MOTION PLANNING FOR MOBILE SENSOR NETWORKS

BY

SULEMA EPIGMENIA ARANDA

B.S., University of Illinois at Chicago, 2002

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2004

Urbana, Illinois

Con mucho amor, respeto, y admiracion

A mis padres Maria Castillo-Aranda y Francisco Javier Aranda

A mi esposo Rogelio C. Santana I

y mis hermanas Jasmin C. Avila, Maribel Hernandez y Ana Maria Aranda

ACKNOWLEDGMENTS

This thesis would not have been possible without the contribution and assistance of my advisor Professor Francesco Bullo and Sonia Martínez. This thesis has been a joint effort with them. I would like to thank them both, for all the support, inspiration, teachings and motivation they have provided me with.

I would also like to thank my friends Matthew P. Angert, Mithun Das Gupta, Jon Holm, Greg McRobbie, Yidnekachew S. Mekonnen, Paul Miller, Laura Ruppalt, and Benjamín Tovar for always providing me with inspiration, support, advice, and true friendship during my time here at UIUC. A special thanks to Ben, LaTeX guru, for always coming to my rescue when I had LaTeX questions.

I would like to thank my labmates Anurag Ganguli and Rajan Lakshmi Narasimha for always being there when I had questions. My friends from back home, Oralia and Lucy Ovalle for always believing in me. Special thanks to Professor Goncharoff from the University of Illinois at Chicago, for always providing me with academic advice, support, and direction. A special thank you to the ECE publication office for their helpful comments and suggestions in this manuscript.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CRLB Cramer-Rao lower bound

EKF extended Kalman filter

EIF extended information filter

FIM Fisher information matrix

DEIF decentralized extended information filter

CHAPTER 1

INTRODUCTION

1.1 Background

The deployment of a large number of autonomous vehicles is becoming possible with the advances in development of distributed and decentralized networks as well as electromechanical sensors. The advancements in the field of electromechanical sensors have allowed a number of sensors to become smaller and smaller in dimensions, without affecting the quality of the reading. Having access to these miniature sensors makes it possible to have many sensors on one vehicle, giving it the ability to perform different tasks. In pursuit of having a large number of autonomous vehicles, which can perform different tasks, many interesting problems have been encountered. Some of these problems include, but are not limited to, the following: data fusion; sensor fusion methods, which in turn leads to the motivation of investigating implementations of scalable decentralized estimation; and control algorithms.

The data fusion problem deals with the ability to combine information or knowledge from different sources in order to maximize the usefulness of the information. This might be accomplished by the estimation of specific states of a process or environments, through the combination of data from different or multiple sensors. Sometimes data fusion algorithms are designed as a central process, in which information from all the sensors is sent to one location. As the number of information sources increases, the processing and bandwidth required by the central process may increase dramatically. This in turn will create a bottleneck in pursuit of creating large, centralized data fusion. This may bring the system to fail, which in turn means the system fails as a whole [1].

A method to estimate a desired number of unknown parameters is to collect information/data via sensors called sensor fusion methods. These sensors may be at different locations within an environment and/or in context of this work, on board an autonomous vehicle. Once the data has been collected, it is crucial to be able to process this information in some desired form. An attractive solution for processing the data is via an implementation of a filter proposed by Mutambara. This solution is attractive because it is a scalable decentralized estimation and control algorithm. Mutambara's approach deals with a lot of concepts, each of which will be explained individually in the following. *Scalable* implies that not all available information is needed in order to obtain a result. A decentralized network is usually viewed with its counter part, centralized network. A centralized network is when all the collected information or data is processed by one source. A decentralized network, on the other hand, is when the collected information or data is processed by multiple process. There are advantages and disadvantages for both. One major advantage in the centralized case is that the solution obtained is optimal because it has access to all the information at once. Hence, one major disadvantage is that, when a significant amount of information has to be processed, the potential of creating a bottleneck is high, making the decentralized network attractive. Since the information to be processed is done by multiple process, a bottleneck can be avoided.

1.2 Motivation

In recent years, there has been a great motivation to study sensor networks. For example, detection and localization of vapor-emitting sources [2], unmanned air vehicles [1], and the target tracking problem [3] have been studied. Some of these applications have been motivated by the military, others by search and rescue missions, and yet others for exploratory missions. Take, for example, one of the National Aeronautics and Space Administration (NASA) missions to explore mars. Instead of sending human beings, NASA sent unmanned vehicles like the one in Figure 1.1. According to NASA, the average distance between the Earth and Mars is approximately 78 300 000 km, about half the distance from the Earth to the Sun. Compared to the Moon, 380 000 km away, Mars is about 200 times the distance from Earth, which makes it difficult to send humans, hence leaving the option of sending unmanned vehilces to explore Mars. It is important that the unmanned vehicles sent on the mission have the ability collect and process data. These unmanned robots are designed exactly for that, as shown in Figure 1.1, with all the senors on board the robot. Yet, the unmanned robot is still limited. The unmanned robots, still have not achieved their highest potenial in terms of sensing abilities; therefore, it is important to continue studying sensor networks.

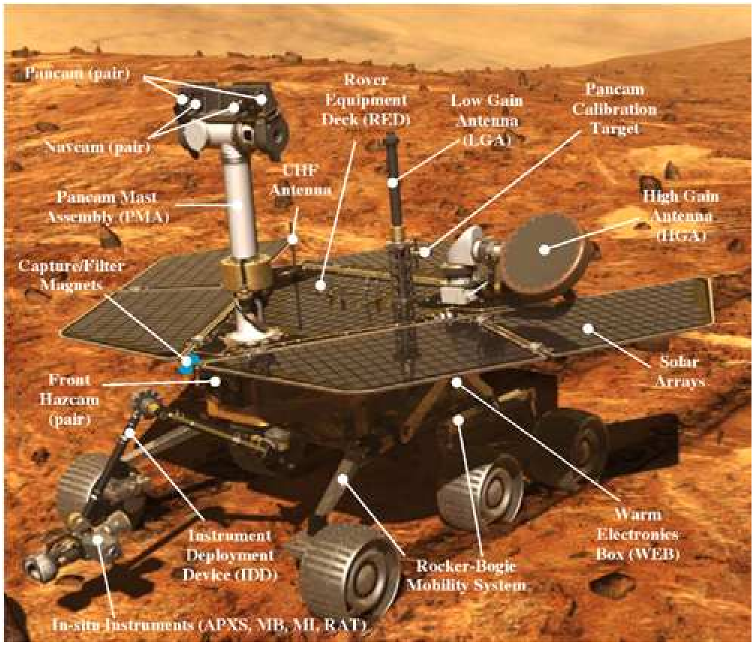
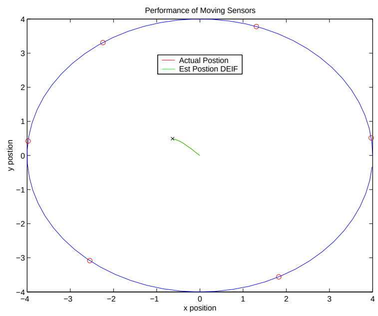


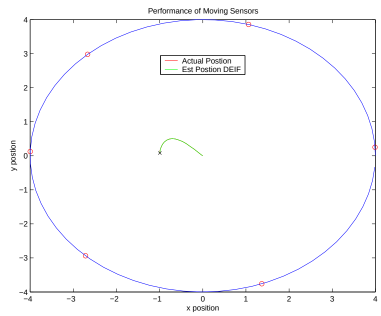
Figure 1.1 The Mars Rover (Courtesy of NASA)

1.3 Objective, Approach, and Contribution

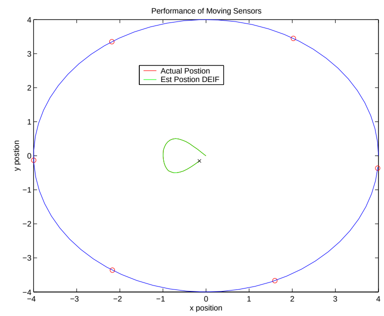
Our objective is to solve practical problems involving a large number of autonomous mobile robots. Figure 1.2 shows a target tracking problem with six mobile robots, solved by the algorithms described in this paper. The proposed technique to best estimate the location of a moving point in a two-dimensional space consists of placing the autonomous vehicles in an optimal location. In order to process the information collected by each robot, Mutambara's decentralized extended information filter, which comes from the extended information filter (EIF), is used. The EIF is an algebraically modified form of the well known extended Kalman filter used to process information/data.



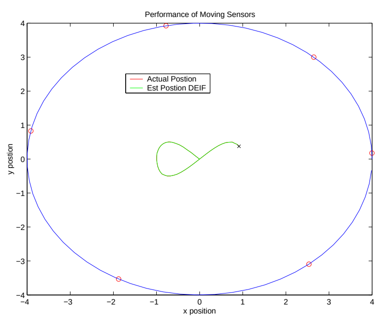
(a) Frame 1



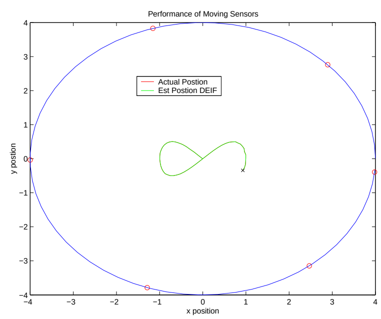
(b) Frame 2



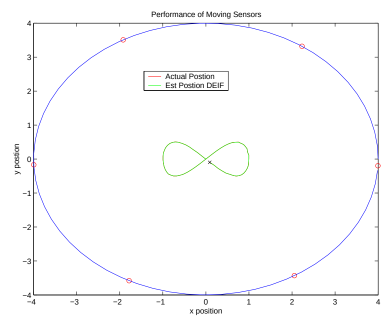
(c) Frame 3



(d) Frame 4



(e) Frame 5



(f) Frame 6

Figure 1.2 Target Localization

CHAPTER 2

DEVELOPMENT OF MOTION PLANNING ALGORITHM

2.1 Motivation

In finding a solution to the problem of target tracking from a multisensor network, it seems clear that the deployment of the agents should maximize the probability of detection of the target to be tracked or provide more accurate estimations of the point source to be localized. On the other hand, a solution to these problems should be built on motion control algorithms for the network and data fusion techniques which allow decentralized implementations.

2.2 Fisher Information Matrix

We now derive the Fisher information matrix for the following situations:

(i) when the source is a (static) non-random parameter,

(ii) when the source is a dynamic random parameter under the influence of white noise.

In both cases the measurements are also perturbed by white noise.

2.2.1 The source/target as a nonrandom parameter

Let $p_j \in \mathbb{R}^n$, $1 \leq j \leq N$, denote the position of N sensors moving on an area $Q \subseteq \mathbb{R}^n$ and let $q_0 \in Q$ be the unknown position of a source or target to be estimated by means of the measurements

$$z_j(q) = f(\|q - p_j\|) + w_j \,, \quad 1 \leq j \leq N \,, \quad q \in Q \,. \tag{2.1}$$

Here we assume that w_j are i.i.d. as $w_j \sim \mathcal{N}(0, \sigma^2)$, $1 \leq j \leq N$, and the function f is defined according to the particular sensors' specifications as

$$f(r) = \begin{cases} c_0, & r \leq R_0 \\ r^\beta + c_1, & R_0 \leq r \leq R_1 \\ R_1^\beta + c_1, & R_1 \leq r, \end{cases}$$

where $c_1 = c_0 - R_0^\beta$ for some $0 < R_0 < R_1$ and $\beta \in \mathbb{N} \cup \{-1\}$.

In other words, the stacked vector of measurements at a certain instant is a random vector normally distributed as

$$Z \equiv \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} f(\|q - p_1\|) \\ \vdots \\ f(\|q - p_N\|) \end{bmatrix}, P \right),$$

where $P = \sigma^2 I_N$ is the covariance matrix defined by I_N, the $N \times N$ identity matrix. From now on, we will use the shorthand notation $Z \equiv (z_1, \ldots, z_N)^T$, $\bar{Z}(q) \equiv (f(\|q - p_1\|), \ldots, f(\|q - p_N\|))^T$.

The *Fisher information matrix* (FIM) J is defined for nonrandom parameters as the following expected value with respect to the conditional probability distribution $p(Z|q)$:

$$J \triangleq E\left[(\nabla_q \log \Lambda(q)) \cdot (\nabla_q \log \Lambda(q))^T \right]_{q = q_0},$$

where q_0 is the true value of the source location or an estimate of it, $\nabla_q = [\frac{\partial}{\partial q^1}, \frac{\partial}{\partial q^2}]^T$, and $\Lambda(q) = p(z_1, \ldots, z_N|q)$ is the *likelihood function* assumed to be

$$\Lambda(q) = \frac{1}{\sqrt{2\pi \det P}} \exp\left(-\frac{1}{2}(Z - \bar{Z})^T P^{-1} (Z - \bar{Z}) \right).$$

In order to compute J, observe that

$$\nabla_q \log \Lambda(q) = -\frac{1}{2} \nabla_q [(Z - \bar{Z})^T P^{-1} (Z - \bar{Z})] = (\nabla_q \bar{Z})^T P^{-1} (Z - \bar{Z}).$$

Then,

$$
\begin{aligned}
J &= E[((\nabla_q \bar{Z})^T P^{-1}(Z-\bar{Z}))\cdot((\nabla_q \bar{Z})^T P^{-1}(Z-\bar{Z}))^T]_{q=q_0}\\
&= E[(\nabla_q \bar{Z})^T P^{-1}(Z-\bar{Z})(Z-\bar{Z})^T P^{-1}(\nabla_q \bar{Z})]_{q=q_0}\\
&= (\nabla_q \bar{Z})^T_{q_0} P^{-1} E[(Z-\bar{Z})(Z-\bar{Z})^T] P^{-1}(\nabla_q \bar{Z})_{q_0}\\
&= (\nabla_q \bar{Z})^T_{q_0} P^{-1}(\nabla_q \bar{Z})_{q_0}\,.
\end{aligned}
$$

In the particular case $P=\sigma^2 I_N$, we have $J=\dfrac{1}{\sigma^2}(\nabla_q \bar{Z})^T_{q_0}(\nabla_q \bar{Z})_{q_0}$. The matrix $G=(\nabla_q \bar{Z})_{q_0}$ is usually called the *sensitivity matrix* associated with the set of measurements.

Taking into account that the position of the source is composed of n directions, we let $q=(q^1,q^2,\dots,q^n)^T$. Then $G\in\mathbb{R}^{N\times n}$ is defined as follows:

$$
G_{ji}=\frac{\partial f_j}{\partial q^i}_{|q=q_0}\,,\qquad f_j \triangleq f(\|z_j-q\|)\,,\quad 1\le j\le N\,,\ 1\le i\le n,
$$

or in matrix format

$$
G=\begin{bmatrix}
\frac{\partial f_1}{\partial q^1} & \dots & \frac{\partial f_1}{\partial q^n}\\
\vdots & & \vdots\\
\frac{\partial f_N}{\partial q^1} & \dots & \frac{\partial f_N}{\partial q^n}
\end{bmatrix}_{q=q_0}.
$$

The Fisher information matrix J can be expressed as

$$
J=\frac{1}{\sigma^2}G^T G=\frac{1}{\sigma^2}\sum_{i=1}^{N}\begin{bmatrix}
(\partial_1 f_i)^2 & \dots & (\partial_1 f_i)(\partial_n f_i)\\
\vdots & \ddots & \vdots\\
(\partial_n f_i)(\partial_1 f_i) & \dots & (\partial_n f_i)^2
\end{bmatrix}, \tag{2.2}
$$

where we denote $\partial_j f_i=\frac{\partial f_i}{\partial q^j}_{|q=q_0}$, $1\le i\le N$, $1\le j\le n$.

2.2.2 The source/target as a (dynamic) random parameter

Suppose a random parameter q is jointly Gaussian with the stacked vector of measurements Z. That is,

$$
\begin{bmatrix} q\\ Z\end{bmatrix}\sim\mathcal{N}\left(\begin{bmatrix}\bar{q}\\ H\bar{q}\end{bmatrix},P\right),
$$

where

$$Z = Hq + w\,, \quad Z = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}\,, \quad H = \begin{bmatrix} H_1, \\ \vdots \\ H_N \end{bmatrix}\,, \quad w = \begin{bmatrix} w_1, \\ \vdots \\ w_N \end{bmatrix}\,, \quad P = \begin{bmatrix} P_{qq} & P_{qZ} \\ P_{Zq} & P_{ZZ} \end{bmatrix}\,,$$

and where we asume that P is invertible, $E[w] = 0$, $E[qw^T] = 0$ and $E[ww^T] = R$.

Then, this next result is obtained from [4], we have that

$$\begin{aligned} P_{qq} &= E[(q - \bar{q})(q - \bar{q})^T]\,, \\ P_{qZ} &= E[(q - \bar{q})(H(q - \bar{q}) + w)^T] = P_{qq}H^T\,, \\ P_{ZZ} &= E[(Z - \bar{Z})(Z - \bar{Z})^T] = E[(H(q - \bar{q}) + w)(H(q - \bar{q}) + w)^T] = HP_{qq}H^T + R\,, \end{aligned}$$

where the expected value is taken with respect to the probability distribution $p(q, Z)$.

The minimimum mean square estimator is given by

$$\hat{q}^{\mathsf{MMSE}} \equiv \hat{q} = E[q|Z] = \bar{q} + P_{qZ}P_{ZZ}^{-1}H(q - \bar{q})\,,$$

and the (conditional) covariance of the error is given by

$$P_{qq|Z} = E[(q - \hat{q})(q - \hat{q})^T|Z] = P_{qq} - P_{qZ}P_{ZZ}^{-1}P_{Zq}\,.$$

On the other hand, the FIM for random parameters, J_R, is defined as the expected value

$$J_R = -E[\nabla_q \nabla_q^T \log p(q, Z)] = E[\nabla_q \log p(q, Z)(\nabla_q \log p(q, Z))^T]_{q=q_0}\,,$$

where q_0 is the true value of the source location or an estimate of it and the expected value is taken with respect to $p(q, Z)$. Under the above assumption, we have that

$$p(q, Z) = \frac{1}{\sqrt{2\pi \det P}} \exp\left(-\frac{1}{2}[(q - \bar{q})^T, (q - \bar{q})^T H^T]P^{-1} \begin{bmatrix} (q - \bar{q}) \\ H(q - \bar{q}) \end{bmatrix}\right).$$

If we denote by

$$P^{-1} = T = \begin{bmatrix} T_{qq} & T_{Zq} \\ T_{qZ} & T_{ZZ} \end{bmatrix},$$

then

$$\nabla_q^T \log p(q,Z) = -\frac{1}{2}\nabla_q \ \big[(q-\bar{q})^T T_{qq}(q-\bar{q}) + (Z-H\bar{q})^T T_{Zq}(q-\bar{q})$$
$$+(q-\bar{q})^T T_{qZ} H(q-\bar{q}) + (q-\bar{q})^T H^T T_{ZZ} H(q-\bar{q})\big] \,.$$

In this way,

$$J_R = T_{qq} = (P_{qq} - P_{qZ}P_{ZZ}^{-1}P_{Zq})^{-1} = (P_{qq|Z})^{-1}\,.$$

It is possible to derive a relationship between the matrix J_R we have just obtained, and the FIM for nonrandom parameters, J_{NR}, of Section 2.2.1. We reproduce it here for the sake of completeness.

Let W denote $W = P_{qZ}P_{ZZ}^{-1}$. Then we can write $P_{qq|Z} = P_{qq} - WP_{ZZ}W^T$. After some manipulations,

$$W = P_{qq}H^T(HP_{qq}H^T + R)^{-1} \quad \Longleftrightarrow \quad W(HP_{qq}H^T + R) = P_{qq}H^T \quad \Longleftrightarrow$$
$$WR = P_{qq}H^T - WHP_{qq}H^T = (I - WH)P_{qq}H^T \quad \Longleftrightarrow \quad W = (I - WH)P_{qq}H^T R^{-1}\,.$$

On the other hand,

$$I - WH = [P_{qq} - WHP_{qq}]P_{qq}^{-1} = [P_{qq} - WP_{ZZ}P_{ZZ}^{-1}HP_{qq}]P_{qq}^{-1}$$
$$= [P_{qq} - WP_{ZZ}W^T]P_{qq}^{-1} = P_{qq|Z}P_{qq}^{-1}\,, \tag{2.3}$$

which in particular implies

$$W = P_{qq|Z}H^T R^{-1}\,. \tag{2.4}$$

Now, using the definition of W we obtain

$$P_{qq|Z} = P_{qq} - WP_{ZZ}W^T = P_{qq} - 2WP_{ZZ}W^T + WP_{ZZ}W^T$$
$$= P_{qq} - P_{qq}H^TW^T - WHP_{qq} + WHP_{qq}H^TW + WRW^T$$
$$= [I - WH]P_{qq}[I - WH]^T + WRW^T\,.$$

Now, using Equations (2.3) and (2.4),

$$P_{qq|Z} = P_{qq|Z} P_{qq}^{-1} P_{qq|Z} + P_{qq|Z} H^T R^{-1} H P_{qq|Z} \; .$$

Finally, pre- and postmultiplying this equation by $P_{qq|Z}^{-1}$, we obtain the expression

$$P_{qq|Z}^{-1} = P_{qq}^{-1} + H^T R^{-1} H \; ,$$

that is,

$$J_R = P_{qq}^{-1} + J_{NR} \; .$$

2.2.2.1 Dynamic random target and Kalman filters

For a dynamic parameter that is modeled as

$$q_k = F_k q_{k-1} + v_k \; ,$$

for which we take measurements

$$Z_k = H_k q_k + w_k \; ,$$

and such that q_k and Z_k are jointly Gaussian distributed, and independent for all $k \geq 1$, we can say:

(i) The FIM is the sum of the information matrices obtained for each step independently:

$$J_R(k) = \sum_{l=1}^{k} J_{R,l} = \sum_{l=1}^{k} P_{qq|Z}^{-1}(l) + H_l^T R^{-1} H_l = \sum_{l=1}^{k} P_{qq|Z}^{-1}(l) + J_{NR,l} = \sum_{l=1}^{k} P_{qq|Z}^{-1}(l) + J_{NR}(k) \; .$$

(ii) What we are going to do in the following is maximize the information of the dynamic filter by maximizing the information of $J_{NR,l}$ $\forall l \geq 1$.

2.3 Optimal Sensor Placement

The FIM defines the *Cramer-Rao lower bound* (CRLB), $J^{-1} = CRLB$, which is known to bound the covariance of the error

$$J^{-1} \leq E[(\hat{q}(z_1, \dots, z_n) - q_0)(\hat{q}(z_1, \dots, z_n) - q_0)^T], \tag{2.5}$$

when the estimator $\hat{q}$ *is unbiased*.[1]

For efficient estimators, this inequality becomes an equality. Then the minimization of the covariance of the error with respect to the sensors' positions is equivalent to the maximization of the FIM. Here, the maximization of a matrix is understood as the maximization of $\det J$. In the following, we compute its particular value for $n = 2$, 3 for the estimation of NONRANDOM parameters.

2.3.1 Two-dimensional configuration space

The determinant of J is found as follows:

$$\begin{aligned}
\sigma^2 \det J &= \left[\textstyle\sum_{i=1}^{N} (\partial_1 f_i)^2\right] \left[\textstyle\sum_{j=1}^{N} (\partial_2 f_j)^2\right] - \left[\textstyle\sum_i (\partial_1 f_i)(\partial_2 f_i)\right]^2 \\
&= \sum_i (\partial_1 f_i)^2 (\partial_2 f_i)^2 + \sum_{i \neq j} (\partial_1 f_i)^2 (\partial_2 f_j)^2 \\
&\quad - \left[\textstyle\sum_i (\partial_1 f_i)^2 (\partial_2 f_i)^2 + \sum_{i \neq j} (\partial_1 f_i)(\partial_2 f_i)(\partial_1 f_j)(\partial_2 f_j)\right] \\
&= \sum_{i \neq j} \left[(\partial_1 f_i)^2 (\partial_2 f_j)^2 - (\partial_1 f_i)(\partial_2 f_i)(\partial_1 f_j)(\partial_2 f_j)\right] \\
&= \sum_{i \leq j} \left[(\partial_1 f_i)^2 (\partial_2 f_j)^2 + (\partial_1 f_j)^2 (\partial_2 f_i)^2\right] - 2 \sum_{i \leq j} (\partial_1 f_i)(\partial_2 f_i)(\partial_1 f_j)(\partial_2 f_j) \\
&= \sum_{i \leq j} \left[\partial_1 f_i \partial_2 f_j - \partial_2 f_i \partial_1 f_j\right]^2 .
\end{aligned}$$

The terms in the last summand can be identified as

$$\sum_{i \leq j} \left[(\mathbf{v}_i \times \mathbf{v}_j) \cdot (0,0,1)\right]^2 = \sum_{i \leq j} \|\mathbf{v}_i \times \mathbf{v}_j\|^2 = \sum_{i \leq j} \|\mathbf{v}_i\|^2 \|\mathbf{v}_j\|^2 \sin^2 \alpha_{ij},$$

where we set $\mathbf{v}_i = (\partial_1 f_i, \partial_2 f_i, 0)$, $\mathbf{v}_j = (\partial_1 f_j, \partial_2 f_j, 0)$. The angle α_{ij} is the one between the vectors $\mathbf{v}_i$ and $\mathbf{v}_j$. The interpretation of $\||\mathbf{v}_i \times \mathbf{v}_j\|$ is the area of the parallelogram formed by $\mathbf{v}_i$ and $\mathbf{v}_j$.

[1] This is true for the type of MMS estimators we work with.

In this way, we have obtained the general expression

$$\det J = \frac{1}{2\sigma^2} \sum_{i,j} \|\mathbf{v}_i\|^2 \|\mathbf{v}_j\|^2 \sin^2 \alpha_{ij} \, . \tag{2.6}$$

Let us develop further this expression for $\det J$ as a function depending on the particular modeling f of our sensors. We have:

$$\partial_1 f_i = \frac{\partial f_i}{\partial q^1}_{|q=q_0} = \begin{cases} \frac{\partial}{\partial q^1} \|p_i - q\|^{\beta}_{|q=q_0} \, , & R_0 \, , \leq \|p_i - q_0\| \leq R_1 \\ 0 \, , & \text{otherwise} \, . \end{cases}$$

$$\frac{\partial}{\partial q^1} \|p_i - q\|^{\beta}_{|q=q_0} = \frac{\partial}{\partial q^1} \left[(p_i^1 - q^1)^2 + (p_i^2 - q^2)^2 \right]^{\frac{\beta}{2}}_{|q=q_0}$$
$$= -\frac{\beta}{2} \left[(p_i^1 - q_0^1)^2 + (p_i^2 - q_0^2)^2 \right]^{\frac{\beta}{2}-1} 2(p_i^1 - q_0^1) = -\beta (p_i^1 - q_0^1) \left[(p_i^1 - q_0^1)^2 + (p_i^2 - q_0^2)^2 \right]^{\frac{\beta}{2}-1} .$$

And analogously,

$$\frac{\partial}{\partial q^2} \|p_i - q\|^{\beta}_{|q=q_0} = -\beta (p_i^2 - q_0^2) \left[(p_i^1 - q_0^1)^2 + (p_i^2 - q_0^2)^2 \right]^{\frac{\beta}{2}-1} .$$

Therefore,

$$(\partial_1 \|p_i - q\|^{\beta}_{q=q_0})^2 + (\partial_2 \|p_i - q\|^{\beta}_{q=q_0})^2 = \beta^2 (p_i^2 - q_0^2)^{2(\beta-2)} \left[(p_i^1 - q_0^1)^2 + (p_i^2 - q_0^2)^2 \right]$$
$$= \beta^2 \|p_i - q_0\|^{2+2\beta-4} = \beta^2 \|p_i - q_0\|^{2(\beta-1)}.$$

In this way, we can write

$$\det J = \frac{1}{2\sigma^2} \sum_{\substack{R_0 < \|p_i - q_0\| < R_1, \\ R_0 < \|p_j - q_0\| < R_1}} \beta^4 \|p_i - q_0\|^{2(\beta-1)} \|p_j - q_0\|^{2(\beta-1)} \sin^2 \alpha_{ij} \, ,$$

where α_{ij} is the angle between the vectors

$$w_i = \beta \|p_i - q_0\|^{2(\beta-1)} (p_i - q_0) \, , \quad w_j = \beta \|p_j - q_0\|^{2(\beta-1)} (p_j - q_0) \, ,$$

which are proportional to $p_i - q_0$ and $p_j - q_0$.

2.3.1.1 Analysis of $\det J$

For $\beta = 1$, $R_0 = 0$, $R_1 = \mathrm{diam}\, Q$, we analyse the maxima of the particular expression of $\det J$:

$$\det J = \frac{1}{2\sigma^2} \sum_{i,j} \sin^2 \alpha_{ij} \; .$$

Let us denote by θ_i the angle of the vector $p_i - q_0$ with the horizontal. Then, $\alpha_{ij} = \theta_i - \theta_j$, $\forall i, j$, and we can write

$$f(\theta_1, \theta_2, \ldots, \theta_N) = 4\sigma^2 \det J = 2 \sum_{i,j} \sin^2(\theta_i - \theta_j) \; . \tag{2.7}$$

2.3.1.2 Critical points

Any critical point of f satisfies

$$\frac{\partial}{\partial \theta_k} \sum_{i,j} \sin^2(\theta_i - \theta_j) = 0 \quad \Longleftrightarrow \quad \sum_i \sin[2(\theta_k - \theta_i)] = 0\,, \qquad k = 1, \ldots, n\,,$$

which is equivalent to

$$\begin{aligned} \sum_i \sin[2(\theta_k - \theta_i)] &= \sin 2\theta_k \sum_i \cos 2\theta_i - \cos 2\theta_k \sum_i \sin 2\theta_i \\ &= \left[(\cos 2\theta_k, \sin 2\theta_k, 0) \times \sum_i (\cos 2\theta_i, \sin 2\theta_i, 0) \right] \cdot \mathbf{e}_3 = 0\,, \quad \forall k. \end{aligned}$$

This implies $\sum_i (\cos 2\theta_i, \sin 2\theta_i, 0) = 0$, or the vectors $(\cos 2\theta_k, \sin 2\theta_k)$ are aligned. That is, a critical point satisfies either

$$\sum_{i=1}^{N} \cos 2\theta_i = 0 \quad \text{or} \quad \sum_{i=1}^{N} \sin 2\theta_i = 0\,.$$

Or the vectors $\{(\cos\theta_k, \sin\theta_k)\}_{k=1}^{N}$ are perpendicular or coincident among them.

2.3.1.3 Tight bounds for f

Let a_{ij} denote $a_{ij} = |\theta_i - \theta_j| = a_{ji}$. For any i, j, k we have

$$\begin{aligned}\sin^2 a_{ik} = (\sin((\theta_i - \theta_j) + (\theta_j - \theta_k)))^2 = (\sin(\theta_i - \theta_j)\cos(\theta_j - \theta_k) + \cos(\theta_i - \theta_j)\sin(\theta_j - \theta_k))^2 \\ = \sin^2 a_{ij}\cos^2 a_{jk} + \cos^2 a_{ij}\sin^2 a_{jk} + \frac{1}{2}\sin[2(\theta_i - \theta_j)]\sin[2(\theta_j - \theta_k)]\,.\end{aligned}$$

Therefore, the critical points of f satisfy the relation

$$\sum_k \sin^2 a_{ik} = \sin^2 a_{ij}\left(\sum_k \cos^2 a_{jk}\right) + \cos^2 a_{ij}\left(\sum_k \sin^2 a_{jk}\right)\,, \qquad \forall i, j\,.$$

In particular, for any i, j such that $\sin^2 a_{ij} \neq 0$, we have

$$\sum_k \sin^2 a_{ik} + \sum_k \sin^2 a_{jk} = \sin^2 a_{ij}\left(\sum_k \cos^2 a_{jk} + \sum_k \cos^2 a_{ik}\right) + \cos^2 a_{ij}\left(\sum_k \sin^2 a_{jk} + \sum_k \sin^2 a_{ik}\right).$$

That is, denoting $X = \sum_k \sin^2 a_{ik} + \sum_k \sin^2 a_{jk}$, we have obtained the relation

$$X = (2N - X)\sin^2 a_{ij} + X\cos^2 a_{ij}\,,$$

which implies $X = N$ when $\sin^2 a_{ij} \neq 0$.

From here, it is clear that if we can establish a bijection $B : \{1, \ldots, N\} \to \{1, \ldots, N\}$, such that $\sin^2 a_{iB(i)} \neq 0$, $\forall i$, then we have

$$\begin{aligned}2\sum_{i,j}\sin^2 a_{ij} = \sum_k \sin^2 a_{1j} + \sum_k \sin^2 a_{B(1)j} + \\ + \sum_k \sin^2 a_{2j} + \sum_k \sin^2 a_{B(2)j} + \cdots + \sum_k \sin^2 a_{Nj} + \sum_k \sin^2 a_{B(N)j} = N^2\,.\end{aligned}$$

Consider all possible maps $M : \{1, \ldots, N\} \to \{1, \ldots, N\}$ such that $\sin^2 a_{iM(i)} \neq 0$. From this finite number of maps there exists one B for which the subset of indices $I = \{i_1, \ldots, i_L\}$, where B is bijective is maximal. In other words, L is the largest cardinal of a subset of indices J where a map M can be bijective.

After a possible reordering of indices, we can assume that $\{i_1, \ldots, i_L\} = \{1, \ldots, L\}$. Let us denote by $B : \{1, \ldots, L\} \to \{B(1), \ldots, B(L)\}$ the restriction of B which is a bijection. Then, because of the symmetry of $\sin^2 a_{ij}$, we can assume that $\{B(1),\dots,B(L)\}=\{1,\dots,L\}$.

Suppose $1\notin\{B(1),\dots,B(L)\}$. Then,

(i) If $B(1)\notin\{1,\dots,L\}$, then we can define a new M such that $M_{|\{1,\dots,L\}}=B$ and $M(B(1))=1$. Since we can assure that $\sin^2 a_{B(1)1}=\sin^2 a_{1B(1)}$, this contradicts the assumption of the largest cardinality L.

(ii) If $B(1)\in\{1,\dots,L\}$, then we can define a new $\bar{B}$ such that $\bar{B}_{|\{1,\dots,L\}\setminus\{B(1)\}}=B_{|\{1,\dots,L\}\setminus\{B(1)\}}$ and $\bar{B}(B(1))=1$.

Consider now a particular index $i\notin\{1,\dots,L\}$. We have that $\sin^2 a_{ij}=0$ for all $j\notin\{1,\dots,L\}$. Otherwise, we can extend B to a map M such that $M(i)=j$, $\sin^2 a_{ij}\neq 0$, which is a contradiction with the maximality condition. Moreover, $\forall l\in\{1,\dots,L\}$ such that $\sin^2 a_{il}\neq 0$, so it must be that

$$\sin^2 a_{iB^{-1}(l)}=0\,,\qquad \sin^2 a_{iB(l)}=0\,.$$

Otherwise, we can define new maps as:

(i)

$$M_{|\{1,\dots,L\}\setminus\{B^{-1}(l)\}}=B\,,\quad M(B^{-1}(l))=i\,,\quad M(i)=l\,,$$

(ii)

$$M_{|\{1,\dots,L\}\setminus\{l\}}=B\,,\quad M(l)=i,\quad M(i)=B(l)\,,$$

both of which violate the condition of maximality of L.

This implies that

$$2\sum_k \sin^2 a_{ik}=2\sum_{k=1}^{L}\sin^2 a_{ik}=(\sin^2 a_{i1}+\sin^2 a_{iB(1)})+\cdots+(\sin^2 a_{iL}+\sin^2 a_{iB(L)})\leq L\leq N\,.$$

Finally, this allows us to conclude that at a critical point,

$$\begin{aligned}f(\theta_1,\dots,\theta_N)=(\sum_k\sin^2 a_{1j}+\sum_k\sin^2 a_{B(1)j})+\cdots+(\sum_k\sin^2 a_{Lj}+\sum_k\sin^2 a_{B(L)j})\\+2\sum_k\sin^2 a_{(L+1)j}+\cdots+2\sum_k\sin^2 a_{Nj}\leq LN+(N-L)N=N^2\,.\end{aligned}$$

In particular, this implies that $f(\theta_1, \ldots, \theta_N) \leq N^2$ for any angle configuration.

2.3.1.4 Some particular global maxima

- Define $\theta_k = \frac{\pi}{N}(k-1)$, $1 \leq k \leq N$. Then, $(\theta_1, \ldots, \theta_N)$ is a maximum for f.

  First, we see that $(\theta_1, \ldots, \theta_N)$ is a critical point. Take $x_0 = \mathrm{e}^{\frac{2\pi i}{N}}$, which satisfies $x_0^N = 1$. In particular, since $x_0 \neq 1$, it must be $p(x_0) = 0$, where

  $$p(x) = x^{N-1} + x^{N-2} + \cdots + x + 1\,, \qquad p(x)(x-1) = x^N - 1\,.$$

  Therefore,

  $$\mathrm{e}^{i\frac{2\pi(N-1)}{N}} + \mathrm{e}^{i\frac{2\pi(N-2)}{N}} + \cdots + \mathrm{e}^{i\frac{2\pi}{N}} + 1 = 0\,,$$

  and this is equivalent to

  $$\sum_{k=1}^{N} \cos(2\frac{\pi}{N}(k-1)) = 0\,, \quad \sum_{k=1}^{N} \sin(2\frac{\pi}{N}(k-1)) = 0\,,$$

  which implies that $(\theta_1, \ldots, \theta_k)$ is a critical point.

  Secondly, it is possible to define a bijection on our set of indices so that $B(i) = i + 1$, $1 \leq i \leq N-1$, $B(N) = 0$ with $\sin^2 a_{iB(i)} = \sin^2(\frac{\pi}{N}) \neq 0$, $N \geq 2$. By the previous reasoning $f(\theta_1, \ldots, \theta_N) = N^2$, and our configuration is a global maximum.

- Following the same arguments as above, it is easy to see that $\theta_k = \frac{l\pi}{N}(k-1)$, $1 \leq k \leq N$, for some $l \in \mathbb{Z}$ is also a critical point and a maximum. In particular, we have that $\theta_k = \frac{2\pi}{N}(k-1)$, $1 \leq k \leq N$, which divides the circle in equal angles $\frac{2\pi}{N}$, and is a maximum for $N \geq 3$.

- Consider an even number of sensors N. Then, the configuration $\theta_{2k+1} = \frac{\pi}{2}$ and $\theta_{2k} = 0$ defines a maximum, too. It is easy to see that it is critical because $(\cos 2\theta_l, \sin 2\theta_l)$ are aligned and we can define a bijection $B(2k) = 2k+1$, $1 \leq k \leq N/2$, $B(N) = 1$, such that $\sin^2 a_{lB(l)} = 1$, $\forall l$.

- Finally, observe that because of the periodicity of $\sin x$ and the symmetry of $\sin^2 x$, given any maximum $(\theta_1, \ldots, \theta_N)$, the set $\{(\theta_1 + k_1\pi, \ldots, \theta_N + k_N\pi) \mid k_1, \ldots, k_N \in \mathbb{Z}\}$.

In view of the first item of the former list and the characterization of the critical points of f, we have the following result:

Proposition 1 *Given $\alpha \in [0, 2\pi)$, define $\theta_k = \frac{\alpha}{N}(k-1)$, $1 \leq k \leq N$. Then, $(\theta_1, \ldots, \theta_N)$ is a critical point of f if and only if $\alpha \in \{\pi l \mid 0 \leq l \leq N-1\}$.*

Proof: Since $\frac{\alpha}{N}(k-1)$ are not all perpendicular or coincident, it must be

$$\sum_{k=1}^{N} \cos(2\frac{\alpha}{N}(k-1)) = 0\,, \qquad \sum_{k=1}^{N} \sin(2\frac{\alpha}{N}(k-1)) = 0\,, \quad \Longleftrightarrow$$
$$\sum_{k=1}^{N} \cos(2\frac{\alpha}{N}(k-1)) + i\sum_{k=1}^{N} \sin(2\frac{\alpha}{N}(k-1)) = 0 \quad \Longleftrightarrow$$
$$\sum_{k=1}^{N} \mathrm{e}^{\frac{2\alpha}{N}(k-1)} = \mathrm{e}^{\frac{2\alpha}{N}(N-1)} + \cdots + \mathrm{e}^{\frac{2\alpha}{N}} + 1 = p(\mathrm{e}^{\frac{2\alpha}{N}}) = 0\,.$$

But the only roots of $p(x)$ are $\{\mathrm{e}^{\frac{2\pi}{N}l} \mid l = 0, \ldots, N-1\}$. ■

A consequence of this proposition is that the subdivision of $[0, \alpha]$ into equal angles does not give a critical point of f restricted to that interval. Moreover, for N sensors and $2\pi > \alpha > \frac{\pi}{N}(N-1)$, we can achieve the global maxima at a configuration where the angles are not equal.

2.3.2 Three-dimensional configuration space

Now, that the two-dimensional configuration space has been understood, it is interesting to analyze the three-dimensional configuration space. This two-dimensional configuration can be used in developing algorithms for objects modeled in the two-dimensional space, for example, vehicles estimating the location of a target in an unknown environment.

Understanding the three-dimensional configuration space is interesting because it occurs naturally in real world applications. For example, an interesting problem to study in the three-dimensional space would be the development of a decentralize three-dimensional configuration target estimation algorithm, using helicopters. Since it follows naturally from the two-dimensional configuration problem, using vehicles.

The determinant of J in (2.2) for $n = 3$ can be computed to be

$$\sigma^2 \det J = \sum_{i,j,k} (\partial_1 f_i)^2 (\partial_2 f_j)^2 (\partial_3 f_k)^2 + \sum_{i,j,k} (\partial_1 f_i)(\partial_2 f_i)(\partial_2 f_j)(\partial_3 f_j)(\partial_3 f_k)(\partial_1 f_k)$$
$$+ \sum_{i,j,k} (\partial_2 f_j)(\partial_1 f_j)(\partial_1 f_i)(\partial_3 f_i)(\partial_3 f_k)(\partial_2 f_k)$$
$$- \sum_{i,j,k} (\partial_1 f_i)(\partial_3 f_i)(\partial_2 f_j)^2 (\partial_3 f_k)(\partial_1 f_k) - \sum_{i,j,k} (\partial_2 f_k)(\partial_3 f_k)(\partial_3 f_j)(\partial_2 f_j)(\partial_1 f_i)^2$$
$$- \sum_{i,j,k} (\partial_2 f_i)(\partial_1 f_i)(\partial_1 f_j)(\partial_2 f_j)(\partial_3 f_k)^2 \, ,$$

which, after regrouping terms, becomes

$$\sigma^2 \det J = \sum_{i,j,k} (\partial_1 f_i)(\partial_2 f_j)(\partial_3 f_k) \left[(\partial_1 f_i)(\partial_2 f_j)(\partial_3 f_k) + (\partial_1 f_k)(\partial_2 f_i)(\partial_3 f_j) + (\partial_1 f_j)(\partial_2 f_k)(\partial_3 f_i) \right.$$
$$\left. -(\partial_1 f_k)(\partial_2 f_j)(\partial_3 f_i) - (\partial_1 f_i)(\partial_2 f_k)(\partial_3 f_j) - (\partial_1 f_j)(\partial_2 f_i)(\partial_3 f_k) \right] \, .$$

This expression reduces to

$$\sigma^2 \det J = \sum_{i,j,k} (\partial_1 f_i)(\partial_2 f_j)(\partial_3 f_k) \left[(\mathbf{v}_i \times \mathbf{v}_j) \cdot \mathbf{v}_k \right] \, ,$$

with

$$\mathbf{v}_i \triangleq (\partial_1 f_i, \partial_2 f_i, \partial_3 f_i) \, , \qquad 1 \leq i \leq N \, .$$

Note that this formula is analogous to that for the two-dimensional configuration space. We can further simplify the determinant as follows:

$$\sum_{i,j,k} (\partial_1 f_i)(\partial_2 f_j) \, (\partial_3 f_k) [(\mathbf{v}_i \times \mathbf{v}_j) \cdot \mathbf{v}_k] =$$
$$= \sum_{i \leq j,k} (\partial_1 f_i)(\partial_2 f_j)(\partial_3 f_k) \left[(\mathbf{v}_i \times \mathbf{v}_j) \cdot \mathbf{v}_k \right] + \sum_{i \geq j,k} (\partial_1 f_i)(\partial_2 f_j)(\partial_3 f_k) \left[(\mathbf{v}_i \times \mathbf{v}_j) \cdot \mathbf{v}_k \right]$$
$$= \sum_{i \leq j,k} (\partial_3 f_k) \left[(\partial_1 f_i)(\partial_2 f_j) - (\partial_1 f_j)(\partial_2 f_i) \right] (\mathbf{v}_i \times \mathbf{v}_j) \cdot \mathbf{v}_k ,$$

this gives

$$\sum_{i\leq j\leq k}(\partial_3 f_k)\left[(\partial_1 f_i)(\partial_2 f_j)\;-(\partial_1 f_j)(\partial_2 f_i)\right](\mathbf{v}_i\times\mathbf{v}_j)\cdot\mathbf{v}_k+\sum_{i\leq j,k\leq j}(\partial_3 f_k)\left[(\partial_1 f_i)(\partial_2 f_j)-(\partial_1 f_j)(\partial_2 f_i)\right](\mathbf{v}_i\times\mathbf{v}_j)\cdot\mathbf{v}_k,$$

where the last term can be expressed as

$$\begin{aligned}
&\sum_{i\leq j,k\leq j}(\partial_3 f_k)\left[(\partial_1 f_i)(\partial_2 f_j)-(\partial_1 f_j)(\partial_2 f_i)\right](\mathbf{v}_i\times\mathbf{v}_j)\cdot\mathbf{v}_k=\\
&\qquad\sum_{k\leq i\leq j}(\partial_3 f_k)\left[(\partial_1 f_i)(\partial_2 f_j)-(\partial_1 f_j)(\partial_2 f_i)\right](\mathbf{v}_i\times\mathbf{v}_j)\cdot\mathbf{v}_k+\\
&\qquad\sum_{i\leq k\leq j}(\partial_3 f_k)\left[(\partial_1 f_i)(\partial_2 f_j)-(\partial_1 f_j)(\partial_2 f_i)\right](\mathbf{v}_i\times\mathbf{v}_j)\cdot\mathbf{v}_k=\\
&\sum_{i\leq j\leq k}(\partial_3 f_i)\left[(\partial_1 f_j)(\partial_2 f_k)-(\partial_1 f_k)(\partial_2 f_j)\right](\mathbf{v}_j\times\mathbf{v}_k)\cdot\mathbf{v}_i+\sum_{i\leq j\leq k}(\partial_3 f_j)\left[(\partial_1 f_i)(\partial_2 f_k)-(\partial_1 f_k)(\partial_2 f_i)\right](\mathbf{v}_i\times\mathbf{v}_k)\cdot\mathbf{v}_j\,.
\end{aligned}$$

Now using that

$$\begin{aligned}
(\mathbf{v}_j\times\mathbf{v}_k)\cdot\mathbf{v}_i&=-(\mathbf{v}_k\times\mathbf{v}_j)\cdot\mathbf{v}_i=-(\mathbf{v}_j\times\mathbf{v}_i)\cdot\mathbf{v}_k=(\mathbf{v}_i\times\mathbf{v}_j)\cdot\mathbf{v}_k\\
(\mathbf{v}_i\times\mathbf{v}_k)\cdot\mathbf{v}_j&=-(\mathbf{v}_k\times\mathbf{v}_i)\cdot\mathbf{v}_j=-(\mathbf{v}_i\times\mathbf{v}_j)\cdot\mathbf{v}_k\,,
\end{aligned}$$

we get

$$\begin{aligned}
\sigma^2\det J=&\sum_{i\leq j\leq k}\big((\partial_3 f_k)\left[(\partial_1 f_i)(\partial_2 f_j)-(\partial_1 f_j)(\partial_2 f_i)\right]\\
&+(\partial_3 f_i)\left[(\partial_1 f_j)(\partial_2 f_k)-(\partial_1 f_k)(\partial_2 f_j)\right]+(\partial_3 f_j)\left[(\partial_1 f_k)(\partial_2 f_i)-(\partial_1 f_i)(\partial_2 f_k)\right]\big)(\mathbf{v}_i\times\mathbf{v}_j)\cdot\mathbf{v}_k,
\end{aligned}$$

which is equivalent to

$$\det J=\frac{1}{\sigma^2}\sum_{i\leq j\leq k}|(\mathbf{v}_i\times\mathbf{v}_j)\cdot\mathbf{v}_k|^2=\frac{1}{\sigma^2}\sum_{i\leq j\leq k}\|\mathbf{v}_i\|^2\|\mathbf{v}_j\|^2\|\mathbf{v}_k\|^2\sin^2\alpha_{ij}\cos^2\beta_{ij,k}\,.$$

This expression is completely analogous to that of (2.6), where now $|(\mathbf{v}_i\times\mathbf{v}_j)\cdot\mathbf{v}_k|$ has the interpretation of the volume generated by the vectors $\mathbf{v}_i$, $\mathbf{v}_j$, and $\mathbf{v}_j$. Here α_{ij} is the angle between $\mathbf{v}_i$ and $\mathbf{v}_j$, and $\beta_{ij,k}$ is the angle between $\mathbf{v}_i\times\mathbf{v}_j$ and $\mathbf{v}_k$.

As in the two-dimensional case, it is easy to see that

$$\det J = \frac{1}{6\,\sigma^2} \sum_{i,j,k} |(\mathbf{v}_i \times \mathbf{v}_j) \cdot \mathbf{v}_k|^2 \,.$$

The conjecture is That for an n-dimensional configuration space,

$$\det J = \frac{1}{\sigma^2} \sum_{i_1 \le i_2 \le \cdots \le i_n} \det(\mathbf{v}_{i_1}, \mathbf{v}_{i_2}, \ldots, \mathbf{v}_{i_n})^2 \,,$$

where $\mathbf{v}_i = (\partial_1 f_i, \ldots, \partial_n f_n)$, $1 \le i \le n$.

In particular, for the 3-D case, and our particular sensing functions, we have

$$\det J = \frac{1}{6\,\sigma^2} \sum_{\begin{bmatrix} R_0 \,<\, \|p_i - q_0\| \,<\, R_1 \\ R_0 \,<\, \|p_j - q_0\| \,<\, R_1 \\ R_0 \,<\, \|p_k - q_0\| \,<\, R_1 \end{bmatrix}} \beta^6 \|p_i - q_0\|^{2(\beta-1)} \|p_j - q_0\|^{2(\beta-1)} \|p_k - q_0\|^{2(\beta-1)} \sin^2 \alpha_{ij} \cos^2 \beta_{ij,k} \,.$$

For $\beta = 1$, this reduces to

$$\det J = \frac{1}{6\,\sigma^2} \sum_{\begin{bmatrix} R_0 \,<\, \|p_i - q_0\| \,<\, R_1 \\ R_0 \,<\, \|p_j - q_0\| \,<\, R_1 \\ R_0 \,<\, \|p_k - q_0\| \,<\, R_1 \end{bmatrix}} \sin^2 \alpha_{ij} \cos^2 \beta_{ij,k} \,.$$

2.3.2.1 Analysis of $\det J$

For the particular cases $\beta = 1$, $R_0 = 0$, $R_1 = +\infty$, we have

$$f(\mathbf{v}_1, \ldots, \mathbf{v}_N) = 6\,\sigma^2 \det J = \sum_{i,j,k} |(\mathbf{v}_i \times \mathbf{v}_j) \cdot \mathbf{v}_k|^2 \,.$$

In the following we analyze the critical points and global maxima of f.

2.3.2.2 Critical points

The function f has been defined on points of the sphere $\mathbb{S}^2 \subseteq \mathbb{R}^3$. That is, $f : \mathbb{S}^{6N} \to \mathbb{R}$. Therefore, the critical points satisfy

$$\mathcal{L}_{\mathbf{w}_k} f = 0\,, \quad \forall \mathbf{w}_k \in T_{\mathbf{v}_k}\mathbb{S}^2 \quad \Longleftrightarrow \quad \sum_{i,j} (\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k)(\mathbf{v}_i \times \mathbf{v}_j) \cdot \mathbf{w}_k = 0\,, \quad \forall \mathbf{w}_k \perp \mathbf{v}_k\,,$$

where we have used the notation $(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k) \equiv (\mathbf{v}_i \times \mathbf{v}_i) \cdot \mathbf{v}_k$.

In other words,

$$\mathbf{v}_k^T \left[\sum_{i,j} (\mathbf{v}_i \times \mathbf{v}_j)(\mathbf{v}_i \times \mathbf{v}_j)^T \right] \mathbf{w}_k = 0\,, \qquad \forall \mathbf{w}_k \perp \mathbf{v}_k\,.$$

If we denote $\Lambda = \sum_{i,j} \Lambda_{i,j} = \sum_{i,j} (\mathbf{v}_i \times \mathbf{v}_j)(\mathbf{v}_i \times \mathbf{v}_j)^T$, then $(\mathbf{v}_1, \ldots, \mathbf{v}_N)$ is a critical point if and only if

$$\mathbf{v}_k^T \Lambda \mathbf{w}_k = 0\,, \qquad \forall \mathbf{w}_k \perp \mathbf{v}_k\,, \quad \forall k\,, \quad \text{where } \Lambda = \Lambda^T \geq 0\,.$$

This leads to the following characterization:

Lemma 2 $(\mathbf{v}_1, \ldots, \mathbf{v}_N)$ *is a critical point of* f *if and only if* $\mathbf{v}_k$ *is an eigenvector of* Λ $\forall k$*.*

Proof: Clearly, if $\mathbf{v}_k$ is an eigenvector of Λ, then

$$\mathbf{v}_k^T \Lambda \mathbf{w}_k = \lambda \mathbf{v}_k^T \mathbf{w}_k = 0\,, \qquad \forall \mathbf{w}_k \perp \mathbf{v}_k\,, \quad \forall k\,.$$

On the other hand, consider $\mathbf{v}_k$ such that the above equation is verified. Since $\Lambda = \Lambda^T$, there exists a basis of orthonormal vectors $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ which are eigenvectors of Λ with corresponding eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$. In this basis, we can express $\mathbf{v}_k = \mu_1 \mathbf{u}_1 + \mu_2 \mathbf{u}_2 + \mu_3 \mathbf{u}_3$ for some $\mu_i \in \mathbb{R}$, $i = 1, 2, 3$. Now define

$$\mathbf{w}_k^{12} = -\mu_2 \mathbf{u}_1 + \mu_1 \mathbf{u}_2\,, \qquad \mathbf{w}_k^{23} = -\mu_3 \mathbf{u}_2 + \mu_2 \mathbf{u}_3\,, \qquad \mathbf{w}_k^{13} = -\mu_3 \mathbf{u}_1 + \mu_1 \mathbf{u}_3$$

Since $\mathbf{v}_k \perp \mathbf{w}_k^{12}, \mathbf{w}_k^{13}, \mathbf{w}_k^{23}$, then

$$\begin{aligned}
\mathbf{v}_k^T \Lambda \mathbf{w}_k^{12} &= -\lambda_1 \mu_1 \mu_2 + \lambda_2 \mu_1 \mu_2 = 0\,, \\
\mathbf{v}_k^T \Lambda \mathbf{w}_k^{23} &= -\lambda_2 \mu_2 \mu_3 + \lambda_3 \mu_3 \mu_2 = 0\,, \\
\mathbf{v}_k^T \Lambda \mathbf{w}_k^{13} &= -\lambda_1 \mu_1 \mu_3 + \lambda_3 \mu_3 \mu_1 = 0\,.
\end{aligned}$$

If $\mu_i = \mu_j = 0$ for $i, j \in \{1, 2, 3\}$, then $\mathbf{v}_k$ is proportional to an eigenvector $\mathbf{u}_l$ and is itself an eigenvector.

If $\exists\, \mu_i, \mu_j \neq 0$ and the third component $\mu_l = 0$, then from the equation $\mathbf{v}_k \Lambda \mathbf{w}_k^{ij} = 0$ we have $\lambda_1 = \lambda_2$ and $\mathbf{v}_k$ is an eigenvector with eigenvalue λ_1. If $\mu_i \neq 0$ for all i, a similar argument leads to $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$ and again, $\mathbf{v}_k$ is an eigenvector associated with λ. ■

CHAPTER 3

DECENTRALIZED CONTROL LAW

3.1 Introduction

It was proved in the previous section that, in order to achieve the global maxima of $f(\theta_1, \cdots, \theta_N)$ defined by Equation (2.7), N number of sensors have to be in a certain configuration summarized by Proposition 1. Therefore, finding the global maxima of the FIM implies obtaining the best estimate possible of a stationary target's location. Then, the objective is to develop a decentralized control law such that

$$\theta_k = \frac{2\pi}{N}(k-1) \qquad \text{for} \quad 1 \leq k \leq N. \tag{3.1}$$

which divides the circle into equal angles of $\frac{2\pi}{N}$ for $N \in \mathbb{N}$. There are two possible control laws that divides the circle into equal angles *Go Towards Midpoint* and *Go Towards Center of Voronoi Cell*. These two laws will be defined and proved in the following sections.

3.2 Algorithm for Control Laws

3.2.1 Law 1: Go towards midpoint

The first control law is defined by

$$\begin{cases} p_i' = \frac{p_{i+1}+p_{i-1}}{2}, \\ \theta_i' = \frac{\theta_{i+1}+\theta_{i-1}}{2} \end{cases} \qquad \text{for} \quad 1 \leq i \leq N, \tag{3.2}$$

such that

$$i+1 \neq i-1, \quad \theta_i = \measuredangle(p_i, p_{i-1}), \quad \sum_{i=1}^{N} \theta_i^k = 2\pi \quad \forall k. \tag{3.3}$$

To get a geometric representation of the angles θ_i, refer to Figure 3.1.

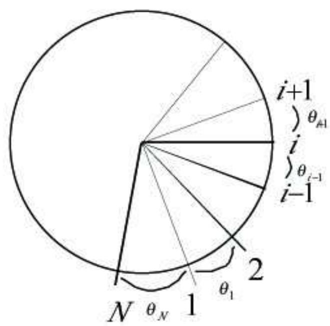


Figure 3.1 Definition of Angles

Under the restrictions described by the set of Equations (3.3), a point on a circle can be represented locally on a line. The goal is to divide the circle into equal angles of $\frac{2\pi}{N}$. This is acheived by using the points (p_i, p_{i-1}) on a line, which describe the angle θ_i. In order to obtain a better intuition on how the angle θ_i can be described by the points (p_i, p_{i-1}), refer to Figure 3.2.

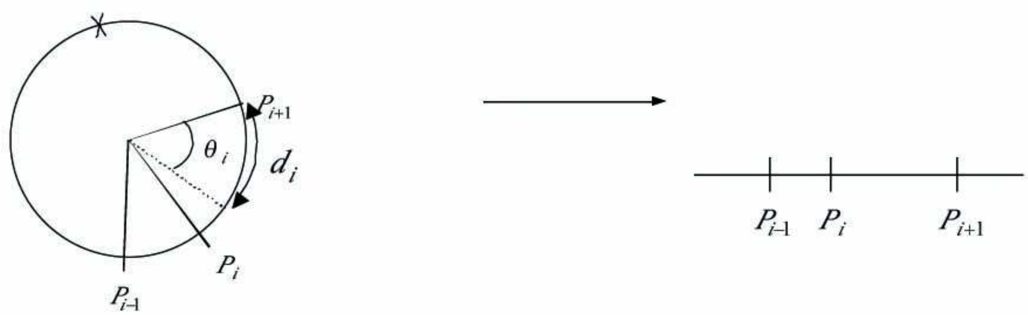


Figure 3.2 Control Law 1

Therefore, in order to divide the circle into equal angles of θ_i where $i = \{1, 2, \cdots, N\}$, the distance between $\{p_i, p_{i-1}\}$ and $\{p_i, p_{i+1}\}$ must be equal. Hence,

$$|p_i - p_{i-1}| = |p_{i+1} - p_i|$$
$$|p_i - p_{i-1}| - |p_{i+1} - p_i| = 0.$$

Since $p_{i-1} < p_{i-1} < p_{i+1}$ holds, it follows that

$$p_i - p_{i-1} - p_{i+1} + p_i = 0$$
$$2p_i - p_{i-1} - p_{i+1} = 0.$$

Therefore, solving for p_i gives

$$p_i = \frac{p_{i-1} + p_{i+1}}{2}, \tag{3.4}$$

which gives the new location of p_i and will be denoted as $p_i^{'}$.

The question that arises now, is: How does a point p_i on a line map to an angle θ_i? Notice that

$$d_i = \theta_i R$$
$$2\pi \to L = 2\pi R \quad and \quad \theta \to d.$$

Therefore,

$$d = \frac{\theta 2\pi R}{2\pi} = \theta R.$$

Also, when, $R = 1$, we have $d_i = \theta_i$. So to get equal angles, we just take

$$\theta_i^{'} = \frac{\theta_{i+1} + \theta_{i-1}}{2}, \tag{3.5}$$

where $\theta_i^{'}$ denotes the new angle of θ_i after the calculation has been done.

3.2.2 Law 2: Go towards the center of Voronoi cell

The second control law is defined by

$$\begin{cases} p_i^{'} = \frac{1}{4}\left[p_{i-1} \ + \ 2p_i \ + \ p_{i+1}\right] \\ \theta_i^{'} = \frac{1}{4}\left[\theta_{i-1} + 2\theta_i + \theta_{i+1}\right] \end{cases} \qquad \text{for} \quad 1 \leq i \leq N. \tag{3.6}$$

Using the restriction described by Equation (3.3), a point on a circle can be represented locally on a line. Since the goal is to divide the circle into equal angles of $\frac{2\pi}{N}$, it will be acheived once again by using the points $\{p_i, p_{i-1}\}$ which describe the angle θ_i, as seen in Figure 3.3. This approach is slightly different from the first control law, since Voronoi partitions are being used to divide the space into equal parts.

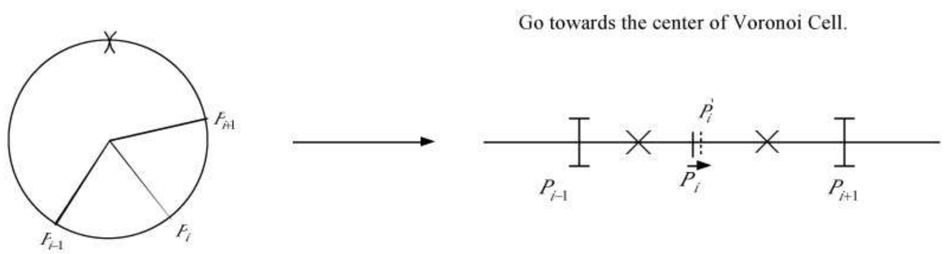


Figure 3.3 The Represention of Points on a Circle to Points on a Line

So only viewing $\{p_{i-1}, p_{i+1}\}$ as neighbors, the center of the voronoi partition is described as

$$p_i^{'} = \frac{1}{2}\left[\frac{p_{i-1} \ + \ p_i}{2} + \frac{p_i \ + \ p_{i+1}}{2}\right]. \tag{3.7}$$

Simplifying Equation (3.7) gives

$$\begin{aligned} p_i^{'} =& \frac{1}{2}\left[\tfrac{p_{i-1}}{2} + \tfrac{p_i}{2} + \tfrac{p_i}{2} + \tfrac{p_{i+1}}{2}\right] \\ =& \frac{1}{2}\left[\tfrac{p_{i-1}}{2} + 2\tfrac{p_i}{2} + \tfrac{p_{i+1}}{2}\right] \\ =& \frac{1}{4}\left[p_{i-1} + 2p_i + p_{i+1}\right]. \end{aligned} \tag{3.8}$$

Following the same logic, to find $\theta_i^{'}$, as for control law 1, it follows that

$$\theta_i^{'} = \frac{1}{4}\left[\theta_{i-1} + 2\theta_i + \theta_{i+1}\right].$$

3.3 Convergence of Algorithms

Now that the algorithm have been defined, it is desirable to understand how each algorithm behaves as time progresses. Therefore, in the following section the first and second control laws will be analyzed.

3.3.1 Convergence of Control Law 1

It is important to model the evolution of all the angles $\{\theta_1(k), \theta_2(k), \cdots, \theta_N(k)\}$ are described by $\Theta(k)$ where

$$\Theta(k) = \left[\theta_1(k), \theta_2(k), \cdots, \theta_N(k)\right]^T \qquad \forall\ k \geq 0, \tag{3.9}$$

such that

$$\sum_{i=1}^{N} \theta_i(k) = 2\pi \qquad \forall\ k.$$

The evolution of the states $\{\theta_1(k), \theta_2(k), \cdots, \theta_N(k)\}$ are described by

$$\Theta(k+1) = B_i\Theta(k), \tag{3.10}$$

where the transition matrix B_1 is defined by

$$B_1 = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \cdots & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & \cdots & 0 \\ \vdots & \ddots & \ddots & \cdots & \vdots \\ 0 & \cdots & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \cdots & 0 & \frac{1}{2} & 0 \end{bmatrix}. \tag{3.11}$$

There exist a matrix called the basic circulant matrix and this matrix possesses many attractive properties. It is important to note that B_1 can be broken up into the basic circulant matrix C,

$$B_1 = \frac{1}{2}C + \frac{1}{2}C^{N-1}$$

where C is known as

$$C = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}_{NxN} .$$

3.3.1.1 Eigenvalues of Control Law 1

Analyzing the eigenvalues of the control law 1, gives insight on the convergence of the system as time progresses. By analyzing the eigenvalues of control law 1, it will be shown in this section that control law 1 has two oscillating points. Therefore, since the control law 1 has two oscillation points, it is important to modify it, in order to obtain a converging algorithm. The eigenvalues are given by $p(\lambda) = \begin{vmatrix} \lambda I - C \end{vmatrix}$, where

$$p(\lambda) = \begin{vmatrix} \lambda & -1 & \cdots & 0 \\ 0 & \ddots & \cdots & \vdots \\ \vdots & \ddots & \ddots & -1 \\ -1 & \cdots & 0 & \lambda \end{vmatrix}.$$

Since

$$\begin{vmatrix} \lambda & -1 & \cdots & 0 \\ 0 & \ddots & \cdots & \vdots \\ \vdots & \ddots & \ddots & -1 \\ -1 & \cdots & 0 & \lambda \end{vmatrix}$$

$$= \lambda \begin{vmatrix} \lambda & -1 & \cdots & 0 \\ 0 & \ddots & \cdots & \vdots \\ \vdots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & \lambda \end{vmatrix} + (-1)(-1)^{N+1} \begin{vmatrix} -1 & 0 & \cdots & 0 \\ \cdot & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \cdot & \cdot & \cdot & -1 \end{vmatrix} =$$

$$= \lambda\lambda^{N-1} + (-1)^N(-1)^{N-1} = \lambda^N - 1.$$

It follows that the, eigenvalues can be defined in a different form as

$$p(\lambda) = \lambda^N - 1 = 0 \quad \Leftrightarrow \quad \lambda \in \{e^{\frac{2\pi ik}{N}} \mid 1 \leq k \leq N\}.$$

One of the properties that the basic circulant matrix C posses is that it is diagonalizable in $\mathbb{C}$. Being a diagonalizable matrix implies the existence of a basis of the eigenvectors, which are orthogonal.

With C possessing the property of being diagonalizable, and also since B_1 can be defined by C, then the eigenvectors of B_1 can be found as follows:

$$\begin{aligned} B\mathbf{e} =& \frac{1}{2}C(I + C^{N-2})(e) = \frac{1}{2}C(\mathbf{e} + \lambda^{N-2}\mathbf{e}) = \\ =& \frac{1}{2}\lambda(1 + \lambda^{N-2})\mathbf{e}. \end{aligned}$$

Therefore the eigenvalues for B_1 are clearly

$$\{\frac{1}{2}\lambda(1 + \lambda^{N-2}) \quad | \quad \lambda = e^{\frac{2\pi ik}{N}} \ \ , \ \ 1 \leq k \leq N\}. \tag{3.12}$$

It is desirable to display the eigenvalues of B_1 in a different form, in order to facilitate the analysis. It can be seen that the eigenvalues of B_1 are also

$$\begin{aligned}\frac{1}{2}\lambda(1+\lambda^{N-2}) &= \frac{1}{2}(e^{\frac{2\pi ik}{N}} + e^{\frac{2\pi ik}{N}} * e^{\frac{-4\pi ik}{N}}) \\ &= \frac{1}{2}(e^{\frac{2\pi ik}{N}} + e^{\frac{-2\pi ik}{N}}) \\ &= \cos\frac{2\pi k}{N}. \end{aligned} \tag{3.13}$$

Notice that, when N is odd, only $\cos\frac{2\pi k}{N} = 1$ and others $|\mu| \lneq 1$. Therefore, no oscillation occurs when N is odd. On the other hand, when N is even we have

$$\begin{cases}\cos\frac{2\pi N}{2N} = -1 \\ \cos\frac{2\pi N}{N} = 1\end{cases}$$

and the others $|\mu| < 1$. Therefore, it is interesting to analyze in detail when N is even, which implies $N = 2k \ \forall k$.

3.3.1.2 Convergence when $N = 2k$

Let $\{\mathbf{1}, \mathbf{v}, \mathbf{e_3}, ..., \mathbf{e_n}\}$ be the basis of eigenvectors of B. Since all the eigenvalues are distinct, there exists a basis of orthogonal eigenvectors. Therefore, we can choose

$$\mathbf{1}^T = (1, 1, \cdots, 1) \quad \text{such that} \quad B_1\mathbf{1} = \mathbf{1}$$

$$\mathbf{w}^T = (-1, 1, -1, \cdots, -1, 1) \quad \text{such that} \quad B_1\mathbf{v} = -\mathbf{v}$$

and

$$\mathbf{1}^T\mathbf{v} = 0 \qquad \mathbf{1}^T\mathbf{e_i} = 0 = \mathbf{v}^T\mathbf{e_i} = \mathbf{e_i}^T\mathbf{e_j} \qquad \text{such that} \qquad i \neq j.$$

Then it can be stated that

$$\Theta(0) = \alpha\mathbf{1} + \beta\mathbf{v} + \sum_{i=3}^{N}\gamma_i\mathbf{e_i}. \tag{3.14}$$

Therefore,

$$\mathbf{1}^T\Theta(0) = \begin{cases} \sum_i^N \Theta(0) \\ \alpha \mathbf{1}^T\mathbf{1} = \alpha N \end{cases} \implies \alpha = \frac{1}{N}\sum_i^N \Theta(0), \tag{3.15}$$

and also

$$\mathbf{v}^T\Theta(0) = \begin{cases} \sum_{i=1}^{\frac{N}{2}} \Theta_{2i}(0) - \sum_{i=1}^{\frac{N}{2}} \Theta_{2i-1}(0) \\ \beta \mathbf{v}^T\mathbf{v} = \beta N \end{cases} \implies \beta = \frac{1}{N}\sum_{i=1}^{\frac{N}{2}} (\Theta_{2i}(0) - \Theta_{2i-1}(0)). \tag{3.16}$$

From the definition of $\Theta(k)$ in Equation (3.10), and by subsituting $\Theta(0)$ from Equation (3.14), we get

$$\begin{aligned} \Theta(1) = B_1\Theta(0) = B_1[\alpha\mathbf{1} + \beta\mathbf{v} + \sum_{i=3}^N \gamma_i \mathbf{e_i}] \\ = \alpha\mathbf{1} + \beta\mathbf{v} + \sum_{i=3}^N \gamma_i \alpha_i \mathbf{e_i} \\ \Theta(2) = B_1\Theta(1) = B_1[\alpha\mathbf{1} + \beta\mathbf{v} + \sum_{i=3}^N \gamma_i \mathbf{e_i}] \\ = \alpha\mathbf{1} + \beta\mathbf{v} + \sum_{i=3}^N \gamma_i \alpha_i^2 \mathbf{e_i} \\ \dots \end{aligned}$$

In general,

$$\begin{cases} \Theta(2k) = \alpha\mathbf{1} + \beta\mathbf{v} + \sum_{i=3}^N \gamma_i \alpha_i^{2k} \mathbf{e_i}, \\ \Theta(2k) = \alpha\mathbf{1} - \beta\mathbf{v} + \sum_{i=3}^N \gamma_i \alpha_i^{2k+1} \mathbf{e_i}. \end{cases} \tag{3.17}$$

Observe from Equation (3.17) that there exists oscillation about

$$\begin{cases} \alpha\mathbf{1} + \beta\mathbf{v} = \mathbf{w}_1 \\ \alpha\mathbf{1} - \beta\mathbf{v} = \mathbf{w}_2, \end{cases} \tag{3.18}$$

where

$$\mathbf{w}_1(2k) = \mathbf{w}_{2k-1} = \frac{2}{N}\sum_{i=1}^{\frac{N}{2}} \Theta_{2i}(0) = \mathbf{w}_e$$
$$\mathbf{w}_{2k-1} = \mathbf{w}_{2k} = \frac{2}{N}\sum_{i=1}^{\frac{N}{2}} \Theta_{2i-1}(0) = \mathbf{w}_o.$$

Therefore, there exists oscillation about these two configurations:

$$\mathbf{w}_1 = \begin{bmatrix} w_o \\ w_e \\ w_o \\ \vdots \\ w_e \end{bmatrix} \qquad \mathbf{w}_2 = \begin{bmatrix} w_e \\ w_o \\ w_e \\ \vdots \\ w_o \end{bmatrix}.$$

3.3.2 Convergence of Control Law 2

Since it is undesirable to have an unstable system, or in other words a system that oscillates between two points, B_2 is defined as a modification of B_1, which brings us to our second control law. Observe that

$$B_2 = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & 0 & \cdots & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \frac{1}{4} \\ \frac{1}{4} & \cdots & \cdots & \cdots & \frac{1}{4} & \frac{1}{2} \end{bmatrix}. \tag{3.19}$$

It is interesting to notice that B_2 is not much different from B_1. Actually, as it was stated before, B_2 and is defined by using B_1. It is also beneficial to define B_2 by B_1 because the eigenvalues of B_1 have already been defined by Equation (3.12). Where B_2 is defined as

$$B_2 = \frac{1}{2}I + \frac{1}{2}B_1 = \frac{1}{2}(I + B_1).$$

3.3.2.1 Eigenvalues of Control Law 2

The eigenvalues of B_2 can be found by observing that

$$\begin{aligned} B_2 =& \{\frac{1}{2}(1+\mu) \mid \mu \text{ eigenvalues of } B_1\} \\ =& \{\frac{1}{2}(1+\cos\frac{2\pi k}{N}) \mid 1 \leq k \leq N\}. \end{aligned}$$

Therefore,

$$0 \leq \frac{1}{2}\left[1+\cos\frac{2\pi k}{N}\right] \leq 1.$$

Notice that there is only one eigenvalue at 1.

3.3.2.2 Convergence

Let $\{\mathbf{1}, \mathbf{e}_2, \cdots, \mathbf{e}_N\}$ be the basis of eigenvectors:

$$\Theta(0) = \alpha\mathbf{1} + \sum_{i=2}^{N}. \tag{3.20}$$

Again,

$$\alpha = \frac{\sum_i \Theta(0)}{N} = \frac{2\pi}{N}$$

because B_2 is symmetric and $B_2 \cdot \mathbf{1} = \mathbf{1}$.

Once again, from the definition of $\Theta(k)$ in Equation (3.10), and by substituting $\Theta(0)$, from Equation (3.20), we get

$$\Theta(k) = B_2^{k-1}\Theta(0) = \alpha\mathbf{1} + \sum_i \gamma_i \alpha_i^{k-1} e_i.$$

This yields

$$\Theta(k) \Rightarrow \alpha\mathbf{1} = \frac{2\pi}{N}\begin{bmatrix} 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix},$$

which is convergence of the exponential type.

CHAPTER 4

ESTIMATION FILTER

4.1 Motivation

In summary, up to this point, the theory behind optimal sensor placement for a single target location estimation has been developed. This was done by first deriving J, the Fisher informtion matrix (FIM) for nonrandom parameters, where the FIM defines the Cramer-Rao lower bound (CRLB), $J^{-1} = CRLB$. It is interesting to analyze the CRLB since it is known to bound the covariance of the error, defined by (2.3). Therefore, the aproach chosen to minimize the CRLB was understood as the maximization of the FIM. From this, the optimal angle was determined, with the horizontal, for N number of sensors, defined as $\theta_N = \frac{2\pi}{N}$. Therefore, we have the ability to obtain the best information possible by placing the sensors in the optimal position. It is now necessary to find a filter that will process this information. In the following sections, the Kalman filter and informtion filter will be derived in detail. Then the extended information filter (EIF), extended Kalman filter (EKF), and decentralized extended information filter algorithms will be presented. From these choices of filters, the decentralized extended information filter was chosen since it processes information from local observation and neighboring nodes as opposed to having a centralized location to process all the data.

4.2 Kalman filter

4.2.1 State-space model

In 1960 R. Kalman published his famous paper on a recursive solution to the discrete-data linear filtering problem. This filter is known as the Kalman filter. Since then, the Kalman filter has been part of a great deal of research in the area of autonomous navigation [5].

In essence , the Kalman filter address the problem of estimating the state $x \in \mathbb{R}^n$ of a discrete-time controlled process. The process is governed by a linear stochastic difference equation

$$x(k) = F(k)x(k-1) + B(k)u(k-1) + w(k-1), \tag{4.1}$$

where $x(k)$ is the state of interest at time k, $F(k)$ is the state transition matrix, B is the control input matrix, $u(k)$ is the control input vector, and $w \sim N(0, Q)$ is the introduced process noise. The process noise is modeled as an uncorrelated, zero-mean, white sequence with process noise covariance

$$E[w(i)w^T(j)] = \gamma_{ij}Q(i). \tag{4.2}$$

The system's states are observed, $z \in \mathcal{R}^m$, by

$$z(k) = H(k)x(k) + v(k), \tag{4.3}$$

where $z(k)$ is the observations made at time k, $H(k)$ is the observation matrix, and $v(k) \sim N(0, Q)$ is the introduced measured noise. The observation noise is modeled as an uncorrelated, zero-mean, white sequence with measurement noise covariance

$$E[v(i)v^T(j)] = \gamma_{ij}R(i). \tag{4.4}$$

It is assumed that the process noise and observation noise are uncorrelated:

$$E[w(i)v^T(j)] = 0.$$

4.2.2 Kalman filter algorithm

With the use of the state-space model, the Kalman filter algorithm is defined without the details of the derivation. For addition details, refer to [5]. The Kalman filter is a recursive estimation algorithm that can be summarized in two stages: the *Prediction* stage and the *Estimation* stage. They are governed by the following equations:

4.2.2.1 Kalman filter algorithm

PREDICTION

$$\hat{x}(k|k-1) = F(k)\hat{x}(k-1|k-1) + B(k)u(k), \tag{4.5}$$

$$P(k|k-1) = F(k)P(k-1|k-1)F^T + Q(k). \tag{4.6}$$

ESTIMATION

$$\hat{x}(k|k) = [1 - W(k)H(k)]\hat{x}(k|k-1) + W(k)z(k), \tag{4.7}$$

$$P(k|k) = P(k|k-1) - W(k)S(k)W^T(k), \tag{4.8}$$

where $W(k)$ and $S(k)$ are known as the gain and innovation covariance matrices, respectively, and are given by

$$W(k) = P(k|k-1)H^T(k)S^{-1}(k), \tag{4.9}$$

$$S(k) = H(k)P(k|k-1)H^T(k) + R(k). \tag{4.10}$$

It is useful to point out that the Kalman filter algorithm can be interpreted as a linear weighted sum of state prediction and observation. Notice that in Equation (4.7), the quantity $\{1 - W(k)H(k)\}$ modifies the amount of $x(k|k-1)$, the prediction, and $W(k)$ modifies $z(k)$, the observation at time k. Therefore, it has the built-in ability to have more trust in the state-space model or to have more confidence in the data collected from the measurements. The amount of confidence in the model or in the observation is specified by the process and observation noise covariances.

To obtain a better understanding on how the algorithm for the Kalman filter works, it is useful to refer to Figure 4.1. It demonstrates the flow diagram of the Kalman filter.

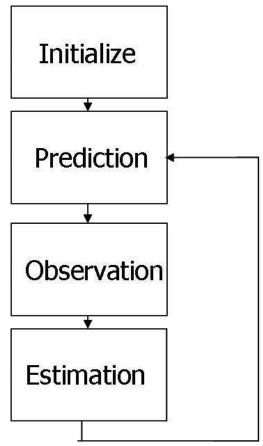


Figure 4.1 Flow Diagram of the Kalman Filter

4.3 Information Filter

The Kalman filter and extended Kalman filter work quite well when estimating, the state x of a single source. When dealing with multiple sources, then the update equations become algebraically quite complicated. Therefore, since the problem we are trying to solve deals with measurements from multiple sensors, it is desirable to have a simple and equivalent form of the Kalman filter. An algebraically equivalent form of the Kalman filter was derived by Arthur G. O. Mutambara called the *information filter*.

The *information filter* is essentially a Kalman filter expressed in terms of measures of *information* about desired states, rather than direct state estimates and their associated covariances [6]. The information filter employs the notion of Fisher information J and the Cramer-Rao lower bound (CRLB), where the Fisher information matrix $J(K)$ is equal to the inverse of the covariance matrix $P(k|k)$ and this is equal to CRLB, where $J(k) = (CRLB)^{-1} = P^{-1}(k|k)$ [6].

4.3.1 Information filter derivation

The following derivation of the information filter is presented here to present a complete picture of how the information filter and the Kalman filter are algebraically similar. This derivation can be found in Mutambara's book [6]. In the information filter there are two key variables, *information matrix* and *information state vector*. The information matrix is defined as the inverse of the covariance matrix:

$$Y(i|j) \triangleq P^{-1}(i|j). \tag{4.11}$$

The information state vector is a product of the inverse of the covariance matrix and the state estimate:

$$\hat{y}(i|j) \triangleq P^{-1}(i|j)\hat{x}(i|j) \tag{4.12}$$

$$= Y(i|j)\hat{x}(i|j). \tag{4.13}$$

The following derivation shows how the information filter is derived from the Kalman filter algorithm by postmultiplying the term $[1 - W(k)H(k)]$ from Equation (4.7) by the term $[P(k|k-1)P^{-1}(k|k-1)]$:

$$\begin{aligned}
[1 - W(k)H(k)][P(k|k-1)P^{-1}(k|k-1)] =& \\
&= [P(k|k-1) - W(k)H(k)P(k|k-1)]P^{-1}(k|k-1) \\
&= [P(k|k-1) - W(k)S(k)S^{-1}(k)H(k)P(k|k-1)]P^{-1}(k|k-1) \\
&= [P(k|k-1) - W(k)S(k)W^T(k)]P^{-1}(k|k-1) \\
&= P(k|k)P^{-1}(k|k-1).
\end{aligned} \tag{4.14}$$

Substituting the expression of the innovation covariance $S(k)$, given in Equation (4.10), into the expression of the filter gain matrix $W(k)$ from Equation (4.9) gives

$$\begin{aligned}
W(k) &= P(k|k-1)H^T(k)[H(k)P(k|k-1)H^T(k) + R(k)]^{-1} \\
\Leftrightarrow W(k)[H(k)P(k|k-1)H^T(k) + R(k)] &= P(k|k-1)H^T(k) \\
\Leftrightarrow W(k)R(k) &= P(k|k-1)H^T(k) - W(k)H(k)P(k|k-1)H^T(k) \\
W(k)R(k) &= [I - W(k)H(k)]P(k|k-1)H^T(k) \\
\Leftrightarrow W(k) &= [I - W(k)H(k)]P(k|k-1)H^T(k)R^{-1}(k).
\end{aligned} \tag{4.15}$$

Substituting Equation (4.14) into Equation (4.15) gives

$$\begin{aligned}
W(k) =& P(k|k)P^{-1}(k|k-1)P(k|k-1)H^T(k)R^{-1}(k) \\
W(k) =& P(k|k)H^T(k)R^{-1}(k).
\end{aligned} \tag{4.16}$$

To get the update equation for the information state vector, substitute Equations (4.14) and (4.16) into Equation (4.7) and premultiply through by $P^{-1}(k|k)$:

$$P^{-1}(k|k)\hat{x}(k|k) = P^{-1}(k|k-1)\hat{x}(k-1|k-1) + H^T(k)R^{-1}(k)z(k),$$

or

$$\hat{y}(k|k) = \hat{y}(k|k-1) + H^T(k)R^{-1}(k)z(k). \tag{4.17}$$

Using the same train of thought, a similar expression can be found for the information matrix associated with this estimate. Using Equations (4.8), (4.9), and (4.14), it follows that

$$\begin{aligned} P(k|k) = [1 - W(k)H(k)]P(k|k-1)[1 - W(k)H(k)]^T \\ + W(k)R(k)W^T(k). \end{aligned} \tag{4.18}$$

Substituting Equations (4.14) and (4.16) gives

$$\begin{aligned} P(k|k) = {} & [P(k|k)P^{-1}(k|k-1)]P(k|k-1)[P(k|k)P^{-1}(k|k-1)]^T \\ & + [P(k|k)H^T(k)R^{-1}(k)]R(k)[P(k|k)H^T(k)R^{-1}(k)]^T. \end{aligned} \tag{4.19}$$

In order to obtain the desired form, pre- and postmultiply by $P^{-1}(k|k)$, giving the information matrix update equation as

$$P^{-1}(k|k) = P^{-1}(k|k-1) + H^T(k)R^{-1}(k)H(k) \tag{4.20}$$

or

$$Y(k|k) = Y(k|k-1) + H^T(k)R^{-1}(k)H(k). \tag{4.21}$$

In order to have the complete algorithm for the information filter, three pieces of information are missing: *information state contribution* $i(k)$, *associated information matrix* $I(k)$, and the *information propagation* *coefficient* $L(k|k-1)$. They are defined respectively as follows:

$$i(k) \triangleq H^T(k)R^{-1}(k)z(k) \tag{4.22}$$

$$I(k) \triangleq H^T(k)R^{-1}(k)H(k). \tag{4.23}$$

The information propagation coefficient, which is independent of the observations made, is given by the expression

$$L(k|k-1) = Y(k|k-1)F(k)Y^{-1}(k-1|k-1). \tag{4.24}$$

All the information needed in the information filter has been well defined. Now the linear Kalman filter can be represented in terms of the *information state vector* and the *information matrix*.

4.3.1.1 Information filter algorithm

PREDICTION

$$\hat{y}(k|k-1) = L(k|k-1)\hat{y}(k-1|k-1) \tag{4.25}$$

$$Y(k|k-1) = [F(k)Y^{-1}(k-1|k-1)F^T(k) + Q(k)]^{-1}. \tag{4.26}$$

ESTIMATION

$$\hat{y}(k|k) = \hat{y}(k|k-1) + i(k) \tag{4.27}$$

$$Y(k|k) = Y(k|k-1) + I(k). \tag{4.28}$$

4.4 Extended Kalman Filter and Extended Information Filter

Note that the algorithms defined above are used to estimate the states $x \in \mathbb{R}^n$ or information state-vectors $y \in \mathbb{R}^n$ of a discrete-time controlled process governed by a linear stochastic difference equation. If one would like to estimate states governed or measured by a nonlinear stochastic difference equation, then an extended Kalman filter (EKF) or extended information filter (EIF) is used. The EKF can be thought of as a Kalman filter that linearizes about the current mean and covariance [5]. The derivation of Mutambara's extended information filter follows from that of the linear Kalman filter, linearizing state and observation models using Taylor's series expansion [6]. Since the problem at hand deals with a nonlinear observation model $z(k)$ defined by Equation (2.1). The interest lies in a filter that has the ability to process nonlinear stochastic difference equations, and that in essence is the extended Kalman filter or the extended information filter.

4.4.1 Nonlinear state space

The the model of interest is described by a nonlinear stochastic difference equation in the form

$$x(k) = f(x(k-1), u(k-1), (k-1)) + w(k), \tag{4.29}$$

where $x(k-1)$ is the state vector and $u(k-1)$ is the known control vector input, both at time $(k-1)$. The process noise introduced at time k is defined as $w(k)$. The nonlinear state transition function is $f(.,.,k-1)$. The observations made by the system are modeled by a nonlinear equation defined as

$$z(k) = h(x(k), k) + v(k), \tag{4.30}$$

where $h(.,k)$ is the nonlinear observation transition function and $v(k)$ is the observation noise. Both $w(k)$ and $v(k)$ are modeled as linearly additive Gaussian, temporally uncorrelated with zero mean, which means

$$E[w(k)] = E[v(k)] = 0 \quad \forall k, \tag{4.31}$$

with the corresponding covariance given by

$$E[w(i)^T w(j)] = \delta_{ij} Q(i), \qquad E[v(i)^T v(j)] = \delta_{ij} R(i).$$

It is assumed that the process noise and observation noise are uncorrelated:

$$E[w(i)^T v(j)] = 0, \quad \forall i, j.$$

4.4.2 EKF and EIF algorithm

Now that the nonlinear state space has been defined, it is possible to present the EKF and EIF algorithms. Both Algorithms the EKF and EIF, are presented here without a derivation. Much has been written on the EKF [4, 5] and the derivation of EIF can be found in Mutambara's book [6].

4.4.2.1 Extended Kalman filter algorithm

PREDICTION

$$\hat{x}(k|k-1) = f(\hat{x}(k-1|k-1), u(k-1), (k-1)) \tag{4.32}$$

$$P(k|k-1) = \nabla f_x(k) P(k-1|k-1) \nabla f_x^T(k) + Q(k-1). \tag{4.33}$$

ESTIMATION

$$\hat{x}(k|k) = \hat{x}(k|k-1) + W(k)[z(k) - h(\hat{x}(k|k-1))] \tag{4.34}$$

$$P(k|k) = P(k|k-1) - W(k)S(k)W^T(k). \tag{4.35}$$

The gain and innovation covariance matrices are given respectively by

$$W(k) = P(k|k-1)\nabla h_x^T(k) S^{-1}(k) \tag{4.36}$$

$$S(k) = \nabla h_x(k) P(k|k-1) \nabla h_x^T(k) + R(k). \tag{4.37}$$

4.4.2.2 Information Kalman filter algorithm

PREDICTION

$$\hat{y}(k|k-1) = Y(k|k-1) f(k, \hat{x}(k-1|k-1), u(k-1), (k-1)) \tag{4.38}$$

$$Y(k|k-1) = [\nabla f_x(k) Y^{-1}(k-1|k-1) \nabla f_x^T(k) + Q(k)]^{-1}. \tag{4.39}$$

$$\hat{y}(k|k) = \hat{y}(k|k-1) + i(k) \tag{4.40}$$

$$Y(k|k) = Y(k|k-1) + I(k). \tag{4.41}$$

The information state contribution and its associated information matrix are given respectively by

$$I(k) = \nabla h_x^T(k) R^{-1}(k) \nabla h_x(k) \tag{4.42}$$

$$i(k) = \nabla h_x^T(k) R^{-1}(k)[\upsilon(k) + \nabla h_x(k)\hat{x}(k|k-1)], \tag{4.43}$$

where $\upsilon(k)$ is the innovation given as

$$\upsilon(k) = z(k) - h(\hat{x}(k|k-1)). \tag{4.44}$$

4.5 Decentralized Extended Information Filter

When working with measurements from different sources, it is desirable to decentralize the system. In a data processing decentralized system, all information is processed locally, where no central processing site exists. In a system like this, all the information is processed at each node locally, based on local observations and information communicated by its neighbors. Therefore, there is no central process, where a global decision is made, each decision is made locally by each node, using the information collected by it and other nodes. It is important to notice the advantage of permitting only node-node communication.

To give a practical application of a decentralize system, let there be N number of vehicles, estimating the location of a target. If one of these vehicles fails, then there are $N-1$ vehicles left to estimate the location of a target. In a centralized approach, the system would not adapt to $N-1$ vehicles, since it was built for N vehicles. In retrospect, for a decentralized system to have $N-1$ vehicles would not matter because it is based on obtaining only local information, node-node communication. Therefore, it would adapt to its new environment and continue to estimate the location of a target without interruption. In essence, the decentralized algorithm has the ability to dynamically adapt to the new number of vehicles in order to continue to estimate the location of a given target. Such an algorithm that has this ability has been derived by Mutambara, and it is called the decentralize extended information filter. Since the derivation of this algorithm is detailed, the algorithm is presented here without a derivation.

4.5.1 DEIF algorithm

PREDICTION

$$\hat{y}_i(k|k-1) = Y_i(k|k-1)f(k, \hat{x}_i(k-1|k-1), u_i(k-1), (k-1)) \tag{4.45}$$

$$Y_i(k|k-1) = [\nabla f_{x_i}(k)Y_i^{-1}(k-1|k-1)\nabla f_{x_i}^T(k) + Q(k)]^{-1}. \tag{4.46}$$

ESTIMATION

$$\hat{y}_i(k|k) = \hat{y}_i(k|k-1) + \sum_{j=1}^{N} i_j(k) \tag{4.47}$$

$$Y_i(k|k) = Y_i(k|k-1) + \sum_{j=1}^{N} I_j(k). \tag{4.48}$$

The local information state contribution and its local associated information matrix are given respectively by

$$I_j(k) = \nabla h_{x_j}^T(k)R_j^{-1}(k)\nabla h_{x_j}(k) \tag{4.49}$$

$$i_j(k) = \nabla h_{x_j}^T(k)R_j^{-1}(k)[\upsilon_j(k) + \nabla h_{x_j}(k)\hat{x}_j(k|k-1)], \tag{4.50}$$

where $\upsilon(k)$ is the innovation given as

$$\upsilon_j(k) = z_j(k) - h_j(\hat{x}_j(k|k-1)). \tag{4.51}$$

With the equation of the DEIF algorithm well defined, Figure 4.2, shows how the algorithm works. Notice that each node has a DEIF built in, giving it the ability to provide an estimation from its local observations $z_i(k)$ and from the information communicated to it by other nodes. It is also important to note that each box labeled “info filter” in Figure 4.2, would actually be the sensor collecting the data locally and communicating it to other sensors. This filter actually has full communication with all the other nodes but it does not have a central process, therefore making a decentralized algorithm.

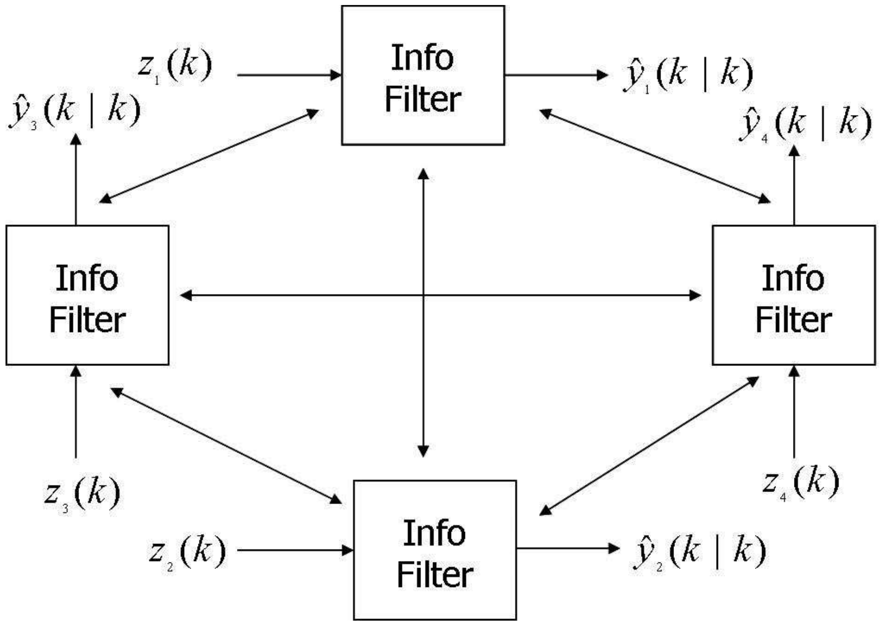


Figure 4.2 Decentralized Extended Information Filter

Part of our research objective is to have the ability to adapt to an environment dynamically, using decentralized methods. Since the decentralized extended information filter, out of the filters presented here, possesses this property, it was chosen as the filter to be used in the simulation.

CHAPTER 5

NUMERICAL SIMULATIONS

5.1 Introduction

In finding a solution to the problem of target tracking from a multisensor network, it has been proven in Chapter 2 that the *deployment of the agents* should maximize the probability of detection of the target to be tracked or provide more accurate estimations of the point source to be localized. Chapter 2 develops a method of obtaining the best possible estimation of nonrandom parameters. Because it is desireable to track a moving target with N number of sensors, and because a method of obtaining the best estimation of nonrandom parameters has been developed and fully understood, it follows from these results that

In order to obtain the best estimate of a moving target, it is desirable to have the sensors move to an optimal position described by Proposition 1(p. 17).

A solution to these problems should be built on motion control algorithms for the network and data fusion techniques which allow decentralized implementations. Such a decentralized motion planning control algorithm has been described and proved to converge in Chapter 3. In Chapter 4, the possible estimation algorithms have been described, and the decentralized extended information filter was chosen to be the estimator used in the simulations. Therefore, this chapter aims at supporting the statement made of estimating the location of a moving target via N number of sensors, using decentralized motion planning and decentralized estimation algorithms.

5.2 Simulation Model

5.2.1 State-space model

The state-space model will consist of modeling the trajectory of a moving target in $\mathbb{R}^2$ and the measurement/observation made by each ith sensor, where $1 \leq i \leq N$. In addition, the model contains both the process and observation/measurement noise.

$$x_i(k+1) = x_i(k) + w(k), \tag{5.1}$$

where $x_i(k+1)$ is the state of interest at time $(k+1)$, $x_i(k)$, random walk process describing the trajectory of a moving target, and $w \sim N(0, Q)$ is the introduced process noise. The process noise is modeled as an uncorrelated, zero-mean, white sequence with process noise covariance:

$$E[w(i)w^T(j)] = \gamma_{ij} Q(i). \tag{5.2}$$

Taking into account that the position of the source q is composed of two directions and since we are dealing with the two-dimensional case, then $q = (q^1, q^2)^T$. For simulation purposes only, and without loss of generality, the trajectory of a point was chosen to be a figure eight defined as

$$x_i(k) = \begin{bmatrix} x - coordinate \\ y - coordinate \end{bmatrix} = \begin{bmatrix} q^1 \\ q^2 \end{bmatrix} = \begin{bmatrix} sin(k) \\ sin(k)cos(k) \end{bmatrix}. \tag{5.3}$$

The system's states are observed, $z \in \mathcal{R}^2$, by

$$z_i(k) = h_i(x(k), k) + v(k), \tag{5.4}$$

where $h(., k)$ is the nonlinear observation transition function defined by

$$h_i(k) = |x_i(k) - p_i(k)|, \quad \text{for } 1 \leq i \leq N, \tag{5.5}$$

where $h_i(k)$ describes the distance measured by the ith sensor, from the moving target $x_i(k)$ as seen by the ith sensor to the $p_i(k)$ sensor at time k. The location of the position of the ith sensor, $p_i(k)$, is composed of two directions, $p_i = (p_i^1, p_i^2)$, and $v(k)$ is the observation noise. The observation noise is modeled as an uncorrelated, zero-mean, white sequence with measurement noise covariance

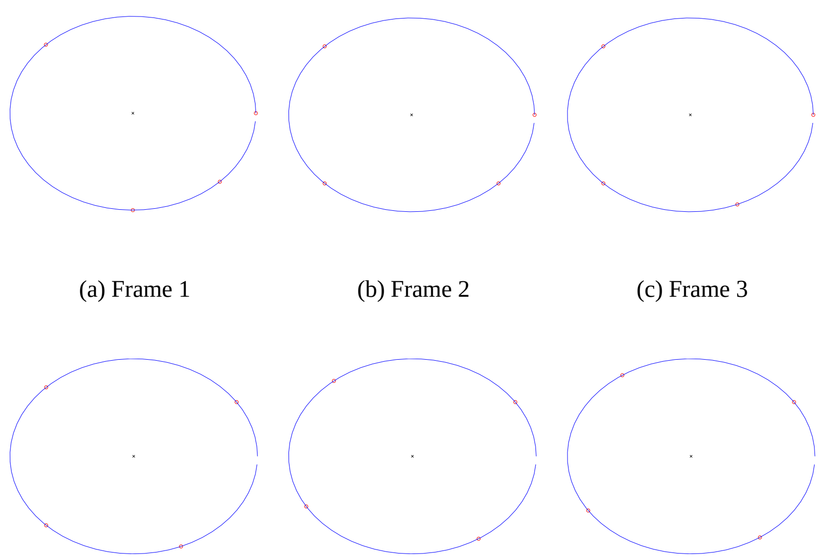
$$E[v(i)v^T(j)] = \gamma_{ij} R(i). \tag{5.6}$$

It is assumed that the process noise and observation noise are uncorrelated:

$$E[w(i)v^T(j)] = 0.$$

5.2.2 Implementation of algorithms

In order to obtain a better estimate of a moving target, the agents/sensors have to move into their optimal positions, described by Proposition 1. To accomplish this goal, the algorithm given in Table 5.1 was implemented in Matlab. Figure 5.1 shows how the algorithm works as time increases. Notice that in Frame 1 the sensors(the circles) are located in a nonoptimal configuration: as time progresses (Frame 6), the sensors are in the optimal configuration, in essence, dividing the circle into $\frac{2\pi}{N}$ equal parts.



(a) Frame 1 (b) Frame 2 (c) Frame 3

(d) Frame 4 (e) Frame 5 (f) Frame 6

Figure 5.1 Control Law as Time Progresses

Table 5.1 Agents Deployment: Decentralize Control Law

Name:	Decentralize Control Law
Goal:	Decentralize Deployment of Agents Control Law
Requires:	(i) Initial locations of sensors $\{p_1, \cdots, p_N\}$
	(ii) Location of target q given by DEIF
	(ii) Counterclockwise where $p_i \neq p_j \;\; \forall i \neq j$
	(iii) Computation of angles $\theta_i = \measuredangle(p_i, p_{i-1})$
	(iv) Positive real θ_i

ALGORITHM

For $i \in \{1, \ldots, N\}$, ith agent, calculate the location of $\{p_1, \cdots, p_n\}$ with respect to target q. While $\theta_i \neq \theta_{i-1} \;\; \forall \;\; 1 \leq i \leq N$

0: set $p_0 = p_n$ and $p_{n+1} = p_1$

0: compute angles θ_i and θ_{i-1}

0: set new $\theta_i' := \frac{\theta_{i+1}+\theta_{i-1}}{2}$

5.3 Matlab Simulation Results

This section describes the results obtained from the implementations of the theory developed in the previous chapters. The state space model and control algorithm used in the simulations are discussed in Sections 5.2 and 5.2.2, respectively. In the simulations, the environment is composed of N number of sensors and one moving target. The objective of the N number of sensors is to estimate the location of the moving target.

The simulations are done with stationary sensors and moving sensors. The moving sensors follow the control algorithm described in Section 5.2.2. This control algorithm allows each sensor to be in its optimal position with respect to the moving point. Hence, being in the optimal configuration ensures that each sensor collects the best information to estimate the location of the point. The sensors, which will also be referred to as vehicles, are restricted to move on a circle. The estimate $\hat{x}$ is obtained by the use of the DEIF algorithm. The user provides the initial guess of $\hat{x}$ to start the DEIF algorithm, which is referred to as *initial guess for*

$\begin{bmatrix} \hat{x} & \hat{y} \end{bmatrix}^T$. The trajectory of the moving point is determined by Equation (5.3).

5.3.1 Location

This section contains the result obtained from placing four stationary and moving vehicles in nonoptimal and optimal positions. The initial position of the sensors is described in the tables as *initial position of sensors (radians)*. It is important to note that the simulation could have been done with more than four sensors, since both the control law and DEIF filter are decentralized. In other words, the filters do not require a set number of sensors. Also, notice that when the moving vehicles are placed in nonoptimal position, it does not make a significant difference in the estimation of the point. Since each moving vehicle is following the control algorithm, it is always in the optimal position for obtaining the best estimate of the moving target.

The parameters of each simulation are described by a table, and the results are displayed as graphs. The graph on the left describes $||x - \hat{x}||$, which provides a measure of the error. The graph on the right gives the final location of the vehicles, the actual position x, and the estimate $\hat{x}$. Only one parameter is varied; in these simulations, the varied parameter is the variance of the measured noise. This allows us to prove numerically that the best estimate $\hat{x}$ of the location x of a moving point is estimated best by the vehicles that are implementing the control algorithm.

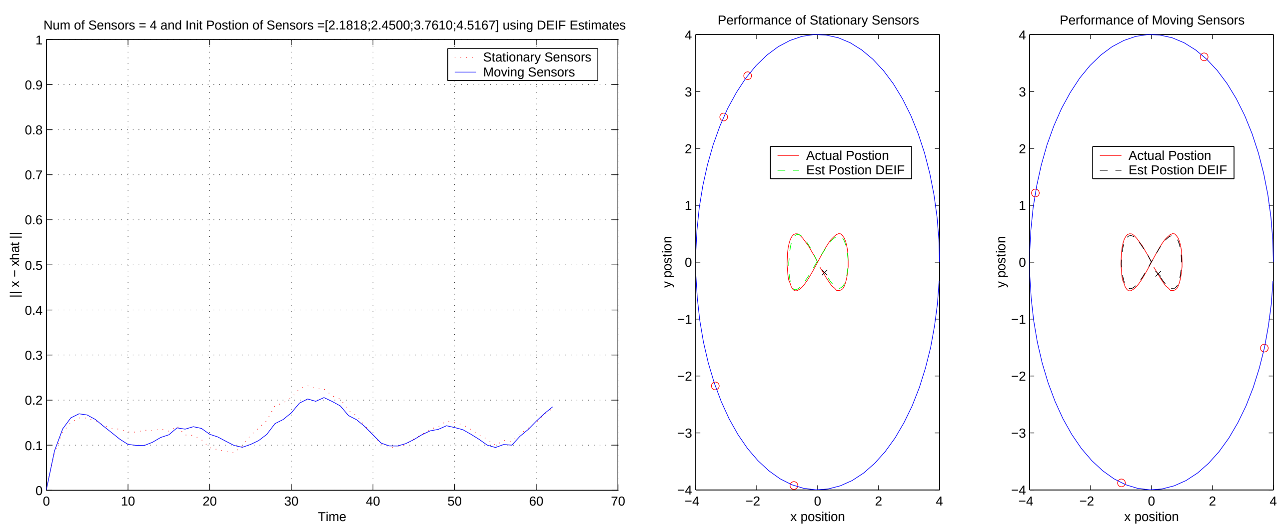
It is important to point out that in simulations 1-4, the initial positions of the vehicles are nonoptimal; in other words, they are placed in random locations. In simulations 5-8, the vehicles are placed in optimal locations. This only matters for the estimates made by the stationary vehicles, since the moving vehicles always position themselves optimally with respect to the moving point. Therefore, the results obtained, will allow us to prove numerically that even in the case of stationary vehicles the best estimate will be obtained by placing the vehicles in an optimal configuration. It will also aid in proving that the vehicles using the control algorithm, designed to implement the results in Proposition 1, ensure a better estimate of the moving target.

5.3.1.1 Nonoptimal position

Analyzing simulation 1 by using Figure 5.2, notice the parameters for this simulation are found in Table 5.2. Since the variance of the measured noise is relatively small, the difference between the estimate of $\hat{x}$ and the actual position x, provided by the moving sensors and the stationary vehicles, is minimal. Still the moving vehicles, using the control algorithm, have an error of less than 0.2, which is better than the stationary vehicles, for which at one time the error $||x - \hat{x}||$ is about 0.25. Also it is interesting to note, that even though the stationary vehicles are placed in nonoptimal positions, the estimate $\hat{x}$ is relatively good; this is due to the fact that the variance of the measured noise is relatively small.

Table 5.2 Parameters for Simulation 1 with Variance of Measured Noise = 0.000 053

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\begin{bmatrix} 2.1818; 2.4500; 3.7160; 4.5167 \end{bmatrix}$	0.000 13	**0.000 053**
Stationary Sensors	4	$\begin{bmatrix} 2.1818; 2.4500; 3.7160; 4.5167 \end{bmatrix}$	0.000 13	**0.000 053**



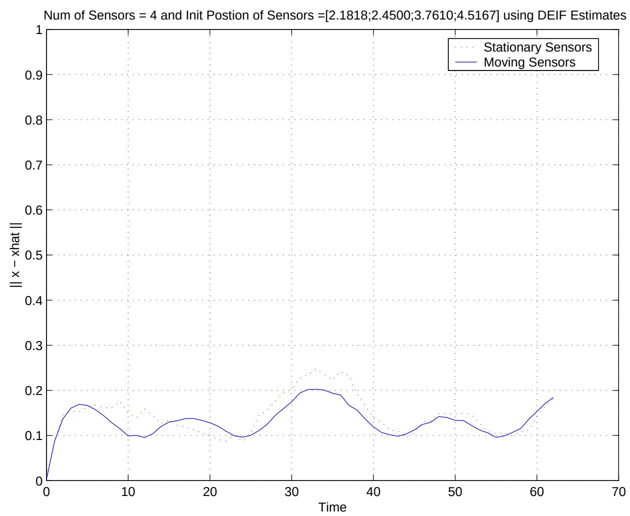
(a) $||x - \hat{x}||$ (b) Final Location of Sensors

Figure 5.2 Simulation 1: Stationary vs. Moving Sensors with Variance of Measured Noise = 0.000 053

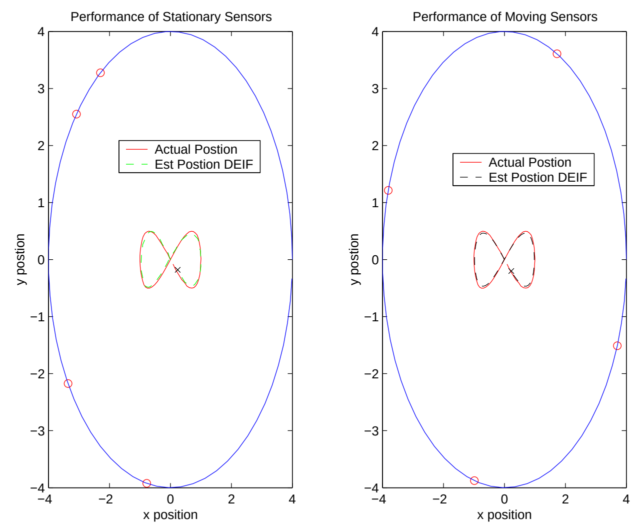
Analyzing simulation 2 by using Figure 5.3, notice the parameters for this simulation are found in Table 5.3. In simulation 2, for the most part, the moving vehicles provide a better estimate $\hat{x}$ of the position of x, since the error of the moving vehicles is for the most part, smaller than that of the stationary vehicles. The results are very similar to simulation 1.

Table 5.3 Parameters for Simulation 2 with Variance of Measured Noise = 0.000 53

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[2.1818; 2.4500; 3.7160; 4.5167\right]$	0.000 13	**0.000 53**
Stationary Sensors	4	$\left[2.1818; 2.4500; 3.7160; 4.5167\right]$	0.000 13	**0.000 53**



(a) $||x - \hat{x}||$



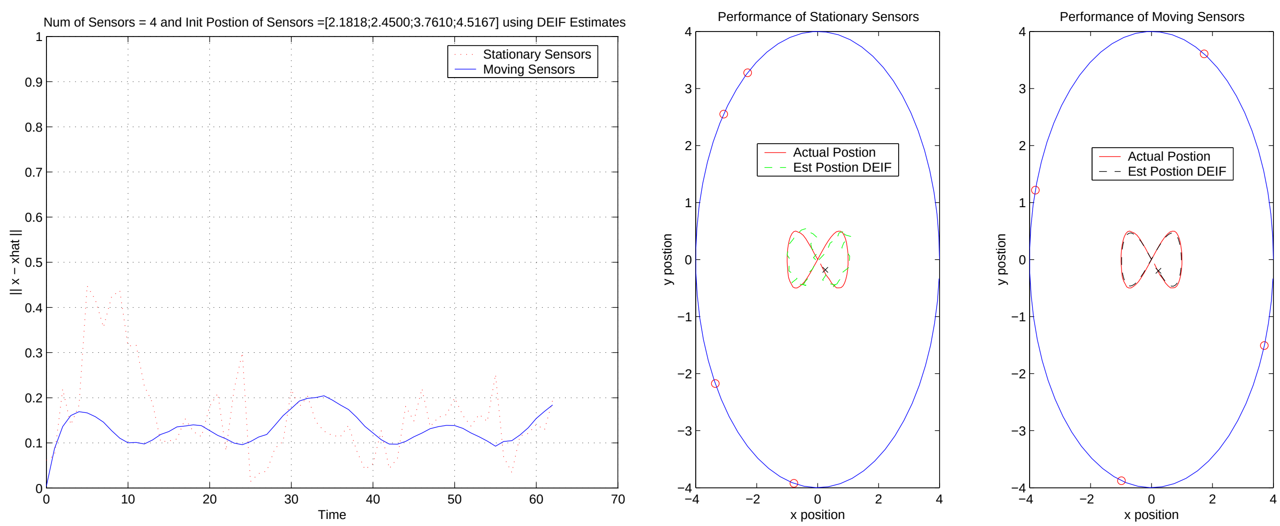
(b) Final Location of Sensors

Figure 5.3 Simulation 2: Stationary vs. Moving Sensors with Variance of Measured Noise = 0.000 53

Analyzing simulation 3 by using Figure 5.4, notice the parameters for this simulation are found in Table 5.4. In simulation 3, the variance of the measured noise is increased by 1000%. Notice that the stationary vehicle no longer provides an error below 0.25, and the error gets as high as 0.45, while the moving vehicles still provide an error below 0.20. Notice, in Figure 5.4, the image on the right gives the results obtained by the stationary vehicles. The estimate $\hat{x}$ fails to follow closely the trajectory of the moving point. On the other hand, it can be seen visually that the moving vehicles still estimate the trajectory of the point relatively well.

Table 5.4 Parameters for Simulation 3 with Variance of Measured Noise = 0.053

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[2.1818; 2.4500; 3.7160; 4.5167\right]$	0.000 13	**0.053**
Stationary Sensors	4	$\left[2.1818; 2.4500; 3.7160; 4.5167\right]$	0.000 13	**0.053**



(a) $||x - \hat{x}||$

(b) Final Location of Sensors

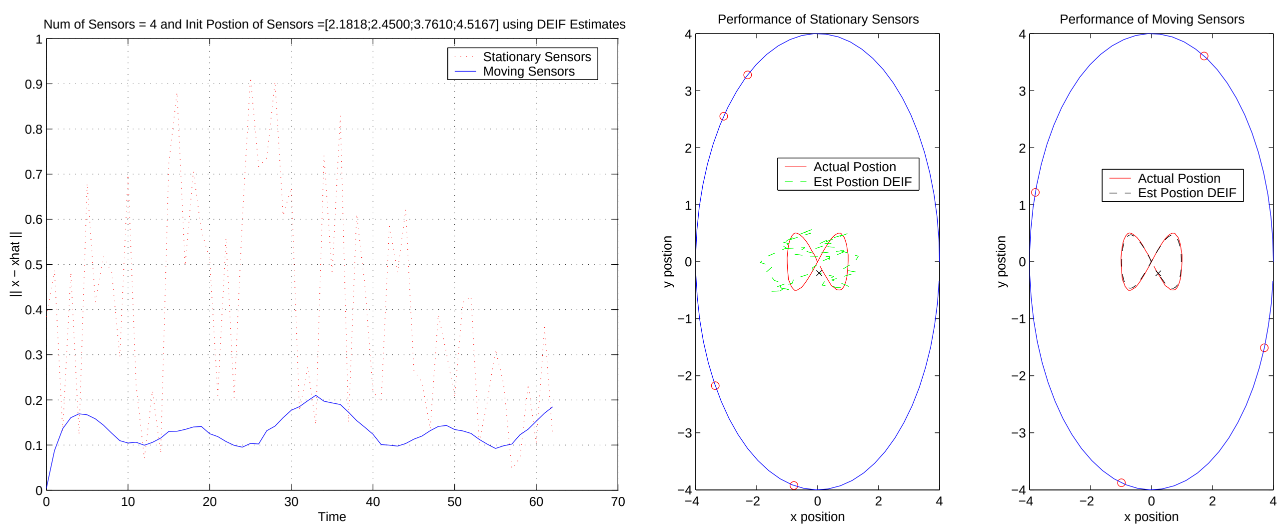
Figure 5.4 Simulation 3: Stationary vs. Moving Sensors with Variance of Measured Noise = 0.053

Analyzing simulation 4 by using Figure 5.5, notice the parameters for this simulation are found in Table 5.5. In simulation 4, the variance of the measured noise is increased by 10 000% of original $0.000\ \ 053$. Notice that the stationary vehicle no longer provides an error below 0.45; the error gets as high as 0.91. Meanwhile, the moving vehicles still provide an error below 0.21. Note, Figure 5.5 on the right, which displays the results of the stationary vehicles. The estimate $\hat{x}$ fails to follow the trajectory of the moving point. On the other hand, it can be seen that the moving vehicles still estimate the trajectory of the point relatively well, even with the variance of the measure noise increased.

Therefore, it can be seen from simulations 1-4 that, as the variance of the measured noise increases, the estimate provided from the stationary vehicles gets worse, while the estimate provided by the moving sensors stays relatively close to the actual position of the moving target. This proves numerically that allowing the vehicles to move will result in a better estimate of a moving target.

Table 5.5 Parameters for Simulation 4 with Variance of Measured Noise = 0.53

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[2.1818; 2.4500; 3.7160; 4.5167\right]$	0.000 13	**0.53**
Stationary Sensors	4	$\left[2.1818; 2.4500; 3.7160; 4.5167\right]$	0.000 13	**0.53**



(a) $||x - \hat{x}||$ (b) Final Location of Sensors

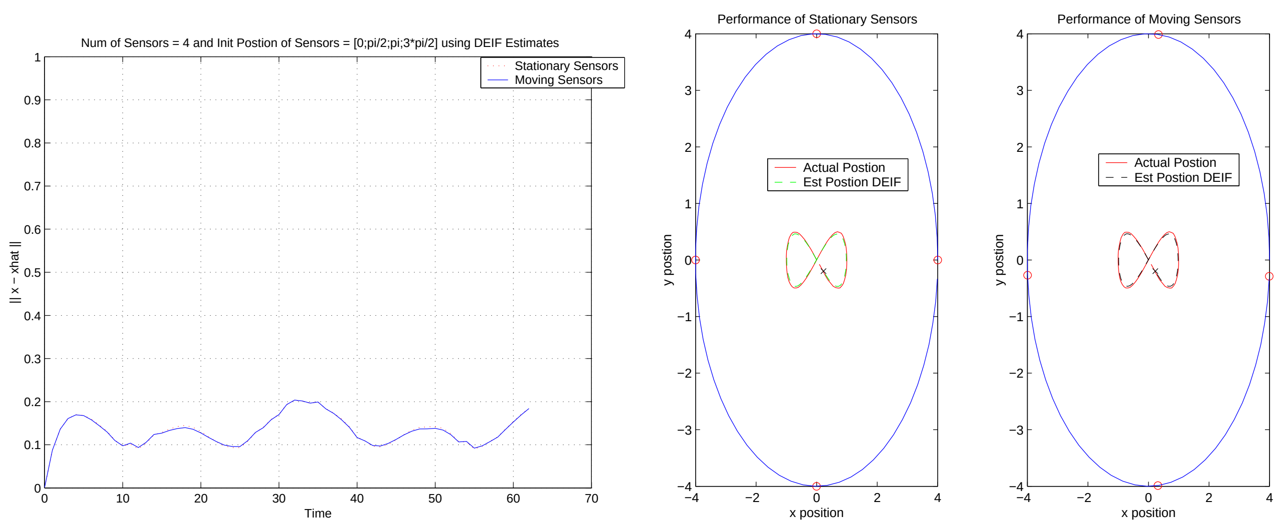
Figure 5.5 Simulation 4: Stationary vs. Moving Sensors with Variance of Measured Noise = 0.53

5.3.1.2 Optimal position

Analyzing simulation 5 by using Figure 5.6, notice the parameters for this simulation are found in Table 5.6. Since the variance of the measured noise is relatively small, the difference between the estimate of $\hat{x}$ and the actual position x, provided by the moving sensors and the stationary vehicles, is almost the same. It is interesting to note that, since the stationary vehicles are placed in optimal positions, the estimate $\hat{x}$ is as good as the moving vehicles.

Table 5.6 Parameters for Simulation 5 with Variance of Measured Noise = 0.000 053

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.000 053**
Stationary Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.000 053**



(a) $||x - \hat{x}||$

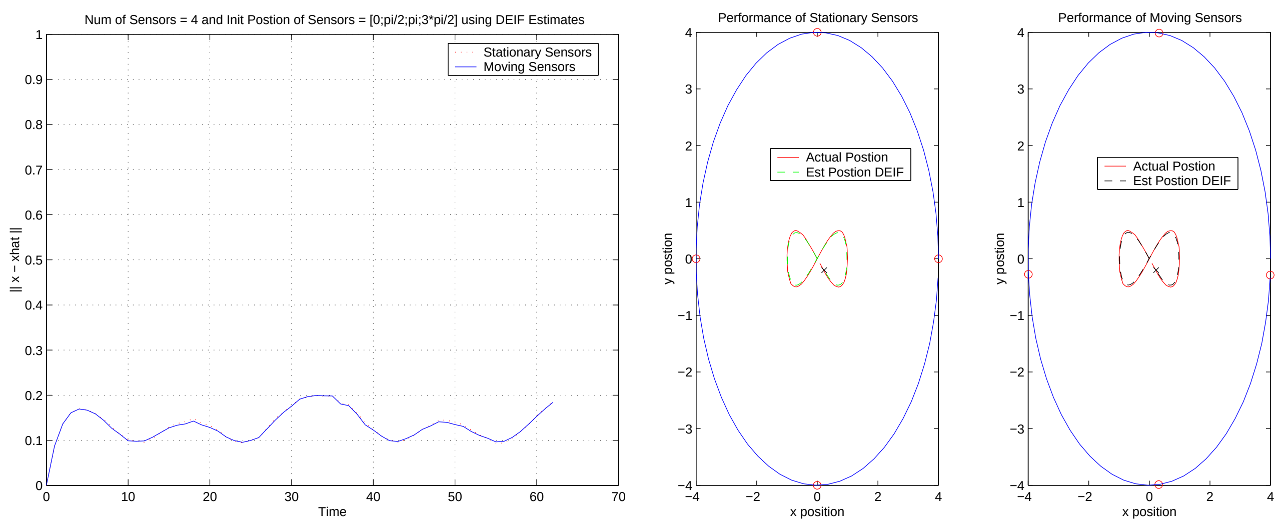
(b) Final Location of Sensors

Figure 5.6 Simulation 5: Stationary vs. Moving Sensors with Variance of Measured Noise = 0.000 053

Analyzing simulation 6 by using Figure 5.7, notice the parameters for this simulation are found in Table 5.7. Visually, the difference between simulations 5 and 6 is minimal, even though the variance of the measured noise has been increased by 10% of the orginal value $0.000\ \ 053$. It is intersting to note that both the stationary vehicles and moving vehicles have an error of less than 0.2. Therefore, up to this point it really does not make a difference if the vehicles follow the control algorithm or not, since the results are almost identical. It can be seen in the simulations that follow, that as the noise increases it becomes more important to use the control algorithm.

Table 5.7 Parameters for Simulation 6 with Variance of Measured Noise = 0.000 53

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.000 53**
Stationary Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.000 53**



(a) $||x - \hat{x}||$

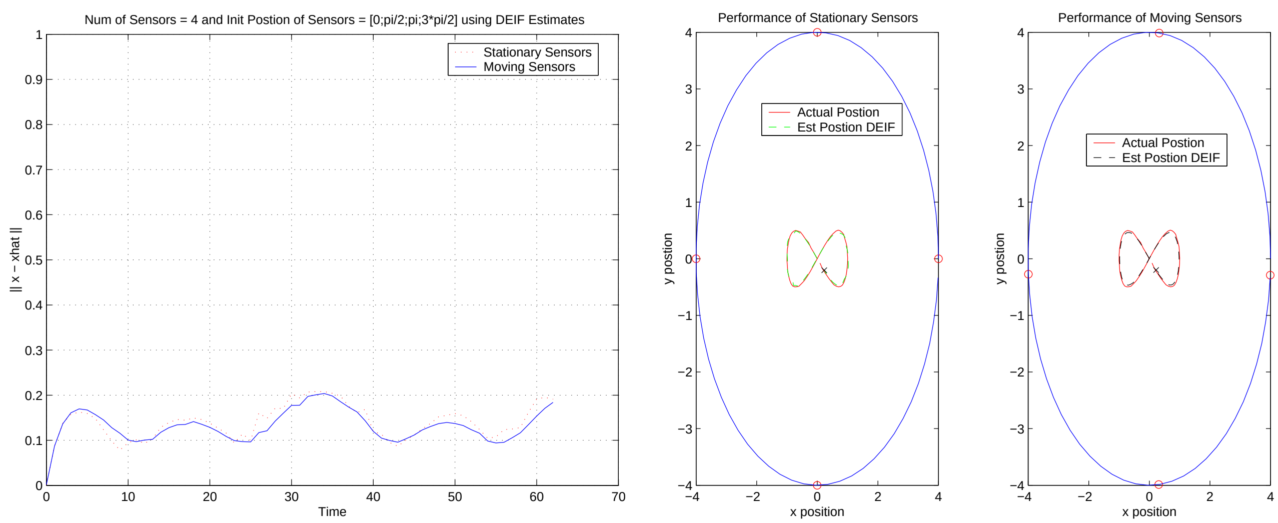
(b) Final Location of Sensors

Figure 5.7 Simulation 6: Stationary vs. Moving Sensors with Variance of Measured Noise = 0.000 53

Analyzing simulation 7 by using Figure 5.8, notice the parameters for this simulation are found in Table 5.8. In simulation 7, the variance of the measured noise is increased by 1000% of $0.000\ \ 053$. Notice that, the stationary vehicles and moving vehicles no longer obtain the same estimate of $\hat{x}$. It is also interesting to note that, even though the variance has been increased, since the stationary vehicles are in the optimal configuration, the error of the stationary vehicles is still under 0.2. This could be expected, from the fact that the stationary vehicles are located in optimal positions.

Table 5.8 Parameters for Simulation 7 with Variance of Measured Noise = 0.053

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.053**
Stationary Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.053**



(a) $||x - \hat{x}||$

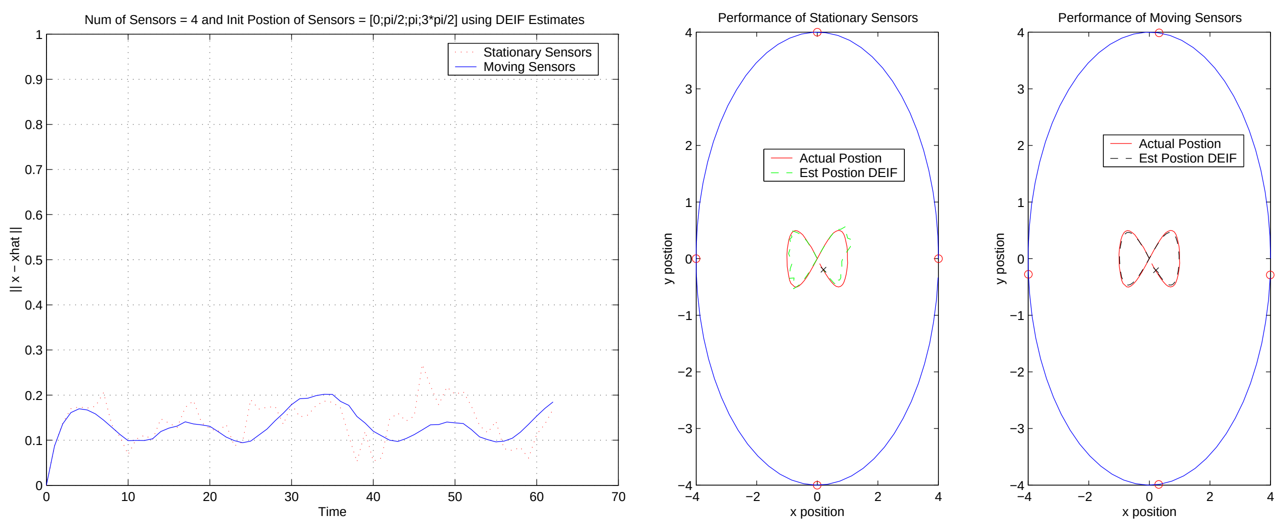
(b) Final Location of Sensors

Figure 5.8 Simulation 7: Stationary vs. Moving Sensors with Variance of Measured Noise = 0.053

Analyzing simulation 8 by using Figure 5.9, notice the parameters for this simulation are found in Table 5.9. Since the variance of the measured noise has been increased by $10\ 000\%$ of the original value in simulation 8, the stationary vechicles no longer obtain the best estimate $\hat{x}$ of the target. It is interesting to note, that even though the error of $\hat{x}$ is below 0.2, the estimate is not the best. This result can be seen visually from the Figure 5.9, by the image on the right providing the performance of the stationary sensor. On the other hand, the moving vehicles still obtain a relatively good estimate of the location of the target, therefore proving numerically that the moving vehicles obtain the best estimate, since they are always at an optimal position, described by Proposition 1.

Table 5.9 Parameters for Simulation 8 with Variance of Measured Noise = 0.53

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\begin{bmatrix}0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\end{bmatrix}$	0.000 13	**0.53**
Stationary Sensors	4	$\begin{bmatrix}0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\end{bmatrix}$	0.000 13	**0.53**



(a) $||x - \hat{x}||$

(b) Final Location of Sensors

Figure 5.9 Simulation 8: Stationary vs. Moving Sensors with Variance of Measured Noise = 0.53

5.3.1.3 Nonoptimal position versus optimal position

In this section, the results obtained from simulations 1-8 will be compiled into graphs. This is done in order to have the ability to analyze the effect on the results due to the stationary vehicles and moving vehicles due to increasing the variance of the measurement noise. In Figure 5.10, the simulations of 1-8 are displayed accordingly, along with the parameters for each, found in Tables 5.10 and 5.11.

In order to have a better understanding of each figure, the following explanation is provided. The top left-hand side of Figure 5.10, displays the error $||x - \hat{x}||$ of the stationary vehicles located in nonoptimal positions. The bottom-left hand side of Figure 5.10 displays the error $||x - \hat{x}||$ of the moving vehicles located in nonoptimal positions. The right-hand top and bottom sides of Figure 5.10 displays the error $||x - \hat{x}||$ of the stationary vehicles and moving vehicles placed in optimal positions, respectively.

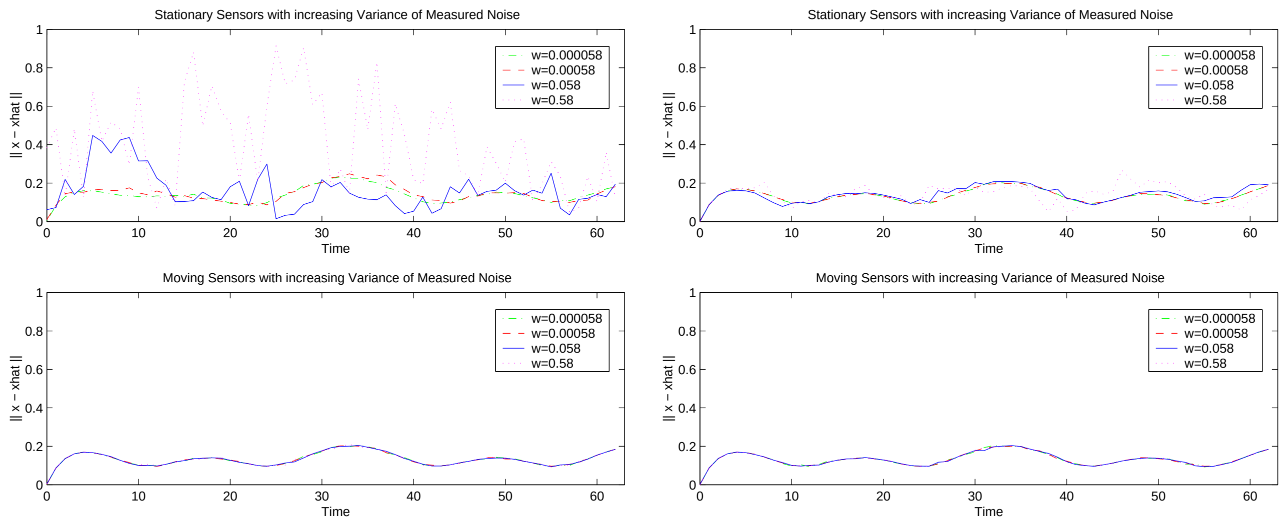
Table 5.10 Parameters for Simulations 1-4

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[2.1818; 2.4500; 3.7160; 4.5167\right]$	0.000 13	**vary**
Stationary Sensors	4	$\left[2.1818; 2.4500; 3.7160; 4.5167\right]$	0.000 13	**vary**

Table 5.11 Parameters for Simulations 5-8

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**vary**
Stationary Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**vary**

With the aid of these results, it is easy to see that the moving vehicles obtain the best estimate, since the error is always less than 0.2. Comparing the results of the stationary vehicles, as the variance of the measured noise increases, the error of $||x - \hat{x}||$ also increases. It is also interesting to note, from Figure 5.10, that for simulations 1-4 and 5-8, it seems like the moving vehicles obtain the same result. This is due to the fact, that the moving vehicles are following the same control alogrithm. The error can be reduced by placing the stationary vehicles in optimal configurations, but a better estimate is obtained with the moving vehicles, therefore, proving that in order to obtain the best estimate of a moving target, it is desirable to have the sensors move to an optimal position. This proves numerically that the moving vehicles obtain the best estimate, regardless of the initial position of the moving vehicles, since the vehicles are always at an optimal configuration, described by Proposition 1.



(a) Simulations: 1-4

(b) Simulations: 5-8

Figure 5.10 Nonoptimal Position vs. Optimal Position

5.3.2 Modifying the trajectory of the moving point

It would be interesting to see if a relatively good estimate is obtained when the trajectory of the moving point is modified. This is accomplished by modifying the trajectory of the moving point described by Equation (5.3) as follows:

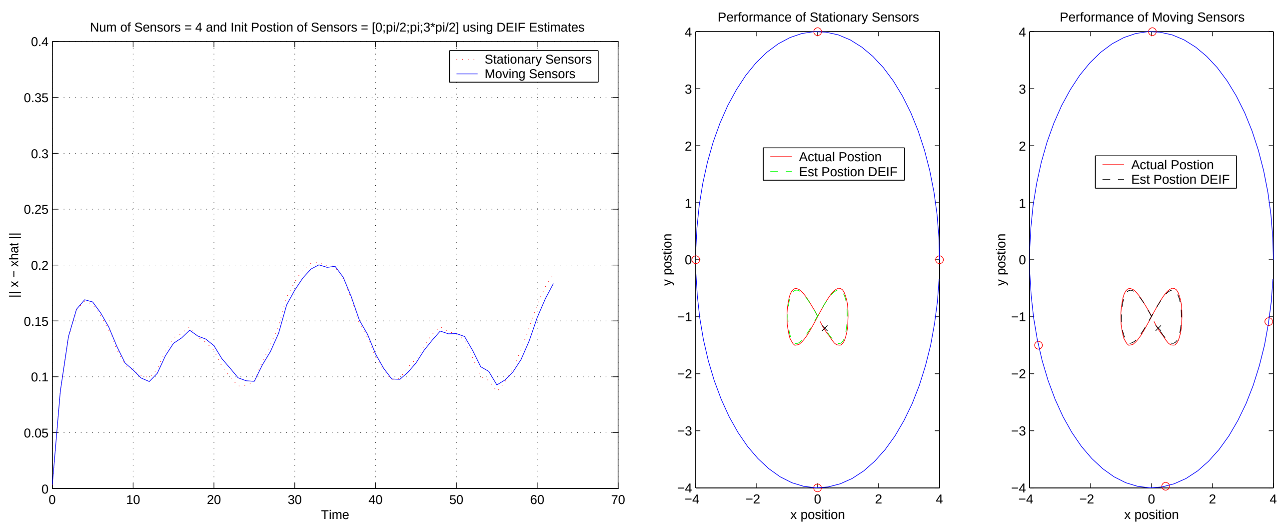
$$x_i(k) = \begin{bmatrix} x - coordinate \\ y - coordinate \end{bmatrix} = \begin{bmatrix} q^1 \\ q^2 \end{bmatrix} = \begin{bmatrix} sin(k) \\ sin(k)cos(k) - 1 \end{bmatrix}. \tag{5.7}$$

In this section, the initial positions of the vehicles are in the optimal configuration. The varying parameter, once again, is the variance of the measured noise. The graphs and tables in this section follow the same format as in the previous section. The goal here is to analyze the results obtained by the stationary and moving vehicles with the modified trajectory of the moving point described by Equation (5.7).

Analyzing simulation 9 by using Figure 5.11, notice the parameters for this simulation are found in Table 5.12. Since the variance of the measured noise is relatively small, the difference between the estimate of $\hat{x}$ and the actual position x, provided by the moving sensors and the stationary vehicles, is almost the same. It is interesting to note, that since the stationary vehicles are placed in optimal positions, the estimate $\hat{x}$ is as good as the moving vehicles, even though the trajectory is not placed at the center of all the vehicles.

Table 5.12 Parameters for Simulation 9 with Variance of Measured Noise = 0.000 053

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.000 053**
Stationary Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.000 053**



(a) $||x - \hat{x}||$

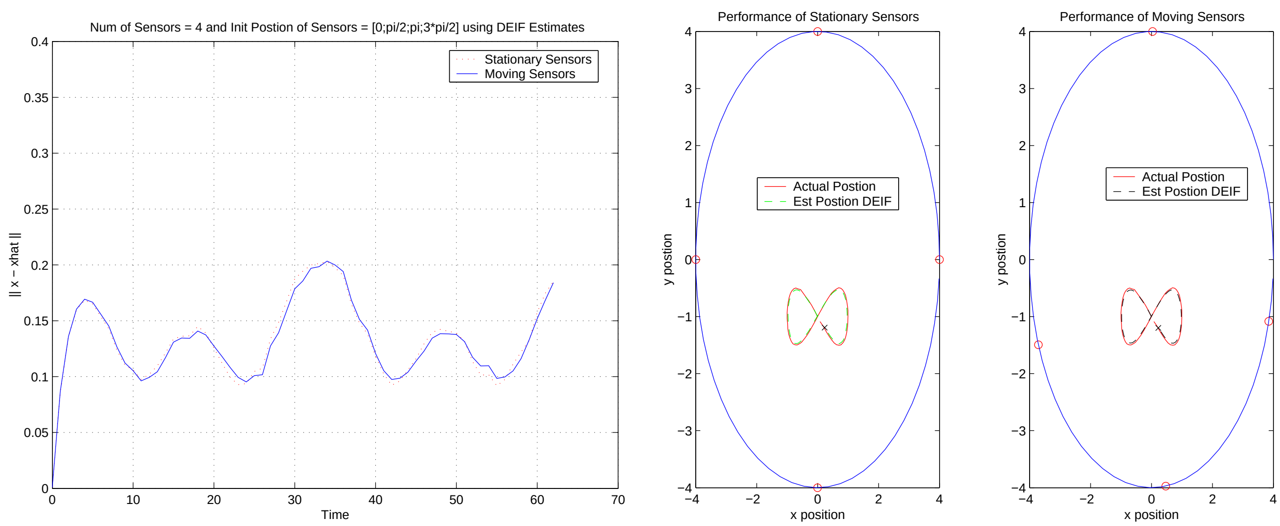
(b) Final Location of Sensors

Figure 5.11 Simulation 9: Stationary vs. Moving Sensors with Variance of Measured Noise = 0.000 053

Analyzing simulation 10 by using Figure 5.12, notice the parameters for this simulation are found in Table 5.13. Visually, the difference between simulations 9 and 10 is minimal, even though the variance of the measured noise has been increased by 10% of the actual $0.000\ 053$. Even though the variance of the measurement noise has been increased, both the stationary and moving vehicles have an error of less than 0.2. Therefore, not surprisingly, since the stationary vehicles are placed in the optimal position; at this point the results of the estimate $\hat{x}$ are the same for the moving vehicles and the stationary vehicles.

Table 5.13 Parameters for Simulation 10 with Variance of Measured Noise = 0.000 53

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.000 53**
Stationary Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.000 53**



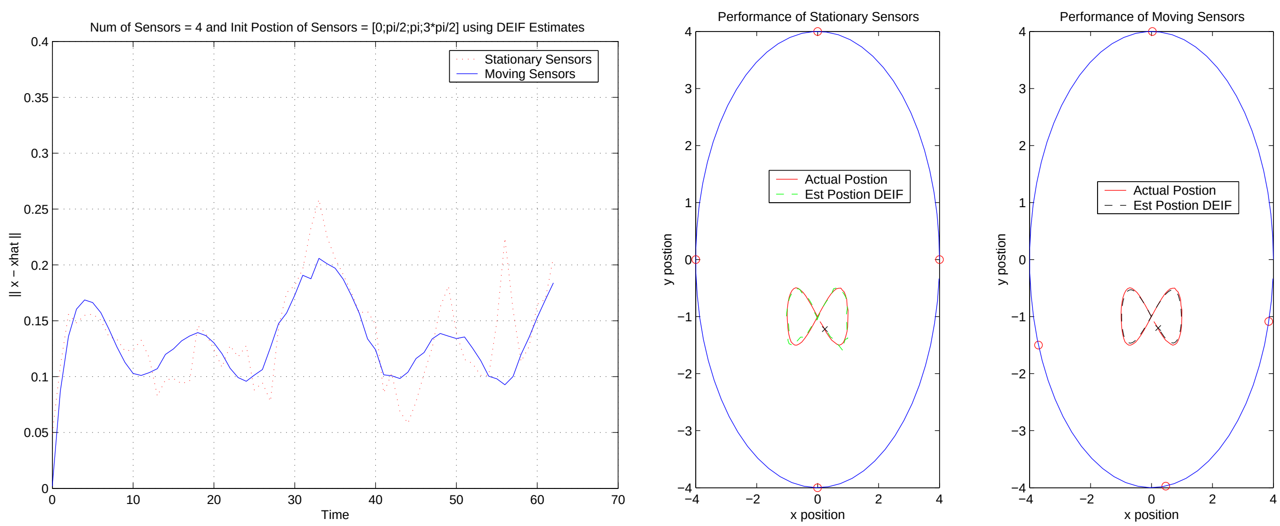
(a) $||x - \hat{x}||$ (b) Final Location of Sensors

Figure 5.12 Simulation 10: Stationary vs. Moving Sensors with Variance of Measured Noise=0.000 53

Analyzing simulation 11 by using Figure 5.13, notice the parameters for this simulation are found in Table 5.14. With the variance of the measured noise increased by 1000% of the orginal value $0.000\ 053$, the error $||x - \hat{x}||$ made by the stationary vehicles can be seen from the Figure 5.13. The error in $\hat{x}$ given by the stationary vehicles, at one point, gets as high as 0.26, while the error of the moving vehicles stays below 0.2 at all times. Note, this is the same result obtained in one of the previous simulations 1-8, when the trajectory of the moving point was described by Equation (5.3). Even with the increase of 1000% of the variance of the measurement noise and the change in the trajectory of the moving point, the moving vehicles still follow closely the trajectory of the moving point.

Table 5.14 Parameters for Simulation 11 with Variance of Measured Noise = 0.053

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.053**
Stationary Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.053**



(a) $||x - \hat{x}||$

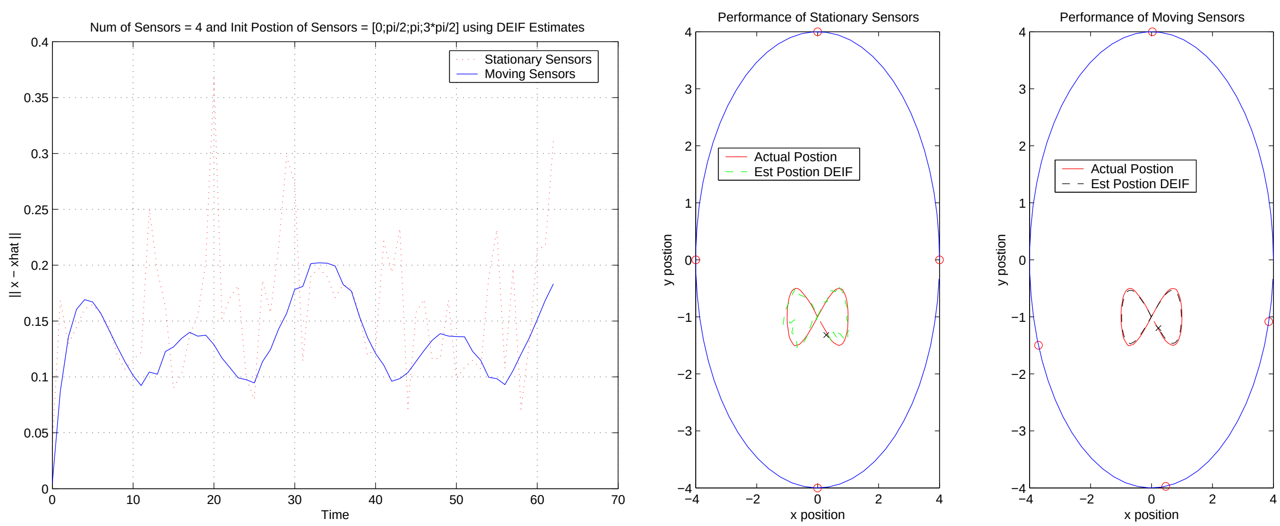
(b) Final Location of Sensors

Figure 5.13 Simulation 11: Stationary vs. Moving Sensors with Variance of Measured Noise = 0.053

Analyzing simulation 12 by using Figure 5.14, notice the parameters for this simulation are found in Table 5.15. Since the variance of the measured noise has been increased by $10\ 000\%$ of the original value in simulation 12, the stationary vehicles no longer obtain the best estimate $\hat{x}$ of the target. It is interesting to note that the error $||x - \hat{x}||$ at one point gets as high as 0.37 even though the stationary vehicles are placed in optimal configuration, such as in simulation 8, Figure 5.9. On the other hand, the moving vehicles still obtain a relatively good estimate of the location of the target. This proves numerically that the moving vehicles obtain the best estimate, regardless of the trajectory of the moving point, since the vehicles are always at an optimal configuration, described by Proposition 1.

Table 5.15 Parameters for Simulation 12 with Variance of Measured Noise = 0.53

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.53**
Stationary Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.53**



(a) $||x - \hat{x}||$ (b) Final Location of Sensors

Figure 5.14 Simulation 12: Stationary vs. Moving Sensors with Variance of Measured Noise = 0.53

5.3.2.1 Nonoptimal position versus optimal position

In Figures 5.15, the simulations of 9-12 are displayed accordingly along with the parameters in Table 5.16. The top portion of Figure 5.15 displays the error $||x - \hat{x}||$ of the stationary vehicles located in optimal configurations. The bottom portion of Figure 5.15, displays the error $||x - \hat{x}||$ of the moving vehicles located in optimal configurations.

With the aid of these results, it is easy to see that the moving vehicles obtain the best estimate, since the error is always less than 0.2. Comparing the results of the stationary vehicles, as the variance of the measured noise increases, the error of $||x - \hat{x}||$ also increases, therefore proving that the estimate $\hat{x}$ is independent of the trajectory of the moving target. The best estimate $\hat{x}$ is still obtained with the moving vehicles.

Table 5.16 Parameters for Simulations 9-12

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*
Moving Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**vary**
Stationary Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**vary**

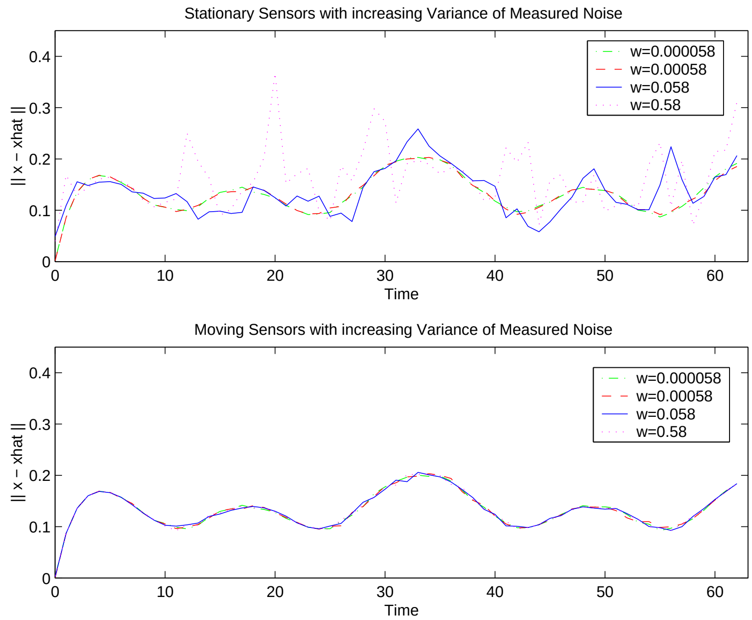


Figure 5.15 Simulations 9-12: Stationary Sensors vs. Moving Sensors

5.3.3 Modifying the parameters of DEIF

The DEIF filter obtains its estimates based on the predicted and actual measurements made. A way to modify which measurement is trusted more, the predicted or actual measurements, is done by modifying the matrix R, the measurement error covariance. As R in the DEIF approaches zero, the actual measurement $z(k)$ is trusted more and more, while the predicted measurement is trusted less and less. Therefore, it would be interesting to see how R affects the estimate $\hat{x}$ of the position of the moving target. Figure 5.16 displays the results obtained.

Table 5.17 Parameters for Simulations 13-16

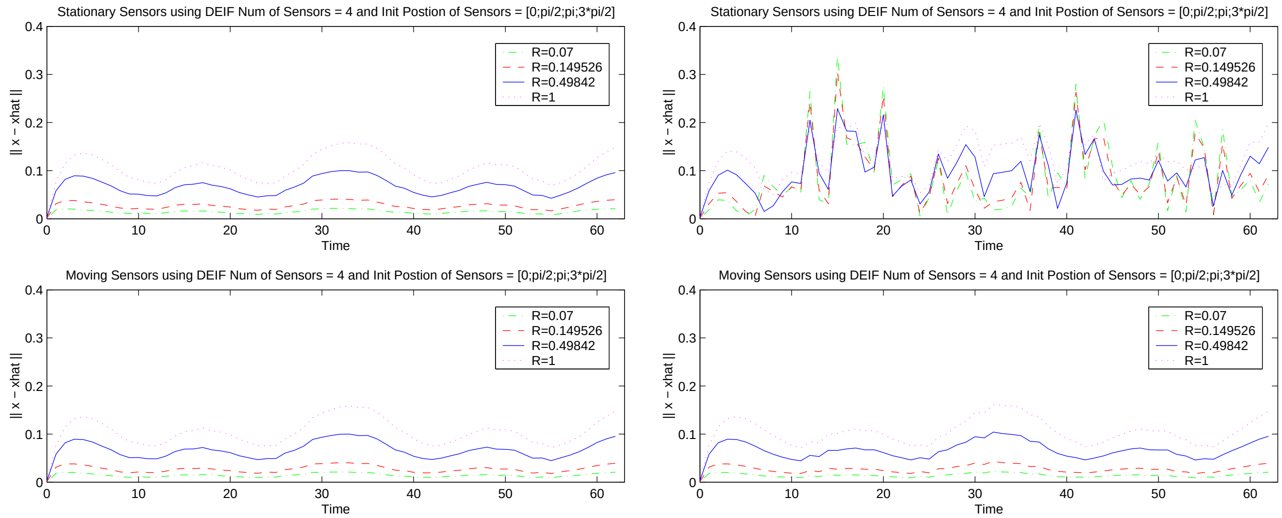
	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*	*Initial Guess for* $\begin{bmatrix} \hat{x} & \hat{y} \end{bmatrix}^T$
Moving Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.000 053**	$\begin{bmatrix} 0 & 0 \end{bmatrix}^T$
Stationary Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.000 053**	$\begin{bmatrix} 0 & 0 \end{bmatrix}^T$

Table 5.18 Parameters for Simulations 17-20

	Number of Sensors	*Initial Position of Sensors (radians)*	*Variance of Process Noise*	*Variance of Measured Noise*	*Initial Guess for* $\begin{bmatrix} \hat{x} & \hat{y} \end{bmatrix}^T$
Moving Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.53**	$\begin{bmatrix} 0 & 0 \end{bmatrix}^T$
Stationary Sensors	4	$\left[0; \frac{pi}{2}; \pi; \frac{3\pi}{2}\right]$	0.000 13	**0.53**	$\begin{bmatrix} 0 & 0 \end{bmatrix}^T$

In this section, the results obtained from simulations 13-20 will be compiled into graphs. This is done in order to facilitate the analysis on how R affects the estimate $\hat{x}$. Graphs are provided for both stationary and moving vehicles because it is important to be able to see the results obtained from the stationary vehicles and moving vehicles due to changing R. These graphs can be seen in Figure 5.16, the simulations of 13-20 are displayed accordingly, along with the parameters for each found in Tables 5.17 and 5.18.

From Figure 5.16, it can be seen that the moving sensors obtain the best estimate of the location of the moving target, since the error $||x - \hat{x}||$ is below 0.2 for the values of R tested. When the variance of the measurement noise is low in this case $0.000\ \ 053$, then trusting the actual measurements gives the best estimate for both the moving and the stationary vehicles. On the other hand, when the variance of the noise is high, for example 0.53, then the best result is obtained by the moving vehicles and trusting the actual measurements more, since the collected data is good. This proves that the actual measurements collected by moving sensors are the best measurements, since the moving vehicles are always in a their optimal positions.



(a) Variance of Measure Noise = 0.000 053

(b) Variance of Measure Noise = 0.53

Figure 5.16 Simulations 13-20: Stationary vs. Moving Sensors with Variance of Measured Noise

CHAPTER 6

CONCLUSION AND FUTURE RESEARCH

In this paper it was proven that, in order to obtain the best estimate of a moving target, it is desirable to have the sensors move to an optimal position described by Proposition 1 (p.17). In finding this solution to the problem of target tracking from a multisensor network, it has been proven in Chapter 2 that the *deployment of the agents* should maximize the probability of detection of the target to be tracked or provide more accurate estimations of the point source to be localized. Chapter 2 develops a method of obtaining the best possible estimation of nonrandom parameters. Since it is desireable to track a moving target with N number of sensors, and since a method of obtaining the best estimation of nonrandom parameters has been developed and fully understood, it follows that the best estimate of a moving target is achieved by allowing N number of sensors to move, using decentralized motion planning and decentralized estimation algorithms.

The solution to these problems has been built on motion control algorithms for the network and data fusion techniques which allowed decentralized implementations. Such a decentralized motion planning control algorithm has been described and proved to converge in Chapter 3. In Chapter 4, the possible estimation algorithms have been described, and the decentralized extended information filter was chosen to be the estimator used in the simulations. Chapter 5 contains a number of simulation supporting the statement that the best estimate of a moving target is achieved by having N number of moving sensors, using decentralized motion planning and decentralized estimation algorithms.

The goal for future research is to implement a scalable decentralized estimation. Since the DEIF algorithm uses collected information from all the sensors, it is decentralized in the sense that it does not need a fixed number of vehicles to function; the alogrithm adapts to its environment. It is also interesting to restrict, in simulations, the range of the sensors, making it closer to real-life enviornment.

REFERENCES

[1] M. Ridley, E. Nettleton, S. Sukkarieh, and H. Durrant-Whyte, "Tracking in decentralised air-ground sensing networks," The University of Sydney, School of Aerospace, Mechanical and Mechatronic Engineering, April 2002.

[2] B. Porat and A. Nehorai, "Localizing vapor-emitting sources by moving sensors," *IEEE Transactions On Signal Processing*, vol. 22, no. 4, pp. 1018–1021, April 1996.

[3] V. Isler, J. Spletzer, S. Khanna, and C. J. Taylor, "Target tracking with distributed sensors: The focus of attention problem," in *Intl. Conference on Intelligent Robots and Systems*, October 2003, pp. 792–798.

[4] Y. Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, New York, New York: John Wiley & Sons, 2001.

[5] G. Welch and G. Bishop, "An introduction to the Kalman filter," University of North Carolina at Chapel Hill, Department of Computer Science, April 2004.

[6] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*, Boca Raton, Fl: CRC Press LLC, 1998.