IMPROVING NAVIGATION THROUGH COOPERATION AND PATH PLANNING

By

JAYADEEP PABBISETTY

Bachelor of Science in Mechanical Engineering

Jawaharlal Nehru Technological University

Andhra Pradesh, INDIA

2012

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2015

IMPROVING NAVIGATION THROUGH COOPERATION
AND PATH PLANNING

Thesis Approved:

Girish Vinayak Chowdhary

Thesis Advisor

Adam James Rutkowski

Jamey Darin Jacob

Acknowledgments

I would like to express my heartfelt gratitude to my advisor, and thesis committee chairman Dr. Girish Chowdhary for his supervision, guidance, patience and motivation. Working under his guidance has been a wonderful learning experience both professionally and personally. I will definitely benefit from the experience working with him.

I would also like to thank Dr. Adam Rutkowski for his help and guidance throughout the entire course of this work. I have learned a lot from his experience in designing and managing the project.

I would like to thank Dr. Jamey Jacob for being on my thesis committee and providing various insights on my work.

I would like to express my gratitude to my mother Rukmini Kumari Pabbisetty and my elder sister Likhitha Pabbisetty, for their constant encouragement and faith in me.

I want to thank my colleagues at Oklahoma State University: Rakshit Allamraju, Benjamin Reish and Alex Suhren for their help and discussion all through the time of my work.

I thank Eglin Air Force Base for their support to the project.

Name: JAYADEEP PABBISETTY

Date of Degree: MAY, 2015

Title of Study: IMPROVING NAVIGATION THROUGH COOPERATION AND PATH PLANNING

Major Field: MECHANICAL AND AEROSPACE ENGINEERING

Abstract: In this thesis, the problem of two autonomous vehicles traveling in a two dimensional environment from an initial location to a predefined goal location without any absolute position reference is considered. The objective of the research is to study the effect of cooperation between the vehicles by considering the measurements such as relative range to help in improving the navigation state estimation and its effect on the observability of the system. Without the aid of an absolute position reference such as Global Positioning System (GPS) the navigation solution of the system will drift with time, since it can be shown that the state estimation problem is unobservable. The idea here is to reduce the navigation solution drift of the system with cooperation between the agents using measured relative information and to study its effect on the observability of the system while taking different paths. Various simulations are performed to compare the performance of the vehicles using different techniques to generate the trajectories. Simulations and theoretical results show that relative motion between the agents helps reduce the navigation drift of the agents when there is no absolute position reference.

Table of Contents

List of Figures

List of Tables

Table Page

CHAPTER 1

Introduction

Recent advancements in Unmanned Aerial Vehicles (UAV), especially Miniature Air Vehicles (MAV) technology, has made it possible to substitute human pilots to keep them out of dangerous aerial situations. A system of multiple UAVs can accomplish tasks which are a combination of several subtasks and can help improve coverage and robustness in performing such operations as compared to deploying a single UAV, whose capability is limited. These vehicles find use in a vast number of applications. They can be used in defense applications like using multiple MAVs for searching a particular area [2], search and rescue, disaster management, covert military missions, hazardous search, rescue operations and a variety of civil applications such as aerial photography and firefighting operations.

Navigation is vital for all manned or unmanned air missions. Without precise navigation it is not possible to accomplish many of the above mentioned tasks. Most navigation systems for UAV state estimation typically consist of an Inertial Measurement Unit (IMU) and a position sensor. They rely on fusing inertial information from accelerometers and gyroscopes with absolute position information from a position sensor. The resulting navigation system, often referred as an Inertial Navigation System (INS), are known to provide excellent estimates of vehicle states despite potential sensor biases and misalignment errors (see e.g. [3]). In the process of obtaining position estimates from the IMU, any errors in measurement are accumulated from point to point which leads to significant drift over time [4] and the position sensor is used to bound this drift. Due to its low cost and ease of availability, GPS is one of the most common absolute position sensors used. Hence, several GPS based INS solutions have been emerging [5]. However, adversarial effects from the external environment such as sky occlusion, GPS jamming, or internal effects such as hardware failure may inhibit the use of GPS. Navigation in such cases where GPS is denied is a challenging problem. This problem is further exacerbated when we have a group of autonomous, collaborating vehicles. As a result, several authors have explored relative INS that leverage locally visible features. The

Figure 1.1: Groups of airplanes moving in a formation [1] found at

navigation accuracy is increased using various sensors along with the INS [6, 7] in the case of a single vehicle. Most of this work has focused on the single agent case; yet, potential improvement in the quality of the navigation solution and robustness to GPS or local-feature visibility loss could be better tolerated when multiple agents collaborate [8].

Even though biological aerial systems like birds, flying insects, etc. cannot use GPS, they are capable of navigating in their environment. With this as an inspiration to the natural world Soloviev et al. did some work on the integration of inertial, visual, and compass information to aid in navigation for the GPS denied environments [9]. Most of the biological systems execute their navigation in groups, such as flocks of birds, colonies of ants, etc. Likewise, groups of airplanes flying in formations such as V-formation is shown in figure 1.1. Inspired by these biological systems, cooperative control in autonomous systems has become an active research area [10, 11]. Though these biological systems do not have luxury of using GPS or an absolute reference in the environment, they may be using cooperative behaviors to improve their navigation missions. Inspired by this, in this research, path planning and control strategies are investigated that optimize the navigation performance of a group of cooperating agents in the absence of absolute position reference. For example, Sharma and Taylor's work [8] indicates that a cooperative INS can improve state estimates. However, they, and others who have considered this problem, have not analyzed how relative motion between agents affects cooperative navigation. Specifically, we are concerned with the problem where the agents must travel from a known initial location to a predefined goal location without the benefit of an absolute position reference. Furthermore, the agents should reach the goal location

as accurately as possible under suitable time constraints. Without an absolute position reference, the navigation solution of the group could eventually drift over time. The path taken by each vehicle has an effect on how quickly the navigation solution drifts. Furthermore, any easily available additional relative measurements, such as relative range or relative velocity, fundamentally affect the observability of the problem. Observability analysis of linear time-varying systems is discussed in [12, 13] and this was used to study the GPS-aided INS for some specific maneuvers in [14].

In this research, the hypothesis is that the relative motion between the agents helps improve the navigation state estimation to reduce the drift, in the absence of an absolute position reference. This is tested by performing an observability analysis to show the effect of the path taken by each vehicle on the observability of the system. Simulations are run to demonstrate the observability analysis on several trajectory types. Here, a simplified version of the cooperative navigation analysis is presented by considering the problem in a one-dimensional environment and extended to two dimensions. In this research, the problem of path planning for vehicles with the help of relative motion between them is illustrated on two Dubins models with a measured relative range between said models, since the Dubins models can serve as an approximation of the UAV models in 2D. The Dubins model was first introduced by Lester Dubins in his paper in 1957 [15]. The Dubins model considers a nonholonomic ground vehicle which can only move forward, with a constant speed and constraints on the turn rate.

The various trajectory types that are considered in this research include straight line paths with zero heading angle for both the vehicles, straight line paths with different but constant heading angles for both the vehicles, sinusoidal trajectories where the vehicles move with in phase, out of phase, and different frequencies. Sampling based path planning algorithms are also demonstrated in this research. Rapidly-Exploring Random Trees (RRT) is a randomized data structure algorithm in the area of path planning designed for problems with nonholonomic constraints [16]. In RRT, the points are sampled randomly from the state space and the algorithm is written to connect these points to reach the goal location. RRT, Iterative sampling with RRT and RRT with cost function included for the selection of points in the graph are the sampling based algorithms used in this thesis to develop path planning for the vehicles. The comparison of mean square error in the final position estimation is done over 1000 Monte Carlo runs for each trajectory type considered in here. The straight line paths and sinusoidal paths can serve as an approximation for several other paths like polygonal path approximated by multiple straight line segments, circular paths approximated by sinusoidals, etc. The simulation results and E-Uniform observability analysis on these paths show that the navigation state estimation error is significantly less when the vehicles travel along sinusoidal

trajectories especially with different frequencies from each other and with out of phase than any other trajectory type. Some of the trajectory types we have not considered include complex polynomial trajectories, completely random trajectories, etc. but, they can be considered for future work. The goals here are to gain a fundamental understanding into the observability of the cooperative INS and leverage this understanding to develop path planning and control strategies that minimize or nearly minimize the drift of the estimated values from the true position values in the absence of an absolute position reference. The observability analysis and the simulations results shown in this research show that the relative motion between the agents has an effect on the navigation state estimation and the drift in the position estimation can be reduced by certain relative motion of the agents while tracking different paths.

CHAPTER 2

Literature Review

Relative navigation using locally visible features has been extensively studied (see e.g. [17, 18, 19]) in the multi-agent case. Two quadrotor helicopter systems which are capable of autonomous navigation in unknown environments using only the sensors without any prior map are developed in [17]. In [19] a new motion planning approach was presented to optimize a particular task for the robot motion planning. In all of these works navigation was done using locally visible features and did not concentrate on the scenarios where there is no absolute position reference. In [20], a framework was presented to improve the localization and mapping performance for a single UAV working in an unknown terrain in the absence of GPS. In this work the navigation of the agent is done using the Simultaneous Localization and Mapping technique (SLAM). They showed that the circular, S-shaped maneuvers and the combination of both improve the navigation state estimation performance than the straight line and steady level paths. An observability analysis of the inertial SLAM algorithm for path planning is also presented. Furthermore, in [21] it is also shown that an optimal path width can be achieved by generating zigzagged trajectories from an initial known goal location towards the final known goal location without direct position measurements. In both of these works, an optimal path in the absence of GPS was not found. In [22], a sequential Monte Carlo optimization technique was used to obtain more accurate position estimates between two known positions taking the help of landmarks, but a guaranteed global optimal trajectory was not found. Also, these works did not consider the benefit of cooperation nor the design of motion planning for improving observability in the absence of GPS. To reduce the drift and improve the navigation accuracy, fixed radio beacons and onboard vision sensors were considered [23, 24, 25, 26], but these techniques depend on the conducive environmental conditions. In [27] a cooperative localization technique is developed with various sensors and communication ranges, but with the help of external landmarks for the navigation of the robots. Using sensors to provide relative observations between two robots and observability analysis done for different relative observations show that the relative bearing is

the best observation for the robots [28]. All these works indicated the benefits of cooperation, however they did not analyze how relative motion between agents affects cooperative navigation.

The problem of motion planning and control has been considered in the context of relative navigation and target tracking by several authors [29, 30, 31, 32, 33]. Bishop et al. showed several necessary and sufficient conditions on the sensor target geometry to improve the localization performance by reducing the estimation uncertainty [29]. In [30], requirements on the angular separation of the sensors have been shown to achieve the best localization performance. In the presence of uncertainties using the sensors for vision, an algorithm for path planning which improves the navigation and mapping accuracy was presented in [32]. Trajectory planning was improved for autonomous robots using a framework developed in [33] for the information gathering tasks. Furthermore, Fox et al. have considered active localization approaches which take into account the ability of a mobile platform to position itself to acquire better measurements [34]. However, most of these works have not considered the benefits of cooperation among the vehicles to study the estimation performance of the system.

The observability of inertial navigation systems in the single agent case has also been studied [35, 36]. The problem is nonlinear and hence the trajectories of the system affect the observability. The problem of inconsistency of the state estimator in the SLAM technique is studied using the observability analysis [37] and an algorithm is presented to improve the estimator's consistency during SLAM. For multi-agent case, Sharma and Taylor have provided preliminary results on observability [8], however, their analysis did not consider the effect of vehicle's motion on the observability of the system. Observability and control of multi-agent networks has also been studied in the networked systems literature for estimating states of all of the agents in the network when only the states of a subset of agents are available for measurements (e.g. see [38, 39, 40, 41]). Also, the observability of multiple robots localizing themselves with the position of other robots taken into consideration as reference [42]. Sharma et al. showed the observability properties of the system for cooperative localization and some conditions on the connection of the robots that improve the observability of the system using bearings only cooperative localization [43]. In [44] the localization accuracy was improved for a group of unmanned vehicles performing cooperative localization in a GPS-denied environment based on the observability conditions obtained in [43]. Using the bearing-only cooperative localization an observability based path planning algorithm for the reduction in the uncertainty of state estimation was presented [45]. Sharma and Taylor in [8] proposed a cooperative INS that was partially decentralized. They showed that by exchanging the relative measurements between the vehicles and fusing the exchanged data with Extended Kalman Filter the estimation drift can be

constrained in the absence of GPS. In a GPS-denied environment, using the position and yaw states with the Extended Kalman Filter a relative navigation capability was provided in [46]. However in all of these works Sharma et al. have not considered the effect of the choice of vehicles' trajectories on the observability of the system. Here, in this thesis, linear analysis is shown to lend preliminary insights into the problem and then extended the work to nonlinear analysis that couples the motion of the vehicle with the observability. We argue that such an approach is mandated by the nonlinearities in the problem. In this thesis, the relative motion of the agents is taken into consideration to see how it affects the observability of the system. The effect of each vehicle's trajectory on the accuracy of the navigation state estimation is studied and analyzed.

2.1 Contribution

Though cooperative navigation of UAVs has been discussed in the literature by lots of authors in the absence of an absolute position reference, any work has not been done to study the relative motion between the agents. The contributions of this thesis are summarized as follows:

- Simulations results are shown for various vehicle trajectories like the straight line, sinusoidal, and sampling based techniques for the generation of the trajectory. The results show that the error between the true position values and the estimated values from the Extended Kalman Filter (EKF) is reduced significantly when the agents move in sinusoidal paths with different frequencies from each other in the absence of an absolute position reference.

- E-Uniform observability analysis is performed to study the effect on the choice of vehicles trajectories and relative motion between them on the observability of the system. From the observability analysis, we can see that the path taken by each vehicle affects the observability of the system and the space of unobservability (more details on this are given in section 3.3.1) is lower for certain paths taken by the vehicles.

CHAPTER 3

Problem Formulation

In this section the dynamic model of the vehicles is considered in 1D and 2D environments. First, for better understanding of the problem the agents are considered in 1D. But the 1D case is linear and it does not take into consideration the effect of trajectories on vehicles' estimation performance and the observability analysis. Hence the analysis is extended into 2D where the problem is non linear. The state space representation, estimation of the states using EKF and the observability analysis are shown for the Dubins model in 2D.

Figure 3.1 shows the general problem scenario where two agents have pre-defined initial locations and desired goal locations. Both vehicles travel at a constant and equal speed and are required to reach the goal locations in 140 seconds in the absence of an absolute position reference.

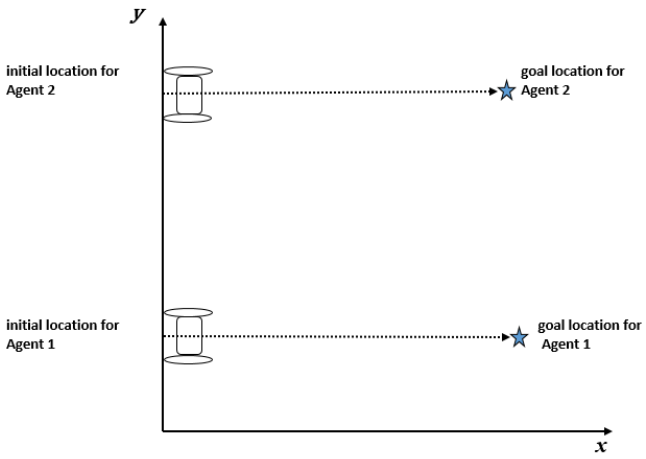


Figure 3.1: General problem scenario

3.1 One-dimensional case with relative range and relative velocity measurements

This section investigates the effect of vehicle trajectories on navigation performance for two vehicles with and without relative range and relative velocity measurements in 1D. In the continuous time model the system model is written in the following manner where x is the displacement, y is the measurement of the vehicle, $\dot{x}$ represents the speed of the vehicle and $\ddot{x}$ represents the turn rate of the vehicle (subscript indicates the vehicle number). The control inputs are assumed to be the accelerations of the vehicles, a_1 and a_2. Since in the real world the accelerometers are not perfect we assume that they are corrupted by white Gaussian noise with a standard deviation of 0.01 units.

$$\begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x_1} \\ x_2 \\ \dot{x_2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \tag{3.1}$$

The measurement equation may be written as follows, with the measurements being the relative range and relative velocity between the vehicles.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x_1} \\ x_2 \\ \dot{x_2} \end{bmatrix} \tag{3.2}$$

For a linear time continuous system of the following form

$$\dot{X} = Ax + Bu \tag{3.3}$$

$$Y = Cx + Du \tag{3.4}$$

The observability matrix, $\mathcal{O}$ may be written as

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{3.5}$$

where n is the number of states for the system shown in equation 3.3. The observability matrix for the two vehicles moving in 1D whose state space representation is shown in equations 3.1 and 3.2 may be written as

$$\mathcal{O} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.6}$$

From the above observability matrix we can see the rank of the matrix is 2 which is less than n e.g. 4, the number of states, thus the system is clearly unobservable which shows that without the absolute position reference the position estimates of the vehicles will drift with time. Also, in the one dimensional case the observability analysis does not take into consideration the effect of trajectories on the vehicle's estimation performance.

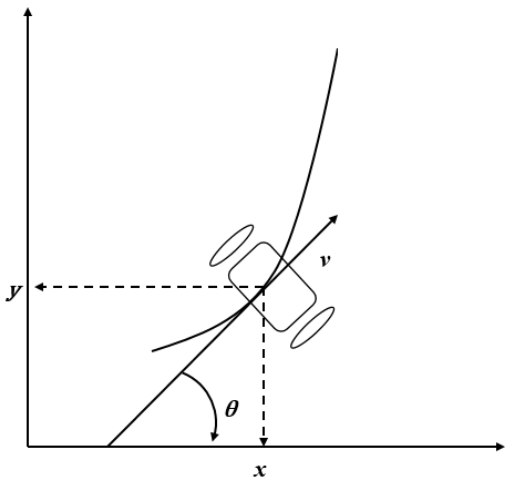


Figure 3.2: Model of a Dubins car

3.2 Two-dimensional case with speed and heading rate measurement

In this section, let us consider the problem of two agents navigating in a two dimensional environment from one point to another, in particular we consider the agents as two Dubins models shown in figure 3.2. Let x and y be the displacements of the Dubins model along the x and y coordinate axes respectively. Also let V be the speed of the vehicle, θ be the heading angle of the vehicle, and ω be the turn rate of the Dubin's model.

The kinematic equations of motion for the Dubins model are as follows [47]:

$$\dot{x} = V\cos(\theta) \tag{3.7}$$

$$\dot{y} = V\sin(\theta) \tag{3.8}$$

$$\dot{\theta} = \omega \tag{3.9}$$

The control inputs are assumed to be the velocity and turn rate. The control input matrix would be:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} V \\ \omega \end{bmatrix} \tag{3.10}$$

The above equations are used for the state space formulation of the two agents in a two dimensional world and the discrete time state space equations are used for the estimation of the states using the extended Kalman filter.

This section investigates the effect of vehicle trajectories on navigation performance for multiple vehicles using inter-agent range and/or bearing measurements. First, let us consider a system architecture with two Dubins cars and a centralized navigation filter in 2D. Also, let us assume that the vehicles can measure their speed and angular rate using sensors such as wheel encoders. In discrete time, the system model for the two Dubins cars with the control input as shown in equation (3.1) can be written in the following manner, where x, y, and θ are the x-position, y-position, and heading of a vehicle, v is the vehicle speed, ω is the vehicle turn rate, Δt is the time-step (i.e. the time between t^{k-1} and t^k, the subscript indicates the vehicle number, the superscript indicates the time index, and a tilde ˜ is used to indicate measured quantities).

$$\mathbf{x}^k = \begin{bmatrix} x_1^k \\ y_1^k \\ \theta_1^k \\ x_2^k \\ y_2^k \\ \theta_2^k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1^{k-1} \\ y_1^{k-1} \\ \theta_1^{k-1} \\ x_2^{k-1} \\ y_2^{k-1} \\ \theta_2^{k-1} \end{bmatrix} + \begin{bmatrix} \cos(\theta_1^{k-1})\Delta t & 0 & 0 & 0 \\ \sin(\theta_1^{k-1})\Delta t & 0 & 0 & 0 \\ 0 & \Delta t & 0 & 0 \\ 0 & 0 & \cos(\theta_2^{k-1})\Delta t & 0 \\ 0 & 0 & \sin(\theta_2^{k-1})\Delta t & 0 \\ 0 & 0 & 0 & \Delta t \end{bmatrix} \begin{bmatrix} \tilde{V}_1^{k-1} \\ \tilde{\omega}_1^{k-1} \\ \tilde{V}_2^{k-1} \\ \tilde{\omega}_2^{k-1} \end{bmatrix} \tag{3.11}$$

The measurement equation may be written as follows, where ρ is the inter-vehicle range and β is the bearing of one vehicle with respect to another.

$$\begin{bmatrix} \tilde{\rho}^k \\ \tilde{\beta}_1^k \\ \tilde{\beta}_2^k \end{bmatrix} = \begin{bmatrix} \sqrt{(x_2^k - x_1^k)^2 + (y_2^k - y_1^k)^2} \\ \arctan\left(\frac{y_2^k - y_1^k}{x_2^k - x_1^k}\right) - \theta_1^k \\ \arctan\left(\frac{y_1^k - y_2^k}{x_1^k - x_2^k}\right) - \theta_2^k \end{bmatrix} \tag{3.12}$$

3.2.1 Estimation

An Extended Kalman Filter (EKF) is used to estimate the states (position and orientation) of the vehicles using input from speed and angular rate sensors, and inter-agent range and/or bearing measurements. Proper filter design requires analyzing how errors in the states (related to the estimation process) evolve over time. The followings are the equations used in the estimation process of the Kalman filter.

$$\begin{aligned} \delta x_k &= \delta(x_{k-1} + V_{k-1}\cos(\theta_{k-1})\Delta t) \\ &= \delta x - Vs\theta\Delta t\delta\theta + c\theta\Delta t\delta v \end{aligned} \tag{3.13}$$

$$\begin{aligned}\delta y_k &= \delta(y_{k-1} + V_{k-1}\sin(\theta_{k-1})\Delta t) \\ &= \delta y + Vc\theta\Delta t\delta\theta + s\theta\Delta t\delta v\end{aligned} \tag{3.14}$$

$$\begin{aligned}\delta\theta_k &= \delta(\theta_{k-1} + \omega_{k-1}\Delta t) \\ &= \delta\theta + \Delta t\delta\omega\end{aligned} \tag{3.15}$$

We can write this in matrix form as

$$\begin{bmatrix}\delta x^k \\ \delta y^k \\ \delta\theta^k\end{bmatrix} = \begin{bmatrix}1 & 0 & -\tilde{V}^{k-1}\Delta t s\theta^{k-1} \\ 0 & 1 & \tilde{V}^{k-1}\Delta t c\theta^{k-1} \\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\delta x^{k-1} \\ \delta y^{k-1} \\ \delta\theta^{k-1}\end{bmatrix} + \begin{bmatrix}c\theta^{k-1}\Delta t & 0 \\ s\theta^{k-1}\Delta t & 0 \\ 0 & \Delta t\end{bmatrix}\begin{bmatrix}w_v \\ w_\omega\end{bmatrix} \tag{3.16}$$

$$\delta\mathbf{x}^k = \mathbf{\Phi}\delta\mathbf{x}^{k-1} + \mathbf{\Gamma}\mathbf{w}^{k-1}$$

Note that it is not actually necessary to estimate the state errors themselves, rather, this is used to keep track of the covariance in the state estimates. Thus, the covariance matrix, $\mathbf{P}$ is propagated as

$$\mathbf{P}^k = \mathbf{\Phi}\mathbf{P}^{k-1}\mathbf{\Phi}^T + \mathbf{Q} \tag{3.17}$$

where the process noise matrix, $\mathbf{Q}$ is given by

$$\mathbf{Q} = \mathbf{\Gamma}\mathbf{\Sigma}_u\mathbf{\Gamma}^T \tag{3.18}$$

and

$$\mathbf{\Sigma}_u = \begin{bmatrix}{\sigma_v}^2 & 0 \\ 0 & {\sigma_\omega}^2\end{bmatrix} \tag{3.19}$$

where σ_v^2 and σ_ω^2 are the covariances in the velocity and turn rate respectively. Linearizing about the current state estimate yields the following measurement matrix.

$$\mathbf{H} = \frac{\partial h}{\partial \mathbf{x}}|_{\mathbf{x}=\hat{\mathbf{x}}} = \begin{bmatrix}\frac{-\Delta x}{\rho} & \frac{-\Delta y}{\rho} & 0 & \frac{\Delta x}{\rho} & \frac{\Delta y}{\rho} & 0 \\ \frac{\Delta y}{\rho^2} & \frac{-\Delta x}{\rho^2} & -1 & \frac{-\Delta y}{\rho^2} & \frac{\Delta x}{\rho^2} & 0 \\ \frac{-\Delta y}{\rho^2} & \frac{\Delta x}{\rho^2} & 0 & \frac{\Delta y}{\rho^2} & \frac{-\Delta x}{\rho^2} & -1\end{bmatrix} \tag{3.20}$$

In this case where only inter-agent range is measured, only the first row of $\mathbf{H}$ is used. Likewise, if only relative bearings are measured, then only the last two rows are used.

3.2.2 Observability Analysis

To determine the observability of the system, the nonlinear observability rank criterion developed by Hermann and Krener in [48] is used. Sharma et al. [43] used the observability analysis for the

cooperation of MAVs using bearing-only measurements. The observability matrix, $\mathcal{O}$ for a nonlinear system may be written as

$$\mathcal{O} = \begin{bmatrix} \nabla\mathcal{L}_f^0 h \\ \nabla\mathcal{L}_f^1 h \\ \vdots \end{bmatrix} \tag{3.21}$$

Where the p-th Lie derivative is defined in a recursive manner as

$$\mathcal{L}_f^p h = \nabla(\mathcal{L}_f^{p-1} h) \cdot f$$

$$\mathcal{L}^0 h = h$$

We can build up the observability matrix in the following sequence.

$$\nabla\mathcal{L}^0 h = \nabla h = \mathbf{H}$$

$$\mathcal{L}_f^1 h = \nabla(\mathcal{L}^0 h)\cdot f = \mathbf{H}\cdot\mathbf{x} = \begin{bmatrix} \frac{(-x_1+x_2)\Delta x+(-y_1+y_2)\Delta y}{\rho} \\ \frac{(-y_1+y_2)\Delta x+(x_1-x_2)\Delta y}{\rho^2} - \theta_1 \\ \frac{(y_1-y_2)\Delta x+(-x_1+x_2)\Delta y}{\rho^2} - \theta_2 \end{bmatrix} = \begin{bmatrix} \frac{(\Delta x)^2+(\Delta y)^2}{\rho} \\ \frac{(\Delta x)(\Delta y)-(\Delta x)(\Delta y)}{\rho} - \theta_1 \\ \frac{-(\Delta x)(\Delta y)+(\Delta x)(\Delta y)}{\rho} - \theta_2 \end{bmatrix} = \begin{bmatrix} \frac{\rho^2}{\rho} \\ -\theta_1 \\ -\theta_2 \end{bmatrix} = \begin{bmatrix} \rho \\ -\theta_1 \\ -\theta_2 \end{bmatrix}$$

Using the gradient of the range with respect to the state that was already written in (3.20), we can immediately write.

$$\nabla\mathcal{L}_f^1 h = \begin{bmatrix} \frac{-\Delta x}{\rho} & \frac{-\Delta y}{\rho} & 0 & \frac{\Delta x}{\rho} & \frac{\Delta y}{\rho} & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\mathcal{L}_f^2 h = \nabla(\mathcal{L}_f^1 h)\cdot f = \nabla(\mathcal{L}_f^1 h)\cdot\mathbf{x} = \begin{bmatrix} \rho \\ -\theta_1 \\ -\theta_2 \end{bmatrix}$$

Now, it is obvious that

$$\nabla\mathcal{L}_f^p h = \begin{bmatrix} \frac{-\Delta x}{\rho} & \frac{-\Delta y}{\rho} & 0 & \frac{\Delta x}{\rho} & \frac{\Delta y}{\rho} & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad \forall p \geq 1$$

Thus, we don't need to include any terms with Lie derivative order higher than one in our observability matrix since they will be linearly dependent with existing terms. We may now write the

observability matrix as

$$\mathcal{O} = \begin{bmatrix} \nabla \mathcal{L}_f^0 h \\ \\ \nabla \mathcal{L}_f^1 h \end{bmatrix} = \begin{bmatrix} \frac{-\Delta x}{\rho} & \frac{-\Delta y}{\rho} & 0 & \frac{\Delta x}{\rho} & \frac{\Delta y}{\rho} & 0 \\ \\ \frac{\Delta y}{\rho^2} & \frac{-\Delta x}{\rho^2} & -1 & \frac{-\Delta y}{\rho^2} & \frac{\Delta x}{\rho^2} & 0 \\ \\ \frac{-\Delta y}{\rho^2} & \frac{\Delta x}{\rho^2} & 0 & \frac{\Delta y}{\rho^2} & \frac{-\Delta x}{\rho^2} & -1 \\ \\ \frac{-\Delta x}{\rho} & \frac{-\Delta y}{\rho} & 0 & \frac{\Delta x}{\rho} & \frac{\Delta y}{\rho} & 0 \\ \\ 0 & 0 & -1 & 0 & 0 & 0 \\ \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{3.22}$$

As part of the fundamental theorem of linear algebra we know that the row rank and the column rank of a matrix are always the same. We can immediately see that column 4 is the negative of column 1, and column 5 is the negative of column 2, thus the rank of the observability matrix is 4. In conclusion, this system is not observable. From the Lie derivative observability analysis, we can observe that this does not take into consideration the choice of the vehicle's trajectory to affect the navigation state estimation performance of the system.

3.3 E-Uniform observability analysis

3.3.1 Preliminaries

Since the Lie derivative observability analysis does not take into account the choice of the vehicle's trajectory on the performance of the system, we consider E-Uniform observability analysis. The E-Uniform observability analysis takes into consideration the choice of the control inputs in the observability analysis of the system. Thus, clearly the path taken by each of the vehicle affects the observability of the system. Hence to verify the hypothesis of the problem that is considered, E-Uniform observability analysis of the system is performed while traversing different paths or trajectories, and analyze how the observability of the system is affected by the choice of the trajectory. The following preliminaries are necessary for better understanding of the analysis that is done in the further sections.

Definitions

Consider a nonlinear system for which the dynamics are represented by

$$\begin{aligned} \dot{x} &= f(u, x) \\ y &= h(x) \end{aligned} \tag{3.23}$$

Definition 3.1 *([49],chapter 2) Let x be the initial state of the system and x^u be the trajectory associated to the initial state x and to the input u. The system represented in* (3.23) *said to be Observable if for every pair of different initial states there exists an input $u \in L^\infty([0,T],U)$ such that $h(x^u(\cdot))$ is not identically equal to $h(\bar{x}^u(\cdot))$ on $[0, T(x, \bar{x}, u)]$ where $T(x, \bar{x}, u) = min\{T(u, x), T(u, \bar{x})\}$.*

Such an input, u is said to distinguish the considered initial states $x, \bar{x}$ on $[0, T]$.

Definition 3.2 *([49],chapter 2) An input which distinguishes every pair of different initial states on $[0, T]$ is called as a Universal input on $[0, T]$.*

Definition 3.3 *([49],chapter 2) If E is any Borelian subset of U then the system shown in* (3.23) *is said to be E Uniformly Observable if and only if for every $T > 0$ and every $u \in L^\infty([0, T, E])$, u is a Universal input on [0,T].*

In the following analysis for all the nonlinear cases considered we show that there exists at least some set of initial conditions, x and $\bar{x}$ for which the E-uniform observability does not hold true. We show that $h(x^u(\cdot))$ is equal to $h(\bar{x}^u(\cdot))$ at least for certain given initial conditions x and $\bar{x}$.

Space of unobservability

The space of unobservability, Φ is defined as the set of all initial conditions for a given input where the E-Uniform observability analysis does not hold true.

$$\Phi = \{h(x^u(\cdot)) = h(\bar{x}^u(\cdot)) \mid \forall [x_{1o}, y_{1o}]^T, [x_{2o}, y_{2o}]^T, [\bar{x}_{1o}, \bar{y}_{1o}]^T, [\bar{x}_{2o}, \bar{y}_{2o}]^T \in \Re^2\} \tag{3.24}$$

Analysis

In the following sections, to demonstrate the E-uniform observability analysis we do the following:

- Pick an input, $u \in L^\infty([0, T, E])$
- Find the set of initial conditions for which the $h(x^u)$ is equal to $h(\bar{x}^u)$
- Define the space of unobservability and compare this in all the cases

3.3.2 Linear case

Let us consider a simple linear system to study the E-Uniform observability analysis. The system dynamics is represented by

$$\begin{aligned} f(X) = \dot{X} &= Ax + Bu \\ Y &= Cx \end{aligned} \tag{3.25}$$

Let us apply the system with two different initial conditions x_{1o} and x_{2o} with the same input u. Then the system dynamics are written as follows

$$f(X_1) = \dot{X_1} = Ax_1 + Bu$$
$$f(X_2) = \dot{X_2} = Ax_2 + Bu$$

Consider the outputs Y_1 and Y_2.

$$Y_1 = Cx_1$$
$$Y_2 = Cx_2$$

The solution to the above linear system is given by

$$\begin{aligned} X_1(t) &= e^{At}X_{1o} + \int_0^t e^{A(t-\tau)}Bu(\tau)\,d\tau \\ X_2(t) &= e^{At}X_{2o} + \int_0^t e^{A(t-\tau)}Bu(\tau)\,d\tau \end{aligned} \tag{3.26}$$

Consider

$$\begin{aligned} Y_1 - Y_2 &= C\left[e^{At}X_{1o} + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau\right] - C\left[e^{At}X_{2o} - \int_0^t e^{A(t-\tau)}Bu(\tau)\,d\tau\right] \\ &= C\left[e^{At}(X_{1o} - X_{2o}) + B\int_0^t e^{A(t-\tau)}(u(\tau) - u(\tau))\,d\tau\right] \\ &= Ce^{At}\left[X_{1o} - X_{2o}\right] \\ &= C\left[1 + At + \frac{(At)^2}{2} + \frac{(At)^3}{6} + \dots\right]\left[X_{1o} - X_{2o}\right] \\ &= \left[C + (CA)t + (CA^2)\frac{t^2}{2} + (CA^3)\frac{t^3}{6} + \dots\right]\left[X_{1o} - X_{2o}\right] \\ &= \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix}\left[1 + t + \frac{t^2}{2} + \frac{t^3}{6} + \dots\right]\left[X_{1o} - X_{2o}\right] \end{aligned} \tag{3.27}$$

From the above equation we can observe that since time 't' is greater than zero and $[X_{1o} - X_{2o}]$ is not equal to zero then we say that the system represented in (3.25) is observable if the matrix $\left[C\, CA^2\, CA^3 ...\right]$ is fully ranked which leads us to the condition of traditional observability for the linear systems. Thus, the choice of the vehicle's trajectory does not effect the system's observability in the linear case.

3.3.3 Dubins model

The concept of E-Uniform observability is now considered for the Dubins model for the illustration of our problem. Let $\dot{X} = f(X, u)$ be a nonlinear system such that $f\colon D \mapsto R^n$ is uniformly Lipschitz on D. If x(t) and y(t) are the solutions to the $f(X)$ with different initial conditions $x(t_0) = x_0$, $y(t_0) = y_0$ then ([49],chapter 2)

$$\begin{aligned} x(t) &= x_0 + \int_{t_0}^{t_1} f(x(s))\, ds \\ y(t) &= y_0 + \int_{t_0}^{t_1} f(y(s))\, ds \end{aligned} \tag{3.28}$$

The state space representation for two Dubins models moving relative to each other in a 2D environment is considered in the following equations 3.29, 3.30 and 3.31. Here x, y and θ are the x-position, y-position and heading of the vehicles, V is the speed of the vehicles, and ω is the vehicle's turn rate (subscript indicates the vehicle number)

$$f(X) \quad = \begin{bmatrix} \dot{x_1} \\ \dot{y_1} \\ \dot{\theta_1} \\ \dot{x_2} \\ \dot{y_2} \\ \dot{\theta_2} \end{bmatrix} = \begin{bmatrix} V\cos\theta_1 \\ V\sin\theta_1 \\ \omega_1 \\ V\cos\theta_2 \\ V\sin\theta_2 \\ \omega_2 \end{bmatrix} \tag{3.29}$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} V_1 \\ \omega_1 \\ V_2 \\ \omega_2 \end{bmatrix} \tag{3.30}$$

$$h(X) \quad = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.31}$$

3.3.4 General case

In this section we consider the E-Uniform observability analysis for two Dubins models moving in a 2D environment. The agents have the relative range between them as the measurement. The state space representation for such a case is shown in equations 3.29, 3.30 and 3.31. The solution to the nonlinear system represented in (3.23) is given in (3.28). Here we consider two Dubins models moving relative to each other whose dynamics are represented in (3.29). For the general case, we rewrite the solution given in (3.28) as follows

$$x_1 = x_{1o} + \int_{t_0}^{t_1} V \cos\theta_1 \, dt \tag{3.32}$$

$$x_2 = x_{2o} + \int_{t_0}^{t_1} V \cos\theta_2 \, dt \tag{3.33}$$

$$y_1 = y_{1o} + \int_{t_0}^{t_1} V \sin\theta_1 \, dt \tag{3.34}$$

$$y_2 = y_{2o} + \int_{t_0}^{t_1} V \sin\theta_2 \, dt \tag{3.35}$$

Consider

$$\begin{aligned} x_2 - x_1 &= x_{2o} - x_{1o} + \int_{t_0}^{t_1} V\cos\theta_2 - V\cos\theta_1 \, dt \\ &= x_{2o} - x_{1o} + V\int_{t_0}^{t_1} (\cos\theta_2 - \cos\theta_1) \, dt \\ y_2 - y_1 &= y_{2o} - y_{1o} + \int_{t_0}^{t_1} V\sin\theta_2 - V\sin\theta_1 \, dt \\ &= y_{2o} - y_{1o} + V\int_{t_0}^{t_1} (\sin\theta_2 - \sin\theta_1) \, dt \end{aligned} \tag{3.36}$$

The measurement equations when we apply two different initial conditions with the same input are

$$\begin{aligned} h(X_u) &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\ h(\bar{X}_u) &= \sqrt{(\bar{x_2} - \bar{x_1})^2 + (\bar{y_2} - \bar{y_1})^2} \end{aligned} \tag{3.37}$$

Consider

$$\begin{aligned} (x_2 - x_1)^2 + (y_2 - y_1)^2 = &\Big[(x_{2o} - x_{1o}) + V\int_{t_0}^{t_1} (\cos\theta_2 - \cos\theta_1) \, dt \Big]^2 \\ &+ \Big[(y_{2o} - y_{1o}) + V\int_{t_0}^{t_1} (\sin\theta_2 - \sin\theta_1) \, dt \Big]^2 \end{aligned} \tag{3.38}$$

Similarly when we apply, with different initial conditions, $\bar{X}$ and the same input u, consider the following

$$(\bar{x_2} - \bar{x_1})^2 + (\bar{y_2} - \bar{y_1})^2 = \left[(\bar{x}_{2o} - \bar{x}_{1o}) + V \int_{t_0}^{t_1} (\cos\theta_2 - \cos\theta_1)\ dt\right]^2 + \left[(\bar{y}_{2o} - \bar{y}_{1o}) + V \int_{t_0}^{t_1} (\sin\theta_2 - \sin\theta_1)\ dt\right]^2$$

Consider the two measurement equations

$$h(X_u) - h(\bar{X}_u) = \left(\sqrt{(A + I_1)^2 + (B + I_2)^2} - \sqrt{(C + I_1)^2 + (D + I_2)^2}\right) \tag{3.39}$$

where

$$A = x_{2o} - x_{1o} \tag{3.40}$$

$$B = y_{2o} - y_{1o} \tag{3.41}$$

$$C = \bar{x}_{2o} - \bar{x}_{1o} \tag{3.42}$$

$$D = \bar{y}_{2o} - \bar{y}_{1o} \tag{3.43}$$

$$I_1 = V \textstyle\int_{t_0}^{t_1} (\cos\theta_2 - \cos\theta_1)\ dt \tag{3.44}$$

$$I_2 = V \textstyle\int_{t_0}^{t_1} (\sin\theta_2 - \sin\theta_1)\ dt \tag{3.45}$$

The solutions for the above equation would be obtained by equating (3.39) to zero.

$$\left(\sqrt{(A + I_1)^2 + (B + I_2)^2} - \sqrt{(C + I_1)^2 + (D + I_2)^2}\right) = 0$$

The solution for the above equation is

$$\begin{aligned} &\sqrt{(A + I_1)^2 + (B + I_2)^2} = \sqrt{(C + I_1)^2 + (D + I_2)^2} \\ &\implies (A + I_1)^2 + (B + I_2)^2 = (C + I_1)^2 + (D + I_2)^2 \\ &\implies A^2 + B^2 + 2(AI_1 + BI_2) = C^2 + D^2 + 2(CI_1 + DI_2) \end{aligned} \tag{3.46}$$

To solve this equation properly for constants A, B, C, D, one must isolate the time invariant and time varying parts of the above equation. In other words, the time invariant part leads to

$$A^2 + B^2 = C^2 + D^2 \tag{3.47}$$

The coefficients on the (time varying) term I_1 lead to the following condition

$$A = C \tag{3.48}$$

And finally, equating the coefficient of the I_2 on both sides leads to the following

$$B = D \tag{3.49}$$

Since the last two equations satisfy the first directly, the first equation is not an independent condition, and thus all of the solutions to (3.46) are given by $A = C$ and $B = D$. The space of unobservability for the general case can be written as

$\Phi = \{(x_{2o} - x_{1o}) = (\bar{x}_{2o} - \bar{x}_{1o}) \mid \forall x_{1o}, x_{2o}, \bar{x}_{1o}, \bar{x}_{2o} \in \Re\} \cap \{(y_{2o} - y_{1o}) = (\bar{y}_{2o} - \bar{y}_{1o}) \mid \forall y_{1o}, y_{2o}, \bar{y}_{1o}, \bar{y}_{2o} \in \Re\}$

Thus, the system would be unobservable for any trajectory considered for the initial conditions shown in the set Φ. In the future sections we analyze the space of unobservability for each different trajectory taken by the vehicle.

3.3.5 Special Case 1: Straight line

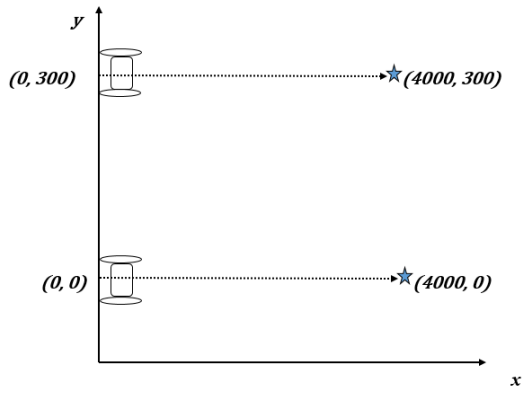


Figure 3.3: Straight line case with zero heading angle of the agents

In this case let us consider that the agents traveling along two straight lines as shown in the figure 3.3, hence there is no deliberate relative motion between the agents. The heading angles of the agents here are zero degrees. The result for the straight line case is verified by substituting $\theta_1 = \theta_2 = 0$ in (3.39). By substituting θ_1 and θ_2 to be zero in (3.44) and (3.45) we obtain the following.

$$\begin{aligned} I_1 &= 0 \\ I_2 &= 0 \end{aligned} \tag{3.50}$$

Substituting back into (3.46) leads to the following condition for unobservability.

$$A^2 + B^2 = C^2 + D^2 \tag{3.51}$$

$$\sqrt{(x_{2o} - x_{1o})^2 + (y_{2o} - y_{1o})^2} = \sqrt{(\bar{x}_{2o} - \bar{x}_{1o})^2 + (\bar{y}_{2o} - \bar{y}_{1o})^2} \tag{3.52}$$

We can conclude here that any pair of initial conditions is indistinguishable from any other pair with the same initial separation distance. Thus, the system is not E-Uniformly observable for the set of initial conditions given by

$\Phi = \{\sqrt{(x_{2o} - x_{1o})^2 + (y_{2o} - y_{1o})^2} = \sqrt{(\bar{x}_{2o} - \bar{x}_{1o})^2 + (\bar{y}_{2o} - \bar{y}_{1o})^2}$

$| \ \forall [x_{1o}, y_{1o}]^T, [x_{2o}, y_{2o}]^T, [\bar{x}_{1o}, \bar{y}_{1o}]^T, [\bar{x}_{2o}, \bar{y}_{2o}]^T \in \Re^2\}$.

If instead we substitute $\theta_1 = \theta_2 = k$ where k is a constant, we obtain the same result for I_1 and I_2, thus leading to the same conclusion that the system is not E-Uniformly observable.

3.3.6 Special Case 2: Different but constant heading directions

In this case let us consider two agents are moving along two different straight lines with different heading directions that are constant over time, i.e, $\theta_1 = k_1$ and $\theta_2 = k_2$ where $k_1 \neq k_2$. This is shown in figure 3.4. Hence there is relative motion between the agents. From (3.44) and (3.45), we obtain

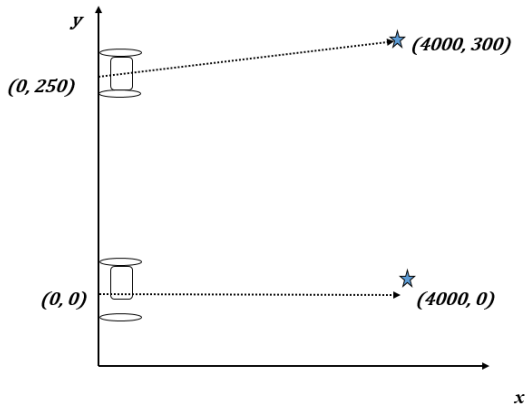


Figure 3.4: Straight line case with different heading angles of the agents

$$\begin{aligned} I_1 &= V(\cos k_2 - \cos k_1)(t_1 - t_0) \\ I_2 &= V(\sin k_2 - \sin k_1)(t_1 - t_0) \end{aligned} \tag{3.53}$$

Solving the above for I_2 as a function of I_1 leads to the following

$$I_2 = \frac{\sin k_2 - \sin k_1}{\cos k_2 - \cos k_1} I_1 = qI_1 \tag{3.54}$$

This leads to the following condition of unobservability.

$$A^2 + B^2 + 2I_1(A + Bq) = C^2 + D^2 + 2I_1(C + Dq) \tag{3.55}$$

As in the general case, we equate the time invariant and time varying parts. In this case, we obtain the following solution conditions

$$\begin{aligned} A^2 + B^2 &= C^2 + D^2 \\ A + Bq &= C + Dq \end{aligned} \tag{3.56}$$

Solving the linear equation for C, we get

$$C = A + Bq - Dq \tag{3.57}$$

Then, substituting back into the equation 3.56, we obtain

$$\begin{aligned} A^2 + B^2 &= (A + Bq - Dq)^2 + D^2 \\ A^2 + B^2 &= A^2 + B^2q^2 + D^2q^2 + 2ABq - 2ADq - 2BDq^2 + D^2 \\ 0 &= D^2q^2 + D^2 - 2ADq - 2BDq^2 + 2ABq + B^2q^2 - B^2 \\ 0 &= (1 + q^2)D^2 + (-2Aq - 2Bq^2)D + 2ABq + B^2q^2 - B^2 \\ D &= \frac{2Aq + 2Bq^2 \pm \sqrt{(-2Aq - 2Bq^2)^2 - 4(1 + q^2)(2ABq + B^2q^2 - B^2)}}{2(1 + q^2)} \\ D &= \frac{2Aq + 2Bq^2 \pm \sqrt{4A^2q^2 - 8ABq + 4B^2}}{2(1 + q^2)} \\ D &= \frac{Aq + Bq^2 \pm \sqrt{(Aq - B)^2}}{(1 + q^2)} \\ D &= \frac{Aq + Bq^2 \pm (Aq - B)}{(1 + q^2)} \\ D &= B \text{ or } \frac{2Aq + Bq^2 - B}{(1 + q^2)} \forall [x_{1o}, y_{1o}]^T, [x_{2o}, y_{2o}]^T, [\bar{x}_{1o}, \bar{y}_{1o}]^T, [\bar{x}_{2o}, \bar{y}_{2o}]^T \in \Re^2 \end{aligned} \tag{3.58}$$

Solving for C by substituting 3.58 into 3.57, we get $C = A$ or

$$\begin{aligned} C &= A + Bq - \frac{2Aq + Bq^2 - B}{(1 + q^2)}q \\ C &= \frac{(A + Bq)(1 + q^2)}{(1 + q^2)} - \frac{2Aq^2 + Bq^3 - Bq}{(1 + q^2)} \\ C &= \frac{A + Aq^2 + Bq + Bq^3 - (2Aq^2 + Bq^3 - Bq)}{(1 + q^2)} \\ C &= \frac{A + 2Bq - Aq^2}{(1 + q^2)} \forall [x_{1o}, y_{1o}]^T, [x_{2o}, y_{2o}]^T, [\bar{x}_{1o}, \bar{y}_{1o}]^T, [\bar{x}_{2o}, \bar{y}_{2o}]^T \in \Re^2 \end{aligned} \tag{3.59}$$

It should be pointed out here that it is not possible for the vehicles to travel to the goal location with any arbitrary choice of constant heading angle; only a single choice of heading angle (directly

to the goal) is possible if the heading angle is constant. Also, the space of unobservability is larger in this case when compared to the following cases, but smaller than the straight line case with zero heading angles.

3.3.7 Special Case 3: Sinusoidal motion with the agents in phase relative to each other

To obtain in phase motion, we substitute $\theta_1 = \theta_2$ without requiring them to be constant over time. For this case, from (3.44) and (3.45), we see that $I_1 = I_2 = 0$, which is equivalent to the straight line case 3.51. This explains why the simulation results for the straight line and in phase cases yield similar (bad) results. The figure 3.5 shows the displacements of agents in sinusoidal motion where the agents are in phase to each other.

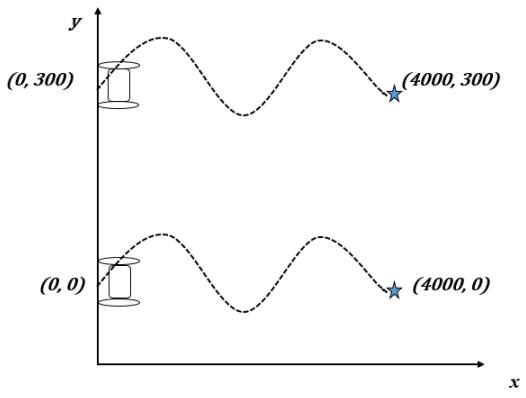


Figure 3.5: Sinusoidal case with agents in phase to each other

3.3.8 Special Case 4: Sinusoidal motion with the agents out of phase relative to each other

The displacements of the agents along the sinusoidal trajectories with out of phase to each other is shown in figure 3.6. To obtain out of phase motion, we substitute $\theta_1 = -\theta_2$. For this case, from (3.44), we see that $I_1 = 0$. Substituting this result into (3.46) yields the following

$$A^2 + B^2 + 2BI_2 = C^2 + D^2 + 2DI_2 \tag{3.60}$$

When we equate the time invariant and time varying parts, we obtain the following solution conditions

$$A^2 + B^2 = C^2 + D^2 \tag{3.61}$$

$$B = D \tag{3.62}$$

Substituting $B = D$ into the nonlinear equation in (3.60) leads to the following solution

$$A^2 = C^2$$

$$C = \pm A \tag{3.63}$$

This result indicates that any two initial states with the same separation distance between the two vehicles along the x-axis, together with the same relative displacement along the y-axis will be indistinguishable. For example, if vehicle 1 starts at (0,0) and vehicle 2 starts at (100,100), this is indistinguishable from vehicle 1 starting at (100,100) and vehicle 2 starting at (200,200) or (0,200). In some sense, this space of unobservability is larger than in the general case (which may explain why out of phase motion is better than in phase, but worse than other types of motion in general).

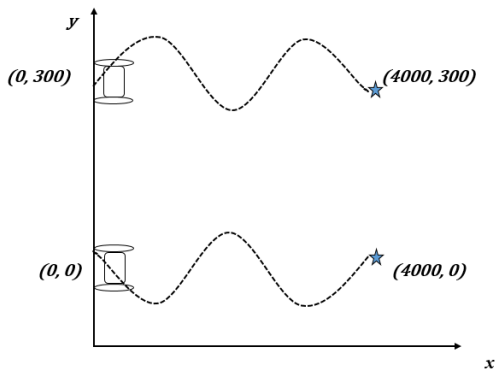


Figure 3.6: Sinusoidal case with agents out of phase to each other

The space of unobservability is given by $\Phi = \{(x_{2o} - x_{1o}) = \pm(\bar{x}_{2o} - \bar{x}_{1o}) \mid \forall x_{1o}, x_{2o}, \bar{x}_{1o}, \bar{x}_{2o} \in \Re\}$ $\cap\{(y_{2o} - y_{1o}) = (\bar{y}_{2o} - \bar{y}_{1o}) \mid \forall y_{1o}, y_{2o}, \bar{y}_{1o}, \bar{y}_{2o} \in \Re\}$

3.3.9 Special Case 5 : Sinusoidal motion with the agents moving with different frequencies

In the above special cases where the agents are moving in sinusoidal motion, the agents are considered to move with the same frequency. In this section we will consider the agents are moving with different frequencies ω_1 and ω_2 as shown in the figure 3.7. Let us assume $\theta_1(t)$ and $\theta_2(t)$ are the heading rates of the agents corresponding to their frequencies respectively which change with time, t. From (3.44) and (3.45), we obtain

$$\begin{aligned} I_1 &= V \int_{t_0}^{t_1} (\cos\theta_2(t) - \cos\theta_1(t))\ dt \\ I_2 &= V \int_{t_0}^{t_1} (\sin\theta_2(t) - \sin\theta_1(t))\ dt \end{aligned} \tag{3.64}$$

Solving the above for I_2 as a function of I_1 leads to the following

$$I_2 = \frac{\int_{t_0}^{t_1} (\sin\theta_2(t) - \sin\theta_1(t))\ dt}{\int_{t_0}^{t_1} (\cos\theta_2(t) - \cos\theta_1(t))\ dt} I_1 = I_3 I_1 \tag{3.65}$$

For the solution to the measurement equation as shown in (3.46) we obtain

$$A^2 + B^2 + 2I_1(A + BI_3) = C^2 + D^2 + 2I_1(C + DI_3)$$

$$A^2 + B^2 + 2I_1A + 2I_1BI_3 = C^2 + D^2 + 2I_1C + 2I_1DI_3$$

Again we equate the time varying and time invariant coefficients on either sides.

$$A^2 + B^2 = C^2 + D^2 \tag{3.66}$$

$$A = C \tag{3.67}$$

$$B = D \tag{3.68}$$

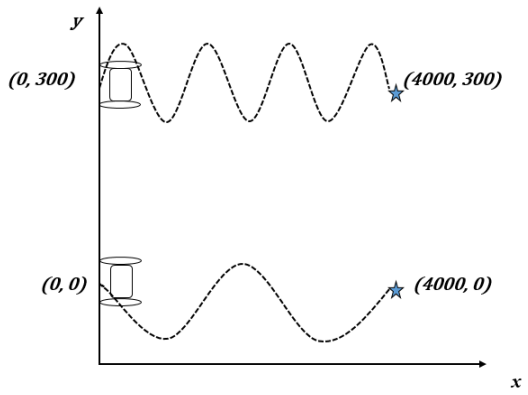


Figure 3.7: Sinusoidal case with agents moving with different frequencies

The result here indicates if the two vehicles with two initial states that have the same separation distance between them on both x and y axes, then they are indistinguishable. Again, this space of unobservability is smaller than in the other cases like the straight line, in phase, and out of phase ones which indicate that the different frequencies case should perform better than any other cases considered in the previous sections. The set of initial conditions for which the E-Uniform observability does not hold true is given by $\Phi = \{(x_{2o} - x_{1o}) = (\bar{x}_{2o} - \bar{x}_{1o}) \mid \forall x_{1o}, x_{2o}, \bar{x}_{1o}, \bar{x}_{2o} \in \Re\}$ $\cap\{(y_{2o} - y_{1o}) = (\bar{y}_{2o} - \bar{y}_{1o}) \mid \forall y_{1o}, y_{2o}, \bar{y}_{1o}, \bar{y}_{2o} \in \Re\}$

From the E-Uniform observability in all the cases considered, we observe that the space of unobservability is highest in the straight line case and the sinusoidal case with the agents moving in phase to each other. The space of unobservability for the sinusoidal case with the agents moving out of phase to each other is less than the in-phase case, while the sinusoidal case where the agents move with different frequencies has the least space of unobservability.

CHAPTER 4

Simulations and Results

4.1 Introduction

The goal of this section is to compare the results obtained by tracking various paths to verify the hypothesis that relative motion between agents can help bound the drift. The hypothesis was motivated through the E-uniform observability analysis, here we test it extensively over 1000 Monte Carlo simulations (for each trajectory type) run for eight different trajectory types. In the following results, we show the simulation results from the straight line paths, paths obtained by tracking various sinusoidal maneuvers (in phase, out of phase, and with different frequencies), Rapidly Exploring Random Trees (RRT) [50] and two other algorithms developed using the RRT algorithm (sections 4.2 through 4.10). In all of these trajectory types the simulations are run for 140 seconds with a tolerance of 0.10 seconds except for the straight line case since it takes only 133 seconds for the vehicles to reach the goal location in this case.

In this section the plots of the simulations are demonstrated for two vehicles (specifically Dubins) moving with a vehicle speed of 30 m/s. The agents start from an initial location of [0,0] and [0,300] respectively([x,y] on the 2-D coordinate axes) and the final goal locations are [4000,0] and [4000,300] respectively. The initial separation distance considered between the agents is 300 units on the y-axis. The vehicles are required to reach the final goal location with the least error possible in the absence of absolute position reference.

The simulation figures for each of the trajectory types include displacement plot for the agents with both the true values and the estimated values from the Extended Kalman Filter (more details on EKF are described in section 3.2.1). The sub-figures in the middle display the errors in the x and y axes as a function of time with 3σ Kalman filter covariance bounds shown in blue and the

actual error shown in the middle for each of the vehicles. In all of the cases the bottom sub-figures display the histograms of final position estimation error of both the vehicles based on 1000 Monte Carlo runs plotted for 100 bins and the root-mean-square error is shown on these histograms. The final position estimation error is the difference between the true final position values and estimated final position values from the EKF.

4.2 Straight line Paths

In this section the Dubins are simulated to move along the straight line paths with zero heading angles. Agent 1 starts at an initial location of [0,0] and agent 2 starts at [0,300], and the final goal location for the agents are [4000,0] and [4000,300] respectively on the 2D coordinate axes. In this case the agents reach the goal location with time around 133 seconds. Since the agents are expected to move along the straight line trajectories the heading angle for both the agents would be zero units which indicates there is no deliberate relative motion between the agents.

Figure 4.1: Displacements of the vehicles along the straight line trajectories

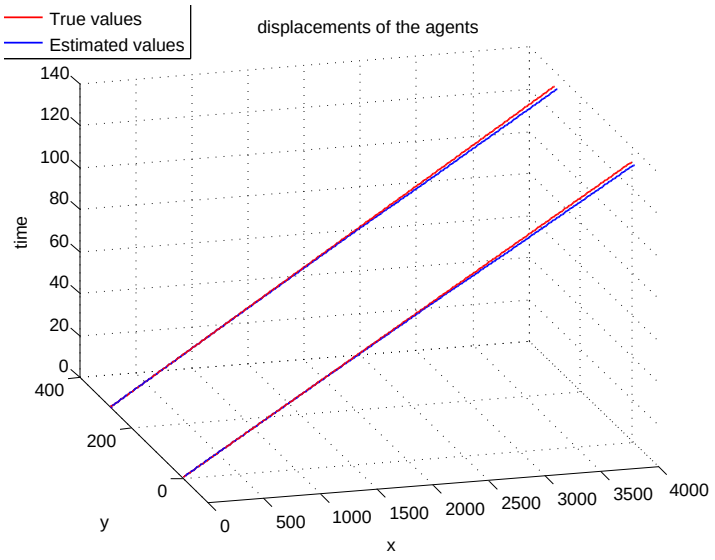
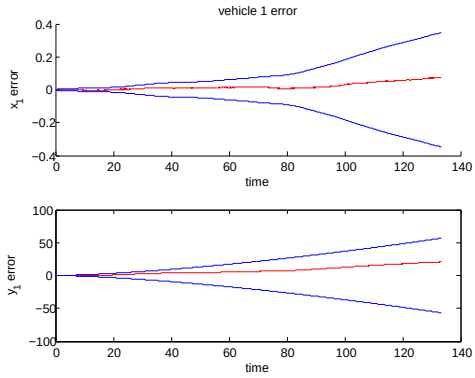


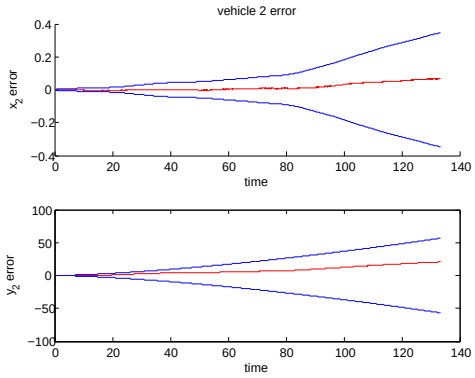
Figure 4.2: 3D plot showing the displacements of the agents with time

In the figure 4.1 the red lines indicate the true values of the agents and the blue dashed lines indicate the estimated values of the agents from the EKF with the x-position of the vehicles considered on the x-axis and y-position of the vehicles on the y-axis. In the figure 4.1 the bottom displacements are of the agent 1 and the top lines indicate the displacements for the agent 2. We can observe from the figure that the estimated values of the agents drift from the true values with the displacement along the goal location. This indicates that in the absence of any absolute position reference the estimated values of the agents drift from the true values of the agents.For the purpose of better understanding, the displacements of the agents are plotted with time on a 3D plot shown in 4.2.

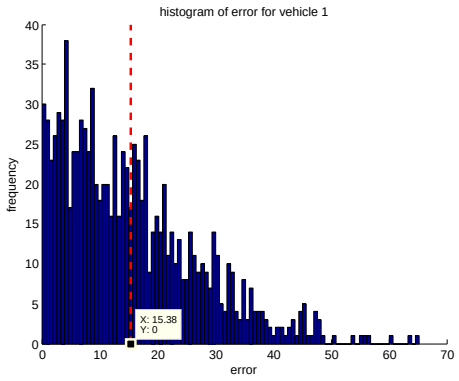
Figure 4.3a indicates the x and y position errors for the vehicle 1. The time is plotted on the x-axis and the error value on the y-axis. In these figures the middle red lines are the actual error between the estimated values from the true values and the blues lines on the top and bottom of the middle red lines are the covariance bounds from the EKF. In figure 4.3c, the histogram of the final position estimation error for 1000 Monte Carlo runs of the vehicle 1 is plotted for 100 bins with the final position estimation error on the x-axis and the frequency of the error on the y-axis. The red line here indicates the root mean square error value of the estimation error. Similarly the figures 4.3b and 4.3d show the x, y position errors and histogram of final position estimation error for the vehicle 2 respectively. We can observe from the figures 4.3a and 4.3b, the x-position error of both the vehicles shows an abrupt change in the covariance values from the EKF after the time of about



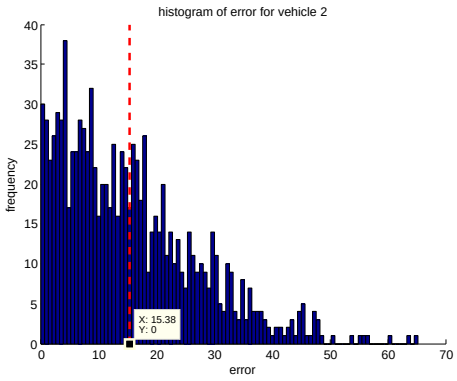
(a) Vehicle 1 x-position and y-position errors



(b) Vehicle 2 x-position and y-position errors



(c) Vehicle 1 histogram of final position estimation error over 1000 Monte Carlo runs



(d) Vehicle 2 histogram of final position estimation error over 1000 Monte Carlo runs

Figure 4.3: Error plots for vehicle 1 and vehicle 2 in the straight line case with zero heading angles for both the vehicles

80 seconds. The reason for this abrupt change in the x-position error is not clearly known. We also observe the growth in the errors along the x and y axes with time. This is due to the absence of any absolute position reference. The figures 4.3c and 4.3d show the histograms of the final position estimation error for vehicle 1 and vehicle 2 respectively. We can observe that the mean square error after 1000 Monte Carlo runs is 15.38 units in the straight line case for both the vehicles.

4.3 Sinusoidal paths in phase with each other

In this section, the Dubins models are simulated to move along the sinusoidal trajectories with both the agents in phase to each other. Again in this section, the start and goal locations for the agents are [0,0], [4000,0] and [0, 300], [4000,300] respectively on the 2D coordinate axes for vehicle 1 and vehicle 2. In this case the agents reach the goal location with time around 140 seconds. Since the agents are in phase to each other the heading angles of both the agents are exactly the same. In figure 4.4 the displacements of the agents with their true values and the estimated values from the EKF are shown.

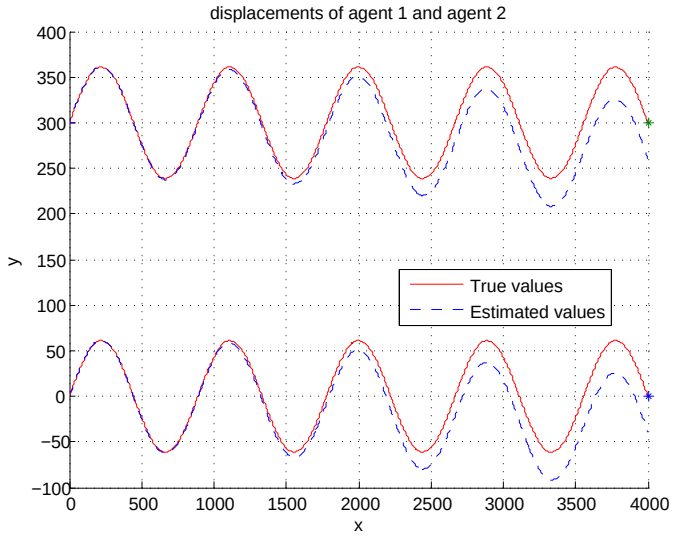


Figure 4.4: Displacements of the vehicles along the sinusoidal trajectories with the vehicles in phase to each other

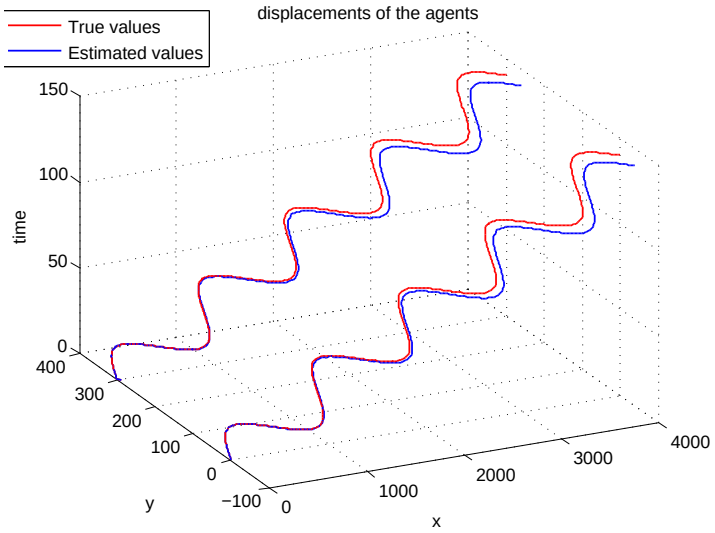


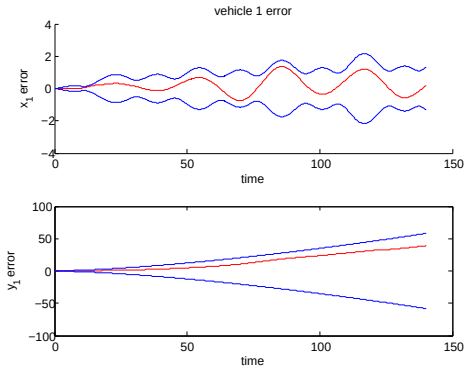
Figure 4.5: 3D plot showing the displacements of the vehicles in the sinusoidal trajectories with time in which the agents are in phase to each other

In figure 4.4, the red lines indicate the true values of the agents and the blue dashed lines indicate the estimated values of the agents from the EKF with the x-position of the vehicles considered on the x-axis and y-position of the vehicles on the y-axis. The bottom displacements are of the agent 1 and the top displacements are of the agent 2. We can again observe from figure 4.4, that the estimated values of the agents drift from the true values similar to the case of straight line paths. The displacements of the agents with time are also shown in a 3D plot in 4.5.

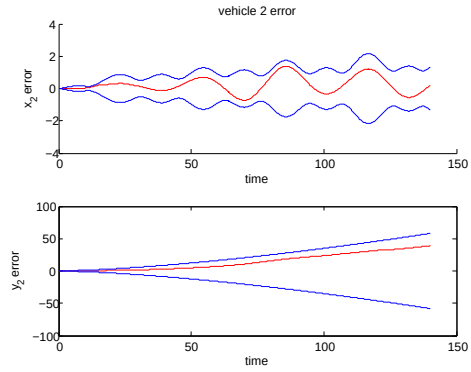
The figures 4.6a and 4.6b show the error plots of the x and y position for the vehicle 1 and vehicle 2 respectively. Again in these figures the middle red lines indicate the actual error of the estimated values from the true position values and the blues lines on the top and bottom of the middle lines indicate the covariance bounds from the EKF. These errors are plotted on the y-axis and the time is plotted on the x-axis of these plots. We can observe from these figures there are abrupt changes in the x-position error for the vehicles. This is due to the nonlinearities that arise in the EKF with the vehicles relative motion of approaching together and moving away from each other. Again, the exact reason for this behavior is not known yet.

The figures 4.6c and 4.6d show the histograms of the final position estimation error for 1000 Monte Carlo runs for vehicle 1 and vehicle 2 respectively. The mean square error in these figures is shown with the dotted red lines. The error is plotted on the x-axis of these figures and the

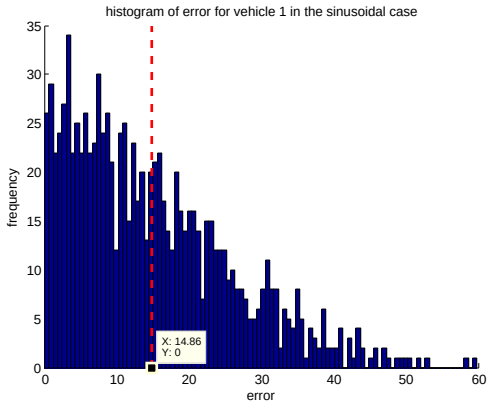
frequency of the error on the y-axis. The mean square error from the sinusoidal in phase case is found to be 14.86 units which is almost the same as the straight line case which actually supports the observability analysis results shown in the previous section 3.3.7.



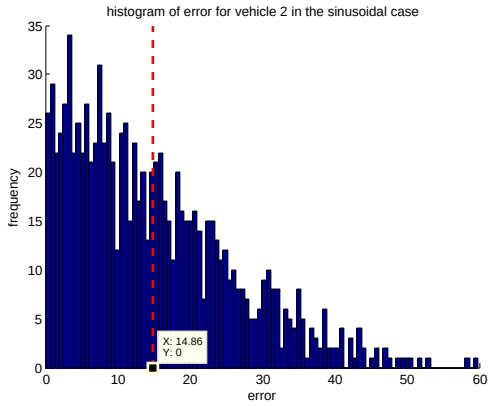
(a) Vehicle 1 x-position and y-position errors



(b) Vehicle 2 x-position and y-position errors



(c) Vehicle 1 histogram of the final position error estimation over 1000 Monte Carlo runs



(d) Vehicle 2 histogram of the final position error estimation over 1000 Monte Carlo runs

Figure 4.6: Error plots for vehicle 1 and vehicle 2 in the sinusoidal case with both the vehicles in phase to each other

4.4 Sinusoidal paths out of phase with each other

In this section the vehicles are simulated along the sinusoidal trajectories like in the above section. The start and goal locations of the vehicles are both the same as in the previous section. Again, the agents take around 140 seconds to reach the goal location. The agents are simulated to travel out of phase to each other which shows that the heading angles of both the agents are exactly opposite to each other. It makes the agents move closer to each other during the highest amplitude of the sinusoidal curve. This helps in understanding better how the relative motion between the agents affects the navigation state estimation accuracy in the absence of absolute position reference.

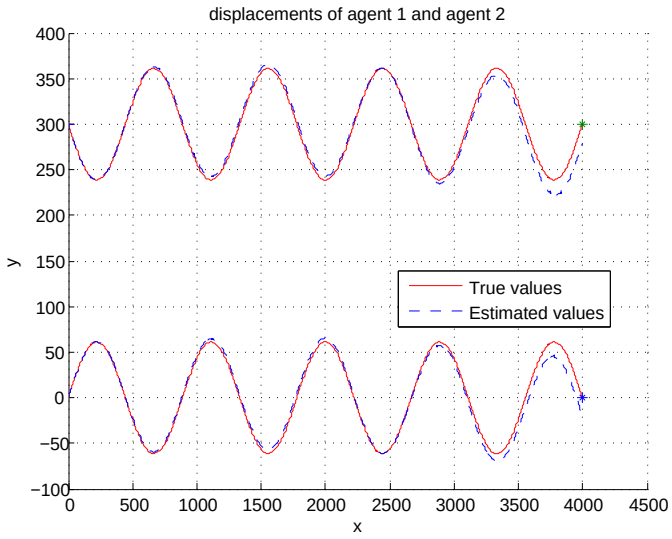


Figure 4.7: Displacements of the vehicles along the sinusoidal trajectories with the vehicles out of phase to each other

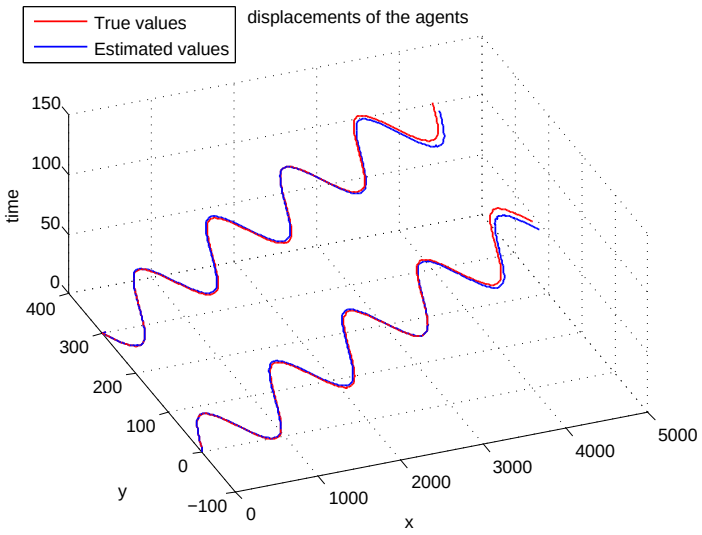
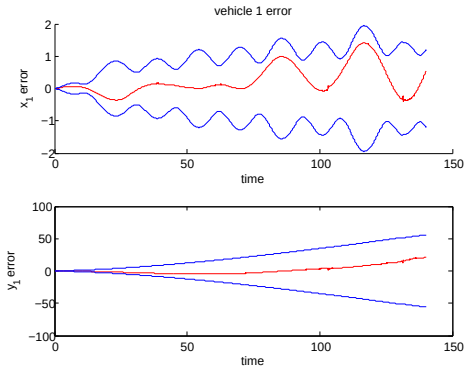
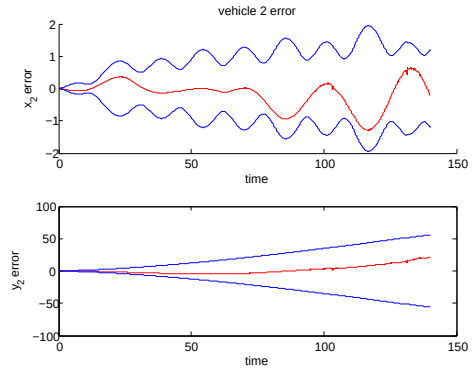


Figure 4.8: 3D plot showing the displacements of the vehicles in the sinusoidal trajectories with time in which the agents are out of phase to each other

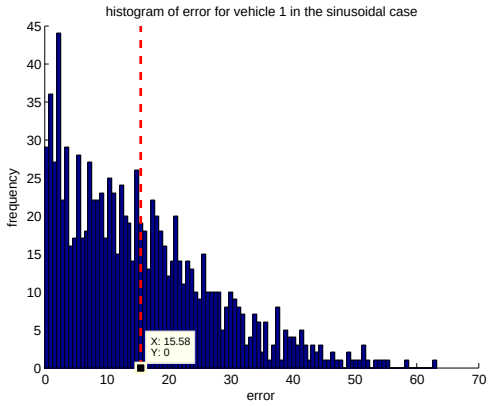
In figure 4.7 we observe the displacements of the agents along the sinusoidal trajectories with the agents moving out of phase to each other. The displacements are also shown in 3D for time in figure 4.8. The bottom displacements in the figure show the trajectory of the agent 1 and the top displacements show the trajectories of the agent 2. Again, the red lines indicate the true position values and the blue dotted lines show the estimated values. We can observe from the figure the drift of the estimated values from the true values similar to the straight line and the sinusoidal in phase cases.



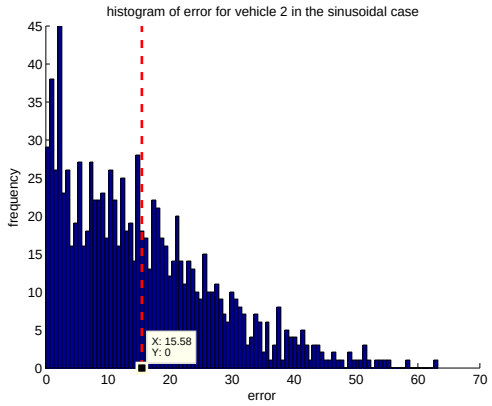
(a) Vehicle 1 x-position and y-position errors



(b) Vehicle 2 x-position and y-position errors



(c) Vehicle 1 histogram of the final position error estimation over 1000 Monte Carlo runs



(d) Vehicle 2 histogram of the final position error estimation over 1000 Monte Carlo runs

Figure 4.9: Error plots for vehicle 1 and vehicle 2 in the sinusoidal case with the agents moving out of phase to each other

Figure 4.9 shows the x and y position errors and the histograms of the final position estimation error for both the vehicles. In figure 4.9a, the x-position and y-position errors are plotted for vehicle 1 on the y-axis and time on the x-axis. The red line in the middle is the actual error between the estimated values, true values in the positions and the blue lines above and below this show the covariance bounds from the EKF. A histogram is drawn in figure 4.9c with the mean square error shown in the red dotted line and the final position estimation error plotted on the x-axis with the frequency of error on the y-axis. Similarly the figures 4.9b and 4.9c are the x and y position errors, histogram of the final position estimation error for the vehicle 2.

We can observe from the figure 4.9 the x-position error for both the vehicles has several nonlinearities due to the vehicle motions. The mean square error for the vehicles in this case is around 15.58 units which is almost as equal to the straight line and the sinusoidal in phase cases. From the section 3.3.8 we noticed that the space of unobservability is lower in the sinusoidal out of phase than sinusoidal in phase and straight line cases. This is not actually seen from the simulation results here. However, from the table 4.2 we observe that the sinusoidal out of phase case does perform better (lower mean square error) than the sinusoidal in phase case. This indicates that the performance of the agents depends on also the frequency and amplitudes which cause much relative motion between them.

4.5 Vehicles moving in straight lines with different heading angles and constant frequency

In this section of the simulations, the vehicles are simulated along the straight lines again with the difference in the agents heading angles which may lead to the relative motion between the agents unlike in the previous straight line case of zero heading angle for the agents. Again, the frequency of the agents is constant. Here, agent 1 start location is [0,0] and the goal location is [4000,0] on the 2D coordinate axes. Agent 2 has a start location of [0,275] and goal location of [4000,300]. The difference in the start location of agent 2 leads to the heading angle difference with agent 1 upon reaching the goal locations.

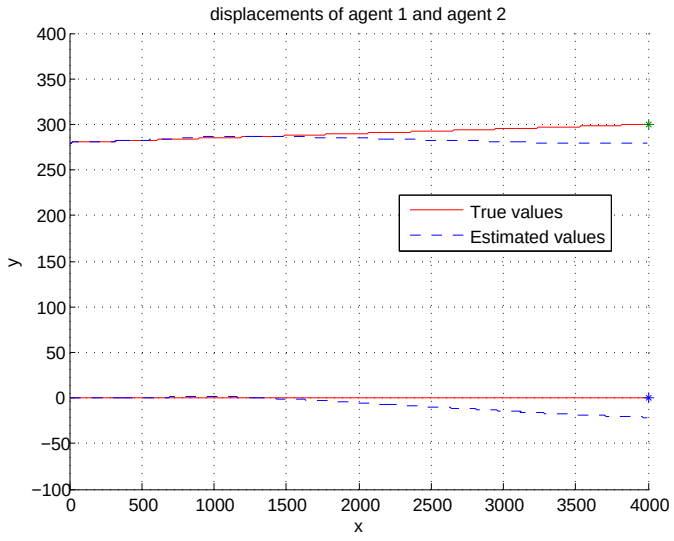


Figure 4.10: Displacements of the vehicles along the straight lines with different heading angles for both the vehicles

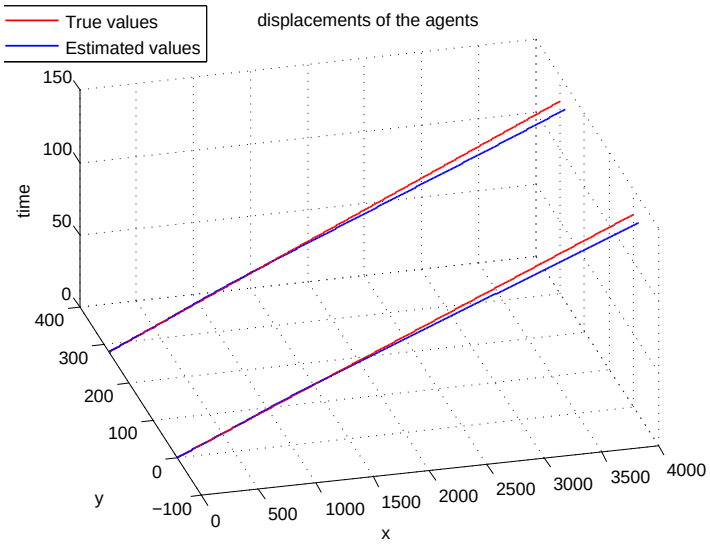
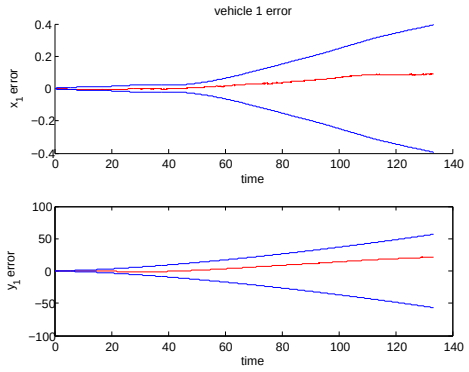
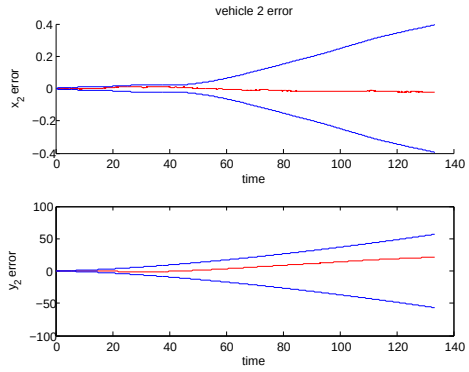


Figure 4.11: 3D plot showing the displacements of the vehicles for time along the straight lines with different heading angles for both the vehicles

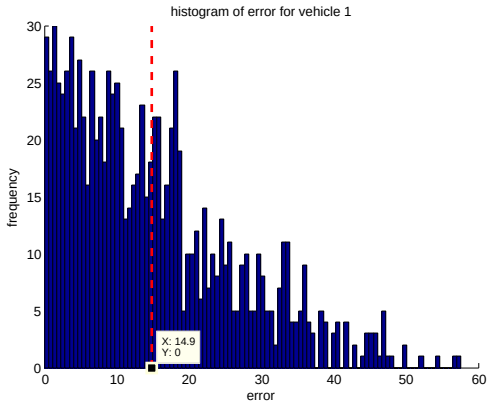
The displacements plot shown in figure 4.10 shows the displacements of agent 1 at the bottom and the displacements of agent 2 on the top. The true values and the estimated values are shown similar to the previous cases. We can observe from the figure 4.10, the heading angles of both the vehicles are different from each other. In this simulation the agents reach the goal location in about 133 seconds. The drift of the estimated values from the true values is also clearly observed from figure 4.10. Figure 4.11 also shows the displacements of the plots in 3D with time.



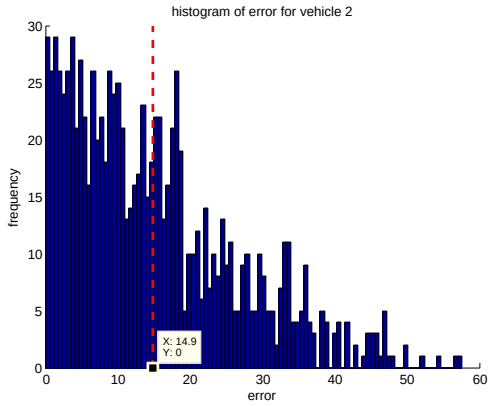
(a) Vehicle 1 x-position and y-position errors



(b) Vehicle 2 x-position and y-position errors



(c) Vehicle 1 histogram of the final position error estimation over 1000 Monte Carlo runs



(d) Vehicle 2 histogram of the final position error estimation over 1000 Monte Carlo runs

Figure 4.12: Error plots for vehicle 1 and vehicle 2 in the straight line case with different heading angles for both the vehicles

From the figures 4.12a and 4.12b, we can observe the x and y position errors for agent 1 and agent 2 respectively. The middle line indicates the error in the estimated values from the true values and the top and bottom lines show the covariance bounds from the filter. We can notice the abrupt increase in the x-position error at time around 60 seconds. Again, the reason is not known yet. The histograms of the final position estimation errors for vehicle 1 and vehicle 2 are shown in figures 4.12c and 4.12d. The mean square error shown in the red line in these figures is 14.9 units. The change in the heading angles in the straight line case did not show much improvement for the navigation state estimation.

4.6 Vehicles moving in sinusoidal paths with different frequencies

In this section, the vehicles are simulated along the sinusoidal paths. Unlike in the previous sections where both the agents travel with same frequencies, in this case the agents have different frequencies with each other. The start location for agent 1 is [0,0] and the goal location is [4000,0]. Similarly, the initial and final positions for the agent 2 are [0,300] and [4000,300]. The time taken for the agents to reach the goal location is 140 seconds. The displacements plot for the agents in this case is shown in figure 4.13. Since the agents have different frequencies the time taken for the agents to reach the goal location will be different. Hence, for both the agents to reach the goal location at the same time the amplitudes of the agents are also different from each other in the simulations. The agent with more frequency will have lower amplitude than the agent with higher frequency. That way, both the agents reach the goal location at the same time.

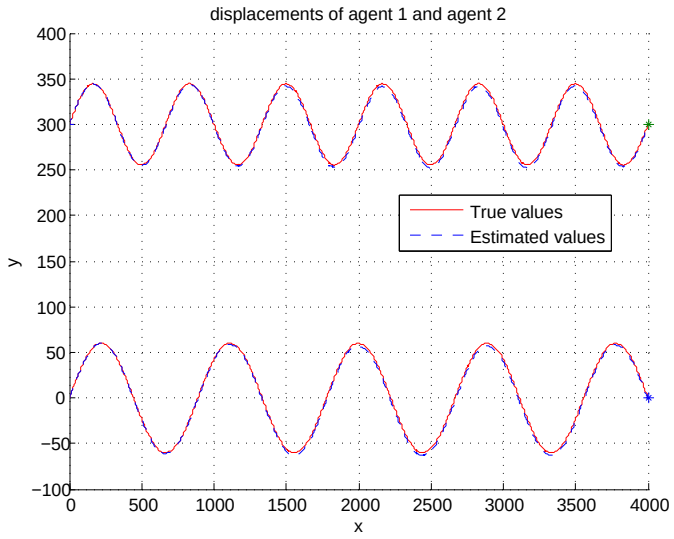


Figure 4.13: Displacements of the vehicles along the sinusoidal trajectories with the vehicles moving with different frequencies from each other

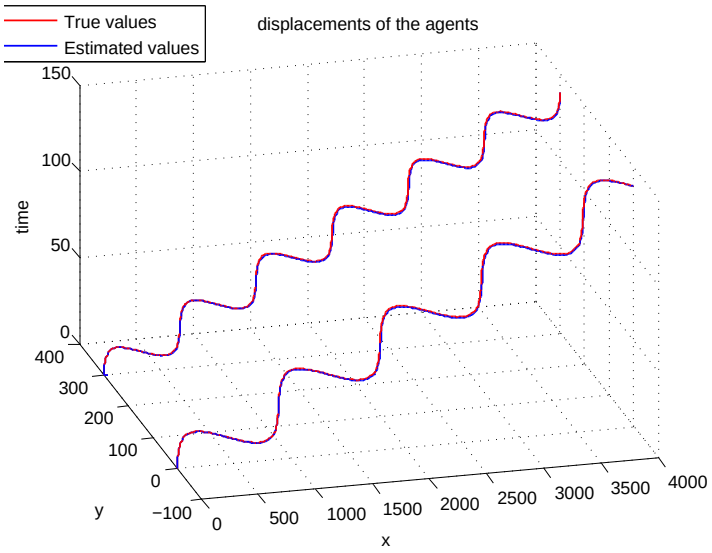
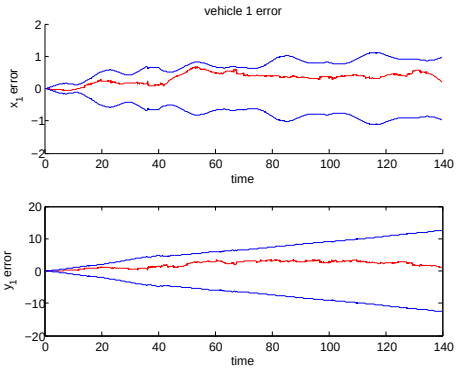
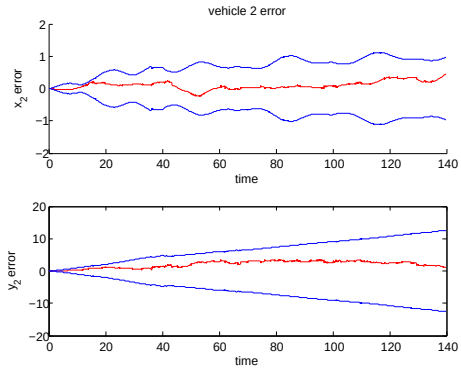


Figure 4.14: 3D plot showing the displacements of the vehicles for time along the sinusoidal trajectories with the vehicles moving with different frequencies from each other

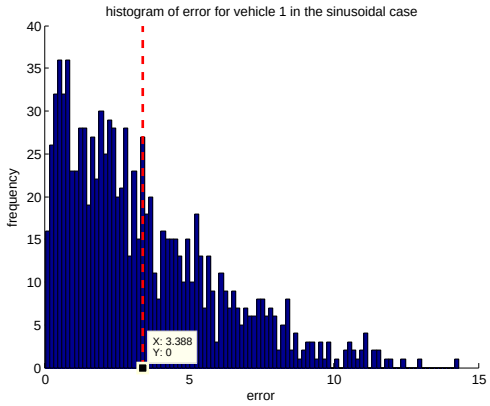
Figure 4.13 shows the displacement plots of both the agents with agent 1's displacement on the bottom and the agent 2's displacement on the top inside the figure. Figure 4.14 also shows the displacements of the agents in 3D with time. The solid red lines show the true position values and the dotted blue lines indicate the estimated values. We can observe from the figure that the frequency of the sinusoidal curve for agent 1 is more than the frequency of agent 2, hence the amplitude of agent 1 is also more than agent 2. We notice from the figure that the drift in the estimated values from the true values of the agents is actually lower than in the straight line cases and the sinusoidal case where the agents are moving with the same frequency. We also observe this from the error plots shown in the following figures.



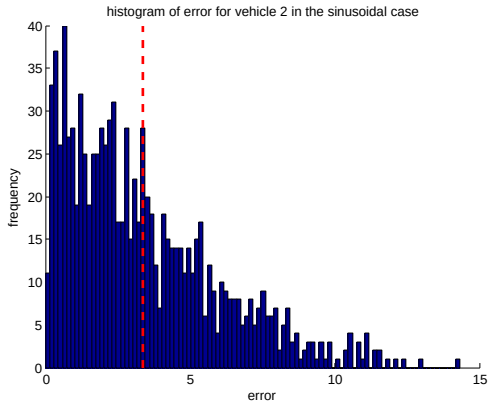
(a) Vehicle 1 x-position and y-position errors



(b) Vehicle 2 x-position and y-position errors



(c) Vehicle 1 histogram of the final position error estimation over 1000 Monte Carlo runs



(d) Vehicle 2 histogram of the final position error estimation over 1000 Monte Carlo runs

Figure 4.15: Error plots for vehicle 1 and vehicle 2 in the sinusoidal case with the vehicles moving with different frequencies from each other

Figures 4.15a and 4.15b show the x-position and the y-position errors for the agent 1 and agent 2. We can observe from the figures that the nonlinearities in the covariance values from the EKF are quite a bit lower than the straight line case and the sinusoidal case where the agents move with the same frequency. The reason for this is not known, however, we observe that the relative motion between the agents while moving with different frequency helps reduce the navigation estimation error better than any other case considered in the previous sections. This can clearly be observed from figures 4.15c and 4.15d where the histograms of the final position estimation error for the vehicles 1 and 2 are shown. We observe from these figures, the variance of the error is significantly lower compared to the previous cases. Also, the mean square error shown as the red dotted line is 3.38 units which is very much lower than the other cases around 15 units. It can be seen that the navigation estimation error is reduced if the agents move along the sinusoidal paths with different frequencies.

The simulation results are supported from the analysis done in the section 3.3 which shows the space of unobservability is least in the sinusoidal trajectory if the agents move with different frequencies. Here we observe that the mean square error is 3.38 units, less than any other trajectory considered.

4.7 Comparison of mean-square-error for all the cases

The comparison plot for the mean-square-error in the final position estimation is shown on a box-plot in figure 4.16. We can observe from the plot, the mean-square-error is highest in the straight line case. This error is lower in the sinusoidal trajectory type. Among the sinusoidal trajectories, the mean-square-error is significantly less when both the vehicles move in a sinusoidal trajectory with different frequencies (section 4.6) than the vehicles with in-phase (section 4.3) and out of phase (section 4.4) with each other in which the frequency of both the vehicles is the same. Inside the box plot shown in figure 4.16, the edges of the box are the 25th and 75th percentiles of the error data and the central mark is the median of the error for 1000 Monte Carlo runs. The diamond inside the box shows the mean-square of the error. This actually validates the hypothesis that the relative motion between the agents help in reducing the estimation error resulting due to the absence of an absolute position reference.

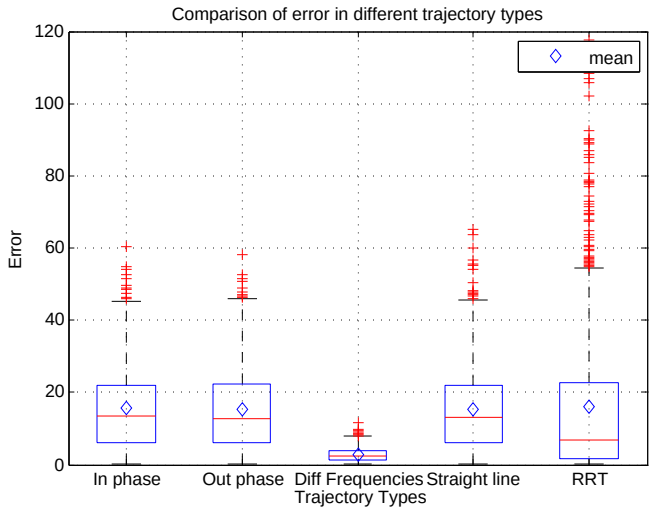


Figure 4.16: Comparison plot of final position estimation error in different trajectory types

4.8 Rapidly Exploring Random Trees (RRT)

Simulations are also done on some sampling based path planning techniques. RRT is one of such sampling based path planning algorithms designed for the problems with nonholonomic constraints. In the following simulations different RRT based algorithms are tried out to observe if these algorithms help improve the navigation state estimation of the agents. In this section, the algorithm used for generation of the paths using the RRT [50] technique is explained. In this algorithm the state space of x contains four states which are x,y values of the first agent and x,y values of the second agent. The RRT algorithm is generated in the following way:

1. A graph is initialized with empty vertices and edges. The start point is added to the graph as the initial vertex.

2. A random state is written which generates a random vertex in the graph.

3. The random vertex from the above function is passed into the Nearest function which searches for the nearest vertex in the graph to the random vertex and returns it.

4. A steer function is written for the nearest vertex which gives a new vertex that is in the direction of the nearest vertex and closer to it.

5. The new vertex is now added to the vertex list of the graph.

6. An edge is created between the new vertex and the nearest vertex and added to the edge list of the graph.

7. The steps from 2 through 7 are repeated until we reach the goal vertex within some epsilon radius. It is more explained in the following pseudo code.

Algorithm 1 RRT algorithm

1: $V \leftarrow x_{init}$; $E \leftarrow \emptyset$;
2: **while** goal is not found **do**
3: $x_{rand} \leftarrow RandomState$;
4: $x_{nearest} \leftarrow Nearest(G_1 = (V_1, E_1), x_{rand})$;
5: $x_{new} \leftarrow Steer(x_{nearest}, x_{rand})$;
6: $V \leftarrow V \cup x_{new}; E \leftarrow E \cup (x_{nearest}, x_{new})$;
7: **end while**
8: **return**$G = (V, E)$;

In figure 4.17, the displacements of the vehicles along the paths generated using the RRT technique are shown and figure 4.18 shows the displacements of the agents with time in 3D. The RRT algorithm is run to generate a path from an initial location of [0,0] for agent 1 and [300,0] for agent 2. The goal locations are [4000,0] for agent 1 and [4000,300] for agent 2. After the paths are generated using this technique the vehicles are simulated along the paths to reach the goal locations. In the figure the blue lines show the actual path generated using the RRT technique, the red lines indicate the true position values of the vehicles and the black lines show the estimated values from the filter. Since this is a sampling based path planning algorithm, the paths generated are random. This is used to see if these paths help in any navigation state estimation in the absence of an absolute position reference.

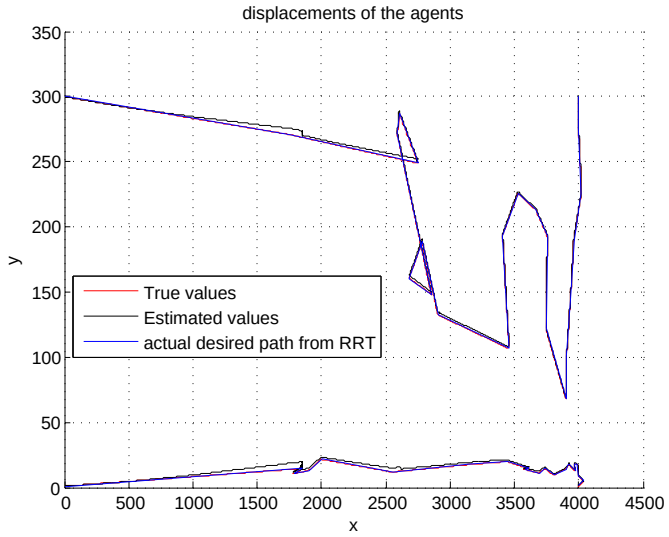


Figure 4.17: Displacements of the vehicles along the paths generated using the RRT technique

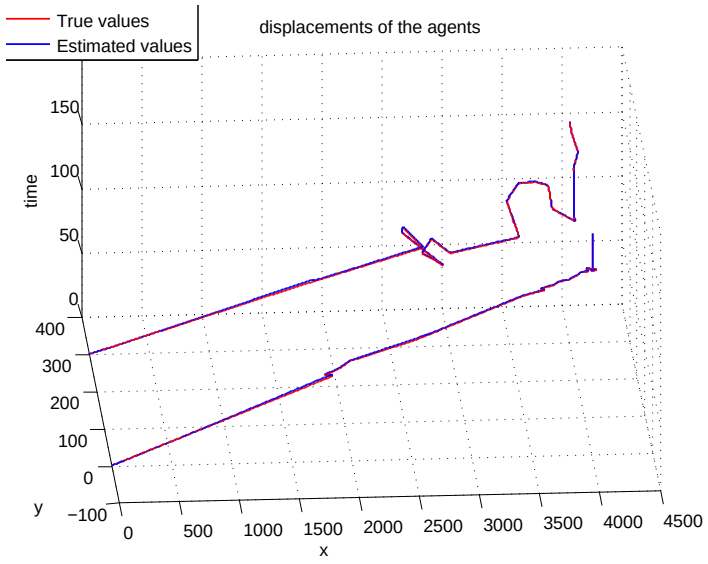
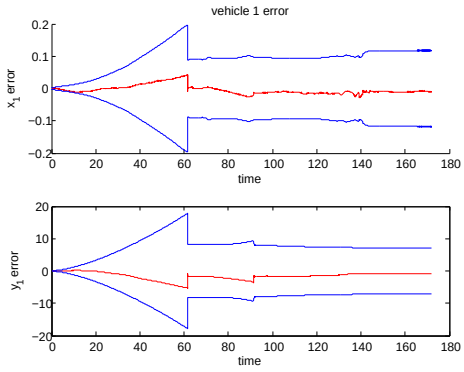
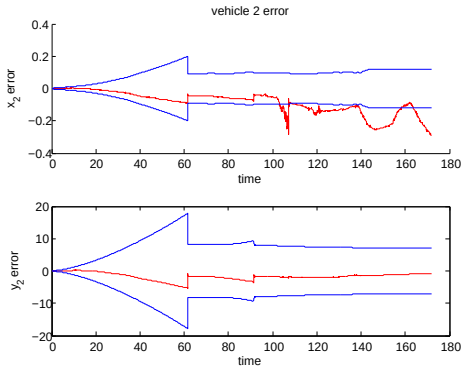


Figure 4.18: 3D plot showing the displacements of the vehicles for time along the trajectories generated using RRT

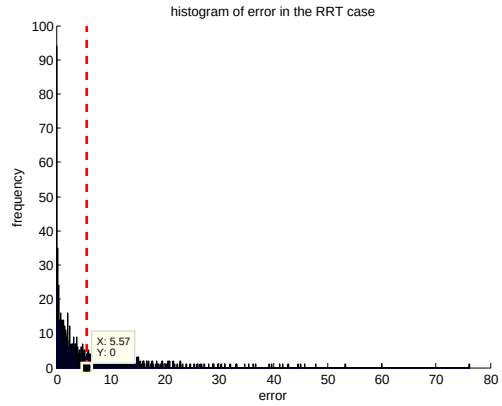
In figures 4.19a and 4.19b, the x and y position errors of agent 1 and agent 2 are shown with the errors in the position plotted on the y-axis and the time on the x-axis. The time taken for the agents to reach the goal location in this case is around 170 seconds. We can also observe from these figures there is a sudden decrease in the x and y position errors at around 60 seconds. Since the paths generated are random, the reason for this change in the errors is not yet analyzed. The histograms of the final position estimation error shown in figures 4.19c and 4.19d are for vehicle 1 and vehicle 2 respectively. These histograms show the data of 1000 Monte Carlo runs from the paths generated using the RRT technique. We can observe from these figures, the mean square of the estimation error in the final position is 5.57 units which is lower than the previous trajectory types except the sinusoidal case where the agents are moving with different frequencies. Using the RRT technique the navigation state estimation is showing improvement though this is not guaranteed as the paths generated are random.



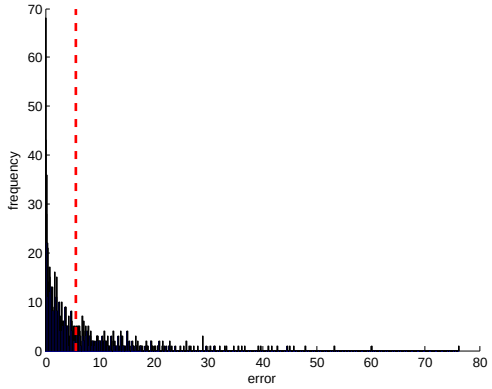
(a) Vehicle 1 x-position and y-position errors



(b) Vehicle 2 x-position and y-position errors



(c) Vehicle 1 histogram of the final position error estimation over 1000 Monte Carlo runs



(d) Vehicle 2 histogram of the final position error estimation over 1000 Monte Carlo runs

Figure 4.19: Error plots for vehicle 1 and vehicle 2 following the paths generated using the RRT technique

4.9 Iterative sampling with RRT

In this section a new approach developed from the RRT technique is explained and the simulation results of the vehicles traveling along the paths using this technique are also shown. Iterative sampling is performed for the selection of random points in the RRT technique. In this idea, the first vertex of the first agent is sampled and around this vertex in an epsilon distance some points (say 10) are sampled for the agent two. Among these points for the agent two, the point with the least error in the estimation from the Kalman filter is selected. The estimation is done by simulating the models between the two points; here first point is the first vertex of agent one and the second point is one among the ten randomly sampled points for the agent two. The estimation is done for all the ten points and the point with the least error is picked up as the vertex for the second agent. Around this vertex of the second agent ten random points are sampled in an epsilon distance for the agent one and again the point with least error as described above will be selected as the next (or say second vertex) vertex for the agent one. This procedure of sampling for both the agents is tried out till they reach the goal location. The pseudo code of this idea is shown below.

Algorithm 2 Iterative sampling with RRT

1: $V_1 \leftarrow x_{1_{init}}$; $E_1 \leftarrow \emptyset$;

2: $V_2 \leftarrow x_{2_{init}}$; $E_2 \leftarrow \emptyset$;

3: **while** goal is not found **do**

4: $x_{1_{rand}} \leftarrow RandomState$;

5: $x_{1_{nearest}} \leftarrow Nearest(G_1 = (V_1, E_1), x_{1_{rand}})$;

6: $x_{1_{new}} \leftarrow Steer(x_{1_{nearest}}, x_{1_{rand}})$;

7: $V_1 \leftarrow V_1 \cup x_{1_{new}}; E_1 \leftarrow E_1 \cup (x_{1_{nearest}}, x_{1_{new}})$;

8: **for** i = 1,...,n **do**

9: $x_{2_{rand(i)}} \leftarrow RandomState(x_{1_{new}})$;

10: $LeastError(x_{2_{rand}}) \leftarrow Estimates(x_{2_{rand(i)}})$;

11: $x_{2_{nearest}} \leftarrow Nearest(G_2 = (V_2, E_2), x_{2_{rand}})$;

12: $x_{2_{new}} \leftarrow Steer(x_{2_{nearest}}, x_{2_{rand}})$;

13: $V_2 \leftarrow V_2 \cup x_{2_{new}}; E_2 \leftarrow E_2 \cup (x_{2_{nearest}}, x_{2_{new}})$;

14: $x_{1_{rand(i)}} \leftarrow RandomState(x_{2_{new}})$;

15: $LeastError(x_{1_{rand}}) \leftarrow Estimates(x_{1_{rand(i)}})$;

16: $x_{1_{nearest}} \leftarrow Nearest(G_1 = (V_1, E_1), x_{1_{rand}})$;

17: $x_{1_{new}} \leftarrow Steer(x_{1_{nearest}}, x_{1_{rand}})$;

18: $V_1 \leftarrow V_1 \cup x_{1_{new}}; E_1 \leftarrow E_1 \cup (x_{1_{nearest}}, x_{1_{new}})$;

19: **end for**

20: **end while**

21: **return**$G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$;

The displacements of the vehicles along the paths generated using the Iterative sampling idea in the RRT technique are shown in figure 4.20 in 2D and figure 4.21 in 3D with time. The initial location is [50,0] for agent 1 and [300,0] for agent 2. The goal locations are [4000,0] for agent 1 and [4000,300] for agent 2. After the paths are generated using this technique the vehicles are simulated along the paths to reach the goal locations. In the figure the blue lines show the actual path generated using the Iterative sampling idea, the red lines indicate the true position values of the vehicles and the black lines show the estimated values from the filter.

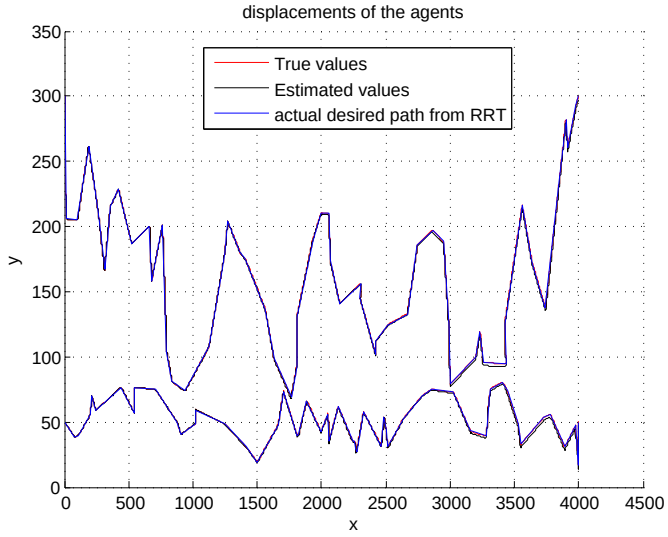


Figure 4.20: Displacements of the vehicles along the paths generated using the Iterative sampling with RRT

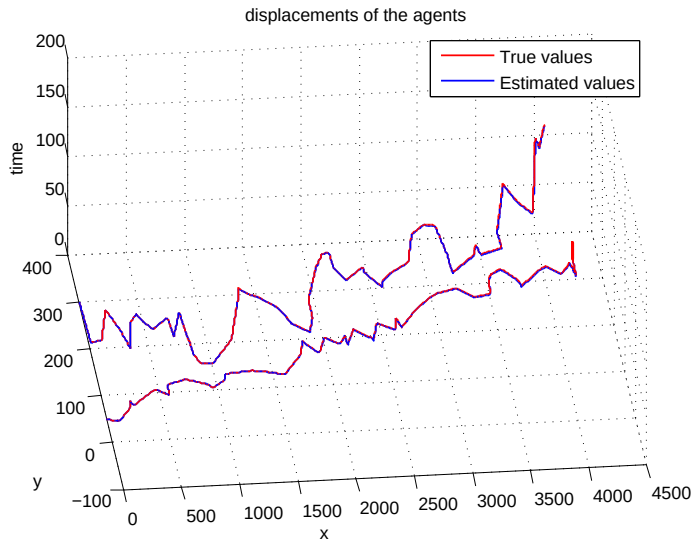
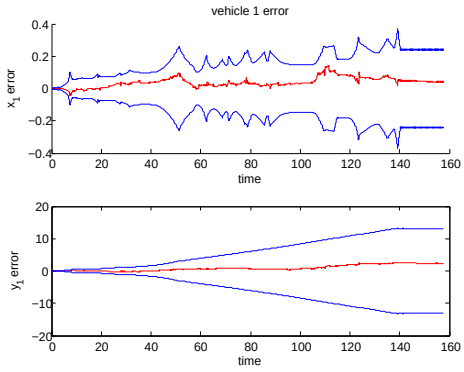
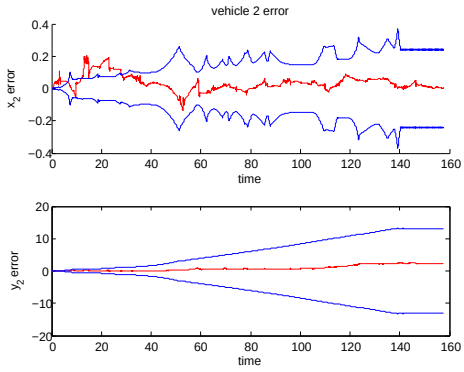


Figure 4.21: 3D plot showing the displacements of the vehicles for time along the trajectories generated using iterative sampling with RRT

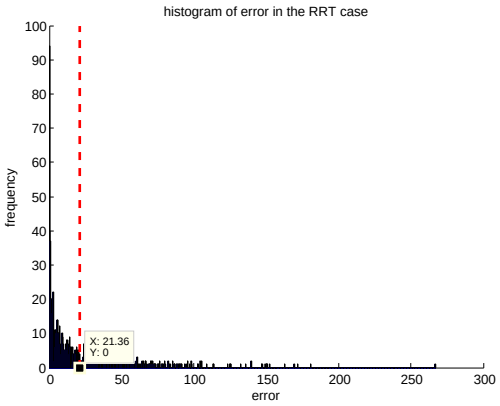
The figures in 4.22a and 4.22b show the x and y position errors of agent 1 and agent 2 respectively. The time taken for the agents to reach the goal location is around 159 seconds. We can observe the irregularities in the x-position errors for both agents. This may be due to the increase of nonlinearities in the filter as the agents move closer and away from each other. Also, in the y-position error after time period of around 140 seconds the error becomes almost constant. The reason for this is not known yet. The histograms of the final position estimation error are shown in the figures 4.22c and 4.22d. We can observe that the variance of the final position estimation error is very high in this scenario. Also, the mean square of the final position estimation error shown in dotted red line is 21.36 units for the vehicle 1 and 21.51 units for the vehicle 2 which is higher than any other trajectory type that are considered in here.



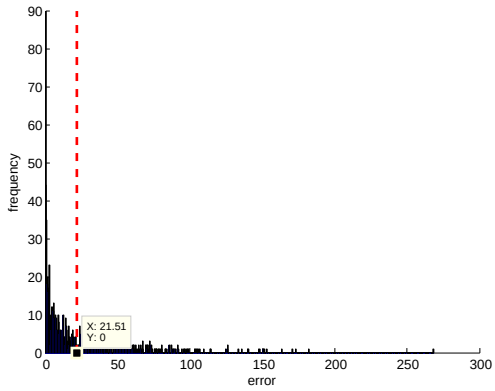
(a) Vehicle 1 x-position and y-position errors



(b) Vehicle 2 x-position and y-position errors



(c) Vehicle 1 histogram of the final position error estimation over 1000 Monte Carlo runs



(d) Vehicle 2 histogram of the final position error estimation over 1000 Monte Carlo runs

Figure 4.22: Error plots for vehicle 1 and vehicle 2 following the paths generated using the iterative sampling idea with the RRT technique

4.10 RRT algorithm with cost function included in the selection of random points

In this section a new idea of sampling the points in the RRT algorithm is shown. In the general RRT algorithm random points are sampled and are selected for the propagation of the graph to the goal location. In this idea, for each iteration in the selection of random points estimation process is done using the EKF. In each iteration, ten random points are sampled and the point with the least error between the true position values and estimated values from the filter is selected rather than selecting a random point (in the RRT). The estimation is done by simulating the models between the latest vertex and each of the ten random points.

In this algorithm an empty graph is initialized with zero vertices and zero edges. A random state function is written which samples ten random points within the goal space. The estimation process is done and the point with least error is selected among these ten points. The vertex in the graph which is nearest to the selected point from the estimation process is returned from the nearest function. The steer function returns a new vertex which is in the direction of the selected point and a little closer to the nearest vertex. This new vertex is added to the graph and an edge is created to the latest vertex and the new vertex generated. This process is continued until the goal location is found. The pseudo code for this idea is shown in the following lines.

Algorithm 3 RRT with cost function for selection of points

1: $V \leftarrow x_{init}$; $E \leftarrow \emptyset$;
2: **while** goal is not found **do**
3: $x_{rand(i)} \leftarrow RandomState(x_{new})$;
4: $LeastError(x_{rand}) \leftarrow Estimates(x_{rand(i)})$;
5: $x_{nearest} \leftarrow Nearest(G1 = (V1, E1), x_{rand})$;
6: $x_{new} \leftarrow Steer(x_{nearest}, x_{rand})$;
7: $V \leftarrow V \cup x_{new}; E \leftarrow E \cup (x_{nearest}, x_{new})$;
8: **end while**
9: **return**$G = (V, E)$;

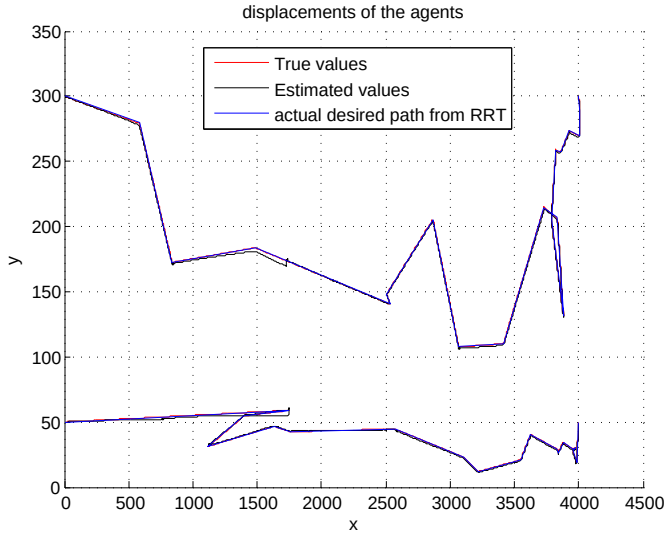


Figure 4.23: Displacements of the vehicles along the paths generated using the RRT idea with cost function included for the selection of random points

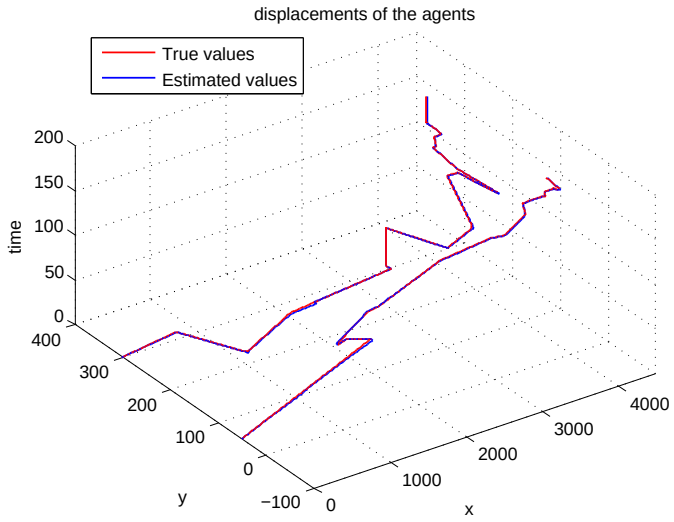
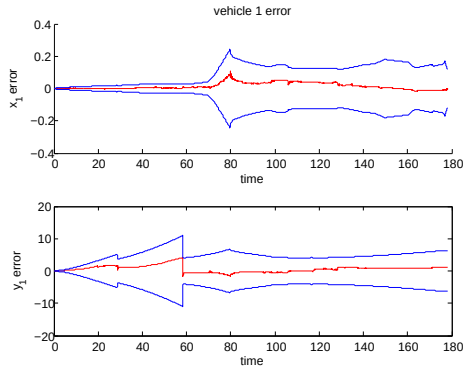


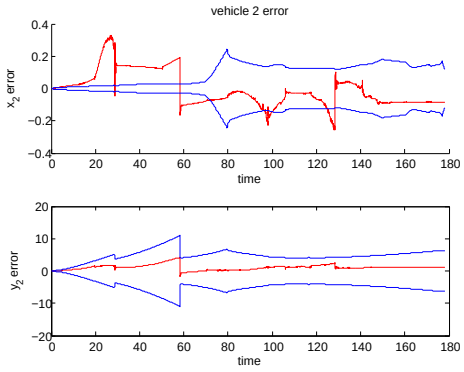
Figure 4.24: 3D plot showing the displacements of the vehicles for time along the trajectories generated using RRT with cost function included for the selection of random points

Figure 4.23 shows the displacements of the agents along the paths generated using the RRT included with cost technique in 2D and figure 4.24 in 3D with time. The blue lines show the actual paths generated using the algorithm, red lines show the true values of the displacements of the agents and the black lines indicate the estimated values from the filter. The start and final locations for the agents are [0,0] , [0,300] and [4000,0] , [4000,300] for agent 1 and agent 2 respectively. Again, the paths generated using this technique are random since this is based on a sampling based algorithm.

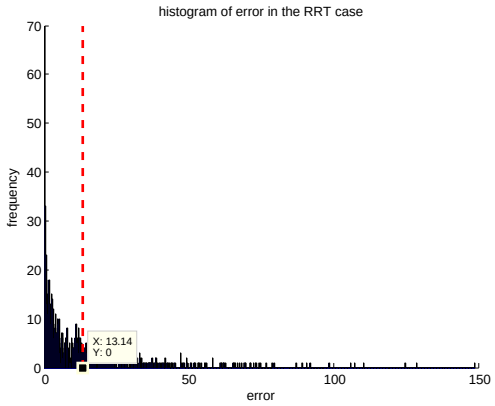
The figures 4.25a and 4.25b show the x and y position errors for vehicle 1 and vehicle 2 respectively. The agents reach the goal location in about 149 seconds in these simulations. There are several abrupt changes in the y-position error shown in the figures 4.25a and 4.25b. Again, since the path generated is random using this technique the reason for the changes is not known yet. The figures 4.25c and 4.25d show the histograms of the final position estimation error for the 1000 Monte Carlo runs of vehicle 1 and vehicle 2. The mean square of this error is 13.14 units for both the vehicles, which is also lower than the straight line case, and some of the sinusoidal cases, but, the variance in this case is very high when compared to the other two cases. This is shown in table 4.1.



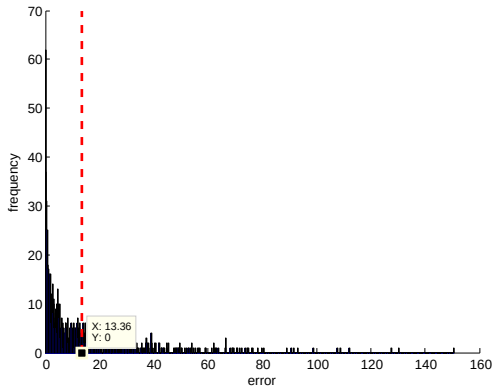
(a) Vehicle 1 x-position and y-position errors



(b) Vehicle 2 x-position and y-position errors



(c) Vehicle 1 histogram of the final position error estimation over 1000 Monte Carlo runs



(d) Vehicle 2 histogram of the final position error estimation over 1000 Monte Carlo runs

Figure 4.25: Error plots for vehicle 1 and vehicle 2 with the vehicles following the paths generated using the RRT technique where a cost function is included for the selection of points in the graph

4.11 Discussion

The simulations shown here are run for eight different trajectory types, each consisting of 1000 Monte Carlo runs. We can observe from the above plot that the vehicles moving in sinusoidal trajectories perform better than those moving in straight line trajectories, although the mean-square error from the straight line trajectory and the sinusoidal in-phase and out-phase are fairly close to each other. This may be due to the fact we need to consider that the vehicles in the sinusoidal case have traveled for more time (140 sec) than the straight line case (133 sec). The estimation performance of the vehicles is also analyzed by varying the frequencies and amplitudes. The simulations are run for 1000 Monte Carlo runs of the final position estimation error and the values are tabulated below.

Table 4.1: Comparison of mean square error in different trajectory types

Vehicle Motion	Transition	Frequency		Amplitude		Mean Error
		Agent1	Agent2	Agent1	Agent2	
Sinusoidal	In phase	615.38	615.38	41.2	41.2	15.51 ± 143.61
	Out phase	615.38	615.38	41.2	41.2	14.8 ± 129.02
	Different	615.38	666.67	40.6	44.65	2.974 ± 4.86
Sinusoidal	In phase	1600	1600	113.5	113.5	15.12 ± 128.25
	Out phase	1600	1600	113.5	113.5	14.82 ± 127.29
	Different	1600	1142.86	112	79.6	3.33 ± 6.51
Straight line	Same heading	NA	NA	NA	NA	15.38 ± 137.39
	Different heading	NA	NA	NA	NA	14.9 ± 132.49
RRT	Random	NA	NA	NA	NA	5.57 ± 63.92
Iterative Sampling	Random	NA	NA	NA	NA	21.36 ± 498.63
RRT with cost function	Random	NA	NA	NA	NA	13.14 ± 339.97

We can observe from the above table the mean square error is least in the case of sinusoidal trajectories when the vehicles are moving with different spatial periods. Also, the error is highest in the case of straight line trajectory and the sinusoidal in phase conditions. The simulations are again run for all the trajectory types shown in the table with different frequencies and amplitudes and the results are again tabulated in the following table.

Table 4.2: Comparison of mean square error in different trajectory types

Vehicle Motion	Transition	Frequency		Amplitude		Mean Error
		Agent1	Agent2	Agent1	Agent2	
Sinusoidal	In phase	800	800	54.8	54.8	15.59 ± 143.43
	Out phase	800	800	54.8	54.8	15. 37 ± 136.59
	Different	800	1142.86	54.8	79.06	2.67 ± 3.97
Sinusoidal	In phase	2000	2000	140	140	14.61 ± 124.01
	Out phase	2000	2000	140	140	12.58 ± 95.32
	Different	2000	1600	142.1	112.1	3.84 ± 8.26
Straight line	Same heading	NA	NA	NA	NA	15.38 ± 137.39
	Different heading	NA	NA	NA	NA	14.9 ± 132.49
RRT	Random	NA	NA	NA	NA	5.57 ± 63.92
Iterative Sampling	Random	NA	NA	NA	NA	21.36 ± 498.63
RRT with cost function	Random	NA	NA	NA	NA	13.14 ± 339.97

The simulation analysis done by changing the frequencies among the vehicles shows that the vehicles improve their performance (lower mean-square error) when both of the agents have different and higher frequencies than the other cases.

From Figure 4.16, we observe that the results from the simulations actually agree with the E-uniform Observability analysis done on different trajectory types for the two agents. From the E-uniform observability analysis in section 3.3.9 we concluded that the different frequencies case should actually perform better than all the other cases that we considered from figure 4.16. We notice that the mean square error in the final position estimation is 3.38 units which is as low as any other trajectory performance. Also, we observed from the analysis that the straight line case and the in phase case should yield similar results which is in accordance again with the simulation results. The mean square error from both the cases is equal with a tolerance of 0.5 units. The fact that the analysis shows the out of phase does perform better than the in phase case can be observed from the table 4.2. The mean square error from the out of phase case is quite lower than the in phase case when we have lower frequencies and higher amplitudes. The figure shown in 4.16 also shows the error obtained by using the RRT technique. The error from this technique is around 21.36 units more than the other cases though the time taken for vehicles using this technique to reach the goal location is around 159 seconds. The mean square error obtained using any of the RRT techniques is not as low as the sinusoidal case with the vehicles moving with different frequencies.

CHAPTER 5

Conclusion and Future work

5.1 Conclusion

In this research, observability analysis is presented for two Dubin's models based on the choice of the vehicle trajectory and the simulations to demonstrate the observability analysis are presented. The E-Uniform Observability analysis and the simulations on different trajectory types we considered show that the relative motion between the agents affects the observability of the system in the absence of an absolute position reference. This helps to reduce the drift in the navigation state estimation which occurs due to the absence of an absolute position reference. In other words the agents have low mean square error in the final position estimation when the agents have relative motion between them. From the tables 4.1 and 4.2, we can observe that the agents performance is better when they have different frequencies from each other and the mean square error is least in this case with 3.38 units when compared to other cases considered, in which the error is more than 14 units. The simulations have been carried out to compare the performances of the vehicles using different heuristic and sampling based techniques. From the results of the simulations we can actually show that the relative motion between the agents has an effect on the navigation state estimation and that the drift in the position estimation uncertainty can be reduced by certain relative motions of the agents while tracking different paths. Though the reason for the reduction in position estimation drift is not shown in this work, the observability analysis showed in this work helps better understand the effect of vehicle trajectory on the performance of the system. Certain trajectory types like the sinusoidal motion in which the vehicles moving with different frequencies guarantee that the position estimation uncertainty is significantly less than the vehicles moving in straight lines in the absence of an absolute position reference.

5.2 Future Work

- The work presented in this research concentrates on a 2D Dubin's model which can be extended in the future to 3D by considering the air vehicles.
- The observability analysis of the system can be analyzed when both inter agent range and bearing measurements are available while tracking different trajectories in the air vehicles.
- In this work, only certain set of trajectories are explored such as straight line paths, sinusoidal paths, RRT generated paths. This can be extended by considering other paths such as complicate polynomial paths, completely random paths, some more sampling based path planning algorithms like RRT-star etc.
- The future work can also be focused on to find a global optimal trajectory when no absolute position information is available.
- The simulations and algorithms can be tested in the real world on hardware.
- The reason for the abrupt changes in the estimated values from EKF can be examined more in detail, if it is related to vehicles relative motion.

Bibliography

[1] Aimee Turner. "tomorrow's engineers give air travel the bird". http://www.flightglobal.com/blogs/future-proof/2009/05/windowless-cabins-in-passenger.html, 2009.

[2] Marc D Richards, Darrell Whitley, J Ross Beveridge, Todd Mytkowicz, Duong Nguyen, and David Rome. Evolving cooperative strategies for uav teams. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 1721–1728. ACM, 2005.

[3] Myron Kayton and Walter R Fried. *Avionics navigation systems*. John Wiley & Sons, 1997.

[4] AD King. Inertial navigation-forty years of evolution. *GEC review*, 13(3):140–149, 1998.

[5] Henrik B Christophersen, R Wayne Pickell, James C Neidhoefer, Adrian A Koller, Suresh K Kannan, and Eric N Johnson. A compact guidance, navigation, and control system for unmanned aerial vehicles. *Journal of aerospace computing, information, and communication*, 3(5):187–213, 2006.

[6] Michael George and Salah Sukkarieh. Inertial navigation aided by monocular camera observations of unknown features. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3558–3564. IEEE, 2007.

[7] Bryce B Ready and Clark N Taylor. Improving accuracy of mav pose estimation using visual odometry. In *American Control Conference, 2007. ACC'07*, pages 3721–3726. IEEE, 2007.

[8] Rajnikant Sharma and Clark Taylor. Cooperative navigation of mavs in gps denied areas. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 481–486. IEEE, 2008.

[9] Andrey Soloviev and Adam J Rutkowski. Fusion of inertial, optical flow, and airspeed measurements for uav navigation in gps-denied environments. In *SPIE Defense, Security, and Sensing*, pages 733202–733202. International Society for Optics and Photonics, 2009.

[10] Y Uny Cao, Alex S Fukunaga, and Andrew Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous robots*, 4(1):7–27, 1997.

[11] CA Rabbath, E Gagnon, and M Lauzon. On the cooperative control of multiple unmanned aerial vehicles. *IEEE Canadian Review*, 46(1):15–19, 2004.

[12] Chi-Tsong Chen. *Linear system theory and design*. Oxford University Press, Inc., 1995.

[13] Leonard M Silverman and HE Meadows. Controllability and observability in time-variable linear systems. *SIAM Journal on Control*, 5(1):64–73, 1967.

[14] Ihnsoek Rhee, Mamoun F Abdel-Hafez, and Jason L Speyer. Observability of an integrated gps/ins during maneuvers. *Aerospace and Electronic Systems, IEEE Transactions on*, 40(2):526–535, 2004.

[15] Lester E Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, pages 497–516, 1957.

[16] Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Iowa State University, 1998.

[17] Markus Achtelik, Abraham Bachrach, Ruijie He, Samuel Prentice, and Nicholas Roy. Stereo vision and laser odometry for autonomous helicopters in gps-denied indoor environments. In *SPIE Defense, Security, and Sensing*, pages 733219–733219. International Society for Optics and Photonics, 2009.

[18] Girish Chowdhary, Eric N Johnson, Rajeev Chandramohan, Scott M Kimbrell, and Anthony Calise. Autonomous guidance and control of airplanes under actuator failures and severe structural damage. *Journal of Guidance Control and Dynamics*, 2012.

[19] Allen D Wu and Eric N Johnson. Methods for localization and mapping using vision and inertial sensors. *AIAA Paper*, 7441:2008, 2008.

[20] Mitch Bryson and Salah Sukkarieh. Observability analysis and active control for airborne slam. *Aerospace and Electronic Systems, IEEE Transactions on*, 44(1):261–280, 2008.

[21] Adam J Rutkowski. The most accurate path from point a to point b is not necessarily a straight line. In *AIAA Guidance, Navigation, and Control Conference*, 2012.

[22] Laurie N Bose and Arthur G Richards. Determining accurate visual slam trajectories using sequential monte carlo optimization. *AIAA Guidance, Navigation, and Control Conference*, 2013.

[23] Eagle S Jones and Stefano Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430, 2011.

[24] Lyudmila Mihaylova, Joris De Schutter, and Herman Bruyninckx. A multisine approach for trajectory optimization based on information gain. *Robotics and Autonomous Systems*, 43(4):231–243, 2003.

[25] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3565–3572. IEEE, 2007.

[26] Jur Van Den Berg, Pieter Abbeel, and Ken Goldberg. Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, 30(7):895–913, 2011.

[27] Stergios I Roumeliotis and George A Bekey. Collective localization: A distributed kalman filter approach to localization of groups of mobile robots. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 3, pages 2958–2965. IEEE, 2000.

[28] Agostino Martinelli and Roland Siegwart. Observability analysis for mobile robot localization. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1471–1476. IEEE, 2005.

[29] Adrian N Bishop, Barış Fidan, Brian DO Anderson, Kutluyıl Doğançay, and Pubudu N Pathirana. Optimality analysis of sensor-target localization geometries. *Automatica*, 46(3):479–492, 2010.

[30] Kutluyıl Doğançay and Hatem Hmam. Optimal angular sensor separation for aoa localization. *Signal Processing*, 88(5):1248–1260, 2008.

[31] Cindy Leung, Shoudong Huang, Ngai Kwok, and Gamini Dissanayake. Planning under uncertainty using model predictive control for information gathering. *Robotics and Autonomous Systems*, 54(11):898–910, 2006.

[32] Eric W Frew, Jack Langelaan, and Sungmoon Joo. Adaptive receding horizon control for vision-based navigation of small unmanned aircraft. In *American Control Conference, 2006*, pages 6–pp. IEEE, 2006.

[33] Alexei A Makarenko, Stefan B Williams, Frederic Bourgault, and Hugh F Durrant-Whyte. An experiment in integrated exploration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 534–539. IEEE, 2002.

[34] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 25(3):195–207, 1998.

[35] Yüksel Subaşi and Mübeccel Demirekler. Quantitative measure of observability for stochastic systems. In *World Congress*, volume 18, pages 4244–4249. IFAC, 2011.

[36] Sinpyo Hong, Man Hyung Lee, Ho-Hwan Chun, Sun-Hong Kwon, and Jason L Speyer. Observability of error states in gps/ins integration. *Vehicular Technology, IEEE Transactions on*, 54(2):731–743, 2005.

[37] Guoquan P Huang, Anastasios I Mourikis, and Stergios I Roumeliotis. Analysis and improvement of the consistency of extended kalman filter based slam. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 473–479. IEEE, 2008.

[38] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[39] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *Automatic Control, IEEE Transactions on*, 51(3):401–420, 2006.

[40] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[41] Ali Jadbabaie, Jie Lin, and A Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48(6):988–1001, 2003.

[42] Philippe Bonnifait and Gaëtan Garcia. Design and experimental validation of an odometric and goniometric localization system for outdoor robot vehicles. *IEEE Transactions on robotics and automation*, 14(4):541–548, 1998.

[43] Rajnikant Sharma, Randy W Beard, Clark N Taylor, and Stephen Quebe. Graph-based observability analysis of bearing-only cooperative localization. *Robotics, IEEE Transactions on*, 28(2):522–529, 2012.

[44] Rajnikant Sharma. Observability based control for cooperative localization. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 134–139. IEEE, 2014.

[45] Huili Yu, Rajnikant Sharma, Randal W Beard, and Clark N Taylor. Observability-based local path planning and collision avoidance for micro air vehicles using bearing-only measurements. In *American Control Conference (ACC), 2011*, pages 4649–4654. IEEE, 2011.

[46] Robert Leishman, John Macdonald, Tim McLain, and Randy Beard. Relative navigation and control of a hexacopter. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4937–4942. IEEE, 2012.

[47] Andrea Balluchi, Antonio Bicchi, Aldo Balestrino, and Giuseppe Casalino. Path tracking control for dubin's cars. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3123–3128. IEEE, 1996.

[48] Robert Hermann and Arthur J Krener. Nonlinear controllability and observability. *IEEE Transactions on automatic control*, 22(5):728–740, 1977.

[49] Gildas Besançon. *Nonlinear observers and applications*, volume 363. Springer, 2007.

[50] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, 2011.

VITA

Jayadeep Pabbisetty

Candidate for the Degree of

MASTER OF SCIENCE

Thesis: IMPROVING NAVIGATION THROUGH COOPERATION AND PATH PLANNING

Major Field: Mechanical and Aerospace Engineering

Biographical:

Personal Data: Born in Vijayawada, India.

Education: Completed the requirements for the Master of Science degree with a major in Mechanical and Aerospace Engineering at Oklahoma State University in May, 2015. He completed the requirements for the Bachelor of Technology with a major in Mechanical Engineering at Jawaharlal Nehru Technological University, Hyderabad, India, in 2012.

Experience:

Intern: University of Florida from May to August 2013 working on the project "Improving Navigation Through Cooperation and Path Planning" with Air Force Research Laboratory.

Graduate Teaching Assistant: School of Mechanical and Aerospace Engineering, Oklahoma State University from August 2012 to present. Courses include: Design of Machines Course (MAE 3033), Dynamic Systems 1 (MAE 3723), and Automatic Control Systems (MAE 4053).

Graduate Research Assistant: School of Mechanical and Aerospace Engineering, Oklahoma State University from August 2013 to present working with DASLAB under Girish Chowdhary.