents, and the proper combination of complimentary information from multiple agents leads to a localisation estimate with increased accuracy and reduced uncertainty [2][6].

In the literature there are a variety of algorithm approaches that have been employed to try and solve the collaborative localisation problem, and these approaches can be differentiated into two categories, these being top-down approaches and bottom-up approaches. These categories

are defined by their initial network architectures and topologies, requirements placed on communications, and resources needed.

The top-down approaches begin with a centralised fusion architecture [5], formulating a joint estimation problem over the entire network. In the centralised fusion architecture, raw measurements made by agents are communicated directly to a central node that is solely responsible for computing and storing the estimates [8]. The storage and computation responsibilities are then distributed to agents across the network.

Top-down approaches, while widely used and extensively studied, require very precise control over communications, tracking of correlations, and a constant ability for nodes to communicate with the central node. The strict requirements placed on the network communications, and the single point of failure in the form of the central node, makes the top-down approach an unreliable option in many real-world applications [2][7]

In contrast, the bottom-up approaches begin with a decentralised fusion architecture, formulating the estimation problem in a way that is largely independent of the underlying network topology [7], where each agent independently makes measurements, and computes and stores estimates of the location of the reference. Communications are then introduced between the nodes and they are allowed to communicate their beliefs in order to improve upon their individual estimates.

Bottom-up approaches can often allow for a more opportunistic approach to information sharing [7], giving the advantage of lower infrastructure and communication costs [8]. The lack of a central processing node that must keep track of communications also makes the decentralised fusion architectures more robust to failures.

The decentralised fusion and distributed fusion architectures resulting from these approaches both encounter the issue of having to account for and mitigate the impacts of correlations in the information communicated over the network [8]. Once estimates have been communicated between agents of the network the estimates resulting can no longer be assumed to be independent and now have unknown cross correlation with estimates already held at the node.

While the top-down approach algorithms can often be fully distributed through careful tracking and control of communications and correlations, and placing assumptions on the network connectivity and reliability [7], these approaches require information about cross correlation for the information fusion to take place, and must make approximations when this information is unavailable.

Bottom-up approaches do not track cross correlation and instead disregard it, performing fusion under unknown correlation. This is achieved through the use of Data Fusion algorithms which seek to provide a fusion of estimates and covariances, where in the case of the unknown correlations, the fusion process ideally guarantees to retain the common uncertainties of the original information [8].

There is a fundamental trade-off between the level of communication and the overall performance of the fusion algorithm. With complete knowledge such as in the centralised fusion architecture the accuracy will be higher, however, it is widely accepted that the overall optimality of this fusion will be sacrificed for the practicality of the decentralised architecture [7].

In the literature there exists a large body of work exploring the top-down approaches to collaborative localisation and their attributes. However, the full extent of the properties and therefore the potential of the bottom-up approach has been significantly less closely examined.

This report seeks to expand the exploration of the bottom-up approach through the application of data fusion algorithms within collaborative localisation algorithms. It will review the key theories and methodologies of data fusion under unknown correlation with a specific view towards their application to collaborative localisation problems.

There are two questions this report aims to answer in regard to this goal.

- Is there a Data Fusion algorithm that provides an acceptable level of performance when used within collaborative localisation algorithms?
- What measures of performance are useful in the comparison data fusion algorithms for collaborative localisation?

In order to answer these questions this report will first cover the necessary background information in Chapter 2, before outlining the methodology that was applied in Chapter 3. Following this, results will be presented for the outlined methods in Chapter 4. Chapter 5 will present conclusions and discuss potential future work.

# Chapter 2 Background

## 2.1 Ellipsoidal Representations

This section covers some of the mathematical background of ellipsoidal representations using matrices that will be essential for understanding in this report.

### 2.1.1 Matrix Representation Ellipsoids

In the 2-dimensional case considered in this report, an ellipse around an undefined centre point can be described by the matrix  $P$ , shown in equation (1), known as the shape matrix.

$$P = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \quad (1)$$

From the shape matrix the radii and rotation of the ellipse can be calculated by determining the eigenvalues  $\lambda_1, \lambda_2$  as shown in equations (2a) and (2b), and the angle  $\theta$  as shown in equation (2c).

$$\lambda_1 = \frac{a+c}{2} + \sqrt{\left(\frac{a-c}{2}\right)^2 + b^2} \quad (2a)$$

$$\lambda_2 = \frac{a+c}{2} - \sqrt{\left(\frac{a-c}{2}\right)^2 + b^2} \quad (2b)$$

$$\theta = \text{atan2}(\lambda_1 - a, b) \quad (2c)$$

Where  $\sqrt{\lambda_1}$  is the radius of the major axis and  $\sqrt{\lambda_2}$  is the radius of the minor axis. The angle of rotation  $\theta$  is measured in radians anti-clockwise from the positive x axis- to the ellipse's major axis. This is visualised in Figure 1.

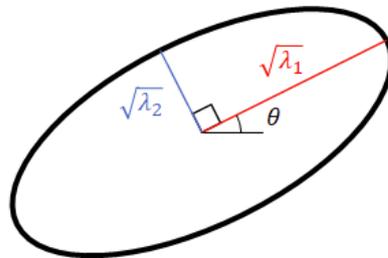


Figure 1. Visual representation of ellipse geometry calculated from matrix representation.<sup>1</sup>

<sup>1</sup> Image sourced from <https://cookierobotics.com/007/>

### 2.1.2 Ellipsoidal Representation of Uncertainty and Noise

Most techniques for parameter estimation assume that the data is corrupted by random noise and uncertainty that can be assumed to be Gaussian [9]. In order to model this, an uncertainty ellipsoid  $\mathcal{E} \in \mathbb{R}^n$  can be constructed around an estimated centre point  $c \in \mathbb{R}^n$ , with the shape matrix  $P \in \mathbb{R}^{n \times n}, P > 0$ . In the 2-dimensional case  $n = 2$  the uncertainty ellipsoid can be viewed as a representation of a bivariate Gaussian distribution where the centre point  $c = [\mu_1 \ \mu_2]$  and the shape matrix  $P$  is called the covariance matrix and is defined as shown below.

$$P = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix} \quad (3)$$

The covariance matrix produces a bounded region around the centre point representing an area in which it is statistically likely that we will find the true point within.

## 2.2 Localisation

Localisation is one of four components of the conceptual framework of navigation for a mobile robot and has received significant research attention over the last few decades [1].

The ability of a mobile robot to determine its position, and sometimes orientation, in space can be achieved using a variety of sensors in a variety of environments. A particular issue of interest is the ability of a mobile robot to localise itself in an environment where use of global positioning systems (GPS) is restricted in some way.

To localise a robot indoors or in obstructed areas with geographical features that will interfere with GPS accuracy, different sensors must be selected to achieve this application. Additionally, the use of GPS will only provide the global position of the robot whereas in many scenarios a relative position is required. This could be relative to a reference point, target, other robots, or humans.

For this purpose, this report explores the use of bearing only localisation. Section 2.2.1, defines bearing only localisation, discusses bearing measurement representation, as well as its limitations in localisation, which motivates the application of collaborative localisation methods.

### 2.2.1 Bearing Only Localisation

Bearing only localisation makes use of relative bearing measurements to determine an agent's position with respect to a target, reference point, or landmark [4]. In bearing localisation angular measurements are taken by the agent to a reference, giving a relative bearing in the local coordinate frame attached to the agent. This then maps to a true bearing in a global coordinate frame. Bearings are the angles measured in degrees, clockwise from north, or in standard 2D coordinate frames, clockwise from the y-axis.

Each agent is equipped with some sensor capable of producing bearing angle measurements  $\theta$ , between a reference point  $p \in \mathbb{R}^n$  and an agent's location  $x \in \mathbb{R}^n$

$$\theta = \text{atan} \left( \frac{p-x}{\|p-x\|} \right) + \delta \quad (4)$$

Where  $\delta$  represents an unknown measurement error with an assumed Gaussian distribution,  $\delta \sim \mathcal{N}(0, \sigma^2)$ , where  $\sigma^2$  is a known standard deviation.

This bearing measurement is non-linear, and would typically require linearisation or the application of non-linear filtering approaches such as the extended Kalman filter [10]. However, as shown in [7], an ellipsoidal measurement modelling approach may be used instead. This is applied by making a further assumption that the minimum and maximum range,  $\underline{r}$  and  $\bar{r}$  respectively, within which the sensor is designed to operate, is known. Placing this assumption on the range allows for the calculation of the measurement ellipse  $\mathcal{E}^m(cm, Pm)$ , where  $cm \in \mathbb{R}^n$  is the centre, with the shape/covariance matrix  $Pm \in \mathbb{R}^{n \times n}$ , with  $n = 2$  describing the 2D case. It is proposed in [7] that a similar approach can be extended to the 3D case; however, this lies outside the scope of this report. The equations below give the mathematical definition of this measurement ellipse. A graphical representation is given in Figure 2.

$$cm = x + \left( \frac{\underline{r} + \bar{r}}{2} \right) \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \quad (5)$$

$$Pm = RDR^T \quad (6)$$

$$wr = \frac{\bar{r} - \underline{r}}{2}, \quad hr = \left( \frac{\underline{r} + \bar{r}}{2} \right) \quad (7a)$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \quad D = \begin{bmatrix} wr^2 & 0 \\ 0 & hr^2 \end{bmatrix} \quad (7b)$$

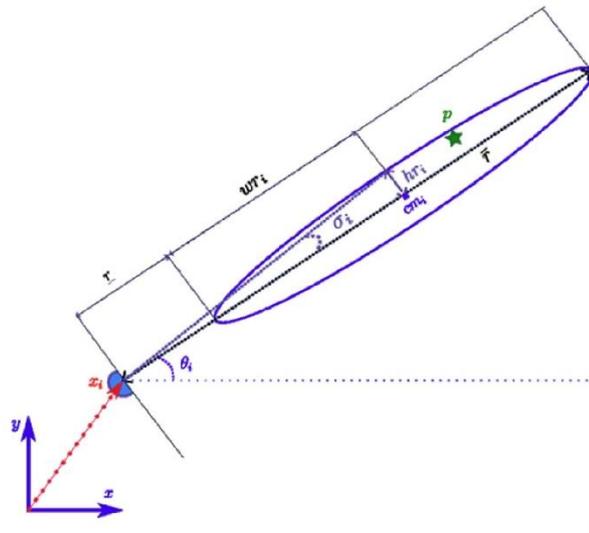


Figure 2. The measurement ellipse for the sensor of an agent located at position  $x$  measuring a noisy relative bearing angle  $\theta$  towards a target  $p$ . It is assumed that the sensor has a measurement range  $[r, \bar{r}]$  and a standard deviation error of  $\sigma$  [7]

This ellipsoidal measurement modelling approach allows for the use of a simplified fusion approach when an agent is fusing a measurement it has taken with its held estimate.

The ellipsoidal representation as seen in Figure 2, additionally demonstrates the lack of depth information conveyed by the measurement, the reference point can be located anywhere within the measurement ellipsoid and therefore it can be anywhere within the range of operation of the sensor. This reinforces the need for collaborative localisation methods to be applied in order to obtain a localisation with a higher accuracy and a decreased uncertainty.

## 2.3 Collaborative Localisation Problem Approaches

In the application of collaborative localisation with a team of mobile robots, there have been a variety of approaches developed to model the network topology and fusion architecture. These approaches vary in network infrastructure and communication required, optimality of final estimations, practicality, and robustness. Based on their starting network topologies, these approaches can be differentiated into top-down approaches and bottom-up approaches. This section will provide an overview of these approaches, their benefits, and their drawbacks.

### 2.3.1 Top-down Approaches and Centralised Architecture

In a top-down approach, such as centralised filtering/estimation or collaborative filtering/estimation, the problem begins with a centralised fusion architecture, in which a central node is responsible for all computations and storage of estimates, taking in measurements

provided by agents in the network, as shown in Figure 3. The data processing, computations, and storage of estimates are then distributed out, allowing each agent to make their own measurements and estimations. This distribution however requires strict control of communications and tracking of the cross correlations across the network [7]. If this data is correctly maintained and accounted for, and there is no limitation on communications or constraint on communication bandwidth the top-down approaches yield theoretically optimal solutions to the state estimation [8].

However, the distributed fusion network formed by the top-down approach has many undesirable characteristics, such as its inflexibility to changes in the network architecture, reliance on the central node making it susceptible to failures, and the high computational load placed on the central node [8]. Additionally, the maintenance and tracking of cross-correlations is expensive as it scales quadratically with the number of updates [3]. These factors make the top-down approach an impractical choice for a variety of applications despite the optimality of the solutions.

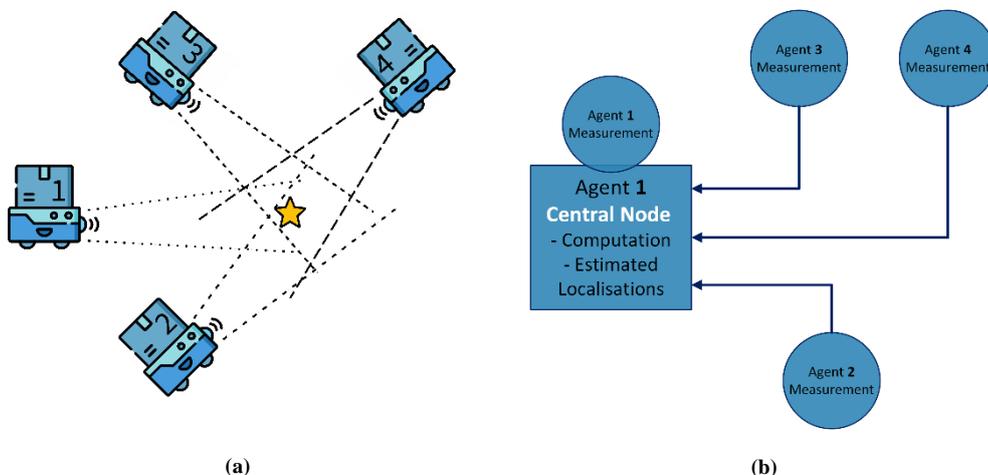


Figure 3. (a) Graphic (b) Network architecture, representation of centralised fusion for a team of 4 mobile robots where agent 1 acts as the central node

### 2.3.2 Bottom-up Approaches and Decentralised Architecture

In a bottom-up approach to collaborative localisation, the initial network is modelled as a decentralised fusion architecture, as represented in Figure 4. In this model data is measured by each agent and processed independently to obtain a local estimate which is stored by the agent. The agents are then allowed to communicate their stored estimates with each other. The bottom-up approach does not introduce a central node at any point resulting in a notably different final network architecture compared to the top-down approach.

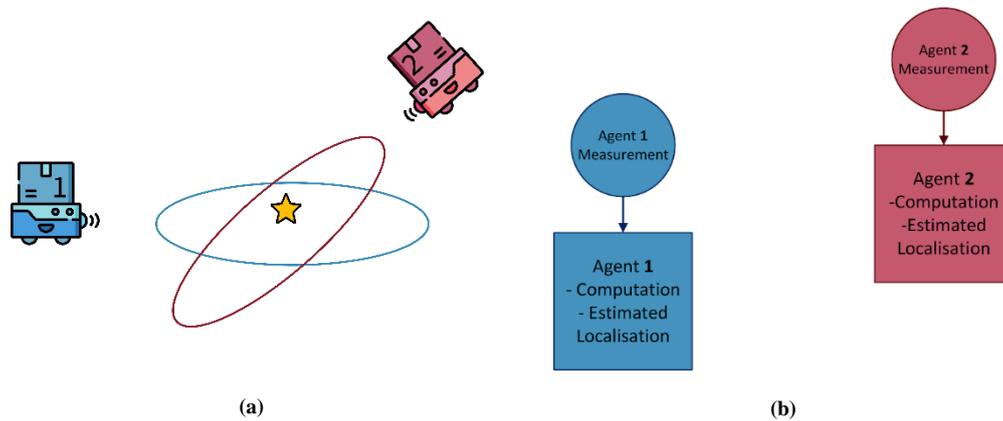


Figure 4. (a) Graphic (b) Network architecture, representation of decentralised fusion for a pair of mobile robots

The bottom-up approach has the benefit of being flexible in its communications. However, this architecture can easily run in to problems with the correlation of data [8]. As there is no centralised node keeping records of how data has been correlated, once estimates have been communicated between agents of the network, the estimates can no longer be assumed to be independent and now have unknown cross correlation with estimates already held by the agent.

The need for methods of mitigating the effects of unknown correlations in the bottom-up approach leads to a sacrifice in the optimality of the solution. However, this sacrifice of optimality is generally accepted in exchange for the practicality of the decentralised architecture [7].

The bottom-up approach is more practical in application as it is inherently more flexible and robust to failure, while requiring fewer resources.

## 2.4 Data Fusion Algorithms

In this report, data fusion algorithms specifically refer to algorithms for distributed data fusion using ellipsoidal methods, such as covariance intersection and ellipsoidal intersection. These methods seek to provide a fused estimate through the calculation of an uncertainty ellipsoid containing the intersection of the two uncertainty ellipsoids being fused. This section will cover a selection of data fusion algorithms that will be considered in later sections.

### 2.4.1 Kalman Fusion Method

The Kalman fusion method can be seen as the most basic data fusion algorithm from which further algorithms are built. Given a pair of unbiased pair of estimates  $(\hat{x}_i, \hat{P}_i)$  and  $(\hat{x}_j, \hat{P}_j)$  for an unknown point of interest  $x \in \mathbb{R}^n$ , where  $\hat{x}_i, \hat{x}_j \in \mathbb{R}^n$ , denote estimated values of point  $x$ , and  $\hat{P}_i, \hat{P}_j \in \mathbb{R}^{n \times n}$ ,  $\hat{P}_i, \hat{P}_j > 0$ , denote the estimated covariance matrices for the estimation errors.

Under the assumption that the prior estimates are independent, it can be shown that the Kalman fusion equations below provides a fused estimate  $(\hat{x}^+, \hat{P}^+)$  that can be considered optimal with respect to a variety of criteria [11][7]. The notation of  $\hat{x}^+$  and  $\hat{P}^+$  represent the updated point estimate and updated estimation error covariance following the fusion.

$$\hat{P}^+ = (\hat{P}_i^{-1} + \hat{P}_j^{-1})^{-1} \quad (8a)$$

$$\hat{x}^+ = \hat{P}^+ (\hat{P}_i^{-1} \hat{x}_i + \hat{P}_j^{-1} \hat{x}_j) \quad (8b)$$

The Kalman fusion method, while it can be considered optimal in some cases, makes the assumption that the two prior estimates are independent. This assumption puts the Kalman fusion method at a disadvantage in bottom-up collaborative localisation problems where data that has been communicated between agents often has an unknown correlation.

When Kalman fusion is applied to estimates that are not independent, it loses its optimality and can suffer from a phenomenon called the over-confidence problem in the fusion literature. The over-confidence problem refers to the Kalman fusion producing an estimated covariance that is small enough to inhibit the progress of the algorithm in future steps, while the estimation error is still high. An over-confident prior estimate passed to future steps will dominate the update of the point estimate, which is calculated as the covariance weighted sum of prior points, shown in equation (8b).

### 2.4.2 Covariance Intersection

The Covariance Intersection method (CI) builds upon the Kalman fusion method in order to perform fusion under unknown correlation. It provides a fused estimate of the covariance matrix

$\hat{P}^+$  of the prior covariances that is guaranteed to contain the intersection, regardless of the actual correlation between the two prior estimates [12].

$$\hat{P}^+ = (\alpha \hat{P}_i^{-1} + (1 - \alpha) \hat{P}_j^{-1})^{-1} \quad (9a)$$

$$\hat{x}^+ = \hat{P}^+ (\alpha \hat{P}_i^{-1} \hat{x}_i + (1 - \alpha) \hat{P}_j^{-1} \hat{x}_j) \quad (9b)$$

Where  $\alpha \in [0,1]$  is a free parameter

The optimal solution of CI can be chosen from the solution set it produces, through the manipulation of the parameter  $\alpha$ . An example of this is selecting  $\alpha$  such that the resulting covariance matrix  $\hat{P}^+$  has a minimal determinant [7][12]. However, CI can be shown to make overly conservative estimates, as it does not necessarily produce a tight bounding of the intersection.

### 2.4.3 Convex Combination Ellipsoid

The Convex Combination Ellipsoid (CCE) method is similar in structure to CI. However, it can be shown to possess a number of properties that make it a more ideal candidate for bearing only localisation [7]; these will be discussed in later sections of this report. It can be considered to provide a tighter bounding of the intersection, guaranteeing that the intersection is contained within the fused estimate of the covariance matrix  $\hat{P}^+$ .

$$\hat{P}^+ = kX \quad (10a)$$

$$\hat{x}^+ = X(\alpha \hat{P}_i^{-1} \hat{x}_i + (1 - \alpha) \hat{P}_j^{-1} \hat{x}_j) \quad (10b)$$

$$X = (\alpha \hat{P}_i^{-1} + (1 - \alpha) \hat{P}_j^{-1})^{-1} \quad (10c)$$

$$k = 1 - d^2 \quad (10d)$$

$$d^2 = \|\hat{x}_j - \hat{x}_i\|^2 \left( \frac{\hat{P}_i + \hat{P}_j}{\alpha + 1 - \alpha} \right)^{-1} \quad (10e)$$

Where  $\alpha \in [0,1]$  is a free parameter

The optimal solution of CCE can also be determined by optimising the parameter  $\alpha$  in a similar way to CI, such as minimising the determinant of the resulting covariance matrix  $\hat{P}^+$ .

#### 2.4.4 S-Procedure for Ellipsoidal Containment

There exists in the literature another algorithm called the S-Procedure for Ellipsoidal Containment, however this method has been shown to provide equivalent results as the CCE method [13] in the 2-dimensional case considered in this report, while CCE is computationally cheaper [9]. Therefore, the S-procedure was not considered further.

#### 2.4.5 Ellipsoidal Intersection

The Ellipsoidal Intersection method (EI) was proposed by [14] as a fusion method in which the resulting fusion always possesses a higher accuracy than that of the prior estimates. It provides an explicit characterisation of the unknown correlations before computing a fusion based on the independent components of the estimates.

EI employs the common error term  $\Gamma$  to model the unknown correlations and reports a far less conservative result compared to CI [14][15].

$$\hat{P}^+ = (\hat{P}_i^{-1} + \hat{P}_j^{-1} - \hat{\Gamma}^{-1})^{-1} \quad (11a)$$

$$\hat{x}^+ = \hat{P}^+ (\hat{P}_i^{-1} \hat{x}_i + \hat{P}_j^{-1} \hat{x}_j - \hat{\Gamma}^{-1} \gamma) \quad (11b)$$

$$\Gamma = T D_\Gamma T^T \quad (11c)$$

$$T = S_i D_i^{1/2} S_j \quad (11d)$$

$$Q_{ij} = D_i^{1/2} S_i^{-1} P_j S_i D_i^{1/2} \quad (11e)$$

$$\gamma = (P_i^{-1} + P_j^{-1} - 2\Gamma^{-1} + 2\eta I_n)^{-1} \times \left( (P_j^{-1} - \Gamma^{-1} + \eta I_n) \hat{x}_i + (P_i^{-1} - \Gamma^{-1} + \eta I_n) \hat{x}_j \right) \quad (11f)$$

$$\eta = \begin{cases} 0 & \text{if } |[D_j]_{qq} - 1| \geq 10\zeta, \forall q = 1, \dots, n \\ \zeta & \text{else} \end{cases} \quad (11g)$$

Where  $\zeta$  is a free parameter

Where  $S_i$  and  $D_i$  are the eigenvectors and eigenvalues respectively resulting from the eigenvalue decomposition of  $P_i$ , and  $S_j$  and  $D_j$  is eigenvectors and eigenvalues respectively resulting from the eigenvalue decomposition of  $Q_{ij}$ .

The EI method however has not yet been demonstrated to be a consistent fusion method [14]. Additionally, EI requires the tuning of a regularisation parameter  $\zeta$  in the formation of the updated position estimate, as this parameter is a free variable.

It can also be seen that due to the explicit characterisation of the error there are many more steps in the process of computing the fusion.

## 2.4.6 Inverse Covariance Intersection

The creation of the Inverse Covariance Intersection method (ICI) was inspired by weak points of the Ellipsoidal Intersection method noted in [15], aiming to provide a more consistent fusion than that of EI and a more accurate fusion than that of CI.

$$(\hat{P}^+)^{-1} = \hat{P}_i^{-1} + \hat{P}_j^{-1} - (\alpha\hat{P}_i + (1 - \alpha)\hat{P}_j)^{-1} \quad (12a)$$

$$\hat{x}^+ = K\hat{x}_i + L\hat{x}_j \quad (12b)$$

$$K = \hat{P}^+ \left( \hat{P}_i^{-1} - \alpha(\alpha\hat{P}_i + (1 - \alpha)\hat{P}_j)^{-1} \right) \quad (12c)$$

$$L = \hat{P}^+ \left( \hat{P}_j^{-1} - (1 - \alpha)(\alpha\hat{P}_i + (1 - \alpha)\hat{P}_j)^{-1} \right) \quad (12d)$$

## 2.5 Measures of Performance

This section provides an overview of the measures of performance that will be applied to assess the data fusion algorithms in chapter 4.

### 2.5.1 Ideal characteristics of fused estimates

It is desirable that when conducting data fusion, the fused ellipsoid does not overlook any possible solution or introduce new errors. In order to define this, there are three ideal properties that the fused ellipsoid solution is desired to possess.

- 1) The ellipsoid  $\mathcal{E}_\alpha$  contains the intersection of the two prior ellipsoids,  $\mathcal{E}_i \cap \mathcal{E}_j \subseteq \mathcal{E}_\alpha$
- 2) The intersections between the boundaries of  $\mathcal{E}_i$  and  $\mathcal{E}_j$  lie on the boundary of  $\mathcal{E}_\alpha$
- 3) The ellipsoid  $\mathcal{E}_\alpha$  is contained within the union of the two prior ellipsoids,  $\mathcal{E}_\alpha \subseteq \mathcal{E}_i \cup \mathcal{E}_j$

As pointed out by the authors of [9], and reiterated by the authors of [7], the first and second properties ensure a tight bounding of the intersection of the prior ellipsoids, while the third

property ensures that the resulting fused uncertainty ellipsoid does not introduce additional uncertainties.

### 2.5.2 Confidence

The estimation covariance, measured as the determinant of the covariance matrix of the estimate, can be viewed as a measure of confidence in the measurement. A minimal determinant represents a smaller area of uncertainty in which the true location can be located, and therefore this measurement can be seen as holding a higher level of confidence.

The property of a fused estimate having a minimal determinant is a property that is also referred to as tightness of the estimate [7].

### 2.5.3 Statistical Consistency

A state estimator is said to be consistent if the means of the estimation errors are zero, meaning that the estimates are unbiased, and if their covariance matrices of the error are as calculated by the fusion algorithm.

The normalised estimation error squared (NEES) is a commonly applied measure for the evaluation of the consistency of a state estimator [16] when ground truth information is available for comparison to the estimates. It is important to consider the consistency of the fused estimates, as if a state estimator is inconsistent and is unable to accurately indicate the quality of its estimate, it cannot provide the most optimal result [17].

NEES is a statistical measure that tests both the mean and covariance. It is defined as shown below in equation (13)

$$\epsilon = (\hat{x} - x)^T \hat{P}^{-1} (\hat{x} - x) \quad (13)$$

Where  $x \in \mathbb{R}^n$  is the true location of the reference,  $\hat{x} \in \mathbb{R}^n$  is the estimated location and  $\hat{P} \in \mathbb{R}^{n \times n}$ ,  $P > 0$ , is the estimated covariance [17].

Under the assumption that the fused estimates are consistent and the distributions of uncertainty are Gaussian  $\epsilon$  has an expected value,  $E[\epsilon] = n_x$ , where  $n_x$  is the dimension of  $x$ .

### **2.5.4 Noise floor**

The noise floor refers to the level of error or noise remaining when the algorithm converges a steady state. Two pieces of information can be taken in regard to this, these being the speed at which the algorithm reaches its noise floor, and the residual noise itself at the noise floor. These can be used to infer the speed at which the algorithm can be expected to perform and can guide a base level of expected accuracy in a given problem scenario.

## **Chapter 3 Methodology**

The methodology applied for this report had four stages: an initial selection of data fusion algorithms, the implementation of these algorithms in a python environment, experimental design, and iterative refinement of the experimentation as new properties and characteristics were observed.

In order to best assess the properties of the various data fusion algorithms, and apply a variety of performance measures, three general experimental setups were defined: isolated data fusion of two ellipses, single simulations of data fusion algorithms in collaborative localisation, and Monte Carlo simulations of the data fusion algorithms in collaborative localisation.

### **3.1 Selection of Data Fusion Algorithms**

The data fusion algorithms to be studied were selected from the literature. All algorithms discussed in section 2.4 were chosen to be candidates, excluding Ellipsoid Containment using the S-Procedure, as it has been shown to yield the same results as the Convex Combination Ellipsoid for the intersection of 2 ellipsoids [13]. However, the CCE method has been proven to be computationally cheaper than the S-Procedure based method [9]

The algorithms selected are believed to be a good sampling of the field and encompass some of the most commonly researched and applied fusion methods as well as some more modern inclusions.

### **3.2 Implementation Environment**

An implementation environment was developed in Python by the authors of [7]. This environment was then provided for use in this report. The provided implementation environment was extended through the addition of the Ellipsoidal Intersection algorithm, and the NEES and noise floor performance measures. Details of the implementation environment and attribution of work can be found in Appendix A.

### 3.3 Experimental Designs

In order to give a good understanding of the characteristics and behaviours of the selected data fusion algorithms, three phases of experimentation were conducted. This section outlines the setup and procedures for each experiment.

#### 3.3.1 Isolated Data Fusion

Prior to the application of the data fusion algorithms to a collaborative localisation problem, the algorithms were trialled on isolated cases, where two prior uncertainty ellipsoids,  $\mathcal{E}(\hat{x}_i, \hat{P}_i)$  and  $\mathcal{E}(\hat{x}_j, \hat{P}_j)$  were given as input and a single fusion step was conducted for each algorithm.

This isolated fusion experiment was selected as a method to visualise the behaviour of each data fusion algorithm, and allows for an analysis of the performance of the algorithms with regard to the ideal fused estimate characteristics outlined in section 2.5.1.

Additionally, as illustrative examples of fusion algorithms are often given using this method in the data fusion literature, this experimental setup was used to conduct replications of these data fusion examples from literature to ensure the fusion algorithms were correctly implemented.

#### 3.3.2 Collaborative Localisation Problem Formulation

The collaborative localisation problem examined in this report is the problem of estimating a reference position  $p \in \mathbb{R}^n$  collaboratively with  $m$  agents located at positions  $x_i \in \mathbb{R}^n$ ,  $i = 1, \dots, m$ . For the purposes of this report only the 2-dimensional case was considered, using 2 agents, therefore  $n = m = 2$ .

Agent  $i$ , with the use of local measurement data, and communicated estimates from its neighbour agent  $j$  calculates  $\hat{x}_i \in \mathbb{R}^n$  as a point estimate for the reference position  $p$ , along with  $\hat{P}_i \in \mathbb{R}^{n \times n}$  as the estimate of the estimation error covariance  $P_i \in \mathbb{R}^{n \times n}$ .

The following collaborative localisation approach outlined in algorithm 1 that was proposed in [7] was used for the application of data fusion algorithms to collaborative localisation for this report. This algorithm is taken directly from the literature. The following sections outline the definition of the bearing measurement model applied to the specific collaborative localisation

problem, further description of the collaborative localisation algorithm and a procedure for handling non-overlapping ellipsoids in communications and measurements.

### 3.3.2.1 Measurement Model

The measurement model used is the same as that described in section 2.2.1, with the only difference being that each calculation and piece of sensor data now possesses the subscript associated with the relevant agent.

### 3.3.2.2 Collaborative Bearing Localisation Algorithm

---

**Algorithm 1** Collaborative Bearing Localisation

---

**Require:** initial belief  $\mathcal{E}_i(\hat{x}_i, \hat{P}_i), \hat{P}_i > 0$

**Require:** sensor parameters  $\underline{r}_i, \bar{r}_i$  and  $\sigma_i$

**while** estimating reference **do**

Broadcast  $(\hat{x}_i, \hat{P}_i)$  to other nodes on the network

**if** new measurement  $\theta_i$  is received **then**

Calculate a measurement ellipse  $\mathcal{E}_i^m(cm_i, Pm_i)$  (Section 3.3.2.1)

Check if measurement overlaps  $\mathcal{E}_i(\hat{x}_i, \hat{P}_i)$  (Section 3.3.2.3)

**if** there is overlap **then**

Calculate  $\hat{x}_i^+$  and  $\hat{P}_i^+$  using the Kalman Fusion method (Equations (8a-8b))

**else**

Discount the measurement  $Pm_i \leftarrow d_m Pm_i$  (Section 3.3.2.3)

Calculate  $\hat{x}_i^+$  and  $\hat{P}_i^+$  using the Kalman Fusion method (Equations (8a-8b))

**end if**

$(\hat{x}_i, \hat{P}_i) \leftarrow (\hat{x}_i^+, \hat{P}_i^+)$

**else if** communication  $\mathcal{E}_j(\hat{x}_j, \hat{P}_j)$  is received **then**

Check if communication overlaps  $\mathcal{E}_i(\hat{x}_i, \hat{P}_i)$  (Section 3.3.2.3)

**if** there is overlap **then**

Calculate  $\hat{x}_i^+$  and  $\hat{P}_i^+$  using a specified data fusion method

**else**

Discard the communication

**end if**

$(\hat{x}_i, \hat{P}_i) \leftarrow (\hat{x}_i^+, \hat{P}_i^+)$

**end if**

**end while**

---

The Kalman Fusion method is employed in the fusion of the held estimate and the local measurement as the new measurement and the held estimate can be assumed to be independent from each other. This independence allows the Kalman Fusion method to provide the optimal fusion in this instance.

In the step where fusion is called for, in the case where an overlapping communication is received, the algorithm states to calculate the updated fused estimate using a specified fusion method. This was stated in an open way as this is where the application of each of the candidate data fusion algorithms occurs. In the implementation environment an estimate is held by each agent using each fusion method allowing the tracking of performance of multiple data fusion algorithms over the same simulation with the same conditions.

### 3.3.2.3 Dealing with Non-overlapping Cases

An issue that can arise in the application of fusion algorithms that seek to bound the intersection of the prior ellipsoids is that the two ellipsoids provided may not overlap. As stated in [9] the fusion of two ellipsoids is only defined if the intersection exists. This is done to avoid the potential attempts of the algorithms to minimise infinite volumes.

A measure called the Mahalanobis distance can be employed to verify if the intersection exists, shown below in equation (14).

$$d_m(\hat{x}_i, \hat{x}_j) = \|\hat{x}_i - \hat{x}_j\|_{(\hat{P}_i + \hat{P}_j)^{-1}} \quad (14)$$

The intersection is non-zero if the  $d_m \leq 2$  allowing the assertion that these ellipsoids are overlapping.

$$\mathcal{E}_i(\hat{x}_i, \hat{P}_i) \cap \mathcal{E}_j(\hat{x}_j, \hat{P}_j) \neq \emptyset \Leftrightarrow d_m \leq 2 \quad (15)$$

If  $d_m > 2$  however, the ellipsoids are said to be disjoint. In the case of a disjoint measurement ellipsoid, it is proposed by [7] that this measurement be discounted by  $\frac{1}{d_m}$ , as the measurement is assumed to be independent to the held estimate, the measurement could still contain corrective information.

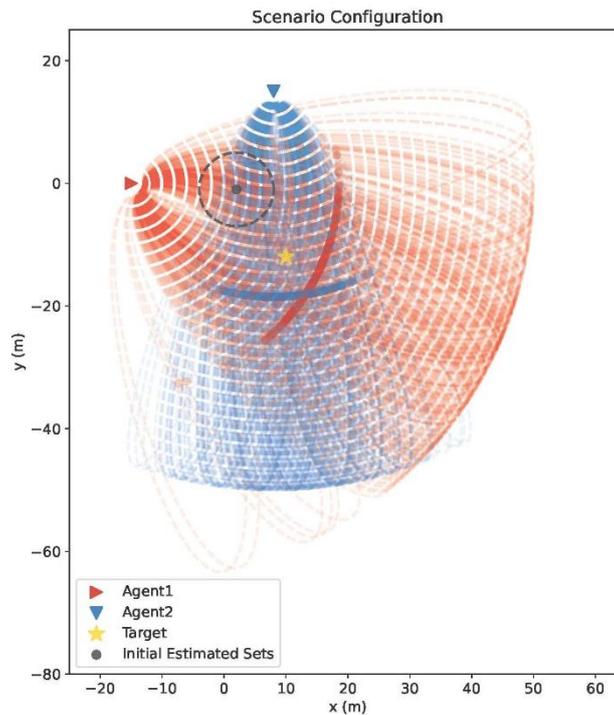
In the case that a communication is received that is disjoint with the held estimate, then this communication should be discarded as there is no evidence that the communication should be favoured over the estimate already held by the agent.

### 3.3.3 Data Fusion in Collaborative Localisation, Plausibility Study

The collaborative localisation algorithm using data fusion algorithms, described in section 3.3.2 was applied in a handful of initial plausibility studies where a singular collaborative localisation scenario was analysed over numerous time steps.

The scenario considered used two stationary agents, collaboratively estimating the location of the reference point in a 2-dimensional plane. For this scenario the agents are intentionally positioned so that neither agent is able to individually make a good localisation.

The scenario configuration depicted in Figure 5 shows the locations of the two agents, the location of the reference point, the shared initial belief, and noisy measurements made by the agents over the simulation.



**Figure 5.** A 2 agent bearing localisation scenario with stationary agents and a stationary reference point. Where the shared initial belief is circular and is centred at  $[2, -1]^T$  and the target reference point is located at  $[10, -12]^T$ . The noisy measurements of each agent are depicted in their respective colour.

Simulations run in this scenario were conducted with the following parameters, the target or reference point is located at  $p = [10, -12]^T$ , agent 1 and agent 2 are located at  $x_1 = [-15, 0]^T$  and  $x_2 = [8, 15]^T$  respectively. The initial belief held by each agent is  $\hat{x}_1 = \hat{x}_2 = [2, -1]^T$ ,  $\hat{P}_1 = \hat{P}_2 = 36I_{2 \times 2}$ . The sensor parameters are  $\underline{r}_1 = \underline{r}_2 = 2$  (m),  $\bar{r}_1 = \bar{r}_2 = 65$  (m),  $\sigma_1 = 14$  (deg), and  $\sigma_2 = 9$  (deg). Lastly the EI regularisation parameter is  $\zeta = 0.0001$

These parameters were held constant across the simulations presented for the singular instance. Two instances were run at 300 steps to allow for an evaluation of the estimation error, covariance, and NEES. An addition instance was run at 1000 steps to evaluate the noise floor of the algorithms.

### 3.3.4 Data Fusion in Collaborative Localisation, Monte Carlo Simulation

Following the assessment of the data fusion algorithms' performances in the singular instance simulation a Monte Carlo simulation was conducted to evaluate the performance of the data fusion algorithms more broadly in the scenario described above. The variables of the scenario were randomised as follows.

Initial estimates were given by  $\hat{x}_1 = p + e_1$  and  $\hat{x}_2 = p + e_2$  where  $e_1$  and  $e_2$  are error vectors, randomised through the use of two random standard deviations  $\zeta_1, \zeta_2 \sim \mathcal{N}(10, 100)$ , defining the error vectors as  $e_1 \sim \mathcal{N}([0,0]^T, [\zeta_1^2, \zeta_1^2]^T)$  and  $e_2 \sim \mathcal{N}([0,0]^T, [\zeta_2^2, \zeta_2^2]^T)$ . The associated initial covariances are given by  $\hat{P}_1 = \zeta_1^2 I_{2 \times 2}$  and  $\hat{P}_2 = \zeta_2^2 I_{2 \times 2}$ .

Sensor characteristics were assigned through the randomisations  $\underline{r}_1, \underline{r}_2 \sim \mathcal{N}(2, 25)$  (m),  $\bar{r}_1, \bar{r}_2 \sim \mathcal{N}(80, 400)$  (m), and  $\sigma_1, \sigma_2 \sim \mathcal{N}(5, 25)$  (deg).

These randomisations were used in [7] and are believed to cover a sufficient range of possibilities that through the use of the Monte Carlo simulation with a sufficient number of instances, broader and more general analysis can be made about the rankings of the performance of the data fusion algorithms in this application.

## Chapter 4 Results and Analysis

This section discusses the results of the simulations conducted. It is divided into four sections: first, the verification of implementation measures conducted, secondly, the results of isolated data fusion cases, thirdly, the results of plausibility study simulations, and finally, the results of Monte Carlo simulation. Within these sections, the performance of each data fusion algorithm method is assessed and compared.

### 4.1 Verification of Implementation

An initial test that was conducted prior to further experimentation was a verification of the implementation environment, through replication of other works. Of main concern was the EI algorithm as this was new in the implementation environment so it was important to verify its correctness. Following this a reverification was done of the Kalman fusion, CI, ICI, and CCE implementations by replicating the results of a single fusion as presented in [7].

#### 4.1.1 Verification of EI

The example provided in the paper [14] for the fused estimates of EI characterises the prior estimates to be fused as  $\hat{x}_i = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}$ ,  $\hat{P}_i = \begin{pmatrix} 2.5 & -1 \\ -1 & 1.2 \end{pmatrix}$ ,  $\hat{x}_j = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ ,  $\hat{P}_j = \begin{pmatrix} 0.8 & -0.5 \\ -0.5 & 4 \end{pmatrix}$  and presents two fused results, using  $\zeta_1 = 10^{-6}$ ,  $\zeta_2 = 0.1$ . The results of the fusion provided by EI and CI in the implementation environment and by the paper [14] for these parameters can be seen in Figure 6.

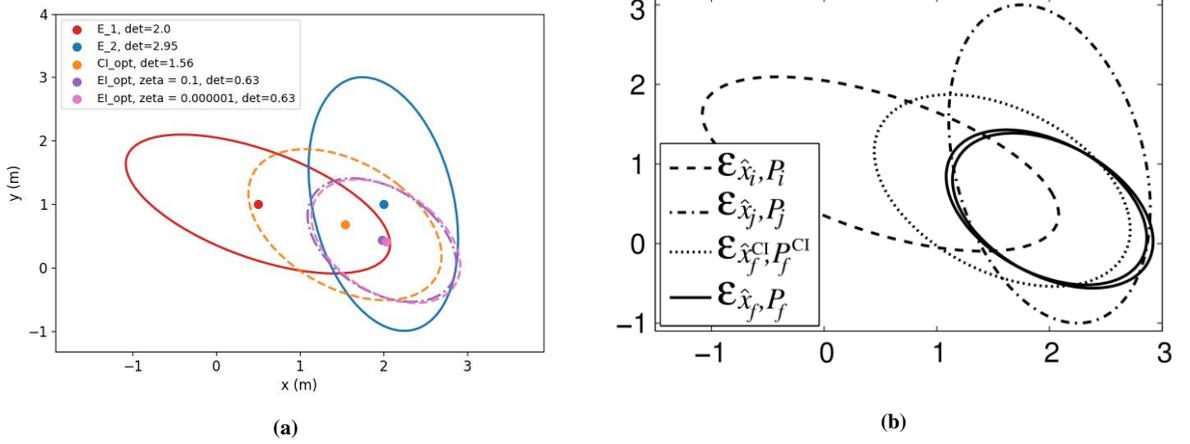


Figure 6. The prior and fused estimates according to EI and CI, with two cases of EI,  $\zeta_1 = 10^{-6}$ ,  $\zeta_2 = 0.1$  (a) shows a replication of (b) sourced from [14].

It can be seen from the replication that the EI algorithm was implemented as was described in the literature. It can also be seen that the value of  $\zeta$  in the EI algorithm does not impact its determinant, this parameter only impacts where the algorithm places the centre or mean of the fused estimate.

It can also be observed that as presented in the literature the EI algorithm is not guaranteed to contain the entire intersection of the prior estimates. This could suggest that the EI algorithm is susceptible to the over confidence problem.

#### 4.1.2 Verification of Kalman Fusion, CI, CCE, and ICI

As a measure to check that the implementation of new material in the environment did not interfere with the algorithms that had been implemented prior to this work a replication was done of the tightness example given by [7]. The prior ellipses were defined by the centres  $\hat{x}_1 = [3, 2]^T$ ,  $\hat{x}_2 = [6.5, 4]^T$  and the uncertainty matrices  $P_1$  and  $P_2$  were defined by the parameters  $wr_1 = 4.4$ ,  $hr_1 = 2.6$ , and  $\theta_1 = -20$  (deg), and  $wr_2 = 4.4$ ,  $hr_2 = 3.2$ , and  $\theta_2 = 90$  (deg).

It can be seen in Figure 7 that the fusion results generated matched those published, and therefore the implementation environment is considered to accurately represent the data fusion algorithms.

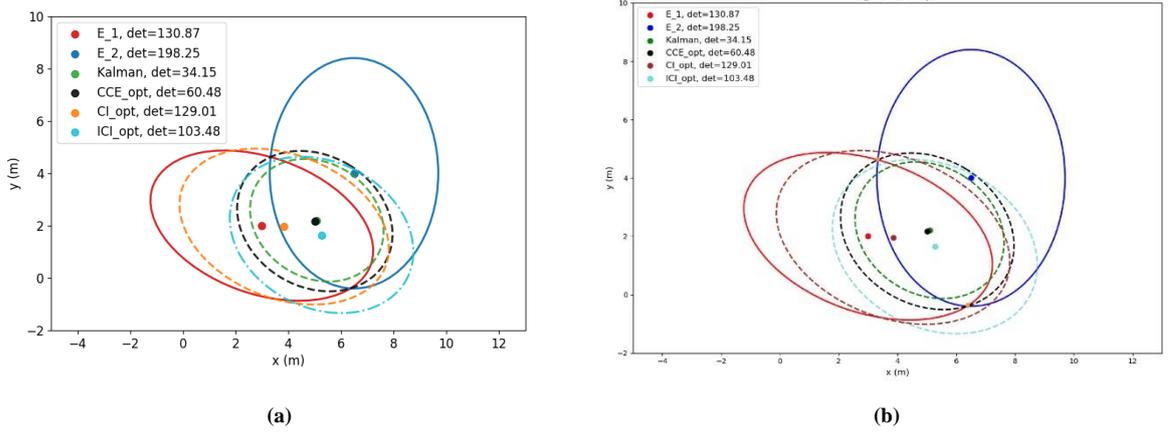


Figure 7. The prior and fused estimates according to Kalman, CCE, CI and ICI (a) shows a replication of (b) sourced from [7]

## 4.2 Isolated Data Fusion

In this section three cases of isolated data fusion are considered; each will be analysed with regard to one of the specific ideal characteristics for fused estimates discussed in section 2.5.1.

The centres of the prior ellipses were held constant across all trials and are defined as,  $\hat{x}_1 = [3, 2]^T$ ,  $\hat{x}_2 = [6.5, 4]^T$ . The uncertainty matrices  $P_1$  and  $P_2$  were defined using the method of ellipsoidal representation discussed in equation (6) with the parameters  $wr$ ,  $hr$ , and  $\theta$ .

The uncertainty matrix  $P_1$  was held constant across all trials, defined with the parameters,  $wr_1 = 5.6$ ,  $hr_1 = 2.7$ , and  $\theta_1 = -30$  (deg). The parameters of  $P_2$  as defined in each trial will be given in their respective sections.

Two values were considered for the parameter  $\zeta$  in the EI method. For simplicity and clarity in discussion of EI the value of  $\zeta$  used for the version of EI discussed will be shown as a subscript. For the isolated fusion trials EI with  $\zeta = 0.1$ ,  $EI_{0.1}$  and EI with  $\zeta = 10^{-6}$ ,  $EI_{10^{-6}}$  were considered.

### 4.2.1 Property 1, Intersection Containment

The first ideal property is that the resulting ellipsoid  $\mathcal{E}_\alpha$  contains the intersection of the two prior ellipsoids,  $\mathcal{E}_1 \cap \mathcal{E}_2 \subseteq \mathcal{E}_\alpha$ . In this trial the matrix  $P_2$  was defined by the parameters  $wr_2 = 3.2$ ,  $hr_2 = 5.6$ , and  $\theta_2 = 60$  (deg).

From Figure 8 a few different behaviours can be seen. CI and ICI have both chosen to disregard  $\mathcal{E}_2$  and wholly favour  $\mathcal{E}_1$ , however, their resulting fusion ellipses both contain the intersection. CCE displays a tighter bounding, containing the intersection. The result that property 1 holds for CI, ICI and CCE can be considered typical of these algorithms.

The Kalman fusion method can be seen to partially contain the intersection however it does not contain its entirety, it can be considered typical of Kalman fusion that property 1 does not hold.

Finally, the EI methods,  $EI_{0.1}$  and  $EI_{10^{-6}}$  show that the EI method does not always contain the intersection, and this property is influenced by the decision of  $\zeta$ , as  $EI_{0.1}$  contains the intersection where  $EI_{10^{-6}}$  does not.

Notably, CI, ICI,  $EI_{0.1}$  and  $EI_{10^{-6}}$  all have the same determinant of the uncertainty matrix. This being 228.61 matching that of  $\mathcal{E}_1$ .

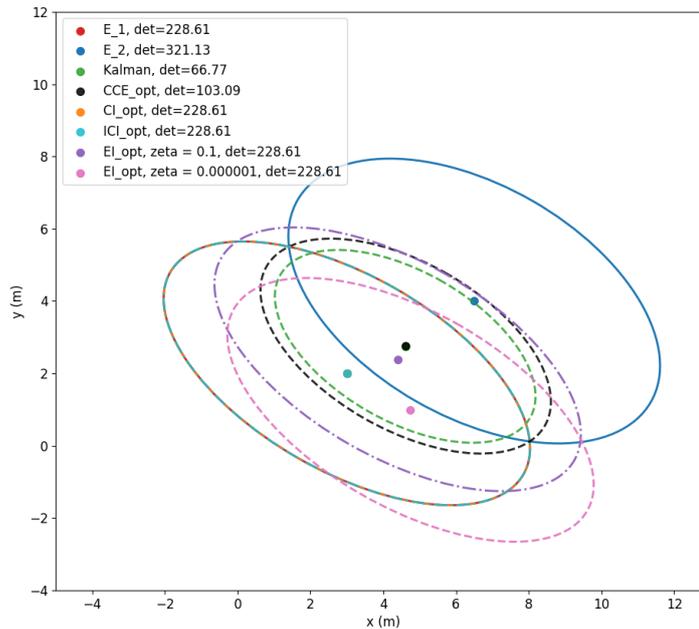


Figure 8. Isolated fusion example demonstrating ideal fusion property 1.

## 4.2.2 Property 2, Boundary Intersections

The second ideal property states that the intersections between the boundaries of  $\mathcal{E}_1$  and  $\mathcal{E}_2$  should lie on the boundary of  $\mathcal{E}_\alpha$ . From Figure 9, where the matrix  $P_2$  was defined by the

parameters  $wr_2 = 5.6$ ,  $hr_2 = 3.2$ , and  $\theta_2 = 60$  (deg), it can be seen that CCE is the only algorithm for which this property holds and this can be considered typical.

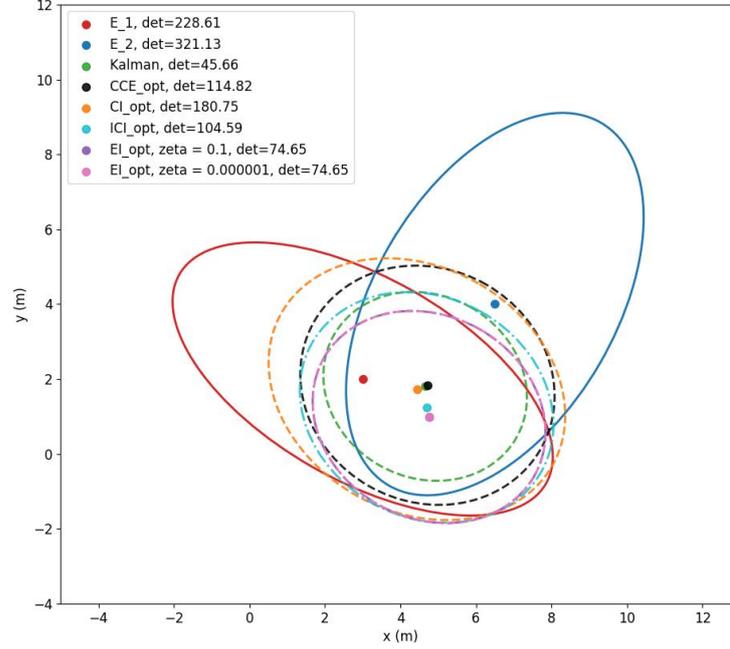


Figure 9. Isolated fusion example demonstrating ideal fusion property 2.

### 4.2.3 Property 3, Containment in the Union

The final desired property is that the ellipsoid  $\mathcal{E}_\alpha$  is contained within the union of the two prior ellipsoids,  $\mathcal{E}_\alpha \subseteq \mathcal{E}_1 \cup \mathcal{E}_2$ . Ensuring that no additional error is introduced through the fusion.

In Figure 10, where the matrix  $P_2$  was defined by the parameters  $wr_2 = 5.6$ ,  $hr_2 = 3.2$ , and  $\theta_2 = -60$  (deg), that this property holds for CCE and Kalman Fusion. However, CI, ICI,  $EI_{0.1}$  and  $EI_{10^{-6}}$  all exist outside of the union of the prior ellipses. While these algorithms do not possess property 3, and can potentially introduce additional uncertainty, they can still be seen in this example to be potentially providing corrective information as the centres of the fused ellipses have been updated in expected manner, moving to be between the centres of the prior estimates.

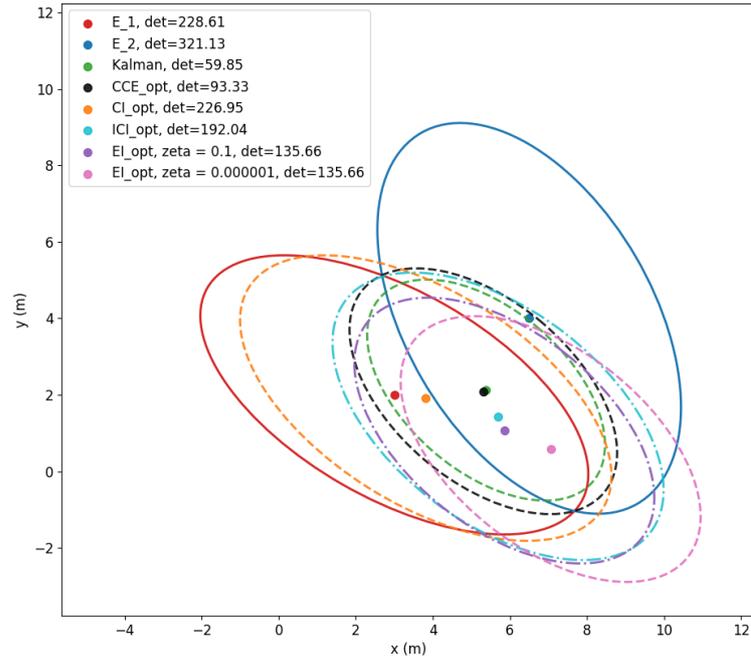


Figure 10. Isolated fusion example demonstrating ideal fusion property 3.

### 4.3 Plausibility Study Simulation

In this section three experiments were conducted according to the scenario configuration outlined in 3.3.3. As stated in 3.3.3, two instances were run at 300 steps to allow for an evaluation of the estimation error, covariance, and NEES. An additional instance was run at 1000 steps to evaluate the noise floor.

From the first instance run at 300 steps, Figures 11, 12 and 13 were generated. In Figure 11 the estimation error at each time step can be seen, for each agent using each method.

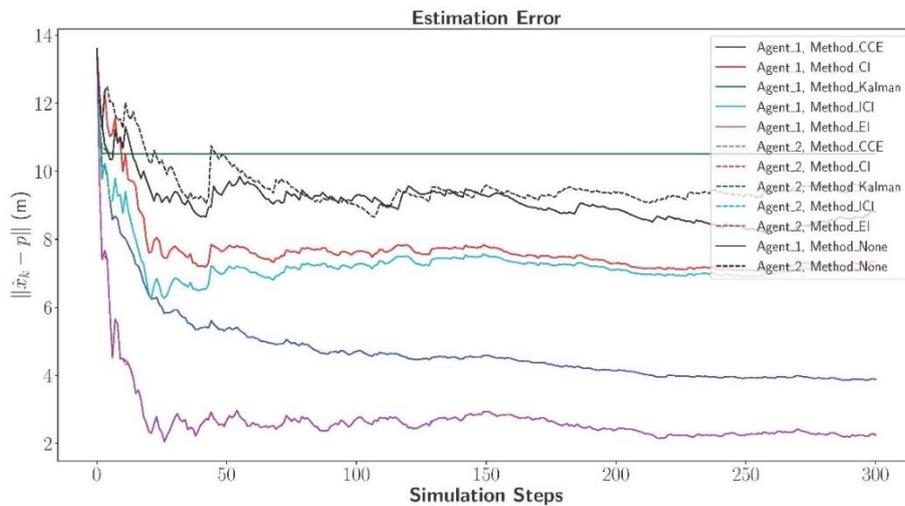
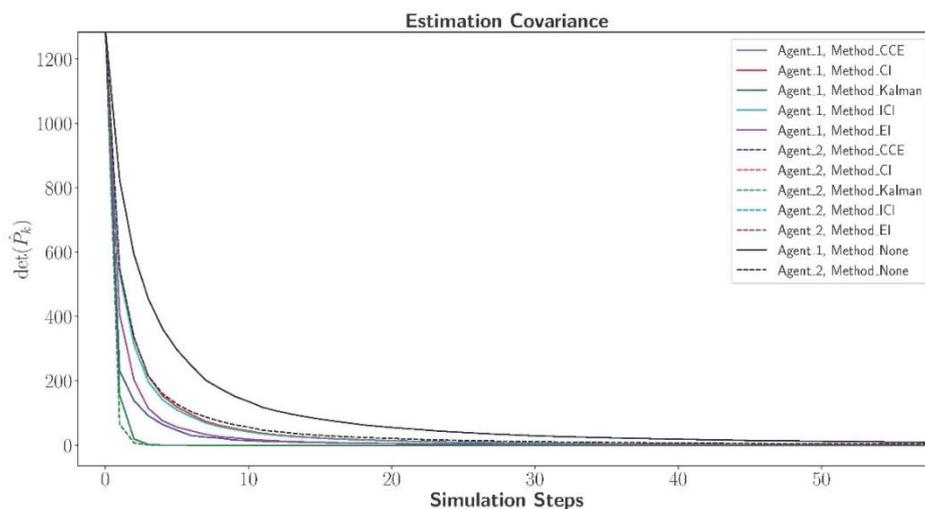


Figure 11. Estimation error over time for the non-collaborative, Kalman, CI, ICI, CCE, and EI methods for each agent.

It can be seen that the EI method performs very well here, with a final estimation error just above 2m, while starting at an initial estimation error of 14m. While CCE follows closely behind achieving a final estimation error of around 4m. The ICI method marginally outperforms the CI method reaching a final error of approximately 7m. The non-collaborative agents, which do not make any communications between each other, reach final errors close to 8m and 9m respectively. Lastly the Kalman fusion quickly becomes overconfident, non-longer improving with new information incredibly early, converging to a final estimation error just above 10m.

The estimation error covariance can be seen over time in Figure 12. This demonstrates the Kalman fusion method's estimation error covariance converges with incredible haste, due to the issue of the independence assumption, as was described in Section 2.4.1. It can also be seen that the CCE method produces the most confident results of the more conservative methods. This is followed closely by EI. It can also be seen that the non-collaborative agents are very slow to improve in their confidence. This is expected as the lack of combined information leads to a worsened localisation with greater uncertainty than collaborative methods.



**Figure 12.** The determinant of the estimated error covariance, tracked over time, for the non-collaborative, Kalman, CI, ICI, CCE, and EI methods for each agent. The figure is zoomed to the first 50 steps of the simulation.

Additionally, the Normalised Estimation Error Squared (NEES) was examined to see the statistical consistency of the algorithms. However, as can be seen in Figure 13, the NEES did not converge to the expected value of 2 for any data fusion method. Yet, it can be seen that the produced information from this test supports the conclusions that can be drawn from the other measures of performance applied in this trial, in regard to the rankings of performance of the data fusion methods.

Additionally, the average NEES (ANEES) was computed at each time step. The ANEES produce results of the same trajectory as NEES however the resulting figure only smoothed the results, providing no additional information.

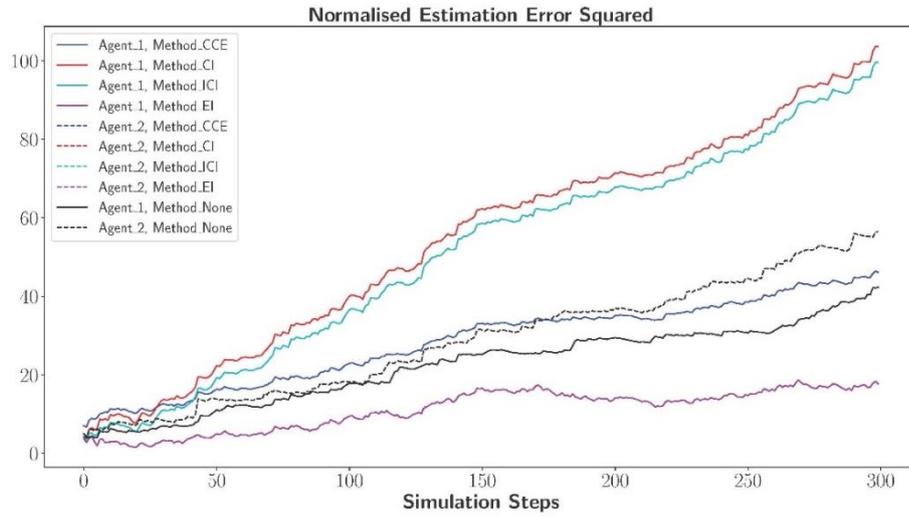


Figure 13. The NEES over time for the non-collaborative, CI, ICI, CCE and EI methods for each agent.

Lastly an addition instance of the 300-step simulation was run, in order to demonstrate the NEES measure applied to the Kalman fusion method. Its error grows excessively, to numbers of such magnitude that other algorithms cannot be visualised on the same figure.

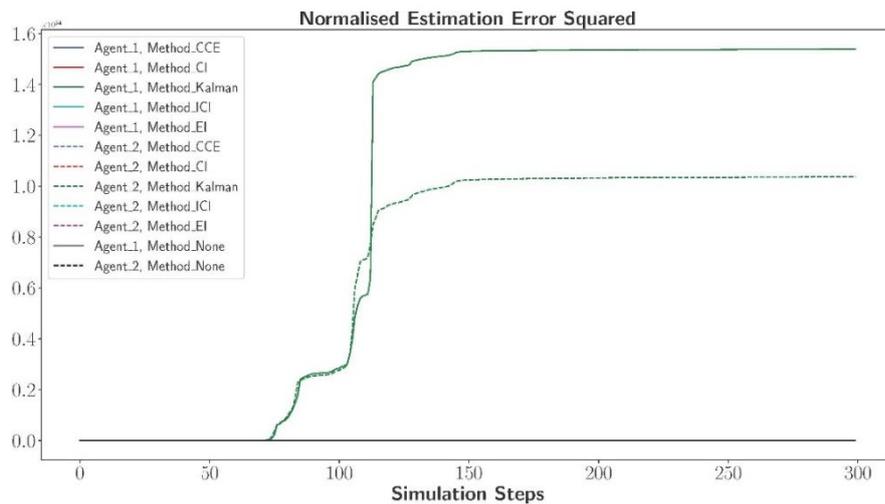


Figure 14. The NEES over time for the non-collaborative, Kalman, CI, ICI, CCE and EI methods for each agent. The Kalman fusion method’s NEES grows to such a magnitude that measures cannot be seen for any other method. The final marking on the y axis is  $1.6 \times 10^{54}$

### 4.3.1 Noise Floor Measurements

As explained in Section 2.5.4 the noise floor represents the level of noise remaining when the algorithm converges to a steady state, for the estimation error. The simulation instance for this test was run to 1000 steps to ensure convergence to the steady state for all algorithms.

The estimation errors tracked over time for this instance can be seen below in Figure 15, and these results show the same ranking of performance that was seen in the 300-step simulation shown in Figure 11.

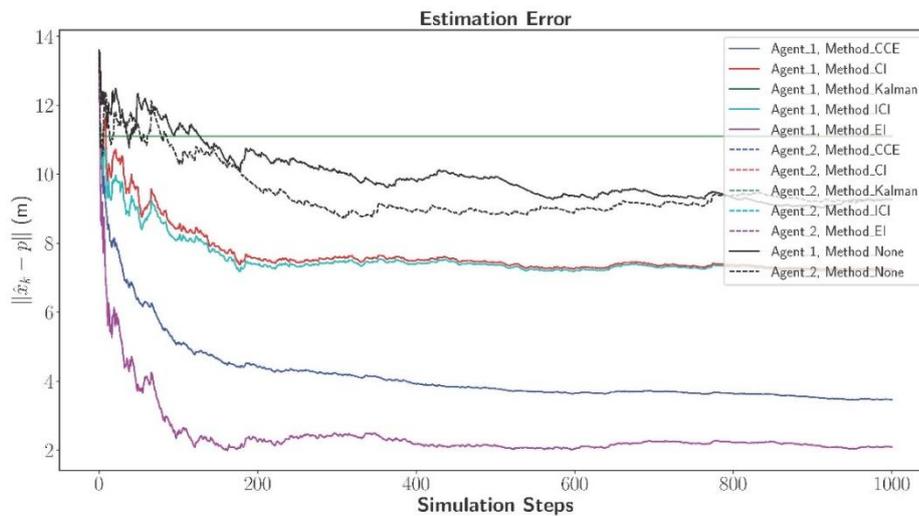


Figure 15. Estimation error over time for the non-collaborative, Kalman, CI, ICI, CCE, and EI methods for each agent.

The convergence of the estimation error was determined by taking the average of the change in the position estimate over the previous 10 steps. The estimation error was said to have converged to the noise floor when this average was under a threshold of 0.015m.

The average change of position over time can be seen in Figure 16. From this two pieces of information can be determined, the noise floor itself, and the number of steps taken to reach the convergence. A summary of these approximate results for agent 1 of each method can be seen in Table 1.

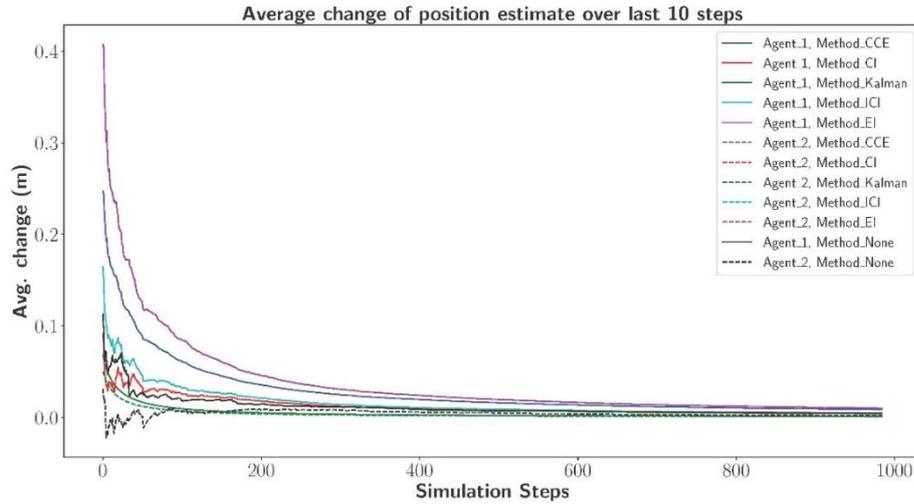


Figure 16. Average change of position estimate over time for the non-collaborative, Kalman, CI, ICI, CCE, and EI methods for each agent

Table 1. Summary of Results of Noise Floor Test

Method	Estimation Error Noise Floor (m)	Steps Taken to Converge
EI	2.10	663
CCE	3.60	554
ICI	7.25	296
CI	7.30	259
Kalman	11.10	20
Non-Collaborative	9.25	1000+

From these results it can be seen that the EI method has the highest capacity to improve its measurements regarding the underlying noise in the scenario, however it took the longest time to reach its noise floor. The CCE method is 1.5m worse but reached its noise floor over 100 steps faster.

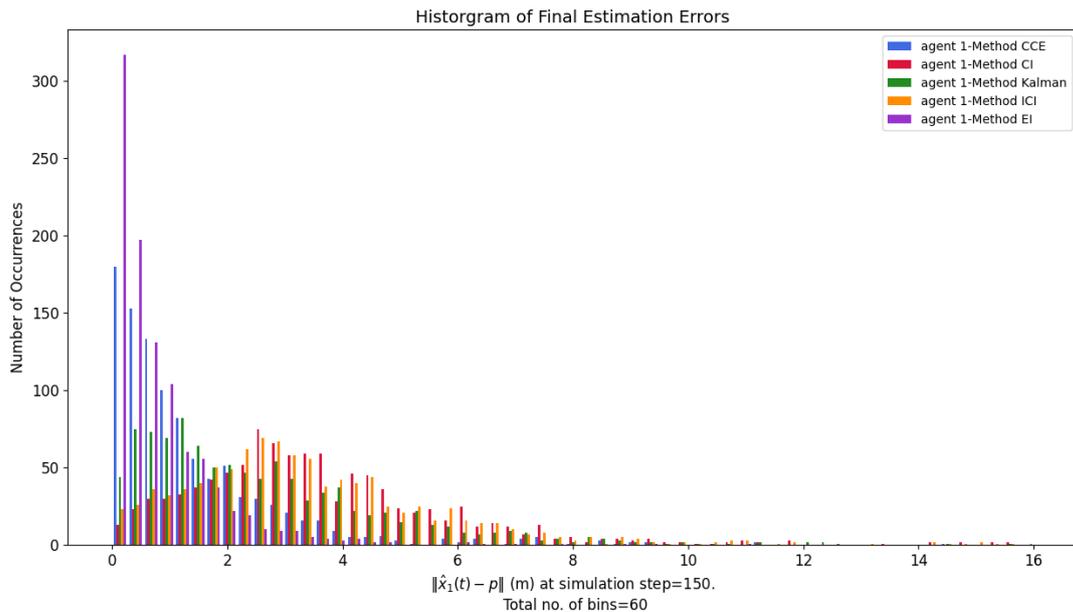
This seems to imply a trade off in the algorithms that in order to achieve high levels of accuracy a large number of recursive steps must be taken. However, it can be seen in both Figure 15 and 16, that the estimation errors reach values close to the noise floor many steps prior to convergence.

## 4.4 Monte Carlo Simulation

The Monte Carlo simulation setup outlined in Section 3.3.4 was conducted for 1000 runs, to a depth of 150 steps per run. From the histogram of final estimation errors, shown in Figure 17. It can be seen in a more general context over the 1000 runs where each method's final estimation errors lie and their relative performance to each other the entire simulation.

The ranked histogram shown in Figure 18 shows performance of the data fusion algorithms in context of each other for each individual run. In order to rank the algorithms in each simulation run, the final estimation errors were taken, and each method was ranked in order, with the minimal estimation error being placed in first position. The histogram represents how many occurrences of each ranking the algorithm received.

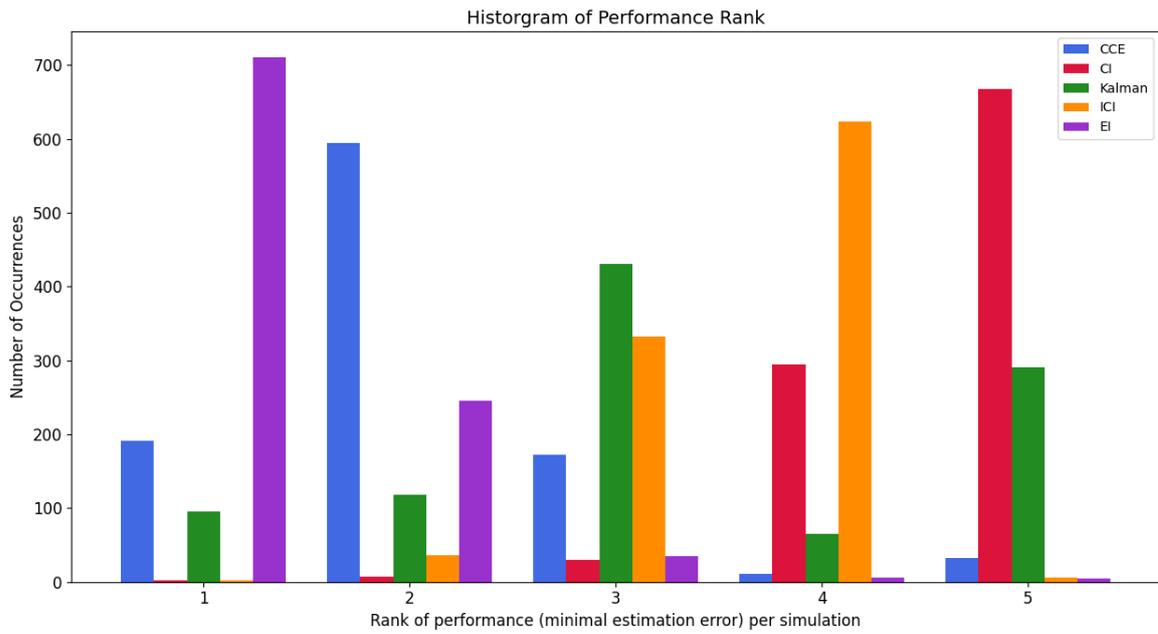
Data from agent 2 has been omitted from these figures to simplify them.



**Figure 17. Histogram of the final estimation error of the Kalman, CI, ICI, CCE, and EI methods at the final fusion step. Run for 1000 simulations to 150 steps.**

It can be seen above in Figure 17 that the EI method performs the best, having the most occurrences of minimal estimation error, with CCE following closely in second.

This is in agreement with Figure 18 seen below where EI is ranked 1<sup>st</sup> in the greatest number of simulation runs, followed by CCE. These results obtained from the Monte Carlo study broadly agree with the performances for each method shown by the measures applied in the plausibility study in section 4.3.



**Figure 18.** Histogram of the rankings of the final estimation error of the Kalman, CI, ICI, CCE, and EI methods in context to each other at the final simulation step. Run for 1000 simulations to 150 steps.

## Chapter 5 Conclusions and Future Work

In this report, several data fusion algorithms were assessed in their application to collaborative bearing localisation algorithms, through a variety of experimental setups, applying several measures of performance.

From the analysis conducted, it can be concluded that there exist data fusion algorithms that provided a suitable level of performance to be used in bearing only collaborative localisation. Data fusion algorithms that showed particular promise for application to the problem discussed in this report are the Ellipsoidal Intersection method and the Convex Combination Ellipsoid method.

A measure of performance that can be useful in the selection of candidate data fusion algorithms, are the ideal characteristics of fused estimates. However, these did not translate well to the performance of the algorithms in application in the case of EI, which consistently showed the best performance in metrics associated with application, while not typically possessing any of these characteristics.

When assessing the performance of data fusion algorithms applied to a collaborative bearing localisation problem, the estimation error and noise floor metrics provide useful information from which to draw comparisons and conclusions. The normalised estimation error squared was not found to be a useful measure of performance for this application, as no algorithm converged to the expected value; however, the NEES supported the conclusions that can be drawn from alternate measures of performance used.

From these metrics EI showed great promise in this application. However further assessment will be needed to capture the full characteristics and potential of EI, as it was seen that changes to  $\zeta$  made a significant difference in the location of the point estimate. The free parameter  $\zeta$  will either need to be more extensively researched to find the optimal tuning for a problem, or an optimisation of some kind could be applied to this parameter to ensure the ideal characteristics of fused estimates are met, ensuring a more reliable performance across applications.

Additionally, EI proved to be initially numerically unstable over recursive fusions. Due to the large amount of explicit calculations, when implemented in the python environment EI suffered a rounding accumulation error which over time degraded the symmetry of the  $\hat{P}^+$  matrix. This required rounding the numerical precision to 10 decimal places to preserve symmetry through recursion.

Further, there is more modern literature on EI that report corrections; these papers may have investigated and corrected these deficiencies and should be looked in to for future work.

Additionally, a different measure of statistical consistency should be investigated for application to this problem, as NEES could not prove statistical consistency for any data fusion algorithm in this scenario.

Lastly, a further line of research on this application would be to investigate the ability to tune the level of communication allowed in the bottom-up approach, potentially employing measures of robustness to the network to see how it can function in the event of additional disturbances.

## References

- [1] R. Siegwart, I.R. Nourbakash, D. Scaramuzza, “Introduction to Autonomous Mobile Robots” 2nd ed., Chapter 5, pp. 265-365, 2011.
- [2] L. C. Carrillo-Arce, E. D. Nerurkar, J. L. Gordillo, and S. I. Roumeliotis, “Decentralized multi-robot cooperative localization using covariance intersection,” *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1412–1417, Nov. 2013. doi:10.1109/iros.2013.6696534
- [3] L. Luft, T. Schubert, S. I. Roumeliotis, and W. Burgard, “Recursive decentralized localization for multi-robot systems with asynchronous pairwise communication,” *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1152–1167, 2018. doi:10.1177/0278364918760698
- [4] M. C. Deans, “Bearings-only localization and mapping,” thesis, Carnegie Mellon University, The Robotics Institute, Pittsburgh, PA, 2005
- [5] P. Zhu and W. Ren, “Fully distributed joint localization and target tracking with Mobile Robot Networks,” *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1519–1532, Jul. 2021. doi:10.1109/tcst.2020.2991126
- [6] S. I. Roumeliotis and G. A. Bekey, “Distributed Multirobot localization,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, Oct. 2002. doi:10.1109/tra.2002.803461
- [7] M. Zamani, J. Trumpf, and C. Manzie, “Collaborative bearing estimation using set membership methods,” *2023 26th International Conference on Information Fusion (FUSION)*, Jun. 2023. doi:10.23919/fusion52260.2023.10224173
- [8] M. A. Bakr and S. Lee, “Distributed multisensor data fusion under unknown correlation and data inconsistency,” *Sensors*, vol. 17, no. 11, p. 2472, Oct. 2017. doi:10.3390/s17112472
- [9] L. Ros, A. Sabater, and F. Thomas, “An ellipsoidal calculus based on propagation and Fusion,” *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 4, pp. 430–442, Aug. 2002. doi:10.1109/tsmcb.2002.1018763
- [10] J. Kim, “Tracking multiple targets using bearing-only measurements in underwater noisy environments,” *Sensors*, vol. 22, no. 15, p. 5512, 2022. doi:10.3390/s22155512
- [11] A. Sabater and F. Thomas, “Set Membership Approach to the Propagation of Uncertain Geometric Information,” in *Proc. of International Conference on Robotics and Automation*, 1991, pp. 2718–2723

- [12] L. Chen, P. O. Arambel, and R. K. Mehra, “Estimation under unknown correlation: Covariance intersection revisited,” *IEEE Transactions on Automatic Control*, vol. 47, no. 11, pp. 1879–1882, Nov. 2002. doi:10.1109/tac.2002.804475
- [13] Z. Wang, X. Shen, and Y. Zhu, “On equivalence of major relaxation methods for minimum ellipsoid covering intersection of ellipsoids,” *Automatica*, vol. 103, pp. 337–345, 2019. doi:10.1016/j.automatica.2019.02.001
- [14] J. Sijs and M. Lazar, “State fusion with unknown correlation: Ellipsoidal intersection,” *Automatica*, vol. 48, no. 8, pp. 1874–1878, 2012. doi:10.1016/j.automatica.2012.05.077
- [15] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck, “Decentralized data fusion with inverse covariance intersection,” *Automatica*, vol. 79, pp. 35–41, 2017. doi:10.1016/j.automatica.2017.01.019
- [16] P. Ivanov, S. Ali-Loytty, and R. Piche, “Evaluating the consistency of estimation,” *International Conference on Localization and GNSS 2014 (ICL-GNSS 2014)*, 2014. doi:10.1109/icl-gnss.2014.6934171
- [17] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. New York: John Wiley & Sons, Inc., 2002.

# Appendix A

The implementation environment was written in Python, and consists of a collection of utility files of functions, which are put together in Jupyter notebooks to construct the experimental environments.

These function files are called `collaborative_bearing_estimation.py`, `Gamma_and_gamma.py`, `Geometry.py`, and `Utilities.py`.

The Jupyter Notebooks used for this report are called `Monte_Carlo_Collaborative_Bearing_Estimation_Demo_3_ICI.ipynb` and `collaborative_bearing_estimation_demo_ICI.ipynb`.

The original contributions to the environment written for this report are all function written in `Gamma_and_gamma.py`, this file is used to calculate the values for the correction terms  $\Gamma$  and  $\gamma$  in the Ellipsoidal Intersection method. Additionally, the `fuse_EI_opt` function in `Utilities.py` in order to compute the Ellipsoidal Intersection method fusion using the parameters calculated.

Further the functions `nees`, `anees`, `anees_moving_frame`, `noiseFloor`, and `avgnoiseFloor` were written in to `collaborative_bearing_estimation.py` in order to compute the NEES and noise floor.

Lastly the functionality to keep track of and rank performance per simulation, producing the ranking histogram of the Monte Carlo simulation was written in to `Monte_Carlo_Collaborative_Bearing_Estimation_Demo_3_ICI`.

The complete code can be found here

[https://github.com/CaitlinLovejoy/Data\\_Fusion\\_and\\_Collaborative\\_Localisation](https://github.com/CaitlinLovejoy/Data_Fusion_and_Collaborative_Localisation)