Vessel Route Prediction from AIS Data

K C L Perera

2017

Vessel Route Prediction from AIS Data

A dissertation submitted for the Degree of Master of Computer Science

K C L Perera
University of Colombo School of Computing
2017

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name:

Registration Number:

Index Number:

____

Signature: Date:

This is to certify that this thesis is based on the work of

Mr./Ms.

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name:

____

Signature: Date:

Abstract

Route prediction for vessels when implemented on a vessel traffic monitoring solution can be used to prevent collisions and illegal activities from happening. Since route prediction will provide in-depth knowledge on a ship's route at least for a few minutes, monitoring officers and ship captains will know an approximated location of all ships after a few minutes. This will help them to find out if there is a possibility of ships will collide with each other or with an object on ground before few minutes. This will help them to change ship's course and speed in order to avoid collisions.

Vessel Traffic Management System (VTMS) which was developed locally was the subject of this research. Even though the system already has a path prediction system, it is not accurate enough to carry out predictions when the ships move at higher speeds and changing their courses quickly. Hence there is a need of better prediction model for this VTMS.

Even though most of the commercial vessel traffic monitoring systems have path prediction algorithms with them, they are not available for public. Thus, this research explores an efficient method of predicting vessel paths using Kalman Filter based techniques. These techniques use previous data when predicting the path. Hence the database of VTMS was used as the test data for this research.

Kalman filter and its variation are very popular methods for the solutions to prediction related problems. Thus, Kalman filter based solution was developed and tested with test cases. Developed solution was able to tackle the objective of this research giving better performances and accuracy.

Acknowledgements

I would like to thank my supervisors, Dr. K. D. Sandaruwan and Prof. N. D. Kodikara who were always available to give me advice and guidance.

A big thank should go to people who did researches in these areas, without whom I wouldn't be able to do this report on my own.

A special thank should go to my parents for supporting me in everything I do by giving me the freedom to do anything I like and making me who I am today.

Finally, I wish to thank all my friends who gave me all the support and strength needed.

Contents

Table of Figures

List of Tables

List of Abbreviations and Acronyms

AIS	Automatic Identication System
ARPA	Automatic Radar Plotting Aids
AVL	Automatic Vehicle Location
COG	Course Over Ground
GPS	Geographical Positioning System
IMO	International Maritime Organization
MMSI	Maritime Mobile Service Identity
MSS	Maritime Safety and Security
SOG	Speed Over Ground
VHF	Very High Frequency
VTMS	Vessel Traffic Management System
VTS	Vessel Traffic Service

Chapter 1 Introduction

1.1 Maritime Transportation

Due to globalization, large amounts of goods need to be transported all over the world. Since large amounts of goods can be transported in a single ship, the transportation costs can be decreased. Many goods do not need to be delivered very urgently. Hence transporting goods by waters has become the most popular way of goods transportation even though it takes more time to deliver than an air plane or truck based transportation. Due to low production costs in some countries like china, many goods have to be exported to other countries. Thus, there will be heavy traffic around harbours of those countries as well as waterways around those countries since the transportation of those exported goods are being done by waters in ships.

1.2 Vessel Traffic

Since maritime transportation has become the most popular way of transportation, there is a heavy traffic around harbours and waterways. Hence there exists a high possibility of happening of an accident around the high traffic areas. Collisions can happen between ships or between ships and an object in sea which would result in heavy losses on the financial side as well as other casualties. Due to the high intensity of traffic around harbours, the port operations can be delayed if the vessel traffic is not controlled properly. Thus having a proper vessel traffic monitoring methodology will increase the efficiency of port operations and improve the navigational safety by avoiding possible accidents. There are situations like illegal immigrants come to countries by waters and smugglings happen around coastal areas. These situations can be monitored by coastal guards if they have a proper way of vessel monitoring. Coast guards monitor vessel movement around the coastal areas of the country and react on noticeable situations using Maritime Safety and Security (MSS) Systems. Monitoring happens around harbours are being managed with Vessel Traffic Services (VTS)

1.3 Vessel Traffic Service

A VTS is a marine traffic monitoring system which is controlled by a harbour or a port. Normally a VTS system contains radar, Closed Circuit Television (CCTV), VHF transceivers, voice communication and Automatic Identification System (AIS) in order to track and get vessel static data and dynamic positional data to provide safety for a limited area [1]. This is a dynamic data driven application system in which the data captures dynamically through AIS and other sources and displays them in 3D or 2D viewer dynamically.

“VTS is governed by SOLAS (safety of life at sea) Chapter V Regulation 12 together with the Guidelines for Vessel Traffic Services [IMO Resolution A.857(20)] adopted by the International Maritime Organization on 27 November 1997” [2].

The purpose of a VTS is to improve the navigational safety by providing information regarding traffic and other geographical characteristics about a specific area and giving advices in order to avoid unnecessary traffic to masters who control the vessel [3].

A harbour VTS or VTMS (Vessel Traffic Management System) should be able to assist the control officers to manage the port operations by providing information and visualizations.

Navy or Coast Guard Officers use VTS or VTMS to monitor coastal areas of their country in order to prevent illegal immigrants’ coming to the country as well as to detect and prevent illegal smuggling activities happening around the coastal areas.

In order to visualize vessel traffic for VTS and VTMS, data has to be captured from data sources. In which AIS is an important source for data capture as it transmits positional and voyage related data.

1.4 Motivation

VTMS solutions plot the Geographical Positioning System (GPS) locations of the vessels it receives via the AIS data on a map or a virtual harbour environment and show the movements and other data of the vessel in real-time to assist vessel traffic monitoring officers that use the VTMS solution in taking decisions. Since AIS data is not continuous, vessels do not move smoothly on the map or the virtual harbour environment and will cause jump/hop routing for vessels. Furthermore, there can be abnormal tracks such as tracks on land when drawing the trails of the vessels since VTMS will connect the previous GPS location and the current location. This will drop the behavioural realism of the VTMS solution.

Many proprietary VTMS solution have tackled these problems and have come up with solutions to smooth vessel routing and good vessel route predictions. Open source VTMS solutions does not have a proper way to tackle the problem of jump/hop vessel routing even though some VTMS has the capability of predicting a position after a pre-defined time. Siyara VTMS is currently facing the same issue as the above mentioned open source VTMSs.

Siyara VTMS’s and many open source VTMS’s route prediction uses the current speed, the course of the ship and the pre-defined time period and do the calculation using the equation $S = vt$. Where *v* is the speed over the ground of the ship, *t* is the pre-defined time period and *S* is the distance from the current position. After finding the distance the prediction system draws a line with a length of the calculated distance on the ship's current position on the direction of ship's course over ground. This prediction is good when a ship is going on a straight line. When the ship is not travelling straight this will fail as the course of the ship will change from one location point to next location point.

In order to avoid above mentioned abnormalities, next location point has to be predicted using the previously received AIS data locations and the prediction process has to be done in real-time. This prediction can be either a short-term prediction or a long-term prediction. These kinds of predictions can be done using either a Dynamic model, a Statistical model or a Trained model.

1.5 Objective

Main objective of the project was to predict next GPS location in real-time using the previous GPS locations received through the AIS and interpolate the trail between two GPS locations in order to avoid abnormal trails of vessels plotted on a map in a VTMS solution. Siyara VTMS was used to collect test data sets and test the outputs through visualization.

First it was needed to identify the most feasible prediction model type for this particular problem. Then an algorithm that is compatible with the data set had to be identified. AIS data contains vessel's position data as well as static vessel data such as its type and dimensions. Hence the data set will consist of a vessel's mechanical properties and position data over time. After analysing these data which prediction model type should be used in this project had to be decided.

Prediction based research projects had to be studied to identify algorithms, develop a solution using the algorithm and refine the solution to achieve better results.

1.6 Research Problem

Most of the commercial VTMS's have the dynamic route prediction modules which have high accuracy and they have smooth vessel routing. Unfortunately, these solutions are not available to access or to find and these are definitely not open source solutions. There are research projects that have been done on path predictions for either vehicle based navigation [4][5] or maritime navigation [6][7][8] domain. Most of these research projects are based on either a mathematical model or a statistical model. Some of these solutions are computationally intense and can't be used in this situation as it needs to predict location very quickly. In this research, it needs to find a method to predict the next location point using the most recent AIS data. Thus, this has to be done on real-time and it has to be efficient in order to give results quickly. This prediction may need to be done in few milliseconds and prediction process may need to be carried on for few minutes.

Can the next location point be predicted in real-time by using most previous location data points?

1.7 Scope

Research study was followed in order to identify the algorithms and prediction model used to refine the vessel routing based on AIS data. At the moment, Kalman filtering algorithm and Particle Filter algorithm seem to be the best algorithms for the task. Kalman Filter variations had to be studied as well. In order to clarify the fact that which algorithm is the most suitable algorithm for the task, few other techniques had to be studied.

After selecting the algorithm, a solution had to be implemented to test using the test data. If the test results are not satisfying, refinements have to be made on the method. After refining the solution, it should be plugged in to a real-time visualizing framework to test the result. Siyara VTMS developed by UCSC will be used to collect test data for the visualization and testing purposes.

1.8 Outline of the Report

In the second chapter, related work in the problem domain will be explored. In the third chapter, analysis on the selected algorithm and methodologies will be explained. In the fourth chapter, design of the proposed solution with the implementation will be discussed. In the fifth chapter, analysis of the test results and comparisons will be discussed. In the sixth chapter, conclusions and future work will be discussed.

Chapter 2 Background

This chapter will describe the key components which will help to determine the methodology for the solution design while describing background information and providing few literatures based on the research domain.

2.1 Automatic Identification System

AIS is a system which used in vessels to automatically track them in order to identify and locate by sending data to surrounding vessels and control stations [9]. AIS uses VHF signal transmission as data source of communication. AIS can provide many static and dynamic information about vessels and transmits them through the VHF frequencies. AIS transmits data through two VHF frequency channels [10].

- 161.975 MHz
- 162.025 MHz

AIS has a horizontal range limit of 74 kilometres and a vertical range of 400 kilometres. AIS transmits data every 2 to 10 seconds on fast moving or manoeuvring vessels and every 3 minutes on anchored or moored vessels [9]. AIS is less affected and will have minor interferences in the sea clutter. AIS will not be affected by weather and changing sea conditions. Information given by AIS complements to Automatic Radar Plotting Aids (ARPA) [11]. Marine AIS has two versions for vessels as class A and class B. According to IMO regulations international ships with gross tonnage of 300 or above, non-international ships with 500 or more gross tonnages and all passenger ships should have class A AIS. Class B AIS is for smaller ships [11].

AIS transceivers can transmit 27 types of messages defined in ITU 1371-4. Those messages contain various static data on ship as well as dynamic positioning data on ship.

Few important message types send by ships [12].

- Message type 1 - Class A position report
- Message type 5 - Class A static and voyage related data
- Message type 18,19 - Class B position report
- Message type 24 - Class B static and voyage related data

Following data transmits from every moving ship with a frequency of 2 to 10 seconds and every anchored ship with a frequency of 3 minutes [9].

- Vessel's MMSI
- Navigation status

- Course over ground
- Speed over ground
- True heading
- Latitude
- Longitude

Following data will be transmitted with a frequency of 6 minutes [9].

- Name of the ship
- Call sign
- Ship type
- Dimensions of ship
- Destination

Since 2007, AIS is mandatory for sea faring vessels [13].

2.2 Literature Review on Prediction Based Researches

2.2.1 Predictions for Vehicle Based Navigation

Adaptive Kalman Filtering for Vehicle Navigation by Congwei Hu, Wu Chen,Yongqi Chen and Dajie Liu [5] discusses on two adaptive algorithms which were applied to GPS data. Few test have been carried out on test data to demonstrate performances of the two algorithms and to compare those results with conventional Kalman filter for vehicle navigation. The adaptive filtering with fading memory algorithm and the adaptive filter with variance component estimation are the two algorithms used in this paper. Both these algorithms are Kalman filter based algorithms. This paper shows that these two algorithms are better than the conventional Kalman filter algorithm.

Vehicle Route Prediction and Time of Arrival Estimation Techniques for Improved Transportation System Management by Abdolreza Karbassi and Matthew Barth [4] describes vehicle route prediction and time of arrival estimation system that has been implemented using cars that have Automatic Vehicle Location systems. These AVL systems send position and time data of the vehicles. In this system's prediction algorithm, initially the system assumes that the current route is the route with the highest probability for that time and day. This is determined from the pre-computed route probability data. When the real-time location data is received, system matches coarse trajectory data to appropriate roadways. When new data are received, a hierarchical tree data structure is used to recalculate most probable route. At the conclusion of the paper authors say by using Kalman filter techniques much better predictions can be achieved.

2.2.2 Predictions for Ship Based Navigation

Ocean Vessel Trajectory Estimation and Prediction Based on Extended Kalman Filter by Lokukaluge P. Perera and Carlos Guedes Soares [7] presents a manoeuvring ship model which predicts position data using Extended Kalman Filter techniques and shows results of this model with respect to the test data. There are three sections for the prediction and estimation of trajectories described in this paper. Target Motion Model, Measurement Model and Associated Techniques and Trajectory Tracking & Estimation are the three sections described. Continuous time Curvilinear Motion Model was selected as the Target Motion Model while Measurement Model was formulated as a discrete time linear model. For Trajectory Tracking and Estimation Process Extended Kalman Filter has been used due to its capability of capturing non-linear states.

Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation and Trajectory Prediction by Lokukaluge P. Perera, Paulo Oliveira and C. Guedes Soares [8] have two sections in which the first section describes a solution for detecting and tracking of multiple vessels. For this, authors propose an Artificial Neural Network based solution. Second section of this paper describes a solution for state estimation and trajectory prediction of a single vessel. For this, authors propose a solution with an Extended Kalman Filter. Apart from that the paper contains test results for the proposed solutions. Prediction has been done on position, velocity and the acceleration.

Knowledge-Based Vessel Position Prediction using Historical AIS Data by Fabio Mazzarella, Virginia Fernandez Arguedas, Michele Vespe [6] propose a system that uses a Particle Filter based Bayesian prediction model to predict vessel positions. Authors have used the knowledge of traffic routes in order to enhance the system. They have applied the solution on historic traffic data and presents those results.

Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction by Giuliana Pallotta, Michele Vespe, Karna Bryan [14] propose a solution that provides decision support capability to Vessel Traffic Monitoring Systems using large amounts of AIS data. System extract knowledge via an incremental learning approach. That enables system to adapt into evolving situations. After extracting vessel traffic and motion data, predicting ship movements, based on the trajectories of past vessels on the same route can be done. After extracting data, routes are being classified and assigned a probability compatible to the vessel position. Using these probability values and considering the state vector of the ship, it can be predicted that the ship is in what route.

2.3 Kalman Filter

In 1960 Rudolf E. Kalman published a paper[15] describing a recursive solution to linear filtering problem of discrete data[16]. This solution was named as Kalman Filter after Rudolf Kalman. Kalman filter is an optimal estimator as it can deduce the parameters under observation from indirect, uncertain and inaccurate observations[17]. Since it is recursive, measurements can be processed in real-time to make corrections.

Kalman filter is a set of mathematical equations[16]. There are two parts of Kalman filter algorithm. 1st one is the predictor part where a prediction will be made on the next state observing the properties of the current state and its controller parameters. 2n part of Kalman filter is corrector phase where a measurement from a sensor which may not be accurate, is observed and some corrections will be made to the output of the predicted phase.

Kalman filter has been very popular in many areas since its introduction, especially in autonomous navigation. This is due to Kalman filter's relative simplicity and the robustness[16] as well as its capability process real-time information[17]. Hence Kalman filter technique can be used for real-time path prediction problems.

2.4 Map Projections

Earth is a spherical object where a location point is given by latitudes and longitudes. In order to do distance and direction calculations, those latitude and longitude coordinates have to be mapped in to a plane. The process of systematic transformation of latitude and longitudes in to locations on a plane is called a map projection[18].

There are three steps to create a map projection. In each step, some information will be lost. 1st step is selection of a model for the shape of the earth where the shape has to be determined from either a sphere or an ellipsoid. 2nd step is transforming geographic coordinates into plane coordinates. 3rd step is the reducing the scale[19].

2.5 Haversine Formula

Haversine formula calculates great circle distance between two points on a sphere[20]. Which means Haversine formula is able to determine the shortest distance between two points on spherical surface[21]. As the earth is a spherical object where a point on the surface of earth is given by a latitude and longitude value pair, distance between two points on the surface of the earth can be calculated by using the Haversine formula without having to project the earth coordinates in to plane coordinates.

2.6 Summary

In this chapter, background information needed for this research were discussed at first. Then the discussion moved on to literature on the domain of predictions. In the end, identified methodologies and techniques were elaborated briefly. In the next chapter, theories and methodologies for implementation will be discussed.

Chapter 3 Methodology

This chapter will elaborate on the theories and methodologies used to design and implement solution for the research problem while justifying the selection of those particular methods thoroughly.

3.1 Kalman Filter

As mentioned earlier main objective of this research is to predict a ship's next location point in real-time using previous location points. This prediction can be either a short-term prediction or a long-term prediction. Thus, the solution for this problem lies in the domain of motion prediction over time. Kalman Filter based techniques are a very popular algorithm set for this category.

Kalman Filter uses recursive approach. Hence it is capable of processing new measurements real-time. If all noise is Gaussian, Kalman filter minimizes the mean square errors of estimation parameters. [17] Kalman filtering process consists of two steps. [16]

1. Prediction Phase
    - Project the state ahead.
    - Project the error covariance ahead
2. Update Phase
    - Compute Kalman Gain
    - Update estimate with measurement
    - Update error covariance

Kalman Filter assumes that the state at time t evolves from a prior state at time $t-1$. [22]

3.1.1 Prediction Model

Kalman filter model assumes that the new system state will evolve from previous state according to the following equation[22].

$$x_{t|t-1} = F_t x_{t-1|t-1} + B_{t-1} u_{t-1} + \mathrm{w_t} \qquad (3.1)$$

Where;

$x_{t|t}$ - the state estimate at time t given observation up to and including time t.

B_t – Control input matrix which applies effect of each control input parameter on state vector

u_t – Vector containing any control input

F_t - the state transition matrix which applies the effect of each system parameter at time t-1 on time t.

w_t - the vector containing process noise for each term in the state vector.

The process noise is assumed to be zero mean multi variate distribution based values with covariance given by covariance matrix Q_t.

3.1.2 Measurement Model

Measurements of the system can be performed using the following equation[22].

$$z_t = H_t x_{t|t-1} + v_t \tag{3.2}$$

Where;

H_t – Transformation matrix that maps state vector parameters in to measurement domain.

z_t – Vector of measurements

v_t - the vector containing measurement noise for each term in the measurement vector.

Measurement noise is assumed to be zero mean Gaussian white noise with covariance R_t.

3.1.3 Prediction Phase

Equations for prediction phase of Kalman filter are as follows.

$$x_{t|t-1} = F_t x_{t-1|t-1} + B_{t-1} u_{t-1} \tag{3.3}$$

$$P_{t|t-1} = F_t P_{t-1|t-1} {F_t}^T + Q_t \tag{3.4}$$

Where;

$P_{t|t}$ – Predicted process covariance matrix at time t given observation up to and including time t. The terms along the main diagonal of $P_{t|t}$ are the variances associated with the corresponding terms in the state vector. The off-diagonal terms of $P_{t|t}$ provide the covariance between terms in the state vector[22].

Equation 3.3 projects the state ahead while equation 3.4 projects process covariance for next state.

3.1.4 Measurement Update Phase

Equations for measurement update phase of Kalman filter are as follows.

$$y_t = z_t - H_t x_{t|t-1} \tag{3.5}$$

$$K_t = P_{t|t-1}H_t{}^T\left(H_tP_{t|t-1}H_t{}^T + R_t\right)^{-1} \quad (3.6)$$

$$x_{t|t} = x_{t|t-1} + K_ty_t \quad (3.7)$$

$$P_{t|t} = (I - K_tH_t)P_{t|t-1} \quad (3.8)$$

Where;

y_t – Measurement residual

K_t – Kalman gain

I – Identity matrix

Equation 3.5 calculates measurement residual while equation 3.6 calculates Kalman gain whereas equation 3.7 and equation 3.8 update estimate with measurement and error covariance respectively.

3.1.5 Kalman Filter Algorithm

As mentioned earlier Kalman filter has two phases and they have separate groups of equations. In Kalman filter algorithm, prediction phase or time update phase equations provide predictions for next state while measurement update phase equations provide corrections to those predictions provided on the previous phase. These two phases will occur recursively one after other while the algorithm is working where final output of state and covariance will be inputs for next state predictions (Figure 3.1).

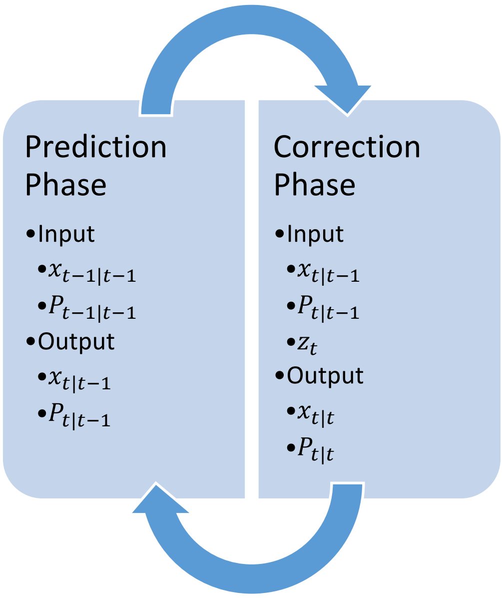

Figure 3.1- Cycle of Kalman Filter Process

3.2 Haversine Formula

Haversine formula is used to calculate great circle distance between two points on a sphere's surface. Thus, by using haversine formula distance between two points on earth's surface can be calculated. In this research project movement distance is considered between two geo coordinates. Hence the haversine formula will be used to calculate distance between two geo-coordinates instead of using map projections where errors can be occurred in each step of projecting geo coordinates in to plane coordinates.

3.2.1 Haversine Function

$$hav(\theta) = sin^2\left(\frac{\theta}{2}\right) = \frac{1-\cos\theta}{2} \qquad (3.9)$$

Haversine function is called haversine as it is half of versine which is also known as versed sine. Versine of an angle equals 1 minus its cosine[23].

3.2.2 The Haversine Formula

For any two points on a sphere, the haversine of the central angle between them is given by

$$hav\left(\frac{d}{r}\right) = hav(\varphi_2 - \varphi_1) + \cos\varphi_1 \cos\varphi_2 hav(\lambda_2 - \lambda_1) \qquad (3.10)$$

Where;

d - Great circle distance between two points

r – Radius of the sphere

φ_1 – Latitude of point 1 in radians

φ_2 – Latitude of point 2 in radians

λ_1 – Longitude of point 1 in radians

λ_2 – Longitude of point 2 in radians

$$d = 2r\sin^{-1}\left(\sqrt{sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right) + \cos\varphi_1\cos\varphi_2 sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right) \qquad (3.11)$$

Using the equation 3.11 the distance between two points on a sphere's surface can be calculated.

3.3 Summary

In this chapter, at first, theories and the process of Kalman filter were elaborated thoroughly while explaining the model equations and the two phases of the process. Then the discussion was moved on to haversine formula which will be used for converting geocoordinates in to distances. In the next chapter, implementation process and the algorithm used to determine the solution will be discussed with the selection of data parameters and data pre-processing.

Chapter 4 Implementation

This chapter will elaborate on selection of parameters and data pre-processing at first. Then it will move on to elaborate on the implementation of the Kalman filter based solution while describing the steps of the implementation.

4.1 Filter Parameters of Interest from Data Received via AIS

As mentioned earlier, AIS messages sends various static and dynamic data about vessels. Thus, Siyara VTMS has a very large database. Necessary data for this research study to be carried out, had to be extracted from the huge database of Siyara VTMS.

Parameters such as longitude, latitude, speed over ground, course over ground and time the message received were selected as the parameters of interest. Formats of those selected parameters as they were transmitted via AIS as follows[24].

- Longitude in 1/10 000 minutes (+/-180 degrees, East = positive (as per 2's complement), West = negative (as per 2's complement). 181degrees = not available = default)
- Latitude in 1/10 000 minutes (+/-90 degrees, North = positive (as per 2's complement), South = negative (as per 2's complement). 91degrees = not available = default)
- Speed over ground in 1/10 knot steps (0-102.2 knots) 1 023 = not available, 1 022 = 102.2 knots or higher.
- Course over ground in 1/10 = (0-3599). 3600 = not available = default. 3 601-4 095 should not be used.
- Timestamp - UTC second when the report was generated by the electronic position system (EPFS) (0-59, or 60 if time stamp is not available, which should also be the default value, or 61 if positioning system is in manual input mode, or 62 if electronic position fixing system operates in estimated (dead reckoning) mode, or 63 if the positioning system is inoperative)

These raw data have to be pre-processed to use for the solution as inputs.

4.2 Data Pre-processing

Since the parameters of interest are in different domains (SOG is on knots, time is in seconds), data has to converted in to the same domain to apply them to Kalman filter.

4.2.1 Converting Speed Over Ground

Speed over ground is received via AIS as nautical miles per hour which is also known as knots. Since we are going to deal with meters and seconds on this research, SOG has to be converted from knots to meters per second (ms^{-1}).

1 nautical mile is exactly 1.852 kilometres. Which means 1 nautical mile is 1852 meters. Thus, in order to convert SOG from knots to ms^{-1}, SOG value has to be multiplied by $\frac{1852}{3600}$.

$$1knot = \frac{1852}{3600} ms^{-1}$$

Thus,

$$1knot = 0.54444ms^{-1} \qquad (4.1)$$

Then the converted SOG values have to be divided into resolution components on x and y directions.

Since the AIS transmits Course Over Ground and the course over ground is determined by the bearing of the ship with respect to the north, SOG can be divided into X and Y direction components using the COG.

If X direction SOG component is $\dot{x}$ and Y direction SOG component is $\dot{y}$. Then,

$$\dot{x} = SOG \sin COG \qquad (4.2)$$

$$\dot{y} = SOG \cos COG \qquad (4.3)$$

4.2.2 Converting Geocoordinates into Distances

Since the latitude and longitude values cannot be used as inputs for Kalman filter and all the measurements are on meters and seconds domain, geo-coordinates have to be converted in to distance values. In this research, the distance values were calculated from the point (0,0) of the world. For the purpose of distance calculations, haversine formula was used.

For the calculations of distances, haversine formula needs the radius of the sphere under consideration. In this case, it is the earth. Thus, the radius of the earth was considered as 6,371,000 meters. Distances were calculated separately on X direction and Y direction.

Implementation of the converting geocoordinates into distance is as follows (Figure 4.1).

```
public static double getDistance(double lat1, double lat2, double lon1, double lon2) {
    double distance;
    double latDiff = Math.toRadians(lat2 - lat1);
    double lonDiff = Math.toRadians(lon2 - lon1);
    lat1 = Math.toRadians(lat1);
    lat2 = Math.toRadians(lat2);
    double r = 6371000;
    double a = Math.pow(Math.sin(latDiff / 2), 2) + Math.cos(lat1) * Math.cos(lat2) * Math.pow(Math.sin(lonDiff / 2), 2);
    double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
    distance = r * c;
    return distance;
}
```

Figure 4.1 - Implementation of Coordinate to Distance Conversion

4.3 Kalman Filter

In order to implement a Kalman filter based solution, latitude, longitude coordinates, SOG (V) and COG (θ) are selected as the interested properties for the prediction process.

In order for the prediction to be accurate, as mentioned earlier, geocoordinates and SOG had to be converted into same domain of measurements. Instead of geocoordinate values, distances were used in meters while SOG was converted from nautical miles per hour to meters per second.

4.3.1 Time Update Process

As the first step of the solution, state transition function had to be defined. Since the prediction is being carried out on X and Y directions using distances and velocities, kinematic equations can be used to define the state transition.

Suppose;

x – Distance on X axis

y – Distance on Y axis

$\dot{x}$ – Velocity on X direction ($V \sin\theta$)

$\dot{y}$ – Velocity on Y direction ($V \cos\theta$)

$\ddot{x}$ – Acceleration on X direction

$\ddot{y}$ – Acceleration on Y direction

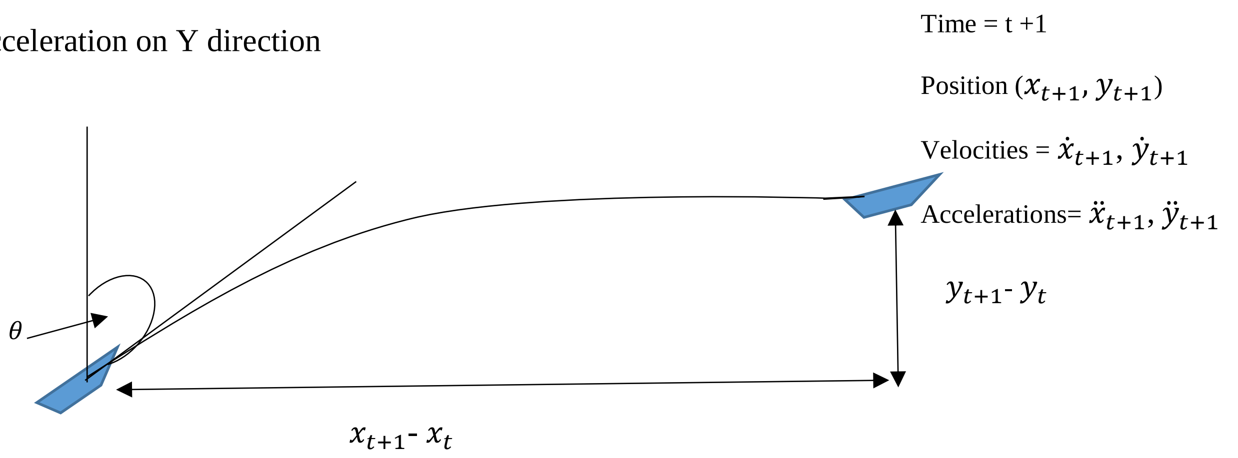

Figure 4.2 - Sample Movement of Vessel

Consider Figure 4.2.

Then using $S = ut + \frac{1}{2}at^2$ on X direction

$$x_{t+1} - x_t = \dot{x}_t dt + \frac{1}{2}\ddot{x}_t (dt)^2$$

$$x_{t+1} = x_t + \dot{x}_t dt + \frac{1}{2}\ddot{x}_t (dt)^2 \qquad (4.4)$$

Furthermore, using $v = u + at$ on X direction

$$\dot{x}_{t+1} = \dot{x}_t + \ddot{x}_t dt \qquad (4.5)$$

These linear equations can be written in matrix form as

$$\begin{pmatrix} x_{t+1} \\ \dot{x}_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & dt \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_t \\ \dot{x}_t \end{pmatrix} + \begin{pmatrix} \frac{(dt)^2}{2} \\ dt \end{pmatrix} \ddot{x}_t \qquad (4.6)$$

It is possible to similarly show that,

$$\begin{pmatrix} y_{t+1} \\ \dot{y}_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & dt \\ 0 & 1 \end{pmatrix} \begin{pmatrix} y_t \\ \dot{y}_t \end{pmatrix} + \begin{pmatrix} \frac{(dt)^2}{2} \\ dt \end{pmatrix} \ddot{y}_t \qquad (4.7)$$

According to equations 4.6 and 4.7 state vectors can be determined as;

$$\begin{pmatrix} X \\ \dot{X} \end{pmatrix}$$

$$\begin{pmatrix} Y \\ \dot{Y} \end{pmatrix}$$

In this research, it uses two parameters for the state vector at once. Thus, the computations are done twice to achieve values for 4 parameters.

State transition matrix for the solution is - $\begin{pmatrix} 1 & dt \\ 0 & 1 \end{pmatrix}$

Control input matrix for this solution is - $\begin{pmatrix} \frac{dt^2}{2} \\ dt \end{pmatrix}$

Control vector for this solution is the acceleration on either direction. Which would be $\ddot{x}$ and $\ddot{y}$.

For the initial process covariance calculation, a process covariance matrix was defined with variances of parameters on the states, placed on diagonals and the covariance between parameters on the off diagonals. Process noise covariance matrix $Q = I.q$ Where I is an identity matrix and q is a positive integer.

4.3.2 Measurement Update Phase

In this phase, it is needed to have a measurement vector for the process to make correction. In this case, the calculations cannot wait till the next AIS data point. Instead the solution has to determine next location by using previous data. For this research as the measurement vector, calculated output of distance and velocity were provided as measurements.

Measurement for distance on x direction was taken as $x_t + \dot{x}_t dt + \frac{1}{2}\ddot{x}_t (dt)^2$

Measurement for velocity on x direction was determined as $\dot{x}_t + \ddot{x}_t dt$

It is possible to similarly determine measurement values for distance and velocity on Y direction.

After determining the measurement vector, using equations 3.5, 3.6 , 3.7 and 3.8 respectively, calculation of measurement residual, calculation of Kalman gain, updating of estimate with measurement and updating of error covariance can be done.

For these processes following matrices were assumed.

Measurement noise covariance matrix $R_t - \begin{pmatrix} Var(X) & 0 \\ 0 & Var(\dot{X}) \end{pmatrix}$ and $\begin{pmatrix} Var(Y) & 0 \\ 0 & Var(\dot{Y}) \end{pmatrix}$

Where; $Var(X)$ – variance of X

Transformation matrix H was assumed as an identity matrix since the parameters on measurement vector and state vector are same.

4.3.3 Kalman Filter Algorithm

Pseudo code snippet for the algorithm used for the solution is as follows. This algorithm continuously executes recursively.

```
Void kalmanPrediction (initialPosition, timeInterval){

        While(true) {

                Calculate measurements for x, y, ẋ, ẏ;

                Project next state;

                Project process covariance;

                Compute Kalman Gain;

                Update estimate with measurement;
```

Update process covariance;

}

}

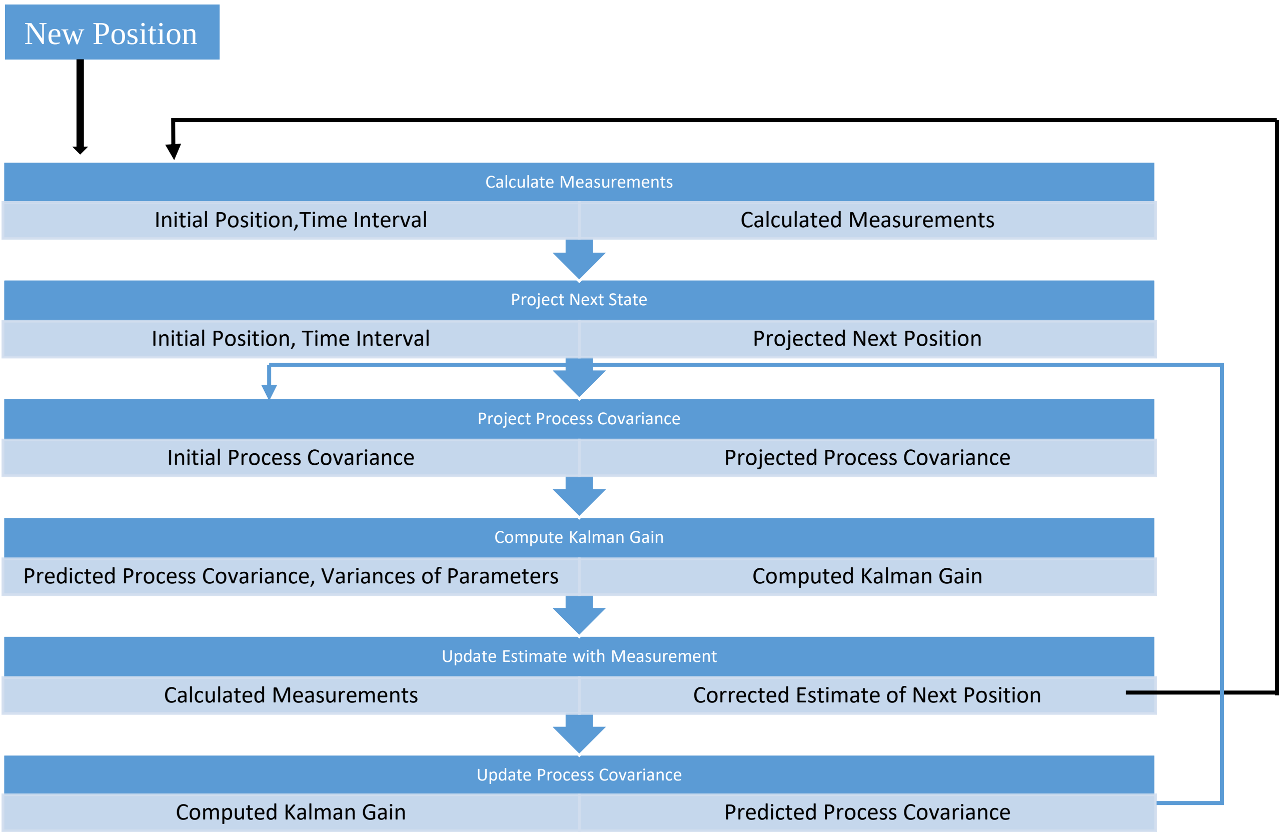

Figure 4.3 - Process with Inputs and Outputs

4.3.4 Distance to Geocoordinates Conversion

After each iteration of the recursion a position value is returned as two distance values (x_t, y_t). These distance values have to be converted into latitude and longitude values in order to compare them and visualize them to conduct the analysis and evaluation.

These distance values contain distances from (0,0) point of earth. By using the haversine formula, distances can be converted back in to the latitudes and longitudes.

Implementation of this process is as follows (Figure 4.4).

```java
public static double getCoordinateValues(double distance, boolean isLat) {
    double lat1 = Math.toRadians(0);
    double lon1 = Math.toRadians(0);
    double c = distance / 6371000;
    if (isLat) {
        double lon2 = Math.toRadians(0);
        double lat2 = Math.asin(Math.sin(lat1) * Math.cos(c) + Math.cos(lat1) * Math.sin(c) * Math.cos(Math.toRadians(0)));
        lat2 = Math.toDegrees(lat2);
        return lat2;
    } else {
        double lat2 = Math.toRadians(0);
        double lon2 = lon1 + Math.atan2(Math.sin(Math.toRadians(90)) * Math.sin(c) * Math.cos(lat1), Math.cos(c) - Math.sin(lat1) * Math.sin(lat2));
        lon2 = Math.toDegrees(lon2);
        lon2 = (lon2 + 540) % 360 - 180;
        return lon2;
    }

}
```

Figure 4.4 - Implementation of Distance to Coordinate Conversion

4.4 Summary

In this chapter, at first, the data parameter selection and pre-processing of data for the algorithm was discussed. Then discussion moved on to the Kalman filter algorithm and the process of the algorithm while depicting how this research problem will be implemented with the Kalman filter algorithm and selection of vectors and matrices for the process. At the end of this chapter, it was discussed how the output of the solution should be processed in order to analyse and evaluate. In next chapter, the evaluation and analysis will be discussed.

Chapter 5 Evaluation and Results

This chapter will first elaborate on the test cases, respective test results with an analysis on the errors. After that an evaluation on the solution will be discussed.

5.1 Test Data Collection

Test data were gathered from Siyara VTMS of UCSC as it has a large historical database of AIS data since few years. Few test cases were selected from the database which are available for a continuous period of time. These data had to be pre-processed and values for latitude, longitude, SOG, COG and time difference between the location points had to be extracted in order to do the prediction process.

As mentioned in earlier chapter, longitude and latitude values will be converted into meter distances while SOG values will be converted from knots to meters per second. After the process output distances and velocities will be converted back to latitude, longitude and knots.

5.2 Test Results for Initial Solution

As the initial solution, a simple conventional Kalman Filter based solution had been implemented. For the measurement update phase of the solution, next true values were used as measurement values in order to implement the initial solution. A sample data set had been tested with that solution and been compared with the true values.

5.2.1 Comparison

<u>Longitude</u>

Figure 5.1 shows the comparison of predicted and actual values of longitude values with time.

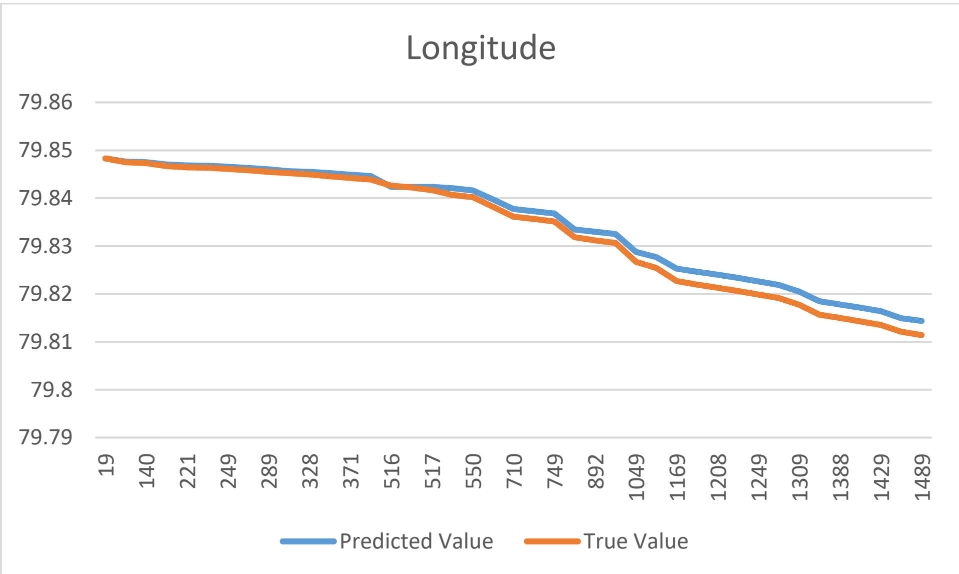

Figure 5.1- Comparison of Longitude with Time for Solution 1

Latitude

Figure 5.2 shows the comparison of predicted and actual values of latitude values with time.

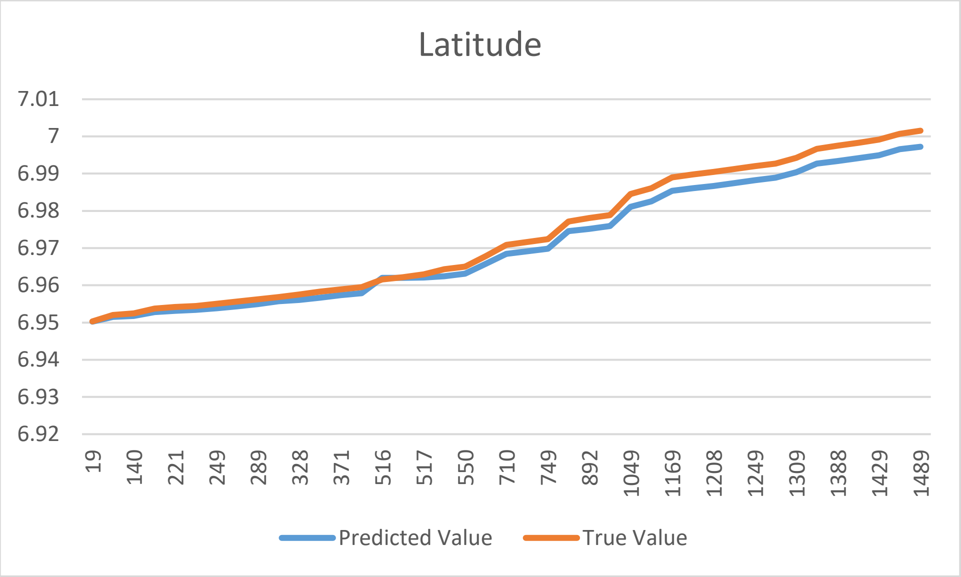

Figure 5.2 - Comparison of Latitude with Time for Solution 1

Speed on X Direction

Figure 5.3 shows the comparison of predicted and actual values of X direction speed values with time.

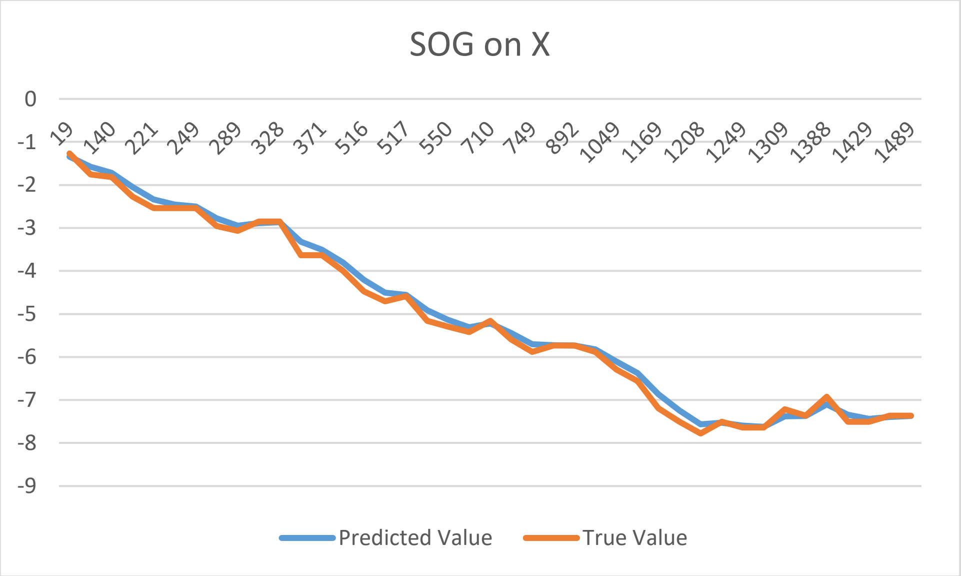

Figure 5.3 - Comparison of Speed on X Direction with Time for Solution 1

Speed on Y Direction

Figure 5.4 shows the comparison of predicted and actual values of Y direction speed values with time.

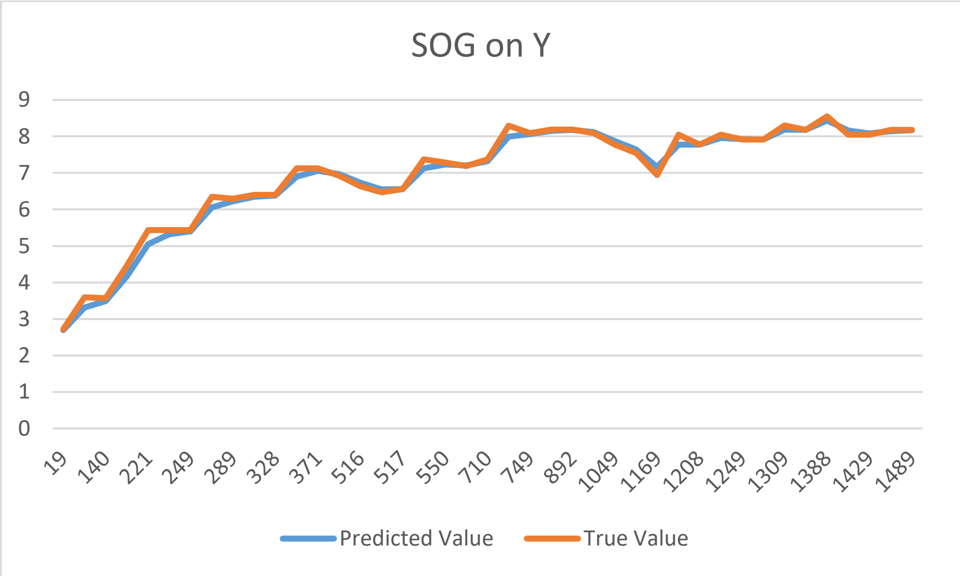

Figure 5.4 - Comparison of Speed on Y Direction with Time for Solution 1

Even though it gives good results for all four parameters on most occasions, there are few deviations. Thus, the solution had to be optimized further.

5.2.2 Test Results for Second Solution

As an optimization in the second solution, when the error covariance matrix is calculated for the prediction, only the current value and previous two values were taken in to account instead of all the values.

5.2.3 Comparison

<u>Longitude</u>

Figure 5.5 shows the comparison of predicted and actual values of longitude values with time.

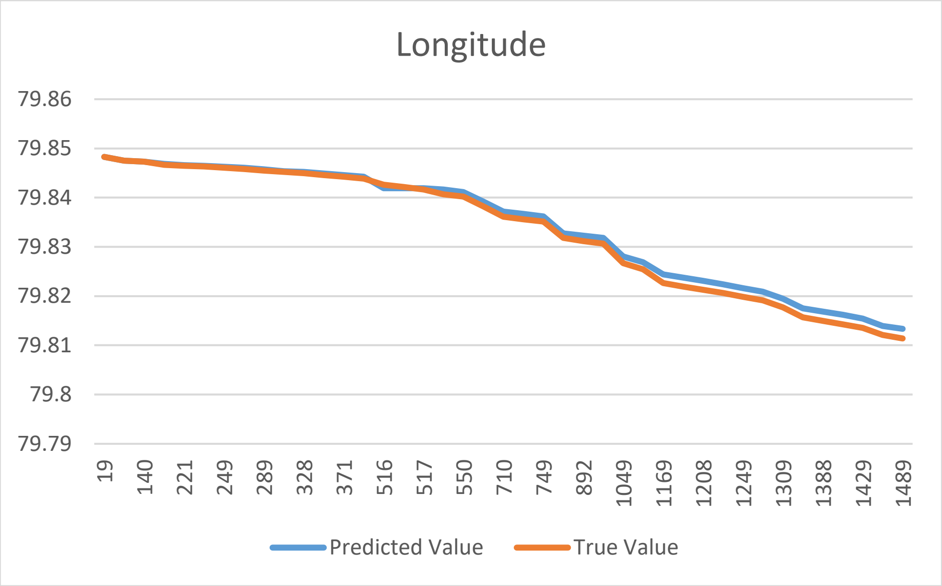

Figure 5.5 - Longitude Comparison for Solution 2

<u>Latitude</u>

Figure 5.6 shows the comparison of predicted and actual values of latitude values with time.

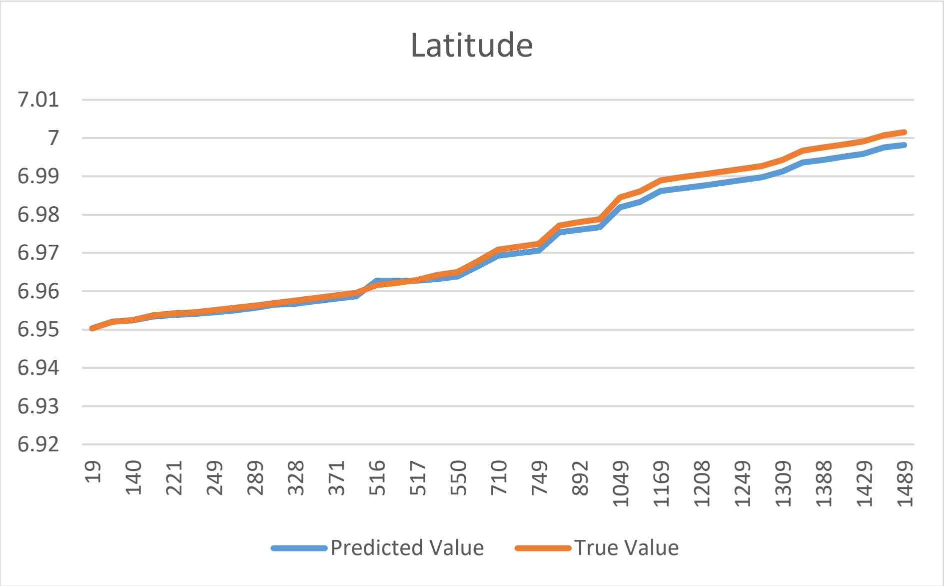

Figure 5.6 - Latitude Comparison for Solution 2

Speed on X Direction

Figure 5.7 shows the comparison of predicted and actual values of X direction speed values with time.

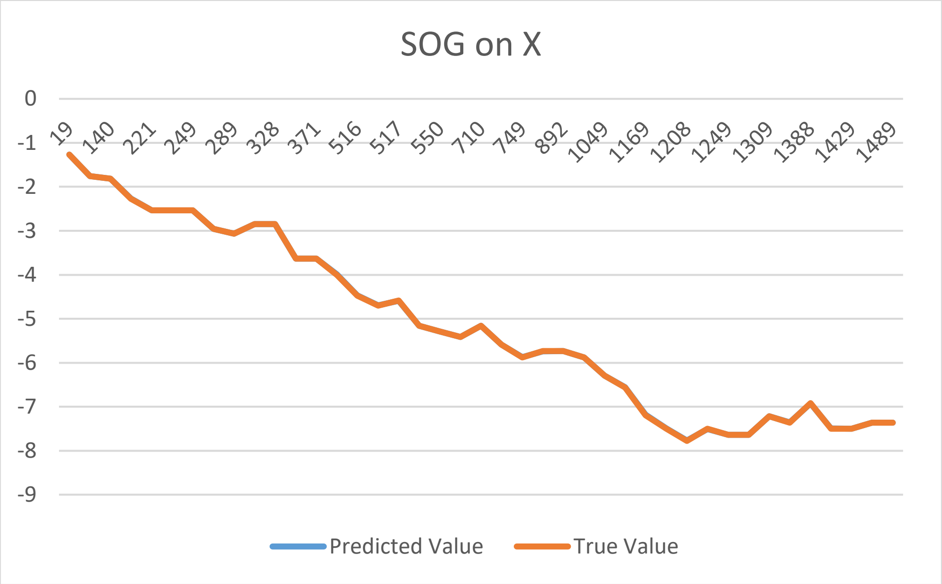

Figure 5.7 - Speed on X Direction Comparison for Solution 2

Speed on Y Direction

Figure 5.8 shows the comparison of predicted and actual values of Y direction speed values with time.

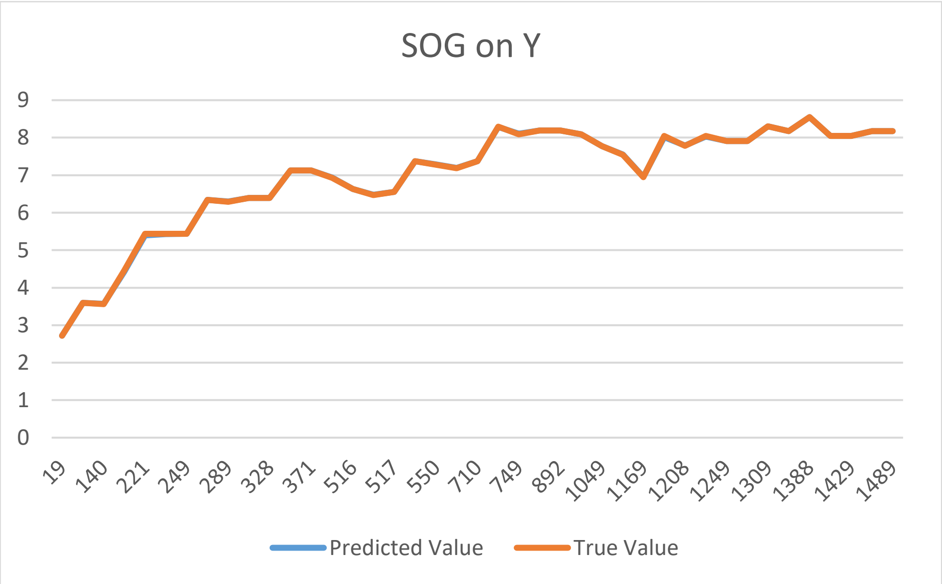

Figure 5.8 - Speed on Y Direction Comparison for Solution 2

These results show that the errors in speed prediction values were not very significant while there were small errors in the last few coordinate value predictions. Thus, this 2nd initial solution will be the base for the final solution.

5.3 Test Results for Final Solution

Using the 2nd method of the initial solution, the real-time location prediction solution was implemented. In this solution, measurements for the measurement update phase was calculated by the solution. Prediction was done every one second time interval until a new position value from the data set was being taken as an input when its time value is reached by the algorithm.

5.3.1 First Test Case

First test case is conducted on ship named Glovis Passion and has the MMSI number 311048200 on 2017-03-07 from 1701h to 1607h. Test results for this test case as follows.

First the test will compare predicted and actual longitude values.

Longitude

Figure 5.9 shows the predicted and actual values for the longitudes for test case 1 aligns with each other.

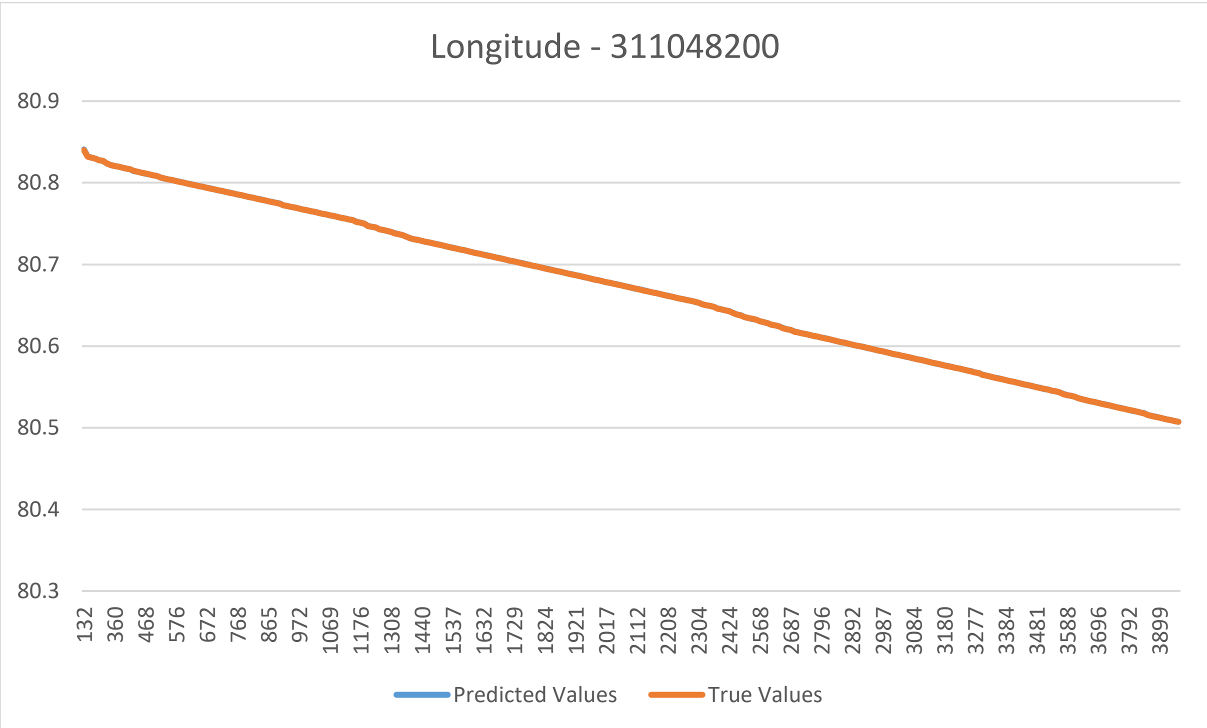

Figure 5.9 - Longitude Comparison for Test Case 1

Next comparison will be done on distances on X direction.

Distance on X Direction

Figure 5.10 elaborates the predicted and actual distances on the X direction overlaps with each other.

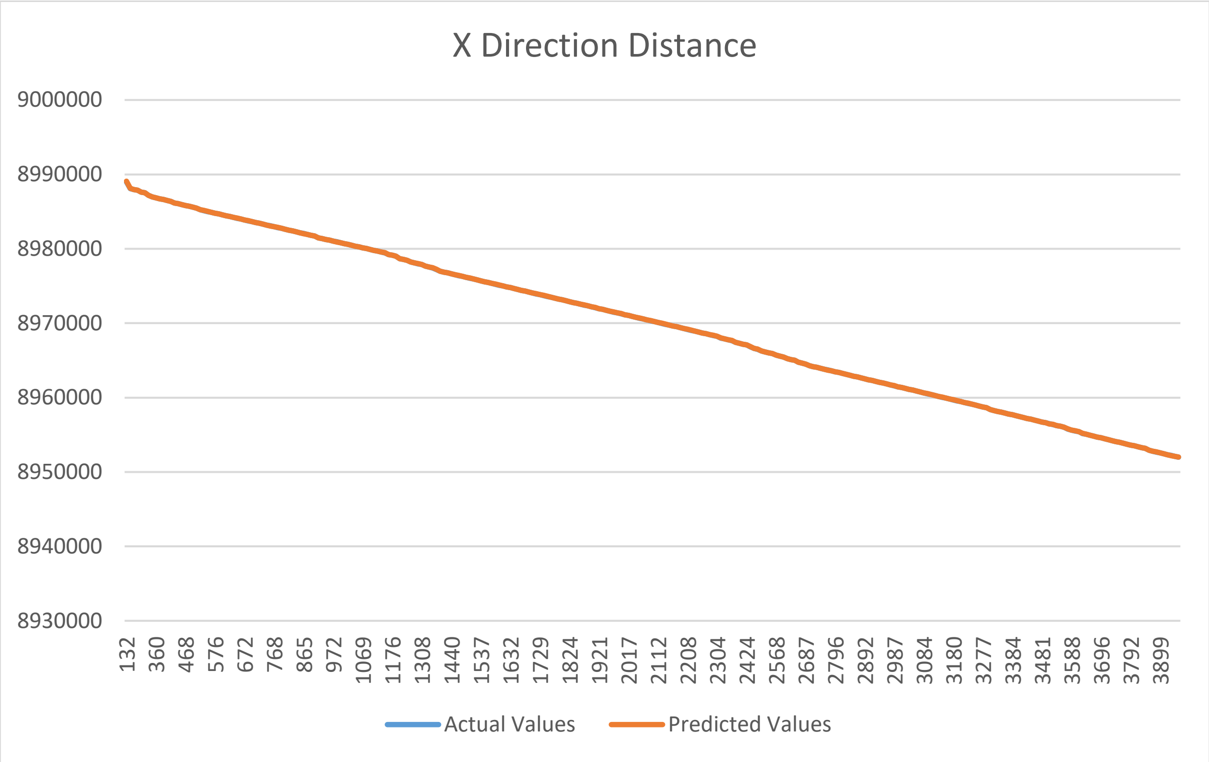

Figure 5.10 - X Direction Distance Comparison for Test Case 1

Furthermore, there were total of 286 data points to be predicted. Out of that 218 data points were with an error less than 10 meters while only 68 data points gave errors greater than or equal to 10 meters.

Next, it is needed to compare the latitudes.

Latitude

Figure 5.11 showed same kind of results as were for longitude comparison. The actual values and the predicted values aligned on each other.

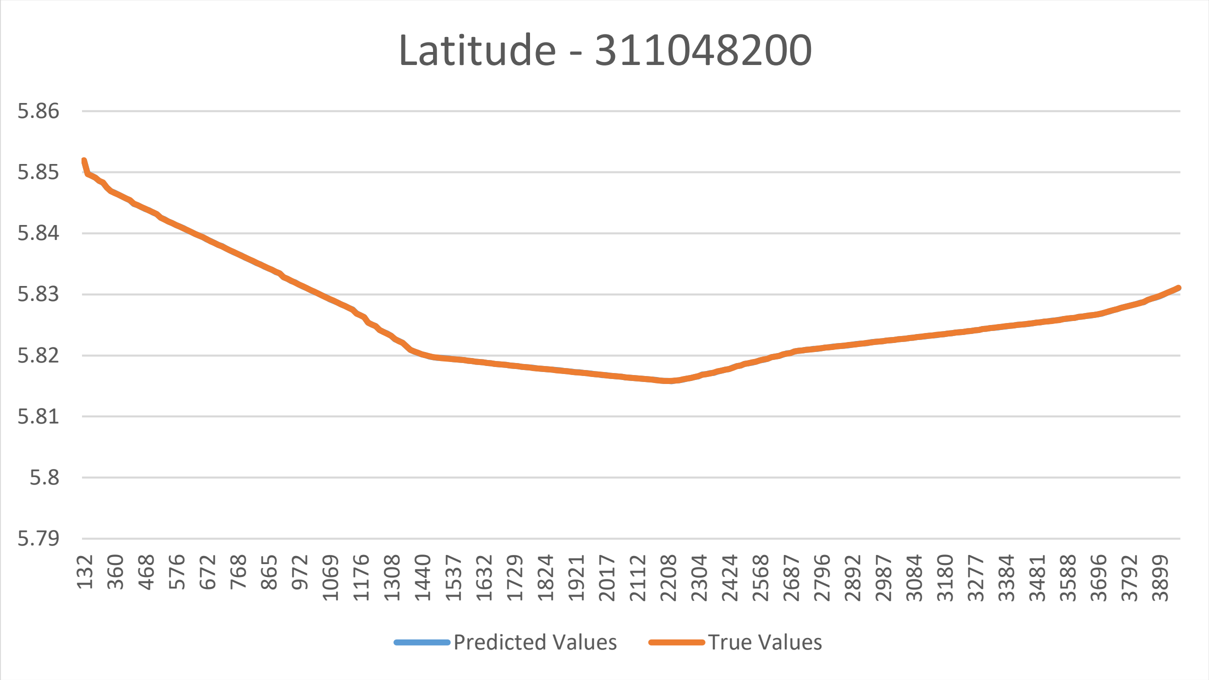

Figure 5.11 - Latitude Comparison for Test Case 1

Next comparison will be done on Y direction distances.

Distance on Y Direction

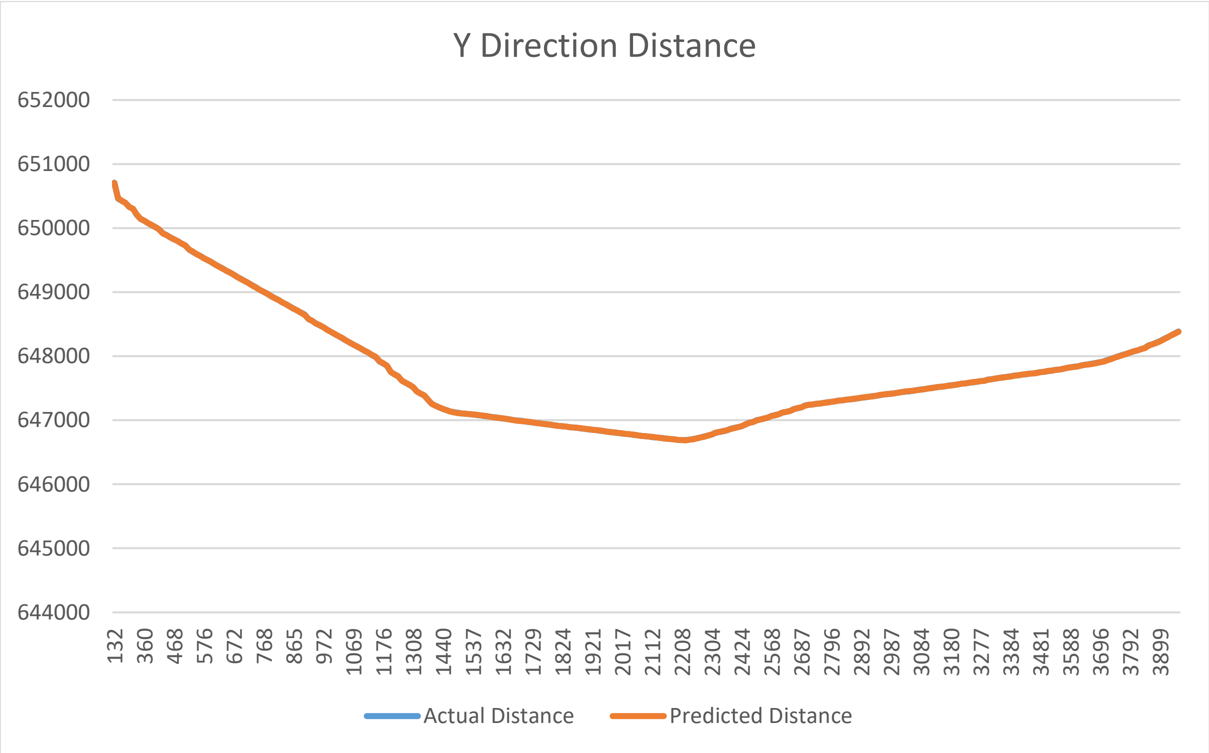

Figure 5.12 - Y Direction Distance Comparison for Test Case 1

This comparison (Figure 5.12) provided further conclusive evidence that the predictions on the Y direction will overlap with the actual values. Furthermore, there were 285 data points, out of total 286 data points were with an error less than 10 meters while the remaining data point gave an error of 12 meters. Out of those 285 data points 280 data points gave an error less than 5 meters.

5.3.2 Test Results for Test Case 2

First test case is conducted on ship named Red Zed 1and has the MMSI number 306095000 on 2017-03-07 from 1701h to 1955h. In this case, there is 22-minute gap between 1st and 2nd data point. Test results for this test case as follows.

First the comparison will be done on longitude values.

Longitude

Figure 5.13 shows that apart from the first two points, where there is 22-minute gap, the actual and predicted values align with each other.

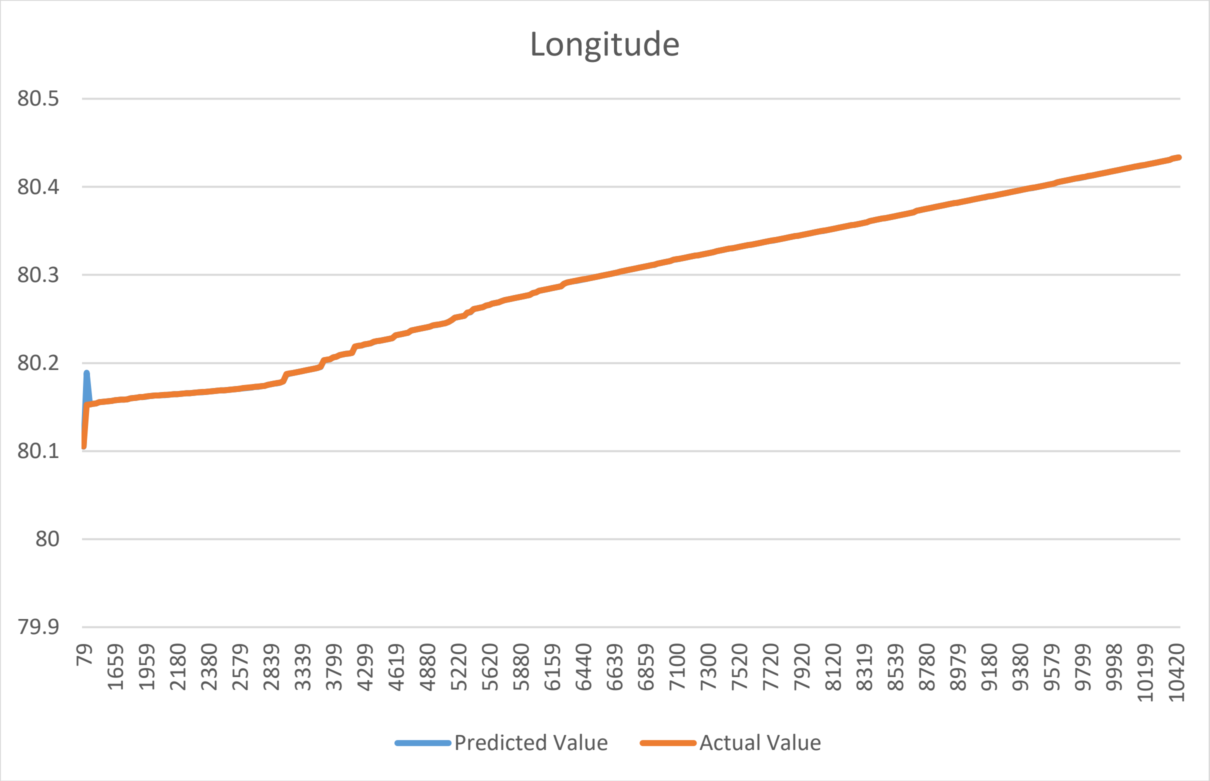

Figure 5.13 - Longitude Comparison for Test Case 2

Next comparison will be done on X direction distances.

Distance on X Direction

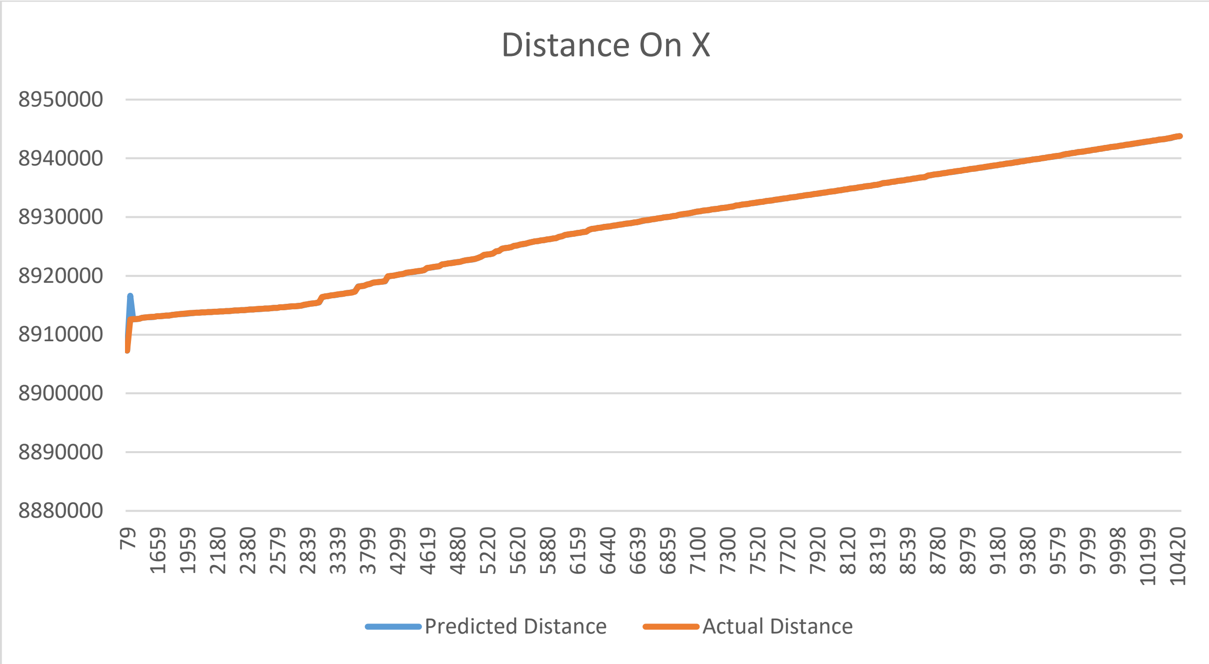

Figure 5.14 - X Direction Distance Comparison for Test Case 2

Figure 5.14 showed similar characteristics to longitude comparison chart. Apart from the first two data points all the predicted and actual values overlap with each other. There was a total of 352 data points to predict. Out of those 352 data points, there were 326 data points with an error less than 10 meters while only 26 data points with an error greater than or equal to 10 meters were present.

As the next step, the comparison was done on latitudes.

Latitude

During the latitude comparison (Figure 5.15), it showed that apart from the first two points, all the other predicted and actual value pair were aligned with each other.

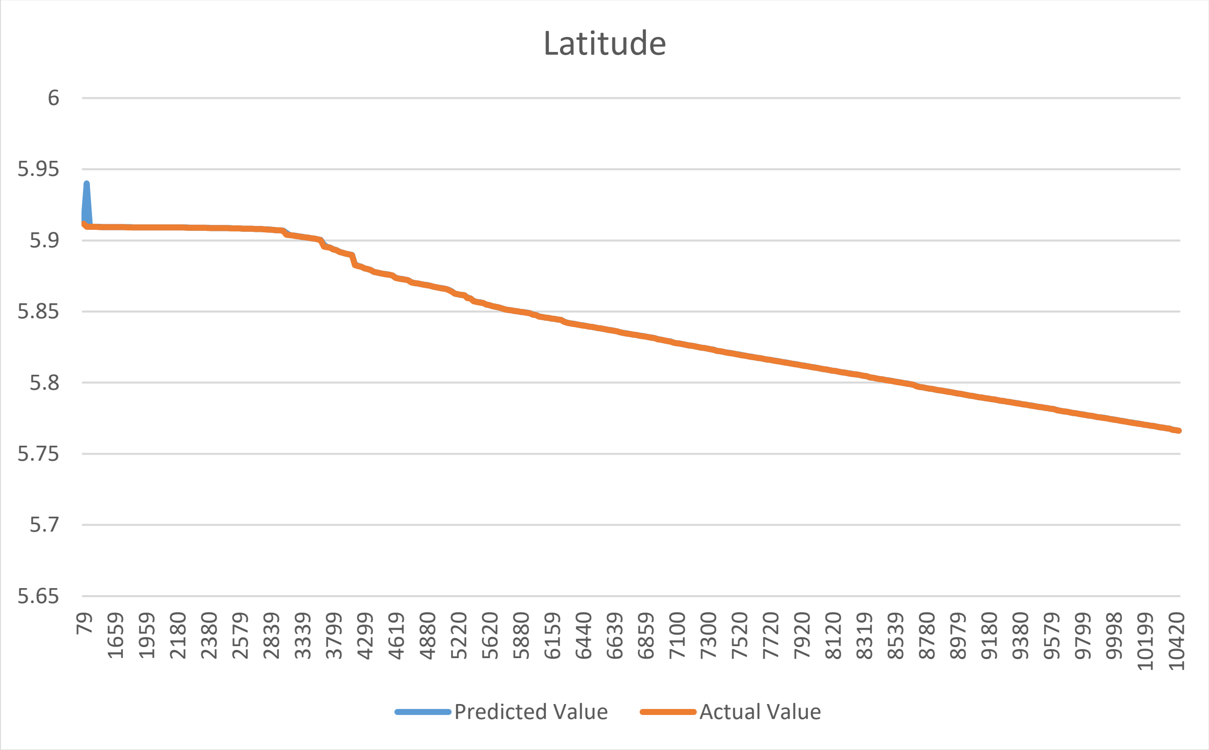

Figure 5.15 - Latitude Comparison for Test Case 2

In order to confirm this, a comparison with the Y direction distances was considered next.

<u>Distance on Y Direction</u>

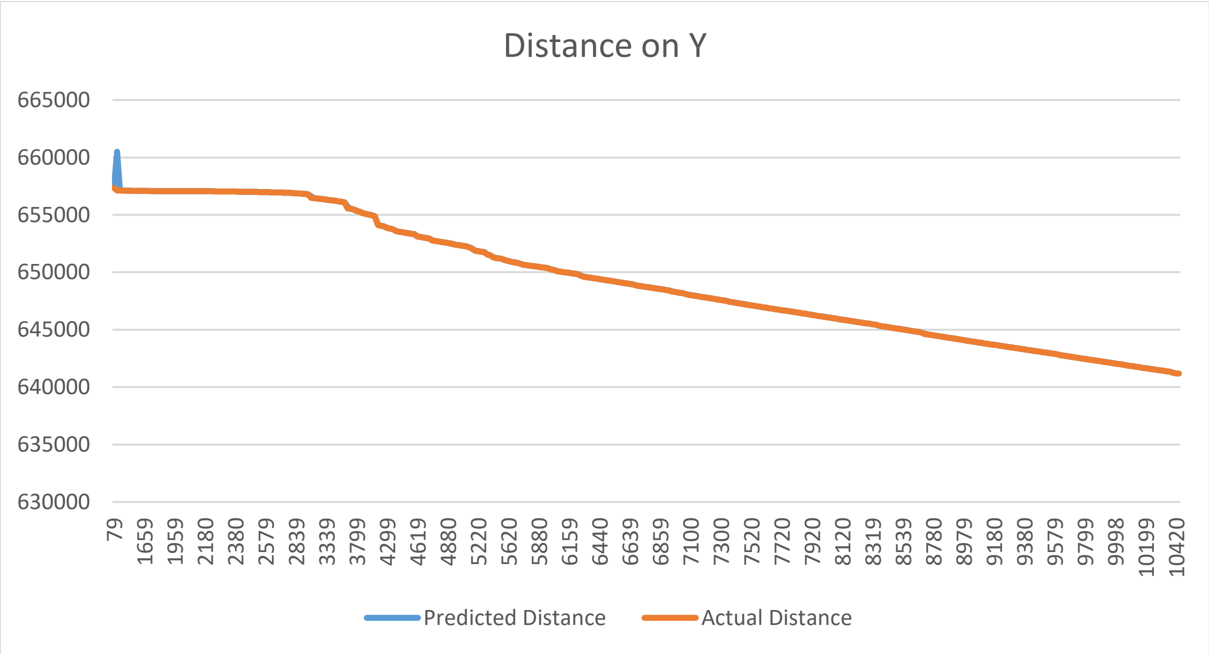

Figure 5.16 - Y Direction Distance Comparison for Test Case 2

Figure 5.16 showed similar results compared to latitude comparison. Apart from the first two data points, all the predicted and actual values overlap with each other. Out of 352 data points there were 329 data points with an error less than 10 meters while only 23 data points had an error greater than or equal to 10 meters.

5.4 Analysis

First, the analysing will be done on the positions to look if there are any deviations. Then the analysis will move on to runtime of the solution.

5.4.1 Analysis of Positions

In order to analyse and compare actual and predicted positions, first it is needed to look and have a better idea on the following two charts which were drawn for the two test cases.

In first test case, the predicted and actual value curves align and overlap with each other.

For the second test case, there is a difference around two points even though the rest of the data points align and overlap with each other. Reason for the difference at the beginning is that there is time gap of 22 minutes. For that kind of a longer prediction without any AIS value, there should be a few measurement vectors with at least 3-5 minutes of time interval since for a 22 minutes a vessel can be faced with many navigational changes like inverting its course. With this solution, it can be predicted maybe up to 5 minutes as during these two-test case solutions has already tackled few cases where prediction had to be carried out for more than 4 minutes.

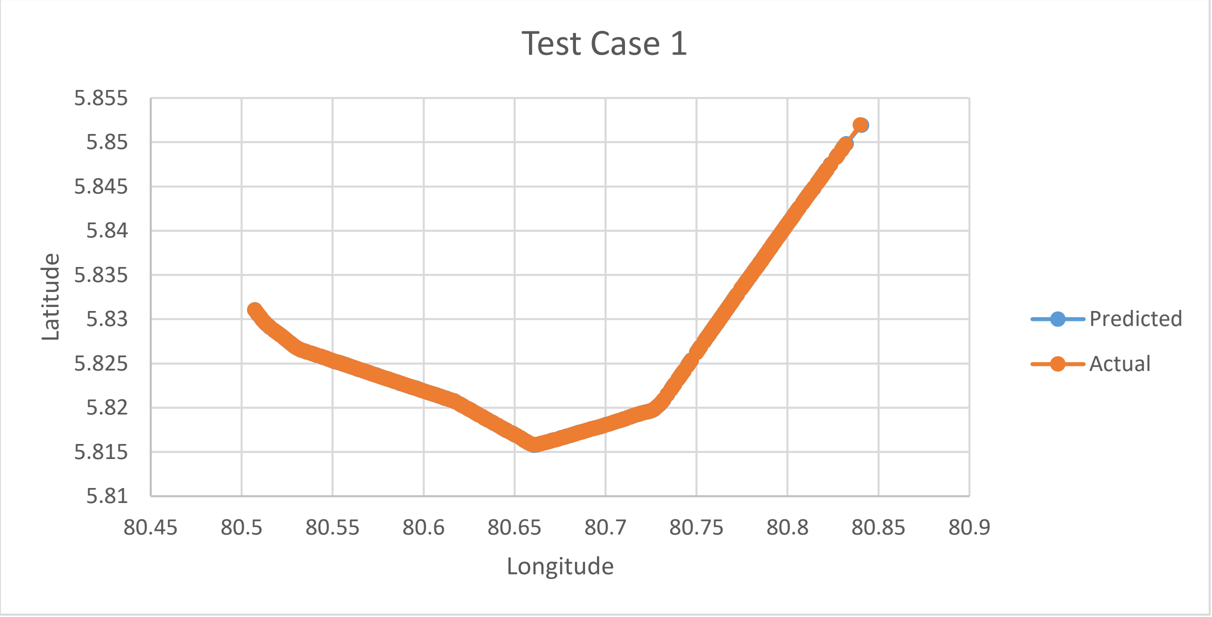

Figure 5.17 - Position Comparison for Test Case 1

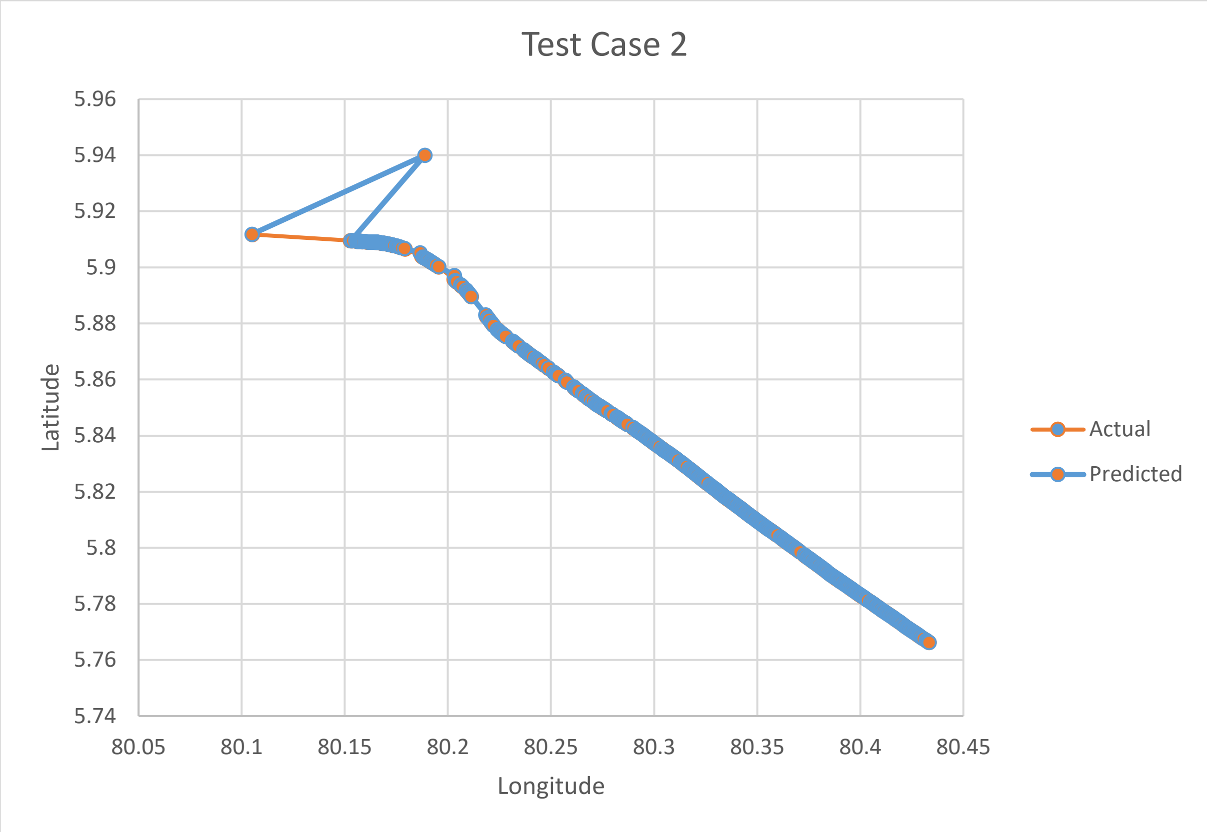

Figure 5.18 - Position Comparison for Test Case 2

After analysing these two charts, it can be concluded that apart from the two data points where there is a gap of 22 minutes, solution provided fair results for position predictions. Thus, it is safe to say that there is conclusive evidence that this solution provides correct predictions for cases which are not very abnormal.

5.4.2 Analysis on Runtime of the Solution

For the first test case, it was needed to carry out the prediction for 3959 times as the prediction continued for 3959 seconds and the prediction's time interval was set as one second. For the 3959 predictions, the system calculated while recursively running the Kalman filter algorithm, solution took approximately 300 milliseconds. That's $\frac{300}{3959}$ milliseconds per prediction which is approximately 0.075milliseconds (75 micro seconds).

For the second test case, the prediction process had to be carried out for 10439 times as the prediction continued for 10439 seconds. In this case, it took 900 milliseconds. Which means $\frac{900}{10439}$milliseconds per prediction. Which is roughly 0.086 milliseconds (86 micro seconds).

In order to fulfil the objective of this research, solution should be able to predict at least 20 times per second which is 50 milliseconds per prediction as it would make the movements more realistic. 50 milliseconds are much higher than what this solution will take to do even a few predictions on a somewhat low-end computer even though the test cases were tested on a high-end computer. Thus, it can be concluded that this system is capable of providing a solution to the objective of this research.

5.5 Summary

This chapter provided test results for both of the initial solutions and the comparison of those solutions outputs at first. Then the discussion moved on to the final solutions test cases and their output comparisons. After that the discussion moved on to analysis of the position outputs and the running time analysis. In the next chapter, discussion will elaborate on the conclusion of the research project and the future work that can be carried out on this research project.

Chapter 6 Conclusion and Future Work

This chapter will discuss on the conclusion of this research and future work that need to be carried out with this research.

6.1 Conclusion

According to the analysis done on the previous chapter and experience had during the research project following conclusions can be made.

- The solution is capable of providing correct position outputs for almost all the cases where cases are not abnormal.
- Prediction can be carried on for about few minutes without an AIS update if there are no abnormal behaviours for ships even though the objective was to predict the next location point.
- Solution is capable providing predictions within few milliseconds which is more than enough to satisfy the objective of this research.
- If there is another sensor (other than the GPS sensor coupled with AIS) on the vessel which can provide the measurements, if the error of those measurements is known, then the predictions will be more accurate since in the current solution, calculated measurements are used.
- If there is a way to get the acceleration of the vessel, then the solution will be able to produce results with higher accuracy. Since AIS does not send acceleration, it has to be calculated.

6.2 Future Work

As the first task of the future work on this research project should be to plug this solution into Siyara VTMS. Next step should be to look new ways to improve the accuracy of the solution even though it is very accurate at present it would be better if it can be made further accurate. Next task should be improving the efficiency even though the efficiency of this system is more than enough for the task it wouldn’t hurt if it can be made further efficient.

After that, the next most important task should be to implement a particle filter based solution as there are some evidence that it will provide better accuracy than a particle filter based solution on some occasions. Thus, it is necessary to check whether if a particle filter based solution would provide better accuracy even though it may be less efficient than the current particle filter based solution. At the start of this research, there was an intention of completing a solution with a particle filter based approach even though due to time related issues it could not be fulfilled.

Then there are other possibilities like having a solution with both the Kalman filter and Particle filter. This Kalman filter and particle filter fusion solution must be able to predict position using both the algorithms and should provide a single value after corrections applied to both outputs from the two algorithms.

Then there should be a location predicter for VTMS where non-linear data are present. For that case, an extended Kalman Filter can be used even though there are other possibilities.

References

[1] Y. Xiao, H. Zhang, and S. Li, “Dynamic Data Driven Multi-agent Simulation in Maritime Traffic,” in *2009 International Conference on Computer and Automation Engineering*, 2009, vol. 0, pp. 234–237.

[2] “Vessel traffic service.” [Online]. Available: http://en.wikipedia.org/wiki/Vessel_traffic_service. [Accessed: 20-Apr-2016].

[3] W. Ren, G. Li, and J. Lv, “Analysis of Adopting Mobile Position System Based on GPSOne Technique in VTS System,” in *2010 International Conference on Intelligent System Design and Engineering Application*, 2010, vol. 1, pp. 281–284.

[4] A. Karbassi and M. Barth, “Vehicle route prediction and time of arrival estimation techniques for improved transportation system management,” *IEEE IV2003 Intell. Veh. Symp. Proc. (Cat. No.03TH8683)*, pp. 511–516, 2003.

[5] C. Hu, W. Chen, Y. Chen, and D. Liu, “Adaptive Kalman Filtering for Vehicle Navigation,” *Positioning*, vol. 1, no. 4, p. 0, 2009.

[6] F. Mazzarella, V. F. Arguedas, and M. Vespe, “Knowledge-based vessel position prediction using historical AIS data,” *2015 Sens. Data Fusion Trends, Solut. Appl.*, pp. 1–6, 2015.

[7] L. P. Perera and C. G. Soares, “Ocean Vessel Trajectory Estimation and Prediction Based on Extended Kalman Filter,” *Second Int. Conf. Adapt. Self-Adaptive Syst. Appl.*, no. c, pp. 14–20, 2010.

[8] L. P. Perera, P. Oliveira, and C. Guedes Soares, “Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1188–1200, 2012.

[9] “Automatic Identification System.” [Online]. Available: http://en.wikipedia.org/wiki/Automatic_Identification_System. [Accessed: 20-Apr-2016].

[10] N. Sawano, “Current Situation of Digitalized Ship Navigation System for Safety,” in *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)*, 2008, vol. 0, pp. 1134–1137.

[11] Y. Shijun, L. Ying, and X. Peng, “Application of AIS in Marine Search and Rescue Radar,” in *2010 Second WRI Global Congress on Intelligent Systems*, 2010, vol. 2, pp.

65–68.

[12] K. Hata, K. Hasegawa, K. Niwa, and H. Fukuda, “AIS Simulator and ITS Applications,” in *Proceedings ELMAR 2006*, 2006, vol. 0, no. June, pp. 223–226.

[13] N. Willems, “Visualization of Vessel Traffic,” Technische Universiteit Eindhoven, 2011.

[14] G. Pallotta, M. Vespe, and K. Bryan, “Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction,” *Entropy*, vol. 15, no. 6, pp. 2218–2245, 2013.

[15] R. E. Kalman and others, “A new approach to linear filtering and prediction problems,” *J. basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[16] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” *In Pract.*, vol. 7, no. 1, pp. 1–16, 2006.

[17] L. Kleeman, “Understanding and applying Kalman filtering,” *Proc. Second Work. Perceptive …*, p. 17, 1996.

[18] “Map Projection.” [Online]. Available: https://en.wikipedia.org/wiki/Map_projection. [Accessed: 17-Nov-2016].

[19] “What is a Map Projection?” [Online]. Available: https://www.gislounge.com/map-projection/. [Accessed: 17-Nov-2016].

[20] “Haversine formula.” [Online]. Available: https://en.wikipedia.org/wiki/Haversine_formula. [Accessed: 20-Nov-2016].

[21] “Haversine Formula – Calculate geographic distance on earth.” [Online]. Available: http://www.igismap.com/haversine-formula-calculate-geographic-distance-earth/. [Accessed: 20-Nov-2016].

[22] R. Faragher, “Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes],” *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 128–132, 2012.

[23] “Versine.” [Online]. Available: https://en.wikipedia.org/wiki/Versine. [Accessed: 20-Nov-2016].

[24] “CLASS A AIS POSITION REPORT (MESSAGES 1, 2, AND 3).” [Online]. Available: https://www.navcen.uscg.gov/?pageName=AISMessagesA. [Accessed: 23-Dec-2016].

Appendices

Appendix A Source Code for Implementation of Kalman Filter

Project State Ahead

```
public static Matrix findXkp(double dt, double x, double v, double a) {
    Matrix A = new Matrix(new double[]{1, 0, dt, 1}, 2);
    Matrix Xkminus1 = new Matrix(new double[]{x, v}, 2);
    Matrix B = new Matrix(new double[]{((dt * dt) / 2), dt}, 2);
    Matrix uk = new Matrix(new double[]{a}, 1);//
    return A.times(Xkminus1).plus(B.times(uk));
}
```

Figure 0.1 - Source Code to Project Next State

Project Process Covariance

```
//Predicted Process Covariance
public static Matrix findPPC(double dt, Matrix m) {
    Matrix A = new Matrix(new double[]{1, 0, dt, 1}, 2);
    Matrix Temp = A.times(m);
    Matrix AT = A.transpose();
    Matrix Q = Matrix.identity(2, 2);
    return Temp.times(AT).plus(Q);
}
```

Figure 0.2 - Source Code for Project Process COvariance

Calculate Kalman Gain

```
public static Matrix findKalmanGain(double dt, Matrix PPC, double xSD, double dxSD) {
    Matrix H = Matrix.identity(2, 2);
    Matrix R = new Matrix(new double[]{xSD, 0, 0, dxSD}, 2);
    Matrix HT = H.transpose();
    Matrix Temp = H.times(PPC);
    Temp = Temp.times(HT);
    Temp = Temp.plus(R);
    Temp = Temp.inverse();
    Matrix K = PPC.times(HT);
    K = K.times(Temp);
    return K;
}
```

Figure 0.3 - Source Code for Calculate Kalman Gain

Calculate Measurement Residual

```
public static Matrix findYk(double x, double v) {
    Matrix C = Matrix.identity(2, 2);
    Matrix Ykm = new Matrix(new double[]{x, v}, 2);
    Matrix Vk = new Matrix(new double[]{0.1, 0.00001}, 2);//
    return C.times(Ykm).plus(Vk);
}
```

Figure 0.4 - Source Code for Calculate Measurement Residual

Update Next State

```
public static Matrix findXk(Matrix Xkp, Matrix K, Matrix Yk) {
    Matrix H = Matrix.identity(2, 2);
    Matrix Temp = H.times(Xkp);
    Temp = Yk.minus(Temp);
    Temp = K.times(Temp);
    return Xkp.plus(Temp);
}
```

Figure 0.5 - Source Code for Update Next State with Measurements

Update Process Covariance

```
//Update Process Covariance
public static Matrix updatePC(Matrix K, Matrix PPC) {
    Matrix H = Matrix.identity(2, 2);
    Matrix I = Matrix.identity(2, 2);
    Matrix Temp = K.times(H);
    Temp = I.minus(Temp);
    return Temp.times(PPC);
}
```

Figure 0.6 - Source Code for Update Process Covariance

Appendix B Few Output Values for Test Cases

Test Case 1

Longitude

Time	Predicted Value	True Value	Error	Error Percentage
132	80.84078438	80.8399617	0.000822685	0.0010176711%
228	80.83230738	80.831805	0.000502375	0.0006215072%
240	80.83079435	80.8307867	7.6501E-06	0.0000094643%
252	80.82977605	80.8297633	1.27501E-05	0.0000157740%
276	80.82774031	80.8277217	1.86135E-05	0.0000230286%
288	80.82676447	80.8267817	1.72339E-05	0.0000213220%
324	80.82373211	80.8237117	2.04058E-05	0.0000252473%
348	80.82168871	80.82166	2.87135E-05	0.0000355270%
360	80.82070277	80.8206367	6.60661E-05	0.0000817441%

372	80.81963128	80.8196133	1.79834E-05	0.0000222513%
384	80.81860788	80.8185083	9.95834E-05	0.0001232186%
396	80.81750288	80.8174883	1.45834E-05	0.0000180449%
408	80.81648288	80.8164633	1.95834E-05	0.0000242319%
432	80.81445078	80.814495	4.42173E-05	0.0000547145%
444	80.81348435	80.81347	1.43501E-05	0.0000177571%
456	80.81245935	80.812355	0.00010435	0.0001291264%
468	80.81134435	80.8114133	6.89499E-05	0.0000853220%
479	80.81048704	80.810385	0.00010204	0.0001262708%
492	80.80928997	80.8093617	7.17338E-05	0.0000887692%
504	80.80835628	80.8083367	1.95834E-05	0.0000242344%
528	80.80631371	80.8063717	5.79865E-05	0.0000717598%
539	80.80545024	80.8052633	0.000186937	0.0002313424%
552	80.80417394	80.8041517	2.22359E-05	0.0000275182%
564	80.80314105	80.8032133	7.22499E-05	0.0000894146%
576	80.80220788	80.8021867	2.11834E-05	0.0000262164%
588	80.80118128	80.8011617	1.95834E-05	0.0000242365%
599	80.80023544	80.8001333	0.00010214	0.0001264106%
612	80.79904394	80.7991067	6.27641E-05	0.0000776792%
624	80.79809605	80.7979917	0.00010435	0.0001291494%
636	80.79698628	80.7970483	6.20166E-05	0.0000767560%
648	80.79604288	80.7960217	2.11834E-05	0.0000262184%
659	80.79509544	80.79508	1.544E-05	0.0000191100%
672	80.79399064	80.79397	2.06359E-05	0.0000255414%
684	80.79296458	80.79286	0.000104583	0.0001294463%
696	80.79184935	80.7919183	6.89499E-05	0.0000853426%
708	80.79091288	80.790895	1.78834E-05	0.0000221354%
719	80.78996874	80.7898683	0.00010044	0.0001243225%
732	80.78877894	80.7888483	6.93641E-05	0.0000858585%
744	80.78784288	80.7877433	9.95834E-05	0.0001232655%
756	80.78673788	80.786805	6.71166E-05	0.0000830787%
768	80.78579958	80.7857833	1.62834E-05	0.0000201563%
779	80.78486184	80.78476	0.000101837	0.0001260595%
792	80.78367064	80.78373	5.93641E-05	0.0000734852%
804	80.78272458	80.7827067	1.78834E-05	0.0000221377%
816	80.78170128	80.7816783	2.29834E-05	0.0000284512%
829	80.78058894	80.7805667	2.22359E-05	0.0000275263%
840	80.77964044	80.7795417	9.874E-05	0.0001222339%
852	80.77853628	80.7785167	1.95834E-05	0.0000242433%
865	80.77742734	80.777485	5.76641E-05	0.0000713864%
876	80.77656354	80.776465	9.85369E-05	0.0001219871%
889	80.77537564	80.77544	6.43641E-05	0.0000796828%
900	80.77451854	80.774505	1.35369E-05	0.0000167588%
925	80.77240858	80.7723883	2.02801E-05	0.0000251077%
936	80.77146684	80.7714567	1.01369E-05	0.0000125501%
949	80.77036734	80.7703567	1.06359E-05	0.0000131681%
960	80.76948395	80.769425	5.89466E-05	0.0000729813%
972	80.76841958	80.76841	9.5834E-06	0.0000118653%
985	80.76732064	80.7673983	7.76641E-05	0.0000961578%
996	80.76648192	80.7662983	0.000183616	0.0002273420%
1009	80.76526651	80.7652833	1.67917E-05	0.0000207908%
1020	80.76436692	80.7643567	1.02157E-05	0.0000126488%
1032	80.76340442	80.763255	0.000149424	0.0001850154%
1045	80.76222321	80.7622433	2.00917E-05	0.0000248776%
1056	80.76132184	80.7612267	9.51369E-05	0.0001178002%
1069	80.76013734	80.7602167	7.93641E-05	0.0000982713%
1080	80.75929524	80.7592833	1.19369E-05	0.0000147808%
1092	80.75827788	80.7582733	4.5834E-06	0.0000056755%
1105	80.75724151	80.757175	6.65083E-05	0.0000823559%
1116	80.75630225	80.75625	5.22466E-05	0.0000646966%
1129	80.75521284	80.75515	6.28364E-05	0.0000778110%

Table B.1 - Longitude Data for Test Case 1

Latitude

Time	Predicted Value	True Value	Error	Error Percentage
132	5.851932344	5.85197333	4.09856E-05	0.00070%
228	5.849821953	5.84971333	0.000108623	0.00186%
240	5.849428034	5.84942667	1.36413E-06	0.00002%
252	5.849141374	5.84913833	3.04413E-06	0.00005%
276	5.848566626	5.84856833	1.70355E-06	0.00003%
288	5.84829848	5.84831	1.15201E-05	0.00020%
324	5.847506133	5.84746833	3.78034E-05	0.00065%
348	5.846896626	5.84691	1.33736E-05	0.00023%
360	5.84664015	5.84662167	1.84799E-05	0.00032%
372	5.846318571	5.84633167	1.3099E-05	0.00022%
384	5.846028571	5.84601333	1.5241E-05	0.00026%
396	5.845710231	5.84571667	6.43903E-06	0.00011%
408	5.845413571	5.84543	1.6429E-05	0.00028%
432	5.844822658	5.84488167	5.90116E-05	0.00101%
444	5.844596374	5.84459333	3.04413E-06	0.00005%
456	5.844308034	5.84428333	2.47041E-05	0.00042%
468	5.843998034	5.84402333	2.52959E-05	0.00043%
479	5.843761917	5.84373667	2.5247E-05	0.00043%
492	5.843427493	5.84344667	1.91766E-05	0.00033%
504	5.843143571	5.84315333	9.75903E-06	0.00017%
528	5.842581626	5.84259	8.37355E-06	0.00014%
539	5.84231227	5.84226833	4.39401E-05	0.00075%
552	5.841939864	5.84194833	8.46605E-06	0.00014%
564	5.841663034	5.841675	1.19659E-05	0.00020%
576	5.841371901	5.84138	8.09903E-06	0.00014%
588	5.841076901	5.84108667	9.76903E-06	0.00017%
599	5.840825257	5.84079333	3.1927E-05	0.00055%
612	5.840464864	5.84050333	3.8466E-05	0.00066%
624	5.840218034	5.840185	3.30341E-05	0.00057%
636	5.839881901	5.83991333	3.1429E-05	0.00054%
648	5.839610231	5.83962	9.76903E-06	0.00017%
659	5.839358587	5.83934667	1.1917E-05	0.00020%
672	5.839018204	5.83903167	1.3466E-05	0.00023%
684	5.838728571	5.83870667	2.1901E-05	0.00038%
696	5.838421374	5.838435	1.36259E-05	0.00023%
708	5.838131901	5.83813667	4.76903E-06	0.00008%
719	5.837875257	5.83784167	3.3587E-05	0.00058%
732	5.837513204	5.83754167	2.8466E-05	0.00049%
744	5.837238571	5.83721	2.8571E-05	0.00049%
756	5.836906901	5.83693667	2.9769E-05	0.00051%
768	5.836633571	5.83664	6.42903E-06	0.00011%
779	5.83636227	5.83634833	1.39401E-05	0.00024%
792	5.836019864	5.83605833	3.8466E-05	0.00066%
804	5.835755231	5.83576667	1.1439E-05	0.00020%
816	5.835463571	5.83547833	1.4759E-05	0.00025%
829	5.835149864	5.83516667	1.6806E-05	0.00029%
840	5.834905257	5.834875	3.0257E-05	0.00052%
852	5.834571901	5.83458833	1.6429E-05	0.00028%
865	5.834259864	5.8343	4.0136E-05	0.00069%
876	5.83402227	5.83400667	1.56001E-05	0.00027%
889	5.833678204	5.83371167	3.3466E-05	0.00057%
900	5.83343394	5.83344167	7.72986E-06	0.00013%
925	5.832808993	5.83281667	7.67689E-06	0.00013%
936	5.83253894	5.83254333	4.38986E-06	0.00008%
949	5.832214864	5.83221667	1.80605E-06	0.00003%
960	5.831953944	5.83194167	1.22738E-05	0.00021%
972	5.831638571	5.83163833	2.40967E-07	0.00000%
985	5.831309864	5.83134167	3.1806E-05	0.00055%
996	5.831047722	5.83102167	2.60518E-05	0.00045%
1009	5.830710941	5.83072667	1.57291E-05	0.00027%

1020	5.830432722	5.830455	2.22782E-05	0.00038%
1032	5.830168271	5.830135	3.32713E-05	0.00057%
1045	5.829824271	5.829835	1.07291E-05	0.00018%
1056	5.82955727	5.82954	1.72701E-05	0.00030%
1069	5.829211534	5.82924333	3.1796E-05	0.00055%
1080	5.8289656	5.828975	9.39986E-06	0.00016%
1092	5.828671901	5.82868	8.09903E-06	0.00014%
1105	5.828369271	5.82835833	1.09409E-05	0.00019%
1116	5.828095604	5.82808833	7.27377E-06	0.00012%
1129	5.827795889	5.82776667	2.92188E-05	0.00050%

Table B.2 - Latitude Data for Test Case 1

Distance on X Direction

Time	Actual Distance	Predicted Distance	Difference	<10
132	8988993.611	8989085.09	91.47839	FALSE
228	8988086.628	8988142.489	55.8616	FALSE
240	8987973.398	8987974.248	0.850652	TRUE
252	8987859.601	8987861.019	1.417746	TRUE
276	8987632.585	8987634.655	2.069724	TRUE
288	8987528.062	8987526.146	1.916325	TRUE
324	8987186.694	8987188.963	2.269025	TRUE
348	8986958.555	8986961.748	3.192793	TRUE
360	8986844.769	8986852.115	7.346212	TRUE
372	8986730.972	8986732.972	1.999662	TRUE
384	8986608.102	8986619.175	11.07317	FALSE
396	8986494.683	8986496.305	1.6216	TRUE
408	8986380.708	8986382.886	2.177574	TRUE
432	8986161.843	8986156.927	4.916736	TRUE
444	8986047.869	8986049.464	1.595658	TRUE
456	8985923.886	8985935.489	11.6032	FALSE
468	8985819.174	8985811.507	7.666879	TRUE
479	8985704.832	8985716.179	11.34633	FALSE
492	8985591.046	8985583.07	7.976434	TRUE
504	8985477.072	8985479.249	2.177574	TRUE
528	8985258.574	8985252.126	6.447807	TRUE
539	8985135.325	8985156.112	20.78643	FALSE
552	8985011.721	8985014.193	2.472518	TRUE
564	8984907.376	8984899.342	8.033822	TRUE
576	8984793.223	8984795.578	2.355486	TRUE
588	8984679.248	8984681.426	2.177574	TRUE
599	8984564.895	8984576.253	11.35745	FALSE
612	8984450.742	8984443.763	6.979051	TRUE
624	8984326.76	8984338.363	11.6032	FALSE
636	8984221.859	8984214.963	6.895932	TRUE
648	8984107.706	8984110.062	2.355486	TRUE
659	8984002.994	8984004.711	1.716845	TRUE
672	8983879.567	8983881.862	2.294606	TRUE
684	8983756.141	8983767.77	11.62914	FALSE
696	8983651.429	8983643.762	7.666879	TRUE
708	8983537.643	8983539.632	1.988543	TRUE
719	8983423.479	8983434.648	11.16841	FALSE
732	8983310.06	8983302.347	7.712938	TRUE
744	8983187.19	8983198.263	11.07317	FALSE
756	8983082.856	8983075.393	7.463026	TRUE
768	8982969.248	8982971.059	1.810631	TRUE
779	8982855.462	8982866.786	11.32374	FALSE
792	8982740.931	8982734.33	6.600988	TRUE

804	8982627.146	8982629.134	1.988543	TRUE
816	8982512.793	8982515.348	2.555637	TRUE
829	8982389.189	8982391.661	2.472518	TRUE
840	8982275.214	8982286.193	10.97938	FALSE
852	8982161.239	8982163.416	2.177574	TRUE
865	8982046.519	8982040.107	6.411957	TRUE
876	8981933.1	8981944.057	10.9568	FALSE
889	8981819.125	8981811.969	7.156963	TRUE
900	8981715.158	8981716.663	1.505231	TRUE
925	8981479.792	8981482.047	2.255045	TRUE
936	8981376.203	8981377.33	1.127168	TRUE
949	8981253.888	8981255.071	1.182657	TRUE
960	8981150.288	8981156.843	6.554561	TRUE
972	8981037.425	8981038.491	1.065625	TRUE
985	8980924.929	8980916.293	8.635855	TRUE
996	8980802.615	8980823.032	20.41714	FALSE
1009	8980689.752	8980687.885	1.867153	TRUE
1020	8980586.719	8980587.855	1.135936	TRUE
1032	8980464.215	8980480.831	16.61525	FALSE
1045	8980351.719	8980349.485	2.234096	TRUE
1056	8980238.679	8980249.257	10.57874	FALSE
1069	8980126.372	8980117.547	8.824887	TRUE
1080	8980022.582	8980023.91	1.327319	TRUE
1092	8979910.276	8979910.785	0.50965	TRUE
1105	8979788.15	8979795.546	7.395384	TRUE
1116	8979685.295	8979691.104	5.809555	TRUE
1129	8979562.98	8979569.968	6.987088	TRUE

Table B.3 - X Direction Data for Test Case 1

Distance on Y Direction

Time	Actual Distance	Predicted Distance	Difference	<10	<5
132	650709.7452	650705.1878	4.557395	TRUE	TRUE
228	650458.4446	650470.523	12.07834	FALSE	FALSE
240	650426.5695	650426.7212	0.151685	TRUE	TRUE
252	650394.5075	650394.846	0.338492	TRUE	TRUE
276	650331.1264	650330.937	0.189426	TRUE	TRUE
288	650302.4014	650301.1205	1.280978	TRUE	TRUE
324	650208.812	650213.0156	4.203547	TRUE	TRUE
348	650146.7285	650145.2415	1.487071	TRUE	TRUE
360	650114.6677	650116.7226	2.05487	TRUE	TRUE
372	650082.4212	650080.9646	1.456546	TRUE	TRUE
384	650047.0234	650048.7181	1.694718	TRUE	TRUE
396	650014.0363	650013.3203	0.715988	TRUE	TRUE
408	649982.1601	649980.3332	1.826825	TRUE	TRUE
432	649921.1885	649914.6268	6.561786	TRUE	FALSE
444	649889.1266	649889.4651	0.338492	TRUE	TRUE
456	649854.6562	649857.4031	2.746974	TRUE	TRUE
468	649825.7455	649822.9327	2.812772	TRUE	TRUE
479	649793.8704	649796.6777	2.80734	TRUE	TRUE
492	649761.6238	649759.4915	2.132344	TRUE	TRUE
504	649729.0059	649727.9207	1.085155	TRUE	TRUE
528	649666.3665	649665.4354	0.931096	TRUE	TRUE
539	649630.5984	649635.4843	4.885921	TRUE	TRUE
552	649595.016	649594.0746	0.941382	TRUE	TRUE
564	649564.6231	649563.2926	1.330544	TRUE	TRUE
576	649531.8206	649530.92	0.900571	TRUE	TRUE
588	649499.2038	649498.1175	1.086267	TRUE	TRUE
599	649466.5859	649470.136	3.550122	TRUE	TRUE
612	649434.3393	649430.0621	4.277229	TRUE	TRUE
624	649398.9427	649402.6159	3.673228	TRUE	TRUE
636	649368.7343	649365.2396	3.494749	TRUE	TRUE
648	649336.1175	649335.0313	1.086267	TRUE	TRUE

659	649305.7246	649307.0497	1.325111	TRUE	TRUE
672	649270.6982	649269.2009	1.497356	TRUE	TRUE
684	649234.5599	649236.9951	2.435276	TRUE	TRUE
696	649204.3515	649202.8364	1.515127	TRUE	TRUE
708	649171.1788	649170.6485	0.530292	TRUE	TRUE
719	649138.3763	649142.111	3.734706	TRUE	TRUE
732	649105.0178	649101.8525	3.16528	TRUE	TRUE
744	649068.1378	649071.3147	3.176947	TRUE	TRUE
756	649037.7448	649034.4347	3.310165	TRUE	TRUE
768	649004.7567	649004.0418	0.714876	TRUE	TRUE
779	648972.3244	648973.8745	1.550073	TRUE	TRUE
792	648940.0779	648935.8007	4.277229	TRUE	TRUE
804	648907.6468	648906.3748	1.271962	TRUE	TRUE
816	648875.5848	648873.9437	1.64113	TRUE	TRUE
829	648840.9298	648839.0611	1.868747	TRUE	TRUE
840	648808.4976	648811.862	3.364426	TRUE	TRUE
852	648776.6214	648774.7945	1.826825	TRUE	TRUE
865	648744.5605	648740.0976	4.462925	TRUE	TRUE
876	648711.9437	648713.6784	1.734657	TRUE	TRUE
889	648679.1412	648675.42	3.721255	TRUE	TRUE
900	648649.1186	648648.2591	0.859521	TRUE	TRUE
925	648579.6218	648578.7681	0.853631	TRUE	TRUE
936	648549.2277	648548.7396	0.48813	TRUE	TRUE
949	648512.9048	648512.704	0.200823	TRUE	TRUE
960	648482.3262	648483.691	1.364781	TRUE	TRUE
972	648448.5963	648448.6231	0.026794	TRUE	TRUE
985	648415.6092	648412.0726	3.536671	TRUE	TRUE
996	648380.0269	648382.9237	2.896824	TRUE	TRUE
1009	648347.2244	648345.4754	1.749	TRUE	TRUE
1020	648317.016	648314.5388	2.477227	TRUE	TRUE
1032	648281.4337	648285.1332	3.699594	TRUE	TRUE
1045	648248.0752	648246.8821	1.193026	TRUE	TRUE
1056	648215.2727	648217.193	1.920352	TRUE	TRUE
1069	648182.2845	648178.7489	3.535559	TRUE	TRUE
1080	648152.4475	648151.4023	1.045216	TRUE	TRUE
1092	648119.645	648118.7445	0.900571	TRUE	TRUE
1105	648083.877	648085.0935	1.216568	TRUE	TRUE
1116	648053.8543	648054.6631	0.808806	TRUE	TRUE
1129	648018.0874	648021.3364	3.24898	TRUE	TRUE

Table B.4 - Y Direction Data for Test Case 1

Test Case 2

Longitude

Time	Predicted Value	True Value	Error	Error Percentage
79	80.10501559	80.10523	0.00021441	0.0002676605%
1399	80.18899177	80.152685	0.036306767	0.0452970061%
1419	80.15306366	80.1531133	4.96419E-05	0.0000619339%
1440	80.15351178	80.15355	3.82153E-05	0.0000476776%
1459	80.1539104	80.15394	2.95978E-05	0.0000369262%
1539	80.15546351	80.15549	2.64855E-05	0.0000330426%
1560	80.15588638	80.1558767	9.684E-06	0.0000120815%
1579	80.15614935	80.15622	7.06472E-05	0.0000881369%
1599	80.15650829	80.1565733	6.50063E-05	0.0000810992%
1620	80.15687611	80.156955	7.88926E-05	0.0000984226%
1659	80.15751544	80.1575983	8.28639E-05	0.0001033762%
1680	80.15790076	80.15794	3.9241E-05	0.0000489546%
1699	80.15821391	80.158245	3.10867E-05	0.0000387817%
1719	80.15853242	80.1585617	2.92772E-05	0.0000365241%
1740	80.15886438	80.1589033	3.89233E-05	0.0000485576%
1800	80.15976995	80.1597867	1.67473E-05	0.0000208924%
1819	80.16006042	80.1600617	1.28382E-06	0.0000016016%

1859	80.16063853	80.160625	1.35319E-05	0.0000168810%
1899	80.16120158	80.1611617	3.98832E-05	0.0000497538%
1919	80.16135965	80.161445	8.53528E-05	0.0001064761%
1959	80.16183983	80.1619717	0.000131866	0.0001645000%
1999	80.16236768	80.1624883	0.000120617	0.0001504651%
2019	80.16268406	80.16274	5.59438E-05	0.0000697878%
2039	80.16293754	80.1629867	4.91643E-05	0.0000613305%
2059	80.16318465	80.163235	5.03528E-05	0.0000628128%
2079	80.16343273	80.1634633	3.05723E-05	0.0000381375%
2099	80.16366116	80.1637033	4.21351E-05	0.0000525614%
2120	80.16391113	80.16397	5.88739E-05	0.0000734418%
2140	80.16416773	80.1642117	4.39723E-05	0.0000548528%
2159	80.16439977	80.1644433	4.3529E-05	0.0000542997%
2180	80.16465012	80.1646933	4.31814E-05	0.0000538659%
2200	80.16489125	80.1649467	5.54528E-05	0.0000691734%
2219	80.1651348	80.1652033	6.8503E-05	0.0000854522%
2240	80.16550611	80.1655067	5.92554E-07	0.0000007392%
2260	80.16579479	80.1657917	3.08623E-06	0.0000038498%
2279	80.16606325	80.1660717	8.44827E-06	0.0000105385%
2300	80.16637385	80.166365	8.85418E-06	0.0000110448%
2320	80.16665081	80.16666	9.19185E-06	0.0000114659%
2339	80.16693088	80.1669583	2.7416E-05	0.0000341986%
2360	80.16726115	80.1672733	1.2149E-05	0.0000151545%
2380	80.16755973	80.1675767	1.69712E-05	0.0000211697%
2399	80.16784935	80.1678483	1.05283E-06	0.0000013133%
2420	80.16815098	80.1681683	1.73233E-05	0.0000216086%
2440	80.16845341	80.16849	3.65946E-05	0.0000456471%
2459	80.16876014	80.1687833	2.31615E-05	0.0000288910%
2480	80.16908402	80.1691133	2.92812E-05	0.0000365242%
2500	80.16940139	80.1694267	2.53138E-05	0.0000315753%
2519	80.16970014	80.1697133	1.31597E-05	0.0000164147%
2539	80.16999841	80.1700517	5.32946E-05	0.0000664770%
2560	80.17035385	80.1703883	3.44458E-05	0.0000429658%
2579	80.17065762	80.1706917	3.40847E-05	0.0000425151%
2620	80.17127283	80.171365	9.21702E-05	0.0001149665%
2639	80.17163588	80.17169	5.4116E-05	0.0000675001%
2659	80.17197697	80.1720333	5.63329E-05	0.0000702651%
2680	80.17232766	80.1724067	7.9044E-05	0.0000985925%
2700	80.17277827	80.1727483	2.99675E-05	0.0000373787%
2719	80.17310133	80.173125	2.36683E-05	0.0000295215%
2740	80.17352063	80.1735317	1.10736E-05	0.0000138120%
2759	80.17388473	80.1739167	3.19683E-05	0.0000398737%
2819	80.17504015	80.1752167	0.000176551	0.0002202060%
2839	80.17567599	80.17568	4.0081E-06	0.0000049991%
2879	80.17660494	80.17664	3.50599E-05	0.0000437284%
2899	80.17709929	80.1771317	3.24081E-05	0.0000404206%
2920	80.17760522	80.17767	6.47754E-05	0.0000807898%
2979	80.179026	80.17925	0.000223997	0.0002793699%
3239	80.18659343	80.18736	0.000766568	0.0009559709%
3260	80.18800662	80.1880683	6.16826E-05	0.0000769224%
3279	80.1886497	80.188715	6.52974E-05	0.0000814296%
3299	80.18932694	80.1893967	6.97563E-05	0.0000869894%
3320	80.19012936	80.19008	4.93587E-05	0.0000615522%
3339	80.19073385	80.190795	6.11549E-05	0.0000762618%
3360	80.19150658	80.1914817	2.48755E-05	0.0000310201%
3380	80.19216477	80.1921683	3.52767E-06	0.0000043990%
3399	80.19281728	80.192825	7.7227E-06	0.0000096302%
3420	80.19354217	80.1935567	1.45299E-05	0.0000181186%
3440	80.19423977	80.1942183	2.14723E-05	0.0000267754%

Table B.5 - Longitude Data for Test Case 2

Latitude

Time	Predicted Value	True Value	Error	Error Percentage
79	5.911872413	5.91172	0.000152413	0.00258%
1399	5.940038454	5.90957667	0.030461784	0.51546%
1419	5.909569128	5.90955333	1.57982E-05	0.00027%
1440	5.909572453	5.90952167	5.0783E-05	0.00086%
1459	5.90952675	5.90949333	3.342E-05	0.00057%
1539	5.909549844	5.90940333	0.000146514	0.00248%
1560	5.909381801	5.909375	6.80129E-06	0.00012%
1579	5.90936476	5.90935667	8.08999E-06	0.00014%
1599	5.909365254	5.90934	2.52536E-05	0.00043%
1620	5.909354086	5.90931667	3.74156E-05	0.00063%
1659	5.909306152	5.90927833	2.78221E-05	0.00047%
1680	5.909282303	5.90926167	2.06329E-05	0.00035%
1699	5.909288012	5.90923833	4.96822E-05	0.00084%
1719	5.90923242	5.90922167	1.07498E-05	0.00018%
1740	5.909230705	5.90920833	2.23753E-05	0.00038%
1800	5.909238057	5.909155	8.30568E-05	0.00141%
1819	5.909158554	5.90914333	1.5224E-05	0.00026%
1859	5.909171195	5.90912	5.11949E-05	0.00087%
1899	5.909138254	5.909095	4.32536E-05	0.00073%
1919	5.909109995	5.90908	2.99946E-05	0.00051%
1959	5.909085423	5.90906667	1.87527E-05	0.00032%
1999	5.909116853	5.909045	7.18527E-05	0.00122%
2019	5.909034309	5.90903833	4.02128E-06	0.00007%
2039	5.909043706	5.90903	1.37062E-05	0.00023%
2059	5.909051383	5.90902833	2.30529E-05	0.00039%
2079	5.909036916	5.90902833	8.58603E-06	0.00015%
2099	5.909040122	5.90903167	8.45244E-06	0.00014%
2120	5.909054143	5.90902833	2.58134E-05	0.00044%
2140	5.909036916	5.90902833	8.58603E-06	0.00015%
2159	5.909042556	5.909025	1.75556E-05	0.00030%
2180	5.90902392	5.90902667	2.74964E-06	0.00005%
2200	5.909048053	5.909025	2.30529E-05	0.00039%
2219	5.909042263	5.90901833	2.39332E-05	0.00041%
2240	5.909032416	5.908985	4.74156E-05	0.00080%
2260	5.908988762	5.90895333	3.5432E-05	0.00060%
2279	5.908933866	5.90892833	5.53554E-06	0.00009%
2300	5.90892723	5.908895	3.22297E-05	0.00055%
2320	5.908874531	5.908865	9.53128E-06	0.00016%
2339	5.90884092	5.90883667	4.25008E-06	0.00007%
2360	5.908855793	5.908795	6.0793E-05	0.00103%
2380	5.908779389	5.908765	1.43885E-05	0.00024%
2399	5.90875476	5.90873667	1.809E-05	0.00031%
2420	5.908745705	5.90869667	4.90353E-05	0.00083%
2440	5.908671342	5.908655	1.63423E-05	0.00028%
2459	5.908626305	5.90861833	7.97515E-06	0.00013%
2480	5.90860196	5.90858333	1.86303E-05	0.00032%
2500	5.908587092	5.90854833	3.8762E-05	0.00066%
2519	5.908547294	5.90851333	3.39637E-05	0.00057%
2539	5.908488002	5.90847667	1.13323E-05	0.00019%
2560	5.90847557	5.90842	5.55697E-05	0.00094%
2579	5.908386693	5.90837167	1.50229E-05	0.00025%
2620	5.908300985	5.90827667	2.43148E-05	0.00041%
2639	5.90825259	5.90823667	1.59201E-05	0.00027%
2659	5.908225912	5.90817333	5.2582E-05	0.00089%
2680	5.9081213	5.90811167	9.63016E-06	0.00016%
2700	5.908052025	5.908055	2.97542E-06	0.00005%
2719	5.907998325	5.90799667	1.65486E-06	0.00003%
2740	5.907968303	5.90791833	4.99733E-05	0.00085%
2759	5.907861655	5.90785	1.16549E-05	0.00020%
2819	5.907714074	5.90758667	0.000127404	0.00216%

2839	5.907499007	5.90748667	1.23366E-05	0.00021%
2879	5.907344914	5.907265	7.99136E-05	0.00135%
2899	5.907177337	5.907135	4.23366E-05	0.00072%
2920	5.907009037	5.90700167	7.36673E-06	0.00012%
2979	5.906746387	5.90657333	0.000173057	0.00293%
3239	5.905196918	5.90392	0.001276918	0.02163%
3260	5.90370061	5.903635	6.56099E-05	0.00111%
3279	5.903426228	5.90337667	4.95585E-05	0.00084%
3299	5.903156884	5.903095	6.18837E-05	0.00105%
3320	5.902828555	5.902795	3.35554E-05	0.00057%
3339	5.902530754	5.90246667	6.40841E-05	0.00109%
3360	5.902149283	5.90215167	2.3871E-06	0.00004%
3380	5.901861403	5.90183333	2.80726E-05	0.00048%
3399	5.901557611	5.90153167	2.59411E-05	0.00044%
3420	5.901226857	5.90119	3.6857E-05	0.00062%
3440	5.900899733	5.90088167	1.80626E-05	0.00031%

Table B.6 - Latitude Data for Test Case 2

Distance on X Direction

Time	Actual Distance	Predicted Distance	Difference	<10	<5
79	8907295.174	8907271.332	23.84131195	FALSE	FALSE
1399	8912571.929	8916609.057	4037.128246	FALSE	FALSE
1419	8912619.554	8912614.034	5.51993042	TRUE	FALSE
1440	8912668.113	8912663.863	4.24934648	TRUE	TRUE
1459	8912711.479	8912708.187	3.291124089	TRUE	TRUE
1539	8912883.831	8912880.886	2.945051691	TRUE	TRUE
1560	8912926.83	8912927.907	1.07681142	TRUE	TRUE
1579	8912965.003	8912957.147	7.855606869	TRUE	FALSE
1599	8913004.288	8912997.06	7.228375031	TRUE	FALSE
1620	8913046.731	8913037.959	8.77245177	TRUE	FALSE
1659	8913118.263	8913109.049	9.214043479	TRUE	FALSE
1680	8913156.258	8913151.895	4.36340346	TRUE	TRUE
1699	8913190.173	8913186.716	3.45668765	TRUE	TRUE
1719	8913225.388	8913222.133	3.255473919	TRUE	TRUE
1740	8913263.372	8913259.044	4.32806794	TRUE	TRUE
1800	8913361.602	8913359.74	1.86221871	TRUE	TRUE
1819	8913392.181	8913392.038	0.14275475	TRUE	TRUE
1859	8913454.817	8913456.321	1.504681729	TRUE	TRUE
1899	8913514.495	8913518.93	4.43481409	TRUE	TRUE
1919	8913545.996	8913536.506	9.490798021	TRUE	FALSE
1959	8913604.563	8913589.9	14.66287997	FALSE	FALSE
1999	8913662.006	8913648.594	13.41195025	FALSE	FALSE
2019	8913689.994	8913683.773	6.220665621	TRUE	FALSE
2039	8913717.426	8913711.959	5.466823401	TRUE	FALSE
2059	8913745.035	8913739.436	5.598975601	TRUE	FALSE
2079	8913770.421	8913767.022	3.399486789	TRUE	TRUE
2099	8913797.108	8913792.423	4.68521259	TRUE	TRUE
2120	8913826.764	8913820.217	6.54647707	TRUE	FALSE
2140	8913853.64	8913848.75	4.889498809	TRUE	TRUE
2159	8913879.392	8913874.552	4.840209231	TRUE	TRUE
2180	8913907.191	8913902.389	4.801553881	TRUE	TRUE
2200	8913935.368	8913929.202	6.16606972	TRUE	FALSE
2219	8913963.9	8913956.283	7.617182489	TRUE	FALSE
2240	8913997.637	8913997.571	0.06588901	TRUE	TRUE
2260	8914029.327	8914029.671	0.343173001	TRUE	TRUE
2279	8914060.462	8914059.523	0.939404521	TRUE	TRUE
2300	8914093.076	8914094.06	0.984539511	TRUE	TRUE
2320	8914125.878	8914124.856	1.02208695	TRUE	TRUE
2339	8914159.047	8914155.999	3.048518099	TRUE	TRUE
2360	8914194.074	8914192.723	1.3509054	TRUE	TRUE
2380	8914227.81	8914225.923	1.887113979	TRUE	TRUE
2399	8914258.011	8914258.128	0.117069371	TRUE	TRUE

2420	8914293.593	8914291.667	1.92625754	TRUE	TRUE
2440	8914329.365	8914325.296	4.06913583	TRUE	TRUE
2459	8914361.978	8914359.403	2.57544432	TRUE	TRUE
2480	8914398.673	8914395.417	3.25591643	TRUE	TRUE
2500	8914433.521	8914430.706	2.81476292	TRUE	TRUE
2519	8914465.39	8914463.926	1.463286851	TRUE	TRUE
2539	8914503.018	8914497.092	5.926091099	TRUE	FALSE
2560	8914540.446	8914536.616	3.83020081	TRUE	TRUE
2579	8914574.183	8914570.393	3.79004159	TRUE	TRUE
2620	8914649.05	8914638.801	10.24886126	FALSE	FALSE
2639	8914685.189	8914679.171	6.017422639	TRUE	FALSE
2659	8914723.362	8914717.098	6.26393747	TRUE	FALSE
2680	8914764.882	8914756.093	8.789290229	TRUE	FALSE
2700	8914802.866	8914806.198	3.33223789	TRUE	TRUE
2719	8914844.753	8914842.121	2.631795419	TRUE	TRUE
2740	8914889.976	8914888.745	1.23132813	TRUE	TRUE
2759	8914932.786	8914929.232	3.554713311	TRUE	TRUE
2819	8915077.34	8915057.708	19.63153222	FALSE	FALSE
2839	8915128.856	8915128.411	0.44568049	TRUE	TRUE
2879	8915235.603	8915231.705	3.898486029	TRUE	TRUE
2899	8915290.278	8915286.674	3.603616411	TRUE	TRUE
2920	8915350.134	8915342.931	7.202692239	TRUE	FALSE
2979	8915525.822	8915500.915	24.90729456	FALSE	FALSE
3239	8916427.613	8916342.375	85.23845739	FALSE	FALSE
3260	8916506.372	8916499.514	6.85879402	TRUE	FALSE
3279	8916578.282	8916571.021	7.26073537	TRUE	FALSE
3299	8916654.084	8916646.327	7.75654595	TRUE	FALSE
3320	8916730.063	8916735.552	5.488440959	TRUE	FALSE
3339	8916809.568	8916802.767	6.800116161	TRUE	FALSE
3360	8916885.925	8916888.691	2.76603015	TRUE	TRUE
3380	8916962.272	8916961.879	0.39225954	TRUE	TRUE
3399	8917035.293	8917034.435	0.858724931	TRUE	TRUE
3420	8917116.655	8917115.039	1.61565344	TRUE	TRUE
3440	8917190.221	8917192.609	2.38761363	TRUE	TRUE

Table B.7 - X Direction Data for Test Case 2

Distance on Y Direction

Time	Actual Distance	Predicted Distance	Difference	<10	<5
79	657353.2717	657370.2193	16.94759	FALSE	FALSE
1399	657114.9443	660502.1401	3387.196	FALSE	FALSE
1419	657112.349	657114.1057	1.756681	TRUE	TRUE
1440	657108.8286	657114.4754	5.646808	TRUE	FALSE
1459	657105.6773	657109.3935	3.716138	TRUE	TRUE
1539	657095.6698	657111.9614	16.29161	FALSE	FALSE
1560	657092.5196	657093.2759	0.756269	TRUE	TRUE
1579	657090.4814	657091.381	0.899566	TRUE	TRUE
1599	657088.6278	657091.4359	2.80807	TRUE	TRUE
1620	657086.0336	657090.1941	4.160429	TRUE	TRUE
1659	657081.7704	657084.8641	3.093676	TRUE	TRUE
1680	657079.9179	657082.2122	2.294271	TRUE	TRUE
1699	657077.3226	657082.847	5.524403	TRUE	FALSE
1719	657075.4701	657076.6654	1.195326	TRUE	TRUE
1740	657073.9868	657076.4748	2.488015	TRUE	TRUE
1800	657068.0568	657077.2922	9.235491	TRUE	FALSE
1819	657066.7591	657068.4519	1.692826	TRUE	TRUE
1859	657064.1649	657069.8575	5.692613	TRUE	FALSE
1899	657061.3851	657066.1946	4.809582	TRUE	TRUE

1919	657059.7171	657063.0524	3.335248	TRUE	TRUE
1959	657058.2349	657060.3201	2.085208	TRUE	TRUE
1999	657055.8253	657063.815	7.989654	TRUE	FALSE
2019	657055.0836	657054.6365	0.447146	TRUE	TRUE
2039	657054.1574	657055.6815	1.524062	TRUE	TRUE
2059	657053.9717	657056.5351	2.56337	TRUE	TRUE
2079	657053.9717	657054.9264	0.954723	TRUE	TRUE
2099	657054.3431	657055.283	0.939869	TRUE	TRUE
2120	657053.9717	657056.842	2.870318	TRUE	TRUE
2140	657053.9717	657054.9264	0.954723	TRUE	TRUE
2159	657053.6014	657055.5535	1.952093	TRUE	TRUE
2180	657053.7871	657053.4814	0.305747	TRUE	TRUE
2200	657053.6014	657056.1648	2.56337	TRUE	TRUE
2219	657052.8597	657055.521	2.661246	TRUE	TRUE
2240	657049.1536	657054.426	5.272378	TRUE	FALSE
2260	657045.6321	657049.5719	3.939864	TRUE	TRUE
2279	657042.8522	657043.4677	0.615524	TRUE	TRUE
2300	657039.1461	657042.7299	3.583779	TRUE	TRUE
2320	657035.8102	657036.8701	1.05983	TRUE	TRUE
2339	657032.6601	657033.1327	0.472587	TRUE	TRUE
2360	657028.0266	657034.7865	6.759869	TRUE	FALSE
2380	657024.6907	657026.2907	1.599929	TRUE	TRUE
2399	657021.5406	657023.5521	2.011515	TRUE	TRUE
2420	657017.0928	657022.5453	5.452472	TRUE	FALSE
2440	657012.4593	657014.2765	1.817183	TRUE	TRUE
2459	657008.3818	657009.2686	0.886796	TRUE	TRUE
2480	657004.49	657006.5615	2.071594	TRUE	TRUE
2500	657000.5981	657004.9083	4.310143	TRUE	TRUE
2519	656996.7063	657000.4829	3.776596	TRUE	TRUE
2539	656992.6299	656993.89	1.260096	TRUE	TRUE
2560	656986.3285	656992.5076	6.179068	TRUE	FALSE
2579	656980.9544	656982.6249	1.670474	TRUE	TRUE
2620	656970.3909	656973.0946	2.703685	TRUE	TRUE
2639	656965.9431	656967.7134	1.770232	TRUE	TRUE
2659	656958.9	656964.7469	5.846849	TRUE	FALSE
2680	656952.0438	656953.1146	1.070825	TRUE	TRUE
2700	656945.7423	656945.4115	0.330851	TRUE	TRUE
2719	656939.2563	656939.4403	0.184012	TRUE	TRUE
2740	656930.5453	656936.1021	5.556778	TRUE	FALSE
2759	656922.9474	656924.2433	1.295961	TRUE	TRUE
2819	656893.6664	656907.8331	14.16665	FALSE	FALSE
2839	656882.5469	656883.9187	1.371772	TRUE	TRUE
2879	656857.8983	656866.7843	8.885989	TRUE	FALSE
2899	656843.443	656848.1506	4.70762	TRUE	TRUE
2920	656828.6174	656829.4365	0.819143	TRUE	TRUE
2979	656780.9882	656800.2312	19.24308	FALSE	FALSE
3239	656485.9513	656627.9381	141.9868	FALSE	FALSE
3260	656454.2608	656461.5562	7.295487	TRUE	FALSE
3279	656425.5358	656431.0464	5.510651	TRUE	FALSE
3299	656394.2155	656401.0967	6.881157	TRUE	FALSE
3320	656360.857	656364.5882	3.73119	TRUE	TRUE
3339	656324.3484	656331.4742	7.125822	TRUE	FALSE
3360	656289.322	656289.0566	0.265433	TRUE	TRUE
3380	656253.9242	656257.0457	3.121533	TRUE	TRUE
3399	656220.3811	656223.2657	2.884515	TRUE	TRUE
3420	656182.3892	656186.4875	4.098307	TRUE	TRUE
3440	656148.1044	656150.1129	2.008471	TRUE	TRUE

Table B.8 - Y Direction Data for Test Case 2